第 8 卷　第 2 期　信 息 与 电 子 工 程　Vo1.8, No.2
2010 年 4 月　INFORMATION AND ELECTRONIC ENGINEERING　Apr., 2010

文章编号：1672-2892(2010)02-0207-05

独立分量分析可调速率相对梯度算法

程　娇，王晓凯，李　锋

(复旦大学 电子工程系，上海 200433)

摘　要：在独立分量分析的相对梯度算法中，要取得较好的效果，选取合适的学习速率是至关重要的。对于这个问题，文章提出了一种可调速率的相对梯度算法，随着迭代次数的变化，使相对梯度算法的学习速率作相应变化，从而较好地解决了收敛速度与稳定性的矛盾。在此基础上，将这个方法应用于盲信号分离并进行仿真，得到了满意的结果。可调速率相对梯度算法在独立分量分析中具有较好的前景。

关键词：独立分量分析；盲信号分离；可调速率；相对梯度

中图分类号：TN911.7　　**文献标识码：**A

Adjustable rate algorithm with relative gradient of ICA

CHENG Jiao, WANG Xiao-kai, LI Feng

(Department of Electronic Engineering, Fudan University, Shanghai 200433, China)

Abstract: In relative gradient algorithms of Independent Component Analysis(ICA), careful selection of step size is important to obtain good performance. In this study, an adjustable rate of relative gradient algorithm was proposed. With the changes of iteration number, the learning rate of relative gradient algorithm changed correspondingly, which solved the problem about the contradiction between the convergence rate and stability well. On this basis, this method was adopted in Blind Signal Separation(BSS) problems, and its effectiveness was validated by simulation. Adjustable rate algorithm with relative gradient has good prospects in independent component analysis.

Key words: Independent Component Analysis; Blind Signal Separation; adjustable rate; relative gradient

独立分量分析(ICA)是由盲信号分离(BSS)技术发展起来的多通道信号处理方法[1]。20 世纪 80 年代末法国学者 Herailt J,Jutten C 和 Ans B 首先提出了 ICA 分析的基本概念并给出了基于神经网络方法的 H–J 算法[2]。1994 年，Common P 第一次使用了独立分量分析(ICA)这个名词并界定了解决 BSS 问题的 ICA 方法的基本假设条件[3]。1995 年，Bell A J 和 Sernowski T J 发表了基于信息最大化原理的盲分离和盲反卷积论文[4]，这是 ICA 技术发展中的里程碑式的文献。1996 年，Cardoso J F 和 Laheld B H 提出了 ICA 学习算法中的“相对梯度”、“等价变化”和有关稳定性和分离精度等重要思路和方法[5]。ICA 着眼于信号的高阶统计特性，以信号非高斯性和统计独立性为依据，从多维统计数据中寻找其内在分量。

独立分量分析的学习算法主要有以下 4 种[6]：1) 相对梯度算法；2) 随机梯度算法；3) 迭代求逆算法；4) 固定点算法。其中最经典最常用的是相对梯度算法，但缺点是：若学习速率大，则算法收敛快，但信号的稳定性差；反之，若学习速率慢，则稳定性好，但算法收敛慢。较好的做法是采用随时间变化的学习速率即变步长[7–9]，但是文献[7–9]针对的是在线独立分量分析，对于批处理应用则没有涉及。本文提出了一种新的可调速率相对梯度算法，对于批处理应用，在稳定性和收敛速度方面都有一定的改进。

1　独立分量分析的基本概念

在盲信号模型中，设有 n 个未知且相互独立的源信号 $\boldsymbol{s}(t)=[s_1(t),s_2(t),\cdots,s_T(t)]^{\mathrm{T}}$，其中 t 为离散时刻，取值为 $0,1,2,\cdots,T$。设 $\boldsymbol{A}$ 是一个 $m\times n$ 维矩阵，一般称为混合矩阵。设 $\boldsymbol{x}(t)=[x_1(t),x_2(t),\cdots,x_m(t)]^{\mathrm{T}}$ 为 m 维的观测信号向量，

收稿日期：2009-07-10；**修回日期：**2009-08-31

$\boldsymbol{x}(t)$ 是由源信号向量 $\boldsymbol{s}(t)$ 线性组合而成，则盲信号分离的线性混合模型可表示为：

$$\boldsymbol{x}(t)=\boldsymbol{A}\boldsymbol{s}(t) \tag{1}$$

设随机向量 $\boldsymbol{x}$ 的 pdf 是 $p_{\boldsymbol{x}}(\boldsymbol{x})$，可由随机向量的样本估计得到。已知 $\boldsymbol{s}(t)$ 的 pdf 为 $p_{\boldsymbol{s}}(\boldsymbol{s})=\prod_{i=1}^{n}p_i(s_i)$，则由 $\boldsymbol{x}(t)=\boldsymbol{A}\boldsymbol{s}(t)=\boldsymbol{W}^{-1}\boldsymbol{s}(t)$，得到 $\boldsymbol{x}(t)$ 的似然 pdf[1]：

$$\hat{p}_{\boldsymbol{x}}(\boldsymbol{x})=\left|\det\boldsymbol{W}\right|p_{\boldsymbol{s}}(\boldsymbol{s})\Big|_{\boldsymbol{s}=\boldsymbol{W}\boldsymbol{x}}=\left|\det\boldsymbol{W}\right|\prod_{i=1}^{n}p_i(s_i)\Big|_{\boldsymbol{s}=\boldsymbol{W}\boldsymbol{x}} \tag{2}$$

记目标函数 $L(\boldsymbol{W})=-\int_{\boldsymbol{x}}p_{\boldsymbol{x}}(\boldsymbol{x})\ln\hat{p}_{\boldsymbol{x}}(\boldsymbol{x})\mathrm{d}\boldsymbol{x}$。如果观察的样本长度有限，假设在离散时刻 1,2,⋯,T 采样，则目标函数只能用有限多个样本来近似，记为 $\tilde{L}(\boldsymbol{W})$，计算公式为：

$$\tilde{L}(\boldsymbol{W})=-\frac{1}{T}\sum_{t=1}^{T}\ln\hat{p}_{\boldsymbol{x}}(\boldsymbol{x})=-\ln\left|\det\boldsymbol{W}\right|-\frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{n}\ln p_i(s_i(t))\Big|_{\boldsymbol{s}=\boldsymbol{W}\boldsymbol{x}} \tag{3}$$

2 可调速率相对梯度学习算法

设 $\boldsymbol{I}$ 是 $n\times n$ 维单位阵，$\boldsymbol{\varepsilon}=(\varepsilon_{ij})$ 是一个 $n\times n$ 维的微小矩阵，则下式成立[10]：

$$\tilde{L}((\boldsymbol{I}+\boldsymbol{\varepsilon})\boldsymbol{W})=\tilde{L}(\boldsymbol{W})+\left\langle\nabla_{\boldsymbol{\varepsilon}}\tilde{L}(\boldsymbol{W})\middle|\boldsymbol{\varepsilon}\right\rangle+O(\boldsymbol{\varepsilon}) \tag{4}$$

式中：$O(\boldsymbol{\varepsilon})$ 是比 $\left\|\boldsymbol{\varepsilon}\right\|$ 小一个数量级的微小量；相对梯度 $\nabla_{\boldsymbol{\varepsilon}}\tilde{L}(\boldsymbol{W})=\dfrac{\partial\tilde{L}((\boldsymbol{I}+\boldsymbol{\varepsilon})\boldsymbol{W})}{\partial\boldsymbol{\varepsilon}}$；$\left\langle\nabla_{\boldsymbol{\varepsilon}}\tilde{L}(\boldsymbol{W})\middle|\boldsymbol{\varepsilon}\right\rangle$ 为 $\nabla_{\boldsymbol{\varepsilon}}\tilde{L}(\boldsymbol{W})$ 和 $\boldsymbol{\varepsilon}$ 的内积。又因为

$$\tilde{L}(\boldsymbol{W}+\Delta\boldsymbol{W})=\tilde{L}(\boldsymbol{W})+\left\langle\nabla_{\boldsymbol{W}}\tilde{L}(\boldsymbol{W})\middle|\Delta\boldsymbol{W}\right\rangle+O(\Delta\boldsymbol{W}) \tag{5}$$

式中：普通梯度 $\nabla_{\boldsymbol{W}}\tilde{L}(\boldsymbol{W})=\dfrac{\partial\tilde{L}(\boldsymbol{W})}{\partial\boldsymbol{W}}$；$O(\Delta\boldsymbol{W})$ 是较 $\left\|\Delta\boldsymbol{W}\right\|$ 小一个数量级的微小量。设 $\boldsymbol{\varepsilon}\boldsymbol{W}=\Delta\boldsymbol{W}$，则由式(4)、式(5)可得 $\left\langle\nabla_{\boldsymbol{W}}\tilde{L}(\boldsymbol{W})\middle|\Delta\boldsymbol{W}\right\rangle=\left\langle\nabla_{\boldsymbol{W}}\tilde{L}(\boldsymbol{W})\middle|\boldsymbol{\varepsilon}\boldsymbol{W}\right\rangle=\left\langle\nabla_{\boldsymbol{\varepsilon}}\tilde{L}(\boldsymbol{W})\middle|\boldsymbol{\varepsilon}\right\rangle$，根据内积的性质，可得：$tr[\nabla_{\boldsymbol{W}}\tilde{L}(\boldsymbol{W})(\boldsymbol{\varepsilon}\boldsymbol{W})^{\mathrm{T}}]=tr[\nabla_{\boldsymbol{\varepsilon}}\tilde{L}(\boldsymbol{W})\boldsymbol{\varepsilon}^{\mathrm{T}}]$。

最后得到相对梯度和普通梯度的关系：

$$\nabla_{\boldsymbol{W}}\tilde{L}(\boldsymbol{W})\boldsymbol{W}^{\mathrm{T}}=\nabla_{\boldsymbol{\varepsilon}}\tilde{L}(\boldsymbol{W}) \tag{6}$$

在按照相对梯度进行迭代计算时，由于 $\boldsymbol{\varepsilon}(k)=-u(k)\nabla_{\boldsymbol{\varepsilon}}\tilde{L}(\boldsymbol{W})\Big|_{\boldsymbol{W}=\boldsymbol{W}(k)}$，由式(6)可得：

$$\Delta\boldsymbol{W}(k)=\boldsymbol{\varepsilon}(k)\boldsymbol{W}(k)=-u(k)\nabla_{\boldsymbol{W}}\tilde{L}(\boldsymbol{W})\boldsymbol{W}^{\mathrm{T}}\boldsymbol{W}\Big|_{\boldsymbol{W}=\boldsymbol{W}(k)} \tag{7}$$

进而可得：

$$\Delta\boldsymbol{W}(k)=u(k)[\boldsymbol{I}-\frac{1}{T}\sum_{t=1}^{T}\varphi(\mathbf{s}(t))\boldsymbol{s}^{\mathrm{T}}(t)]\boldsymbol{W}(k) \tag{8}$$

式中：$\boldsymbol{s}(t)=\boldsymbol{W}(t)\boldsymbol{x}(t)$；$\varphi(\boldsymbol{s}(t))=\left[-\dfrac{p'_1(s_1(t))}{p_1(s_1(t))},-\dfrac{p'_2(s_2(t))}{p_2(s_2(t))},\cdots,-\dfrac{p'_n(s_n(t))}{p_n(s_n(t))}\right]^{\mathrm{T}}$；$p'_i(s_i(t))$ 是 $p_i(s_i(t))$ 对 $s_i(t)$ 的导数。在用 ICA 解决实际问题中，$p_i(s_i)$ 是未知的，一般只能用 $\hat{p}_i(s_i)$ 来尽可能地接近 $p_i(s_i)$，在不能做到二者严格一致的情况下，应保证 $\hat{p}_i(s_i)$ 与 $p_i(s_i)$ 具有同样的超高斯或次高斯特性[1]。

如果需要对数据进行实时处理，则每得到一个观察向量 $\boldsymbol{x}(t)$，就需要进行 1 次迭代计算，迭代计算节拍 k 与 $\boldsymbol{x}(t)$ 的时序 t 相重合，由文献[1]可知实时在线自适应学习算法公式：

$$\Delta\boldsymbol{W}(k)=u(k)[\boldsymbol{I}-\varphi(\boldsymbol{s}(k))\boldsymbol{s}^{\mathrm{T}}(k)]\boldsymbol{W}(k) \tag{9}$$

和很多迭代算法一样，在式(8)中，迭代计算的步幅 $u(k)$ 的选择是至关重要的。$u(k)$ 太小时收敛很慢，太大时则会造成失调。以音频信号为例，取 $p_i(s_i(t))=(\dfrac{2}{\mathrm{e}^{s_i(t)}+\mathrm{e}^{-s_i(t)}})^2$，$u(k)$ 的选择对独立分量分析的影响如图 1~图 5 所示。

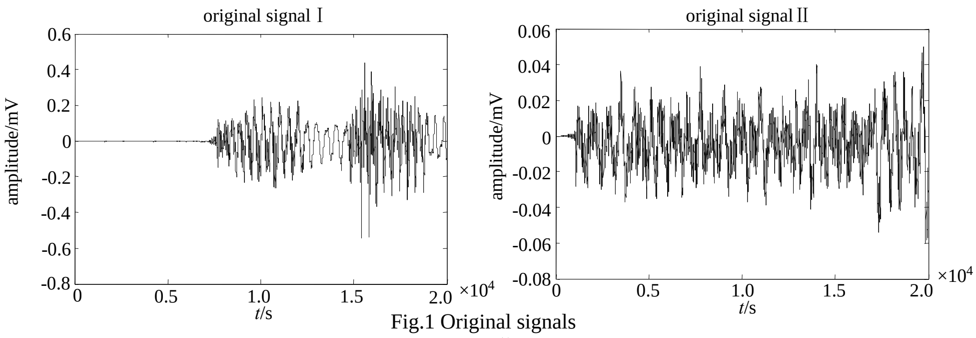

Fig.1 Original signals
图 1　源信号

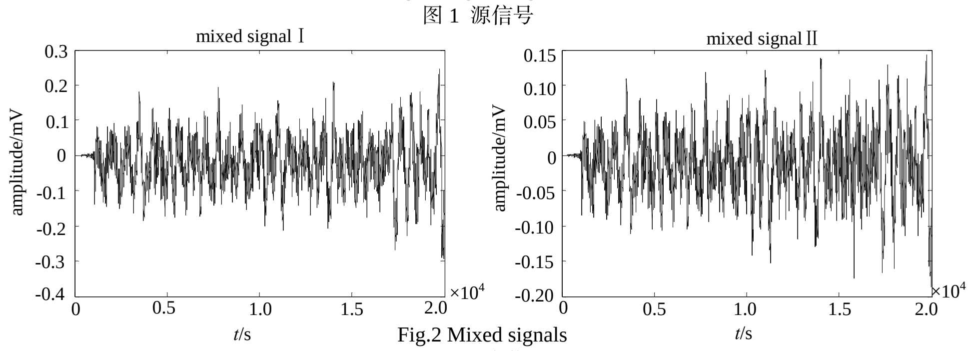

Fig.2 Mixed signals
图 2　混合信号

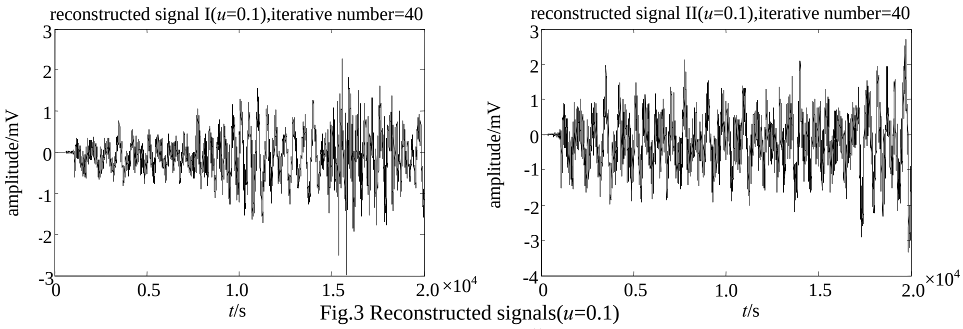

Fig.3 Reconstructed signals(u=0.1)
图 3 u=0.1 时输出信号

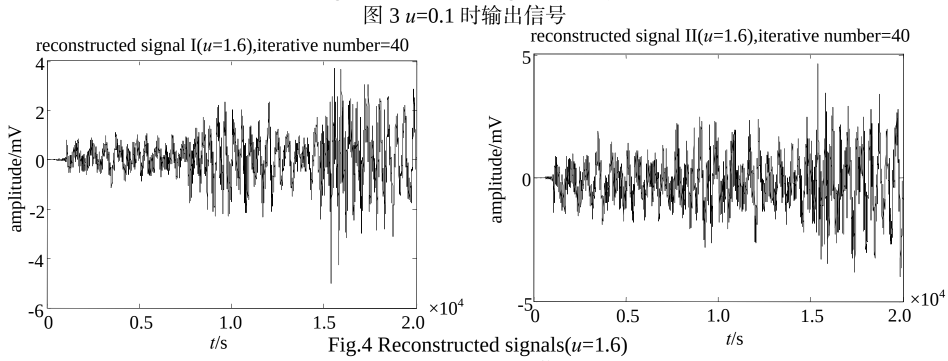

Fig.4 Reconstructed signals(u=1.6)
图 4 u=1.6 时输出信号

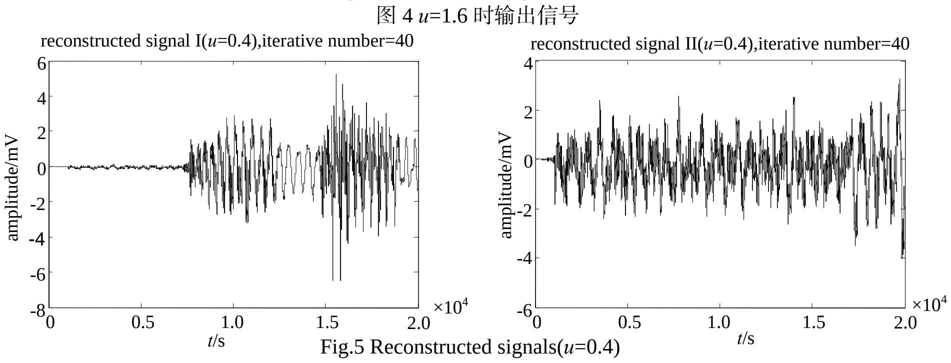

Fig.5 Reconstructed signals(u=0.4)
图 5 u=0.4 时输出信号

由图 1~图 5 可以看出选取合适的 u 值对独立分量分析的结果至关重要。图 1 是 2 个原始信号；图 2 为混合

后的信号；图 3 中取 $u=0.1$，导致在一定的迭代次数里不能收敛；图 4 中取 $u=1.6$ 时，造成结果失调；图 5 中取 $u=0.4$，有较好的效果。实际应用中，选取合适 u 值则很有必要，但如何选取一个合适 u 值一直没有较好的方法。

本文提出一种可调速率相对梯度算法，改进部分如下：

1) 独立分量分析的目的是使目标函数最小，故选初始目标函数 $\tilde{L}(\boldsymbol{W})\big|_{\boldsymbol{W}=\boldsymbol{I}}$，又由于目标函数极小值非零，故需加一个常数值，由式(3)得：

$$\tilde{L}(I)=-\frac{1}{T}\sum_{t=1}^{T}\ln\hat{p}_{\boldsymbol{x}}(\boldsymbol{x})=-\ln\left|\det(\boldsymbol{I})\right|-\frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{n}\ln p_i(s_i(t))\Big|_{\boldsymbol{s}=\boldsymbol{I}\boldsymbol{x}} \tag{10}$$

简化，得：

$$\tilde{L}(I)=-\frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{n}\ln p_i(x_i(t)) \tag{11}$$

取

$$u(0)=\tilde{L}(I)+\beta \tag{12}$$

式中 β 值一般在 0.75~0.85 之间选取。

2) 随迭代次数 k 增加，$u(k)$ 值应相应变化，取如下式：

$$u(k+1)=\alpha(k)u(k) \tag{13}$$

$\alpha(k)$ 一般在 0.97~0.99 之间选取。

3 仿真和实验结果

在文献[10]中，介绍一种简单的变速率算法，任意确定一个初始 $u(0)$，然后迭代速率变化如下：

$$u(k)=u(0)/k \tag{14}$$

对于算法的误差判定，由于最终分离信号是使 $\boldsymbol{W}\times\boldsymbol{A}$ 经过尺度交换变换，趋近于单位阵，定义误差系数：

$$Err=20\times\lg\left\{\sum_{i=1}^{n}\frac{\mathrm{Sum}(\mathrm{abs}(\boldsymbol{W}\times\boldsymbol{A}(i,:)))-\mathrm{Max}(\mathrm{abs}(\boldsymbol{W}\times\boldsymbol{A}(i,:)))}{\mathrm{Max}(\mathrm{abs}(\boldsymbol{W}\times\boldsymbol{A}(i,:)))}\right\} \tag{15}$$

式中 abs(□) 为取绝对值函数。

从图 6 可以看出当 u 值取太小或太大时，都有收敛速度慢，或稳态误差大的问题。从图 7 中可以看出，采用简单的变速率算法式(19)，效果还不如选取合适 u 值的定速率算法，而本文的算法不仅收敛速度变快，而且误差系数也要优于前者。图 7 可以发现本文算法的稳态误差大于 $u=0.4$ 的定速率算法。在本实验中，$u=0.4$ 是经过多次实验的最优值，在实际应用中，由于混合矩阵 $\boldsymbol{A}$ 未知，很难得到最优的 u 值。故可调速率相对梯度算法较好地解决了收敛速度和稳态误差的矛盾，实际意义明显。

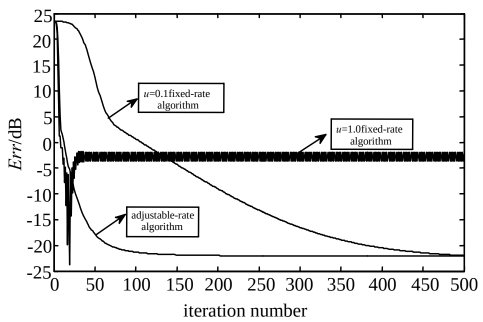

Fig.6 Error factors comparison 1 of these algorithms
图 6 算法误差系数比较 1

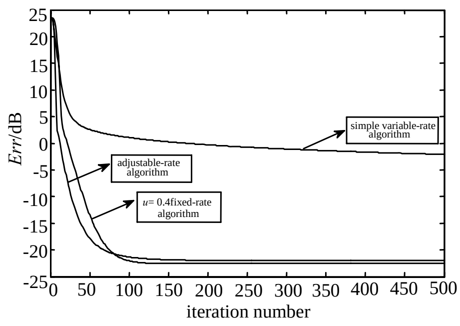

Fig.7 Error factors comparison 2 of these algorithms
图 7 算法误差系数比较 2

4 结论

本文提出了一种可调速率的相对梯度算法，对于传统相对梯度算法的收敛速度和稳态误差相互矛盾的问题提出了解决方法。仿真证明本方法的有效性，能够兼顾收敛速度和稳态误差，具有一定的应用价值。

参考文献：

[1] Hyvarinen A,Oja E. Independent Component Analysis[M]. New York:Joh Wiley, 2001.

(下转第 222 页)