基于时间序列分段的气象数据压缩算法①

程 敏[1,2], 翁宁泉[1,2,3], 刘 庆[1], 孙 刚[1], 陈小威[1,2]

[1](中国科学院安徽光学精密机械研究所 中国科学院大气成分与光学重点实验室, 合肥 230031)
[2](中国科学院大学, 北京 100049)
[3](中国科学技术大学 环境科学与光电技术学院, 合肥 230022)

摘 要: 直接采用风速、温湿压等气象参数原始时间序列对其进行短期预测、相似匹配、分类聚类等数据挖掘工作不但效率低下, 而且会影响时间序列数据挖掘的准确性和可靠性. 提出了一种简单快速的基于特征点的筛选算法对时间序列进行分段线性表示. 对气象参数等时间序列进行实验, 并就计算性能和拟合误差与另外一种序列分段算法进行了对比分析, 结果表明该方法能有效地提取序列的主要形态, 同时降低对于阈值的依赖, 具有计算代价小、快速方便、通用性强等特点, 在气象数据压缩上具有较好的应用前景.

关键词: 时间序列; 模式表示; 特征点; 数据压缩

Meteorological Data Compression Algorithm Based on Time-Series Segmentation

CHENG Min[1,2], WENG Ning-Quan[1,2,3], LIU Qing[1], SUN Gang[1], CHEN Xiao-Wei[1,2]

[1](Key Laboratory of Atmospheric Composition and Optical Radiation, Anhui Institute of Optics and Fine Mechanics, Chinese Academy of Sciences, Hefei 230031, China)
[2](University of Chinese Academy of Sciences, Beijing 100049, China)
[3](Institute of Environmental Science and Photoelectric Technology, University of Science and Technology of China, Hefei 230022, China)

Abstract: It is not only inefficient to use the raw time series of meteorological parameter such as temperature refractive index structure parameter, wind speed and temperature to make short-term prediction, query similarity and classify and cluster time series, but also affects accuracy and reliability of data mining of time series. This article proposes a simple and fast method which based on the election of extrema point and tendency turning point to make the piecewise linear representation of time series. The method can extract the main pattern of series effectively, and reduce the dependency of threshold. It has the characteristic of small cost of computing, efficient and convenient and strong commonality. Then based on that, the experiments on temperature refractive index structure parameter and other kinds of meteorological parameter are implemented and conduct the comparison analysis between the method and another kind of sequence segmentation algorithm. The result shows that the method proposed is capable of reflecting the pattern of time series effectively and accurately.

Key words: time series; piecewise linear representation; feature point; data compressing

0 引言

时间序列广泛存在于各个领域, 如气象领域中实验观测值、金融领域股票价格波动和医学监测数据等, 这些数据反映了观测量随时间的状态变化及其他隐含的重要信息. 大部分时间序列数据维数高, 数据量大, 且常常有噪声干扰, 直接对原序列进行分析挖掘会消耗大量的计算时间和存储空间, 而且会影响算法的准确性和可靠性, 因此对时间序列进行特征提取或模式表示具有重要的研究意义. 模式表示是用原序列的特征点来刻画整个序列的主要形态而不计细节上的差异. 时间序列的模式表示有四个方面的优点: 首先是对原序列进行压缩, 大大减小了存储和后续计算的代

① 基金项目:国家自然科学基金(41375017,41205023)
收稿时间:2013-12-10;收到修改稿时间:2014-01-14

价; 第二是去除了细节干扰而只保留原序列的主体特征, 更鲜明地反映时间序列的自身特点, 有利于提高后续数据挖掘算法的效率和准确性; 此外还可以控制序列的压缩度来实现不同精度层次上的搜索和匹配; 最后, 多数应用领域只关心待观测量在某时间段内的变化模式或规律, 而不是序列中具体某个点的值, 序列的模式表示更符合其需求.

1 相关研究

时间序列的模式表示方法主要有四类方法, 即分段线性法、频域法、奇异值分解法和符号表示法. 其中, 分段线性表示(Piecewise Linear Representation, PLR)因其直观高效的特点一直是时间序列模式表示方法中研究最多的方法之一. 分段线性表示的基本思想是用 M 条首尾相接的线段来近似表示一条长度为 N(N>>M)的时间序列. Keogh 在文献[1]中引入了目标函数, 使用目标函数来控制原始序列和其线性近似表示之间的残差平方和最小. 实验表明该方法能有效地对时间序列进行压缩, 因而在时间序列数据挖掘中得到大量的应用. 随后提出的时间序列分段聚集近似[2](Piecewise Aggregate Approximation, PAA), 将时间序列等宽度划分, 每个子段用时间序列在该子段上的平均值来表示. 近些年国内的研究人员也提出了很多新的 PLR 方法, 主要有基于斜率的[3,4]、基于三角极值的[5]、基于函数的[6]、基于特征点的[7]、基于趋势转折点的[8] 和基于重要点的 PLR 方法[9,10]等. 这些研究方法都表明, 在进行分段线性表示原始时间序列数据时, 原序列中的一些重要数据点是必须被保留的, 如边缘点、极值点以及变化趋势发生改变的拐点, 这些数据点保存着时间序列变化的主要特征模式[8]. 但是目前的 PLR 算法例如自顶向下 TD 算法、自底向上 BU 算法、滑动窗口 SW 算法等都依赖于一个阈值 R 的选取, 而 R 的确定往往是很困难的, 且算法不具有通用性, 一些局部算法如局部极值法又容易造成划分过于精确而忽略了趋势.

本文提出了一种基于特征点的选择分段算法, 对构成特征点的部分边缘极值点和趋势点进行筛选实现对原始时间序列的分段线性表示. 该算法不再依赖于阈值, 降低了输入的难度, 计算简单, 分段效果好, 同时支持在线划分, 在模式表示方面具有一定的优越性.

2 基于特征点选择的时间序列分段线性表示

2.1 时间序列特征点和特征点序列

时间序列的每条记录可以抽象为一个二元组. 其中 t 为时间变量, x 为数据变量, 反映数据单元的实际意义, 例如股票价格、实验观测值等. 由此, 时间序列可以给出如下定义:

定义 1.

时间序列 X 是一个有限集, 即: $X =< (t_1, x_1), (t_2, x_2), ..., (t_n, x_n) >$ 满足: $t_i < t_{i+1}, (i = 1, 2, 3, ..., n-1)$

通常情况下时间序列的间隔时间相等, 因此可以将时间序列表示为:

$$X =< x_1, x_2, x_3, ..., x_n >$$

时间序列中的极值点和趋势转折点(即拐点)统称为特征点, 特征点的左右序列的主变化趋势不同. 用以表示时间序列的特征点定义如下:

定义 2.

对于任意一个序列$(x_1, x_2, x_3, ..., x_n)$, 当 1<<n 时, 如果某个点满足以下条件:

1) $x_i > x_{i-1}$ 且 x_i 或 $\geq x_{i+1}$ 或 $x_i \geq x_{i-1}$ 且 $x_i > x_{i+1}, 1 \leq i \leq n$;

2) $x_i < x_{i-1}$ 且 $x_i \leq x_{i+1}$ 或 $x_i \leq x_{i-1}$ 且 $x_i < x_{i+1}, 1 \leq i \leq n$;

3) x_i 为非极值点, 且 $0<(x_{i+1} - x_i)/(x_i - x_{i-1})<1/K$ 或 $(x_{i+1} - x_i)/(x_i - x_{i-1})>K$, $K \in N$;

则 x_i 为该序列的特征点.

由条件 1)和条件 2)确定的是所有极值点和部分趋势转折点, 具体如下图 1 所示:

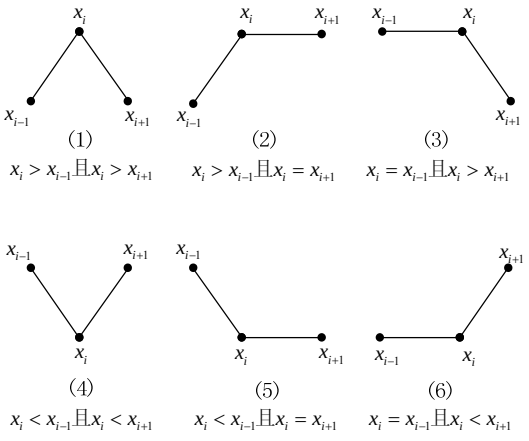


图 1　极值点和部分趋势转折点示意图

由条件 3)可以得到另一部分趋势转折点, 如图 2 所示:

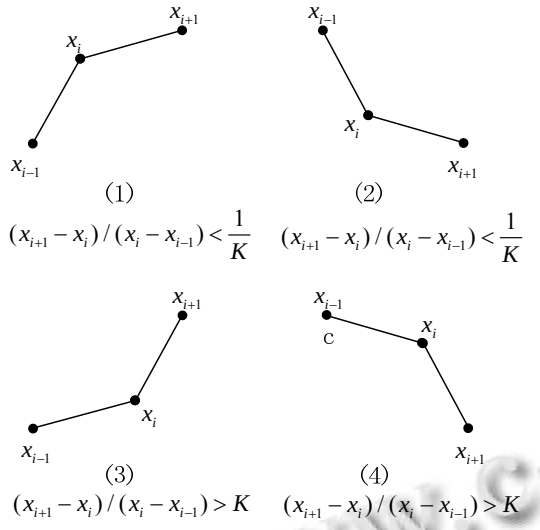


图 2 部分趋势转折点示意图

由此合并可得到特征点序列 P. 如果直接用这些特征点来表示时间序列, 由于局部特征点过多, 容易造成模式表示的过度精确而降低压缩效果, 所以要对这些特征点筛选.

筛选算法采用的是滑动时间窗口, 即对于特征点序列 $P<x_{p_1},x_{p_2},...,x_{p_m}>$, 若 $j\in[1,K]$, x_{p_i-j}、x_{p_i+j} 为非特征点, $p_1\le p_i-j\le p_m$, $p_1\le p_i+j\le p_m$, $K\in N$, 则 x_{p_i} 为满足条件的特征点, K 为一个指定的整数, 表示时间窗口大小, 由输入来指定.

2.2 基于特征点选择算法的分段线性表示

基于特征点选择算法的分段线性表示(PLR Based on Elective Feature Point)在选择特征点时分三个步骤. 使用 Python 作为主要编程语言, 具体步骤如下:

S1: 输入整数 K 和原始序列 A, 同时新建一个列表 L, 遍历原始序列, 依据 3.1 中 1)、2)、3)条件, 得到特征点的下标序列

S2: 根据时间窗口 K, 遍历下标序列 L 并以 j 滑动判断条件 L[i]-j 且 L[i]+j 在 L 中是否成立,若成立则将 L[i]从 L 中删除, 继续运行直到列表 L 的长度不再改变

S3: 由 S2 得到特征点序列 A[L[t]], 输出.

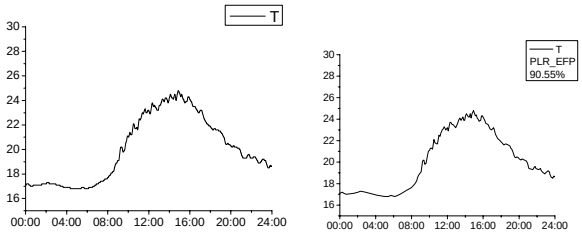
```
L=[]//新建下标列表
val_1=float (A[Index][1])-float (A [Index] [0])
for tab in range (line_count-1):
    val_2=float(A[Index][tab+1])-float(A[Index][tab])
    if val_1==0 and val_2==0:
        continue
//极值点和趋势转折点
    if (val_1>0 and val_2<=0) or (val_1>=0 and val_2<0) or (val_1<0 and val_2>=0) or (val_1<=0 and val_2>0):
        L.append (tab)
//趋势转折点
    elif 0<abs (val_2/val_1) <1/float (K) or K<abs (val_2/val_1):
        L.append (tab)
    val_1=val_2
//边界点
L.insert (0, 0)
L.insert (len (L), line_count-1)
//对特征点序列 L 进行筛选
tab=0
while tab<len(L)-2:
    if L [tab+1]-L[tab] <=K and L [tab+1]-L [tab+2] <=K:
        L.remove (L [tab+1])
        tab=tab-1
    tab=tab+1
```

3 仿真实验和对比分析

3.1 PLR_EFP 的适应能力

实验数据选取三个不同领域的数据, 即气象领域的 T(Temperature)、WS(Wind_Speed)、 (Refractive Index Structure Constant) 和 Video 以及 ECG(Electrocardiograph)资料,来探讨其适应能力.

I) 对于长度为 1440 的温度数据 T, 原序列曲线和拟合后的序列曲线如图 3 所示.

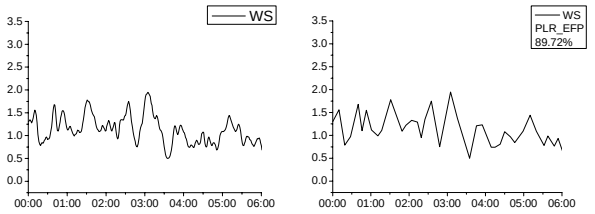


(a) T 原始序列曲线 (b) T 拟合后序列曲线

图 3 长度为 1440 的温度数据 T 曲线

图 3(a)为原序列的 24 小时曲线, 每分钟有一个数据点; 拟合后的序列曲线如图 3(b)的曲线, 序列约简后的长度为 136 个点, 拟合误差为 0.0024.

II) 对于长度为 360 的风速数据 WS, 原序列曲线和拟合后的时间序列曲线如图 4 所示.

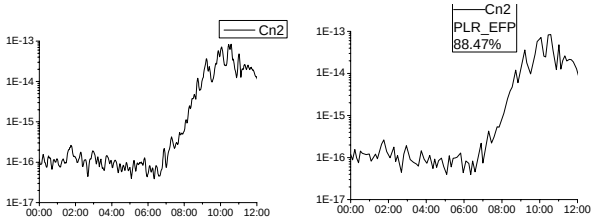


(a) WS 原始序列曲线　(b) WS 拟合后序列曲线

图 4　长度为 360 的风速 WS 曲线

图 4(a)为长度为 360 的原序列曲线; 拟合后的序列曲线如图 4(b)所示, 序列约简后的长度为 37 个点, 拟合误差为 0.0685.

III) 对于长度为 720 的 C_n^2 数据, 原序列曲线和拟合后的序列曲线如图 5 所示.

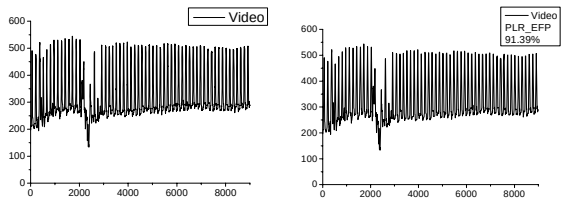


(a) C_n^2 原始序列曲线　(b) C_n^2 拟合后序列曲线

图 5　长度为 720 的 C_n^2 曲线

图 5(a)为原序列曲线; 图 5(b) 为拟合后的序列曲线, 序列约简后的长度为 83 个点. C_n^2 的量级比较小, 通常在 10-15 左右, 因此计算时取对数, 得到拟合误差为 0.00325.

IV) 对于长度为 9002 的 Video 数据, 原序列曲线和拟合后的序列曲线如图 6 所示.



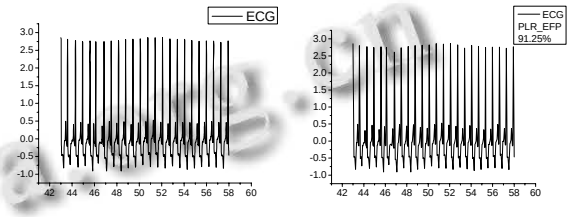
(a) Video 原始序列曲线 (b) Video 拟合后序列曲线

图 6　长度为 9002 的 Video 曲线

图 6(a)为长度为 9002 的原序列曲线; 拟合后的序列曲线为图 6(b)的曲线, 序列约简后的长度为 727 个点, 拟合误差为 0.0348.

V) 对于长度为 3750 的 ECG 数据, 原序列曲线和拟合后的序列曲线如图 7 所示.

图 7(a)为长度为 3750 的原序列曲线; 拟合后的序列曲线如图 7(b)的曲线, 序列约简后的长度为 328 个点, 拟合误差为 1.32.



(a)ECG 原始序列曲线　(b)ECG 拟合后序列曲线

图 7　长度为 3750 的 ECG 曲线

再选取 2006 年 09 月 11 日到 2006 年 11 月 02 日的共 53 个数据集合, 考察 WS、T 和的平均拟合误差, 结果如下:

数据集	平均压缩度	平均拟合误差
Temperature	89.34%	0.00209
Wind Speed	88.51%	0.0870
C_n^2	88.07%	0.00360

图 8　多个数据集的平均拟合误差

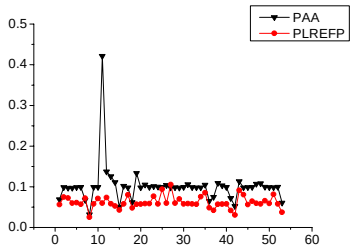
从上述结果可以看出, 本算法提取出的数据点可以很好地对气象领域的原始时间序列进行特征表示, 同时也适用于其他领域. 由于是对特征点进行筛选, 所以该算法是支持在线划分的, 此外特征点在获取和选择过程中只需一个整数 K (上述实验中 K 均取 5), 而不像其他算法中需要提供一个比较复杂的阈值 R, 因此降低了对于输入阈值的依赖, 同时也提高了算法的通用性.

3.2 对比分析

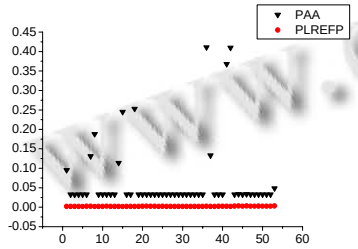
选择 PAA 分段算法来进行对比分析. 在 PAA 算法中, 用固定长度的时间窗口分割时间序列, 每个窗口内的时间序列值用窗口内数据值的平均值来表示, 输入的参数为要分段的段数 M 或窗口的长度 W. 由于 PAA 算法的距离度量灵活, 以及其计算性能和分段效果的优势, 使 PAA 成为常被比较的参照算法之一, 所

以本文也选择了 PAA 算法作为比较.

选取 0911 到 1102 的 53 个数据集, 对温度时间序列进行实验, 在相同压缩比下对 PAA 和 PLR_EFP 算法的计算代价和拟合误差进行对比分析:



(a)两种算法计算代价对比



(b)两种算法拟合误差对比

图 9 PAA 和 PLR_EFP 算法效果对比图

从图 9 的对比中可知, 在相同压缩比下, PLR_EFP 的计算代价和拟合误差要比 PAA 小, 计算的时间代价只约为 PAA 的一半, 而拟合误差小一个数量级以上, 同时, 从变化趋势来看, PLR_EFP 算法更平稳, 变化趋势较明确, 而 PAA 算法的拟合误差具有明显的抖动现象. 从拟合误差和计算性能来看, 本算法在模式表示方面具有一定的优势.

4 结论

直接采用原始时间序列来对其进行相关数据挖掘工作既复杂又费时, 同时会降低挖掘算法的准确性, 因此采用某种模式表示方法来提取刻画原始时间序列主要形态的数据点显得十分必要. 大部分时间序列分段算法需要输入不容易确定的阈值来控制压缩度或拟合误差, 同时算法比较复杂. 本文提出了一种特征点的选择算法, 应用到不同时间序列分段中, 同时基于观测实验所获得的大量气象数据集, 就不同的参数值分别与 PAA 算法进行了对比. 实验结果表明该算法的拟合误差和计算性能要整体优于 PAA 算法, 同时该算法计算简单快速, 不依赖于较复杂的阈值, 在时间序列的模式表示方面具有一定的代表性.

参考文献

1 Keogh E. A fast and robust method for pattern matching in time series databases. Proc. of the 9th International Conference on Tools with Artificial Intelligence. 1997. 578–584.

2 Keogh E, Chakrabarti K, Pazzani M. Dimensionality reduction for fast similarity search in large time series databases. Journal of Knowledge and Information Systems, 2000, 3(3): 263–286.

3 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.计算机科学,2006,33(11):139–142,161.

4 刘贺红,张毅坤.确定时间序列分段点的方法研究.计算机工程与应用,2010,46(13):44–46.

5 戴爱明,高学东.时间序列三角极值点线性分段算法.南昌航空大学学报,2009,23(2):25–28,41.

6 谢福鼎,王赫楠,张永,孙岩.基于函数的时间序列分段线性表示方法.计算机科学,2011,38(11):153–155,160.

7 喻高瞻,彭宏,胡劲松,等.时间序列的分段线性表示.计算机应用与软件,2007,24(12):17–18.

8 尚福华,孙达辰.基于时间序列趋势转折点的分段线性表示.计算机应用研究,2010,27(6):2075–2077,2092.

9 周黔,吴铁军.基于重要点的时间序列趋势特征提取方法.浙江大学学报(工学版),2007,41(11):1782–1787.

10 陈然,戴齐.基于重要点的时间序列固定分段数分段算法.计算机技术与发展,2011,21(9):103–106.