第 34 卷 第 23 期 计 算 机 工 程 2008 年 12 月
Vol.34 No.23 Computer Engineering December 2008
·博士论文· 文章编号：1000—3428(2008)23—0014—03 文献标识码：A 中图分类号：TP311

# 基于序列重要点的时间序列分割

周大镯[1,2]，李敏强[1]
(1. 天津大学管理学院，天津 300072；2. 河北经贸大学计算机中心，石家庄 050061)

**摘 要：**时间序列包含的数据量大、维数高、数据更新快，很难直接在原始时间序列上进行数据挖掘。该文提出一种基于序列重要点(SIP)的时间序列分割算法——PLR_SIP，用 SIP 组成的直线段近似描述时间序列。将 SIP 作为时间序列的分割点，反映时间序列的主要特征，降低时间序列的维数，使整体误差达到最小。

**关键词：**时间序列；序列重要点；分割

# Time Series Segmentation Based on Series Importance Point

ZHOU Da-zhuo[1,2], LI Min-qiang[1]
(1. School of Management, Tianjin University, Tianjin 300072;
2. Computer Center, Hebei University of Economics and Trade, Shijiazhuang 050061)

**【Abstract】** Time series data is characterized as large in data size, high dimensionality and updates continuously. It is hard to manipulate for data analysis and mining in its original structure. Defining a more effective and efficient time series segmentation algorithm is of fundamental importance. This paper proposes a time series segmentation algorithm based on Series Importance Point (SIP), which can approximately represent time series by linear composed of SIP. This method adopts SIP as segmentation point in time series reflecting mostly character of time series. The dimensionality of time series is reduced, and the error of the whole is least.

**【Key words】** time series; Series Importance Point(SIP); segmentation

时间序列是一类重要的数据对象，在经济、气象、医疗等领域都普遍存在。时间序列数据通常是高维多变量数据，若直接利用原始数据进行相似查询、序列聚类和分类等操作，效率很低。解决该问题的方法是将一个长时间序列分割为若干个相对短但不重叠的子序列，并将各个子序列转换为某种高级数据表示形式。时间序列分割的用途有 3 个方面：(1)可以提取时间序列的主要特征，去除细节干扰，只保留时间序列的主要形态，更能反映时间序列的自身特征，有利于提高时间序列查询的效率和准确性。(2)将时间序列从高维空间变换到低维特征空间，降低数据维数，实现对数据的压缩，提高查询效率。(3)用来发现时间序列中的异常模式。

## 1 背景介绍

文献[1]将输入为时间序列、输出为直线段的算法统称为分段线性算法。该方法比较符合人的直观经验，而且通常索引结构维数低、计算速度较快，被较多人采用。近年来，国内外学者对分段线性表示法进行深入研究，吸收其他方法和技术的优势，提出许多新的分段表示方法。目前已有的PLR分段算法可分为：(1)由拟合误差确定分段：主要是通过直线段来拟合原始时间序列，直到达到输入的误差阈值。如滑动窗口PLR_SW[2]、自顶向下PLR_ TD[3]、自底向上PLR_BU[4]等，该类算法只关注局部的最小误差，在拟合的过程中平滑掉原始数据上的一些主要特征，从而忽略了序列整体的变化特征。(2)通过特殊点分段：主要有界标模型法[5]、边缘点分段法[6]、重要点分段法PLR_IP[7]。该类方法通过一些特殊点如局部极值点、边缘点或重要点等对时间序列进行线形分段，避免了直线拟合等方法中平滑所导致的重要信息遗失，但已有的算法比较复杂，在分段的过程中没有考虑时间序列的整体误差情况，并且降维效果不如拟合法。

本文提出基于序列重要点(Series Importance Point, SIP)的时间序列线形分段算法 PLR_SIP，在选择 SIP 作为分段点时，既考虑了时间序列的整体特征，也考虑了时间序列的整体误差最低，算法的执行效率较高。

## 2 问题描述及定义

时间序列是由记录时间和记录值组成的元素的有序集合，记为：$X=<x_1=(t_1, v_1), x_2=(t_2, v_2), \cdots, x_i=(t_i, v_i), \cdots, x_n=(t_n, v_n)>$，其中，元素$x_i=(t_i, v_i)$表示时间序列在$t_i$时刻的记录值$v_i$。人类的视觉通常将平滑的曲线分为多个直线段处理，对序列第 1 印象最深的点就是序列的极值点。时间序列中的极值点对时间序列的形状有不同的影响，有的影响很大，有的对整体形状影响微乎其微。Pratt等[7]提出基于重要点的分段方法，重要点被定义为局部范内与端点的比值超过参数$R$的极值点，通过选择不同的参数$R$，可获得不同细程度的分段。该算法不能很好地体现重要点在这个时间序列中的整体性，同时没有考虑到整体误差。文献[8]提到点到区域端点的距离采用的 3 种距离测量方法：欧几里德距离，正交距离，垂直距离。图 1 给出了数据点$X_4$到$X_1$和$X_6$的 3 种距离，其中，$a+b$是欧几里德距离；$c$是垂直距离；$d$是正交距离。

欧几里德距离：

$$D=a+b=\sqrt{(t_4-t_1)^2+(v_4-v_1)^2}+\sqrt{(t_4-t_6)^2+(v_4-v_6)^2} \quad (1)$$

**基金项目：**河北省科技攻关计划基金资助项目(062135140)
**作者简介：**周大镯(1971－)，女，副教授、博士研究生，主研方向：数据挖掘；李敏强，教授、博士生导师
**收稿日期：**2008-04-25 **E-mail：**zhou_zhuo@163.com

垂直距离：

$$D = c = |(v_1 + (v_6 - v_1) \times \frac{t_4 - t_1}{t_6 - t_1}) - v_4| \quad (2)$$

正交距离：

$$s = \frac{v_6 - v_1}{t_6 - t_1} \quad (3)$$

$$tc = \frac{t_4 + s \times v_4 + s \times v_6 - s^2 \times t_6}{1 + s^2} \quad (4)$$

$$vc = s \times tc - s \times t_6 + v_6 \quad (5)$$

$$D = d = \sqrt{(tc - t_4)^2 + (vc - v_4)^2} \quad (6)$$

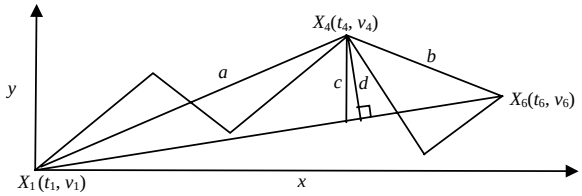



**图 1　$X_4$到$X_1$和$X_6$的 3 种距离**

这 3 种计算距离的度量方法各有侧重点，文献[8]通过实验数据说明采用垂直距离和正交距离度量产生的 SIP 顺序一致，而用欧几里德距离距离度量产生 SIP 的顺序则不同。垂直距离和正交距离度量比较见图 2，*A* 点到端点 *GH* 的正交距离为|*AC*|、垂直距离为|*AB*|，*D* 点到端点 *GH* 的正交距离为|*DF*|、垂直距离为|*DE*|。显然三角形 *ABC* 和三角形 *DEF* 相似，则 *AB* 和 *DE*, *AC* 和 *DF* 等比例缩放。在时间区域[*G*, *H*]中，若 *D* 点到 *GH* 的正交距离|*DF*|为所有点中最大正交距离，可得出 *D* 点到 *GH* 的垂直距离|*DE*|也是所有点中最大垂直距离。因此，使用垂直距离和正交距离度量产生的 SIP 顺序一致。

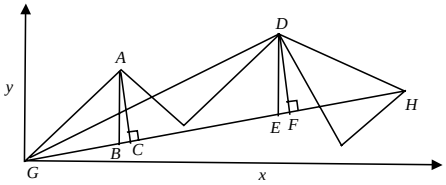



**图 2　垂直距离、正交距离度量比较**

欧几里德距离和正交距离度量比较见图 3，*B* 点到端点 *AC* 的正交距离为|*BE*|，*D* 点到端点 *AC* 的正交距离为|*DC*|。若|*BE*|=|*DC*|，可以得出|*AB*|+|*BC*|>|*AD*|+|*DC*|，在时间区域[*A*, *C*]中，若 2 点到 *AC* 的正交距离相等，并不能推出这 2 点到[*A*, *C*]端点的距离和相等。因此，使用欧几里德距离和正交距离度量产生的 SIP 顺序可能不一致。

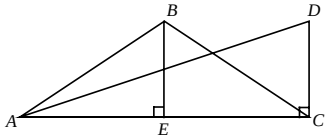



**图 3　欧几里德距离、正交距离度量比较**

**定义**　所在区域误差最大，距离区域端点距离最远，这样的点叫序列重要点(SIP)。

以前的研究者在对时间序列做分段时，考虑到每个分段后的直线段在拟和原始数据时，得到误差值应尽量最小，计算误差时一般使用的是点到拟和线段的正交距离。上文证明了垂直距离和正交距离度量产生的 SIP 顺序一致，并且垂直距离计算方法简单。

## 3　基于序列重要点分割算法 PLR_SIP

由 SIP 定义可知，在选择时间序列中 SIP 应遵循 2 点：

(1)该点必须是距离邻近 SIP 距离最远的点；

(2)该点是否要加入到 SIP 中，要考虑该区域误差是否全局最大。

选择序列重要点 SIP 的算法描述如下：

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

**输出**　序列重要点 SIP

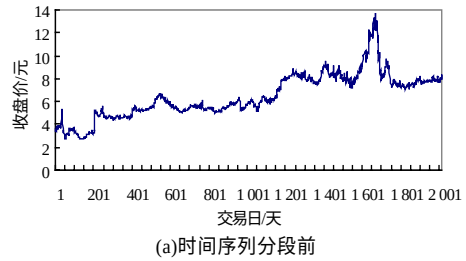
```
SIP=[]；SIP←x1; SIP←xn
//初始化 SIP，初始点和终止点加入到 SIP
BSIP←Seg(x1,xn)
//计算区域[x1,xn]内距离端点距离最大点 xc 和误差 s 存入 BSIP
While s>=e
    SIP←xc              //将 xc 从 BSIP 移入 SIP
   If b-a>=2            //a 和 b 是 xc 在 SIP 中邻近的左右点
     BSIP←Seg(a,xc)
     BSIP←Seg(xc,b)
   end
   [xc,s]←max(BSIP) //选择 BSIP 中 s 最大的点
End
```

函数 *Seg* 的作用是求区域[*u*, *v*]中的 *xc* 和 *s*，算法描述如下：

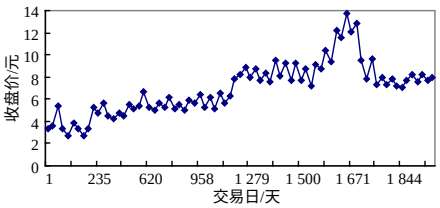
```
function BSIP=Seg(u,v)   //u 和 v 是主程序传递过来的区域端点
s=0;max=0;
for all x∈(u,v)
    dist←dist(u,x,v)
//利用垂直距离度量计算 x 到 uv 直线段距离
    s←s+dist;
    if dist>max
        max←dist;xc←x;
    end
end
return(xc,s)
```

## 4　实验结果

该算法只要求输入时间序列、分段的区域误差值，参数要求少，算法容易实现。算法实现使用工具软件 Matlab 7.1 编写了所有的程序，并在方正笔记本(CPU 1.4 GHz，内存 256 MB，硬盘 40 GB，Windows XP 操作系统)上实现。采用的数据集为上海证券交易所股票交易代号为 SH600001 的数据集，该数据集只对收盘指数时间序列进行分割，时间范围是 1998 年 3 月 11 日～2006 年 7 月 21 日，共 2 014 个交易日。当输入误差为 10，时间序列被分割为 76 段，压缩比例为 2 014/76，时间序列分段前后比较见图 4。




(a)时间序列分段前




(b)时间序列分段后

**图 4　基于 PLR_SIP 的时间序列分割前后比较**

在相同时间序列数据下，对其分别使用 PLR_SIP, PLR_IP, PLR_TD 和 PLR_SW 线形分割的方法进行分割，实验结果表明，PLR_SIP 分割很好地反映了时间序列的整体特征；PLR_IP 在分割的过程中过于强调局部点的重要性，没有考虑全局最优的问题；PLR_TD 和 PLR_SW 分割方法在拟合的过程中平滑掉了一些重要信息，没有很好地描述时间序列中的异常数据。算法的时间复杂度上 PLR_SIP 和 PLR_IP 所占用的 CPU 时间相近为 $O(l\times\log n)$，PLR_SW 的时间复杂度为 $O(l\times n)$，PLR_TD 的是 $O(n\times n)$。通过实验比较这些分割算法在不同的输入误差下算法所占用的 CPU 时间，结果见图 5。

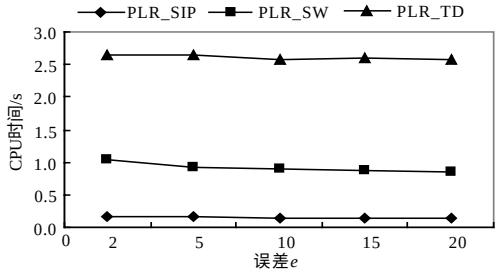



**图 5　3 种算法在不同误差下占用的 CPU 时间**

## 5　结束语

与其他 PLR 分割算法相比，采用本算法对时间序列分段分割效果好、效率高，能很好地描述时间序列的整体特征。在选择 SIP 的过程中考虑到局部最优和全局最优的问题，基本上能做到既保留原始时间序列的整体形态，又可达到分段误差最小。通过本算法，可提供给时间序列研究者一个对时间序列描述的根据，并在此算法的基础上进一步发现时间序列模式。

## 参考文献


[1] 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 Systems, 2001, 3(3): 263-286.

[2] Qu Yunyao, Wang Changzhou. Supporting Fast Search in Time Series for Movement Patterns in Multiples Scales[C]//Proc. of the 7th ACM CIKM Int'l Conference on Information and Knowledge Manage- ment. Bethesda, USA: [s. n.], 1998.

[3] Keogh E, Pazzani M. An Enhanced Representation of Time Series Which Allows Fast and Accurate Classification, Clustering and Relevance Feedback[C]//Proc. of the 4th Int'l Conference on Knowledge Discovery and Data Mining. New York, USA: [s. n.], 1998.

[4] Park S, Lee D. Fast Retrieval of Similar Subsequences in Long Sequence Databases[C]//Proc. of the 3rd IEEE Knowledge and Data Engineering Exchange Workshop. Chicago, USA: [s. n.], 1999.

[5] Perng C S, Wang Haixun. Landmarks: A New Model for Similarity-based Pattern in Time Series Databases[C]//Proc. of the 16th IEEE Int'l Conf. on Data Engineering. California, USA: [s. n.], 2000.

[6] 肖　辉. 时间序列的相似性查询与异常检测[D]. 上海: 复旦大学, 2005.

[7] Pratt K B, Eugene F. Search for Patterns in Compressed Time Series[J]. International Journal of Image and Graphics, 2002, 2(1): 89-106.

[8] Fu Tak-chung, Chung Fulai, Luk R, et al. Representing Financial Time Series Based on Data Point Importance[Z]. (2007-04-09). http://www.elsevier.com/locate/engappai.


（上接第 8 页）

将表 2 的数据作为评价样本，运行系统结果如表 5 所示。

**表 5　系统评价结果**

| 样本号 | TSK 预测评价值 | 对应水质级别 | CTSK 预测评价值 | 对应水质级别 |
|---|---|---|---|---|
| 1 | 0.602 | 4 | 0.643 | 4 |
| 2 | 0.131 | 1 | 0.164 | 1 |
| 3 | 0.183 | 2 | 0.151 | 1 |
| 4 | 0.172 | 2 | 0.167 | 1 |
| 5 | 0.624 | 4 | 0.660 | 4 |
| 6 | 0.587 | 4 | 0.621 | 4 |
| 7 | 0.322 | 3 | 0.312 | 2 |
| 8 | 0.302 | 2 | 0.318 | 2 |
| 9 | 0.406 | 3 | 0.413 | 3 |
| 10 | 0.634 | 4 | 0.654 | 4 |

用改进的BP神经网络学习算法[6]进行评价，结果见表 6。

**表 6　改进的 BP 神经网络学习算法评价结果**

| 样本号 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| 水质级别 | 4 | 1 | 1 | 1 | 3 | 4 | 2 | 2 | 3 | 4 |

表 4、表 5 给出了 CTSK 与 TSK 的对训练样本以及测试样本的结果比较，可以看出，与 TSK 相比，CTSK 评价效果更准确，精度更高，CTSK 具有比 TSK 更好的泛化能力、建模能力和更强的鲁棒性。从表 6 可以看出，CTSK 系统的分析结果与改进的 BP 神经网络学习算法评价的评判结果基本一致。改进的 BP 算法虽然可在一定程度上解决标准 BP 算法的局部极小问题，但收敛速度仍然很慢，而且需要很大的内存空间。而 CTSK 系统的收敛速度要比改进的 BP 算法快得多，而且无须消耗大量的内存。因此，CTSK 系统更可行。

## 4　结束语

本文对新型的中心 TSK 模糊系统 CTSK 进行了推导和实验分析。仿真实验结果表明，与 TSK 模糊系统相比，CTSK 对地下水质的判别、模型的拟合效果更好。因此，CTSK 在建模、环境质量评价等方面有很好的潜在价值，值得推广和应用。

## 参考文献


[1] 王士同. 模糊系统、模糊神经网络及应用程序设计[M]. 上海: 上海科学技术文献出版社, 1997.

[2] Alata M, Su Chunyi, Demirli K. Adaptive Control of a Class of Nonlinear Systems with a First-order Parameterized Sugeno Fuzzy Approximator[J]. IEEE Trans. on Systems, Man and Cybernetics, 2001, 31(3): 410-419.

[3] Wong Chinchang, Lai Hungren. Generating Fuzzy Control Rules by a Clustering Algorithm Based on a Grey Relational Measure[C]// Proc. of IEEE International Conf. on Fuzzy Systems. Seoul, Korea: [s. n.], 1999.

[4] 赵振宇, 徐用懋. 模糊理论和神经网络的基础与应用[M]. 北京: 清华大学出版社, 1996.

[5] Chung Fulai, Duan Jicheng. On Multistage Fuzzy Neural Network Modeling[J]. IEEE Trans. on Fuzzy Systems, 2000, 8(2): 125-142.

[6] 吴凌云. BP 神经网络学习算法的改进及其应用[J]. 信息技术, 2003, 27(7): 42-44.