Helsinki University of Technology
Dissertations in Information and Computer Science
Espoo 2009 TKK-ICS-D13

ADVANCES IN INDEPENDENT COMPONENT ANALYSIS AND NONNEGATIVE MATRIX FACTORIZATION

Zhijian Yuan

Dissertation for the degree of Doctor of Science in Technology to be presented with due permission of the Faculty of Information and Natural Sciences for public examination and debate in Auditorium T2 at Helsinki University of Technology (Espoo, Finland) on the 24th of April, 2009, at 12 o'clock.

Helsinki University of Technology
Faculty of Information and Natural Sciences
Department of Information and Computer Science

Teknillinen korkeakoulu
Informaatio- ja luonnontieteiden tiedekunta
Tietojenkäsittelytieteen laitos

Distribution:
Helsinki University of Technology
Faculty of Information and Natural Sciences
Department of Information and Computer Science
P.O.Box 5400
FIN-02015 HUT
FINLAND
URL: http://ics.tkk.fi
Tel. +358-9-451 1
Fax +358-9-451 3277
E-mail: series@ics.tkk.fi

ISBN 978-951-22-9830-3 (Print)
ISBN 978-951-22-9831-0 (Online)
ISSN 1797-5050 (Print)
ISSN 1797-5069 (Online)
URL: http://lib.tkk.fi/Diss/2009/isbn9789512298310

Multiprint Oy/Otamedia
Espoo 2009

Yuan, Z.: **Advances in Independent Component Analysis and Nonnegative Matrix Factorization**. Doctoral thesis, Helsinki University of Technology, Dissertations in Information and Computer Science, TKK-ICS-D13, Espoo, Finland.

Keywords: Independent Component Analysis, FastICA algorithms, Nonnegative Matrix Factorization.

ABSTRACT

A fundamental problem in machine learning research, as well as in many other disciplines, is finding a suitable representation of multivariate data, i.e. random vectors. For reasons of computational and conceptual simplicity, the representation is often sought as a linear transformation of the original data. In other words, each component of the representation is a linear combination of the original variables. Well-known linear transformation methods include principal component analysis (PCA), factor analysis, and projection pursuit. In this thesis, we consider two popular and widely used techniques: independent component analysis (ICA) and nonnegative matrix factorization (NMF).

ICA is a statistical method in which the goal is to find a linear representation of nongaussian data so that the components are statistically independent, or as independent as possible. Such a representation seems to capture the essential structure of the data in many applications, including feature extraction and signal separation. Starting from ICA, several methods of estimating the latent structure in different problem settings are derived and presented in this thesis. FastICA as one of most efficient and popular ICA algorithms has been reviewed and discussed. Its local and global convergence and statistical behavior have been further studied. A nonnegative FastICA algorithm is also given in this thesis.

Nonnegative matrix factorization is a recently developed technique for finding parts-based, linear representations of non-negative data. It is a method for dimensionality reduction that respects the nonnegativity of the input data while constructing a low-dimensional approximation. The non-negativity constraints make the representation purely additive (allowing no subtractions), in contrast to many other linear representations such as principal component analysis and independent component analysis. A literature survey of Nonnegative matrix factorization is given in this thesis, and a novel method called Projective Nonnegative matrix factorization (P-NMF) and its applications are provided.

Preface

The work presented in this thesis has been performed in the Neural Networks Research Centre and the Adaptive Informatics Research Centre of the Laboratory of Computer and Information Science (Faculty of Information and Natural Sciences, Department of Information and Computer Science since 2008) at the Helsinki University of Technology. The research has been funded by the Helsinki Graduate School in Computer Science and Engineering, and Finnish Center of Excellence in Adaptive Informatics Research supported by the Academy of Finland.

I would like to express my deepest gratitude to my supervisor Professor Erkki Oja. Without his advise, support and patience this thesis would have never become a reality. During the years of preparing this dissertation, Erkki has shared his vast scientific knowledge with me, offered countless helpful suggestions and ideas as well as hands-on help with more practical issues. For this he will always have my utmost gratitude and respect.

I would also like to express my gratitude to all my co-authors, Professor Erkki Oja, Professor Scott C. Douglas, Docent Jorma Laaksonen, and Dr. Zhirong Yang who have spent much time working on our joint publications.

Special thanks to the members of Department of Information and Computer Science for creating a nurturing academic atmosphere to work in.

The pre-examiners of my thesis, Professor Andrzej Cichocki and Dr. Patrick Hoyer truly made my thesis much better with their thorough examination and offering their knowledge through a multitude of very invaluable and constructive comments on the work and suggestions for improvements – for your input I am most grateful.

Last but certainly not least I would like to thank the most important thing in my life, my family. My two wonderful children Minjing and Ryan and my beloved wife Jun – without your support this would not have been possible.

Otaniemi, March 2009

Zhijian Yuan

Contents

List of symbols

$\mathbf{s}$	Vector of independent sources
$\mathbf{x}$	Mixture vector
$\mathbf{y}$	Solution vector
$\mathbf{z}$	Whitened mixture vector
$\mathbf{A}$	Mixing matrix
$\mathbf{B}$	Inverse matrix of the mixing matrix
$\mathbf{D}$	Diagonal matrix
$\mathbf{I}$	Identity matrix
$\mathbf{Q}, \mathbf{R}$	Orthogonal matrices
$\mathbf{T}$	Whitening matrix in ICA
$\mathbf{W}$	Unkown parameter matrix in both ICA and NMF
$\mathbf{H}$	Unkown parameter matrix in NMF
$\mathbf{V}$	Nonnegative data matrix in NMF
$I(\mathbf{x}; \mathbf{y})$	Mutual information of two variables $\mathbf{x}$ and $\mathbf{y}$
$H(\mathbf{x})$	The entropy of $\mathbf{x}$
κ_i	ith cumulant
κ_4	Kurtosis
$\mathrm{E}\{\}$	Expectation operator
$N(x)$	Nongaussianity of variable x

List of abbreviations

ALS	Alternating least square
BSS	Blind source separation
CNMF	Constrained NMF
CRB	Cramer-Rao lower bound
EFICA	Efficient FastICA
FOBI	Fourth-order blind identification
ICA	Independent component analysis
ICI	Inter-channel interference
JADE	Joint approximate diagonalization of eigenmatrices
LNMF	Local NMF
ML	Maximum likelihood
NMF	Nonnegative matrix factorization
NMFSC	NMF with sparseness constraint
NNLS	Nonnegative least square
NNSC	Nonnegative sparse coding
nsNMF	NonSmooth NMF
P-NMF	Projective nonnegative matrix factorization
PCA	Principal component analysis
PMF	Positive matrix factorization
PSNMF	Principal sparse NMF
RQI	Rayleigh quotient iteration
SVD	Singular value decomposition
tSVD	Truncated singular value decomposition

Chapter 1

Introduction

1.1 Motivation and overview

A suitable representation of data is central to applications in fields such as machine learning, statistics, and signal processing. The manner in which data are represented determines the course of subsequent processing and analysis. A useful representation has two primary desiderata. First, an amenability to interpretation and second, computational feasibility. Central to obtaining useful representations is the process of dimensionality reduction, wherein one constructs a lower complexity representation of the input data. The reduced dimensionality offers advantages such as denoising, computational efficiency, greater interpretability and easier visualization, among others.

Linear algebra has become a key tool in almost all modern techniques for data analysis. Linear models constitute a special class of general models because of their tractable analytical properties. In this thesis, we discuss the problem of linear source separation. In linear source separation, the model consists of two parts: a set of sources and a linear mapping that links the sources to the observations. This means that one wants to fit a general linear model to the data without knowing almost anything of the sources nor of the linear mapping. This process is usually called blind source separation (BSS). Independent component analysis (ICA) is one of the most powerful techniques to solve the BSS problem. With the assumption of the independence of the sources which is true in many cases of the real world, ICA is able to separate the sources that are linearly mixed in several sensors.

In practice, most data takes the form of matrices. The need to process and conceptualize large sparse matrices effectively and efficiently (typically via low-rank approximations) is essential for many data mining applications, including document and image analysis, recommendation systems, and gene expression analysis. Naturally, many data are nonnegative. While performing dimensionality reduction for inherently nonnegative data such as color intensities, chemical concentrations, frequency counts etc., it makes sense to respect the nonnegativity to avoid physically absurd and uninterpretable results. This viewpoint has both computational as well as philosophical underpinnings. These lead to the problem of nonnegative matrix approximation: Given a set of nonnegative inputs find a small set of nonnegative representative vectors whose nonnegative combinations approximate the input data.

1.2 Contributions of the thesis

The main contributions of this thesis are:

- With the nonnegative constraint to the sources, a nonnegative FastICA algorithm was developed. The convergence of the algorithm was analysed.
- The global convergence of FastICA was studied and the statistical behaviour of FastICA using inter-channel interference was analysed.
- A projective nonnegative matrix factorization was proposed. A family of P-NMF algorithms based on different measures were developed.
- The orthogonality and sparseness of P-NMF were studied. Compared to the NMF method, P-NMF gives more orthogonal columns in the base matrix. Therefore, it is able to learn more localized features.

1.3 Publications of the thesis

This thesis consists of an introduction part and seven publications. Chapter 2 introduces independent component analysis, discusses its properties and the basic results. Chapter 3 discusses the convergence of FastICA algorithms, both local and global. Chapter 4 reviews the nonnegative ICA, and a nonnegative FastICA algorithm is presented. Chapter 5 presents the nonnegative matrix factorization (NMF), summarizes the NMF algorithms and its applications. Chapter 6 briefly introduces the projective NMF algorithms.

Publication I. Zhijian Yuan and Erkki Oja. A FastICA Algorithm for Nonnegative Independent Component Analysis. *In Puntonet, Carlos G.; Prieto, Alberto (Eds.), Proceedings of the Fifth International Symposium on Independent Component Analysis and Blind Signal Separation (ICA 2004)*, Springer Lecture Notes in Computer Science 3195, pp. 1-8, Granada, Spain, 2004.

In this work, a nonnegative FastICA algorithm was developed. It used the idea of the FastICA algorithm with an additional constraint - the sources were assumed to be nonnegative. The algorithm is ended in finite steps. The convergence of the algorithm has also been ensured. The current author implemented the algorithm and performed the experiments.

Publication II. Scott C. Douglas, Zhijian Yuan and Erkki Oja. Average Convergence Behavior of the FastICA Algorithm for Blind Source Separation. *In Rosca, J., Erdogmus, D., Prncipe, J.C. and Haykin, S. (Eds.), Proceedings of the Sixth International Symposium on Independent Component Analysis and Blind Signal Separation (ICA 2006)*, Springer Lecture Notes in Computer Science 3889, pp. 790-798, Charleston, SC, USA, 2006.

In this work, the convergence behaviour of the FastICA algorithm has been investigated using a statistical concept called inter-channel interference (ICI). The analysis of ICI confirms the cubic convergence speed of FastICA algorithm with kurtosis based cost function. The average behavior of ICI obeys a "1/3" rule. The current author analysed the general case and proved that for any amount of sources, the "1/3" rule is true.

Publication III. Erkki Oja and Zhijian Yuan. The FastICA Algorithm revisited: convergence analysis. *IEEE Transactions on Neural Networks.* pp.1370-1381, vol. 17:6, 2006.

In this work, the local convergence of FastICA algorithms with symmetrical orthogonalization is considered. FastICA algorithms have quadratic convergence speed with the general cost function, and cubic with kurtosis cost function. These generalize the behavior of the one-unit algorithms. The global convergence has also been investigated with two sources and two mixtures. The current author analysed the global convergence with general cost function.

Publication IV. Zhijian Yuan and Erkki Oja. Projective nonnegative matrix factorization for image compression and feature extraction. *In: Kalviainen, H.,* *Parkkinen, J. and Kaarna, A. (Eds.), Proceedings of the 14th Scandinavian Conference on Image Analysis*, Springer Lecture Notes in Computer Science 3540, pp. 333-342, Joensuu, Finland, 2005.

In this work, two Projective nonnegative matrix factorization algorithms based on Euclidean distance and Kullback-Leibler divergence have been developed. The experiment shows that these methods give better localization and sparsity compared to nonnegative matrix factorization methods. The current author implemented the algorithms and performed the experiments.

Publication V. Zhirong Yang, Zhijian Yuan and Jorma Laaksonen. Projective Nonnegative Matrix Factorization with Applications to Facial Image Processing. *International Journal on Pattern Recognition & Artificial Intelligence*, Volume 21, Number 8, pp. 1353-1362, 2007.

In this work, the Projective nonnegative matrix factorization algorithms were applied to facial images for clustering. The experimental results reveal better image classification and reconstruction compared to traditional methods. The current author was responsible for the algorithms.

Publication VI. Zhijian Yuan and Erkki Oja. A family of projective nonnegative matrix factorization algorithms. *In: Al-mualla, M. (Ed.), Proceedings of the 9th International Symposium on Signal Processing and its Applications (ISSPA)*, Sharjah, United Arab Emirates, pp. 1-4, 2007.

In this work, several iterative positive projection algorithms were suggested, one based on minimizing Euclidean distance and the others on minimizing the divergence of the original data matrix and its non-negative approximation. Several versions of divergence such as the Kullback-Leibler, Csiszár, and Amari divergence are considered, as well as the Hellinger and Pearson distances. Experimental results show that versions of P-NMF derive bases which are somewhat better suitable for a localized and sparse representation than NMF, as well as being more orthogonal. The current author implemented the algorithms and performed the experiments.

Publication VII. Zhijian Yuan, Zhirong Yang and Erkki Oja. Projective nonnegative matrix factorization: Sparseness, Orthogonality, and Clustering. Submitted to a journal.

In this work, the sparseness, orthogonality, and clustering of the projective nonnegative matrix factorization algorithms were discussed. The algorithm is applied to document clustering and MRI data set. The current author was responsible for the algorithms and experiments.

Chapter 2

Independent component analysis

2.1 Linear ICA model

Independent Component Analysis (ICA) is a statistical and computational technique for revealing hidden factors that underlie sets of random variables, measurements, or signals [9, 27, 37, 75]. It has a number of applications in many fields such as speech enhancement systems, telecommunications, medical signal processing and data mining [5, 27, 45, 82, 83, 98, 99]. The general (nonlinear) ICA model can be expressed as

$$\mathbf{x} = f(\mathbf{s}) \tag{2.1}$$

where $\mathbf{x} = (x_1, x_2, \cdots, x_m)^T$ is the vector of observed random variables, often called the mixtures of unknown signals, $\mathbf{s} = (s_1, s_2, \cdots, s_n)^T$ is the vector of latent variables called independent components (ICs) or source signals, and f is a general unknown function. A special case of (2.1) is the linear ICA model, that is, the function f is linear

$$\mathbf{x} = \mathbf{A}\mathbf{s} \tag{2.2}$$

where the $m \times n$ matrix $\mathbf{A}$ is an unknown constant matrix called the mixing matrix. The task is to identify the mixing matrix $\mathbf{A}$, and separate the source signals $\mathbf{s}$ while only knowing a sample of observed vectors $\mathbf{x}$. To solve this problem, there should be some assumptions/restrictions in linear ICA model [75]:

1. Independence: the source variables are assumed to be statistically independent. Independence is the principle assumption in ICA. Two variables x_1 and x_2 are said to be independent if and only if the joint probability density function (pdf) is the product of two marginal probability density functions
$$p_{x_1 x_2}(x_1, x_2) = p_{x_1}(x_1) p_{x_2}(x_2). \tag{2.3}$$
2. Nongaussianity: the independent components must have at most one Gaussian distribution. Gaussian sources cannot be separated by the independence assumption since the higher-order cumulants are zero for Gaussian distributions. With more than one Gaussian variables, we cannot achieve the original independent components. All we can do is to whiten the data.
3. For simplicity, we assume that the unknown mixing matrix is square and invertable. Thus, the number of the independent sources is equal to the number of mixtures. This assumption can be relaxed. If the number of mixtures is smaller than the number of the independent sources, the linear ICA model (2.2) becomes overcomplete, see [74, 75, 93, 160].

Under the above assumptions, it is easy to see that there are still some indeterminancies:

1. The order and signs of the independent sources cannot be determined.
2. The variances (energies) of the independent sources cannot be determined.

2.2 Data preprocessing for ICA

It is often beneficial to reduce the dimensionality of the data before performing ICA. It might well be that there are only a few latent components in the high-dimensional observed data, and the structure of the data can be presented in a compressed format. Estimating ICA in the original, high-dimensional space may lead to poor results. For example, several of the original dimensions may contain only noise. The dimension reduction should only remove the redundant dimensions and the structure of the data is not flattened as the data are projected to a lower dimensional space. In this section one of the most popular dimensionality reduction methods is discussed: principal component analysis.

In addition to dimensionality reduction, another often used preprocessing step in ICA is to make the observed signals zero mean and decorrelate them. The decorrelation removes the second-order dependencies between the observed signals. It is often accomplished by principal component analysis which will be briefly described next.

2.2.1 Principal component analysis (PCA)

Principal Components Analysis (PCA) is a useful statistical technique that has found application in many fields - from neuroscience to computer graphics - because it is a simple, non-parametric method of extracting relevant information from high-dimensional data.

PCA involves a mathematical procedure that transforms a number of (possibly) correlated variables into a (smaller) number of uncorrelated variables called principal components.

Performing PCA is the equivalent of performing Singular Value Decomposition (SVD) on the covariance matrix of the data. For an observed zero mean vector $\mathbf{x}$ (for nonzero mean variable, centered by removing its mean), PCA starts to work with the covariance matrix $\mathrm{E}\{\mathbf{x}\mathbf{x}^T\}$. The eigenvectors of this covariance matrix form the principal components of the data set. The eigenvector associated with the largest eigenvalue has the same direction as the first principal component. The eigenvector associated with the second largest eigenvalue determines the direction of the second principal component, and so on. Using PCA, it is easy to reduce the high dimensional data to a lower dimension space without loss of too much information: for a given reduced dimension r, choosing the principal components corresponding to the first r highest eigenvalues.

PCA is also called the (discrete) Karhunen-Loéve transform [77] (or KLT, named after Kari Karhunen and Michel Loéve) or the Hotelling transform. PCA has the distinction of being the optimal linear transformation for keeping the subspace that has largest variance. This advantage, however, comes at the price of greater computational requirement if compared, for example, to the discrete cosine transform. Unlike other linear transforms, PCA does not have a fixed set of basis vectors. Its basis vectors depend only on the data set.

2.3 Measuring the independence

2.3.1 Uncorrelation and whitening

Two variables x_1, x_2 are uncorrelated if their covariance is zero:

$$cov(x_1, x_2) = \mathrm{E}\{x_1 x_2\} - \mathrm{E}\{x_1\}\mathrm{E}\{x_2\} = 0. \tag{2.4}$$

It is easy to see that independent variables are uncorrelated by the definition of independence. However, uncorrelatedness does not imply independence. For example, assume that (y_1, y_2) are discrete valued and follow such a distribution that the pair are with probability 1/4 equal to any of the following values: $(0, 1)$, $(0, -1)$, $(1, 0)$, $(-1, 0)$. Then y_1 and y_2 are uncorrelated, as can be simply calculated. On the other hand,

$$\mathrm{E}\{y_1^2 y_2^2\} = 0 \neq 1/4 = \mathrm{E}\{y_1^2\}\mathrm{E}\{y_2^2\}. \tag{2.5}$$

If y_1 and y_2 are independent, then y_1^2 and y_2^2 will also be independent, which is not true because of the above equation. Therefore the variables y_1 and y_2 cannot be independent. Since independence implies uncorrelatedness, many ICA methods constrain the estimation procedure so that it always gives uncorrelated estimates of the independent components. This reduces the number of free parameters, and simplifies the problem. PCA as an orthogonal linear transformation transforms the variables into the uncorrelated components.

A zero mean random vector $\mathbf{z}$ is said to be white if its elements are uncorrelated and have unit variance. That is, for the random vector $\mathbf{z}$, its covariance matrix (as well as correlation matrix) is equal to identity matrix:

$$\mathrm{E}\{\mathbf{z}\mathbf{z}^T\} = \mathbf{I}. \tag{2.6}$$

The way to make a random vector uncorrelated is called whitening or sphering. For the random vector $\mathbf{y}$, the whitening transform $\mathbf{T}$ which makes $\mathbf{z} = \mathbf{T}\mathbf{y}$ white is often given by

$$\mathbf{T} = \mathbf{D}^{-\frac{1}{2}}\mathbf{E}^T \tag{2.7}$$

where $\mathbf{D}$ is a diagonal matrix of the eigenvalues of the covariance matrix $\mathbf{C} = \mathrm{E}\{\mathbf{y}\mathbf{y}^T\}$, and $\mathbf{E}$ is the matrix whose columns are the unit-norm eigenvectors of the covariance matrix $\mathbf{C}$. By a simple calculation, it is easy to find that $\mathrm{E}\{\mathbf{z}\mathbf{z}^T\} = \mathbf{I}$ which means the transform $\mathbf{T}$ is a whitening transform. One thing we should notice is that any orthogonal matrix multiplied by $\mathbf{T}$ is also a whitening transform.

2.3.2 Entropy and cumulants

For a continuous variable x with probability density function $p(x)$, the entropy (often called the differential entropy) $H(x)$ is defined as

$$H(x) = -\int p_x(x) \log p_x(x) dx. \tag{2.8}$$

The entropy can be interpreted as a measure of randomness. The fundamental result for entropy is that a Gaussian variable has the largest entropy among all random variables of unit variance. Therefore, we can use entropy as a measure of concussion.

The cumulants of the variable x are defined by the cumulant generating function $\phi(\omega)$:

$$\phi(\omega) = \ln(\mathrm{E}\{\exp(j\omega x)\}) = \sum_{k=0}^{n} \kappa_k \frac{(j\omega)^k}{k!}, \tag{2.9}$$

where $j = \sqrt{-1}$. The coefficient term κ_k of this expansion is called the kth cumulant. For a zero mean random variable x, the first four cumulants are

$$\kappa_1 = 0, \kappa_2 = \mathrm{E}\{x^2\}, \kappa_3 = \mathrm{E}\{x^3\}, \kappa_4 = \mathrm{E}\{x^4\} - 3[\mathrm{E}\{x^2\}]^2. \tag{2.10}$$

The fourth cumulant κ_4 is called kurtosis. It is a classical measure of nongaussianity. For a Gaussian random variable, its kurtosis is zero. Kurtosis can be both positive or negative. Random variables that have a negative kurtosis are called subgaussian, and those with positive kurtosis are called supergaussian.

Kurtosis, or rather its absolute value, has been widely used as a measure of nongaussianity in ICA and related fields because of its simplicity, both computational and theoretical.

2.3.3 Mutual information

Mutual information was introduced by Shannon [119] in 1948. The mutual information of two random variables is a quantity that measures the mutual dependence of the two variables. Formally, the mutual information of two discrete random variables x and y with supports $\Sigma(x), \Sigma(y)$ can be defined as:

$$I(x; y) = \sum_{x \in \Sigma(x)} \sum_{y \in \Sigma(y)} p(x, y) \log \frac{p(x, y)}{p(x)p(y)} \tag{2.11}$$

where $p(x, y)$ is the joint probability distribution function of x and y, and $p(x)$ and $p(y)$ are the marginal probability distribution functions of x and y, respectively. In the continuous case, we replace summation by a definite double integral:

$$I(x; y) = \int_y \int_x p(x, y) \log \frac{p(x, y)}{p(x)p(y)} dxdy. \tag{2.12}$$

where $p(x, y)$ is now the joint probability density function of x and y, and $p(x)$ and $p(y)$ are the marginal probability density functions of x and y, respectively. Mutual information $I(x; y)$ is the amount of information gained about x when y is learned, and vice versa. $I(x, y) = 0$ if and only if x and y are independent.

Mutual information has close relationship with entropy, it can be equivalently expressed as

$$\begin{aligned} I(x; y) &= H(x) - H(x|y) && (2.13) \\ &= H(y) - H(y|x) && (2.14) \\ &= H(x) + H(y) - H(x, y) && (2.15) \end{aligned}$$

where $H(x|y)$ is called conditional entropy, and $H(x, y)$ is called joint entropy.

For multiple variables $\mathbf{x} = (x_1, \cdots, x_n)$, the mutual information is defined as

$$I_{\mathbf{x}} = \sum_i H(x_i) - H(\mathbf{x}). \tag{2.16}$$

Mutual information can be interpreted as a measure using Kullback-Leibler divergence. For two $n-$dimensional probability density functions p_1 and p_2, the Kullback-Leibler divergence is defined as

$$D(p_1, p_2) = \int p_1(\eta) \log \frac{p_1(\eta)}{p_2(\eta)} d\eta. \tag{2.17}$$

Using Kullback-Leibler divergence implies an important fact that mutual information is always nonnegative, and it is zero if and only if the variables are independent. Therefore, minimizing mutual information gives as independent variables as possible.

In linear ICA model (2.2), we need to search for a linear transformation matrix $\mathbf{W}$ that produces maximally independent variables $\mathbf{y} = \mathbf{W}\mathbf{x}$:

$$I_{\mathbf{y}} = \sum_i H(\mathbf{y}_i) - H(\mathbf{y}) = \sum_i H(\mathbf{y}_i) - H(\mathbf{x}) - \log|\det \mathbf{W}|. \tag{2.18}$$

The above equation uses the form $H(\mathbf{y}) = H(\mathbf{x}) + \log|\det \mathbf{W}|$. It can be derived with $\mathbf{y} = \mathbf{W}\mathbf{x}$ as follow:

$$\begin{aligned}
H(\mathbf{y}) &= -\int p_{\mathbf{y}}(\mathbf{y}) \log p_{\mathbf{y}}(\mathbf{y}) d\mathbf{y} && (2.19)\\
&= -\int p_{\mathbf{x}}(\mathbf{x}) \log \frac{p_{\mathbf{x}}(\mathbf{x})}{det\mathbf{W}} d\mathbf{x} && (2.20)\\
&= -\int p_{\mathbf{x}}(\mathbf{x}) \log p_{\mathbf{x}}(\mathbf{x}) d\mathbf{x} + \int p_{\mathbf{x}}(\mathbf{x}) \log|\det \mathbf{W}| d\mathbf{x} && (2.21)\\
&= H(\mathbf{x}) + \log|\det \mathbf{W}|. && (2.22)
\end{aligned}$$

The problem with mutual information is that it is difficult to estimate. Since using the definition of entropy, one needs an estimate of the density. This problem has severely restricted the use of mutual information in ICA estimation. However, since mutual information is the natural information-theoretic measure of the independence of random variables, we could use it as the criterion for finding the ICA algorithms. Although ICA objective functions could be derived from different starting points and criteria, they are often equivalent to the criteria in the above equation (2.18).

2.4 ICA algorithms

Many scientists have made their contributions on ICA. We can roughly say that most of the ICA algorithms start from one of the four different criteria: maximization of non-Gaussianity of the components [71, 69], minimizing mutual information [171], maximum likelihood (ML) estimation [1, 8, 104] and tensorial methods [20, 21, 22]. In fact, there are quite close connections among non-Gaussianity, mutual information and maximum likelihood, see [19, 75]. FastICA algorithms use maximization of non-Gaussianity, we will introduce these in the next section. Since negentropy is very difficult to compute, in practice, it is approximated by cumulants.

ML based methods include Bell-Sejnowski algorithm, also called the infomax principle [8] and natural gradient algorithm. FOBI (first-order blind identification) and JADE (joint approximate diagonalization of eigenmatrices) are the two well-known tensorial methods.

2.4.1 Algorithms by maximum likelihood

In linear ICA model (2.2), we have the density function $p_{\mathbf{x}}(\mathbf{x})$ as following:

$$p_x(\mathbf{x}) = |\det \mathbf{B}| p_s(\mathbf{s}) = |\det \mathbf{B}| \prod_i p_i(s_i), \tag{2.23}$$

where $\mathbf{B} = \mathbf{A}^{-1}$, and p_i denote the densities of the independent components. Assume that we have T observations of $\mathbf{x}$, denoted by $\mathbf{x}(t), t = 1, \cdots, T$. Then the likelihood of the data $\mathbf{x}$ can be considered as the function of the demixing matrix $\mathbf{B}$:

$$L(\mathbf{B}) = \prod_{t=1}^{T} |\det \mathbf{B}| \left(\prod_{i=1}^{n} p_i(\mathbf{b}_i^T \mathbf{x}(t)) \right) \tag{2.24}$$

where $\mathbf{B} = (\mathbf{b}_1, \cdots, \mathbf{b}_n)^T$. The log-likelihood is

$$\log L(\mathbf{B}) = \sum_{t=1}^{T} \sum_{i=1}^{n} \log p_i(\mathbf{b}_i^T \mathbf{x}(t)) + T \log |\det \mathbf{B}|. \tag{2.25}$$

The Bell-Sejnowski algorithm [8], originally derived from the infomax principle, can be easily derived by ML estimator. Applying the gradient method to equation (2.25), we obtain

$$\Delta \mathbf{B} \propto [\mathbf{B}^T]^{-1} + \mathrm{E}\{g(\mathbf{B}\mathbf{x})\mathbf{x}^T\}, \tag{2.26}$$

where the function $g = (g_1(y_1), \cdots, g_n(y_n))$ is a component-wise vector function that consists of the score function g_i of the distributions of s_i, defined as

$$g_i = (\log p_i)' = \frac{p_i'}{p_i}. \tag{2.27}$$

The Bell-Sejnowski algorithm usually suffers from slow convergence. Furthermore, calculation of one iteration is rather intensive because of the matrix inversion. This can be avoided by presphering the data, and especially by using natural gradient.

The natural gradient method [1] simplifies the gradient method considerably, and makes it better conditioned. The principle of the natural gradient is based on the geometrical structure of the parameter space, and is related to the principle of relative gradient that uses the Lie group structure of the ICA problem. In the case of ICA, multiplying the right-hand side of (2.26) by $\mathbf{B}^T\mathbf{B}$ gives

$$\Delta \mathbf{B} \propto (\mathbf{I} + \mathrm{E}\{g(\mathbf{y})\mathbf{y}^T\})\mathbf{B}. \tag{2.28}$$

where $\mathbf{y} = \mathbf{B}\mathbf{x}$.

2.4.2 Tensorial methods

Tensors are generalizations of linear operators, in particular, cumulant tensors are generalizations of the covariance matrix. Minimizing the higher order cumulants approximately amounts to higher order decorrelation, and can thus be used to solve the linear ICA model. The most well-known among these tensorial methods are fourth-order blind identification (FOBI) and joint approximate diagonalization of eigenmatrices (JADE) [20, 21, 22].

The FOBI algorithm starts from the following matrix of the whitened data $\mathbf{z}$,

$$\Sigma = \mathrm{E}\{\mathbf{z}\mathbf{z}^T ||\mathbf{z}||^2\} \tag{2.29}$$

Using the independence of the sources s_i, we have

$$\Sigma = \mathbf{W}^T \mathrm{diag}(\mathrm{E}\{s_i^4\} + n - 1)\mathbf{W}, \tag{2.30}$$

where $\mathbf{W}$ is the orthogonal separating matrix. If the above diagonal matrix has distinct elements, we can simply compute the decomposition on Σ, and the separating matrix $\mathbf{W}$ is obtained immediately. However, FOBI works only under the restriction that the kurtoses of the ICs are all different. This limits its applications.

Jade is probably the most popular method among tensorial methods. Since eigenvalue decomposition can be viewed as diagonalization, in linear ICA model, the matrix $\mathbf{W}$ diagonalizes $\mathbf{F}(\mathbf{M})$ for any $\mathbf{M}$, where $\mathbf{F}$ represents the cumulant tensor and $\mathbf{M}$ the corresponding eigenmatrices. Thus, we could take a set of different matrices $\mathbf{M}_i, i = 1, \cdots, k$, and try to make the matrices $\mathbf{W}\mathbf{F}(\mathbf{M}_i)\mathbf{W}^T$ as diagonal as possible. One possible objective function of the joint diagonalization process is

$$J_{JADE}(\mathbf{W}) = \sum_i ||\mathrm{diag}(\mathbf{W}\mathbf{F}(\mathbf{M}_i)\mathbf{W}^T)||^2 \tag{2.31}$$

However, tensorial methods are normally restricted, for computational reasons, to small dimensions, and they have statistical properties inferior to those methods using nonpolynomial cumulants or likelihood.

2.5 FastICA

FastICA was first developed by Hyvärinen and Oja [71, 72] for the kurtosis cost function in one unit case. Late, Hyvärinen et al [15, 67, 68, 69, 70] extended this method to general forms. This section is mainly based on [66, 75].

Consider the linear combination of $\mathbf{x}$, $y = \mathbf{b}^T\mathbf{x}$, using equation (2.2), we have

$$y = \mathbf{b}^T\mathbf{x} = \mathbf{b}^T\mathbf{A}\mathbf{s} =: \mathbf{q}^T\mathbf{s} \tag{2.32}$$

where $\mathbf{q} = \mathbf{A}^T\mathbf{b}$. That is, y is a linear combination of independent components s_i. The central limit theorem, a classical result in probability theory, states that as the sample size n increases, the distribution of the sample average of these random variables approaches the normal distribution. Thus, the sum of even two independent variables is usually more Gaussian than the original variables. Therefore, y is more Gaussian than any of the original source s_i and becomes least Gaussian when it is equals to one of the sources s_i. Therefore, when we maximize the nongaussianity of $\mathbf{b}^T\mathbf{x}$ with respect to vector $\mathbf{b}$, it will give us one of the independent components.

FastICA is a family of algorithms derived by maximizing nongaussianity. Negentropy is used to measure the nongaussianity:

$$N(x) = H(x_{Gauss}) - H(x) \tag{2.33}$$

where x is a random variable, and x_{Gauss} is a Gaussian random variable of the same correlation (and covariance) matrix as x. The general approximation of negentropy $N(x)$ using only one nonquadratic function G is [75]

$$N(x) \approx [\mathrm{E}\{G(x)\} - \mathrm{E}\{G(v)\}]^2 \tag{2.34}$$

where v is a Gaussian variable of zero mean and unit variance.

We consider the whitened data $\mathbf{z}$ instead of $\mathbf{x}$. Let $y = \mathbf{w}^T\mathbf{z}$. According to Kuhn-Tucker conditions, which are necessary conditions for an optimal solution in nonlinear optimization problem with some regularity conditions satisfied [85], the optima of $\mathrm{E}\{G(y)\}$ under the constraint $\mathrm{E}\{(\mathbf{w}^T\mathbf{z})^2\} = ||\mathbf{w}||^2 = 1$ are obtained at the points where

$$\mathrm{E}\{\mathbf{z}g(\mathbf{w}^T\mathbf{z}) - \beta\mathbf{w}\} = 0 \tag{2.35}$$

where g is the derivative function of G, and $\beta = \mathrm{E}\{\mathbf{w}_0^T\mathbf{z}g(\mathbf{w}_0^T\mathbf{z})\}$ is a constant with $\mathbf{w}_0$ the value of $\mathbf{w}$ at the optimum. Using Newton's method, we get the following approximative Newton iteration:

$$\mathbf{w}^+ = \mathbf{w}_i - [\mathrm{E}\{\mathbf{z}g(\mathbf{w}_i^T\mathbf{z})\} - \beta\mathbf{w}_i]/[\mathrm{E}\{g'(\mathbf{w}_i^T\mathbf{z})\} - \beta] \tag{2.36}$$

$$\mathbf{w}_{i+1} = \mathbf{w}^+/||\mathbf{w}^+|| \tag{2.37}$$

where $\beta = \mathrm{E}\{\mathbf{w}^T\mathbf{z}g(\mathbf{w}^T\mathbf{z})\}$ which is approximated using current value of $\mathbf{w}$ instead of $\mathbf{w}_0$. Simplifying the above algorithm, the one unit FastICA algorithm is derived as

$$\mathbf{w} \leftarrow \mathrm{E}\{\mathbf{z}g(\mathbf{w}^T\mathbf{z}) - \mathrm{E}\{g'(\mathbf{w}^T\mathbf{z})\}\mathbf{w}\}, \tag{2.38}$$

$$\mathbf{w} \leftarrow \mathbf{w}/||\mathbf{w}||. \tag{2.39}$$

A useful modification of the fixed-point algorithm (2.36) is done by adding a step size to ameliorate the uncertaincy of the convergence in Newton method, obtaining the stabilized fixed-point algorithm

$$\mathbf{w}^{+} = \mathbf{w}_i + \eta[\mathrm{E}\{\mathbf{z}g(\mathbf{w}_i^T\mathbf{z})\} - \beta\mathbf{w}_i] \tag{2.40}$$

$$\mathbf{w}_{i+1} = \mathbf{w}^{+}/||\mathbf{w}^{+}|| \tag{2.41}$$

Estimation of several independent components can be done by two ways: deflationary orthogonalization or symmetric orthogonalization. In deflationary orthogonalization, the independent components are found one by one, and orthogonalized by Gram-Schmidt method. Symmetric orthogonalization is done by first doing one-unit algorithm for every vector $\mathbf{w}_i$ in parallel, and then orthogonalizing by a method using matrix square roots:

$$\mathbf{W} \leftarrow (\mathbf{W}\mathbf{W}^T)^{-1/2}\mathbf{W}. \tag{2.42}$$

A contrast function is any non-linear function which is invariant to permutation and scaling matrices, and attains its minimum value in correspondence of the mutual independence among the output components. A widely used contrast function both in FastICA and also in many other ICA algorithms [23, 38, 118] is the kurtosis. This approach can be considered as an extension of the algorithm by Shalvi and Weinstein [142]. Using kurtoses, for the sphered data $\mathbf{z}$, the one unit FastICA algorithm has the following form:

$$\mathbf{w} \leftarrow \mathrm{E}\{\mathbf{z}(\mathbf{w}^T\mathbf{z})^3\} - 3\mathbf{w}. \tag{2.43}$$

After each iteration, the weighted vector $\mathbf{w}$ is normalized to remain on the constraint set. The final $\mathbf{w}$ gives one of the independent components as the linear combination $\mathbf{w}^T\mathbf{z}$.

Different properties of the one-unit version have been illustrated by computer simulations in [52] where the accuracy is also shown to be very good in most cases. More references, see [62, 154].

Chapter 3

Convergence of FastICA algorithms

3.1 Local convergence

FastICA is not a gradient descent but an approximative Newton method. An empirical comparison study [52] shows that FastICA outperformed some popular gradient descent type ICA methods [2, 8, 23]. The convergence of the FastICA algorithms was first proven by Hyvärinen & Oja in [69, 71, 75] for the one unit case. To guarantee the convergence, the function g, which is used in FastICA algorithms (2.38) is assumed to be a sufficiently smooth odd function, and the following theorem holds [69, 75]

Theorem 1 *Assume that the input data follows the ICA model with whitened data:* $\mathbf{z} = \mathbf{TAs}$ *where* $\mathbf{T}$ *is the whitening matrix, and that* G *is a sufficiently smooth even function. Then the local maxima (resp. minima) of* $E\{G(\mathbf{w}^T\mathbf{z})\}$ *under the constraint* $||\mathbf{w}|| = 1$ *include those rows of the mixing matrix* $\mathbf{TA}$ *such that the corresponding independent components* s_i *satisfy*

$$E\{s_i g(s_i) - g'(s_i)\} > 0 (\textit{resp. } < 0) \tag{3.1}$$

where $g(.)$ *is the derivative of* $G(.)$*, and* $g'(.)$ *is the derivative of* $g(.)$*.*

From this theorem, using Taylor series to expand the expection in FastICA algorithms (2.38), the convergence can be achieved. The details of the proof can be found in [75]. Further investigation of the proof revealed that the one unit algorithm has at least quadratic convergence speed. If $\mathrm{E}\{g''(\mathbf{s}_i)\} = 0$, which is true when the sources have symmetric distributions or the function $g(x) = x^3$, the convergence speed is cubic. For the kurtosis cost function case, the cubic convergence was also ensured in [103, 42].

The monotonic convergence and the convergence speed for a general cost function for the related gradient algorithm was considered in [134]. Considering algorithm (2.40), P.A. Regalia and E. Kofidis derived the step size bounds which ensure that FastICA has a monotonic convergence to a local extremum for any initial condition.

3.2 Global convergence

The global convergence of FastICA was first considered in [103]. E. Oja analyzed the kurtosis based FastICA algorithm with two sources. First, a simplifying linear transformation is made by considering the matrices

$$\mathbf{U} = \mathbf{W}(\mathbf{TA}),\ \bar{\mathbf{U}} = \bar{\mathbf{W}}(\mathbf{TA}). \tag{3.2}$$

Then, from eq. (2.2) we have $\mathbf{Wz} = \mathbf{W}(\mathbf{TA})\mathbf{s} = \mathbf{Us}$. Denoting the rows of matrix $\mathbf{U}$ by $\mathbf{u}_i^T$, we have $\mathbf{w}_i^T\mathbf{z} = \mathbf{u}_i^T\mathbf{s}$. Multiplying both sides of the general one-unit FastICA algorithm (2.38) from the left by $(\mathbf{TA})^T$ and remembering that $\mathbf{s} = (\mathbf{TA})^T\mathbf{z}$ yields now

$$\bar{\mathbf{u}}_i = \mathrm{E}\{\mathbf{s}g(\mathbf{u}_i^T\mathbf{s})\} - \mathrm{E}\{g'(\mathbf{u}_i^T\mathbf{s})\}\mathbf{u}_i. \tag{3.3}$$

This form of the equation is much easier to analyze than (2.38) because now the independent source vector $\mathbf{s}$ appears explicitly and we can make use of the independence of its elements.

The normalization (2.42) is actually unaffected by this transformation. Multiplying both sides of (2.42) from the right by $(\mathbf{TA})$ we get

$$\mathbf{W}(\mathbf{TA}) = (\bar{\mathbf{W}}(\mathbf{TA})(\mathbf{TA})^T\bar{\mathbf{W}}^T)^{-1/2}\bar{\mathbf{W}}(\mathbf{TA})$$

where we have used the fact that $\mathbf{TA}(\mathbf{TA})^T = \mathbf{I}$. So, we have

$$\mathbf{U} = (\bar{\mathbf{U}}\bar{\mathbf{U}}^T)^{-1/2}\bar{\mathbf{U}} \tag{3.4}$$

giving the equivalent orthonormalization for matrix $\mathbf{U}$.

Further, matrix $\mathbf{U} = \mathrm{diag}(\pm 1, \cdots, \pm 1)$, **PU** and **UP** with an orthogonal permutation matrix **P** are the fixed points of (3.3,3.4) under the assumption

$$\mathrm{E}\{s_i g(s_i) - g'(s_i)\} \neq 0 \tag{3.5}$$

for all sources $s_i, i = 1, ..., n$. E. Oja [103] stated that these fixed points are asymptotically stable and the order of convergence is three in kurtosis case. In **Publication III**, E. Oja and the current author proved that this is also true for general FastICA algorithms except the convergence speed is at least two instead of three.

In **Publication III**, the asymptotic stability of $\mathbf{U} = \mathrm{diag}(\pm 1, \cdots, \pm 1)$ means that asymptotically the solution is arbitrarily close to one of the sign combinations, but may flip from one combination to another one, within this set. Recently, Shen et al [144] analyzed this flipping phenomenon in a more rigorous way.

Any 2×2 orthogonal matrix **U** can be parameterized with a single parameter with the following form:

$$\mathbf{U} = \begin{pmatrix} x & \pm\sqrt{1-x^2} \\ \pm\sqrt{1-x^2} & -x \end{pmatrix},$$

where $|x| \leq 1$. Since the off-diagonal elements have the same sign, to guarantee orthogonality of the rows, only positive sign will be considered. After one iteration step (3.3) and followed by symmetric orthogonalization (3.4), the change of the elements u_{ij} of matrix **U** is the same for all the elements, following the equation [103]

$$x \leftarrow f(x) = \frac{x^3}{\sqrt{x^6 + (1-x^2)^3}}. \tag{3.6}$$

The function $f(x)$ has the unstable fixed points $\pm\sqrt{1/2}$ and stable fixed points $0, \pm 1$ with the order of convergence to these points three, see Figure 3.2.

In **Publication III**, the global convergence analysis for general FastICA algorithms with two sources can be treated in a similar way. Consider the FastICA algorithm (3.3) with symmetric orthogonalization (3.4). The changes of all the elements of matrix **U** in one step of iteration follow the same algorithm

$$x \leftarrow f(x) \tag{3.7}$$

where

$$f(x) = \begin{cases} \frac{\bar{u}_{11} - \bar{u}_{22}}{\sqrt{(\bar{u}_{11} - \bar{u}_{22})^2 + (\bar{u}_{21} + \bar{u}_{12})^2}}, & det(\bar{\mathbf{U}}) < 0, \\ \\ \frac{\bar{u}_{11} + \bar{u}_{22}}{\sqrt{(\bar{u}_{11} + \bar{u}_{22})^2 + (\bar{u}_{21} - \bar{u}_{12})^2}}, & det(\bar{\mathbf{U}}) > 0. \end{cases}$$

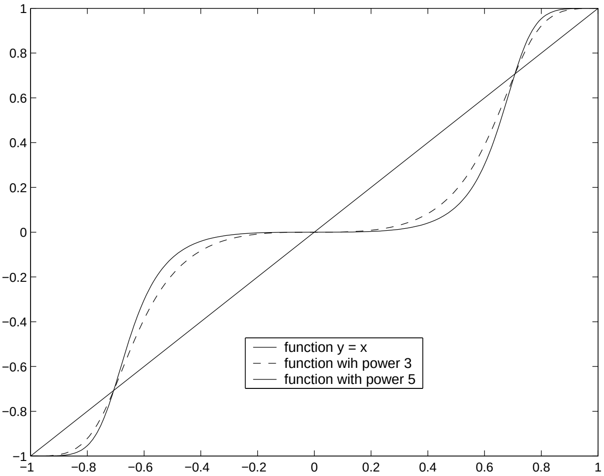


Figure 3.1: Comparison of the iteration functions $f(x)$ in the cubic and fifth degree case

where the matrix $\bar{\mathbf{U}}$ is the matrix after iteration (2.38) before symmetric orthogonalization (2.42), and $\bar{u}_{ij}$ are the elements of $\bar{\mathbf{U}}$. We use two popular functions of $g(x)$ to illustrate this general result. One is $g(x) = x^5$, and the other one is $g(x) = \tanh(x)$. Figure 3.1, 3.2, and 3.3 show the function $f(x)$ with these two nonlinearities and different source densities.

3.3 Statistical behavior of FastICA algorithms

In a practical situation, we always have finite samples. The theoretical expectations are replaced by sample averages, therefore, the limit of convergence is not exactly as for the ideal case. There is a residual error due to the finite sample size. A classical measure of error is the asymptotic (co)variance of the obtained estimator. The goal of designing an ICA algorithm is then to make this error as small as possible. For the FastICA algorithm, such an asymptotic performance analysis for a general cost function was proposed in [68, 150].

S. Douglas [42, 43] gave a statistical analysis of the convergence behavior of the FastICA algorithm using inter-channel interference (ICI). The linear ICA model with the given m-dimensional data $\mathbf{x}(n) = [x_1(n) \cdots x_m(n)]^T$, $1 \leq n \leq N$ is

$$\mathbf{x}(n) = \mathbf{A}\mathbf{s}(n), \tag{3.8}$$

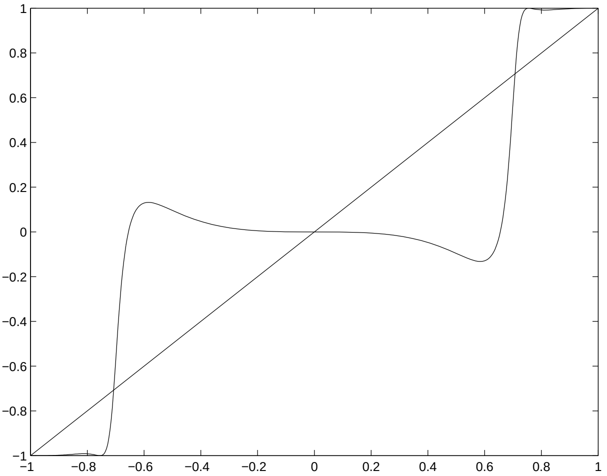


Figure 3.2: Plot of the function $f(x)$ with $g(x) = \tanh(x)$. Both s_1 and s_2 are binary distributed

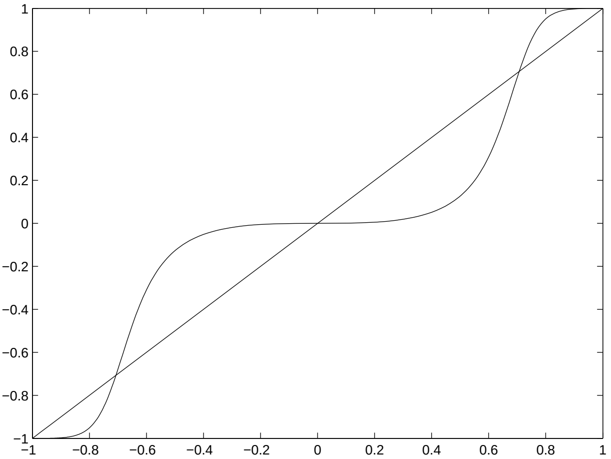


Figure 3.3: Plot of the function $f(x)$ with $g(x) = \tanh(x)$. s_1 is binary distributed, s_2 takes the values $\{-2, 2\}$ with equal probabilities 1/8, and zero with probability 3/4

where $\mathbf{A}$ is an unknown $m \times m$ mixing matrix, and $\mathbf{s}(n) = [s_1(n) \cdots s_m(n)]^T$ is the source. We need to find the matrix $\mathbf{W}$ such that $\mathbf{y}(n) = \mathbf{Wz}(n)$ contains the estimates of the independent sources, where $\mathbf{z}(n) = \mathbf{Tx}(n)$ is the whitening of the $\mathbf{x}(n)$. The FastICA algorithm is

$$\hat{\mathbf{w}}_{i(k+1)} = \frac{1}{N}\sum_{n=1}^{N} g(\mathbf{w}_{ik}^T\mathbf{z}(n))\mathbf{z}(n) - g'(\mathbf{w}_{ik}^T\mathbf{z}(n))\mathbf{w}_{ik} \tag{3.9}$$

$$\mathbf{w}_{i(k+1)} = \frac{\hat{\mathbf{w}}_{i(k+1)}}{||\hat{\mathbf{w}}_{i(k+1)}||} \tag{3.10}$$

where the index k means the kth iterative step. Let $\mathbf{C} = \mathbf{WTA}$, the inter-channel interference (ICI) is given by

$$ICI_k = \sum_{i=1}^{m}\left(\frac{\sum_{j=1}^{m}\mathbf{c}_{ijk}^2}{\max_{1\le j\le m}\mathbf{c}_{ijk}^2} - 1\right) \tag{3.11}$$

where the index k means the kth iteration step, m is the dimension of sources and c_{ijk} is the (i,j)th element of the combined system coefficient matrix.

S. Douglas [42] analyzed the behavior of the ICI and stated that all the stable stationary points of the single-unit FastICA algorithm under normalization conditions correspond to the desirable separating solutions. For the two equal kurtosis sources, he states [44] that the weight vector $\mathbf{w}$ for the FastICA algorithm evolves identically to that for the Rayleigh quotient iteration (RQI) applied to a symmetric matrix $\mathbf{\Gamma}$ diag $\{\lambda_\mathbf{1}, \lambda_\mathbf{2}\}\mathbf{\Gamma^T}$ when $\mathbf{\Gamma} = \mathbf{PA}$ is orthonormal, except for the sign changes associated with the alternating update directions in the RQI. Furthermore, he conjectured that the average behavior of ICI obeyed a "$\frac{1}{3}$" rule with the nonlinearty $g(y) = y^3$ [42], and proved the two-source case [44]:

$$\mathrm{E}\{ICI_k\} \approx \left(\frac{1}{3}\right)^k \mathrm{E}\{ICI_0\}. \tag{3.12}$$

In **Publication II**, a joint work of S. Douglas, Z.Yuan and E. Oja, the general case with kurtosis constrast function is solved. For m sources, the average ICI at iteration k is actually equal to

$$\mathrm{E}\{ICI_k\} = g_m(\kappa_1, \cdots, \kappa_m)(\frac{1}{3})^k + R(k, \kappa_1, \cdots, \kappa_m) \tag{3.13}$$

where κ_i is the kurtosis of the ith source, $g_m(\cdot)$ is a function that only depends on the kurtoses of the sources, and $R(k, \cdot)$ is approximately proportional to $(1/9)^k$ as $k \to \infty$.

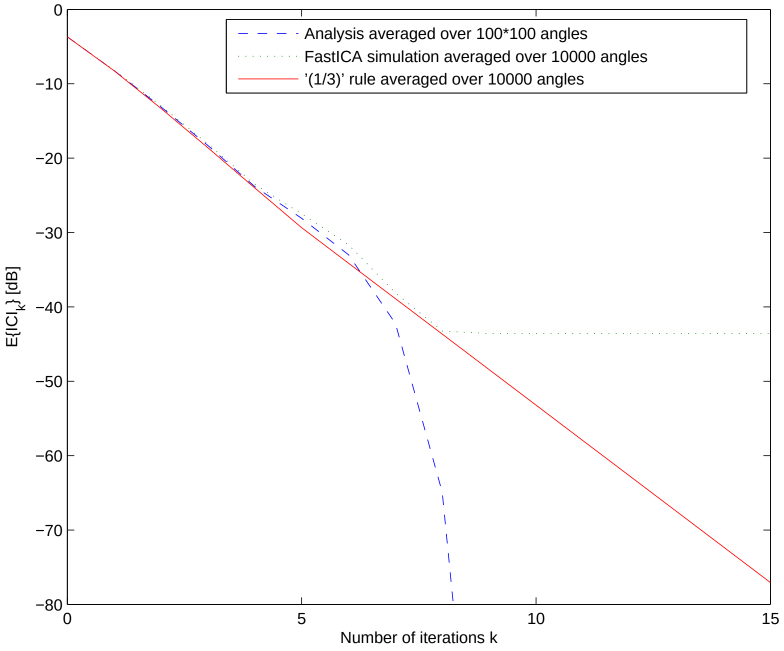


Figure 3.4: Evolutions of the average ICI as determined by various methods, m =4, [43].

Figure 3.4 shows the actual performance of FastICA and the prediction equation (3.12) with binary source. Because of the finite data records (10000 samples), FastICA algorithm exhibits a limit E$\{ICI_k\}$ value. However, during its convergence period, the prediction in equation (3.12) accurately describes the behavior of the FastICA algorithm.

A series of work [150, 80, 81] use Cramer-Rao lower bound (CRB) to analyze the performance of FastICA algorithms. The authors computed CRB for the demixing matrix of FastICA algorithm based on the score functions of the sources, which shows that FastICA is nearly statistically efficient. An improved algorithm called efficient FastICA (EFICA) was proposed, based on the concept of statistical efficiency. In EFICA, the asymptotic variance of the gain matrix, defined as the product of the estimated unmixing matrix and the original mixing matrix, attains the CRB which is the theoretical minimum for the variance.

Chapter 4

Nonnegative ICA

4.1 Nonnegative ICA algorithms

In the real world, many data have nonnegative properties. It is natural to consider adding nonnegative constraint on the linear ICA model. The combination of non-negativity and independence assumptions on the sources is refered as *non-negative independent component analysis* [124, 126, 128]. Recently, Plumbley [124, 125, 126, 127, 128] considered the non-negativity assumption on the sources and introduced an alternative way of approaching the ICA problem. Using the probability function $Pr(\cdot)$, he made the following definitions:

Definition 1 *A source s is called* non-negative *if $Pr(s < 0) = 0$.*

Definition 2 *A* non-negative *source s is* well-grounded *if $Pr(s < \delta) > 0$ for any $\delta > 0$.*

From the definition, a *well-grounded non-negative* source s has non-zero pdf all the way down to zero. Using these concepts, Plumbley proved the following key result [124]:

Theorem 1. Suppose that $\mathbf{s}$ is a vector of non-negative well-grounded independent unit-variance sources $s_i,\ i = 1, ..., n$, and $\mathbf{y} = \mathbf{Q}\mathbf{s}$ where $\mathbf{Q}$ is a square orthonormal rotation, i.e. $\mathbf{Q}^T\mathbf{Q} = \mathbf{I}$. Then $\mathbf{Q}$ is a permutation matrix, i.e. the elements y_j of $\mathbf{y}$ are a permutation of the sources s_i, if and only if all y_j are non-negative.

The result of Theorem 1 can be used for a simple solution of the non-negative ICA problem. The sources of course are unknown, and $\mathbf{Q}$ cannot be found directly. However, it is a simple fact that an arbitrary rotation of $\mathbf{s}$ can also be expressed as a rotation of a pre-whitened observation vector. Denote it by $\mathbf{z} = \mathbf{Vx}$ with $\mathbf{V}$ the whitening matrix. Assume that the dimensionality of $\mathbf{z}$ has been reduced to that of $\mathbf{s}$ in the whitening, which is always possible in the overdetermined case (number of sensors is not smaller than number of sources).

It holds now $\mathbf{z} = \mathbf{TAs}$. Because both $\mathbf{z}$ and $\mathbf{s}$ have unit covariance (for $\mathbf{s}$, this is assumed in Theorem 1), the matrix $\mathbf{TA}$ must be square orthogonal, although $\mathbf{s}$ and $\mathbf{z}$ have non-zero means. We can write

$$\mathbf{y} = \mathbf{Qs} = \mathbf{Q}(\mathbf{TA})^T\mathbf{z} = \mathbf{Wz}$$

where the matrix $\mathbf{W}$ is a new parametrization of the problem. The key fact is that $\mathbf{W}$ is orthogonal, because it is the product of two orthogonal matrices $\mathbf{Q}$ and $(\mathbf{TA})^T$.

By Theorem 1, to find the sources, it now suffices to *find an orthogonal matrix* $\mathbf{W}$ *for which* $\mathbf{y} = \mathbf{Wz}$ *is non-negative.* The elements of $\mathbf{y}$ are then the sources. This brings the additional benefit over other ICA methods that as a result we will always have a positive permutation of the sources, since both the $\mathbf{s}$ and $\mathbf{y}$ are non-negative. The sign ambiguity present in standard ICA vanishes here.

A suitable cost function for actually finding the rotation was also suggested by Plumbley [124, 126, 128]. It is constructed as follows: consider the output truncated at zero, $\mathbf{y}^+ = (y_1^+, ..., y_n^+)$ with $y_i^+ = \max(0, y_i)$, and use $\hat{\mathbf{z}} = \mathbf{W}^T\mathbf{y}^+$ as a reestimate of $\mathbf{z} = \mathbf{W}^T\mathbf{y}$. Then we can construct a suitable cost function given by

$$J(\mathbf{W}) = \mathrm{E}\{\|\mathbf{z} - \hat{\mathbf{z}}\|^2\} = \mathrm{E}\{\|\mathbf{z} - \mathbf{W}^T\mathbf{y}^+\|^2\}. \tag{4.1}$$

Due to the orthogonality of matrix $\mathbf{W}$, this is in fact equal to

$$J(\mathbf{W}) = \mathrm{E}\{\|\mathbf{y} - \mathbf{y}^+\|^2\} = \sum_{i=1}^{n} \mathrm{E}\{\min(0, y_i)^2\}. \tag{4.2}$$

Obviously, the value will be zero if $\mathbf{W}$ is such that all the y_i are positive.

The minimization of this cost function gives various numerical algorithms [126, 128, 108, 109]. In fact, minimization of the least mean squared reconstruction error (4.1) has been proposed as an objective principle for many neural-network principal component analysis (PCA) algorithms and PCA subspace algorithms [164]. In particular, this led to the nonlinear PCA algorithm [**?**], where $\mathbf{W}(t{+}1) = \mathbf{W}(t) + \Delta\mathbf{W}$ with

$$\Delta\mathbf{W} = \eta_t g(\mathbf{y})[\mathbf{z}^T - g(\mathbf{y}^T)\mathbf{W}] \tag{4.3}$$

where $g(\mathbf{y}^T) = (g(y_1), \cdots, g(y_n))^T$, and g is a nonlinear function. Algorithm (4.3) is a nonlinear version of the Oja and Karhunen PCA subspace algorithm [107, 105], which used this algorithm with $g(y) = y$.The nonlinear PCA algorithm was shown to perform ICA on whitened data, if g is an odd, twice-differentiable function [106].

Thus an obvious suggestion for our nonnegative ICA problem is the nonlinear PCA algorithm (5) with the rectification nonlinearity $g(y) = y^+ = \max(y, 0)$, giving us

$$\Delta\mathbf{W} = \eta_t \mathbf{y}^+[\mathbf{z}^T - (\mathbf{y}^+)^T\mathbf{W}] \tag{4.4}$$

which we call the nonnegative PCA algorithm. The rectification nonlinearity $g(y) = \max(y, 0)$ is neither an odd function, nor is it twice differentiable, so the standard convergence proof for nonlinear PCA algorithms does not apply. Mao [100] gave a global convergence proof of the discrete time “non-negative PCA” under certain assumptions.

In fact, the cost function (4.2) is a Liapunov function for a certain matrix flow in the Stiefel manifold, providing a global convergence [109, 100, 101].

However, the problem with the gradient type of learning rules is slow speed of convergence. It would be tempting therefore to develop a ”fast” numerical algorithm for this problem, perhaps along the lines of the well-known FastICA method [75]. In **Publication I**, we introduced such an algorithm with convergence analysis. A review is given in the following.

4.2 The nonnegative FastICA algorithm

The nonnegative FastICA algorithm, developed in **Publication I**, combined the nonnegative constraint and FastICA algorithm. Non-centered but whitened data $\mathbf{z}$ is used, which satisfies $\mathrm{E}\{(\mathbf{z}-\mathrm{E}\{\mathbf{z}\})(\mathbf{z}-\mathrm{E}\{\mathbf{z}\})^T\} = \mathbf{I}$ to keep the nonnegativity. A control parameter μ is added on the FastICA update rule (2.38), giving the following update rule:

$$\mathbf{w} \leftarrow \mathrm{E}\{(\mathbf{z} - \mathrm{E}\{\mathbf{z}\})g_-(\mathbf{w}^T\mathbf{z})\} - \mu\mathrm{E}\{g'_-(\mathbf{w}^T\mathbf{z})\}\mathbf{w}, \tag{4.5}$$

where g_- is the function

$$g_-(y) = -min(0, y) = \begin{cases} -y, & y < 0 \\ 0, & y \geq 0. \end{cases}$$

and g'_- is the derivative of g_-. This formulation shows the similarity to the classical FastICA algorithm. Substituting function g_- into (4.5) simplifies the terms; for example, $\mathrm{E}\{g'_-(\mathbf{w}^T\mathbf{z})\} = -\mathrm{E}\{1|\mathbf{w}^T\mathbf{z} < 0\}P\{\mathbf{w}^T\mathbf{z} < 0\}$, and $\mathrm{E}\{(\mathbf{z} - \mathrm{E}\{\mathbf{z}\})g_-(\mathbf{w}^T\mathbf{z})\} = \mathrm{E}\{(\mathbf{z} - \mathrm{E}\{\mathbf{z}\})(-\mathbf{w}^T\mathbf{z})|\mathbf{w}^T\mathbf{z} < 0\}P\{\mathbf{w}^T\mathbf{z} < 0\}$. The scalar $P\{\mathbf{w}^T\mathbf{z} < 0\}$, appearing in both terms, can be dropped because the vector $\mathbf{w}$ will be normalized anyway.

In (4.5), μ is an adjustable parameter which keeps $\mathbf{w}$ nonnegative. It is determined by:

$$\mu = \min_{\{\mathbf{z}:\mathbf{z}\in\Delta)\}} \frac{\mathrm{E}\{(\mathbf{z} - \mathrm{E}\{\mathbf{z}\})\mathbf{w}^T\mathbf{z}|\mathbf{w}^T\mathbf{z} < 0\}^T\mathbf{z}}{\mathrm{E}\{1|\mathbf{w}^T\mathbf{z} < 0\}\mathbf{w}^T\mathbf{z}}. \tag{4.6}$$

There the set $\Delta = \{\mathbf{z} : \mathbf{z}^T\mathbf{z}(0) = 0\}$, with $\mathbf{z}(0)$ the vector satisfying $||\mathbf{z}(0)|| = 1$ and $\mathbf{w}^T\mathbf{z}(0) = \max(\mathbf{w}^T\mathbf{z})$. Computing this parameter is computationally somewhat heavy, but on the other hand, now the algorithm converges in a fixed number of steps.

The nonnegative FastICA algorithm is shown below.

1. Whiten the data to get vector $\mathbf{z}$.
2. Set counter $p \leftarrow 1$.
3. Choose an initial vector $\mathbf{w}_p$ of unit norm, and orthogonalize it as

   $$\mathbf{w}_p \leftarrow \mathbf{w}_p - \sum_{j=1}^{p-1}(\mathbf{w}_p^T\mathbf{w}_j)\mathbf{w}_j$$

   and then normalize by $\mathbf{w}_p \leftarrow \mathbf{w}_p/||\mathbf{w}_p||$.
4. If $\max_{\mathbf{z}\neq 0}(\mathbf{w}_p^T\mathbf{z}) \leq 0$, update $\mathbf{w}_p$ by $-\mathbf{w}_p^T$.
5. If $\min_{\mathbf{z}\neq 0}(\mathbf{w}_p^T\mathbf{z}) \geq 0$, update $\mathbf{w}_p$ by $\mathbf{w}_p \leftarrow \mathbf{w}(r)(\mathbf{w}(r)^T\mathbf{w}_p)\mathbf{w}_p$, where $\mathbf{w}(r)$ is the vector in the null space $null(\mathbf{Z})$ with $\mathbf{Z} =: \{\mathbf{z} \neq 0 : \mathbf{w}_p^T\mathbf{z} = 0\}$.
6. Update $\mathbf{w}_p$ by the equation (4.5), replacing expectations by sample averages.
7. Let $\mathbf{w}_p \leftarrow \mathbf{w}_p/||\mathbf{w}_p||$.

8. If $\mathbf{w}_p$ has not converged, go back to step (4).
9. Set $p \leftarrow p+1$. If $p < n$ where n is the number of independent components, go back to step (3).

4.3 Analysis of the algorithm

To analyse the convergence of the above nonnegative FastICA algorithm, the following orthogonal variable change is useful:

$$\mathbf{q} = \mathbf{A}^T \mathbf{T}^T \mathbf{w} \tag{4.7}$$

where $\mathbf{A}$ is the mixture matrix and $\mathbf{T}$ is the whitening matrix. Then

$$\mathbf{w}^T \mathbf{z} = \mathbf{q}^T (\mathbf{TA})^T (\mathbf{TAs}) = \mathbf{q}^T \mathbf{s}. \tag{4.8}$$

Note that matrix $\mathbf{TA}$ is orthogonal.

By theorem 1, our goal is to find the orthogonal matrix $\mathbf{W}$ such that $\mathbf{Wz}$ is non-negative. This is equivalent to find a permutation matrix $\mathbf{Q}$, whose rows will be denoted by vectors $\mathbf{q}^T$, such that $\mathbf{Qs}$ is non-negative. In the space of the $\mathbf{q}$ vectors, the convergence result of the non-negative FastICA algorithm must be a unit vector $\mathbf{q}$ with exactly one entry nonzero and equal to one.

Using the above transformation in eq. (4.7), the definition of the function g_-, and the parameter μ, the update rule (4.5) for the variable $\mathbf{q}$ becomes

$$\mathbf{q} \leftarrow \mu \mathrm{E}\{1 | \mathbf{q}^T \mathbf{s} < 0\} \mathbf{q} - \mathrm{E}\{(\mathbf{s} - \mathrm{E}\{\mathbf{s}\})(\mathbf{q}^T \mathbf{s}) | \mathbf{q}^T \mathbf{s} < 0\}. \tag{4.9}$$

The idea to prove the convergence of non-negative FastICA algorithm, is to show that after each iteration, the updated vector $\mathbf{q}$ keeps the old zero entries zero and gains one more zero entry. Therefore, within $n-1$ iteration steps, the vector $\mathbf{q}$ is updated to be a unit vector e_i for certain i. With total iterative steps $\sum_{i=1}^{n-1} i = n(n-1)/2$, the permutation matrix $\mathbf{Q}$ formed. The details of the proof can be found in the Publication.

It is obvious that most of the functions chosen in the classic FastICA also work in this nonnegative FastICA algorithm. However, the adjustable parameter μ will change according to the choice of the function. One exception is the function $g(y) = y$, since its original function $G(y)$ is an even and smooth function, which satisfies the requirement for the function $G(y)$ in FastICA. However, it can only find one vector $\mathbf{w}$ such that $\mathbf{w}^T\mathbf{z}$ is nonnegative, and therefore fails to separate the independent components.

We first make the change of variable as before from $\mathbf{w}$ to $\mathbf{q}$. Since $g(\mathbf{q}^T\mathbf{s}) = \mathbf{q}^T\mathbf{s}$ and $g'(\mathbf{q}^T\mathbf{s}) = 1$, then the ith element of $\mathbf{q}$ is updated by

$$\mathbf{q}_i \leftarrow \mathrm{E}\{\mathbf{s}_i(\mathbf{q}^T\mathbf{s})\} - \mathbf{q}_i. \tag{4.10}$$

Simplify the right-hand side of equation (4.10), note that all the sources are unit variance, we get

$$\begin{aligned}
\mathrm{E}\{\mathbf{s}_i(\mathbf{q}^T\mathbf{s})\} - \mathbf{q}_i &= \mathrm{E}\{\mathbf{s}_i \sum_{j=1}^{n} \mathbf{q}_j\mathbf{s}_j)\} - \mathbf{q}_i && (4.11)\\
&= \mathrm{E}\{\mathbf{s}_i\mathbf{q}_i\mathbf{s}_i)\} + \mathrm{E}\{\mathbf{s}_i \sum_{j=1,j\neq i}^{n} \mathbf{q}_j\mathbf{s}_j)\} - \mathbf{q}_i && (4.12)\\
&= \mathrm{E}\{\mathbf{s}_i\} \sum_{j=1,j\neq i}^{n} \mathbf{q}_j\mathrm{E}\{\mathbf{s}_j\}. && (4.13)
\end{aligned}$$

The last equation uses the fact that all the sources $\mathbf{s}_i$ for $i = 1, \cdots, n$ are independent. Then after normalization, the updated vector $\mathbf{q}$ does not depend on the initial data. Therefore the matrix $\mathbf{W}$ cannot be found.

The computation of each iteration takes more time compared to FastICA. During each iteration, the computational differences compared to classic FastICA come from step 4, 5 and 6. The step 4 is a simple value assignment. In step 5, we need to calculate the value of $\mathbf{w}_p^T\mathbf{z}$ once, and solve an $m \times n$ linear equation (m is the number of vectors in the source space $\{\mathbf{s} \neq 0 : \mathbf{q}(k)^T\mathbf{s} = 0\}$). Step 6 is the main update rule, just as in FastICA, and the extra calculation is the expectation $\mathrm{E}\{\mathbf{z}-\mathrm{E}\{\mathbf{z}\}\}$ and the parameter μ. The total computation complexity takes $\mathcal{O}(N^2)$ with N the number of samples, more time compared to FastICA in each step. However, one should note that as the analysis shows above, the total number of iteration steps of our algorithm are less than or equal to $n(n-1)/2$.

Chapter 5

Nonnegative matrix factorization

5.1 Introduction

In the linear ICA model, its matrix form can be mathematically explained as the factorization of data matrix $\mathbf{X}$ into two matrices $\mathbf{A}$ and $\mathbf{S}$, where $\mathbf{S}$ is the basis matrix and $\mathbf{A}$ is a coefficient matrix. Often the data to be analyzed is nonnegative, and the low rank data are further required to be comprised of nonnegative values in order to avoid contradicting physical realities. So the nonnegative constraint is proposed on all three matrices. This new model leads to a method called Non-negative Matrix Factorization technique (NMF). Classical tools such as factor analysis and principal component analysis cannot guarantee to maintain the nonnegativity. The approach of finding reduced rank nonnegative factors to approximate a given nonnegative data matrix thus becomes a natural choice.

The idea of Nonnegative matrix factorization (NMF) can be traced back to a paper of P. Paatero and U. Tapper [116] in 1994, which was named Positive matrix factorization (PMF). Suppose $\mathbf{V}$ is a positive $m \times n$ matrix, Paatero and Tapper advocated a positive low-rank approximation $\mathbf{WH}$ by optimizing the functional

$$\min_{\mathbf{W},\mathbf{H}\geq 0} ||\mathbf{A} \cdot (\mathbf{V} - \mathbf{WH})||_F. \tag{5.1}$$

where matrix $\mathbf{A}$ is the weighted matrix whose elements are associated to the elements of the matrix $\mathbf{V} - \mathbf{WH}$, $\cdot$ denotes the Hadamard (also known as the Schur or elementwise) product. Paatero and Tapper proposed an alternating least squares algorithm (ALS) by fixing one matrix and solving the optimization with respect to the other, and reversing the matrices. Later, Paatero developed a series of algorithms [111, 112, 113] using a longer product of matrices to replace the approximant $\mathbf{WH}$.

Nonnegative matrix factorization gained more applications and became popular because of Lee and Seung's work [90, 91, 92]. Lee and Seung introduced the NMF model defined as following[1]:

Given a nonnegative $m \times n$ data matrix $\mathbf{V}$, find nonnegative $m \times r$ matrix $\mathbf{W}$ and $r \times n$ matrix $\mathbf{H}$ with the reduced rank r such that the product of $\mathbf{W}$ and $\mathbf{H}$ minimizes

$$f(\mathbf{W}, \mathbf{H}) = \frac{1}{2}||\mathbf{V} - \mathbf{WH}||^2. \tag{5.2}$$

Here, $\mathbf{W}$ is often thought of as the basis matrix and $\mathbf{H}$ as the mixing matrix associated with the data in $\mathbf{V}$. The measure $|| \cdot ||$ could be any matrix norm, or other measurements. The rank r is often chosen such that $r << \min(m, n)$. An appropriate decision on the value of r is critical in practice, but the choice of r is very often problem dependent.

Important challenges affecting the numerical minimization of (5.2) include the existence of local minima due to the non-convexity of $f(\mathbf{W}, \mathbf{H})$ in both $\mathbf{W}$ and $\mathbf{H}$, and perhaps more importantly the lack of a unique solution which can be easily seen by considering $\mathbf{WDD}^{-1}\mathbf{H}$ for any nonnegative invertible matrix $\mathbf{D}$ with nonnegative inverse $\mathbf{D}^{-1}$.

The objective function (5.2) of the general NMF model can be modified in several ways to reflect the application need. For example, penalty terms can be added in order to gain more localization or enforce sparsity; or more constraints such as sparseness can be imposed.

5.2 The truncated singular value decomposition

The singular value decomposition (SVD) is a classic technique in numerical linear algebra. For a given $m \times n$ matrix $\mathbf{V}$, its n columns are the data items, for example, a set of images that have been vectorized by row-by-row scanning. Then m is the number of pixels in any given image. The Singular Value Decomposition

[1] For clarity, we use here the same notation as in the original NMF theory by Lee and Seung

(SVD) for matrix $\mathbf{V}$ is

$$\mathbf{V} = \mathbf{Q}\mathbf{D}\mathbf{R}^T, \tag{5.3}$$

where $\mathbf{Q}$ $(m \times m)$ and $\mathbf{R}$ $(n \times m)$ are orthogonal matrices consisting of eigenvectors of $\mathbf{V}\mathbf{V}^T$ and $\mathbf{V}^T\mathbf{V}$, respectively, and $\mathbf{D}$ is a diagonal $m \times m$ matrix where the diagonal elements are the ordered singular values of $\mathbf{V}$.

Choosing the r largest singular values of matrix $\mathbf{V}$ to form a new diagonal $r \times r$ matrix $\hat{\mathbf{D}}$, with $r < m$, we get the compressive SVD matrix $\mathbf{U}$ with given rank r,

$$\mathbf{U} = \hat{\mathbf{Q}}\hat{\mathbf{D}}\hat{\mathbf{R}}^T. \tag{5.4}$$

Now both eigenvector matrices $\hat{\mathbf{Q}}$ and $\hat{\mathbf{R}}$ have only r columns, corresponding to the r largest eigenvalues. The compressive SVD gives the best approximation (in Frobenius matrix norm) of the matrix $\mathbf{V}$ with the given compressive rank r [16, 54].

In the case that we consider here, all the elements of the data matrix $\mathbf{V}$ are *non-negative*. Then the above compressive SVD matrix $\mathbf{U}$ fails to keep the non-negative property. In order to further approximate it by a non-negative matrix, the following truncated SVD (tSVD) is suggested. We simply truncate away the negative elements by

$$\hat{\mathbf{U}} = \frac{1}{2}(\mathbf{U} + abs(\mathbf{U})) \tag{5.5}$$

where the absolute value is taken element by element.

However, it turns out that typically the matrix $\hat{\mathbf{U}}$ in (5.5) has higher rank than $\mathbf{U}$. Truncation destroys the linear dependences that are the reason for the low rank. In order to get an equal rank, we have to start from a compressive SVD matrix $\mathbf{U}$ with lower rank than the given r. To find the truncated matrix $\hat{\mathbf{U}}$ with the compressive rank r, we search all the compressive SVD matrices $\mathbf{U}$ with the rank from 1 to r and form the corresponding truncated matrices. The one with the largest rank that is less than or equal to the given rank r is the truncated matrix $\hat{\mathbf{U}}$ what we choose as the final non-negative approximation. This matrix can be used as a baseline in comparisons, and also as a starting point in iterative improvements. We call this method truncated SVD (tSVD).

Note that the tSVD only produces the non-negative low-rank approximation $\hat{\mathbf{U}}$ to the data matrix $\mathbf{V}$, but does not give a separable expansion for basis vectors and weights, like the usual SVD expansion.

5.3 The fundamental NMF algorithms

Quite many numerical algorithms have been developed for solving the NMF. The methodologies adapted are following more or less the principles of alternating direction iterations, the projected Newton, the reduced quadratic approximation, and the descent search. Specific implementations generally can be categorized into alternating least squares (ALS) algorithms [116], multiplicative update algorithms [91, 92], gradient descent algorithm, and hybrid algorithm [122, 123]. Some general assessments of these methods can be found in [87, 151]. Actually, the multiplicative update algorithm can also be considered as a gradient descent method. Below we will briefly have a look at gradient descent methods and ALS methods.

5.3.1 NMF algorithms by Lee and Seung

One of the fundamental NMF algorithms developed by Lee and Seung [91] based on the optimal equation (5.2) with Frobenius norm is

$$\mathbf{W}_{ia} \leftarrow \mathbf{W}_{ia} \sum_{\mu} \frac{\mathbf{V}_{i\mu}}{(\mathbf{WH})_{i\mu}} \mathbf{H}_{a\mu}, \mathbf{W}_{ia} \leftarrow \frac{\mathbf{W}_{ia}}{\sum_j \mathbf{W}_{ja}} \tag{5.6}$$

$$\mathbf{H}_{a\mu} \leftarrow \mathbf{H}_{a\mu} \sum_{i} \mathbf{W}_{ia} \frac{\mathbf{V}_{i\mu}}{(\mathbf{WH})_{i\mu}}. \tag{5.7}$$

In practice, a small constant 10^{-9} in each update rule is added to the denominator to avoid division by zero. Lee and Seung used the gradient descent to form the above multiplicative update algorithm by choosing the right step size. Lee and Seung [92] claimed the convergence of the above algorithm, which is not true. Lin [95, 96] pointed out their error, and proposed some modified algorithms of Lee and Seung's method. Gonzalez and Zhang [55] presented numerical examples showing that Lee and Seungs algorithm [92] fails to approach a stationary point.

Most of gradient descent methods like the above multiplicative update algorithm take a step in the direction of the negative gradient, the direction of steepest descent. Since the step size parameters of **W** and **H** vary depending on the algorithm, the trick comes in choosing the values for the stepsizes of **W** and **H**. Some algorithms initially set these stepsize values to 1, then multiply them by one-half at each subsequent iteration [13]. This is simple, but not ideal because there is no restriction that keeps elements of the updated matrices **W** and **H** from becoming negative. A common practice employed by many gradient descent algorithms is a simple projection step [140, 64, 26, 121]. That is, after each update rule, the updated matrices are projected to the nonnegative orthant by setting all negative elements to the nearest nonnegative value, 0.

5.3.2 Alternating least squares (ALS) algorithms

Another class of the fundamental NMF algorithms is the alternating least squares (ALS). ALS algorithms were first used by Paatero [116]. The basic ALS Algorithm for NMF is

1. Initialize $\mathbf{W}$ as random $m \times r$ matrix
2. (ls) Solve for $\mathbf{H}$ in matrix equation $\mathbf{W}^T\mathbf{W}\mathbf{H} = \mathbf{W}^T\mathbf{V}$.

   (nonneg) Set all negative elements in $\mathbf{H}$ to 0.

   (ls) Solve for $\mathbf{W}$ in matrix equation $\mathbf{H}\mathbf{H}^T\mathbf{W}^T = \mathbf{H}\mathbf{V}^T$.

   (nonneg) Set all negative elements in $\mathbf{W}$ to 0.
3. Repeat step 2 until convergence.

In this algorithm, a least squares step is followed by another least squares step in an alternating fashion, thus giving rise to the ALS name. Although the function (5.2) is not convex in both $\mathbf{W}$ and $\mathbf{H}$, it is convex in either $\mathbf{W}$ or $\mathbf{H}$. Thus, given one matrix, the other matrix can be found with a simple least squares computation [86].

In the above algorithm, a simple projection step, which sets all negative elements resulting from the least squares computation to 0, is used to keep nonnegativity. This simple technique also has a few added benefits. Of course, it aids sparsity. Moreover, it allows the iterates some additional flexibility not available in other algorithms, especially those of the multiplicative update class. One drawback of the multiplicative algorithms is that once an element in $\mathbf{W}$ or $\mathbf{H}$ becomes 0, it must remain 0. This locking of 0 elements is restrictive, meaning that once the algorithm starts heading down a path towards a fixed point, even if it is a poor fixed point, it must continue in that vein. The ALS algorithms are more flexible, allowing the iterative process to escape from a poor path.

Depending on the implementation, ALS algorithms can be very fast. The implementation shown above requires significantly less work than other NMF algorithms and slightly less work than an SVD implementation. Improvements to the basic ALS algorithm appear in [113, 87].

5.4 NMF algorithms with application-dependent auxiliary constraints

Since the popularization of the NMF problem by Lee and Seung, a great deal of work has been devoted to the analysis, extension, and application of NMF algorithms in science, engineering and medicine. The NMF problem has been cast into alternate formulations by various authors. One of the improvements is to develop the algorithms by using different objective functions. Lee and Seung [92] provided an information theoretic formulation based on the Kullback-Leibler divergence of **V** from **WH**. Dhillon and Sra [39] generalized the NMF methods with Bregman divergence. Cichocki et al. [28] have proposed cost functions based on Csisz*á*r's φ-divergence. Wang et al. [156] propose a formulation that enforces constraints based on Fisher linear discriminant analysis for improved determination of spatially localized features. Guillamet et al. [56] have suggested the use of a diagonal weight matrix **Q** in a new factorization model, $\mathbf{VQ} \approx \mathbf{WHQ}$ in an attempt to compensate for feature redundancy in the columns of **W**.

Other approaches were done by proposing alternative cost function formulations. Smoothness constraints have been used to regularize the computation of spectral features in remote sensing data [123, 121]. Chen and Cichocki [24] employed temporal smoothness and spatial correlation constraints to improve the analysis of EEG data for early detection of Alzheimer's disease. Hoyer [63, 64] employed sparsity constraints on either **W** or **H** to improve local rather than global representation of data. The extension of NMF to include such auxiliary constraints is problem dependent and often reflects the need to compensate for the presence of noise or other data degradations in **V**.

Below, we will give a brief review on some of the modified NMF algorithms.

5.4.1 NMF algorithms with sparsity constraints

The Nonnegative Sparse Coding (NNSC) [63] method is intended to decompose multivariate data into a set of positive sparse components by using theory inherited from Linear Sparse Coding. Combining a small reconstruction error with a sparseness criterion, Patrik Hoyer's NNSC [63] defines the objective function as:

$$E(\mathbf{V}, \mathbf{WH}) = \frac{1}{2}||\mathbf{V} - \mathbf{WH}||^2 + \lambda \sum_{ij} f(\mathbf{H}_{ij}) \tag{5.8}$$

where the form of f defines how sparseness on $\mathbf{H}$ is measured and λ controls the trade-off between sparseness and accuracy of the reconstruction. In [63], the author used a linear activation penalty function to measure the sparseness, leading to the following objective function:

$$E(\mathbf{V}, \mathbf{WH}) = \frac{1}{2}||\mathbf{V} - \mathbf{WH}||^2 + \lambda \sum_{ij} \mathbf{H}_{ij}. \tag{5.9}$$

The iterative algorithm is

$$\mathbf{W} \leftarrow \mathbf{W} - \mu(\mathbf{WH} - \mathbf{V})\mathbf{H}^T. \tag{5.10}$$

$$\mathbf{H}_{ij} \leftarrow \mathbf{H}_{ij} \frac{(\mathbf{W}^T\mathbf{V})_{ij}}{((\mathbf{W}^T\mathbf{WH})_{ij} + \lambda)} \tag{5.11}$$

where $\mu > 0$ denotes the step-size. After each iteration for $\mathbf{W}$, any negative values in $\mathbf{W}$ are set to zero, and then each column of $\mathbf{W}$ is normalized.

Later, Hoyer extended the above NNSC to NMF with sparseness constraints (NMFSC) [64]. NMFSC minimizes

$$E(\mathbf{W}, \mathbf{H}) = ||\mathbf{V} - \mathbf{WH}||^2 \tag{5.12}$$

under the optional sparseness constraints

$$sparseness(\mathbf{w}_i) = \mathbf{S}_{\mathbf{w}}, \forall i \tag{5.13}$$

$$sparseness(\mathbf{h}_i) = \mathbf{S}_{\mathbf{h}}, \forall i \tag{5.14}$$

where $\mathbf{S}_{\mathbf{w}}$ and $\mathbf{S}_{\mathbf{h}}$ are the desired sparseness values for $\mathbf{W}$ and $\mathbf{H}$, respectively, and are user-defined parameters. The sparseness criteria proposed in [64] uses a measure based on the relationship between the L1 and L2 norm of a given n dimensional vector $\mathbf{x}$:

$$sparseness(\mathbf{x}) = \frac{\sqrt{n} - (\sum |\mathbf{x}_i|)/\sqrt{\sum \mathbf{x}_i^2}}{\sqrt{n} - 1}. \tag{5.15}$$

The details of NMFSC algorithm can be found in Hoyer [64].

Liu [97] modified the above method [63] by simply using a divergence measure instead of Euclidean least-square type functional. Thus, the sparse NMF functional is:

$$D(\mathbf{V}, \mathbf{WH}) = \sum_{i=1}^{p} \sum_{j=1}^{n} \left(\mathbf{V}_{ij} \ln \frac{\mathbf{V}_{ij}}{(\mathbf{WH})_{ij}} - \mathbf{V}_{ij} + (\mathbf{WH})_{ij} \right) + \alpha \sum_{ij} \mathbf{H}_{ij} \tag{5.16}$$

for $\alpha \geq 0$.

This method forces sparseness via minimizing the sum of all $\mathbf{H}_{ij}$. The update rule for matrices $\mathbf{W}$ and $\mathbf{H}$ are:

$$\mathbf{H}_{ab} \leftarrow \mathbf{H}_{ab} \frac{\sum_{i=1}^{p}(\mathbf{W}_{ia}\mathbf{V}_{ib})/\sum_{k=1}^{q}(\mathbf{W}_{ik}\mathbf{H}_{kb})}{1+\alpha} \tag{5.17}$$

$$\mathbf{W}_{ab} \leftarrow \mathbf{W}_{ab} \frac{\sum_{j=1}^{n}(\mathbf{H}_{bj}\mathbf{V}_{aj})/\sum_{k=1}^{q}(\mathbf{W}_{ak}\mathbf{H}_{kj})}{\sum_{j=1}^{n}\mathbf{H}_{bj}}, \mathbf{W}_{ab} \leftarrow \frac{\mathbf{W}_{ab}}{\sum_{j}\mathbf{W}_{jb}} \tag{5.18}$$

Furthermore, Principal Sparse Non-Negative Matrix Factorization (PSNMF) (Hu et al. [65]) defined a support measurement for every column vector w_i of $\mathbf{W}$:

$$support(w_i) = \frac{\sum_j \mathbf{H}_{ij}}{\sum_{ij}\mathbf{H}_{ij}} \tag{5.19}$$

Then selecting the principal k rules as the Ratio Rules by minimizing

$$\min_k \left(\frac{\sum_{i=1}^{k} support(w_i)}{\sum_{i=1}^{m} support(w_i)} > threshold \right) \tag{5.20}$$

5.4.2 Local NMF

To gain more localization information, Li [94] developed a method called Local NMF (LNMF). LNMF requires three additional constraints on the NMF basis:

1. A basis component should not be further decomposed into more components, so as to minimize the number of basis components required to represent $\mathbf{V}$. That is, we wish $\sum_i \mathbf{W}_{ij}^2$ should be as small as possible with $\sum_i \mathbf{W}_{ij} = 1$, so that $\mathbf{W}$ contains as many non-zero elements as possible. This can be imposed by minimizing $\sum_i(\mathbf{W}^T\mathbf{W})_{ii}$.

2. Different bases should be as orthogonal as possible, so as to minimize redundancy between different bases. This can be imposed by minimizing $\sum_{i\neq j}(\mathbf{W}^T\mathbf{W})_{ij}$.

3. Only components giving most important information should be retained. Given that every image in $\mathbf{V}$ has been normalized into certain range, the total "activity" on each retained component, defined as the total squared projection coefficients summed over all training images, should be maximized. This can be imposed by maximizing $\sum_i(\mathbf{H}\mathbf{H}^T)_{ii}$.

Using constrained divergence, LNMF has the following objective function

$$D(\mathbf{V}||\mathbf{WH}) = \sum_{i,j} \left(\mathbf{v}_{ij} \log \frac{\mathbf{V}_{ij}}{(\mathbf{WH})_{ij}} - \mathbf{v}_{ij} + (\mathbf{WH})_{ij} \right) \tag{5.21}$$

$$+ \alpha \sum_{ij} (\mathbf{W}^T\mathbf{W})_{ij} - \beta \sum_{i} (\mathbf{HH}^T)_{ii} \tag{5.22}$$

where $\alpha, \beta > 0$ are some constrains. The update rules are

$$\mathbf{W}_{ia} \leftarrow \frac{\mathbf{W}_{ia} \sum_\mu \mathbf{V}_{i\mu} \frac{\mathbf{H}_{a\mu}}{\sum_j \mathbf{W}_{ja}\mathbf{H}_{a\mu}}}{\sum_j \mathbf{H}_{\mu j}}, \mathbf{W}_{ia} \leftarrow \frac{\mathbf{W}_{ia}}{\sum_j \mathbf{W}_{ja}} \tag{5.23}$$

$$\mathbf{H}_{a\mu} \leftarrow \sqrt{\mathbf{H}_{a\mu} \sum_i \mathbf{W}_{ia} \frac{\mathbf{V}_{i\mu}}{\sum_j \mathbf{W}_{ij}\mathbf{H}_{j\mu}}}. \tag{5.24}$$

We note that the parameters α, β have disappeared in the learning rules. In the original analysis of the algorithm, a middle parameter γ, which is a function of β but without clear expression, is used. The ratio $\frac{\alpha}{\gamma}$ is set to be not too large. This means that the learning rules should be sensitive to the the parameters α, β. But the values of the parameters α, β are hard to choose since the middle parameter γ is unclear.

Xu et al. [163] proposed a method called Constrained NMF, which is a very similar algorithm as LNMF. Xu et al. [163] added three additional constraints on the matrices $\mathbf{W}$ and $\mathbf{H}$, which is aimed to obtain the semantic relations as orthogonal as possible. Constrained NMF has the following objective function:

$$D(\mathbf{V}||\mathbf{WH}) = \sum_{i,j} \left(\mathbf{v}_{ij} \log \frac{\mathbf{V}_{ij}}{(\mathbf{WH})_{ij}} - \mathbf{v}_{ij} + (\mathbf{WH})_{ij} \right) \tag{5.25}$$

$$+ \alpha \sum_{i \neq j} (\mathbf{W}^T\mathbf{W})_{ij} - \beta \sum_{i} (\mathbf{W}^T\mathbf{W})_{ii} + \gamma \sum_{i} (\mathbf{HH}^T)_{ii} \tag{5.26}$$

where α, β and γ are positive constants. The update rules are

$$\mathbf{W}_{ia} \leftarrow \frac{\mathbf{W}_{ia} \sum_\mu \mathbf{V}_{i\mu} \frac{\mathbf{H}_{a\mu}}{\sum_j \mathbf{W}_{ij}\mathbf{H}_{j\mu}}}{\sum_j \mathbf{H}_{aj}} \frac{1}{\sum_j \mathbf{H}_{aj} + 2\alpha \sum_{j \neq a} \mathbf{W}_{ij} - 2\beta} \tag{5.27}$$

$$\mathbf{W}_{ia} \leftarrow \frac{\mathbf{W}_{ia}}{\sum_j \mathbf{W}_{ja}} \tag{5.28}$$

$$\mathbf{H}_{a\mu} \leftarrow \frac{1}{4\gamma} \left(\sqrt{8\gamma \mathbf{H}_{a\mu} \sum_i \mathbf{W}_{ia} \frac{\mathbf{V}_{i\mu}}{\sum_j \mathbf{W}_{ij}\mathbf{H}_{j\mu}} + 1} - 1 \right). \tag{5.29}$$

5.4.3 Constrained NMF

In general, we can state the constrained NMF optimization problem as

$$\min_{\mathbf{W},\mathbf{H}\geq 0}\{||\mathbf{V}-\mathbf{WH}||_F^2+\alpha J_1(\mathbf{W})+\beta J_2(\mathbf{H})\}. \tag{5.30}$$

The functions $J_1(\mathbf{W})$ and $J_2(\mathbf{H})$ are penalty terms used to enforce certain constraints on the solution of Equation (5.30), and α and β are their corresponding Lagrangian multipliers, or regularization parameters. Different penalty terms may be used depending upon the desired effects on the computed solution. The above NNSC, NMFSC, LNMF and constraint NMF algorithms can be considered as special cases for solving equation (5.30) by choosing different penalty terms.

Pauca et al [121] presented an NMF algorithm with incorporating additional constraints (CNMF) by setting

$$J_1(\mathbf{W})=||\mathbf{W}||^2, J_2(\mathbf{H})=||\mathbf{H}||^2. \tag{5.31}$$

The update rules become

$$\mathbf{W}_{ij}\leftarrow\mathbf{W}_{ij}\frac{(\mathbf{VH}^T)_{ij}-\alpha\mathbf{W}_{ij}}{(\mathbf{WHH}^T)_{ij}}, \mathbf{W}_{ij}\leftarrow\frac{\mathbf{W}_{ij}}{\sum_k\mathbf{W}_{kj}} \tag{5.32}$$

$$\mathbf{H}_{ij}\leftarrow\mathbf{H}_{ij}\frac{(\mathbf{W}^T\mathbf{V})_{ij}-\beta\mathbf{H}_{ij}}{(\mathbf{W}^T\mathbf{WH})_{ij}} \tag{5.33}$$

5.4.4 NonSmooth NMF

NonSmooth Nonnegative Matrix Factorization (nsNMF) [120], is defined as:

$$\mathbf{V}\approx\mathbf{WSH} \tag{5.34}$$

where the “smoothing” matrix $\mathbf{S}\in\mathbf{R}^{q\times q}$ is a positive symmetric matrix defined as:

$$\mathbf{S}=(1-\theta)\mathbf{I}+\frac{\theta}{q}\mathbf{1}\mathbf{1}^T, \tag{5.35}$$

where $\mathbf{I}$ is the identity matrix, $\mathbf{1}$ is a vector of ones, and the parameter θ satisfies $0\leq\theta\leq 1$.

The nsNMF algorithm is very straightforward to derive by simply substituting the nsNMF model into the divergence functional in (4) and following the same procedure to minimize the functional as performed.

The interpretation of $\mathbf{S}$ as a smoothing matrix can be explained as follows: Let $\mathbf{x}$ be a positive vector. Consider the transformed vector $\mathbf{y} = \mathbf{S}\mathbf{x}$. If $\theta = 0$, then $\mathbf{y} = \mathbf{x}$ and no smoothing on $\mathbf{x}$ has occurred. However, as $\theta \to 1$, the vector $\mathbf{y}$ tends to the constant vector with all elements almost equal to the average of the elements of $\mathbf{x}$. This is the smoothest possible vector in the sense of "nonsparseness" because all entries are equal to the same nonzero value, instead of having some values close to zero and others clearly nonzero. Note that the parameter θ controls the extent of smoothness of the matrix operator $\mathbf{S}$. However, due to the multiplicative nature of the model, strong smoothing in $\mathbf{S}$ will force strong sparseness in both the basis and the encoding vectors in order to maintain faithfulness of the model to the data. Therefore, the parameter θ controls the sparseness of the model. Note that, when $\theta = 0$, the model corresponds to the basic NMF.

5.4.5 A class of NMF algorithms using different measures

In Cichocki [28], the authors discuss a wide class of cost functions for NMF using Csiszár's φ-divergences defined as

$$D_C(z||y) = \sum_{k=1}^{N} z_k \varphi(\frac{y_k}{z_k}), \tag{5.36}$$

where $y_k \geq 0$, $z_k \geq 0$ and $\varphi : [0, \infty) \to (-\infty, \infty)$ is a function which is convex on $(0, \infty)$ and continuous at zero. To use the Csiszar's φ-divergences as a distance measure, we assume that $\varphi(1) = 0$ and that it is strictly convex at 1.

Choosing different function φ gives us many different distance measurements, for example:

1. Hellinger distance: If $\varphi(u) = (\sqrt{u} - 1)^2$, then $D_{C-H} = \sum_{ik}(\sqrt{y_{ik}} - \sqrt{z_{ik}})^2$.

2. Pearson's distance: If $\varphi(u) = (u - 1)^2$, then $D_{C-P} = \sum_{ik}(y_{ik} - z_{ik})^2/z_{ik}$.

3. Amari's alpha divergences: If $\varphi(u) = u(u^{\beta-1} - 1)/(\beta^2 - \beta) + (1 - u)/\beta$, then

$$D_A^{(\beta)}(Z||Y) = \sum_{ik} y_{ik} \frac{(y_{ik}/z_{ik})^{\beta-1} - 1}{\beta(\beta - 1)} + \frac{z_{ik} - y_{ik}}{\beta}. \tag{5.37}$$

For $\beta \to 1$ we get the generalized Kullback-Leibler divergence and $\beta \to 1$ the generalized dual Kullback-Leibler divergence. Table 1 [28] lists the learning algorithms based on different cost functions derived from Csiszár's φ-divergences.

A series of work on NMF type algorithms have been done by Cichocki and his team [29, 30, 31, 32, 33, 34, 35, 36, 169].

5.5 Convergence issues and optimality

5.5.1 First order optimality condition

The problem of minimizing the function $f(\mathbf{W},\mathbf{H})$ defined in (5.2) with nonnegative constraints $\mathbf{W} \geq 0$ and $\mathbf{H} \geq 0$ is equivalent to minimizing the function

$$g(\mathbf{E},\mathbf{F}) = \frac{1}{2}||\mathbf{V} - (\mathbf{E}\cdot\mathbf{E})(\mathbf{F}\cdot\mathbf{F})||_F^2 \tag{5.38}$$

for any $m \times r$ matrix $\mathbf{E}$ and $r \times n$ matrix $\mathbf{F}$, where $\mathbf{E}\cdot\mathbf{E}$ denotes the Hadamard product of two matrices. Taking the partial derivative of g with respect to $\mathbf{E}$ and $\mathbf{F}$ respectively, one can get the first order optimality condition, which is also regarded as the Kuhn-Tucker condition [18, 26].

Theorem 2 *If $(\mathbf{E},\mathbf{F})$ is a local minimizer of the objective function g defined in (5.38), then we have*

$$\mathbf{E}\cdot((\mathbf{V} - (\mathbf{E}\cdot\mathbf{E})(\mathbf{F}\cdot\mathbf{F}))(\mathbf{F}\cdot\mathbf{F})^T) = 0 \tag{5.39}$$

$$\mathbf{F}\cdot((\mathbf{E}\cdot\mathbf{E})^T(\mathbf{V} - (\mathbf{E}\cdot\mathbf{E})(\mathbf{F}\cdot\mathbf{F}))) = 0. \tag{5.40}$$

Consequently, we have the Kuhn-Tucker condition for NMF optimal problem,

Corollary 1 *The necessary condition for $(\mathbf{W},\mathbf{H})$ to solve the NMF optimal problem is*

$$\mathbf{W}\cdot((\mathbf{V} - \mathbf{W}\mathbf{H})\mathbf{H}^T) = 0 \tag{5.41}$$

$$\mathbf{H}\cdot(\mathbf{W}^T(\mathbf{V} - \mathbf{W}\mathbf{H})) = 0. \tag{5.42}$$

and furthermore, we have the following inequalities

$$(\mathbf{V} - \mathbf{W}\mathbf{H})\mathbf{H}^T \leq 0 \tag{5.43}$$

$$\mathbf{W}^T(\mathbf{V} - \mathbf{W}\mathbf{H}) \leq 0. \tag{5.44}$$

5.5.2 Convergence issues

In Lee and Seung's NMF algorithms [91, 92], they claimed that their algorithms converge to a local minimum. However, a closer inspection of these proofs revealed that, while these two algorithms do guarantee continual descent, this does not rule out the possibility of descending to a saddle point [48, 55, 96]. In fact, numerical experiments have shown that these two NMF algorithms may converge to a saddle point that does not satisfy the first-order optimality conditions of a local minimum [55]. Nevertheless, some experiments show that saddle point solutions can give reasonable results in the context of the problem, a finding confirmed by experiments with ALS- type algorithms in other contexts [53, 148, 161, 162]. It would be beneficial to determine what percentage of fixed points are local minima as opposed to saddle points for particular applications. It may be that in practice, while saddle points are possible, the algorithms often converge to local minima. In general, one should input the fixed point solution produced by an NMF algorithm into optimality conditions [26, 55] to determine if it is indeed a minimum. If the solution passes the optimality conditions, then it is at least a local minimum.

Most NMF algorithms conduct an alternating process, approximating **W**, then **H**, and so on. Mathematically, algorithms following this alternating process are actually variants of a simple optimization technique which is known under various names such as alternating variables, coordinate search, or the method of local variation [102]. No global convergence has been proven for this method in the most general cases. Powell [129, 130] and Zangwill [168] prove convergence for special classes of objective functions, such as convex quadratic functions. Furthermore, it is known that an ALS algorithm that properly enforces nonnegativity, for example, through the nonnegative least squares (NNLS) algorithm of [86], will converge to a local minimum [14, 96].

Quite many NMF algorithms use a fixed number of iterations as the convergence criterion. However, a fixed number of iterations is not a mathematically appealing way to control the number of iterations executed because the most appropriate value for maximum number of iterations is problem-dependent. The first paper to mention this convergence criterion problem is Lin [96], which includes experiments and comparisons. Another alternative is also suggested in [87].

5.6 Initialization and uniqueness

All NMF algorithms are iterative and they are sensitive to the initialization of **W** and **H** [157]. Some algorithms require that both **W** and **H** be initialized [63, 64, 91, 92, 121], while others require initialization of only **W** [116, 111, 140]. In all cases, a good initialization can improve the speed and accuracy of the algorithms, as it can produce faster convergence to an improved local minimum [148]. A good initialization can sidestep some of the convergence problems mentioned above, which is precisely why they are so important.

Most NMF algorithms use random matrices as the initializations for the factors $\mathbf{W}$ and $\mathbf{H}$. It is well-known that random initialization does not generally provide a good first estimate for NMF algorithms [148], especially those of the ALS-type of [137, 139]. Wild et al. [158, 157, 159] provided a method called the centroid initialization, built from the centroid decomposition [40]. The authors showed that it is a much better alternative to random initialization. Unfortunately, this decomposition is expensive as a preprocessing step for the NMF.

In [87], the authors propose four initialization methods,

1. SVD-centroid initialization. It initializes $\mathbf{W}$ with a centroid decomposition of the low dimensional SVD factor [145]. While the centroid decomposition of $\mathbf{V}$ can be too time-consuming, the centroid decomposition of the SVD factor is fast because it is much smaller than $\mathbf{V}$.

2. Random Acol initialization. It forms an initialization of each column of the basis matrix $\mathbf{W}$ by averaging r random columns of $\mathbf{V}$. It makes more sense to build basis vectors from the given data, the sparse document vectors themselves, than to form completely dense random basis vectors, as random initialization does.

3. Random $\mathbf{C}$ initialization. This method is similar to the random Acol method, except it chooses p columns at random from the longest (in the 2-norm) columns of $\mathbf{V}$, which generally means the densest columns since often the data matrices are sparse. The idea is that these might be more likely to be the centroid centers.

4. Co-occurrence matrix. It first forms a matrix $\mathbf{C} = \mathbf{V}\mathbf{V}^T$ which the autnors [138] called a co-occurrence matrix. Next, the method for forming the columns of initial $\mathbf{W}$ described as Algorithm 2 of [138] is applied to $\mathbf{C}$. The co-occurrence method is very expensive.

It is easy to see that the optimal solution for NMF problem is not unique. In fact, the NMF problem does not have a unique global minimum. Consider that a minimum solution given by the matrices $\mathbf{W}$ and $\mathbf{H}$ can also be given by an infinite number of equally good solution pairs such as $\mathbf{WD}$ and $\mathbf{D}^{-1}\mathbf{H}$ for any nonnegative $\mathbf{D}$ and $\mathbf{D}^{-1}$. Since scaling and permutation cause uniqueness problems, some algorithms enforce row or column normalizations at each iteration to alleviate these. Sufficient conditions for uniqueness of solutions to the NMF problem can be considered in terms of simplicial cones [10], and have been studied in [41].

5.7 Some applications

5.7.1 Environmetrics and chemometrics

The application of NMF in environmetrics and chemometrics comes from a series of work done by Paatero et al. [3, 111, 112, 113, 115, 116]. They applied the ideas of PMF to environmental data as early as 1991 [117]. Later Paatero [113] applied his multi-linear engine to analyze atmospheric emission and pollution data. A paper discussing the application of orthogonal projection approach, alternating least squares and PMF to analyze chromatographic spectral data (which is used to analyze mixtures of chemicals) was presented by Frenich et al. [49]. The results obtained by these three methods are compared by evaluating measures of dissimilarity between real and estimated spectra. Qin et al. [131] used PMF on a large aerosol database incorporating error estimates through the $\mathbf{W}$ matrix. Paatero et al. [114] discuss the resolution of the problem of rotational indeterminacy in the PMF (and extended algorithms) solutions using a specific two factor model as an example. Ramadan et al. [132] applied PMF to pollutant concentrations. Sajda et al. [135] applied their constrained version of NMF to recovering constituent spectra in 3D chemical shift imaging.

5.7.2 Image processing

Starting from Lee and Seung [91, 92], many NMF algorithms have been developed for applications in image processing. Since the image data are represented as nonnegative matrix arrays, it is desirable to process data sets of images represented by column vectors as composite objects, or as separated parts. It is suggested that the factorization in the linear model would enable the identification and classification of intrinsic 'parts' that make up the object being imaged by multiple observations [41, 91], see Figure 5.2. More specifically, since $||\mathbf{V} - \mathbf{W}\mathbf{H}||^2 = \sum_{j=1}^{n} ||\mathbf{v}_j - \mathbf{W}\mathbf{h}_j||^2 = \sum_{j=1}^{n} ||\mathbf{v}_j - \sum_{k=1}^{r} \mathbf{w}_k h_{kj}||^2$, each column $\mathbf{v}_j$ of a nonnegative matrix $\mathbf{V}$ now represents m pixel values of one image. The columns $\mathbf{w}_k$ of $\mathbf{W}$ are basis elements in R^m. The columns of $\mathbf{H}$, belonging to R^r, can be thought of as coefficient sequences representing the n images in the basis elements. In other words, the NMF model can be thought of as that there are standard parts $\mathbf{w}_k$ in a variety of positions and that each image represented as a vector $\mathbf{v}_j$ is made by superposing these parts together in specific ways by a mixing matrix represented by $\mathbf{H}$. Those parts, being images themselves, are necessarily nonnegative. The superposition coefficients, each part being present or absent, are also necessarily nonnegative.

Feng et al. [47] used their local NMF algorithm for learning a spatially localized, parts-based representation for images. They compare their method to PCA and NMF to demonstrate the situations where a spatially localized approach has advantages (such as highly occluded faces during face recognition). Guillamet and Vitrià [59] suggest using the Earth Movers Distance as a relevant metric for doing face recognition using NMF. Spratling [149] evaluates the empirical performance of some NMF algorithms for recognizing elementary image features, especially in the presence of occlusion.

Other work on face and image processing applications of NMF includes [56, 57, 58, 60, 155, 88, 89, 170, 157].

5.7.3 Text mining

Lee and Seung [91] also used their NMF methods to text documents and highlighted the ability of NMF to tackle semantic issues such as synonymy. Owing to the low-rank approximations produced NMF is a natural candidate for a clustering procedure. Xu et al. [166] described clustering experiments with NMF, wherein they compared NMF against spectral methods, suggesting that the former can obtain higher accuracy. Other related work on text analysis using NMF includes [6, 12, 40, 122, 140]. An application to email surveillance was discussed in [11].

5.7.4 Bioinformatics

Brunet et al. [17] apply NMF to form metagenes to infer biological information from cancer-related microarray data. They use the KL-Divergence based NMF algorithm and also provide heuristic methods for model selection. Kim and Tidor [79] apply NMF for performing dimensionality reduction to aid in the identification of subsystems from gene microarray data. They hinged their arguments on the ability to detect local features from the data using NMF. Other applications include lung cancer prognosis [76], analysis of lung cancer profiles [50], sparse NMF for cancer class discovery [51], gene data [7, 133], and Alzheimer's disease [25].

5.7.5 Miscellaneous applications

Nonnegative matrix factorization also has many other applications, such as computer vision [141], linear sparse coding [46, 61, 63, 152], neural learning process [93], sound recognition [78], remote sensing and object characterization [123, 159], polyphonic music [146, 147], object characterization [123], spectral data analysis [122], learning sound dictionaries [4], mining ratio-rules [65], and multiway clustering [6, 143].

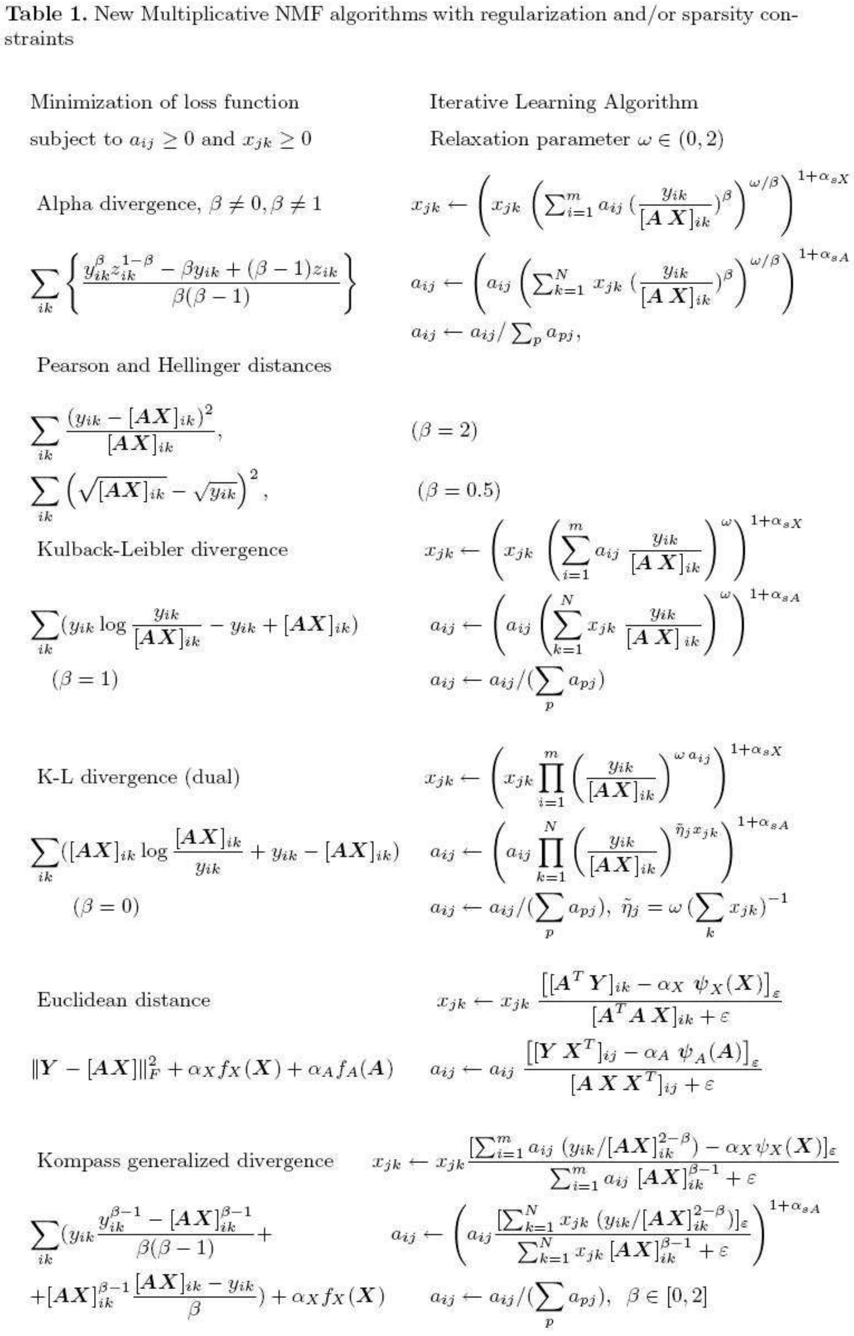


Table 1. New Multiplicative NMF algorithms with regularization and/or sparsity constraints

Minimization of loss function subject to $a_{ij} \geq 0$ and $x_{jk} \geq 0$	Iterative Learning Algorithm Relaxation parameter $\omega \in (0, 2)$
Alpha divergence, $\beta \neq 0, \beta \neq 1$ $\sum_{ik} \left\{ \frac{y_{ik}^{\beta} z_{ik}^{1-\beta} - \beta y_{ik} + (\beta - 1) z_{ik}}{\beta(\beta - 1)} \right\}$	$x_{jk} \leftarrow \left(x_{jk} \left(\sum_{i=1}^{m} a_{ij} \, (\frac{y_{ik}}{[\boldsymbol{A}\,\boldsymbol{X}]_{ik}})^{\beta} \right)^{\omega/\beta} \right)^{1+\alpha_{sX}}$ $a_{ij} \leftarrow \left(a_{ij} \left(\sum_{k=1}^{N} x_{jk} \, (\frac{y_{ik}}{[\boldsymbol{A}\,\boldsymbol{X}]_{ik}})^{\beta} \right)^{\omega/\beta} \right)^{1+\alpha_{sA}}$ $a_{ij} \leftarrow a_{ij} / \sum_{p} a_{pj}$,
Pearson and Hellinger distances $\sum_{ik} \frac{(y_{ik} - [\boldsymbol{AX}]_{ik})^2}{[\boldsymbol{AX}]_{ik}}$, $\sum_{ik} \left(\sqrt{[\boldsymbol{AX}]_{ik}} - \sqrt{y_{ik}} \right)^2$,	 $(\beta = 2)$ $(\beta = 0.5)$
Kulback-Leibler divergence $\sum_{ik} (y_{ik} \log \frac{y_{ik}}{[\boldsymbol{AX}]_{ik}} - y_{ik} + [\boldsymbol{AX}]_{ik})$ $(\beta = 1)$	$x_{jk} \leftarrow \left(x_{jk} \left(\sum_{i=1}^{m} a_{ij} \frac{y_{ik}}{[\boldsymbol{A}\,\boldsymbol{X}]_{ik}} \right)^{\omega} \right)^{1+\alpha_{sX}}$ $a_{ij} \leftarrow \left(a_{ij} \left(\sum_{k=1}^{N} x_{jk} \frac{y_{ik}}{[\boldsymbol{A}\,\boldsymbol{X}]_{ik}} \right)^{\omega} \right)^{1+\alpha_{sA}}$ $a_{ij} \leftarrow a_{ij} / (\sum_{p} a_{pj})$
K-L divergence (dual) $\sum_{ik} ([\boldsymbol{AX}]_{ik} \log \frac{[\boldsymbol{AX}]_{ik}}{y_{ik}} + y_{ik} - [\boldsymbol{AX}]_{ik})$ $(\beta = 0)$	$x_{jk} \leftarrow \left(x_{jk} \prod_{i=1}^{m} \left(\frac{y_{ik}}{[\boldsymbol{AX}]_{ik}} \right)^{\omega\, a_{ij}} \right)^{1+\alpha_{sX}}$ $a_{ij} \leftarrow \left(a_{ij} \prod_{k=1}^{N} \left(\frac{y_{ik}}{[\boldsymbol{AX}]_{ik}} \right)^{\tilde{\eta}_j x_{jk}} \right)^{1+\alpha_{sA}}$ $a_{ij} \leftarrow a_{ij} / (\sum_{p} a_{pj}),\ \tilde{\eta}_j = \omega \, (\sum_{k} x_{jk})^{-1}$
Euclidean distance $\lVert \boldsymbol{Y} - [\boldsymbol{AX}] \rVert_F^2 + \alpha_X f_X(\boldsymbol{X}) + \alpha_A f_A(\boldsymbol{A})$	$x_{jk} \leftarrow x_{jk} \frac{\left[[\boldsymbol{A}^T \boldsymbol{Y}]_{ik} - \alpha_X \ \psi_X(\boldsymbol{X}) \right]_{\varepsilon}}{[\boldsymbol{A}^T \boldsymbol{A}\,\boldsymbol{X}]_{ik} + \varepsilon}$ $a_{ij} \leftarrow a_{ij} \frac{\left[[\boldsymbol{Y}\,\boldsymbol{X}^T]_{ij} - \alpha_A \ \psi_A(\boldsymbol{A}) \right]_{\varepsilon}}{[\boldsymbol{A}\,\boldsymbol{X}\,\boldsymbol{X}^T]_{ij} + \varepsilon}$
Kompass generalized divergence $\sum_{ik} (y_{ik} \frac{y_{ik}^{\beta-1} - [\boldsymbol{AX}]_{ik}^{\beta-1}}{\beta(\beta - 1)} +$ $+ [\boldsymbol{AX}]_{ik}^{\beta-1} \frac{[\boldsymbol{AX}]_{ik} - y_{ik}}{\beta}) + \alpha_X f_X(\boldsymbol{X})$	$x_{jk} \leftarrow x_{jk} \frac{[\sum_{i=1}^{m} a_{ij} \ (y_{ik} / [\boldsymbol{AX}]_{ik}^{2-\beta}) - \alpha_X \psi_X(\boldsymbol{X})]_{\varepsilon}}{\sum_{i=1}^{m} a_{ij} \ [\boldsymbol{AX}]_{ik}^{\beta-1} + \varepsilon}$ $a_{ij} \leftarrow \left(a_{ij} \frac{[\sum_{k=1}^{N} x_{jk} \ (y_{ik} / [\boldsymbol{AX}]_{ik}^{2-\beta})]_{\varepsilon}}{\sum_{k=1}^{N} x_{jk} \, [\boldsymbol{AX}]_{ik}^{\beta-1} + \varepsilon} \right)^{1+\alpha_{sA}}$ $a_{ij} \leftarrow a_{ij} / (\sum_{p} a_{pj}),\ \ \beta \in [0, 2]$

Figure 5.1: This table is taken from [28]

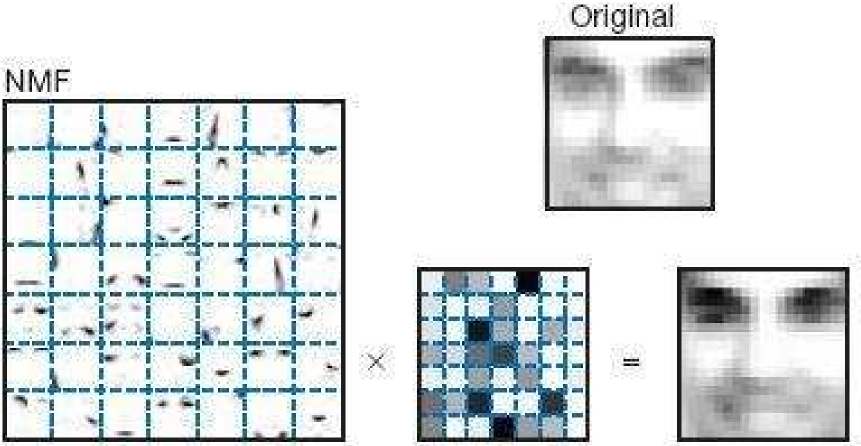


Figure 5.2: Non-negative matrix factorization (NMF) with a database of $m = 2429$ facial images, each consisting of $n = 19 \times 19$ pixels, and constituting an $n \times m$ matrix $\mathbf{V}$. NMF learns to represent faces with a set of basis images resembling parts of faces [91].

Chapter 6

The Projective NMF method

The Projective nonnegative matrix factorization (P-NMF) method was first developed in **Publication IV**. It simplifies the parameters of the standard nonnegative matrix factorization. **Publication V, VI, VII** analyse and extend this method further. In this chapter, a review based on these publications will be given.

6.1 Definition of the problem

The idea of the projective NMF comes from the compressive SVD method. The compressive SVD is a projection method. It projects the nonnegative $m \times n$ data matrix $\mathbf{V}$ onto the subspace of the first r eigenvectors of the data covariance matrix:

$$\mathbf{X} = \hat{\mathbf{Q}}\hat{\mathbf{Q}}^T\mathbf{V}.$$

where matrix $\hat{\mathbf{Q}}\hat{\mathbf{Q}}^T$ is the projection matrix on the eigenvector subspace. Generally, matrix $\hat{\mathbf{Q}}$ is not nonnegative. To keep the property of the nonnegativity, we try to find a nonnegative $m \times m$ approximative projection matrix $\mathbf{P}$ with given rank r, which minimizes the difference $||\mathbf{V} - \mathbf{PV}||$. Any symmetrical projection matrix of rank r can be written in the form

$$\mathbf{P} = \mathbf{W}\mathbf{W}^T$$

with $\mathbf{W}$ an $m \times r$ matrix, and its columns are orthogonal. Thus, we can try to solve the problem by searching for a nonnegative $m \times r$ matrix $\mathbf{W}$.

Based on this, Projective Non-negative Matrix Factorization (P-NMF) was introduced in **Publication IV**, which finds the solution to the following optimality problem

$$\min_{\mathbf{W} \geq 0} ||\mathbf{V} - \mathbf{W}\mathbf{W}^T\mathbf{V}||, \tag{6.1}$$

where $|| \cdot ||$ is a matrix norm. Note that with the positivity constraint, the orthogonality of $\mathbf{W}$ is not ensured any more, and the method is projective only approximately.

Lee and Seung's NMF algorithms [92] are parts based, localized algorithms. The base images in Figure 5.2 reveal this important characteristic. However, the overlap between the neighbourhood base images are still a little bit heavy. Therefore, some application-dependent auxiliary constraints, such as sparsity, have been added to produce algorithms to gain more localized and sparse information. Actually, although the orthogonality of the base matrix $\mathbf{W}$ is not ensured, more orthogonal base matrix $\mathbf{W}$ means less overlap between the neighbourhood base images, and more localized and sparse information. Instead of adding a penalty term in the objective function, the projective method uses projection matrix with nonnegative restriction, which leads to only one parameter matrix, and the orthogonality seems to be approached approximately.

6.2 Projective NMF algorithms

The Projective Non-negative Matrix Factorization (P-NMF) uses only one parameter matrix $\mathbf{W}$ instead of $\mathbf{W}$ and $\mathbf{H}$ in Non-negative Matrix Factorization. The weight matrix $\mathbf{H}$ in NMF is simply replaced by $\mathbf{W}^T\mathbf{V}$ in P-NMF algorithms. The update rules could be obtained similar to Lee and Seung's algorithms [92]. First, we give two auxiliary results:

Lemma 1 *For the given matrices* $\mathbf{W}$ *and* $\mathbf{V}$*, the minimization of the Frobenius norm* $f(\lambda) = ||\mathbf{V} - \lambda\mathbf{W}\mathbf{W}^T\mathbf{V}||^2$ *corresponding to* λ *is reached at*

$$\lambda = \frac{tr\mathbf{W}\mathbf{W}^T\mathbf{V}\mathbf{V}^T}{tr\mathbf{W}\mathbf{W}^T\mathbf{V}\mathbf{V}^T\mathbf{W}\mathbf{W}^T}. \tag{6.2}$$

Lemma 2 *For the given matrices* $\mathbf{W}$ *and* $\mathbf{V}$*, the minimization of the Kullback-*

Leibler divergence $f(\lambda) = \mathbf{D}(\mathbf{V}||\lambda\mathbf{W}\mathbf{W}^T\mathbf{V})$ *corresponding to* λ *is reached at*

$$\lambda = \frac{\sum_{ij}\mathbf{V}_{ij}}{\sum_{ij}(\mathbf{W}\mathbf{W}^T\mathbf{V})_{ij}}. \tag{6.3}$$

The proofs of these two lemmas are straightforward: take the gradient of $f(\lambda)$ to λ, and find the value of λ by setting $\frac{\partial f(\lambda)}{\partial\lambda} = 0$.

By the same technique used in Lee and Seung's algorithms [92], the update rules for Euclidean distance is:

$$\mathbf{W}_{ij} \leftarrow \mathbf{W}_{ij}\frac{2(\mathbf{V}\mathbf{V}^T\mathbf{W})_{ij}}{(\mathbf{W}\mathbf{W}^T\mathbf{V}\mathbf{V}^T\mathbf{W})_{ij} + (\mathbf{V}\mathbf{V}^T\mathbf{W}\mathbf{W}^T\mathbf{W})_{ij}} \tag{6.4}$$

The 'normalization' step followed according to Lemma 1 is

$$\mathbf{W} \leftarrow \mathbf{W}\sqrt{tr\mathbf{W}\mathbf{W}^T\mathbf{V}\mathbf{V}^T/tr\mathbf{W}\mathbf{W}^T\mathbf{V}\mathbf{V}^T\mathbf{W}\mathbf{W}^T}, \tag{6.5}$$

and for the Kullback-Leibler divergence measure, the update rule becomes

$$\mathbf{W}_{ij} \leftarrow \mathbf{W}_{ij}\times \tag{6.6}$$

$$\frac{\sum_k \mathbf{V}_{ik}\left((\mathbf{W}^T\mathbf{V})_{jk}/(\mathbf{W}\mathbf{W}^T\mathbf{V})_{ik} + \sum_l \mathbf{W}_{lj}\mathbf{V}_{lk}/(\mathbf{W}\mathbf{W}^T\mathbf{V})_{lk}\right)}{\sum_k\left((\mathbf{W}^T\mathbf{V})_{jk} + \sum_l \mathbf{W}_{lj}\mathbf{V}_{ik}\right)} \tag{6.7}$$

with the 'normalization' step according to Lemma 1

$$\mathbf{W} \leftarrow \mathbf{W}\sqrt{\frac{\sum_{ij}\mathbf{V}_{ij}}{\sum_{ij}(\mathbf{W}\mathbf{W}^T\mathbf{V})_{ij}}}. \tag{6.8}$$

The nonincreasing of the two algorithms is complicated, and has just been given by Yang and Oja [167]. Below, we will prove that in the case that the basis matrix $\mathbf{W}$ is a vector, both update rules (6.4 with the normalization 6.5) and (6.6 with normalization 6.8) make the objective function value nonincreasing.

First, let us prove this for the Euclidean error norm. Use the notations $\hat{\mathbf{W}}$ and $\overline{\mathbf{W}}$ be the updated vectors of vector $\mathbf{W}$ under equations (6.4) and (6.5), respectively. Let $\Delta\mathbf{W} := \hat{\mathbf{W}} - \mathbf{W}$. Set

$$\begin{aligned}
\mathbf{G}(\mathbf{W}): &= ||\mathbf{V} - \mathbf{W}\mathbf{W}^T\mathbf{V}||^2 - ||\mathbf{V} - \overline{\mathbf{W}\mathbf{W}}^T\mathbf{V}||^2 \\
&= Tr\left(\mathbf{V}\mathbf{V}^T - 2\mathbf{W}\mathbf{W}^T\mathbf{V}\mathbf{V}^T + \mathbf{W}\mathbf{W}^T\mathbf{V}\mathbf{V}^T\mathbf{W}\mathbf{W}^T\right) \\
&\quad -Tr\left(\mathbf{V}\mathbf{V}^T - 2\overline{\mathbf{W}\mathbf{W}}^T\mathbf{V}\mathbf{V}^T + \overline{\mathbf{W}\mathbf{W}}^T\mathbf{V}\mathbf{V}^T\overline{\mathbf{W}\mathbf{W}}^T\right) \\
&= Tr\left(\overline{\mathbf{W}\mathbf{W}}^T\mathbf{V}\mathbf{V}^T - \mathbf{W}\mathbf{W}^T\mathbf{V}\mathbf{V}^T\right)
\end{aligned}$$

The above equations use the following equations,

$$\begin{aligned}
Tr(\mathbf{W}\mathbf{W}^T\mathbf{V}\mathbf{V}^T\mathbf{W}\mathbf{W}^T) &= \mathbf{W}^T\mathbf{V}\mathbf{V}^T\mathbf{W}\cdot Tr(\mathbf{W}\mathbf{W}^T) \\
&= Tr(\mathbf{W}\mathbf{W}^T\mathbf{V}\mathbf{V}^T)\mathbf{W}^T\mathbf{W} \\
&= Tr(\mathbf{W}\mathbf{W}^T\mathbf{V}\mathbf{V}^T),
\end{aligned}$$

note that $||\overline{\mathbf{W}\mathbf{W}}^T||_F = ||\hat{\mathbf{W}}\hat{\mathbf{W}}^T||_F = 1$. Similar results for matrix $\overline{\mathbf{W}}$ hold. With a simple calculation,

$$\begin{aligned}
\overline{\mathbf{W}\mathbf{W}}^T &= \frac{\hat{\mathbf{W}}\hat{\mathbf{W}}^T}{\left(||\hat{\mathbf{W}}||\right)^2} \\
&= \frac{\hat{\mathbf{W}}\hat{\mathbf{W}}^T}{\hat{\mathbf{W}}^T\hat{\mathbf{W}}}
\end{aligned}$$

we have

$$\begin{aligned}
\mathbf{G}(\mathbf{W}) &= \frac{Tr(\hat{\mathbf{W}}\hat{\mathbf{W}}^T\mathbf{V}\mathbf{V}^T)}{\hat{\mathbf{W}}^T\hat{\mathbf{W}}} - Tr(\mathbf{W}\mathbf{W}^T\mathbf{V}\mathbf{V}^T) \\
&= \frac{1}{\hat{\mathbf{W}}^T\hat{\mathbf{W}}}\left(\hat{\mathbf{W}}^T\mathbf{V}\mathbf{V}^T\hat{\mathbf{W}} - \mathbf{W}^T\mathbf{V}\mathbf{V}^T\mathbf{W}\hat{\mathbf{W}}^T\hat{\mathbf{W}}\right)
\end{aligned}$$

The update rule equation (6.4) becomes

$$\hat{\mathbf{W}}_i = \mathbf{W}_i\frac{2(\mathbf{V}\mathbf{V}^T\mathbf{W})_i}{\mathbf{W}_i\mathbf{W}^T\mathbf{V}\mathbf{V}^T\mathbf{W} + (\mathbf{V}\mathbf{V}^T\mathbf{W})_i}. \tag{6.9}$$

Since

$$\begin{aligned}
\hat{\mathbf{W}}^T\mathbf{V}\mathbf{V}^T\hat{\mathbf{W}} &= (\mathbf{W}+\Delta\mathbf{W})^T\mathbf{V}\mathbf{V}^T(\mathbf{W}+\Delta\mathbf{W}) \\
&= \mathbf{W}^T\mathbf{V}\mathbf{V}^T\mathbf{W} + 2\mathbf{W}^T\mathbf{V}\mathbf{V}^T\Delta\mathbf{W} + \Delta\mathbf{W}^T\mathbf{V}\mathbf{V}^T\Delta\mathbf{W}
\end{aligned}$$

and

$$\begin{aligned}
\hat{\mathbf{W}}^T\hat{\mathbf{W}} &= (\mathbf{W}+\Delta\mathbf{W})^T(\mathbf{W}+\Delta\mathbf{W}) \\
&= \mathbf{W}^T\mathbf{W} + 2\mathbf{W}^T\Delta\mathbf{W} + \Delta\mathbf{W}^T\Delta\mathbf{W} \\
&= 1 + 2\mathbf{W}^T\Delta\mathbf{W} + \Delta\mathbf{W}^T\Delta\mathbf{W}
\end{aligned}$$

Thus, using the fact $\Delta\mathbf{W}^T\mathbf{V}\mathbf{V}^T\Delta\mathbf{W} \geq 0$

$$\begin{aligned}
\mathbf{G}(\mathbf{W}) &= \frac{2\mathbf{W}^T\mathbf{V}\mathbf{V}^T\Delta\mathbf{W} + \Delta\mathbf{W}^T\mathbf{V}\mathbf{V}^T\Delta\mathbf{W} - 2\mathbf{W}^T\Delta\mathbf{W}\mathbf{W}^T\mathbf{V}\mathbf{V}^T\mathbf{W}}{\hat{\mathbf{W}}^T\hat{\mathbf{W}}} \\
&- \frac{\Delta\mathbf{W}^T\Delta\mathbf{W}\mathbf{W}^T\mathbf{V}\mathbf{V}^T\mathbf{W}}{\hat{\mathbf{W}}^T\hat{\mathbf{W}}} \\
&\geq \frac{\sum_i 2\Delta\mathbf{W}_i\left((\mathbf{V}\mathbf{V}^T\mathbf{W})_i - \mathbf{W}_i\mathbf{W}^T\mathbf{V}\mathbf{V}^T\mathbf{W}\right) - \sum_i \Delta\mathbf{W}_i^2\mathbf{W}^T\mathbf{V}\mathbf{V}^T\mathbf{W}}{\hat{\mathbf{W}}^T\hat{\mathbf{W}}} \\
&= \frac{1}{\hat{\mathbf{W}}^T\hat{\mathbf{W}}}\sum_i \Delta\mathbf{W}_i^2\left(\frac{2(\mathbf{V}\mathbf{V}^T\mathbf{W})_i + 2\mathbf{W}_i\mathbf{W}^T\mathbf{V}\mathbf{V}^T\mathbf{W}}{\mathbf{W}_i} - \mathbf{W}^T\mathbf{V}\mathbf{V}^T\mathbf{W}\right) \\
&\geq 0.
\end{aligned}$$

For the divergence case, we use the same technique as Lee's [91].

The update rule equation (6.6) becomes

$$\hat{\mathbf{W}}_i = \mathbf{W}_i\frac{\sum_k \mathbf{V}_{ik}\left(1/\mathbf{W}_i + \sum_l \mathbf{V}_{lk}/(\mathbf{W}^T\mathbf{V})_k\right)}{\sum_k\left((\mathbf{W}^T\mathbf{V})_k + \sum_l \mathbf{W}_l\mathbf{V}_{ik}\right)}. \tag{6.10}$$

Let

$$\mathbf{F}(\mathbf{W}) = \sum_{i,j}\left(\mathbf{V}_{ij}\log\frac{\mathbf{V}_{ij}}{(\mathbf{W}\mathbf{W}^T\mathbf{V})_{ij}} - \mathbf{V}_{ij} + (\mathbf{W}\mathbf{W}^T\mathbf{V})_{ij}\right) \tag{6.11}$$

and

$$\begin{aligned}
\mathbf{G}(\mathbf{W},\mathbf{W}^t) &= \sum_{i,j}(\mathbf{V}_{ij}\log\mathbf{V}_{ij} - \mathbf{V}_{ij}) + \sum_{i,j}(\mathbf{W}\mathbf{W}^T\mathbf{V})_{ij} \qquad (6.12) \\
&- \sum_{i,j,a}\mathbf{V}_{ij}\frac{\mathbf{W}_a^t\mathbf{V}_{aj}}{(\mathbf{W}^t)^T\mathbf{V}_j}\left(\log\mathbf{W}_i\mathbf{W}_a\mathbf{V}_{aj} - \log\frac{\mathbf{W}_a^t\mathbf{V}_{aj}}{(\mathbf{W}^t)^T\mathbf{V}_j}\right) \qquad (6.13)
\end{aligned}$$

We need to show that $\mathbf{G}$ is an auxiliary function of $\mathbf{F}$. It is easy to see that $\mathbf{G}(\mathbf{W},\mathbf{W}) = \mathbf{F}(\mathbf{W})$. To show $\mathbf{G}(\mathbf{W},\mathbf{W}^t) \geq \mathbf{F}(\mathbf{W})$, use the convexity of the log function:

$$-\log\sum_a \mathbf{W}_a\mathbf{V}_{aj} \leq -\sum_a \alpha_a\log\frac{\mathbf{W}_a\mathbf{V}_{aj}}{\alpha_a} \tag{6.14}$$

holds for all nonnegative α_a that sum to unity. Setting

$$\alpha_a = \frac{\mathbf{W}_a^t\mathbf{V}_{aj}}{(\mathbf{W}^t)^T\mathbf{V}_j}. \tag{6.15}$$

We have

$$-\log(\mathbf{W}\mathbf{W}^T\mathbf{V})_{ij} \leq -\sum_{i,j,a} \mathbf{V}_{ij} \frac{\mathbf{W}_a^t \mathbf{V}_{aj}}{(\mathbf{W}^t)^T \mathbf{V}_j} \left(\log \mathbf{W}_i \mathbf{W}_a \mathbf{V}_{aj} - \log \frac{\mathbf{W}_a^t \mathbf{V}_{aj}}{(\mathbf{W}^t)^T \mathbf{V}_j} \right), \tag{6.16}$$

which imply that $\mathbf{G}$ is an auxiliary function of $\mathbf{F}$.

Furthermore, the minimum of $\mathbf{G}(\mathbf{W}, \mathbf{W}^t)$ with respect to $\mathbf{W}$ is determined by setting the following gradient to zero

$$\frac{\partial \mathbf{G}(\mathbf{W}, \mathbf{W}^t)}{\partial \mathbf{W}_a} = \sum_j \left(\sum_i \mathbf{W}_i \mathbf{V}_{aj} + \sum_k \mathbf{W}_k \mathbf{V}_{kj} - \frac{\mathbf{V}_{aj} + \sum_i \mathbf{V}_{ij} \frac{\mathbf{W}_a^t \mathbf{V}_{aj}}{(\mathbf{W}^t)^T \mathbf{V}_j}}{\mathbf{W}_a} \right) = 0. \tag{6.17}$$

Solving the above equations, we have

$$\mathbf{W}_i^{t+1} = \mathbf{W}_i^t \frac{\sum_k \mathbf{V}_{ik} \left(1/\mathbf{W}_i^t + \sum_l \mathbf{V}_{lk} / ((\mathbf{W}^t)^T \mathbf{V})_k \right)}{\sum_k \left(((\mathbf{W}^t)^T \mathbf{V})_k + \sum_l \mathbf{W}_l^t \mathbf{V}_{ik} \right)}. \tag{6.18}$$

This proves that the algorithm in divergence when $\mathbf{W}$ is a vector causes the objective function value to be nonincreasing.

P-NMF was shown to work well, especially, it gives a more localized and sparse representation than general NMF algorithms, see **Publication IV, V, VII**. The question we address here is whether variants of the method can be developed using some other distance or divergence measures, and how would these variants compare with NMF in terms of localized, sparse representations and the orthogonality of the basis vectors.

6.3 P-NMF algorithms with different divergence measurements

The algorithms for minimizing these distances and preserving positivity follow the same idea as in NMF and P-NMF: starting from a gradient descent, we find a suitable step size such that the algorithms become multiplicative instead of additive. When everything is positive or non-negative initially, this property will be maintained by the multiplicative update rules and a non-negative solution is guaranteed after convergence.

1. For Amari's alpha divergence

$$\begin{aligned}\mathbf{D}_A^{(\beta)}(\mathbf{W}\mathbf{W}^T\mathbf{V}||\mathbf{V}) = &\sum_{ik}\mathbf{V}_{ik}\frac{(\mathbf{V}_{ik}/(\mathbf{W}\mathbf{W}^T\mathbf{V})_{ik})^{\beta-1}-1}{\beta(\beta-1)}\\ &+\sum_{ik}\frac{(\mathbf{W}\mathbf{W}^T\mathbf{V})_{ik}-\mathbf{V}_{ik}}{\beta}\end{aligned}$$

Computing the partial differential of $\mathbf{D}_A^{(\beta)}(\mathbf{W}\mathbf{W}^T\mathbf{V}||\mathbf{V})$ with respect to $\mathbf{W}$

$$\begin{aligned}\frac{\partial D_A^{(\beta)}(\mathbf{W}\mathbf{W}^T\mathbf{V}||\mathbf{V})}{\partial \mathbf{w}_{ij}} = &-\sum_{k,l}\frac{(\mathbf{V}_{kl}/(\mathbf{W}\mathbf{W}^T\mathbf{V})_{kl})^{\beta}}{\beta}\mathbf{W}_{kj}\mathbf{V}_{il}\\ &-\sum_{k,l}\frac{(\mathbf{V}_{il}/(\mathbf{W}\mathbf{W}^T\mathbf{V})_{il})^{\beta}}{\beta}\mathbf{W}_{kj}\mathbf{V}_{kl}\\ &+\frac{1}{\beta}\sum_{k,l}(\mathbf{W}_{kj}\mathbf{V}_{il}+\mathbf{W}_{kj}\mathbf{V}_{kl})\end{aligned}$$

we obtain the following algorithm:

$$\begin{aligned}\mathbf{W}_{ij} \leftarrow \mathbf{W}_{ij}&\frac{\sum_{k,l}(\mathbf{V}_{kl}/(\mathbf{W}\mathbf{W}^T\mathbf{V})_{kl})^{\beta}\mathbf{W}_{kj}\mathbf{V}_{il}}{\sum_{kl}\mathbf{W}_{kj}\mathbf{V}_{il}+\sum_{l}(\mathbf{W}^T\mathbf{V})_{jl}}\\ +\mathbf{W}_{ij}&\frac{\sum_{k,l}(\mathbf{V}_{il}/(\mathbf{W}\mathbf{W}^T\mathbf{V})_{il})^{\beta}\mathbf{W}_{kj}\mathbf{V}_{kl}}{\sum_{kl}\mathbf{W}_{kj}\mathbf{V}_{il}+\sum_{l}(\mathbf{W}^T\mathbf{V})_{jl}}\end{aligned}$$

2. For Hellinger distance

$$\mathbf{D}_H(\mathbf{W}\mathbf{W}^T\mathbf{V}||\mathbf{V}) = \sum_{ik}(\sqrt{(\mathbf{W}\mathbf{W}^T\mathbf{V})_{ik}}-\sqrt{(\mathbf{V})_{ik}})^2,$$

Applying gradient method

$$\begin{aligned}\frac{\partial D_H(\mathbf{W}\mathbf{W}^T\mathbf{V}||\mathbf{V})}{\partial \mathbf{w}_{ij}} = &\sum_{k,l}(\mathbf{W}_{kj}\mathbf{V}_{il}+\mathbf{W}_{kj}\mathbf{V}_{kl})\\ &-\sum_{k,l}\sqrt{\mathbf{V}_{kl}/(\mathbf{W}\mathbf{W}^T\mathbf{V})_{kl}}\mathbf{W}_{kj}\mathbf{V}_{il}\\ &-\sum_{k,l}\sqrt{\mathbf{V}_{il}/(\mathbf{W}\mathbf{W}^T\mathbf{V})_{il}}\mathbf{W}_{kj}\mathbf{V}_{kl}\end{aligned}$$

we get the update rule

$$\mathbf{W}_{ij} \leftarrow \mathbf{W}_{ij} \frac{\sum_{k,l} \sqrt{\mathbf{V}_{kl}/(\mathbf{W}\mathbf{W}^T\mathbf{V})_{kl}} \mathbf{W}_{kj}\mathbf{V}_{il}}{\sum_{kl} \mathbf{W}_{kj}\mathbf{V}_{il} + \sum_l (\mathbf{W}^T\mathbf{V})_{jl}} + \mathbf{W}_{ij} \frac{\sum_{k,l} \sqrt{\mathbf{V}_{il}/(\mathbf{W}\mathbf{W}^T\mathbf{V})_{il}} \mathbf{W}_{kj}\mathbf{V}_{kl}}{\sum_{kl} \mathbf{W}_{kj}\mathbf{V}_{il} + \sum_l (\mathbf{W}^T\mathbf{V})_{jl}}$$

3. For Pearson's distance

$$\mathbf{D}_P(\mathbf{W}\mathbf{W}^T\mathbf{V}||\mathbf{V}) = \sum_{ik} \frac{((\mathbf{W}\mathbf{W}^T\mathbf{V})_{ik} - \mathbf{V}_{ik})^2}{(\mathbf{W}\mathbf{W}^T\mathbf{V})_{ik}}$$

$$\begin{aligned}\frac{\partial D_P(\mathbf{W}\mathbf{W}^T\mathbf{V}||\mathbf{V})}{\partial \mathbf{w}_{ij}} &= \sum_{k,l} (\mathbf{W}_{kj}\mathbf{V}_{il} + \mathbf{W}_{kj}\mathbf{V}_{kl}) \\ &- \sum_{k,l} \frac{(\mathbf{V}_{kl})^2}{((\mathbf{W}\mathbf{W}^T\mathbf{V})_{kl})^2} \mathbf{W}_{kj}\mathbf{V}_{il} \\ &- \sum_{k,l} \frac{(\mathbf{V}_{il})^2}{((\mathbf{W}\mathbf{W}^T\mathbf{V})_{il})^2} \mathbf{W}_{kj}\mathbf{V}_{kl}.\end{aligned}$$

The update rule is

$$\mathbf{W}_{ij} \leftarrow \mathbf{W}_{ij} \frac{\sum_{k,l} \frac{(\mathbf{V}_{kl})^2}{((\mathbf{W}\mathbf{W}^T\mathbf{V})_{kl})^2} \mathbf{W}_{kj}\mathbf{V}_{il}}{\sum_{kl} \mathbf{W}_{kj}\mathbf{V}_{il} + \sum_l (\mathbf{W}^T\mathbf{V})_{jl}} + \mathbf{W}_{ij} \frac{\sum_{k,l} \frac{(\mathbf{V}_{il})^2}{((\mathbf{W}\mathbf{W}^T\mathbf{V})_{il})^2} \mathbf{W}_{kj}\mathbf{V}_{kl}}{\sum_{kl} \mathbf{W}_{kj}\mathbf{V}_{il} + \sum_l (\mathbf{W}^T\mathbf{V})_{jl}}$$

4. For Pearson's dual distance

$$\mathbf{D}_{dP}(\mathbf{W}\mathbf{W}^T\mathbf{V}||\mathbf{V}) = \sum_{ik} \frac{((\mathbf{W}\mathbf{W}^T\mathbf{V})_{ik} - \mathbf{V}_{ik})^2}{\mathbf{V}_{ik}}$$

$$\begin{aligned}\frac{\partial D_{dP}(\mathbf{W}\mathbf{W}^T\mathbf{V}||\mathbf{V})}{\partial \mathbf{w}_{ij}} &= \sum_{k,l} \frac{2(\mathbf{W}\mathbf{W}^T\mathbf{V})_{kl}}{\mathbf{V}_{kl}} \mathbf{W}_{kj}\mathbf{V}_{il} \\ &+ \sum_{k,l} \frac{2(\mathbf{W}\mathbf{W}^T\mathbf{V})_{il}}{\mathbf{V}_{il}} \mathbf{W}_{kj}\mathbf{V}_{kl} \\ &- \sum_{k,l} 2\,(\mathbf{W}_{kj}\mathbf{V}_{il} + \mathbf{W}_{kj}\mathbf{V}_{kl})\end{aligned}$$

The update rule is

$$\mathbf{W}_{ij} \leftarrow \mathbf{W}_{ij} \times \\ \frac{\sum_{kl} \mathbf{W}_{kj}\mathbf{V}_{il} + \sum_{l} (\mathbf{W}^T\mathbf{V})_{jl}}{\sum_{k,l} \frac{2(\mathbf{W}\mathbf{W}^T\mathbf{V})_{kl}}{\mathbf{V}_{kl}} \mathbf{W}_{kj}\mathbf{V}_{il} + \sum_{k,l} \frac{2(\mathbf{W}\mathbf{W}^T\mathbf{V})_{il}}{\mathbf{V}_{il}} \mathbf{W}_{kj}\mathbf{V}_{kl}}.$$

The optimal choice of the above algorithms depends on the distribution of data. If such knowledge is not available, we may run all these algorithms to find an optimal solution. In some tasks and distributions there are particular divergence measures that are uniquely suited. On the other hand, if the approximating model fits the true distribution well, then it does not matter which divergence measure is used, since all of them will give similar results, see Figure 6.2.

6.4 P-NMF with regularization

In the standard NMF, we can impose some additional constraints such as sparsity [63, 64], 'smoothing' [120] , etc. Kompass [84] generalized a divergence measure for nonnegative matrix factorization with adding regularization terms

$$\mathbf{D}_{Ko}(\mathbf{W}\mathbf{W}^T\mathbf{V}||\mathbf{V}) = \sum_{ik} \mathbf{V}_{ik} \frac{\mathbf{V}_{ik}^{\beta-1} - (\mathbf{W}\mathbf{W}^T\mathbf{V})_{ik}^{\beta-1}}{\beta(\beta-1)} \tag{6.19}$$

$$+ \sum_{ik} (\mathbf{W}\mathbf{W}^T\mathbf{V})_{ik}^{\beta-1} \frac{(\mathbf{W}\mathbf{W}^T\mathbf{V})_{ik} - \mathbf{V}_{ik}}{\beta} \tag{6.20}$$

$$+ \alpha_{\mathbf{W}} f_{\mathbf{W}}(\mathbf{W}). \tag{6.21}$$

where the regularization term $\alpha_{\mathbf{W}} f_{\mathbf{W}}(\mathbf{W})$ is used to enforce a certain application dependent characteristic of solutions such as smoothness or sparsity. If we set $\alpha_{\mathbf{W}} = 0$, it reduces to special Csiszár's φ-divergence case, for example, $\beta = 2$, it simplifies to Euclidean distance; $\beta \to 1$, it tends to Kulback-Leibler divergence.

Applying gradient method

$$\begin{aligned}\frac{\partial D_{Ko}(\mathbf{W}\mathbf{W}^T\mathbf{V}||\mathbf{V})}{\partial \mathbf{w}_{ij}} &= \sum_{k,l}(\mathbf{W}\mathbf{W}^T\mathbf{V})_{kl}^{\beta-1}\mathbf{W}_{kj}\mathbf{V}_{il} && (6.22)\\ &- \sum_{k,l}\mathbf{V}_{kl}(\mathbf{W}\mathbf{W}^T\mathbf{V})_{kl}^{\beta-2}\mathbf{W}_{kj}\mathbf{V}_{il} && (6.23)\\ &+ \sum_{k,l}(\mathbf{W}\mathbf{W}^T\mathbf{V})_{il}^{\beta-1}\mathbf{W}_{kj}\mathbf{V}_{kl} && (6.24)\\ &- \sum_{k,l}\mathbf{V}_{il}(\mathbf{W}\mathbf{W}^T\mathbf{V})_{il}^{\beta-2}\mathbf{W}_{kj}\mathbf{V}_{kl} && (6.25)\\ &+ \alpha_{\mathbf{W}}\psi_{\mathbf{W}}(\mathbf{W}) && (6.26)\end{aligned}$$

where the function $\psi_{\mathbf{W}}(\mathbf{W})$ is the derivative of $f_{\mathbf{W}}(\mathbf{W})$. By setting the step size η_{ij} to be

$$\eta_{ij} = \qquad (6.27)$$

$$\frac{\mathbf{W}_{ij}}{\sum_{k,l}\Big((\mathbf{W}\mathbf{W}^T\mathbf{V})_{kl}^{\beta-1}\mathbf{W}_{kj}\mathbf{V}_{il} + (\mathbf{W}\mathbf{W}^T\mathbf{V})_{il}^{\beta-1}\mathbf{W}_{kj}\mathbf{V}_{kl}\Big)}, \qquad (6.28)$$

We have the update rule (6.29):

$$\mathbf{W}_{ij} \leftarrow \mathbf{W}_{ij}\times \qquad (6.29)$$

$$\frac{\sum_{k,l}\Big(\mathbf{V}_{kl}(\mathbf{W}\mathbf{W}^T\mathbf{V})_{kl}^{\beta-2}\mathbf{W}_{kj}\mathbf{V}_{il} + \mathbf{V}_{il}(\mathbf{W}\mathbf{W}^T\mathbf{V})_{il}^{\beta-2}\mathbf{W}_{kj}\mathbf{V}_{kl}\Big) - \alpha_{\mathbf{W}}\psi_{\mathbf{W}}(\mathbf{W})}{\sum_{k,l}\Big((\mathbf{W}\mathbf{W}^T\mathbf{V})_{kl}^{\beta-1}\mathbf{W}_{kj}\mathbf{V}_{il} + (\mathbf{W}\mathbf{W}^T\mathbf{V})_{il}^{\beta-1}\mathbf{W}_{kj}\mathbf{V}_{kl}\Big)} \qquad (6.30)$$

The basis images for NMF and for the family of P-NMF with dimension 49 are shown in Figure 6.1, see **Publication VI**. These are the 49 columns of the corresponding matrices $\mathbf{W}$, again shown as 19×19 images. All the basis images for NMF and P-NMF are non-negative. The basis components of P-NMF type algorithms are spatially more localized and non-overlapped compared to NMF algorithm.

Figure 6.2 shows the reconstructions for one of the face images in the NMF, and P-NMF subspaces of dimension $r = 49$. For comparison, also the original face image is shown. Visually, the P-NMF method is comparable to NMF.

6.5 Propertities of P-NMF

6.5.1 Physical model of P-NMF

In **Publication V**, a physical model of the objective function 6.1 has been set up as follows. Suppose that each observation $\mathbf{v}$ is composed of r non-overlapped parts, i.e. $\mathbf{v} = \sum_{p=1}^{r} \mathbf{v}_p$. We model each part $\mathbf{v}_p$ by the scaling of a base vector $\mathbf{w}_p$ plus a noise vector ϵ_p

$$\mathbf{v}_p = \alpha_p \mathbf{w}_p + \epsilon_p. \tag{6.31}$$

If the base vectors are normalized so that $\mathbf{w}_p^T \mathbf{w}_q = 1$ for $p = q$ and 0 otherwise, then the reconstructed vector of this part is

$$\begin{aligned} \sum_{q=1}^{r} \mathbf{w}_q \mathbf{w}_q^T \mathbf{v}_p &= \sum_{q=1}^{r} \mathbf{w}_q \mathbf{w}_q^T (\alpha_p \mathbf{w}_p + \epsilon_p) && (6.32) \\ &= \sum_{q=1}^{r} \mathbf{w}_q (\alpha_p \mathbf{w}_q^T \mathbf{w}_p + \mathbf{w}_q^T \epsilon_p) && (6.33) \\ &= \alpha_p \mathbf{w}_p + \sum_{q=1}^{r} \mathbf{w}_q \mathbf{w}_q^T \epsilon_p. && (6.34) \end{aligned}$$

The norm of the reconstructed vector is therefore bounded by

$$||\mathbf{v}_p - \sum_{q=1}^{r} \mathbf{w}_q \mathbf{w}_q^T \mathbf{v}_p|| = \left\| \left(I - \sum_{q=1}^{r} \mathbf{w}_q \mathbf{w}_q^T \right) \epsilon_p \right\| \leq (m - r)||\epsilon_p||. \tag{6.35}$$

That is, $\mathbf{w}_p \mathbf{w}_p^T \mathbf{v}_p$ reconstructs $\mathbf{v}_p$ well if the noise level ϵ_p is small enough. According to this model, P-NMF can potentially be applied to signal processing problems where the global signals can be divided into several parts and for each part the observations mainly distribute along a straight line modeled by $\alpha_p \mathbf{w}_p$. This is closely related to Oja's PCA subspace rule [106], which finds the direction of the largest variation, except that the straight line found by P-NMF has to pass through the origin.

6.5.2 Orthogonality and sparseness

The digital images can be represented using sparse representation, in which any given image window is spanned by just a small subset of the available features [9, 73, 110, 153]. As shown in Figure 6.1, the base images of P-NMF algorithm reveal more localized and sparse features than that of NMF algorithm. This means that P-NMF gives better orthogonality of each column in the base matrix compared to the NMF method. For the quantitative analysis of the orthogonality and sparseness, we define two concepts. The first one is the entropy for each of the normalized columns of the basis matrix $\mathbf{W}$ (the basis images) as

$$en_j = -\sum_{i=1}^{m} \mathbf{W}_{ij} \log \mathbf{W}_{ij}, \qquad \sum_{i=1}^{m} \mathbf{W}_{ij} = 1, \tag{6.36}$$

where m is the number of the rows of the basis matrix $\mathbf{W}$. Then we calculate the average of entropies over columns of basis matrix $\mathbf{W}$ (that is, over the basis images),

$$en = \frac{1}{n} \sum_{j=1}^{n} en_j. \tag{6.37}$$

where n is the number of the columns of the basis matrix $\mathbf{W}$.

Generally, a smaller entropy value en shows more localization and sparseness. The experiments show that the P-NMF algorithms clearly have smaller entropy than NMF, hence sparseness for the basis images.

Another way to measure the sparseness is the orthogonality of the basis vectors. Two nonnegative vectors are orthogonal if and only if they do not share any nonzero elements. Therefore the orthogonality between the learned bases reveals the sparsity of the resulting representations, and the amount of localization for facial images. We measure the orthogonality of the learned bases by the following

$$\rho = ||\mathbf{W}^T\mathbf{W} - \mathbf{I}||, \tag{6.38}$$

where $|| \cdot ||$ refers to the Euclidean matrix norm. The columns of $\mathbf{W}$ are first normalized to unit length, so that ρ measures the deviation of the off-diagonal elements of $\mathbf{W}$ from zero. A smaller value of ρ indicates higher orthogonality and ρ equals to 0 when the columns of $\mathbf{W}$ are completely orthogonal.

Figure (6.3, top) compares the orthogonal behavior among PNMF, NNSC and NMF as the learning proceeds using MIT-CBCL image database. PNMF converges to a local minimum with much lower ρ value, that is, higher orthogonality. Figure (6.3, bottom) shows that PNMF is not sensitive to the initial values.

6.5.3 Clustering

P-NMF also has close relation to the k-means clustering, see **Publication VII**. P-NMF can be used for clustering the columns of a data matrix $\mathbf{V}$ in the following way: apply P-NMF for the transposed matrix $\mathbf{V}^T$ under the usual non-negativity constraint and obtain the weight matrix $\mathbf{W}$. The rank r of $\mathbf{W}$ should be set equal to k, the desired number of clusters. If necessary, the columns of $\mathbf{W}$ may be normalized to unit length. $\mathbf{W}$ is then an approximation of the cluster indicator matrix, whose m rows correspond to the m data vectors to be clustered, and k columns correspond to the k clusters: the elements $\mathbf{W}_{ij},\ j = 1, ..., k$ along the i-th row show a "soft" clustering of the i-th data vector into the clusters $C_j,\ j = 1, ..., k$. Correspondingly, the same elements along the j-th column show the degrees by which each of the data vectors belongs to the j-th cluster C_j. Because of the constraint, all these degrees are non-negative. If a unique "hard" clustering is desired, the maximum element on each row can be chosen to indicate the cluster.

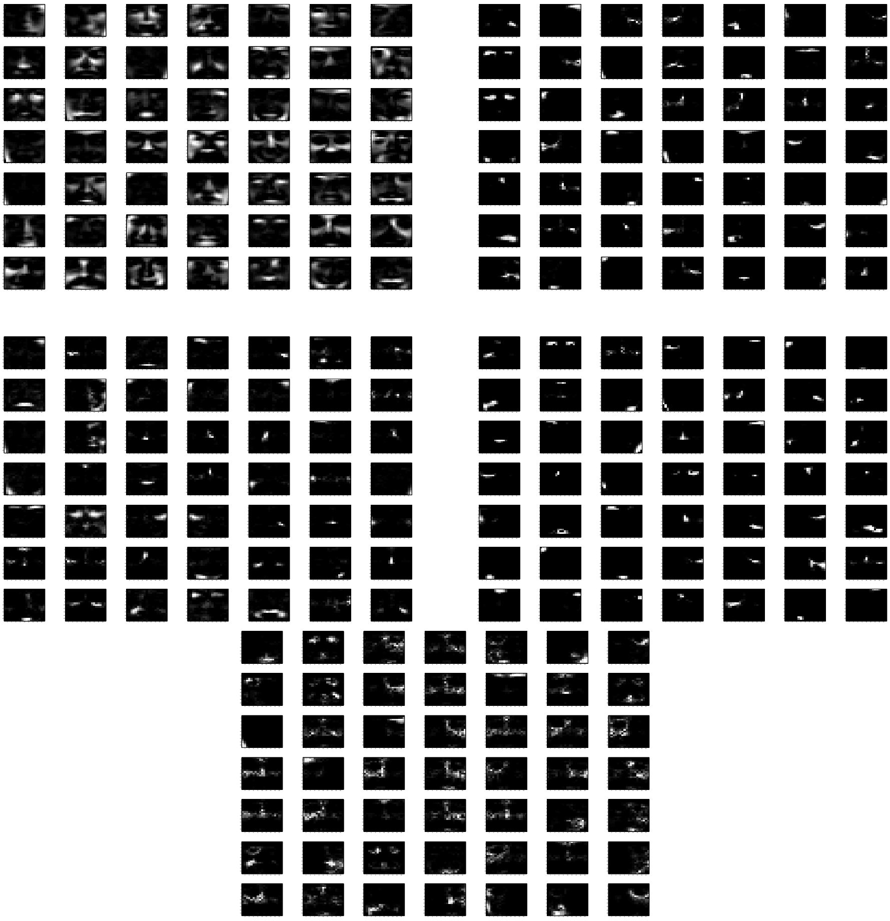


Figure 6.1: The bases of face images by Lee and Seung's NMF (top, left), P-NMF using KL divergence (top, right), Pearson divergence (middle, left), dual Pearson divergence (middle, right), and Hellinger divergence (bottom), with the dimension 49. Each basis component consists of 19 × 19 pixels.

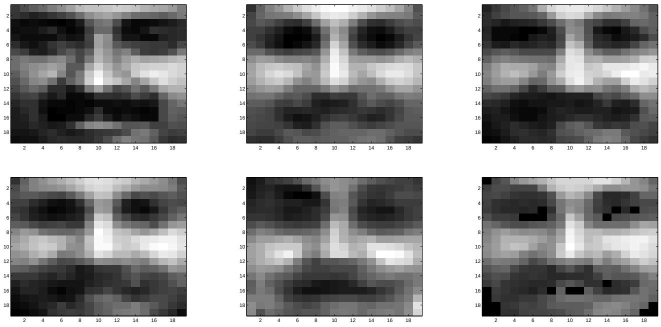
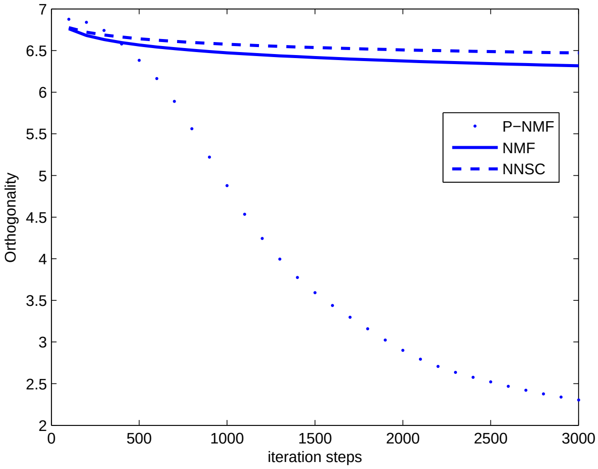


Figure 6.2: From top left to bottom right: The original face image and its reconstructions by Lee and Seung's NMF, P-NMF using KL divergence, Hellinger divergence, Pearson divergence, and dual Pearson divergence.



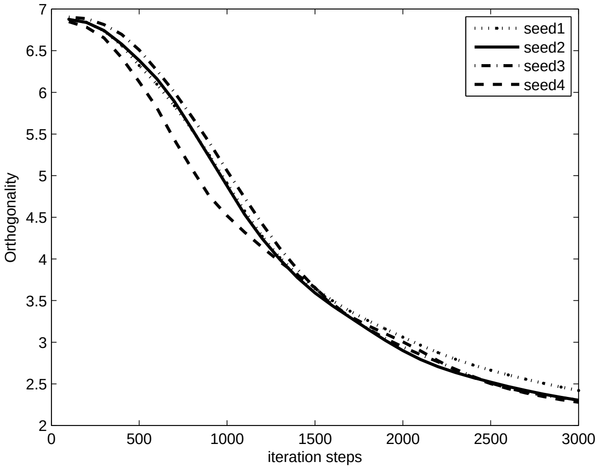


Figure 6.3: Orthogonality versus iterative steps using NMF and PNMF with subdimension 49. Up: PNMF and NMF. Down: PNMF with four different random seeds.

Chapter 7

Conclusions

This thesis addressed the problem of independent component analysis (ICA) and nonnegative matrix factorization (NMF). These methods allow solving many difficult signal processing problems in different application domains. The ICA-based methods have shown to be very successful, and perhaps the most widely used, for performing blind source separation in the general case. If it is known a priori that a nonnegativity condition is valid, then NMF is a widely used technique.

In this thesis, the theoretical conditions for the linear ICA model to qualify as blind separation model were reviewed. It is both sufficient and necessary that the mixing matrix is of full column rank and that there is at most one Gaussian source signal. This gives conditions for the linear ICA techniques to be used as blind system identification methods. The FastICA algorithm is a computationally highly efficient and very popular method for performing the estimation of ICA. The iterative algorithm finds the direction for the weight vector maximizing the nongaussianity of the projection for the data. The local and global convergence of FastICA algorithm have been studied in the thesis. Symmetric FastICA has shown to have at least quadratic convergent speed under a wide range of nonlinearities and source densities. Adding nonnegative constraint for the source data, a nonnegative FastICA algorithm is developed.

Nonnegative Matrix Factorization (NMF) is another type of technique, that was also reviewed in this thesis. Projective NMF is a variant of Nonnegative Matrix Factorization (NMF) in which only one parameter matrix is used instead of two matrices. This makes the method somewhat simpler to compute. An open question is what would be the most appropriate distance measure to be used in minimizing the approximation error, since each different distance measure gives a different solution. Generally, the choice of the distance measures depends on the distribution of data. If the distribution of data fits a true distribution quite well, it might be the same for all those distance measures. Here, several distance measures were introduced for the problem, using variants of Csiszár's φ-divergence as the starting point. Multiplicative gradient algorithms were derived for each, which guarantee the positivity of the approximation, when the algorithms are started from positive initial values.

The sparsity of the ensuing solutions was studied and compared experimentally with each other and NMF. As relevant measures of sparsity, the entropy of the non-negative basis vectors as well as their orthogonality were used. It turned out that on both terms, the P-NMF variants produce significantly sparser representations that NMF. Such sparse representations might act as a bridge between statistical and structural pattern recognition.

Bibliography

[1] Amari, S., Natural gradient works efficiently in learning. *Neural Computation*, 10: 251-276, 1998.

[2] Amari, S., Cichocki, A. and Yang, H. A new learning algorithm for blind source separation. *In Advances in Neural Information Processing Systems*, 8:757-763. MIT Press. 1996.

[3] Anttila, P., Paatero, P., Tapper, U. and Jarvinen, O., Source identification of bulk wet deposition in Finland by positive matrix factorization. *Atmospheric Environment*, 29(14):1705-1718, 1995.

[4] Asari, H., Nonnegative matrix factorization: a possible way to learn sound dictionaries. *Unpulished note*, 2005. Available at http://zadorlab.cshl.edu/∼ asari/pdf/nmf.pdf.

[5] Beckmann, C. F., Noble, J. A. and Smith, S. M., Spatio-temporal accuracy of ICA for FMRI. *NeuroImage*, 13(6):S75, 2001.

[6] Badea, L., Clustering and Metaclustering with Nonnegative Matrix Decompositions. *In 16th European Conference on Machine Learning*, Porto, Portugal, October 2005. Springer Lecture Notes in Computer Science, pp. 10-22, 2005.

[7] Badea, L. and Tilivea, D., Sparse factorizations of gene expressions guided by binding data. *In Pacific Symposium on Biocomputing*, vol. 10, pp. 447-458, 2005.

[8] Bell, A.J. and Sejnowski, T.J., An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7:1129-1159, 1995.

[9] Bell, A.J. and Sejnowski, T.J., The 'independent components' of natural scenes are edge filters. *Vision Research*, 37:3327-3338, 1997.

[10] Berman, A. and Plemmons, R., Non-Negative Matrices in the Mathematical Sciences. *SIAM Press Classics Series*, Philadelphia, PA. 1994.

[11] Berry, M. and Browne, M., Email Surveillance Using Nonnegative Matrix Factorization. *Computational & Mathematical Organization Theory*, 11: 249 - 264, 2005.

[12] Berry, M. and Browne, M., Understanding Search Engines: Mathematical Modeling and Text Retrieval, 2nd Edition. *SIAM Press Classics Series*, Philadelphia, PA, 2005.

[13] Berry, M., Browne, M., Langville, A., Pauca, V. and Plemmons, J., Algorithms and Applications for the approximate Nonnegative Matrix Factorization. *Computational Statistics and Data Analysis.* 52(1):155–173, 2007.

[14] Bertsekas, D., Nonlinear Programming. *Athena Scientific*, Belmont, MA. 1999.

[15] Bingham, E. and Hyvarinen, A., A fast fixed-point algorithm for independent component analysis of complex valued signals. *International Journal of Neural Systems*, 10(1):1-8, February 2000.

[16] Boutsidis, C. and Gallopoulos, E., On SVD-based initialization for nonnegative matrix factorization. *Technique Report HPCLAB-SCG-6/08-05*, University of Patras, Patras, Greece. 2005.

[17] Brunet, J.-P., Tamayo, P., Golub, T. R. and Mesirov, J. P., Metagenes and molecular pattern discovery using matrix factorization. *Proceedings of the National Academy of Sciences of the United States of America*, 101(12):4164-4169, 2004.

[18] Catral, M., Han, L., Neumann, M. and Plemmons, R., On reduced rank nonnegative factorization for symmetric non-negative matrices. *Linear Algebra and Applications*, 393: 107 - 126, 2004

[19] Cardoso, J.-F., Infomax and maximum likelihood for source separation. *IEEE Letters on Signal Processing*, 4:112-114, 1997.

[20] Cardoso, J.-F., Source separation using higher order moments. *In Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'89)*, pp. 2109-2112, Glasgow, UK, 1989.

[21] Cardoso, J.-F., Eigen-structure of the fourth-order cumulant tensor with application to the blind source separation problem. *In Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'90)*, pp 2655-2658, Albuquerque, New Mexico, 1990.

[22] Cardoso, J.-F., High-order contrasts for independent component analysis. *Neural Computation*, 11(1):157-192, 1999.

[23] Cardoso, J.-F. and Laheld, B. H., Equivariant adaptive source separation. *IEEE Transactions on Signal Processing*, 44(12):3017-3030, 1996.

[24] Chen, Z. and Cichocki, A., Nonnegative matrix factorization with temporal smoothness and/or spatial decorrelation constraints. *Technique Report*, Laboratory for Advanced Brain Signal Processing, RIKEN, 2005.

[25] Chen, Z., Cichocki, A. and Rutkowski, T. M., Constrained Non-Negative Matrix Factorization Method for EEG Analysis in Early Detection of Alzheimer's Disease. *In IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP2006)*, pp. 893–896, Toulouse, France, 2006.

[26] Chu, M., Diele, F., Plemmons, R. and Ragni, S., Optimality, computation, and interpretations of nonnegative matrix factorizations. *SIAM Journal on Matrix Analysis*, pp. 4-21, 2004.

[27] Cichocki, A. and Amari, S., Adaptive Blind Signal and Image Processing: Learning Algorithms and Applications. John Wiley & Sons, 2002.

[28] Cichocki, A., Zdunek, R., and Amari, S., Csiszar's divergences for non-negative matrix factorization: family of new algorithms. *In Proceedings of the 6th International Conference on Independent Component Analysis and Blind Signal Separation*, Charleston SC, USA, 2006. Springer Lecture Notes in Computer Science 3889, pp. 32-39, 2006.

[29] Cichocki, A. and Zdunek, R., Multilayer Nonnegative Matrix Factorization using Projected Gradient Approaches. *International Journal of Neural Systems*, 17(6): 431-446, 2007.

[30] Cichocki, A., Zdunek, R. and Amari, S., Hierarchical ALS Algorithms for Nonnegative Matrix and 3D Tensor Factorization, *In: International Conference on Independent Component Analysis and Blind Signal Separation*, London, UK, 2007. Springer Lecture Notes in Computer Science 4666, pp. 169-176, 2007.

[31] Cichocki, A., Phan, A-H., Zdunek, R. and Zhang, L.-Q., Flexible Component Analysis for Sparse, Smooth, Nonnegative Coding or Representation. *In: 14th International Conference on Neural Information Processing (ICONIP07)*, Kitakyushu, Japan, 2007. Springer Lecture Notes in Computer Science, pp. 811 - 820, 2007.

[32] Cichocki, A., Zdunek, R., Plemmons, R. and Amari, S., Novel Multi-layer Nonnegative Tensor Factorization with Sparsity Constraints. *In: Proceedings of the 8th international conference on Adaptive and Natural Computing Algorithms*, Warsaw, Poland, 2007. Springer Lecture Notes in Computer Science 4432, pp. 271-280, 2007.

[33] Cichocki, A. and Zdunek, R., Regularized Alternating Least Squares Algorithms for Non-negative Matrix/Tensor Factorizations. *In: Advances in Neural Networks (ISNN 2007)*, Nanjing, China, 2007. Springer Lecture Notes in Computer Science 4493, pp. 793-802, 2007.

[34] Cichocki, A., Zdunek, R., Choi, S., Plemmons, R. and Amari, S., Nonnegative Tensor Factorization using Alpha and Beta Divergences. *International Conference on Acoustics, Speech, and Signal Processing (ICASSP 07)*, pp. 13931396, Honolulu, Hawaii, USA, 2007.

[35] Cichocki, A., Zdunek, R. and Amari, S., New Algorithms for Non-Negative Matrix Factorization in Applications to Blind Source Separation. *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 06)*, pp. 621-625, Toulouse, France, 2006.

[36] Cichocki, A. and Zdunek, R., Multilayer Nonnegative Matrix Factorization. *Electronics Letters*, 42(16):947-948, 2006.

[37] Comon, P., Independent component analysis - a new concept?. *Signal Processing*, 36:287-314, 1994.

[38] Delfosse, N. and Loubaton, P., Adaptive blind separation of independent sources: A deflation approach. *Signal Processing*, 45: 59-83, 1995.

[39] Dhillon, I. and Sra, S., Generalized nonnegative matrix approximations with bregman divergences. *In: Proceeding of the Neural Information Processing Systems (NIPS) Conference*, Vancouver, B.C. 2005.

[40] Dhillon, I., Concept decompositions for large sparse text data using clustering. *Machine Learning*, 42(1/2):143-175, 2001.

[41] Donoho, D. and Stodden, V., When does nonnegative matrix factorization give a correct decomposition into parts? *In: Seventeenth Annual Conference on Neural Information Processing Systems*, 2003.

[42] Douglas, S.C., On the convergence behavior of the FastICA algorithm. *In: Proceedings of 4th International Conference on Independent Component Analysis and Blind Source Separation*, pp. 409-414, Nara, Japan, 2003.

[43] Douglas, S.C., A statistical convergence analysis of the FastICA algorithm for two-source mixtures. *Proceedings of 39th Asilomar Conference on Signals, Signals, Systems and Computers*, pp. 335-339, Pacific Grove, CA, 2005.

[44] Douglas, S.C., Relatitionships between the FastICA algorithm and the Rayleigh Quotient iteration. *In: Proceedings of 4th International Conference on Independent Component Analysis and Blind Source Separation*, Charleston, SC, 2006. Springer Lecture Notes in Computer Science 3889, pp. 781-789, 2006.

[45] Duann, J.-R., Jung, T.-P., Kuo, W.-J., Yeh, T.-C., Makeig, S., Hsieh, J.-C. and Sejnowski, T., Blind decomposition reveals novel hemodynamics response features. *NeuroImage*, 13(6):S111.

[46] Eggert, J. and Korner, E., Sparse coding and NMF. *Proceedings of IEEE International Joint Conference on Neural Networks*, vol. 4, pp. 2529 - 2533, 2004.

[47] Feng, T., Li, S. Z., Shum, H-Y. and Zhang, H., Local nonnegative matrix factorization as a visual representation. *In Proceedings of the 2nd International Conference on Development and Learning*, pp. 178-193, Cambridge, MA, June 2002.

[48] Finesso, L. and Spreij, P., Approximate nonnegative matrix factorization via alternating minimization. *In: Proceedings of the 16th International Symposium on Mathematical Theory of Networks and Systems*, Leuven, Belgium, 2004.

[49] Frenich, A. G., Galera, M. M., Vidal, J. L. M., Massart, D. L., Torres-Lapasio, J.R., De Braekeleer, K., Wang, J-H. and Hopke, P. K., Resolution of multicomponent peaks by orthogonal projection approach, positive matrix factorization and alternating least squares. *Analytica Chimica Acta*, 411:145-155, 2000.

[50] Fujiwara, T., Ishikawa, S., Hoshida, Y., Inamura, K., Isagawa, T., Shimane, M., Aburatani, H., Ishikawa, Y. and Nomura, H., Non-Negative Matrix Factorization of Lung Adenocarcinoma Expression Profiles. *In 16th International Conference on Genome Informatics*, Yokohama Pacifico, Japan, December 2005.

[51] Gao, Y. and Church, G., Improving molecular cancer class discovery through sparse non-negative matrix factorization. *Bioinformatics*, 21(21):3970-3975, 2005.

[52] Giannakopoulos, X., Karhunen, J. and Oja, E., Experimental comparison of neural algorithms for independent component analysis and blind separation. *International Journal of Neural Systems*, 9: 651-656, 1999.

[53] Gill, P., Murray, W. and Wright, M., Practical Optimization. *Academic Press*, London. 1981

[54] Golub, G. and van Loan, C., Matrix Computations. *The Johns Hopkins University Press*, 1996.

[55] Gonzalez, E. and Zhang, Y., Accelerating the Lee-Seung algorithm for non-negative matrix factorization. *Technique Report TR-05-02*, Rice University. Grieve, T., October 2003.

[56] Guillamet, D., Bressan, M. and Vitrià, J., A weighted non-negative matrix factorization for local representations. *In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. 1. Kavai, HI, pp. 942 - 947, 2001.

[57] Guillamet, D., Schiele, B. and Vitrià, J.,P. O. Analyzing non-negative matrix factorization for image classification. *In Proceedings of the 16th International Conference on Pattern Recognition (ICPR02)*, Vol. II, pp. 116-119, 2002.

[58] Guillamet, D. and Vitrià, J., Classifying faces with nonnegative matrix factorization. *In: Proceedings of the 5th Catalan Conference for Artificial Intelligence (CCIA 2002)*. pp. 2431, Castelló de la Plana, Spain, 2002.

[59] Guillamet, D. and Vitrià, J., Determining a suitable metric when using non-negative matrix factorization. *In 16th International Conference on Pattern Recognition*, vol. 2, pp. 128- 131, 2002.

[60] Guillamet, D., Vitrià, J. and Schiele, B., Introducing a weighted nonnegative matrix factorization for image classification. *Pattern Recognition Letters*, 24(14):2447-2454, October 2003.

[61] Hamza, A. B. and Brady, D., Reconstruction of reflectance spectra using robust non-negative matrix factorization. *IEEE Transactions on Signal Processing*, 54(9):3637-3642, 2006

[62] Hesse, C. W. and James, C. J., The FastICA Algorithm With Spatial Constraints. *SPLetters*, 12(11):792-795, November 2005.

[63] Hoyer, P. O., Nonnegative sparse coding. *IEEE Workshop on Neural Networks for Signal Processing*, Martigny, 2002.

[64] Hoyer, P. O., Non-negative Matrix Factorization with sparseness constraints. *Journal of Machine Learning Research*, 5:1457-1469, 2004.

[65] Hu, C., Zhang, B., Yan, S., Yang, Q., Yan, J., Chen, Z. and Ma, W.-Y., Mining Ratio Rules Via Principal Sparse Non-NegativeMatrix Factorization. *In Fourth IEEE International Conference on DataMining (ICDM'04)*, pp. 407-410, 2004.

[66] Hyvärinen, A., Survey on independent component analysis. *Neural Computation Surveys*, 2:94-128, 1999.

[67] Hyvärinen, A., New Approximations of Differential Entropy for Independent Component Analysis and Projection Pursuit. *In Advances in Neural Information Processing Systems (NIPS*97)*, 10: 273-279, MIT Press, 1998.

[68] Hyvärinen, A., One-Unit Contrast Functions for Independent Component Analysis: A Statistical Analysis. *In Neural Networks for Signal Processing VII (Proceedings of the IEEE NNSP Workshop '97, Amelia Island, Florida)*, pp. 388–397, 1997.

[69] Hyvärinen, A., Fast and robust fixed-point algorithms for independent component analysis. *IEEE Transactions on Neural Networks*, 10(3):626-634, 1999.

[70] Hyvärinen, A., The fixed-point algorithm and maximum likelihood estimation for independent component analysis. *Neural Processing Letters*, 10(1):1-5, 1999.

[71] Hyvärinen, A. and Oja, E., A fast fixed-point algorithm for independent component analysis. *Neural Computation*, 9(7):1483-1492, 1997.

[72] Hyvärinen, A. and Oja, E., One-unit learning rules for independent component analysis. *In Advances in Neural Information Processing Systems*, 9:480486, MIT Press, 1997.

[73] Hyvärinen, A. and Hoyer, P. O., Emergence of phase and shift invariant features by decomposition of natural images into independent feature subspaces. *Neural Computation*, 13: 1527–1558, 2001.

[74] Hyvärinen, A. and Inki, M., Estimating Overcomplete Independent Component Bases for Image Windows. *Journal of Mathematical Imaging and Vision*, 17(2): 139-152, September 2002.

[75] Hyvärinen, A., Karhunen, J. and Oja, E., Independent Component Analysis. Wiley, New York, 2001.

[76] Inamura, K., Fujiwara, T., Hoshida, Y., Isagawa, T., Jones, M. H., Virtanen, C., Shimane, M., Satoh, Y., Okumura, S., Nakagawa, K., Tsuchiya, E., Ishikawa, S., Aburatani, H., Nomura, H. and Ishikawa, Y., Two subclasses of lung squamous cell carcinoma with different gene expression profiles and prognosis identified by hierarchical clustering and non-negative matrix factorization. *Oncogene*, 24:7105-7113, June 2005.

[77] Jolliffe, I. T., Principal Component Analysis. *Springer-Verlag*, New York, NY, 2nd edition, October 2002.

[78] Kawamoto, T., Hotta, K., Mishima, T., Fujiki, J., Tanaka, M. and Kurita, T., Estimation of single tones from chord sounds using non-negative matrix factorization. *Neural Network World*, 3(2000), 429-436.

[79] Kim, P. M. and Tidor, B., Subsystem Identification Through Dimensionality Reduction of Large-Scale Gene Expression Data. *Genome Research*, 13:1706-1718, 2003.

[80] Koldovsky, Z., Tichavsky, P. and Oja, E., Cramer-Rao lower bound for linear independent component analysis. *Proceedings of International Conference on Acoustics, Speech, and Signal Processing (ICASSP '05)*, March 2005.

[81] Koldovsky, Z., Tichavsky, P. and Oja, E., Efficient variant of algorithm FastICA for independent component analysis attaining the Cramer-Rao lower bound. *IEEE Transactions on Neural Networks*, 17(5): 1265-1277, 2006.

[82] Kolenda, T., Hansen, L. K. and Larsen, J., Signal detection using ICA: Application to chat room topic spotting. *In Proceedings of International Workshop on Independent Component Analysis and Blind Source Separation (ICA'01)*, pp. 540-545, San Diego, USA. 2001.

[83] Kolenda, T., Hansen, L. K. and Sigurdsson, S., Independent components in text. *In Girolami, M., editor, Advances in Independent Component Analysis, Perspectives on Neural Computation*, chapter 13. Springer Verlag, Berlin, Germany. 2000.

[84] Kompass, R., A generalized divergence measure for nonnegative matrix factorization. *Neuroinformatics Workshop*, Torun, Poland. September 2005.

[85] Kuhn, H. W. and Tucker, A. W., Nonlinear programming. *Proceedings of 2nd Berkeley Symposium*, pp. 481-492, Berkeley, University of California Press, 1951.

[86] Lawson, C. and Hanson, R., Solving Least Squares Problems. *SIAM Press Classics Series*, Philadelphia, PA, 1995.

[87] Langville, A., Meyer, C., Albright, R., Cox, J. and Duling, D., Algorithms, initializations, and convergence for the nonnegative matrix factorization. *Twelfth Annual SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2006.

[88] Lawrence, J., Artzi, A., DeCoro, C., Matusik, W., Pfister, H., Ramamoorthi, R. and Rusinkiewicz, S., Inverse shade trees for non-parametric material representation and editing. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 25(3):735-745, 2006.

[89] Lawrence, J., Rusinkiewicz, S. and Ramamoorthi, R., Efficient BRDF Importance Using a Factored Representation. *ACM Transactions on Graphics (TOG)*, 23(3): 496 - 505, 2004.

[90] Lee, D. D. and Seung, H. S., Unsupervised learning by convex and conic coding. *In Advances in Neural Information Processing Systems*, vol. 9, pp. 515-521, The MIT Press, 1997.

[91] Lee, D. D. and Seung, H. S., Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788–791, 1999.

[92] Lee, D. D. and Seung, H. S., Algorithms for non-negative matrix factorization. *In: Advances in neural information processing*, Vol. 13, MIT Press, 2001.

[93] Lewicki, M. S. and Sejnowski,T. J., Learning overcomplete representations. *Neural Computation*, 12:337-365, 2000.

[94] Li, S. Z., Hou, X. W., Zhang, H. J. and Cheng, Q. S., Learning spatially localized, parts-based representation. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 207-212, 2001.

[95] Lin, C.-J., On the convergence of multiplicative update algorithms for non-negative matrix factorization. *IEEE Transactions on Neural Networks*, 18(6):1589-1596, 2007.

[96] Lin, C.-J., Projected gradient methods for non-negative matrix factorization. *Neural Computation*, 19(10): 2756-2779, 2007.

[97] Liu, W. X., Zheng, N. N. and Lu, X. F., Non-negative matrix factorization for visual coding. *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'03)*, Vol. 3, pp. 293-296, April 2003.

[98] Makeig, S., Jung, T.-P., Bell, A. J., Ghahremani, D. and Sejnowski, T. J., Blind separation of auditory event-related brain responses into independent components. *Proceedings of the National Academy of Sciences of the United States of America*, 94(20):10979-10984, 1997.

[99] Makeig, S.,Bell, A.J., Jung, T.-P. and Sejnowski, T.-J., Independent component analysis of electroencephalographic data. *In Advances in Neural Information Processing Systems*, 8: 145-151, MIT Press, 1996.

[100] Mao, Y., Global convergence analysis of a Discrete Time nonnegative ICA algorithm. *IEEE Transactions on Neural Networks*, 17(1):253- 256, 2006.

[101] Mao, Y., Fan, X. and Liu, Q., Monotonic Convergence of a Nonnegative ICA Algorithm on Stiefel Manifold. *In: 14th International Conference on Neural Information Processing (ICONIP 06)*, Hong Kong, China, 2006. Springer Lecture Notes in Computer Science 4232, pp. 1098-1106, 2006.

[102] Nocedal, J., Wright, S., Numerical Optimization. *Springer*, 1999.

[103] Oja, E., Convergence of the symmetrical FastICA algorithm. *In 9th International Conference on Neural Information Processing (ICONIP)*, pp. 18-22, Singapore, November 2002.

[104] Oja, E., Nonlinear PCA criterion and maximum likelihood in independent component analysis. *In Proceedings of International Workshop on Independent Component Analysis and Blind Signal Separation (ICA'99)*, pp. 143-148, Aussois, France, 1999.

[105] Oja, E., The nonlinear PCA learning rule in independent component analysis. *Neurocomputing*, 17(1): 25-46, 1997.

[106] Oja, E., Principal components, minor components and linear neural networks. *Neural Networks*, 5: 927-935, 1992.

[107] Oja, E. and Karhunen, J., On stochastic approximation of the eigenvectors and eigenvalues of the expectation of a random matrix. *Journal of Math. Analysis and Applications*, 106:69-84, 1985.

[108] Oja, E. and Plumbley, M. D., Blind separation of positive sources by globally convergentgradient search. *Neural Computation*, 16(9): 1811-1825, 2004.

[109] Oja, E. and Plumbley, M. D., Blind separation of positive sources using nonnegative PCA. *In Proceedings of the International Conference on Independent Component Analysis and Blind Signal Separation (ICA'03)*, pp. 11-16, Nara, Japan, Apr. 2003.

[110] Olshausen, B. A. and Field, D. J., Natural image statistics and efficient coding. *Network*, 7: 333–339, 1996.

[111] Paatero, P., A weighted nonnegative least squares algorithm for three-way 'PARAFAC' factor analysis. *Chemometrics and Intelligent Laboratory Systems*, 38:223-242, 1997.

[112] Paatero, P., Least squares formulation of robust non-negative factor analysis. *Chemometrics and Intelligent Laboratory Systems*, 37(1):23-35, 1997 May.

[113] Paatero, P., The Multilinear Engine - a table-driven least squares program for solving multilinear problems, including the n-way parallel factor analysis model. *Journal of Computational and Graphical Statistics*, 8(4):854-888, 1999.

[114] Paatero, P., Hopke, P. K., Song, X-H. and Ramadan, Z., Understanding and controlling rotations in factory analytic models. *Chemometrics and Intelligent Laboratory Systems*, 60:253-264, 2002.

[115] Paatero, P. and Tapper, U., Analysis of different modes of factor analysis as least squares fit problems. *Chemometrics and Intelligent Laboratory Systems*, 18(2):183194, 1993.

[116] Paatero, P. and Tapper, U., Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5:111-126, 1994.

[117] Paatero, P., Tapper, U., Aalto, P. and Kulmala, M., Matrix factorization methods for analysing diffusion battery data. *Journal of Aerosol Science*, 22(Supplement 1):S273-S276, 1991.

[118] Papadias, C. B., Globally convergent blind source separation based on a multiuser kurtosis maximization criterion. *IEEE Transactions on Signal Processing*, 48(12): 3508-3519, December 2000.

[119] Papoulis, A., Probability, Random Variables, and Stochastic Processes. *McGraw- Hill*, 3rd edition, 1991.

[120] Pascual-Montano, A., Carazo, J. M., Kochi, k., Lehmann, D. and Pascual-Marqui, R. D., Nonsmooth Nonnegative Matrix Factorization (nsNMF). *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(3):403-415, March 2006.

[121] Pauca, P., Piper, J., Plemmons, R., Nonnegative matrix factorization for spectral data analysis. *Linear Algebra and Its Applications*, 416(11): 29-47, 2006

[122] Pauca, V., Shahnaz, F., Berry, M., Plemmons, R., Text Mining Using Non-Negative Matrix Factorizations. *In: Proceedings of the Fourth SIAM International Conference on Data Mining*, Lake Buena Vista, FL. April 22-24, 2004.

[123] Piper, J., Pauca, V. P., Plemmons, R. J. and Giffin, M., Object Characterization from Spectral Data using Nonnegative Factorization and Information Theory. *In Proceedings of the AMOS Technical Conference*, pp. 1-12, Maui, Hawaii, USA, September 2004.

[124] Plumbley, M. D., Conditions for nonnegative independent component analysis. *SPLetters*, 9(6): 177-180, June 2002.

[125] Plumbley, M. D., Adaptive lateral inhibition for nonnegative ICA. *In: Proceedings of the International Conference on Independent Component Analysis and Blind Signal Separation*, pp. 516-521, San Diego, CA, December 2001.

[126] Plumbley, M. D., Algorithms for Nonnegative Independent Component Analysis. *IEEE Transactions on Neural Networks*, 4(3): 534-543, May 2003

[127] Plumbley, M. D., Optimization using Fourier expansion over a geodesic for non-negative ICA. *In: Proceedings of the International Conference on Independent Component Analysis and Blind Signal Separation (ICA'04)*, pp. 49-56, Granada, Spain, September 2004.

[128] Plumbley, M. D., Oja, E., A "nonnegative PCA" algorithm for independent component analysis. *IEEE Transactions on Neural Networks*, 15(1):66 - 76, 2004.

[129] Powell, M., An efficient method for finding the minimum of a function of several variables without calculating derivatives. *The Computer Journal*, 7: 155-162, 1964.

[130] Powell, M., On search directions for minimization. *Mathematical Programming*, 4: 193-201, 1974.

[131] Qin, Y., Oduyemi, K. and Chan, L. Y., Comparative testing of PMF and CFA models. *Chemometrics and Intelligent Laboratory Systems*, 61:75-87, 2002.

[132] Ramadan, Z., Eickhout, B., Song, X-H. Buydents, L. M. C. and Hopke, P. K., Comparison of positive matrix factorization and multilinear engine for the source apportionment of particulate pollutants. *Chemometrics and Intelligent Laboratory Systems*, 66:15-28, 2003.

[133] Rao, N., Shepherd, S. J. and Yao, D., Extracting characteristic patterns form genome-wide expression data by non-negative matrix factorization. *In IEEE Computational Systems Bioinformatics Conference*, pp. 570- 571, 2004.

[134] Regalia, P. A. and Kofidis, E., Monotonic convergence of fixed-point algorithms for ICA. *IEEE Transactions on Neural Network*, 14(4): 943-949, July 2003.

[135] Sajda, P. Du, S., Brown, T., Parra, L. and Stoyanova, R., Recovery of Constituent Spectra in 3D Chemical Shift Imaging using Nonnegative Matrix Factorization. *In 4th International Symposium on Independent Component Analysis and Blind Signal Separation*, pp. 71-76, Nara, Japan, April 2003.

[136] Salakhutdinov, R., Roweis, S. T. and Ghahramani, Z., On the convergence of bound optimization algorithms. *Uncertainty in Artificial Intelligence*, 19: 509 - 516, 2003.

[137] Sanchez, E. and Kowalski, B. R., Tensorial resolution: A direct trilinear decomposition. *Journal of Chemometrics*, 4:29-45, 1990.

[138] Sandler, M., On the use of linear programming for unsupervised text classification. *In The Eleventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Chicago, IL, 2005.

[139] Sands, R. and Young, F. W., Component models for three-way data: an alternating least squares algorithm with optimal scaling features. *Psychometrika*, 45:39-67, 1980.

[140] Shahnaz, F., Berry, M., Pauca, V. and Plemmons, R., Document clustering using nonnegative matrix factorization. *Information Processing & Management*, 42(2): 373 - 386. 2006.

[141] Shashua, A. and Hazan, T., Non-Negative Tensor Factorization with Applications to Statistics and Computer Vision. *Proceedings of the 22nd international conference on Machine learning*, pp. 792 - 799, 2005.

[142] Shalvi, O. and Weinstein, E., New criteria for blind deconvolution of non-minimum phase systems (channels). *IEEE Transactions on Information Theory*, 36(2): 312-321, March 1990.

[143] Shashua, A., Zass, R. and Hazan, T., Multiway Clustering using Super-symmetric Nonnegative Tensor Factorization. *In: Proceedings of the 9th European Conference on Computer Vision*, Graz, Austria, May 7-13, 2006. Springer Lecture Notes in Computer Science 3954, pp. 595-608. 2006.

[144] Shen, H., Kleinsteuber, M. and Huper, K., Local Convergence Analysis of FastICA and Related Algorithms. *IEEE Transactions on Neural Networks*, 19(6): 1022 - 1032, June 2008.

[145] Skillicorn, D. B., McConnell, S. M. and Soong, E.Y., Handbooks of data mining using matrix decompositions. *School of Computing*, Queen's University, Kingston, Canada, 2003.

[146] Smaragdis, P., Non-negative matrix factor deconvolution; extraction of multiple sound sources from monophonic inputs. *In 5th International Conference on Independent Component Analysis and Blind Source Separation (ICA04)*, pp. 494-499, Granada, Spain, 2004. Springer Lecture Notes in Computer Science 3195, pp. 494-499, 2004.

[147] Smaragdis, P. and Brown, J. C., Non-negative matrix factorization for polyphonic music transcription. *In IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA03)*, pp. 177-180, October 2003.

[148] Smilde, A., Bro, R. and Geladi, P., Multi-way Analysis. *Wiley*, West Sussex, England, 2004.

[149] Spratling, M. W., Learning image components for object recognition. *Journal of Machine Learning Research*, 7:793-815, 2006.

[150] Tichavsky, P., Koldovsky, Z. and Oja, E., Performance analysis of the FastICA algorithm and Cramer-rao bounds for linear independent component analysis. *IEEE Transactions on Signal Processing*, 54(4):1189 - 1203, April 2006.

[151] Tropp, J., Literature survey: Nonnegative matrix factorization. *University of Texas at Austin*, Unpublished, 2003. Available at

[152] Tropp, J., Topics in Sparse Approximation, *Ph.D. Dissertation*, University of Texas at Austin, 2004.

[153] van Hateren J. H. and van der Schaaf, A., Independent component filters of natural images compared with simple cells in primary visual cortex. *Proc. Royal Soc. London B*, 265: 2315–2320, 1998.

[154] Vigaario, R., Dipole modeling in FastICA decomposition of evoked responses. *In Proceedings of the 2nd International Workshop on Independent Component Analysis and Blind Source Separation (ICA'00)*, Helsinki, Finland, 2000.

[155] Wang, K., Zheng, N. and Liu, W., Natural image matting with nonnegative matrix factorization. *In IEEE International Conference on Image Processing*, vol. 2, pp. 1186-1189, September 2005.

[156] Wang, Y., Jiar, Y., Hu, C. and Turk, M., Fisher non-negative matrix factorization for learning local features. *In: Asian Conference on Computer Vision*, Korea, January 27-30 2004.

[157] Wild, S. M., Seeding Non-Negative Matrix Factorizations with the Spherical K-Means Clustering. *Master thesis*, University of Colorado, April 2003.

[158] Wild S, Curry J. and Dougherty A. Motivating Non-Negative Matrix Factorizations. *In: Proceedings of the Eighth SIAM Conference on Applied Linear Algebra, Williamsburg*, VA, July 15-19, 2003.

[159] Wild, S. M., Curry, J. H. and Dougherty, A. M., Improving Non-Negative Matrix Factorizations Through Structured Initialization. *Pattern Recognition*, 37(11):2217-2232, November 2004.

[160] Winter, S., Sawada, H., Araki, S. and Makino, S., Overcomplete BSS for convolutive mixtures based on hierarchical clustering. *In Proceedings of International Workshop on Independent Component Analysis and Blind Signal Separation (ICA'04)*, pp. 652 - 660, Granada, Spain, 2004.

[161] Wold, H., Nonlinear estimation by iterative least squares procedures. *In: David, F. (Ed.), Research Papers in Statistics*, pp. 411-444, John Wiley and Sons, Inc., New York, 1966.

[162] Wold, H., Soft modelling by latent variables: nonlinear iterative partial least squares approach. *Perspectives in Probability and Statistics*, pp. 411-444, Academic Press, London, 1975.

[163] Xu, B., Lu, J., Huang, G., A constrained non-negative matrix factorization in information retrieval. *IEEE International Conference on Information Reuse and Integration*, pp. 273 - 277, 2003.

[164] Xu, L., Least mean square error reconstruction principle for self-organizing neural nets. *Neural Networks*, 6(5): 627-648, 1993.

[165] Xu, L., Oja, E. and Suen, C., Modified Hebbian learning for curve and surface fitting. *Neural Networks*, 5:441 - 457, 1992.

[166] Xu, W., Liu, X. and Gong, Y., Document clustering based on nonnegative matrix factorization. *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pp. 267-273, Toronto, 2003.

[167] Yang, Z. and Oja, E., Linear and Nonlinear Projective Nonnegative Matrix Factorization. In submission.

[168] Zangwill, W., Minimizing a function without calculating derivatives. *The Computer Journal*, 10: 293-296, 1967.

[169] Zdunek, R. and Cichocki, A., Non-negative matrix factorization with quasi-newton optimization. *In: Proceedings of the 8th International Conference on Artificial Intelligence and Soft Computing*, Zakopane, Poland, June 25-29 2006. Springer Lecture Notes in Artificial Intelligence, Vol. 4029, pp. 870-879, 2006.

[170] Zhang, J., Wei, L., Miao, Q. and Wang, Y., Image fusion based on nonnegative matrix factorization. *In International Conference on Image Processing*, vol. 2, pp. 973-976, 2004.

[171] Zheng, C. H., Huang, D. S., Sun, Z. L., Lyu, M. R. and Lok, T. M., Nonnegative independent component analysis based on minimizing mutual information technique. *Neurocomputing*, 69: 878 - 883, 2006.