doi:10. 11920/xnmdzk. 2020. 01. 011

非负矩阵 Perron 根与主特征向量的粒子迭代算法

曾　莉,肖　明

(西南民族大学计算机科学与技术学院,四川 成都　610041)

摘　要:非负矩阵 Perron 根问题在数值分析、经济学及控制论等多方面有着重要的应用,如何快速求出非负矩阵 Perron 根与主特征向量,一直是矩阵理论研究的热点. 结合矩阵 Perron 根和主特征向量的特性,给出基于粒子迭代的快速寻优算法,同时给出该方法的收敛性的证明,最后通过数值实验说明了该方法的优越性.

关键词:非负矩阵;Perron 根;特征向量;粒子群算法

中图分类号:O24　　**文献标志码:**A　　**文章编号:**2095-4271(2020)01-0071-06

Particle iterative algorithm for non-negative matrix Perron root and principal eigenvector

ZENG Li, XIAO Ming

(School of Computer Science and Technology, Southwest Minzu University , Chengdu 610041, P. R. C.)

Abstract: Perron root problem of non – negative matrix has important applications in numerical analysis, economics and cybernetics, etc. How to quickly find the values of Perron root of non – negative matrix and corresponding eigenvectors has always been the focus of matrix theory research. Combining the characteristics of Perron root of matrix and principal eigenvector, a fast optimization algorithm based on particle iteration is given. At the same time, the convergence of the method is proved. Finally, the advantages of the method are illustrated by numerical experiments.

Key words: non – negative matrix; Perron root; feature vector; particle swarm optimization algorithm

对矩阵 $A \in C^{n\times n}$,则称 $\rho(A) = \max\{|\lambda|:\lambda \in \lambda(A)\}$ 为 A 的谱半径. 由 Perron – Frobenius 定理[1]可知,对非负矩阵 $A = (a_{ij}) \in R^{n\times n}$, $a_{ij} \geq 0$,有 $\rho(A) > 0$,且 $\rho(A)$ 的值等于 A 的最大特征值,此时,称 $\rho(A)$ 为 A 的 Perron 根. 非负矩阵 Perron 根在数值分析、经济学及控制论等各个方面有着重要的应用,如大型桥梁或建筑物的振动问题;机械和机件的振动问题;飞机机翼的颤振问题等.

在非负矩阵 Perron 根的研究中,最著名的是 G. Frobenius[2]不等式:

$$\min_i r_i(A) \leq \rho(A) \leq \max_i r_i(A) , \tag{1}$$

这里 $r_i(A)$ $(i = 1, 2, \cdots, n)$ 表示矩阵 A 的第 i 行行和,简写为 r_i .

对于有非零行和的非负矩阵,H. Minc[3]对此作出进一步改进,将 Perron 根的取值范围缩小为:

$$\min_i\left(\frac{1}{r_i}\sum_{t=1}^{n} a_{it}r_i\right) \leq \rho(A) \leq \max_i\left(\frac{1}{r_i}\sum_{t=1}^{n} a_{it}r_i\right) .$$

收稿日期:2019-09-27

作者简介:曾莉(1974 –),女,汉族,重庆涪陵人,副教授,研究方向:矩阵计算、信息处理. E – mail: Lzeng@ 126. com

基金项目:西南民族大学中央高校基本科研业务费专项资金项目资助(2014NZYQN33)

T. Zh. Huang 和曾莉[4]引入非负矩阵 $B = A + \alpha I$ ($\alpha > 0$)，对任意正整数 m，k，有：

$$\min_i \left(\frac{r_i(A^m B^k)}{r_i(B^k)}\right)^{\frac{1}{m}} \le \rho(A) \le \max_i \left(\frac{r_i(A^m B^k)}{r_i(B^k)}\right)^{\frac{1}{m}}.$$

文献[5－6]给出了系列算法,利用上述公式通过迭代来计算非负矩阵 Perron 根,这些算法方法简便,准确性高,运算速度较快,但仅考虑了 Perron 根的计算. 而在一些工程应用中,我们还希望同时求出 Perron 根及其对应的主特征向量,此时,可以应用常规的求特征值、特征向量的方法[7],如：LR 方法、QR 方法、幂法和反幂法、雅可比法,然后从中挑选出最大的特征值及对应的特征向量即可. 但这些方法往往存在存储量大,运算精度低,收敛速度慢等问题. 曾莉[8]提出了针对实对称矩阵的子空间迭代算法,该算法能快速迭代出实对称矩阵的所有特征值与特征向量,但该算法仅适用实对称矩阵,具有一定的局限性. 本文使用粒子群优化算法来求解非负矩阵的 Perron 根及对应的主特征向量.

1 粒子群 PSO 算法

粒子群(Particle Swarm Optimization, PSO)算法[9－14],由 Kennedy 和 Eberhart 于 1995 年提出,是一种基于种群搜索的自适应进化计算技术. PSO 算法模拟鸟群觅食活动,将求解问题的潜在解看作为鸟群中的一只鸟,即 PSO 中的粒子,问题的最优解对应食物. 类似鸟群在寻找食物的过程中,不停改变自己飞行位置和速度的行为,每个粒子均有位置和速度特性,还有一个由目标函数决定的适应值,代表离食物的距离. 每个粒子根据自身和同伴的飞行经验动态调整位置,标准 PSO 算法速度和粒子位置的更新公式如下：

$$v_i = w * v_i + c_1 * rand() * (pbest_i - x_i) + c_2 * rand() * (gbest - x_i), \tag{2}$$

$$x_i = x_i + v_i. \tag{3}$$

这里, x_i 和 v_i 代表第 i 个粒子的位置和速度, w 代表惯性权值, c_1 和 c_2 表示学习参数, $rand()$ 表示在 0－1 之间的随机数, $pbest_i$ 代表第 i 个粒子搜索到的最优值, $gbest$ 代表整个集群搜索到的最优值.

2 直接应用标准 PSO 求解 Perron 根及主特征向量

2.1 粒子表示形式

对矩阵求特征值和特征向量,可以如文献[15]一样,先迭代出特征值,再以此特征值为基础迭代出特征向量. 为了充分体现粒子迭代的优点,本文先尝试将两者统一封装在一个粒子中,即每个粒子 i 的位置 x_i 和速度 v_i 都用一个 $n+1$ 维向量来表示,前 n 个分量表示特征向量的值,第 $n+1$ 个分量表示特征值：

$$(x_i, v_i) = ((x_{i1}, x_{i2}, \cdots, x_{in}, x_{i\lambda}), (v_{i1}, v_{i2}, \cdots, v_{in}, v_{i\lambda})). \tag{4}$$

2.2 适应度函数设计

由特征值特征向量的性质可知：$Ax = \lambda x$,即 $(A - \lambda I)x = 0$,将粒子的个体表达式代入,可得：$(A - x_{i\lambda} I)(x_{i1}, x_{i2}, \cdots, x_{in})^T = 0$. 设：

$$y_i = (y_{i1}, y_{i2}, \cdots y_{in}) = (A - x_{i\lambda} I)(x_{i1}, x_{i2}, \cdots x_{in})^T,$$

由此,可定义迭代算法的适应度函数为：

$$\min f = \left| \sum_{j=1}^{n} y_{ij}^{\ 2} \right|. \tag{5}$$

2.3 求解算法

算法 1 直接使用粒子迭代算法求解非负矩阵的 Perron 根和主特征向量.

步骤 1：设定算法结束条件 ε 的值和算法迭代的最大步数 MAXLOOP,计算给定矩阵 A 的各行行和,按(1)式求出 Perron 根 $v_{i\lambda}$ 的上、下限. 随机生成 x 个粒子的初始位置和初始速度 (x_i, v_i) ,分别对 $(x_{i1}, x_{i2}, \cdots, x_{in})$ 、

$(v_{i1},v_{i2},\cdots,v_{in})$ 做规范化处理,并限定 $x_{i\lambda}$ 在 Perron 根的上、下限范围内，$v_{i\lambda}$ 取值范围限定于 $\pm 0.2*(\max(r_i)-\min(r_i))$ 之间.

步骤 2:按(5)式计算各粒子的适应度,更新 $pbest_i$ 和 $gbest$,若 $gbest$ 的适应度小于 ε ,则输出 $gbest$ 中的 Perron 根与对应特征向量,算法结束;否则,执行步骤 3.

步骤 3:按(2)式更新粒子的速度参数,这里惯性 w 取值 0.6,学习参数 c_1 和 c_2 均取 0.4,再按(3)式更新粒子的位置参数.

步骤 4:重复执行步骤 2 和步骤 3,若重复次数已到上限 MAXLOOP,则算法结束.

2.4　算法实践及分析

将算法用 VC + + 编程实现后,发现多数测试矩阵都不能得到满意的解. 分析原因,存在两方面的问题:一是对任一 n 维非负矩阵,都有 n 组可重复的特征值及对应特征向量,每组特征值、特征向量都会满足(5)式的适应度函数,虽然算法 1 中第 1 步对 Perron 根 $v_{i\lambda}$ 的上、下限做了限定,仍不能保证完全排除其它特征值出现;另一方面,即使将 $v_{i\lambda}$ 限定为 Perron 根的正确解,使用(2)、(3)式迭代出的粒子也不具备收敛性,不能保证每一轮迭代后全局最优粒子用(5)式的计算适应度值一定更优.

3　改进的非负矩阵主特征向量与 Perron 根的粒子迭代算法

3.1　新的迭代公式

由幂法思想可知,对任意的非零初始向量 $v_0\in R^n$,通过非负矩阵 A 的不断作用,能得到一个向量序列:

$$\begin{cases} v_1 = Av_0, \\ v_2 = Av_1 = A^2v_0, \\ \quad\vdots \\ v_{k+1} = Av_k = A^{k+1}v_0, \\ \quad\vdots \end{cases}$$

不妨设非负矩阵 A 的特征值为 $\lambda_1,\lambda_2,\cdots,\lambda_n$, $|\lambda_1|>|\lambda_2|\geq|\lambda_3|\geq\cdots\geq|\lambda_n|$,对应的特征向量为 $\chi_1,\chi_2,\cdots,\chi_n$, v_0 在 $\chi_i(i=1,2,\cdots n)$ 下分解为 $v_0=\sum_{i=1}^{n}a\chi_i\ (i=1,2,\cdots n)$.

于是　$v_k=Av_{k-1}=A^kv_0=\sum_{i=1}^{n}a_i\lambda_i^k\chi_i$, 当 $a_1\neq 0$ 时,

$v_k=\sum_{i=1}^{n}a_i\lambda_i^k\chi_i=a_1\lambda_1^k\chi_1+a_2\lambda_2^k\chi_2+\cdots+a_n\lambda_n^k\chi_n=\lambda_1^k[a_1\chi_1+\sum_{i=2}^{n}a_i(\lambda_i/\lambda_1)^k\chi_i]=\lambda_1^k(a_1\chi_1+\varepsilon_k)$ 显然,当 k 充分大时有 $v_k\approx\lambda_1^ka_1\chi_1$, $v_{k+1}\approx\lambda_1^{k+1}a_1\chi_1\approx\lambda_1v_k$. 所以, $\lambda_1\approx v_{k+1}/v_k$. 又 $v_{k+1}=Av_k\approx\lambda_1v_k$,由此, v_k 可作为 λ_1 的近似特征向量.

由以上分析,可知任一非零向量 v_0 在矩阵 A 的作用下,会逐渐向 A 的主特征向量靠近,由此,我们在算法 2 中,粒子修改为仅表示特征向量,其飞行惯性修改为用矩阵 A 乘粒子向量. 显然,向量 v_0 会收敛于矩阵 A 主特征向量,从而,每个粒子的速度和粒子位置的更新公式修改如下:

$$v_i = w*x_i + r_1*pbest_i + r_2*gbest\ ; \tag{6}$$

$$x_i = Av_i\ . \tag{7}$$

当主特征向量确定后,即可用以下公式(8)计算 Perron 根的值:

$$p(A) = \frac{|x|}{|v|} = \sqrt{\sum_{i=1}^{n}x^2}\Big/\sqrt{\sum_{i=1}^{n}v^2}. \tag{8}$$

3.2 新的适应度函数

由(5)式表示的适应度函数中,同时会用到 Perron 根和主特征值特征向量的迭代值,两者都会影响适应度函数的结果,而由上面的分析可知,(7)式表示的粒子向量会不断收敛于主特征向量,由此,我们可用 1 减粒子迭代前后的夹角余弦的差作为新的适应度函数. 即:

$$\min f = 1 - \cos(v,x) = 1 - \frac{v \cdot x}{|v||x|} = 1 - \frac{\left|\sum_{i=1}^{n} v_i \cdot x_i\right|}{\sqrt{\sum_{i=1}^{n} v_i^{\ 2}} \cdot \sqrt{\sum_{i=1}^{n} x_i^{\ 2}}} \tag{9}$$

3.3 改进的算法

算法 2:利用改进的粒子迭代公式(6)、(7)求非负矩阵的主特征向量和 Perron 根的值.

步骤 1:设定算法结束条件 ε 的值和算法迭代的最大步数 MAXLOOP. 随机生成 x 个粒子的初始速度 $(v_{i1}, v_{i2}, \cdots, v_{in})$,对每个粒子做规范化处理后,再利用(7)求出各粒子的初始位置 $(x_{i1}, x_{i2}, \cdots, x_{in})$.

步骤 2:按(9)式计算各粒子的适应度,更新 $pbest_i$ 和 $gbest$,若 $gbest$ 的适应度小于 ε ,则 $gbest$ 中即为所求的主特征向量,再按公式(8)计算出 Perron 根的值,算法结束;否则,执行步骤 3.

步骤 3:按(6)式更新粒子的速度参数,规范化处理后再按(7)式更新粒子的位置参数.

步骤 4:重复执行步骤 2 和步骤 3,若重复次数已到上限 MAXLOOP,则算法结束.

3.4 算法收敛性分析

不妨设,目标矩阵 A 的特征值为 $\lambda_1, \lambda_2, \cdots, \lambda_n$, $|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \cdots \geq |\lambda_n|$,对应的规范化特征向量为 $\chi_1, \chi_2, \cdots, \chi_n$, $\rho(A) = |\lambda_1|$, χ_1 是主特征向量. 一个粒子 j 在第 k 轮的位置向量 $x_j^{\ (k)}$ 规范化后在 $\chi_i (i = 1,2, \cdots n)$ 下分解分别为:

$$x_j^{\ (k)} = \sum_{i=1}^{n} a_i \chi_i, \ (i = 1,2,\cdots n), \left(\sum_{i=1}^{n} a_i^{\ 2} = 1\right).$$

假设此时,粒子 $x_j^{\ (k)}$ 对应的 $pbest_j$ 在 $\chi_i (i = 1,2, \cdots n)$ 下分解为:

$$pbest_j^{\ (k)} = \sum_{i=1}^{n} b_i \chi_i, \ (i = 1,2,\cdots n), \left(\sum_{i=1}^{n} b_i^{\ 2} = 1\right).$$

全局最优粒子 $gbest^{(k)}$ 在 $\chi_i (i = 1,2, \cdots n)$ 下分解为:

$$gbest^{(k)} = \sum_{i=1}^{n} c_i \chi_i, \ (i = 1,2,\cdots n), \left(\sum_{i=1}^{n} c_i^{\ 2} = 1\right).$$

由定义可知, $c_1 >= b_1 >= a_1$. 由式(6)可得

$$v_j^{\ (k+1)} = \sum_{i=1}^{n} d_i \chi_i, \ d_i = wa_i + r_1 b_i + r_2 c_i, \ (i = 1,2,\cdots n).$$

$v_j^{\ (k+1)}$ 做规范化后为: $\bar{v}_j^{\ (k+1)} = \sum_{i=1}^{n} e_i \chi_i$,

$$e_i = d_i / \left|\sqrt{\sum_{t=1}^{n} d_t^{\ 2}}\right| = (wa_i + r_1 b_i + r_2 c_i) / \left|\sqrt{\sum_{t=1}^{n} (wa_t + r_1 b_t + r_2 c_t)^2}\right|, \left(\sum_{i=1}^{n} e_i^{\ 2} = 1\right),$$

再由(7)式可得, $x_j^{\ (k+1)} = \sum_{i=1}^{n} e_i \lambda \chi_i$,此时,

$$\cos(x_j^{(k+1)},\chi_1) = \frac{x_j^{(k+1)}\cdot\chi_1}{|x_j^{(k+1)}||\chi_1|} = \frac{|e_1\lambda_1|}{\sqrt{\sum_{i=1}^{n} e_i^2\lambda_i^2}} > \frac{|e_1\lambda_1|}{\sqrt{\sum_{i=1}^{n} e_i^2\lambda_1^2}} = |e_1|$$

$$= \frac{|wa_1 + r_1b_1 + r_2c_1|}{\sqrt{\sum_{t=1}^{n}(wa_t + r_1b_t + r_2c_t)^2}} \geq \frac{|wa_1 + r_1a_1 + r_2a_1|}{\sqrt{\sum_{t=1}^{n}(wa_t + r_1b_t + r_2c_t)^2}}$$

$$\geq \frac{|\mathrm{w} + r_1 + r_2||a_1|}{\sqrt{(\sum_{t=1}^{n} wa_t + \sum_{t=1}^{n} r_1b_t + \sum_{t=1}^{n} r_1c_t)^2}} = \frac{|\mathrm{w} + r_1 + r_2||a_1|}{|w\sum_{t=1}^{n} a_t + r_1\sum_{t=1}^{n} b_t + r_2\sum_{t=1}^{n} c_t|}$$

$$= \frac{|\mathrm{w} + r_1 + r_2||a_1|}{|\mathrm{w} + r_1 + r_2|} = |a_1|.$$

$$\cos(x_j^{(k)},\chi_1) = \frac{x_j^{(k)}\cdot\chi_1}{|x_j^{(k)}||\chi_1|} = |a_1|.$$

故, $\cos(x_j^{(k+1)},\chi_1) > \cos(x_j^{(k)},\chi_1)$,说明粒子 j 在第 k+1 轮比 k 轮与主特征向量的夹角更小,即每个粒子都不断地向主特征向量趋近.

4　数值实验

例、求矩阵 $A = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 3 & 1 \\ 0 & 1 & 4 \end{bmatrix}$ 的 perron 根与主特征向量.

本例设定 $\varepsilon = 10^{-6}$,粒子数为 3,选择单位矩阵的 3 个列向量为初始粒子,设 $w = 0.6$, $r_1 = 0.4$、$r_2 = 0.4$,初始最优粒子为全 1 向量. 通过 C++编程实现算法,程序运行结果如表 1 所示:

表 1　主特征向量与 Perron 根的迭代情况

Table 1　Iteration of Principal Eigenvector andPerron Root

迭代轮数	粒子编号	粒子向量			适应值
1	X1	0.617389	0.336758	0.710933	0.150034
	X2	0.693103	0.198030	0.693103	0.449177
	X3	0.710933	0.336758	0.617389	0.150034
2	X1	0.672633	0.332801	0.660915	0.00278468
	X2	0.665364	0.345035	0.661998	0.00645117
	X3	0.664281	0.332810	0.669304	0.000522838
3	X1	0.666217	0.333398	0.667084	1.54123e-005
	X2	0.666863	0.332305	0.666984	4.85008e-005
	X3	0.666964	0.333398	0.666336	8.17399e-005
4	X1	0.666696	0.333325	0.666641	6.42380e-008
	X2	0.666638	0.333423	0.666650	3.67806e-007
	X3	0.666629	0.333325	0.666708	1.27720e-007
主特征向量及 perron 根		0.666696	0.333325	0.666641	8.00000

从表1可以看出,每个粒子的每轮迭代适应值都在不断变优,4轮后即收敛到满足要求的解.

5 结束语

本文尝试将粒子迭代算法用于求解非负矩阵的 Perron 根与主特征向量,数值试验表明若直接使用粒子算法原有迭代公式,则迭代效果较差.本文从幂法思想出法,对迭代公式进行了修改,并证明了修改后迭代算法的收敛性,最后通过数值实验说明了算法的有效性.

参考文献

[1]徐树方. 矩阵计算的理论与方法[M]. 北京:北京大学出版社,1995.

[2]FROBENIUS G. Über Matrizen aus nichtnegativen Elementen[M] Stizungsber. Kön. uss. Akad. Wiss. Berlin, 1912:465 - 477.

[3] MINC H. Nonnegative Matrices[M]. New York:Wiley,1988:11 - 19,24 - 36.

[4]HUANG T ZH , ZENG L. Bounds for the Perron roots of nonnegative matrices[J]. Far East Journal of Applied Mathematics,2006,23(1):113 - 120.

[5]曾莉,肖明,杨军,等. 非负不可约矩阵 Perron 根的一种迭代算法[J].安徽大学学报(自然科学版),2013,37(3):31 - 37.

[6]曾莉,肖明.非负不可约矩阵 Perron 根的迭代算法研究[J].科技通报,2013,29(9):4 - 6.

[7]李庆扬, 王超能, 易大义. 数值分析[M].4 版. 武汉: 华中科技大学出版社, 2006: 219 - 245.

[8]曾莉,肖明.计算实对称矩阵特征值特征向量的幂法[J].南昌大学学报(理科版),2016,40(4):399 - 402.

[9] KENNEDY J, EBERHART R C. Swarm Intelligence[M]. Morgan Kaufmann Publishers,SanFrancisco,CA,2001.

[10]SHI Y H, EBERTHART R C. A modified particle swarm optimizer[C]. Proceedings of the International Joint Conference on Evolutionary Computation,1998:69 - 73.

[11]KENNEDY J, EBERTHART R C. A discrete binary version of the particle swarm optimization algorithm[C]//Proceeding of International Conference on System,Man and Cybemetics,1997:4104 - 4109.

[12]YOSHIDA H,FUKUAMA Y,TAKAYAMA S A. A Particle Swarm Optimization for Reactive Power and Voltage Control in Electric Power System Considering Voltage Security Assessment[C]//IEEE International Conference on System,Man and Cybemetics,1999(6):497 - 502.

[13]KRINK T, VESTERSTROM J S, RIGET J. Particle swarm optimization with spatial particle extension[C]//Proceeding of International Conference on Evolution ComPutation,2002:1474 - 1497.

[14]KAZEMI B AL, MOHAN C K. Multi - phase generalization of the particle swarm optimization algorithm[C]//Proceeding IEEE Conference on Evolutionary ComPutation. 2002:489 - 494.

[15]韦杏琼,周永权.求解矩阵特征值和特征向量的 PSO 算法[J].计算机工程,2010,36(3):189 - 191.

(责任编辑:张阳,付强,李建忠,罗敏;英文编辑:周序林)