Sparse Higher-Order Principal Components Analysis

Genevera I. Allen
Baylor College of Medicine & Rice University

Abstract

Traditional tensor decompositions such as the CANDECOMP / PARAFAC (CP) and Tucker decompositions yield higher-order principal components that have been used to understand tensor data in areas such as neuroimaging, microscopy, chemometrics, and remote sensing. Sparsity in high-dimensional matrix factorizations and principal components has been well-studied exhibiting many benefits; less attention has been given to sparsity in tensor decompositions. We propose two novel tensor decompositions that incorporate sparsity: the *Sparse Higher-Order SVD* and the *Sparse CP Decomposition.* The latter solves an ℓ_1-norm penalized relaxation of the single-factor CP optimization problem, thereby automatically selecting relevant features for each tensor factor. Through experiments and a scientific data analysis example, we demonstrate the utility of our methods for dimension reduction, feature selection, signal recovery, and exploratory data analysis of high-dimensional tensors.

1 Introduction

High-dimensional tensor or multi-modal data is becoming prevalent in areas such as neuroimaging, microscopy, chemometrics, bibliometrics and remote sensing. Often one tries to understand this large quantity of data through dimension reduction or by finding major patterns and modes of variation. To this end, tensor decompositions such as the CANDECOMP / PARAFAC (CP) (Harshman, 1970; Carroll and Chang, 1970) and the Tucker decomposition (Tucker, 1966) have been employed and used for higher-order principal components analysis (PCA) (Kolda and Bader, 2009). While many have studied these decompositions, relatively few have advocated encouraging sparsity in the factors. In the context of non-negative tensor factorizations, several have discussed sparsity (Hazan et al., 2005; Mørup et al., 2008; Liu et al., 2012), and a few have gone on to propose sparsity in one of the tensor factors (Ruiters and Klein, 2009; Pang et al., 2011). In this paper, we are interested in mathematically developing a framework for incorporating sparsity in each tensor factor that will lead to computationally attractive algorithms for high-dimensional tensors.

Sparsity in tensor decompositions and higher-order PCA 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 hence attractive from a data storage perspective. Second, in high-dimensional settings, many features are often irrelevant. With neuroimaging data such as functional MRIs, for example, there are often hundreds of thousands of voxels in each image and many of these voxels are inactive for the entire length of the scanning session. Sparsity gives one an automatic tool for feature selection in high-dimensional tensors. Third, many have noted that PCA is asymptotically inconsistent in high-dimensional settings (Johnstone and Lu, 2009). As this is true for matrix data, it is not hard to surmise that asymptotic inconsistency of the factors holds for higher-order PCA as well. Sparsity in PCA, however, has been shown to yield consistent principal component directions (Johnstone and Lu, 2009; Amini and Wainwright, 2009). Finally for high-dimensional tensors, visualizing and interpreting the higher-order PC's can be a challenge. Sparsity limits the number of features and hence simplifies visualization and interpretation of exploratory data analysis results.

In this paper, we present two novel algorithms for incorporating sparsity into tensor decompositions, or higher-order principal components analysis: the *Sparse Higher-Order SVD* and the *Sparse CP Decom-*

Appearing in Proceedings of the 15^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2012, La Palma, Canary Islands. Volume 22 of JMLR: W&CP 22. Copyright 2012 by the authors.

position. A major theoretical contribution of our work is proving that the latter solves a multi-way concave relaxation of the CP optimization problem, thus providing the mathematical context for algorithms employing a similar structure. Through simulated experiments and a scientific data analysis of tensor microarray data, we demonstrate the effectiveness of our methods for dimension reduction, feature selection, and pattern recognition.

We adopt the notation of 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. Sometimes it is necessary to convert a tensor into a matrix, or matricize the tensor. This is denoted as $\mathbf{X}_{(1)}$ where the subscript denotes 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 $||\,\boldsymbol{\mathcal{X}}\,||_F = \sqrt{\sum_i \sum_j \sum_k \boldsymbol{\mathcal{X}}_{ijk}^2}$. 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.

2 A Sparse Higher-Order SVD

The Higher-Order SVD (HOSVD), or Tucker decomposition, is a popular tool for computing higher-order principal components (Tucker, 1966; De Lathauwer et al., 2000). This decomposition models a three-mode tensor, $\boldsymbol{\mathcal{X}} \in \Re^{n \times p \times q}$ as $\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{D}} \times_1 \mathbf{U} \times_2 \mathbf{V} \times_3 \mathbf{W}$ 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. The factors can be interpreted as the principal components of each tensor mode.

Algorithm 1 Sparse (Truncated) Higher-Order SVD

```
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. $\boldsymbol{\mathcal{D}} \leftarrow \boldsymbol{\mathcal{X}} \times_1 \mathbf{U} \times_2 \mathbf{V} \times_3 \mathbf{W}$.
```

A common method for computing the HOSVD is to estimate 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 matricized along each of the three dimensions. This leads to a simple strategy for obtaining a Sparse HOSVD: Replace PCA with sparse PCA to obtain sparse factors for each tensor mode. This approach is outlined in Algorithm 1. 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 calculate the Sparse HOSVD. We also note that as sparse PC's are commonly not constrained to be orthonormal, the factors of the Sparse HOSVD are no longer orthonormal.

While the Sparse HOSVD is conceptually simple, it is not ideal for several reasons. First, the algorithm does not correspond to solving any optimization problem or minimizing a tensor loss function (Kolda and Bader, 2009). Mathematically, this is less appealing. 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 is simple, it is mathematically and computationally less desirable. To this end, we introduce a novel Sparse CP decomposition in the next section that directly addresses each of these concerns.

3 A Sparse CANDECOMP / PARAFAC Decomposition

The CP decomposition seeks to model a tensor as a sum of rank one tensors: $\boldsymbol{\mathcal{X}} = \sum_{k=1}^{K} d_k\, \mathbf{u}_k \circ \mathbf{v}_k \circ \mathbf{w}_k$, 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 section, we develop a novel Sparse CP decomposition that directly optimizes a relaxation of the CP optimization problem. Before introducing our Sparse CP problem, however, we introduce the algorithmic framework upon which this is based. Namely, we begin by showing that the rank-one CP problem can be solved using a novel algorithm, the Tensor Power Method.

3.1 Tensor Power Method

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

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

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

$$\underset{\mathbf{u},\mathbf{v},\mathbf{w}}{\text{maximize}} \;\; \boldsymbol{\mathcal{X}} \times_1 \mathbf{u} \times_2 \mathbf{v} \times_3 \mathbf{w}$$
$$\text{subject to} \;\; \mathbf{u}^T \mathbf{u} = 1, \mathbf{v}^T \mathbf{v} = 1, \;\&\; \mathbf{w}^T \mathbf{w} = 1. \tag{2}$$

As (2) is separable in the factors, we can optimize this in an iterative block-wise manner:

Proposition 1 *The block coordinate-wise solutions for* (2) *are given by:*

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

Proof 1 *Consider optimizing* (2) *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 Karush-Kuhn-Tucker (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.*

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 (2), 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 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 Method* and summarize this in Algorithm 2.

Notice that the Tensor Power 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: If we define $\mathbf{U}_k = [\mathbf{u}_1 \ldots \mathbf{u}_k]$, then orthogonal updates for $\mathbf{u}_k$ are given by: $\mathbf{u}_k = (\mathbf{I}_{(n)} - \mathbf{U}_{k-1} \mathbf{U}_{k-1}^T) \boldsymbol{\mathcal{X}} \times_2 \mathbf{v}_k \times_3 \mathbf{w}_k /||(\mathbf{I}_{(n)} - \mathbf{U}_{k-1} \mathbf{U}_{k-1}^T) \boldsymbol{\mathcal{X}} \times_2 \mathbf{v}_k \times_3 \mathbf{w}_k ||_2$. Orthogonal updates for $\mathbf{v}$ and $\mathbf{w}$ are analogous.

Algorithm 2 Tensor Power Method

```
1. Initialize $\hat{\boldsymbol{\mathcal{X}}} = \boldsymbol{\mathcal{X}}$.
2. For $k = 1 \ldots K$
    (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) $\hat{\boldsymbol{\mathcal{X}}} \leftarrow \hat{\boldsymbol{\mathcal{X}}} - d_k \, \mathbf{u}_k \circ \mathbf{v}_k \circ \mathbf{w}_k$.
```

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 Method 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 5. 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.

3.2 Sparse CP Decomposition

We introduce a novel Sparse CP decomposition that incorporates sparsity by regularizing the factors with an ℓ_1-norm penalty. Our method solves a direct relaxation of the CP optimization problem (2) and has a computationally attractive solution.

3.2.1 Problem & Solution

Many have sought to encourage sparsity in principal components analysis by solving ℓ_1-norm penalized optimization problems related the to 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 Sparse CP decomposition by placing ℓ_1-norm penalties on each of the tensor factors and relaxing the alternative formulation of the CP optimization problem, (2). Thus, we define our single-factor Sparse CP decomposition as the solution to the following problem:

$$\underset{\mathbf{u},\mathbf{v},\mathbf{w}}{\text{maximize}} \;\; \boldsymbol{\mathcal{X}} \times_1 \mathbf{u} \times_2 \mathbf{v} \times_3 \mathbf{w} - \rho_{\mathbf{u}} || \mathbf{u} ||_1 - \rho_{\mathbf{v}} || \mathbf{v} ||_1 - \rho_{\mathbf{w}} || \mathbf{w} ||_1$$
$$\text{subject to} \;\; \mathbf{u}^T \mathbf{u} \leq 1, \mathbf{v}^T \mathbf{v} \leq 1, \;\&\; \mathbf{w}^T \mathbf{w} \leq 1. \tag{3}$$

Here $\rho_{\mathbf{u}}$, $\rho_{\mathbf{v}}$ and $\rho_{\mathbf{w}}$ are non-negative bandwidth parameters controlling the amount of sparsity in the tensor factors. Notice that we have relaxed the equality constraints from (2) to inequality constraints in (3). Relaxing the constraints greatly simplifies the optimization problem, leading to a simple solution and algorithmic approach. Notice that (3) is concave in each factor with the other factors fixed. Thus, one can optimize this problem by updating one factor at a time and iterating until convergence. As each update involves solving a concave optimization problem, the objective monotonically increases, converging to a local maximum of (3).

The update of each factor in our Sparse CP problem has a simple analytical solution:

Theorem 1 *The block coordinate-wise solutions to the Sparse CP problem are given by:*

$$\hat{\mathbf{u}} = \begin{cases} \frac{S(\boldsymbol{\mathcal{X}} \times_2 \mathbf{v} \times_3 \mathbf{w}, \rho_{\mathbf{u}})}{||S(\boldsymbol{\mathcal{X}} \times_2 \mathbf{v} \times_3 \mathbf{w}, \rho_{\mathbf{u}})||_2} & ||S\left(\boldsymbol{\mathcal{X}} \times_2 \mathbf{v} \times_3 \mathbf{w}, \rho_{\mathbf{u}}\right)||_2 > 0 \\ 0 & \textit{otherwise}, \end{cases}$$

$$\hat{\mathbf{v}} = \begin{cases} \frac{S(\boldsymbol{\mathcal{X}} \times_1 \mathbf{u} \times_3 \mathbf{w}, \rho_{\mathbf{v}})}{||S(\boldsymbol{\mathcal{X}} \times_1 \mathbf{u} \times_3 \mathbf{w}, \rho_{\mathbf{v}})||_2} & ||S\left(\boldsymbol{\mathcal{X}} \times_1 \mathbf{u} \times_3 \mathbf{w}, \rho_{\mathbf{v}}\right)||_2 > 0 \\ 0 & \textit{otherwise}, \end{cases}$$

$$\& \ \hat{\mathbf{w}} = \begin{cases} \frac{S(\boldsymbol{\mathcal{X}} \times_1 \mathbf{u} \times_2 \mathbf{v}, \rho_{\mathbf{w}})}{||S(\boldsymbol{\mathcal{X}} \times_1 \mathbf{u} \times_2 \mathbf{v}, \rho_{\mathbf{w}})||_2} & ||S\left(\boldsymbol{\mathcal{X}} \times_1 \mathbf{u} \times_2 \mathbf{v}, \rho_{\mathbf{w}}\right)||_2 > 0 \\ 0 & \textit{otherwise}, \end{cases}$$

where $S(\cdot, \rho)$ is the soft-thresholding operator: $S(\cdot, \rho) = \text{sign}(\cdot)(|\cdot| - \rho)_+$.

Proof 2 *The proof follows from an extension of results in Witten et al. (2009); Allen et al. (2011). In short, consider optimizing* (3) *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.*

Thus, even with the relaxed constraints in (3), our solution for each factor is guaranteed to either have norm one or be set to zero. Before presenting the algorithm in full, we pause to note some major advantages of our approach: (i) the scale of each factor is directly constrained and thus degenerate solutions are avoided, (ii) (3) is a tri-concave relaxation of (2), thus assuring convergence to a local maximum, and (iii) the coordinate-wise solutions have a simple analytical form.

3.2.2 Algorithm

We employ a deflation approach similar to that of the Tensor Power Method to calculate multiple factors of our Sparse CP decomposition. Specifically, each factor is calculated by solving the single-factor CP problem, (3), for the residuals from the previously computed single-factor solutions. This approach is outlined in Algorithm 3. Notice that we do not enforce orthogonality to the previously computed factors. In fact, many have advocated against enforcing orthogonality of sparse principal components (Zou et al., 2006; Shen and Huang, 2008; Journée et al., 2010). For factors with $\rho = 0$, however, if orthogonality is desired, this can be accomplished by altering the factor updates as described in Section 3.1.

Algorithm 3 Sparse CP Decomposition

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

3.2.3 Selecting Bandwidth Parameters

Our Sparse CP decomposition has three bandwidth parameters that control the amount of sparsity in the factors. Several methods exist for selecting these bandwidth 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 bandwidth parameters via the Bayesian Information Criterion (BIC) (Allen et al., 2011): $\rho_{\mathbf{u}}^* = \text{argmin}_{\rho_{\mathbf{u}}} BIC(\rho_{\mathbf{u}})$ where $BIC(\rho_{\mathbf{u}}) = \log\left(\frac{||\,\boldsymbol{\mathcal{X}} - d\,\mathbf{u} \circ \mathbf{v} \circ \mathbf{w}\,||_F^2}{npq}\right) + \frac{\log(npq)}{npq}|\{\mathbf{u}\}|$, where $|\{\mathbf{u}\}|$ is the number of non-zero elements of $\mathbf{u}$. This BIC formulation can be derived by considering that each update in the Sparse CP Algorithm solves an ℓ_1-norm penalized regression problem. Selection criteria for $\mathbf{v}$ and $\mathbf{w}$ are analogous. Experimental results evaluating the efficacy of this bandwidth selection method are given in Section 4.

3.2.4 Amount of Variance Explained

In principal components analysis, a critical quantity in determining the fit of the SVD model is the amount of variance explained by the rank-K matrix factorization. The same can be used in our sparse higher-order PCA models to determine the extent of dimension reduction achieved. Consider the following result:

Theorem 2 *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 higher-order PC's or sparse higher-order PC's is given by* $||\,\boldsymbol{\mathcal{X}} \times_1 \mathbf{P}_k^{(U)} \times_2 \mathbf{P}_k^{(V)} \times_3 \mathbf{P}_k^{(W)}\,||_F^2 \;/\; ||\,\boldsymbol{\mathcal{X}}\,||_F^2$.

Proof 3 *The proof is an extension of that in Shen and Huang (2008). Recall that the cumulative proportion of variance explained by the first k traditional principal 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,* ψ_k *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 higher-order PCA factors.*

This result requires some further comments and explanation. One familiar with the PCA literature may inquire as to why we cannot simply sum the first k eigenvalues of the covariance matrix or the first k squared singular values of the data matrix. For matrix data, the set of singular vectors form a complete basis and hence can be used to recover the data matrix exactly. The same is not true of tensor decompositions. That is, common tensor factorizations cannot be written in the form of a set of orthonormal basis vectors along each mode multiplied by a diagonal tensor core that recover the tensor exactly (Kolda and Bader, 2009). Because of this, summing d_k^2 does not give the cumulative proportion of variance explained by the tensor decomposition. Instead, one must project the tensor onto the first k factors to compute this quantity.

3.3 Extensions

As our Sparse CP decomposition directly solves a well defined optimization problem, there are many possible extensions of our methodology. We briefly outline two of these here, omitting formal proofs and details.

3.3.1 Regularized CP Decomposition with General Penalties

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 $P_{\mathbf{w}}()$:

$$\begin{aligned} \underset{\mathbf{u},\mathbf{v},\mathbf{w}}{\text{maximize}} \quad & \boldsymbol{\mathcal{X}} \times_1 \mathbf{u} \times_2 \mathbf{v} \times_3 \mathbf{w} \\ & - \rho_{\mathbf{u}} P_{\mathbf{u}}(\mathbf{u}) - \rho_{\mathbf{v}} P_{\mathbf{v}}(\mathbf{v}) - \rho_{\mathbf{w}} P_{\mathbf{w}}(\mathbf{w}) \\ \text{subject to} \quad & \mathbf{u}^T\,\mathbf{u} \le 1, \mathbf{v}^T\,\mathbf{v} \le 1, \;\&\; \mathbf{w}^T\,\mathbf{w} \le 1. \end{aligned} \tag{4}$$

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:* $\hat{\mathbf{u}} = argmin_{\mathbf{u}}\{\frac{1}{2}||\,\boldsymbol{\mathcal{X}} \times_2 \hat{\mathbf{v}} \times_3 \hat{\mathbf{w}} - \mathbf{u}\,||_2^2 + \rho_{\mathbf{u}} P_{\mathbf{u}}(\mathbf{u})\}$*,* $\hat{\mathbf{v}} = argmin_{\mathbf{v}}\{\frac{1}{2}||\,\boldsymbol{\mathcal{X}} \times_1 \hat{\mathbf{u}} \times_3 \hat{\mathbf{w}} - \mathbf{v}\,||_2^2 + \rho_{\mathbf{v}} P_{\mathbf{v}}(\mathbf{v})\}$*, and* $\hat{\mathbf{w}} = argmin_{\mathbf{w}}\{\frac{1}{2}||\,\boldsymbol{\mathcal{X}} \times_1 \hat{\mathbf{u}} \times_2 \hat{\mathbf{v}} - \mathbf{w}\,||_2^2 + \rho_{\mathbf{w}} P_{\mathbf{w}}(\mathbf{w})\}$*. Then, the block coordinate-wise solutions for* (4) *are given by:* $\mathbf{u}^* = \begin{cases} \frac{\hat{\mathbf{u}}}{||\hat{\mathbf{u}}||_2} & ||\hat{\mathbf{u}}||_2 > 0 \\ 0 & \mathit{otherwise} \end{cases}$*,* $\mathbf{v}^* = \begin{cases} \frac{\hat{\mathbf{v}}}{||\hat{\mathbf{v}}||_2} & ||\hat{\mathbf{v}}||_2 > 0 \\ 0 & \mathit{otherwise} \end{cases}$*, and* $\mathbf{w}^* = \begin{cases} \frac{\hat{\mathbf{w}}}{||\hat{\mathbf{w}}||_2} & ||\hat{\mathbf{w}}||_2 > 0 \\ 0 & \mathit{otherwise} \end{cases}$.

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).

3.3.2 Sparse Non-negative CP Decomposition

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 1 allows us to solve (3) 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.

4 Experiments

We evaluate the performance of our Sparse HOSVD and Sparse CP algorithms on simulated data sets. All data is simulated from the following model: $\boldsymbol{\mathcal{X}} =$

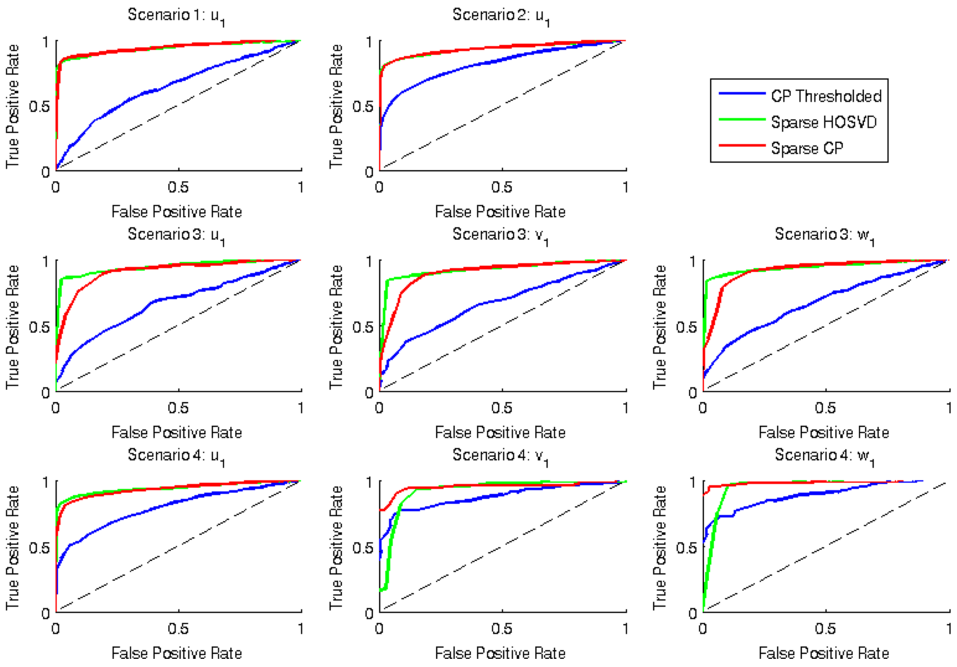

Figure 1: *Mean ROC curves for each of the four simulation scenarios.*

$\sum_{k=1}^{K} d_k \, \mathbf{u}_k \circ \mathbf{v}_k \circ \mathbf{w}_k + \boldsymbol{\mathcal{E}}$, 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 below:

- Scenario 1: 100 × 100 × 100 with $\mathbf{U}$ sparse.
- Scenario 2: 1000 × 20 × 20 with $\mathbf{U}$ sparse.
- Scenario 3: 100 × 100 × 100 with $\mathbf{U}$, $\mathbf{V}$ and $\mathbf{W}$ sparse.
- Scenario 4: 1000 × 20 × 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 $d_2 = 100$.

First, we test the accuracy of our methods in selecting the correct non-zero features. Mean receiver-operator curves over ten replicates computed by varying the bandwidth parameters are given for each of the four scenarios in Figure 1. The Sparse HOSVD (using the Sparse PCA method of (Shen and Huang, 2008)) and Sparse CP are compared to taking the CP-ALS solution and hard-thresholding the factor elements. From these comparisons, we see that the Sparse HOSVD and Sparse CP methods perform equally well at recovering the true relevant features in the tensor. The Sparse HOSVD method performs slightly better than the Sparse CP method for scenario 3 where the dimensions are equal. The Sparse CP method performs better when there are dimensions of unequal sizes.

Next, we assess the performance of our methods for a fixed value of ρ, set to the optimal value as estimated via BIC. The accuracy in terms of true and false positives as well as signal recovery of the tensor measured by the mean squared error is given in Table 1. Simulations were repeated 50 times and the results averaged. Sparse HOSVD and Sparse CP perform comparably for scenarios where only the factor associated with the first mode is sparse. Sparse HOSVD, however, selects less false positives for the scenarios where all factors are sparse, although Sparse CP has better error rates for the second components. Both sparse higher-order PCA methods perform much better than the CP decomposition in terms of signal recovery.

Finally, we compare the average time until convergence of ten replicates of the Sparse HOSVD and Sparse CP methods in Table 2. Data was simulated according to the model described above with all factors sparse and all bandwidth parameters set to one. 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). These results indicate that the Sparse CP method is much faster than the Sparse HOSVD method, especially for tensors in which one dimension is large.

	Scenario 1			Scenario 2		
	CP	Sparse HOSVD	Sparse CP	CP	Sparse HOSVD	Sparse CP
TP / FP $\mathbf{u}_1$	-	0.9428 / 0.0512	0.9512 / 0.0808	-	0.8278 / 0.0430	0.8387 / 0.0717
TP / FP $\mathbf{u}_2$	-	0.8460 / 0.0544	0.8920 / 0.0812	-	0.6530 / 0.0467	0.6665 / 0.0584
Signal Recovery $\hat{\mathcal{X}}$	1.0076	0.0499	0.0503	1.0109	0.1240	0.1247
	Scenario 3			Scenario 4		
	CP	Sparse HOSVD	Sparse CP	CP	Sparse HOSVD	Sparse CP
TP / FP $\mathbf{u}_1$	-	0.9344 / 0.0424	0.9592 / 0.1944	-	0.8268 / 0.0447	0.8562 / 0.1416
TP / FP $\mathbf{u}_2$	-	0.8476 / 0.0660	0.9260/ 0.2564	-	0.6570 / 0.0468	0.7158 / 0.1310
TP / FP $\mathbf{v}_1$	-	0.9440 / 0.0404	0.9628 / 0.2088	-	0.9680 / 0.0500	0.9720 / 0.1480
TP / FP $\mathbf{v}_2$	-	0.8456 / 0.0580	0.9292 / 0.2548	-	0.9380 / 0.1500	0.9580 / 0.1440
TP / FP $\mathbf{w}_1$	-	0.9308 / 0.0540	0.9560 / 0.2112	-	0.9800 / 0.0680	0.9800 / 0.1160
TP / FP $\mathbf{w}_2$	-	0.8556 / 0.0544	0.9348 / 0.2636	-	0.9600 / 0.1820	0.9680 / 0.1600
Signal Recovery $\hat{\mathcal{X}}$	1.0084	0.0495	0.0502	1.0115	0.1238	0.1254

Table 1: *True and false positives and signal recovery measured in mean squared error for the four simulation scenarios.*

	$100 \times 100 \times 100$	$1000 \times 20 \times 20$	$2000 \times 20 \times 20$
	$K = 1$		
Sparse CP	0.181	0.084	0.200
Sparse HOSVD	7.915	6.097	23.911
	$K = 2$		
Sparse CP	0.521	0.154	1.437
Sparse HOSVD	16.525	12.456	49.082

Table 2: *Average time in seconds.*

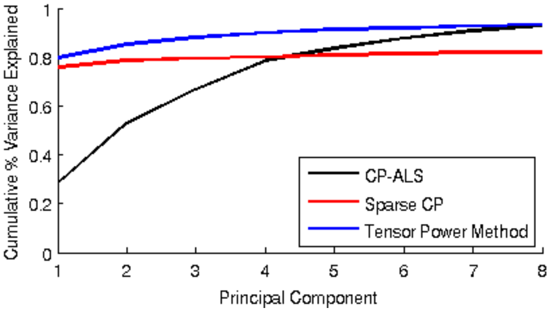

Figure 2: *Cumulative proportion of variance explained by higher-order principal components computed via the CP-ALS, Tensor Power Method, and Sparse CP methods for the AGEMAP microarray data.*

Overall, our experiments indicate that both the Sparse HOSVD and Sparse CP methods perform well at selecting the relevant features and in signal recovery. As the Sparse CP is computationally much more efficient, this is then our preferred method for data analysis.

5 Scientific Data Analysis: AGEMAP Microarray Data

We use Sparse HOPCA to understand the multi-way AGEMAP microarray data. 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 flattened tensor.

The Tensor Power Method, CP decomposition, and Sparse CP decomposition were applied to this data to reduce the dimension and understand patterns among tissues and genes. In Figure 2, the cumulative proportion of variance explained by the first eight components is given. Notice that both the CP-ALS and Tensor Power Method explain roughly the same proportion of variance with eight components, but the Tensor Power Method explains much more initial variance. As often scientists are only interested in the first couple principal components, this illustrates an important advantage of our approaches in the analysis of real data: The Tensor Power Method and Sparse CP methods achieve greater initial dimension reduction.

In Figure 3, we explore patterns found in the AGEMAP data via sparse higher-order PCA. The results were computed using the Sparse CP method placing a penalty on the gene mode with the BIC used to select bandwidth parameters. In the top panel, we show scatterplots of the first eight principal components for the tissue mode. We see many clusters of tissue types for the various pairs of principal components. For example, adrenal, gonads and bones often cluster together.

As only a subset of genes are selected by each of the principal components, we can analyze the genetic patterns further for each tissue type. Gonads has a higher

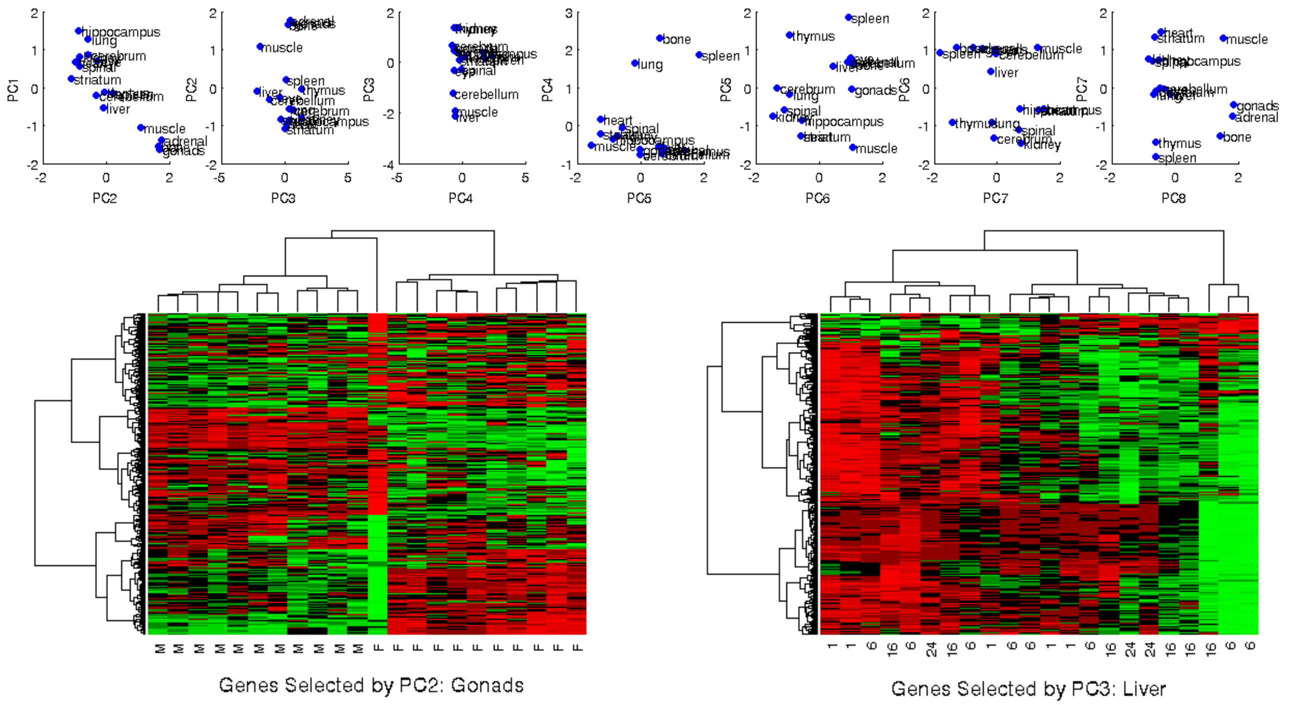

Figure 3: *Analysis of AGEMAP Microarray data via Sparse Higher-Order PCA. (Top Panel) Scatterplots of the first eight principal components for the 16 tissue types. (Lower Left) A cluster heatmap labeled by gender of the genes selected by sparse PC 3 for the tissue Gonads. (Lower Right) A cluster heatmap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

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. Note that Ward linkage with Euclidean distance is used to compute the dendograms. 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 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 analysis of this type reveals subsets of important genes associated with various tissues and mice ages and gender.

This scientific data analysis has illustrated the strengths of our Sparse CP method for dimension reduction and finding relevant features and patterns in high-dimensional tensor data.

6 Discussion

In this paper, we have introduced novel methods for incorporating sparsity into tensor decompositions and higher-order PCA. The Sparse HOSVD performs sparse PCA on matricized tensors. While this method is conceptually simple and performs well, it is slower computationally and less desirable mathematically as it does not minimize a loss function. The Sparse CP decomposition, on the other hand, directly maximizes an ℓ_1-norm penalized optimization problem related to the CP decomposition. This method performs nearly as well as the Sparse HOSVD and is computationally much faster.

We have presented our methods for three-mode tensors for notational convenience, but all of our methods can be trivially extended to tensors of higher-order. There are many other aspects of our work that require further investigation. The Tensor Power Algorithm appears to be an attractive alternative to the CP-ALS algorithm. Further comparisons of timings, convergence rate, variance explained, and signal recovery are needed. Also, the extensions of our methods outlined in Section 3.3 can be further developed and formally compared to existing methods in the literature. Finally, there are possibly many other ways to formulate sparse tensor decompositions based on existing algorithms for the CP and Tucker decompositions.

In conclusion, we have developed methods to perform sparse higher-order principal components analysis that open many new possibilities both methodologically and in applications to high-dimensional tensor data.

References

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.*

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.

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.

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.

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

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

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.

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

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.

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.

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.

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.