

# 基于增量动态时间弯曲的时间序列相似性度量方法

李海林 杨丽彬

(华侨大学工商管理学院 泉州 362021)

**摘要** 针对动态时间弯曲方法计算时间过长的问题,提出增量动态时间弯曲来度量较长时间序列之间的相似性。首先利用动态时间弯曲方法对历史时间序列数据进行相似性度量,得到相应的历史最优弯曲路径和路径中各元素的累积距离代价。其次,通过逆向弯曲度量方法完成当前序列数据的相似性度量,结合历史数据信息找到与历史弯曲路径相交且度量时间序列距离为当前最小值的新路径,进而实现增量动态时间弯曲的相似性度量。该方法不仅具有良好的度量质量,还具有较高的时间效率。数值实验表明,对于大部分时间序列数据集,新方法的分类准确率和计算性能要优于经典动态时间弯曲。

**关键词** 时间序列数据挖掘,动态时间弯曲,增量动态时间弯曲,相似性度量

**中图分类号** TP18 **文献标识码** A

## Similarity Measure for Time Series Based on Incremental Dynamic Time Warping

LI Hai-lin YANG Li-bin

(College of Business Administration, Huaqiao University, Quanzhou 362021, China)

**Abstract** To address the issues on the over expensive time cost, an incremental dynamic time warping (IDTW) to measure the similarity between two time series was proposed. First of all, dynamic time warping (DTW) was used to measure similarity of the past time sequences and retrieves the best warping path and the cumulated distance cost of each element in the warping path. Next, after computing the similarity between the two current time series by backward warping method, a new warping path intersects with the past one was obtained and its warping distance was minimal. Finally, the incremental dynamic warping method was realized to measure similarity. The new method not only has the good quality to measure the similarity but also is efficient to compute. The numerical experiments demonstrate that the classification accuracy and computing performance of IDTW are better than DTW.

**Keywords** Time series data mining, Dynamic time warping, Incremental dynamic time warping, Similarity measure

## 1 引言

在时间序列数据挖掘中,相似性度量是一项重要而又基础的任务,大部分数据挖掘算法都需要事先度量数据对象之间的相似性。例如,时间序列数据聚类<sup>[1,2]</sup>、分类<sup>[2,3]</sup>、模式发现<sup>[5,6]</sup>和相似性检索<sup>[7,8]</sup>等。因此,相似性度量质量的好坏对后期数据挖掘结果有直接影响。欧氏距离是计算速度最快的相似性度量方法,通常结合特征表示方法<sup>[4]</sup>来更有效地实现原时间序列的相似性度量。然而,它对异常数据点极其敏感,并且只能度量等长时间序列。动态时间弯曲(Dynamic Time Warping, DTW)<sup>[9-11]</sup>是另外一种应用较为广泛的时间序列度量方法,它通过时间弯曲方法可以对不等长时间序列进行弯曲度量,而且对数据异常点具有鲁棒性,但时间效率较低,不利于大量较长时间序列的相似性度量。

为了使DTW更好地应用于时间序列数据挖掘,大量研究主要集中在提高DTW时间性能方面<sup>[12-15]</sup>。其中,快速时

间弯曲方法(Fast Time Warping, FTW)<sup>[14]</sup>利用最大值和最小值来分别表示每个分段序列的特征,并提出了一种基于极值点且满足下界要求的近似弯曲度量方法,其通常被用于时间序列相似性检索。FastDTW<sup>[15]</sup>是经典动态时间弯曲的近似度量方法,在影响因子较小的情况下,它的时间复杂度线性于时间序列的长度,但具体时间消耗还是取决于影响因子。这些方法虽然在某些具体应用中提高DTW的时间效率,但相似性度量质量取决于参数设定,且无法对时间序列进行增量弯曲度量。针对这些问题,本文提出一种增量态时间弯曲方法(Incremental Dynamic Time Warping, IDTW),它不但能对时间序列进行增量弯曲度量,而且具有良好的度量性能,提高了时间序列数据挖掘算法的效率。

## 2 经典动态时间弯曲

经典动态时间弯曲(DTW)通过调整时间序列不同时间点对应元素之间的关系来获取一条最优弯曲路径,使其能很

到稿日期:2012-06-25 返修日期:2012-11-08 本文受中央高校基本科研业务费(12SKGC-QG03),江西省自然科学基金项目(20122BA BA201044)资助。

李海林(1982-),男,博士,讲师,主要研究方向为数据挖掘与人工智能,E-mail, dr. lihailin@gmail.com; 杨丽彬(1982-),女,硕士,讲师,主要研究方向为数据挖掘和决策支持。

好地度量时间序列之间的关系。假设有两条时间序列  $Q = \{q_1, q_2, \dots, q_{N_Q}\}$  和  $C = \{c_1, c_2, \dots, c_{N_C}\}$ ,  $N_Q$  和  $N_C$  分别表示时间序列  $Q$  和  $C$  的长度。DTW 通过在两条时间序列之间寻找一条最优弯曲路径来得到最小距离度量值  $v = DTW(Q, C)$ 。

两条时间序列之间可以通过弯曲时间戳来配对相同形态的数据点, 这些数据映射关系所形成的路径为弯曲路径。如图 1(a) 和图 1(b) 所示, 通过 DTW 找出时间序列  $Q$  与  $C$  之间的最优弯曲路径  $P$ , 具有相同形态的数据点被相互配对。图 1(b) 中的单元格表示两条时间序列中各数据点的时间对应关系, 横轴和纵轴分别代表时间序列  $Q$  和  $C$  的时间长度, 红色路径为最优弯曲路径。从图 1 中易知, 不同长度的时间序列可以被相互度量, 且具有相同形态的数据点也会被弯曲配对, 因此可以说, DTW 是一种鲁棒性较强的时间序列度量方法。

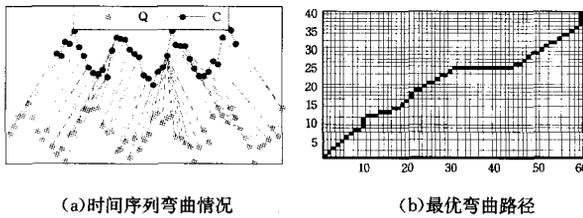


图 1 动态时间弯曲度量

任意满足边界条件、连续性及单调性的路径<sup>[9]</sup>可以表示成  $P = \{p_1, p_2, \dots, p_k\}$ , 其中弯曲元素  $p_k$  表示  $q_i$  与  $c_j$  之间的对应关系, 即  $p_k = (i, j)_k$ 。边界性要求弯曲路径的起点和终点分别是两条时间序列起点和终点的对应关系。连续性和单调性说明了弯曲路径中元素  $p_{k+1}$  只可能出现在  $p_k = (i, j)_k$  的右上角相邻 3 个单元格中。如图 1(b) 所示, 弯曲路径是从起始单元格  $(1, 1)$  一直连续且单调延伸到末尾单元格  $(N_Q, N_C) = (60, 40)$ 。  $d(p_k)$  表示  $q_i$  与  $c_j$  之间的距离, 可以理解为  $q_i$  与  $c_j$  的弯曲代价, 通常取  $d(p_k) = d(i, j) = (q_i - c_j)^2$ 。在这些路径中, 存在一条最优路径使得它的弯曲总代价最小, 即

$$DTW(Q, C) = \min_P \sum_{k=1}^K d(p_k)$$

为了求解上式, 可通过动态规划来构造一个累积代价矩阵  $R$ , 即

$$R(i, j) = d(i, j) + \min \begin{cases} R(i, j-1) \\ R(i-1, j-1) \\ R(i-1, j) \end{cases}$$

式中,  $i = 1, 2, \dots, N_Q, j = 1, 2, \dots, N_C, R(0, 0) = 0, R(i, 0) = R(0, j) = +\infty$ 。  $R(N_Q, N_C)$  就是 DTW 度量时间序列  $Q$  和  $C$  的最小距离值, 即

$$v = R(N_Q, N_C) = DTW(Q, C)$$

### 3 增量动态时间弯曲

由于 DTW 是通过寻找最优弯曲路径来度量时间序列的最小距离, 且需要构造距离代价矩阵  $R$ , 其时间复杂度和空间复杂度为时间序列的长度的平方阶, 不利于大量较长时间序列的相似性度量。为了提高 DTW 的计算速度, 提出一种增量式的动态弯曲度量方法 (Incremental Dynamic Time Warping, IDTW), 同时给出了增量动态时间弯曲方法的时间代价

分析过程。

#### 3.1 算法过程

DTW 度量长度分别为  $N_Q$  和  $N_C$  的两条时间序列相似性的时间复杂度为  $O(N_Q N_C)$ 。若被度量的两条时间序列分别增加  $K_Q$  和  $K_C$  个数据点, 则需要重新运行 DTW 算法程序, 且时间消耗会增长到  $O((N_Q + K_Q)(N_C + K_C))$ 。针对这种情况, 提出增量式的动态时间弯曲方法, 它借助历史时间序列之间的最优弯曲路径和路径元素代价, 结合当前序列数据信息, 增量计算得到近似最优弯曲路径, 进而加快时间序列之间的弯曲距离计算。

对于两条时间序列  $Q$  和  $C$ , 其长度分别为  $N_Q + K_Q$  和  $N_C + K_C$ , 把前  $N_Q(N_C)$  个数据序列点称为时间序列  $Q(C)$  的历史序列  $Q_{N_Q}(C_{N_C})$ , 把后  $K_Q(K_C)$  个数据序列点称为时间序列  $Q(C)$  的当前序列  $Q_{K_Q}(C_{K_C})$ 。

增量动态时间弯曲的基本思想是先利用传统 DTW 来计算两条历史序列之间的最优弯曲路径, 再逆向计算当前序列之间的代价矩阵, 最后从历史序列中的最优弯曲路径最后一个元素开始, 结合当前序列的代价矩阵, 考查该元素是否为历史序列最优弯曲路径和当前序列最优弯曲路径的公共连接元素。增量动态时间弯曲 IDTW 的具体算法步骤如下:

输入: 时间序列  $Q$  和  $C$ , 历史序列长度分别为  $N_Q$  和  $N_C$ , 当前序列长度分别为  $K_Q$  和  $K_C$ 。

输出: 近似最优弯曲路径  $P$  和近似最优弯曲距离  $v$ 。

(1) 利用 DTW 计算两条历史序列的最优弯曲路径  $P^0 = \{p_1^0, p_2^0, \dots, p_{K_0}^0\}$ , 并依次记录该路径中每个元素  $p_k^0 = (i, j)$  在代价矩阵中的代价  $v_k^0 = R(p_k^0) = R(i, j)$ , 即该路径所有元素对应的代价为  $V^0 = \{v_1^0, v_2^0, \dots, v_{K_0}^0\}$ 。根据边界性条件可知,  $p_1^0 = (1, 1)$  和  $p_{K_0}^0 = (N_Q, N_C)$ , 且  $v_{K_0}^0$  为两条历史序列之间的弯曲距离  $DTW(Q_{N_Q}, C_{N_C})$ , 即  $v_{K_0}^0 = DTW(Q_{N_Q}, C_{N_C})$ 。

(2) 逆向计算当前序列之间的代价矩阵  $R_{K_Q \times K_C}$ 。将当前序列  $Q_{K_Q}$  和  $C_{K_C}$  按逆时读取数据并记为  $Q'_{K_Q}$  和  $C'_{K_C}$ , 再利用 DTW 对  $Q'_K$  和  $C'_K$  进行相似性度量并得到当前序列之间的累积代价矩阵  $R_{K_Q \times K_C}$ , 即

$$DTW(Q'_{K_Q}, C'_{K_C}) = R(K_Q, K_C)$$

(3) 从历史最优弯曲路径  $P^0$  中的最后一个元素  $p_{K_0}^0$  开始, 即  $k = K_0$ , 把元素  $p_k^0$  归为当前代价矩阵的新起始元素, 结合当前代价矩阵  $R_{K_Q \times K_C}$ , 利用 DTW 计算新的代价矩阵中的最优弯曲代价  $v^1$  和最优弯曲路径  $P^1$ , 连接历史弯曲路径中的剩余路径元素和弯曲路径  $P^1$  形成全局弯曲路径  $P$ , 并计算全局弯曲距离值  $v$ 。直到历史弯曲路径中相连两个元素形成的全局弯曲距离值相等且到目前为止最小时停止计算, 最终得到的全局弯曲路径  $P$  和全局弯曲距离值  $v$  分别为两条时间序列之间的近似最优弯曲路径和近似最优弯曲距离。

如图 2(a) 所示, 通过算法步骤 (1) 计算历史序列的累积代价矩阵 (粉红色单元格) 和最优弯曲路径 (蓝色单元格); 步骤 (2) 则计算了当前序列的累积代价矩阵 (黄色单元格); 通过步骤 (3), 绿色区域会随着历史弯曲路径元素向前推移而逐步扩大, 直到满足终止条件为止, 且可以得到全局最优弯曲路径 (红色单元格) (如图 2(b) 所示)。对比图 1(b) 和图 2(b) 中的

弯曲路径易知,在该实例中 IDTW 得到的最优弯曲路径与 DTW 产生的最优弯曲路径相同,进而得到相同的度量距离。

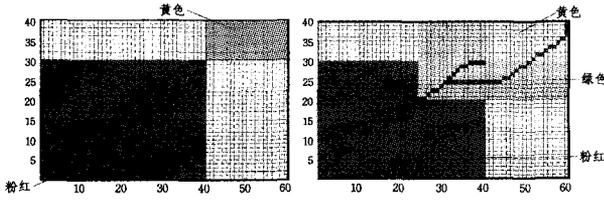


图 2 IDTW 计算最优弯曲路径与历史序列弯曲路径的重叠情况

在步骤(3)中,首先将元素  $p_k^i$  (其中  $1 \leq k \leq K_0$ ) 作为新的当前序列之间代价矩阵的起始元素,通过结合之前的代价矩阵(即图 2(b)中黄色区域),利用 DTW 来计算新反序列  $Q'$  和  $C'$  中的弯曲代价  $v_k = DTW(Q'_{[p_k^i]_Q}, C'_{[p_k^i]_C})$ 。同时,计算并且保留该元素在反序列中的近似最优弯曲路径  $P = \{[p_1^i], [p_2^i], \dots, [p_{k-1}^i]\} \cup P^1$  和对应的近似最优弯曲距离  $v_k = v_{k-1} + v_k$ 。  $P^1$  表示反序列中从初始位置到当前元素位置之间的最优弯曲路径,  $[p_k^i]_Q$  和  $[p_k^i]_C$  分别表示历史最优路径的最后一个元素在反序列  $Q'$  和  $C'$  中的时间位置。  $[p_k^i]$  表示历史最优弯曲路径中第  $i$  个路径元素在反序列中对应的单元格,例如,  $[p_1^i] = ([p_1^i]_Q, [p_1^i]_C) = (N_Q + K_Q, N_C + K_C)$  表示历史最优弯曲路径中第 1 个路径元素在反全序列中对应的单元格为代价矩阵中的最后一个单元格  $(N_Q + K_Q, N_C + K_C)$ 。

如果当前弯曲路径  $P^1$  的最后两个弯曲元素出现在历史弯曲路径  $P^0$  中的相邻位置,它们在反序列中产生的全局弯曲代价相等且到目前为止最小,就可以把这两条相同的弯曲路径视为最优弯曲路径。事实上,统计实验分析表明(见 4.2 节),全局弯曲路径不为最优弯曲路径,而是接近最优弯曲路径,因此在步骤(3)中称全局弯曲路径为近似最优弯曲路径。但是在多数情况下,这两个元素所形成的全局弯曲路径为最优弯曲路径,进而可以得到最小距离度量值。同时,在实际应用中发现,最小弯曲度量值不一定能够完全描述时间序列之间的相似性,有些情况下,近似弯曲度量能在时间序列数据挖掘中得到更好的结果。

### 3.2 时间代价分析

利用 IDTW 计算时间序列相似性的时间复杂度等于  $O(N_Q N_C + K_Q K_C + \delta)$ ,其中  $O(N_Q N_C)$  和  $O(K_Q K_C)$  分别表示利用 DTW 计算历史序列之间相似性和当前序列之间相似性的时间复杂度,如图 2(b)所示,它们分别计算粉红色和黄色单元格的时间。然而,  $O(\delta)$  表示计算历史最优序列弯曲路径中后若干个路径元素在反序列中的距离代价所用的时间,且通常情况下该过程时间消耗较小,如图 2(b)中绿色单元格的计算时间。最坏情况下,它不会超过  $N_Q K_C + N_C K_Q$ ,即  $O(\delta) < N_Q K_C + N_C K_Q$ 。因此,基于增量动态时间弯曲 IDTW 度量时间序列相似性的时间复杂度会小于 DTW,即

$$\begin{aligned} & O(N_Q N_C + K_Q K_C + \delta) \\ & < O(N_Q N_C + N_Q K_C + N_C K_Q + K_Q K_C) \\ & = O((N_Q + N_C)(K_Q + K_C)) \end{aligned}$$

## 4 数值实验

首先需要分析历史序列长度  $N$  和当前序列长度  $K$  对度量时间效率的影响。再次,对增量动态时间弯曲方法(IDTW)做 UCI 时间序列数据集<sup>[16]</sup>分类实验,以进一步确定新方法度量时间序列的性能。

### 4.1 序列长度影响

由于 IDTW 在进行时间序列度量时需要把时间序列划分为历史序列和当前序列,因此,需要分析历史序列和当前序列不同长度对度量时间效率的影响。若时间序列的长度为  $N+K$ ,则把历史序列长度  $N$  在原时间序列长度  $N+K$  的比值称为长度因子  $r$ ,即  $r = \frac{N}{N+K}$ ,其中  $K$  为当前序列的长度。

在数据集 Control Chart 中,根据  $r$  的不同取值,即  $r = [0.1, 0.2, \dots, 0.9]$ ,分别利用 IDTW 和 DTW 对测试集中的每条时间序列和训练集中的每条时间序列进行弯曲度量,最终统计不同  $r$  值情况下的平均时间消耗  $t$ ,其结果如图 3 所示。

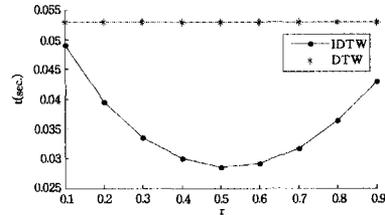


图 3 不同长度因子下的 IDTW 平均时间消耗量

从图 3 易发现,随着  $r$  值的增大,IDTW 的平均时间消耗越来越小;当  $r$  值超过 0.5 后,IDTW 的平均时间消耗逐渐增大;当  $r=0.5$  时,即历史序列与当前序列长度相等时,IDTW 的平均时间消耗会最小。同时,与 DTW 相比,IDTW 利用较少的时间来弯曲度量时间序列之间的相似性。

### 4.2 分类实验

为了说明 IDTW 算法的可行性和优越性,利用 IDTW 和 DTW 对 UCI 时间序列数据集<sup>[16]</sup>(见表 1)进行分类实验,比较两者之间的分类准确率和时间效率。同时,通过对比与 DTW 的弯曲路径或弯曲距离的一致性,统计 IDTW 度量人工时间序列相似性与 DTW 的接近程度,并且结合分类结果来考查 DTW 的最优弯曲度量是否一定优于 IDTW 的近似弯曲度量。

利用这两种方法对表中的所有时间序列数据集做最近邻分类实验,观察它们的分类结果、平均时间消耗量以及弯曲度量时间序列的距离等同率。通过利用 IDTW 和 DTW 分别对测试集中每条测试序列在训练集中查找最相似(距离最小)的序列,并且判断测试序列与最相似序列的标签是否一致来实现基于 IDTW 和 DTW 的最近邻分类。记录两种方法对每组数据集分类所消耗的时间  $T$ ,根据测试集中的序列数目对时间  $T$  做平均化处理,最终得到平均时间消耗量。同时,判断每对时间序列被 IDTW 与 DTW 度量的弯曲距离值是否相等。若 IDTW 与 DTW 度量两条时间序列的距离相等,则说

明两种方法度量这两条时间序列的弯曲路径相同,即说明 IDTW 得到最优弯曲路径;否则说明 IDTW 得到近似最优弯曲路径。统计两者相等情况的数目,并最终计算距离等同率。本次实验中,把每条时间序列对半分成历史序列和当前序列,即  $r=0.5$ ,分类结果如表 1 所列。

表 1 人工时间序列数据集及分类结果和时间效率

数据集信息			分类错误率		时间代价(秒)		距离等同率
序号	名称	长度	DTW	IDTW	DTW	IDTW	
1	Adiac	176	0.396	0.399	168.57	88.43	0.98
2	Beef	470	0.5	0.5	101.39	47.75	0.18
3	CBF	128	0.003	0.008	6.92	3.67	0.13
4	Coffee	286	0.179	0.179	32.64	16.92	0.45
5	ECG200	96	0.23	0.20	12.99	7.31	0.72
6	FISH	463	0.167	0.149	563.86	269.36	0.53
7	FaceAll	131	0.192	0.208	124.64	71.02	0.40
8	FaceFour	350	0.170	0.170	39.20	21.02	0.39
9	Gun_Point	150	0.093	0.087	14.65	8.40	0.23
10	Lighting2	637	0.131	0.131	365.60	175.38	0.02
11	Lighting7	319	0.274	0.260	94.36	51.39	0.05
12	OSULeaf	427	0.409	0.417	503.58	266.53	0.65
13	OliveOil	570	0.133	0.133	142.70	69.63	0.35
14	SwedishLeaf	128	0.210	0.206	105.85	60.834	0.87
15	Trace	275	0	0	100.56	59.79	0.62
16	Two_Patterns	128	0	0	212.27	123.25	0.66
17	Control	60	0.007	0.013	13.91	8.53	0.97
18	Wafer	152	0.02	0.02	201.71	109.22	0.58
19	50Words	270	0.310	0.310	429.12	245.68	0.24
20	Yoga	426	0.164	0.164	752.99	381.35	0.78

从分类结果不难发现,对于大部分数据集来说,与 DTW 相比,IDTW 能获得较好的结果。虽然 IDTW 对小部分数据集的度量结果不如 DTW(如表 1 中的粗体数值所示),但其分类误差很小。从时间性能上看,IDTW 明显要优于 DTW,且几乎节约了 50%的时间。

若距离等同率为 1,说明 IDTW 与 DTW 针对同一时间序列数据集具有相同的度量质量且能找到相同的弯曲路径。若距离等同率为 0,说明这两种方法没有相同的度量结果。若距离等同率在 0 与 1 之间(如表 1 中最后一列所示),说明在同一个数据集中 IDTW 在弯曲度量时间序列时,对不同序列之间的度量可能进行近似弯曲度量,也可能得到与 DTW 相同的最优弯曲度量。同时,结合分类结果分析得知,即使 IDTW 不能完全得到最优弯曲路径和最优度量距离,也可以得到比 DTW 更好的分类结果,说明近似最优弯曲路径和近似最优弯曲距离也能够提高时间序列的相似性度量质量。

通过分类实验说明,IDTW 不仅时间效率较高,而且对于大部分数据集来说,可以得到较好的分类结果;同时,最优弯曲路径所产生的最优度量距离不一定能够很好地度量时间序列之间的相似性,也就是说,近似弯曲路径所产生的近似度量距离可以较好地度量时间序列之间的相似性。

**结束语** 动态时间弯曲(DTW)是一种鲁棒性较强的时间序列度量方法,但其较高的时间复杂度成为度量较长时间序列相似性的瓶颈。针对此问题,本文提出了一种基于增量动态弯曲度量的时间序列相似性度量方法(IDTW)。通过利用历史时间序列之间的最优弯曲路径和路径元素代价,结合

当前序列数据信息来增量计算时间序列的近似最优弯曲距离,同时得到近似最优弯曲路径。通常情况下,IDTW 由于计算弯曲路径的单元格数目小于 DTW 中累积矩阵单元格的数目,可以较快地对时间序列进行弯曲距离度量。数值实验结果表明,该方法不仅具有良好的相似性度量质量,而且时间效率也得到了改善,提高了时间序列数据挖掘中相应算法的性能。

## 参考文献

- [1] 谢福鼎,李迎,孙岩,等.一种基于关键点的时间序列聚类算法[J].计算机科学,2012,39(3):160-162
- [2] Li Hai-lin, Guo Chong-hui, Qiu Wang-ren. Similarity measure based on piecewise linear approximation and derivative dynamic time warping for time series mining[J]. Expert Systems with Applications, 2011, 38(12): 14732-14743
- [3] Xi X, Keogh E, Shelton C, et al. Fast time series classification using numerosity reduction[C]//Proceedings of the 23rd International Conference on Machine Learning. 2006:1033-1040
- [4] 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):1525-1529
- [5] 陈晓华.基于核密度估计的时间序列多模式发现[J].山东大学学报:工学版,2011(5):46-52
- [6] 贾澎涛,何华灿,刘丽,等.时间序列挖掘综述[J].计算机应用研究,2007,24(11):15-29
- [7] 林洵,李志蜀,周勇.时间序列模式相似性研究[J].计算机科学,2011,38(9):245-247
- [8] 李海林,郭崇慧.基于形态特征的时间序列符号聚合近似方法[J].模式识别与人工智能,2011,24(5):665-672
- [9] Keogh E, Pazzani M J. Derivative dynamic time warping[C]//Proceedings of the 1st SIAM International Conference on Data Mining. 2001:1-11
- [10] Keogh E, Ratanamahatana C A. Exact indexing of dynamic time warping[J]. Knowledge and Information Systems, 2005, 7(3): 358-386
- [11] 高学东,崔巍,徐章艳.基于动态时间弯曲的区间值时间序列匹配算法[J].系统工程学报,2007,22(6):664-668
- [12] Itakura F. Minimum prediction residual principle applied to speech recognition[J]. IEEE Transaction on Acoustics, Speech and Signal Process, 1975, 23(1): 52-72
- [13] Sakoe H, Chiba S. Dynamic programming algorithm optimization for spoken word recognition[J]. IEEE Transaction on Acoustics, Speech and Signal Process, 1978, 26(1): 43-49
- [14] Sakurai Y, Yoshikawa M, Faloutsos C. FTW: fast similarity search under the time warping distance[C]//Proceedings of the 2005 Principles of Database Systems. 2005:326-337
- [15] Salvador S, Chan P. FastDTW: toward accurate dynamic time warping in linear time and space[J]. Intelligent Data Analysis, 2007, 11(5): 561-580
- [16] Keogh E, Zhu Q, Hu B, et al. The UCR Time Series Classification/Clustering Homepage[EB/OL]. [http://www.cs.ucr.edu/~eamonn/time\\_series\\_data/](http://www.cs.ucr.edu/~eamonn/time_series_data/), 2012-04-03