第 17 卷 第 3 期　　控 制 与 决 策　　2002 年 5 月
Vol 17 No. 3　　Control and Decision　　May 2002

文章编号: 1001-0920(2002)03-0324-04

一种时序数据的离群数据挖掘新算法

郑斌祥, 杜秀华, 席裕庚
(上海交通大学 自动化研究所, 上海 200030)

摘　要: 离群数据挖掘是数据挖掘的重要内容, 针对时序数据进行离群数据挖掘方法的研究。首先通过对时序数据进行离散傅立叶变换将其从时域空间变换到频域空间, 将时序数据映射为多维空间的点, 在此基础上, 提出一种新的基于距离的离群数据挖掘算法。对某钢铁企业电力负荷时序数据进行仿真实验, 结果表明了算法的有效性。

关键词: 离群挖掘; 离群数据; 数据挖掘; 知识发现
中图分类号: TP 311　　**文献标识码**: A

A new algorithm of outlier mining in time series data

ZHENG Bin-xiang, DU Xiu-hua, XI Yu-geng
(Institute of Automation, Shanghai Jiaotong University, Shanghai 200030, China)

Abstract: The outlier mining method for time series data is investigated. DFT is used to transform the time series data from time domain to frequency domain. The time series data can be mapped into the multidimensional points in multidimensional space. A distance based algorithm is proposed to mine the outliers. The time series data of the electrical load of a steel plant are used for simulation test. The simulation results show the effectiveness of the algorithm.

Key words: outlier mining; outlier data; data mining; knowledge discovery

1 引　　言

数据挖掘就是从大型数据库的数据中提取人们感兴趣的知识[1]。离群数据挖掘(简称离群挖掘)是从大量的数据中挖掘出明显偏离其它数据、不满足数据的一般行为或模式、与存在的其它数据不一致的数据。对离群数据挖掘的研究往往可以使人们发现一些潜在的有用信息, 如电力系统运行中的异常、银行信用卡欺骗行为的监测等。

时序数据是指按时间顺序取得的一系列观测值。时序数据的数据挖掘就是挖掘时序数据中潜在的有用的知识。目前时序数据的挖掘主要集中于时序数据相似性的挖掘等, 对于离群数据一般将其删除或忽略, 然而时序数据的离群数据使人们能够发现时序数据的一些潜在的有用知识。

离群数据已在统计学领域得到广泛研究[2], 但基于统计的方法需要用户建立数据点的概率分布模型, 应用时需事先知道数据集的分布和分布参数等信息。Knorr 和 Ng[3]提出基于距离的离群数据挖掘方法, 但这种方法中的距离难以确定, 而且没有离群数据的离群衡量测度。Arning 等[4]提出一种基于偏离的离群数据发现方法, 需要确定相异函数进行离群数据挖掘, 但若其相异函数的选取不合适, 则得不

收稿日期: 2000-12-11; **修回日期**: 2001-02-26
作者简介: 郑斌祥(1973—), 男, 江西余江人, 博士生, 从事数据仓库和数据挖掘等研究; 席裕庚(1946—), 男, 上海人, 教授, 博士生导师, 从事复杂系统控制理论和智能机器人等研究。

到满意的结果。

鉴于上述离群数据挖掘方法对于时序数据的离群数据挖掘具有一定的局限性, 本文针对时序数据的离群数据挖掘, 提出一种新的基于距离的时序数据离群数据挖掘方法, 并用该算法对某钢铁企业电力负荷时序数据进行了离群数据挖掘。

2　时序数据的离群数据挖掘

本文对时序数据的离群数据挖掘的目标为: 针对时序数据 X 发现数据中一部分与其余数据有明显不同的例外数据。由于时序数据的长度一般较长, 直接对整个时序数据进行离群数据挖掘, 其计算复杂性增大。因此, 本文将时序数据的离群数据挖掘分为两步: 1) 将时序数据依据应用要求划分为一系列子序列, 用离散傅立叶变换将子序列时序数据从时域空间变换到频域空间, 根据 Parseval 的理论, 时域能量函数与频域能量函数相同, 且频域空间的大部分能量集中在前几个系数上, 因此可以考虑只选用傅立叶变换得到的前 k 个系数。把这些系数看作从时间序列上提取的特征, 于是从每个序列可获得 k 个特征, 进一步将它们作为到 k 维空间上的一个映射, 即将时序数据的子序列映射为 k 维空间上的点。这样即可保留时序数据的主要特征, 同时又降低了时序数据的维数, 减小了计算的复杂性。2) 针对 k 维空间上的时序数据点, 采用一种新的离群数据的距离, 即用点 p 的第 k 个最近邻与点 p 的距离 $D^k(p)$ 作为衡量离群数据的度量。基于 $D^k(p)$ 的离群数据距离定义对时序数据点进行 k 近邻查询, 再根据离群数据的定义, 搜索到的输入数据集合中前 n 个 $D^k(p)$ 最大的数据点即为离群数据点。

与已有的基于统计的和基于偏离的离群数据挖掘算法相比, 本文提出的离群数据挖掘算法无需确定数据点的概率分布模型或相异函数等, 而是直接以离群数据点的第 k 个最近邻与其距离 $D^k(p)$ 作为衡量离群数据的度量, 从而克服了以往离群数据挖掘算法的局限性。

2 1　时序数据的离散傅立叶变换(DFT)

给定一个时间序列 $\vec{X} = \{x_t \mid t = 0, 1, \cdots, N\}$, 将 $\vec{X}$ 划分为互不重叠的子序列 $\vec{X}_1, \vec{X}_2, \cdots, \vec{X}_m$, 其中 $\vec{X}_i = \{x_{it} \mid t = 0, 1, \cdots, n-1\}, N = nm$。

对 $\vec{X}$ 的子序列 $\vec{X}_i$ 进行离散傅立叶变换, 得

$$x_{if} = \frac{1}{\sqrt{n}} \sum_{t=0}^{n-1} x_{it} \exp\left(-i * 2\pi f t / n\right)$$

$$f = 0, 1, \cdots, n-1, \quad i = 1, 2, \cdots, m \tag{1}$$

这里, $\vec{X}_i$ 和 x_{it} 代表时域信息, x_{if} 为子序列 $\vec{X}_i$ 离散傅立叶变换的傅立叶系数。

$$\vec{X}_{iF} = \{x_{if} \mid f = 0, 1, \cdots, n-1\} \tag{2}$$

根据 Parseval 的理论, 时域能量谱函数与频域能量谱函数相同, 则有

$$\| \vec{X}_i - \vec{Y}_i \|^2 = \| \vec{X}_{iF} - \vec{Y}_{iF} \|^2 \tag{3}$$

即

$$D(\vec{X}_i, \vec{Y}_i) = (E(\vec{X}_i - \vec{Y}_i))^{1/2} = (E(\vec{X}_{iF} - \vec{Y}_{iF}))^{1/2} = D(\vec{X}_{iF}, \vec{Y}_{iF}) \tag{4}$$

因此, 时序数据经离散傅立叶变换由时域空间变换到频域空间, 其距离保持不变。于是可以考虑只选用傅立叶变换得到的前 k 个系数, 并把这 k 个系数看作从时间序列上提取的特征, 将时序数据的子序列映射为 k 维空间上的点。

2 2　定　　义

对映射到 k 维空间上的时序点, 本文提出一种用 $D^k(p)$ 作为衡量离群数据度量的离群数据定义。

定义 1　对于有 N 个数据点的集合 Ω, 任取 $p \in \Omega$ 且记 $D(p, x)$ 为点 p 与 $x \in \Omega$ 的点之间的距离。对于 Ω 中不包含点 p 的数据点集合的 $D(p, x)$ 按从小到大顺序排列, 得到序列 $(D(p, x_1), D(p, x_2), \cdots, D(p, x_{N-1}))$, 记序列中第 k 个值 $D(p, x_k)$ 为点 p 的第 k 个最近邻点与点 p 的距离, 记为 $D^k(p)$。

定义 2　对于集合 Ω 中的所有点, 给定整数 n 和 k, 将点 $p \in \Omega$ 的 $D^k(p)$ 按从大到小顺序排列, 其中前 n 个点为离群数据点。

在以上定义中, 我们用 $D^k(p)$ 作为衡量离群数据的度量, $D^k(p)$ 越大, 表示点 p 邻域内的数据分布点离群程度越强。如果用户感兴趣的离群数据点的个数为 n, 而 n 的数值一般较小, 易于设定。时序数据的离群数据挖掘主要是时序数据点的 k 近邻查询, 找到输入数据集合中前 n 个 $D^k(p)$ 最大的数据点为离群数据点。

本文基于定义 1 和定义 2 的离群数据定义, 以欧几里德距离作为距离函数, 采用最小边界矩形 MBR 进行离群数据挖掘。这里 MBR 是在某点的指定范围内包含一系列点的最小矩形。

首先定义数据点 p 与 MBR 之间的最小距离和最大距离[5]。若记 δ 维空间的点 p 为 $[p_1, p_2, \cdots, p_\delta]$, δ 维空间的最小边界矩形 R 用其对角线上的顶点表示, 即 $r = [r_1, r_2, \cdots, r_\delta]$, $r' = [r_1', r_2', \cdots, r_\delta']$, 且 $\forall 1 \le i \le n, r_i \le r_i'$。

定义 3 用 $\mathrm{dist}(p_i, p_j)$ 表示 δ 维空间的点 p_i 与 p_j 之间的距离, 即

$$\mathrm{dist}^{\delta}(p_i, p_j) = \left(\sum_{i=1}^{\delta} (p_i - p_j)^2 \right)^{1/2} \tag{5}$$

定义 4 用 $\mathrm{MinDist}(p, R)$ 和 $\mathrm{MaxDist}(p, R)$ 分别表示点 p 与最小边界矩形 R 之间的最小和最大距离, 则有如下定义

$$\mathrm{MinDist}(p, R) = \sum_{i=1}^{\delta} x_i^2$$

$$x_i = \begin{cases} r_i - p_i, & p_i < r_i \\ p_i - r_i', & r_i' < p_i \\ 0, & \text{others} \end{cases} \tag{6a}$$

$$\mathrm{MaxDist}(p, R) = \sum_{i=1}^{\delta} x_i^2$$

$$x_i = \begin{cases} r_i' - p_i, & p_i < (r_i + r_i')/2 \\ p_i - r_i, & \text{others} \end{cases} \tag{6b}$$

在 $\mathrm{MinDist}(p, R)$ 的定义中, 当点 p 在最小边界矩形 R 的左边时, 用最小边界矩形 R 的左顶点与点 p 的距离计算 MinDist; 当点 p 在最小边界矩形 R 的右边时, 用最小边界矩形 R 的右顶点与点 p 的距离计算 MinDist; 当点 p 在最小边界矩形 R 的内部时, 最小边界矩形 R 的左顶点与点 p 的距离为 0。在 $\mathrm{MaxDist}(p, R)$ 的定义中, 当点 p 在最小边界矩形 R 的中心点的左边时, 用最小边界矩形 R 的左顶点与点 p 的距离计算 MaxDist; 反之用最小边界矩形 R 的右顶点与点 p 的距离计算 MaxDist。

2.3 算 法

本文将离群数据挖掘算法分为两部分: 1) 求取数据集合的每个点的第 k 个最近邻与点的距离 $D^k(p)$; 2) 根据 $D^k(p)$ 挖掘离群数据。

算法 1 计算离群数据

输入: 经离散傅立叶变换后的 k 维空间上的时序点集合 P;

输出: 离群数据点集合 C。

Step 1: 设置离群数据点集合 C 初始值为空, MinDkDist (用于保存 D^k 的最小值) 初始值为 0;

Step 2: 将 P 中的每个点 p 和 MBR 插入 R-Tree;

Step 3: 对 P 中的每个点 p 计算其 $D^k(p)$, 调用算法 2;

Step 4: 如果点 p 的 $D^k(p)$ 即 p.DkDist 大于 MinDkDist, 将点 p 插入离群数据点集合 C;

Step 5: 如果离群数据点集合 C 中点的数目大于 n, 将 C 中最前面的点即 $D^k(p)$ 最小的点删去;

Step 6: 如果 C.numpoints() = n, 将 C 中最前面的点 $D^k(p)$ 最小的值更新为 MinDkDist, 返回 Step 4;

Step 7: 返回 C。

算法 1 中步骤 3 调用了算法 2 计算点 p 的 $D^k(p)$, 当计算越来越多点的 D^k 值时, D^k 值是单调减小的, 因此集合 C 中的离群数据点是按 D^k 值的升序排列, 即越在集合 C 中前面的数据点, 其 D^k 值越小。

算法 2 计算点 p 的 $D^k(p)$

输入: 时序点集合 P 中的点 p, k, D^k 的最小值 MinDkDist 和根结点;

输出: 点 p 的第 k 个最近邻与点 p 的距离 $D^k(p)$, 即 DkDist。

Step 1: 建立一链表 nodelist;

Step 2: 初始化 p.DkDist = ∞, 点 p 的 k 最近邻集合置为空;

Step 3: 当链表非空时, 若结点为叶子结点, 则对于结点中的每个点 q;

Step 4: 如果 p, q 间的距离 $\mathrm{dist}(p, q) < p$.DkDist;

Step 5: 点 q 进入点 p 的 k 最近邻集合 Ck, 如果 Ck 点的个数 Ck.points > k, 删除 Ck 中最前面的点; 如果 Ck 点的个数 Ck.points = k, 则 p.DkDist = $\mathrm{dist}(p, Ck.\mathrm{Top}())$;

Step 6: 如果 p.DkDist $\le$ MinDkDist, 返回算法 1, 否则返回 Step 4;

Step 7: 若结点不为叶子结点, 将结点的子结点加入链表, 并将结点链表按 MinDist 排序;

Step 8: 对链表中的每个结点, 如果 p.DkDist $\le \mathrm{MinDist}(p, \mathrm{Node})$, 则从链表中删除该结点, 返回 Step 3。

算法 2 通过对 R 树的结点计算点 p 的 $D^k(p)$, 这里建立一链表 nodelist, 当结点是叶子结点时, 计算叶子结点中存储的点 q 与点 p 的距离, 找到点 p 的 k 最近邻点, 并按其与点 p 的距离降序排, 即 Ck 中 q 与点 p 的距离最大的 k 近邻在 Ck 的最前面。当 Ck 中的点大于 k 时, 删除 Ck 中最前面的点, 当 Ck 中的点等于 k 时, 用点 p 与最远的近邻点的距离更新点 p 的 $D^k(p)$。当点 p 的 $D^k(p)$ 小于 MinDkDist 时, 点 p 不是离群数据点, 返回。因为在每一步中存储前 n 个计算的离群数据点, 记 $D_{\min}$ 是这些离群数据点的最小的 D^k, 如果计算点 p 的 $D^k(p)$ 时, $D^k(p)$ 始终小于 $D_{\min}$, 则点 p 不可能是离群数据点, 可以将其剪

除。若结点不为叶子结点, 将结点的子结点加入链表, 并将链表按MinDist 排序, 对链表中的每个结点, 若点 p 与结点中的MBR 间的距离小于 MinDist(p, Node) 时, 从链表中删除该结点。因为当点p 和R^* 树的一个结点存储的MBR 的距离超过 $D^k(p)$ 值时, 显然该结点下的子树中的所有点都不会在点 p 的 k 最近邻邻域内, 因此可以剪去这个与查询点 p 的 k 最近邻无关的点所在的子树。

3　仿真算例

运用本文算法对某钢铁企业的电力负荷时序数据库挖掘其中的离群数据。我们以负荷的功率数据为例, 取1998 年4 月的数据, 其时序曲线如图1 所示。

对上述时序数据运用离群挖掘算法, 根据电力

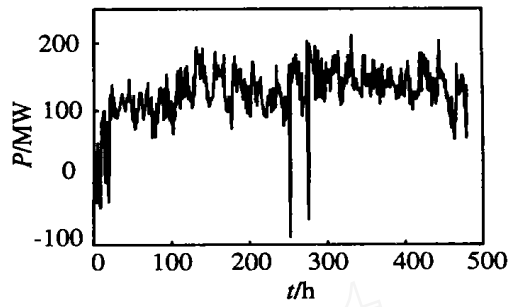
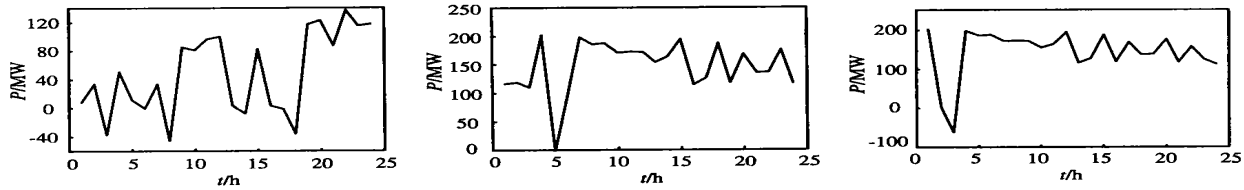


图 1　电力负荷功率曲线

负荷的特点, 以天即 24 h 为时间区间划分时序数据, 对时序子序列用DFT 变换到频域 k 维空间上的点, 取 $n = 3, k = 5$, 由文中的离群挖掘算法挖掘到离群程度最强的前3 个点, 分别表示第1, 10 和11 天的功率点, 将其进行DFT 反变换获得时域中的离群数据, 其结果如图2 所示。



(a)　第 1 天　　(b)　第 10 天　　(c)　第 11 天

图 2　电力负荷离群数据曲线

由仿真获得的电力负荷离群数据曲线中含有负值或零值功率的离群数据点, 这明显偏离了正常的功率值。通过查询该钢铁企业的电力能量管理系统(EMS) 数据库和相关的生产计划、生产实绩及设备检修数据库发现, 每当出现电力负荷离群数据时, 均有设备故障等情况的记录, 这表明了本文的时序数据离群挖掘算法的有效性。因此通过本文的电力负荷离群数据挖掘, 能够为电力管理和调度人员检测电力系统的异常, 并对其进行相应的异常分析提供有力的手段。

4　结　　论

时序数据的离群数据挖掘可以使人们发现潜在的有用信息。本文针对时序数据的离群数据挖掘, 提出了一种新的基于距离的时序数据离群挖掘算法, 该算法的距离概念可以度量离群数据程度。通过对某钢铁企业的电力负荷时序数据的离群挖掘, 证明了该算法的有效性。

参考文献(References):

[1] Fayyad U, Piatetsky-Shapiro G, Smyth P. From data mining to knowledge discovery: An overview [A]. Advances in Knowledge Discovery and Data Mining [C]. USA: AAAI/MIT Press, 1996.

[2] Barnet V, Lewis T. Outliers in statistical data [M]. New York: John Wiley & Sons, 1994.

[3] Edwin Knorr, Raymond Ng. Algorithms for mining distance-based outliers in large databases [A]. Proc of the VLDB Conf [C]. New York: 1998. 392-403.

[4] A Arning, Rakesh Agrawal, P Raghavan. A linear method for deviation detection in large database [A]. Int Conf on Knowledge Discovery in Databases and Data Mining [C]. Portland, 1996. 164-169.

[5] N Roussopoulos, S Kelley, F Vincent. Nearest neighbour queries [A]. Proc of ACM SIGMOD [C]. San Jose: ACM Press, 1995. 71-79.