

生存迁移分析及其应用

李豆豆¹, 邵世煌¹, 王 晴²

(1. 东华大学信息科学与技术学院, 上海 201620; 2. 贾汪区职教中心计算机教研室, 徐州 221011)

摘要: 通过模拟自然界生物的生存迁移过程建立一种自适应概率算法——生存迁移算法(LMA)。它模拟了生物通过比较生存状况和生存极限确定是否需要迁移, 进而寻求更好生存状况的机制。生存状况使生物逐步了解未知的生存空间, 生存极限使生物具有判断的标准, 迁移使生物具有寻求生存的能力。给出 LMA 的机理和模型, 分析 LMA 求解全局优化问题的收敛性。实验结果表明 LMA 在应用中具有较好的收敛性。

关键词: 生物迁移; 全局优化; 收敛性; 概率

Living Migration Analysis and Its Application

LI Dou-dou¹, SHAO Shi-huang¹, WANG Qing²

(1. College of Information Science and Technology, Donghua University, Shanghai 201620;

2. Computer Staff Room, Jiawang Vocational Education Center, Xuzhou 221011)

【Abstract】 A new adaptive-probability algorithm for global optimization, Living Migration Algorithm(LMA), is proposed by simulation of bio-subsistence and bio-migration in this paper. It simulates the mechanism that biology decides whether to migrate by comparing the standard of living with the minimal life-needs for searching better life. Unknown search space can be acquainted gradually by means of the standard of living, and biology can judge whether to migrate in terms of the minimal life-needs for seeking better living space. The mechanism and the model of LMA are given. And also given are convergence quality of LMA dealing with the global optimizations in terms of probability. Experimental results show that LMA has perfect convergence.

【Key words】 bio-migration; global optimization; convergence; probability

1 概述

达尔文在《物种起源》一书中系统阐述了生物迁移命题。他认为一个物种只能有一个起源中心, 在条件允许下, 它会从这个中心迁移到力所能及的地区。它和演化构成了物种起源理论的 2 个基本命题, 发展成为生物进化的 2 个基本方向。它如同演化, 在达尔文力的作用下时刻进行着, 不仅是生物自身的一种能力, 而且是环境给予的一种机会^[1]。

生物迁移主要分为: 扩张, 转移和隔离。绝大多数单一类群的迁移属于转移, 它指一个类群沿一定方向从一个地域向另一个地域一步一步地迁移, 到达新地域后, 老地域不再存在此类群。它在时空上是连续的, 但在特定时间内, 又仅分布在特定地域, 因此它是不连续的, 孤立分布在空间中^[2]。

生物为了生存和繁衍后代, 在演化和迁移的过程中寻找着最佳的生存环境。可见, 生存迁移是生物界中普遍存在的一种规律。笔者受到生物迁移和生存规律的启发, 提出一种新的算法——生存迁移算法(Living Migration Algorithm, LMA)。

2 生存迁移算法(LMA)

2.1 LMA 的机理

在算法中, 个体称之为“生物体”, 每次迭代称为“一代”。每个生物体都会记录各自生存位置的生存状况, 它是生物适应环境而获得的生存需要情况。同时, 每个生物体都有一个起码的生存极限, 它们反映了生物体最基本的生存需要。一旦生存状况低于生存极限, 生物迁移被激励, 迁移的能量释放出来, 生物迁移便随即发生, 生物体将随机迁移到其他位

置。生物体就是按照这种生存和迁移机理, 在搜索环境中寻找最佳的生存位置。

生存迁移过程的示意图如图 1 所示, 首先, 3 个生物体: $Life(1)$, $Life(2)$, $Life(3)$ 分别在搜索空间中随机得到各自的初始位置: $Site(1)$, $Site(2)$, $Site(3)$, 且有各自生存极限: $MiniLN(1)$, $MiniLN(2)$, $MiniLN(3)$ 。生存迁移过程由生物体的 3 个基本行为组成: 移动(Move), 分辨(Judge)和迁移(Migrate)。

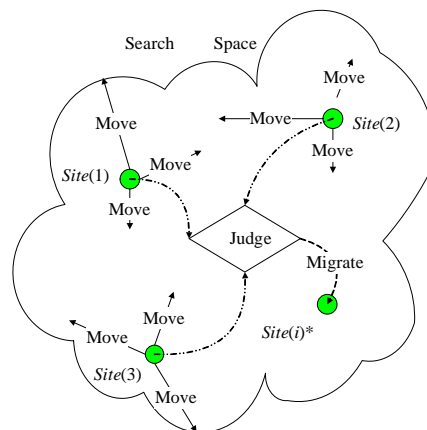


图 1 生存迁移算法机理示意图

基金项目: 教育部高校博士点基金资助项目(20060255006)

作者简介: 李豆豆(1983 -), 男, 硕士研究生、主研方向: 智能算法, 基因调控网络; 邵世煌, 教授、博士生导师; 王 晴, 本科生

收稿日期: 2007-09-29 **E-mail:** lidoudou0516@126.com

生物体通过随机地移动来寻找最优值，移动的步幅不是固定的，它既能以很精细的移动步幅搜索，也能以很大的移动步幅搜索。生物体通过分辨来决定下一代是否需要迁移，其迁移位置的选择也是随机的。当生存状况不能满足生存极限时，生物体就要随机迁移到其他的搜索位置： $Site(i)^*$ ($i=1,2,3$)。

2.2 LMA 建立

根据生存迁移算法的机理，下面给出算法的具体步骤，表 1 介绍了算法中所使用的变量及参数。

表 1 符号说明

符号	说明
N	生物体的个数, $i=1,2,\dots,N$, $p \in [1,2,\dots,N]$
n	每个生物体的维数, $j=1,2,\dots,n$
X_i^j	第 i 个生物体第 j 维的搜索位置
T	每一代中生物体移动的步数, $t=1,2,\dots,T$
G	算法的迭代次数, $g=1,2,\dots,G$
X_{\min}	最小的搜索边界
X_{\max}	最大的搜索边界
R	邻接搜索空间, $R \in [X_{\min}, X_{\max}]$
$f()$	适应度函数
$G()$	生存状况评估函数
$MiniLN_i$	第 i 个生物体的生存极限 $MiniLN_i \in [MiniLN_{\min}, MiniLN_{\max}]$
LS_i	第 i 个生物体的生存状况
$random(0,1)$	在 $[0,1]$ 的随机数
λ	常数, 范围: $[0,1]$

算法步骤:

步骤 1 初始化。所有生物体被随机分布到搜索空间中，根据下式得到各自的初始位置： X_1, X_2, \dots, X_N 。

$$X_i^j = X_{\min}^j + (X_{\max}^j - X_{\min}^j) \cdot random(0,1) \quad (1)$$

$$X_i^j = a_i^j, \quad a \in [X_{\min}, X_{\max}] \quad (2)$$

步骤 2 生存极限。每个生物体根据下式得到各自的生存极限，范围： $[0,1]$ 。

$$MiniLN_i = MiniLN_{\min} + (MiniLN_{\max} - MiniLN_{\min}) \cdot random(0,1) \quad (3)$$

$$MiniLN_i = b_j, \quad b_j \in [MiniLN_{\min}, MiniLN_{\max}] \quad (4)$$

步骤 3 移动。每个生物体第 t 步移动后的搜索位置由下式得到。

$$X_i^j(t) = X_i^j - \Delta x_i^j(t) + 2 \cdot \Delta x_i^j(t) \cdot random(0,1) \quad (5)$$

$$\Delta x_i^j(t) = R^j \cdot random(0,1) \quad (6)$$

$$R^j = \lambda \cdot (X_{\max}^j - X_{\min}^j) \quad (7)$$

步骤 4 搜索结果。由下式得到每步移动后的搜索结果。

$$f_i(t) = f(X_i^j(t)) \quad (8)$$

步骤 5 移动步数若小于预先给定的移动步数 T ，则转步骤 3。

步骤 6 生存状况。每个生物体都会按照下式得到各自最大的搜索结果。然后从所有生物体的搜索结果中选择全局最优值： $\max(f_{i\max})$ 。

$$f_{i\max} = \max(f_i(t)) \quad (9)$$

按照下式得到每个生物体的生存状况。

$$LS_i = G(f_{i\max}, \max(f_{i\max})) = \frac{f_{i\max}}{\max(f_{i\max})} \quad (10)$$

式(10)是对大于 0 的搜索空间定义的。对于 $f_i(t)$ 为负的

搜索空间，通过在函数中加入足够大的正数，保证式(10)的要求。

步骤 7 终止条件。假如所有生物体的生存状况都满足生存极限： $LS_i \geq MiniLN_i (i=1,2,\dots,N)$ 或者 $g > G$ ，则算法终止。

步骤 8 迁移。对生存状况满足生存极限的生物体 LS_p $MiniLN_p$ ，则它的搜索位置不变。否则，它将按照式(11)或式(12)随机迁移到搜索空间的其他位置。然后，转到步骤 2。

$$X_p^j = X_{\min}^j + (X_{\max}^j - X_{\min}^j) \cdot random(0,1) \quad (11)$$

$$X_p^j = a_p^j, \quad a \in [X_{\min}, X_{\max}] \quad (12)$$

3 生存迁移分析

3.1 收敛性证明

生存迁移算法会在下面 2 种情况下以局部最优值收敛：

(1)算法在很大的水平面上搜索；(2)偶然地，在某一代每个生物体所搜索的最大值 $f_{i\max} (i=1,2,\dots,N)$ 几乎相等。这 2 种情况都能满足步骤 7 的终止条件，使算法终止。尽管如此，但由步骤 3 的随机移动和步骤 8 的随机迁移机制，上面 2 种情况发生的概率非常小，可以认为是小概率事件。

对全局优化问题，设其有全局最优值 $f(X^*)$ ，根据概率收敛的定义^[3]，证明生存迁移算法依概率收敛于全局最优值。

$$\max f(x) \text{ s.t. } X \in U \quad (13)$$

其中， $f: U \rightarrow R^1$ 为实值映射； $X \in R^n$ ； $U = \prod_{j=1}^n [X_{\min}^j, X_{\max}^j]$ 。

证明：设 $o \in U$ 为全局最优点。算法步骤 1 产生 N 个生物体的初始位置： X_1, X_2, \dots, X_N ，步骤 3 以这 N 个初始位置为中心形成 N 个搜索区域 $\Gamma^k: \prod_{j=1}^n [X_k^j - R^j, X_k^j + R^j]$ ($k=1,2,\dots,N$)。要求 Γ^k 覆盖 o 点的概率，即求 o 点落入区域 Γ^k 内的概率，设 H 表示事件“ Γ^k 覆盖 o 点”，则

$$P(H) = P\{o \in \Gamma^k\} = \alpha \frac{L[\Gamma^k]}{LU} = \alpha \frac{\prod_{j=1}^n 2R_j}{\prod_{j=1}^n (X_{\max}^j - X_{\min}^j)} = 2\alpha\lambda^n \quad (14)$$

其中， α 为域 Γ^k 的测度系数， $1/2^n \leq \alpha \leq 1$ ； $0 < \lambda < 1$ 为常数。

设 $S(\varepsilon)$ 为满意解集，其 Lebesgue 测度大于 0 ^[4]，且满足：任给 $\varepsilon > 0$ ，有

$$S(\varepsilon) = \{X \in U \mid |f(X) - f(X^*)| < \varepsilon\} \quad (15)$$

设 H_i 表示事件“有且只有 i 个搜索区域覆盖最优点 o ”， $i=0,1,\dots,N$ ，由于算法步骤 1 实质上是做 N 次 Bernoulli 试验，因此有

$$P(H_i) = C_N^i [P(H_i)]^i [1 - P(H_i)]^{N-i} = C_N^i (2\alpha\lambda^n)^i (1 - 2\alpha\lambda^n)^{N-i} \quad (16)$$

设 S 表示事件“在 T 次步骤 3 的移动中至少有一点落入 $S(\varepsilon)$ 域”。显然， H_0, H_1, \dots, H_N 互不相容， $P(\bigcup_{i=0}^N H_i) = 1$ 。由全概率公式及 $P(S \mid H_0) = 0$ ，得

$$P(S) = \sum_{i=1}^N P(H_i) P(S \mid H_i) \quad (17)$$

其中， $P(S \mid H_i)$ 是有且只有一个搜索区域覆盖最优点 o 的条件下事件 S 发生的概率。假如最优点 o 为空间 U 内的点，则得下式：

$$P(\{X^k(t) \in S(\varepsilon)\}) = \frac{L[S(\varepsilon)]}{L[\Gamma^k]} = \frac{\prod_{j=1}^n \varphi(\varepsilon)(X_{\max}^j - X_{\min}^j)}{\prod_{j=1}^n 2\lambda(X_{\max}^j - X_{\min}^j)} = \left(\frac{\varphi(\varepsilon)}{2\lambda}\right)^n \quad (18)$$

若 o 在 U 的边界上, 则 Γ^k 的部分区域会覆盖到 U 的外部, $P(\{X^k \in S(\varepsilon)\})$ 会增大。

按照最优优点 o 为空间 U 内的点的原则, 有

$$P(S|H_1) = 1 - (1 - P(\{X^k(t) \in S(\varepsilon)\}))^T = 1 - (1 - (\frac{\varphi(\varepsilon)}{2\lambda})^n)^T \quad (19)$$

其中, $\varphi(\varepsilon)$ 是关于 ε 的正数。

同理可得

$$P(S|H_2) = 1 - (1 - P(\{X^k(t) \in S(\varepsilon)\}))^{ixT} = 1 - (1 - (\frac{\varphi(\varepsilon)}{2\lambda})^n)^{ixT} \quad (20)$$

由式(16)、式(17)、式(20)得

$$P(S) = \sum_{i=1}^N \{C_N^i (2\alpha\lambda^n)^i (1 - 2\alpha\lambda^n)^{N-i} \times (1 - (1 - (\frac{\varphi(\varepsilon)}{2\lambda})^n)^{ixT})\} \quad (21)$$

设 S_g 代表事件“LMA 迭代 g 次至少有一个搜索位置落入 $S(\varepsilon)$ 域”, 则

$$P(S_g) = 1 - (1 - P(S))^g \quad (22)$$

根据式(21)可知 $P(S) > 0$, 得

$$\lim_{g \rightarrow \infty} P(S_g) = \lim_{g \rightarrow \infty} [1 - (1 - P(S))^g] = 1 \quad (23)$$

由算法的迁移机制, 可知

$$\lim_{g \rightarrow \infty} P(\{X(g) \in S(\varepsilon)\}) = 1 \quad (24)$$

其中, $X(g)$ 为算法第 g 代中不需要移动的点, 即最优值位置。

根据式(15), 得

$$\lim_{g \rightarrow \infty} P(\{|f(X(g)) - f(X^*)| < \varepsilon\}) = 1 \quad (25)$$

因此, 算法依概率收敛于全局最优值。

3.2 收敛速度估计

按照算法以给定概率收敛到全局最优值所运行的迭代次数来计算 LMA 的收敛速度。设 LMA 以给定概率 P 收敛到全局最优值, 即 $P(S_g) = P$ 。

根据式(21)、式(22)得

$$g^* = \ln(1-P) / \ln(1 - \sum_{i=1}^N \{C_N^i (2\alpha\lambda^n)^i (1 - 2\alpha\lambda^n)^{N-i} \times (1 - (1 - (\frac{\varphi(\varepsilon)}{2\lambda})^n)^{ixT})\}) \quad (26)$$

3.3 收敛时间估计

根据 LMA 以给定概率 P 迭代 g^* 次收敛到全局最优值, 讨论函数 $f()$ 和 $g()$ 计算的总次数来研究算法的时间复杂性。由算法步骤可知, 式(8)和式(10)被计算的总次数为

$$N_{g^*} = (1+T) \times N \times g^* \quad (27)$$

4 生存迁移应用

4.1 问题的提出

笔者选择了 4 个典型的最优化问题作为测试生存迁移算法性能的对象, 以下分别介绍了 4 个问题的特点。

(1)Bump问题是 Keane 提出的^[5], 测试函数见下式:

$$\text{Maximise: } f(x_i) = \frac{\text{abs}[\sum_{i=1}^n \cos^4(x_i) - 2\prod_{i=1}^n \cos^2(x_i)]}{\sqrt{\sum_{i=1}^n ix_i}} \quad (28)$$

$$0 < x_i < 10, \quad i = 1, 2, \dots, n$$

$$\text{subject: } \prod_{i=1}^n x_i > 0.75, \quad \sum_{i=1}^n x_i < 15n/2$$

由于 Bump 问题具有超多峰、全局最优值未知、变量被约束的特点, 它非常接近现实的最优化问题, 可以把解决 Bump 问题作为检测算法性能的复合标准^[6]。

(2)Flat 问题是由 Rosenbrock 提出的^[7], 其最优值在光滑的顶部, 且临近空间的函数值都类似, 算法必须能够完成精细的局部搜索。Flat 问题能够作为检测算法鲁棒性的标准^[6]。

(3)Multi-modal 问题是由 Shubert 提出的^[8], 测试函数见下

式。它总共有 18 个最高峰, 分布在不同的位置, 如果能够找到所有的峰值, 就能说明此算法具有较好的鲁棒性^[6]。

$$\text{Maximise: } f(x_i) = -\prod_{i=1}^n \{ \sum_{j=1}^5 j \cdot \sin[(j+1) \cdot x_i + j] \} \quad (29)$$

$$-10 \leq x_i \leq 10$$

(4)Levy 提出了含有噪音的 Multi-modal 问题^[9], 它主要用于检验算法在噪音环境下解决优化问题的能力^[6]。

4.2 实验结果

对以上 4 个问题, 分别在每种迭代次数下各做了 320 次实验, 算法的参数设置见表 2。

表 2 算法的参数设置

G	N	n	T	λ	$MiniLN_{\min}$	$MiniLN_{\max}$
100, 1 000, 2 000	10	2	5	1	0.95	1

可被接受的全局最优值设置为: (1)在 Bump 问题中, 最优值在 0.67 附近, 认为大于 0.65 的值都可被接受为最优值; (2)对于 Flat 问题, 其最优值为 0, 认为大于 -0.1 的搜索结果都可以被接受为最优值; (3)对于 Multi-modal 问题, 其最优值在 178.5 附近, 认为大于 175 的搜索结果都可接受为最优值; (4)而对于 Multi-modal 噪音问题, 其最优值在 186.6 附近, 大于 180 的搜索结果都可以接受为最优值。

为了验证生存迁移算法的有效性, 与 Free Search 算法进行了比较, 它是最近被提出的一种综合搜索能力优于粒子群算法、遗传算法、微分算法的全局优化算法^[6]。比较结果见表 3 及图 2。在表 3 中, (*)表示在 Free Search 算法中, bump 问题的迭代次数分别被设置为: 1 000, 10 000 和 20 000, 每个生物体被设为 20 维变量。对于 Bump 问题和 Multi-modal noisy 问题的搜索结果见图 3 和图 4。从比较的结果可以看出, 生存迁移算法具有更好的全局优化能力和收敛速度, 能够较好地解决局部最优问题, 有效地处理含有噪音的优化问题。

表 3 实验结果

算法	G	测试问题			
		Bump	Flat	Multi-modal	Multi-modal noisy
Free Search	100	0(*)	1	60	8
Free Search	1 000	217(*)	91	303	65
Free Search	2 000	305(*)	199	303	70
LMA	100	163	157	69	69
LMA	1 000	187	168	234	224
LMA	2 000	202	200	293	249

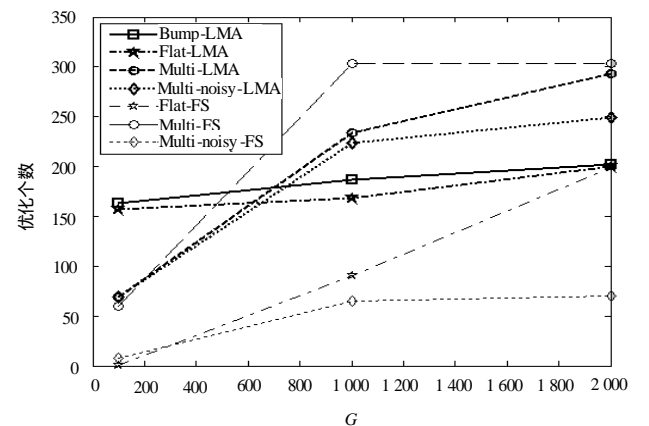


图 2 实验结果

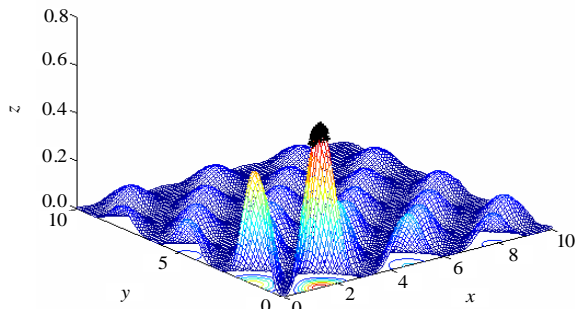


图3 Bump 问题的搜索结果

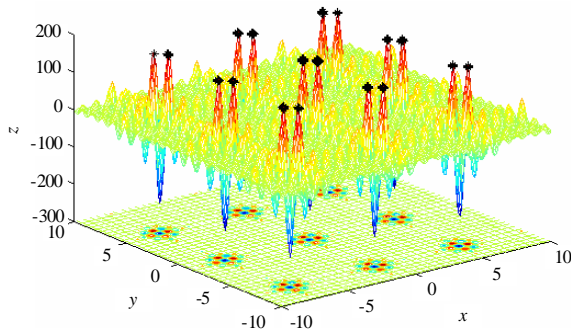


图4 Multi-modal 问题的搜索结果

5 结束语

生存迁移算法充分利用移动和迁移的随机机制,避免算法陷入局部最优,通过生存状况了解未知的搜索空间,借助生存极限约束整体向最优搜索位置迁移。本文按照生存迁移算法的步骤给出了算法收敛性分析,证明以概率收敛到全局

最优值,并给出了算法的收敛速度估计和时间计算复杂度估计。对4个问题的实验结果表明,生存迁移算法具有较快的收敛速度和全局优化能力。

参考文献

- [1] 徐炳川. 生存迁移思想的起源与发展[J]. 地质力学学报, 1995, 1(3): 89-95.
- [2] 徐炳川. 古生物迁移理论的历史、现状及研究方法[J]. 古地理学报, 1999, 1(2): 62-71.
- [3] 茆诗松, 诚依明, 濮晓龙. 概率论与数理统计教程[M]. 北京: 高等教育出版社, 2004.
- [4] 郭懋正. 实变函数与范函分析[M]. 北京: 北京大学出版社, 2005.
- [5] Keane A J. Experience with Optimizers in Structural Design[C]// Proc. of the Conference on Adaptive Computing in Engineering Design and Control. [S. l.]: IEEE Press, 1994: 14-27.
- [6] Penev K, Littlefair G. Free Search—A Comparative Analysis[J]. Information Sciences, 2005, 172(1/2): 173-193.
- [7] Rosenbrock H H. An Automate Method for Finding the Greatest or Least Value of a Function[J]. Computer Journal, 1960, 3(3): 175-184.
- [8] Simones A B, Costa E. Transposition Versus Crossover: An Empirical Study[C]//Proc. of the Genetic and Evolutionary Computation Conference. Orlando, FL, USA: [s. n.], 1999: 612-619.
- [9] Levy A, Montalvo A, Gomez S, et al. Topics in Global Optimization[C]//Lecture Notes in Mathematics. New York, USA: Springer-Verlag, 1981.

(上接第 193 页)

4.3 结果分析

通过对3种初始化方法的理论分析和实验结果的对比可以得到如下结论:

(1)由理论可知,PCA和SPCA初始化完全由数据集确定且对于大矩阵分解都有转换算法;随机和FCM初始化都受到随机性的影响,但FCM经过一系列的迭代过程后受初始化的影响明显小于随机初始化。

(2)由图1和图2可知,2种损失函数相应的迭代算法都收敛于稳定值,但以KL距离定义的损失函数比以欧氏距离定义的收敛速度快。

(3)由图1和图2可知,欧氏距离损失函数值小于KL距离损失函数值,直观上,欧氏距离损失函数应该得到更好的分类结果,图3和图4对应的平均F1值就很好地说明了这一点。

(4)由图3和图4可知,对于2种损失函数,SPCA的效果明显优于另外2种初始化方法,且结果相对稳定。当维数高于200时,SPCA的效果要差一些,这也反映了信息冗余的现象。

5 结束语

本文针对文本挖掘提出了PCA、SPCA和FCM3种初始化方法,并将其应用于文本分类。介绍了NMF和SPCA、FCM的基本原理,针对大矩阵分解提出SPCA的适应性算法,对于多类文本分类采用KNN分类器和平均F1值进行度量。实验结果表明:3种初始化方法有效地解决了随机初始化中受

初始化影响显著的问题,其中,SPCA在稳定性和有效性方面都优于其他2种初始化方法;以欧氏距离定义的损失函数的结果优于以KL距离定义的损失函数对应的结果,但其收敛速度不如后者。下一步需要对NMF理论作进一步研究,寻求NMF的低维 r 的有效确定方法。

参考文献

- [1] Lee D D, Seung H S. Learning the Parts of Objects by Non-negative Matrix Factorization[J]. Nature, 1999, 401(21): 788-791.
- [2] Hou Xinwen, Zhang Hongjiang. Learning Spatially Localized, Parts-based Representation[C]//Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. [S. l.]: IEEE Computer Society Press, 2001: 207-212.
- [3] Satoru Tsuge, Masami Shishibori, Shingo Kuroiwa, et al. Dimensionality Reduction Using Non-negative Matrix Factorization for Information Retrieval[C]//Proceedings of IEEE International Conference on Systems. [S. l.]: IEEE Press, 2001: 960-965.
- [4] Paul S, Du Shuyan, Parra L. Recovery of Constituent Spectra Using Non-negative Matrix Factorization[C]//Proc. of SPIE'03. Columbia, USA: [s. n.], 2003: 321-331.
- [5] Koren Y, Carmel L. Robust Linear Dimensionality Reduction[J]. IEEE Transactions on Visualization and Computer Graphics, 2004, 10(4): 459-470.
- [6] 北京大学数学系几何与代数教研室代数小组. 高等代数[M]. 2版. 北京: 高等教育出版社, 2002.
- [7] 徐月芳. 基于遗传模糊C-均值聚类算法的图像分割[J]. 西北工业大学学报, 2002, 20, 12(4): 549-553.