文章编号: 1001-0920 (2014) 06-1033-06 **DOI:** 10.13195/j.kzyjc.2014.0388

基于不等长序列相似度挖掘的数据关联算法

关 欣, 孙贵东, 衣 晓, 郭 强

(海军航空工程学院 电子信息工程系，山东 烟台 264001)

摘 要: 针对不等长序列数据的关联问题, 提出基于滑动窗口的最优匹配增权法不等长序列相似度度量算法. 以较短序列作为滑动窗口遍历较长序列得到一组滑动相似度, 利用这组相似度形成最优权重, 加权得到不等长序列的相似度, 并根据相似度大小对序列数据进行关联判决, 以解决截断法相似度度量仅能反映截断序列局部相似度的问题. 仿真实验验证了所提出算法对不等长序列数据关联的有效性, 并对序列长度和量测误差等因素对相似度度量和关联效果的影响进行了讨论.

关键词: 数据关联；序列相似度；不等长度；滑动窗口；最优匹配增权

中图分类号: TN95 **文献标志码:** A

Data association algorithm based on unequal length sequence data similarity mining

GUAN Xin, *SUN Gui-dong*, *YI Xiao*, *GUO Qiang*

(Department of Electronics and Information Engineering，Naval Aeronautical and Astronautical University，Yantai 264001，China．Correspondent：SUN Gui-dong，E-mail：SDWHSGD@gmail.com)

Abstract: An optimal matching increasing weight algorithm for the unequal length sequence similarity measurement based on the sliding window is proposed to solve the unequal length sequence data association, which uses shorter sequences slide longer ones to get slidable similarity, forming the optimal weight with this similarity at the same time, then weighting the slidable similarity to get the unequal length sequence similarity. According to the degree of the sequence similarity, the judgment of the association of unequal length sequence data is obtained, which solves the local similarity problem of the truncated measurement algorithm. Simulation experiments show that the proposed algorithm can associate unequal length data effectively and also discuss the influence of the variation of sequence and measurement error on the sequence similarity and the association effect.

Keywords: data association；sequence similarity；unequal length；slide window；optimal matching increasing weight

0 引 言

传感器类型和量测周期的不同往往会产生不等长序列数组, 对不等长度序列数据相似度的挖掘和度量是处理此类数据关联问题的关键. 关于不等长序列相似度的研究, Agrawal等[1-3]在 1993 年发表了序列相似搜索的论文, 此后 Faloutsos 等[1,4]、Sang-wook 等[5-6]、Keogh 等[7-10]相继完善了序列相似度查询. 目前已有的离散序列数据相似查询主要有离散傅里叶变换[1,11-12]、奇异值分解[13-14]、离散小波变换[15-17]、分段合计近似[10,18-20]、动态时间弯曲[6-8]、分段线性表示[10,21-22]和分段多项式表示[23-24]等. 这些方法从不同角度对不等长序列进行度量, 但其实质都是先对不等长数据进行变换, 成为能够直接处理的等长度序列或某个变换域内的数据, 再利用等长序列或变换域的某种相似度度量对不等长数据进行相似度度量. Keogh 等[10]提出一种分段合计近似 PAA 方法, 对序列进行分段, 研究每段的近似表示, 然后按段进行相似度量, 此方法在处理不等长序列的度量时采用截断的方法, 极大地增加了度量误差.

为了克服 Keogh 等方法的缺点, 本文提出基于滑动窗口的最优匹配增权法不等长序列相似度度量算法. 首先, 以比较序列或参考序列中长度较短的序列作为滑动窗口, 沿长度较长的序列按单位长度依次滑动遍历整个长度序列, 得到一组滑动相似度; 然后, 在

收稿日期: 2014-03-20；**修回日期:** 2014-06-12.

基金项目: 国家自然科学基金重点项目(61032001)；教育部新世纪优秀人才支持计划项目(NCET-11-0872).

作者简介: 关欣(1978−), 女, 教授, 博士后, 从事智能信息处理、多源信息融合等研究；孙贵东(1989−), 男, 博士生, 从事信息融合理论、智能数据挖掘的研究.

加权融合滑动相似度得到序列相似度时, 采用最优匹配增权策略, 以达到突出匹配效果的目的; 最后, 根据序列相似度, 采用最大准则对序列类型传感器数据进行关联判决.

1 基本概念

1.1 序列的矩阵表示

定义 1 序列

$$\boldsymbol{S}_i = (S_{i1}, S_{i2}, \cdots, S_{in})^{\mathrm{T}}. \tag{1}$$

其中: S_{ij} 为序列 i 的第 j 个分量; n 为序列的长度, 记为 $|\boldsymbol{S}_i|$, 如果 $|\boldsymbol{S}_i| \neq |\boldsymbol{S}_j|$, 则称序列 $\boldsymbol{S}_i$ 和序列 $\boldsymbol{S}_j$ 为不等长序列.

由 m 条序列数据组成的某目标在多个特征参数描述下的量测数据可以通过矩阵的形式表示为

$$\boldsymbol{S} = (\boldsymbol{S}_1, \boldsymbol{S}_2, \cdots, \boldsymbol{S}_m), \tag{2}$$

其中 $\boldsymbol{S}$ 为序列矩阵.

1.2 序列的相似度度量

定义 2 序列矩阵 $\boldsymbol{S}$ 与 $\boldsymbol{Q}$ 之间的相似度

$$\mathrm{Sim}_e(\boldsymbol{S}, \boldsymbol{Q}) = \frac{1}{\dim(\boldsymbol{S})} \sum_{i=1}^{\dim(\boldsymbol{S})} \lambda_i \mathrm{Sim}_{ei}(\boldsymbol{S}_i, \boldsymbol{Q}_i). \tag{3}$$

其中: $\mathrm{Sim}_{ei}(\boldsymbol{S}_i, \boldsymbol{Q}_i)$ 为不等长序列 $\boldsymbol{S}_i$ 与 $\boldsymbol{Q}_i$ 之间的相似度度量, λ_i 为序列 i 的权重, $\dim(\boldsymbol{S})$ 为序列的数量.

1.3 序列的关联

对于查询序列矩阵 $\boldsymbol{S}$ 和比较参考序列矩阵 $\boldsymbol{Q}$, 给定任意一种度量 ρ, 如果

$$\rho(\boldsymbol{S}, \boldsymbol{Q}) \to 1, \tag{4}$$

或对于给定约束 ε, 使得

$$\rho(\boldsymbol{S}, \boldsymbol{Q}) \geqslant \varepsilon, \tag{5}$$

则称序列矩阵 $\boldsymbol{S}$ 与序列矩阵 $\boldsymbol{Q}$ 关联. 在问题的解决过程中, 常以相似度的大小度量关联的程度. 序列矩阵 $\boldsymbol{S}$ 与序列矩阵 $\boldsymbol{Q}$ 之间的相似度越大, 序列矩阵 $\boldsymbol{S}$ 与序列矩阵 $\boldsymbol{Q}$ 的数据关联度越高.

2 基于滑动窗口的最优匹配增权法不等长序列相似度度量算法

2.1 算法简述

对不等长序列矩阵 $\boldsymbol{S}$ 与 $\boldsymbol{Q}$ 的相似度度量的本质是其中不等长序列的度量问题, 假设有某两条物理关系对应的不等长序列 $\boldsymbol{S}_i$ 和 $\boldsymbol{Q}_i$, 令

$$|\boldsymbol{S}_i| < |\boldsymbol{Q}_i|. \tag{6}$$

文献 [10] 中的方法是将不等长序列中长序列多余的序列点赋值为 0, 直接舍去, 而本文采用一种滑动窗口的思想, 将长度较短的序列作为滑动窗口, 沿长度较长的序列依次滑动一个窗口单位, 直至遍历长序列的所有点, 如图 1 所示.

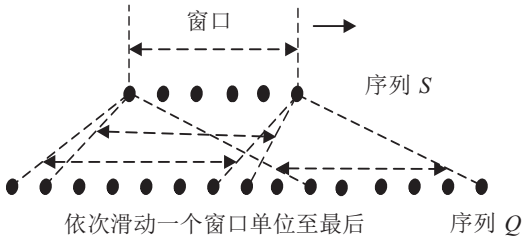


图 1 滑动窗口

$\boldsymbol{Q}_i$ 在 $\boldsymbol{S}_i$ 窗口内对应的子序列为

$$\boldsymbol{W}(\boldsymbol{Q}_i)_j = \boldsymbol{Q}_{i(j:j+|\boldsymbol{S}_i|-1)},$$
$$j = 1, 2, \cdots, |\boldsymbol{Q}_i| - |\boldsymbol{S}_i| + 1. \tag{7}$$

其中: $\boldsymbol{Q}_{i(j:j+|\boldsymbol{S}_i|-1)}$ 为窗口内的第 j 个子序列, 长度为窗口长度 $|\boldsymbol{S}_i|$, 共有 $|\boldsymbol{Q}_i| - |\boldsymbol{S}_i| + 1$ 个子序列.

$$\boldsymbol{Q}_{i(j:j+|\boldsymbol{S}_i|-1)} = (Q_{ij}, Q_{ij+1}, \cdots, Q_{ij+|\boldsymbol{S}_i|-1}). \tag{8}$$

在窗口滑动的过程中, 即时计算对应窗口长度序列的滑动相似度

$$\mathrm{Sim}_{ei}(\boldsymbol{S}_i, \boldsymbol{W}(\boldsymbol{Q}_i)_j) = 1 - \frac{D_{e2ij}(\boldsymbol{S}_i, \boldsymbol{W}(\boldsymbol{Q}_i)_j)}{D_{\mathrm{emax}}}. \tag{9}$$

其中: $D_{e2ij}(\boldsymbol{S}_i, \boldsymbol{W}(\boldsymbol{Q}_i)_j)$ 为窗口滑动过程中查询序列与参考序列中对应窗口长度序列之间的距离度量, D_{emax} 为距离度量中的最大值.

$$D_{e2ij}(\boldsymbol{S}_i, \boldsymbol{W}(\boldsymbol{Q}_i)_j) = \Big(\sum_{k=1}^{|\boldsymbol{S}_i|} (S_{ik} - Q_{i(j+k-1)})^2\Big)^{\frac{1}{2}}, \tag{10}$$

$$D_{\mathrm{emax}} = \max\{D_{e2ij}(\boldsymbol{S}_i, \boldsymbol{W}(\boldsymbol{Q}_i)_j), j = 1, 2, \cdots, |\boldsymbol{Q}_i| - |\boldsymbol{S}_i| + 1. \tag{11}$$

得到滑动相似度后, 加权组合相似度就可以得到两条不等长序列之间的相似度. 这里需要注意: 在滑动窗口的计算过程中, 得到的序列滑动相似度是不断变化的, 因此在加权组合滑动相似度时, 需要进行权重有效的分配, 以突出滑动相似度所对应序列的匹配程度.

2.2 权重确定

按照式 (9) ~ (11) 计算出滑窗过程中对应的滑动相似度向量

$$\mathbf{Sim}_{ueisw}(\boldsymbol{S}_i, \boldsymbol{Q}_i) = (\mathrm{Sim}_{ei}(\boldsymbol{S}_i, \boldsymbol{W}(\boldsymbol{Q}_i)_1), \mathrm{Sim}_{ei}(\boldsymbol{S}_i, \boldsymbol{W}(\boldsymbol{Q}_i)_2), \cdots, \mathrm{Sim}_{ei}(\boldsymbol{S}_i, \boldsymbol{W}(\boldsymbol{Q}_i)_{(|\boldsymbol{Q}_i|-|\boldsymbol{S}_i|+1)})). \tag{12}$$

根据滑动相似度的大小将权向量 w_{jsw} 定义为

$$w_{jsw} = \frac{\mathrm{Sim}_{ei}(\boldsymbol{S}_i, \boldsymbol{W}(\boldsymbol{Q}_i)_j)}{\sum_{j=1}^{|\boldsymbol{Q}_i|-|\boldsymbol{S}_i|+1} \mathrm{Sim}_{ei}(\boldsymbol{S}_i, \boldsymbol{W}(\boldsymbol{Q}_i)_j)},$$

$$j = 1, 2, \cdots, |\boldsymbol{Q}_i| - |\boldsymbol{S}_i| + 1. \tag{13}$$

这样定义可以突出滑窗过程中的滑窗相似度作用, 按照最优匹配增权法加权组合滑动相似度 $\mathrm{Sim}_{ei}(\boldsymbol{S}_i, \boldsymbol{W}(\boldsymbol{Q}_i)_j)$, 得到不等长序列 $\boldsymbol{S}_i$ 与 $\boldsymbol{Q}_i$ 之间的相似度

$$\mathrm{Sim}_{uei}(\boldsymbol{S}_i, \boldsymbol{Q}_i) = \sum_{j=1}^{|\boldsymbol{Q}_i|-|\boldsymbol{S}_i|+1} w_{jsw}\mathrm{Sim}_{ei}(\mathrm{S}_i, \boldsymbol{W}(\boldsymbol{Q}_i)_j). \tag{14}$$

得到不等长序列 $\boldsymbol{S}_i$ 与 $\boldsymbol{Q}_i$ 的相似度后, 根据序列在序列矩阵中所占的权重, 加权组合这些不等长序列的相似度, 得到不等长序列矩阵 $\boldsymbol{S}$ 与 $\boldsymbol{Q}$ 之间的相似度度量, 参照式 (3) 可以给出

$$\mathrm{Sim}_{ue}(\boldsymbol{S}, \boldsymbol{Q}) = \frac{1}{\dim(\boldsymbol{S})}\sum_{i=1}^{\dim(\boldsymbol{S})} \lambda_i \mathrm{sim}_{uei}(\boldsymbol{S}_i, \boldsymbol{Q}_i). \tag{15}$$

2.3 具体算法

根据第 2.1 节和 2.2 节所述, 传感器对目标进行量测的数据经前端处理得到待关联的目标序列数据作为查询序列. 将查询序列数据与其他传感器得到的目标序列数据进行相似度查询, 得到每两条序列的相似度, 融合序列之间的相似度得到描述某类目标的序列矩阵的相似度. 按照最大相似度关联准则, 判定序列矩阵相似度最大的序列数据关联, 即序列矩阵所描述的目标为同一类. 基于滑动窗口的最优匹配增权法不等长序列相似度度量的具体算法如下.

Input: 查询序列矩阵组 $\{\boldsymbol{S}^i\}$ 和参考序列矩阵组 $\{\boldsymbol{Q}^j\}$;

Output: 查询序列矩阵组 $\{\boldsymbol{S}^i\}$ 对应参考序列矩阵组 $\{\boldsymbol{Q}^j\}$ 的相似度度量向量组 $\{\mathbf{Sim}_{ue}^i\}$, 而

$$\mathbf{Sim}_e^i = (\mathrm{Sim}_e^i(1), \mathrm{Sim}_e^i(2), \cdots, \mathrm{Sim}_e^i(\mathrm{num}(\boldsymbol{Q}^j))),$$

其中 $\mathrm{num}(\boldsymbol{Q}^j)$ 为参考序列矩阵组 $\{\boldsymbol{Q}^j\}$ 的组数.

Step 1: 如果查询序列矩阵组 $\{\boldsymbol{S}^i\}$ 与参考序列矩阵组 $\{\boldsymbol{Q}^j\}$ 的度量对应维数不一致, 则执行 Step 2, 否则执行 Step 3.

Step 2: 将查询序列矩阵组 $\{\boldsymbol{S}^i\}$ 内行向量对应参考序列矩阵组 $\{\boldsymbol{Q}^j\}$ 的行向量物理意义进行重组, 形成新的查询序列矩阵组 $\{\boldsymbol{S}^i\}$.

Step 3: 对于某序列矩阵 $\boldsymbol{S}^i$, 分别抽取 $\boldsymbol{S}^i$ 和参考序列矩阵组 $\boldsymbol{Q}^j$ 中对应维数序列 $\boldsymbol{S}_k^i$ 和 $\boldsymbol{Q}_k^j$ (k 为矩阵 $\boldsymbol{S}^i$ 维数号), 形成比较向量组

$$\mathrm{Com}_{ek}^i = (\boldsymbol{S}_k^i, \boldsymbol{Q}_k^1, \boldsymbol{Q}_k^2, \cdots, \boldsymbol{Q}_k^{\mathrm{num}(\boldsymbol{Q}^j)}).$$

Step 4: 若 $|\boldsymbol{S}_k^i| < |\boldsymbol{Q}_k^j|$, 则 $\boldsymbol{S}_k^i$ 为窗, 否则 $\boldsymbol{Q}_k^j$ 为窗.

Step 5: 按照式 (12) ~ (14) 分别计算滑动相似度向量 $\mathbf{Sim}_{ueisw}^i(\boldsymbol{S}_k^i, \boldsymbol{Q}_k^j)$、最优匹配权重 w_{jsw}^i 和不等长序列 $\boldsymbol{S}_k^i$ 与 $\boldsymbol{Q}_k^j$ 的相似度向量 $\mathbf{Sim}_{uek}^i$.

Step 6: 如果 $k \leqslant \dim(\boldsymbol{S}^i)$, 则执行 Step 3 ~ Step 5, 否则执行 Step7.

Step 7: 按照式 (15) 计算序列矩阵 $\boldsymbol{S}^i$ 与参考序列矩阵组 $\{\boldsymbol{Q}^j\}$ 的相似度向量 $\mathbf{Sim}_{ue}^i$.

Step 8: 如果 $i \leqslant \mathrm{num}(\boldsymbol{S}^i)$, 则执行 Step 3 ~ Step 7, 否则执行 Step 9.

Step 9: 输出查询序列矩阵组 $\{\boldsymbol{S}^i\}$ 对应参考序列矩阵组 $\{\boldsymbol{Q}^j\}$ 的相似度度量向量组 $\{\mathbf{Sim}_{ue}^i\}$.

Step 10: 结束.

3 仿真分析

3.1 仿真环境

假设 A 地传感器对目标的雷达载频 RF、脉冲重复频率 PRF 和脉宽 PW 三类身份信息进行测量, 经前端数据处理, 得到两个序列数据矩阵 $\boldsymbol{S}^1$ 和 $\boldsymbol{S}^2$ 分别由 3 条序列组成, 代表 RF、PRF 和 PW 三类参数. 这里假设每条序列信息描述同一目标的身份数据, 不出现一条序列数据描述多个目标的情况. 同样, B 地传感器的量测数据分别用序列矩阵 $\boldsymbol{Q}^1$、$\boldsymbol{Q}^2$、$\boldsymbol{Q}^3$ 和 $\boldsymbol{Q}^4$ 表示, 其中矩阵内数据参数与序列矩阵 $\boldsymbol{S}^1$ 和 $\boldsymbol{S}^2$ 对应.

仿真数据序列 data 按下式产生:

$$\mathrm{data} = a + \alpha b. \tag{16}$$

其中: a 为服从均匀分布的离散序列值; b 为服从高斯分布的离散序列值; α 为高斯分布的标准差, 可以用来描述量测误差.

量测值服从的分布值范围和量测误差如表 1 所示. 假设在进行仿真时, 首先进行数据的标准化去量纲处理, 故数据在处理时暂不考虑数量级的问题.

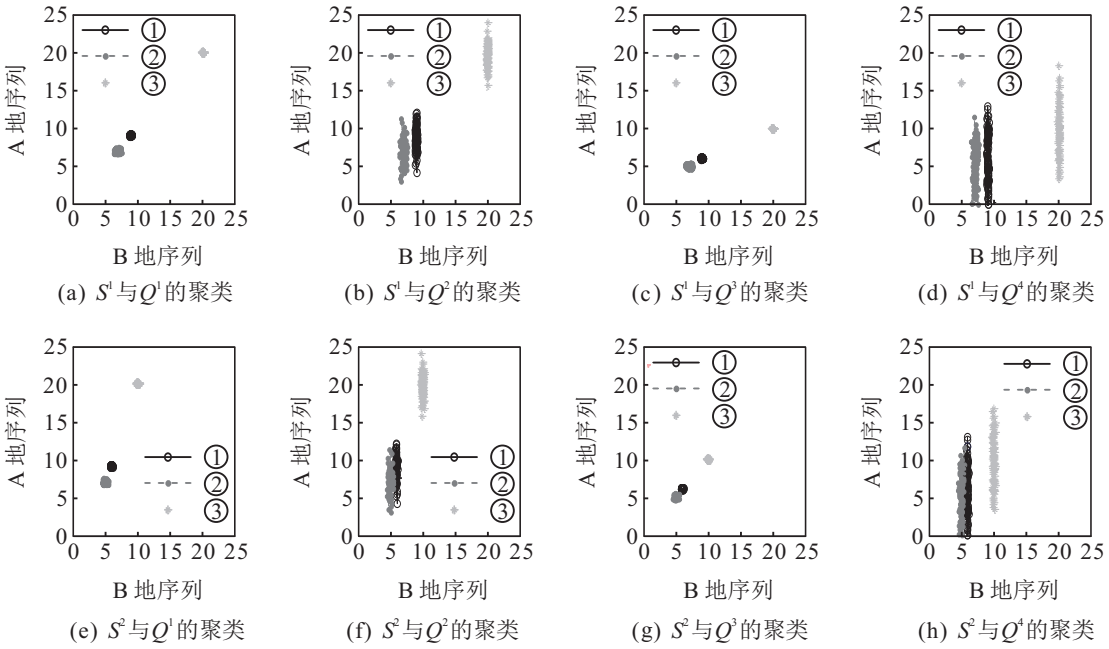
表 1　仿真数据

序列	$a \sim U(\mathrm{start}, \mathrm{end})$			$b \sim N$	α
	RF	PRF	PW	(S, T)	
$\boldsymbol{Q}^1$	(8.9, 9.1)	(18.9, 21.1)	(6.5, 7.5)	(0,1)	0.5
$\boldsymbol{Q}^2$	(8.5, 9.5)	(15.8, 24.2)	(5.2, 8.8)	(0,1)	0.5
$\boldsymbol{Q}^3$	(5.9, 6.1)	(8.9, 11.1)	(4.5, 5.5)	(0,1)	0.5
$\boldsymbol{Q}^4$	(4.9, 7.1)	(7.9, 12.1)	(3.4, 6.6)	(0,1)	0.5
$\boldsymbol{S}^1$	(8.8, 9.2)	(19.9, 20.1)	(6.1, 7.9)	(0,1)	0.5
$\boldsymbol{S}^2$	(5.5, 6.5)	(9.5, 10.5)	(4.7, 5.3)	(0,1)	0.5

3.2 仿真实验

3.2.1 不等长序列仿真实验

实验中, A 地不等长序列仿真数据采用传感器 100 个测量周期形成的长度为 100 的目标序列矩阵; B 地采用长度为 200 的序列矩阵. 其中, 为了突出对比效果, 前100长度数据的量测误差增大, 具体数据值按照表 1 产生, 仿真数据宏观上的二维对比如图 2 所示.



(a) S^1与Q^1的聚类 (b) S^1与Q^2的聚类 (c) S^1与Q^3的聚类 (d) S^1与Q^4的聚类

(e) S^2与Q^1的聚类 (f) S^2与Q^2的聚类 (g) S^2与Q^3的聚类 (h) S^2与Q^4的聚类

图 2 不等长序列仿真数据二维对比

在图2中: 横坐标为B地传感器量测序列数据大小, 纵坐标为A地传感器量测序列数据大小; 曲线①为载频数据, 曲线②为脉宽数据, 曲线③为脉冲重频数据. 图2(a)～图2(d)和图2(e)～图2(h)分别为$\boldsymbol{S}^1$、$\boldsymbol{S}^2$与$\boldsymbol{Q}^1$、$\boldsymbol{Q}^2$、$\boldsymbol{Q}^3$、$\boldsymbol{Q}^4$的聚类. 可以看出, 在数值上, 图2(a)和图2(c)聚类点斜率接近1, 说明A地量测序列数据$\boldsymbol{S}^1$、$\boldsymbol{S}^2$与B地量测数据$\boldsymbol{Q}^1$、$\boldsymbol{Q}^3$相似度高.

根据式(7)～(13)加权组合相似度得到A地量测数据所描述的两类目标与B地量测数据所描述的4类目标身份的相似度结果如表2所示.

表 2 A组量测数据与B组数据的相似度

序列	$\boldsymbol{Q}^1$	$\boldsymbol{Q}^2$	$\boldsymbol{Q}^3$	$\boldsymbol{Q}^4$
$\boldsymbol{S}^1$	0.7125	0.5266	0.2056	0
$\boldsymbol{S}^2$	0.1264	0.0759	0.7106	0.1787

由表2数据可知: A地量测数据所描述目标1与B地量测数据所描述第1类目标的相似度最大, A地量测数据所描述目标2与B地量测数据所描述第3类目标的相似度最大. 根据最大相似度关联准则可以得到: A地量测数据所描述目标1与B地量测数据所描述第1类目标数据关联, A地量测数据所描述目标2与B地量测数据所描述第3类目标数据关联.

3.2.2 算法性能分析

本组实验对本文提出的不等长序列的相似度度量算法性能进行分析, 与文献[10]的方法进行比较, 两种方法的相似度对比结果如表3所示.

由表3对比可知, 文献[10]算法由于在计算时采取了数据截断的方法, 不能很好地反应整条序列的相似度关系, 得到的只是局部相似度, 相似度数值均匀, 没有突出的结果. 本文算法在计算相似度时, 从整体考虑, 得到的相似度比文献[10]算法变化明显, 重点突出, 能够用于目标的关联.

表 3 相似度对比

算法	序列	$\boldsymbol{Q}^1$	$\boldsymbol{Q}^2$	$\boldsymbol{Q}^3$	$\boldsymbol{Q}^4$
本文	$\boldsymbol{S}^1$	0.7168	0.5424	0.2442	0
	$\boldsymbol{S}^2$	0.1164	0.0731	0.7173	0.1683
文献[10]	$\boldsymbol{S}^1$	0.4170	0.3061	0.2498	0.0272
	$\boldsymbol{S}^2$	0.1769	0.0262	0.4291	0.3677

由于处理时间与序列长度紧密相关, 从算法的相似度和处理的时间分别随序列长度的变化来讨论算法的性能. 在算法相似度和处理时间随序列长度变化的实验中, 将本文实验数据与文献[10]的实验数据进行对比分析, 将A地序列长度设为从0、50、100按步长50变化到500. 相应地, B地序列数据的长度为其两倍, 记录算法的处理时间和相似度的变化, 得到它们随序列长度的变化曲线分别如图3和图4所示.

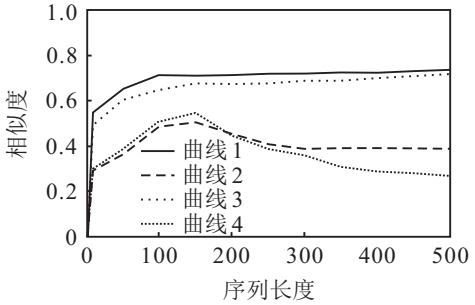


图 3 算法相似度随序列长度变化

在图3～图5中, 曲线1为本文数据本文算法结果, 曲线2为本文数据文献[10]结果, 曲线3为文献[10]数据本文算法结果, 曲线4为文献[10]数据文献[10]结果.

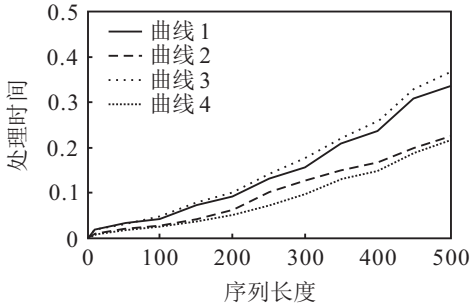


图 4　算法处理时间随序列长度变化

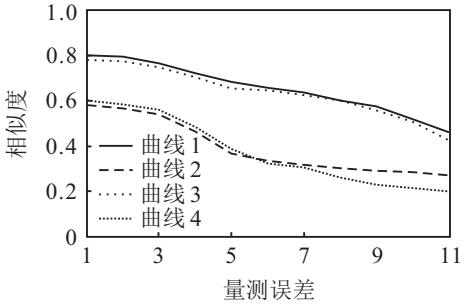


图 5　量测误差影响下的相似度变化

由图 3 可知, 一定范围内序列长度的增加可以提高相似度, 但是序列长度增大到一定数值时, 相似度不再提高. 这种变化的原因主要是少序列度量不可避免地会产生较大的偶然误差影响相似度, 而随着序列长度的增加, 误差减小, 相似度增加, 但达到一定限度时, 相似度饱和不再随序列长度变化. 对比本文算法和文献 [10] 算法, 无论采用本文实验数据还是文献 [10] 的实验数据, 本文算法的相似度结果明显优于文献 [10] 算法.

由图 4 可知, 随着序列长度的增大, 算法处理时间变长, 就具体数值而言, 算法的处理时间尽管增大, 但是变化不明显, 在合理的范围之内. 产生这种变化的原因主要是算法在滑动过程中直接计算了距离, 节省了处理时间, 说明算法在计算上具有时间优越性. 对比本文算法和文献 [10] 算法, 本文算法的计算时间比文献 [10] 算法有所增加, 主要是因为利用文献 [10] 算法计算时, 采取截断的思想, 而本文算法度量整条序列, 故时间有所增加.

综合图 3 和图 4 可知, 尽管本文算法时间有所增加, 但是在相似度稳定时, 对应的序列长度为 100 附近, 时间只增加 0.02 秒, 而相似度提高 20%, 所以相对于相似度提高的程度, 时间上增加的程度是微弱的, 就相似度提高的角度而言, 一定范围内计算时间的增加是必要的.

最后通过量测误差 α 的变化来检验算法的性能. 实验中, 量测数据的 α 值按步长 0.2 增加到 2, 得到的两种仿真数据条件下的最大相似度变化如图 5 所示.

由图 5 可知, 量测误差的增大会导致相似度变差, 特别是当量测误差增大到一定程度时, 相似度很低, 度量变得没有意义. 因此, 在测量过程中应该采取有效的措施减小误差以提高度量的性能. 对比本文算法和文献 [10] 算法的相似度变化可知, 在两种数据仿真条件下, 文献 [10] 算法得到的相似度在量测误差增加时, 相似度减小的速度都比本文算法快, 说明文献 [10] 算法对量测误差的变化更敏感, 而本文算法在误差允许的范围内适应量测误差的变化更稳定.

4　结　论

针对不等长序列不易度量的问题, 本文提出了基于不等长序列相似度挖掘的数据关联算法. 该算法对不等长序列数据有效地进行相似度度量, 并根据最大相似度原则实现对不等长序列数据所描述的目标进行关联. 算法相似度随着序列长度增加而提高直至某个最大值. 处理时间随序列长度增加而增长, 但是时间变化相对长度增加十分缓慢, 说明算法具有时间优越性. 相比于文献 [10], 本文计算的相似度结果更准确, 在计算序列相似度时比截断法的全局性更好. 随着量测误差增大, 两种算法计算的相似度变差, 但本文算法相比文献 [10] 算法对误差的适应能力强, 实际量测时应采取措施减小量测误差以提高计算的精度.

参考文献(References)

[1] Agrawal R, Faloutsos C, Swami A. Efficient similarity search in sequence databases[J]. Lecture Notes in Computer Science, 1993, 730: 69-84.

[2] Agrawal R, Sreenivas Gollapudi, Anitha Kannan, et al. Data mining for improving textbooks[J]. ACM SIGKDD Explorations Newsletter, 2012, 13(2): 7-19.

[3] Agrawal R, Amit Somani. System and method for distributed querying and presentation of information from heterogeneous data sources[P]. US: 7702617. 2010-04-20.

[4] Faloutsos C, Ranganathan M, Manolopoulos Y. Fast subsequence matching in time-series databases[J]. ACM SIGMOD Record, 1994, 23(2): 419-429.

[5] Seung-woo Kim, Sanghyun Park, Jung-im Won, et al. Privacy preserving data mining of sequential patterns for network traffic data[J]. Information Sciences, 2008, 178(3): 694-713.

[6] Sang-wook Kim, Sanghyun Park, Wealey W Chu. An index-based approach for similarity search supporting time warping in large sequence databases[C]. Proc of the 17th Int Conf on Data Engineering. Heidelberg: IEEE Computer Society Press, 2001: 607-614.

[7] Gustavo E, Batista, Eamonn J, et al. CID: An efficient complexity-invariant distance for time series[J]. Data Mining and Knowledge Discovery, 2014, 28(3): 634-669.

[8] Thanawin Rakthanmanon, Bilson Campana, Keogh E. Searching and mining trillions of time series subsequences under dynamic time warping[C]. Proc of the 18th ACM SIGKDD Int Conf on Knowledge Discovery and Data Mining. Beijing: ACM Press, 2012: 262-270.

[9] Keogh E. How to do good data mining research and get it published in top venues[C]. 2010 IEEE Int Conf on Data Mining. Sydney: IEEE Computer Society Press, 2010: 1219.

[10] Keogh E, Chakrabarti K, Pazzani M, et al. Dimensionality reduction for fast similarity search in large time series databases[J]. J of Knowledge and Information Systems, 2001, 3(3): 263-286.

[11] Xiang Lian, Lei Chen. Efficient similarity search over future stream time series[J]. IEEE Trans on Knowledge and Data Engineering, 2008, 20(1): 40-54.

[12] Luou Shen, Chenxi Lu, Fang Zhao, et al. Discrete fourier transformation for seasonal-factor pattern classification and assignment[J]. IEEE Trans on Intelligent Transportation Systems, 2013, 14(2): 511-516.

[13] Saari P, Eerola T. Semantic computing of moods based on tags in social media of music[J]. IEEE Trans on Knowledge and Data Engineering, 2013, 26(10): 2548-2560.

[14] Tang Jie, Zhang Jun, Geng Xinyu, et al. SVD based factorization technique for dual privacy protection data mining[C]. 2011 Int Conf on Computational and Information Sciences. Chengdu: IEEE Computer Society Press, 2011: 357-360.

[15] Radovic M, Djokovic M, Peulic A, et al. Application of data mining algorithms for mammogram classification[C]. 2013 IEEE Int Conf on Bioinformatics and Bioengineering. Chania: IEEE Computer Society Press, 2013: 1-4.

[16] Peng Zhu, Ming-sheng Zhao, Tian-chi He. A DWT based time series outlier data mining algorithm[C]. 2010 Int Conf on Electronics and Information Engineering. Kyoto: IEEE Computer Society Press, 2010: 239-241.

[17] 张海勤, 蔡庆生. 基于小波变换的时间序列相似模式匹配[J]. 计算机学报, 2003, 26(3): 373-377.
(Zhang H Q, Cai Q S. Time series similarity Querying based on wavelets[J]. J of Computer, 2003, 26(3): 373-377.)

[18] 李海林, 郭崇慧. 基于云模型的时间序列分段聚合近似方法[J]. 控制与决策, 2011, 26(10): 373-377.
(Li H L, Guo C H. Piecewise aggregate approximation method based on cloud model for time series[J]. Control and Decision, 2011, 26(10): 373-377.)

[19] Chonghui Guo, Hailin Li, Donghua Pan. An improved piecewise aggregate approximation based on statistical features for time series mining[C]. Proc of the 4th Int Conf KSEM. Belfast: Springer Berlin Heidelberg Press, 2010: 234-244.

[20] Armita Karachi, Mohammad G Dezfuli, Mostafa S, et al. PLR: A benchmark for probabilistic data stream management systems[C]. The 4th Asian Conf on Intelligent Information and Database Systems. Taiwan: Springer Berlin Heidelberg Press, 2012: 405-415.

[21] Yuelong Zhu, De Wu, Shijin Li. A piecewise linear representation method of time series based on feature points[C]. KES 2007, XVII Italian Workshop on Neural Networks. Vietri sul Mare: Springer Berlin Heidelberg Press, 2007: 12-14.

[22] Wenwei Xue, Qiong Luo, Hejun Wu. Pattern-based event detection in sensor networks[J]. Distributed and Parallel Databases, 2012, 30(1): 27-62.

[23] Guanglei Wu, Shaoping Bai, Kepler J A. Error modelling and experimental validation for a planar 3-PPR parallel manipulator[C]. 2011 Int Conf on Advanced Robotics. Tallinn: Springer Berlin Heidelberg Press, 2011: 259-264.

[24] Rong Tong, Bin Ma, Haizhou Li, et al. A target-oriented phonotactic front-end for spoken language recognition[J]. IEEE Trans on Audio, Speech, and Language Processing, 2009, 17(7): 1335-1347.

(责任编辑：闫 妍)