SIAM J. Matrix Anal. Appl.
Vol. 33, No. 4, pp. 1272–1299

ON TENSORS, SPARSITY, AND NONNEGATIVE FACTORIZATIONS*

ERIC C. CHI† AND TAMARA G. KOLDA‡

Abstract. Tensors have found application in a variety of fields, ranging from chemometrics to signal processing and beyond. In this paper, we consider the problem of multilinear modeling of *sparse count* data. Our goal is to develop a descriptive tensor factorization model of such data, along with appropriate algorithms and theory. To do so, we propose that the random variation is best described via a Poisson distribution, which better describes the zeros observed in the data as compared to the typical assumption of a Gaussian distribution. Under a Poisson assumption, we fit a model to observed data using the negative log-likelihood score. We present a new algorithm for Poisson tensor factorization called CANDECOMP–PARAFAC alternating Poisson regression (CP-APR) that is based on a majorization-minimization approach. It can be shown that CP-APR is a generalization of the Lee–Seung multiplicative updates. We show how to prevent the algorithm from converging to non-KKT points and prove convergence of CP-APR under mild conditions. We also explain how to implement CP-APR for large-scale sparse tensors and present results on several data sets, both real and simulated.

Key words. nonnegative tensor factorization, nonnegative CANDECOMP-PARAFAC, Poisson tensor factorization, Lee–Seung multiplicative updates, majorization-minimization algorithms

AMS subject classifications. 15A69, 65F99, 65C60, 65K99

DOI. 10.1137/110859063

1. Introduction.

Tensors have found application in a variety of fields, ranging from chemometrics to signal processing and beyond. In this paper, we consider the problem of multilinear modeling of *sparse count* data. For instance, we may consider data that encodes the number of papers published by each author at each conference per year for a given time frame [13], the number of packets sent from one IP address to another using a specific port [49], or to/from and term counts on emails [2]. Our goal is to develop a descriptive model of such data, along with appropriate algorithms and theory.

Let $\mathcal{X}$ represent an N-way data tensor of size $I_1 \times I_2 \times \cdots \times I_N$. We are interested in an R-component nonnegative CANDECOMP/PARAFAC [8, 21] factor model

$$\mathcal{M} = \sum_{r=1}^{R} \lambda_r \, \mathbf{a}_r^{(1)} \circ \cdots \circ \mathbf{a}_r^{(N)}, \tag{1.1}$$

where $\circ$ represents outer product and $\mathbf{a}_r^{(n)}$ represents the rth column of the nonnegative *factor matrix* $\mathbf{A}^{(n)}$ of size $I_n \times R$. We refer to each summand as a *component*.

*Received by the editors December 15, 2011; accepted for publication (in revised form) September 5, 2012; published electronically December 4, 2012.
http://www.siam.org/journals/simax/33-4/85906.html

†Department of Human Genetics, University of California, Los Angeles, CA (ecchi@ucla.edu). The work of this author was fully supported by the U.S. Department of Energy Computational Science Graduate Fellowship under grant DE-FG02-97ER25308.

‡Sandia National Laboratories, Livermore, CA (tgkolda@sandia.gov). The work of this author was funded by the applied mathematics program at the U.S. Department of Energy and Sandia National Laboratories, a multiprogram laboratory operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Assuming each factor matrix has been column-normalized to sum to one, we refer to the nonnegative λ_r's as *weights*.

In many applications such as chemometrics [48], we fit the model to the data using a least squares (LS) criteria, implicitly assuming that the random variation in the tensor data follows a Gaussian distribution. In the case of sparse count data, however, the random variation is better described via a Poisson distribution [39, 46], i.e.,

$$x_{\mathbf{i}} \sim \text{Poisson}(m_{\mathbf{i}})$$

rather than $x_{\mathbf{i}} \sim N(m_{\mathbf{i}}, \sigma^2_{\mathbf{i}})$, where the subscript $\mathbf{i}$ is shorthand for the multi-index $(i_1, i_2, \ldots, i_N)$. In fact, a Poisson model is a much better explanation for the zero observations that we encounter in sparse data—these zeros just correspond to events that were very unlikely to be observed. Thus, we propose that rather than using the LS error function given by $\sum_{\mathbf{i}} |x_{\mathbf{i}} - m_{\mathbf{i}}|^2$, for count data we should instead minimize the (generalized) Kullback–Leibler (KL) divergence

$$f(\mathcal{M}) = \sum_{\mathbf{i}} m_{\mathbf{i}} - x_{\mathbf{i}} \log m_{\mathbf{i}}, \tag{1.2}$$

which equals the negative log-likelihood of the observations up to an additive constant. Unfortunately, minimizing KL divergence is more difficult than LS error.

1.1. Contributions. Although other authors have considered fitting tensor data using KL divergence [52, 9, 53], we offer the following contributions:

- We develop the nonnegative CANDECOMP–PARAFAC alternating Poisson regression (CP-APR) model. The subproblems are solved using a majorization-minimization (MM) approach. If the algorithm is restricted to a single inner iteration per subproblem, it reduces to the standard Lee–Seung multiplicative for KL updates [31, 32] as extended to tensors by Welling and Weber [52]. However, using multiple inner iterations is shown to accelerate the method, similar to what has been observed for LS [19].
- It is known that the Lee–Seung multiplicative updates may converge to a non-stationary point [20], and Lin [34] has previously introduced a fix for the LS version of the Lee–Seung method. We introduce a different technique for avoiding *inadmissible zeros* (i.e., zeros that violate stationarity conditions) that is only a trivial change to the basic algorithm and prevents convergence to nonstationary points. This technique is straightforward to adapt to the matrix and/or LS cases as well.
- Assuming the subproblems can be solved exactly, we prove convergence of the CP-APR algorithm. In particular, we can show convergence even for sparse input data and solutions on the boundary of the nonnegative orthant.
- We explain how to efficiently implement CP-APR for large-scale sparse data. Although it is well-known how to do large-scale sparse tensor calculations for the LS fitting function [3], the Poisson likelihood fitting algorithm requires new sparse tensor kernels. To the best of our knowledge, ours is the first implementation of any KL-divergence-based method for large-scale sparse tensors.
- We present experimental results showing the effectiveness of the method on both real and simulated data. In fact, the Poisson assumption leads quite naturally to a generative model for sparse data.

1.2. Related work. Much of the past work in nonnegative matrix and tensor analysis has focused on the LS error [44, 43, 6, 20, 26, 25, 23, 17, 27], which corresponds

to an assumption of normal independently identically distributed noise. The focus of this paper is KL divergence, which corresponds to maximum likelihood estimation under an independent Poisson assumption; see section 2.2. The seminal work in this domain are the papers of Lee and Seung [31, 32], which propose very simple *multiplicative* update formulas for both LS and KL divergence, resulting in a very low cost per iteration. Welling and Weber [52] were the first to generalize the Lee and Seung algorithms to nonnegative tensor factorization (NTF). Applications of NTF based on KL-divergence include EEG analysis [40] and sound source separation [16]. We note that generalizations of KL divergence have also been proposed in the literature, including Bregman divergence [11, 10, 33] and beta divergence [9, 14].

In terms of convergence, Lin [34] and Gillis and Glineur [18] have shown convergence of two different modified versions of the Lee–Seung method for LS. Finesso and Spreij [15] (tensor extension in [53]) have shown convergence of the Lee–Seung method for KL divergence; however, we show later that numerical issues arise if the iterates come near the boundary. This is related to the problems demonstrated by Gonzalez and Zhang [20] that show, in the case of LS loss, that the Lee and Seung method can converge to non-KKT points; we show a similar example for KL divergence in section 6.2.

Our convergence theory is not focused on the Lee–Seung algorithm but rather on a Gauss–Seidel approach. The closest work is that of Lin [35], in which he considers the matrix problem in the LS sense; in the same paper, he dismisses the KL divergence problem as ill-defined but we address that issue in this paper by showing that the convex hull of the level sets of the KL divergence problem is compact.

2. Notation and preliminaries.

2.1. Notation. Throughout, scalars are denoted by lowercase letters (a), vectors by boldface lowercase letters ($\mathbf{v}$), matrices by boldface capital letters ($\mathbf{A}$), and higher-order tensors by boldface Euler script letters ($\mathcal{X}$). We let $\mathbf{e}$ denote the vector of all ones and $\mathbf{E}$ denote the matrix of all ones. The jth column of a matrix $\mathbf{A}$ is denoted by $\mathbf{a}_j$. We use multi-index notation so that a boldface $\mathbf{i}$ represents the index $(i_1, \ldots, i_N)$. We use subscripts to denote iteration index for infinite sequences, and the difference between its use for an entry and its use as an iteration index should be clear by context.

The notation $\|\cdot\|$ refers to the two-norm for vectors or Frobenius norm for matrices, i.e., the sum of the squares of the entries. The notation $\|\cdot\|_1$ refers to the one-norm, i.e., the sum of the absolute values of the entries.

The outer product is denoted by $\circ$. The symbols $*$ and $\oslash$ represent elementwise multiplication and division, respectively. The symbol $\odot$ denotes Khatri–Rao matrix multiplication. The mode-n matricization or unfolding of a tensor $\mathcal{X}$ is denoted by $\mathbf{X}_{(n)}$. See Appendix A for further details on these operations.

2.2. The Poisson distribution and KL divergence. In statistics, count data is often best described as following a Poisson distribution. For a general discussion of the Poisson distribution, see, e.g., [46]. We summarize key facts here.

A random variable X is said to have a Poisson distribution with parameter $\mu > 0$ if it takes integer values $x = 0, 1, 2, \ldots$ with probability

$$P(X = x) = \frac{e^{-\mu}\mu^x}{x!}. \tag{2.1}$$

The mean and variance of X are both μ; therefore, the variance increases along with the mean, which seems like a reasonable assumption for count data. It is also useful

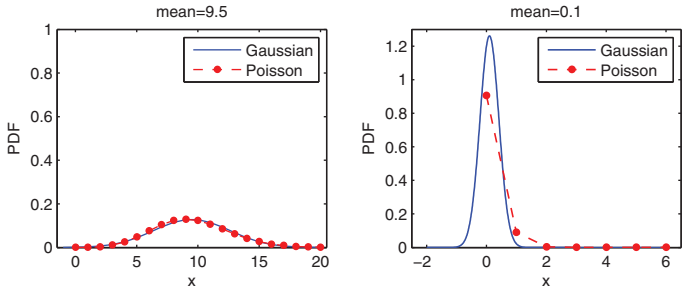


FIG. 2.1. *Illustration of Gaussian and Poisson distributions for two parameters. For both examples, we assume that the variance of the Gaussian is equal to the mean m.*

to note that the sum of independent Poisson random variables is also Poisson. This is important in our case since each Poisson parameter is a multilinear combination of the model parameters. We contrast Poisson and Gaussian distributions in Figure 2.1. Observe that there is good agreement between the distributions for larger values of the mean, μ. For small values of μ, however, the match is not as strong and the Gaussian random variable can take on negative values.

We can determine the optimal Poisson parameters by maximizing the likelihood of the observed data. Let $\mathbf{x}$ be a vector of observations and let $\boldsymbol{\mu}$ be the vector of Poisson parameters. (We assume that μ_i's are not independent; otherwise the function would entirely decouple in the parameters to be estimated.) Then the negative of the log of the likelihood function for (2.1) is the KL divergence

$$\sum_i \mu_i - x_i \log \mu_i, \tag{2.2}$$

excepting the addition of the constant term $\sum_i \log(x_i!)$, which is omitted.

Because we are working with sparse data, there are many instances for which we expect $x_i = 0$, which leads to some ambiguity in (2.2) if $\mu_i = 0$. We assume throughout that $0 \cdot \log(\mu) = 0$ for all $\mu \geq 0$. This is for notational convenience; otherwise, we would write (2.2) as $\sum_i \mu_i - \sum_{i:x_i \neq 0} x_i \log \mu_i$.

3. CP-APR: Alternating Poisson regression. In this section we introduce the CP-APR algorithm for fitting a nonnegative *Poisson tensor decomposition (PTF)* to count data. The algorithm employs an alternating optimization scheme that sequentially optimizes one factor matrix while holding the others fixed; this is nonlinear Gauss–Seidel applied to the PTF problem. The subproblems are solved via an MM algorithm, as described in section 4.

3.1. The optimization problem. Our optimization problem is defined as

$$\min \; f(\mathcal{M}) \equiv \sum_{\mathbf{i}} m_{\mathbf{i}} - x_{\mathbf{i}} \log m_{\mathbf{i}} \quad \text{s.t. } \mathcal{M} = \left[\!\!\left[\boldsymbol{\lambda}; \mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(N)} \right]\!\!\right] \in \Omega, \tag{3.1}$$

$$\text{where} \quad \Omega = \Omega_\lambda \times \Omega_1 \times \cdots \times \Omega_n \quad \text{with}$$
$$\Omega_\lambda = [0, +\infty)^R \quad \text{and} \quad \Omega_n = \left\{ \mathbf{A} \in [0,1]^{I_n \times R} \;\middle|\; \|\mathbf{a}_r\|_1 = 1 \text{ for } r = 1, \ldots, R \right\}. \tag{3.2}$$

Here $\mathcal{M} = [\![\boldsymbol{\lambda}; \mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(N)}]\!]$ is shorthand notation for (1.1) [3]. Depending on context, $\mathcal{M}$ represents the tensor itself or its constituent parts. For example, when

we say $\mathcal{M} \in \Omega$, it means that that the factor matrices have stochasticity constraints on the columns.

The function f is not finite on all of Ω. For example, if there exists $\mathbf{i}$ such that $m_{\mathbf{i}} = 0$ and $x_{\mathbf{i}} > 0$, then $f(\mathcal{M}) = +\infty$. If $m_{\mathbf{i}} > 0$ for all $\mathbf{i}$ such that $x_{\mathbf{i}} > 0$, however, then we are guaranteed that $f(\mathcal{M})$ is finite. Consequently, we will generally wish to restrict ourselves to a domain for which $f(\mathcal{M})$ is finite. We define

$$\Omega(\zeta) \equiv \operatorname{conv}(\{\, \mathcal{M} \in \Omega \mid f(\mathcal{M}) \leq \zeta \,\}), \tag{3.3}$$

where $\operatorname{conv}(\cdot)$ denotes the convex hull. We observe that $\Omega(\zeta) \subset \Omega$ (strict subset) since, for example, the all-zero model is not in $\Omega(\zeta)$. The following lemma states that $\Omega(\zeta)$ is compact for any $\zeta > 0$; the proof is given in Appendix B.

LEMMA 3.1. *Let f be as defined in* (3.1) *and $\Omega(\zeta)$ be as defined in* (3.3)*. For any $\zeta > 0$, $\Omega(\zeta)$ is compact.*

3.2. CP-APR main loop: Nonlinear Gauss–Seidel. We solve problem (3.1) via an alternating approach, holding all factor matrices constant except one. Consider the problem for the nth factor matrix. We note that there is scaling ambiguity that allows us to express the same $\mathcal{M}$ in different ways, i.e.,

$$\mathcal{M} = \left[\!\!\left[\mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(n-1)}, \mathbf{B}^{(n)}, \mathbf{A}^{(n+1)}, \ldots, \mathbf{A}^{(N)} \right]\!\!\right], \tag{3.4}$$

$$\text{where} \quad \mathbf{B}^{(n)} = \mathbf{A}^{(n)}\boldsymbol{\Lambda} \quad \text{and} \quad \boldsymbol{\Lambda} = \operatorname{diag}(\boldsymbol{\lambda}). \tag{3.5}$$

The weights in (3.4) are omitted because they are absorbed into the nth mode. From [3], we can express $\mathcal{M}$ as $\mathbf{M}_{(n)} = \mathbf{B}^{(n)}\boldsymbol{\Pi}^{(n)}$, where $\mathbf{B}^{(n)}$ is defined in (3.5) and

$$\boldsymbol{\Pi}^{(n)} \equiv \left(\mathbf{A}^{(N)} \odot \cdots \odot \mathbf{A}^{(n+1)} \odot \mathbf{A}^{(n-1)} \odot \cdots \odot \mathbf{A}^{(1)} \right)^{\mathsf{T}}. \tag{3.6}$$

Thus, we can rewrite the objective function in (3.1) as

$$f(\mathcal{M}) = \mathbf{e}^{\mathsf{T}} \left[\mathbf{B}^{(n)}\boldsymbol{\Pi}^{(n)} - \mathbf{X}_{(n)} * \log\left(\mathbf{B}^{(n)}\boldsymbol{\Pi}^{(n)} \right) \right] \mathbf{e},$$

where $\mathbf{e}$ is the vector of all ones, $*$ denotes the elementwise product, and the log function is applied elementwise. We note that it is convenient to update $\mathbf{A}^{(n)}$ and $\boldsymbol{\lambda}$ simultaneously since the resulting constraint on $\mathbf{B}^{(n)}$ is simply $\mathbf{B}^{(n)} \geq 0$.

Thus, at each inner iteration of the Gauss–Seidel algorithm, we optimize $f(\mathcal{M})$ restricted to the nth block, i.e.,

$$\mathbf{B}^{(n)} = \arg\min_{\mathbf{B} \geq 0} f_n(\mathbf{B}) \equiv \mathbf{e}^{\mathsf{T}} \left[\mathbf{B}\boldsymbol{\Pi}^{(n)} - \mathbf{X}_{(n)} * \log\left(\mathbf{B}\boldsymbol{\Pi}^{(n)} \right) \right] \mathbf{e}. \tag{3.7}$$

The updates for $\boldsymbol{\lambda}$ and $\mathbf{A}^{(n)}$ come directly from $\mathbf{B}^{(n)}$. Note that some care must be taken if an entire column of $\mathbf{B}^{(n)}$ is zero; if the rth column is zero, then we can set $\lambda_r = 0$ and $\mathbf{b}_r^{(n)}$ to an arbitrary nonnegative vector that sums to one. The full procedure is given in Algorithm 1; this is a variant (because of the handling of $\boldsymbol{\lambda}$) of nonlinear Gauss–Seidel. We note that the scaling and unscaling of the factor matrices are common in alternating algorithms, though not always explicit in the algorithm statement. There are many variations of this basic device; for instance, in the context of the LS version of NTF, [17, Algorithm 2] collects the scaling information into an explicit scaling vector that is "amended" after each inner iteration

We defer the proof of convergence until section 3.3, but we discuss how to check for convergence here. First, we mention an assumption that is important to the theory

ALGORITHM 1. CP-APR algorithm (ideal version).

Let $\mathcal{X}$ be a tensor of size $I_1 \times \cdots \times I_N$. Let $\mathcal{M} = [\![\boldsymbol{\lambda}; \mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(N)}]\!]$ be an initial guess for an R-component model such that $\mathcal{M} \in \Omega(\zeta)$ for some $\zeta > 0$.

repeat
  for $n = 1, \ldots, N$ **do**
    $\boldsymbol{\Pi} \leftarrow \left(\mathbf{A}^{(N)} \odot \cdots \odot \mathbf{A}^{(n+1)} \odot \mathbf{A}^{(n-1)} \odot \cdots \odot \mathbf{A}^{(1)}\right)^{\mathsf{T}}$
    $\mathbf{B} \leftarrow \arg\min_{\mathbf{B} \geq 0} \mathbf{e}^{\mathsf{T}} \left[\mathbf{B}\boldsymbol{\Pi} - \mathbf{X}_{(n)} * \log\left(\mathbf{B}\boldsymbol{\Pi}\right)\right] \mathbf{e}$ ▷ subproblem
    $\boldsymbol{\lambda} \leftarrow \mathbf{e}^{\mathsf{T}}\mathbf{B}$
    $\mathbf{A}^{(n)} \leftarrow \mathbf{B}\boldsymbol{\Lambda}^{-1}$
  end for
until convergence

and also arguably practical. Let

$$\mathcal{S}_i^{(n)} = \left\{ j \;\middle|\; (\mathbf{X}_{(n)})_{ij} > 0 \right\} \tag{3.8}$$

denote the set of indices of columns for which the ith row of $\mathbf{X}_{(n)}$ is nonzero. If $N = 3$, then $\mathbf{X}_{(1)}(i,:)$ corresponds to a vectorization of the ith horizontal slice of $\mathcal{X}$, $\mathbf{X}_{(2)}(i,:)$ to a vectorization of the ith lateral slice, and $\mathbf{X}_{(3)}(i,:)$ to a vectorization of the ith frontal slice. More generally, we can think of vectorizing "hyperslices" with respect to each mode.

Assumption 3.2. *The rows of the submatrix* $\boldsymbol{\Pi}^{(n)}(:, \mathcal{S}_i^{(n)})$ *(i.e., only the columns corresponding to nonzero rows in* $\mathbf{X}_{(n)}$ *are considered) are linearly independent for all* $i = 1, \ldots, I_n$ *and* $n = 1, \ldots, N$.

Assumption 3.2 implies that $|\mathcal{S}_i^{(n)}| \geq R$ for all i. Thus, we need to observe at least $R \cdot \max_n I_n$ counts in the data tensor $\mathcal{X}$, and the counts need to be sufficiently distributed across $\mathcal{X}$. Consequently, the conditions appeal to our intuition that there are concrete limits on how sparse the data tensor can be with respect to how many parameters we wish to fit. If, for example, we had $\mathbf{X}_{(1)}(i,:) = 0$, it is clear that we can remove element i from the first dimension entirely since it contributes nothing. We are making a stronger requirement: each element in each dimension must have at least R nonzeros in its corresponding hyperslice.

A potential problem is that Assumption 3.2 depends on the current iterate, which we cannot predict in advance. However, we observe that if $\boldsymbol{\lambda} > 0$ and the factor matrices have random uniform [0,1] positive entries and $R \leq \min_n \prod_{m \neq n} I_m$, then this condition is satisfied with probability one.[1] This condition can be checked as the iterates progress.

The matrix

$$\boldsymbol{\Phi}^{(n)} \equiv \left[\mathbf{X}_{(n)} \oslash \left(\mathbf{B}^{(n)}\boldsymbol{\Pi}^{(n)}\right)\right] \boldsymbol{\Pi}^{(n)\mathsf{T}}, \tag{3.9}$$

with $\oslash$ denoting elementwise division, will come up repeatedly in the remainder of the paper. For instance, we observe that the partial derivative of f with respect to $\mathbf{A}^{(n)}$ is $\partial f / \partial \mathbf{A}^{(n)} = (\mathbf{E} - \boldsymbol{\Phi}^{(n)})\boldsymbol{\Lambda}$, where $\mathbf{E}$ is the matrix of all ones. Consequently, the matrix $\boldsymbol{\Phi}^{(n)}$ plays a role in checking convergence as follows.

[1] We can actually appeal to a weaker assumption: if the entries are drawn from any distribution that is absolutely continuous with respect to the Lebesgue measure on [0,1], then the condition is satisfied with probability one.

THEOREM 3.3. *If* $\boldsymbol{\lambda} > 0$ *and* $\mathcal{M} = [\![\boldsymbol{\lambda}; \mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(N)}]\!] \in \Omega(\zeta)$ *for some* $\zeta > 0$, *then* $\mathcal{M}$ *is a KKT point of* (3.1) *if and only if*

$$\min\left(\mathbf{A}^{(n)}, \mathbf{E} - \boldsymbol{\Phi}^{(n)}\right) = 0 \text{ for } n = 1, \ldots, N. \tag{3.10}$$

Proof. Since $\boldsymbol{\lambda} > 0$, we can assume that $\boldsymbol{\lambda}$ has been absorbed into $\mathbf{A}^{(m)}$ for some m. Thus, we can replace the constraints $\boldsymbol{\lambda} \in \Omega_\lambda$ and $\mathbf{A}^{(m)} \in \Omega_n$ with $\mathbf{B}^{(m)} \geq 0$. In this case, the partial derivatives are

$$\frac{\partial f}{\partial \mathbf{B}^{(m)}} = \mathbf{E} - \boldsymbol{\Phi}^{(m)} \quad \text{and} \quad \frac{\partial f}{\partial \mathbf{A}^{(n)}} = \left(\mathbf{E} - \boldsymbol{\Phi}^{(n)}\right)\boldsymbol{\Lambda} \text{ for } n \neq m. \tag{3.11}$$

Since $\mathcal{M} \in \Omega(\zeta)$ for some $\zeta > 0$, we know that not all elements of $\mathcal{M}$ are zero; thus, the set of active constraints are linearly independent. The following conditions define a KKT point [42]:

$$\begin{gathered}
\mathbf{E} - \boldsymbol{\Phi}^{(m)} - \boldsymbol{\Upsilon}^{(m)} = 0, \\
(\mathbf{E} - \boldsymbol{\Phi}^{(n)})\boldsymbol{\Lambda} - \boldsymbol{\Upsilon}^{(n)} - \mathbf{e}(\boldsymbol{\eta}^{(n)})^{\mathsf{T}} = 0,\ \mathbf{e}^{\mathsf{T}}\mathbf{A}^{(n)} = 1 \quad \text{for} \quad n \neq m, \\
\mathbf{A}^{(n)} \geq 0,\ \boldsymbol{\Upsilon}^{(n)} \geq 0,\ \boldsymbol{\Upsilon}^{(n)} * \mathbf{A}^{(n)} = 0 \quad \text{for} \quad n \neq m, \\
\mathbf{B}^{(m)} \geq 0,\ \boldsymbol{\Upsilon}^{(m)} \geq 0,\ \boldsymbol{\Upsilon}^{(m)} * \mathbf{B}^{(m)} = 0.
\end{gathered} \tag{3.12}$$

Here $\boldsymbol{\Upsilon}^{(n)}$ are the Lagrange multipliers for the nonnegativity constraints and $\boldsymbol{\eta}^{(n)}$ are the Lagrange multipliers for the stochasticity constraints.

If $\mathcal{M} = \langle \boldsymbol{\lambda}; \mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(N)} \rangle$ is a KKT point, then from (3.12) we have that $\boldsymbol{\Upsilon}^{(m)} = \mathbf{E} - \boldsymbol{\Phi}^{(m)} \geq 0$, $\mathbf{B}^{(m)} \geq 0$, and $\boldsymbol{\Upsilon}^{(m)} * \mathbf{B}^{(m)} = 0$. Thus, $\min(\mathbf{A}^{(m)}\boldsymbol{\Lambda}, \mathbf{E} - \boldsymbol{\Phi}^{(m)}) = 0$. Since $\boldsymbol{\lambda} > 0$ and m is arbitrary, (3.10) follows immediately.

If, on the other hand, (3.10) is satisfied, choosing $\boldsymbol{\Upsilon}^{(m)} = \mathbf{E} - \boldsymbol{\Phi}^{(m)}$ and $\boldsymbol{\Upsilon}^{(n)} = (\mathbf{E} - \boldsymbol{\Phi}^{(n)})\boldsymbol{\Lambda}$ and $\boldsymbol{\eta}^{(n)} = 0$ for $n \neq m$ satisfies the KKT conditions in (3.12). Hence, $\mathcal{M}$ must be a KKT point. ∎

Observe that the condition $\boldsymbol{\lambda} > 0$ makes $\boldsymbol{\lambda}$ moot in the KKT conditions—this reflects the scaling ambiguity that is inherent in the model.

From Theorem 3.3 and because feasibility is always maintained, we can check for convergence by verifying $|\min(\mathbf{A}^{(n)}, \mathbf{E} - \boldsymbol{\Phi}^{(n)})| \leq \tau$ for $n = 1, \ldots, N$, where $\tau > 0$ is some specified convergence tolerance.

3.3. Convergence theory for CP-APR. We require the strict convexity of f in each of the block coordinates. This is ensured under Assumption 3.2.

LEMMA 3.4 (strict convexity of subproblem). *Let* $f_n(\cdot)$ *be the function* f *restricted to the* n*th block as defined in* (3.7). *If Assumption* 3.2 *is satisfied, then* $f_n(\mathbf{B})$ *is strictly convex over* $\mathcal{B}_n = \{\mathbf{B} \in [0, +\infty)^{I_n \times R} : \mathbf{B}\boldsymbol{\Pi}^{(n)} \neq \mathbf{0}\}$.

Proof. In the proof, we drop the n's for convenience. First note that $\mathcal{B}$ is convex. Let $\mathbf{C} = \mathbf{B}^{\mathsf{T}}$. We can rewrite (3.7) as $\min f(\mathbf{C}^{\mathsf{T}}) = \sum_{ij} \mathbf{c}_i^{\mathsf{T}}\boldsymbol{\pi}_j - x_{ij}\log(\mathbf{c}_i^{\mathsf{T}}\boldsymbol{\pi}_j)$ subject to $\mathbf{C} \geq 0$. Hence, it is sufficient to show that the function $\hat{f}(\mathbf{C}) = -\sum_{ij} x_{ij}\log(\mathbf{c}_i^{\mathsf{T}}\boldsymbol{\pi}_j)$ is strictly convex over the convex set $\mathcal{C} = \{\mathbf{C} \in [0, +\infty)^{R \times I_n} : \mathbf{C}^{\mathsf{T}}\boldsymbol{\Pi} \neq \mathbf{0}\}$. Fix $\bar{\mathbf{C}}, \hat{\mathbf{C}} \in \mathcal{C}$ such that $\bar{\mathbf{C}} \neq \hat{\mathbf{C}}$. Since the inner product is affine and log is a strictly concave function, we need only show that there exists some i and j such that $x_{ij} \neq 0$ and $\hat{\mathbf{c}}_i^{\mathsf{T}}\boldsymbol{\pi}_j \neq \bar{\mathbf{c}}_i^{\mathsf{T}}\boldsymbol{\pi}_j$. We know at least one column must differ since $\bar{\mathbf{C}} \neq \hat{\mathbf{C}}$; let i correspond to that column and define $\mathbf{d} = \hat{\mathbf{c}}_i - \bar{\mathbf{c}}_i \neq 0$. By Assumption 3.2, we know that $\boldsymbol{\Pi}(:, S_i)$ has full row rank. Thus, there exists a column j of $\boldsymbol{\Pi}$ such that $x_{ij} \neq 0$ and $\mathbf{d}^{\mathsf{T}}\boldsymbol{\pi}_j \neq 0$. Hence, the claim. ∎

Here we state our main convergence result. Although this result assumes that the subproblems can be solved exactly (which is not the case in practice), it gives some idea as to the convergence behavior of the method. We follow the reasoning of the proof of convergence of nonlinear Gauss–Seidel [5, Proposition 3.9], adapted here for the way that $\boldsymbol{\lambda}$ is handled.

THEOREM 3.5 (convergence of CP-APR). *Suppose that $f(\mathcal{M})$ is strictly convex with respect to each block component and that it is minimized exactly for each block component subproblem of CP-APR. Let $\mathcal{M}_*$ be a limit point of the sequence $\{\mathcal{M}_k\}$ such that $\boldsymbol{\lambda}_* > 0$. Then $\mathcal{M}_*$ is a KKT point of* (3.1).

Proof. Let $\mathcal{M}_k = \langle \boldsymbol{\lambda}_k, \mathbf{A}_k^{(1)}, \ldots, \mathbf{A}_k^{(N)} \rangle$ be the kth iterate produced by the *outer* iterations of Algorithm 1. Define $\mathcal{Z}_k^{(n)}$ to be the nth iterate in the inner loop of outer iteration k with the $\boldsymbol{\lambda}$-vector absorbed into the nth factor, i.e.,

$$\mathcal{Z}_k^{(n)} = \langle \mathbf{A}_{k+1}^{(1)}, \ldots, \mathbf{A}_{k+1}^{(n-1)}, \mathbf{B}_{k+1}^{(n)}, \mathbf{A}_k^{(n+1)}, \ldots, \mathbf{A}_k^{(N)} \rangle,$$

where $\mathbf{B}_{k+1}^{(n)}$ is the solution to the nth subproblem at iteration k. This defines $\mathbf{A}_{k+1}^{(n)}$ to be the column-normalized version of $\mathbf{B}_{k+1}^{(n)}$, i.e., $\mathbf{A}_{k+1}^{(n)} = \mathbf{B}_{k+1}^{(n)}(\operatorname{diag}(\mathbf{B}_{k+1}^{(n)}\mathbf{e}))^{-1}$. Taking advantage of the scaling ambiguity to shift the weights between factors yields

$$\begin{aligned} f(\mathcal{Z}_k^{(n)}) &= f(\langle \mathbf{A}_{k+1}^{(1)}, \ldots, \mathbf{A}_{k+1}^{(n-1)}, \mathbf{A}_{k+1}^{(n)} \operatorname{diag}(\mathbf{B}_{k+1}^{(n)}\mathbf{e}), \mathbf{A}_k^{(n+1)}, \ldots, \mathbf{A}_k^{(N)} \rangle), \\ &= f(\langle \mathbf{A}_{k+1}^{(1)}, \ldots, \mathbf{A}_{k+1}^{(n-1)}, \mathbf{A}_{k+1}^{(n)}, \mathbf{A}_k^{(n+1)} \operatorname{diag}(\mathbf{B}_{k+1}^{(n)}\mathbf{e}), \ldots, \mathbf{A}_k^{(N)} \rangle), \\ &\geq f(\langle \mathbf{A}_{k+1}^{(1)}, \ldots, \mathbf{A}_{k+1}^{(n-1)}, \mathbf{A}_{k+1}^{(n)}, \mathbf{B}_{k+1}^{(n+1)}, \ldots, \mathbf{A}_k^{(N)} \rangle) = f(\mathcal{Z}_k^{(n+1)}). \end{aligned}$$

Observe that $\mathcal{Z}_k^{(N)} = \langle \mathbf{A}_{k+1}^{(1)}, \ldots, \mathbf{A}_{k+1}^{(N-1)}, \mathbf{A}_{k+1}^{(N)} \operatorname{diag}(\boldsymbol{\lambda}_{k+1}) \rangle$, so there is a correspondence between $\mathcal{Z}_k^{(N)}$ and $\mathcal{M}_{k+1}$ such that $f(\mathcal{Z}_k^{(N)}) = f(\mathcal{M}_{k+1})$. For convenience, we define $\mathcal{Z}_k^{(0)} = \langle \mathbf{A}_k^{(1)} \operatorname{diag}(\boldsymbol{\lambda}_k), \mathbf{A}_k^{(2)}, \ldots, \mathbf{A}_k^{(N)} \rangle$. Since we assume the subproblem is solved exactly at each iteration, we have

$$f(\mathcal{M}_k) \geq f(\mathcal{Z}_k^{(1)}) \geq f(\mathcal{Z}_k^{(2)}) \geq \cdots f(\mathcal{Z}_k^{(N-1)}) \geq f(\mathcal{M}_{k+1}) \text{ for all } k. \tag{3.13}$$

Recall that $\Omega(\zeta)$ is compact by Lemma 3.1. Since the sequence $\{\mathcal{M}_k\}$ is contained in the set $\Omega(\zeta)$, it must have a convergent subsequence. We let $\{k_\ell\}$ denote the indices of that convergent subsequence and $\mathcal{M}_* = \langle \boldsymbol{\lambda}_*, \mathbf{A}_*^{(1)}, \ldots, \mathbf{A}_*^{(N)} \rangle$ denote its limit point. By continuity of f, $f(\mathcal{M}_{k_\ell}) \to f(\mathcal{M}_*)$.

We first show that $\|\mathbf{A}_{k_\ell+1}^{(1)} - \mathbf{A}_{k_\ell}^{(1)}\| \to 0$. Assume the contrary, i.e., that it does not converge to zero. Let $\gamma_{k_\ell} = \|\mathcal{Z}_{k_\ell}^{(1)} - \mathcal{Z}_{k_\ell}^{(0)}\|$. By possibly restricting to a subsequence of $\{k_\ell\}$, we may assume there exists some $\gamma_0 > 0$ such that $\gamma(k_\ell) \geq \gamma_0$ for all ℓ. Let $\mathbf{S}_{k_\ell}^{(1)} = (\mathcal{Z}_{k_\ell}^{(1)} - \mathcal{Z}_{k_\ell}^{(0)})/\gamma_{k_\ell}$; then $\mathcal{Z}_{k_\ell}^{(1)} = \mathcal{Z}_{k_\ell}^{(0)} + \gamma_{k_\ell}\mathbf{S}_{k_\ell}^{(1)}$, $\|\mathbf{S}_{k_\ell}^{(1)}\| = 1$, and $\mathbf{S}_{k_\ell}^{(1)}$ differs from zero only along the first block component. Notice that $\{\mathbf{S}_{k_\ell}^{(1)}\}$ belongs to a compact set and therefore has a limit point $\mathbf{S}_*^{(1)}$. By restricting to a further subsequence of $\{k_\ell\}$, we assume that $\mathbf{S}_{k_\ell}^{(1)} \to \mathbf{S}_*^{(1)}$.

Let us fix some $\epsilon \in [0, 1]$. Notice that $0 \leq \epsilon\gamma_0 \leq \gamma_{k_\ell}$. Therefore, $\mathcal{Z}_{k_\ell}^{(0)} + \epsilon\gamma_0\mathbf{S}_{k_\ell}^{(1)}$ lies on the line segment joining $\mathcal{Z}_{k_\ell}^{(0)}$ and $\mathcal{Z}_{k_\ell}^{(0)} + \gamma_{k_\ell}\mathbf{S}_{k_\ell}^{(1)} = \mathcal{Z}_{k_\ell}^{(1)}$ and belongs to $\Omega(\zeta)$ because $\Omega(\zeta)$ is convex. Using the convexity of f with respect to the first block component and the fact that $\mathcal{Z}_{k_\ell}^{(1)}$ minimizes f over all $\mathcal{Z}$ that differ from $\mathcal{Z}_{k_\ell}^{(1)}$ in the first block component, we obtain

$$f(\mathcal{Z}_{k_\ell}^{(1)}) = f(\mathcal{Z}_{k_\ell}^{(0)} + \gamma_{k_\ell}\mathbf{S}_{k_\ell}^{(1)}) \leq f(\mathcal{Z}_{k_\ell}^{(0)} + \epsilon\gamma_0\mathbf{S}_{k_\ell}^{(1)}) \leq f(\mathcal{Z}_{k_\ell}^{(0)}).$$

Since $f(\mathcal{Z}_{k_\ell}^{(0)}) = f(\mathcal{M}_{k_\ell}) \to f(\mathcal{M}_*)$, (3.13) shows that $f(\mathcal{Z}_{k_\ell}^{(1)})$ also converges to $f(\mathcal{M}_*)$. Taking limits as ℓ tends to infinity, we obtain

$$f(\mathcal{M}_*) \leq f(\mathcal{Z}_*^{(0)} + \epsilon\gamma_0 \mathbf{S}_*^{(1)}) \leq f(\mathcal{M}_*),$$

where $\mathcal{Z}_*^{(0)}$ is just $\mathcal{M}_*$ with $\boldsymbol{\lambda}_*$ absorbed into the first component. We conclude that $f(\mathcal{M}_*) = f(\mathcal{Z}_*^{(0)} + \epsilon\gamma_0 \mathbf{S}_*^{(1)})$ for every $\epsilon \in [0,1]$. Since $\gamma_0 \mathbf{S}_*^{(1)} \neq 0$, this contradicts the strict convexity of f as a function of the first block component. This contradiction establishes that $\|\mathbf{A}_{k_\ell+1}^{(1)} - \mathbf{A}_{k_\ell}^{(1)}\| \to 0$. In particular, $\mathcal{Z}_{k_\ell}^{(1)}$ converges to $\mathcal{Z}_*^{(0)}$.

By definition of $\mathcal{Z}_{k_\ell}^{(1)}$ and the assumption that each subproblem is solved exactly, we have

$$f(\mathcal{Z}_{k_\ell}^{(1)}) \leq f(\langle \mathbf{B}, \mathbf{A}_{k_\ell}^{(2)}, \ldots, \mathbf{A}_{k_\ell}^{(N)} \rangle) \text{ for all } \mathbf{B} \geq 0.$$

Taking limits as $\ell \to \infty$, we obtain

$$f(\mathcal{M}_*) \leq f(\langle \mathbf{B}, \mathbf{A}_*^{(2)}, \ldots, \mathbf{A}_*^{(N)} \rangle) \text{ for all } \mathbf{B} \geq 0.$$

In other words, $\mathbf{B}_*^{(1)} = \mathbf{A}_*^{(1)} \operatorname{diag}(\boldsymbol{\lambda}_*)$ is the minimizer of f with respect to the first block components with the remaining components are fixed at $\mathbf{A}_*^{(2)}$ through $\mathbf{A}_*^{(N)}$. From the KKT conditions [42], we have that

$$\mathbf{B}_*^{(1)} \geq 0, \quad \frac{\partial f}{\partial \mathbf{B}^{(1)}}(\mathbf{B}_*^{(1)}) \geq 0, \quad \mathbf{B}_*^{(1)} * \frac{\partial f}{\partial \mathbf{B}^{(1)}}(\mathbf{B}_*^{(1)}) = 0.$$

In turn, since $\boldsymbol{\lambda}_* > 0$, we have $\min(\mathbf{A}_*^{(1)}, \mathbf{E} - \boldsymbol{\Phi}_*^{(1)}) = 0$.

Repeating the previous argument shows that $\|\mathbf{A}_{k_\ell+1}^{(2)} - \mathbf{A}_{k_\ell}^{(2)}\| \to 0$ and that $\min(\mathbf{A}_*^{(2)}, \mathbf{E} - \boldsymbol{\Phi}_*^{(2)}) = 0$. Continuing inductively, $\min(\mathbf{A}_*^{(n)}, \mathbf{E} - \boldsymbol{\Phi}_*^{(n)}) = 0$ for $n = 1, \ldots, N$. Thus, by Theorem 3.3, $\mathcal{M}_*$ is a KKT point of $f(\mathcal{M})$. ∎

Before proceeding to the discussion solving the subproblem, we point out that remarkably very little is assumed about the objective function f in Theorem 3.5. The proof required that f is differentiable, strictly convex in each of its block components, and there is a $\xi > 0$ such that the level set $\Omega(\xi)$ is compact. The upshot is that Theorem 3.5 applies equally well to other choices of f corresponding to other members in the family of beta distributions that are convex, namely, the divergences that correspond to $\beta \in [1,2]$ [14]. In fact, it was also observed in [17] that "rescaling does not interfere with the convergence of the Gauss–Seidel iterations" (in the context of the LS formulation of NTF).

4. Solving the CP-APR subproblem via MM. The basic idea of an MM algorithm is to convert a hard optimization problem (e.g., nonconvex and/or nondifferentiable) into a series of simpler ones (e.g., smooth convex) that are easy to minimize and that majorize the original function, as follows.

DEFINITION 4.1. *Let f and g be real-valued functions on $\mathbb{R}^n$ and $\mathbb{R}^n \times \mathbb{R}^n$, respectively. We say that g* majorizes *f at $\mathbf{x} \in \mathbb{R}^n$ if $g(\mathbf{y}, \mathbf{x}) \geq f(\mathbf{y})$ for all $\mathbf{y} \in \mathbb{R}^n$ and $g(\mathbf{x}, \mathbf{x}) = f(\mathbf{x})$.*

If $f(\mathbf{x})$ is the function to be optimized and $g(\cdot, \mathbf{x})$ majorizes f at $\mathbf{x}$, the basic MM iteration is $\mathbf{x}_{k+1} = \arg\min_{\mathbf{y}} g(\mathbf{y}, \mathbf{x}_k)$. It is easy to see that such iterates always take nonincreasing steps with respect to f since $f(\mathbf{x}_{k+1}) \leq g(\mathbf{x}_{k+1}, \mathbf{x}_k) \leq g(\mathbf{x}_k, \mathbf{x}_k) = f(\mathbf{x}_k)$, where $\mathbf{x}_k$ is the current iterate and $\mathbf{x}_{k+1}$ is the optimum computed at that iterate.

ALGORITHM 2. Subproblem solver for Algorithm 1.

1: $\mathbf{B} \leftarrow \mathbf{A}^{(n)}\mathbf{\Lambda}$
2: **repeat** ▷ subproblem loop
3: $\quad\mathbf{\Phi} \leftarrow \left(\mathbf{X}_{(n)} \oslash (\mathbf{B\Pi})\right)\mathbf{\Pi}^{\mathsf{T}}$
4: $\quad\mathbf{B} \leftarrow \mathbf{B} * \mathbf{\Phi}$
5: **until** convergence

Consider the nth subproblem in (3.7). Here we drop the n's for convenience so that (3.7) reduces to

$$\min_{\mathbf{B}\geq 0} f(\mathbf{B}) \equiv \mathbf{e}^{\mathsf{T}}\left[\mathbf{B\Pi} - \mathbf{X} * \log(\mathbf{B\Pi})\right]\mathbf{e}. \tag{4.1}$$

Recall that $\mathbf{X}$ is the nonnegative data tensor reshaped to a matrix of size $I \times J$, $\mathbf{\Pi}$ is a nonnegative matrix of size $R \times J$ with rows that sum to 1, and $\mathbf{B}$ is a nonnegative matrix of size $I \times R$. For clarity in the ensuing discussion, we also restate Assumption 3.2 in terms of the local variables for this section as follows.

Assumption 4.2. *The rows of the submatrix* $\mathbf{\Pi}(:, \{\, j \mid \mathbf{X}_{ij} > 0 \,\})$ *(i.e., only the columns corresponding to nonzero rows in* $\mathbf{X}$ *are considered) are linearly independent for all* $i = 1, \ldots, I$.

According to Assumption 4.2, for every i there is at least one j such that $x_{ij} > 0$. Thus, we can assume that we have $\bar{\mathbf{B}} \geq 0$ such that $f(\bar{\mathbf{B}})$ is finite. We now introduce the majorization used in our subproblem solver. This majorization is also a special case of the one derived in [14] when $\beta = 1$ and has a long history in image reconstruction that predates its use in NMF [38, 47, 30]. The objective f is majorized at $\bar{\mathbf{B}}$ by the function

$$g(\mathbf{B}, \bar{\mathbf{B}}) = \sum_{rij}\left[b_{ir}\pi_{rj} - \alpha_{rij}x_{ij}\log\left(\frac{b_{ir}\pi_{rj}}{\alpha_{rij}}\right)\right], \quad \text{where} \quad \alpha_{rij} = \frac{\bar{b}_{ir}\pi_{rj}}{\sum_r \bar{b}_{ir}\pi_{rj}}. \tag{4.2}$$

The proof of this fact is straightforward and thus relegated to Appendix C. The advantage of this majorization is that the problem is now completely separable in terms of b_{ir}, i.e., the individual entries of $\mathbf{B}$. Moreover, $g(\cdot, \bar{\mathbf{B}})$ has a unique global minimum with an analytic expression, given by $\mathbf{B} * \mathbf{\Phi}$, where $\mathbf{\Phi}$ is as defined in (3.9) and depends on $\mathbf{B}$. A proof is provided in Appendix C. The MM algorithm iterations are then defined by

$$\mathbf{B}_{k+1} = \psi(\mathbf{B}_k) \equiv \mathbf{B}_k * \mathbf{\Phi}(\mathbf{B}_k), \quad \text{where} \quad \mathbf{\Phi}(\mathbf{B}_k) = \left[\mathbf{X} \oslash (\mathbf{B}_k\mathbf{\Pi})\right]\mathbf{\Pi}, \tag{4.3}$$

and $\mathbf{X}$ and $\mathbf{\Pi}$ come from (4.1). If $\mathbf{B}_0 \geq 0$, clearly $\mathbf{B}_k \geq 0$ for all k. Observe that $\nabla f(\mathbf{B}) = \mathbf{E} - \mathbf{\Phi}(\mathbf{B})$. We discuss in section 5.3 how to exploit this simple relationship to quickly compute stopping rules for the algorithm. The MM algorithm to solve the Gauss–Seidel subproblem of line 4 in Algorithm 1 is given in Algorithm 2.

The monotonic decrease in objective function does not guarantee that the MM iterates will converge to the desired global minimizer of the subproblem. Nonetheless, the following theorem shows that under mild conditions on the starting point $\mathbf{B}_0$ (discussed further in section 5.2), the MM iterates will converge to the unique global minimum of (4.1). The proof follows the reasoning of the convergence proof of an algorithm for fitting a regularized Poisson regression problem given in [28] and is given in Appendix D.

THEOREM 4.3 (convergence of MM algorithm). *Let f be as defined in* (4.1) *and assume Assumption* 4.2 *holds, let $\mathbf{B}_0$ be a nonnegative matrix such that $f(\mathbf{B}_0)$ is finite and $(\mathbf{B}_0)_{ir} > 0$ for all (i,r) such that $(\mathbf{\Phi}(\mathbf{B}_*))_{ir} > 1$, and let the sequence $\{\mathbf{B}_k\}$ be defined as in* (4.3). *Then $\{\mathbf{B}_k\}$ converges to the global minimizer of f.*

Note that we make a modest but very useful generalization of existing results by allowing iterates to be on (or very close to) the boundary. Prior convergence results, including [28, 15, 53], assume that all iterates are strictly positive. Though true in exact arithmetic, in numerical computations it is not uncommon for some iterates to become zero numerically. In section 5.2, we show how to ensure the condition on $\mathbf{B}_0$ holds in practice.

5. CP-APR implementation details. The previous algorithms omit many details and numerical checks that are needed in any practical implementation. Thus, Algorithm 3 provides a detailed version that can be directly implemented. A highlight of this implementation is the "inadmissible zero" avoidance, which fixes the problem of getting stuck at a zero value with multiplicative updates.

5.1. Lee–Seung is a special case of CP-APR. If we take only one iteration of the subproblem loop (i.e., setting $\ell_{\max} = 1$), then CP-APR is the Lee–Seung multiplicative update algorithm for the KL divergence. Thus, we can view the Lee–Seung algorithm as a special case of our algorithm where we do not solve the subproblems exactly; quite the contrary, we take only one step toward the subproblem solution.

5.2. Inadmissible zero avoidance. A well-known problem with multiplicative updates is that some elements may get "stuck" at zero; see, e.g., [20]. For example, if $a_{ir}^{(n)} = 0$, then the multiplicative updates will never change it. In many cases, a zero entry may be the correct answer, so we want to allow it. In other cases, though, the zero entry may be incorrect in the sense that it does not satisfy the KKT conditions, i.e., $a_{ir}^{(n)} = 0$ but $1 - \Phi_{ir}^{(n)} < 0$. We refer to these values as *inadmissible zeros*. We correct this problem before we enter into the multiplicative update phase of the algorithm. In lines 4 thru 5 of Algorithm 3, any inadmissible zeros (or near-zeros) are "scooched" away from zero and into the interior. The amount of the scooch is controlled by the user-defined parameter κ. The condition in Theorem 4.3 is exactly that the starting point should not have any zeros that are ultimately inadmissible. If we discover that a sequence of iterates leads to an inadmissible zero (or almost-zero), we restart the method by restarting the method with a new starting point. This adjustment prevents convergence to non-KKT points. Note that all the quantities needed to perform the check are precomputed and that there is no change to the algorithm besides adjusting a few zero entries in the current factor matrix. The fix for the inadmissible zeros is compatible with the Lee–Seung algorithm for LS error as well.

Lin [34] has made a similar observation in the LS case and applied changes to his gradient descent version of the Lee–Seung method. Our correction is different and is directly incorporated into the multiplicative update scheme rather than requiring a different update formula. Gillis and Glineur [18] proposed a more drastic fix by restricting the factor matrices to have entries in $[\epsilon, \infty)$ for some small positive ϵ. Avoiding all zeros clearly rules out the possibility of getting stuck at an inadmissible zero, but does so at the expense of eliminating any hope of obtaining sparse factor matrices, a desirable property in many applications.

5.3. Practical considerations on convergence. The convergence conditions on the subproblem require that $\min(\mathbf{B}^{(n)}, \mathbf{E} - \mathbf{\Phi}^{(n)}) = 0$. We do not require the value

ALGORITHM 3. Detailed CP-APR algorithm.

Let $\mathcal{X}$ be a tensor of size $I_1 \times \cdots \times I_N$. Let $\mathcal{M} = \langle \boldsymbol{\lambda}; \mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(N)} \rangle$ be an initial guess for an R-component model such that $\mathcal{M} \in \Omega(\zeta)$ for some $\zeta > 0$.
Choose the following parameters:

- $k_{\max}$ = Maximum number of outer iterations
- $\ell_{\max}$ = Maximum number of inner iterations (per outer iteration)
- τ = Convergence tolerance on KKT conditions (e.g., 10^{-4})
- κ = Inadmissible zero avoidance adjustment (e.g., 0.01)
- κ_{tol} = Tolerance for identifying a potential inadmissible zero (e.g., 10^{-10})
- ϵ = Minimum divisor to prevent divide-by-zero (e.g., 10^{-10})

for $k = 1, 2, \ldots, k_{\max}$ **do**
  `isConverged` $\leftarrow$ true
  for $n = 1, \ldots, N$ **do**
    $\mathbf{S}(i,r) \leftarrow \begin{cases} \kappa, & \text{if } k > 1, \mathbf{A}^{(n)}(i,r) < \kappa_{\text{tol}}, \text{ and } \boldsymbol{\Phi}^{(n)}(i,r) > 1, \\ 0, & \text{otherwise} \end{cases}$
    $\mathbf{B} \leftarrow (\mathbf{A}^{(n)} + \mathbf{S})\boldsymbol{\Lambda}$
    $\boldsymbol{\Pi} \leftarrow \left(\mathbf{A}^{(N)} \odot \cdots \odot \mathbf{A}^{(n+1)} \odot \mathbf{A}^{(n-1)} \odot \cdots \odot \mathbf{A}^{(1)}\right)^{\mathsf{T}}$
    for $\ell = 1, 2, \ldots, \ell_{\max}$ **do** ▷ subproblem loop
      $\boldsymbol{\Phi}^{(n)} \leftarrow \left(\mathbf{X}_{(n)} \oslash (\max(\mathbf{B}\boldsymbol{\Pi}, \epsilon))\right)\boldsymbol{\Pi}^{\mathsf{T}}$
      if $|\min(\mathbf{B}, \mathbf{E} - \boldsymbol{\Phi}^{(n)})| < \tau$ **then**
        break
      end if
      `isConverged` $\leftarrow$ false
      $\mathbf{B} \leftarrow \mathbf{B} * \boldsymbol{\Phi}^{(n)}$
    end for
    $\boldsymbol{\lambda} \leftarrow \mathbf{e}^{\mathsf{T}}\mathbf{B}$
    $\mathbf{A}^{(n)} \leftarrow \mathbf{B}\boldsymbol{\Lambda}^{-1}$
  end for
  if `isConverged` = true **then**
    break
  end if
end for

to be exactly zero but instead check that it is smaller in magnitude than the user-defined parameter τ. We break out of the subproblem loop as soon as this condition is satisfied.

From Theorem 3.3, we can check for overall convergence by verifying (3.10). We do not want to calculate this at the end of every n-loop because it is expensive. Instead, we know that the iterates will stop changing once we have converged and so we can validate the convergence of all factor matrices by checking that no factor matrix has been modified and every subproblem has converged.

5.4. Sparse tensor implementation. Consider a large-scale sparse tensor that is too large to be stored as a dense tensor requiring $\prod_n I_n$ memory. In this case, we can store the tensor as a sparse tensor as described in [3], requiring only $(N+1) \cdot \text{nnz}(\mathcal{X})$ memory.

The elementwise division in the update of $\boldsymbol{\Phi}$ requires that we divide the tensor (in matricized form) $\mathbf{X}$ by the current model estimate (in matricized form) $\mathbf{M} = \mathbf{B}\boldsymbol{\Pi}$.

Unfortunately, we cannot afford to store $\mathbf{M}$ explicitly as a dense tensor because it is the same size as $\mathcal{X}$. In fact, we generally cannot even form $\mathbf{\Pi}$ explicitly because it requires almost as much storage as $\mathcal{M}$. We observe, however, that we need only calculate the values of $\mathbf{M}$ that correspond to nonzeros in $\mathbf{X}$.

Let $P = \text{nnz}(\mathcal{X})$. Then we can store the sparse tensor $\mathcal{X}$ as a set of values and multi-indices, $(v^{(p)}, \mathbf{i}^{(p)})$ for $p = 1, \ldots, P$. In order to avoid forming the current model estimate, $\mathcal{M}$, as a dense object, we will store only selected rows of $\mathbf{\Pi}$, one per nonzero in $\mathcal{X}$; we denote these rows by $\mathbf{w}^{(p)}$ for $p = 1, \ldots, P$. The pth vector is given by the elementwise product of rows of the factor matrices, i.e.,

$$\mathbf{w}^{(p)} = \mathbf{A}^{(1)}(i_1^{(p)}, :) * \cdots * \mathbf{A}^{(n-1)}(i_{n-1}^{(p)}, :) * \mathbf{A}^{(n+1)}(i_{n+1}^{(p)}, :) * \cdots * \mathbf{A}^{(N)}(i_N^{(p)}, :).$$

In order to determine $\hat{\mathcal{X}} = \mathcal{X} \oslash \mathcal{M}$ in the calculation of $\mathbf{\Phi}$, we proceed as follows. The tensor $\hat{\mathcal{X}}$ will have the same nonzero pattern as $\mathcal{X}$, and we let $\hat{v}^{(p)}$ denote its values. It can be determined that

$$\hat{v}^{(p)} = x^{(p)} / \left\langle \mathbf{w}^{(p)}, \mathbf{A}^{(n)}(i_n^{(p)}, :) \right\rangle.$$

To calculate $\mathbf{\Phi} = \hat{\mathbf{X}}\mathbf{\Pi}$, we simply have

$$\mathbf{\Phi}(i', r) = \sum_{p : i_n^{(p)} = i'} \hat{v}^{(p)} \mathbf{w}^{(p)}(r).$$

The storage of the $\mathbf{w}^{(p)}$ for $p = 1, \ldots, P$ vectors and the entries $\hat{v}^{(p)}$ requires $(R+1)P$ additional storage.

6. Numerical results for CP-APR.

6.1. Comparison of objective functions for sparse count data. We contend that for sparse count data, KL divergence (1.2) is a better objective function. To support our claim, we consider simulated data where we know the correct answer. Specifically, we consider a three-way tensor ($N = 3$) of size $1000 \times 800 \times 600$ and $R = 10$ factors. It will be generated from a model $\mathcal{M} = [\![\boldsymbol{\lambda}; \mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(N)}]\!]$. The entries of the vector $\boldsymbol{\lambda}$ are selected uniformly at random from $[0, 1]$. Each factor matrix $\mathbf{A}^{(n)}$ is generated as follows: (1) For each column in $\mathbf{A}^{(n)}$, randomly select 10% (i.e., $1/R$) of the entries uniformly at random from the interval $[0, 100]$. (2) The remaining entries are selected uniformly at random from $[0, 1]$. (3) Each column is scaled so that its 1-norm is 1 (i.e., its sum is 1). An "observed" tensor can be thought of as the outcome of tossing $\nu \ll \prod I_n$ balls into $\prod I_n$ empty urns, where each entry of the tensor corresponds to an urn. For each ball, we first draw a factor r with probability $\lambda_r / \sum \lambda_r$. The indices (i, j, k) are selected randomly proportional to $\mathbf{a}_r^{(n)}$ for $n = 1, 2, 3$. In other words, the ball is then tossed into the (i, j, k)th urn with probability $a_{ir}^{(1)} a_{jr}^{(2)} a_{kr}^{(3)}$. In this manner, the balls are allocated across the urns independently of each other. This procedure generates entries $x_{\mathbf{i}}$ that are each distributed as Poisson($m_{\mathbf{i}}$). We adjust the final $\boldsymbol{\lambda}$ so that the scale matches that of $\mathcal{X}$, i.e., $\boldsymbol{\lambda} \leftarrow \nu\boldsymbol{\lambda}/\|\boldsymbol{\lambda}\|$. We generate problems where the number of observations ranges from 480,000 (0.1%) down to 24,000 (0.005%). Recall that Assumption 3.2 implies that the absolute minimum number of observations is $R \cdot \max_n I_n = 10,000$. We have used very few observations, as real problems do indeed tend to be this sparse.

Table 6.1 shows comparisons of four methods. The first two are optimizing LS: Lee–Seung for LS and alternating LS with no nonnegativity constraints (CP-ALS).

TABLE 6.1
Accuracy comparison (mean of 10 trials) using the FMS and the number of columns correctly identified in the first factor matrix.

	LS				KL Divergence			
	Lee–Seung LS		CP-ALS		Lee–Seung KL		CP-APR	
Observations	FMS	#Cols	FMS	#Cols	FMS	#Cols	FMS	#Cols
480000 (0.100%)	0.58	6.4	0.71	7.3	0.89	8.7	0.96	9.5
240000 (0.050%)	0.51	5.4	0.72	7.4	0.83	8.2	0.91	9.2
48000 (0.010%)	0.37	3.8	0.59	6.3	0.76	7.5	0.80	7.9
24000 (0.005%)	0.33	3.5	0.51	5.7	0.72	6.6	0.74	6.9

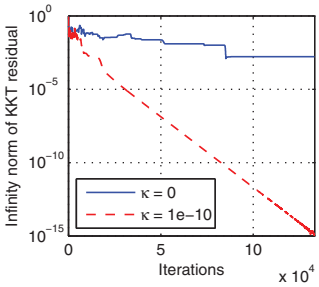


FIG. 6.1. *Lee–Seung permitting inadmissible zeros (blue solid line) and avoiding inadmissible zeros (red dashed line).*

The last two are optimizing KL divergence: Lee–Seung for KL divergence and our method (CP-APR). We have also tested the modified Lee–Seung method of Finesso and Spreij [15, 53], but it is only a scaled version of the Lee–Seung method for KL divergence and gave nearly identical results, which are omitted. All implementations are from version 2.5 of Tensor Toolbox for MATLAB [4, 3, 2]; exact parameter settings are provided in Appendix E. We report the factor match score (FMS), a measure in $[0, 1]$ of how close the computed solution is to the true solution. A value of 1 is ideal. Since the FMS measure is somewhat abstract, we also report the number of columns in the first factor matrix such that the cosine of the angle between the true solution and the computed solution is greater than 0.95. A value of 10 is ideal since we have used $R = 10$. The reported values are averages over 10 problems. See Appendix E for precise formulas for both measures. Although these problems are extremely sparse, all methods are able to correctly identify components in the data. Overall, the methods optimizing KL divergence are superior to those optimizing LS. We also observe that CP-APR is an improvement compared to Lee–Seung KL; we provide later evidence that this improvement is more likely due to the inadmissible zero fix than the extra inner iterations (which provide a benefit of enhanced speed rather than accuracy).

6.2. Fixing misconvergence of Lee–Seung. We demonstrate the effectiveness of our simple fix for avoiding inadmissible zeros, as described in section 5.2. Our technique is based on the same observation on inadmissible zeros as in Lin [34], but the change to the algorithm is different. As in [20], we consider fitting a rank-10 bilinear model for a 25×15 dense positive matrix with entries drawn independently and uniformly from $[0, 1]$. We apply CP-APR using $\ell_{\max} = 1, \tau = 10^{-15}, \epsilon = 0, \kappa_{\text{tol}} = 100 \cdot \epsilon_{\text{mach}}$. We do two runs: one with $\kappa = 0$, corresponding to the standard Lee–Seung (KL version) algorithm, and the other with $\kappa = 10^{-10}$ to move away from inadmissible zeros. In both runs we use the same strictly positive initial guess. Figure 6.1 shows

TABLE 6.2
CP-APR with different values of $\ell_{\max}$ for sparse count data over 100 trials.

(a) FMS

$\ell_{\max}$	1	5	10
Median	0.9858	0.9858	0.9862
Mean	0.9483	0.9514	0.9603

(b) Number of multiplicative updates

$\ell_{\max}$	1	5	10
Median	9819	7655	7290
Mean	16370	11710	11660

(c) Time (seconds)

$\ell_{\max}$	1	5	10
Median	168.70	68.98	55.00
Mean	299.60	106.10	87.92

the magnitude of the KKT residual over more than 10^5 iterations. When $\kappa > 0$, the sequence clearly convergences. On the other hand, when $\kappa = 0$, the iterates appear to get stuck at a non-KKT point. Closer inspection of the factor matrix iterates reveals a single offending inadmissible zero, i.e., its partial derivative is -0.0016 but should be nonnegative. Hence, we use positive values of κ in our experiments.

6.3. The benefit of extra inner iterations. We show that increasing the maximum number of inner iterations $\ell_{\max}$ can accelerate the convergence in Table 6.2. Recall that $\ell_{\max} = 1$ corresponds to the Lee–Seung algorithm [31, 52]. We consider a three-way tensor ($N = 3$) of size $500 \times 400 \times 300$ and $R = 5$ factors. We generate 100 problem instances from 100 randomly generated models $\mathcal{M} = [\![\boldsymbol{\lambda}; \mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(N)}]\!]$ as described in section 6.1 with 0.1% observations. We compare CP-APR with $\ell_{\max} = 1, 5$, and 10, and the other parameters are set as $k_{\max} = 10^6$, $\tau = 10^{-4}$, $\kappa = 10^{-8}$, $\kappa_{\text{tol}} = 100 \cdot \epsilon_{\text{mach}}$, $\epsilon = 0$. We track both the number of multiplicative updates (line 8 of Algorithm 3) and the CPU time using the MATLAB command `cputime`. The experiments were performed on an iMac computer with a 3.4-GHz Intel Core i7 processor and 8 GB of RAM. Table 6.2(a) reports the FMS scores as we vary $\ell_{\max}$, and we observe that the value of $\ell_{\max}$ does not significantly impact accuracy. However, we observe that increasing $\ell_{\max}$ can decrease the overall work and runtime. Tables 6.2(b) and 6.2(c) present the average number of multiplicative updates and total runtimes, respectively. The distribution of updates and times was highly skewed as some problems required a substantial number of iterations. Nonetheless, we see a monotonic decrease in the number of updates and time as $\ell_{\max}$ increases. The differences are more substantial when comparing wall clock time. The reason for the disproportionate decrease in wall clock time compared to the tally of updates is that the cost of the calculation of $\boldsymbol{\Pi}$ (in line 6 of Algorithm 3) is amortized over all the subproblem iterations.

6.4. Enron data. We consider the application of CP-APR to email data from the infamous Federal Energy Regulatory Commission investigation of Enron Corporation. We use the version of the data set prepared by Zhou et al. [56] and further processed by Perry and Wolfe [45], which includes detailed profiles on the employees. The data is arranged as a three-way tensor $\mathcal{X}$ arranged as sender $\times$ receiver $\times$ month, where entry (i, j, k) indicates the number of messages from employee i to employee j in month k. The original data set had 38,388 messages (technically, there were only 21,635 messages but some messages were sent to multiple recipients and so are counted multiple times) exchanged between 156 employees over 44 months (November 1998–June 2002). We preprocessed the data, removing months that had fewer than 300 messages and removing any employees that did not send and receive an average of at least

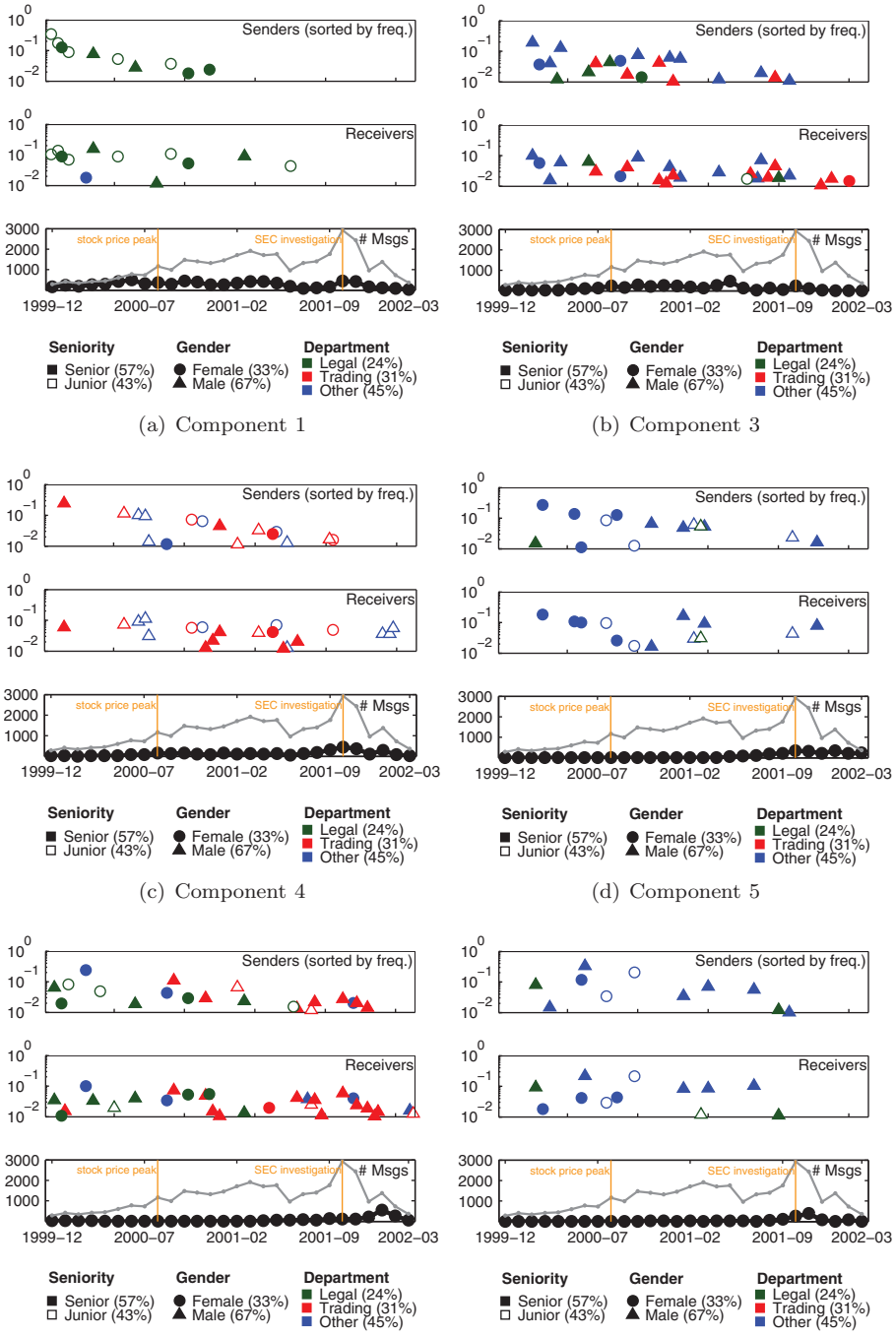
one message per month. Ultimately, our data set spanned 28 months (December 1999–March 2002) and involved 105 employees and a total of 33,079 messages. The data is arranged so that the senders are sorted by frequency (greatest to least). The tensor representation has a total of 8,540 nonzeros. (Many of the messages occur between the same sender-receiver pair in the same time period.) The tensor is 2.7% dense.

We apply CP-APR to find a model for the data. There is no ideal method for choosing the number of components. Typically, this value is selected through trial and error, trading off accuracy (as the number of components grows) and model simplicity. Here we show results for $R = 10$ components. We use the same settings for CP-APR as specified in Appendix E.

Figure 6.2 illustrates six components in the resulting factorization; the other four are shown in Appendix F. For each component, the top two plots shows the activity of senders and receivers, with the employees ordered from left to right by frequency of sending emails. Each employee has a symbol indicating their seniority (junior or senior), gender (male or female), and department (legal, trading, other). The sender and receiver factors have been normalized to sum to one, so the height of the marker indicates each employee's relative activity within the component. The third component (in the time dimension) is scaled so that it indicates total message volume explained by that component. The light gray line shows the total message volume. It is interesting to observe how the components break down into specific subgroups. For instance, component 1 in Figure 6.2(a) consists of nearly all "legal" and is majority female. This can be contrasted to component 5 in Figure 6.2(d), which is nearly all "other" and also majority female. Component 3 in Figure 6.2(b) is a conversation among "senior" staff and mostly male; on the other hand, "junior" staff are more prominent in Component 4 in Figure 6.2(c). Component 8 in Figure 6.2(e) seems to be a conversation among "senior" staff after the SEC investigation has begun. Component 10 in Figure 6.2(f) indicates that a couple of "legal" staff are communicating with many "other" staff immediately after the SEC investigation is announced, perhaps advising the "other" staff on appropriate responses to investigators.

6.5. SIAM data. As another example, we consider 5 years (1999–2004) of SIAM publication metadata that has previously been used by Dunlavy et al. [12]. Here, we build a three-way sparse tensor based on title terms (ignoring common stop words), authors, and journals. The author names have been normalized to last name plus initial(s). The resulting tensor is of size 4,952 (terms) × 6,955 (authors) × 11 (journals) and has 64,133 nonzeros (0.017% dense). The highest count is 17 for the triad ("education," "Schnabel B," "SIAM Rev."), which is a result of Prof. Schnabel's writing brief introductions to the education column for *SIAM Review*. In fact, the next four highest counts correspond to the terms "problems," "review," "survey," and "techniques" and to authors "Flaherty J" and "Trefethen N."

Computing a 10-component factorization yields the results shown in Table 6.3. We use the same settings for CP-APR as specified in Appendix E. In the table, for the term and author modes, we list any entry whose factor score is greater than $10^{-7} \cdot I_n$, where I_n is the size of the nth mode; in the journal mode, we list any entry greater than 0.01. The tenth component corresponds to introductions written by section editors for *SIAM Review*. The first component shows that there is overlap in both authors and title keyword between *SIAM J. Computing* and *SIAM J. Discrete Math*. The second and third components have some overlap in topic and two overlapping authors, but different journals. Both components 8 and 9 correspond to the same journal but reveal two subgroups of authors writing on slightly different topics.



(a) Component 1 (b) Component 3

(c) Component 4 (d) Component 5

(e) Component 8 (f) Component 10

FIG. 6.2. *Components from factorizing the Enron data.*

TABLE 6.3
Highest-scoring items in a 10*-term factorization of the term* × *author* × *journal tensor from five years of SIAM publication data.*

#	Terms	Authors	Journals
1	graphs, problem, algorithms, approximation, algorithm, complexity, optimal, trees, problems, bounds	Kao MY, Peleg D, Motwani R, Cole R, Devroye L, Goldberg LA, Buhrman H, Makino K, He X, Even G	SIAM J Comput, SIAM J Discrete Math
2	method, equations, methods, problems, numerical, multigrid, finite, element, solution, systems	Chan TF, Saad Y, Golub GH, Vassilevski PS, Manteuffel TA, Tuma M, Mccormick SF, Russo G, Puppo G, Benzi M	SIAM J Sci Comput
3	finite, methods, equations, method, element, problems, numerical, error, analysis, equation	Du Q, Shen J, Ainsworth M, Mccormick SF, Wang JP, Manteuffel TA, Schwab C, Ewing RE, Widlund OB, Babuska I	SIAM J Numer Anal
4	control, systems, optimal, problems, stochastic, linear, nonlinear, stabilization, equations, equation	Zhou XY, Kushner HJ, Kunisch K, Ito K, Tang SJ, Raymond JP, Ulbrich S, Borkar VS, Altman E, Budhiraja A	SIAM J Control Optim
5	equations, solutions, problem, equation, boundary, nonlinear, system, stability, model, systems	Wei JC, Chen XF, Frid H, Yang T, Krauskopf B, Hohage T, Seo JK, Krylov NV, Nishihara K, Friedman A	SIAM J Math Anal
6	matrices, matrix, problems, systems, algorithm, linear, method, symmetric, problem, sparse	Higham NJ, Guo CH, Tisseur F, Zhang ZY, Johnson CR, Lin WW, Mehrmann V, Gu M, Zha HY, Golub GH	SIAM J Matrix Anal Appl
7	optimization, problems, programming, methods, method, algorithm, nonlinear, point, semidefinite, convergence	Qi LQ, Tseng P, Roos C, Sun DF, Kunisch K, Ng KF, Jeyakumar V, Qi HD, Fukushima M, Kojima M	SIAM J Optim
8	model, nonlinear, equations, solutions, dynamics, waves, diffusion, system, analysis, phase	Venakides S, Knessl C, Sherratt JA, Ermentrout GB, Scherzer O, Haider MA, Kaper TJ, Ward MJ, Tier C, Warne DP	SIAM J Appl Math
9	equations, flow, model, problem, theory, asymptotic, models, method, analysis, singular	Klar A, Ammari H, Wegener R, Schuss Z, Stevens A, Velazquez JJL, Miura RM, Movchan AB, Fannjiang A, Ryzhik L	SIAM J Appl Math
10	education, introduction, health, analysis, problems, matrix, method, methods, control, programming	Flaherty J, Trefethen N, Schnabel B, [None], Moon G, Shor PW, Babuska IM, Sauter SA, Van Dooren P, Adjei S	SIAM Rev

7. Conclusions and future work. We have developed an alternating Poisson regression fitting algorithm, CP-APR, for PTF for sparse count data. When such data is generated via a Poisson process, we show that methods based on KL divergence such as CP-APR recover the true CP model more reliably than methods based on LS. Indeed, in classical statistics, it is well-known that the randomness observed in sparse count data is better explained and analyzed by the Poisson model (KL divergence) than a Gaussian one (LS error).

Our algorithm can be considered an extension of the Lee–Seung method for KL divergence with multiple inner iterations (similar to [19] for LS). Allowing for multiple inner iterations has the benefit of accelerating convergence. Moreover, being very

similar to an existing method, CP-APR is simple to implement with the exception of some details of the sparse implementation as described in section 5.4. To the best of our knowledge, ours is the first implementation of any KL-divergence-based method for large-scale sparse tensors.

In section 3.3, we provide a general-purpose convergence proof for the alternating Gauss–Seidel approach. The regularity conditions imposed in our proofs make rigorous and concrete our intuition that in the context of sparse count data, CP-APR will converge provided that the data tensor meets a minimal density and that nonzeros are sufficiently spread throughout the data tensor with respect to the size of the factor matrices being fit. Any subproblem solver can be substituted for the MM method without changing the theory. A benefit of the MM subproblem solver is that its multiplier matrix can be used to explicitly track convergence based on the KKT conditions. Moreover, we observe that we can use the KKT information to identify and correct inadmissible zeros using a scooch. Lin [34] had a similar observation in the LS case but came up with a different correction technique. We analyze convergence of the MM subproblem with the scooch in order to show that it will always converge. Our results are stronger than past results because they allow iterates with some zero entries. Even though zero entries are possible to avoid in exact arithmetic, they often occur in numerical computations and so are important to consider.

There remains much room for future work. Foremost among practical considerations is speed of convergence. Although multiplicative updates are relatively simple to compute, CP-APR can require many iterates. One approach to accelerating convergence would be to replace the MM algorithm subproblem solver. For example, Kim, Sra, and Dhillon [24] present fast quasi-Newton methods for minimizing box-constrained convex functions that can be used to solve a nonnegative LS or minimum KL-divergence subproblem in a nonlinear Gauss–Seidel solver. A second approach is to focus on the sequence of outer iterates. Zhou, Alexander, and Lange [55] provide a general quasi-Newton acceleration scheme for iterative methods based on a quadratic approximation of the iteration map instead of the loss.

There has also been significant work in finding sparse factors via ℓ_1-penalization for matrices [37] and tensors [41, 51, 17, 36]. Sparse factors often provide more easily interpreted models, and penalization may also accelerate the convergence. While the factor matrices generated by CP-APR may be naturally sparse without imposing an ℓ_1-penalty, the degree of sparsity is not currently tunable. One may also consider extensions of this work in the context of missing data [22, 7, 50, 1] and for alternative tensor factorizations such as Tucker [17].

Perhaps most challenging, however, are open questions related to rank and inference. Questions about how to choose rank are not new, but given the context of sparse count data, might that structure be exploited to derive a sensible heuristic or even rigorous criterion for choosing the rank? We already see that Assumption 3.2 imposes an upper bound on the rank to ensure algorithmic convergence. Regarding inference, our focus in this work was in thoroughly developing the algorithmic groundwork for fitting a PTF model for sparse count data. CP-APR can be used to estimate latent structure. Once an estimate is in hand, however, it is natural to ask how much uncertainty there is in that estimate. For example, is it possible to put a confidence interval around the entries in the fitted factor matrices, especially zero or near-zero entries? Given that inference for the related but simpler case of Poisson regression has been worked out, we suspect that a sensible solution is waiting to be found. The benefits of answering these questions warrant further investigation. We highlight them as important topics for future research.

Appendix A. Notation details.

Outer product. The outer product of N vectors is an N-way tensor. For example, $(\mathbf{a} \circ \mathbf{b} \circ \mathbf{c})_{ijk} = a_i b_j c_k$.

Elementwise multiplication and division. Let $\mathcal{A}$ and $\mathcal{B}$ be two same-size tensors (or matrices). Then $\mathcal{C} = \mathcal{A} * \mathcal{B}$ yields a tensor that is the same size as $\mathcal{A}$ (and $\mathcal{B}$) such that $c_{\mathbf{i}} = a_{\mathbf{i}} b_{\mathbf{i}}$ for all $\mathbf{i}$. Likewise, $\mathcal{C} = \mathcal{A} \oslash \mathcal{B}$ yields a tensor that is the same size as $\mathcal{A}$ (and $\mathcal{B}$) such that $c_{\mathbf{i}} = a_{\mathbf{i}}/b_{\mathbf{i}}$ for all $\mathbf{i}$.

Khatri–Rao product. Give two matrices $\mathbf{A}$ and $\mathbf{B}$ of sizes $I_1 \times R$ and $I_2 \times R$; then $\mathbf{C} = \mathbf{A} \odot \mathbf{B}$ is a matrix of size $I_1 I_2 \times R$ such that

$$\mathbf{C} = \begin{bmatrix} \mathbf{a}_1 \otimes \mathbf{b}_1 & \mathbf{a}_2 \otimes \mathbf{b}_2 & \cdots & \mathbf{a}_R \otimes \mathbf{b}_R \end{bmatrix},$$

where the Kronecker product of two vectors of size I_1 and I_2 is a vector of length $I_1 I_2$ given by

$$\mathbf{a} \otimes \mathbf{b} = \begin{bmatrix} a_1 \mathbf{b} \\ a_2 \mathbf{b} \\ \vdots \\ a_{I_1} \mathbf{b} \end{bmatrix}.$$

Matricization of a tensor. The mode-n matricization or unfolding of a tensor $\mathcal{X}$ is denoted by $\mathbf{X}_{(n)}$ and is of size $I_n \times J_n$, where $J_n \equiv \prod_{m \neq n} I_n$. In this case, tensor element $\mathbf{i}$ maps to matrix element (i, j), where

$$i = i_n \quad \text{and} \quad j = 1 + \sum_{\substack{k=1 \\ k \neq n}}^{N} (i_k - 1) \left(\prod_{\substack{m=1 \\ m \neq n}}^{k-1} I_m \right).$$

Appendix B. Proof of Lemma 3.1. In this section, we provide a proof for Lemma 3.1. We first establish two useful lemmas.

LEMMA B.1. *Let $\mathcal{X}$ be fixed, let $\mathcal{M} = [\![\boldsymbol{\lambda}; \mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(N)}]\!]$, and let $f(\mathcal{M})$ be the objective function as in* (3.1)*. If $f(\mathcal{M}) \leq \zeta$ for some constant $\zeta > 0$, then there exists constants $\xi', \xi > 0$ (depending on $\mathcal{X}$ and ζ) such that $\mathbf{e}^{\mathsf{T}} \boldsymbol{\lambda} \in [\xi', \xi]$.*

Proof. Because the factor matrices are column stochastic, we can observe that

$$\begin{aligned} f(\mathcal{M}) &= \mathbf{e}^{\mathsf{T}} \boldsymbol{\lambda} - \sum_{\mathbf{i}} x_{\mathbf{i}} \log \left(\sum_r \lambda_r \, a^{(n)}_{i_1 r} \cdots a^{(n)}_{i_N r} \right) \\ &\geq \mathbf{e}^{\mathsf{T}} \boldsymbol{\lambda} - \vartheta \log \left(\mathbf{e}^{\mathsf{T}} \boldsymbol{\lambda} \right), \quad \text{where} \quad \vartheta = \left(\prod_{n=1}^{N} I_n \right) \max_{\mathbf{i}} x_{\mathbf{i}}. \end{aligned} \tag{B.1}$$

We have $\zeta \geq \mathbf{e}^{\mathsf{T}} \boldsymbol{\lambda} - \vartheta \log \left(\mathbf{e}^{\mathsf{T}} \boldsymbol{\lambda} \right)$. Let $g(\alpha) = \alpha - \vartheta \log(\alpha)$, where $\alpha > 0$. We show that $g(\alpha) \leq \zeta$ implies there exists $\xi', \xi > 0$ such that $\alpha \in [\xi', \xi]$. First assume there is no such lower bound ξ'. Then there is a sequence α_n tending to zero such that $g(\alpha_n) \leq \zeta$. But for sufficiently large n, we have that $-\vartheta \log(\alpha_n) > \zeta$. Since $\alpha_n > 0$ for all n, we have that for sufficiently large n the function $g(\alpha_n) > \zeta$. Therefore, there is such a lower bound ξ'.

Now suppose there is no such upper bound ξ, and therefore there is an unbounded and increasing sequence α_n tending to infinity such that $g(\alpha_n) \leq \zeta$ for all n. Note that $g'(\alpha) = 1 - \vartheta/\alpha$. Since $g(\alpha)$ is convex, we have that

$$g(\alpha) \geq g(2\vartheta) + g'(2\vartheta)(\alpha - 2\vartheta) = g(2\vartheta) + \frac{1}{2}\alpha - \vartheta.$$

This inequality, however, indicates that for sufficiently large n, the right-hand side is greater than ζ. Therefore, there must be an upper bound ξ. Substituting $\alpha = \mathbf{e}^{\mathsf{T}}\boldsymbol{\lambda}$ completes the proof. □

LEMMA B.2. *Let $\mathcal{X}$ be fixed, and let $f(\mathcal{M})$ be the objective function as in* (3.1). *Let $\Omega(\zeta)$ be the convex hull of the level set of f as defined in* (3.3). *The function $f(\mathcal{M})$ is bounded for all $\mathcal{M} \in \Omega(\zeta)$.*

Proof. Let $\bar{\mathcal{M}}, \hat{\mathcal{M}} \in \{\, \mathcal{M} \mid f(\mathcal{M}) \leq \zeta \,\}$. Define $\tilde{\mathcal{M}}$ to be the convex combination

$$\tilde{\mathcal{M}} = \alpha\bar{\mathcal{M}} + (1-\alpha)\hat{\mathcal{M}}, \quad \text{where} \quad \alpha \in [0.5, 1).$$

Note that the restriction on α is arbitrary but makes the proof simpler later on. Observe that

$$\tilde{m}_{\mathbf{i}} = \sum_r \left\{ \left(\alpha\bar{\lambda}_r + (1-\alpha)\hat{\lambda}_r\right) \prod_n \left(\alpha\bar{a}^{(n)}_{i_n r} + (1-\alpha)\hat{a}^{(n)}_{i_n r}\right) \right\}.$$

On the one hand, by Lemma B.1, there exists $\xi > 0$ such that

$$\tilde{m}_{\mathbf{i}} \leq \sum_r \left(\alpha\bar{\lambda}_r + (1-\alpha)\hat{\lambda}_r\right) = \alpha \sum_r \bar{\lambda}_r + (1-\alpha) \sum_r \hat{\lambda}_r \leq \alpha\xi + (1-\alpha)\xi = \xi.$$

On the other hand,

$$\tilde{m}_{\mathbf{i}} \geq \sum_r \left\{ \alpha\bar{\lambda}_r \prod_n \alpha\bar{a}^{(n)}_{i_n r} \right\} = \alpha^{N+1}\bar{m}_{\mathbf{i}}.$$

Thus,

$$\alpha^{N+1}\bar{m}_{\mathbf{i}} \leq \tilde{m}_{\mathbf{i}} \leq \bar{m}_{\mathbf{i}} + \xi.$$

Now consider

$$\begin{aligned} \tilde{m}_{\mathbf{i}} - x_{\mathbf{i}} \log \tilde{m}_{\mathbf{i}} &\leq \bar{m}_{\mathbf{i}} + \xi - x_{\mathbf{i}} \log \alpha^{N+1}\bar{m}_{\mathbf{i}} \\ &= (\bar{m}_{\mathbf{i}} - x_{\mathbf{i}} \log \bar{m}_{\mathbf{i}}) + \xi - (N+1)x_{\mathbf{i}} \log \alpha \\ &\leq (\bar{m}_{\mathbf{i}} - x_{\mathbf{i}} \log \bar{m}_{\mathbf{i}}) + \xi + (N+1)x_{\mathbf{i}} \log 2. \end{aligned}$$

Thus,

$$f(\tilde{\mathcal{M}}) \leq f(\bar{\mathcal{M}}) + \xi \prod_n I_n + (N+1)\log 2 \sum_{\mathbf{i}} x_{\mathbf{i}} \leq \xi \left(1 + \prod_n I_n\right) + (N+1)\log 2 \sum_{\mathbf{i}} x_{\mathbf{i}}. \quad \square$$

Given these two lemmas, we are finally ready to provide the proof of Lemma 3.1.

Proof of Lemma 3.1. Fix ζ. If $\{\, \mathcal{M} \in \Omega \mid f(\mathcal{M}) \leq \zeta \,\}$ is empty, then $\Omega(\zeta)$ is empty and there is nothing left to do. Thus, assume $\{\, \mathcal{M} \in \Omega \mid f(\mathcal{M}) \leq \zeta \,\}$ is nonempty.

Since f is continuous at all $\mathcal{M} \in \Omega$ for which $f(\mathcal{M})$ is finite, f is obviously continuous on $\Omega(\zeta)$ by Lemma B.2. Since f is continuous, $\{\, \mathcal{M} \in \Omega \mid f(\mathcal{M}) \leq \zeta \,\}$ is closed because it is the preimage of the closed set $(-\infty, \zeta]$ under f; thus, $\Omega(\zeta)$ is closed because it is a convex combination of closed sets. Consequently, we only need to show that $\Omega(\zeta)$ is bounded. Assume the contrary. Then there exists a sequence of models $\mathcal{M}_k = [\![\boldsymbol{\lambda}_k; \mathbf{A}_k^{(1)}, \ldots, \mathbf{A}_k^{(N)}]\!] \in \Omega(\zeta)$ such that $\mathbf{e}^{\mathsf{T}}\boldsymbol{\lambda}_k \to \infty$. By Lemma B.2, $f(\mathcal{M})$ is finite on $\Omega(\zeta)$, but this contradicts Lemma B.1. Hence, the claim. ∎

Appendix C. Deriving the MM updates. In this section we derive the MM update rules used to solve the subproblem. We first verify that (4.2) majorizes (4.1). For convenience let $\mathbf{C} = \mathbf{B}^{\mathsf{T}}$ so that (4.1) reduces to

$$\min_{\mathbf{C} \geq 0} f(\mathbf{C}^{\mathsf{T}}) = \sum_{ij} \mathbf{c}_i^{\mathsf{T}} \boldsymbol{\pi}_j - x_{ij} \log\left(\mathbf{c}_i^{\mathsf{T}} \boldsymbol{\pi}_j\right). \tag{C.1}$$

Proofs of the next two lemmas are given by Lee and Seung in [32], but their arguments do not carefully handle boundary points. The following two lemmas and their proofs treat with more rigor the existence and value of updates when anchor points lie on admissible regions of the boundary.

LEMMA C.1. *Let $x \geq 0$ be a scalar and $\boldsymbol{\pi} \geq 0$, $\boldsymbol{\pi} \neq 0$, be a vector of length R. For a vector $\mathbf{c} \geq 0$, $\mathbf{c} \neq 0$, of length R, let the function f be defined by*

$$f(\mathbf{c}) = \mathbf{c}^{\mathsf{T}} \boldsymbol{\pi} - x \log\left(\mathbf{c}^{\mathsf{T}} \boldsymbol{\pi}\right).$$

Then f is majorized at $\bar{\mathbf{c}} \geq 0$ by

$$g(\mathbf{c}, \bar{\mathbf{c}}) = \mathbf{c}^{\mathsf{T}} \boldsymbol{\pi} - x \sum_{r=1}^{R} \alpha_r \log\left(\frac{c_r \pi_r}{\alpha_r}\right), \quad \textit{where} \quad \alpha_r = \frac{\bar{c}_r \pi_r}{\bar{\mathbf{c}}^{\mathsf{T}} \boldsymbol{\pi}}.$$

Proof. If $x = 0$, then $g(\mathbf{c}, \bar{\mathbf{c}}) = f(\mathbf{c})$ for all $\mathbf{c}$, and g trivially majorizes f at $\bar{\mathbf{c}}$. Consider the case when $x > 0$. It is immediate that $g(\bar{\mathbf{c}}, \bar{\mathbf{c}}) = f(\bar{\mathbf{c}})$. The majorization follows from the fact that log is strictly concave and that we can write $\mathbf{c}^{\mathsf{T}} \boldsymbol{\pi}$ as a convex combination of the elements $c_r \pi_r / \alpha_r$. Note that if any elements $\bar{c}_r \pi_r$ are zero, they do not contribute to the sum since we assume $0 \cdot \log(\mu) = 0$ for all $\mu \geq 0$ and $\alpha_r = 0$. ∎

We now derive an expression for the unique global minimizer of majorization. The majorization defined in (4.2) can be expressed in terms of $\mathbf{C}$ as

$$g(\mathbf{C}, \bar{\mathbf{C}}) = \sum_{rij} \left[c_{ri} \pi_{rj} - \alpha_{rij} x_{ij} \log\left(\frac{c_{ri} \pi_{rj}}{\alpha_{rij}}\right)\right], \quad \text{where} \quad \alpha_{rij} = \frac{\bar{c}_{ri} \pi_{rj}}{\sum_r \bar{c}_{ri} \pi_{rj}}. \tag{C.2}$$

LEMMA C.2. *Let f and g be as defined in* (C.1) *and* (4.2)*, respectively. Then for all $\bar{\mathbf{C}} \geq \mathbf{0}$ such that $f(\bar{\mathbf{C}}^{\mathsf{T}})$ is finite, the function $g(\cdot, \bar{\mathbf{C}})$ has a unique global minimum $\mathbf{C}_*$ which is given by $(\mathbf{C}_*)_{ri} = \sum_j \alpha_{rij} x_{ij}$, where $\alpha_{rij} = \bar{c}_{ri} \pi_{rj} / \bar{\mathbf{c}}_i^{\mathsf{T}} \boldsymbol{\pi}_j$ for all $r = 1, \ldots, R$, $i = 1, \ldots, I$.*

Proof. Because $g(\mathbf{C}, \bar{\mathbf{C}})$ separates in the elements of $\mathbf{C}$ we focus on solving each elementwise minimization problem. Dropping subscripts, the minimization problem with respect to c_{ri} can be rewritten as

$$\min_{c \geq 0} \; c - \sum_j \alpha_j x_j \log\left(\frac{c\pi_j}{\alpha_j}\right), \tag{C.3}$$

where we have used the fact that $\sum_j \pi_j = 1$. It is sufficient to prove that this univariate problem has a unique global minimizer, $c_* = \sum_j \alpha_j x_j$. First, consider the case where the second term is nonzero. Inspecting the stationarity condition reveals the solution. Moreover, the function is strictly convex and so has a unique global minimum. Second, consider the case where the second term is zero. Then, it is immediate that the unique global minimum is $c_* = 0$. Moreover, the second term can vanish only when $\sum_j \alpha_j x_j = 0$, and so the formula applies. ∎

Appendix D. Proof of Theorem 4.3. In this section, we prove that the MM algorithm in Algorithm 2 solves (4.1). We first establish the following general result for algorithm maps. Part (a) is a simple version of Zangwill's convergence theorem [54, p. 91] in the case where the objective function and the algorithm map are both continuous. The proof of part (b) follows arguments of part of a proof for a different but related property on MM iterates in [29, p. 198].

THEOREM D.1. *Let f be a continuous function on a domain $\mathcal{D}$, and let ψ be a continuous iterative map from $\mathcal{D}$ into $\mathcal{D}$ such that $f(\psi(\mathbf{x})) < f(\mathbf{x})$ for all $\mathbf{x} \in \mathcal{D}$ with $\psi(\mathbf{x}) \neq \mathbf{x}$. Suppose there is an $\mathbf{x}_0$ such that the set $\mathcal{L}_f(\mathbf{x}_0) \equiv \{\, \mathbf{x} \in \mathcal{D} \mid f(\mathbf{x}) \leq f(\mathbf{x}_0) \,\}$ is compact. Define $\mathbf{x}_{k+1} = \psi(\mathbf{x}_k)$ for $k = 0, 1, \ldots$. Then* (a) *the sequence of iterates $\{\mathbf{x}_k\}$ has at least one limit point and all its limit points are fixed points of ψ, and* (b) *the distance between successive iterates converges to 0, i.e., $\|\mathbf{x}_{k+1} - \mathbf{x}_k\| \to 0$.*

Proof. The proof of (a) follows that of Proposition 10.3.2 of [29]. First note that the sequence of iterates must be in $\mathcal{L}_f(\mathbf{x}_0)$ because $f(\mathbf{x}_k) \leq f(\mathbf{x}_0)$ for all k. Since $\mathcal{L}_f(\mathbf{x}_0)$ is compact, $\{\mathbf{x}_k\}$ has a convergent subsequence whose limit is in $\mathcal{L}_f(\mathbf{x}_0)$; denote this as $\mathbf{x}_{k_\ell} \to \mathbf{x}_*$ as $\ell \to \infty$. Since f is assumed to be continuous, $\lim f(\mathbf{x}_{k_\ell}) = f(\mathbf{x}_*)$. Moreover, clearly $f(\mathbf{x}_*) \leq f(\mathbf{x}_{k_\ell})$ for all k_ℓ.

Note that $f(\psi(\mathbf{x}_{k_\ell})) \leq f(\mathbf{x}_{k_\ell})$. Taking the limit of both sides and applying the continuity of ψ and f, we must have that $f(\psi(\mathbf{x}_*)) \leq f(\mathbf{x}_*)$. But we also have that

$$f(\mathbf{x}_*) \leq f(\mathbf{x}_{k_{\ell+1}}) \leq f(\mathbf{x}_{k_\ell+1}) = f(\psi(\mathbf{x}_{k_\ell})).$$

Again taking limits we obtain $f(\mathbf{x}_*) \leq f(\psi(\mathbf{x}_*))$. Therefore $f(\mathbf{x}_*) = f(\psi(\mathbf{x}_*))$. But by assumption, this equality implies that $\mathbf{x}_*$ is a fixed point of ψ, and thus (a) is proved.

We now turn to the proof of (b), which follows the proof of Proposition 10.3.3 in [29]. Recall $\{\mathbf{x}_k\}$ denotes the iterate sequence. Since $f(\mathbf{x}_k)$ is decreasing and f is bounded below on $\mathcal{L}_f(\mathbf{x}_0)$, we can assert that $f(\mathbf{x}_k)$ is a convergent sequence with a limit f_*. Assume the contrary of (b), i.e., that there exists an $\epsilon > 0$ and a subsequence $\{k_\ell\}$ of the indices such that

$$\|\mathbf{x}_{k_\ell+1} - \mathbf{x}_{k_\ell}\| > \epsilon \text{ for all } k_\ell. \tag{D.1}$$

Note that this subsequence is different from the one discussed in proving part (a).

Since $\mathbf{x}_{k_\ell} \in \mathcal{L}_f(\mathbf{x}_0)$, by possibly restricting $\{k_\ell\}$ to a further subsequence, we may assume that $\mathbf{x}_{k_\ell}$ converges to a limit $\mathbf{u}$. By possibly restricting $\{k_\ell\}$ to yet a further subsequence, we may additionally assume that $\mathbf{x}_{k_\ell+1}$ converges to a limit $\mathbf{v}$. By (D.1), we can conclude $\|\mathbf{v}-\mathbf{u}\| \geq \epsilon$. Note that $\mathbf{x}_{k_\ell+1} = \psi(\mathbf{x}_{k_\ell})$. Taking the limit of both sides and using the continuity of ψ we obtain $\psi(\mathbf{u}) = \mathbf{v}$. Additionally, using the continuity of f,

$$f(\mathbf{u}) = \lim_{\ell\to\infty} f(\mathbf{x}_{k_\ell}) = f_* = \lim_{\ell\to\infty} f(\mathbf{x}_{k_\ell+1}) = f(\mathbf{v}).$$

Since $\mathbf{v} = \psi(\mathbf{u})$, we have that $f(\mathbf{u}) = f(\psi(\mathbf{u}))$, which by assumption occurs if and only if $\mathbf{u} = \psi(\mathbf{u})$. This implies that $\mathbf{u} = \mathbf{v}$, and we have arrived at a contradiction. □

We now prove a series of lemmas leading up to a proof of the desired convergence result.

LEMMA D.2. *Let $\mathbf{B} \geq 0$ such that $f(\mathbf{B})$ is finite and suppose $\mathbf{B} \neq \mathbf{B} * \mathbf{\Phi}$. Then $f(\mathbf{B}) > f(\mathbf{B} * \mathbf{\Phi})$.*

Proof. By Lemma C.2 $(\mathbf{B} * \mathbf{\Phi})^\mathsf{T}$ is the unique global minimizer of $g(\cdot, \mathbf{B}^\mathsf{T})$ which majorizes f at $\mathbf{B}^\mathsf{T}$. Therefore, if $\mathbf{B} \neq \mathbf{B} * \mathbf{\Phi}$, we must have $f(\mathbf{B}) = g(\mathbf{B}^\mathsf{T}, \mathbf{B}^\mathsf{T}) > g((\mathbf{B} * \mathbf{\Phi})^\mathsf{T}, \mathbf{B}^\mathsf{T}) \geq f(\mathbf{B} * \mathbf{\Phi})$. □

LEMMA D.3. *Let f be as defined in (4.1). For any nonnegative matrix $\mathbf{B}_0$ such that $f(\mathbf{B}_0)$ is finite, the level set $\mathcal{L}_f(\mathbf{B}_0) = \{\, \mathbf{B} \geq 0 \mid f(\mathbf{B}) \leq f(\mathbf{B}_0) \,\}$ is compact.*

Proof. The proof follows the same logic as the proof for Lemma B.1. □

LEMMA D.4. *Let f be as defined in (4.1) and ψ be as defined in (4.3), and suppose Assumption 4.2 is satisfied. For any nonnegative matrix $\mathbf{B}_k$ such that $f(\mathbf{B}_0)$ is finite, the sequence $\mathbf{B}_{k+1} = \psi(\mathbf{B}_k)$ converges.*

Proof. Note that all limit points of ψ are fixed points of f by Theorem D.1.

First, we show that the set of fixed point is finite. Suppose that $\mathbf{B}$ is a fixed point of ψ. Then we must have $\mathbf{B} * (\mathbf{E} - \mathbf{\Phi}(\mathbf{B})) = 0$. By Lemma 3.4, it can be verified that $\mathbf{B}$ is the *unique* global minimizer of

$$\min\ f(\mathbf{U}) \quad \text{s.t. } \mathbf{U} \in \{\, \mathbf{U} \geq 0 \mid u_{ir} = 0 \text{ if } b_{ir} = 0 \,\},$$

where f is as defined in (4.1). Therefore, any fixed point that has the same zero pattern of $\mathbf{B}$ must be equal to $\mathbf{B}$. Since there are only a finite number of possible zero patterns, the number of fixed points is finite.

Since every limit point is a fixed point by Theorem D.1(a), there are only finitely many limit points. Let $\{\mathcal{N}_p\}$ denote a collection of arbitrarily small neighborhoods around each fixed point indexed by p. Only finitely many iterates $\mathbf{B}_k$ are in $\mathcal{L}_f(\mathbf{B}_0) - \cup_p \mathcal{N}_p$. So, all but finitely many iterates $\mathbf{B}_k$ will be in $\cup_p \mathcal{N}_p$. But $\|\mathbf{B}_{k+1} - \mathbf{B}_k\|$ eventually becomes smaller than smallest distance between any two neighborhoods by Theorem D.1(b). Therefore the sequence $\mathbf{B}_k$ must belong to one of the neighborhoods for all but finitely many k. So, any sequence of iterates must eventually converge to exactly one of the fixed points of ψ. □

We now argue that it is impossible for the MM iterate sequence to converge to a non-KKT point if it has been appropriately initialized.

LEMMA D.5. *Let f be as defined in (4.1) and suppose Assumption 4.2 is satisfied. Suppose $\mathbf{B}_k \to \mathbf{B}_*$ is a convergent sequence of iterates defined by (4.3) with $\mathbf{B}_0 \geq 0$ and $f(\mathbf{B}_0)$ finite. If $(\mathbf{B}_0)_{ir} > 0$ for all (i,r) such that $(\mathbf{\Phi}(\mathbf{B}_*))_{ir} > 1$, then $\nabla f(\mathbf{B}_*) \geq \mathbf{0}$.*

Proof. We give a proof by contradiction. Suppose there exists (i,r) such that $(\mathbf{B}_0)_{ir} > 0$ but $(\nabla f(\mathbf{B}_*))_{ir} < 0$. Since $\mathbf{B}_*$ is a fixed point of ψ, we must have $[1 - (\mathbf{\Phi}(\mathbf{B}_*))_{ir}](\mathbf{B}_*)_{ir} = 0$. By our assumption, however, $(\nabla f(\mathbf{B}_*))_{ir} = [1 - (\mathbf{\Phi}(\mathbf{B}_*))_{ir}] < 0$. Thus, we must have $(\mathbf{B}_*)_{ir} = 0$. On the other hand, $(\mathbf{B}_k)_{ir} > 0$ for all k (proof left to reader). Since $\mathbf{\Phi}(\cdot)$ is a continuous function of $\mathbf{B}$ on $\mathcal{L}_f(\mathbf{B}_0)$, we know that there exists some K such that $k > K$ implies $\mathbf{B}_k$ is close enough to $\mathbf{B}_*$ such that $(\nabla f(\mathbf{B}_k))_{ir} = [1 - (\mathbf{\Phi}(\mathbf{B}_k))_{ir}] < 0$. Since $(\mathbf{B}_k)_{ir} > 0$, we have $[1 - (\mathbf{\Phi}(\mathbf{B}_k))_{ir}](\mathbf{B}_k)_{ir} < 0$, which implies $(\mathbf{B}_k)_{ir} < (\mathbf{B}_{k+1})_{ir}$ for all $k > K$. But this contradicts $\lim_{k\to\infty}(\mathbf{B}_k)_{ir} = (\mathbf{B}_*)_{ir} = 0$. Hence, the claim. ∎

We now prove Theorem 4.3.

Proof of Theorem 4.3. By Lemma D.4, the sequence $\{\mathbf{B}_k\}$ converges; we call the limit point $\mathbf{B}_*$. At this limit point, we have (a) $\mathbf{B}_* \geq 0$, (b) $\nabla f(\mathbf{B}_*) \geq 0$ by Lemma D.5, and (c) $\mathbf{B}_* * \nabla f(\mathbf{B}_*) = 0$ by virtue of $\mathbf{B}_*$ being a fixed point of ψ. Thus, the point $\mathbf{B}_*$ satisfies the KKT conditions with respect to (4.1). Furthermore, since f is strictly convex by Lemma 3.4, we can conclude that $\mathbf{B}_*$ is the global minimum of f. ∎

Appendix E. Numerical experiment details for section 6.1. All implementations are from version 2.5 of Tensor Toolbox for MATLAB [4]. All methods use a common initial guess for the solution.

- Lee–Seung LS: Implemented in `cp_nmu` as descibed in [3]. We use the default parameters except that the maximum number of iterations (`maxiters`) is set to 200 and the tolerance on the change in the fit (`tol`) is set to 10^{-8}.
- CP-ALS: Implemented in `cp_als` as described in [3]. We use the default parameter settings except that the maximum number iterations (`maxiters`) is 200 and the tolerance on the changes in fit (`tol`) is 10^{-8}.
- Lee–Seung KL: Implemented in `cp_apr` as described in this paper. The parameters are set as follows: $k_{\max} = 200$ (`maxiters`), $\tau = 10^{-8}$ (`tol`), $\kappa = 0$ (`kappa`), $\ell_{\max} = 1$ (`maxinneriters`), $\epsilon = 0$ (`epsilon`).
- CP-APR: Implemented in `cp_apr` as described in this paper. The parameters are set as follows: $k_{\max} = 200$ (`maxiters`), $\tau = 10^{-4}$ (`tol`), $\kappa = 10^{-2}$ (`kappa`), $\kappa_{\text{tol}} = 10^{-10}$ (`kappatol`), $\ell_{\max} = 10$ (`maxinneriters`), $\epsilon = 0$ (`epsilon`).

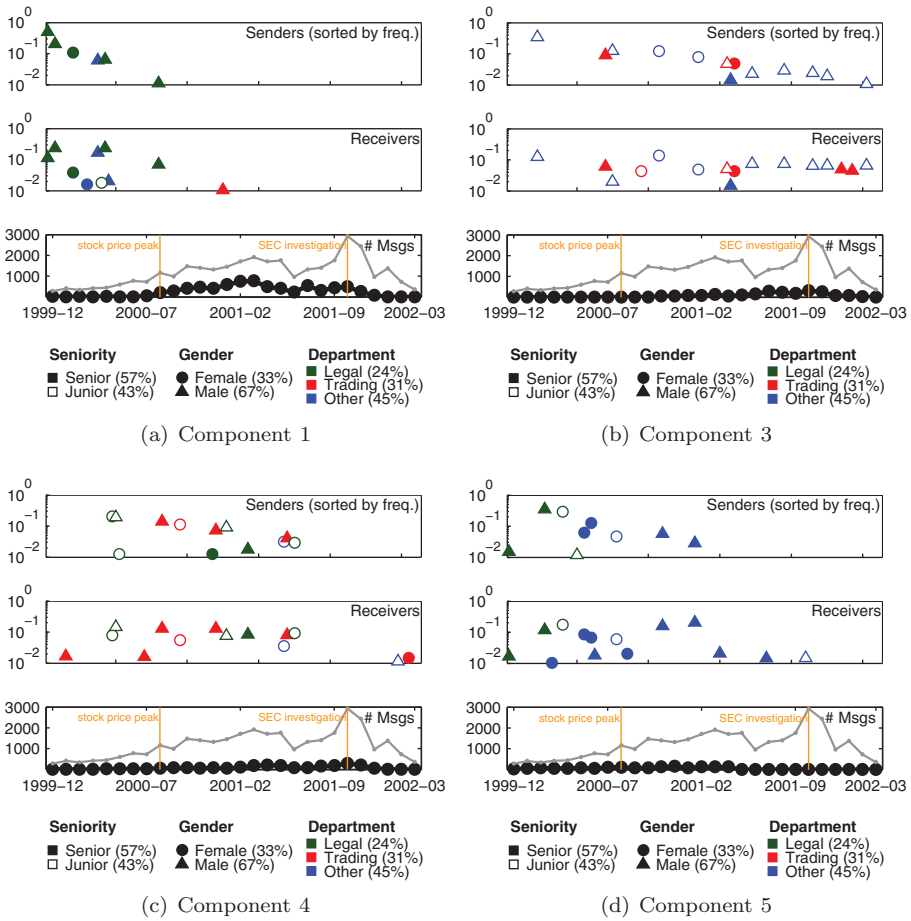
We compare the methods in terms of their FMS, defined as follows. Let $\mathcal{M} = [\![\boldsymbol{\lambda}; \mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(N)}]\!]$ be the true model and let $\bar{\mathcal{M}} = [\![\bar{\boldsymbol{\lambda}}; \bar{\mathbf{A}}^{(1)}, \ldots \bar{\mathbf{A}}^{(N)}]\!]$ be the computed solution. The score of $\bar{\mathcal{M}}$ is computed as

$$\text{score}(\bar{\mathcal{M}}) = \frac{1}{R}\sum_r \left(1 - \frac{|\xi_r - \bar{\xi}_r|}{\max\{\xi_r, \bar{\xi}_r\}}\right) \prod_n \frac{\mathbf{a}_r^{(n)\mathsf{T}} \bar{\mathbf{a}}_r^{(n)}}{\|\mathbf{a}_r^{(n)}\| \|\bar{\mathbf{a}}_r^{(n)}\|},$$

$$\text{where} \quad \xi_r = \lambda_r \prod_n \|\mathbf{a}_r^{(n)}\| \quad \text{and} \quad \bar{\xi}_r = \bar{\lambda}_r \prod_n \|\bar{\mathbf{a}}_r^{(n)}\|.$$

The FMS is a rather abstract measure, so we also give results for the number of columns in $\mathbf{A}^{(1)}$ that are correctly identified. In other words, we count the number of times that the cosine of the angle between the true solution and the computed solution is greater than 0.95, mathematically, $\mathbf{a}_r^{(1)\mathsf{T}} \bar{\mathbf{a}}_r^{(1)} / \|\mathbf{a}_r^{(1)}\| \|\bar{\mathbf{a}}_r^{(1)}\| \geq 0.95$. We use the first mode, but the results are representative of the other modes.

Appendix F. Additional Enron results. Figure F.1 illustrates the four components omitted in Figure 6.2.



(a) Component 1

(b) Component 3

(c) Component 4

(d) Component 5

FIG. F.1. *Remaining components from factorizing the Enron data.*

Acknowledgments. We thank our colleagues at Sandia for numerous helpful conversations in the course of this work, especially Grey Ballard and Todd Plantenga. We also thank Kenneth Lange for pointing us to relevant references on emission tomography. Finally, we thank the anonymous referees and associate editor for suggestions which greatly improved the quality of the manuscript.

REFERENCES

[1] E. ACAR, D. M. DUNLAVY, T. G. KOLDA, AND M. MØRUP, *Scalable tensor factorizations for incomplete data*, Chemometrics and Intelligent Laboratory Systems, 106 (2011), pp. 41–56.

[2] B. W. BADER, M. W. BERRY, AND M. BROWNE, *Discussion tracking in Enron email using PARAFAC*, in Survey of Text Mining II: Clustering, Classification, and Retrieval, M. W. Berry and M. Castellanos, eds., Springer, London, 2008.

[3] B. W. BADER AND T. G. KOLDA, *Efficient MATLAB computations with sparse and factored tensors*, SIAM J. Sci. Comput., 30 (2007), pp. 205–231.

[4] B. W. BADER, T. G. KOLDA, ET AL., *MATLAB Tensor Toolbox version* 2.5, http://www.sandia.gov/~tgkolda/tensor_toolbox (2012).

[5] D. P. BERTSEKAS AND J. N. TSITSIKLIS, *Parallel and Distributed Computation: Numerical Methods*, Prentice-Hall, Englewood Cliffs, NJ, 1989.

[6] R. Bro and S. De Jong, *A fast non-negativity-constrained least squares algorithm*, J. Chemometrics, 11 (1997), pp. 393–401.
[7] A. M. Buchanan and A. W. Fitzgibbon, *Damped Newton algorithms for matrix factorization with missing data*, in CVPR'05: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, IEEE Computer Society, New York, 2005, pp. 316–322.
[8] J. D. Carroll and J. J. Chang, *Analysis of individual differences in multidimensional scaling via an N-way generalization of "Eckart-Young" decomposition*, Psychometrika, 35 (1970), pp. 283–319.
[9] A. Cichocki, R. Zdunek, S. Choi, R. Plemmons, and S.-I. Amari, *Non-negative tensor factorization using alpha and beta divergences*, in ICASSP 07: Proceedings of the International Conference on Acoustics, Speech, and Signal Processing, 2007.
[10] V. de Silva and L.-H. Lim, *Tensor rank and the ill-posedness of the best low-rank approximation problem*, SIAM J. Matrix Anal. Appl., 30 (2008), pp. 1084–1127.
[11] I. Dhillon and S. Sra, *Generalized nonnegative matrix approximations with Bregman divergences*, Adv. Neural Inf. Process. Syst. 18, MIT Press, Cambridge, MA, 2006, pp. 283–290.
[12] D. M. Dunlavy, T. G. Kolda, and W. P. Kegelmeyer, *Multilinear algebra for analyzing data with multiple linkages*, in Graph Algorithms in the Language of Linear Algebra, J. Kepner and J. Gilbert, eds., Fundam. Algorithms, SIAM, Philadelphia, 2011, pp. 85–114.
[13] D. M. Dunlavy, T. G. Kolda, and E. Acar, *Temporal link prediction using matrix and tensor factorizations*, ACM Trans. Knowledge Discovery from Data, 5 (2011), 10.
[14] C. Févotte and J. Idier, *Algorithms for nonnegative matrix factorization with the β-divergence*, Neural Computation, 23 (2011), pp. 2421–2456.
[15] L. Finesso and P. Spreij, *Nonnegative matrix factorization and* I*-divergence alternating minimization*, Linear Algebra Appl., 416 (2006), pp. 270–287.
[16] D. FitzGerald, M. Cranitch, and E. Coyle, *Non-negative tensor factorisation for sound source separation*, in Proceedings of the Irish Signals and Systems Conference, Dublin, 2005, pp. 8–12.
[17] M. P. Friedlander and K. Hatz, *Computing nonnegative tensor factorizations*, Comput. Optim. Appl., 23 (2008), pp. 631–647.
[18] N. Gillis and F. Glineur, *Nonnegative Factorization and the Maximum Edge Biclique Problem*, arXiv:0810.4225, 2008.
[19] N. Gillis and F. Glineur, *Accelerated multiplicative updates and hierarchical ALS algorithms for nonnegative matrix factorization*, Neural Comput., 24 (2011), pp. 1085–1105.
[20] E. F. Gonzalez and Y. Zhang, *Accelerating the Lee-Seung Algorithm for Nonnegative Matrix Factorization*, Technical report, Rice University, 2005.
[21] R. A. Harshman, *Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multi-modal factor analysis*, UCLA Working Papers in Phonetics, 16 (1970), pp. 1–84; also available online from http://www.psychology.uwo.ca/faculty/harshman/wpppfac0.pdf.
[22] H. A. L. Kiers, *Weighted least squares fitting using ordinary least squares algorithms*, Psychometrika, 62 (1997), pp. 215–266.
[23] D. Kim, S. Sra, and I. S. Dhillon, *Fast projection-based methods for the least squares nonnegative matrix approximation problem*, Statist. Anal. Data Mining, 1 (2008), pp. 38–51.
[24] D. Kim, S. Sra, and I. S. Dhillon, *Tackling box-constrained optimization via a new projected quasi-Newton approach*, SIAM J. Sci. Comput., 32 (2010), pp. 3548–3563.
[25] H. Kim and H. Park, *Nonnegative matrix factorization based on alternating nonnegativity constrained least squares and active set method*, SIAM J. Matrix Anal. Appl., 30 (2008), pp. 713–730.
[26] H. Kim, H. Park, and L. Elden, *Non-negative tensor factorization based on alternating large-scale non-negativity-constrained least squares*, in Proceedings of the 7th IEEE International Conference on Bioinformatics and Bioengineering, 2007, pp. 1147–1151.
[27] J. Kim and H. Park, *Fast nonnegative matrix factorization: An active-set-like method and comparisons*, SIAM J. Sci. Comput., 33 (2011), pp. 3261–3281.
[28] K. Lange, *Convergence of EM image reconstruction algorithms with Gibbs smoothing*, IEEE Trans. Medical Imaging, 9 (1990), pp. 439–446.
[29] K. Lange, *Optimization*, Springer, New York, 2004.
[30] K. Lange and R. Carson, *EM reconstruction algorithms for emission and transmission tomography*, J. Comput. Assisted Tomography, 8 (1984), pp. 306–316.
[31] D. D. Lee and H. S. Seung, *Learning the parts of objects by non-negative matrix factorization*, Nature, 401 (1999), pp. 788–791.
[32] D. D. Lee and H. S. Seung, *Algorithms for non-negative matrix factorization*, in Adv. Neural Inf. Process. Syst. 13, MIT Press, Cambridge, MA, 2001, pp. 556–562.

[33] L.-H. Lim and P. Comon, *Nonnegative approximations of nonnegative tensors*, J. Chemometrics, 23 (2009), pp. 432–441.
[34] C.-J. Lin, *On the convergence of multiplicative update algorithms for nonnegative matrix factorization*, IEEE Trans. Neural Networks, 18 (2007), pp. 1589–1596.
[35] C.-J. Lin, *Projected gradient methods for nonnegative matrix factorization*, Neural Computation, 19 (2007), pp. 2756–2779.
[36] J. Liu, J. Liu, P. Wonka, and J. Ye, *Sparse non-negative tensor factorization using columnwise coordinate descent*, Pattern Recognition, 45 (2012), pp. 649–656.
[37] W. Liu, S. Zheng, S. Jia, L. Shen, and X. Fu, *Sparse nonnegative matrix factorization with the elastic net*, in BIBM2010: Proceedings of the IEEE International Conference on Bioinformatics and Biomedicine, 2010, pp. 265–268.
[38] L. Lucy, *An iterative technique for the rectification of observed distributions*, Astronomical J., 79 (1974), pp. 745–754.
[39] P. McCullagh and J. A. Nelder, *Generalized Linear Models*, 2nd ed., Chapman & Hall, London, 1989.
[40] M. Mørup, L. Hansen, J. Parnas, and S. M. Arnfred, *Decomposing the time-frequency representation of EEG using nonnegative matrix and multi-way factorization*, http://www2.imm.dtu.dk/pubdb/views/edoc_download.php/4144/pdf/imm4144.pdf (2006).
[41] M. Mørup, L. K. Hansen, and S. M. Arnfred, *Algorithms for sparse nonnegative tucker decompositions*, Neural Computation, 20 (2008), pp. 2112–2131.
[42] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer, New York, 1999.
[43] P. Paatero, *A weighted non-negative least squares algorithm for three-way "PARAFAC" factor analysis*, Chemometrics and Intelligent Laboratory Systems, 38 (1997), pp. 223–242.
[44] 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.
[45] P. O. Perry and P. J. Wolfe, *Point Process Modeling for Directed Interaction Networks*, arXiv:1011.1703v1, 2010.
[46] G. Rodríguez, *Poisson models for count data*, in Lecture Notes on Generalized Linear Models, http://data.princeton.edu/wws509/notes (2007).
[47] L. A. Shepp and Y. Vardi, *Maximum likelihood reconstruction for emission tomography*, IEEE Trans. Medical Imaging, 1 (1982), pp. 113–122.
[48] A. Smilde, R. Bro, and P. Geladi, *Multi-Way Analysis: Applications in the Chemical Sciences*, Wiley, West Sussex, UK, 2004.
[49] J. Sun, D. Tao, and C. Faloutsos, *Beyond streams and graphs: Dynamic tensor analysis*, in KDD '06: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM Press, New York, 2006, pp. 374–383.
[50] G. Tomasi and R. Bro, *PARAFAC and missing values*, Chemometrics and Intelligent Laboratory Systems, 75 (2005), pp. 163–180.
[51] Z. Wang, A. Maier, N. K. Logothetis, and H. Liang, *Single-trial decoding of bistable perception based on sparse nonnegative tensor decomposition*, Computational Intelligence and Neuroscience, http://www.hindawi.com/journals/cin/2008/642387/ (2008).
[52] M. Welling and M. Weber, *Positive tensor factorization*, Pattern Recognition Letters, 22 (2001), pp. 1255–1261.
[53] S. Zafeiriou and M. Petrou, *Nonnegative tensor factorization as an alternative Csiszar–Tusnady procedure: Algorithms, convergence, probabilistic interpretations and novel probabilistic tensor latent variable analysis algorithms*, Data Mining and Knowledge Discovery, 22 (2011), pp. 419–466.
[54] W. I. Zangwill, *Nonlinear Programming: A Unified Approach*, International Series in Management, Prentice-Hall, Englewood Cliffs, NJ, 1969.
[55] H. Zhou, D. Alexander, and K. Lange, *A quasi-Newton acceleration for high-dimensional optimization algorithms*, Statist. Comput., 21 (2011), pp. 261–273.
[56] Y. Zhou, M. Goldberg, M. Magdon-Ismail, and A. Wallace, *Strategies for cleaning organizational emails with an application to Enron email dataset*, NAACSOS 07: 5th Conference of North American Association for Computational Social and Organizational Science, 2007.