On the Separation of T Tauri Star Spectra using Non-negative Matrix Factorization and Bayesian Positive Source Separation

by

Colleen Kenney

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree Doctor of Philosophy

APPROVED, THESIS COMMITTEE:

Marina Vannucci
Professor of Statistics

Dennis Cox
Professor of Statistics

Christopher Johns-Krull
Associate Professor of Physics and Astronomy

Houston, Texas
April, 2008
ABSTRACT

On the Separation of T Tauri Star Spectra using Non-negative Matrix Factorization and Bayesian Positive Source Separation

by

Colleen Kenney

The objective of this study is to compare and evaluate Bayesian and deterministic methods of positive source separation of young star spectra. In the Bayesian approach, the proposed Bayesian Positive Source Separation (BPSS) method uses Gamma priors to enforce non-negativity in the source signals and mixing coefficients and a Markov Chain Monte Carlo (MCMC) algorithm, modified by suggesting simpler proposal distributions and randomly initializing the MCMC to correctly separate spectra. In the deterministic approach, two Non-negative Matrix Factorization (NNMF) algorithms, the multiplicative update rule algorithm and an alternating least squares algorithm, are used to separate the star spectra into sources. The BPSS and NNMF algorithms are applied to the field of Astrophysics by applying the source separation techniques to T Tauri star spectra, resulting in a successful decomposition of the spectra into their sources. These methods are for the first time being applied and evaluated in optical spectroscopy. The results show that, while both methods perform well, BPSS outperforms NNMF. The NNMF and BPSS algorithms improve upon the current methodology used in Astrophysics in two important ways. First, they permit the
identification of additional components of the spectra in addition to the photosphere and boundary layer which can be modeled with current methods. Second, by applying a statistical algorithm, the modeling of T Tauri stars becomes less subjective. These methods may be further extrapolated to model spectra from other types of stars or astrophysical phenomena.
Acknowledgements

I thank Dr. Marina Vannucci, my advisor, for guiding me through the research for this thesis. I also thank my other committee members, Dr. Dennis Cox and Professor Johns-Krull, who provided the real data and guided me through creating the simulated data. A special thank you goes to Alejandro Villagran, who was a great resource and contributor to my research.

I thank my parents for there unending love and support through my graduate school years and my husband Captain Todd Jaszewski for his love and support over the past few years that we have known eachother. Finally, I thank my husband Todd and friend Sarah Thomas for getting me moving when I didn’t know in which direction to go.
## Contents

Abstract ii

List of Illustrations vii

List of Tables viii

1 Introduction 1

1.1 Motivation ........................................... 1

1.2 Contributions ........................................ 3

1.3 T Tauri Star Data .................................... 4

2 Non-negative Matrix Factorization 5

2.1 Literature Review .................................... 7

2.2 Alternating Least Squares ............................ 9

2.3 Multiplicative Update Rule ........................... 13

3 Bayesian Positive Source Separation 16

3.1 Introduction ........................................... 16

3.2 Model and Prior Distributions ....................... 17

3.3 Posterior Inference ................................... 18

4 Methods Comparison 24

4.1 Simulations ............................................ 24

4.1.1 Performance Assessment .......................... 25

4.1.2 Simulation Results ................................. 26

4.2 Real Data .............................................. 29

4.3 Real Data Comparison ............................... 31
4.4 Conclusions ............................................ 34

Bibliography ............................................. 37
### Illustrations

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>(a) Simulated data set 1; (b) $\hat{X}$ as computed by BPSS</td>
<td>26</td>
</tr>
<tr>
<td>4.2</td>
<td>(a),(b) Estimated sources from ALS algorithm; (c),(d) Estimated sources from MUR algorithm; (e),(f) Estimated sources from BPSS</td>
<td>27</td>
</tr>
<tr>
<td>4.3</td>
<td>(a) Simulated data set 2; (b) $\hat{X}$ as computed by BPSS</td>
<td>29</td>
</tr>
<tr>
<td>4.4</td>
<td>(a),(e),(i) Original O, G, and M4 stars, respectively; (b),(f),(j) Estimated sources from ALS algorithm; (c),(g),(k) Estimated sources from MUR algorithm; (d),(h),(l) Estimated sources from BPSS</td>
<td>30</td>
</tr>
<tr>
<td>4.5</td>
<td>(a) Real data set 1; (b) $\hat{X}$ as computed by BPSS</td>
<td>31</td>
</tr>
<tr>
<td>4.6</td>
<td>Data set 1: (a),(b),(c) Estimated sources from ALS algorithm; (d),(e),(f) Estimated sources from MUR algorithm; (g),(h),(i) Estimated sources from BPSS algorithm</td>
<td>32</td>
</tr>
<tr>
<td>4.7</td>
<td>(a) Real data set 2; (b) $\hat{X}$ as computed by BPSS</td>
<td>34</td>
</tr>
<tr>
<td>4.8</td>
<td>Data set 2: (a),(b),(c) Estimated sources from ALS algorithm; (d),(e),(f) Estimated sources from MUR algorithm; (g),(h),(i) Estimated sources from BPSS algorithm</td>
<td>35</td>
</tr>
</tbody>
</table>
# Tables

4.1 Index results for simulated data set 1 ........................................... 28

4.2 CT and PI results for simulated data set 2 (O, G, and M4 type stars) 28

4.3 Index results for data set 1 ......................................................... 33

4.4 Index results for data set 2 ......................................................... 34
Chapter 1

Introduction

1.1 Motivation

T Tauri Stars (TTS) are young, pre-main sequence stars classified by their placement on the Hertzprung-Russel diagram and are characterized by the emission lines hydrogen(H), calcium (Ca), and magnesium (Mg), among others, and and strong lithium (Li) absorption lines in their spectra. T Tauri stars are classified into three types: classical, weak, or naked TTS. CTTS and WTTS are differentiated by the strength of their Hα line in their spectrum. The difference between the three types depends on whether or not the are actively accreting from a disk. CTTS actively accrete from a disk in all cases, while WTTS may or may not accrete from a disk; in order to determine if they are, additional information is needed. NTTS do not accrete from a disk. The study of these young stars allows us to better understand the formation of stars like our sun. Since TTS are characterized by their spectra, their spectral classification is key to the identification of the star type (Gray and Corbally, 2009 [1]).

Astronomers believe there are main potential contributors of classical TTS that can be identified. Currently, two are identified clearly in optical spectra: the photosphere and the boundary layer. The photosphere is the visible part of a star, and the absorption lines of the photospheric spectrum indicate the effective temperature of the star. Generally, stars are classified into one of seven spectral types: O, B, A, F, G, K, M (Gray, 2005 [2]). TTS are young, low mass G, K, and M type stars. The boundary layer is the interface zone between the star and the accretion disk. It is a stellar magnetic field which heats to high temperatures. Originally thought to be just a small area between the star and the accretion disk, it was determined that the difference between the relatively slow rotation of the star and the relatively fast ro-
tation of the accretion disk could not be possible by a thin boundary layer. Because
the stellar magnetic field produces the same type of emission as seen by the boundary
layer, the emission from the field is called boundary layer emission (Bouvier et. al
(2007, [3]).

In addition to the photosphere and the boundary layer, current models of classical
T Tauri stars indicate that there is a circumstellar accretion disk around the TTS.
The disk is made up of gas and dust, and disk material accretes onto the surface of the
star along magnetic field lines. While T Tauri stars show excesses in the ultraviolet
and optical ranges in addition to the infrared spectrum, it is the infrared excess that
suggests the existence of an accretion disk (Alencar, 2007 [4]). It is presumed that
stellar wind carries out enough angular velocity to balance the angular momentum
carried to the star from the accretion disk.

Currently, a forward fitting method is usually used to model the components of a
star. The spectral type is first estimated based on the ratios of photospheric absorp-
tion lines. A template of the estimated spectral type is then taken and added to a
boundary layer spectrum computed from an idealized model. The parameters of this
final spectrum are then varied, including the effects of reddening, until the spectrum
matches the observations well (Valenti et al. 1993 [5]). One problem with the current
method is that it does not indicate whether or not the final spectra are complete.
In particular, the current method obtains estimates for the photospheric and bound-
ary layer contributions, leaving the residual spectrum that may have contributions
from the disk and possibly the wind. The existing approach to source separation
incorporates some user bias in it; it is desirable to remove this bias to the extent
possible.

I consider this source separation problem as solving the problem

$$X = AS + E$$  \hspace{1cm} (1.1)$$

where \(X\) is observed and the mixing coefficients \(A\) and the source signals \(S\) are to be
determined. The error matrix $E$ takes into account that the data contains a certain amount of error, so a perfect separation is unlikely. For the specific application here, as is the case with many types of spectroscopy, the source signals and the mixing coefficients are constrained to be non-negative for interpretability reasons.

I use deterministic and statistical algorithms to identify the spectral components of T Tauri stars. In particular, I apply TTS data to two non-negative matrix factorization (NNMF) algorithms that are widely used in the literature and a Bayesian positive source separation (BPSS) algorithm. These algorithms have been used in fluorescence spectroscopy, chemometrics, and nuclear magnetic resonance (NMR) spectroscopy, among others. Other algorithms such as Principle Component Analysis (PCA), Singular Value Decomposition (SVD), and Independent Component Analysis (ICA) have also been used to separate different types of spectra into components; however, the methods are not always successful in producing positive source signals and mixing coefficients. The algorithms I use are introduce in chapters 2 and 3 and tested with simulated data and real data in sections 4.1 and 4.3, respectively.

1.2 Contributions

This thesis investigates the use of deterministic and statistical algorithms to perform positive source separation with data collected through optical spectroscopy. While many types of spectroscopy have been studied using deterministic methods, only chemical spectroscopy has been looked at with the Bayesian positive source separation (BPSS) algorithm used in this thesis. Also, while BPSS has been shown to outperform Independent Component Analysis (ICA) algorithms in Moussaoui et al. (2006, [6]), I present here a comparison with non-negative matrix factorization (NNMF) which also, unlike ICA, constrains the output to be non-negative.
1.3 T Tauri Star Data

Two data sets consisting of T Tauri spectral data were looked at to compare the three factorization methods. Additionally, two simulation studies were performed. Data set one is a sample of weak and classical TTS that was observed at the Kitt Peak National Observatory using the RC Spectrometer on the four meter Mayall Telescope. The sample size is 26 spectra.

Data set two was taken from the data included in Valenti et al. (1993, [5]). Data set two spans wavelengths from 3400 Å to 4950 Å and consists of 30 spectra. According to Valenti et al., the spectra included in the sample consisted of weak, moderate, and extreme TTS systems.
Chapter 2
Non-negative Matrix Factorization

Classic methods such as singular value decomposition (SVD), principle component analysis (PCA), and factor analysis have usually been the methods used for low-rank approximations of a data matrix that detect structures and reduce the number of variables. While these methods do successfully factor the original data matrix $X$ into coefficient matrix $A$ and source signal matrix $S$, the solutions may be negative, even after they have been rotated. However, the nature of the data oftentimes require that the factorization of the data matrix be non-negative for interpretability, which is the case for the spectra of TTS. The measured data in this thesis are the stars' flux at a given wavelength, where flux is the inherent brightness of a star. In these cases, another method is necessary to reduce the rank of the data so that non-negativity requirements are met.

Paatero et al. (1991, [7]) introduced the notion of adding non-negativity constraints to factor analysis problems, where they analyzed diffusion battery data using a three-matrix positive matrix factorization. Paatero and Tapper then introduce a better known algorithm that involved alternating least squares (1994, [8]). Renamed non-negative matrix factorization (NNMF) by Lee and Seung (1999, [9]), I use the idea of factor analysis with non-negativity constraints to decompose the intrinsically non-negative data into non-negative parts as part of this thesis.

Formally, the definition is as follows:

**Definition 2.1**

*Given a non-negative data matrix $X \in \mathbb{R}^{m \times N}$ and $n < \min\{m, N\}$, find $A \in \mathbb{R}_{+}^{m \times n}$ and $S \in \mathbb{R}_{+}^{n \times N}$ such that*

$$X \approx AS.$$
For the real data we apply factorization methods to, we have $N$ spectra in each data set that span $m$ wavelengths in Angstroms ($\text{Å}$). The algorithms used do not only produce lower ranked matrices than the original data matrix, but they also identify structures in the data that turn out to be our source signals.

To determine the quality of the approximation $\mathbf{AS}$ of $\mathbf{X}$, cost (or objective) functions are used. NNMF algorithms minimize the cost function chosen; the most popular being are the Frobenius norm and the Kullback-Leibler divergence, discussed in sections 2.2 and 2.3, respectively. These two objective functions are sometimes expanded, as well, with the invention of new algorithms that include sparsity and smoothness constraints in addition to the non-negativity constraints. No matter what cost function is used, though, they all have a common goal in mind: to approximate $\mathbf{X}$ with as much accuracy as possible while simultaneously producing results with desired properties.

In NNMF, there is an identifiability issue, so when an algorithm reaches a minimum solution for $\mathbf{A}$ and $\mathbf{S}$, it is a local rather than a global minimum. This occurs since, for any minimum solution for $\mathbf{A}$ and $\mathbf{S}$, there are an infinite number of good solutions given by $\mathbf{AD}$ and $\mathbf{D}^{-1}\mathbf{S}$ for non-negative $\mathbf{D}$ and $\mathbf{D}^{-1}$. While I did not run into any problems with this fact, it may cause a problem for some.

Before a NNMF algorithm can be implemented, the number of components $n$ and initial values for the algorithm must be determined. PCA and SVD are two methods commonly used to estimate $n$. I utilize PCA to determine the number of components. To use PCA for this purpose, the eigenvalues of the non-centered covariance matrix are taken into consideration, where the eigenvalues indicate how much of the variability is explained by a variable. The number of components $n$ chosen should reflect both a large cumulative percentage of variability explained and the number of variables chosen and where the addition of another variable does not significantly increase the variability explained by those additional components. Once
is determined, non-negative matrix factorization algorithms can be used to obtain non-negative, low rank, approximate matrices of the original data.

### 2.1 Literature Review

In general, NNMF algorithms can be classified as multiplicative update rule algorithms, gradient descent algorithms, or alternating least squares (ALS) algorithms (Berry et al., 2006 [10]). I use the ALS algorithm of Paatero and Tapper (1994 [8]), which minimizes the square of the Frobenius norm, and the widely used multiplicative update rule by Lee and Seung (2001 [11]) that utilizes the Kullback-Leibler divergence. The Frobenius norm is the most commonly used objective function and is optimal when the error distribution is Gaussian (Cichocki, 2006a [12]). For poisson distributed errors, the Kullback-Leibler (KL) divergence is more appropriate. Févotte and Cemgil (2009, [13]) show that the factorization of X into A and S is equivalent to maximum likelihood estimation of the mean, variance, or intensity parameters, and that the underlying distribution of the data matrix X is important in choosing a cost function. The two algorithms chosen, the ALS and multiplicative update rule algorithms, were used for their simplicity in both their derivations and their computer code. While the two algorithms minimize different objective functions, the squared Frobenius norm and the Kullback Leibler divergence, respectively, they both produce lower-rank matrices A and S that approximate the data matrix X.

The basic idea behind NNMF is to alternate between minimizing a specified cost function with respect to A and S while holding the variable not being minimized constant (Cichocki et al., 2006a [12]). Cost functions are key in NNMF because they quantify the quality of the approximation AS to X. Considering the statistical distribution of the data and the additive noise is the best way to choose an optimal cost function (Cichocki et al., 2006c [14]). The optical spectrometer, which was used to collect the stars' spectra, has errors related to the poisson distribution because
the data can be considered as photon count data. Because of this, the KL divergence would seem to be the optimal choice for the TTS data, as the KL divergence is related to the Poisson likelihood (Cichocki et al., 2006a). However, for comparison purposes, I do utilize the ALS method, which uses the Frobenius norm as its cost function.

As the research of NNMF developed, other objective functions have been considered. Cichocki et al. (2006a) derive a family of new algorithms using the Csiszár divergence function, which is a generalized divergence function, to improve efficiency and robustness to noise and outliers. Zdunek and Cichocki (2006, [15]) use the Amari alpha divergence, which can be reduced to the KL divergence by changing the limits of parameters, to improve convergence times and the performance of the NNMF method. Speed of convergence of the algorithms for the data in this thesis is not important, however, as the data sets we use are typically small compared to data sets used in other applications. Even with the notoriously slow Lee and Seung multiplicative update rule algorithm, the results for the data sets in this thesis, both real and simulated, take only seconds to compute.

In addition to introducing new objective functions, parameters can be included in the objective function to add smoothness and sparsity constraints. Hoyer (2002 [16], 2004 [17]) introduced sparseness constraints to a projected gradient descent NNMF algorithm to improve the decomposition of data into parts; on iterations when the sparsity constraints are not to be used, Hoyer’s algorithm applies the multiplicative update rule of Lee and Seung (2001) that is derived by minimizing the Frobenius norm. While results from NNMF algorithms without sparseness constraints do exhibit some sparseness, specifically encoding it improves upon decomposition results. I do not include sparseness constraints in the algorithms I utilize, as the data sets used for this thesis are not large enough for sparseness to be an issue. Smoothness constraints have also been considered (Piper, 2004, [18]), and other constraints can be added to improve upon decomposition of the data. Sajda et al. (1993b, [19]) added additional
constraints to Lee and Seung’s Frobenius norm minimizing algorithm to bound the amplitude of the recovered spectra, viewing small or negative values that might be in the data as noise. Therefore, even if negative values do appear, the algorithm does not stop running as other algorithms typically do. They apply their algorithm to Raman spectral data, hyperspectral images, and human brain data.

To set the initial values, random matrices are usually used. However, some have explored alternative initializations. Wild et al. (2004, [20]) examined K-means and K-means clustering to initialize the algorithm for image decomposition. They found that, while these methods reduced the error in the short term, they did not do so in the long term; in fact, these methods performed worse overall than random initializations. I found the same results when trying to perform the initialization methods on the simulated data and the TTS spectral data. Boutsidis and Gallopoulos (2007, [21]) describe an initialization based on applying two SVDs that does outperform the K-means methods. However, I did not pursue this since random initializations worked satisfactorily for the purposes of this thesis.

The ALS and multiplicative update rules will now be described in detail in sections 2.2 and 2.3, respectively.

2.2 Alternating Least Squares

As mentioned previously, principle component analysis (PCA) and singular value decomposition (SVD) are two commonly used factor analysis methods to find reduced rank approximations to matrices and to detect structures in the data matrix. However, because these methods do not constrain the coefficient and source signal matrices to have all their elements be positive, a suitable rotation must be found that is, in essence, a linear transformation, so that all the elements in both the coefficient and source signal matrices are non-negative in applications where negative solutions are not interpretable.
In PCA, the observed data $X$ is usually centered, producing both positive and negative elements. For a non-negative factorization to be achieved, the data matrix should not be centered. Even without centering the data matrix, one must find a suitable rotation matrix $T$ and its inverse $T^{-1}$ so that one of the matrices has a positive coefficient and the other a negative coefficient, thus possibly producing new negative elements.

Matrices which are positive-rotatable (p-rotatable) do exist, however. By definition, "a factorization $X = AS + E$ of rank $n$ is called p-rotatable if it can be transformed to a factorization $X = AT T^{-1}S + E$ so that every element of the new factors $AT$ and $T^{-1}S$ are non-negative with $T$ non-diagonal" (Paatero and Tapper, 1994, [8]). However, there do also exist matrices which are not p-rotatable, so that algorithms performed on matrices after applying PCA or SVD with rotations are not always successful.

To combat the problem of negative elements in a factorization, a different approach using alternating regression can be used. The idea of using alternating regression for factor analysis was introduced by Paatero and Tapper (1993, [22]). The following year, Paatero and Tapper introduced positive matrix factorization using alternating regression by adding the constraints that the coefficient and source signal matrices be positive (1994, [8]). In their original formulation, they start with a matrix of observed data $X$ and assume that the standard deviations $\sigma$ of the elements of $X$ are known. They define their weighted factorization problem as

**Definition 2.1**

*Given $X$ and $\sigma$ and the selected rank $n$, positive matrix factorization is defined as*
follows:

\[ X = AS + E, \quad A : m \times n, \quad S : n \times N, \]  
\[
A_{ik} \geq 0, \quad S_{kj} \geq 0, 
\]
\[
Q = \sum_{i=1}^{m} \sum_{j=1}^{N} \frac{E_{ij}^2}{\sigma_{ij}^2}, 
\]
\[
\{ A, S \} = \arg \min_{A,S} Q. 
\]

Paatero and Tapper (1994) introduced two algorithms based on this concept of minimizing the normalized error term in the bilinear model \( X = AS + E \). The first starts from \( A = A_0 \), and \( S = S_0 \), where in the first iteration these values are randomly chosen. Then, \( A = A_0 + \Delta A \) and \( S = S_0 + \Delta S \) are solved for one at a time while keeping the other constant. Next, solve for the coefficient \( \alpha \) in the equation

\[
(A_0 + \alpha \Delta A)(S_0 + \alpha \Delta S) = X + E, 
\]

where \( \Delta A \) and \( \Delta S \) were found in the prior two steps. This procedure is iterated until convergence, which usually takes 30 to 100 steps. The extension coefficient \( \alpha \) acts as a step size for the algorithm and assists in the algorithm with convergence by increasing towards the end of the iterations. If any negative values are introduced in an iteration, the element is set to zero.

The second algorithm Paatero and Tapper propose converges in fewer steps. Instead of alternating between finding the solution to \( A \) and \( S \), \( \Delta A \) and \( \Delta S \) are determined simultaneously by minimization of the Frobenius norm \( ||E||_F \) in the equation

\[
(A_0 + \Delta A)(S_0 + \Delta S) = X + E, 
\]

In this equation, the term \( \Delta A \Delta S \) is ignored. Again, negative elements produced in an iteration are set to zero.

Minimizing \( ||E||_F \) is equivalent to minimizing the Frobenius norm of \( X - AS \). A common objective function in the literature, which is used by many NNMF algorithms (Berry et al., 2006, [10]), is
\[ f(A, S) = \frac{1}{2} \| X - AS \|^2_F, \] (2.5)

where \( \|A\|_F = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} |a_{ij}|^2} \). Note that equation (2.5) is not convex in \( A \) and \( S \), but the equation is convex in \( A \) or \( S \). In the algorithm I use, \( A \) is first initialized with a random \( m \times n \) matrix or with another method of initialization. Then, until the maximum number of iterations is reached, the algorithm performs a least squares step on \( S \) while holding \( A \) constant and vice versa. Specifically, for \( S \), we solve the equation \( A^TAS = W^TX \) for \( S \), and for \( A \), we solve the equation \( SS^TA = SX^T \) for \( A \). After each least squares step, any negative components of the elements of the matrix being solved for are set to zero. While the algorithm converges to a minimum, it is not necessarily the global minimum; therefore, the solution is not unique ( [10]). The algorithm does find a local minimum.

The idea of solving an optimization problem by alternatively solving for two variables has been around for quite some time. It should be noted that, in general, it has not been proven that any alternating functions converges to a global minimum except in special cases (Berry et al., 1996 [10]).

The ALS algorithm works well in practice. It also adds sparsity, although it is ad hoc since elements that are negative are set to zero. It allows for more flexibility in the iterates, which is not the case for multiplicative update algorithms. Specifically, when an element becomes zero, it is not locked at zero, so it doesn’t necessarily get locked into poor paths. The algorithm also only requires the initialization of \( A \). However, no sparsity is encoded in the mathematical setup, and non-negativity is ad hoc since non-negative elements are simply set to zero. Also, there is no convergence theory behind the ALS algorithm, though linear least squares theory can be used to justify it.
2.3 Multiplicative Update Rule

In 1999, Lee and Seung ([9]) introduced a new algorithm for non-negative matrix factorization based on the Frobenius norm and applied it to the decomposition of the features of a face. In addition to a multiplicative update rule for NNMF using the Euclidean distance, Lee and Sueng (2001, [11]) introduced a multiplicative update rule for non-negative matrix factorization that was based on the Kullback-Leibler divergence (KL divergence),

\[ D(X||AS) = \sum_{ij} \left( X_{ij} \log \frac{X_{ij}}{(AS)_{ij}} - X_{ij} + (AS)_{ij} \right), \tag{2.6} \]

under the constraints that \( A, S \geq 0 \). The KL divergence has a lower bound of zero and vanishes if \( X = AS \). As mentioned before, minimizing the KL divergence can be shown to be related to Maximum Likelihood Estimation of a Poisson random variable.

Lee and Seung introduce the following theorem for the divergence:

**Theorem 2.1**

The divergence \( D(X||AS) \) is nonincreasing under the update rules

\[
S_{a\mu} \leftarrow S_{a\mu} \frac{\sum_i A_{ia} X_{i\mu} / AS_{i\mu}}{\sum_k A_{ka}} \\
A_{ia} \leftarrow A_{ia} \frac{\sum_{\mu} S_{a\mu} X_{i\mu} / AS_{i\mu}}{\sum_{\nu} S_{a\nu}} \tag{2.7}
\]

While Lee and Seung ([11]) proved in their 2001 paper that the divergence is nonincreasing under the update rules in theorem 2.1, it remains to be shown that the algorithm converges to a local minimum or a stationary point. Despite the fact that there is no proof that Lee and Seung’s algorithm reaches a local minimum, it remains the most popular algorithm because of its simplicity (Lin, 2007 [23]).

The multiplicative update rules can be derived by minimizing the Kullback-Leibler divergence, diagonally rescaling the variables with a small and positive number \( \eta \), and adding to it the component being updated. This procedure, which is a diagonally rescaled gradient descent, gives us the update rules found in equation (2.7). To see this, we take the derivative of the equation (2.1) with respect to \( S \) and then of \( A \) :
\[
\frac{\partial D}{\partial S_{ij}} = - \sum_a A_{ai} \frac{X_{aj}}{(AS)_{aj}} + A_{ai}
\]  
(2.8)

\[
\frac{\partial D}{\partial A_{ij}} = - \sum_a S_{ja} \frac{X_{ia}}{(AS)_{ia}} + S_{ja}
\]  
(2.9)

As long as the terms \(\eta_{\alpha \mu} = \frac{S_{a\mu}}{\sum_i A_{ai}}\) and \(\eta_{\alpha a} = \frac{W}{\sum_i A_{au}}\) are small and positive in

\[
S_{a\mu} = S_{a\mu} + \frac{S_{a\mu}}{\sum_i A_{ai}} \left[ \sum_a A_{ai} \frac{X_{a\mu}}{(AS)_{a\mu}} - \sum_a A_{ai} \right]
\]  
(2.10)

and

\[
A_{ia} = A_{ia} + \frac{A_{ia}}{\sum_j S_{ja}} \left[ \sum_a S_{ja} \frac{X_{ia}}{(AS)_{ja}} - S_{ja} \right],
\]  
(2.11)

the update should decrease the Kullback-Liebler divergence. It is easy to see that equations 2.10 and 2.11 do simplify to the multiplicative update rules which uses the Kullback-Leibler divergence cost function in equation (2.7).

Although Lee and Seung admit that the choices of \(\eta\) are not small, they prove through the use of auxiliary functions similar to those found in an Expectation-Maximization (EM) algorithm that the proposed multiplicative update rules are non-increasing as claimed in theorem 2.1. Specifically, Lee and Seung (2001) define an auxiliary function \(G(h, h')\) for \(F(h)\) that satisfy the conditions \(G(h, h') \geq F(h)\) and \(G(h, h) = F(h)\). They use this auxiliary function to prove a lemma that \(F\) is non-increasing under the update rule \(h^{t+1} = \arg\min_h G(h, h^t)\). Proving this lemma is quite simple, and shows that \(F(h^{t+1}) \leq F(h^t)\). They also define another auxiliary function, which is for the divergence cost function to establish that \(F(h) \leq G(h, h')\). This leads them to prove theorem 2.1, which involves taking the derivative of the defined \(G(t, t')\). Solving for \(h^{t+1}\), it is shown that \(f\) is indeed non-increasing under the update rules in theorem 2.1.

Unlike the ALS algorithm, there is some convergency theory; Lee and Seung (2001) show that their algorithm converges to a fixed point and that the update rules are non-increasing. Also, if \(A\) and \(S\) are initialized well, convergence is sped up and a better
fixed point is found. However, the fixed point it settles on may be a local minimum or a saddle point. Also, it is slow for large data sets; many matrix multiplications must be performed in each step, and it takes many iterations for the algorithm to converge. Also, there is no mathematical setup in $A$ and $S$ for sparsity. Finally, once an element becomes zero, it remains zero. This may cause the algorithm to start on a poor path that cannot be changed.
Chapter 3
Bayesian Positive Source Separation

3.1 Introduction

Non-negative source separation can be accomplished in a statistical framework, where it has some advantages. In particular, Bayesian statistics allows for flexible modeling and incorporation of known information about the parameters and other prior knowledge through the use of prior distributions. Through the use of Markov Chain Monte Carlo (MCMC) methods, we can sample from the desired posterior distribution so that we may find the marginal distributions of interest which, in this case, are the source signals and mixing coefficients. In my work, I consider the algorithm of Moussaoui et al. (1996, [6]), which uses a MCMC method to find the mixing coefficients and the source signals for chemical spectrometric data sets.

Prior to Moussaoui et al.'s work, little was done regarding non-negative source separation using a Bayesian framework. In 1999, Ochs et al. ([24]) used a modified Massive Inference Gibbs from Maximum Entropy Data Consultants (Cambridge, England), which enforces positivity on the solution. Miskin and MacKay (2001, [25]) use Ensemble Learning approximation to find an approximate posterior distribution for the model parameters by minimizing the Kullback-Leibler divergence between the approximate distribution and the true posterior. A fully Bayesian approach to Independent Component Analysis (ICA) was presented in Roberts and Choudrey (2003, [26]) which used a hierarchical approach by letting the unknown sources be a generalized nonlinear function of a set of intermediate variables which themselves were functions of the observed data. Additionally, they used variational Bayes learning in their algorithm, but only considered the non-negativity on the mixing matrix A.
3.2 Model and Prior Distributions

The Bayesian approach to the positive source separation problem uses both the likelihood \( P(\mathbf{X}|\mathbf{A}, \mathbf{S}) \) of the data and any prior information about the mixing coefficients and the source signals \( P(\mathbf{A}, \mathbf{S}) \). Assuming that \( \mathbf{A} \) and \( \mathbf{S} \) are independent and excluding the normalization constant, Bayes’ theorem gives us the joint posterior density to sample from,

\[
P(\mathbf{A}, \mathbf{S}|\mathbf{X}) \propto P(\mathbf{X}|\mathbf{A}, \mathbf{S})P(\mathbf{A})P(\mathbf{S}).
\]  

(3.1)

The joint distribution (3.1) can be sampled to get estimates of \( \mathbf{A} \) and \( \mathbf{S} \) using MCMC methods. To do this, we must build a model that incorporates prior knowledge on the noise sequences \( \{e_{it}, t = 1, ..., N\} \), source signals \( \mathbf{S} \), and mixing coefficients \( \mathbf{A} \) to obtain the posterior distribution of interest.

The elementwise version of the equation (1.1) is

\[
x_{it} = \sum_{j=1}^{n} a_{ij} s_{jt} + e_{it},
\]

(3.2)

where we have \( i = 1, ..., m \) measured signals, \( j = 1, .., n \) source signals, and \( t = 1, ..., N \) wavelengths. Unlike the two NNMF algorithms, the error sequences are explicitly modeled. In particular, they are assumed to be zero mean independent and identically distributed (iid) Gaussian random variables, independent of the source signals, with variances \( \{\sigma_i^2\}_{i=1}^{m} \). Letting \( \theta_1 = \{\sigma_i^2\}_{i=1}^{m} \), we have the likelihood function

\[
P(\mathbf{X}|\mathbf{A}, \mathbf{S}, \theta_1) = \prod_{i=1}^{N} \prod_{t=1}^{m} \left( \frac{1}{2\pi \sigma_i^2} \right)^{1/2} \exp \left( -\frac{1}{2\sigma_i^2} (x_{it} - \sum_{j=1}^{n} a_{ij} s_{jt})^2 \right)
\]

(3.3)

For interpretability, the source signals \( \mathbf{S} \) and the the mixing coefficients \( \mathbf{A} \) should be non-negative. To enforce this, Gamma priors are used on both the source signals and mixing coefficients. For the source signals, each \( j \)th signal is Gamma distributed with parameters \( (\alpha_j, \beta_j) \). While the parameters are constant for each signal, they may differ between signals. Similarly, the \( j \)th column of the mixing coefficients is
Gamma distributed with parameters \((\gamma_j, \lambda_j)\) and is associated with the \(j\)th source proportion in the mixture; each column, or profile, is considered constant.

Letting \(\theta_2 = (\alpha_j, \beta_j)\) and \(\theta_3 = (\gamma_j, \lambda_j)\), the vector of prior parameters is \(\theta = \{\theta_1, \theta_2, \theta_3\}\). With the prior parameters defined, the posterior distribution becomes

\[
P(S, A, \theta | X) \propto P(X | A, S, \theta_1)P(S | \theta_2)P(A | \theta_3)
\]  

(3.4)

3.3 Posterior Inference

For the BPSS algorithm presented here, the use of both Gibbs sampling (Geman and Geman, 1984 [27]) and Metropolis-Hastings (Hastings, 1970 [28]) steps are required to sample from the posterior distribution (3.4).

Metropolis-Hastings steps in MCMC provide the means to sample from the posterior distribution when the posterior distribution does not have a closed form up to a normalization constant or is difficult to sample from. Samples from the proposal distribution are accepted or rejected based on an acceptance probability ratio where the goal is to accept sufficient proposals so that the chain travels through the entire parameter space (Gamerman and Lopes, 2006 [29]). Although using the algorithm requires sampling from a proposal density, after many iterations the variable being sampled for converges to the true posterior values.

Gibbs Sampling is a special case of the Metropolis-Hastings algorithm. In Gibbs sampling, the full conditional distributions are sampled one at a time, using previous values for the parameters when they have not been updated and using new parameter values when they have been. For Gibbs sampling, the acceptance probability is always one.

The modified MCMC algorithm by Moussaoui et al. follows. Some modifications were made to their proposal densities, as their high acceptance rate of 80% could be indicative of not sampling from the entire sample space. Such a high acceptance rate indicates this because it means that most of the proposed values are being accepted,
meaning that the entire sample space, which includes both the previous and proposed parameters.

1. Sampling the source signals $S$.

The conditional posterior density for the source signals at the $(r + 1)$th iteration of the Gibbs sampler can be obtained by

$$P(S^{(r+1)}|X, A^{(r)}, \theta^{(r)}) \propto P(X|S, A^{(r)}, \theta^{(r)}) P(S|\theta_2^{(r)})$$

(3.5)

To sample source by source, we fixed $j$ and $t$.

$$P(s_{jt}^{(r+1)}|\cdot) \propto P(\{x_{it}\}_{i=1}^m | s_{j-1,t}^{(r)}, s_{j+1:n,t}^{(r)}, a_i^{(r)} A_{i,m,1:n}^{(r)}, \{\sigma_i^{(r)}\}_{i=1}^m) P(s_{jt}|\alpha_j^{(r)}, \beta_j^{(r)})$$

(3.6)

$$P(s_{jt}^{(r+1)}|\cdot) \propto s_{jt}^{\alpha_j^{(r)}-1} \exp\{-\beta_j^{(r)} s_{jt} - \frac{\tau_s}{2} (s_{jt} - \mu_s)^2\}$$

(3.7)

where

$$\tau_s = \sum_{i=1}^m \left( \frac{a_{ij}^{2(r)}}{\sigma_i^{2(r)}} \right),$$

(3.8)

$$\mu_s = \tau_s^{-1} \sum_{i=1}^m \left( \frac{a_{ij}^{(r)} \delta_{s,jt}^{(r)}}{\sigma_i^{2(r)}} \right),$$

(3.9)

and

$$\delta_{s,jt} = x_{jt} - \sum_{k=1}^{j-1} a_{ik}^{(r)} s_{kt}^{(r+1)} - \sum_{k=j+1}^n a_{ik}^{(r)} s_{kt}^{(r)}$$

(3.10)

A Metropolis-Hastings step is required to sample from (3.7). A left-truncated normal density with mean $S_{jt}$, variance determined using an adaptive MH algorithm, and a left-truncation at zero is taken as the proposal density for the sources $s_{jt}^*$, which will ensure the candidate values are positive. We accept $S_{jt}^*$ with probability

$$\alpha(S_{jt}^{(r)}, S_{jt}^*) = \min \left( 1, \frac{f(S_{jt}^*) Q(S_{jt}^*, S_{jt}^{(r)})}{f(S_{jt}^{(r)}) Q(S_{jt}^{(r)}, S_{jt}^*)} \right)$$

(3.11)

where

$$f(S_{jt}) = S_{jt}^{\alpha_j^{(r)}-1} \exp\left(-\beta_j^{(r)} S_{jt} - \frac{\tau_s}{2} (S_{jt} - S_{jt}^{(r)})^2\right)$$

(3.12)

and

$$Q(S_{jt}^{(r)}, S_{jt}^*) = \frac{1}{1 - \Phi\left(\frac{-s_{jt}^{(r)}}{\sqrt{\tau_s}}\right)} \frac{1}{\sqrt{s\pi \tau_s}} \exp\left(\frac{-1}{2\tau_s} (S_{jt}^* - S_{jt}^{(r)})^2\right)$$

(3.13)
$V_s$ is the variance stated above, and $Q(S_{jt}^{(r)}, S_{jt}^*)$ means that $S_{jt}^*$ will be generated from another point $S_{jt}^{(r)}$. In the computation of $\alpha(S_{jt}^{(r)}, S_{jt}^*)$, the ratio $\frac{Q(S_{jt}^*, S_{jt}^{(r)})}{Q(S_{jt}^{(r)}, S_{jt}^*)}$ reduces to

$$
\frac{Q(S_{jt}^*, S_{jt}^{(r)})}{Q(S_{jt}^{(r)}, S_{jt}^*)} = \frac{1 - \Phi \left( \frac{s_{jt}^{(r)}}{\sqrt{V_s}} \right)}{1 - \Phi \left( \frac{s_{jt}^{(r)}}{\sqrt{V_s}} \right)}
$$

(3.14)

and the ratio $\frac{f(S_{jt}^*)}{f(S_{jt}^{(r)})}$ reduces to

$$
\frac{f(S_{jt}^*)}{f(S_{jt}^{(r)})} = \left( \frac{S_{jt}^*}{S_{jt}^{(r)}} \right)^{a^{(r)-1}} \exp \left( -\beta_j^{(r)} (S_{jt}^* - S_{jt}^{(r)}) - \frac{\tau_s}{2} [(S_{jt}^* - \mu_s)^2 - (S_{jt}^{(r)} - \mu_s)^2] \right),
$$

(3.15)

where $\tau_s$ and $\mu_s$ are defined in (3.8) and (3.9), respectively. Drawing a $u =$ uniform$(0,1)$ random variable and comparing it to $\alpha(S_{jt}^{(r)}, S_{jt}^*)$, we set $S_{jt}^{(r+1)}$ to $S_{jt}^*$ if $u$ is less than $\alpha(S_{jt}^{(r)}, S_{jt}^*)$ and to $S_{jt}^{(r)}$ otherwise.

2. Sampling the mixing coefficients $A$.

At the $(r + 1)\text{th}$ iteration of the Gibbs sampler for the coefficients, we have

$$
P(A^{(r+1)}|X, S^{(r+1)}, \theta^{(r)}) \propto P(X|S^{(r+1)}, A, \theta^{(r)})
$$

(3.16)

To sample $A$ coefficient by coefficient, we fixed $i$ and $j$.

$$
P(a_{ij}^{(r+1)}|\cdot) \propto P(\{x_{it}\}_{t=1}^N | a_{ij}^{(r+1)}, d_{i,j+1:n}^{(r)}, s_{1:n, 1:N}^{(r+1)}, \{ \sigma_i^{(r)} \}_{i=1}^m ) P(a_{ij} | \gamma_j^{(r)}, \lambda_j^{(r)})
$$

(3.17)

$$
P(a_{ij}^{(r+1)}|\cdot) \propto a_{ij}^{\gamma_j^{(r)}-1} \exp\{-\lambda_j^{(r)} a_{ij} - \frac{\tau_a}{2} (a_{ij} - \mu_a)^2\}
$$

(3.18)

where

$$
\tau_a = \frac{\sigma_i^{(r)}}{\sum_{t=1}^N s_{jt}^{2(r+1)}},
$$

(3.19)

$$
\mu_a = \frac{\sum_{t=1}^N s_{jt}^{2(r+1)} \delta^{-j}_{a, it}}{\sum_{t=1}^N s_{jt}^{2(r+1)}},
$$

(3.20)

and

$$
\delta^{-j}_{a, it} = x_{it} - \sum_{k=1}^{j-1} a_{ik}^{(r+1)} s_{kt}^{(r+1)} - \sum_{k=j+1}^n a_{ik}^{(r)} s_{kt}^{(r+1)}
$$

(3.21)
As with the sources signals, a Metropolis-Hasting step is needed in order to sample from (3.18). Again, a left-truncated normal distribution is used to propose new values \( a_{ij}^* \), which will ensure positivity of the candidate values. The derivation of the acceptance probability and the decision rule are similar to that of the source signals \( S \), so the computations are not included.

3. Sampling \( \theta_1 \), i.e. the precision \( \tau_i = 1/\sigma_i^2 \)

The conditional posterior density of the precision \( \tau_i = 1/\sigma_i^2 \), or the noise variances, is

\[
P(\tau_i|\cdot) \propto P(\{x_{it}\}_{t=1}^N | a_{i1:n}^{(r+1)}, s_{i1:n,1:N}^{(r+1)}) P(\tau_i)
\]

(3.22)

\[
P(\tau_i|\cdot) \propto \tau_i^{N/2} \exp\left\{-\frac{\tau_i}{2} \sum_{t=1}^N (x_{it} - \sum_{j=1}^n a_{ij}^{(r+1)} s_{jt}^{(r+1)})^2 \right\} \tau_i^{-\alpha-1} \exp\{-\beta_0 \tau_i\}
\]

(3.23)

Recognizing (3.23) as a Gamma kernel, (3.22) can be sampled from a Gamma distribution with parameters \( (\alpha_0 + N/2, \beta_0 + 1/2 \sum_{t=1}^N (x_{it} - \sum_{j=1}^n a_{ij}^{(r+1)} s_{jt}^{(r+1)})^2) \).

4. Sampling the source hyperparameters \( \theta_2 = (\alpha_j, \beta_j) \)

Having found the source signals, the Gamma prior parameters can be specified. The posterior density of each parameter \( \alpha_j \) is given by

\[
P(\alpha_j|\cdot) \propto P(s_{j1:N}^{(r+1)}|\alpha_j, \beta_j^{(r)}) P(\alpha_j)
\]

(3.24)

To sample each \( \alpha_j \) one at a time, we fixed \( j \) and found that

\[
P(\alpha_j|\cdot) \propto \left[ \prod_{t=1}^N \frac{\beta_j^{\alpha_j}}{\Gamma(\alpha_j)} s_{jt}^{\alpha_j-1} \exp\{-\beta_j s_{jt}\} \right] P(\alpha_j)
\]

(3.25)

Since \( \alpha \) is a non-negative parameter, we assign an exponential hyperprior distribution to it, \( \alpha_j \sim \exp(\lambda_0) \), to find that

\[
P(\alpha_j|\cdot) \propto \left( \frac{\beta_j}{\Gamma(\alpha_j)} \right)^N \left[ \prod_{t=1}^N s_{jt}^{\alpha_j-1} \lambda_0 \exp(-\lambda_0 \alpha_j) \right]
\]

(3.26)

or

\[
P(\alpha_j|\cdot) \propto \exp\left( \alpha_j \left[ N \log(\beta_j^{(r)}) + \sum_{t=1}^N \log(s_{jt}^{(r+1)}) - \lambda_0 \right] - N \log(\Gamma(\alpha_j)) \right).
\]

(3.27)
For the new candidate \( \alpha_j^* \), which follows a Gamma distribution with parameters \( a_0 \) and \( b_0 \), a new value will be accepted with probability

\[
\alpha(\alpha_j^{(r)}, \alpha_j^*) = \min \left\{ 1, \frac{f(\alpha_j^*) Q(\alpha_j^*, \alpha_j^{(r)})}{f(\alpha_j^{(r)}) Q(\alpha_j^{(r)}, \alpha_j^*)} \right\}
\]

(3.28)

where

\[
\frac{f(\alpha_j^*)}{f(\alpha_j^{(r)})} = \exp \left\{ (\alpha_j^* - \alpha_j^{(r)}) \left[ N \log(\beta_j^{(r)}) + \sum_{t=1}^{N} \log(s_{jt}^{(r+1)}) - \lambda_0 \right] - N \log(\Gamma(\alpha_j^*)) \right\}
\]

(3.29)

\[
= \exp \left\{ \left( \alpha_j^* - \alpha_j^{(r)} \right) \left[ N \log(\beta_j^{(r)}) + \sum_{t=1}^{N} \log(s_{jt}^{(r+1)}) - \lambda_0 \right] - N \log \frac{\Gamma(\alpha_j^*)}{\Gamma(\alpha_j^{(r)})} \right\}
\]

(3.30)

and

\[
\frac{Q(\alpha_j^*, \alpha_j^{(r)})}{Q(\alpha_j^{(r)}, \alpha_j^*)} = \left( \frac{\alpha_j^{(r)}}{\alpha_j^*} \right)^{a_0 - 1} \exp\{-b_0(\alpha_j^{(r)} - \alpha_j^*)\}
\]

(3.31)

The decision rule is

\[
\alpha_j^{(r+1)} = \alpha_j^* \text{ if } u \leq \alpha(\cdot), u \sim U(0, 1)
\]

(3.32)

\[
\alpha_j^{(r+1)} = \alpha_j^{(r)} \text{ otherwise.}
\]

(3.33)

For the hyperparameter \( \beta_j \), the conditional posterior density is expressed as

\[
P(\beta_j \mid \cdot) \propto P(s_j^{(r+1)} \mid \alpha_j^{(r+1)}, \beta_j) P(\beta_j)
\]

(3.34)

\[
\propto \beta_j^{\alpha_j^{\cdot} - 1} \prod_{t=1}^{N} s_{jt}^{(r+1)} \exp\{-\beta_j s_{jt}\} P(\beta_j)
\]

(3.35)

Since \( \beta_j \) is a positive scale parameter, we assign a Gamma\((a_0, d_0)\) hyperprior to \( \beta_j \).

The resulting conditional posterior density is then expressed as

\[
P(\beta_j \mid \cdot) \propto \beta_j^{N(\alpha_j^{(r+1)} - 1)} \exp\{-\beta_j \sum_{t=1}^{N} s_{jt}^{(r+1)}\} \frac{\Gamma(a_0)}{\Gamma(a_0 + N(\alpha_j^{(r+1)} - 1))} \beta_j^{a_0 - 1} \exp\{-d_0 \beta_j\}
\]

(3.36)

\[
\propto \beta_j^{a_0 + N(\alpha_j^{(r+1)} - 1) - 1} \exp\{-d_0 + \sum_{t=1}^{N} s_{jt}^{(r+1)} \beta_j\}
\]

(3.37)

Hence, the sampling distribution of \( \beta_j \) is a Gamma distribution with parameters \((c_0 + N(\alpha_j^{(r+1)} - 1), d_0 + \sum_{t=1}^{N} s_{jt}^{(r+1)})\).
5. Sampling the mixing coefficients hyperparameters $\theta_3 = (\gamma_j, \lambda_j)$

The final hyperparameters to be sampled are those from the mixing coefficients, which have $\text{Gamma}(\gamma_j, \lambda_j)$ distributions. We sample from the conditional posterior densities

$$P(\gamma_j | \cdot) \propto P(a_{i,m,j}^{(r+1)} | \gamma_j, \lambda_j^{(r)}) P(\gamma_j)$$

(3.38)

and

$$P(\lambda_j | \cdot) \propto P(a_{1:m,j}^{(r+1)} | \gamma_j^{(r+1)}, \lambda_j) P(\lambda_j)$$

(3.39)

The sampling scheme mirrors that of the source signal hyperparameters, with $a_{ij}$ replacing $s_{jt}$ and changing the hyperparameters from $(\alpha_j, \beta_j)$ to $(\gamma_j, \lambda_j)$. With $j$ held constant, the posterior densities for each iteration are given by

$$P(\gamma | \cdot) \propto \exp\{\gamma_j [m \log(\lambda_j^{(r)}) + \sum_{i=1}^{m} \log(a_{i,j}^{(r+1)}) - f_0] - m \log(\Gamma(\gamma_j))\}$$

(3.40)

and

$$\lambda_j \sim \text{Gamma}(c_1 + m(\gamma_j^{(r+1)} - 1), d_1 + \sum_i = 1^m a_{i,j}^{(r+1)}).$$

(3.41)

For $\gamma_j$, the same acceptance rule is used as in step 4.

50,000 iterations were used in the MCMC with a burn-in of 10,000 iterations. The point estimates were calculated using ergodic means. The ergodic mean is, in general terms

$$\bar{\theta}_t = \frac{1}{t} \theta_t + \frac{1}{t}(t-1)\bar{\theta}_{t-1}$$

(3.42)

This is the best way to compute the point estimates since the ergodic mean does not require that all values be stored until the end of the MCMC.

The described MCMC algorithm is implemented in the following chapter. The algorithm is first tested on simulated data and then with real T Tauri star spectral data.
Chapter 4

Methods Comparison

4.1 Simulations

Two data sets were simulated to evaluate and compare the algorithms described in chapters 2 and 3. One data set took spectra simulated from the same star with ten varying boundary layers (simulated data set 1) to determine if the algorithms correctly decomposed the data set into the photosphere and the boundary layer. The spectra were constructed following the formula

\[ F^*_{\lambda} = F^{\text{ph}}_{\lambda} + F^{\text{BL}}_{\lambda} (\times/\div) x \times 10^{21}, \]

where \( F^*_{\lambda} \) is the signal of the star, \( F^{\text{ph}}_{\lambda} \) and \( F^{\text{BL}}_{\lambda} \) are the signals for the photosphere and the boundary layer, respectively, the value of \( x \) is a uniform random variable between one and five, and \( 10^{21} \) is randomly multiplied or divided by this number. While no error was introduced to this simulated data, it will be added in the paper based on this thesis.

The second data set simulated a galaxy made up of three different types of stars to determine if the algorithms were able to distinguish the star type components of the data set. This data set, simulated data set 2, is made up of O, G, and M4 type stars. The stars for this three data set are from the Coudè Stellar Library from Valdes et al. (2004, [30]) and can be found at http://www.noao.edu/cflib/. Though technically real data, these stars are good templates for the true stars photosperic spectrum.

For the two data sets, PCA was used to determine the number of components to specify in the multiplicative update rule, ALS algorithms, and the BPSS MCMC. The resulting eigenvalues from applying PCA indicated that 100% of the variability was accounted for by the true number of components comprising the data sets in both
data sets.

The multiplicative update rule (MUR) algorithm by Lee and Seung (2001, [11]), the Alternating Least Squares (ALS) algorithm by Paatero and Tapper (1994, [8]), and the Bayesian Positive Source Separation (BPSS) MCMC by Moussaoui et al. (2006, [6]) were applied to the two simulated data sets to see how well they performed in separating the simulated data into their correct source signals. Before presenting the results, I present two performance measures in order to compare the results from the three algorithms.

4.1.1 Performance Assessment

Because it is difficult to accurately assess the performance of the source separation through figures alone, performance measures are necessary. We use two measures, the performance index (PI) (Cichocki and Amari, 2002 ([31])) and residual crosstalk index (CT) (Hosseini et al., 2003 ([32])). Both take on small values, with perfect separation as zero, when a good separation is achieved. The performance index is given by

\[
PI = \frac{1}{n(n-1)} \left[ \sum_{i=1}^{n} \sum_{k=1}^{n} \frac{|g_{ik}|^2}{\max_j |g_{ij}|^2} + \sum_{i=1}^{n} \sum_{k=1}^{n} \frac{|g_{ki}|^2}{\max_j |g_{kj}|^2} - 2n \right]
\]

(4.1)

where \(g_{ij}\) is the \((i,j)\)th element of the matrix \(G = (\mathbf{A}'\mathbf{A})^{-1}\), \(\max_j |g_{ij}|\) is the maximum value among the elements in the \(i\)th row vector of \(G\), and \(\max_j |g_{ji}|\) is the maximum value among the elements of the \(i\)th column vector of \(G\). While the performance does assess the quality of the source separation, it mainly measures the quality of the estimation of the mixing matrix. Note that this is an empirical measure. The residual crosstalk index, on the other hand, gives the means to measure the quality of the reconstruction of each source signal. The CT is defined as

\[
CT_j = \frac{1}{\sum_{t=1}^{N} s_{jt}^2} \sum_{t=1}^{N} (s_{jt} - \hat{s}_{jt})^2 \quad j = 1, ..., n,
\]

(4.2)

where the signals being compared are the true and estimated source signals.
4.1.2 Simulation Results

Figure 4.1: (a) Simulated data set 1; (b) $\hat{X}$ as computed by BPSS

Figure 4.1 shows the ten simulated spectra that make up simulated data set 1. The estimated sources resulting from applying the three factorization techniques can be found in figure 4.2.

Performing PCA on the data set indicated that, as expected, there were two components. While the two NNMF methods and the BPSS method do factor the original data set into two source components that resemble a photosphere and a boundary layer, the boundary layer component for methods still contains traces of the photosphere, which is illustrated by the upward trend of the spectra in figures 4.2(a), (c), and (e). Table 4.1 shows the PI and CT for simulated data set 1. While the BPSS method is a MCMC algorithm, which oftentimes is computationally expensive, it outperforms both NNMF methods significantly.
Next, I simulated a galaxy made up of O, G, and M4 type stars and mixed this data with a 30x3 matrix of uniform(0,1) random variables. The three stars that comprise the simulated data set 2 are shown with the results from using the MUR, ALS, and BPSS methods in figure 4.4. The NNMF methods successfully factor the three components from the mixed galaxy of 30 stars, with the ALS method slightly
Table 4.1: Index results for simulated data set 1

<table>
<thead>
<tr>
<th>Method</th>
<th>NNMF-MUR</th>
<th>NNMF-ALS</th>
<th>BPSS</th>
</tr>
</thead>
<tbody>
<tr>
<td>CT-S</td>
<td>0.0015</td>
<td>0.0010</td>
<td>1.4E-5</td>
</tr>
<tr>
<td>PI</td>
<td>0.7737</td>
<td>1.3993</td>
<td>0.2839</td>
</tr>
</tbody>
</table>

Table 4.2: CT and PI results for simulated data set 2 (O, G, and M4 type stars)

<table>
<thead>
<tr>
<th>Method</th>
<th>NNMF-MUR</th>
<th>NNMF-ALS</th>
<th>BPSS</th>
</tr>
</thead>
<tbody>
<tr>
<td>CT-S₁</td>
<td>0.1042</td>
<td>0.1144</td>
<td>0.0090</td>
</tr>
<tr>
<td>CT-S₂</td>
<td>0.1574</td>
<td>0.0678</td>
<td>0.0827</td>
</tr>
<tr>
<td>CT-S₃</td>
<td>0.0289</td>
<td>0.0028</td>
<td>0.0039</td>
</tr>
<tr>
<td>PI</td>
<td>0.5490</td>
<td>0.9824</td>
<td>0.3572</td>
</tr>
</tbody>
</table>

Visually outperforming the MUR algorithm, which is seen in the estimation of the M4 and G type stars. This is surprising, as data collected from an optical spectroscope would tend to have poisson errors, as the data represents photon counts. As the likelihood for the particular MUR used here, using the Kullback-Liebler divergence, is poisson, one would expect the algorithm by Lee and Seung to perform better than the ALS, which better deals with gaussian errors. The Performance Index indicates that BPSS does better than the NNMF methods in the reconstruction of the sources signals; the Crosstalk results are, however, inconclusive.

Based on both simulations, BPSS slightly outperforms the other two methods. In the next sections, real T Tauri star data sets are introduced, and the three factorization techniques are used to determine the component spectra of T Tauri stars.
Two data sets consisting of T Tauri spectral data were looked at to compare the three factorization methods. Data set one is a sample of weak and classical TTS that were observed at the Kitt Peak National Observatory using the RC Spectrometer on the four meter Mayall Telescope. The resolution for the spectrometer is three Å. This data set spans wavelengths from 3600 Å to 7324.4 Å, and the sample size is 26 spectra. The data for this set were interpolated onto a common x-axis, starting with 3600 Å, since the spectrometer is not very sensitive bluer than 3600 Å, and ending at the minimum of the maximum wavelength for all of the 26 spectra observed so the endpoints of the spectra were common among all spectra. The interpolation allowed us to use all 26 spectra in a single analysis.

Data set two was taken from the data included in Valenti et al. (1993). Data set
two spans wavelengths from 3400 Å to 4950 Å and consists of 30 spectra. According to Valenti et al., the spectra included in the sample of the 96 stars they used consisted of weak, moderate, and extreme TTS systems. Note that we use a subset of the 96 spectra presented in their catalogue. The other data sets had similar wavelength ranges as the one described; because little of a star's spectrum is shown in this range, one data set is sufficient to analyze.

The results from applying the 3 algorithms from chapters 2 and 3 can be found in the next section.
4.3 Real Data Comparison

As described in the previous section, the real data we present consists of 2 data sets. Data set one consists of 26 spectra of weak and classical TTS and can be found in figure 4.5. The eigenvalues of this data set indicated that 98.92% of the variability was explained by the first three components, and the first three eigenvalues were significant. While with the simulated data we looked at the number of components at which PCA indicated that 100% of the variability was explained, finding the number of source signals at which 100% of the variability was explained for real data led to components that were clearly unusable; also, when we executed the algorithms specifying that the number of components as the number of all non-zero eigenvalues, the estimated source signals were zero for executions specifying more than three components.

![Figure 4.5](image)

(a) (b)

Figure 4.5: (a) Real data set 1; (b) $\tilde{X}$ as computed by BPSS

The resulting components from using the ALS, MUR, and BPSS methods can be found in figure 4.6. The first and second component of both NNMF methods appear
to be an M-type star's photosphere and a boundary layer. While the third component does not look like anything in particular, PCA indicates that a third component does exist.

Figure 4.6: Data set 1: (a),(b),(c) Estimated sources from ALS algorithm; (d),(e),(f) Estimated sources from MUR algorithm; (g),(h),(i) Estimated sources from BPSS algorithm

While the performance index uses the estimated mixture matrices to compute the measure, the cross-talk index uses both the known and estimated source signals. Since it is therefore not possible to use the crosstalk index as a performance measure, we propose a new algorithm. To determine the effectiveness of how the BPSS algorithm recovers the measured data, we use
\[ CI - Total = \frac{1}{\sum_{t=1}^{N} x_{it}^2} \sum_{t=1}^{N} (x_{it} - \hat{x}_{it})^2 \quad t = 1, \ldots, N. \] (4.3)

Along with the performance index, table 4.3 shows the results of the new index for data set 1.

<table>
<thead>
<tr>
<th>Method</th>
<th>NNMF-MUR</th>
<th>NNMF-ALS</th>
<th>BPSS</th>
</tr>
</thead>
<tbody>
<tr>
<td>CT-total</td>
<td>12.935</td>
<td>7.5675</td>
<td>7.2789</td>
</tr>
<tr>
<td>PI</td>
<td>0.4976</td>
<td>0.5253</td>
<td>0.5141</td>
</tr>
</tbody>
</table>

Table 4.3 : Index results for data set 1

The results are encouraging but inconclusive. BPSS does only slightly better than the ALS algorithm and much better than the MUR algorithm when compared using the new CT-total index, and the three algorithms do very similarly according to the performance index.

Next are the results for data set 2. Figures of \( \mathbf{X} \) and \( \hat{\mathbf{X}} \) can be found in 4.7. Note that it looks like the majority of spectra are WTTS.

Figures of the computed source signals for data set 2 can be found in figure 4.8, and the performance measure results can be found in table 4.4. Again, the results are ambiguous. Lee and Seung’s MUR algorithm outperforms the other algorithms with respect to the CI-total index, but BPSS outperforms the other algorithms with respect to the PI. For figure 4.8, little detail of star spectra is shown in the decomposition into source signals. This perhaps is primarily due to the fact that blueward of 3600 Å, and the wavelengths of the data set only go to 4950 Å. Therefore, we do not see many of the main features of star.
4.4 Conclusions

The two simulation data sets favor BPSS. However, it would be interesting to see how the algorithms would perform when applied to galaxies containing more star types. While not presented here, galaxies containing four and five star types were tested using the NNMF algorithms. The results showed that these algorithms had difficulty separating stars with similar spectra. Additionally, they had trouble separating
Figure 4.8: Data set 2: (a),(b),(c) Estimated sources from ALS algorithm; (d),(e),(f) Estimated sources from MUR algorithm; (g),(h),(i) Estimated sources from BPSS algorithm spectra when an M-type star, which has a very strong upward trend, was present. More simulations will be necessary to draw stronger conclusions regarding the overall effectiveness of NNMF algorithms in this application.

While the real data results are largely inconclusive, they show promise. The results from data set 2 may be ambiguous because of the short wavelength span it covers, but the results from data set 1 show a tendency to favor BPSS. A data set with a wider wavelength span towards the red side of the spectrum is necessary to see how the algorithms, in particular the BPSS MCMC, do with the spectral decomposition.
The main objectives of this thesis were to show T Tauri stars have three components and compare the statistical BPSS algorithm to deterministic algorithms. While more studies are required, the BPSS MCMC shows the most promise. PCA suggests that the results should be composed of three basic components. While the photosphere and boundary layer figure prominently, the accretion disk does not clearly appear in the results of the real data.

In this thesis, I introduce using Bayesian methods to analyze optical spectra, in particular the spectra of T Tauri stars. Through simulations, I also show that the BPSS method along with two deterministic algorithms can decompose a group of observed spectra into components. Using PCA and analyzing real T Tauri star data, I show that T Tauri stars likely are composed of three components.
Bibliography


