ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.20, No.10, October 2009, pp.2867−2884
doi: 10.3724/SP.J.1001.2009.03548

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

基于适应性分段估计的数据流相似性搜索*

吴 枫+, 仲 妍, 吴泉源, 贾 焰, 杨树强

(国防科学技术大学 计算机学院,湖南 长沙 410073)

Similarity Search in Data Stream with Adaptive Segmental Approximations

WU Feng+, ZHONG Yan, WU Quan-Yuan, JIA Yan, YANG Shu-Qiang

(School of Computers, National University of Defense Technology, Changsha 410073, China)

+ Corresponding author: E-mail: wfttyy_2000@163.com

Wu F, Zhong Y, Wu YQ, Jia Y, Yang SQ. Similarity search in data stream with adaptive segmental approximations. *Journal of Software*, 2009,20(10):2867−2884. http://www.jos.org.cn/1000-9825/3548.htm

Abstract: Similarity search has attracted many researchers from various communities (real-time stock quotes, network security, sensor networks). Due to the infinite, continuous, fast and real-time properties of the data from these communities, a method is needed for online similarity search in data stream. This paper first proposes the lower bound function $LB\_seg\_WF_{global}$ for DTW (dynamic time warping) in the presence of global warping constraints and *LB_seg_WF* for DTW without global warping constraints, which are not applied to any index structures. They are segmented DTW techniques, and can be applied to sequences and queries of varying lengths in data stream. Next, several tighter lower bounds are proposed to improve the approximate degree of the $LB\_seg\_WF_{global}$ and *LB_seg_WF*. Finally, to deal with the possible continuously non-effective problem of $LB\_seg\_WF_{global}$ or *LB_seg_WF* in data stream, it is believed that lower-bound LB_WF_{global} (in the presence of global warping constraints) and lower-bound *LB_WF*, upper-bound *UB_WF* (without global warping constraints) can fast estimate DTW and hence reduce a lot of redundant computations by incrementally computing. The theoretical analysis and statistical experiments confirm the validity of the proposed methods.
Key words: similarity search; data stream; time series analysis; dynamic time warping

摘 要: 相似性搜索在股票交易行情、网络安全、传感器网络等众多领域应用广泛.由于这些领域中产生的数据具有无限的、连续的、快速的、实时的特性,所以需要适合数据流上的在线相似性搜索算法.首先,在具有或不具有全局约束条件下,分别提出了没有索引结构的DTW(dynamic time warping)下限函数 $LB\_seg\_WF_{global}$ 和 *LB_seg_WF*,它们是一种分段 DTW 技术,能够处理数据流上的非等长序列间在线相似性匹配问题.然后,为了进一步提高 $LB\_seg\_WF_{global}$ 和 *LB_seg_WF* 的近似程度,提出了一系列的改进方法.最后,针对流上使用 $LB\_seg\_WF_{global}$ 或 *LB_seg_WF* 可能会出现连续失效的情况,分别提出了 DTW 的下限函数 LB_WF_{global}(具有全局约束条件)和上限函数 *UB_WF*、下限函数 *LB_WF*(不具有全局约束条件).通过增量方式快速估计 DTW,极大地减少了估计 DTW 的冗余计算量.通过理论分析和统计实验,验证了该方法的有效性.
关键词: 相似性搜索;数据流;时间序列分析;动态时间扭曲

* Supported by the National High-Tech Research and Development Plan of China under Grant Nos.2006AA01Z451, 2007AA01Z474 (国家高技术研究发展计划(863))

Received 2008-11-28; Accepted 2008-12-30

中图法分类号: TP311 文献标识码: A

相似性搜索在股票交易行情、网络安全、传感器网络等众多领域有着越来越广泛的应用.例如:发现具有类似销售模式的产品或具有相似价格趋势的股票,等等.由于这些领域中产生的数据具有无限、连续、快速、实时的特性,所以,迫切需要一种数据流上在线相似性搜索算法.数据流上在线相似性搜索[1−3]的研究目标是:给定查询序列 $Q=\langle q_1,q_2,\ldots,q_n\rangle$,数据流 $S=\langle s_1,s_2,\ldots\rangle$,当新的数据 s_{i+m-1} 到达时,对于最近 m 个数据形成的子序列 $S(i,i+m-1)=\langle s_i,s_{i+1},\ldots,s_{i+m-1}\rangle$,需要实时、快速地度量 Q 和 $S(i,i+m-1)$之间的相似性程度,从而找出所有满足搜索条件的流上子序列 $D(Q,S(i,i+m-1))\le\varepsilon$.

传统的距离度量函数是欧氏距离.它计算简单且具有良好的性质(如满足三角不等式),但是无法处理时间扭曲情况[4]:如果原本相似的两个时间序列发生了不同步的时间扭曲,那么在利用欧氏距离度量时,很可能被认定为不相似.另外,欧氏距离只能度量等长的时间序列.针对欧氏距离的缺陷,提出了动态时间扭曲(DTW)距离.然而,由于 DTW 的计算复杂性过高($O(n\cdot m)$),大部分研究工作采用 DTW 下限函数 $LB(Q,S)\le DTW(Q,S)$来近似 DTW[4].通常,下限函数 LB 需要满足[4]:计算量小;近似程度尽可能地高.

Yi 和 Faloutsos[5]提出了下限函数 *Lb_Yi*.给定两个序列 Q,S,令两个序列中的最大值分别为 $\max_Q,\max_S$,最小值分别为 $\min_Q,\min_S$;定义 $\min\_\max=\min(\max_Q,\max_S)$,$\max\_\min=\max(\min_Q,\min_S)$.不失一般性,假设 $\max_Q>\max_S$,显然,任意 $q>\min\_\max$,$q\in Q$,q 至少为 DTW 距离贡献 $d(q,\min\_\max)$;同样地,假设 $\min_Q>\min_S$,显然,任意 $s<\max\_\min$,$s\in S$,s 至少为 DTW 距离贡献 $d(s,\max\_\min)$.基于这种思想,下限函数 *Lb_Yi* 定义为

$$Lb\_Yi(Q,S)=\begin{cases}\max\left(\sum_i|q_i-\max_S|,\sum_j|s_j-\min_Q|\right), & \min_Q>\max_S\\ \sum_i\phi(q_i-\max_Q)+\sum_j\phi(\min_Q-s_j), & \min_Q\le\max_S,\min_Q\ge\min_S\\ \sum_i\phi(q_i-\max_S)+\sum_i\phi(\min_S-q_i), & \min_Q<\min_S\end{cases}.$$

Keogh[6−8]提出了具有全局约束条件下的下限函数 *Lb_Keogh*.主要思想是:首先建立查询序列 Q 的限制区域(记为 $E(Q)$),然后计算 S 中每个元素 s 到 $E(Q)$的最小垂直距离 d_s(如果 s 包含在 $E(Q)$之中,则 d_s 为 0),最后累加这些距离,得到 $Lb\_Keogh(E(Q),S)$:

$$Lb\_Keogh(E(Q),S)=\sqrt[p]{\sum_{i=1}^{N}\begin{cases}|S[i]-U[i]|^p, & \text{if } S[i]>U[i]\\ |S[i]-L[i]|^p, & \text{if } S[i]<L[i]\\ 0, & \text{otherwise}\end{cases}},$$

其中,U,L 分别为 $E(Q)$的最大和最小边界,且 $U_i=\max_{-\rho\le r\le\rho}(Q[i+r])$, $L_i=\min_{-\rho\le r\le\rho}(L[i+r])$.由于 *Lb_Keogh* 缺少有效的索引机制,所以,Keogh 又提出了另一个下限函数 *Lb_PAA*.其核心思想是:利用 PAA 将原始长为 N 的序列分成 f($f<<N$)个等长分段序列,每个分段保存其中子序列的平均值.令查询序列 Q 的分段序列为 $P(Q)=[\overline{Q[1]},\ldots,\overline{Q[f]}]$,$E(Q)$的分段序列为 $P(E(Q))$,其中,第 i 个元素记为 $(\overline{L[i]},\overline{U[i]})$.定义 $Lb_PAA(P(E(Q),P(S))$如下:

$$Lb\_PAA(P(E(Q),P(S))=\sqrt[p]{\sum_{i=1}^{f}\frac{N}{f}\begin{cases}|\overline{S[i]}-\overline{U[i]}|^p, & \text{if } \overline{S[i]}>\overline{U[i]}\\ |\overline{S[i]}-\overline{L[i]}|^p, & \text{if } \overline{S[i]}<\overline{L[i]}\\ 0, & \text{otherwise}\end{cases}}.$$

Zhu 和 Shasha[9]对 *Lb_PAA* 进行了优化,并在具有全局约束条件下,为 PAA 分段序列建立了索引结构.

综上,这些下限函数只在全局约束条件下并且序列等长时有效[4].另外,它们都具有一定的索引结构,因此难以适用于数据流上的在线实时搜索.针对这些情况,本文提出了具有或不具有全局约束条件下,没有索引结构的非等长序列的数据流上在线相似性搜索算法(online similarity search on data stream,简称 OSSDS).首先,在具有或不具有全局约束条件下,分别提出了没有索引结构的 DTW 下限函数 $LB\_seg\_WF_{global}$ 和 *LB_seg_WF*,它们是一种分段 DTW 技术,能够处理数据流上的非等长序列间在线相似性匹配问题.然后,为了进一步提高

$LB\_seg\_WF_{global}$ 和 LB_seg_WF 的近似程度,提出了一系列的改进方法.最后,针对流上使用 $LB\_seg\_WF_{global}$ 或 LB_seg_WF,可能会出现连续失效的情况,分别提出了 DTW 的下限函数 LB_WF_{global}(具有全局约束条件)和上限函数 UB_WF、下限函数 LB_WF(不具有全局约束条件),通过增量的方式快速估计 DTW,大幅度地减少了估计 DTW 的冗余计算量.

本文第 1 节介绍相关的预备知识,包括动态时间扭曲方法和问题定义.第 2 节进行本文算法的推导和证明:第 2.1 节提出一种没有全局约束的分段 DTW 技术——DTW 下限函数 LB_seg_WF,并进一步提高 LB_seg_WF 的近似程度;第 2.2 节提出具有全局约束的分段 DTW 技术——DTW 下限函数 $LB\_seg\_WF_{global}$;针对流上使用 $UB\_WF(LB\_seg\_WF_{global})$,可能会出现连续失效的情况,第 2.3 节提出没有全局约束的 DTW 的上限函数 UB_WF 与下限函数 LB_WF 和具有全局约束的 DTW 的下限函数 LB_WF_{global},并结合第 2.1 节、第 2.2 节的工作,提出数据流上在线相似性搜索算法(OSSDS).第 3 节给出统计实验结果和分析.最后,给出总结.

1 预备知识

1.1 动态时间扭曲方法

首先,介绍 DTW 原理.给定两个时间序列 X,Y,DTW 的计算公式[10]如下:

$$DTW(<>,<>)=0,$$

$$DTW(X,<>)=DTW(<>,Y)=\infty,$$

$$DTW(X,Y)=\sqrt[p]{|X[1]-Y[1]|^2+\min\begin{cases}DTW(Rest(X),Rest(Y))\\DTW(Rest(X),Y)\\DTW(X,Rest(Y))\end{cases}}.$$

其中,$Rest(X),Rest(Y)$代表序列 X,Y 中去掉第 1 个元素后的剩余序列;扭曲路径 $W=\langle w_1,w_2,\ldots,w_k,\ldots,w_L\rangle$,$\max(n,m)\le L\le n+m-1$ 可由矩阵中的元素序列表示(图 1 黑色元素),代表了两个序列的最优对整.

通常,DTW 需要满足如下约束[3]:端点约束:扭曲路径开始于(1,1),结束于(n,m).单调性和连续性约束:给定 $w_k=(i_k,j_k)$和 $w_{k+1}=(i_{k+1},j_{k+1})$,必须满足 $0\le i_{k+1}-i_k\le 1$ 和 $0\le j_{k+1}-j_k\le 1$.

另外,为了减少计算量,同时避免病态扭曲,引入了全局约束[4](global constraints),如 Sakoe-Chiba 和 Itakura Parallelogram 约束.在 Sakoe-Chiba 约束下,有$|i-j|\le\rho$,其中,ρ为扭曲宽度(如图 1 所示).

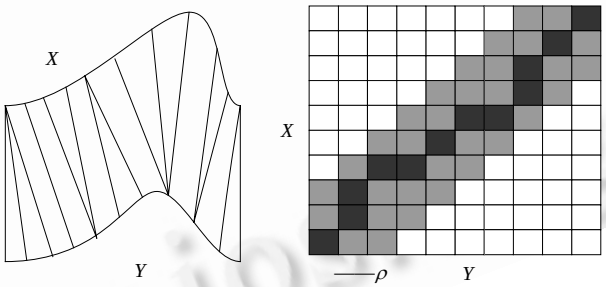


Fig.1 Warping matrix, warping path and alignment

图 1 扭曲矩阵,扭曲路径和对整

可用动态规划方法寻找最优扭曲路径,如下面的迭代公式所示[3]:

$$\gamma(i,j)=d(i,j)+\min\begin{cases}\gamma(i-1,j)\\\gamma(i,j-1)\\\gamma(i-1,j-1)\end{cases},\quad 1\le i\le n,1\le j\le m.$$

$$\gamma(0,0)=0,\ \gamma(i,0)=0,\ \gamma(0,j)=\infty,$$

所有的$\gamma(i,j)$构成了累积距离矩阵 CDM,其中,$\gamma(i,j)$代表了(1,1)和(i,j)之间的最小累积距离.图 2 中表示了 X,Y

之间的累积距离矩阵.类似地,也可以反向寻找最优扭曲路径.令反向累积距离矩阵 $CDM_{反向}$起始点为(n,m),终止点为(1,1),图 3 中表示了 X,Y 之间的后向累积距离矩阵.该方法基于如下迭代公式[3]:

$$\gamma(i,j)=d(i,j)+\min\begin{cases}\gamma(i+1,j)\\ \gamma(i,j+1)\\ \gamma(i+1,j+1)\end{cases},\qquad 1\le i\le n,1\le j\le m.$$

$$\gamma(n+1,m+1)=0,\ \gamma(i,m+1)=\gamma(n+1,j)=\infty,$$

0.5	2.5	2.5	2.5	2.75	2.5
–1	2.25	4.25	2.5	4.25	0.25
1	1.25	0.25	0.5	0.25	4.25
0.5	0.25	0.25	0.25	0.5	2.75
0	0	1	1.25	2.25	3.25
X/*Y*	0	1	0.5	1	–1

Fig.2 Forward cumulative distance matrix for sequence X and Y

图 2 序列 X 和 Y 的前向累积距离矩阵

0.5	3	2.75	2.5	2.5	2.25
–1	3.75	6.5	4.75	6.25	2.25
1	3.5	2.5	2.5	2.25	6.25
0.5	2.75	2.5	2.25	2.5	8.5
0	2.5	3.25	2.5	3.5	9.5
X/*Y*	0	1	0.5	1	–1

Fig.3 Backward cumulative distance matrix for sequence X and Y

图 3 序列 X 和 Y 的后向累积距离矩阵

1.2 问题定义

下面,给出本文研究的数据流上在线相似性搜索问题的相关定义和约定.

定义 1[11]:1) 区间搜索:如果 $S(i,i+m-1)\in\Omega_{Range}$,则有 $DTW(Q,S(i,i+m-1))\le\varepsilon$;

2) 优化区间搜索:如果 $S(i,i+m-1)\in\Omega_{ORange}$,则有:

① $DTW(Q,S(i,i+m-1))\le\varepsilon$;

② 在Ω_{ORange}中,不存在与 $S(i,i+m-1)$相重合的子序列.

因为数据流上存在很多相重合的子序列,同时满足区间搜索条件,而且还反映出相同或相似的信息.所以,为了避免增加冗余计算开销,本文仅关注优化区间搜索问题.

为便于阅读本文,定义如下标记(见表 1).另外约定,在累积距离矩阵 CDM 中,将矩阵的行标自底而上地取为 1 到 n,将列标自左而右地与当前流上子序列在流中的位置保持一致,即 $CDM(Q,S(i,k+m-1))$中的列标起始于 i,终止于 $k+m-1$,行标为 1 到 n;类似地,$CDM(Q,S(k,k+m-1))$中的列标起始于 k,终止于 $k+m-1$,行标为 1 到 n.

Table 1 Symbols and definitions

表 1 标记和定义

Symbol	Definition
S,s_i	Stream data, ith element of S
Q	Query sequence $Q=\langle q_1,q_2,\ldots,q_n\rangle$
Q^{seg}	$Q^{seg}=\langle Q_1^{seg},Q_2^{seg},\ldots,Q_N^{seg}\rangle$
$S(i,i+m-1)$	Subsequence of S which from time i to $i+m-1$ of length m
$S^{seg}(i,i+m-1)$	$S^{seg}(i,i+m-1)=\langle S_1^{seg}(i,i+m-1),\ldots,S_M^{seg}(i,i+m-1)\rangle$
W_1	The optimal warping path between $(1,i)$ and (n,k) in cumulative distance matrix
W_2	The optimal warping path between $(1,k)$ and $(n,k+m-1)$
W_3	The optimal warping path between $(1,i)$ and $(n,k+m-1)$
W_4	The optimal warping path between $(1,i)$ and $(n,i+m-1)$
$d(W_i)$	The correspond distance for W_i
$d(i,t)$	Distance of (i,t) element of time matrix: $d(i,t)=d(q_i,s_t)=(q_i-s_t)^2$
d_1	$\sum_{t=i}^{k-1}d(1,t)$
d_2	$\sum_{t=1}^{n}d(t,k)$
D_0	$DTW^2(Q,S(i,k+m-1))$
D_1	$DTW^2(Q,S(k,k+m-1))$
D_2	$DTW^2(Q,S(i,i+m-1))$
$CDM(Q,S)$	The cumulative distance matrix for Q,S
$CDM_{global}(Q,S)$	The cumulative distance matrix for Q,S under global constraints
$CDM(Q^{seg},S^{seg})$	The cumulative distance matrix for Q^{seg},S^{seg}

2 算法推导

2.1 分段DTW

给定查询序列 Q,流上子序列 $S(i,i+m-1)$,对应的 APCA(adaptive piecewise constant approximation)分段[7,8]子序列分别记为 Q^{seg} 和 $S^{seg}(i,i+m-1)$分段长度记为 $M=|S^{seg}(i,i+m-1)|$,$N=|Q^{seg}|$.令

$$Q^{seg}=\langle Q_1^{seg},Q_2^{seg},...,Q_N^{seg}\rangle$$
$$Q_j^{seg}=\langle Q_{j,1}^{seg},Q_{j,2}^{seg},...,Q_{j,n_j}^{seg}\rangle,\ 1\le j\le N,$$

其中,n_j 代表分段 Q_j^{seg} 中元素的个数;

$$S^{seg}(i,i+m-1)=\langle S_1^{seg}(i,i+m-1),...,S_M^{seg}(i,i+m-1)\rangle$$
$$S_j^{seg}(i,i+m-1)=\langle S_{j,1}^{seg}(i,i+m-1),...,S_{j,m_j}^{seg}(i,i+m-1)\rangle,\ 1\le j\le M,$$

其中,m_j 代表分段 $S_j^{seg}(i,i+m-1)$ 中元素的个数.根据两个序列的分段估计,可以构造 $DTW(Q,S(i,i+m-1))$的一个下限函数 $LB\_seg\_WF(Q,S(i,i+m-1))$.

首先构造一个 $N\times M$ 矩阵,其中,元素(j,h)的值代表了 Q_j^{seg} 和 $S_h^{seg}(i,i+m-1)$ 之间的分段距离 $d^{seg}(j,h)$:

$$d^{seg}(j,h)=\min\{d(Q_{j,k}^{seg},S_{h,l}^{seg}(i,i+m-1))\,|\,1\le k\le n_j,1\le l\le m_h\}.$$

利用该距离,可以计算 $LB\_seg\_WF(Q,S(i,i+m-1))=DTW(Q^{seg},S^{seg}(i,i+m-1))$.

下面,证明 $LB\_seg\_WF(Q,S(i,i+m-1))\le DTW(Q,S(i,i+m-1))$.

定理 1. 给定长度为 n 的查询序列 Q、流上子序列 $S(i,i+m-1)$以及它们的 APCA 分段子序列 Q^{seg} 和 $S^{seg}(i,i+m-1)$,则有 $LB\_seg\_WF(Q,S(i,i+m-1))\le DTW(Q,S(i,i+m-1))$.

证明:令 $CDM(Q,S(i,i+m-1))$为 $Q,S(i,i+m-1)$的累积距离矩阵,$W=w_1,w_2,\ldots,w_K$,$\max(m,n)\le K\le m+n-1$ 为 $CDM(Q,S(i,i+m-1))$的最优扭曲路径;令 $CDM(Q^{seg},S^{seg}(i,i+m-1))$为 $Q^{seg},S^{seg}(i,i+m-1)$的累积距离矩阵,$P=p_1,p_2,\ldots,p_{K'}$,$\max(M,N)\le K'\le M+N-1$ 为 W 在 $CDM(Q^{seg},S^{seg}(i,i+m-1))$中对应的扭曲路径.由于连续性约束,任意的 $p_k\in P$ 中所包含的 $w_j\in W$ 将形成 W 的一条子路径.所以,可以定义每个 p_k 为 $w_{s_k},...,w_{e_k}$,其中,w_{s_k} 为包含在 p_k 中的第 1 个元素,w_{e_k} 为包含在 p_k 中的最后一个元素.最后,令 R 为 $CDM(Q^{seg},S^{seg}(i,i+m-1))$中的最优扭曲路径.期望证明 $d(R)\le d(W)$.

首先,可以根据分段距离 $d^{seg}(j,h)$得知 $d(p_k)\le\sum_{j=s_k}^{e_k}d(w_j)$,进而有

$$\sum_{k=1}^{K'}d(p_k)\le\sum_{k=1}^{K'}\sum_{j=s_k}^{e_k}d(w_j)=\sum_{j=1}^{K}d(w_j).$$

另外,因为 R 为 $CDM(Q^{seg},S^{seg}(i,i+m-1))$中的最优扭曲路径,所以 $d(R)\le d(P)$.结合上式,得到 $d(R)\le d(W)$.显然,$LB_seg_WF(Q,S(i,i+m-1))\le DTW(Q,S(i,i+m-1))$结论得证. □

注意到,$CDM(Q,S(i,i+m-1))$中的元素$(1,i)$和$(n,i+m-1)$必定包含在其最优扭曲路径中;沿用定理 1 的记号,令 $p_1\in P$, $p_1=w_{s_1},...,w_{e_1}$, $s_1=1$,由于连续性约束,$\min(n_1,m_1)\le e_1$,故 p_1 中至少包含 W 中的 $\min(n_1,m_1)$个元素.同理,$p_{K'}$ 中至少包含 W 中的 $\min(n_N,m_M)$个元素.

基于上述观察,修改分段距离 $d^{seg}(j,h)$的定义如下:

$$d_{modify}^{seg}(j,h)=\begin{cases}d_1^{seg}(j,h), & \text{if } (j=1 \text{ and } h=1) \text{ or } (j=N \text{ and } h=M)\\ d^{seg}(j,h), & \text{otherwise}\end{cases}.$$

令 $d_{0,\min}=\min\{d(Q_{1,k}^{seg},S_{1,l}^{seg}(i,i+m-1))\,|\,1\le k\le n_1,1\le l\le m_1;k=1\text{时},l\ne1;l=1\text{时},k\ne1\}$,

$d_{1,\min}=\min\{d(Q_{N,k}^{seg},S_{M,l}^{seg}(i,i+m-1))\,|\,1\le k\le n_N,1\le l\le m_M;k=n_N\text{时},l\ne m_M;l=m_M\text{时},k\ne n_N\}$.

于是,修改得到的 $d_1^{seg}(j,h)$ 如下:

$$d_1^{seg}(j,h)=\begin{cases}d_{0,\min}\cdot(\min(n_1,m_1)-1)+d(q_1,s_i), & \text{if } j=1 \text{ and } h=1\\ d_{1,\min}\cdot(\min(n_N,m_N)-1)+d(q_n,s_{i+m-1}), & \text{if } j=N \text{ and } h=M\end{cases}.$$

定义 2. 给定长度为 n 的查询序列 Q、流上子序列 $S(i,i+m-1)$以及它们的 APCA 分段子序列 Q^{seg} 和 $S^{seg}(i,i+m-1)$.令 $WDM(Q,S(i,i+m-1))$为 $Q,S(i,i+m-1)$的扭曲距离矩阵,其中,元素(j,h)取值为 $d(Q_j,S_h(i,i+m-1))$;令 $CDM(Q,S(i,i+m-1))$为 $Q,S(i,i+m-1)$的累积距离矩阵,$CDM(Q^{seg},S^{seg}(i,i+m-1))$为 $Q^{seg},S^{seg}(i,i+m-1)$的累积距离矩阵.假定 P 为 $CDM(Q,S(i,i+m-1))$(或 $CDM(Q^{seg},S^{seg}(i,i+m-1))$)中的一条扭曲(子)路径,则有 $d(P)=\sum d_j(P)$,其中,$d_j(P)\in WDM(Q,S(i,i+m-1))$.定义 $d_j(P)$为路径 P 中的第 j 个计算元素(ce).

特别地,令 $CDM(Q^{seg},S^{seg}(i,i+m-1))$的最优扭曲路径为 $R=r_1,r_2,\ldots,r_{K''}$,$\max(M,N)\le K''\le M+N-1$,其中,$r_k=(j,h)$,$j$ 代表 Q^{seg} 中的第 j 个分段,h 代表 $S^{seg}(i,i+m-1)$中的第 h 个分段.定义 $CE_{j,h}^{SCDM}$ 为(j,h)中的计算元素的集合,CE^{SCDM} 为所有 $CE_{j,h}^{SCDM}$ 的集合.令 $d_{\min}=\min\{d(Q_{j,k}^{seg},S_{h,l}^{seg}(i,i+m-1))\mid 1\le k\le n_j,1\le l\le m_h,1\le j\le N,1\le h\le M\}$,代表所有分段内元素间距的最小值.由于 $CDM(Q,S(i,i+m-1))$的最优扭曲路径中包含的计算元素的个数最少为 $C_1=\max(m,n)$,而 $CDM(Q^{seg},S^{seg}(i,i+m-1))$的最优扭曲路径中包含的计算元素的个数至多为 $C_2=M+N-3+\min(n_1,m_1)+\min(n_N,m_M)$(由 $d_1^{seg}(j,h)$ 的定义可知,在分块 $(Q_1^{seg},S_1^{seg}(i,i+m-1))$ 和 $(Q_N^{seg},S_M^{seg}(i,i+m-1))$ 中,实际的计算元素的个数分别为 $\min(n_1,m_1)$,$\min(n_N,m_M)$).而理想情况下,根据连续性约束,$CDM(Q^{seg},S^{seg}(i,i+m-1))$的最优扭曲路径中计算元素的个数至少应为 C_1.基于这种观察,得到如下结论:

定理 2. 给定长度为 n 的查询序列 Q、流上子序列 $S(i,i+m-1)$以及它们的 APCA 分段子序列 Q^{seg} 和 $S^{seg}(i,i+m-1)$,则有 $LB\_seg\_WF^2(Q,S(i,i+m-1))+d_{\min}\times(C_1-C_2)\le DTW^2(Q,S(i,i+m-1))$.

证明:令 $W=w_1,w_2,\ldots,w_K$,$\max(m,n)\le K\le m+n-1$ 为 $CDM(Q,S(i,i+m-1))$的最优扭曲路径;令 $P=p_1,p_2,\ldots,p_{K'}$,$\max(M,N)\le K'\le M+N-1$ 为 W 在 $CDM(Q^{seg},S^{seg}(i,i+m-1))$中对应的扭曲路径.定义每个 p_k 为 $w_{s_k},\ldots,w_{e_k}$,其中,w_{s_k} 为包含在 p_k 中的第 1 个元素,w_{e_k} 为包含在 p_k 中的最后一个元素.令 p_k 中的计算元素的个数为 c_k,则显然有 $e_k-s_k+1\ge c_k$.由 $d_{modify}^{seg}(j,h)$ 的定义可知,$d(w_{k_1})+d(w_{k_2})+\ldots+d(w_{k_{c_k}})\ge d(p_k)(k_1,\ldots,k_{c_k}\in(s_k,\ldots,e_k))$,而任意的 $d(W_j)$ $(1\le j\le K)$均大于或等于 $d_{\min}$,所以 $\sum_{j=s_k}^{e_k}d(w_j)\ge d(p_k)+(e_k-s_k+1-c_k)\times d_{\min}$,进而有

$$\sum_{k=1}^{K'}\sum_{j=s_k}^{e_k}d(w_j)\ge\sum_{k=1}^{K'}(d(p_k)+(e_k-s_k+1-c_k)\times d_{\min}).$$

由 C_1,C_2 定义可知 $\sum_{k=1}^{K'}(e_k-s_k+1-c_k)\ge C_1-C_2$,所以

$$DTW^2(Q,S(i,i+m-1))=\sum_{k=1}^{K'}\sum_{j=s_k}^{e_k}d(w_j)\ge\sum_{k=1}^{K'}d(p_k)+(C_1-C_2)\times d_{\min}.$$

最后,令 R 为 $CDM(Q^{seg},S^{seg}(i,i+m-1))$中的最优扭曲路径,显然 $\sum_{k=1}^{K'}d(p_k)\ge d(R)$,所以,

$$DTW^2(Q,S(i,i+m-1))\ge d(R)+d_{\min}\times(C_1-C_2)=LB\_seg\_WF^2(Q,S(i,i+m-1))+d_{\min}\times(C_1-C_2).$$ □

于是,得到新的下限函数 $LB\_seg\_WF1^2(Q,S(i,i+m-1))=LB\_seg\_WF^2(Q,S(i,i+m-1))+d_{\min}\times(C_1-C_2)$.

图 4 中,图 4(a)代表 $CDM(Q,S(i,i+m-1))$;图 4(b)代表利用分段距离 $d^{seg}(j,h)$求出的 $CDM(Q^{seg},S^{seg}(i,i+m-1))$,其中,$LB\_seg\_WF^2(Q,S(i,i+m-1))=6$;图 4(c)代表利用分段距离 $d_{modify}^{seg}(j,h)$ 求出的 $CDM(Q^{seg},S^{seg}(i,i+m-1))$,其中,$LB\_seg\_WF^2(Q,S(i,i+m-1))=38$.显然,$d_{modify}^{seg}(j,h)$ 明显优于 $d^{seg}(j,h)$.此时,$LB\_seg\_WF1^2(Q,S(i,i+m-1))=38+1\times(-6+3+2-3+2+2)=38$.可以看出,$d_{\min}\times(C_1-C_2)$部分对 $LB_seg_WF1^2(Q,S(i,i+m-1))$的贡献不是很明显.显然,大部分时候,要么$(C_1-C_2)=0$,要么 $d_{\min}$ 非常小,所以,还可以更进一步地提高 LB_seg_WF1 的近似程度.首先我们来观察下面一个例子:

图 4 中,图 4(a1)中最优扭曲路径的计算元素为$(1,9)_{1,1},(1,9)_{1,2},(1)_{1,3},(1,1)_{2,3}$,其中,$(1,9)_{1,1}$ 代表包含在分块 $(Q_1^{seg},S_1^{seg}(i,i+m-1))$ 中的计算元素,其他类似;图 4(c1)中的最优扭曲路径的计算元素为$(d(5,6),4)_{1,1},(1)_{1,2},(1,d(4,3))_{2,3}$,其中,$(d(5,6),4)_{1,1}$ 代表 $(Q_1^{seg},S_1^{seg}(i,i+m-1))$ 中的计算元素.显然,两者最大的差距在于$(1,9)_{1,2}$ 和$(1)_{1,2}$.观察发现,图 4(c1)的最优扭曲路径中实际包含的计算元素的个数为 5,显然比 $C_1=\max(4,6)$小 1.由前面的分析,在理想情况下,根据连续性约束,$CDM(Q^{seg},S^{seg}(i,i+m-1))$的最优扭曲路径中的计算元素的个数至少应该为 C_1.所以,再至少合理增加图 4(c1)中的一个计算元素是可行的.

5	46	25	25	22	26	42
1	21	21	21	43	70	111
2	20	17	18	34	47	80
4	16	17	18	22	31	56
Q/*S*	0	3	3	6	7	9

(a)

4	27	7	15	40	41	23
2	26	6	30	79	25	22
1	17	5	41	75	21	25
6	1	10	11	20	56	65
Q/*S*	5	3	7	9	0	3

(a1)

1,5	2	2	6
4,2	1	2	11
Q/*S*	0,3	3,6	7,9

(b)

2,4	2	10	3
6,1	1	2	3
Q/*S*	5,3	7,9	0,3

(b1)

1,5	18	18	4+18+*d*(5,9)=38
4,2	1+*d*(4,0)	18	27
Q/*S*	0,3	3,6	7,9

(c)

2,4	6	14	6+1+*d*(4,3)=8
6,1	4+*d*(6,5)	6	7
Q/*S*	5,3	7,9	0,3

(c1)

Fig.4 Cumulative distance matrix and segmental cumulative distance matrix

图 4 累积距离矩阵和分段累积距离

在讨论如何确定这个增加的计算元素之前,我们先提出若干定义.

定义 2(行(列)约束度). 在 $CDM(Q^{seg},S^{seg}(i,i+m-1))$的最优扭曲路径中,行 $j,1\le j\le N$(列 $h,1\le h\le M$)的计算元素个数之和应该大于或等于 $N_j^{Row}=n_j(N_h^{Column}=m_h)$,其中,$n_j$代表分段 Q_j^{seg} 中元素的个数(m_h代表分段 $S_h^{seg}(i,i+m-1)$中元素的个数).此时,称 $N_j^{Row}(N_h^{Column})$ 为行(列)约束度.

定理 3. 在 $CDM(Q^{seg},S^{seg}(i,i+m-1))$的最优扭曲路径中,行、列约束度不一定同时得到满足.

证明:观察图 5,由所有网格标记的路径代表 $CDM(Q^{seg},S^{seg}(i,i+m-1))$的最优扭曲路径.显然,第 2 列不满足列约束度(1<3),而行约束度全部满足. □

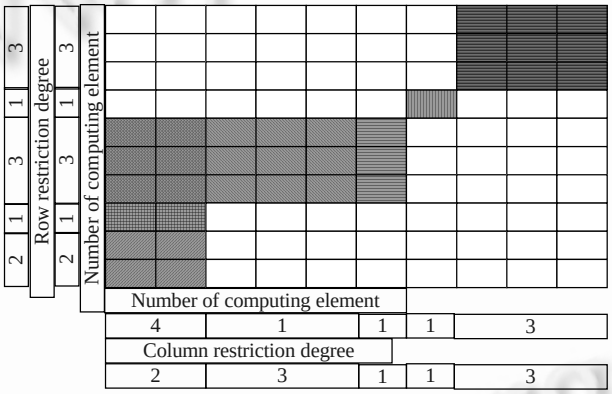


Fig.5 Number of computation elements and row, column constrain degree in the optimal warping path in segmental cumulative distance matrix

图 5 分段累积距离矩阵中最优扭曲路径的计算元素个数和行、列约束度

定义 4. 为确保同时满足行、列约束度,需要在 $CDM(Q^{seg},S^{seg}(i,i+m-1))$的某些特定区域内填入额外的计算元素,这些特定区域构成的集合称为填充区域集Σ^{fill};令违反行约束度的行的集合为 $J=(J_1,...,J_{n_{row}})$,违反列约束度的列的集合为 $H=(H_1,...,H_{n_{column}})$,则 $\Sigma^{fill}=\{\Sigma_{J_j,H_h}^{fill}=(J_j,H_h)\mid J_j\in J,H_h\in H\}$.其中,$\Sigma_{J_j,H_h}^{fill}$ 称为填充区域;如果 $H=\phi(J=\phi)$,则$(J_j,\phi)((\phi,H_h))$代表第 J_j 行(第 H_h 列).

定义 5. 需要在填充区域集Σ^{fill} 内填入的计算元素的个数 n^{fill} 称为填充个数.令根据行约束度确定的填入元素的个数记为 n_1^{fill},根据列约束度确定的填入元素的个数记为 n_2^{fill},则 $n^{fill}=\max(n_1^{fill},n_2^{fill})$.填充区域 Σ_{J_j,H_h}^{fill} 中需要填入的计算元素的个数记为 n_{J_j,H_h}^{fill},这些填入个数构成的集合记为 $N^{fill}=\{n_{J_j,H_h}^{fill}\}$.显然应该有 $\sum n_{J_j,H_h}^{fill}=n^{fill}$.

定义 6. 若 $n_1^{fill}>n_2^{fill}(n_1^{fill}\le n_2^{fill})$,则填充区域 Σ_{J_j,H_h}^{fill} 所在的行 J_j(列 H_h)组成的取值区域称为该填充区域的填

充元素取值区域:

$$\Omega_{J_j,H_h}^{region}=\begin{cases}\{d(Q_{j',k}^{seg},S_{h',l}^{seg}(i,i+m-1))\mid 1\le k\le n_{j'},1\le l\le m_{h'};j'=J_j,1\le h'\le M\}, & \text{If } n_1^{fill}>n_2^{fill}\\ \{d(Q_{j',k}^{seg},S_{h',l}^{seg}(i,i+m-1))\mid 1\le k\le n_{j'},1\le l\le m_{h'};h'=H_h,1\le j'\le N\}, & \text{If } n_1^{fill}\le n_2^{fill}\end{cases}.$$

这些填充元素取值区域构成的集合记为Ω^{region}.

定义 7. 填充区域 Σ_{J_j,H_h}^{fill} 内填充元素取值记为 d_{J_j,H_h}^{fill},这些取值构成的集合记为 $D^{fill}=\{d_{J_j,H_h}^{fill}\}$.

$$d_{J_j,H_h}^{fill}=\min\{d\mid d\in\Omega_{J_j,H_h}^{region}/\{CE_{j',h'}^{SCDM}\};\text{If } n_1^{fill}>n_2^{fill},\ j'=J_j\ ;\text{else } n_1^{fill}\le n_2^{fill},\ h'=H_h;$$

$$\text{where row } J_j,\text{column } H_h \text{ constitute } \Omega_{J_j,H_h}^{region};\ j',h' \text{ are subscripts in } CE_{j',h'}^{SCDM}\in CE^{SCDM}\}.$$

通常,更加关注填充元素的个数及其取值范围,而对于该填充元素的具体填充位置,并不关心.

观察图 6,显然,第 4 行不满足行约束度(1<2),则需要在第 4 行填入 1 个计算元素;第 2 列不满足列约束度(1<3),则需要在第 2 列填入两个计算元素.所以,填充区域是 $CDM(Q^{seg},S^{seg}(i,i+m-1))$中的元素(4,2),见图 6 中标示;填充个数为 n^{fill}=max(1,2)=2;2>1,所以填充元素取值范围为第 2 列中元素构成的取值范围.

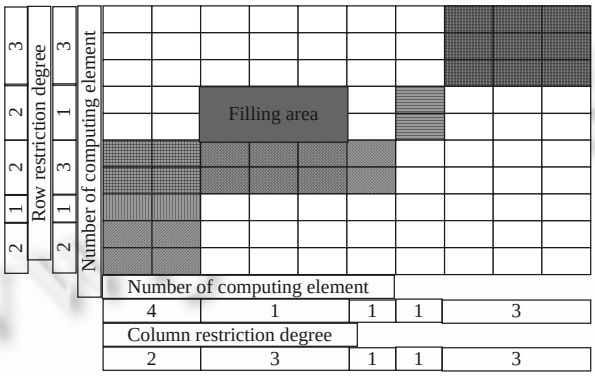


Fig.6 Filling area for computation elements in segmental cumulative distance matrix

图 6 分段累积距离矩阵中计算元素的填充区域

定理 4. 不失一般性,假定 $n_1^{fill}\le n_2^{fill}$.给定违反列约束度的列集合为 $H=(H_1,...,H_{n_{column}})$, N^{fill}, Ω^{region} 和 D^{fill},得到如下结论:

$$LB\_seg\_WF^2(Q,S(i,i+m-1))+\sum_{J_j\in J,H_h\in H}d_{J_j,H_h}^{fill}\times n_{J_j,H_h}^{fill}\le DTW^2(Q,S(i,i+m-1)).$$

证明:令 $W=w_1,w_2,\ldots,w_K$,max$(m,n)\le K\le m+n-1$ 为 $CDM(Q,S(i,i+m-1))$的最优扭曲路径;令 $P=p_1,p_2,\ldots,p_{K'}$, max$(M,N)\le K'\le M+N-1$ 为 W 在 $CDM(Q^{seg},S^{seg}(i,i+m-1))$中对应的扭曲路径.由连续性约束可知,$\Omega^{region}$ 必然与 P 相交,令相交的子路径的集合记为 P^{inter},再将 P^{inter} 划分为若干个不相交的子集合 P_l^{inter},每个子集合中的路径是连续的,且 $P^{inter}=\cup P_l^{inter}$.

令 P_l^{inter} 中的元素个数为 $n_{1,l}^{inter}$, P_l^{inter} 包含的分段块中包含 $S(i,i+m-1)$中元素的个数为 $n_{2,l}^{inter}$,包含 W 中元素的个数为 $n_{3,l}^{inter}$;令 P_l^{inter} 所在列中需要填入计算元素的个数为 $n_{4,l}^{inter}$(即在与 P_l^{inter} 相交的填充元素取值区域所对应的填充区域中,填充元素个数的总和);令 P_l^{inter} 包含的列数为 $n_{5,l}^{inter}$.根据连续性约束,容易得到 $n_{3,l}^{inter}\ge n_{2,l}^{inter}$.由列约束度的定义可知, $n_{4,l}^{inter}$ 的最大值为 $n_{2,l}^{inter}-n_{5,l}^{inter}$,而 $n_{3,l}^{inter}\ge n_{2,l}^{inter}$,所以有 $n_{3,l}^{inter}-n_{4,l}^{inter}\ge n_{5,l}^{inter}$.

接下来证明,在 P_l^{inter} 所包含的 W 的子路径集合 $W_{1,l}^{inter}=\{w_{1,s_l}^{inter},...,w_{1,e_l}^{inter}\}$ 中取出 $n_{4,l}^{inter}$ 个元素之后,剩下的子路径集合 $W_{2,l}^{inter}=\{w_{2,s_l}^{inter},...,w_{2,e_l}^{inter}\}$,仍能在 $CDM(Q^{seg},S^{seg}(i,i+m-1))$中构成一条连续子路径,而且这条子路径必然包含在 P_l^{inter} 中:

1. 当 $n_{1,l}^{inter}=1$时.因为有 $n_{3,l}^{inter}\ge n_{4,l}^{inter}+n_{5,l}^{inter}$, 所以在 $W_{1,l}^{inter}$ 中取出 $n_{4,l}^{inter}$ 个元素后,至少还剩 1 个距离($d(w)$)尽

可能小的元素,而这个元素仍能在 $CDM(Q^{seg},S^{seg}(i,i+m-1))$中构成 P_l^{inter},即构成一条连续子路径;

2. 当 $n_{1,l}^{inter}>1$,且 $n_{1,l}^{inter}=n_{5,l}^{inter}$ 时.因为有 $n_{3,l}^{inter}\geq n_{4,l}^{inter}+n_{5,l}^{inter}$,所以在 $W_{1,l}^{inter}$ 中取出 $n_{4,l}^{inter}$ 个元素后,至少还剩 $n_{5,l}^{inter}$ 个元素,只要确保 P_l^{inter} 所在的 $n_{5,l}^{inter}$ 个列中每列都保留 $W_{1,l}^{inter}$ 中的一个距离($d(w)$)尽可能小的元素,这样,剩下的子路径集合 $W_{2,l}^{inter}$ 就仍能在 $CDM(Q^{seg},S^{seg}(i,i+m-1))$中构成 P_l^{inter},即构成一条连续子路径;
3. 当 $n_{1,l}^{inter}>1$,且 $n_{1,l}^{inter}>n_{5,l}^{inter}$ 时.因为有 $n_{3,l}^{inter}\geq n_{4,l}^{inter}+n_{5,l}^{inter}$,所以在 $W_{1,l}^{inter}$ 中取出 $n_{4,l}^{inter}$ 个元素后,至少还剩 $n_{5,l}^{inter}$ 个元素;又因为 $n_{1,l}^{inter}>n_{5,l}^{inter}$,所以,此时剩下的 $n_{5,l}^{inter}$ 个元素在 $CDM(Q^{seg},S^{seg}(i,i+m-1))$中未必能够构成 P_l^{inter};但是,仍可以在每列都保留 $W_{1,l}^{inter}$ 中的一个距离($d(w)$)尽可能小的元素,这样也能构成长度为 $n_{5,l}^{inter}$ 的连续子路径,而且,这条路径为 P_l^{inter} 的子集.

令 $W_1^{inter}=\cup W_{1,l}^{inter}$, $W^{discon}=W-W_1^{inter}$, $W_2^{inter}=\cup W_{2,l}^{inter}$, $W^{remain}=W^{discon}\cup W_2^{inter}$, $W^{out}=W-W^{remain}$.经过上述分析可以得知,在 W 中取出总计 N^{fill} 个的元素之后,剩余部分 W^{remain} 在 $CDM(Q^{seg},S^{seg}(i,i+m-1))$中对应的扭曲路径 P^{remain} 是一条完整的(连续的)扭曲路径,且 $P^{remain}\subseteq P$.显然,$LB\_seg\_WF^2(Q,S(i,i+m-1))\leq d(P^{remain})\leq d(W^{remain})$.再分析 $\sum_{J_j\in J,H_h\in H}d_{J_j,H_h}^{fill}\times n_{J_j,H_h}^{fill}$.在 $CDM(Q^{seg},S^{seg}(i,i+m-1))$的第 $H_h\in H$ 列中,实际取出了 W 中的 $\sum_{J_j=1}^{N}n_{J_j,H_h}^{fill}$ 个距离($d(w)$)尽可能大的元素.而由 d_{J_j,H_h}^{fill} 的定义可知,d_{J_j,H_h}^{fill} 肯定小于或等于这 $\sum_{J_j=1}^{N}n_{J_j,H_h}^{fill}$ 个被取出的元素.所以可以得到 $\sum_{J_j\in J,H_h\in H}d_{J_j,H_h}^{fill}\times n_{J_j,H_h}^{fill}\leq d(W^{out})$.综上所述,可以得到

$$LB\_seg\_WF^2(Q,S(i,i+m-1))+\sum_{J_j\in J,H_h\in H}d_{J_j,H_h}^{fill}\times n_{J_j,H_h}^{fill}\leq d(W^{remain})+d(W^{out})=DTW^2(Q,S(i,i+m-1)).$$

结论得证. □

于是,得到新的下限函数:

$$LB\_seg\_WF2^2(Q,S(i,i+m-1))=LB\_seg\_WF^2(Q,S(i,i+m-1))+\sum_{J_j\in J,H_h\in H}d_{J_j,H_h}^{fill}\times n_{J_j,H_h}^{fill}.$$

注:根据 n_1^{fill},n_2^{fill} 的大小确定填充区域集内的填充个数、填充元素取值区域和填充元素的取值的方法,不一定就能确保由此得到的 $LB_seg_WF2^2(Q,S(i,i+m-1))$就是最优的.例如:当 $n_1^{fill}\leq n_2^{fill}$ 时,我们的方法是仅仅根据列约束度的要求确定填充区域集内的填充个数、填充元素取值区域和填充元素的取值,并由此求解 $LB\_seg\_WF2^2(Q,S(i,i+m-1))$;而很有可能出现的情况是,根据行约束度求得的 $LB_seg_WF2^2(Q,S(i,i+m-1))$要更大一些.这说明本文方法还有改进的余地,但这种改进的效果可能并不明显.所以,为了行文简单明了,在正文部分就不再给出进一步的改进说明,而是直接将这种思想体现在本文的算法实现中.

接下来,利用行(列)约束度来指导确定应该增加哪个计算元素.在图 4 中,图 4(c1)中最优扭曲路径的计算元素为$(d(5,6)+4)_{1,1}$,$(1)_{1,2}$,$(1+d(4,3))_{2,3}$.首先观察行,计算元素个数为(3,2),而相应的行约束度为(2,2),满足行约束度条件;再观察列,计算元素个数为(2,1,2),而相应的列约束度为(2,2,2),显然违背了列约束度条件.据此可以确定,在第 2 列再增加一个计算元素 n^{fill}=1.为了确定这个元素的取值,观察图 7,图中着色区域标记的元素就是图 4 中图 4(c1)包含的计算元素.根据列约束度的定义,只要能够确保第 3 列、第 4 列满足有两个计算元素即可,故而,选取的填充区域和填充元素取值区域为图中标记了网格的部分,再根据填充元素取值的定义,可以得到 $d_{\phi,2}^{fill}=d(4,7)$.进一步可知,$LB\_seg\_WF2^2(Q,S(i,i+m-1))$=8+9=17.

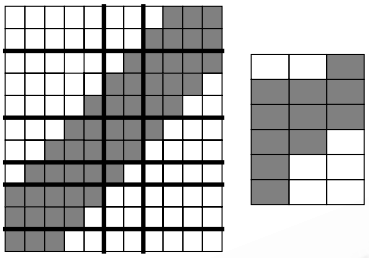
4	27	7	15	40	41	23
2	26	6	30	79	25	22
1	17	5	41	75	21	25
6	1	10	11	20	56	65
Q/*S*	5	3	7	9	0	3

Fig.7 Value area for filling elements in segmental cumulative distance matrix

图 7 分段累积距离矩阵中填充元素的取值区域

2.2 全局约束的分段DTW

为使本文的 SDTW(segmented dynamic time warping)技术满足 DTW 的全局约束条件,首先通过下列方式[4]构造 $CDM_{global}(Q^{seg},S^{seg}(i,i+m-1))$:限制 $CDM_{global}(Q^{seg},S^{seg}(i,i+m-1))$中的元素(对应 $CDM(Q,S(i,i+m-1))$中的某个分段块),均必须至少包含 $CDM_{global}(Q,S(i,i+m-1))$中的一个元素.具体过程如图 8 所示:图 8(a)代表 $CDM_{global}(Q,S(i,i+m-1))$,着色区域代表满足 Sakoe-Chiba 约束的元素,空白代表没有元素,扭曲宽度 ρ 为 $\rho=|m-n|+1$,其中,n 为 Q 的长度;图 8(b)代表通过上述构造方法得到的 $CDM_{global}(Q^{seg},S^{seg}(i,i+m-1))$,着色区域代表其中的元素,它们均包含 $CDM_{global}(Q,S(i,i+m-1))$中的至少一个元素.



(a) (b)

Fig.8 Cumulative distance matrix and segmental cumulative distance matrix under global constrains

图 8 全局约束下累积距离矩阵和分段累积距离矩阵

接下来,根据 Sakoe-Chiba 约束对第 2.1 节中的相关理论进行修改,并给出满足 Sakoe-Chiba 约束条件的 SDTW 技术,其中,ρ为扭曲宽度.由于分段距离 $d^{seg}_{modify}(j,h)$ 仅与该分段块中的元素相关,只要该分段块中的元素不变,则分段距离不会发生变化.根据得到 $CDM_{global}(Q^{seg},S^{seg}(i,i+m-1))$的方法可知,只要该分段块存在,其中的元素是不会发生任何变化的,所以分段距离 $d^{seg}_{modify}(j,h)$ 不需要进行任何修改.由于 $LB\_seg\_WF(Q,S(i,i+m-1))=DTW_{global}(Q^{seg},S^{seg}(i,i+m-1))$,显然,这不满足 Sakoe-Chiba 的约束条件,所以,将其修改为

$$LB\_seg\_WF_{global}(Q,S(i,i+m-1))=DTW_{global}(Q^{seg},S^{seg}(i,i+m-1)),$$

其中,$DTW_{global}(Q^{seg},S^{seg}(i,i+m-1))$是利用 CDMglobal 构造方法和分段距离 $d^{seg}_{modify}(j,h)$ 计算 $CDM_{global}(Q^{seg},S^{seg}(i,i+m-1))$得到的.

下面证明 $LB\_seg\_WF_{global}(Q,S(i,i+m-1))\le DTW_{global}(Q,S(i,i+m-1))$.

定理 5. 给定长度为 n 的查询序列 Q、流上子序列 $S(i,i+m-1)$以及它们的 APCA 分段子序列 Q^{seg} 和 $S^{seg}(i,i+m-1)$,则有 $LB\_seg\_WF_{global}(Q,S(i,i+m-1))\le DTW_{global}(Q,S(i,i+m-1))$.

证明:令 $GW=gw_1,gw_2,\ldots,gw_K$,$\max(m,n)\le K\le m+n-1$ 为 $CDM_{global}(Q,S(i,i+m-1))$的最优扭曲路径;令 $GP=gp_1,gp_2,\ldots,gp_{K'}$,$\max(M,N)\le K'\le M+N-1$ 为 GW 在 $CDM_{global}(Q^{seg},S^{seg}(i,i+m-1))$中对应的路径;令 GR 为 $CDM_{global}(Q^{seg},S^{seg}(i,i+m-1))$中的最优扭曲路径.根据 $CDM_{global}(Q^{seg},S^{seg}(i,i+m-1))$的构造方法,可以确保 GW 中的任何一个元素均可以找到一个 $CDM_{global}(Q^{seg},S^{seg}(i,i+m-1))$中的元素(分段块)包含它,而且根据连续性约束,我们可以确定 GP 是一条连续的扭曲路径.定义每个 gp_k 为 $gw_{s_k},\ldots,gw_{e_k}$,其中,gw_{s_k} 为包含在 gp_k 中的第 1 个元素,gw_{e_k} 为包含在 gp_k 中的最后一个元素.令 gp_k 中的计算元素的个数为 c_k,因为连续性约束不变,所以仍然有 $e_k-s_k+1\ge c_k$.又因为分段距离 $d^{seg}_{modify}(j,h)$ 不变,所以,GW,GP 和 GR 之间的相互关系完全类似于没有 Sakoe-Chiba 约束时的 W,P 和 R 之间的相互关系,只不过 GW,GP 和 GR 的存在范围比 W,P 和 R 要小.所以,类似于第 2.1 节中的证明,容易得知 $LB\_seg\_WF_{global}(Q,S(i,i+m-1))\le DTW_{global}(Q,S(i,i+m-1))$. □

下面考察行、列约束度的相关定义.由于其基本思想是增加额外的计算元素,使得 $CDM_{global}(Q^{seg},S^{seg}(i,i+m-1))$的计算元素个数尽可能地接近 $CDM(Q,S(i,i+m-1))$中最优扭曲路径的长度,从而使得下限函数的估计尽可能地逼近真实的 DTW 距离.由于它利用 $CDM(Q^{seg},S^{seg}(i,i+m-1))$中的行列数和最优扭曲路径长度,计算填充元素的个数,而与 $CDM(Q^{seg},S^{seg}(i,i+m-1))$的具体元素的分布情况和取值无关,所以,在 Sakoe-Chiba 约束下,定义

2、定义 4 仍然有用.通常,我们更加关注填充元素的个数及其取值范围,而对于该填充元素的具体填充位置不予关注,所以,定义 3 也不必修改.由于 $CDM_{global}(Q^{seg},S^{seg}(i,i+m-1))$ 中行、列元素个数不同于 $CDM_{global}(Q^{seg},S^{seg}(i,i+m-1))$,所以对填充元素取值区域和填充元素取值有所影响,对定义 5 做出如下修改:增加了"(j',h') exists",根据 $CDM_{global}(Q^{seg},S^{seg}(i,i+m-1))$中行、列元素的存在情况,对填充元素取值区域进行了界定.定义 6 原先是建立在定义 5 的基础上的,现在修改为建立在定义 5′的基础上.

定义 5′. 若 $n_1^{fill} > n_2^{fill}(n_1^{fill} \le n_2^{fill})$,则填充区域 Σ_{J_j,H_h}^{fill} 所在的行 J_j(列 H_h)组成的取值区域,称为该填充区域的填充元素取值区域:

$$\Omega_{J_j,H_h}^{region}=\begin{cases}\{d(Q_{j',k}^{seg},S_{h',l}^{seg}(i,i+m-1))\,|\,1\le k\le n_{j'},1\le l\le m_{h'};\ j'=J_j,1\le h'\le M,(j',h')\ \text{exists}\}, & \text{If } n_1^{fill}>n_2^{fill}\\ \{d(Q_{j',k}^{seg},S_{h',l}^{seg}(i,i+m-1))\,|\,1\le k\le n_{j'},1\le l\le m_{h'};h'=H_h,1\le j'\le N,(j',h')\ \text{exists}\}, & \text{If } n_1^{fill}\le n_2^{fill}\end{cases}.$$

这些填充元素取值区域构成的集合记为Ω^{region}.

最后,用行、列约束度相关定义(定义 2、定义 3、定义 4、定义 5′、定义 6)来改进 $LB\_seg\_WF_{global}(Q,S(i,i+m-1))$.

定理 6. 不失一般性,假定 $n_1^{fill} \le n_2^{fill}$.给定违反列约束度的列的集合为 $H=(H_1,\ldots,H_{column})$,$N^{fill}$,$\Omega^{region}$ 和 D^{fill}.得到如下结论:

$$LB\_seg\_WF_{global}^2(Q,S(i,i+m-1))+\sum_{J_j\in J,H_h\in H} d_{J_j,H_h}^{fill}\times n_{J_j,H_h}^{fill}\le DTW_{global}^2(Q,S(i,i+m-1)).$$

证明:证明过程完全类似于定理 4. □

至此,得到了 Sakoe-Chiba 约束下的 $DTW_{global}^2(Q,S(i,i+m-1))$ 的下限函数:

$$LB\_seg\_WF1_{global}^2(Q,S(i,i+m-1))=LB\_seg\_WF_{global}^2(Q,S(i,i+m-1))+\sum_{J_j\in J,H_h\in H} d_{J_j,H_h}^{fill}\times n_{J_j,H_h}^{fill}.$$

2.3 数据流上在线相似性搜索算法(OSSDS)

本文提出的 DTW 下限函数 $LB_seg_WF(LB_seg_WF_{global})$可能面临如下问题[3]:

给定(如图 9 所示)查询序列 Q 和流上两个子序列 $S(i,i+m-1),S(k,k+m-1),i<k\le i+m-1$.显然,两个子序列之间存在重复序列 $S(k,i+m-1)$.一般地,如果 $LB(Q,S(i,i+m-1))\le\varepsilon$,则很有可能 $LB(Q,S(k,k+m-1))\le\varepsilon$.所以,如果计算了 $DTW(Q,S(i,i+m-1))$,很可能还需要计算 $DTW(Q,S(k,k+m-1))$.也就是说,使用下限函数 $LB_seg_WF(LB_seg_WF_{global})$可能面临连续失效的问题.

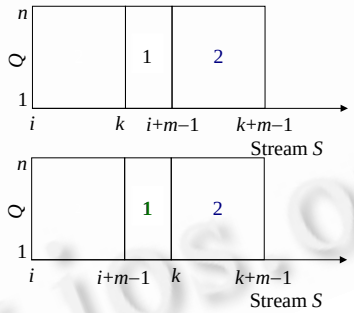


Fig.9 Overlapping of cumulative distance matrixs

图 9 相互重合的累积距离矩阵

观察 $CDM(Q,S(k,k+m-1)),CDM(Q,S(i,k+m-1))$发现,通过反向计算 $CDM(Q,S(i,k+m-1))$(起始点为$(n,k+m-1)$,终止点为$(1,i)$),$CDM_{反向}(Q,S(i,k+m-1))$中的元素$(1,k)$即为 $d(W_2)$,也就是在 $d(W_2)$和 $d(W_3)$之间存在直接计算关系.再观察 $CDM(Q,S(k,k+m-1)),CDM(Q,S(i,k+m-1))$,显然,$CDM(Q,S(i,k+m-1))$中的元素$(n,i+m-1)$即为 $d(W_4)$.于是,在 $d(W_4)$和 $d(W_3)$之间也存在直接计算关系.由此,$d(W_2)$可以通过 $d(W_3)$与 $d(W_4)$建立间接计算关系.因此,考虑利用这一点来减少估计 DTW 的冗余计算量,从而进一步减少计算量.

给定 $d(W_1),D_0$,根据上述分析,在没有全局约束的条件下,试图估算 $\sqrt{D_1}$ 的上限 UB_WF 和下限 LB_WF 以达

到减少计算量的目的.下面,首先给出 *UB_WF*:

引理 1. 令 $P_j=(j,k)$为列$(1,n:k)$中的某个点,$W_{opt}(P_j)$为$(1,i)$和$(n,k+m-1)$之间通过 P_j 的最优扭曲路径,则有

$$d(W(P_j)) \ge d(W_{opt}(P_j)) \ge \min(\{d(W_{opt}(P_j)) \mid P_j \in (1:n,k)\}) .$$

证明:对$(1,i)$和$(n,k+m-1)$之间的任意一条扭曲路径,它必须经过列$(1:n,k)$上的一点,记为 P_j,定义该路径为 $W(P_j)$,显然, $d(W(P_j)) \ge d(W_{opt}(P_j)) \ge \min(\{d(W_{opt}(P_j)) \mid P_j \in (1:n,k)\})$.

这意味着所有 $W_{opt}(P_j)$中,具有最小累积距离的路径为最优扭曲路径 W_{opt},则

$$D_0 = d(W_{opt}) = \min(\{d(W_{opt}(P_j)) \mid P_j \in (1:n,k)\}) .$$ □

定理 7. 1) 令 $P_j=(j,k)$为列$(1:n,k)$中的任意点,W_1 为$(1,i)$和(n,k)之间的最优扭曲路径,W_2 为$(1,k)$和$(n,k+m-1)$之间的最优扭曲路径,则有 $D_0 \ge d(W_1)+D_1-d_2$.

2) 令 $Q_j=(j,k)(1\le j\le n)$为$(1,i)$和$(n,k+m-1)$之间最优扭曲路径与列$(1:n,k)$的交点, W_1'' 为$(1,i)$和 Q_j 之间的最优扭曲路径, W_2'' 为 Q_j 和$(n,k+m-1)$之间的最优扭曲路径.如果 $W_1'' \cup (j+1:n,k)$ 构成$(1,i)$和(n,k)之间的最优扭曲路径,且 $W_2'' \cup (1:j-1,k)$ 构成$(1,k)$和$(n,k+m-1)$间的最优扭曲路径,则有 $D_0=d(W_1)+D_1-d_2$.

证明:1) 令 W_1' 为$(1,i)$和 P_j 之间的最优扭曲路径,则 $W_1' \cup (j+1:n,k)$ 形成了$(1,i)$和(n,k)之间的一条扭曲路径.

因为 W_1 为$(1,i)$和(n,k)之间的最优扭曲路径,所以有

$$d(W_1') + \sum_{t=j+1}^{n} d(t,k) \ge d(W_1) \quad (1)$$

令 W_2' 为 P_j 和$(n,k+m-1)$之间的最优扭曲路径,则 $W_2' \cup (1:j-1,k)$ 形成了$(1,k)$和$(n,k+m-1)$之间的一条扭曲路径.因为 W_2 为$(1,k)$和$(n,k+m-1)$之间的最优扭曲路径,所以

$$d(W_2') + \sum_{t=1}^{j-1} d(t,k) \ge d(W_2) \quad (2)$$

合并公式(1)、公式(2),则有: $d(W_2') + \sum_{t=1}^{j-1} d(t,k) + d(W_1') + \sum_{t=j+1}^{n} d(t,k) \ge d(W_2) + d(W_1)$.因为 $W_{opt}(P_j)$为$(1,i)$和$(n,k+m-1)$之间通过 P_j 的最优扭曲路径,所以 $W_{opt}(P_j) = W_1' \cup W_2' - (j,k)$,即

$$d(W_{opt}(P_j)) = d(W_1') + d(W_2') - d(j,k) \ge d(W_1) + d(W_2) - d_2 .$$

所以 $\min(d(W_{opt}(P_j))) \ge d(W_1) + d(W_2) - d_2$.又 $d(W_2)=D_1$,由引理 1 可得 $D_0 \ge d(W_1)+D_1-d_2$.

2) 结论显然. □

下面给出 *LB_WF*:

定理 8. 1) 令 $P_1=(1,k)$为列$(1,n:k)$中的第 1 行元素,W_2 为$(1,k)$和$(n,k+m-1)$之间的最优扭曲路径,则有

$$D_1 \ge D_0-d_1.$$

2) 如果 $W_2 \cup (1,i:k-1)$构成$(1,i)$和$(n,k+m-1)$之间一条最优扭曲路径,则 $D_1=D_0-d_1$.

证明:1) 令 $W_{opt}(P_1)$为$(1,i)$和$(n,k+m-1)$之间通过 P_1 的最优扭曲路径,则 $W_{opt}(P_1)=(1,i:k-1) \cup W_2$,所以

$$d(W_{opt}(P_1))=d_1+d(W_2).$$

令 W_{opt} 为$(1,i)$和$(n,k+m-1)$之间的最优扭曲路径,则 $D_0=d(W_{opt}) \le d(W_2)+d_1$.因为 $d(W_2)=D_1$,所以 $D_1 \ge D_0-d_1$.

2) 结论显然. □

综上所述, $\sqrt{D_1}$ 的上限 *UB_WF*、下限 *LB_WF* 分别为 $\sqrt{D_0 - d(W_1) + d_2}$ 和 $\sqrt{D_0 - d_1}$.由于在全局约束条件下 $DTW_{global}>DTW$,所以我们可以得到 $LB\_WF_{global}=LB\_WF$.

据此,在不具有全局约束条件下给出数据流上在线相似性搜索算法.算法中首先使用 *LB_seg_WF*$(Q,S(i,i+m-1))$排除明显不相似的子序列,如果不能排除,则计算 *DTW*$(Q,S(i,i+m-1))$.对于新形成的 $S(k,k+m-1)$,使用 *UB_WF*$(Q,S(k,k+m-1))$和 *LB_WF*$(Q,S(k,k+m-1))$作为过滤规则;如果成功过滤,则更新计算新到来的子序列的 *UB_WF* 和 *LB_WF*;如果不能成功过滤,则计算 *LB_seg_WF*$(Q,S(k,k+m-1))$:若不能成功,则计算 *DTW*$(Q,S(k,k+m-1))$(全局约束条件下的搜索算法与该算法类似,只需将 *LB_seg_WF* 替换为 $LB\_seg\_WF_{global}$,同时将 *UB_WF*, *LB_WF* 替换为 LB_WF_{global} 即可,在此不再给出).

算法 1. OSSDS.

输入:数据流 S、查询序列 Q、搜索阈值ε;

输出:满足搜索条件的查询结果.

1. 数据流 S 不断进行{
2. $Restart$=false;
3. 新数据 s_{i+m-1} 到达时,最近的 m 个数据组成流上子序列 $S(i,i+m-1)=\langle s_i,s_{i+1},\ldots,s_{i+m-1}\rangle$;
4. If ($LB\_seg\_WF(Q,S(i,i+m-1))\le\varepsilon$) {$dtw\_dist=DTW(Q,S(i,i+m-1))$;
5. If ($dtw\_dist\le\varepsilon$) {将子序列 $S(i,i+m-1)$放入查询结果集;$i=i+m$;$Restart$=true;}}
6. Else {$i=i+1$;$Restart$=true;}
7. If ($Restart$==false) {$k=i+1$;
8. While s_{k+m-1} arrives {s_{k+m-1} 到达后,最近 m 个数据组成序列 $S(k,k+m-1)=\langle s_k,s_{k+1},\ldots,s_{k+m-1}\rangle$;
9. $d(W_1)=CDM(Q,S(i,k+m-1))[n,k]$; $d_2=\sum_{t=1}^{n}d(t,k)$;$d_1=d_1+d(1,k-1)$;
10. If ($LB\_WF(Q,S(k,k+m-1))>\varepsilon$) {$k=k+1$ 转到第 8 行,继续 While 循环;}
11. ElseIf ($UB\_WF(Q,S(k,k+m-1))\le\varepsilon$) {将 $S(k,k+m-1)$放入查询结果集;$k=k+m$;Break;}
12. Break;}
13. $i=k$;}}

3 统计实验

本节将通过一系列的统计实验来验证 OSSDS 算法的有效性.采用的数据集来源于 UCR Time Series Data Mining Archive (Keogh 2002)[12].文中的实验数据使用了其中的部分子集,见表 2.

Table 2 Datasets

表 2 数据集

Name	Length
Power_data	1 274
Synthetic_data	1 410
Posture_data	4 778
Physio_data	242

为了简化实验,分别将上述 4 个数据子集读入内存,然后顺序访问其中的每个数据,从而模拟出 4 个数据流场景.每个待查询序列均来源于模拟的数据流,待查询序列的总数为 $Num_{TotalSeq}=Len_{stream}-len_{queried}+1$,其中,$Len_{stream}$ 代表数据流的长度,$len_{queried}$ 代表流上待查询序列的长度.令 Q,S 为两个序列,下限函数 $Lb(Q,S)$的近似程度 T 定义[6,9]为 $T=Lb(Q,S)/DTW(Q,S)$,T 的取值范围为[0,1].T 越大,近似程度越好.实验中,采用流上搜索得到的所有近似程度的均值 T_{ave} 作为度量下限函数近似程度的指标 $T_{ave}=\sum_{i=1}^{Num_{TotalSeq}}T_i\big/Num_{TotalSeq}$.

为了验证本文算法的有效性,首先通过统计实验 1 对第 2.1 节中提出的若干 DTW 下限函数的近似程度进行度量,并与 Lb_Yi[5]进行对比分析;其次,通过统计实验 2 对第 2.2 节中提出的若干 DTW 下限函数的近似程度进行度量;最后,通过统计实验 3 分别在具有和不具有全局约束条件下对 OSSDS 算法的性能进行度量.

统计实验 1. 统计实验 1 的主要目的是,在没有全局约束的条件下比较 DTW 下限函数 LB_seg_WF,LB_seg_WF2 和 Lb_Yi 的近似程度.查询序列长度取定为 n=60,流上待查询序列的长度分别取为 m=80,100,120.分段情况为:查询序列分成 N=6 段,待查询序列无论长度如何均分成 M=8 段.

表 3 列出了对比实验的结果:在数据集 1 和数据集 2 中,LB_seg_WF2 的 T_{ave} 均不如 Lb_Yi;在数据集 3 中,LB_seg_WF2 的 T_{ave} 优于 Lb_Yi;在数据集 4 中,LB_seg_WF2 的 T_{ave} 与 Lb_Yi 相当.Lb_Yi 是当前近似程度最好的 DTW 下限函数,但它不是 SDTW 技术,而 LB_seg_WF2 是一种 SDTW 技术,所以两者在估计分段序列的近似 DTW 方面是不具可比性的(通常,估计分段序列的近似 DTW 的近似程度比非分段序列要差).但是,利用 Lb_Yi 的近似程度作为衡量 LB_seg_WF2 近似程度的一种参考指标是可取的.定义两者的近似程度比为 $T_{com}=T_{ave}^{-wf2}\big/T_{ave}^{-yi}$,其中,$T_{ave}^{-wf2}$ 代表表 3 中所有 LB_seg_WF2 的 T_{ave} 平均值,T_{ave}^{-yi} 代表表 3 中所有 Lb_Yi 的 T_{ave}

平均值.计算得到 T_{com}=0.8463,这说明 *LB_seg_WF*2 的近似程度接近于 *Lb_Yi*,有效性得到了验证.

Table 3 Tightness comparison among different lower bounds without global constrains

表 3 没有全局约束条件下,不同下限函数的近似程度比较

ID	Name	*Lb_Yi*			*LB_seg_WF*			*LB_seg_WF*2		
		80	100	120	80	100	120	80	100	120
1	Power_data	0.762 0	0.781 2	0.800 2	0.429 9	0.360 3	0.339 7	0.539 6	0.495 3	0.478 0
2	Synthetic_data	0.946 1	0.959 7	0.968 4	0.477 8	0.431 6	0.397 8	0.776 0	0.790 6	0.750 6
3	Posture_data	0.274 7	0.267 8	0.265 3	0.363 0	0.331 8	0.300 8	0.369 7	0.336 9	0.304 6
4	Physio_data	0.818 5	0.863 3	0.898 2	0.578 9	0.547 8	0.537 3	0.797 9	0.862 5	0.781 1

统计实验 2. 统计实验 2 的主要目的是,在全局约束的条件下度量 DTW 下限函数 *LB_seg_WF_Global* 和 *LB_seg_WF*2*_Global* 的近似程度.查询序列长度取定为 *n*=60,流上待查询序列的长度分别取为 *m*=80,100,120.分段情况为:查询序列分成 *N*=6 段,待查询序列无论长度如何均分成 *M*=8 段.

表 4 列出了实验结果.对比发现,*LB_seg_WF*2*_Global* 的近似程度与 *LB_seg_WF*2 相当,这说明受全局约束条件的影响较弱.同样,利用 *Lb_Yi* 的近似程度作为衡量 *LB_seg_WF*2*_Global* 近似程度的一种参考指标,得到 T_{com}=0.8547.这说明 *LB_seg_WF*2*_Global* 的近似程度接近于 *Lb_Yi*,有效性得到了验证.

Table 4 Tightness comparison among different lower bounds under global constrains

表 4 全局约束条件下,不同下限函数的近似程度比较

ID	Name	*LB_seg_WF_Global*			*LB_seg_WF2_Global*		
		80	100	120	80	100	120
1	Power_data	0.441 7	0.374 1	0.352 4	0.566 8	0.527 1	0.494 4
2	Synthetic_data	0.480 3	0.435 3	0.401 3	0.785 2	0.792 7	0.752 7
3	Posture_data	0.336 3	0.318 7	0.293 9	0.346 2	0.327 1	0.298 2
4	Physio_data	0.577 0	0.548 3	0.537 5	0.807 0	0.876 8	0.781 3

统计实验 3. 统计实验 3 的主要目的是,分别在具有和不具有全局约束条件下度量 OSSDS 算法的性能.查询序列长度取定为 *n*=60,流上待查询序列的长度分别取为 *m*=80,100,120.分段情况为:查询序列分成 *N*=8 段,待查询序列无论长度如何均分成 *M*=8 段.

下面,对算法的性能指标进行定义.

定义过滤率为 $FR=\frac{Num_{TotalSear}-Num_{DTW}}{Num_{TotalSear}}$.其中,$Num_{TotalSear}$ 代表流上搜索的待查询序列总数,Num_{DTW} 代表需要计算 DTW 距离的待查询序列的总数.显然,过滤率 *FR* 代表流上被过滤掉(包括被确定为相似的待查询序列和被排除的不相似的待查询序列)的待查询序列总数占总共搜索的待查询序列总数的比率,反映了算法能够多大程度地减少计算 DTW 的次数.

定义相似率为 $SR=\frac{Num_{TotalMat}}{Num_{TotalSeq}}$.其中,$Num_{TotalSeq}$ 代表流上待查询序列的总数,取值为 $Len_{stream}-len_{queried}+1$;$Num_{TotalSeq}$ 代表符合搜索条件的待查询序列总数.显然,相似率主要反映出搜索阈值的大小.

将待查询序列的长度和相似率作为过滤率的两个主要影响因子.表 5~表 7 和图 10 给出了没有全局约束条件下,OSSDS 算法在各种数据集中取不同待查询序列长度 *m* 和相似率 *SR* 时过滤率 *FR* 的大小.

Table 5 *FR* under different length and *SR* in Posture_data dataset without global constrains

表 5 没有全局约束条件下,Posture_data 数据集中取不同待查询序列长度和相似率时的过滤率

ID	*length*=80		*length*=100		*length*=120	
	SR	*LB_seg_WF*2 & *UB_WF* & *LB_WF*	*SR*	*LB_seg_WF*2 & *UB_WF* & *LB_WF*	*SR*	*LB_seg_WF*2 & *UB_WF* & *LB_WF*
1	0.004 5	0.914 4	0.001 5	0.941 2	0.006 4	0.900 8
2	0.002 5	0.927 9	0.002 8	0.940 0	0.005 8	0.921 0
3	0.001 7	0.930 8	0.005 1	0.925 2	0.004 3	0.937 5
4	0.001 5	0.932 1	0.006 8	0.905 3	0.003 9	0.942 2
5	0.006 2	0.914 8	0.008 5	0.869 7	0.002 1	0.946 0

Table 6 *FR* under different length and *SR* in Power_data dataset without global constrains

表 6 没有全局约束条件下,Power_data 数据集中取不同待查询序列长度和相似率时的过滤率

ID	*length*=80		*length*=100		*length*=120	
	SR	*LB_seg_WF2* & *UB_WF&LB_WF*	*SR*	*LB_seg_WF2* & *UB_WF&LB_WF*	*SR*	*LB_seg_WF2* & *UB_WF&LB_WF*
1	0.001 7	0.959 5	0.001 7	0.976 5	0.004 3	0.982 1
2	0.003 3	0.961 3	0.002 6	0.981 8	0.002 6	0.967 4
3	0.004 2	0.975 0	0.004 3	0.975 0	0.001 7	0.978 2
4	0.005 0	0.962 6	0.006 0	0.966 8	0.005 2	0.963 7
5	0.005 9	0.982 9	0.010 2	0.825 6	0.007 8	0.821 4

Table 7 *FR* under different length and *SR* in Synthetic_data dataset without global constrains

表 7 没有全局约束条件下,Synthetic_data 数据集中取不同待查询序列长度和相似率时的过滤率

ID	*length*=80		*length*=100		*length*=120	
	SR	*LB_seg_WF2*& *UB_WF&LB_WF*	*SR*	*LB_seg_WF2*& *UB_WF&LB_WF*	*SR*	*LB_seg_WF2*& *UB_WF&LB_WF*
1	0.003 8	0.993 6	0.003 1	0.994 5	0.007 0	0.954 5
2	0.004 5	0.991 8	0.009 2	0.894 3	0.008 5	0.891 1
3	0.012 8	0.746 3	0.003 8	0.992 6	0.005 4	0.982 5
4	0.008 3	0.974 0	0.010 7	0.416 7	0.003 9	0.991 4
5	0.011 3	0.890 4	0.007 6	0.965 7	0.006 2	0.973 5

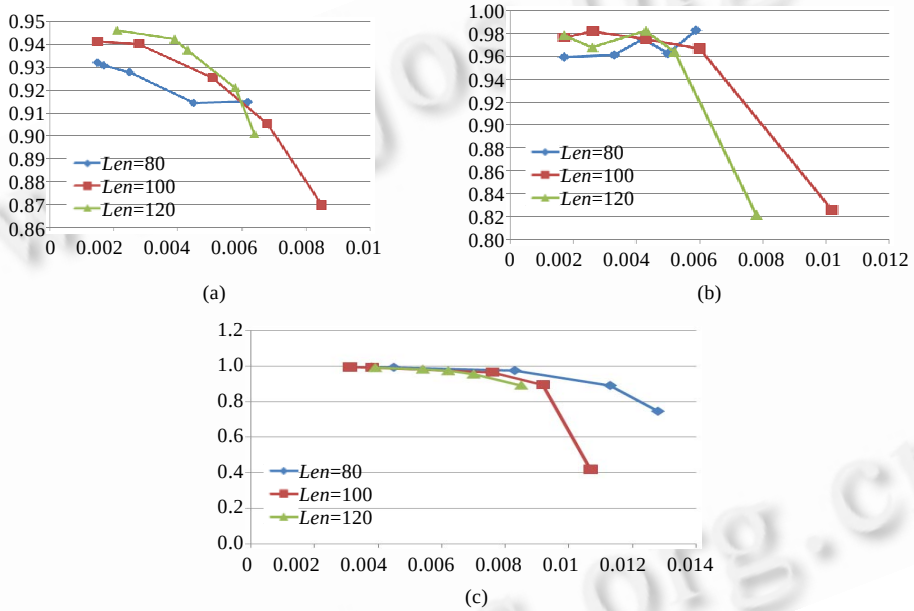


Fig.10 *FR* under different length and *SR* in different datasets without global constrains

图 10 没有全局约束条件下,不同数据集中取不同待查询序列长度和相似率时的过滤率

从图 10 可以看出,当 *SR* 不超过 0.01 时,*FR* 均保持在 0.8 以上;当 *SR* 不超过 0.005 时,*FR* 均保持在 0.9 以上;但当 *SR* 超过 0.01 时,图 10(c)中出现了 *FR* 急剧下降的情况.由此可以看出,只要 *SR* 保持在(0,0.01)范围内,本文算法的性能就能保持在较高的水平上.另外,当待查询序列长度保持不变时,过滤率基本上是随着相似率递减的.这是因为相似率变大,意味着搜索阈值变大,即候选的相似子序列变多.此时,过滤条件失效的次数 Num_{DTW} 也随之增高,而总的搜索次数一般会保持不变或减少,故此过滤率有所降低.综上所述,当 *SR* 保持在(0,0.01)范围内,算法的性能比较稳定.

表 8~表 10 和图 11 中给出了全局约束条件下,本文算法在各种数据集中取不同待查询序列长度 *m* 和相似率 *SR* 时过滤率 *FR* 的大小.

Table 8 *FR* under different length and *SR* in Posture_data dataset under global constrains

表 8 全局约束条件下,Posture_data 数据集中取不同待查询序列长度和相似率时的过滤率

ID	*length*=80		*length*=100		*length*=120	
	SR	*LB_seg_WF2_Global* & *LB_WF_Global*	*SR*	*LB_seg_WF2_Global* & *LB_WF_Global*	*SR*	*LB_seg_WF2_Global* & *LB_WF_Global*
1	0.003 2	0.882 0	0.001 1	0.940 6	0.005 8	0.872 9
2	0.001 7	0.919 5	0.001 9	0.928 1	0.005 4	0.912 6
3	0.005 1	0.861 4	0.004 5	0.905 0	0.004 1	0.930 2
4	0.006 2	0.851 9	0.006 2	0.869 1	0.003 4	0.931 3
5	0.008 5	0.794 7	0.010 0	0.629 6	0.002 1	0.936 8

Table 9 *FR* under different length and *SR* in Power_data dataset under global constrains

表 9 全局约束条件下,Power_data 数据集中取不同待查询序列长度和相似率时的过滤率

ID	*length*=80		*length*=100		*length*=120	
	SR	*LB_seg_WF2_Global* & *LB_WF_Global*	*SR*	*LB_seg_WF2_Global* & *LB_WF_Global*	*SR*	*LB_seg_WF2_Global* & *LB_WF_Global*
1	0.001 7	0.951 8	0.004 3	0.986 8	0.003 5	0.905 7
2	0.002 5	0.929 0	0.002 6	0.963 6	0.004 3	0.982 1
3	0.004 2	0.918 7	0.010 2	0.860 5	0.005 2	0.961 5
4	0.005 9	0.982 9	0.006 0	0.966 8	0.008 7	0.845 2
5	0.006 7	0.955 6	0.001 7	0.951 9	0.002 6	0.919 8

Table 10 *FR* under different length and *SR* in Synthetic_data dataset under global constrains

表 10 全局约束条件下,Synthetic_data 数据集中取不同待查询序列长度和相似率时的过滤率

ID	*length*=80		*length*=100		*length*=120	
	SR	*LB_seg_WF2_Global* &*LB_WF_Global*	*SR*	*LB_seg_WF2_Global* &*LB_WF_Global*	*SR*	*LB_seg_WF2_Global* &*LB_WF_Global*
1	0.002 3	0.996 3	0.004 6	0.990 2	0.007 7	0.891 1
2	0.001 5	0.997 4	0.009 9	0.416 7	0.003 1	0.993 9
3	0.003 0	0.995 1	0.005 3	0.987 1	0.004 6	0.987 9
4	0.006 8	0.983 9	0.003 1	0.994 5	0.001 5	0.997 2
5	0.006 0	0.987 1	0.007 6	0.965 7	0.002 3	0.995 7

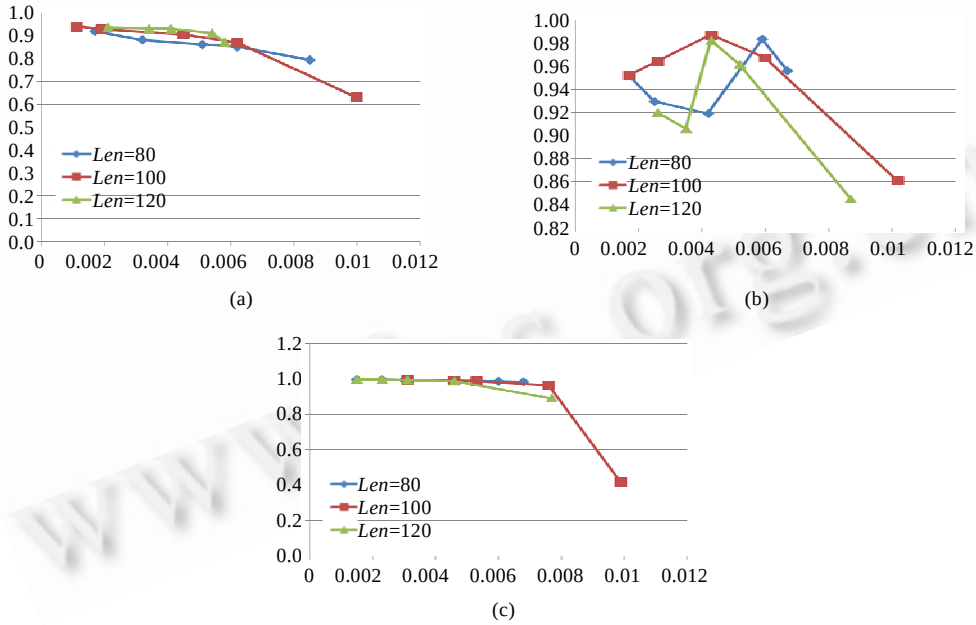


Fig.11 *FR* under different length and *SR* in different datasets under global constrains

图 11 全局约束条件下,不同数据集中取不同待查询序列长度和相似率时的过滤率

从图 11 可以看出,当 *SR* 不超过 0.008 时,*FR* 均保持在 0.8 以上;当 *SR* 不超过 0.002 时,*FR* 均可保持在 0.9 以上;同样地,当 *SR* 超过 0.01 时,图 11(a)和图 11(c)中均出现了 *FR* 急剧下降的情况.由此可以看出,只要 *SR* 保持在(0,0.008)范围内,本文算法的性能就能保持在较高的水平上.与非全局约束条件下的算法性能相比,全局约束条件下 OSSDS 的算法性能较差一些,这是因为过滤手段中缺少了上限函数 *UB_WF_Global* 的缘故.

4 结束语

当前,大部分下限函数都是针对全局约束条件下等长序列相似性搜索提出来的,都不能处理不具有全局约束条件的情况或非等长序列相似性搜索的情况.而且,根据数据流上在线搜索的实时特性,难以为 DTW 建立相应的索引结构.针对这些情况,本文首先在具有或不具有全局约束条件下分别提出了没有索引结构的 DTW 下限函数 $LB\_seg\_WF_{global}$ 和 *LB_seg_WF*;接着,为了进一步提高 $LB\_seg\_WF_{global}$ 和 *LB_seg_WF* 的近似程度,又提出了一系列的改进方法.然后,针对流上使用 $LB\_seg\_WF_{global}$ 或 *LB_seg_WF* 可能会出现连续失效的情况,分别提出了 DTW 的下限函数 LB_WF_{global}(具有全局约束条件)和上限函数 *UB_WF*、下限函数 *LB_WF*(不具有全局约束条件),通过增量的方式快速估计 DTW,大幅度地减少了估计 DTW 的冗余计算量.最后,在这些工作的基础上,提出了具有或不具有全局约束条件下没有索引结构的非等长序列的数据流上在线相似性搜索算法.通过理论分析和统计实验,验证了本文方法的有效性.但也发现了一些问题,比如:当 *SR* 大于 0.01 时,OSSDS 的性能急剧下降,不太稳定.这些问题将留待以后的工作中加以研究和克服.

References:

[1] Faloutsos C, Ranganathan M, Manolopoulos Y. Fast subsequence matching in time-series databases. In: Snodgrass RT, Winslett M, eds. Proc. of the 1994 ACM SIGMOD Int'l Conf. on Management of Data. Minneapolis: ACM Press, 1994. 419−429.

[2] Pan D, Shen JY. Similarity discovery techniques in temporal data mining. Journal of Software, 2007,18(2):246−258 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/18/246.htm

[3] Zhou M, Wong MH. Efficient online subsequence searching in data streams under dynamic time warping distance. In: Cilia M, ed. Proc. of the 24th Int'l Conf. on Data Engineering (ICDE 2008). Cancun: IEEE Computer Society, 2008. 686−694.

[4] Shou Y, Mamoulis N, Cheung DW. Fast and exact warping of time series using adaptive segmental approximations. Journal of Machine Learning, 2005,58(2):231−267.

[5] Yi B, Jagadish H, Faloutsos C. Efficient retrieval of similar time sequences under time warping. In: Sipple RS, ed. Proc. of the 14th Int'l Conf. on Data Engineering (ICDE'98). Orlando: IEEE Computer Society, 1998. 201−208.

[6] Keogh E. Exact indexing of dynamic time warping. In: Papadias D, ed. Proc. of the 28th ACM VLDB Int'l Conf. on Very Large Data Bases. Hong Kong: ACM Press, 2002. 406−417.

[7] Keogh EJ, Chakrabarti K, Mehrotra S, Pazzani MJ. Locally adaptive dimensionality reduction for indexing large time series databases. In: Aref WG, ed. Proc. of the 2001 SIGMOD Int'l Conf. on Management of Data. Santa Barbara: ACM Press, 2001. 151−162.

[8] Vlachos M, Hadjieleftheriou M, Gunopulos D, Keogh EJ. Indexing multi-dimensional time-series with support for multiple distance measures. In: Getoor L, Senator TE, eds. Proc. of the 9th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. Washington: ACM Press, 2003. 216−225.

[9] Zhu Y, Shasha D. Warping indexes with envelope transforms for query by humming. In: Halevy AY, Ives ZG, Doan A, eds. Proc. of the 2003 ACM SIGMOD Int'l Conf. on Management of Data. San Diego: ACM Press, 2003. 181−192.

[10] Han WS, Lee J. Ranked subsequence matching in time-series databases. In: Koch C, Gehrke J, Garofalakis MN, Srivastava D, Aberer K, Deshpande A, Florescu D, Chan CY, Ganti V, Kanne CC, Klas W, Neuhold EJ, eds. Proc. of the 33rd ACM VLDB Int'l Conf. on Very Large Data Bases. Vienna: ACM Press, 2007. 423−432.

[11] Sakurai Y, Faloutsos C, Yamamuro M. Stream monitoring under the time warping distance. In: Korpeoqlu I, ed. Proc. of the 23rd Int'l Conf. on Data Engineering (ICDE 2007). Istanbul: IEEE Computer Society, 2007. 1046−1055.

[12] The UCR time series data mining archive. http://www.cs.ucr.edu/~eamonn/TSDMA/index.html

附中文参考文献:

[2] 潘定,沈钧毅.时态数据挖掘的相似性发现技术.软件学报,2007,18(2):246−258. http://www.jos.org.cn/1000-9825/18/246.htm

吴枫(1980−),男,湖北宜昌人,博士生,主要研究领域为数据挖掘,网络安全,人工智能.

仲妍(1982−),女,博士生,主要研究领域为高性能计算,数据挖掘.

吴泉源(1942−),男,教授,博士生导师,主要研究领域为分布计算,人工智能,数据库管理.

贾焰(1960−),女,博士,教授,博士生导师,CCF 高级会员,主要研究领域为数据库,分布计算,人工智能.

杨树强(1969−),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为海量信息处理,数据库,分布计算.