

求解关联维数的快速算法研究

周 越, 杨 杰

(上海交通大学图像处理与模式识别研究所, 上海 200030)

摘 要: 针对关联维数计算速度慢、耗时多和运算效率低的缺陷, 本文从算法和结构两个方面研究了关联维数的快速求解方法, 提出了区间阈值累加法和区间维值累加法. 这两种算法通过对 $[1 \times m_{\max}] \cdot [1 \times r_{\max}]$ 表格的批处理填写以达到减少重复运算和提高计算效率的要求. 文中详细地介绍了算法步骤, 分析了算法的运算量, 通过与标准算法和一般改进算法的对比, 显示出本文提出算法的优越性.

关键词: 分形; 关联维数; 维值区间累加法; 阈值区间累加法

中图分类号: TV911 **文献标识码:** A **文章编号:** 0372-2112 (2002) 10-1526-04

Research of the Optimized Algorithm for the Calculation of Correlation Integrals

ZHOU Yue, YANG Jie

(Institute of Image Processing & Pattern Recognition, Shanghai Jiaotong University, Shanghai, 200030, China)

Abstract: In order to solve the problem of slow speed, based on the canonical algorithm of calculation of the correlation integrals, this paper not only provides several improvements, but also puts forward two optimized algorithms. The algorithms are based on the region of embedded dimension and region of distance threshold. They are filled in table of $[m_{\max} \times r_{\max}]$ ($m_{\max}$ is number of embedded dimension, $r_{\max}$ is number of distance threshold) by batch methods giving up the one by one processing methods. In this paper, the theory and algorithm are narrated in detail. The performance of two optimal algorithms is compared with canonical algorithm and improvement method. The experimental result shows that optimal algorithms need least time to calculate the correlation dimension for same length time series with same number of distance threshold and embedded dimension since they have little redundant calculations and enhance the calculation efficiency obviously.

Key words: fractal; correlation dimension; region of embedded dimension method; region of distance threshold method

1 引言

由于关联维数能够在一定程度上反映时间序列在各个时刻的相关性程度的变化率, 因此它在混沌分形信号处理中有着极为重要的地位. 求解关联维数的算法一直是混沌与分形理论中研究的一个重点问题, 许多研究者都从关联维数的定义式出发对标准算法进行了不同的改进^[1-3], 运算量有了一定的减少, 计算效率也有所提高, 但仍然没有从本质上解决重复运算的问题. 本文提出了两种计算关联维的快速算法, 在一定程度上能够提高计算效率, 减少了运算时间, 取得了较好的效果.

2 标准关联维数冗余计算量分析与改进

2.1 关联维数的定义

采用 Takens 提出的时间延迟技术, 从一维时间序列中构造出一个 m 维的向量空间 $\{X_i^{(m)} = \{x_i, x_{i+1}, \dots, x_{i+m-1}\}\}$, 这里是子空间的嵌入维数, 用上述向量计算关联积分:

$$C(r, m) = \frac{1}{N^2} \sum_{i=1}^{N-1} \sum_{j=i+1}^N H[r - d(X_i^{(m)}, X_j^{(m)})] \quad (1)$$

$d(X_i^{(m)}, X_j^{(m)})$ 为向量 $X_i^{(m)}$ 和 $X_j^{(m)}$ 的距离, N 为样本点总数, r 是在向量空间 R^m 中, 两个向量间距离阈值, H 为阶跃函数, 定义为 $H(x) = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0 \end{cases}$. 关联维数为 $D_2^{(m)} = \lim_{r \rightarrow 0} \frac{\ln C(r, m)}{\ln r}$; 关联维数就为 $D_2 = \lim_{m \rightarrow \infty} D_2^{(m)}$. 对某一信号的 m 值是无法先知, 需经过一系列 m 的计算后确定. 例如 EEG 的嵌入维数 m 从 20 开始 D_2 趋于稳定, 因此在计算时 m 取为 25 ~ 30 即可.

2.2 标准算法的运算量分析与改进

计算关联维数的运算量主要来自求解关联积分, 而关联积分的运算量就是在 $[1 \times m_{\max}] \cdot [1 \times r_{\max}]$ 表格上逐点填写 $H = 1$ 的计数值; 再求得关联维数 D_2 . 关联积分运算量来自两个方面, 一方面是空间向量的距离运算, 另一方面是各距离与

不同阈值的比较.按标准的算法,对不同嵌入维空间中所有向量间的距离与所有阈值 r 进行比较,需做的比较次数为

$$SUM_{cp} = \frac{r_{\max}}{2} \cdot \sum_{k=1}^{m_{\max}} [N - k + 1] \cdot [N - k + 1] \text{ 其中存在许多的冗余计算,原因如下:}$$

(1) 若时间序列的长度为 N , 距离阈值 r 取 K 个不同的值, 需要计算 $K \cdot N(N-1)/2$ 次向量距离计算. 距离可用 $1, 2, \dots, \infty$ -范数表示并且是等价的, 有定理保证,

定理 1 假设 $d(X_i^{(m_1)}, X_j^{(m_2)})$ 的范数形式有 $X_i - X_j$, 则对于任何范数表示 $d(X_i^{(m_1)}, X_j^{(m_2)})$, 得到的关联维数 D_2 都是相同的. (证明略)

因此, 在计算相空间中两两向量之间的距离时, 向量的 $X_i - X_j$ 来表征向量间的距离

$$d(X_i^{(m)}, X_j^{(m)}) = |X_i - X_j|_{\infty} = \max(|X_i - X_j|) \quad (m = 1, 2, \dots, m_{\max}) \quad (2)$$

本文如未特殊说明, 距离一律用 ∞ -范数来计算, 这样可以大大节省运算量.

(2) 在计算 $|X_i^{(m)} - X_j^{(m)}|$ 时, 对于不同的 m , r 在计算向量间的距离时各个分量运算存在大量的重复计算. 在不同嵌入维 m_k 和 m_l ($m_k < m_l$) 的向量组中, $d(X_i^{(m_k)}, X_j^{(m_k)})$ 的计算与 $d(X_i^{(m_l)}, X_j^{(m_l)})$ 的前 m_k 的减法计算是重复的应予以祛除, 用以下的递推式求得

$$d(X_i^{(m_l)} - X_j^{(m_l)}) = \max[d(X_i^{(m_k)} - X_j^{(m_k)}), |X_i^{(m_l)} - X_j^{(m_l)}|_{\infty}] \quad (m_l = m_k + 1, m_k + 2, \dots, m_{\max}) \quad (3)$$

(3) 对每一个 r , 相空间中任何两个向量的距离都需要计算一遍, 那么对于每一向量对都需要与 K 个 r 进行比较, 其中有 $(K-1)$ 次是多余的. 因此在计算向量距离之后应将其进行取整量化, 引入等距积分^[3], 通过累加得到关联积分,

$$C(ir, d) = \frac{1}{N^2} \sum_{r=0}^{ir} Q_s(r) \quad (4)$$

其中:

$$Q_s(r) = \sum_{i=1}^{N-1} \sum_{j=i+1}^N Q(r - |X_i - X_j|_{\infty}), Q(x) = \begin{cases} 1, & x = 0 \\ 0, & x \neq 0 \end{cases} \quad r = 1, 2, \dots, n_{\max}$$

这种方法虽然比原始方法节省时间, 它解决了不同距离阈值 r 下的重复运算, 根据式 (4), 只要计算一次两两向量间的距离, 通过累加就可得到关联积分, 从而使关联积分的运算量大幅度下降. 但对于不同嵌入维 m 下的这类重复运算无法清除.

2.3 改进措施的评价和小结

以上这些改进措施在一定程度上降低了运算量, 同时提高了运算效率, 但最关键的一点并没有从根本上得到解决, 即对 $[1 \times m_{\max}] \cdot [1 \times n_{\max}]$ 表格逐点填写与 $|x_i - x_j|$ 的重复运算. 因此需要打破常规的约束, 寻找出全新理念的算法. 本文从两个角度考虑, 即从距离阈值变量的变化和嵌入维变量的变化规律出发, 提出了阈值区间累加法和维值区间累加法的快速算法, 以解决上述问题, 下面两节将对它们作详细的介绍.

3 第一种关联积分的快速算法—维值区间累加法

在第 2 节中论述的改进算法并没有脱离对 $[1 \times m_{\max}] \cdot [1 \times n_{\max}]$ 表格上逐点填表的方式, 显然这种算法仍然存在计算效率不高的问题, 本文针对这一点提出了第一种优化算法, 即维值区间累加法. 对于某一个距离阈值, 寻找出满足它的一个嵌入维值区间.

定理 2 对于 $\forall n, Z^+$, 有 $X_i^{(m_k)} - X_j^{(m_k)} < r_n$, 则对于所有的 $m < m_k$, 都有 $X_i^{(m)} - X_j^{(m)} < r$ 成立, 其中, $X_i^{(m)} = [x_1, x_2, \dots, x_{i+m}, \dots, x_{i+m_k-1}]$.

首先将式 (1) 变换成下式

$$C(r_n, m) = \frac{1}{N^2} \sum_{i=1}^N \sum_{j=i+1}^{N-i} H[r_n - d(X_{i+j}^{(m)}, X_j^{(m)})] \quad (5)$$

式 (5) 与式 (1) 在下标变量遍历过程是不同的, 有助于分离下标变量 i 和 j . 求 $|X_i - X_j| < r$ 其实质就是计算和分析 $|x_i - x_j|$, 因此可以直接对时间序列的元素进行操作以达到减少重复运算的目的. 具体的求解方法如下所述:

求各个元素间的距离 $|x_j - x_{i+j}|$, 且将其存入上三角矩阵 $\Lambda_{[N, N]}$. 考虑到式 (5) 中的 m , 对于所有的 (i, j) 对不是常数, 是按照 $m = 1, 2, \dots, m_{\max}$ 变化. 若按照向量下标的遍历方式来做 $\Lambda_{[N, N]}$ 的遍历, i 被分离出来, 从下标变量 j 延平行于对角线的方向出发, 则会有

$$m = j_f - j_s + 1 \quad (6)$$

且 $|x_j - x_{i+j}| < r_n$ 对于 j 满足 $j_s \leq j \leq j_f$ (7)

$$|x_j - x_{i+j}| \geq r_n \text{ 对于 } j \text{ 满足 } j = j_s - 1, j = j_f + 1 \quad (8)$$

j_s 和 j_f 是表示在 j 下标下满足上述条件的元素队列的起止位置. 其队列的长度定义为 w . w 是与 r 直接相关的, 且当给定某一个 r_p 时在 j 坐标下可能具有 l 个长度不同的队列, 故另记作 $w(r_p, k)$, $k = 1, 2, \dots, l$.

简单地, 当外循环变量 $i = 1$ 时, 在给定的距离阈值为 r_p 时, 可得

$$\Delta_{i=1}(m, r_p) = \sum_{k=1}^l \sum_{m=m_{\max}}^{m_{\max}} [w_{i=1}(r_p, k) - m + 1]$$

那么拓展到所有的 i 时, 则可得

$$\Delta(m, r) = \sum_{i=1}^N \sum_{k=1}^l \sum_{m=m_{\max}}^{m_{\max}} [w_i(r_p, k) - m + 1] \triangleq z(m, r)$$

$z(m, r)$ 给出了在 m 维向量空间中队列的个数. 当然 $C(m, r_n)$ 中不仅仅包含了由这些队列产生的计数, 而且包含了由 $m > m_{\max}$ 产生队列而引起的计数, 因此可以用下式表示:

$$C(m, r) = \frac{1}{N^2} \sum_{i=1}^{m_{\max}} (i - m + 1) \cdot z(i, r) + \tilde{z}(M, M_{\max}, r)$$

函数 $\tilde{z}(m, m_{\max}, r)$ 是表示由 $m > m_{\max}$ 时的队列引起的计数. 显然存在一个 $\tilde{m}_{\max}$, 使得对于所有的 $m > \tilde{m}_{\max}$, 有 $z(m, r) = 0$. $\tilde{m}_{\max}$ 的值是与原始序列长度 N 和距离阈值 r_n 有关, $\tilde{m}_{\max}$ 的值随 N 和 r_n 的增大而增大. 但是在实际运算中我们不可能取到 $\tilde{m}_{\max}$, 必然取一个适当的 m ($m < \tilde{m}_{\max}$), 这样就会引起 m 的截断效应.

以上讨论的是针对给定的一个阈值 r_p 的情况, 然而在关

联积分 $C(m, r)$ 中阈值 r 是变化的. 可以选取一个 r_0 , 有 $r_{\max} = n_{\max} \times r_0$, 这样 r_n 的变化转变为 $n (n = 1, 2, \dots, n_{\max})$ 的变化, 因此在这种情况下, 对以上讨论的条件和函数应做相应地改变:

$$m(n) = j_f(n) - j_s(n) + 1$$

在固定外循环变量 i 时, 对于不同的 n 在 j 轴上运用式(7)和式(8), 形成若干宽度为 $w(n, m)$ 的队列, 随 n 的增加队列宽度不断增加. 当某一队列在 n 变化时, 队列拓展的方式为若 $n_j^* = |x_j - x_{i+j}| / r_0$, 则

$$\text{当 } n_j^* < n_{j-1}^* \text{ 时 } j_s(n) = j \quad n_j^* \leq n < n_{j-1}^*$$

$$\text{当 } n_j^* \geq n_{j-1}^* \text{ 时 } j_f(n) = j \quad n_{j-1}^* \leq n < n_j^*$$

当出现 $m(n) > m_{\max}$ 时, 按照下面公式计算截断效应函数 $\tilde{z}(m, m_{\max}, r_n)$:

$$\tilde{z}(m, m_{\max}, r_n) = \tilde{z}(m, m_{\max}, r_n) + [m(n) - m_{\max} + 1]$$

运用这种方法, 可以很快地通过一次遍历所有的 $\frac{N^2-1}{2}$ 个 (x_i, x_j) 且得到表格 $[1 \times m_{\max}] \cdot [1 \times n_{\max}]$ 上的 $z(m, r_n)$ 函数值, 没有任何重复的 $|x_i - x_j|$ 计算. 而在标准算法中为了得到不同嵌入维数下的向量对 $(X_i^{(m)}, X_j^{(m)})$ 间的距离, 需要多次遍历所有的 (x_i, x_j) 对, 因此多余的计算非常大.

4 第二种关联积分的快速算法—阈值区间累加法

上面讲述第一种方法是在寻找维数区间的基础之上完成 $[1 \times m_{\max}] \cdot [1 \times m_{\max}]$ 表格的填写, 本节将要论述的是从寻找阈值区间为出发点, 完成 $[1 \times m_{\max}] \cdot [1 \times m_{\max}]$ 表格的填写. 这两种方法是从问题的两个不同方面进行的, 只是具体操作手法不同. 下面简要地介绍阈值区间法求解关联维数.

时间序列中将任意两个元素 (x_i, x_j) 的距离 $|x_i - x_j|$ 求出存入矩阵 S , 并将其各元素的位置重新安排 $C_{(N-1)(N-1)}^1$. 继续将 $C_{(N-1)(N-1)}^1$ 每一列中相邻两个元素的距离存入 $C_{(N-2)(N-2)}^2$, 这样一直做下去, 直到 $C_{(m_{\max}+1)(m_{\max}+1)}^{N-m_{\max}+1}$. 其实很容易看出, 上三角矩阵 $C_{(N-1)(N-1)}^1$ 的各个元素是 $m=1$ 时所有向量间的距离; $m=2$ 时, 所有向量间的距离存放在上三角矩阵 $C_{(N-2)(N-2)}^2$ 的各个元素中; 依次类推, $C_{(m_{\max}+1)(m_{\max}+1)}^{N-m_{\max}+1}$ 上三角元素存放的是 $m=m_{\max}$ 所有向量间的距离. 它们对应的个数可以用以下的式子归纳, $Num^{(m)} = \frac{(N-m)(N-m-1)}{2}$, 其中, $Num^{(m)}$ 是嵌入维数为 m 时上三角矩阵 $C_{(N-m)(N-m)}^m$ 中元素的总个数, 这些数据在编程时无需计算, 直接将它们预先计算好, 存入内存. 由定理 1 可知, $C^2, C^3, \dots, C^{N-m_{\max}+1}$ 之间存在着密切的联系, 在实际计算中不必求出 $C^2, C^3, \dots, C^{N-m_{\max}+1}$, 因为 C^1 已经可以反映出所有的信息.

将矩阵 C^1 的各个元素 c_{ij} 与最小的阈值 r_0 求 Heaviside 函数值, 得到矩阵 Z , 它是一个 $[0, 1]$ 分布的矩阵, 即 $Z = \{z_{ij} | z_{ij} = 0, \text{ 或 } z_{ij} = 1, i = 1, 2, \dots, N-1; j = N-i\}$.

对于每一列寻找连续出现“0”的队列 $w(l, k)$, k 表示连续出现“0”的队列个数, l 为队列的长度. 可以得到下面的结

论, 对于给定的距离阈值 r , 所有 $m (m < l)$ 维的向量间距离的总个数需要减去 $(l-m+1)$ 个使得 Heaviside 函数输出为 0 向量对的个数. 即

$$Num_H^{(m)} = \frac{(N-m)(N-m-1)}{2} - \sum_{i=1}^k \{ [l_i \cdot H(l_i - m)] - m + 1 \}$$

其中, $H(\cdot)$ 是 Heaviside 函数.

当距离阈值发生变化时, 队列 $w(l, k)$ 也会变化. 同样地, 可以选取一个 r_0 , 有 $r_{\max} = n_{\max} \times r_0$, 这样 r_n 的变化转变为 $n (n = 1, 2, \dots, n_{\max})$ 的变化, 则有 $w(l(m), k(n))$.

首先, 按照图 1 的排列, 分离下标变量 i 和 j . 给定一个 i , 取最小的阈值 r_0 , 得到队列 $w_i(l, k)$, 记录下每一个队列的起始位置 j_s 和 j_f , 随着的变化有 $j_s(n)$ 和 $j_f(n)$.

以后当 n 由小变大时, 只需要比较“0”队列中各个元素与阈值的大小. 这样计算量随着阈值的变大而大幅度减小.

在 $[1 \times m_{\max}] \cdot [1 \times m_{\max}]$ 表格中填写给定 r 时可以将所有的嵌入维的向量对满足 Heaviside 函数输出为 1 的总数计算出来.

$$c(m, r_0) = \frac{(N-m)(N-m-1)}{2} - \sum_{k=1}^M \frac{m_{k,r_0}(m_{k,r_0}-1)}{2} - \frac{m_{\max}(m_{\max}-1)}{2}$$

检查对于每一列是否所有的距离阈值 r_n 已全部遍历, 是否所有的维数值 d 已经全部遍历, 直到全部完成为止. 最后用式(1)求出关联维数.

5 算法的运算量粗略分析与实验对比

对于标准算法, 其运算量很大. 其中减法运算次数: N_{sub}

$$= \sum_{k=1}^{m_{\max}} \frac{(N-k+1)(N-k)}{2} \times k, \text{ 其中包含了大量的重复运算;}$$

$$\text{除法运算次数: } N_{\text{div}} = \sum_{k=1}^{m_{\max}} \frac{(N-k+1)(N-k)}{2}; \text{ 加法运算次数:}$$

$$N_{\text{add}} = \sum_{k=1}^{m_{\max}} \sum_{i=1}^{N-k} \sum_{j=1}^{N-k-i} S_{ij}, \text{ 其中: } S_{ij} = X_i^{(m)} - X_j^{(m)} / r_0.$$

维值区间法和阈值区间法的运算量: 减法运算次数: N_{sub}

$$= \frac{N(N-1)}{2}, \text{ 只做一次元素之间的两两相减, 没有重复运算.}$$

比较运算次数: 由于维值区间累加法中维值区间的个数和宽度各不相同, 因此比较运算次数无法准确地以一个通用的解析式概括. 但有一点是可以肯定的, 即它的次数肯定小于标准算法中的比较运算次数.

表 1 各种算法所需时间的相对比较结果

信号长度	512	1024	2048
标准算法	1	1	1
改进算法	6.5 %	4 %	0.8 %
维值区间法	< 6 %	< 1.5 %	0.1 %
阈值区间法	< 6 %	< 1.5 %	0.15 %

表 1 是产生了三种长度的分形时间序列, 利用计算机按照文中所涉及的不同算法计算关联维数, 用相对于标准算法所需时间来描述各个算法的性能. 其中, 距离阈值取 40 个, 嵌

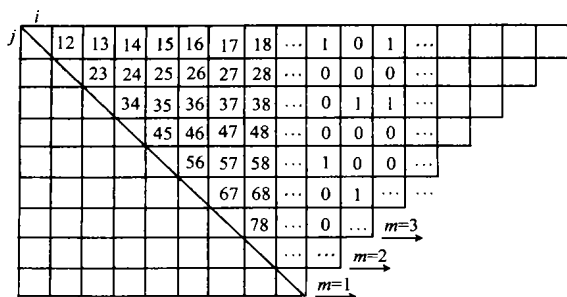


图 1 区间阈值累加法的基本原理示意
入维数取 15 个。

6 结论

第 5 节中的计算量估计很粗略,这是因为优化算法从计算思路已完全不同于标准算法,因此可比性较小,但是在减法运算量上的确减少许多;另一方面在比较操作中虽然不能精确估计运算量,但是从所有向量在所有距离阈值下的比较简化到局部区间向量的距离比较,已经能够充分体现出其优越性。当固定时间序列的长度、距离阈值的个数和最小阈值的大小,当仅仅改变计算嵌入维数个数的多少时,两种优化算法运算时间没有明显的增加趋势,而标准算法呈现出随嵌入维数区间增大而运算时间急剧增长的趋势。这就体现出以区间的形式对表格填写实施批处理的优势,在实验中也得到了充分的认证。

参考文献:

- [1] Wang Ying, Han Yueqiu, Mao Erke. Fractal Dimension Studying of Random Sequence [A]. Proc. ICSP '96 [C]. Beijing: ICSP, 1996. 257 - 260.

- [2] James Theiler. Efficient Algorithm for Estimating the Correlation Dimension from a Set of Discrete Points [J]. Phys. Rev. 1987, 36 (9) : 4456 - 4462.
- [3] 张涛, 文学章. 吸引子维数计算的几点改进 [J]. 浙江大学学报 (自然科学版). 1993, 27 (5) : 673 - 679.

作者简介:



周 越 男, 1969 年生于江苏常州, 1997 年获得西北工业大学机械电子工程硕士学位, 2000 年获得西北工业大学信号与信息处理专业的博士学位, 现在上海交通大学图像处理与模式识别研究所做博士后研究, 主要研究方向为: 非高斯信号处理, 图像处理与模式识别, 已在期刊上发表二十余篇文章。

杨 杰 男, 1964 年生于天津, 1988 年获上海交通大学模式识别与智能系统专业硕士学位, 1994 年获德国汉堡大学计算机科学博士学位, 现担任上海交通大学图像处理及模式识别研究所副所长、教授、博士生导师, 已主持承担多项国家级科研项目 (国家自然科学基金、国家 863 - CIMS 主题等), 已发表专著一本 (德国 Shaker Verlag 出版社出版), 国际国内学术期刊上论文七十多篇。