基于弧度距离的时间序列相似度量

丁永伟*①　杨小虎①　陈根才①　Kavs A J②
①(浙江大学计算机科学与技术学院　杭州　310027)
②(美国道富银行　波士顿　02111)

摘　要：时间序列的近似表示和相似度量是时间序列数据挖掘的重要任务之一，是进行相似匹配的关键。该文针对现有的各种基于分段线性表示(Piecewise Linear Representation，PLR)相似度量方法存在的序列长度依赖和多分辨率条件下的潜在识别误差等缺点，提出了一种序列分段线性弧度表示和基于弧度距离的相似度量方法，实现了序列的快速在线分割和相似度计算。该方法简洁直观，利用分段弧度对分段趋势进行细粒度划分来保留序列主要形态特征，有效地提高了度量结果的准确性和多分辨率条件下的稳定性。该方法具有序列分割算法独立性特点，可用于时间序列的相似查询、模式匹配、分类和聚类。

关键词：时间序列；分段线性表示；分段趋势；弧度距离；相似性
中图分类号：TP311　　**文献标识码**：A　　**文章编号**：1009-5896(2011)01-0122-07
DOI: 10.3724/SP.J.1146.2010.00136

Radian-distance Based Time Series Similarity Measurement

Ding Yong-wei①　Yang Xiao-hu①　Chen Gen-cai①　Kavs A J②
①(*College of Computer Science and Technology*, *Zhejiang University*, *Hangzhou* 310027, *China*)
②(*State Street Corporation*, *Boston*, *Massachusetts* 02111, *United States*)

Abstract: Time series approximation representation and similarity measurement is one of the fundamental tasks in time series data mining, and the key to similarity matching. In view of shortcomings of various existing PLR (Piecewise Linear Representation) based similarity measure approaches, like series-length dependent issue and potential recognition error under multi-resolution, a radian based time series piecewise linear representation and radian-distance based similarity measurement are presented to cater for the rapid online segmentation and similarity calculation in this paper. The proposed method is really simple but intuitive, it retains major shape features of the series by using segment radian for fine grained division, and effectively improves the accuracy and reliability of the measurement under multi-resolution. This method is segmentation algorithm independent and can be further applied to similarity query, pattern matching, classification and clustering for time series.

Key words: Time series; Piecewise Linear Representation (PLR); Segment trend; Radian-distance; Similarity

1　引言

时间序列数据广泛存在于包括商业、金融、医药、天文气象、航空航天等各种不同的应用领域[1]。和传统的包含离散数据类型的事务数据不同，时间序列数据通常是数值型的，且具有时间连续性特点[2]。更重要的是，时间序列数据往往规模巨大，甚至在持续不断地增长，因此，对时序数据的处理、分析和挖掘变得异常困难。时间序列的相似搜索一直是时间序列数据挖掘(Time Series Data Mining，TSDM)的研究热点[3–5]，这其中，序列的近似表示和相似度量是解决相似性搜索的关键[6]，直接决定了相似匹配的效果。

传统的基于点距离的欧式距离由于缺乏分段趋势信息，不具备形态识别能力，也不能识别时间序列在不同分辨率下的模式变化[7]。针对欧式距离在描绘分段趋势上的先天不足，文献[8]提出了三元分段趋势的模式划分，通过计算模式距离从而度量两个等长序列的趋势差异程度。文献[9]在此基础上提出了基于七元模式的形态距离，一定程度上提高了度量的精度。模式距离和形态距离虽然都保留了时间序列的分段趋势信息，但本质上都是基于对序列分段模式的有限划分，因此，任意两个序列对应分段间的距离值都是离散的。相似匹配的精确程度依赖于模式划分粒度。此外，文献[10]提出的夹角距离，虽然具有平移和旋转不变性优点，但是由于没有保留序列的分段趋势信息，会错误地认为图 1 所示的两个序列全相似(即夹角距离为 0)。序列 S_1 和 S_2 形态特征对称于 X 轴(S_1 单调递增，S_2 单调递减)，S_1 和 S_2 显然不相似。

2010-02-05 收到，2010-07-26 改回
*通信作者：丁永伟　ywding@zju.edu.cn

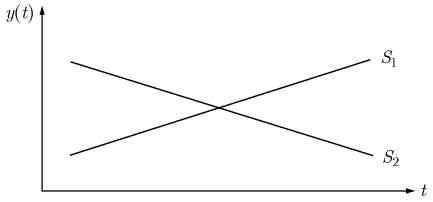


图 1 文献[10]无法区分的两个序列

针对现有方法存在的不足，本文首先提出一种时间序列的分段线性弧度表示(PLR_RAD)，定义了弧度距离及基于弧度距离的相似度量。该方法继承了模式距离的多分辨率特性及夹角距离的几何优点，在实现序列快速分段压缩的同时完整地保留了序列分段的局部趋势，同时具有序列分割算法独立性特点。此外，分段弧度值的连续性特点为时间序列提供了细粒度的模式划分，保证了相似度量的精度，能有效改善相似查询、模式匹配和序列分类的效果。

2 时间序列的分段线性弧度表示

为清晰起见，对文中常见的符号及其定义约定如表 1 所示。

表 1 符号定义

符号	定义
S	原始时间序列
S'	分段线性表示的时间序列
$\tilde{S}$	齐弧度时间序列
n	S 长度
K	S' 分段数
M	$\tilde{S}$ 分段数
rad	分段弧度
CRE	Cumulative Radian Error(累积弧度差)
TI	Turning Intensity(转向强度)
TP	Turning Point(转向点)
PP	Peering Point(对等点)
ppRad	对等分段弧度
D_{euc}	欧式距离
D_{ptn}	模式距离
D_{shape}	形态距离
D_{ia}	夹角距离
D_{rad}	弧度距离

2.1 序列的分段线性表示

时间序列的分段线性表示(Piecewise Linear Representation，PLR)[11]最早由 Keogh 于 1997 年引入时间序列数据挖掘领域。

直观上，PLR 的基本思想是用 K 条连续相邻的直线段来近似表示长度为 n 的时间序列，其中 $K << n$ [12]。时间序列也可以看作是 n 维空间的一个点，是一种特殊的多维数据。时间序列近似表示的目的是降低原始序列的维度，达到降低索引空间开销和加快搜索的目的。相较于其他常见的时间序列表示方法诸如单值分解(SVG)、离散傅里叶变换(DFT)和离散小波变换(DWT)，PLR 由于算法相对简单且更加符合人们的直观经验，因此被广泛采用[13,14]。时间序列的分段线性表示如式(1)所示，其中 K 表示序列 S 划分的分段数目，t_i 是第 i 个分段的结束时刻，$t_K = n$ 。

$$S' = \{(y_{1L}, y_{1R}, t_1), (y_{2L}, y_{2R}, t_2), \cdots, (y_{iL}, y_{iR}, t_i), \cdots, (y_{KL}, y_{KR}, t_K)\} \tag{1}$$

PLR 方法一般是通过选取序列中的特殊数据点[15]或者视觉重要点(Perceptually Important Point，PIP)[2,16]来近似拟合原始序列，达到压缩数据的目的。

2.2 时间序列的分段弧度集

定义 1 分段弧度(rad)是指时间序列中相邻两点构成的直线与时间轴的夹角(锐角)弧度。规定夹角在第 1 象限弧度为正值，在第 4 象限则为负值。

根据定义 1，图 2 中，$\alpha < 0, \beta > 0$ 。式(2)给出了时间序列分段弧度的计算方法，因此，$\text{rad}(\alpha) < 0$ ，$\text{rad}(\beta) > 0$ 。

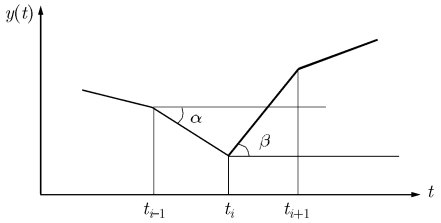


图 2 时间序列分段弧度

$$\text{rad}_i = \arctan\frac{y_{i+1} - y_i}{\Delta t}, \quad i \in [1, n-1] \tag{2}$$

理论上，$\text{rad}_i \in (-\pi/2, \pi/2)$，由于正规化后的序列 y 值范围为[0,1]，故有 $\Delta y \in [-1, 1]$，通常，取 $\Delta t = 1$，因此 $\text{rad}_i \in [-\pi/4, \pi/4]$ 。按照经典的三元模式划分，可得到趋势值的分段函数如下。显然，三元模式划分的趋势值是离散的，比较粗糙，而用分段弧度表示的趋势值是连续的，更为精确。

$$\text{segTrend}_i = \begin{cases} 1, & (\text{rad}_i > 0) \\ 0, & (\text{rad}_i = 0) \\ -1, & (\text{rad}_i < 0) \end{cases} \tag{3}$$

根据式(1)和定义 1，可以得到时间序列的弧度集表示，如式(4)所示。

$$S'=\{(\mathrm{rad}_1,t_1),(\mathrm{rad}_2,t_2),\cdots,(\mathrm{rad}_i,t_i),\cdots,(\mathrm{rad}_K,t_K)\} \quad (4)$$

与式(1)不同的是，这里 t_i 是第 i 个分段的起始时刻，t_{i+1} 为第 i 个分段的终止时刻，故第 i 个分段趋势的保持时间为 $t_{i+1}-t_i$，需要说明的是第 K 个分段的终止时刻为 t_n。因此时间序列 S 可以转化为包含分段趋势信息(分别是上升趋势、下降趋势、保持趋势)[17]的分段弧度集表示。

2.3 基于累积弧度差的序列分段线性分割算法

定义 2 累积弧度差(Cumulative Radian Error, CRE)是指线性分段表示的时间序列中前一转向时刻到当前时刻连续相邻分段间的弧度差之和。

$$\mathrm{CRE}_i=\sum_{\mathrm{j=last\_tp\_idx+1}}^{i}\left(\mathrm{rad}_j-\mathrm{rad}_{j-1}\right) \quad (5)$$

该算法首先将长度为 n 的原始时间序列表示转化为长度为 $n-1$ 的弧度集表示序列。然后根据设定的累积弧度差阈值(CRE_Th)和转向强度(TI)确定转向点，进而得到长度为 K 的分段序列，其中 $K<<n$。

式(2)中，$\mathrm{rad}_i\in[-\pi/4,\pi/4]$，可知 $|\Delta\mathrm{rad}|\in[0,\pi/2]$，因此 $|\Delta\mathrm{CRE}|\in[0,\pi/2]$。取值 0 意味着对原始序列进行分段线性分割之后所有的数据点都得到保留。而在实际情况中，根据序列波动特征以及目标压缩率，累积弧度差阈值通常取 0.01~1 之间。一般地，序列波动越剧烈，阈值的取值就越高，序列波动越平缓，阈值取值就越低；而目标压缩率越高，阈值的取值就越高，反之，取值就越低。

引入转向强度($\mathrm{TI}\ge 1$)的目的是为了进一步减小序列弧度同向连续缓慢变化(即弧度差小于阈值的转向)造成的拟合误差，尽可能地保留原始序列整体形态特征。转向强度也是序列特征相关的，可根据不同领域时序数据的波动特点酌情设置，序列波动幅度越小，转向强度取值越高，一个强度单位值为 0.0001，即 TI=1.0001。而对于波动相对剧烈的时间序列，可以关闭转向强度，即 TI=1。实际应用中，转向强度一般取 1.0001~1.01 之间。

图 3 给出了基于累积弧度差的序列分段线性分割算法伪代码。

算法名称：基于累积弧度差的时间序列分段线性分割算法(CRE based Piecewise Linear Segmentation， PLS_CRE)

算法输入：(1)长度为 n 的时间序列 S；(2)累积弧度差阈值 $\mathrm{CRE\_Th}>0$；(3)转向强度 $\mathrm{TI}\ge 1$；

算法输出：基于弧度集表示的时间序列 S'，如式(4)所示。

```
S' = ∅
normalize(S) // 对S进行标准化处理
last_tp_idx = 1
rad_err_n = rad_n = 0
CRE_1 = rad_err_1 = rad_1 = calc_rad(1)
S' = {(x_1, t_1)} // 加入第1个点
for i = 2 : n − 1
{
  rad_i = calc_rad(i);
  rad_err_i = rad_i − rad_{i−1}
  CRE_i = rad_err_i + CRE_{i−1} × TI^{i−last_tp_idx}
  if(|CRE_i| >= CRE_Th)
  {
    S' = S' ∪ {(x_i, t_i)} // 加入该转向点
    last_tp_idx = i
  }
}
S' = S' ∪ {(x_n, t_n)} // 加入最后一个点
return S'
```

图 3 PLS_CRE 算法伪代码

图 4 为长度为 2735 的纽约标准普尔 500 指数(S&P500)序列压缩前后对比图，转向强度取值 1.0001。图 4(a)为包含全部 2735 个原始数据点的原始序列图，图 4(b)为仅由 136 个压缩之后的原始数据点经线性插值得到的拟合序列图。压缩率为 95.03%，拟合误差仅为 0.623。PLS_CRE 分割算法只需对目标序列进行一次顺序扫描，时间复杂度为 $O(n)$。该算法在实现了序列的快速压缩的同时，有效地保留了序列的主要形态特征。支持序列的动态在线分割。

3 时间序列的弧度距离

3.1 时间序列的模式距离[8]

时间序列模式表示时间序列中某个子序列单一的变化趋势(上升、保持、下降)。模式距离是表示具有相同保持时间长度的两个模式的距离。

$$D_M(s_i,s_j)=\left|m_i-m_j\right|$$

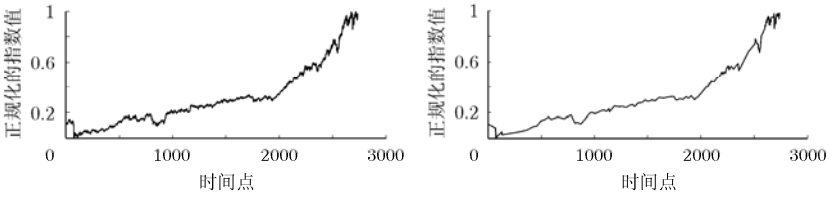
时间序列的模式距离是表示具有相同长度两个序列趋势的差异程度。基于分段线性表示的模式距离具有多分辨率的特性。

3.2 弧度集的时刻对等

通常情况下，任意两个等长序列 S_1 和 S_2 在线性分段后各端点不会完全对齐，相应地，每个分段的长度(也称分段趋势保持时间)也不尽相同，为了计算序列间的弧度距离，必须对分段后的序列进行时刻对等，也叫等模数(EPN)[8]或齐序列[9]处理。序列中任一非转向点(turning point)的对等点(peering point)弧度值等于前一转向点的弧度值。如图 5 所示，实心点为转向点，空心点为对等点。

弧度集时刻对等算法见图 6。

算法名称：线性分段弧度集对齐算法(Piecewise



(a) 压缩前原始序列 (全部 2735 个原始数据点)　　(b) 压缩后拟合序列 (136 个原始数据点)

图 4 利用 PLS_CRE 算法压缩的纽约标准普尔 500 指数序列对比图

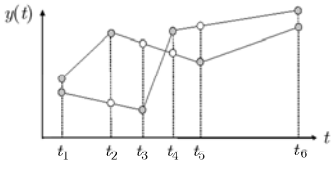


图 5 时间序列线性分段弧度集对齐示意图

Linear Radian-set Alignment, PLRA)

算法输入：长度均为 K 的分段线性弧度集表示序列对 (S_1', S_2')

算法输出：弧度集对齐的序列对 $(\tilde{S}_1, \tilde{S}_2)$

$\tilde{S}_1 = S_1', \tilde{S}_2 = S_2'$
$\text{last\_tp\_idx\_in\_}S_1 = 1$
$\text{last\_tp\_idx\_in\_}S_2 = 1$
for $i = 2 : K$
{
  if(x_{1i} is TP and x_{2i} is not TP)
  {
    $x_{2i}.\text{ppRad} = x_{2(\text{last\_tp\_idx\_in\_}S_2)}.\text{ppRad}$
    $= x_{2(\text{last\_tp\_idx\_in\_}S_2)}.\text{rad}$
    $\tilde{S}_2 \cup \{(x_{2i}, t_i)\}$
    $\text{last\_tp\_idx\_in\_}S_1 = i$
  }
  if(x_{1i} is not TP and x_{2i} is TP)
  {
    $x_{1i}.\text{ppRad} = x_{1(\text{last\_tp\_idx\_in\_}S_1)}.\text{ppRad}$
    $= x_{1(\text{last\_tp\_idx\_in\_}S_1)}.\text{rad}$
    $\tilde{S}_1 \cup \{(x_{1i}, t_i)\}$
    $\text{last\_tp\_idx\_in\_}S_2 = i$
  }
}

图 6 弧度集对齐算法伪代码

3.3 时间序列的弧度距离

定义 3　给定两个弧度集时刻对等后的线性分段表示时间序列 $\tilde{S}_1$ 和 $\tilde{S}_2$，

$$\tilde{S}_1 = \{\text{rad}_1^{'}, \text{rad}_2^{'}, \cdots, \text{rad}_i^{'}, \cdots, \text{rad}_M^{'}\}$$

$$\tilde{S}_2 = \{\text{rad}_1^{''}, \text{rad}_2^{''}, \cdots, \text{rad}_i^{''}, \cdots, \text{rad}_M^{''}\}$$

M 为序列弧度集对齐之后的分段数，一般情况下，$K < M << n$。对应两个分段间的弧度距离，即分段趋势差异度量为

$$D_{\text{rad}}(\text{rad}_i^{'}, \text{rad}_i^{''}) = \left|\text{rad}_i^{'} - \text{rad}_i^{''}\right| \tag{6}$$

因此，根据式(2)和式(6)，可以得到序列 $\tilde{S}_1$ 和 $\tilde{S}_2$ 的弧度距离(S_1 和 S_2 的近似弧度距离)计算公式如下，

$$D_{\text{rad}}(S_1, S_2) \approx D_{\text{rad}}(\tilde{S}_1, \tilde{S}_2)$$

$$= \frac{\sum_{i=1}^{M} \left|\Delta y_i^{'} - \Delta y_i^{''}\right| \times \left|\text{rad}_i^{'} - \text{rad}_i^{''}\right| \times \Delta t_{\text{rad}_i}}{\pi \times (n-1)} \in [0,1] \tag{7}$$

Δy_i 为第 i 个分段内 y 值的变化幅度，Δt_{rad_i} 为自 t_i 时刻起弧度 rad_i 的保持时间，n 为原始序列的长度，

$$\Delta y_i = y_{i+1} - y_i,\ \ \Delta t_{\text{rad}_i} = t_i - t_{\text{last\_tp\_idx}},$$

$$n = \sum_{i=1}^{M} \Delta t_{\text{rad}_i} \tag{8}$$

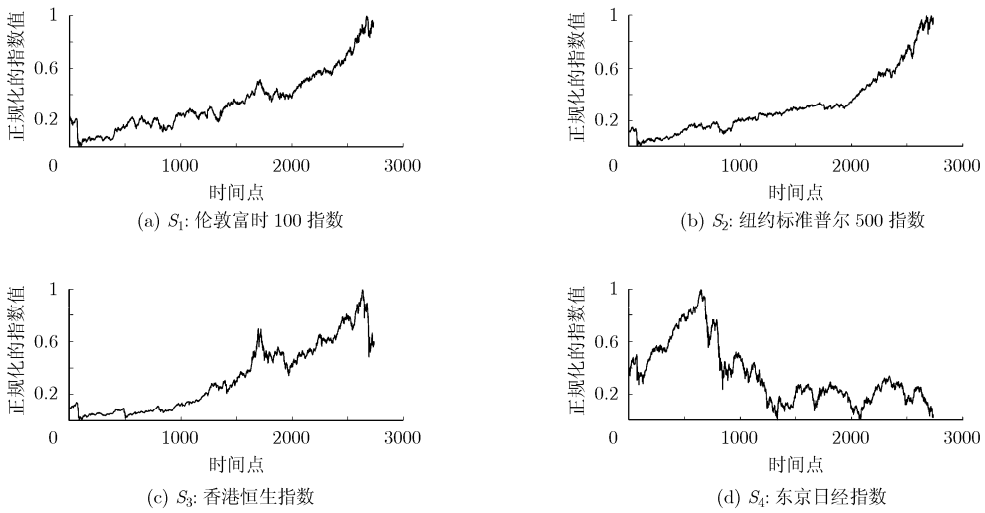
显然，两个序列间的弧度距离越小，两个序列间的形态越相似。距离为 0 说明两个序列完全一致或全相似。与模式距离和形态距离不同的是，弧度距离是对分段模式的无限划分，且是连续的。对于长度均为 n 的序列对 (S_1, S_2)，其欧式距离的取值范围是序列长度依赖的，即 $D_{\text{euc}}(S_1, S_2) \in [0, n)$。

4　实验结果及分析

为保证对比实验结果的公正性，采用文献[8-10]使用的金融数据集，选取从 1987 年 7 月 9 号开始到 1997 年 12 月 31 日止 2735 个有效交易日内全球 4 大证券市场指数，分别为 S_1(FTSE100，伦敦富时 100 指数)、S_2(SPCOMP，纽约标准普尔 500 指数)、S_3(HNGKNKI，香港恒生指数)和 S_4(JAPDOWA，日经指数)。在对给定序列进行压缩之前，先对序列数据进行正规化处理，使得各数据点的 y 值落在[0,1]范围内，如图 7 所示。

通过两组实验来验证本文提出的弧度距离在相似度量上的有效性和优越性，实验 1 是比较分别采用模式距离、形态距离、夹角距离和弧度距离进行相似度量的准确性，用正规化后未经压缩原始序列的欧式距离(见表 2)作为度量标准。实验 2 用于验证在不同分辨率下基于弧度距离相似度量结果的稳定性和一致性。

实验 1　首先将长度为 2735 的 4 个指数序列用 PLS_CRE 分割算法压缩至 80 段，然后进行距离计算，结果如表 3 所示，可见，弧度距离对各序列对相似度的度量结果与欧式距离一致，即 S_1 与 S_2 最相似，S_3 与 S_4 最不相似，序列对(S_1,S_3)和(S_2,S_3)的差异相当，符合直观判断。模式距离认为，S_2 和 S_4 的趋势差异最大，形态距离认为 S_2 和 S_3 差异最小，夹角距离则认为 S_1 和 S_4 最不相似。这 3 种距离度量与



(a) S_1: 伦敦富时 100 指数　(b) S_2: 纽约标准普尔 500 指数

(c) S_3: 香港恒生指数　(d) S_4: 东京日经指数

图 7 全球 4 大证券市场指数序列

表 2 未经压缩的正规化序列的欧式距离

距离类型	S_1 与 S_2	S_1 与 S_3	S_1 与 S_4	S_2 与 S_3	S_2 与 S_4	S_3 与 S_4
欧式距离(D_{euc})	**2.879**	4.647	21.111	5.463	21.344	**23.191**

表 3 模式、形态、夹角、弧度距离比较(采用 CRE 分割算法)

距离类型	S_1 与 S_2	S_1 与 S_3	S_1 与 S_4	S_2 与 S_3	S_2 与 S_4	S_3 与 S_4
模式距离(D_{ptn})	**0.690**	0.751	0.924	0.836	**1.030**	0.945
形态距离(D_{shape})	0.054	0.082	0.081	**0.036**	0.073	**0.103**
夹角距离(D_{ia})	0.426	0.457	**0.545**	**0.398**	0.475	0.520
弧度距离(D_{rad})	**0.016**	0.027	0.076	0.031	0.057	**0.079**

直观判断结果存在较大偏差。为了进一步说明弧度距离是序列分割算法独立的，分别采用文献[2]中提出的PIP分割算法和文献[18]中提出的PF分割算法将上述4个实验序列压缩至80段，并计算各距离值，表 4 和表 5 列出两种分割算法下的实验结果。可见，模式距离、形态距离以及夹角距离在对序列的相似度量上存在潜在误差。而弧度距离对于序列相似性的判断较其他距离更为准确，并且不依赖于某种特定的序列分割算法。实验同时发现，由于弧度距离是基于序列的分段线性弧度表示，因此采用 PLS_CRE 分割算法只须对序列进行一遍扫描，效率最高。

实验 2 为了证明弧度距离在多分辨条件下相较于欧式距离、模式距离、形态距离以及夹角距离的稳定性优势，实验 2 分别将序列压缩至 5 种不同分段数进行比较，距离计算结果详见表 6。欧式距离在分段数为 50、100 和 200 时，模式距离在各分段数，形态距离在分段数为 50 和 100 时，夹角距离在分段数为 80 时，对序列对的趋势差异识别不够合理，均未能准确地判断出差异最大的序列对。相对地，基于弧度距离的相似度量在不同分辨率下对于序列对的相似性都保持了持续稳定的识别能力。在各种分辨率下，使用弧度距离进行相似度量得到的结果与视觉判断完全一致，即 S_1 与 S_2 最相似，S_3 与 S_4 最不相似，适用于序列的相似性分析、分类和聚类。

5 结束语

基于分段弧度的时间序列近似表达及时间序列的弧度距离能够有效地度量序列的形态和分段趋势相似性，实验证明，弧度距离算法简洁，同时具有稳定的多分辨率特性，能够准确地比较序列对的趋势差异，同时具有序列分割算法独立性的特点，为序列相似性分析提供了新的方法，可用于时序数据的相似查询、模式匹配、分聚类及其他挖掘任务。

表 4 模式、形态、夹角、弧度距离比较(采用 PIP 分割算法)

距离类型	S_1 与 S_2	S_1 与 S_3	S_1 与 S_4	S_2 与 S_3	S_2 与 S_4	S_3 与 S_4
模式距离(D_{ptn})	**0.205**	0.401	1.064	0.357	**1.283**	1.164
形态距离(D_{shape})	**0.026**	0.081	0.195	0.071	**0.312**	0.172
夹角距离(D_{ia})	0.603	0.674	**0.704**	**0.565**	0.590	0.655
弧度距离(D_{rad})	**0.027**	0.061	0.169	0.077	0.103	**0.184**

表 5 模式、形态、夹角、弧度距离比较(采用 PF 分割算法)

距离类型	S_1 与 S_2	S_1 与 S_3	S_1 与 S_4	S_2 与 S_3	S_2 与 S_4	S_3 与 S_4
模式距离(D_{ptn})	**0.633**	0.849	0.842	0.952	**0.970**	0.699
形态距离(D_{shape})	**0.053**	0.095	**0.112**	0.076	0.087	0.081
夹角距离(D_{ia})	0.664	0.742	**0.744**	0.659	**0.656**	0.732
弧度距离(D_{rad})	**0.035**	0.078	0.199	0.063	0.208	**0.239**

表 6 多分辨率下欧式、模式、形态、夹角、弧度距离比较

分段数	距离类型	S_1 与 S_2	S_1 与 S_4	S_2 与 S_4	S_3 与 S_4
50	D_{euc}	0.448	5.562	**5.652**	4.706
	D_{ptn}	1.455	**1.481**	1.175	1.051
	D_{shape}	0.122	0.285	**0.072**	0.371
	D_{ia}	0.569	0.605	0.639	0.647
	D_{rad}	0.120	0.463	0.528	0.599
80	D_{euc}	0.743	5.132	5.359	5.506
	D_{ptn}	0.690	0.924	**1.030**	0.945
	D_{shape}	0.054	0.081	0.073	0.103
	D_{ia}	0.426	**0.545**	0.475	0.52
	D_{rad}	0.017	0.076	0.057	0.079
100	D_{euc}	0.701	7.27	**7.471**	6.856
	D_{ptn}	0.958	**0.665**	1.13	1.141
	D_{shape}	0.035	**0.246**	0.042	0.152
	D_{ia}	0.584	0.646	0.623	0.669
	D_{rad}	0.035	0.299	0.227	0.359
200	D_{euc}	0.982	8.769	**9.790**	8.922
	D_{ptn}	0.741	**1.044**	1.032	0.806
	D_{shape}	0.054	0.086	0.117	0.133
	D_{ia}	0.611	0.628	0.647	0.66
	D_{rad}	0.021	0.063	0.075	0.135

参 考 文 献

[1] Aigner W, Miksch S, and Müller W, *et al.*. Visual methods for analyzing time-oriented data[J]. *IEEE Transactions on Visualization and Computer Graphics* (*TVCG*), 2008, 14(1): 47–60.

[2] Fu T, Chung F, and Luk R, *et al.*. Stock time series pattern matching: template-based vs. rule-based approaches[J]. *Engineering Applications of Artificial Intelligence*, 2007, 20(3): 347–364.

[3] 林子雨，杨冬青，王腾蛟. 用基于移动均值的索引实现时间序列相似查询[J]. 软件学报, 2008, 19(9): 2349–2361.
Lin Zi-yu, Yang Dong-qing, and Wang Teng-jiao. Similarity search of time series with moving average based indexing[J]. *Journal of Software*, 2008, 19(9): 2349–2361.

[4] 靳碧，荣冈. BT: 一种快速序列搜索算法[J]. 浙江大学学报(工学版), 2007, 41(4): 621–625.
Jin Bi and Rong Gang. BT: fast sequence search algorithm[J]. *Journal of Zhejiang University* (*Engineering Science*), 2007, 41(4): 621–625.

[5] Johannes Aßfalg, Hans-Peter Kriegel, and Peer Kröger, *et al.*. Probabilistic similarity search for uncertain time series[C]. Proceedings of the 21st International Conference on Scientific and Statistical Database Management (SSDBM), New Orleans, LA, USA, Jun. 2–4, 2009: 435–443.

[6] Al-Naymat G and Taheri J. Effects of dimensionality reduction techniques on time series similarity measurement[C]. The 6th ACS/IEEE International Conference on Computer Systems and Applications, Doha, Qatar, Mar.31–Apr.4, 2008: 188–195.

[7] Shatkay H and Zdonik S B. Approximate queries and representations for large data sequences[C]. Proceedings of the 12th International Conference on Data Engineering, New Orleans, Louisiana, Feb.26–Mar.1, 1996: 536–545.

[8] 王达，荣冈. 时间序列的模式距离[J]. 浙江大学学报(工学版), 2004, 38(7): 795–798.
Wang Da and Rong Gang. Pattern distance of time series[J]. *Journal of Zhejiang University* (*Engineering Science*), 2004, 38(7): 795–798.

[9] 董晓莉，顾成奎，王正欧. 基于形态的时间序列相似性度量研究[J]. 电子与信息学报, 2007, 29(5): 1228–1231.
Dong Xiao-li, Gu Cheng-kui, and Wang Zheng-ou. Research on shape-based time series similarity measure[J]. *Journal of Electronics & Information Technology*, 2007, 29(5): 1228–1231.

[10] 张鹏，李学仁，张建业，等. 时间序列的夹角距离及相似性搜索[J]. 模式识别与人工智能, 2008, 21(6): 763–767.
Zhang Peng, Li Xue-ren, and Zhang Jian-ye, *et al*.. Included angle distance of time series and similarity search[J]. *Pattern Recognition and Artificial Intelligence*, 2008, 21(6): 763–767.

[11] Keogh E J. Fast similarity search in the presence of longitudinal scaling in time series databases[C]. Proceedings of the 9th International Conference on Tools with Artificial Intelligence, Newport Beach, CA, USA, Nov.3–8, 1997: 578–584.

[12] Keogh E J, Chu S, and Hart D, *et al*.. Segmentation Time Series: A Survey and Novel Approach[M]. Data Mining in Time Series Databases. Singapore: World Scientific Publishing Co., 2004: 1–22.

[13] Keogh E J and Pazzani M J. Relevance feedback retrieval of time series data[C]. Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Berkeley, CA, USA, Aug. 15–19, 1999: 183–190.

[14] Chang Pei-chann, Fan Chin-yuan, and Liu Chen-hao. Integrating a piecewise linear representation method and a neural network model for stock trading points prediction[J]. *IEEE Transactions on System, Man, and Cybernetics*, 2009, 30(1): 80–92.

[15] Perng C S, Wang H, and Zhang S R. Landmarks: a new model for similarity-based pattern querying in time series databases[C]. Proceedings of the 16th International Conference on Data Engineering, San Diego, CA, USA, Feb.28–Mar.3, 2000: 33–42.

[16] Phetking C, Sap M, and Selamat A. Identifying zigzag based perceptually important points for indexing financial time series[C]. Proceedings of the 8th International Conference on Cognitive Informatics, Hong Kong, China, Jun. 15–17, 2009: 295–301.

[17] Kirkpatrick C D and Dahlquist J R. Technical Analysis: The Complete Resource for Financial Market Technicians[M]. 1st edition, Canada: Financial Time Prentice Hall, 2006: 11–12.

[18] Pratt K and Fink E. Search for patterns in compressed time series[J]. *International Journal of Image and Graphics, World Scientific*, 2000, 2(1): 89–106.

丁永伟： 男，1983 年生，博士生，研究方向为金融时序数据处理与可视化分析.

杨小虎： 男，1966 年生，副教授，研究方向为金融信息学、软件再工程.

陈根才： 男，1950 年生，教授，博士生导师，研究方向为智能信息处理、数据库技术及数据挖掘.