Regularized Tensor Factorizations and Higher-Order Principal Components Analysis

Genevera I. Allen*

Abstract

High-dimensional tensors or multi-way data are becoming prevalent in areas such as biomedical imaging, chemometrics, networking and bibliometrics. Traditional approaches to finding lower dimensional representations of tensor data include flattening the data and applying matrix factorizations such as principal components analysis (PCA) or employing tensor decompositions such as the CANDECOMP / PARAFAC (CP) and Tucker decompositions. The former can lose important structure in the data, while the latter Higher-Order PCA (HOPCA) methods can be problematic in high-dimensions with many irrelevant features. We introduce frameworks for sparse tensor factorizations or Sparse HOPCA based on heuristic algorithmic approaches and by solving penalized optimization problems related to the CP decomposition. Extensions of these approaches lead to methods for general regularized tensor factorizations, multi-way Functional HOPCA and generalizations of HOPCA for structured data. We illustrate the utility of our methods for dimension reduction, feature selection, and signal recovery on simulated data and multi-dimensional microarrays and functional MRIs.

Keywords: tensor decompositions, principal components analysis, sparse PCA, functional PCA, generalized PCA, multi-way data

[0] Department of Pediatrics-Neurology, Baylor College of Medicine, Jan and Dan Duncan Neurological Research Institute, Texas Children's Hospital, & Department of Statistics, Rice University, MS 138, 6100 Main St., Houston, TX 77005 (email: gallen@rice.edu)

1 Introduction

Recently high-dimensional tensor data has become prevalent in areas of biomedical imaging, remote sensing, bibliometrics, chemometrics and the Internet. We define high-dimensional tensor data to be multi-dimensional data, with three or more dimensions or *tensor modes*, in which the dimension of one or more modes is large compared to the remainder. Examples of this include functional magnetic resonance imaging (fMRI) in which three-dimensional images of the brain measured in voxels (when vectorized, these form mode 1) are taken over time (mode 2) for several experimental conditions (mode 3) and for a set of independent subjects (mode 4). Here, the "samples" in this data set are the independent subjects ($\approx$ 30) or experimental tasks ($\approx$ 20), the dimension of which are small compared to the number of voxels ($\approx$ 60,000) and time points ($\approx$ 2,000). Exploratory analysis, including dimension reduction, pattern recognition, visualization, and feature selection, for this massive tensor data is a major mathematical and computational challenge. In this paper, we propose flexible unsupervised multivariate methods, specifically regularized Higher-Order Principal Components Analysis (HOPCA), to understand and explore high-dimensional tensor data.

Classically, multivariate methods such as principal components analysis (PCA) have been applied to matrix data in which the number of independent observations n is larger than the number of variables p. Exploring tensor data using these methods would require flattening the tensor into a matrix with one dimension enormous compared to the other. This approach is not ideal as to begin with, the important multi-dimensional structure of the tensor is lost. To solve this, many use traditional tensor decompositions such as the CANDECOMP / PARAFAC (CP) (Harshman, 1970; Carroll and Chang, 1970) and Tucker decompositions (Tucker, 1966) to compute HOPCA (Kolda and Bader, 2009). These factorizations seek to approximate the tensor by a low-rank set of factors along each of the tensor modes. Unlike the singular value decomposition (SVD) for matrix data, these tensor decompositions do not uniquely decompose the data (Kolda and Bader, 2009) further complicating the analysis of tensor data. Especially in the statistics community, there has been relatively little work done on methodology for high-dimensional tensor data (McCullagh, 1987; Kroonenberg, 2008; Hoff, 2011).

Many, especially in applied mathematics, have studied tensor decompositions, but relatively few have advocated encouraging sparsity or other types of regularization in the tensor factors. The exception is the non-negative tensor factorization, which is a well developed area and has been used to cluster tensor data (Cichocki et al., 2009). In the context of these non-negative factorizations, several have discussed sparsity (Hazan et al., 2005; Mørup et al., 2008; Lim and Comon, 2009; Cichocki et al., 2009; Liu et al., 2012; Chi and Kolda, 2011), and a few have gone on to explicitly encourage sparsity in one of the tensor factors (Ruiters and Klein, 2009; Pang et al., 2011). A general framework for sparsity and other types of flexible regularization in tensor factorizations and HOPCA is lacking.

Sparsity in tensor decompositions and HOPCA is desirable for many reasons. First, tensor decompositions are often used to compress large multi-dimensional data sets (Kolda and Bader, 2009). Sparsity in the tensor factors compresses the data further and is attractive from a data storage perspective. Second, in high-dimensional settings, many features are often irrelevant. With fMRIs, for example, there are often hundreds of thousands of voxels in each image with many voxels inactive for much of the scanning session. Sparsity gives one an automatic tool for feature selection. Third, many have noted that PCA is asymptotically inconsistent in high-dimensional settings (Johnstone and Lu, 2009; Jung and Marron, 2009). As this is true for matrix data, it is not hard to surmise that asymptotic inconsistency of the factors holds for HOPCA as well. Sparsity in PCA, however, has been shown to yield consistent PC directions (Johnstone and Lu, 2009; Amini and Wainwright, 2009). Finally for high-dimensional tensors, visualizing and interpreting the HOPCs can be a challenge. Sparsity limits the number of features and hence simplifies visualization and interpretation of exploratory data analysis results. Similar arguments can be made for smoothing the tensor factors or encouraging other types of regularization.

In this paper, we seek a mathematically sound and computationally attractive framework for regularizing one, two, or all of the tensor factors, or HOPCs. Our approach is based on optimization theory. That is, we seek to find a coherent optimization criterion for regularized HOPCA related to traditional tensor decomposition approaches and an algorithm that converges to a solution to this optimization problem. Recently, many have proposed similar optimization-based approaches to perform regularized PCA. Beginning with Jolliffe et al. (2003), several have proposed to find Sparse PCs by solving an optimization problem related the PCA with an additional ℓ_1-norm penalty (Zou et al., 2006; Shen and Huang, 2008). Others have shown that Functional PCA can be achieved by solving a PCA optimization problem with a penalty to induce smoothness (Silverman, 1996; Huang et al., 2008). More recently, several have extended these optimization approaches to two-way penalized SVDs by placing penalties on both the row and column factors simultaneously (Witten et al., 2009; Huang et al., 2009; Lee et al., 2010; Allen et al., 2011). Specifically, our approach to regularized HOPCA most closely follows the work of Shen and Huang (2008) and later Witten et al. (2009) and Allen et al. (2011) who iteratively solve penalized regression problems converging to a rank-one solution with subsequent factors found via a greedy deflation method.

To limit the scope of our exploration, we first develop a framework for Sparse HOPCA and later show that extensions of this framework yield methods for regularization with general penalties, Functional HOPCA, and Generalized HOPCA for structured tensor data. Additionally, as the concept of regularization in HOPCA is relatively undeveloped, we employ a less common strategy by first introducing several flawed approaches to Sparse HOPCA. We do this for a number of reasons, namely to illuminate these problematic paths for future researchers, to give us a basis for comparison for our optimization-based methods, and to provide further background and justification for our approach. Our optimization-based approach directly solves a multi-way concave relaxation of a rank-one HOPCA method. Overall, this method has many mathematical and computational advantages including guaranteed convergence to a local optimum.

This paper is organized as follows. As the general statistical audience may be unfamiliar with tensors and tensor decompositions for HOPCA, we begin by introducing notation, reviewing these decompositions, and discussing these in the context of PCA in Section 2.

We take a first step at considering how to introduce sparsity in this context in Section 3 by walking the reader through several novel (but, flawed!) algorithmic approaches to Sparse HOPCA. Then, in Section 4, we introduce our main novel method and algorithm for Sparse HOPCA, based on deflation and the CP factorization. In Section 5, we outline several novel extensions of this approach to allow flexible modeling with general penalties and functional or structured tensor data. Simulation studies are presented in Section 6 followed by two case studies to high-dimensional three-way microarray data and an fMRI study. We conclude with a discussion, Section 7, of the implications of our work as well as the many directions for future work on tensor data.

2 Preliminaries: Tensor Decompositions

In this section, we review the two major approaches to tensor decompositions and HOPCA, the CANDECOMP / PARAFAC (CP) and Tucker decompositions. We also introduce notation and discuss considerations for evaluating HOPCA such as the amount of variance explained.

As notation with multi-way data can be cumbersome, we begin by reviewing the notation used throughout this paper, mostly adopted from Kolda and Bader (2009). Tensors will be denoted as $\boldsymbol{\mathcal{X}}$, matrices as $\mathbf{X}$, vectors as $\mathbf{x}$ and scalars as x. As there are many types of multiplication with tensors, the outer product will be denoted by $\circ$, $\mathbf{x} \circ \mathbf{y} = \mathbf{x}\,\mathbf{y}^T$. Specific dimensions of the tensor will be called modes and multiplication by a matrix or vector along a tensor mode will be denoted as $\times_1$; here, the subscript refers to the mode being multiplied using regular matrix multiplication. For example, if $\boldsymbol{\mathcal{X}} \in \Re^{n \times p \times q}$ and $\mathbf{U} \in \Re^{n \times K}$, then $\boldsymbol{\mathcal{X}} \times_1 \mathbf{U} \in \Re^{K \times p \times q}$. Sometimes it is necessary to flatten the tensor into a matrix, or matricize the tensor. This is denoted as $\mathbf{X}_{(1)}$ where the subscript indicates the mode along which the matricization has occurred. For example, if $\boldsymbol{\mathcal{X}} \in \Re^{n \times p \times q}$, then $\mathbf{X}_{(1)} \in \Re^{n \times pq}$. The tensor Frobenius norm, $||\,\boldsymbol{\mathcal{X}}\,||_F$, refers to $||\,\boldsymbol{\mathcal{X}}\,||_F = \sqrt{\sum_i \sum_j \sum_k \boldsymbol{\mathcal{X}}_{ijk}^2}$. Two types of matrix multiplication that will be important in this paper are the Kronecker product and the perhaps less familiar Khatri-Rao product. The former is denoted by $\otimes$ and the latter by $\odot$. For matrices $\mathbf{U} \in \Re^{n \times K}$ and $\mathbf{V} \in \Re^{p \times K}$, the Khatri-Rao product is defined as $\mathbf{U} \odot \mathbf{V} = [\mathbf{u}_1 \otimes \mathbf{v}_1 \quad \dots \quad \mathbf{u}_K \otimes \mathbf{v}_K]$. For notational simplicity, all results in this paper will be presented for the three-mode tensor. Our methods can all be trivially extended to multi-dimensional tensors with an arbitrary number of modes.

As discussed in the introduction, there is no equivalent to the SVD for multi-way data with three or more dimensions. Thus, for tensor data, people commonly use one of two tensor decompositions that capture desirable aspects of the SVD. The CP decomposition seeks to model a tensor as a sum of rank one tensors, where rank one three-mode tensors are defined as an outer product of three vectors: $\boldsymbol{\mathcal{X}} \approx \sum_{k=1}^{K} d_k \, \mathbf{u}_k \circ \mathbf{v}_k \circ \mathbf{w}_k$, where $\mathbf{u}_k \in \Re^n$, $\mathbf{v}_k \in \Re^p$, $\mathbf{w}_k \in \Re^q$ and $d_k \geq 0$ (Harshman, 1970; Carroll and Chang, 1970). In this paper, we will assume that the three vectors have norm one, i.e. $\mathbf{u}^T \mathbf{u} = 1$, and define the factor matrices as $\mathbf{U} \in \Re^{n \times K} = [\mathbf{u}_1 \quad \dots \quad \mathbf{u}_K]$ and so forth. Thus, similar to the SVD, the CP seeks to factorize the data into a sum of rank-one arrays, although unlike the SVD these rank-one arrays do not uniquely decompose the data and in general, are not orthonormal. The Tucker decomposition, sometimes referred to as the Higher-Order SVD (HOSVD) or HOPCA, seeks to model a three-mode tensor as $\boldsymbol{\mathcal{X}} \approx \boldsymbol{\mathcal{D}} \times_1 \mathbf{U} \times_2 \mathbf{V} \times_3 \mathbf{W}$ where the factors $\mathbf{U} \in \Re^{n \times K_1}$, $\mathbf{V} \in \Re^{p \times K_2}$ and $\mathbf{W} \in \Re^{q \times K_3}$ are orthonormal and $\boldsymbol{\mathcal{D}} \in \Re^{K_1 \times K_2 \times K_3}$ is the core tensor (Tucker, 1966; De Lathauwer et al., 2000). Hence, the factors of the Tucker decomposition are orthonormal, similar to those of the SVD, but these orthonormal factors do not uniquely decompose the array into a sum of rank-one factors with a diagonal core.

Both the CP and Tucker decompositions can be used for tensor data in a similar manner to PCA for matrix data. In other words, one can think of the tensor factors as major modes of variation or patterns in the data that can be used for exploratory analysis, visualization, and dimension reduction. A critical quantity in assessing the dimension reduction achieved for PCA is the amount of variance explained by each of the SVD factors. As existing tensor factorizations and the methods we will propose in this paper do not uniquely decompose the data array, we cannot simply sum the equivalent of the squared singular values to measure the variance explained. Since the Tucker decomposition imposes orthonormality on the factors, the proportion of variance explained by the decomposition has a simple form: $\|\, \boldsymbol{\mathcal{D}} \,\|_F^2 / \|\, \boldsymbol{\mathcal{X}} \,\|_F^2$ (De Lathauwer et al., 2000; Kolda and Bader, 2009). This is not the case for the CP decomposition where many have erroneously referred the the variance explained by the k^{th} component as $d_k^2 / \|\, \boldsymbol{\mathcal{X}} \,\|_F^2$. Instead, as the factors of the CP decomposition may be correlated, we must project out these effects to determine the proportion of variance explained:

Theorem 1. *Define* $\mathbf{P}_k^{(U)} = \mathbf{U}_k (\mathbf{U}_k^T \mathbf{U}_k)^{-1} \mathbf{U}_k^T$ *where* $\mathbf{U}_k = [\mathbf{u}_1, \dots \mathbf{u}_k]$ *and define* $\mathbf{P}_k^{(V)}$ *and* $\mathbf{P}_k^{(W)}$ *analogously. Then, the cumulative proportion of variance explained by the first k HOPCs (or regularized HOPCs) is given by* $\|\, \boldsymbol{\mathcal{X}} \times_1 \mathbf{P}_k^{(U)} \times_2 \mathbf{P}_k^{(V)} \times_3 \mathbf{P}_k^{(W)} \,\|_F^2 \;/\; \|\, \boldsymbol{\mathcal{X}} \,\|_F^2$.

As our regularized tensor decompositions presented in the next two sections will not impose orthogonality on the factors, this result is critical in evaluating the dimension reduction achieved by both the CP decomposition and our regularized HOPCA methods.

3 Algorithmic Approaches to Sparse HOPCA

We begin our exploration of methods for finding Sparse Higher-Order Principal Components by proposing three algorithmic approaches. These methods are direct extensions of popular algorithms for the CP and Tucker decompositions that one might first think of when seeking sparsity. We will show, however, that while these approaches are intuitively simple, they are problematic both mathematically and computationally as they do not solve a coherent mathematical objective. Despite this, these methods are important to present to both steer future investigators from problematic paths and provide background, rationale, and a basis for comparison to our subsequent optimization-based Sparse HOPCA methods. Here, we walk through each of these algorithmic approaches; outlines of the full algorithms are given in the Supplemental Materials.

The Higher-Order SVD (HOSVD) and Higher-Order Orthogonal Iteration (HOOI), sometimes referred to as the Tucker Alternating Least Squares (Tucker-ALS), are two popular algorithms for computing the Tucker decomposition (Tucker, 1966; De Lathauwer et al., 2000;

Kolda and Bader, 2009). The former estimates each factor matrix by calculating the singular vectors of the tensor matricized along each mode (Tucker, 1966; De Lathauwer et al., 2000). In other words for a three-mode tensor, the HOSVD can be found by performing PCA three times on data flattened along each of the three dimensions. The HOOI seeks to improve upon this decomposition by minimizing a Frobenius norm loss between $\boldsymbol{\mathcal{X}}$ and the Tucker decomposition $\boldsymbol{\mathcal{D}} \times_1 \mathbf{U}^T \times_2 \mathbf{V}^T \times_3 \mathbf{W}^T$ subject to orthogonality constraints on the factors. De Lathauwer et al. (2000) showed that this problem is solved by an iterative algorithm where each Tucker factor is estimated by computing the singular vector of the tensor projected onto the other factors. In other words, the HOOI updates $\mathbf{U}$ by performing PCA on the matricized $(\boldsymbol{\mathcal{X}} \times_2 \mathbf{V} \times_3 \mathbf{W})_{(1)}$. Thus, both the HOSVD and HOOI algorithms estimate the factors by performing PCA on a matricized tensor. This leads to a simple strategy for obtaining Sparse HOSVD and Sparse HOOI: Replace PCA with Sparse PCA in each algorithm to obtain sparse factors for each tensor mode. Many algorithms exist for performing Sparse PCA (Jolliffe et al., 2003; Zou et al., 2006; Shen and Huang, 2008; Johnstone and Lu, 2009; Journée et al., 2010), any of which can be used to compute the Sparse HOSVD or Sparse HOOI.

While these methods for Sparse HOPCA based on the Tucker decomposition are conceptually simple, they are not ideal for several reasons. First, the methods are purely heuristic algorithmic approaches and do not solve any coherent mathematical optimization problem. For the Sparse HOOI method, this is particularly problematic as one cannot guarantee convergence of the algorithm without additional constraints. Secondly in high-dimensional settings, matricizing the tensor along each mode and performing Sparse PCA is computationally intensive and requires large amounts of computer memory. Employing the Sparse PCA methods of Jolliffe et al. (2003); Zou et al. (2006) requires forming and computing the leading sparse eigenvalues of $n \times n$, $p \times p$, and $q \times q$ matrices, which in high-dimensional settings are typically much larger than the original data array. The methods of Shen and Huang (2008); Journée et al. (2010) require computing the sparse singular vectors of $n \times pq$, $p \times nq$, and $q \times np$, which corresponds to calculating several unnecessary and extremely large singular vectors. Hence, even though the Sparse HOSVD and HOOI are attractive in their simplicity, mathematically and computationally they are less desirable.

Next, we consider incorporating sparsity in the CP decomposition through the popular CP Alternating Least Squares (CP-ALS) Algorithm. This algorithm updates one factor at a time by minimizing a Frobenius norm loss with the other factors fixed (Harshman, 1970; Carroll and Chang, 1970; Kolda and Bader, 2009). Consider, for example, solving for $\mathbf{U}$ with $\mathbf{V}$ and $\mathbf{W}$ fixed. Then, the loss function can be written as $||\, \boldsymbol{\mathcal{X}} - \sum_{k=1}^{K} d_k \, \mathbf{u}_k \circ \mathbf{v}_k \circ \mathbf{w}_k \, ||_F^2 = ||\, \mathbf{X}_{(1)} - \hat{\mathbf{U}}(\mathbf{V} \odot \mathbf{W})^T ||_F$, where $\hat{\mathbf{U}} = \mathbf{U} \, \text{diag}(\mathbf{d})$ (Harshman, 1970; Carroll and Chang, 1970). Thus, the CP-ALS algorithm estimates each factor sequentially by performing simple least squares and normalizing the columns of the factor matrix to have norm one, i.e. $\hat{d}_k = ||\hat{\mathbf{u}}_k||$. A simple strategy for encouraging sparsity is then to replace the least squares problem with an ℓ_1-norm penalized least squares, or LASSO problem (Tibshirani, 1996). In other words, to estimate a sparse $\mathbf{U}$, we minimize $||\, \mathbf{X}_{(1)} - \hat{\mathbf{U}}(\mathbf{V} \odot \mathbf{W})^T ||_F + \lambda_{\mathbf{u}} \, ||\hat{\mathbf{U}}||_1$, where $\lambda_{\mathbf{u}}$ is a non-negative regularization parameter and $|| \cdot ||_1 = \sum \sum | \cdot |$ is the ℓ_1-norm. Hence, a possible method for Sparse HOPCA is to replace each least squares update in the CP-ALS algorithm with a LASSO update; we call this the Sparse CP-ALS method.

A first glance, it seems that the Sparse CP-ALS is less heuristic than the Sparse HOSVD and HOOI methods. As each update solves a LASSO regression problem, it is natural to presume that the algorithm jointly minimizes the Frobenius norm loss with ℓ_1 penalties on each of the factors:

$$\begin{aligned} &\underset{\mathbf{U},\mathbf{V},\mathbf{W}}{\text{minimize}} \;\; \frac{1}{2} ||\, \boldsymbol{\mathcal{X}} - \sum_{k=1}^{K} d_k \, \mathbf{u}_k \circ \mathbf{v}_k \circ \mathbf{w}_k \, ||_F^2 + \lambda_{\mathbf{u}} \, ||\, \mathbf{U} \,||_1 + \lambda_{\mathbf{v}} \, ||\, \mathbf{V} \,||_1 + \lambda_{\mathbf{w}} \, ||\, \mathbf{W} \,||_1 \\ &\text{subject to} \;\; d_k \geq 0, \;\; \mathbf{u}_k^T \mathbf{u}_k = 1, \;\; \mathbf{v}_k^T \mathbf{v}_k = 1, \;\; \& \;\; \mathbf{w}_k^T \mathbf{w}_k = 1, \; \forall \; k = 1, \dots K. \end{aligned} \qquad (1)$$

Interestingly, this is not the case!

Proposition 1. *The Sparse CP-ALS algorithm does not minimize* (1)*.*

In fact, it can be shown that the Sparse CP-ALS does not optimize any coherent objective.

Similar results, while not proved explicitly, have been implied for two-way penalized SVDs (Witten et al., 2009; Lee et al., 2010; Allen et al., 2011). Thus, similar to the Sparse HOSVD and HOOI methods, convergence of the Sparse CP-ALS algorithm cannot be guaranteed without further constraints. In the next section, we present a framework for incorporating sparsity in the CP decomposition with provable results in terms of convergence to a solution of an optimization problem.

4 Deflation Approach to Sparse HOPCA

In this section, we develop a novel deflation approach to Sparse HOPCA that find each rank-one factorization by solving a tri-concave relaxation of the CP optimization problem. But first, we show how the CP decomposition can be obtained by an alternative greedy algorithm, the deflation-based Tensor Power Algorithm.

4.1 Tensor Power Algorithm

We introduce an alternative form of the rank-one CP optimization problem and a corresponding algorithm that will form the foundation of our deflation approach to Sparse HOPCA. The single-factor CP decomposition solves the following optimization problem (Kolda and Bader, 2009):

$$\underset{\mathbf{u},\mathbf{v},\mathbf{w},d}{\text{minimize}} \ \ || \, \boldsymbol{\mathcal{X}} - d \, \mathbf{u} \circ \mathbf{v} \circ \mathbf{w} \, ||_2^2 \ \text{ subject to } \ \mathbf{u}^T \mathbf{u} = 1, \mathbf{v}^T \mathbf{v} = 1, \mathbf{w}^T \mathbf{w} = 1, \ \& \ d > 0. \tag{2}$$

Some algebra manipulation (see Kolda and Bader (2009)) shows that an equivalent form to this optimization problem is given by the following:

$$\underset{\mathbf{u},\mathbf{v},\mathbf{w}}{\text{maximize}} \ \ \boldsymbol{\mathcal{X}} \times_1 \mathbf{u} \times_2 \mathbf{v} \times_3 \mathbf{w} \ \text{ subject to } \ \mathbf{u}^T \mathbf{u} = 1, \mathbf{v}^T \mathbf{v} = 1, \ \& \ \mathbf{w}^T \mathbf{w} = 1. \tag{3}$$

As (3) is separable in the factors, we can optimize this in an iterative block-wise manner:

Proposition 2. *The block coordinate-wise solutions for* (3) *are given by:*

$$\hat{\mathbf{u}} = \frac{\boldsymbol{\mathcal{X}} \times_2 \mathbf{v} \times_3 \mathbf{w}}{||\, \boldsymbol{\mathcal{X}} \times_2 \mathbf{v} \times_3 \mathbf{w} \,||_2}, \hat{\mathbf{v}} = \frac{\boldsymbol{\mathcal{X}} \times_1 \mathbf{u} \times_3 \mathbf{w}}{||\, \boldsymbol{\mathcal{X}} \times_1 \mathbf{u} \times_3 \mathbf{w} \,||_2}, \ \& \ \hat{\mathbf{w}} = \frac{\boldsymbol{\mathcal{X}} \times_1 \mathbf{u} \times_2 \mathbf{v}}{||\, \boldsymbol{\mathcal{X}} \times_1 \mathbf{u} \times_2 \mathbf{v} \,||_2}.$$

Updated iteratively, these converge to a local optimum of (3)*.*

As each coordinate update increases the objective and the objective is bounded above by d, convergence of this scheme is assured. Note, however, that this approach only converges to a local optimum of (3), but this is true of all other algorithmic approaches to solving the CP problem as well (Kolda and Bader, 2009).

To compute multiple CP factors, one could apply this single-factor approach sequentially to the residuals remaining after subtracting out the previously computed factors. This so called deflation approach is closely related in structure to the power method for computing eigenvectors (Golub and Van Loan, 1996). We then call this greedy method the *Tensor Power Algorithm*. Notice that this method does not enforce orthogonality in subsequently computed components. The algorithm can be easily modified, however, to ensure orthogonality by employing a Graham-Schmidt scheme (Golub and Van Loan, 1996).

Before moving on to our Sparse CP method, we pause to discuss the Tensor Power Method and compare it to more common algorithms to compute the CP decomposition such as the Alternating Least Squares algorithm (Harshman, 1970; Carroll and Chang, 1970). As the Tensor Power Algorithm is a greedy approach, the first several factors computed will tend to explain the most variance in the data. In contrast, the CP-ALS algorithm seeks to maximize the set of d_k for all K factors simultaneously. Thus, the set of K CP-ALS factors may explain as much variance as those of the Tensor Power Algorithm, but the first several factors typically explain much less. This is illustrated on the tensor microarray data in Section 6. Also, while the CP-ALS algorithm returns d_k in descending order, the d_k computed via the Tensor Power Method are not necessarily ordered.

4.2 Sparse HOPCA via Deflation

We introduce a novel deflation-based method for Sparse HOPCA that incorporates sparsity by regularizing factors with an ℓ_1-norm penalty. Our method solves a direct relaxation of the CP optimization problem (3) and has a computationally attractive solution.

Many have sought to encourage sparsity in PCA by greedily solving ℓ_1-norm penalized optimization problems related the to rank-one SVD (Jolliffe et al., 2003; Zou et al., 2006; Shen and Huang, 2008; Witten et al., 2009; Lee et al., 2010; Journée et al., 2010; Allen et al., 2011). We formulate our method similarly by placing ℓ_1-norm penalties on each of the tensor factors of the CP optimization problem (3) and relaxing the equality constraints to inequalities. In this manner, our approach is most similar to the two-way Sparse PCA methods of Witten et al. (2009) and Allen et al. (2011). We define our single-factor Sparse CP-TPA (named to denote its relation to our Tensor Power Algorithm) factorization as the solution to the following problem:

$$\begin{aligned} &\underset{\mathbf{u},\mathbf{v},\mathbf{w}}{\text{maximize}} \quad \boldsymbol{\mathcal{X}} \times_1 \mathbf{u} \times_2 \mathbf{v} \times_3 \mathbf{w} - \lambda_{\mathbf{u}} \,||\, \mathbf{u} \,||_1 - \lambda_{\mathbf{v}} \,||\, \mathbf{v} \,||_1 - \lambda_{\mathbf{w}} \,||\, \mathbf{w} \,||_1 \\ &\text{subject to} \quad \mathbf{u}^T \mathbf{u} \leq 1, \mathbf{v}^T \mathbf{v} \leq 1, \;\&\; \mathbf{w}^T \mathbf{w} \leq 1. \end{aligned} \tag{4}$$

Here $\lambda_{\mathbf{u}}$, $\lambda_{\mathbf{v}}$ and $\lambda_{\mathbf{w}}$ are non-negative regularization parameters controlling the amount of sparsity in the tensor factors. Relaxing the constraints greatly simplifies the optimization problem, leading to a simple solution and algorithmic approach:

Theorem 2. *Let* $\hat{\mathbf{u}} = S(\boldsymbol{\mathcal{X}} \times_2 \mathbf{v} \times_3 \mathbf{w}, \lambda_{\mathbf{u}})$*,* $\hat{\mathbf{v}} = S(\boldsymbol{\mathcal{X}} \times_1 \mathbf{u} \times_3 \mathbf{w}, \lambda_{\mathbf{v}})$*, and* $\hat{\mathbf{w}} = S(\boldsymbol{\mathcal{X}} \times_1 \mathbf{u} \times_2 \mathbf{v}, \lambda_{\mathbf{w}})$ *where* $S(\cdot, \lambda) = \text{sign}(\cdot)(|\cdot| - \lambda)_+$ *is the soft-thresholding operator. Then, the factor-wise solutions to the Sparse CP-TPA problem are given by:*

$$\mathbf{u}^* = \begin{cases} \frac{\hat{\mathbf{u}}}{||\hat{\mathbf{u}}||_2} & ||\hat{\mathbf{u}}||_2 > 0 \\ 0 & \textit{otherwise}, \end{cases} \quad \mathbf{v}^* = \begin{cases} \frac{\hat{\mathbf{v}}}{||\hat{\mathbf{v}}||_2} & ||\hat{\mathbf{v}}||_2 > 0 \\ 0 & \textit{otherwise}, \end{cases} \quad \& \,\mathbf{w}^* = \begin{cases} \frac{\hat{\mathbf{w}}}{||\hat{\mathbf{w}}||_2} & ||\hat{\mathbf{w}}||_2 > 0 \\ 0 & \textit{otherwise}, \end{cases}$$

Each factor-wise update monotonically increases the objective and when iterated, they con-

verge to a local maximum of (4)*.*

Thus, (4) can be solved a factor at a time by soft-thresholding and then re-scaling the update. While this type of algorithmic scheme may seem familiar to some, a major contribution of our work is determining the underlying optimization problem being solved by this algorithmic structure.

There are many mathematical and computational advantages to this approach. First, notice that (4) is a concave optimization problem in each factor with the other factors fixed. Thus, it is a multi-way concave optimization problem, meaning that we can solve for one factor at a time with the others fixed. This scheme yields a monotonic algorithm that converges to a local maximum of (4) (Tseng, 2001). This is important as, unlike our algorithmic based Sparse HOPCA methods, the progress of the algorithm can be tracked and convergence is assured. Second, notice that in (4), the scale of the factors is directly controlled. Even with the relaxed constraints, the solution for each factor is guaranteed to either have norm one or be set to zero. This is a major computational advantage as it leads to numerically well-conditioned algorithmic updates and solutions. Third, this method is computationally attractive as each factor-wise update has a simple analytical solution, and thus each update is computationally inexpensive. Additionally, this scheme requires far less computer memory than our algorithmic Sparse HOPCA approaches as only factors comprising the final solution need to be computed, and the tensor never needs to be matricized. The final advantage of this approach is its generality; as we will show in the next section, there are many possible extensions of this methodology.

To compute multiple factors, we employ a deflation approach as in the Tensor Power Algorithm. Specifically, each factor is calculated by solving the single-factor CP problem, (4), for the residuals from the previously computed single-factor solutions. Notice that we do not enforce orthogonality in the factors. In fact, many have advocated against enforcing orthogonality of sparse PCs (Zou et al., 2006; Shen and Huang, 2008; Journée et al., 2010). For factors with $\lambda = 0$, however, if orthogonality is desired, this can be accomplished by altering the factor updates as described in Section 4.1. Thus, we have developed a deflation approach to Sparse HOPCA by greedily solving single-factor optimization problems related to the CP criterion. This Sparse CP-TPA method enjoys mathematical advantages such as assured convergence to a local optimum and is computationally attractive for high-dimensional tensors.

4.2.1 Selecting Dimension and Regularization Parameters

Another important item of consideration for our Sparse CP-TPA method and all of our Sparse HOPCA methods is the number of factors, K, to compute for a given data sets. Several have proposed heuristic methods for choosing K in HOPCA (see Kolda and Bader (2009) and Kroonenberg (2008) and references therein). While several non-heuristic methods exist for PCA on matrix data (Buja and Eyuboglu, 1992; Owen and Perry, 2009), extending these to the tensor framework is non-trivial and beyond the scope of this paper.

Our Sparse CP-TPA problem has three regularization parameters that control the amount of sparsity in the factors. Several methods exist for selecting these regularization parameters in the Sparse PCA literature (Troyanskaya et al., 2001; Owen and Perry, 2009; Shen and Huang, 2008; Lee et al., 2010). As cross-validation can be slow to run for high-dimensional tensors, we choose to select regularization parameters via the Bayesian Information Criterion (BIC) (Lee et al., 2010; Allen et al., 2011): $\lambda_{\mathbf{u}}^{*} = \text{argmin}_{\lambda_{\mathbf{u}}} BIC(\lambda_{\mathbf{u}})$ where $BIC(\lambda_{\mathbf{u}}) = \log\left(\frac{||\boldsymbol{\mathcal{X}} - d\,\mathbf{u} \circ \mathbf{v} \circ \mathbf{w}||_F^2}{npq}\right) + \frac{\log(npq)}{npq}|\{\mathbf{u}\}|$, where $|\{\mathbf{u}\}|$ is the number of non-zero elements of $\mathbf{u}$. This BIC formulation can be derived by considering that each update of the Sparse CP-TPA algorithm solves an ℓ_1-norm penalized regression problem. Selection criteria for $\mathbf{v}$ and $\mathbf{w}$ are analogous. Similar BIC-based selection methods can be derived for our previous algorithmic Sparse HOPCA methods as well. Experimental results evaluating the efficacy of this regularization parameter selection method are given in Section 6.1.

5 Extensions

As our frameworks introduced for Sparse HOPCA are general, there are many possible extensions of our methodology. In this section, we outline novel results and methodology for three extensions we consider most important: general penalties and non-negativity, generalizations for structured tensors, and multi-way functional HOPCA. Our intent here is to refrain from fully describing each development, but instead to give the reader enough details to understand and use each of these extensions.

5.1 General Penalties & Non-negativity

In certain applications, one may wish to regularize tensor factors with a penalty other than an ℓ_1-norm. Consider the following optimization problem which incorporates general penalties, $P_{\mathbf{u}}()$, $P_{\mathbf{v}}()$ and $P_{\mathbf{w}}()$:

$$\begin{aligned} \underset{\mathbf{u},\mathbf{v},\mathbf{w}}{\text{maximize}} \quad & \boldsymbol{\mathcal{X}} \times_1 \mathbf{u} \times_2 \mathbf{v} \times_3 \mathbf{w} - \lambda_{\mathbf{u}}\, P_{\mathbf{u}}(\mathbf{u}) - \lambda_{\mathbf{v}}\, P_{\mathbf{v}}(\mathbf{v}) - \lambda_{\mathbf{w}}\, P_{\mathbf{w}}(\mathbf{w}) \\ \text{subject to} \quad & \mathbf{u}^T \mathbf{u} \leq 1, \mathbf{v}^T \mathbf{v} \leq 1, \;\&\; \mathbf{w}^T \mathbf{w} \leq 1. \end{aligned} \tag{5}$$

An extension of a result in Allen et al. (2011) reveals that one may solve this optimization problem for general penalties that are convex and order one by solving penalized regression problems:

Theorem 3. *Let $P_{\mathbf{u}}()$, $P_{\mathbf{v}}()$ and $P_{\mathbf{w}}()$ be convex and homogeneous or order one. Consider the following penalized regression problems: $\hat{\mathbf{u}} = argmin_{\mathbf{u}}\{\frac{1}{2}||\, \boldsymbol{\mathcal{X}} \times_2 \hat{\mathbf{v}} \times_3 \hat{\mathbf{w}} - \mathbf{u}\,||_2^2 + \lambda_{\mathbf{u}}\, P_{\mathbf{u}}(\mathbf{u})\}$, $\hat{\mathbf{v}} = argmin_{\mathbf{v}}\{\frac{1}{2}||\, \boldsymbol{\mathcal{X}} \times_1 \hat{\mathbf{u}} \times_3 \hat{\mathbf{w}} - \mathbf{v}\,||_2^2 + \lambda_{\mathbf{v}}\, P_{\mathbf{v}}(\mathbf{v})\}$, and $\hat{\mathbf{w}} = argmin_{\mathbf{w}}\{\frac{1}{2}||\, \boldsymbol{\mathcal{X}} \times_1 \hat{\mathbf{u}} \times_2 \hat{\mathbf{v}} - \mathbf{w}\,||_2^2 + \lambda_{\mathbf{w}}\, P_{\mathbf{w}}(\mathbf{w})\}$. Then, the block coordinate-wise solutions for (5) are given by:*

$$\mathbf{u}^* = \begin{cases} \frac{\hat{\mathbf{u}}}{||\hat{\mathbf{u}}||_2} & ||\hat{\mathbf{u}}||_2 > 0 \\ 0 & otherwise \end{cases}, \mathbf{v}^* = \begin{cases} \frac{\hat{\mathbf{v}}}{||\hat{\mathbf{v}}||_2} & ||\hat{\mathbf{v}}||_2 > 0 \\ 0 & otherwise \end{cases}, \& \, \mathbf{w}^* = \begin{cases} \frac{\hat{\mathbf{w}}}{||\hat{\mathbf{w}}||_2} & ||\hat{\mathbf{w}}||_2 > 0 \\ 0 & otherwise \end{cases}.$$

Iteratively updating these factors converges to a local optimum of (5).

An example of a possible penalty type of interest in many tensor applications is the group lasso, which encourages sparsity in groups of variables (Yuan and Lin, 2006).

Much attention in the literature has been given to the non-negative and sparse non-negative tensor decompositions (Hazan et al., 2005; Shashua and Hazan, 2005; Mørup et al., 2008; Cichocki et al., 2009; Liu et al., 2012). These techniques have been used for multi-way clustering of tensor data. A simple modification of Theorem 3 allows us to solve (4) when non-negativity constraints are added for each factor: Replace the soft-thresholding function $S(x, \rho) = \text{sign}(x)(|x| - \rho)_+$ with the positive-thresholding function $P(x, \rho) = (x - \rho)_+$ (Allen and Maletić-Savatić, 2011). This, then, is a computationally attractive alternative to estimating sparse non-negative tensor factors. Extensions of our deflation approach for Sparse CP-TPA allow for regularized HOPCA with general penalties and non-negativity constraints.

5.2 Generalized HOPCA

For structured data, Allen et al. (2011) showed that generalizing PCA by working with an alternative matrix norm capturing the known structure can dramatically improve results. The same techniques generalizing PCA can be used to generalized HOPCA for structured tensors. Examples of these include image data as seen in neuroimaging, microscopy and hyper-spectral imaging as well as multi-dimensional NMR spectroscopy, remote sensing, and environmetrics. Here, we illustrate how to extend Generalized PCA to the framework introduced for Sparse CP-TPA. Generalized Tucker, Sparse HOSVD, and Sparse HOOI methods are a straightforward extension of these approaches.

The Generalized PCA optimization problem of Allen et al. (2011) replaces minimizing a Frobenius norm with a transposable-quadratic norm comprised of two quadratic operators in the row and column space of the data matrix. To generalize a three-way Frobenius norm, let $\mathbf{Q}^{(1)} \in \Re^{n \times n}$, $\mathbf{Q}^{(2)} \in \Re^{p \times p}$ and $\mathbf{Q}^{(3)} \in \Re^{q \times q}$ and define the three-way quadratic norm as the following: $|| \boldsymbol{\mathcal{X}} ||_{\mathbf{Q}^{(1)}, \mathbf{Q}^{(2)}, \mathbf{Q}^{(3)}} = \left(\sum_{i=1}^{n} \sum_{i'=1}^{n} \sum_{j=1}^{p} \sum_{j'=1}^{p} \sum_{k=1}^{q} \sum_{k'=1}^{q} \mathbf{Q}^{(1)}_{ii'} \mathbf{Q}^{(2)}_{jj'} \mathbf{Q}^{(3)}_{kk'} \boldsymbol{\mathcal{X}}_{ijk} \boldsymbol{\mathcal{X}}_{i'j'k'} \right)^{1/2} =$

$\left(\sum_{i=1}^{n}\sum_{j=1}^{p}\sum_{k=1}^{q}\tilde{\boldsymbol{\mathcal{X}}}_{ijk}\,\boldsymbol{\mathcal{X}}_{ijk}\right)^{1/2}$, where $\tilde{\boldsymbol{\mathcal{X}}} = \boldsymbol{\mathcal{X}} \times_1 \mathbf{Q}^{(1)} \times_2 \mathbf{Q}^{(2)} \times_3 \mathbf{Q}^{(3)}$. A major assumption implied by this three-way quadratic norm is separability of the structured errors along each of the tensor modes. In fact, one can show that this norm implies a three-way Kronecker covariance structure related to the array-normal distribution (Hoff, 2011). Give this, we define the rank-one Generalized CP decomposition as the solution to the following optimization problem:

$$\begin{aligned}
&\underset{\mathbf{u},\mathbf{v},\mathbf{w},d}{\text{minimize}} \;\; \frac{1}{2}||\,\boldsymbol{\mathcal{X}} - d\,\mathbf{u}\circ\mathbf{v}\circ\mathbf{w}\,||^2_{\mathbf{Q}^{(1)},\mathbf{Q}^{(2)},\mathbf{Q}^{(3)}} \\
&\text{subject to} \;\; \mathbf{u}^T\,\mathbf{Q}^{(1)}\,\mathbf{u} = 1,\;\; \mathbf{v}^T\,\mathbf{Q}^{(2)}\,\mathbf{v} = 1,\;\; \mathbf{w}^T\,\mathbf{Q}^{(3)}\,\mathbf{w} = 1,\;\; \&\; d > 0. \qquad (6)
\end{aligned}$$

In the same manner in which we motivated the Sparse CP-TPA optimization problem, we can define the rank-one Sparse Generalized CP-TPA decomposition to be the solution to the following:

$$\begin{aligned}
&\underset{\mathbf{u},\mathbf{v},\mathbf{w},}{\text{maximize}} \;\; \boldsymbol{\mathcal{X}} \times_1 \mathbf{Q}^{(1)}\,\mathbf{u} \times_2 \mathbf{Q}^{(2)}\,\mathbf{v} \times_3 \mathbf{Q}^{(3)}\,\mathbf{w} - \lambda_{\mathbf{u}}\,||\,\mathbf{u}\,||_1 - \lambda_{\mathbf{v}}\,||\,\mathbf{v}\,||_1 - \lambda_{\mathbf{w}}\,||\,\mathbf{w}\,||_1 \\
&\text{subject to} \;\; \mathbf{u}^T\,\mathbf{Q}^{(1)}\,\mathbf{u} \leq 1,\;\; \mathbf{v}^T\,\mathbf{Q}^{(2)}\,\mathbf{v} \leq 1,\;\; \&\;\; \mathbf{w}^T\,\mathbf{Q}^{(3)}\,\mathbf{w} \leq 1. \qquad (7)
\end{aligned}$$

Following from results in (Allen et al., 2011), these rank-one problems have closed form solutions.

Proposition 3. *The coordinate-wise updates for* $\mathbf{u}$ *are:*

1. *Generalized CP:* $\mathbf{u}^* = (\boldsymbol{\mathcal{X}} \times_2 \mathbf{Q}^{(2)}\,\mathbf{v} \times_3 \mathbf{Q}^{(3)}\,\mathbf{w})/||(\boldsymbol{\mathcal{X}} \times_2 \mathbf{Q}^{(2)}\,\mathbf{v} \times_3 \mathbf{Q}^{(3)}\,\mathbf{w})||_{\mathbf{Q}^{(1)}}$.
2. *Sparse Generalized CP: Define* $\hat{\mathbf{u}} = argmin_{\mathbf{u}}\{\frac{1}{2}||\,\boldsymbol{\mathcal{X}} \times_2 \mathbf{Q}^{(2)}\,\mathbf{v} \times_3 \mathbf{Q}^{(3)}\,\mathbf{w} - \mathbf{u}\,||^2_{\mathbf{Q}^{(1)}} + \lambda_{\mathbf{u}}\,||\,\mathbf{u}\,||_1\}$*, then* $\mathbf{u}^* = \hat{\mathbf{u}}/||\hat{\mathbf{u}}||_{\mathbf{Q}^{(1)}}$ *if* $||\hat{\mathbf{u}}||_{\mathbf{Q}^{(1)}} > 0$*, and* $\mathbf{u}^* = 0$ *otherwise.*

The updates for $\mathbf{v}$ *and* $\mathbf{w}$ *are analogous. Together these updates converge to a local solution to the Generalized CP or Sparse Generalized CP problem.*

As with our TPA and Sparse CP-TPA methods, subsequent components can be calculated via deflation. Thus, we have outlined a generalization of our results on HOPCA and Sparse HOPCA for use with structured tensor data.

5.3 Multi-way Functional PCA

Many examples of high-dimensional tensors have functional valued elements. Examples include multi-dimensional spectroscopy data commonly studied in chemometrics, series of images measured over time such as in neuroimaging, and hyper-spectral imaging data. For matrix data, many have used Functional PCA (FPCA) to reduced the dimension and discover patterns in functional valued data (Silverman, 1996). Recently, Huang et al. (2009) extended the Functional PCA framework to encompass two-way functional data. These same techniques can be further extended to multi-way data to study the examples of functional tensor data described above.

Huang et al. (2009) elegantly showed that estimating functional principal components by solving a bi-convex optimization problem and using deflation is equivalent to two-way half-smoothing the data, an extension of the FPCA half-smoothing result of Silverman (1996). Both the optimization approach and half-smoothing approach can be extended for multi-way tensor data. For tensors, however, these two approaches do not yield equivalent tensor factorizations. We briefly summarize these two approaches and their resulting properties. Consider functional data that has been discretized, yielding the tensor, $\boldsymbol{\mathcal{X}} \in \Re^{n \times p \times q}$. Define $\mathbf{S}_{\mathbf{u}} = \mathbf{I}_{(n)} + \boldsymbol{\Omega}_{\mathbf{u}}$, $\mathbf{S}_{\mathbf{v}} = \mathbf{I}_{(p)} + \boldsymbol{\Omega}_{\mathbf{v}}$, and $\mathbf{S}_{\mathbf{w}} = \mathbf{I}_{(q)} + \boldsymbol{\Omega}_{\mathbf{w}}$, where $\boldsymbol{\Omega}$ is a smoothing matrix commonly used in FPCA; examples include the squared second or fourth differences matrices (Ramsay, 2006). Then, we can define Tensor FPCA via half-smoothing as follows: (1) Half smooth the data, $\tilde{\boldsymbol{\mathcal{X}}} \leftarrow \boldsymbol{\mathcal{X}} \times_1 \mathbf{S}_{\mathbf{u}}{}^{-1/2} \times_2 \mathbf{S}_{\mathbf{v}}{}^{-1/2} \times_3 \mathbf{S}_{\mathbf{w}}{}^{-1/2}$; (2) Take the Tucker decomposition of $\tilde{\boldsymbol{\mathcal{X}}} \approx \tilde{\boldsymbol{\mathcal{D}}} \times_1 \tilde{\mathbf{U}} \times_2 \tilde{\mathbf{V}} \times_3 \tilde{\mathbf{W}}$; (3) and half-smooth the components, $\mathbf{u}_k \leftarrow \mathbf{S}_{\mathbf{u}}{}^{-1/2}\, \tilde{\mathbf{u}}_k$, $\mathbf{v}_k \leftarrow \mathbf{S}_{\mathbf{v}}{}^{-1/2}\, \tilde{\mathbf{v}}_k$, and $\mathbf{w}_k \leftarrow \mathbf{S}_{\mathbf{w}}{}^{-1/2}\, \tilde{\mathbf{w}}_k$. We can also define the rank-one Tensor FPCA as the solution to the following tri-convex optimization problem:

$$\underset{\mathbf{u},\mathbf{v},\mathbf{w}}{\text{minimize}} \quad || \boldsymbol{\mathcal{X}} - \mathbf{u} \circ \mathbf{v} \circ \mathbf{w} ||_F^2 + \mathbf{u}^T \mathbf{S}_{\mathbf{u}} \mathbf{u} \mathbf{v}^T \mathbf{S}_{\mathbf{v}} \mathbf{v} \mathbf{w}^T \mathbf{S}_{\mathbf{w}} \mathbf{w} - || \mathbf{u} ||_2^2 || \mathbf{v} ||_2^2 || \mathbf{w} ||_2^2. \qquad (8)$$

Subsequent components can be calculated via deflation.

Proposition 4. *1. The coordinate-wise updates converging to a local minimum of* (8) *are given by:* $\mathbf{u} = \mathbf{S}_{\mathbf{u}}{}^{-1} \left(\boldsymbol{\mathcal{X}} \times_2 \mathbf{v} \times_3 \mathbf{w} \right) / (\mathbf{v}^T \mathbf{S}_{\mathbf{v}} \mathbf{v} \mathbf{w}^T \mathbf{S}_{\mathbf{w}} \mathbf{w})$*, with the updates for* $\mathbf{v}$ *and* $\mathbf{w}$ *are defined analogously.*

2. The solution to (8) *is not equivalent to Tensor FPCA via half-smoothing. In other words, the latter is not necessarily a local minimum to* (8)*.*

Although half-smoothing and the solution to (8) are not equivalent, they both solve their respective optimization problems and thus the problem of non-convergence as with Sparse HOOI is avoided. Thus, methods for Tensor FPCA are another possible extension of our frameworks developed for Sparse HOPCA.

6 Results

We compare the performance of the various methods introduced for HOPCA and Sparse HOPCA in terms of signal recovery and feature selection on simulated data and in terms of dimension reduction and feature selection on a microarray and functional MRI example.

6.1 Simulation Studies

We evaluate the comparative performance of our methods on a simulated low rank three-way tensor model comprised of sparse rank-one factors. All data is simulated from the following model: $\boldsymbol{\mathcal{X}} = \sum_{k=1}^{K} d_k \, \mathbf{u}_k \circ \mathbf{v}_k \circ \mathbf{w}_k + \boldsymbol{\mathcal{E}}$, where the factors $\mathbf{u}_k, \mathbf{v}_k$, and $\mathbf{w}_k$ are random, d_k is fixed and $\boldsymbol{\mathcal{E}}_{i,j,l} \overset{iid}{\sim} N(0,1)$. Four scenarios are simulated and summarized as follows. Scenario 1: 100 × 100 × 100 with $\mathbf{U}$ sparse; Scenario 2: 1000 × 20 × 20 with $\mathbf{U}$ sparse; Scenario 3:

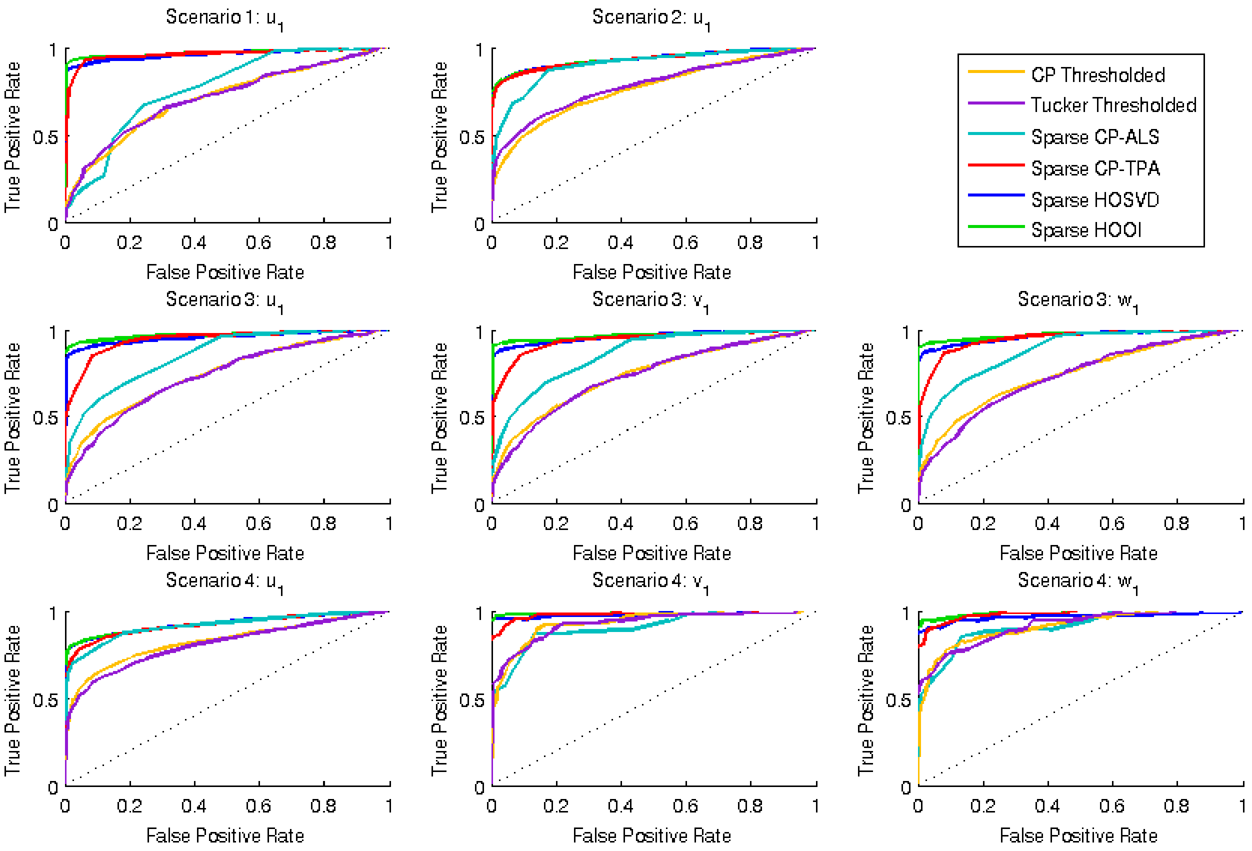

Figure 1: *ROC curves averaged over 50 replicates for each of the four simulation scenarios described in Section 6.1. The Sparse CP-TPA method performs similarly to Sparse HOSVD and Sparse HOOI in all but Scenario 3 where the later two have slight advantages. The Sparse CP-ALS method, on the other hand, exhibits poor recovery of the true features, with performance often comparable to naive thresholding.*

$100 \times 100 \times 100$ with $\mathbf{U}$, $\mathbf{V}$ and $\mathbf{W}$ sparse; and Scenario 4: $1000 \times 20 \times 20$ with $\mathbf{U}$, $\mathbf{V}$ and $\mathbf{W}$ sparse. Sparse factors are simulated with 50% randomly selected elements set to zero and non-zero values are i.i.d. $N(0,1)$. Non-sparse factors are simulated as the first K left and right singular vectors of a data matrix with i.i.d. $N(0,1)$ entries. In simulations where $K = 1$, $d_1 = 100$. In simulations where $K = 2$, $d_1 = 200$ and $d_2 = 100$. Additional simulation results for these four scenarios with various signal to noise levels and a differing percentages of sparse elements are given in the Supplemental Materials.

First, we test the accuracy of our methods in selecting the correct non-zero features. Receiver-operator curves (ROC) averaged over fifty replicates computed by varying the reg-

	Scenario 1					
	CP	Tucker	Sparse CP-ALS	Sparse CP-TPA	Sparse HOSVD	Sparse HOOI
TP / FP $\mathbf{u}_1$	-	-	0.8308 / 0.8108	0.9332 / 0.0568	0.9372 / 0.0720	0.8472 / 0.0000
TP / FP $\mathbf{u}_2$	-	-	0.8204 / 0.8304	0.8688 / 0.0324	0.8960 / 0.2060	0.8704 / 0.0304
Signal Recovery $\hat{\boldsymbol{\mathcal{X}}}$	1.0052	1.0053	0.9993	0.0504	0.0501	0.0500
	Scenario 2					
	CP	Tucker	Sparse CP-ALS	Sparse CP-TPA	Sparse HOSVD	Sparse HOOI
TP / FP $\mathbf{u}_1$	-	-	0.9310 / 0.2149	0.8874 / 0.0186	0.8902 / 0.0197	0.8346 / 0.0002
TP / FP $\mathbf{u}_2$	-	-	0.8707 / 0.2366	0.7373 / 0.0329	0.6954 / 0.0066	0.7805 / 0.0197
Signal Recovery $\hat{\boldsymbol{\mathcal{X}}}$	1.0068	1.0032	0.9981	0.1239	0.1236	0.1236
	Scenario 3					
	CP	Tucker	Sparse CP-ALS	Sparse CP-TPA	Sparse HOSVD	Sparse HOOI
TP / FP $\mathbf{u}_1$	-	-	0.9832 / 0.7068	0.9468 / 0.1620	0.9500 / 0.0776	0.9108 / 0.0012
TP / FP $\mathbf{u}_2$	-	-	0.9732 / 0.7008	0.9116 / 0.2380	0.9008 / 0.2072	0.8940 / 0.0732
TP / FP $\mathbf{v}_1$	-	-	0.9840 / 0.7000	0.9412/ 0.1696	0.9420 / 0.0784	0.9024 / 0.0012
TP / FP $\mathbf{v}_2$	-	-	0.9676 / 0.6872	0.9152 / 0.2392	0.9000 / 0.2220	0.8916 / 0.0748
TP / FP $\mathbf{w}_1$	-	-	0.9860 / 0.6868	0.9460 / 0.1684	0.9436 / 0.0744	0.9112 / 0.0000
TP / FP $\mathbf{w}_2$	-	-	0.9740 / 0.7148	0.9140 / 0.2524	0.9092 / 0.2484	0.8888 / 0.0612
Signal Recovery $\hat{\boldsymbol{\mathcal{X}}}$	1.0066	1.0060	0.9994	0.0503	0.0499	0.0496
	Scenario 4					
	CP	Tucker	Sparse CP-ALS	Sparse CP-TPA	Sparse HOSVD	Sparse HOOI
TP / FP $\mathbf{u}_1$	-	-	0.8972 / 0.1618	0.8617 / 0.0256	0.8876 / 0.0184	0.8335 / 0.0002
TP / FP $\mathbf{u}_2$	-	-	0.7987 / 0.1838	0.7986 / 0.1455	0.7083 / 0.0067	0.7860 / 0.0190
TP / FP $\mathbf{v}_1$	-	-	0.9780 / 0.5220	0.9320 / 0.0580	0.9760 / 0.1440	0.9400 / 0.0000
TP / FP $\mathbf{v}_2$	-	-	0.9400 / 0.4720	0.9080 / 0.1880	0.9700 / 0.5620	0.9520 / 0.1360
TP / FP $\mathbf{w}_1$	-	-	0.9840 / 0.4560	0.9260 / 0.0620	0.9660 / 0.1080	0.9120 / 0.0000
TP / FP $\mathbf{w}_2$	-	-	0.9360 / 0.4800	0.9000 / 0.1640	0.9660 / 0.6140	0.9380 / 0.1380
Signal Recovery $\hat{\boldsymbol{\mathcal{X}}}$	1.0090	1.0025	0.9997	0.1252	0.1238	0.1234

Table 1: *True and false positive rates (TP and FP) and signal recovery measured in mean squared error for the four simulation scenarios described in Section 6.1 averaged over 50 replicates. The Sparse CP-TPA method performs similarly to Sparse HOSVD and Sparse HOOI, with the later having a slight advantage in some scenarios. The Sparse CP-ALS method performs poorly both in terms of feature selection and signal recovery.*

ularization parameter, λ, are given for each of the four scenarios in Figure 1. We compare the Sparse HOSVD, Sparse HOOI, Sparse CP-ALS, and Sparse CP-TPA to naive thresholding of the CP and Tucker decompositions that act as a baseline. In this paper, we implement the Sparse HOSVD and Sparse HOOI using the Sparse PCA method of Shen and Huang (2008). From these comparisons, we see that the Sparse HOSVD and Sparse HOOI consistently perform the best across all four simulation scenarios. The Sparse CP-TPA method performs equally well in all but Scenario 3 where all factors are sparse and the samples sizes are equal. The Sparse CP-ALS method has considerably worse performance and barely bests the naive thresholding methods.

Next in Table 6.1, we compare the performance of our methods in terms of feature

	100 × 100 × 100	1000 × 20 × 20	250 × 250 × 250	5000 × 50 × 50
Sparse CP-ALS	0.075	0.030	1.665	3.240
Sparse CP-TPA	0.090	0.030	1.800	5.335
Sparse HOSVD	0.720	0.330	13.355	24.300
Sparse HOOI	0.835	0.430	14.230	26.950

Table 2: *Median time in seconds over ten replicates for convergence of each algorithm for a fixed value of the regularization parameter. A rank-one model in which each factor is 50% sparse is employed. The Sparse CP methods are markedly faster than the Sparse Tucker methods.*

selection and low rank signal recovery, measured in mean squared error, at one value of the regularization parameter, λ. To be consistent, the BIC method was used to select λ for all methods. Again, the Sparse HOOI is the best performing method with the Sparse CP-TPA method performing similarly in all but Scenario 3. Simulation results under different percentages of sparsity and different signal levels presented in the Supplemental Materials exhibit similar behavior.

Finally, we compare the time until convergence for each of our methods for Sparse HOPCA in Table 2. Note that while convergence cannot be mathematically guaranteed for the methods discussed in Section 3, in practice effective use of warm starts and constraints on the change in factors permitted in each iteration often yield convergent algorithms. Timings were carried out on a Intel Xeon X5680 3.33Ghz processor and methods were coded as single-thread processes run in Matlab utilizing the Tensor Toolbox (Bader and Kolda, 2010). From these timing results, we see that the Sparse CP methods are considerably faster than those based on the Tucker decomposition. While the Sparse CP-ALS method is fastest, it also has the worst performance. Overall, in addition to having nice mathematical properties, the Sparse CP-TPA method offers a good compromise between fast computation and strong performance results in terms of feature selection and signal recovery.

6.2 AGEMAP Microarray Example

We analyze the high-dimensional AGEMAP microarray data set (publicly available from `http://www.grc.nia.nih.gov/branches/rrb/dna/agemap_data.htm`) with Sparse HOPCA.

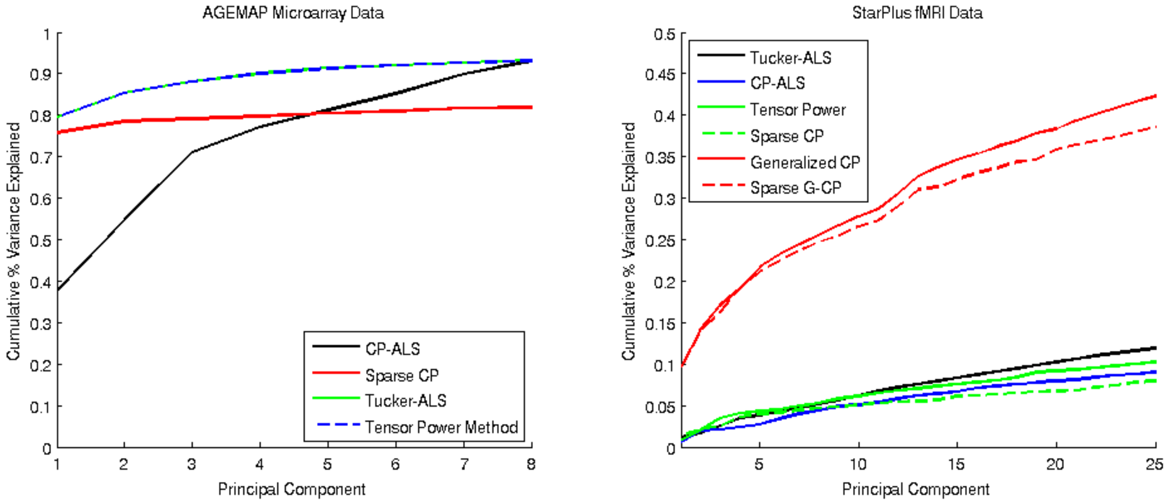

Figure 2: *Cumulative proportion of variance explained by third order PCs on the AGEMAP microarray data (left) and the StarPlus fMRI data (right).*

This data consists of gene expression measurements for 8,932 genes measured for 16 tissue types on 40 mice of ages 1, 6, 16, or 24 months (Zahn et al., 2007). As measurements for several mice are missing for various tissues, we eliminate any tensor slices that are entirely missing, yielding a data array of dimension $8932 \times 16 \times 22$. Scientists seek to discover relationships between tissue types of aging mice and the subset of genomic patterns that contribute to these relationships. These patterns in each tensor mode cannot be found by simply applying PCA or Sparse PCA to the fattened tensor.

The CP-TPA, CP-ALS, and Tucker decompositions as well as Sparse CP-TPA were applied to this data to reduce the dimension and understand patterns among tissues and genes. In the left panel of Figure 2, the cumulative proportion of variance explained by the first eight components is given. Notice that the CP-ALS, CP-TPA, and Tucker decompositions explain roughly the same proportion of variance with eight components, but the CP-TPA and Tucker decomposition explain much more initial variance in the first several PCs. As often scientists are only interested in the first couple principal components, this illustrates an important advantage of our CP-TPA and Sparse CP-TPA approach as compared to CP-ALS in the analysis of real data.

In Figure 3, we explore patterns found in the AGEMAP data via Sparse HOPCA. The results were computed using the Sparse CP-TPA method placing a penalty on the gene mode with the BIC used to select λ. In the top panel, we show scatterplots of the first six PCs for the tissue mode. We see many clusters of tissue types for the various pairs of PCs. For example, adrenal, gonads and bones often cluster together. As only a subset of genes are selected by each of the PCs, we can analyze the genetic patterns further for each tissue type. Gonads has a higher value for the second principal component, for example, so we display a cluster heatmap of the 1,439 genes selected in PC2 for this tissue type in the lower left panel of Figure 3. We see that the genes selected by this sparse PC perfectly separate the male and female mice. As liver has a lower PC value for the third tissue component, we display the cluster heatmap for the 514 genes selected by this component for liver in the lower right panel of Figure 3. Again, we see that this component clusters the mice well according to their ages. Further plots and analyses of this type reveal subsets of important genes associated with various tissues and mice ages or gender. This type of multi-mode analysis is an important advantage of applying Sparse HOPCA as opposed to Sparse PCA on a flattened tensor.

6.3 StarPlus fMRI Example

As functional MRIs are a common source of high-dimensional tensor data, we apply our methods to understand patterns for subject 04847 of the StarPlus fMRI experiment (publicly available from `http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-81/www/`) (Mitchell et al., 2004). This data set consists of 4,698 voxels or spatial locations in the brain sampled on a $64 \times 64 \times 8$ grid, measured over 54 - 55 time points for each of 40 tasks. In each task, the subject was shown an image and read a sentence. The sentence either explained the image, for example an image of a star with a plus above paired with the sentence, “The plus is above the star.”, or in which the sentence negated the image. We analyze data for each of the 36 tasks lasting for 55 time points, yielding a tensor array of dimension $4,698 \times 55 \times 36$.

We apply HOPCA, Sparse HOPCA, Generalized CP, and Sparse Generalized CP-TPA

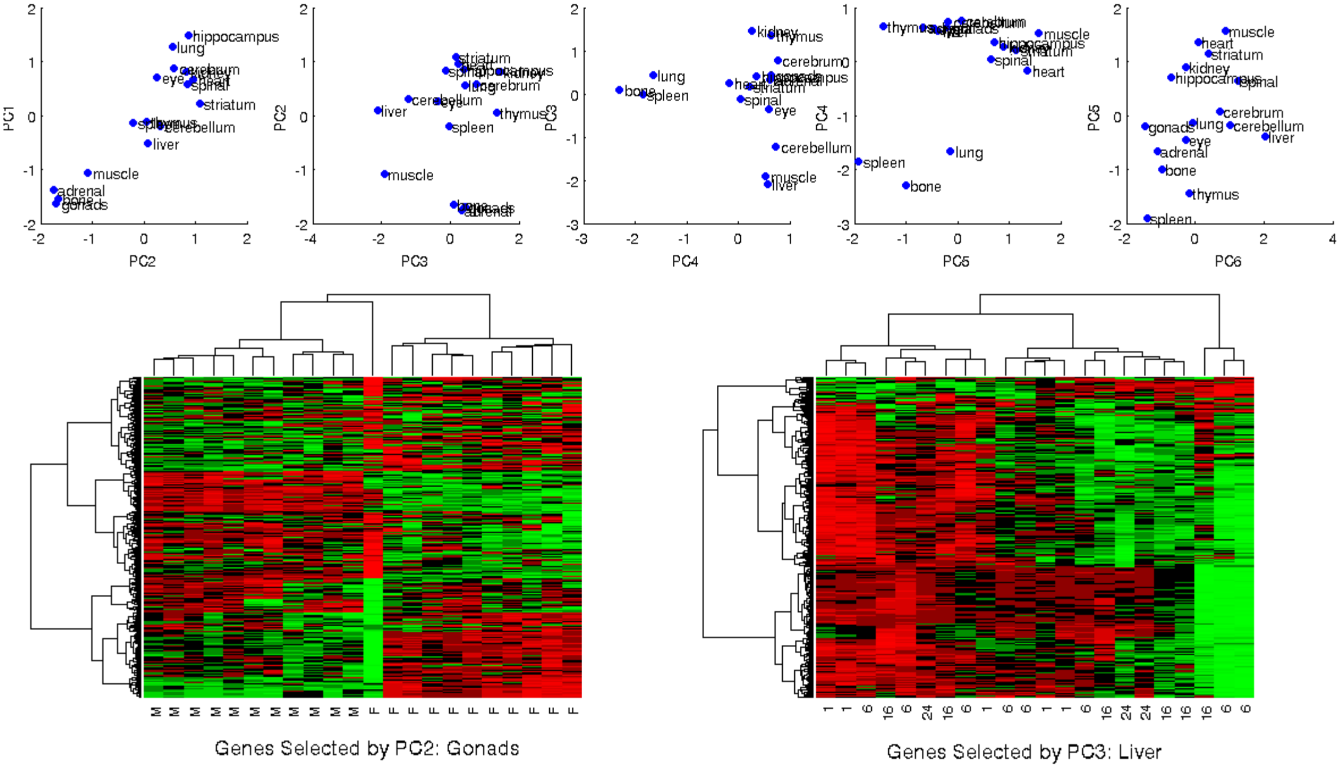

Figure 3: *Analysis of AGEMAP Microarray data via Sparse Higher-Order PCA. (Top Panel) Scatterplots of the first eight principal components for the 16 tissue types. (Lower Left) A cluster heatmap of the genes selected by sparse PC 3 for the tissue Gonads labeled by gender. (Lower Right) A cluster heatmap of the genes selected by sparse PC 8 for the tissue Heart labeled by age in months.*

methods to understand the spatial, task and temporal patterns in the fMRI data (Mørup et al., 2008). The Generalized HOPCA methods were applied with $\mathbf{Q}^{(1)}$ a graph Laplacian of the nearest neighbor graph connecting the voxels, $\mathbf{Q}^{(2)}$ a kernel smoother over the 55 time points and $\mathbf{Q}^{(3)} = \mathbf{I}$. These quadratic operators were selected such that the variance explained by the first GPC was maximal as described in Allen et al. (2011). Sparsity was incorporated into the spatial component to select relevant regions of interest. In the right panel of Figure 2, we present the cumulative proportion of variance explained by each of the methods. Generalized HOPCA methods strikingly explain a much great proportion of variance as they incorporate known spatio-temporal structure into the tensor factorization. In Figure 4, we compare the first two HOPCs of the Tucker decomposition to those of the Sparse Generalized CP-TPA method. The latter yield interpretable temporal patterns and discrete regions of interest related to the tasks. The temporal patterns of the Tucker decomposition appear to be confounded by sinusoidal noise with noisy, uninterpretable spatial components.

7 Discussion

We have developed methodology for regularizing HOPCA in the context of high-dimensional tensor data. Beginning with algorithmic approaches, we presented Sparse CP-ALS, Sparse HOSVD, and Sparse HOOI methods based on popular algorithms for the CP and Tucker decompositions. While these methods are intuitive, they fail to solve a coherent optimization problem, and are thus mathematically and computationally less desirable. Next, we develop a greedy framework for computing HOPCs based on deflation and show that a simple multi-way concave relaxation of this optimization problem leads to an approach we term Sparse CP-TPA. As this method converges to a local solution of a well-defined optimization problem, it enjoys many mathematical and computational advantages. Finally, we show how the deflation approach to HOPCA can be extended to work with general penalties and constraints, and with structured or functional tensor data. A major strength of our approach is its flexibility to model different types of high-dimensional tensor data, a quality illustrated on our examples to microarray and fMRI data.

There are several items related to our work to study further. First, we note that while we have presented all of our methods with three-mode tensors for notational convenience, extensions to tensors with an arbitrary number of modes is trivial. The extensions of our Sparse HOPCA frameworks to encompass general penalties and constraints, structured tensors, and multi-way functional data outlined in Section 5 deserve further investigation and development. Simulation studies and comparisons to existing approaches are needed to fully evaluate the utility of these methods.

Two important items related to our regularized HOPCA frameworks are beyond the scope of this paper and require further investigation. Determining how many HOPCs to

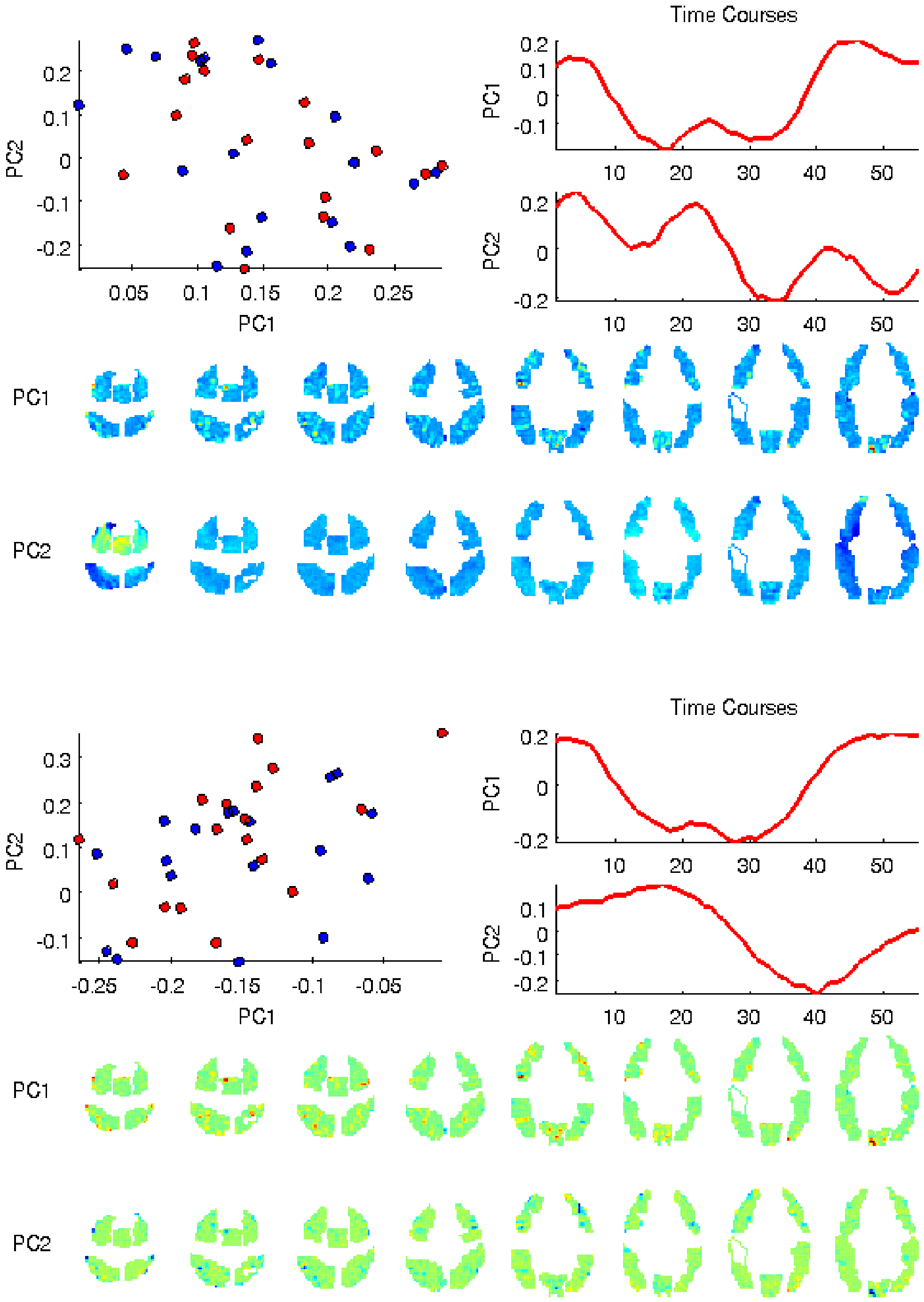

Figure 4: *Depictions of the first two task PCs, time course PCs and spatial PCs for the Tucker decomposition (top panel) and Sparse Generalized CP-TPA (bottom panel) for the StarPlus fMRI data. In the scatterplots, red denotes the sentence negating the image while blue denotes sentence and image agreement.*

extract for a given data set is an important problem. For PCA on matrix data, there are several data-driven methods available (Buja and Eyuboglu, 1992; Owen and Perry, 2009), but extending these approaches to the tensor domain is a non-trivial task. Our result on the amount of variance explained by each of the HOPCs or Sparse HOPCs, Theorem 1, gives users of our methodology a way to gage the dimension reduction achieved, but further work needs to be done to determine the optimal number of factors. Secondly, results for every regularized HOPCA method, and in fact all methods for HOPCA, depend heavily on initial algorithmic starting values (Kroonenberg, 2008). This occurs as all HOPCA and regularized HOPCA methods find at best a local optimum. With Sparse PCA methods on matrix data, the same problem occurs but is less critical as one can initialize algorithms to the SVD which is a global solution. With tensor factorizations, further work is required to determine the best initializations for our regularized HOPCA algorithms.

As the statistical attention given to high-dimensional tensor data has been limited, topics for further study abound. Recently, many have made progress on understanding the asymptotic properties of PCA and Sparse PCA by using random matrix theory (Johnstone and Lu, 2009; Jung and Marron, 2009; Amini and Wainwright, 2009). As yet, there is no work on developing consistency results for HOPCA which would serve as a precursor to studying consistency for the methods presented in this paper. Additionally, our framework for regularized HOPCA is a foundation upon which to study other multivariate analysis techniques for tensor data. These include methods for clustering, canonical correlations analysis, partial least squares, discriminant analysis, and multi-dimensional scaling. Variable selection in the context of regression and classification for high-dimensional matrix data has been a topic of great interest recently. One can imagine that variable selection for high-dimensional tensors may be more critical in these supervised prediction methods.

Finally, there are a plethora of possible applications of our methodology. Technologies in biomedical imaging are producing massive multi-way data sets that are a challenge to understand and analyze. Examples of these are especially common in neuroimaging, radiology, and chemometrics. Other imaging data sets such as from hyper-spectral cameras and remote sensing often contain three or more measured dimensions. Additional examples of high-dimensional tensor data come from environmental and climate studies, bibliometrics, and network reliability. Overall, our methods for regularized HOPCA are foundational tools for understanding high-dimensional tensors that will enjoy wide-ranging applicability.

8 Acknowledgments

This paper is based, in part, on a previous conference paper Allen (2012). The author would like to thank Eric Chi for helpful discussions on tensors. Supplemental materials contain proofs of all theoretical results, algorithm outlines for all methods, and additional simulation results. Software based on the `Matlab Tensor Toolbox` (Bader and Kolda, 2010) will be made available from `http://www.stat.rice.edu/ gallen/software.html`.

A Algorithms for HOPCA & Sparse HOPCA

Algorithm 1 Sparse Higher-Order SVD (Sparse HOSVD)

```
1. $\mathbf{U} \leftarrow$ First $K_1$ sparse principal components of $\mathbf{X}_{(1)}$.
2. $\mathbf{V} \leftarrow$ First $K_2$ sparse principal components of $\mathbf{X}_{(2)}$.
3. $\mathbf{W} \leftarrow$ First $K_3$ sparse principal components of $\mathbf{X}_{(3)}$.
4. $\boldsymbol{\mathcal{D}} \leftarrow \boldsymbol{\mathcal{X}} \times_1 \mathbf{U} \times_2 \mathbf{V} \times_3 \mathbf{W}$.
```

B Additional Simulation Results

C Proofs

Proof of Theorem 1. The proof is an extension of a result in Shen and Huang (2008). Recall that the cumulative proportion of variance explained by the first k traditional principal

Algorithm 2 Sparse Higher-Order Orthogonal Iteration (Sparse HOOI), or Sparse Tucker-ALS

```
1. Initialize $\mathbf{U}$, $\mathbf{V}$, and $\mathbf{W}$ using the Sparse HOSVD algorithm.
2. Repeat until convergence or maximum number of iterations reached:
    (a) $\mathbf{U} \leftarrow$ First $K_1$ sparse principal components of $(\boldsymbol{\mathcal{X}} \times_2 \mathbf{V} \times_3 \mathbf{W})_{(1)}$.
    (b) $\mathbf{V} \leftarrow$ First $K_2$ sparse principal components of $(\boldsymbol{\mathcal{X}} \times_1 \mathbf{U} \times_3 \mathbf{W})_{(2)}$.
    (c) $\mathbf{W} \leftarrow$ First $K_3$ sparse principal components of $(\boldsymbol{\mathcal{X}} \times_1 \mathbf{U} \times_2 \mathbf{V})_{(3)}$.
3. $\boldsymbol{\mathcal{D}} \leftarrow \boldsymbol{\mathcal{X}} \times_1 \mathbf{U} \times_2 \mathbf{V} \times_3 \mathbf{W}$
```

Algorithm 3 Sparse CP-ALS Algorithm

```
1. Initialize $\mathbf{U}^{(0)}$, $\mathbf{V}^{(0)}$ and $\mathbf{W}^{(0)}$.
2. Repeat until convergence or maximum number of iterations reached:
    (a) Estimate $\mathbf{U}^{(t+1)}$:
        i. $\hat{\mathbf{U}} \leftarrow \text{argmin}_{\mathbf{U}} \left\{ \frac{1}{2} || \, \mathbf{X}_{(1)} - \mathbf{U}(\mathbf{V}^{(t)} \odot \mathbf{W}^{(t)})^T ||_F^2 + \lambda_{\mathbf{U}} || \, \mathbf{U} \, ||_1 \right\}$.
        ii. $\hat{d}_k \leftarrow ||\hat{\mathbf{u}}_k||$.
        iii. $\mathbf{u}_k^{(t+1)} \leftarrow \hat{\mathbf{u}}_k / \hat{d}_k$.
    (b) Estimate $\mathbf{V}^{(t+1)}$:
        i. $\hat{\mathbf{V}} \leftarrow \text{argmin}_{\mathbf{V}} \left\{ \frac{1}{2} || \, \mathbf{X}_{(2)} - \mathbf{V}(\mathbf{U}^{(t+1)} \odot \mathbf{W}^{(t)})^T ||_F^2 + \lambda_{\mathbf{V}} || \, \mathbf{V} \, ||_1 \right\}$.
        ii. $\hat{d}_k \leftarrow ||\hat{\mathbf{v}}_k||$.
        iii. $\mathbf{v}_k^{(t+1)} \leftarrow \hat{\mathbf{v}}_k / \hat{d}_k$.
    (c) Estimate $\mathbf{W}^{(t+1)}$:
        i. $\hat{\mathbf{W}} \leftarrow \text{argmin}_{\mathbf{W}} \left\{ \frac{1}{2} || \, \mathbf{X}_{(3)} - \mathbf{W}(\mathbf{U}^{(t+1)} \odot \mathbf{V}^{(t+1)})^T ||_F^2 + \lambda_{\mathbf{W}} || \, \mathbf{W} \, ||_1 \right\}$.
        ii. $\hat{d}_k \leftarrow ||\hat{\mathbf{w}}_k||$.
        iii. $\mathbf{w}_k^{(t+1)} \leftarrow \hat{\mathbf{w}}_k / \hat{d}_k$.
```

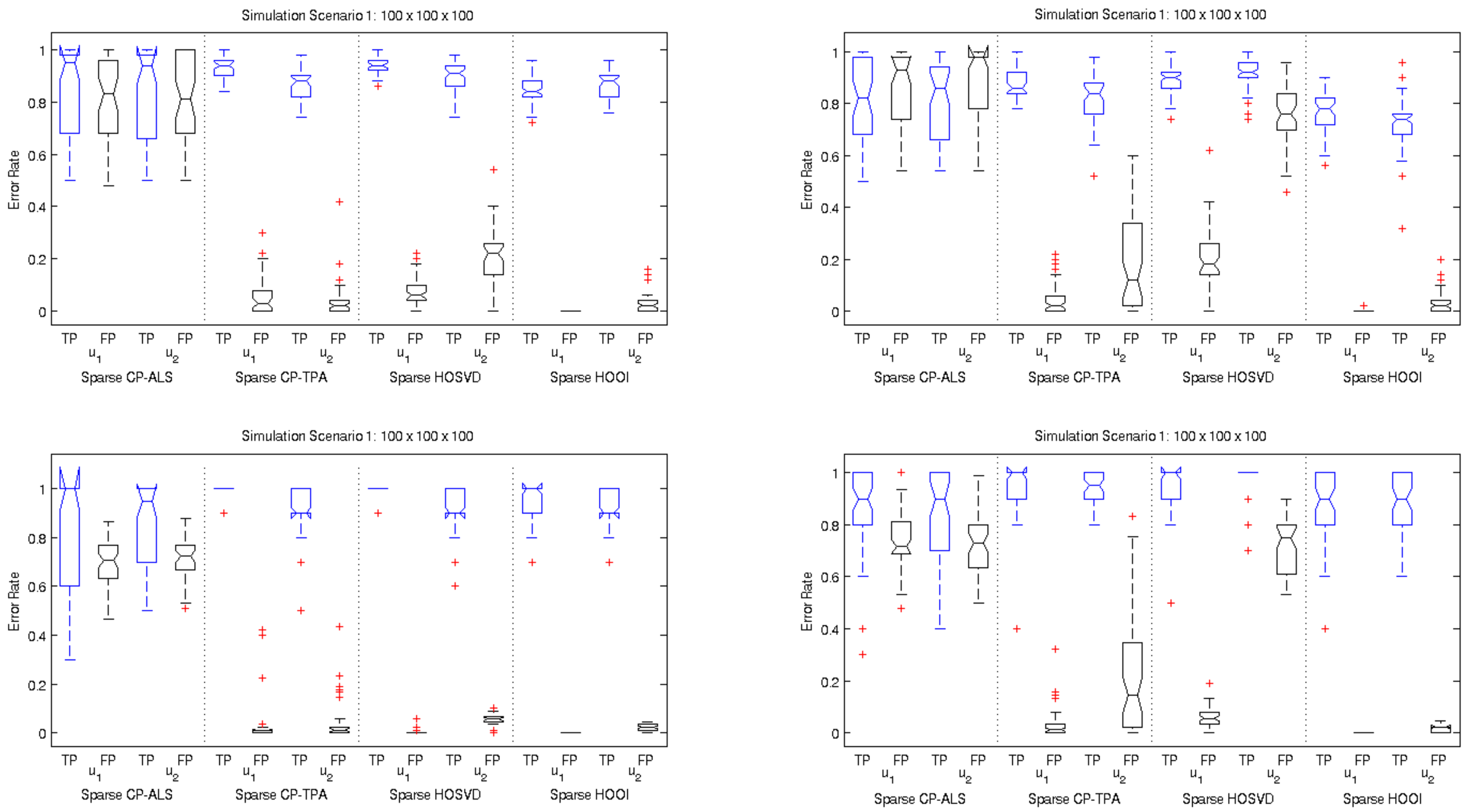

Figure 5: *Boxplots of true and false positives for simulation scenario 1, $K = 2$, for factors with 50% sparsity (upper panels) or 90% sparsity (lower panels) and with high signal (left panels), $D = [200 \quad 100]^T$ or lower signal (right panels), $D = [100 \quad 50]^T$.*

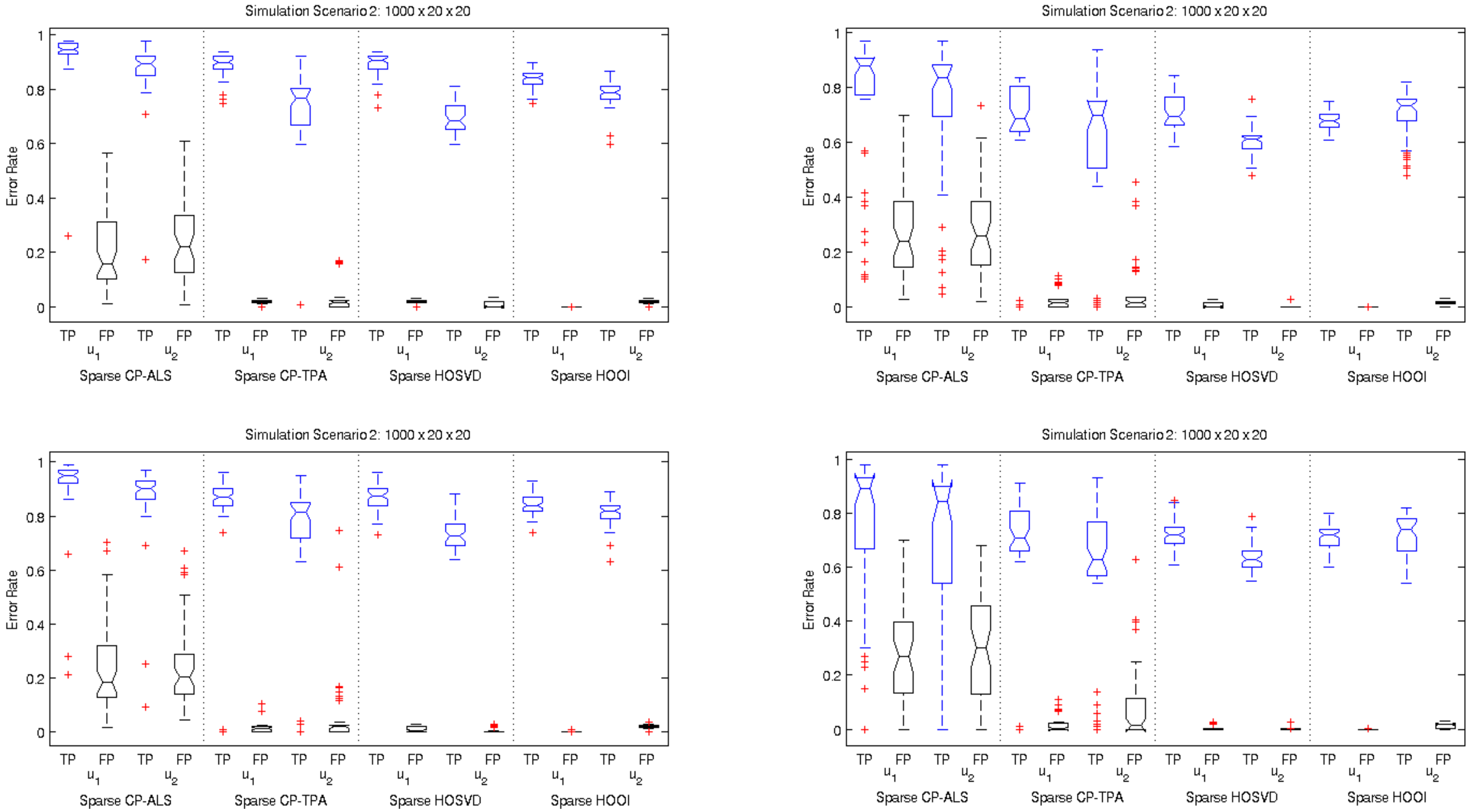

Figure 6: *Boxplots of true and false positives for simulation scenario 2, $K = 2$, for factors with 50% sparsity (upper panels) or 90% sparsity (lower panels) and with high signal (left panels), $D = [200 \ \ 100]^T$ or lower signal (right panels), $D = [100 \ \ 50]^T$.*

Algorithm 4 Tensor Power Algorithm

```
1. Initialize $\hat{\boldsymbol{\mathcal{X}}} = \boldsymbol{\mathcal{X}}$.
2. For $k = 1 \dots K$
    (a) Repeat until converge:
        i. $\mathbf{u}_k \leftarrow \hat{\boldsymbol{\mathcal{X}}} \times_2 \mathbf{v}_k \times_3 \mathbf{w}_k \;\; / \; ||\hat{\boldsymbol{\mathcal{X}}} \times_2 \mathbf{v}_k \times_3 \mathbf{w}_k \, ||_2$.
        ii. $\mathbf{v}_k \leftarrow \hat{\boldsymbol{\mathcal{X}}} \times_1 \mathbf{u}_k \times_3 \mathbf{w}_k \;\; / \; ||\hat{\boldsymbol{\mathcal{X}}} \times_1 \mathbf{u}_k \times_3 \mathbf{w}_k \, ||_2$.
        iii. $\mathbf{w}_k \leftarrow \hat{\boldsymbol{\mathcal{X}}} \times_1 \mathbf{u}_k \times_2 \mathbf{v}_k \;\; / \; ||\hat{\boldsymbol{\mathcal{X}}} \times_1 \mathbf{u}_k \times_2 \mathbf{v}_k \, ||_2$.
    (b) $d_k \leftarrow \hat{\boldsymbol{\mathcal{X}}} \times_1 \mathbf{u}_k \times_2 \mathbf{v}_k \times_3 \mathbf{w}_k$.
    (c) $\hat{\boldsymbol{\mathcal{X}}} \leftarrow \hat{\boldsymbol{\mathcal{X}}} - d_k \, \mathbf{u}_k \circ \mathbf{v}_k \circ \mathbf{w}_k$.
```

Algorithm 5 Sparse CP-TPA

```
1. Initialize $\hat{\boldsymbol{\mathcal{X}}} = \boldsymbol{\mathcal{X}}$.
2. For $k = 1 \dots K$
    (a) Repeat until converge:
        i. $\hat{\mathbf{u}}_k = S\left(\hat{\boldsymbol{\mathcal{X}}} \times_2 \mathbf{v}_k \times_3 \mathbf{w}_k, \lambda_{\mathbf{u}}\right)$.   $\mathbf{u}_k \leftarrow \begin{cases} \hat{\mathbf{u}}_k/||\hat{\mathbf{u}}_k||_2 & ||\hat{\mathbf{u}}_k||_2 > 0 \\ 0 & \text{otherwise.} \end{cases}$
        ii. $\hat{\mathbf{v}}_k = S\left(\hat{\boldsymbol{\mathcal{X}}} \times_1 \mathbf{u}_k \times_3 \mathbf{w}_k, \lambda_{\mathbf{v}}\right)$.   $\mathbf{v}_k \leftarrow \begin{cases} \hat{\mathbf{v}}_k/||\hat{\mathbf{v}}_k||_2 & ||\hat{\mathbf{v}}_k||_2 > 0 \\ 0 & \text{otherwise.} \end{cases}$
        iii. $\hat{\mathbf{w}}_k = S\left(\hat{\boldsymbol{\mathcal{X}}} \times_1 \mathbf{u}_k \times_2 \mathbf{v}_k, \lambda_{\mathbf{w}}\right)$.   $\mathbf{w}_k \leftarrow \begin{cases} \hat{\mathbf{w}}_k/||\hat{\mathbf{w}}_k||_2 & ||\hat{\mathbf{w}}_k||_2 > 0 \\ 0 & \text{otherwise.} \end{cases}$
    (b) $d_k \leftarrow \hat{\boldsymbol{\mathcal{X}}} \times_1 \mathbf{u}_k \times_2 \mathbf{v}_k \times_3 \mathbf{w}_k$.
    (c) $\hat{\boldsymbol{\mathcal{X}}} \leftarrow \hat{\boldsymbol{\mathcal{X}}} - d_k \, \mathbf{u}_k \circ \mathbf{v}_k \circ \mathbf{w}_k$.
```

components can be written as the ratio of the squared Frobenius norm of the data projected onto the first k left and right singular vectors to the squared Frobenius norm of the data (Jolliffe and MyiLibrary, 2002). Notice that $\mathbf{P}_k^{(U)}$, $\mathbf{P}_k^{(V)}$ and $\mathbf{P}_k^{(W)}$ are projection matrices with exactly k eigenvalues equal to one. Therefore, the result is an extension of the definition of cumulative proportion of variance explained to the tensor framework. Namely, the numerator is simply the projection of the tensor onto the subspace spanned by the first k HOPCA factors. □

Proof of Proposition 1. We will show that the updates for $\mathbf{U}$ in the Sparse CP-ALS algorithm are not equivalent to the optimal point for $\mathbf{U}$ from (1). Arguments for $\mathbf{V}$ and $\mathbf{W}$ are analogous. First, note that the problem (1) with respect to $\mathbf{U}$ can be rewritten using the Khatri-Rao product as minimize$_{\mathbf{U}}$ $\frac{1}{2}||\,\mathbf{X}_{(1)} - \mathbf{U}(\mathbf{V} \odot \mathbf{W})^T||_F^2 + \lambda_{\mathbf{u}}\,||\,\mathbf{U}\,||_1$ subject to $\mathbf{u}_k^T\,\mathbf{u}_k \leq 1 \;\forall\; k = 1, \dots K$. To simplify notation, we will write $\mathbf{H} = \mathbf{V} \odot \mathbf{W}$ and $\mathbf{Y} = \mathbf{X}_{(1)}$. As this problem is convex in $\mathbf{U}$, the KKT conditions are necessary and sufficient for optimality. These conditions include the sub-gradient equation:

$$\mathbf{Y}\,\mathbf{H}^T - \mathbf{U}^*\,\mathbf{H}\,\mathbf{H}^T + \lambda_{\mathbf{u}}\,\Gamma(\mathbf{U}^*) - 2\Psi^*\,\mathbf{U}^* = 0, \tag{9}$$

where $\Gamma()$ is the sub-gradient of the ℓ_1-norm and Ψ^* is the Lagrange multiplier, namely a diagonal matrix that from complimentary slackness is strictly non-zero, $\Psi^* \succ 0$ if and only if the columns of $\mathbf{U}^*$ have norm one.

Now consider the sub-gradient equations implied by the Sparse CP-ALS updates for $\mathbf{U}$ before scaling the solution $\hat{\mathbf{U}}$ to have column norm one:

$$\mathbf{Y}\,\mathbf{H}^T - \hat{\mathbf{U}}\,\mathbf{H}\,\mathbf{H}^T + \lambda_{\mathbf{u}}\,\Gamma(\hat{\mathbf{U}}) = 0.$$

We can represent scaling the columns of $\hat{\mathbf{U}}$ as right multiplying by a diagonal matrix: $\hat{\mathbf{U}}\,\mathbf{D}$. Since $\Gamma()$ is order one, then $\Gamma(\hat{\mathbf{U}}\,\mathbf{D}) = \Gamma(\hat{\mathbf{U}})$ and the above sub-gradient equation is equal to:

$$\mathbf{Y}\,\mathbf{H}^T - (\hat{\mathbf{U}}\,\mathbf{D})\,\mathbf{D}^{-1}\,\mathbf{H}\,\mathbf{H}^T + \lambda_{\mathbf{u}}\,\Gamma(\hat{\mathbf{U}}\,\mathbf{D}) = 0, \mathbf{Y}\,\mathbf{H}^T - (\tilde{\mathbf{U}})\,\mathbf{D}^{-1}\,\mathbf{H}\,\mathbf{H}^T + \lambda_{\mathbf{u}}\,\Gamma(\tilde{\mathbf{U}}) \quad = 0,$$

where $\tilde{\mathbf{U}} = \hat{\mathbf{U}}\,\mathbf{D}$. Now, for the Sparse CP-ALS update to solve (1), the above sub-gradient equation must be equivalent to (9) for some diagonal $\Psi^* \succ 0$ and some diagonal $\mathbf{D} \succ 0$. This means that $\mathbf{U}^*(\mathbf{H}\,\mathbf{H}^T + 2\Psi^*) = \tilde{\mathbf{U}}\,\mathbf{D}^{-1}\,\mathbf{H}\,\mathbf{H}^T$. Assuming that $\tilde{\mathbf{U}} = \mathbf{U}^*$ and solving for Ψ^*, we have that $\Psi^* = \mathbf{H}\,\mathbf{H}^T(\mathbf{D}^{-1} - \mathbf{I})/2$. Thus, Ψ^* is diagonal if and only if $\mathbf{H}\,\mathbf{H}^T$ is diagonal. This is a contradiction. Therefore, the updates from the Sparse CP-ALS algorithm do not solve (1). □

Proof of Proposition 2. Consider optimizing (3) with respect to $\mathbf{u}$. The Lagrangian is given by $L(\mathbf{u}, \gamma) = (\boldsymbol{\mathcal{X}} \times_2 \mathbf{v} \times_3 \mathbf{w}) \times_1 \mathbf{u} - \gamma(\mathbf{u}^T \mathbf{u} - 1)$. The KKT conditions imply that $\mathbf{u}^* = \frac{\boldsymbol{\mathcal{X}} \times_2 \mathbf{v} \times_3 \mathbf{w}}{2\gamma^*}$ and γ^* is such that $(\mathbf{u}^*)^T \mathbf{u}^* = 1$. Putting these together we have the desired result for $\mathbf{u}$. The arguments for $\mathbf{v}$ and $\mathbf{w}$ are analogous. □

Proof of Theorem 2. The proof follows from an extension of results in Witten et al. (2009) and Allen et al. (2011). In short, consider optimizing (4) with respect to $\mathbf{u}$. The KKT conditions imply that $\boldsymbol{\mathcal{X}} \times_2 \mathbf{v} \times_3 \mathbf{w} - \rho_{\mathbf{u}} \boldsymbol{\Gamma}(\mathbf{u}^*) - 2\gamma^* \mathbf{u}^* = 0$ and $\gamma^*((\mathbf{u}^*)^T \mathbf{u}^* - 1) = 0$ where $\boldsymbol{\Gamma}(\mathbf{u})$ is the subgradient of $||\, \mathbf{u} \,||_1$, and γ is a Lagrange multiplier. Consider $\hat{\mathbf{u}} = S(\boldsymbol{\mathcal{X}} \times_2 \mathbf{v} \times_3 \mathbf{w}, \rho_{\mathbf{u}})$. Then, taking $\mathbf{u}^* = \hat{\mathbf{u}}/||\hat{\mathbf{u}}||_2$ and $\gamma^* = ||\hat{\mathbf{u}}||_2/2$ simultaneously satisfies the KKT conditions. Since the problem is convex in $\mathbf{u}$, the conditions are necessary and sufficient; hence, the pair $(\mathbf{u}^*, \gamma^*)$ are the optimal points. It is easy to verify that the pair $(0, 0)$ also satisfy the KKT conditions and are optimal points. □

Proof of Theorem 3. The proof follows from an extension of results in Allen et al. (2011). Let us briefly consider the case for $\mathbf{u}$ and the arguments for $\mathbf{v}$ and $\mathbf{w}$ are analogous. We will show that the optimal points, $(\mathbf{u}^*, \mathbf{v}^*)$, implied by (5) are equivalent to the said solution. The subgradient equation for (5) is: $\boldsymbol{\mathcal{X}} \times_2 \mathbf{v} \times_3 \mathbf{w} - 2\gamma^* \mathbf{u}^* - \lambda_{\mathbf{u}} \nabla P_{\mathbf{u}}(\mathbf{u}^*) = 0$, where $\nabla P_{\mathbf{u}}()$ is the subgradient of $P_{\mathbf{u}}()$. Now, consider the subgradient equation of minimize$_{\mathbf{u}}$ $\frac{1}{2}||\, \boldsymbol{\mathcal{X}} \times_2 \mathbf{v} \times_3 \mathbf{w} - \mathbf{u} \,||_2^2 + \lambda_{\mathbf{u}} P_{\mathbf{u}}(\mathbf{u})$ and for some $c > 0$:

$$0 = \boldsymbol{\mathcal{X}} \times_2 \times_3 \mathbf{w} - \hat{\mathbf{u}} - \lambda_{\mathbf{u}} \nabla P_{\mathbf{u}}(\mathbf{u}) = \boldsymbol{\mathcal{X}} \times_2 \times_3 \mathbf{w} - \frac{1}{c}(c\hat{\mathbf{u}}) - \lambda_{\mathbf{u}} \nabla P_{\mathbf{u}}(\mathbf{u}) = \boldsymbol{\mathcal{X}} \times_2 \times_3 \mathbf{w} - \tilde{\mathbf{u}}/c - \lambda_{\mathbf{u}} \nabla P_{\mathbf{u}}(\tilde{\mathbf{u}}),$$

where $\tilde{\mathbf{u}} = c\hat{\mathbf{u}}$. Since $P_{\mathbf{u}}()$ is order one, $\nabla P_{\mathbf{u}}(x) = \nabla P_{\mathbf{u}}(cx)\ \forall c > 0$. Taking $c = 1/||\hat{\mathbf{u}}||_2$ and letting $\gamma^* = \frac{1}{2c} = ||\hat{\mathbf{u}}||_2/2$, we see that for the pair $(\mathbf{u}^* = \hat{\mathbf{u}}/||\hat{\mathbf{u}}||_2, \gamma^* = ||\hat{\mathbf{u}}||_2/2)$ the subgradient equation of the penalized regression problem is equivalent to that of (5). Following from complimentary slackness, $\gamma^* = 0$ if and only if $\hat{\mathbf{u}} \equiv 0$, and hence the pair $(0, 0)$ also satisfy the KKT conditions of (5). We have thus proven the desired result. □

Proof of Proposition 3. The proof for part one follows from an extension of Proposition

2 to the generalized eigenvalue case as shown in (Allen et al., 2011). In brief, the gradient equation of (6) is $\boldsymbol{\mathcal{X}} \times_1 \mathbf{Q}^{(1)} \times_2 \mathbf{Q}^{(2)} \mathbf{v} \times_3 \mathbf{Q}^{(3)} \mathbf{w} - 2\gamma \mathbf{Q}^{(1)} \mathbf{u} = 0$ meaning that $\mathbf{u}^* = \boldsymbol{\mathcal{X}} \times_2 \mathbf{Q}^{(2)} \mathbf{v} \times_3 \mathbf{Q}^{(3)} \mathbf{w} / 2\gamma^*$ where γ^* is such that $\mathbf{u}^T \mathbf{Q}^{(1)} \mathbf{u}^* = 1$. Putting these together we have the desired result.

The argument for part two follows from an extension of Theorem 3 and Allen et al. (2011). For completeness, we show that the subgradient of (7) with respect to $\mathbf{u}$, $\boldsymbol{\mathcal{X}} \times_1 \mathbf{Q}^{(1)} \times_2 \mathbf{Q}^{(2)} \mathbf{v} \times_3 \mathbf{Q}^{(3)} \mathbf{w} - 2\gamma \mathbf{Q}^{(1)} \mathbf{u} - \lambda_{\mathbf{u}} \Gamma(\mathbf{u}) = 0$ where $\Gamma(\cdot)$ is the sub-gradient of $|| \cdot ||_1$ is equivalent to the Sparse Generalized CP update given in Proposition 3. Let $\mathbf{y} = \boldsymbol{\mathcal{X}} \times_2 \mathbf{Q}^{(2)} \mathbf{v} \times_3 \mathbf{Q}^{(3)} \mathbf{w}$ and let $\hat{\mathbf{u}}$ be the argument minimizing $\frac{1}{2} || \mathbf{y} - \mathbf{u} ||^2_{\mathbf{Q}^{(1)}} + \lambda_{\mathbf{u}} || \mathbf{u} ||_1$. Then, for some $c > 0$, $0 = \mathbf{Q}^{(1)} \mathbf{y} - \mathbf{Q}^{(1)} \hat{\mathbf{u}} - \lambda_{\mathbf{u}} \Gamma(\hat{\mathbf{u}}) = \mathbf{Q}^{(1)} \mathbf{y} - \mathbf{Q}^{(1)} \tilde{\mathbf{u}}/c - \lambda_{\mathbf{u}} \Gamma(\tilde{\mathbf{u}})$, where $\tilde{\mathbf{u}} = c\hat{\mathbf{u}}$, since $\Gamma(\hat{\mathbf{u}}) = \Gamma(c\hat{\mathbf{u}})$ $\forall c > 0$. Taking $\gamma^* = 1/2c = ||\hat{\mathbf{u}}||_{\mathbf{Q}^{(1)}}/2$ we see that the for both pairs $(\mathbf{u}^* = \hat{\mathbf{u}}/||\hat{\mathbf{u}}||^{(1)}_{\mathbf{Q}}, \gamma^* = ||\hat{\mathbf{u}}||_{\mathbf{Q}^{(1)}}/2)$ and $(\mathbf{u}^* = 0, \gamma^* = 0)$ satisfy the KKT conditions of (7), thus proving the desired result. □

Proof of Proposition 4. For part 1, consider the gradient equation of (8) with respect to $\mathbf{u}$: $\frac{\partial}{\partial \mathbf{u}} = -2 \boldsymbol{\mathcal{X}} \times_2 \mathbf{v} \times_3 \mathbf{w} + 2 \mathbf{u} \mathbf{v}^T \mathbf{v} \mathbf{w}^T \mathbf{w} - 2 \mathbf{u} || \mathbf{v} ||_2^2 || \mathbf{w} ||_2^2 + 2 \mathbf{S}_{\mathbf{u}} \mathbf{u} \mathbf{v}^T \mathbf{S}_{\mathbf{v}} \mathbf{v} \mathbf{w}^T \mathbf{S}_{\mathbf{w}} \mathbf{w} = 0$. It is then easy to see that the coordinate update is $\mathbf{u}^* = \mathbf{S}_{\mathbf{u}}^{-1} \boldsymbol{\mathcal{X}} \times_2 \mathbf{v} \times_3 \mathbf{w} / (\mathbf{v}^T \mathbf{S}_{\mathbf{v}} \mathbf{v} \mathbf{w}^T \mathbf{S}_{\mathbf{w}} \mathbf{w})$.

For part 2, let us review the two-way matrix case proved in Huang et al. (2009). They show that two-way FPCA via half-smoothing implies that $\mathbf{u} \propto \mathbf{S}_{\mathbf{u}}^{-1} \mathbf{X} \mathbf{v}$ and $\mathbf{v} \propto \mathbf{S}_{\mathbf{v}}^{-1} \mathbf{X}^T \mathbf{u}$ which are the stationary points of the two gradient equations for the penalized optimization problem. For our tensor case, the gradient equations of (8) imply $\mathbf{u} \propto \mathbf{S}_{\mathbf{u}}^{-1} \boldsymbol{\mathcal{X}} \times_2 \mathbf{v} \times_3 \mathbf{w}$ and so forth, but these are not satisfied by tensor half-smoothing. As the Tucker decomposition has a non-diagonal core, $\mathbf{u}_1$ is proportional to the first column of $\mathbf{S}_{\mathbf{u}}^{-1} (\boldsymbol{\mathcal{X}} \times_2 \mathbf{V} \times_3 \mathbf{W}) \boldsymbol{\mathcal{D}}$. Thus, the stationary points implied by half-smoothing and (8) are not equivalent. □

References

Allen, G. I. (2012). Sparse higher-order principal components analysis. In *AISTATS*, Volume 15.

Allen, G. I., L. Grosenick, and J. Taylor (2011). A generalized least squares matrix decomposition. Rice University Technical Report No. TR2011-03.

Allen, G. I. and M. Maletić-Savatić (2011). Sparse non-negative generalized pca with applications to metabolomics. *Bioinformatics 27*(21), 3029–3035.

Amini, A. and M. Wainwright (2009). High-dimensional analysis of semidefinite relaxations for sparse principal components. *The Annals of Statistics 37*(5B), 2877–2921.

Bader, B. and T. Kolda (2010). *Matlab tensor toolbox version 2.4.*

Buja, A. and N. Eyuboglu (1992). Remarks on parallel analysis. *Multivariate Behavioral Research 27*(4), 509–540.

Carroll, J. and J. Chang (1970). Analysis of individual differences in multidimensional scaling via an n-way generalization of eckart-young decomposition. *Psychometrika 35*(3), 283–319.

Chi, E. and T. Kolda (2011). On tensors, sparsity, and nonnegative factorizations. *Arxiv preprint arXiv:1112.2414*.

Cichocki, A., R. Zdunek, A. Phan, and S. Amari (2009). *Nonnegative matrix and tensor factorizations: applications to exploratory multi-way data analysis and blind source separation.* Wiley.

De Lathauwer, L., B. De Moor, and J. Vandewalle (2000). A multilinear singular value decomposition. *SIAM Journal on Matrix Analysis and Applications 21*(4), 1253–1278.

Golub, G. H. and C. F. Van Loan (1996). *Matrix Computations (Johns Hopkins Studies in Mathematical Sciences)(3rd Edition)* (3rd ed.). The Johns Hopkins University Press.

Harshman, R. (1970). Foundations of the parafac procedure: Models and conditions for an" explanatory" multimodal factor analysis. UCLA Working Papers in Phonetics.

Hazan, T., S. Polak, and A. Shashua (2005). Sparse image coding using a 3d non-negative tensor factorization. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, Volume 1, pp. 50–57. IEEE.

Hoff, P. (2011). Separable covariance arrays via the tucker product, with applications to multivariate relational data. *Bayesian Analysis 6*(2), 179–196.

Huang, J., H. Shen, and A. Buja (2008). Functional principal components analysis via penalized rank one approximation. *Electronic Journal of Statistics 2*, 678–695.

Huang, J., H. Shen, and A. Buja (2009). The analysis of two-way functional data using two-way regularized singular value decompositions. *Journal of the American Statistical Association 104*(488), 1609–1620.

Johnstone, I. and A. Lu (2009). On consistency and sparsity for principal components analysis in high dimensions. *Journal of the American Statistical Association 104*(486), 682–693.

Jolliffe, I. and MyiLibrary (2002). *Principal component analysis*, Volume 2. Wiley Online Library.

Jolliffe, I., N. Trendafilov, and M. Uddin (2003). A modified principal component technique based on the LASSO. *Journal of Computational and Graphical Statistics 12*(3), 531–547.

Journée, M., Y. Nesterov, P. Richtárik, and R. Sepulchre (2010). Generalized power method for sparse principal component analysis. *The Journal of Machine Learning Research 11*, 517–553.

Jung, S. and J. Marron (2009). Pca consistency in high dimension, low sample size context. *The Annals of Statistics 37*(6B), 4104–4130.

Kolda, T. and B. Bader (2009). Tensor decompositions and applications. *SIAM review 51*(3), 455–500.

Kroonenberg, P. (2008). *Applied multiway data analysis*, Volume 702. Wiley Online Library.

Lee, M., H. Shen, J. Huang, and J. Marron (2010). Biclustering via Sparse Singular Value Decomposition. *Biometrics 66*(4), 1087–1095.

Lim, L. and P. Comon (2009). Nonnegative approximations of nonnegative tensors. *Journal of Chemometrics 23*(7-8), 432–441.

Liu, J., J. Liu, P. Wonka, and J. Ye (2012). Sparse non-negative tensor factorization using columnwise coordinate descent. *Pattern Recognition 45*(1), 649–656.

McCullagh, P. (1987). *Tensor methods in statistics*. Chapman and Hall London.

Mitchell, T., R. Hutchinson, R. Niculescu, F. Pereira, X. Wang, M. Just, and S. Newman (2004). Learning to decode cognitive states from brain images. *Machine Learning 57*(1), 145–175.

Mørup, M., L. Hansen, and S. Arnfred (2008). Algorithms for sparse nonnegative tucker decompositions. *Neural computation 20*(8), 2112–2131.

Mørup, M., L. Hansen, S. Arnfred, L. Lim, and K. Madsen (2008). Shift-invariant multilinear decomposition of neuroimaging data. *NeuroImage 42*(4), 1439–1450.

Owen, A. and P. Perry (2009). Bi-cross-validation of the SVD and the nonnegative matrix factorization. *Annals 3*(2), 564–594.

Pang, Y., Z. Ma, J. Pan, and Y. Yuan (2011). Robust sparse tensor decomposition by probabilistic latent semantic analysis. In *Image and Graphics (ICIG), 2011 Sixth International Conference on*, pp. 893–896. IEEE.

Ramsay, J. (2006). *Functional data analysis*. Wiley Online Library.

Ruiters, R. and R. Klein (2009). Btf compression via sparse tensor decomposition. In *Computer Graphics Forum*, Volume 28, pp. 1181–1188. Wiley Online Library.

Shashua, A. and T. Hazan (2005). Non-negative tensor factorization with applications to statistics and computer vision. In *Proceedings of the 22nd international conference on Machine learning*, pp. 792–799. ACM.

Shen, H. and J. Huang (2008). Sparse principal component analysis via regularized low rank matrix approximation. *Journal of multivariate analysis 99*(6), 1015–1034.

Silverman, B. (1996). Smoothed functional principal components analysis by choice of norm. *The Annals of Statistics 24*(1), 1–24.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological) 58*(1), 267–288.

Troyanskaya, O., M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein, and R. Altman (2001). Missing value estimation methods for DNA microarrays. *Bioinformatics 17*(6), 520.

Tseng, P. (2001). Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of optimization theory and applications 109*(3), 475–494.

Tucker, L. (1966). Some mathematical notes on three-mode factor analysis. *Psychometrika 31*(3), 279–311.

Witten, D. M., R. Tibshirani, and T. Hastie (2009). A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. *Biostatistics 10*(3), 515–534.

Yuan, M. and Y. Lin (2006). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 68*(1), 49–67.

Zahn, J., S. Poosala, A. Owen, D. Ingram, A. Lustig, A. Carter, A. Weeraratna, D. Taub, M. Gorospe, K. Mazan-Mamczarz, et al. (2007). Agemap: a gene expression database for aging in mice. *PLoS genetics 3*(11), e201.

Zou, H., T. Hastie, and R. Tibshirani (2006). Sparse principal component analysis. *Journal of computational and graphical statistics 15*(2), 265–286.