

改进的高阶收敛 FastICA 算法

季 策¹, 胡祥楠¹, 朱丽春², 张志伟²

(1. 东北大学 信息科学与工程学院, 辽宁 沈阳 110819; 2. 中国科学院 国家天文台, 北京 100012)

摘 要: 高阶收敛的 FastICA 具有形式简单、收敛速度快的特点, 但其对初始值的选择比较敏感, 若初始值选择不当很容易影响收敛的效果, 甚至造成不收敛的结果. 针对这一问题, 采用最速下降法对三阶和五阶收敛的 FastICA 算法进行改进. 首先, 应用最速下降法求出初值, 再用高阶收敛的 FastICA 算法求出最优解. 语音信号的分离实验表明: 改进后的算法对混合信号进行了较好的分离, 并且有效地克服了初值敏感性的问题.

关 键 词: 独立分量分析; 牛顿迭代法; FastICA; 最速下降法; 初值敏感性

中图分类号: TN 911.6 文献标志码: A 文章编号: 1005-3026(2011)10-1390-04

Improved Higher Order Convergent FastICA Algorithm

Ji Ce, HU Xiang-nan, ZH U Li-chun, ZHANG Zhi-wei

(1. School of Information Science & Engineering, Northeastern University, Shenyang 110819, China; 2. National Astronomical Observatories, Chinese Academy of Sciences, Beijing 100012, China. Corresponding author: Ji Ce, E-mail: huxiangnan121@163.com)

Abstract: High order FastICA algorithms have the advantages of simple form and fast convergence rate. However, they are sensitive to their initial values affecting convergence effect and even resulting in inconvergence if the initial values are not chosen appropriately. To solve the problem, the FastICA algorithms of the third and fifth order convergence were improved with the steepest descent method. First, the initial values were calculated with the steepest descent method. Then, the optimal solution was calculated with the high order convergence FastICA algorithm. Speech signal separation experiments showed that the improved algorithm can separate mixed signal and overcome the initial value sensitivity problems effectively.

Key words: independent component analysis; Newton's iteration method; FastICA; steepest descent method; initial value sensitivity

独立分量分析^[1]是处理盲信号分量的一种有效方法. FastICA 算法, 即固定点算法^[2], 是独立分量分析中重要的优化算法, 它是采用牛顿迭代法的原理^[3], 具有收敛速度快的优点. 特别是改进的 FastICA 算法, 具有三阶^[4]和五阶^[5]的收敛速度. 虽然改进的高阶算法具有较快的收敛速度, 但仍存在对初始权值要求较高的问题, 若初始矩阵为随机矩阵, 不同的运算, 收敛次数相差较大, 无法保证得到均匀的收敛速度.

本文在三阶和五阶收敛的 FastICA 算法的基础上, 通过对初始值进行有效的处理, 达到改善算法的收敛性能和初值敏感性的目的, 进而提高了

盲信号分离的效率.

1 高阶收敛的 FastICA 算法

传统的牛顿迭代法具有二阶的收敛速度, 高阶收敛的牛顿迭代法, 具有三阶和五阶的收敛速度^[6-7].

在三阶和五阶收敛的牛顿迭代法的基础上, 文献[4-5]分别给出了三阶和五阶收敛的 FastICA 算法.

2 改进的高阶 FastICA 算法

FastICA 算法的优点是收敛快、形式简单; 缺

收稿日期: 2011-03-05

基金项目: 国家自然科学基金资助项目(10878017).

作者简介: 季 策(1969-), 女, 辽宁沈阳人, 东北大学副教授; 朱丽春(1964-), 女, 内蒙古赤峰人, 中国科学院国家天文台研究员.

©1994-2013 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

点是对初值要求较高. 由于 FastICA 算法的初始权值一般是随机选取的, 因此就会因为初始权值的不同而导致每次迭代效率不同, 得到的独立分量也会稍有不同. 特别是在源信号独立性不太好的情况, 对权值的选取比较敏感, 不同的初始权值可能导致收敛性能的不同, 甚至会出现收敛和不收敛的两种极端的情况. 针对这一问题, 考虑既要放宽算法对初始权值的要求, 又不能影响算法的收敛速度. 从这个角度出发, 需要找到一种方法与牛顿迭代法联用, 先用该方法求得较好的近似值, 再利用牛顿迭代法继续求出最优解. 理论分析表明: 最速下降法具有对初值不敏感的特点. 因此, 可利用最速下降法先求出近似值, 再结合高阶收敛的 FastICA 算法求出最优解.

2.1 最速下降法

最速下降法是以负梯度方向作为下降方向的优化算法, 又称梯度法. 其计算步骤如下.

- 1) 选取一初始化随机正交阵:
 $w = [w_1, w_2, \dots, w_n]$, 首先令 $k = 0$.
- 2) 计算 $E\{xg(w^T x)\}$ 在 w 处的梯度值:

$$\lambda = - \begin{bmatrix} \frac{\partial E\{xg(w^T x)\}}{\partial w_1} & 0 & \dots & 0 \\ 0 & & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \dots & 0 & \frac{\partial E\{xg(w^T x)\}}{\partial w_n} \end{bmatrix}^{-1}, \tag{1}$$

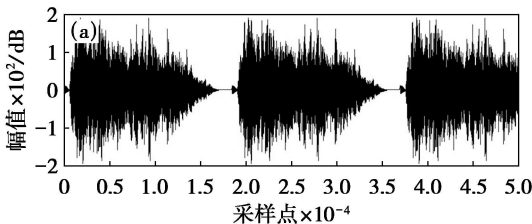
式中: $\frac{\partial E\{xg(w^T x)\}}{\partial w_i} = E\{(x^2 g'(w^T x))\}$, $i = 1, 2, \dots, n$. 由多元微分学可知, 负梯度的方向就是函数值下降最快的方向, 因此, 将梯度值作为松弛因子代入下式:

$$w' = w + \lambda E\{xg(w^T x)\}.$$

- 3) 检验是否收敛, 若收敛取 $w_0 = w'$, 若不收敛, $k = k + 1$, 转 2).

2.2 三阶收敛的 FastICA 算法的改进步骤

- 1) 将观测信号 x 进行中心化处理.



- 2) 对中心化后的 x 进行白化处理.
- 3) 初始化随机权矢量 w , 利用最速下降法求出 w_0 , 设置收敛误差 $0 < \varepsilon \ll 1$.
- 4) 利用文献[4]中的公式(9)调整 w_{k+1} , 计算出 w_{k+1} 后用 $w(k+1) = \frac{w(k+1)}{\|w(k+1)\|}$ 去相关和归一化^[8].
- 5) 如果 $|w_{k+1} - w_k| < \varepsilon$, 则算法收敛, 估算出一个独立分量, 否则重复步骤 4) 和 5).
- 6) 由分离矩阵得到源信号的所有估计 y_i , 输出估计信号 Y .

2.3 五阶收敛的 FastICA 算法的改进步骤

- 1) 将观测信号 x 进行中心化处理.
- 2) 对中心化后的 x 进行白化处理.
- 3) 初始化随机权矢量 w , 利用最速下降法求出 w_0 , 设置收敛误差 $0 < \varepsilon \ll 1$.
- 4) 利用文献[5]中的公式(12)调整 w_{k+1} , 计算出 w_{k+1} 后用 $w(k+1) = \frac{w(k+1)}{\|w(k+1)\|}$ 去相关和归一化.
- 5) 如果 $|w_{k+1} - w_k| < \varepsilon$, 则算法收敛, 估算出一个独立分量, 否则重复步骤 4) 和 5).
- 6) 由分离矩阵得到源信号的所有估计 y_i , 输出估计信号 Y .

在改进算法中, 将两种算法结合使用, 扬长避短, 不仅收敛快, 而且实现了大范围收敛的目的.

3 仿真实验和算法分析

3.1 仿真实验

仿真实验是应用 Matlab 软件, 对语音信号进行分离, 选定两路语音信号, 采样频率为 50 kHz.

首先对两路音频信号(图 1), 按随机生成的 2×2 矩阵 $A = \begin{bmatrix} 0.8976 & 0.5745 \\ 0.7465 & 0.8796 \end{bmatrix}$ 混合, 得到一组混合信号(图 2); 然后分别用原始 FastICA 和改进算法对语音信号进行处理, 得到分离后的结果, 如图 3~ 图 5 所示. 可以看出, 改进算法可以成功分离语音信号.

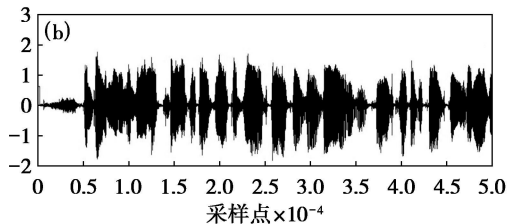


图 1 原始信号
Fig. 1 Original signals
(a) 一报警声音信号; (b) 一讲话声音信号.

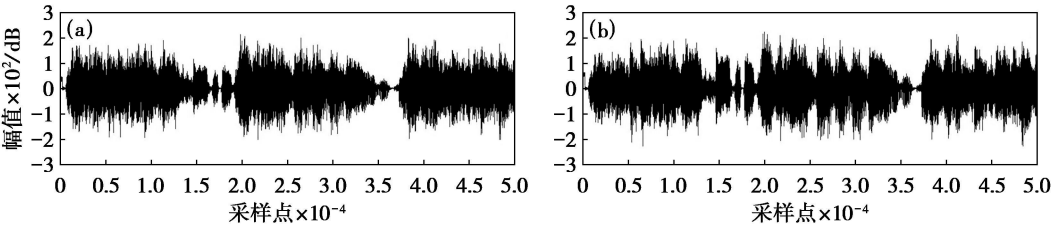


图 2 混合后的信号

Fig.2 Mixed signals

(a) —报警声音信号; (b) —讲话声音信号.

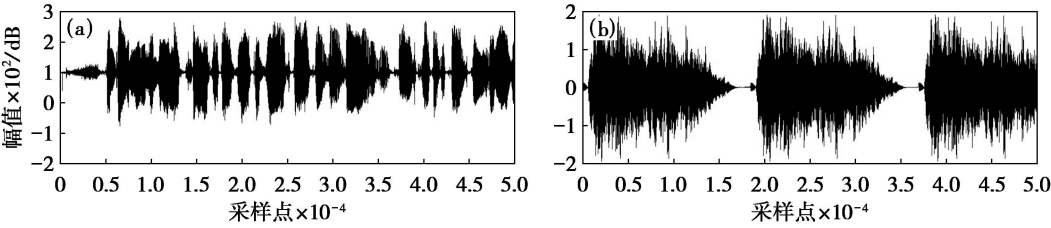


图 3 原始 FastICA 分离后的信号

Fig.3 Signals separated by original FastICA

(a) —讲话声音信号; (b) —报警声音信号.

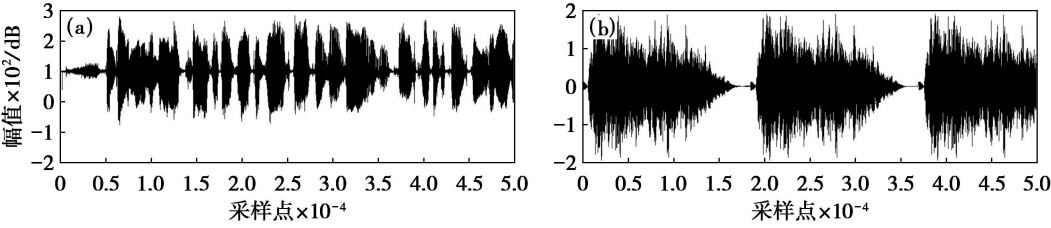


图 4 改进的三阶收敛 FastICA 算法分离后的信号

Fig.4 Signals separated by improved third-order convergent FastICA algorithm

(a) —讲话声音信号; (b) —报警声音信号.

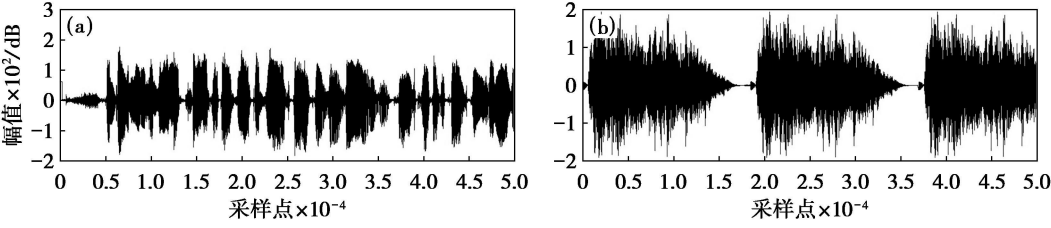


图 5 改进的五阶收敛 FastICA 算法分离后的信号

Fig.5 Signals separated by improved fifth-order convergent FastICA algorithm

(a) —讲话声音信号; (b) —报警声音信号.

3.2 算法的性能分析

通过ICA求解得到混合矩阵和源信号的估计值,与混合矩阵和源信号的实际值进行比较,就可以评价ICA算法的性能.

通常应用式(2)来衡量算法的分离性能:

$$PI = \frac{1}{2M} \sum_{i=1}^M \left\{ \left(\sum_{j=1}^M \frac{|c_{ij}|^2}{\max_k |c_{ik}|^2} - 1 \right) + \left(\sum_{j=1}^M \frac{|c_{ji}|^2}{\max_k |c_{ki}|^2} - 1 \right) \right\} \quad (2)$$

式中: M 为变量的个数; PI 表示算法的分离性能,其值越小表示分离的性能越好.当 $PI=0$ 时表示信号完全分离.计算可得各算法的 PI 值见表1

(FastICA-T 为三阶收敛的 FastICA 算法, FastICA-F 为五阶收敛的 FastICA 算法).

表 1 PI 值的比较
Table 1 Comparison of PI values

算 法	PI 值
FastICA	0.0455
改进 FastICA-T	0.0397
改进 FastICA-F	0.0462

由表1可知,改进后的算法与原算法的性能指标几乎相同,说明改进的算法也较好地保留了原语音信号的信息.

对 FastICA 和改进算法的收敛速度进行比

较, 每种算法均随机进行了 8 次分离运算, 每次运算随机给定一个初始值, 记录每种算法的迭代次数和平均迭代次数. 每次迭代选定的初值分别为

$$\begin{aligned}w_{01} &= \begin{bmatrix} 0.856\ 3 & 0.695\ 4 \\ 0.743\ 2 & 0.579\ 5 \end{bmatrix}, \\w_{02} &= \begin{bmatrix} 0.739\ 1 & -1.494\ 1 \\ 0.965\ 4 & 2.539\ 5 \end{bmatrix}, \\w_{03} &= \begin{bmatrix} 1.254\ 3 & -1.495\ 6 \\ -1.989\ 7 & 0.886\ 4 \end{bmatrix}, \\w_{04} &= \begin{bmatrix} 4.396\ 5 & 3.254\ 8 \\ 2.539\ 6 & 1.493\ 6 \end{bmatrix}, \\w_{05} &= \begin{bmatrix} 4.396\ 5 & -3.254\ 8 \\ -2.539\ 6 & 1.493\ 6 \end{bmatrix}, \\w_{06} &= \begin{bmatrix} 0.125\ 6 & 0.439\ 5 \\ 0.243\ 1 & 0.159\ 6 \end{bmatrix},\end{aligned}$$

$$\begin{aligned}w_{07} &= \begin{bmatrix} 0.887\ 9 & 0.765\ 9 \\ 0.997\ 6 & 0.901\ 2 \end{bmatrix}, \\w_{08} &= \begin{bmatrix} 0.929\ 9 & -0.909\ 5 \\ 0.994\ 6 & -0.898\ 8 \end{bmatrix}.\end{aligned}$$

从表 2 可知, 使用原始 FastICA 算法的收敛速度不均匀, 波动较大, 迭代次数在 9 到 13 之间波动. 在改进算法中, 开始使用最速下降法求出分离矩阵值 w_0 , 然后进行迭代. 改进的 FastICA-T 算法在 8 和 9 之间波动; 改进的 FastICA-F 算法在 7 和 9 之间波动. 由此可见改进的算法波动较小, 有效地克服了对初始值的依赖性, 达到了收敛速度均衡的目的. 尽管使用的最速下降法消耗了一定的时间, 但改进的算法较好地解决了初值敏感性的问题, 避免了收敛速度不均匀的现象.

表 2 迭代次数的比较

Table 2 Comparison of iteration numbers

算 法	运 算 序 号								平均次数
	1	2	3	4	5	6	7	8	
FastICA	11	10	11	11	11	9	12	13	11
改进 FastICA-T	8	8	8	9	9	8	9	9	8.5
改进 FastICA-F	8	7	7	8	8	8	8	9	7.8

4 结 论

传统的牛顿迭代法存在对初始值敏感的问题, 为克服这一缺点, 本文在三阶收敛和五阶收敛的牛顿迭代算法的基础上, 结合最速下降法给出了一种改进的高阶 FastICA 算法. 语音信号的分离实验证明改进后的算法在保持收敛速度的情况下, 能避免收敛速度不均衡的问题, 较好实现了大范围的收敛.

参考文献:

[1] Castano S, Ferrara A, Montanelli S, *et al.* HELIOS: a general framework for ontology-based knowledge sharing and an evolution in P2P systems[C] // The 14th International Workshop on Database and Expert Systems Applications. Prague: IEEE Press, 2003: 597– 603.

[2] Hyvärinen A. Independent component analysis: algorithms and applications[J]. *Neural Networks*, 2000, 13: 411– 430.

[3] von Hoff T P, Lindgren A G, Kaelin A N. Step-size control in

blind source separation[C] // Proceedings of ICABSS' 2000. Helsinki, 2000: 509– 513.

[4] Ji C, Yu Y, Yu P. A new FastICA algorithm of Newton' iteration[C] // ICETC 2010. Helsinki, 2010: 481– 484.

[5] Ji C, Yu Y, Yu P. Improved algorithm for independent component analysis[C] // Processing of the 2010 International Conference on Future Computer and Communication. New York, 2010: 467– 470.

[6] 方建斌, 李自玲, 管琼. 具有三阶收敛的“牛顿类”迭代法[J]. 武汉大学学报, 2009, 31(3): 398– 400.
(Fang Jianbin, Li Ziling, Guan Qiong. “Newton like” iteration method with third-order convergence[J]. *Journal of Wuhan University*, 2009, 31(3): 398– 400.)

[7] 苏岐芳, 李希文. 一类具有五阶收敛的牛顿改进法[J]. 台州学院学报, 2008, 30(6): 9– 11.
(Su Qifang, Li Xiwen. A class of modifications to Newton' s method with fifth-order convergence[J]. *Journal of Taizhou University*, 2008, 30(6): 9– 11.)

[8] Hyvarinen A. Fast and robust fixed-point algorithm for independent component analysis[J]. *IEEE Transactions on Neural Network*, 1999, 10(3): 626– 634.