OPTIMALITY, COMPUTATION, AND INTERPRETATION OF NONNEGATIVE MATRIX FACTORIZATIONS (VERSION: October 18, 2004)

M. CHU*, F. DIELE†, R. PLEMMONS‡, AND S. RAGNI§

Abstract. The notion of low rank approximations arises from many important applications. When the low rank data are further required to comprise nonnegative values only, the approach by nonnegative matrix factorization is particularly appealing. This paper intends to bring about three points. First, the theoretical Kuhn-Tucker optimality condition is described in explicit form. Secondly, a number of numerical techniques, old and new, are suggested for the nonnegative matrix factorization problems. Thirdly, the techniques are employed to two real-world applications to demonstrate the difficulty in interpreting the factorizations.

Key words. linear model, nonnegative matrix factorization, least squares, quadratic programming, Kuhn-Tucker condition, Newton method, ellipsoid method, reduced quadratic model, gradient method, mass balance equation

1. Introduction. Let $Y = [y_{ij}] \in \mathbb{R}^{m\times n}$ denote the matrix of "observed" data where y_{ij} represents, in a broad sense, the *score* obtained by entity j on variable i. One way to characterize the interrelationships among multiple variables that contribute to the observed data Y is to assume that y_{ij} is a linearly weighted score by entity j based on several factors. We shall temporarily assume that there are p factors, but it is precisely the point that the factors are to be retrieved in the mining process. A linear model, therefore, assumes the relationship

$$Y = AF, \tag{1.1}$$

where $A = [a_{ik}] \in \mathbb{R}^{m\times p}$ is a matrix with a_{ik} denoting the *loading* of variable i from factor k or, equivalently, the *influence* of factor k on variable i, and $F = [f_{kj}] \in \mathbb{R}^{p\times n}$ with f_{kj} denoting the *score* of factor k by entity j or the *response* of entity j to factor k. Depending on the applications, there are many ways to interpret the meaning of the linear model.

The *receptor model*, for example, is an observational technique within the air pollution research community which makes use of the ambient data and source profile data to apportion sources or source categories [13, 14, 16]. The fundamental principle in this model is that mass conservation can be assumed and a mass balance analysis can be used to identify and apportion sources of airborne particulate matter in the atmosphere. One approach to obtaining a data set for receptor modelling is to determine a large number of chemical constituents such as elemental concentrations in a number of samples. The relationships between p sources which contribute m chemical species to n samples, therefore, lead to a *mass balance equation*,

$$y_{ij} = \sum_{k=1}^{p} a_{ik} f_{kj}, \tag{1.2}$$

where y_{ij} is the elemental concentration of the ith chemical measured in the jth sample, a_{ik} is the gravimetric concentration of the ith chemical in the kth source, and f_{kj} is the airborne

*Department of Mathematics, North Carolina State University, Raleigh, NC 27695-8205. (chu@math.ncsu.edu)
This research was supported in part by the National Science Foundation under grants DMS-0073056 and CCR-0204157.

†Istituto per le Applicazioni del Calcolo M. Picone, CNR, Via Amendola 122, 70126 Bari, Italy. (f.diele@area.ba.cnr.it)

‡Departments of Computer Science and Mathematics, Wake Forest University, Winston-Salem, NC 27109. (plemmons@wfu.edu) This research was supported in part by the Air Force Office of Scientific Research under grant AFOSR-F49620-02-1-0107 and the Army Research Office under grant DAAD-19-00-1-0540.

§Facoltà di Economia, Università di Bari, Via Camillo Rosalba 56, 70100 Bari, Italy. (irmasr18@area.ba.cnr.it)

mass concentration that the kth source has contributed to the jth sample. In a typical scenario, only values of y_{ij} are observable whereas neither the sources are known nor the compositions of the local particulate emissions are measured. Thus, a critical question is to estimate the number p, the compositions a_{ik}, and the contributions f_{kj} of the sources. Tools that have been employed to analyze the linear model include principal component analysis, factor analysis, cluster analysis, and other multivariate statistical techniques. In this modelling, however, there is a physical constraint imposed upon the data. That is, the source compositions a_{ik} and the source contributions f_{kj} must all be nonnegative. The identification and apportionment, therefore, becomes a nonnegative matrix factorization problem of Y. Classical tools cannot guarantee to maintain the nonnegativity.

As another example, the notion of linear model has been proposed as a way to find a set of basis functions for representing nonnegative data [6, 18, 29]. It is argued that the notion is particularly applicable to image articulation libraries made up of images showing a composite object in many articulations and poses. It is suggested that the factorization would enable the identification and classification of intrinsic "parts" that make up the object being imaged by multiple observations. More specifically, each column $\mathbf{y}_j$ of a nonnegative matrix Y now represents m pixel values of one image. The columns $\mathbf{a}_k$ of A are basis elements in $\mathbb{R}^m$. The columns of F, belonging to $\mathbb{R}^p$, can be thought of as coefficient sequences representing the n images in the basis elements. In other words, the relationship,

$$\mathbf{y}_j = \sum_{k=1}^{p} \mathbf{a}_k f_{kj}, \tag{1.3}$$

can be thought of as that there are *standard parts* $\mathbf{a}_k$ in a variety of positions and that each image $\mathbf{y}_j$ is made by superposing these parts together in some ways. Those parts, being images themselves, are necessarily nonnegative. The superposition coefficients, each part being present of absent, are also necessarily nonnegative.

In either case above and in many other applications [2, 28, 30], we see that the p factors, interpreted as either the sources or the basis elements, play a vital role. In practice, there is a need to determine as fewer factors as possible and, hence, a low rank nonnegative matrix factorization (**NNMF**) of the data matrix Y arises. The mathematical problem can be stated as follows:

(NNMF) *Given a nonnegative matrix $Y \in \mathbb{R}^{m\times n}$ and a positive integer $p < \min\{m, n\}$, find nonnegative matrices $U \in \mathbb{R}^{m\times p}$ and $V \in \mathbb{R}^{p\times n}$ so as to minimize the functional*

$$f(U, V) := \frac{1}{2}\|Y - UV\|_F^2. \tag{1.4}$$

We shall call the product UV of the least squares solution a nonnegative matrix factorization of Y, though Y is not necessarily equal to the product UV. Clearly the product UV is of rank at most p. The objective function (1.4) can be modified in several ways to reflect the application need. For example, a penalty term could be added to $f(U, V)$ in order to enforce sparsity or to enhance smoothness in V [15, 27]. Also, because $UV = (UD)(D^{-1}V)$ for any invertible matrix $D \in \mathbb{R}^{p\times p}$, sometimes it is desirable to "normalize" columns of U. For simplicity, we shall concentrate on (1.4) only in this paper, but the idea certainly can be generalized.

There are quite a few numerical algorithms proposed in the literature for NNMF. See, for example, [15, 19, 20, 27]. Nonetheless, it is mentioned in the survey [33] that the current developments seem to lack a firm theoretical foundation in general. The difficulty lies in the fact that nonnegative matrices form a cone with many facets which make it hard to characterize which and when a facet is active or not in the optimization. Indeed, the article [6] interprets the NNMF

geometrically as the problem of finding a simplicial cone which contains a cloud of data points and which is contained in the positive orthant.

The purpose of the note is threefold: First, we describe a first order optimality condition which can be regarded as the Kuhn-Tucker condition in closed-form. We then discuss general ideas, old and new, for solving the NNMF. Some of the approaches can automatically detect which facet is active along the integration while others have apparent simplicity for computation. The objective of this study is to offer some additional insights into this challenging NNMF problem. Finally, we apply these methods to some real-world problems and demonstrate by comparisons the limitation and difficulty in interpreting the NNMF results.

2. First Order Optimality Condition. The cone $\mathbb{R}_+^{m\times p}$ of nonnegative matrices in $\mathbb{R}^{m\times p}$ can be written as

$$\mathbb{R}_+^{m\times p} = \{E.*E | E\in\mathbb{R}^{m\times p}\}, \tag{2.1}$$

where we have adopted the MATLAB syntax $M.*N = [m_{ij}n_{ij}]$ to denote the Hadamard product of two matrices. In a sense, the expression $E.*E$ is one way to parameterize nonnegative matrices over the open set $\mathbb{R}^{m\times p}$. In this way, the parametrization is differentiable and the problem of nonnegative matrix factorization can now be expressed as the minimization of

$$g(E,F) := \frac{1}{2}\|Y-(E.*E)(F.*F)\|_F^2 \tag{2.2}$$

where variables (E,F) are from the open set $\mathbb{R}^{m\times p}\times\mathbb{R}^{p\times n}$. This parametrization effectively transformed the constrained optimization over the cones into a problem with no constraint at all.

Consider g as a differentiable functional over the space $\mathbb{R}^{m\times p}\times\mathbb{R}^{p\times n}$ with product Frobenius inner product,

$$\langle (X_1,Y_1),(X_2,Y_2)\rangle = \langle X_1,X_2\rangle + \langle Y_1,Y_2\rangle, \tag{2.3}$$

whenever $X_1,X_2\in\mathbb{R}^{m\times p}$ and $Y_1,Y_2\in\mathbb{R}^{p\times n}$. The Fréchet derivative of g therefore can be calculated component by component. In particular, the partial derivative of g with respect to E acting on an arbitrary $H\in\mathbb{R}^{m\times p}$ is given by

$$\begin{aligned}\frac{\partial g}{\partial E}.H &= \langle -(H.*E+E.*H)(F.*F), \delta(E,F)\rangle \\ &= \langle -2H, E.*\left(\delta(E,F)(F.*F)^\top\right)\rangle,\end{aligned} \tag{2.4}$$

where, for convenience, we have adopted the notation

$$\delta(E,F) := Y-(E.*E)(F.*F). \tag{2.5}$$

Similarly, the partial derivative of g with respect to F acting on an arbitrary $K\in\mathbb{R}^{p\times n}$ is given by

$$\frac{\partial g}{\partial F}.K = \langle -2K, F.*\left((E.*E)^\top\delta(E,F)\right)\rangle. \tag{2.6}$$

By the Riesz representation theorem, the gradient of g at (E,F) can be expressed as

$$\nabla g(E,F) = \left(-2E.*\left(\delta(E,F)(F.*F)^\top\right), -2F.*\left((E.*E)^\top\delta(E,F)\right)\right). \tag{2.7}$$

We are now ready to characterize the first order optimality condition for the nonnegative matrix factorization problem as follows:

THEOREM 2.1. *If (E, F) is a local minimizer of the objective functional g in (2.2), then necessarily the equations*

$$E. * \left(\delta(E,F)(F. * F)^\top\right) = 0 \in \mathbb{R}^{m\times p}, \tag{2.8}$$

$$F. * \left((E. * E)^\top \delta(E,F)\right) = 0 \in \mathbb{R}^{p\times n}, \tag{2.9}$$

are satisfied. The corresponding stationary point to the nonnegative matrix factorization problem is given by $U = E. * E$ and $V = F. * F$.

COROLLARY 2.2. *The necessary condition for $(U, V) \in \mathbb{R}_+^{m\times p} \times \mathbb{R}_+^{p\times n}$ to solve the nonnegative matrix factorization problem is*

$$U. * \left((Y - UV)V^\top\right) = 0 \in \mathbb{R}^{m\times p}, \tag{2.10}$$

$$V. * \left(U^\top(Y - UV)\right) = 0 \in \mathbb{R}^{p\times n}, \tag{2.11}$$

It is interesting to note the complementarity condition of zeros in (2.10) and (2.11), that is, if the (i, j) entry of U is not zero, then the corresponding entry in the product $(Y - UV)V^\top$ must be zero, and vise versa. Indeed, (2.10) and (2.11) effectively characterize the Kuhn-Tucker conditions for the minimization of (1.4) subject to the nonnegative constraint [17]. In particular, the following inequalities are also necessary.

COROLLARY 2.3. *The two matrices $-(Y - UV)V^\top$ and $-U^\top(Y - UV)$ are precisely the Lagrangian multipliers specified in the Kuhn-Tucker condition. At a solution (U, V) of the nonnegative matrix factorization problem, it is necessary that*

$$(Y - UV)V^\top \le 0, \tag{2.12}$$

$$U^\top(Y - UV) \le 0. \tag{2.13}$$

3. Numerical Methods. Several existing numerical methods for solving the NNMF have already been reviewed in [20, 33]. Although schemes and approaches are different, any numerical method is essentially centered around satisfying the first order optimality condition. That is, at the local solution either the nonlinear systems (2.8) and (2.9) for general E and F or the systems (2.10) and (2.11) for nonnegative U and V must be satisfied. It should be cautioned, however, that merely satisfying the first order optimality condition is not enough to guarantee that the critical point be a minimizer. Various additional mechanisms, such as the Hessian information or some descent properties, are built into the different schemes to ensure that a critical point is a solution to (1.4).

In this section, we shall study some old and develop some new numerical methods for solving the NNMF problem. We have learned in our experiments that not all methods work equally well. Thus far, there is no absolutely superior method. This presentation that considers the pros and cons of each method therefore might have its merits. We believe that there is much room for further improvement of any of these methods for the NNMF problem.

3.1. Newton-Type Approach. Any critical points of the NNMF problem must satisfy the nonlinear matrix equations (2.8) and (2.9) simultaneously. The Newton method seems to be a reasonable choice of techniques to tackle this problem. We briefly outline three possible variants in this subsection.

3.1.1. Constrained Quasi-Newton Methods. The Kuhn-Tucker conditions form the basis to many nonlinear programming algorithms. The article [11], for example, compares the performance of 27 computer codes which are all designed to solve the general constrained nonlinear optimization problem. Among these, one of the most efficient techniques is the Sequential Quadratic Programming (SQP) methods. The SQP methods solve successively a sequence of quadratic programming subproblems obtained by linearizing the original nonlinear problems at various approximate solutions. One unique feature in the SQP methods is that they accumulate second order information via a quasi-Newton updating procedure. For that reason, the SQP methods are also known as constrained quasi-Newton methods [17]. There are many established results concerning the SQP technique, including its superlinear convergence. An overview of the SQP methods can be found in [8, 9].

We shall not discuss this often very elaborated SQP implementation in this paper. We only point out that when applied to the NNMF problem, the Kuhn-Tucker conditions are explicitly given by (2.10), (2.11), (2.12) and (2.13). We believe that the SQP methods can take further advantage of the underlying structure in a similar way as the ADI Newton method which we outline below.

3.1.2. ADI Newton Iteration. The matrix size involved in the NNMF problem is usually very large. An endeavor tackling the system directly would be quite computationally extensive, if possible at all. A commonly used approach is to alternate between U and V by fixing the other. This idea of alternating direction iteration (ADI) has been used in many applications.

In our situation, we may start by fixing V in (2.10) and solve the system,

$$U. * [B - UC] = 0, \tag{3.1}$$

for a nonnegative matrix $U \in \mathbb{R}_+^{m\times p}$, where both $B = YV^\top \in \mathbb{R}^{m\times p}$ and $C = VV^\top \in \mathbb{R}^{p\times p}$ are fixed and nonnegative matrices. We then fix U and solve next the system

$$V. * [R - SV] = 0, \tag{3.2}$$

for a nonnegative matrix $V \in \mathbb{R}_+^{p\times n}$ with fixed $R = U^\top Y \in \mathbb{R}^{p\times n}$ and $S = U^\top U \in \mathbf{R}^{p\times p}$. We call this one sweep of the *outer loop iteration*. Note that because p is low, the sizes of the square matrices C and S are relatively small.

It is obvious that merely taking $U = BC^{-1}$ is not good enough for (3.1) because U could have negative entries. Extra efforts are needed to satisfy the complementary condition in Corollary 2.2 and the inequalities in Corollary 2.3. Also, it is not clear under what conditions the outer loop iteration will converge.

One structure inherited in the matrix equation (3.1) is that its solution U could be solved row by row. Note that each row of U is not related to any other rows. Each *row* gives rise to a nonlinear system of equations of the form

$$\mathbf{u}^\top . * [\mathbf{b}^\top - \mathbf{u}^\top C] = \mathbf{0}. \tag{3.3}$$

Likewise, if U is fixed, then V in (3.2) can be solved column by column in a similar manner. Though there are m rows for U and n columns for V to be solved, respectively, note that the coefficient matrices involved are either $C = VV^\top$ or $S = U^\top U$. These coefficient matrices need to be updated once per sweep of the outer loop iteration and, more importantly, are of the much smaller size $p \times p$.

To guarantee the nonnegativity of $\mathbf{u}^\top$, we rewrite (3.3) as the equation

$$\psi(\mathbf{e}) = (C(\mathbf{e}. * \mathbf{e}) - \mathbf{b}). * \mathbf{e} = \mathbf{0}, \tag{3.4}$$

by the fact that C is symmetric and by taking $\mathbf{e}.*\mathbf{e}=\mathbf{u}$ with $\mathbf{e}\in\mathbb{R}^p$. It is easy to see that the Frèchet of ψ acting on an arbitrary vector $\mathbf{h}\in\mathbb{R}^p$ can be calculated as

$$\psi'(\mathbf{e}).\mathbf{h}=2(C(\mathbf{e}.*\mathbf{h})).*\mathbf{e}+(C(\mathbf{e}.*\mathbf{e})-\mathbf{b})\,.*\,\mathbf{h}. \tag{3.5}$$

This expression is equivalent to the matrix-vector multiplication

$$\psi'(\mathbf{e}).\mathbf{h}=\{2\mathrm{diag}(\mathbf{e})C\mathrm{diag}(\mathbf{e})+\mathrm{diag}(C(\mathbf{e}.*\mathbf{e})-\mathbf{b})\}\,\mathbf{h}. \tag{3.6}$$

In other words, we have the Jacobian matrix of ψ calculated. A standard Newton iteration scheme can be applied now to solve $\psi(\mathbf{e})=0$ as follows.

ALGORITHM 3.1. *Given* $\mathbf{e}^{(0)}$ *such that* $C(\mathbf{e}^{(0)}.*\mathbf{e}^{(0)})-\mathbf{b}\geq 0$, *do the following for* $k=0,1,\ldots$ *until convergence:*

1. *Compute* $\mathbf{r}^{(k)}=C(\mathbf{e}^{(k)}.*\mathbf{e}^{(k)})-\mathbf{b}$.
2. *Solve for* $\mathbf{h}$ *from the linear system*

$$\left\{2\mathrm{diag}(\mathbf{e}^{(k)})C\mathrm{diag}(\mathbf{e}^{(k)})+\mathrm{diag}(\mathbf{r}^{(k)})\right\}\mathbf{h}=-\mathbf{r}^{(k)}.*\mathbf{e}^{(k)}; \tag{3.7}$$

3. *Update* $\mathbf{e}^{(k+1)}=\mathbf{e}^{(k)}+\alpha^{(k)}\mathbf{h}$.

With $|\mathbf{e}^0|$ large enough, the step size $\alpha^{(k)}$ is adapted so as to maintain $\mathbf{r}^{(k)}\geq 0$ for all k. Obviously, at the convergence, the row vector $\mathbf{u}=\mathbf{e}.*\mathbf{e}$ is a nonnegative solution to (3.3). Repeating the process for each row of U (and indeed these rows can be processed in parallel), we obtain a nonnegative solution U to (2.10) which also satisfies the inequality requirement (2.12), for each fixed V. Exchanging roles of U and V, we can obtain a nonnegative solution V to (2.11), for each fixed U. This completes one sweep of the outer loop iteration.

We stress again that it is not clear under what conditions the outer loop iteration converges. Even if the outer loop converges, we have to point out that the Newton iteration only finds critical points satisfying the first order optimality condition. The iteration does not distinguish a minimizer from a maximizer unless additional second order information is brought in. It is possible that the iteration converges to, for example, a saddle point.

3.1.3. Projected Newton Method. Consider the fact that the objective function (1.4) is separable in columns. For a fixed $U\in\mathbb{R}^{m\times p}$, each single column of V amounts to a least squares minimization for an objective function of the form

$$\phi(\mathbf{v})=\frac{1}{2}\|\mathbf{y}-U\mathbf{v}\|_2^2, \tag{3.8}$$

subject to the constraint that $\mathbf{v}\in\mathbb{R}^p$ is nonnegative. Such a nonnegative least squares problem has been studied extensively in the literature. For example, the MATLAB routine LSQNONNEG using a scheme that essentially is a projected Newton method [22, Chapter 23] is readily available for exactly this type of least squares problem.

Alternating between U and V and employing the projected Newton method or the existing LSQNONNEG for each column of V and each row of U, we now have another numerical method for the NNMF problem.

3.2. Reduced Quadratic Model Approach. In contrast to the Newton-type approach outlined above, the notion of reduced quadratic model approach is considerably simpler to use. Similar to the SQP methods where the original nonlinear programming problem is approximated by a sequence of quadratic programming subproblems, the idea is to replace the quadratic function $\phi(\mathbf{v})$ defined in (3.8) by a sequence of simpler quadratic functions. More specifically, near any given $\mathbf{v}^c$, the quadratic function $\phi(\mathbf{v})$ which can be rewritten as

$$\phi(\mathbf{v})=\phi(\mathbf{v}^c)+(\mathbf{v}-\mathbf{v}^c)^\top\nabla\phi(\mathbf{v}^c)+\frac{1}{2}(\mathbf{v}-\mathbf{v}^c)^\top U^\top U(\mathbf{v}-\mathbf{v}^c) \tag{3.9}$$

is approximated by a *simpler* quadratic model of the form

$$\varphi(\mathbf{v};\mathbf{v}^c) = \phi(\mathbf{v}^c) + (\mathbf{v}-\mathbf{v}^c)^\top \nabla\phi(\mathbf{v}^c) + \frac{1}{2}(\mathbf{v}-\mathbf{v}^c)^\top D(\mathbf{v}^c)(\mathbf{v}-\mathbf{v}^c), \tag{3.10}$$

where $D(\mathbf{v}^c)$ is a diagonal matrix depending on $\mathbf{v}^c$. The minimizer of $\phi(\mathbf{v})$ is approximated by the minimizer $\mathbf{v}^+$ of $\varphi(\mathbf{v};\mathbf{v}^c)$, near which a new quadratic model is created. The definition of $D(\mathbf{v}^{(c)})$ is quite intriguing, which we now describe below.

3.2.1. Lee and Seung Method. Let the entries be noted by $\mathbf{v}^c = [v_i^c] \in \mathbb{R}^p$, $D(\mathbf{v}^c) = \mathrm{diag}\{d_1(\mathbf{v}^c),\ldots,d_p(\mathbf{v}^c)\}$, and so on. First introduced by Lee and Seung [19], one way to define the diagonal entries is by

$$d_i(\mathbf{v}^c) := \frac{(U^\top U\mathbf{v}^c)_i}{v_i^c}, \quad i = 1,\ldots,p. \tag{3.11}$$

Four important consequences follow from this choice of $D(\mathbf{v}^{(c)})$. First, it can be shown that [19]

$$(\mathbf{v}-\mathbf{v}^c)^\top \left(D(\mathbf{v}^c) - U^\top U\right)(\mathbf{v}-\mathbf{v}^c) \geq 0 \tag{3.12}$$

for all $\mathbf{v}$. In other words, the matrix $D(\mathbf{v}^c) - U^\top U$ is positive semi-definite, implying that φ dominates ϕ in the sense that $\phi(\mathbf{v}) \leq \varphi(\mathbf{v};\mathbf{v}^c)$ for all $\mathbf{v}$. Secondly, the minimum of any quadratic function always has a closed form solution, but with $D(\mathbf{v}^c)$ being diagonal the close form solution is easy. In fact, the minimum $\mathbf{v}^+$ of $\varphi(\mathbf{v};\mathbf{v}^c)$ is given by

$$\mathbf{v}^+ := \mathbf{v}^c - D^{-1}(\mathbf{v}^c)(U^\top U\mathbf{v}^c - U^\top\mathbf{y}). \tag{3.13}$$

Thirdly, note from the definition of $D(\mathbf{v}^c)$ that the entries of $\mathbf{v}^+$ are precisely

$$v_i^+ = v_i^c \frac{(U^\top\mathbf{y})_i}{(U^\top U\mathbf{v}^c)_i}, \quad i = 1,\ldots,p. \tag{3.14}$$

and, hence, remain nonnegative if $\mathbf{v}^c$ is nonnegative. Finally, it is important to note that

$$\phi(\mathbf{v}^+) \leq \varphi(\mathbf{v}^+;\mathbf{v}^c) \leq \varphi(\mathbf{v}^c;\mathbf{v}^c) = \phi(\mathbf{v}^c), \tag{3.15}$$

showing that $\mathbf{v}^+$ is an improved update from $\mathbf{v}^c$.

Repeating the above process for each individual column and assembling all columns together, the updated matrix $V^+ = [v_{ij}^+]$ for (1.4) from a given nonnegative matrix $V^c = [v_{ij}^c]$ and a fixed nonnegative matrix U can be defined by the multiplicative rule:

$$v_{ij}^+ := v_{ij}^c \frac{(U^\top Y)_{ij}}{(U^\top UV^c)_{ij}}, \quad i = 1,\ldots,p, \quad j = 1,\ldots,n. \tag{3.16}$$

In terms of the element-by-element multiplication $.*$ and division $./$, the relationship (3.16) can simply be written as

$$V^+ := V^c .* (U^\top Y)./(U^\top UV^c). \tag{3.17}$$

In a similar way, the update $U^+ = [u_{ij}^+]$ for (1.4) from a given nonnegative matrix $U^c = [u_{ij}^c]$ and a fixed nonnegative matrix V can be defined by the rule:

$$U^+ := U^c .* (YV^\top)./(U^cVV^\top). \tag{3.18}$$

Alternating these multiplicative update rules between U and V has been proposed in [19] as means of solving (1.4).

Distinguishing itself from that Newton-type approach, note that the descent property (3.15) of the Lee and Seung method ensures that the objective function $f(U,V)$ is nonincreasing under the update rules.

3.2.2. Ellipsoid Method. The choice of $D(\mathbf{v}^c)$ as is defined in (3.11) deserves further comments. It is clear that there are many other ways to set forth the simpler model (3.10). For example, if all diagonal entries of D are sufficiently large, say, larger than the spectral radius of $U^\top U$, then $D - U^\top U$ is positive definite. Nonetheless, the larger the D, the smaller the D^{-1} and, hence, the less difference between $\mathbf{v}^+$ and $\mathbf{v}^c$ according to (3.13). The challenge thus lies in finding a diagonal matrix D that is large enough to make $D - U^\top U$ positive definite, yet is also small enough to signify the difference between $\mathbf{v}^+$ and $\mathbf{v}^c$. In this section we propose a different quadratic model in the form of (3.10) where the diagonal matrix $D = \mathrm{diag}\{d_1, \ldots, d_k\}$ carries the additional property that its trace is minimized. The notion is based on the semidefinite programming (SDP) technique [34].

We outline the idea by working on the least squares problem (3.8) where $U \in \mathbb{R}^{m\times p}$ is fixed. Denote $S = U^\top U \in \mathbb{R}^{p\times p}$ and $\mathbf{g}^c = [g_i^c] := U^\top U\mathbf{v}^c - U^\top \mathbf{y} \in \mathbb{R}^p$. Let $\lambda_i(D)$ denote the i^{th} eigenvalue of $D - S$. Consider the function

$$\omega(D) := \sum_{i=1}^{p} \ln \frac{1}{\lambda_i(D)} + \sum_{i=1}^{p} \ln \frac{1}{d_i} + \sum_{i=1}^{p} \ln \frac{1}{d_i v_i^c - g_i^c}. \tag{3.19}$$

Because the (real-valued) logarithm is defined only for positive arguments, the function ω can be defined only for diagonal matrices D such that $D - S$ is positive definite, D has positive diagonal entries, and $D\mathbf{v}^c - \mathbf{g}^c$ is a positive vector. The barrier function $\omega(D)$ is introduced because its level curves serve as reasonable approximations to the boundary of the desirable feasible domain.

The following two results give rise to the gradient $\nabla\omega(D)$ and the Hessian $\nabla^2\omega(D)$. The proofs can be found in [4]. More general results can be found in [1, 24].

LEMMA 3.1. *The gradient vector of $\omega(D)$ with $D = \mathrm{diag}\{d_1, \ldots, d_p\}$ is given by*

$$\nabla\omega(D) = \mathrm{diag}\left((D-S)^{-1} - D^{-1}\right) - \begin{bmatrix} \frac{v_1^c}{d_1 v_1^c - g_1^c} \\ \vdots \\ \frac{v_k^c}{d_k v_k^c - g_k^c} \end{bmatrix}. \tag{3.20}$$

LEMMA 3.2. *The Hessian matrix $H(D)$ of $\omega(D)$ is given by*

$$H(D) = (D-S)^{-1}.*(D-S)^{-1} + D^{-1}.*D^{-1} + \mathrm{diag}\left\{\left(\frac{v_1^c}{d_1 v_1^c - g_1^c}\right)^2, \ldots, \left(\frac{v_k^c}{d_k v_k^c - g_k^c}\right)^2\right\}. \tag{3.21}$$

We note from the well known Schur product theorem [12, Theorem 7.5.3] that $H(D)$ is positive definite if D is feasible, which also shows that the function ω is strictly convex over its feasible domain.

Recall that an ellipsoid $\mathcal{E} \subset \mathbb{R}^p$ can best be characterized by its center $\boldsymbol{\gamma} \in \mathbb{R}^p$ and a symmetric and positive definite matrix $\Gamma \in \mathbb{R}^{p\times p}$ in such a way that

$$\mathcal{E} = \mathcal{E}(\Gamma, \boldsymbol{\gamma}) := \left\{\mathbf{x} \in \mathbb{R}^p | (\mathbf{x} - \boldsymbol{\gamma})^T \Gamma^{-1} (\mathbf{x} - \boldsymbol{\gamma}) \le 1\right\}. \tag{3.22}$$

Within the feasible domain of ω, we can approximate its level curves by a sequence of inscribed ellipsoids determined by the Hessians of ω in the following sense due to Dikin [5].

THEOREM 3.3. *Suppose $D^c = \mathrm{diag}(\mathbf{d}^c)$ is a strictly feasible point with respect to (3.19). Then every diagonal matrix $D^+ = \mathrm{diag}(\mathbf{d}^+)$ with $\mathbf{d}^+$ from the ellipsoid $\mathcal{E}(H(D^c)^{-1}, \mathbf{d}^c)$ is also strictly feasible.*

Proof. It has been argued in [4] that D^+ is positive and that $D^+ - S$ is positive definite. It only remains to show that $D^+\mathbf{v}^c - \mathbf{g}^c$ is positive. Denote $\Delta := D^+ - D^c = diag\{\delta_1, \ldots, \delta_p\}$. It follows that

$$D^+\mathbf{v}^c - \mathbf{g}^c = (D^c\mathbf{v}^c - \mathbf{g}^c) + \Delta\mathbf{v}^c.$$

Since

$$\sum_{i=1}^{p} \frac{(\delta_i v_i^c)^2}{(d_i^c v_i^c - g_i^c)^2} < 1,$$

it is clear that $|\delta_i v_i^c| < d_i^c v_i^c - g_i^c$ for all $i = 1, \ldots, p$. □

Given a feasible D^c, any point from the ellipsoid $\mathcal{E}(H(D^c)^{-1}, d^c)$ will carry the four properties that Lee and Seung's choice (3.11) possesses. It is a matter of which point $\mathbf{d}^+$ on $\mathcal{E}(H(D^c)^{-1}, d^c)$ will serve the "goal" better. For instance, in attempting to make D^+ small, one possible objective is to minimize the trace of D^+, i.e.,

$$\text{minimize} \quad \mathbf{1}^\top\mathbf{d}, \tag{3.23}$$

$$\text{subject to} \quad \mathbf{d} \in \mathcal{E}(H(D^c)^{-1}, \mathbf{d}^c) \tag{3.24}$$

where $\mathbf{1} := [1, \ldots, 1]^\top$. Clearly, one can choose to weight the diagonal entries of D differently and end up with different linear objective functional. Such an optimization of linear objective functional over ellipsoids has a closed form solution [10, Page 68].

LEMMA 3.4. *For $\mathbf{p} \neq 0$, the minimal value of the linear functional $\mathbf{p}^\top\mathbf{x}$ subject to the condition $\mathbf{x} \in \mathcal{E}(\Gamma, \boldsymbol{\gamma})$ occurs at*

$$\mathbf{x}^* := \boldsymbol{\gamma} - \frac{1}{\sqrt{\mathbf{p}^T\Gamma\mathbf{p}}}\Gamma\mathbf{p}. \tag{3.25}$$

Using (3.21) and by the fact that p is low, it is extremely easy to be implemented a basic Dikin algorithm as follows:

ALGORITHM 3.2. *(Basic Dikin Method)*
Given $\mathbf{d}^{(0)} \in \mathbb{R}^p$ strictly feasible, do for $k = 0, 1, \ldots$ the following:

1. *If $D^{(k)} - S$ is singular, then stop;*
2. *Otherwise,*
   (a) *Solve $H(D^{(k)})\mathbf{d} = \mathbf{1}$ for $\mathbf{d}$;*
   (b) *Update $\mathbf{d}^{(k+1)} := \mathbf{d}^{(k)} - \frac{1}{\sqrt{\mathbf{1}^\top\mathbf{d}}}\mathbf{d}$.*

Theorem 3.3 guarantees that $\mathbf{d}^{(k)}$ is strictly feasible and hence $D^{(k)} - S$ is never singular in exact arithmetic. However, in floating point arithmetic, one has to settle the singularity (or rank deficiency) of a matrix for an eigenvalue (or a singular value) less than a prescribed tolerance. A usual choice of tolerance for zero is $\epsilon\|S\|$ where ϵ is the machine dependent floating point relative accuracy. For this reason it is possible that the algorithm stops at a point where $D - S$ is *numerically* semi-definite yet trace(D) may have not reached its minimal value. To reduce the risk of hitting the boundary of the feasible domain too soon, we find it is a good idea to start out the Dikin's method from a sufficiently large scalar matrix.

The main difference between the Lee and Seung diagonal matrix $D_{Lee\&Seung}$ defined by (3.11) and the Dikin diagonal matrix D_{Dikin} defined by Algorithm 3.2 is that $D_{Lee\&Seung}$ is *always* on the boundary of the feasible domain because $D_{Lee\&Seung} - S$ has a zero eigenvalue with eigenvector $\mathbf{v}^c$. While the Dikin algorithm produces a diagonal matrix that has minimal trace, the Lee and Seung algorithm is remarkably cheap for computation.

3.3. Gradient Approach.

3.3.1. Gradient Flow. The gradient of the objective function $g(E, F)$ is explicitly known in (2.7). The dynamical system

$$\frac{dE}{dt} = E. * \left(\delta(E, F)(F. * F)^\top\right) \in \mathbb{R}^{m\times p}, \tag{3.26}$$

$$\frac{dF}{dt} = F. * \left((E. * E)^\top \delta(E, F)\right) \in \mathbb{R}^{p\times n}, \tag{3.27}$$

therefore defines a continuous flow that moves in the space $\mathbb{R}^{m\times p} \times \mathbb{R}^{p\times n}$ along the steepest descent direction of the objective functional g. It is easy to check that along the solution flow $(E(t), F(t))$,

$$\begin{aligned}\frac{dg(E(t), F(t))}{dt} = &-\langle\left(\delta(E, F)(F. * F)^\top\right) . * E, \left(\delta(E, F)(F. * F)^\top\right) . * E\rangle \\ &-\langle F. * \left((E. * E)^\top \delta(E, F)\right), F. * \left((E. * E)^\top \delta(E, F)\right)\rangle \le 0.\end{aligned}$$

The objective functional $g(E, F)$ therefore can be used as the Lyapunov function for the dynamical system. Furthermore, because the gradient flow is defined by an analytic vector field, by the well-known Łojasiewicz-Simon theorem [3, 23, 32] the flow converges to a single point of equilibrium. At the limit point, the first order optimality conditions (2.8) and (2.9) are satisfied. Although the global convergence is guaranteed, there might be multiple and separated limit points each of which is a local solution the NNMF problem.

Employing any available ODE solvers to integrate the system (3.26) and (3.27) constitutes another numerical method for solving the NNMF problem.

3.3.2. Steepest Descent Method. Instead of integrating (3.26) and (3.27) by high precision ODE integrators, the Euler method with appropriate step size selection is another way of making use of the gradient information. One way to implement the steepest descent scheme is to update E and F in the following iterations:

$$E^{(k+1)} := E^{(k)} + \mu_k E^{(k)}. * \left(\delta(E^{(k)}, F^{(k)})(F^{(k)}. * F^{(k)})^\top\right), \tag{3.28}$$

$$F^{(k+1)} := F^{(k)} + \mu_k F^{(k)}. * ((F^{(k)}. * F^{(k)})^\top \delta(E^{(k)}, F^{(k)})). \tag{3.29}$$

Recently, Shepherd [31] has proposed an update scheme as follows:

$$U^{(k+1)} = U^{(k+1)}(\mu_k) := \max\left\{0, U^{(k)} + \mu_k(Y - U^{(k)}V^{(k)})(V^{(k)})^\top\right\}, \tag{3.30}$$

$$V^{(k+1)} = V^{(k+1)}(\mu_k) := V^{(k)} + \mu_k(U^{(k)})^\top(Y - U^{(k)}V^{(k)}), \tag{3.31}$$

where max is taken component by component. In either case, the selection of μ_k is critical. In general practice, a backtracking line search using, say, a cubic interpolation and a merit function, is performed to determine the step length μ_k [8, 9]. For the NNMF problem, the selection of step length is easier. For example, the function,

$$\Theta(\mu) := F(U^{(k+1)}(\mu), V^{(k+1)}(\mu)), \tag{3.32}$$

with $U^{(k+1)}(\mu)$ and $V^{(k+1)}(\mu)$ defined by (3.31) is a quartic polynomial in μ. It has been suggested by Shepherd [31] to use the Tartaglia formula to compute directly the roots of $\Theta'(\mu)$ and hence locate the optimal μ.

4. Numerical Experiments. We have outlined three distinct approaches — Newton-type, reduced quadratic model, and gradient — to the NNMF problem. Each approach itself has several variants. Because these methods have different features and wide-ranging degrees of complexities, they perform differently. It is not easy to make a fair comparison of their performance. In this section, we apply the various techniques to two real-world problems. We demonstrate the limits and difficulties in interpreting the factorizations.

Example 1. Consider the 10 irises given in Figure 4.1, each of which is represented by a matrix of size 120×160. These image came from a large database used to test computational iris recognition methods for biometric identification [29]. As grey-scaled images, the entries of these matrices are values between 0 and 1. Form the matrix Y of size 19200×10 by "vectorizing" each iris matrix into a column. The NNMF of Y is meant to seek and identify any intrinsic parts that

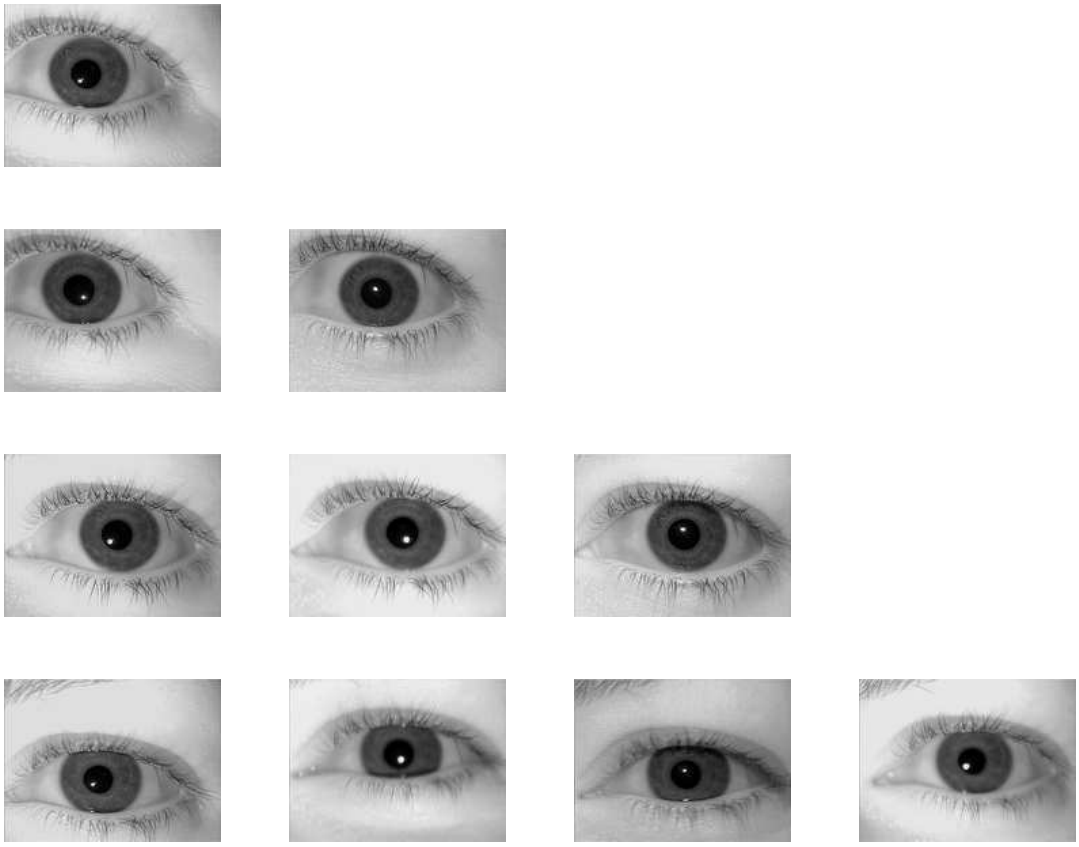

FIG. 4.1. *Intensity image of an iris*

make up these poses. We do not know a priori the number p of parts. We thus experiment with different numbers of p. Once we have found a factorization UV of Y, we normalize columns of U to unit length for uniformity. This can be done via

$$UV = (UD^{-1})(DV))$$

where $D = \text{diag}(\text{diag}(U^\top U))$. Columns of the normalized U will be considered as the bases of these images. The bases for the case $p = 2$ and $p = 4$ are plotted in Figure 4.2 and Figure 4.3, respectively. While Figure 4.2 suggests quite clearly that there are two positions of the irises in the

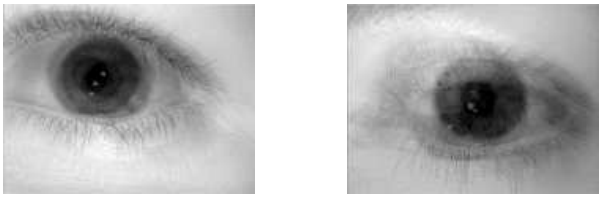

FIG. 4.2. *Basis images for* $p = 2$.

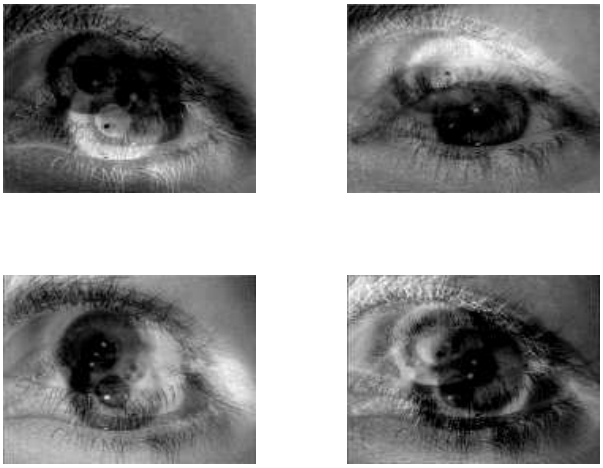

FIG. 4.3. *Basis images for* $p = 4$.

ten images, the somewhat fuzzier Figure 4.3 indicates that there are two basic images overlaying each other. In either case, the basic "parts" that make up these irises remain disappointedly complicated. We note that separation by parts is generally much more effective if a larger database of images is used, as shown by Lee and Seung [18] . We would expect better results would be obtained using a larger database, say 100 or more images, but we used only a small number for computational simplicity.

Example 2. The 8×15 matrix Y in Table 4.2 represents the annual total masses (in thousand short tons) of eight pollutants estimated by the EPA over fifteen years [7]. The blanks at the lower left corner indicate that no data are collected during those years and are assumed zero (and hence bias the analysis). The 4×15 matrix F in Table 4.3 represents the annual total emissions by four principal sectors across the national economy, each of which contains a spectrum of many more pertinent subsectors. The collection of such data often is monitored county by county throughout the USA as a continual task. Details can be found in the report [7] and other EPA publications.

In our first scenario, suppose that both Y and F are available. The problem is to determine

a nonnegative matrix A of size 8×4 that solves the following optimization problem:

$$\begin{aligned} &\text{minimize} && \frac{1}{2}\|Y - AF\|_F^2, \\ &\text{subject to} && A \geq 0, \quad \text{and} \quad \sum_{i=1}^{8} a_{ij} = 1, \quad j = 1, \ldots 4. \end{aligned} \tag{4.1}$$

Each column of A represents the best fitting percentage distribution of pollutants from the emission of the corresponding sector. This is a convex programming problem and the global minimizer is unique.

Using existing software, such as FMINCON in MATLAB, we find that the optimal distribution A_{opt} to Problem (4.1) is given in Table 4.1. This best fitting distribution is in contrast to the average distribution A_{avg} in Table 4.4 that would have to be obtained, otherwise, by extensive efforts in gathering *itemized* pollutant emissions of each sector per year [7]. There are several discrepancies that warrant attention. For example, it is estimated in A_{opt} that 32.70% emissions from the fuel burning contribute to the volatile organic compounds whereas A_{avg} counts only 2.65%. It is estimated in A_{opt} that only 6.31% emissions from the fuel goes to the nitrogen oxides whereas A_{avg} count 27.54%. It is clear that the estimates from A_{opt}, though best fitting the data, is inconsistent with the scientific truth.

	Fuel	Industrial	Transportation	Miscellaneous
Carbon Monoxide	0.1925	0.3400	0.8226	0.0090
Lead	0	0.0000	0	0.0000
Nitrogen Oxides	0.0631	0	0.1503	0.1524
Volatile Organic	0.3270	0.2759	0.0272	0
PM_{10}	0.0000	0.1070	0.0000	0.6198
Sulfur Dioxide	0.4174	0.2771	0.0000	0
$PM_{2.5}$	0.0000	0.0000	0	0.1326
Ammonia	0.0000	0	0	0.0862

TABLE 4.1
Optimal distribution of pollutants from sectors with fixed emission estimates.

In our second scenario, suppose that only Y is available. The problem is to determine four sectors, *not necessarily in any order or any definition*, and their corresponding percentage distributions U and total emissions per year V so as to best fit the observed data Y. This is precisely a NNMF problem.

By using the Lee and Seung algorithm, we obtain local solutions U and V indicated in Table 4.5 and Table 4.6, respectively. We stress that we do not know what each column of U really stand for. It requires a careful interpretation to identify what *factor* is being represented. It is likely that a single column could represent a mixture of two or more known economy sectors. We have noted the improvement in the objective functions, that is,

$$\frac{1}{2}\|Y - UV\|_F^2 = 1.5873 \times 10^7 < \|Y - A_{opt}F\|_F^2 = 2.7017 \times 10^8 < \frac{1}{2}\|Y - A_{avg}F\|^2 = 7.1548 \times 10^8.$$

However, the somewhat unevenness in the NNMF emission estimates per sector given in Table 4.6 seems to make it more difficult to predict the estimate.

Similarly, by using the constrained quasi-Newton method, we obtain another percentage distribution of pollutants from sectors in Table 4.7. (To save space, the corresponding emission estimates are not listed.) This much more sophisticated method is computationally more expensive but is able to find local solutions that give smaller objective values (1.0645×10^7). Again, it is not clear how to identify the sectors and to interpret the distributions of pollutants.

	1970	1975	1980	1985	1989	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999
Carbon Monoxide	129444	116756	117434	117013	106438	99119	101797	99307	99790	103713	94057	101294	101459	96872	97441
Lead	221	160	74	23	5	5	4	4	4	4	4	4	4	4	4
Nitrogen Oxides	20928	22632	24384	23197	23892	24170	24338	24732	25115	25474	25052	26053	26353	26020	25393
Volatile Organic	30982	26080	26336	24428	22513	21052	21249	11862	21100	21682	20919	19464	19732	18614	18145
PM_{10}	13165	7677	7109	41397	40963	27881	27486	27249	27502	28756	25931	25690	25900	26040	23679
Sulfur Dioxide	31161	28011	25906	23658	23294	23678	23045	22814	22475	21875	19188	18859	19366	19491	18867
$PM_{2.5}$						7429	7317	7254	7654	7012	6909	7267	7065	6773	6773
Ammonia						4355	4412	4483	4553	4628	4662	4754	4851	4929	4963

TABLE 4.2

Annual pollutants estimates (in thousand short tons).

	1970	1975	1980	1985	1989	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999
Fuel	41754	40544	43512	41661	40659	39815	39605	40051	38926	38447	36138	36018	35507	34885	34187
Industrial	48222	32364	29615	22389	21909	21120	20900	21102	21438	21467	21190	17469	17988	17868	20460
Transportation	125637	121674	117527	119116	107978	100877	106571	105114	106328	108125	99642	106069	104748	103523	100783
Miscellaneous	10289	6733	10589	46550	46560	45877	42572	40438	41501	45105	39752	43829	46487	42467	39836

TABLE 4.3

Annual emissions estimates (in thousand short tons.

	Fuel	Industrial	Transportation	Miscellaneous
Carbon Monoxide	0.1535	0.3116	0.7667	0.3223
Lead	0.0001	0.0002	0.0002	0
Nitrogen Oxides	0.2754	0.0417	0.1177	0.0113
Volatile Organic	0.0265	0.4314	0.0908	0.0347
PM_{10}	0.0368	0.0768	0.0074	0.4911
Sulfur Dioxide	0.4923	0.0996	0.0112	0.0012
$PM_{2.5}$	0.0148	0.0272	0.0043	0.0761
Ammonia	0.0007	0.0115	0.0016	0.0634

TABLE 4.4

Average distribution of pollutants from sectors.

	Sector 1	Sector 2	Sector 3	Sector 4
Carbon Monoxide	0.2468	0.0002	0.7969	0.0001
Lead	0	0.0008	0	0.0000
Nitrogen Oxides	0.0000	0	0.1641	0.1690
Volatile Organic	0.3281	0.2129	0.0391	0
PM_{10}	0.0000	0.5104	0.0000	0.5532
Sulfur Dioxide	0.4251	0.2757	0.0000	0
$PM_{2.5}$	0.0000	0.0000	0	0.1680
Ammonia	0.0000	0	0	0.1097

TABLE 4.5

NNMF distribution estimates of pollutants from sectors (Lee and Seung algorithm)

.

	1970	1975	1980	1985	1989	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999
Sector 1	58705	57455	57162	3718	4974	47464	46314	47175	47864	43630	44643	42657	43578	42926	43585
Sector 2	25487	11755	7431	81042	75327	10313	10784	8313	6848	12613	4069	3403	3541	3159	1
Sector 3	143614	128945	130225	145512	132349	109442	113118	109881	110295	116521	104440	113926	113910	108437	108828
Sector 4	0	3139	6254	2	4702	40785	39618	41539	43358	40302	43236	43319	43599	44239	42832

TABLE 4.6

NNMF emission estimates (in thousand short tons

(Lee and Seung algorithm).

	Sector 1	Sector 2	Sector 3	Sector 4
Carbon Monoxide	0.3124	0.4468	0.5426	0.6113
Lead	0	0	0.0000	0.0007
Nitrogen Oxides	0.1971	0.1299	0.0366	0.1412
Volatile Organic	0.0239	0.0654	0.1720	0.1191
PM_{10}	0.1936	0.3101	0.0401	0.0220
Sulfur Dioxide	0.0287	0.0477	0.2087	0.1058
$PM_{2.5}$	0.1480	0.0000	0	0
Ammonia	0.0963	0	0.0000	0.0000

TABLE 4.7

NNMF distribution estimates of pollutants from sectors (constrained quasi-Newton method).

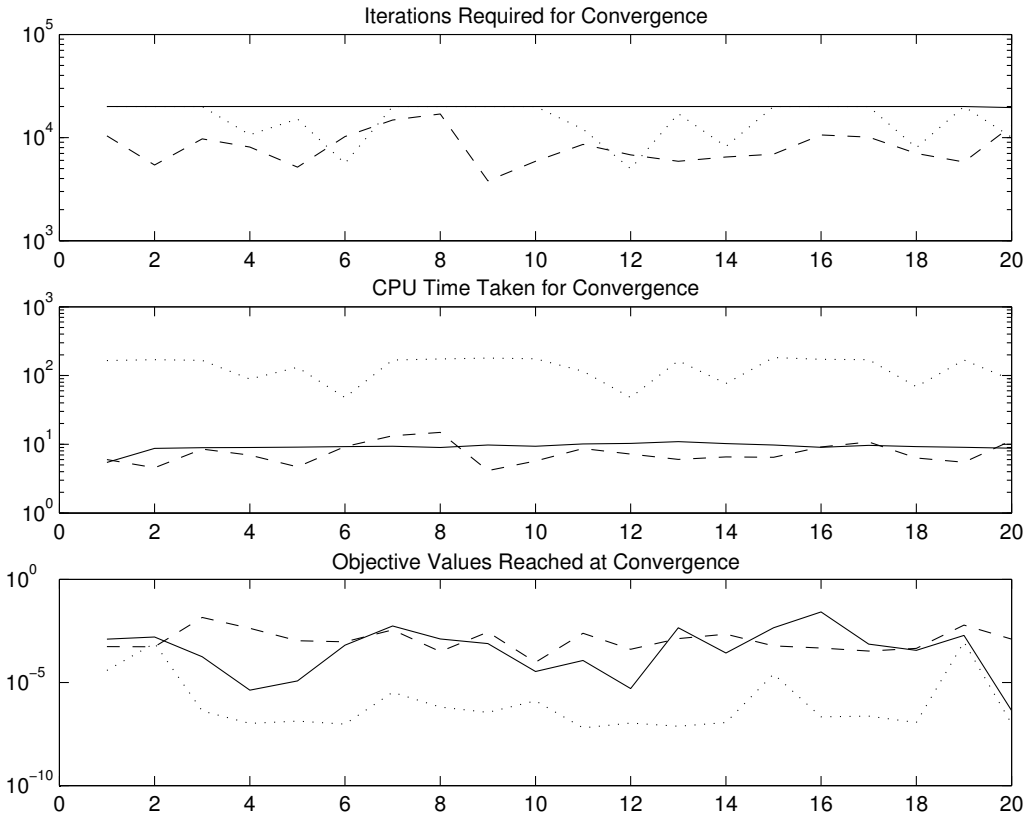

FIG. 4.4. *Performance comparison of Lee and Sueng's reduced quadratic model method (solid), Shepherd quartic line search steepest descent method (dotted), and classical cubic line search steepest descent method (dashed) for the case m=50,n=30,p=5. The x-axis identifies the random test number.*

Example 3. We randomly generate nonnegative matrices $A \in \mathbb{R}^{m\times p}$ and $B \in \mathbb{R}^{p\times n}$ and use the product $Y = AB$ as the test matrix to see if any of the numerical methods can return a factorization U and V such that $Y = UV$. Because some of the methods are considerably more sophisticated than others with features that others do not have, it is difficult to make an across-the-board comparison. We report in Figure 4.4 only the performance of three descent methods, these are, the Lee and Sueng's reduced quadratic model method, the Shepherd's quartic line search steepest descent method, and the classical cubic line search steepest descent method. We choose to compare the number of iterations involved, the CPU time (in seconds) taken, and the objective value reached by each method.

We carry out twenty random tests by using identical stopping criteria. The algorithm is terminated whenever the norm of the gradient is less than 10^{-12} or the number of iterations exceeds 20050. The numbers of iterations, the CPU time, and the objective values are compared at the termination of computation. We notice that in all tests the Lee and Sueng algorithm has used maximal allowable number of iterations without meeting the gradient criterium, yet its objective values are compatible with those of the classical steepest descent algorithm. The Shepherd steepest descent method with quartic line search is most expensive in CPU time, yet it provides much smaller objective values. The classical steepest descent method with cubic line search is most efficient in CPU time, yet its objective values are not as good as those of the quartic line search.

We stress that, though efforts are taken to equalize the complexity of the codes, the comparison are based on implementations that might not have been uniformly optimized. As we can find local solutions only, the limit points produced by these iterations are not necessarily the same even though they start from the same initial value.

5. Conclusion. The nonnegative matrix factorization has been desired by many important applications. We have specified in closed form the first-order optimality condition and suggested a number of numerical procedures that can be employed to obtain a factorization that is at least locally optimal.

Nonetheless, we have demonstrated by two real-world problems that the factorization itself does not necessarily provide immediate interpretation of the real data — the basic parts of the irises are themselves complicated images (and sometimes with overlapped irises); and the percentage distributions of pollutants from economical sectors are not always consistent with data obtained by other means (and could represent mixtures across several sectors.) Proper interpretations or additional constraints on the factors are needed for NNMF applications.

REFERENCES

[1] S. Boyd and L. E. Ghaoui, Method of centers for minimizing generalized eigenvalues, Linear Alg. Appl., 188/189(1993), 63-111.

[2] M. Catral, L. Han, M. Neumann and R. J. Plemmons, On reduced rank nonnegative matrix factorizations for symmetric matrices, to appear in Lin. Alg. and Appl., 2004, http://www.wfu.edu/~plemmons

[3] R. Chill, On the Łojasiewicz-Simon gradient inequality, J. Func. Anal., 201(2003), 572-601.

[4] M. T. Chu and J. W. Wright, Educational testing problem and non-smooth optimization, IMA J. Numer. Anal., 15(1995), 141-160.

[5] I. Dikin, Iterative solution of problems of linear and quadratic programming, Soviet Math. Dokl., 8(1967), 674-675.

[6] D. Donoho and V. Stodden, When does nonnegative matrix factorization give a correct decomposition into parts, Stanford University, 2003, report, available at http://www-stat.stanford.edu/~donoho.

[7] EPA, National air quality and emissions trends report, Office of Air Quality Planning and Standards, EPA, Research Traingle Park, EPA 454/R-01-004, 2001.

[8] R. Fletcher, Practical Methods of Optimization, 2nd edition, John Wiley and Sons, New York, 1987.

[9] P. E. Gill, W. Murray and M. H. Wright, Practical optimization. Academic Press, London-New York, 1981.

[10] M. Grötschel, L. Lovász and A. Schrijver, Geometric Algorithms and Combinatorial Optimization, Springer-Verlag, Berlin, 1988.

[11] W. Hock and K. Schittkowski, A Comparative Performance Evaluation of 27 Nonlinear Programming Codes, Computing, 30(1983), 335-358.

[12] R. A. Horn and C. R. Johnson, Matrix Analysis, Cambridge University Press, New York, 1991.

[13] P. K. Hopke, Receptor Modeling in Environmental Chemistry, Wiley and Sons, New York, 1985.

[14] P. K. Hopke, Receptor Modeling for Air Quality Management, Elsevier, Amsterdam, Hetherlands, 1991.

[15] P. O. Hoyer, Nonnegative sparse coding, Neural Networks for Signal Processing XII, Proc. IEEE Workshop on Neural Networks for Signal Processing, Martigny, 2002.

[16] E. Kim, P. K. Hopke, and E. S. Edgerton, Source identification of Atlanta aerosol by positive matrix factorization, J. Air Waste Manage. Assoc., 53(2003), 731-739.

[17] C. L. Lawson and R. J. Hanson, Solving least squares problems, Classics in Applied Mathematics, 15, SIAM, 1995, Philadelphia.

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

[19] D. D. Lee and H. S. Seung, Algorithms for nonnegative matrix factorization, in Advances in Neural Information Processing 13, MIT Press, 2001, 556-562.

[20] W. Liu and J. Yi, Existing and new algorithms for nonnegative matrix factorization, University of Texas at Austin, 2003, report, available at http://www.cs.utexas.edu/users/liuwg/383CProject/final_report.pdf.

[21] E. Lee, C. K. Chun, and P. Paatero, Application of positive matrix factorization in source apportionment of particulate pollutants, Atmos. Environ., 33(1999), 3201-3212.

[22] C. L. Lawson and R. J. Hanson, Solving Least Squares Problems, Prentice-Hall, 1974.

[23] S. Łojasiewicz, Une propriété topologique des sous-ensembles analytiques réels, in Les Équations aux Dérivées Partielles (Paris, 1962), Vol. 117, 87-89, Éditions du Centre National de la Recherche Scientifique, Paris, 1963.

[24] Yu. E. Nesterov and A. S. Nemirovsky, Interior Point Polynomial Methods in Convex Programming, SIAM Studies in Applied Mathematics, 13, SIAM, Philadelphia, 1994.
[25] P. Paatero and U. Tapper, Least squares formulation of robust nonnegative factor analysis, Chemomet. Intell. Lab. Systems, 37(1997), 23-35.
[26] P. Paatero, User's Guide for Positive Matrix Factorization Programs PMF2 and PMF3, Part 1: Tutorial, ftp://rock.helsinki.fi/pub/misc/pmf.
[27] V. P. Pauca, F. Shahnaz, M. W. Berry, and R. J. Plemmons, Text mining using nonnegative matrix factorizations, preprint, Proc. SIAM Inter. Conf. on Data Mining, Orlando, FL, April 2004..
[28] J. Piper, V. P. Pauca, R. J. Plemmons, and M. Giffin, Object characterization from spectral data using nonnegative factorization and information theory. To appear in the Proc. Amos Technical Conf., Maui, September 2004, http://www.wfu.edu/~plemmons.
[29] R. J. Plemmons, M. Horvath, E. Leonhardt, V. P. Pauca, S. Prasad, S. Robinson, H. Setty, T. Torgersen, J. van der Gracht, E. Dowski, R. Narayanswamy, and P. Silveira, Computational imaging Systems for iris recognition, to appear in Proc. SPIE Annual Meeting, Denver, CO, August 2004, http://www.wfu.edu/~plemmons.
[30] F. Shahnaz, M. Berry, P. Pauca, and R. Plemmons, Document clustering using nonnegative matrix factorization, preprint, Wake Forest University, August 2004, http://www.wfu.edu/~plemmons.
[31] S. J. Shepherd, private communication, http://www.simonshepherd.supanet.com/aa.htm.
[32] L. Simon, Asymptotics for a class of nonlinear evolution equations with applications to geometric problems, Annals Math., 118(1983), 525-571.
[33] J. Tropp, Literature survey: Nonnegative matrix factorization, University of Texas at Asutin, preprint, 2003.
[34] L. Vandenberghe, L. and S. Boyd, Semidefinite programming, SIAM Rev., 38(1996), 49-95.