

并行提取多个次成分的改进型Möller算法

高迎彬^{1,2}, 孔祥玉^{1†}, 胡昌华¹, 侯立安¹

(1. 第二炮兵工程大学 控制工程系, 西安 710025; 2. 第二炮兵驻石家庄地区军事代表室, 石家庄 050081)

摘要: 次成分分析是信号处理领域一门重要的工具. 然而, 到目前为止能够进行多个次成分提取的算法并不多见, 一些现存算法还存在很多限制条件. 针对这些问题, 采用加权矩阵的方法将Möller算法扩展为多个次成分提取算法. 该算法对于输入信号的特征值没有要求, 而且在不需要模值限制措施的情况下, 仍然具有很好的收敛性. 仿真结果表明, 该算法可并行提取多个次成分, 而且收敛速度优于一些现有算法.

关键词: 多个次成分; Möller算法; 加权矩阵; 神经网络

中图分类号: TP273 **文献标志码:** A

Modified Möller algorithm for multiple minor components extraction

GAO Ying-bin^{1,2}, KONG Xiang-yu^{1†}, HU Chang-hua¹, HOU Li-an¹

(1. Department of Control Engineering, the Second Artillery Engineer University, Xi'an 710025, China; 2. The Military Deputy Office of the Second Artillery in Shijiazhuang, Shijiazhuang 050081, China)

Abstract: Minor component analysis (MCA) is a powerful tool in the signal processing field. Up to now, there are few algorithms, which can extract multiple minor components from input signals, and many limitation conditions exist before using some existing algorithms. In order to solve these problems, the Möller algorithm, which can only extract one minor component, is modified into a multiple minor components extraction algorithm by using the weighted matrix method. The proposed algorithm has no limitation on the smallest eigenvalue and has a good convergence property, while no norm operation is needed. Simulation results show that the proposed algorithm can efficiently extract the multiple minor components of an input signal and has a faster convergence speed than some existing algorithms.

Keywords: multiple minor components; Möller algorithm; weighted matrix; neural networks

0 引言

在信息处理领域,通常将与输入向量自相关矩阵的前 r 个最小特征值所对应的特征向量称为次成分,而将由这个次成分张成的特征空间称为次子空间.次成分和次子空间已经在信息领域得到了广泛的应用,如总体最小二乘^[1]、多维曲线/曲面拟合^[2]和波达方向估计^[3]等.

近些年来,采用神经网络方法进行次成分/次子空间提取成为了一个研究热点,学者们相继提出了一大批优秀的次成分或次子空间分析算法,如Oja/Xu算法^[2]、Möller算法^[4]、OJAm算法^[3]、Kong算法^[1]等.虽然这些算法已经在实际应用中取得了很好的效果,但是仍有一些问题无法解决.例如,Oja/Xu算法和Möller算法只能提取一维次成分,而OJAm算法和Kong算法只能对次子空间进行跟踪.当需要提

取多个次成分的时候,这些算法就都失去了意义.

迄今为止,能够进行多个次成分提取的算法并不多,一些算法还存在很多限制条件,难以满足实际需要.例如,Oja+算法^[5]虽然可以提取多个次成分,但要求最小特征值不得大于1;文献[6]中的算法虽然没有特征值大小的限制,但是该算法不适用于几个最小特征值不相同的情况;文献[7]将Douglas次子空间跟踪算法改造成为多个次成分提取算法,虽然这一算法对于特征值没有要求,但是该算法需要预先选取一个合适的参数,而且参数的选取与算法的收敛性有着很大的关系;文献[8]采用主/次成分转换机制,提出了一种可以并行提取多个次成分的算法,虽然该算法不存在过多限制条件,但要事先对最小特征值进行估计.

到目前为止,学者们已经提出了很多次子空间跟踪算法^[1,3,9],能否将这些次子空间加以改进使其能

收稿日期: 2015-11-10; 修回日期: 2016-01-15.

基金项目: 国家杰出青年科学基金项目(61025014); 国家自然科学基金项目(61673387, 61374120, 61074072).

作者简介: 高迎彬(1986—),男,博士生,从事自适应信号处理的研究;孔祥玉(1967—),男,教授,博士生导师,从事系统特征提取、自适应信号处理等研究.

†通讯作者. E-mail: xiangyukong01@163.com

够提取多个次成分呢?沿着这一思路人们进行了探索.第1种思路是首先采用单个次成分提取算法提取第1个次成分,然后采用膨胀技术^[9],提取下一个次成分,重复以上步骤,即可以提取多个次成分,该方法称为序贯提取法;第2种思路是首先采用次子空间跟踪算法进行次子空间的提取,然后利用空间旋转的方法提取次成分.两类方法虽然都可以完成多个次成分的提取,但是都需要一个中间步骤,因此研究有没有一种可以直接提取多个次成分,而不要中间步骤的方法就显得很有意义.

文献[10-12]分别对几种主子空间算法进行了研究,采用加权矩阵的方法提出了一些多个主成分提取算法.因此,能否将加权矩阵的思想应用于次成分的提取成为了一个研究方向.到目前为止,这方面的文献并不多见.文献[13]提出了一种加权的次成分分析算法,但是发现该算法是不收敛的.为了确保算法的收敛性,文献[13]不得不在算法迭代过程中添加一些额外的模值限制措施.

基于对Möller次成分分析算法和加权矩阵思想的研究,本文提出一种可以同时提取多个次成分的算法.该算法可直接并行地提取多个次成分,而不需要中间步骤和额外的模值限制措施.相比一些现有算法,本文所提出的算法具有较快的收敛速度.

1 问题描述

考虑如下具有多维输入输出的神经网络:

$$\mathbf{y}_k = \mathbf{W}_k^T \mathbf{x}_k. \quad (1)$$

其中: $\mathbf{y}_k \in \mathbf{R}^{r \times 1}$ 为神经网络的输出; $\mathbf{W}_k \in \mathbf{R}^{n \times r}$ 为神经网络的权矩阵; $\mathbf{x}_k \in \mathbf{R}^{n \times 1}$ 为一个零均值随机过程,这里作为神经网络的输入; n 为输入向量的维数; r 为所需提取次成分的维数.令 $\mathbf{R} = E[\mathbf{x}_k \mathbf{x}_k^T]$ 为输入序列的自相关矩阵,对 \mathbf{R} 进行特征值分解可得

$$\mathbf{R} = \mathbf{U} \mathbf{\Lambda} \mathbf{U}^T. \quad (2)$$

其中: $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_n]$ 是由矩阵 \mathbf{R} 的特征向量构成的矩阵, $\mathbf{\Lambda} = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n)$ 由矩阵 \mathbf{R} 的特征值组成.为了方便使用,这里将特征值按照升序方式进行排列,即特征值满足

$$\lambda_n > \lambda_{n-1} > \dots > \lambda_r > \dots > \lambda_1 > 0. \quad (3)$$

由次成分定义可知,特征值 $\lambda_1, \lambda_2, \dots, \lambda_r$ 所对应的特征向量称为矩阵 \mathbf{R} 的前 r 次成分;而多个次成分提取算法的任务即是构造合适的神经网络权矩阵迭代更新方程,使神经网络的权矩阵最终能够收敛到

矩阵 \mathbf{R} 的前 r 次成分.

2 自适应提取算法

在文献[4]中,Möller提出了以下单维次成分提取算法:

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \eta \{ \mathbf{w}^T(k) \mathbf{R} \mathbf{w}(k) \mathbf{w}(k) - [2\mathbf{w}^T(k) \mathbf{w}(k) - 1] \mathbf{R} \mathbf{w}(k) \}. \quad (4)$$

其中: η 为神经网络的学习因子,且满足 $0 < \eta < 1$; $\mathbf{w}(k)$ 为一个 $n \times 1$ 维向量.然而该算法只能提取矩阵 \mathbf{R} 的第1个次成分,并不能进行次子空间跟踪,更不能进行多个次成分的提取.这里首先将其作适当调整,使其能够进行次子空间跟踪.为此,提出以下次子空间跟踪算法:

$$\mathbf{W}(k+1) = \mathbf{W}(k) + \eta \{ \mathbf{W}(k) \mathbf{W}^T(k) \mathbf{R} \mathbf{W}(k) - \mathbf{R} \mathbf{W}(k) [2\mathbf{W}^T(k) \mathbf{W}(k) - \mathbf{I}_r] \}. \quad (5)$$

其中: $\mathbf{W}(k)$ 为一个 $n \times r$ 维矩阵, \mathbf{I}_r 为一个 $r \times r$ 维单位阵.当式(5)所描述的神经网络收敛之后,权矩阵 $\mathbf{W}(k)$ 将收敛于由特征值 $\lambda_1, \lambda_2, \dots, \lambda_r$ 所对应特征向量张成空间的一组正交基.一般而言, $\mathbf{W}(k)$ 并不会收敛于矩阵 \mathbf{R} 的次成分.当需要对输入信号的多个次成分进行提取时,需进一步对算法(5)进行改进.这里提出如下多个次成分提取算法:

$$\mathbf{W}(k+1) = \mathbf{W}(k) + \eta \{ \mathbf{W}(k) \mathbf{W}^T(k) \mathbf{R} \mathbf{W}(k) \mathbf{A} - \mathbf{R} \mathbf{W}(k) [2\mathbf{W}^T(k) \mathbf{W}(k) \mathbf{A} - \mathbf{I}_r] \}. \quad (6)$$

其中:矩阵 $\mathbf{A} = \text{diag}(a_1, a_2, \dots, a_r)$ 为加权矩阵,且其对角线元素满足 $a_1 > a_2 > \dots > a_r$.显然,加权矩阵 \mathbf{A} 的选取方法与文献[10-13]中加权矩阵的选取方法一致.同时,加权矩阵的作用也与文献[10-13]中加权矩阵的作用一样,即:在每一步迭代过程中对神经网络的权矩阵施加Gram-Schmidt正交化(GSO)操作,使其能够收敛到信号自相关矩阵的次成分.由于加权矩阵 \mathbf{A} 选取比较简单而且限制条件较少,可便于算法的实际应用.

下面对加权矩阵的作用作进一步讨论.令算法(6)的学习步长为0,那么通过适当变化可得

$$\mathbf{A}^{-\frac{1}{2}} \mathbf{W}^T(k) \mathbf{W}(k) \mathbf{A}^{\frac{1}{2}} \approx \mathbf{I}. \quad (7)$$

假设

$$\mathbf{W}^T \mathbf{W} \mathbf{W}^T \mathbf{R} \mathbf{W} \approx \mathbf{W}^T \mathbf{R} \mathbf{W} \mathbf{W}^T \mathbf{W}.$$

该假设条件在 \mathbf{W} 接近于平稳点时通常是成立的,且广泛应用于各类算法的推导过程中^[1,9].为了方便计算,这里令 $\mathbf{W}^T \mathbf{W} = \mathbf{W}'$,则有

$$\mathbf{A}^{-\frac{1}{2}}\mathbf{W}'\mathbf{A}^{\frac{1}{2}} = \begin{bmatrix} w_{11} & \sqrt{a_2/a_1}w_{12} & \cdots & \sqrt{a_r/a_1}w_{1r} \\ \sqrt{a_1/a_2}w_{21} & w_{22} & \cdots & \sqrt{a_r/a_2}w_{2r} \\ \vdots & \vdots & \ddots & \vdots \\ \sqrt{a_1/a_r}w_{r1} & \sqrt{a_2/a_r}w_{r2} & \cdots & w_{rr} \end{bmatrix}, \quad (8)$$

其中 $w_{ij}(i, j = 1, 2, \dots, r)$ 为矩阵 \mathbf{W}' 的元素. 由式(8)可得: 矩阵 \mathbf{W}' 的上三角部分均是乘了一个小于1的数, 而下三角部分则是乘了一个大于1的数, 这与文献[5]中所提及的GSO方法是相一致的. 通过每一步迭代过程中的GSO操作, 神经网络权矩阵会收敛到输入信号的次成分, 这也正是引入矩阵 \mathbf{A} 的原因.

假设在时刻 $k = k_0$, 权矩阵收敛到了 $\mathbf{W} = \mathbf{U}'\mathbf{A}^{-\frac{1}{2}}$ (\mathbf{U}' 为矩阵 \mathbf{R} 的次成分组成的矩阵), 则根据矩阵特征值分解的性质可得

$$\mathbf{W}^T(k_0)\mathbf{R}\mathbf{W}(k_0) = \mathbf{A}\mathbf{A}', \quad (9)$$

$$\mathbf{R}\mathbf{W}(k_0) = \mathbf{U}'\mathbf{A}'\mathbf{A}^{\frac{1}{2}}, \quad (10)$$

$$\mathbf{W}^T(k_0)\mathbf{W}(k_0) = \mathbf{A}^{-1}, \quad (11)$$

其中 \mathbf{A}' 为对角线元素为矩阵 \mathbf{U}' 对应的特征值组成的对角矩阵. 将式(9)~(11)代入(6)可得

$$\mathbf{W}(k_0 + 1) = \mathbf{W}(k_0), \quad (12)$$

即当 $k \geq k_0$ 时, 神经网络算法(6)中的权矩阵将不再变化, 而在此时权矩阵 \mathbf{W} 正好收敛到矩阵 \mathbf{R} 的次成分, 因此认为算法(6)可提取矩阵 \mathbf{R} 的次成分.

通过将算法(6)和(5)进行比较易知, 式(6)仅多出了一个加权矩阵 \mathbf{A} . 当矩阵 $\mathbf{A} = \mathbf{I}_r$ 时, 算法(6)便退化为(5). 由此可知, 算法(6)是(5)的一种广义形式. 通常来讲, 在实时信号处理领域, 只能获得实时的输入向量 $\mathbf{x}(k)$, 而输入向量的自相关矩阵 \mathbf{R} 是未知的. 为此, 本文采用下式对矩阵 \mathbf{R} 进行实时估计:

$$\hat{\mathbf{R}}(k) = \frac{(k-1)}{k}\alpha\hat{\mathbf{R}}(k-1) + \frac{\mathbf{x}_k\mathbf{x}_k^T}{k}. \quad (13)$$

其中: α 为遗忘因子, 且满足 $0 < \alpha < 1$. 遗忘因子 α 的取值一般根据输入向量的变化快慢来决定. 一般而言, 对于快速变化的信号, α 应选择一个小一点的值; 而对于缓慢变化的信号, 则应该选择一个较大的遗忘因子. 由式(13)可得, 当 $k \rightarrow \infty$ 时, $\hat{\mathbf{R}}(k) \rightarrow \mathbf{R}$. 因此, 采用式(13)代替(6)中的 \mathbf{R} 是可行的. 为了使用方便, 这里将算法(6)称为加权 Möller 算法(简记为 WMöller 算法). 该算法的具体步骤如下:

Step 1: 初始化. 确定学习因子 η 和遗忘因子 α , 选取合适的加权矩阵 \mathbf{A} .

Step 2: 采用式(7)对自相关矩阵 \mathbf{R} 进行估计, 得

到 $\hat{\mathbf{R}}(k)$;

Step 3: 将式(6)中的 \mathbf{R} 用 $\hat{\mathbf{R}}(k)$ 替代, 然后利用式(6)对输入信号的次成分进行提取.

3 算法的自稳定性分析

自稳定特性对于神经网络算法极为重要. 自稳定特性是指在神经网络迭代过程中, 权矩阵的模值是确定的, 并且能够收敛于一个固定值, 而与初始化时权矩阵的选择无关. 由于缺乏自稳定特性的算法最终都是发散的, 有必要对 WMöller 算法的自稳定性进行研究.

定理 1 如果 WMöller 算法的学习因子 η 足够小, 且输入信号向量是有界的, 则 WMöller 算法中权矩阵的模值是有界的.

证明 由于学习因子 η 非常小, 且输入向量有界, 有

$$\begin{aligned} \|\mathbf{W}(k+1)\|_F^2 &= \text{tr}\{\mathbf{W}^T(k+1)\mathbf{W}(k+1)\} = \\ &\text{tr}\{\{\mathbf{W}(k) + \eta\{\mathbf{W}(k)\mathbf{W}^T(k)\mathbf{R}\mathbf{W}(k)\mathbf{A} - \\ &\mathbf{R}\mathbf{W}(k)[2\mathbf{W}^T(k)\mathbf{W}(k)\mathbf{A} - \mathbf{I}_r]\}\}^T \times \\ &\{\mathbf{W}(k) + \eta\{\mathbf{W}(k)\mathbf{W}^T(k)\mathbf{R}\mathbf{W}(k)\mathbf{A} - \\ &\mathbf{R}\mathbf{W}(k)[2\mathbf{W}^T(k)\mathbf{W}(k)\mathbf{A} - \mathbf{I}_r]\}\} = \\ &\text{tr}\{\mathbf{W}^T(k)\mathbf{W}(k) + 2\eta\mathbf{W}^T(k)\mathbf{R}\mathbf{W}(k) \times \\ &[\mathbf{I}_r - \mathbf{W}^T(k)\mathbf{W}(k)\mathbf{A}] + \\ &\eta^2[(\mathbf{W}^T(k)\mathbf{R}\mathbf{W}(k))^2\mathbf{W}^T(k)\mathbf{W}(k)\mathbf{A}^2 - \\ &2(\mathbf{W}^T(k)\mathbf{R}\mathbf{W}(k))^2[2\mathbf{A}\mathbf{W}^T(k)\mathbf{W}(k)\mathbf{A} - \mathbf{A}] + \\ &\mathbf{W}^T(k)\mathbf{R}^2\mathbf{W}(k)[4(\mathbf{W}^T(k)\mathbf{W}(k))^2 - \\ &4\mathbf{W}^T(k)\mathbf{W}(k)\mathbf{A} + \mathbf{I}_r]]\} \approx \\ &\text{tr}\{\mathbf{W}^T(k)\mathbf{W}(k) + 2\eta\mathbf{W}^T(k)\mathbf{R}\mathbf{W}(k) \times \\ &[\mathbf{I}_r - \mathbf{W}^T(k)\mathbf{W}(k)\mathbf{A}]\}. \end{aligned}$$

上式省去了有关学习因子的二阶项. 通过计算前后两个时刻模值的变化情况可得

$$\begin{aligned} \|\mathbf{W}(k+1)\|_F^2 / \|\mathbf{W}(k)\|_F^2 &\approx \\ 1 + 2\eta \frac{\text{tr}\{\mathbf{W}^T(k)\mathbf{R}\mathbf{W}(k)[\mathbf{I}_r - \mathbf{W}^T(k)\mathbf{W}(k)\mathbf{A}]\}}{\text{tr}\{\mathbf{W}^T(k)\mathbf{W}(k)\}} & \\ \begin{cases} > 1, \text{tr}[\mathbf{W}^T(k)\mathbf{W}(k)] > \text{tr}(\mathbf{A}^{-1}); \\ = 1, \text{tr}[\mathbf{W}^T(k)\mathbf{W}(k)] = \text{tr}(\mathbf{A}^{-1}); \\ < 1, \text{tr}[\mathbf{W}^T(k)\mathbf{W}(k)] < \text{tr}(\mathbf{A}^{-1}). \end{cases} & \quad (14) \end{aligned}$$

由式(14)可得, 权矩阵的模值将趋近于一个常值, 即不论初始权矩阵如何选择, 权矩阵的模值是收敛且有界的, 也就是说算法(6)具有自稳定性. \square

4 仿真实验

下面将提供3个实验来验证所提出算法的性能.为了衡量算法的性能,本文引入方向余弦和权向量模值两个评价函数.方向余弦

$$\text{DirectionCosine}(k) = \frac{\|\mathbf{w}_i^T(k)\mathbf{u}_i\|}{\|\mathbf{w}_i(k)\| \cdot \|\mathbf{u}_i\|} \quad (15)$$

其中: $i = 1, 2, \dots, r$, 且 \mathbf{w}_i 代表矩阵 $\mathbf{W}(k)$ 的第 i 列.方向余弦可以衡量算法的收敛精度,而权向量模值则可以衡量算法的收敛性能,即

$$\text{Norm}_i(k) = \|\mathbf{w}_i(k)\|, \quad i = 1, 2, \dots, r. \quad (16)$$

4.1 多个次成分提取

输入向量采用以下模型产生:

$$x(k) = 0.75x(k-1) + e(k), \quad (17)$$

其中 $e(k)$ 为一个均值为零方差为1的高斯白噪声.取模型(17)中10个连续不重叠的输出作为神经网络的输入向量.接下来采用WMöller算法对其进行多个次成分的提取.

在此次实验中,本文提取4个次成分,即 $r = 4$.参数设置为:加权矩阵 $\mathbf{A} = \text{diag}(4, 3, 2, 1)$,学习因子 $\eta = 0.1$,遗忘因子 $\alpha = 0.998$.初始权矩阵为随机产生,且将其模值标准化为1.仿真结果如图1和图2所示.

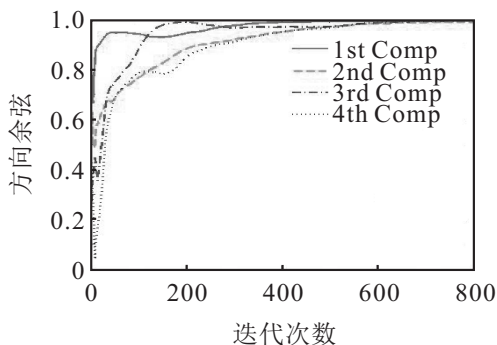


图1 WMöller算法的方向余弦

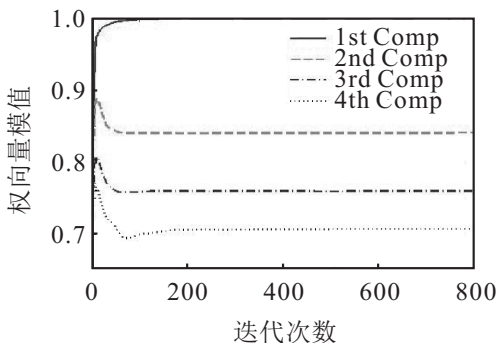


图2 WMöller算法的权向量模值

由图1可知,大约经历700次的迭代后,方向余弦均收敛到1,也就是说WMöller算法的权矩阵已收敛到了几个次成分的方向;由图2可知,此时权矩阵的模值已经收敛,这说明WMöller算法可有效地并行提

取输入向量的多个次成分.

4.2 自稳定性实验

文献[8]中的自相关矩阵

$$\mathbf{R} = \begin{bmatrix} 0.1681 & -0.0664 & 0.0111 \\ -0.0664 & 1.3960 & -0.0165 \\ 0.0111 & -0.0165 & 1.1345 \\ 0.0237 & -0.0187 & -0.0262 \\ 0.1064 & 0.0311 & 0.0051 \\ -0.0262 & 0.0170 & -0.0016 \\ 0.0237 & 0.1064 & -0.0262 \\ -0.0187 & 0.0311 & 0.0170 \\ -0.0262 & 0.0051 & -0.0016 \\ 1.0156 & -0.0045 & 0.0213 \\ -0.0045 & 0.9240 & -0.0700 \\ 0.0213 & -0.0700 & 1.9574 \end{bmatrix} \quad (18)$$

通过特征值分解可得到上述矩阵的特征值为:1.9637, 1.4026, 1.1397, 1.0091, 0.9321, 0.1487.然后,采用WMöller算法对其前4个特征值进行提取.算法的参数选择均与4.1节相同,即:学习因子 $\eta = 0.1$,加权矩阵 $\mathbf{A} = \text{diag}(4, 3, 2, 1)$.唯一不同的是:初始化权矩阵的模值分别取为1,1.36和2三个不同的值.这里主要观察神经网络权矩阵模值的收敛情况.图3为不同初始权矩阵情况下权向量模值的仿真结果.

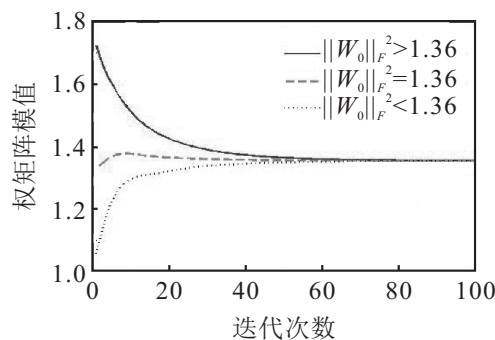


图3 不同初始化条件下的权向量模值

由图3可以看出,不论初始化权矩阵如何选择,权矩阵的模值将最终收敛到一个常值,这与定理1中的分析是吻合的,同时也说明WMöller算法具有自稳定性.

4.3 算法性能对比

文献[8]中提出了一种可以并行提取次成分的算法——AMMD算法,并对其收敛性能进行了详细的研究.下面将所提出的WMöller算法与AMMD算法进行对比.为了确保对比的公平性,同时采用两种算法对4.2节所使用矩阵的前3个次成分进行提取.两

个算法的初始化参数设置如下:AMMD算法中,常值 $\mu = 2$,学习因子 $\eta = 0.1$;WMöller算法中,学习因子 $\eta = 0.1$,加权矩阵 $\mathbf{A} = \text{diag}(3, 2, 1)$.两种算法的初始化矩阵/向量均是随机产生的,且标准化为1.两种算法的仿真结果如图4和图5所示.

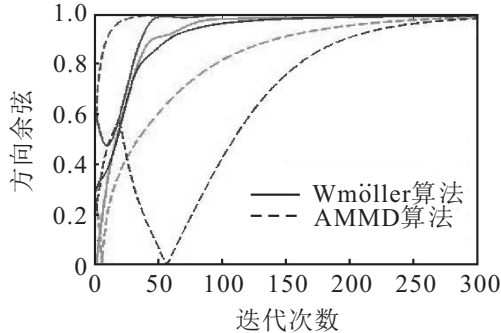


图4 两种算法的方向余弦

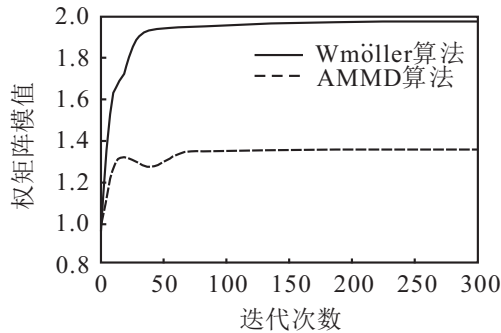


图5 两种算法的权矩阵模值

由图4可见,虽然AMMD算法可很快地提取第1个次成分,但其第2、第3个次成分的提取速度要稍逊于本文算法.图5中AMMD算法的权矩阵是指将迭代过程中的3个权向量进行组合形成的矩阵,具体可以参见文献[8].由图5可见,WMöller算法权矩阵模值的收敛速度要优于AMMD算法.综上所述,WMöller算法的收敛速度优于AMMD算法.

5 结 论

在对Möller算法进行详细研究的基础上,本文提出了一种加权的Möller算法.相对比只能提取一维次成分的Möller算法,加权的Möller算法可并行提取多个次成分,而不需要额外的模值限制措施.理论分析表明,该算法具有自稳定性.仿真实验结果表明,该算法具有很好的收敛性和很快的收敛速度.

参考文献(References)

- [1] Kong Xiangyu, Hu Changhua, Han Chongzhao. A self-stabilizing MSA algorithm in high-dimension data stream[J]. *Neural Networks*, 2010, 23(7): 865-871.
- [2] Lei Xu, Erkki Oja, Ching Y Suen. Modified Hebbian learning for curve and surface fitting[J]. *Neural Networks*, 1992, 5(3): 441-457.
- [3] Feng Dazheng, Zheng Weixing, Jia Ying. Neural network learning algorithms for tracking minor subspace in high-dimensional data stream[J]. *IEEE Trans on Neural Networks*, 2005, 16(3): 513-521.
- [4] Ralf Möller. A self-stabilizing learning rule for minor component analysis[J]. *Int J of Neural Systems*, 2004, 14(1): 1-8.
- [5] Erkki Oja. Principal components, minor components and linear neural networks[J]. *Neural Networks*, 1992, 5(6): 927-935.
- [6] Mathew G, Reddy V. Orthogonal eigensubspace estimation using neural networks[J]. *IEEE Trans on Signal Processing*, 1994, 42(1): 1803-1811.
- [7] Marko V Jankovic, Branimir Reljin. A new minor component analysis method based on Douglas Kung Amari minor subspace analysis method[J]. *IEEE Signal Processing Letters*, 2005, 12(2): 859-862.
- [8] Tan Kok Kiong, Lv Jiancheng, Zhang Yi. Adaptive multiple minor directions extraction in parallel using a PCA neural network[J]. *Theoretical Computer Science*, 2010, 411(48): 4200-4215.
- [9] Lou Fanglong Luo, Unbehauen R. A minor subspace analysis algorithm[J]. *IEEE Trans on Neural Networks*, 1997, 8(5): 1149-1155.
- [10] Oja E, Ogawa H, Wangviwattana J. Principal component analysis by homogeneous neural networks, Part I: Weighted subspace criterion[J]. *IEICE Trans on Information and Systems*, 1992, 75(3): 366-375.
- [11] Shan Ouyang, Zheng Bao. Fast principal component extraction by a weighted information criterion[J]. *IEEE Trans on Signal Processing*, 2002, 50(8): 1994-2002.
- [12] Toshihisa Tanaka. Generalized weighted rules for principal components tracking[J]. *IEEE Trans on Signal Processing*, 2005, 53(4): 1243-1253.
- [13] Bartelmaos S. Fast adaptive algorithms for minor component analysis using Householder transformation[J]. *Digital Signal Processing*, 2011, 21(6): 667-678.

(责任编辑:滕 蓉)