

时间序列数据趋势转折点提取算法

邢 邗^a, 石晓达^a, 孙连英^b, 葛 娜^a

(北京联合大学 a. 信息学院; b. 自动化学院, 北京 100101)

摘 要: 时间序列数据蕴含趋势信息, 可以根据数据的趋势信息提取趋势转折点, 达到压缩数据、减少噪声影响的目的。通过分析时间序列数据的趋势信息, 提出自适应数据趋势转折点提取算法。该算法不依赖任何先验知识, 根据数据本身的趋势特征自动提取趋势转折点, 提取信息包括坐标索引和对应数据。UCR 时间序列分类数据集与 SEEP、CAP 和 PAA 等算法进行对比的实验结果表明, 在多种数据情况下, 该算法拟合误差和分类错误率更小, 平均拟合误差为 0.373 6, 分类错误率同原始数据的分类错误率相比减少 3.39%。

关键词: 时间序列; 趋势转折点; UCR 时间序列分类数据集; 分段线性表示; 拟合误差

中文引用格式: 邢 邗, 石晓达, 孙连英, 等. 时间序列数据趋势转折点提取算法[J]. 计算机工程, 2018, 44(1): 56-61, 68.

英文引用格式: XING Han, SHI Xiaoda, SUN Lianying, et al. Trend Turning Point Extraction Algorithm for Time Series Data[J]. Computer Engineering, 2018, 44(1): 56-61, 68.

Trend Turning Point Extraction Algorithm for Time Series Data

XING Han^a, SHI Xiaoda^a, SUN Lianying^b, GE Na^a

(a. School of Information; b. School of Automation, Beijing Union University, Beijing 100101, China)

[Abstract] The time series data contains trend information, which can extract the trend turning point according to the trend information of the data, and can achieve the purpose of compressing the data and reducing the influence of noise. By analyzing the trend information of time series data, an adaptive data trend turning point extraction algorithm is proposed. The algorithm does not rely on any prior knowledge, only according to the trend characteristics of the data itself automatically extract the trend turning point, extracted information including the coordinate index and the corresponding data. Compared with SEEP, CAP and PAA algorithm, experimental results show that the fitting error and classification error rate of the algorithm are smaller in the case of multiple data, and the average fitting error is 0.373 6, the classification error rate compared with the original data classification error rate decreases by 3.39%.

[Key words] time series; trend turning point; UCR time series classification dataset; Piecewise Linear Representation (PLR); fitting error

DOI: 10.3969/j.issn.1000-3428.2018.01.009

0 概述

时间序列数据正在以一种难以想象的规模和速度快速增长, 已经在商业、医药和科学研究等领域占据了大量的存储空间。对原始数据直接进行数据挖掘, 不仅要花费大量时间, 而且由于噪声点的存在还会导致算法准确度下降。在通常情况下, 需要对原始序列数据进行数据的高级表示^[1], 常用方法有奇异值分解^[2-3]、离散傅里叶变换^[4-5]、离散小波变换^[6-7]、主成分分析^[8]和分段线性表示^[9-10]等。

其中分段线性表示 (PLR) 方法是使用直线段近

似表示时间序列, 具有时间多解析性、支持快速相似性搜索和数据压缩度灵活等优点。而在分段线性表示方法中, 分段点的选择显得极为重要。使用提取的分段点可以近似拟合原始数据和相似性度量等运算, 大大减少运算成本。有许多学者已经在分段点提取方面做出贡献, 如文献[11]提出用等宽度窗口中数据的平均值表示时间序列 (PAA), 文献[12]提出基于重要点的分段线性方法 (PCA)。在国内, 文献[13]提出根据相邻两点间斜率的变化提取时间序列上的分段点 (SEEP), 文献[14]提出基于一阶滤波的时间序列分段线性表示方法 (SFWF), 文献[15]

基金项目: 国家重点研发计划项目 (2016YFC0802107)。

作者简介: 邢 邗 (1992—), 男, 硕士研究生, 主研方向为时间序列数据挖掘、机器学习; 石晓达, 硕士研究生; 孙连英 (通信作者), 教授、博士; 葛 娜, 硕士研究生。

收稿日期: 2016-12-06

修回日期: 2017-02-14

E-mail: xh1992442@163.com

提出基于序列重要点概念达到时间序列分割算法(SIP),文献[16]提出根据偏离度提取变化点的时间序列分段线性表示(CPA)。

本文给出一种提取时间序列数据趋势转折点算法(FTTO)。首先定义时间序列数据4个相邻数据点的运算,随后讨论运算结果 P 并且对 P 值小于零和等于零2种出现趋势转折点的情况分别进行讨论,然后提取原始数据中上升、下降和平缓3种基本趋势转折点。最后将相邻2个基本趋势转折点中间的数据做坐标旋转变换,对旋转变换得到的新的时间序列数据使用FTTO算法,得到缓慢上升、快速上升、缓慢下降、快速下降4种拐点。

1 基本定义

定义1(时间序列) 时间序列是有序的信息集合。每一个元素包括记录信息和时间。长度为 n 时间序列 X 记为:

$$X = \{ \langle x_1, t_1 \rangle, \langle x_2, t_2 \rangle, \dots, \langle x_i, t_i \rangle, \dots, \langle x_n, t_n \rangle \} \quad (1)$$

其中, $\langle x_i, t_i \rangle$ 表示时间序列 X 在 t_i 时刻记录的数据值为:

$$x_i = \{a_1, a_2, \dots, a_m\}, m \geq 1 \quad (2)$$

在一般情况下,时间序列的采样间隔 Δt ($\Delta t = t_{i+1} - t_i$) 相等且大于零,也可以将时间序列 X 简记为 $\{x_1, x_2, \dots, x_n\}$ 。

定义2(分段线性表示) 时间序列数据 X 记为 $\{x_1, x_2, \dots, x_n\}$,趋势转折点集合为 $\{x_{t_1}, x_{t_2}, \dots, x_{t_i}, \dots, x_{t_m}\}$,其中, x_{t_1} 和 x_{t_m} 分别为时间序列的起始点和终点。其分段线性表示为:

$$X_{PLR} = \{ \langle x_{t_1}, x_{t_2} \rangle, \dots, \langle x_{t_i}, x_{t(i+1)} \rangle, \dots, \langle x_{t(n-1)}, x_{t_n} \rangle \} \quad (3)$$

其中, $\langle x_{t_i}, x_{t(i+1)} \rangle$ 表示在 $[t_i, t(i+1)]$ 时间段内的线性函数。

定义3(压缩率) 原始时间序列长度为 N ,通过提取趋势转折点算法得到新的时间序列长度为 M ,那么该算法的压缩率为 $(1 - \frac{M}{N}) \times 100\%$ 。

定义4 在时间序列 X 中,如果 x_i 是趋势转折点,那么 x_i 必是局部极值点^[15],即 x_i 满足:

$$\begin{aligned} & \{ (x_i \geq x_{i+1}) \cup (x_i > x_{i+1}) \} \cap \\ & \{ (x_i > x_{i+1}) \cup (x_i \geq x_{i+1}) \} \cap \{ (x_i \leq x_{i+1}) \cup (x_i < x_{i+1}) \} \cap \\ & \{ (x_i < x_{i+1}) \cup (x_i \leq x_{i+1}) \} \end{aligned}$$

定义5(拟合误差) 将 X_{PLR} 经过线性插值后得到的时间序列记为 $X^c = \{x_1^c, x_2^c, \dots, x_n^c\}$,与原始时间序列 $X = \{x_1, x_2, \dots, x_n\}$ 的拟合误差定义为^[13]:

$$E = \sqrt{\sum_{i=1}^n (x_i - x_i^c)^2} \quad (4)$$

定义6(基本趋势转折点) 时间序列数据的趋势可分为上升、平缓 and 下降3种基本趋势,根据这种趋

势提取的转折点可以被称为基本趋势转折点。

定义7(拐点) 缓慢上升、快速上升、缓慢下降和快速下降属于在一个基本趋势中单位时间内数据的变化速率不同,依据变化速率不同提取的趋势转折点被称为拐点。

2 趋势转折点提取算法与分析

2.1 提取趋势转折点算法

如图1所示^[15],在时间序列中相邻3个数据点分别为 x_i, x_{i+1} 和 x_{i+2} ,它们的发展趋势共有9种情况。当 $x_{i+2} - x_i > 0$ 时,即图1(a)、图1(b)、图1(e)这3种情况属于总体趋势上升。当 $x_{i+2} - x_i < 0$ 时,即图1(f)、图1(g)、图1(i)这3种情况属于总体趋势下降。当 $x_{i+2} - x_i = 0$ 时,即图1(c)、图1(d)、图1(h)这3种情况属于总体保持不变。

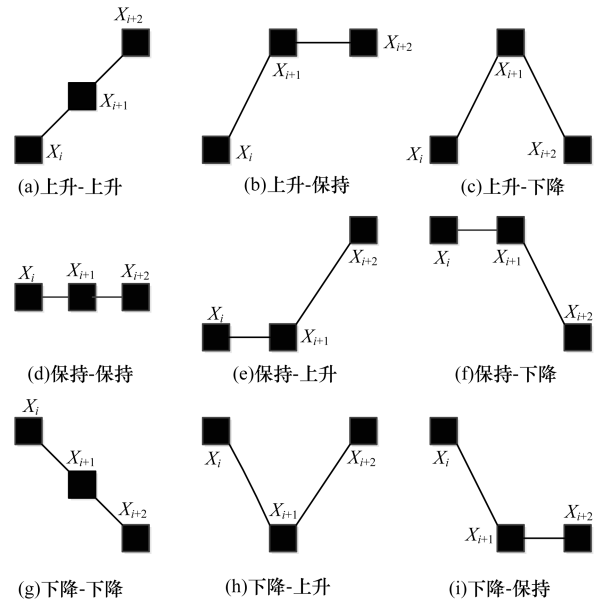


图1 时间序列中相邻3点的变化趋势

在时间序列 X 中相邻的4个数据点分别为 $x_i, x_{i+1}, x_{i+2}, x_{i+3}$,令:

$$(x_{i+2} - x_i) \times (x_{i+3} - x_{i+1}) = P \quad (8)$$

当 P 大于零时,在相邻的4个数据点中,前3个数据点与后3个数据点趋势相同。当 P 小于零时,在相邻的4个数据点中前后趋势不同,出现趋势转折点。当 P 等于零时,相邻的4个数据点中前后趋势可能不同。下面分别讨论 P 小于、等于零的情况。

1) 当 P 小于零时, $(x_{i+2} - x_i) > 0$ 且 $(x_{i+3} - x_{i+1}) < 0$, 前3点为上升趋势,分别对应图1(a)、图1(b)、图1(e)这3种情况,后3点为下降趋势。此时4个相邻数据点共有3种分布情况,如图2所示。依据定义4,趋势转折点是局部极值点,当上升趋势转变为下降趋势时,趋势转折点应为局部极大值点,所以比较 x_{i+1} 与 x_{i+2} 的大小,最大值是趋势转折点,若相等,则取 x_{i+1} 作为趋势转折点。

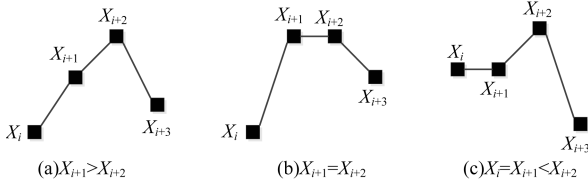


图 2 4 个相邻数据点上升转下降示意图

2) 当 P 小于零时, $(x_{i+2} - x_i) < 0$ 、 $(x_{i+2} - x_i) > 0$ 、 $(x_{i+2} - x_i) < 0$ 且 $(x_{i+3} - x_{i+1}) > 0$, 前 3 点为下降趋势, 分别对应图 1(f)、图 1(g) 和图 1(i) 3 种情况, 后 3 点为上升趋势。此时 4 个相邻数据点共有 3 种分布情况, 如图 3 所示。同理, 比较 x_{i+1} 与 x_{i+2} 的大小, 下降趋势转变为上升趋势, 所以局部极小值点是趋势转折点。若相等, 取 x_{i+1} 作为趋势转折点。

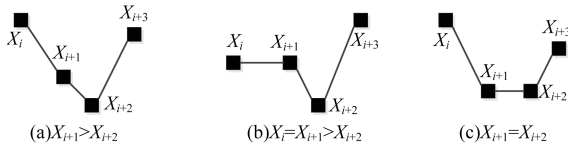


图 3 4 个相邻数据点下降转上升示意图

3) 当 P 等于零时, $(x_{i+2} - x_i) = 0$ 且 $(x_{i+3} - x_{i+1}) \neq 0$, 前 3 点趋势为保持不变, 后 3 点趋势并不确定, 如图 4 所示, 后 3 点的变化趋势共有 8 种情况。由图 4 可知, x_{i+2} 是 2 种不同趋势变化交点, 即取 x_{i+2} 作为趋势转折点。

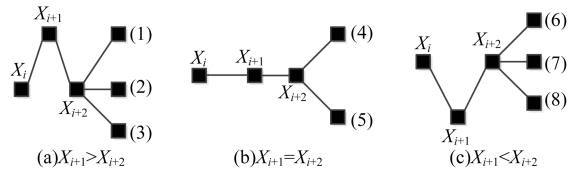


图 4 4 个相邻数据点保持转变化示意图

4) 当 P 等于零时, $(x_{i+2} - x_i) \neq 0$ 且 $(x_{i+3} - x_{i+1}) = 0$, 前 3 点趋势并不确定, 后 3 点趋势为保持不变, 如图 5 所示, 前 3 点的变化趋势共有 8 种情况。由图 5 可以发现, x_{i+1} 是 2 种不同趋势的变化交点, 即取 x_{i+1} 作为趋势转折点。

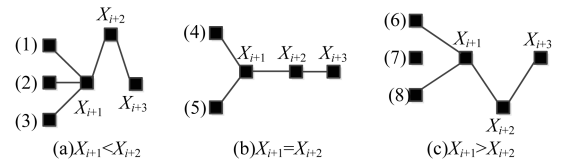


图 5 4 个相邻数据点变化转保持示意图

5) 当 P 等于零时, $(x_{i+2} - x_i) = 0$ 且 $(x_{i+3} - x_{i+1}) = 0$, 前 3 点与后 3 点的趋势相同保持不变, 如图 6 所示, 共有 3 种情况, 趋势均视为平缓, 所以此时无趋势转折点。

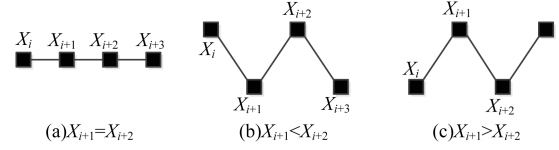


图 6 4 个相邻数据点趋势保持不变示意图

以上 5 种情况如表 1、表 2 所示。

表 1 当 $P < 0$ 时的情况

$x_{i+3} - x_{i+1}$	x_{i+1} 与 x_{i+2} 比较大小	趋势转折点
< 0	$x_{i+1} > x_{i+2}$	$Add(x_{i+1})$
	$Add(x_{i+2})$	$x_{i+1} < x_{i+2}$
	$Add(x_{i+1})$	$x_{i+1} = x_{i+2}$
> 0	$x_{i+1} > x_{i+2}$	$Add(x_{i+2})$
	$Add(x_{i+1})$	$x_{i+1} < x_{i+2}$
	$Add(x_{i+1})$	$x_{i+1} = x_{i+2}$

表 2 当 $P = 0$ 时的情况

$x_{i+3} - x_{i+1}$	$x_{i+2} - x_i$	趋势转折点
$\neq 0$	$= 0$	$Add(x_{i+2})$
$= 0$	$\neq 0$	$Add(x_{i+1})$
$= 0$	$= 0$	无转折点

2.2 拐点提取

对原始时间序列数据使用以上算法可以得到 3 种基本趋势转折点。利用 3 种基本趋势转折点还原数据。为减少拟合误差, 提高拟合精度, 定义阈值 Q , Q 代表相邻 2 个趋势转折点的距离, 即 $t_i - t_{(i-1)}$, 当趋势转折点与前一个转折点间的距离大于 Q 时, 提取拐点。

假设相邻 2 个基本趋势转折点为 x_{i_i} 和 $x_{i(i+1)}$, 2 个趋势转折点间的数据如图 7 所示, 图 7 中共有 A 和 B 2 个拐点。

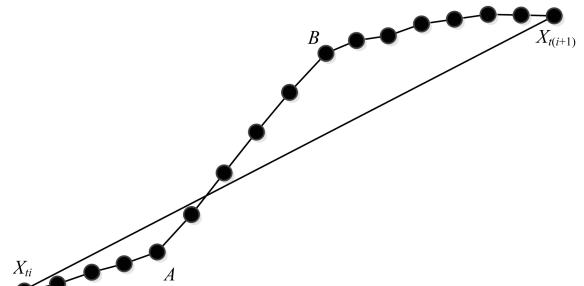


图 7 坐标转换前原始数据示意图

首先连接相邻的 2 个趋势转折点 x_{i_i} 和 $x_{i(i+1)}$, 以连线作为 X 轴。其次将 2 个趋势转折点之间的原始数据与连线的距离作为新的时间序列数据, 如图 8 所示。经过坐标旋转变换后, 将缓慢上升、快速上升、缓慢下降和快速下降等拐点的提取问题转换为基本趋势转折点提取问题。最后对新的时间序列数据采用 FTTO 算法便可提取拐点。因为 FTTO 算法运算使用相邻的 4 个数据点, 所以距离阈值 Q 应不小于 5。

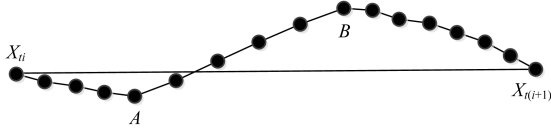


图8 坐标旋转转换后数据示意图

2.3 算法实现步骤

依据2.1节与2.2节提出的趋势转折点提取算法,实现FTTO算法共分为主函数和FTTO方法2个部分,如果需要提取拐点减少拟合误差,则添加FTTO-G方法部分。

主函数方法描述如下:

输入 时间序列 $X = \{x_1, x_2, \dots, x_n\}$

输出 时间序列趋势转折点集合 $X^c = \{x_1^c, x_2^c, \dots, x_n^c\}$

Begin

$X^c = \{\}\{\}\{\}$; /* 实例化一个序列,存储趋势转折点 */

$fi = 0$; /* 保存前一个趋势转折点时间索引 */

$se = 0$; /* 保存当前趋势转折点时间索引 */

$X^c.add(x_1)$ /* 数据起始点默认为趋势转折点 */

for($i = 1$; $i < n - 3$; $i++$)

/* 遍历原始数据,相邻4个点计算P值 */

$(x_{i+2} - x_i) \times (x_{i+3} - x_{i+1}) = P$

/* 依据FTTO算法,判断是否存在趋势转折点 */

FTTO($P, x_i, x_{i+1}, x_{i+2}, x_{i+3}$)

/* 如果需要提取拐点,则根据输入的Q值和相邻2个转折点间距离判断是否使用FTTO-G算法 */

if($se - fi > Q$)

FTTO - G(fi, se);

/* 在提取拐点后,更新fi坐标索引 */

$fi = se$;

$X^c.add(x_n)$

End

FTTO方法描述如下:

输入 P 和相邻的4个数据点 $x_{i+1}, x_{i+2}, x_{i+3}$ 和 x_{i+4}

输出 如果输入数据中存在趋势转折点,则 X^c 集合中添加转折点,如果输入数据中不存在趋势转折点,则不向 X^c 中添加数据

Begin

if($P < 0$)

if($x_{i+3} - x_{i+1} < 0$)

if($x_{i+1} \geq x_{i+2}$)

$X^c.add(x_{i+1})$; $se = i + 1$;

else if($x_{i+1} < x_{i+2}$)

$X^c.add(x_{i+2})$; $se = i + 2$;

else if($x_{i+3} - x_{i+1} > 0$)

if($x_{i+1} > x_{i+2}$)

$X^c.add(x_{i+2})$; $se = i + 2$

else if($x_{i+1} \leq x_{i+2}$)

$X^c.add(x_{i+1})$; $se = i + 1$;

else if($p = 0$)

if($(x_{i+3} - x_{i+1}) \neq 0 \&\& (x_{i+2} - x_i) = 0$)

$X^c.add(x_{i+2})$; $se = i + 2$;

else if($(x_{i+3} - x_i) = 0 \&\& (x_{i+2} - x_i) \neq 0$) $X^c.add(x_{i+1})$; $se = i + 1$;

End

FTTO-G方法描述如下:

输入 前一个趋势转折点时间坐标索引 fi ,当前趋势转折点时间坐标索引 se

输出 如果相邻2个趋势转折点之间存在拐点,则向 X^c 中添加数据,如果相邻2个趋势转折点之间不存在拐点,则 X^c 中数据不变

Begin

$G = \{\}$; /* 实例存储新的时间序列数据的序列 */

Calclinearequation(); /* 计算2个转折点间的直线方程 */

for(int? $w = fi$; $w \leq se$; $w++$)

$G.add(distance(x_w))$; /* 计算每个点到直线的距离,并添加到G中 */

for(int?? $j = 0$; $j < G.size - 4$; $j++$)

/* 对新的时间序列数据G,采用FTTO算法,将提取的趋势转折点添加到XC中 */

$(G_{j+2} - G_j) \times (G_{j+3} - G_{j+1}) = P$

FTTO($P, G_j, G_{j+1}, G_{j+2}, G_{j+3}$)

End

本文提出的FTTO算法时间复杂度为 $O(6n)$ 。在程序运行过程中,首先需要遍历原始数据,计算相邻的4个数据点的值 P ,之后通过3次判断得出该算法是否存在趋势转折点。那么,此时的时间复杂度 $T_1(n) = O((1+3) \times (n-3)) = O(4n)$ 。如果需要提取拐点,则增加FTTO-G方法部分。首先计算相邻2个趋势转折点间直线方程的斜率与截距。之后遍历相邻2个趋势转折点间数据,形成新的时间序列数据,此时的时候复杂度 $T_2(n) = O(n)$ 。最后对新的时间序列数据使用FTTO算法。此时的时间复杂度 $T_3(n) = O(n)$ 。所以,FTTO算法的时间复杂度为 $O(4n + n + n) = O(6n)$ 。

在提取基本趋势转折点过程中,FTTO算法将相邻的3个数据点作为基本趋势单元,不依赖任何先验知识,自动提取数据中基本趋势转折点。在提取拐点过程中,增加相邻2个基本趋势转折点间的距离阈值 Q ,对经过坐标变换后的数据使用FTTO算法提取拐点,从而提高拟合精度。

3 对比实验

3.1 对比拟合误差

本文实验数据采用美国加州大学河滨分校收集整理的UCR时间序列分类数据集^[17]。选择SEEP^[13]、PAA^[11]、CPA^[16]算法与本文提出的FTTO算法进行对比,使用不同算法提取数据点分段线性表示原始数据,之后对比拟合误差。拟合误差越小,证明算法拟合性能越优秀。

在下文实验中,由于数据集中数据来自不同领域,首先进行数据归一化处理,数据归一化公式如下:

$$\text{norm}(x_i) = \frac{x_i - \min(X)}{\max(X) - \min(X)} \quad (5)$$

文献[13]提出的 SEEP 算法是基于斜率提取数据点的代表性算法。首先计算连接同一点的左右两条线段的斜率。之后对比斜率的变化量是否超出阈值 d , 若大于阈值 d , 则提取分段点。最终连接分段点得到原始数据的分段线性表示。

文献[11]提出的 PAA 算法是分段线性表示的经典算法。首先使用等宽窗口划分原始数据, 之后每个窗口内使用平均值代替原始数据, 最后得到原始数据的分段线性表示。

文献[16]提出的 CPA 算法在提取分段点过程中具有良好效果。首先依据各个数据点在局部范围内的偏离程度定义了时间序列数据的变化点, 之后连接每一个变化点, 最后得到原始数据的分段线性表示。

以 synthetic_control 数据集的第一条数据为例, 比较 4 种算法的拟合效果。图 9 ~ 图 13 分别为归一化原始数据、FTTO 算法拟合数据、SEEP 算法拟合数据、CPA 算法拟合数据和 PAA 算法拟合数据效果示意图。

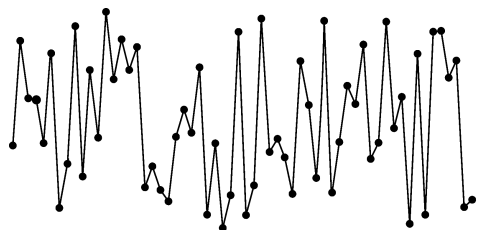


图 9 原始数据效果示意图

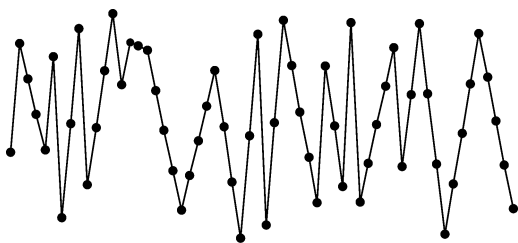


图 10 本文算法 FTTO 拟合数据效果示意图

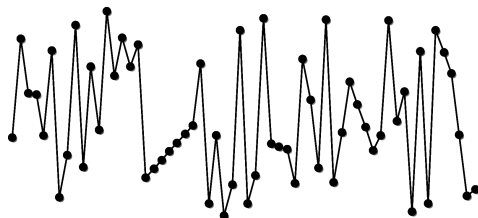


图 11 SEEP 算法拟合数据效果示意图

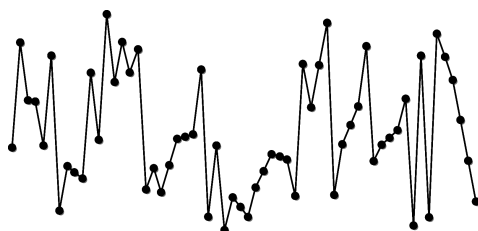


图 12 CPA 算法拟合数据效果示意图

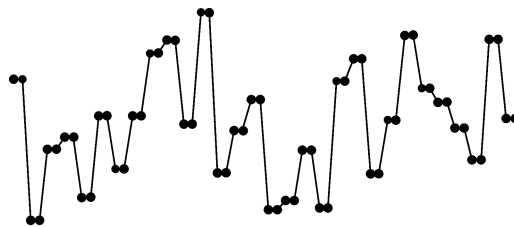


图 13 PAA 算法拟合数据效果示意图

从图 9 ~ 图 13 对比结果可以看出, 本文提出的 FTTO 算法能够提取数据主要趋势转折点达到保留原始数据的趋势信息的目的。SEEP 与 CPA 算法均存在遗漏提取分段点的情况, PAA 算法虽然保存了原始信息的整体趋势特征, 但是丢失了每个窗口内的趋势信息。

4 种算法在相同压缩率情况下拟合误差对比实验结果如表 3 所示, 表 3 中 R 表示压缩率, FTTO 算法提取拐点的阈值 Q 等于 5。由于 PAA 算法的自身特点, 实验过程中 PAA 算法压缩率大于且最接近表 3 中的压缩率。表 3 中下划线表示每行拟合误差最小值。

表 3 拟合误差对比结果

数据名称	SEEP	PAA	CPA	FTTO	R
synthetic_control	0.764	0.901	1.007	<u>0.683</u>	0.51
Gun_Point	<u>0.105</u>	0.275	0.280	0.260	0.74
CBF	0.800	0.873	0.974	<u>0.663</u>	0.50
FaceAll	0.848	0.607	0.949	<u>0.472</u>	0.69
OSULeaf	0.262	0.302	0.297	<u>0.145</u>	0.76
SwedishLeaf	0.308	0.379	0.422	<u>0.209</u>	0.74
50words	0.884	0.563	0.683	<u>0.360</u>	0.87
Trace	<u>0.109</u>	0.242	0.348	0.294	0.53
Two_Patterns	0.873	1.051	1.655	<u>0.593</u>	0.62
Wafer	0.855	0.919	1.393	<u>0.804</u>	0.68
FaceFour	0.830	0.795	1.013	<u>0.663</u>	0.66
Lighting2	<u>0.192</u>	0.580	0.838	0.630	0.63
Lighting7	<u>0.177</u>	0.620	0.824	0.688	0.61
ECG200	<u>0.257</u>	0.329	0.484	0.280	0.56
Adiac	0.324	0.450	0.313	<u>0.221</u>	0.90
yoga	0.168	0.322	0.147	<u>0.119</u>	0.81
FISH	0.180	0.288	0.217	<u>0.142</u>	0.88
Plane	0.193	0.511	0.362	<u>0.174</u>	0.78
Car	0.163	0.271	0.214	<u>0.128</u>	0.81
Beef	0.324	0.316	0.282	<u>0.214</u>	0.73
Coffee	0.371	0.414	<u>0.231</u>	0.277	0.74
OliveOil	0.605	0.638	0.668	<u>0.589</u>	0.86
CinC_ECG_torso	<u>0.225</u>	0.232	0.263	0.230	0.65
ChlorineConcentration	0.834	0.751	1.100	<u>0.726</u>	0.58
DiatomSizeReduction	0.176	0.325	0.198	<u>0.127</u>	0.84
ECGFiveDays	<u>0.076</u>	0.349	0.606	0.309	0.63
FacesUCR	0.773	0.891	0.687	<u>0.473</u>	0.69
Haptics	<u>0.133</u>	0.607	0.290	0.266	0.68
InlineSkate	0.081	0.101	0.106	<u>0.080</u>	0.54
ItalyPowerDemand	0.448	0.362	0.378	<u>0.333</u>	0.63

从表3可以看出,在30组时间序列数据中,SEEP算法在8类时间序列数据中拟合误差最小,CPA算法在1类时间序列数据中拟合误差最小,本文提出的FTTO算法在21类时间序列数据中拟合误差最小。

3.2 对比分类误差率

Keogh团队公布了在UCR时间序列分类数据集上,使用DTW算法在采用1-NN分类的情况下各个数据集的分类错误率。本文运用SEEP、CPA和FTTO算法对比拟合时间序列数据在采用1-NN分类和DTW距离度量情况下的分类错误率。对比结果如表4所示。在表4中,A列是Keogh团队公布的在原始数据上的分类错误率。各个数据的压缩率同表3中各个数据的压缩率相等。表4中下划线代表各行中的最小数据。

表4 UCR数据集对比分类结果

数据名称	A	SEEP	CPA	FTTO
synthetic_control	<u>0.007 0</u>	0.036 7	0.137	0.047 00
Gun_Point	<u>0.093 0</u>	0.127 0	0.140	0.127 00
CBF	<u>0.003 0</u>	0.024 0	0.022	0.178 00
FaceAll	<u>0.192 0</u>	0.234 0	0.273	0.216 00
OSULeaf	0.409 0	0.421 0	0.479	<u>0.120 00</u>
SwedishLeaf	0.208 0	0.278 0	0.322	<u>0.021 00</u>
50words	<u>0.310 0</u>	0.358 0	0.356	0.347 00
Trace	<u>0.000 0</u>	<u>0.000 0</u>	0.010	0.020 00
Two_Patterns	<u>0.000 0</u>	<u>0.000 0</u>	0.001	0.000 25
wafer	0.020 0	0.018 0	0.159	<u>0.012 00</u>
FaceFour	<u>0.170 0</u>	0.261 0	0.295	0.182 00
Lighting2	<u>0.131 0</u>	0.230 0	0.262	0.279 00
Lighting7	<u>0.274 0</u>	0.329 0	0.438	0.397 00
ECG200	0.230 0	0.180 0	0.190	<u>0.060 00</u>
Adiac	0.396 0	0.624 0	0.642	<u>0.125 00</u>
Yoga	0.164 0	0.191 0	0.192	<u>0.092 00</u>
FISH	0.177 0	0.223 0	0.206	<u>0.063 00</u>
Plane	<u>0.000 0</u>	0.010 0	0.019	0.010 00
Car	0.267 0	0.267 0	0.367	<u>0.067 00</u>
Beef	0.367 0	0.400 0	0.433	<u>0.300 00</u>
Coffee	<u>0.000 0</u>	0.071 0	0.071	0.071 00
OliveOil	<u>0.167 0</u>	0.200 0	0.333	0.333 00
CinC_ECG_torso	<u>0.349 0</u>	0.569 0	0.557	0.554 00
ChlorineConcentration	0.352 0	0.377 0	0.377	<u>0.143 00</u>
DiatomSizeReduction	<u>0.033 0</u>	0.291 0	0.258	0.098 00
ECGFiveDays	0.232 0	0.203 0	0.194	<u>0.181 00</u>
FacesUCR	<u>0.095 0</u>	0.102 0	0.134	0.149 00
Haptics	0.623 0	0.740 0	0.685	<u>0.584 00</u>
InlineSkate	<u>0.616 0</u>	0.667 0	0.664	0.620 00
ItalyPowerDemand	0.050 0	0.188 0	0.137	<u>0.024 00</u>

在表4中,有17组数据显示使用原始数据得到

的分类错误率最小,使用SEEP算法有2组数据的分类错误率同原始数据相同达到最小,使用本文提出的FTTO算法在13组数据中分类错误率最小。并且Keogh团队公布的使用原始时间序列数据的分类错误率平均值为0.209 2,使用SEEP算法获得的分类错误率平均值为0.254 0,使用CPA算法获得的分类错误率平均值为0.273 7,使用本文提出的算法FTTO获得的分类错误率平均值为0.175 4,平均分类错误率达到最小。平均分类错误率相比原始数据的平均分类错误率下降0.033 9,整体分类错误率明显下降。

4 结束语

依据趋势转折点在时间序列数据中的重要特征,本文提出一种基于数据自身趋势特征提取数据转折点的算法(FTTO)。该算法计算简单,易于实现。通过在UCR时间序列分类数据集上与SEEP、CAP和PAA算法对比实验可以看出,该算法在还原原始序列数据时的拟合误差较小,具有去除干扰噪音数据的功能。实验结果表明,采用1-NN分类和DTW算法,与SEEP、CAP算法拟合的数据和Keogh团队公布的原始数据分类错误率相比,FTTO算法整体分类错误率最小。

参考文献

- [1] ESLING P, AGON C. Time-series Data Mining [J]. ACM Computing Surveys, 2012, 45(1): 11-12.
- [2] HU Zhikun, XU Fei, GUI Weihua, et al. Wavelet Matrix Transform for Time-series Similarity Measurement [J]. Journal of Central South University of Technology, 2009, 16(5): 802-806.
- [3] AGRWAL R, FALOUTSOS C, SWAMI A. Efficient Similarity Search in Sequence Databases [C]// Proceedings of the 4th International Conference on Foundations of Data Organization and Algorithms. London, UK: Springer-Verlag, 1993: 69-84.
- [4] RAFIEI D, MENDELZON A O. Querying Time Series Data Based on Similarity [J]. IEEE Transactions on Knowledge & Data Engineering, 2000, 12(5): 675-693.
- [5] REN J M. Similarity Search over Time Series Data Using Wavelets [C]// Proceedings of the 18th International Conference on Data Engineering. San Jose, USA: IEEE Computer Society Press, 2002: 212-221.
- [6] CHAN K, FU A. Efficient Time Series Matching by Wavelets [C]// Proceedings of the 15th IEEE International Conference on Data Engineering. Washington D. C., USA: IEEE Press, 1999: 126-133.
- [7] KEOGH E J, CHU S, HART D, et al. An Online Algorithm for Segmenting Time Series [C]// Proceedings of IEEE International Conference on Data Mining. Washington D. C., USA: IEEE Press, 2002: 289-298.

(下转第68页)

参考文献

- [1] RAHIMI S M, WANG X. Location Recommendation Based on Periodicity of Human Activities and Location Categories[M]//PEI J, TSENG V S, CAO L. Advances in Knowledge Discovery and Data Mining. Berlin, Germany: Springer, 2013: 377-389.
- [2] JIE B, YU Z, WILKIE D, et al. A Survey on Recommendations in Location-based Social Networks[J]. GeoInformatica, 2015, 19(3): 525-565.
- [3] ZHANG J, CHOWMEMBER C, LI Y. iGeoRec: A Personalized and Efficient Geographical Location Recommendation Framework[J]. IEEE Transactions on Services Computing, 2015, 8(5): 701-714.
- [4] ZHANG J D, CHOW C Y, LI Y. LORE: Exploiting Sequential Influence for Location Recommendations[C]//Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. New York, USA: ACM Press, 2014: 103-112.
- [5] SADILEK A, KAUTZ H, BIGHAM J P. Finding Your Friends and Following Them to Where You Are[C]//Proceedings of the 5th International Conference on Web Search and Web Data Mining. Seattle, USA: DBLP, 2012: 723-732.
- [6] YUAN Q, CONG G, MA Z, et al. Who, Where, When and What: Discover Spatio-Temporal Topics for Twitter Users[C]//Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York, USA: ACM Press, 2013: 605-613.
- [7] CHENG C, YANG H, KING I, et al. Fused Matrix Factorization with Geographical and Social Influence in Location-based Social Networks[C]//Proceedings of the 26th AAAI Conference on Artificial Intelligence. Toronto, Canada: AAAI Press, 2012: 17-23.
- [8] JIA Y, WANG Y, JIN X, et al. Location Prediction: A Temporal-Spatial Bayesian Model[J]. ACM Transactions on Intelligent Systems & Technology, 2016, 7(3): 31.
- [9] LIU Y, WEI W, SUN A, et al. Exploiting Geographical Neighborhood Characteristics for Location Recommendation[C]//Proceedings of the 23rd ACM International Conference on Information and Knowledge Management. New York, USA: ACM Press, 2014: 739-748.
- [10] LIAN D, ZHAO C, XIE X, et al. GeoMF: Joint Geographical Modeling and Matrix Factorization for Point-of-Interest Recommendation[C]//Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York, USA: ACM Press, 2014: 831-840.
- [11] ZHANG S, REN K. User Preferences-based and Time-sensitive Location Recommendation Using Check-in Data[J]. Journal of Computer & Communications, 2015, 3(9): 18-27.
- [12] BAO J, ZHENG Y, MOKBEL M F. Location-based and Preference-aware Recommendation Using Sparse Geosocial Networking Data[C]//Proceedings of the 20th International Conference on Advances in Geographic Information Systems. New York, USA: ACM Press, 2012: 199-208.
- [13] CHO E, MYERS S A, LESKOVEC J. Friendship and Mobility: User Movement in Location-based Social Networks[C]//Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. San Diego, USA: DBLP, 2011: 1082-1090.
- [14] ZHONG Y, YUAN N J, ZHONG W, et al. You Are Where You Go: Inferring Demographic Attributes from Location Check-ins[C]//Proceedings of the 8th ACM International Conference on Web Search and Data Mining. New York, USA: ACM Press, 2015: 295-304.
- [15] ZHENG V W, CAO B, ZHENG Y, et al. Collaborative Filtering Meets Mobile Recommendation: A User-centered Approach[C]//Proceedings of 24th AAAI Conference on Artificial Intelligence. Atlanta, USA: AAAI, 2010: 236-241.
- [16] 陈芸, 董西伟, 荆晓远. 联合混合范数约束和增量非负矩阵分解的目标跟踪[J]. 计算机工程, 2015, 41(12): 260-264.
- [17] 乔少杰, 金琨, 韩楠, 等. GMTP: 基于高斯混合模型的轨迹预测算法[J]. 软件学报, 2015, 26(5): 1048-1049.
- [18] 冀续烨, 陈明, 冯国富, 等. 一种多模型协同的目标提取方法[J]. 计算机工程, 2015, 41(5): 254-258.

编辑 金胡考

(上接第 61 页)

- [8] KEOGH E. Exact Indexing of Dynamic Time Warping[C]//Proceedings of the 28th International Conference on Very Large Databases. Hong Kong, China: [s. n.], 2002: 406-417.
- [9] 潘定, 沈钧毅. 时态数据挖掘的相似性发现技术[J]. 软件学报, 2007, 18(2): 246-258.
- [10] YI B K, FALOUTSOS C. Fast Time Sequence Indexing for Arbitrary LP Norms[C]//Proceedings of the 26th International Conference on Very Large Data Bases. [S. l.]: Morgan Kaufmann Publishers Inc., 2000: 385-394.
- [11] KEOGH E, CHAKRABARTI K, PAZZANI M, et al. Dimensionality Reduction for fast Similarity Search in Large Time Series Databases[J]. Journal of Knowledge and Information System, 2001, 3(3): 263-286.
- [12] PRAT K B, FINK E. Search for Patterns in Compressed Time Series[J]. International Journal of Image and Graphics, 2002, 2(1): 89-106.
- [13] 詹艳艳, 徐荣聪, 陈晓云. 基于斜率提取边缘点的时间序列分段线性表示方法[J]. 计算机科学, 2006, 33(11): 139-142.
- [14] 林意, 王智博. 基于一阶滤波的时间序列分段线性表示方法[J]. 计算机工程, 2016, 42(9): 151-157.
- [15] 周大镛, 李敏强. 基于序列重要点的时间序列分割[J]. 计算机工程, 2008, 34(23): 14-16.
- [16] 张娜, 李莉娟. 时间序列分段线性表示的几种算法比较[J]. 中国西部科技, 2009(14): 80-81.
- [17] CHEN Yanping, KEOGH E. UCR Time Series Classification Archive[EB/OL]. [2016-08-07]. http://www.cs.ucr.edu/~eamonn/time_series_data/.

编辑 索书志