A Study of Outliers for Robust Independent Component Analysis

by

Neil Gadhok

A Thesis
Submitted to the Faculty of Graduate Studies
in Partial Fulfillment of the Requirements for the Degree of

MASTER OF SCIENCE

Department of Electrical and Computer Engineering
University of Manitoba
Winnipeg, Manitoba

Thesis Advisor: W. Kinsner, Ph.D. P.Eng.

© Neil Gadhok, December 2006

THE UNIVERSITY OF MANITOBA

FACULTY OF GRADUATE STUDIES

*****

COPYRIGHT PERMISSION

A Study of Outliers for Robust Independent Component Analysis

by

Neil Gadhok

A Thesis/Practicum submitted to the Faculty of Graduate Studies of The University of Manitoba in partial fulfillment of the requirement of the degree

of

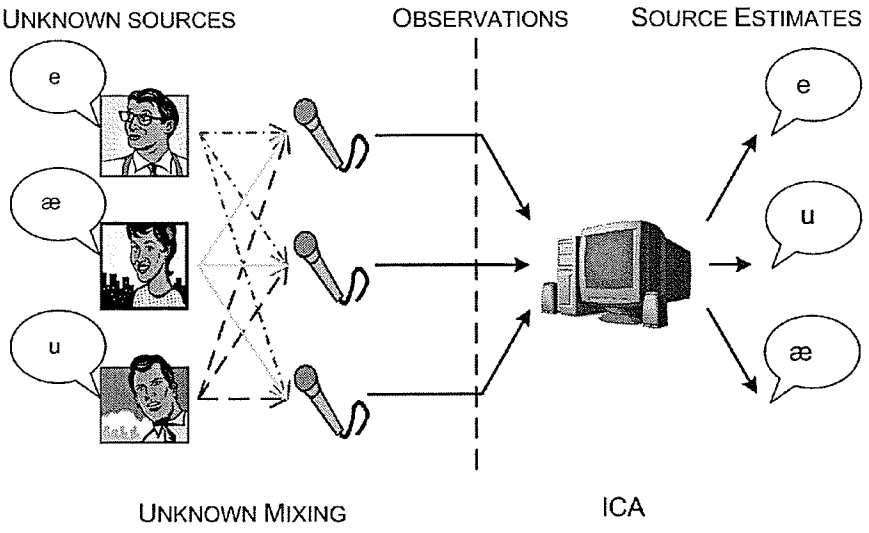
Master of Science

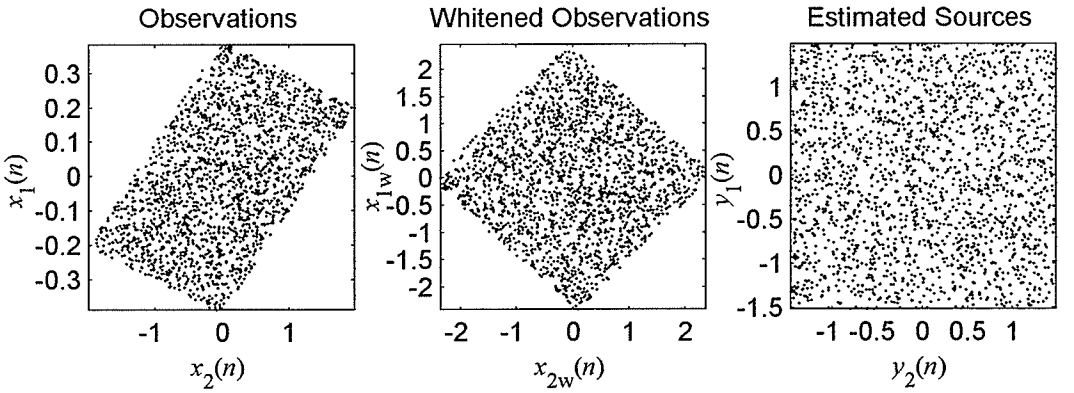
Neil Gadhok © 2006

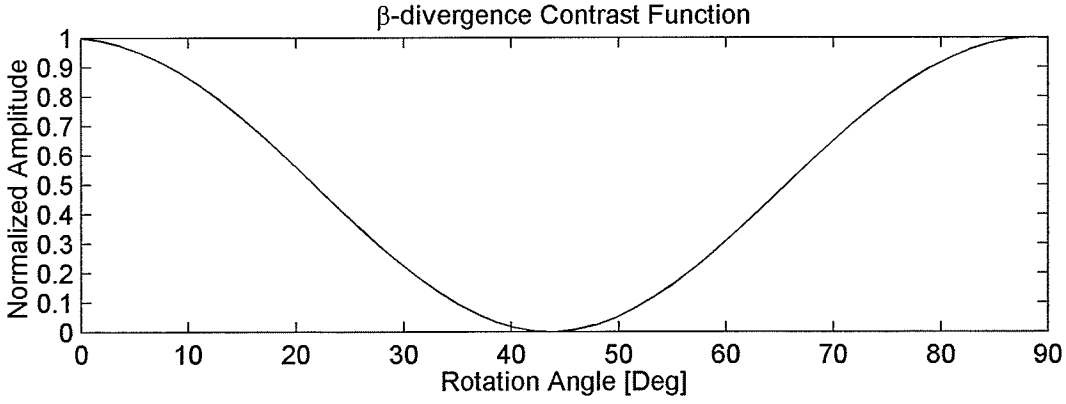
Permission has been granted to the Library of the University of Manitoba to lend or sell copies of this thesis/practicum, to the National Library of Canada to microfilm this thesis and to lend or sell copies of the film, and to University Microfilms Inc. to publish an abstract of this thesis/practicum.

This reproduction or copy of this thesis has been made available by authority of the copyright owner solely for the purpose of private study and research, and may only be reproduced and copied as permitted by copyright laws or with express written authorization from the copyright owner.

Visual Abstract







Abstract

This thesis presents a study on the outlier robustness of *independent component analysis* (ICA) algorithms. ICA is a higher-order statistical method that learns from sampled data how to separate out the true source signals from their linear mixture; *e.g.*, demixing two heart signals, one from a mother and the other from a fetus, based only on external observations of that mixture. ICA is successful in demixing signals up to an arbitrary scale (amplitude) and permutation in the source sequence. Unfortunately, since contamination of biomedical recordings by outliers is an unavoidable aspect in signal processing, ICA algorithm implementations are inherently sensitive to outliers, (*i.e.*, extreme values that do not comply with the *probability density function*, PDF, of the signal) because they are based on *higher-order statistics* (HOS). Thus, the primary objective of this thesis is to measure the outlier sensitivity of five well-known ICA algorithms (FastICA, Extended-Infomax, JADE, RADICAL and Beta (β)-divergence), and to rank their robustness. The problem is approached from the view that an ICA algorithm is a contrast function and an optimization technique. Specifically, the contrast function is one which rotates the data in order to separate the sources. The interest in the rotation sensitivity of the contrast function to outliers led to the development of two new outlier robustness metrics called the optimum angle of rotation error, and the contrast function difference. To rank the outlier robustness of the algorithms, a number of simulations were performed in an unbiased optimization landscape. This required an implementation of the β-divergence algorithm in Matlab. An unbiased optimization landscape was selected to ensure that all of the algorithms had the ability to search the same optimization space to find a solution. The simulations revealed that the β-divergence algorithm had an average Amari separation performance index near 0.1, an optimum angle of rotation error near 4 degrees, and a contrast function difference of less than 1 %. When compared to the other algorithms, our implementation of β-divergence was the most robust to outliers, and should be used when dealing with outlier contaminated mixtures.

Acknowledgements

I would like to thank my advisor, Dr. Kinsner, for suggesting this research topic, his guidance on this research, and numerous improvements to this thesis. In addition, I would like to thank him for his tremendous support during ventures in Space-related projects (*e.g.* the 6th European Space Agency student parabolic flight campaign in Bordeaux, France, July 2003) and the broader aspects of engineering.

I would like to acknowledge everyone in the Delta Research Group, past and present, including Michael Potter, Aram Faghfouri, Stephen Dueck, Hakim El-Boustani, Lelia Safavian, Sharjeel Siddiqui and Robert Barry for their help, support and insightful discussions.

Finally, I would like to thank my family and friends for their support during my study.

Table of Contents

List of Figures

List of Tables

List of Abbreviations and Acronyms

API	Amari separation performance index
BFGS	Broyden-Fletcher-Goldfarb-Shanno
BP	breakdown point
BSS	blind source separation
CF	contrast function
CVF	change of variance function
DOA	direction of arrival
DS-CDMA	direct-sequence code division multiple access
DSE	differential Shannon entropy
ECG	electrocardiogram
EEG	electroencephalogram
EMG	electromyogram
EVD	eigenvalue decomposition
FastICA	fast fixed-point independent component analysis
FS	full scale
HOS	higher-order statistics
ICA	independent component analysis
IF	influence function
JADE	joint approximate diagonalization of eigenmatrices
ksps	kiloSamples per second
MCD	minimum covariance determinant
MLE	maximum likelihood estimation
MUD	multiuser detection
PCA	principal component analysis
PDF	probability density function
PMF	probability mass function

PP	projection pursuit
RADICAL	robust, accurate, direct independent component analysis algorithm
RV	random variable
SOS	second order statistics
SSS	strong-sense stationary
WSS	weak-sense stationary

List of Symbols

β	A scalar parameter, $0 \leq \beta \leq 1$, used in the β-divergence linear-ICA algorithm to determine the influence of data points on the divergence estimate, p. 2.
$\mathbf{A}$	A mixing matrix.
$\bar{x}$	A random variable, p. 6.
$\bar{x}_m$	The mth random variable from, $\bar{\mathbf{x}}$, a column vector of random variables, p. 6.
$\check{x}$	A 1-dimensional digital signal where the N samples are ordered in increasing magnitude.
$C(\mathbf{x}(n), \mathbf{W})$	A contrast function, which at its maximum or minimum, $\mathbf{W}$ should separate a mixture of sources., p. 8.
$\mathbf{D}$	A diagonal matrix, p. 31.
$g(\bullet)$	A nonlinear function, p. 38.
$G(\bullet)$	A function used in the approximation of differential Shannon entropy, p. 43.
$h(\bullet)$	A nonlinear function, p. 38.
H_n	The negentropy of a random variable, p. 42.
$I(\bullet)$	The mutual information between random variables, p. 37.
K	The number of bits a signal has been quantized to, p. 3.
κ	The normalized kurtosis or kurtosis excess of a random variable, p. 40.
M	The number of 1-dimensional signals in a column vector of signals, p. 6.
$\bar{\mathbf{x}}$	A column vector of M random variables, p. 6.
$\mathcal{E}(\bullet)$	Expectation operator, p. 6.
μ	The mean of a random variable, p. 39.
N	The number of samples in a 1-dimensional digital signal, p. 6.
$\mathbf{P}$	A permutation matrix, p. 31.
$p_m(\bullet)$	A family of parametric distributions, where m ranges from 1 to M, p. 36.
$p(s_1(t))$	Probability distribution function of the random variable and analog signal $s_1(t)$, p. 28.
Q	Quantization step, p. 3.
σ_2	The variance or 2nd statisitcal central moment of a random variable, p. 40.

σ_4	The kurtosis or 4th statistical central moment of a random variable, p. 40.
$\mathbf{s}(t)$	A column vector of M 1-dimensional analog source signals, p. 8.
$\mathbf{V}$	A matrix used for measuring the Amari separation performance index, p. 69.
$\mathbf{W}$	A demixing matrix.
$\mathbf{w}$	A row of a demixing matrix W, p. 49.
$x_m(n)$	The mth signal of a column vector of M 1-dimensional digital or digital mixture signals, where m ranges from 1 to M. Also represents a digital mixture signal, p. 6.
$x_m(t)$	The mth signal of a column vector of M 1-dimensional analog or analog mixture signals, where m ranges from 1 to M, p. 6.
$x(n)$	A 1-dimensional digital signal where n is the index of the time, t_n, at which the analog signal was sampled. The integer value n ranges from 1 to N, p. 6.
$\mathbf{x}(n)$	A column vector of M 1-dimensional digital or digital mixture signals, p. 6.
$x(t)$	A 1-dimensional analog or analog mixture signal evolving over time t, p. 6.
$\mathbf{x}(t)$	A column vector of M 1-dimensional analog or analog mixture signals, p. 6.
$\mathbf{y}(n)$	A column vector of M 1-dimensional estimated digital source signals, p. 8.

Chapter I

INTRODUCTION

The standard assumptions in conventional signal processing include Gaussianity, linearity, and stationarity. However, in modern-day signal processing, assumptions have now shifted to allow for non-Gaussian distributions, non-linear transforms and non-stationarities [30]. In addition, the assumption of sampled data originating from a single source is being replaced with one where the observed data comes from a mixture of multiple sources. The separation of sources based only on observations of those mixtures, known as the *blind source separation* (BSS) problem, is seen by researchers and scientists as a necessary preprocessing step in order to obtain uncontaminated data for analysis. A method from the field of intelligent signal processing called *independent component analysis* (ICA) [36], [17], [60], [43], [30] is a promising solution to this problem.

ICA is a higher-order statistical method that learns from sampled data how to separate the true source signals from their mixture. ICA is successful in demixing data up to an arbitrary scale (amplitude) and permutation in the source sequence. The ICA method studied in this thesis is linear-ICA. Linear-ICA assumes the unknown sources are (i) mixed linearly [17] , (ii) stationary [54], (iii) statistically independent [54], and (iv) non-Gaussian distributed [54]. However, most implementations of this method are inherently sensitive to outliers, (*i.e.*, extreme values that do not comply with the *probability density function* (PDF) of the signal) because they are based on *higher-order statistics* (HOS) [54]. HOS are outlier sensitive because errors in their calculation are increased by a power greater than 2; *e.g.*, by a power of 3 for skewness [54] and 4 for kurtosis [54].

Our interest is in the demixing of signals found in nature, especially biomedical signals such as *electrocardiograms* (ECG) [58], *electroencephalograms* (EEG) [58], and those signals with power-law (long-tailed) distributions, *e.g.*, ion-channel kinetics [46], for developing a health monitoring

device. However, life sustaining biomedical signal processing demands a guarantee that the results produced are accurate and precise. Algorithms and their implementation must be robust to interference, including numerical errors.

Unfortunately, contamination of biomedical recordings by outliers is an unavoidable aspect of signal processing. For example, Hampel *et al.* [29] provided a real-world situation related to electroencephalographic data obtained by a fully automatic recorder. The recorder equipment was working properly, and the histogram was adequate, except for some seemingly unimportant jitter of the plotter in the tails in the PDF. Yet, the third and fourth statistical moments were far too large. A search revealed that there was a spike of about two dozen out of a quarter million data points when the equipment was turned on, and these few points caused the high moments and the jitter in the plot. Since outliers are so devastating in estimating higher-order statistics, the impact of outliers on the signal separation performance of an ICA algorithm is an important characteristic in assessing the algorithm's utility.

Unfortunately, the sensitivity of ICA algorithms to outliers has not been studied in depth, outside a few places [36], [49], [42]. Robust methods have been developed for dealing with outliers in conventional signal processing, but those were for situations with Gaussian data, and are not directly applicable to situations with HOS and long-tailed distributions without careful study [6]. Thus, the primary objective of this thesis is to measure the outlier sensitivity of five well-known ICA algorithms (FastICA, Extended-Infomax, JADE, RADICAL and Beta(β)-divergence), and report the results. The novelty in this thesis is the development of an unbiased optimization-landscape environment for assessing outlier sensitivity, as well as the optimum angle of rotation error and the contrast function difference as new measures for assessing the outlier sensitivity of ICA algorithms. To grasp the exact issues considered by this thesis, Sec. 1.2 explores the BSS and ICA, as well as outliers with some examples.

1.1 Signal Definitions

1.1.1 Analog Signal

An analog signal is defined as a continuous, infinite resolution function over time or space [39].

1.1.2 Discrete Signal

A discrete signal is defined as an analog signal whose amplitude is sampled at discrete, equally spaced points in time or space [39].

1.1.3 Quantized Signal

A quantized signal is defined as an analog signal whose amplitude is rounded to a finite resolution of K bits. This resolution is determined such that the quantization step Q; *i.e.*, the *full-scale* (FS) value of the signal divided by 2^K, equals the system noise level [39].

1.1.4 Digital Signal

A digital signal is defined as a sampled and quantized analog signal [39].

1.1.5 Nyquist Sampling Theorem

The sampling frequency, f_s, must be at least greater than double the cut-off frequency, f_c, ($f_s > 2f_c$). The cut-off frequency is the frequency at which 50% (3 dB) of the signal power is lost relative to the DC value [39].

1.1.6 Dynamic Range

The dynamic range of a digital signal is the ratio of the loudest (FS) to the quietest (equivalent to one quantization step, Q) signal ($6.02 \times K$ dB) [39].

1.1.7 Signal Stationarity

A signal is defined as *weak (wide) sense stationary* (WSS) if its statistical moments are unchanging over time or space for orders less than and equal to two. A signal defined as *strong (strict) sense stationary* (SSS) if its statistical moments are unchanging over time or space for all orders.

1.1.8 Linear Function

A linear function is one which satisfies the additive ($f(x+y) = f(x) + f(y)$) and homogeneity ($f(ax) = af(x)$) properties [71].

1.1.9 Random Variable

A random variable describes the possible outcomes from an experiment. Two types of random variables exist: discrete and continuous. A discrete random variable describes the finite number of outcomes from an experiment. Associated with a discrete random variable is a *probability mass function* (PMF). A continuous random variable describes the range of outcomes from an experiment. Associated with a continuous random variable is a *probability density function* (PDF). This thesis only deals with modelling the unknown sources as continuous random variables as they are a good model for the stochastic processes we are interested in separating.

1.1.10 Deterministic and Stochastic Signals

Signals can be deterministic or random (stochastic). A deterministic signal can be determined uniquely by a mathematical expression (*e.g.*, $\sin(x)$), or a rule (*e.g.*, a hysteresis loop), or a table lookup. A stochastic signal (process) is best described by a random variable with a specific PDF. A stochastic signal can evolve either in time (time series) or in space. Examples of stochastic signals are speech, the ECG, and a synthetic aperture radar signal [54], [51].

1.1.10.1 White Noise

White noise is a stochastic signal that has a flat power spectrum (equal power at all frequencies), and has zero auto-correlation [54]. There is no specific density function associated with white noise, but Gaussian distributed white noise is a common model of many additive processes.

1.1.10.2 Coloured Noise

Coloured noise is a stochastic process that has a spectral density proportional to $1/f^b$, where f is frequency and b is the order of decay. White noise has $b = 0$, pink noise (often used to model ECG signals) has $b = 1$, and brown noise has $b = 2$. Noise in nature can have a functional b [39].

1.1.11 Statistical Independence

Statistical independence means that the occurrence of one event does not change the probability of another event from occurring. If two random variables are independent, knowing something

about one does not give any insight on the other. For example, the result of flipping two coins is statistically independent, as flipping one coin should not predict what the result of flipping the other coin will be. A mathematical definition of statistical independence is given in Sec. 2.2.2.

1.1.12 Whitening

Whitening or sphereing is the process of making a zero-mean random vector have elements that are uncorrelated and have unit variance [36]. Whitening is essentially decorrelation followed by a scaling [36]. Section 2.4 presents an in-depth discussion of whitening.

1.1.13 Structural and Probabilistic Outliers

There are a number of definitions of outliers. One of them refers to outliers with respect to a given time series. A temporal signal with a value either outside a known range, or a known trend, belong to a class called *structural outliers*. For example, a deviation in a linear regression model, or spike in a time series. This thesis does not consider such outliers because they can be deleted and removed from the time series by reducing them to either the maximum value, or an expected value. Signals considered in this thesis have non-Gaussian distributions, usually power-law with long tails. For such signals, a small departure may cause profound changes in the outcome, due to the use of HOS. Consequentially, this thesis considers outliers defined in terms of a probabilistic model such as a PDF [51].

1.1.14 Notation

1.1.14.1 Scalar

A scalar is denoted by a non-bold, italicized, lower- or upper-case character; *e.g.*, a, b, c, A, B, C.

1.1.14.2 Vector

A vector is denoted by a bold, non-italicized, lower-case character; *e.g.*, $\mathbf{a}, \mathbf{b}, \mathbf{c}$.

1.1.14.3 Matrix

A matrix is denoted by a bold, non-italicized, upper-case character; *e.g.*, $\mathbf{A}, \mathbf{B}, \mathbf{C}$.

1.1.14.4 Function of a Scalar, Vector or Matrix

The function of a scalar, vector or matrix is denoted by a non-bold, italicized, lower-case character, followed by parentheses surrounding the scalar, vector or matrix; *e.g.*, $a(a), b(\mathbf{b}), c(\mathbf{C})$.

1.1.14.5 Multiparameter Function

A multiparameter function is denoted by a non-bold, italicized, lower-case character, followed by parentheses surrounding the scalar, vector and/or matrix parameters; *e.g.*, $a(a, \mathbf{b}, \mathbf{C})$

1.1.14.6 Analog Signal

A 1-dimensional analog signal is denoted as $x(t)$, where x denotes a continuous function, evolving over time t. A column vector of M 1-dimensional analog signals is denoted as $\mathbf{x}(t)$. The mth analog signal is denoted as $x_m(t)$, where integer value m ranges from 1 to M.

1.1.14.7 Digital Signal

A 1-dimensional digital signal is denoted as $x(n)$, where n is the time index, t_n, at which the analog signal was sampled. The integer value n ranges from 1 to N. A column vector of M 1-dimensional digital signals is denoted as $\mathbf{x}(n)$. The mth digital signal is denoted as $x_m(n)$, where integer value m ranges from 1 to M.

1.1.14.8 Random Variable

A single continuous random variable is denoted as $\bar{x}$. A column vector of M random variables is denoted as $\bar{\mathbf{x}}$. The mth random variable from a column vector of random variables is denoted as $\bar{x}_m$. The expectation of a random variable is denoted as $\mathcal{E}(\bullet)$.

This thesis only considers analog and digital signals. The analog signals are assumed to be strong-sense stationary (SSS) and evolve over time. The digital signals are assumed to originate from analog signals that are SSS, evolve over time, sampled according to the Nyquist sampling theorem, and quantized to a resolution of at least 16-bits. Although digitizing an analog signal introduces errors, the impact is not considered to be a significant factor on the results of this thesis.

1.2 Problem Specification

Consider the cocktail party problem of Fig. 1.1 as an example of the BSS problem. The figure shows three people speaking at the same time, and their combined voices being recorded by three microphones. In this situation, humans listening to any of the recordings have an innate ability to focus on one person's voice. However, the objective is to use a computer to demix the recorded signals without any knowledge of the original source signals, the information contained within, or their mixing; *i.e.*, everything left of the dashed line in Fig. 1.1. In the mid 1990s, ICA was shown, with a few weak assumptions, to perform this type of task with excellent results [36]. For example, in 1995 Bell and Sejnowksi [8] published a paper on the successful separation of mixtures containing up to ten speakers.

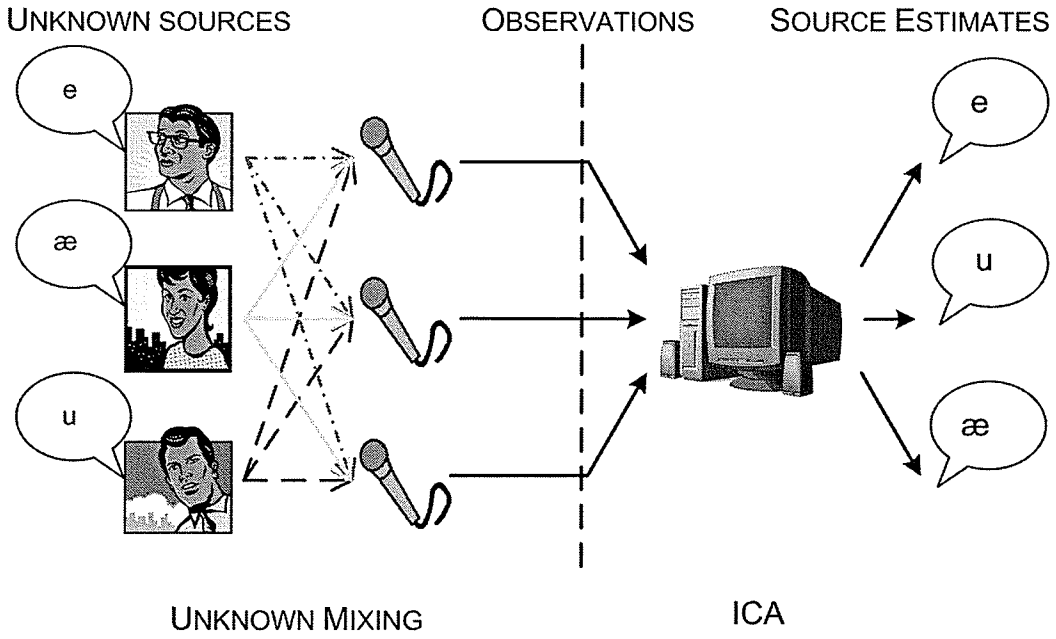


Fig. 1.1 Cocktail party problem.

Now, to solve the BSS problem, ICA assumes that the original sources are statistically independent, stationary (SSS) and contain at most one Gaussian signal (see Sec. 2.2.2.2 for reasoning). The focus in this thesis is linear-ICA which also assumes that the mixing is linear. Figure 1.2 shows a simulated cocktail party problem as solved by the FastICA linear-ICA algorithm (Figures 1.2 to

1.11 are grouped at the end of the chapter to facilitate comparison). Here, three stationary analog source signals, denoted by the vector $\mathbf{s}(t)$, are assumed to be linearly mixed by a mixing matrix, $\mathbf{A}$, to produce the digital observation vector, $\mathbf{x}(n)$, where everything to the left of the dashed line is unknown. These observations are fed into the linear-ICA algorithm to produce an estimate of the original sources, $\mathbf{y}(n)$. The demixing matrix, $\mathbf{W}$, is estimated by means of a contrast function and an optimization technique. For example, the contrast function is based on the source separation principle of minimizing the mutual information between the joint density of the source estimate and the product of its marginals by a Kullback-Leibler divergence [36], and the optimization technique is gradient descent. Thus, a contrast function, $C(\mathbf{x}(n), \mathbf{W})$, is selected such that the components of $\mathbf{y}(n)$ become statistically independent during the minimization of its expectation. Figure 1.2 shows the successful separation of the voices, however only up to an arbitrary scale and permutation. The primary reasons this thesis has selected to study ICA for solving the BSS problem is because (i) it has weak assumptions about the sources and their mixing, (ii) it can handle broadband signals as it does no frequency filtering, and (iii) it is currently the most successful method for solving the BSS in a growing number of applications [36].

It is important to note that this thesis explains the relationships and theorems of ICA with a slant towards the methodology adopted by the well-known ICA researcher, Hyvärinen [36]. Cichocki and Amari [17], two other prominent ICA researchers, have a broader view of ICA, and explain the concepts in a more generalized way that (unfortunately) is unnecessarily complicated for this thesis.

To formalize the problem specification, it is best to evaluate a detailed example of the BSS problem, ICA, and the impact of an outlier on the separation performance of an ICA algorithm. To begin, Fig.1.3 is a plot of two recorded signals, $x_1(n)$ and $x_2(n)$, representing two analog signals, $x_1(t)$ and $x_2(t)$, which we hypothesize is a mixture of two unknown analog signals, $s_1(t)$ and $s_2(t)$. Figure 1.4 shows the linear-ICA estimate (using the JADE algorithm), $y_1(n)$ and $y_2(n)$, of the original sources, $s_1(t)$ and $s_2(t)$, while Fig. 1.5 shows a time series plot of two digital signals $s_1(n)$ and $s_2(n)$ representing the original analog sources. In this thesis, the original source signals are known, and thus, the exact difference between the original and demixed sources can be measured. For example,

the non-blind metric known as the *Amari separation performance index*, API, (see Sec. 3.2), is commonly used to measure the separation performance. In this example, the normalized API is calculated to be near 0, and indicates an excellent separation.

To understand our approach to the thesis, it is necessary to visualize how ICA is able to estimate the sources through scatter and contrast function plots. Figure 1.6a, shows a scatter plot of the observed signals $x_1(n)$ and $x_2(n)$ after whitening; *i.e.*, they have been processed to have zero mean and unit variance. Figure 1.6b, shows the scatter plot of the source estimate signals $y_1(n)$ and $y_2(n)$. The only change between the two scatter plots is a rotation of the points. Thus, the optimization of a linear-ICA contrast function essentially separates the signals by finding a rotation matrix, **W**, via the optimization of a contrast function.

Figure 1.7 shows a plot of the contrast function of the JADE ICA algorithm used for this example (see Sec. 2.3.3 for details). The objective is to find the minimum of the contrast function in order to separate the signals. In this case selecting a 45 degree rotation matrix, **W**, should minimize the contrast function. Note, the actual implementation used in the example has a minimum at 43 degrees.

Now, to see the effect of an outlier on the situation, we begin with the same recorded signals as before, but introduce an outlier with an amplitude of 5; *i.e.*, an outlier 5 standard deviations away from the mean (Fig. 1.8).

Figure 1.9 shows the new source estimate. It is clear that the separation is worse. The smoothness of $s_1(t)$ is lost, as well, the maxima of $s_1(t)$ are missed. Clearly, the separation performance of the JADE algorithm is sensitive to this outlier (the normalized API is 0.31). Thus the question, how sensitive is the technique and should another algorithm be used with a lesser sensitivity to outliers? Another question is why not remove the outlier by an outlier detection method? This can be done, but this is difficult because as stated earlier, we are interested in signals with long-tailed distributions, for which points that are located 5 or even 10 times the standard deviation from the mean of the distribution are still a part of the distribution. Furthermore, most outlier detection methods require a-priori knowledge of the distribution to be cleaned of outliers, which does not apply to ICA

[6].

Figure 1.10b shows the scatter plot of the new source estimates $y_1(n)$ and $y_2(n)$. Compared to Fig. 1.6b, it shows that the rotation is affected by the outlier. This is because the outlier affected the minimum of the contrast function which determines the rotation angle for separation (Fig. 1.11). A 20 degree change of the minimum occurs due to the outlier (Fig. 1.11). This example shows that the HOS used in the JADE contrast function were estimated incorrectly. Consequently, the question of how the outlier affected the contrast function is explored in the thesis.

Thus, the problem of the thesis is to study the outlier sensitivity of ICA algorithms through the impact of outliers on the rotation angle, and other shape changes to the contrast function.

1.3 Thesis Statement and Objectives

This thesis studies the impact of an outlier on the separation performance of an ICA algorithm through the measurement of the changes in the identified rotation angle, and deviations in the algorithms contrast function in an unbiased optimization landscape. The objectives of this thesis are to:

(a) Measure the outlier sensitivity of five well-known ICA algorithms (FastICA, Extended-Infomax, JADE, RADICAL and β-divergence);

(b) Rank the outlier sensitivity of these ICA algorithms; and

(c) Suggest how to reduce the outlier sensitivity of ICA algorithms.

The algorithms named in Objective (a) were selected due to (i) their popularity in research, (ii) the distinct separation principles they follow to solve the BSS problem, (iii) their claimed outlier robustness in literature, and (iv) they are linear-ICA algorithms, the class of algorithms to which the thesis is limited (non-linear and other types of ICA algorithms are discussed in Sec. 2.2.2) . This thesis only studies linear-ICA algorithms because they have been well-researched, and most of them have code suitable to conduct simulations.

The outlier sensitivity of these algorithms is measured by

(i) The Amari separation performance index,

(ii) The optimum angle of rotation error, and

(iii) The contrast function difference.

1.4 Organization of the Thesis

This thesis is organized into six chapters involving ICA, outlier robustness measures, and an outlier sensitivity assessment of five ICA algorithms. Chapter 1 introduces the BSS problem, and the motivation for investigating the outlier sensitivity of ICA algorithms. Chapter 2 provides an in-depth background on BSS, ICA, and the five ICA algorithms (FastICA, Extended-Infomax, JADE, RADICAL and β-divergence) studied. It also provides an insight to how this thesis has approached measuring the outlier sensitivity of these algorithms. Chapter 3 covers outliers and the three approaches (Amari separation performance, rotation sensitivity and contrast function difference) used to measure the outlier-sensitivity of ICA algorithms. Chapter 4 contains information on the setup of experiments to assess the outlier sensitivity. Chapter 5 presents and discusses the results of these experiments. Finally, Chapter 6 summarizes the conclusions garnered from this thesis, and presents recommendations for future work.

A reoccurring feature of the thesis is a visual layout of each chapter, and is intended to provide the reader an alternate to the table of contents.

1.5 Thesis Contributions

The major contributions of this thesis are

(a) A review of ICA from the perspective of outliers;

(b) Development of the contrast function and unbiased optimization landscape as the key to the analysis of the outlier robustness and potential separation performance of ICA algorithms;

(c) Development and use of the optimum angle of rotation error outlier sensitivity metric;

(d) Development and use of the contrast function difference outlier sensitivity metric;

(e) Development of an outlier dataset for ICA algorithm benchmarking;

(f) Benchmarking the ICA algorithm separation performance with and without outliers in a unbiased optimization landscape; and

(g) Implementing the β-divergence algorithm and making it available to the research community.

1.6 Summary

In addition to the problem specification and organizational structure of the thesis, this chapter provides a basis for the approach to outliers and ICA. This study sees ICA as a contrast function, based on some separation principle, and an optimization technique. Second, it hypothesizes that the outlier sensitivity is best determined through both the rotation sensitivity of a contrast function (due to its close relationship to the separation performance of an ICA) and simplicity. This introduction is intended to provide the reason why ICA is explained via separation principles and contrast functions in Ch. 2.

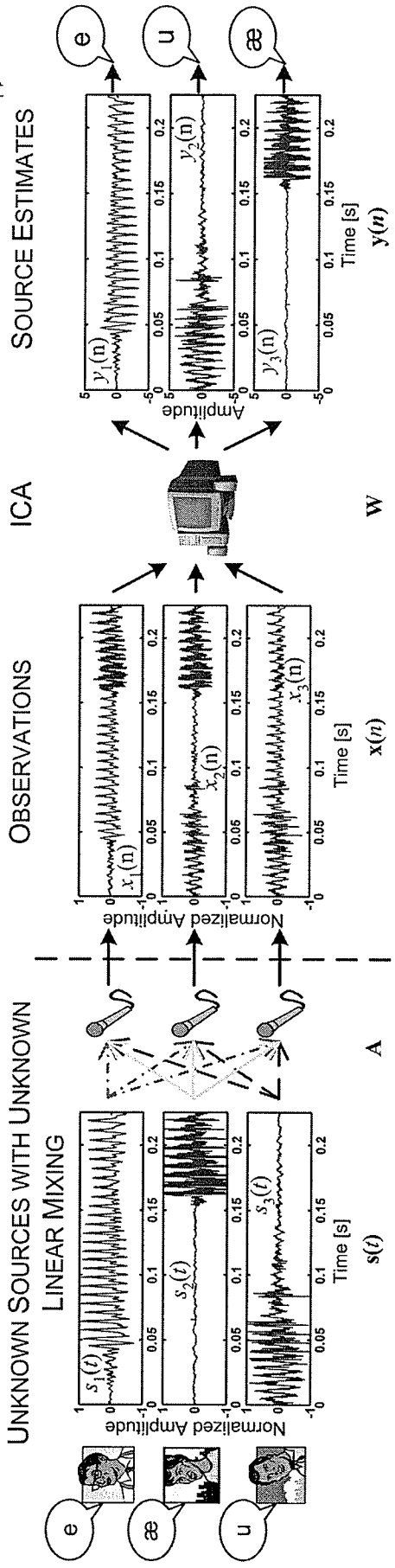


Fig. 1.2 Cocktail party problem and linear-ICA.

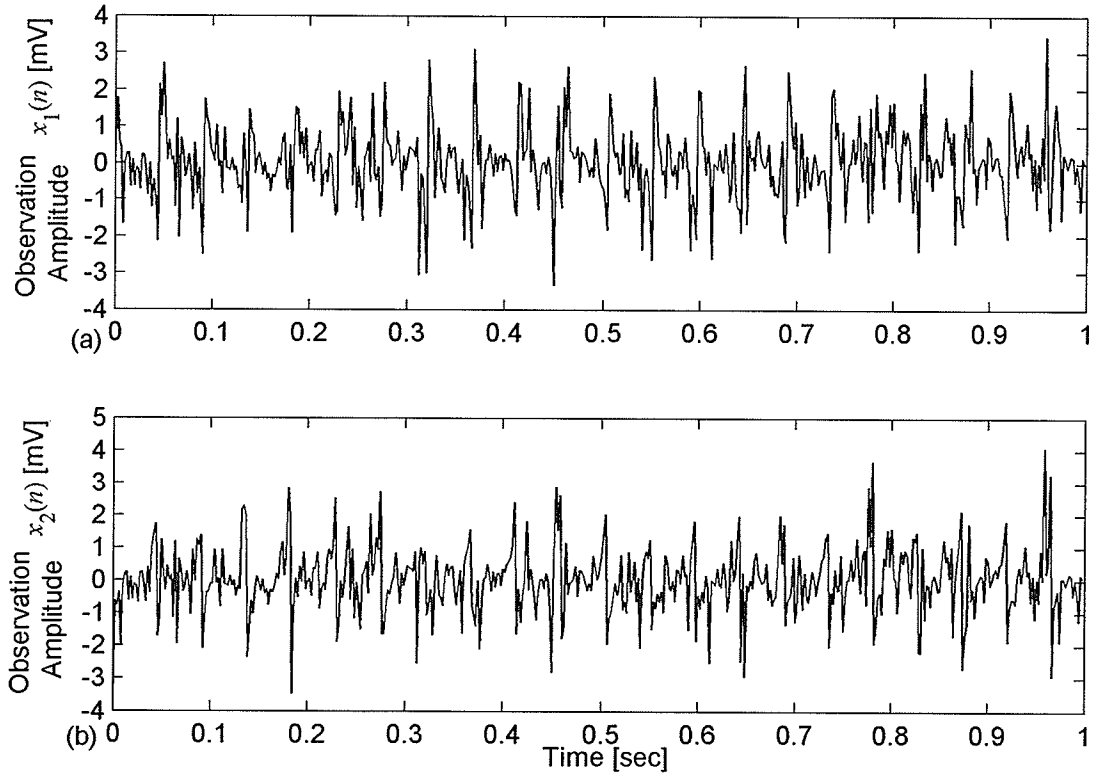


Fig. 1.3 Amplitude plots of two recorded signals (a) $x_1(n)$, and (b) $x_2(n)$.

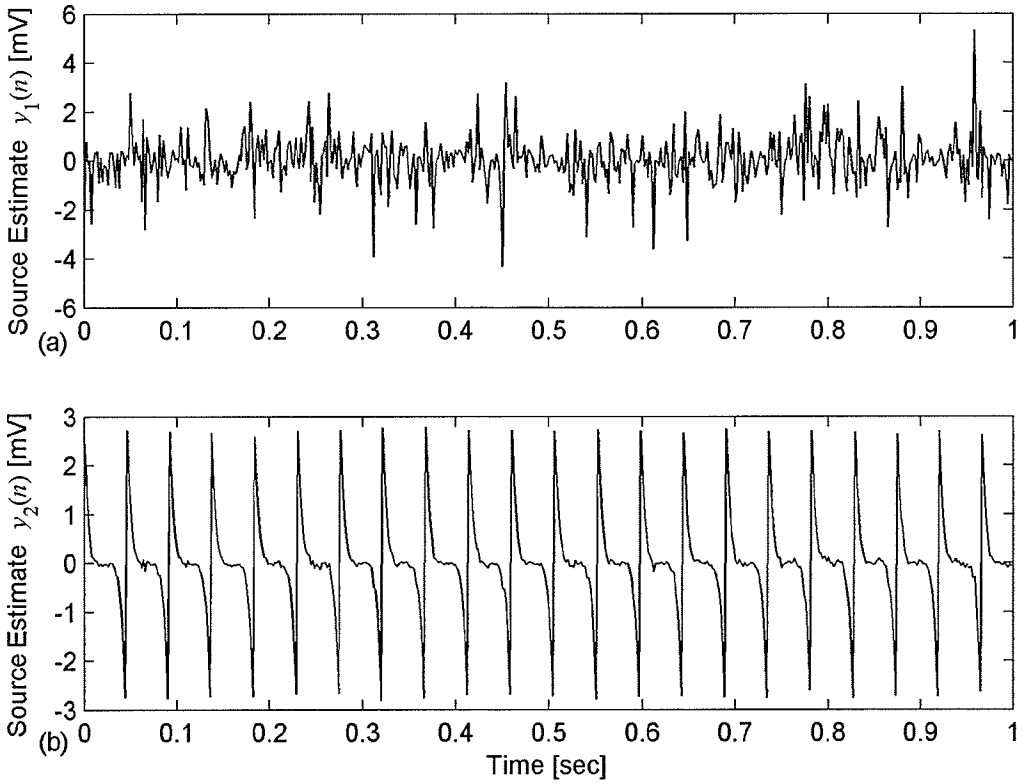


Fig. 1.4 Source estimates (a) $y_1(n)$ and (b) $y_2(n)$ of two analog signals $s_1(t)$ and $s_2(t)$.

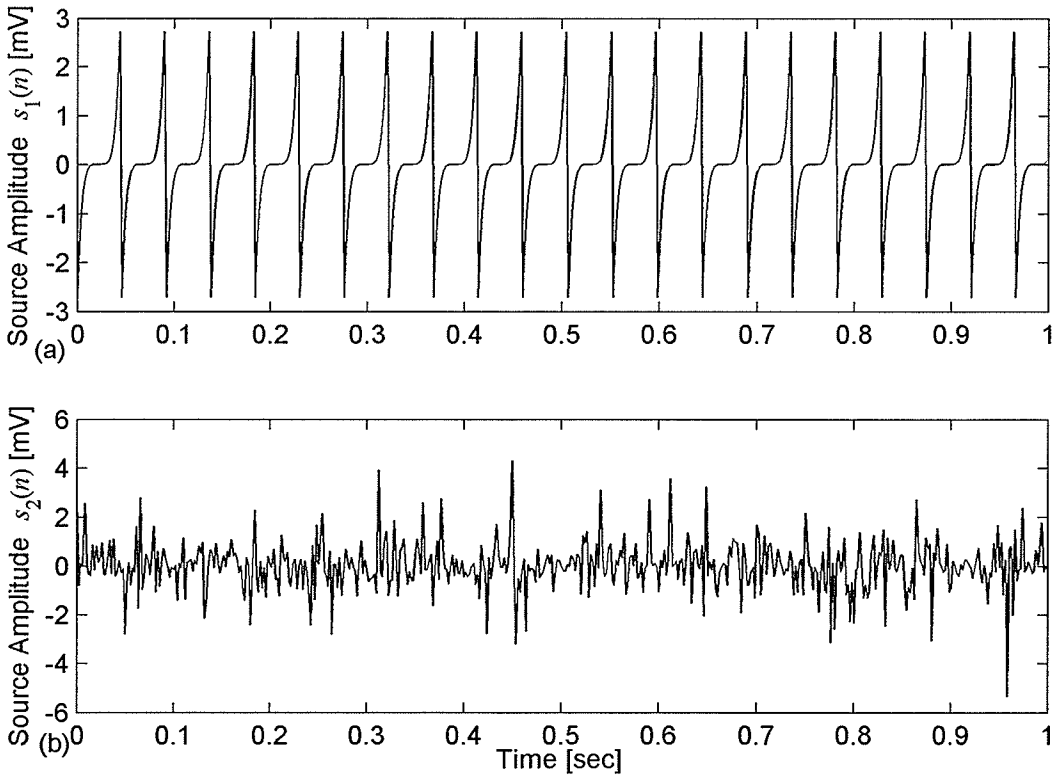


Fig. 1.5 Amplitude plots of the digitized source signals (a) $s_1(n)$, and (b) $s_2(n)$. Notice the source estimates shown in Fig. 1.4 permuted the sequence of the source signals. In addition, the estimates were scaled -1. The permutation and scaling of the sources constitute an inherent limitation to ICA.

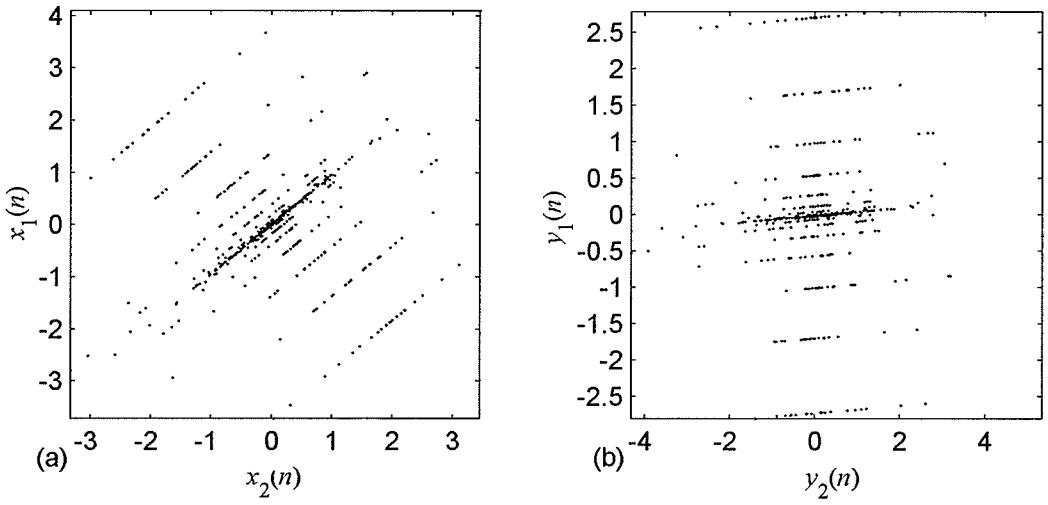


Fig. 1.6 Scatter plot of (a) two 1-dimensional digital signal mixtures, and (b) the resulting source estimates. The stripped pattern is due to the time correlated signal having a larger number of samples at certain values. Notice the source estimates are essentially the observations rotated by approximately 45 degrees.

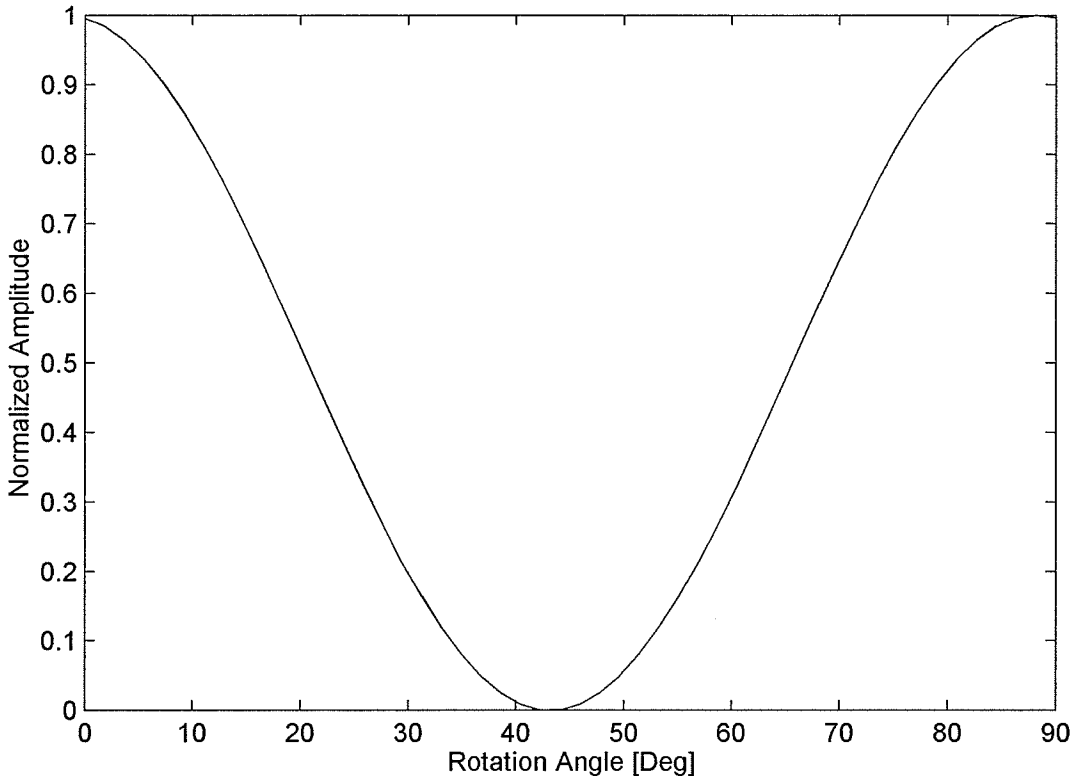


Fig. 1.7 Contrast function plot of the JADE ICA algorithm. The algorithm produces a minimum near 43 degrees, and not the expected 45 degrees.

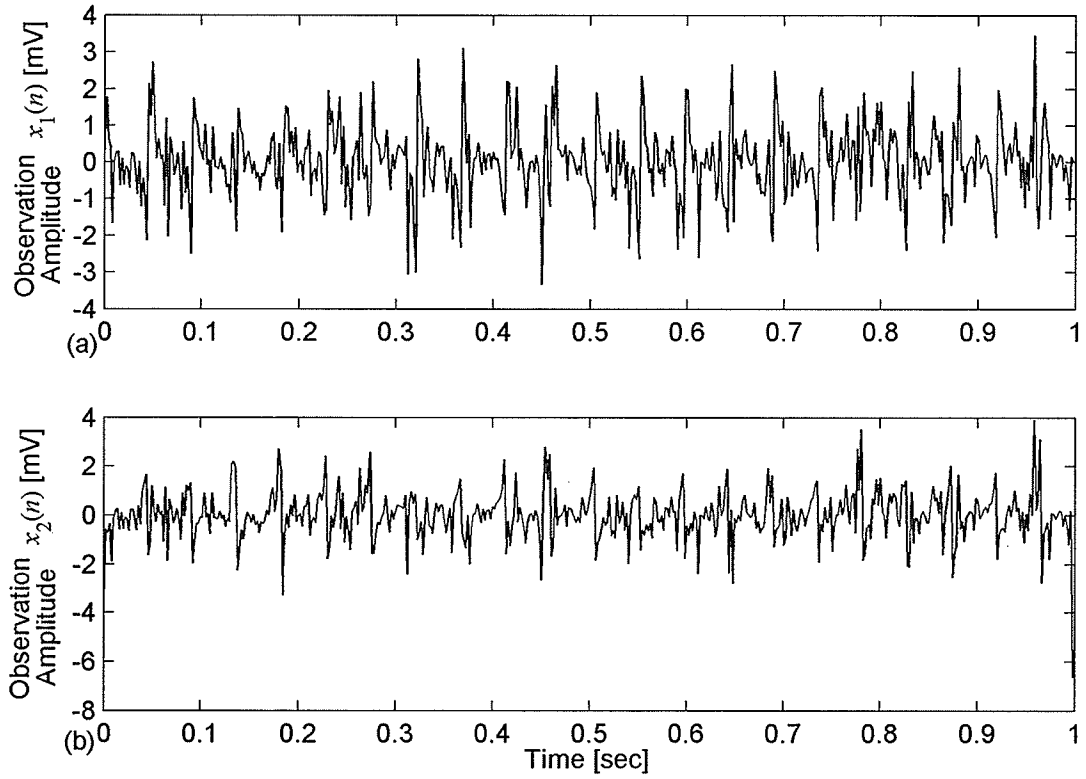


Fig. 1.8 Amplitude plots of the two recorded signals (a) $x_1(n)$, and (b) $x_2(n)$. We hypothesize these signals are linear mixtures of two unknown analog signals, $s_1(t)$ and $s_2(t)$. In contrast to Fig. 1.3, $x_2(n)$ is contaminated by an outlier at time 1.

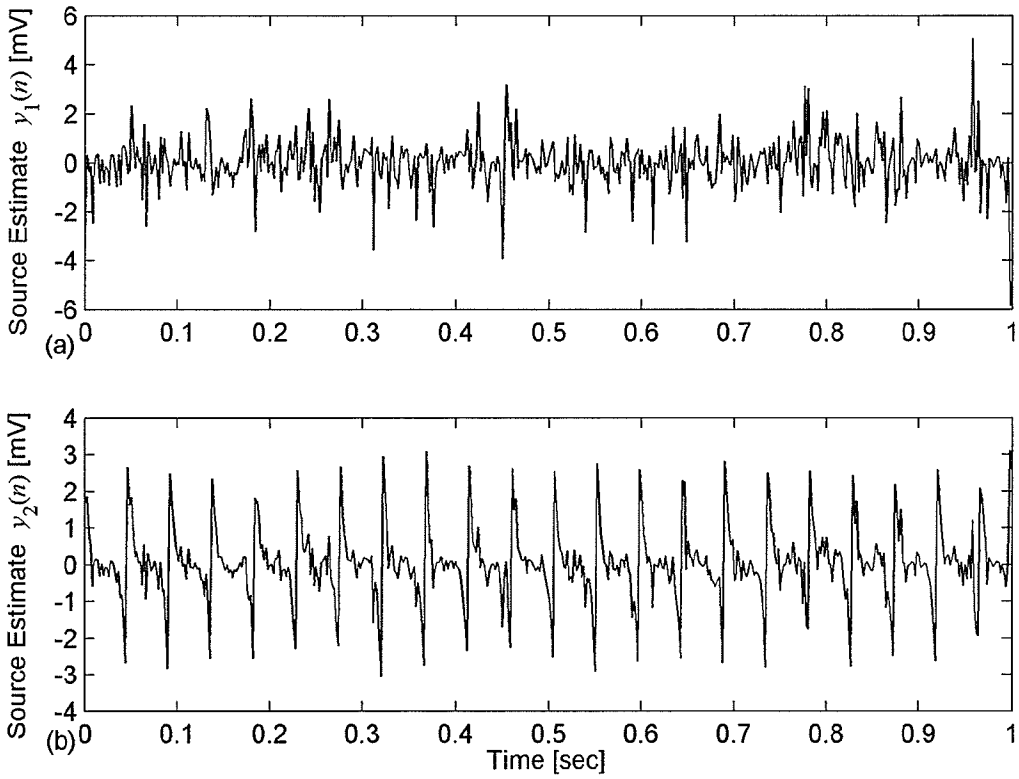


Fig. 1.9 Source estimates (a) $y_1(n)$ and (b) $y_2(n)$ of two analog signals $s_1(t)$ and $s_2(t)$. In contrast to Fig.1.4, these source estimates are based on observations contaminated by an outlier.

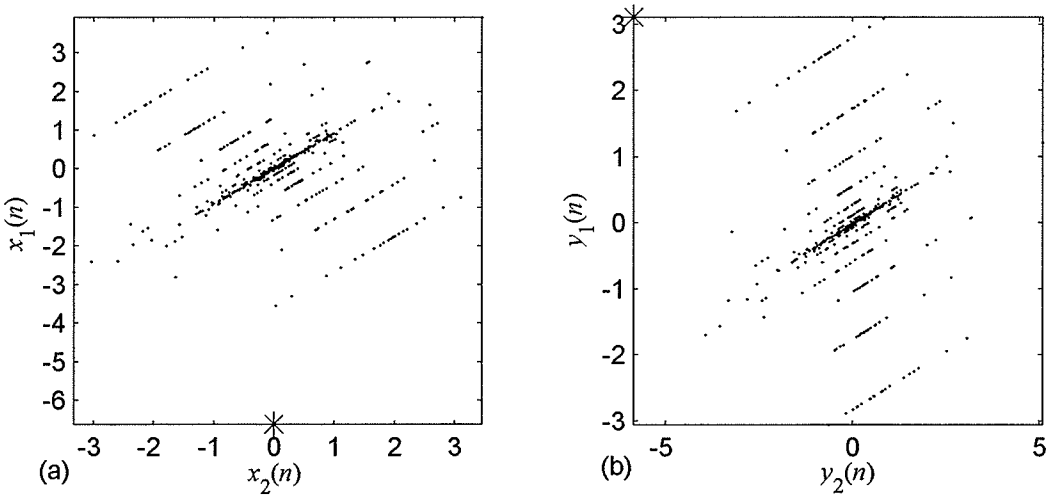


Fig. 1.10 Scatter plot of (a) two 1-dimensional digital signal mixtures, and (b) the resulting source estimates. In contrast to Fig.1.6, the observed mixtures are contaminated by an outlier denoted by the asterisk, as well the source estimates are based on the contaminated mixtures.

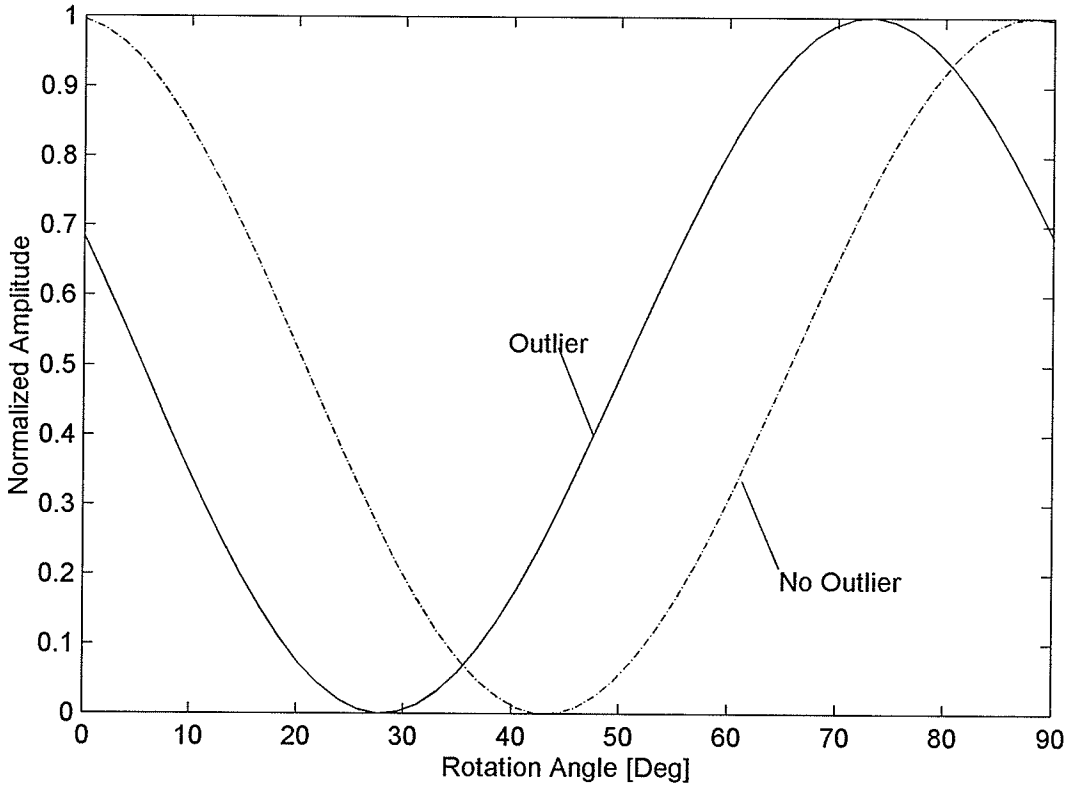


Fig. 1.11 Contrast overlay.

Chapter II

Background on ICA Sensitivity to Outliers

There are several good books on *independent component analysis* (ICA) [36], [17], [43], [60]. These books provide overviews of the origins, principles and applications of ICA for *blind source separation* (BSS), algorithm derivations and proofs, and collections of seminal papers on ICA. However, none is dedicated to the sensitivity of ICA to outliers. This chapter is a contribution by reviewing ICA from the perspective of outliers, as defined in Sec. 1.1.13. The chapter begins with a description of the BSS problem, discusses the various methods used to solve it, and then gives reasoning for why ICA is a promising solution. Then the chapter moves on to explain the principles of ICA, various ICA algorithms, and finally the preprocessing of data for these algorithms. Throughout, the potential effects of outliers are discussed, and research questions are highlighted. Figure 2.1 is a visual guide to the chapter. It is assumed that the reader has sufficient knowledge in probability theory and random variables. For a comprehensive review of probability theory and random variables, please consult a book by Papoulis [54]. In summary, this chapter gives a basis for the thesis objectives.

2.1 What is BSS?

Blind source separation is defined as the problem of demixing a combination of signals or "sources" based on observations of those mixtures only. The keyword "blind" stems from the fact that no *a priori* information is available on the sources or their mixing [17]. Linear BSS assumes the sources were mixed linearly. Solving the BSS problem is seen by researchers and scientists as a necessary preprocessing step to obtain uncontaminated data for analysis. Examples of the BSS problem are discussed in the next section.

Related to the BSS problem is blind beam forming. One of the objectives of beam forming is to

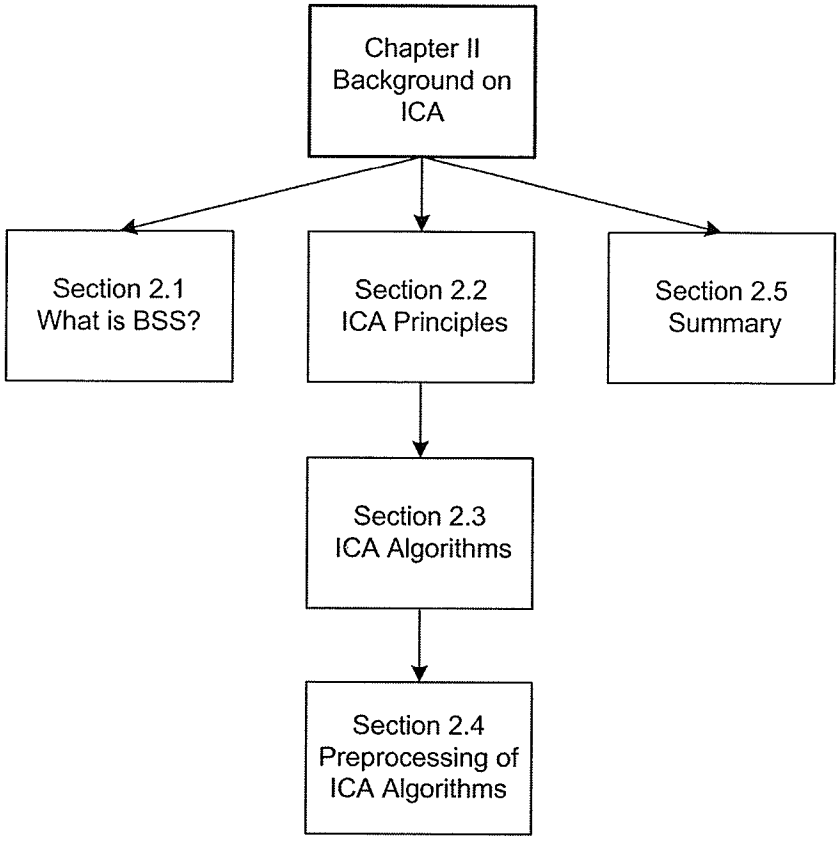


Fig. 2.1 Layout of Ch. II.

receive a signal with an array of sensors from a desired direction, and enhance the received signal from the desired spatial direction while reducing the signals from other directions. In blind beam forming, the locations of the sensor array are unknown. Applications of blind beam forming are found in radar and sonar processing where source location and *direction-of-arrival* (DOA) are of importance [16].

2.1.1 BSS Examples

The BSS problem arises in areas such as biomedical signal processing, cognitive informatics and telecom signal processing.

In biomedical signal processing, a bio-potential sensor affixed to the skin is not selective at all.

The sensor picks up all bio-potential signals within its vicinity; *e.g.* heart, muscle, and brain. Consider the measurement of a fetal heart signal [72]. The bio-potential sensors affixed to a mother's skin measure the superposition of the fetal and parent ECGs. In addition, these heart signals are further attenuated by flesh and bone, and contaminated by stray muscle, *electromyogram* (EMG), signals. Thus, solving this BSS problem is a necessary preprocessing step in order to obtain uncontaminated data suitable for further fetal heart-monitoring analysis.

In the area of cognitive informatics, the *electroencephalogram* (EEG) is a standard tool for investigating brain activity. Of interest is (i) the location of various activities within the brain, and (ii) the dynamics of those functional regions. Unfortunately, the recorded temporal EEG signal is not the best indicator of the dynamics, because it does not originate from the brain alone, but is a mixture of the heart, and other signals from the muscular system. Consequently, to remove these unwanted signals, demixing of the recorded EEG signal should be done in order to perform a directed analysis [25].

Finally, in telecom signal processing, an antenna receives all emitting sources within its receptive field. Blind *multiuser detection* MUD in a *direct-sequence code division multiple access* (DS-CDMA) communication system is an application where BSS for interference suppression is of use [59]. Other areas where the BSS problem arises are in compression [21], the denoising of signals (including images) [45], and seismic signal processing [36].

From the examples listed, the ECG and EEG signals are considered to be (i) stationary over short periods of time, (ii) non-Gaussian distributed, and (iii) have the potential to be contaminated by outliers (discussed in detail in Ch. 3) [26]. Clearly, an outlier-robust ICA method could solve BSS problems involving these signals. For completeness, the next section describes other approaches to deal with the BSS problem.

2.1.2 Solutions to the BSS Problem

There are four primary approaches to solving the BSS problem [17], [30]. The distinction between the approaches is that each makes a different assumption about the unknown sources. The first assumes the sources are stationary, statistically independent and are without temporal structure.

ICA is based on this premise. However, these methods do not allow for more than one Gaussian source (see Sec. 2.2.2.2 for more detail). The second approach exploits the temporal structures of the signal using *second-order statistics* (SOS). However, this approach does not allow for the separation of sources with identical power spectra or independent and *identically distributed sources* (i.i.d). The third approach utilizes the non-stationarity properties of the sources along with SOS. However, these methods do not allow for the separation of sources with identical non-stationary structures. Finally, the fourth approach attempts to exploit some space-time-frequency characteristic of the sources. Although each approach is distinct, they overlap greatly, as ICA can be extended into each of the other approaches by making additional assumptions on the mixing of the sources. An in-depth discussion of these approaches and their relationship is found in [17]. Out of the four approaches, ICA is the most well studied, has algorithm implementations available for download on the Web, and has been applied successfully in many applications [36].

There is a method called *projection pursuit* (PP) which can also solve the BSS problem. Projection pursuit is a method from statistics for finding projections of multidimensional data that are "interesting" visually [36]. This objective sometimes leads to the separation of a mixture of sources. However, PP is primarily a method for visualizing data while ICA is primarily for finding independent components. For a discussion on this topic, see Ch. 8 in [36].

In summary, the BSS problem has been introduced, a method called independent component analysis has been selected for study, and thus, the details of ICA can now be introduced.

2.2 ICA Principles

This section describes the principles of ICA and associated algorithms. To cover this material, the section is broken up into five subsections beginning with (i) the origins of ICA, (ii) the mathematical definition of ICA, (iii) ICA separation principles, (iv) entropy and PDF estimation techniques and finally, (v) optimization techniques.

For an expanded coverage of ICA there are a few important textbooks, journals and newsgroups to reference. First and foremost, the textbook by Hyvärinen, Karhunen, and Oja [36] contains an

overview of the origins, principles and applications of ICA for BSS. The chapters range from rudimentary statistics to explaining the implementation of various ICA algorithms. However, the book contains content and explanations revolving around the FastICA algorithm. A textbook by Cichocki and Amari [17] has a view of ICA from the perspective of adaptive blind signal processing. This book is heavy on derivations, and is too disjoint for a beginner. However, for those interested in proofs, it is an excellent resource. The other books on ICA include Lee [43] and Roberts [60]. However, these books are collections of papers originating from various journals. The primary journals in which ICA research is published are Neural Computation by MIT Press, Signal Processing by Elsevier, IEEE Transactions on Signal Processing, and IEEE Transactions on Neural Networks. The primary conference in which ICA research is presented are the International Symposium on Independent Component Analysis and Blind Signal Separation, and the *International Conference on Acoustics, Speech, and Signal Processing* (ICASSP). Finally, discussions regarding ICA are found on the ICA-list newsgroup run by the ICA researcher Cardoso [10].

2.2.1 Origins of ICA

The concept of ICA was first introduced by Jutten and Hérault for determining the motion encoding in muscle contraction [38]. Their feedback neural network sought to decorrelate nonlinearly (see Sec. 2.2.3.5) the observed muscle signals, in order to infer what information the brain was using to determine the angular position and velocity of a moving joint. In 1994, Comon [18] formalized the work of Jutten and Hérault (along with the blind separation of sources problem) by using a mathematically rigorous approach. This seminal paper defined the properties of the linear BSS problem, and proposed a solution using negentropy which is a non-Gaussianity measure approximated by higher-order statistics. However, Comon's solution to the BSS problem was processor intensive, and did not perform well outside a few examples. In 1995, Bell and Sejnowski [8], developed a solution to the linear BSS problem using the information-maximization (Infomax) principle. This method was demonstrated to be more successful at solving a larger number of linear BSS problems, and brought ICA to researchers outside of France. Since that point, numerous ICA algorithms have been developed for improved separation performance and reduced CPU usage [33], [36]. Today,

ICA research is advancing from solving the linear BSS problem to solving the BSS problem with convolved sources, non-linear mixing, and non-stationary sources. For a detailed account of the origins of ICA see [36], and [24] for an in-depth discussion. Now that the origins of ICA are known, the mathematical definition of ICA is given.

2.2.2 Definition of ICA

Independent component analysis is a statistical method for estimating a set of unknown sources based on sensor observations of a mixture (either linear, or non-linear, or time-varying) of the sources, and the knowledge that the original sources were statistically independent.

Consider analog signals $s_1(t)$ and $s_2(t)$ with PDFs $p(s_1(t))$ and $p(s_2(t))$. These signals are said to be statistically independent if and only if

$$p(s_1(t), s_2(t)) = p(s_1(t))p(s_2(t)) \tag{2.1}$$

where $p(s_1(t), s_2(t))$ is the joint PDF of $s_1(t)$ and $s_2(t)$.

Statistical independence is a stronger criterion than one of uncorrelatedness, as it requires the analysis of signal statistics greater than order 2. *Principal component analysis* (PCA) is a technique (discussed in Sec. 2.4) that seeks to decorrelate random variables, while ICA seeks to make random variables statistically independent, *i.e.*, decorrelate all statistical moments, to solve the blind source separation problem [36].

The given definition of ICA is quite broad, and additional assumptions about the sources and their mixing are usually made. Figure 2.2 is a breakdown of assumptions that most ICA algorithms have. The first level, common to all ICA algorithms, is the assumption of statistically independent sources. Next, an assumption on the stationarity of the sources is made. In the majority of ICA research, including this thesis, stationary sources are assumed. However, the assumption of non-stationary sources is becoming more prevalent in research [55], [14]. The third level is an assumption on the type of mixing that the sources have undergone. This is broken up into linear, convolutive and non-linear mixing. At the 5th International Conference on Independent Component Analysis and Blind Signal Separation in 2005, over 40 papers were dedicated to convolutive and

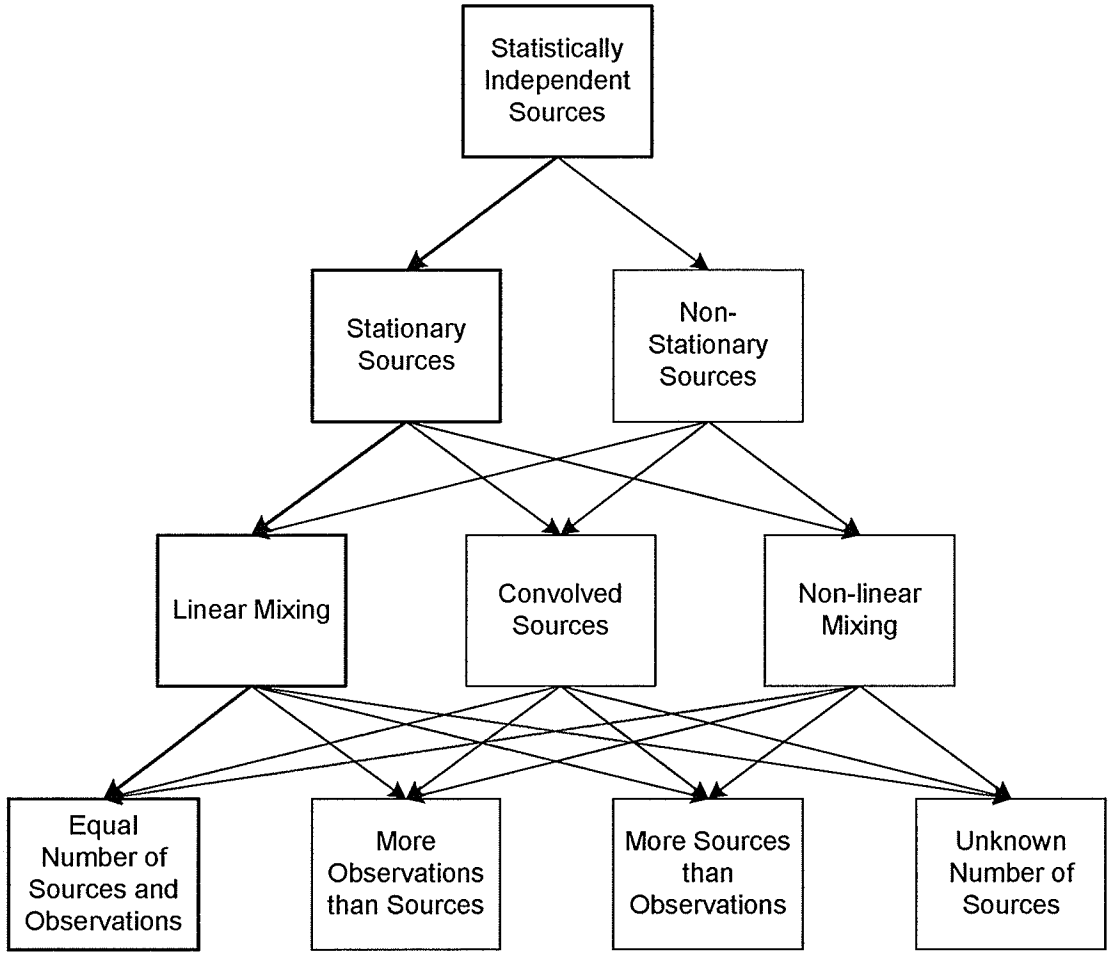


Fig. 2.2 ICA assumption hierarchy.

non-linear mixing in ICA [57]. Finally, the last assumption concerns the number of source signals. Usually, the number of sources is said to be equal to the number of observation signals. However, there are situations where it is more appropriate to assume more observation signals than source signals. This scenario is known as over-complete ICA [3]. As well, there are situations where is it best to assume there are more sources than observations. This is known as under-complete ICA [3]. Finally, the number of independent components may be unknown, and must be estimated [36].

This thesis focuses on linear-ICA which includes independent and stationary sources, linear mixing, and an equal number of source and observation signals. Our interest in the effect of outliers

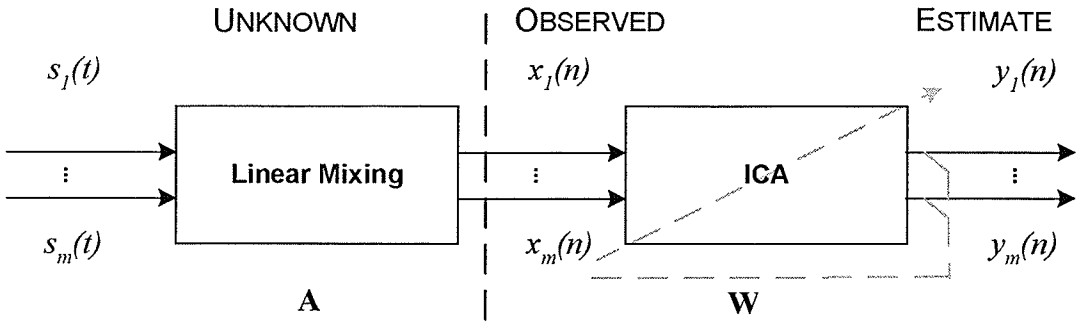


Fig. 2.3 Linear ICA model.

on this category of ICA is because (i) it is the most well studied, (ii) a number of algorithms have been developed for it, and (iii) the added complexity of the other categories distracts from the outlier-robustness objective.

2.2.2.1 Linear ICA Model

The linear ICA model assumes that (i) the sources are stationary, (ii) the sources are mutually independent, (iii) the number of sources equals the number of sensors, (iv) mixing is linear, (v) the mixing matrix is invertible, and (vi) at most one source, $s_i(n)$, has a Gaussian distribution. The reason that only one Gaussian source is allowed is discussed in Sec. 2.2.2.2. Figure 2.3 shows the linear ICA model, with M unknown linearly-mixed sources $\mathbf{s}(t)$, and M output observations, $\mathbf{x}(n)$. These observations are used to estimate adaptively the demixing matrix $\mathbf{W}$, such that the estimates $\mathbf{y}(n)$ are statistically independent, and approximate the unknown sources.

More specifically, let $\bar{\mathbf{s}}$ be a random vector of M unknown, mutually independent, non-Gaussian source signals, and $\mathbf{A}$ an unknown, square, non-singular mixing matrix of order M. These are mixed to produce a random vector, $\bar{\mathbf{x}}$, as

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

A linear ICA algorithm attempts to find the demixing matrix, $\mathbf{W}$, that produces, $\bar{\mathbf{y}}$, an estimate of the unknown sources $\bar{\mathbf{s}}$. Comon proved that under ideal circumstances, the sources can be estimated

only up to a scale and permutation of the sources as shown in Eq. 2.3 [18].

$$\bar{\mathbf{y}} = \mathbf{W}\bar{\mathbf{x}} = \mathbf{DP}\bar{\mathbf{s}} \tag{2.3}$$

where **D** is a diagonal scaling matrix and **P** is a permutation matrix [36]. The scale and permutation ambiguity is discussed in Sec. 2.2.2.3.

The demixing matrix **W** is estimated by means of a *contrast function* (CF) (*e.g.*, based on the source separation principle of minimizing the mutual information between the joint density of the source estimate and the product of its marginals by a Kullback-Leibler divergence [36]), and an optimization technique (*e.g.*, gradient descent) [53]. To determine **W**, a CF $C(\bar{\mathbf{x}}, \mathbf{W})$ is selected such that the components of $\bar{\mathbf{x}}$ become statistically independent at the minimization or maximization of its expectation. Alternatively, the determination of the demixing matrix is viewed as whitening (see Sec. 1.1.12) of the observed signals and then searching for a rotation matrix that produces the independent sources.

Consider how the linear mixture of two source signals with uniform distributions is solved by ICA. Figure 2.4 shows the scatter plots and histograms of two observed signals, $x_1(n)$ and $x_2(n)$, their whitened observations, $x_{1w}(n)$ and $x_{2w}(n)$, and finally their estimated source signals, $y_1(n)$ and $y_2(n)$. The first step in ICA is to whiten the observed signals. Whitening has been proved to reduce the complexity of the linear-BSS problem to finding an orthogonal transformation that determines the original sources [36]. In the scatter plot of the whitened signals, $x_{1w}(n)$ and $x_{2w}(n)$ (Fig.2.4b), it is clear that a rotation of the diamond would produce two uniformly distributed sources. The last step is to determine a demixing matrix via a contrast function and optimization technique that rotates signals to produce the independent uniform distributions. This concept of a rotation is important for determining the impact of an outlier on the separation performance. Note, the kurtosis of the estimated sources is -1.2; the distribution with the most sub-Gaussian signal possible. The reason the Gaussianity of the signals is brought up, is because maximizing non-Gaussianity is a standard source separation principle used in linear-ICA. This point is further discussed in Sec. 2.2.3.1.

In summary, linear-ICA has been broken down into four key components, (i) a contrast function, (ii) an optimization technique, (iii) whitening, and (iv) rotation. These components are the basis

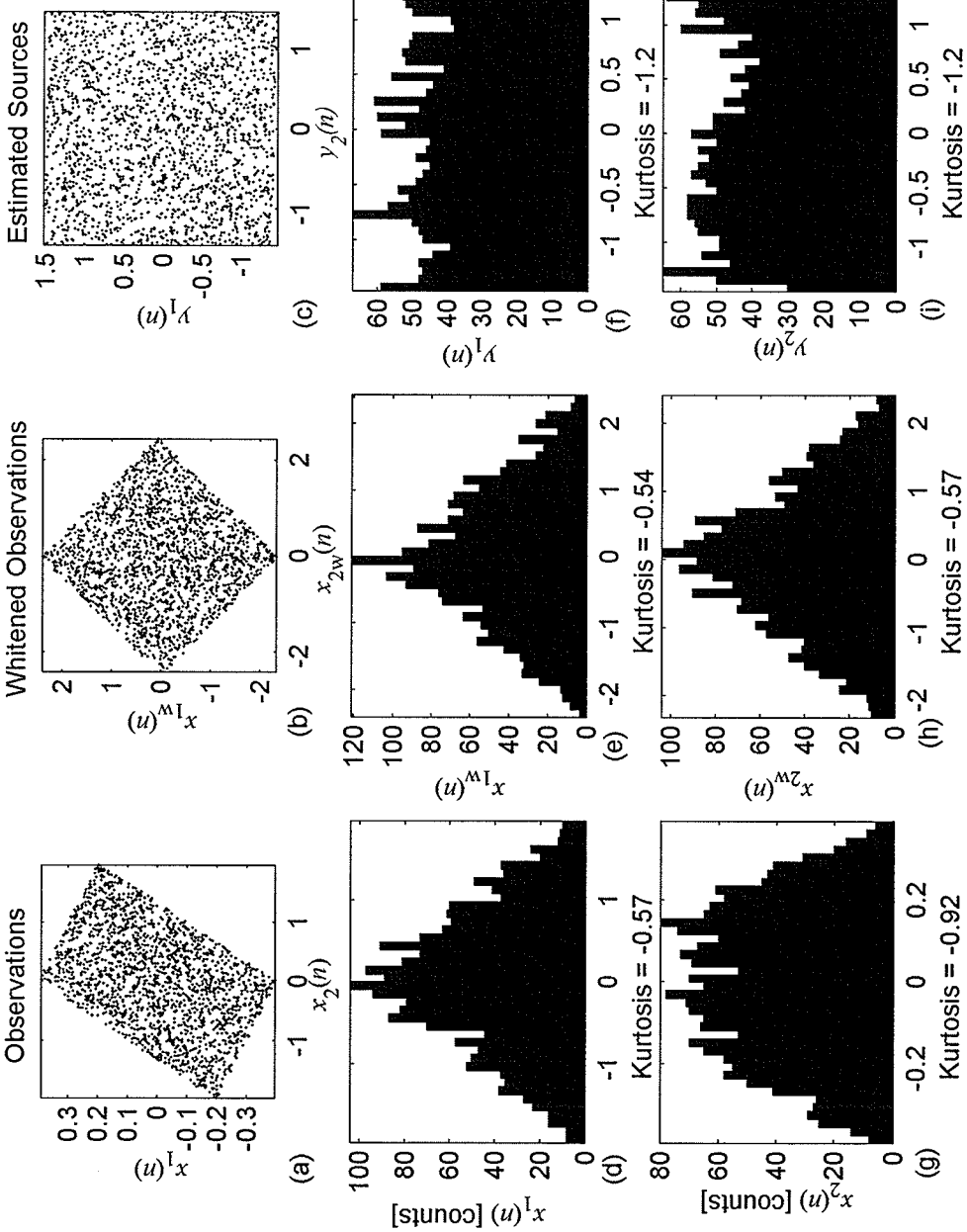


Fig. 2.4 Linear-ICA by whitening and a rotation. (a) Scatter plot of the observation mixtures $x_1(n)$ and $x_2(n)$. (b) Scatter plot of the whitened observation mixtures $x_{1w}(n)$ and $x_{2w}(n)$. (c) Scatter plot of the estimated sources $y_1(n)$ and $y_2(n)$. (d) Histogram of the observation mixture $x_1(n)$. (e) Histogram of the whitened observation mixture $x_{1w}(n)$. (f) Histogram of the source estimate $y_1(n)$. (g) Histogram of the observation mixture $x_2(n)$. (h) Histogram of the whitened observation mixture $x_{2w}(n)$. (i) Histogram of the source estimate $y_1(n)$.

for determining the sensitivity of an ICA algorithm to outliers. However, although this thesis has claimed that ICA performs well in all situations, there are a few restrictions on the distribution of signals that linear-ICA can separate before it has a chance of producing useful results.

2.2.2.2 Restrictions and Assumptions

ICA is a method for blind source separation, although linear-ICA is not completely blind because it has two restrictions/assumptions in addition to those discussed in Sec. 2.2.2. The first restriction is that it limits the original sources to no more than one Gaussian distribution (*i.e.*, the moments higher than two are zero). A proof regarding this is found in [18], but a simple example is used here to demonstrate why this is not allowed. Consider Fig. 2.5, where two signals with Gaussian distributions have been mixed to produce the observations $x_1(n)$ and $x_2(n)$. After viewing the circular shaped scatter plot of the whitened signals, it is clear that a rotation does not produce a distribution that would optimize a contrast function that is attempting to find a rotation that would produce two independent sources. Considering the statistics of the two components, after whitening, all moments of the two distributions are equal. Thus, ICA has no way of utilizing the higher-order statistics of the signals to separate them. In situations like this, alternate (temporal based) methods must be used to separate the signals.

The second restriction of linear-ICA is that the source distributions do not reduce to a point-like mass. Again, Comon [18] discusses the rigorous mathematical reasoning for this restriction. Note that some ICA algorithms require the PDFs to have no discontinuities; *i.e.*, jumps. Allowing these distributions would lead to divergent derivatives, and an inability to converge to a solution.

2.2.2.3 Scale and Permutation Ambiguity in ICA

There is an inherent scale and permutation ambiguity in the source estimates that ICA produces. The scale ambiguity is simply due to the fact that with both $\bar{s}$ and **A** being unknown, any scalar multiplier on $\bar{s}$ can be cancelled by a corresponding column in **A** [36]. Similarly, the permutation ambiguity is due to the fact that $\bar{s}$ and **A** are unknown. Equation 2.4 shows that the inversion and multiplication with a permutation matrix, **P**, results in a new unknown mixing matrix, but with

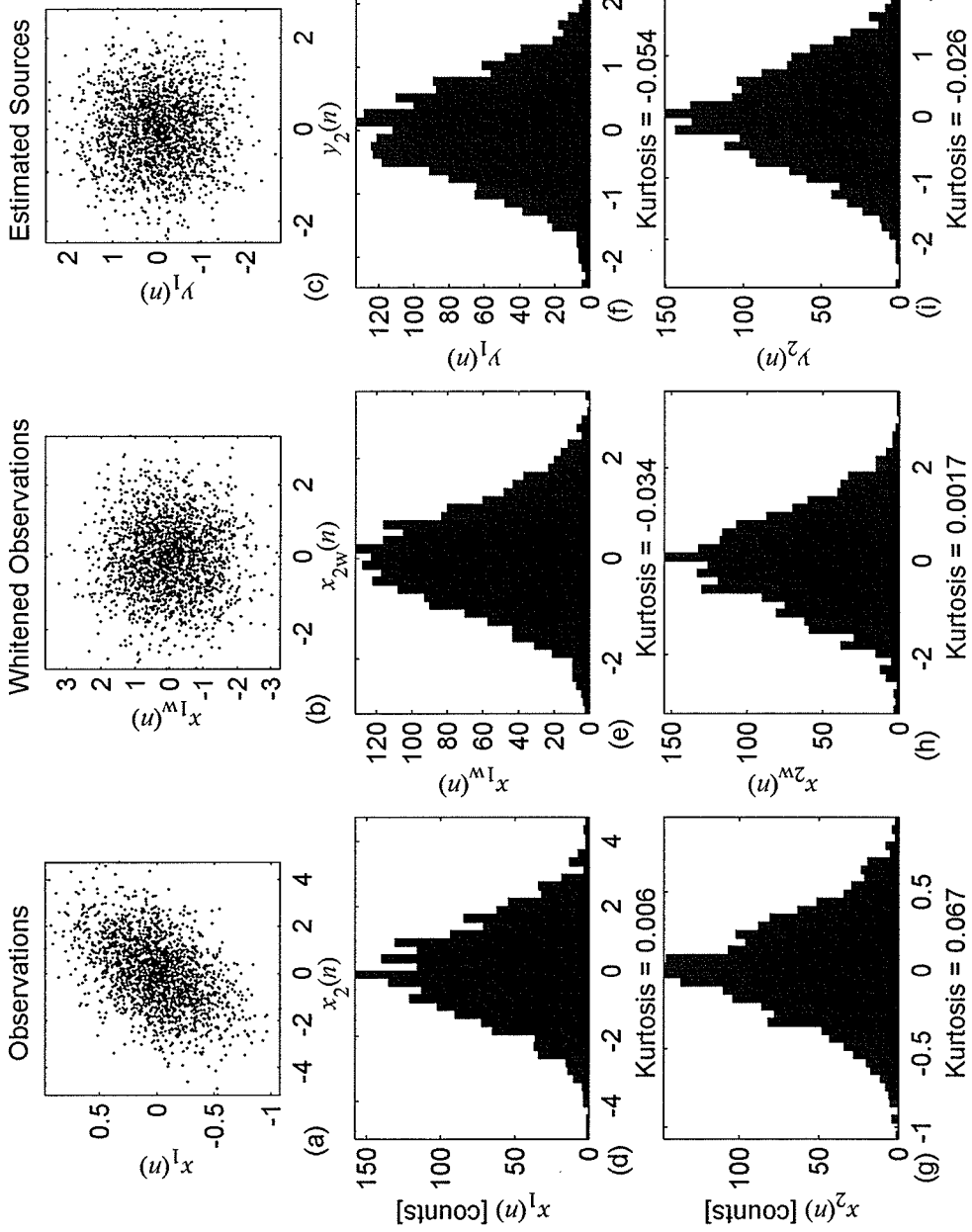


Fig. 2.5 Linear-ICA by whitening and a rotation of a mixture of two Gaussian distributed signals. (a) Scatter plot of the observation mixtures $x_1(n)$ and $x_2(n)$. (b) Scatter plot of the whitened observation mixtures $x_{1w}(n)$ and $x_{2w}(n)$. (c) Scatter plot of the estimated sources $y_1(n)$ and $y_2(n)$. (d) Histogram of the observation mixture $x_1(n)$. (e) Histogram of the whitened observation mixture $x_{1w}(n)$. (f) Histogram of the source estimate $y_1(n)$. (g) Histogram of the observation mixture $x_2(n)$. (h) Histogram of the whitened observation mixture $x_{2w}(n)$. (i) Histogram of the source estimate $y_1(n)$.

permuted sources.

$$\bar{x} = (\mathbf{AP}^{-1})(\mathbf{P}\bar{s}) = \tilde{\mathbf{A}}\tilde{\bar{s}} \tag{2.4}$$

where $\tilde{\mathbf{A}}$ is just a new unknown mixing matrix, and $\tilde{\bar{s}}$ is the same set of unknown sources but permuted. However, the effects of these ambiguities are usually inconsequential as most source estimates are normalized, and signal features can be used to determine the signal of interest [36].

2.2.3 ICA Separation Principles

The five major source separation principles used in linear-ICA to develop a contrast function for separating mixtures of signals are (i) maximizing non-Gaussianity, (ii) maximum likelihood estimation, (iii) minimization of mutual information, (iv) diagonalizing cumulant tensors, and finally (v) non-linear decorrelation. Although each principle is distinct, some contrast functions developed from these principles have been proved to be equivalent [49].

2.2.3.1 Non-Gaussianity and Its Measures

The maximization of non-Gaussianity separation principle follows from the additive central limit theorem which states that an additive mixture of non-Gaussian distributions moves closer to a Gaussian. Thus, the demixing matrix that creates the least Gaussian distribution must produce the independent sources [36], [67]. Two popular measures of non-Gaussianity are negentropy and kurtosis.

Negentropy (Sec. 2.2.4.3) measures the distance (non-negative measure) between the PDF of interest and a Gaussian distribution using differential Shannon entropy. A PDF that is Gaussian has a negentropy of zero. Thus, finding a demixing matrix that maximizes the negentropy measure should separate the signals.

Kurtosis (Sec. 2.2.4.1) is a 4th-order measure of the peakedness and tail weight of a distribution [63]. A Gaussian distribution (with unit variance and zero mean) has a kurtosis measure of 0. A distribution that is flatter at the mean and has has shorter tails than a Gaussian distribution is defined as sub-Gaussian distribution (kurtosis < 0). As well, a distribution that is sharper at the mean

and has longer tails than a Gaussian distribution is defined as super-Gaussian (kurtosis > 0). Thus, kurtosis is a Gaussianity measure because the deviation from 0 is considered as a distance of the distribution from a Gaussian.

One final point to consider is that both negentropy and kurtosis require the PDF of the random signal to calculate their metric. Approximations of the PDF are very susceptible to outliers, and thus so is negentropy and kurtosis (discussed in Sec. 2.2.4.3 and 2.2.4.1).

2.2.3.2 Maximum Likelihood Estimation

Maximum likelihood estimation (MLE) for linear-ICA selects a family of parametric distributions (*i.e.*, $p_m(\bullet)$ where m ranges from 1 to M) to represent the unknown sources, and then attempts to determine the demixing matrix that gives the highest probability for those observations (maximizes the estimator) [36], [54]. Consider applying two demixing matrices to a set of data, and suppose the model of the data is a Gaussian distribution with a mean of 10. The first demixing matrix results in a distribution with the majority of the samples near 0 , and the second results in a distribution with the majority of the samples near 10. Using the demixing matrix that results with the data having a mean near 0 rather than 10 will result in a lower likelihood as the probability of the model is low at 0. Thus, by attempting to maximize the likelihood estimator, the demixing matrix resulting in the majority of the samples near 0 will be selected.

The selection of a family of parametric distributions that models the unknown sources is a difficult task. *A priori* information on the sources maybe used, but usually we are dealing with unknown sources. The primary method used in ICA is to estimate if the unknown sources are sub- or super-Gaussian.

To understand MLE mathematically, consider the probability density of $\bar{\mathbf{x}}$ from Eq. 2.2 (see [54] or [36] on how to calculate the density of a random variable after a linear transformation).

$$p(\bar{\mathbf{x}}) = \frac{1}{|\det\mathbf{A}|} p(\bar{\mathbf{s}}) = \frac{1}{|\det\mathbf{A}|} \prod_{m=1}^{M} p_m(\bar{s}_m) = |\det(\mathbf{W})| \prod_{m=1}^{M} p_m(\mathbf{w}_m \bar{\mathbf{x}}) \tag{2.5}$$

where M is the number of sources, $\mathbf{W} = \mathbf{A}^{-1}$ and is square with order M, and $\mathbf{w}_m$ is the mth row of $\mathbf{W}$. Similar to the derivation in [36], assume that we have N observations of $\bar{\mathbf{x}}$, then the likelihood

(same as contrast function), $L(\mathbf{W})$, is the product of the density evaluated at the N sample points

$$L(\mathbf{W}) = \prod_{n=1}^{N} \prod_{m=1}^{M} p_m(\mathbf{w}_m \mathbf{x}(n)) |\det(\mathbf{W})| \tag{2.6}$$

Taking the logarithm we obtain the following log-likelihood ratio which we seek to maximize in order to separate the sources.

$$\log L(\mathbf{W}) = \sum_{n=1}^{N} \sum_{m=1}^{M} \log p_m(\mathbf{w}_m \mathbf{x}(n)) + N \log|\det(\mathbf{W})| \tag{2.7}$$

Theorem 9.1 from [36] provides a constraint on the $p_m(\bullet)$ that allows mis-specification in the densities not to effect the consistency of the $L(\mathbf{W})$ estimator. A pair of example log-densities that meet this constraint are

$$\log p_1(\mathbf{w}_m \mathbf{x}(n)) = a_1 - 2\log \cosh(\mathbf{w}_m \mathbf{x}(n)) \tag{2.8}$$

$$\log p_2(\mathbf{w}_m \mathbf{x}(n)) = a_2 - \left((\mathbf{w}_m \mathbf{x}(n))^2/2 - \log \cosh(\mathbf{w}_m \mathbf{x}(n))\right) \tag{2.9}$$

where a_1 and a_2 are positive scalar values that make the equations a logarithm of a PDF. Note that Eq. 2.8 is a super-Gaussian density, and Eq. 2.9 is a sub-Gaussian density. Extended-Infomax (Sec. 2.3.2), and Bell-Sejnowski [36] algorithms are two examples of linear-ICA algorithms that maximize a log-likelihood between the source density estimate and the hypothesized source density to estimate the independent sources.

2.2.3.3 Minimization of Mutual Information

Entropy is an expected measure of the uncertainty (self-information) of a random variable; *i.e.*, the average amount of information required to describe a random variable [19]. Mutual information is a measure of the information shared between two random variables. If two random variables are statistically independent, then their mutual information is zero. The mutual information, $I(\bullet)$, between M random variables, $\overline{\mathbf{y}}$ is defined as

$$I(\overline{\mathbf{y}}) = \sum_{m=1}^{M} H_s(\overline{y}_m) - H_s(\overline{\mathbf{y}}) \tag{2.10}$$

where, $H_s(\bullet)$, is the Shannon-differential entropy of the signals [19]. The Shannon differential entropy of the random variable, $\bar{y}$, is defined as

$$H_s(\bar{y}) = -\int_{-\infty}^{\infty} p(\bar{y}) log p(\bar{y}) d\bar{y} \tag{2.11}$$

The Shannon-differential entropy is discussed in detail in Sec. 2.2.4.3. Equation 2.10 can be rewritten as the Kullback-Leibler divergence [19], or relative entropy, between the density of $\bar{\mathbf{y}}$ and the product of its marginals.

$$D_k\left(p(\bar{\mathbf{y}}, \prod_{m=1}^{M} p(\bar{y}_m)\right) = \int_{-\infty}^{\infty} p(\bar{\mathbf{y}}) log \frac{p(\bar{\mathbf{y}})}{p(\bar{y}_1)p(\bar{y}_2)...p(\bar{y}_M)} dh \tag{2.12}$$

where $dh = d\bar{y}_1 d\bar{y}_2 ... d\bar{y}_M$ [42]. Again, Eq. 2.12 equals zero if and only if the signals are mutually independent. Thus, minimization of mutual information between the observations is one method to obtain the independent components in linear-ICA. RADICAL and β-divergence are two linear-ICA techniques that use the minimization of mutual information source separation principle.

2.2.3.4 Tensorial Methods

A covariance matrix is a 2nd-order cumulant tensor [36]. It is used in whitening to decorrelate random variables such that they are independent at the 2nd-order. This concept extended to the 4th-order cumulant tensor is a technique used in linear-ICA for solving the BSS problem. The objective is to make the 4th-order correlations as small as possible (by diagonalizing the tensors) in order to achieve independence. JADE is a linear-ICA tensorial method, and is covered in Sec. 2.3.3 where the mathematics for the tensorial method is covered.

2.2.3.5 Nonlinear Decorrelation

Nonlinear decorrelation for ICA is defined as estimating $\bar{\mathbf{y}}$ such that any two components; *i.e.*, $\bar{y}_i$ and $\bar{y}_j$, $i \neq j$, are uncorrelated, and their transformed components $g(\bar{y}_i)$ and $h(\bar{y}_j)$ are uncorrelated, where $g(\bullet)$ and $h(\bullet)$ are some nonlinear functions [36]. The method by Jutten and Hérault [38] is considered a non-linear decorrelation technique. However, non-linear decorrelation methods are not covered in this thesis, as they are not as well-known or well-used as the other techniques covered.

2.2.3.6 Equivalence of Separation Principles

Although maximizing non-Gaussianity, maximum likelihood estimation, and minimizing mutual information appear to be three distinct approaches for creating source separation contrast functions, they are mathematically similar [12], [36]. Section 10.2 of [36] proves that ICA estimation by minimization of mutual information is equivalent (not equal) to maximizing the sum of non-Gaussianities of the estimates of the independent components when the estimates are constrained to be uncorrelated. Section 10.3 of [36] proves that mutual information can be approximated with a likelihood estimator. The importance of the above information is that although these three principles are equivalent it does not prove that their sensitives to outliers are similar. As discussed in Sec. 2.2.4, there are many ways to approximate PDFs and entropy.

2.2.4 PDF and Entropy Estimation

This section details the various techniques used in the implementation of linear-ICA algorithms studied in this thesis for estimating higher-order statistics, PDFs, and entropy. The impact of outliers on these estimation techniques is highlighted, and alternative approaches are mentioned when appropriate. Note, these techniques are just a few of the various approaches to estimate PDFs and entropy (see [17] and [30] for additional techniques). The section begins with the calculation of the higher-order statistical measure named kurtosis.

2.2.4.1 Measures of Kurtosis

Kurtosis (a non-Gaussianity measure) is a 4th-order measure of the peakedness and tail weight of a random variables PDF [63], [64]. Contrary to common knowledge, there is more than one kurtosis measure; each having a different sensitivity to outliers. First, let us define the mean of a continuous random variable, $\bar{x}$, as

$$\mu(\bar{x}) = \mathcal{E}(\bar{x}) = \int_{-\infty}^{\infty} \bar{x} d\bar{x} \tag{2.13}$$

The mean of a digital signal, $x(n)$, is defined as

$$\mu(x(n)) = \frac{1}{N} \sum_{n=1}^{N} x(n) \tag{2.14}$$

The variance or 2nd central statistical moment of a continuous variable, $\bar{x}$, is defined as

$$\sigma_2(\bar{x}) = \mathcal{E}((\bar{x} - \mu(\bar{x}))^2) = \int_{-\infty}^{\infty} (\bar{x} - \mu(\bar{x}))^2 d\bar{x} \tag{2.15}$$

The variance (bias corrected) of a discrete variable, $x(n)$, is defined as

$$\sigma_2(x(n)) = \frac{1}{N-1} \sum_{n=1}^{N} (x(n) - \mu))^2 \tag{2.16}$$

where μ is the mean of $x(n)$. The kurtosis is the 4th central statistical moment of $\bar{x}$, and is defined as

$$\sigma_4(\bar{x}) = \mathcal{E}((\bar{x} - \mu(\bar{x}))^4) = \int_{-\infty}^{\infty} (\bar{x} - \mu(\bar{x}))^4 d\bar{x} \tag{2.17}$$

The kurtosis (bias corrected) of a discrete variable, $x(n)$, is defined as

$$\sigma_4(x(n)) = \frac{N(N+1)}{(N-1)(N-2)(N-3)} \sum_{n=1}^{N} ((x(n) - \mu)^4 / \sigma^4) - \frac{3(N-1)^2}{(N-2)(N-3)} \tag{2.18}$$

Finally, the normalized kurtosis or kurtosis excess of $\bar{x}$ and $x(n)$ is defined as

$$\kappa(\bar{x}) = \frac{\sigma_4(\bar{x})}{(\sigma_2(\bar{x}))^2} - 3 \tag{2.19}$$

$$\kappa(x(n)) = \frac{\sigma_4(x(n))}{(\sigma_2(x(n)))^2} - 3 \tag{2.20}$$

A random variable with $\kappa < 0$ is said to have a sub-Gaussian, or platykurtic, distribution. A random variable with $\kappa > 0$ is said to have a super-Gaussian, or leptokurtic, distribution. A bias corrected estimate of kurtosis is sensitive to points far away from the mean of $x(n)$. The difference between a sample point and the mean has an influence on the estimate on the order of 4. Of course, this assumes the mean and the variance were robustly estimated. If they were not, the kurtosis estimate would have an even greater error. Thus, having outlier robust kurtosis (and mean and variance) measures are important.

Other kurtosis measures are by (i) interfractile ranges, and (ii) by the ratio of two interfractile ranges [63]. This paper by Ruppert described an influence function analysis of all three kurtosis measures mentioned. An influence function analysis is a robust statistical technique that quantifies the impact of sample points on a statistical measure. The details of this technique is covered in Ch.

3. Ruppert showed that kurtosis excess is the most sensitive to extreme points, followed by inter-fractile, and finally by the two interfractile range kurtosis measures. All ICA algorithms known to the author that use kurtosis, use kurtosis excess. It is of interest to investigate using these alternative kurtosis (non-Gaussianity) measures in an algorithm. Thus, when this thesis refers to kurtosis it is intended to mean kurtosis excess (Eqs. 2.17, 2.18).

2.2.4.2 Probability Density Estimation

Parametric, non-parametric and semi-parametric density estimations are three approaches used in implementation of linear-ICA algorithms [9].

The parametric approach assumes a specific density model, and then goes about optimizing a limited number of parameters to fit the model to the data. The drawbacks are that the density model may not suit the data very well, and the density is usually unknown in ICA. Extended-Infomax and β-divergence use the parametric approach, as both assume specific densities in their implementations. However, in both cases, it has been proven that mis-specifications in the density model still allows for a consistent estimator [44], [49]. From an outlier sensitivity perspective, the effects of outliers on the estimation of these parameters is important. The possible impacts are discussed for each algorithm in Sec. 2.3.

The non-parametric approach attempts to determine a density entirely from the sample data. Unfortunately, this approach usually has the number of parameters linked to the sample size. Thus as sample size increases the number of parameters increases creating a more complex model. This approach is the closest for the JADE algorithm, as its implementation approximates HOS, and assumes no density [15]. RADICAL is also considered a non-parametric linear-ICA algorithm, as it does not assume any density models [42]. The possible effects of outliers are discussed for each algorithm in Sec. 2.3. Notice that JADE would be very sensitive as it uses HOS directly.

Finally, semi-parametric methods attempt to combine the best of both approaches by allowing a class of density models, but with the number of parameters independent of the sample size. FastICA, falls into this category as it has derived an entropy approximation based on a Gram-Charlier expansion [64], [36]. Unfortunately, the Gram-Charlier expansion uses raw HOS in its method for

PDF approximation, and thus should be outlier sensitive. However, Welling [68] has proposed a robust alternative to conventional HOS estimation of moments and cumulants for Gram-Charlier, by weighting an expectation with a multivariate Gaussian distribution. The impact on re-deriving FastICA's entropy approximation based on this outlier robust method of Welling is of interest. However, it is important to note that the Gram-Charlier expansion used assumes the densities are close to Gaussian, and thus may not be suitable for long tailed distributions.

In summary, there are three approaches to probability density estimation, and there is a large number of algorithms in each category. Outlier sensitivity must be studied for each method, but may not have an overall impact on the separation performance of an ICA algorithm. PDF estimation is important, as it is usually used as a part of entropy estimation in ICA.

2.2.4.3 Negentropy

Negentropy is a measure of non-Gaussianity. The negentropy of a *random variable* (RV), $\bar{x}$, is defined as the difference between the *differential Shannon entropy* (DSE) of a Gaussian RV and the DSE of $\bar{x}$. Negentropy is a non-Gaussianity measure, because of all RVs with equal variance, the Gaussian RV has the largest DSE. Thus, the RV with the smallest DSE would be the farthest from Gaussian, and would maximize the negentropy measure. Negentropy is a non-negative value, invariant to any linear transformation, and is zero if and only if $\bar{x}$ is Gaussian. Negentropy is used to develop contrast functions for the FastICA algorithm (Sec. 2.3.1), and an algorithm by Comon [18]. The negentropy, H_n, of a signal, $\bar{x}$, is defined as

$$H_n(\bar{x}) = H_s(\bar{x}_g) - H_s(\bar{x}) \tag{2.21}$$

where $\bar{x}_g$ is a Gaussian signal (RV) with the same covariance matrix as $\bar{x}$, and H_s is the DSE of $\bar{x}$. The DSE of $\bar{x}$ with probability density function $p(\bar{x})$ is defined as

$$H_s(\bar{x}) = \mathcal{E}\left(\log\frac{1}{p(\bar{x})}\right) = -\mathcal{E}(\log p(\bar{x})) = -\int_{-\infty}^{\infty} p(\bar{x}) \log_b p(\bar{x})\, d\bar{x} \tag{2.22}$$

where b is the base of the logarithm. For a natural logarithm ($b = e \approx 2.718281828$), the units are in nats, and when $b = 2$, the units are in bits [19].

Unfortunately, Eq. 2.22 is difficult to implement in practice as it requires the estimation of the density $p(\bar{x})$. To solve this problem, Hyvärinen uses the Gram-Charlier expansion and the maximum entropy method to approximate negentropy [34]. First, the probability density $p(\bar{x})$ is expanded using the Gram-Charlier expansion, but truncated to include only the first two non-constant terms. This leads to the first approximation of negentropy as [35, p.115]

$$H_n(\bar{x}) \approx \frac{1}{12}\mathcal{E}\left((\bar{x})^3\right)^2 + \frac{1}{48}\kappa(\bar{x})^2 \tag{2.23}$$

where $\kappa(\bullet)$ is the kurtosis excess of $\bar{x}$. However, Hyvärinen approximated the high-order moments of Eq. 2.23 with expectations of non-polynomial functions, $G(\bullet)$. This was done because (i) the estimation the higher-order moments are highly sensitive to outliers, and (ii) if the cumulants are estimated perfectly, they mainly measure the tails of the distribution, and are largely unaffected by data at the centre of the distribution [36], [34]. Approximating the higher-order moments with expectations of non-polynomial functions stems from the maximum entropy method. The maximum entropy method determines the density function that has the maximum entropy among all densities that meet a set of constraints, such as the mean is zero and the variance is unity. This leads to the final approximation of negentropy as

$$H_n(\bar{x}) \approx \tilde{H}_n(\bar{x}) = \left(\mathcal{E}(G(\bar{x})) - \mathcal{E}\left(G(\bar{x}_g)\right)\right)^2 \tag{2.24}$$

where $\bar{x}_g$ is a Gaussian random variable of zero mean and unit variance, and $G(\bullet)$ is some non-polynomial function.

With the careful selection of the $G(\bullet)$ functions, Hyvärinen argues that this DSE approximation is less sensitive to outliers and to the tails of a distribution when compared to those based on polynomial expansions like Gram-Charlier or Edgeworth [36]. Section 2.3.1 discusses the $G(\bullet)$ functions, and their susceptibility to outliers. For an alternate method to approximate negentropy see Comon's paper [18]. Note that it is of interest to reevaluate these techniques for approximating negentropy (Hyvärinen's and Comon's) by using the robust Gram-Charlier and Edgeworth PDF expansions of Welling [68]. However, the DSE approximation for long-tailed distributions is still an open question

as the Gram-Charlier and Edgeworth expansions assume the density is close to Gaussian. Furthermore, Hyvärinen's approximation should be revisited as assumptions were made due to computing constraints, and these may no longer be a concern.

2.2.4.4 Vasicek Entropy

The m_N-spacing estimator of entropy (which is referred to as v_N-spacing estimate of entropy, or Vasicek entropy from here on) originated as a test for normality based on the knowledge that the Gaussian distribution has the largest Shannon-entropy of any random variable of equal variance [42], [66]. The calculation of Vasicek entropy approximates Eq. 2.22 with the derivative of the cumulative distribution function of $\bar{x}$. This calculation requires the use of order statistics to replace the derivative with a difference operation [64]. Consider a one-dimensional random variable $\bar{x}$, and a random sample of $\bar{x}$ denoted by $x(n)$. The order statistics of a random sample of $\bar{x}$ are simply the elements of the sample rearranged in non-decreasing order: $\check{x}(1) \leq \check{x}(2) \leq, ..., \leq \check{x}(N)$ [42]. A spacing of order v, or v-spacing, is then defined to be $\check{x}(i+v)-\check{x}(i)$, for $1 \leq i \leq i+v \leq N$. Finally, if v is a function of N, one may define the v_N spacing as $\check{x}(i+v_N)-\check{x}(i)$. Thus, the v_N-spacing estimator of entropy is defined as

$$H_v(x(n)) = \frac{1}{N}\sum_{i=1}^{N-v_N} \log\left(\frac{N}{v_N}(\check{x}(i+v_N)-\check{x}(i))\right) \tag{2.25}$$

For the complete derivation see [42]. The implementation of the RADICAL linear-ICA algorithm uses a modified-Vasicek entropy as a part of its contrast function. The details are explained in Sec. 2.3.4.

2.2.5 Other Entropies in ICA

A major reason Vasicek entropy was selected for use in the RADICAL algorithm was its ease of implementation. Renyi entropy is another uncertainty measure used in ICA algorithms [20]. Renyi entropy is a family of entropies for measuring the average information of a random variable [40]. It is of interest to perform an outlier analysis of ICA algorithms using this metric as its claimed ease of implementation and low computational complexity make it desirable for a real-time heart monitoring device, for example.

2.2.6 Optimization Techniques

The ICA source separation principles lead to contrast functions that must be optimized; *i.e.,* the optima (minimum/maximum) of the function must be determined in order to separate the sources. For example, the objective is to *maximize* the non-Gaussianity of the observations, or *minimize* the mutual information between the observations. This section contains a description of optimization techniques used in the linear-ICA algorithms studied. A few books on optimization theory include Bishop [9], Nocedal and Wright [53], and Ascher [5].

This thesis shows that the optimization techniques do not play a significant role in the sensitivity of a contrast function to outliers. However, the overall separation performance of an ICA algorithm is affected, as an optimization method may select a local optimum, instead of the global minimum/maximum. Furthermore, the initial conditions and the stopping criterion of an optimization technique may result in a biased and/or non-optimal result. Thus, when comparing the overall separation performance of ICA algorithms, it is imperative that the optimization landscape be unbiased. This aspect is discussed in Ch. 4. Another point to highlight is that this thesis does not analyze the computation requirements of the linear-ICA algorithms studied. The selected optimization technique is a crucial factor in computational considerations, but the sensitivity of the algorithm is of the utmost importance.

2.2.6.1 Constrained and Unconstrained Optimization

Most linear-ICA algorithms have the basic form of minimizing a contrast function with respect to the demixing matrix $\mathbf{W}$, or one of its rows $\mathbf{w}$. In many cases there are constraints that restrict the possible solutions, such as requiring the rows of $\mathbf{W}$ to have unit norm (Euclidean norm = 1). This is known of constrained optimization. Unconstrained optimization, *e.g.* gradient descent, does not have such restrictions.

2.2.6.2 Gradient Descent

Gradient descent is an optimization algorithm that approaches a local minimum of a function by taking steps proportional to the negative of the gradient (or the approximate gradient) of the function

at the current point [70], [9]. This method begins with a weight vector (which is usually initialized with a random value) $\mathbf{w}^{(0)}$, that is updated iteratively such that at step τ we move in the direction of the largest decrease in the contrast function evaluated at $w^{(\tau)}$.

$$\mathbf{w}^{\{\tau+1\}} = \mathbf{w}^{\{\tau\}} - \eta \nabla C(\bullet)|_{w^{\{\tau\}}} \tag{2.26}$$

where η is the learning rate, and the nabla operator denotes the gradient. Note there are different versions of gradient descent; one for sequential updating and one for batch [9]. For ICA, $\nabla C(\bullet)$ refers to the gradient of the contrast function with respect to the weight vector; *i.e.*, a row of $\mathbf{W}$. The details of such an optimization are discussed for each algorithm in Sec. 2.3. In this thesis, the implementation always attempts to minimize the contrast function, and thus a negative sign is introduced when appropriate. For example, when the objective is to maximize the non-Gaussianity of the signals in order to separate them, a negative sign is introduced into the contrast function such that the contrast function must be minimized.

A similar method to gradient descent is the natural gradient. The natural gradient is based on differential geometry and employs knowledge of the Riemannian structure of the parameter space to adjust the gradient search direction [4]. The Extended-Infomax algorithm by Lee *et al.* [44] uses the natural gradient in its optimization of the Extended-Infomax contrast. This is discussed in detail in Sec. 2.3.2.

2.2.6.3 Fixed Point

A fixed point of a function, $f(\bullet)$, is a number, q, such that $q = f(q)$. The iteration $q^{\{n\}} = f(q^{\{n-1\}})$ for $n = 0,1,2,...$ is called fixed point iteration [5]. Originally, the (Fast Fixed-point ICA) FastICA algorithm used a fixed-point iteration to optimize a kurtosis-based contrast function. This evolved to the current FastICA algorithm which uses a negentropy based contrast function that is optimized by a quasi-Newton method. Fixed-point is left in for heritage reasons. The details of the FastICA algorithm are discussed in Sec. 2.3.1.

2.2.6.4 *Newton*

The Newton method for optimization uses the second-order information of a contrast function, represented by the local Hessian matrix, to determine the direction to the minimum of the error function [36]. The Hessian matrix for the β-divergence contrast function is shown in [50], but the authors selected the Broyden-Fletcher-Goldfarb-Shanno (BFGS) quasi-Newton method for optimization. This thesis has implemented the β-divergence method using BFGS optimization (see Sec. 2.3.5 for details).

2.2.6.5 *Quasi-Newton*

The Broyden-Fletcher-Goldfarb-Shanno (BFGS) method is a quasi-Newton method, as it approximates the Hessian matrix of the contrast function in order to determine the direction to the minimum. BFGS is usually combined with a line search, to prevent over shooting the minimum [9], [5]. FastICA uses a quasi-Newton method for optimization. The details of this quasi-Newton method is discussed in Sec 2.3.1.

2.2.6.6 *Line Search*

The Newton and quasi-Newton optimization methods each have two steps; the first is to determine the direction of the minimum of the function, and the second is to determine how far to move in that direction to reach the minimum. The determination of how far to move in this direction is known as a line search. The line search is a one dimensional search. An example criterion that stops a line search (sets the step length) are the Armijo-Goldstein (or Wolfe) conditions. This thesis has implemented the β-divergence algorithm using the BFGS optimization technique and Wolfe conditions [5].

2.2.6.7 *Jacobi Method*

The Jacobi method is an iterative technique of optimization [13]. The method consists of a sequence of plane rotations (Givens rotations) designed to annihilate the off-diagonal matrix elements [69]. Essentially, a Givens rotation is a numerical linear algebra technique to introduce zeros into

a matrix. A sweep is the calculation of a Givens rotation for all pairs of off-diagonal indices in a matrix. The JADE ICA algorithm uses the Jacobi method to minimize the 4th-order correlations of the mixed signals in order to achieve separation. The details on the JADE algorithm are discussed in Sec. 2.3.3.

2.2.7 Summary

This section has summarized the principles of ICA. It started from the origins of ICA, and proceeded to the mathematical model of linear ICA, ICA separation principles, entropy, PDF approximation techniques, and finally to optimization techniques. Now that the basis of linear-ICA algorithms has been discussed, the details of the individual linear-ICA algorithm are explained with comments on outlier robustness brought up when appropriate.

2.3 ICA Algorithms

This section details the theory and implementation details of the FastICA [36], Extended-Infomax [44], JADE [13], RADICAL [42] and β-divergence [49] ICA algorithms . The subsections explain each algorithm following this general format.

(a) Introduction to section,

(b) One line description of the algorithm and key references,

(c) Why and by whom the algorithm originated,

(d) Contrast function,

(e) Optimization method,

(f) Assumptions and requirements of the algorithm,

(g) Implementation details of algorithm,

(h) Implications of parameter selection for implementation,

(i) Computational requirements,

(j) Outlier robustness comments and studies,

(k) Limitations of algorithm, and

(l) Summary of section.

2.3.1 FastICA

Fast fixed-point ICA (FastICA), originally developed by Hyvärinen and Oja [37] to make the neural network optimization of a kurtosis-based ICA contrast faster by using a fixed-point iteration, uses a quasi-Newton iteration to maximize an approximation of negentropy (non-Gaussianity measure) in order to estimate the independent sources [37], [35], [36]. The weight vector $\mathbf{w}$ (*i.e.*, a row of $\mathbf{W}$) is selected such that the negentropy of $\mathbf{w}\bar{\mathbf{x}}$ is maximized (under the constraint that the $L2$-norm of $\mathbf{w}$, $\|\mathbf{w}\|_2 = 1$, and the rows of $\mathbf{W}$ are orthogonal). For tractability, Hyvärinen devised an approximation to negentropy based on non-polynomial moments (see Sec. 2.2.4.3 for details). Consequently, this approximation requires the selection of an appropriate non-polynomial function [36].

FastICA uses the DSE approximation to derive the approximate negentropy contrast function of one source estimate, $\bar{y} = \mathbf{w}\bar{\mathbf{x}}$, as

$$C_f(\bar{y}, G(\bullet)) = H_n(\bar{y}) = \left(\mathcal{E}(G(\bar{y})) - \mathcal{E}\left(G(\bar{y}_g)\right)\right)^2 \tag{2.27}$$

The selection of $G(\bullet)$ for the negentropy estimate of Eq. 2.27 is now discussed (see [35] for a detailed discussion on the selection of a $G(\bullet)$). Ideally, if $p(\bar{y})$ were known $G(\bar{y}) = -\log p(\bar{y})$ would be selected, as it $\mathcal{E}(G(\bar{y}))$ gives the true DSE. However for FastICA, the two criteria used to select $G(\bullet)$ are that (i) the estimation of $\mathcal{E}(G(\bar{y}))$ should not be difficult and not be too sensitive to outliers, and (ii) $G(\bar{y})$ should not grow faster than quadratic as a function of $|\bar{y}|$ (otherwise it might lead to the non-integrability of $p(\bar{y})$. The four $G(\bullet)$ functions commonly used in FastICA are

(a) $G_1(\bar{y}) = 1/a_1 \times \log(\cosh(a_1 \times \bar{y}))$, where $1 \leq a_1 \leq 2$, also referred to as the "tanh" as its derivative is a tanh;

(b) $G_2(\bar{y}) = -\exp\left(-\bar{y}^2/2\right)$, also referred to as the "gauss" function;

(c) $G_3(\bar{y}) = \bar{y}^4/4$, also referred to as the "pow3" function; and

(d) $G_4(\bar{y}) = \bar{y}^3/3$, also referred to as the "skew" function.

Hyvärinen states that $G_1(\bullet)$ is a good general purpose contrast function. $G_2(\bullet)$ is good for sources that are super-Gaussian, or when outlier-robustness is important. $G_3(\bullet)$ is useful in estimating sub-Gaussian sources. Finally, $G_4(\bullet)$ is effective in approximating the negentropy of skewed distributions [35], [36]. However, $G_3(\bullet)$ and $G_4(\bullet)$ do not meet the outlier-robustness criterion. Once a $G(\bullet)$ function is selected, the negentropy contrast function must be maximized by searching for a demixing vector $\mathbf{w}$.

To reduce the allowed values of $\mathbf{w}$ (optimization landscape) and to speed up the optimization of the negentropy contrast function, the Matlab implementation of FastICA requires that, (i) the observed sample vector $\mathbf{x}(n)$ be centred and whitened, (ii) the rows of $\mathbf{W}$ be orthogonal, and (iii) L2-norm of each $\mathbf{w}$ (*i.e.*, a row of $\mathbf{W}$) $||\mathbf{w}||_2 = 1$. The whitening of $\mathbf{x}(n)$ implies that the demixing matrix must have orthogonal rows. Requiring each row of $\mathbf{W}$ to be of unit norm creates a stable vector $\mathbf{w}$ that maximizes the approximate negentropy contrast function. FastICA converges to this vector by using a quasi-Newton method (see [35] and pp. 201-202 [36] for proofs).

The quasi-Newton method used to maximize Eq. 2.27, leads to the following update equation

$$\mathbf{w}^{\{+\}} = \mathcal{E}\left(\bar{\mathbf{x}}G'(\mathbf{w}\bar{\mathbf{x}})\right) - \mathcal{E}\left((G''(\mathbf{w}\bar{\mathbf{x}})\right)\mathbf{w} \tag{2.28}$$

where $G'(\bullet)$ is the derivative of $G(\bullet)$, and $G''(\bullet)$ is the second derivative of $G(\bullet)$. Unfortunately, the iteration is not guaranteed to converge, but a stabilized update equation is available as

$$\mathbf{w}^{\{+\}} = \mathbf{w} - \nu[\mathcal{E}\left(\bar{\mathbf{x}}G'(\mathbf{w}\bar{\mathbf{x}})\right) - \Psi\mathbf{w}]/[\mathcal{E}\left((G''(\mathbf{w}\bar{\mathbf{x}})\right) - \Psi] \tag{2.29}$$

where $\Psi = \mathbf{E}(\mathbf{w}\bar{\mathbf{x}}G'(\mathbf{w}\bar{\mathbf{x}}))$, and ν is a step size that may change with the iteration count. $\nu = 1$, gives Eq. 2.28, and as ν approaches 0, the convergence becomes more certain. Hyvärinen also developed an update equation for when the observation vector was not whitened. However, this

Table 2.1 Hyvärinen's negentropy FastICA algorithm.

Step	Action
1.	Choose an initial weight vector $\mathbf{w}$
2.	Solve Eq. 2.28
3.	Let $\mathbf{w} = \mathbf{W}^{(+)}/\|\mathbf{w}^{(+)}\|$
4.	If not converged, go back to Step 2.

method calculates the covariance matrix of the data, and is susceptible to outlier contamination. Thus, assuming that the observation vector $\mathbf{x}(n)$ has been centred and whitened the estimation of a single demixing vector by FastICA without stabilization is shown in Table 2.1.

In the final step (not shown in Table 2.1) of the Matlab implementation of FastICA the orthogonalization of the rows of $\mathbf{W}$ by either a deflationary (where the $\mathbf{w}$s are estimated one at a time, and then orthogonalized by a Gram-Schmidt procedure) or symmetric approach (where each $\mathbf{w}$ is estimated in parallel, and then the entire matrix $\mathbf{W}$ is orthogonalized by matrix square roots) is done. An advantage of symmetric orthogonalization is that it minimizes the effect of cumulative estimation errors. Table 8.3 and 8.4 in [36] lists the symmetric and deflation orthogonalization steps.

Some important implementation details not yet discussed are the selection of an initial demixing matrix vector, and a stopping criterion for optimization. The initial demixing matrix is not important for an individual result, but in experiments where averaging the separation performance over a number of different datasets is done, a random demixing matrix should be used to avoid any potential optimization biases. The stopping criterion for convergence in FastICA is if the old weight vector and the new weight vector point in the same direction. However, reaching this stopping criterion is not guaranteed, as the optimization can stop when a set maximum number of iterations is passed.

To our knowledge, there have been only two studies of the outlier-robustness of FastICA [35], [42]. The first study did an influence function analysis of the negentropy approximation. Theorem 3 in [35] states that to have an outlier-robust negentropy approximation, $G(\bullet)$ should be bounded, or at least not grow too fast. The second study was a simulation where outliers were introduced into a mixture, and the Amari separation performance was measured. The simulation measured

the outlier robustness of the RADICAL, KernelICA, FastICA (gauss), Infomax, FastICA (tanh), FastICA (pow3) and JADE linear-ICA algorithms. The simulation revealed that the RADICAL ICA algorithm was the least affected by outliers followed by, KernelICA, FastICA (gauss), Infomax, FastICA (tanh), FastICA (pow3), and finally JADE. Although these results seem to be significant in the context of this thesis, the experiment setup was not done carefully. The optimization methods of each algorithm are distinct, and some were limited in their search for the minimum of their contrast function. This thesis conducts a similar experiment, but with an unbiased experiment setup to demonstrate which linear-ICA contrast function is truly the most robust to outliers (see Ch. 4 for experiment setup).

The primary limitation of FastICA is that the negentropy approximation is for densities near those of a Gaussian, and (as implemented) requires the source estimates be pre-whitened. Although, the whitening requirement can be removed [35], as it is implemented, the optimization landscape maybe suboptimal due to outliers influencing the non-outlier robust whitening method (discussed in Sec. 2.4). Another limitation is that FastICA is its batch algorithm. This imposes a requirement that the mixing matrix cannot change for a batch of samples, as the FastICA must converge to a mixing matrix. Thus, it is not very adaptive to changing mixing environment. However, throughout this thesis, static mixing is assumed. Many assumptions and compromises have been made to come up with the negentropy approximation without outliers. A study is needed to go back and reevaluate the necessities of these assumptions from an outlier perspective.

However, some advantages of FastICA are that it is computationally efficient (proven to have cubic convergence), and that it does not require the selection of a learning rate or other adjustable parameters. See Ch. 14 in [36] for an analysis of the computational load of this algorithm compared to other ICA algorithms

In summary, FastICA uses a quasi-Newton-iteration to maximize an approximation of negentropy in order to estimate the independent sources [37], [36]. The weight vector $\mathbf{w}$ (*i.e.*, a row of $\mathbf{W}$) is selected by a quasi-Newton iteration such that the negentropy of $\mathbf{w}\bar{\mathbf{x}}$ is maximized (under the

constraint that the L2-norm of $\mathbf{w}$, $\|\mathbf{w}\|_2 = 1$, and the rows of $\mathbf{W}$ are orthogonal). Finally, the negentropy approximation requires the selection of an appropriate estimation function to be outlier-robust [36].

2.3.2 Extended-Infomax

The Extended-Infomax algorithm, developed by Lee, Girolami and Sejnowski [44], estimates the independent sources by using the natural gradient to maximize a log-likelihood between the source density estimates and hypothesized source densities. The extended nature of this algorithm is due to the selection of a hypothesized source density based on an estimate of the super/sub-Gaussianity of the unknown sources. This choice allows Extended-Infomax to separate mixtures of super- and sub-Gaussian sources, as opposed to only super-Gaussian sources as in the original Infomax algorithm [8].

The Infomax source separation principle arose from the field of neural networks and the goal of maximizing the information transmission (Infomax) between the inputs and outputs of a neural network. Infomax is achieved by maximizing the output Shannon entropy of a nonlinear transformation of the inputs. This occurs when the transformation is equal to the cumulative density function of the input [8]. The by product is a minimization of the redundancy between the output units of the neural network, and thus creating independent sources. Cardoso [12] proved that the Infomax concept is equivalent to the minimization of the Kullback-Leibler divergence (KLD) between the distribution of the source estimate and the distribution of a hypothesized source distribution [12].

To understand the Extended-Infomax algorithm it is necessary to derive the Infomax algorithm from the perspective of maximum likelihood estimation. Consider that the PDF of $\bar{\mathbf{x}}$ can be expressed as [44]

$$p(\bar{\mathbf{x}}) = |\det\mathbf{W}| p(\bar{\mathbf{u}}) \tag{2.30}$$

where

$$p(\bar{\mathbf{u}}) = \prod_{i=1}^{N} p_i(\bar{u}_i) \tag{2.31}$$

is the hypothesized distribution of the true sources $p(\bar{\mathbf{s}})$. The log-likelihood (a contrast function

which must be maximized in order to separate the sources) equation of Eq. 2.30 is

$$C_i(\bar{\mathbf{u}}, \mathbf{W}) = \log|\det(\mathbf{W})| + \sum_{i=1}^{N} \log p_i(\bar{u}_i) \tag{2.32}$$

Taking the natural gradient of Eq. 2.32, the following update equation proposed by Amari [2] is found

$$\Delta\mathbf{W} = \left(\mathbf{I} - \varphi(\bar{\mathbf{u}})\bar{\mathbf{u}}^T\right)\mathbf{W} \tag{2.33}$$

where

$$\varphi(\bar{\mathbf{u}}) = \frac{\frac{\partial p(\bar{\mathbf{u}})}{\partial \mathbf{u}}}{p(\bar{\mathbf{u}})} = \left(-\frac{\frac{\partial p(\bar{u}_1)}{\partial u_1}}{p(\bar{u}_1)}, \ldots, \frac{\frac{\partial p(\bar{u}_N)}{\partial u_N}}{p(\bar{u}_N)}\right)^T \tag{2.34}$$

and $\mathbf{I}$ is the identity matrix of order N.

If $p(\bar{u}) = tanh(\bar{u})$, then $\varphi(\bar{u}) = 2\tanh(\bar{u})$, and the learning rule of Eq. 2.33 becomes

$$\Delta\mathbf{W} = \left(\mathbf{I} - 2\tanh(\bar{\mathbf{u}})\bar{\mathbf{u}}^T\right)\mathbf{W} \tag{2.35}$$

Equation 2.35 is the same update equation derived by Bell and Sejnowski Infomax [8]. However, it was proved that Eq. 2.35 can only separate mixtures of signals with super-Gaussian distributions [44]. Lee *et al.* had the objective to provide a simple learning rule with a fixed nonlinearity that could separate sources with a variety of distributions [44]. They determined that only two parametric densities were needed, one for sub-Gaussian distributions and one for super-Gaussian distributions.

Lee *et al.* chose to model hypothesized sub-Gaussian densities as

$$p(\bar{u}) = \frac{1}{2}\left(N(\mu, \sigma^2) + N(-\mu, \sigma^2)\right) \tag{2.36}$$

where $N(\bullet)$ is a normal distribution with mean and variance parameters. If $\mu = 0$ and $\sigma = 1$, the learning rule of Eq. 2.33 is

$$\Delta\mathbf{W} = \left(\mathbf{I} + \tanh(\bar{\mathbf{u}})\mathbf{u}^T - \bar{\mathbf{u}}\bar{\mathbf{u}}^T\right)\mathbf{W} \tag{2.37}$$

On the other hand, to model hypothesized super-Gaussian densities, Lee *et al.* chose

$$p(u) = N(0, 1)\mathrm{sech}^2(u) \tag{2.38}$$

where $N(\bullet)$ is a normal distribution with mean and variance parameters. If $\mu = 0$ and $\sigma = 1$, then the learning rule of Eq. 2.33 is

$$\triangle \mathbf{W} = \left(\mathbf{I} - \tanh(\overline{\mathbf{u}})\overline{\mathbf{u}}^T - \overline{\mathbf{u}\mathbf{u}}^T\right)\mathbf{W} \tag{2.39}$$

It is seen that the difference for learning rules for sub and super-Gaussian distribution is only a sign. Thus, determining if the source being estimated is sub or super-Gaussian allows to switch between learning rules. Lee suggested Eq. 2.40 to determine if the unknown source is sub- or super-Gaussian

$$k_i = \text{sign}\left(\left(\mathcal{E}(\text{sech}^2(\overline{u}_i)\right)\mathcal{E}(\overline{u}_i)^2 - \mathcal{E}\left(\tanh(\overline{u}_i)\overline{u}_i\right)\right) \tag{2.40}$$

The update rule for $\mathbf{W}$ then becomes

$$\triangle \mathbf{W} = \left(\mathbf{I} - \mathbf{K}\tanh(\overline{\mathbf{u}})\overline{\mathbf{u}}^T - \overline{\mathbf{u}\mathbf{u}}^T\right)\mathbf{W} \tag{2.41}$$

where k_i are the diagonal elements of the diagonal matrix $\mathbf{K}$. Thus, the algorithm for Extended-Infomax is to whiten the data, estimate Gaussianity of each source, update $\mathbf{W}$, and repeat until converged. The stopping criterion of the implemented algorithm is quite complicated, but is primarily based on whether the new weight vector and the old weight vector point in the same direction [47].

To our knowledge, only the paper by Miller has used the Extended-Infomax in an outlier experiment [42]. The paper showed experimentally that the algorithm, on average, had a worse separation performance than FastICA but better than the JADE algorithm.

2.3.3 JADE

The *joint approximate diagonalization of eigenmatrices* (JADE) algorithm uses the algebraic structure of the 4th-order cumulant tensor of whitened observations to devise a contrast that requires the minimization of the off-diagonal components of a maximal set of cumulant matrices by orthogonal transformations to estimate the demixing matrix. The minimization is achieved by a joint approximate diagonalization (Jacobi method) which, in the case of two sources, optimizes by a plane rotation. However, the optimization becomes unwieldy as the number of sources increases

Table 2.2 Cardoso and Souloumiac's JADE algorithm [15].

Step	Action
1.	Estimate a whitening matrix $\mathbf{M}_w$ and set $\bar{\mathbf{y}} = \mathbf{M}_w\bar{\mathbf{x}}$, where $\bar{\mathbf{x}} = \mathbf{A}\bar{\mathbf{s}}$.
2.	Estimate a maximal set of $N_q = N(N-1)/2$ cumulant matrices $\mathbf{Q}$
3.	Find a rotation matrix $\mathbf{V}$ such that the cumulant matrices are as diagonal as possible; that is find the matrix $\mathbf{V}$ that minimizes the sum of the sum of the off-diagonal components of $\mathbf{V}^T\mathbf{Q}_i\mathbf{V}$, where $i = 1$ to N_q.
4.	Estimate the unknown sources, as $\mathbf{UM}_w$ equals $\mathbf{A}$.

[13]. The JADE algorithm, summarized in Table 2.2 [13], was first introduced by Cardoso and Souloumiac in a paper on blind beamforming for non-Gaussian signals [15].

To derive the algorithm shown in Table 2.2, begin by defining a cumulant matrix, $\mathbf{Q}$ of size $N \times N$, component wise as

$$q_{i,j} = Q(\bar{\mathbf{x}}, \mathbf{Z}) = \sum_{k,l=1}^{N} \mathrm{Cum}(\bar{x}_i, \bar{x}_j, \bar{x}_k, \bar{x}_l) z_{k,l} \tag{2.42}$$

where $q_{i,j}$ is the ith row and jth column entry of $\mathbf{Q}$, $\bar{\mathbf{x}}$ is a vector of $N \times 1$ random variables, $\bar{x}_i$ is the ith random variable, $\mathbf{Z}$ is some $N \times N$ matrix, and $z_{k,l}$ is the kth row and lth column entry of $\mathbf{Z}$. Finally, $\mathrm{Cum}(\bar{x}_i, \bar{x}_j, \bar{x}_k, \bar{x}_l)$ are the 4th order cumulants, and are defined as

$$\mathrm{Cum}(\bar{x}_i, \bar{x}_j, \bar{x}_k, \bar{x}_l) = \mathcal{E}(\bar{x}_i\bar{x}_j\bar{x}_k\bar{x}_l) - \mathcal{E}(\bar{x}_i\bar{x}_j)\mathcal{E}(\bar{x}_k\bar{x}_l) - \mathcal{E}(\bar{x}_i\bar{x}_k)\mathcal{E}(\bar{x}_j\bar{x}_l) - \mathcal{E}(\bar{x}_i\bar{x}_l)\mathcal{E}(\bar{x}_j\bar{x}_k) \tag{2.43}$$

where $1 \le i, j, k, l \le N$. Now, consider that $\bar{\mathbf{x}} = \mathbf{A}\bar{\mathbf{s}}$, then the cumulant matrix, $\mathbf{Q}$, reduces to

$$\mathbf{Q} = \mathbf{A}D(\mathbf{Z})\mathbf{A}^T \tag{2.44}$$

where

$$D(\mathbf{Z}) = \mathrm{Diag}(\kappa(\bar{s}_1)\mathbf{a}_1^T\mathbf{Z}\mathbf{a}_1, ..., \kappa(\bar{s}_N)\mathbf{a}_N^T\mathbf{Z}a_N) \tag{2.45}$$

and $\mathbf{a}_i$ is the ith column of A.

Now, let $\mathbf{Z}_1$ and $\mathbf{Z}_2$ be two arbitrary matrices of size $N \times N$. Let $\mathbf{Q}_1 = Q(\bar{\mathbf{x}}, \mathbf{Z}_1)$, $\mathbf{Q}_2 = Q(\bar{\mathbf{x}}, \mathbf{Z}_2)$, and $\mathbf{J} = \mathbf{Q}_1^{-1}\mathbf{Q}_2^{-1}$. Then though simple manipulation the relationship $\mathbf{JA} = \mathbf{AD}$ appears, where $\mathbf{D}$ is some diagonal matrix. This relationship shows that the columns of $\mathbf{A}$ are the eigenvectors of $\mathbf{J}$.

However, to solve issues regarding matrix inversions and distinct eigenvalues, $\bar{\mathbf{x}}$ must be whitened by matrix $\mathbf{M}_w$ and the $\mathbf{Q}$s must be a set of maximal matrices.

A set of maximal cumulant matrices is defined as

$$\{Q(\bar{\mathbf{x}}, \mathbf{Z}_i) | i = 1, N^2\} \tag{2.46}$$

where

$$\{\mathbf{Z}_i | i = 1, N^2\} = \{\mathbf{e}_p \mathbf{e}_q^T | 1 \leq p, q \leq N\} \tag{2.47}$$

and $\mathbf{e}_p$ is a column vector with a 1 in the pth position, and 0s elsewhere. Now, instead of finding the eigenvectors of $\mathbf{J}$ to determine the mixing matrix $\mathbf{A}$, Cardoso proves that the rotation matrix, $\mathbf{V}$, that minimizes the sum of the off-diagonal components of $\mathbf{VQV}^T$ for the entire set of maximal cumulant matrices is equal to the mixing matrix $\mathbf{A}$ [13]. Thus, the objective is to find a rotation matrix, $\mathbf{V}$, that minimizes Eq. 2.48

$$\sum_{i=1}^{N^2} \mathbf{V}\mathbf{Q}_i\mathbf{V}^T \tag{2.48}$$

Actually, the data is whitened before performing the analysis so, the true mixing matrix is estimated as $\mathbf{VM}_w = \mathbf{A}$. Now, in order to find the rotation matrix that minimizes Eq. 2.48, Cardoso devised a method using repeated Givens rotations to zero elements of the matrices, and minimize the contrast function.

Table 2.3 lists the steps required to find the rotation matrix that minimizes Eq. 2.48 [13]. This method minimizes Eq. 2.48 by optimizing all $N(N-1)/2$ distinct pairs of random variables. Thus, it makes each pair of mixtures independent of each other, and repeats this process until no more significant rotations (rotation angle is less than some value) are found. Selecting pairs of mixing sources makes the cumulant matrix of size 2×2. Thus if only 2 observation signals are being separated only a single Givens rotation (plane rotation) must be calculated to minimize the contrast.

Typically ϕ_{min} is selected as $1/sqrt(T)$, where T is the number of samples of a random variable. In ICA it should be selected such that the remaining rotation is not significant as measured by the Amari separation performance index. Note that if only separating two sources, then only a single rotation matrix must be determined. Furthermore, due to symmetry, only $N(N-1)/2$ of the set of

Table 2.3 Jacobi method [13].

Step	Action
1.	Initialization: Compute a whitening matrix $\mathbf{M}_w$ and set $\bar{\mathbf{y}} = \mathbf{M}_w\bar{\mathbf{x}}$.
2.	One sweep: For all $N(N-1)/2$ pairs; *i.e.* $1 \leq i < j \leq N$, of random variables of $\bar{\mathbf{y}}$ do
2a.	Compute the Givens angle ϕ_{ij}, optimizing 2.48 when the pair $(\bar{y}_i, \bar{y}_j)$ is rotated.
2b.	If Φ_{ij} is $> \phi_{min}$, then rotate the pair $(\bar{y}_i, \bar{y}_j)$. Note, this is different from [13], as the author's believe greater than sign was misprinted.
3.	If no pair has been rotated in the previous sweep, exit, else go to step 2.

maximal matrices are needed minimize Eq. 2.48. In summary, [13] shows that the JADE method minimizes following contrast function in order to determine the mixing matrix required to separate the sources

$$C_J(\bar{\mathbf{y}}) = \sum_{ijkl \neq ijkl} (\text{Cum}(\bar{y}_i, \bar{y}_j, \bar{y}_k, \bar{y}_l)^2 \tag{2.49}$$

where $\bar{\mathbf{y}} = \mathbf{V}\mathbf{M}_w\bar{\mathbf{x}}$.

The limitations of the JADE algorithm are due to statistical and computational aspects. First, the algorithm only looks at 4th order statistics in order to separate. Thus, the skewness of a density is completely ignored. Computationally, as the number of sources increases the number of 4th order statistics that must be calculated increases. Estimating a maximal set of cumulant matrices requires $O(n^4N)$ operations (where N is the number of samples and n is the number of sensors).

Although, the JADE algorithm has a distinct advantage, the lack of parameters to set. The algorithm only requires the setting of a stopping criterion for the Jacobi optimization method.

There is only one outlier robustness study of the algorithm known to us. Learned-Miller [42], in an experiment with 1000 sample points from 100 mixings of 18 of source distributions, demonstrated that JADE had the highest Amari separation performance index compared to the RADICAL,

FastICA and Extended-Infomax algorithms. It is of interest to re-implement the JADE algorithm using the alternative kurtosis methods [63] to calculate 4th-order cumulants. This modification might change the relative performance of the algorithms completely.

In summary, the joint approximate diagonalization of eigenmatrices (JADE) algorithm uses the algebraic structure of the 4th-order cumulant tensor of whitened observations to devise a contrast that requires the minimization of the off-diagonal components of a maximal set of cumulant matrices by orthogonal transformations to estimate the demixing matrix. The minimization is achieved by a joint approximate diagonalization (Jacobi method), which (in the case of two sources) optimizes by a plane rotation. However, the optimization becomes unwieldy as the number of sources increases [13].

2.3.4 RADICAL

The *robust, accurate, direct, independent component analysis algorithm* (RADICAL) uses a modified *v*-spacings estimate of entropy (see Sec. 2.2.4.4) and an exhaustive rotation matrix search to solve the linear-BSS problem. Pre-whitening of the input mixtures reduces the optimization to searching exhaustively for a rotation matrix that optimizes the contrast function. An exhaustive search is required because the optimization landscape is not smooth, but has many local minima [42].

The derivation of the algorithm starts with rewriting Eq. 2.10 as a function of the observation sources, $\bar{\mathbf{x}}$, and the demixing matrix $\mathbf{W}$

$$J(Y) = \sum_{i=1}^{N} H(\bar{y}_i) - H(\bar{\mathbf{x}}) - \log(|\mathbf{W}|) \tag{2.50}$$

If $\bar{\mathbf{x}}$ is prewhitened, and the optimization is restricted to rotation matrices, Eq. 2.50 reduces to finding the demixing matrix $\mathbf{W}$ that minimizes the following contrast function

$$C_R(\bar{\mathbf{y}}) = \sum_{i=1}^{N} H(\bar{y}_i) \tag{2.51}$$

where $H(\bullet)$ for RADICAL is selected as a modified Vasicek entropy estimator. The entropy estimator by Vasicek is selected because it is consistent, has $\sqrt{(N)}$ convergence, and is computable in

Table 2.4 Two-dimensional RADICAL algorithm.

Step	Action
1.	Input: Whiten the the observation vector $x_i(n)$ where $1 < i < 2$, and having N samples.
2.	Parameters: Select the spacing size m (Usually $\sqrt{N}$). Select the noise variance for replicated points σ_r^2. Select the number of replicated points per original data point R. Select the number of angles K at which to evaluate the contrast function at.
3.	Create $\hat{\mathbf{x}}$ by replicating R points with Gaussian noise for each point.
4.	For each θ, rotate the data to this angle ($\mathbf{y} = \mathbf{W}(\theta)\hat{\mathbf{x}}$) and evaluate the contrast function.
5.	Output the W corresponding to the optimal θ.

$O(T\log T)$ where T is the number of sample points. The modified Vasicek entropy estimator that is implemented is derived from Eq. 2.25 as

$$\hat{H}_R(\check{z}) \equiv \frac{1}{N-m}\sum_{i=1}^{N-m}\log\left(\frac{N+1}{m}(\check{z}(i+m)-\check{z}(i))\right) \tag{2.52}$$

Although, the Matlab implementation of the RADICAL algorithm actually implements the following as the modified Vasicek entropy estimator

$$\hat{H}_R(\check{z}) \equiv \sum_{i=1}^{N-m} log\left((\check{z}(i+m)-\check{z}(i))\right) \tag{2.53}$$

The removal of the scalar terms does not change the location of the minimum. Now, as noted in Sec. 2.2.4.4, the selection of m changes the results of the entropy estimator dramatically. To reduce the variance of the result, Learned-Miller and Fisher [42] augment the data set by upsampling each sampled point by replacing it with a set of Gaussian points with the mean of the original sample point. This reduces the dependence on the selected m value, and also results in a smoothing of the optimization landscape, although an exhaustive rotation search is still necessary. Thus, the RADICAL algorithm for 2-dimensions is summarized in Table 2.4. For higher dimensions, a Jacobi like optimization similar to that of JADE is required.

Learned-Miller and Fisher demonstrated that RADICAL was the most outlier-robust in experiments involving FastICA, Extended-Infomax, and JADE algorithms [42]. However, there is one

problem with the results. The experiment measured the overall outlier-robustness of the contrast function and optimization technique. A poor optimization technique could result in discarding an outlier-robust contrast function. Specifically, each of the algorithms had a pre-whitening stage. However, these were non-outlier robust whitening techniques. Thus, if a non-outlier robustness whitening technique resulted in an optimization landscape that was biased towards algorithms that could search the entire landscape, rather than just rotation matrices, premature discarding of a contrast function could result. This aspect is discussed in detail in Ch. 4.

The advantages of the RADICAL algorithm is that it requires few tuning parameters, and few computational demands. In the case of outliers, it has been demonstrated to be outlier robust, although in a study without an unbiased optimization landscape.

In summary, RADICAL uses a modified v-spacings estimate of entropy and an exhaustive rotation matrix search to solve the linear-BSS problem.

2.3.5 Beta-Divergence

The β-divergence is an extension of the *Kullback-Leibler divergence* (KLD) for $\beta = 0$, and changes continuously up to the mean-squared-error distance for $\beta = 1$. The β parameter is introduced to reduce the impact of outliers on the distance measure. This is an important algorithm to be studied as it has been proved to be B-robust (the influence of an outlier on the estimator is finite), as apposed to most other ICA algorithms that have been proved to be non-B-robust [49]. In addition, the β-divergence is important because it provides a non-negative distance measure.

The β-divergence distance measure between two density functions, g and f, with respect to some carrier measure v (see [1]) is expressed as

$$D_\beta(g,f) = \begin{cases} \frac{1}{\beta}\int_\chi \left(g^\beta(\bar{\mathbf{x}}) - f^\beta(\bar{\mathbf{x}})\right) g(\bar{\mathbf{x}}) dv(\bar{\mathbf{x}}) \\ -\frac{1}{\beta+1}\int_\chi \left(g^{\beta+1}(\bar{\mathbf{x}}) - f^{\beta+1}(\bar{\mathbf{x}})\right) dv(\bar{\mathbf{x}}) & \beta > 0 \\ \int_\chi g(\bar{\mathbf{x}}) \log\left(\frac{g(\bar{\mathbf{x}})}{f(\bar{\mathbf{x}})}\right) dv(\bar{\mathbf{x}}) & \text{for } \beta = 0 \\ \frac{1}{2}\int_\chi (g(\bar{\mathbf{x}}) - f(\bar{\mathbf{x}}))^2 dv(\bar{\mathbf{x}}) & \text{for } \beta = 1 \end{cases} \tag{2.54}$$

The β-divergence is related to the density power divergence of Basu *et al.* [7], $D_b(g, f)$, as

$$D_\beta(g, f) = \frac{D_b(g, f)}{1 + \beta} \tag{2.55}$$

Note, Minami and Eguchi [50] list the relationship as

$$D_b(g, f) = \frac{D_\beta(g, f)}{1 + \beta} \tag{2.56}$$

which we believe is a typographical error after re-deriving the equation.

As β reaches 0, $D_\beta(g, f)$ approaches KLD, and when $\beta = 0$, D_β equals the KLD. If we find $D_\beta(r, r_0(\mathbf{x}(n), \mathbf{W}, \mu))$, the β-divergence distance between the empirical distribution of the observations r and the product of the approximated marginal densities of $\mathbf{x}$, $r_0(\mathbf{x}(n), \mathbf{W}, \mu)$, we obtain the following contrast function which must be maximized in order to separate a mixture of sources.

$$C_\beta(\mathbf{x}(n); \mathbf{W}, \mu) = \begin{cases} \frac{1}{N}\frac{1}{\beta}\sum_{n=1}^{N} r_0^\beta(\mathbf{x}(n), \mathbf{W}, \mu) - b_\beta(\mathbf{W}) - \frac{1-\beta}{\beta} & \text{for } 0 < \beta < 1 \\ \frac{1}{N}\sum_{n=1}^{N} \log(r_0(\mathbf{x}(n), \mathbf{W}, \mu)) & \text{for } \beta = 0 \end{cases} \tag{2.57}$$

where

$$r_0(\mathbf{x}(n); \mathbf{W}, \mu) = |\det(\mathbf{W})| \prod_{m=1}^{M} p_m(\mathbf{w}_m^T \mathbf{x}(n) - \mu_m) \tag{2.58}$$

$$b_\beta(\mathbf{W}) = \frac{|\det(\mathbf{W})|^\beta}{\beta + 1} \int_{-\infty}^{\infty} \prod_{m=1}^{M} p_m^{\beta+1}(z_m) dz_m \tag{2.59}$$

$$z_m = \mathbf{w}_m^T \mathbf{x}(n) - \mu_m \tag{2.60}$$

where p_m is a specific density function, and $\mu = (\mu_1, ..., \mu_m)^T$ is a vector of shift parameters to be estimated. Minami and Eguchi proved that, with the use of various p_ms, the gradient of Eq. 2.57 is a consistent estimator; *i.e.*, converges to a result as the sample size grows. Minami and Eguchi have also provided density functions for the p_m. Note, this algorithm does not assume the observations were prewhitened, thus the reasoning for estimating the shift-vector (mean).

Typical p_is are

$$p(z) = c\exp(-dz^4) \tag{2.61}$$

$$p(z) = c/cosh(z) \tag{2.62}$$

where c and d are scalars set such that the integral of $p(z)$ is 1 and is a probability density function. Minami and Eguchi suggested that Eq. 2.61 be used when working with sub-Gaussian distributions, and Eq. 2.62 for super-Gaussian distributions.

Regarding outlier robustness, Minami and Eguchi proved the algorithm is B-robust; that is the influence of an outlier on the estimate is limited. Unfortunately, Minami and Eguchi have not provided source code that implemented this algorithm. Since the characteristics of the algorithm are of interest, an implementation of the algorithm is explained in the next section.

The remaining issue to be discussed is the selection of β. Recently Minami and Eguchi published a paper on the adaptive selection of β using cross validation [50]. The paper suggests to validate the selection of β by running the algorithm on a subset of the data with a range of βs. The smallest β that has a result that matches with the majority of the results with different βs is selected. In this thesis, this selection technique is not implemented. A range of βs are used in experiments, and all of the results are reported.

2.3.6 Implementation of β-Divergence Algorithm

Determining the demixing matrix **W**, and shift vector μ, that maximizes Eq. 2.57 is accomplished by using the BFGS optimization method, combined with an Armijo conditioned line-search to find the minimum of the negative of Eq. 2.57 [9], [5]. Note Minami and Eguchi state that if Eq. 2.59 is left as a constant or zero, then the results are still good. Thus, in our implementation Eq. 2.59 is left as zero to reduce complexity. Now, this implementation combines **W** and μ (when $\beta \neq 0$) to form a single 1-dimensional vector, **v**, that is updated by the BFGS method. The algorithm implemented in Matlab is based on the following (Table 2.5) BFGS algorithm originating from [5]. See *BetaD_ICA.m* in Appendix B for the code implementation. The code is also available on the Web [23].

where k is the iteration step, and $C(\bullet)$ is the gradient of Eq. 2.57 with respect to both **W** and μ (see

Table 2.5 Implemented β-divergence algorithm.

Step	Action
1.	While $\mathbf{v}^{\{k+1\}} - \mathbf{v}^{\{k\}} <$ stopping criterion.
2.	Calculate the direction to move in as $d^{\{k\}} = -G^{\{k\}} \nabla C(x(t), \mathbf{W}, \mu)^{\{k\}}$
3.	Determine how far to move in that direction by a line search (solve for $\alpha^{\{k\}}$ via Table 2.6).
4.	Determine the new vector. $\mathbf{v}^{\{k+1\}} = \mathbf{v}^{\{k\}} + \alpha d^{\{k\}}$
5.	Update the Hessian approximation $G^{\{k+1\}} = \left(I - \frac{s^{\{k\}} y^{T\{k\}}}{y^{T\{k\}} s^{\{k\}}}\right) G^{\{k\}} \left(I - \frac{y^{\{k\}} s^{T\{k\}}}{s^{T\{k\}} y^{\{k\}}}\right) + \frac{s^{\{k\}} s^{T\{k\}}}{y^{T\{k\}} y^{\{k\}}}$
6.	Go to step 1.

Eq. 3.6 and 3.7 in [49]). The line search is based on Armijo conditioned line search derived in [5]. The steps are outlined in Table 2.

Table 2.6 Line search.

Step	Action
1.	Solve $a = d^{T\{k\}} \nabla C(x(t), \mathbf{W}, \mu)^{\{k\}}$
2.	Let $\alpha^{\{k\}} = 1 and solve \mathbf{v}^{\{k+1\}} = \mathbf{v}^{\{k\}} + \alpha d^{\{k\}}$
3.	While $(C(x(t), \mathbf{W}, \mu)^{\{k+1\}}) > C(x(t), \mathbf{W}, \mu)^{\{k\}}) + \sigma\alpha^{\{k\}} a$ and $\alpha^{\{k\}} > \alpha_{\min}$
4.	Update the value of α as $\alpha^{\{k+1\}} = \frac{\alpha^{\{k\}}((-0.5)a\alpha^{\{k\}})}{C(x(t),\mathbf{W},\mu)^{\{k+1\}} - C(x(t),\mathbf{W},\mu)^{\{k\}} - \alpha^{\{k\}} a}$
5.	Update $\mathbf{v}^{\{k+1\}} = \mathbf{v}^{\{k+1\}} + \alpha d^{\{k\}}$
6.	Go to step 3.

If the line search fails to find a direction which reduces the solution to the contrast function, then either the parameters have found a minimum (local or global), or the Hessian approximation has become corrupt due to a singular matrix. In this implementation, the Hessian approximation is reset, and the algorithm is repeated, but starting from the current location. The stopping criterion is based on that of FastICA and Extended-Infomax, that the new and the previous weight vector (excluding the μ) points in the same direction. However, this does not ensure that this criterion was

met, as each algorithm also has a maximum number of iterations allowed before halting.

The primary limitation of the implementation is that it is only handles the separation of 2 signals. It is of interest to adapt the implementation to a larger number of signals.

2.4 Whitening, Rotation Matrices and ICA

Whitening (or sphereing) is a common preprocessing step in most linear-ICA algorithms. A zero-mean random vector, $\bar{\mathbf{z}}$, is said to be white (or sphered) if its elements are uncorrelated and have unit variance [36]. Whitening reduces the complexity of ICA. Whitening determines the independent components up to an orthogonal transformation [36]. An orthogonal matrix contains $m(m-1)/2$ degrees of freedom, where m is the number of observation vectors. In 2-dimensions, this reduces the optimization of the contrast function to a search for a single rotation matrix. Common techniques for whitening are *eigenvalue decomposition* (EVD) and *principal component analysis* (PCA) (see [17] for additional methods). The method used in the ICA algorithms studied is EVD.

2.4.0.1 Eigenvalue Decomposition

Eigenvalue decomposition determines a linear transformation matrix $\mathbf{M}_w$ to whiten observed samples $\bar{\mathbf{x}}$ as

$$\bar{\mathbf{z}} = \mathbf{M}_w \bar{\mathbf{x}} \tag{2.63}$$

where

$$\mathbf{M}_w = \mathbf{D}^{-1/2}\mathbf{E}^T \tag{2.64}$$

where $\mathbf{D}$ is the diagonal matrix of the eigenvalues of the covariance matrix of $\bar{\mathbf{x}}$, and $\mathbf{E}$ is a matrix whose columns are the unit-norm eigenvectors of the same covariance matrix. The concern with whitening and outliers is the estimation of the covariance matrix. The use of a non-outlier-robust covariance technique will ruin the ICA estimation, and lead to ICA algorithms (depending on the optimization technique) incapable of estimating the optimum demixing matrix. For example, the RADICAL algorithm assumes whitened data, and thus only searches for a rotation matrix to separate the sources. Unfortunately, if improper whitening occurs, then it is impossible to find a rotation

matrix that would produce the best results. Barnett and Lewis [6] discuss a number of techniques to estimate covariance matrices robustly.

2.5 Summary

This section has introduced the ICA algorithms studied in this thesis. The separation principles and implementation details of each algorithm were given, and are critical for understanding the outlier sensitivity of each algorithm. Much emphasis was given to how each algorithm is broken down into a contrast function and an optimization technique. In addition, how the contrast function is essentially a search for a rotation that separates the signals. This perspective has give rise to two new outlier sensitivity measures, optimum angle of rotation error and contrast function difference, which are discussed in Ch. 3.

Chapter III

Measures of Outlier Robustness

This chapter presents outliers and outlier robustness measures for ICA. Section 3.1 begins by defining what an outlier is. Simply, outliers are defined as values that do not comply with the *probability density function* (PDF) of a signal. The remainder of the chapter covers outlier robustness measures for ICA. The measures presented are (i) the *Amari separation performance index* (API), (ii) the optimum angle of rotation error, (iii) the contrast function difference, and (iv) the influence function. The API measures the overall (implemented *contrast function* (CF) and optimization technique) separation performance of an ICA algorithm. The optimum angle of rotation error measures the sensitivity of the rotation angle calculated by a contrast function. The contrast function difference measures the shape change of a CF due to outliers. Finally, the influence function (IF), describes the standardized effect of an infinitesimal contamination at a point (outlier) on the asymptotic value of the contrast function. The outlier robustness measures were selected/designed such that they focus primarily on the sensitivity of a contrast function, not the optimization technique, to outliers. Together, these measures combined with simulations give an objective assessment of the outlier robustness of an ICA algorithm.

3.1 What is an Outlier?

As defined in Sec. 1.1.13, an outlier is a value of the signal which differs from a great majority of the other signal values. In signal processing, outliers fall under the category of probabilistic or structured. A probabilistic outlier is an observation that is outside the expected statistical model [52]. A structured outlier is an observation that does not follow the expected pattern. For example, a deviation in a linear regression model, or spike in a time series. This thesis focuses on probabilistic outliers.

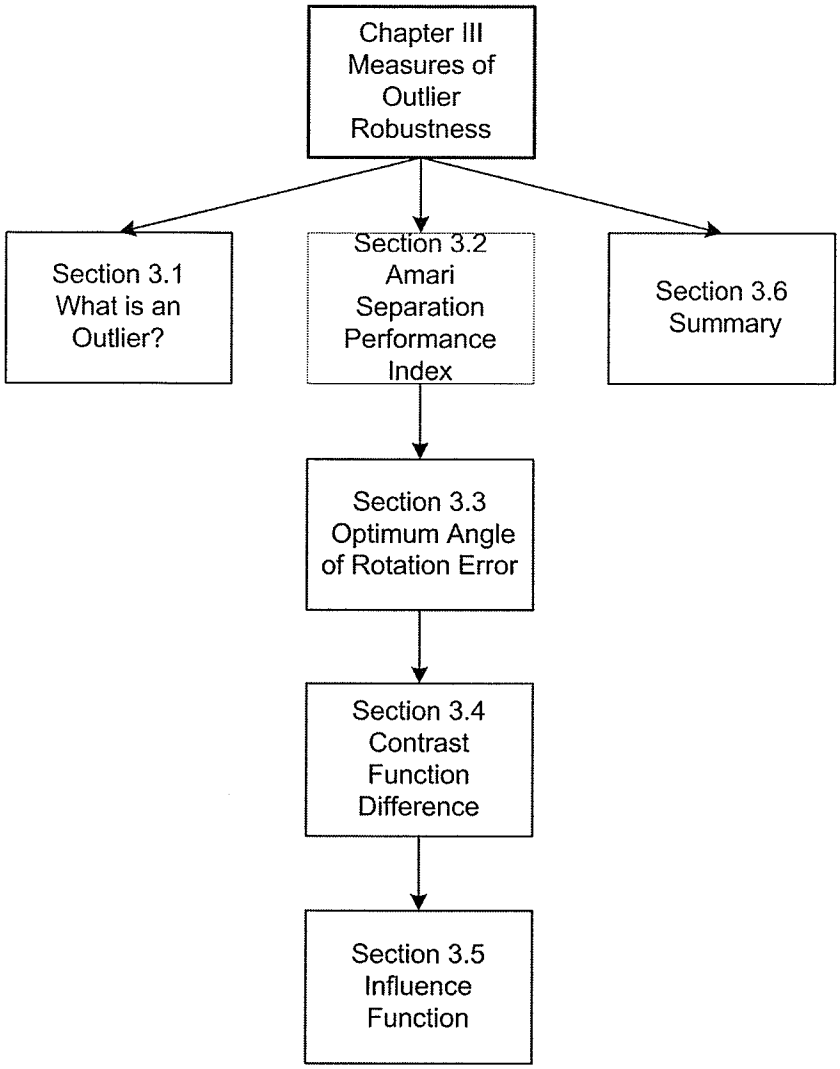


Fig. 3.1 Layout of Ch. III.

For this thesis an observation is categorized as an outlier if it is farther than 3 standard deviations from the mean of the observed distribution, or if it originated from an outlier generation model. An outlier generation model is some statistical distribution which samples are drawn and are considered outliers. Thus, outliers are defined as values that do not comply with the PDF of the observed signal.

Once outliers are known to exist, there are two extremes when dealing with them, rejection and accommodation [6]. Rejection involves removing observations from a data set prior to processing. Accommodation involves processing all data, but reducing the influence of the outliers. In this

thesis, the objective is to measure the outlier sensitivity of ICA algorithms which accommodate outliers.

The book by Barnett and Lewis has an in-depth coverage outlier detection and accommodation techniques [6]. A promising outlier accommodation technique for robust mean and variance estimation is the *minimum covariance determinant* (MCD) estimator [61], [62]. This method in conjunction with PCA may reduce the influence of outliers during whitening.

3.2 Amari Separation Performance Index

The *Amari separation performance index* (API) is a nonnegative measure of the matrix $\mathbf{V} = \mathbf{WA}$, where $\mathbf{W}$ is the estimated demixing matrix, and $\mathbf{A}$ is the original mixing matrix. This measure takes into account the scale and permutation invariance of ICA to gauge the accuracy of the demixing matrix [56]. The API is a non-blind measure as the original mixing matrix must be known. In a perfect simulation, $\mathbf{V} = \mathbf{DP}$, where $\mathbf{D}$ is some diagonal matrix and $\mathbf{P}$ is some permutation matrix, which results in an API of zero. Equation 3.1 is the API measure. The normalized version is shown in Eq. 3.2.

$$A_\epsilon = \sum_{i=1}^{I}\left(\sum_{j=1}^{J}\frac{|v_{ij}|}{max_{j'}|v_{ij'}|} - 1\right) + \sum_{j=1}^{J}\left(\sum_{i=1}^{I}\frac{|v_{ij}|}{max_{i'}|v_{i'j}|} - 1\right) \tag{3.1}$$

$$A_{\bar{\epsilon}} = \frac{A_\epsilon}{2IJ - I - J} \tag{3.2}$$

where I is the number of rows in $\mathbf{V}$, J the number of columns in $\mathbf{V}$, v_{ij} is the ijth element of $\mathbf{V}$, $max_{j'}|v_{ij'}|$ is the absolute value of the maximum value in row i, and $max_{i'}|v_{i'j}|$ is the absolute value of the maximum value in column j.

Figure 3.2 shows two demixed signals with normalized API ranging from 1 (worst) to 0 (best). These pictures demonstrate that an improving API does reflect a better separation performance.

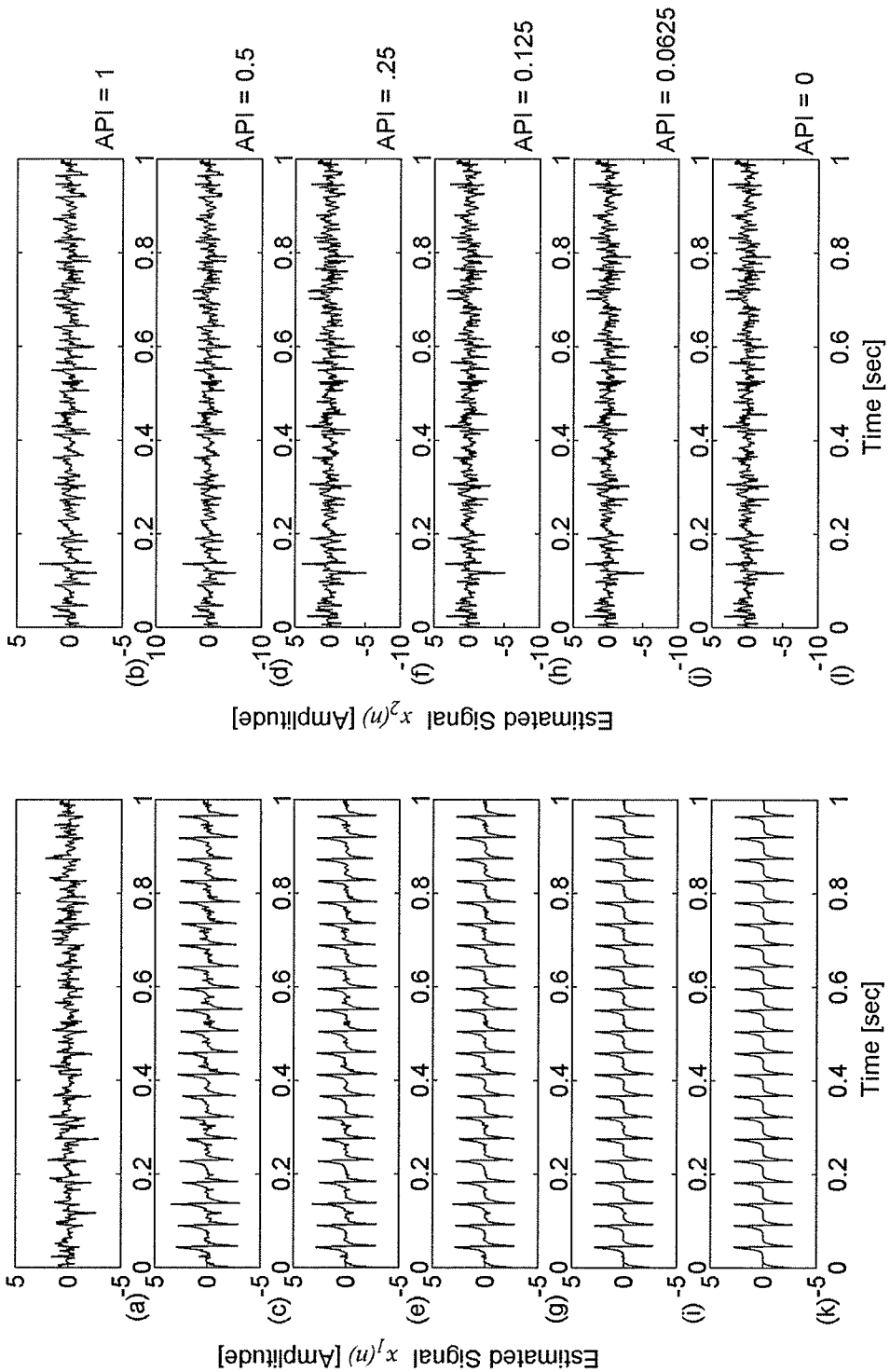


Fig. 3.2 Amari separation performance indices of mixed signals.

3.3 Optimum Angle of Rotation Error

The optimum angle of rotation error is a new non-blind measure for assessing the outlier sensitivity of an ICA algorithm. Recall from Sec. 1.2 and Sec. 2.4 that an ICA algorithm can be decomposed into whitening and a rotation matrix. The optimum angle of rotation error seeks to measure the difference between the optimum rotation angle required for source separation and the angle identified by an ICA contrast function for two sources [25].

In a non-blind simulation, two statistically independent sources are whitened and then mixed with a rotation matrix (optimum angle). Next, the contrast function of the ICA algorithm is solved with the data and various rotation matrices to find the angle it identifies for the separating rotation matrix (usually this would be the rotation matrix which results in the lowest value of the contrast function). The difference between the optimum angle and the estimated angle (absolute maximum error of 45 degrees) is defined as the optimum angle of rotation error. This technique removes the masking of a poor optimization technique that can occur with the API. Note improper whitening alters the mixing such that a perfect separation by a rotation matrix cannot be achieved.

Figure 3.3 shows two demixed signals with optimum angle of rotation errors ranging from 45 to 0 degrees. This picture demonstrates that a lower optimum angle of rotation error does reflect a better separation performance. In a simulation, it is of interest to see the change in optimum angle of rotation error with outlier free and outlier contaminated data.

3.4 Contrast Function Difference

The contrast function difference is a new measure for assessing the outlier sensitivity of an ICA algorithm. Rotation matrices between 0 and 90 degrees are used to solve a given CF and a set of observed samples (that have been whitened robustly). Next, an outlier is introduced into the observed data set, and the process is repeated. The results are normalized, aligned at the optima, and the difference is taken. This shape difference is defined as the contrast function difference curve. The area under the absolute value of the curve, divided by the maximum area of the curve, is the contrast function difference metric.

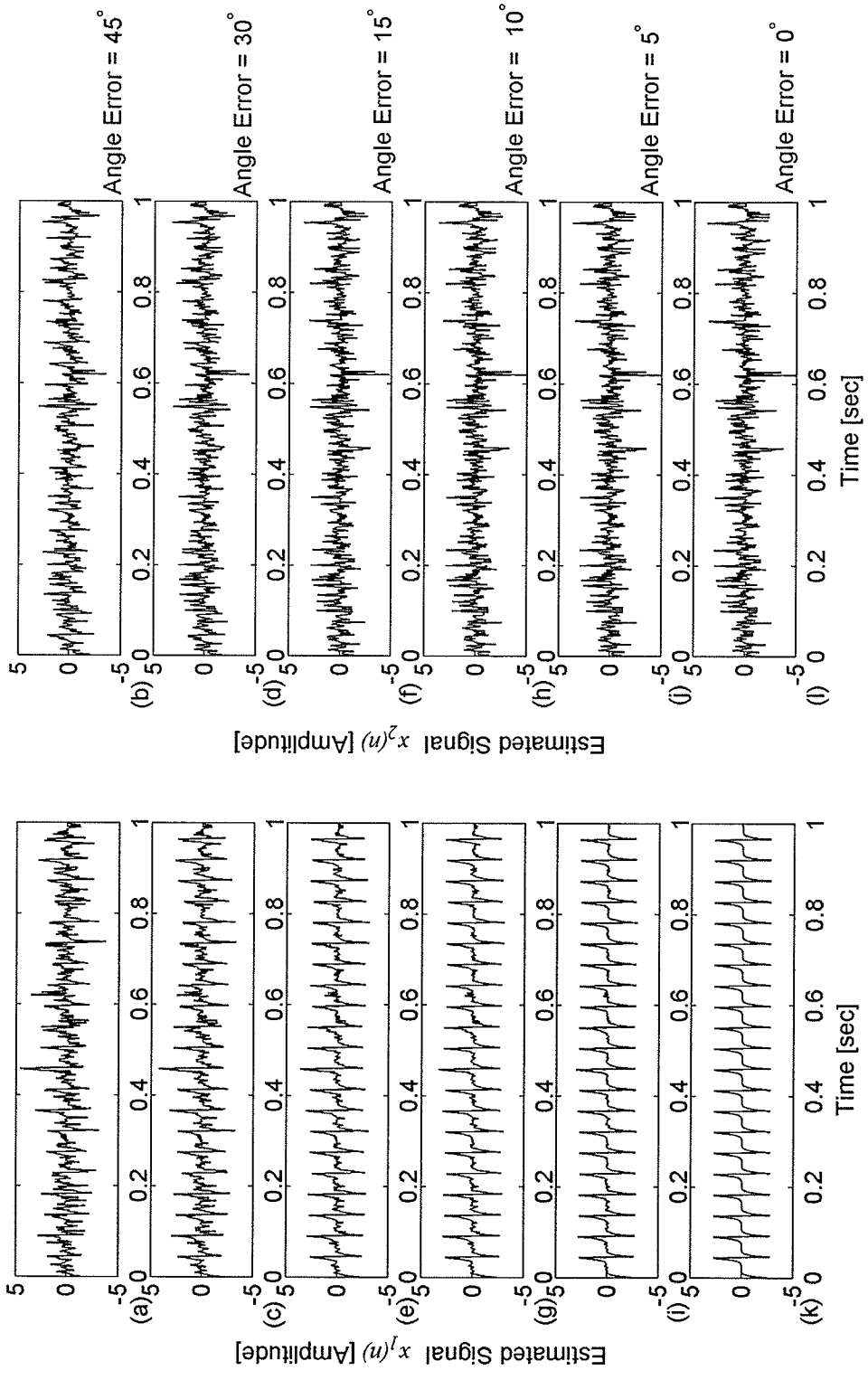


Fig. 3.3 Optimum angle of rotation error of mixed signals.

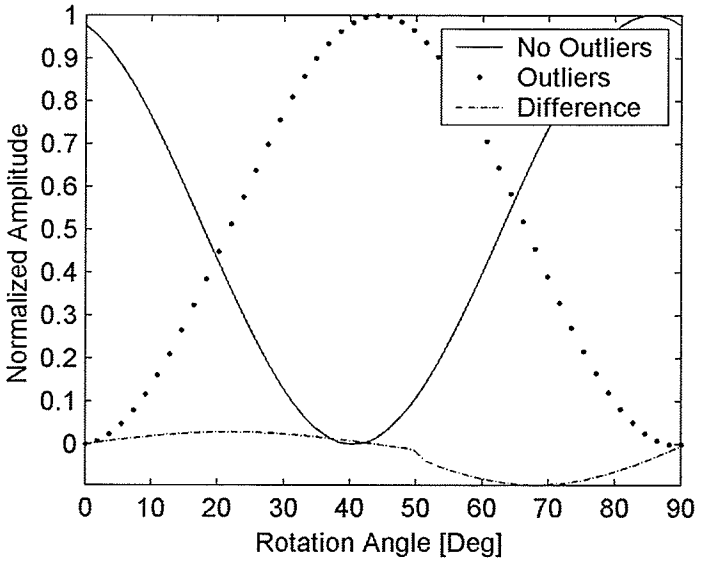


Fig. 3.4 Contrast function difference.

3.5 Influence Function

Robust statistics are in part the measures used to determine how to react to outliers, and whether or not they should be rejected, accommodated or be left alone. This section discusses the theory from the field of robust statistics called the influence function (related to the influence curve) [29]. An *influence function* (IF) is a heuristic tool used to formalize the bias on an estimator due to an outlier. In general, if the influence of an outlier is finite on an estimator, *i.e.*, contrast function, the estimator is known as B-robust. The focus of this section is an overview of the IF, and a discussion of IF analyses performed on the kurtosis measure, the FastICA algorithm and β-divergence algorithm. The IF is selected for study as it is the only systematic and mathematically rigorous method known to us for characterizing the impact of an outlier on a statistical estimator [33], [49].

The IF of estimator $T(\bullet)$ under a distribution model F and contaminant δ_χ (a probability measure which puts mass 1 at the point χ) is

$$IF(\chi, T, F) = \lim_{t \to 0^+} \frac{T((1-t)F + t\delta_\chi) - T(F)}{t} \tag{3.3}$$

The IF describes the standardized effect of an infinitesimal contamination at the point χ (outlier)

on the asymptotic value of the estimator. The gross-error sensitivity, sup || $IF(\chi; T, F)$ || taken over all χ where $IF(\chi; T, F)$ exists, measures the worst effect a small amount of contamination of fixed size can have on the value of the estimator. If the gross error is finite, the estimator T is said to be B-robust under F [29]. Ruppert [63] has published a well written influence function analysis of three kurtosis measures. This paper gives examples on how to calculate the influence function for an estimator. In addition, it alerts the reader to a kurtosis measure based on 2 interfractile ranges that is B-robust. This kurtosis measure should be considered in robustifying any ICA algorithm that uses a kurtosis estimate.

To determine the B-robustness of FastICA, Hyvärinen transformed approximate negentropy (Eq. 2.27) into the form of an M-estimator (see [29] for the definition of a M-estimator). He determined that the gross error of the M-estimator's influence function was not finite for all χ, and therefore FastICA was not B-robust. However, the study did reveal that the selection of $G(\bullet)$, such that it grows slowly with **wx**, reduces the estimator's sensitivity to outliers [33]. Minami and Eguchi followed a similar derivation based on M-estimators to show that ICA algorithms based on entropy maximization (FastICA), minimizing cross cumulants (JADE), and the natural gradient based approaches (Extended-Infomax) estimators are not B-robust. However, Minami and Eguchi did show that the β-divergence estimator is B-robust, but only if β is nonzero, and only for certain density functions p_i. No IF analysis has been performed on the RADICAL algorithm.

Additional tools from the field of robust statistics for investigating the effects of outliers and departures from an assumed distributional model on an estimator *breakdown point* (BP), *change-of-variance function* (CVF), and the minimax approach [29]. The BP is the maximum percentage of gross errors (outliers) an estimator can handle prior to becoming unreliable. The CVF shows the influence of outliers on the variance of the estimator. Finally, the minimax approach optimizes the worst that can happen over the neighbourhood of a distributional model, as measured by the asymptotic variance of the estimator. However, an in-depth analysis of ICA estimators by IFs, BPs and CVFs, similar to a location analysis found in [31] Ch. 11, has not been published to the authors knowledge. For a well-written overview of RS see [29] Ch. 1 and 8, [32] Ch. 1, and [65].

3.6 Summary

This chapter has presented the concept of outliers and outlier robustness measures for ICA. The Amari separation performance index is an overall outlier sensitivity measure when used in non-blind simulations. It measures the effect of outliers on the contrast function and optimization technique. The optimum angle of rotation error measures the influence of outliers on the rotation angle a contrast function identifies for separation. It measures the influence of an outlier on the contrast function alone, and removes the impact an outlier would have on whitening or optimization. The contrast function difference is another measure for the influence of outliers. It measures the overall effect of an outlier on a contrast function shape. Finally, the influence function characterizes the impact of an outlier on an estimator (contrast function). Previous analyzes, have discovered that the β-divergence algorithm is B-robust, while the FastICA, Extended-Infomax, and JADE algorithms are not. Together these tools combined with simulations is the basis for determining the outlier robustness of an ICA algorithm.

Chapter IV

Design of Experiments

This chapter presents the design of experiments for measuring the outlier sensitivity of the FastICA, Extended-Infomax, JADE, RADICAL and β-divergence algorithms. The chapter begins with a discussion on verifying the implementation of the β-divergence algorithm. This verification is necessary to confirm that this implementation matches published theory. In addition, it provides quantitative separation performance results to allow a comparison to the implementation. Next, Sec. 4.2 describes a simulation consisting of mixing samples drawn from pairs of 1-dimensional densities, introducing outliers, and measuring the Amari separation performance of the algorithms. This simulation is based on simulations by Miller *et. al* [42]. The intent of this experiment is to measure the overall outlier robustness of an ICA algorithm (contrast function and optimization technique) for a variety PDFs. The focus of Sec. 4.3 is measuring the optimum angle of rotation error and contrast function difference of the algorithms. The purpose of this final experiment is to measure the outlier sensitivity of the algorithms contrast function.

An important concept that is brought up during this chapter is the unbiased optimization landscape. A simple comparison of the separation performance between ICA algorithms is unfair if one algorithm is able to search a larger optimization space than the other. Certain ICA algorithms maybe dismissed due to a poor choice of an optimization technique rather than due to the contrast function. This thesis seeks to level the playing field by only allowing rotational matrices to mix the sources. The constraint allows for an unbiased experiment since all of the ICA algorithms studied can search the rotation optimization space.

Together these experiments provide a benchmark for the separation performance of ICA algorithms with outliers, and metrics for gaining insight on the outlier robustness of the algorithms.

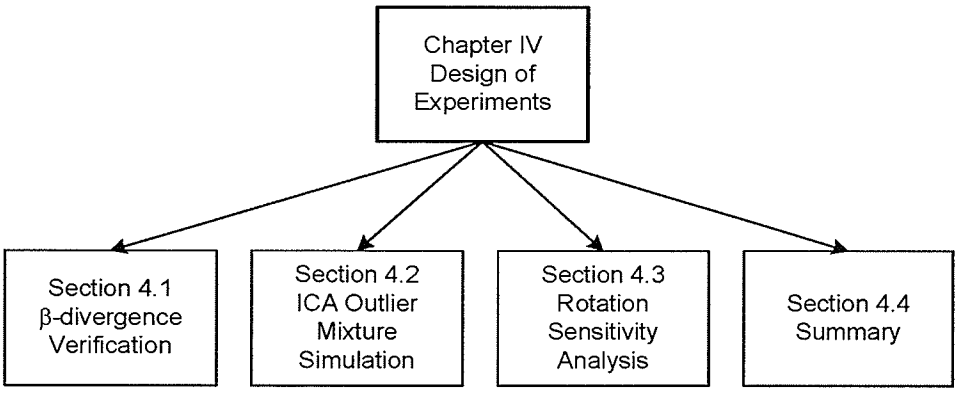


Fig. 4.1 Layout of Ch. IV.

4.1 β-divergence Verification

The section contains the design of an experiment to verify the β-divergence implementation described in Ch. 2. This verification is necessary to confirm that the implementation matches published theory. An additional objective is to provide quantitative separation performance results to allow a comparison to the implementation (which is not possible with the original paper by Minami and Eguchi [49]).

The criteria selected to meet these objectives are divided into 3 categories, (i) contrast function verification, (ii) optimization verification, and (iii) separation performance confirmation.

4.1.1 Contrast Function Verification

Two tests are selected to verify the correct implementation of the β-divergence contrast function (Eq. 2.57).

The first test is intended to check (i) if the value of the contrast function is stable for a mixture of Gaussian distributions, and (ii) converges to fixed values as β decreases from 1 to 0. For this test, samples are drawn from two 1-dimensional Gaussian distributions, and are mixed with rotation matrices between 0 and 90 degrees. Theory predicts the value of the contrast function should be fixed for a specific β, regardless of mixing by a rotation matrix. Since we are dealing with a mixture of two Gaussian distributions, any rotation of the distributions will result in the same distribution. When

$\beta = 0$, the contrast function should equal the Kullback-Liebler divergence between the empirical distribution of the observations and the product of the marginal densities. When $\beta = 1$, the contrast function becomes a mean-squared error between the empirical distribution and the predicted distribution. For the distributions selected, KLD should be 0.0236 NATS, and the MSE should be $7.8x10^{-4}$. However, it is unlikely to obtain these exact numbers, as approximations have been made in the derivation of the β-divergence algorithm. Suffice to say, that the value of the contrast at $\beta = 0$ should be larger than the value at $\beta = 1$.

Pairs of 250, 1000, 5000 and 10000 samples are drawn from 1-dimensional Gaussian distributions each of zero mean and unit variance. The β-divergence parameters **W** and μ are set as identity and 0 respectively, with β set to range between 0 and 1. The reason for selecting these sample sizes is to eliminate effects from insufficient sampling. In addition, these sample sizes refer to typical sample sizes with signals that are stationary only over small periods of time, but non-stationary over a long period of time. For example, in voice processing, since speech can be considered stationary within 10 to 50 ms, then 44 *kilosamples per second* (ksps) produce a few hundred samples. In biomedical signal processing (*e.g.*, the electrocardiogram), 0.3 ksps are common.

The second test selects pairs of N = 250, 1000, 5000 and 10000 samples from 1-dimensional uniform distributions each of zero mean and unit variance. The μ is set as 0 and β is set to range from 0 to 1. Again, the contrast function should be equal to the KLD when $\beta = 0$. If **W** is selected as a rotation matrix ranging from 0 to 90 degrees, the contrast function should be a minimum near 0 degrees and a maximum 45 degrees later. See *BetaDCntrstFuncVerification.m* in Appendix B for the code.

4.1.2 Optimization Verification

To verify the optimization technique a simple algebraic contrast function is used. The objectives of the test are to (i) confirm the optimization technique minimizes the contrast, and (ii) the optimization technique if starting at the minimum it should stay at the minimum. Equation 4.1 is selected as the contrast function. Clearly, the minimum is obtained when **W** is identity and μ is a

zero vector.

$$C_o(\mathbf{W},\mu) = (W_{1,1} - 1)^2 + (W_{1,2})^2 + (W_{2,1} - 1)^2 + (W_{2,2})^2 + \mu_1^2 + \mu_2^2 \tag{4.1}$$

where $\mathbf{W}$ is a 2 x 2 matrix, $W_{i,j}$ is the i,jth element of the matrix, μ is a 2 x 1 vector, and μ_i is the ith element of the vector. See Appendix B for the Matlab code (*OptVer.m, OptVerCaller.m, OptVerDerv.m and OptVerLbeta.m*).

4.1.3 Separation Performance Confirmation

To confirm that the β-divergence implementation works, experiments from [49] paper are repeated. The data set is the same as published by Minami and Eguchi, originating from a common database for empirically testing and comparing various independent components algorithms by Fisher [22]. The three distributions considered are: (i) the gamma with outliers, (ii) exponential power with bivariate Gaussian, and (iii) exponential power with two different means. Dataset 1 consists of 100 pairs of independent random numbers from the gamma distribution with parameter 1.5 and 3, and 2 outliers at (25,25). Dataset 2 introduces 150 pairs of independent random numbers from the exponential power distribution, and 50 pairs of random numbers from the bivariate normal distribution with mean vector 0, variances 16, and correlation 0.8. The parameter value for the exponential power distribution is 1.25 for one source and 1.45 the other. Finally, Dataset 3 is 200 pairs of independent random numbers from the same exponential power distribution, but the first 150 pairs of are centred at the origin and the last 50 pairs are centred at (5,5).

The β values are selected to range from 0 to 1. Based on the results from Minami's and Eguchi's paper, the optimum performance should occur at $\beta = 0.14$ for data set 1, $\beta = 0.28$ for data set 2, and $\beta = 0.21$ for data set 3. The p_is are selected to match those used in Minami and Eguchi's simulations. In addition, they are selected to match the assumed Gaussianity of the unknown sources. Specifically, the p_is for Dataset 1 have super-Gaussian characteristics, while those for Dataset 2 and 3 have sub-Gaussian characteristics. The objective is to setup the implementation to perform optimally. The parameters selected are found in Table 4.1.

where c is selected such that the area under the p_i equals 1.

Table 4.1 Experiment setup parameters.

Dataset	Optimal β	$p_1(z)$	$p_2(z)$
1.	0.14	$c/\cosh(z)$	$c/\cosh(z)$
2.	0.28	$c/(-.25z^4)$	$c/(-.25z^4)$
3.	0.21	$c/(-.25z^4)$	$c/(-.25z^4)$

The remainder of the algorithm setup parameters deal with the BFGS and line search optimization technique. The algorithm is implemented to stop searching for an optimum when the stopping criterion is less than 0.01 degrees; *i.e.,* the angle of the solution vector between two iterations changes less than 0.01 degrees. This was chosen such that minimizes the biasing effect on the API. The remaining setup parameters are the max iterations allowed for each of the BFGS and line search. These are selected at 25 and 20 respectively. They are selected such that the stopping criterion is met rather than maxing out the iterations, thus the best performance is achieved rather than the most efficient. The API calculated from 10 runs of each dataset is to be reported.

All random numbers are generated using the random number generation algorithms found in Matlab v6.1. The following paper from MathWorks describes random number generation found in Matlab [48]. See *BetaDSepPerf.m, BetaDSepPerfDataSet1,2* and *3.m* in Appendix B for the code.

4.2 ICA Outlier Mixture Simulation

The objective of the ICA outlier mixture simulation is to benchmark the overall outlier robustness of ICA algorithms (contrast function and optimization technique) for a variety PDFs. The simulation consists of mixing samples drawn from pairs of 1-dimensional densities, introducing outliers, and then measuring the Amari separation performance of the algorithms. This simulation is based on experiments published by Miller *et. al* [42]. The experiment is broken up into two simulations; one outlier free and the other with outliers. Figures 4.4 and 4.5 depict the flow of data for each simulation.

4.2.1 Simulation 1 Setup

The objective of simulation 1 is to benchmark the separation performance of optimally performing ICA algorithms in an unbiased optimization landscape for a variety of PDFs. Optimally performing implies the ICA algorithms are setup such that they perform optimally for the given data set. For example, using apriori information, such as the super- or sub-Gaussianity of the source signals, to configure parameters for the algorithm. An unbiased optimization landscape is created by only allowing mixing matrices that are rotations. All of the ICA algorithms studied are able to search the rotation landscape for a solution. The reason the ICA algorithms are setup to perform optimally is to benchmark the upper bound performance of the algorithm given ideal conditions.

Figure 4.4 shows that samples are drawn from two 1 dimensional PDFs, each of zero mean and unit variance, to produce the source signals $\mathbf{s}_1(n)$ and $\mathbf{s}_2(n)$. These signals are then mixed by a rotation matrix $\mathbf{A}$ to produce the observed signals $\mathbf{x}_1(n)$ and $\mathbf{x}_2(n)$. Finally, an ICA algorithm operates on the data in a batch format to produce estimates of the sources $\mathbf{y}_1(n)$ and $\mathbf{y}_2(n)$. The sample sizes to be studied are 250, 500 and 1000. These sample sizes are selected (i) to see the effects of sample size, and (ii) to match the sizes used by Miller. In addition, they are typical sample sizes for biomedical signals that are stationary over short time periods, but non-stationary otherwise. The PDFs selected for the simulation are shown in Fig. 4.2. The parameters and code for the densities are found in Appendix B (*plotpdfs.m* and *genRndData.m*). All the densities are adjusted to have 0 mean and unit variance prior to mixing.

Figure 4.2 shows the PDFs with their kurtosises, and Fig. 4.3 shows the histogram of samples generated from this distributions. Table A.1 in Appendix A lists the empirical statistics of these samples. These PDFs are selected because they allow a direct comparison of the results of this simulation to published work, and because they have a variety of higher-order statistics. Both zero and non-zero skewness exist due to the symmetric and non-symmetric distributions. In addition, the kurtosises range from negative to positive, giving sub to super-Gaussianity distributions. Finally, these PDFs are selected for benchmarking because of their prevalence in nature and biological processes.

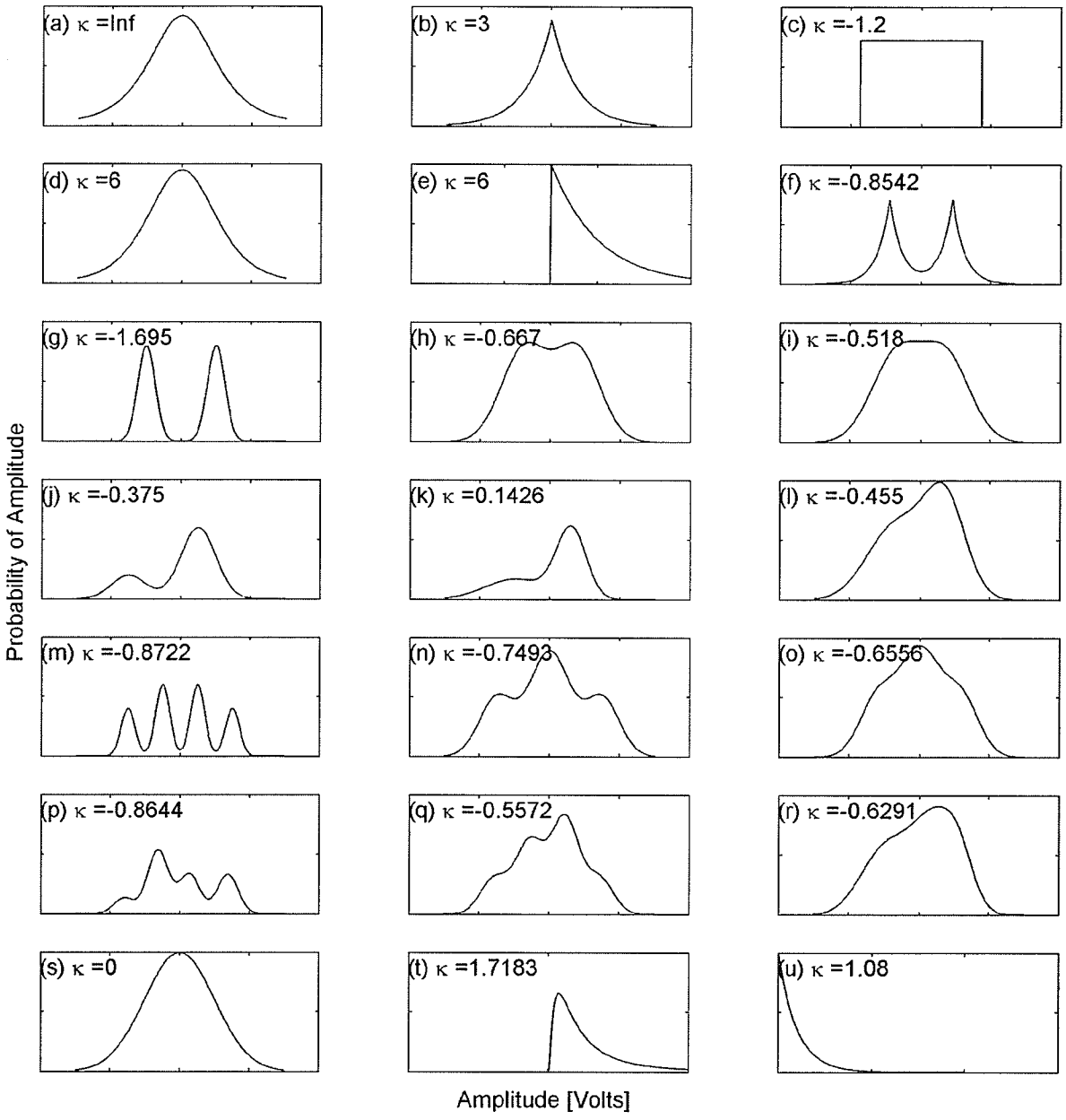


Fig. 4.2 Probability density functions of zero mean, unit variance and their respective theoretical kurtosises. (a) Student-t 3 degrees of freedom, (b) double exponential (Laplace), (c) uniform, (d) Student-t 5 degrees of freedom , (e) exponential, (f) mixture of 2 double exponentials, (g)-(h)-(i) Symmetric mixture of 2 Gaussians: multimodal, transitional, unimodal, (j)-(k)-(l) asymmetric mixture of 2 Gaussians: multimodal, transitional, unimodal, (m)-(n)-(o) symmetric mixture of 4 Gaussians: mutlimodal, transitional, unimodal, (p)-(q)-(r) asymmetric mixture of 4 Gaussians: multimodal, transitional, unimodal, (s) normal, (t) log-normal, (u) Pareto.

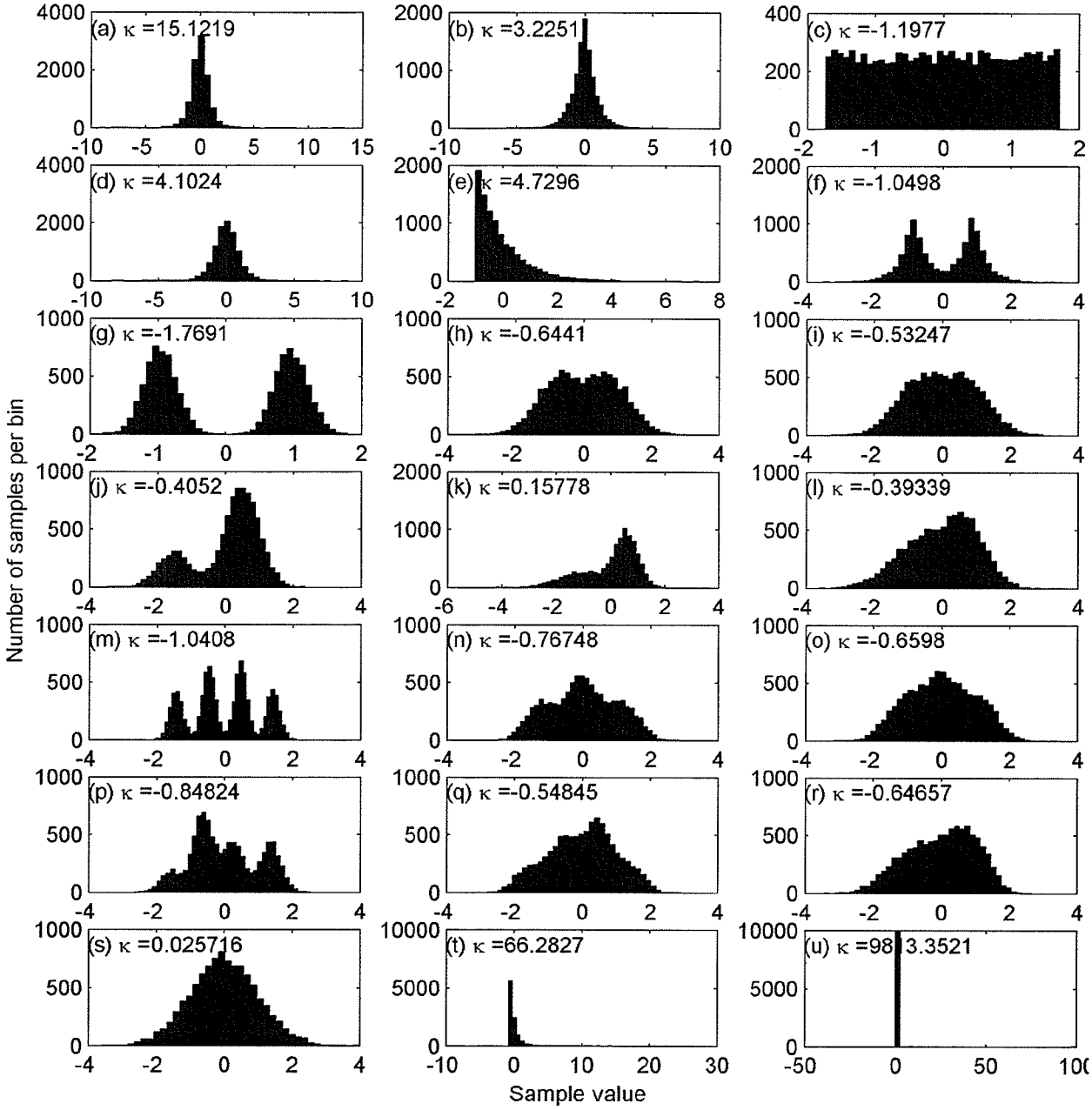


Fig. 4.3 Histogram of 10000 samples from probability density functions of zero mean, unit variance and their respective calculated kurtosises.(a) Student-t 3 degrees of freedom, (b) double exponential (Laplace), (c) uniform, (d) Student-t 5 degrees of freedom , (e) exponential, (f) mixture of 2 double exponentials, (g)-(h)-(i) Symmetric mixture of 2 Gaussians: multimodal, transitional, unimodal, (j)-(k)-(l) asymmetric mixture of 2 Gaussians: multimodal, transitional, unimodal, (m)-(n)-(o) symmetric mixture of 4 Gaussians: mutlimodal, transitional, unimodal, (p)-(q)-(r) asymmetric mixture of 4 Gaussians: multimodal, transitional, unimodal, (s) normal, (t) log-normal, (u) Pareto.

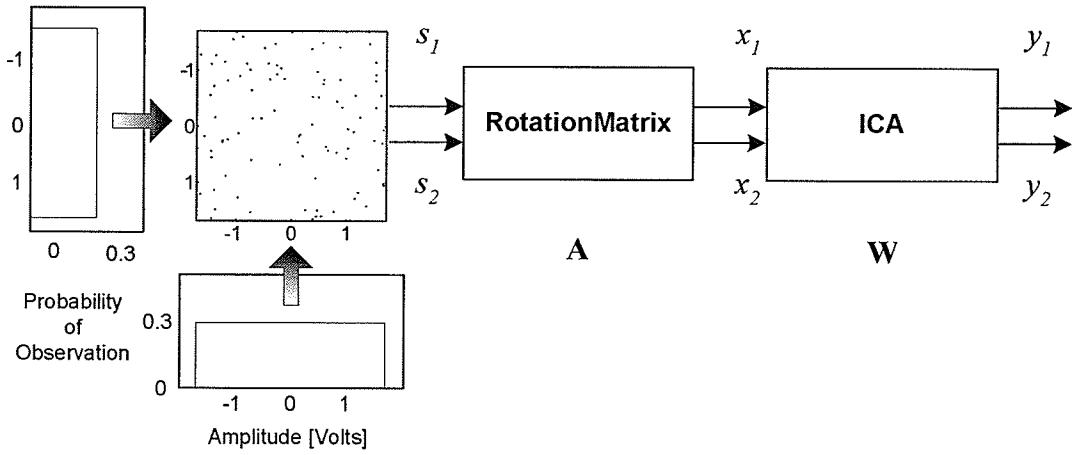


Fig. 4.4 Mixture simulation.

4.2.2 Simulation 2 Setup

Simulation 2 consists of the same elements as simulation 1, except that outliers are introduced after mixing, as shown in Figure 4.5. Outliers are introduced after whitening and mixing to create an ideal situation in which perfect whitening is achieved. A value of +/- 3, +/- 5 or +/- 7 (each selected with probability 0.5) is added to 1 data point, 0.5 % or 1 % of the data samples. This is to replicate the simulations performed by Miller. The points selected should be sufficient to rank the outlier robustness of the ICA algorithms as demonstrated in the results of Miller's paper. Selecting outliers from the 3rd, 5th and 7th standard deviation is done to determine a lower bound on the standard deviation upon which outliers have an impact on the algorithm.

See *xxx_MixSim.m*, where xxx is the density name, in Appendix B for the code to implement Simulation 1 and 2. Within each *xxx_MixSim.m* is *xxx_Calc.m*, which contains the list of .m files that call the ICA algorithm implementations.

4.2.3 ICA Algorithm Setups

The following sections discuss the setup of each ICA algorithm for the experiments to be conducted. Specifically, the sections discuss parameter selections and modifications of the algorithms.

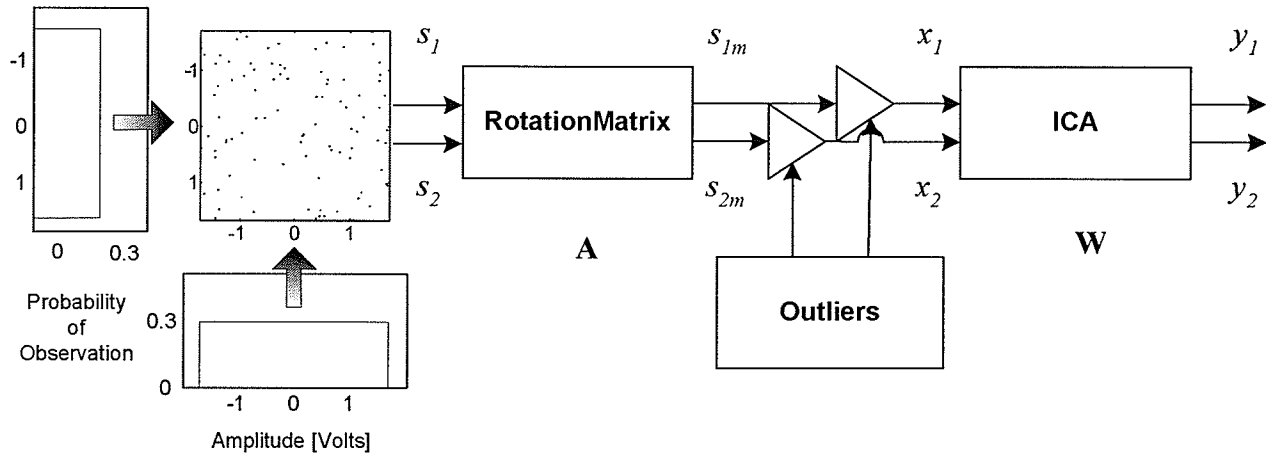


Fig. 4.5 Outlier contaminated mixture simulation.

4.2.3.1 FastICA Algorithm Setup

The FastICA algorithm is implemented in Matlab by Gavert *et. al*, and has a number of parameters that must specified in order to function [28]. First, the implementation allows a precalculated whitening matrix to be specified as an input parameter. Thus, an identity matrix is specified as the whitening matrix. Symmetric decorrelation of the estimated sources is selected to minimize cumulative errors. In addition, the stabilized version of the algorithm is selected to optimize separation performance while sacrificing computation efficiency. The step size "mu" for the the algorithm is set to 1. The stopping criterion is set at 0.0001. This ensures that the bias on the API is insignificant if convergence is reached. Should the stopping criterion not be met, a maximum iteration count of 100 is set. However, as long as the stopping criterion is met, the convergence related parameters should not have an impact on the separation performance. Finally, to ensure unbiased results for each simulation the elements of the initial demixing matrix are drawn from a uniform random distribution. This is followed by making the matrix positive definite to ensure the matrix is nonsingular. The remaining parameter to be specified is the non-linearity for the negentropy approximation. Since all 4, *pow3, tanh, gauss* and *skew*, have a major impact on the contrast function, all are selected for use. Thus, experiments with 4 FastICA algorithms shall be conducted.

4.2.3.2 Extended-Infomax Algorithm Setup

The EEGLAB toolbox is the source of the Extended-Infomax algorithm used for the thesis [47]. The algorithm is modified to prevent whitening of the input data. In addition, two variants of the algorithm are created, one with the sign of kurtosis calculated, the other where the sign is provided as apriori knowledge. All of the algorithm parameters (blocksize, learning rate, learning factor, momentum constant, number of iterations and block size for kurtosis estimation) relate to optimization properties, and are left as default values. However, if the algorithm has not sufficiently converged, the number of iterations shall be increased.

4.2.3.3 JADE Algorithm Setup

The JADE algorithm implementation is by Cardoso [11]. This is modified not to whiten the incoming signals. There are no other parameters for this algorithm.

4.2.3.4 RADICAL Algorithm Setup

The Matlab implementation of RADICAL is by Miller [41]. The algorithm is modified in two ways for the experiments. First, whitening is disabled. Second, the resolution of the rotation angles the algorithm searches for a solution between -45 and 45 degrees is changed from 90/150 degrees to 90/300 degrees to minimize the bias on the API.

4.2.3.5 β-Divergence Algorithm Setup

The β-divergence algorithm selected for use is by the thesis author [27]. There are a few contrast function and optimization related parameters that must be set. First, the assumed PDF of the sources must be specified. There are two PDFs possible, one for subGuassian densities and one for superGaussian densities. The most appropriate will be chosen such that the algorithm performs optimally. The second major contrast function related parameter is the β value. As this is a difficult parameter to select, a range of βs between 0 and 1 are to be tested. Finally, the remaining parameters are those dealing with convergence. The stopping criterion is set to 0.01. This is to ensure a minimal bias on the API. The max iterations for the BFGS and linesearch is set to 25 and 20 respectively.

This is to ensure the stopping criterion is met. However, the optimization parameters should not have an impact on the final result.

4.2.4 Experiment Analysis Setup

All experiments shall have the API measured. The results section shall perform a statistical analysis of the API to provide meaningful conclusions.

4.3 Rotation Sensitivity Analysis

The objective of this experiment is to measure the outlier sensitivity of ICA contrast functions using the rotation angle error and contrast difference measures. The interest of this experiment is the outlier sensitivity of the contrast function to outliers, and not the ability or inability of the ICA algorithm to separate the mixture. ICA contrast functions are implemented in Matlab to support this experiment. The dataset is the same as discussed in Sec. 4.2.

4.3.1 Simulation Setup

An outline of the simulation is depicted in Fig.4.6. Data is drawn from pairs of 1-dimensional, unit variance, zero mean densities and then mixed with rotation matrices between 0 and 90 degrees. The intent is to find the angle that minimizes the contrast function. Next, outliers (as described earlier) are introduced and the new angle that minimizes the contrast is found. The difference is the optimum angle of rotation error. Next, the curves generated to find the angle which minimizes the contrast are used to determine the contrast function difference. The curves with and without outliers are aligned at their minimums, normalized (0 to 1) and then subtracted. The area underneath the absolute value of curve divided by the maximum possible area under the curve is taken as the contrast function difference. The maximum area underneath the curve is 90, considering a rotation angle between 0 and 90 degrees, and a normalized contrast function difference always equal to 1.

4.3.2 Contrast Function Implementation

The following lists the contrast functions that are implemented in Matlab for the respective ICA algorithm. The FastICA contrast function implemented is Eq. 2.27 using the 4 nonlinearities *tanh*,

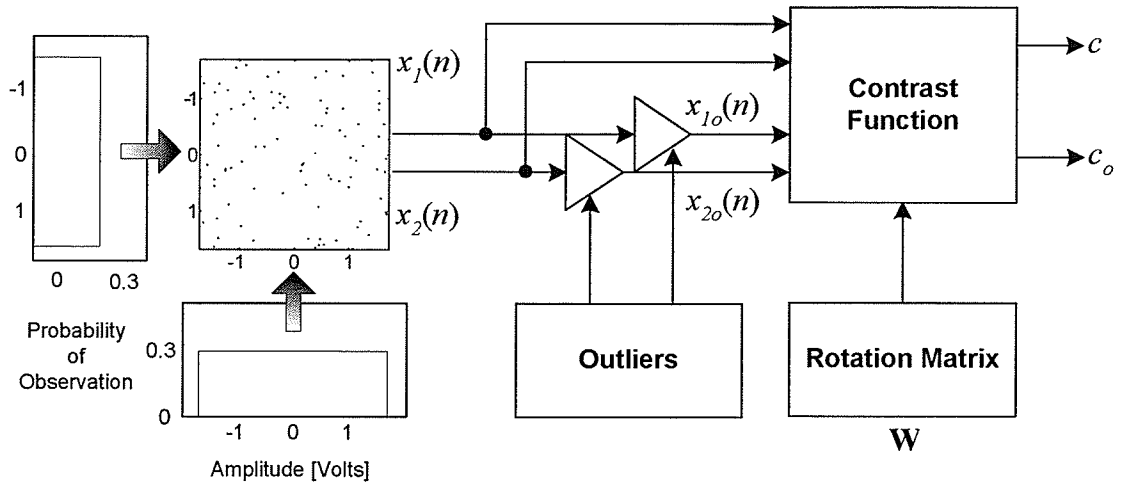


Fig. 4.6 Rotation sensitivity simulation.

gauss, *skew* and *pow3*. These contrasts have been modified with a negative sign, as we want to minimize the contrast. See *FastICA_Cntrst.m* in Appendix B for the code. The Extended-Infomax contrast function implemented is Eq. 2.32. It has been modified with a negative sign, as we want to find the minimum. See *Infomax_Cntrst.m* and *Infomax_Cntrst2.m* in Appendix B for the code. The JADE contrast function implemented is Eq. 2.49. See *JADE_Cntrst.m* in Appendix B for the code. The RADICAL contrast function implemented is Eq. 2.51. See *Radical_Cntrst.m* in Appendix B for the code. The β-divergence contrast function implemented is Eq. 2.57. It has been modified with a negative sign as we want to find the minimum. The nonlinearities and β values for β-divergence are the same as in Sec. 4.2. See *BetaD_Lbeta.m* in Appendix B for the code.

4.4 Summary

This chapter presented experiments intended to reveal the outlier sensitivity of ICA algorithms. Important aspects of the experiments are (i) an unbiased optimization landscape, (ii) a separation performance upperbound, and (iii) a focus on the outlier sensitivity of the contrast functions. An unbiased optimization landscape allows a fair comparison of an ICA algorithms in an outlier contaminated situation as all of the algorithms have the potential to find the exact solution. Seeking a

separation performance upper bound given ideal conditions, shall demonstrate if the implemented algorithm is capable of achieving perfect separation. Finally, focusing on the contrast function removes the unnecessary complications of the optimization aspect of an ICA algorithm as it should not impact the outlier sensitivity of the algorithm. The results of the experiments should provide strong experimental evidence of which ICA algorithm is the most outlier robust. In addition, it should show what aspects of its contrast function make it outlier robust, and give insight on how to design outlier robust ICA algorithms.

Chapter V

Experimental Results and Discussion

Chapter 5 presents the results of the outlier simulations described in Ch. 4. Figure 5.1 shows the layout of the chapter. The chapter begins with Sec. 5.1 where the verification of the β-divergence algorithm is presented. This section discusses how experimental data produced by the algorithm implementation agrees with theory. However, the experiments did reveal that the implementation is sensitive to initial conditions, especially when dealing with mixtures of asymmetric distributions. This sensitivity required the development of a second β-divergence implementation where an exhaustive rotation search replaces the BFGS optimization technique. The results of the outlier mixture simulation are found in Sec. 5.2. In simulations with and without outliers, the β-divergence (version 2) algorithm had the lowest average API. The results of the rotation sensitivity analysis and contrast function difference (Sec. 5.3 and 5.4) showed that the β-divergence contrast function was the least sensitive to outliers. Finally, the chapter concludes with Sec. 5.5 where suggestions to improve the outlier robustness of the ICA algorithms are made.

5.1 β-divergence Verification

This section discusses the experiments used to verify the implementation of the β-divergence for BSS. The contrast function verification produced empirical evidence that contrast function implementation agrees with theory. The first test confirms that (i) the value of the contrast function is stable for a mixture of Gaussian distributions, and (ii) the result of the contrast function is larger when $\beta = 0$ then when $\beta = 1$. The second test supports the notion that a uniform distribution minimizes the contrast function with a 0 degree rotation matrix and maximizes the contrast function with a 45 degree rotation mixing matrix.

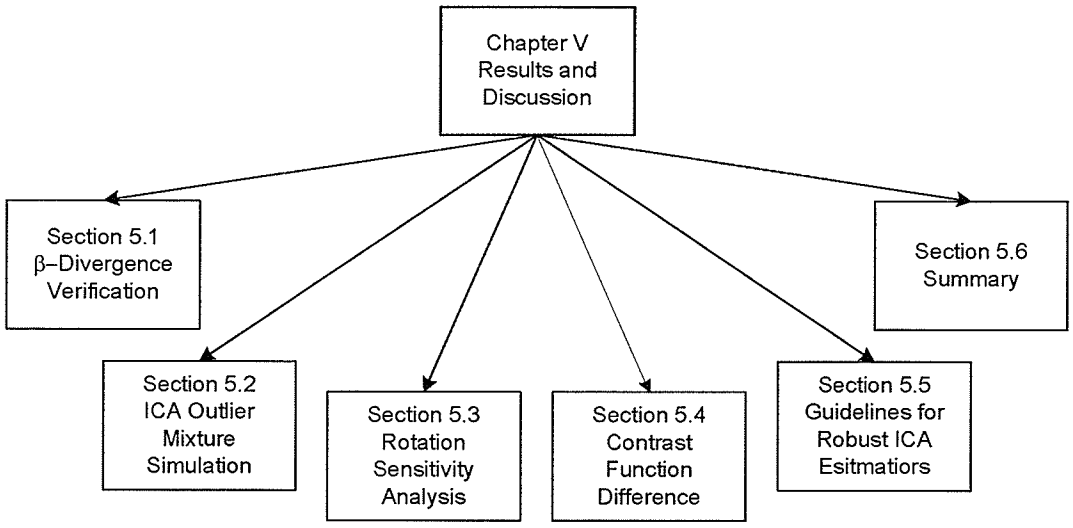


Fig. 5.1 Layout of Ch. V.

5.1.1 Contrast Function Verification

Figure 5.2 shows the value of the contrast function for a variety of sample sizes of Gaussian distributed data, rotation matrices, and values of β. The first aspect to make note of is the effect of sample size on the results of the experiment. The figure shows groups of lines for each β. The sample size had little effect on the solution to the contrast function. This result demonstrates the consistency characteristic of the algorithm. Next, notice the angle of the rotation mixing matrix had little effect on the result of the contrast function. The percent deviation (maximum value of the contrast minus the minimum of the contrast divided by the maximum times 100) for all of the lines does not exceed 0.05 percent. This agrees with the expected result. The numeric result of the experiment does not agree with the values calculated from Ch. 4, however the value of β at 0 is larger than at 0.99. When $\beta = 0$ the contrast equals 3.36, and 0.08 when $\beta = 0.99$. The factor of change is 42. The values calculated in Ch. 4 had a factor of change of 30. One reason for the difference is because in our implementation Eq. 2.59 was left as zero as it simplified the implementation.

Figure 5.3 shows the value of the contrast function for a variety of sample sizes of uniformly distributed data, rotation matrices and values of β. The first aspect to make note of is the effect

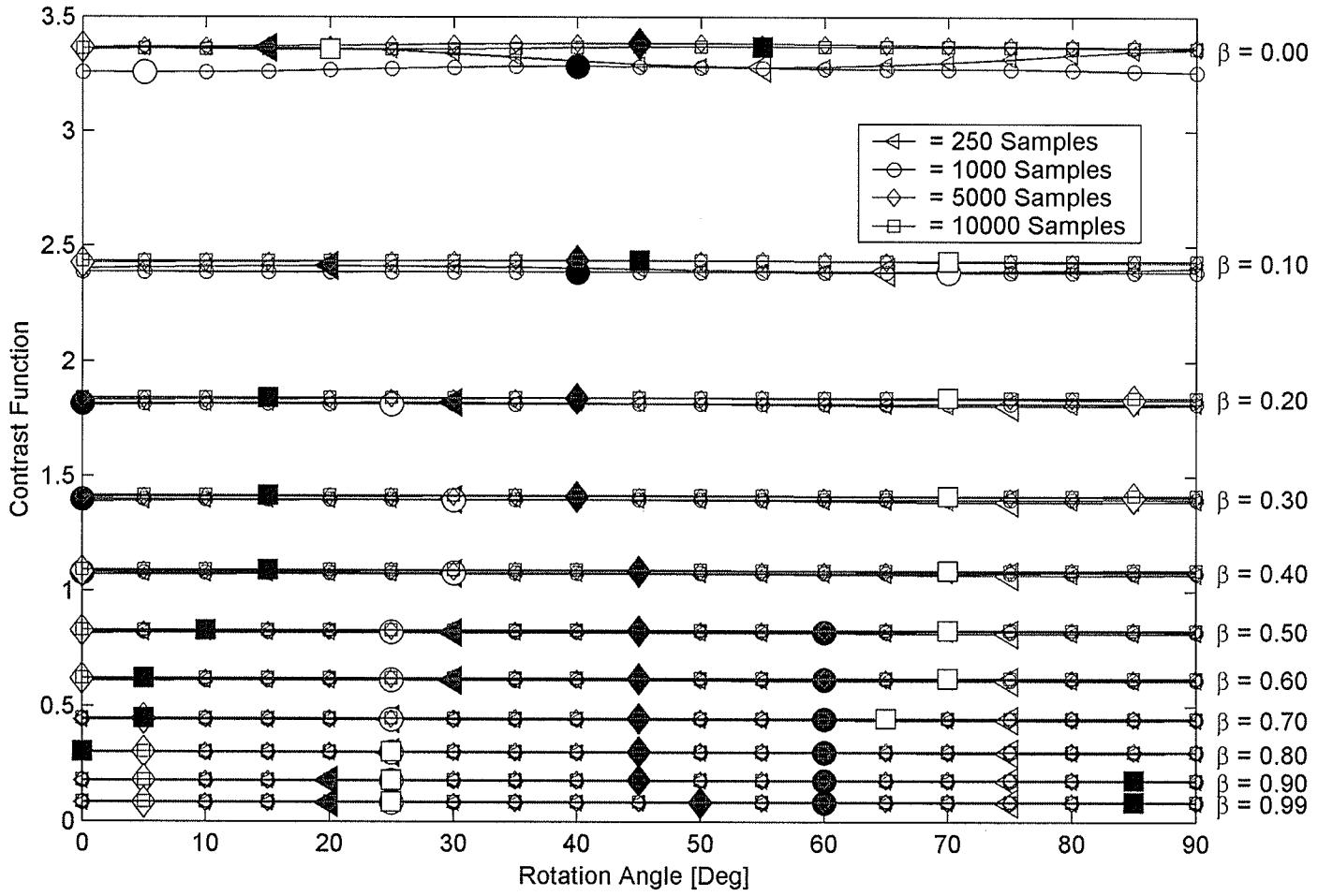


Fig. 5.2 Mixture of Gaussian Distributions with 250, 1000, 5000 and 10000 pairs of samples. The filled in symbol is the maximum value of the contrast, and the larger clear symbol is the minimum value of the contrast.

of sample size on the results of the experiment. The figure shows groups of lines for each β. The sample size had little effect on the solution to the contrast function. Next, notice the angle of the rotation mixing matrix had an important effect on the result of the contrast function. The contrast function was maximum at a rotation matrix of 45 degrees, and a minimum at 0 degrees. This agrees with the expected result, as a rotation matrix of 45 degrees makes the pairs of samples dependent on each other, while the 0 degree mixing matrix makes independent distributions. Thus, for these fundamental experiments, the implemented contrast function agrees with theory.

5.1.2 Optimization Technique Verification

The experiments confirmed that (i) the optimization technique minimizes the contrast, and (ii) the optimization technique stays at the minimum when starting at the minimum. A variety of starting conditions were selected, and each resulted in selecting $\mathbf{W}$ as identity and μ as a zero vector.

5.1.3 Separation Performance Verification

The separation performance verification experiment demonstrated that the β-divergence implementation has a low API when separating mixtures of sources contaminated with outliers. However, the implementation is sensitive to initial conditions (primarily the mean), and performs inconsistently for mixtures of asymmetric distributions. The sensitivity is because the contrast function landscape has a number of peaks and valleys, and depending on the starting position in the landscape, the optimization technique fails to find the global minimum. The reason the algorithm performs poorly for mixtures of asymmetric distributions is due to the hypothesized density being symmetric. Recall as β increases from 0 to 1, the contrast function becomes a mean squared error between the hypothesized density and the empirical density. In the experiment, only symmetric (sub and super-Gaussian) hypothesized densities were used, and thus as β increased the contrast attempted to find a rotation that skewed the distribution. Further investigation of this asymmetry result is required to confirm this reason as the optimization technique might have performed poorly with this new dataset.

Figures 5.4, 5.5, and 5.6 are typical plots of the results of our implementation for Dataset 1, 2

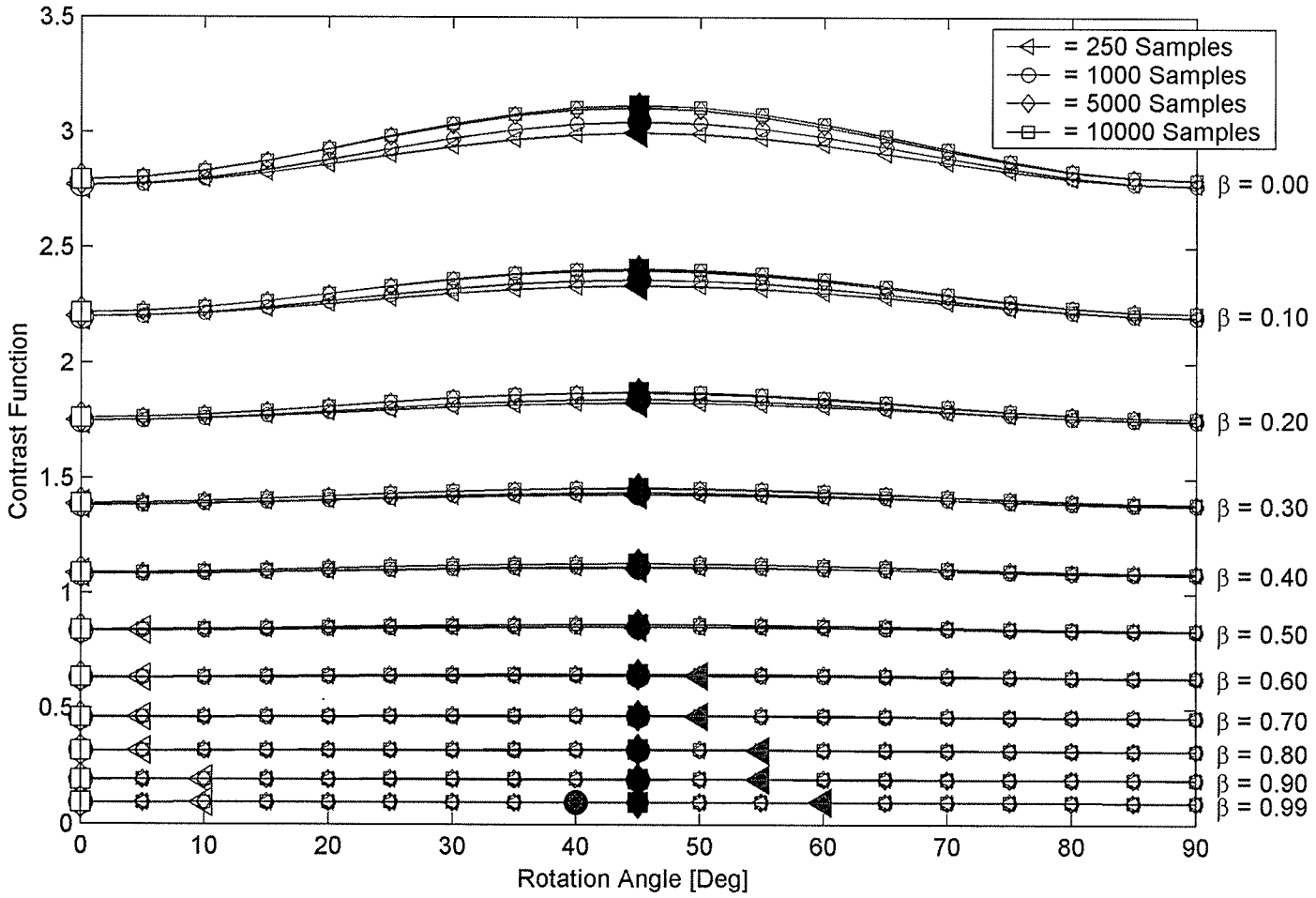


Fig. 5.3 Mixture of uniform distributions with 250, 1000, 5000 and 10000 pairs of samples. The filled in symbol is the maximum value of the contrast, and the larger clear symbol is the minimum value of the contrast.

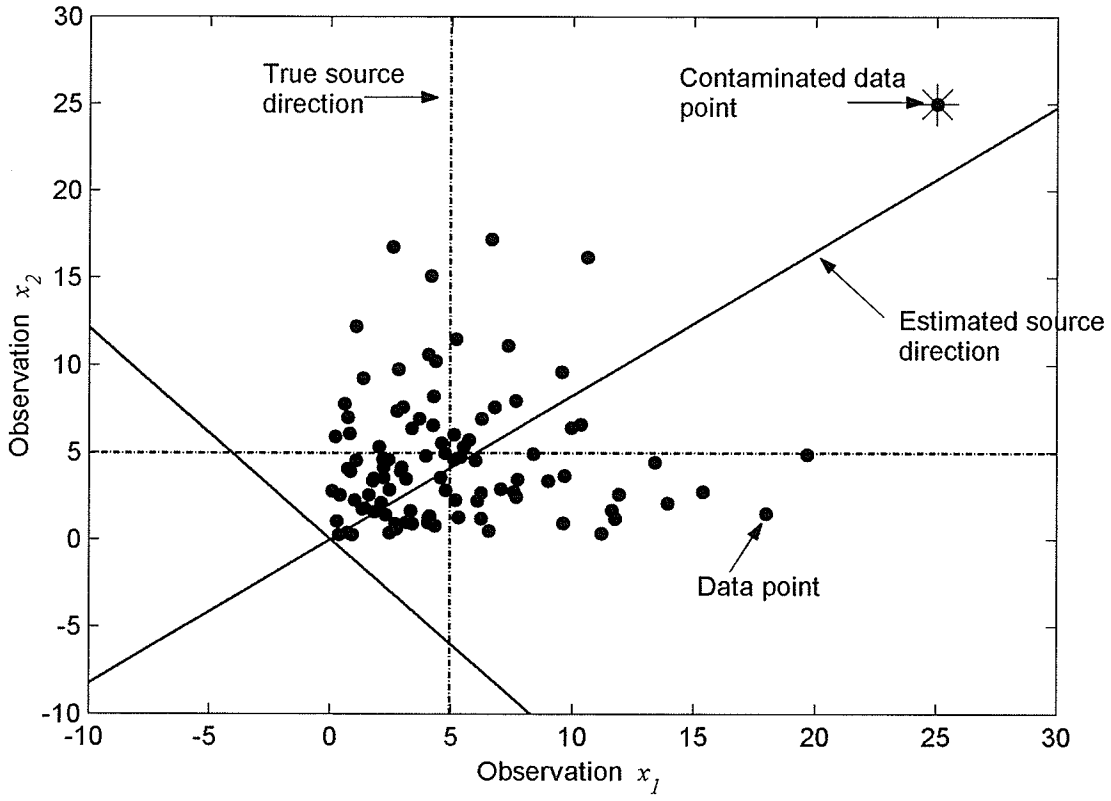


Fig. 5.4 Scatter plot of Dataset 1 where the dotted lines indicate the true directions of the independent sources, while the solid lines are the estimated directions of the independent sources using the β-divergence algorithm with $\beta = 0.25$. The solid dots are the uncontaminated data, while the starred points are the outliers .

and 3 with the parameter setup described in Ch. 4. The dotted lines indicate the true directions of the independent sources, while the solid lines are the estimated directions of the independent sources. The solid dots are the uncontaminated data, while the starred points are the outliers. Figures 5.7, 5.8, and 5.9 are typical plots of the results for a range of βs. Tables 5.1 and 5.2 contain the API for the plots presented. The experimental results show that the implementation of the β-divergence algorithm achieves similar separation performances (although not with the same βs) to those published by Minami and Eguchi except for Dataset 1. Unfortunately, Minami and Eguchi's paper did not present quantitative numbers on the separation performance of their implementation.

To investigate the poor results for Dataset 1, the contrast function curve for rotation matrices are

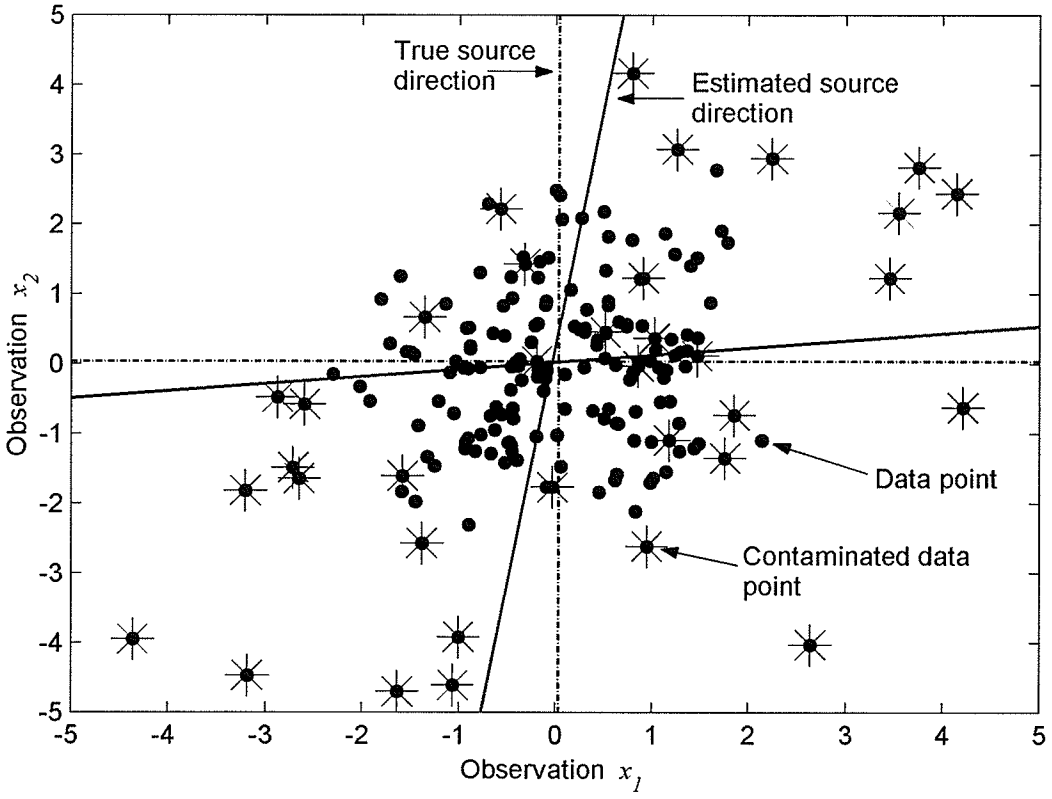


Fig. 5.5 Scatter plot of Dataset 2 where the dotted lines indicate the true directions of the independent sources, while the solid lines are the estimated directions of the independent sources using the β-divergence algorithm with $\beta = 0.25$. The solid dots are the uncontaminated data, while the starred points are the outliers .

Table 5.1 β-divergence verification performance for dataset 1, 2 and 3 with $\beta = 0.25$.

	Dataset 1	Dataset 2	Dataset 3
Mean initial normalized API	0	0	0
Mean final normalized API	0.901	0.073	0.035
The best normalized API	0.373	0.009	0.001
The worst normalized API	1	0.1862	0.087
Mean execution time [sec]	0.6398	1.4222	1.367

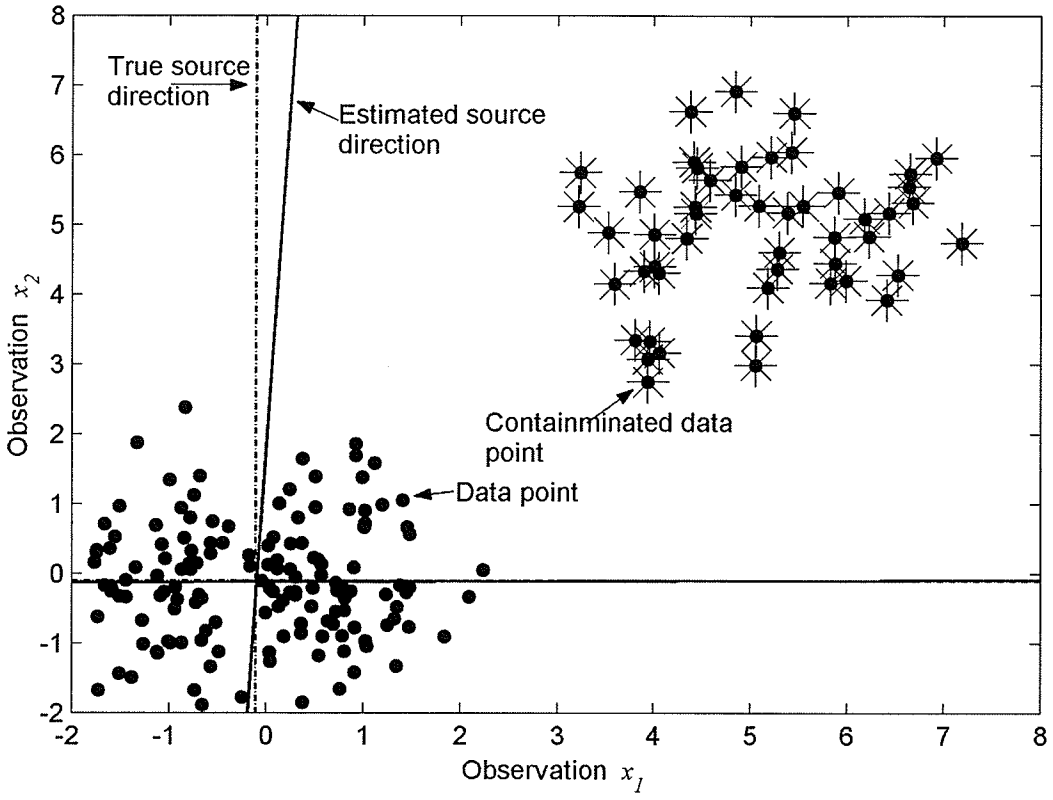


Fig. 5.6 Scatter plot of Dataset 3 where the dotted lines indicate the true directions of the independent sources, while the solid lines are the estimated directions of the independent sources using the β-divergence algorithm with $\beta = 0.25$. The solid dots are the uncontaminated data, while the starred points are the outliers .

Table 5.2 β-deference verification performance dataset 1,2 and 3 with β ranging from 0 to 0.99 .

β	Dataset 1		Dataset 2		Dataset 3	
	Initial API	Final API	Initial API	Final API	Initial API	Final API
0	0	0.642	0	0.175	0	0.550
0.12375	0	0.144	0	0.159	0	0.043
0.2475	0	0.741	0	0.0272	0	0.030
0.37125	0	0.657	0	0.035	0	0.007
0.495	0	0.175	0	0.004	0	0.003
0.61875	0	0.001	0	0.003	0	0.009
0.7425	0	0.000	0	0.0035	0	0.010
0.86625	0	0.000	0	0.006	0	0.010
0.99	0	0.000	0	0.006	0	0.008

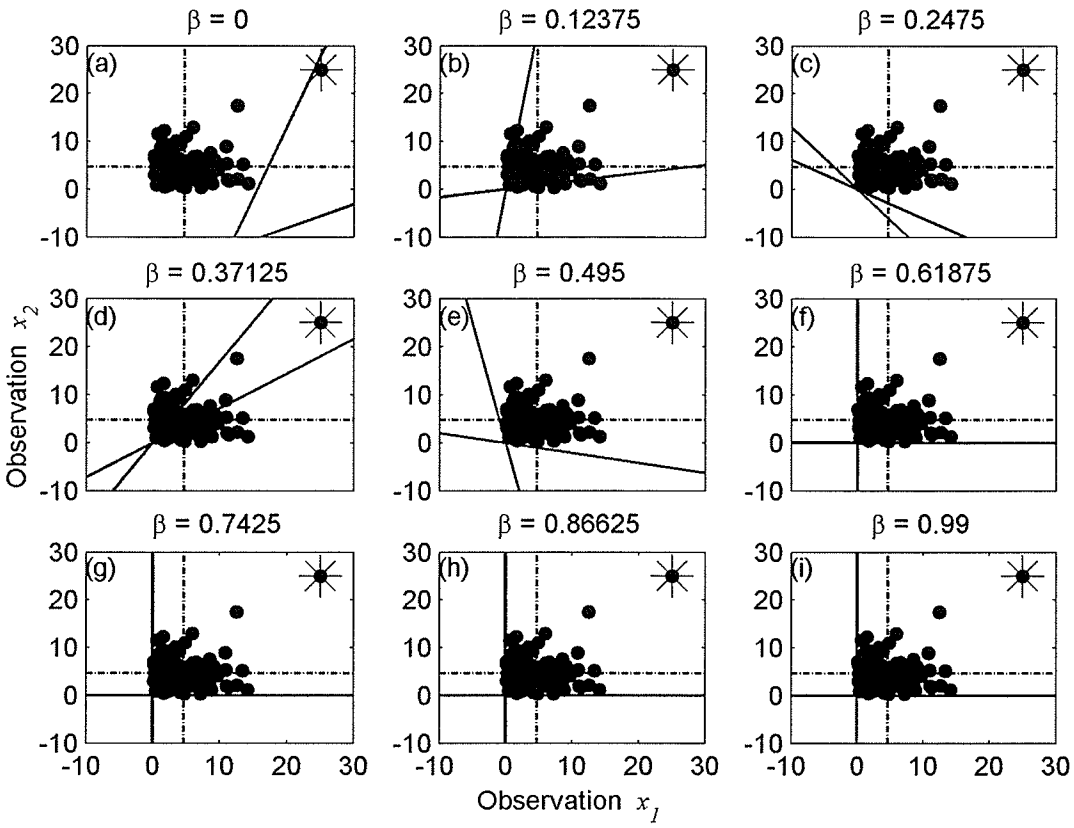


Fig. 5.7 Scatter plot of Dataset 1 where the dotted lines indicate the true directions of the independent sources, while the solid lines are the estimated directions of the independent sources using the β-divergence algorithm with β ranging from 0 to 0.99. The solid dots are the uncontaminated data, while the starred points are the outliers.

plotted for Dataset 1, 2 and 3 (Fig. 5.10, 5.11 and 5.12). Clearly the contrast function for Dataset 1 is not a minimum at 0 degree rotation matrix. Through experimentation it is determined that the non-zero mean of the dataset biases the contrast function. In Eq.2.58 when the mean is not near zero the implemented BFGS optimization technique is unable to iterate to find the global minimum, as the demixing matrix vector is in a very "flat" region. Thus, dataset 1 was adjusted to a zero mean and unit variance, and the experiment is re-run; Fig. 5.13 is the result. The results improved to where the API was near zero for most βs. Looking at the plots of the contrast functions, it seems the contrast functions can be minimized with a rotation matrix (although the mixing is a 0 degree mixing matrix).

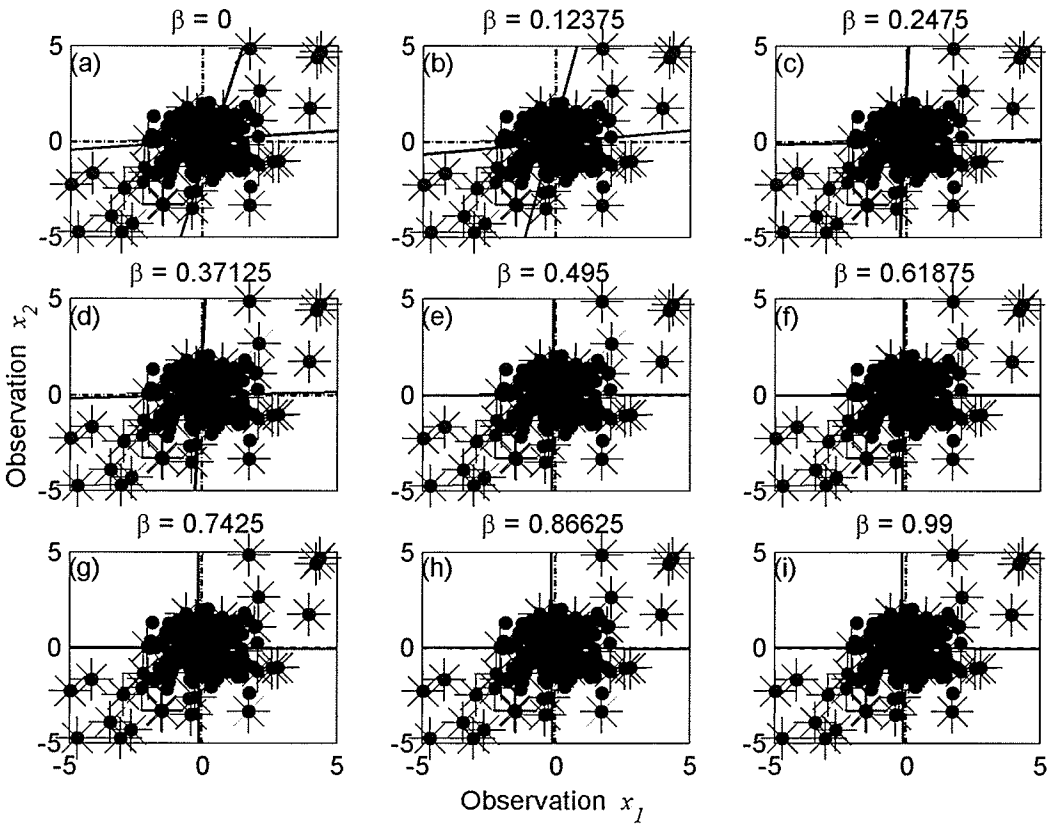


Fig. 5.8 Scatter plot of Dataset 2 where the dotted lines indicate the true directions of the independent sources, while the solid lines are the estimated directions of the independent sources using the β-divergence algorithm with β ranging from 0 to 0.99. The solid dots are the uncontaminated data, while the starred points are the outliers.

Thus, for the rest of the experiments a second algorithm for β-divergence is used. In β-divergence version 2, the BFGS optimization is replaced with an exhaustive rotation matrix search (the same as done by RADICAL), and a new requirement is that the data be zero mean and unit variance (as discussed in Sec. 2.4). See *BetaD_ICA2.m* in Appendix B for the code implementation. Figures 5.14, 5.15 and 5.16 show the results when using the β-divergence (version 2) ICA algorithm. Table 5.3 is the API for the experiment. Although the results improved, a drawback of the rotation search is an increased number of computations. The time to execute the algorithm increased 2-3 times as compared to the BFGS β-divergence algorithm.

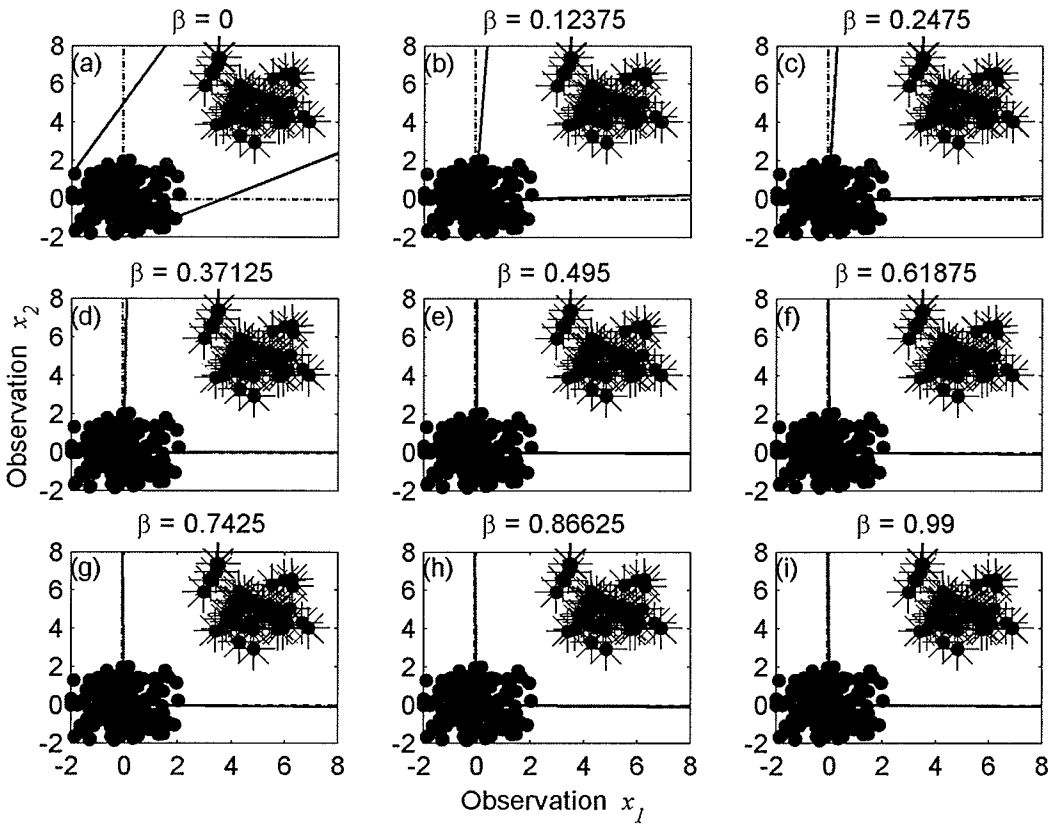


Fig. 5.9 Scatter plot of Dataset 3 where the dotted lines indicate the true directions of the independent sources, while the solid lines are the estimated directions of the independent sources using the β-divergence algorithm with β ranging from 0 to 0.99. The solid dots are the uncontaminated data, while the starred points are the outliers.

The limitations of both β-divergence algorithm implementations are the (i) processor intensive optimization, and (ii) ability to only handle the separation of two source signals.

In summary, the β-divergence (version 2) algorithm implementation has been verified to function correctly when the data is zero mean and unit variance.

5.2 Mixture Simulation Results and Analysis

Figure 5.17 shows the average API for all simulations performed. The outlier free simulations demonstrated that the β-divergence (version 2) algorithm had the lowest average API. However, RADICAL, JADE and FastICA (pow3, tanh and gauss nonlinearities) were within 0.05 API of

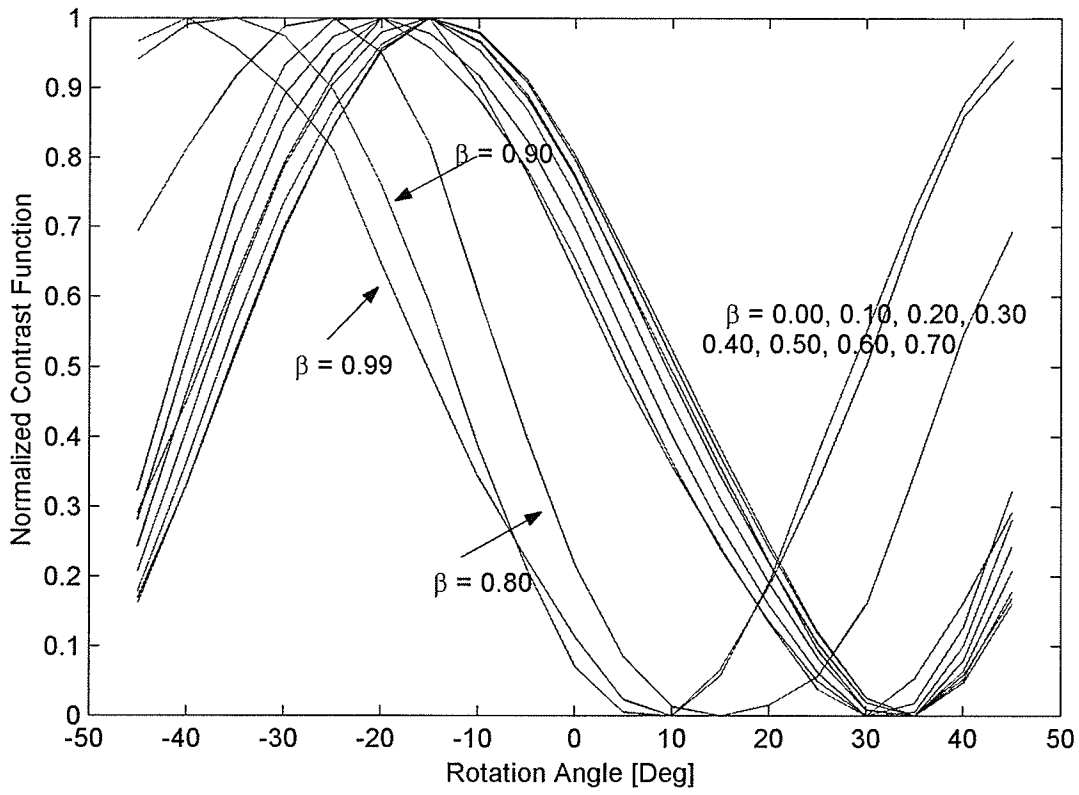


Fig. 5.10 Contrast function plot of Dataset 1 with rotation matrices between -45 and 45 degrees.

Table 5.3 β-divergence 2 verification performance dataset 1,2 and 3 with β ranging from 0 to 0.99 .

β	Dataset 1		Dataset 2		Dataset 3	
	Initial API	Final API	Initial API	Final API	Initial API	Final API
0	0	0.773	0	0	0	0
0.12375	0	0.748	0	0	0	0
0.2475	0	0.143	0	0	0	0
0.37125	0	0.095	0	0	0	0
0.495	0	0.0795	0	0	0	0
0.61875	0	0.0635	0	0	0	0
0.7425	0	0.0556	0	0	0	0
0.86625	0	0.040	0	0	0	0
0.99	0	0.024	0	0	0	0

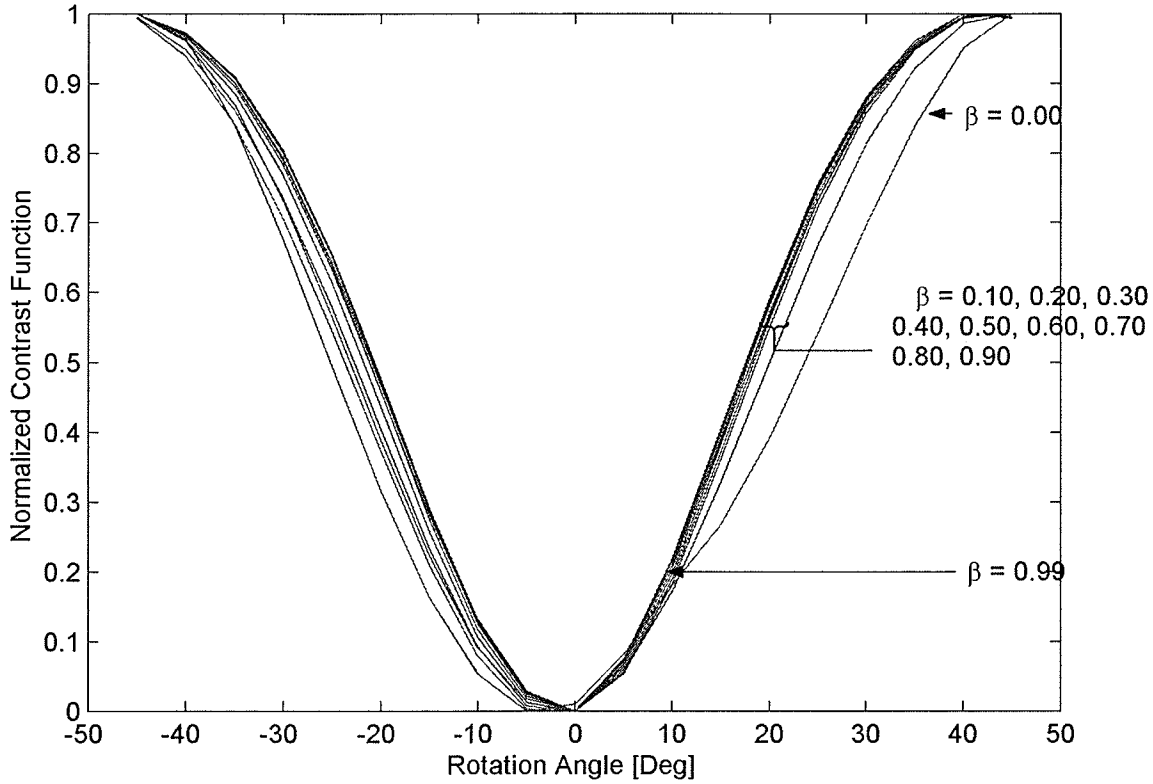


Fig. 5.11 Contrast function plot of Dataset 2 with rotation matrices between -45 and 45 degrees .

β-divergence (version 2) and performed nearly as well. The relative performances are similar to those published by Miller [42] with regards to the relative performances of RADICAL, FastICA, Extended-Infomax and JADE. The outlier contaminated simulations demonstrated that the API of β-divergence (version 2) had the lowest API.

Prior to discussing the results in detail, an explanation of the simulation results displayed in Appendix A.2 is required. Figure 5.18 shows the average API for a mixture simulation where two uniform distributions were separated. The 3-dimensional bar chart uses the height and color of each bar to display the API of each algorithm. Ten simulations, ranging from mixtures with data containing no outliers to mixtures with 1% of the data containing outliers at 7 standard deviations from the mean of the distribution, were conducted to benchmark the performance. Fourty-eight

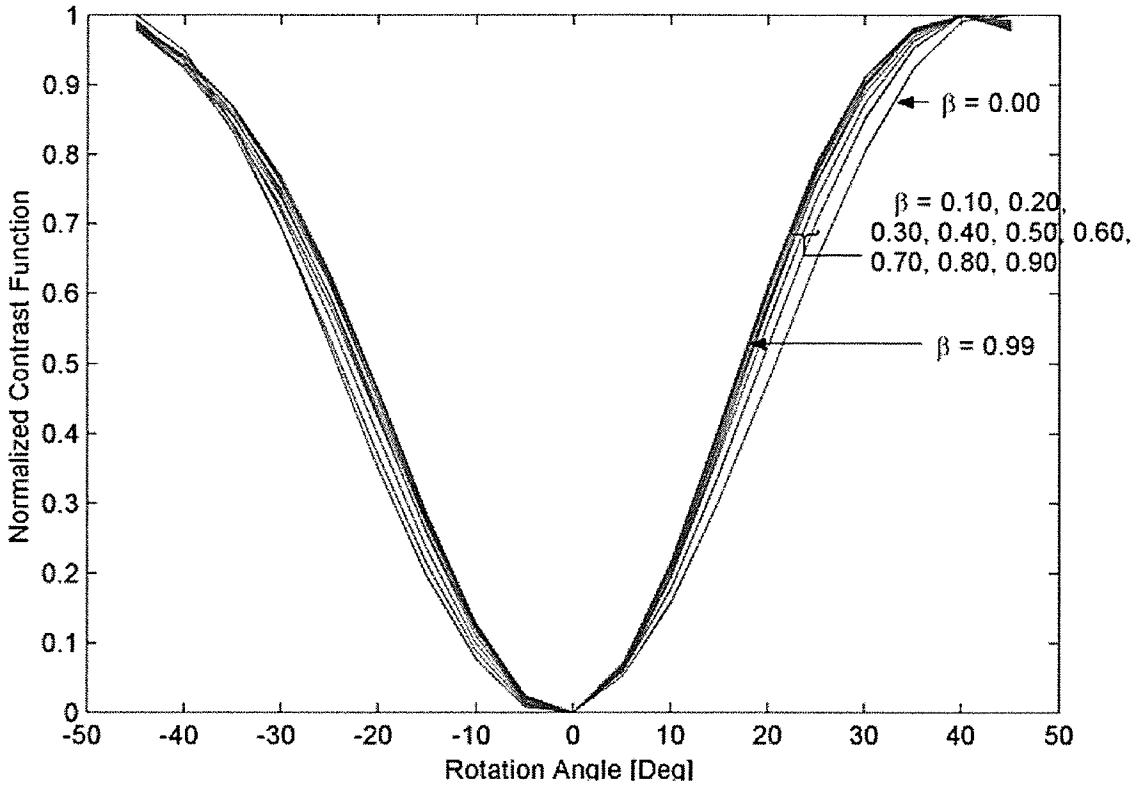


Fig. 5.12 Contrast function plot of Dataset 3 with rotation matrices between -45 and 45 degrees .

algorithm setups were used to separate the mixtures. The right most numbers of each row text indicate the number of samples used for separation (250 to 1000 pairs of samples). The middle row text is information on the algorithm setup, from the nonlinearity used to the value of β used; *e.g.*, 0_1 = β = 0.1. Finally, the left most row text is the algorithm used. The range of the API is from 0 (best) to 1 (worst).

To extract relationships between API and the simulation parameters, a linear regression and covariance analysis were performed on all of the simulations (Appendix A.5). The linear regression is used to determine the influence the number of outliers, outlier location, sample size, skewness and kurtosis on the API. To complement this, the covariance between the simulation parameters and API is also calculated.

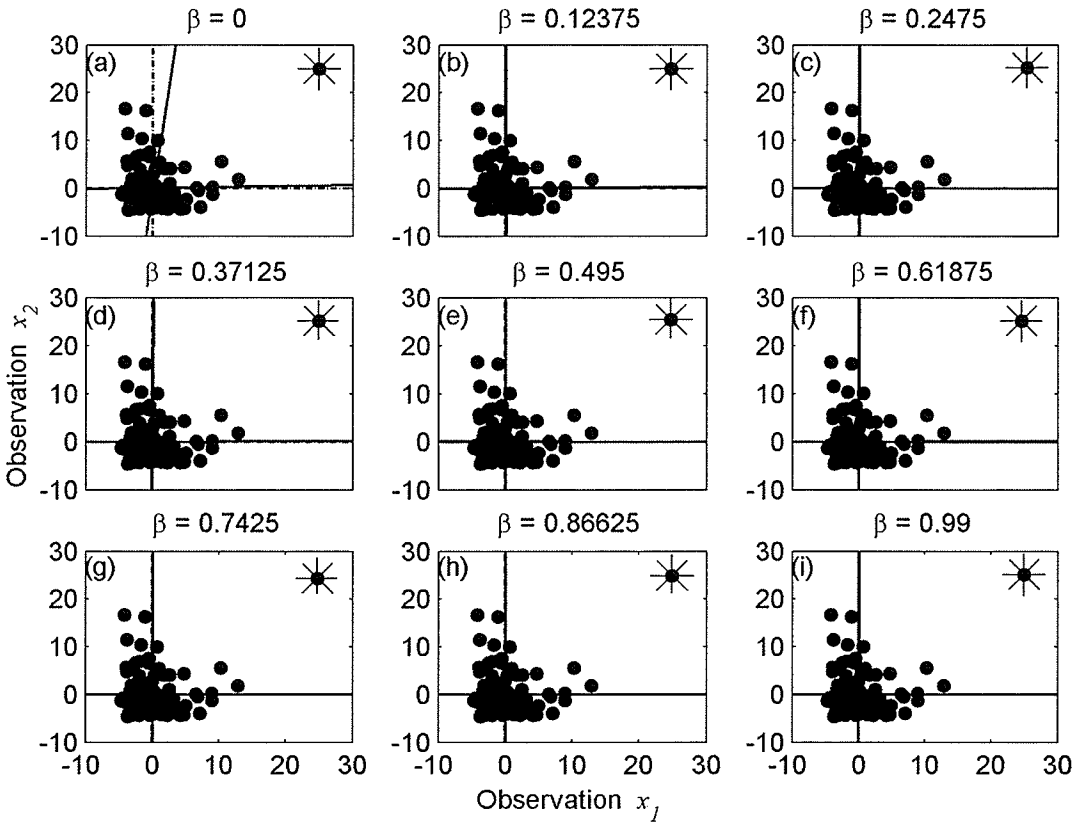


Fig. 5.13 Scatter plot of Dataset 1 (0 mean and unit variance) where the dotted lines indicate the true directions of the independent sources, while the solid lines are the estimated directions of the independent sources using the β-divergence algorithm with β ranging from 0 to 0.99. The solid dots are the uncontaminated data, while the starred points are the outliers.

5.2.1 FastICA Mixture Simulation Results and Analysis

In outlier free simulations, FastICA with the *pow3*, *tanh* and *gauss* non-linearities had average APIs near 0.1, while the *skew* non-linearity resulted in an API near 0.25. In outlier contaminated simulations the *tanh* and *gauss* non-linearities remained near 0.1 API, while the *pow3* and *skew* non-linearities increased the API above 0.4.

Recall from Sec. 2.3.1, Hyvärinen states that *tanh* is a good general purpose contrast function, *gauss* is good for sources that are super-Gaussian, or when outlier-robustness is important, and *pow3* is useful in estimating sub-Gaussian sources. Finally, skew is effective in approximating the

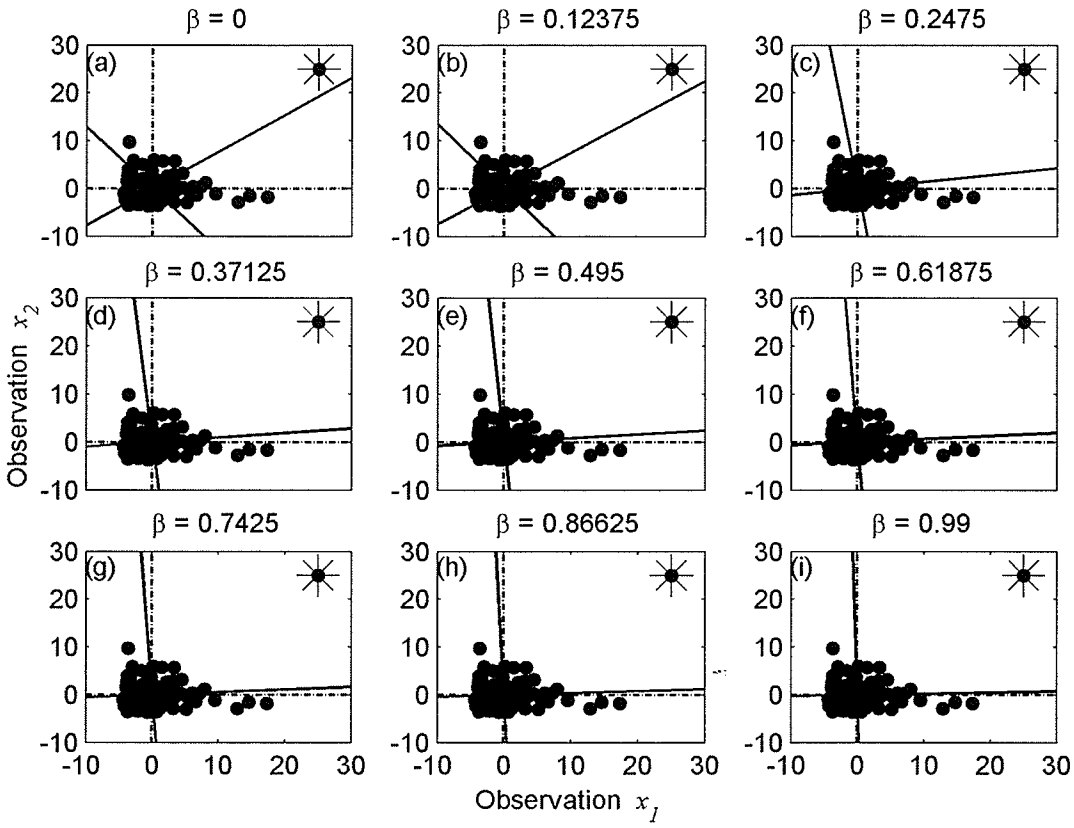


Fig. 5.14 Scatter plot of Dataset 1 where the dotted lines indicate the true directions of the independent sources, while the solid lines are the estimated directions of the independent sources using the β-divergence (version 2) algorithm with β ranging from 0 to 0.99. The solid dots are the uncontaminated data, while the starred points are the outliers.

negentropy of skewed distributions. However, *pow3* and *skew* do not meet the outlier-robustness criterion. The results of the simulations confirm these statements.

Specifically, *skew* performed worse than the other nonlinearities in the outlier free simulation due to a larger number of symmetric mixtures than asymmetric mixtures. Regarding the outlier contaminated simulations, *pow3* and *skew* performed much worse because their non-linearities are calculating the 3rd and 4th moments directly, and thus very susceptible to outliers. Using a robust 4th order suggested by Ruppert may improve the results. Examining the relationships of the outlier simulation parameters to results, outlier location rather than number of outliers had a larger influence

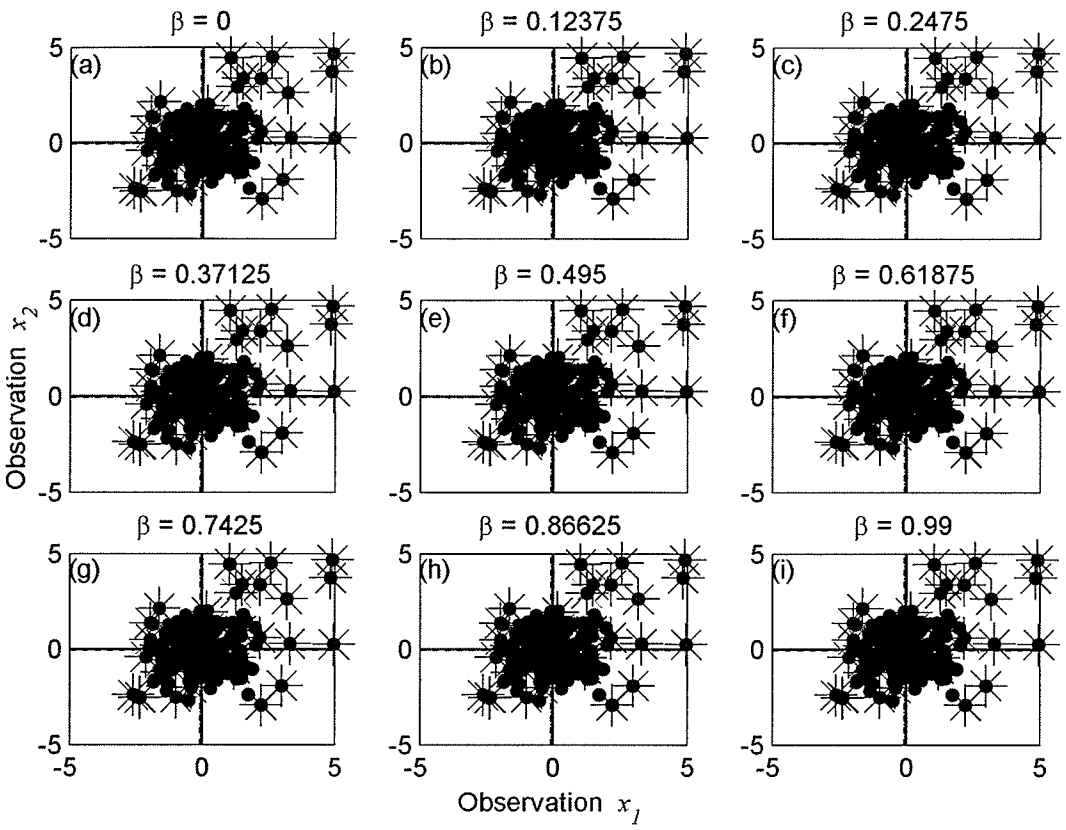


Fig. 5.15 Scatter plot of Dataset 2 where the dotted lines indicate the true directions of the independent sources, while the solid lines are the estimated directions of the independent sources using the β-divergence (version 2) algorithm with β ranging from 0 to 0.99. The solid dots are the uncontaminated data, while the starred points are the outliers.

on the API for *skew* and *pow3*. Sample size, skewness and kurtosis had minor influences on the API for FastICA.

5.2.2 Extended-Infomax Mixture Simulation Results and Analysis

In outlier free simulations Extended-Infomax had an average API near 0.4, while the apriori version had an API near 0.3. In outlier contaminated simulations the Extended-Infomax API increased to near 0.6, while the apriori version remained near 0.3. These results with the regression and covariance analysis reveal that sample size, rather than the number of outliers and outlier location had the most impact on the API.

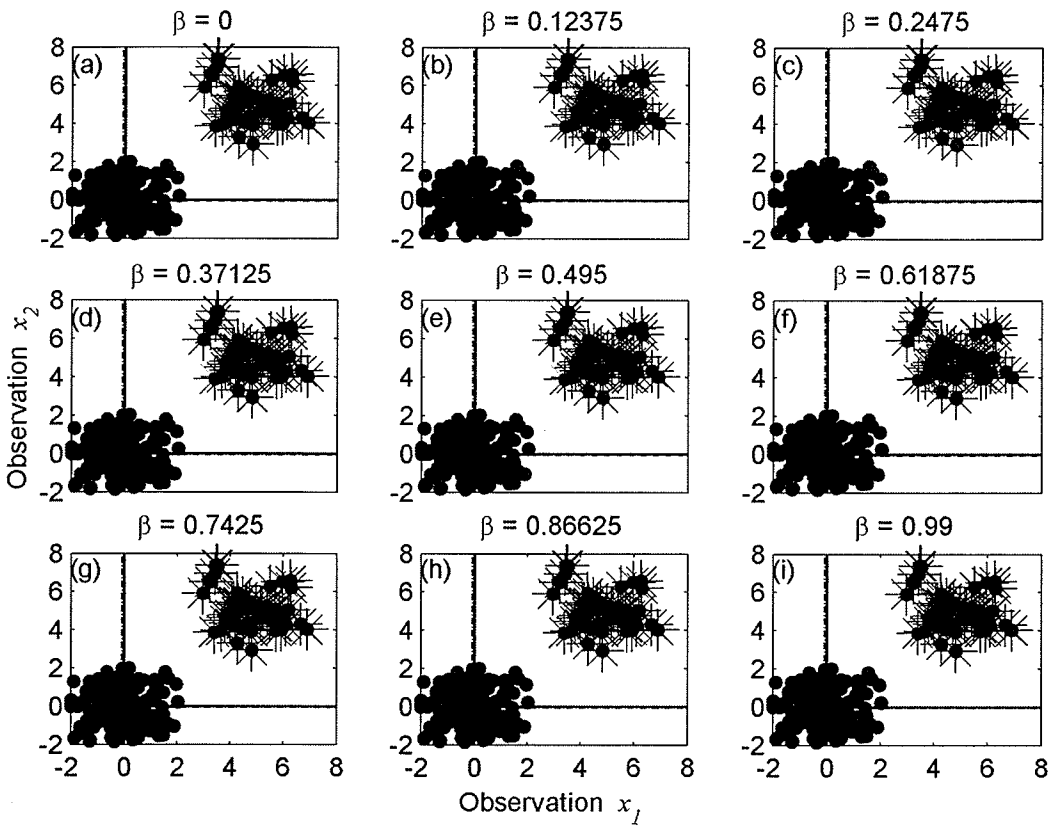


Fig. 5.16 Scatter plot of Dataset 3 where the dotted lines indicate the true directions of the independent sources, while the solid lines are the estimated directions of the independent sources using the β-divergence (version 2) algorithm with β ranging from 0 to 0.99. The solid dots are the uncontaminated data, while the starred points are the outliers.

Having a sample size of less than 1000 lead to the incorrect determination of which non-linearity to use (sub or super-Gaussian). The a priori version selected the correct nonlinearity each time, and had little variation in performance when presented with outliers. This suggests that the calculations to select which non-linearity to use must be improved in-order for Extended-Infomax to perform well.

5.2.3 JADE Mixture Simulation Results and Analysis

In outlier free simulations JADE had an average API near 0.1, and in outlier contaminated simulations near 0.6. These results with the regression and covariance analysis indicate that the

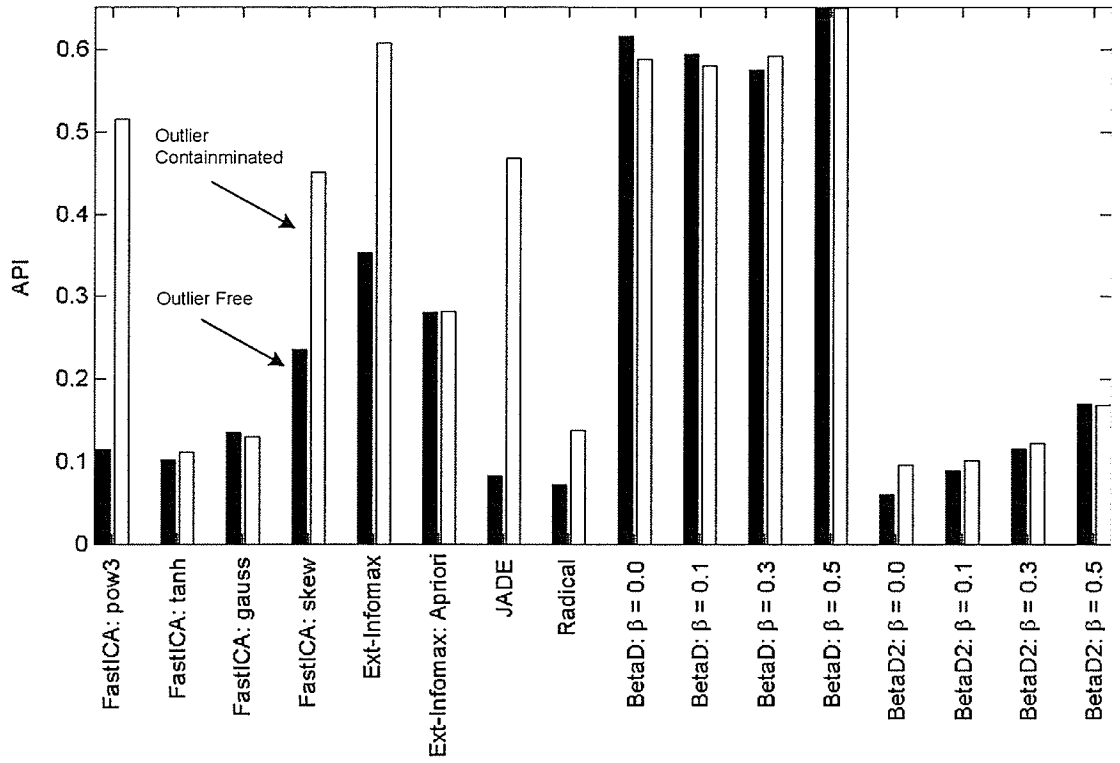


Fig. 5.17 Average API of mixture benchmarking simulation.

number of outliers and outlier location had a major impact on the performance of the algorithm. The major reason for the sensitivity is that the JADE algorithm calculates the 4th order cumulant directly, and thus is very susceptible to outliers. Switching to a more robust 4th order estimator (those suggested by Ruppert) could improve the performance of the algorithm significantly.

5.2.4 RADICAL Mixture Simulation Results and Analysis

In outlier free simulations RADICAL had an average API near 0.075, and in outlier contaminated simulations near 0.15. These results with the regression and covariance analysis indicate that the number of outliers and outlier location both have similar effects on the performance of the algorithm.

When presented with outliers, the Vasicek entropy estimator adds the log of their distance to the entropy estimate. The entropy estimator makes the data set seem more non-Gaussian than it

really should be. Since RADICAL uses a rotation search to find the optimum, the outliers adjust the contrast such that the angle which produces the minimum is off by a few degrees.

5.2.5 β-divergence Mixture Simulation Results and Analysis

In outlier free simulations β-divergence had average APIs above 0.5, and β-divergence (version 2) had average APIs near 0.1. In outlier contaminated simulations β-divergence had average APIs above 0.5, and β-divergence (version 2) had average APIs slightly above 0.1.

These results indicate that β-divergence with the BFGS optimization performs poorly. The optimization technique was unable to locate a minima that would result in a low API. The reasons for this poor performance are linked to the reasons given in the previous section, which is over sensitivity to initial conditions. However, as demonstrated by β-divergence (version 2) the contrast function does have a minima that leads to a low API.

The analysis of the mixing simulation did not show a major sensitivity to the number of outliers or outlier location. However, $\beta = 0$ was the most sensitive to outliers since the contrast weights samples with a logarithm rather than to the power of β. If Eq. 2.58 is less than 1, the logarithm could reduce the overall contrast by a significant amount. However, if $\beta! = 0$ then the contrast will increase by a maximum value of 1. The power limits the influence of a small result from Eq. 2.58.

The upward-trend of API as β increased is due to the performance of the algorithm with asymmetric distributions. As β increases, the contrast approaches a mean squared error between the hypothesized density and the empirical density. For all of the simulations, β-divergence used a symmetric (sub or super-Gaussian) hypothesized density, and thus performed poorly for skewed densities as it tries to find the demixing matrix that skews the data. The development of a density for asymmetric distributions (similar to FastICA which has the *skew* non-linearity) is needed to improve the β-divergence contrast.

5.2.6 Summary of Mixture Simulation Results and Analysis

The API analysis showed that β-divergence (version 2) had the lowest API. Although the outlier sensitivity of ICA algorithms and their contrast functions were demonstrated, two important considerations must be kept in mind. First, the ICA algorithms were designed to perform optimally for the cases considered, and may not be optimal for other cases. The APIs presented should be considered as a lower bound on the effects of outliers ICA separation performance. Second, the computational complexity of the different algorithms studied varies considerably; *e.g.*, the implementation of β-divergence with BFGS optimization requires more processing time than the other ICA algorithms.

5.3 Rotation Sensitivity Results and Analysis

Figure 5.19 shows the average change in rotation error for all outlier simulations performed. The outlier contaminated simulations demonstrated that β-divergence has the lowest average change in rotation error. (Appendix A.3 contains the results for all of the mixture simulations.)

Prior to discussing the results in detail, an explanation of the data displayed in the figures found in Appendix A.3 is required. Figure 5.20 shows the average rotation error for two uniform distributions that were mixed and then separated. The 3-dimensional bar chart displays the rotation error of each algorithm where the height and color of each bar represents the error in degrees. Ten simulations were conducted; ranging from mixtures with data containing no outliers to mixtures with 1% of the data containing outliers at 7 standard deviations from the mean of the distribution. Fourty-eight algorithm setups were used to separate the mixtures. The right most numbers of the each row label indicate the number of samples used for separation (250 to 1000 pairs of samples). The middle row text is information on the algorithm setup, from nonlinearity used to value of β used; *e.g.*, 0_1 = β = 0.1. Finally, the left most row text is the algorithm used. The range of rotation error is from 0 (best) to 45 degrees (worst). Note, the rotation error for the simulation with no outliers is the rotation error from the true rotation angle required for separation. The remaining simulations are displaying the rotation error from the angle identified from the simulation with no outliers. Thus, a

simulation with 0.5% outliers at 3 standard deviations and a 0 rotation error, indicates the change of rotation error from the simluation with no ouliers is 0. It does not indicate the rotation error from the true rotation angle error is zero. Recall, the interest is in indenitfying if the algorithm is sensitive to outliers.

In order to extract relationships between rotation error and the simulation parameters, a linear regression and covariance analysis were performed on all of the simulations (Appendix A.5). The linear regression is a method to extract the strength of dependencies of outlier number and outlier location, on the rotation error. To complement this, the covariance between the simulation parameters and the rotation error is also calculated.

5.3.1 FastICA Rotation Sensitivity Results and Analysis

In outlier contaminated simulations FastICA with the *pow3*, *tanh*, *gauss* and *skew* non-linearities had average rotation errors above 10 degrees. The linear regression and covariance analysis indicate that the outlier location had a larger impact on the angle than the number of outliers. The relatively large change in rotation error combined with the low API from the previous simulation suggest that the minimum of the contrast moved away from the rotation landscape. However, the optimization algorithm used by FastICA was able to locate the new minimum. In terms of outlier robustness, the FastICA *gauss* contrast was the least sensitive to outliers. Note that using the FastICA contrast function with a different optimization technique may result in a poor performance with outlier contaminated data.

5.3.2 Extended-Infomax Rotation Sensitivity Results and Analysis

In outlier contaminated simulations Extended-Infomax had an average rotation error of 10 degrees, and the apriori version had a rotation error near 5 degrees. The linear regression and covariance analysis indicate that the outlier location has a larger impact on the angle than the number of outliers.

The results suggest that the incorrect selection of the non-linearity lead to an increase in rotation

error. The apriori results indicate a relatively good performance. Further analysis is required to determine if the optimization technique failed to find the minimum or if the contrast function changed such that the global minimum no longer yields a low rotation error.

5.3.3 JADE Mixture Rotation Sensitivity Results and Analysis

In outlier contaminated simulations, JADE had an average rotation error of 15 degrees. The linear regression and covariance analysis indicate that the outlier location has a larger impact on the angle than the number of outliers. As explained earlier, the estimation of 4th order moments by JADE is very sensitive to outliers. A change to a more outlier robust 4th order moment estimator may lead to significantly improved results.

5.3.4 RADICAL Rotation Sensitivity Results and Analysis

In outlier contaminated simulations, RADICAL had an average rotation error of 5 degrees. The linear regression and covariance analysis indicate that the number of outliers and outlier location had similar effects on the rotation error.

As discussed earlier, when presented with outliers the Vasicek entropy estimator adds the log of their distance to the entropy estimate. Thus, an outlier at 7 standard deviations has the same effect as two outliers at 3.5 standard deviations.

5.3.5 β-divergence Rotation Sensitivity Results and Analysis

In outlier contaminated simulations, β-divergence had an average rotation error between 2 and 4 degrees. The algorithm with $\beta = 0$ had the worst rotation error as expected. Overall, the algorithm was insensitive to any outlier disturbance.

5.3.6 Summary of Rotation Sensitivity Results and Analysis

The results demonstrate that β-divergence is the most outlier robust in terms of rotation error. Contrary to the API metric, the FastICA contrasts are sensitive to outliers. The FastICA optimization technique masks this sensitivity. The optimum angle of rotation error has revealed sensitivities of ICA contrast functions to outliers.

5.4 Contrast Function Difference Results and Analysis

Figure 5.21 shows the average change in an algorithms contrast function for all outlier simulations performed. The outlier contaminated simulations demonstrated that β-divergence has the lowest average change in the contrast function. (Appendix A.4 contains the results for all of the mixture simulations.)

Prior to discussing the results in detail, an explanation of the data displayed in the figures found in Appendix A.4 is required. Figure 5.22 shows the average contrast function percent difference for two uniform distributions that were mixed and then separated. The 3-dimensional bar chart displays the contrast function difference of each algorithm where the height and color of each bar represents the percentage the contrast function has changed from the contrast function with no outliers. Ten simulations were conducted; ranging from mixtures with data containing no outliers to mixtures with 1% of the data containing outliers at 7 standard deviations from the mean of the distribution. Fourty-eight algorithm setups were used to separate the mixtures. The right most numbers of the each row text indicate the number of samples used for separation (250 to 1000 pairs of samples). The middle row text is information on the algorithm setup, from nonlinearity used to value of β used; *e.g.*, 0_1 = β = 0.1. Finally, the left most row text is the algorithm used. The range of contrast function is from 0 (best) to 100 % (worst).

In order to extract relationships between contrast function difference and the simulation parameters, a linear regression and covariance analysis were performed on all of the simulations (Appendix A.5). The linear regression is a method to extract the strength of dependencies of outlier number and outlier location on the rotation error. To complement this, the covariance between the simulation parameters and the rotation error is also calculated.

5.4.1 FastICA Contrast Function Difference Results and Analysis

In outlier contaminated simulations FastICA with the *pow3*, *tanh*, *gauss* and *skew* non-linearities had average contrast function difference above 4 percent. The linear regression and covariance analysis indicate that the outlier location had a larger impact on the angle than the number of outliers

for the *pow3*, *tanh* and *gauss* non-linearities. The *skew* non-linearity had the opposite tendency.

These results indicate, not only did the minimum of the contrast function change (rotation error), but the shape of the contrast changed as well. This metric suggests the contrast function is more sensitive to outliers than the other contrast functions.

5.4.2 Extended-Infomax Contrast Function Difference Results and Analysis

In outlier contaminated simulations Extended-Infomax had an average contrast function difference near 10 percent, and the apriori version near 2 percent. The linear regression and covariance analysis indicate that the outlier location has a larger impact on the angle than the number of outliers.

Again, the incorrect selection of the non-linearity was the major factor for the large change in the contrast function for the non-apriori version.

5.4.3 JADE Contrast Function Difference Results and Analysis

In outlier contaminated simulations, JADE had an average contrast function difference near 2 percent. The linear regression and covariance analysis indicate that the outlier location has a larger impact on the shape of the contrast than the number of outliers.

The slight change of the shape of the contrast with the large change in rotation error indicate that outliers shift the minimum of the contrast function. This suggests the JADE contrast function is fairly outlier robust and could have an improved API if an outlier-robust 4th-order cumulant estimate were used (such as Hogg's measure of the tail weight or the ratio of two interfractile ranges [63]).

5.4.4 RADICAL Contrast Function Difference Results and Analysis

In outlier contaminated simulations, RADICAL had an average contrast function difference near 2 percent. The linear regression and covariance analysis indicate that the number of outliers has a slightly larger impact on the shape of the contrast than the outlier location.

5.4.5 β-divergence Contrast Difference Function Results and Analysis

The change in the shape of the β-divergence contrast function changed less than one percent in outlier contaminated simulations. The linear regression and covariance analysis indicate that the

number of outliers had a larger impact on the shape of the contrast rather than the outlier location. Overall the contrast function was relatively robust to outliers. The metric seems to confirm the B-robustness of the algorithm.

5.4.6 Summary of Contrast Function Difference Results and Analysis

The β-divergence contrast function was the most outlier robust with respect to contrast function shape change. Extended-Infomax (apriori), JADE and RADICAL had similar outlier sensitivity. FastICA in general was the most outlier sensitive as measured by the metric.

5.5 Guidelines for Robust ICA Estimators

The primary observation made about which algorithms were the most robust is if the contrast limited the effect of an individual sample point on the overall result of the contrast. β-divergence limits the influence of sample points far from the hypothesized distribution. RADICAL contrast adds the log of of samples to its estimator. FastICA *gauss* takes the log of the hyperbolic cosine of samples. FastICA *tanh*, *pow3* and *skew* take the exponential of squared samples, and 3rd and 4th order moments of the sample points (and thus are increasingly sensitive to outliers). JADE uses the 4th order cumulant (and thus is very sensitive to outliers). Finally, Extended-Infomax uses a maximum likelihood, which is robust to outliers as outliers have a low probability, but the metric to chose the parametric family is outlier sensitive. Thus, an important aspect for an ICA algorithm to have is a contrast function that limits the influence of any sample. Conversely, it is important not to have a contrast that increases the influence of outliers by a power greater than 2. Another point to note is the perfect whitening was used in all of the simulations. Perfect whitening is not usually the case, and thus, robust whitening as a part of an ICA algorithm is imperative for an outlier insensitive algorithm (see Sec. 2.4 for suggestions for robust whitening).

5.6 Summary

This chapter has presented and analyzed the results of the outlier simulations. The outlier free mixture simulations show that β-divergence (version 2) had the best outlier separation performance

of the algorithms studied. In outlier contaminated simulations, the API metrics demonstrates that again that β-divergence (version 2) had the best separation performance. In addition, the rotation error and contrast function difference metrics reveal that the β-divergence contrast function was the most outlier robust. Given these results, the conclusion is that the β-divergence algorithm is the most outlier robust algorithm studied, and should be used when dealing with outlier contaminated data sets. However, if the dataset has a large skew, then RADICAL is suggested for use.

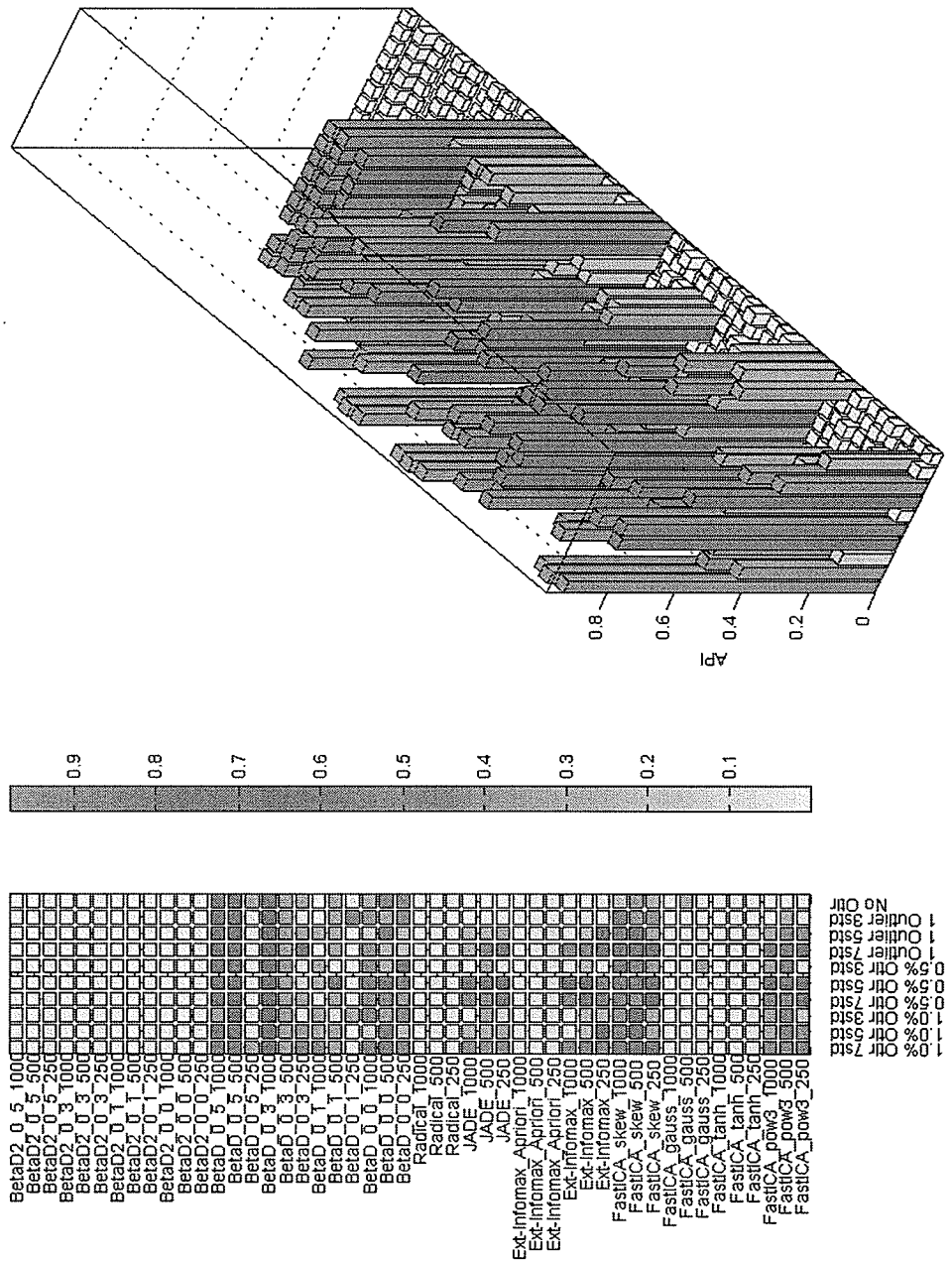


Fig. 5.18 API of Mixture Benchmarking Simulation: Uniform distribution.

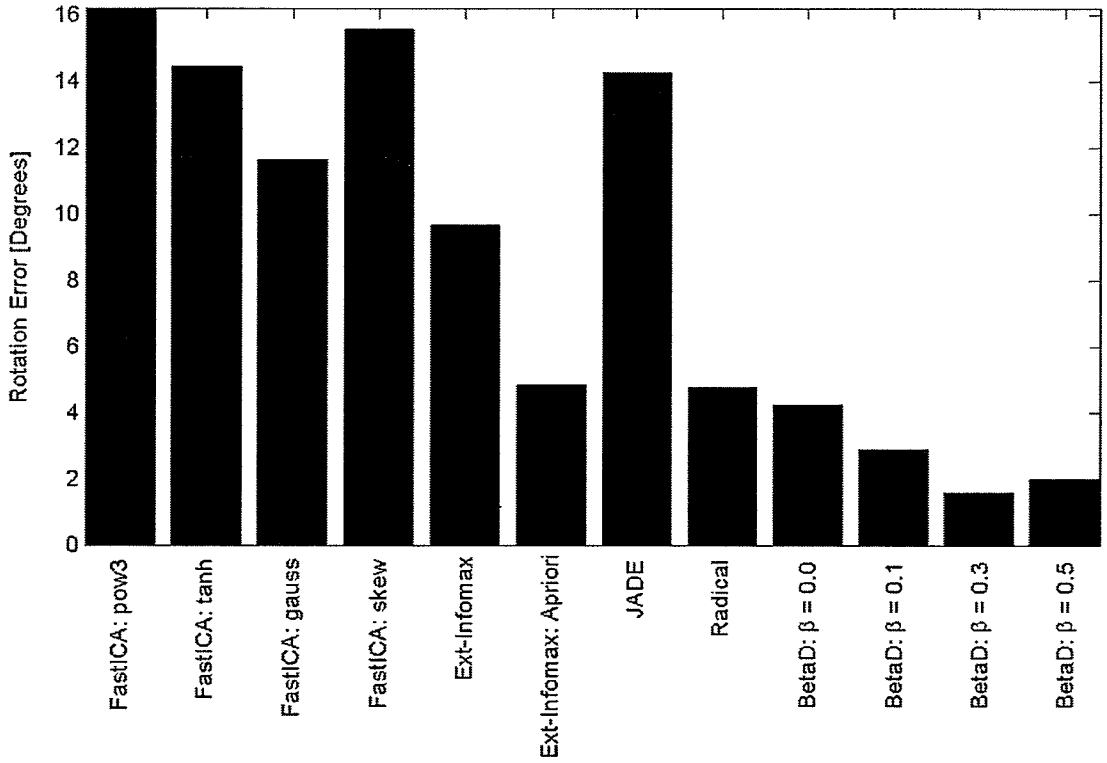


Fig. 5.19 Average change in rotation error for mixture benchmarking simulation.

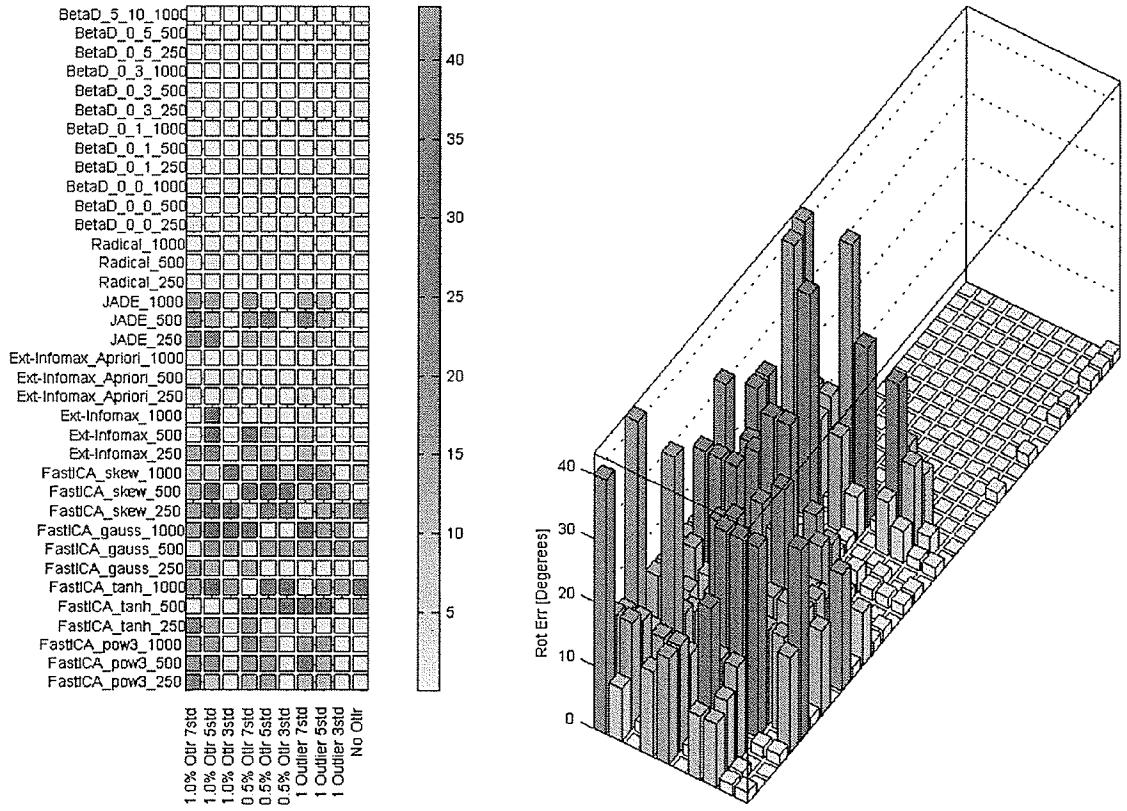


Fig. 5.20 Rotation error: Uniform distribution.

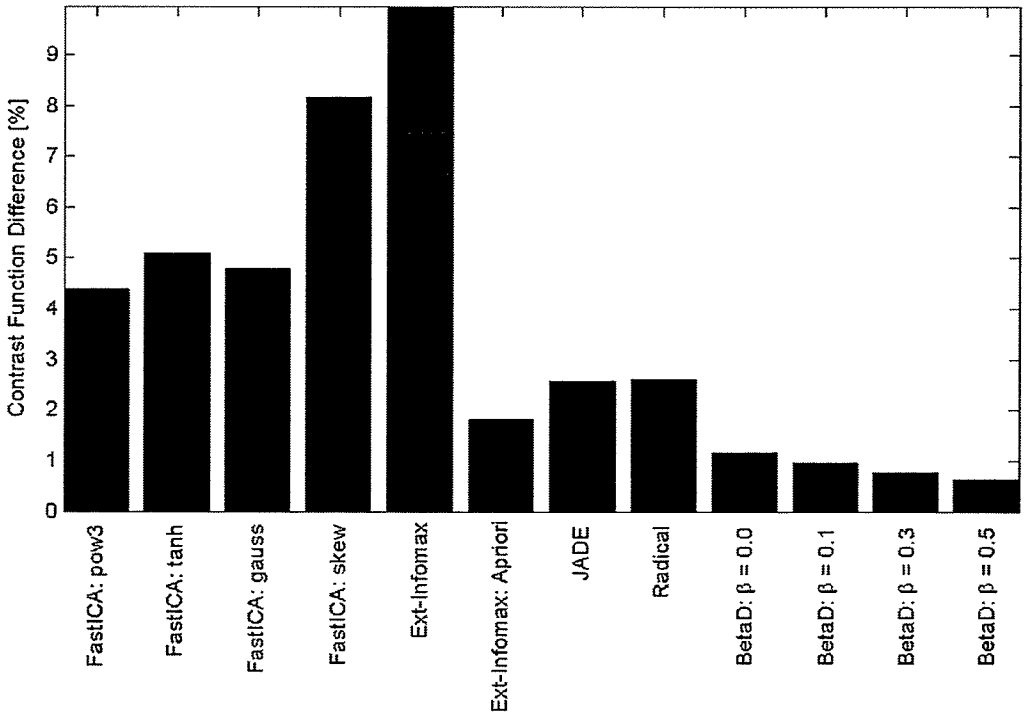


Fig. 5.21 Average change in contrast functions for mixture benchmarking simulation.

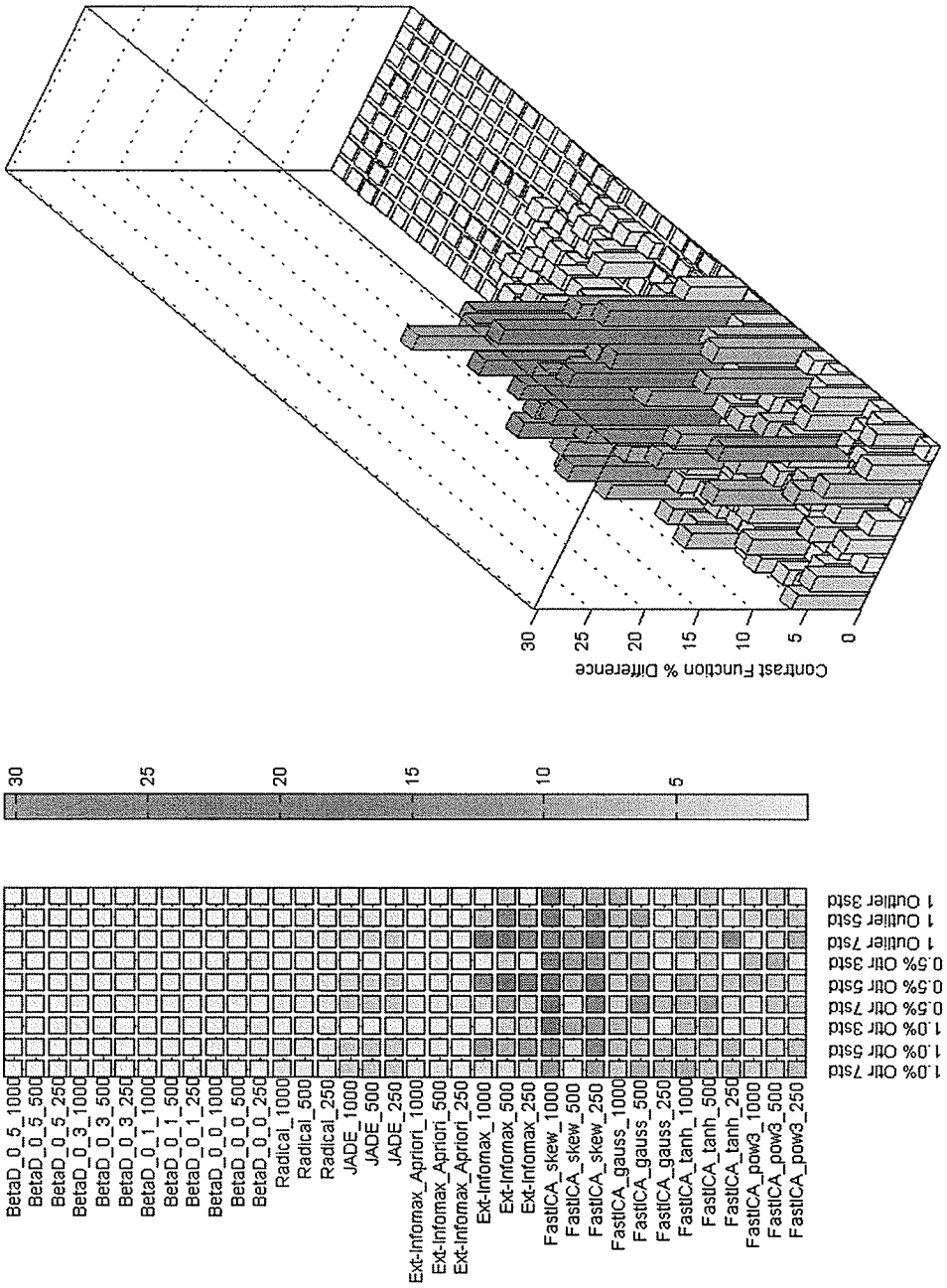


Fig. 5.22 Contrast function difference: Uniform distribution.

Chapter VI

Conclusions and Recommendations

This thesis has presented a study of the outlier robustness of *independent component analysis* (ICA) algorithms. The approach to the issue was to view ICA as a contrast function and an optimization technique, where the contrast function performs a rotation to separate the mixtures. This lead to using the *Amari separation performance index* (API), optimum angle of rotation error, and contrast function difference as three metrics used to measure the sensitivity of FastICA, JADE, Extended-Infomax, RADICAL and β-divergence algorithm implementations. Results from simulations conducted in an unbiased optimization landscape lead to the following conclusions on the outlier robustness of the ICA algorithms.

6.1 Conclusions

The objectives of the thesis were to measure the outlier sensitivity of five well-known ICA algorithms (FastICA, Extended-Infomax, JADE, RADICAL and β-divergence), rank the outlier sensitivity of these algorithms, and finally suggest how to reduce the outlier sensitivity of these ICA algorithms.

To begin, in simulations free of outliers, the average API indicated that the β-divergence algorithm (version 2) had the best separation performance (API = 0.06, β = 0.0), followed by RADICAL, JADE, β-divergence (version 2) (β = 0.1), FastICA *gauss*, FastICA *pow3*, β-divergence (version 2) (β = 0.3), FastICA *tanh*, β-divergence (version 2) (β = 0.5), FastICA *skew*, Infomax-Apriori, Infomax, and finally the original β-divergence algorithm (Sec. 5.2).

In simulations contaminated with outliers, β-divergence (version 2) (β = 0.0 and 0.1) had the lowest APIs (0.095 and 0.10), followed by FastICA *tanh, gauss*, β-divergence (version 2) (β = 0.3), RADICAL, β-divergence (version 2) (β = 0.5), Extended-Infomax Apriori, FastICA *skew*, JADE,

FastICA *pow3* and finally Extended-Infomax (Sec. 5.2).

The optimum angle of rotation error metric revealed that the β-divergence contrast had the smallest average rotation error (between 2 and 4 degrees), followed by RADICAL, Extended-Infomax Apriori, Extended-Infomax, FastICA *gauss*, FastICA *tanh*, JADE, FastICA *skew* and finally FastICA pow3 (Sec. 5.3). Interestingly, Extended-Infomax Apriori had a smaller rotation error than Extended-Infomax. This suggests that the metric to determine the Gaussianity of the unknown sources, if improved, would lead to a more outlier robust contrast function.

The contrast function difference metric revealed that the β-divergence contrast function was the most robust to outliers (average change in contrast function shape of less than 1%), followed by Extended-Infomax Apriori, JADE, RADICAL, FastICA *pow3, gauss, tanh, skew*, and finally Extended-Infomax (Sec. 5.4). Interestingly, the JADE contrast had a very small change in its contrast function shape. This aspect with the large rotation error suggests that if the contrast used a more rotation insensitive contrast, it would have better results.

The reasons for the different outlier sensitivities is related to the contrast function of each algorithm (Sec. 5.5). β-divergence is a B-robust estimator and it limits the influence of sample points far from the hypothesized distribution. RADICAL contrast adds the log of of samples to its estimator. FastICA *gauss* takes the log of the hyperbolic cosine of samples. FastICA *tanh*, *pow3* and *skew* take the exponential of squared samples, and 3rd and 4th order moments of the sample points (and thus are increasingly sensitive to outliers). JADE uses the 4th order cumulant (and thus is very sensitive to outliers). Finally, Extended-Infomax uses a maximum likelihood, which is robust to outliers as outliers have a low probability, but the metric to chose the parametric family is outlier sensitive.

Thus, the study concludes that of the algorithms studied the β-divergence (version 2) algorithm should be used when dealing with outlier contaminated data, and when dealing with life sustaining biomedical signal processing where results produced must accurate and precise. The next best algorithm to use is RADICAL due to its relatively good performance and lack of required tuning parameters, followed by FastICA *gauss*. The remaining ICA algorithms are not recommended for use with outlier contaminated data as their contrast functions are too sensitive to outliers.

Regarding the usefulness of the optimum rotation error and contrast function difference metrics, these metrics did reveal very specific aspects of the algorithms that were outlier sensitive. Specifically, the non-Gaussianity measure for Infomax, and the highly rotation sensitive JADE contrast function. Also, the rotation sensitivity of FastICA confirmed that an optimization technique that searched outside the rotation landscape is required for this algorithm to achieve a good separation performance. Finally, the low rotation error and contrast function difference of the β-divergence algorithm did correlate to a low overall API when the algorithm was presented with outliers. Thus, the optimum angle of rotation error and the contrast difference were good measures for revealing the sensitivities of ICA algorithms to outliers, and the potential separation performance of ICA algorithms.

6.2 Contributions

The thesis has made the following contributions:

(a) A review of ICA from the perspective of outliers was provided in Ch. 2;

(b) The development of the contrast function and unbiased optimization landscape as the key to the analysis of the outlier robustness and potential separation performance of ICA algorithms was discussed in Sec. 1.2;

(c) Optimum angle of rotation error outlier sensitivity metric (Sec. 3.3) ;

(d) Contrast function difference outlier sensitivity metric (Sec. 3.4);

(e) Development of an outlier dataset for ICA algorithm benchmarking (Sec. 4.2);

(f) Benchmarking the ICA algorithm separation performance with and without outliers in a unbiased optimization landscape (Ch. 5 and App. A); and

(g) Implementing the β-divergence algorithm and making it available to the research community (Sec. 2.3.6 and 5.1).

6.3 Recommendations for Future Work

The following are recommendations for future work based on the investigation of outlier robustness for ICA:

(a) Implement the automatic selection of β using the technique suggested by Minami and Eguchi [50]. The primary reason is because it leads to a more self-contained algorithm that requires less user input.

(b) Investigate alternate optimization techniques to use with the β-divergence contrast function as the BFGS optimization as implemented performs poorly, and the exhaustive rotation search is computationally intensive. This would result in an algorithm that would be able run closer to real-time with today's computers.

(c) Implement an alternative 4th-order cumulant measure for JADE, as the simulations reveal that this estimator is highly sensitive to outliers. This is feasible as Ruppert [63] has provided 4th order measures, that are outlier robust, which maybe adapted for this purpose.

(d) Investigate alternate outlier robust Gaussianity measures for Infomax as the simulations reveal that selecting the correct density lead to improved performance.

(d) Reevaluate approximating negentropy (Hyvärinen's and Comon's) by using the robust Gram-Charlier and Edgeworth PDF expansions of Welling [68]. This task may result in a more outlier robust FastICA algorithm with relatively few changes to code.

(e) Update the β-divergence implementation to handle more than 2 sensor observations. The reason being for this algorithm to be more useful in real-world signal processing, it must be able to handle more than 2 signals; *e.g.,* ECG signal processing usually has 6 sensor leads.

(f) Investigate moving the algorithms from Matlab to firmware (FPGA or DSP) for running in real time as we are interest is in using ICA algorithms as a part of real-time heart monitor.

References

[1] S. Amari, *Differential-Geometrical Methods in Statistics. Lecture Notes in Statistics 28.* New York, NY: Springer-Verlag, 1985, 290 pp.

[2] ——, "Natural gradient works efficiently in learning," *Neural Computation*, vol. 10, no. 2, pp. 251–276, 1998.

[3] ——, "Natural gradient learning for over- and undercomplete bases in ICA," *Neural Computation*, vol. 11, no. 8, pp. 1875–1883, 1999.

[4] S. Amari and S. Douglas, "Why natural gradient?" *Acoustics, Speech, and Signal Processing, 1998. ICASSP '98*, vol. 2, pp. 1213–1216, 1998.

[5] U. Ascher. Numerical Optimization. 542b-03-notes.pdf. [Online]. http://www.cs.ubc.ca/spider/ascher/542.html (available as of Sept 22, 2005).

[6] V. Barnett and T. Lewis, *Outliers in Statistical Data.* New York, NY: Wiley, 1994 (3rd ed.), 584 pp.

[7] A. Basu, I. Harris, N. Hjort, and M. Jones, "Robust and efficient estimation by minimising a density power divergence," *Biometrika*, vol. 85, no. 3, pp. 549–559, 1998.

[8] A. Bell and T. Sejnowski, "An information-maximization approach to blind separation and blind deconvolution," *Neural Computation*, vol. 7, no. 6, pp. 1129–1159, 1995.

[9] C. Bishop, *Neural Networks for Pattern Recognition.* Oxford, UK: Oxford University Press, 1995, 482 pp.

[10] J. Cardoso. ICA list. [Online]. http://www.tsi.enst.fr/icacentral/mailinglist.html (available as of Sept 22, 2005).

[11] ——. JADE for Matlab: Version 1.5. [Online]. http://www.tsi.enst.fr/~cardoso/Algo/Jade/jadeR.m (available as of February 2, 2005).

[12] ——, "Infomax and maximum likelihood for blind source separation," *IEEE Signal Processing Letters*, vol. 4, no. 4, pp. 112–114, 1997.

[13] ——, "Higher-order contrasts for independent component analysis," *Neural Computation*, vol. 11, no. 1, pp. 157–192, 1999.

[14] J. Cardoso and D. Pham, "Separation of non-stationary sources: Algorithms and performance," in *Indpendent Component Analysis, Principles and Practices.* S. Roberts and R. Everson (eds). Cambridge, UK: Cambridge Univ. Press, 2001, Ch. 5, pp. 158–180.

[15] J. Cardoso and A. Souloumiax, "Blind beamforming for non-Gaussian signals," *IEE Proceedings F*, vol. 140, no. 6, pp. 362–370, 1993.

[16] J. Chen, Y. Kung, and R. Hudson, "Source localization and beamforming," *IEEE Signal Processing Magazine*, vol. 19, no. 2, pp. 30–39, 2002.

[17] A. Cichocki and S. Amari, *Adaptive Blind Signal and Image Processing*. New York, NY: Wiley, 2002, 554 pp.

[18] P. Comon, "Indpendent component analysis: A new concept?" *Signal Processing (Elsevier)*, vol. 36, no. 3, pp. 287–314, 1994.

[19] T. Cover and J. Thomas, *Elements of Information Theory*. New York, NY: Wiley, 1991, 542 pp.

[20] D. Erdogmus, K. Hild II, and J. Principe, "Blind source separation using Renyi's marginal entropies," *Neurocomputing*, vol. 49, no. 1, pp. 25–38, 2002.

[21] A. Ferreria and M. Figueiredo, "Class-adapted image compression using independent component analysis," *2003 International Conference on Image Processing*, vol. 1, pp. 625–628, 2003.

[22] J. Fisher. ICA data. [Online]. http://www.ai.mit.edu/people/fisher/ica_data/ (available as of February 13, 2004).

[23] N. Gadhok and W. Kinsner. ICA by Beta-divergence for Matlab: Version 1.0. [Online]. http://www.ee.umanitoba.ca/~kinsner/projects/ (available as of February 2, 2005).

[24] ——. Newsgroup discussion on the origins of ICA. [Online]. http://www.ee.umanitoba.ca/~kinsner/projects/ (available as of November 20, 2005).

[25] ——, "Robust independent component analysis for cognitive informatics," *The 4th IEEE International Conference on Cognitive Informatics, ICCI 2005*, pp. 86–92, 2005.

[26] ——, "Rotation sensitivity of independent component analysis to outliers," *Canadian Conference on Electrical and Computer Engineering 2005*, pp. 1437–1442, 2005.

[27] ——, "An implementation of β-divergence for blind source separation," *Canadian Conference on Electrical and Computer Engineering 2006*, vol. TBD, no. TBD, p. TBD, 2006.

[28] H. Gävert, J. Hurri, J. Särelä, and A. Hyvärinen. FastICA for Matlab: Version 2.3. [Online]. http://www.cis.hut.fi/projects/ica/fastica/ (available as of February 2, 2005).

[29] F. Hampel, E. Ronchetti, P. Rousseeuw, and W. Stahel, *Robust Statistics: The Approach Based on Influence Functions*. New York, NY: Wiley, 1986, 502 pp.

[30] S. Haykin, *Unsupervised Adaptive Filtering*. New York, NY: Wiley, 2000, 446 pp.

[31] D. Hoaglin, F. Mosteller, and J. Tukey, *Understanding Robust and Exploratory Data Analysis*. New York, NY: Wiley, 1983, 447 pp.

[32] P. Huber, *Robust Statistics*. New York, NY: Wiley, 1981, 308 pp.

[33] A. Hyvärinen, "One-unit contrast functions for independent component analysis: A statistical analysis," in *Proc. IEEE Neural Networks for Signal Processing VII (NNSP Workshop '97)*, pp. 388–397, 1997.

[34] ——, "New approximations of differential entropy for independent component analysis and projection pursuit," in *Proc. Advances in Neural Information Processing Systems (NIPS'97)*, pp. 273–279, 1998.

[35] ——, "Fast and robust fixed-point algorithms for independent component analysis," *IEEE Transcations on Neural Networks*, vol. 10, no. 3, pp. 626–634, 1999.

[36] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent Component Analysis*. New York, NY: Wiley, 2001, 481 pp.

[37] A. Hyvärinen and E. Oja, "A fast fixed-point algorithm for independent component analysis," *Neural Computation*, vol. 9, no. 7, pp. 1483–1492, 1997.

[38] C. Jutten and J. Herault, "Blind separation of sources, Part I: An adaptive algorithm based on neuromimetic architecture," *Signal Processing (Elsevier)*, vol. 24, pp. 1–10, 1991.

[39] W. Kinsner, *Microcontroller, Microprocessor, and Microcomputer Interfacing for Real-Time Systems*. Winnipeg, MB: University of Manitoba Dept. of Electrical and Computer Engineering and TRLabs, 2000, 204 pp.

[40] ——, *Fractal and Chaos Engineering*. Winnipeg, MB: University of Manitoba Dept. of Electrical and Computer Engineering, 2003, 765 pp.

[41] E. Learned-Miller. RADICAL-ICA for Matlab: Version 1.1. [Online]. http://www.cs.umass.edu/~elm/ICA/ (available as of February 2, 2005).

[42] E. Learned-Miller and J. Fisher III, "ICA using spacing estimates of entropy," *J. Machine Learning Research*, vol. 4, pp. 1271–1295, Dec. 2003.

[43] T. Lee, *Independent Component Analysis - Theory and Applications*. New York, NY: Kluwer, 1998, 248 pp.

[44] T. Lee, M. Girolami, and T. Sejnowski, "Independent component analysis using an extended infomax algorithm for mixed subgaussian and supergaussian sources," *Neural Computation*, vol. 11, no. 2, pp. 417–441, 1999.

[45] T. Lee and M. Lewicki, "Unsupervised image classification, segmentation, and enhancement using ICA mixture models," *IEEE Transactions on Image Processing*, vol. 11, no. 3, pp. 270–279, 2002.

[46] S. Lowen, S. Liebovitch, and J. White, "Fractal ion-channel behavior generates fractal firing patterns in neuronal models," *Physical Review E*, vol. 59, no. 5, pp. 5970–5980, 1999.

[47] S. Makeig. EEGLAB: ICA toolbox for psychophysiological research: Version 4.5. [Online]. http://sccn.ucsd.edu/eeglab/ (available as of February 2, 2005).

[48] Mathworks. Chapter 9, random number generation. [Online]. http://www.mathworks.com/moler/random.pdf (available as of May. 12, 2006).

[49] M. Minami and S. Eguchi, "Robust blind source separation by β-divergence," *Neural Computation*, vol. 14, pp. 1859–1886, 2002.

[50] ——, "Adaptive selection for minimum β-divergence method," in *4th International Symposium on Indepedent Component Analysis and Blind Signal Separation (ICA2003)*, pp. 475–480, 2003.

[51] S. K. Mitra, *Digital Signal Processing: A Computer-Based Approach*. New York, NY: McGraw-Hill, 1998, 864 pp.

[52] R. Ng. Robust space transformations for distance based outliers. [Online]. http://www.pims.math.ca/science/2002/icors/videos/raymond_ng (available as of Sept. 22, 2005).

[53] J. Nocedal and S. Wright, *Numerical Optimization*. New York, NY: Springer Verlag, 1999, 636 pp.

[54] A. Papoulis and S. Pillai, *Probability, Random Variables and Stochastic Processes*. New York, NY: McGraw Hill, 2002 (4th ed.), 852 pp.

[55] L. Parra and C. Spence, "Separation of non-stationary natural signals," in *Indpendent Component Analysis, Princples and Practices*, S. Roberts and R. Everson, Eds. Cambridge, UK: Cambridge Univ. Press, 2001, Ch. 5, pp. 135–157.

[56] M. Potter, N. Gadhok, and W. Kinsner, "Separation performance of ICA on simulated EEG and ECG signals contaminated by noise," *Can. J. Electrical and Computer Engineering*, vol. 27, no. 3, pp. 123–127, 2002.

[57] C. Puntonet and A. Prieto, Eds., *Independent Component Analysis and Blind Signal Separation: Fifth International Conference, ICA 2004, Granada, Spain, September 22-24, 2004. Proceedings*, ser. Lecture Notes Computer Science. Berlin, Germany: Springer-Verlag, 2004, vol. 3195.

[58] R. Rangayyan, *Biomedical Signal Analysis: A Case Study Approach*. New York, NY: IEEE Press/Wiley, 2002, 516 pp.

[59] T. Ristaniemi and J. Joutsensalo, "Advanced ICA-based receivers for DS-CDMA systems," *Proc. 11th IEEE International Symposium on Personal, Indoor, and Mobile Radio Communications*, pp. 276–281, 2000.

[60] S. Roberts and R. Everson, *Independent Component Analysis - Principles and Practice*. Cambridge, UK: Cambridge University Press, 2001, 338 pp.

[61] P. Rousseeuw and A. Leroy, *Robust Regression and Outlier Detection*. New York, NY: Wiley, 1987, 329 pp.

[62] R. Rousseeuw and K. V. Driessen, "A fast algorithm for the minimum covariance determinant estimator," *Technometrics*, vol. 41, pp. 212–223, 1999.

[63] D. Ruppert, "What is kurtosis? An influence function approach," *American Statistician*, vol. 41, no. 1, pp. 1–4, 1987.

[64] A. Stuart and J. Ord, *Kendall's Advanced Theory of Statistics: Distribution Theory*. New York, NY: Wiley, 1993 (6th ed.), vol. 1, 676 pp.

[65] J. Tukey, "A survey of sampling from contaminated distributions," in *Contributions to Probability and Statistics*. I. Olkin and *et al.* (eds). Stanford, CA: Stanford Univ. Press, 1960, Ch. 39, pp. 448–485.

[66] O. Vasicek, "Test for normality based on sample entropy," *Journal of the Royal Statistical Society. Series B*, vol. 38, no. 1, pp. 54–59, 1976.

[67] F. Vrins and M. Verleysen, "On the entropy minimization of a linear mixture of variables for source separation," *Signal Processing (Elsevier)*, vol. 85, pp. 1029–1044, 2005.

[68] M. Welling, "Robust higher order statistics," Department of Computer Science, University of Toronto, 6 King's College Road Toronto, M5S 3G5, Canada, Tech. Rep., 2003.

[69] Wikipedia. Givens rotation. [Online]. http://en.wikipedia.org/wiki/Givens_rotation (available as of Dec. 12, 2005).

[70] ——. Gradient descent. [Online]. http://en.wikipedia.org/wiki/Gradient_descent (available as of Dec. 12, 2005).

[71] ——. Linear. [Online]. http://en.wikipedia.org/wiki/Linear (available as of Dec. 12, 2005).

[72] V. Zarzoso and A. Nandi, "Noninvasive fetal electrocardiogram extraction: blind separation versus adaptive noise cancelation," *IEEE Trans. Biomedical Engineering*, vol. 48, no. 1, pp. 12–18, 2001.

APPENDIX A

RESULTS

A.1 Statistics of Randomly Generated Data

The following table contains the statistics of the 10000 samples generated data from each of the probability distribution functions shown in Fig. 4.2.

Table A.1 Statistics of randomly generated data.

Distribution	Mean	Variance	Skewnewss	Kurtosis
Student-t 3 degrees of freedom	0	1	-0.39	15.12
Double exponential (Laplace)	0	1	0.05	3.22
Uniform	0	1	-0.01	-1.20
Student-t 5 degrees of freedom	0	1	-0.10	4.10
Exponential	0	1	1.89	4.73
Mixture of 2 double exponentials	0	1	-0.01	-1.05
Symmetric mixture of 2 Gaussians: multimodal	0	1	0.00	-1.77
Symmetric mixture of 2 Gaussians: transitional	0	1	0.01	-0.64
Symmetric mixture of 2 Gaussians: unimodal	0	1	0.01	-0.53
Asymmetric mixture of 2 Gaussians: multimodal	0	1	-0.75	-0.41
Asymmetric mixture of 2 Gaussians: transitional	0	1	-0.92	0.16
Asymmetric mixture of 2 Gaussians: unimodal	0	1	-0.31	-0.39
Symmetric mixture of 4 Gaussians: multimodal	0	1	-0.01	1.04
Symmetric mixture of 4 Gaussians: transitional	0	1	0.00	-.77
Symmetric mixture of 4 Gaussians: unimodal	0	1	0.00	-.66
Asymmetric mixture of 4 Gaussians: multimodal	0	1	0.14	-.85
Asymmetric mixture of 4 Gaussians: transitional	0	1	-0.14	-.55
Asymmetric mixture of 4 Gaussians: unimodal	0	1	-.28	-.65
Normal	0	1	0.02	0.03
Log-normal	0	1	3.44	66.28
Pareto	0	1	98.7	9812.35

A.2 APIs of Benchmarking Simulation

The following sidewaysfigures contain the APIs of various ICA algorithms performed on a mixture of the given density. The experiment setup is explained in Sec. 4.3.

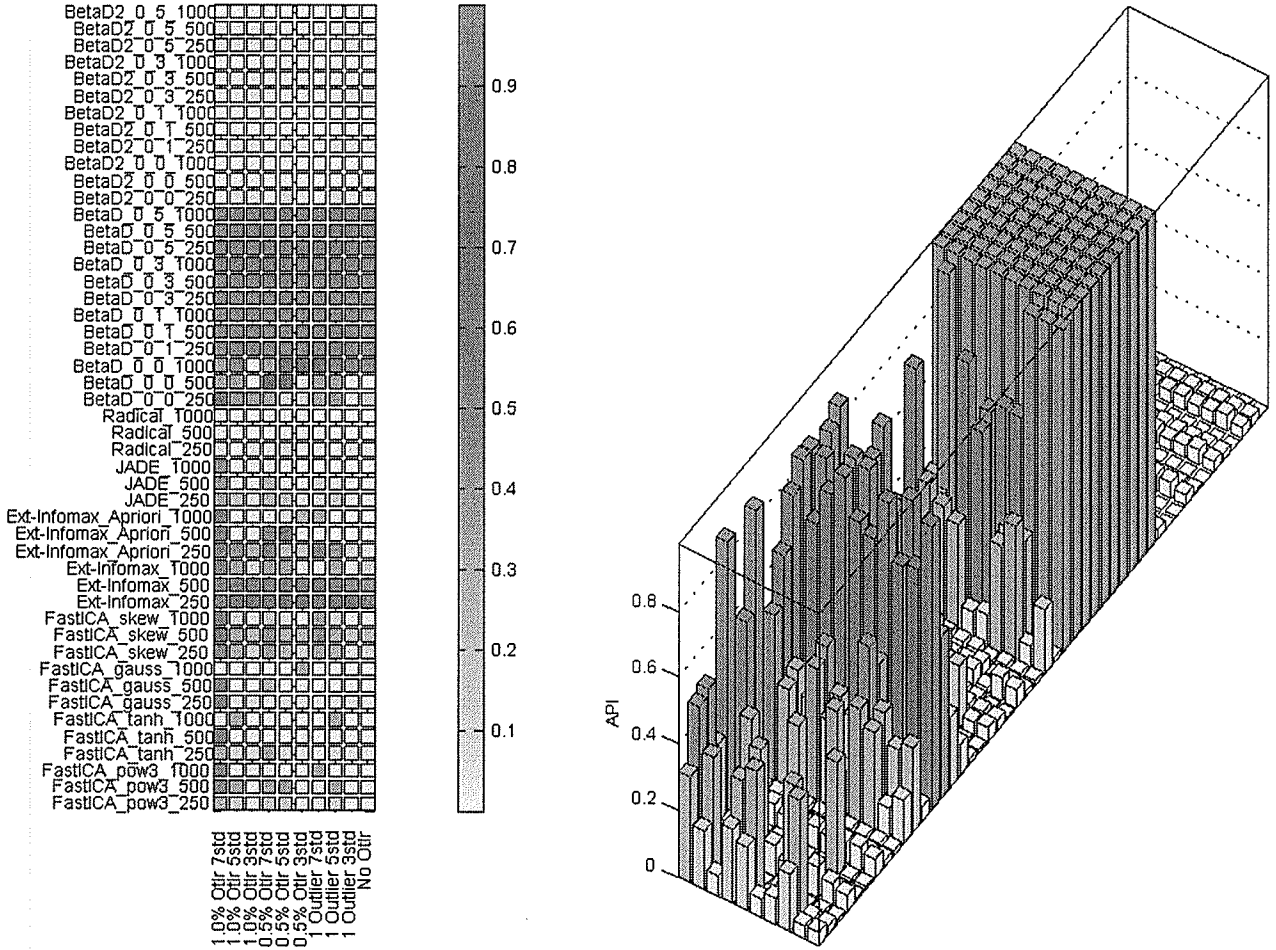


Fig. A.1 API of mixture benchmarking simulation: Student-t 3 degrees of freedom.

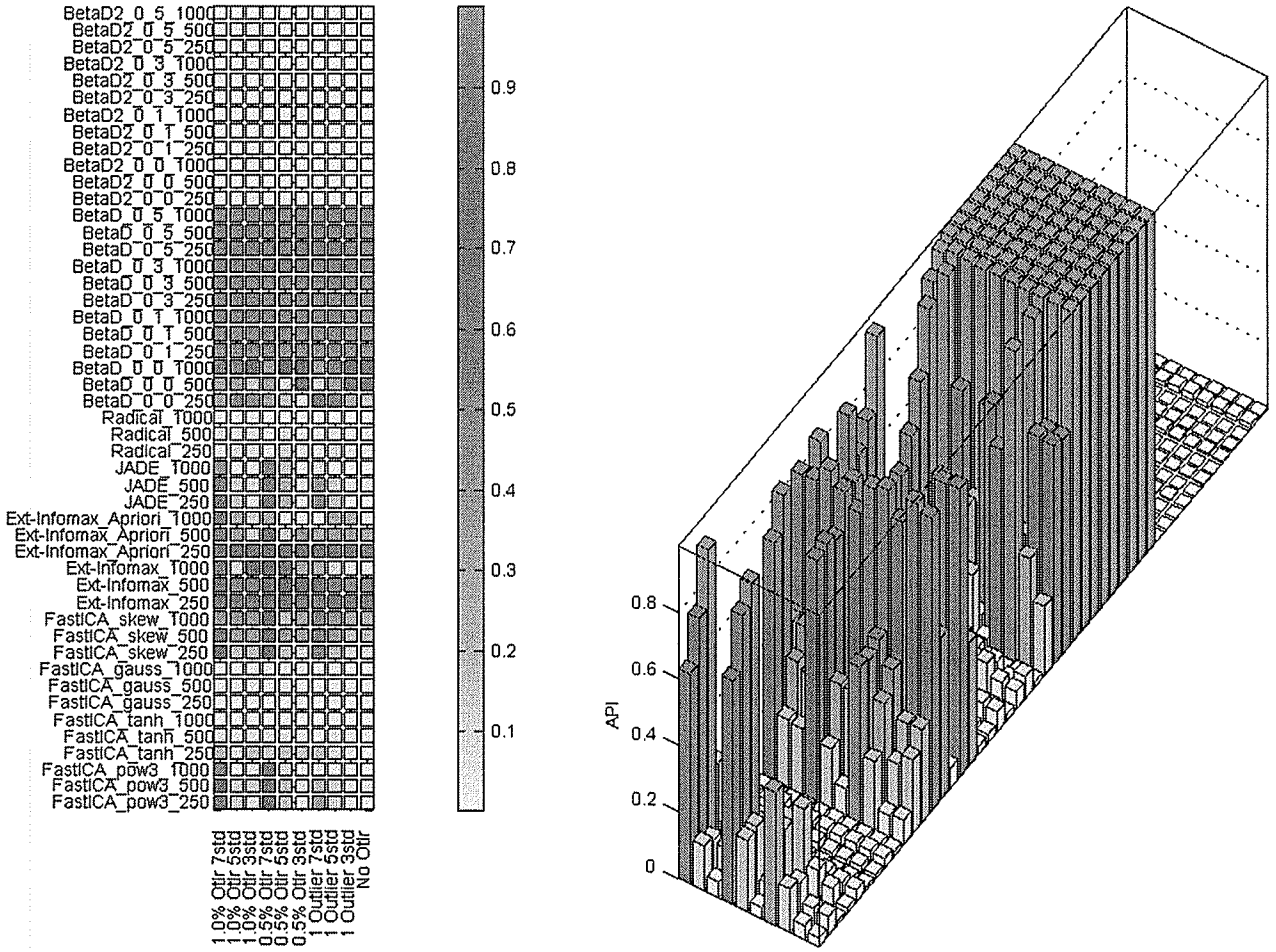


Fig. A.2 API of mixture benchmarking simulation: Double Exponential.

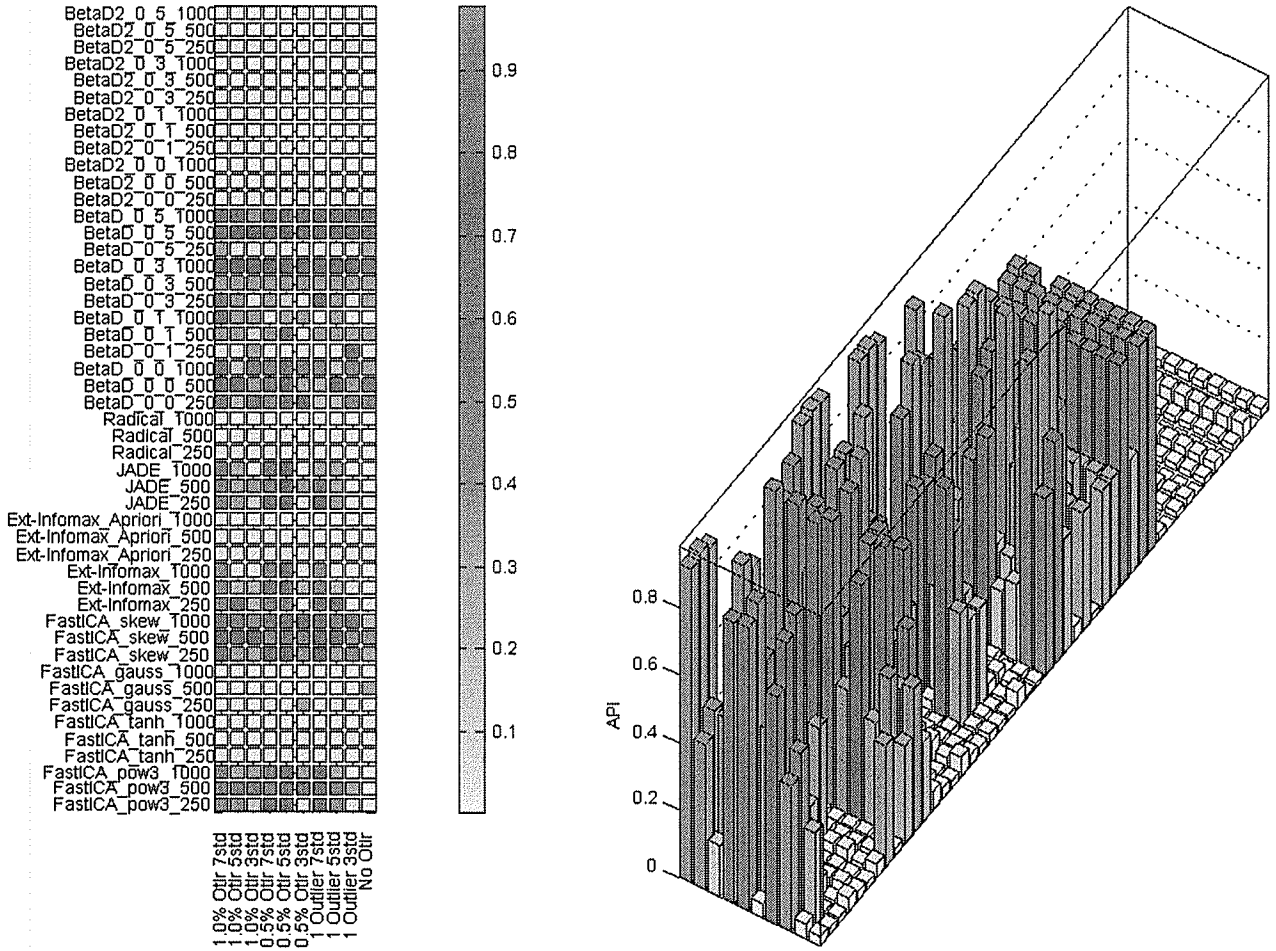


Fig. A.3 API of mixture benchmarking simulation: Uniform.

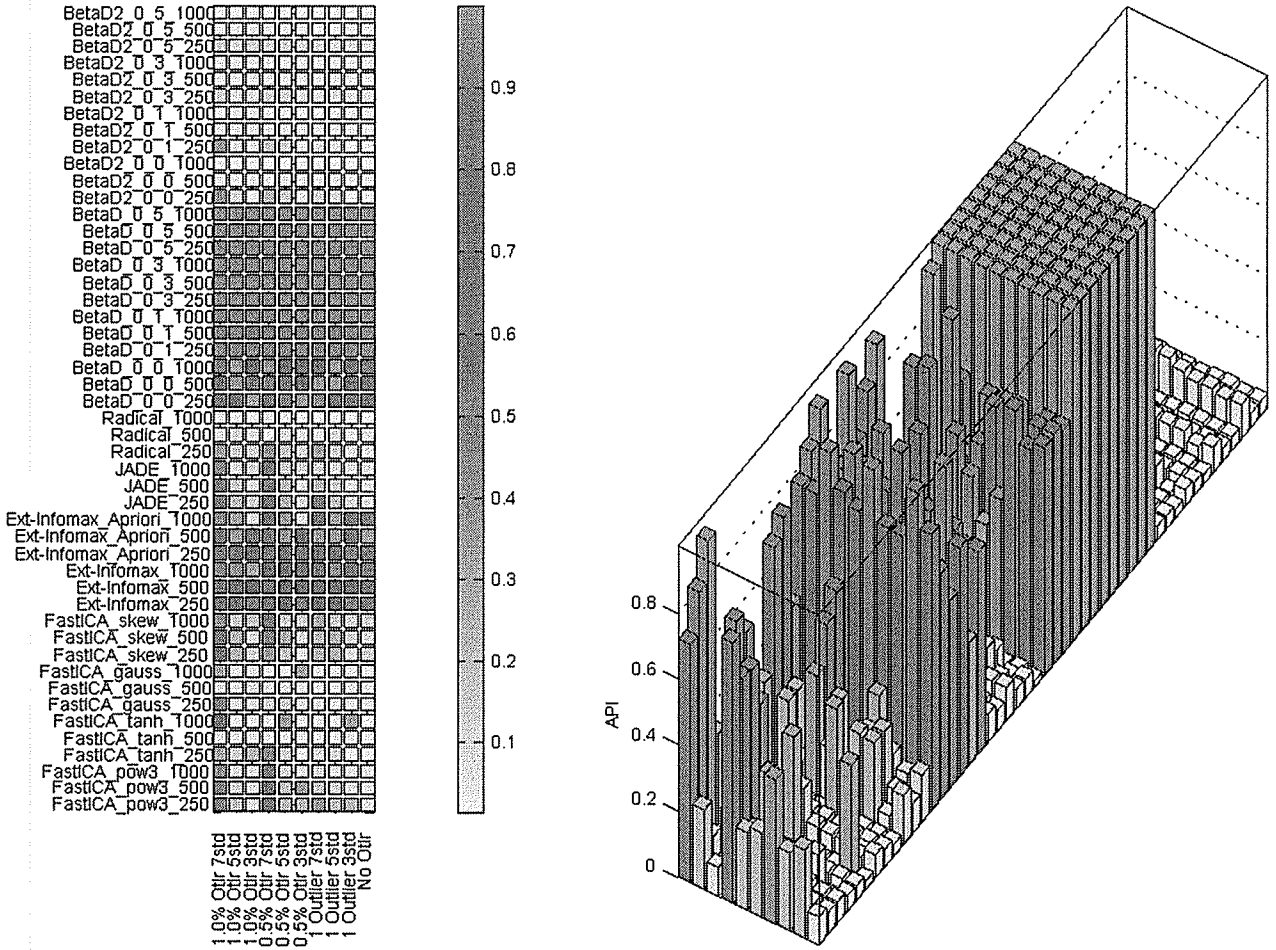


Fig. A.4 API of mixture benchmarking simulation: Student-t 5 degrees of freedom.

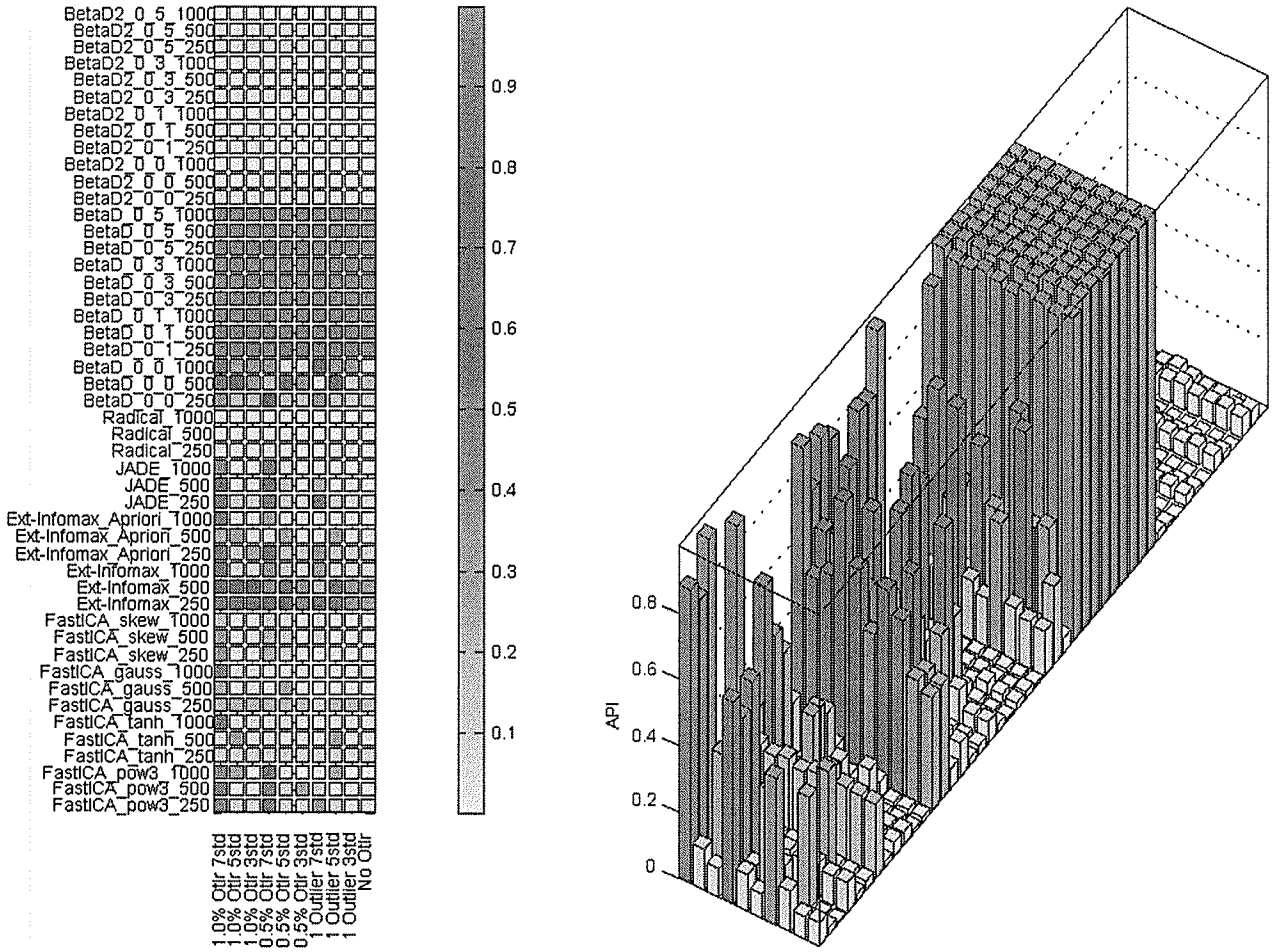


Fig. A.5 API of mixture benchmarking simulation: Exponential.

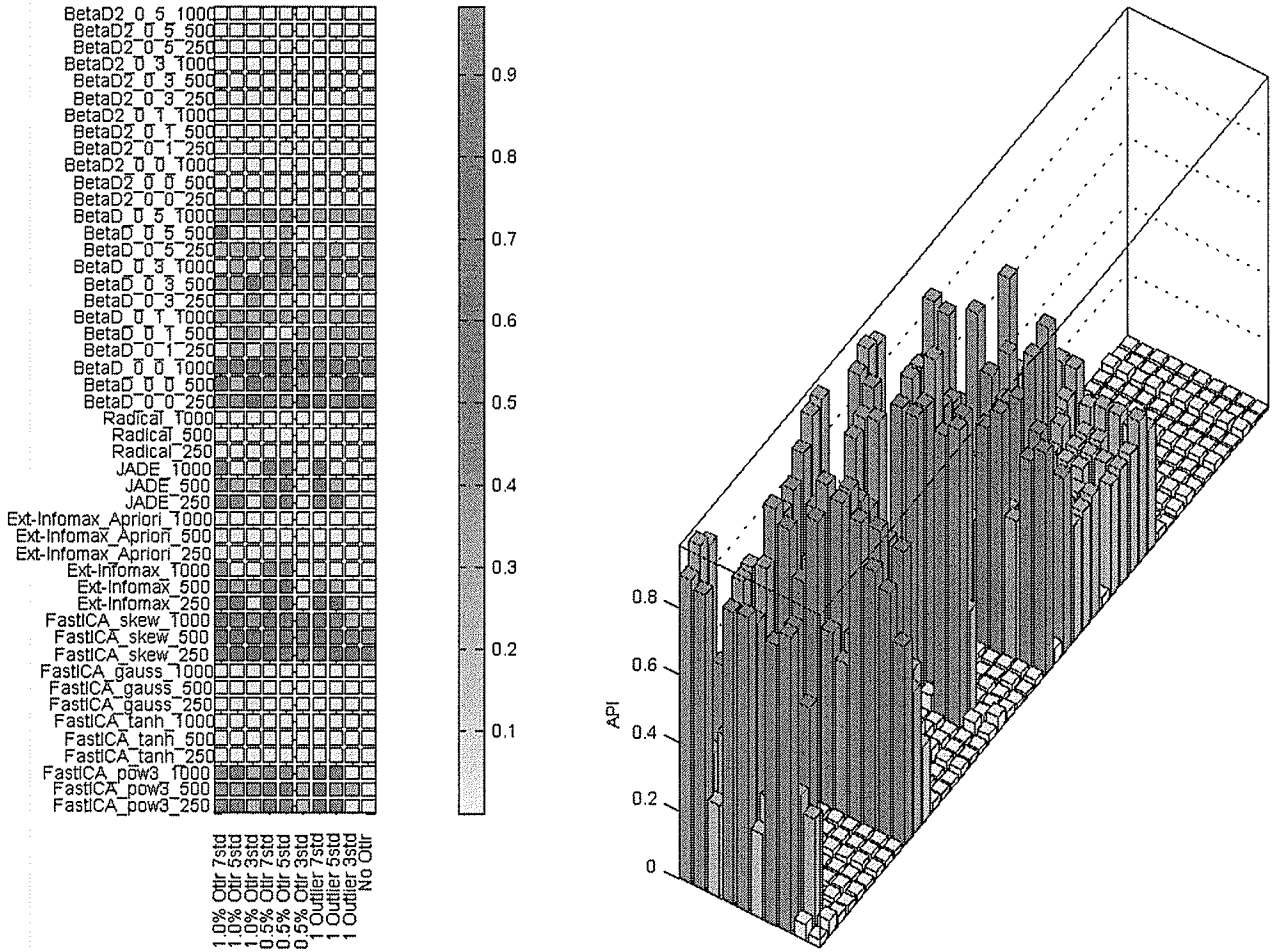


Fig. A.6 API of mixture benchmarking simulation: Mixture of 2 double exponentials.

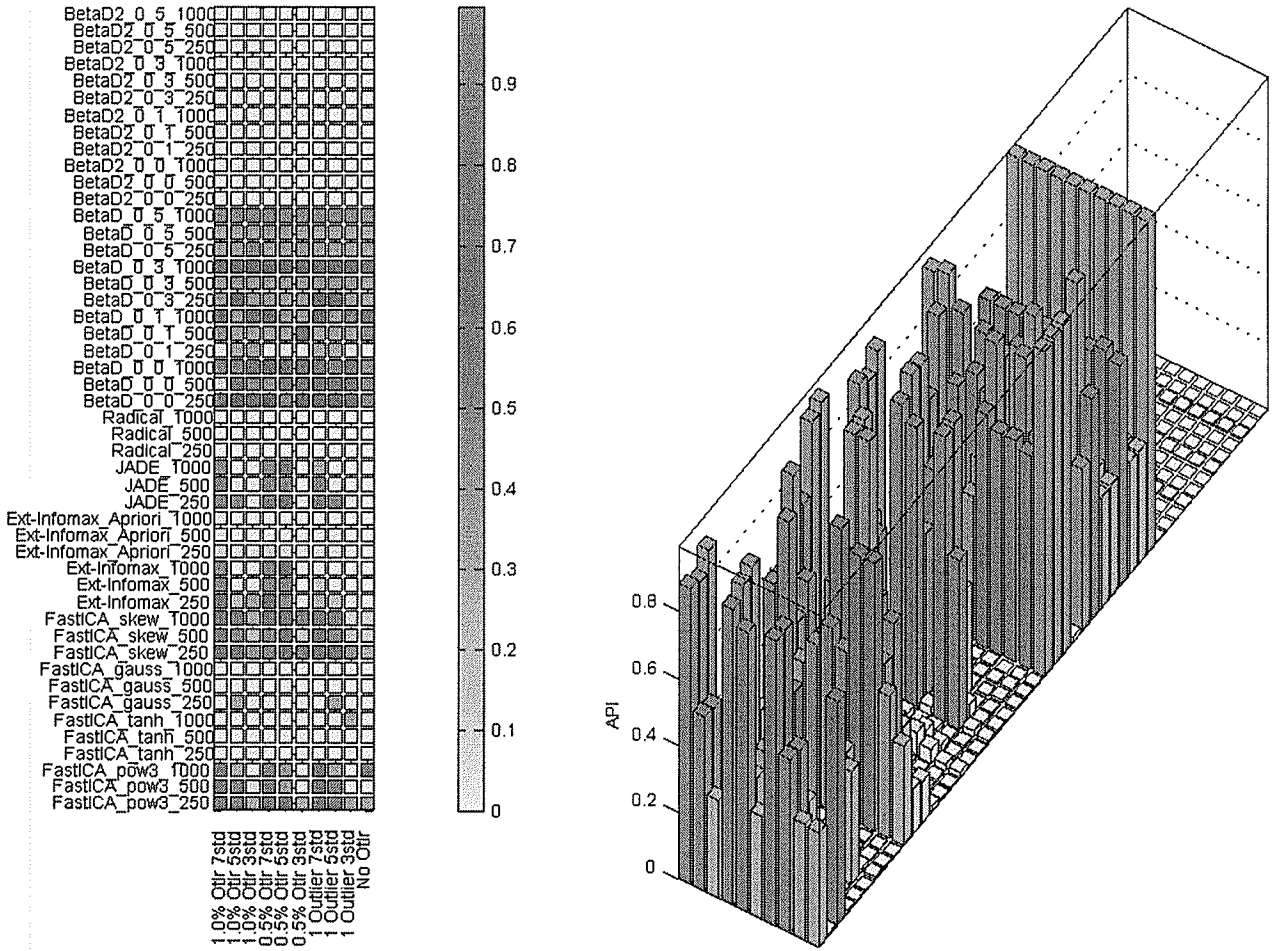


Fig. A.7 API of mixture benchmarking simulation: Symmetric mixture of 2 Gaussians (Multimodal).

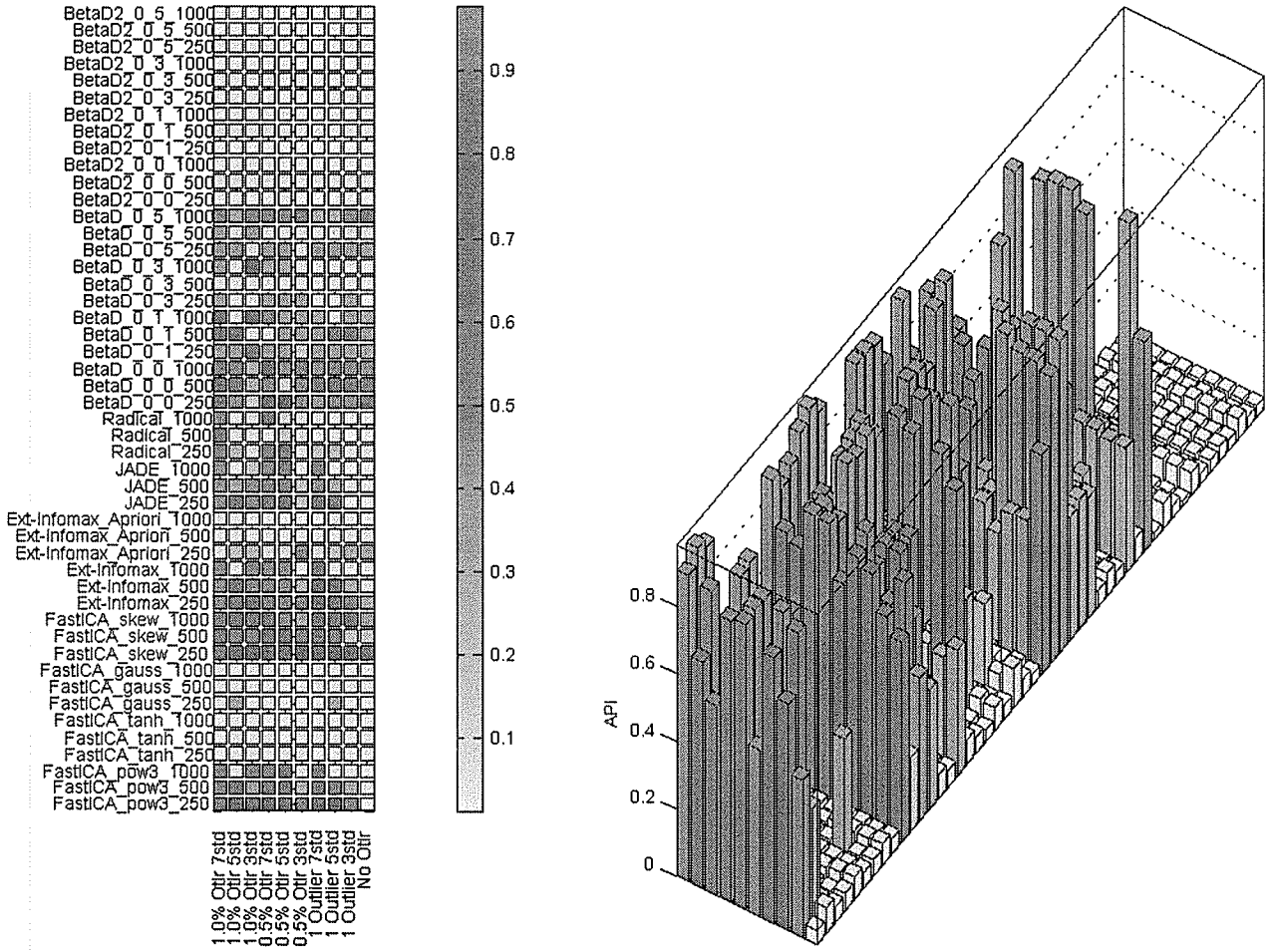


Fig. A.8 API of mixture benchmarking simulation: Symmetric mixture of 2 Gaussians (Transitional).

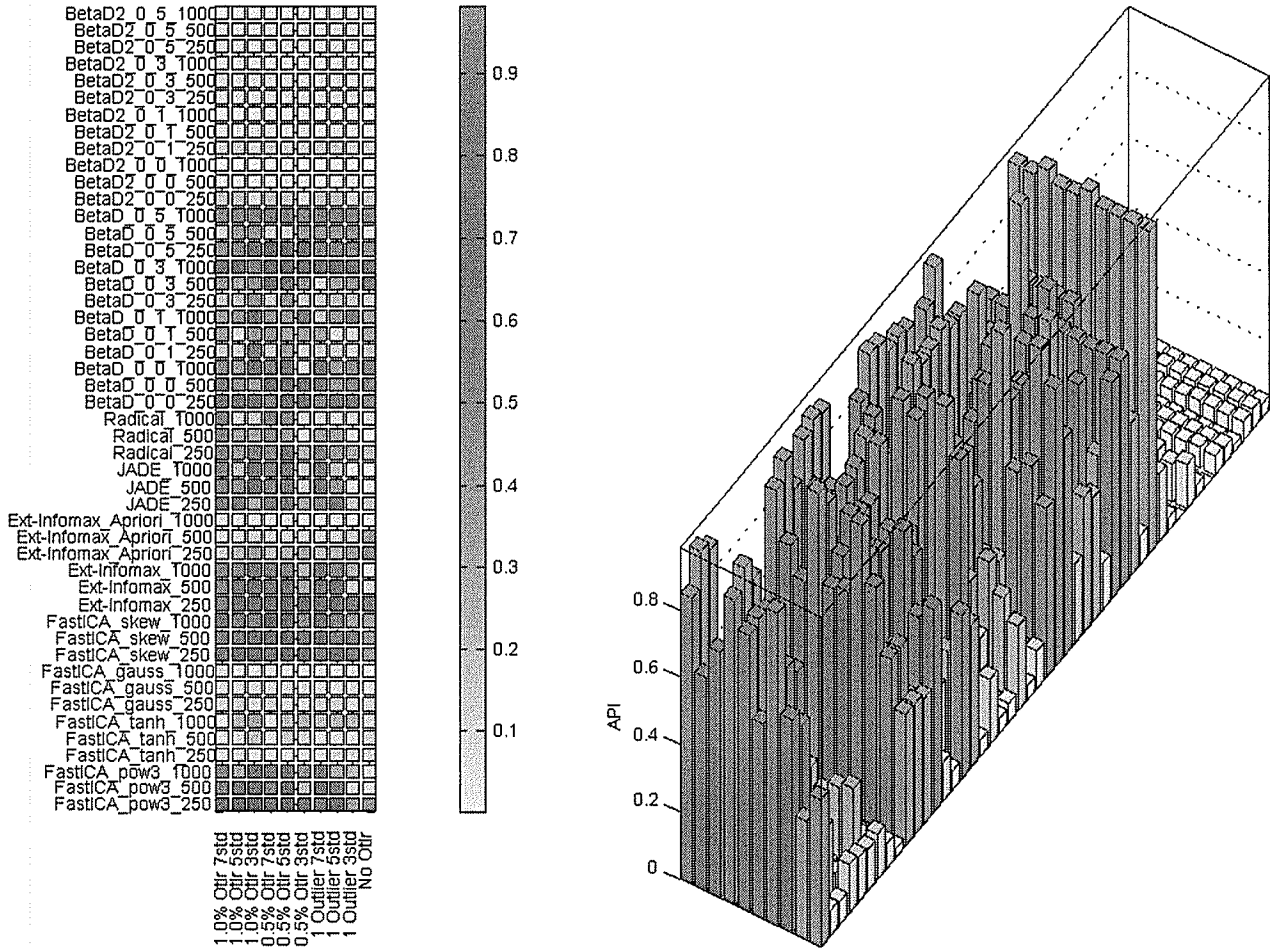


Fig. A.9 API of mixture benchmarking simulation: Symmetric mixture of 2 Gaussians (Unimodal).

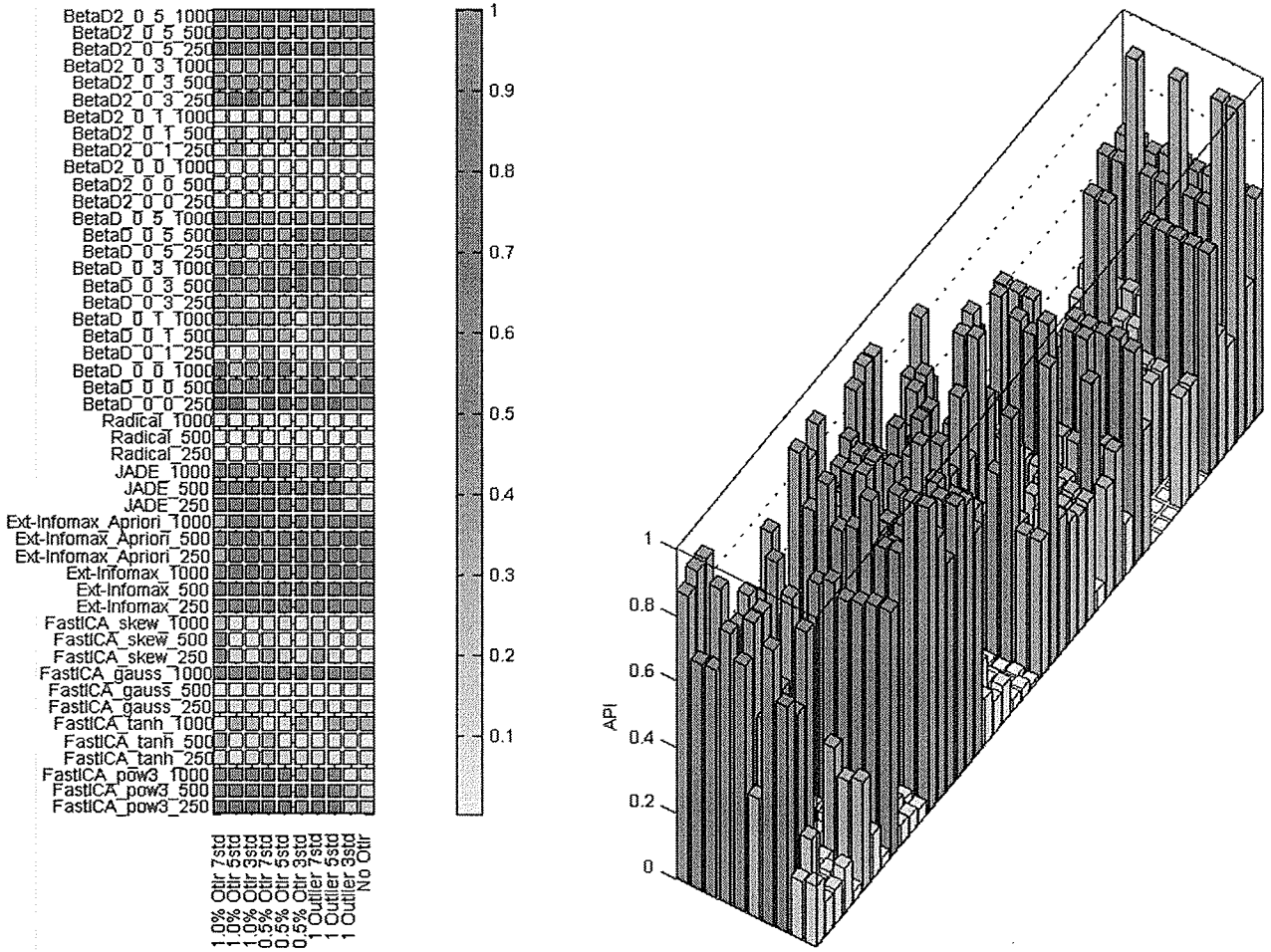


Fig. A.10 API of mixture benchmarking simulation: Asymmetric mixture of 2 Gaussians (Multimodal).

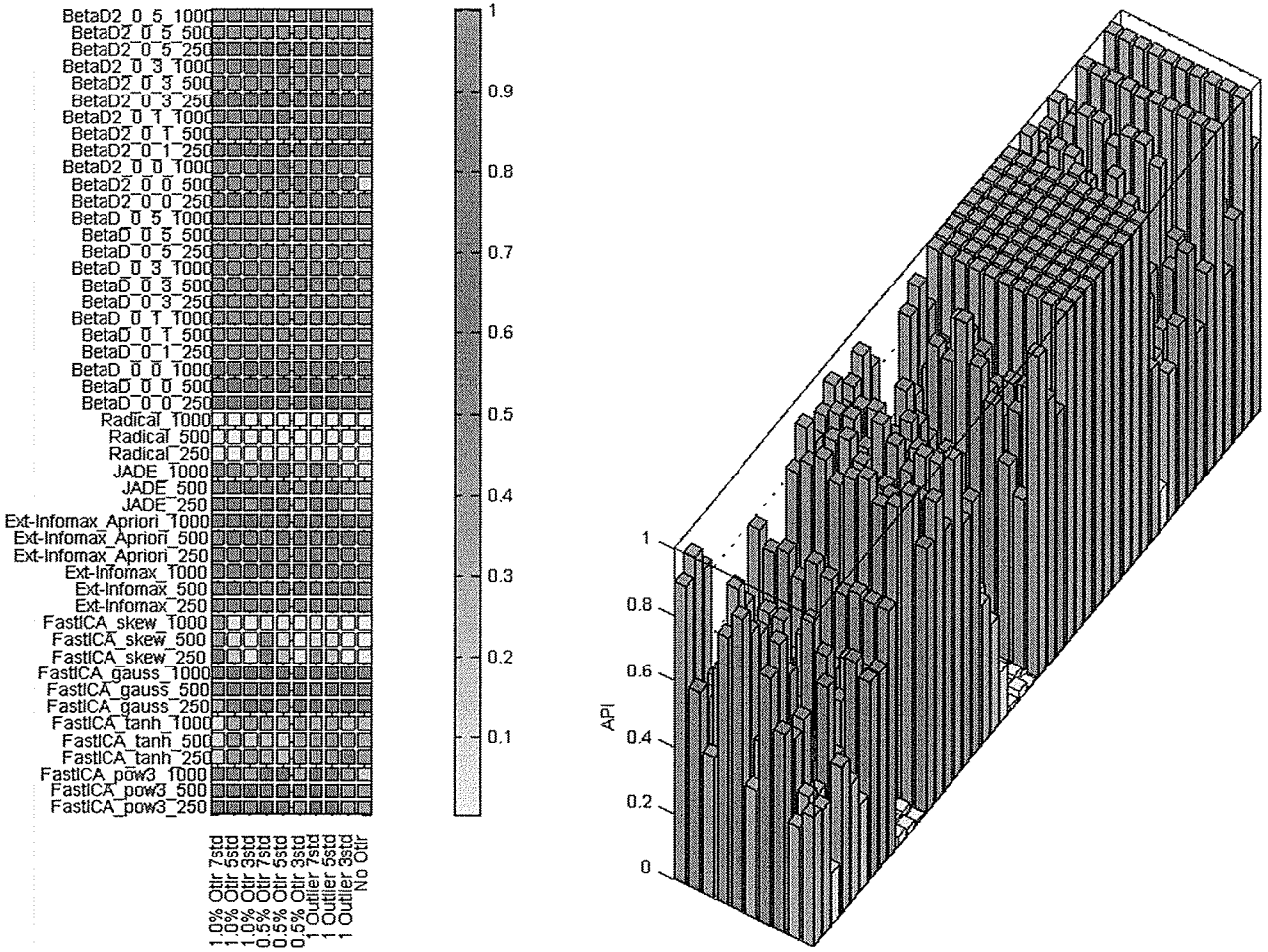


Fig. A.11 API of mixture benchmarking simulation: Asymmetric mixture of 2 Gaussians (Transitional).

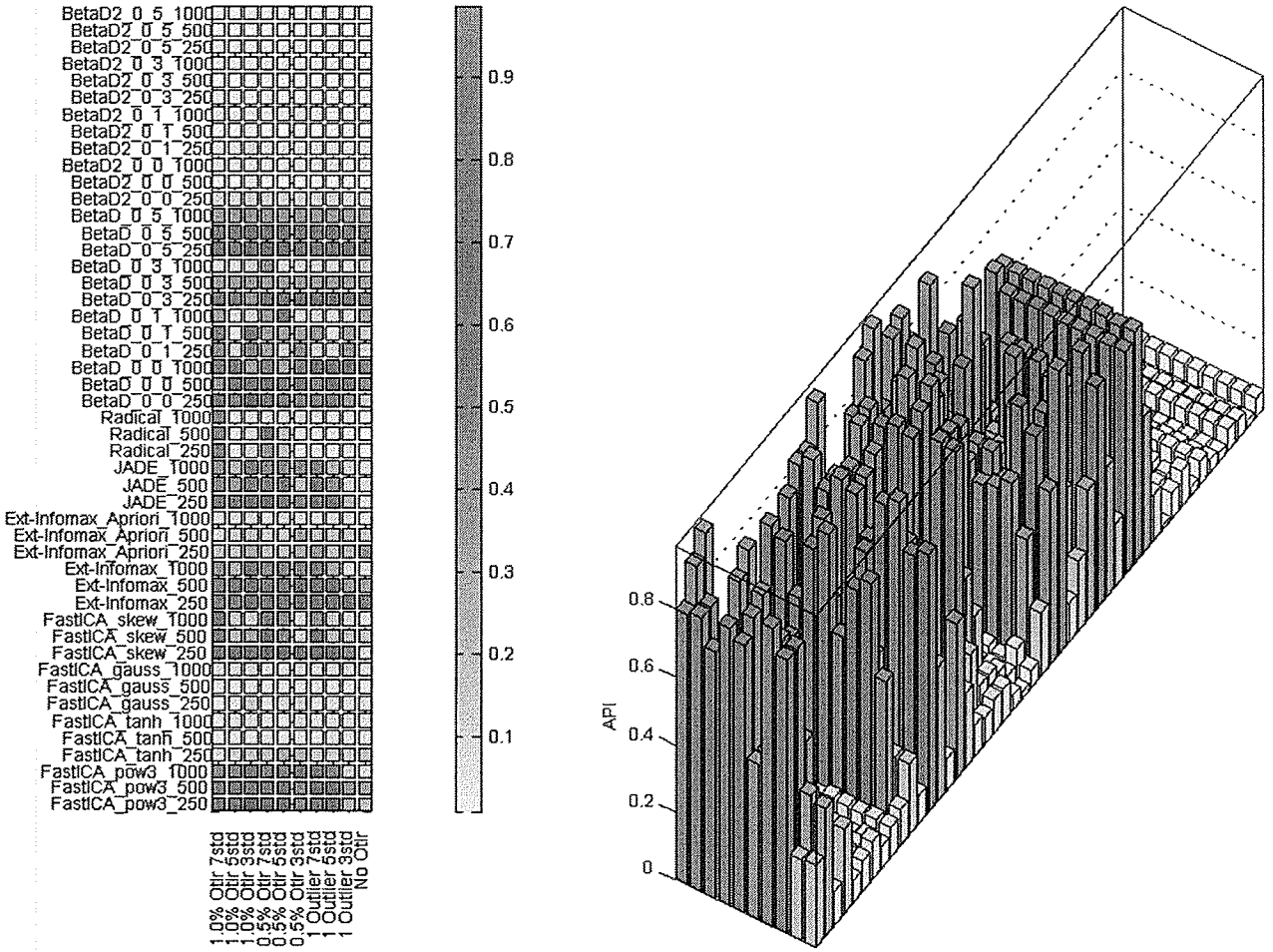


Fig. A.12 API of mixture benchmarking simulation: Asymmetric mixture of 2 Gaussians (Unimodal).

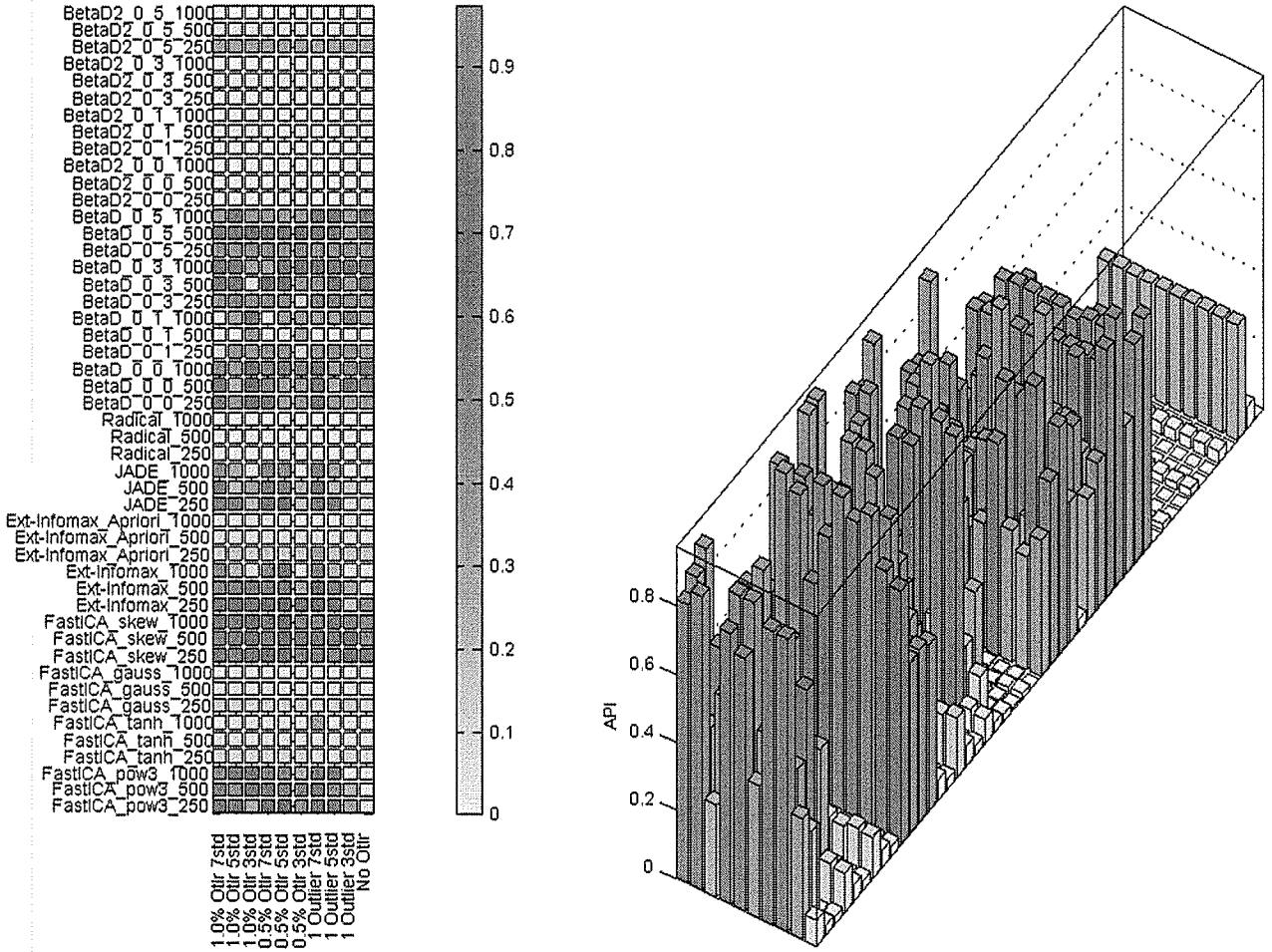


Fig. A.13 API of mixture benchmarking simulation: Symmetric mixture of 4 Gaussians (Multimodal).

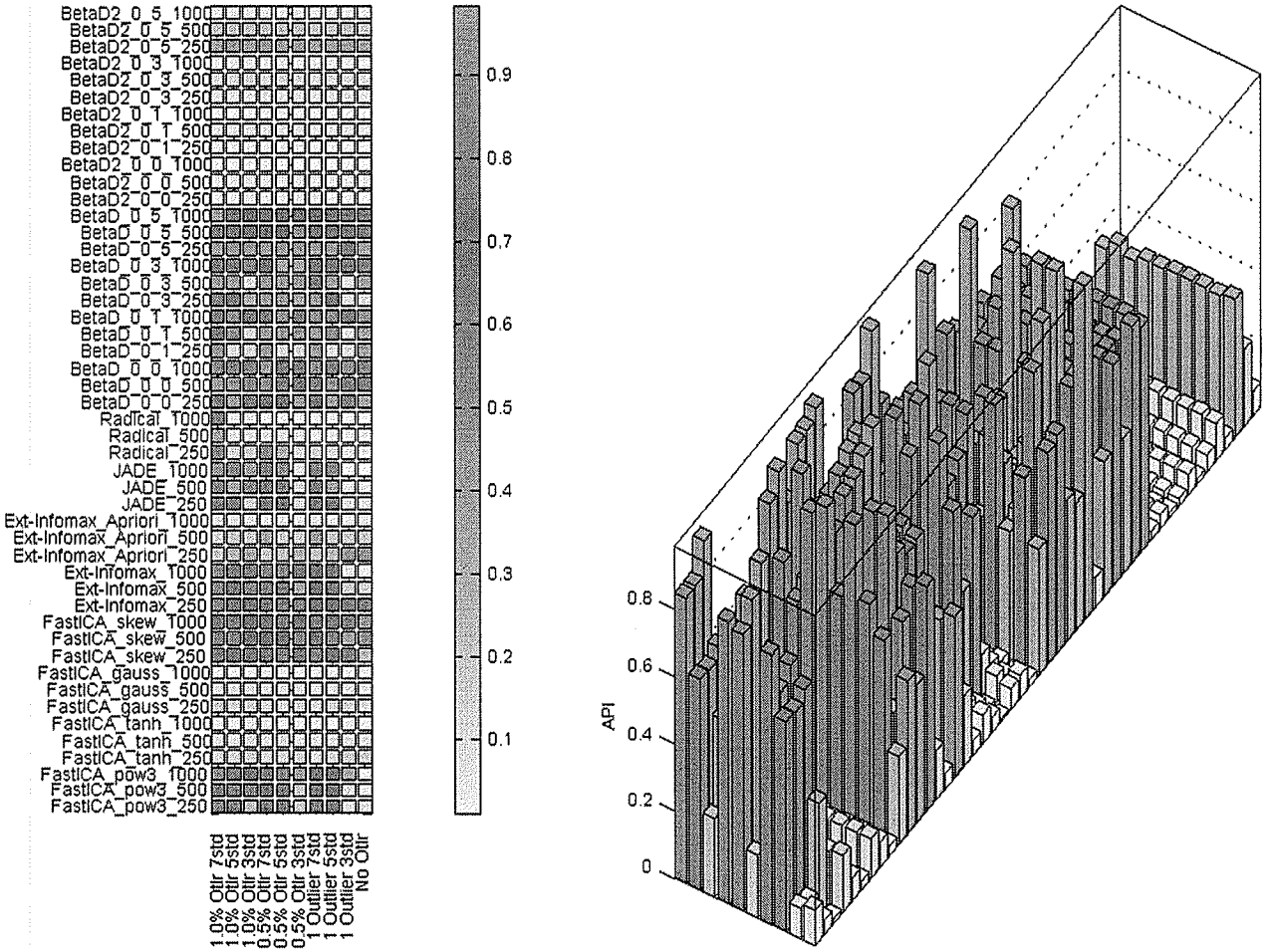


Fig. A.14 API of mixture benchmarking simulation: Symmetric mixture of 4 Gaussians (Transitional).

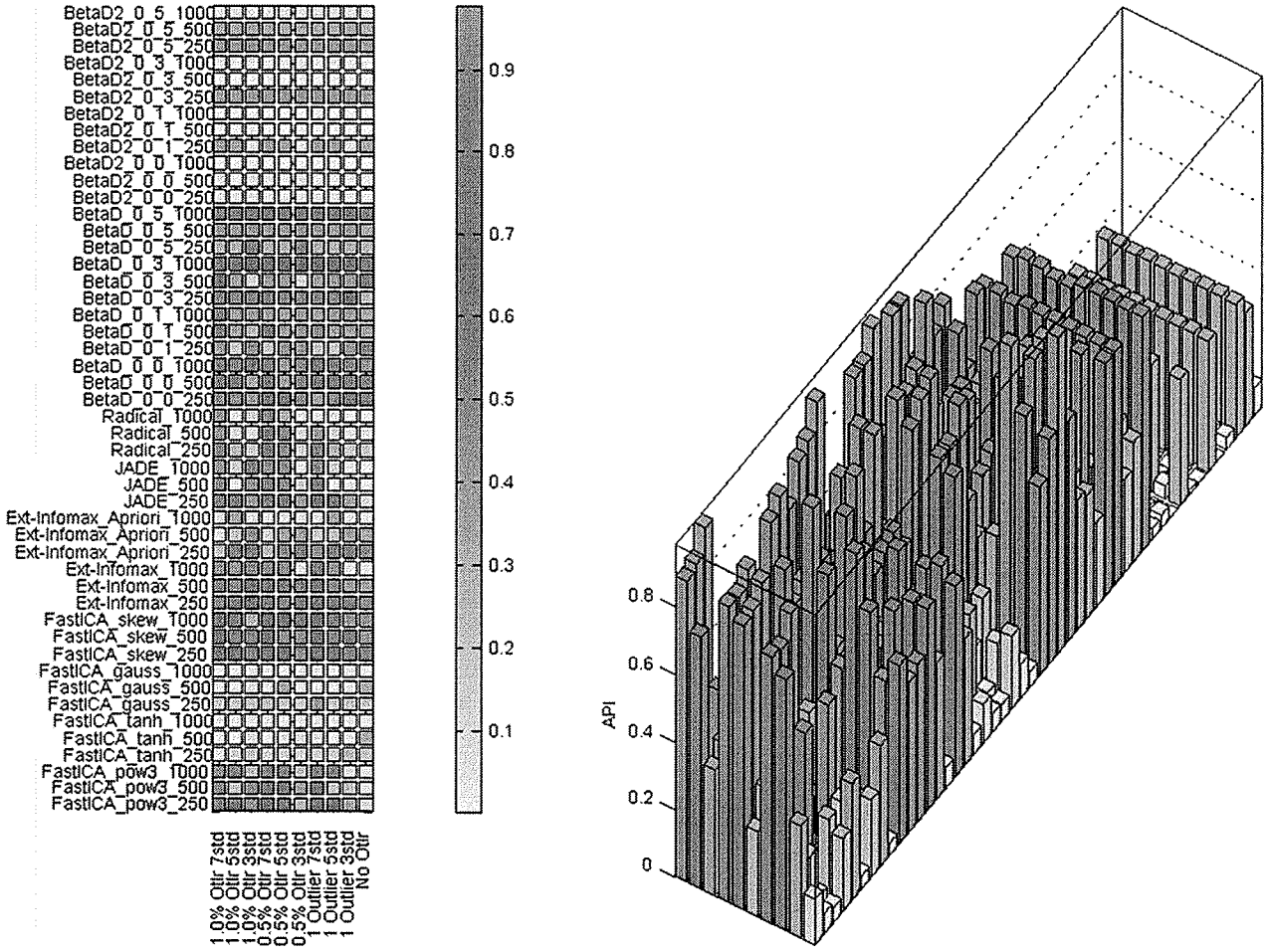


Fig. A.15 API of mixture benchmarking simulation: Symmetric mixture of 4 Gaussians (Unimodal).

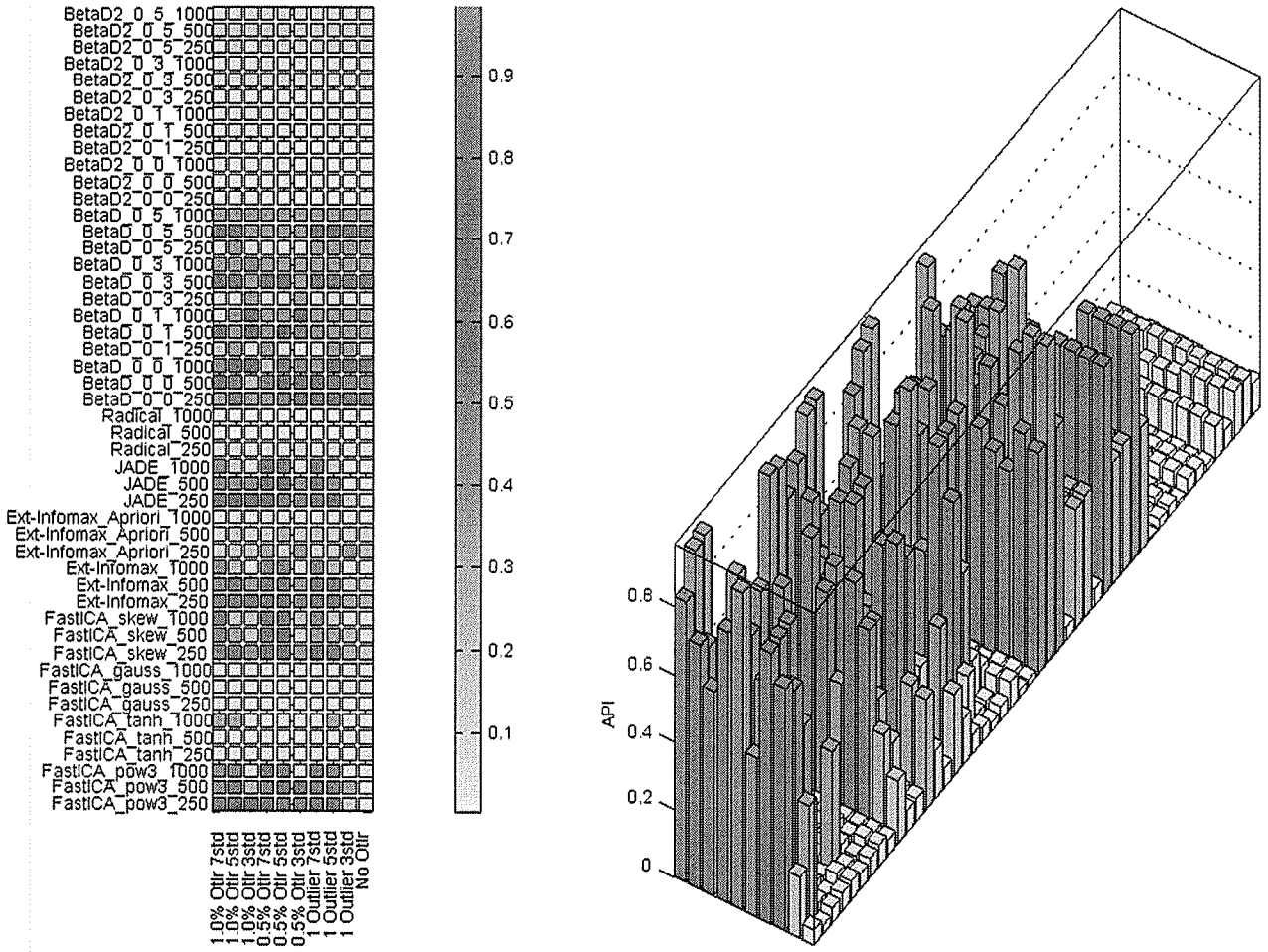


Fig. A.16 API of mixture benchmarking simulation: Asymmetric mixture of 4 Gaussians (Multimodal).

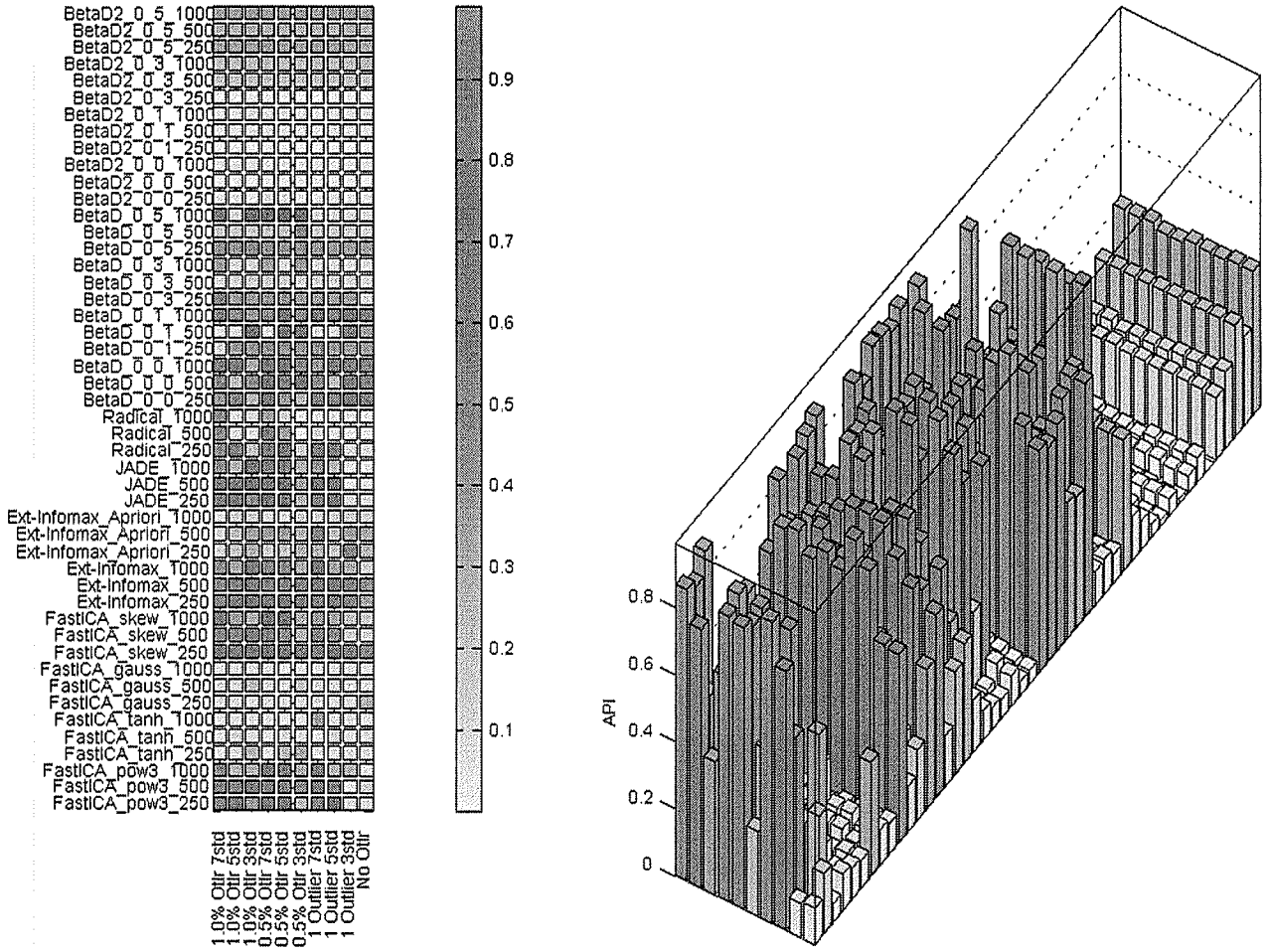


Fig. A.17 API of mixture benchmarking simulation: Asymmetric mixture of 4 Gaussians (Transitional).

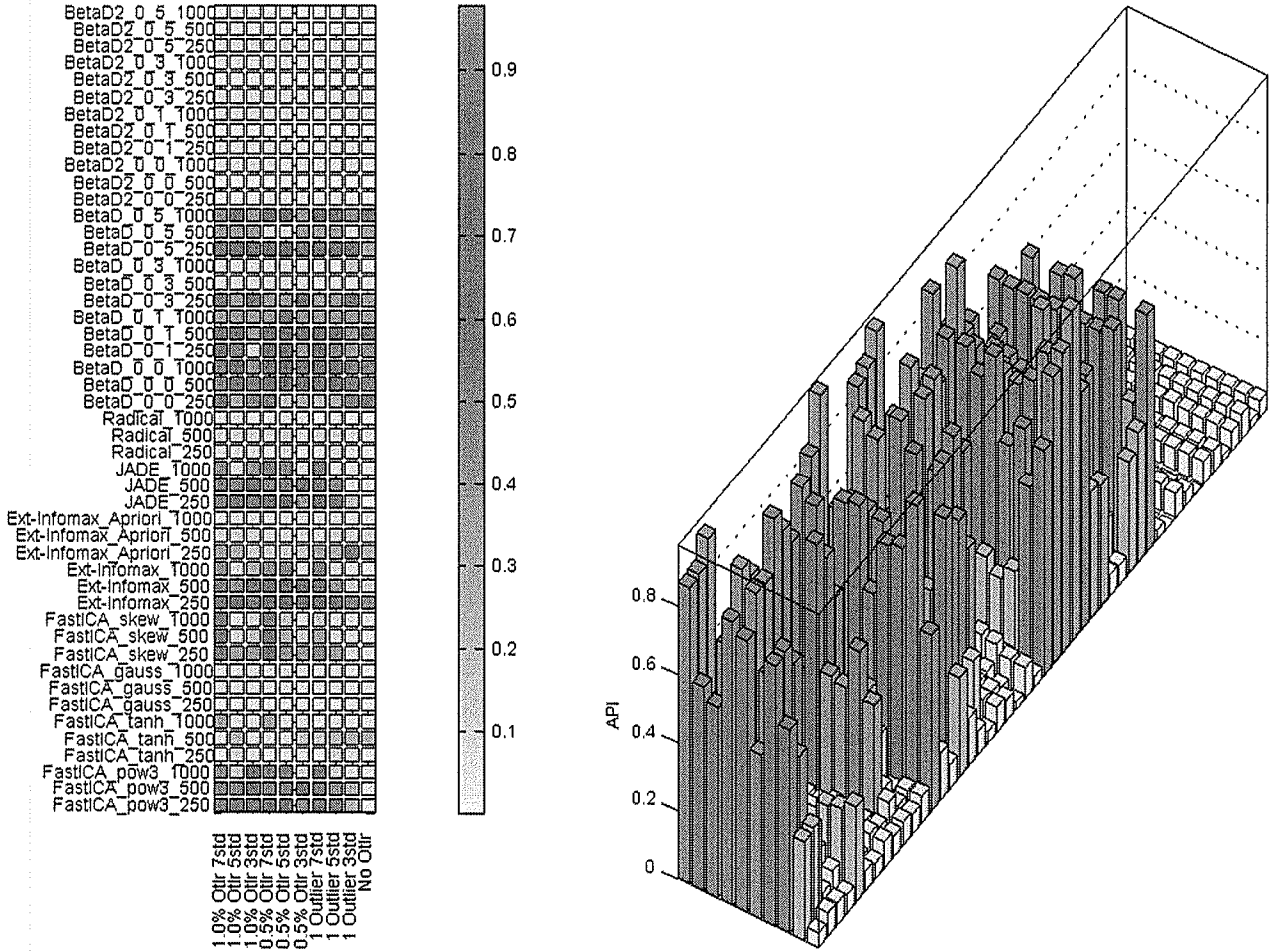


Fig. A.18 API of mixture benchmarking simulation: Asymmetric mixture of 4 Gaussians (Unimodal).

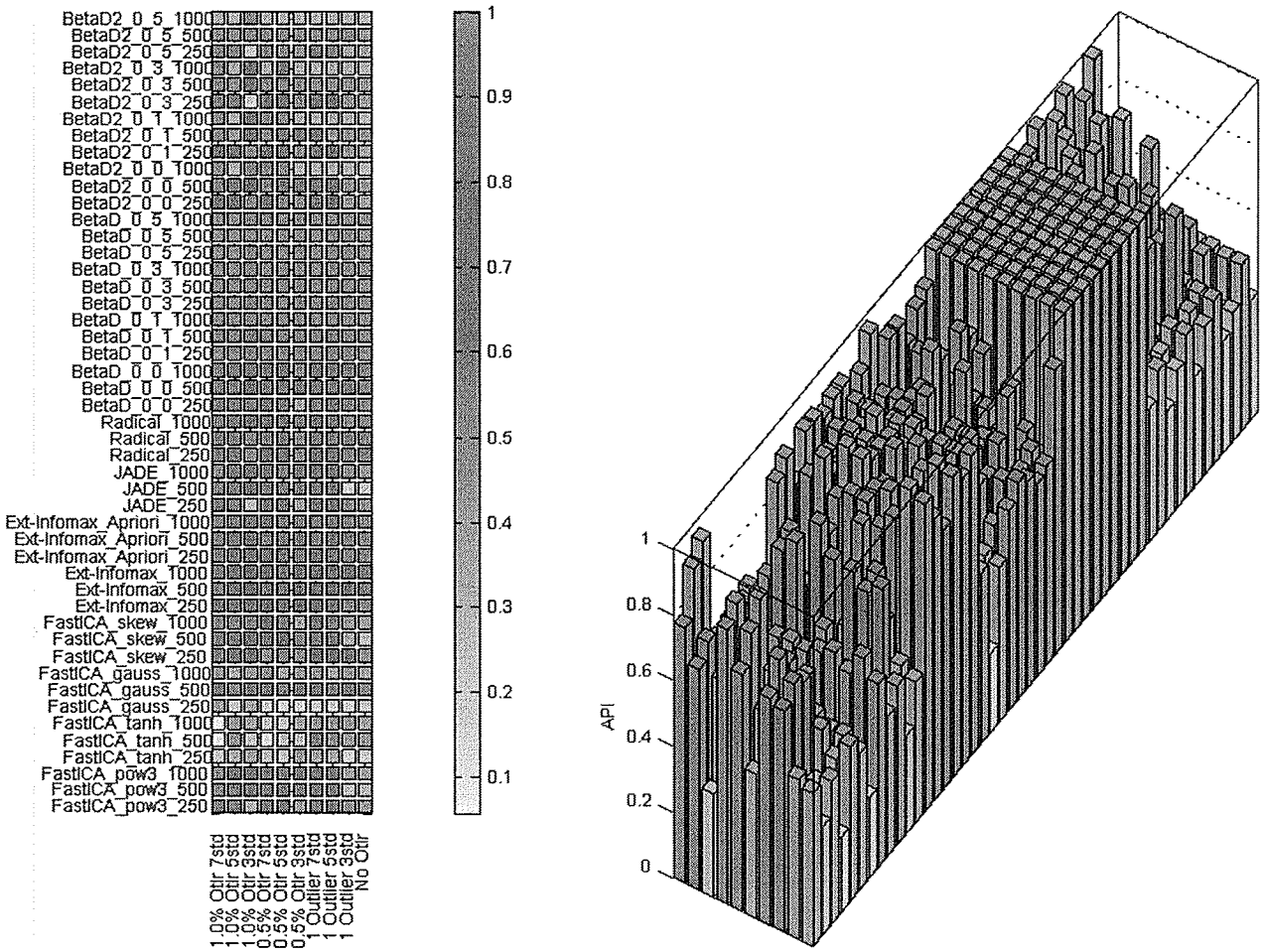


Fig. A.19 API of mixture benchmarking simulation: Gaussian.

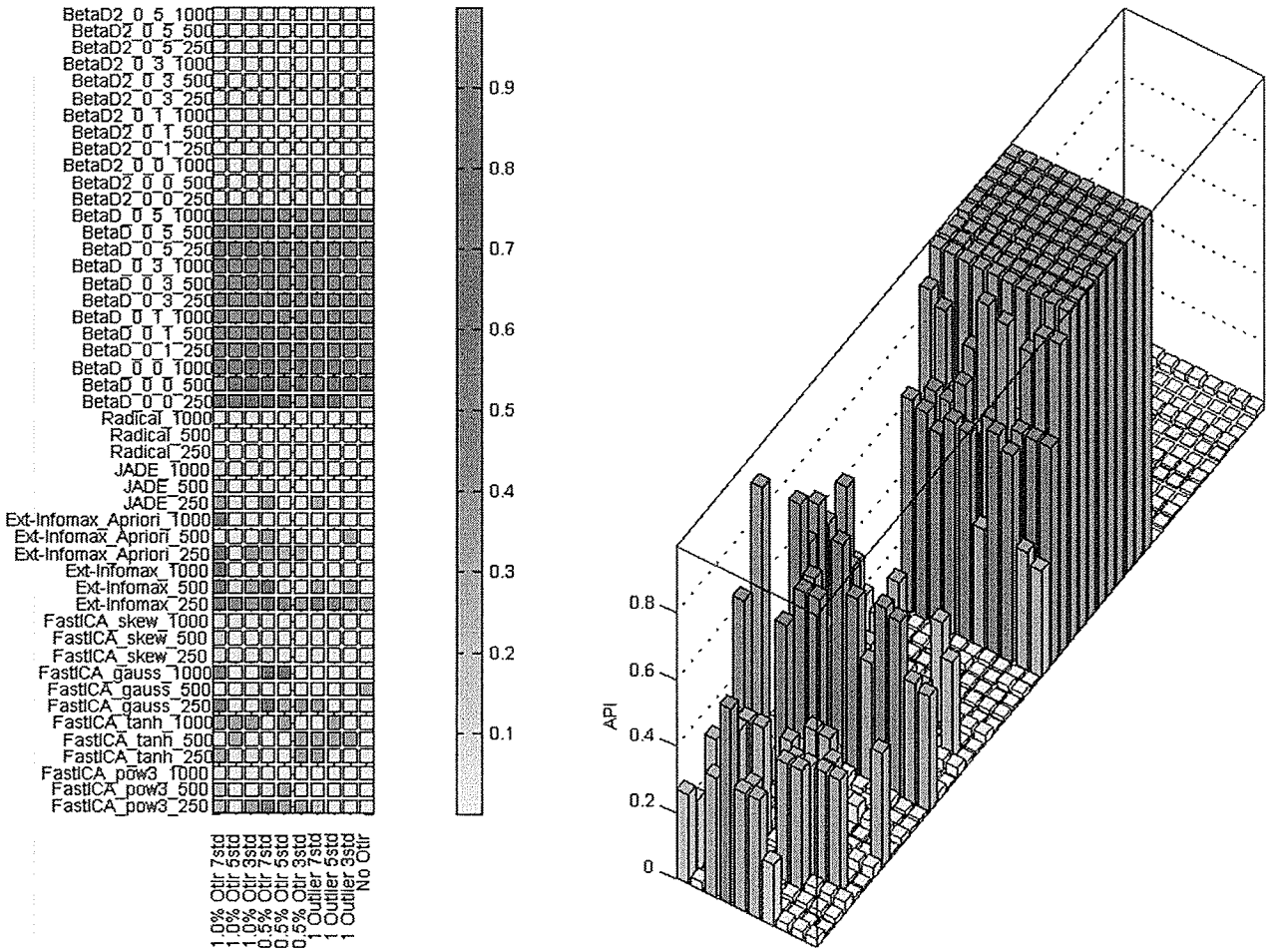


Fig. A.20 API of mixture benchmarking simulation: LogNormal.

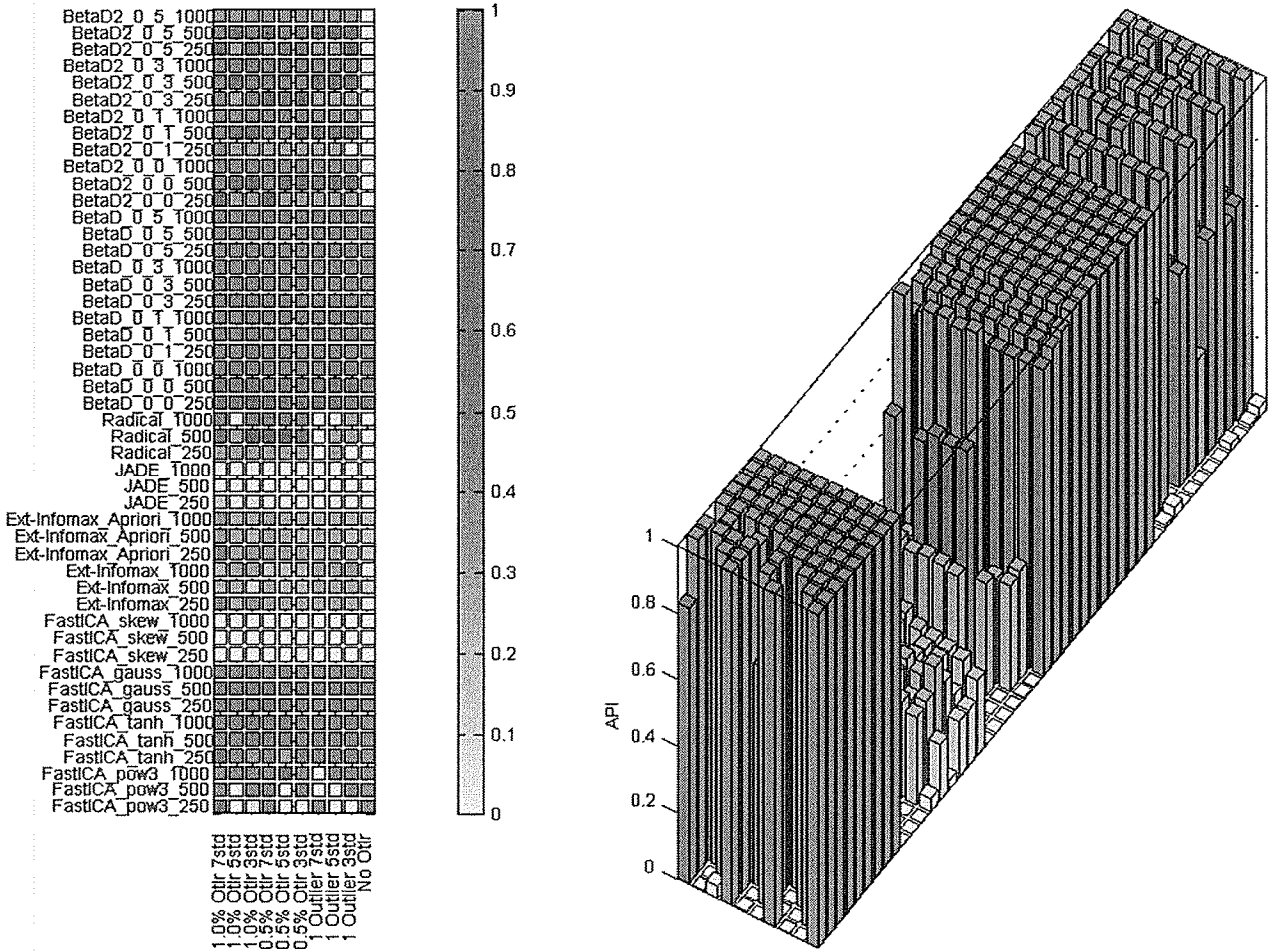


Fig. A.21 API of mixture benchmarking simulation: Pareto.

A.3 Rotation Error of Mixtures of Densities

The following sidewaysfigures contain the rotation error of the ICA contrast functions for mixture of the given density. The experiment setup is explained in Sec. 4.3.

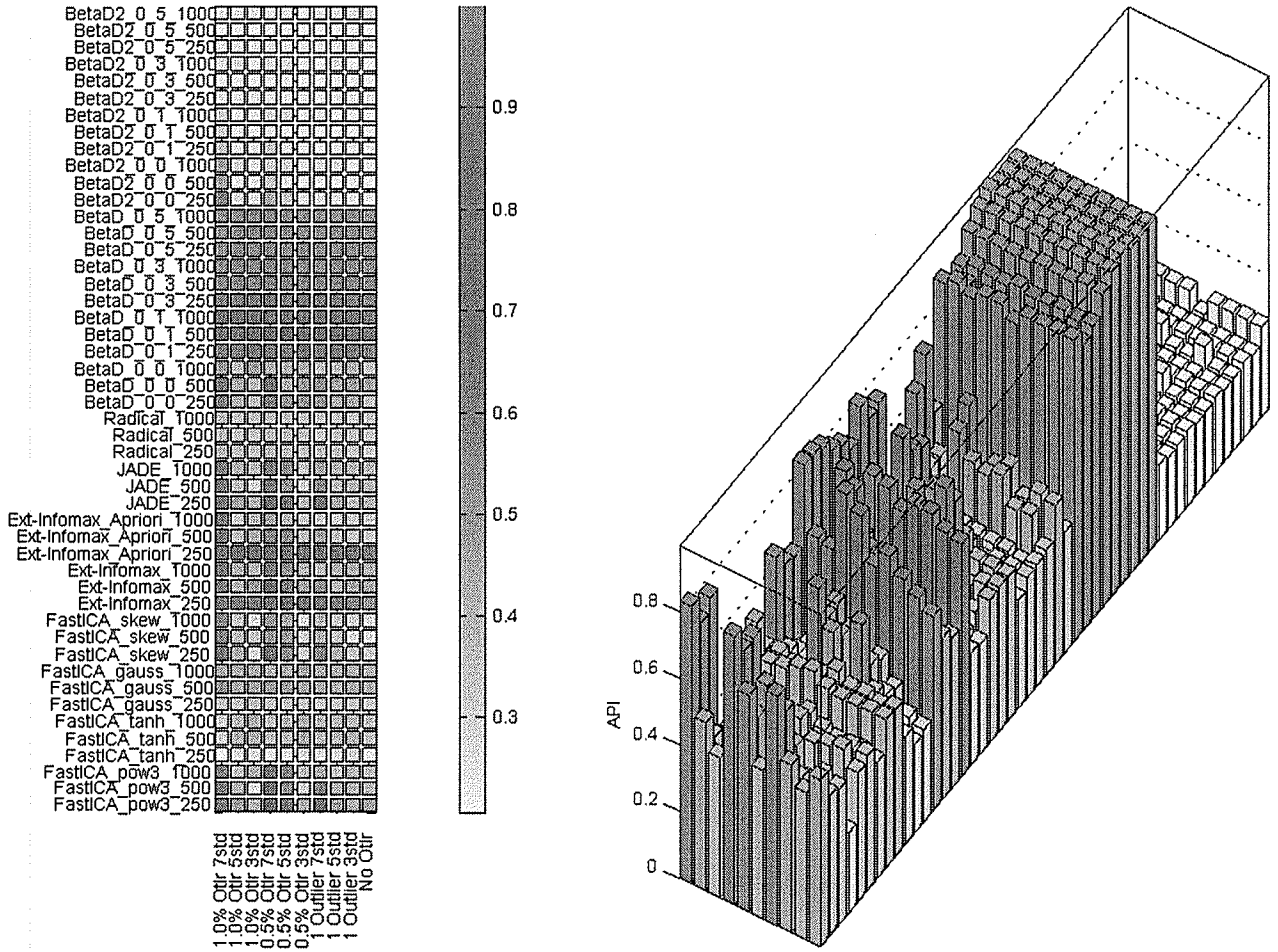


Fig. A.22 API of mixture benchmarking simulation: Random mixtures of all densities.

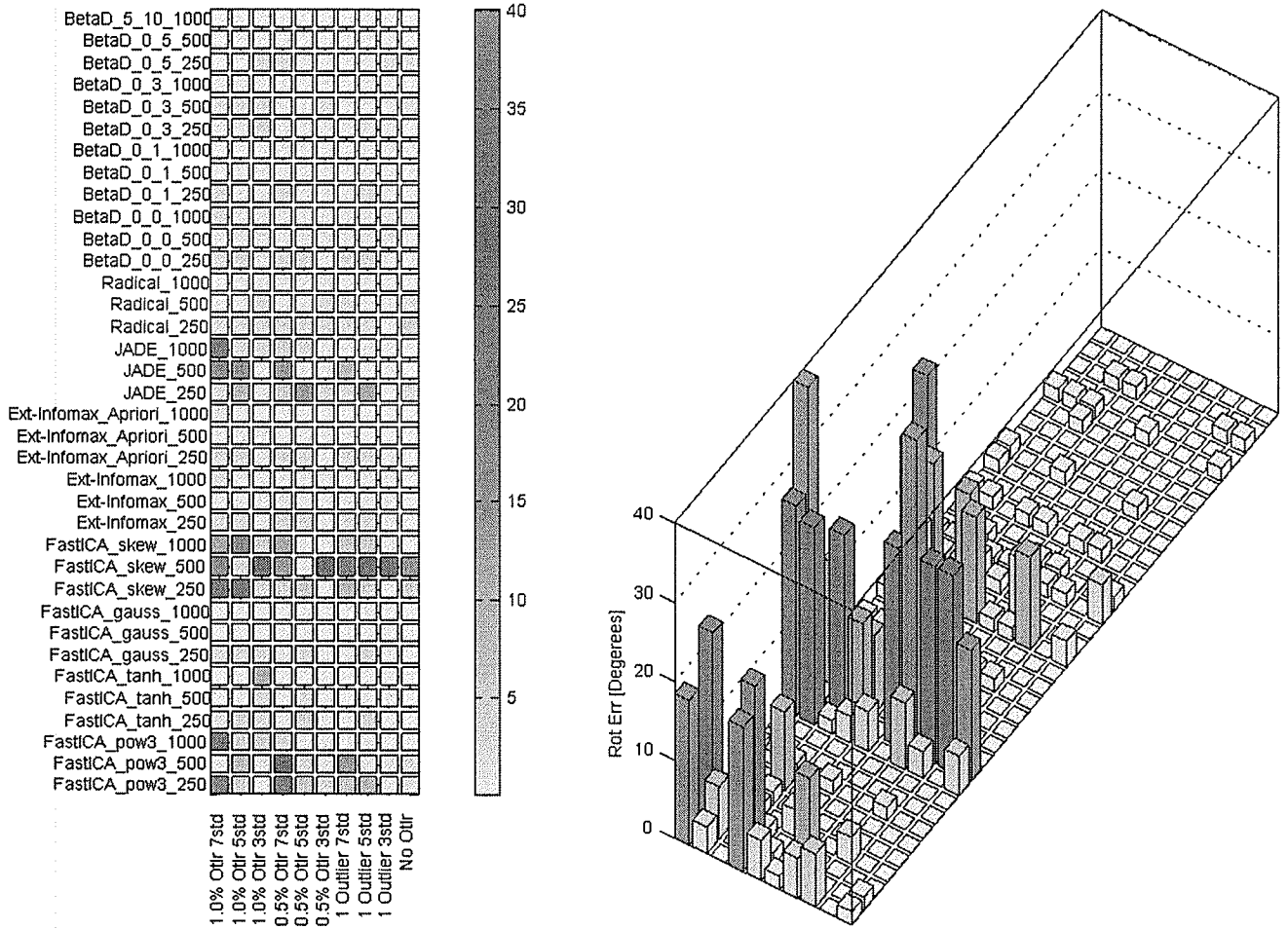


Fig. A.23 Rotation error: Student-t 3 degrees of freedom.

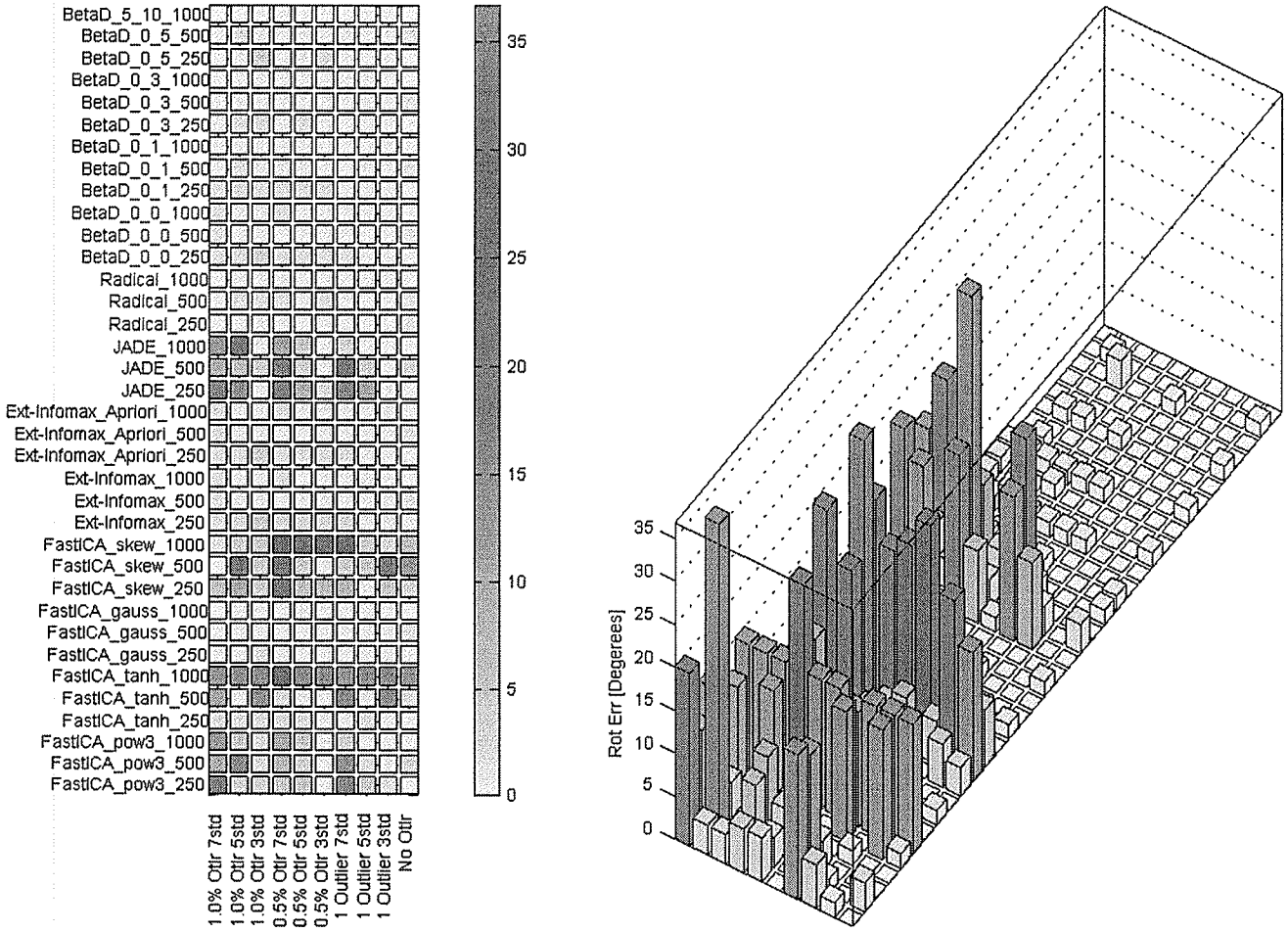


Fig. A.24 Rotation error: Double Exponential.

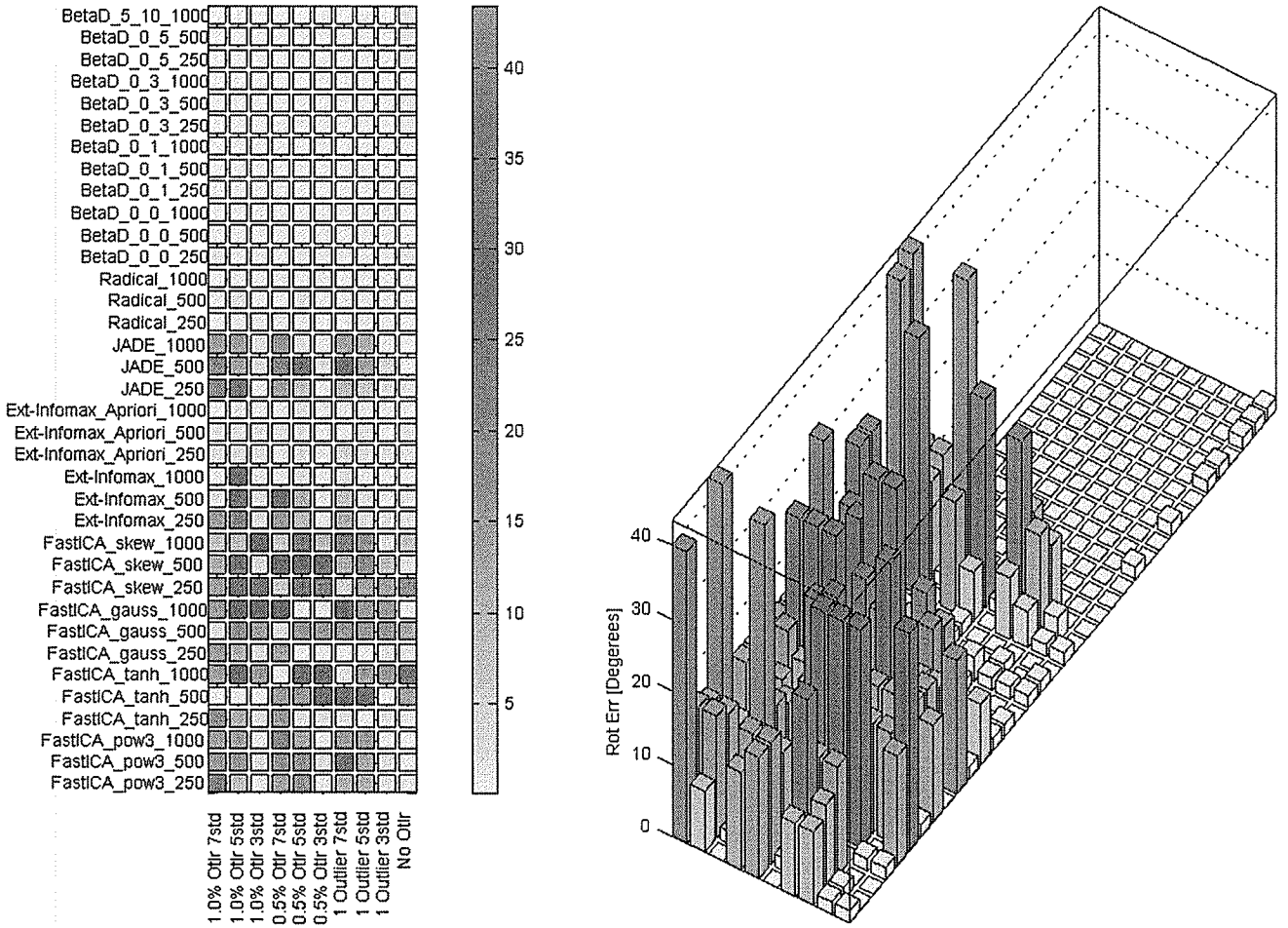


Fig. A.25 Rotation error: Uniform.

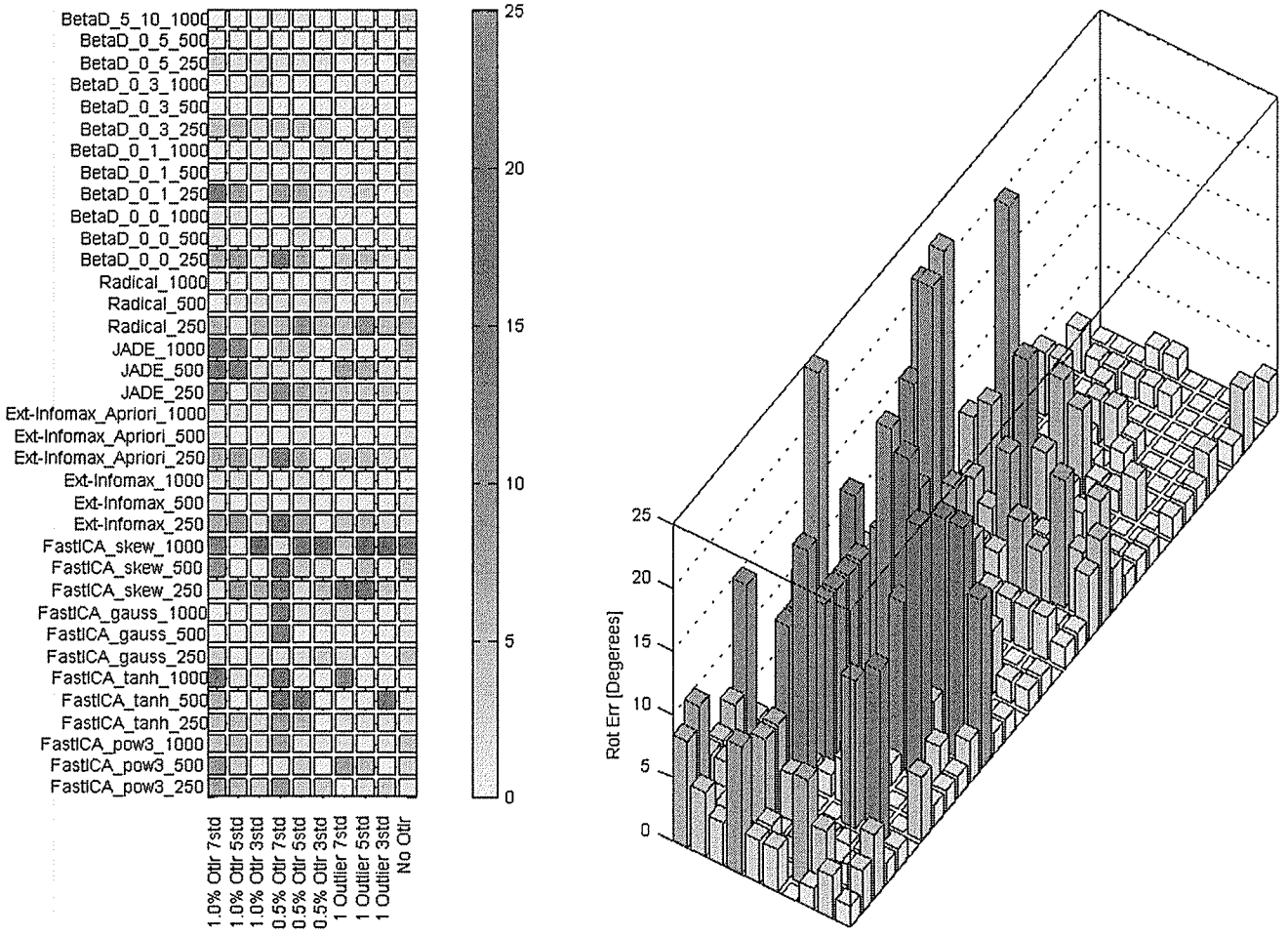


Fig. A.26 Rotation error: Student-t 5 degrees of freedom.

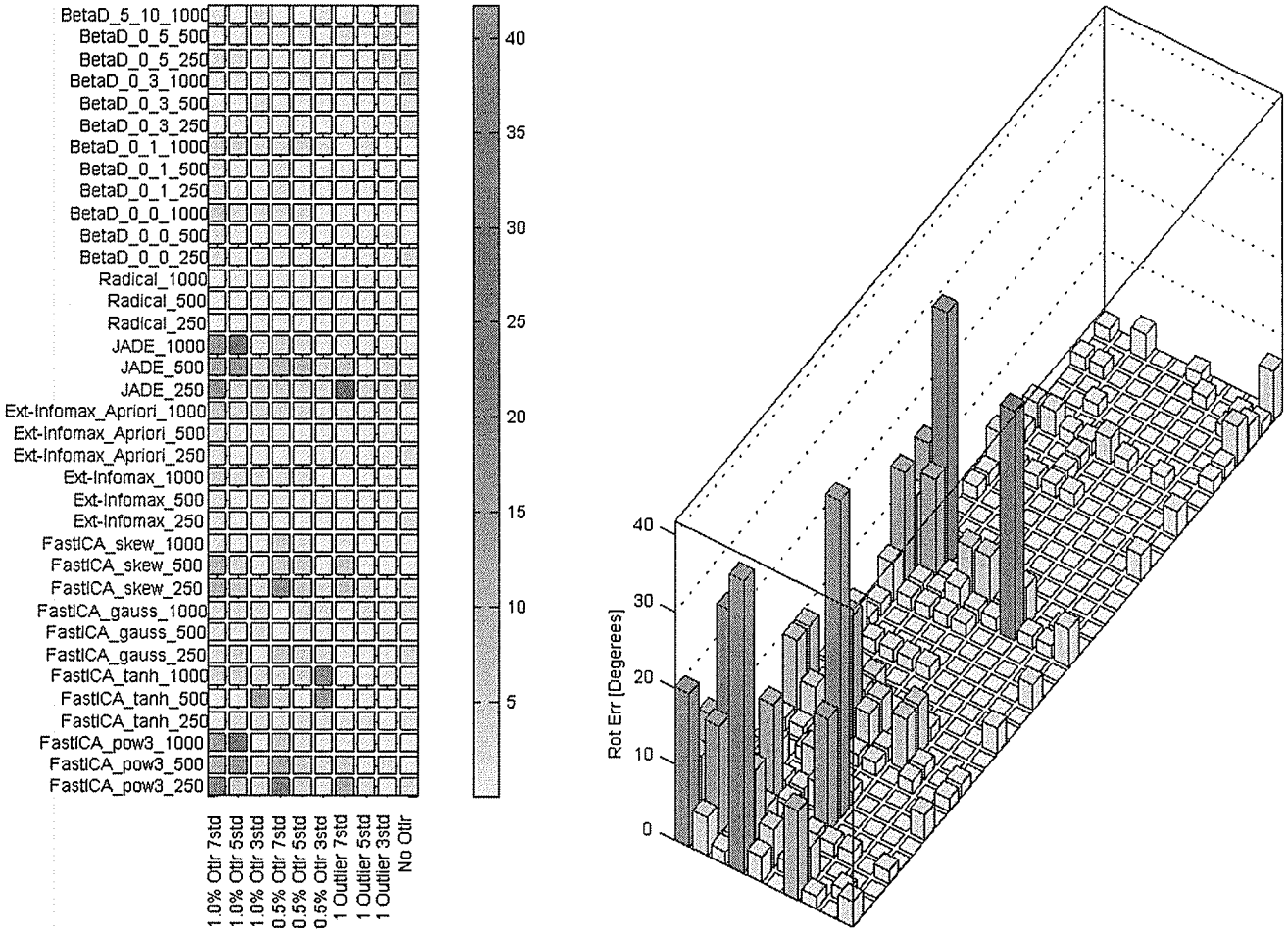


Fig. A.27 Rotation error: Exponential.

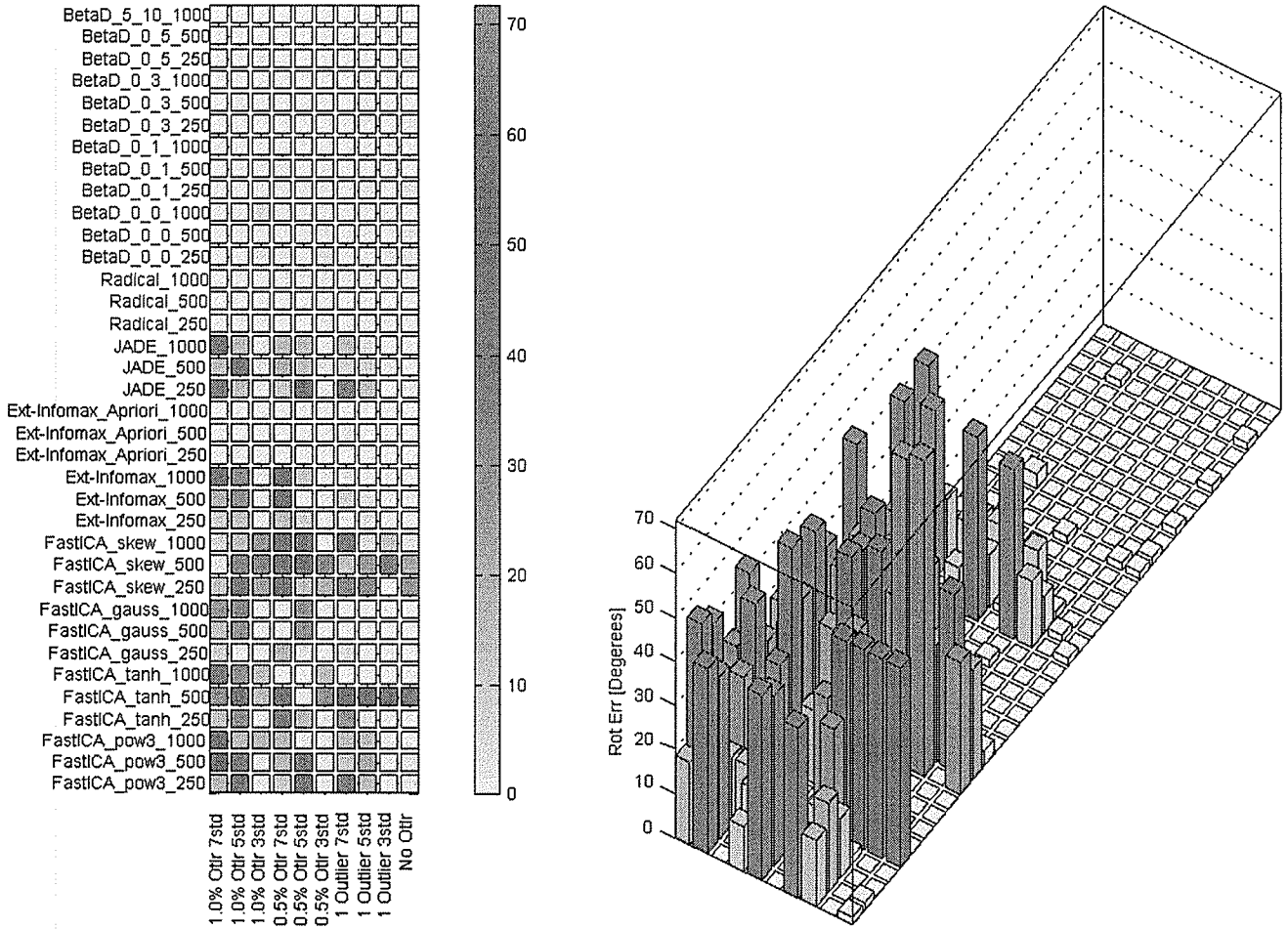


Fig. A.28 Rotation error: Mixture of 2 double exponentials.

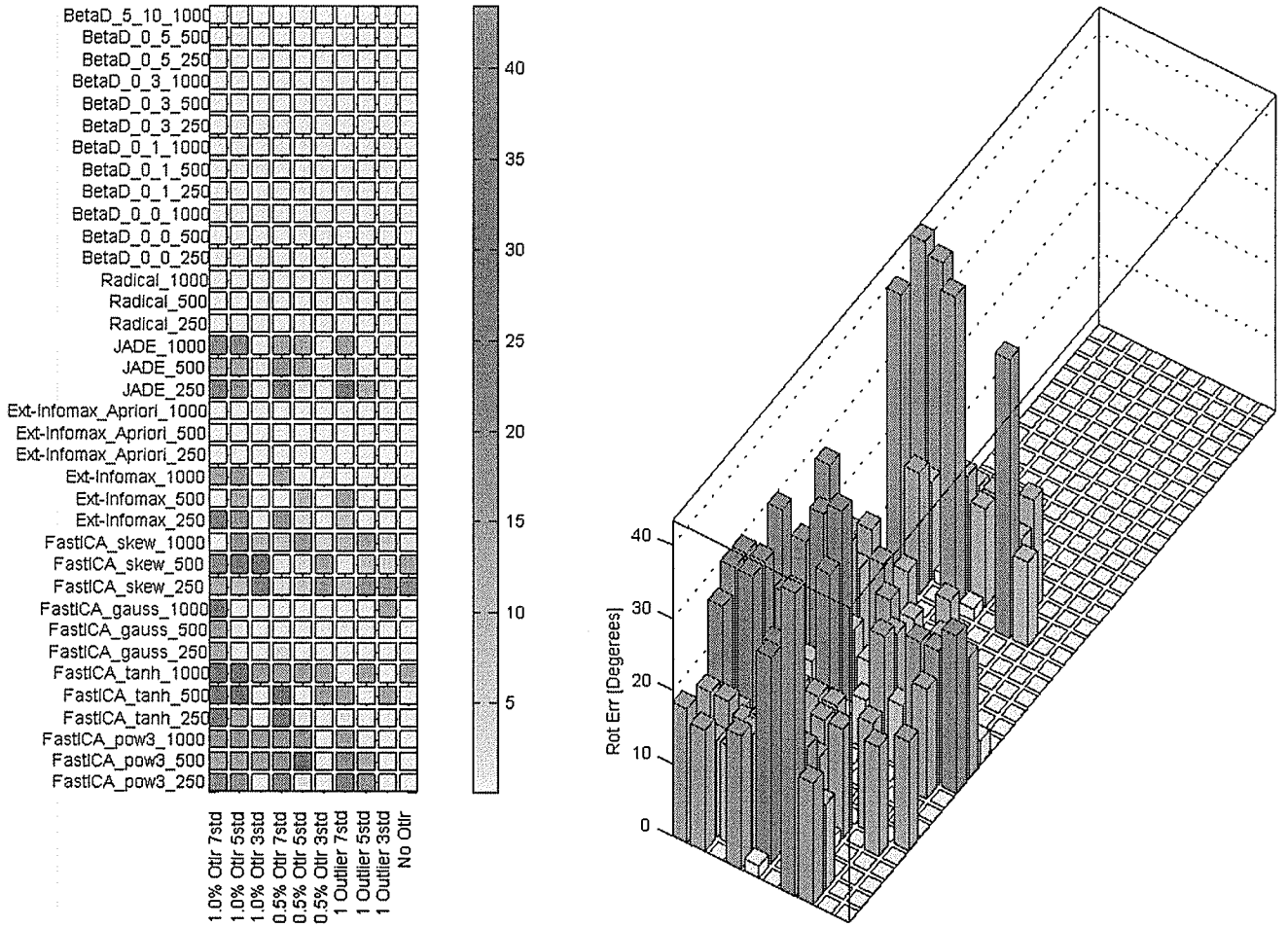


Fig. A.29 Rotation error: Symmetric mixture of 2 Gaussians (Multimodal).

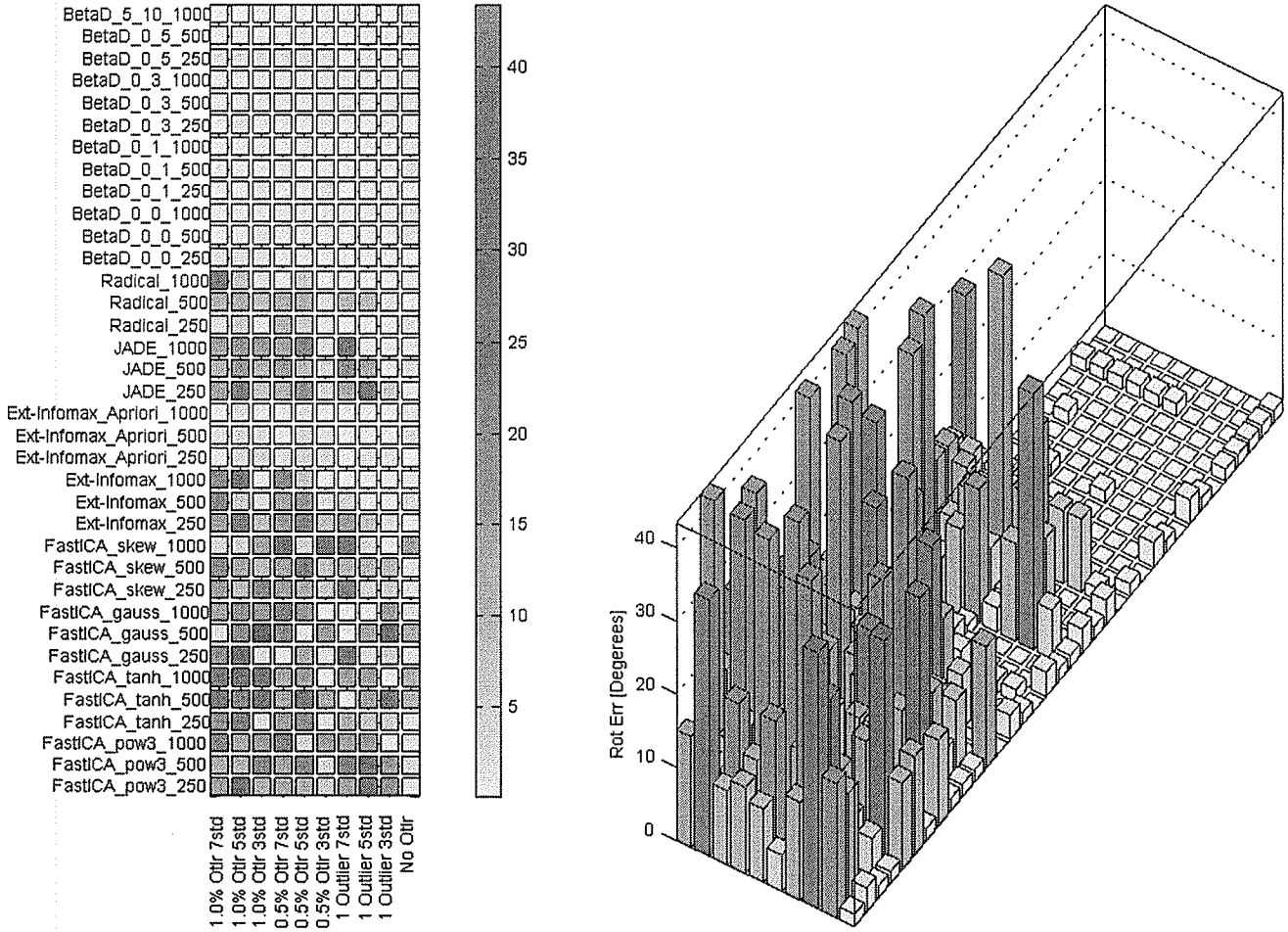


Fig. A.30 Rotation error: Symmetric mixture of 2 Gaussians (Transitional).

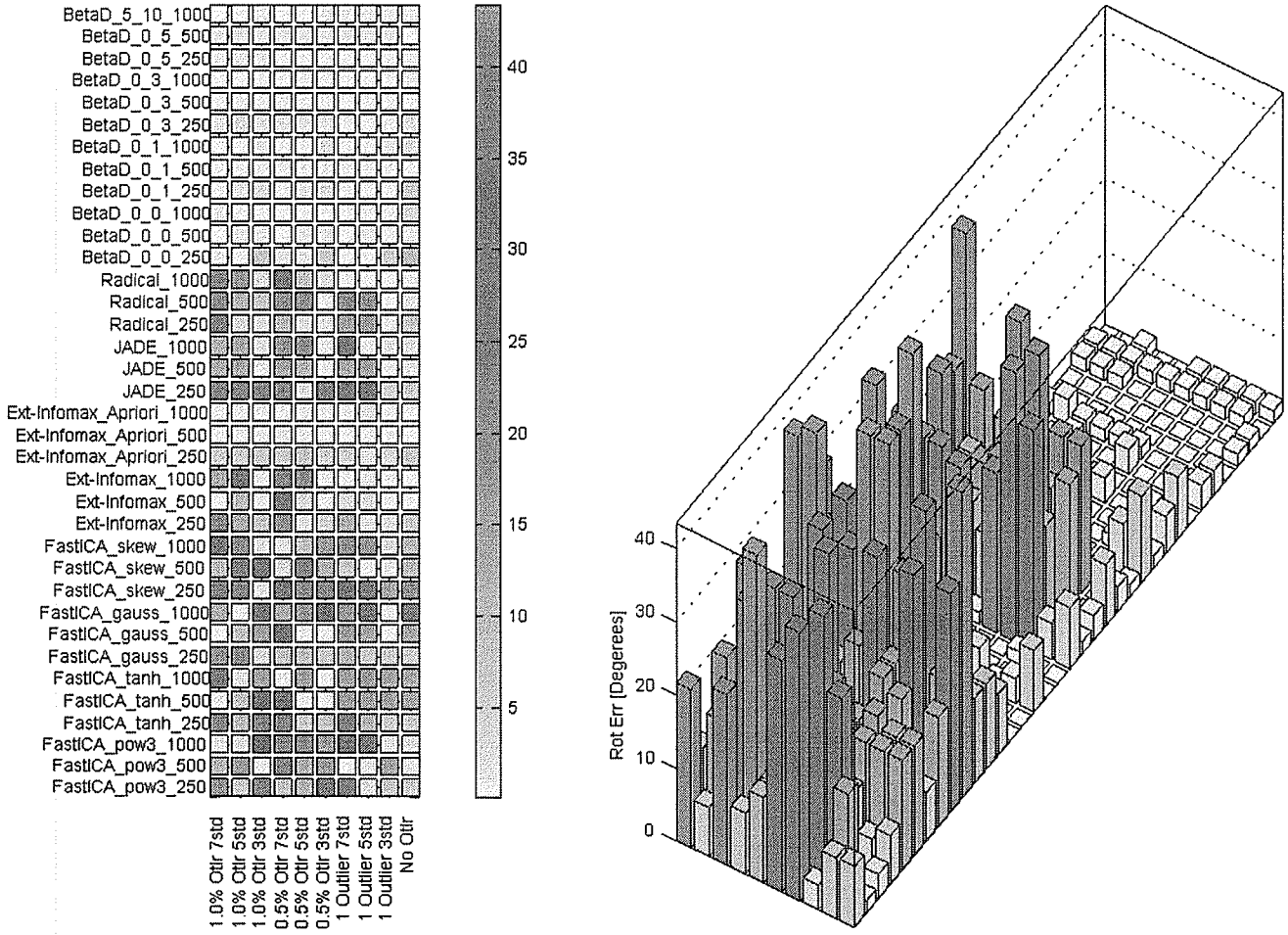


Fig. A.31 Rotation error: Symmetric mixture of 2 Gaussians (Unimodal).

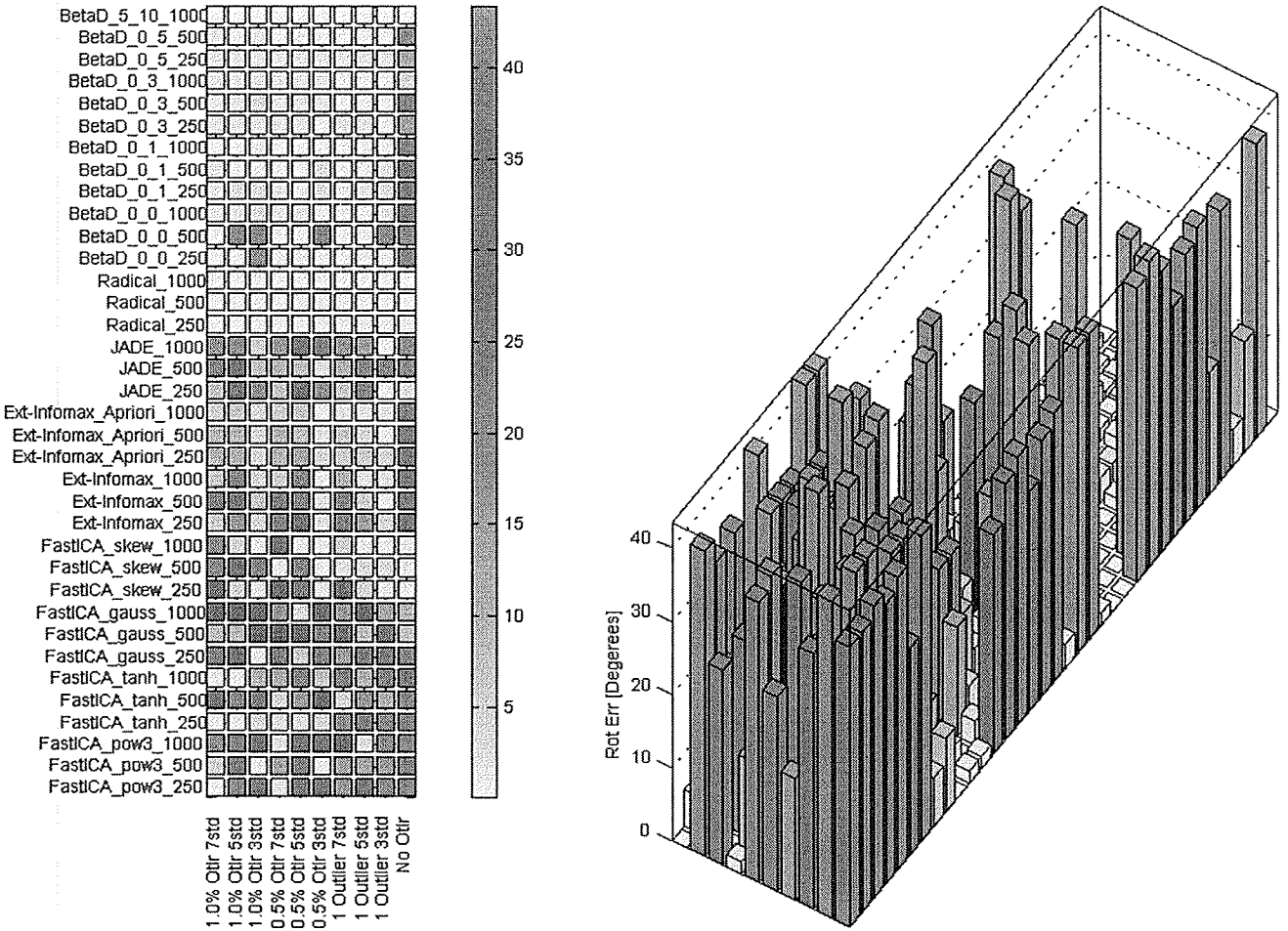


Fig. A.32 Rotation error: Asymmetric mixture of 2 Gaussians (Multimodal).

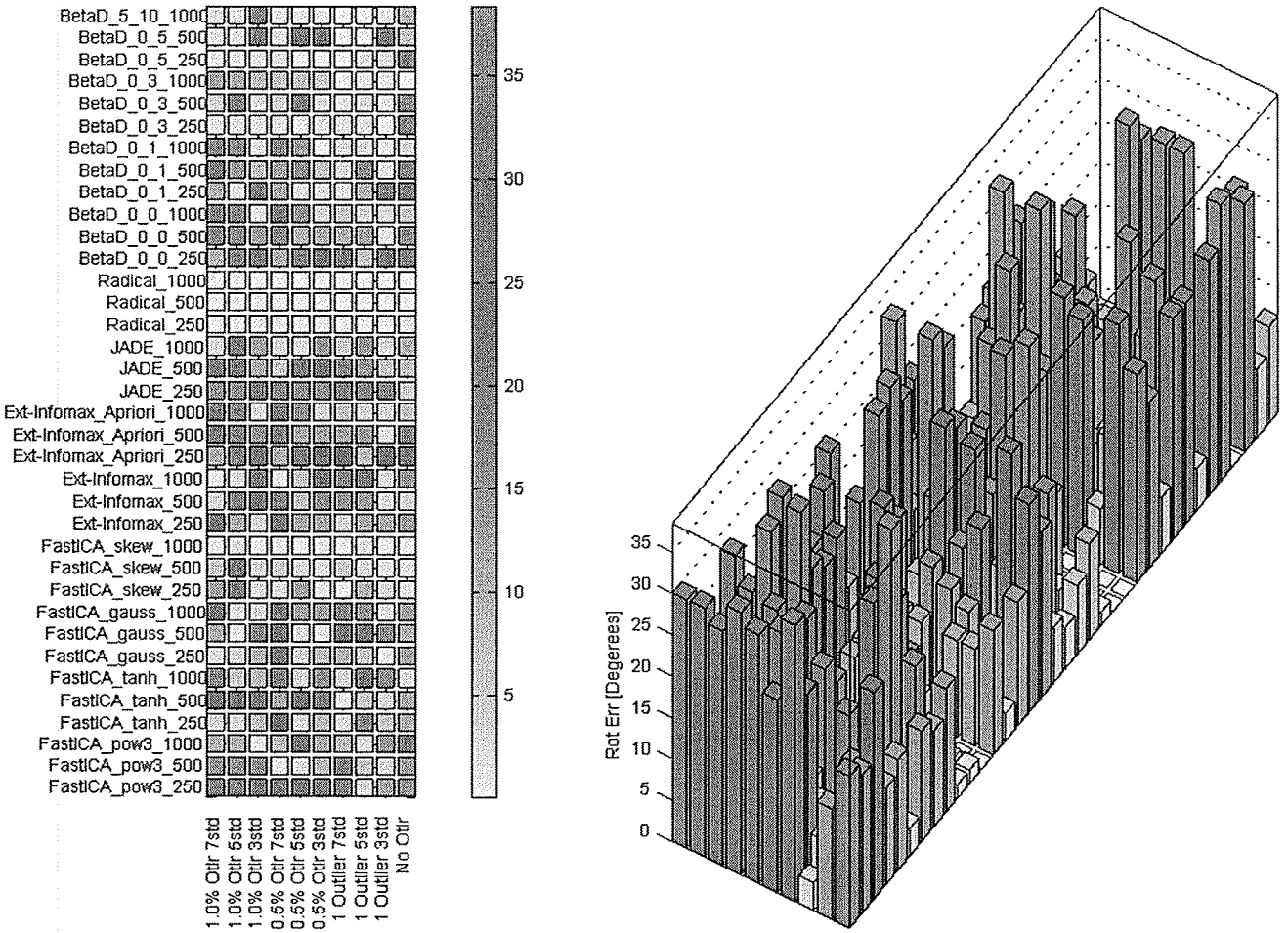


Fig. A.33 Rotation error: Asymmetric mixture of 2 Gaussians (Transitional).

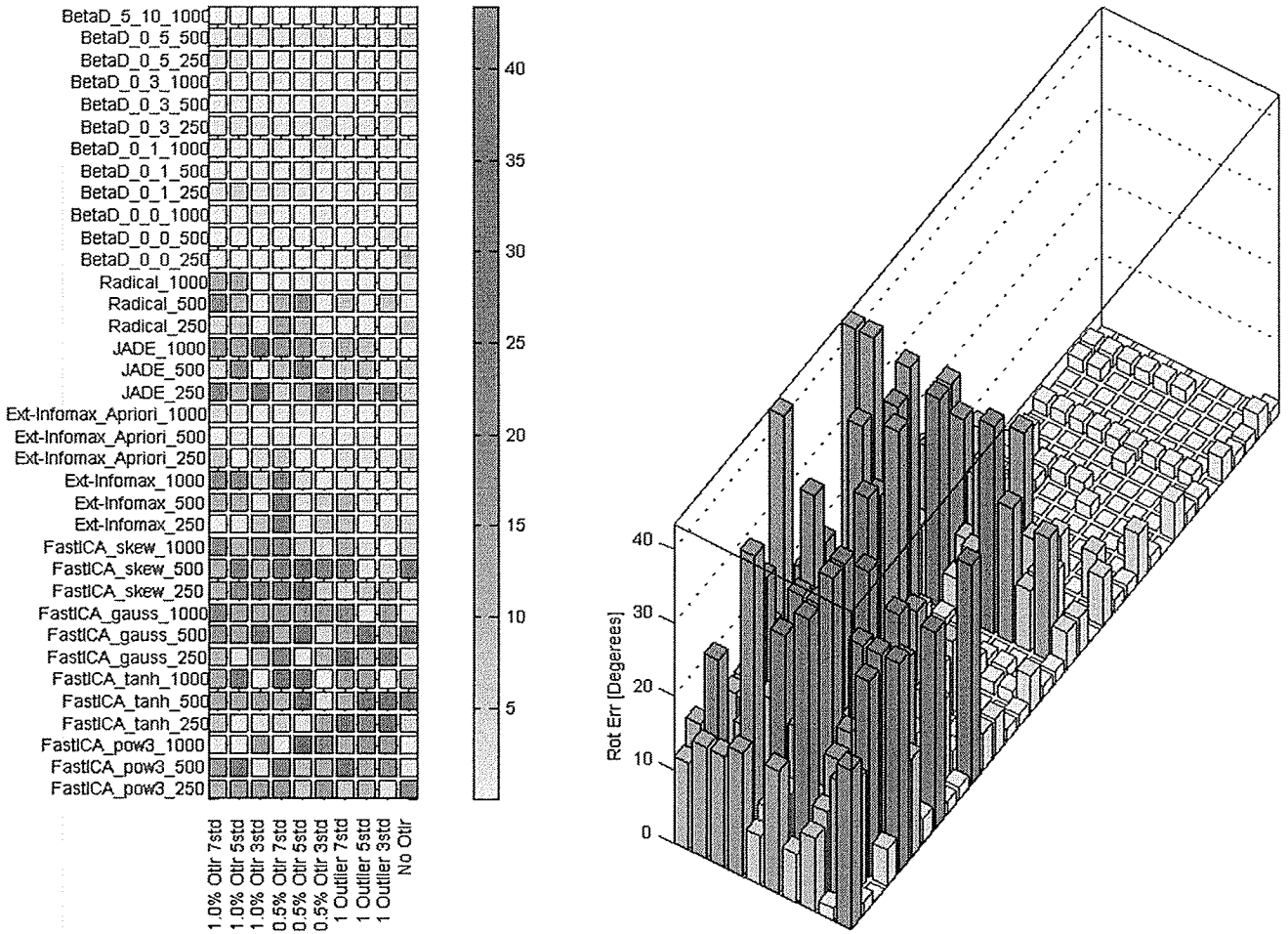


Fig. A.34 Rotation error: Asymmetric mixture of 2 Gaussians (Unimodal).

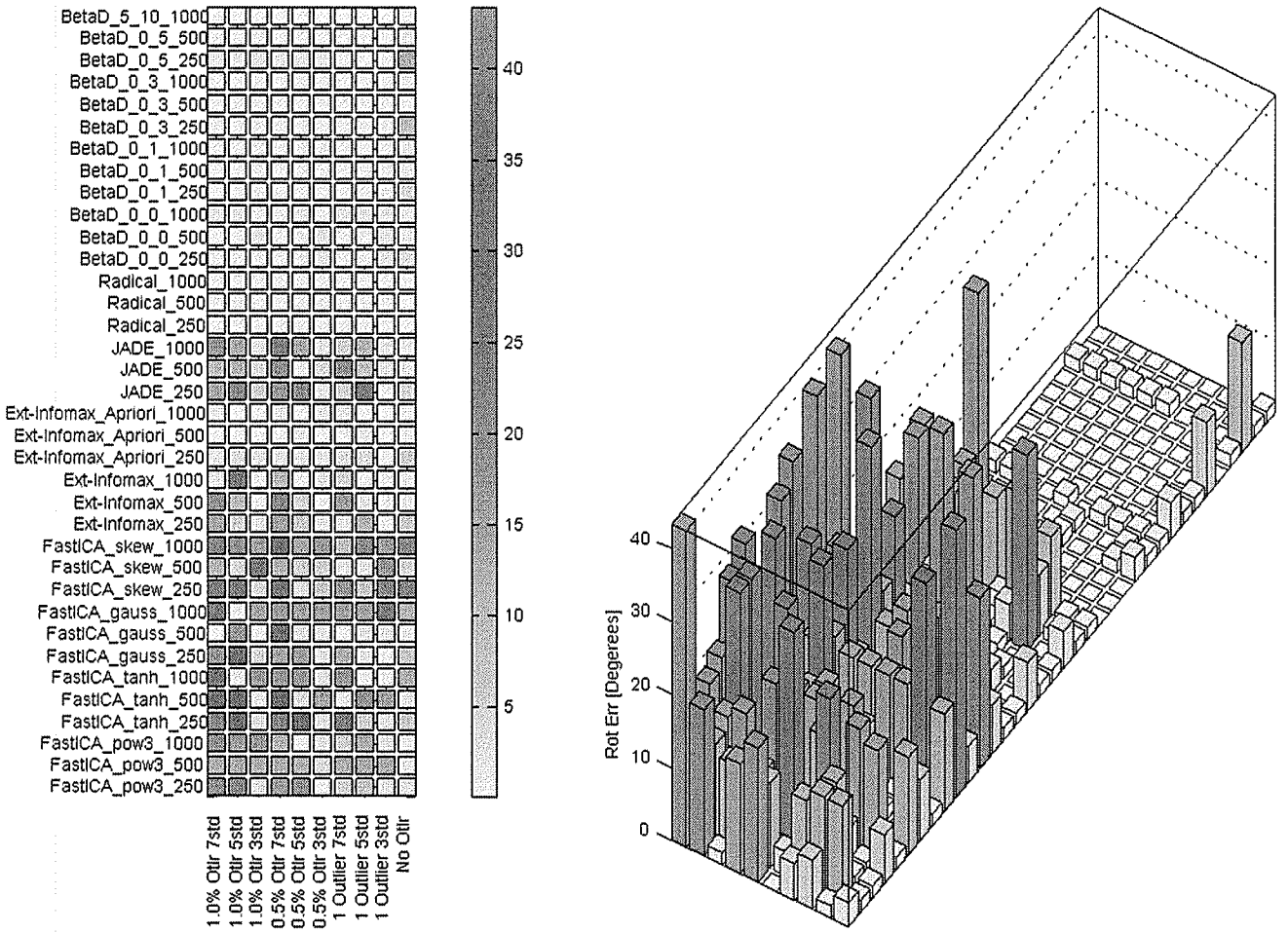


Fig. A.35 Rotation error: Symmetric mixture of 4 Gaussians (Multimodal).

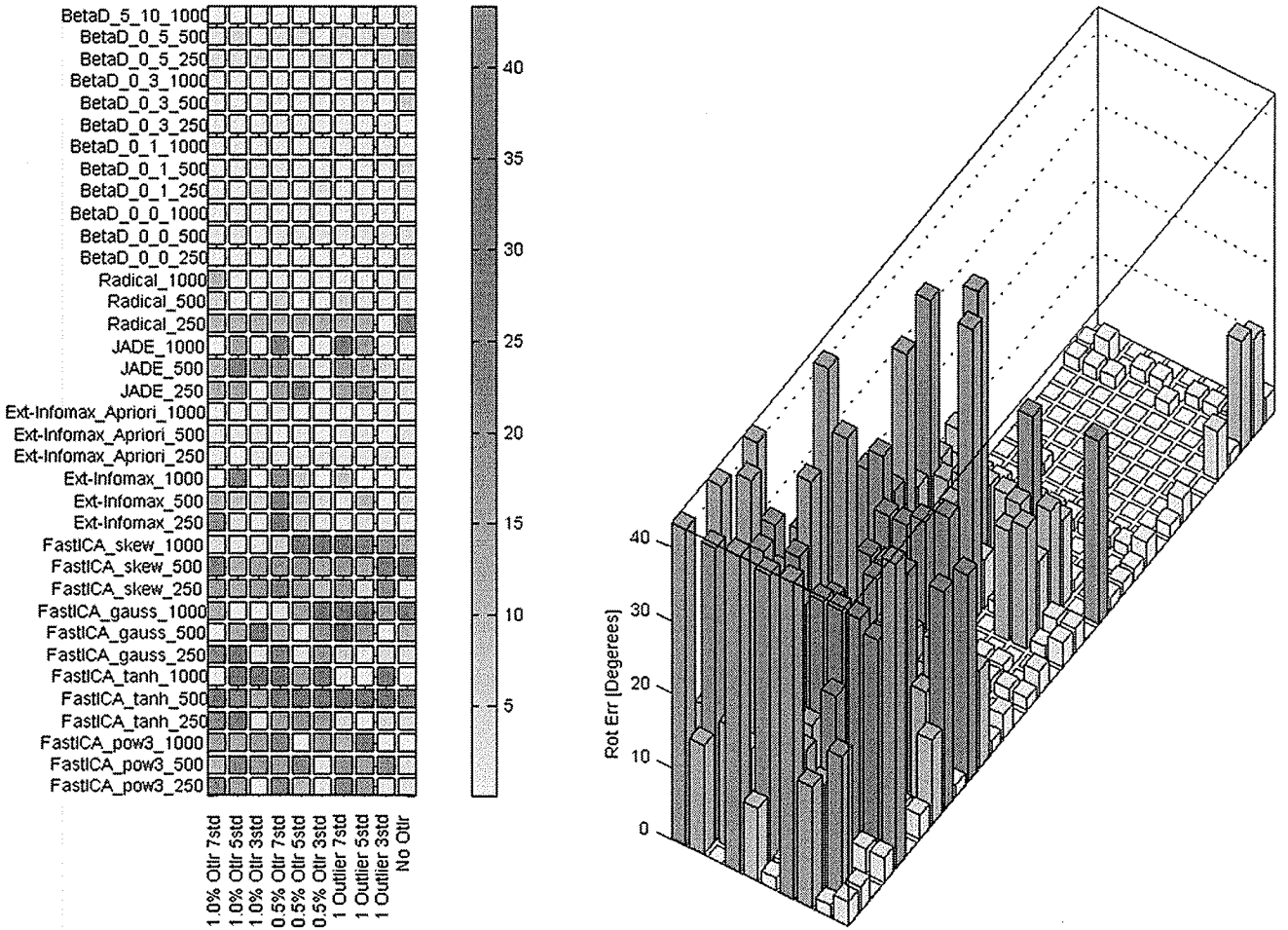


Fig. A.36 Rotation error: Symmetric mixture of 4 Gaussians (Transitional).

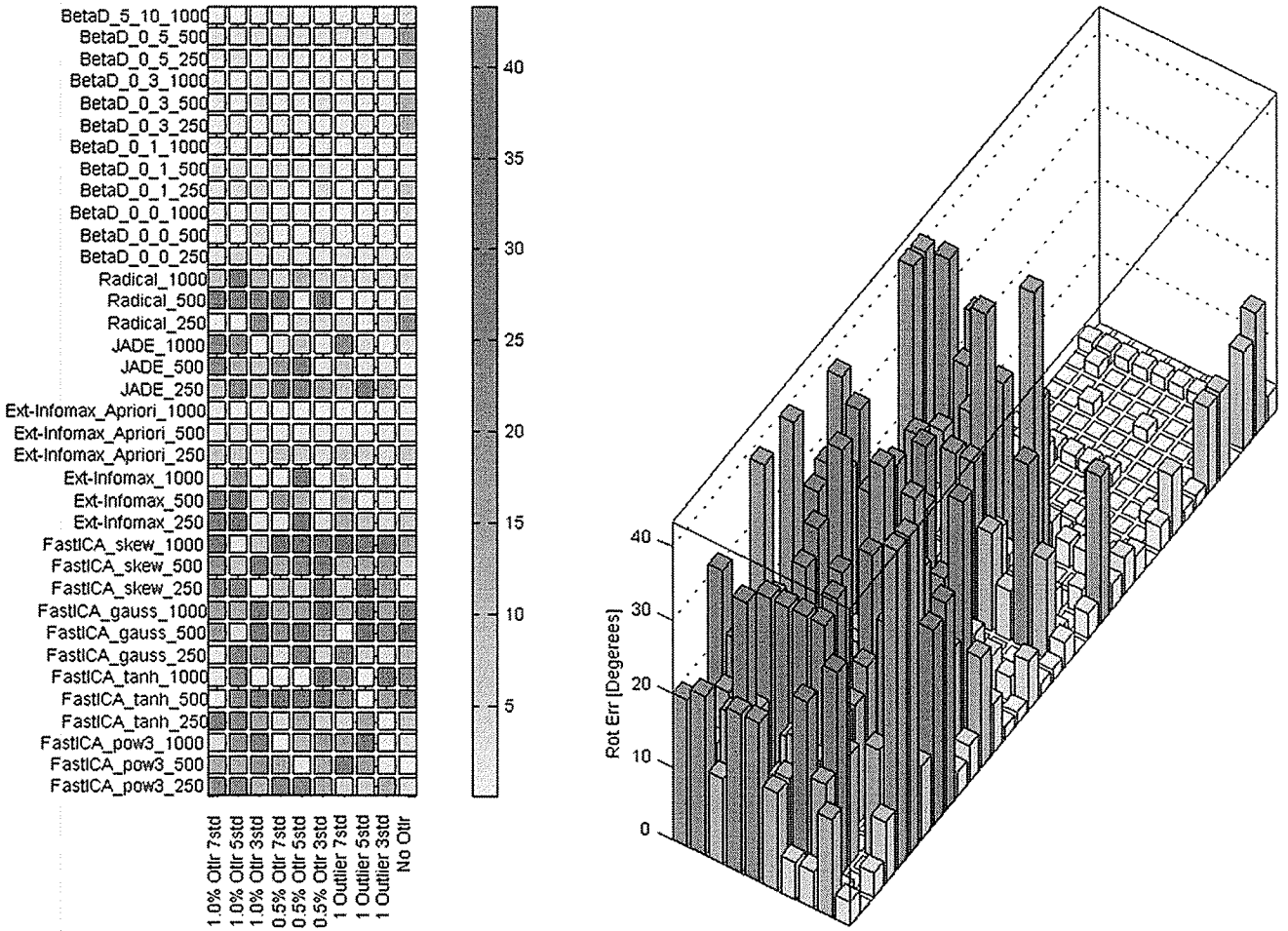


Fig. A.37 Rotation error: Symmetric mixture of 4 Gaussians (Unimodal).

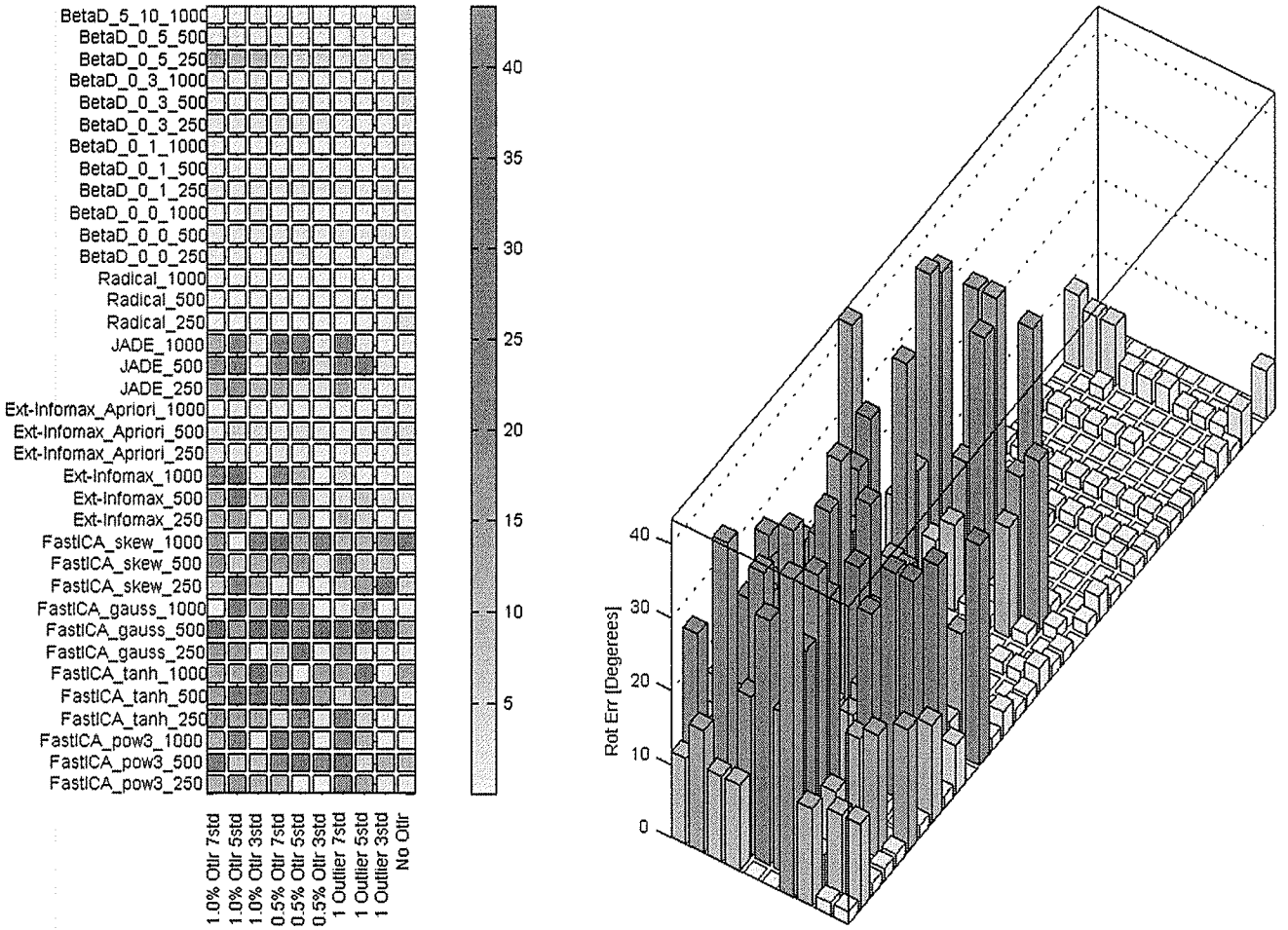


Fig. A.38 Rotation error: Asymmetric mixture of 4 Gaussians (Multimodal).

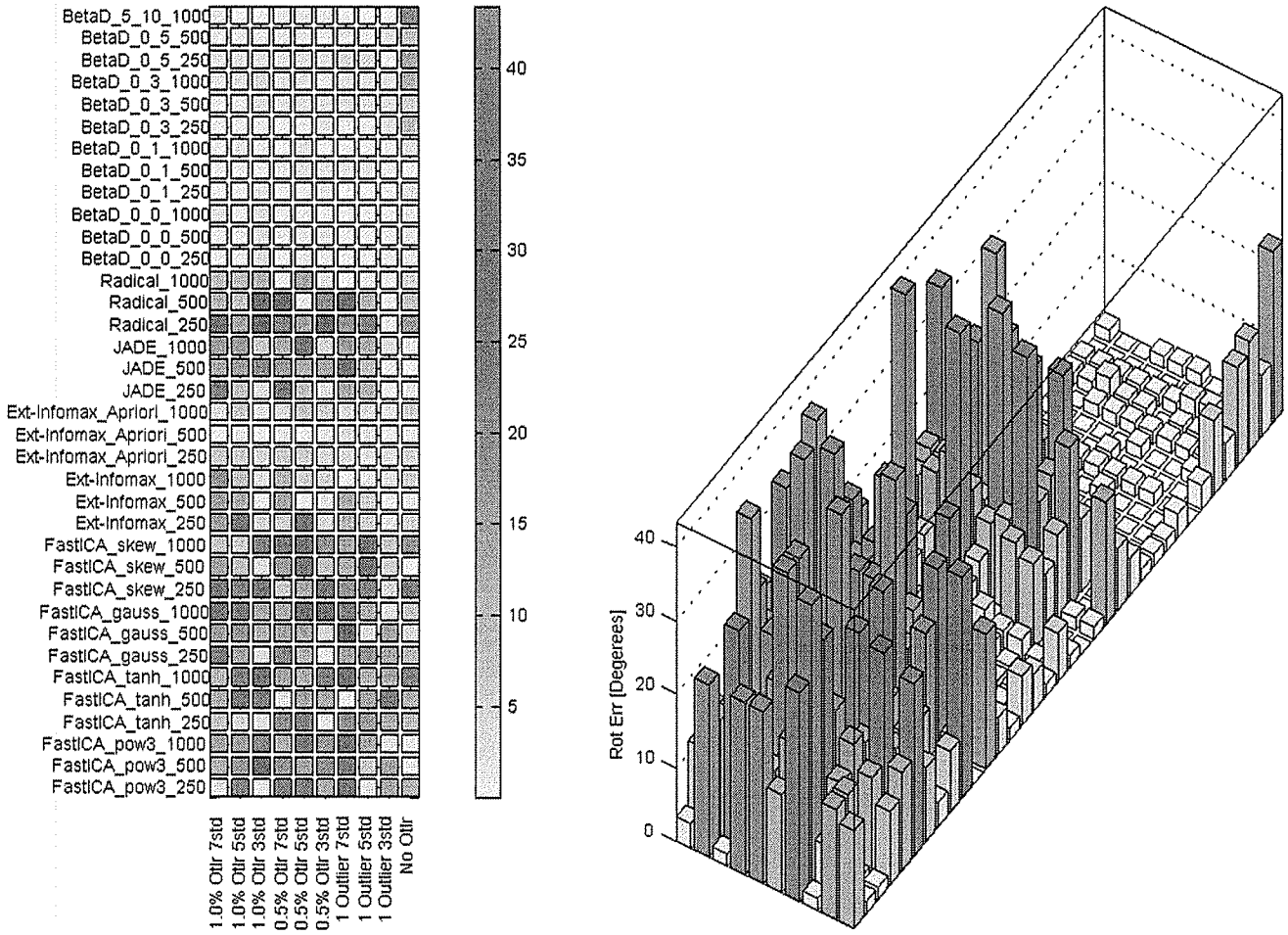


Fig. A.39 Rotation Error: Asymmetric mixture of 4 Gaussians (Transitional).

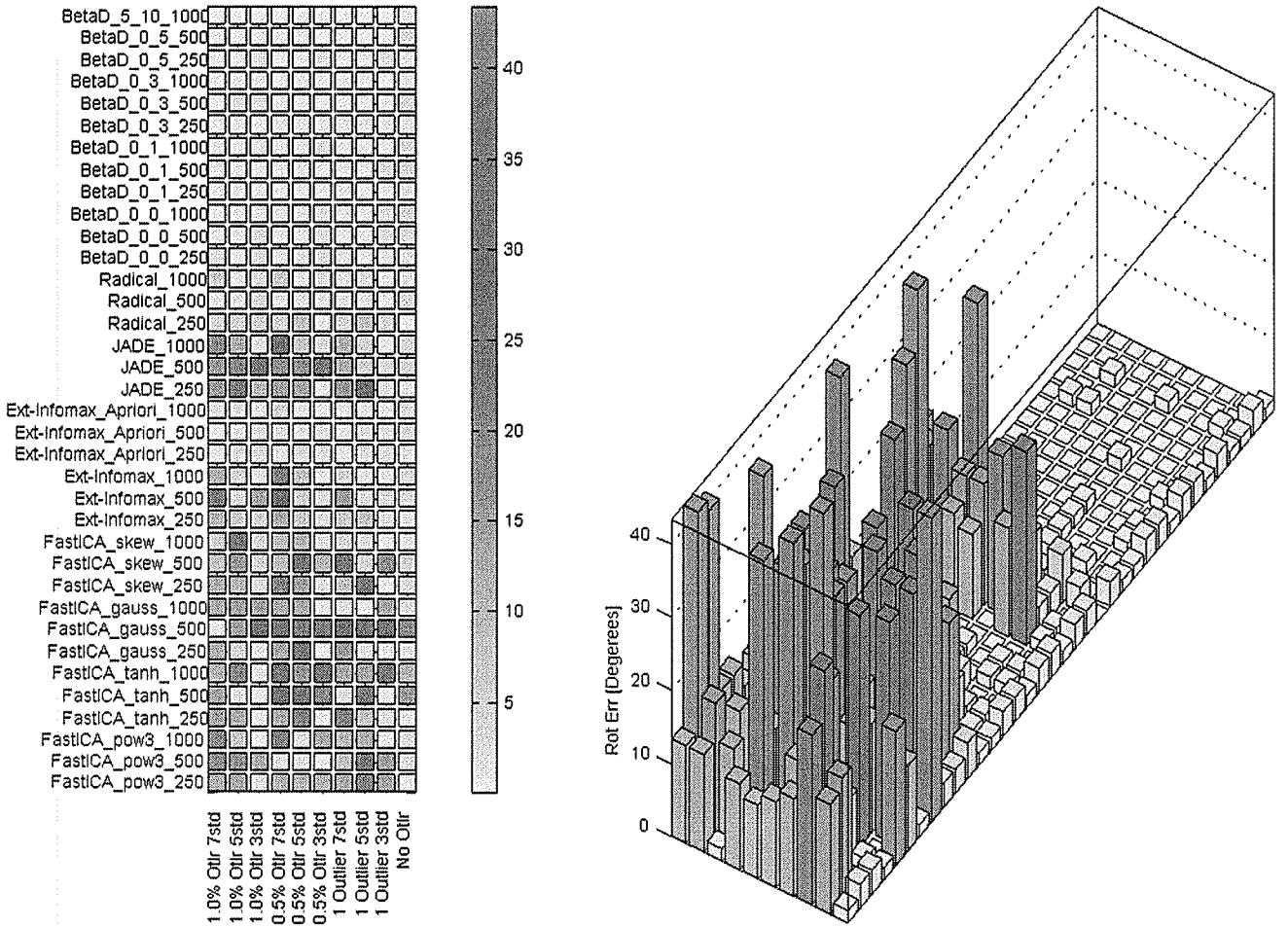


Fig. A.40 Rotation error: Asymmetric mixture of 4 Gaussians (Unimodal).

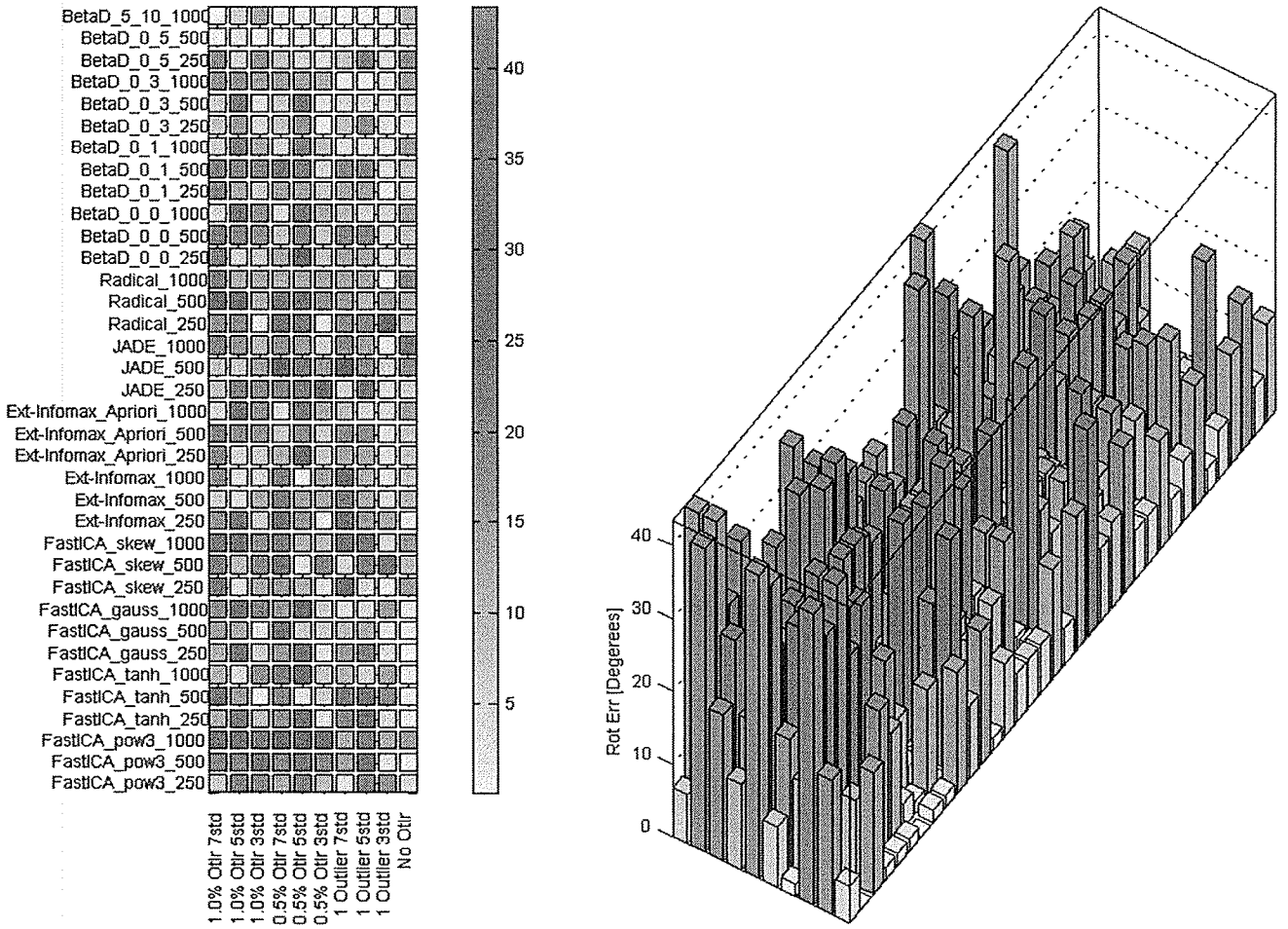


Fig. A.41 Rotation error: Gaussian.

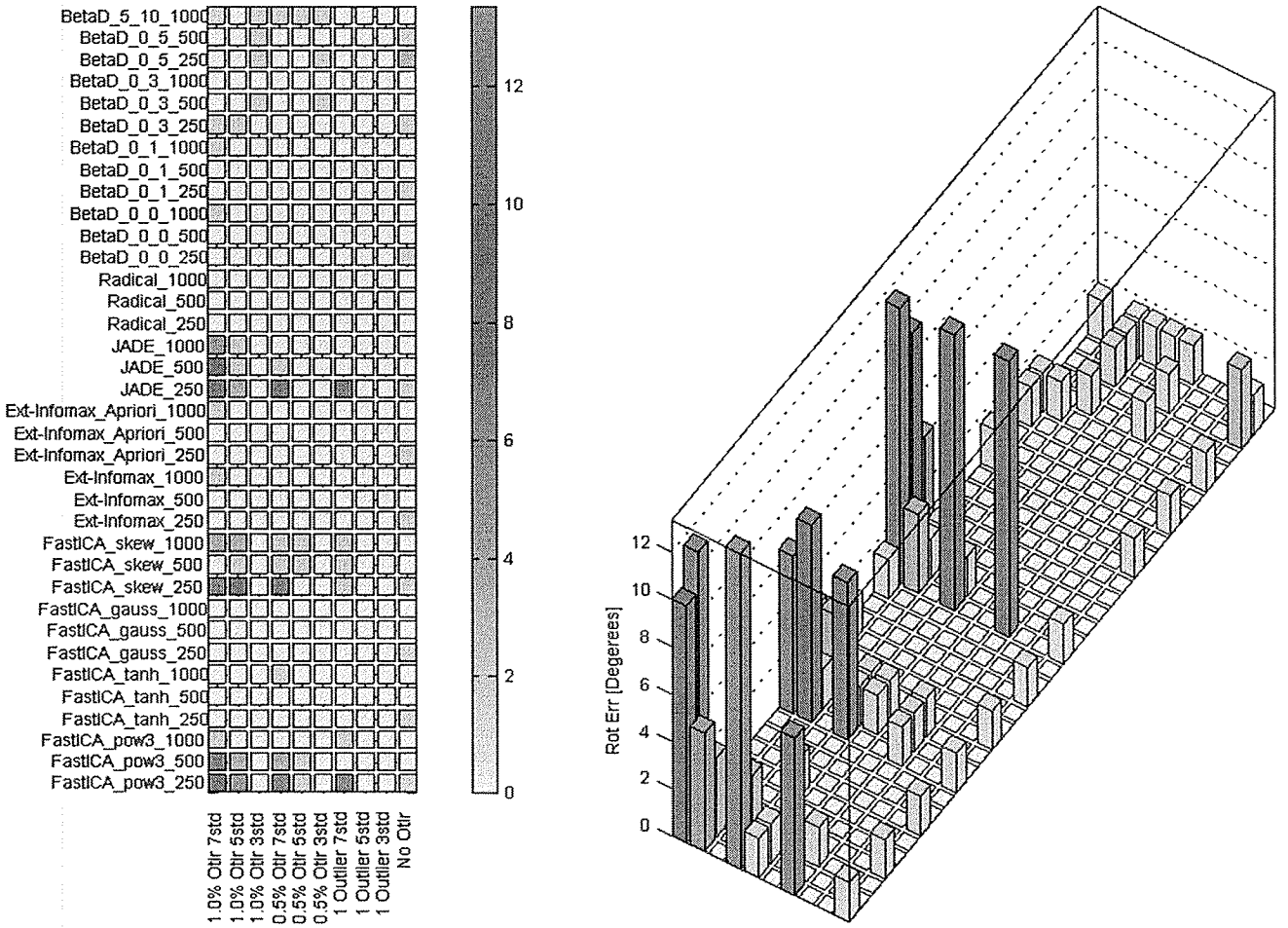


Fig. A.42 Rotation error: LogNormal.

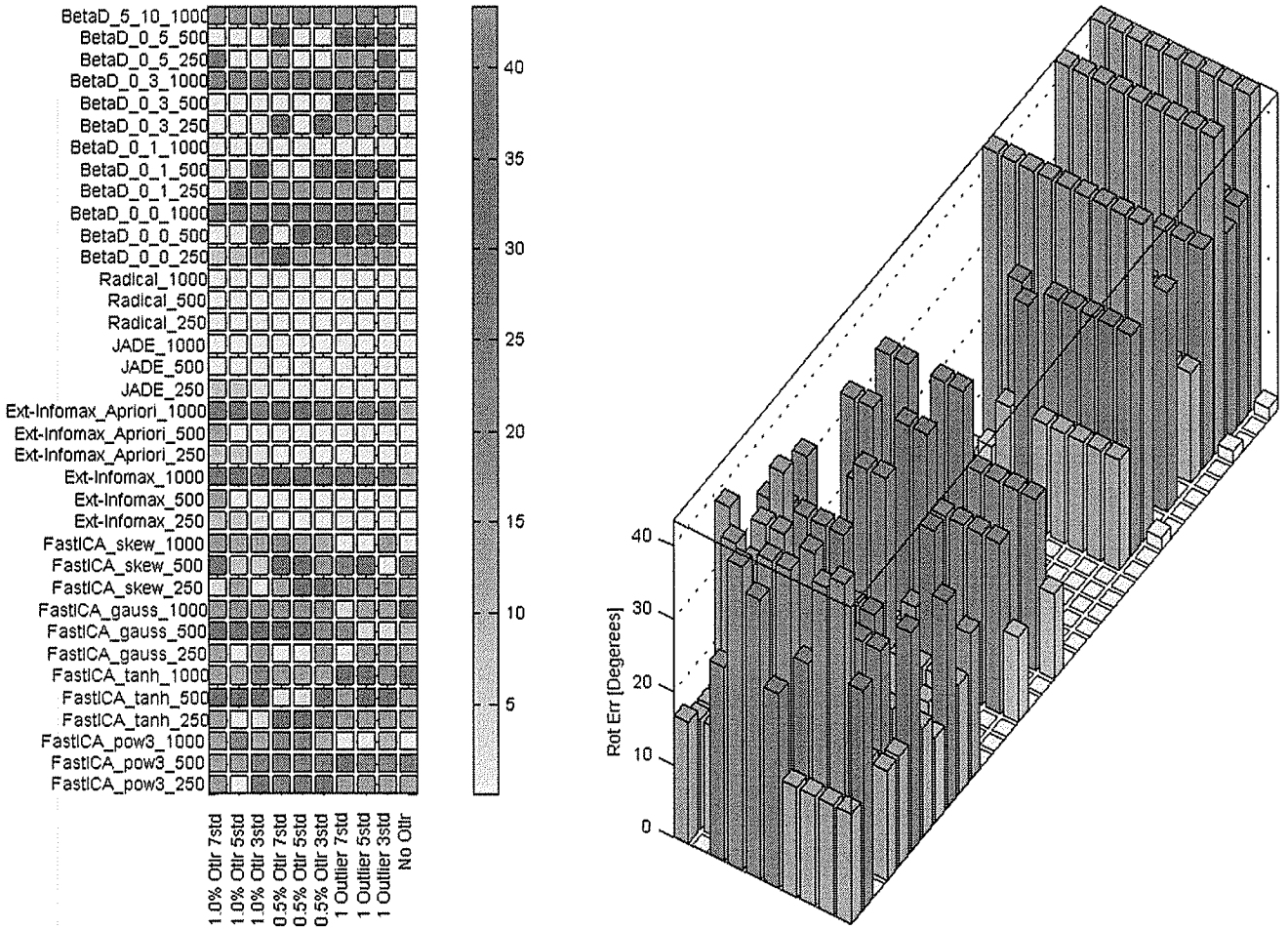


Fig. A.43 Rotation error: Pareto.

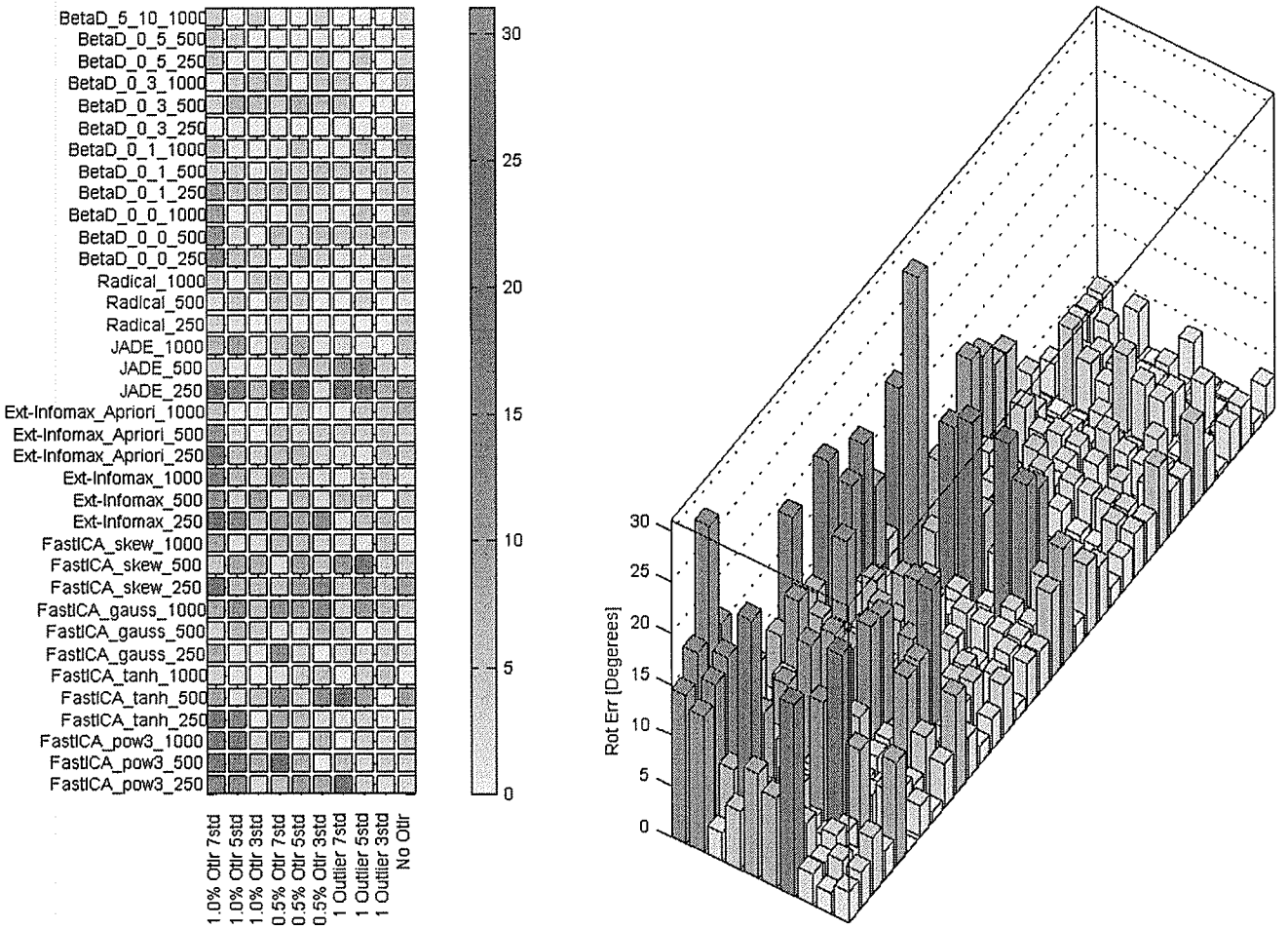


Fig. A.44 Rotation error: Random Mixture.

A.4 Contrast function difference for Mixtures of Densities

The following tables contain the contrast function difference of the ICA contrast functions for mixtures of the given density. The experiment setup is explained in Sec. 4.3.

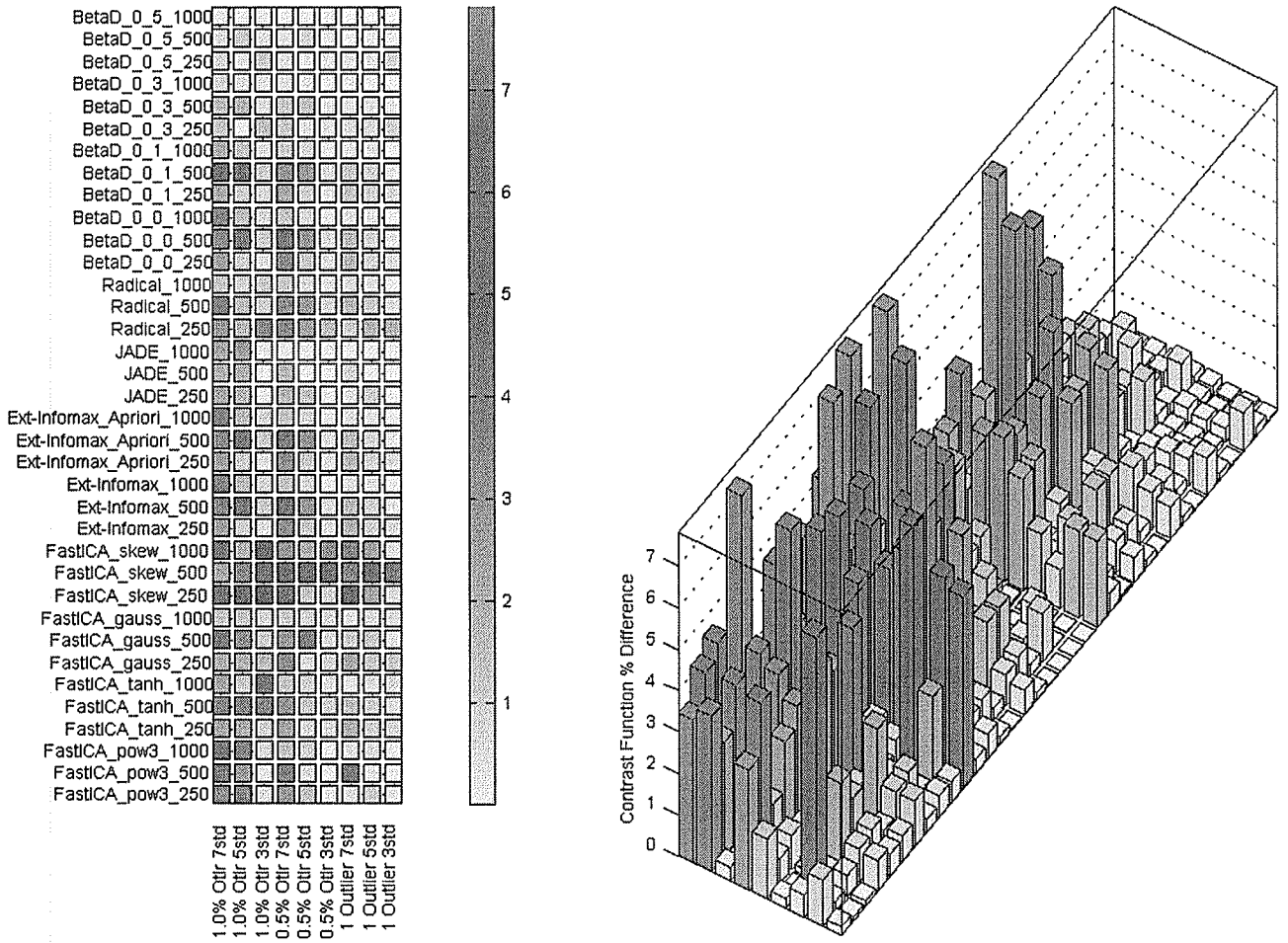


Fig. A.45 Contrast function difference: Student-t 3 degrees of freedom.

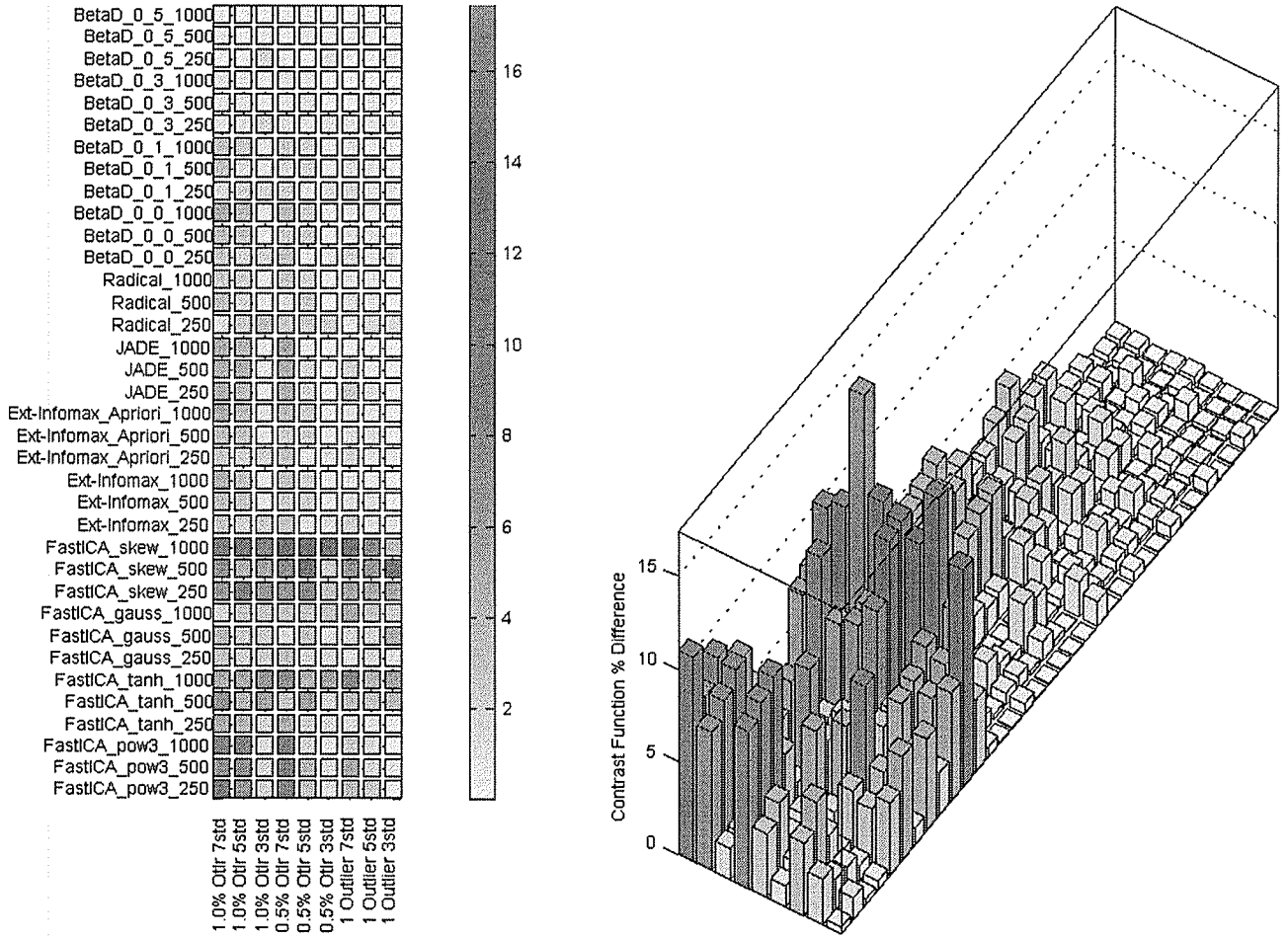


Fig. A.46 Contrast function difference: Double Exponential.

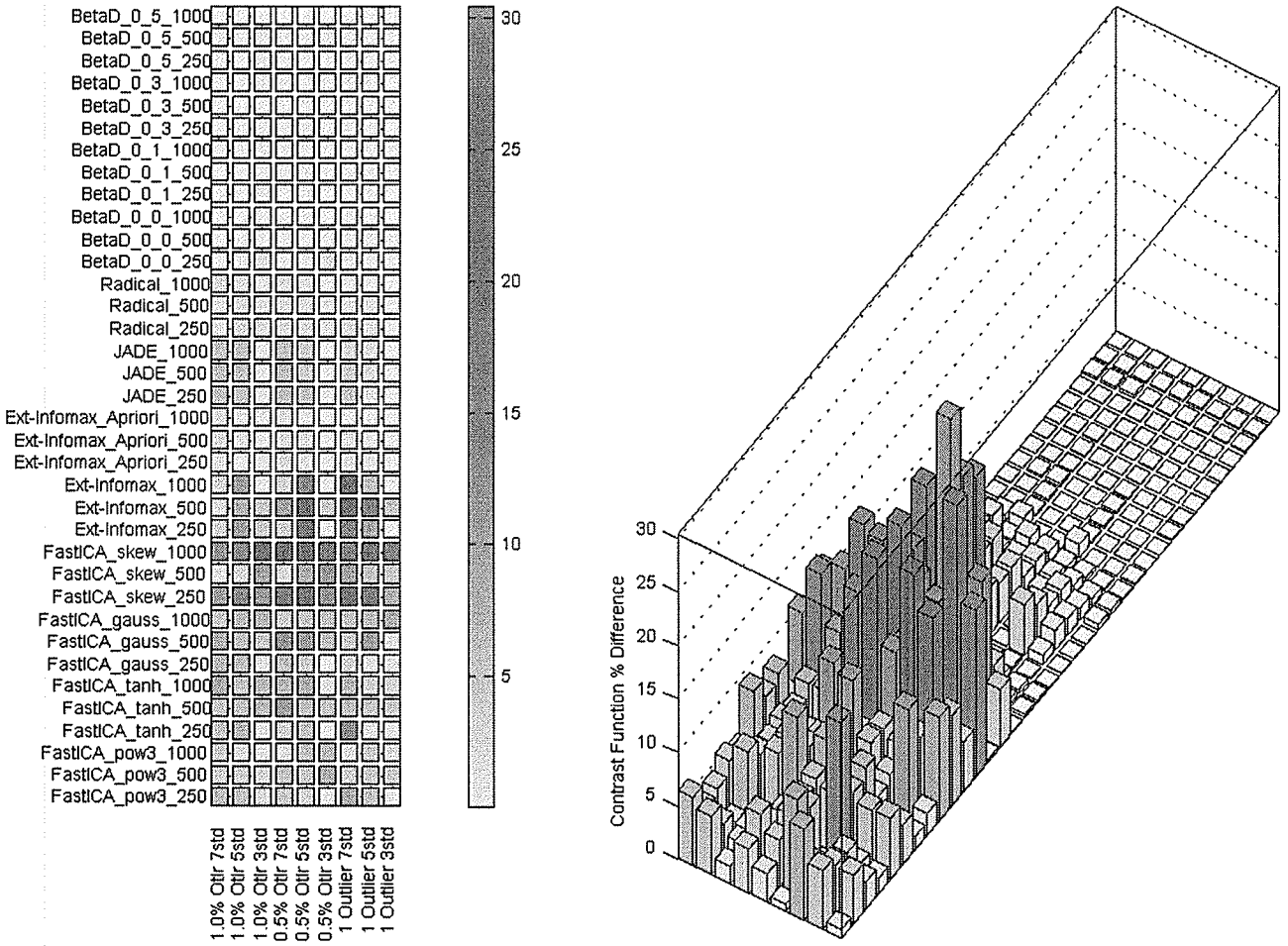


Fig. A.47 Contrast function difference: Uniform.

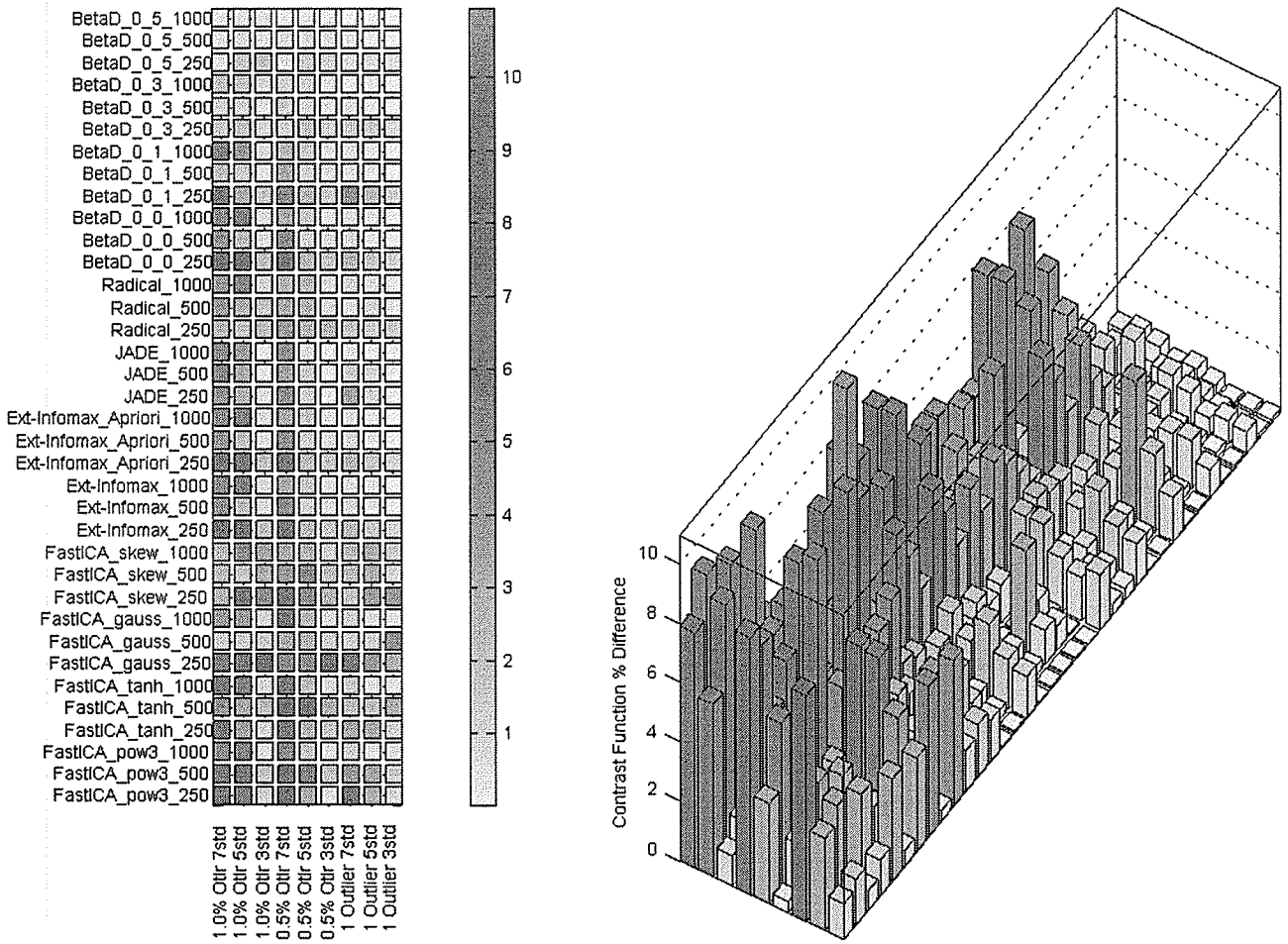


Fig. A.48 Contrast function difference: Student-t 5 degrees of freedom.

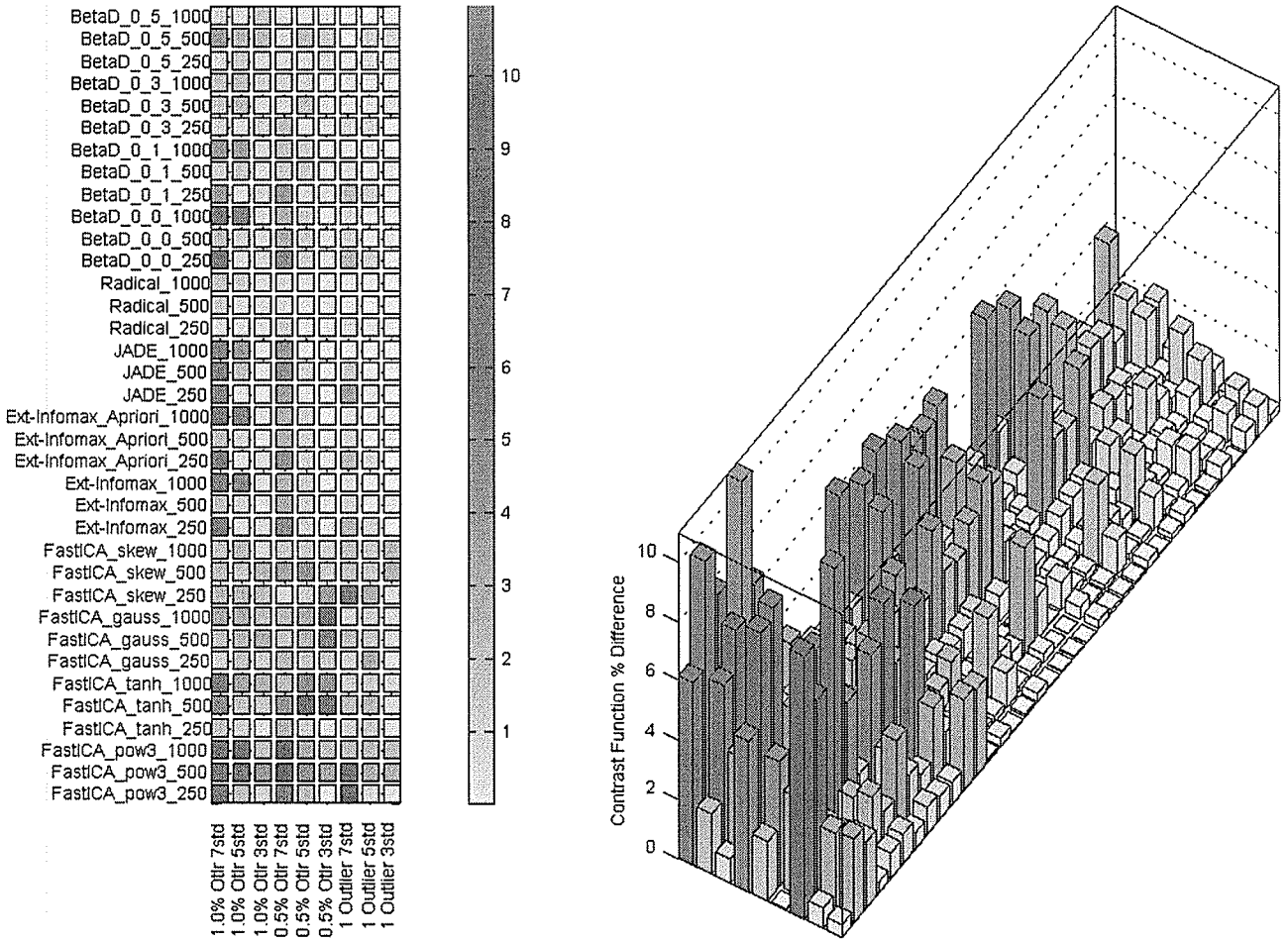


Fig. A.49 Contrast function difference: Exponential.

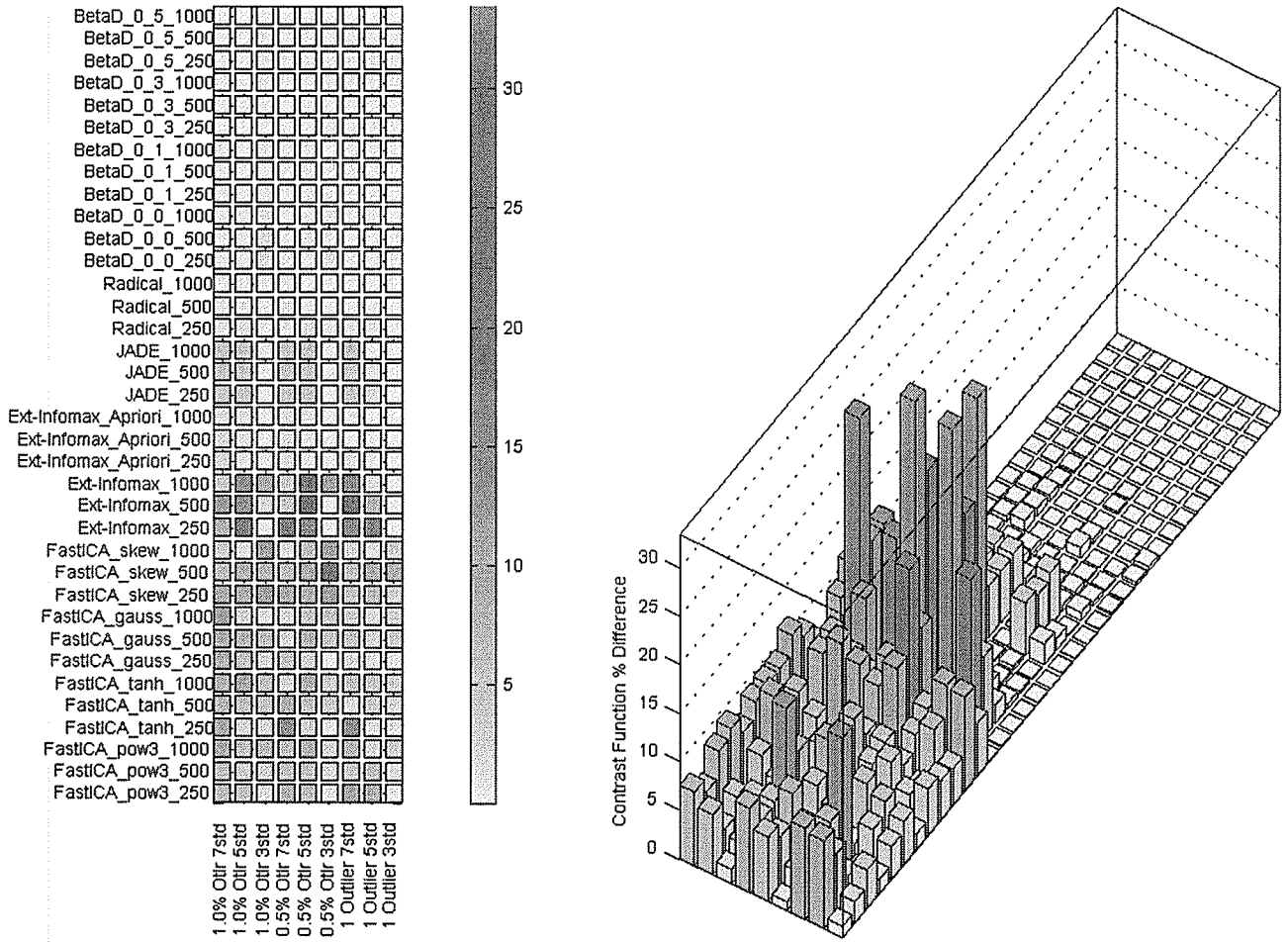


Fig. A.50 Contrast function difference: Mixture of 2 double exponentials.

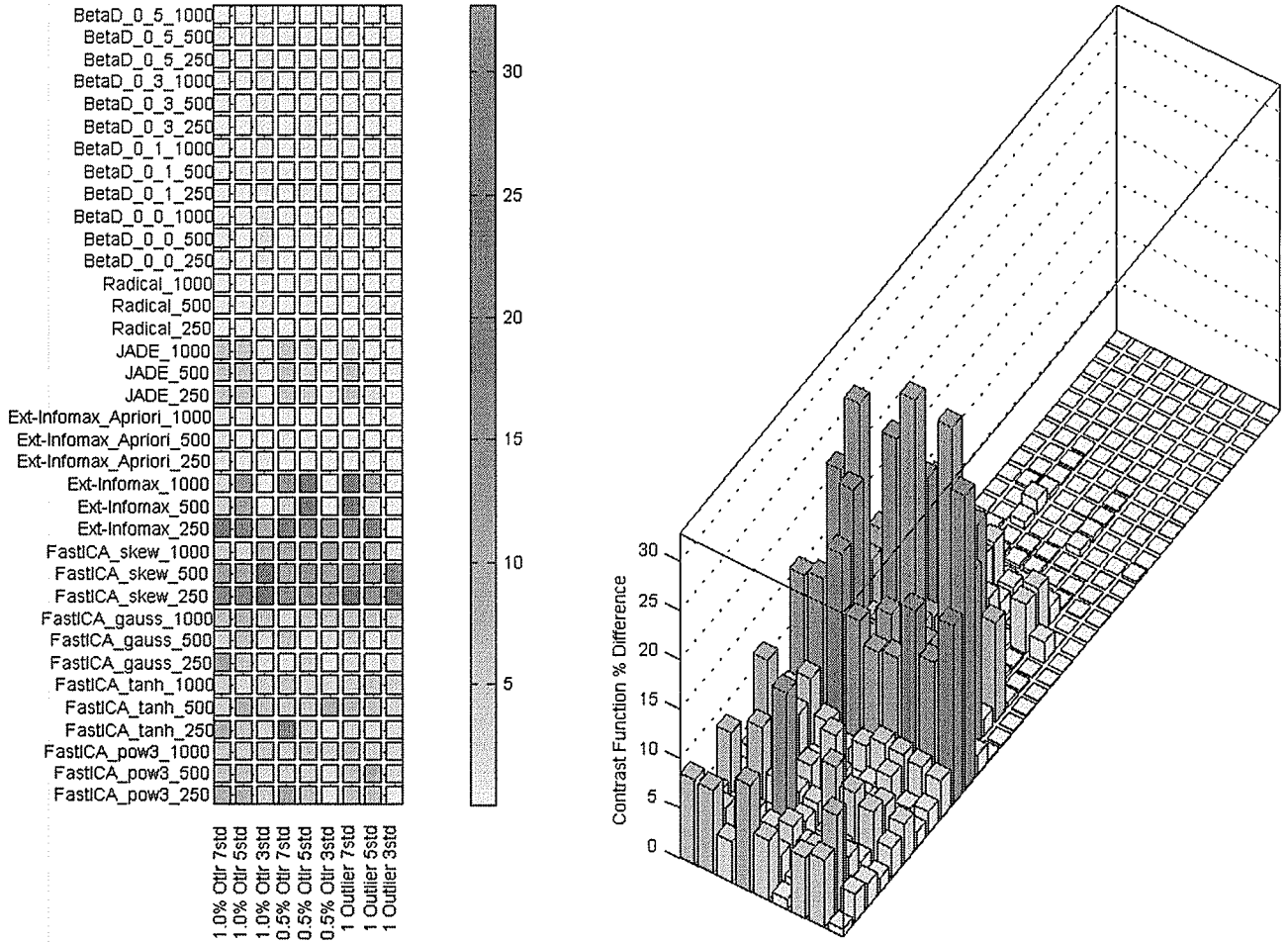


Fig. A.51 Contrast function difference: Symmetric mixture of 2 Gaussians (Multimodal).

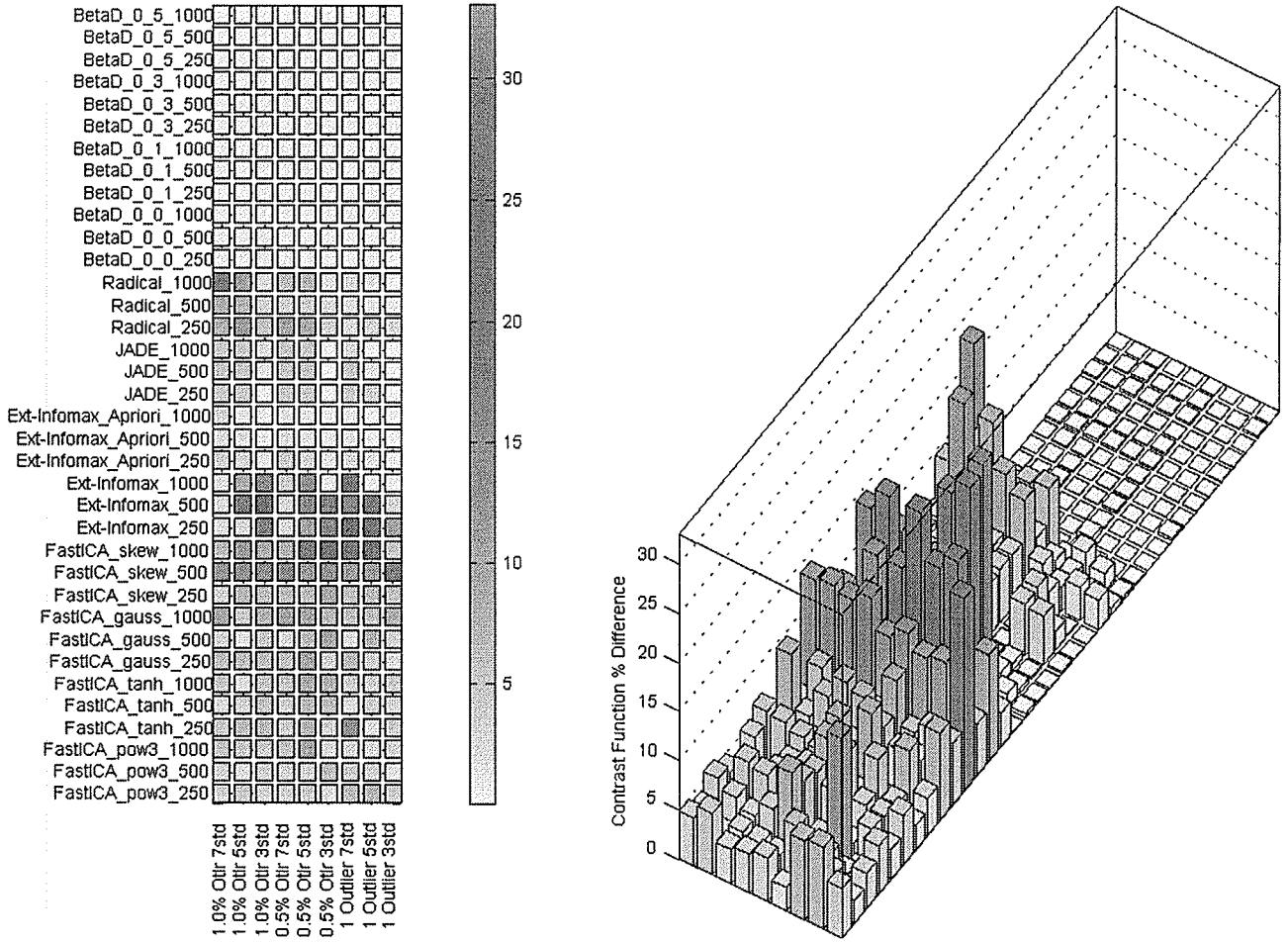


Fig. A.52 Contrast function difference: Symmetric mixture of 2 Gaussians (Transitional).

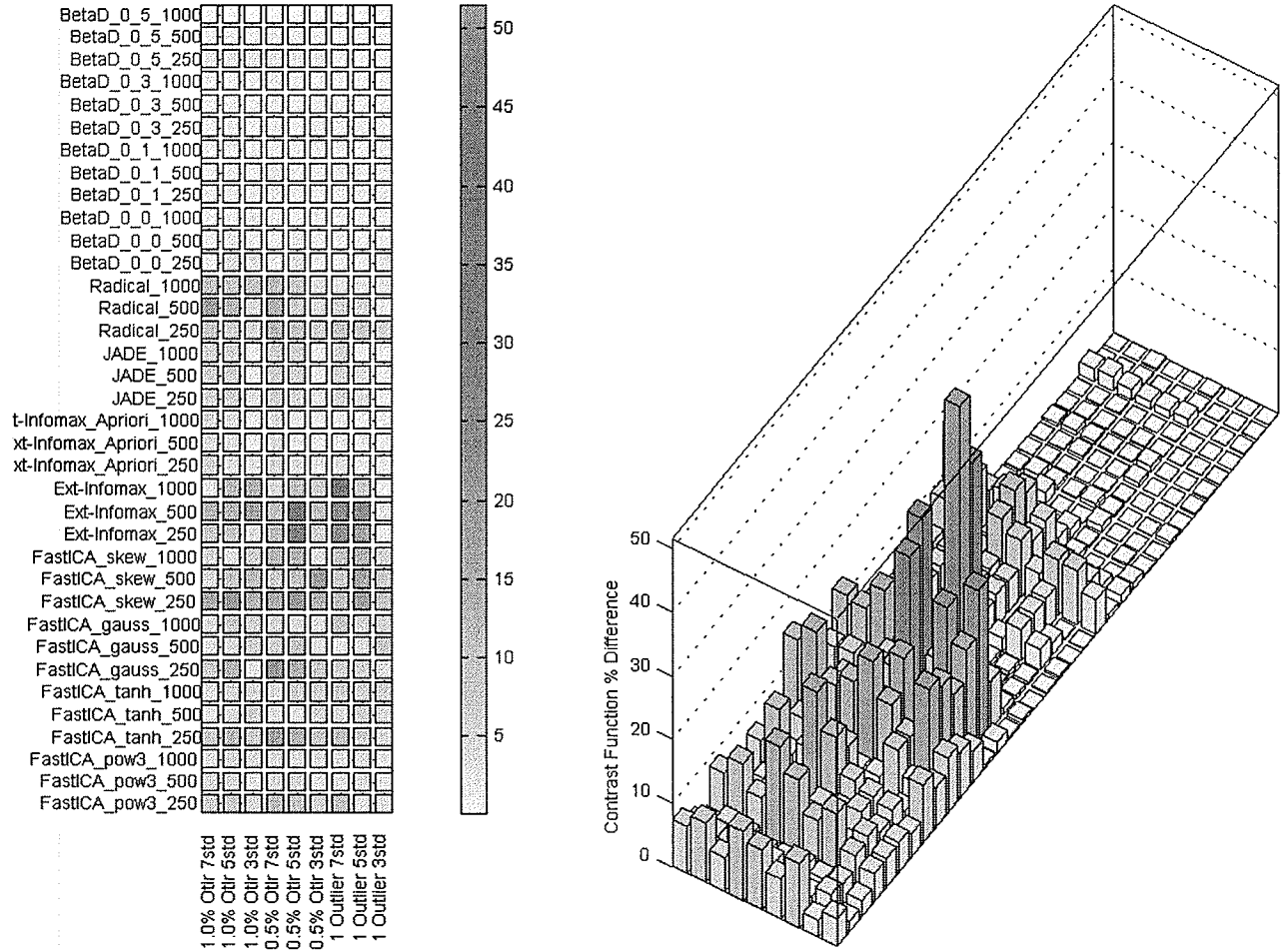


Fig. A.53 Contrast function difference: Symmetric mixture of 2 Gaussians (Unimodal).

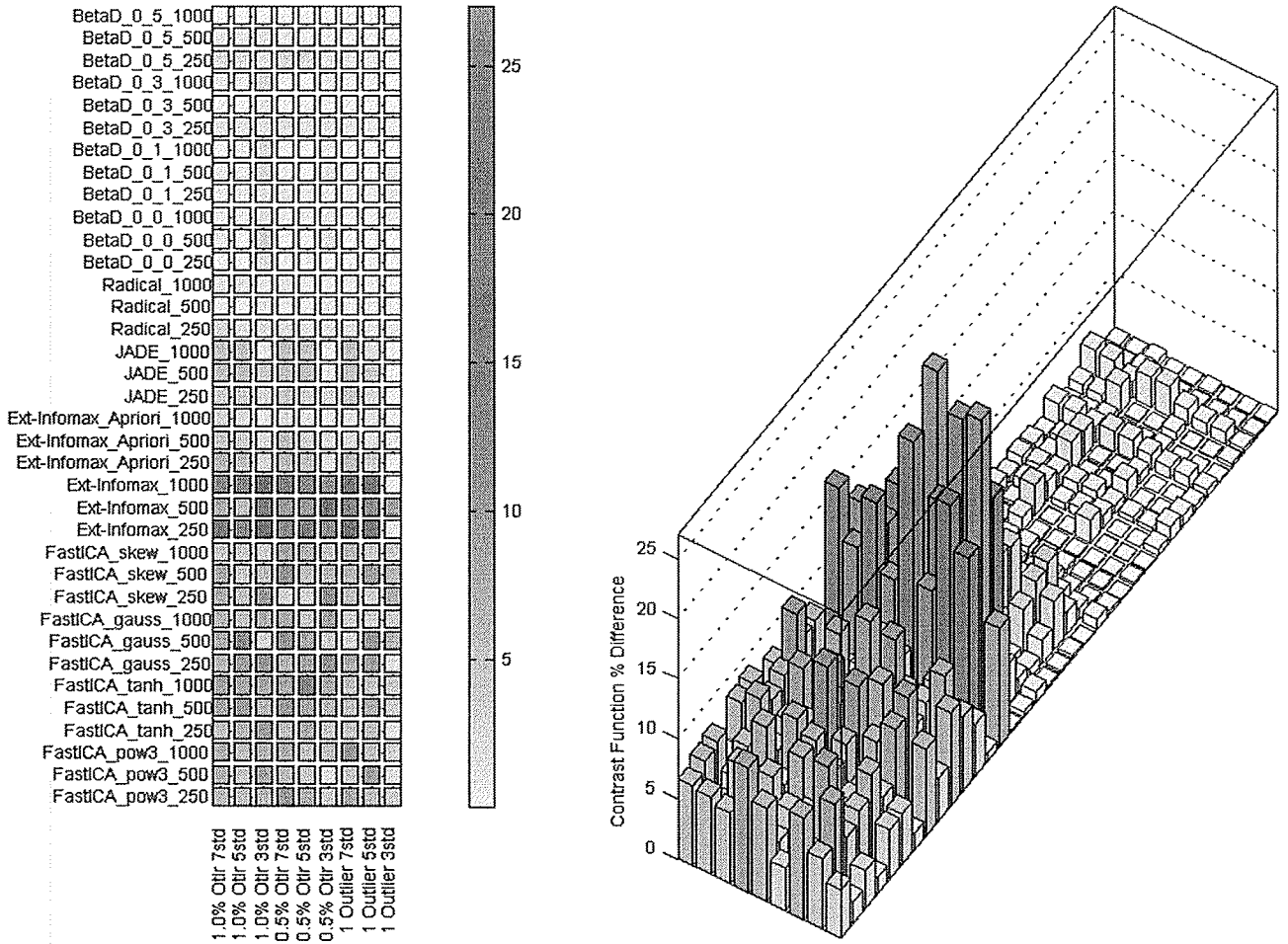


Fig. A.54 Contrast function difference: Asymmetric mixture of 2 Gaussians (Multimodal).

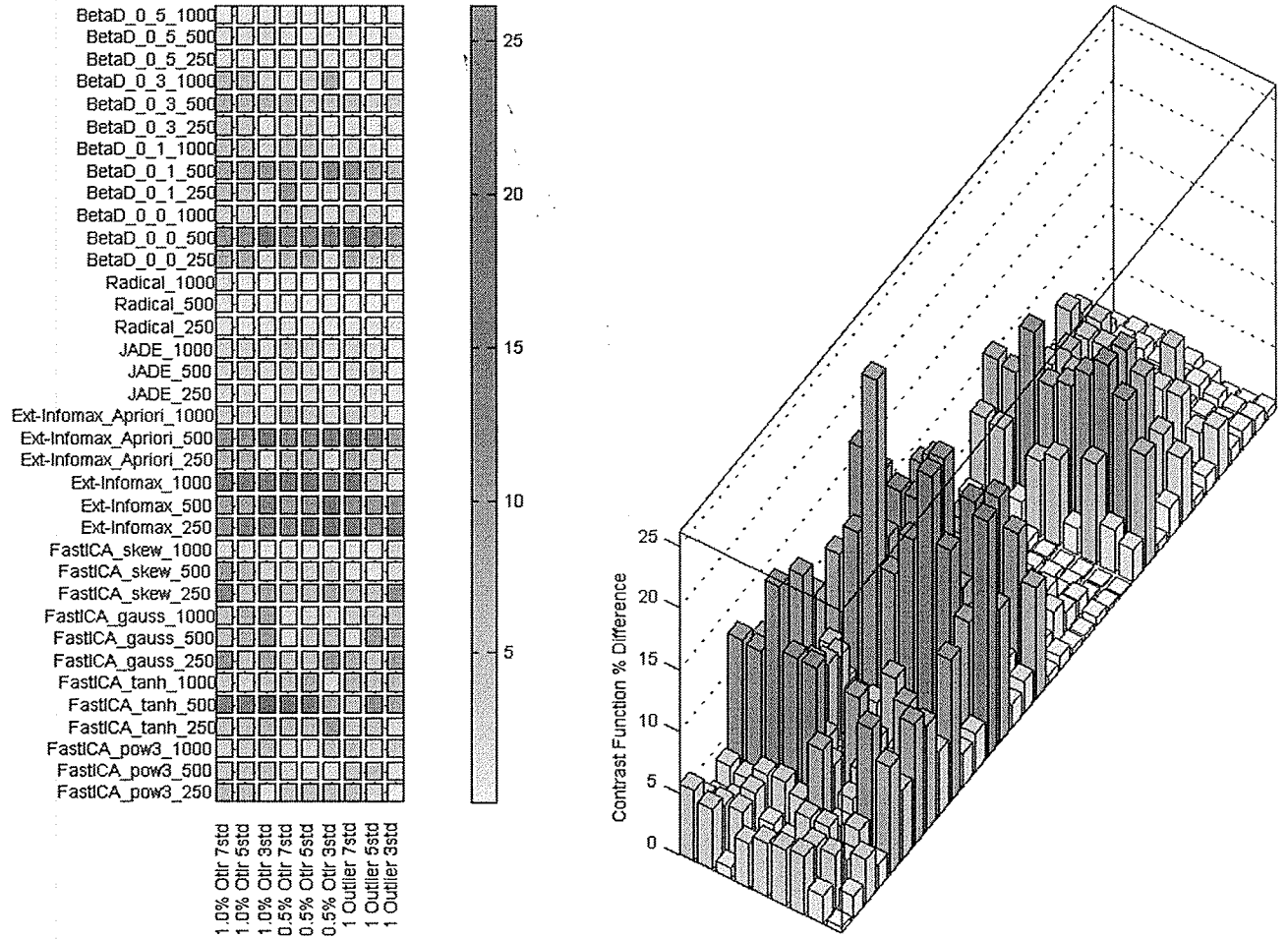


Fig. A.55 Contrast function difference: Asymmetric mixture of 2 Gaussians (Transitional).

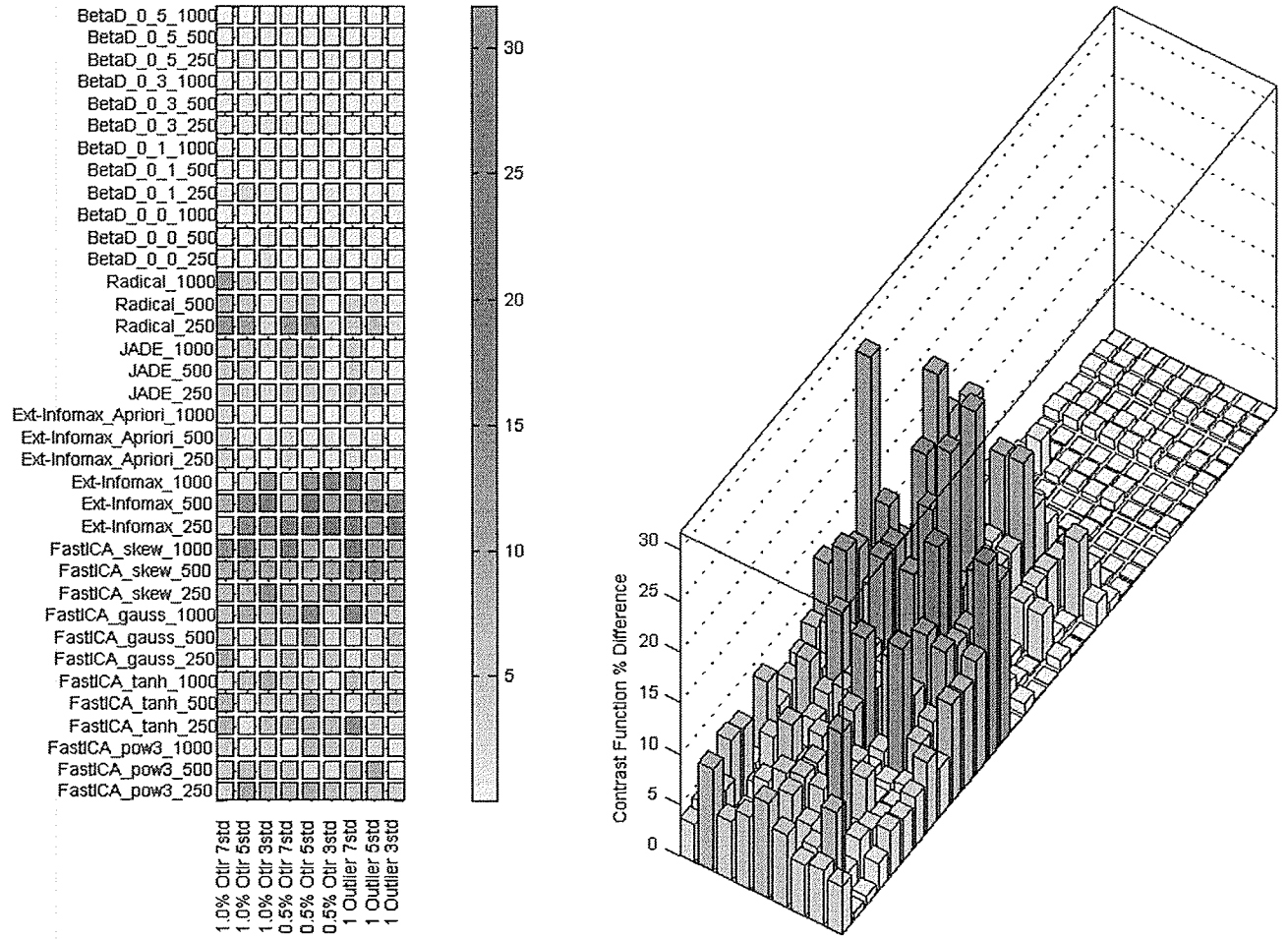


Fig. A.56 Contrast function difference: Asymmetric mixture of 2 Gaussians (Unimodal).

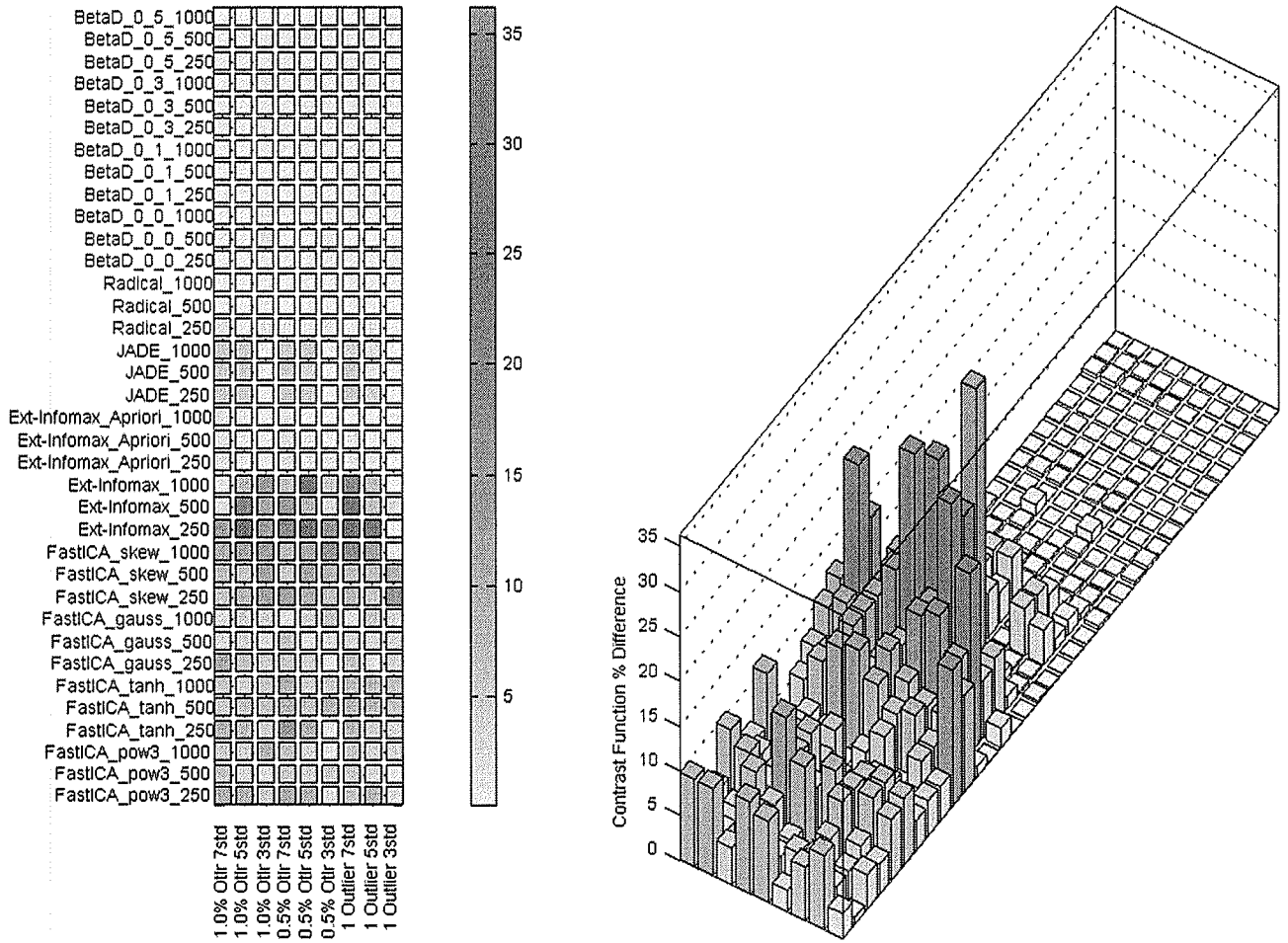


Fig. A.57 Contrast function difference: Symmetric mixture of 4 Gaussians (Multimodal).

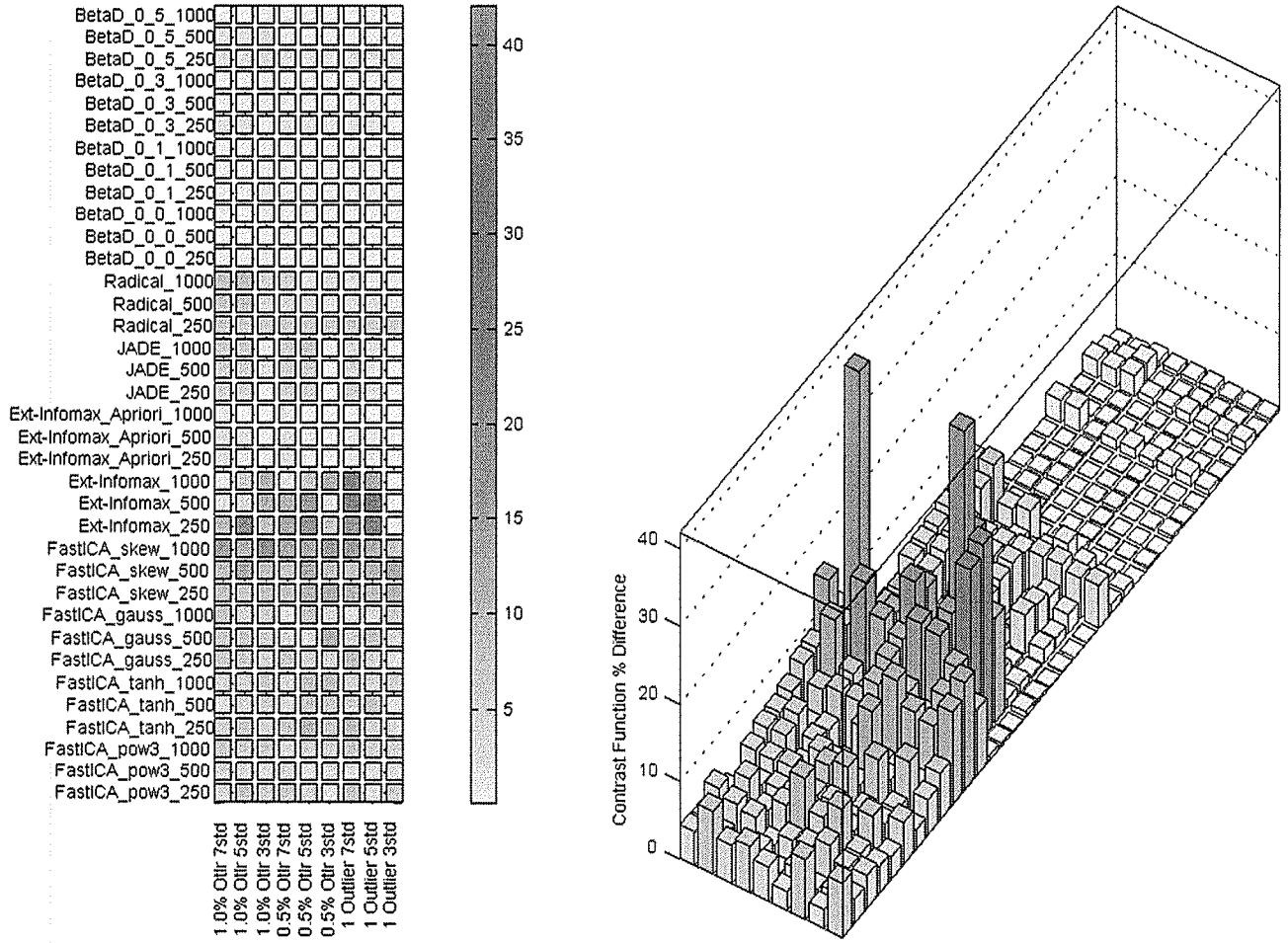


Fig. A.58 Contrast function difference: Symmetric mixture of 4 Gaussians (Transitional).

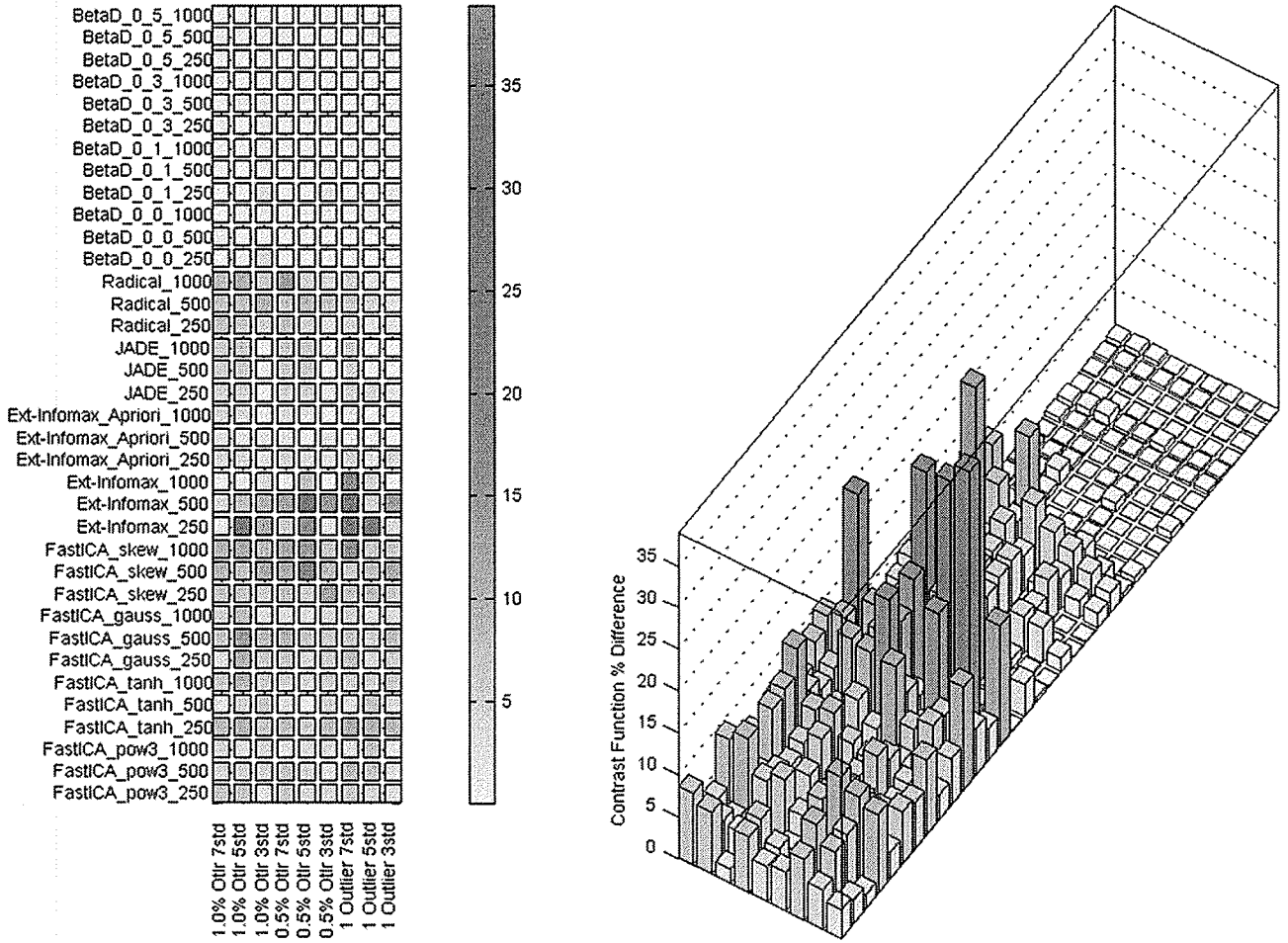


Fig. A.59 Contrast function difference: Symmetric mixture of 4 Gaussians (Unimodal).

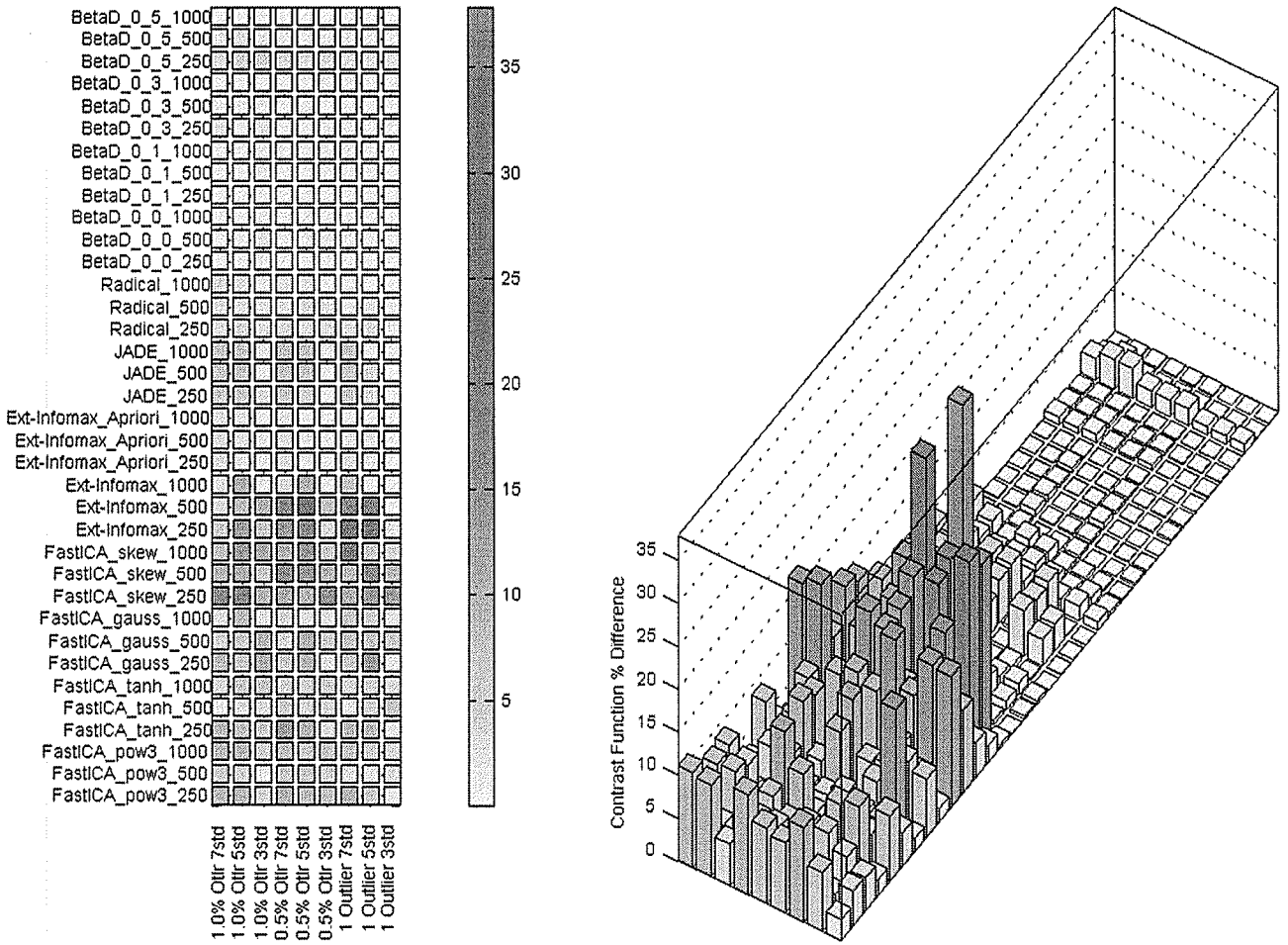


Fig. A.60 Contrast function difference: Asymmetric mixture of 4 Gaussians (Multimodal).

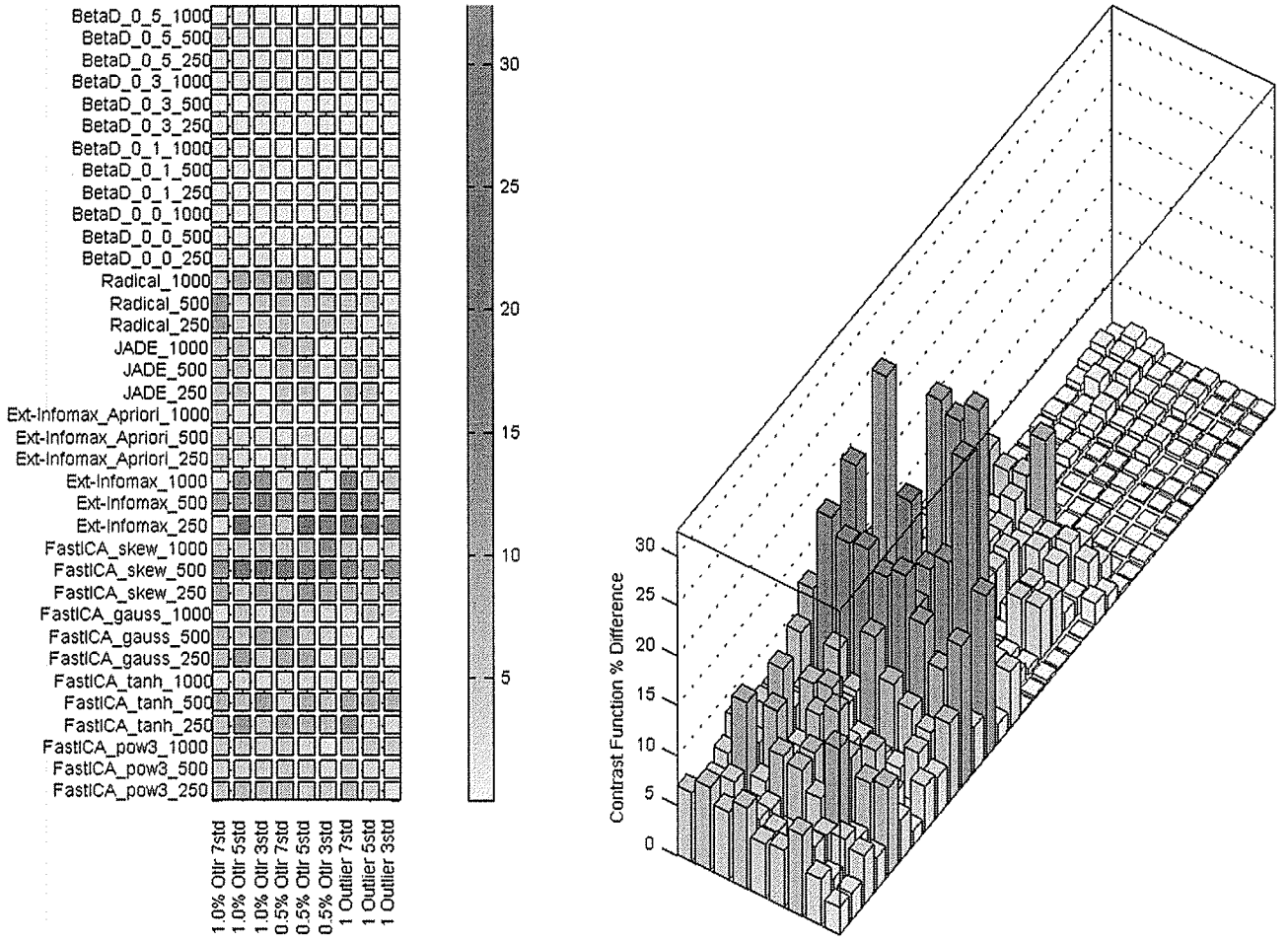


Fig. A.61 Contrast function difference: Asymmetric mixture of 4 Gaussians (Transitional).

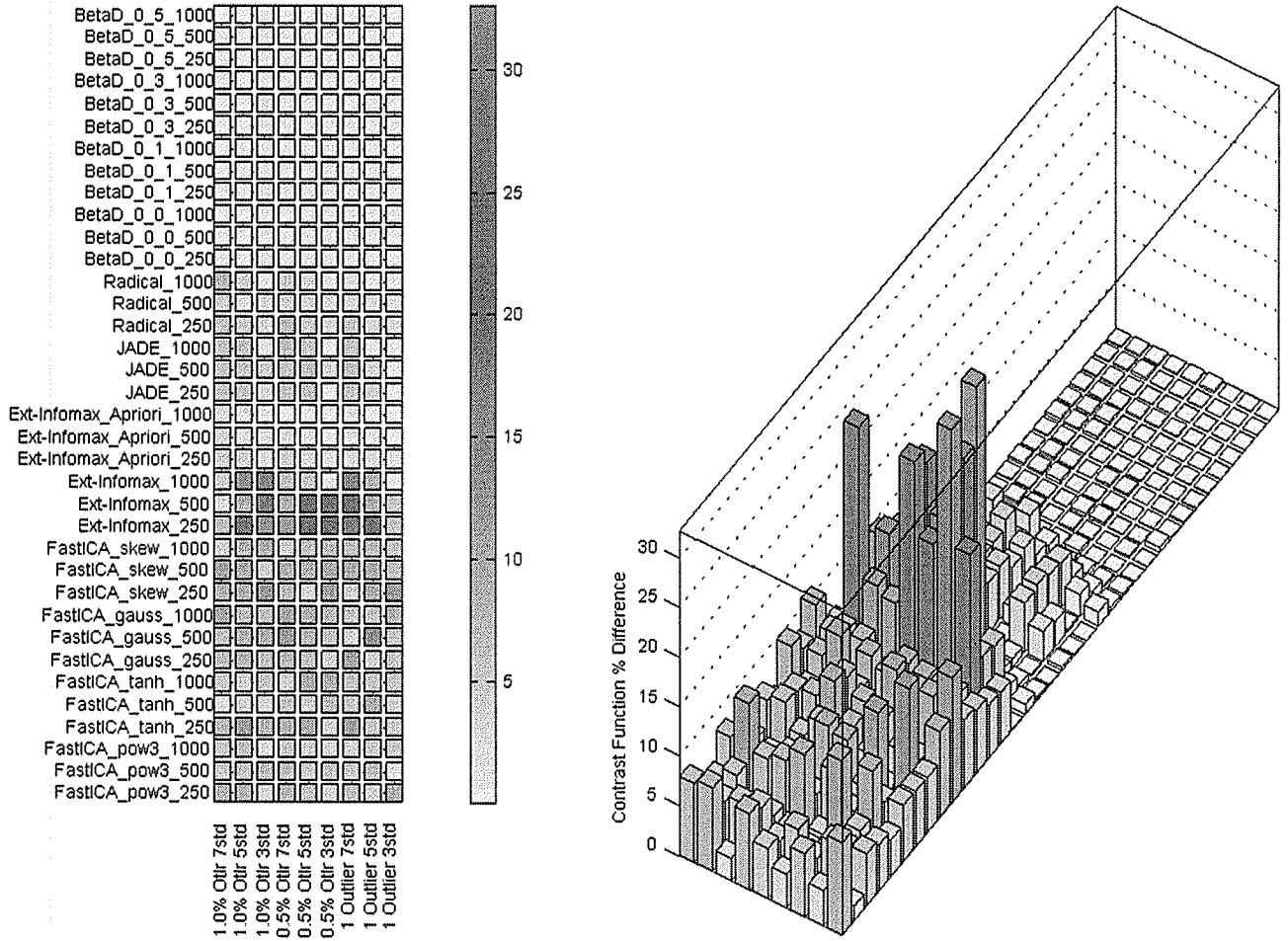


Fig. A.62 Contrast function difference: Asymmetric mixture of 4 Gaussians (Unimodal).

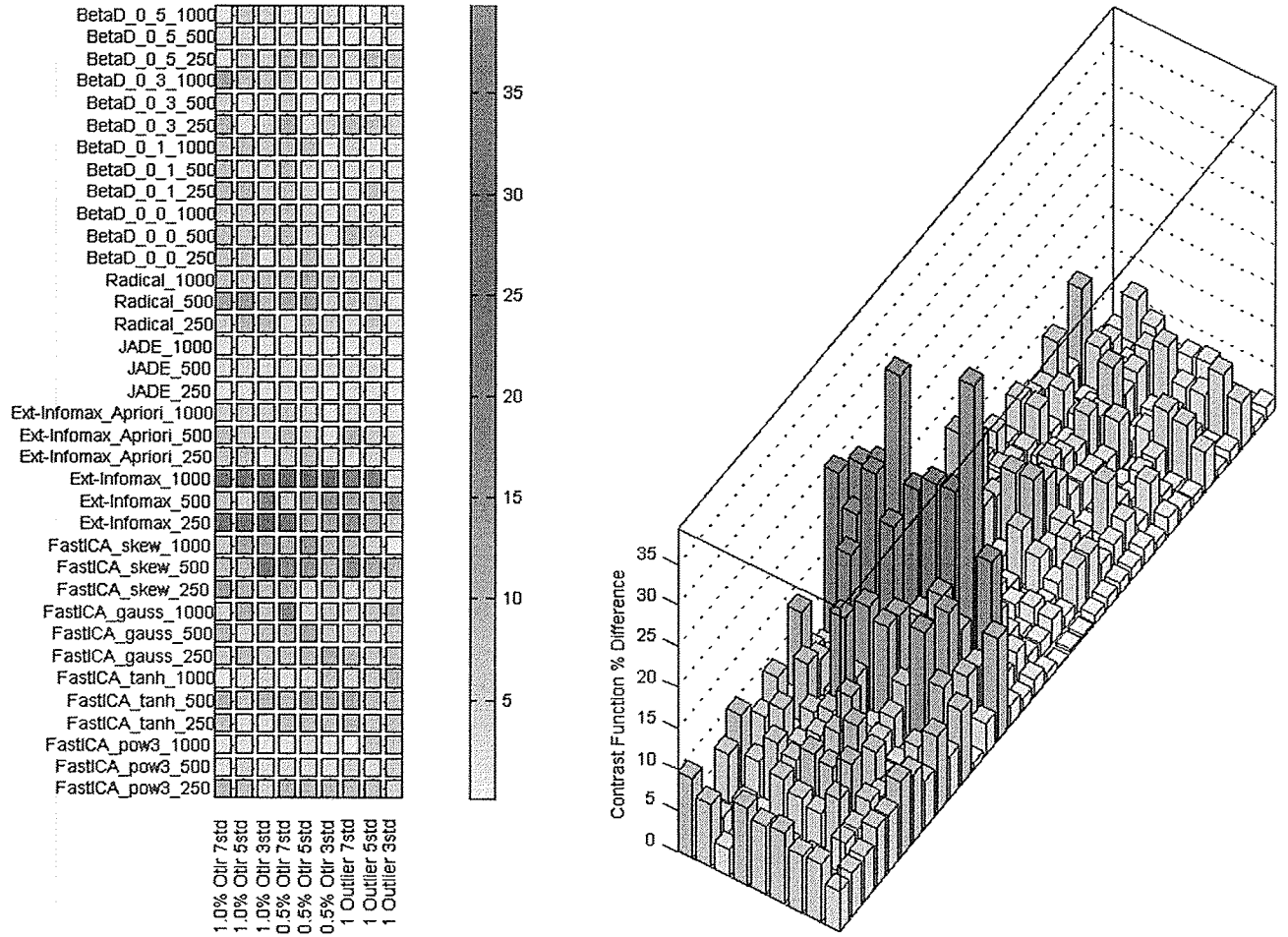


Fig. A.63 Contrast function difference: Gaussian .

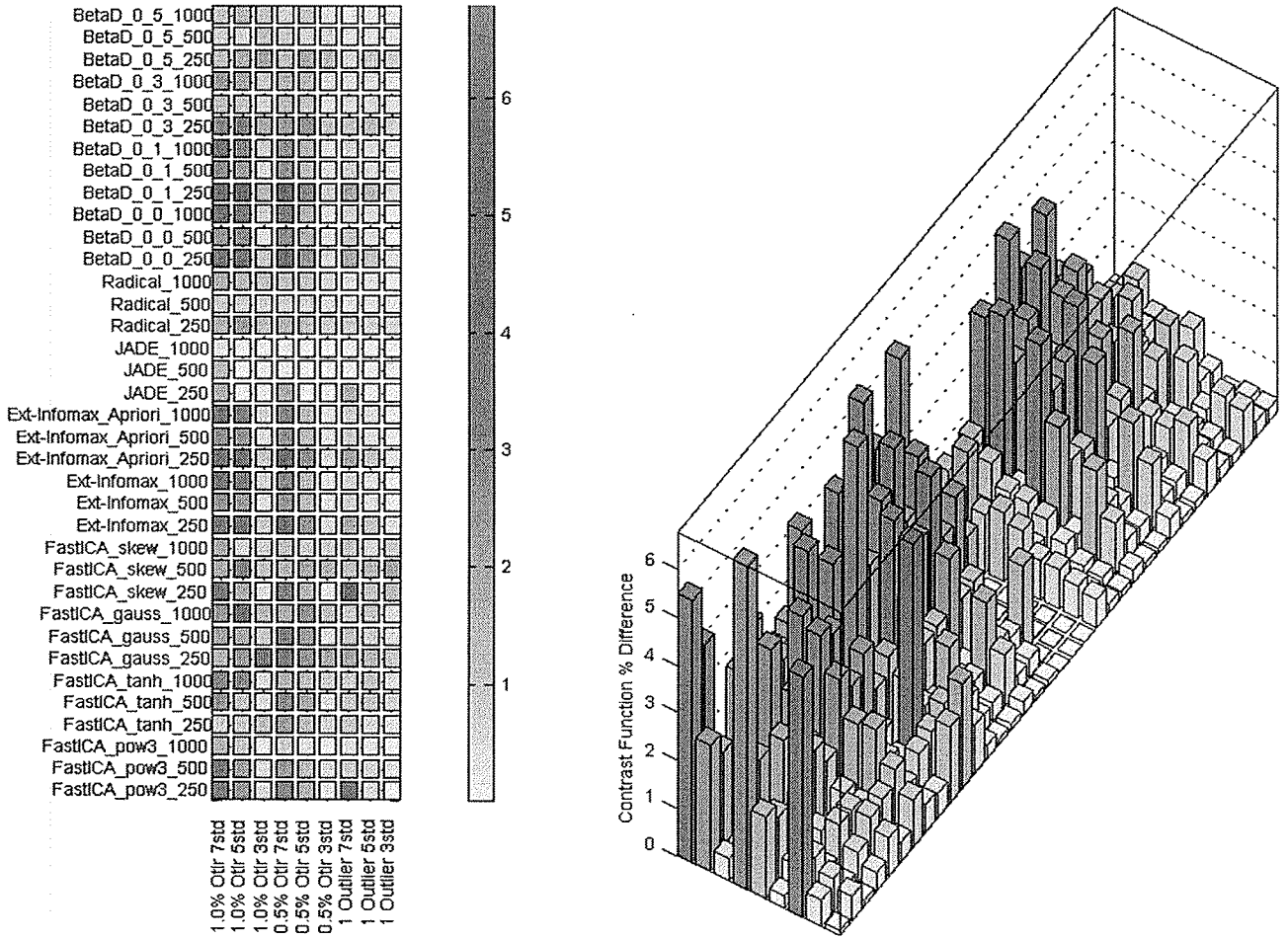


Fig. A.64 Contrast function difference: LogNormal.

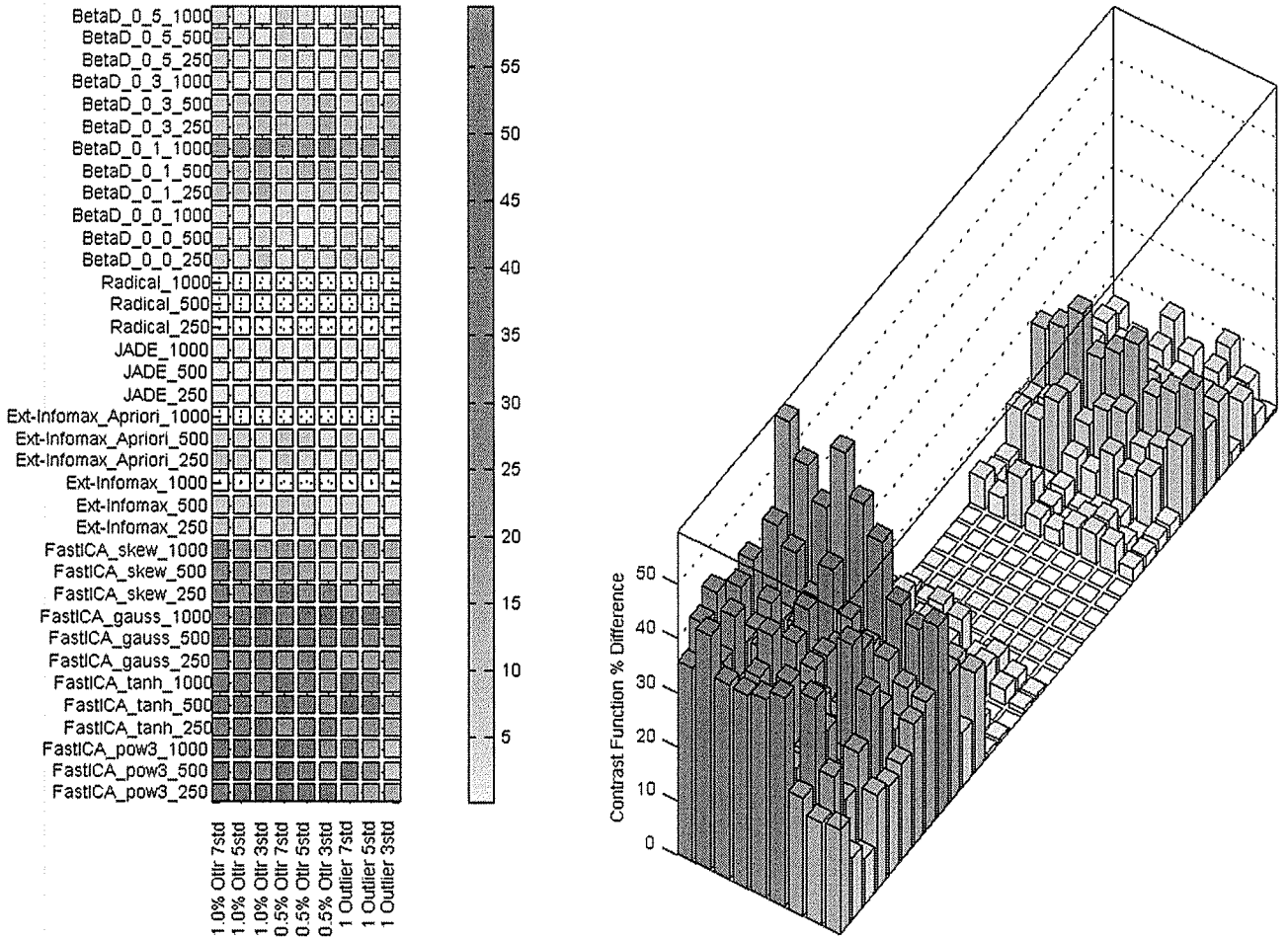


Fig. A.65 Contrast function difference: Pareto.

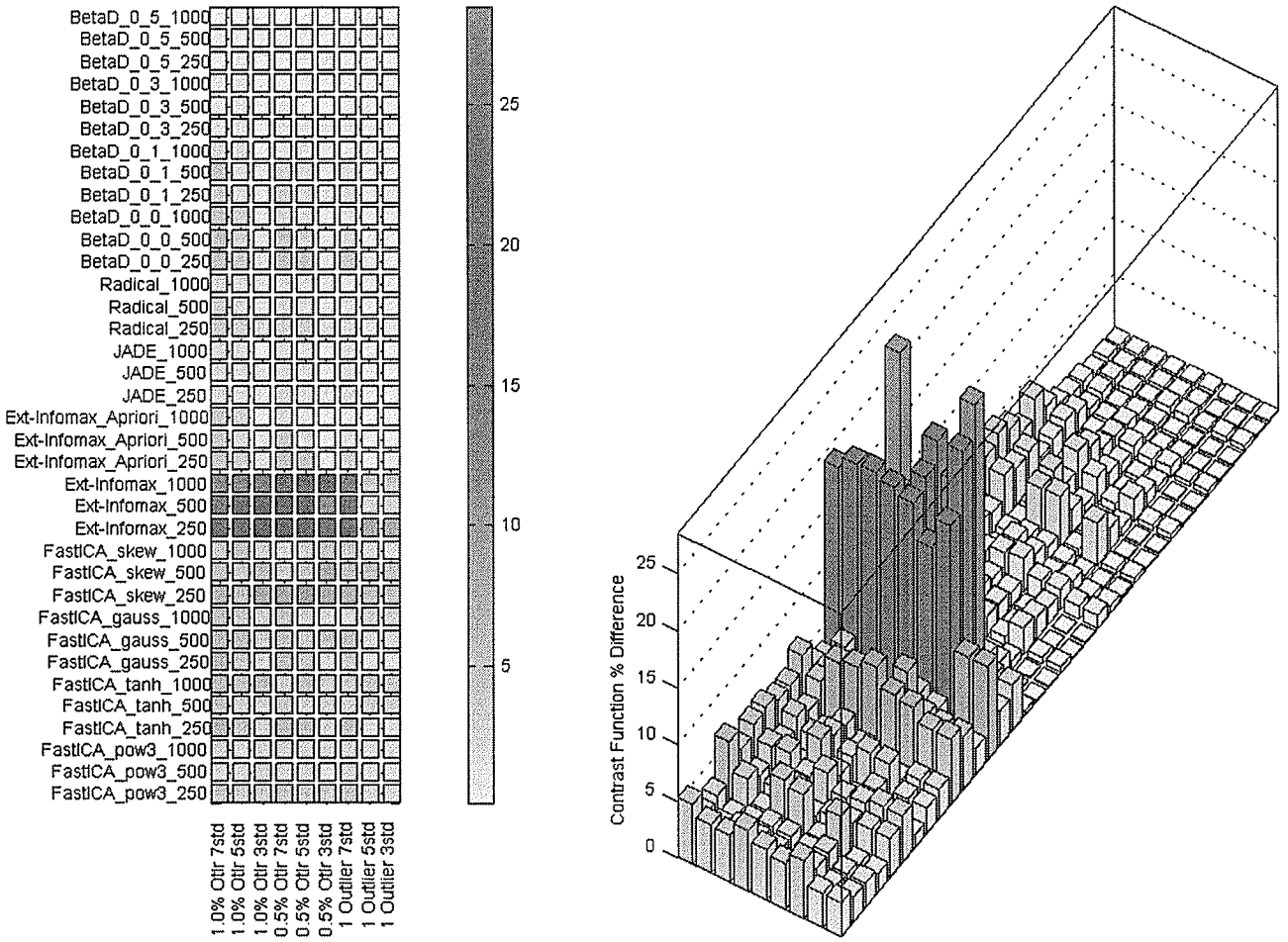


Fig. A.66 Contrast function difference: Random Mixture.

Table A.4 Contrast function difference mixture analysis: Linear regression coefficents and covariance of rotation error with experiment parameters .

Algorithm	Analysis	NumOutliers	Outlier Location
FastICA: pow3	Regression	0.1269	0.4174
FastICA: pow3	Covariance	0.1269	0.4174
FastICA: tanh	Regression	0.0857	0.1776
FastICA: tanh	Covariance	0.0857	0.1776
FastICA: gauss	Regression	0.1399	0.1109
FastICA: gauss	Covariance	0.1399	0.1109
FastICA: skew	Regression	0.0201	-0.0124
FastICA: skew	Covariance	0.0201	-0.0124
Ext-Infomax	Regression	-0.0880	0.1969
Ext-Infomax	Covariance	-0.0880	0.1969
Ext-Infomax: Apriori	Regression	0.2482	0.3888
Ext-Infomax: Apriori	Covariance	0.2482	0.3888
JADE	Regression	0.2685	0.6987
JADE	Covariance	0.2685	0.6987
Radical	Regression	0.2827	0.2465
Radical	Covariance	0.2827	0.2465
BetaD: $\beta = 0.0$	Regression	0.1682	0.1291
BetaD: $\beta = 0.0$	Covariance	0.1682	0.1291
BetaD: $\beta = 0.1$	Regression	0.1705	0.1472
BetaD: $\beta = 0.1$	Covariance	0.1705	0.1472
BetaD: $\beta = 0.3$	Regression	0.1755	0.0928
BetaD: $\beta = 0.3$	Covariance	0.1755	0.0928
BetaD: $\beta = 0.5$	Regression	0..2109	-0.0289
BetaD: $\beta = 0.5$	Covariance	0..2109	-0.0289

Appendix B

Matlab Code and the CD-ROM

The Matlab code and experiment results are available on a Website and are provided on a CD-ROM. The CD-ROM is available from the author and his supervisor. The Website address is http://www.ee.umanitoba.ca/~kinsner/projects/ .

The Website contains the code used in the thesis. The CD-ROM contains the code, the experiment results, the thesis with colour diagrams, and the thesis presentation. The directories are broken into "Code" and "Results". The directories under "Code" are broken up per algorithm and per chapter of the thesis. The "Results" directory is broken up per chapter, and per experiment.