基于形态距离及自适应权重的相似性度量

曹洋洋，林 意，王智博，鲍国强
（江南大学 数字媒体学院，江苏 无锡 214122）

摘 要：针对传统的动态时间弯曲算法的性能容易受到离群点以及局部噪声点的影响，同时对于复杂数据的处理能力较差，提出基于形态距离及自适应权重的相似性度量算法。该算法首先利用 l_1 趋势滤波对原始待比较序列进行降维、压缩；其次引入形态距离计算两时间序列的距离矩阵；最后利用自适应赋权的距离函数抽取出各个子序列所含的信息量差异并结合动态时间弯曲完成最终时间序列相似度量。实验表明，该算法有更强的鲁棒性，能够更好地利用序列的形态特征完成宏观的相似性度量，同时在处理复杂数据时更加精确、高效、稳定。

关键词：时间序列；相似性度量；动态时间弯曲；形态距离；自适应赋权
中图分类号：TP391.9 **文献标志码**：A **文章编号**：1001-3695(2018)09-2638-05
doi:10.3969/j.issn.1001-3695.2018.09.018

Similarity measure based on morphological distance and adaptive weights

Cao Yangyang, Lin Yi, Wang Zhibo, Bao Guoqiang
(*School of Digital & Media*, *Jiangnan University*, *Wuxi Jiangsu* 214122, *China*)

Abstract: The performance of the traditional dynamic time bending algorithm is susceptible to outliers and local noise points, and the processing capacity of complex data is poor. In this regard, this paper proposed a similarity measure based on morphological distance and adaptive weight. The algorithm first used the l_1 trend filter to reduce the dimension and compression of the original comparison sequence. Secondly, the algorithm introduced morphological distance to calculate the distance matrix of two time series. Finally, the algorithm used the distance function of adaptive weight to extract the difference of information contained in each sub-sequence and completed the final time series similarity measure with dynamic time bending. Experiments show that the algorithm has stronger robustness and can make better use of the morphological features of the sequence to complete the macro similarity measure, while dealing with complex data more accurate, efficient and stable.

Key words: time series; similarity measure; dynamic time bending; morphological distance; adaptive weight function

0 引言

时间序列是按时间顺序排列的一系列观察值，并广泛地存在于金融、医药、商业和工业等领域，几乎无处不在。其中时间序列的相似性度量是衡量两个时间序列相似程度的方法，也是时间序列进行分类[1]、聚类[2]、相似性搜索[3]、预测和异常值检测[4]等问题的基础和核心。近年来，距离度量序列间相似性得到研究人员的广泛关注，其中常见的距离度量方法包括欧氏距离[5]、动态时间弯曲距离[6]、编辑距离[7]、最长公共子序列[8]以及符号聚合近似距离[9]等。这些方法应用广泛，但是仍然存在许多局限性，如在两个时间序列长度相等的情况下使用的最常见的距离度量方式是欧氏距离，在处理股票序列的实际问题中，特别地，在一维股票数据中，采用欧氏距离的方法不能取得良好的学习效果，这是因为序列本身的形状反映了包括值轴和时间轴在内的趋势信息，而欧氏距离关注两个点之间的差异。

此外，欧氏距离要候选序列等长，序列各点一一对应，这也大大地限制了欧氏距离的使用范围。其他基于形状的距离度量方法，如基于离散符号[10]表示的度量方法，也同样需要两时间序列具有相等的长度，并且分割后的时间序列点一一对齐。针对不等长度或者时间粒度不相等的时间序列对，Berndt 等人[6]提出动态时间弯曲(DTW)算法，不仅能够度量不同长度时间序列的相似性，而且对时间轴的局部压缩、伸展和弯曲不敏感，可以进行错位匹配。但是 DTW 算法存在以下三点缺陷：a)DTW 算法具有高时间和空间复杂性，限制了其仅适用于小规模时间序列；b)DTW 算法针对序列点大小进行比较，没有考虑序列中前后点之间的联系，忽略了时间序列的整体形状；c)弯曲路径时间上的偏离会对序列相似度产生影响。为了减少时间开销和降低空间复杂度，Yu 等人[2]进一步提出了粒度动态时间规整(GDTW)，大大提高了计算效率。但是，GDTW 算法用等长粒度窗口分割原始时间序列，获得的分段序列不能反映任何粒度窗口的趋势信息。因此，GDTW 同样忽略了时间轴，也是一个单一的面向序列值的度量。

形态是时间序列的重要属性，形态距离度量是通过考虑待计算对象各维的差值具体分布因素进行相似度计算的一种方法，根据时间序列的形态进行相似度评估，主要有基于形态的自然语言描述如文献[11]提出的基于趋势的时间序列相似性度量方法，一定程度上提高了度量的效率，但该方法的模式和趋势划分有限、对应分段距离值是离散的，计算精度不高。基于符号聚合近似(SAX)方法，如文献[12]提出了基于形态特征的时间序列符号聚合近似的相似性度量方法 MINDIST，文献[13]对 MINDIST 作了改进并提出基于趋势的符号化表示的距离函数 TSX_DIST，取得了更好的度量效果，两种方法均利用各个子序列的均值特征和斜率特征共同描述数据的变化趋势，并将其符号化后进行相似性度量，兼顾了时间序列的统计和形态特征，但由于符号化和距离计算过程复杂，应用受到限制。基于某种几何性质的相似度方法，如弧度距离[14]通过相邻两点

收稿日期：2017-04-18；**修回日期**：2017-06-01

作者简介：曹洋洋(1991-)，女，硕士研究生，主要研究方向为数据挖掘、人工智能(136193805@qq.com)；林意(1960-)，男，副教授，硕导，主要研究方向为计算机图形学、微分几何学；王智博(1989-)，男，硕士研究生，主要研究方向为数据挖掘、计算机图形学；鲍国强(1992-)，男，硕士研究生，主要研究方向为模式识别、人工智能.

构成的直线与时间轴夹角的弧度进行相似度计算,部分反映时间序列的趋势变化,但形态信息挖掘不充分,完整性差,基于分段拟合曲线在各时刻处的曲率距离方法[15],实验效果较好但计算复杂、效率不高。

针对上述度量方法的不足,本文提出了两个一维时间序列之间精确的形状距离测度。在计算距离之前,首先,引入 l_1 趋势滤波将待比较的原始序列预处理,将原始时间序列映射到分段近似的空间,并且利用成熟的分段算法处理第一步的结果;其次,通过线性段的平移和连接构造一个趋势三角形,由于趋势三角形兼顾了时间轴和值轴信息,所以可以用三角形的面积作为线性段间的形态距离;再者,提出了自适应函数作为子序列的权重,减小分段后不同长度子序列信息的差异对相似性造成的影响;最后,在分段线性空间中采用动态时间弯曲计算两个给定时间序列之间的距离。

1　相关研究

本章简要描述了 l_1 趋势滤波[16]在时间序列分段中的应用和传统动态时间弯曲算法的改进。

为了方便研究,文中符号定义如下:

$X=\{(t_1,v_1),(t_2,v_2),\cdots,(t_m,v_m)\}$:长度为 m 的一维时间序列数据,t_i 和 v_i 分别表示第 i 个元素的时间值和序列值;

$x=(x_1,x_2,\cdots,x_m)\in\mathrm{R}^m$:长度为 m 的一维时间序列,x_i 为时间 $t(t=1,2,\cdots,m)$ 时刻对应的序列值,x 可以简写为 $x=\{x_t\}_{t=1}^m$;

$[1:N]$:$[1:N]=\{1,2,\cdots,N\}$;

$a\wedge b$:$a\wedge b=\min\{a,b\}$;

$|D|$:矩阵 D 的绝对值;

$|X|$:$|X|=\sum_{i=1}^m t_i/m$,$X=\{(t_1,v_1),(t_2,v_2),\cdots,(t_m,v_m)\}$;

$\hat{X}$:$\hat{X}=\sum_{i=1}^m v_i/m$,$X=\{(t_1,v_1),(t_2,v_2),\cdots,(t_m,v_m)\}$;

$\|u\|_1$:向量 $u=(u_1,u_2,\cdots,u_n)$ 的 l_1 范式,记为 $\|u\|_1=\sum_i|u_i|$;

$\|u\|_2$:向量 $u=(u_1,u_2,\cdots,u_n)$ 的 l_2 范式,记为 $\|u\|_2=\sqrt{\sum_i|u_i^2|}$;

$\overline{L}=\langle(t_L,v_L),(t_R,v_R)\rangle$:$\overline{L}$ 表示线性分段,(t_L,v_L) 和 (t_R,v_R) 分别表示 $\overline{L}$ 左边和右边的端点。

1.1　l_1 趋势滤波

给定长度为 n 的一维时间序列 $y=\{y_t\}_{t=1}^n$,由趋势项二阶差分的绝对值 $x=\{x_t\}_{t=1}^n$ 和波动项的平方和 $z=\{z_t\}_{t=1}^n$ 组成,通过最小化式(1)中的目标函数获得整体趋势 x:

$$\min Q(x)=\frac{1}{2}\sum_{t=1}^{n}(y_t-x_t)^2+\lambda\sum_{t=2}^{n-1}|x_{t-1}-2x_t+x_{t+1}| \tag{1}$$

其中:λ 是规则化非负参数,控制了 x 的平滑度和残差 $z(z=y-x)$ 的权重。目标函数式(1)中的第一项表示残差 $y-x$;第二项定量描述趋势 x 的平滑度,$x_{t-1}-2x_t+x_{t+1}$ 是在时间 t 计算的时间序列 x 的二阶差分,当且仅当三个点 x_{t-1}、x_t、x_{t+1} 在一条直线上时,二阶差分值为 0。

式(1)可以简写成式(2),$D\in\mathrm{R}^{(n-2)\times n}$ 是一个二阶差分矩阵:

$$\min Q(x)=\frac{1}{2}\|y-x\|_2^2+\lambda\|Dx\|_1 \tag{2}$$

$$D=\begin{bmatrix}1&-2&1&0&\cdots&0&0\\0&1&-2&1&\cdots&0&0\\&&\ddots&\ddots&\ddots&&\\0&0&\cdots&1&-2&1&0\\0&0&\cdots&0&1&-2&1\end{bmatrix}$$

目标函数式(2)是严格的凸函数,因此具有唯一的最小值 x^{lt}(上标 l_t 代表 l_1 趋势)。令 $x=A\theta$;目标函数式(2)可以转换为 l_1 正则化约束的最小二乘拟合问题:

$$\min Q(\theta)=\|A\theta-y\|_2^2+\lambda\sum_{i=3}^{n}|\theta_i| \tag{3}$$

其中:$\theta=(\theta_1,\theta_2,\cdots,\theta_n)\in\mathrm{R}^n$ 是变量 θ_i 的向量表示;A 是下三角矩阵,如下:

$$A=\begin{bmatrix}1&0&0&0&0&0\\1&1&0&0&0&0\\1&2&1&0&0&0\\1&3&2&1&0&0\\\vdots&\vdots&\vdots&\ddots&\ddots&\\1&n-1&n-2&\cdots&2&1\end{bmatrix}$$

优化问题式(3)的 θ^{lt} 和 l_1 趋势估计 x^{lt} 之间的关系为

$$x^{lt}=A\theta^{lt} \tag{4}$$

从 l_1 正则化约束最小二乘法拟合的结果[17]可知,式(3)的解 θ^{lt} 是正则化参数 λ 的分段线性函数。由式(4)可知,l_1 趋势滤波的正则化路径也是分段线性的。也就是有 p 个整数时间点 t_i 满足条件 $1=t_1<t_2<\cdots<t_p=n$,x_t^{lt} 是第 i 个时间窗口 $[t_i,t_{i+1}]$ 的函数,即 x_t^{lt},$t\in[t_i,t_{i+1}]$ 可以表示为线性段 $\overline{X_i}=\langle(t_i,x_{t_i}),(t_{i+1},x_{t_{i+1}})\rangle(i=1,2,\cdots,p-1)$,因此,原始时间序列 y 可以通过 l_1 趋势滤波用 $p-1$ 个线性段近似表示。

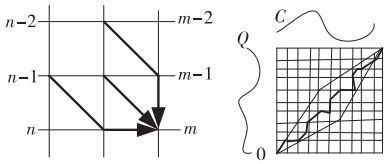
1.2　动态时间弯曲和自适应赋权函数

时间序列 $Q=q_1,q_2,\cdots,q_n$,$C=c_1,c_2,\cdots,c_m$,长度分别为 n 和 m,构造 $n\times m$ 的距离矩阵 $D_{n\times m}$,矩阵 $D_{n\times m}$ 中的元素 $d_{ij}=d(q_i,c_j)$ 表示 q_i 和 c_j 之间的距离,如图 1 在距离矩阵 $D_{n\times m}$ 中,找出由一组相邻矩阵元素的集合组成的最优弯曲路径,记为 $W=\langle w_1,w_2,\cdots,w_k\rangle$,$w_k=(i,j)$ 表示序列 Q 的第 i 个点和序列 C 中的第 j 个点匹配。这条路径满足下列三个条件:

a) $\max(n,m)\leqslant K\leqslant n+m-1$;

b) $w_1=(1,1)$,$w_k=(n,m)$ 即弯曲路径起于矩阵的左下角,止于右上角;

c) W 上任意两相邻元素在矩阵 D 中也相邻,且向前发展。即 $w_k=(a_k,b_k)$,$w_{k+1}=(a_{k+1},b_{k+1})$ 满足 $0\leqslant a_{k+1}-a_k\leqslant 1$,$0\leqslant b_{k+1}-b_k\leqslant 1$。



(a)三条不同路径　(b)最终弯曲路径

图1　时间序列的DTW弯曲路径

DTW 弯曲路径是所有可能的路径中累积距离最小的路径,DTW 算法可通过动态规划的方式求得,如式(5)所示。

$$\begin{aligned}D_{\mathrm{dtw}}(Q,C)=\min\{&D_{\mathrm{dtw}}(Q_{i-1},C_{j-1})+d(q_i,c_j),\\&D_{\mathrm{dtw}}(Q_{i-2},C_{j-1})+d(q_{i-1},c_j)+d(q_i,c_j),\\&D_{\mathrm{dtw}}(Q_{i-1},C_{j-2})+d(q_i,c_{j-1})+d(q_i,c_j)\}\end{aligned} \tag{5}$$

其中:q_i 和 c_j 分别为时间序列 Q 和 C 中的点;$D_{\mathrm{dtw}}(Q,C)$ 表示序列 Q 和 C 的 DTW 距离;$d(q_i,c_j)$ 表示 q_i 和 c_j 两点之间的欧氏距离。由于计算 DTW 时需要遍历整个 D 矩阵,导致 DTW 的计算复杂度为 $O(n\times m)$。式(5)中的三个元素表示到达 n 和 m 网格线交集的三个不同的路径,图 1(a)中每一个交叉点代表两个对应点间的欧氏距离。

对时间序列分段后,得到的子序列由不同个数的原始序列组成,直接用 DTW 距离不加区别地寻找最优弯曲路径,会忽略子序列个数的不同造成的子序列信息的丢失,因此在传统的 DTW 距离中引入了自适应赋权函数。改进的 DTW 公式如下:

$$\begin{aligned}D_{\mathrm{dtw}}(Q_{\mathrm{vec}},C_{\mathrm{vec}})=\min\{&D_{\mathrm{dtw}}(Q_{v-1},C_{u-1})+\delta_{v,u}\times\mathrm{SIM}_{i,j}(\bar{q}_v,\bar{c}_u),\\&D_{\mathrm{dtw}}(Q_{v-1},C_u)+\upsilon_{v,u}\times\mathrm{SIM}_{i,j}(\bar{q}_v,\bar{c}_u),\\&D_{\mathrm{dtw}}(Q_v,C_{u-1})+\eta_{v,u}\times\mathrm{SIM}_{i,j}(\bar{q}_v,\bar{c}_u)\}\end{aligned} \tag{6}$$

其中：$\bar{q}_v=(\bar{q}_1,\bar{q}_2,\cdots,\bar{q}_v)$ 和 $\bar{c}_u=(\bar{c}_1,\bar{c}_2,\cdots,\bar{c}_u)$ 分别是 Q_{vec} 和 C_{vec} 的子序列；$\delta_{v,u}$、$\upsilon_{v,u}$ 和 $\eta_{v,u}$ 是线性段 $\bar{q}_v$ 和 $\bar{c}_u$ 的权重函数，定义为

$$\begin{aligned}\delta_{v,u}&=2\min(n(\bar{q}_v)/m,n(\bar{c}_u)/n)+|n(\bar{q}_v)/m-n(\bar{c}_u)/n|\\ \upsilon_{v,u}&=2\min(n(\bar{q}_v)-1/m,n(\bar{c}_u)/n)+|(n(\bar{q}_v)-1)/m-n(\bar{c}_u)/n|\\ \eta_{v,u}&=2\min(n(\bar{q}_v)/m,n(\bar{c}_u)-1/n)+|n(\bar{q}_v)/m-(n(\bar{c}_u)-1)/n|\end{aligned}\quad(7)$$

其中：$n(x)/m$ 或者 $n(x)/n$ 表示子序列 x 包含的原始序列的个数在整个序列中所占的比重，每次迭代时，算法可根据线性段中原始序列的数量自适应改变权重，如此可以最大限度地减小因分段表示造成子序列信息的丢失，使算法具有动态自适应全局搜索与局部搜索能力。

2 基于形态距离的相似性测量

给定两个一维时间序列 $X=\{(t_1,v_1),(t_2,v_2),\cdots,(t_n,v_n)\}$ 和 $Y=\{(s_1,u_1),(s_2,u_2),\cdots,(s_m,u_m)\}$。在 l_1 趋势滤波之后，时间序列 X 由 K_1 个线性段组成，记为 $X=\{\bar{X}_1,\bar{X}_2,\cdots,\bar{X}_{K_1}\}$，其中 $\bar{X}_i=\langle(t_{Li},v_{Li}),(t_{Ri},v_{Ri})\rangle(i=1,2,\cdots,K_1)$。时间序列 Y 由 K_2 个线性段组成：

$$Y=\{\bar{Y}_1,\bar{Y}_2,\cdots,\bar{Y}_{K_2}\},\bar{Y}_i=\langle(s_{Lj},u_{Lj}),(s_{Rj},u_{Rj})\rangle\quad j=1,2,\cdots,K_2$$

对于任意两个一维时间序列，在计算距离之前必须首先计算两个线性段之间的距离，$\bar{X}=\langle(t_L,v_L),(t_R,v_R)\rangle$ 和 $\bar{Y}=\langle(s_L,u_L),(s_R,u_R)\rangle$ 分别表示时间序列 X 和 Y 中任意一条线性段。考虑到线性段的趋势信息，将两个给定的线性段在二维空间中通过平移和连接构成三角形，三角形的形态可以量化值轴与时间轴的信息。

2.1 线性段间的形态距离测量

通常在二维空间中不平行线段 $\bar{X}=\langle(t_L,v_L),(t_R,v_R)\rangle$ 和 $\bar{Y}=\langle(s_L,u_L),(s_R,u_R)\rangle$ 的位置关系分为有交点和没有交点两种，如图 2 所示。无论线段 $\bar{X}$ 和 $\bar{Y}$ 有无交点，平移其中一条线段（以 $\bar{Y}$ 为例），使得平移后两线段某一侧的端点重合，平移后的线段记为 $\bar{Y}'$，平移过程如图 3 所示。

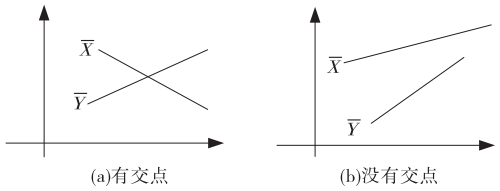


图 2 $\bar{X}$ 和 $\bar{Y}$ 不平行

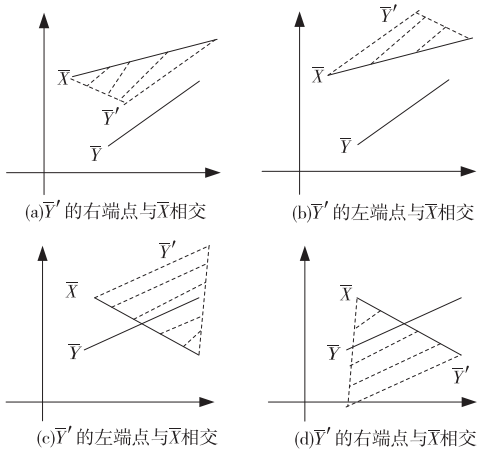


图3 线段 $\bar{Y}$ 平移至 $\bar{Y}'$

连接线段 $\bar{Y}'$ 和 $\bar{X}$ 不重合的端点，在二维空间中形成一个三角形区域。不难发现，图 3(a) 中三角形的面积与图 3(b) 中的相同。图 3(c) 中三角形面积与图 3(d) 相同。三角形的面积用三阶行列式表示：

$$A=\frac{1}{2}\left|\begin{vmatrix}t_L & v_L & 1\\ t_R & v_R & 1\\ s'_R & u'_R & 1\end{vmatrix}\right|$$

其中：$s'_R=t_L-s_L+s_R$，$u'_R=v_L-u_L+u_R$。

两条不平行线段间的形态距离 d 可以表示为

$$d(\bar{X},\bar{Y})=w_1\times|s_L-t_L|\Lambda|s_R-t_R|+w_2\times\sqrt{\frac{1}{2}\left|\begin{vmatrix}t_L & v_L & 1\\ t_R & v_R & 1\\ s'_R & u'_R & 1\end{vmatrix}\right|}\quad(8)$$

公式由两部分组成，第一项的几何意义是使两条线段的某一端点重合，由于平移操作不会影响线段的几何特征，在下面论证利用三角形计算形态距离的合理性中默认序列的端点已经重合，第二项为平移后的线段 $\bar{Y}'$ 和 $\bar{X}$ 之间的形态距离，也是趋势三角形计算形态距离的核心部分。系数 $w_1=\dfrac{1}{1+\dfrac{(t_R-t_L)+(s_R-s_L)}{(v_R-v_L)+(u_R-u_L)}}$ 表示原始时间序列 X 和 Y 的值轴信息，系数 $w_2=\sqrt{\dfrac{1}{|(t_R-t_L)(s_R-s_L)|}}$ 为原始时间序列 X 和 Y 的时间轴信息。

对于如图 4 所示的两条平行的线段 $\bar{X}=\langle(t_L,v_L),(t_R,v_R)\rangle$ 和 $\bar{Y}=\langle(s_L,u_L),(s_R,u_R)\rangle$，通过平移和连接不能形成三角形区域。计算形态距离时同样平移线段 $\bar{Y}$ 至 $\bar{Y}'$ 的位置，使线段 $\bar{Y}'$ 与 $\bar{X}$ 的某一侧端点重合。用 $\bar{X}$ 和 $\bar{Y}'$ 之间的长度差异表示线段间的形态距离。平行线段间的距离 d 定义为

$$d=w_1\times|s_L-t_L|\Lambda|s_R-t_R|+w_2\times|L_{\bar{X}}-L_{\bar{Y}}|\quad(9)$$

其中：参数 w_1 和 w_2 与式(8)定义一致，$L_{\bar{X}}=\sqrt{(t_R-t_L)^2+(v_R-v_L)^2}$，$L_{\bar{Y}}=\sqrt{(s_R-s_L)^2+(u_R-u_L)^2}$。

根据式(9)，若两个长度相同的平行线段对应相同的时间段（图 4(a)），距离 $d=0$；若长度相同的两个平行线段 $\bar{X}$ 和 $\bar{Y}$ 对应不同的时间段（图 4(b)），距离 $d=w_1\times|s_L-t_L|$，参数 w_1 与式(8)中定义的相同。

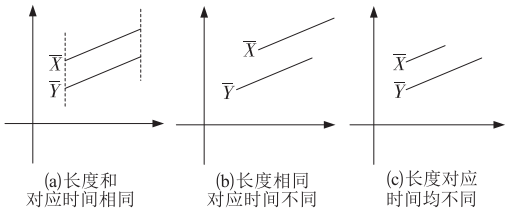


图4 $\bar{X}$ 和 $\bar{Y}$ 平行

2.2 形态距离理论说明

形态距离是如何有效度量两条线段间的距离，本节给出证明。

定理 若定义的距离函数满足：a)非负性，$d(A,B)>0$，if $(A\neq B)$；b)自相似性，$d(A,B)=0$，if$(A=B)$；c)对称性，$d(A,B)=d(B,A)$，$\forall A,B$；d)三角不等式性，$d(A,B)+d(A,C)\geqslant d(B,C)$。那么此距离函数可以称做一个相似性度量标准。

下面从以上四个方面证明将式(8)作为线段相似性度量标准的有效性。

由 2.1 节分析可知,式(8)的度量核心为后半部分 $w_2 \times \sqrt{\frac{1}{2}\left|\begin{vmatrix} t_L & v_L & 1 \\ t_R & v_R & 1 \\ s'_R & u'_R & 1 \end{vmatrix}\right|}$,即 $\sqrt{\frac{1}{2}\times w_2^2 \times \left|\begin{vmatrix} t_L & v_L & 1 \\ t_R & v_R & 1 \\ s'_R & u'_R & 1 \end{vmatrix}\right|}$。由于当 $a<b+c$ 时 $a<b+c+2\sqrt{bc}(a\geqslant 0,b\geqslant 0,c\geqslant 0)$,所以 $a<(\sqrt{b}+\sqrt{c})^2\Rightarrow\sqrt{a}<\sqrt{b}+\sqrt{c}$,可知根号不影响公式的三角不等式性、非负性、对称性以及自相似性。为了证明简便,不妨令

$$\frac{1}{2}\times\frac{1}{|(t_R-t_L)(s_R-s_L)|}\left|\begin{vmatrix} t_L & v_L & 1 \\ t_R & v_L & 1 \\ s'_R & u'_R & 1 \end{vmatrix}\right| = d(X,Y)$$

a)非负性。$d(X,Y)>0$,if$(X\neq Y)$由公式定义本身可知显然成立。

b)自相似性。即 $s_L=t_L,u_L=v_L,s_R=t_R,u_R=v_R$,此时两线段 X 与 Y 重合,$s'_R=s_R=t_R,u'_R=u_R=v_R$,故 $d(X,Y)=d(Y,X)=0$,if$(X=Y)$。

c)三角不等式性。证明:设 Z 是(o_L,p_L)到(o_R,p_R)的直线,则

$$d(X,Y)=\frac{1}{2}\times\frac{1}{|(t_R-t_L)(s_R-s_L)|}\left|\begin{vmatrix} t_L & v_L & 1 \\ t_R & v_L & 1 \\ s'_R & u'_R & 1 \end{vmatrix}\right| =$$

$$\frac{1}{2}\times\frac{1}{|(t_R-t_L)(s_R-s_L)|}|(v_R-v_L)(s_R-s_L)-(t_R-t_L)(u_R-u_L)| =$$

$$\frac{1}{2}\times\left|\frac{v_R-v_L}{t_R-t_L}-\frac{u_R-u_L}{s_R-s_L}\right|$$

由图 5 可得 $d(X,Y)=\frac{1}{2}\times\left|\frac{v_R-v_L}{t_R-t_L}-\frac{u_R-u_L}{s_R-s_L}\right|=\frac{1}{2}\times|\tan\theta_1-\tan\theta_2|$。

同理,$d(X,Z)=\frac{1}{2}|\tan\theta_1-\tan\theta_3|$,$d(Y,Z)=\frac{1}{2}|\tan\theta_2-\tan\theta_3|$,故

$$d(X,Y)=\frac{1}{2}|\tan\theta_1-\tan\theta_2|=\frac{1}{2}|\tan\theta_1-\tan\theta_3+\tan\theta_3-\tan\theta_2|\leqslant$$

$$\frac{1}{2}|\tan\theta_1-\tan\theta_3|+\frac{1}{2}|\tan\theta_3-\tan\theta_2|=d(X,Z)+d(Y,Z)$$

所以 $d(X,Y)\leqslant d(X,Z)+d(Y,Z)$满足三角不等式性质。证毕。

d)对称性。从 c)中 $d(X,Y)=|\tan\theta_1-\tan\theta_2|$可知 $d(Y,X)=|\tan\theta_2-\tan\theta_1|$显然 $d(X,Y)=d(Y,X),\forall X,Y$。

综上,用三角形面积可以作为形态距离的度量标准。趋势三角形的边角关系如图 5 所示。

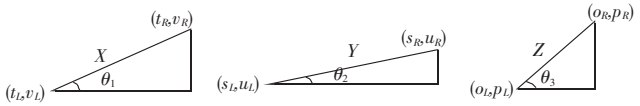


图 5　趋势三角形的边角关系

2.3　基于形态距离的相似性算法

在 2.1 节中,定义了两条不平行线段和两条平行线段之间的形态距离公式,对于长度为 n 的时间序列 $X=\{(t_1,v_1),(t_2,v_2),\cdots,(t_n,v_n)\}$ 和长度为 m 的时间序列 $Y=\{(s_1,u_1),(s_2,u_2),\cdots,(s_m,u_m)\}$之间的相似性度量,算法步骤如下:

a)l_1 趋势滤波。

用 l_1 趋势滤波得到时间序列 $X(i=1,2,\cdots,K_1+1;p_1<p_2<\cdots<p_{K_1}<p_{K_1+1})$ 的趋势估计和 K_1+1 个关键点$\{(t_{pi},v_{pi})\}$;同样,得到时间序列 $Y(j=1,2,\cdots,K_2+1;q_1<q_2<\cdots<q_{K_2}<q_{K_2+1})$ 的 K_2+1 个关键点$\{(s_{pi},u_{pi})\}$。

b)分段线性表示。

用分段线性表示法将时间序列 X 和 Y 分别近似表示为 $X=\{\overline{X}_1,\overline{X}_2,\cdots,\overline{X}_{K_1}\}$ 和 $Y=\{\overline{Y}_1,\overline{Y}_2,\cdots,\overline{Y}_{K_2}\}$,其中$\overline{X}_i=\langle(t_{p_i},v_{p_i}),(t_{p_{i+1}},v_{p_{i+1}})\rangle(i=1,2,\cdots,K_1;j=1,2,\cdots,K_1)$和$\overline{Y}_i=\langle(s_{q_j},u_{q_j}),(s_{q_{j+1}},u_{q_{j+1}})\rangle(i=1,2,\cdots,K_1;j=1,2,\cdots,K_2)$。

c)计算线性段之间的形态距离。

对于原始时间序列 X 和 Y 的任意两个线性段$\overline{X}_i$ 和$\overline{Y}_i$:若$\overline{X}_i$和$\overline{Y}_i$平行,用式(9)计算线段间的距离;否则用式(8)计算。

d)计算时间序列之间的距离。

在分段线性空间中,基于自适应函数约束下的动态时间弯曲计算时间序列 $X=\{\overline{X}_1,\overline{X}_2,\cdots,\overline{X}_{K_1}\}$ 和 $Y=\{\overline{Y}_1,\overline{Y}_2,\cdots,\overline{Y}_{K_2}\}$间的距离。

3　实验研究与分析

3.1　实验数据集及其平台

实验使用 UCR 数据集[18]与 Datamarket[19]中的两条数据集,与现有的相似性度量方法在分类性能上进行比较,评估 SD_AWTD 算法的有效性。数据集的相关信息以及本文的实验平台如表 1 ~ 3 所示。

表 1　UCR 数据集

名称	训练集	测试集	序列长度	类别数
adiac	390	391	176	37
gun-point	50	150	150	2
ECG	500	4 500	140	5
face(all)	560	1 690	131	14
face(four)	24	88	350	4
fish	175	175	463	7
OSU leaf	200	242	427	6
wafer	1 000	6 164	152	2
MALLAT	55	2 345	4 024	8
inlineSkate	100	550	1 882	7
handOutlines	370	1 000	2 709	2
haptics	155	308	6 869	5
phoneme	214	1 896	2 310	39

表 2　Datamarket 中的两条数据集

序列名称	序列长度	序列名称	序列长度
commercial	3 094	bank	2 510

表 3　实验平台

结构	配置
CPU	Intel Core i3-2350M CPU @2.30 GHz
内存	4.00 GB
操作系统	64 bit Windows 7
软件平台	MATLAB R2012a

3.2　算法的时间开销

根据 2.3 节的算法描述,本文的 SD_AWTD 算法包括三个过程:l_1 趋势滤波(分段表示)、线性段之间的距离计算以及时间序列之间的距离计算。因此,SD_AWTD 算法的时间成本也由上述三个过程组成:时间序列 PLR 表示的时间复杂度为 $O(n)$,子序列之间的形态距离计算复杂度为 $O(K_1K_2)$、序列间的弯曲距离计算复杂度为 $O(K_1K_2)$(K_1、K_2 为两时间序列分得的线性段数)。其中 $K_1<<n,K_2<<m$,因此,SD_AWTD 算法的时间复杂度为 $O(n)$。

一般采用 DTW 和 DDTW 算法计算两个不等长时间序列的复杂度。因此,将 DTW 和 DDTW 算法与本文提出的 SD_AWTD 算法在 Datamarket 提供的两条数据集上进行时间上的比较。实验运行时间如表 4 所示。

表 4 SD_AWTD、DTW 和 DDTW 算法运行的时间

比较项	SD_AWTD(80%)	SD_AWTD(90%)	DTW	DDTW
分段表示	3.200 0	1.394 1		
线性段间距离计算	1.181 8	0.020 2	41.205 3	47.030 4
两序列间距离计算	1.237 0	0.198 4		
总时间/s	5.618 8	1.612 7		

按照表 4 进行定量分析,不难发现在 SD_AWTD 算法中第一步(分段线性近似)消耗的时间要远远大于其他两步,但是,SD_AWTD 算法消耗的总时间比 DTW 和 DDTW 算法减小了一个很大的常数倍,这个常数主要依赖于分段的压缩率。当压缩率达到 60% 时,时间减少了 10 倍左右;当压缩率达到 90% 时,时间复杂度减少 30 倍左右。理论上原始时间序列由 l_1 趋势滤波分段表示,提取时间序列的主要特征,同时大大降低了序列的维度,将线性段视为某种模式,提出的方法将基于点距离转换为基于模式距离,都是降低计算复杂度的有效手段。综上,对于长时间序列的处理,SD_AWTD 比 DTW 和 DDTW 算法有更高的计算效率。

3.3 分类效果对比实验

为了证明 SD_AWTD 算法的可行性和优越性,对 SD_AWTD 和粒度动态时间规整(GDTW)、欧氏距离(Euclidean)以及经典时间弯曲距离(DTW)进行分类实验,观察比较各算法的分类结果。利用 SD_AWTD 和经典的距离度量算法对测试集中的测试序列在训练集中查找最相似即形态距离最小的序列实现最近邻分类,并通过判断测试序列与最相似序列标签的一致性度量分类效果的好坏。各算法的实验结果如表 5 所示。

表 5 各算法的运行结果

数据集	Euclid	DTW	GDTW	EDR	SD_AWTD
adiac	0.390	0.392	0.415	0.385	**0.320**
gun-point	0.088	0.088	0.008	0.021	**0.001**
ECG	0.121	0.121	0.171	**0.101**	0.171
face(all)	0.287	0.193	0.266	0.195	**0.128**
face(four)	0.217	0.115	0.376	**0.035**	0.262
fish	0.218	0.161	0.104	0.081	**0.018**
OSU leaf	0.484	0.385	0.121	0.216	**0.104**
wafer	0.006	0.023	**0.006**	0.012	0.008
MALLAT	0.379	0.252	0.225	0.318	**0.198**
inlineSkate	0.246	0.180	0.328	0.148	**0.131**
handOutlines	0.213	0.157	0.115	0.125	**0.096**
haptics	0.425	0.391	0.315	**0.288**	0.301
phoneme	0.170	0.155	0.180	0.194	**0.123**

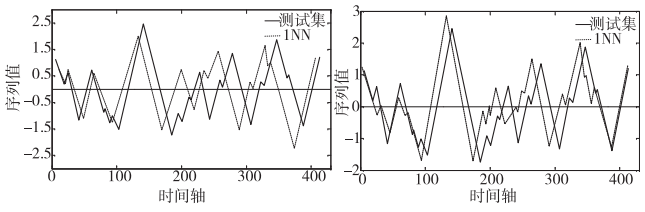
表中加粗的数据表示 SD_AWTD 算法在该数据集上错误率最小的实验结果。从表 5 中不难看出,SD_AWTD 算法在大部分数据集中分类错误率普遍较低,同时具有较好的学习效果,其中在 wafer 和 haptics 数据集上的实验错误率也接近最小值。

3.4 SD_AWTD 算法的鲁棒性实验

3.4.1 针对振幅和时间偏移的鲁棒性实验

给定数据集 OSU leaf 中的一条测试序列 x,在训练集中找到与 x 最相似(距离最小)的序列,比较 SD_AWTD 与 DTW 算法的优劣,讨论 SD_AWTD 算法的性质。如图 6 中的实线表示给定的测试序列 x,虚线表示分别为用 SD_AWTD 和 DTW 算法在训练集中找到的与测试序列最相似的序列。其中使用 SD_AWTD 算法得到的测试序列与最相似序列类别标签一致,DTW 算法得到的测试序列的类别标签与最相似序列不一致。

比较图 6(a)和(b),虽然 DTW 算法找到的相似序列似乎比 SD_AWTD 算法更加接近原始序列,但是仔细研究序列的形态特征发现,DTW 算法找到的相似序列中存在微小波动,不能很好地匹配原始序列。与此相反,SD_AWTD 算法找到的相似序列与原始序列虽然有相对不同的振幅和相位,但是在序列形态特征和轨迹上更加相似。这个实验表明,SD_AWTD 算法通过比较序列间的形态距离和特征对于涉及振幅和时间偏移的数据有更强的鲁棒性,相似性度量结果更加精确。



(a)SD_AWTD算法得到的相似序列　(b)DTW算法得到的相似序列

图6 实线表示的测试序列查找到的相似序列

3.4.2 SD_AWTD 算法对突变点的鲁棒性实验

从每一个 UCR 数据集的测试集中随机选取一条时间序列 Q,用一个异常值替换 Q 中的一个值得到序列 Q',异常值随机设置为 3,分别用 SD_AWTD 和 DDTW 算法计算生成序列与测试序列间的最佳弯曲路径长度,即 SD_AWTD(Q,Q')和 DDTW(Q,Q'),并且 SD_AWTD(Q,Q)和 DDTW(Q,Q)表示原测试序列之间弯曲路径的长度,为了直接比较两者的不同,定义 $\Delta SD\_AWTD = \frac{SD\_AWTD(Q,Q')}{SD\_AWTD(Q,Q)}$ 以及 $\Delta DDTW = \frac{DDTW(Q,Q')}{DDTW(Q,Q)}$ 为最佳弯曲路径长度(即相似性)的变化率。

图 7 显示了两种算法弯曲路径距离间的倍数关系。数据集的顺序与表 3 中的顺序相同,即最左边和最右边的数据分别为 adiac 和 phoneme。可以看出,用 DDTW 算法计算 Q 与 Q'之间的相似性除了 water 和 phoneme 数据集外,其他数据集均超过了两倍的原始最佳距离值,SD_AWTD 算法得到的结果则普遍较小。实验结果表明,对于包含离群点的数据,SD_AWTD 比 DDTW 算法有更强的鲁棒性。

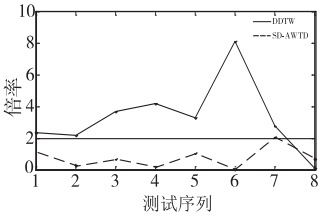


图7 最佳路径倍率关系

3.5 实验小结

综合以上四个实验可知:SD_AWTD 算法在通过 l_1 趋势滤波提取序列特征将原始序列分段表示的基础上,采用形态距离代替欧氏距离计算各个路径点的值构造距离矩阵,在动态时间弯曲的思想上结合自适应权值函数查找最优弯曲路径,对时间序列中偏移、幅值和时间轴方向伸缩等欧氏距离不能妥善解决的问题不敏感,可以完成两条序列宏观意义上的形态相似,SD_AWTD 算法比传统 DTW 算法有更高的精确性和较低的时间复杂度,对不等长或大规模的候选序列能够很好地完成相似性搜索任务,实验表明算法有着较强的鲁棒性。

4 结束语

针对传统的动态时间弯曲算法对高复杂度时间序列学习能力不足的问题,本文提出了基于形态距离及自适应权重的相似性度量算法,实验结果表明该方法具有强大而高效的离散数据处理能力。同时 DTW 算法对离群点以及时间偏移不敏感,有利于用户的实际选用。当前,随着序列相似性度量算法研究的不断深入,人们基于不同的理论提出了更为 (下转第 2647 页)

表 6　本文算法估算的自相关函数矩阵秩的下界值

n 值	AR(5)	AR(10)	AR(15)
$n=100$	4	8	10
$n=500$	4	8	13
$n=1000$	4	8	13

综上所述,在满足一定样本长度的要求下,本文算法的辨识精度和计算性能的鲁棒性能均优于现有算法,据此,本文算法是有效和可行的。

4　结束语

自回归模型是平稳时序数据中最常用和最广泛的辨识模型,基于现有算法的基础上,提出了一种改进的快速辨识算法,该算法较现有算法而言具有更稳定的辨识精度和更良好的计算效能。下一步的主要工作有,研究基于小样本场合的自相关函数的高精度估算方法,同时,也需要研究更为精确的矩阵的秩的下界值估计方法,以便进一步提升算法的适用范围和计算性能。

参考文献:

[1] 姜婷婷,肖卫东,张翀,等. 基于桑基图的时间序列文本可视化方法[J]. 计算机应用研究,2016,33(9):2683-2687.

[2] 王宏禹,邱天爽. 确定性信号分解与平稳随机信号分解的统一研究[J]. 通信学报,2016,37(10):1891-1898.

[3] 邹柏贤,刘强. 基于 ARMA 模型的网络流量预测[J]. 计算机研究与发展,2002,39(12):1645-1652.

[4] 程浩,刘国庆,成孝刚. 一种分段平稳随机过程自相关函数逼近模型[J]. 计算机应用,2012,32(2):589-591.

[5] 黄雄波. 非平稳时序数据的分段辨识及其递推算法[J]. 计算机系统应用,2017,26(5):180-185.

[6] Deng Feng, Bao Changchun. Speech enhancement based on AR model parameters estimation[J]. Speech Communication, 2016, 79(5): 30-46.

[7] 丁锋. 系统辨识算法的复杂性、收敛性及计算效率研究[J]. 控制与决策,2016,31(10):1729-1741.

[8] Boshnakov G N, Lambert-Lacroix S. A periodic Levinson-Durbin algorithm for entropy maximization[J]. Computational Statistics and Data Analysis, 2012, 56(1): 15-24.

[9] 胡明慧,王永山,邵惠鹤. 基于改进的莱文森算法对电机转速特性的研究[J]. 中国电机工程学报,2007,27(30):77-80.

[10] Matsuura M. On a recursive method including both CG and Burg's algorithms[J]. Applied Mathematics and Computation, 2012, 219(10): 773-780.

[11] 周毅,丁锋. 依等价 AR 模型阶次递增的自回归滑动平均模型辨识[J]. 华东理工大学学报:自然科学版,2008,34(3):425-431.

[12] 张仪萍,王士金,张土乔. 沉降预测的多层递阶时间序列模型研究[J]. 浙江大学学报:工学版,2005,39(7):983-986.

[13] Liu Yanjun, Ding Feng, Shi Yang. An efficient hierarchical identification method for general dual-rate sampled-data systems[J]. Automatica, 2014, 50(3): 962-970.

[14] 陈茹雯,湛时时. 基于非线性自回归时序模型的振动系统辨识[J]. 计算机应用研究,2016,33(10):3021-3025.

[15] 胡兴凯,伍俊良. 矩阵秩的下界和特征值估计[J]. 山东大学学报:理学版,2009,44(8):46-50.

[16] 黄廷祝. 矩阵秩的下界估计与 Schur 不等式的改进[J]. 电子科技大学学报,1993,22(5):537-541.

[17] Koltchinskii V, Lounici K, Tsybakov A B. Estimation of low-rank covariance function[J]. Stochastic Processes and Their Applications, 2016, 126(12): 3952-3967.

[18] 马铁丰,王松桂. 线性混合模型方差分量的检验[J]. 高校应用数学学报:A 辑,2007,22(4):433-440.

[19] 刘晓鹏, 刘坤会. F 分布密度函数之性质[J]. 应用概率统计, 2005,21(3):304-314.

(上接第 2642 页)先进的相似性度量算法。在 l_1 趋势滤波的基础上如何更好地对时间序列进行分段,在去噪的同时也能保留时间序列的特征点或关键点是以后要深入研究的方向。

参考文献:

[1] 杨一鸣,潘嵘,潘嘉林,等. 时间序列分类问题的算法比较[J]. 计算机学报,2007,30(8):1259-1266.

[2] Yu Fusheng, Dong Keqiang, Chen Fei, *et al*. Clustering time series with granular dynamic time warping method[C]//Proc of IEEE International Conference on Granular Computing. Washington DC: IEEE Computer Society, 2007: 393.

[3] Wan Yuqing, Gong Xueyuan, Si Y W. Effect of segmentation on financial time series pattern matching[J]. Applied Soft Computing, 2016, 38(1): 346-359.

[4] Rasheed F, Alhajj R. A framework for periodic outlier pattern detection in time-series sequences[J]. IEEE Trans on Cybernetics, 2014, 44(5): 569-582.

[5] Frawley C. Fast subsequence matching in time-series database[J]. Proc Sigmod, 2008, 23(2): 419-429.

[6] Berndt D J, Clifford J. Using dynamic time warping to find patterns in time series[C]//Proc of Working Notes of the Knowledge Discovery in Databases Workshop. Palo Alto, CA: AAAI Press, 1994: 359-370.

[7] Chen Lei, Ng R. On the marriage of Lp-norms and edit distance[C]// Proc of the 13th International Conference on Very Large Data Bases. Toronto: VLDB Endowment, 2004: 792-803.

[8] Vlachos M, Kollios G, Gunopulos D. Discovering similar multidimensional trajectories[C]//Proc of the 18th International Conference on Data Engineering. Piscataway, NJ: IEEE Press, 2002: 673-684.

[9] Sun Youqiang, Li Jiayong, Liu Jixue, *et al*. An improvement of symbolic aggregate approximation distance measure for time series[J]. Neurocomputing, 2014, 138(8): 189-198.

[10] Wang Xiaoyue, Mueen A, Ding Hui, *et al*. Experimental comparison of representation methods and distance measures for time series data[J]. Data Mining and Knowledge Discovery, 2013, 26(2): 275-309.

[11] 肖瑞,刘国华. 基于趋势的时间序列相似性度量和聚类研究[J]. 计算机应用研究,2014,31(9):2600-2605.

[12] 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.

[13] 李桂玲. 时间序列的分割及不一致发现研究[D]. 武汉:华中科技大学,2012.

[14] 丁永伟,杨小虎,陈根才,等. 基于弧度距离的时间序列相似度量[J]. 电子与信息学报,2011,33(1):122-128.

[15] 刘博宁,张建业,张鹏,等. 基于曲率距离的时间序列相似性搜索方法[J]. 电子与信息学报,2012,34(9):2200-2207.

[16] 秦磊,谢邦昌. L1 和 L2 规则化趋势滤波的稳健集成方法[J]. 统计研究,2013,30(11):99-102.

[17] Efron B, Hastie T, Johnstone I, *et al*. Least angle regression[J]. Annals of Statistics, 2004, 32(2): 407-451.

[18] Keogh E, Xi X, Wei L, *et al*. The UCR time series classification/clustering homepage [EB//OL]. 2006. http://www.cs.ucr.edu/~eamonn/time_series_data/.

[19] Derryberry D W R. Appendix A: using datamarket[M]//Basic Data Analysis for Time Series with R. Hoboken: Wiley, 2014.