

# 基于动态规划求解时间序列 DTW 中心

孙 焘<sup>1</sup> 夏 斐<sup>1</sup> 刘洪波<sup>2</sup>

(大连理工大学创新实验学院 大连 116024)<sup>1</sup> (大连海事大学计算机学院 大连 116024)<sup>2</sup>

**摘 要** 中心时间序列表明了一个时间序列集合中的公共特征,是时间序列聚类的重要手段。提出了一个利用动态规划求解两条时间序列 DTW 中心的方法,即以最小化中心序列到两条样本序列的 DTW 距离平方和为目标,递归求解最优解。在此基础上,给出了基于中心与样本匹配度的剪枝方法,降低了时间复杂度。并在理论上证明了该方法可以获得最优解。实验结果显示,相比于 DBA 算法,该算法能够获得更小的 DTW 距离平方和,并且有更好的鲁棒性。

**关键词** 中心时间序列,DTW,动态规划,匹配度剪枝

**中图法分类号** TP391.4 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2015.12.060

## Calculating DTW Center of Time Series Using Dynamic Planning

SUN Tao<sup>1</sup> XIA Fei<sup>1</sup> LIU Hong-bo<sup>2</sup>

(School of Innovation and Experiment, Dalian University of Technology, Dalian 116024, China)<sup>1</sup>

(School of Computer Science, Dalian Maritime University, Dalian 116024, China)<sup>2</sup>

**Abstract** The central time series plays an important role in time series clustering, which indicates the common features of time series. We proposed dynamic planning approach called DPSSD to calculate central time series of two time series. The approach is recursive based on the minimizing sum of squares of DTW (SSD) distance from central series to two sample series. Degree-pruning was also introduced to decrease the algorithm time complexity. The proposed algorithm was proved theoretically. It can achieve the optimal solution. In the experiments, the results illustrate that our approaches have much better performance and robustness than DBA, which is measured by SSD.

**Keywords** Central time series, DTW, Dynamic planning, Degree-pruning

## 1 引言

作为一种重要的数据形式,时间序列被广泛地应用于各个领域,如生物、金融、多媒体以及图像分析等<sup>[1]</sup>,由于其重要性和普遍性,被世界范围内的学者广泛研究。时间序列聚类是其中最为典型的一种,其目标是通过将未标注的时间序列集合中的序列划分到不同的组别中,确保组内的序列相似程度最大、组间序列差异程度最大,从而识别时间序列集合的结构<sup>[2]</sup>,这个问题常常转化为如何衡量时间序列的距离并计算序列的中心<sup>[3]</sup>。就时间序列的相似程度而言,有多种衡量方法,如欧氏距离、Mikowski 距离、Pearsons 相关系数距离、KullbackLiebler 距离和动态时间规整(DTW)距离,其中,欧氏距离和 DTW 距离是使用最为广泛的两种。前者非常简单且容易应用,大多数关于时间序列聚类研究都是简单地计算每一条时间序列在相同位置处的欧氏距离的平均值,以此作为中心时间序列<sup>[4]</sup>,当时间序列变化的同步性较好时,这种基于欧氏距离的中心时间序列非常有效。但在实际应用中,大多数的时间序列都会有一定程度的伸缩和偏移,这种情况下欧氏距离的计算方法会损失序列的形状特征。相反,DTW 距离允许在时间上伸缩和偏移,从而发现不同相位上相似的

形状<sup>[5]</sup>,但同时带来的问题是很难计算这种情况下的中心时间序列。本文提出了一种动态规划方法,通过最小化中心序列到两条样本序列的动态时间规整(Dynamic Time Warping, DTW)的距离平方和(Sum of Squares of DTW, SSD)来求解两条时间序列的中心序列,称为 DPSSD(Dynamic Planning based on SSD)算法,并使用基于度的剪枝方法来降低其时间复杂度。

本文第 2 节讨论了早期研究的相关工作以及对应的方法;第 3 节指出了相关概念并给出了相关理论和证明;第 4 节阐述了 DPSSD 方法;第 5 节详细给出了实验的结果和讨论;最后总结了本文工作并提出了未来的进一步工作计划。

## 2 相关工作

对于时间序列聚类问题,国内的研究者多从降维的角度出发,通过使用关键点抽取<sup>[6]</sup>、独立成分分析<sup>[7]</sup>等特征提取手段,降低原始序列维度,提高聚类效率,而鲜有关于求取时间序列中心的研究。

由于很难给出时间序列的中心,一些国外的研究者尝试找到自己的解决方法。Gupta 等人<sup>[8]</sup>介绍了一种非线性校准和平均滤波方法来寻找两个时间序列的平均值。但是该方法

到稿日期:2014-12-04 返修日期:2015-03-16 本文受国家自然科学基金(61472058)资助。

孙 焘(1975—),男,博士,副教授,主要研究方向为数据挖掘, E-mail: dltst@dlut.edu.cn; 夏 斐(1990—),男,硕士,主要研究方向为数据挖掘; 刘洪波(1971—),男,博士,教授,主要研究方向为数据挖掘。

无法确保中心时间序列和两个原始时间序列的距离相等,这与 Salvador 的说法相矛盾,他认为中心时间序列与两个原始时间序列的距离应该是相等的<sup>[9]</sup>。Niennattrakul<sup>[10]</sup>介绍了一种三次样条插值的方法,使平均时间序列到原始时间序列的距离相等,然后通过迭代,不断调整平均时间序列,从而平衡其到原始序列的距离。但是该方法有一个缺点:中心时间序列仅仅通过简单计算每一条时间序列在相同位置上的欧氏距离的平均值得到,这损失了时间序列的形状特征,所以平均时间序列到所有时间序列的距离之和不是最小的<sup>[9-11]</sup>。动态时间规整(DTW)是 Berndt 和 Lifford<sup>[5]</sup>提出的,用于衡量时间序列的相似程度,它使用非线性的对齐方式,通过在时间轴上的伸缩和偏移来校准时间序列。

Petitjean 等人<sup>[11]</sup>提出了 DTW 重心平均(DTW Barycenter Averaging, DBA)的方法,通过平均一个时间序列集合来获得中心时间序列。初始的时间序列是随机选择的,然后计算每一个独立的时间序列和临时的中心时间序列的距离,利用计算的结果优化并生成新的临时中心时间序列,重复迭代这个过程,直到得到的新的中心时间序列稳定,其采用组内 DTW 距离平方和(Within-Group Sum of Squares, WGSS)来衡量算法效果。DBA 是一个近似迭代算法,也是目前最流行的方法,但是其获得的中心时间序列是次优解。

本文提出了一种新的方法,通过其能够得到与两条时间序列 DTW 距离平方和(即 SSD)最小的中心时间序列的理论上的最优解,其中 SSD 在组内仅有两条序列时,其值与 WGSS 相同。

### 3 DTW 距离和匹配度理论

为了规范化我们的问题,本节定义了一些相关概念,并对一些定理做了证明。

#### 3.1 DTW 距离和约束

时间序列是欧氏空间中的一个  $n$  维向量  $x[x_1, x_2, \dots, x_n]$ ,  $x_i$  或者  $x(i)$  是其第  $i$  个元素,且  $x(i_1:i_2)=[x_{i_1}, x_{i_1+1}, \dots, x_{i_2}]$  ( $i_1 \leq i_2$ ) 是  $x$  的子序列。

在计算给定的两个时间序列的距离时,两者的匹配关系常常不是一对一的,而是为了匹配形状特征进行了规整。给定长度为  $m$  的时间序列  $x[x_1, x_2, \dots, x_m]$  和长度为  $n$  的时间序列  $y[y_1, y_2, \dots, y_n]$ , 用  $2 \times L$  的矩阵  $W[[w_{11}, w_{12}]^T, [w_{21}, w_{22}]^T, \dots, [w_{L1}, w_{L2}]^T]$  来表示  $x$  和  $y$  的匹配关系,这里的  $L$  是匹配路径的长度,满足  $w_{k1} \in \{1, 2, \dots, m\}$ ,  $w_{k2} \in \{1, 2, \dots, n\}$ ,  $k \in \{1, 2, \dots, L\}$ , 即  $x_{w_{k1}}$  和  $y_{w_{k2}}$  匹配。若交换矩阵  $W$  的两行,则用  $W^{-1}$  表示新得到的矩阵。如此,若  $[w_{k1}, w_{k2}]^T$  是  $W$  的第  $k$  列,则  $W^{-1}$  的第  $k$  列为  $[w_{k2}, w_{k1}]^T$ 。

对于给定的长度为  $m$  的时间序列  $x$  和长度为  $n$  的时间序列  $y$ , 以及它们的匹配路径  $W$ , 用  $w_{ij}$  表示  $W$  的第  $i$  行、第  $j$  列的元素。式(1)定义了操作符  $\mathcal{M}(x, y, W, i)$ , 表示在路径  $W$  上与  $x(i)$  匹配的所有  $y$  上的元素的集合。式(2)定义了操作符  $\mathcal{R}(W, i)$ , 表示在路径  $W$  上与  $x(i)$  匹配的  $y(j_a:j_b)$  的元组  $(j_a, j_b)$ 。

$$\mathcal{M}(x, y, W, i) = \{y(j) | w_{k2} = i, j = w_{k1}\}, i = 1, \dots, m \quad (1)$$

$$\mathcal{R}(W, i) = (\min\{w_{j1} | w_{j2} = i\}, \max\{w_{j1} | w_{j2} = i\}), i = 1, \dots, m \quad (2)$$

定义 1(匹配度) 给定一个从时间序列  $x$  到时间序列  $y$  的 DTW 匹配路径  $W_{xy}$ ,  $Degree(W, i)$  代表  $\mathcal{M}(x, y, W, i)$  集合的大小,称为匹配度。

令  $\rho(W, x, y)$  表示在匹配路径  $W$  约束下的  $x$  和  $y$  的距离,其大小和  $W$  相关。其值可以用式(3)来定义。这里的  $[w_{k1}, w_{k2}]^T$  是  $W$  的第  $k$  列,  $L$  是匹配路径的长度。

$$\rho(W, x, y) = \sum_{k=1}^L (x_{w_{k1}} - y_{w_{k2}})^2 \quad (3)$$

在 DTW 度量中,其规整路径是受一些约束的,如边界条件、连续性和单调性。对于给定的长度为  $m$  的时间序列  $x$  和长度为  $n$  的时间序列  $y$ , 它们长度为  $L$  的匹配路径  $W$  满足以下约束:

$$1) [w_{11}, w_{12}] = [1, 1] \text{ 且 } [w_{L1}, w_{L2}] = [m, n];$$

$$2) \text{ 对于 } k = 1, 2, \dots, L-1, w_{k+1,1} - w_{k1} \leq 1 \text{ 且 } w_{k+1,2} - w_{k2} \leq 1;$$

$$3) \text{ 对于 } k = 1, 2, \dots, L-1, w_{k+1,1} - w_{k1} \geq 0 \text{ 且 } w_{k+1,2} - w_{k2} \geq 0.$$

#### 3.2 匹配度理论和剪枝定理

定理 1 对于给定的两个时间序列  $x$  和  $y$  以及它们的 DTW 匹配路径  $W$ , 若  $(i, j)$  是  $W$  上的一个元素, 那么有  $Degree(W, i) = 1$  或  $Degree(W^{-1}, j) = 1$ 。

证明: 假设  $[i, j]^T$  是  $W$  的一列,  $Degree(W, i) > 1$  或  $Degree(W^{-1}, j) > 1$ 。那么删除  $[i, j]^T$  不会影响 DTW 约束,但是减少了 DTW 距离,所以假设不成立,命题得证。

对于给定的长度为  $m$  的时间序列  $x$  和长度为  $n$  的时间序列  $y$ , 为了简便, 对于  $i = 1, \dots, m, j = 1, \dots, n$ , 用  $DTW(i, j)$  表示  $DTW(x(1:i), y(1:j))$ 。令  $d^2(i, j) = (x(i) - y(j))^2$ 。式(4)给出了  $DTW(i, j)$  的递归关系。

$$DTW(i, j) = \begin{cases} 0, & \text{if } i=0 \text{ and } j=0 \\ \infty, & \text{else if } i=0 \text{ or } j=0 \\ d^2(i, j) + \min \begin{cases} DTW(i-1, j-1) \\ DTW(i-1, j) \\ DTW(i, j-1) \end{cases}, & \text{otherwise} \end{cases} \quad (4)$$

定义 2(两个时间序列的 DTW 中心) 给定两个时间序列  $x$  和  $y$ , 关于  $x$  和  $y$  的最小距离的 DTW 中心定义为式(5)。

$$Cent(x, y) = \operatorname{argmin}_t (DTW(x, t) + DTW(t, y)) \quad (5)$$

图 1 给出了一个时间序列中心的例子, 图中时间序列  $c$  就是  $x$  和  $y$  的中心。注意, 该中心时间序列与  $x, y$  有着相同的长度和整体上相似的形状。

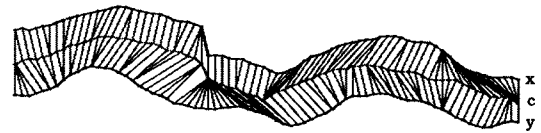


图 1 一个时间序列中心的例子

给定两个时间序列  $x[x_1, x_2, \dots, x_m]$  和  $y[y_1, y_2, \dots, y_n]$ , 设时间序列  $c[x]$  是  $x$  和  $y$  长度为 1 的中心序列, 则式(6)定义了最小距离和。

$$err(x, y) = \min_x (\sum_{i=1}^m (x_i - x)^2 + \sum_{j=1}^n (y_j - x)^2) \quad (6)$$

为了找到满足使  $\sum_{i=1}^m (x_i - x)^2 + \sum_{j=1}^n (y_j - x)^2$  取得最小值的  $x$ , 令:

$$\frac{d(\sum_{i=1}^m (x_i - x)^2 + \sum_{j=1}^n (y_j - x)^2)}{dx} = 0 \quad (7)$$

求解可得对应的  $x$ , 用  $avg(x, y)$  表示, 其定义为式(8)。

$$avg(x, y) = \frac{\sum_{i=1}^m x_i + \sum_{j=1}^n y_j}{m+n} \quad (8)$$

根据式(6)、式(8)以及 DTW 路径匹配关系, 我们得出定理 2。

**定理 2** 给定两个时间序列  $x$  和  $y$ , 令  $c$  为二者  $k$  长度的 DTW 中心, 用  $W_\alpha$  和  $W_\gamma$  分别表示从  $c$  到  $x$  和从  $c$  到  $y$  的匹配路径。那么对于  $t=1, 2, \dots, k$ , 有  $c(t) = avg(\mathcal{M}(x, y, W_\alpha, i) \cup \mathcal{M}(x, y, W_\gamma, i))$ 。

根据 DTW 以及 DTW 中心的定义, 派生出定理 3 的剪枝策略。

**定理 3** 对于给定的两个时间序列  $x$  和  $y$ , 存在二者的 DTW 中心  $c$  及其相应的 DTW 匹配路径  $W_\alpha$  和  $W_\gamma$ , 满足: 若  $[t, p]^T \in W_\alpha$ ,  $[t, q]^T \in W_\gamma$ , 则有  $Degree(W_\alpha, t) = Degree(W_\gamma^{-1}, q) = 1$  或  $Degree(W_\gamma, t) = Degree(W_\alpha^{-1}, p) = 1$ 。

证明: 设  $c, W_\alpha$  和  $W_\gamma$  是式(5)的任意解, 证明的思路是用递归的方法构造另一个解  $c', W'_\alpha$  和  $W'_\gamma$ , 同时满足式(5)和定理 3。假设  $[t, p]^T$  是  $W_\alpha$  的第  $a$  列,  $[t, q]^T$  是  $W_\gamma$  的第  $b$  列。根据定义 1, 有  $Degree(W_\alpha, t) = 1$  或  $Degree(W_\alpha^{-1}, p) = 1$ , 以及  $Degree(W_\gamma, t) = 1$  或  $Degree(W_\gamma^{-1}, q) = 1$ 。故不失一般性, 我们考虑下面两种情况。

若  $Degree(W_\alpha^{-1}, p) > 1$  且  $Degree(W_\gamma^{-1}, q) > 1$ , 从  $c$  中删掉第  $t$  个元素得到  $c'$ , 从  $W_\alpha$  和  $W_\gamma$  中分别删去第  $a$  列和第  $b$  列得到  $W'_\alpha$  和  $W'_\gamma$ 。

若  $1 < Degree(W_\alpha, t) \leq Degree(W_\gamma, t)$ , 用子序列  $z = [\lambda, \lambda, \dots, \lambda]$  替代  $c$  的第  $t$  个元素, 得到  $c'$ 。这里的  $\lambda = c(t)$ ,  $z$  的长度是  $Degree(W_\alpha, t)$ , 然后用从子序列  $z$  到  $\mathcal{M}(c, x, W_\alpha, t)$  一一对应替换掉  $[t, p]^T$ , 得到  $W'_\alpha$ 。相似地, 用从子序列  $z$  到  $\mathcal{M}(c, x, W_\gamma, t)$  一一对应替换掉  $[t, q]^T$ , 得到  $W'_\gamma$ 。

注意到以上两种情况下得到的  $c', W'_\alpha$  和  $W'_\gamma$ , 相应的距离和都没有增加, 但是“合法的”的匹配都已经被删除或者替换, 递归地使用上面的方法, 即可满足该定理的条件。

证毕。

#### 4 最小 SSD 距离算法

本节提出了利用最小 SSD 距离计算两条时间序列的 DTW 中心的算法 DPSSD。对于给定的长度为  $m$  的时间序列  $x$  和长度为  $n$  的时间序列  $y$ , 引入了动态规划的方法来计算二者的 DTW 中心以及相应的距离和, 算法的主要步骤描述如下。

对于  $i=1, \dots, m$  和  $j=1, \dots, n$ , 递归计算  $x(1:i)$  和  $y(1:j)$  的 DTW 中心以及相应的 DTW 距离和。对于每一个  $i$  和  $j$ , 其 DTW 距离和的计算是以下两部分相加的和。

a) 第一部分, 考虑从  $c$  的最后一项到  $x(1:i)$  和  $y(1:j)$  的所有可能的匹配, 由定理 2 可知,  $x(1:i)$  和  $y(1:j)$  的平均值正是式(6)的解。注意到, 对于 DTW 中心, 由定理 3 可知, 其匹

配关系在搜索空间中被剪枝了。

b) 第二部分, 除去第一部分中的匹配关系, 我们得到了一个由更小的  $i$  和  $j$  组成的递归子问题。

$$Dist_{ij} = \begin{cases} 0, & \text{if } i=0 \text{ and } j=0 \\ \infty, & \text{else if } i=0 \text{ or } j=0 \\ \min \begin{cases} \min_{q \in \{1, \dots, j\}} Dist_{i-1, q-1} + err_{p,i}^{q,j} \\ \min_{q \in \{1, \dots, j\}} Dist_{i, q-1} + err_{p,i}^{q,j} \\ \min_{p \in \{1, \dots, i\}} Dist_{p-1, j-1} + err_{p,i}^{q,j} \\ \min_{p \in \{1, \dots, i\}} Dist_{p-1, j} + err_{p,i}^{q,j} \end{cases}, & \text{otherwise} \end{cases} \quad (9)$$

$$Cent_{ij} = \begin{cases} \emptyset, & \text{if } i=0 \text{ or } j=0 \\ Cent_{p,q} \oplus avg_{p,i}^{q,j}, & \text{otherwise} \end{cases} \quad (10)$$

对于给定的长度为  $m$  的时间序列  $x$  和长度为  $n$  的时间序列  $y$ , 对于  $i=1, \dots, m$  和  $j=1, \dots, n$ , 用  $Cent_{ij}$  表示  $x(1:i)$  和  $y(1:j)$  的 DTW 中心,  $Dist_{ij}$  表示  $DTW(Cent_{ij}, x[1:i])$  和  $DTW(Cent_{ij}, y[1:j])$  的和。式(9)和式(10)分别给出了  $Dist_{ij}$  和  $Cent_{ij}$  的递归关系。

这里  $err_{p,i}^{q,j} = err(x(ia:ib), y(ja:jb))$ ,  $avg_{p,i}^{q,j} = avg(x(ia:ib), y(ja:jb))$ ;  $p, q$  满足  $Dist_{p,q} + err_{p,i}^{q,j} = Dist_{ij}$  ( $p \in \{1, \dots, i\}, q \in \{1, \dots, j\}$ ); 符号“ $\oplus$ ”是用于连接时间序列和一个数的操作符。举个例子,  $[1, 2, 3, 4] \oplus 5 = [1, 2, 3, 4, 5]$ 。图 2 演示了这种递归关系。

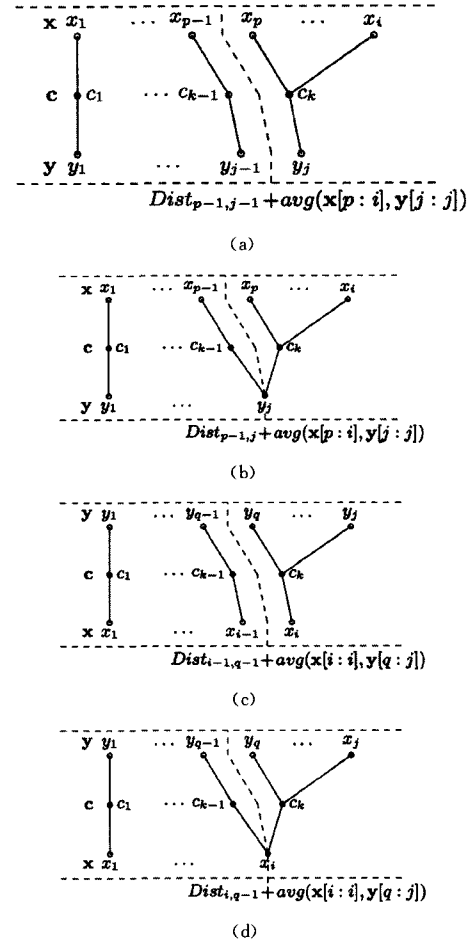


图 2 递归匹配关系示例

## 算法 1 DPSSD 算法

Input: m-length time series  $x$  and n-length  $y$

Ensure: Cent and Dist.

```

1. let  $M$  be a  $(m+1) \times (n+1)$ 
   matrix of tuples  $(ip, jp, ia, ie, error, length)$ 
2. for  $i=0$  to  $m, j=0$  to  $n$  do
3.    $M(i, j) = (0, 0, 0, 0, \infty, 0)$ 
4. end for
5.  $M(0, 0) = (0, 0, 0, 0, 0, 0)$ 
6. for  $i=1$  to  $m, j=1$  to  $n$  do
7.   for  $p=i$  downto 1 do
8.     update Matrix( $i, j, p-1, j-1, p, j$ )
9.     update Matrix( $i, j, p-1, j, p, j$ )
10.  end for
11.  for  $q=j$  downto 1 do
12.    update Matrix( $i, j, i-1, q-1, j-1, i, q$ )
13.    update Matrix( $i, j, i, q-1, i, q$ )
14.  end for
15. end for
16. Dist= $M(m, n)$ , error
17. Let  $(i, j) = (m, n)$ 
18. while  $p > 0$  and  $q > 0$  do
19.   Let  $(i', j', ia, ja, error, k) = M(i, j)$ 
20.   Let Cent[k]=avg( $x(ia:i), y(ja:j)$ )
21.   Let  $(i, j) = (i', j')$ 
22. end while
23. return Cent, Dist
24. function update Matrix( $i, j, ip, jp, ia, ja$ )
25. error= $M(ip, jp)$ . error+err( $x(ia:i), y(ja:j)$ )
26. if error< $M(i, j)$ . error
27.    $M(i, j) = (ip, jp, ia, ja, error, M(ip, jp). length+1)$ 
28. end if
29. end function

```

DPSSD 算法过程如算法 1 所示。算法的第一行定义了匹配误差矩阵  $M$ ,  $M(i, j) = (ip, jp, ia, ja, error, length)$ ,  $length$  表示  $x(1:i)$  和  $y(1:j)$  中心的长度, 且有递归关系  $M(i, j). error = M(ip, jp). error + avg(x(ia:i), y(ja:j))$ 。如图 2 所示,  $(ip, jp, ia, ja)$  的值由 4 个可能的分支  $(p-1, j-1, p, j)$ 、 $(p-1, j, p, j)$ 、 $(i-1, q-1, i, q)$  以及  $(i, q-1, i, q)$  得到, 分别对应着算法 1 中的第 8、9、12 和 13 行。这个动态规划的过程是为了得到最小的误差, 且误差  $M(i, j)$  即是从中心到  $x(1:i)$  和  $y(1:j)$  的距离之和。

从算法 1 中的第 6、7 和 11 行可以得出, 算法的时间复杂度是  $O(n^3)$ 。

## 5 实验

实验使用的数据集来自 UCR 的聚类/分类数据集“dataset1”中的训练集合<sup>[12]</sup>, 共 20 个。每一个数据集中的时间序列都有着相同的长度且属于不同的内部分类, 这些时间序列的长度从 60 到 637 不等, 数据集的大小从 14 到 1000 不等。实验内容是计算不同的算法在求解给定的两个时间序列的 DTW 中心的效果, 实验中的每一个时间序列  $x[x_1, \dots, x_n]$  都

被归一化为  $y[y_1, \dots, y_n]$ , 对于  $i=1, \dots, n$ , 满足  $y_i = \frac{x_i - \bar{x}}{\delta}$ ,

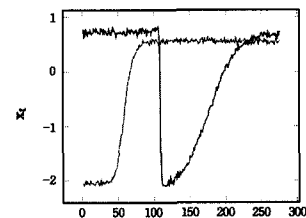
这里的  $\bar{x}$  是  $x$  的平均数,  $\delta$  是标准差, 实验的效果采用 SSD 来衡量。

表 1 DBA 和 DPSSD 算法的结果比较

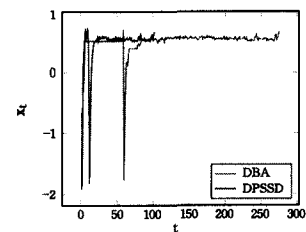
| Dataset           | Length | Size | DBA   | DPSSD | SSD ratio |
|-------------------|--------|------|-------|-------|-----------|
| 50words           | 270    | 450  | 3.72  | 2.73  | 73.39%    |
| Adiac             | 176    | 390  | 0.17  | 0.11  | 64.71%    |
| Beef              | 470    | 30   | 35.03 | 24.24 | 69.20%    |
| CBF               | 128    | 30   | 17.42 | 4.73  | 27.15%    |
| Coffee            | 286    | 28   | 0.31  | 0.22  | 70.97%    |
| EKG200            | 96     | 100  | 14.44 | 11.60 | 80.33%    |
| FaceAll           | 560    | 14   | 6.87  | 3.03  | 44.10%    |
| FaceFour          | 350    | 24   | 44.92 | 31.12 | 69.28%    |
| FISH              | 463    | 175  | 0.36  | 0.16  | 44.44%    |
| Gun_Point         | 150    | 50   | 0.08  | 0.06  | 75.00%    |
| Lighting2         | 637    | 60   | 32.47 | 20.19 | 62.18%    |
| Lighting7         | 319    | 70   | 12.95 | 8.73  | 67.41%    |
| OliveOil          | 570    | 30   | 0.02  | 0.01  | 50.00%    |
| OSULeaf           | 427    | 200  | 21.04 | 9.12  | 43.37%    |
| SwedishLeaf       | 128    | 500  | 2.47  | 1.83  | 74.09%    |
| synthetic_control | 60     | 300  | 19.60 | 10.83 | 55.26%    |
| Trace             | 275    | 100  | 20.09 | 7.36  | 36.64%    |
| Two               | 128    | 1000 | 43.25 | 10.03 | 23.19%    |
| wafer             | 152    | 1000 | 92.03 | 15.01 | 16.31%    |
| yoga              | 426    | 300  | 8.54  | 3.40  | 39.81%    |

表 1 展示了分别采用 DPSSD 算法和 DBA 算法得到的 DTW 中心的结果。第一列是数据集的名字, 第二列是数据集中时间序列的长度, 第三列是数据集的大小, 第四列是 DBA 算法的结果, 第五列是 DPSSD 算法的结果, 第六列是第五列和第四列的比值。DBA 算法对初始化数据敏感, 实验中使用了不同的初始化数据, 表中记录的是较小的结果。由于 DPSSD 算法可以在理论上证明能得到最优解, 因此对于不同维度和大小的数据集, 其效果明显好于 DBA。

图 3 展示了一个使用 DPSSD 算法和 DBA 计算两条样本时间序列 DTW 中心的例子。图 3(a) 给出了“Trace”数据集的第二条和第三条时间序列, 图 3(b) 则说明了两种算法的结果, 该实验使用了第三条时间序列作为 DBA 算法递归步骤的初始化数据。本文算法的结果是理论上全局最小的结果, 其长度短于原始序列。



(a) 时间序列数据



(b) DBA 与 DPSSD 结果对比

图 3 DBA 与 DPSSD 算法得到的 DTW 中心序列

总体上,对于给定的时间序列,DPSSD 算法可以获得理论上的全局最优解,而 DBA 算法则常常落入局部最优解。

**结束语** 本文提出了基于最小化中心序列到样本序列的 DTW 距离平方和来求解两条序列的中心序列的动态规划算法 DPSSD。本文通过引入匹配度理论,在理论上证明了该算法能够得到两条时间序列的 DTW 中心的最优解,同时还基于该理论,提出了相应的剪枝策略,从而降低了算法的时间复杂度。实验结果显示,应用该算法求得中心序列与两条样本序列的 DTW 距离和小于 DBA 的解,序列长度也小于 DBA 的解,同时保留了时间序列的形状特征,算法在鲁棒性上也优于 DBA。

在下一步的研究中,我们将进一步讨论如何求解多条时间序列的 DTW 中心,并将我们的研究用于生物、金融和图像分析等领域。提出的算法也可以被用来对大脑认知时间序列进行动态时变分析。

## 参考文献

- [1] Anami B S, Pagi V B. Acoustic signal-based approach for fault detection in motorcycles using chaincode of the pseudospectrum and dynamic time warping classifier[J]. IET Intelligent Transport Systems, 2014, 8(1): 21-27
- [2] Trajcevski G, Gunopulos D, Aggarwal C C. Time-series data clustering[M]. Aggarwal C C, Reddy C K, eds. Data Clustering: Algorithms and Applications, CRC Press, 2013: 357-375
- [3] Hu B, Rakthanmanon T, Hao Y, et al. Using the minimum description length to discover the intrinsic cardinality and dimensionality of time series[J]. Data Mining and Knowledge Discovery, 2015, 29(2): 358-399
- [4] Hautamaki V, Nykanen P, Franti P. Time-series clustering by

approximate prototypes[C]//Proceedings of International Conference on Pattern Recognition. IEEE, 2008: 1-4

- [5] Berndt D J, Clifford J. Using dynamic time warping to find patterns in time series[J]. KDD Workshop, ser. Seattle, WA, 1994, 10(16): 359-370
- [6] 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): 157-159  
Xie F D, Li Y, Sun Y, et al. Cluster Algorithm for Time Series Based on Key Points[J]. Computer Science, 2012, 39(3): 157-159
- [7] 郭崇慧, 苏木亚. 基于独立成分分析的时间序列谱聚类方法[J]. 系统工程理论与实践, 2011, 31(10): 1921-1931  
Guo Chong-hui, Su Mu-ya. Spectral clustering method based on independent component analysis for time series[J]. Systems Engineering Theory & Practice, 2011, 31(10): 1921-1931
- [8] Gupta L, Molfese D, Tammana R, et al. Nonlinear alignment and averaging for estimating the evoked potential[J]. IEEE Transactions on Biomedical Engineering, 1996, 43(4): 348-356
- [9] Salvador S, Chan P. Toward accurate dynamic time warping in linear time and space[J]. Intelligent Data Analysis, 2007, 11(5): 561-580
- [10] Niennattrakul V, Ratanamahatana C. Shape averaging under time warping[C]//Proceedings of International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology. IEEE, 2009: 626-629
- [11] Petitjean F, Ketterlin A, Gançarski P. A global averaging method for dynamic time warping, with applications to clustering[J]. Pattern Recognition, 2011, 44(3): 678-693
- [12] Keogh E, Folias T. The UCR time series data mining archive [OL]. [http://www.cs.ucr.edu/~eamonn/time series data](http://www.cs.ucr.edu/~eamonn/time%20series%20data)

(上接第 246 页)

## 参考文献

- [1] Sabelfeld A, Sands D. Declassification: dimensions and principles [J]. Journal of Computer Security, 2009, 17(5): 517-548
- [2] Sabelfeld A, Russo A. Securing interaction between threads and the scheduler [C]//19th IEEE Computer Security Foundations Workshop. 2006: 177-189
- [3] Sabelfeld A. The impact of synchronisation on secure information flow in concurrent programs [M]//Perspectives of System Informatics, LNCS 2244. 2001: 225-239
- [4] Smith G, Volpano D. Secure information flow in a multi-threaded imperative language [C]//25th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages. 1998: 355-364
- [5] Mantel H, Reinhard A. Controlling the what and where of declassification in language-based security [M]//Programming Languages and Systems, LNCS 4421. 2007: 141-156
- [6] Lux A, Mantel H, Perner M. Scheduler-Independent Declassification [M]//Mathematics of Program Construction, LNCS 7342. 2012: 25-47
- [7] 姜勋, 陈健, 平玲娣, 等. 多线程程序的信息抹除和降密安全策略[J]. 浙江大学学报(工学版), 2010(5): 854-862

Jiang L, Chen J, Ping L D, et al. Security policy for information erasing and leaking in multithreaded codes[J]. Journal of Zhejiang University(Engineering Science), 2010(5): 854-862

- [8] Focardi R, Gorrieri R, Martinelli F. Non Interference for the Analysis of Cryptographic Protocols[M]//Automata Languages and Programming, LNCS 1853. 2000: 354-372
- [9] Russo A, Sabelfeld A. Securing interaction between threads and the scheduler [C]//19th IEEE Computer Security Foundations Workshop. 2006: 177-189
- [10] 朱浩, 庄毅, 薛羽, 等. 基于内容和地点维度的机密信息降级策略[J]. 计算机科学, 2012, 39(8): 153-157  
Zhu H, Zhuang Y, Xue Y, et al. Declassification Policy Based on Content and Location Dimensions[J]. Computer Science, 2012, 39(8): 153-157
- [11] Askarov A, Myers A. A semantic framework for declassification and endorsement[M]//Programming Languages and Systems, LNCS 6012. 2010: 64-84
- [12] 李沁, 袁志祥. 一种宽容的多线程程序内部时间信息流类型系统[J]. 计算机科学, 2014, 41(3): 163-168  
Li Qin, Yuan Zhi-xiang. Permissive Type System for Internal Timing Information Flow in Multi-thread Programs[J]. Computer Science, 2014, 41(3): 163-168