一种确定滑动窗口规模的边界距离算法

彭 成[1,2] 贺 婧[1] 池 昊[1]
(湖南工业大学计算机学院 湖南 株洲 412007)[1] (中南大学自动化学院 长沙 410083)[2]

摘 要 由于大多装备的原始测量数据采集信息量大、密度高,现有的时间序列滑动窗口的降维方法采用经验值确定窗口大小,无法最大限度地保留数据的重要信息点,并且计算复杂度高。为此,文中研究了实际应用中滑动窗口对时间序列相似性技术的影响,提出了一种确定滑动窗口初始规模的算法。该算法构建拟合度更高的上下边界曲线,将趋势加权引入 LB_Hust 距离计算方法中,从而降低了数学建模难度,提高了装备数据相似性聚类与状态评估的效率。

关键词 滑动窗口,LB_Hust 距离,数据挖掘
中图法分类号 TP301 **文献标识码** A

Boundary Distance Algorithm for Determining Sliding Window Size

PENG Cheng[1,2] HE Jing[1] CHI Hao[1]
(School of Computer, Hunan University of Technology, Zhuzhou, Hunan 412007, China)[1]
(School of Automation, Central South University, Changsha 410083, China)[2]

Abstract Due to a large amount of information and high density of the original measurement data collected by most equipment, the existing time series sliding window dimension reduction method uses the empirical value to determine the window size, which cannot retain important information points of the data to the utmost extent, and has high computational complexity. To this end, the influence of sliding window on time series similarity technology in practical applications was discussed, and an algorithm for determining the initial scale of sliding window was proposed. The upper and lower boundary curves with higher fitting degree are constructed, and the trend weighting is introduced into the LB_Hust distance calculation method, which reduces the difficulty of mathematical modeling and improves efficiency of equipment data similarity classification and state evaluation.

Keywords Sliding window, LB_Hust distance, Data mining

1 引言

时间序列是按时间顺序排列、随时间变化且彼此间相互关联的量,时间序列的时间间隔长度也可以是不确定的[1]。时间序列分析是概率统计、计算机等学科中应用型较强的分支之一,是数据分析的重要领域[2],如相似性查询、异常检测、聚类、状态评估、趋势分析等[3],分析后可发现时间序列中隐含的关联性等深层次关系,揭示事物运动、变化以及发展的内在规律,对人们把握规律、正确认识事物并据此做出科学决策具有重要的现实意义。时间序列挖掘是数据挖掘领域的十大挑战性研究问题之一,其关键步骤是将原始数据从高维特征空间映射到低维空间,以降低时间复杂度[4],而相似性聚类是时间序列挖掘中一个重要研究方向,很多其他的时间序列挖掘方法都是建立在它的基础之上。

2 国内外现状及分析

在时间序列研究领域,学者们提出的多种时间序列降维方法都与滑动窗口有着密切的联系,一般通过设计滑动窗口的大小来对时间序列进行分段,以降低计算复杂度。主要包括滑动点对聚集近似方法(Piecewise Aggregate Approch, PAA)[5],基于 PAA 改进的支持增量划分的滑动窗口(preserving the salient points, PIP)分段表示法[6],符号聚集近似(Symbolic Aggregate ApproXimation, SAX)[7]以及离散傅里叶变换(Discrete FourierTransform, DFT)[8],动态时间弯曲(Dynamic Time Warping, DTW)[9]等。在实际应用中采用上述方法也较普遍,文献[10-11]通过可穿戴惯性传感器得到 IMU 数据,使用固定滑动窗口大小对连续 IMU 样本提取信息,结果缺乏理论支持和实验证明。文献[12]通过多尺度滑动窗口方法对地铁车门的亚健康状态进行识别,仅设定了初始化窗口尺度,没有考虑多尺度时该如何进行确定,而多尺度问题对识别起着重要作用。文献[13]讨论了预测水文时间序列算法中的滑动窗口规模大小和参数设置,仍然是根据经验推导出最佳窗口宽度,应用场合有限,不具备普适性。

上述研究成果都是基于滑动窗口的初始规模按照经验判断来设定这一假设条件。在这种策略在实际操作中重现,极有可能无法有效地保留幅度偏差大、有价值的信息点,忽略滑动窗口规模对时间序列相似性实验带来的影响。因此,需要在已有的时间序列降维处理技术中,寻求一种确定滑动窗口

本文受国家自然科学基金面上项目(61871432),湖南省自然科学基金青年项目(20173065)资助。
彭 成(1982—),男,博士,硕士生导师,主要研究领域为工业大数据处理;贺 婧(1989—),女,硕士生,主要研究领域为工业装备数据处理,E-mail:492508527@qq.com(通信作者)。

规模的方法,有效地保持数据不失真的同时降低计算复杂度。

本文首先阐述了相关定义,其次在分析现有算法不足的基础上,提出了一种确定滑动窗口初始规模的算法,并将权重引入 LB_Hust 距离算法中,使时间序列进行相似度划分时具有更高的精确性(本文算法流程图如图 1 所示),最后,从多个方面进行验证,对实验结果进行分析和总结。

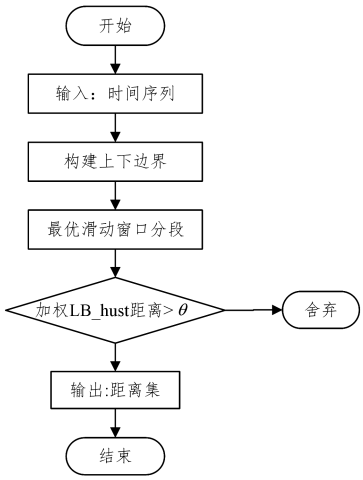


图 1　本文算法的流程

3　相关定义

定义 1(时间序列)　任意时刻,每个滑动窗口中的一组测量数据可以表示成序列[14],记为 $S_n=(S_1=(V_1,T_1),(S_2=(V_2,T_2),\cdots,(S_i=(V_i,T_i))$,其中点 $S_i=(V_i,T_i)$ 表示 T_i 时刻对应的属性数据值 V_i,记录时间 $T_i<T_i+1$。

定义 2(滑动窗口)　一个长度为 s 且含有 n 个变量的时间序列的表达形式为 $S(t)=(S_1(t),S_2(t),\cdots,S_n(t)),1\leqslant t\leqslant s$,其变量 i 的第 j 个长度为 w 的滑动窗口为 $s(i,j)=(x(i,j),x(i,(j+w)),\cdots,x(i,(j+w-1)))$,总共可以得到 $(s-w+1)$ 个子序列[15]。

定义 3(DTW)　DTW 是在时间轴方向上允许时间序列向其弯曲的距离[16]。假设两条长度分别 m 和 n 的时间序列 q 和 p,$q[1:m]=\{q_1,q_2,\cdots,q_m\}$,$p[1:n]=\{p_1,p_2,\cdots,p_n\}$,两者之间的距离可构造一个大小为 $m*n$ 的矩阵,$D(q,p)$ 为最佳匹配路径所对应的匹配距离。

定义 4(LB_Hust 距离)　LB_Hust 距离是基于 DTW 距离发展起来的,是一种对称的下界距离[17]。如时间序列 A、B,其中 $D(A,B)=D(B,A)$。

定义 5(提早停止策略)　两时间序列之间的累加距离一般是递进式增加的,对时间序列 A 和 B,计算他们的子序列 A' 和 B'(其中 $A'<A;B'<B$)时,若 $D_{\text{lb\_hust}}(A',B')>\theta$,此时可提前停止计算,减少 CPU 的运行时间。若 $D_{\text{lb\_hust}}(S,Q)\leqslant\theta$,则将该序列加入集合中,继续运算[18]。

4　确定滑动窗口初始规模的算法

4.1　滑动窗口降维算法

滑动窗口方法,是对时间序列进行分割,向后迭代滑动 r 个数据值,组成 $(s-w+1)$ 个长度为 w 的子序列,连续滑动 $(s-w)/r$ 次,形成等步长等窗口大小的时间区间段。该方法既便于理解规则又计算简单,易于应用到现实实际问题中,具有相当大的优势,滑动窗口的原理如图 2 所示。

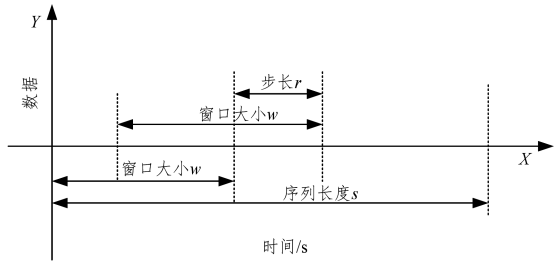


图 2　滑动窗口原理

若时间序列 A 在一定时间范围内波动小、较平缓,则此区间可被判断为滑动窗口分割点;若波动剧烈,则应极大地保留时序片段中的重要数据信息,此区间不应为滑动窗口分割点[20]。使滑动窗口规模 $window=[2:length(A)/2]$,步长 $Steplength=[0:window]$,设定窗口分段内最大值与最小值的差值为 β。若窗口内最大值与最小值的差值小于阈值 β,即 $|S_{\max}-S_{\min}|<\beta$,则该点可以是分割点,若差值大于阈值 β,即 $|S_{\max}-S_{\min}|>\beta$,则该点不可以是分割点,将其加入到 $DifferentArea$ 集合中。判断 $DifferentArea$ 集合与滑动窗口是否有并集,若有,则表示该窗口规模不好,结束运算。若无并集,则将当前窗口加入到 W 集合、将当前步长 $Steplength$ 加入到 S 集合。最终,选取集合中最大窗口 $window$ 与 $Steplength$。算法 1 描述如下。

算法 1　Function Window

```
For Window=2:Len(A)/2
  For Steplength=0:Window
    I=1;J=Window
    I=1+Steplength
    J=1+ Window+Steplength
    swArea=[I,J]
    DifferentArea=(Aamax-Aamin>β)
    IF swArea∩DifferentArea>0
    End
    IF swArea∩DifferentArea≤0
    Window=Window +1
  End
End
Window=max(Window)
Steplength=min(Steplength)
```

4.2　边界算法

DTW 算法给定了两个时间序列:标准参考序列 $A=\{a_1,a_2,\cdots,a_n\}$,测试模板序列 $B=\{b_1,b_2,\cdots,b_m\}$,每个分量可以是一个数或者是一个更小的向量。序列 A 和 B 的弯曲路径为:$W=\{w_1,w_2,\cdots,w_k\}$,$\max(m,n)\leqslant k\leqslant m+n-1$,满足式(1):

$$DTW(A,B)=\min\left(\sqrt{\sum_{k=1}^{k}wk}\right) \tag{1}$$

可采用公式递归求解 DTW 距离,如式(2)所示:

$$P(a_1,b_1)=D(a_1,b_1)$$

$$P(a_i,b_j)=D(a_i,b_j)+\min(P(a_i-1,b_j),P(a_i-1,b_j-1),P(a_i,b_j-1))$$

$$P(a_i,b_j)=(a_i-b_j)^2 \tag{2}$$

基于 DTW 算法思想[19],Kim 提出了一元查询策略,以计算最大值、最小值、起始值和结束值 4 种特征向量,构建对应特征的欧氏距离索引。Yi 将两组序列的其中一组数据的

极大值、极小值与另一组时序的数据的和构成时序间距离。Keogh 对时序间的 DTW 距离计算方法进行了观察，引入了弯曲路径限制 w，定义了上下边界 U 和 L，第 i 个区间的上下边界如式(3)所示：

$$U_i=\mathrm{MAX}(X_{i-w},X_{i+w})$$
$$L_i=\mathrm{MIN}(X_{i-w},X_{i+w}) \tag{3}$$

在图形上用形如带状的区间将时序涵盖在内，如图 3 所示。

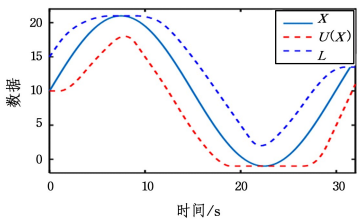


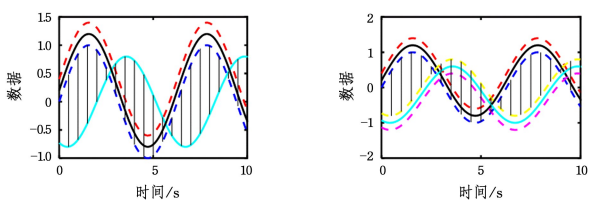
图 3 上下边界示意图

区间宽度为 U_i-L_i，而在不同的情况下该宽度是不同的，一定时间内幅度差值大的，宽度较宽，幅度表现平缓的，宽度较窄。长度为 n 的两个时间序列 A 和 B，对于 A 中的 a_i 及 B 中的 b_j，需满足 $j-w\leqslant i\leqslant j+w$，及 $D_{\mathrm{LB\_Hust}}(A,B)\leqslant D_{\mathrm{DTW}}(A,B)$，下界距离 $D_{\mathrm{LB\_Keogh}}(A,B)$ 的定义如式(4)所示：

$$D_{\mathrm{LB\_Keogh}}(A,B)=\sqrt{\sum_{i=1}^{n}\begin{cases}(B_i-U_i)^2, & \text{if } B_i<L\\(B_i-L_i)^2, & \text{if } B_i<L\\0, & \text{otherwise}\end{cases}} \tag{4}$$

明显发现，LB_Keogh 距离[20]不具有对称性，$LB\_Keogh(A,B)\neq LB\_Keogh(B,A)$。不对称性使得计算次数与复杂度增加，不便于聚类分析，如 4(a)所示。在 LB_Keogh 距离算法基础上发展起来的 LB_Hust 距离计算方法满足了对对称性的要求，且 LB_Hust 距离计算方法的时间复杂度为 $O(n)$，比 DTW 距离方法的时间复杂度 $O(nm)$好。其长度为 n 的两个时间序列 A 和 B，对于 A 中的 a_i 及 B 中的 b_j，需满足 $j-w\leqslant i\leqslant j+w$，及 $D_{\mathrm{LB\_Bust}}(A,B)\leqslant D_{\mathrm{LB\_Keogh}}(A,B)$，$A$ 和 B 对称性要求需满足 $D_{\mathrm{LB\_Bust}}(A,B)=D_{\mathrm{LB\_Keogh}}(B,A)$，其下界距离如图 4(b)所示，定义如式(5)所示：

$$LB\_Hust(A,B)=\sqrt{\sum_{i=1}^{n}\begin{cases}(L_{ai}-U_{bi})^2, & \text{if } L_{ai}>U_{bi}\\(L_{bi}-U_{ai})^2, & \text{if } L_{bi}<U_{ai}\\0, & \text{otherwise}\end{cases}} \tag{5}$$



(a)LB_Keogh 示意图 (b)LB_Hust 示意图

图 4 边界距离示意图

由图 5 不难发现 LB_Hust 距离的不便利之处，时间序列 A 的上下边界 UL 索引中的下边界在图中表示为 L_A，而 U_B 和 U_C 分别表示了时序 B 和 C 的上下边界 UL 索引中的上边界序列。则时序 A 和时序 B 之间的距离为 X 区间与 Z 区间的距离之和，表示为 $D_{AB}=D_X+D_Z$。时序 A 和时序 C 之间的距离为 X 区间与 Y 区间的距离之和，表示为 $D_{AC}=D_X+D_Y$。时序 B 和时序 C 之间的距离为 Y 区间与 Z 区间的距离之和，表示为 $D_{BC}=D_Y+D_Z$。可直观感受到，$D_X>D_Z>D_Y$，Z 区间大于 Y 区间，这将导致我们推断时序 A 与时序 B 之间的距离大于时序 A 与时序 C 之间的距离，但就趋势走向来看，时序 A 与时序 B 之间的距离理应小于时序 A 与时序 C 之间的距离。

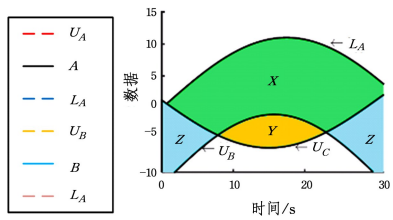


图 5 3 个序列之间的距离图

因此，在我们的普遍认知中，原始 LB_Hust 算法计算的距离不具备实际意义。将符号引入原始 LB_Hust 算法中，将两序列间的距离按趋势走向分为正趋势走向和负趋势走向。对序列间距离进行趋势走向和数值大小的双重判断，符号 LB_Hust 距离如式(6)所示：

$$D_{np}=w_n*D_n+w_p*D_p \tag{6}$$

其中，w_n 和 w_p 是负趋势走向距离 D_n 和正趋势轴向距离 D_p 的权重影响因子，权重系数满足 $w_n+w_p=1$，为了判断趋势走向的正负，引入 Symbol 确定因子，其计算式如(7)所示：

$$Symbol=(\max U_{ai}-\min U_{ai})*(\max U_{bi}-\min U_{bi}) \tag{7}$$

对于时间序列 A 的 i 元素，$\max U_{ai}$ 与 $\min L_{ai}$ 在时间区间 $[t_{i-w},t_{i+w}]$反映了曲线的趋势走向。当 $\max U_{ai}>\min L_{ai}$ 时，在时间区间 $[t_{i-w},t_{i+w}]$上的曲线呈上涨趋势；当 $\max U_{ai}<\min L_{ai}$ 时，在时间区间 $[t_{i-w},t_{i+w}]$上的曲线呈下跌趋势。

实现引入符号的加权 LB_Hust 距离的算法的具体过程如算法 2 所示。

算法 2 引入符号的加权 LB_Hust 距离的算法 Function=sy_LB_Hust

$D_{np}=0$
$D_n=0$
$D_p=0$
IF Len<length //判断长度
  len++
IF $U_{bi}>L_{ai}$
  $Dist=(U_{bi}-L_{ai})^2$
IF $U_{bi}<L_{ai}$
  $Dist=(U_{ai}-L_{bi})^2$//计算原始 LB_Hust 距离
$Symbol=(\max U_{ai}-\min L_{ai})*(\max U_{bi}-\min L_{bi})$ //引入 Symbol 确定因子
IF Symbol<0
  $D_n+=Dist$
//symbol<0，对负距离进行累加
IF Symbol>0
  $D_p+=Dist$
    //symbol>0，对正距离进行累加
  $D_{np}=w_n*D_n+w_p*D_p$
    //加入权重对距离进行累加
End

4.3 算法实现步骤

Step1 对时间序列 A 和 B 进行归一化处理，使得曲线取得较好的相似性比较效果。

Step2　运用 LB_Hust 算法对时间序列 A 和 B 构造上下边界曲线 U_1 与 L_1，U_2 与 L_2。

Step3　使用滑动窗口方法对时间序列进行降维，将长度为 N 的时间序列 A 和 B 用规模为 w 的窗口滑动，遍历时间序列 A 和 B，分别得到 x，x_2 个子序列，从而得到相应的 $\overline{S}$，$\overline{U}_1$，$\overline{L}_1$，$\overline{T}$，$\overline{U}_2$，$\overline{L}_2$。

Step4　运用加权 LB_Hust 距离公式对处理后的时间序列进行计算并求得边界距离 $D_{\text{LB\_Hust}}(A,B)$。

Step5　$D_{\text{LB\_Hust}}(A,B)$ 为边界距离度量值，θ 为设定的经验阈值，在运算过程中进行两两比较。若 $D_{\text{LB\_Hust}}(A,B)>\theta$，则说明两时序并不相似，停止计算；若 $D_{\text{LB\_Hust}}(A,B)\leqslant\theta$，则将该子序列加入集合 R 中。

5　实验分析

实验仿真环境：Win 7 Intel(R)Core(TM) i3-7100 CPU @3.90 GHz，Memory 4.00 GB，Matlab 2014b。

5.1　实验数据

为了确保实验的公正性与可靠性，选用的数据集是 UCI 学校的 KDD archive 文档中的一个实例综合控制时间序列。该数据集由 Alcock 和 Manolopoulos 通过工业过程综合生成的 600 个控制图的例子组成，其中有 6 种不同类型的时间序列，1－6 类特点状态分别为正常、循环、上升趋势、下行趋势、向上移动、向下移动，并且每一类有 100 条序列，每条序列包括 60 个数据[21]。其时间序列的属性如表 1 所列，600 条时间序列的归一化样本状态如图 6 所示。其归一化定义如式(8)所示：

$$y=\frac{(y\ \max-y\ \min)*(x-x\ \min)}{x\ \max-x\ \min}+y\ \min \tag{8}$$

表 1　数据集属性

类型	序列	状态
1	1～100	正常
2	101～200	循环
3	201～300	上升趋势
4	301～400	下行趋势
5	401～500	向上移动
6	501～600	向下移动

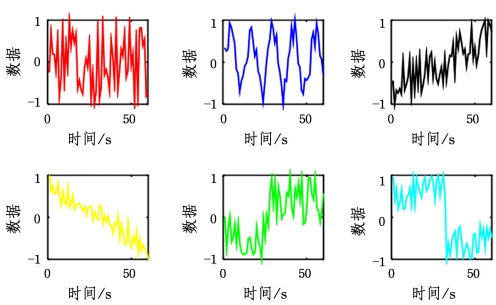
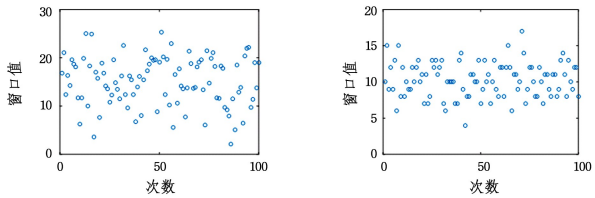


图 6　1－6 类时间序列的状态特点图

实验 1　确定滑动窗口的规模范围

在 synthetic control time series 6 类序列中，将随机选取 6 类中的 7 条时间序列(分别命名为 S_{11}，S_{12}，S_2，S_3，S_4，S_5，S_6，其中 S_{11} 与 S_{12} 来自同一类别)作为实验对象。

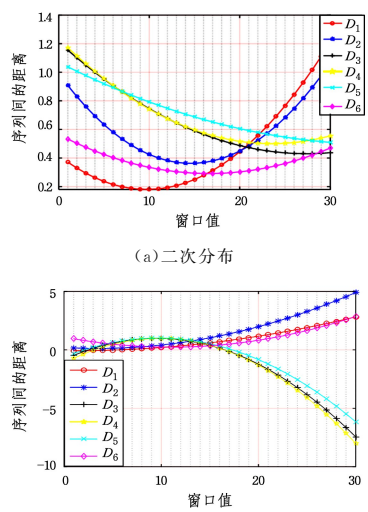
采用上文提出的滑动窗口降维算法对 7 条时间序列进行运算，得出最优窗口为 8，步长为 2。为了证明算法的可行性，运用最小二乘法对 7 条时序间的距离(S_{11} 依次与剩余序列进行距离计算，得到 $D_1(S_{11},S_{12})$，$D_2(S_{11},S_2)$，$D_3(S_{11},S_3)$，$D_4(S_{11},S_4)$，$D_5(S_{11},S_5)$，$D_6(S_{11},S_6)$进行拟合。其中滑动窗口 w 的取值采用两种不同的随机方法，分别为二次分布与泊松分布随机取值方法，如图 7(a)和图 7(b)所示。



(a)二次分布　　(b)泊松分布

图 7　二次分布与泊松分布产生的随机窗口值

二次分布与泊松分布的距离偏离度拟合曲线如图 8(a)和图 8(b)所示。其中二次分布最小二乘法拟合图比泊松分布更能清楚地展现 7 条序列之间的距离关系。当滑动窗口在 1－5 之间时，虽然 D_1 距离与 D_2，D_3，D_4，D_5，D_6 距离相隔较远，但同处一类的 S_{11} 和 S_{12} 序列之间的 D_1 距离值也较大，因此滑动窗口规模不可取于 1～5 间；当滑动窗口在 11～30 之间时，D_1 距离值较大且与 D_2，D_3，D_4，D_5，D_6 距离相隔较小，且存在相交的现象。因此滑动窗口规模不可取在 11－30 之间；当滑动窗口在 6～10 之间时，D_1 距离值较小且与 D_2，D_3，D_4，D_5，D_6 距离相隔较远，在图 8 中，大致也能看到当 $w=8$ 时最能满足上述要求，$w=8$ 为最优滑动窗口。因此本文提出的滑动窗口降维算法是可行、有效的。



(a)二次分布

(b)泊松分布

图 8　距离偏离度拟合曲线

实验 2　优化滑动 LB_Hust 距离权重系数

在确定滑动窗口规模的基础上，权重系数在滑动 LB_Hust 距离中也占有重要作用。w_n 和 w_p 的选择要根据实际时间序列的相似性比较来确定，它决定了哪个分量在 LB_Hust 距离的运算中产生较大的影响。w_n 的大小决定了负距离的影响大小，w_p 的大小决定了正距离的影响大小。

为了评估本文方法的有效性，我们首先生成如图 9 所示的 3 条序列。其中序列 A 和序列 B 的趋势走向相差较大，序列 A 和序列 C 的趋势走向相差较小，只存在相位差。将 3 条时间序列运用原始 LB_Hust 距离运算和加权的 LB_Hust 距离运算进行聚类。

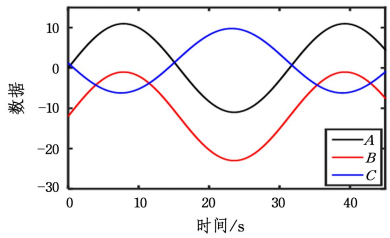


图 9 趋势相同(A,B)和相反(C)的时间序列比较图

根据 3 条自定义数据,原始 LB_Hust 距离与滑动加权 LB_Hust 距离在 w_n 和 w_p 取值不同的情况下聚类的结果如图 10 所示。图中"*"表示 A 曲线,"○"为 B 曲线,"□"为 C 曲线。当 $K=2$,$w=4$ 时,原始 LB_Hust 距离的聚类结果如图 10(a)所示,A 曲线与 B 曲线为一类,C 曲线是另一类。当 $w_n=w_p=0.5$ 时,加权 LB_Hust 距离的聚类结果如图 10(b)所示,与原始 LB_Hust 距离的聚类结果无差,可得出原始LB_Hust 距离等同于 $w_n=w_p=0.5$ 时的加权 LB_Hust 距离。当 $w_n=0.4$,$w_p=0.6$ 时,加权 LB_Hust 距离的聚类结果如图 12 (c)所示,A 曲线与 C 曲线为一类,B 曲线为另一类。根据分为 6 类的 UCI 的 KDD archive 数据集,讨论加权 LB_Hust 距离方法的不同权重系数 w_n 和 w_p,得到在 1-NN 聚类器下不同权重系数的聚类错误率图(见图 11)。当 $w_n>w_p$ 时,加权 LB_Hust 距离算法在对数值差异运算的同时也着重对趋势走向差异进行了计算,当 $w_n=0.4$,$w_p=0.6$ 时聚类错误率较低,随着 w_n 的继续增加,聚类错误率增大,因此 w_n 的比重也不宜过高。

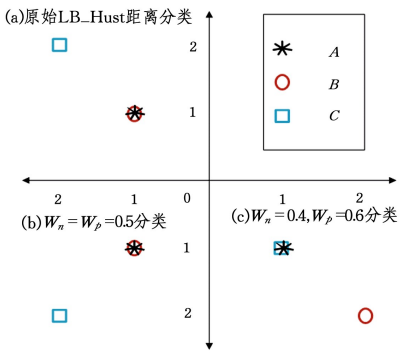


图 10 聚类结果的对比

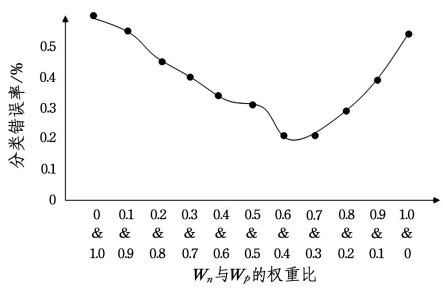


图 11 不同权重系数的聚类错误率

实验 3 确定滑动窗口规模的边界距离算法

本实验采用 UCI 的 5 组常用的真实数据集来对聚类算法进行测试,并用 I 指数、精确率和召回率[22]来评价不同窗口大小与不同权重因子下的距离方法的聚类效果。5 组数据分别是 IRIS 数据集、WINE 数据集、ECOLI 数据集、ZOO 数据集和 GLASS 数据集。其具体信息如表 2 所列。

表 2 各数据集的属性

数据集名称	样本数	类别数	属性数
IRIS	150	3	4
WINE	178	3	13
ECOLI	336	8	18
ZOO	101	7	18
GLASS	214	6	9

I 指标是根据选择类与类中心间距离的最大值来衡量类间分离度,同时使用类中各点与类中心的距离之和来衡量紧密度,$I(NC)$值越大,表示聚类质量越好,定义如式(9)所示:

$$I(NC)=\left[\frac{1}{NC}\frac{\sum_{x\in D}d(x,c)}{\sum_{i=1}^{NC}\sum_{x\in CC_i}d(x,C_i)}\max d(C_i,C_j)\right]^P \tag{9}$$

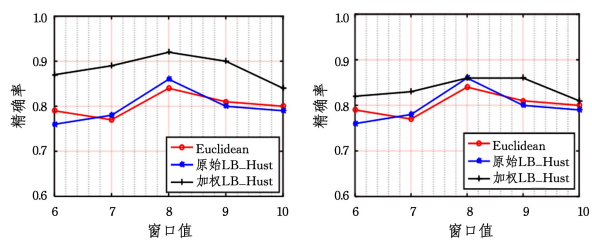
精确率(precision)反映了被聚类器判定的正例中真正的正例样本的比重,如式(10)所示:

$$P=TP/(TP+FP) \tag{10}$$

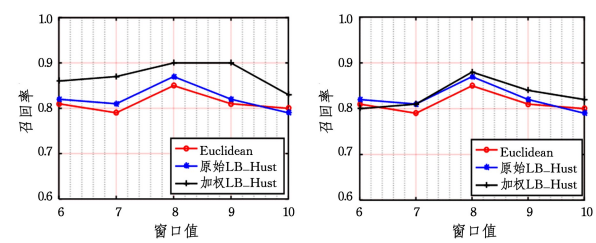
召回率(recall)反映了被正确判定的正例占总的正例的比重,如式(11)所示:

$$R=TP/(TP+FN)=1-FN/TP \tag{11}$$

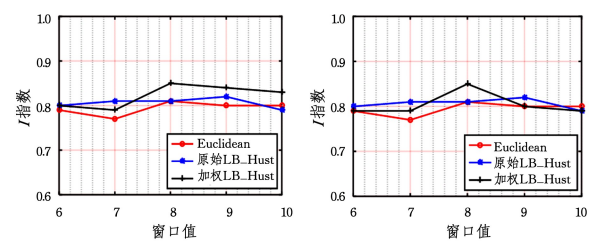
根据实验 2 得出:当 $w_n>w_p$ 时,加权 LB_Hust 边界距离效果要好,但不能完全只关注趋势影响,选取 $w_n=0.6$ 与 $w_n=0.7$ 做进一步实验。图 12 是 w_n 为 0.6 和 0.7 时 3 种距离计算方法在 IRIS 数据集上根据滑动窗口规模的不同,精确率(a)、召回率(b)和 I 指数(c)相应变化的示意图。当 $w_n=0.6$ 时,加权 LB_Hust 算法的精确率、召回率和 I 指数都远远高于其他两种距离算法,且 $w=8$ 时,达到最高;当趋势权重 w_n 从 0.6 增大为 0.7 时,由于趋势影响过大,加权的 LB_Hust 算法在召回率与 I 指数上等于甚至低于原始 LB_Hust 和 Euclidean 算法。图 13 是窗口规模 $w=8$,正负趋势因子 $w_n=0.6$,$w_p=0.4$ 时,3 种距离计算方法在 5 组数据集上的不同精确率。



(a)精确率



(b)召回率



(c)I 指数

图 12 3 种距离方法在 IRIS 上的效果

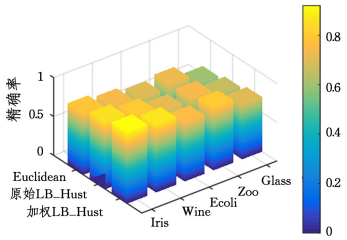


图 13　3 种距离计算方法的精确率对比

可以直观看到，加权 LB_Hust 算法在 5 组数据集上皆有较高的精确率，尤其是在 Iris 数据集上。总体而言，根据确定滑动窗口初始规模算法得出窗口大小 $w=8$，加权 LB_Hust 算法的正负趋势因子 $w_n=0.6$，$w_p=0.4$，该办法在 5 个数据集上的精确率分别为 0.92 0.87 0.75 0.80 0.70，其效果最好。

结束语　本文在传统算法的基础上，提出了一种确定滑动窗口规模的边界距离算法。该方法可根据时序的振幅特征确定滑动窗口初始规模，能有效地保留重要的数据点；并在 LB_Hust 算法中引入了权重因素，使其在时序相似聚类中具有更高的精度。实验结果表明，确定窗口初始规模的加权 LB_Hust 算法与原始 LB_Hust 算法相比，其聚类精确率、召回率和 I 指数都更好，可以更好地降低时间复杂度，提高相似性聚类效率。

参 考 文 献

[1] CHERNICK M R. Wavelet Methods for Time Series Analysis [J]. Technometrics, 2016, 43(4): 491-497.

[2] ANDREW B, KELVYN J. Explaining Fixed Effects: Random Effects Modeling of Time-Series Cross-Sectional and Panel Data* [J]. Political Science Research & Methods, 2015, 3(1): 133-153.

[3] BULLMORE E, LONG C, SUCKLING J, et al. Colored noise and computational inference in neurophysiological (fMRI) time series analysis: Resampling methods in time and wavelet domains[J]. Human Brain Mapping, 2015, 12(2): 61-78.

[4] 李正欣，张凤鸣，张晓丰，等. 多元时间序列特征降维方法研究[J]. 小型微型计算机系统，2013，34(2)：338-344.

[5] ADWAN S, ALSALEH I, MAJED R. A new approach for image stitching technique using Dynamic Time Warping (DTW) algorithm towards scoliosisX-ray diagnosis[J]. Measurement, 2016, 84: 32-46.

[6] CHEN T L, CHEN F Y. An intelligent pattern recognition model for supporting investment decisions in stock market[J]. Information Sciences, 2016, 346: 261-274.

[7] 刘芬，郭躬德. 基于符号化聚合近似的时间序列相似性复合度量方法[J]. 计算机应用，2013，33(1)：192-198.

[8] XIAO J, BAI L, LI F, et al. Sizing of Energy Storage and Diesel Generators in an Isolated Microgrid Using Discrete Fourier Transform (DFT)[J]. IEEE Transactions on Sustainable Energy, 2014, 5(3): 907-916.

[9] 李正欣，张凤鸣，李克武，等. 一种支持 DTW 距离的多元时间序列索引结构[J]. 软件学报，2014，25(3)：560-575.

[10] HU B, DIXON P C, JACOBS J V, et al. Machine learning algorithms based on signals from a single wearable inertial sensor can detect surface- and age-related differences in walking[J]. Journal of Biomechanics, 2018, 71: 37-42.

[11] YAO R, LIN G S, SHI Q F, et al. Efficient Dense Labelling of Human Activity Sequences from Wearables using Fully Convolutional Networks[J]. Pattern Recognition, 2017, 78: 252-266.

[12] 薛钰，梅雪，支有冉，等. 基于时间序列数据挖掘的地铁车门亚健康状态识别方法[J]. 计算机应用，2018，38(3)：905-910.

[13] 余宇峰，朱跃龙，万定生，等. 基于滑动窗口预测的水文时间序列异常检测[J]. 计算机应用，2014，34(8)：2217-2220.

[14] 李海峰，章宁，朱建明，等. 时间敏感数据流上的频繁项集挖掘算法[J]. 计算机学报. 2012，35(11)：2283-2293.

[15] LEE G, YUN U, RYU K H. Sliding window based weighted maximal frequent pattern mining over data streams[J]. Expert Systems with Applications, 2014, 41(2): 694-708.

[16] 陈树广，李俊奎，陈胜利. CSDTW：一种时间序列流上的受限动态弯曲距离[J]. 计算机应用研究，2012，29(8)：2939-2942.

[17] 李俊奎. 时间序列相似性问题研究[D]. 武汉：华中科技大学，2008.

[18] BIAN W, TAO D. Max-Min Distance Analysis by Using Sequential SDP Relaxation for Dimension Reduction[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2011, 33(5): 1037-1050.

[19] NIENNATTRAKUL V, RUENGRONGHIRUNYA P, RATANAMAHATANA C A. Exact indexing for massive time series databases under time warping distance[J]. Data Mining & Knowledge Discovery, 2010, 21(3): 509-541.

[20] KEOGH E, RATANAMAHATANA C A. Exact indexing of dynamic time warping[J]. Knowledge & Information Systems, 2005, 7(3): 358-386.

[21] KDD' Datasets. The UCI KDD Archive[Z]. 1999.

[22] KOU G, PENG Y, WANG G. Evaluation of clustering algorithms for financial risk analysis using MCDM methods[J]. Information Sciences, 2014, 275(11): 1-12.

(上接第 460 页)

[6] 何熊熊，管俊轶，叶宣佐，等. 一种基于密度和网格的簇心可确定聚类算法[J]. 控制与决策，2017，32(5)：913-919.

[7] 戴娇，张明新，郑金龙，等. 基于密度峰值的快速聚类算法优化[J]. 计算机工程与设计，2016，37(11)：2979-2984.

[8] 张素洁，赵怀慈. 最优聚类个数和初始聚类中心点选取算法研究[J]. 计算机应用研究，2017，34(6)：1617-1620.

[9] 夏庆亚. 基于密度峰值和网格的自动选定聚类中心算法[J]. 计算机科学，2017，44(11)：403-406.

[10] DU M J, DING S F, JIA H J. Study on density peaks clustering based on k-nearest neighbors and principal component analysis [J]. Knowledge-Based Systems, 2016, 99(2): 135-145.

[11] YANG W, WANG T, LI J D. Clustering parameter selection algorithm based on density for divisional clustering process[J]. Control andDecision, 2016, 31(1): 21-29.