TECHNISCHE UNIVERSITÄT DRESDEN

DRESDEN concept

Faculty for Electrical and Computer Engineering Deutsche Telekom Chair of Communication Networks

Study Thesis

Fast Source Separation

Yunbin Shen
Matriculation number: 4724075
Matriculation year: 2015

15th November 2019

Supervisor
M.Sc. Huanzhuo Wu
Supervising professor
Prof. Dr.-Ing. Dr. h.c. Frank H. P. Fitzek

Statement of authorship

I hereby certify that I have authored this thesisentitled *Fast Source Separation* independently and without undue assistance from third parties. No other than the resources and references indicated in this thesis have been used. I have marked both literal and accordingly adopted quotations as such. There were no additional persons involved in the intellectual preparation of the present thesis. I am aware that violations of this declaration may lead to subsequent withdrawal of the degree.

Dresden, 15th November 2019

Yunbin Shen

Abstract

Independent Component Analysis (ICA) is a method to separate a group of signals, one of the most popular and effective algorithms is FastICA, which has a very good performance in a wide range of applications. However, the speed of FastICA is also not enough for some time-sensitive applications like some data analysis or feature extraction in IoT, which require very low latency. In this thesis, three Improved FastICA algorithms are introduced, which greatly accelerate the speed of source separation. The first one is Improved FastICA with Initial Matrix Estimation, the second one is Improved FastICA with Multi-level Newton' iteration, the third one is Ultra-FastICA, which is the combination of the first and second algorithms. These three new proposed methods reduce the time by 53%-88% compare to the FastICA, without affecting the quality of the separation.

Keywords: ICA, BSS, FastICA, Time-sensitive application, Initial Matrix Estimation, Multi-Level Newton' Iteration

Contents

List of Abbreviations

BSS	Blind source separation
ICA	Independent component analysis
IoT	Internet of things
ECG	Electrocardiogram
Tol	The convergence tolerance of the Newton's Iteration part in FastICA
Lim	Limit of the Tol in Newton's Iteration
SNR	Signal noise ratio

Notation

$\boldsymbol{M}$	A Matrix
$\boldsymbol{m}_i$	The i-th row of matrix $\boldsymbol{M}$
$m_{i,j}$	The element of matrix $\boldsymbol{M}$ in the i-th row and j-th column
${}_{e}^{i}\boldsymbol{M}$	Extracted columns from $\boldsymbol{M}$ with fixed extraction interval e and the position of the first extracted column is i.
$\widehat{\boldsymbol{M}}$	Estimated signals
$\widetilde{\boldsymbol{M}}$	Whitened signals
$length(\boldsymbol{M})$	The number of the columns of matrix $\boldsymbol{M}$
$min\{a_1, \dots, a_n\}$	The minimum value of $a_i, \mathrm{i} = 1, \dots, \mathrm{n}$
$Max\{\boldsymbol{M}\}$	The value of the maximum element in matrix $\boldsymbol{M}$
Λ	The deviation of the randomly generated mixing matrix
Δ	Cosine distance
R_t	The ratio of the time consumption with the tested method to the time consumption with FastICA

1. Introduction

BSS, i.e. Blind Source Separation [1], is widely used nowadays. It is a method for separating the independent components from the mixtures of sources. As an example, sounds are always a mixture of various sources and can be presented as a linear combination of the sources. Then BSS method is used to get the estimated sources from the mixtures, which are obtained from observers. The BSS methods are also used in ECG extraction because the obtained ECG signals are always accompanied by noise, BSS is applied to separate the noise and the useful signals. There are many ICA methods for BSS. FastICA is one of the currently most notable ICA algorithms [2][3] because it is very efficient compared to other ICA methods. However, FastICA is also considered a time-consuming and computational method as it separates the mixed components using Newton's Iteration and a large amount of data. Therefore, the FastICA is not suitable for some time-sensitive applications or some low-power and low-performance devices. Therefore, we need to greatly reduce the computation and increase the speed of FastICA for these situations.

There are also many possibilities for improving the performance of FastICA, like using eighth-order Newton's Iteration [4], finding the optimal iterative step length [5], using a special overrelaxation factor [6], using the modified-M estimated function [7], etc. These methods only improve the performance very little, and not enough for some time-sensitive situations. Therefore, in this project, three improved FastICA algorithms with very high performance are proposed and evaluated, which greatly accelerate the speed of FastICA.

In this thesis, three ICA algorithms are introduced, which are based on the FastICA. The first algorithm is Improved FastICA with Initial Matrix Estimation. The initial separation matrix for Newton's Iteration in FastICA here is replaced by an estimated initial separation matrix, therefore, the time consumption is reduced by decreasing the number of Newton's iteration. The second proposed algorithm is Improved FastICA with Multi-level Newton' iteration, it speeds up the FastICA by reducing the input data of each iteration in FastICA. The third algorithm Ultra-FastICA is the combination of the first 2 methods, i.e. it uses both estimated initial separation matrix and multi-level Newton's Iteration.

In addition, the estimation of the initial separation matrix for FastICA is based on amplitude analysis. With the amplitude analysis we can estimate a mixing matrix, which can be transformed to an initial separation matrix. The algorithm for reducing the input data is data extraction with different intervals for each iteration. However, the original FastICA must be greatly modified to incorporate the initial separation matrix estimation and multi-level Newton's Iteration, which will also be detailed in this thesis.

The rest of the thesis is structured as follows. In Section 2, there are some basis concepts introduced, which is the basic of the new proposed ICA algorithms. In Section 3, the 3 new ICA methods are detailly described. Section 4 are the results of the performance compare to FastICA. Section 5 covers the result discussion.

2. Problem Statement

In this Section, we introduce the required background of 3 new proposed Improved FastICA methods in this thesis. Firstly, is the structure of BSS, which is detailed in Section 2.1. Then in Section 2.1, FastICA is as a BSS method mentioned. In Section 2.2, some details of FastICA is introduced, which not only provides the possibility of improving the FastICA but also is the most important backgrounds for these 3 in this thesis new proposed methods.

2.1. Blind Source Separation

The purpose of blind source separation is to get a set of source signals $\boldsymbol{S} \in \mathbb{R}^{n\times m}$ from a set of mixed signals $\boldsymbol{X} \in \mathbb{R}^{k\times m}$. The source signals $\boldsymbol{S}$ are n independent signals in m time slots. Meanwhile, the mixed signals $\boldsymbol{X}$ are the mixtures of the source signals $\boldsymbol{S}$ obtained from k observers, which are also in m time slots. When the noises are not considered, each component $\boldsymbol{x}_i$ of the mixed signals $\boldsymbol{X}$ can be described as a linear combination of source components $\boldsymbol{s}_i$:

$$x_{i,j} = \sum_{l=1}^{n} a_{i,l} s_{l,j} \tag{1}$$

or

$$\boldsymbol{X} = \boldsymbol{A}\boldsymbol{S} \tag{2}$$

Where $i = 1, \dots, k\ and\ j = 1, \dots, n$ and the matrix $\boldsymbol{A}$ is the mixing matrix. Here assume that the numbers of the source signals and the mixed signals are the same, i.e. $k = n$, the mixing matrix will be a square matrix and has an inverse. The ideal source separation problem without considering noise can be shown in Figure 1.

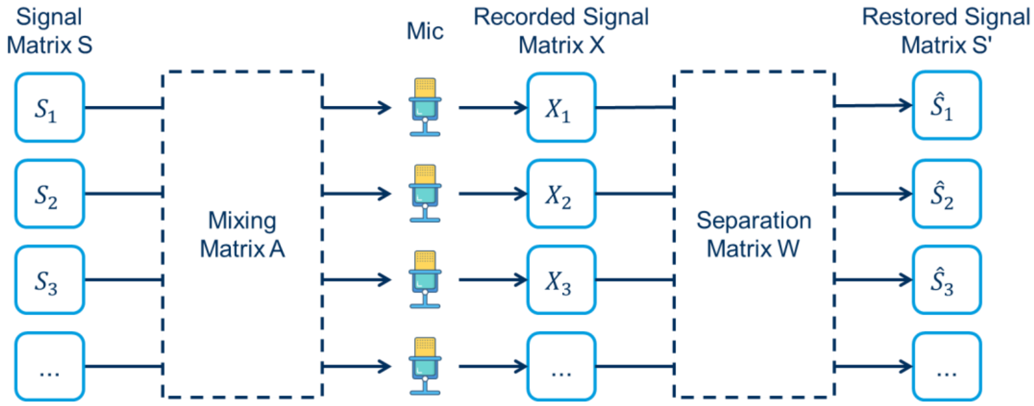

Fig. 1 Blind Source Separation Model

The task of the BSS is to calculate the separation matrix $\boldsymbol{W}$ and so that the estimated source signals $\widehat{\boldsymbol{S}}$ can be obtained by the separation matrix $\boldsymbol{W}$ and the mixed signals $\boldsymbol{X}$ as in Eq. (3). Meanwhile, in the ideal situation, the separation matrix $\boldsymbol{W}$ and the inverse of mixing matrix $\boldsymbol{A}^{-1}$ are equal.

$$\widehat{\boldsymbol{S}} = \boldsymbol{A}^{-1}\boldsymbol{A}\boldsymbol{S} = \boldsymbol{A}^{-1}\boldsymbol{X} = \boldsymbol{W}\boldsymbol{X} \tag{3}$$

The are several algorithms for BSS, and the most widely used method is FastICA [3], which is proposed by A. Hyvärinen and E. Oja in 1999. However, the speed of FastICA is also not enough for some time-sensitive or low-power applications because of its large amount of calculation.

2.2. The FastICA Algorithm

2.2.1. Newton's Iteration of FastICA Algorithm

The proposed BSS algorithm in this thesis is based on the FastICA algorithm in scikit-learn [8]. This whole algorithm includes three parts, i.e. centering, whitening and Newton's Iteration. The first 2 parts are preprocessing for ICA, and the third part, Newton's Iteration is the main part, which is the most important part and takes most of the time. For a normal BSS situation, the source signals $\widehat{\boldsymbol{S}}$ are independent, i.e. the mutual information between each other is 0, i.e. the negentropy is maximum. The principle of Newton's Iteration in FastICA is to find the separation matrix $\boldsymbol{W}$, so that the estimated source signals $\widehat{\boldsymbol{S}}$ have the maximum negentropy. The equation for solving $\boldsymbol{W}$ is

$$\boldsymbol{E}\{\boldsymbol{X}\boldsymbol{g}(\boldsymbol{W}^{\boldsymbol{T}}\boldsymbol{X})\} - ß\boldsymbol{W} = 0 \tag{4}$$

This equation can be solved by Newton's Iteration, which can be approximated as:

$$W^{+} = W - \frac{E\{Xg(W^{T}X)\} - ßW}{E\{g'^{(W^{T}X)}\} - ß} \tag{5}$$

Then we can get a more simplified approximation from Eq. (5):

$$W^{+} = E\{Xg(W^{T}X)\} - E\left\{g'^{(W^{T}X)}\right\}W \tag{6}$$

And this is the used equation in scikit-learn [8], with which the separation matrix for BSS problems can be calculated.

2.2.2. Tolerance of Newton's Iteration

Generally, the tolerance Tol of Newton's Iteration part in FastICA is presented in Eq. (7), it indicates the convergence of the separation matrix.

$$Tol = Max\{|W^{+}W^{T} - I|\} \tag{7}$$

In practical application, a determined tolerance limit Lim is needed, which is used to decide the end of the Newton's Iteration, otherwise this iteration is endless. When the value of Tol is smaller than tolerance limit Lim the program will jump out of the iteration part and return the results. Therefore, the form of the FastICA algorithm in scikit-learn is as follows:

1. Centralization
2. Whiten, $\widetilde{X} = VX$
3. Choose an initial (e.g. random) W_0
4. Let $W^{+} = E\{\widetilde{X}g(W^{T}\widetilde{X})\} - E\left\{g'^{(W^{T}\widetilde{X})}\right\}W$
5. Decorrelation, $W = W^{+}/\|W^{+}\| = \left(W^{+}{W^{+}}^{T}\right)^{-\frac{1}{2}} W^{+}$
6. If $Tol > Lim$, go back to 4, otherwise go to 7
7. Let $\widehat{S} = W\widetilde{X}$

Where the step 3-6 is the Newton's Iteration part in FastICA. Note that the W must be decorrelated in each loop, which is very important for the initial separation matrix transformation in Section 3.1.3 and the tolerance determination in Section 3.2.2. The step 4-6 can be presented as a specially defined function in Eq. (8):

$$W, Tol = f_{Newton's\,Iteration}(\widetilde{X}, W_0, Lim) \tag{8}$$

Where the Tol is calculated in Eq. (7), here is the value of tolerance when the program jumps out of Newton's Iteration. For a multi-level Newton's Iteration, the tolerance limit of each level of iteration should be different, with will be detailly talked in section 3.2.

Together with the Eq. (6) and Eq. (8), which is used in FastICA. The FastICA has many ways to optimize, in this thesis, there mainly are two ways of them are talked about, one is giving a proper initial separation matrix $\boldsymbol{W_0}$ instead of a random matrix to reduce the number of iteration, which is introduced in Section 3.1, another is to adjust the number of used time slots of $\boldsymbol{X}$ in each step of Newton's Iteration to reduce the time consumption, which is detailed in Section 3.2.

3. Methodology

In this Section, 3 improved FastICA algorithms are introduced. In Section 3.1, a method is introduced to estimate a roughly mixing matrix, then this matrix can be transformed to a separation matrix, which can used as an initial separation matrix in FastICA. In Section 3.2, a method of reducing the input data in each step of Newton's Iteration is introduced, the relationship of data extraction interval, change of amplitude and tolerance limit is given and proved. In Section 3.3 is the description of Ultra-FastICA, this method uses both initial matrix estimation and multi-level data extraction for each step of Newton's Iteration.

3.1. Initial Separation Matrix $\boldsymbol{W_0}$ Estimation

In this section, we will firstly introduce a method to calculate an estimated mixing matrix $\widehat{\boldsymbol{A}}$ from the mixed signals $\boldsymbol{X}$, which can be obtained from the observers. Then, we will introduce how to convert this estimated mixing matrix $\widehat{\boldsymbol{A}}$ into an initial separation matrix $\boldsymbol{W_0}$, which can be directly used in Eq. (6) in FastICA to reduce the iteration number.

3.1.1. Relationship of Mixed Signals $\boldsymbol{X}$ and Mixing Matrix $\widehat{\boldsymbol{A}}$

According to the Eq. (1), the expected value of mixtures $\boldsymbol{X}$ can also be described by a linear combination of the means of source signals $\boldsymbol{S}$ in the Eq. (9):

$$E(\boldsymbol{x}_i) = \sum_{l=1}^{n} a_{i,l} E(\boldsymbol{s}_l) \tag{9}$$

Where $i = 1, \dots, n$. Therefore, the ratio of means from two arbitrary mixtures of $\boldsymbol{X}$ can be described by Eq. (10):

$$\frac{E(\boldsymbol{x}_i)}{E(\boldsymbol{x}_j)} = \frac{\sum_{l=1}^{n} a_{i,l} E(\boldsymbol{s}_l)}{\sum_{l=1}^{n} a_{j,l} E(\boldsymbol{s}_l)} = \frac{a_{i,1}E(\boldsymbol{s}_1) + a_{i,2}E(\boldsymbol{s}_2) + \cdots + a_{i,n}E(\boldsymbol{s}_n)}{a_{j,1}E(\boldsymbol{s}_1) + a_{j,2}E(\boldsymbol{s}_2) + \cdots + a_{j,n}E(\boldsymbol{s}_n)} \tag{10}$$

When the value of $E(\boldsymbol{s}_k)$ is big enough and satisfies the Eq. (11):

$$
\begin{aligned}
|E(\boldsymbol{s}_k)| &\gg \left| \sum_{l=1, l\neq k}^{n} a_{i,l} E(\boldsymbol{s}_l) \right| \\
|E(\boldsymbol{s}_k)| &\gg \left| \sum_{l=1, l\neq k}^{n} a_{j,l} E(\boldsymbol{s}_l) \right|
\end{aligned} \tag{11}
$$

Then we can get the approximate ratio of all the elements $a_{i,l}$ in the mixing matrix $\boldsymbol{A}$ according to the Eq. (12):

$$
\frac{a_{i,1}}{a_{j,1}} \approx \frac{E(\boldsymbol{x}_i)}{E(\boldsymbol{x}_j)} \tag{12}
$$

Assume that the elements on the diagonal of matrix A are the largest in each column, i.e. each observer must be close to a different source, then we can use a special algorithm, which will be introduced next, to get the following estimated mixing matrix $\widehat{\boldsymbol{A}}$:

$$
\widehat{\boldsymbol{A}} = \begin{bmatrix} 1 & \cdots & \hat{a}_{1,n} \\ \hat{a}_{2,1} & \cdots & \hat{a}_{1,n} \\ \vdots & \ddots & \vdots \\ \hat{a}_{n,1} & \cdots & 1 \end{bmatrix} \approx \begin{bmatrix} \frac{a_{1,1}}{a_{1,1}} & \cdots & \frac{a_{1,n}}{a_{n,n}} \\ \frac{a_{2,1}}{a_{1,1}} & \cdots & \frac{a_{1,n}}{a_{n,n}} \\ \vdots & \ddots & \vdots \\ \frac{a_{n,1}}{a_{1,1}} & \cdots & \frac{a_{n,n}}{a_{n,n}} \end{bmatrix} = \boldsymbol{A} \times \begin{bmatrix} \frac{1}{a_{1,1}} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{1}{a_{n,n}} \end{bmatrix} = \boldsymbol{AK} \tag{13}
$$

Where the estimated mixing matrix $\widehat{\boldsymbol{A}}$ can be expressed as mixing matrix $\boldsymbol{A}$ multiplied by a diagonal matrix $\boldsymbol{K}$. So that we can get the relationship of the ideal separation matrix $\boldsymbol{W}$ and the estimated separation matrix $\widehat{\boldsymbol{W}}$, which is the inverse of the matrix $\widehat{\boldsymbol{A}}$:

$$
\widehat{\boldsymbol{W}} = \widehat{\boldsymbol{A}}^{-1} \approx (\boldsymbol{AK})^{-1} = \boldsymbol{K}^{-1}\boldsymbol{A}^{-1} = \boldsymbol{K}^{-1}\boldsymbol{W} \tag{14}
$$

Note that $\boldsymbol{K}^{-1}$ is a diagonal matrix, which only changes the amplitude of each separated source signal. When the error of the estimated mixing matrix $\widehat{\boldsymbol{A}}$ is small enough, this estimated separation matrix $\widehat{\boldsymbol{W}}$ can be directly used to separate the mixed signals. The matrix $\widehat{\boldsymbol{W}}$ is very similar to the FastICA calculated separation matrix, which can only separate the mixed signals without considering the change of amplitude of source signals. So that the matrix $\widehat{\boldsymbol{W}}$ is very similar to the calculated separation matrix of FastICA.

3.1.2. Construction of Estimated Mixing Matrix $\widehat{\boldsymbol{A}}$

Note that the ideal condition for this algorithm is that the elements on the diagonal of matrix A are the largest in each column, under which the next to be introduced method can be optimal. To simplify the problem, the mixing matrix $\boldsymbol{A}$ can be scaled and shifted to match the Eq. (15), i.e. the diagonal element is 1 and the other elements are between 0 and 1.

$$\begin{cases} a_{i,j} = 1 & \boldsymbol{i = j} \\ \boldsymbol{0} < a_{i,j} < 1 & \boldsymbol{i \neq j} \end{cases} \tag{15}$$

The results of scaling and shifting $\boldsymbol{A}$ are the permutation of estimated sources order scaling of estimated sources' amplitude. These two effects are also resulted by FastICA itself.

Starting with analyzing the 2 sources conditions, we can also extend to the arbitrary number of sources. In this situation the mixed signals can be described by Eq. (16):

$$\begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \end{bmatrix} = \begin{bmatrix} 1 & a_{1,2} \\ a_{2,1} & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{s}_1 \\ \boldsymbol{s}_2 \end{bmatrix} \quad s.t. 0 < a_{1,2}, a_{2,1} < 1 \tag{16}$$

When we replace the $\boldsymbol{x}_1, \boldsymbol{x}_2$ with the function of $\boldsymbol{s}_1$, the new equation can be represented in the Eq. (17):

$$\begin{aligned} f_1(s_{1,j}) &= s_{1,j} + a_{1,2}s_{2,j} = x_{1,j} \\ f_2(s_{1,j}) &= a_{2,1}s_{2,j} + s_{2,j} = x_{2,j} \end{aligned} \tag{17}$$

When $f_1(s_{1,j}) > f_2(s_{1,j})$, it is easily to get:

$$s_{1j} > \frac{1 - a_{1,2}}{1 - a_{2,1}} \geq 0 \quad s.t. s_{2,j} \geq 0 \tag{18}$$

When $f_1(s_{1,j}) \succ f_2(s_{1,j})$,

$$s_{1,j} > -\frac{1 + a_{1,2}}{1 + a_{2,1}} \geq 0 \quad s.t. s_{2,j} < 0 \tag{19}$$

As illustrated in Fig. 2.

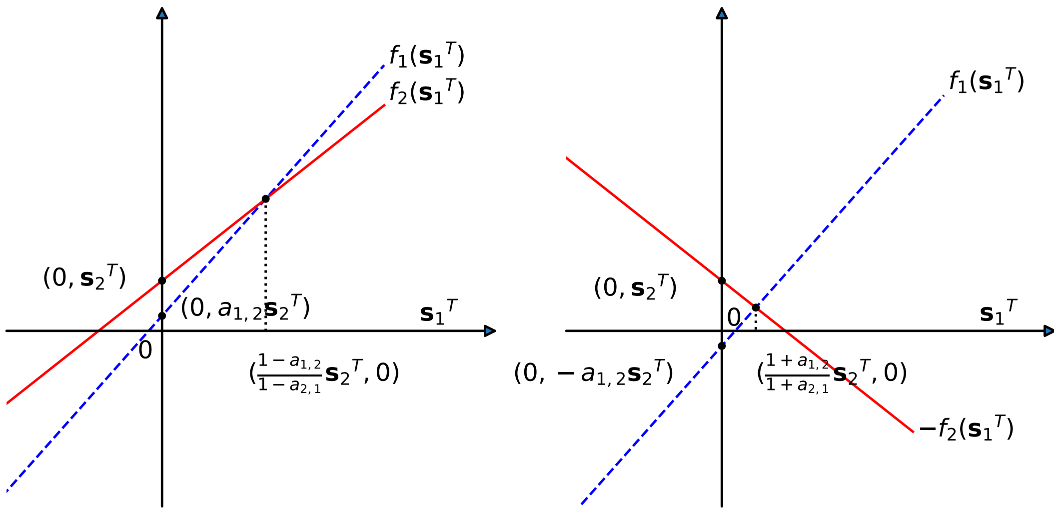

Fig. 2 $s_{1,j}$ *and* $x_{1,j}, -x_{2,j}$, when $s_{2,j} \geq 0$ (left) and $s_{2,j} < 0$ (right)

Note the mixed the signals $x_{1,j}$ and $x_{2,j}$ are signals from different observers at the same moment. The known mixed signals $x_{1,j}$ and $x_{2,j}$, that meet both conditions (18) (19), i.e. fulfill the constraint in Eq. (20) , are selected to build the new matrix $\widehat{\boldsymbol{X}}$, which will be used to build the estimated mixing matrix $\widehat{\boldsymbol{A}}$ in the next.

$$\widehat{\boldsymbol{X}} = \begin{bmatrix} x_{1,b_1} & \cdots & x_{1,b_t} \\ x_{2,b_1} & \cdots & x_{2,b_t} \end{bmatrix} \quad s.t. \quad \forall x_{1,b_i} > |x_{2,b_i}|, t < m \tag{20}$$

Where $i = 1, \ldots, t$ and m is the number of the time slots of $\boldsymbol{X}$. Then the original signals in the same moments can be represented in Eq. (21):

$$\widehat{\boldsymbol{S}} = \begin{bmatrix} s_{1,b_1} & \cdots & s_{1,b_t} \\ s_{2,b_1} & \cdots & s_{2,b_t} \end{bmatrix} \quad s.t. \quad \forall x_{1,b_i} > |x_{2,b_i}|, t < m \tag{21}$$

It is easy to see

$$\begin{gathered} x_{1,b_i} + x_{2,b_i} = f_1(s_{1,b_i}) + f_2(s_{1,b_i}) > 0 \\ x_{1,b_i} - x_{2,b_i} = f_1(s_{1,b_i}) - f_2(s_{1,b_i}) > 0 \end{gathered} \tag{22}$$

With Eq. (17) and Eq. (22) it is easy to get that

$$\begin{gathered} f_1(s_{1,b_i}) + f_2(s_{1,b_i}) = (1 + a_{2,1})s_{1,b_i} + (1 + a_{1,2})s_{2,b_i} > 0 \\ f_1(s_{1,b_i}) - f_2(s_{1,b_i}) = (1 - a_{2,1})s_{1,b_i} - (1 - a_{1,2})s_{2,b_i} > 0 \end{gathered} \tag{23}$$

Hence,

$$\begin{cases} s_{1,b_i} > -\dfrac{1 + a_{1,2}}{1 + a_{2,1}} s_{2,b_i} = f_3(s_{2,b_i}) \geq 0 & s_{2,b_i} < 0 \\ s_{1,b_i} > \dfrac{1 - a_{1,2}}{1 - a_{2,1}} s_{2,b_i} = f_4(s_{2,b_i}) \geq 0 & s_{2,b_i} > 0 \end{cases} \tag{24}$$

The Fig. 3 shows the joint distribution of all the selected signals s_{1,b_i}, s_{2,b_i}, the values marked as red are constrained by the Eq. (24), i.e. the selected signals. For an arbitrary number of sources, the principle is very similar. Together with (16) and (24)， the boundary of the joint distribution of s_{1,b_i}, s_{2,b_i} can be obtained in Eq. (25):

$$\begin{gathered} -2 \leq \frac{f_3(s_{2,b_i})}{\partial s_{2,b_i}} \leq -\frac{1}{2} \\ 0 \leq \frac{f_4(s_{2,b_i})}{\partial s_{2,b_i}} \leq \infty \end{gathered} \tag{25}$$

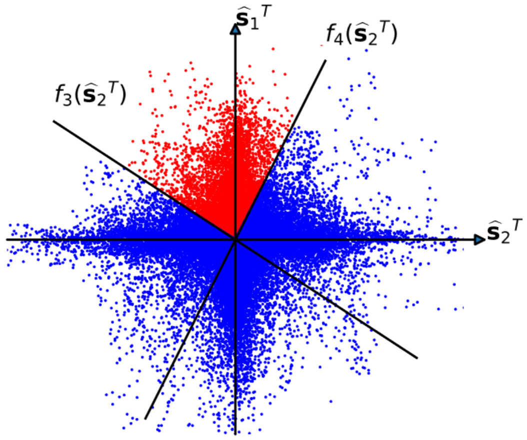

Fig. 3 Joint Distribution of the Signals s_{1,b_i}, s_{2,b_i}

Note that in Fig. 3 the locations of red points are (s_{2,b_i}, s_{1,b_i}), which are from $\widehat{\boldsymbol{S}}$. According to the Eq. (24), it is easy to see that the original signals $\widehat{\boldsymbol{S}}$ of the selected signals $\widehat{\boldsymbol{X}}$ fulfil the Eq. (26). And the values of s_{1,b_i} are always positive.

$$\begin{cases} \exists s_{2,b_i} \geq 0 \\ \exists s_{2,b_i} \leq 0 \\ \forall s_{1,b_i} \geq 0 \end{cases} \tag{26}$$

Together with the Eq. (25) and Eq. (26), it clearly shows that $\left|\sum_{i=1}^{t} s_{1,b_i}\right| > \left|\sum_{i=1}^{t} s_{2,b_i}\right|$, when the original signals $\boldsymbol{S}$ are as normal as the signal in nature. There are 3 extreme situations presented in Eq. (27), Eq. (28) and Eq. (29):

$$0 < a_{1,2} \approx a_{2,1} < 1 \quad \begin{cases} \dfrac{f_3(s_{2,b_i})}{\partial s_{2,b_i}} \approx -1 \\ \dfrac{f_4(s_{2,b_i})}{\partial s_{2,b_i}} \approx 1 \end{cases} \tag{27}$$

$$a_{1,2} \approx 1, a_{2,1} = 0 \quad \begin{cases} \dfrac{f_3(s_{2,b_i})}{\partial s_{2,b_i}} \approx -2 \\ \dfrac{f_4(s_{2,b_i})}{\partial s_{2,b_i}} \approx 0 \end{cases} \tag{28}$$

$$a_{1,2} = 0, a_{2,1} \approx 1 \quad \begin{cases} \dfrac{f_3(s_{2,b_i})}{\partial s_{2,b_i}} \approx -\dfrac{1}{2} \\ \dfrac{f_4(s_{2,b_i})}{\partial s_{2,b_i}} \approx \infty \end{cases} \tag{29}$$

Where Eq. (27) shows the lines of f_3 and f_4 are symmetric, that means $\left|E(s_{2,b_i})\right| \approx 0$, so that $\left|E(s_{1,b_i})\right| \gg \left|E(s_{2,b_i})\right| \approx 0$, and the mixing matrix can already be estimated. Eq. (28) and Eq. (29)

shows the situations that $\left|E\left(s_{1,b_i}\right)\right|$ is a little bigger than $\left|E\left(s_{2,b_i}\right)\right|$, under which the estimated mixing matrix is not so exactly. Therefore, with Eq. (17), we have

$$\frac{E\left(x_{2,b_i}\right)}{E\left(x_{1,b_i}\right)} = \frac{a_{2,1}E\left(s_{1,b_i}\right) + E\left(s_{2,b_i}\right)}{E\left(s_{1,b_i}\right) + a_{1,2}E\left(s_{2,b_i}\right)} = \hat{a}_{2,1} \approx \frac{a_{2,1}E\left(s_{1,b_i}\right)}{E\left(s_{1,b_i}\right)} = a_{2,1} \tag{30}$$

And another element $a_{1,2}$ can be got in the same way.

Generally, the elements in estimated mixing matrix $\widehat{\boldsymbol{A}}$ can be obtained in the Eq. (31):

$$\begin{gathered}\frac{E\left(x_{k,b_{l_i}}\right)}{E\left(x_{l,b_{l_i}}\right)} = \hat{a}_{i,l} \approx \frac{a_{i,l}}{a_{l,l}} \\ s.t. \boldsymbol{B_l} = \left[b_{l_1}, b_{l_2}, b_{l_3}, \dots, b_{l_{t_l}}\right], \forall x_{l,b_{l_i}} > \left|x_{k,b_{l_i}}\right| \\ l = 1, \cdots, n \quad k = 1, \cdots, n \quad k \neq l\end{gathered} \tag{31}$$

Where $\boldsymbol{B}_l$ are time positions of all the mixed signals, which fulfills the constraint in Eq. (31). For different l the matrix $\boldsymbol{B}_l$ is also variable.

3.1.3. Transformation of Estimated Mixing Matrix $\widehat{\boldsymbol{A}}$ to initial Matrix $\widehat{\boldsymbol{W}}_{\boldsymbol{0}}$

In the last section, we have introduced the calculation of matrix $\widehat{\boldsymbol{A}}$. Because the initial matrix in FastICA is not a mixing matrix but a separation matrix, the matrix $\widehat{\boldsymbol{A}}$ must be transformed to a separation matrix. According to Eq. (3), the initial matrix can be presented in Eq. (32):

$$\widehat{\boldsymbol{W}}_0 = \widehat{\boldsymbol{A}}^{-1} \tag{32}$$

However, this separation matrix is also not ideal to be used in FastICA, according to the basic form of FastICA, the finally used matrix $\widehat{\boldsymbol{W}}_0$ should be got from Eq. (33)

$$\widehat{\boldsymbol{W}}_0 = \frac{\widehat{\boldsymbol{W}}_0}{\left\|\widehat{\boldsymbol{W}}_0\right\|} \tag{33}$$

According to Eq. (31), Eq. (32) and Eq. (33), we can get the roughly initial separation matrix $\widehat{\boldsymbol{W}}_0$, and note that the function for initial separation matrix $\widehat{\boldsymbol{W}}_0$ estimation is marked as $f_{Initial\ Matrix}(\boldsymbol{X})$ in the next.

3.1.4. Estimation of Initial Separation Matrix $\widehat{\boldsymbol{W}}_{\boldsymbol{0}}$ with Partial Signal

Above we have introduced the algorithm to get the estimated initial separation matrix $\widehat{\boldsymbol{W}}_{\boldsymbol{0}}$. However, due to its character, the calculated initial matrix $\widehat{\boldsymbol{W}}_{\boldsymbol{0}}$ is always not exactly, so that it is no need to use all the signals to make the results accurately. According to the Nyquist–Shannon sampling theorem, the sampling frequency must fulfil the Eq. (34):

$$f_s \geq 2f_m \tag{34}$$

Normally, the sampling frequency is much higher than the signal frequency. However, for rough separation matrix computation, it is no need to use all the frequency domain. Meanwhile, a low-frequency domain has less effect of the propagation delay of signals. Therefore, a signal extraction is very suitable for getting partial signals to the computation of the estimated initial separation matrix $\widehat{\boldsymbol{W}}_{\boldsymbol{0}}$.

Assume that ${}_{e}^{i}\boldsymbol{X}$ is the extracted signal from $\boldsymbol{X}$ with extraction interval e and the position of the moment of firstly extracted signals i. Where $\boldsymbol{i = 1, 2, \ldots, e}$. Usually, the value of $\boldsymbol{i}$ is 1. The relationship of ${}_{e}^{i}\boldsymbol{X}$ and $\boldsymbol{X}$ is presented in Eq. (35)

$${}_{e}^{i}\boldsymbol{X} = \begin{bmatrix} x_{e_1} & x_{e_2} & \cdots & x_{e_{\lfloor n/e \rfloor}} \end{bmatrix} = \begin{bmatrix} x_i & x_{i+e} & x_{i+2e} & \cdots & x_{i+e\lfloor n/e-1 \rfloor} \end{bmatrix} \tag{35}$$

However, the optimal extraction for different original signals is variable, because the frequency response of different signals is not the same. According to the Eq. (31), the calculation accuracy of the estimated initial separation matrix $\widehat{\boldsymbol{W}}_0$ is related to the number of elements in vector $\boldsymbol{B_l}$, especially the vector $\boldsymbol{B}_l$ with the smallest number of elements. Note that the smallest number of Elements of $\boldsymbol{B}_l$:

$$N_{B_l,min} = min\{length(\boldsymbol{B}_1), length(\boldsymbol{B}_2), \ldots, length(\boldsymbol{B}_n)\} \tag{36}$$

To determine the optimal extraction interval, there are the following steps:

1. Extraction interval $e = step$
2. Get $\widehat{\boldsymbol{S}}$ through BSS
3. Generate random mixing matrix $\boldsymbol{A}$, let $\boldsymbol{X}' = \boldsymbol{A}\widehat{\boldsymbol{S}}$
4. Signal Extraction and get ${}_{e}^{1}\boldsymbol{X}'$ for obtained mixed signal $\boldsymbol{X}'$
5. Estimation of initial separation matrix $\widehat{\boldsymbol{W}}_0$ with ${}_{e}^{1}\boldsymbol{X}$ and get $\boldsymbol{B}_l$
6. Find the smallest number of Elements of $\boldsymbol{B}_l$, i.e. $N_{\boldsymbol{B}_l,min}$
7. Repeat step 3-6 for $N_{samples}$ times, get the average $\overline{N}_{\boldsymbol{B}_l,min} = \frac{1}{N_{samples}} \sum N_{\boldsymbol{B}_l,min}$
8. When $\overline{N}_{\boldsymbol{B}_l,min} > N_{minimum}$, then $e = e + step$ and go back to the step 2, otherwise the extraction interval is $e - step$

Where $N_{minimum} = 20$, $N_{samples} = 30$ in this project, i.e. the sources are human voices with sampling frequency 16000Hz. The value of $N_{minimum}$ can be determined according to the actual situation. For more unstable source signals, i.e. the frequency varies widely, the value of $N_{minimum}$ should be bigger. Meanwhile, the larger the $N_{minimum}$ is, the more stable but longer the time spent on Estimation of initial separation matrix $\widehat{\boldsymbol{W}}_0$. Once the extraction interval is determined for a situation, it should be stable, and shouldn't be changed for the next use. Because of its robustness, the extraction interval shouldn't be very precious so that $step = 5$ in this thesis to speed up the calculation.

3.1.5. Improved FastICA with Initial Matrix Estimation

The basic form of improved FastICA with initial matrix estimation is presented as follows:

1. Centralization
2. Signal Extraction, ${}_{e}^{i}\boldsymbol{X} = f_{Extraction}(\boldsymbol{X}, e)$
3. Initial Matrix $\widehat{\boldsymbol{W}}_0$ Estimation, $\widehat{\boldsymbol{W}}_0 = f_{Initial\ Matrix}({}_{e}^{i}\boldsymbol{X})$
4. Whiten, $\tilde{X} = VX$
5. Newton's Iteration with $\boldsymbol{W}_0$, $\boldsymbol{W} = f_{Newton's\ Iteration}(\widetilde{\boldsymbol{X}}, \widehat{\boldsymbol{W}}_0, Lim)$
6. Let $\hat{S} = W\tilde{X}$

Where the equation of Newton's Iteration here is the same as in FastICA. The signal extraction is only used for initial matrix $\widehat{\boldsymbol{W}}_{\boldsymbol{0}}$ estimation. The $\widetilde{\boldsymbol{X}}$ are centralized and whitened mixed signals. Note that step 2 and 3 for initial separation matrix $\widehat{\boldsymbol{W}}_0$ estimation is marked as $f_{Initial\ Matrix}(\boldsymbol{X}, e)$ in the next.

3.2. Multi-level Newton's Iteration

As in section 3.1.4 introduced, a signal extraction is also needed in Newton's Iteration in FastICA, which can save a lot of time. However, a direct signal extraction without change of its amplitude is not suitable in this case. Meanwhile, the determination of signal extraction interval and tolerance limit for each extraction interval is also very important for a better performance.

3.2.1. Multi-Level Signal Extraction in Newton' Iteration

Multi-level Newton' iteration is Newton's Iteration with multi-level signal extraction. To embed the multi-level signal extraction into Newton's Iteration in FastICA, a further processing is required. From Eq. (6), it is easy to see, the size of the separation matrix $\boldsymbol{W}$ is dependent on the mixed signal $\boldsymbol{X}$. However, for a multi-level Newton' iteration, the size of $\boldsymbol{W}$ in each level signal extraction should be the same, otherwise the transform of matrix $\boldsymbol{W}$ between different levels of Newton's Iteration will cost a lot of time.

In FastICA, the mixed signals $\boldsymbol{X}$ is whitened. Therefore, the inverse and transpose of $\boldsymbol{X}$ are the same, which is presented in Eq. (37):

$$\boldsymbol{X}\boldsymbol{X}^{-1} = \boldsymbol{X}\boldsymbol{X}^{T} = \boldsymbol{I} \tag{37}$$

Therefore, the mixed signals $\boldsymbol{X}$ after extraction must be also whitened or approximatively whitened. The relationship of $\boldsymbol{X_e}$ and $\boldsymbol{X}$ is presented in Eq. (38)

$$\begin{aligned} {}_{e}^{i}\boldsymbol{X}' = \sqrt{e} \cdot {}_{e}^{i}\boldsymbol{X} &= \sqrt{e} \cdot \begin{bmatrix} x_{e_1} & x_{e_2} & \cdots & x_{e_{\lfloor n/e \rfloor}} \end{bmatrix} \\ &= \sqrt{e} \cdot \begin{bmatrix} x_i & x_{i+e} & x_{i+2e} & \cdots & x_{i+e\lfloor n/e-1 \rfloor} \end{bmatrix} \end{aligned} \tag{38}$$

Where and $\boldsymbol{i} = \boldsymbol{1}, \boldsymbol{2}, \ldots, \boldsymbol{e}$. Usually, the value of $\boldsymbol{i}$ is 1. The value of ${}_{e}^{i}\boldsymbol{X}' {}_{e}^{i}\boldsymbol{X}'^{\boldsymbol{T}}$ should be approximate to $\boldsymbol{I}$ when the values of different time slots in matrix $\boldsymbol{X}$ are independent. Therefore, the special processed extracted signal ${}_{e}^{i}\boldsymbol{X}'$ can be directly used in Newton's Iteration with different signal extraction interval.

3.2.2. Relationship of Signal Extraction Interval and Tolerance

For a multi-level Newton's Iteration, the determination of extraction interval and tolerance for each level is very important, and a certain ratio of data extraction interval and the tolerance in each level of Newton's Iteration, we can further reduce the variables that need to be optimized next. In this section, the relationship of extraction interval and the tolerance limit in Newton's Iteration will be talked.

According to the FastICA algorithm, the separation matrix $\boldsymbol{W}$ is decorrelated, so we have

$$\boldsymbol{W}\boldsymbol{W}^{-1} = \boldsymbol{W}\boldsymbol{W}^{T} = \boldsymbol{I} \tag{39}$$

Then together with the Eq. (3), Eq. (37) and Eq. (39), the matrixes $\boldsymbol{W}$ and $\boldsymbol{W}^{-1}$ can be represented as a formula containing a matrix $\boldsymbol{X}$ and a matrix $\widehat{\boldsymbol{S}}$ in Eq. (40) and Eq. (41)

$$\boldsymbol{W} = \widehat{\boldsymbol{S}}\boldsymbol{X}^{-1} = \widehat{\boldsymbol{S}}\boldsymbol{X}^{T} \tag{40}$$

$$\boldsymbol{W}^{-1} = \boldsymbol{W}^{\boldsymbol{T}} = \boldsymbol{X}\widehat{\boldsymbol{S}}^{-1} = \boldsymbol{X}\widehat{\boldsymbol{S}}^{\boldsymbol{T}} \tag{41}$$

Then, from the Eq. (7), the ideal result for Newton's Iteration is when $Tol = 0$, then this equation can be represented as

$$\boldsymbol{W}^{+}\boldsymbol{W}^{T} = \boldsymbol{I} \tag{42}$$

Hence,

$$\boldsymbol{W}^{+}\boldsymbol{W}^{T} = \boldsymbol{W}^{-1}\boldsymbol{W}^{+} = \boldsymbol{X}\widehat{\boldsymbol{S}}^{-1}\widehat{\boldsymbol{S}}^{+}\boldsymbol{X}^{+^{-1}} = \boldsymbol{X}\widehat{\boldsymbol{S}}^{-1}\widehat{\boldsymbol{S}}^{+}\boldsymbol{X}^{+^{\boldsymbol{T}}} \tag{43}$$

Because it is the ideal result so that the separated signal $\widehat{\boldsymbol{S}}^{+}$ and $\widehat{\boldsymbol{S}}$ are the same, and it is easy to get that

$$\boldsymbol{W}^{+}\boldsymbol{W}^{T} = \boldsymbol{X}\boldsymbol{X}^{+^{\boldsymbol{T}}} = \boldsymbol{I} \tag{44}$$

Together with Eq. (7) and Eq. (44), the tolerance of the Newton's Iteration can also be presented in Eq. (45):

$$Tol_{\boldsymbol{X}\boldsymbol{X}^{+^{T}}} = Max\left\{\left|\boldsymbol{X}\boldsymbol{X}^{+^{\boldsymbol{T}}} - \boldsymbol{I}\right|\right\} = Tol \tag{45}$$

Together with Eq. (38) and Eq. (45), we can get the relationship of $Tol_{\boldsymbol{X}\boldsymbol{X}^{+^{T}}}$ and $Tol_{{}_{e}^{i}\boldsymbol{X}'\,{}_{e}^{i}\boldsymbol{X}'^{+^{T}}}$ in Eq.(46)

$$\begin{aligned}
Tol_{\boldsymbol{X}\boldsymbol{X}^{+^{T}}} &= Max\left\{\left|[{}_{e}^{1}\boldsymbol{X}, {}_{e}^{2}\boldsymbol{X}, \dots, {}_{e}^{e}\boldsymbol{X}][{}_{e}^{1}\boldsymbol{X}, {}_{e}^{2}\boldsymbol{X}, \dots, {}_{e}^{e}\boldsymbol{X}]^{+^{\boldsymbol{T}}} - \boldsymbol{I}\right|\right\} \\
&= Max\left\{\left|\sum_{\boldsymbol{i}=\boldsymbol{1}}^{\boldsymbol{e}}\left({}_{e}^{i}\boldsymbol{X}\,{}_{e}^{i}\boldsymbol{X}^{+^{T}} - \frac{\boldsymbol{I}}{e}\right)\right|\right\} \\
&= Max\left\{\left|\frac{1}{e}\sum_{\boldsymbol{i}=\boldsymbol{1}}^{\boldsymbol{e}}\left({}_{e}^{i}\boldsymbol{X}'\,{}_{e}^{i}\boldsymbol{X}'^{+^{T}} - \boldsymbol{I}\right)\right|\right\} \\
&= Max\left\{\left|\frac{1}{e}\sum_{\boldsymbol{i}=\boldsymbol{1}}^{\boldsymbol{e}} Tol_{{}_{e}^{i}\boldsymbol{X}'\,{}_{e}^{i}\boldsymbol{X}'^{+^{T}}}\right|\right\}
\end{aligned} \tag{46}$$

Where when the matrix $\boldsymbol{X}$ is 1-D array, the tolerance of Newton's Iteration can be approximated as variance, then according to the Eq. (46) the value of $Tol_{\boldsymbol{X}\boldsymbol{X}^{+^{T}}}$ can be represented in Eq. (47):

$$\begin{gathered}
Tol_{\boldsymbol{X}\boldsymbol{X}^{+^{T}}} \approx \frac{1}{\sqrt{e}} Tol_{{}_{e}^{i}\boldsymbol{X}'\,{}_{e}^{i}\boldsymbol{X}'^{+^{T}}} \\
Tol_{{}_{e}^{i}\boldsymbol{X}'\,{}_{e}^{i}\boldsymbol{X}'^{+^{T}}} \approx \sqrt{e} \cdot Tol
\end{gathered} \tag{47}$$

Hence, the limit of tolerance in each level of Newton's Iteration is

$$Lim_{{}_{e}^{i}\boldsymbol{X}'\,{}_{e}^{i}\boldsymbol{X}'^{+^{T}}} = \sqrt{e} \cdot Lim \tag{48}$$

Where the tolerance limit and the extraction interval in each level of Newton's Iteration is determined, which can greatly reduce the calculation of optimization of parameters in the next.

3.2.3. Improved FastICA with Multi-level Newton' iteration

The multi-level Newton's Iteration algorithm is also based on FastICA. According to Eq. (8) and Eq. (48), each level of Newton's Iteration in FastICA is presented in Eq. (49):

$$\boldsymbol{W}^{+}, Tol = f_{Newton's\,iteration}\left(\tfrac{1}{e}\widetilde{\boldsymbol{X}}', \boldsymbol{W}, \sqrt{e}\cdot Lim\right) \tag{49}$$

The matrix $\boldsymbol{W}$ here is the separation matrix got from the last level of Newton's Iteration. Therefore, the basic form of the improved FastICA with multi-level Newtons iteration algorithm is as follows:

1. Centralization
2. Whiten, $\widetilde{\boldsymbol{X}} = \boldsymbol{V}\boldsymbol{X}$
3. Choose an initial (e.g. random) $\widehat{\boldsymbol{W}}_0$ and $\boldsymbol{W} = \widehat{\boldsymbol{W}}_0$
4. Choose an initial extraction interval $\mathrm{e} = e_0$
5. Let $\boldsymbol{W}^{+}, Tol = f_{Newton's\,iteration}\left(\tfrac{1}{e}\widetilde{\boldsymbol{X}}', \boldsymbol{W}, \sqrt{e}\cdot Lim\right)$
6. Let $Tol' = Tol \times \sqrt{e}$
7. Let $\mathrm{e} = \mathrm{e} - 1$
8. If $\mathrm{e} > 0$ and $\frac{Tol\prime}{\sqrt{e}} \geq \sqrt{e}\cdot Lim$, go back to 5, if $\mathrm{e} > 0$ and $Tol < \sqrt{e}\cdot Lim$, go back to 6, otherwise go to 5
8. Get the separation matrix $\boldsymbol{W}$ and let $\widehat{\boldsymbol{S}} = \boldsymbol{W}\widetilde{\boldsymbol{X}}$

Where the initial extraction interval e_0 is also depend on the character of the original signals. The determination of the initial extraction interval will be introduced in the next. The step 1-8 can be presented as a specially defined function in Eq. (50):

$$\widehat{\boldsymbol{S}} = f_{Multi-Level\,ICA}(\boldsymbol{X}, e_0, Lim) \tag{50}$$

Here the value of the initial extraction interval is still not determined, in the next section, a method will be introduced to find the suitable initial extraction interval.

3.2.4. Determination of Initial Extraction Interval

The structure of the determination of the initial Extraction Interval is shown in Fig. 5. Where for each interval e_0, the function $f_{Multi-Level\,FastICA}(\boldsymbol{X}, e_0, Lim)$ will run for $N_{samples}$ times with different mixing matrixes.

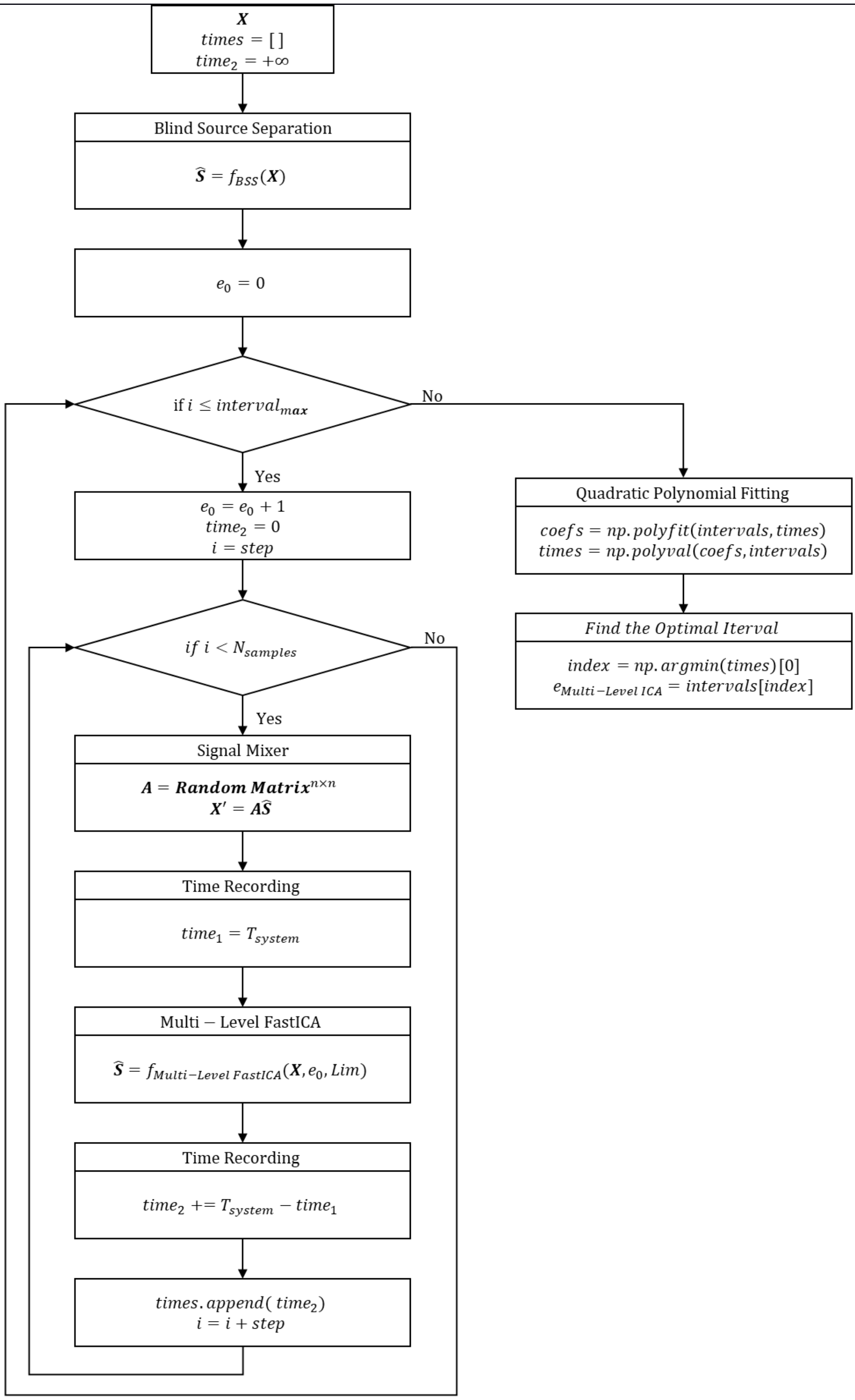

Fig. 5 Determination of Initial Extraction Interval

Where $interval_{max}$ is preferably bigger than the optimal extraction interval, so that we can get the best result, in this thesis $interval_{max} = 100$. Because of the instability of time consumption of $f_{Multi-Level\ FastICA}(\boldsymbol{X}, e_0, Lim)$, a quadratic polynomial fitting is used to let the curve more smoothly. Therefore, the determined initial extraction interval $e_{Multi-Level\ ICA}$ is the position of the minimum of the curve. Because the performance of the Multi-Level ICA is robust, the value of $e_{Multi-Level\ ICA}$ doesn't need to be very precious so $step$ is used to speed up the optimization, in this thesis $step = 5$. With this method, we can get an initial extraction interval roughly. The value of $N_{samples}$ can be chosen according to the practical situation. Usually, the value of $N_{samples}$ is between 20 to 50, in this thesis $N_{samples} = 30$.

3.3. The Ultra-FastICA Algorithm

In this Section, an Ultra-Fast ICA algorithm is introduced, which is the algorithm combined with Section 3.1 and 3.2. The basic form of the Ultra-FastICA is as follows, it uses an estimated separation matrix as an initial separation matrix in Newton's Iteration and multi-level data extraction for each iteration to reduce the computation of FastICA.

1. Centralization
2. Initial separation matrix estimation, $\boldsymbol{W}_0 = f_{Initial\ Matrix}(\boldsymbol{X}, e_{Matrix\ Estimation})$
3. Multi-Level FastICA, $\widehat{\boldsymbol{S}} = f_{Multi-Level\ ICA}(\boldsymbol{X}, e_{Multi-Level\ ICA}, Lim)$

Where the values of $e_{Matrix\ Estimation}$ and $e_{Multi-Level\ ICA}$ can be determined according to Section 3.1.4 and Section 3.2.4. The value of $e_{Matrix\ Estimation}$ and $e_{Multi-Level\ ICA}$ can affect greatly the performance of Ultra-FastICA. So, in practice, it is important to select a robust value of two extraction intervals.

4. Evaluation

In this Section, the new proposed BSS algorithm are compared with each other. In Section 4.1, the structure of testing data is descripted. In Section 4.2 is the determination of the important parameters for the following improved FastICA methods. In Section 4.3 to 4.5, there are the simulated results compare to the FastICA, it shows the relationship between performance and mixing matrix or the number of the sources. In Section 4.6, it covers the comparison of 3 proposed BSS algorithm with each other.

4.1. Setup and Testing Data

We picked random positive numbers following the normal distribution $\mathcal{N}(\mu = 0, 0 < \sigma^2 < 0.11)$ as the non-diagonal elements of the mixing matrix $\boldsymbol{A}$, the value of the diagonal elements is set to 1. The elements constructing the mixing matrix $\boldsymbol{A}$ are not only randomly distributed, but also covered 99.73% possible values under three standard deviations. Therefore, a metric Λ can be determined to represent the deviation of the randomly generated mixing matrix:

$$\Lambda = \frac{\sigma}{0.33} \times 100\% \tag{51}$$

$n = 4,8,16$ audio signals are randomly selected from 63 human voices, as the original sources $\boldsymbol{S}$. Each of them is a 4-second audio with 16000Hz sampling rate. The length of each source is $m = 64000$. The mixtures $\boldsymbol{X}$ are generated by Eq. (2), as input for following blind source separation algorithms. In the following tests, for each given Λ, 30 samples were tested. A confidence interval with 95% confidence level was also given. The environment for test is Intel-CPU I5 6200U 2.8GHz 8G RAM.

4.2. Extraction Intervals Determination

Before we use the following algorithms, the initial extraction interval for $\widehat{\boldsymbol{W}}_{\boldsymbol{0}}$ and the maximum extraction interval for Multi-Level ICA should be determined. Here the maximum tested extraction interval is 100, i.e. $interval_{max} = 100$.

4.2.1. Determination of Extraction Interval for Initial $\widehat{\boldsymbol{W}}_{\boldsymbol{0}}$ Estimation

In the Table 1 are the results of determination of extraction interval for initial $\widehat{\boldsymbol{W}}_{\boldsymbol{0}}$, the vales are the average of $N_{B_l,min}$. Note that $N_{minimum} = 20$, $N_{samples} = 30$ here. Therefore, we can get the extraction interval for initial $\widehat{\boldsymbol{W}}_{\boldsymbol{0}}$ is 80,35,20 when $n = 4,8,16$.

n / interval	n=4	n=8	n=16
5	345.77	178.97	95.70
10	185.27	82.00	36.77
15	151.90	58.90	26.27
20	138.53	40.13	20.23
25	88.03	29.97	17.17
30	97.33	25.57	15.10
35	49.47	23.07	13.37
40	50.63	18.80	8.83
45	40.83	18.70	8.70
50	45.43	14.13	8.23
55	38.97	17.37	8.27
60	40.13	14.43	6.40
65	30.70	13.07	5.70
70	27.70	10.60	3.97
75	26.87	8.87	4.47
80	27.17	11.03	5.37
85	19.10	8.53	4.43
90	19.90	9.50	4.63
95	19.90	8.57	3.87
100	18.43	6.77	5.20

Table 1 Results of Determination of Extraction Interval for Initial $\widehat{\boldsymbol{W}}_{\boldsymbol{0}}$

4.2.2. Determination of Maximum Extraction Interval for Multi-Level ICA

In the Table 1 are the results of determination of maximum extraction interval for Multi-Level ICA, the values are the totally time consumption of 30 samples of each extraction interval.

n interval	n=4	n=8	n=16
5	0.74	2.38	8.46
10	0.52	1.58	4.91
15	0.53	1.41	4.32
20	0.53	1.41	4.13
25	0.74	1.46	4.15
30	0.71	1.40	4.01
35	0.58	1.33	3.66
40	0.62	1.33	3.24
45	0.54	1.39	3.58
50	0.51	1.23	3.49
55	0.54	1.22	3.36
60	0.49	1.20	3.35
65	0.52	1.12	3.39
70	0.50	1.05	3.09
75	0.45	1.08	3.13
80	0.45	1.06	3.03
85	0.44	1.03	3.25
90	0.43	1.09	3.04
95	0.48	1.08	3.11
100	0.44	1.17	3.80

Table 2 Results of Determination of Maximum Extraction Interval for Multi-Level ICA

Then these results should be quadratic regressed, the regressed, the regressed curves are shown in Fig. 6, Fig. 7 and Fig. 8. Then, we can find the minimum value of these curves, i.e. 100,80,70 for $n = 4,8,16$.

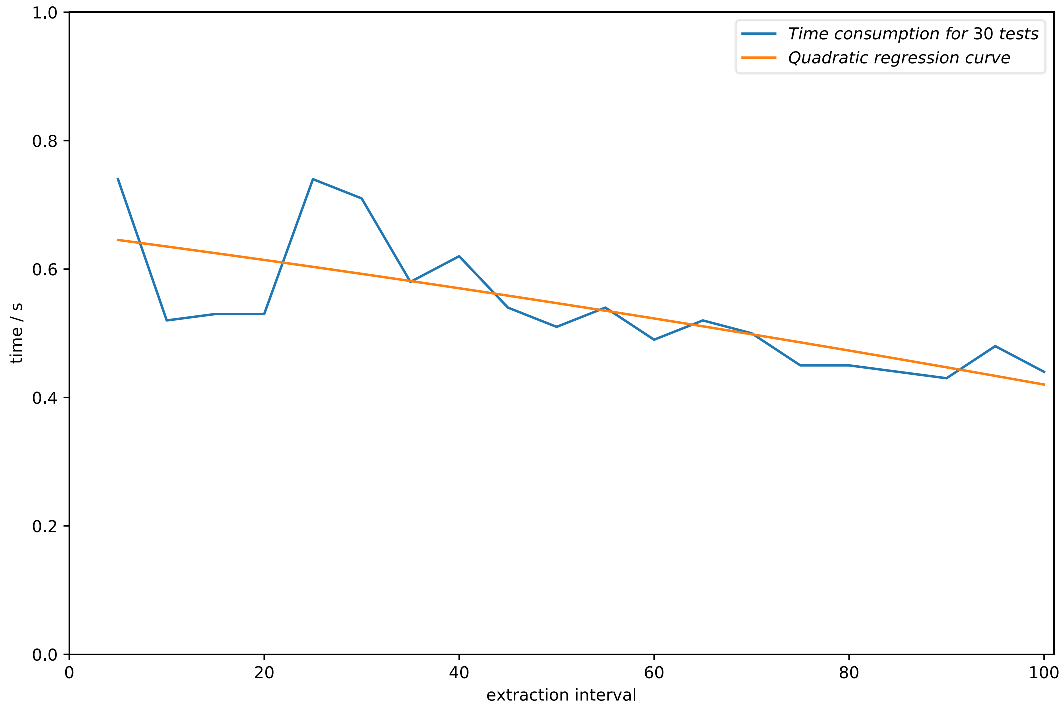

Fig 6 Time Consumptions for Each Extraction Interval when $n = 4$

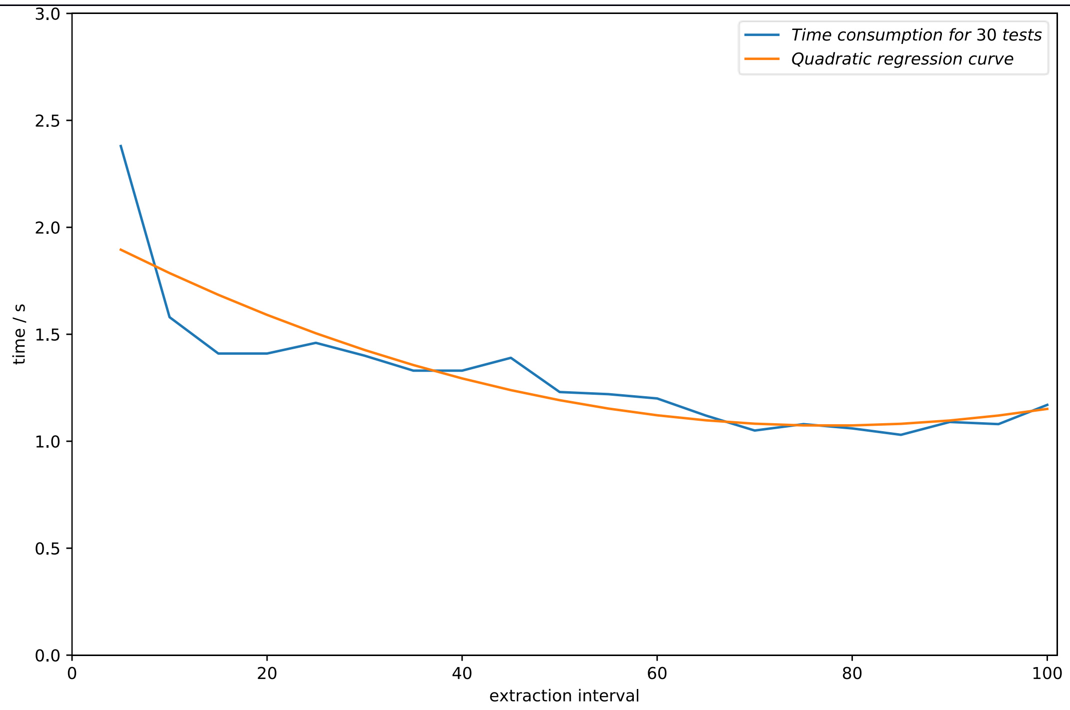

Fig 7 Time Consumptions for Each Extraction Interval when $n = 8$

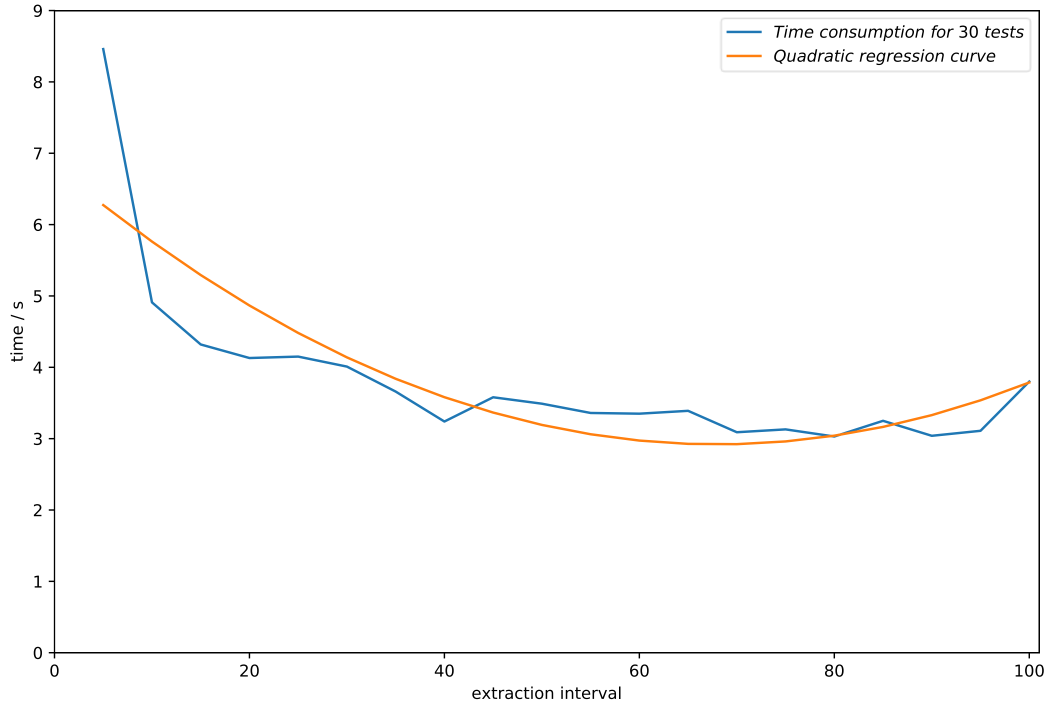

Fig 8 Time Consumptions for Each Extraction Interval when $n = 16$

4.3. Results of Improved FastICA with Initial Matrix Estimation

4.3.1. Difference from Estimated $\widehat{\boldsymbol{W}}_{\mathbf{0}}$

Cosine distance Δ in the Eq. (24) was used to indicate the similarity of two matrices (52). The cosine distance is advantageous because even if the two similar matrices are far apart by the Euclidean distance, they could still have a smaller orientation. Using different mixing matrices $\boldsymbol{A}$, the initial estimated matrices $\widehat{\boldsymbol{W}}_{\mathbf{0}}$ and the random initial matrices $\boldsymbol{W}_{\mathbf{0}}$ of FastICA are built, their cosine distance [9] Δ to the ideal separation matrices $\widehat{\boldsymbol{W}}_{\mathbf{0}} = \boldsymbol{A}^{-1}$ are shown in Fig. 9.

$$\Delta = 1 - \frac{\boldsymbol{W}_{\boldsymbol{initial}} \cdot \boldsymbol{W}}{\|\boldsymbol{W}_{\boldsymbol{initial}}\|_2 \cdot \|\boldsymbol{W}\|_2} \tag{52}$$

It can be clearly seen that by a given number of sources, the distance between the estimated initial separation matrix $\widehat{\boldsymbol{W}}_{\mathbf{0}}$ and the ideal separation matrix $\boldsymbol{W}$ is much smaller than the distance between the random matrix $\boldsymbol{W}_{\mathbf{0}}$ and $\boldsymbol{W}$, which, can cause smaller iteration numbers in Newton's Iteration.

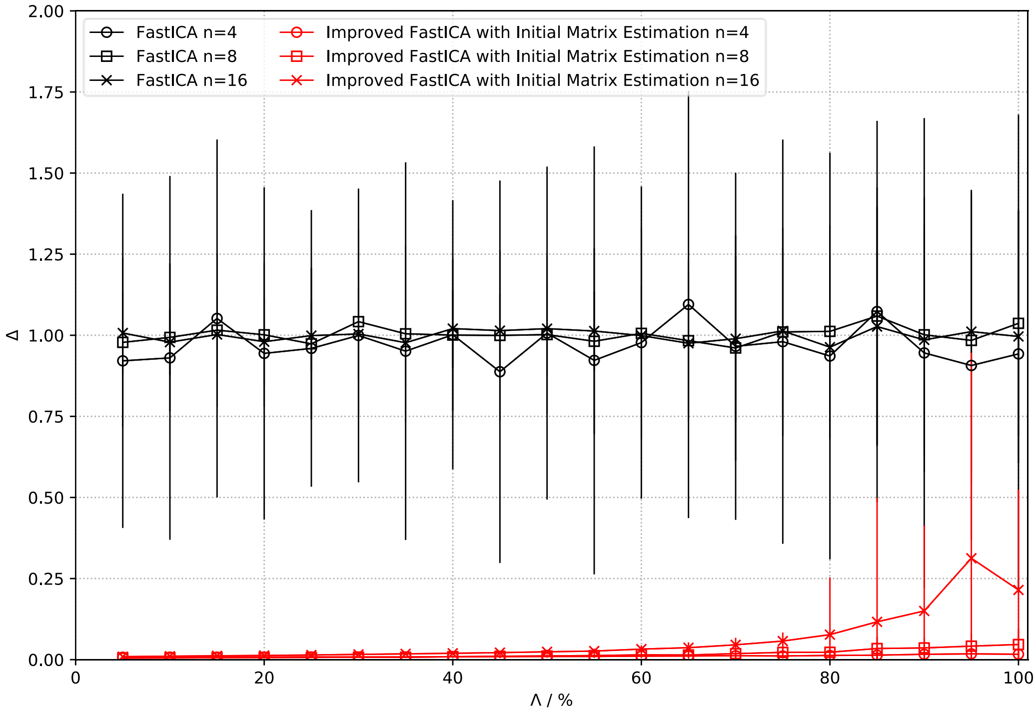

Fig. 9 Difference from Separation Matrix

4.3.2. Time Consumption of Improved FastICA with Initial Matrix Estimation

The separation speed compare to the FastICA is shown in Fig. 10. Because of the estimated initial separation matrix $\widehat{\boldsymbol{W}}_{\mathbf{0}}$, the time consumption of Improved FastICA with Initial Matrix Estimation is much less than FastICA.

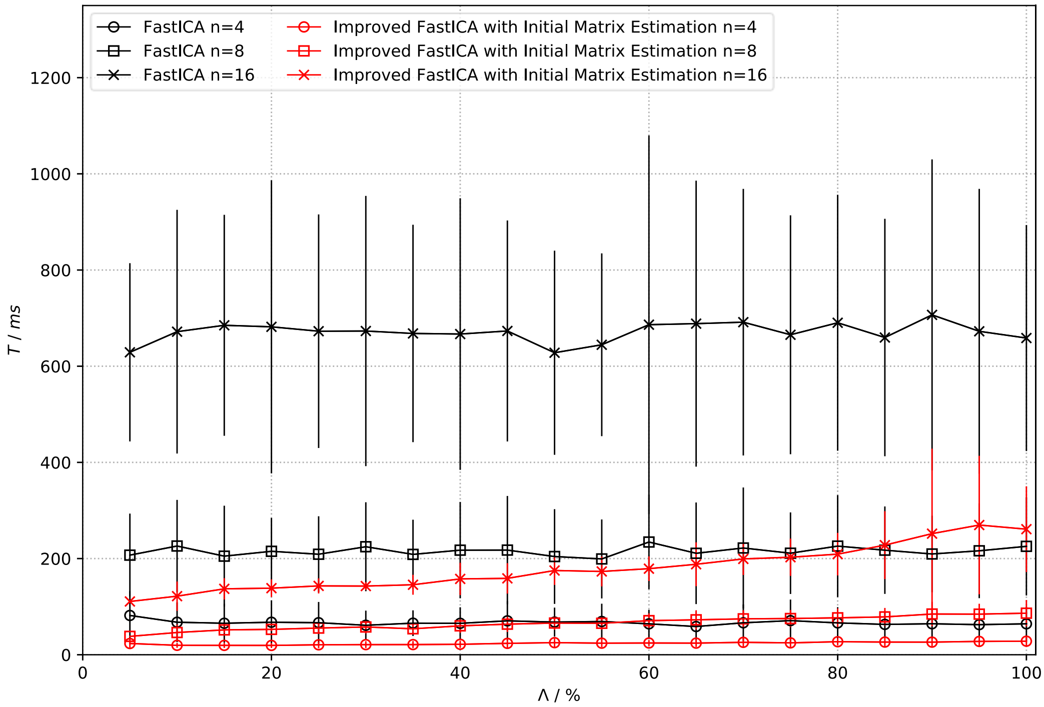

Fig. 10 Time Consumption of Improved FastICA with Initial Matrix Estimation

Fig. 11 shows the ratio R_t of the time consumption with the Improved FastICA with Initial Matrix Estimation to the original time consumption of FastICA:

$$R_t = \frac{T_{Tested\ BSS\ Algorithm}}{T_{FastICA}} \times 100\% \tag{53}$$

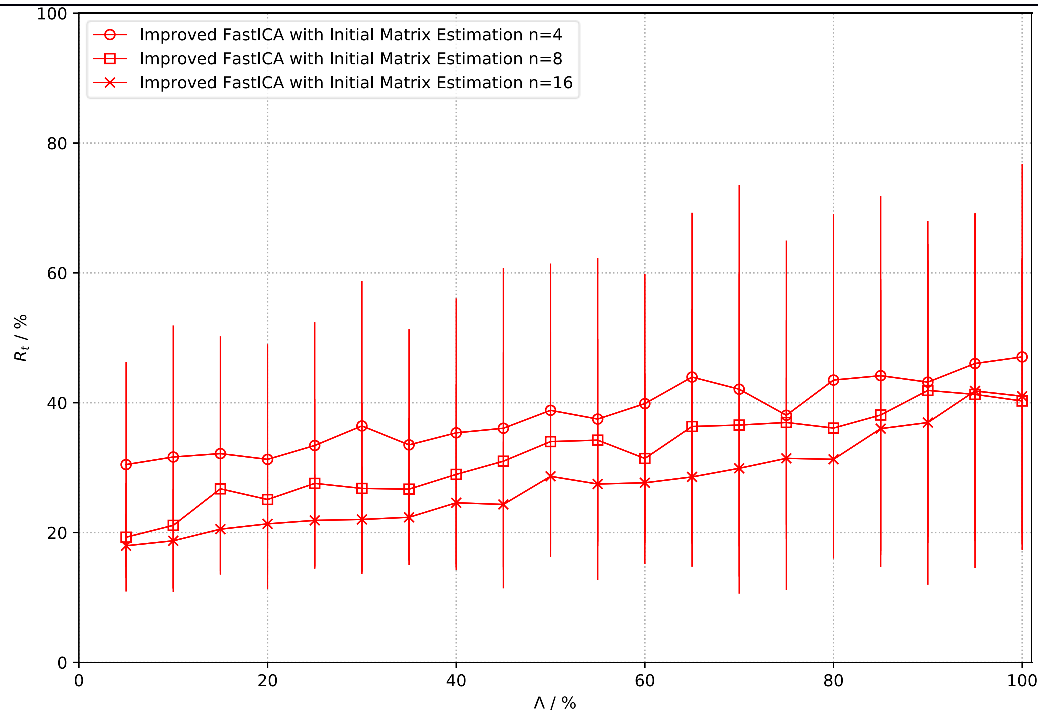

Fig. 11 Reduction of Time Consumption

It is noticeable that Improved FastICA with Initial Matrix Estimation reduces the time requirement to separate the mixtures considerably by 53% to 82% compared to FastICA. For more sources, the acceleration performance will be better. Because for more sources, the iteration number of FastICA is relative bigger, which provide a larger space for performance optimization. Furthermore, the larger the number of sources is, the more the reduction in time. However, the R_t grows as the growth of Λ, i.e. the performance of Initial Matrix Estimation depends on the mixing matrix $\boldsymbol{A}$.

4.3.3. SNR of Improved FastICA with Initial Matrix Estimation

SNR here is the ratio of the original sources $\boldsymbol{S}$ to the difference between $\boldsymbol{S}$ and the estimated sources $\widehat{\boldsymbol{S}}$, which is the indicator of the separation quality:

$$SNR = \sum_{i=1}^{n} 20log_{10} \frac{|\boldsymbol{S}_i|}{|\widehat{\boldsymbol{S}} - \boldsymbol{S}_i|} \tag{54}$$

The separation quality of Improved FastICA with Initial Matrix Estimation is shown in Fig. 12. One can notice that the simulation results of Improved FastICA with Initial Matrix Estimation and FastICA have no significant difference.

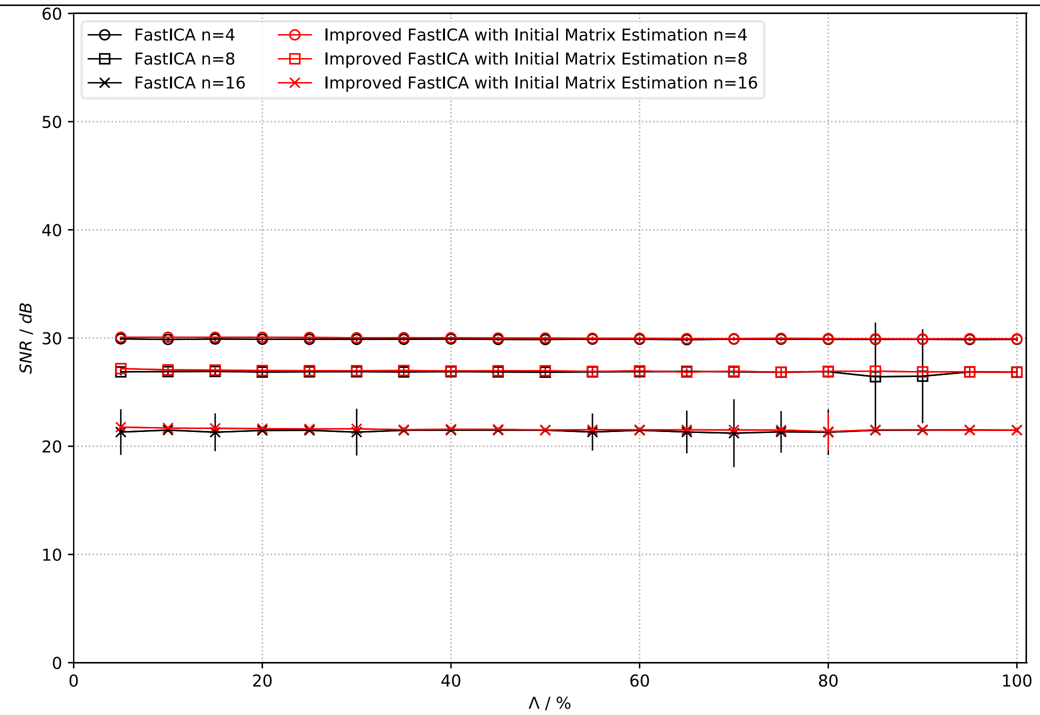

Fig. 12 Separation Quality

4.4. Results of Improved FastICA with Multi-level Newton's iteration

4.4.1. Time Consumption of Improved FastICA with Multi-level Newton' iteration

The separation speed compare to the FastICA is shown in Fig. 13. It is obviously, the time consumption of Improved FastICA with Multi-level Newton' iteration is also much less than FastICA.

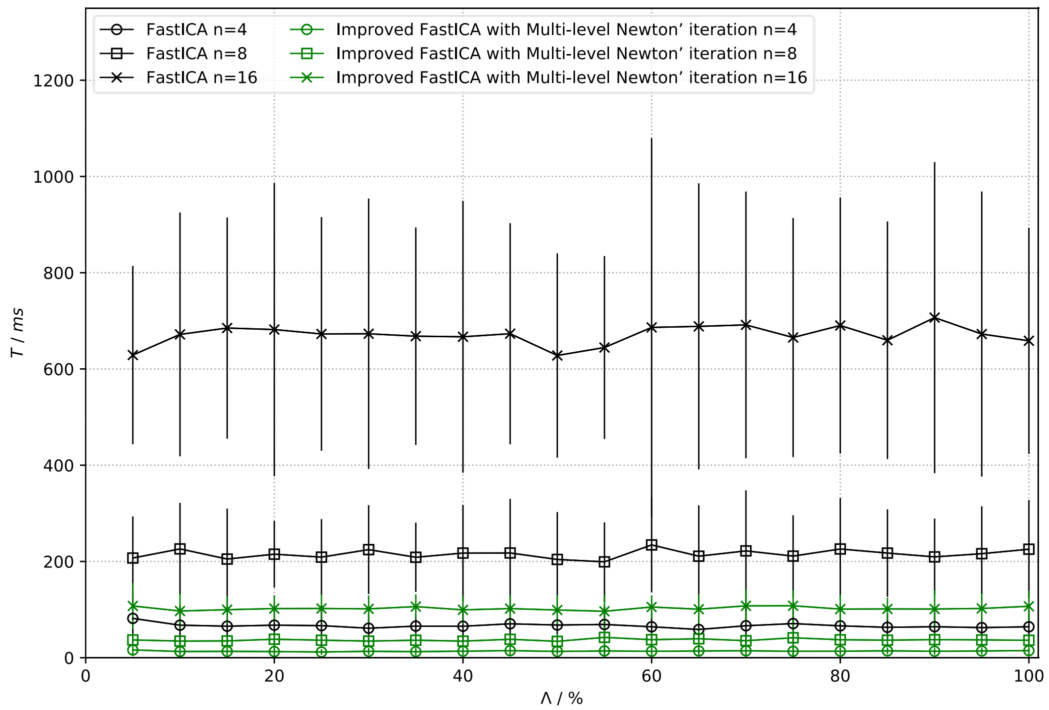

Fig. 13 Time Consumption of Improved FastICA with Multi-level Newton' Iteration

Fig. 14 shows the ratio R_t of the time consumption with Improved FastICA with Multi-level Newton' iteration to the original time consumption of FastICA:

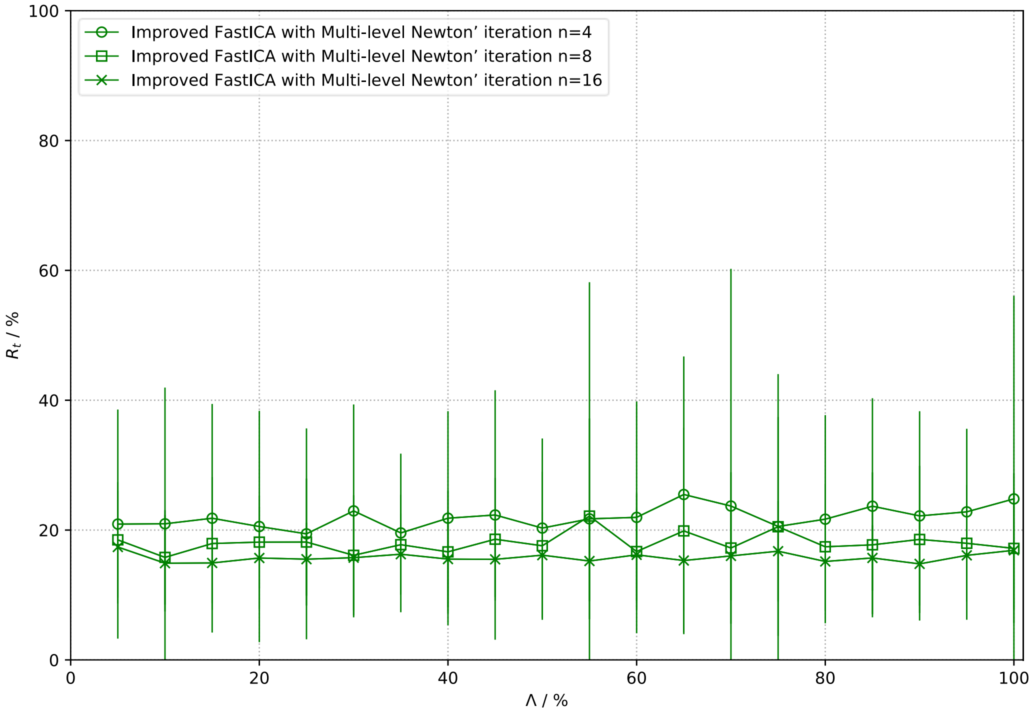

Fig. 14 Reduction of Time Consumption

It is obviously that, Improved FastICA with Multi-level Newton' iteration reduces the time of 74% to 85%. Compare to the Improved FastICA with Initial Matrix Estimation, the reduction of time with Improved FastICA with Multi-level Newton' iteration has no relationship with Λ. For more source numbers, the acceleration performance will also be better.

4.4.2. SNR of Improved FastICA with Multi-level Newton' iteration

The separation quality of Improved FastICA with Multi-level Newton' iteration is shown in Fig. 15. One can notice that the simulation results of Improved FastICA with Multi-level Newton' iteration and FastICA have no significant difference.

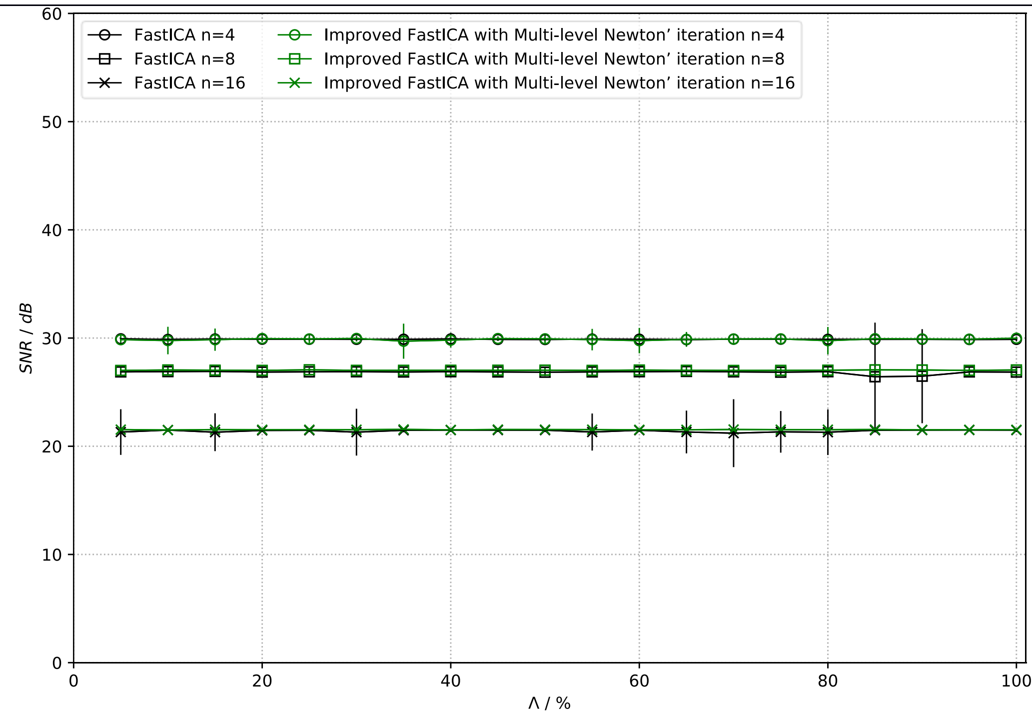

Fig. 15 Separation Quality

4.5. Results of Ultra-FastICA

4.5.1. Time Consumption of Ultra-FastICA

The separation speed compare to the FastICA is shown in Fig. 16. The time consumption of Ultra-FastICA is also much less than FastICA.

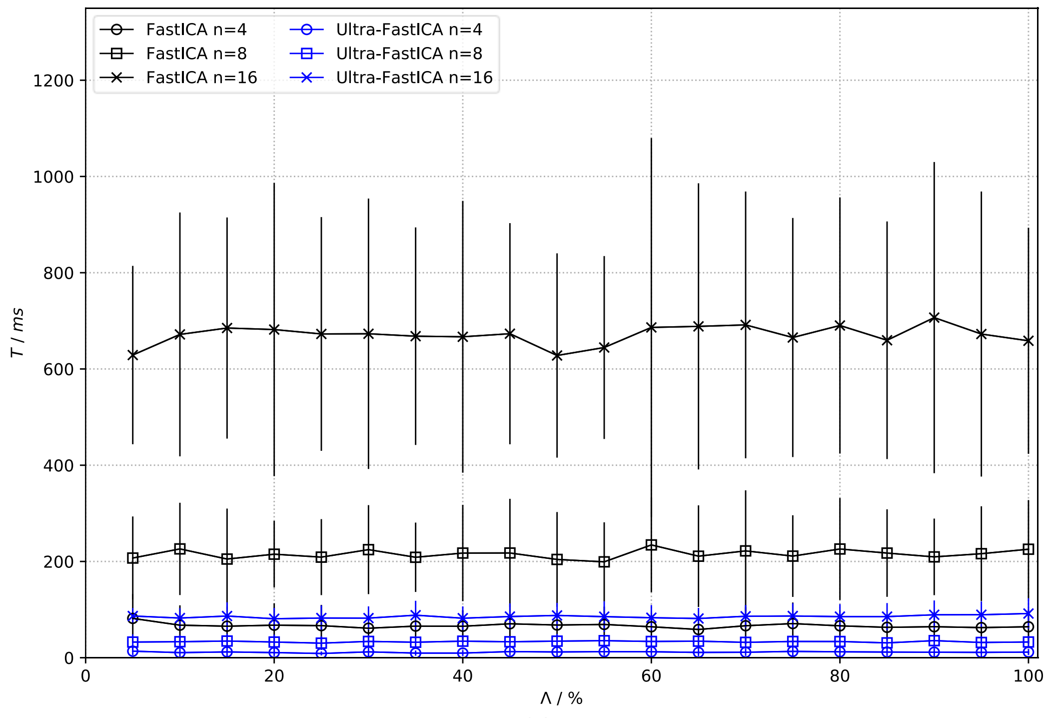

Fig. 16 Time Consumption of Ultra-FastICA

Fig. 17 shows the ratio R_t of the time consumption with Ultra-FastICA to the original time consumption of FastICA:

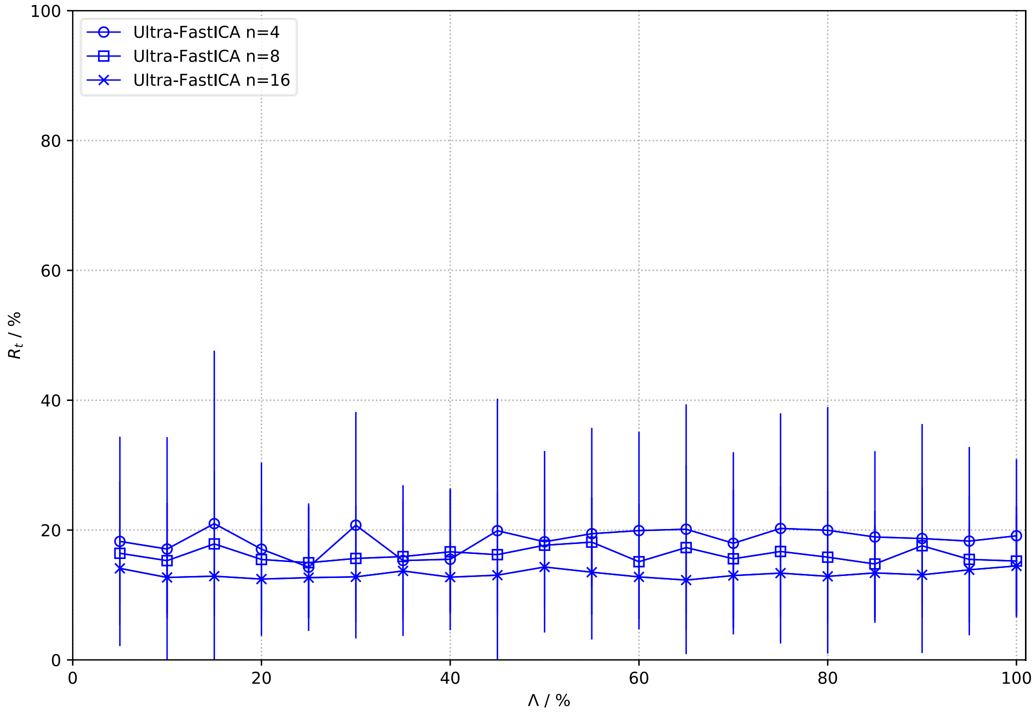

Fig. 17 Reduction of Time Consumption

It is obviously that, Ultra-FastICA reduces the time of 79% to 88%. The reduction of time with Ultra-FastICA also has no relationship with Λ. Like other two methods in Section 4.3 and Section 4.4, for more source numbers, the acceleration performance will be better.

4.5.2. SNR of Ultra-FastICA

The separation quality of Ultra-FastICA is shown in Fig. 18. One can notice that the simulation results of Ultra-FastICA and FastICA have no significant difference.

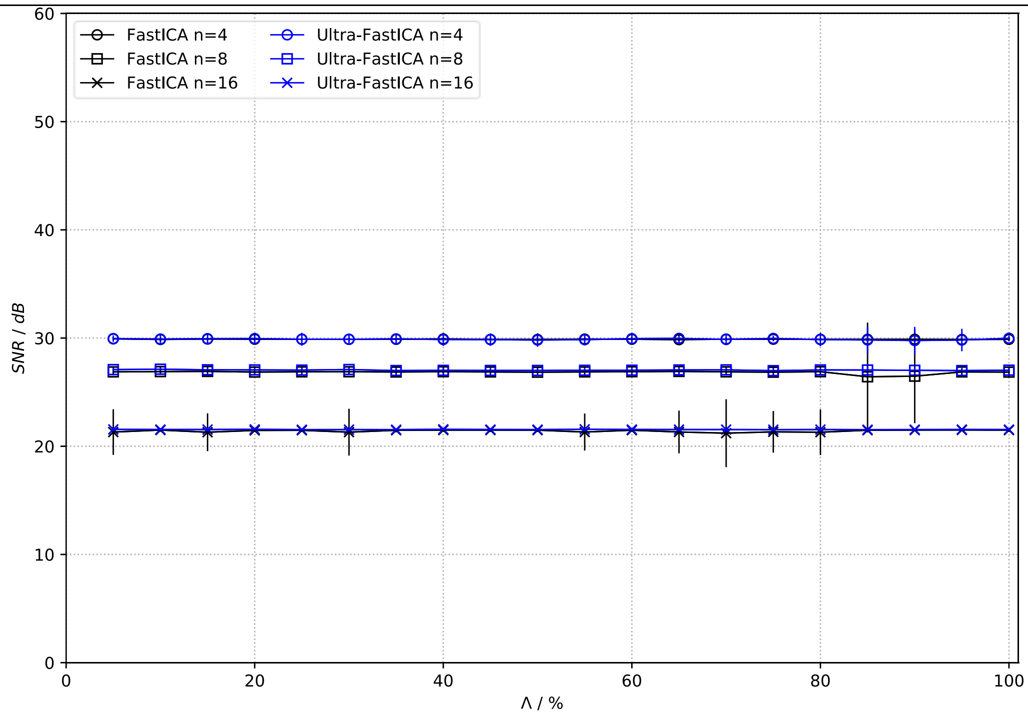

Fig. 18 Separation Quality

4.6. Comparison of All Introduced BSS Methods

In this section we present the comparison of performances form the Improved FastICA with Initial Matrix Estimation, Improved FastICA with Multi-level Newton' iteration, Ultra-FastICA and FastICA. Firstly, the ratio R_t of the time consumption is compared in Section 4.6.1, then is the separation quality in Section 4.6.2.

4.6.1. Time Consumption Comparison

Fig. 19, Fig. 20 and Fig. 21 show the comparison of R_t from 3 in this project proposed ICA algorithms. We can see that Ultra-FastICA algorithm is the fastest ICA method. The Improved FastICA with Multi-level Newton' iteration is a bit worse than Ultra-FastICA, while Improved FastICA with Initial Matrix Estimation has the worst performance and robustness.

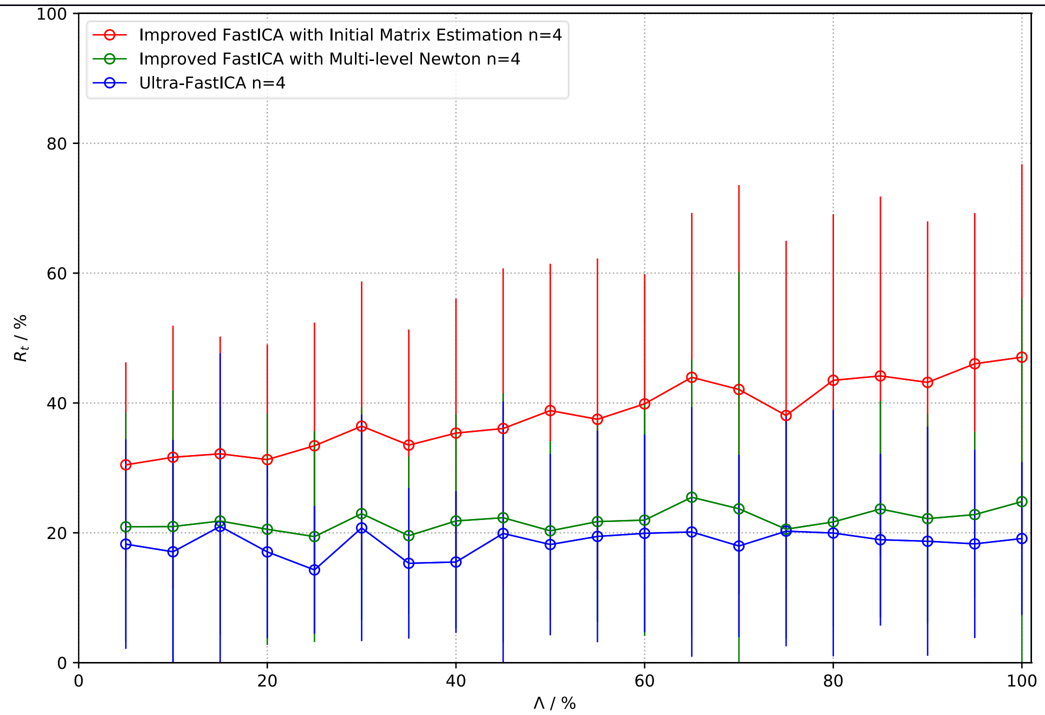

Fig. 19 Time Consumption Comparison, n=4

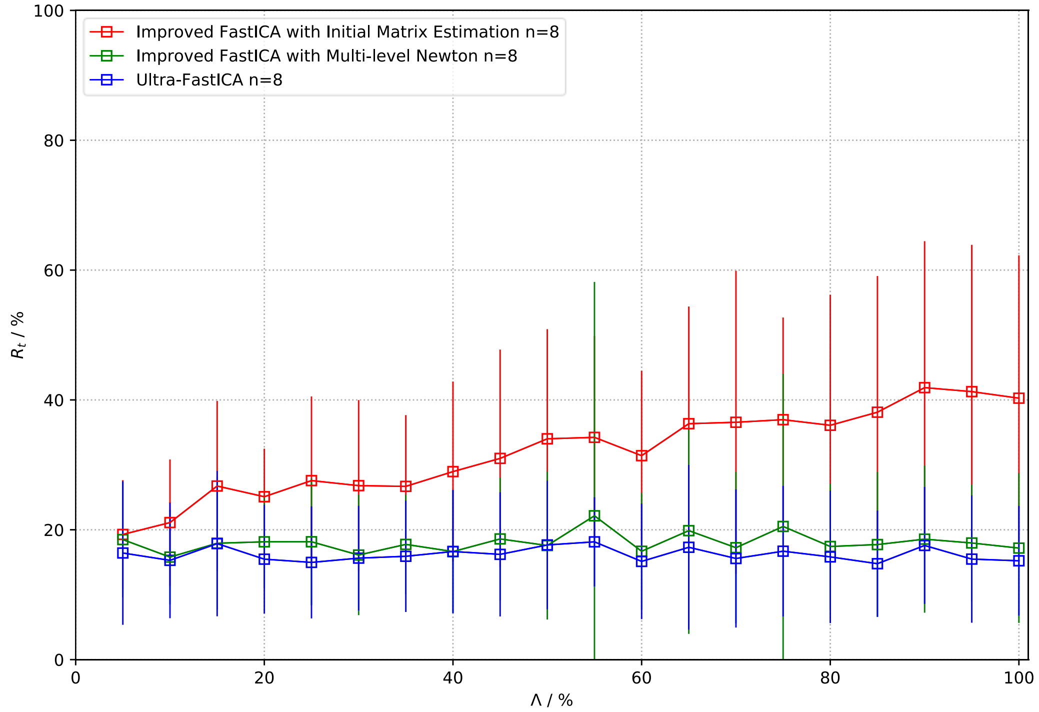

Fig. 20 Time Consumption Comparison, n=8

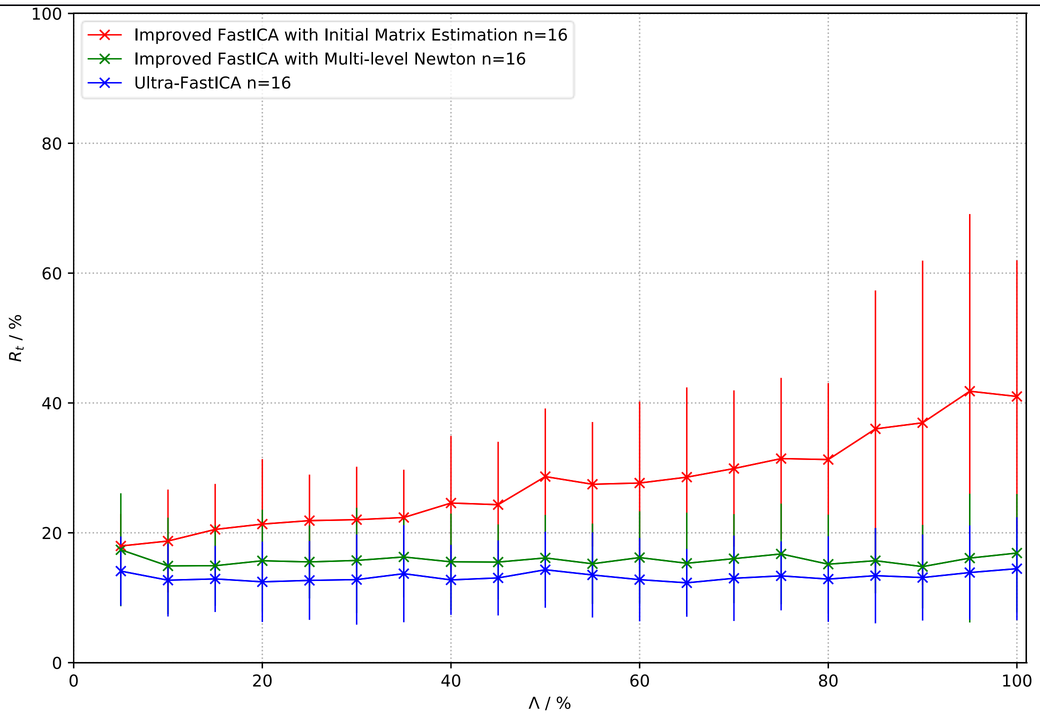

Fig. 21 Time Consumption Comparison, n=16

4.6.2. SNR of Ultra-FastICA

Fig. 22, Fig. 23 and Fig. 24 show the separation quality of three newly proposed ICA methods and FastICA. We can see the separation quality of these 4 methods are almost the same.

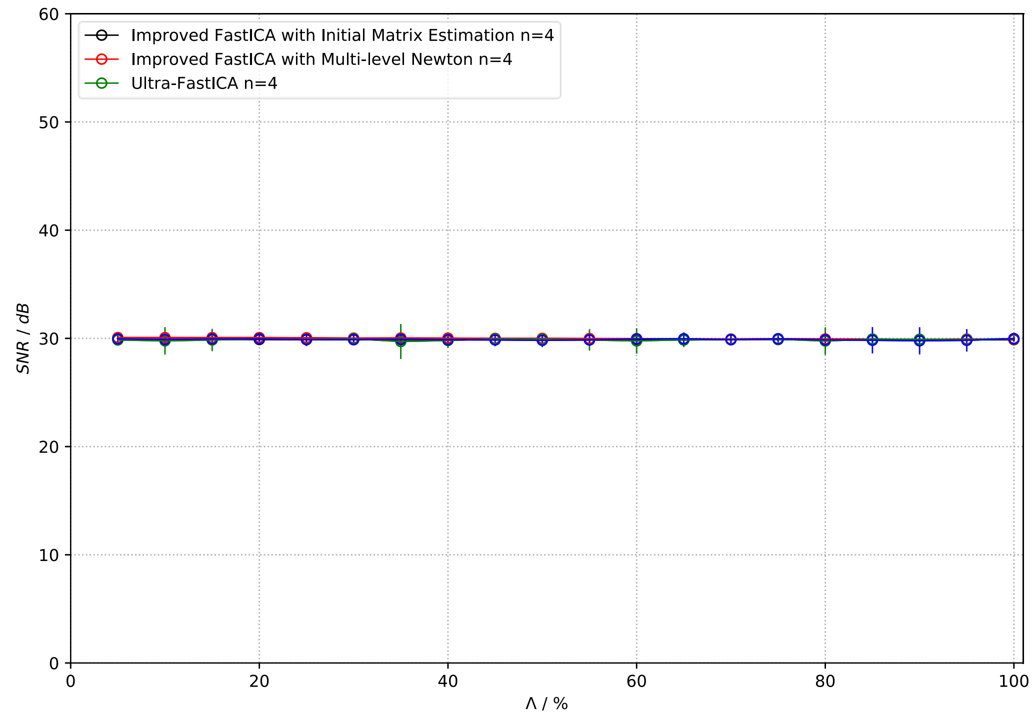

Fig. 22 Comparison of Separation Quality, n=4

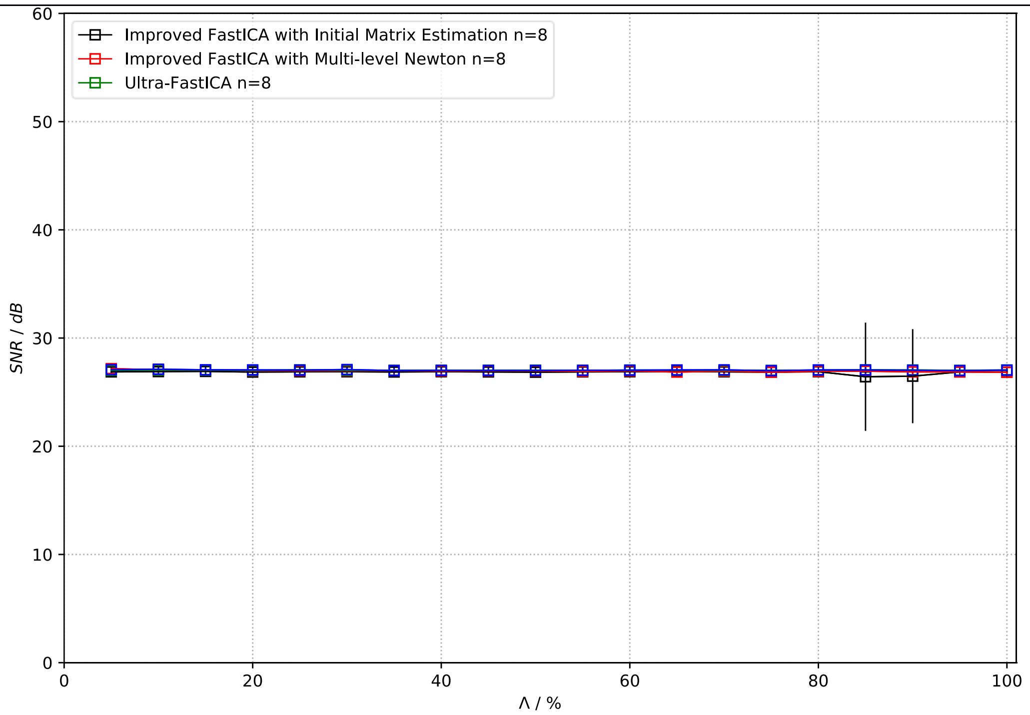

Fig. 23 Comparison of Separation Quality, n=8

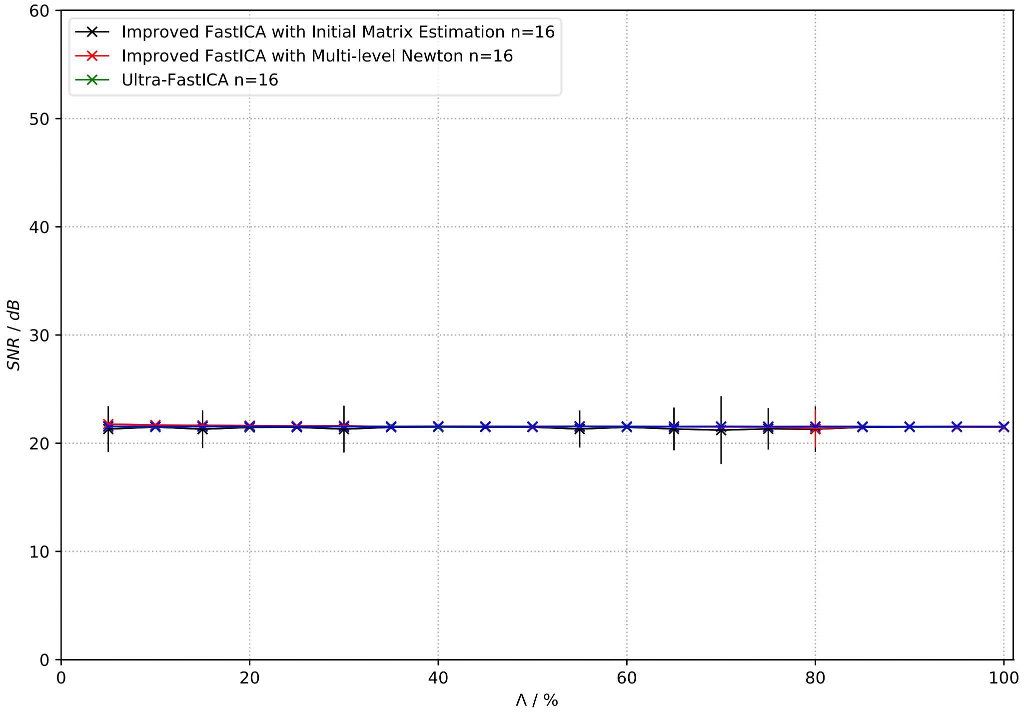

Fig. 24 Comparison of Separation Quality, n=16

5. Conclusion

In this thesis, three novel ICA algorithms are proposed and evaluated for time-sensitive applications. The first one is Improved FastICA with Initial Matrix Estimation, an mixing matrix is estimated and transformed to an initial separation matrix for Newton's Iteration in FastICA. The second one is Improved FastICA with Multi-level Newton' Iteration, a data extraction is embedded in each iteration, so that the computation is greatly reduced. The third one is Ultra-FastICA, which benefits from the two algorithms above and has a better performance.

Unlike many other improved FastICA algorithms, which accelerate FastICA by increasing iterative convergence speed in each iteration, all these in this thesis proposed novel ICA algorithms are based on reducing the number of iterations or the computation for each iteration, which are new directions to accelerate the FastICA and have a better performance.

Three proposed algorithms reduce the time of source separation by 53% to 88% compare to the FastICA, which is the currently most notable method. Meanwhile these novel ICA algorithms do not loss any quality, since the convergence of Newton's Iteration is not changed. Moreover, the Ultra-FastICA save the time of source separation by 79% to 88% compare to FastICA, which is faster and more robust compare to other two algorithms.

Because of the high performance of Ultra-FastICA, it is very suitable for time sensitive applications. Since Ultra-FastICA includes Multi-Level Newton's Iteration, the computation at the beginning does not require all the data, which make it possible to calculate while transmitting data in network computing. In addition, Ultra-FastICA can also be integrated into other Improved FastICA algorithms like eighth-order Newton's Iteration [4] because the principles they use to accelerate the FastICA are different. Therefore, Ultra-FastICA is a potential algorithm and there are still many areas for improvement in the future.

References

[1] Handbook of Blind Source Separation: Independent component analysis and applications[M]. Academic press, 2010.

[2] Jung T P, Makeig S, McKeown M J, et al. Imaging brain dynamics using independent component analysis[J]. Proceedings of the IEEE, 2001, 89(7): 1107-1122.

[3] Hyvärinen A, Oja E. Independent component analysis: algorithms and applications[J]. Neural networks, 2000, 13(4-5): 411-430.

[4] Ahmad T, Alias N, Ghanbari M, et al. Improved Fast ICA Algorithm Using Eighth-Order Newton's Method[J]. Research Journal of Applied Sciences, Engineering and Technology, 2013, 6(10): 1794-1798.

[5] Huang X. An improved FastICA algorithm for blind signal separation and its application[C]//2012 International Conference on Image Analysis and Signal Processing. IEEE, 2012: 1-4.

[6] Yuan L, Zhou Z, Yuan Y, et al. An improved FastICA method for fetal ECG extraction[J]. Computational and mathematical methods in medicine, 2018, 2018.

[7] Meng X, Yang L, Xie P, et al. An Improved FastICA Algorithm Based on Modified-M Estimate Function[J]. Circuits, Systems, and Signal Processing, 2018, 37(3): 1134-1144.

[8] Pedregosa F, Varoquaux G, Gramfort A, et al. Scikit-learn: Machine learning in Python[J]. Journal of machine learning research, 2011, 12(Oct): 2825-2830.

[9] Li B, Han L. Distance weighted cosine similarity measure for text classification[C]//International Conference on Intelligent Data Engineering and Automated Learning. Springer, Berlin, Heidelberg, 2013: 611-618.