SANDIA REPORT

SAND2011-1877
Unlimited Release
Printed March 2011

Making Tensor Factorizations Robust to Non-Gaussian Noise

Eric C. Chi and Tamara G. Kolda

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia National Laboratories is a multi-program laboratory managed and 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.

Approved for public release; further dissemination unlimited.

Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government, nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, make any warranty, express or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represent that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone:	(865) 576-8401
Facsimile:	(865) 576-5728
E-Mail:	reports@adonis.osti.gov
Online ordering:	http://www.osti.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone:	(800) 553-6847
Facsimile:	(703) 605-6900
E-Mail:	orders@ntis.fedworld.gov
Online ordering:	http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2011-1877
Unlimited Release
Printed March 2011

Making Tensor Factorizations Robust to Non-Gaussian Noise

Eric C. Chi
Statistics Department
Rice University
Houston, TX 77251-1892
Email: echi@rice.edu

Tamara G. Kolda
Informatics and Systems Assessments Department
Sandia National Laboratories
Livermore, CA 94551-9159
Email: tgkolda@sandia.gov

Abstract

Tensors are multi-way arrays, and the CANDECOMP/PARAFAC (CP) tensor factorization has found application in many different domains. The CP model is typically fit using a least squares objective function, which is a maximum likelihood estimate under the assumption of independent and identically distributed (i.i.d.) Gaussian noise. We demonstrate that this loss function can be highly sensitive to non-Gaussian noise. Therefore, we propose a loss function based on the 1-norm because it can accommodate both Gaussian and grossly non-Gaussian perturbations. We also present an alternating majorization-minimization (MM) algorithm for fitting a CP model using our proposed loss function (CPAL1) and compare its performance to the workhorse algorithm for fitting CP models, CP alternating least squares (CPALS).

Acknowledgments

We thank Grey Ballard, David Gleich, Philip Kegelmeyer, and Todd Plantenga for helpful discussions. Chi's work was fully supported by the U.S. Department of Energy Computational Science Graduate Fellowship. Kolda's work was fully supported by the U.S. Department of Energy Applied Mathematics Program.

Contents

Figures

Algorithms

This page intentionally left blank.

1 Introduction

The CANDECOMP/PARAFAC (CP) tensor factorization [6, 11] can be considered a higher-order generalization of the matrix singular value decomposition and has many applications. The canonical fit function for the CP tensor factorization is based on the least squares error, meaning that it is a maximum likelihood estimate (MLE) under the assumption of additive independent and identically distributed (i.i.d.) Gaussian perturbations. It turns out, however, that this loss function can be sensitive to violations in the Gaussian assumption. This is important to note because many other types of noise are relevant for CP models. For example, in fMRI neuroimaging studies, movement by the subject can lead to sparse high-intensity changes that can be confused with brain activity [9]. Likewise, in foreground/background separation problems in video surveillance, a subject walking across the field of view represents a sparse high intensity change [20]. In both examples, there is a relatively large perturbation in magnitude that affects only a small fraction of data points; we call this *artifact noise*. These scenarios are particularly challenging because the perturbed values are on the same scale as normal values (i.e., true brain activity signals and background pixel intensities). Consequently, there is a need to explore factorization methods that are robust against violations in the Gaussian assumption. In this paper, we consider a loss based on the 1-norm which is known to be robust or insensitive to gross non-Gaussian perturbations [12].

Vorobyov et al. previously described two ways of solving the least 1-norm CP factorization problem based on a linear programming and weighted median filtering [25]. We offer yet another approach based on a majorization-minimization (MM) strategy [13]. Like both methods described in [25] our method performs alternating minimization.

The rest of this paper is organized as follows. Section 2 describes the notation and common operations used throughout the paper. Section 3 reviews probability basics needed to understand maximum likelihood estimation. Section 4 introduces maximum likelihood estimation and discusses several examples. Section 5 frames the CP tensor factorization problem as a maximum likelihood estimation problem. Our robust iterative algorithm - CP Alternating Least 1-norm (CPAL1) - is derived in Section 6. The global convergence of CPAL1 is proven in Section 7. In Section 8 we compare CPAL1 and the standard CP factorizations by alternating least squares (CPALS) in the presence of non-Gaussian perturbations on simulated data. Section 9 discusses related problems and possible extensions. Concluding remarks are given in Section 10.

This page intentionally left blank.

2 Notation and Preliminaries

2.1 Tensors

We use the notation defined in [15]. The *order* of a tensor is the number of dimensions, also known as ways or modes. Vectors (tensors of order one) are denoted by boldface capital letters, e.g., $\mathbf{a}$. All vectors are column vectors. Matrices (tensors of order two) are denoted by boldface capital letters, e.g., $\mathbf{A}$. Higher-order tensors (order three or higher) are denoted by boldface Euler script letters, e.g., $\boldsymbol{\mathscr{X}}$. Scalars are denoted by lowercase letters, e.g., a.

The ith entry of a vector $\mathbf{a}$ is denoted by a_i, element (i,j) of a matrix $\mathbf{A}$ is denoted by a_{ij}, and element (i,j,k) of a third-order tensor $\boldsymbol{\mathscr{X}}$ is denoted by x_{ijk}. Indices typically range from 1 to their capital version, e.g., $i = 1, \ldots, I$. The nth element in a sequence is denoted by a superscript in parentheses, e.g., $\mathbf{A}^{(n)}$ denotes the nth matrix in a sequence. The transpose of the ith row of a matrix $\mathbf{A}$ is denoted by $\mathbf{a}_{[i]}$. The jth column of a matrix $\mathbf{A}$ is denoted by $\mathbf{a}_j$.

Fibers are the higher-order analogue of matrix rows and columns. A fiber is defined by fixing every index but one. *Slices* are two-dimensional sections of a tensor, defined by fixing all but two indices. The mode-n matricization of a tensor $\boldsymbol{\mathscr{X}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is denoted by $\mathbf{X}_{(n)}$ and arranges the mode-n fibers to be the columns of the resulting matrix.

The Khatri-Rao matrix product is important in the sections that follow, so we define it here. First recall that the *Kronecker product* of vectors $\mathbf{a} \in \mathbb{R}^M$ and $\mathbf{b} \in \mathbb{R}^N$, denoted by $\mathbf{a} \otimes \mathbf{b}$, is a vector of length MN and is defined by

$$\mathbf{a} \otimes \mathbf{b} = \begin{bmatrix} a_1\mathbf{b}^{\mathsf{T}} & a_2\mathbf{b}^{\mathsf{T}} & \cdots & a_M\mathbf{b}^{\mathsf{T}} \end{bmatrix}^{\mathsf{T}}$$

The *Khatri-Rao product* [23] is the "matching columnwise" Kronecker product. Given matrices $\mathbf{A} \in \mathbb{R}^{I \times K}$ and $\mathbf{B} \in \mathbb{R}^{J \times K}$, their Khatri-Rao product is denoted by $\mathbf{A} \odot \mathbf{B}$. The result is a matrix of size $(IJ) \times K$ defined by

$$\mathbf{A} \odot \mathbf{B} = \begin{bmatrix} \mathbf{a}_1 \otimes \mathbf{b}_1 & \mathbf{a}_2 \otimes \mathbf{b}_2 & \cdots & \mathbf{a}_K \otimes \mathbf{b}_K \end{bmatrix}.$$

This page intentionally left blank.

3 Review of Probability Basics

Some basic fluency of random variables, probabilities, densities, and probability mass functions will be necessary to understand the subsequent discussion on likelihood functions. A thorough treatment of probability basics and likelihood functions can be found in, e.g., [22].

Random variables will be denoted by capital letters, e.g., X. The observed values they take on will be denoted by lower case letters, e.g., x. The nth random variable in a sequence or set of them is denoted by a subscript, e.g. X_i, and similarly the observed value of the nth random variable is denoted by a subscript, e.g., x_i.

For any random variable, X, its randomness is completely characterized by its associated cumulative distribution function[1] (CDF), denoted by F_X which is defined as

$$F_X(a) = P(X \leq a),$$

i.e., the probability that the random variable X is less than or equal to a. We will be dealing with only two kinds of random variables: continuous and discrete. We say X is a discrete random variable if it can take on at most countably many values; for example, the value of a die toss is a discrete random variable. Without loss of generality, we assume that discrete random variables only take on nonnegative integer values. We say X is a continuous random variable if it can take on uncountably many values.

Example 3.1 If X is distributed uniformly from zero to one, denoted as $X \sim U[0,1]$, then its CDF is

$$F_X(a) = P(X \leq a) = \begin{cases} 0 & \text{if } a < 0 \\ a & \text{if } 0 \leq a \leq 1 \\ 1 & a > 1. \end{cases} \qquad \square$$

The CDF of every continuous random variable can be written as the integral of an almost everywhere (a.e.) unique nonnegative function called the probability density function (PDF)[2]. The PDF is denoted by f_X and is related to the CDF by:

$$F_X(a) = \int_{-\infty}^{a} f_X(x)dx.$$

Example 3.2 If $X \sim U[0,1]$, then its PDF is

$$f_X(x) = \begin{cases} 1 & x \in [0,1] \\ 0 & \text{otherwise.} \end{cases} \qquad \square$$

The CDF of a discrete random variable, X, can be written as the partial sum of a unique nonnegative sequence called the probability mass function (PMF) and denoted by $f_X(k)$. The

[1] Also commonly referred as probability distribution, distribution function, or simply distribution.

[2] Density, density function, and probability density function are synonymous.

PMF is related to CDF by,

$$F_X(a) = \sum_{k=0}^{a} f_X(k)$$

where $f_X(k) = P(X = k)$.

Since the randomness of a random variable is completely characterized by its CDF, and a CDF for a continuous (discrete) random variable has an a.e.-unique PDF (unique PMF), we can also characterize the randomness of a continuous (discrete) random variable by its PDF (PMF).

3.1 Multivariate Probability Distributions

CDFs, PDFs, and PMFs can be defined for collections of random variables, $X_1, \ldots, X_n$. The joint CDF of the collection of random variables $X_1, \ldots, X_n$ is defined to be

$$F_{X_1,\ldots,X_n}(a_1, \ldots, a_n) = P(X_1 \leq a_1, \ldots, X_n \leq a_n).$$

If the variables are continuous, the joint PDF of $X_1, \ldots, X_n$ is defined to be the a.e. unique nonnegative function $f_{X_1,\ldots,X_n}$ such that

$$F_{X_1,\ldots,X_n}(a_1, \ldots, a_n) = \int_{-\infty}^{a_1} \cdots \int_{-\infty}^{a_n} f_{X_1,\ldots,X_n}(x_1, \ldots, x_n) dx_1 \ldots dx_n.$$

If the variables are discrete, the joint PMF is defined to be the unique nonnegative function $f_{X_1,\ldots,X_n}$ such that

$$F_{X_1,\ldots,X_n}(a_1, \ldots, a_n) = \sum_{x_1=0}^{a_1} \cdots \sum_{x_n=0}^{a_n} f_{X_1,\ldots,X_n}(x_1, \ldots, x_n).$$

Given a joint CDF $F_{X_1,\ldots,X_n}$, the marginal CDF of the random variable X_i is denoted F_{X_i} and given by

$$F_{X_i}(a_i) = \lim_{a_1 \to \infty} \cdots \lim_{a_{i-1} \to \infty} \lim_{a_{i+1} \to \infty} \cdots \lim_{a_n \to \infty} F_{X_1,\ldots,X_n}(a_1, \ldots, a_n).$$

If the random variables are continuous then,

$$F_{X_i}(a_i) = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{a_i} \cdots \int_{-\infty}^{\infty} f_{X_1,\ldots,X_n}(x_1, \ldots, x_n) dx_1 \ldots dx_n.$$

If they are discrete, then

$$F_{X_i}(a_i) = \sum_{x_1=0}^{\infty} \cdots \sum_{x_i=0}^{a_i} \cdots \sum_{x_n=0}^{\infty} f_{X_1,\ldots,X_n}(x_1, \ldots, x_n).$$

Definition 3.3 (Independence) *Suppose the random variables $X_1, \ldots, X_n$ have a joint CDF $F_{X_1,\ldots,X_n}$ and each X_i has marginal CDF F_{X_i}. If*

$$F_{X_1,\ldots,X_n}(x_1, \ldots, x_n) = \prod_{i=1}^{n} F_{X_i}(x_i),$$

then $X_1, \dots, X_n$ *are said to be* independent*.*

If the X_i are all continuous (discrete) random variables with joint PDF (PMF) $f_{X_1,\dots,X_n}$ and marginal PDFs (PMFs), f_{X_i}, then they are independent if and only if

$$f_{X_1,\dots,X_n}(x_1, \dots, x_n) = \prod_{i=1}^{n} f_{X_i}(x_i).$$

Definition 3.4 (Independent and Identically Distributed) *We say that* $X_1, \dots, X_n$ *are* independent and identically distributed (i.i.d.) *random variables if they are independent and* $F_{X_1} = F_{X_2} = \cdots = F_{X_n}$*.*

3.2 Common Distributions

We will be using the following PDFs in our tensor factorizations.

Normal or Gaussian

A continuous random variable X has a Normal or Gaussian distribution with mean $\mu \in \mathbb{R}$ and standard deviation $\sigma > 0$, denoted $X \sim N(\mu, \sigma^2)$, if it has the PDF

$$f_X(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right).$$

Laplace or Double exponential

A continuous random variable X has the Laplace or double exponential distribution with mean $\mu \in \mathbb{R}$ and scale parameter $\gamma > 0$, denoted $X \sim$LAPLACE(μ, γ), if it has the PDF

$$f_X(x) = \frac{1}{2\gamma} \exp\left(-\frac{|x-\mu|}{\gamma}\right).$$

Poisson

A discrete random variable X has the Poisson distribution with rate or intensity parameter $\lambda > 0$ denoted $X \sim$ POISSON(λ), if it has the PMF

$$f_X(k) = \begin{cases} \frac{\lambda^k \exp(-\lambda)}{k!} & \text{if } k \in \mathbb{Z}_+ \\ 0 & \text{otherwise.} \end{cases}$$

3.3 Parametric Families

All the distributions given in the previous section are completely specified by a finite number of parameters. A Poisson distribution is specified by its intensity parameter which is a positive real number. So, the set of all Poisson distributions can be indexed by positive reals. The following definition formalizes the idea that families of related distributions can be indexed by a finite number of parameters.

Definition 3.5 (Shao [22]) *A set of CDFs $\{P_\theta\}$ indexed by a parameter $\theta \in \Theta$ is said to be a* parametric family *if $\Theta \subseteq \mathbb{R}^d$ for some fixed positive integer d and each P_θ is a known CDF when θ is known. The set Θ is called the* parameter space *and d is called its* dimension*.*

Note that if a continuous random variable X has a PDF $f_X(\cdot;\theta)$ that is from a parametric family indexed by $\theta \in \Theta$, then we can view $f_X(x;\theta)$ as a mapping from $\mathbb{R} \times \Theta$ into $\mathbb{R}$. If we observe the value of X to be x, then $f_X(x;\theta)$ maps $\theta \in \Theta$ to a point in $\mathbb{R}$. This mapping, $f_X(x|\cdot) : \Theta \to \mathbb{R}$, is called the likelihood function and is denoted by $\ell : \Theta \to \mathbb{R}$. The likelihood function is similarly defined for the discrete random variable with a parametric distribution, viewing its PMF as a function of its parameter given the data.

We are now ready to pose the following question: Given some observations or data and assuming the data has been generated from a parametric family (e.g., the family of all Poisson distributions), which member of the family is most consistent with the data? Since each member of a given parametric family is indexed by its parameter, this problem is referred to as estimation since we are using the data and assumptions about its distribution to estimate the parameter of the distribution that best agrees with the observed distribution of the data. Best can be interpreted in many ways, but we will discuss in the next section one of the most widely used notions of best in statistics: the estimate that maximizes the likelihood.

4 Maximum Likelihood Estimation

Let $\bar{\Theta}$ denote the closure of Θ. Then the maximum likelihood estimate (MLE) is a parameter value $\theta^* \in \bar{\Theta}$ that maximizes the likelihood function

$$\ell(\theta^*) = \max_{\theta \in \bar{\Theta}} \ell(\theta).$$

Example 4.1 (The Sample Mean) Suppose $X_1, \ldots, X_n$ are i.i.d. random variables with shared distribution $N(\mu, 1)$ and observed values $x_1, \ldots, x_n$. We will show that the MLE of μ is the unique solution to a least squares problem. Since the random variables are independent, the likelihood function is the product of the marginal densities of each X_i:

$$\ell(\mu) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}} \exp(-(x_i - \mu)^2/2).$$

Often we will maximize the log of the likelihood, the log-likelihood. Since the log function is monotonically increasing, an argument that a maximizes the log-likelihood is an MLE. Maximizing the log-likelihood turns the product of marginal likelihoods into the sum of marginal log-likelihoods:

$$\begin{aligned}\max_{\mu \in \mathbb{R}} \log \ell(\mu) &= \max_{\mu \in \mathbb{R}} \log \left(\prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}} \exp(-(x_i - \mu)^2/2) \right), \\ &= -\sum_{i=1}^{n} \log(\sqrt{2\pi}) - \frac{1}{2} \min_{\mu \in \mathbb{R}} \sum_{i=1}^{n} (x_i - \mu)^2,\end{aligned}$$

The unique stationary point is the global minimizer and is given by the sample mean:

$$\mu^* = \frac{1}{n} \sum_{i=1}^{n} x_i. \quad \square$$

Example 4.2 (Linear Regression) Building on the previous example, we next show that the ordinary least squares problem is equivalent to a maximum likelihood estimation problem. Suppose $\mathbf{X} \in \mathbb{R}^{n \times p}$ and $Y_1, \ldots, Y_n$ are independent random variables and $Y_i \sim N(\mathbf{x}_{[i]}^{\mathsf{T}} \mathbf{b}, 1)$ where $\mathbf{b} \in \mathbb{R}^p$. We calculate the MLE of $\mathbf{b}$. Since the random variables are independent, the likelihood function is the product of the marginal densities of each Y_i:

$$\ell(\mathbf{b}) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}} \exp(-\left(y_i - \mathbf{x}_{[i]}^{\mathsf{T}} \mathbf{b}\right)^2/2).$$

Maximizing the log-likelihood function yields

$$\begin{aligned}
\max_{\mathbf{b}\in\mathbb{R}^p} \log \ell(\mathbf{b}) &= \max_{\mathbf{b}\in\mathbb{R}^p} \log \left(\prod_{i=1}^n \frac{1}{\sqrt{2\pi}} \exp(- \left(y_i - \mathbf{x}_{[i]}^\mathsf{T}\mathbf{b} \right)^2 /2) \right) \\
&= \max_{\mathbf{b}\in\mathbb{R}^p} \sum_{i=1}^n \left(-\log(\sqrt{2\pi}) - \left(y_i - \mathbf{x}_{[i]}^\mathsf{T}\mathbf{b} \right)^2 /2 \right). \\
&= -\sum_{i=1}^n \log(\sqrt{2\pi}) - \frac{1}{2} \min_{\mathbf{b}\in\mathbb{R}^p} \|\mathbf{y} - \mathbf{X}\mathbf{b}\|^2 .
\end{aligned}$$

If $\mathbf{X}$ is full rank, then the MLE is unique and is given by:

$$\mathbf{b}^* = \left(\mathbf{X}^\mathsf{T}\mathbf{X} \right)^{-1} \mathbf{X}^\mathsf{T}\mathbf{y}.$$

If $\mathbf{X}$ is not full rank, then the MLE is not unique, but the MLE with the least 2-norm is given by:

$$\mathbf{b}^* = \left(\mathbf{X}^\mathsf{T}\mathbf{X} \right)^{\dagger} \mathbf{X}^\mathsf{T}\mathbf{y},$$

where $\left(\mathbf{X}^\mathsf{T}\mathbf{X} \right)^{\dagger}$ denotes the Moore-Penrose pseudoinverse of $\mathbf{X}^\mathsf{T}\mathbf{X}$ [10]. □

Example 4.3 (Principal Components) We next show that the optimization problem behind the principal components decomposition of $\mathbf{X}$ is equivalent to a maximum likelihood estimation problem. Suppose we observe a two-way array of independent random variables $\{X_{ij}\}$ where $i = 1, \ldots, n$; $j = 1, \ldots, p$; and $X_{ij} \sim N(\mathbf{a}_i^\mathsf{T}\mathbf{b}_j, 1)$ with $\mathbf{a}_i, \mathbf{b}_j \in \mathbb{R}^k$, $k = \min(n, p)$ and both $\{\mathbf{a}_i\}$ and $\{\mathbf{b}_j\}$ are sets of pairwise orthogonal vectors. Let x_{ij} denote the observed value of X_{ij}, and let $\mathbf{X}$ denote the matrix of observed values. We calculate the MLE of $(\mathbf{A}, \mathbf{B})$ where the ith column of $\mathbf{A} \in \mathbb{R}^{k\times n}$ is $\mathbf{a}_i$ and the jth column of $\mathbf{B} \in \mathbb{R}^{k\times p}$ is $\mathbf{b}_j$. Let $\mathcal{C}$ denote the set of matrices that have pairwise orthogonal columns. Since the random variables are independent the likelihood function is the product of the marginal densities of each X_{ij}:

$$\ell(\mathbf{A}, \mathbf{B}) = \prod_{i=1}^n \prod_{j=1}^p \frac{1}{\sqrt{2\pi}} \exp(- \left(x_{ij} - \mathbf{a}_i^\mathsf{T}\mathbf{b}_j \right)^2 /2).$$

We then maximize the log-likelihood function:

$$\begin{aligned}
\max_{\mathbf{A},\mathbf{B}\in\mathcal{C}} \log \ell(\mathbf{A}, \mathbf{B}) &= \max_{\mathbf{A},\mathbf{B}\in\mathcal{C}} \log \left(\prod_{i=1}^n \prod_{j=1}^p \frac{1}{\sqrt{2\pi}} \exp(- \left(x_{ij} - \mathbf{a}_i^\mathsf{T}\mathbf{b}_j \right)^2 /2) \right) \\
&= \max_{\mathbf{A},\mathbf{B}\in\mathcal{C}} \sum_{i=1}^n \sum_{j=1}^p \left(-\log(\sqrt{2\pi}) - \left(x_{ij} - \mathbf{a}_i^\mathsf{T}\mathbf{b}_j \right)^2 /2 \right). \\
&= -\sum_{i=1}^n \sum_{j=1}^p \log(\sqrt{2\pi}) - \frac{1}{2} \min_{\mathbf{A},\mathbf{B}\in\mathcal{C}} \left\| \mathbf{X} - \mathbf{A}^\mathsf{T}\mathbf{B} \right\|_F^2 .
\end{aligned}$$

The MLE is not unique, but an MLE can be obtained from the SVD. Let $\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\mathsf{T}$ denote the SVD of $\mathbf{X}$, then an MLE is given by $(\mathbf{A}^*, \mathbf{B}^*) = (\mathbf{U}^\mathsf{T}, \boldsymbol{\Sigma}\mathbf{V}^\mathsf{T})$. □

Example 4.4 (nonnegative matrix factorization) In nonnegative matrix factorization (NNMF), the goal is to approximate a nonnegative matrix by nonnegative components. One way of obtaining an NNMF is by finding a nonnegative factorization that minimizes the generalized Kullback-Liebler divergence loss [18, 19]. We show that that a solution that minimizes a popular loss function for nonnegative matrix factorization is an MLE under a Poisson model for uncertainty. Suppose we observe a two-way array of independent random variables $\{X_{ij}\}$ where $i = 1, \dots, n$; $j = 1, \dots, p$; and X_{ij} ~Poisson($\mathbf{a}_i^\mathsf{T}\mathbf{b}_j$) where $\mathbf{a}_i, \mathbf{b}_j \in \mathbb{R}_+^k$ and $k \le \min(n, p)$. We calculate the MLE of $(\mathbf{A}, \mathbf{B})$ where the ith column of $\mathbf{A} \in \mathbb{R}_+^{k\times n}$ is $\mathbf{a}_i$ and the jth column of $\mathbf{B} \in \mathbb{R}_+^{k\times p}$ is $\mathbf{b}_j$. Since the random variables are independent, the likelihood function is the product of the marginal densities of each X_{ij}:

$$\ell(\mathbf{A}, \mathbf{B}) = \prod_{i=1}^{n}\prod_{j=1}^{p} \frac{(\mathbf{a}_i^\mathsf{T}\mathbf{b}_j)^{x_{ij}} \exp(-\mathbf{a}_i^\mathsf{T}\mathbf{b}_j)}{x_{ij}!}.$$

We then maximize the log-likelihood function:

$$\max \log \ell(\mathbf{A}, \mathbf{B}) = -\sum_{i=1}^{n}\sum_{j=1}^{p} \log x_{ij}! + \max \sum_{i=1}^{n}\sum_{j=1}^{p} \Big\{ (x_{ij} \log(\mathbf{a}_i^\mathsf{T}\mathbf{b}_j) - \mathbf{a}_i^\mathsf{T}\mathbf{b}_j) \Big\},$$

where the $\mathbf{A}$ and $\mathbf{B}$ are required to have strictly positive entries. □

This page intentionally left blank.

5 Tensor Factorizations as Maximum Likelihood Estimates

The CANDECOMP/PARAFAC (CP) tensor factorization [6, 11] can be considered a higher-order generalization of the matrix singular value decomposition (SVD). Just as the SVD of rank R approximates a matrix as the sum of R rank-1 matrices, the CP factorization of rank R approximates a tensor as the sum of R rank-1 tensors. For example, given a 3-way tensor $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$, we seek factor matrices $\mathbf{A} \in \mathbb{R}^{I \times R}, \mathbf{B} \in \mathbb{R}^{J \times R}, \mathbf{C} \in \mathbb{R}^{K \times R}$ such that

$$x_{ijk} \approx \sum_{r=1}^{R} a_{ir} b_{jr} c_{kr}.$$

Following Kolda [14] (see also Kruskal [16]), the CP model can be concisely expressed as

$$\mathcal{X} \approx [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!] \equiv \sum_{r=1}^{R} \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r.$$

A CP factorization is often unique up to a scaling indeterminacy and permutations; see [15] for a review of conditions for uniqueness. The scaling indeterminacy can be addressed by requiring that the columns of $\mathbf{A}, \mathbf{B}$, and $\mathbf{C}$ have unit Euclidean norm and introducing a scaling constant, $\lambda_r \geq 0$, for $r = 1, \ldots, R$. Under these constraints the CP factorization becomes:

$$\mathcal{X} \approx [\![\boldsymbol{\lambda}; \mathbf{A}, \mathbf{B}, \mathbf{C}]\!] \equiv \sum_{r=1}^{R} \lambda_r \; \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r.$$

Determining the rank of a tensor is NP-hard [24] and is beyond the scope of this report; see [15] for discussion on the rank of a tensor. Instead our attention is on how to compute a CP factorization given a rank R. Thus, given a rank R and a tensor $\mathcal{X}$, our goal is to find $\mathbf{A}, \mathbf{B}, \mathbf{C}$, and $\boldsymbol{\lambda}$. Clearly we would like $\sum_{r=1}^{R} \lambda_r a_{ir} b_{jr} c_{kr}$ to be "close" to x_{ijk}. The standard loss function used to measure closeness is the 2-norm of the difference and the "workhorse" algorithm is alternating least squares (CPALS) which was proposed in the original CP papers [6, 11]. Specifically the objective function used in CPALS is

$$\min \sum_{i=1}^{I} \sum_{j=1}^{J} \sum_{k=1}^{K} \left(x_{ijk} - \sum_{r=1}^{R} \lambda_r a_{ir} b_{jr} c_{kr} \right)^2 . \tag{1}$$

For computational expediency, in CPALS the objective function is minimized with respect to one factor matrix at a time holding all other factor matrices constant. Each update of a factor matrix is a least squares problem, hence the name.

Another loss function is given by 1-norm sense. Finding the least 1-norm factorization yields the following problem

$$\min \sum_{i=1}^{I} \sum_{j=1}^{J} \sum_{k=1}^{K} \left| x_{ijk} - \sum_{r=1}^{R} \lambda_r a_{ir} b_{jr} c_{kr} \right| . \tag{2}$$

Both least 1-norm and 2-norm solutions are equivalent to maximum likelihood estimation assuming i.i.d. additive noise. That is

$$x_{ijk} = \sum_{r=1}^{R} \lambda_r a_{ir} b_{jr} c_{kr} + e_{ijk},$$

where e_{ijk} are i.i.d. N(0,1) for the 2-norm and i.i.d. LAPLACE$(0, 1)$ for the 1-norm.

The relationship between the loss function and assumptions about the statistical behavior of the residuals is important because the fitted factorization can be sensitive to differences between the true behavior of the residuals and the assumed behavior.

6 Majorization Solution to ℓ_1 Minimization

MM algorithms have been applied to factorization problems previously [19, 5, 8]. The basic idea of an MM algorithm is to convert a hard optimization problem (e.g., non-convex, non-differentiable) into a series of simpler ones (e.g., smooth convex), which are easier to minimize than the original. To do so, we use majorization functions.

Definition 6.1 *Suppose g and h are real-valued functions on $\mathbb{R}^n$. We say that h* majorizes *g at $\mathbf{x} \in \mathbb{R}^n$ if $h(\mathbf{u}) \geq g(\mathbf{u})$ for all $\mathbf{u} \in \mathbb{R}^n$ and $h(\mathbf{x}) = g(\mathbf{x})$.*

Thus if h majorizes g at some point, then the graph of h lies above the graph of g everywhere except where the graph of h touches the graph of g. Algorithm 1 outlines a simple iterative strategy for finding the minimizer of a function g. It is easy to see that Algorithm 1 always takes non-increasing steps with respect to g for the following reason. Consider the iteration starting at $\mathbf{x}^{(k)}$. Since $\mathbf{x}^{(k+1)}$ minimizes $h(\cdot|\mathbf{x}^{(k)})$, we have

$$g(\mathbf{x}^{(k)}) = h(\mathbf{x}^{(k)}|\mathbf{x}^{(k)}) \geq h(\mathbf{x}^{(k+1)}|\mathbf{x}^{(k)}) \geq g(\mathbf{x}^{(k+1)}).$$

Algorithm 1 Majorization-Minimization for minimizing a function

```
$\mathbf{x}^{(0)} \leftarrow$ random point in $\mathbb{R}^n$
$h(\cdot|\mathbf{x}^{(0)}) \leftarrow$ majorization of $g$ at $\mathbf{x}^{(0)}$.
$k \leftarrow 0$
repeat
  $\mathbf{x}^{(k+1)} \leftarrow \operatorname*{argmin}_{\mathbf{x}} h(\mathbf{x}|\mathbf{x}^{(k)})$
  $h(\cdot|\mathbf{x}^{(k+1)}) \leftarrow$ majorization of $g$ at $\mathbf{x}^{(k+1)}$
  $k \leftarrow k+1$
until convergence
return $\mathbf{x}^{(k+1)}$
```

We now derive a majorization for use in an MM algorithm for ℓ_1 regression as originally suggested in [17] for use in our basic robust tensor factorization introduced in the next section.

Proposition 6.2 (Majorizing the square root function) *Given $\epsilon > 0$, let $g_\epsilon(u) = \sqrt{u + \epsilon}$ for $u \geq 0$. If $\tilde{u} \geq 0$, then the following function majorizes g_ϵ at $\tilde{u}$:*

$$h_\epsilon(u|\tilde{u}) = \sqrt{\tilde{u} + \epsilon} + \frac{u - \tilde{u}}{2\sqrt{\tilde{u} + \epsilon}}.$$

Proof. It is immediate that $g_\epsilon(\tilde{u}) = h_\epsilon(\tilde{u}|\tilde{u})$. We show that the required inequality holds by rearranging a well chosen initial inequality:

$$\begin{aligned}
0 &\leq (\sqrt{u+\epsilon} - \sqrt{\tilde{u}+\epsilon})^2 \\
0 &\leq (u+\epsilon) - 2\sqrt{u+\epsilon}\sqrt{\tilde{u}+\epsilon} + (\tilde{u}+\epsilon), \\
2\sqrt{u+\epsilon}\sqrt{\tilde{u}+\epsilon} &\leq (u+\epsilon) + (\tilde{u}+\epsilon), \\
\sqrt{u+\epsilon} &\leq \frac{(u+\epsilon)+(\tilde{u}+\epsilon)}{2\sqrt{\tilde{u}+\epsilon}}, \\
\sqrt{u+\epsilon} &\leq \frac{(u+\epsilon)-(\tilde{u}+\epsilon)+2(\tilde{u}+\epsilon)}{2\sqrt{\tilde{u}+\epsilon}}, \\
\sqrt{u+\epsilon} &\leq \sqrt{\tilde{u}+\epsilon} + \frac{u-\tilde{u}}{2\sqrt{\tilde{u}+\epsilon}}, \\
g_\epsilon(u) &\leq h_\epsilon(u,\tilde{u}).
\end{aligned}$$

□

6.1 Solving the ℓ_1 Regression Problem by an MM Algorithm

We now show how to use the majorization derived in Proposition 6.2 to perform robust linear regression using the ℓ_1 norm. Given a vector $\mathbf{y} \in \mathbb{R}^I$ and a matrix $\mathbf{M} \in \mathbb{R}^{I \times J}$. We would like to find a vector $\mathbf{u} \in \mathbb{R}^J$ such that $\mathbf{Mu}$ is a good approximation to $\mathbf{y}$ in a least ℓ_1 sense. Recall that $\mathbf{m}_{[i]}$ denotes the transpose of the ith row of $\mathbf{M}$. We search for the vector $\mathbf{u}$ that minimizes the loss:

$$\min_{\mathbf{u}\in\mathbb{R}^J} L(\mathbf{u}) = \min_{\mathbf{u}\in\mathbb{R}^J} \sum_{i=1}^{I} \left| y_i - \mathbf{m}_{[i]}^{\mathsf{T}}\mathbf{u} \right| = \min_{\mathbf{u}\in\mathbb{R}^J} \sum_{i=1}^{I} |r_i(\mathbf{u})|, \tag{3}$$

where $r_i(\mathbf{u}) = y_i - \mathbf{m}_{[i]}^{\mathsf{T}}\mathbf{u}$.

Our MM algorithm will minimize the following smooth approximation to $L(\mathbf{u})$:

$$L_\epsilon(\mathbf{u}) = \sum_{i=1}^{I} \sqrt{r_i(\mathbf{u})^2 + \epsilon},$$

where ϵ is a small positive number. We apply this approximation to curb potential numerical instability if a residual gets close to zero. There is little downside from a robustness perspective to making this approximation. The 1-norm loss is statistically robust because it is less influenced by unusually large residuals, and $L_\epsilon(\mathbf{u})$ is a good approximation of $L(\mathbf{u})$ for large values of $r_i(\mathbf{u})$. We now use Proposition 6.2 to derive a majorization for $L_\epsilon(\mathbf{u})$.

Proposition 6.3 *The function $L_\epsilon(\mathbf{u})$ is majorized at a point $\tilde{\mathbf{u}} \in \mathbb{R}^J$ by the function*

$$h_\epsilon(\mathbf{u}|\tilde{\mathbf{u}}) = \sum_{i=1}^{I} \left\{ \sqrt{r_i(\tilde{\mathbf{u}})^2 + \epsilon} + \frac{r_i(\mathbf{u})^2 - r_i(\tilde{\mathbf{u}})^2}{2\sqrt{r_i(\tilde{\mathbf{u}})^2 + \epsilon}} \right\}. \tag{4}$$

The proof follows immediately from Proposition 6.2.

The majorization in equation (4) allows us to solve the minimization problem (3) by solving a sequence of weighted least squares problems. Given the mth iterate, $\mathbf{u}^{(m)}$, the next iterate, $\mathbf{u}^{(m+1)}$, is set to be the minimizer of the majorization of $L_\epsilon(\mathbf{u})$ at $\mathbf{u}^{(m)}$:

$$\mathbf{u}^{(m+1)} = \underset{\mathbf{u}}{\operatorname{argmin}} \sum_{i=1}^{I} \left\{ \sqrt{r_i(\mathbf{u}^{(m)})^2 + \epsilon} + \frac{r_i(\mathbf{u})^2 - r_i(\mathbf{u}^{(m)})^2}{2\sqrt{r_i(\mathbf{u}^{(m)})^2 + \epsilon}} \right\}. \tag{5}$$

Note that only terms that depend on $\mathbf{u}$ are relevant to the minimization in equation (5); thus

$$\mathbf{u}^{(m+1)} = \underset{\mathbf{u}}{\operatorname{argmin}} \sum_{i=1}^{I} \frac{r_i(\mathbf{u})^2}{\sqrt{r_i(\mathbf{u}^{(m)})^2 + \epsilon}}. \tag{6}$$

Let $\mathbf{W}^{(m)} \in \mathbb{R}^{I \times I}$ be the diagonal matrix defined by $(\mathbf{W}^{(m)})_{ii} = 1/\sqrt{r_i(\mathbf{u}^{(m)})^2 + \epsilon}$ for $i = 1, \dots, I$. Then the minimization problem (6) is a weighted least squares problem with weight matrix $\mathbf{W}^{(m)}$:

$$\mathbf{u}^{(m+1)} = \underset{\mathbf{u}}{\operatorname{argmin}}\, (\mathbf{y} - \mathbf{M}\mathbf{u})^{\mathsf{T}} \mathbf{W}^{(m)} (\mathbf{y} - \mathbf{M}\mathbf{u}). \tag{7}$$

The minimization problem (7) will have multiple solutions if $\mathbf{M}$ is not full rank. There is a unique solution with the least ℓ_2 norm among all solutions, however, and it is given by

$$\mathbf{u}^{(m+1)} = (\mathbf{M}^{\mathsf{T}} \mathbf{W}^{(m)} \mathbf{M})^{\dagger} \mathbf{M}^{\mathsf{T}} \mathbf{W}^{(m)} \mathbf{y}. \tag{8}$$

The relationship between the ℓ_1 regression fit and ordinary least squares (OLS) regression fit can be understood through the effect of the weight matrix $\mathbf{W}$. If the ith data point has a poor fit in iteration m, then $r_i(\mathbf{u}^{(m)})^2$ will large and its contribution to the optimization problem at iteration $m+1$ will be downweighted by the factor $1/\sqrt{r_i(\mathbf{u}^{(m)})^2 + \epsilon}$. Conversely the contributions of data points with small residuals, or good fits, in the mth iteration will be upweighted in the optimization problem at iteration $m+1$ by a factor of $1/\sqrt{r_i(\mathbf{u}^{(m)})^2 + \epsilon}$. In contrast, OLS can be thought of as a weighted least squares problem with the identity matrix as the weight matrix. Since OLS weights all residuals equally, the resulting fit can be significantly influenced by a single outlier.

6.2 Robust Tensor Factorization with Alternating Least ℓ_1 Regression

We now show how the preceding ℓ_1 regression solution can be used to perform robust CP factorization. Consider a 3-way tensor, $\boldsymbol{\mathcal{X}}$, of dimensions I_1, I_2, and I_3. We propose to perform a robust CP factorization of rank R by minimizing the total ℓ_1 loss, $L(\mathbf{A}, \mathbf{B}, \mathbf{C}, \boldsymbol{\lambda})$, where $\mathbf{A} \in \mathbb{R}^{I_1 \times R}, \mathbf{B} \in \mathbb{R}^{I_2 \times R}$, and $\mathbf{C} \in \mathbb{R}^{I_3 \times R}$ have columns with unit Euclidean norm, and $\boldsymbol{\lambda} \in \mathbb{R}^R_+$ and

$$L(\mathbf{A}, \mathbf{B}, \mathbf{C}, \boldsymbol{\lambda}) = \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \sum_{i_3=1}^{I_3} \left| x_{i_1 i_2 i_3} - \sum_{r=1}^{R} \lambda_r a_{i_1 r} b_{i_2 r} c_{i_3 r} \right|.$$

This minimization problem is nonlinear and non-convex. Instead of simultaneously solving for $\mathbf{A}, \mathbf{B}$, and $\mathbf{C}$, we can instead resort to minimizing $\mathbf{A}, \mathbf{B}$, and $\mathbf{C}$ in turn while holding the other two fixed. This is an ℓ_1 equivalent version of alternating least squares (CPALS), which we call CP alternating least 1-norm (CPAL1). Each subproblem is a convex minimization problem. Let us consider the subproblem where $\mathbf{B}$ and $\mathbf{C}$ are fixed in the alternating ℓ_1 minimization. The mode-1 matricization of the rank-R approximation $[\![\boldsymbol{\lambda}; \mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$ is $\hat{\mathbf{A}}(\mathbf{C} \odot \mathbf{B})^{\mathsf{T}}$ where $\hat{\mathbf{A}} = \mathbf{A} \cdot \operatorname{diag}(\boldsymbol{\lambda})$; see [15]. Thus, we search for $\hat{\mathbf{A}}$ that minimizes

$$\|\mathbf{X}_{(1)} - \hat{\mathbf{A}}(\mathbf{C} \odot \mathbf{B})^{\mathsf{T}}\|_{\ell_1}$$

where

$$\|\mathbf{M}\|_{\ell_1} = \sum_{ij} |m_{ij}|.$$

Upon finding such an $\hat{\mathbf{A}}$, we normalize the columns of $\hat{\mathbf{A}}$ to get $\mathbf{A}$; in other words, let $\lambda_r = \|\hat{\mathbf{a}}_r\|$ and $\mathbf{a}_r = \hat{\mathbf{a}}_r / \lambda_r$.

We proceed using the mode-1 matricization. Let $\mathbf{Z} = \mathbf{X}_{(1)}$ and $\mathbf{Q} = \mathbf{C} \odot \mathbf{B}$. So, $\mathbf{Z} \in \mathbb{R}^{I_1 \times I_2 I_3}$ and $\mathbf{Q} \in \mathbb{R}^{I_2 I_3 \times R}$. Recall that $\hat{\mathbf{a}}_{[i]} \in \mathbb{R}^R$ and $\mathbf{q}_{[j]} \in \mathbb{R}^R$ denote the transposes of the ith row of $\hat{\mathbf{A}}$ and the jth row of $\mathbf{Q}$ respectively. The minimization can then be expressed as

$$\min_{\hat{\mathbf{A}}} \sum_{i=1}^{I_1} \sum_{j=1}^{I_2 I_3} \left| z_{ij} - \mathbf{q}_{[j]}^{\mathsf{T}} \hat{\mathbf{a}}_{[i]} \right|.$$

This minimization problem is separable by rows. So, for each row i of $\hat{\mathbf{A}}$, we minimize

$$L(\hat{\mathbf{a}}_{[i]}) = \sum_{j=1}^{I_2 I_3} \left| z_{ij} - \mathbf{q}_{[j]}^{\mathsf{T}} \hat{\mathbf{a}}_{[i]} \right|. \tag{9}$$

Note that this minimization problem is a least ℓ_1 regression problem where we are looking for the best linear predictor $\mathbf{Q}\hat{\mathbf{a}}_{[i]}$ of $\mathbf{z}_{[i]}$ in a least ℓ_1 sense. Compare equation (9) with equation (3) and note that $n = I_2 I_3, \mathbf{y} = \mathbf{z}_{[i]}, \mathbf{M} = \mathbf{Q}$, and $\mathbf{u} = \hat{\mathbf{a}}_{[i]}$.

We solve the neighboring problem with small $\epsilon > 0$, minimizing the following loss

$$L_\epsilon(\hat{\mathbf{a}}_{[i]}) = \sum_{j=1}^{I_2 I_3} \sqrt{\left(z_{ij} - \mathbf{q}_{[j]}^{\mathsf{T}} \hat{\mathbf{a}}_{[i]} \right)^2 + \epsilon},$$

and apply the iterative update rule derived in equation (8) from section 6.1, to get the following iterative update for the ith row of $\hat{\mathbf{A}}$

$$\hat{\mathbf{a}}_{[i]}^{(m+1)} = (\mathbf{Q}^{\mathsf{T}} \mathbf{W}_i^{(m)} \mathbf{Q})^{\dagger} \mathbf{Q}^{\mathsf{T}} \mathbf{W}_i^{(m)} \mathbf{z}_{[i]},$$

where $\mathbf{W}_i^{(m)} \in \mathbb{R}^{I_2 I_3 \times I_2 I_3}$ is a diagonal matrix whose jth diagonal entry is the inverse of the jth residual from the mth iteration for the ith row regression, i.e.,

$$\left(\mathbf{W}_i^{(m)} \right)_{jj} = \left[\left(z_{ij} - \mathbf{q}_{[j]}^{\mathsf{T}} \hat{\mathbf{a}}_{[i]}^{(m)} \right)^2 + \epsilon \right]^{-\frac{1}{2}}.$$

We have I_1 independent ℓ_1 regressions to solve to obtain one iterate of $\mathbf{A}$. Similarly we wil have I_2 independent ℓ_1 regressions to solve to obtain one iterate of $\mathbf{B}$, and I_3 independent ℓ_1 regressions

Algorithm 2 CP using alternating 1-norm minimization (CPAL1)

```
 1: initialize $\mathbf{A}^{(n)} \in \mathbb{R}^{I_n \times R}$ for $n = 1, \ldots, N$.
 2: repeat
 3:   for $n = 1, \ldots, N$ do
 4:     repeat
 5:       $\mathbf{Q} \leftarrow \left(\mathbf{A}^{(N)} \odot \cdots \odot \mathbf{A}^{(n+1)} \odot \mathbf{A}^{(n-1)} \odot \cdots \odot \mathbf{A}^{(1)}\right)^{\ddagger}$
 6:       $\mathbf{Z} \leftarrow \mathbf{X}_{(n)}$
 7:       for $i = 1, \ldots, I_i$ do
 8:         for $j = 1, \ldots, \prod_{k \neq i} I_k$ do
 9:           $w_j \leftarrow \left[\left(z_{ij} - \mathbf{q}_{[j]}^{\mathsf{T}} \mathbf{a}_{[i]}^{(n)}\right)^2 + \epsilon\right]^{-\frac{1}{2}}$
10:         end for
11:         $\mathbf{W} \leftarrow \mathrm{diag}(\mathbf{w})$
12:         $\mathbf{a}_{[i]}^{(n)} \leftarrow (\mathbf{Q}^{\mathsf{T}} \mathbf{W} \mathbf{Q})^{\dagger} \mathbf{Q}^{\mathsf{T}} \mathbf{W} \mathbf{z}_{[i]}$
13:       end for
14:     until fit for $\mathbf{A}^{(n)}$ ceases to improve or maximum iterations exhausted
15:     normalize columns of $\mathbf{A}^{(n)}$ (storing norms as $\boldsymbol{\lambda}$)
16:   end for
17: until fit ceases to improve or maximum iterations exhausted
18: return $\boldsymbol{\lambda}, \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \ldots, \mathbf{A}^{(N)}$
```

to solve to obtain one iterate of **C**. Algorithm 2 outlines pseudocode for the CPAL1 decomposition with R components for the Nth-order tensor $\boldsymbol{\mathcal{X}}$ of size $I_1 \times I_2 \times \cdots I_N$. The symbol ‡ denotes the transpose pseudoinverse. Note that in line 5 the transpose pseudoinverse has a special form which facilitates efficient computation. See [15] for details.

This page intentionally left blank.

7 Global Convergence

We prove that iterates generated by a Tychonoff regularized version of CPAL1 globally converge to stationary points of the penalized loss. The proof relies on piecing together global convergence results for the nonlinear Gauss-Seidel (GS) method and for MM algorithms. We begin by reviewing the GS method. Let $f : \mathbb{R}^n \to \mathbb{R}$ and partition the vector $\mathbf{x} \in \mathbb{R}^n$ so that

$$\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m),$$

where $\mathbf{x}_i \in \mathbb{R}^{n_i}$ and $n_1 + n_2 + \cdots + n_m = n$. Suppose for every $\mathbf{x} \in \mathbb{R}^n$ and every $i = 1, \dots, m$, the optimization problem

$$\begin{aligned}&\text{minimize } f(\mathbf{x}_1, \dots, \mathbf{x}_{i-1}, \mathbf{u}, \mathbf{x}_{i+1}, \dots, \mathbf{x}_m)\\ &\text{subject to } \mathbf{u} \in \mathbb{R}^{n_i},\end{aligned}$$

has at least one solution. Then the GS method generates a sequence of iterates $\{\mathbf{x}^{(k)}\}$ with the following update rule for the ith coordinate block

$$\mathbf{x}_i^{(k+1)} \in \underset{\mathbf{u}\in\mathbb{R}^{n_i}}{\operatorname{argmin}} f(\mathbf{x}_1^{(k+1)}, \dots, \mathbf{x}_{i-1}^{(k+1)}, \mathbf{u}, \mathbf{x}_{i+1}^{(k)}, \dots, \mathbf{x}_m^{(k)}), \tag{10}$$

where $\mathbf{x}^{(k)} = (\mathbf{x}_1^{(k)}, \dots, \mathbf{x}_m^{(k)})$. The following proposition gives conditions for convergence of the GS iterates.

Proposition 7.1 (Proposition 2.7.1 in [4]) *Suppose that f is continuously differentiable on $\mathbb{R}^n$. Furthermore, suppose that for each i and $\mathbf{x} \in \mathbb{R}^n$, the minimum below*

$$\min_{\mathbf{u}\in\mathbb{R}^{n_i}} f(\mathbf{x}_1, \dots, \mathbf{x}_{i-1}, \mathbf{u}, \dots, \mathbf{x}_m)$$

is uniquely attained. Let $\{\mathbf{x}^{(k)}\}$ be the sequence generated by the GS method (10). Then, every limit point of $\{\mathbf{x}^{(k)}\}$ is a stationary point.

If each GS block coordinate update (10) is accomplished through an MM algorithm, as in CPAL1, then we need to ensure that the MM algorithm has a *unique* global minimizer for the block minimization problem. The following proposition gives conditions under which this is true.

Proposition 7.2 (Proposition 15.4.3 in [17]) *Let $g : \mathbb{R}^p \to \mathbb{R}$ denote the objective function to be minimized and $h(\mathbf{x}|\mathbf{y})$ be a majorization of g at the point $\mathbf{y}$. Suppose g is continuously differentiable, coercive in the sense that all its level sets are compact, and is strictly convex. Suppose further that $h(\mathbf{x}|\mathbf{y})$ is jointly twice continuously differentiable in $(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^{2p}$ and is strictly convex in $\mathbf{x}$ with $\mathbf{y}$ fixed. Let $\{\mathbf{x}^{(k)}\}$ denote a set of iterates generated by the MM algorithm*

$$\mathbf{x}^{(k+1)} = \underset{\mathbf{u}\in\mathbb{R}^p}{\operatorname{argmin}}\, h(\mathbf{u}|\mathbf{x}^{(k)}).$$

Then $\lim_{k\to\infty} \mathbf{x}^{(k)}$ exists and is the unique global minimizer of g.

Recall that a given block coordinate minimization problem for updating a factor matrix in CPAL1 can be solved row-by-row by minimizing the following loss function

$$L_\epsilon(\hat{\mathbf{a}}) = \sum_{j=1}^{I_2 I_3} \sqrt{\left(z_{ij} - \mathbf{q}_{[j]}^{\mathsf{T}}\hat{\mathbf{a}}\right)^2 + \epsilon}.$$

The loss function $L_\epsilon(\hat{\mathbf{a}})$ does not meet all the conditions specified in Proposition 7.2. Specifically, while $L_\epsilon(\hat{\mathbf{a}})$ is convex, it cannot be guaranteed to be coercive and strictly convex in $\hat{\mathbf{a}}$. The following proposition, however, shows that coerciveness and strict convexity can be enforced through Tychonoff regularization.

Proposition 7.3 *If $f : \mathbb{R}^p \to \mathbb{R}$ is convex and differentiable then $g(\mathbf{x}) = f(\mathbf{x}) + \mu\|\mathbf{x}\|^2$ is strictly convex and coercive for any $\mu > 0$.*

Proof. It is immediate that g is strictly convex from the definition of strict convexity. We show that g is coercive. Fix arbitrary points $\mathbf{x}, \mathbf{z} \in U$ and $\mathbf{x} \neq \mathbf{z}$. Since f is differentiable and convex

$$f(\mathbf{z}) \geq f(\mathbf{x}) + df(\mathbf{x})(\mathbf{z} - \mathbf{x}).$$

Adding $\mu\|\mathbf{z}\|^2$ to both sides we get

$$g(\mathbf{z}) \geq f(\mathbf{x}) + df(\mathbf{x})(\mathbf{z} - \mathbf{x}) + \mu\|\mathbf{z}\|^2.$$

Therefore, $g(\mathbf{z}) \to \infty$ when $\|\mathbf{z}\| \to \infty$. Thus, for all $\alpha \in \mathbb{R}$, the corresponding level set $\{\mathbf{x} | g(\mathbf{x}) \leq \alpha\}$ is compact. □

We now describe the regularized version of CPAL1 for a three-way tensor. Let $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$. Let $\boldsymbol{\lambda} \in \mathbb{R}_+^R$, and $\mathbf{A} \in \mathbb{R}^{I_1 \times R}, \mathbf{B} \in \mathbb{R}^{I_2 \times R}$, and $\mathbf{C} \in \mathbb{R}^{I_3 \times R}$ have columns with unit Euclidean norm. Fix $\epsilon, \mu > 0$. We wish to minimize the Tychonoff penalized loss

$$L_{\epsilon,\mu}(\mathbf{A}, \mathbf{B}, \mathbf{C}, \boldsymbol{\lambda}) = L_\epsilon(\mathbf{A}, \mathbf{B}, \mathbf{C}, \boldsymbol{\lambda}) + \frac{\mu}{2}\|\boldsymbol{\lambda}\|^2 \tag{11}$$

where

$$L_\epsilon(\mathbf{A}, \mathbf{B}, \mathbf{C}, \boldsymbol{\lambda}) = \sum_{i_1=1}^{I_1}\sum_{i_2=1}^{I_2}\sum_{i_3=1}^{I_3} \sqrt{\left(x_{i_1 i_2 i_3} - \sum_{r=1}^{R} \lambda_r a_{i_1 r} b_{i_2 r} c_{i_3 r}\right)^2 + \epsilon}.$$

Consider the following alternating algorithm. Generate iterates using the GS method. The first block minimization consists of fixing $\mathbf{B}$ and $\mathbf{C}$ and searching for $\hat{\mathbf{A}}$ that minimizes

$$\sum_{i_1=1}^{I_1} \left\{ \sum_{j=1}^{I_2 I_3} \sqrt{\left(z_{ij} - \mathbf{q}_{[j]}^{\mathsf{T}}\hat{\mathbf{a}}_{[i]}\right)^2 + \epsilon} + \frac{\mu}{2}\|\hat{\mathbf{a}}_{[i]}\|_2^2 \right\},$$

where $\mathbf{Z} = \mathbf{X}_{(1)}$ and $\mathbf{Q} = \mathbf{C} \odot \mathbf{B}$. As in the unregularized version, this problem is separable in the rows of $\hat{\mathbf{A}}$. For each i we use an MM algorithm to update the estimate for the ith row with the following majorization

$$h(\hat{\mathbf{a}}|\tilde{\mathbf{a}}) = \sum_{j=1}^{I_2 I_3} \left\{ \sqrt{r_j(\tilde{\mathbf{a}})^2 + \epsilon} + \frac{r_j(\hat{\mathbf{a}})^2 - r_j(\tilde{\mathbf{a}})^2}{2\sqrt{r_j(\tilde{\mathbf{a}})^2 + \epsilon}} \right\} + \frac{\mu}{2}\|\hat{\mathbf{a}}\|^2, \tag{12}$$

where $r_j(\hat{\mathbf{a}}) = z_j - \mathbf{q}_{[j]}^{\mathsf{T}}\hat{\mathbf{a}}$. After obtaining $\hat{\mathbf{A}}$, and consequently $\mathbf{A}$ and $\boldsymbol{\lambda}$, we update the other factors analogously in cyclic order until convergence. We now prove the global convergence of a Tychonoff regularized version of CPAL1 for a three-way tensor. The proof for higher order tensors is essentially the same. The proof does not depend on the algorithm used to solve the subproblems provided the subproblems are solved to optimality.

Proposition 7.4 *Perform the GS method with (11) as the objective function updating one factor matrix at a time holding the other two fixed. Then the limit points of the sequence of factorizations are stationary points of (11). Moreover, every sequence has at least one limit point.*

Proof.

Note that $L_{\epsilon,\mu}$ can be viewed as a function of the rows of $\mathbf{A}, \mathbf{B}$, and $\mathbf{C}$ and consequently we seek to apply Proposition 7.1. Since $L_{\epsilon,\mu}$ is a continuously differentiable function, we just need to show the block minimization problem has a unique solution. Fixing $\mathbf{B}$ and $\mathbf{C}$ and we search for $\hat{\mathbf{A}}$ that minimizes

$$\sum_{i=1}^{I_1}\sum_{j=1}^{I_2 I_3} \sqrt{\left(z_{ij} - \mathbf{q}_{[j]}^{\mathsf{T}}\hat{\mathbf{a}}_{[i]}\right)^2 + \epsilon} + \frac{\mu}{2}\|\hat{\mathbf{a}}_{[i]}\|_2^2.$$

We focus on calculating the update for $\hat{\mathbf{A}}_{[i]}$, the ith row of $\hat{\mathbf{A}}$. For readability we suppress the subscripts denoting the row index. Let $g(\hat{\mathbf{a}}) = f(\hat{\mathbf{a}}) + \frac{1}{2}\mu\|\hat{\mathbf{a}}\|^2$ where $f(\hat{\mathbf{a}}) = \sum_{j=1}^{I_2 I_3}\sqrt{(z_j - \mathbf{q}_{[j]}^{\mathsf{T}}\hat{\mathbf{a}})^2 + \epsilon}$. Thus, it suffices to show that g has a unique global minimum.

First, note that f is convex. The gradient of f is given by

$$\nabla f(\hat{\mathbf{a}}) = -\sum_{j=1}^{I_2 I_3} \frac{(z_j - \mathbf{q}_{[j]}^{\mathsf{T}}\hat{\mathbf{a}})}{\sqrt{(z_j - \mathbf{q}_{[j]}^{\mathsf{T}}\hat{\mathbf{a}})^2 + \epsilon}}\mathbf{q}_{[j]}.$$

The Hessian, $\nabla^2 f(\hat{\mathbf{a}})$ is given by

$$\begin{aligned}
\nabla^2 f(\hat{\mathbf{a}}) &= \sum_{j=1}^{I_2 I_3} \frac{\sqrt{(z_j - \mathbf{q}_{[j]}^{\mathsf{T}}\hat{\mathbf{a}})^2 + \epsilon} - \frac{(z_j - \mathbf{q}_{[j]}^{\mathsf{T}}\hat{\mathbf{a}})^2}{\sqrt{(z_j - \mathbf{q}_{[j]}^{\mathsf{T}}\hat{\mathbf{a}})^2 + \epsilon}}}{(z_j - \mathbf{q}_{[j]}^{\mathsf{T}}\hat{\mathbf{a}})^2 + \epsilon}\mathbf{q}_{[j]}\mathbf{q}_{[j]}^{\mathsf{T}} \\
&= \epsilon \sum_{j=1}^{I_2 I_3} \left[(z_j - \mathbf{q}_{[j]}^{\mathsf{T}}\hat{\mathbf{a}})^2 + \epsilon\right]^{-3/2} \mathbf{q}_{[j]}\mathbf{q}_{[j]}^{\mathsf{T}} \\
&= \epsilon\mathbf{Q}^{\mathsf{T}}\mathbf{W}\mathbf{Q},
\end{aligned}$$

where $\mathbf{W}$ is diagonal and $w_{ii} = \left[(z_j - \mathbf{q}_{[j]}^{\mathsf{T}}\hat{\mathbf{a}})^2 + \epsilon\right]^{-3/2}$. Therefore, $\nabla^2 f(\hat{\mathbf{a}})$ is positive-semidefinite for all $\hat{\mathbf{a}} \in \mathbb{R}^p$, and consequently f is convex. By Proposition 7.3 g is coercive and strictly convex and therefore has a unique global minimum.

Algorithm 3 CP using alternating regularized 1-norm minimization (CPAL1)

```
1: initialize $\mathbf{A}^{(n)} \in \mathbb{R}^{I_n \times R}$ for $n = 1, \ldots, N$.
2: repeat
3:   for $n = 1, \ldots, N$ do
4:     repeat
5:       $\mathbf{Q} \leftarrow \left(\mathbf{A}^{(N)} \odot \cdots \odot \mathbf{A}^{(n+1)} \odot \mathbf{A}^{(n-1)} \odot \cdots \odot \mathbf{A}^{(1)}\right)^{\ddagger}$
6:       $\mathbf{Z} \leftarrow \mathbf{X}_{(n)}$
7:       for $i = 1, \ldots, I_i$ do
8:         for $j = 1, \ldots, \prod_{k \neq i} I_k$ do
9:           $w_j \leftarrow \left[\left(z_{ij} - \mathbf{q}_{[j]}^{\mathsf{T}} \mathbf{a}_{[i]}^{(n)}\right)^2 + \epsilon\right]^{-\frac{1}{2}}$
10:        end for
11:        $\mathbf{W} \leftarrow \mathrm{diag}(\mathbf{w})$
12:        $\mathbf{a}_{[i]}^{(n)} \leftarrow (\mathbf{Q}^{\mathsf{T}}\mathbf{W}\mathbf{Q} + \mu\mathbf{I})^{-1}\mathbf{Q}^{\mathsf{T}}\mathbf{W}\mathbf{z}_{[i]}$
13:      end for
14:    until fit for $\mathbf{A}^{(n)}$ ceases to improve or maximum iterations exhausted
15:    normalize columns of $\mathbf{A}^{(n)}$ (storing norms as $\boldsymbol{\lambda}$)
16:  end for
17: until fit ceases to improve or maximum iterations exhausted
18: return $\boldsymbol{\lambda}, \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \ldots, \mathbf{A}^{(N)}$
```

By Proposition 7.1 the limit points of the sequence of iterates generated by the Tychonoff regularized CPAL1 algorithm are stationary points. Furthermore, every sequence has at least one limit point since $L_{\epsilon,\mu}$ is coercive. This is true because L_ϵ is bounded below and the sum of a coercive function and a continuous function that is bounded below is coercive. □

Finally we use Proposition 7.2 to show that the subproblem can be solved to optimality with an MM algorithm using the majorization (12). Note that g is continuously differentiable. Since $r_j(\hat{\mathbf{a}})^2$ is convex in $\hat{\mathbf{a}}$, by Proposition 7.3, $h(\hat{\mathbf{a}}|\tilde{\mathbf{a}})$ is a strictly convex function of $\hat{\mathbf{a}}$ with $\tilde{\mathbf{a}}$ fixed. Lastly note that $h(\mathbf{a}|\tilde{\mathbf{a}})$ is twice differentiable with respect to $(\mathbf{a}, \tilde{\mathbf{a}})$. To see this, recall that if two functions are twice continuously differentiable, their product, ratio, and sum are twice continuously differentiable. If $\psi : \mathbb{R} \to \mathbb{R}$ and $\varphi : \mathbb{R}^n \to \mathbb{R}$ are twice continuously differentiable then the composition $\psi \circ \varphi$ is twice continuously differentiable.

With regularization the update for the i^{th} row at iteration $m + 1$ is only slightly different from the original update rule. Algorithm 3 outlines the pseudocode for the regularized CPAL1 decomposition with R components for the Nth order tensor $\boldsymbol{\mathcal{X}}$ of size $I_1 \times I_2 \times \cdots \times I_N$. Note that Algorithm 3 differs from Algorithm 2 only in the row estimate in line 14

$$\hat{\mathbf{a}}_{[i]}^{(m+1)} = (\mathbf{Q}^{\mathsf{T}}\mathbf{W}_i^{(m)}\mathbf{Q} + \mu\mathbf{I})^{-1}\mathbf{Q}^{\mathsf{T}}\mathbf{W}_i^{(m)}\mathbf{z}_{[i]}.$$

For the rest of the paper, CPAL1 will refer to the regularized version outlined in Algorithm 3.

8 Simulations

We compare CPAL1 (from Algorithm 2) with CPALS (as implemented in the Tensor Toolbox [2]) on two sets of simulated data. The purpose of the first set is to demonstrate the qualitative differences in solutions between CPAL1 and CPALS. The purpose of the second set is to quantitatively assess those differences.

8.1 Qualitative Comparisons between CPALS and CPAL1

We construct a 3-way tensor $\mathcal{X}$ with dimensions $25 \times 25 \times 25$ and rank two as follows. Let $\phi : \mathbb{R}^n \rightarrow \mathbb{R}^n$ denote the mapping $\phi(\mathbf{a}) = (\phi_1(a_1), \dots, \phi_n(a_n))$ where ϕ_i is the standard univariate Gaussian PDF for all i. We choose $\sigma_1 = 0.75, \sigma_2 = 0.5, \sigma_3 = 0.25, \sigma_4 = 0.1$. Define $\boldsymbol{\omega} \in \mathbb{R}^{25}$ with $\omega_i = -1 + (i-1)/12$. Then we set

$$\begin{aligned}
\mathbf{a}_1 &= \frac{1}{\sigma_1}\phi\left(\frac{\boldsymbol{\omega}}{\sigma_1}\right), \\
\mathbf{a}_2 &= \frac{1}{2}\frac{1}{\sigma_2}\phi\left(\frac{\boldsymbol{\omega}-\mathbf{1}}{\sigma_2}\right) + \frac{1}{2}\frac{1}{\sigma_2}\phi\left(\frac{\boldsymbol{\omega}+\mathbf{1}}{\sigma_2}\right), \\
\mathbf{b}_1 &= \frac{1}{\sigma_1}\phi\left(\frac{\boldsymbol{\omega}}{\sigma_1}\right), \text{and} \\
\mathbf{b}_2 &= \frac{1}{4}\frac{1}{\sigma_3}\phi\left(\frac{\boldsymbol{\omega}-\mathbf{1}}{\sigma_3}\right) + \frac{1}{2}\frac{1}{\sigma_4}\phi\left(\frac{\boldsymbol{\omega}}{\sigma_4}\right) + \frac{1}{4}\frac{1}{\sigma_3}\phi\left(\frac{\boldsymbol{\omega}+\mathbf{1}}{\sigma_3}\right).
\end{aligned}$$

For $i = 1, \dots, 25$ the elements of $\mathbf{C}$ are set to be $c_{1,i} = (i-1)/24$ and $c_{2,i} = 1 - (i-1)/24$. The columns of $\mathbf{A}$ and $\mathbf{B}$ are the values of mixtures of scaled and shifted Gaussian PDFs evaluated over 25 evenly spaced points between -1 and 1. We set $\mathcal{X} = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$. The first frontal slice is $\mathbf{X}_1 = \mathbf{a}_1 \circ \mathbf{b}_1$; the last frontal slice is $\mathbf{X}_{25} = \mathbf{a}_2 \circ \mathbf{b}_2$; and all other frontal slices are convex combinations of $\mathbf{X}_1$ and $\mathbf{X}_{25}$.

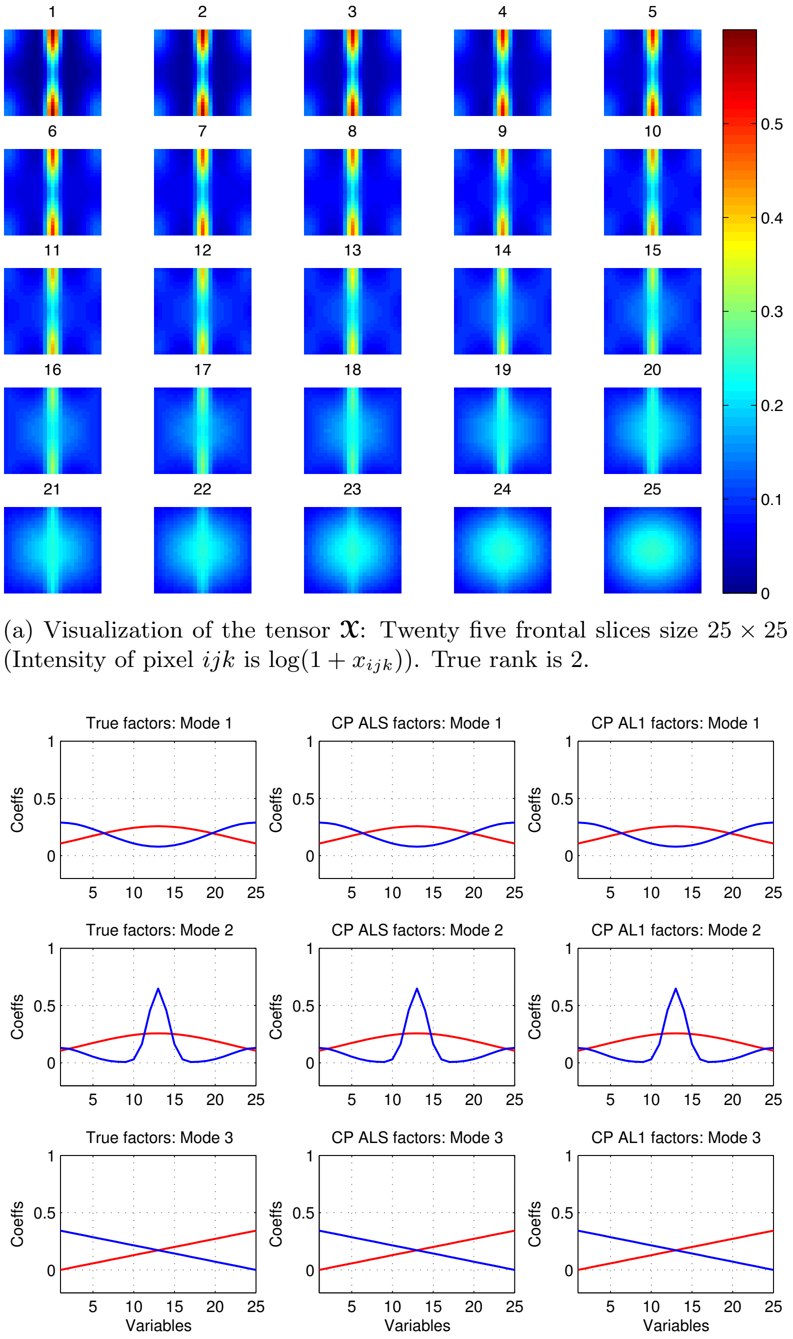
We performed a rank two factorization using the two different algorithms. For all CPAL1 computations we used set $\epsilon = 10^{-1}$ and $\mu = 10^{-8}$. Let the fit of the model, $[\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$, to $\mathcal{X}$ be defined as

$$1 - \frac{\|\mathcal{X} - [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]\|_F}{\|\mathcal{X}\|_F}.$$

Both algorithms were run until the absolute value of the difference between a current fit and the last fit fell below 10^{-5}.

Figure 1a shows the frontal slices of $\mathcal{X}$. Figure 1b shows the true factors for each mode in the left column; the factors for each mode estimated by CPALS in the second column; and factors for each mode estimated by CPAL1 in the third column. The fitted values of $\mathbf{a}_1, \mathbf{b}_1, \mathbf{c}_1$ are in red; fitted values of $\mathbf{a}_2, \mathbf{b}_2, \mathbf{c}_2$ are in blue. The color coding is the same for all plots in the qualitative comparisons between CPALS and CPAL1.

We then added Gaussian noise to every element in the tensor $\mathcal{X}$. Let $\mathcal{N}$ denote a tensor of the same size as $\mathcal{X}$ where $n_{ijk} \sim N(0,1)$ and let $\eta = 0.2$. We created a noisy version of $\mathcal{X}$, $\tilde{\mathcal{X}}$ as follows:



(a) Visualization of the tensor $\mathcal{X}$: Twenty five frontal slices size 25×25 (Intensity of pixel ijk is $\log(1 + x_{ijk})$). True rank is 2.

(b) Rank two CP factorizations by CPALS and CPAL1 agree with each other and the true generative factorization.

Figure 1: Comparison on noise-free data

$$\tilde{\mathcal{X}} = \mathcal{X} + \eta \frac{\| \mathcal{X} \|_F}{\| \mathcal{N} \|_F} \mathcal{N}.$$

Figure 2a and Figure 2b show the frontal slices $\tilde{\mathcal{X}}$ and resulting rank two factorizations by CPALS and AL1. Both factorizations recover the generative model.

We then added artifact noise to $\mathcal{X}$ consisting of a single 5×5 square to each frontal slice (Figure 3a). For $k = 1, \ldots, 25$ we make two independent uniformly random draws U_k and V_k from the set $\{3, 4, \ldots, 22, 23\}$. We created an artifact noise contaminated tensor $\mathcal{X}'$ from $\mathcal{X}$ as follows:

$$x'_{ijk} = \begin{cases} 0.75 & i \in \{U_k - 2, \ldots, U_k + 2\} \text{ and } j \in \{V_k - 2, \ldots, V_k + 2\} \\ x_{ijk} & \text{otherwise} \end{cases}$$

Note that the largest value of $\mathcal{X}$ is 0.7961. So, the intensity of the artifact noise is within the range of the true signal. Figure 3b shows the resulting factorizations. We see that the CPALS factorization gives less accurate answers, while the CPAL1 factorization is unaffected.

We then considered a combination of the dense Gaussian errors and sparse non-Gaussian perturbations. We constructed $\tilde{\mathcal{X}}'$ as follows,

$$\tilde{\mathcal{X}}' = \mathcal{X}' + \eta \frac{\| \mathcal{X} \|_F}{\| \mathcal{N} \|_F} \mathcal{N}.$$

We constructed two more noisy tensors, $\tilde{\mathcal{X}}''$ and $\tilde{\mathcal{X}}'''$ in a similar manner. The resulting frontal slices are shown in Figures 4a, 5a, and 6a. The resulting factorizations are shown in Figures 4b, 5b, and 6b. Again we see that both computations handle the presence of dense Gaussian noise about equally well but that CPALS struggles with the sparse non-Gaussian perturbations in the tensor while CPAL1 does not.

We then considered more subtle artifact noise. We add to each slice pyramids of random size and location. Let g be the real valued function,

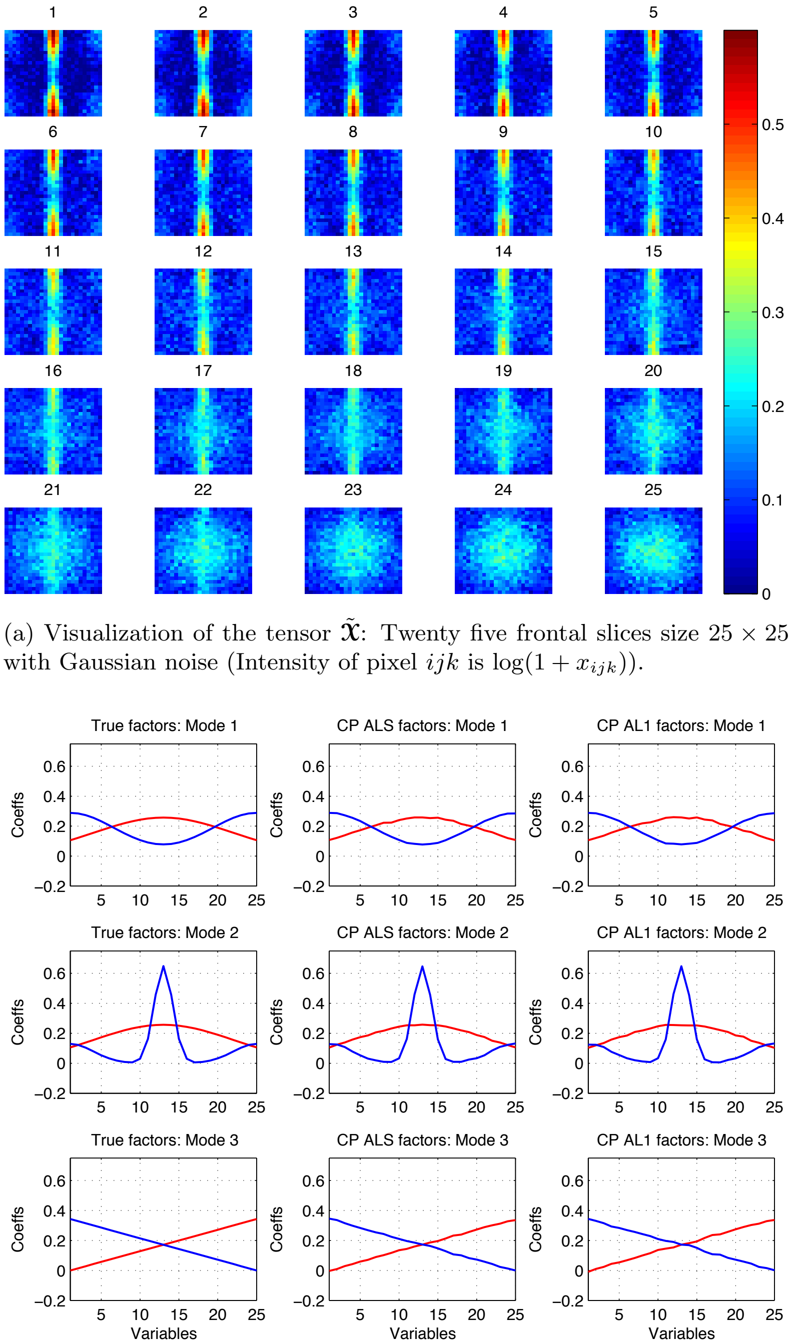
$$g(x, y, s, h) = \max(s \min\{x, y\} + h, 0).$$

Let N_i denote the number of outlying pyramids on the ith slice. For the ith slice and n_ith pyramid on it, we generate random variables that specify the height, H_{x,n_i} and H_{y,n_i}, slope, S_{x,n_i} and S_{y,n_i}, and location, U_{x,n_i} and U_{y,n_i}, of a pyramid.

$$\begin{aligned} H_{x,n_i}, H_{y,n_i} &\overset{iid}{\sim} U[0.2, 1] \\ S_{x,n_i}, S_{y,n_i} &\overset{iid}{\sim} U[0.2, 0.6] \\ U_{x,n_i}, U_{y,n_i} &\overset{iid}{\sim} UNIFORM\{1, 2, \ldots, 23, 24, 25\}, \end{aligned}$$

where UNIFORM$\{1, 2, \ldots, 23, 24, 25\}$ denotes the discrete distribution where the integers between 1 and 25 are drawn with equal probability. We constructed a sparse pyramid noise tensor $\mathcal{P}$ as follows:

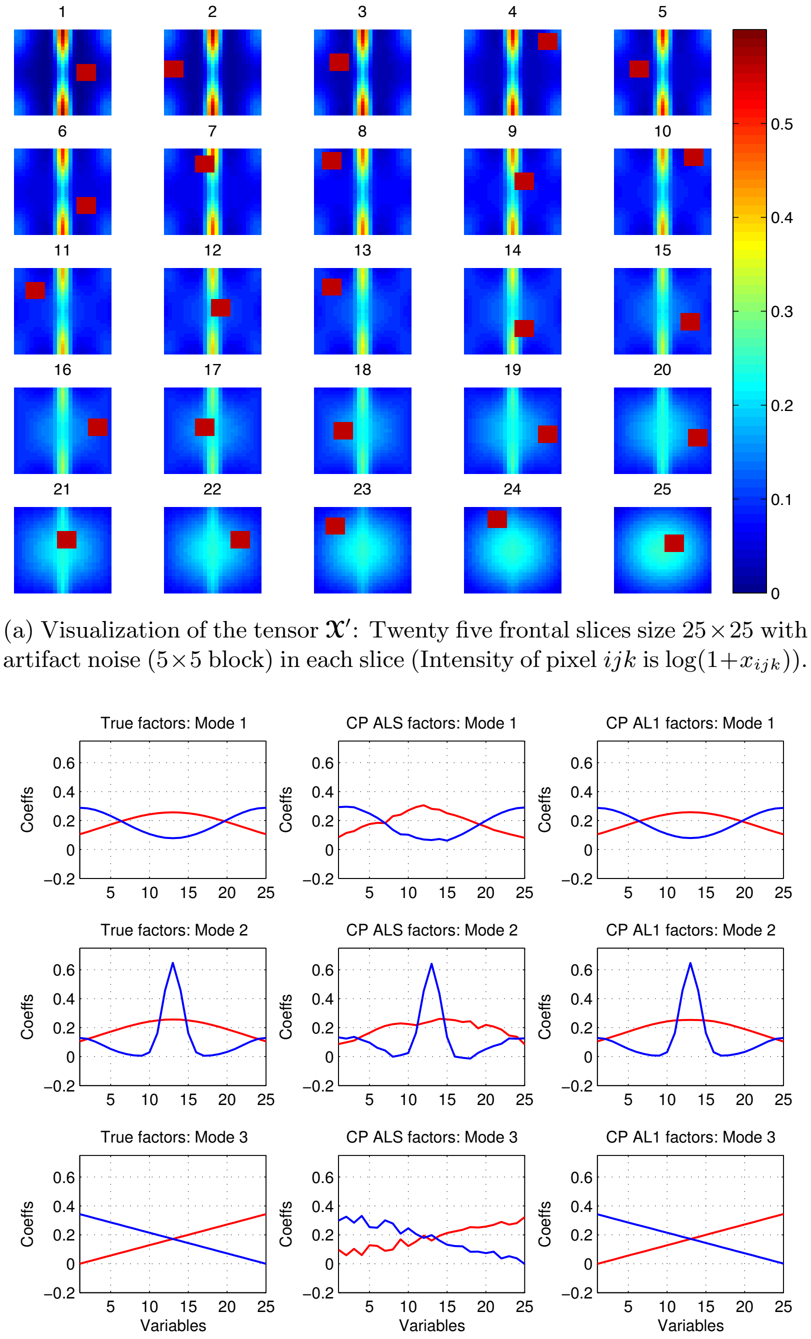
$$p_{ijk} = \max_{n_i = 1, \ldots, N_i} \{ g(S_{x,n_i}, j - U_{x,n_i}, j + U_{x,n_i}, H_{x,n_i}) \times g(S_{y,n_i}, k - U_{y,n_i}, k + U_{y,n_i}, H_{y,n_i}) \}.$$



(a) Visualization of the tensor $\tilde{\mathfrak{X}}$: Twenty five frontal slices size 25×25 with Gaussian noise (Intensity of pixel ijk is $\log(1 + x_{ijk})$).

(b) Rank two CP factorizations of $\tilde{\mathfrak{X}}$ by CPALS and CPAL1 agree with each other and the true generative factorization.

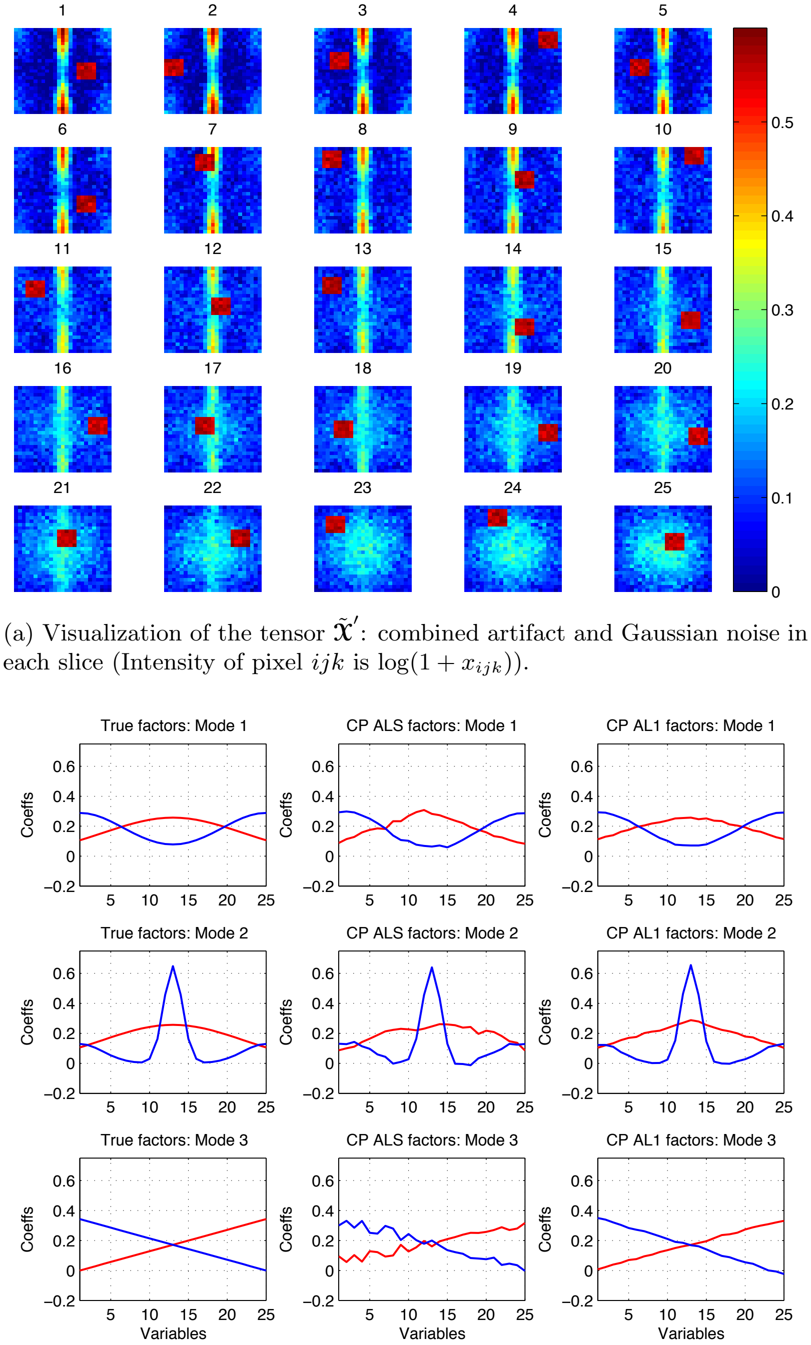
Figure 2: Comparison on data with Gaussian noise



(a) Visualization of the tensor $\mathfrak{X}'$: Twenty five frontal slices size 25×25 with artifact noise (5×5 block) in each slice (Intensity of pixel ijk is $\log(1+x_{ijk})$).

(b) The rank two CPALS factorization displays sensitivity to non-Gaussian perturbations. The rank two CPAL1 factorization still agrees with the true factorization.

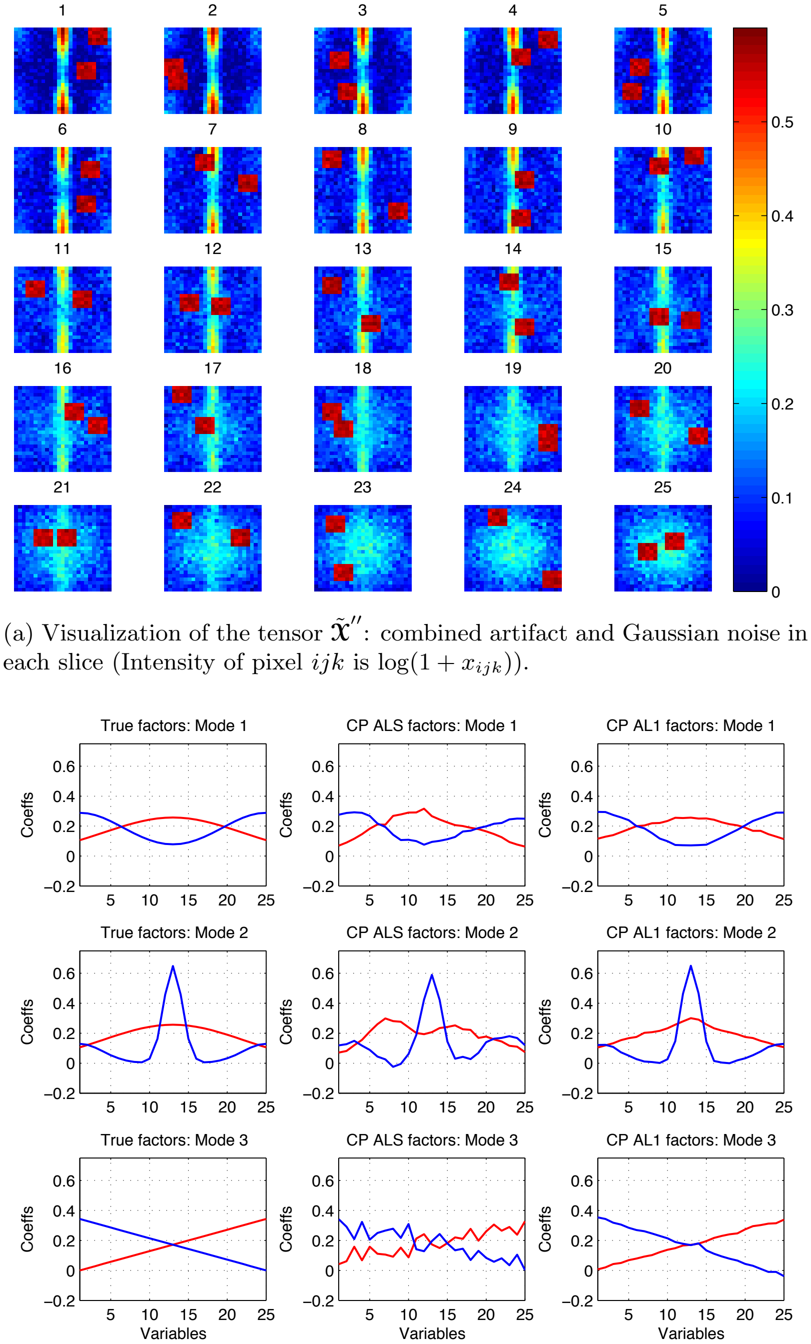
Figure 3: Comparison on data with artifact noise



(a) Visualization of the tensor $\tilde{\mathcal{X}}'$: combined artifact and Gaussian noise in each slice (Intensity of pixel ijk is $\log(1 + x_{ijk})$).

(b) The rank two CPALS factorization displays sensitivity to sparse non-Gaussian perturbations and dense additive Gaussian noise. The rank two CPAL1 factorization still agrees with the true factorization.

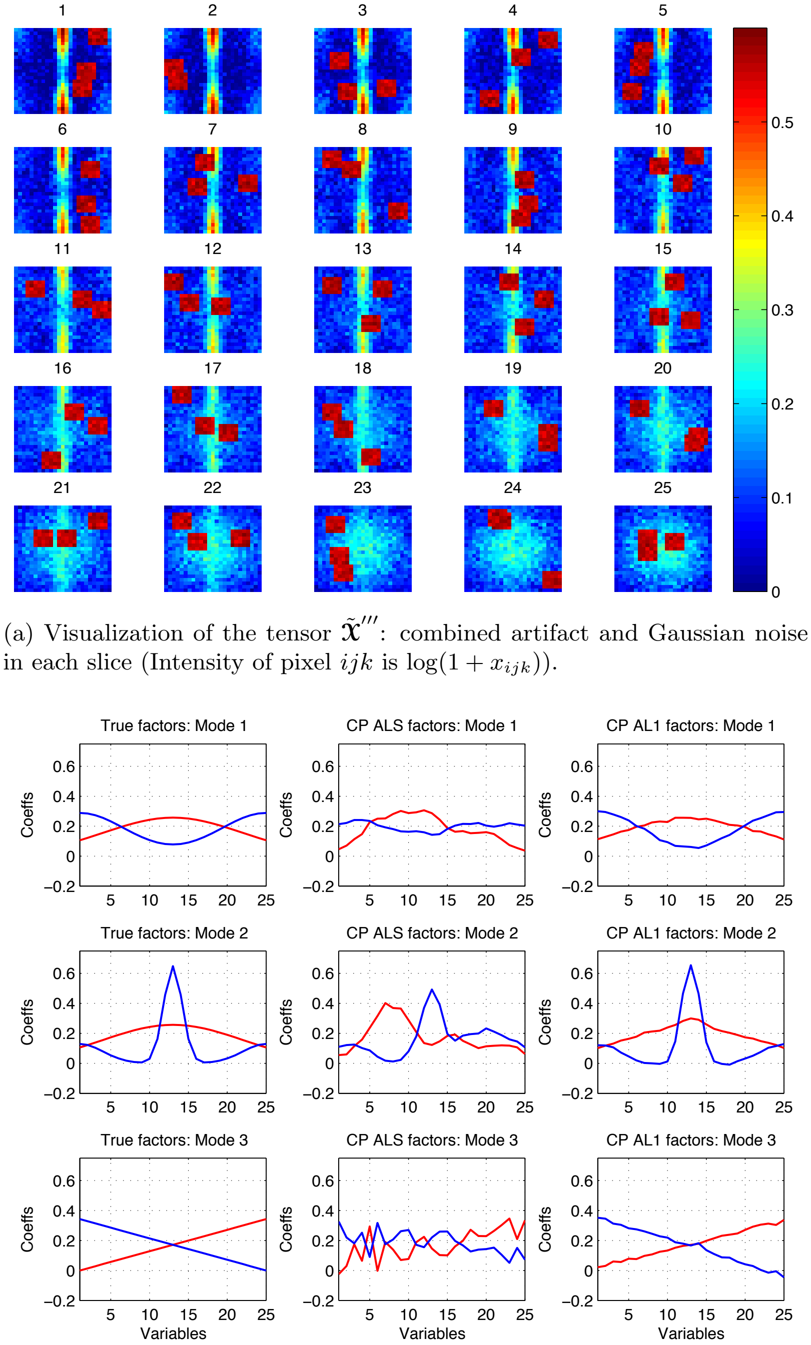
Figure 4: Comparison on data with artifact (1 block per slice) and Gaussian noise



(a) Visualization of the tensor $\tilde{\mathfrak{X}}''$: combined artifact and Gaussian noise in each slice (Intensity of pixel ijk is $\log(1 + x_{ijk})$).

(b) The rank two CPALS factorization continues to diverge from the true factorization as more non-Gaussian perturbations add added and Gaussian noise is also added to all entries. The two CPAL1 factorization still agrees with the true factorization.

Figure 5: Comparison on ata with artifact (2 blocks per slice) and Gaussian noise



(a) Visualization of the tensor $\tilde{\mathfrak{X}}'''$: combined artifact and Gaussian noise in each slice (Intensity of pixel ijk is $\log(1 + x_{ijk})$).

(b) The rank two CPALS factorization disagrees quite a bit from the true factorization as even more non-Gaussian perturbations add added. The rank two CPAL1 factorization still agrees with the true factorization.

Figure 6: Comparison on data with artifact (3 blocks per slice) and Gaussian noise

The sparse perturbations were superposed onto the data tensor $\mathcal{X}$ by taking element-wise maxima to generate the tensor $\mathcal{X}'$:

$$x'_{ijk} = \max\{p_{ijk}, x_{ijk}\}.$$

Gaussian noise was then added as before, with $\eta = 0.2$:

$$\tilde{\mathcal{X}}' = \mathcal{X}' + \eta \frac{\| \mathcal{X} \|_F}{\| \mathcal{N} \|_F} \mathcal{N}.$$

The resulting frontal slices are shown in Figure 7a. The resulting factorizations are shown in Figure 7b. We see that CPAL1 produces a rank two factorization in better agreement with the true generative model. The differences with the rank two CPALS factorization, however, are less pronounced. In these examples, the story is that the greatest differences between CPALS and CPAL1 occur when the outlier signal is sparsely distributed over the tensor and the magnitude of outlying entries is very large compared to the underlying signal.

8.2 Quantitative Comparisons between CPALS and CPAL1

We create 3-way tensors $\mathcal{X}' \in \mathbb{R}^{50 \times 50 \times 50}$ of rank-5 as follows.

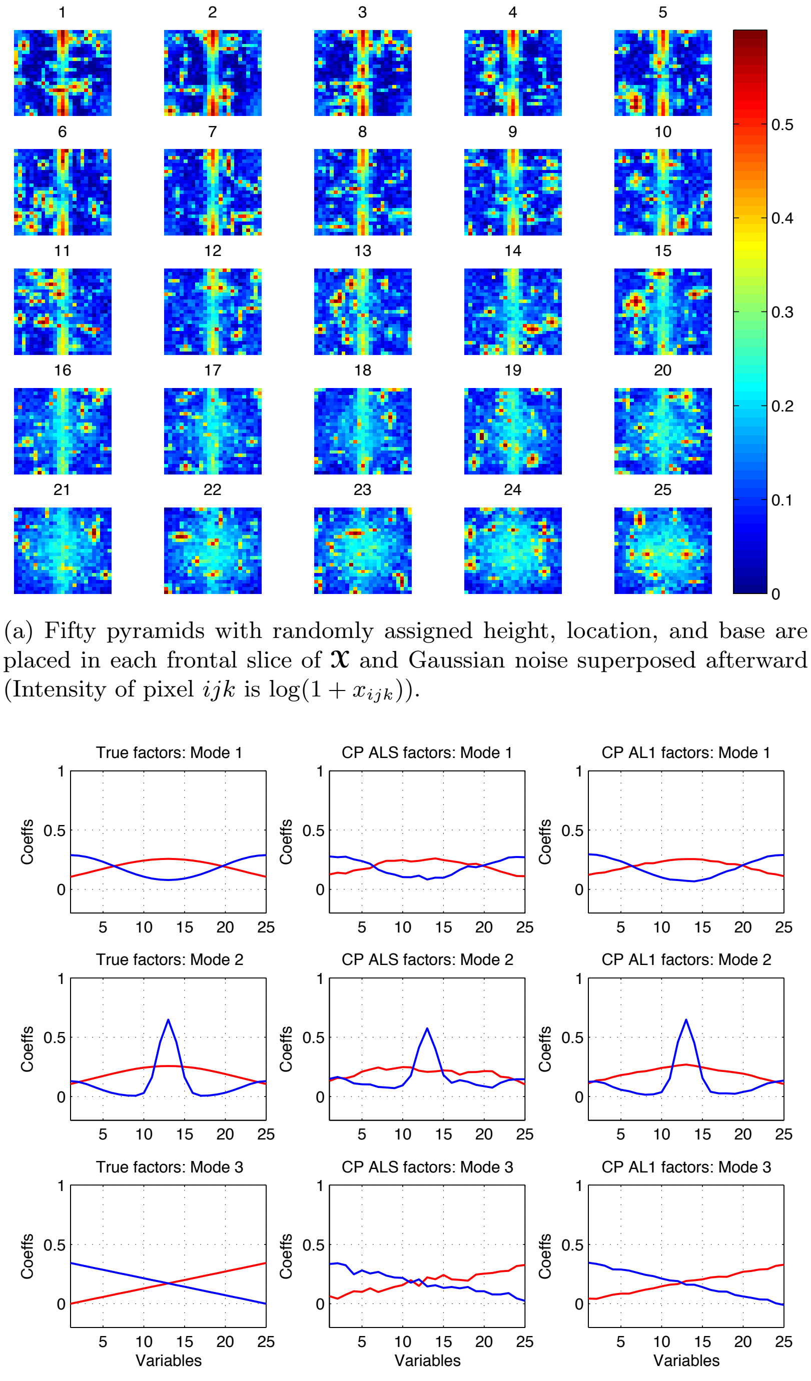
$$\mathcal{X}' = \mathcal{X} + \gamma \frac{\|\mathcal{X}\|_F}{\|\mathcal{P}\|_F} \mathcal{P} + 0.1 \frac{\|\mathcal{X}\|_F}{\|\mathcal{Q}\|_F} \mathcal{Q}$$

for $\eta = 0.1, 0.2$ and $\gamma = 0.5, 1.0, 1.5$ and 2.0. For all combinations of η and γ the scaled values of q_{ijk} were less than the largest value of $\mathcal{X}$.

We first generated random factor matrices $\mathbf{A}, \mathbf{B}, \mathbf{C} \in \mathbb{R}^{50 \times 5}$ where the matrix elements were the absolute values of i.i.d. draws from a standard Gaussian. The ijk^{th} entry of the noise free tensor $\mathcal{X}$ was then set to be $\sum_{r=1}^{R} a_{i_1 r} b_{i_2 r} c_{i_3 r}$. Then to each $\mathcal{X}$ we added dense Gaussian noise and artifact outliers. All random variables we describe were independently drawn. We generated an artifact tensor $\mathcal{P}$ as follows. A fraction η of the tensor entries was selected randomly. We then assigned to each of the selected entries a value drawn from a Gamma distribution with shape parameter 50 and scale parameter 1/50. All other entries were set to 0. For the dense Gaussian noise tensor $\mathcal{Q}$, the entries q_{ijk} were i.i.d. draws from a standard Gaussian. The tensor $\mathcal{X}'$ was obtained by adding the noise and artifact tensors to $\mathcal{X}$.

For every pair (η, γ) we performed 100 rank-5 factorizations under the two methods. For CPAL1 computations we set $\epsilon = 10^{-10}$ and $\mu = 10^{-8}$. Initial points for all tests were generated using the n-mode singular vectors of the tensor (i.e., the `nvecs` command in the Tensor Toolbox). To assess the goodness of a computed factorization we calculated the factor match score (FMS) between the estimated and true factors [1]. The FMS ranges between 0 and 1; an FMS of 1 corresponds to a perfect recovery of the original factors.

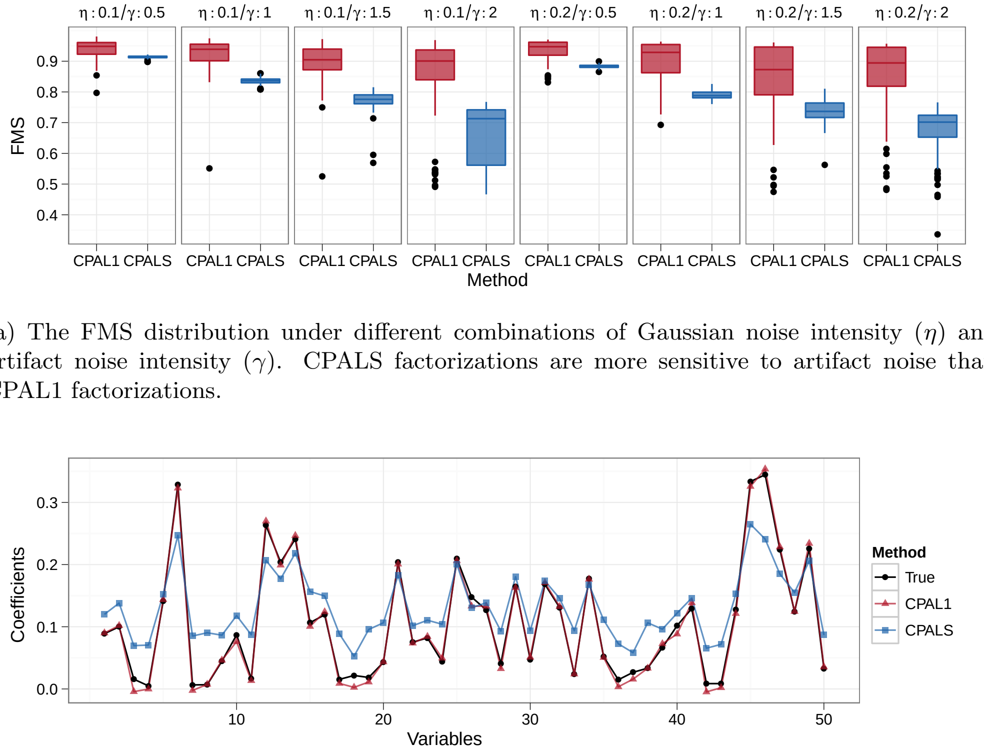
Figure 8a shows boxplots of the FMS under both methods. The scores for CPALS decreased as the contribution of non-Gaussian noise increased. In contrast regardless of the noise distributions applied CPAL1 tended to recover the true factorization with the exception of occasionally finding local minima,



(a) Fifty pyramids with randomly assigned height, location, and base are placed in each frontal slice of $\mathcal{X}$ and Gaussian noise superposed afterward (Intensity of pixel ijk is $\log(1 + x_{ijk})$).

(b) The rank two CPALS factorization is less sensitive to the presence of the pyramid outliers. The rank two CPAL1 factorization has slightly better aggreement with the true factorization.

Figure 7: Comparison on data with random-shaped artifacts and Gaussian noise



(a) The FMS distribution under different combinations of Gaussian noise intensity (η) and artifact noise intensity (γ). CPALS factorizations are more sensitive to artifact noise than CPAL1 factorizations.

(b) A comparison of a single recovered factor for a replicate when $\eta = 0.2$ and $\gamma = 2$. Here the FMS was 0.91 and 0.64 for CPAL1 and CPALS respectively. Factor columns were normalized for comparison.

Figure 8: Quantitative comparisons

Figure 8b compares an example factor (one column of a factor matrix) when $\eta = 0.2$ and $\gamma = 2$ for the two methods. We can see that CPAL1 is a much closer match to the true underlying factor. Specifically, the FMS was 0.91 and 0.64 for CPAL1 and CPALS, respectively. The median CPALS FMS was 0.7, so this is typical.

This page intentionally left blank.

9 Extensions to the Poisson Likelihood

The CP factorization can be extended to handle binary and nonnegative tensor entries through the machinery of generalized linear models just as Principal Components Analysis has been extended for matrices of binary and nonnegative matrices [7, 21, 18, 19]. Here, we consider nonnegative factorization formulated as a Poisson likelihood maximization problem. We see again that if the variation in the data violates the assumptions on the statistical model of the variation, the resulting factorizations can be suboptimal. We show that using the 1-norm loss in the case of nonnegative factorization is inadequate, motivating the need for different robust alternatives.

9.1 The Poisson Likelihood and Nonnegative Tensor Factorizations

When tensors consist of nonnegative integers or counts, a Gaussian likelihood model for the randomness is a poor description for the observed data. Recall that for an order N tensor $\boldsymbol{\mathcal{X}}$ the best rank R CP decomposition $[\![\lambda; \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \ldots, \mathbf{A}^{(N)}]\!]$ of $\boldsymbol{\mathcal{X}}$ with respect to the Frobenius norm corresponds to the MLE of $(\boldsymbol{\lambda}; \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \ldots, \mathbf{A}^{(N)})$ where $x_{i_1,\ldots,i_N}$ are independently distributed and

$$x_{i_1,\ldots,i_N} \sim N\left(\sum_{r=1}^{R} \lambda_r a_{i_1 r}^{(1)} a_{i_2 r}^{(2)} \cdots a_{i_N r}^{(N)}, 1\right).$$

This model is inappropriate for count data in two ways; under this model $x_{i_1,\ldots,i_N}$ can take on negative values and/or non-integer values. To put it another way, if one were interested in simulating multi-way data from the MLE of $(\boldsymbol{\lambda}; \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \ldots, \mathbf{A}^{(N)})$ the generated samples would be on the wrong scale; they would need to be transformed to create count data.

A more elegant alternative is to directly model the discrete nonnegative nature of the tensor elements with a discrete distribution such as the Poisson distribution. In fact, the popular method of minimizing the generalized Kullback-Leibler divergence introduced by Lee and Seung [18, 19] is equivalent to maximum likelihood estimation under the assumption that $x_{i_1,\ldots,i_N}$ are independently distributed and

$$x_{i_1,\ldots,i_N} \sim \text{POISSON}\left(\sum_{r=1}^{R} \lambda_r a_{i_1 r}^{(1)} a_{i_2 r}^{(2)} \cdots a_{i_N r}^{(N)}\right), \tag{13}$$

and $\boldsymbol{\lambda}, \mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(N)}$ are all constrained to be nonnegative. To see, this note that the log-likelihood

$$\begin{aligned} &\log\left\{\prod_{i_1,\ldots,i_N} \frac{\left(\sum_{r=1}^{R} \lambda_r a_{i_1 r}^{(1)} a_{i_2 r}^{(2)} \cdots a_{i_N r}^{(N)}\right)^{x_{i_1,\ldots,i_N}}}{x_{i_1,\ldots,i_N}!} \exp\left(-\sum_{r=1}^{R} \lambda_r a_{i_1 r}^{(1)} a_{i_2 r}^{(2)} \cdots a_{i_N r}^{(N)}\right)\right\} \\ = &\sum_{i_1,\ldots,i_N}\left\{x_{i_1,\ldots,i_N} \log\left(\sum_{r=1}^{R} \lambda_r a_{i_1 r}^{(1)} a_{i_2 r}^{(2)} \cdots a_{i_N r}^{(N)}\right) - \sum_{r=1}^{R} \lambda_r a_{i_1 r}^{(1)} a_{i_2 r}^{(2)} \cdots a_{i_N r}^{(N)}\right\} - \sum_{i_1,\ldots,i_N} \log\left(x_{i_1,\ldots,i_N}!\right) \end{aligned}$$

equals the generalized Kullback-Leibler divergence up to a scalar shift. Note that (13) is the generalization of the log-likelihood that is shown to be maximized in NNMF in Example 4.4.

An advantage of Lee and Seung's factorization is that it tends to produce a "parts-based" representation of the data. We can understand why this is so from the model (13). Recall that the sum of independent Poisson random variables is also Poisson.

Proposition 9.1 *Let $X_i \sim$ POISSON(λ_i), for $i = 1, \dots, N$, be independent random variables. Then*

$$\sum_{i=1}^{N} X_i \sim \mathit{POISSON}\left(\sum_{i=1}^{N} \lambda_i\right).$$

Proof. We show the claim is true for $N = 2$. The general case follows from an induction argument.

$$\begin{aligned} P(X_1 + X_2 = k) &= \sum_{r=0}^{k} P(X_1 = k - r, X_2 = r) \\ &= \sum_{r=0}^{k} P(X_1 = k - r) P(X_2 = r) \\ &= \sum_{r=0}^{k} \frac{\lambda_1^{k-r}}{(k-r)!} e^{-\lambda_1} \frac{\lambda_2^{r}}{r!} e^{-\lambda_2} \\ &= \frac{\exp^{-(\lambda_1+\lambda_2)}}{k!} \sum_{r=0}^{k} \frac{k!}{(k-r)!r!} \lambda_1^{k-r} \lambda_2^{r} \\ &= \frac{(\lambda_1 + \lambda_2)^k \exp^{-(\lambda_1+\lambda_2)}}{k!}. \end{aligned}$$

The first equality follows from the law of total probability. The second equality follows from the independence of X_1 and X_2. The last equality follows from the binomial theorem. □

Model (13) is equivalent to the following superposition model: $\mathcal{X} = \mathcal{X}^{(1)} + \mathcal{X}^{(2)} + \cdots + \mathcal{X}^{(R)}$ and the $x_{i_1,\dots,i_N}^{(r)}$ are independent Poisson random variables:

$$x_{i_1,\dots,i_N}^{(r)} \sim \text{POISSON}\left(\lambda_r a_{i_1 r}^{(1)} a_{i_2 r}^{(2)} \cdots a_{i_N r}^{(N)}\right).$$

So, model (13) expresses $\mathcal{X}$ as the sum of nonnegative tensors $\mathcal{X}^{(r)}$ where

$$\mathcal{X}^{(r)} \sim \text{POISSON}\left(\llbracket \lambda_r; \mathbf{a}_r^{(1)}, \mathbf{a}_r^{(2)}, \dots, \mathbf{a}_r^{(N)} \rrbracket\right)$$

The next subsections review algorithms for computing the NNMF decomposition under the generalized Kullback-Leibler divergence. The algorithm for matrix and tensor factorizations amount to solving alternating Poisson regression problems with an MM algorithm. The algorithm is identical to the Lee and Seung algorithm and its tensor extension [26] except that for each subproblem it iterates until convergence, whereas the Lee and Seung algorithm and its tensor extension complete just a single iteration for each subproblem. We first review how to do Poisson regression.

9.2 Poisson Regression

Let Y_i be independent Poisson$(\mathbf{x}_{[i]}^{\mathsf{T}}\mathbf{b})$ for $i = 1, \ldots, N$ and $\mathbf{X} \in \mathbb{R}_{+}^{N \times P}$ and $\mathbf{b} \in \mathbb{R}_{+}^{P}$. Let y_i denote the observed value of Y_i. The negative log-likelihood is

$$-\log \ell(\mathbf{b}) = -\log \left(\prod_{i=1}^{N} \frac{\left(\mathbf{x}_{[i]}^{\mathsf{T}}\mathbf{b}\right)^{y_i} \exp\left(\mathbf{x}_{[i]}^{\mathsf{T}}\mathbf{b}\right)}{y_i!} \right) = \sum_{i=1}^{N} \left(\mathbf{x}_{[i]}^{\mathsf{T}}\mathbf{b} - y_i \log\left(\mathbf{x}_{[i]}^{\mathsf{T}}\mathbf{b}\right) + \log(y_i!) \right).$$

Take the loss function $L(\mathbf{b})$ to be

$$L(\mathbf{b}) = \sum_{i=1}^{N} f_i\left(\mathbf{x}_{[i]}^{\mathsf{T}}\mathbf{b}\right),$$

where $f_i(t) = t - y_i \log(t)$. Since log is concave and $y_i \in \mathbb{Z}_{+}$, f_i is convex, $L(\mathbf{b})$ is convex.

Proposition 9.2 *The following function majorizes $f_i(\mathbf{x}_{[i]}^{\mathsf{T}}\mathbf{b})$ at $\tilde{\mathbf{b}}$:*

$$g_i(\mathbf{b}|\tilde{\mathbf{b}}) = \sum_{j=1}^{P} \alpha_{ij} f_i\left(\frac{\mathbf{x}_{[i]}^{\mathsf{T}}\tilde{\mathbf{b}}}{\tilde{b}_j} b_j \right),$$

where $\alpha_{ij} = x_{ij}\tilde{b}_j / \mathbf{x}_{[i]}^{\mathsf{T}}\tilde{\mathbf{b}}$.

Proof. It is immediate that $g_i(\tilde{\mathbf{b}}|\tilde{\mathbf{b}}) = f_i\left(\mathbf{x}_{[i]}^{\mathsf{T}}\tilde{\mathbf{b}}\right)$. We next need to verify the inequality. Since $\mathbf{b}$ has positive elements and $\mathbf{x}_{[i]}$ has positive elements for all i, we have that $\alpha_{ij} \in [0, 1]$ and $\sum_{j=1}^{P} \alpha_{ij} = 1$. We then make the inspired observation that the inner product $\mathbf{x}_{[i]}^{\mathsf{T}}\mathbf{b}$ can be expressed in terms of the α_{ij}:

$$\mathbf{x}_{[i]}^{\mathsf{T}}\mathbf{b} = \sum_{j=1}^{P} x_{ij} b_j = \sum_{j=1}^{P} \left(\frac{x_{ij}\tilde{b}_j}{\mathbf{x}_{[i]}^{\mathsf{T}}\tilde{\mathbf{b}}} \right) \left(\frac{\mathbf{x}_{[i]}^{\mathsf{T}}\tilde{\mathbf{b}}}{\tilde{b}_j} \right) b_j = \sum_{j=1}^{P} \alpha_{ij} \left(\frac{\mathbf{x}_{[i]}^{\mathsf{T}}\tilde{\mathbf{b}}}{\tilde{b}_j} \right) b_j.$$

To complete the proof use the convexity of f_i to get

$$f_i\left(\mathbf{x}_{[i]}^{\mathsf{T}}\mathbf{b}\right) = f_i\left(\sum_{j=1}^{P} \alpha_{ij} \left(\frac{\mathbf{x}_{[i]}^{\mathsf{T}}\tilde{\mathbf{b}}}{\tilde{b}_j} \right) b_j \right) \leq \sum_{j=1}^{P} \alpha_{ij} f_i\left(\frac{\mathbf{x}_{[i]}^{\mathsf{T}}\tilde{\mathbf{b}}}{\tilde{b}_j} b_j \right) = g_i(\mathbf{b}|\tilde{\mathbf{b}}).$$

□

Note that $L(\mathbf{b})$ is majorized at $\tilde{\mathbf{b}}$ by $\sum_i g_i(\mathbf{b}|\tilde{\mathbf{b}})$. To find the MLE of $\mathbf{b}$, we minimize $\sum_i g_i(\mathbf{b}|\tilde{\mathbf{b}})$. Let us calculate the derivative with respect to b_j:

$$\frac{\partial}{\partial b_j} \sum_{i=1}^{N} g_i(\mathbf{b}|\tilde{\mathbf{b}}) = \sum_{i=1}^{N} x_{ij} f_i'\left(\frac{\mathbf{x}_{[i]}^{\mathsf{T}}\tilde{\mathbf{b}}}{\tilde{b}_j} b_j \right) = \sum_{i=1}^{N} x_{ij} \left(1 - y_i \frac{\tilde{b}_j}{b_j \mathbf{x}_{[i]}^{\mathsf{T}}\tilde{\mathbf{b}}} \right).$$

By setting the derivatives equal to zero we obtain a multiplicative relationship between b_j and $\tilde{b}_j$:

$$
\begin{aligned}
0 &= \sum_{i=1}^{N} x_{ij} \left(1 - y_i \frac{\tilde{b}_j}{b_j \mathbf{x}_{[i]}^{\mathsf{T}} \tilde{\mathbf{b}}}\right) \\
\sum_{i=1}^{N} x_{ij} &= \frac{\tilde{b}_j}{b_j} \sum_{i=1}^{N} \left(\frac{y_i}{\mathbf{x}_{[i]}^{\mathsf{T}} \tilde{\mathbf{b}}}\right) x_{ij} \\
b_j &= \tilde{b}_j \left[\left\{\sum_{i=1}^{N} \left(\frac{y_i}{\mathbf{x}_{[i]}^{\mathsf{T}} \tilde{\mathbf{b}}}\right) x_{ij}\right\} \Big/ \sum_{i=1}^{N} x_{ij}\right]
\end{aligned}
$$

Note that when y_i is close to $\mathbf{x}_{[i]}^{\mathsf{T}} \tilde{\mathbf{b}}$ for all i, $\mathbf{b}$ is very close to $\tilde{\mathbf{b}}$. Recall that if $y_i \sim \text{Poisson}\left(\mathbf{x}_{[i]}^{\mathsf{T}} \tilde{\mathbf{b}}\right)$, then the expected value of y_i is $\mathbf{x}_{[i]}^{\mathsf{T}} \tilde{\mathbf{b}}$. So when y_i is close to $\mathbf{x}_{[i]}^{\mathsf{T}} \tilde{\mathbf{b}}$ for all i, the parameter $\tilde{\mathbf{b}}$ explains the data well and the updates should be expected to be minor.

9.3 Poisson PCA using the Non-canonical link

We next show how our NNMF algorithm is equivalent to alternating Poisson regression and can be solved using calls the algorithm described in Section 9.2. Let Y_{ij} be independent Poisson($\mathbf{a}_{[i]}^{\mathsf{T}} \mathbf{b}_{[j]}$) for $i = 1, \ldots, M$ and $j = 1, \ldots, N$ and $\mathbf{A} \in \mathbb{R}_+^{M \times K}$ and $\mathbf{B} \in \mathbb{R}_+^{N \times K}$. The negative log-likelihood is given by

$$
\begin{aligned}
-\log \ell(\mathbf{A}, \mathbf{B}) &= -\log \left(\prod_{i=1}^{M} \prod_{j=1}^{N} \frac{(\mathbf{a}_{[i]}^{\mathsf{T}} \mathbf{b}_{[j]})^{y_{ij}} \exp(\mathbf{a}_{[i]}^{\mathsf{T}} \mathbf{b}_{[j]})}{y_{ij}!}\right) \\
&= \sum_{i=1}^{M} \sum_{j=1}^{N} \left(\mathbf{a}_{[i]}^{\mathsf{T}} \mathbf{b}_{[j]} - y_{ij} \log(\mathbf{a}_{[i]}^{\mathsf{T}} \mathbf{b}_{[j]}) + \log(y_{ij}!)\right).
\end{aligned}
$$

Take the loss function $L(\mathbf{A}, \mathbf{B})$ to be

$$
L(\mathbf{A}, \mathbf{B}) = \sum_{i=1}^{M} \sum_{j=1}^{N} f_{ij}\left(\mathbf{a}_{[i]}^{\mathsf{T}} \mathbf{b}_{[j]}\right),
$$

where

$$
f_{ij}(t) = t - y_{ij} \log(t).
$$

We alternate minimizing $\mathbf{A}$ and $\mathbf{B}$ with the other fixed using the Poisson regression algorithm described in Section 9.2. First fix $\mathbf{B}$ and minimize $L(\mathbf{A}, \mathbf{B})$ with respect to $\mathbf{A}$. Note that $L(\mathbf{A}, \mathbf{B})$ is separable in $\mathbf{a}_{[i]}$. Consequently if we find the $\mathbf{a}_{[i]}$ that minimizes

$$
\sum_{j=1}^{N} f_{ij}\left(\mathbf{a}_{[i]}^{\mathsf{T}} \mathbf{b}_{[j]}\right) \tag{14}
$$

Algorithm 4 Poisson PCA for Nonnegative Matrix Factorization (CPAPR)

```
repeat
    repeat
        for $i = 1, \ldots, M$, and $k = 1, \ldots, R$ do
            $a_{ik} \leftarrow a_{ik} \left[ \left\{ \sum_{j=1}^{N} \left( \frac{y_{ij}}{\mathbf{b}_{[j]}^{\mathsf{T}} \mathbf{a}_{[i]} + \varepsilon} \right) b_{jk} \right\} \Big/ \left( \sum_{j=1}^{N} b_{jk} + \varepsilon \right) \right].$
        end for
    until $\mathbf{A}$ converges
    repeat
        for $j = 1, \ldots, N$, and $k = 1, \ldots, R$ do
            $b_{jk} \leftarrow b_{jk} \left[ \left\{ \sum_{i=1}^{M} \left( \frac{y_{ij}}{\mathbf{a}_{[i]}^{\mathsf{T}} \mathbf{b}_{[j]} + \varepsilon} \right) a_{ik} \right\} \Big/ \left( \sum_{i=1}^{M} a_{ik} + \varepsilon \right) \right].$
        end for
    until $\mathbf{B}$ converges
until $(\mathbf{A}, \mathbf{B})$ converges
return  $\mathbf{A}$ and $\mathbf{B}$
```

for every i, then we have found the $\mathbf{A}$ that minimizes $L(\mathbf{A}, \mathbf{B})$. Since minimizing (14) is a Poisson regression problem, we can use the MM algorithm introduced in Section 9.2 to obtain the following update rule for a_{ik}

$$a_{ik} \leftarrow a_{ik} \left[\left\{ \sum_{j=1}^{N} \left(\frac{y_{ij}}{\mathbf{b}_{[j]}^{\mathsf{T}} \mathbf{a}_{[i]}} \right) b_{jk} \right\} \Big/ \sum_{j=1}^{N} b_{jk} \right].$$

By symmetry, minimizing $L(\mathbf{A}, \mathbf{B})$ with respect to $\mathbf{B}$ with $\mathbf{A}$ fixed, gives the update rule for b_{jk}

$$b_{jk} \leftarrow b_{jk} \left[\left\{ \sum_{i=1}^{M} \left(\frac{y_{ij}}{\mathbf{a}_{[i]}^{\mathsf{T}} \mathbf{b}_{[j]}} \right) a_{ik} \right\} \Big/ \sum_{i=1}^{M} a_{ik} \right].$$

Minimizing $L(\mathbf{A}, \mathbf{B})$ with respect to $\mathbf{A}$ with $\mathbf{B}$ fixed amounts to performing N independent Poisson regressions. Similarly, minimizing $L(\mathbf{A}, \mathbf{B})$ with respect to $\mathbf{B}$ with $\mathbf{A}$ fixed amounts to performing M independent Poisson regressions. We alternate updating $\mathbf{A}$ and $\mathbf{B}$ until convergence. Algorithm 4 outlines pseudocode for the described algorithm for computing the decomposition with R components for the two-way tensor tensor $\boldsymbol{\mathcal{X}}$ of size $M \times N$. Note that to prevent potential division by zero, we add a small positive value ε to all denominators. This is a heuristic, however, and the updates no longer correspond to an MM algorithm.

Algorithm 4 can be generalized to handle factorizations of tensors of arbitrary number of modes. Since the basic step in Algorithm 4 is Poisson Regression, we refer to the generalization to tensors of arbitrary number of modes as CP Alternative Poisson Regression (CPAPR).

The effect of inliers

We consider the effect on CPAPR when some of the observations grossly violate the Poisson distribution assumptions. To establish a baseline comparison between the three methods we compare their resulting factorizations on data that is generated according to the assumed Poisson likelihood model. Specifically we generated a matrix $\mathbf{Y} \in \mathbb{N}^{161 \times 161}$ as follows. Define $\boldsymbol{\omega} \in \mathbb{R}^{161}$ with

$\omega_i = -4 + (i-1)/20$. Then we set

$$\begin{aligned}
\mathbf{a}_1 &= \phi(\boldsymbol{\omega}) / \parallel \phi(\boldsymbol{\omega}) \parallel, \\
\mathbf{a}_2 &= \phi(\boldsymbol{\omega} - \mathbf{1}) / \parallel \phi(\boldsymbol{\omega} - \mathbf{1}) \parallel, \\
\mathbf{b}_1 &= \phi(\boldsymbol{\omega}) / \parallel \phi(\boldsymbol{\omega}) \parallel, \text{and} \\
\mathbf{b}_2 &= \phi(\boldsymbol{\omega} + \mathbf{1}) / \parallel \phi(\boldsymbol{\omega} + \mathbf{1}) \parallel.
\end{aligned}$$

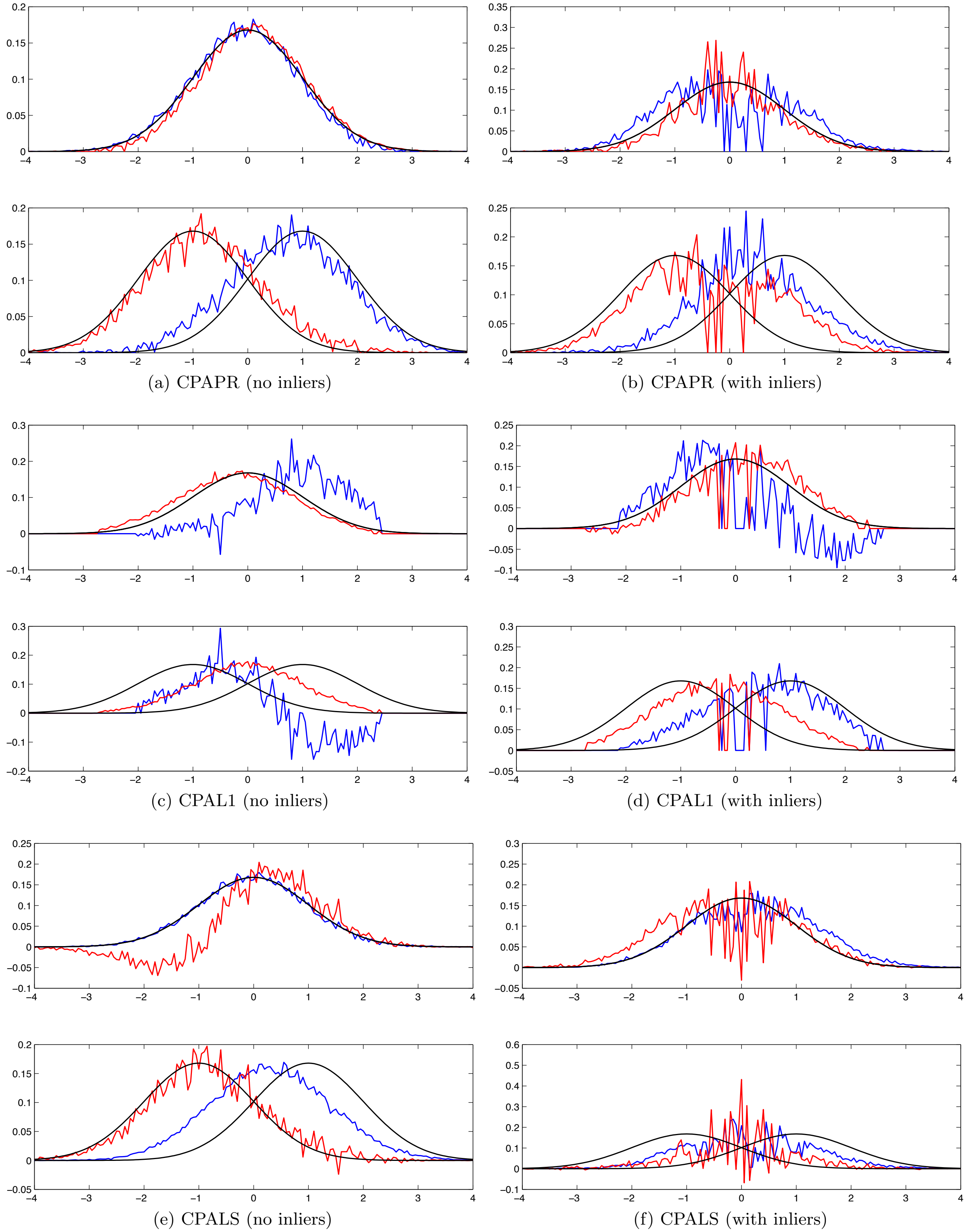
Then we drew Y_{ij} independently as

$$Y_{ij} \sim \text{POISSON}(200 a_{i1} \circ b_{j1} + 50 a_{i2} \circ b_{j2}).$$

Figure 9a, Figure 9c, and Figure 9e show the factorization of a rank 2 nonnegative matrix by CPAPR, CPAL1, and CPALS respectively. Only CPAPR comes close to recovering the true underlying factors. Both CPAL1 and CPALS return factors that have negative elements. Moreover, CPAL1 tends to assign more zeros to factor matrix elements when the data is sparse than CPAPR and CPALS.

We next considered the effects of extreme inliers. An inlier is a data value within the nominal range but incorrectly specified. In the following example we set a randomly selected half of the values that are above 25 to 0. In this particular example, the observed data values ranged from 0 to 49. Figure 9b, Figure 9d, and Figure 9f show the factorization of a rank 2 nonnegative matrix by CPAPR, CPAL1, and CPALS respectively. The presence of inliers only exacerbates the poor performance of CPALS and CPAL1, but CPAPR is also adversely affected by the inliers. The inlier experiment suggests the need for a robust version of CPAPR. On the other hand, both experiments indicate that CPAL1 is not the solution.

One approach to making a robust version of CPAPR is to replace the loss function with a member of the family of density power divergence associated with the Poisson distribution [3]. Members of this family of divergence measures are indexed by a parameter that explicitly trades off statistical efficiency for robustness. Maximum likelihood estimation corresponds to minimum Kullback-Leibler divergence. The Kullback-Leibler divergence in turn is the member of the family with greatest efficiency and least robustness. Estimates using another member of the family in contrast would be less efficient but would have greater robustness against non-Poisson variation. The same strategy could be applied for other assumptions on the statistical behavior of the variation (e.g., binary data).



(a) CPAPR (no inliers)

(b) CPAPR (with inliers)

(c) CPAL1 (no inliers)

(d) CPAL1 (with inliers)

(e) CPALS (no inliers)

(f) CPALS (with inliers)

Figure 9: Example Poisson data (with and without inliers)

This page intentionally left blank.

10 Conclusion

We derived a robust tensor factorization algorithm, CPAL1, based on an approximate 1-norm loss. We have shown that a Tychonoff regularized version of CPAL1 generates iterates whose limit points are stationary points of the the regularized approximate 1-norm loss.

Our simulations demonstrated that there are scenarios in which CPAL1 outperforms CPALS in the presence of artifact noise. Simulation studies also showed that not all non-Gaussian perturbations cause noticeable degradation in the CPALS factorization. Conversely, there are situations when CPAL1 struggles as much as CPALS in the presence of artifact noise, e.g. when the data tensor is sparse as well as seen in the example of nonnegative factorization of sparse count data. We conjecture that CPAL1 is most suited to handle artifact noise when the data tensor is dense. Finding an alternative to the 1-norm loss for sparse data with non-Gaussian noise is a direction for future research.

We also briefly reviewed nonnegative tensor factorization and derived a CP factorization based on maximizing a Poisson likelihood with an alternating MM algorithm, CPAPR. In simulation experiments, CPAPR was shown to be sensitive to the presence of inliers. But CPAL1 was shown to perform poorly in the context of nonnegative tensor factorizations with or without inliers. Finding a robust alternative loss for count tensor data is yet another direction for future research.

In summary CPAL1 is an algorithmically well-behaved and simple-to-implement method to perform CP factorization when the data may include variations that violate the standard Gaussian assumption. Cases where CPAL1 succeeds over CPALS and even when CPAL1 fails to correctly recover generative factor models represent good first steps in improving the state of the art in tensor factorizations.

References

[1] E. Acar, D. M. Dunlavy, T. G. Kolda, and M. Mørup, *Scalable tensor factorizations for incomplete data*, Chemometrics and Intelligent Laboratory Systems, In Press, Corrected Proof (2010).

[2] B. W. Bader and T. G. Kolda, *MATLAB Tensor Toolbox Version 2.4*. `http://csmr.ca.sandia.gov/~tgkolda/TensorToolbox/`, March 2010.

[3] A. Basu, I. R. Harris, N. L. Hort, and M. Jones, *Robust and efficient estimation by minimising a density power divergence*, Biometrika, 85 (1998), pp. 549–559.

[4] D. Bertsekas, *Nonlinear programming*, Optimization and neural computation series, Athena Scientific, Belmont, Massachusetts, 1995.

[5] R. Bro, N. Sidiropoulos, and A. Smilde, *Maximum likelihood fitting using ordinary least squares algorithms*, Journal of Chemometrics, 16 (2002), pp. 387–400.

[6] 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.

[7] M. Collins, S. Dasgupta, and R. E. Schapire, *A generalization of principal component analysis to the exponential family*, in Advances in Neural Information Processing Systems, vol. 14, MIT Press, 2002, pp. 617–624.

[8] I. Dhillon and S. Sra, *Generalized nonnegative matrix approximations with bregman divergences*, in Advances in Neural Information Processing Systems, vol. 18, MIT Press, 2006, pp. 283–290.

[9] K. J. Friston, S. Williams, R. Howard, R. S. Frackowiak, and R. Turner, *Movement-related effects in fMRI time-series*, Magnetic Resonance in Medicine, 35 (1996), pp. 346–355.

[10] G. H. Golub and C. F. V. Loan, *Matrix Computations*, Johns Hopkins University Press, Baltimore, Maryland, 3rd ed., 1996.

[11] 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. Available at `http://www.psychology.uwo.ca/faculty/harshman/wpppfac0.pdf`.

[12] P. J. Huber and E. M. Ronchetti, *Robust Statistics*, John Wiley & Sons Inc, Hoboken, New Jersey, 2009.

[13] D. R. Hunter and K. Lange, *A Tutorial on MM Algorithms*, The American Statistician, 58 (2004), pp. pp. 30–37.

[14] T. G. Kolda, *Multilinear operators for higher-order decompositions*, Tech. Report SAND2006-2081, Sandia National Laboratories, Albuquerque, NM and Livermore, CA, April 2006.

[15] T. G. Kolda and B. W. Bader, *Tensor decompositions and applications*, SIAM Review, 51 (2009), pp. 455–500.

[16] J. B. Kruskal, *Three-way arrays: rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics*, Linear Algebra and its Applications, 18 (1977), pp. 95–138.

[17] K. Lange, *Numerical Analysis for Statisticians*, Springer, New York, New York, 2010.

[18] D. Lee and H. Seung, *Learning the parts of objects by non-negative matrix factorization*, Nature, 401 (1999), pp. 788–791.

[19] ——, *Algorithms for non-negative matrix factorization*, in Advances in Neural Information Processing Systems, vol. 13, MIT Press, 2001, pp. 556–562.

[20] L. Li, W. Huang, I. Y. Gu, and Q. Tian, *Statistical modeling of complex backgrounds for foreground object detection*, Image Processing, IEEE Transactions on, 13 (2004), pp. 1459 –1472.

[21] A. Schein, L. Saul, and L. Ungar, *A generalized linear model for principal component analysis of binary data*, in Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics, 2003, pp. 14–21.

[22] J. Shao, *Mathematical Statistics*, Springer, New York, New York, 2nd ed., July 2003.

[23] A. Smilde, R. Bro, and P. Geladi, *Multi-way Analysis with Applications in the Chemical Sciences*, Wiley, West Sussex, England, 2004.

[24] J. H. Stad, *Tensor rank is NP-complete*, Journal of Algorithms, 11 (1990), pp. 644–654.

[25] S. Vorobyov, Y. Rong, N. Sidiropoulos, and A. Gershman, *Robust iterative fitting of multilinear models*, Signal Processing, IEEE Transactions on, 53 (2005), pp. 2678–2689.

[26] M. Welling and M. Weber, *Positive tensor factorization*, Pattern Recognition Letters, 22 (2001), pp. 1255–1261.