基于重要点的时间序列固定分段数分段算法

陈 然，戴 齐
（西南交通大学 信息科学与技术学院，四川 成都 610031）

摘 要：基于重要点探测技术的时间序列线性分段算法能较好地保留序列的全局特征和拟合高精确度。传统的基于重要点时间序列分段算法，只能通过误差阈值来控制分段，该方法不能预计分段数量，不能适应后期要求分段数量一定的应用。提出一种基于序列重要点的时间序列固定分段数的分段算法—PLR_FPIP，该方法借用二叉树层次遍历的思路，重新调整原方法的分段次序，使用重要点组成的直线段近似描述时间序列，该方法能够在分段数量一定的情况下对时间序列分段。实验证明，该分段算法能在固定分段数的情况下反映时间序列的主体特征，算法简单快速，整体拟合误差小。

关键词：时间序列；固定分段数；序列重要点；时间序列分割
中图分类号：TP311 **文献标识码**：A **文章编号**：1673-629X(2011)09-0103-04

Time Series Segmentation Based on Fixed Number of PIPs Detection

CHEN Ran, DAI Qi
(School of Information Science & Technology, Southwest Jiaotong University, Chengdu 610031, China)

Abstract: The algorithm of time series segmentation based on the series important points can better retain global characteristics of series and fitting high accuracy. The traditional segmentation algorithm chooses segment point only through error threshold but fixed number of subsection. It can not meet the application which requires fix segment number. It proposes an algorithm based on fixed number of PIPs detection(PLR_FPIP), which uses the ideas of binary tree level traversal, re-adjust the order of the original method and use PIPs composed of straight time series. Experimental results show that this algorithm can reflect the main characteristics of time series in cases of fixed number of PIPs, and the algorithm is simple, fast, low total error.

Key words: time series; fixed number of series; series important point; time series segmentation

0 引 言

时间序列数据是一种在金融、气象等许多领域都普遍存在的重要数据。由于时间序列数据所具有的海量性、复杂性以及连续数值形式的表示方式，直接在原始数据上进行数据挖掘往往不能取得良好的效果。为此，一般需将连续数值形式表示的时间序列转换成离散的、相对抽象的符号序列[1]。Keogh E 等认为分段线性算法可以概括为：输入原始时间序列、输出首尾相连的直线段的算法[2]，并在文献[3]中阐述了该方法所具有的低索引结构维数、计算速度快等特点，因此被研究者认为是比较先进的时间序列表示方法。近几年来，国内外的数据挖掘研究者对时间序列的分段线性表示法进行大量深入的研究，也提出许多新的分段思路和方法。

收稿日期：2011-02-22；**修回日期**：2011-05-27
作者简介：陈 然(1984-)，男，四川成都人，硕士研究生，主要研究领域为人工智能、数据挖掘等；戴 齐，副教授，主要研究领域为人工智能、智能信息处理、数据挖掘、MIS、智能机器人等。

1 相关工作

线性分段算法的原理是使用一些连续的、首尾相连的直线段来近似地表示原始时间序列，直线段的数量决定了分段拟合的精度，直线段越多，拟合精度相对越高，反之则越低。文献[4]中阐述了时间序列线性分段算法可以根据分段策略的不同分为两种类型，一种是限制分段数 K 的分段算法，即将给定时间序列转换为只包括 K 个直线段的近似序列，但是不能控制每一子段和全段的误差。该类型中具有代表意义的算法是文献[2,5]都分别独立提出的时间序列分段集近似 PLR_PAA 方法。该方法先将时间序列分成 n 个宽度相等的子序列，再用每个子段平均值所组成的序列来近似表示原始序列。文献[6]证明了 PLR_PAA 方法不仅支持任意长度的时间序列相似查询和所有的 Minkowski 相似度量和加权欧氏距离，还能通过使用索引来提高查询的效率。随后，文献[7]提出了比 PLR_PAA 更为准确合理的 PLR_APCA。PLR_APCA 方法和 PLR_PAA 方法的不同处在于对时间序列采用

不等宽度分割。另一种分段类型是只通过控制分段误差来选择合适的分段点,不预先规定分段的数目。这种方法又可以细分为两种:其一使用局部阈值来控制单个分段,让当前子段的误差不超过该局部阈值,其二是使用全局阈值,让所有分段的误差和不超过该阈值。文献[4]介绍了 3 种该类型的具有代表意义线性分段算法:即滑动窗口(SW)、自顶向下(TD)、自底向上(BU)。其中,SW 支持时间序列的在线分段,但分段效果一般[8],而且不支持保留分段历史信息以及二次拟合[9],相比之下,TD 和 BU 算法尽管分段效果较好,但不支持对时间序列进行在线分段[10],而且算法空间复杂度较高。此外,文献[11]提出了一种基于时态边缘算子的时间序列分段算法,文献[12]提出一种基于斜率提取边缘点的时间序列分段算法。文献[13]在此基础上提出了一种基于层次聚类的时间序列在线分割算法(OSHC)。

2 问题描述以及定义

2.1 时间序列的线性分段表示

时间序列可以概括为由记录时间和记录值组成的元素的有序集合,记为:$X = \{v_1, v_2, \cdots, v_n\}$,其中元素 v_i 表示时间序列在 t_i 时刻的记录值。直观上讲,时间序列的分段线性表示方法就是用 k 条首尾邻接的直线段来近似表示一条长度为 $n(n > k)$ 的时间序列,由于直线段数量远远小于时间序列长度,因此表示方法使得数据存储、变换和计算等更加高效。

2.2 时间序列重要点探测技术

重要点探测技术最早在文献[14]中,由 F. L. Chung 等提出,在时间序列分段算法中可用该方法生成一连串重要点序列作为时间序列的分段点,从而达到线性拟合的目的,重要点即是时间序列中 dis 最大的点,dis 见公式(1),如图 1 所示,对于整段序列来说,点 pc 是距离序列首尾连线垂直距离最大的点(文献[15]中,周大镯等已经证明使用垂直距离相对于使用正交距离的优越性),所以 pc 就是序列重要点。

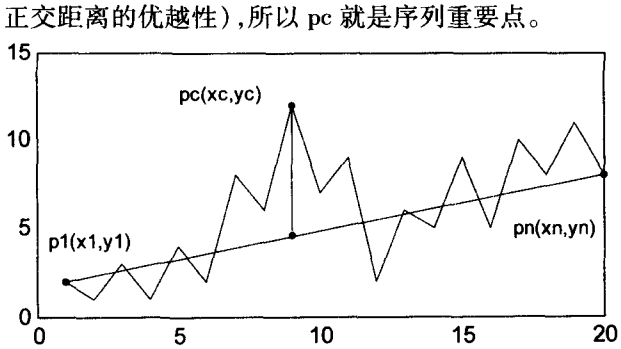

图 1 基于重要点的探测技术

$$dis = \left| \frac{y_1 + (y_n - y_1) \times (x_i - x_1)}{(x_n - x_1)} - y_i \right| \quad (1)$$

2.3 基于重要点探测技术的时间序列分段算法

基于重要点探测技术的时间序列分段算法 PLR_PIP 可以有效地控制分段的全局拟合误差,该算法首先找出时间序列的重要点,并以此点为首次分段点将原序列分为两个子段,然后运用递归的思路分别找寻两个子段的重要点并分割,以此往复直到该子段的拟合误差小于一个阈值 e 为止,基于重要点探测技术的时间序列分段算法描述如下:

输入:时间序列 $X = < x_1, x_2, \cdots, x_i, \cdots, x_n >$ 和误差值阈值 e

输出:序列重要点 PIP[ ]

```
PLR_PIP( stpos, endpos, X, e )
Begin
CalculatePip ( stpos, endpos ); //计算从 stpos 到 endpos 的重要点坐标 segpos,然后返回累计误差 total_err
If total_err > e
PIP←segpos; //将 segpos 移入 PIP
PLR_PIP( stpos, segpos, X, e );
PLR_PIP( segpos, endpos, X, e );
End
Return PIP;
End
```

值得注意的是在调用该算法之前需要初始化 PIP 为空,并将初始点 stpos 和终止点 endpos 加入到 PIP 中。由于 PLR_PIP 算法的递归特性,程序总是选择将最左端的子序列进行分割,直到不能继续分割为止,才会返回上一级,这种分割方法造成的一个错误是每次选择需要被分割的子段总是强行选择最左端的子段,而不是选择最"需要"被分割的子段。第一次分段后,虽然右边的子段更应该被分段,但是由于递归的缘故,还是选择了将左边的子段一直分割下去。这样的分割流程和二叉树的先续遍历十分类似,从而不能通过计数器的方式来控制分段数。

3 基于重要点的时间序列固定分段数分段算法

输入:时间序列 $X = < x_1, x_2, \cdots, x_i, \cdots, x_n >$ 和误差值 e 以及分段数 k

输出:序列重要点 FPIP

```
PLR_FPIP( X,e,k )
Begin
FPIP=[ ];//初始化 FPIP
FPIP←stpos;//初始点 stpos 加入到 FPIP
FPIP←endpos;//终止点 endpos 加入到 FPIP
InitQueue( Q );//初始化队列
```

```
InQueue(stpos,Q);// 初始点 stpos 入队
InQueue(endpos,Q);// 终止点 endpos 入队
While ! EmptyQueue And FPIP not full
stpos= DeQueue(Q); //序列起始点
endpos= DeQueue(Q); //序列结束点
CalculatePip(stpos, endpos); //计算从 stpos 到 endpos 的重要点坐标 segpos,然后返回累计误差 total_err
if total_err > e //对累计误差大于 e 的直线段进行分段
FPIP←segpos;//将 segpos 放入分段点集合
Left_err=Calerr(stpos,segpos); //计算从 stpos 到 segpos 的拟合误差
Right_err=Calerr(segpos,endpos); //计算从 segpos 到 endpos 的拟合误差
If Left_err > Right_err
InQueue(stpos,Q);
InQueue(segpos,Q);
InQueue(segpos,Q);
InQueue(endpos,Q);
Else
InQueue(segpos,Q);
InQueue(endpos,Q);
InQueue(stpos,Q);
InQueue(segpos,Q);
End
End
End
Return FPIP;
End
```

其中 FPIP 为存储分段点的一维数组,Q 是一个用于存贮分段点的队列,CalculatePip(stpos,endpos)是用于计算 stpos 到 endpos 的重要点坐标,选出这段时间序列中 dis 最大的点。Left_err = Calerr(stpos,segpos)和 Right_err=Calerr(segpos,endpos)则是计算由分段点分成的两段子序列的拟合度,拟合度大的子序列(序列首尾)先入队列。该算法原理类似于二叉树的层次遍历,固定分段数的分段算法要求对序列每一次的重要点选取都保证在全局是当前序列中最突出的点,直到"填满"长度固定的 FPIP 数组,采用队列存储当前分割点正好可以满足分段粒度由"粗"到"细"的分段要求。

4　实验结果

4.1　拟合结果

基于重要点的时间序列固定分段数分段算法输入参数为分段误差阈值 e 以及分段数 k,算法实现采用 Winxp + vc6.0,运行得到分段点后采用 matlab7.0 绘制图像,实验环境为:CPU 2.4GHz,内存 2GB,硬盘 320GB,操作系统 Windows Xp sp3.0。试验数据集采用 Series B. IBM Common Stock Closing Prices, Daily, May 17, 1961 to Nov. 2, 1962 .txt,该数据集数据长度为 369,最小分段误差选取 e =1000,原始时间序列如图 2 所示,分段线性拟合后如图 3(50 段,压缩比 369/50)所示,线性拟合结果可以较好地反映原序列的主题特征。

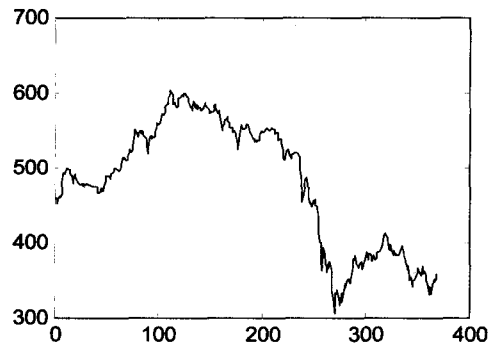

图 2　原时间序列

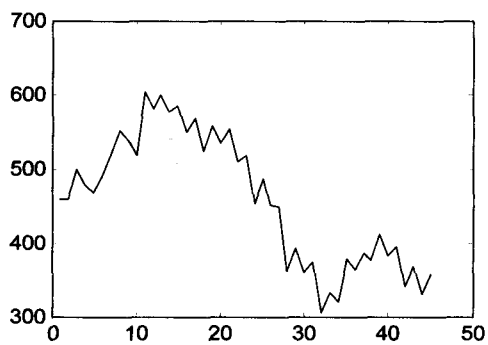

图 3　线性拟合(50 段)

4.2　算法耗时比较

在相同分段数的情况下比较 PLR_PAA 和 PLR_FPIP 这两种同样是固定分段数的分段算法性能,如图 4 所示,随着时间序列分段数的增加,从整体上看,PLR_FPIP 的 CPU 耗时略微高于 PLR_PAA,且线性增长的幅度比 PLR_PAA 算法微快。

4.3　算法分段拟合误差比较

在相同分段数的情况下比较 PLR_PAA 和 PLR_FPIP 这两种算法的拟合误差,如图 5 所示,随着分段数的增加,PLR_FPIP 方法的拟合误差明显小于 PLR_PAA。

5　结束语

本算法能在固定分段点的情况下使用重要点探测技术对时间序列进行分段,分割效果好、效率高,能很好地描述时间序列的整体特征。在选择重要点集合的过程中,从全局角度出发,始终坚持选择相对更需要被

分割的子段作为分割对象，避免了问题陷入局部最优的困境。通过实验证明，与同为固定分段数的时间序列分段算法 PLR_PAA 方法比较，虽然计算耗时略高，但是 PLR_FPIP 算法的拟合误差也相对较小，拟合精度更高。

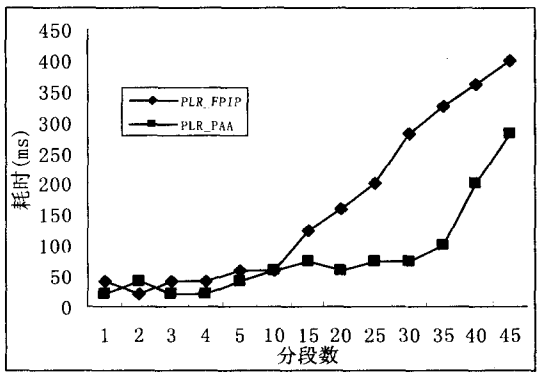

图 4 算法耗时比较

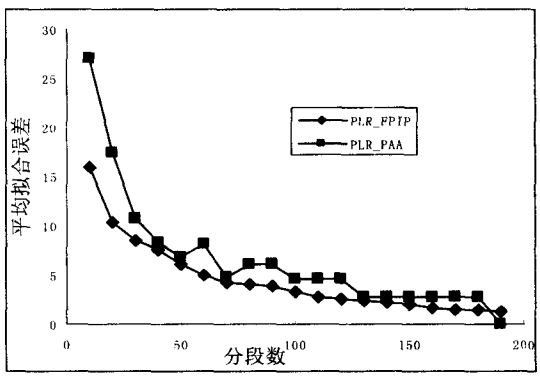

图 5 拟合误差比较

参考文献：

[1] 张 军，马志民. 基于时间序列的相似子模式发现算法[J]. 计算机技术与发展，2006，16(1)：140-141.

[2] Keogh E，Chakrabarti K，Pazzani M J，et al. Dimensionality reduction for fast similarity search in large time series databases [J]. Knowledge and Information Systems，2008，3(3)：263-286.

[3] Keogh E，Kasetty S. On the need for time series data mining be-nchmarks：A survey and empirical demonstration[C]// Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. [s. l.]：[s. n.]，2002：102-111.

[4] Tewari G，Snyder J，Sander P V，et al. Signal-specialized parameterization for piecewise linear reconstruction [C]//Eurographics Symposium on Geometry Processing. [s. l.]：[s. n.]，2004：39-52.

[5] Yi B K，Faloustsos C. Fast time sequence indexing for arbitrary Lp norms[C]//Proc. of the 26th International Conference on Very Large Databases. San Francisco：Morgan Kautmann Publishers Inc，2002：385-394.

[6] Keogh E，Chu S，Hart D，et al. Segmenting time-series：A survey and novel approach[C]//Data Mining in Time-series Databases. [s. l.]：World Scientific，2004：1-22.

[7] Keogh E，Chakrabarti K，Pazzani M J，et al. Locally adaptive dimensionality reduction for indexing larger time series databases[C]//Proc. of ACM SIGMOD Conference on Management of Data. Santa Barbara，CA：[s. n.]，2001：151-162.

[8] Park S，Kim S W，Cho J S，et al. Prefix querying：an approach for effective subsequence matching under time warping in sequence databases[C]//Proceedings of the 10th International Conference on Information and Knowledge Management. New York：ACM Press，2002：255-262.

[9] Lavrenko V，Schmill M，Lawrie D，et al. Mining of concurrent text and time series [C]//Proceedings of the 6th ACM SIGKDD Int'l Conference on Knowledge Discovery and Data Mining Workshop on Text Mining. Boston，MA：[s. n.]，2002：37-44.

[10] 孙焕良，邱邦华，魏溯华. 一种优化的自底向上时间序列分段算法[J]. 沈阳建筑大学学报自然科学版，2007，23(6)：1050-1051.

[11] 肖 辉，马海兵，龚 薇. 基于时态边缘算子的时间序列分段线性表示[J]. 计算机工程与应用，2008，44(19)：156-159.

[12] 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.

[13] 杜 奕，卢德唐，李道伦，等. 基于层次聚类的时间序列在线划分算法[J]. 模式识别与人工智能，2007，20(3)：23-27.

[14] Chung F L，Fu T C，Luk R，et al. Time series pattern matching based on perceptually important points[C]//Proc. of the IJCAI Workshop on Learning from Temporal and Spatial Data. [s. l.]：[s. n.]，2001：1-7.

[15] 周大镯，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)：15-16.

(上接第 102 页)

registration[C]// International Conference on Image Processing. Singapore：[s. n.]，2004：1081-1084.

[11] 周 柯. 基于 HOG 特征的图像人体检测技术的研究与实现[D]. 武汉：华中科技大学，2008.

[12] Zhu Q，Avidan S，Yeh M C，et al. Fast Human Detection Using a Cascade of Histograms of Oriented Gradients [C]// IEEE Conference on Computer Vision and Pattern Recognition. [s. l.]：[s. n.]，2006：1491-1498.