

# 一种并行主偏度分析算法及其在盲源分离上的应用

李康宁<sup>1,2</sup>, 郭永刚<sup>1</sup>, 王肃静<sup>1</sup>, 黄石羽<sup>1</sup>

(1. 中国科学院声学研究所声场声信息国家重点实验室, 北京, 100190; 2. 中国科学院大学电子电气与通信工程学院, 北京, 100049)

**摘要:** 盲源分离(Blind source separation, BSS)是一种从混合信号中提取和恢复源信号的信号处理方法。在众多BSS算法中,主偏度分析(Principal skewness analysis, PSA)算法是近年来出现的一种以三阶统计量为目标函数的BSS算法,其运算速度快于常规的BSS算法,但因其采用了串行的计算方式,在使用中存在误差累积问题。针对这一问题,本文在PSA算法基础上进行改进,提出了一种并行PSA算法。该算法以并行计算代替串行计算,可以同时估计出各个独立成分对应的方向,避免了误差累积问题。数值仿真实验表明,与PSA算法相比,并行PSA算法既保持了计算速度,同时提高了对源信号的估计准确性。

**关键词:** 盲源分离(BSS); 偏度; 张量; 并行主偏度分析; 正交化

**中图分类号:** TN911.7      **文献标志码:** A

## A Parallel Principal Skewness Analysis Algorithm and Its Application to Blind Source Separation

LI Kangning<sup>1,2</sup>, GUO Yonggang<sup>1</sup>, WANG Sujing<sup>1</sup>, HUANG Shiyu<sup>1</sup>

(1. State Key Laboratory of Acoustics, Institute of Acoustics, Chinese Academy of Sciences, Beijing, 100190, China; 2. School of Electronic, Electrical and Communication Engineering, University of Chinese Academy of Sciences, Beijing, 100049, China)

**Abstract:** Blind source separation (BSS) algorithm is utilized in extracting and recovering source signals from mixed signals. Among many different BSS algorithms, the principal skewness analysis (PSA) is a BSS algorithm taking third-order statistics as the objective function. One of its advantages is that its calculation speed is faster than the conventional BSS algorithms. However, because of the serial calculation method, error accumulation exists in the calculation process. In order to solve this problem, this paper proposes an algorithm called parallel PSA. In this algorithm, parallel calculation is used instead of serial calculation, and the corresponding directions of each independent component can be estimated simultaneously, so the problem of error accumulation is avoided. Simulation results prove that, compared with the PSA algorithm, the parallel PSA algorithm not only maintains the fast calculation speed, but also improves the estimation accuracy of each source signal.

**Key words:** blind source separation (BSS); skewness; tensor; parallel principal skewness analysis; orthogonalization

## 引言

盲源分离(Bind source separation, BSS)是一种在源信号和传输信道未知的情况下,仅根据观测到的信号估计出源信号的信号处理方法,主要用于提取和恢复混合信号中的源信号。如图1所示,BSS的最初研究动机之一是解决“鸡尾酒会问题”,如今BSS已广泛应用于无线通信、图像处理、语音信号处理、生物医学等领域。一般而言,BSS与独立成分分析(Independent component analysis, ICA)是等价的:BSS中的各个源信号是相互独立的独立成分。

自20世纪80年代以来出现了大量的BSS算法,如快速ICA(Fast independent component analysis, FastICA)<sup>[1]</sup>、特征矩阵的联合近似对角化(Joint approximate diagonalization of eigen-matrices, JADE)<sup>[2]</sup>、二阶盲辨识(Second order blind identification, SOBI)<sup>[3]</sup>等。多数BSS算法属于依据一定的理论构造目标函数并进行优化的无监督学习问题<sup>[4]</sup>,高阶统计量、互信息、负熵<sup>[5]</sup>等均可以作为目标函数<sup>[6]</sup>。

主偏度分析(Principal skewness analysis, PSA)<sup>[7]</sup>算法是近年来出现的一种以三阶统计量——峭度为目标函数的BSS算法,该方法应用于高光谱遥感图像的特征提取,即从高光谱遥感图像中提取出不同波段的图像,这一问题满足BSS的模型。PSA算法结合了张量计算和不动点迭代,计算速度相比常规的BSS算法有较大提升。

然而PSA算法需要逐次估计混合信号中的各个源信号,在这种串行的计算过程中,难以避免地会存在误差累积问题(下文对该问题进行了详细说明)。本文针对这个问题,对现有的PSA算法进行改进,提出了一种并行PSA算法。仿真结果表明,在保证计算速度的同时,并行PSA算法避免了现有PSA算法中出现的误差累积问题,提高了对源信号的估计准确性。

## 1 BSS模型及求解方法

BSS模型可以分为瞬态混合模型和卷积混合模型,考虑到瞬态混合模型是卷积混合模型的一种特殊情况,用于求解瞬态混合模型的BSS算法可以推广到卷积混合模型中,本文仅对瞬态混合模型进行分析。

### 1.1 瞬态混合BSS模型

瞬态混合BSS模型如下

$$\boldsymbol{x}(t) = \boldsymbol{A}\boldsymbol{s}(t) + \boldsymbol{n}(t) = \sum_{i=1}^M \boldsymbol{a}_i s_i(t) + \boldsymbol{n}(t) \quad (1)$$

式中, $\boldsymbol{x}(t) = [x_1(t), x_2(t), \dots, x_M(t)]^T \in \mathbf{R}^{M \times 1}$ 代表 $M$ 个传感器接收到的信号,由各个源信号以不同方式混合而成; $\boldsymbol{s}(t) = [s_1(t), s_2(t), \dots, s_M(t)]^T \in \mathbf{R}^{M \times 1}$ 代表 $M$ 个源信号,各个源信号之间是相互独立的; $\boldsymbol{n}(t) = [n_1(t), n_2(t), \dots, n_M(t)]^T \in \mathbf{R}^{M \times 1}$ 代表传感器接收到的噪声,各个噪声 $n_i(t)$ ( $i = 1, 2, \dots, M$ )相互独立,且与源信号 $s_k(t)$ ( $k = 1, 2, \dots, M$ )独立; $\boldsymbol{A} \in \mathbf{R}^{M \times M}$ 代表源信号间的混合矩阵, $\boldsymbol{a}_i$ 代表混合矩阵的第 $i$ 列。

如图2所示,BSS系统的任务是在未知源信号 $\boldsymbol{s}(t)$ 和混合矩阵 $\boldsymbol{A}$ 的情况下,通过对 $\boldsymbol{x}(t)$ 进行处理,得到分离矩阵 $\boldsymbol{W}$ ,进而得到对源信号 $\boldsymbol{s}(t)$ 的估计 $\boldsymbol{y}(t) = [y_1(t), y_2(t), \dots, y_M(t)]^T$ ,其中 $y_k(t)$ ( $k =$

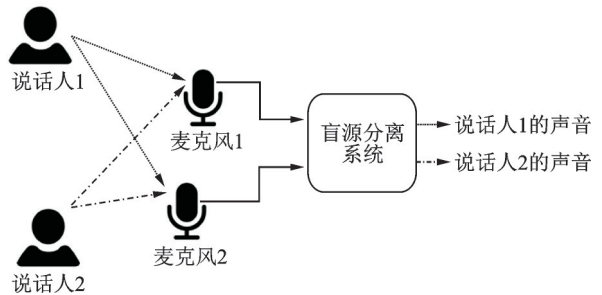


图1 鸡尾酒会问题

Fig.1 Cocktail party problem

1, 2, ..., M)代表BSS系统第k个输出信号。 $y(t)$ 的计算过程为

$$y(t) = Wx(t) = WAs(t) + Wn(t) \quad (2)$$

式中  $W = [w_1, w_2, \dots, w_M]^T$ 。定义  $Q = WA$ , 当成功实现对各个源信号的估计时满足<sup>[8]</sup>

$$Q = \Lambda P \quad (3)$$

式中,  $\Lambda$  为非奇异的对角矩阵,  $P$  为置换矩阵。

### 1.2 BSS算法的性能指标

根据BSS的不确定性<sup>[3]</sup>,  $y(t)$ 中信号的排序和幅值可能与源信号  $s(t)$  存在差异。假设BSS系统的输出信号  $y_k(t)$  是对源信号  $s_i(t)$  的估计, 通过二者的相关系数来衡量BSS算法对  $s_i(t)$  的估计准确性

$$r(y_k(t), s_i(t)) = \frac{|E[y_k(t)s_i(t)]|^2}{\sqrt{E[|y_k(t)|^2]E[|s_i(t)|^2]}} \quad (4)$$

相关系数越大代表BSS算法对源信号  $s_i(t)$  的估计准确性越高。

考虑到BSS算法的不确定性, 如式(5)构造相关系数矩阵  $R$  来衡量BSS算法的整体性能

$$R = \begin{bmatrix} r(y_1(t), s_1(t)) & r(y_1(t), s_2(t)) & \cdots & r(y_1(t), s_M(t)) \\ r(y_2(t), s_1(t)) & r(y_2(t), s_2(t)) & \cdots & r(y_2(t), s_M(t)) \\ \vdots & \vdots & \ddots & \vdots \\ r(y_M(t), s_1(t)) & r(y_M(t), s_2(t)) & \cdots & r(y_M(t), s_M(t)) \end{bmatrix} \quad (5)$$

根据相关系数的定义, 当相关系数矩阵每行每列都有且仅有一个元素接近于1且其他元素都接近于0时, 可以认为BSS算法较好地估计出了各个源信号。接下来从两个角度观察相关系数矩阵  $R$ :

(1) 误差累积的角度, 取  $R$  各行的最大值

$$r_1 = [\max(R(1, :)), \max(R(2, :)), \dots, \max(R(M, :))]^T \quad (6)$$

由于  $y_k(t)$  代表了BSS系统第k个输出信号, 所以从  $r_1$  的第一个元素到最后一个元素体现了BSS系统的估计准确性随输出次序的变化情况, 可以体现BSS算法是否存在误差累积问题。

(2) 对各个源信号估计精度的角度, 取  $R$  各列的最大值

$$r_2 = [\max(R(:, 1)), \max(R(:, 2)), \dots, \max(R(:, M))]^T \quad (7)$$

式中  $r_2$  的第  $i$  个元素代表了BSS系统对第  $i$  个源信号  $s_i(t)$  的估计准确性。

此外根据矩阵  $Q = \Lambda P$  的性质也可以判断BSS算法的估计准确性, 定义评判标准<sup>[9]</sup>为

$$\text{Index}(Q) = \frac{1}{2} \left[ \sum_i \left( \sum_j \frac{|Q(i, j)|}{\max_l |Q(i, l)|} - 1 \right) + \sum_j \left( \sum_i \frac{|Q(i, j)|}{\max_l |Q(l, j)|} - 1 \right) \right] \quad (8)$$

该值越小, 代表算法的估计准确性越高。

### 1.3 极大化非高斯性的BSS算法

非高斯性<sup>[10]</sup>在BSS中扮演着极为重要的角色, 高阶统计量、负熵等都是衡量非高斯性的指标。根据中心极限定理, 若干个独立随机变量之和所服从的分布的高斯性要比各个原始变量更强, 因此单个独立成分(源信号)的非高斯性要比混合后强。估计独立成分的过程就是极大化非高斯性的过程。用权向量  $w$  对数据向量  $x(t)$  进行加权

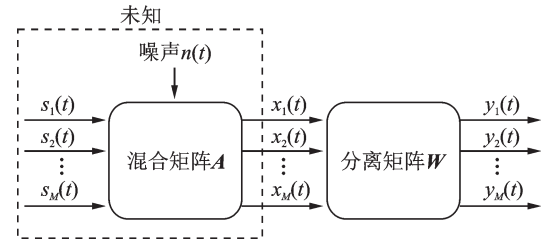


图2 BSS系统

Fig.2 Blind source separation system

$$\mathbf{y}(t) = \mathbf{w}^T \mathbf{x}(t) \quad (9)$$

通过极大化  $\mathbf{w}^T \mathbf{x}(t)$  的非高斯性,可以得到一个权向量  $\mathbf{w}_1$ ,  $\mathbf{y}_1(t) = \mathbf{w}_1^T \mathbf{x}(t)$  即是  $\mathbf{x}(t)$  中的一个独立成分,将向量  $\mathbf{w}_1$  称为独立成分  $\mathbf{y}_1(t)$  对应的方向。

以 FastICA 方法为例,该方法是一种典型的极大化非高斯性的 BSS 算法。FastICA 方法将数据的四阶统计量——峭度作为非高斯性的度量,通过极大化峭度来实现 BSS。与峭度类似,随机变量的三阶统计量——偏度<sup>[11]</sup>也可以衡量非高斯性,本文即从极大化偏度的角度来进行 BSS。

随机变量  $x$  的偏度定义为

$$\text{skew}(x) = \frac{E[x - E(x)]^3}{[\text{Var}(x)]^{3/2}} \quad (10)$$

偏度衡量了一个随机变量概率密度分布的偏斜程度,可以反映出变量的非高斯性。如图 3 所示,当一个变量服从高斯分布时,其概率密度关于  $x=0$  对称,偏度为 0;当长拖尾在右侧时为正偏,偏度值大于 0;当长拖尾在左侧时为负偏,偏度值小于 0。

BSS 算法中常常需要对数据进行预处理<sup>[12]</sup>,预处理包括中心化及白化。对式(1)中  $\mathbf{x}(t)$  进行预处理后得到数据向量  $\mathbf{z}_1(t)$ ,  $\mathbf{z}_1(t)$  满足  $E[\mathbf{z}_1(t)] = \mathbf{0}$ 、 $E[\mathbf{z}_1(t)\mathbf{z}_1(t)^T] = \mathbf{I}$ ,其中  $\mathbf{0}$  为零向量, $\mathbf{I}$  为单位矩阵。接下来通过极大化  $\mathbf{w}^T \mathbf{z}_1(t)$  的偏度可以求解出独立成分的方向。由于  $\mathbf{w}^T \mathbf{z}_1(t)$  的偏度值同时依赖于权向量  $\mathbf{w}$  的范数和方向,一般约束  $\mathbf{w}$  的  $l_2$  范数  $\|\mathbf{w}\|_2 = 1$ 。

将  $E[\mathbf{w}^T \mathbf{z}_1(t)] = 0$ 、 $\text{var}[\mathbf{w}^T \mathbf{z}_1(t)] = E\{[\mathbf{w}^T \mathbf{z}_1(t)]^2\} = \mathbf{w}^T \mathbf{I} \mathbf{w} = 1$  代入式(10),可以得到偏度  $\text{skew}(\mathbf{w}^T \mathbf{z}_1(t))$  的计算公式

$$\text{skew}(\mathbf{w}^T \mathbf{z}_1(t)) = E\{[\mathbf{w}^T \mathbf{z}_1(t)]^3\} = \sum_{i=1}^M \sum_{j=1}^M \sum_{k=1}^M w_i w_j w_k E[z_{1i}(t) z_{1j}(t) z_{1k}(t)] \quad (11)$$

式中,  $w_i (i=1, 2, \dots, M)$  代表  $\mathbf{w}$  的第  $i$  个元素,  $z_{1i}(t) (i=1, 2, \dots, M)$  代表  $\mathbf{z}_1(t)$  的第  $i$  个元素。 $E[z_{1i}(t) z_{1j}(t) z_{1k}(t)]$  可以看作是某个三阶张量的第  $(i, j, k)$  个元素,在 PSA 算法中将这个三阶张量命名为协偏度张量<sup>[7]</sup>。

考虑到将  $\mathbf{w}$  的  $l_2$  范数约束为 1,极大化  $\mathbf{w}^T \mathbf{z}_1(t)$  的偏度可以表示为如式(12)所示的优化问题

$$\begin{cases} \max_{\mathbf{w}} \text{skew}(\mathbf{w}^T \mathbf{z}_1(t)) \\ \text{s.t. } \mathbf{w}^T \mathbf{w} = 1 \end{cases} \quad (12)$$

借助拉格朗日乘子法,式(12)可以转化为

$$L(\mathbf{w}, \lambda) = \text{skew}(\mathbf{w}^T \mathbf{z}_1(t)) + \frac{1}{2} \lambda (1 - \mathbf{w}^T \mathbf{w}) \quad (13)$$

## 2 PSA 算法及并行 PSA 算法

### 2.1 PSA 算法介绍

PSA 算法的核心是能够快速计算式(13)中的偏度  $\text{skew}(\mathbf{w}^T \mathbf{z}_1(t))$ 。计算步骤<sup>[7]</sup>如下:

首先根据式(11)计算协偏度张量,其计算公式为

$$\mathbf{S}(\mathbf{z}_1(t)) = E[\mathbf{z}_1(t) \otimes \mathbf{z}_1(t) \otimes \mathbf{z}_1(t)] \quad (14)$$

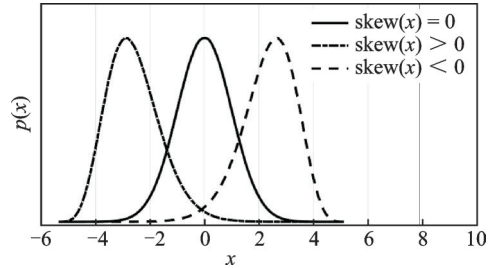


图3 不同偏度值对应的概率密度曲线形态  
Fig.3 Shape of probability density curves corresponding to different skewness values

式中,  $\mathbf{S}(\mathbf{z}_1(t)) \in \mathbf{R}^{M \times M \times M}$  代表数据  $\mathbf{z}_1(t)$  的三阶协偏度张量,  $\otimes$  代表向量的外积运算, 式中的期望在实际计算中用数据样本的平均值来估计。

将式(11)中的偏度  $\text{skew}(\mathbf{w}^T \mathbf{z}_1(t))$  表示为  $\mathbf{S}(\mathbf{z}_1(t))$  与  $\mathbf{w}$  的 3-模式向量积<sup>[13]</sup>

$$\text{skew}(\mathbf{w}^T \mathbf{z}_1(t)) = \sum_{i=1}^M \sum_{j=1}^M \sum_{k=1}^M s_{i,j,k} \mathbf{w}_i \mathbf{w}_j \mathbf{w}_k = \mathbf{S}(\mathbf{z}_1(t)) \times_1 \mathbf{w} \times_2 \mathbf{w} \times_3 \mathbf{w} \quad (15)$$

式中,  $s_{i,j,k}$  代表  $\mathbf{S}(\mathbf{z}_1(t))$  的第  $(i, j, k)$  个元素,  $\times_1, \times_2, \times_3$  分别代表三阶张量的 Tucker 模式-1 积、Tucker 模式-2 积、Tucker 模式-3 积运算符<sup>[13]</sup>。

将式(15)代入式(13)可得

$$L(\mathbf{w}, \lambda) = \mathbf{S}(\mathbf{z}_1(t)) \times_1 \mathbf{w} \times_2 \mathbf{w} \times_3 \mathbf{w} + \frac{1}{2} \lambda (1 - \mathbf{w}^T \mathbf{w}) \quad (16)$$

令  $L_{\mathbf{w}}'(\mathbf{w}, \lambda) = 0$ , 可得

$$\mathbf{S}(\mathbf{z}_1(t)) \times_1 \mathbf{w} \times_3 \mathbf{w} = \lambda \mathbf{w} \quad (17)$$

利用不动点迭代方法求解式(17), 迭代过程如下

$$\begin{aligned} \mathbf{w} &= \mathbf{S}(\mathbf{z}_1(t)) \times_1 \mathbf{w} \times_3 \mathbf{w} \\ \mathbf{w} &= \frac{\mathbf{w}}{\|\mathbf{w}\|} \end{aligned} \quad (18)$$

式(18)计算收敛后, 可以得到一个解  $\mathbf{w} = \mathbf{w}_1$ 。由  $\mathbf{y}_1(t) = \mathbf{w}_1^T \mathbf{z}_1(t)$  得到一个独立成分  $\mathbf{y}_1(t)$ , 即估计出一个源信号。

根据之前的假设, 混合信号中包含  $M$  个独立成分, 接下来需要计算剩余的  $M - 1$  个独立成分所对应的方向。PSA 算法采用了串行的计算方法, 逐次估计各个独立成分对应的方向  $\mathbf{w}_i (i = 2, 3, \dots, M)$ 。假设当前已经估计出  $i - 1$  个方向, 当要估计第  $i$  个独立成分的方向  $\mathbf{w}_i$  时, 该方向应该正交于之前计算得到的  $i - 1$  个方向  $\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_{i-1}$ 。PSA 算法如式(19)将数据  $\mathbf{z}_{i-1}(t) (i = 2, 3, \dots, M)$  投影到由  $\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_{i-1}$  张成的线性子空间的正交补空间, 得到  $\mathbf{z}_i(t)$ 。

$$\begin{aligned} \mathbf{P}_{i-1} &= \mathbf{I} - [\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_{i-1}] [\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_{i-1}]^\# \\ \mathbf{z}_i(t) &= \mathbf{P}_{i-1} \mathbf{z}_{i-1}(t) \quad i = 2, 3, \dots, M \end{aligned} \quad (19)$$

式中,  $\mathbf{P}_{i-1}$  代表投影矩阵, “#” 代表伪逆运算。

接着对  $\mathbf{z}_i(t)$  进行式(18)的迭代, 计算收敛后可以求解出第  $i$  个独立成分对应的方向  $\mathbf{w}_i$ 。经过  $M$  次计算逐次得到  $\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_M$  后, 令  $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_M]^T$ , 则对独立成分的估计  $\mathbf{y}(t)$  为

$$\mathbf{y}(t) = \mathbf{W} \mathbf{z}_1(t) \quad (20)$$

如前文所述, 现有的 PSA 算法采用串行的方法逐次估计  $\mathbf{w}_i (i = 1, 2, \dots, M)$ 。由于每次计算出的独立成分的方向都难以避免地存在一定的误差, 根据式(19), 在数据投影的过程中先前计算产生的误差会累积到后面的方向计算中, 这将导致计算误差随着计算次序的增加而增加。当计算到最后一个独立成分时, 方向误差将累积到较大的水平, 在后面的仿真实验中可以明显看出这种误差累积现象。

## 2.2 并行 PSA 算法

为解决误差累积问题, 本文在 PSA 算法基础上进行改进, 提出了一种并行 PSA 算法。改进后的算法并非逐次估计各个独立成分对应的方向  $\mathbf{w}_i (i = 1, 2, \dots, M)$ , 而是将各个方向并行地、一次全部估计出来, 有效避免了误差累积问题。

根据独立成分间的独立性, 可以证明不同的独立成分所对应的方向  $\mathbf{w}_i$  和  $\mathbf{w}_k$  是正交的

$$E[\mathbf{y}_i(t) \mathbf{y}_k(t)] = E[\mathbf{w}_i^T \mathbf{z}_1(t) \mathbf{z}_1(t)^T \mathbf{w}_k] = \mathbf{w}_i^T E[\mathbf{z}_1(t) \mathbf{z}_1(t)^T] \mathbf{w}_k = \mathbf{w}_i^T \mathbf{w}_k = 0 \quad i \neq k \quad (21)$$

式中  $E[\mathbf{z}_1(t) \mathbf{z}_1(t)^T] = \mathbf{I}$ 。

并行PSA算法与现有PSA算法的差异在于正交化的方式,现有PSA算法采用投影的方式对 $\mathbf{w}_i (i=1,2,\dots,M)$ 进行正交化,由于存在多个独立成分,因此在计算过程中需要进行多次投影,从而实现各个独立成分的正交;为避免多次正交化导致的误差累积,并行PSA算法直接对分离矩阵 $\mathbf{W}=[\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_M]^T$ 进行正交化

$$\mathbf{W} = (\mathbf{W}\mathbf{W}^T)^{-1/2}\mathbf{W} \quad (22)$$

式中逆平方根 $(\mathbf{W}\mathbf{W}^T)^{-1/2}$ 通过对 $\mathbf{W}\mathbf{W}^T$ 进行特征值分解<sup>[12]</sup>得到

$$\begin{aligned} \mathbf{W}\mathbf{W}^T &= \mathbf{E}\text{diag}(d_1, d_2, \dots, d_M)\mathbf{E}^T \\ (\mathbf{W}\mathbf{W}^T)^{-1/2} &= \mathbf{E}\text{diag}(d_1^{-1/2}, d_2^{-1/2}, \dots, d_M^{-1/2})\mathbf{E}^T \end{aligned} \quad (23)$$

通过对分离矩阵 $\mathbf{W}$ 的正交化保证了任意两个不同独立成分对应的方向 $\mathbf{w}_i$ 和 $\mathbf{w}_k$ 是正交的。不同于现有PSA的串行计算方式,该方法可以同时估计出各个方向,因此不存在误差累积的问题。

并行PSA算法计算步骤如下:

(1)对数据向量 $\mathbf{x}(t)$ 预处理得到数据向量 $\mathbf{z}(t)$ 。

(2)按照式(14)计算 $\mathbf{z}(t)$ 的协偏度张量 $\mathbf{S}(\mathbf{z}(t))$ 。

(3)随机初始化分离矩阵 $\mathbf{W}$ ,令 $\mathbf{U}_0 = \mathbf{W}^T = [\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_M]$ ,并根据式(22)对矩阵 $\mathbf{U}_0$ 进行正交化: $\mathbf{U}_0 = (\mathbf{U}_0\mathbf{U}_0^T)^{-1/2}\mathbf{U}_0$ 。

(4)对矩阵 $\mathbf{U}_0$ 的各列 $\mathbf{w}_i (i=1,2,\dots,M)$ 执行式(18)得到 $\mathbf{w}_i' (i=1,2,\dots,M)$ ,令 $\mathbf{U}_1 = [\mathbf{w}_1', \mathbf{w}_2', \dots, \mathbf{w}_M']$ 。

(5)根据式(22)对矩阵 $\mathbf{U}_1$ 进行正交化: $\mathbf{U}_1 = (\mathbf{U}_1\mathbf{U}_1^T)^{-1/2}\mathbf{U}_1$ 。

(6)当 $\mathbf{U}_0$ 与 $\mathbf{U}_1$ 满足收敛条件时,令分离矩阵 $\mathbf{W} = \mathbf{U}_1^T$ ;若不收敛,则令 $\mathbf{U}_0 = \mathbf{U}_1$ ,返回执行第(4)步。

(7)对独立成分的估计 $\mathbf{y}(t) = \mathbf{W}\mathbf{z}(t)$ 。

根据分离矩阵的正交性,在理想情况下,当算法收敛时

$$\mathbf{U}_0^T\mathbf{U}_1 = \mathbf{I} \quad (24)$$

因此在并行PSA方法的第(6)步中,将正交性作为判断算法是否收敛的标准

$$C = 1 - \min(|\text{diag}(\mathbf{U}_0^T\mathbf{U}_1)|) \quad (25)$$

式中 $\text{diag}(\mathbf{U}_0^T\mathbf{U}_1)$ 代表取 $\mathbf{U}_0^T\mathbf{U}_1$ 的对角线元素。 $C$ 越小,代表 $\mathbf{U}_0$ 与 $\mathbf{U}_1$ 的正交性越好。当 $C$ 值小于某一个阈值 $\text{tol}$ 时,判断算法收敛。为方便与现有PSA算法比较,本文中 $\text{tol} = 0.0001$ ,与文献[7]中一致。

### 3 仿真与分析

#### 3.1 并行PSA算法与现有PSA算法的性能比较

本节通过几组数据仿真,说明PSA算法的误差累积问题及并行PSA算法相对PSA算法的性能提升。仿真数据选取了4段音乐信号,信号的采样率 $f_s = 44.1\text{kHz}$ ,信号长度 $T = 34\text{s}$ ,4段信号的波形如图4所示。

第1组仿真实验随机生成混合矩阵 $\mathbf{A}$ ,根据式(1)对4个信号进行混合,该组数据不加入噪声;第2组仿真实验在第1组实验的基础上增加了噪声:将4段音乐随机混合后,按照20 dB的信噪比增加高斯白噪声。之后分别采用PSA算法、并行PSA算法对2组仿真数据进行BSS。以第1组实验为例,图5列出了混合后的波形和BSS系统输出的信号(图中估计信号1到估计信号4分别代表BSS系统依次输出的信号)。从图5可以看出,由于BSS存在不确定性,通过不同方法所估计出的信号次序是随机的,一般与源信号的次序是不相同的。



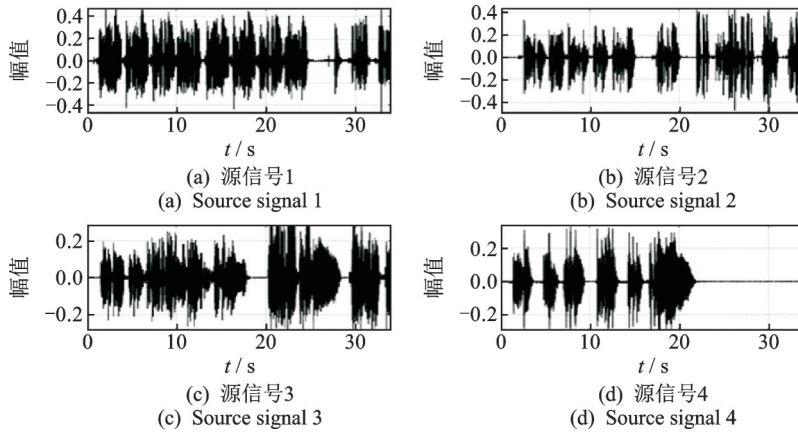


图4 源信号波形

Fig.4 Waveforms of source signals

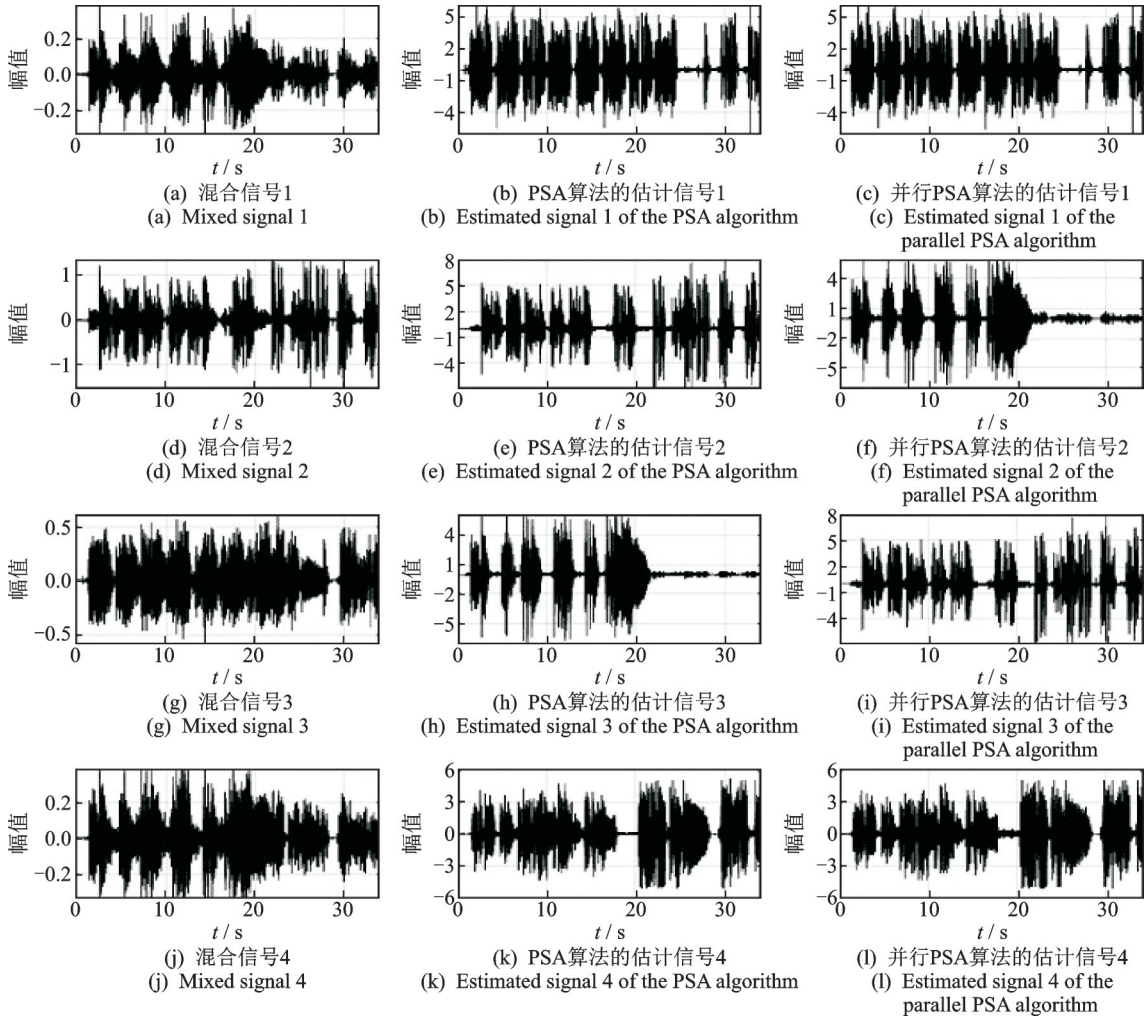


图5 无噪声情况下混合信号及两种算法的计算结果

Fig.5 Mixture signals and results of two algorithms under noise-free condition

在1.2节中定义了 $r_1$ 和 $r_2$ ,这两个量分别从误差累积和估计精度的角度衡量一个BSS算法的整体性能,仿真过程中按照式(6),(7)分别计算两组仿真实验的 $r_1$ 和 $r_2$ ,进行200次蒙特卡洛仿真后,取 $r_1$ 和 $r_2$ 的平均值来对比2个算法的性能。第1组仿真实验的结果如图6(a),(b)所示,第2组仿真实验的结果如图7(a),(b)所示。

图6(a)、图7(a)(两图中估计信号1到估计信号4代表BSS系统依次输出的信号)显示的是根据式(6)计算出的 $r_1$ ,即从误差累积的角度比较并行PSA算法与PSA算法。两图都显示出了PSA算法的误差累积问题,即随着输出次序的增加,误差不断累积,导致相关系数不断下降。与无噪声的情况相比,增加噪声后PSA算法的误差累积问题更加明显:在图7(a)中,PSA算法对前3个输出信号有较高的估计精度,保持在0.9左右;然而对于最后一个输出信号,由于误差累积,相关系数下降到0.65左右,没有准确地估计出对应的源信号。根据前文的分析,并行PSA算法通过并行计算方式避免了误差累积问题:对于无噪声的情况,并行PSA算法的各项相关系数可以稳定在0.998左右;对于包含噪声的情况,各项相关系数可以稳定在0.88左右。2种情况下并行PSA算法都较准确地估计出了各个信号,整体性能优于PSA算法。

图6(b)、图7(b)显示的是根据式(7)计算出的 $r_2$ ,即从对各个源信号的估计精度的角度来比较并行PSA算法和PSA算法。从两张图中可以看出,无论是无噪声情况还是存在高斯白噪声的情况,并行PSA算法对各个源信号的估计精度整体上是优于PSA算法的。在包含噪声的情况下,并行PSA算法的优势较为明显,此时并行PSA算法各项相关系数平均高出PSA算法0.02左右。然而对比前文提到

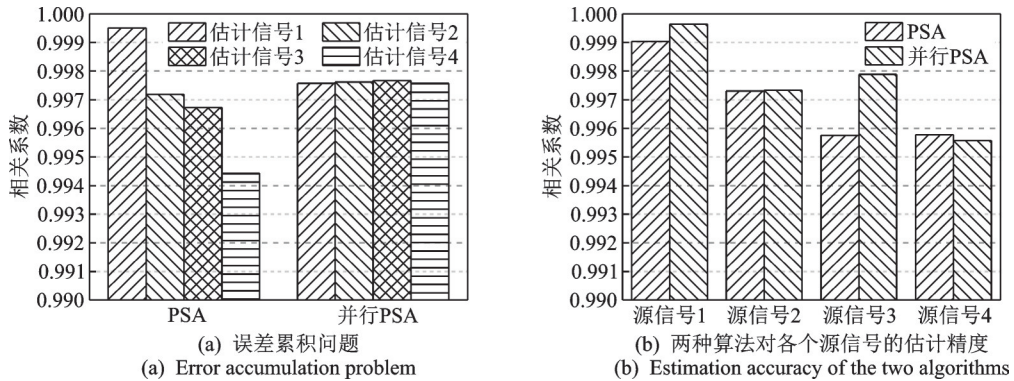


图6 无噪声情况下两种算法的性能对比

Fig.6 Performance comparison of two algorithms under noise-free condition

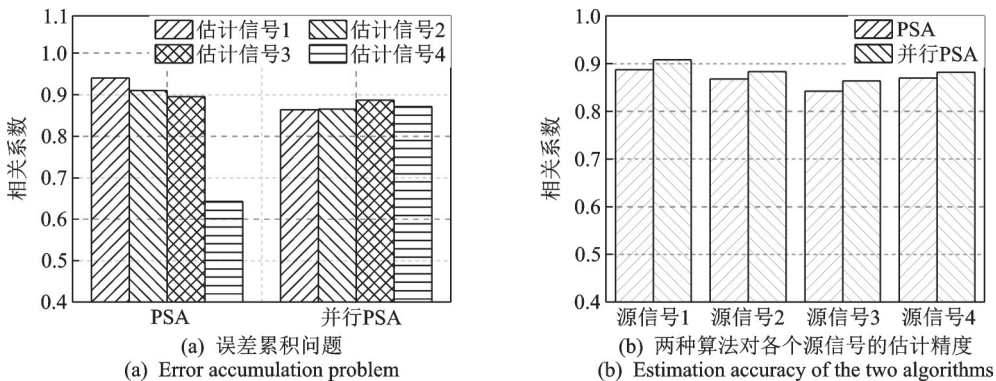


图7 高斯白噪声情况下两种算法性能对比

Fig.7 Performance comparison of two algorithms with white Gaussian noise



的误差累积问题,并行PSA算法的相关系数可以高出PSA算法0.2左右,0.02的精度优势并不明显,造成这种情况的原因是BSS存在不确定性:由于每一次估计出的源信号的顺序是随机的,因此每次计算中误差累积的效果可能会作用于不同的源信号,这样任意一个源信号都有可能出现较大的估计误差,然而数据仿真过程中 $r_2$ 是对200次蒙特卡洛仿真结果取平均值得到的,取平均之后导致各项估计误差都不再那么明显。

第3组仿真实验计算了2种算法的 $\text{Index}(\mathbf{Q})$ 随信噪比变化的情况。仿真数据仍采用第1组仿真的4个源信号,将源信号随机混合后按照不同的信噪比增加高斯白噪声。之后分别采用PSA算法、并行PSA算法对数据进行BSS。根据式(8)计算 $\text{Index}(\mathbf{Q})$ ,进行200次蒙特卡洛仿真后,取平均值。计算结果如图8所示。

由图8可知,随着信噪比的增加, $\text{Index}(\mathbf{Q})$ 不断下降,表示算法的精度在不断提升;同时在不同信噪比下,并行PSA算法的估计精度始终优于PSA算法。

综上,从多个角度比较了并行PSA算法与PSA算法的性能后,由分析可知,并行PSA算法通过并行的计算方式避免了误差累积,可以对各个信号保持较高的估计精度。

### 3.2 并行PSA算法的快速性

文献[7]中的实验结果表明,在保证对源信号估计精度的同时,PSA算法的计算速度快于常规BSS算法。本节将对并行PSA算法、PSA算法及常规BSS算法的计算时间。

本节数据采用3.1节中的第1组仿真数据,分别采用JADE、基于峭度的FastICA、基于偏度的FastICA、PSA、并行PSA等算法对这组数据进行BSS。进行200次蒙特卡洛仿真后,计算各个算法的平均计算时间。以上运算均在同一计算平台上进行,结果如表1所示。

表1 各种BSS算法的平均计算时间

Table 1 Average computing time of different blind source separation algorithms

算法	JADE算法	基于峭度的FastICA算法	基于偏度的FastICA算法	PSA算法	并行PSA算法
计算时间/s	3.099	1.885	0.515	0.395	0.395

表中结果显示并行PSA算法的计算速度与现有PSA算法相当,均快于常规的BSS算法。需要指出,在运算过程中,无论是基于峭度的FastICA算法还是基于偏度的FastICA算法都出现了不能收敛的情况<sup>[14]</sup>,这是FastICA算法存在的问题,而其他3种方法没有出现不收敛的情况。

## 4 结束语

本文研究发现,现有的PSA算法由于采用了串行方式逐次估计各个源信号,在BSS过程中存在误差累积问题,这将导致对源信号的估计准确性降低。为解决这一个问题,本文提出了一种并行PSA算法,该方法采用并行正交化的方式,可以并行地估计各个源信号。通过数值仿真证实了并行PSA算法,避免了误差累积问题,可以对各个源信号保持较高的估计准确性。仿真结果还表明,并行PSA算法的计算速度与PSA算法相当,均快于常规的BSS方法。目前本文仅考虑了瞬态混合BSS模型,下一步将继续研究如何将并行PSA算法应用于更为复杂的卷积混合BSS模型,进一步推广该算法的应用范围。

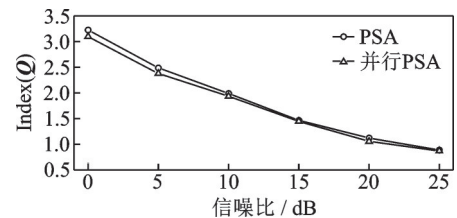


图8 不同信噪比下两种算法的性能对比  
Fig.8 Performance comparison of two algorithms under different signal-to-noise ratios

## 参考文献:

- [1] HYVARINEN A. Fast and robust fixed-point algorithms for independent component analysis[J]. IEEE Transactions on Neural Networks, 1999, 10(3): 626-634.
- [2] CARDOSO J F, SOULOUMIAC A. Blind beamforming for non-Gaussian signals[J]. IEE Proceedings F—Radar and Signal Processing, 1993, 140(6): 362-370.
- [3] BELOUCHRANI A, ABED-MERAIM K, CARDOSO J, et al. A blind source separation technique using second-order statistics[J]. IEEE Transactions on Signal Processing, 1997, 45(2): 434-444.
- [4] JUTTEN C, COMON P. Handbook of blind source separation[M]. Oxford: Academic Press, 2010: 6-7.
- [5] 董晓杰. 基于负熵的独立成分分析算法与应用研究[D]. 哈尔滨: 哈尔滨工业大学, 2009.  
DONG Xiaojie. Algorithms and applications research on independent component analysis based on negentropy[D]. Harbin: Harbin Institute of Technology, 2009.
- [6] 陈锡明, 黄硕翼. 盲源分离综述——问题、原理和方法[J]. 电子信息对抗技术, 2008(2): 1-5.  
CHEN Ximing, HUANG Shuoyi. Blind source separation: Problem, principle and method[J]. Electronic Information Warfare Technology, 2008(2): 1-5.
- [7] GENG X, JI L, KANG S. Principal skewness analysis: Algorithm and its application for multispectral/hyperspectral images indexing[J]. IEEE Geoscience & Remote Sensing Letters, 2014, 11(10): 1821-1825.
- [8] 倪晋平. 水声信号盲分离技术研究[D]. 西安: 西北工业大学, 2002.  
NI Jinping. A study on blind separation of underwater acoustics signals[D]. Xi'an: Northwestern Polytechnical University, 2002.
- [9] MOREAU E. A generalization of joint-diagonalization criteria for source separation[J]. IEEE Transactions on Signal Processing, 2001, 49(3): 530-541.
- [10] 王刚. 基于最大非高斯估计的独立分量分析理论研究[D]. 长沙: 国防科学技术大学, 2005.  
WANG Gang. Research on the theory of maximum nongaussianity estimation in independent component analysis[D]. Changsha: National University of Defense Technology, 2005.
- [11] 张琼, 杨俊安, 夏建明. 基于偏度的低空目标声信号盲抽取算法[J]. 数据采集与处理, 2011, 26(1): 69-73.  
ZHANG Qiong, YANG Junan, XIA Jianming. Novel blind source extraction algorithm to low altitude acoustic signals based on skewness[J]. Journal of Data Acquisition and Processing, 2011, 26(1): 69-73.
- [12] 海韦里恩, 卡尔胡恩, 奥亚. 独立成分分析[M]. 北京: 电子工业出版社, 2014: 147-149, 132-133.  
HYVARINEN A, KARHUNEN J, OJA E. Independent component analysis[M]. Beijing: Publishing House of Electronics Industry, 2014: 147-149, 132-133.
- [13] 张贤达. 矩阵分析与应用[M]. 北京: 清华大学出版社, 2013: 579-581.  
ZHANG Xianda. Matrix analysis and applications[M]. Beijing: Tsinghua University Press, 2013: 579-581.
- [14] GENG X, MENG L, LIN L, et al. Momentum principal skewness analysis[J]. IEEE Geoscience & Remote Sensing Letters, 2015, 12(11): 2262-2266.

## 作者简介:



李康宁(1996-),男,硕士研究生,研究方向:数字信号处理理论与算法,E-mail: likangning18@mails.ucas.ac.cn。



郭永刚(1976-),通信作者,男,研究员,研究方向:水声信号处理、海底观测网,E-mail: guoyong@mail.ioa.ac.cn。



王肃静(1986-),男,副研究员,研究方向:海底观测网、海洋地球物理仪器,E-mail: wangsujing@mail.ioa.ac.cn。



黄石羽(1992-),女,研究生实习员,研究方向:海底观测网、海洋观测传感器,E-mail: huangshiyu@mail.ioa.ac.cn。