时间序列重要点分割的异常子序列检测

张力生[1] 杨美洁[2] 雷大江[2]

(重庆邮电大学软件学院 重庆 400065)[1] (重庆邮电大学计算机学院 重庆 400065)[2]

摘 要 时间序列具有数据量大的特点,直接对其检测复杂度高。因此提出了一种基于时间序列重要点的异常子序列检测算法。子序列的异常检测弥补了点异常检测的局限性。该算法首先获得了一系列平滑后的重要点,然后根据其进行子序列划分,并提取每个子序列的4个特征值:长度、高度、均值和标准差,将其运用到欧氏距离中,最后通过KNN算法来检测异常子序列。实验证明了该算法的有效性和可行性。

关键词 重要点分割,平滑处理,特征值,KNN算法

中图法分类号 TP301.6 **文献标识码** A

Outlier Sub-sequences Detection for Importance Points Segmentation of Time Series

ZHANG Li-sheng[1] YANG Mei-jie[2] LEI Da-jiang[2]

(College of Software, Chongqing University of Posts and Telecommunications, Chongqing 400065, China)[1]

(College of Computer Science, Chongqing University of Posts and Telecommunications, Chongqing 400065, China)[2]

Abstract Because the time series has a large amount of data, detecting it directly will has a high complexity. And this paper proposed an outlier sub-sequences detection algorithm based on importance points segmentation of time series to relieve the problem. Outlier detection of sub-sequences can offset the limitations of the outlier detection of points. This algorithm firstly obtains a series of smoothed important points, and then divides the sequences according to them, meanwhile extracts the four characteristic values of each sub-sequence: length, height, mean and standard deviation, and applies these four characteristic values to the european euclidean distance. Finally, it detects the outlier sub-sequences with the KNN algorithm. Experimental results show that the algorithm is effective and reasonable.

Keywords Importance points segmentation, Smooth process, Characteristic values, KNN algorithm

1 引言

时间序列数据主要是指按照时间的先后顺序得到的各个观测值的有序集合,目前广泛应用于商业、经济、医疗、基因表达[1]等领域。随着时间的推移,序列含有越来越多的数据,同时存在多变量[2]的表现形式,例如多媒体领域、上市公司的股票交易情况等。本文只针对一维数据和实数值的情形进行研究。

目前,对于异常没有一个公认的说法,普遍采用Hawkin给出的定义[3]:异常是指在数据集中偏离大部分数据的那些数据。让人怀疑这些数据并非随机偏差,而是由不同的机制产生的。从20世纪80年代起,异常检测在各个领域中得到了广泛的应用,先是由Knorr和Ng[4]提出了基于距离的异常检测算法,其通过计算数据点和对象之间的距离来检测异常点。当数据集中含有不同的密度子集时,Breuing等[5]提出了基于密度的异常检测算法,这种算法检测精度较高,但当处理大数据集时其复杂度过高,无法获得令人满意的效果。上述算法多是检测时间序列的异常点,近年来异常序列检测也有了一定的进展。Eamonn K[6]等研究如何检测时间序列中最异常的子序列,算法首先将时间序列符号化,通过符号检索出时间序列最不寻常的子序列[7];林果园等[8]在改进的隐马尔科夫方法的基础上,提出了将动态行为和全局特征结合起来进行异常检测的方法,但面对较大数据集时,算法的时间复杂度较高。基于上述缺陷,提出了时间序列重要点分割的异常子序列检测算法。

2 相关定义

时间序列具有数据量大、更新速度快等特点,如果直接在原始时间序列上进行数据检测,将会变得异常复杂。通过序列重要点将时间序列分割成若干个相对较短但不重叠的子序列,来降低时间复杂度。

2.1 序列重要点分段[9]

定义1(时间序列) 时间序列是由一系列的观测值和观测时间组成的有序集合[9],记为 $X=((x_1,t_1),(x_2,t_2)\cdots(x_n,t_n))$。

定义2(重要点) 任意时间序列 $Q=((x_1,t_1),(x_2,t_2)\cdots$

到稿日期:2011-06-28 返修日期:2011-09-29 本文受重庆邮电大学青年基金(A2007-53),重庆自然科学基金(Catch,2008bb2086)资助。

张力生(1965-),男,高级工程师,副教授,硕士生导师,主要研究领域为数据挖掘、智能信息系统,E-mail:zhangls@cqupt.edu.cn;杨美洁(1984-),女,硕士,主要研究领域为数据挖掘、智能信息系统;雷大江(1979-),男,博士,讲师,主要研究领域为智能信息系统。

$(x_n,t_n))$，对于$\forall x_i(i=1,2,\cdots,n)$，给定常数 p 后，如果

(1)当 $i-p\leqslant 1$ 且 $i+p<n$ 时，$x_j\in(x_1,x_2,x_3,\cdots,x_{i+p})$；

(2)当 $1<i-p<i+p<n$ 时，$x_j\in(x_{i-p},x_{i-p+1},x_{i-p+2},\cdots,x_{i+p})$；

(3)当 $i+p\geqslant n$ 时，$x_j\in(x_{i-p},x_{i-p+1},x_{i-p+2},\cdots,x_n)$。

均有 $x_i\geqslant x_j(x_i\leqslant x_j)$，则称 x_i 为极大值点(极小值点)。

重要点分段法是时间序列的一个重要算法，主要利用极值点来进行描述。由于人们所关注的信息大部分包含在重要点中[9]，因此在分段时选择合适的重要点显得尤为关键。

假设采用定义 1 得到一个含有 n 个重要点的序列 $Q=((k_1,t_1),(k_2,t_2),\cdots,(k_n,t_n))$，对其采用式(1)[10]进行平滑[11]处理，可以进一步完善该算法。

$$\begin{aligned}&|t_i-t_{i+1}|<w\\&|(k_i-k_{i+1})/k_{i+1}|\leqslant\varepsilon\end{aligned}\tag{1}$$

对于序列 Q 中的任意两个相邻的重要点(k_i,t_i)和(k_{i+1},t_{i+1})，如果满足式(1)，则删除点(k_{i+1},t_{i+1})。通过上述方法可以得到平滑后的重要点序列，此序列具有很重要的现实意义。例如一个销售部门想要以周(一周 5 个工作日)来分析本周的销售情况，并只对涨幅在 4%以内的数据感兴趣，那么此时的(w,ε)应该取为(5,0.04)。

定义 3(时间序列异常子序列)[10] 时间序列中与其它子序列具有显著差异的、异常行为的子序列称为时间序列的异常子序列。

2.2 特征值求解

由于时间序列的子序列可能存在长度异常、高度异常、平均值异常和标准差异常，因此需要通过这 4 个特征值[12]对其进行检测，计算公式如下所示。

定义 4 子序列长度：

$$Sql=ik-i1+1\tag{2}$$

式中，ik 代表该子序列的最后一个数据在序列中的位置，$i1$ 代表该子序列的第一个数据在序列中的位置。

定义 5 子序列高度：

$$Sqh=x(i)\max-x(i)\min\tag{3}$$

式中，$x(i)\max$ 是该子序列的最大数据，$x(i)\min$ 是该子序列的最小数据。

定义 6 子序列均值：

$$Sqx=\frac{1}{Sql}\tag{4}$$

定义 7 子序列的标准差：

$$Sq\sigma=\sqrt{\frac{1}{Sql}\sum_{j=i1}^{ik}(x(j)-Sqx)^2}\tag{5}$$

定义 8 子序列对象 p 和 q 的距离[12]：

通过定义 4 到定义 7 求得每个子序列的 4 个特征值，并将其投影到四维特征空间中，然后计算任意两个子序列对象之间的距离。例如假设对象 $P(xp,yp,kp,lp)$和对象 $Q(xq,yq,kq,lq)$是四维空间 $C(sql,sqh,sqx,sq\sigma)$中的任意两个点，则二者之间的距离为：

$$Dist(P,Q)=\sqrt{(xp-xq)^2+(yp-yq)^2+(kp-kq)^2+(lp-lq)^2}\tag{6}$$

定义 9 对象 p 的 k 近邻距离 $k\text{-}dist(p)$[12]：

给定 $k\in N^+$，$\forall p\in C(Sql,Sqh,Sqx,Sq\sigma)$，则对象 P 的 k 近邻距离 $k\text{-}dist(p)$要满足以下 2 点：

(1)至少有 k 个对象，$o\in C\backslash\{p\}$，$Dist(p,o)\leqslant k\text{-}dist(p)$；

(2)至多有 $k-1$ 个对象，$o\in C\backslash\{p\}$，$Dist(p,o)<k\text{-}dist(p)$；

因此称 $Dist(p,o)$为对象 p 的 k 近邻距离，记为 $k\text{-}dist(p)$。

定义 10 对象 q 到对象 p 的 k 近邻可达距离 $r\text{-}distk(q,p)$[12]：

$$r\text{-}distk(q,p)=\max(Dist(q,p),k\text{-}dist(p))$$

如果 $Dist(q,p)>k\text{-}dist(p)$，则 $r\text{-}distk(q,p)=Dist(q,p)$；否则，$r\text{-}distk(q,p)=k\text{-}dist(p)$。

定义 11 对象 q 的 k 局部可达密度 $lrd(q)$[12]：

$$lrd(q)=\frac{k}{\sum\limits_{p\in k(q)}r\text{-}distk(q,p)}\tag{7}$$

式中，$k(q)$表示 q 的 k 近邻范围，局部可达密度反映了该对象的周围对象的分布密度。具有较小局部密度的对象成为异常对象的可能性较大；反之亦然。

定义 12 对象 q 的局部异常系数 $lof(q)$[12]：

$$lof(q)=\frac{\frac{1}{k}\sum\limits_{p\in k(q)}lrd(p)}{lrd(q)}\tag{8}$$

如果对象 q 的局部异常系数较大，则意味着该对象的局部范围所包含的对象较少，具有较小的局部密度，从而说明该对象是异常的可能性较大；反之亦然。

3 时间序列重要点分割的异常子序列检测算法

3.1 异常子序列检测算法的设计模型

异常子序列检测算法的设计模型如图 1 所示。

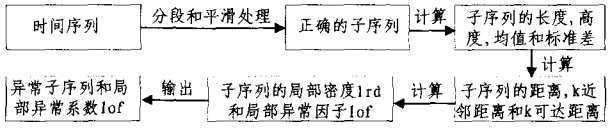


图 1 异常子序列检测算法的设计模型

3.2 算法描述

输入：时间序列 $X=((x_1,t_1),(x_2,t_2)\cdots(x_n,t_n))$，常数 p，w，ε，k，num(异常系数最大的前 num 个子序列)；

输出：前 num[13]个局部异常系数最大的异常子序列及其局部异常系数。

Step1 根据已给的常数 p 和定义 2 对给定的时间序列进行分段，获得了一系列的子序列；

Step2 根据已给的常数 w、ε 和式(1)对已得到的子序列进行平滑处理，删除一些冗余的点，从而得到了正确的子序列；

Step3 根据定义 4 到定义 7，获取每一个子序列的长度、高度、均值和标准差，并将这 4 个特征值映射到四维空间中，然后通过定义 8 计算任意两个子序列对象之间的距离；

Step4 根据定义 9 求得任意一个子序列对象 p 的 k 近邻距离；

Step5 根据定义 10 求得任意一个子序列对象到已知的子序列对象的 k 近邻可达距离；

Step6 根据定义 11 和定义 12 分别求得任意一个子序列对象的 k 局部可达密度和局部异常系数；

Step7 根据求得的子序列的局部异常系数对其进行排序，将前 num 个子序列进入异常子序列集合，并输出这些子序列及其相应的局部异常系数。

4 实验与结果分析

为了验证算法，本实验使用 C++语言编写了所有的程序，并在 CPU 2.81GHz，内存 2GB，硬盘 500GB，Windows XP 操作系统的计算机上进行了算法的验证，然后将本实验算法应用于从 2005 年 8 月 30 日到 2011 年 5 月 25 日的 TCL 集团收盘价的数据集。

4.1 实验结果

当常数 $p=12$，$w=5$，$\varepsilon=0.08$，取不同的 k 近邻（k 分别取值为 7、9、11、13）时，局部异常因子最大的前 4 个子序列如表 1 所列。

表 1 TCL 集团收盘价数据集的实验结果

k-近邻	子序列编号	局部异常系数
7-近邻	S10	4.22578
	S19	2.78234
	S31	1.96366
	S12	1.57323
9-近邻	S10	4.33918
	S19	2.86004
	S31	2.01306
	S12	1.55767
11-近邻	S10	4.25664
	S19	2.81629
	S31	1.96694
	S12	1.46926
13-近邻	S10	3.14664
	S19	2.53997
	S31	2.24807
	S12	1.44924

从表 1 可以看出，当固定系数 p，w，ε，并且 k 分别取值为 7、9、11、13 时，4 次结果相同。因此证明了该算法的有效性和健壮性。

当固定常数 p，w 不变，ε 为 0.1，取不同的 k 近邻（k 分别取值为 7、9、11、13）时，局部异常因子最大的前 4 个子序列如表 2 所列。

表 2 不同 ε 的实验结果

k-近邻	子序列编号	局部异常系数
7-近邻	S10	3.99751
	S23	1.95902
	S19	1.80875
	S33	1.77015
9-近邻	S10	4.21544
	S23	1.96321
	S19	1.77694
	S33	1.72938
11-近邻	S10	4.02503
	S23	1.96484
	S19	1.76925
	S33	1.71933
13-近邻	S10	3.96655
	S23	1.98911
	S19	1.78887
	S33	1.74376

当其他参数保持不变，ε 的取值发生变化时，表 2 的结果和表 1 不一致，说明 ε 是影响参数之一。

当固定常数 p，$\varepsilon=0.1$ 不变，$w=7$，取不同的 k 近邻（k 分别取值为 7、9、11、13）时，局部异常因子最大的前 4 个子序列如表 3 所列。

表 3 不同 w 的实验结果

k-近邻	子序列编号	局部异常系数
7-近邻	S9	3.85083
	S19	2.90209
	S23	2.11623
	S33	1.94206
9-近邻	S9	3.53198
	S19	2.65724
	S23	1.77552
	S33	1.52761
11-近邻	S9	3.21065
	S19	2.41427
	S23	1.65404
	S33	1.45738
13-近邻	S9	3.12389
	S19	2.39810
	S23	1.73858
	S33	1.55593

从表 3 和表 2 的结果来看，当 w 取不同值时，结果也随之发生变化，这说明 w 也是影响因素之一。

4.2 算法分析

算法时间复杂度主要包括 4 部分：(1)时间序列分割：时间复杂度为 $O(np)$（p 为分段时给定的常数，n 为数据点总数）。(2)计算子序列之间的距离：时间复杂度为 $O(c^2)$，其中 c 为子序列的个数。(3)计算子序列的局部可达密度 lrd，时间复杂度为 $O(ck)$；(4)计算模式的局部异常因子，时间复杂度为 $O(ck)$。因此算法总共的时间复杂度为 $O(np)$。

4.3 分段算法时间复杂度对比[14]

目前存在的几种时间序列分段算法的时间复杂度的对比如表 4 所列。

表 4 分段算法时间复杂度

算法名称	时间复杂度
自顶向下分段算法	$O(N^2K)$
自底向上分段算法	$O(LN)$
滑动窗口分段算法	$O(LN)$
序列重要点分段算法	$O(PN)$

其中，N 为数据点总数，K 为分段数，L 为平均段的长度，P 为此算法给定的常数，且 $P<L$。

结束语 时间序列数据量大、维数高[15]、更新速度快，如果直接对其进行挖掘将异常困难。提出了基于时间序列重要点分割的异常子序列检测算法，利用序列重要点对其进行子序列划分，并在子序列的基础上检测时间序列的异常，提高了算法的效率和准确性，可以快速地检测出时间序列的异常子序列。

参 考 文 献

[1] Carla S. Moller-Levet. Clustering of gene expression time-series. Second Year Report[EB/OL]. http://images ee. umist. ac. Uk/Carla/Report. pdf, 2003

[2] 程文聪，邹鹏，贾焰. 多维时序数据中的相似子序列搜索研究[J]. 计算机研究与发展，2010，47(3)：416-425

[3] Hawkin D. Identification of outliers[M]. London: Chapman and Hall, 1980

[4] Knorr E M, Ng R T. A Unified notion of outliers: properties and computation[C] //ICDM'97. [S. l.]: AAAI Press, 1997: 219-222

[5] Breunig M M, Kriegel H P, Ng R, et al. LOF: Identifying Density-based local Outliers[C] //ACM SIGMOD. 2000: 93-104

[6] Keogh E, Lin J. Finding unusual medical time- series subsequences: algorithms and applications[J]. IEEE Transactions on Information Technology in Biomedicine, 2006, 10(3): 429-439

[7] Keogh E, Lin J, Lee S-H, et al. Finding the most unusual time series subsequence: algorithms and Applications[J]. Knowl Inf Syst, 2006, 11(1): 1-27

[8] 林果园,郭山清. 基于动态行为和特征模式的异常检测模型[J]. 计算机学报, 2006, 29(9): 1553-1559

[9] 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(27): 14-16

[10] 贾素玲,陈当阳,姜浩. 时序数据挖掘中的数据表示算法[J]. 计算机工程与应用, 2006, 29: 184-186

[11] Ameen J R M, Basha R. Hierarchical Data Mining for Unusual Sub-sequence Identifications in Time Series Processes[C]// ICICIC. 2009

[12] 詹艳艳,陈晓云,徐荣聪. 基于时间序列模式表示的异常检测算法[J]. 计算机应用, 2007, 24(11): 96-99

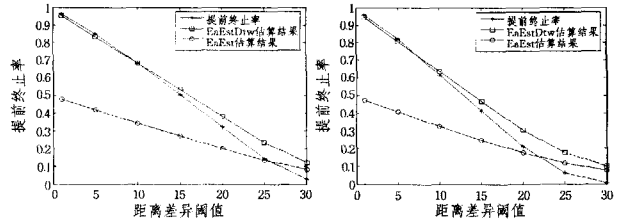
[13] Bu Y, Leung T W, Fu A W C, et al. WAT: Finding Top-K Discords in Time Series Database[C]// Proceedings of 7th SIAM. 2007

[14] 喻高瞻,彭宏,胡劲松. 时间序列数据的分段线性表示[J]. 计算机应用与软件, 2007, 24(12): 17-18

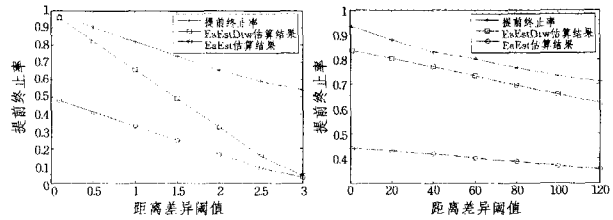
[15] Ameen J, Basha R. Mining Multiple Periodic Time Series for Detecting Unusual Sub-Sequences[C]// ICICIC. 2009: 1473

(上接第 167 页)

精度优于 ASL、LP1 数据集。



(a) NormDs 中的实验结果　(b) ExpDs 中的实验结果



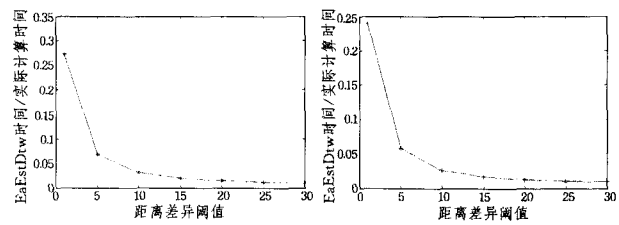
(c) ASL 中的实验结果　(d) LP1 中的实验结果

图 3　两种估算方法在 4 组数据集中的实验结果

表 1　EaEstDtw、EaEst 方法的估算误差

数据集	EaEstDtw 方法		EaEst 方法	
	最大绝对误差	平均绝对误差	最大绝对误差	平均绝对误差
NormDs	0.0938	0.0432	0.4854	0.2389
ExpDs	0.1156	0.0580	0.4851	0.2173
ASL	0.4968	0.2523	0.5065	0.4945
LP1	0.0970	0.0759	0.4916	0.4067

针对 NormDs、ExpDs，用实验 3 描述的方法，比较 EaEstDtw 与实际求解提前终止率的耗费时间，结果见图 4。从图中可以看出，EaEstDtw 方法估算提前终止率的计算时间远小于求解实际值。这是因为 EaEstDtw 只需获取一定数目的点对距离作为样本点，而不用逐个计算数据集中 DTW 累积距离矩阵，因此比求解实际值节省了大量计算。



(a) NormDs 中的实验结果　(b) ExpDs 中的实验结果

图 4　EaEstDtw 与实际求解提前终止率耗费时间的比较

结束语　本文分析了 DTW 提前终止产生的机理，提出了一种 DTW 提前终止率的估算模型，并通过实验验证了方法的有效性。实验结果表明，EaEstDtw 能够以相对较小的计算代价，有效地估算 DTW 提前终止率；估算精度明显优于 EaEst 方法；EaEstDtw 方法对随机序列有着较高的估算精度，在一定程度上也适用于实际产生的多元时间序列数据集。

参 考 文 献

[1] 李俊奎,王元珍,李海波,等. 一种时间序列相似搜索中提前终止效率的估算方法[J]. 计算机科学, 2009, 36(1): 114-117

[2] Berndt D J, Clifford J. Using dynamic time warping to find patterns in time series[C]// Proc of the KDD Workshop. Seattle, WA, 1994: 359-370

[3] Vlachos M, Hadjieleftheriou M, Gunopulos D, et al. Indexing multi-dimensional time-series with support for multiple distance measures[C]// Proc of the 9th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. Washington, 2003: 216-255

[4] Keogh E J. Exact indexing of dynamic time warping[C]// Proceedings of 28th International Conference on Very Large Data Bases. Hong Kong, China, 2002: 406-417

[5] Kim M-S, Kim S-W, Shin M. Optimization of Subsequence Matching Under Time Warping in Time-series Databases[C]// ACM Symposium on Applied Computing. New Mexico, USA, 2005: 581-586

[6] Li Jun-kui, Wang Yuan-zhen. EA_DTW: Early Abandon to Accelerate Exactly Warping Matching of Time Series[C]// Proc of Int'l Conf. on Intelligent Systems and Knowledge Engineering. 2007

[7] 陈胜利,李俊奎,刘小东. 基于提前终止的加速时间序列弯曲算法[J]. 计算机应用, 2010, 30(4): 1068-1071

[8] 李爱国,覃征. 大规模时间序列数据库降维及相似搜索[J]. 计算机学报, 2005, 28(9): 1467-1475

[9] 李俊奎. 时间序列相似性问题研究[D]. 武汉: 华中科技大学计算机科学与技术学院, 2008

[10] 穆斌,闫金来. 高效的时间序列下界技术[J]. 计算机工程与应用, 2009, 45(11): 168-171

[11] 盛骤,谢式千,潘承毅. 概率论与数理统计[M]. 北京: 高等教育出版社, 2001

[12] http://kdd.ics.uci.edu/databases/High-quality Australian Sign Language/High-quality Australian Sign Language.html

[13] http://kdd.ics.uci.edu/databases/robotfailure/robotfailure.html