NON-NEGATIVE MATRIX FACTORIZATION WITH FACTORIZABLE FEATURE MATRIX

MARCEL OLIVER AND SERGIY VASYLKEVYCH

ABSTRACT. We study the following generalization of the classical non-negative matrix factorization (NMF) problem: Given a non-negative matrix V, i.e., a matrix with non-negative elements, find a low rank approximation $V \approx CWH$ where all right hand factors are non-negative, C is a given generally non-invertable "feature map," and V and H are low rank factors to be determined by best-approximation in a weighted Frobenius norm. We shall refer to this setting as the FFNMF problem. In this paper, we propose a non-multiplicatively regularized gradient descent algorithm for the FFNMF problem, prove its consistency, and show that a stationary or a limit point of the algorithm is a stationary point for the cost functional except possibly at the boundary of the admissible region, where the cost is then locally increasing when moving away from the boundary.

1. INTRODUCTION

Non-negative matrix factorization (NMF) seeks to approximate a non-negative $m \times n$ matrix V (in this context, a matrix is called non-negative if all of its elements are non-negative) by a product

$$V \approx WH \tag{1}$$

of non-negative matrices W and H of dimensions $m \times k$ and $k \times n$, respectively, with a given and typically low maximal rank k. It forms the basis of unsupervised learning and data reduction algorithms with applications to image recognition [8], environmental monitoring [2, 4, 12], speech recognition [5], and data mining and collaborative filtering [19].

The NMF problem is typically cast as the minimization of $V - WH$ with respect to the Frobenius norm [11], the Kullback–Leibler divergence [7, 19, 3, 5], or, more generally, Bregman divergences [3]. In all cases, the problem is that of non-convex constrained optimization. In this paper, we restrict ourselves to the Frobenius norm setting.

The factorization (1) can be interpreted as an approximation of V in terms of "basis elements" or "features" given by the columns of W with non-negative expansion coefficients. In some applications, however, the features have an additional pre-determined structure which should be encoded into the factorization. In this paper, we specifically look at what we call the *factorizable feature matrix NMF* (FFNMF) problem, which is of the form

$$V \approx CWH\,. \tag{2}$$

The matrix C is a known non-negative *feature map* which can be interpreted as a mapping from the logical degrees of freedom encoded in the columns of W to the

Date: September 26, 2013.

observable degrees of freedom. In other words, we impose a factorization of the feature matrix CW into a known and an unknown factor; as in (1), the task of FFNMF is to determine the non-negative matrices W and H.

For example, in an application to environmental monitoring similar to that reported in [2], the feature map C may map chemicals into outputs of measuring devices, decompose complex chemicals into more elementary components, or aggregate compounds into broader classes relevant for further analysis. In the context of collaborative filtering, the feature map may encode existing knowledge, such as users' demographic and social data, or items' genre information.

Interest in this problem is not new. Guillamet, Bressan, and Vitrià [6] considered the closely related problem $V \approx WHC$, albeit only for C square. Dhillon and Sra [3, 15] discuss a general class of "weighted non-negative matrix approximation" problems which encompasses the former.[1] Their setting is clearly general enough to derive multiplicative update rules for solving the FFNMF problem as stated; however, the weights introduced in their explicit examples are different from the ones we are using in this paper, and the authors do not consider regularizations as shall be explained below.

Multiplicative update rules go back to the seminal paper by Lee and Seung [7]. They remain popular as they are simple to implement and typically give good results. From the theoretical point of view, however, the situation is not entirely satisfactory as convergence is not guaranteed and, moreover, convergence does not imply that the limit point is a local minimizer of the cost functional [1, 9, 10]. A second issue is that basic multiplicative updates may become singular when the sequence of iterates approaches the boundary of the admissible region.

In practice, therefore, the update rules need to be regularized near the boundary of the admissible region. Lin [10] uses a regularization of the generating auxiliary function for the classical Lee and Seung [7] update and shows that the resulting algorithm preserves monotonicity. In this paper, we provide another such regularization which is similar, but not identical to Lin's. We show that that a stationary or a limit point of the algorithm is necessarily a stationary point for the cost functional except possibly at the boundary of the admissible region so long as the cost is locally decreasing toward the boundary.

The purpose of our paper is three-fold. First, we explicitly state a regularized algorithm for a weighted variant of FFNMF, to be introduced in Section 2 below, which we developed in a particular industrial context and which should be more widely applicable in a variety of modeling scenarios. Second, we show that this algorithm can be analyzed in the spirit of previous work on algorithms for scaled gradient-descent algorithms for NMF-type problems. Third, we clarify the results of this analysis since the respective statements are not always sufficiently precise in the NMF literature.

The article is structured as follows. In Section 2, we introduce basic notation and state our regularized FFNMF algorithm. In Section 3, we prove some auxiliary results on the non-negative least square problem which can be used to analyze the NNMF updates; their application to the full FFNMF problem is discussed in

[1] Dhillon and Sra advocate the term *non-negative matrix approximation* (NNMA) in place of *non-negative matrix factorization* (NMF). However, for the better or worse, the latter remains overwhelmingly used in the literature.

Section 4. In Section 5, we discuss simplified update rules and compare them to the literature. The paper concludes with a short discussion of the results.

2. An algorithm for weighted FFNMF

In the following, we write Mat^+ to denote the set of matrices with non-negative real elements, $\mathrm{Mat}_{m,n}$ to denote the space of real $m \times n$ matrices, and set $\mathrm{Mat}^+_{m,n} \equiv \mathrm{Mat}_{m,n} \cap \mathrm{Mat}^+$. Further, for matrices A and B, we write $A \odot B$, A/B, and $\sqrt{A}$ to denote element-wise multiplication, division, and square root; standard matrix multiplication is written AB as usual. In complex formulas, standard matrix multiplication takes precedence over element-wise operators. Finally, $\mathbb{1}$ denotes a matrix with all entries equal to one with implicitly defined dimension, and $\Sigma(A)$ denotes the sum of all elements of A.

Then, if $\mathcal{D}\colon \mathrm{Mat}^+_{m,n} \times \mathrm{Mat}^+_{m,n} \to \mathbb{R}^+$ is some measure of distortion, $V \in \mathrm{Mat}^+_{m,n}$, and k is a fixed desired rank of the approximation, the task is to *minimize*

$$F(W,H) \equiv \mathcal{D}(V, WH) \tag{3}$$

over all matrices $W \in \mathrm{Mat}^+_{m,k}$ *and* $H \in \mathrm{Mat}^+_{k,n}$. The choice of $\mathcal{D}$ is dictated by the application. We note that Dhillon and Sra [3] set up a general framework in which $\mathcal{D}$ can be any so-called Bregman divergence. Here, however, we restrict ourselves to cost functions that arise via a Frobenius norm. The classical choice

$$\mathcal{D}(A,B) = \tfrac{1}{2}\,\|A-B\|^2 \equiv \tfrac{1}{2}\sum_{i,j}|A_{ij}-B_{ij}|^2 \tag{4}$$

for the NMF problem (1) has been suggested in [11, 13]; Lee and Seung [7] proposed a simple iterative algorithm with multiplicative update rules for this minimization problem. A natural slight generalization is weighted NMF where, for a prescribed weight matrix $M \in \mathrm{Mat}^+$,

$$\mathcal{D}(A,B) = \tfrac{1}{2}\,\|\sqrt{M} \odot (A-B)\|^2 \,. \tag{5}$$

Extensions of Lee and Seung's method to weighted NMF can be found in [3, 6, 19].

For FFNMF, we set $F(W,H) = \mathcal{D}(V, CWH)$ and take the weighted Frobenius squared distance function (5), so that the FFNMF problem for given $M, V \in \mathrm{Mat}^+_{m,n}$ and $C \in \mathrm{Mat}^+_{m,l}$ reads: *minimize*

$$F(W,H) \equiv \tfrac{1}{2}\,\|\sqrt{M} \odot (V - CWH)\|^2 \tag{6}$$

where the minimum is taken over all $W \in \mathrm{Mat}^+_{l,k}$ *and* $H \in \mathrm{Mat}^+_{k,n}$. Clearly, when $C = I$, the problem reduces to weighted NMF; if further $M = \mathbb{1}$, it reduces to the classical NMF problem. We emphasize that C is not assumed to be invertable or even square. Typically, in applications $k \ll m, l, n$, however, we do not require this explicitly. The efficient choice of the approximation rank k is a problem which is still not completely solved, for a discussion, we refer to [18].

The technical advantage of using the Frobenius norm is that it comes from an inner product of the form

$$\langle A, B\rangle = \Sigma(A \odot B) = \sum_{i,j} A_{ij}\, B_{ij} \,. \tag{7}$$

We note, as shall become useful when computing gradient descent directions for (6), that

$$\langle AB, C\rangle = \langle B, A^T C\rangle = \langle A, CB^T\rangle \tag{8}$$

for all matrices A, B, C with compatible dimensions.

Let $\varepsilon > 0$ be a regularization parameter which is fixed. For matrices X, A, and B of same dimension, we define define the matrix $X_\varepsilon(A, B)$ by

$$[X_\varepsilon(A,B)]_{ij} = \begin{cases} \frac{\varepsilon}{\Sigma(A)+1} & \text{if } X_{ij} < \frac{\varepsilon}{\Sigma(A)+1} \text{ and } A_{ij} - B_{ij} < 0\,, \\ X_{ij} & \text{otherwise}\,. \end{cases} \tag{9}$$

We propose the iterative algorithm for solving the minimization problem (6) defined by the update rules

$$A_W = C^T(M \odot CWH)H^T\,, \qquad B_W = C^T(M \odot V)H^T\,, \tag{10a}$$

$$W^{\text{new}} = W - W_\varepsilon + \frac{(\varepsilon\mathbb{1} + B_W) \odot W_\varepsilon}{A_W + \varepsilon\mathbb{1}}\,, \tag{10b}$$

$$A_H = W^T C^T (M \odot CWH)\,, \qquad B_H = W^T C^T (M \odot V)\,, \tag{10c}$$

$$H^{\text{new}} = H - H_\varepsilon + \frac{(\varepsilon\mathbb{1} + B_H) \odot H_\varepsilon}{A_H + \varepsilon\mathbb{1}}\,, \tag{10d}$$

where $W_\varepsilon \equiv W_\varepsilon(A_W, B_W)$ and $H_\varepsilon \equiv H_\varepsilon(A_H, B_H)$, and $(W^{\text{new}}, H^{\text{new}})$ is the one step update of (W, H). The order in which the update is performed is non-essential, however, the new value of the first updated component must be used in the rule for the second one.

Rewriting the update rule additively, we can interpret the algorithm as a scaled gradient descent for (6):

$$W^{\text{new}} = W - \gamma_W \odot \nabla_W F\,, \tag{11a}$$

$$H^{\text{new}} = H - \gamma_H \odot \nabla_H F\,, \tag{11b}$$

where

$$\nabla_W F = A_W - B_W\,, \tag{12a}$$

$$\nabla_H F = A_H - B_H\,, \tag{12b}$$

and the scaling coefficients are given by

$$\gamma_W = \frac{W_\varepsilon}{A_W + \varepsilon\mathbb{1}}\,, \tag{13a}$$

$$\gamma_H = \frac{H_\varepsilon}{A_H + \varepsilon\mathbb{1}}\,. \tag{13b}$$

When $C = I$ and $\varepsilon = 0$, (10) coincides with the WNMF update [19]; if furthermore $M = \mathbb{1}$, we reduce to the classical NMF algorithm by Lee and Seung [7].

3. Non-negative least-squares

As commonplace in the NMF literature, we break up the analysis into interleaved non-negative least-squares (NNLS) problems. Each of these problems is convex, although the entire FFNMF problem, discussed in Section 4, is not. We use the technique of auxiliary functions which was introduced to the NMF literature in [7] and has been used by many authors since.

In the following, we identify matrices $Y \in \mathrm{Mat}_{m,n}$ with vectors $y \in \mathbb{R}^{mn} \equiv \mathbb{R}^d$. The inner product (7) is thus identified with the canonical inner product on $\mathbb{R}^d$, a linear operator $L\colon \mathrm{Mat}_{m,n} \to \mathrm{Mat}_{p,q}$ is identified, without change in notation, with $L \in \mathrm{Mat}_{mn,pq} \equiv \mathrm{Mat}_{d,\ell}$, and its adjoint L^* with respect to the inner product (7) is identified with the corresponding matrix transpose.

Definition 1. *Let $D \subset \mathbb{R}^d$ and $f\colon D \to \mathbb{R}$. Then $G\colon D \times D \to \mathbb{R}$ is called an auxiliary function for f if for all $x, x' \in D$,*

$$G(x,x') \geq f(x) \quad \textit{and} \quad G(x,x) = f(x)\,. \tag{14}$$

Lemma 2 ([7])**.** *Suppose G is an auxiliary function for f. Then f is non-increasing under the update rule*

$$x^{\text{new}} = \operatorname*{arg\,min}_{y \in D} G(y,x)\,. \tag{15}$$

Moreover, $f(x^{\text{new}}) = f(x)$ if and only if $x = \operatorname{arg\,min}_{y \in D} G(y,x)$.

We now let $D = \mathbb{R}^d_+$, i.e., the set of vectors in $\mathbb{R}^d$ with non-negative components, fix $y \in \mathbb{R}^\ell_+$, and consider the so-called non-negative least squares (NNLS) problem: *minimize*

$$f(x) = \tfrac{1}{2}\,\langle Lx - y, Lx - y\rangle = \tfrac{1}{2}\,\|Lx - y\|^2 \tag{16}$$

over all vectors $x \in D$.

Proposition 3. *Let $K(x)\colon \mathbb{R}^d \to \mathbb{R}^d$ be a family of self-adjoint operators such that $K(x) - L^*L$ is positive semi-definite for every $x \in D$. Then*

$$G(x,x') = f(x') + \langle x - x', L^*(Lx' - y)\rangle + \tfrac{1}{2}\,\langle x - x', K(x')(x - x')\rangle \tag{17}$$

is an auxiliary function for f.

Proof. Clearly, $G(x,x) = f(x)$. Further, we can rewrite (16) as

$$f(x) = f(x') + \langle x - x', L^*(Lx' - y)\rangle + \tfrac{1}{2}\,\langle x - x', L^*L(x - x')\rangle\,. \tag{18}$$

Comparing (17) and (18), we obtain

$$G(x,x') - f(x) = \tfrac{1}{2}\langle x - x', [K(x') - L^*L](x - x')\rangle \geq 0\,. \tag{19}$$

Thus, $K(x') - L^*L$ being positive semi-definite is equivalent to G being an auxiliary function. □

Proposition 4. *Assume that $L \in \mathrm{Mat}^+$. For every $x \in D$, set $x_\varepsilon = x_\varepsilon(L^*Lx, L^*y)$ and define the linear operator*

$$K(x)z = \frac{L^*Lx + \varepsilon\mathbb{1}}{x_\varepsilon} \odot z \tag{20}$$

for every $z \in \mathbb{R}^d$. Then G given by (17) is an auxiliary function for f.

Proof. Since $K(x)$ acts multiplicatively, it is self-adjoint. Hence, by Proposition 3, it suffices to check that $K(x) - L^*L$ is positive semi-definite. We begin by noting that, by assumption, $A \equiv L^*L \in \mathrm{Mat}^+_{d,d}$. Thus, for $z \in \mathbb{R}^d$ and $x \in \mathbb{R}^d_+$,

$$\begin{aligned}
\langle z \odot x, Ax \odot z\rangle - \langle z \odot x, A(z \odot x)\rangle &= \sum_{\alpha,\beta} x_\alpha\, z_\alpha^2\, A_{\alpha\beta}\, x_\beta - x_\alpha\, z_\alpha\, A_{\alpha\beta}\, x_\beta\, z_\beta \\
&= \sum_{\alpha,\beta} x_\alpha\, A_{\alpha\beta}\, x_\beta \left(\tfrac{1}{2}\, z_\alpha^2 + \tfrac{1}{2}\, z_\beta^2 - z_\alpha\, z_\beta\right) \\
&= \frac{1}{2}\sum_{\alpha,\beta} x_\alpha\, A_{\alpha\beta}\, x_\beta\, (z_\alpha - z_\beta)^2 \geq 0\,,
\end{aligned} \tag{21}$$

where the second equality is due to the symmetry of A. Then

$$\begin{aligned}
\langle z \odot x_\varepsilon, [K(x) - L^*L](z \odot x_\varepsilon)\rangle &= \langle z \odot x_\varepsilon, Ax_\varepsilon \odot z\rangle - \langle z \odot x_\varepsilon, A(z \odot x_\varepsilon)\rangle \\
&\quad + \langle z \odot x_\varepsilon, (A(x - x_\varepsilon) + \varepsilon\mathbb{1}) \odot z\rangle\,.
\end{aligned} \tag{22}$$

The difference of the first two terms on the right is non-negative due to (21), the last term on the right of (22) is also non-negative due to the definition of x_ε. $\square$

Proposition 5. *Assume that $L \in \operatorname{Mat}^+$. Then the update*

$$x^{\text{new}} = x - x_\varepsilon + \frac{x_\varepsilon \odot (\varepsilon \mathbb{1} + L^* y)}{L^* L x + \varepsilon \mathbb{1}} \tag{23}$$

maps D into D. Moreover, f is non-increasing under this rule and stationary if and only if x is a local minimum of f.

Proof. Define K as in Proposition 4. Then the corresponding G is an auxiliary function for f. Since G is quadratic in its first argument, the global minimum of $G(\,\cdot\,, x)$ is attained at

$$\operatorname*{arg\,min}_{x' \in \mathbb{R}^d} G(x', x) = x - K^{-1}(x) L^*(Lx - y) \tag{24}$$

where

$$K^{-1}(x) z = \frac{x_\varepsilon}{L^* L x + \varepsilon \mathbb{1}} \odot z \,. \tag{25}$$

Rearranging these expressions, we obtain (23). Moreover, by direct calculation, we verify that the update actually maps into D; by Lemma 2, f is non-increasing under this rule.

Now suppose that $x \in D$ is a stationary point of (23). Then

$$x_\varepsilon \odot (L^* L x - L^* y) = 0 \,, \tag{26}$$

which implies, for each $\alpha \in 1, \ldots, d$, that either $\partial f / \partial x_\alpha = 0$ or, due to the way that $x_\varepsilon = x_\varepsilon(L^* L x, L^* y)$ is defined via (9), $x_\alpha = 0$ and $\partial f / \partial x_\alpha \geq 0$. $\square$

4. Consistency of the regularized FFNMF updates

Returning to the full FFNMF problem, we now consider $F(W, H)$ as function of one argument with the other held fixed. We write

$$F(W, H) = \tfrac{1}{2} \, \|L_1 W - Y\|^2 = \tfrac{1}{2} \, \|L_2 H - Y\|^2 \tag{27}$$

where $L_1 W = L_2 H \equiv \sqrt{M} \odot CWH$ and $Y = \sqrt{M} \odot V$. Using (8), we see that

$$L_1^* Z = C^T (\sqrt{M} \odot Z) H^T \qquad \text{and} \qquad L_2^* Z = W^T C^T (\sqrt{M} \odot Z) \,. \tag{28a}$$

Then, identifying matrices and vectors as described in the introduction to Section 3 and substituting L_1 and L_2 for L into formula (23), we obtain our FFNMF updates (10b) and (10d), respectively. By Proposition 5, F is non-increasing under such updates.

As the problem is not jointly convex in W and H, a conclusion as strong as Proposition 5 does not hold for the joint problem. Now suppose that (W, H) is a stationary point for the FFNMF update (10). Then, we can still assert that the equivalent of (26) holds in each variable, i.e.,

$$W_\varepsilon \odot \nabla_W F(W, H) = 0 \qquad \text{and} \qquad H_\varepsilon \odot \nabla_H F(W, H) = 0 \,. \tag{29}$$

These conditions imply that F is stationary in all its degrees of freedom except possibly for some on the boundary of the admissible region. If there are any such directions in which F is non-stationary, it is increasing along the inward normal to the boundary. For the sake of succinctness and following the implicit trend in the NMF literature we will call any point satisfying these conditions simply a *stationary point* of F. Vice versa, it is easily checked that if F has any stationary point (in this sense), it is invariant under update (10). We summarize this discussion in the following statement.

Theorem 6. *The cost function F is non-increasing under the update rules* (10)*. Moreover, the sequence of updates has a stationary point (W, H) if and only if (W, H) is a stationary point of F.*

A similar conclusion holds if we already know that the algorithm converges.

Theorem 7. *Suppose the sequence of updates* (10) *converges. Then the limit is a stationary point of F.*

Proof. Let (W^n, H^n) denote the sequence of updates under rule (10) with limit point (W, H). Then, due to (11) and (13),

$$W_\varepsilon^n \odot \nabla_W F(W^n, H^n) \to 0 \qquad \text{and} \qquad H_\varepsilon^n \odot \nabla_H F(W^n, H^n) \to 0\,. \tag{30}$$

Passing to the limit yields the stationarity condition (29). □

This result expresses that (10) defines a consistent method. However, it does not guarantee that the sequence of updates converges, nor does it prove that a limit point, if it exists, is a local, much less global minimizer of F. Note that due to convexity of the cost function in each of the arguments, a stationary point (W_0, H_0) is necessarily a partial local minimum in the following sense. There is a neighborhood U of (W_0, H_0) in $\mathrm{Mat}^+ \times \mathrm{Mat}^+$ such that for all $(W, H) \in U$,

$$F(W_0, H) \le F(W, H) \qquad \text{and} \qquad F(W, H_0) \le F(W, H)\,. \tag{31}$$

However, since F is not jointly convex, it is possible that the cost can be further decreased in a neighborhood of a partial local minimum using a simultaneous update of W and H. Although this result is not completely satisfactory, a proof of convergence is not available for any method of this type, and the full NMF problem is considered NP hard [17].

5. Simplified updates for non-degenerate problems

If the FFNMF problem is sufficiently non-degenerate, namely

(i) all elements of M are strictly positive,
(ii) V does not have a zero column or row,
(iii) C does not have a zero column or row, and
(iv) the initial W and H lie strictly in the interior of the admissible region,

then we may use either of the simplified updates

$$W^{\text{new}} = \frac{(\varepsilon\mathbb{1} + B_W) \odot W}{A_W + \varepsilon\mathbb{1}}\,, \qquad H^{\text{new}} = \frac{(\varepsilon\mathbb{1} + B_H) \odot H}{A_H + \varepsilon\mathbb{1}}\,, \tag{32}$$

obtained by replacing $(W_\varepsilon, H_\varepsilon)$ by (W, H) in (13), or

$$W^{\text{new}} = \frac{B_W \odot W}{A_W}\,, \qquad H^{\text{new}} = \frac{B_H \odot H}{A_H}\,, \tag{33}$$

which arises from further setting $\varepsilon = 0$. The non-degeneracy conditions guarantee that the matrices A_W, B_W, A_H, B_H have no zero entries, hence the new iterates W^{new} and H^{new} are well defined and remain strictly inside the admissible region. Similar non-degeneracy conditions appear in [9].

The simplified update rules are covered by the framework Section 3 if we replace K in Proposition 4 by

$$K_1(x)z = \frac{L^*Lx + \varepsilon\mathbb{1}}{x} \odot z \,, \tag{34}$$

respectively

$$K_2(x)z = \frac{L^*Lx}{x} \odot z \,. \tag{35}$$

Noting that

$$\begin{aligned} \langle x \odot z, [K_1(x) - L^*L]x \odot z\rangle &\geq \langle x \odot z, [K_2(x) - L^*L]x \odot z\rangle \\ &= \langle z \odot x, Ax \odot z\rangle - \langle z \odot x, A(z \odot x)\rangle \,, \end{aligned} \tag{36}$$

the proof of Proposition 4 in both cases follows from (21). As in Proposition 5, this yields the updates

$$x^{\text{new}} = \frac{x \odot (\varepsilon\mathbb{1} + L^*Y)}{L^*Lx + \varepsilon\mathbb{1}} \,, \tag{37}$$

respectively

$$x^{\text{new}} = \frac{x \odot L^*Y}{L^*Lx} \,. \tag{38}$$

Plugging L_1 and L_2 into these expressions, we conclude that F is non-increasing under updates (32) or (33).

We remark that (33) with $C = I$ corresponds to the classical NMF algorithm of Lee and Seung [7] when $M = \mathbb{1}$ and the classical multiplicative WNMF updates discussed, for example, in [3, 19].

Whenever the nondegeneracy conditions do not hold, update (33) can fail for two reasons. An obvious problem is a zero denominator which, in practice, is often remedied by an ad hoc modification [14, 16]

$$W^{\text{new}} = \frac{W \odot VH^T}{\varepsilon\mathbb{1} + W^TWH} \,, \qquad H^{\text{new}} = \frac{H \odot W^TV}{\varepsilon\mathbb{1} + W^TWH} \tag{39}$$

which is missing one term from (32). However, there does not appear to be a simple way to prove that the cost function is non-increasing under update (39), whereas (32) is guaranteed to non-increase F.

The second issue concerns all non-regularized multiplicative updates, including (32), (33), and the updates considered in [3, 15, 7, 8, 19]. As is easily seen from (33), whenever $W_{ij} = 0$ for some indices i, j, it will remain zero (i.e., on the boundary of the admissible region) for all consecutive updates; similarly for the matrix H. Thus, even when the cost function is decreasing normal to the boundary, the algorithm cannot ever get away from the boundary. This issue can arise due to rounding errors even if the non-degeneracy condition is satisfied. It is also of significant practical importance since the solutions to the factorization problems are often found on the boundary of the admissible region and boundary solutions are preferred due to their sparsity. Therefore, imposing a strictly positive initialization is undesirable. Furthermore, stationarity of the update only implies

$$W \odot \; \nabla_W F = 0 \qquad \text{and} \qquad H \odot \; \nabla_H F, \tag{40}$$

which does not imply that (W, H) is stationary or increasing in components normal to the boundary—the conclusion of Theorems 6 and 7 does not hold.

In contrast, the FFNMF algorithm (10) robustly handles zero components in the data or in the intermediate solution. In particular, one does not need to assume that all elements of the weight matrix M are positive. If M has zero elements, ordinary WMNF updates can get stuck on a the boundary hyperplane which they are unable to leave.

6. Discussion

In the article we consider a slight, but useful generalization of the classical non-negative matrix factorization problem and study is approximate solution via a regularized gradient descent algorithm. We find that our updates are consistent, the best known result for a large class of simple NMF-type solvers. The algorithm has been successfully used in an application setting for computing very low rank approximations for moderately sized non-negative matrices with about 10^4 elements. In this context, it was found to be trivial to implement, performs adequately fast, and produces results of seemingly good quality even though we do not have proof that it finds optimal or near-optimal solutions.

In practice, the simplified update rule (32) also appears attractive, being slightly faster than (10) and, although we can only prove for the latter that a limit point of the algorithm is a stationary point of the cost functional, appears to perform well and is definitely more robust than (33), the direct generalization of the classical multiplicative NMF solver of Lee and Seung [7].

References

[1] R. Badeau, N. Bertin, and E. Vincent, *Stability analysis of multiplicative update algorithms and application to nonnegative matrix factorization*, IEEE Trans. Neural Networ., 21 (2010), pp. 1869–1881.

[2] S. Comero, D. Servida, L. De Capitani, and B.M. Gawlik, *Using positive matrix factorization (PMF) to characterize an abandoned mine site: principal component analysis comparison and combination with a GIS-based approach*, J. Geochem. Explor., 118 (2012), pp. 30–37.

[3] I.S. Dhillon and S. Sra, *Generalized nonnegative matrix approximations with Bregman divergences*, in Advances in Neural Information Processing Systems 18: 19th Annual Conference on Neural Information Processing Systems, Y. Weiss, ed., Curran Associates, Red Hook, NY, 2005, pp. 283–290.

[4] P.K. Hopke, P. Paatero, H. Jia, R.T. Ross, and R.A. Harshman, *Three-Way (PARAFAC) factor analysis: examination and comparison of alternative computational methods as applied to ill-conditioned data*, Chemom. Intel. Lab. Syst., 43 (1998), pp. 25–42.

[5] P.D. O'Grady and B.A. Pearlmutter, *Discovering speech phones using convolutive non-negative matrix factorisation with a sparseness constraint*, Neurocomputing, 72 (2008), pp. 88–101.

[6] D. Guillamet, M. Bressan, and J. Vitrià, *A weighted nonnegative matrix factorization for local representations*, in Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 1, A. Jacobs and T. Baldwin, eds., IEEE Computer Society, Los Alamitos, CA, 2001, pp. 942–947.

[7] D.D. Lee and H.S. Seung, *Algorithms for non-negative matrix factorization*, in Advances in Neural Information Processing Systems 13, T. K. Leen, T. G. Dietterich, and V. Tresp, eds., MIT Press, Cambridge, MA, 2001, pp. 556–562.

[8] D.D. Lee and H.S. Seung, *Learning the parts of objects by nonnegative matrix factorization*, Nature, 401 (1999), pp. 788–791.

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

[10] C.-J. Lin, *On the convergence of multiplicative update algorithms for nonnegative matrix factorization*, IEEE Trans. Neural Networ., 18 (2007), pp. 1589–1596.

[11] 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.

[12] P. Paatero, U. Tapper, and Järvinen, *Source identification of bulk wet deposition in Finland by positive matrix factorization*, Atmos. Environ., 29 (1995), pp. 1705–1718.
[13] P. Paatero, *Least squares formulation of robust, non-negative factor analysis*, Chemometr. Intell. Lab., 37 (1997), pp. 23–35.
[14] J. Piper, P. Pauca, R. Plemmons, and M. Ginn, *Object characterization from spectral data using nonnegative factorization and information theory*, in Proceedings of AMOS Technical Conference, Curran Associates, Maui, HI, 2004.
[15] S. Sra and I.S. Dhillon, *Nonnegative matrix approximation: algorithms and applications*, Technical Report TR-06-27, Department of Computer Sciences, University of Texas at Austin, 2006.
[16] C. Thurau, *PyMF – Matrix factorization methods for Python*, `http://pymf.googlecode.com`, Version 0.1, 2010.
[17] S.A. Vavasis, *On the complexity of nonnegative matrix factorization*, SIAM J. Optim., 20 (2009), pp. 1364–1377.
[18] Y. Wang and Y. Zhang, *Nonnegative matrix factorization: a comprehensive review*, IEEE Trans. Knowl. Data Eng., 25 (2013), pp. 1336–1353.
[19] S. Zhang, W. Wang, J. Ford, and F. Makedon, *Learning from incomplete ratings using non-negative matrix factorization*, in Proceedings of the 6th SIAM Conference on Data Mining (SDM), J. Ghosh, D. Lambert, D. Skillicorn, and J. Srivastava, eds., SIAM, Bethesda, MD, 2006, pp. 549–553.

E-mail address, M. Oliver: `oliver@member.ams.org`

E-mail address, S. Vasylkevych: `s.vasylkevych@jacobs-university.de`

School of Engineering and Science, Jacobs University, 28759 Bremen, Germany