Title
Array Independent Component Analysis with Application to Remote Sensing

Author
Kukuyeva, Irina A.

Publication Date
2012

Peer reviewed|Thesis/dissertation

University of California
Los Angeles

Array Independent Component Analysis with Application to Remote Sensing

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Statistics

by

Irina A. Kukuyeva

2012

Abstract of the Dissertation

Array Independent Component Analysis with Application to Remote Sensing

by

Irina A. Kukuyeva
Doctor of Philosophy in Statistics
University of California, Los Angeles, 2012
Professor Jan de Leeuw, Chair

There are three ways to learn about an object: from samples taken directly from the site, from simulation studies based on its known scientific properties, or from remote sensing images [64]. All three are carried out to study Earth and Mars. Our goal, however, is to learn about the second largest storm on Jupiter, called the White Oval, whose characteristics are unknown to this day [55]. As Jupiter is a gas giant and hundreds of millions of miles away from Earth [13], we can only make inferences about about the planet from retrieval algorithms and remotely sensed images.

Our focus is to find latent variables from the remotely sensed data that best explain its underlying atmospheric structure. Principal Component Analysis (PCA) is currently the most commonly employed technique to do so. For a data set with more than two modes, this approach fails to account for all of the variable interactions, especially if the distribution of the variables is not multivariate normal; an assumption that is rarely true of multispectral images. The thesis presents an overview of PCA along with the most commonly employed decompositions in other fields: Independent Component Analysis, Tucker-3 and CANDECOMP/PARAFAC and discusses their limitations in finding unobserved, independent structures in a data cube.

We motivate the need for a novel dimension reduction technique that generalizes existing decompositions to find latent, statistically independent variables for one side of a multimodal (number of modes greater than two) data set while accounting for the variable interactions with its other modes. Our method is called Array Independent Component Analysis (AICA). As the main question of any decomposition is how to select a small number of latent variables that best capture the structure in the data, we extend the heuristic developed by Ceulemans and Kiers in [10] to aid in model selection for the AICA framework.

The effectiveness of each dimension reduction technique is determined by the degree of interpretability of the uncovered hidden variables. AICA discovered two temperature gradients of the White Oval that matched the ones above the visible clouds of the Great Red Spot (in the North-South and West-East directions) [26], isolated the small storm to the South-East of the White Oval and identified the Raleigh scatter of gas molecules [52]. The other techniques, including PCA, did not isolate these or any other interpretable components.

The dissertation of Irina A. Kukuyeva is approved.

Amy Braverman

Mark Hansen

Kuo-Nan Liou

Hongquan Xu

Jan de Leeuw, Committee Chair

University of California, Los Angeles

2012

To the love of my life – Neal.

Table of Contents

List of Figures

List of Tables

Acknowledgments

First I want to thank my advisor Jan de Leeuw, without whose immense experience and expertise in multivariate analysis this would not have been possible. Thank you for allowing me to pursue this topic and for the patience to see it through. I would also like to thank my committee members Amy Braverman, Mark Hansen, Kuo-Nan Liou and Hongquan Xu – for input and trust in me and my formulas with four indices.

Amy Simon-Miller and Padma Yanamandra-Fisher – thank you for generously providing the data set for the analysis, and, along with Amy Braverman, for introducing me to Jupiter.

I also would like to thank Nicolas Christou, Vivian Lew, Juana Sanchez, Rob Gould and Glenda Jones, whom I had the pleasure of working with, even as an undergraduate at UCLA. Thank you for your encouragement, academic and otherwise, and for showing me what it means to be a great teacher and consultant. Nicolas, thank you for teaching 100B and C many years ago; I might not have been a Statistician otherwise.

I am grateful for the encouragement of my fiends – too many to name here, unfortunately. I want to include a special thank you to Nadya Kramerova, Haining Ren, Jean Wang, Rakhee Patel and Natasha Ericta, whose support helped me see the light at the end of the tunnel more than once. I also cannot leave out thanking my friends, office mates and the Tuesday lunch crew – Yuliya Marchetti and Linda Zanontian – your healthy dose of humor and reality always make my day.

My deepest gratitude goes out to my parents, who sacrificed a lot for their daugh-

ters' futures in America. While my mom always wanted a doctor in the family, I hope she would have been proud of the one we got now. Dad, thank you for having faith me, one page at a time.

Last – but not least – I would like to express my sincerest appreciation to Neal. Without your unwavering patience and support I might not have finished. Your hard work paid off!

This research has been made possible with partial support by Grant Number NNX08AW15H from National Aeronautics Space Administration's Graduate Student Researchers Program. The content of this thesis is solely the responsibility of the author and does not necessarily represent the official views of the National Aeronautics Space Administration.

Thank you all!

Vita

2007	B. of A. and S. (Statistics, Economics) Minor. (Mathematics) UCLA, Los Angeles, California.
2006-2008	Research Assistant. Graduate Student Researcher in Sept 2007. UCLA Center for Environmental Statistics. UCLA, Los Angeles, California.
2008-2011	NASA Graduate Student Researchers Program.
2009	M.S. (Statistics) UCLA, Los Angeles, California.
2009-2011	Graduate Student Consultant. UCLA Department of Statistics, Statistical Consulting Center. UCLA, Los Angeles, California.
2011	C.Phil. (Statistics) UCLA, Los Angeles, California.
2011-2012	Data Analyst, ValueClick Media. Westlake Village, California.
2012	Director of Marketing Sciences, Ipsos Science Center. Culver City, California.

Publications and Presentations

I. Kukuyeva. *Array Component Analysis with Application to Remote Sensing Atmospheric Science Data.* Joint Statistical Meetings 2012. San Diego Convention Center, San Diego. 29 Jul. 2012.

I. Kukuyeva. *Exploratory Data Analysis of Three Mode Atmospheric Science Data.* Center for Environmental Statistics. UCLA, Los Angeles. 30 Nov. 2010.

I. Kukuyeva. *Multivariate Statistical Analysis for Planetary Science.* Center for Environmental Statistics. UCLA, Los Angeles. 18 May 2010.

I. Kukuyeva. Review of Principles of Modeling and Simulation: A Multidisciplinary Approach, by John A. Sokolowski and Catherine M. Banks (eds.). In *Journal of Statistical Software, Book Reviews.* 25 Sept. 2009. Vol. 31, Book Review 3. `http://www.jstatsoft.org/v31/b03`

I. Kukuyeva., A. Braverman, P. Yanamandra-Fisher, J. de Leeuw and A. Simon-Miller. *Multivariate Analysis in Planetary Science: Understanding Jupiter's Atmosphere.* Joint Statistical Meetings 2009. Walter E Washington Convention Center, Washington DC. 3 Aug. 2009.

A. Braverman, I. Kukuyeva, P. Yanamandra-Fisher, D. Holt, J. de Leeuw and A. Simon-Miller. *Multivariate Analysis in Planetary Science: Understanding the Atmospheres of Jupiter and Saturn.* Joint Statistical Meetings 2008.

CHAPTER 1

Introduction

1.1 Statement of the Problem

Multivariate data sets are common in statistics, though it is not usually the case that all variables and their interactions can be modeled simultaneously. In many cases, analyses suffer from the "curse of dimensionality" and need to find a compromise between model complexity and fit. Because of this tradeoff, researchers use dimension reduction techniques to simultaneously reduce the data volume while retaining most of the information in the data set. We extend these techniques by relaxing the normality assumption for a multivariate data set with more than two modes in order to reduce its volume and to find latent, independent variables for one of its modes, accounting for the variable interactions with its neighbors.

We call our method Array Independent Component Analysis (AICA) and apply it to multispectral images of the White Oval, a storm on Jupiter, captured by the Hubble Space Telescope and the Infrared Telescope Facility. By reducing the data volume, we are interested in learning about the characteristics of the storm, which are unknown to this day [55]. AICA is able to extract temperature profiles, the small storm to the South-East of the White Oval, and the Raleigh scatter of gas molecules [52]. These findings outperform previously developed and most commonly used dimension reduction techniques because they arrived at directly interpretable components.

1.2 Data Representation

We consider all data sets to be arrays, which generalize the notion of vectors and matrices, and make a distinction between a mode of an array and the dimension of a mode. With this formulation, a vector X_N is an array that has only one mode of dimension N, where N is the number of elements in the vector; a matrix $\mathbf{X}_{N\times M}$ is an array with two modes, with dimensions N and M, respectively for number of rows and columns; a data cube $\mathbf{X}_{N\times M\times K}$ is an array with three modes, with dimensions N, M, and K, respectively for its height, width and depth. For a more concrete example, an array with two modes may store responses of 10 subjects to five questions, as shown in Figure 1.1, where Mode 1 contains the 10 subjects, one per row; and Mode 2 corresponds to the subjects' responses to five questions, one per column. Therefore, the dimensionality of Mode 1 is 10, and five for Mode 2.

An example of a data cube $\mathcal{X}_{N\times M\times K}$, which has three modes, is shown in Figure 1.2. We can use it, for instance, to represent 10 subjects' ratings for three brands on five attributes, as shown in Figure 1.3. Then Mode 1 corresponds to 10 subjects, one in each row; Mode 2 to each of five questions, one per column, asked of each subject about a particular brand; and Mode 3 associates that information with each brand; there will be one questionnaire for each brand, from each respondent. The resulting dimensions of the modes of this array are: ten, five, and three respectively.

1.3 Notation

In this thesis, we use the following conventions. Upper case, italicized letters in calligraphic font, as in $\mathcal{X}$, are arrays with three or more modes. Its slices or two mode arrays are denoted by boldface upper case letters, for instance, $\mathbf{X}$. Columns

	Q_1	Q_2	Q_3	Q_4	Q_5
$Subject_1$					
$Subject_2$					
$Subject_3$					
$Subject_4$					
$Subject_5$					
$Subject_6$					
$Subject_7$					
$Subject_8$					
$Subject_9$					
$Subject_{10}$					

Figure 1.1: Schematic of a two mode array that contains 10 subjects' responses to five questions about a particular brand.

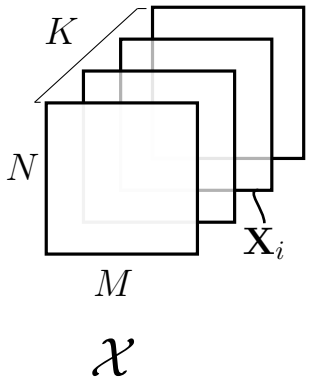


Figure 1.2: Schematic of a three mode data set $\mathcal{X}_{N\times M\times K}$, where each $\mathbf{X}_i$ is a slice of the data set.

of two mode arrays are represented by italicized upper case letters, such as X_i. Scalars or elements of an array with an appropriate number of subscripts are in italicized lower case letters: x, x_i, x_{ij} or $x_{ij\cdots k}$.

1.4 Thesis Outline

The thesis is organized as follows: Chapter 2 provides an overview of the most commonly used dimension reduction techniques; Chapter 3 introduces a novel

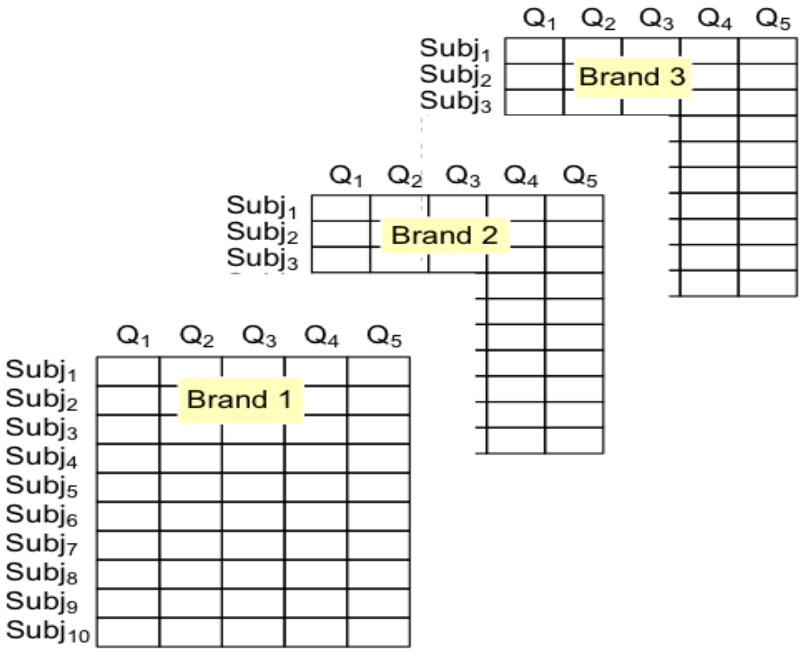


Figure 1.3: Schematic of a three mode array that contains 10 subjects' responses to five questions for three brands.

dimension reduction technique called AICA, which extends previously developed methodology and finds independent components for one side of a data cube, accounting for the variable interactions with its neighboring sides; Chapter 4 introduces a heuristic for selecting the number of components per mode for the dimension reduction techniques discussed in the thesis; Chapter 5 presents an overview of the White Oval on Jupiter and the interpretable results of AICA in the analysis of the remotely sensed images of the storm. We close with a discussion of findings and remarks on further research in Chapter 6. The relevant background needed to understand the concepts presented here are introduced in Appendices A-C. We found results of previously developed dimension reduction techniques uninterpretable; we include them for completeness in Appendix D.

CHAPTER 2

Previous Work in Dimension Reduction Techniques

2.1 Overview

Dimension reduction techniques find latent (or unobserved) variables, called components, which reveal the data set's structure more effectively than the original (raw) variables. Usually a relatively small number of components are extracted and analyzed as a new data set, thereby reducing the data volume [45], [47].

The methods discussed in the thesis assume that the latent variables or components are linearly mixed within our data set. These techniques find loading matrices that tell us how to weight (or load) the observed variables such that we un-mix the components. By examining the size and magnitude of the coefficients used to form these components, we gain insight into which variables are more important in explaining the original relationships in the data set [44]. There have been many approaches developed for finding the components. They differ in how they quantify the structure in the data set, determine how many components capture most of its information and how to find the coefficients needed to form these components.

We reduce the volume of the data set by extracting a smaller number of components than the number of variables in a given mode. Suppose that our data set $\mathcal{X}$ is an $N \times M \times K$ data cube whose latent structure we want to find by reducing

its volume to get a smaller data cube $\mathcal{Y}$. There are three scenarios for the sizes of the modes for $\mathcal{Y}$. We can (without loss of generality) either reduce the depth of $\mathcal{X}$ only, to get an $N \times M \times L$ ($L < K$) cube; or reduce all three of its sides to get a $P \times Q \times R$ ($P < N$, $Q < M$, $R < K$) cube[1]; or reduce all three of its sides such that they are also equal, to get a cube that is $R \times R \times R$ ($R < N$, $R < M$, $R < K$). This chapter discusses how to find $\mathcal{Y}$ via the most commonly employed dimension reduction approaches for two and three mode data sets. First we present the techniques aimed at analyzing a two mode representation[2] of a data cube. We then outline some of the existing methods developed for analyzing the data cube directly via three mode data decompositions. The last section of this chapter contrasts the approaches, and paves the way to understanding the AICA framework and its contribution to extending existing dimension reduction techniques.

2.2 Two Mode Dimension Reduction Techniques

2.2.1 Principal Component Analysis of Two Mode Data

Principal Component Analysis (PCA) is the most commonly used dimension reduction technique. Its goal is to find uncorrelated components that best recreate the variance of a two mode data set in fewer dimensions. We only need to know the mean and variance of the data set (second order statistics) [3] to do so.

To perform PCA, we unfold the original data set $\mathcal{X}_{N \times M \times K}$ as outlined in Appendix A.2.1.1 to get $\tilde{\mathbf{X}}_{NM \times K}$. Then, we standardize[4] the columns of $\tilde{\mathbf{X}}_{NM \times K}$ and denote the new data set by $\mathbf{X}_S$. Let $\mathbf{Z} = \mathbf{X}_S^T \mathbf{X}_S$ be the $K \times K$ full rank

[1] If there were only two strict inequalities, the decomposition is called Tucker-2; the one with only one strict inequality is a Tucker-1 or PCA [42]. We focus on the most general case, which reduces all three modes and is hence called Tucker-3.

[2] Please see Appendix A for an overview of how to reshape a data set to have two modes.

[3] Please see Appendix B for an introduction to order statistics.

[4] To standardize the data values, first subtract the column means from the column's observations and then divide by the column's standard deviation.

variance-covariance[5] data matrix, where K is the dimensionality of the third mode of $\mathcal{X}$. We then minimize the loss function in Equation 2.1 to find $\mathbf{U}$ and $\mathbf{V}$ with diagonal constraints on their sums of squares.

$$\min_{\substack{\mathbf{U}^T\mathbf{U} \text{ diagonal} \\ \mathbf{V}^T\mathbf{V} \text{ diagonal}}} ||\mathbf{Z} - \mathbf{UV}||^2. \tag{2.1}$$

Following the derivation of Appendix C.3, PCA is deterministic and we only need to perform Singular Value Decomposition (SVD) on $\mathbf{Z}$ to decompose it without error and find orthogonal eigenvectors $\mathbf{U}$ ($\mathbf{U}^T\mathbf{U} = \mathbf{I_K}$) such that[6]:

$$\mathbf{Z} = \mathbf{UDU}^T = \sum_{i=1}^{K} \lambda_i^2 U_i U_i^T = \sum_{i=1}^{K} \lambda_i^2 (U_i \otimes U_i), \tag{2.2}$$

where $\mathbf{D}$ is a $K \times K$ diagonal matrix of eigenvalues denoted by λ_i^2 (or squares of singular values λ_i); and $\otimes$ denotes the outer product.[7] The expression on the right hand side of Equation (2.2) [43] states that $\mathbf{Z}$ can also be expressed as a sum of rank-one matrices. We return to this idea in Section 2.3.2 when discussing generalizations of SVD to data sets with more than two modes.

To find the latent variables in PCA, called the Principal Components (PCs) and denoted by $\mathbf{Y}$, weight the columns of the variance-covariance matrix $\mathbf{Z}$ by the eigenvectors $\mathbf{U}$ [2] as expressed in Equation 2.3:

$$\mathbf{Y} = \mathbf{ZU}. \tag{2.3}$$

In PCA literature, $\mathbf{U}$ is usually called the loading matrix. By the properties of SVD, the components resulting from the rotation of the variables by the loading matrix are uncorrelated with one another and have maximum variance. For

[5]Because, in this case, we have standardized the data values prior to finding their variance, the matrix $\mathbf{Z}$ is also the correlation matrix. To be consistent with the methodology discussed later in the thesis, we will, nonetheless, refer to $\mathbf{Z}$ as a variance-covariance matrix.

[6]Because $\mathbf{Z}$ is symmetric, the result of SVD is equivalent to an Eigen Decomposition (ED). PCA can also be used to decompose the original standardized (not square) data set $\mathbf{X}_S$ via Singular Value Decomposition as $\mathbf{X}_S = \mathbf{UDV^T}$, where the singular vectors $\mathbf{U}$ from SVD are equivalent to eigenvectors $\mathbf{U}$ from the ED of the variance-covariance matrix of $\mathbf{X}_S$.

[7]For an overview of outer product, please see Appendix A.

Gaussian data, uncorrelated components are also independent [33].[8]

Since we want to reduce the depth of the original data set, suppose that we chose to extract L components ($L < K$).[9] As a result, the smaller loading matrix $\mathbf{U}_0$ will have L columns, corresponding to the eigenvectors associated with the largest eigenvalues.[10] Then, following the logic of Equation 2.3, we will have L principal components, each of length NM:

$$\mathbf{Y}_0 = \mathbf{Z}\mathbf{U}_0. \tag{2.4}$$

We can approximate the original data by manipulating Equation (2.4) to get:

$$\hat{\mathbf{Z}} = \mathbf{Y}_0\mathbf{U}_0^T, \tag{2.5}$$

as the inverse of an orthonormal matrix is its matrix transpose. We use Equation 2.5 to evaluate how well we captured the information in the preprocessed data set by extracting a limited number of PCs. To do so, we compute and analyze the residuals for patterns. Equation 2.6 expresses the residuals as any structure left over in the data set after we approximate it:

$$\begin{aligned} \mathbf{\Psi} &= \mathbf{Z} - \hat{\mathbf{Z}} \\ &= \mathbf{Z} - \mathbf{Y}_0\mathbf{U}_0^T \\ &= \mathbf{Z} - (\mathbf{Z}\mathbf{U}_0)\,\mathbf{U}_0^T. \end{aligned} \tag{2.6}$$

If patterns are present in the residuals, then the model has not captured all of the important features of the data. In that case, one can either include a different number of components (per rule of thumb discussed in Chapter 4) or switch to a different dimension reduction technique.

[8] Please see Appendix B for further details.

[9] Please see Appendix D for a discussion on how to choose the number of PCs to extract.

[10] If matrix $\mathbf{Z}$ was not of full rank, then in the decomposition, some eigenvalues will be exactly zero. Hence we will be able to reduce the dimensionality of the matrix without compromising the model fit by omitting the columns of $\mathbf{U}$ associated with null eigenvalues.

Chapter 5 provides an example of PCA applied to a remotely sensed data set and compares the resulting components to those of other techniques discussed in the thesis.

2.2.2 Independent Component Analysis of Two Mode Data

For arrays with two modes, PCA finds uncorrelated components [33]. They are also independent if the distribution of the variables is jointly normally distributed [33]. If that is not the case, we can find independent components of a two mode data set via ICA and reduce its depth by selecting a smaller number of Independent Components (ICs) than the size of its third mode.[11] Unlike PCA, there is no closed form solution to ICA and components are not ordered in terms of a percentage of variance of the data set explained, but by largest measure of non-gaussianity, which acts as a proxy for quantifying variable independence [33]. To familiarize the reader with ICA, we begin with an overview and then discuss the most common optimization criteria used for finding the components.

The preprocessing steps of ICA are similar to performing PCA. We first unfold the data array (per Appendix A) and center the columns to get an $NM \times K$ array $\tilde{\mathbf{X}}_C$ [33]. We then decompose the full-rank variance-covariance matrix $\mathbf{Z}_C = \tilde{\mathbf{X}}_C^T \tilde{\mathbf{X}}_C$ of the centered data via SVD: $\mathbf{Z}_C = \mathbf{U}\mathbf{D}\mathbf{U}^\mathbf{T}$, where $\mathbf{D}$ is a $K \times K$ diagonal matrix of eigenvalues denoted by λ_i^2 (or squares of singular values λ_i) and $\mathbf{U}^T\mathbf{U} = \mathbf{I_K}$ [33]. If we define the whitening matrix to be $\mathbf{V} = \mathbf{D}^{-\frac{1}{2}}\mathbf{U}^T$, then we can decorrelate the data to get $\mathbf{Z}_D = \tilde{\mathbf{X}}_C\mathbf{V}$ [33]. It is this data set with uncorrelated components that we rotate toward independence via $\mathbf{W}$. As ICA does not have a closed form solution, we optimize an objective function or contrast to find it.[12] The resulting components $\mathbf{S} = \mathbf{Z}_D\mathbf{W}$, will be statistically independent.

[11] Please see Chapter 4 for a discussion of how to choose the number of Independent Components to extract.

[12] Please see Appendix C for an overview of contrast functions.

To reduce the dimensionality of the original data set, suppose we chose to extract L components ($L < K$).[13] We denote the smaller set of these via Equation 2.7:

$$\mathbf{S}_0 = \mathbf{Z}_D \mathbf{W}_0 \tag{2.7}$$

and use it to evaluate how well a limited number of components L ($L < K$) capture the salient features of the preprocessed data set. To do so, we evaluate the residuals from the approximation:

$$\begin{aligned} \mathbf{\Phi} &= \mathbf{Z}_D - \hat{\mathbf{Z}}_D \\ &= \mathbf{Z}_D - \mathbf{S}_0 \mathbf{A}_0, \end{aligned} \tag{2.8}$$

where $\mathbf{A}_0$ is the Moore-Penrose inverse[14] of the matrix $\mathbf{W}_0$. One way to evaluate model fit is to examine plots of residuals versus each of the original variables for patterns. If such patterns exist, our model did not capture all of the important information and either we need to include a different number of components (as outlined in Chapter 4) or another technique is more appropriate.

We now take a step back and present the most common optimization criteria one can employ to quantify variable independence and find $\mathbf{W}$.[15] By definition of ICA, we know that the observed centered variables of the data set $\tilde{\mathbf{X}}_C$ are made up of a sum of independent sources that we have to un-mix via $\mathbf{W}$ [33]. The Central Limit Theorem tells us that (under certain conditions) the distributions of these sums tends toward a Normal distribution as we extract a large (large $\rightarrow \infty$) number of components [33], [15]. Therefore, to un-mix the ICs, we search for one component in each maximally non-gaussian direction [33]. We begin by describing how to do so in terms of Mutual Information.

[13] Please see Chapter 4 for a discussion on how to choose the number of components to extract.

[14] The Moore-Penrose inverse is a generalized inverse for non-square matrices.

[15] Please see Appendix C.2 for an overview of properties for a loss function, aimed at evaluating source separation criteria.

2.2.2.1 Contrast Based on Mutual Information

Mutual Information (MI) is one way to quantify variable independence for a random variable; in our case, the random variable is an independent component. MI measures how much information is lost when we estimate the joint distribution of the sums of the components by one whose components are independent [15], [16], [33], [71]. The quantity is always nonnegative and vanishes if and only if a variable is statistically independent [33], [32].[16] It is formulated via Kullback-Leibler divergence [15], [16], [33], [71]:

$$I(p_{\mathbf{S}}) = \mathrm{KL}\left(p_{\mathbf{S}}; \prod_{i=1}^{K} p_{S_i}\right) = \int p_{\mathbf{S}} \log \frac{p_{\mathbf{S}}}{\prod_{i=1}^{M} p_{S_i}} \, d_{\mathbf{S}}, \tag{2.9}$$

where $p_{\mathbf{S}}$ represents the joint probability distribution function (PDF) for the sums of the components (or the PDF of the original data set); and p_{S_i} represents the marginal PDF of one component [16].

MI is difficult to estimate in practice, as we have to integrate over the PDF [33]. As a result, there are two possible estimation scenarios. Authors in [14] and [5] (as discussed in [32]) recommend approximating MI via polynomial density estimations employing higher order cumulants, which, unfortunately, are sensitive to outliers [33]. Alternatively, we can use kernel density estimation, but the result will be sensitive to the kernel function and smoothing parameters used [33], [32]. We discuss the former in more details in the following section, as it is the least subjective among the two approaches.

2.2.2.2 Contrast Based on Approximations of Mutual Information

One way to approximate the probability density function of a variable is via an Edgeworth expansion [33], [32]. For a standardized univariate random variable x,

[16] Because our goal is to extract a predetermined number of independent components at the same time (deemed the "parallel" approach by Hyvärinen et al. in [32] as opposed to the "deflatory" approach where the components are extracted one-by-one), we use MI to quantify nongaussianity.

an estimate for MI via an Edgeworth expansion is:

$$I(p_x) \approx \text{Constant} + \frac{1}{12}\kappa_3^2 + \frac{1}{48}\kappa_4^2 + \frac{7}{48}\kappa_3^4 - \frac{1}{8}\kappa_3^2\kappa_4 + o(m^{-2}), \tag{2.10}$$

where κ_3 is skewness (or third order cumulant[17]) and κ_4 is kurtosis (or fourth order cumulant) [33], [15], [32].

To derive a multivariate estimate of MI via an Edgeworth expansion, we need to account for all possible dependencies between the variables. The generalization simplifies for the joint PDF of the ICs, because, by definition, they are mutually independent of one another [16]. Furthermore, if the variables that make up the components are whitened beforehand (as outlined in Section 2.2.2), the multivariate approximation of MI will involve only marginal cumulants [15], [16], as shown in Equation 2.11:

$$I(p_{\mathbf{S}}) \approx \sum_{i=1}^{K} \left[\frac{1}{12}\kappa_{iii}^2 + \frac{1}{48}\kappa_{iiii}^2 + \frac{7}{48}\kappa_{iii}^4 - \frac{1}{8}\kappa_{iii}^2\kappa_{iiii} \right]. \tag{2.11}$$

The subscripts emphasize the fact that only diagonal elements of the joint cumulant array matter, as the sources are independent at the optimal solution [15], [16].

Unfortunately, there are drawbacks to approximating MI by an Edgeworth expansion as well. First, the approximation is valid only when the distribution of the original data set is close to normal [32]. Second, there has been no research refuting or proving the claim that Equation 2.11 is a contrast function [15]. As such, we cannot proceed to optimize it, since convergence is no longer guaranteed [15]. In the next section we present a simpler approach, and one that is also a contrast function, to approximate the independence of the components via multivariate cumulants.

[17] Please see Appendix B for an overview of cumulants.

2.2.2.3 Contrast Based on Multivariate Marginal Cumulants

An alternative approach to quantifying independence of components is via multivariate cumulants. Close inspection of Equation 2.11 suggests that if the fourth order marginal cumulants are significantly smaller than those of the third order, we can instead optimize over the sum of the third order marginal cumulants only [15]. Equation 2.12 shows one formulation of the loss function for evaluating independence of components [20]:

$$\sum_{i=1}^{K}\sum_{j=1}^{K}\sum_{k=1}^{K}\left(c_{ijk}-\sum_{i=p}^{L}\gamma_p a_{ip}a_{jp}a_{kp}\right)^2, \tag{2.12}$$

where c_{ijk} denotes the elements of the third order cumulant array for a standardized data set $\tilde{\mathbf{X}}$ with K variables; L is the number of components we wish to extract; γ_i is the theoretical univariate skewness of each independent component; and $\mathbf{A}$ is the loading matrix that is equivalent for each mode of $\mathcal{C}$, as cumulant arrays are symmetric.[18]

Similarly, if the third order marginal cumulants are significantly smaller than those of the fourth order, we can diagonalize the sum of the fourth order marginal cumulants instead [15]. Equation 2.13 shows a generalization of Equation 2.12 for finding independent components in this case [20]:

$$\sum_{i=1}^{K}\sum_{j=1}^{K}\sum_{k=1}^{K}\sum_{l=1}^{K}\left(c_{ijkl}-\sum_{i=p}^{L}\kappa_p a_{ip}a_{jp}a_{kp}a_{lp}\right)^2, \tag{2.13}$$

where c_{ijkl} are the elements of the fourth order cumulant array for a standardized data set $\tilde{\mathbf{X}}$ with K variables; L is the number of components we wish to extract; κ_i is the theoretical univariate kurtosis for each independent component; and $\mathbf{A}$ is the loading matrix that is equivalent in each mode of $\mathcal{C}$.

[18] Please see Appendix B for more details.

As mentioned in Section 2.2.2.1, the main drawback of optimizing over cumulants is their sensitivity to outliers [33]. Their main advantage is that Comon in [15] proved that Equations 2.12 and 2.13, which find the marginal cumulants, are in fact contrast functions, if no more than one independent component has a standard cumulant that vanishes for order higher than two. We return to these contrasts in Chapter 3 when we discuss how we find independent components for a data set of three or more modes via Array Independent Component Analysis.

2.3 Three Mode Dimension Reduction Techniques

While data cubes can be analyzed by two mode dimension reduction techniques, as we saw in previous sections, they do not take into account the relationships between all three modes and tend to overfit [37]. As a result, three-mode dimension reduction technique applied to a three-mode data set will be more robust and interpretable at the expense of a better fit [63], [37].

To reduce the volume of a data cube $\mathcal{X}$ to a smaller data cube $\mathcal{Y}$ that is $P \times Q \times R$ ($P < N$, $Q < M$, $R < K$) or $R \times R \times R$ ($R < N$, $R < M$, $R < K$) that also sheds light on the hidden structures of $\mathcal{X}$, we employ three mode dimension reduction techniques Tucker-3 and CP. They are two generalizations of SVD to data sets of three or more modes [41]. Tucker-3 explains a percentage of the total variance in the data set [50]; its solution is unique up to an orthogonal transformation [41]. CP offers a unique solution of the decomposition up to scale and permutation of the indices [41]. The two techniques are also the most commonly used for reducing the volume of three mode data sets [40]. We discuss these techniques in more detail in the next two sections.

2.3.1 Principal Component Analysis for Three Mode Data via Tucker Decomposition

Tucker-3 [69] is one generalization of Principal Component Analysis for data sets consisting of more than two modes[19], where the "3" stands for the number of modes that we want the loading matrices for [42], [27]. Tucker-3 finds components for each mode of the data set that best explain the variability in each mode, without accounting for the variable relationships in all the other modes [41].

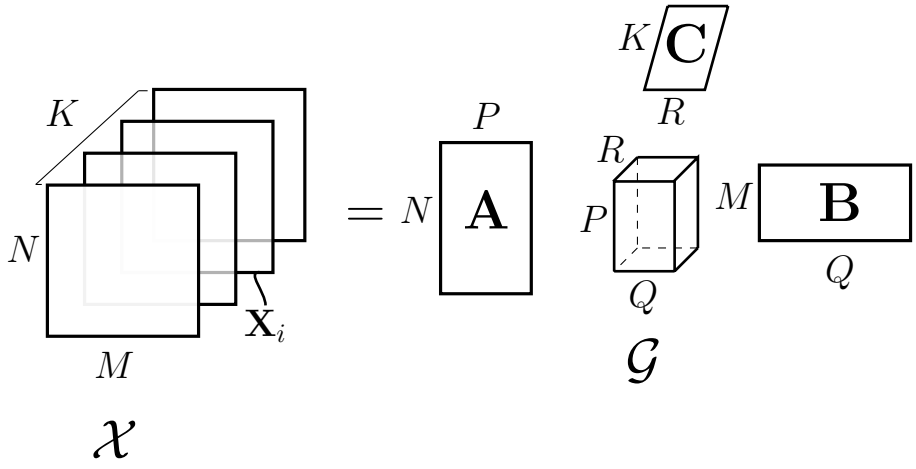


Figure 2.1: Schematic of the Tucker-3 decomposition, where the loading matrices for each mode are stored in $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$; $\mathcal{G}$ is the core array that captures how each side of the data cube $\mathcal{X}$ relates to one another [43].

The decomposition is shown graphically in Figure 2.1, where the loading matrices for each mode are stored in $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$; $\mathcal{G}$ is the core array that captures how each side of the data cube $\mathcal{X}$ relates to one another [43]. As in the previous formulations of dimension reduction techniques, if the aim is also to reduce the volume of $\mathcal{X}_{N\times M\times K}$, we choose a smaller number of components per mode, denoted

[19]Please see [41] for references on extensions of Tucker-3 to higher mode data sets as well as [40], [36].

by P, Q and R, respectively ($P < N$, $Q < M$ and $R < K$).[20] To find them, we minimize the following loss function [43]:

$$\begin{aligned} &\inf_{\mathbf{A},\mathbf{B},\mathbf{C},\mathcal{G}} \sigma(\mathbf{A},\mathbf{B},\mathbf{C},\mathcal{G}) = \\ &\qquad = \sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{k=1}^{K}\left(x_{ijk} - \sum_{p=1}^{P}\sum_{q=1}^{Q}\sum_{r=1}^{R} a_{ip}b_{jq}c_{kr}g_{pqr}\right)^2. \end{aligned} \tag{2.14}$$

As Tucker-3 does not have a closed form solution, re-expressing Equation 2.14 by Equation 2.15 per Appendix C, we get a formulation that is no longer a function of $\mathcal{G}$, and hence, is faster to optimize:

$$\begin{aligned} &\inf_{\mathbf{A},\mathbf{B},\mathbf{C}} \sigma(\mathbf{A},\mathbf{B},\mathbf{C}) = \\ &\qquad = ||\mathcal{X}||^2 - ||\mathcal{X} \times_1 \mathbf{A}^T \times_2 \mathbf{B}^T \times_3 \mathbf{C}^T||^2. \end{aligned} \tag{2.15}$$

where the n-mode product, denoted by $\times_i$ is described in Appendix A.2.5; and the inverse and the transpose of an orthonormal matrix are equivalent. At the infimum, Tucker-3 generates loading matrices for each side of the data cube (stored in matrices $\mathbf{A}_{N\times P}$, $\mathbf{B}_{M\times Q}$ and $\mathbf{C}_{K\times R}$, respectively) as well as a core array $\mathcal{G}_{P\times Q\times R}$.

We now describe one way to optimize Equation 2.14 and perform the decomposition. First preprocess the data prior to analysis. There are many possibilities to do so, such as those described in [43] and [38]. For the remotely sensed data set discussed in Chapter 5, we standardize each slice of the data set $\mathcal{X}$ to get $\tilde{\mathcal{X}}$, and fix the number of components to be extracted per mode.[21] Denote the new column dimensions of $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ by P, Q, and R, respectively ($P < N$, $Q < M$, $R < K$), and initialize the loading matrices $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ [4] as:

- $\mathbf{A}_{I\times P} \longleftarrow P$ left, leading singular vectors of $\tilde{\mathbf{X}}_{(1)}$ associated with the largest singular values, where notation $\tilde{\mathbf{X}}_{(i)}$ denotes matricizing or unfolding of the array $\mathcal{X}$ in the i^{th} mode, as discussed in Appendix A.

[20] Please see Appendix 4 for details on choosing how many components to extract.

[21] Please see Chapter 4 for a discussion on diagnostics for determining how many components to extract for a Tucker-3 decomposition.

- $\mathbf{B}_{J\times Q} \longleftarrow$ first Q left singular vectors of $\tilde{\mathbf{X}}_{(2)}$ associated with the largest singular values;
- $\mathbf{C}_{K\times R} \longleftarrow$ first R left singular vectors of $\tilde{\mathbf{X}}_{(3)}$ associated with the largest singular values,

where $\longleftarrow$ denotes an update to the matrix.

Next, we proceed to find a solution to the loss function in Equation 2.15 via Alternating Least Squares (ALS). The idea behind ALS is to fix all the unknown parameters except for one, update it from the fixed quantities, and cycle through each of the other unknowns, similarly updating each one, one at a time, until convergence [42].[22] One iteration of the ALS approach, derived in Appendix C.4, with three steps for updating each of the three unknowns, one at a time, appears below [4], [41]:

Step 1: Fix $\mathbf{B}^{(t)}$ and $\mathbf{C}^{(t)}$. Then $\mathbf{A}^{(t+1)} \longleftarrow 1^{st}$ leading P eigenvectors of:

$$\tilde{\mathcal{X}} \times_1 \mathbf{I} \times_2 \mathbf{B}^{T(t)} \times_3 \mathbf{C}^{T(t)} = \tilde{\mathcal{X}} \times_2 \mathbf{B}^{T(t)} \times_3 \mathbf{C}^{T(t)}, \tag{2.16}$$

Step 2: Fix $\mathbf{A}^{(t+1)}$ and $\mathbf{C}^{(t)}$. Then $\mathbf{B}^{(t+1)} \longleftarrow 1^{st}$ leading Q eigenvectors of

$$\tilde{\mathcal{X}} \times_1 \mathbf{A}^{T(t+1)} \times_3 \mathbf{C}^{T(t)}. \tag{2.17}$$

Step 3: Fix $\mathbf{A}^{(t+1)}$ and $\mathbf{B}^{(t+1)}$. Then $\mathbf{C}^{(t+1)} \longleftarrow 1^{st}$ leading R eigenvectors of

$$\tilde{\mathcal{X}} \times_1 \mathbf{A}^{T(t+1)} \times_2 \mathbf{B}^{T(t+1)}. \tag{2.18}$$

We iterate over these three steps to find the optimal loading matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$. The algorithm finishes when the array approximation is within the tolerance level:

$$\left| ||\tilde{\mathcal{X}}||^2 - ||\mathcal{G}||^2 \right| < \epsilon.$$

[22] The motivation for the steps of the Tucker-3 ALS algorithm are derived in Appendix C.4.

(We use $\epsilon = 10^{-6}$.)

To check for global convergence and to evaluate the quality of the solution, authors in [27] recommend to run the algorithm with a different set of initial conditions (for instance with random starts). At convergence, compute the core array $\mathcal{G}$ as

$$\mathcal{G} \triangleq \tilde{\mathcal{X}} \times_1 \mathbf{A}^{T(t+1)} \times_2 \mathbf{B}^{T(t+1)} \times_3 \mathbf{C}^{T(t+1)}, \tag{2.19}$$

which is also our desired data cube $\mathcal{Y}$ that reduces the volume of $\mathcal{X}$.[23]

To evaluate how well the method approximates the preprocessed data set, we find the residuals via Equation 2.20:

$$\hat{\Theta} \triangleq \tilde{\mathcal{X}} - \tilde{\mathcal{X}} \times_1 \mathbf{A}^{T(t+1)} \times_2 \mathbf{B}^{T(t+1)} \times_3 \mathbf{C}^{T(t+1)}, \tag{2.20}$$

which we reshape to be an array with three modes. By plotting the residuals for each slice in the array, we can determine if patterns are present; if that is the case, the algorithm does not account for all of the information in the data set. We can try to correct for this by extracting a different number of components (as discussed in Chapter 4) or choosing a different decomposition technique.

One drawback of Tucker-3, however, is its lack of a unique solution without further constraints [41], which is not the case for the Canonical Decomposition discussed in the next section.

[23] To find uncorrelated components for a three mode data set, the technique can be applied to virtually any three mode data set where the slices are of equivalent dimensions. For instance, authors of Principal Component Cumulant Analysis [57] and Moment Component Analysis [35] employ this framework to analyze higher order cumulants and moments, respectively, of a two mode data set of stocks. Here we do not restrict $\mathcal{X}$ to be symmetric.

2.3.2 Principal Component Analysis for Three Mode Data via CANDECOMP/PAFARAC

The Tucker-3 decomposition allowed us to reduce the volume of the data cube $\mathcal{X}_{N\times M\times K}$ to $\mathcal{Y}$ that is $R \times R \times R$ by setting $P = R$ and $Q = R$. However, if the researcher's main objective is to address the uniqueness of the solution and to overcome its indeterminacy, we recommend a technique called Canonical Decomposition [9] or Parallel Factors Analysis [30], which we abbreviate as CP (following the convention of [41]). The solution of CP is unique up to scale of the components [41].

The decomposition is shown graphically in Figure 2.2, where the loading matrices are stored in $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ [43]. The diagram shows that here, each component is linked exclusively with one component from the remaining modes [43]. Its formulation is equivalent to expressing our data set as a sum of R ($R < N$, $R < M$, $R < K$) rank-one arrays [41]. As such, it is also a generalization of SVD to higher mode data sets, and a special case of Tucker-3, where P, Q and R are equal and the core array $\mathcal{G}$ is diagonal [41].

Like Tucker-3, the CP decomposition does not have a closed form solution. As such, to find the unknown loading matrices, we minimize the following loss function:

$$\inf_{\mathbf{A},\mathbf{B},\mathbf{C},\mathbf{\Lambda}} \sigma(\mathbf{A},\mathbf{B},\mathbf{C},\mathbf{\Lambda}) = \sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{k=1}^{K}\left(x_{ijk} - \sum_{p=1}^{R}\lambda_i a_{ip} b_{jp} c_{kp}\right)^2. \tag{2.21}$$

At the infimum, CP generates the loading matrices for each side of the data cube (stored in matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$, respectively), where each column is normalized to unit length, and λ_i, $i = 1, \ldots, R$ absorbs the column lengths (as in the case of PCA, shown in Equation 2.2.) [41]. A more common formulation is to further

absorb λ's into one of the loading matrices, as shown in Equation 2.22 [41]:

$$\inf_{\tilde{\mathbf{A}},\mathbf{B},\mathbf{C}} \sigma(\tilde{\mathbf{A}},\mathbf{B},\mathbf{C}) = \sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{k=1}^{K}\left(x_{ijk} - \sum_{p=1}^{R}\tilde{a}_{ip}b_{jp}c_{kp}\right)^2. \tag{2.22}$$

(In the following equations, we drop the tilde for readability.) The steps to optimizing the simplified loss function of Equation 2.22 and performing the decomposition are outlined below.

The first step is to preprocess the data by centering each column of the original data set $\mathcal{X}$ to get $\tilde{\mathcal{X}}$ [8]. We then fix the number of components we wish to extract per mode to R.[24] To finish the preprocessing steps, initialize the loading matrices $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ as follows [4]:

- $\mathbf{A}_{I\times R} \longleftarrow$ first R left singular vectors of $\tilde{\mathbf{X}}_{(1)}$ associated with the largest eigenvalues[25];
- $\mathbf{B}_{J\times R} \longleftarrow$ first R left singular vectors of $\tilde{\mathbf{X}}_{(2)}$ associated with the largest eigenvalues;
- $\mathbf{C}_{K\times R} \longleftarrow$ first R left singular vectors of $\tilde{\mathbf{X}}_{(3)}$ associated with the largest eigenvalues.

We proceed to find a solution to the loss function in Equation 2.22 via iterating over the three steps of the ALS [4] algorithm, derived in Appendix C.5:

Step 1: Fix $\mathbf{B}^{(t)}, \mathbf{C}^{(t)}$. Then:

$$\mathbf{T} \longleftarrow \mathbf{C}^{(t)} \odot \mathbf{B}^{(t)}$$

$$\mathbf{A}^{(t+1)} \longleftarrow \tilde{\mathbf{X}}_{(1)}\mathbf{T}(\mathbf{T}^T\mathbf{T})^{-1}.$$

[24] Please see Chapter 4 for a discussion on diagnostics for determining how many components to extract for a CP decomposition.

[25] Notation $\tilde{\mathbf{X}}_{(1)}$ denotes matricizing or unfolding of the array $\mathcal{X}$ in the first mode, as discussed in Appendix A.

where $\odot$ is the Khatri-Rao product and is described in Appendix A.

Step 2: Fix $\mathbf{A}^{(t+1)}, \mathbf{C}^{(t)}$. Then

$$\mathbf{T} \longleftarrow \mathbf{C}^{(t)} \odot \mathbf{A}^{(t+1)}$$

$$\mathbf{B}^{(t+1)} \longleftarrow \tilde{\mathbf{X}}_{(2)}\mathbf{T}(\mathbf{T}^T\mathbf{T})^{-1}.$$

Step 3: Fix $\mathbf{A}^{(t+1)}, \mathbf{B}^{(t+1)}$. Then

$$\mathbf{T} \longleftarrow \mathbf{B}^{(t+1)} \odot \mathbf{A}^{(t+1)}$$

$$\mathbf{C}^{(t+1)} \longleftarrow \tilde{\mathbf{X}}_{(3)}\mathbf{T}(\mathbf{T}^T\mathbf{T})^{-1}.$$

We iterate over the steps until a desired level of tolerance ϵ is reached:

$$\left|\left|\mathbf{X}_{(1)} - \mathbf{A}^{(t+1)}\left(\mathbf{C}^{(t+1)} \odot \mathbf{B}^{(t+1)}\right)^T\right|\right|^2 < \epsilon. \tag{2.23}$$

(We use $\epsilon = 10^{-6}$.)

To check for global convergence and evaluate the quality of the solution, authors in [27] recommend to run the algorithm with a different set of initial conditions (for instance with random starts). At convergence, we find the residuals via Equation 2.24:

$$\hat{\Theta} \triangleq \tilde{\mathbf{X}}_{(1)} - \mathbf{A}^{(t+1)}\left(\mathbf{C}^{(t+1)} \odot \mathbf{B}^{(t+1)}\right)^T, \tag{2.24}$$

which we reshape to be an array with three modes. By plotting the residuals for each slice of the array, we can determine if patterns are present; if that is the case, the algorithm does not account for all of the information in the data set. In that case, we can include a different number of components (as discussed in Chapter 4) or conclude that the CP decomposition is not appropriate for this data set and use another approach.

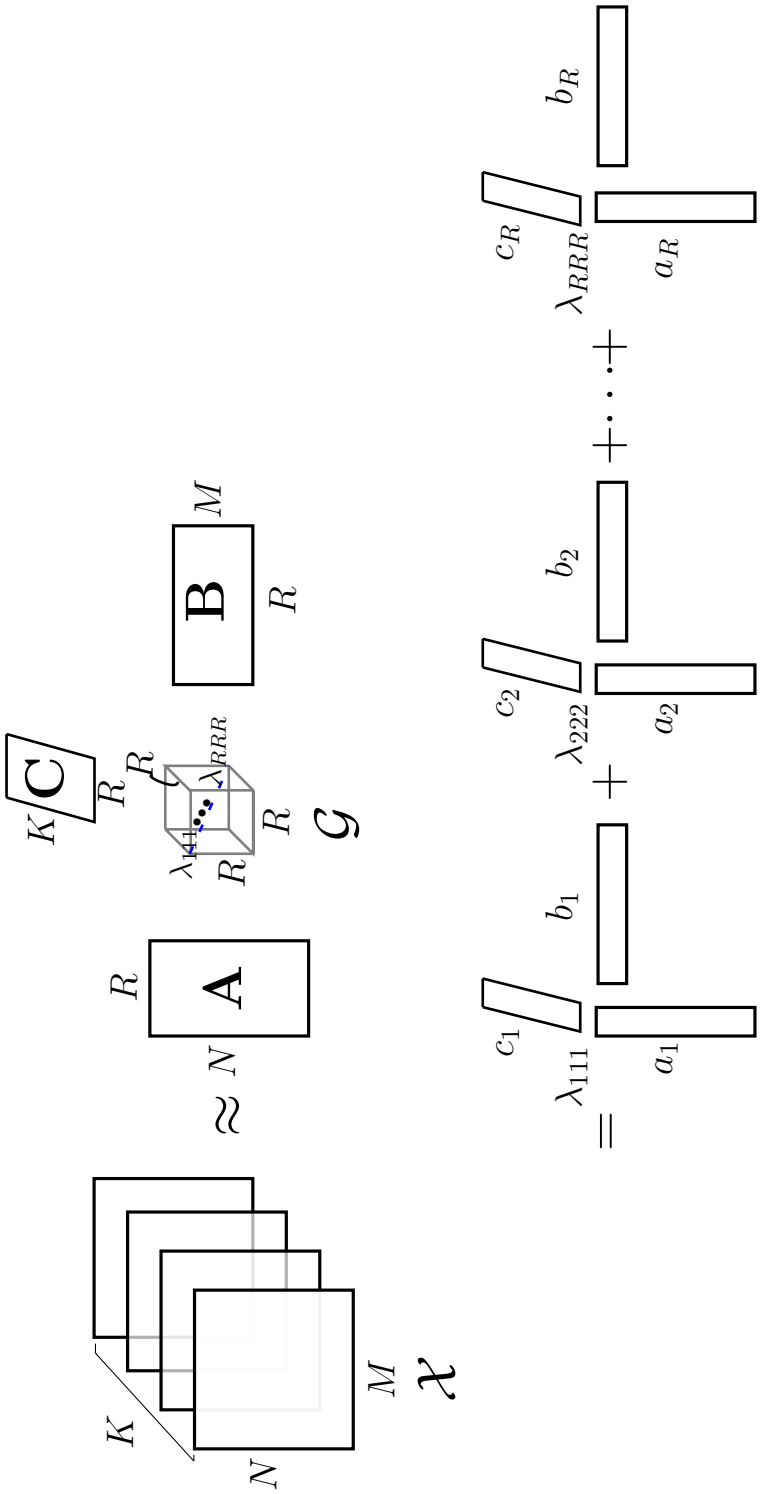


Figure 2.2: Schematic of the CP decomposition extracting R ($R < N$, $R < M$, $R < K$) components per mode, where each component is linked exclusively with one component of the remaining modes [43].

After performing the decomposition, we find the data cube $\mathcal{Y}$ that is smaller in volume than the original data cube $\mathcal{X}$, also via Equation 2.19[26]:

$$\mathcal{Y} = \tilde{\mathcal{X}} \times_1 \mathbf{A}^{T(t+1)} \times_2 \mathbf{B}^{T(t+1)} \times_3 \mathbf{C}^{T(t+1)}.$$

2.4 Discussion

In this chapter, we introduced the most commonly used decompositions for reducing the volume of a data cube $\mathcal{X}_{N \times M \times K}$ to obtain a data cube $\mathcal{Y}$ that was smaller in at least one mode. We discussed scenarios where $\mathcal{Y}$ was $N \times M \times L$ ($L < K$) via PCA or ICA, if we extracted L components. If the data set was jointly multivariate normal, the Principal Components would be independent and uncorrelated otherwise. In the latter case, we recommended ICA, a generalization of PCA, for finding independent components. For three mode data sets, we presented generalizations of PCA: Tucker-3 and CP, which obtained a data cube $\mathcal{Y}$ that is $P \times Q \times R$ ($P < N$, $Q < M$, $R < K$) and $R \times R \times R$ ($R < N$, $R < M$, $R < K$), respectively. We build on these previously developed methods to extend the framework of ICA to find latent, independent components for one mode of a three (or higher) mode non-normally distributed data set, and reduce its volume. We call this approach Array Independent Component Analysis (AICA). While the AICA framework quantifies the variable independence via cumulants similar to ICA, it builds on algorithms of Tucker and CP to find the optimal loading matrices.

[26]Note, that if the λ's are absorbed into the estimation of the loading matrices in the CP decomposition, $\mathcal{Y}$ will be an identity matrix $\mathbf{I}_R$.

CHAPTER 3

Array Independent Component Analysis

3.1 Introduction

In the previous chapter we reduced the volume of the data cube $\mathcal{X}$ and found uncorrelated latent variables from its two and three mode representations, and its independent components from the two mode representation only. There has been no analog for finding independent components via ICA for three (or higher) mode data sets until now. We call this generalization of ICA – Array Independent Component Analysis (AICA).

The AICA framework allows us to find independent components for one mode of the data cube $\mathcal{X}$ while accounting for its relationship with the other modes. To do so, similar to its two mode counterpart, AICA leverages the non-normality of the distribution of each slice of the data set. In the process, it can also reduce the volume of $\mathcal{X}_{N\times M\times K}$ to get a smaller data cube $\mathcal{Y}$ that is $N \times P \times P$ ($P < M$ and $P < K$).[1]

There are many different ways to quantify variable independence, as we saw in the previous chapter. In the case of AICA, we focus on quantifying independence for a group of variables via joint cumulants [31].[2,3] Appendix B emphasizes that

[1] By the nature of the framework, it is not possible to reduce all three modes of the data set $\mathcal{X}$. However, since the order of the modes is arbitrary, this is only a weak constraint.

[2] Please see Appendix B for a discussion on cumulants and their properties.

[3] If we know *a priori* that some variables are independent of the others in a given mode, the AICA framework is flexible to allow dimension reduction for a partition of a disjoint set of variables of $\mathcal{X}$, analyzing each disjoint data cube separately and pooling the results.

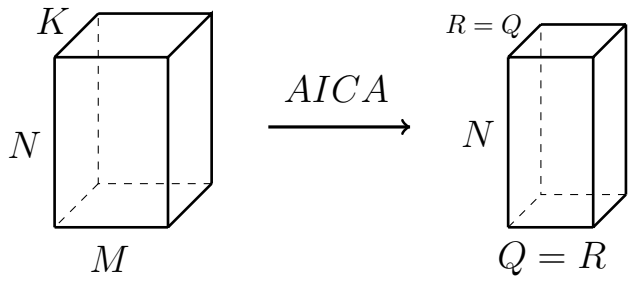


Figure 3.1: Illustration of volume reduction for a data cube as a result of AICA.

the order of the cumulant dictates how many modes it has. As a result, we work with third order cumulants (of standardized variables) in the AICA framework for ease of model derivation, computational efficiency and interpretation.

In Section 2.2.2 we learned that there are two possible contrast functions to consider for diagonalizing third or fourth order joint cumulants. Because in the application of the thesis (discussed in the following chapter), the slices of the remotely sensed data are not symmetric, and hence the joint third order cumulants do not vanish, we proceed to introduce AICA as a technique to decompose third order cumulants for each slice of the data set.[4]

For the remainder of the chapter, we familiarize the reader with AICA. First we discuss how to preprocess the original data set with three (or more) modes and find the cumulants for each data slice. We then introduce the loss function we need to optimize in order to find the loading matrices that will rotate the variables in one of the modes of the original data set towards independence. In closing, we present the algorithm we developed to find the solution of the optimization problem. The next chapter focuses on the application of this new methodology in the context

[4]In the case of symmetric data sets (where third orders vanish and fourth do not), the framework developed can easily be generalized to decomposing fourth order cumulants. In that case, we need to add only one extra step to the Alternating Least Squares algorithm to find and update the loading matrix for the additional mode.

of remote sensing.

3.2 Methodology

3.2.1 Overview

AICA is a generalization of ICA to three (or higher) mode data sets. To begin the decomposition, we standardize each $N \times M$ slice of the original three mode data set $\mathcal{X}_{N \times M \times K}$ and find its joint third order cumulant (for M column variables this results in an $M \times M \times M$ data cube); we do that for all K slices. We then combine all K third order cumulants into one array $\mathcal{A}$, which will now be $M \times M \times M \times K$, as shown in Figure 3.2. The goal is to decompose $\mathcal{A}$ to find loading matrices for each side of $\mathcal{A}$ that will rotate the third order cumulants to be diagonal. If the cumulants are diagonal, this means that each rotated variable is independent of the others. These rotated variables are our independent components. Three of the matrices responsible for the rotation are equal because cumulants are symmetric. Figure 3.3 highlights these steps of AICA graphically. Since AICA does not have a closed form solution, the remainder of this chapter discusses the details for finding the independent components by diagonalizing array $\mathcal{A}$, namely, the optimization problem and the algorithm.

3.2.2 Formulation of Loss Function

To derive the loss function of AICA, we begin by introducing the most general and unconstrained loss function for decomposing and reducing the volume of a (general) four mode array $\mathcal{Z}$. This is essentially the Tucker-4 model, where we reduce the dimensionality of an $N \times M \times K \times L$ array $\mathcal{Z}$ to one that is $O \times P \times Q \times R$ ($O < N$, $P < M$, $Q < K$ and $R < L$). In the process, we find matrices $\mathbf{U}$, $\mathbf{V}$, $\mathbf{W}$, $\mathbf{H}$ and a four mode core array $\mathcal{G}$, where:

- $\mathbf{U}$ is an $N \times O$ matrix, where O is the number of components to be extracted

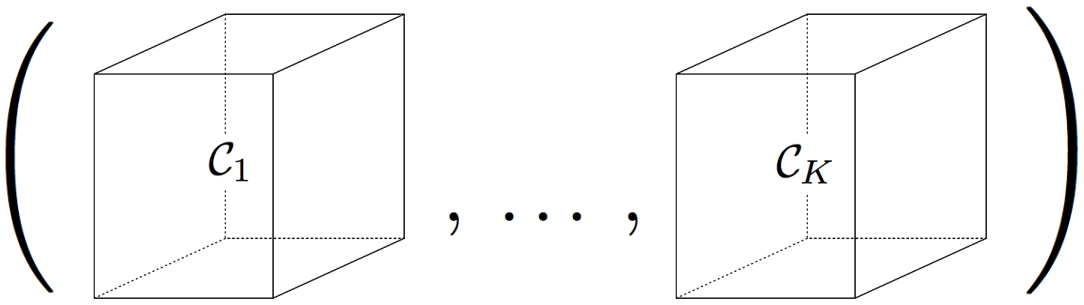


Figure 3.2: Schematic of a four mode $\mathcal{A}_{M\times M\times M\times K}$ that is composed of third order cumulants, denoted by $\mathcal{C}_i$ for each i^{th} $(i = 1, \ldots, K)$ slice of the standardized data set $\mathcal{X}$.

from $\mathbf{Z}_{(1)}$;

- $\mathbf{V}$ is an $M \times P$ matrix, where P is the number of components to be extracted from $\mathbf{Z}_{(2)}$;
- $\mathbf{W}$ is a $K \times Q$ matrix, where Q is the number of components to be extracted from $\mathbf{Z}_{(3)}$;
- $\mathbf{H}$ is an $L \times R$ matrix, where R is the number of components to be extracted from $\mathbf{Z}_{(4)}$;
- $\mathcal{G}$ is a core array of dimensions $O \times P \times Q \times R$ that links the components from each mode [43].

We find the unknown core array and loading matrices by optimizing the loss function in Equation 3.1:

$$\inf_{\mathbf{U},\mathbf{V},\mathbf{W},\mathbf{H},\mathcal{G}} \sigma(\mathbf{U},\mathbf{V},\mathbf{W},\mathbf{H},\mathcal{G}) = \\ = \sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{k=1}^{K}\sum_{l=1}^{L}\left(a_{ijkl} - \sum_{o=1}^{O}\sum_{p=1}^{P}\sum_{q=1}^{Q}\sum_{r=1}^{R} u_{io}v_{jp}w_{kq}h_{lr}g_{opqr}\right)^2. \quad (3.1)$$

Because one of the goals of AICA is to find a relatively small number of independent components from the original data by diagonalizing array $\mathcal{A}$ containing third

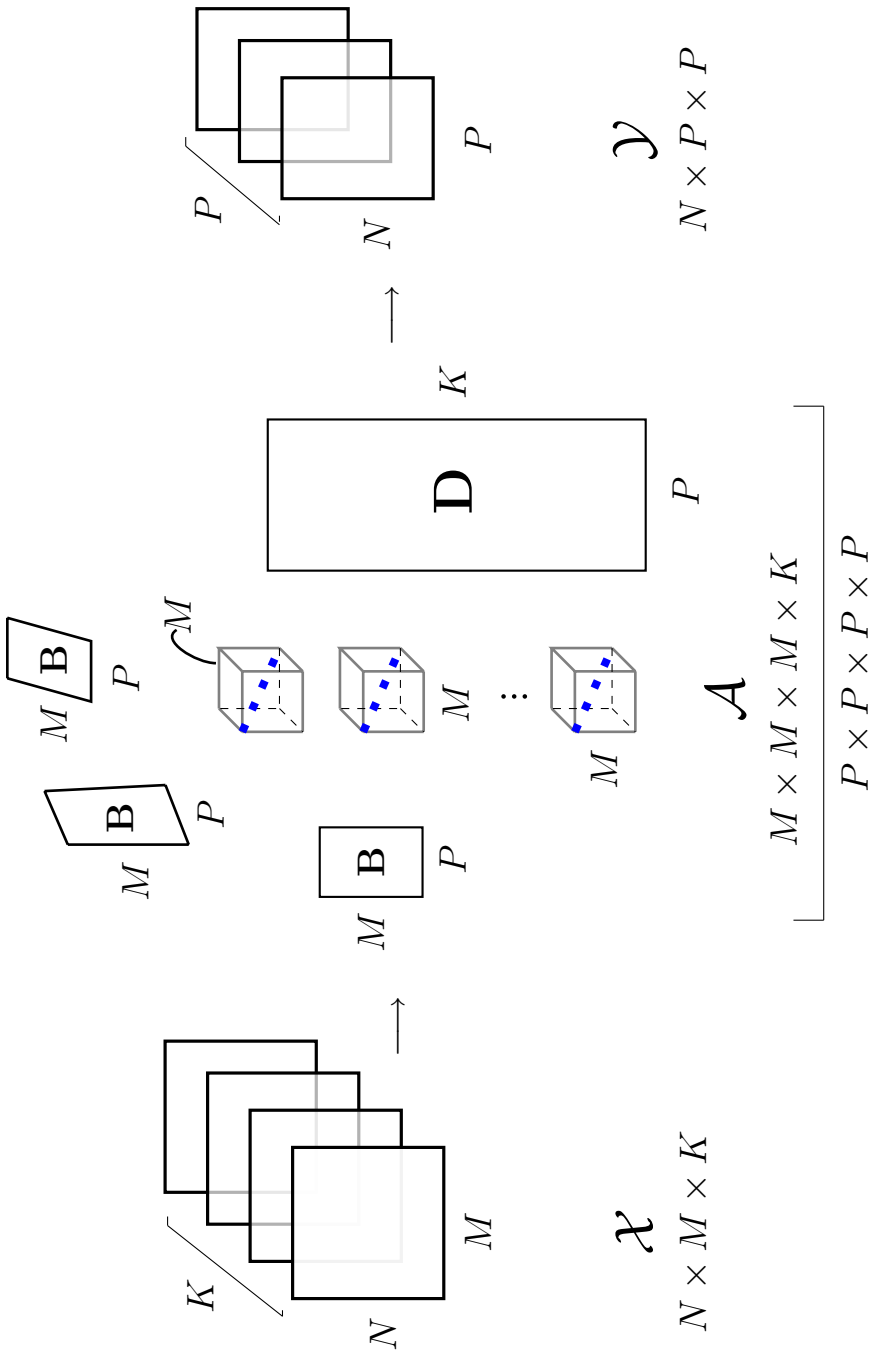


Figure 3.3: Schematic of AICA, where $\mathcal{X}$ is the preprocessed three mode data set; $\mathcal{A}$ is the core array that stores the joint third order cumulants for each $N \times M$ slice of $\mathcal{X}$; **B** and **D** are the loading matrices (accounting for symmetry in first three modes) that rotate $\mathcal{A}$ to be diagonal; and $\mathcal{Y}$ is the smaller data cube, resulting from the analysis, with $P < M$ and $P < K$.

order cumulants, which is symmetric in three modes (by properties of cumulants discussed in Appendix B), we need to modify the loss function in Equation 3.1 to include this constraint.

Suppose we want to reduce the volume of the core array $\mathcal{A}$ such that it is also diagonal. We restrict ourselves to a special case of Equation 3.1, where the reduced sizes of its modes are smaller and equal; we denote the new dimension of the modes by P ($P < M$ and $P < K$). The loss function that accounts for these constraints is presented in Equation 3.2:

$$\inf_{\mathbf{B},\mathbf{D}} \sigma(\mathbf{B},\mathbf{D}) = \sum_{i=1}^{M}\sum_{j=1}^{M}\sum_{k=1}^{M}\sum_{l=1}^{K}\left(a_{ijkl} - \sum_{p=1}^{P} b_{ip}b_{jp}b_{kp}d_{lp}\right)^2, \tag{3.2}$$

where

- $\mathbf{B}$ is an $M \times P$ matrix ($P < M$) that rotates the variables in mode 2 of $\mathcal{X}$ to obtain P independent components;
- $\mathbf{D}$ is a $K \times P$ matrix ($P < K$), which rotates the variables in the third mode of $\mathcal{X}$ to obtain P uncorrelated components;
- the core array $\mathcal{G}$ is diagonal and absorbed into $\mathbf{D}$, as in the CP decomposition [41].

From the decomposition, we find the smaller data cube $\mathcal{Y}$ by

$$\mathcal{Y} = \mathcal{X} \times_2 \mathbf{B}^{T(t+1)} \times_3 \mathbf{D}^{T(t+1)},$$

where the second mode of $\mathcal{Y}$ contains the independent components.

3.2.3 Optimization of Array Independent Component Analysis

We now proceed to discuss how to minimize the loss function introduced in the previous section. While we expect the loading matrices for the first three modes of $\mathcal{A}$ to be equivalent at the optimum, as array $\mathcal{A}$ is symmetric in three modes,

this is not guaranteed. To account for this, we denote the three loading matrices with different letters $\mathbf{U}$, $\mathbf{V}$ and $\mathbf{W}$. The initial conditions for the method are as follows[5]:

- $\mathbf{U}_{M\times P} \longleftarrow$ first P left singular vectors of $\mathbf{A}_{(1)}$; [6]
- $\mathbf{V}_{M\times P} \longleftarrow$ first P left singular vectors of $\mathbf{A}_{(2)}$;
- $\mathbf{W}_{M\times P} \longleftarrow$ first P left singular vectors of $\mathbf{A}_{(3)}$;
- $\mathbf{D}_{K\times L} \longleftarrow$ first P left singular vectors of $\mathbf{A}_{(4)}$.

Note that by the symmetry of the third order cumulants, matrices $\mathbf{U}$, $\mathbf{V}$ and $\mathbf{W}$ are initially equal.

To reduce the width and depth of the data cube $\mathcal{X}$ to get $\mathcal{Y}$ whose new width and depth are smaller and equal, we find the infimum of Equation 3.2. To do so, we introduce an ALS algorithm that is based on the CP decomposition for arrays with four modes and derived in Appendix C.6:

Step 1: Fix $\mathbf{V}^{(t)}, \mathbf{W}^{(t)}, \mathbf{D}^{(t)}$. Then

$$\mathbf{T} \longleftarrow \mathbf{D}^{(t)} \odot \mathbf{W}^{(t)} \odot \mathbf{V}^{(t)}$$

$$\mathbf{U}^{(t+1)} \longleftarrow \mathbf{A}_{(1)}\mathbf{T}(\mathbf{T}^T\mathbf{T})^{-1}$$

Step 2: Fix $\mathbf{U}^{(t+1)}, \mathbf{W}^{(t)}, \mathbf{D}^{(t)}$. Then

$$\mathbf{T} \longleftarrow \mathbf{D}^{(t)} \odot \mathbf{W}^{(t)} \odot \mathbf{U}^{(t+1)}$$

$$\mathbf{V}^{(t+1)} \longleftarrow \mathbf{A}_{(2)}\mathbf{T}(\mathbf{T}^T\mathbf{T})^{-1}$$

[5]Please see Chapter 4 to determine how many components to extract.

[6]Notation $\mathbf{A}_{(i)}$ denotes matricizing or unfolding of the array $\mathcal{A}$ in the i^{th} mode, as discussed in Appendix A.2.1.2.

Step 3: Fix $\mathbf{U}^{(t+1)}, \mathbf{V}^{(t+1)}, \mathbf{D}^{(t)}$. Then

$$\mathbf{T} \longleftarrow \mathbf{D}^{(t)} \odot \mathbf{V}^{(t+1)} \odot \mathbf{U}^{(t+1)}$$

$$\mathbf{W}^{(t+1)} \longleftarrow \mathbf{A}_{(3)}\mathbf{T}(\mathbf{T}^T\mathbf{T})^{-1}$$

Step 4: Fix $\mathbf{U}^{(t+1)}, \mathbf{V}^{(t+1)}, \mathbf{W}^{(t+1)}$. Then

$$\mathbf{T} \longleftarrow \mathbf{W}^{(t+1)} \odot \mathbf{V}^{(t+1)} \odot \mathbf{U}^{(t+1)}$$

$$\mathbf{D}^{(t+1)} \longleftarrow \mathbf{A}_{(4)}\mathbf{T}(\mathbf{T}^T\mathbf{T})^{-1}$$

We repeat Steps 1-4 until a desired level of tolerance is reached, per Equation 3.2. There are two scenarios to check for global convergence. Either the first three loading matrices $\mathbf{U}, \mathbf{V}$ and $\mathbf{W}$ are the same, or they are equivalent up to scale. The former is straightforward to check. We can check the latter via Equation 3.3:

$$\mathbf{U} \simeq \mathbf{V} \quad \text{if and only if} \quad \left(\mathbf{U}^T\mathbf{U}\right)^{-1}\left(\mathbf{U}^T\mathbf{V}\right)\left(\mathbf{V}^T\mathbf{V}\right)^{-1}\left(\mathbf{V}^T\mathbf{U}\right) = \mathbf{I}_P \tag{3.3}$$

for $\mathbf{U}$ and $\mathbf{V}$ as shown, and by symmetry for $\mathbf{U}$ and $\mathbf{W}$ and $\mathbf{V}$ and $\mathbf{W}$. If this is not the case, we recommend initializing the algorithm with a different set of initial conditions and proceeding from Step (1).

Alternatively, we can guarantee equivalent loading matrices for the first three modes of $\mathcal{A}$ by adding Step 5 to the ALS algorithm above, where we update $\mathbf{U}$, $\mathbf{V}$ and $\mathbf{W}$ by the element-wise average of the three matrices:

Step 5: $\mathbf{T} \longleftarrow \text{avg}\left(\mathbf{U}^{(t+1)}, \mathbf{V}^{(t+1)}, \mathbf{W}^{(t+1)}\right)$

$$\mathbf{U}^{(t+1)} \longleftarrow \mathbf{T}$$

$$\mathbf{V}^{(t+1)} \longleftarrow \mathbf{T}$$

$$\mathbf{W}^{(t+1)} \longleftarrow \mathbf{T}$$

One caveat with the approach is that monotone convergence is no longer guaranteed.

Even at a global minimum, we need to evaluate model fit. One way to do so is to compute the residuals via Equation 3.4:

$$\hat{\Theta} = \mathbf{A}_{(1)} - \mathbf{U}^{(t+1)} \left(\mathbf{D}^{(t+1)} \odot \mathbf{W}^{(t+1)} \odot \mathbf{V}^{(t+1)} \right)^T \tag{3.4}$$

and reshape the residual matrix to be an array with four modes. By plotting the residuals for each slice of the array, we can determine if patterns are present. If that is the case, the algorithm does not account for all of the information in the data set. We can try to correct for this by choosing a different number of components as outlined in Chapter 4, or another decomposition method. The decomposition is practical if the resulting components are also interpretable.

3.3 Discussion

In this chapter, we introduced and discussed a generalization of ICA to three (or higher) mode data sets called AICA. The technique allows us to reduce the volume of a data cube $\mathcal{X}_{N \times M \times K}$ to obtain a smaller cube $\mathcal{Y}$ that is $N \times P \times P$, and find latent, independent components for the second mode of the data set with slices that are not normally distributed. As there is no closed form solution of the decomposition, we developed a loss function and an optimization scheme to find the loading matrices, to do so, once the number of components are fixed at P. The next chapter discusses a heuristic for determining how to best select the sizes of the modes for $\mathcal{Y}$ to reduce each mode of two to four mode arrays ($\mathcal{X}$ or $\mathcal{A}$) via the decompositions presented in the thesis.

CHAPTER 4

Determining the Number of Components to Extract

4.1 Introduction

One of the key questions of any dimension reduction technique is how many components to extract, as we want to reduce the dimensionality of the data set while explaining most of its variability with as few components as possible. In this chapter, we discuss how to find the optimal tradeoff between fit and model complexity in decompositions of two to four modes. We do so via the first programable diagnostic originally developed by Ceuleimans and Kiers in [10], and its extension to the case of AICA.

4.2 Methodology

The main idea behind the heuristic is that it finds the optimal solution on the convex hull of all possible solutions, which allows the researcher to compare model fit across decompositions with different numbers of components [10]. To find the hull, we specify the complexity of the model either as the number of free parameters in the model or the total number of components [10]. For preiously developed dimension reduction techniques, we are approximating $\mathcal{X}_{N\times M\times K}$. In the case of AICA, we approximate array $\mathcal{A}$ containing third order cumulants for the variables in each $N \times M$ slice of $\mathcal{X}$. We evaluate fit as a percentage of sum of squares of $\mathcal{X}$ or $\mathcal{A}$ (depending on the decomposition) that we are able to explain.

The details for quantifying model complexity and fit follow.

4.2.1 Model Complexity

First we discuss how to evaluate model complexity for three mode data sets. Let $\mathcal{X}$ be the data cube that we wish to reduce the volume of, to get a smaller data cube $\mathcal{Y}$.

4.2.1.1 Three Mode Asymmetric Decompositions

There are three possible scenarios for the dimensions of the modes of $\mathcal{Y}$ that will reduce the volume of $\mathcal{X}_{N\times M\times K}$ via previously developed dimension reduction techniques:

- $N \times M \times L$, with $L < K$ via PCA and ICA;
- $P \times Q \times R$, with $P < N$, $Q < M$ and $R < K$ via Tucker-3;
- $R \times R \times R$, with $R < N$, $R < M$ and $R < K$ via CP.

There are two ways to quantify model complexity; either as the total number of components for all modes of the data set we wish to extract, or as the number of free parameters in the model. The former is straightforward to calculate. To find the latter, we use Equation 4.1 [43], [10], [39]:

$$\begin{aligned}\text{number of free parameters} = {} & \text{number of observations in loading matrices}\\ & + \text{number of observations in core array}\\ & - \text{rotational indeterminacy of orthogonal loading matrices}\\ & \qquad\qquad\qquad\qquad\text{(in Tucker models)}\\ & - \text{scale indeterminacy of loading matrices (in CP models)},\end{aligned} \tag{4.1}$$

where the indeterminacy of the Tucker models comes from the loading matrices that are unique up to an orthogonal transformation [43], [41]. A non-singular

orthogonal transformation of order O has $O(O-1)$ parameters, as the last column of the matrix is determined from the other $(O-1)$ columns [18]. Table 4.1 shows us how to quantify model complexity for both metrics for three mode arrays.[1]

Decomposition	Total Number of Components	Free Parameters
PCA	L	$KL + NML - L(L-1)$
ICA	L	—
Tucker-3	$P + Q + R$	$NP + MQ + KR + PQR$ $-P(P-1) - Q(Q-1)$ $-R(R-1)$
CP	$R + R + R$	$(N + M + K)R + R - 3R$

Table 4.1: Quantification of model complexity for previously developed dimension reduction techniques for three mode data arrays.

Simulations performed by authors in [10] showed that if model complexity for Tucker-3 is calculated as the total number of components, the diagnostic was able to identify the correct model more often. As a result and for simplicity, we use this quantification to access model complexity for the methodologies discussed in the thesis when we apply them to analyze a remote sensing data set in the following chapter.

4.2.1.2 Four Mode Asymmetric Decompositions

The generalization of the framework developed in the previous section is straightforward for four mode asymmetric decompositions. Let $\mathcal{X}$ be $N \times M \times K \times L$. There are two possible scenarios for the dimensions of the modes of $\mathcal{Y}$ that will reduce all four modes of $\mathcal{X}$ via previously developed dimension reduction tech-

[1] In the case of ICA, the number of free parameters depends on the optimization method. It is beyond the scope of the thesis to list all possible quantifications of free parameters for a two mode decomposition that we do not recommend using, for finding independent components of a three mode data array.

niques:

- $O \times P \times Q \times R$, with $O < N$, $P < M$ $Q < K$ and $R < L$ via Tucker-4;
- $P \times P \times P \times P$, with $P < N$, $P < M$, $P < K$ and $P < L$ via CP.

The summary of quantification of complexity for the decompositions is shown in Table 4.2.

Decomposition	Total Number of Components	Free Parameters
Tucker-4	$O + P + Q + R$	$NO + MP + KQ + LR$ $+OPQR - O(O-1)$ $-P(P-1) - Q(Q-1)$ $-R(R-1)$
CP	$P + P + P + P$	$(N+M+K+L)P + P^2 - 4P$

Table 4.2: Quantification of model complexity for previously developed dimension reduction techniques for four mode asymmetric data arrays.

4.2.1.3 AICA Decomposition

We extend the framework developed in the previous section to allow for symmetry in some modes of the array, to be consistent with the AICA model. In the case of AICA, we reduce an array $\mathcal{X}$ that is $M \times M \times M \times K$ to one that is $P \times P \times P \times P$, with $P < M$ and $P < K$. Following the conventions described in the previous section, model complexity for AICA is shown in Table 4.3.

Decomposition	Total Number of Components	Free Parameters
AICA	$P + P + P + P$	$(M + K)P + P^2 - 2P$

Table 4.3: Quantification of model complexity for AICA.

4.2.2 Model Fit

There are numerous ways to evaluate model fit. One way is via residual analysis for each slice of the data cube. If there are many slices or modes, the diagnostic proves cumbersome. Another approach is cross validation, which is computationally intensive [7], especially when combined with the slow convergence of the ALS algorithm.

For practical purposes, we evaluate fit by comparing how well a given model is able to explain the sum of squares in the preprocessed data set, per Equation 4.2:

$$100 \times \frac{\sum_{ij\ldots k} \hat{z}_{ij\ldots k}^2}{\sum_{ij\ldots k} z_{ij\ldots k}^2}, \tag{4.2}$$

where $\hat{z}_{ij\ldots k}$ is the approximation to the data set $\mathbf{Z}$ (which is either a preprocessed array $\mathcal{X}$ or $\mathcal{A}$, depending on the decomposition) we can derive via each decomposition discussed in the thesis. Table 4.4 summarizes how to do so in each case; the matrix notation follows definitions of Chapters 2 and 3.

Number of Modes (D)	Model	$\mathcal{Z}$	$\hat{\mathcal{Z}}$
$D \geq 2$	PCA	$\mathbf{Z}$	$\mathbf{Z}\mathbf{U}_0\mathbf{U}_0^T$
	ICA	$\mathbf{Z}_D$	$\mathbf{S}\mathbf{A}_0$
3	Tucker	$\tilde{\mathcal{X}}$	$\mathbf{A}\mathbf{G}_{(1)}\left(\mathbf{C} \otimes \mathbf{B}\right)^T$
	CP	$\tilde{\mathcal{X}}$	$\mathbf{A}\left(\mathbf{C} \odot \mathbf{B}\right)^T$
4	AICA	$\mathcal{A}$	$\mathbf{U}\left(\mathbf{D} \odot \mathbf{W} \odot \mathbf{V}\right)^T$

Table 4.4: Quantification of data approximations for dimension reduction techniques discussed in the thesis; the matrices are defined in Chapters 2 and 3.

4.2.3 Approximate Model Fit for Tucker and CP Models

The ALS algorithm for the Tucker and CP models is iterative, with potentially slow convergence. Without an approximation to the decomposition, to find the

convex hull among the models with different complexities, we would have to fit each of the models. In the next two sections we present previously developed work that addresses this difficulty by approximating the model fit by a deterministic approach.

4.2.3.1 Approximate Model Fit for Tucker Model

Authors in [39] and [10] suggest a deterministic approximation for the Tucker-3 decomposition, based on the original formulation by Tucker [69] (as referenced in [39]), where all of the model approximations are calculated in one go. The approach finds the loading matrices for each mode of the array by performing PCA on the data array unfolded in each of its modes [39], [10]. We outline the steps for the preprocessed three mode array $\mathcal{X}$ below.

Step 1: Unfold the data array in mode 1 to get $\mathbf{X}_{N\times MK}$. Decompose it via SVD and extract P components that are associated with the largest eigenvalues. Denote the resulting loading matrix by $\mathbf{A}$ [39], [10].

Step 2: Unfold the data array in mode 2 to get $\mathbf{X}_{M\times NK}$. Decompose it via SVD and extract Q components that are associated with the largest eigenvalues. Denote the resulting loading matrix by $\mathbf{B}$ [39], [10].

Step 3: Unfold the data array in mode 3 to get $\mathbf{X}_{K\times NM}$. Decompose it via SVD and extract R components that are associated with the largest eigenvalues. Denote the resulting loading matrix by $\mathbf{C}$ [39], [10].

Step 4: Find $\mathcal{G}$ as $\mathbf{G}_{(1)} = \mathbf{A}^T\mathbf{X}_{(1)}\,(\mathbf{C}\otimes\mathbf{B})$ [39], [10].

We estimate model fit by $\hat{\mathcal{X}} = \mathcal{G}\times_1\mathbf{A}\times_2\mathbf{B}\times_3\mathbf{C}$ [39], [10]. Notice that if we set P, Q and R to the largest desired number of components per mode, we need to perform SVD for each mode only once, and subset the loading matrices to calculate model fit with a smaller number of extracted components [39], [10].

4.2.3.2 Approximate Model Fit for CP Model

We can also use a modification of the Tucker-3 deterministic framework presented above, called CORCONDIA, to approximate the CP model fit [7]. The idea behind the approach is that it evaluates how different the core array $\mathcal{G}$ from the Tucker framework is, when compared to an array with ones on the diagonal (denoted by $\mathcal{D}$) in the case of CP [7]. It can be found via Equation 4.3 [7]:

$$100 \times \frac{\sum_{i=1}^{R}\sum_{j=1}^{R}\sum_{k=1}^{R}(g_{ijk} - d_{ijk})^2}{\sum_{i=1}^{R}\sum_{j=1}^{R}\sum_{k=1}^{R} d_{ijk}^2}. \tag{4.3}$$

From the equation, we can see that if the CP model is the true one, the diagnostic can be close to or equal to one [7]. To find $\mathcal{G}$, we fit a Tucker-3 model extracting the same number of components for all three modes.

4.2.4 Finding Solution on the Convex Hull

To find a solution on the convex hull, we first have to find the hull. Following the convention of [10], let fp denote model complexity of the decompositions either as the number of free parameters or the total number of components, as defined in Section 4.2.1 [10]. Let f denote model fit, as defined in Section 4.2.2 [10].

The steps to finding the solution on the convex hull are as follows. First, determine the largest number of components we wish to extract per mode (denoted by P, Q, R for three mode data sets, and O, P, Q, R for four mode data arrays) [10]. Calculate model complexity fp and its associated model fit f for the decompositions, as shown in Figure 4.1(a) [10]. For each level of fp, choose only the best solution, which is the one with the best fit, as shown in Figure 4.1(b) [10]. Order solutions from smallest to largest number of fp (such as when f is plotted against fp) [10]. For all pairwise relationships between fp and f, exclude suboptimal solutions [10]. That is, exclude those models with more free parameters (or components) but worse fit as shown in Figure 4.1(c) [10]. From the remaining pairwise

relationships between fp and f, analyze all triplets [10]. Exclude the point in the middle of the triplet if it falls on or below the line that could be drawn between the other two points, up to a tolerance level λ as shown in Figure 4.1(d) [10]. (We used $\lambda = 0.5$.)

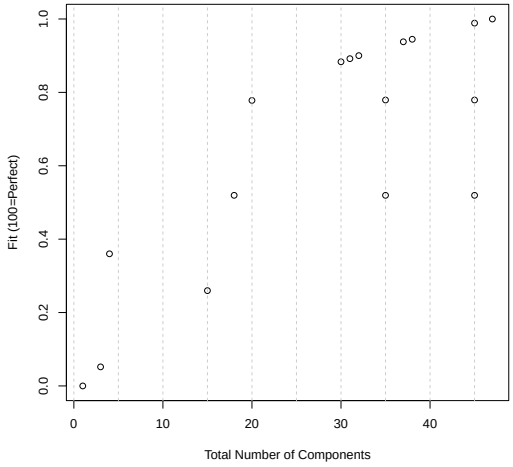
When we can no longer delete any points, the result is the convex hull of solutions [10], as shown in blue in Figure 4.1(e). To choose the solution that best balances fit and complexity and is also on the hull, authors in [10] have devised a scree test, denoted by st, to find it. We calculate it for all of the remaining points, excluding the first and last point due to boundary conditions, via Equation 4.4 [10]:

$$st_i = \frac{f_i - f_{i-1}}{fp_i - fp_{i-1}} \Big/ \frac{f_{i+1} - f_i}{fp_{i+1} - fp_i}. \tag{4.4}$$

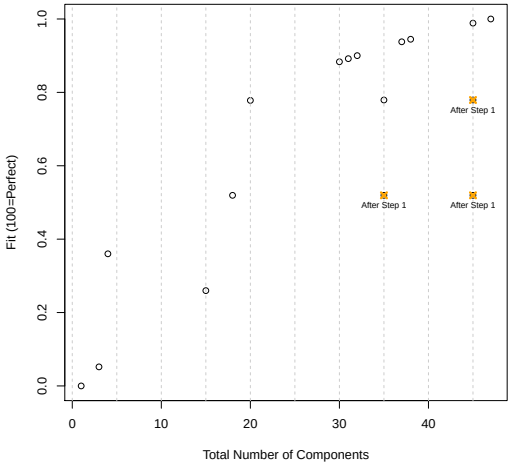
The recommended solution is the one with the largest st value [10], shown in green in Figure 4.1(f). Authors in [10] emphasize that this procedure is a rule of thumb that is best used as a recommendation tool and suggest further analysis of the top subset of solutions on the hull based on interpretability.

4.3 Discussion

In this chapter we introduced a diagnostic that can be used as a rule of thumb for determining the number of components to extract for previously developed dimension reduction techniques as well as for AICA. We assess model fit and complexity for each of the decompositions discussed in the thesis, along with this approach, in the next chapter, to learn about a storm on Jupiter.

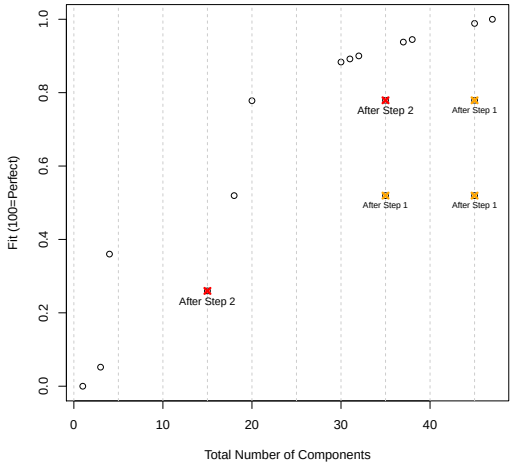


(a) Step 0: Plot fit versus model complexity.

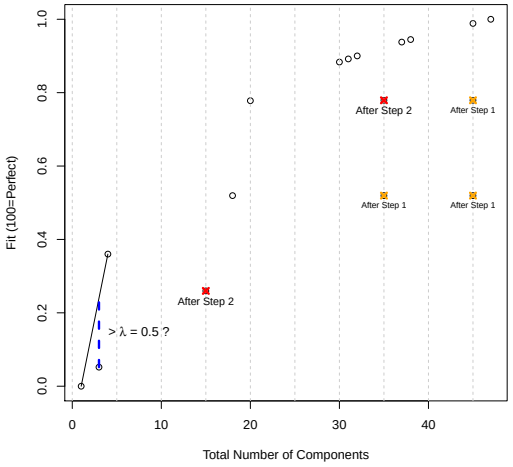


(b) Step 1: For each complexity level, choose the best solution.

Figure 4.1: Steps 0 and 1 (out of 5) for finding an optimal solution on the convex hull to determine how many components to extract.

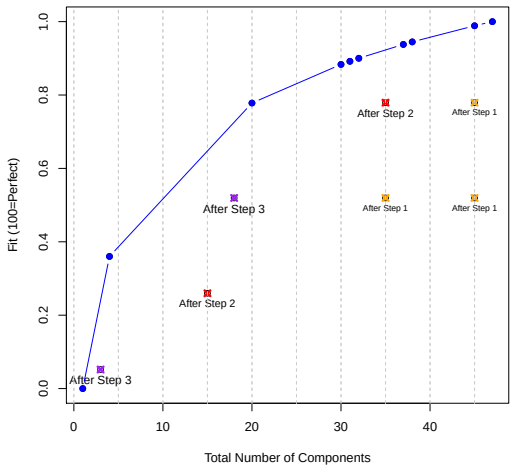


(c) Step 2: Exclude suboptimal solutions.

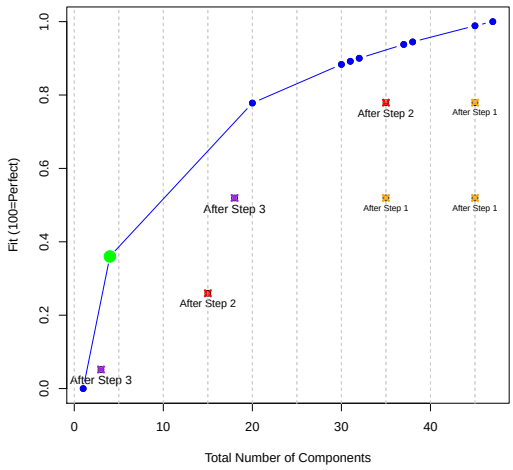


(d) Step 3: Exclude solutions if predicted fit is greater than observed fit.

Figure 4.1: Steps 2 and 3 (out of 5) for finding an optimal solution on the convex hull to determine how many components to extract.



(e) Step 4: Find the convex hull (blue).



(f) Step 5: Find optimal solution on the hull (green).

Figure 4.1: Steps 4 and 5 for finding an optimal solution on the convex hull to determine how many components to extract.

CHAPTER 5

Application of Array Independent Component Analysis to Study the White Oval

5.1 Introduction

Researchers and amateurs have been fascinated with Jupiter for years, because it is the largest planet in our solar system, and its two largest storms – the Great Red Spot (GRS) and the White Oval are, respectively – three times and about the size of Earth [22], [56]. In addition, understanding Jupiter's chemical composition may shed light on the solar system as a whole [44], since it is similar to the solar nebula, which gave rise to our solar system [34].

While the dynamics of the GRS[1] have not changed for hundreds of years, the White Oval is relatively new to Jupiter [61]. It is composed of three storms that formed in the 1930's and merged by the year 2000 into what is now known as Oval BA or the White Oval, as shown in Figure 5.1 [60]. It turned red late in 2005 [67].[2] The cause of its color change is still unknown [55].

Because Jupiter is hundreds of millions of miles away from Earth [13], we cannot directly observe its chemical and atmospheric composition [44]. We can learn about it from analysis of multispectral images of the planet or by retrieval algorithms that incorporate the known chemical and physical properties of the planet

[1]For a time-lapse video of Jupiter as seen from the 1979 Voyager 1 flyby, please see [3].

[2]Because of the color change, it is now also referred to as Little Red Spot [12] or Red Spot Junior [2] in the literature.

[64]. Our thesis focuses on the former: analyzing remotely sensed images of the sotrm captured in different wavelengths. Each wavelength tells us how much radiation is transmitted, absorbed or reflected by the object in the image [6]; for instance, the storm.[3] Since different elements have unique spectral footprints, the more images we take of the target in different wavelengths, the more accurately we can determine its composition [6]. Because no single atmospheric gas can explain Jupiter's color [11] or that of the White Oval, we need more than a handful of multispectral images to learn about the storm [67]. However, the more data we have, the harder it is to analyze. As a result, we turn to dimension reduction techniques to filter out noise and redundant information to understand the characteristics of the White Oval and the cause of its color change, which are unknown to this day [55].

The outline of the chapter is as follows. First, we discuss the data in more detail, with a focus on the joint distribution of the variables and motivate the need for volume reduction and robust inference. Second, we present the results of AICA, which is the only decomposition that uncovered interpretable components in the data, ones that explained spatial patterns in the White Oval in the longitudinal direction (or mode one of $\mathcal{X}$). For completeness, the results of previously developed decompositions, whose components did not shed light on the structure of the data, are presented in Appendix D.

[3]For more details on wavelengths as well as the electromagnetic spectrum, please see [59] and [70].

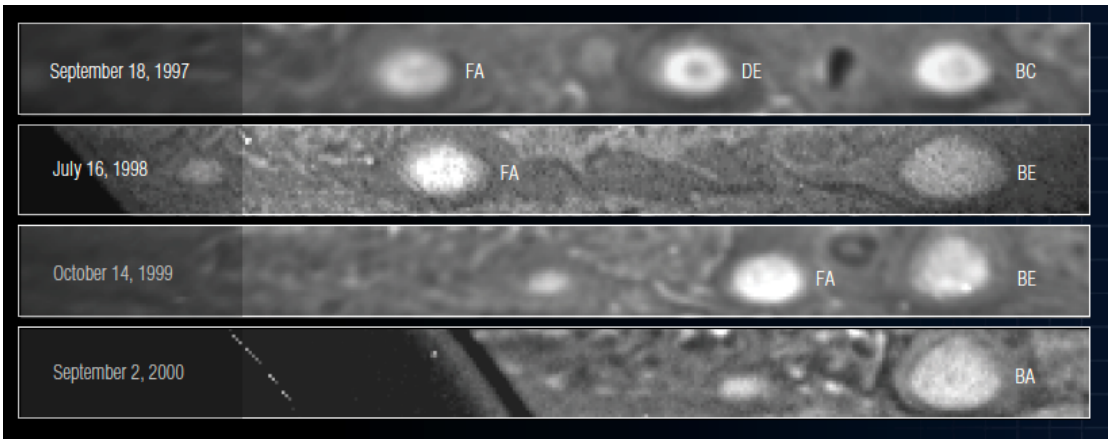


Figure 5.1: Formation of the White Oval [1][60].

5.2 Overview of the Data Set

We analyze images of the White Oval taken on April 8, 2006 shortly after it turned red in late 2005. The data set comes from two sources[4]: Infrared Telescope Facility (IRTF) with ground-based observations, and the Hubble Space Telescope's (HST) Advanced Camera for Surveys (ACS) with orbit-based images [67], [44]. There are six IRTF images taken in infrared channels (8.59, 10.77, 13.04, 17.65, 18.73 and 19.50 micrometer (μm)[5] wavelengths) and seven from HST (0.22, 0.25, 0.33, 0.435, 0.55, 0.658 and 0.892 μm) [67], [44]; the first three channels from the HST are in the ultraviolet; 435 nm, 550 nm, and 658 nm correspond to blue, green and red visible channels, respectively; the 892 nm wavelength is in near-infrared [29], [70]. The color-enhanced image [2] of the White Oval as captured by the HST is shown in Figure 5.2 [1].

[4]The data set was "(b)ased on observations made with the NASA/ESA Hubble Space Telescope, obtained at the Space Telescope Science Institute, which is operated by the Association of Universities for Research in Astronomy, Inc., under NASA contract NAS 5-26555. These observations are associated with programs GO10192, GO10782 and GO10783" [67]. It has been generously provided by Dr. Amy Simon-Miller of NASA's Goddard Space Flight Center and Dr. Padma Yanamandra-Fisher of (formerly Jet Propulsion Laboratory and currently) Space Science Telescope.

[5]1 μm (micrometer) = 1000 nm (nanometer).

The spectral channels are sensitive to certain characteristics of the Jovian atmosphere, which is shown in [73] and [11]. Past studies have indicated that remote sensing images of the Jovian atmosphere captured in the six infrared wavelengths (8.59, 10.77, 13.04, 17.65, 18.73 and 19.50 μm) can be used to derive temperature variations [12], [25] (as referenced in [12]). In addition, the 10.77 μm wavelength detects ammonia gas [12]. The 17.65 μm channel senses the temperatures in the tropopause, the area right above the troposphere and above the visible cloud tops [11], [73]; the 13.04 μm wavelength discerns the temperatures in deeper cloud layers [11], [73]. Images in the 0.25 μm and 0.892 μm, which correspond to the ultraviolet and infrared channels respectively, probe the haze and high clouds on Jupiter [67]. The 0.892 μm wavelength is also the methane absorption band [67]. In addition, short wavelengths (shorter than $3.5\mu m$) pick up the Raleigh scatter of gas molecules [68].

The data set, visualized in Figure 5.3, is composed of 13 images (or slices of our data cube), one per wavelength, with an image size of 61×43 pixels. Summary statistics for radiances across wavelengths are shown in Figure 5.4, and illustrate that (a) the means and standard deviations at different wavelengths vary; and (b) the distributions also vary by wavelength and need not be normally distributed or symmetric. As such, we will need to use higher order cumulants (order greater than two) in our search for independent components.

Two previous studies focused on analysis of the same remotely sensed images of the White Oval. The first focused on interpreting the Principal Components to shed light on the color change of the White Oval [67]. We show that while we also cannot explain the color change, our approach yields more interpretable components. The second study directly compared images of the White Oval with that of the GRS in order to make inferences about equality of their temperature

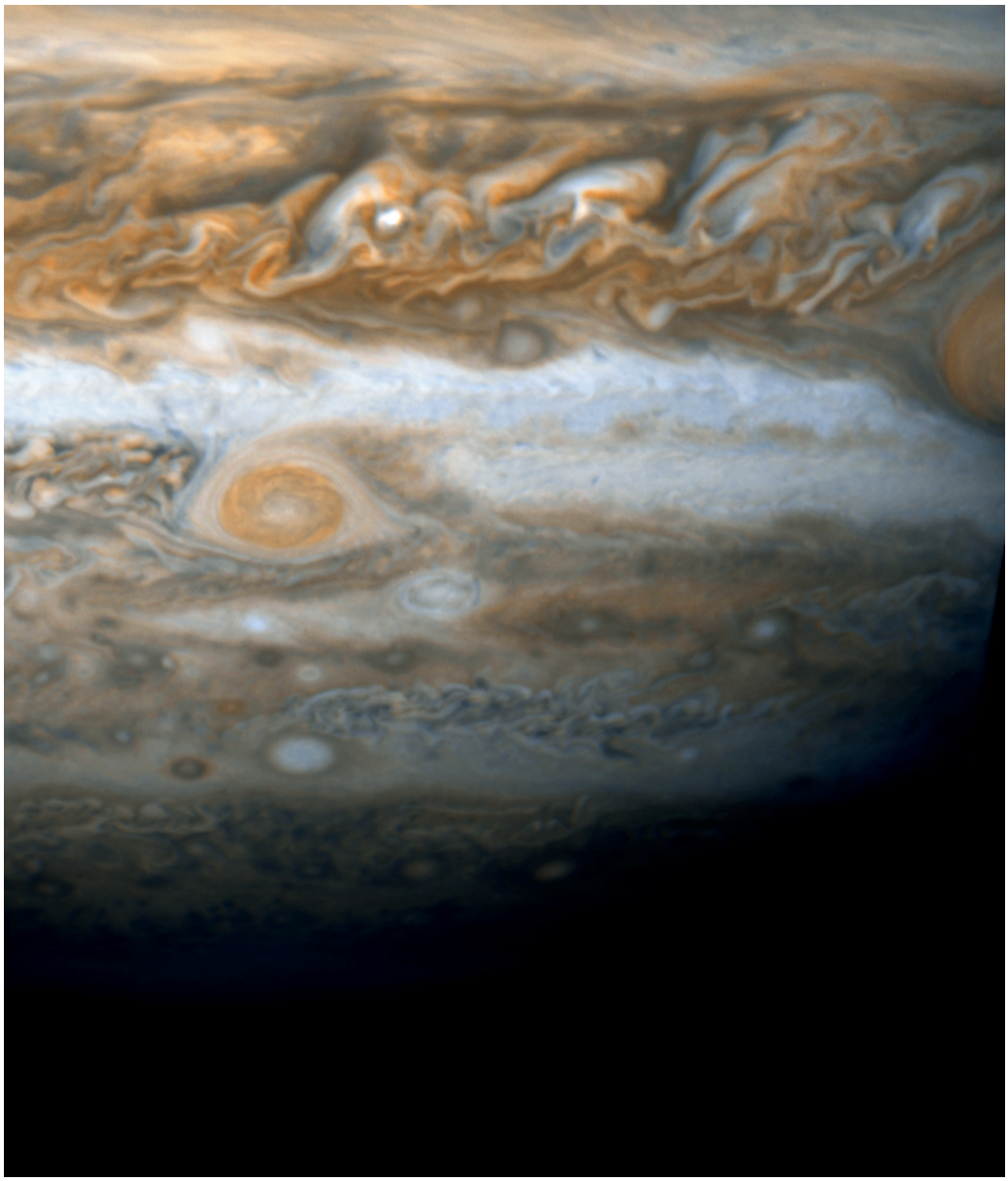


Figure 5.2: Contrast-enhanced image taken of the White Oval in red and blue light taken by HST ACS at 02:33 Universal Time on April 8, 2006 [2]. The Great Red Spot is partially visible on the far right [1].

profiles [12]. We confirm these findings via AICA.

5.3 Results of Array Independent Component Analysis

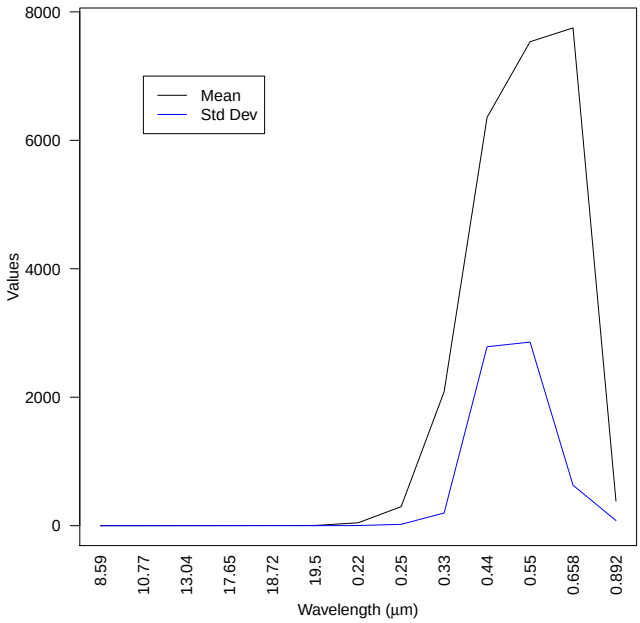
The goal of the decomposition is to reduce the dimensionality of the data set and find latent interpretable components that shed light on the characteristics of the White Oval. We use the framework developed in Chapter 4 to find an optimal solution on the convex hull of points to make a decision as to how many components to extract. For simplicity, we quantify model complexity as the total number of components we choose to extract from all modes.

We use AICA to find independent components for one mode of the data cube, accounting for its neighboring sides and vaiable interactions. For the analysis, we reorient our data to be $43 \times 61 \times 13$, with a corresponding $61 \times 61 \times 61 \times 13$ array $\mathcal{A}$, in order to try to understand the longitudinal direction of the storm. In Chapter 3, we introduced two versions of the AICA decomposition: one that allowed three out of four loading matrices to be equivalent up to span, and another approach that guaranteed exact equivalence of the three loading matrices. We present results of both decompositions in this section.

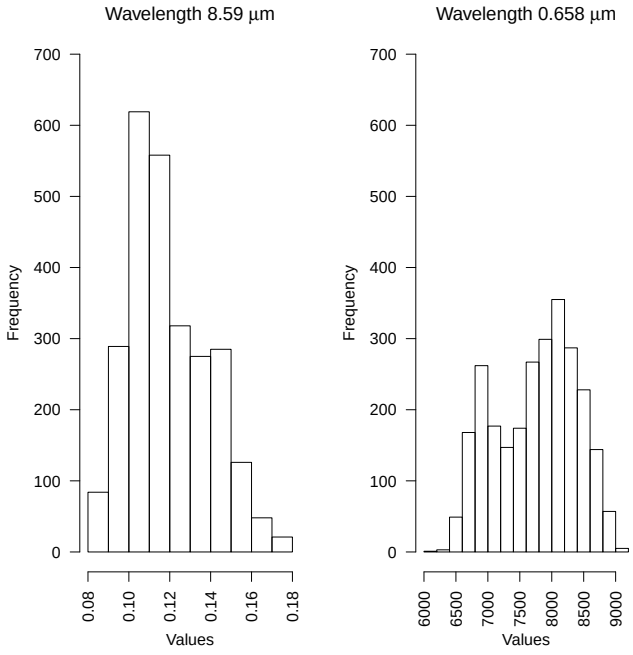
5.3.1 Array Independent Component Analysis with Loading Matrices up to Span

In the first version of AICA, we allow the first three loading matrices of $\mathcal{A}$ that relate the third order cumulants of each slice of our data set $\mathcal{X}$ to be equivalent up to span. We employ the convex hull diagnostic visualized in Figure 5.5 to determine how many components to choose. The top five models are highlighted in Table 5.1. While the diagnostic suggests we interpret six components from Modes 2 and 3 of the data set $\mathcal{X}$, the third-best option of 11 components, shown in Figure 5.6, gave us more interpretable components.

Figure 5.3: Plots of the raw images of the White Oval captured on April 8, 2005 in different wavelengths (μm). We can see that the ranges of the radiances vary across wavelengths.



(a) Summary statistics of radiances across wavelengths.



(b) Histograms of radiances for different wavelengths.

Figure 5.4: Exploratory analysis of the radiances shows that: (a) means and standard deviations vary by wavelength; and (b) distributions for the radiances at different wavelengths also vary and need not be gaussian.

Two out of 11 loadings, colored in grey and green, match the temperature gradients of the White Oval (up to an inversion along the x-axis) to that of the Great Red Spot [26]. In particular, the loadings shown in grey suggest equivalence in the temperature of the White Oval with that of the GRS in the North-South direction at 140 millibars or 0.14 bars, as shown in Figure 8 of [26] and reproduced in Figure 5.7 with the permission of the authors. The weights highlighted in green uncover another correspondence in the temperature gradient of the White Oval with that of the GRS in the West-East direction at 240 millibars or 0.24 bars, as shown in Figure 8 of [26] and reproduced in Figure 5.8 with the permission of the authors. The temperature gradients of the GRS were derived from remotely sensed images of the storm by the Very Large Telescope's (VLT) spectrometer and imager VISIR [23] captured over a period of three years, from 2006 to 2008 [26].

Figure 11.4 of [11] and image in [73] show that 0.14 bar and 0.24 bar (on the right hand side of the figures) correspond to the region right above the visible clouds, called the tropopause on Jupiter. This is precisely the region where the infrared wavelengths are known to be sensitive to temperature gradient, confirming our findings. The equivalence of the gradients in the two storms confirms a stable atmospheric layer [28] on Jupiter that is responsible for the creation of storms on the planet [54] (as referenced in [55]).

5.3.2 Array Independent Component Analysis with Equivalent Loading Matrices

Our second implementation of AICA guarantees equivalence of three of the loading matrices that diagonalize array $\mathcal{A}$ and find independent components; this is at the expense of monotone convergence. Because monotone convergence of the

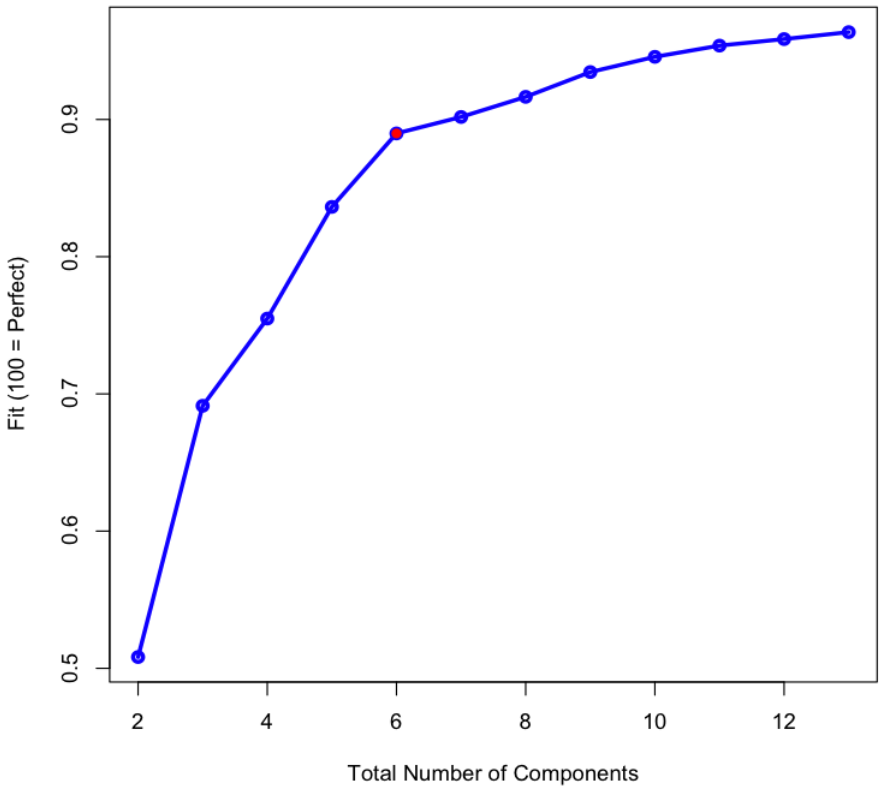


Figure 5.5: Tradeoff between model complexity and fit for AICA. Convex hull is highlighted in blue and the optimal solution in red.

algorithm is no longer guaranteed, we did not build a convex hull of solutions, but extracted 13 components – as many as there are wavelengths – for interpretation. The results show that the method also finds interpretable components. We suggest interpreting the four components derived from the loadings highlighted in color in Figure 5.9 as follows.

Components 9 and 3, created from the yellow and green loadings respectively, illustrate, similar to the discussion in Section 5.3.1 above, that the temperature gradients of the White Oval matches the ones above the high clouds of the Great Red Spot in the North-South and West-East directions [26] as shown in Figures 5.7 and 5.8, respectively.

Rank	Complexity	Fit (%)	Scree-test Value
1	6	0.89	4.46
2	3	0.69	2.88
3	11	0.95	1.72
4	9	0.93	1.62
5	5	0.84	1.52
		⋮	

Table 5.1: Top five solutions on the convex hull for AICA.

AICA uncovered two more interpretable components. Component 7, derived from the loadings in red, isolates the smaller storm to the South-East of the White Oval, shown in Figure 5.2. Component 2, formed by the blue weights in the region of columns seven through 15, picks up the Raleigh scatter of gas molecules [68].

While currently we cannot interpret the remaining nine components, it is something that we will strive to do with further research and collaboration and is discussed in more detail in the following chapter. For completeness, the loading matrix responsible for finding uncorrelated components from the third mode of $\mathcal{X}$ (and the fourth mode of $\mathcal{A}$) is shown in Figure 5.10.

5.4 Discussion

In this chapter we applied the new methodology introduced in the thesis to analyze remotely sensed images of the White Oval. Comparing the results of AICA to those of previously developed dimension reduction techniques (which did not find any interpretable components and have been included for completeness in Appendix D), the components uncovered by AICA included a discovery of similarity in the temperature gradients between the White Oval and the GRS, a relationship

previously developed dimension reduction techniques did not pick up on. As a result, we conclude that AICA was the most appropriate decomposition for analysis of the remote sensing data set of the White Oval.

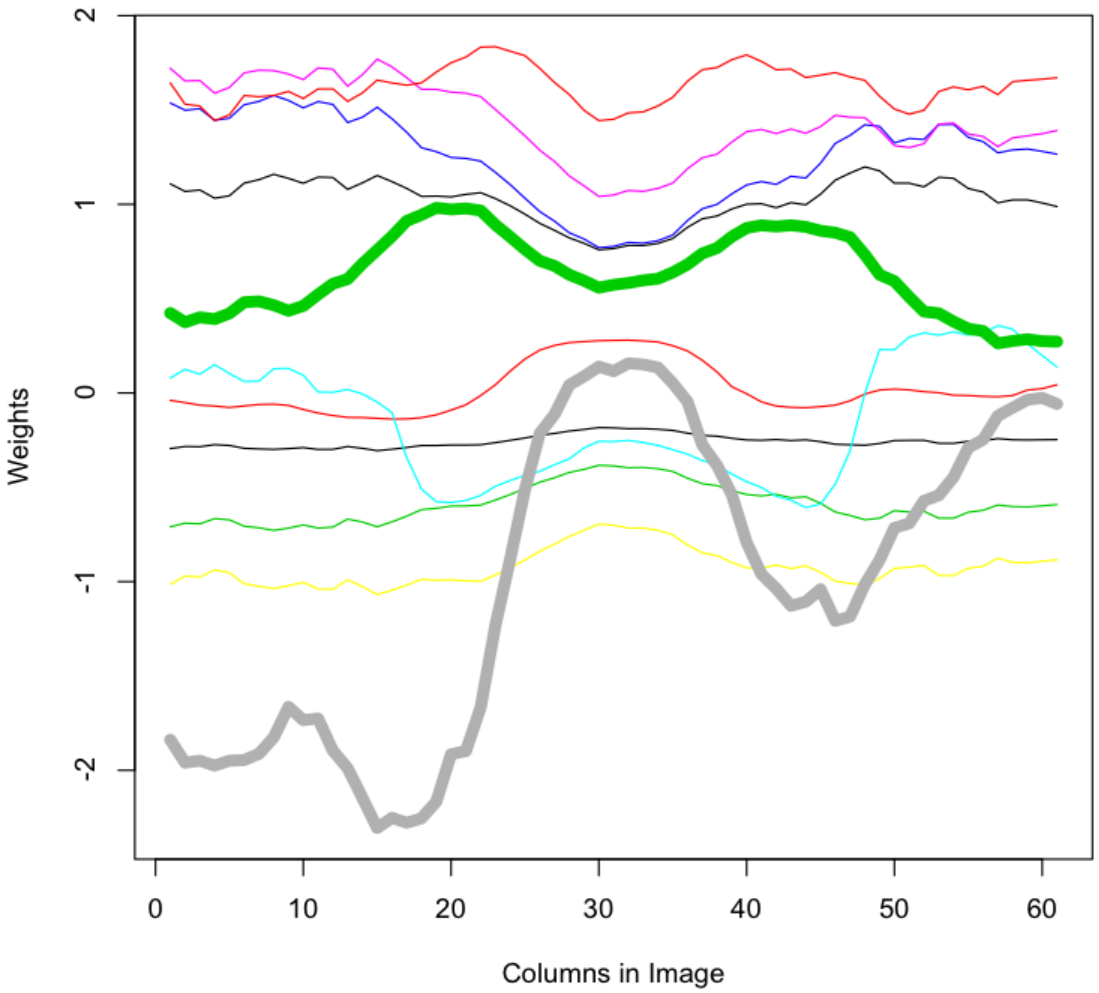


Figure 5.6: Results of AICA decomposition extracting 11 components, where the highlighted loadings suggest that the temperature gradient of the GRS [26] and the White Oval match. The weights in gray mimic the North-South profile, reproduced in Figure 5.7; the weights in green mimic the West-East profile, reproduced in Figure 5.8 with permission of the authors [26].

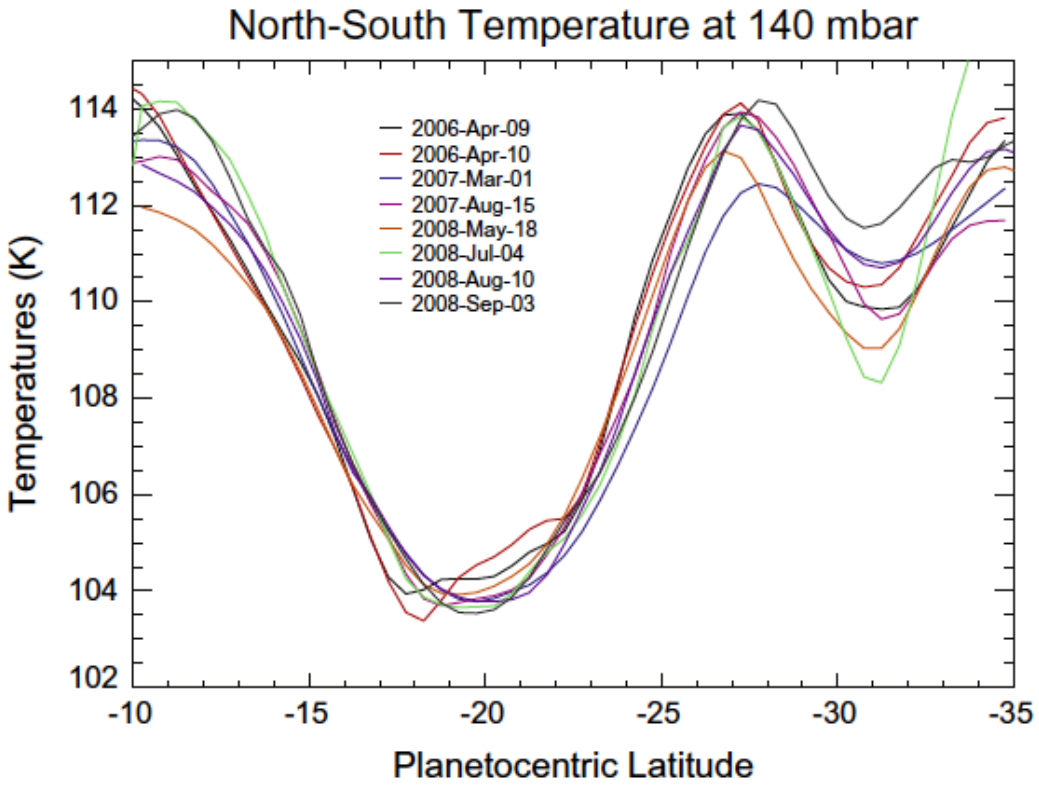


Figure 5.7: Reproduction of a subfigure of Figure 8 of [26] with the North–South temperature gradient of the Great Red Spot derived from remotely sensed images by the Very Large Telescope's (VLT) spectrometer and imager VISIR.

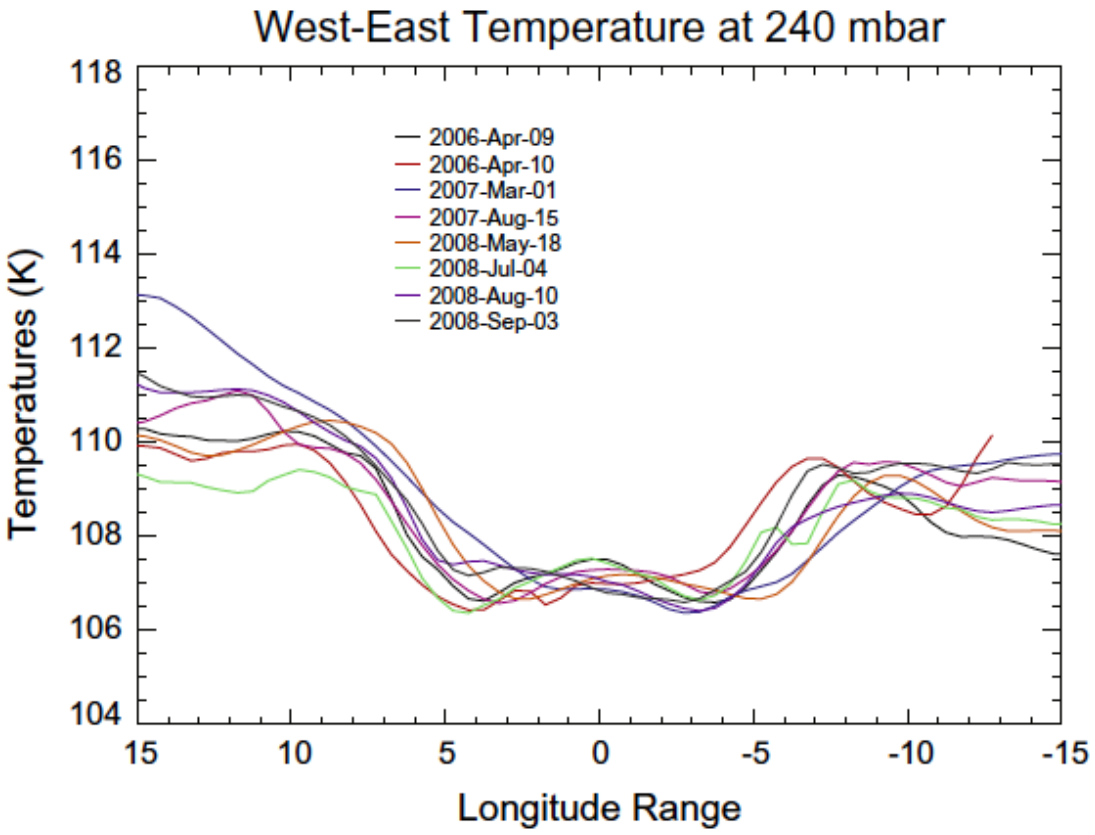


Figure 5.8: Reproduction of a subfigure of Figure 8 of [26] with the West–East temperature gradient of the Great Red Spot derived from remotely sensed images by the Very Large Telescope's (VLT) spectrometer and imager VISIR.

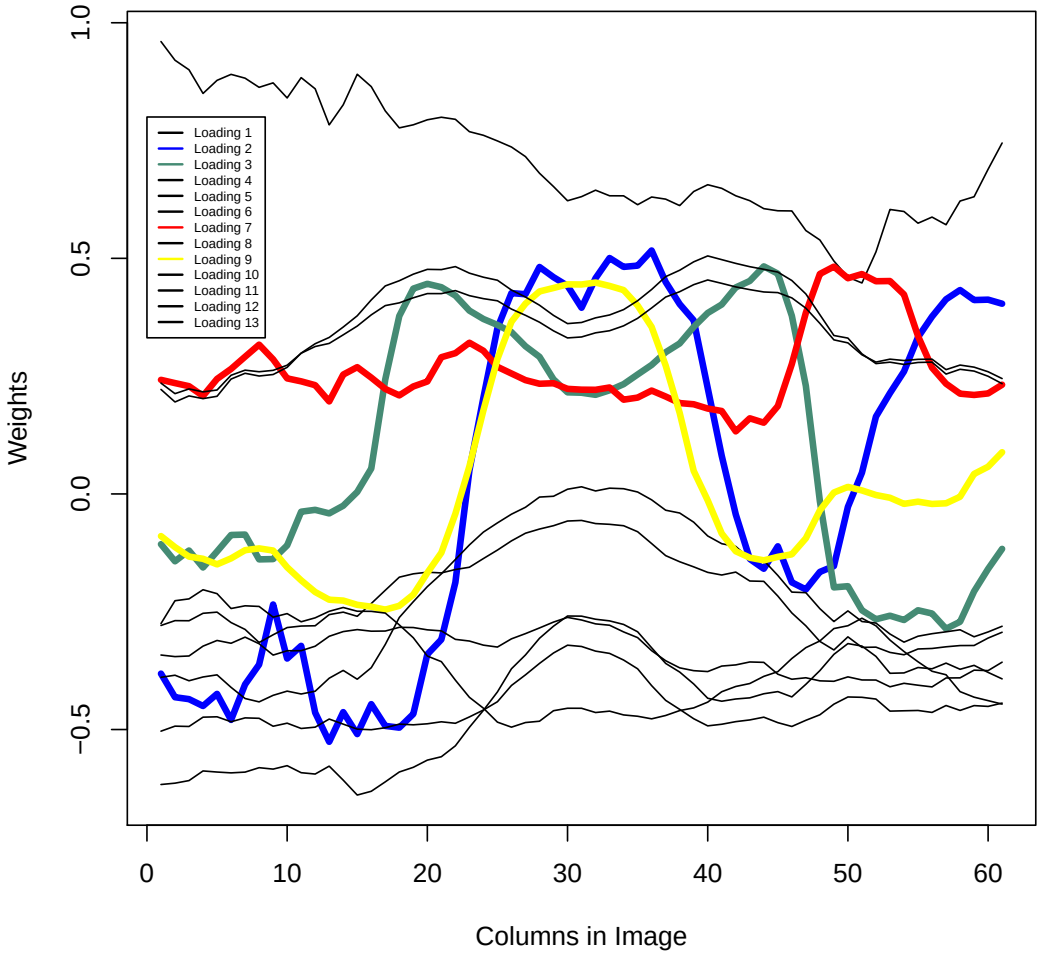


Figure 5.9: Loading matrix from the AICA decomposition that is equivalent across the first three modes of $\mathcal{A}$, extracting 13 components. The loadings shown in yellow and green suggest that the temperature gradients of the GRS [26] and the White Oval match in the North-South and West-East directions (respectively); the weights in red discern the smaller storm to the South-East of the White Oval; the loadings in blue (in the region of columns seven through 15) sense the Raleigh scatter [68].

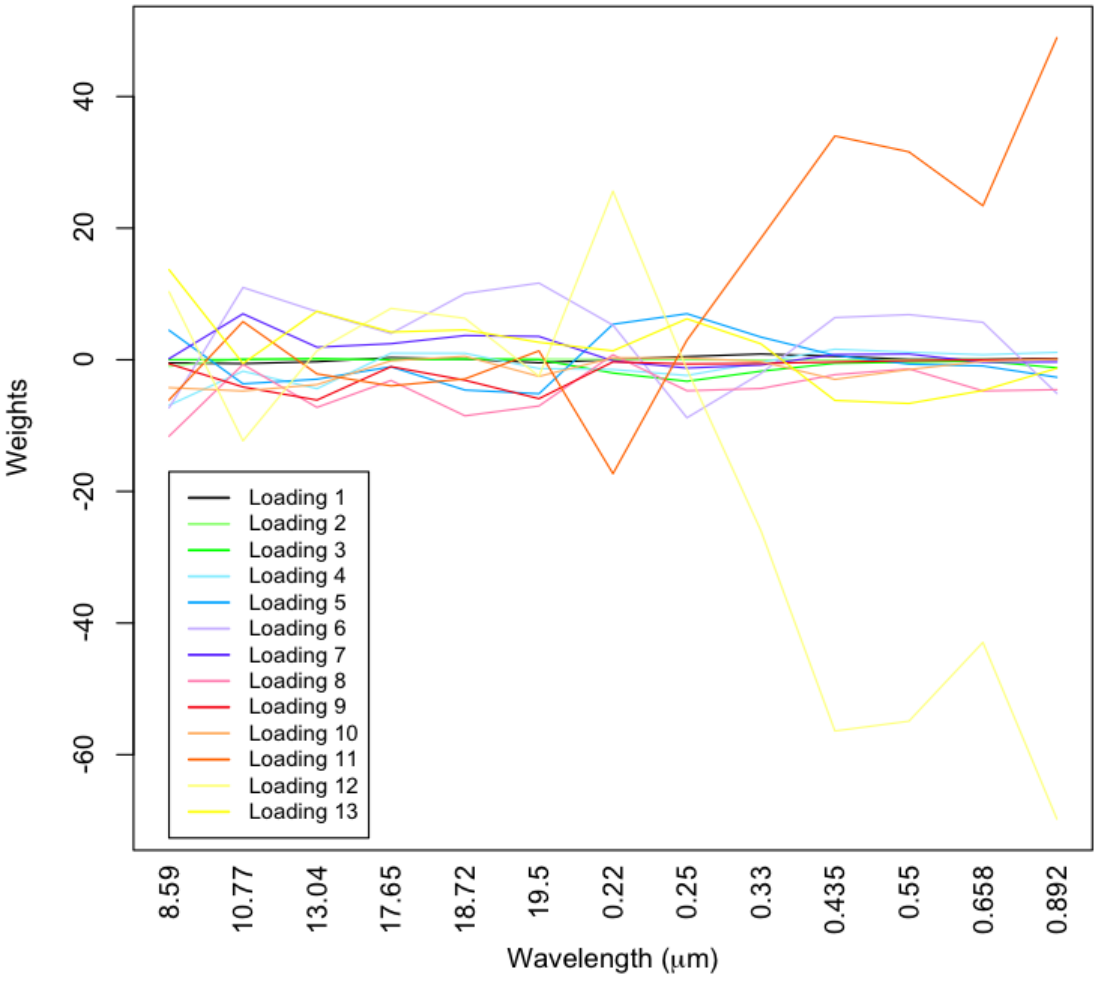


Figure 5.10: Loading matrix from the AICA decomposition for the fourth mode of $\mathcal{A}$, extracting 13 components.

CHAPTER 6

Conclusion

6.1 Discussion

The goal of the thesis was to introduce a new dimension reduction technique called Array Independent Component Analysis (AICA) that found independent components for a mode of a data cube that incorporated variable interactions with its other modes. To motivate the need for the approach, we presented an overview of the most commonly used dimension reduction methods and discussed their limitations for finding independent components for a three mode data set. The techniques found independent components via ICA for an array with two modes, or uncorrelated components for an array with two or three modes, via PCA and its three mode analogs, Tucker-3 and CP. Our approach is a generalization of ICA for three mode data sets, decomposition that has not been possible until now.

In addition, we presented an overview of a diagnostic that can be used as a rule of thumb to answer one of the key questions of any dimension reduction technique: how many components to extract for each mode of the data cube. It has been applied to study decompositions for two and three mode data sets previously. We extended its framework to aid with AICA in a decomposition of a four mode array, with symmetries in three modes.

If we did not need to compare across the dimension reduction techniques, residual analysis of each slice of the array would have sufficed for this remotely sensed

data set, as we only had 13 slices to analyze. However, this is not practical for other remotely sensed applications, for instance, in the case of Atmospheric Infrared Sounder (AIRS), where a new remotely sensed data cube of Earth is captured in over 2,000 wavelengths and generated virtually every second for a different part of the globe [58]. In addition, we did not evaluate the techniques based on cross-validation, because it would greatly increase computation time of each decomposition, most of which are based on the slowly converging Alternating Least Squares algorithm [7]. As a result, we evaluated all of the decompositions described in the thesis based on interpretability of the components derived from analysis of remotely sensed images of the White Oval on Jupiter; AICA was the only technique to meet our criteria.

Although PCA is the most commonly employed decomposition, by evaluating the interpretability of the components derived from the techniques discussed in the thesis, we learned about more general methods.

6.2 Further Research

We conclude with remarks on further extensions of the AICA approach, focusing on the areas of application, methodology and algorithm implementation.

First and foremost, we would like to see more of the components derived from AICA interpreted. This may be possible by comparing the components derived by AICA from analyzing the White Oval before and after the color change. While the primary goal of the thesis was to introduce a new dimension reduction technique and discuss it via a case study, it is worthwhile to evaluate the robustness of AICA and its contribution to discovering meaningful latent variables, by decomposing other data sets as well. We recommend analyzing remotely sensed Atmospheric Infrared Sounder (AIRS) data and comparing the resulting compo-

nents with the chemical and atmospheric patterns that are known and derived via retrieval algorithms.

In its current formulation, AICA finds a set of independent components for one mode of a data array, diagonalizing an array of third order cumulants for each slice of the original data set to do so. We would like to see each of those conditions relaxed: to find independent components for multiple modes of the data cube, while simultaneously diagonalizing cumulants of orders one through four (for instance), and allowing the possibility of extracting one component at a time. Another extension we envision is a framework flexible enough to handle augmenting the current data set, for instance, as in the case of AIRS data, where a new region of the Earth is sensed approximately every second [58].

Our further considerations for the current algorithm, based on the Alternating Least Squares Algorithm, are as follows. As with any computing problem, we recommend an acceleration of the algorithm to obtain faster results. While we have tried speeding up the second version of the AICA algorithm (which guarantees equivalent loading matrices for the first three modes of array $\mathcal{A}$ that is symmetric in three modes) in serial via a vector-epsilon approach (as in [48] and [51]), the original algorithm's lack of a monotone convergence hindered the process. As a result, we recommend solving the convergence problem, which will allow us to implement the AICA framework in distributed environment. Only then will it be able to handle decompositions of massive data sets that are too large to be stored on one machine, such as analysis of AIRS data, whose acquisition of raw data is about 13.4 GB per day [74].

APPENDIX A

Introduction to Array Operations

A.1 Overview

In this thesis, we focus on dimension reduction techniques for finding latent variables for data sets with at least two modes. To best retain the variables' relationships, the data sets are stored and analyzed as arrays, which generalize the notion of a matrix.[1] In this chapter, we present the concepts needed to understand the array operations discussed in the body of the thesis.

A.2 Array Operations

A.2.1 Matricizing an Array

As we saw in Section 1.2, three mode data sets are common in practice. Unfortunately, the usual practice is not to use the three-mode data set's representation for analysis; instead, researchers reshape a higher mode (number of modes greater than two) data set to two modes (as discussed in the following section) and use two mode dimension reduction techniques, such as PCA and ICA for the decomposition. There are drawbacks to analyzing a matricized array, as discussed in Section 2.4. For more robust [37] and interpretable results, we do not recommend matricizing a higher mode (order greater than two) data set.

There are other applications for a matricized array, however. For instance, we

[1] A vector is an array with one mode; a matrix is an array with two modes.

need to matricize an array as part of a higher mode data decomposition that finds loading matrices for each of its modes. To do so, we unwrap the array in a given mode; this matricization scheme is slightly different than unwrapping an array for PCA or ICA. There are many ways to unfold an array in a given mode, as described, for instance in [38] and [40]. Authors in [40] mention that the order in which we unwrap the modes does not affect the results of the decomposition as long as we are consistent, as we can always rotate the array. We outline one way to do so in the following sections for three and four mode arrays, respectively.

A.2.1.1 Matricizing a Three Mode Array for PCA or ICA

One way to reshape a three-mode data set such that it will have two modes is to have the column vectors in the new data set correspond to the vectorized slices[2] of the original data set; the rows of this new matrix then, will correspond to all possible values Mode 1 and Mode 2 can take. For instance, the three mode example discussed in Section 1.2 (and visualized in Figure 1.3) can be unwrapped as shown in Figure A.1.

A.2.1.2 Matricizing a Three Mode Array for a Higher Order Data Decomposition

We illustrate how to unwrap a three-mode array $\mathcal{X}_{I\times J\times K}$ in each of its modes by example. Suppose that our array $\mathcal{X}_{4\times 3\times 2}$ has two slices:

$$\mathbf{X}_1 = \begin{pmatrix} 1 & 5 & 9 \\ 2 & 6 & 10 \\ 3 & 7 & 11 \\ 4 & 8 & 12 \end{pmatrix} \tag{A.1}$$

[2]A slice that is vectorized has the matrix columns stacked one below the other to form a matrix with one long column.

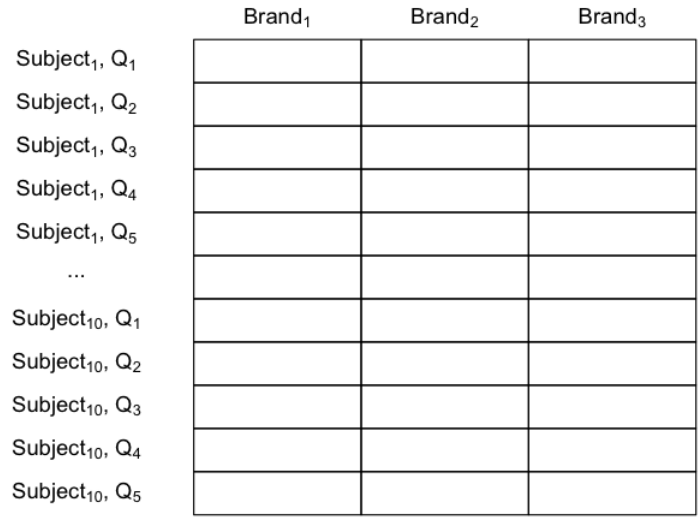


Figure A.1: Schematic of one way we can unwrap a three mode array introduced in Figure 1.3, by vectorizing the slices.

and

$$\mathbf{X}_2 = \begin{pmatrix} 13 & 17 & 21 \\ 14 & 18 & 22 \\ 15 & 19 & 23 \\ 16 & 20 & 24 \end{pmatrix} \tag{A.2}$$

$$\mathbf{X}_{(1)} = \begin{array}{c} \\ \text{row}_1 \\ \text{row}_2 \\ \text{row}_3 \\ \text{row}_4 \end{array} \begin{array}{c} \begin{array}{ccc|ccc} \text{col}_1\text{,slice}_1 & \text{col}_2\text{,slice}_1 & \text{col}_3\text{,slice}_1 & \text{col}_1\text{,slice}_2 & \text{col}_2\text{,slice}_2 & \text{col}_3\text{,slice}_2 \end{array} \\ \begin{pmatrix} 1 & 5 & 9 & 13 & 17 & 21 \\ 2 & 6 & 10 & 14 & 18 & 22 \\ 3 & 7 & 11 & 15 & 19 & 23 \\ 4 & 8 & 12 & 16 & 20 & 24 \end{pmatrix} \end{array}$$

Figure A.2: An example of unwrapping an array defined in Equations A.1 and A.2 in Mode 1.

By unwrapping $\mathcal{X}$ in Mode 1 (by our convention), we get a two mode array shown

in Figure A.2, which is a $4 \times (3 \times 2) = 4 \times 6$. In the general case, by unwrapping an $I \times J \times K$ array $\mathcal{X}$ in mode 1, we get an $I \times (JK)$ matrix. Figure A.3 illustrates how to unwrap the array $\mathcal{X}$ in Mode 2, which in our example results in a $3 \times (4 \times 2) = 3 \times 8$ array; in general, this corresponds to an $J \times (IK)$ array. For completeness, the consistent way to matricize our array in Mode 3 is to generate a $2 \times (4 \times 3) = 2 \times 12$ array as shown in Figure A.4; in general, unfolding an array $\mathcal{X}_{I \times J \times K}$ in mode 3 results in a $K \times (IJ)$ array.

A.2.1.3 Matricizing a Four Mode Array for a Higher Order Data Decompositions

AICA decomposes four-mode arrays by operating on their unfolded versions. Let $\mathcal{Y}$ be one such array, with dimensions I, J, K, and L, respectively, as shown in Figure A.5. Figures A.6–A.9 illustrate how to unwrap the array $\mathcal{Y}$ in each of its modes, resulting in matrices $I \times JKL$, $J \times IKL$, $K \times IJL$ and $L \times IJK$, respectively. For arrays with higher modes, we proceed to unfold the arrays in each of its modes similarly.

A.2.2 Hadamand Product

The Hadamand product is the element-wise product of two matrices with the same dimensions; it is denoted by $*$. To illustrate this via an example, let $\mathbf{A} = \mathbf{X}_{(1)}$ and $\mathbf{B} = \mathbf{X}_{(2)}$, as defined in Equations A.1 and A.2, respectively. We find their Hadamand product as described in Equations A.3 and A.4:

$$\mathbf{A} * \mathbf{B} \triangleq \begin{pmatrix} a_{11}b_{11} & a_{12}b_{12} & a_{13}b_{13} \\ a_{21}b_{21} & a_{22}b_{22} & a_{23}b_{23} \\ a_{31}b_{31} & a_{32}b_{32} & a_{33}b_{33} \\ a_{41}b_{41} & a_{42}b_{42} & a_{43}b_{43} \end{pmatrix} = \tag{A.3}$$

$$
\mathbf{X}_{(2)} = \begin{array}{c} \\ \mathrm{col}_1 \\ \mathrm{col}_2 \\ \mathrm{col}_3 \end{array}
\begin{array}{c}
\begin{array}{cccc|cccc}
\mathrm{row}_1,\mathrm{slice}_1 & \mathrm{row}_2,\mathrm{slice}_1 & \mathrm{row}_3,\mathrm{slice}_1 & \mathrm{row}_4,\mathrm{slice}_1 & \mathrm{row}_1,\mathrm{slice}_2 & \mathrm{row}_2,\mathrm{slice}_2 & \mathrm{row}_3,\mathrm{slice}_2 & \mathrm{row}_4,\mathrm{slice}_2 \\
\hline
1 & 2 & 3 & 4 & 13 & 14 & 15 & 16 \\
5 & 6 & 7 & 8 & 17 & 18 & 19 & 20 \\
9 & 10 & 11 & 12 & 21 & 22 & 23 & 24
\end{array}
\end{array}
$$

Figure A.3: An example of unwrapping an array defined in Equations A.1 and A.2 in Mode 2.

$$
\mathbf{X}_{(3)} = \begin{array}{c} \\ \text{slice}_1 \\ \text{slice}_2 \end{array} \begin{array}{c} \begin{array}{cccccccc} \text{row}_1,\text{col}_1 & \text{row}_2,\text{col}_1 & \text{row}_3,\text{col}_1 & \text{row}_4,\text{col}_1 & \cdots & \text{row}_1,\text{col}_3 & \text{row}_2,\text{col}_3 & \text{row}_3,\text{col}_3 & \text{row}_4,\text{col}_3 \end{array} \\ \begin{pmatrix} 1 & 2 & 3 & 4 & \cdots & 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 & \cdots & 21 & 22 & 23 & 24 \end{pmatrix} \end{array}
$$

Figure A.4: An example of unwrapping an array defined in Equations A.1 and A.2 in Mode 3.

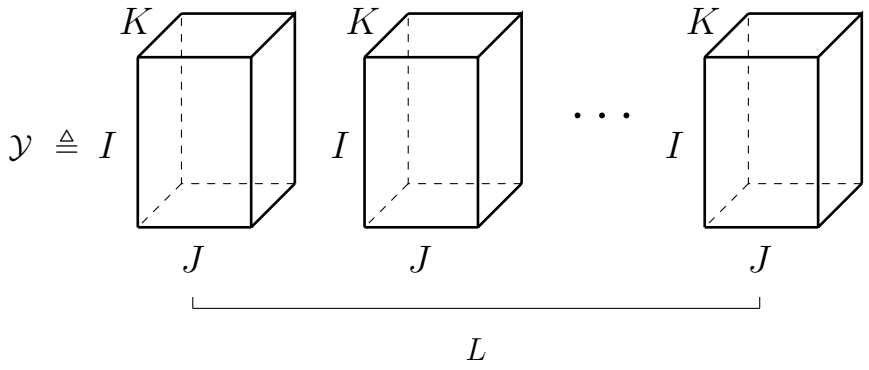


Figure A.5: Schematic of array $\mathcal{Y}$ that we unwrap in different modes.

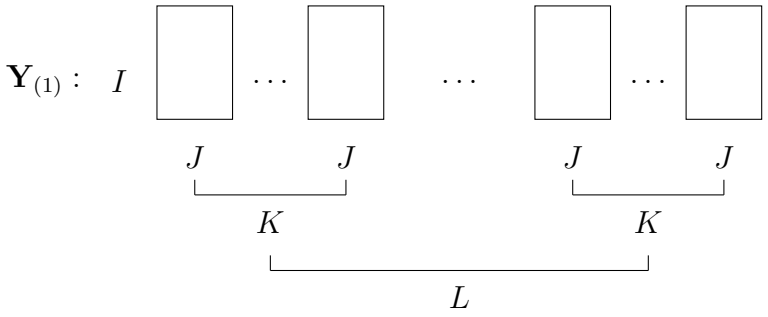


Figure A.6: Visualization of one way to matricize $\mathcal{Y}$ with dimensions $I \times J \times K \times L$ in mode 1, to get an $I \times JKL$ matrix.

$$= \begin{pmatrix} 13 & 85 & 189 \\ 28 & 108 & 220 \\ 45 & 133 & 253 \\ 64 & 160 & 288 \end{pmatrix}. \tag{A.4}$$

A.2.3 Kronecker Product

A matrix that can be expressed as a Kronecker product of two matrices $\mathbf{A}$ and $\mathbf{B}$ simplifies the formulas for finding its eigenvalues [66]. We define the product

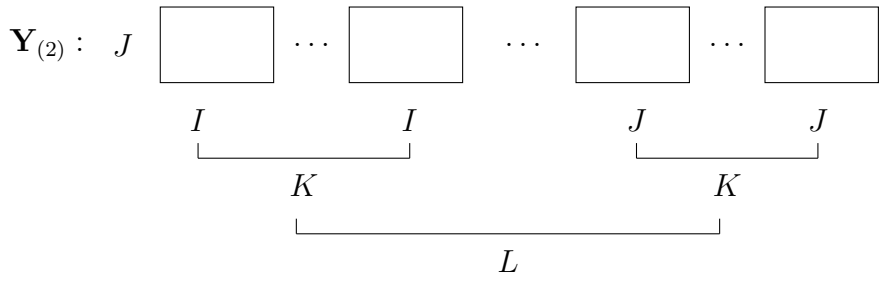


Figure A.7: Visualization of one way to matricize $\mathcal{Y}$ with dimensions $I \times J \times K \times L$ in mode 2, to get a $J \times IKL$ matrix.

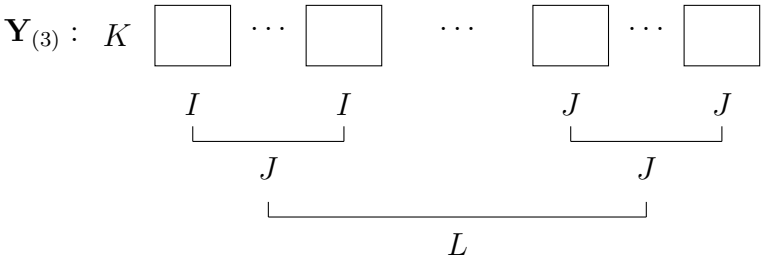


Figure A.8: Visualization of one way to matricize $\mathcal{Y}$ with dimensions $I \times J \times K \times L$ in mode 3, to get a $K \times IJL$ matrix.

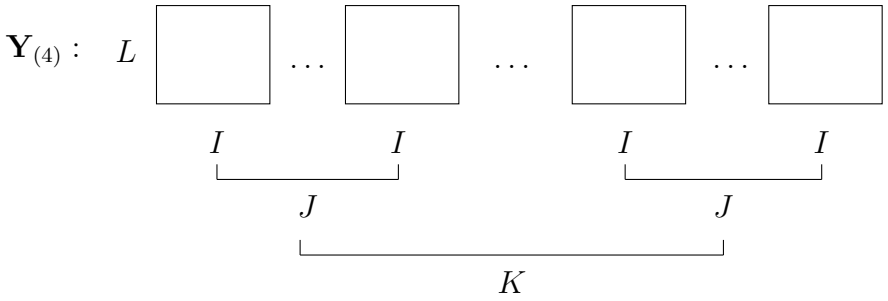


Figure A.9: Visualization of one way to matricize $\mathcal{Y}$ with dimensions $I \times J \times K \times L$ in mode 4, to get an $L \times IJK$ matrix.

below and return to it in the discussion of the Tucker-3 decomposition for three (or higher) mode arrays.

The Kronecker product is a generalization of the outer product between two vectors, to a product of two matrices. Suppose that

$$\mathbf{A} = \begin{pmatrix} 1 & 3 \\ 2 & 4 \end{pmatrix}$$

and

$$\mathbf{B} = \begin{pmatrix} 5 & 7 & 9 \\ 6 & 8 & 10 \end{pmatrix}.$$

Then the Kronecker product of matrix $\mathbf{A}$ and $\mathbf{B}$, denoted by $\mathbf{A} \otimes \mathbf{B}$, is defined as:

$$\mathbf{A} \otimes \mathbf{B} \triangleq \begin{pmatrix} a_{11}\mathbf{B} & a_{12}\mathbf{B} \\ a_{21}\mathbf{B} & a_{22}\mathbf{B} \end{pmatrix} \tag{A.5}$$

$$= \left(\begin{array}{ccc:ccc} 5 & 7 & 9 & 15 & 21 & 27 \\ 6 & 8 & 10 & 18 & 24 & 30 \\ \hdashline 10 & 14 & 18 & 20 & 28 & 36 \\ 12 & 16 & 20 & 24 & 32 & 40 \end{array}\right). \tag{A.6}$$

In general, we can perform this operation on any two matrices; that is, if $\dim(\mathbf{A}) = I \times J$ and $\dim(\mathbf{B}) = K \times L$, then $\dim(\mathbf{A} \otimes \mathbf{B}) = (IK) \times (JL)$.

A.2.4 Khatri-Rao Product

The Khatri-Rao product is a column-wise Kronecker product [40]. It is useful in the formulation of the CP decomposition discussed in the body of the thesis [40]. It is defined for two matrices $\mathbf{A}$ and $\mathbf{B}$ that have the same number of columns M:

$$\mathbf{A} \odot \mathbf{B} \triangleq (A_1 \otimes B_1 \quad A_2 \otimes B_2 \quad \cdots \quad A_M \otimes B_M)$$

That is, if $\mathbf{A}$ is as defined in Section A.2.3, and

$$\mathbf{B} = \begin{pmatrix} 5 & 7 \\ 6 & 8 \end{pmatrix}$$

with $B_1 = \begin{pmatrix} 5 \\ 6 \end{pmatrix}$ and $B_2 = \begin{pmatrix} 7 \\ 8 \end{pmatrix}$. Then

$$\mathbf{A} \odot \mathbf{B} \triangleq \begin{pmatrix} a_{11}B_1 & a_{12}B_2 \\ a_{21}B_1 & a_{22}B_2 \end{pmatrix} \tag{A.7}$$

$$= \left(\begin{array}{c:c:c} 5 & 21 \\ 6 & 24 \\ \hdashline --- & --- & --- \\ 10 & 28 \\ 12 & 32 \end{array}\right) \tag{A.8}$$

where the dimension of $\mathbf{A} \odot \mathbf{B}$ is 4×2. In general, if $\dim(\mathbf{A}) = I \times K$ and $\dim(\mathbf{B}) = J \times K$, then $\dim(\mathbf{A} \odot \mathbf{B}) = (IJ) \times K$.

A.2.5 n-mode Product

A matrix is a two mode array, where we can operate on each of its two modes via a vector or a matrix. That means that for an n-mode array, we can operate on any of its n modes. Figure A.10 shows graphically how a three mode array $\mathcal{X}_{N\times M\times K}$ can be multiplied on each side by matrices $\mathbf{A}_{P\times N}$, $\mathbf{B}_{Q\times M}$ and $\mathbf{C}_{R\times K}$ to result in an array that is $P \times Q \times R$. This concept will be useful when we discuss the Tucker, CP and AICA decompositions. The multiplication symbol $\times_i$ specifies that we want the multiplication to take place in the i^{th} mode of the array. Examples of how to carry out the product for one and more than one mode follow.

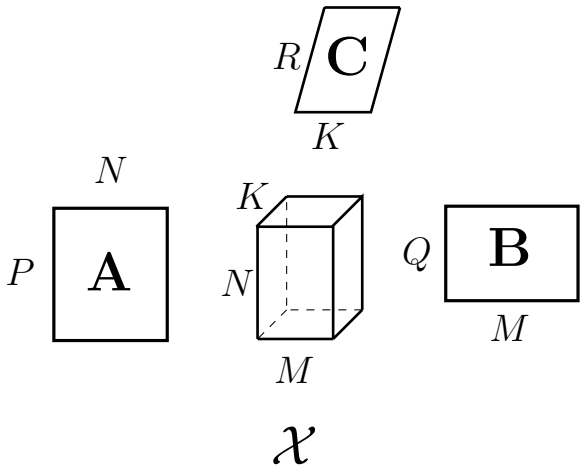


Figure A.10: Schematic of multiplying a three mode array $\mathcal{X}_{N\times M\times K}$ on all three sides by matrices $\mathbf{A}_{P\times N}$, $\mathbf{B}_{Q\times M}$ and $\mathbf{C}_{R\times K}$ to get a $P \times Q \times R$ array.

A.2.5.1 Multiplying an Array in One Mode

Suppose that we wish to multiply the first mode of array $\mathcal{X}_{4\times 3\times 2}$ defined in Equations A.1 and A.2 by $\mathbf{C}$, where

$$\mathbf{C} = \begin{pmatrix} 1 & 3 & 5 & 7 \\ 2 & 4 & 6 & 8 \end{pmatrix}. \tag{A.9}$$

Mathematically, this is expressed as follows:

$$\begin{aligned}\mathcal{Y} = \mathcal{X} \times_1 \mathbf{C} = \\ &\triangleq \mathbf{C}\,\mathbf{X}_{(1)} \\ &= \begin{pmatrix} 50 & 114 & 178 & 242 & 306 & 370 \\ 60 & 140 & 220 & 300 & 380 & 460 \end{pmatrix},\end{aligned} \tag{A.10}$$

with elements arranged as follows:

$$\mathbf{Y}_1 = \begin{pmatrix} 50 & 114 & 178 \\ 60 & 140 & 220 \end{pmatrix} \tag{A.11}$$

and

$$\mathbf{Y}_2 = \begin{pmatrix} 242 & 306 & 370 \\ 300 & 380 & 460 \end{pmatrix}. \tag{A.12}$$

In our case, the dimension of the modes for array $\mathcal{Y} = \mathcal{X} \times_1 \mathbf{C}$ are $2 \times 3 \times 2$. In general, if $\dim(\mathbf{C}) = L \times I$ and $\dim(\mathcal{X}) = I \times J \times K$, then $\dim(\mathcal{X} \times_1 \mathbf{C}) = L \times J \times K$.

A.2.5.2 Multiplying an Array in Multiple Modes

We use similar notation to multiply more than one mode of array $\mathcal{X}_{I \times J \times K}$. For instance, suppose we want to multiply $\mathcal{X}$ by matrices $\mathbf{C}_{P \times I}$, $\mathbf{D}_{Q \times J}$ and $\mathbf{H}_{R \times K}$ in each of the modes of $\mathcal{X}$, respectively. We can express the product as follows [40], [41], [27]:

$$\begin{aligned}\mathcal{Y} &= \mathcal{X} \times_1 \mathbf{C} \times_2 \mathbf{D} \times_3 \mathbf{H} \\ &= \mathbf{H}\Big(\mathbf{D}\Big(\mathbf{C}\mathbf{X}_{(1)}\Big)_{(2)}\Big)_{(3)}\end{aligned} \tag{A.13}$$

which means that, reading from the inside of Equation A.13, we:

1. multiply $\mathbf{C}$ by array $\mathcal{X}$ that is unfolded in the first mode to get a $P \times JK$ matrix;

2. reshape the matrix into a $P \times J \times K$ array;

3. multiply $\mathbf{D}$ by array from Step (2) that is unfolded in the second mode to get a $Q \times PK$ matrix;

4. reshape the matrix into a $P \times Q \times K$ array;

5. multiply $\mathbf{H}$ by array from Step (4) that is unfolded in the third mode to get an $R \times PQ$ matrix;

6. reshape the matrix into a $P \times Q \times R$ array.

This framework is useful in the Tucker, CP and AICA decompositions, where we use n-mode multiplication to reduce the dimensionality of each mode.

APPENDIX B

Cumulants

B.1 Motivation for Using Cumulants

The goal of AICA is to find a new set of variables from the original data set, such that one mode of the new data set contains independent variables. One way to quantify independence of a group of variables is via joint cumulants; if they vanish (or are zero), then we can conclude that the variables are independent [31]. In this chapter, we describe how to find cumulants for univariate and multivariate random variables.

B.2 Overview of Cumulants

The mean and variance of a random variable are its first and second order cumulants, respectively. From elementary statistics books, we know that a Normal distribution is exactly determined by its mean and variance. As a result, for normally distributed variables, we say that only second order statistics are needed to quantify the distribution, and higher order cumulants[1] vanish (or are zero) [33].[2]

For non-normally distributed variables, we need to know all of their non-zero cumulants to determine the joint multivariate distribution correctly. For a set of variables, the cumulant of order one is a vector (or an array with one mode)

[1] As higher order moments need not vanish for independent variables, we will use cumulants to define the distribution [15].

[2] For instance, the third order cumulant, or skewness, is zero for a Normal distribution because it is symmetric; and its fourth order cumulant, or kurtosis, usually measured in deviations from the Normal distribution, hence is also zero.

of variable means; the cumulant of order two is a matrix (or an array with two modes) with variables' variances and covariances; the third order cumulant measures skewness for the variables and is an array with three modes; the fourth order cumulant measures kurtosis and is a four more array; and so on. We can see that the order of the cumulant dictates the number of modes it has and its complexity. For practical purposes, we work with third order cumulants (of standardized variables) in the AICA framework for ease of derivation, computational efficiency and interpretation.

In general, we can think of the relationship between moments and cumulants as correlation and covariance matrices; that is, both contain the same information and can be used to express one in terms of the other [33]. However, because of certain properties of cumulants (discussed below), they are more preferable than moments to work with [33]. Denote the cumulant array by $\mathcal{C}$, with elements c_{ijk} for those of third order. While cumulants and moments share the following properties:

a). are not affected by Gaussian noise for cumulants of order greater than two [16],

b). are equivalent for any permutation of the variables [16], [57]:

$$c_{ijk} = c_{ikj} = c_{jik} = c_{jki} = c_{kij} = c_{kji};$$

c). and are multilinear [16], [15]:

$$\text{Cumulant}\left(X_1, X_2, AX_3, X_4, \ldots\right) = A \cdot \text{Cumulant}\left(X_1, X_2, X_3, X_4, \ldots\right),$$

it is the additional characteristics specific to cumulants, that make them desirable optimization criteria.[3] The ones most relevant to the methodology discussed in the body of the thesis are as follows:

[3]Please see [15] for more details on cumulants that are beyond the scope of the thesis.

d). they vanish in odd cumulants for variables with even probability distributions [16];

e). they vanish in even cumulants for variables with odd probability distributions [16]; and

f). are diagonal for independent variables [16]. That is, if $X_1, X_2, \ldots, X_n$ are independent of $Y_1, Y_2, \ldots, Y_m$, then [16]:

$$\text{Cumulant}\left(X_1, X_2, \ldots, X_n, Y_1, Y_2, \ldots, Y_m\right) = 0.$$

Point (f) suggests that if we diagonalize the joint cumulants of the data array, we will find the independent components.[4] We return to this notion in the formulation of AICA in Chapter 3.

B.3 Theoretical Cumulants

The cumulants are found from the Cumulant Generating Function, which is related to the Moment Generating Function. The multivariate Moment Generating Function (MGF) is [16]:

$$M_{\mathbf{X}}(\mathbf{t}) = E\left(e^{\mathbf{t}^T\mathbf{X}}\right), \tag{B.1}$$

and the Cumulant Generating Function (CGF) is:

$$\Psi_{\mathbf{X}}(\mathbf{t}) = \log M_{\mathbf{X}}(\mathbf{t}). \tag{B.2}$$

By expanding the Cumulant Generating Function via Taylor's Series around zero and finding the roots of the polynomial, we can find the expressions for the cumulants in terms of moments. Notice that the CGF is a composition of two functions

[4]For instance, in the case of two variables, the variance-covariance matrix is the variables' second joint cumulant; their marginal second order cumulants are the univariate variances. If we use PCA to find two Principal Components (PCs) from the data (not performing dimension reduction), then the covariance of the PCs will be zero and they will be uncorrelated. If the original variables were also normally distributed, then the PCs would also be independent of each other.

[17]:

$$f = \log(\mathbf{Y}) \text{ and } \mathbf{Y} = g(\mathbf{t}) = M_{\mathbf{X}}(\mathbf{t}). \tag{B.3}$$

We will use this relationship between the functions to derive the formula for the univariate cumulants in terms of moments in the next section. We present its multivariate analog in Appendix B.3.2.

B.3.1 Univariate Case

To find univariate cumulants in terms of moments, we first derive the equation to do so, and apply it to find the third order univariate cumulant. We close the section by introducing a simplified formulation of the relationship between cumulants and moments for standardized variables.

B.3.1.1 Derivation

To find the formula for the n^{th} univariate cumulant in terms of moments, we use the MGF:

$$\mu_n = \frac{d^n}{dt^n} M_X(0) = E[X^n]. \tag{B.4}$$

We begin the derivation by stating the univariate analog of Equation B.3, as it directly relates cumulants and moments:

$$f = \log(y) \text{ and } y = g(t) = M_X(t). \tag{B.5}$$

We proceed to expand the function f via a Taylor's series around zero. To do so, we need to differentiate a composition of two functions. We can either use the chain rule or a shortcut called Faà di Bruno's formula [49], which tells us that [65]:

$$\begin{aligned} \kappa_n = \frac{d^n}{dt^n} f\Big(g(t)\Big) = \sum_k \frac{n!}{m_1!(1!)^{m_1} m_2!(2!)^{m_2} \dots m_n!(n!)^{m_n}} \times \\ f^{(m_1+m_2+\dots+m_n)}\Big(g(t)\Big) \times \prod_{j=1}^{n} \Big(g^{(j)}(t)\Big)^{m_j}, \end{aligned} \tag{B.6}$$

where we sum over all possible non-negative solutions to the following equations:

$$\sum_{i=1}^{n} m_i = k \tag{B.7}$$

and

$$\sum_{i=1}^{n} i m_i = n, \tag{B.8}$$

and n is the order of the cumulant [49].

Equation B.6 shows us that we will be taking the derivatives of f, which in our case, is the logarithm function. The k^{th} univariate derivative of the logarithm function is:

$$f^{(k)}(y) = \log^{(k)}(y) = \frac{(-1)^{k-1}(k-1)!}{y^k}$$

and evaluating it at 0 yields:

$$f^{(k)}\Big(M_x(t)\Big)\Big|_{x=0} = \frac{(-1)^{k-1}(k-1)!}{\left[E\left(e^0\right)\right]^k} = (-1)^{k-1}(k-1)!. \tag{B.9}$$

Substituting Equations B.4 and B.9 into Equation B.6, we get:

$$\begin{aligned}
\kappa_n &= \frac{d^n}{dt^n}\log\Big(M_X(t)\Big)\Big|_{t=0} \\
&= \sum_k \frac{n!}{m_1!(1!)^{m_1} m_2!(2!)^{m_2}\ldots m_n!(n!)^{m_n}} \times \\
&\qquad \underbrace{\log^{(k)}\Big(M_X(t)\Big)\Big|_{t=0}}_{(-1)^{k-1}(k-1)!} \prod_{j=1}^{n} \Big(\underbrace{g^{(j)}(t)\Big|_{t=0}}_{=\mu_j}\Big)^{m_j} \\
&= \sum_k \binom{n}{m_1, m_2, \ldots, m_n} \frac{(-1)^{k-1}(k-1)!}{(1!)^{m_1}(2!)^{m_2}\ldots(n!)^{m_n}} \times \mu_1^{m_1}\mu_2^{m_2}\cdots\mu_n^{m_n} \qquad \square
\end{aligned} \tag{B.10}$$

where the summation is over the sums defined in Equations B.7 and B.8 [49].

B.3.1.2 Equation Verification for a Third Order Cumulant

We verify Equation B.10 by finding the univariate skewness (or third order cumulant) denoted by κ_3, for a non-standardized random variable. To do so, we need to know n, k and m_i. We now present the details to finding these unknowns.

1. n is the order of the cumulant, which is three in this case.

2. To solve for m_i, expand $\sum_{i=1}^{n} i m_i = n$ to get $1m_1 + 2m_2 + 3m_3 = 3$. There are three possibilities where the relationships hold:

   (a) $m_3 = 1 \Rightarrow m_1 = m_2 = 0$

   (b) $m_1 = m_2 = 1 \Rightarrow m_3 = 0$

   (c) $m_1 = 3 \Rightarrow m_2 = m_3 = 0$.

3. Next, substitute the three scenarios from Step (2) into $\sum_{i=1}^{3} m_i = k$ to find the value for k in each case:

   (a) $k = m_1 + m_2 + m_3 = 0 + 0 + 1 = 1$

   (b) $k = m_1 + m_2 + m_3 = 1 + 1 + 0 = 2$

   (c) $k = m_1 + m_2 + m_3 = 3 + 0 + 0 = 3$

4. For each k, each term in Equation B.10 will look like:

   (a) $k = 1$:
   $$\binom{3}{0,0,1} \frac{(-1)^{1-1}(1-1)!}{(1!)^0(2!)^0(3!)^1} \times \mu_1^0 \mu_2^0 \mu_3^1 = \mu_3$$

   (b) $k = 2$:
   $$\binom{3}{1,1,0} \frac{(-1)^{2-1}(2-1)!}{(1!)^1(2!)^1(3!)^0} \times \mu_1^1 \mu_2^1 \mu_3^0 = -3\mu_1\mu_2$$

   (c) $k = 3$:
   $$\binom{3}{3,0,0} \frac{(-1)^{3-1}(3-1)!}{(1!)^3(2!)^0(3!)^0} \times \mu_1^3 \mu_2^0 \mu_3^0 = 2\mu_1^3$$

5. Combining terms from Steps 4(a)–(c), we get the desired result:

   $$\kappa_3 = \mu_3 - 3\mu_2\mu_1 + 2\mu_1^3. \qquad \square$$

.

B.3.1.3 Finding Cumulants from Moments

In the previous section we found the univariate third order cumulant in terms of moments. Expanding Equation B.10 similarly for orders one, two and four, we can express the first four cumulants in terms of moments:

$$\kappa_1 = \mu_1 \tag{B.11}$$

$$\kappa_2 = \mu_2 - \mu_1^2 \tag{B.12}$$

$$\kappa_3 = \mu_3 - 3\mu_1\mu_2 + 2\mu_1^3 \tag{B.13}$$

$$\kappa_4 = \mu_4 + 4\mu_1\mu_3 - 3\mu_2^2 + 12\mu_1\mu_2^3 - 6\mu_1^4. \tag{B.14}$$

For a standardized univariate random variable, the expressions for the cumulants simplify; the first four appear below:

$$\kappa_1 = 0 \tag{B.15}$$

$$\kappa_2 = 1 \tag{B.16}$$

$$\kappa_3 = \mu_3 \tag{B.17}$$

$$\kappa_4 = \mu_4 - 3. \tag{B.18}$$

B.3.2 Multivariate Case

In this section we present the multivariate analog of Equation B.10 (omitting the proof) and illustrate how to use the formula.

B.3.2.1 Finding Cumulants from Moments

The multivariate analog of Equation B.10 shows us how to find all of the multivariate cumulants of order D in terms of moments [16]:

$$\kappa_{[D]} = \sum_{\nu} (-1)^{|\nu|-1}(|\nu|-1)!\, \mathbb{E}\left\{\prod_{i\in V_1} x_i\right\} \mathbb{E}\left\{\prod_{i\in V_2} x_i\right\} \dots\ \mathbb{E}\left\{\prod_{i\in V_L} x_i\right\}, \tag{B.19}$$

where ν represents different partitions or (different) group sizes that, say, k people can arrange themselves into, where each group $\{V_1, V_2, \dots, V_L\}$, for $1 \leq L \leq k$ is

composed of at least one person and the number of blocks $|\nu|$ tells us how many groups there are ($|\nu| = L$). It is easier to understand the formula via an example, which we present in the following section.

B.3.2.2 Equation Verification

To better understand Equation B.19, suppose we wish to find the expression for the fourth order multivariate cumulant $\kappa_{[4]}$ for four random vectors X_1, X_2, X_3 and X_4. Because cumulants capture all possible relationships between the variables, we can think of the variables as four people given instructions to arrange themselves into any size group they wanted. Then the only possible group sizes are: a group of four, four groups of one, two groups of two each, or two groups with three people in one and one person in the other group. We discuss these scenarios (or partitions) in the context of Equation B.19 below, in order to find the joint fourth order cumulant for a matrix with four random variables (represented here by people).

- In the first scenario, everyone chose to work together and there is only one group of four. In the notation of Equation B.19, we see that there is one block (blocks are separated by commas) and one partition of size four (partitions are separated by parentheses):

  $$\{V_1 = (X_1X_2X_3X_4)\}.$$

  Plugging this information into Equation B.19, we get the first term in the formula for $\kappa_{[4]}$:

  $$(1-1)!(-1)^{1-1}E[X_1X_2X_3X_4] = E[X_1X_2X_3X_4] = \mu_{[4]}.$$

- Suppose that next we look at the different ways that four people can be arranged into two groups, with three people in one and one person in the other group. There are $\binom{4}{3} = 4$ different ways that four people can be selected to be a member of a group of three. They are: $X_1X_2X_3$, $X_1X_2X_4$,

$X_1X_3X_4$ and $X_2X_3X_4$. This means that the remaining person is in a group by him/herself. Thus, we get four blocks of two partitions each:

$$\{V_1 = (X_1X_2X_3)(X_4),\ V_2 = (X_1X_2X_4)(X_3),$$

$$V_3 = (X_1X_3X_4)(X_2),\ V_4 = (X_2X_3X_4)(X_1)\}.$$

As a result, for this set, the associated term in the summation in Equation B.19 reduces to:

$$(2-1)!(-1)^{2-1}E\left[X_1X_2X_3\right]E[X_4] + \cdots$$

$$+(2-1)!(-1)^{2-1}E[X_2X_3X_4]\,E[X_1] = 4\mu_{[1]}\mu_{[3]}.$$

- Another way to have a set with two partitions each, is if there are two groups with two members each. There are $\binom{4}{2} = 6$ different ways that four people can be selected to be a member of a group of two. However, since the order of a group the person belongs to does not matter, there are only three ways we can choose two variables out of four, for two groups (without repeats and up to permutation):

$$\{V_1 = (X_1X_2)(X_3X_4),\, V_2 = (X_1X_4)(X_2X_3),\, V_3 = (X_1X_3)(X_2X_4)\}.$$

As such, there are three blocks of two partitions each, suggesting that the next term in the summation in Equation B.19 reduces to:

$$\begin{aligned}(2-1)!(-1)^{2-1}E[X_1X_2]\,E[X_3X_4] + (2-1)!(-1)^{2-1}E[X_1X_4]\,E[X_2X_3]\\ + (2-1)!(-1)^{2-1}E[X_1X_3]\,E[X_2X_4] =\\ = -\Big(E[X_1X_2]\,E[X_3X_4]+E[X_1X_4]\,E[X_2X_3]+E[X_1X_3]\,E[X_2X_4]\Big) = -3\mu_{[2]}^2.\end{aligned} \tag{B.20}$$

- Four people can also work in three groups: one of size two and two of one person each. From Case (3), we saw that there are six ways that we can

have four people arrange themselves into groups of two. We can break up the second group of two people into two groups of one to get three people to work individually. This gets us six blocks of three partitions each:

$$\{V_1 = (X_1X_2)(X_3)(X_4), V_2 = (X_1X_4)(X_2)(X_3), V_3 = (X_1X_3)(X_2)(X_4),$$
$$V_4 = (X_2X_4)(X_1)(X_3), V_5 = (X_2X_3)(X_1)(X_4), V_6 = (X_3X_4)(X_1)(X_2)\}, \tag{B.21}$$

which corresponds to the following term in Equation B.19:

$$(3-1)!(-1)^{3-1}E[X_1X_2]\,E[X_3]\,E[X_4] + \cdots$$
$$+(3-1)!(-1)^{3-1}E[X_3X_4]\,E[X_1]\,E[X_2] = 12\mu_{[1]}^2\mu_{[2]}.$$

- Our last scenario is the case when all four people decide to work by themselves. Then there is only one block with four partitions. This is the final term in the summation in Equation B.19:

$$(4-1)!(-1)^{4-1}E[X_1]\,E[X_2]\,E[X_3]\,E[X_4] =$$
$$6\,E[X_1]\,E[X_2]\,E[X_3]\,E[X_4] = -6\mu_{[1]}^4.$$

Combining all the terms together, we get all the possible scenarios that we can get information out of four variables, and form the joint fourth order cumulant:

$$\kappa_{[4]} = \mu_{[4]} + 4\mu_{[1]}\mu_{[3]} - 3\mu_{[2]}^2 + 12\mu_{[1]}^2\mu_{[2]} - 6\mu_{[1]}^4. \qquad \square$$

B.4 Empirical Cumulants

In the previous section we discussed theoretical moments and cumulants. We now present how to use the equations to estimate moments and cumulants from the data.

B.4.1 Univariate Case

In the univariate case, we can find the k^{th} moment of a random variable X (represented here as a column vector) as follows:

$$m_k = \frac{1}{N} \sum_{i=1}^{N} x_i^k,$$

where x_i are the N elements of a column vector X.

B.4.2 Multivariate Case

For multivariate variables, we illustrate how to find the third and fourth order joint cumulants from the data; cumulants of higher order can be found similarly.

Let $\mathcal{C}$ correspond to the empirical higher order cumulant array (order higher than two) of a standardized data matrix $\mathbf{X}_{N \times M}$. We can find each element of the third order cumulant array $M \times M \times M$ as follows:

$$c_{ijk} = \sum_{m=1}^{M} x_{im} x_{jm} x_{km}, \tag{B.22}$$

where M the number of variables (or columns) in matrix $\mathbf{X}$. The other terms are omitted because the variables are assumed to be standardized.

Similarly, we can find the elements of the fourth order joint cumulant array $M \times M \times M \times M$ as follows:

$$c_{ijkl} = \sum_{m=1}^{M} x_{im} x_{jm} x_{km} x_{lm} - 3 \sum_{p=1}^{M} x_{ip} x_{jp} \sum_{q=1}^{M} x_{kq} x_{lq}. \tag{B.23}$$

The other terms are omitted because the variables are assumed to be standardized.

APPENDIX C

Derivations of Alternating Least Squares Algorithms for Dimension Reduction Techniques Discussed in the Thesis

C.1 Introduction

In this section we define and motivate the need for contrast functions in the formulation of data decompositions, and proceed to derive algorithms for finding the unknown loading matrices and core arrays (where applicable) in the formulations of contrast functions for PCA, Tucker-3, CP and AICA decompositions.

C.2 Overview of Contrast Functions

Since we are interested in the underlying unobserved or latent variables, we cannot evaluate model fit via a Mean Squared Error criterion [16].[1] As a result, we turn to contrast functions as an alternative optimization criteria for determining how the sources were mixed and how to separate them via consistent estimates [16]. Inherent to the optimization criteria discussed in this section is the assumption of a noise-free model for ICA [16], [15].[2]

[1] Mean Squared Error Criterion (MSE) is found as the $E[(\text{estimated sources} - \text{true sources})^2]$. Because the independent components are, by definition, unobserved, we cannot use MSE to evaluate model fit.

[2] A noise-free ICA model can be formulated mathematically as $\mathbf{Z} = \mathbf{SA}$. This assumption is equivalent to assuming Gaussian noise in the ICA model because higher order cumulants (for order greater than two) vanish for Gaussian random variables [33]. (Please see Appendix B for more details on Gaussian random variables.)

To be a contrast function, the optimization criteria has to satisfy the following properties:

- should not be affected by a permutation of the components [14];
- be scale invariant [14];
- and decrease the contrast function if sources were independent to begin with [15], [14].

We now discuss how to optimize the contrast function of PCA and its higher mode analogs Tucker-3 and CP introduced in Chapter 2.

C.3 Principal Component Analysis

The goal of PCA is to express the observed, preprocessed data set $\mathbf{X}_{NM\times K}$ as a product of two orthonormal matrices $\mathbf{U}$ and $\mathbf{V}$. To find the unknown matrices, we minimize the following loss function [19]:

$$\begin{aligned}\min_{\substack{\mathbf{U}^T\mathbf{U}\text{ diagonal}\\ \mathbf{V}^T\mathbf{V}\text{ diagonal}}} ||\mathbf{X}-\mathbf{U}\mathbf{V}^T||^2 = \\ &= \min\left\{\text{tr}\left[\left(\mathbf{X}-\mathbf{U}\mathbf{V}^T\right)^T\left(\mathbf{X}-\mathbf{U}\mathbf{V}^T\right)\right]\right\} \\ &= \min \text{tr}\Big(\mathbf{X}^T\mathbf{X}-\mathbf{X}^T\mathbf{U}\mathbf{V}^T-\mathbf{V}\mathbf{U}^T\mathbf{X}-\mathbf{V}\mathbf{U}^T\mathbf{U}\mathbf{V}^T\Big).\end{aligned} \tag{C.1}$$

Using the properties of trace [72], we can rearrange the loss function in Equation C.1 as follows:

$$L_1 = \text{tr}\Big(\mathbf{X}^T\mathbf{X}-\mathbf{V}\mathbf{U}^T\mathbf{X}-\mathbf{V}\mathbf{U}^T\mathbf{X}-\mathbf{V}\mathbf{U}^T\mathbf{U}\mathbf{V}^T\Big) \tag{C.2}$$

and

$$L_2 = \text{tr}\Big(\mathbf{X}^T\mathbf{X}-\mathbf{U}\mathbf{V}^T\mathbf{X}^T-\mathbf{V}^T\mathbf{X}^T\mathbf{U}-\mathbf{U}\mathbf{V}^T\mathbf{V}\mathbf{U}^T\Big). \tag{C.3}$$

Taking the derivative of Equation C.2 with respect to the unknown matrix $\mathbf{V}$, and employing the property of derivatives with respect to the trace function [21]:

$$\frac{\partial}{\partial \mathbf{V}} \text{trace}\left(\mathbf{V}\mathbf{U}\right) = \mathbf{U}^T, \tag{C.4}$$

we get one of the stationary equations [19]:

$$\begin{aligned} \frac{\partial L_1}{\partial \mathbf{V}} &= -2\left(\mathbf{U}^T\mathbf{X}\right)^T + \left(\mathbf{U}^T\mathbf{U}\mathbf{V}^T\right)^T + \left(\mathbf{V}\mathbf{U}^T\mathbf{U}\right) = 0 \\ &= -2\left(\mathbf{X}^T\mathbf{U}\right) + 2\left(\mathbf{V}\mathbf{U}^T\mathbf{U}\right) = 0 \\ &\Rightarrow \mathbf{X}^T\mathbf{U} = \mathbf{V}\left(\mathbf{U}^T\mathbf{U}\right). \end{aligned} \tag{C.5}$$

Taking the derivative of Equation C.3 with respect to the unknown matrix $\mathbf{U}$, and employing the property of derivatives with respect to the trace function shown in Equation C.4 [21], we get the second stationary equation [19]:

$$\begin{aligned} \frac{\partial L_2}{\partial \mathbf{U}} &= -2\left(\mathbf{V}^T\mathbf{X}^T\right)^T + \left(\mathbf{V}^T\mathbf{V}\mathbf{U}^T\right)^T + \left(\mathbf{U}\mathbf{V}^T\mathbf{V}\right) = 0 \\ &= -2\left(\mathbf{X}\mathbf{V}\right) + 2\left(\mathbf{U}\mathbf{V}^T\mathbf{V}\right) = 0 \\ &\Rightarrow \mathbf{X}\mathbf{V} = \mathbf{U}\left(\mathbf{V}^T\mathbf{V}\right). \end{aligned} \tag{C.6}$$

Because we required $\mathbf{U}^T\mathbf{U}$ and $\mathbf{V}^T\mathbf{V}$ to be diagonal, this is equivalent to [19]:

$$\mathbf{X}^T\mathbf{U} = \mathbf{V}\mathbf{\Lambda} \tag{C.7}$$

$$\mathbf{X}\mathbf{V} = \mathbf{U}\mathbf{\Lambda} \tag{C.8}$$

As such, we find $\mathbf{U}$, $\mathbf{V}$ and $\mathbf{\Lambda}$ by decomposing $\mathbf{X}$ without error by Singular Value Decomposition: $\mathbf{X} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^T$, where $\mathbf{U}$ contains K left singular vectors; $\mathbf{V}$ contains K right singular vectors; $\mathbf{U}^T\mathbf{U} = \mathbf{I}_K$, $\mathbf{V}\mathbf{V}^T = \mathbf{I}_{NM}$ as desired; and $\mathbf{\Lambda}$ is a diagonal matrix that stores the singular values.

As a result of SVD, the loss function of Equation C.1 simplifies to [19]:

$$\min_{\substack{\mathbf{U}^T\mathbf{U} \text{ diagonal} \\ \mathbf{V}^T\mathbf{V} \text{ diagonal}}} ||\mathbf{X} - \mathbf{U}\mathbf{V}^T||^2 = \min \sum_{i=L+1}^{K} \lambda_i^2. \tag{C.9}$$

That is, we want to minimize the information in the data set left over after extracting L components.

C.4 Tucker Decomposition

In the formulation of the loss function of the Tucker-3 decomposition, we have four unknowns: three loading matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ and one core array $\mathcal{G}$. We present one way to simplify the loss function: eliminating $\mathcal{G}$ from the model [40]. The derivations of the associated loss function and ALS algorithm follow the logic of [40].

C.4.1 Derivation

We begin the derivation of the simplified loss function for the Tucker-3 decomposition by expressing its loss function as a way of minimizing the error in the approximation of our data set:

$$\begin{aligned}
\inf_{\substack{||\mathbf{A}||=\mathbf{I}_P \\ ||\mathbf{B}||=\mathbf{I}_Q \\ ||\mathbf{C}||=\mathbf{I}_R \\ \mathcal{G}}} ||\mathcal{X} - \hat{\mathcal{X}}||^2 &= \inf_{\substack{||\mathbf{A}||=\mathbf{I}_P \\ ||\mathbf{B}||=\mathbf{I}_Q \\ ||\mathbf{C}||=\mathbf{I}_R \\ \mathcal{G}}} ||\mathcal{X} - \mathcal{G} \times_1 \mathbf{A} \times_2 \mathbf{B} \times_3 \mathbf{C}||^2 \\
&= \inf_{\substack{||\mathbf{A}||=\mathbf{I}_P \\ ||\mathbf{B}||=\mathbf{I}_Q \\ ||\mathbf{C}||=\mathbf{I}_R}} ||\mathcal{X}||^2 - 2 < \mathcal{X}, \mathcal{G} \times_1 \mathbf{A} \times_2 \mathbf{B} \times_3 \mathbf{C} > \\
&\qquad + ||\mathcal{G} \times_1 \mathbf{A} \times_2 \mathbf{B} \times_3 \mathbf{C}||^2 \\
&= \inf_{\substack{||\mathbf{A}||=\mathbf{I}_P \\ ||\mathbf{B}||=\mathbf{I}_Q \\ ||\mathbf{C}||=\mathbf{I}_R \\ \mathcal{G}}} ||\mathcal{X}||^2 - 2 < \mathcal{X} \times_1 \mathbf{A}^T \times_2 \mathbf{B}^T \times_3 \mathbf{C}^T, \mathcal{G} > \\
&\qquad + ||\mathcal{G} \times_1 \mathbf{A} \times_2 \mathbf{B} \times_3 \mathbf{C}||^2 \\
&= \inf_{\substack{||\mathbf{A}||=\mathbf{I}_P \\ ||\mathbf{B}||=\mathbf{I}_Q \\ ||\mathbf{C}||=\mathbf{I}_R \\ \mathcal{G}}} ||\mathcal{X}||^2 - 2||\mathcal{G}||^2 + ||\mathcal{G} \times_1 \mathbf{A} \times_2 \mathbf{B} \times_3 \mathbf{C}||^2 \\
&= \inf_{\mathcal{G}} ||\mathcal{X}||^2 - ||\mathcal{G}||^2
\end{aligned} \tag{C.10}$$

where the:

- second equality follows from the definition of the norm;
- third equality is due to Proposition 3.11 of [40], which states that the n-mode

product is commutative in the inner product;

- fourth equality is by definition of $\mathcal{G}$. That is, if

$$\mathcal{X} = \mathcal{G} \times_1 \mathbf{A} \times_2 \mathbf{B} \times_3 \mathbf{C}$$

then

$$\begin{aligned} \mathcal{G} &= \mathcal{X} \times_1 \mathbf{A}^{-1} \times_2 \mathbf{B}^{-1} \times_3 \mathbf{C}^{-1} \\ &= \mathcal{X} \times_1 \mathbf{A}^{T} \times_2 \mathbf{B}^{T} \times_3 \mathbf{C}^{T}, \end{aligned} \tag{C.11}$$

because each matrix is orthonormal, its matrix inverse is equivalent to its matrix transpose.

- fifth equality is due to Proposition 3.12 of [40], which states that multiplying an array in a particular mode by an orthogonal matrix leaves its norm unchanged [40].

This means that to minimize Equation C.10, we need to maximize $||\mathcal{G}||^2$ [40]. By making $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ larger and larger (to infinity), we can maximize the norm [46]. To have meaningful matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$, we constrain them to be orthonormal. Hence, the optimization becomes:

$$\sup ||\mathcal{G}||^2 = \sup_{\substack{||\mathbf{A}||=\mathbf{I}_P \\ ||\mathbf{B}||=\mathbf{I}_Q \\ ||\mathbf{C}||=\mathbf{I}_R}} ||\mathcal{X} \times_1 \mathbf{A}^T \times_2 \mathbf{B}^T \times_3 \mathbf{C}^T||^2. \tag{C.12}$$

C.4.2 Alternating Least Squares Algorithm Algorithm

To find the unknown loading matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$, we proceed to use ALS. We iterate by fixing all of the unknowns, one at a time except for one, and updating it from the others until convergence. Suppose $\mathbf{B}$ and $\mathbf{C}$ are fixed. Then, to update

$\mathbf{A}$, we need to find:

$$\begin{aligned}
\sup ||\mathcal{G}||^2 &= \sup_{||\mathbf{A}||=\mathbf{I}_P} ||\mathcal{X} \times_1 \mathbf{A}^T \times_2 \mathbf{B}^T \times_3 \mathbf{C}^T||^2 \\
&= \sup_{||\mathbf{A}||=\mathbf{I}_P} \left|\left|\left(\mathcal{X} \times_1 \mathbf{I} \times_2 \mathbf{B}^T \times_3 \mathbf{C}^T\right) \times_1 \mathbf{A}^T\right|\right|^2 \\
&= \sup_{||\mathbf{A}||=\mathbf{I}_P} ||\mathcal{Y} \times_1 \mathbf{A}^T||^2 \qquad \text{(C.13)} \\
&= \sup_{||\mathbf{A}||=\mathbf{I}_P} ||\mathbf{A}^T\mathbf{Y}_{(1)}||^2 \\
&= \sup_{||\mathbf{A}||=\mathbf{I}_P} \left(\mathbf{A}^T\mathbf{Y}_{(1)}\right)^T \left(\mathbf{A}^T\mathbf{Y}_{(1)}\right).
\end{aligned}$$

Via Singular Value Decomposition, let $\mathbf{Y}_{(1)} = \mathbf{U}\mathbf{D}\mathbf{V}^T$. Then Equation C.13, becomes:

$$\begin{aligned}
\sup ||\mathcal{G}||^2 &= \sup_{||\mathbf{A}||=\mathbf{I}_P} \left(\mathbf{A}^T\mathbf{Y}_{(1)}\right)^T \left(\mathbf{A}^T\mathbf{Y}_{(1)}\right) \\
&= \left(\mathbf{V}\mathbf{D}\mathbf{U}^T\mathbf{A}\right)\left(\mathbf{A}^T\mathbf{U}\mathbf{D}\mathbf{V}^T\right) \\
&= \mathbf{V}\mathbf{D}\left(\mathbf{U}^T\mathbf{A}\right)\left(\mathbf{A}^T\mathbf{U}\right)\mathbf{D}\mathbf{V}^T \\
&= \mathbf{V}\mathbf{D}\left(e_1e_1^T + e_2e_2^T + \cdots + e_Pe_P^T\right)\mathbf{D}\mathbf{V}^T \qquad \text{(C.14)} \\
&= \left(\sum_{i=1}^{P} \lambda_i^2 e_i e_i\right)\mathbf{V}\mathbf{V}^T \\
&= \sum_{i=1}^{P} \lambda_i^2
\end{aligned}$$

That is, we want to maximize the information contained in the P components by choosing the P largest singular values.

This means that the ALS algorithm for finding the optimal loading matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ (by symmetry of the arguments) is:

1. Fix $\mathbf{B}^{(t)}$ and $\mathbf{C}^{(t)}$. Then $\mathbf{A}^{(t+1)} \longleftarrow 1^{st}$ leading P eigenvectors of

$$\mathcal{X} \times_1 \mathbf{I} \times_2 \mathbf{B}^{T(t)} \times_3 \mathbf{C}^{T(t)} = \mathcal{X} \times_2 \mathbf{B}^{T(t)} \times_3 \mathbf{C}^{T(t)} \qquad \text{(C.15)}$$

2. Fix $\mathbf{A}^{(t+1)}$ and $\mathbf{C}^{(t)}$. Then $\mathbf{B}^{(t+1)} \longleftarrow 1^{st}$ leading Q eigenvectors of

$$\mathcal{X} \times_1 \mathbf{A}^{T(t+1)} \times_3 \mathbf{C}^{T(t)} \tag{C.16}$$

3. Fix $\mathbf{A}^{(t+1)}$ and $\mathbf{B}^{(t+1)}$. Then $\mathbf{C}^{(t+1)} \longleftarrow 1^{st}$ leading R eigenvectors of

$$\mathcal{X} \times_2 \mathbf{A}^{T(t+1)} \times_3 \mathbf{B}^{T(t+1)}. \tag{C.17}$$

We iterate over these steps to find the optimal loading matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$, until a desired level of tolerance is achieved:

$$\left| ||\mathcal{X}||^2 - ||\mathcal{G}||^2 \right| < \epsilon.$$

C.5 CANDECOMP/PARAFAC Decomposition

In the formulation of the loss function of the CP decomposition, we have three unknowns: the three loading matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$. We present the derivation for the ALS algorithm for the CP decomposition following the logic of [40].

C.5.1 Derivation

Similar to Tucker-3, we begin the derivation of the ALS formulation for CP by expressing its loss function as a way of minimizing the error in the approximation of our data set:

$$\inf_{\mathbf{A},\mathbf{B},\mathbf{C}} ||\mathcal{X} - \hat{\mathcal{X}}||^2 = \inf_{\mathbf{A},\mathbf{B},\mathbf{C}} ||\mathbf{X}_{(1)} - \mathbf{A}\left(\mathbf{C} \odot \mathbf{B}\right)^T||^2, \tag{C.18}$$

where the loading matrices are stored in $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$. This means that at infimum,

$$\mathbf{X}_{(1)} = \mathbf{A}\left(\mathbf{C} \odot \mathbf{B}\right)^T$$

and

$$\mathbf{X}_{(1)}^T = \left(\mathbf{C} \odot \mathbf{B}\right)\mathbf{A}^T.$$

Let $Y = \text{vec}\left(\mathbf{X}_{(1)}^T\right)$, $\mathbf{Z} = (\mathbf{C} \odot \mathbf{B})$ and $\beta = \text{vec}\left(\mathbf{A}^T\right)$, where vec$(\cdot)$ results in one column with NM elements. Then,

$$Y = \mathbf{Z}\beta.$$

Multiplying both sides by $\mathbf{Z}^T$, we get:

$$\mathbf{Z}^T Y = \mathbf{Z}^T \mathbf{Z}\beta.$$

Isolating β, we get the familiar regression equation:

$$\beta = \left(\mathbf{Z}^T\mathbf{Z}\right)^{-1} \mathbf{Z}^T Y.$$

Substituting back our variables, we get:

$$\mathbf{A}^T = \left((\mathbf{C} \odot \mathbf{B})^T (\mathbf{C} \odot \mathbf{B})\right)^{\dagger} (\mathbf{C} \odot \mathbf{B})^T \mathbf{X}_{(1)}^T,$$

where † denotes the generalized Moore-Penrose inverse. Using the properties of the Khatri-Rao product [40], the equation simplifies as follows:

$$\mathbf{A}^T = \left(\left(\mathbf{C}^T\mathbf{C}\right) * \left(\mathbf{B}^T\mathbf{B}\right)\right)^{\dagger} (\mathbf{C} \odot \mathbf{B})^T \mathbf{X}_{(1)}^T. \tag{C.19}$$

Transposing both sides, we get:

$$\mathbf{A} = \mathbf{X}_{(1)} (\mathbf{C} \odot \mathbf{B}) \left(\left(\mathbf{C}^T\mathbf{C}\right) * \left(\mathbf{B}^T\mathbf{B}\right)\right)^{\dagger}; \tag{C.20}$$

we drop the transpose on the last term because it is symmetric.

C.5.2 Alternating Least Squares Algorithm Algorithm

To find the unknown matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ via ALS, we use Equation C.20 and the symmetry of the arguments in the loss function to specify the loading matrix updates:

$$\mathbf{A}^{(t+1)} \longleftarrow \mathbf{X}_{(1)} (\mathbf{C} \odot \mathbf{B}) \left(\left(\mathbf{C}^T\mathbf{C}\right) * \left(\mathbf{B}^T\mathbf{B}\right)\right)^{\dagger}$$

$$\mathbf{B}^{(t+1)} \longleftarrow \mathbf{X}_{(1)} (\mathbf{C} \odot \mathbf{A}) \left(\left(\mathbf{C}^T\mathbf{C}\right) * \left(\mathbf{A}^T\mathbf{A}\right)\right)^{\dagger}$$

$$\mathbf{C}^{(t+1)} \longleftarrow \mathbf{X}_{(1)} (\mathbf{B} \odot \mathbf{A}) \left(\left(\mathbf{B}^T\mathbf{B}\right) * \left(\mathbf{A}^T\mathbf{A}\right)\right)^{\dagger}.$$

C.6 Array Independent Component Analysis Decomposition

The formulation of the AICA loss function contains four unknowns: the four loading matrices $\mathbf{U}$, $\mathbf{V}$, $\mathbf{W}$ and $\mathbf{D}$. It is based on the generalization of the CP decomposition to four mode arrays. We present the derivation for the ALS algorithm for the AICA decomposition following the logic of Section C.5.

C.6.1 Derivation

Similar to CP, we begin the derivation of the ALS formulation for AICA by expressing its loss function as a way of minimizing the error in the approximation of the four mode array $\mathcal{A}$ that contains the third order cumulants for each $N \times M$ slice of $\mathcal{X}$:

$$\inf_{\mathbf{U},\mathbf{V},\mathbf{W},\mathbf{D}} ||\mathcal{A} - \hat{\mathcal{A}}||^2 = \inf_{\mathbf{U},\mathbf{V},\mathbf{W},\mathbf{D}} ||\mathbf{A}_{(1)} - \mathbf{U}\left(\mathbf{D} \odot \mathbf{W} \odot \mathbf{V}\right)^T||^2, \tag{C.21}$$

where the loading matrices are stored in $\mathbf{U}$, $\mathbf{V}$, $\mathbf{W}$ and $\mathbf{D}$. This means that at infimum,

$$\mathbf{A}_{(1)} = \mathbf{U}\left(\mathbf{D} \odot \mathbf{W} \odot \mathbf{V}\right)^T.$$

and

$$\mathbf{A}_{(1)}^T = \left(\mathbf{D} \odot \mathbf{W} \odot \mathbf{V}\right)\mathbf{U}^T.$$

Following the derivation of the ALS for the CP decomposition above, we can derive the ALS updates shown in Chapter 3 for the AICA framework.

APPENDIX D

Application of Previously Developed Dimension Reduction Techniques to Study the White Oval

D.1 Introduction

We evaluate the success of a decomposition based on its ability to uncover interpretable latent structures in the remotely sensed images of the White Oval. The components extracted via AICA are meaningful and are discussed in Chapter 5. As the results of existing dimension reduction methods not interpretable, which is our main criteria, we present them in this chapter for completeness.

D.2 Results of Previously Developed Dimension Reduction Techniques

D.2.1 Principal Component Analysis

Principal Component Analysis is by far the most commonly employed dimension reduction technique, especially in remote sensing applications. We use it to reduce the volume of the data cube by shrinking the size of one mode. As with any dimension reduction technique, we first have to determine how many components to extract. For our data set, following the discussion in Chapter 4, we considered extracting three and four components based on solutions on the convex hull shown in Table D.1. The loading matrix $\mathbf{U}$ that will rotate the unfolded data cube to have uncorrelated components is shown in Figure D.1.

Given the preprocessed data set $\mathbf{Z}_{K\times K}$ and the loading matrix $\mathbf{U}$, we can find the first component:

$$\begin{aligned} PC_1 &= u_{11}Z_1 + u_{21}Z_2 + \cdots + u_{K1}Z_K \\ &= 0.2 \times Z_1 + 0.3 \times Z_2 + \ldots - 0.2Z_{K-1} - 0.1Z_K \end{aligned} \tag{D.1}$$

where Z_i, $i = 1, \ldots, 13$ is a column of the preprocessed data set, as outlined in Chapter 2.2.1. The second component is found by weighting the columns of $\mathbf{Z}$ by the second column of $\mathbf{U}$ shown in red in Figure D.1; the third component is found by weighting the columns of $\mathbf{X}$ by the third column of $\mathbf{U}$ shown in green. We find components from the loading matrices resulting from other decompositions similarly.

Rank	Complexity	Fit	Scree-test Value
1	3	0.89	4.60
2	4	0.94	2.57
3	11	1	1.85
4	10	0.99	1.68
5	6	0.97	1.64
		⋮	

Table D.1: Top five solutions on the convex hull for PCA.

Among all of the dimension reduction techniques presented in the thesis, PCA is the only one whose formulation for finding the components is deterministic and solutions are nested[1]. In addition, PCs are also ordered in terms of explaining the most variability. However, its major drawbacks are overfitting noise when higher

[1]Nested solutions are those, that if we compute the Singular Value Decomposition of the preprocessed data $\mathbf{X}_{NM\times K}$ (as outlined in Section 2.2.1) and find all K principal components, to find a solution that will reduce the depth of $\mathbf{X}$ to P ($P < M$), we choose those P columns of the complete solution that are associated with P largest singular values.

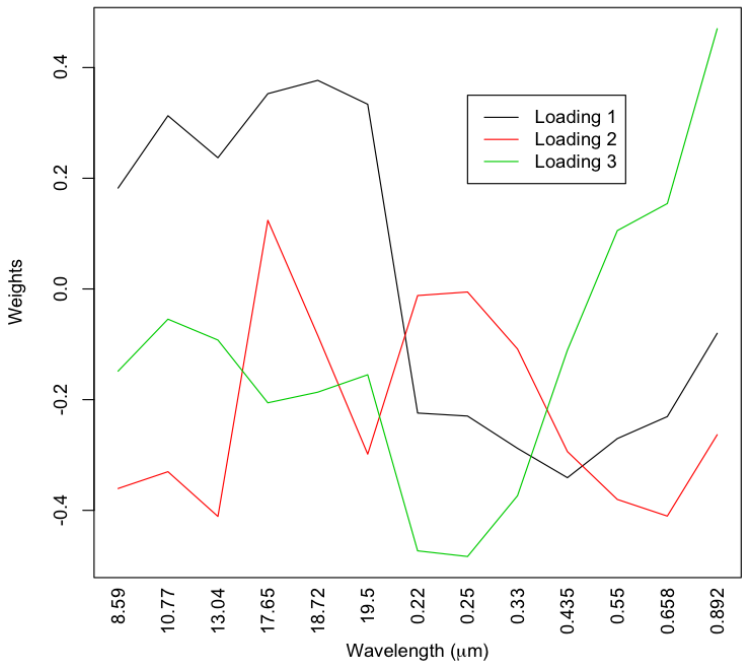


Figure D.1: Loading matrix from PCA, extracting three components that explain 89% of variance.

mode data sets are reshaped into matrices [37]; and assuming normally distributed variables. Based on Figure 5.4, which showed the distributions of the radiances across wavelengths, this is not the case for our data set. With these problems, we expect methods that relax either or both of these assumptions to be more robust.

D.2.2 Independent Component Analysis

Because we saw that the distribution of the original data cube containing the White Oval was not normal, we also tried to reduce its third mode via ICA. To do so, we decompose a two mode representation of the data cube as described in Chapter 2.2.2. In the process, ICA finds the un-mixing matrix $\mathbf{W}$ that rotates the Principal Components towards independence.

We begin ICA by choosing the number of components we want to extract. The

convex hull approach introduced in Chaper 4 suggests that ICA fits the data perfectly with eight components, as shown in Table D.2. Yet the un-mixing matrix [2] **W** for these components, shown in Figure D.2, illustrates that they are not informative, as the weights forming the components are multiples of each other and seem to mimic the variance in the data cube.[3,4] As a result, we conclude that, for this data set, ICA is not effective in providing inference about the White Oval on Jupiter. We now proceed to analyze the data set as a cube rather than a matrix.

Rank	Complexity	Fit	Scree-test Value
1	8	1	116.04
2	3	0.9992710	9.09
3	4	0.9998863	6.59
4	5	0.9999796	6.21
5	6	0.9999947	4.65
		⋮	

Table D.2: Top five solutions on the convex hull for ICA.

D.2.3 Tucker-3

One technique generalizing Singular Value Decomposition (SVD) to three mode data sets is Tucker-3 [41]. Similar to SVD, the components of Tucker-3 are also found to explain the variability in each mode, regardless of other modes [41]. The technique makes it possible to reduce each mode of the data cube.

[2] We used the `R` package `fastICA` to perform the analysis for simultaneously extracting L components [53], [62].

[3] Because preprocessing for ICA requires only centering of the column variables, it is possible to have the components mimic the variance of each slice of the data cube.

[4] Even when we extracted three components, which is the recommended next-best option based on the hull, the weights of the resulting un-mixing matrix closely resembled those in Figure D.2.

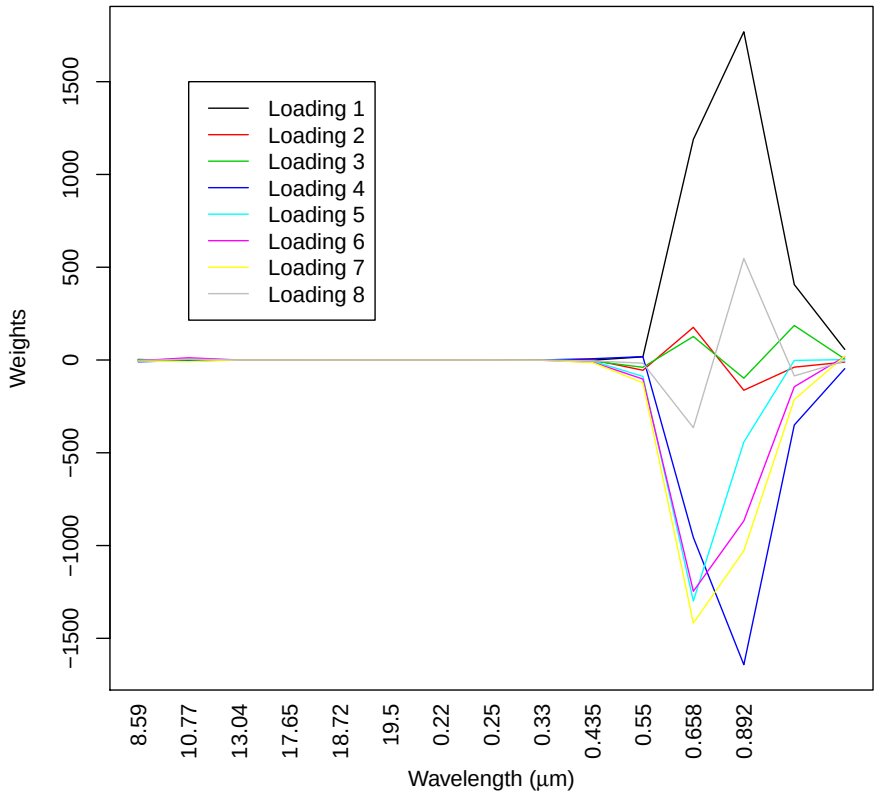


Figure D.2: Plot of the columns of the loading matrix that shows how the original variables would be combined to form eight Independent Components; they cannot discern the hidden structure in the data set.

Similar to the other decompositions, we begin by choosing the number of components to extract by employing the diagnostic introduced in Chapter 4. The plot with all possible model constraints and the associated data approximations are shown in Figure D.3.[5,6] The top five models are shown in Table D.3.

[5]Multiple models may have the same number of components but varying degrees of fit, because to get a model with a total of 10 components, we can extract (3,4,3) or (5,3,2) or (2,3,5), etc. total number of components.

[6]We used the `R` package `ThreeWay` to perform the analysis for extracting P, Q and R components for each of the modes, respectively [24], [62].

Rank	P	Q	R	Fit (%)	# Constraints	Scree-test Value
1	2	10	13	97.22	25	5.60
2	5	10	13	97.30	28	5.51
3	2	7	13	96.28	22	3.69
4	7	10	13	97.30	30	3.26
5	2	5	10	87	17	2.12
			⋮			

Table D.3: Top five solutions on the convex hull for Tucker-3.

While the diagnostic recommends we use the decomposition with 2, 10 and 13 components for each of the three modes respectively, we analyze the loading matrices from the second best model with 5, 10 and 13 components. This is because the columns of the loading matrix, in this case, are more distinct. Similar to the findings of PCA and ICA, the plots of the loading matrices for each mode of the data set, shown in Figures D.4–D.6 suggest that they cannot uncover the structure in the data set, as they are not easily interpretable. As a result, we turn to CP and evaluate the interpretability of its components.

D.2.4 CANDECOMP/PARAFAC

Another technique generalizing SVD to three mode data sets is CP, whose decomposition is unique up to scale [41], [43]. It reduces each mode of the data cube to form a smaller cube with equal sides. To determine how many components to extract per mode, we use the diagnostic tool presented in Chapter 4 and shown for the CP decomposition in Table D.4. It recommends we try to interpret the decomposition with 2 components per mode. However, the weights forming

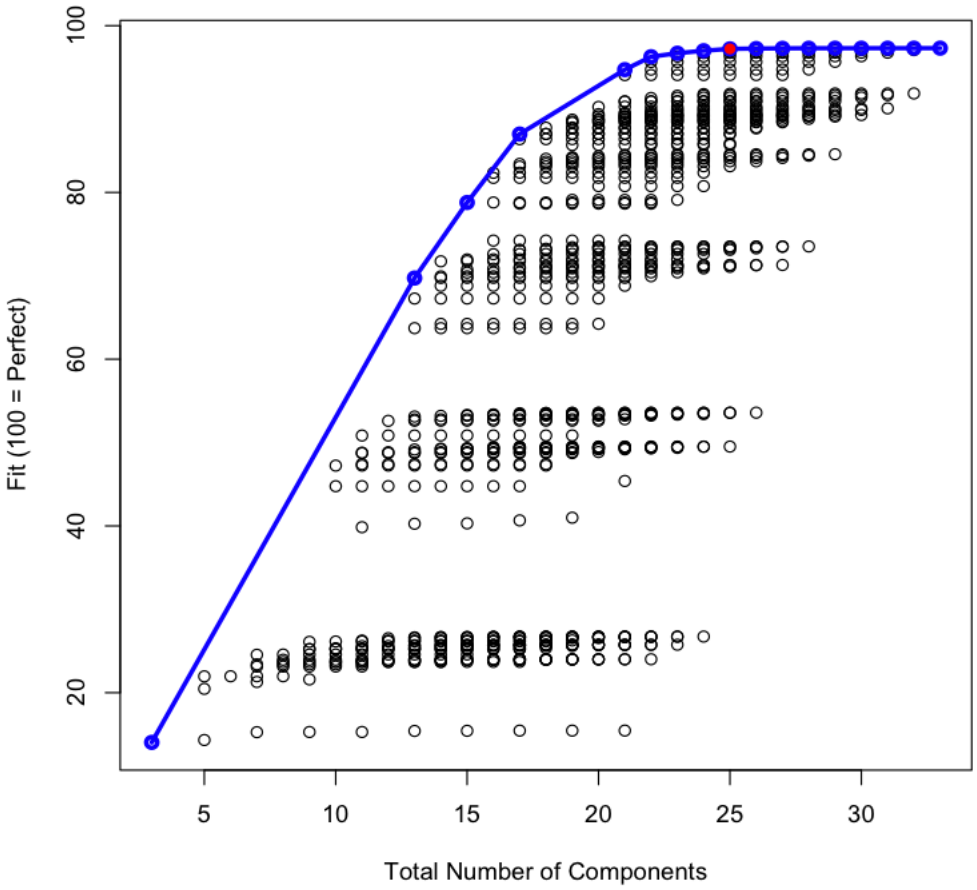


Figure D.3: Tradeoff between model complexity and fit for the Tucker-3 model. Convex hull is highlighted in blue and the optimal solution based on the scree test in red.

the components were near multiples of one another and lacked clear patterns.[7] We considered extracting four components per mode, which resulted in two distinct non-interpretable components for the first mode of the data set shown in Figures D.7–D.9.

[7]We used the `R` package `ThreeWay` to perform the analysis for extracting R components for each of the modes [24], [62].

Rank	Complexity	Fit (%)	Scree-test Value
1	2	99.6354	14.21
2	4	99.8171	2.81
3	5	99.851	1.37
4	6	99.8797	1.72
5	7	99.8993	1.62
		⋮	

Table D.4: Top five solutions on the convex hull for CP.

D.3 Discussion

This chapter showed that the previously developed dimension reduction techniques are not adequate at quantifying the structure of the White Oval. As a result, a new technique has been developed, called AICA, to generalize these approaches and uncover the characteristics of the storm. Its findings are presented in Chapter 5.

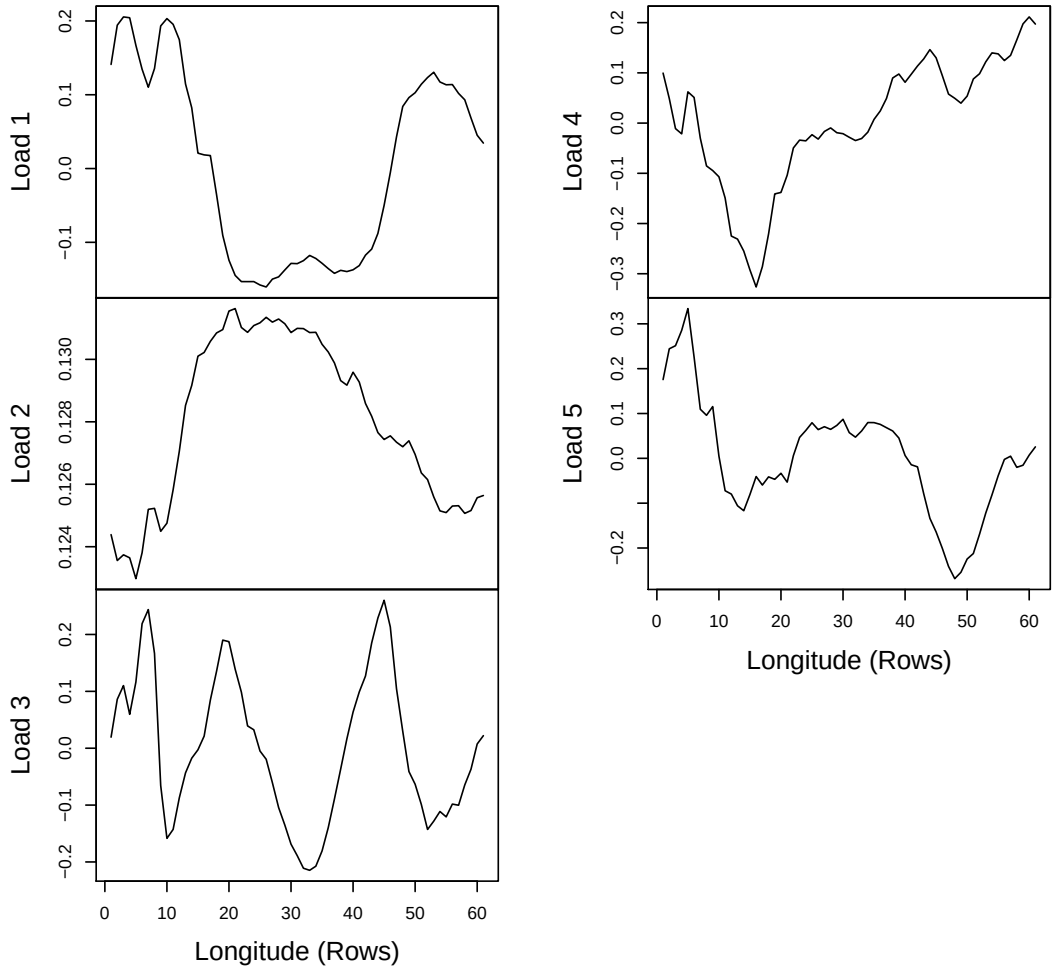


Figure D.4: Loading matrix from the Tucker-3 decomposition for the first mode, extracting five components.

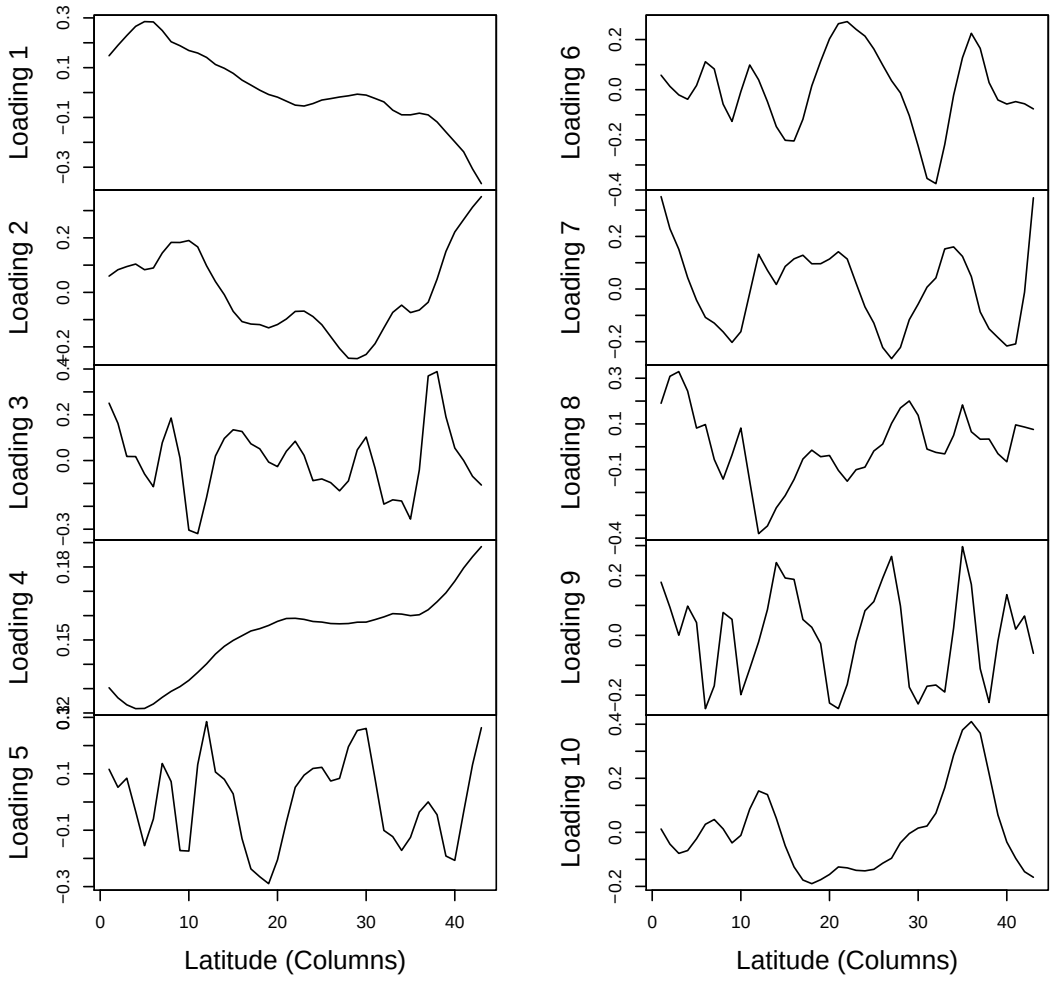


Figure D.5: Loading matrix from the Tucker-3 decomposition for the second mode, extracting 10 components.

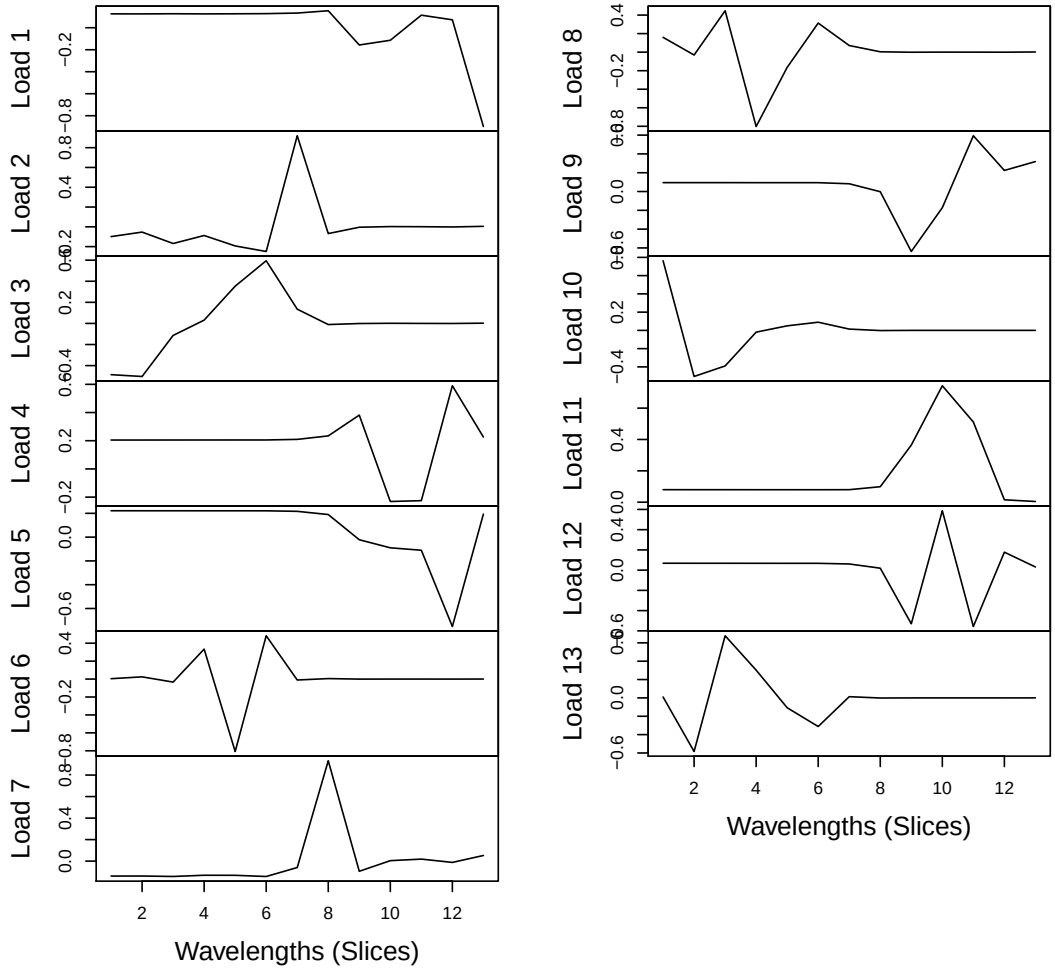


Figure D.6: Loading matrix from the Tucker-3 decomposition for the third mode, extracting 13 components.

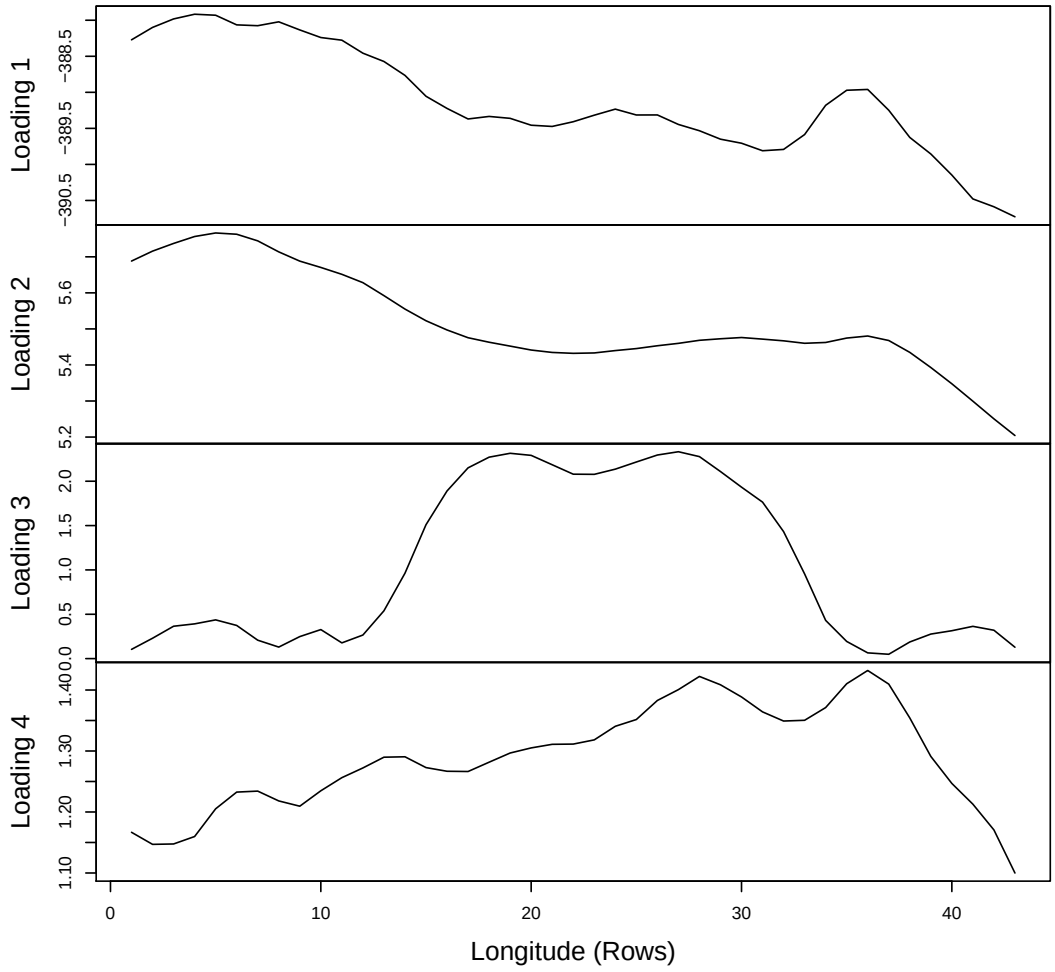


Figure D.7: Loading matrix from the CP decomposition for the first mode, extracting four components.

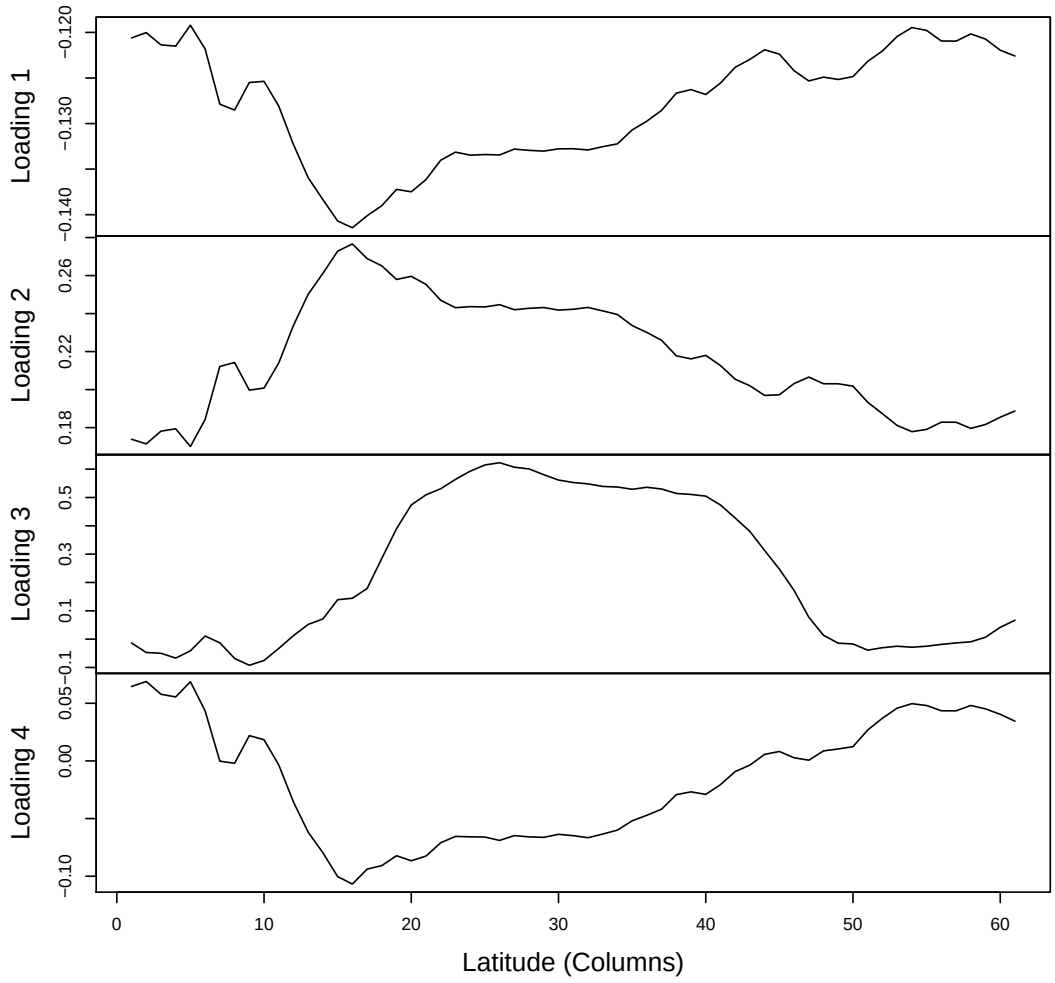


Figure D.8: Loading matrix from the CP decomposition for the second mode, extracting four components.

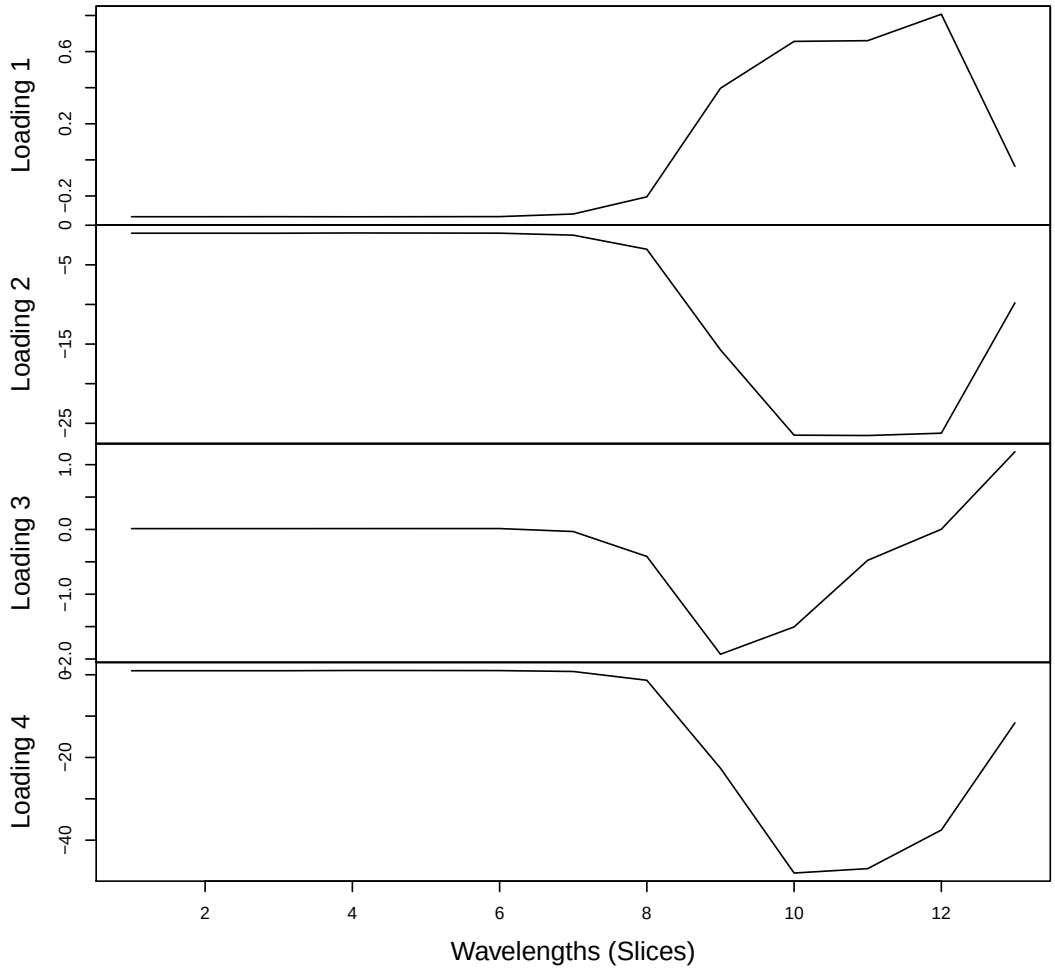


Figure D.9: Loading matrix from the CP decomposition for the third mode, extracting four components.

Bibliography

[1] Hubble 2006: Science year in review. Baltimore: Space Telescope Science Institute (STScI) for NASA Goddard Space Flight Center, 2006. URL http://hubblesite.org/hubble_discoveries/science_year_in_review/2006.

[2] Hubble snaps baby pictures of Jupiter's "Red Spot Jr.". HubbleSite by Space Telescope Science Institute (STScI), May 2006. URL http://hubblesite.org/newscenter/archive/releases/2006/19/.

[3] Voyager - Jupiter, 2010. URL http://voyager.jpl.nasa.gov/science/jupiter.html.

[4] E. Acar and B. Yener. Unsupervised multiway data analysis: A literature survey. *Knowledge and Data Engineering, IEEE Transactions on*, 21(1): 6 –20, Jan 2009. URL http://doi.ieeecomputersociety.org/10.1109/TKDE.2008.112.

[5] S. Amari, A. Cichocki, and H. H. Yang. A new learning algorithm for blind source separation. *Advances in Neural Information Processing 8*, pages 757–763, 1996. URL hil.t.u-tokyo.ac.jp/~kameoka/SAP/papers/Amari1996__A_new_learning_algorithm_for_blind_signal_separation.pdf.

[6] M. W. Binford. Tutorial 1: Remote Sensing. Remote Sensing: Platforms, physics, data, data availability and acquisition. IAI Summer Institute Lectures, Department of Geography University of Florida, 2000. URL yyy.rsmas.miami.edu/IAI/Inst2000/lectures/binford_jul18/rstut.pdf[sic].

[7] R. Bro and H. Kiers. A new efficient method for determining the number of components in parafac models. *Journal of Chemometrics*, 17(5):274–286, 2003. URL http://dx.doi.org/10.1002/cem.801.

[8] R. Bro and A. K. Smilde. Centering and scaling in component analysis. *Journal of Chemometrics*, 17(1):16–33, 2003. ISSN 1099-128X. doi: 10.1002/cem.773. URL http://dx.doi.org/10.1002/cem.773.

[9] J. Carroll and J. Chang. Analysis of individual differences in multidimensional scaling via an n-way generalization of 'Eckart-Young' decomposition. *Psychometrika*, 35:283–319, 1970. URL http://dx.doi.org/10.1007/BF02310791.

[10] E. Ceulemans and H. Kiers. Selecting among three-mode principal component models of different types and complexities: A numerical convex hull based method. *British Journal of Mathematical and Statistical Psychology*, 59(1): 133–150, 2006. URL http://dx.doi.org/10.1348/000711005X64817.

[11] E. Chaisson and S. McMillan. *Astronomy Today 2E.* Media Edition, 1997. URL http://staff.on.br/jlkm/astron2e/start.htm.

[12] A. F. Cheng, A. A. Simon-Miller, H. A. Weaver, K. H. Baines, G. S. Orton, P. A. Yanamandra-Fisher, O. Mousis, E. Pantin, L. Vanzi, L. N. Fletcher, J. R. Spencer, S. A. Stern, J. T. Clarke, M. J. Mutchler, and K. S. Noll. Changing Characterisitcs of Jupiter's Little Red Spot. *The Astronomical Journal*, 135(6), 2008. URL http://dx.doi.org/10.1088/0004-6256/135/6/2446.

[13] J. Coffey. How far is Jupiter from Earth [sic]. *Universe Today*, 2008. URL http://www.universetoday.com/14514/how-far-is-jupiter-from-earth/.

[14] P. Comon. Independent component analysis, a new concept? *Signal Processing*, pages 287–314, 1994. URL www.sciencedirect.com/science/article/pii/0165168494900299.

[15] P. Comon. Blind techniques. http://www.i3s.unice.fr/~pcomon/FichiersPdf/polyD16-2006.pdf, Jan 2010.

[16] P. Comon and C. Jutten, editors. *Handbook of Blind Source Separation*. Elsevier Ltd., 2010.

[17] G. Constantine and T. Savits. A multivariate Faa di Bruno formula with applications. *Transactions-American Mathematical Society*, 348:503–520, 1996. URL www.ams.org/journals/tran/1996-348-02/S0002-9947-96-01501-2/S0002-9947-96-01501-2.pdf.

[18] J. de Leeuw. Number of parameters. Personal Correspondence, July 2012.

[19] J. de Leeuw. Class Lecture. Introduction to Matrix Calculus and Optmization. Technical report, University of California, Los Angeles. Los Angeles, CA, 29 Jan., 2009.

[20] J. de Leeuw and I. Kukuyeva. Component analysis using multivariate cumulants. June 2012.

[21] J. Duchi. Properties of the trace and matrix derivatives. URL http://www.cs.berkeley.edu/~jduchi/projects/matrix_prop.pdf.

[22] European Planetary Science Congress 2008. Diffusion Caused Jupiter's Red Spot Junior To Color Up. *Europlanet*, September 2008. URL http://www.europlanet-eu.org/outreach/index.php?option=com_content&task=view&id=122&Itemid=41.

[23] European Southern Observatory. ESO - VLT Instruments, June 2011. URL http://www.eso.org/public/teles-instr/vlt/vlt-instr.html.

[24] M. A. D. Ferraro, H. A. Kiers, and P. Giordani. *ThreeWay: Three-way component analysis*, 2011. URL http://CRAN.R-project.org/package=ThreeWay. R package version 1.0.

[25] F. M. Flasar, V. G. Kunde, R. K. Achterberg, B. J. Conrath, A. A. Simon-Miller, C. A. Nixon, P. J. Gierasch, P. N. Romani, B. Bézard, P. Irwin, G. L. Bjoraker, J. C. Brasunas, D. E. Jennings, J. C. Pearl, M. D. Smith, G. S. Orton, L. J. Spilker, R. Carlson, S. B. Calcutt, P. L. Read, F. W. Taylor, P. Parrish, A. Barucci, R. Courtin, A. Coustenis, D. Gautier, E. Lellouch, A. Marten, R. Prangé, Y. Biraud, T. Fouchet, C. Ferrari, T. C. Owen, M. M. Abbas, R. E. Samuelson, F. Raulin, P. Ade, C. J. Césarsky, K. U. Grossman, and A. Coradini. An intense stratospheric jet on Jupiter. *Nature*, 427:132–135, Jan. 2004. URL `http://dx.doi.org/10.1038/nature02142`.

[26] L. Fletcher, G. Orton, O. Mousis, P. Yanamandra-Fisher, P. Parrish, P. Irwin, B. Fisher, L. Vanzi, T. Fujiyoshi, T. Fuse, et al. Thermal structure and composition of Jupiter's Great Red Spot from high-resolution thermal imaging. *Icarus*, 208(1):306–328, 2010. URL `http://dx.doi.org/10.1016/j.icarus.2010.01.005`.

[27] P. Giordani, H. Kiers, and M. Del Ferraro. Three-way component analysis using the R package ThreeWay. URL `http://www.dss.uniroma1.it/sites/default/files/pubblicazioni/RT_1_2012_giordani.pdf`.

[28] J. Haby. Barotropic and baroclinic defined. URL `www.theweatherprediction.com/habyhints/49/`.

[29] H. Hammel. A detailed plan for Neptune. URL `http://passporttoknowledge.com/solarsystem/researchers/journals/hst/hammel_jnl8.html`.

[30] R. Harshman. Foundations of the PARAFAC procedure: models and conditions for an 'explanatory' multi-mode factor analysis. *UCLA Working Papers in Phonetics*, 16:1–84, 1970. URL `www.psychology.uwo.ca/faculty/harshman/wpppfac0.pdf`.

[31] S.-L. J. Hu. Probabilistic independence of joint cumulants. *Journal of Engineering Mechanics*, 117(3):640–653, March 1990. URL link.aip.org/link/jenmdt/v117/i3/p640/s1.

[32] A. Hyvärinen. Survey on independent component analysis. *Neural Computing Surveys*, 2:94–218, 1999. URL http://www.cs.helsinki.fi/u/ahyvarin/papers/review.shtml.

[33] A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis.* Wiley Series, 2001.

[34] D. Isbell. Galileo probe suggests planetary science reappraisal. Technical Report 96-10, NASA, 1996. URL http://www2.jpl.nasa.gov/sl9/gll38.html.

[35] E. Jondeau, E. Jurczenko, and M. Rockinger. Moment component analysis: an illustration with international stock markets. Technical report, University of Geneva, 2010. URL www3.unil.ch/wpmu/ibf2011/files/2011/01/Rockinger_MCA.pdf.

[36] A. Kapteyn, H. Neudecker, and T. Wansbeek. An approach ton-mode components analysis. *Psychometrika*, 51(2):269–275, 1986. URL http://dx.doi.org/10.1007/BF02293984.

[37] H. Kiers. Hierarchical relations among three-way methods. *Psychometrika*, 56(3):449–470, 1991. URL http://dx.doi.org/10.1007/BF02294485.

[38] H. Kiers. Towards a standardized notation and terminology in multiway analysis. *Journal of Chemometrics*, 14(3):105–122, 2000. URL eu.wiley.com/legacy/wileychi/chemometrics/582_ftp.pdf.

[39] H. Kiers and A. Kinderen. A fast method for choosing the numbers of components in Tucker3 analysis. *British Journal of Mathematical and Statis-*

tical Psychology, 56(1):119–125, 2003. URL http://dx.doi.org/10.1348/000711003321645386.

[40] T. Kolda. *Multilinear operators for higher-order decompositions*. United States. Department of Energy, 2006. URL prod.sandia.gov/techlib/access-control.cgi/2006/062081.pdf.

[41] T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455, June 2009. URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.153.2059.

[42] P. Kroonenberg and J. De Leeuw. Principal component analysis of three-mode data by means of alternating least squares algorithms. *Psychometrika*, 45(1):69–97, 1980. URL http://dx.doi.org/10.1007/BF02293599.

[43] P. M. Kroonenberg. *Applied Multiway Data Analysis*. John Wiley and Sons, 2008.

[44] I. Kukuyeva. Multivariate statistical analysis for planetary science. Grant NNX08AW15H proposal to the NASA Graduate Student Researchers Program, March 2008.

[45] I. Kukuyeva. Multivariate statistical analysis for planetary science: Progress report and renewal proposal. Grant NNX08AW15H proposal to the NASA Graduate Student Researchers Program, February 2009.

[46] I. Kukuyeva. Principal component analysis: An intuitive approach. Technical report, University of California, Los Angeles, March 2009. URL https://sites.google.com/site/ikukuyeva/about-me/tutorials.

[47] I. Kukuyeva. Multivariate statistical analysis for planetary science: Progress report and renewal proposal. Grant NNX08AW15H proposal to the NASA Graduate Student Researchers Program, March 2010.

[48] M. Kuroda, Y. Mori, M. Iizuka, and M. Sakakihara. Acceleration of the alternating least squares algorithm for principal components analysis. *Computational Statistics and Data Analysis*, 55(1):143 – 153, 2011. ISSN 0167-9473. doi: 10.1016/j.csda.2010.06.001. URL http://dx.doi.org/10.1016/j.csda.2010.06.001.

[49] K. Lange. *Applied probability*. Springer Verlag, 2010. URL http://dx.doi.org/10.1007/978-1-4419-7165-4.

[50] D. Leibovici and R. Sabatier. A singular value decomposition of a k-way array for a principal component analysis of multiway data, pta-k. *Linear Algebra and its Applications*, 269(1):307–329, 1998. URL http://dx.doi.org/10.1016/S0024-3795(97)81516-9.

[51] J. Li, L. Lian, and J. Xu. Gradient method based on epsilon algorithm for large-scale nonlinearoptimization. *World Journal of Modelling and Simulation*, 4(1):64–68, 2008. URL http://www.worldacademicunion.com/journal/1746-7233WJMS/wjmsVol04No01paper08.pdf.

[52] Library 4 Science. Raleigh scattering. URL http://www.chromatography-online.org/topics/raleigh/scattering.html.

[53] J. L. Marchini, C. Heaton, and B. D. Ripley. *fastICA: FastICA Algorithms to perform ICA and Projection Pursuit*, 2012. URL http://CRAN.R-project.org/package=fastICA. R package version 1.1-16.

[54] P. Marcus. Numerical simulation of jupiter's great red spot. *Nature*, 331 (6158):693–696, 1988. URL http://dx.doi.org/10.1038/331693a0.

[55] P. Marcus, X. Asay-Davis, M. Wong, and I. de Pater. Jupiter's Red Oval BA: Dynamics, Color, and Relationship to Jovian Climate Change. URL http://cfd.me.berkeley.edu/research/atmospheric-dynamics-

of-the-giant-planets-jupiters-great-red-spot-red-oval-zonal-winds-climate/.

[56] A. Mika. *Jupiter's Great Red Spot – National Geographic Education*. Wildest Weather in the Solar System. National Geographic Program. URL http://education.nationalgeographic.com/education/activity/jupiter-s-great-red-spot/?ara=1http://education.nationalgeographic.com/education/activity/jupiter-s-great-red-spot/?ar_a=1.

[57] J. Morton and L.-H. Lim. Principal cumulant components analysis. preprint, University of Chicago, 2010. URL http://galton.uchicago.edu/~lekheng/work/pcca.pdf.

[58] NASA Jet Propulsion Laboratory. AIRS: How AIRS works. URL http://airs.jpl.nasa.gov/instrument/how_AIRS_works/.

[59] NASA Science Directorate at NASA Langley Research Center. What wavelength goes with color?, November 2011. URL http://science-edu.larc.nasa.gov/EDDOCS/Wavelengths_for_Colors.html.

[60] NASA/JPL/WFPC2. Pia02823: Oval storms merging on jupiter. NASA's Photojournal, October 2000. URL http://photojournal.jpl.nasa.gov/catalog/pia02823.

[61] T. Phillips. Jupiter's new red spot. *NASA Science*, March 2006. URL http://science.nasa.gov/science-news/science-at-nasa/2006/02mar_redjr/.

[62] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2012. URL http://www.R-project.org/. ISBN 3-900051-07-0.

[63] B. Rasmus. PARAFAC. tutorial and applications. http://www.sciencedirect.com/science/article/pii/S0169743997000324, 1997.

[64] N. Reshetnikov and G. Orton. Radiative-transfer analysis of oval ba's color change at 2.30 microns. Technical report, NASA, 2010. URL usrp.usra.edu/technicalPapers/jpl/ReshetnikovAug10.pdf.

[65] S. Roman. The formula of Faa di Bruno. *The American Mathematical Monthly*, 87(10):805–809, 1980. URL http://www.jstor.org/stable/2320788.

[66] J. R. Schott. *Matrix Analysis for Statistics*. Wiley Series in Probability and Statistics. John Wiley & Sons, Inc., 2 edition, 2005.

[67] A. Simon-Miller, N. Chanover, G. Orton, M. Sussman, I. Tsavaris, and E. Karkoschka. Jupiter's White Oval turns red. *Icarus*, 185(2): 558–562, 2006. URL www.sciencedirect.com/science/article/pii/S0019103506002764.

[68] F. W. Taylor, S. K. Atreya, T. Encrenaz, D. M. Hunten, P. G. J. Irwin, and T. C. Owen. The Composition of the Atmosphere on Jupiter. In F. Bagenal, T. Dowling, and W. McKinnon, editors, *Jupiter: the Planet, Satellites and Magnetosphere*, volume 1 of *Cambridge Planetary Science*, pages 59–78. Cambridge University Press, 2007.

[69] L. Tucker. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31:279–311, 1966. ISSN 0033-3123. URL http://dx.doi.org/10.1007/BF02289464. 10.1007/BF02289464.

[70] Victory Lighting. Infrared technical information, 2012. URL http://www.victorylighting.co.uk/infrared.html.

[71] J. Walters-Williams and Y. Li. Estimation of Mutual Information: A survey. *Rough Sets and Knowledge Technology*, pages 389–396, 2009. URL www.springerlink.com/index/p747r77451013704.pdf.

[72] E. W. Weisstein. Matrix trace. *From MathWorld–A Wolfram Web Resource.* URL http://mathworld.wolfram.com/MatrixTrace.html.

[73] Windows to the Universe. A Graph (Diagram) of Jupiter's Atmospheric Structure. URL http://www.windows2universe.org/jupiter/atmosphere/J_atm_structure_2.html.

[74] Y.-I. Won. README Document for AIRS Level-1B Version 5 IR Calibrated Radiance Products: AIRIBRAD, AIRIBRAD_NRT, AIRIBQAP, ARIBQAP_NRT. Technical report, Goddard Earth Sciences Data and Information Services Center, 2008. URL http://disc.sci.gsfc.nasa.gov/AIRS/documentation/readmes/README.AIRIBRAD.pdf.