Check for
updates

Algorithms for Nonnegative Matrix Factorization with the Kullback–Leibler Divergence

Le Thi Khanh Hien[1] · Nicolas Gillis[1]

Abstract

Nonnegative matrix factorization (NMF) is a standard linear dimensionality reduction technique for nonnegative data sets. In order to measure the discrepancy between the input data and the low-rank approximation, the Kullback–Leibler (KL) divergence is one of the most widely used objective function for NMF. It corresponds to the maximum likelihood estimator when the underlying statistics of the observed data sample follows a Poisson distribution, and KL NMF is particularly meaningful for count data sets, such as documents. In this paper, we first collect important properties of the KL objective function that are essential to study the convergence of KL NMF algorithms. Second, together with reviewing existing algorithms for solving KL NMF, we propose three new algorithms that guarantee the non-increasingness of the objective function. We also provide a global convergence guarantee for one of our proposed algorithms. Finally, we conduct extensive numerical experiments to provide a comprehensive picture of the performances of the KL NMF algorithms.

Keywords Nonnegative matrix factorization · Kullback–Leibler divergence · Poisson distribution · Algorithms

1 Introduction

Given a nonnegative matrix data $V \in \mathbb{R}_+^{m \times n}$ and a positive integer $r \leq \min(m, n)$, nonnegative matrix factorization (NMF) is the problem of finding $W \in \mathbb{R}_+^{m \times r}$ and $H \in \mathbb{R}_+^{r \times n}$ such that $V \approx WH$. The quality of the approximation is measured using an objective function, which typically has the form

$$D(V|WH) \triangleq \sum_{i=1}^{m} \sum_{j=1}^{n} d\left(V_{ij}|(WH)_{ij}\right),$$

✉ Nicolas Gillis
nicolas.gillis@umons.ac.be

Le Thi Khanh Hien
thikhanhhien.le@umons.ac.be

[1] Department of Mathematics and Operational Research, Faculté Polytechnique, Université de Mons, Rue de Houdain 9, 7000 Mons, Belgium

where $d(x|y)$ is a scalar cost function such that $d(x|y) \geq 0$ for all x, $y \geq 0$ and $d(x|y) = 0$ if and only if $x = y$. NMF is then written as the following problem

$$\min_{W \in \mathbb{R}_+^{m \times r}, H \in \mathbb{R}_+^{r \times m}} D(V|WH).$$

The most widely used class of the scalar cost functions $d(x|y)$ is the β-divergence [15]

$$d_\beta(x|y) = \begin{cases} \frac{1}{\beta(\beta-1)}\left(x^\beta + (\beta-1)y^\beta - \beta x y^{\beta-1}\right) & \text{if } \beta \in \mathbb{R} \setminus \{0, 1\}, \\ x \log \frac{x}{y} - x + y & \text{if } \beta = 1, \\ \frac{x}{y} - \log \frac{x}{y} - 1 & \text{if } \beta = 0. \end{cases}$$

In this paper, we are interested in the Kullback–Leibler (KL) divergence (also known as the I-divergence), which is the β-divergence with $\beta = 1$. The KL NMF problem can be rewritten as follows:

$$\min_{W \in \mathbb{R}_+^{m \times r}, H \in \mathbb{R}_+^{r \times m}} D_{\mathrm{KL}}(V|WH), \tag{1}$$

where

$$D_{\mathrm{KL}}(V|WH) := \sum_{i=1}^m \sum_{j=1}^n \left((WH)_{ij} - V_{ij} \log(WH)_{ij}\right) + \sum_{i=1}^m \sum_{j=1}^n \left(V_{ij} \log V_{ij} - V_{ij}\right).$$

With the convention that $0 \times \log 0 = 0$ and $a \times \log 0 = -\infty$ for $a > 0$, the objective function in (1) is well-defined and it is an extended-value function, that is, $D_{\mathrm{KL}} \in [0, +\infty]$. KL NMF (1) is well-posed, that is, a solution always exists (see Sect. 2.1 for more details), but the solution is in general non-unique even when removing the scaling and permutation ambiguities of the low-rank approximation WH; see [17] and the references therein.

1.1 Motivation

Since the seminal paper of Lee and Seung [31], NMF has been shown to be a very powerful model to extract perceptually meaningful features from high-dimensional data sets. Applications include facial feature extraction [22,31], recovery and document classification [11,31,43], unmixing hyperspectral images [4,39]; see also [9,17,18] and the references therein. The most widely used objective function for NMF is the Frobenius norm $\|V - WH\|_F^2$ which corresponds to the β-divergence with $\beta = 2$. For nonnegative data sets, the Frobenius norm is however not the theoretically most reasonable choice. In fact, it corresponds to the maximum likelihood estimator in the presence of additive i.i.d. Gaussian noise. Under this noise distribution, observing negative entries in a data set has a positive probability. Moreover, many nonnegative data sets are sparse in which case Gaussian noise is clearly not appropriate [8,14,31].

Let us assume that V_{ij} is a sample of the random variable $\tilde{V}_{ij}$ following the Poisson distribution of parameter $(WH)_{ij}$, that is,

$$\mathbb{P}(\tilde{V}_{ij} = k) = \frac{(WH)_{ij}^k e^{-(WH)_{ij}}}{k!} \text{ for } k = 0, 1, 2, \ldots$$

Then the maximum likelihood estimator of W and H, given V, is the solution of (1). The Poisson distribution is particularly well suited for integer-valued data sets, such a documents represented by vector of word counts [8,33], or images which can be interpreted as a photon counting process [23]. KL NMF has also been used successfully in bioinformatics [35], e.g.,

to cluster samples of RNA sequencing gene expression data [10]. It is worth noting that KL NMF also makes sense when V contains non-integer entries. In that case, V_{ij} can be interpreted as the average of several samples of the random variable $\tilde{V}_{ij}$.

In the literature, NMF with the Frobenius norm (Fro NMF) has been thoroughly studied and well-documented. Among algorithms for Fro NMF, the block coordinate (BC) methods, which update one block of the variables at a time, have the best performance in practice, see for example [1,9,18,24,28,49] and the references therein. Note that one "block" here can be a full matrix, a column, a row, or even just a scalar component of the matrices W or H. The objective function of Fro NMF has several nice properties that allow us to apply some advanced development of BC methods for solving composite block-wise convex optimization problems, which subsume Fro NMF as a special case, to derive very efficient algorithms with some rigorous convergence guarantee [24,49]. One of the most important properties is that the gradients of the objective with respect to W and H, that is, $W \mapsto \nabla_W D_{\text{Fro}}$ and $H \mapsto \nabla_H D_{\text{Fro}}$, are Lipschitz-continuous over $W \in \mathbb{R}_+^{m \times r}$, $H \in \mathbb{R}_+^{r \times n}$. Although the objective of KL NMF, similarly to D_{Fro}, is block-wise convex (that is, the function $W \mapsto D_{\text{KL}}$ and $H \mapsto D_{\text{KL}}$ are convex), D_{KL} is even not differentiable at W or H when $(WH)_{ij} = 0$ for some i, j. Since D_{KL} does not possess the nice smooth properties of Fro NMF, the extension of the analysis of BC methods from Fro NMF to KL NMF is restricted. Proposing a good algorithm for solving KL NMF is therefore a more difficult task compared to the Fro NMF. In fact, there are much fewer papers studying algorithms for KL NMF in the literature; in particular, algorithms with convergence guarantee are scarce. To the best of our knowledge, the multiplicative updates (MU) with some modification of KL NMF (see Sect. 3.1 for more details) is the only algorithm that has a subsequential convergence guarantee. Moreover, the MU are the most widely used algorithm for KL NMF, while it is well known that, for Fro NMF, the MU are slow and should not be used; see [18] and the references therein. These observations motivate this work which analyses in details the properties and algorithms for KL NMF (1).

1.2 Contribution and Outline

Our main contribution is threefold.

1. In Sect. 2, we present important properties of KL NMF (1) for the convergence analysis of KL-NMF algorithms. Some are well-known while others have not yet been highlighted in the literature, to the best of our knowledge.
2. While existing algorithms for (1) are briefly reviewed in Sect. 3, we present two new algorithms in Sect. 4. They are (i) a block mirror descent method (BMD), for which we prove the global convergence of its generated sequence to a stationary point of a slightly perturbed version of KL NMF, and (ii) a scalar Newton-type algorithm which monotonically decreases the objective function. To the best of our knowledge, BMD is the first algorithm for KL NMF that has a global convergence guarantee. We also propose a hybridization between the scalar Newton algorithm and the MU for which the objective function is guaranteed to be non-increasing.
3. In Sect. 5, we perform extensive numerical experiments on synthetic as well as real data sets to compare the performance of the algorithms. To the best of our knowledge, this is the first time such a comparison is performed. It provides a good picture on the performance of the algorithms in different scenarios.

We hope that the paper will be a good reference for whomever is using KL NMF, or is interested in KL NMF algorithms.

2 Some Properties of KL NMF

In the remainder of the paper, for simplicity, we denote $D(V|WH) = D_{KL}(V|WH)$ since we only consider the KL divergence. The objective function of KL NMF (1) is not finite at every point of its constraint set since $D(V|WH) = \infty$ when $V_{ij} > 0$ and $(WH)_{ij} = 0$. This fact makes the convergence analysis of algorithms for (1) very challenging. Hence, in parallel with considering KL NMF (1), we propose to study the following perturbed version

$$\min_{W,H} \quad D(V|WH)$$
$$\text{such that } W_{ik} \geq \varepsilon, \, H_{kj} \geq \varepsilon \text{ for } i \in [m], \, j \in [n], \, k \in [r], \tag{2}$$

where $\varepsilon \geq 0$, and $[n]$ denotes the set $\{1, \dots, n\}$. Problem (2) is equivalent to KL NMF (1) when $\varepsilon = 0$. When $\varepsilon > 0$, the objective of (2) is finite at every point of its constraint set. Moreover, a solution of (2) has all its entries strictly positive when $\varepsilon > 0$. However, for ε sufficiently small (we recommend to use the machine precision), then such entries can be considered as zeros, which will not influence the objective function of KL NMF much; see Proposition 1 below.

2.1 Existence of Solutions

The following proposition proves the existence of solutions of Problem (2) and provides a connection between the optimal value of (1) and its perturbed problem (2) with $\varepsilon > 0$. The proof is given in Appendix 1.

Proposition 1 *(A) Given $\varepsilon \geq 0$ and a nonnegative matrix V, Problem (2) attains its minimum, that is, it has at least one optimal solution.*

(B) Let $\nu = \sum_{i=1}^{m} \sum_{j=1}^{n} V_{ij}$. Denote $D^*(V, \varepsilon)$ be the optimal value of Problem (2). We have $D^*(V, \varepsilon) \leq D^*(V, 0) + (\min\{n + mr, m + nr\}\sqrt{\nu} + mn\varepsilon)\varepsilon$.

Proposition 1(A) is known for the case $\varepsilon = 0$; see for example [16, Proposition 2.1], while Proposition 1(B) is new. Given $\delta \geq 0$, Proposition 1(B) shows that by choosing ε such that $\min\{n + mr, m + nr\}\sqrt{\nu}\varepsilon + mn\varepsilon^2 = \delta$, an optimal solution of (2) is a δ-optimal solution of KL NMF (1). Proposition 1(B) shows that, for ε sufficiently small, the objective function of (2) is not significantly larger than that of (1). However, it says nothing on the corresponding optimal solutions. To the best of our knowledge, no sensitivity analysis of the optimal solutions exist in the NMF literature. In fact, there could exist several isolated optimal solutions (see, e.g., [19], for some examples) which makes a sensitivity analysis difficult in the general case because the optimal solution can change drastically for an arbitrarily small perturbation (leading to an infinite condition number of the problem). This is an interesting direction of further research.

2.2 KKT and Stationary Points

A pair (W^*, H^*) is a Karush–Kuhn–Tucker (KKT) point of (2) if it satisfies the KKT optimality condition of (2), that is, if the objective function $f(W, H) := D(V, WH)$ is differentiable

at (W^*, H^*), and for all $i \in [m]$ and $k \in [r]$: $W_{ik}^* \geq \varepsilon$ and

$$\frac{\partial f(W^*, H^*)}{\partial W_{ik}} = \sum_{j=1}^{n} H_{kj}^* - \sum_{\{j \mid V_{ij} > 0\}}^{n} V_{ij} \frac{H_{kj}^*}{(W^* H^*)_{ij}} \geq 0, \quad (3a)$$

$$(W_{ik}^* - \varepsilon) \frac{\partial f(W^*, H^*)}{\partial W_{ik}} = 0, \quad (3b)$$

$$(3)$$

and similarly for H^*, by symmetry. A pair (W^*, H^*) is a stationary point of (2) if it is a feasible point of (2) that lies in the domain of f and for all $i \in [m]$ and $k \in [r]$, it satisfies

$$\frac{\partial f(W^*, H^*)}{\partial W_{ik}} (W_{ik} - W_{ik}^*) \geq 0 \text{ for any } W_{ik} \geq \varepsilon, \quad (4)$$

and similarly for H^*. For Problem (2), it turns out that KKT points and stationary points coincide. Since $W_{ik}^* - \varepsilon \geq 0$, (3a) and (3b) hold if and only if (4) holds. Indeed, let (W^*, H^*) satisfy (4) then choosing W_{ik} in (4) to be $W_{ik}^* + 1$ gives (3a) while choosing W_{ik} in (4) to be ε or $2W_{ik}^* - \varepsilon$ gives (3b). The reverse direction is obvious. The same reasoning applies to H^*.

The following proposition provides an interesting property of KKT points of KL NMF (1), see [25, Theorem 1] for its proof. Note that this property does not hold for its perturbed variant (2) with $\varepsilon > 0$.

Proposition 2 *If (W^*, H^*) is a KKT point of* (1)*, then $W^* H^*$ preserves the row sums and the column sums of V, that is,*

$$V e = (W^* H^*) e \quad and \quad e^\top V = e^\top (W^* H^*),$$

where e denotes the vector of all ones of appropriate dimension.

Let us define the following notion of a scaled pair (W, H).

Definition 1 We say (W, H) is scaled if the optimal solution of the problem $\min_{\alpha \in \mathbb{R}} D(V, \alpha W H)$ is equal to 1.

Hence WH is scaled implies that one cannot multiply WH by a constant α (that is, one cannot scale WH) to reduce the error $KL(X, WH)$.

By Proposition 2, if (W^*, H^*) is a KKT point of (1) then $\sum_{i=1}^{m} \sum_{j=1}^{n} V_{ij} = \sum_{i=1}^{m} \sum_{j=1}^{n} (W^* H^*)_{ij}$ and hence is scaled. Moreover, we have the following result.

Proposition 3 *A pair (W, H) is scaled if and only if*

$$\sum_{i=1}^{m} \sum_{j=1}^{n} V_{ij} = \sum_{i=1}^{m} \sum_{j=1}^{n} (WH)_{ij}.$$

Proof We have

$$D(V, \alpha W H) = \alpha \sum_{i,j} (WH)_{ij} - \sum_{i,j} V_{ij} \log(WH)_{ij}$$

$$- \sum_{i,j} V_{ij} \log \alpha + \sum_{i,j} (V_{ij} \log V_{ij} - V_{ij}).$$

The result follows from the equation $\nabla_\alpha D(V, \alpha W H) = 0$.   □

Proposition 3, although simple to prove, is not present explicitly in the literature. However, it is rather interesting from a practical point of view: any feasible solution (W, H) to KL NMF can be improved simply by scaling it. In fact, combining Definition 1 and Proposition 3, we can compute

$$\alpha^* = \operatorname{argmin}_{\alpha \in \mathbb{R}} D(V, \alpha W H) = \frac{e^\top V e}{e^\top (W H) e},$$

and scale $W \leftarrow \alpha^* W$ to improve the feasible solution (W, H). This could be used within any algorithm for KL NMF.

2.3 Relative Smoothness

In Sect. 4.1, we will propose a new algorithm that globally converges to a stationary point of (2) with $\varepsilon > 0$, namely a block mirror descent method. An instrumental property of (2) to prove such a result is relative smoothness. Let us describe this property in details.

The objective $D(V|WH)$ is convex in each block variable W and H, but it is not jointly convex in (W, H). Furthermore, $D(V|WH)$ does not possess the Lipschitz smoothness property, that is, the derivative of $D(V|WH)$ with respect to W or H is not Lipschitz continuous. Recently, the authors in [3] and [37] introduce the notion of relative smoothness that is a generalization of the Lipschitz smoothness.

Definition 2 [37, Definition 1.1] Let $\kappa(\cdot)$ be any given differentiable convex function defined on a convex set Q. The convex function $g(\cdot)$ is L-smooth relative to $\kappa(\cdot)$ on Q if for any $x, y \in \operatorname{int} Q$, there is a scalar L for which

$$g(y) \leq g(x) + \langle \nabla g(x), y - x \rangle + L \mathcal{B}_\kappa(y, x), \tag{5}$$

where

$$\mathcal{B}_\kappa(y, x) := \kappa(y) - \kappa(x) - \langle \nabla \kappa(x), y - x \rangle, \text{ for all } x, y \in Q. \tag{6}$$

The following proposition shows that, when restricted to a column of H, the KL objective function is a relative smooth function. Since $D(V|WH) = D(V^\top|H^\top W^\top)$, this result implies that the KL objective function when restricted to a row of W is also relative smooth.

Proposition 4 [3, Lemma 7] *Let $v \in \mathbb{R}^m_+$ and $W \in \mathbb{R}^{m \times r}_+$, and*

$$D(v|Wh) := \sum_{i=1}^m \Big((Wh)_i - v_i \log(Wh)_i + v_i \log v_i - v_i \Big). \tag{7}$$

Then the function $h \mapsto D(v|Wh)$ is relative smooth to $\kappa(h) = -\sum_{j=1}^r \log h_j$ with the relative smooth constant $L = \|v\|_1$.

2.4 Self-concordant Properties

In Sect. 4.2, we will propose a new monotone algorithm to solve (2), namely a scalar Newton-type algorithm. An instrumental property of (2) to prove the monotonicity is the self-concordant properties of its objective function. Let us first define a self-concordant function. We adopt the definition in [40, Chapter 5].

Consider a closed convex function $g : \mathbb{E} \to \mathbb{R}$ with an *open domain*. Fixing a point $x \in \operatorname{dom}(g)$ and a direction $d \in \mathbb{E}$, let $\varphi_x(\tau) = g(x + \tau d)$. We define $\mathbf{D} g(x)[d] = \varphi'_x(0)$, $\mathbf{D}^2 g(x)[d, d] = \varphi''_x(0)$, and $\mathbf{D}^3 g(x)[d, d, d] = \varphi'''_x(0)$.

Definition 3 We say the function g belongs to the class $\mathcal{F}_M$ of self-concordant functions with parameter $M \geq 0$, if

$$\left| \mathbf{D}^3 g(x)[d, d, d] \right| \leq 2M \|d\|^3_{\nabla^2 g(x)},$$

where $\|d\|^2_{\nabla^2 g(x)} = \langle \nabla^2 g(x) d, d \rangle = \varphi''_x(0)$. The function g is called standard self-concordant when $M = 1$.

The scalar function $g(x) = -\log(x)$ is a standard self-concordant function; see [40, Example 5.1.1]. The following proposition provides some useful properties to determine the self-concordant constant of a function.

Proposition 5 [40, Theorems 5.1.1 and 5.1.2]

(i) *Let g_1, g_2 be self-concordant functions with constants M_1, M_2. Then the function $g(x) = \alpha g_1(x) + \beta g_2(x)$, where α, β are positive constants, is self-concordant with constant $M_g = \max\left\{ \frac{1}{\sqrt{\alpha}} M_1, \frac{1}{\sqrt{\beta}} M_2 \right\}$.*

(ii) *If $g(\cdot)$ is self-concordant with constant M_g, then $\phi(x) = g(\mathcal{A}(x))$, where $\mathcal{A}(x) = Ax + b$ is a linear operator, is also self-concordant with constant $M_\phi = M_g$.*

Using Proposition 5, we can prove that the objective of the perturbed KL NMF problem (2) is self-concordant with respect to a single entry of W and H.

Proposition 6 *Given $V \in \mathbb{R}_+^{m \times n}$, $W \in \mathbb{R}_+^{m \times r}$ and $H \in \mathbb{R}_+^{r \times n}$, the scalar function*

$$H_{kj} \mapsto D(V|WH) = \sum_{i=1}^{m} \left(\sum_{a=1}^{n} W_{ia} H_{aj} - V_{ij} \log \sum_{a=1}^{n} W_{ia} H_{aj} \right) + \text{constant}$$

is a self-concordant function with constant $\mathbf{c}_{H_{kj}} = \max_{\{i | V_{ij} > 0\}} \left\{ \frac{1}{\sqrt{V_{ij}}} \right\}$.

3 Existing Algorithms

In this section, we briefly review the most efficient algorithms for KL NMF.

3.1 Multiplicative Updates

Let us consider the linear regression problem: Given $v \in \mathbb{R}_+^m$ and $W \in \mathbb{R}_+^{m \times r}$,

$$\text{minimize}_{h \in \mathbb{R}_+^r} \; D_{\text{KL}}(v, Wh).$$

For W fixed in KL NMF, this is the subproblem to be solved for each column of H. The multiplicative updates (MU) for solving this problem are given by

$$h \; \leftarrow \; h \circ \left(\frac{W^\top \frac{[v]}{[Wh]}}{W^\top e e^\top} \right),$$

where $\circ$ and $\frac{[\cdot]}{[\cdot]}$ are the component-wise product and division between two matrices, respectively. They were derived by Richardson [42] and Lucy [38], who used them for image restoration, and rectification and deconvolution in statistical astronomy, respectively. This algorithm was later referred to as the Richardson–Lucy algorithm, and is guaranteed to decrease the KL NMF objective function.

In the context of NMF, Lee and Seung [31] derived these updates again, in a matrix form, to update W and H alternatively in KL NMF. The MU can be easily generalized to the perturbed KL NMF problem (2) [45], and are given by

$$H \leftarrow \max\left(\varepsilon, H \circ \frac{\left[W^\top \frac{[V]}{[WH]}\right]}{[W^\top ee^\top]}\right), \ W \leftarrow \max\left(\varepsilon, W \circ \frac{\left[\frac{[V]}{[WH]} H\right]}{[ee^\top H]}\right). \tag{8}$$

Note that, for $\varepsilon = 0$, the $\max(\varepsilon, \cdot)$ can be removed since the entries of V, W and H are nonnegative, which corresponds to the MU used by Lee and Seung.

The MU can be derived using the majorization–minimization (MM) framework, which is a two-step approach:

1. majorization: find a majorizer, that is, a function that is equal to the objective function at the current iterate while being larger everywhere else on the feasible domain, and
2. minimization: minimize the majorizer to obtain the next iterate.

We refer the interested reader to [13] for all the details. Moreover, the MU can be shown to belong to a specific MM framework, namely, the block successive upper-bound minimization (BSUM) framework [26,41]. For completeness, we describe the BSUM framework and its convergence guarantees in Appendix 1. This allows to provide convergence guarantees for the MU. Let us anaylze two cases separately.

3.1.1 Case $\varepsilon = 0$

In this case, the MU are not well-defined if $(WH)_{ij} = 0$ for some (i, j). Furthermore, the MU would encounter a zero locking phenomenon, that is, the MU cannot modify an entry of W or H when it is equal to 0. This phenomenon can be fixed by choosing an initial pair with strictly positive entries. Moreover, the objective function is not directionally differentiable when $(WH)_{ij} = 0$ and $V_{ij} > 0$ for some i, j. Hence, the convergence to stationary points obtained in [41, Theorem 2] for BSUM does not apply. In fact, [21] provided a numerical evidence that the generated sequence may not converge to a KKT point, and, [8, Section 6.2] gave an example that MU may converge to a non KKT point. Although the convergence of the generated sequence by MU is not guaranteed, it is worth noting that MU in this case possesses an interesting scale-invariant property.

Proposition 7 *Let $\varepsilon = 0$ and denote H^+ (resp. W^+) the update of H (resp. W) after applying one MU (8) of H (resp. of W) on (W, H). Suppose the MU for H (resp. W) is well-defined, that is, $(WH)_{ij} > 0$ for all i, j and W has no zero column (resp. H has no zero row). Then the MU of H preserve the column sum of V, that is, $e^\top V = e^\top (WH^+)$, while the MU of W preserve the row sum of V, that is, $Ve = (W^+H)e$.*

Proof We prove the result for H, the result for W can be obtained by symmetry. We have for $j \in [n]$ that

$$\begin{aligned}
\sum_{i=1}^m (WH^+)_{ij} &= \sum_{i=1}^m \sum_{k=1}^r W_{ik} H_{kj}^+ = \sum_{i=1}^m \sum_{k=1}^r W_{ik} H_{kj} \frac{\sum_{l=1}^m W_{lk} V_{lj}/(WH)_{lj}}{\sum_{l=1}^m W_{lk}} \\
&= \sum_{k=1}^r \sum_{i=1}^m W_{ik} H_{kj} \frac{\sum_{l=1}^m W_{lk} V_{lj}/(WH)_{lj}}{\sum_{l=1}^m W_{lk}} = \sum_{k=1}^r H_{kj} \sum_{l=1}^m \frac{W_{lk} V_{lj}}{(WH)_{lj}} \\
&= \sum_{l=1}^m \sum_{k=1}^r V_{lj} \frac{W_{lk} H_{kj}}{(WH)_{lj}} = \sum_{l=1}^m V_{lj}.
\end{aligned}$$

$\square$

Proposition 7 shows that any iterate of the MU, except the first one, is scaled (Definition 1). Hence, sometimes in the literature, the MU are used to scale the pair (W, H) within KL NMF algorithms.

3.1.2 Case $\varepsilon > 0$

For this case, [41, Theorem 2] implies that every limit point of the generated sequence of MU for solving Problem (2) with $\varepsilon > 0$ is a stationary point of Problem (2) (which is also a KKT point, see Sect. 2.2). It is worth noting that the sub-sequential convergence of MU in this case can also be proved by using Zangwill's convergence theory as in [45].

3.2 ADMM

The alternating direction method of multiplier (ADMM) is a standard technique to tackle low-rank matrix approximation problems [29], and it was used to solve KL NMF (1) in [44]. The first step is to reformulate (1) as

$$\min_{W, H, W_+, H_+} D(V|\hat{V})$$
$$\text{such that } \hat{V} = WH, W = W_+, H = H_+, W_+ \geq 0, H_+ \geq 0, \tag{9}$$

where $\hat{V}$, W_+ and H_+ are auxiliary variables. ADMM alternately minimizes the augmented Lagrangian of (9) over the variables $(\hat{V}, W, H, W_+, H_+)$, and updates the dual variables at each iteration. We refer the reader to [44] for more details. It is important noting that the objective is not monotonically decreasing under the updates of ADMM. Also, convergence to stationary points is not guaranteed. In fact, we will see in the numerical experiments that ADMM does not converge in some cases.

3.3 Primal-Dual Approach

In [50], Yanez and Bach proposed a first-order primal-dual (PD) algorithm for KL NMF (1). PD employs the primal-dual method proposed in [7] to tackle the convex subproblems in the columns of H and rows of W; see (7). To improve the performance of the primal-dual method, PD uses an automatic heuristic selection for the step sizes; see [50] for the details. PD is a heuristic algorithm, and it does not guarantee the monotonicity of the objective.

3.4 A Cyclic Coordinate Descent Method

In [28], Hsieh and Dhillon proposes to solve KL NMF (1) using a cyclic coordinate descent (CCD) method. CCD alternately updates the scalars W_{ik} or H_{kj}. The subproblems in one variable are solved by a Newton method, in which a full Newton step is used, without line search. Again, there is no convergence guarantee for this algorithm, although the Newton method used to approximately solve the subproblems in one variable may have some convergence guarantee, if properly tuned. In fact, we realized that there is a gap in the convergence proof of the Newton method without line search to the global minimum of the scalar subproblems [28, Theorem 1]: Equation (29) is not correctly computed which makes the proof incorrect. This observation motivated us to introduce a new scalar Newton-type algorithm in

Sect. 4.2. Note that a similar algorithm was independently developed in [36], along with an R package. Another R package is available that implements both CCD and MU [6].

3.5 Two Other Algorithmic Approaches

In [34], Li, Lebanon and Park proposed a Taylor series expansion to express Bregman divergences in term of Euclidean distances, leading to a heuristic scalar coordinate descent algorithm for solving NMF problems with Bregman divergences. The algorithm using this approach for solving KL NMF underperforms the CCD method from [28], and hence we will not compare it in the numerical experiment section.

More recently, Kim, Kim and Klabjan [30] proposed a scale invariant power iteration (SCI-PI) algorithm to solve a class of scale invariant problems, and apply it to KL NMF (1). To establish convergence results for SCI-PI (see [30, Theorem 9, 11]), the objective function of the scale invariant problem needs to be twice continuously differentiable on an open set containing the set $\{x : \|x\| = 1\}$. However, the objective function of the corresponding sub-problem (see [30, Lemma 14]) when applying SCI-PI to KL NMF violates this condition. Therefore, the theory of SCI-PI in [30] does not apply to KL NMF.

4 New Members of the Algorithm Collection

In this section, we present two new algorithms for KL NMF, a block mirror descent method in Sect. 4.1, and a new scalar Newton-type algorithm in Sect. 4.2.

4.1 Block Mirror Descent Method

Let us first give details on the analysis of the block mirror descent (BMD) method (Sect. 4.1.1), and then apply the result to solve the perturbed KL NMF problem (2) (Sect. 4.1.2).

4.1.1 BMD Method

The standard gradient descent (GD) scheme for solving the smooth optimization problem $\min_{x \in Q} g(x)$, with g being L-smooth, uses the following classical update

$$x^{k+1} = \arg\min_{x \in Q} \left\{ g(x^k) + \langle \nabla g(x^k), x\text{-}x^k \rangle + \frac{L}{2} \|x\text{-}x^k\|_2^2 \right\}. \tag{10}$$

When $g(\cdot)$ is not L-smooth but L-relative smooth to $\kappa(\cdot)$ (see Definition 2), the GD scheme can be generalized by replacing the Euclidean norm in (10) by the Bregman divergence $\mathcal{B}$, leading to the following mirror descent step

$$x^{k+1} = \arg\min_{x \in Q} \{ g(x^k) + \langle \nabla g(x^k), x - x^k \rangle + L \mathcal{B}_\kappa(x, x^k) \}. \tag{11}$$

Let us now present BMD, that uses the above generalized GD scheme, but updating the variable block by block. For that, let us consider a problem of the form

$$\min_{x \in \mathcal{X}} f(x), \tag{12}$$

where f is continuously differentiable on $\mathcal{X}$, $x := (x_1, x_2, \ldots, x_s)$ and $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2 \times \cdots \times \mathcal{X}_s$ with $\mathcal{X}_i$ $(i = 1, 2, \ldots, s)$ being closed convex sets. Let us also make the following assumption.

Assumption 1 (i) For all $\bar{x} \in \mathcal{X}$, the function $x_i \mapsto f(\bar{x}_1, \ldots, \bar{x}_{i-1}, x_i, \bar{x}_{i+1}, \ldots, \bar{x}_s)$ is convex and relative smooth to $\kappa_i(x_i)$ with constant $L_i(\bar{x}_1, \ldots, \bar{x}_{i-1}, \bar{x}_{i+1}, \ldots, \bar{x}_s)$.

(ii) There exist positive constants $\underline{L}_i$ and $\overline{L}_i$ such that

$$\underline{L}_i \leq L_i(\bar{x}_1, \ldots, \bar{x}_{i-1}, \bar{x}_{i+1}, \ldots, \bar{x}_s) \leq \overline{L}_i.$$

BMD for solving Problem (12) is described in Algorithm 1.

Algorithm 1 Block Mirror Descent for solving (12)

1: Choose an initial point x^0 such that $x_i^0 \in \text{int dom} \mathcal{X}_i$ for $i = 1, \ldots, s$.
2: **for** $k = 1, \ldots$ **do**
3:    **for** $i = 1, \ldots, s$ **do**
4:       Let $f_i^k(x_i) = f\big(x_1^{k+1}, \ldots, x_{i-1}^{k+1}, x_i, x_{i+1}^k, \ldots, x_s^k\big)$.
5:       Update

$$x_i^{k+1} = \arg \min_{x_i \in \mathcal{X}_i} \big\{ f_i^k(x_i^k) + \langle \nabla f_i^k(x_i^k), x_i - x_i^k \rangle + L_i^k \mathcal{B}_{\kappa_i}(x_i, x_i^k) \big\}, \qquad (13)$$

      where $L_i^k = L_i(x_1^{k+1}, \ldots, x_{i-1}^{k+1}, x_{i+1}^k, \ldots, x_s^k)$.
6:    **end for**
7: **end for**

BMD described in Algorithm 1 is structurally identical to the block Bregman proximal gradient (BBPG) method presented in [46] with cyclic update. However, BMD has an improvement over BBPG: it uses the step size $1/L_i^k$ which is larger than ρ/L_i^k used in BBPG, as $\rho \in (0, 1)$. Our convergence analysis of Algorithm 1 (Theorem 1 below, see its proof in Appendix 2) is an extension of the primal gradient scheme (which is BMD for $s = 1$) analysed in [37]. It is worth noting that our result can be extended to composite optimization with essentially cyclic regime by using the technique in [37, Section A.2] and [46, Section 2]. In this paper, we only present the result of BMD with cyclic regime for (12) to simplify the presentation.

Theorem 1 *Suppose Assumption 1 is satisfied. Let $\{x^k\}$ be the sequence generated by Algorithm 1. Let also $\Phi(x) = f(x) + \sum_i I_{\mathcal{X}_i}(x_i)$, where $I_{\mathcal{X}_i}$ is the indicator function of $\mathcal{X}_i$. We have*

(i) $\Phi(x^k)$ *is non-increasing;*

(ii) *Suppose $\mathcal{X}_i \subset \text{int dom}\, \kappa_i$. If $\{x^k\}$ is bounded and $\kappa_i(x_i)$ is strongly convex on bounded subsets of $\mathcal{X}_i$ that contain $\{x_i^k\}$, then every limit point of $\{x^k\}$ is a critical point of Φ;*

(iii) *If together with the conditions in (ii) we assume that $\nabla \kappa_i$ and $\nabla_i f$ are Lipschitz continuous on bounded subsets of $\mathcal{X}_i$ that contain $\{x_i^k\}$, then the whole sequence $\{x^k\}$ converges to a critical point of Φ.*

Let us now apply this new algorithm and convergence result to KL NMF.

4.1.2 BMD for KL NMF

Problem (2) has the form of Problem (12). Proposition 4 allows us to apply BMD to solve (2), where the blocks of variables are the columns of H and the rows of W. Using the notation

of Proposition 4, we have $\nabla_h D(v|Wh) = \sum_{i=1}^m \left(1 - \frac{v_i}{(Wh)_i}\right)W_{i,:}$, where $W_{i,:}$ is the i-th row of W, and $\mathcal{B}_\kappa(h, h^k) = \sum_{j=1}^r \frac{h_j}{(h^k)_j} - \log \frac{h_j}{(h^k)_j} - 1$. The following proposition provides the closed-form solution for the mirror descent step, see its proof in [3, Section 5.2].

Proposition 8 *Using the notation of Proposition 4, the problem*

$$\min_{h \geq \varepsilon} \left\{ D(v|Wh^k) + \langle \nabla_h D(v|Wh)[h^k], h - h^k \rangle + L\mathcal{B}_\kappa(h, h^k) \right\} \qquad (14)$$

has the following unique closed-form solution h^{k+1}: for $l \in [r]$,

$$(h^{k+1})_l = \max \left\{ \frac{(h^k)_l}{1 + \frac{1}{L}(h^k)_l \left(\sum\limits_{i=1}^m W_{il} - \sum\limits_{i=1}^m \frac{v_i W_{il}}{(Wh^k)_i} \right)}, \varepsilon \right\}. \qquad (15)$$

Recall that, $h \mapsto D(v|Wh)$ is $\|v\|_1$-relative smooth to $\kappa(h) = -\sum_{j=1}^r \log h_j$. Together with Proposition 8 which provides a closed-form solution of the update (13), we can therefore easily apply BMD to KL NMF. Assumption 1 is satisfied, and Theorem 1 (i) implies that $\Phi(x^k)$ is non-increasing. Moreover, using a similar method as in the proof Proposition 1 and noting that $\Phi(x^k) \leq \Phi(x^0)$, we can prove that BMD for KL NMF (2) with $\varepsilon > 0$ generates a bounded sequence. Together with the assumption $x \geq \varepsilon$, we see that all conditions of Theorem 1 (iii) are satisfied. Hence, we obtain the following convergence guarantee for BMD applied to (2).

Theorem 2 *Algorithm 1 applied on the KL NMF problem (2) monotonically decreases the objective function for any $\varepsilon \geq 0$. Moreover, for $\varepsilon > 0$, the generated sequence by BMD is bounded and globally converges to a stationary point of (2).*

It is important noting that $\nabla\kappa(h)$ is not Lipschitz continuous on bounded sets containing 0. We hence cannot apply Theorem 1 (iii) to the KL NMF problem (1).

4.2 A Scalar Newton-Type Algorithm

In this section, we propose a new scalar Newton-type (SN) algorithm that makes use of the self-concordant property of the objective function (see Sect. 2.4) to guarantee the non-increasingness for the objective sequence, that consequently guarantees the convergence of the objective sequence since it is also bounded from below. The motivation to propose this new method comes from our observation that CCD does not come with any convergence guarantee, as explained in Sect. 3.4, although it performs well in many cases.

Let us first have a brief review on Newton methods. Unconstrained minimization problem of a self-concordant function can be efficiently solved by Newton methods, see [40, Section 5.2]. Tran-Dinh et al. [47] brought the spirit of Newton methods for unconstrained optimization to composite optimization problems of the form

$$\min_{x \in \mathbb{R}^n} \Psi(x) := \psi(x) + \phi(x), \qquad (16)$$

where $\psi(x)$ is a standard self-concordant function and $\phi(x)$ is a proper, closed, convex but possibly non-smooth function. In particular, the authors propose a proximal Newton method (PNM) with the following update

$$x^{k+1} = x^k + \alpha_k d^k, \qquad (17)$$

where $\alpha_k \in (0, 1]$ is a step size, $d^k = s^k - x^k$ and

$$s^k = \arg\min_x \left\{ \psi(x^k) + \langle \nabla \psi(x^k), x - x^k \rangle + \frac{1}{2}(x - x^k)^\top \nabla^2 \psi(x^k)(x - x^k) + \phi(x) \right\}.$$

Denoting $\lambda_k = \|d^k\|_{x^k} = \left((d^k)^\top \nabla^2 \psi(x^k) d^k \right)^{1/2}$, it follows from [47, Theorem 6] that when the stepsize $\alpha_k = (1 + \lambda_k)^{-1}$ is used, then the PNM generates the sequence x^k satisfying $\Psi(x^{k+1}) \le \Psi(x^k) - \omega(\lambda_k)$, where $\omega(t) = t - \ln(1 + t)$.

Recall that the KL objective function with respect to a scalar component of W or H is a self-concordant function, see Proposition 6. We can hence make use of the update in (17) to propose the SN method; see Algorithm 2.

Algorithm 2 SN for solving Problem (1)

1: Choose initial points $W > 0, H > 0$.
2: Compute the self-concordant constants $\mathbf{c}_{W_{ik}}$ and $\mathbf{c}_{H_{kj}}$ of the function $W_{ik} \mapsto f$ and $H_{kj} \mapsto f$, see Proposition 6.
3: **repeat**
4:     Alternately update each scalar component of W and H. We update W_{ik} several times (similarly for H_{kj}) as follows.
5:     Calculate

$$f'_{W_{ik}} = \sum_{l=1}^n H_{kl} - \sum_{l=1}^n V_{il} \frac{H_{kl}}{(WH)_{il}}, f''_{W_{ik}} = \sum_{l=1}^n V_{il} \frac{H_{kl}^2}{(WH)_{il}^2},$$

and let $s = \max\left\{ W_{ik} - \frac{f'_{W_{ik}}}{f''_{W_{ik}}}, 0 \right\}$, $d = s - W_{ik}$, and $\lambda = \mathbf{c}_{W_{ik}} \sqrt{f''_{W_{ik}}}|d|$.
6:     **if** $f'_{W_{ik}} \le 0$ or† $\lambda \le 0.683802$ **then**
7:         update W_{ik} by a full proximal Newton step $W_{ik} \leftarrow s$,
8:     **else**
9:         update W_{ik} by $W_{ik} \leftarrow W_{ik} + \frac{1}{1+\lambda}d$.
10:    **end if**
11: **until** some criteria is satisfied
$\dagger$ Such λ guarantees that $\lambda^2 + \lambda + \log(1 - \lambda) > 0$ implying that the objective is non-increasing under a full Newton step; see the proof in Appendix 3.

The following proposition proves that SN monotonically decreases the objective function. The proof is provided in Appendix 3.

Proposition 9 *The objective function for the perturbed KL-NMF problem* (1) *is non-increasing under the updates of Algorithm* 2.

4.3 A Hybrid SN-MU Algorithm

We recall that MU possesses an important property, namely that (W, H) scaled after any MU update; see Proposition 7. On the other hand, we note that KKT points of Problem (1) are also scaled. However, SN does not possess this scale-invariant property. Hence we propose to combine SN with MU to result in a hybrid SN-MU algorithm. Specifically, we alternately run several SN steps before scaling the sequence by running one or several updates of MU. In Sect. 5, we will use 10 steps of SN followed by one step of MU for all numerical experiments. As we will see, this combination sometimes significantly improves the performance of SN.

Table 1 Properties of KL NMF algorithms presented in this paper, applied on an m-by-n matrix V to be approximated with a rank-r approximation WH. The parameter d is the number of inner iterations

Algorithms	Complexity (flops)	Convergence	Monotonicity	References
MU	$O(mnr)$	✓	✓	Sect. 3.1
ADMM	$O(mnr)$	✗	✗	Sect. 3.2
PD	$O(mnrd)$	✗	✗	Sect. 3.3
CCD	$O(mnrd)$	✗	✗	Sect. 3.4
BMD	$O(mnr)$	✓	✓	Sect. 4.1
SN	$O(mnrd)$	✗	✓	Sect. 4.2
SN-MU	$O(mnrd)$	✗	✓	Sect. 4.3

5 Experiments

In this section, we report comparisons of the KL NMF algorithms listed in Table 1.

The second column of Table 1 provides the complexity of one iteration to update all entries of (W, H). The parameter d in the second column of PD, CCD, SN and SN-MU is the number of inner iterations of one main iteration of updating (W, H). The third column indicates whether the corresponding algorithm has some convergence guarantee for its generated sequence. We note that, considering Problem (2) with $\varepsilon > 0$, MU guarantees a subsequential convergence while BMD guarantees a global convergence. The fourth column indicates if the sequence of the objective function values is non-decreasing.

5.1 Implementation

We have implemented MU and BMD in Matlab, SN in C++ and use its mex file to run it from Matlab, as for CCD provided by the authors[1] (for which we have fixed an issue on maintaining $(WH)_{ij}$, otherwise it sometimes run into numerical issues generating NaN objective function values because $(WH)_{ij}$ could take negative values). We use the Matlab code provided by the authors for ADMM[2] and PD[3]. We used the best possible programming language for each algorithm. For example, if CCD was implemented on Matlab, it would be extremely slow as it loops over each variable (and Matlab is very ineffective to handle loops). On the other side, the MU run faster on Matlab because the main computational cost resides in matrix–matrix multiplications for which Matlab is more effective than C++. All tests are preformed using Matlab R2018a on a laptop Intel CORE i7-8550U CPU @1.8GHz 16GB RAM. The code is available at https://github.com/LeThiKhanhHien/KLNMF. We choose the penalty parameter of ADMM to be equal to 1 in all of the experiments. In each run for a data set, we use the same random initialization and the same maximal running time for all algorithms.

We use the Matlab commands $W = rand(m, r)$ and $H = rand(r, n)$ to generate a random initial point; and to avoid initial points with a large value $D(V, WH)$, we then scale W and H by $W = \sqrt{\alpha}W$, $H = \sqrt{\alpha}H$, where $\alpha = \frac{\sum_{i,j} V_{ij}}{\sum_{i,j} (WH)_{ij}}$; see Definition 1 and Proposition 3. We define the relative error rel $D(V, WH)$ to be the objective $D(V, WH)$

1 http://www.cs.utexas.edu/~cjhsieh/nmf.

2 http://statweb.stanford.edu/~dlsun/admm.html.

3 https://github.com/felipeyanez/nmf.

divided by $\sum_{i,j} V_{ij} \log \frac{V_{ij}}{(\sum_j V_{ij})/n}$, and denote $E(t)$ the value of rel $D(V, WH) - e_{\min}$ at time t, where $e_{\min}$ is the smallest value among all rel $D(V, WH)$ produced by all algorithms and all initializations within the allotted time. Hence $E(t)$ goes to zero for the best run among all algorithms and initializations.

5.2 Experiments with Synthetic Data Sets

For each type of synthetic data sets, we generate 10 random matrices V, then for each random V, we generate 10 random initial points.

5.2.1 Low-Rank Synthetic Data Sets

We will consider several types of low-rank synthetic data sets depending on the parameter $\ell \in (0, 1]$ which is the density of the underlying factors, denoted W^* and H^*. We will use $\ell = 1, 0.9, 0.3$. More precisely, to generate a low-rank synthetic data set V, we use the Matlab commands $W^* = sprand(m, r, \ell)$ and $H^* = sprand(r, n, \ell)$, where ℓ is the density of non-zero elements (that is, $1 - \ell$ is the percentage of zero elements), and let $V = W^* H^*$. We will also either keep $V = W^* H^*$ as is, which is the noiseless case, or generate each entry of V following a Poisson distribution of parameters $W^* H^*$ as described in Sect. 1.1, which is a noisy case and is achieved with the Matlab command $V = poissrnd(W^* H^*)$.

The results of applying the different algorithms on such matrices are reported in Fig. 1 for 200-by-200 matrices with $r = 10$, and in Fig. 2 for 500-by-500 matrices with $r = 20$. We report the evolution of the median value of $E(t)$. Although this is not an ideal choice, comparing the performance in term of iterations would be worse since the cost of one iteration can be rather different for each algorithm; for example, CCD has inner iterations, which is not the case of MU.

We also report the average and standard deviation (std) of the relative errors over 200 runs for the 6 types of synthetic data sets (100 runs for each size 200×200 or 500×500) in Table 2, and provide a ranking over the total 1200 runs between the different algorithms in Table 3: the ith entry of the ranking vector indicates how many times the corresponding algorithm obtained the ith best solution (that is, with the ith lowest objective function value). This table allows to see which algorithms performs on average the best on these data sets.

For these low-rank synthetic data sets, let us discuss the behaviour of the various algorithms:

- ADMM is not stable, it diverges in many cases. Although the results are for ADMM with the penalty parameter $\varrho = 1$, we also tried other values for ϱ but the algorithm still did not converge for the other values we have tried. ADMM has the largest number of worst solutions (503 out of 1200). Note however that it also has a high number of best solutions (301 out of 1200), because it performs well in the simple scenario when the factors W and H are dense in the absence of noise. In summary, ADMM is unstable but, when it converges, it provides good solutions.
- CCD performs very well, among the best in most cases. When looking at Table 2, we observe that CCD has average results, having most of its solutions ranked third to fifth (out of 7). However, it has the second lowest average error right after SN-MU.
- SN monotonically decreases the objective function, as proved in Proposition 9. However, in some cases, it may converge rather slowly. Its ranking are well distributed, hence it performs close to the average.
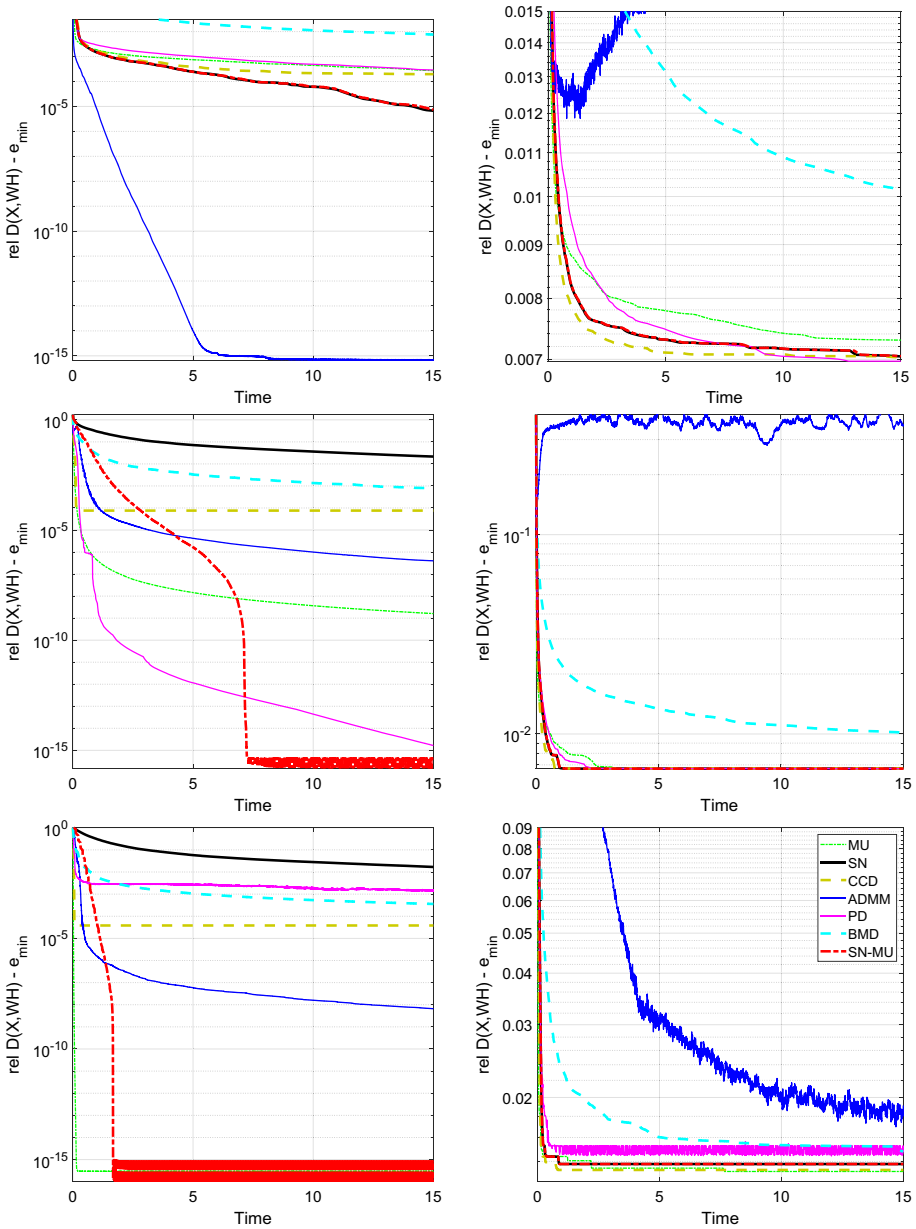
Fig. 1 Median value of the error measure $E(t)$ on 200×200 low-rank matrices for various KL NMF algorithms (see Table 1). The left column corresponds to noiseless input matrices, the right column to noisy matrices using the Poisson distribution. The top row corresponds to dense factors ($\ell = 1$), the middle row to slightly sparse factors ($\ell = 0.9$), and the bottom row to sparse factors ($\ell = 0.3$)
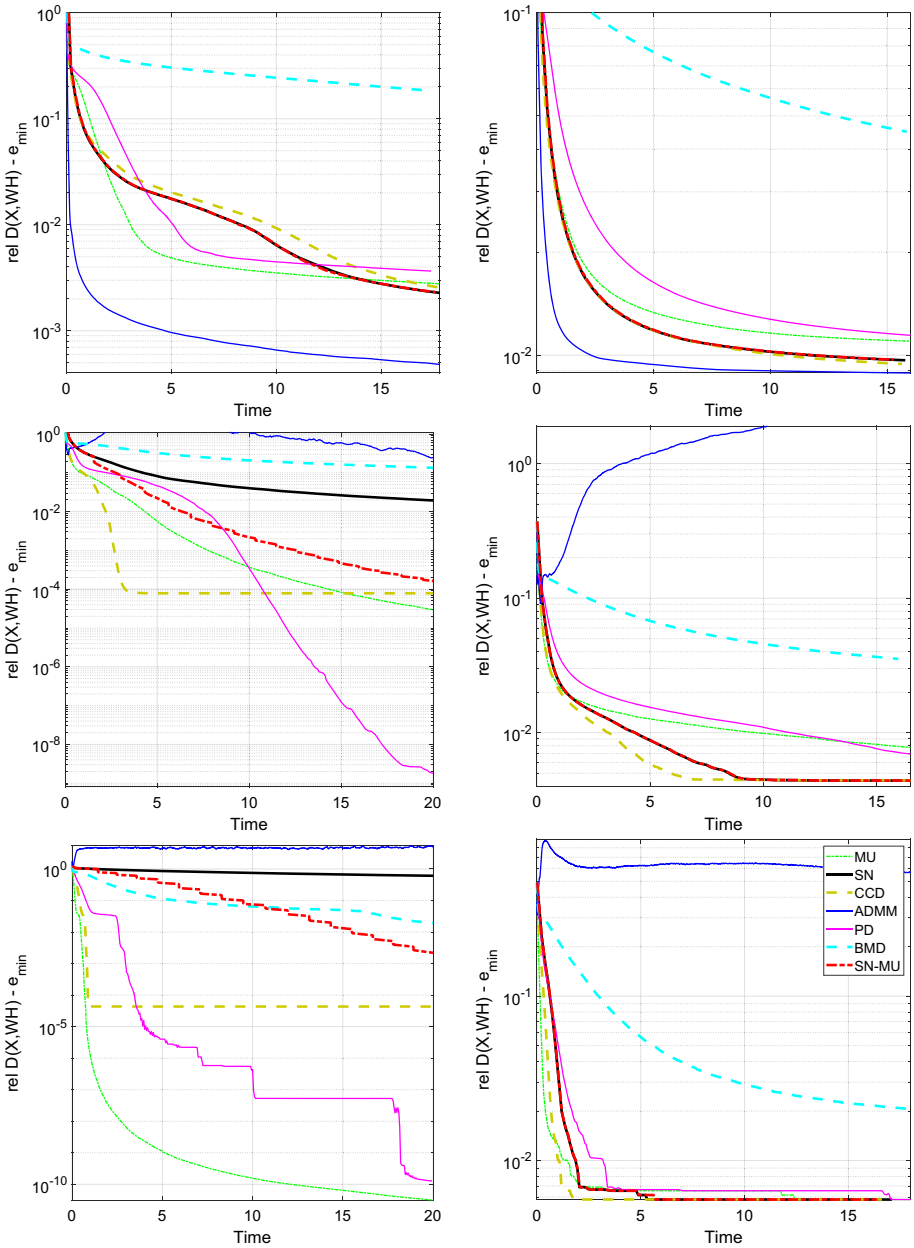
Fig. 2 Median value of the error measure $E(t)$ on 500×500 low-rank matrices for various KL NMF algorithms (see Table 1). The left column corresponds to noiseless input matrices, and the right column to noisy matrices using the Poisson distribution. The top row corresponds to dense factors ($\ell = 1$), the middle row to slightly sparse factors ($\ell = 0.9$), and the bottom row to sparse factors ($\ell = 0.3$)

Table 2 Average error, standard deviation over 200 different runs for each type of low-rank synthetic data sets (100 runs for each type with the size 200×200 or 500×500). The middle column correspond to noiseless input matrices, the right column to noisy matrices using the Poisson distribution. The first subtable corresponds to dense factors ($\ell = 1$), the second subtable slightly sparse factors ($\ell = 0.9$), and the third subtable to sparse factors ($\ell = 0.3$)

Algorithm	Mean $\pm$ std (noiseless matrices)	Mean $\pm$ std (noisy matrices)
ADMM	$\mathbf{2.257\,10^{-4} \pm 2.519\,10^{-4}}$	$8.030\,10^{-1} \pm 7.445\,10^{-3}$
CCD	$1.119\,10^{-3} \pm 9.145\,10^{-4}$	$\mathbf{8.025\,10^{-1} \pm 7.966\,10^{-3}}$
SN	$9.432\,10^{-4} \pm 9.106\,10^{-4}$	$8.026\,10^{-1} \pm 8.014\,10^{-3}$
MU	$1.423\,10^{-3} \pm 1.154\,10^{-3}$	$8.032\,10^{-1} \pm 8.380\,10^{-3}$
BMD	$7.962\,10^{-2} \pm 7.325\,10^{-2}$	$8.172\,10^{-1} \pm 1.890\,10^{-2}$
PD	$1.803\,10^{-3} \pm 1.551\,10^{-3}$	$8.033\,10^{-1} \pm 8.527\,10^{-3}$
SN-MU	$9.133\,10^{-4} \pm 8.780\,10^{-4}$	$8.026\,10^{-1} \pm 8.009\,10^{-3}$
ADMM	$7.107\,10^{-2} \pm 1.129\,10^{-1}$	$8.497\,10^{-1} \pm 1.149\,10^{-2}$
CCD	$7.807\,10^{-5} \pm 3.642\,10^{-6}$	$\mathbf{7.563\,10^{-1} \pm 1.671\,10^{-2}}$
SN	$2.013\,10^{-2} \pm 1.335\,10^{-2}$	$7.564\,10^{-1} \pm 1.661\,10^{-2}$
MU	$8.795\,10^{-6} \pm 1.088\,10^{-5}$	$7.573\,10^{-1} \pm 1.759\,10^{-2}$
BMD	$5.950\,10^{-2} \pm 5.917\,10^{-2}$	$7.703\,10^{-1} \pm 2.715\,10^{-2}$
PD	$\mathbf{6.156\,10^{-9} \pm 1.917\,10^{-8}}$	$7.567\,10^{-1} \pm 1.702\,10^{-2}$
SN-MU	$4.515\,10^{-5} \pm 6.725\,10^{-5}$	$7.564\,10^{-1} \pm 1.661\,10^{-2}$
ADMM	$5.493\,10^{-1} \pm 5.671\,10^{-1}$	$7.292\,10^{-1} \pm 1.869\,10^{-1}$
CCD	$1.340\,10^{-3} \pm 9.142\,10^{-3}$	$5.762\,10^{-1} \pm 3.462\,10^{-2}$
SN	$2.911\,10^{-1} \pm 2.790\,10^{-1}$	$\mathbf{5.761\,10^{-1} \pm 3.468\,10^{-2}}$
MU	$\mathbf{3.525\,10^{-4} \pm 4.985\,10^{-3}}$	$5.759\,10^{-1} \pm 3.502\,10^{-2}$
BMD	$1.592\,10^{-2} \pm 2.041\,10^{-2}$	$5.830\,10^{-1} \pm 3.924\,10^{-2}$
PD	$9.711\,10^{-4} \pm 6.808\,10^{-3}$	$5.769\,10^{-1} \pm 3.419\,10^{-2}$
SN-MU	$1.468\,10^{-3} \pm 6.274\,10^{-3}$	$\mathbf{5.761\,10^{-1} \pm 3.468\,10^{-2}}$

Bold indicates the best value obtained among the tested algorithms

Table 3 Ranking among 1200 different runs for low-rank synthetic data sets

Algorithm	Ranking
ADMM	(301, 6, 97, 121, 55, 117, 503)
CCD	(52, 131, 302, 344, 328, 32, 11)
SN	(148, 273, 249, 89, 109, 147, 185)
MU	(140, 243, 172, 198, 369, 71, 7)
BMD	(1, 0, 2, 26, 184, 522, 465)
PD	(222, 217, 161, 164, 122, 293, 21)
SN-MU	(**343**, 328, 214, 257, 32, 18, 8)

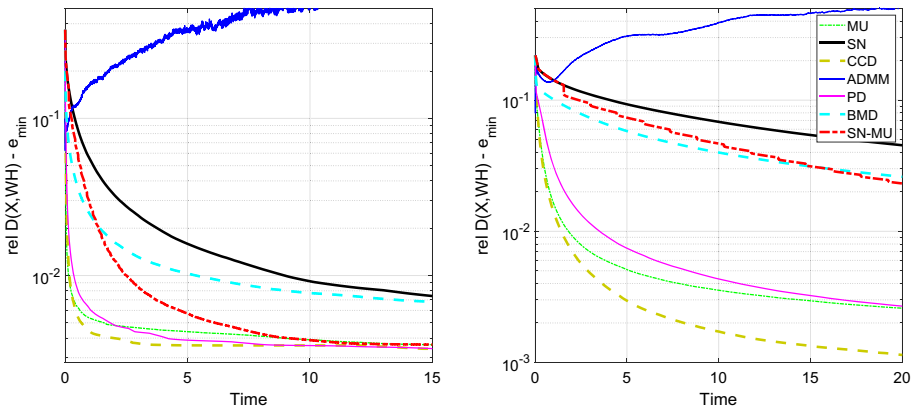Bold indicates the best value obtained among the tested algorithms

Fig. 3 Median value of $E(t)$ on Full-rank synthetic data sets: left—200×200, right—500×500

- SN-MU improves SN and performs well, in all cases among the best algorithms. In fact, it obtained the lowest objective function values among all algorithm, 343 out of the 1200 experiments. Also, it generates only 8 out of 1200 solutions as the worst solutions. Also, it has the lowest relative error on average.
- MU performs well, on average better than the other algorithms (it only provides 7 worst solutions, out of 1200 tests). It has a low relative error on average, ranked third, right behind SN-MU and CDD.
- BMD converges very slowly (it only provides a solution among the third best ones in 3 cases out of 1200). Although it is the only algorithm with global convergence guarantee (Proposition 2), this comes at the expense of slow convergence.
- PD performs well, although it provides in many cases (293 out of 1200) the second worst solutions.

In summary, SN-MU performs on average the best, followed by the MU, CCD, PD, and SN. ADMM does not always converge but can produce good solutions. BMD has strong convergence guarantees but converges very slowly. However, there is no clear winner, and, depending on the types of data sets, some algorithms might perform better than others.

5.2.2 Full-Rank Synthetic Data Sets

We generate a full-rank synthetic data set V by the Matlab command $V = rand(m, n)$. Results for full-rank synthetic 200×200 (with $r = 10$) and 500×500 (with $r = 20$) data sets are reported in Fig. 3. We also report the average, the standard deviation of the relative errors and the ranking vector over 200 runs (100 runs for each size) in Table 4.

We observe that the behavior can be quite different than in the low-rank cases. In particular,

- CCD now clearly performs best in term of convergence speed and average relative error.
- The MU, CCD and SN-MU performs well, while BMD and SN perform relatively poorly (they never produce the best solution).
- ADMM never converges, and produces the worst solution in all cases.

Table 4 Average error, standard deviation and ranking among 200 different runs for full-rank synthetic data sets

Algorithm	Mean ± std	Ranking
ADMM	$9.369 \, 10^{-1} \pm 2.774 \, 10^{-2}$	(0, 0, 0, 0, 0, 0, 200)
CCD	$\mathbf{8.709 \, 10^{-1}} \pm \mathbf{1.418 \, 10^{-2}}$	(**138**, 27, 23, 12, 0, 0, 0)
SN	$8.923 \, 10^{-1} \pm 3.218 \, 10^{-2}$	(0, 0, 0, 0, 47, 153, 0)
MU	$8.717 \, 10^{-1} \pm 1.480 \, 10^{-2}$	(12, 62, 91, 35, 0, 0, 0)
BMD	$8.835 \, 10^{-1} \pm 2.370 \, 10^{-2}$	(0, 0, 0, 0, 153, 47, 0)
PD	$8.716 \, 10^{-1} \pm 1.486 \, 10^{-2}$	(37, 93, 58, 12, 0, 0, 0)
SN-MU	$8.793 \, 10^{-1} \pm 2.241 \, 10^{-2}$	(13, 18, 28, 141, 0, 0, 0)

Bold indicates the best value obtained among the tested algorithms

Table 5 Real data sets

Data set	Size	Run time (seconds)
Audio [20]		
Mary	129×586	25
Prelude JSB	$129 \times 2\,582$	45
ShanHur sunrise	$129 \times 4\,102$	75
Voice cell	$129 \times 2\,181$	45
Images [27,31]		
Cbclim	$361 \times 2\,429$	50
ORLfaces	$10\,304 \times 400$	95
Documents [51]		
Classic	$7\,094 \times 41\,681$	500
Hitech	$2\,301 \times 10\,080$	500
Reviews	$4\,069 \times 18\,483$	500
Sports	$8\,580 \times 14\,870$	500
Ohscal	$11\,162 \times 11\,465$	500
La1	$3\,204 \times 31\,472$	500

5.3 Experiments with Real Data Sets

We report in this section experiments on various widely used real data sets that are summarized in Table 5. We use $r = 10$ for all real data sets.

5.3.1 Audio Data Sets

For each audio data set, we generate 30 random initial points. We report the evolution of the median of $E(t)$ in Fig. 4, and report the average, the standard deviation of the relative errors and the ranking vector over 120 runs (30 runs for each audio data set) in Table 6.

As for full-rank synthethic data sets, ADMM diverges while SN and BMD converges slowly. However, for these data sets, MU outperforms the other algorithms, followed by CCD, SN-MU, and PD.
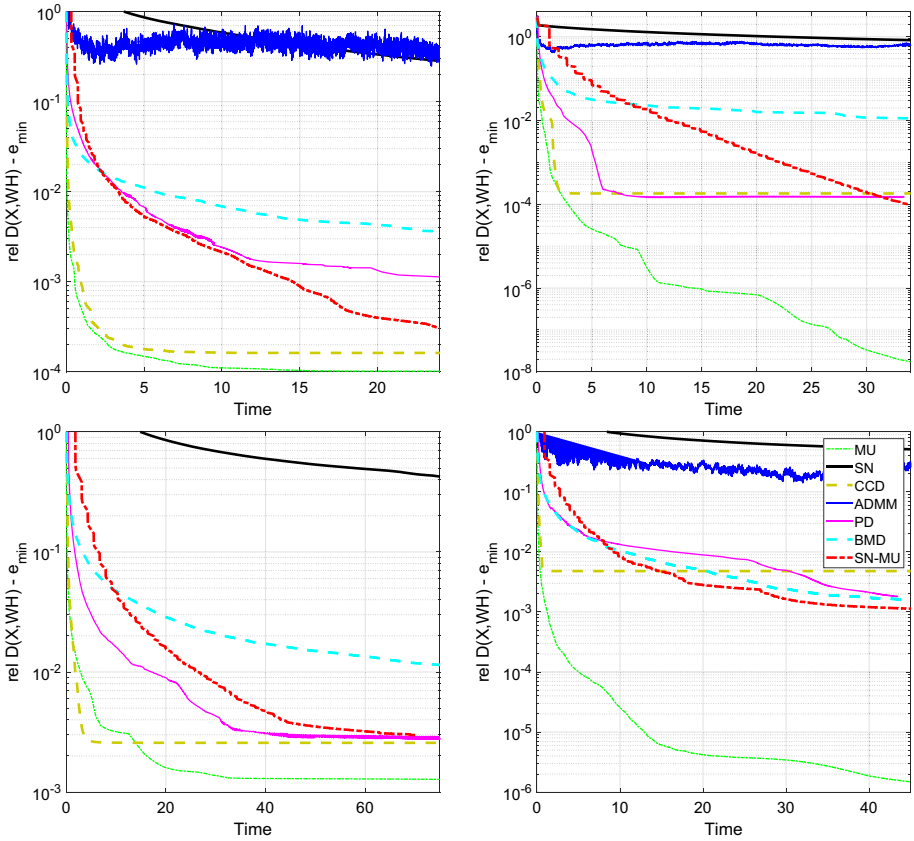
Fig. 4 Median value of $E(t)$ on audio data sets: top left—mary, top right—prelude JSB, bottom left—ShanHur sunrise, bottom right—voice cell

Table 6 Average error, standard deviation and ranking among 120 different runs for audio data sets

Algorithm	Mean ± std	Ranking
ADMM	$2.533\,10^{-1} \pm 1.916\,10^{-1}$	$(0, 0, 0, 0, 1, 110, 9)$
CCD	$7.588\,10^{-2} \pm 4.853\,10^{-2}$	$(5, 42, 33, 23, 17, 0, 0)$
SN	$5.213\,10^{-1} \pm 2.002\,10^{-1}$	$(0, 0, 0, 0, 0, 9, 111)$
MU	$\mathbf{7.468\,10^{-2} \pm 4.849\,10^{-2}}$	$(\mathbf{89}, 7, 13, 7, 4, 0, 0)$
BMD	$7.898\,10^{-2} \pm 4.946\,10^{-2}$	$(0, 12, 21, 25, 61, 1, 0)$
PD	$7.642\,10^{-2} \pm 4.841\,10^{-2}$	$(9, 20, 23, 46, 22, 0, 0)$
SN-MU	$7.498\,10^{-2} \pm 4.815\,10^{-2}$	$(17, 39, 30, 19, 15, 0, 0)$

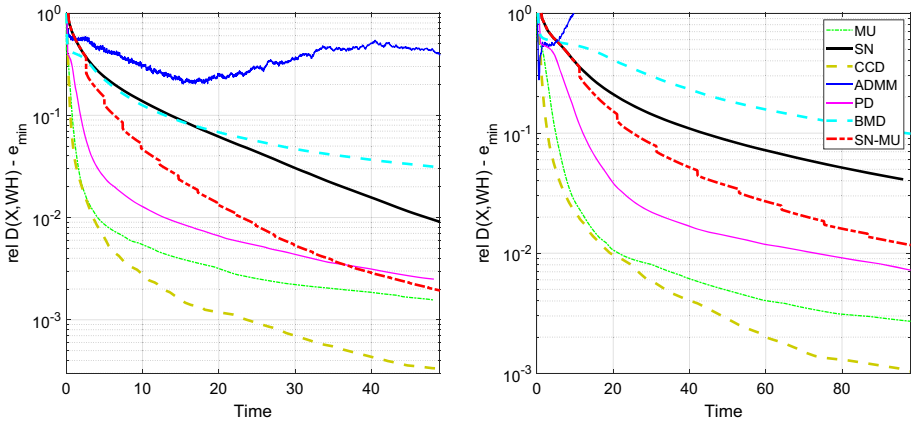Bold indicates the best value obtained among the tested algorithms

Fig. 5 Median value of $E(t)$ on image data sets: left—cbclim, right—ORLfaces

Table 7 Average error, standard deviation and ranking among 60 different runs for the 2 image data sets

Algorithm	Mean ± std	Ranking
ADMM	$5.492 \, 10^{-1} \pm 2.167 \, 10^{-1}$	(0, 0, 0, 0, 0, 0, 60)
CCD	$\mathbf{3.339 \, 10^{-1} \pm 1.529 \, 10^{-1}}$	(**51**, 8, 0, 1, 0, 0, 0)
SN	$3.569 \, 10^{-1} \pm 1.679 \, 10^{-1}$	(0, 0, 0, 0, 60, 0, 0)
MU	$3.354 \, 10^{-1} \pm 1.533 \, 10^{-1}$	(8, 37, 10, 5, 0, 0, 0)
BMD	$3.977 \, 10^{-1} \pm 1.867 \, 10^{-1}$	(0, 0, 0, 0, 0, 60, 0)
PD	$3.379 \, 10^{-1} \pm 1.550 \, 10^{-1}$	(0, 4, 34, 22, 0, 0, 0)
SN-MU	$3.387 \, 10^{-1} \pm 1.565 \, 10^{-1}$	(1, 11, 16, 32, 0, 0, 0)

Bold indicates the best value obtained among the tested algorithms

5.3.2 Image Data Sets

As for audio data sets, we generate 30 random initial points. We report the result in Fig. 5 and Table 7.

As for full-rank and audio data sets, ADMM diverges, and SN and BMD converge slowly. CCD outperforms the other algorithms followed by MU, SN-MU and PD (in that order).

5.3.3 Document Data Sets

For each document data set, we generate 10 random initial points and record the final relative errors (the reason of using only 10 initializations is that these data sets are rather large, and the computational time is high–we used 500 seconds for each run as shown on Table 5). We report the average, the standard deviation of the final relative errors and the ranking vector over 60 runs (10 runs for each document data set) in Table 8.

We observe that, CCD performs best, followed by SN-MU, in terms of the average relative error.

Performance profiles. Fig. 6 reports the performance profiles for the experiments for synthetic (left) and real (right) data sets. It displays the performance of each algorithm as a function

Table 8 Average error, standard deviation and ranking among 60 different runs on the document data sets

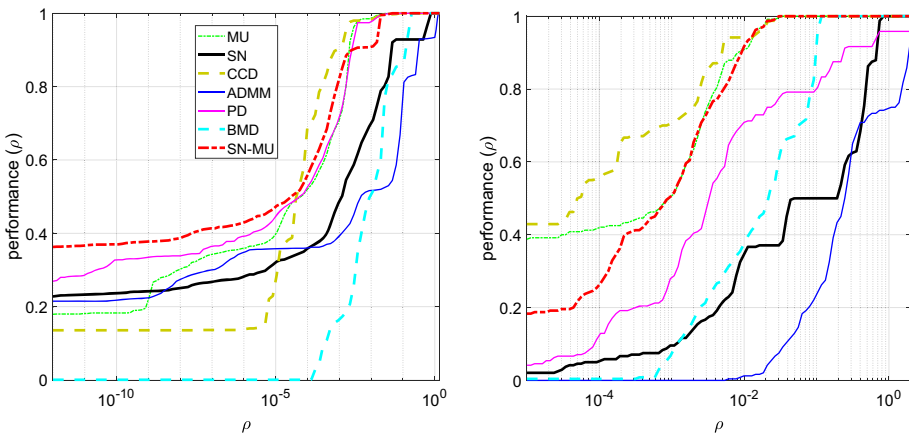| Algorithm	Mean $\pm$ std	Ranking
ADMM	$2.333\,10^0 \pm 3.276\,10^{-1}$	(0, 0, 0, 0, 0, 10, 50)
CCD	$\mathbf{5.366\,10^{-1} \pm 3.483\,10^{-2}}$	(**37**, 3, 16, 4, 0, 0, 0)
SN	$5.388\,10^{-1} \pm 3.443\,10^{-2}$	(3, 13, 30, 14, 0, 0, 0)
MU	$5.402\,10^{-1} \pm 3.394\,10^{-2}$	(4, 9, 7, 40, 0, 0, 0)
BMD	$6.057\,10^{-1} \pm 2.476\,10^{-2}$	(0, 0, 0, 0, 60, 0, 0)
PD	$1.783\,10^0 \pm 2.270\,10^0$	(0, 0, 0, 0, 0, 50, 10)
SN-MU	$5.370\,10^{-1} \pm 3.472\,10^{-2}$	(16, 35, 7, 2, 0, 0, 0)

Fig. 6 Performance profile for all experiments; see (18). On the left: synthetic data set, on the right: real data sets

of $\rho \geq 0$. For a given value of ρ, we define the performance of an algorithm as follows

$$\text{performance}(\rho) = \frac{\sharp\big\{\text{solution } (W, H) \mid \text{rel } D(V, WH) - \text{rel } D(V, W^*H^*) \leq \rho\big\}}{\sharp\text{runs}}, \quad (18)$$

where a solution (W, H) is the final pair obtained within the total allotted time by the algorithm, and (W^*, H^*) is the best solution obtained among all algorithms using the *same* initialization. Hence, for example, performance(0) aggregates the values of the rankings at the first position provided in Tables 3 and 4 for the synthetic data sets, and in Tables 6, 7 and 8 for the real data sets.

Performance profiles allow us to compare the algorithms meaningfully over different instances [12], that is, different matrices and initializations in our case. Looking at the curves for ρ equal zero, we observe the percentage of the time each algorithm was able to obtain the best solution. The right of the curve, as ρ increases, reports the robustness of an algorithm, that is, the percentage of times it was able to obtain a solution close to the best solutions found among all algorithms. In all cases, the higher the curve the better.

Figure 6 confirms our observations: CCD, MU and SN-MU perform the best. On synthetic data sets, SN-MU provides the best solutions in most cases (left part of the left figure), while on real data sets, MU and CCD perform better (left part of the right figure). In terms

of robustness, the three algorithms are comparable: their curves get closer together as ρ increases. Note that, as we have observed as well, PD performs on par with CCD, MU and SN-MU on synthetic data sets, while it is less effective for real data sets as it does not reach a performance of 1 even for $\rho = 1$.

5.4 Conclusions of the Numerical Experiments

Surprisingly, for KL NMF, the behaviour of the algorithms can be highly dependent on the input data. For example, CCD performs best for images and documents, while MU performs best for audio data sets. To the best of our knowledge, this has not been reported in the literature. As far as we know, most papers focus only on a few numerical examples, introducing an undesirable bias towards certain algorithms. It is interesting to note that for Fro NMF, such different behaviors depending on the input data has not been reported in the literature, despite numerous studies. The reason is most likely that KL NMF is a more difficult optimization problem, for which the subproblem in W and H, although convex, does not admit an L-Lipschitz gradient.

The main take-home message of our experiments is that CCD, SN-MU and MU appear to be the most reliable algorithms for KL NMF, performing among the best in most scenarios.

6 Conclusion

In this paper, we have presented important properties of KL NMF that are useful to analyze algorithms (Sect. 2). Then, we have reviewed existing algorithms, and proposed three new algorithms: a block mirror descent (BMD) method with global convergence guarantees, a scalar Newton-type (SN) algorithm which monotonically decreases the objective function, and an hybridization between SN and MU. Finally, in Sect. 5, we performed extensive numerical experiments on synthetic and real data sets. Although no KL NMF algorithms clearly outperforms the others, it appears that the CCD, MU and SN-MU provide the best results on average.

Declarations

Conflict of interest Not applicable.

Availability of data and material The data sets that support the findings of this study are freely available online, and available from the corresponding author, Nicolas Gillis, upon reasonable request.

Code availability The codes used to perform the experiments in this paper are available from https://github.com/LeThiKhanhHien/KLNMF.

A Technical Proofs

A.1 Proof of Proposition 1

Proposition 1: (A) Given $\varepsilon \geq 0$ and a nonnegative matrix V, Problem (2) attains its minimum, that is, it has at least one optimal solution.

(B) Let $\nu = \sum_{i=1}^{m} \sum_{j=1}^{n} V_{ij}$. Denote $D^*(V, \varepsilon)$ be the optimal value of Problem (2). We have $D^*(V, \varepsilon) \leq D^*(V, 0) + (\min\{n + mr, m + nr\}\sqrt{\nu} + mn\varepsilon)\varepsilon$.

Proof Problem (2) can be rewritten as follows

$$\min_{W,H} \quad D\big(V|(W + \varepsilon ee^\top)(H + \varepsilon ee^\top)\big)$$

such that $W_{ik} \geq 0$, $H_{kj} \geq 0$ for $i \in [m], k \in [r], j \in [n]$. (19)

We note that

$$(W + \varepsilon ee^\top)(H + \varepsilon ee^\top) = WH + \varepsilon(Wee^\top + ee^\top H) + \varepsilon^2 ee^\top. \quad (20)$$

Let us now prove the parts (A) and (B) of Proposition 1 separately.

(A) We note that

$$\frac{1}{\nu} D\big(V|(W + \varepsilon ee^\top)(H + \varepsilon ee^\top)\big) = D\Big(\frac{V}{\nu}\Big|\Big(\frac{W}{\sqrt{\nu}} + \frac{\varepsilon}{\sqrt{\nu}}ee^\top\Big)\Big(\frac{H}{\sqrt{\nu}} + \frac{\varepsilon}{\sqrt{\nu}}ee^\top\Big)\Big).$$

Then we have

$$D^*(V, \varepsilon) = \nu D^*(V/\nu, \varepsilon/\sqrt{\nu}). \quad (21)$$

Let us now consider Problem (19) with V such that $\sum_{i=1}^{m} \sum_{j=1}^{n} V_{ij} = 1$. In the following, we separate the proof into 2 cases: $\varepsilon = 0$, which corresponds to the original KL NMF Problem (1), and $\varepsilon > 0$, for which the objective of Problem (19) is finite at every pair of non-negative matrices (W, H).

Case 1 $\varepsilon = 0$. We use the methodology in the proof of [16, Proposition 2.1]. We first observe that $WH = \sum_{i=1}^{r} W_{:,i} H_{i,:}$, hence without loss of generality we can consider the case when H has no zero rows; otherwise we could then consider Problem (1) with smaller rank than r. Given a feasible solution (W, H) of (1), let $h_k = \sum_{j=1}^{n} H_{kj}$ and let $\text{Diag}(h)$ be the diagonal matrix with its diagonal being h. We then have $(\tilde{W}, \tilde{H})$, where $\tilde{W} = W\text{Diag}(h)$ and $\tilde{H} = (\text{Diag}(h))^{-1}H$, is also a feasible solution of (1) and it preserves the objective value since $\tilde{W}\tilde{H} = WH$. We observe that $\sum_{j=1}^{n} \tilde{H}_{kj} = 1$. Hence, we can restrict our search for the optimal solution to the set

$$\mathcal{A}_1 = \{(W, H) : W, H \geq 0, He = e\}.$$

We note that $\lim_{x \to 0}(x - V_{ij} \log x) = +\infty$ for a given $V_{ij} > 0$, and $x \mapsto (x - V_{ij} \log x)$ is a decreasing function when $x < V_{ij}$. Therefore, there exists a positive constant δ such that $(WH)_{ij} \geq \delta$ for all i, j with $V_{ij} > 0$; otherwise, if for all $\delta > 0$ there exists (i, j) with $V_{ij} > 0$ that $(WH)_{ij} \leq \delta$ then we can choose arbitrary small $\delta < \min\{V_{ij} : V_{ij} > 0\}$ which makes $(WH)_{ij} - V_{ij} \log(WH)_{ij}$ arbitrary large. Therefore, we can further restrict the constraint set of (1) to the set

$$\mathcal{A}_2 = \big\{(W, H) : (W, H) \in \mathcal{A}_1, (WH)_{ij} \geq \delta \forall (i, j) \in \{(i, j) : V_{ij} > 0\}\big\}.$$

It follows from [32, Theorem 2] (see also Sect. 3.1) that, given a feasible solution (W, H), a multiplicative update step $W_{ik}^+ = W_{ik}\big(\sum_{l=1, V_{il}>0}^{n} \frac{H_{kl}V_{il}}{(WH)_{il}}\big)\big/\big(\sum_{l=1}^{n} H_{kl}\big)$, for $i \in [m], k \in$

$[r]$, supposed to be well-defined, decreases the objective function. Hence, given $(W, H) \in \mathcal{A}_2$, we observe that

$$\sum_{k=1}^{m} W_{ik}^{+} = \sum_{k=1}^{m} W_{ik} \sum_{l=1}^{n} \frac{H_{kl} V_{il}}{(WH)_{il}} = \sum_{l=1}^{n} \sum_{k=1}^{m} \frac{W_{ik} H_{kl} V_{il}}{(WH)_{il}} = \sum_{l=1}^{n} V_{il}.$$

Therefore, we can further restrict our search for the optimal solutions of (1) to

$$\mathcal{A}_3 = \{(W, H) : (W, H) \in \mathcal{A}_1, We = Ve, (WH)_{ij} \geq \delta \forall (i, j) \in \{(i, j) : V_{ij} > 0\}\}.$$

By choosing appropriate δ, we note that $(W, 1/nee^{\top}) \in \mathcal{A}_3$ with $W_{ik} = 1/m \sum_l V_{il}$ for $i \in [m], k \in [r]$, then the set $\mathcal{A}_3$ is non-empty. We see that $\mathcal{A}_3$ is closed and bounded, and $D(V|WH))$ is continuous on $\mathcal{A}_3$, hence, Problem (1) has at least one solution.
Case 2 $\varepsilon > 0$. When $\varepsilon > 0$, we see $(0, 0)$ is a feasible solution of (19). Let

$$C = f_\varepsilon(0, 0) = mnr\varepsilon^2 - \log(r\varepsilon^2) + \sum_{i=1}^{m} \sum_{j=1}^{n} V_{ij} \log V_{ij} - 1.$$

We then can restrict our search for the optimal solutions of Problem (19) to the constraint set $\mathcal{A}_4 = \{(W, H) : W, H \geq 0, f_\varepsilon(W, H) \leq C\}$. Using inequality $\log x \leq x - 1$ for all $x > 0$, we obtain

$$\begin{aligned}
f_\varepsilon(W, H) &\geq \sum_{i=1}^{m} \sum_{j=1}^{n} (1 - V_{ij}) \big((WH)_{ij} + \varepsilon(Wee^{\top} + ee^{\top}H)_{ij} + \varepsilon^2 \big) + \sum_{i=1}^{m} \sum_{j=1}^{n} V_{ij} \log V_{ij} \\
&\geq \sum_{i=1}^{m} \sum_{j=1}^{n} (1 - V_{ij}) \big(\varepsilon \sum_k W_{ik} + \varepsilon \sum_k H_{kj} + \varepsilon^2 \big) + \sum_{i=1}^{m} \sum_{j=1}^{n} V_{ij} \log V_{ij} \\
&\geq (mn - 1)\varepsilon^2 + (1 - v_{\max})\varepsilon \big(n \sum_{ik} W_{ik} + m \sum_{kj} H_{kj} \big) + \sum_{i=1}^{m} \sum_{j=1}^{n} V_{ij} \log V_{ij},
\end{aligned}$$

where $v_{\max} = \max_{ij} V_{ij}$, we use $(WH)_{ij} \geq 0$ in the second inequality and $\sum_{ij} V_{ij} = 1$ in the third inequality. Hence, from $f_\varepsilon(W, H) \leq C$, W and H are bounded. Therefore, $\mathcal{A}_4$ is bounded. We also see that $\mathcal{A}_4$ is closed; hence, it is a compact set. The objective of (19) is continuous on this set since $\varepsilon > 0$. Therefore, Problem (19) has at least one solution.

(B) Let us fix $\varepsilon > 0$. Note that all points of $\mathcal{A}_3$ are feasible solutions of Problem (19). We hence have

$$\begin{aligned}
D^*(V, \varepsilon) &\leq \min_{(W,H) \in \mathcal{A}_3} f_\varepsilon(W, H) \\
&\leq \min_{(W,H) \in \mathcal{A}_3} D(V|WH) + \max_{(W,H) \in \mathcal{A}_3} (f_\varepsilon(W, H) - D(V|WH)) \quad (22) \\
&= D^*(V, 0) + \max_{(W,H) \in \mathcal{A}_3} (f_\varepsilon(W, H) - D(V|WH)),
\end{aligned}$$

where in the first inequality we use the fact that the optimal value over bigger set does not exceed the optimal value over smaller set. On the other hand, it follows from (20) and the

case $\sum_{i,j} V_{ij} = 1$ that, for $(W, H) \in \mathcal{A}_3$, we have

$$
f_\varepsilon(W, H) - D(V|WH) = \sum_{i=1}^{m} \sum_{j=1}^{n} \left(\varepsilon \sum_{k=1}^{r} W_{ik} + \varepsilon \sum_{k=1}^{r} H_{kj} + \varepsilon^2 \right)
$$

$$
- \sum_{i=1}^{m} \sum_{j=1}^{n} V_{ij} \log \frac{((W + \varepsilon ee^\top)(H + \varepsilon ee^\top))_{ij}}{(WH)_{ij}}
$$

$$
\leq n\varepsilon + mr\varepsilon + mn\varepsilon^2.
$$

Hence $D^*(V, \varepsilon) \leq D^*(V, 0) + n\varepsilon + mr\varepsilon + mn\varepsilon^2$. By exchanging the role of W and $H^\top$ and noting that $D(V|WH) = D(V^\top|H^\top W^\top)$, we can prove a similar bound $D^*(V, \varepsilon) \leq D^*(V, 0) + m\varepsilon + nr\varepsilon + mn\varepsilon^2$. Together with (21), we obtain Result (B).　　□

BSUM Framework for the MU

MU is a block successive upper-bound minimization algorithm (BSUM) in which each column of H and each row of W are updated by minimizing majorized functions of the KL objective. Let us first introduce BSUM, then derive MU for solving (2).

BSUM was proposed in [41] to solve the minimization problem in (12).

Putting in the notations of (12), first, let us formally define a majorized function.

Definition 4 A function $u_i : \mathcal{X}_i \times \mathcal{X} \to \mathbb{R}$ is said to majorize $f(x)$ if

$$
\begin{aligned}
&u_i(x_i, x) = f(x), \forall x \in \mathcal{X}, \\
&u_i(y_i, x) \geq f(x_1, \ldots, x_{i-1}, y_i, x_{i+1}, \ldots, x_s), \forall y_i \in \mathcal{X}_i, x \in \mathcal{X}.
\end{aligned}
\tag{23}
$$

Using the majorized functions, at iteration k, BSUM fixes the latest values of block $j \neq i$ and updates block x_i by

$$
x_i^k = \arg \min_{x_i \in \mathcal{X}_i} u_i(x_i, x_1^k, \ldots, x_{i-1}^k, x_i^{k-1}, x_{i+1}^{k-1}, \ldots, x_s^{k-1}).
\tag{24}
$$

From Definition 4 we have

$$
\begin{aligned}
f(x^k) &= u_1(x_1^k, x_1^k, \ldots, x_s^k) \geq u_1(x_1^{k+1}, x_1^k, \ldots, x_s^k) \\
&\geq f(x_1^{k+1}, x_2^k, \ldots, x_s^k) = u_2(x_2^k, x_1^{k+1}, x_2^k, \ldots, x_s^k) \\
&\geq u_2(x_2^{k+1}, x_1^{k+1}, x_2^k, \ldots, x_s^k) \geq f(x_1^{k+1}, x_2^{k+1}, \ldots, x_{s-1}^k, x_s^k) \geq \cdots \geq f(x^{k+1}).
\end{aligned}
$$

In other words, BSUM produces a non-increasing sequence $\{f(x^k)\}$. In the following, we give a majorized function for $D(t|Wh)$ defined in (7).

Proposition 10 [31, Lemma 3] *Let*

$$
u_{\mathrm{MU}}(h, h^k) := \sum_{i=1}^{m} \left((Wh)_i - t_i \sum_{l=1}^{r} \frac{W_{il}(h^k)_l}{(Wh^k)_i} \left(\log(W_{il} h_l) - \log \frac{W_{il}(h^k)_l}{(Wh^k)_i} \right) + t_i \log t_i - t_i \right).
$$

Then $u_{\mathrm{MU}}(h, h^k)$ majorizes the function $h \mapsto D(t|Wh)$.

When BSUM uses $u_{\mathrm{MU}}(\cdot, \cdot)$ to update a column h of H (similarly for a row of W) by $h^{k+1} = \arg\min_{h \geq \varepsilon} u_{\mathrm{MU}}(h, h^k)$, we obtain $(h^{k+1})_l = \max \left\{ \frac{(h^k)_l}{\sum_{i=1}^{m} W_{il}} \sum_{i=1}^{m} \frac{t_i W_{il}}{(Wh^k)_i}, \varepsilon \right\}$, leading to

the MU. More specifically, the updates of MU for solving Problem (2) are

$$W_{ik}^+ = \max\left\{ W_{ik}\frac{\sum_{l=1}^{n}\frac{H_{kl}V_{il}}{(WH)_{il}}}{\sum_{l=1}^{n}H_{kl}}, \varepsilon \right\}, H_{kj}^+ = \max\left\{ H_{kj}\frac{\sum_{l=1}^{m}\frac{W_{lk}V_{lj}}{(WH)_{lj}}}{\sum_{l=1}^{m}W_{lk}}, \varepsilon \right\}. \tag{25}$$

We note that the non-increasing property of $f(x^k)$ produced by BSUM for Problem (12) does not guarantee the convergence of $\{x^k\}$. To guarantee some convergence for $\{x^k\}$, we need $\varepsilon > 0$ for the objective function to be directionally differentiable, which is not the case when $\varepsilon = 0$, $(WH)_{ij} = 0$ and $V_{ij} > 0$ for some i, j.

Proof of Theorem 1

Theorem 1: Suppose Assumption 1 is satisfied. Let $\{x^k\}$ be the sequence generated by Algorithm 1. Let also $\Phi(x) = f(x) + \sum_i I_{\mathcal{X}_i}(x_i)$, where $I_{\mathcal{X}_i}$ is the indicator function of $\mathcal{X}_i$. We have

(i) $\Phi(x^k)$ is non-increasing;
(ii) Suppose $\mathcal{X}_i \subset \text{int dom } \kappa_i$. If $\{x^k\}$ is bounded and $\kappa_i(x_i)$ is strongly convex on bounded subsets of $\mathcal{X}_i$ that contain $\{x_i^k\}$, then every limit point of $\{x^k\}$ is a critical point of Φ;
(iii) If together with the conditions in (ii) we assume that $\nabla\kappa_i$ and $\nabla_i f$ are Lipschitz continuous on bounded convex subsets of $\mathcal{X}_i$ that contain $\{x_i^k\}$, then the whole sequence $\{x^k\}$ converges to a critical point of Φ.

Proof We follow the methodology established in [5] that bases on the following theorem to prove the global convergence of BMD. □

Theorem 3 *Let $\Phi : \mathbb{R}^N \to (-\infty, +\infty]$ be a proper and lower semicontinuous function which is bounded from below. Let $\mathcal{A}$ be a generic algorithm which generates a bounded sequence $\{x^k\}$ by $x^{k+1} \in \mathcal{A}(x^k)$, $k = 0, 1, \ldots$ Assume that the following conditions are satisfied.*
(B1) ***Sufficient decrease property****: There exists some $\rho_1 > 0$ such that*

$$\rho_1\|x^k - x^{k+1}\|^2 \leq \Phi(x^k) - \Phi(x^{k+1}), \forall k = 0, 1, \ldots.$$

(B2) ***Boundedness of subgradient****: There exists some $\rho_2 > 0$ such that*

$$\|w^{k+1}\| \leq \rho_2\|x^k - x^{k+1}\|, w^{(k)} \in \partial\Phi(x^k), \forall k = 0, 1, \ldots$$

(B3) ***KL property****: Φ is a KL function.*
(B4) ***A continuity condition****: If a subsequence $\{x^{k_n}\}$ converges to $\bar{x}$ then $\Phi(x^{k_n}) \to \Phi(\bar{x})$ as $n \to \infty$.*
Then $\sum_{k=1}^{\infty}\|x^k - x^{k+1}\| < \infty$ and $\{x^k\}$ converges to a critical point of Φ.

We will prove Theorem 1 (iii) by verifying all the conditions in Theorem 3 for Φ. Let us first recall the following three-point property.

Proposition 11 (Property 1 of [48]) *Let $z^+ = \arg\min_{u \in \mathcal{X}_i} \phi(u) + \mathcal{B}_\kappa(u, z)$, where ϕ is a proper convex function, $\mathcal{X}_i$ is convex and $\mathcal{B}_\kappa$ is the Bregman distance with respect to κ_i. Then for all $u \in \mathcal{X}_i$ we have*

$$\phi(u) + \mathcal{B}_\kappa(u, z) \geq \phi(z^+) + \mathcal{B}_\kappa(z^+, z) + \mathcal{B}_\kappa(u, z^+).$$

Denote $\Phi_i^k(x_i) = f_i^k(x_i) + I_{\mathcal{X}_i}(x_i)$. We have

$$
\begin{aligned}
f_i^k(x_i^{k+1}) &\leq f_i^k(x_i^k) + \langle \nabla f_i^k(x_i^k), x_i^{k+1} - x_i^k \rangle + L_i^k \mathcal{B}_{\kappa_i}(x_i^{k+1}, x_i^k) \\
&\leq f_i^k(x_i^k) + \langle \nabla f_i^k(x_i^k), x_i - x_i^k \rangle + L_i^k \mathcal{B}_{\kappa_i}(x_i, x_i^k) - L_i^k \mathcal{B}_{\kappa_i}(x_i, x_i^{k+1}) \\
&\leq f_i^k(x) + L_i^k \mathcal{B}_{\kappa_i}(x, x_i^k) - L_i^k \mathcal{B}_{\kappa_i}(x, x_i^{k+1}),
\end{aligned}
$$

where the first inequality uses Assumption 1, the second inequality uses Proposition 11 applied for (13) and the third uses convexity of f_i. Note that $I_{\mathcal{X}_i}(x) = 0$ if $x \in \mathcal{X}$ and $I_{\mathcal{X}_i}(x) = +\infty$ otherwise. Hence, for all x_i,

$$
\Phi_i^k(x_i^{k+1}) \leq \Phi(x_1^{k+1}, \ldots, x_{i-1}^{k+1}, x_i, x_{i+1}^k, \ldots, x_s^k) + L_i^k\big(\mathcal{B}_{\kappa_i}(x_i, x_i^k) - \mathcal{B}_{\kappa_i}(x_i, x_i^{k+1})\big). \quad (26)
$$

Choosing $x = x_i^k$ in (26), we have $\Phi_i^k(x_i^{k+1}) \leq \Phi_i^k(x_i^k) - \underline{L}_i \mathcal{B}_{\kappa_i}(x_i^k, x_i^{k+1})$. Therefore,

$$
\Phi(x^{k+1}) - \Phi(x^k) = \sum_{i=1}^{s} \Phi_i^k(x_i^{k+1}) - \Phi_i^k(x_i^k) \leq - \sum_{i=1}^{s} \underline{L}_i \mathcal{B}_{\kappa_i}(x_i^k, x_i^{k+1}). \quad (27)
$$

From (27), we see that $\Phi(x^k)$ is non-increasing. Theorem 1(i) is proved.

As $\{x^k\}$ is assumed to be bounded, then there exists positive constants A_i such that $\|x_i^k\| \leq A_i$, $i = 1, \ldots, s$, for all k. Suppose a subsequence $x^{k_n} \to x^*$. As $\mathcal{X}$ is closed convex set and $x^{k_n} \in \mathcal{X}$, then $x^* \in \mathcal{X}$. Hence, $\Phi(x^{k_n}) \to \Phi(x^*)$, i.e., the continuity condition (B4) is satisfied.

Since κ_i is strongly convex on the sets $\{x_i : x_i \in \mathcal{X}_i, \|x_i^k\| \leq A_i\}$, we have $\mathcal{B}_{\kappa_i}(x_i^k, x_i^{k+1}) \geq \sigma_i/2 \|x_i^k - x_i^{k+1}\|^2$, for some constant σ_i. Hence, from (27) we derive the sufficient decrease property (B1) and $\sum_{k=1}^{\infty} \|x^k - x^{k+1}\|^2 < +\infty$. Consequently, $\|x^k - x^{k+1}\| \to 0$. Then we also have $x^{k_n+1} \to x^*$. In (26) we let $k = k_n \to \infty$ and note that $x_i^* \in \mathcal{X}_i \subset \text{int dom } \kappa_i$, then we obtain $\Phi(x^*) \leq \Phi(x_1^*, \ldots, x_i, \ldots, x_s^*)$, $\forall x_i$. This means x_i^* is a local minimizer of $\min_{x_i} \Phi(x_1^*, \ldots, x_i, \ldots, x_s^*)$. Hence $0 \in \partial_i \Phi(x^*)$, for $i = 1, \ldots s$. In other words, we have $0 \in \partial \Phi(x^*)$, i.e., Theorem 1 (ii) is proved.

To prove Theorem 1 (iii), it remains to verify the boundedness of subgradients. From [2, Proposition 2.1] we have

$$
\partial \Phi(x^k) = \partial \Phi_1(x^k) \times \partial_s \Phi(x^k) = \{\nabla_1 f(x^k) + \partial I_{\mathcal{X}_1}(x_1^k)\} \times \cdots \times \{\nabla_s f(x^k) + \partial I_{\mathcal{X}_s}(x_s^k)\}.
$$

It follows from (13) that $0 \in \nabla f_i^k(x_i^k) + \partial I_{\mathcal{X}_i}(x_i^{k+1}) + L_i^{k+1}\big(\nabla \kappa_i(x_i^{k+1}) - \nabla \kappa_i(x_i^k)\big)$. Hence,

$$
\begin{aligned}
\omega_i^{k+1} &= \nabla_i f(x^{k+1}) + \partial I_{\mathcal{X}_1}(x_1^{k+1}) \\
&= \nabla_i f(x^{k+1}) - \nabla f_i^k(x_i^k) + L_i^k\big(\nabla \kappa_i(x_i^k) - \nabla \kappa_i(x_i^{k+1})\big) \in \partial \Phi_i(x^{k+1}).
\end{aligned} \quad (28)
$$

As $\nabla_i f$ and $\nabla \kappa_i$ are Lipschitz continuous on $\{x_i : \|x_i^k\| \leq A_i\}$, we derive from (28) that there exists some constant a_i such that $\|\omega_i^{k+1}\| \leq a_1 \|x^k - x^{k+1}\|$, for $i = 1, \ldots, s$. Then the boundedness of subgradients in (B4) is satisfied. Theorem 1 is proved. $\qquad \square$

Proof of Proposition 9

Proposition 9: The objective function for the perturbed KL-NMF problem (1) is non-increasing under the updates of Algorithm 2.

Proof We note that if $g(x)$ is a self-concordant function with constant **c** then $\mathbf{c}^2 g(x)$ is a standard self-concordant function. Hence, using the result of [47, Theorem 6] (see also

Sect. 2.4), we derive that the objective function of (1) is strictly decreasing when a damp Newton step is used. We now prove the proposition for the case when a full Newton step is used, that is when the gradient $f'_{W_{ik}} \leq 0$ or $\lambda \leq 0.683802$ and the update of W_{ik} then is

$$W_{ik}^+ = s = W_{ik} - \frac{f'_{W_{ik}}}{f''_{W_{ik}}}.$$

Consider the case $f'_{W_{ik}} \leq 0$. Let us use $f_{W_{ik}}$ to denote the objective of (1) with respect to W_{ik}. Considering the function $W_{ik} \mapsto g(W_{ik}) = f'_{W_{ik}}$ with $W_{ik} \geq 0$, we see that $g(W_{ik})$ is a concave function. Hence, we have $g(W_{ik}^+) - g(W_{ik}) \leq g'(W_{ik})(W_{ik}^+ - W_{ik})$. This implies that $f'_{W_{ik}^+} \leq f'_{W_{ik}} - f''_{W_{ik}} \frac{f'_{W_{ik}}}{f''_{W_{ik}}} = 0$. Since $W_{ik} - W_{ik}^+ = \frac{f'_{W_{ik}}}{f''_{W_{ik}}} \leq 0$ and $W_{ik} \mapsto f_{W_{ik}}$ is convex, we then obtain $f_{W_{ik}} \geq f_{W_{ik}^+} + f'_{W_{ik}^+}(W_{ik} - W_{ik}^+) \geq f_{W_{ik}^+}$. Hence, the objective of (1) is non-increasing when we update W_{ik} to W_{ik}^+.

Consider the case $\lambda \leq 0.683802$. Denote $W_{ik}^\alpha = W_{ik} + \alpha d$. When $\alpha = 1$ we have $W_{ik}^\alpha = s$, that is when a full proximal Newton step is applied. It follows from [47, Inequality (58)] that $f_{W_{ik}^\alpha} \leq f_{W_{ik}} - \alpha \lambda^2 + \omega_*(\alpha \lambda)$ for $\alpha \in (0, 1]$, where $\omega_*(t) = -t - \log(1 - t)$. Hence, when $\alpha = 1$ we get $f_{W_{ik}^\alpha} \leq f_{W_{ik}} - (\lambda^2 + \lambda + \log(1 - \lambda))$. It is not difficult to see that when $\lambda \leq 0.683802$ the value of $\lambda^2 + \lambda + \log(1 - \lambda)$ is positive. Hence, in this case, a full proximal Newton step also decreases the objective function. □

References

1. Ang, A.M.S., Gillis, N.: Accelerating nonnegative matrix factorization algorithms using extrapolation. Neural Comput. **31**(2), 417–439 (2019)
2. Attouch, H., Bolte, J., Redont, P., Soubeyran, A.: Proximal alternating minimization and projection methods for nonconvex problems: an approach based on the Kurdyka-Łojasiewicz inequality. Math. Oper. Res. **35**(2), 438–457 (2010)
3. Bauschke, H.H., Bolte, J., Teboulle, M.: A descent lemma beyond Lipschitz gradient continuity: first-order methods revisited and applications. Math. Oper. Res. **42**(2), 330–348 (2017). https://doi.org/10.1287/moor.2016.0817
4. Bioucas-Dias, J.M., Plaza, A., Dobigeon, N., Parente, M., Du, Q., Gader, P., Chanussot, J.: Hyperspectral unmixing overview: geometrical, statistical, and sparse regression-based approaches. IEEE J. Select. Top. Appl. Earth Obs. Remote Sens. **5**(2), 354–379 (2012)
5. Bolte, J., Sabach, S., Teboulle, M.: Proximal alternating linearized minimization for nonconvex and nonsmooth problems. Math. Program. **146**(1), 459–494 (2014)
6. Carbonetto, P., Luo, K., Dey, K., Hsiao, J., Stephens, M.: fastTopics: fast algorithms for fitting topic models and non-negative matrix factorizations to count data (2021). R package version 0.4-11. https://github.com/stephenslab/fastTopics
7. Chambolle, A., Pock, T.: A first-order primal-dual algorithm for convex problems with applications to imaging. J. Math. Imaging Vis. **40**(1), 120–145 (2011)
8. Chi, E.C., Kolda, T.G.: On tensors, sparsity, and nonnegative factorizations. SIAM J. Matrix Anal. Appl. **33**(4), 1272–1299 (2012)
9. Cichocki, A., Zdunek, R., Phan, A.H., Amari, S.: Nonnegative matrix and tensor factorizations: applications to exploratory multi-way data analysis and blind source separation. Wiley (2009)
10. Dey, K.K., Hsiao, C.J., Stephens, M.: Visualizing the structure of RNA-SEQ expression data using grade of membership models. PLoS Genet. **13**(3), e1006599 (2017)
11. Ding, C., Li, T., Peng, W.: On the equivalence between non-negative matrix factorization and probabilistic latent semantic indexing. Comput. Stat. Data Anal. **52**(8), 3913–3927 (2008)
12. Dolan, E.D., Moré, J.J.: Benchmarking optimization software with performance profiles. Math. Program. **91**(2), 201–213 (2002)
13. Févotte, C., Bertin, N., Durrieu, J.L.: Nonnegative matrix factorization with the Itakura-Saito divergence: with application to music analysis. Neural Comput. **21**(3), 793–830 (2009)
14. Févotte, C., Cemgil, A.T.: Nonnegative matrix factorizations as probabilistic inference in composite models. In: 2009 17th European Signal Processing Conference, pp. 1913–1917 (2009)

15. Févotte, C., Idier, J.: Algorithms for nonnegative matrix factorization with the β-divergence. Neural Comput. **23**(9), 2421–2456 (2011). https://doi.org/10.1162/NECO_a_00168
16. Finesso, L., Spreij, P.: Nonnegative matrix factorization and I-divergence alternating minimization. Linear Algebra Appl. **416**(2), 270–287 (2006). https://doi.org/10.1016/j.laa.2005.11.012
17. Fu, X., Huang, K., Sidiropoulos, N.D., Ma, W.K.: Nonnegative matrix factorization for signal and data analytics: Identifiability, algorithms, and applications. IEEE Signal Process. Mag. **36**(2), 59–80 (2019)
18. Gillis, N.: The why and how of nonnegative matrix factorization. In: Suykens, J., Signoretto, M., Argyriou, A. (eds.) Regularization, Optimization, Kernels, and Support Vector Machines, Machine Learning and Pattern Recognition chap 12, pp. 257–291. Chapman & Hall/CRC, Boca Raton, Florida (2014)
19. Gillis, N.: Introduction to nonnegative matrix factorization. SIAG/OPT Views News **25**(1), 7–16 (2017)
20. Gillis, N., Hien, L.T.K., Leplat, V., Tan, V.Y.: Distributionally robust and multi-objective nonnegative matrix factorization. IEEE Trans. Pattern Anal. Mach. Intell. (2021). https://doi.org/10.1109/TPAMI.2021.3058693
21. Gonzalez, E.F., Zhang, Y.: Accelerating the Lee-Seung algorithm for nonnegative matrix factorization. Technical report, Rice University (2005). https://scholarship.rice.edu/handle/1911/102034%20Technical%20report
22. Guillamet, D., Vitrià, J.: Non-negative matrix factorization for face recognition. In: Escrig, M.T., Toledo, F., Golobardes, E. (eds.) Topics in Artificial Intelligence, pp. 336–344. Springer, Berlin (2002)
23. Hasinoff, S.W.: Photon, Poisson Noise (2014). http://people.csail.mit.edu/hasinoff/pubs/hasinoff-photon-2011-preprint.pdf
24. Hien, L.T.K., Gillis, N., Patrinos, P.: Inertial block proximal method for non-convex non-smooth optimization. In: Thirty-seventh International Conference on Machine Learning (ICML 2020) (2020)
25. Ho, N.D., Dooren, P.V.: Non-negative matrix factorization with fixed row and column sums. Linear Algebra and its Applications **429**(5), 1020–1025 (2008). Special Issue devoted to selected papers presented at the 13th Conference of the International Linear Algebra Society
26. Hong, M., Razaviyayn, M., Luo, Z.Q., Pang, J.S.: A unified algorithmic framework for block-structured optimization involving big data: with applications in machine learning and signal processing. IEEE Signal Process. Mag. **33**(1), 57–77 (2015)
27. Hoyer, P.O.: Non-negative matrix factorization with sparseness constraints. J. Mach. Learn. Res. **5**, 1457–1469 (2004)
28. Hsieh, C.J., Dhillon, I.S.: Fast coordinate descent methods with variable selection for non-negative matrix factorization. In: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1064–1072 (2011)
29. Huang, K., Sidiropoulos, N.D., Liavas, A.P.: A flexible and efficient algorithmic framework for constrained matrix and tensor factorization. IEEE Trans. Signal Process. **64**(19), 5052–5065 (2016)
30. Kim, C., Kim, Y., Klabjan, D.: Scale invariant power iteration (2019). arXiv:1905.09882
31. Lee, D.D., Seung, H.S.: Learning the parts of objects by nonnegative matrix factorization. Nature **401**, 788–791 (1999)
32. Lee, D.D., Seung, H.S.: Algorithms for non-negative matrix factorization. In: Advances in Neural Information Processing Systems, pp. 556–562 (2001)
33. Leskovec, J., Rajaraman, A., Ullman, J.D.: Mining of Massive Datasets, 2nd edn. Cambridge University Press (2014). http://mmds.org
34. Li, L., Lebanon, G., Park, H.: Fast Bregman divergence NMF using Taylor expansion and coordinate descent. In: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-12), pp. 307–315. Association for Computing Machinery, New York, NY, USA (2012)
35. Lin, C.J.: On the convergence of multiplicative update algorithms for nonnegative matrix factorization. IEEE Trans. Neural Netw. **18**(6), 1589–1596 (2007)
36. Lin, X., Boutros, P.C.: Optimization and expansion of non-negative matrix factorization. BMC Bioinform. **21**(1), 1–10 (2020)
37. Lu, H., Freund, R.M., Nesterov, Y.: Relatively smooth convex optimization by first-order methods, and applications. SIAM J. Optim. **28**, 333–354 (2018)
38. Lucy, L.B.: An iterative technique for the rectification of observed distributions. Astron. J. **79**, 745 (1974)
39. Ma, W., Bioucas-Dias, J.M., Chan, T., Gillis, N., Gader, P., Plaza, A.J., Ambikapathi, A., Chi, C.: A signal processing perspective on hyperspectral unmixing: insights from remote sensing. IEEE Signal Process. Mag. **31**(1), 67–81 (2014)
40. Nesterov, Y.: Lectures on Convex Optimization. Springer (2018)
41. Razaviyayn, M., Hong, M., Luo, Z.: A unified convergence analysis of block successive minimization methods for nonsmooth optimization. SIAM J. Optim. **23**(2), 1126–1153 (2013)
42. Richardson, W.H.: Bayesian-based iterative method of image restoration. JoSA **62**(1), 55–59 (1972)

43. Shahnaz, F., Berry, M.W., Pauca, V., Plemmons, R.J.: Document clustering using nonnegative matrix factorization. Inf. Process. Manag. **42**(2), 373–386 (2006). http://www.sciencedirect.com/science/article/pii/S0306457304001542

44. Sun, D.L., Févotte, C.: Alternating direction method of multipliers for non-negative matrix factorization with the beta-divergence. In: 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 6201–6205 (2014)

45. Takahashi, N., Hibi, R.: Global convergence of modified multiplicative updates for nonnegative matrix factorization. Comput. Optim. Appl. **57**(2), 417–440 (2014)

46. Teboulle, M., Vaisbourd, Y.: Novel proximal gradient methods for nonnegative matrix factorization with sparsity constraints. SIAM J. Imaging Sci. **13**(1), 381–421 (2020)

47. Tran-Dinh, Q., Kyrillidis, A., Cevher, V.: Composite self-concordant minimization. J. Mach. Learn. Res. **16**(12), 371–416 (2015)

48. Tseng, P.: On accelerated proximal gradient methods for convex-concave optimization. Tech. rep. (2008)

49. Xu, Y., Yin, W.: A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion. SIAM J. Imaging Sci. **6**(3), 1758–1789 (2013). https://doi.org/10.1137/120887795

50. Yanez, F., Bach, F.: Primal-dual algorithms for non-negative matrix factorization with the Kullback-Leibler divergence. In: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 2257–2261 (2017)

51. Zhong, S., Ghosh, J.: Generative model-based document clustering: a comparative study. Knowl. Inf. Syst. **8**(3), 374–384 (2005)