

文章编号: 1001-0920(2006)05-0541-05

基于方形邻域的离群点查找新方法

黄添强^{1,2}, 秦小麟, 叶飞跃

(1. 南京航空航天大学 计算机科学与工程系, 南京 210016; 2 福建师范大学 数学与计算机科学学院, 福州 350007)

摘要: 提出一种基于密度的快速查找离群点的算法——基于方形邻域的离群点查找算法(ODBSN), 该算法把DBSCAN 算法的邻域改造成方形邻域, 并吸收基于网格算法的思想, 用密集的方形邻域快速排除非离群点; 用邻域扩张的思想代替网格划分克服了基于网格算法中“维灾”缺点; 同时用局部偏离指数指示离群点的偏离程度, 又具有识别精度高和偏离程度可度量的优点. 理论分析表明该算法性能优于著名的基于密度的算法, 实验表明, ODBSN 算法能在各种形状分布与各种密度的数据中有效地查找离群点, 速度明显优于LOF 与DBSCAN 算法.

关键词: 数据挖掘; 离群点; 方形邻域

中图分类号: TP311.13; TP392; TP18

文献标识码: A

New Approach Based on Square Neighborhood to Detect Outliers

HUANG Tian-qiang^{1,2}, QIN Xiao-lin, YE Fei-yue

(1. Department of Computer Science and Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China; 2 School of Mathematics and Computer Science, Fujian Normal University, Fuzhou 350007, China
Correspondent: HUANG Tian-qiang, E-mail: huangtianqiang2000@yahoo.com.cn)

Abstract: A new quick density-based approach to detect outliers, called outlier detecting based on square neighborhood (ODBSN), is presented. This algorithm changes the ϵ -neighborhood in DBSCAN to a square neighborhood and judges if the neighbors in the dense square neighborhood are not outlier. The algorithm partitions objects with square neighborhood, not with spatial grids, and thus does not cause “dimension curse”. The algorithm can indicate the degree of outlier with the local deviate factor, so the outlier can be identified exactly and the precision is measurable. Theoretical comparison shows that this method is more efficient than the well-known algorithm based on density, DBSCAN and LOF. Experimental results more efficient that the proposed approach can effectively identify outliers in databases within clusters that have different shape and varied density, and it is several times faster than the original DBSCAN and LOF algorithm.

Key words: Data mining; Outliers; Square neighborhood

1 引言

离群点查找是数据挖掘的一项重要技术, 其目标是发现数据集中少量异常的数据对象, 在金融欺诈识别、数据清洗、股票分析、恶劣天气预报等许多领域都得到广泛应用. 常见的查找离群点的数据挖掘方法有: 基于分布的、深度的、聚类的、距离的和密度的方法.

最著名和实用的方法是基于密度的方法, 如 LOF^[1]与DBSCAN^[2](也是基于聚类的方法)等. 基于密度的方法都必须计算每一点的邻域, 但查询某一点的邻域就要扫描整个数据库, 故查找速度不理想. 在聚类算法中基于网格的聚类方法是最快的算法之一, 该算法采用一个多分辨率的网格数据结构, 将空间划分为有限个数目的网格, 所有的聚类在网

收稿日期: 2005-02-02; 修回日期: 2005-04-18

基金项目: 国家自然科学基金项目(49971063); 国家“863”海洋监测主题子课题基金项目(2001AA 633010-04); 江苏省自然科学基金项目(BK2001045).

作者简介: 黄添强(1971—), 男, 福建莆田人, 博士生, 从事空间数据挖掘的研究; 秦小麟(1953—), 男, 江苏苏州人, 教授, 博士生导师, 从事安全数据库、空间数据库等研究.

格上进行。这种方法的主要优点是处理速度快,但也有处理精度不高和“维灾”的缺点。

本文提出了一种新的算法,该算法吸收了两者的优点,排除聚类点后发现离群点,并借鉴LOF^[11]方法用一个局部离群因子来指示对象偏离度的思想,采用了一个更为简单易行的局部偏离指数来对候选离群点进一步求精,使本算法既处理速度快,又有高的精确度。

2 相关研究

已有的离群点查找方法可分为以下几类:

1) 基于分布的方法在统计领域较为常见。人们用各种统计模型来测试,将偏离这些模型的对象作为离群点^[3-5]。但是,大多数分布模型只能直接应用于单变量特征空间,难以应用于多维空间。而且,这种模型要求预先知道数据的分布,但往往难以获得,需要进行耗时的测试来决定。

2) 基于深度的方法^[6]是一种基于计算几何的方法,这种方法通过计算不同层面的 k -凸面来查找离群点,凸面的外层认为是离群点。但该算法有个明显的缺点,即维度的灾难。

3) Knorr等^[7,8]提出了一个基于距离的离群点概念。在数据库中,若 $p\%$ 的对象与某对象的距离超过 d ,则这个对象为离群点。之后,有人提出 K -最邻近邻域的方法,Top- k 离群点等类似的概念^[9]。

4) 在查找的过程中,除了聚类剩下的就是离群点。聚类的方法可以用来查找离群点,如DBSCAN^[2],ROCK^[10],C2P^[11]等算法都可以用来查找离群点。

5) Breunig等^[11]提出了局部离群因子的概念LOF,这是一种基于密度的方法。通过数据空间的所有维度来计算对象的距离,进而计算对象的可达密度,最后通过局部的偏离度来判断离群点。

6) 有人提出了基于偏离的技术来查找离群点,这种方法检查对象的特征,将偏离这些特征的对象作为离群点^[12]。Aggarwal等^[13]用投影的方法来发现聚类,将高维空间数据投影到低维空间,局部区域中极低密度的点为离群点。Hu等^[14]提出了基于模式的方法,认为离群点有两种模式:偏离簇的稀疏点和偏离常规的密集点。He等^[15,16]提出了一种基于聚类的局部离群点因子来发现聚类,还提出了分类离群点的概念,他们认为离群点应该考虑分类标记。

3 基于方形邻域的离群点查找算法

点表示一个对象,故在下文中,点与对象的概念是等同的。为了描述简化,易于理解,下文以二维空间为例,这些概念与算法容易推广到高维空间。

3.1 与算法有关的定义

给定一个空间数据库 DB ,实数参数 ϵ (方形邻域的边长),正整数参数 M_{inP} (邻域中邻居个数的阈值)。

定义1 以对象 $p(x, y)$ 为中心, ϵ 为边长的方形区域内的对象集,即 $\{q(u, v) \in DB \mid |x - u| \leq \epsilon/2, |y - v| \leq \epsilon/2\}$ 称为点 p 的方形邻域,表示为 $SN_\epsilon(p)$ 。

定义2 与一个方形邻域有相邻点的8个方形邻域,称为邻近方形邻域,如图1所示。

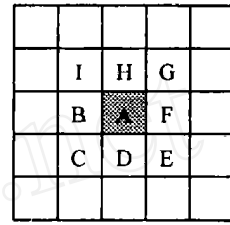


图1 邻近方形邻域示意图

定义3 点 p 的方形邻域中除自己以外的点称为点 p 的邻居。 $|SN_\epsilon(p)|$ 表示方形邻域的基数,即方形邻域内邻居数。

定义4 如果一个点 p 的方形邻域内有至少 M_{inP} 个点,则称方形邻域是密集的,点 p 为核心点。

定义5 如果点的方形邻域包含少于 M_{inP} 个点,则称这个方形邻域是稀疏的。如果一个点的方形邻域是稀疏的,但这个点是核心点的邻居,则称这个点为边缘点。

定义6 对象 p 的局部密度如下:

$$LD(p) = \frac{|N(p)|}{\sum_{o \in N(p)} \text{Dist}(p, o)}$$

其中: $\text{Dist}(p, o)$ 是点 p 与 o 的 n 维的欧氏空间距离, $|N(p)|$ 是点 p 的方形邻域中邻居的数目。显然,对象 p 的局部密度是对象 p 所有邻居到 p 的平均距离的倒数。

定义7 如果对象 p 的邻域不是密集的,即 $|N(p)| < M_{inP}$,则 p 的局部偏离指数定义为

$$LDI(p) = \frac{LD(o)/LD(p)}{|N(p)|}$$

对象 p 的局部偏离指数表示对象 p 偏离邻居的程度,它是 p 的所有邻居的局部密度的平均值与 p 的局部密度的比值。显然, p 的局部密度越小, p 的邻居的局部密度越大,则局部偏离指数越大。

这个偏离指数看起来与LOF^[11]的偏离指数类似,但它们是不同的。首先它不是针对所有的点,只是对稀疏邻域的对象有意义。故比LOF针对所有的点(包括聚类点)计算偏离指数更能体现“偏离”的

意义,更合理 其次,这个局部偏离指数不需要计算 LOF 中的可达距离等,比 LOF 偏离指数简单,又有表达对象偏离程度的能力

定义 8 稀疏领域中偏离指数 LDI 大于某个阈值的对象称为离群点

3.2 基于方形邻域查找的离群点算法

本文将基于密度和基于网格两种方法的优点融合在一起,产生一个较好的算法——基于方形邻域的离群点查找算法 (ODBSN),这个算法包含两个阶段:粗选与精选 主要思想如下:

1) 方形邻域粗选: 算法首先任选一个开始点,并计算它的方形邻域 若方形邻域为密集的,则此方形邻域内所有的点(包括这点本身与它的所有邻居)为聚类点; 否则方形邻域内所有的点为候选离群点 然后在这个方形邻域外任选一个邻近的方形邻域,这个方形邻域的中心点不一定存在,如果不存在则为虚拟点,继续计算,直到方形邻域内没有点或所有的点都已访问 接着再次任意选择一个未访问的开始点,这样一直到把所有的点划分为聚类点与初选点两类为止(分别作不同类别的标记,把它们看作不同的集合).

在上述划分中有可能把一些在聚类簇边缘的点也误划入到初选集中,故算法要进一步将其排除

2) 偏离指数精选: 算法计算初选集中所有的对象的偏离指数 LDI, LDI 大于某个阈值 f (通常与 LOF 算法取值类似) 的对象为离群点

Algorithm ODBSN (DB, ϵ , M in P)

```

1 While (!DB. Visited())
2 {  $p = \text{SelectUnvisitedPoint}(DB)$ ;
3  $PNB = \text{DB. SquareNeighbors}(p, \epsilon)$ 
4  $\text{NextP} = \text{PNB}$ 
5 While! (( $\text{NextP. isallvisited}$ )
   or ( $\text{NextP} = \text{Empty}$ ))
6 {if(  $|\text{NextP}| > M \text{ in } P$ )
7    $\text{DBNB}[\text{NextP}] = \text{ClusterLabel}$ ;
8   else
9     if(!  $\text{IsEdge}(\text{NextP})$ )
10       $\text{DBNB}[\text{NextP}] = \text{CandidateLabel}$ ;
11   end if;
12    $\text{NextP} =$ 
13      $\text{NextSquareNeighborhood}(\text{NextP})$ 
14 };
15 For every CandidateLabeled point
16   { $q = \text{SelectOnePoint}()$ ;
17     if (  $\text{LDI}(q) > f$  )
```

```

17        $\text{OutlierSet} =$ 
18          $\text{OutlierSet} \cup q$ ;
19   }
```

第 1~ 13 行是算法的粗选阶段,用于把对象划分成两个集合 第 1 行是一个循环,当所有的点划分结束时循环结束 算法在第 2 行中选择一个未访问的点 p . 第 3 行函数 $\text{DB. SquareNeighbors}()$ 计算 p 的邻域,如果正好遇到在方形邻域边上的点,处理如下: 如果本方形邻域内的邻居少于阈值,则此点归入本方形邻域; 否则,将它归入邻近方形邻域 第 4 行把邻居集合赋值给 NextP ,以便进入第 5~ 13 行的循环 第 5~ 13 行的循环是一个邻域扩张的过程,第 6~ 10 行的程序把邻居分成两类,聚类点或候选离群点 第 8 行的函数 $\text{IsEdge}()$ 判断对象是否为边缘点,第 11 行函数 $\text{NextSquareNeighborhood}()$ 选择下一个方形邻域: 这个函数随机选择一个邻近方形邻域(见定义 2),当这个邻近方形邻域没有点或所有的邻居都已访问,则选择下一个,一直到第 8 个(由定义可知邻近方形邻域最多 8 个). 第 14~ 18 行为算法的精选阶段 第 14 行为 For 循环选择每一个初选点(候选的离群点),第 16 行计算这些点的偏离指数并进行判断,第 17 行把偏离指数大于阈值 f 的点加入离群点集合

在已有的基于密度的离群点查找方法(如 LOF 与 DBSCAN) 中,算法为了查找离群点必须计算每个点的邻域,这些邻域与本文的方形邻域类似 邻域查找是较费时的操作,应尽量避免这种操作 由网格算法得到启示,当一个点的方形邻域是密集的,这个点和它所有的邻居不可能是离群点,所以当计算一个方形邻域后,发现它是密集的,则不必对其邻居进行方形邻域计算,这样提高了效率 同时,本文利用局部偏离指数来选择离群点,使算法又具有很好的精度

为了便于理解算法,图 2 给出一个具有 2 个聚类和 4 个离群点的示例 聚类为簇 A 与簇 B,点 a, b, c, d 为离群点 假设图中的方形是算法选择的方形邻域,图中有实线方形表示此方形邻域是密集的,没有离群点; 虚线方形表示此方形邻域是稀疏的,其中

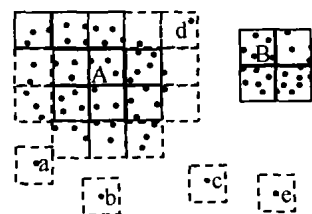


图 2 ODBSN 算法解析示意图

的点将被标为初选点 算法的偏离指数精选阶段将计算这些点的偏离指数, 超过阈值的点为离群点

4 算法的时间复杂性评估与理论比较

算法最复杂的操作是方形邻域查询操作, 邻域查询操作的复杂性为 $O(n)$, 若使用空间索引, 如 R^+ -trees, 则复杂性为 $O(\log n)$. 本算法用方形邻域对空间进行分割, 对于密集的方形邻域内的邻居不计算它们的邻域 设密集的方形邻域有 k 个, 初选点为 c 个 ($c \ll n$), 则计算的邻域为 $(k + c)$ 个, 复杂性为 $(k + c)O(\log n)$, $(k + c)$ 的大小与方形邻域的边长和数据的密集度有关 在一定的范围内, ϵ 越大, 算法查询的邻域越少; 数据越密集, 算法查询的邻域越少. 通常, ODBSN 算法查询的邻域数只有其他基于密度算法的几分之一, 故算法 ODBSN 比其他算法快几倍

本文算法用方形邻域对空间进行分割, 但这种分割与基于网格的方法有重要的区别: 基于网格的方法要求对所有的空间进行分割, 无论是否存在数据对象, 都需要过多的空间网格, 因此造成维数的灾难; 而本文方法利用方形邻域对空间进行扩张, 而不是对所有的空间进行分割, 所以不会有维灾的问题. 同时, 基于网格的算法不能对对象的偏离程度进行度量, 而 ODBSN 算法具有这个性能

基于密度的离群点挖掘算法, 如 LOF^[1] 与 DBSCAN^[2], 都需要对每个点进行非常耗时的操作, 即邻域查找操作. LOF 算法通过计算每一点离群因子来识别离群点, 故必须进行所有点的邻域查询. DBSCAN 通过计算 ϵ 邻域来计算可达距离, 以查找聚类点与离群点, 故 DBSCAN 算法必须计算数据库中每一点的 ϵ -邻域. ODBSN 算法并不需要计算每

个点的邻域, 因为当一个点的邻域是密集的, 它的所有邻居将不要再计算邻域, 所以本文算法大大地提高了计算的效率

5 实验评价

所有的实验在 CPU 为 Pentium 4 2.2G, 内存为 256MB, 系统为 Windows XP professional 的个人电脑上进行

5.1 影响效率的因素评价

方形邻域的边长越大, 一次领域扩张所处理的对象越多, 显然, 邻域查询的操作越少, 但这个规律不一定总成立. 本文用含有各种密度的 8 000 个点的人工数据库来分析其的性能, 图 3 表明了算法 ODBSN, DBSCAN 和 LOF 在该数据库中进行邻域

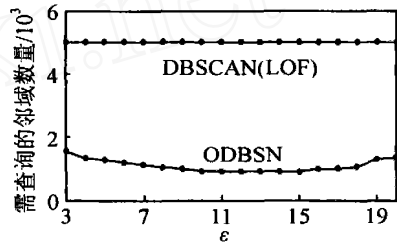


图 3 ϵ 的取值与需要查找邻域数量的关系

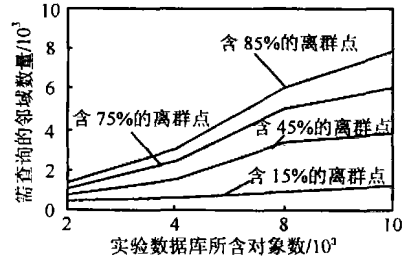


图 4 算法查找的方形邻域数与所含离群点百分比的关系

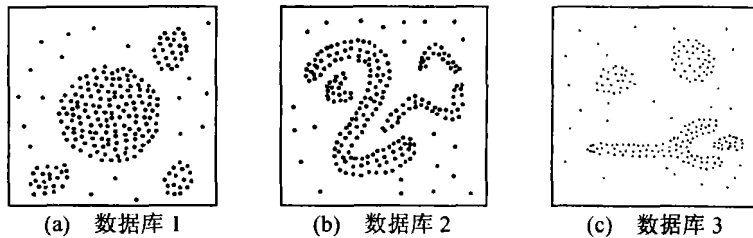


图 5 3 个不同形状与密度的合成数据

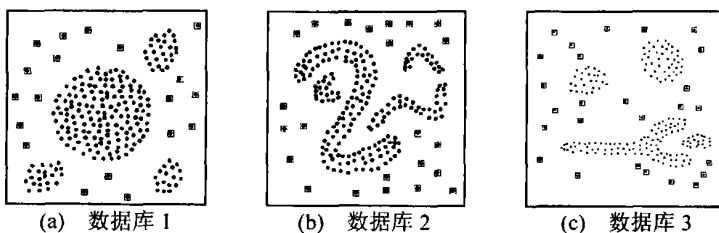


图 6 算法 ODBSN 查找到的离群点示意图

查询的次数与 ϵ 取值的关系。因为算法 DBSCAN 与 LOF 要查询所有点的 8 000 个邻域, 故邻域查询次数成一直线。在 ϵ 在 30 到 200 的范围中, 随着 ϵ 值的增加, 邻域查找次数减少; 但当 ϵ 大于 120 时, 邻域查找的次数开始增加。从图中可以看出, 算法 ODBSN 查询的邻域明显少于其他两个算法。

另一个主要的影响因素数据的密度, 在这里粗略地用离群点所占的比例来表示。图 4 表明对包含 2 000, 4 000, 8 000, 10 000 个对象的数据库进行离群点查找时算法需要查询邻域的次数。显然, 邻域查询次数随离群点所占的比例增加而增加, 随数据库的增大而增大。

5.2 算法的有效性与效率评价

为了说明本算法的有效性, 用文献[2]中用到的 3 个合成数据库, 再适当增加一定数量的离群点来验证算法的有效性。如图 5 所示, 在第 1 个数据库中有 4 个不同半径的球形与一些离群点; 第 2 个数据库中有 4 个非凸形聚类与一些离群点; 第 3 个数据库包含 4 个不同形状的聚类与一些离群点。

离群点查找结果如图 6 所示, 带框的点为离群点。这里方形邻域的边长设为 3.0, 密集邻域的最少邻居个数为 5, 偏离指数取 1.2。算法 ODBSN 能够正确识别全部的离群点, 在第 3 个数据库中分别发现全部的 19, 23, 26 个离群点。

用含有 5 000, 8 000, 10 000, 20 000, 50 000, 80 000, 100 000 个点的模拟数据进行实验。算法 ODBSN 方形邻域查询参数为: $\epsilon = 20, M \ln P = 10, LD I = 1.2$; 算法 DBSCAN 邻域查询的参数分别设为: $\epsilon = 6, M \ln P = 10$; 算法 LOF 的参数 $M \ln P = 10, LOF > 1.2$ 。图 7 表明数据库规模与计算时间的关系, 可以看出算法 ODBSN 的计算时间随数据的量接近线性地增加, 计算时间明显少于 LOF 与 DBSCAN。

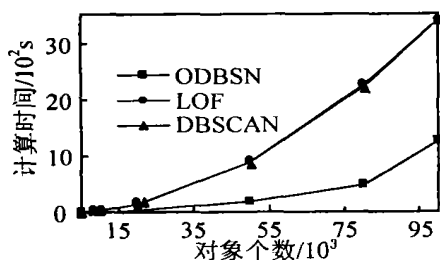


图 7 算法 ODBSN 与算法 LOF, DBSCAN 的时间效率比较

6 结 论

已有的基于密度的离群点查找方法都需要进行邻域查找的操作, 这种操作较为耗时, 是算法效率高低的关键性部分。本文吸收了网格算法、DBSCAN

算法与 LOF 算法的优点, 提出了一个新的基于密度的算法, 成倍地减少了邻域查询的次数, 使离群点查找的速度比已有的基于密度的离群点查找算法提高数倍。本算法用邻域扩张的思想代替网格划分可以处理高维数据, 用局部离群指数来克服网格算法精度低的缺点, 使它具有较多优良的性能。理论分析与实验验证了算法的可行性与效率。

参考文献 (References)

- [1] Breunig M M, Kriegel H P, Ng R T, et al. LOF: Identifying Density-based Local Outliers[A]. *Proc of SIGMOD '00* [C]. Dallas, 2000: 427-438.
- [2] Ester M, Kriegel H P, Sander J, et al. A Density-based Algorithm for Discovering Clusters in Large Spatial Databases[A]. *Proc of KDD '96* [C]. Portland OR, 1996: 226-231.
- [3] Barnett V, Lewis T. *Outliers in Statistical Data* [M]. New York: John Wiley, 1994.
- [4] Hawkins D M. *Identification of Outliers* [M]. London: Chapman and Hall, 1980.
- [5] Rousseeuw P J, Leroy A M. *Robust Regression and Outlier Detection* [M]. New York: John Wiley and Sons, 1987.
- [6] Johnson T, Kwok I, Ng R T. Fast Computation of 2-dimensional Depth Contours[A]. *Proc KDD* [C]. New York: AAAI Press, 1998: 224-228.
- [7] Knorr E, Ng R. A Unified Notion of Outliers: Properties and Computation[A]. *Proc of the Int Conf on Knowledge Discovery and Data Mining* [C]. New York: AAAI Press, 1997: 219-222.
- [8] Knorr E, Ng R. Algorithms for Mining Distance-based Outliers in Large Datasets[A]. *Proc 24th VLDB Conf* [C]. New York: Morgan Kaufmann Publisher, 1998.
- [9] Ramaswamy S, Rastogi R, Kyuseok S. Efficient Algorithms for Mining Outliers from Large Data Sets[A]. *Proc of SIGMOD '00* [C]. Dallas, 2000: 93-104.
- [10] Guha S, Rastogi R, Kyuseok S. Rock: A Robust Clustering Algorithm for Categorical Attributes[A]. *Proc of ICDE '99* [C]. Sydney, 1999: 512-521.
- [11] Nanopoulos A, Theodoridis Y, Manolopoulos Y. C2P: Clustering Based on Closest Pairs[A]. *Proc of VLDB '01* [C]. Rome, 2001: 331-340.
- [12] Arning A, Agrawal R, Raghavan P. A Linear Method for Deviation Detection in Large Databases[A]. *Proc of KDD '96* [C]. Portland OR, 1996: 164-169.
- [13] Aggarwal C, Yu P. Outlier Detection for High Dimensional Data[A]. *Proc of SIGMOD '01* [C]. Santa Barbara, 2001: 37-46.

(下转第 554 页)

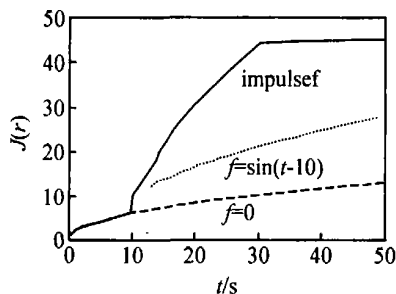


图3 残差评价信号

情况进行了 Simulink 仿真 图 1 给出了故障为脉冲信号时的残差信号; 图 2 为故障是正弦信号的情况

假设 $J(r) = \sqrt{\int_0^t r^T(\tau) r(\tau) d\tau}$ 作为残差评价函数, 图 3 给出了残差评价函数的变化曲线 当阈值选取为 $J_{th} = \sup_{f=0} J(r) = 13$ 时, 仿真结果表明, 本文给出的脉冲、正弦故障分别在发生 1.7 s 和 3.7 s 后可检测到

5 结 语

在系统观测性矩阵满足一定的秩的条件下, 可以通过广义坐标变换, 将线性时滞系统转换为无状态时滞的表达形式 根据新得到的系统模型, 选用基于观测器的故障检测滤波器作为残差产生器, 将时滞系统的故障检测转换为仅包含未知输入时滞、故障时滞的线性时不变形式, 从而可应用针对 LTI 系统的研究成果设计时滞系统故障检测滤波器 本文将 LTI 系统故障检测滤波 H_∞ 最优化方法应用于具有状态和输入时滞的情况, 得到了线性时滞系统 FDF 设计的一种新方法 算例进一步验证了所提出方法的有效性

参考文献(References)

- [1] Chen J, Patton R J. *Robust Model-based Fault Diagnosis for Dynamic Systems* [M]. Boston: Kluwer Academic Publishers, 1999.
 [2] Frank P M, Ding Steven X, Koepfen Seliger B.

Current Development in the Theory of FD I[A]. *Proc of the SA FEPROCESS 2000* [C]. Budapest: Elsevier Science, 2000: 16-27.

- [3] Kinnaert M. Fault Diagnosis Based on Analytical Models for Linear and Nonlinear Systems—A Tutorial [A]. *Proc of the SA FEPROCESS 2003* [C]. Washington: Elsevier Science, 2003: 37-50
 [4] Jiang Bin, Staroswieck M, Cocquempot V. Fault Identification for a Class of Time-delay Systems [A]. *Proc of American Control Conf* [C]. Anchorage: Elsevier Science, 2002: 8-10
 [5] Jiang Bin, Staroswieck M, Cocquempot V. H_∞ Fault Detection Filter for a Class of Discrete-time Systems with Multiple Time Delays [A]. *Proc of 15th IFA C World Congress* [C]. Barcelona, 2002: 770-775
 [6] Ding Steven X, Zhong M Y, Tang B Y. An LM I Approach to the Design of Fault Detection Filter for Time-delay LTI Systems with Unknown Inputs [A]. *Proc of the ACC* [C]. Arlington: Elsevier Science, 2001: 2137-2142
 [7] Zhong M Y, Ding Steven X, Zhang C H, et al. Fault Detection Filter Design for LTI Systems with Time Delays [A]. *The 42th IEEE CDC* [C]. Hawaii: IEEE Press, 2003: 1467-1472
 [8] Hou M, Zitek P, Patton R J. An Observer Design for Linear Time-delay Systems [J]. *IEEE Trans on Automatic Control*, 2002, 47(1): 121-125
 [9] Ding Steven X, Ding E L, Jeinsch T. A New Optimization Approach to the Design of Fault Detection Filters [A]. *Proc of the SA FEPROCESS 2000* [C]. Budapest: Elsevier Science, 2000: 250-255
 [10] 钟麦英, 张承慧, Ding Steven X. 一种鲁棒故障检测与反馈控制的最优集成设计方法 [J]. *自动化学报*, 2004, 30(2): 294-299.
 (Zhong M Y, Zhang C H, Ding Steven X. An Optimization Approach to Feedback Controller and Robust Fault Detection Filter Integrated Design [J]. *Acta Automatica Sinica*, 2004, 30(2): 294-299.)

(上接第 545 页)

- [14] Hu T, Sung S Y. Detecting Pattern-based Outliers [J]. *Pattern Recognition Letters*, 2003, 24(16): 3059-3068
 [15] He Z, Xu X, Deng S. Discovering Cluster-based Local Outliers [J]. *Pattern Recognition Letters*, 2003, 24(9-

10): 1642-1650

- [16] He Z, Xu X, Huang J Z, et al. Mining Class Outliers: Concepts, Algorithms and Applications in CRM [J]. *Expert Systems with Applications*, 2004, 27(4): 681-697.