孤立点挖掘在教务管理中的应用研究
Research and Application of Outlier Mining in Educational Adminstration

黄万华[1]，陆声链[2]，林士敏[2]
Huang Wanhua[1], Lu Shenglian[2], Lin Shimin[2]
(1. 广西师范大学教务处，广西桂林 541004；
2. 广西师范大学数学与计算机科学学院，广西桂林 541004)
(1. Dean's Office, Guangxi Normal Univ., Guilin, Guangxi, 541004, China;
2. Dept. of Comp. Sci., Guangxi Normal Univ., Guilin, Guangxi, 541004, China)

摘要：孤立点挖掘是一个重要的知识发现任务，在介绍孤立点及其挖掘算法的基础上，利用孤立点检测方法对教务管理系统中积累的数据进行分析，并提出基于距离和的孤立点检测算法。实验结果分析表明，该算法降低了检测过程对用户设置阀值的要求，在时间复杂度上，稍微优于循环一嵌套算法。

关键词：孤立点 数据挖掘 教务管理
中图法分类号：TP311.6

Abstract: Outlier Mining is an important task in knowledge discovery. The knowledge of outlier and the algorithms for detecting outliers are introduced. An algorithm based on distance sum is proposed. The test result shows that the algorithm presented can reduce requirements of thresholds to users in the detection. This algorithm is a bit better than the nested-loop algorithm in the time consumption.

Key words: outlier, data mining, educational adminstration

目前，很多大学的在校生人数都已达到上万甚至十几万的规模。学校运行着各种各样的软件系统，如学籍管理、成绩管理、人事管理、教务管理等管理信息系统，这些系统中的数据库积累了大量的数据。但由于缺乏信息意识和技术，管理人员只能够通过简单的统计或排序等功能获得表面的信息，隐藏在这些大量数据中的信息一直没有得到应用。

数据挖掘研究的是从大量数据中发现有用的知识。目前，数据挖掘主要应用于商业尤其是电子商务，而对其在高等教育领域的研究和应用在国内尚不多见[1]。利用这些数据理性地分析学校各方面工作的成效特别是学生培养过程中的得失变得十分重要，对高校教学管理的决策支持也将是十分有意义的。目前，教学管理系统中的数据挖掘大多利用关联分析或者分类分析，以发现一些大的模式[2,3]。但关联规则在发现大的规则的同时也会忽略那些不经常出现的情况，有时这些例外情况更应该引起教育决策者的注意。本文尝试利用孤立点检测方法，对教务管理系统中积累的数据进行分析，发现那些值得注意的例外对象，从而为教学管理者提供决策支持。

2003-12-04 收稿。

1 孤立点挖掘

孤立点检测是数据挖掘中一个重要方面，用来发现“小的模式”(相对于聚类)，即数据集中间显著不同于其它数据的对象。这些对象通常被忽略或视为噪音。一个人的噪音可能是另一个人的信号。在很多应用里，例外事件常常比普通的事件更有意义[4]。在国外，孤立点检测大多用于电信和信用卡诈骗检测、贷款审批、医药研究、天气预报、电子贸易中的犯罪活动检测，甚至在NBA比赛和NHL(Natinal Hockey League)数据中，孤立点检测都有其应用。在数据仓库领域，孤立点检测被用来发现不一致的数据，提高数据质量。

在孤立点检测中，主要是解决两个问题：什么是孤立点和如何发现孤立点。对第一个问题，Hawkins[5]给出了孤立点的一个本质性的定义：孤立点是数据集中与众不同的数据，使人怀疑这些数据并非随机偏差，而是产生于完全不同的机制；聚类算

法对孤立点的定义是：孤立点是聚类嵌于其中的背景噪声；而孤立点检测算法则认为孤立点是既不属于聚类也不属于背景噪声的点，它们的行为与正常的行为有很大不同。

为有效地挖掘孤立点，研究者们根据孤立点存在的不同假设，开发了许多孤立点检测算法，大体可以分为基于统计的算法、基于距离的算法[5]、基于偏离的算法[4]、基于密度的算法[6]。由于很多算法对高维数据异常检测效果不理想，Aggarwal 和 Yu[7]提出了一个高维数据异常检测的方法，通过将高维数据投影到低维空间，然后在低维空间中发现异常。他们采用遗传优化算法，获得了良好的计算性能。

1.1 基于统计的孤立点挖掘

在统计界，孤立点已被广泛研究。统计意义上的孤立点定义为与给定的分布或统计模型的显著性差异超过某个阀值的数据。这种检测算法中，首先假定数据服从某个分布模型，然后采用不一致性检测发现孤立点。这种方法的主要缺点在于需要预先指定（假定）数据的分布模型，这往往比较难，而且现实数据也往往不符合任何一种理想状态的数学分布；即使在低维时的数据分布已知，在高维情况下，估计数据点的分布是极其困难的。

1.2 基于距离的孤立点挖掘

Knorr 和 Ng[5]提出了基于距离的孤立点定义：在数据集 S 中，对象 o 是一个孤立点，仅当 S 中至少有 p 部分对象与 o 的距离大于 d。换句话说，如果 o 在 d 范围内有不多于 M 个邻居，则 o 是一个带参数 p 和 d 的 $DB(p,d)$ 孤立点（n 为数据对象的个数，$M = n^*(1-p)$）。Rastogi 和 Ramaswamy[6]给出了另一个孤立点定义：孤立点是数据集中 n 个与其 k 个最近邻居的平均距离最大的对象，称为 D_n^k 孤立点。

至今已开发了不少用于挖掘基于距离的孤立点算法，包括基于索引的算法、循环一嵌套（nested-loop，NL）算法、基于单元（cell-based）的算法等。最近，Bay，et al[7]提出了一种基于随机抽样的检测方法，他们仍然采用 Rastogi 和 Ramaswamy 的孤立点定义，但他们利用随机抽样的方法减少了寻找 k 近邻的范围，在实验数据上获得了几乎线性的计算复杂度。

1.3 基于偏离的孤立点挖掘

Arning 和 Agrawal 在 1996 年提出了“序列异常”（sequential exception）的概念，他们的算法有优异的计算性能，其复杂度与数据集大小呈线性关系。但是序列异常对异常存在的假设太过理想化，对现实复杂数据效果不太好，所以该方法并没有得到普遍的认可。

1.4 基于密度的孤立点挖掘

Breunig 等[8]提出了局部异常的概念，指出当存在不同密度数据集的情形下，$DB(p,d)$ 孤立点定义往往会遗漏一部分异常数据。他们进一步总结出数据是否异常不仅仅取决于它与周围数据的距离大小，而且跟邻域内的密度状况有关。在他们的方法里，给每个数据赋予一个局部异常因子（Local Outlier Factor，LOF）的属性，作为数据异常程度的度量。LOF 越大，其作为孤立点的理由越充分。同时，LOF 屏弃了以往的方法中数据对象非此即彼的概念。

由于很多算法对高维数据异常检测效果不理想，Aggarwal 和 Yu[9]提出了一个高维数据异常检测的方法，通过将高维数据投影到低维空间，然后在低维空间中发现异常。他们采用遗传优化算法，获得了良好得计算性能。

某些聚类算法也具备发现孤立点的能力。在聚类中，那些数据点很少的类或者不能聚类的数据点即为孤立点。但在聚类算法中，孤立点更多地被作为噪声处理，而且这些算法的出发点是优化聚类效果，而不是孤立点检测。

2 教务管理系统中的孤立点分析

在教务管理系统中，积累了大量的院系信息、人事信息、课程信息、选课信息、成绩、学生奖励和处分信息。院系信息包括院系代号、名称、教职工人数、本科生人数等基本信息；人事信息包括姓名、性别、部门等；课程信息包括课程名称、课时、学分等；选课信息包括选课学生的学号、专业、年级等；成绩数据包括学号、课程名称、学分、绩点等；奖励信息包括奖励时间、奖励名称等；处分信息包括处分结果、处分原因等。可以对这些信息进行关联规则挖掘、类描述、聚类分析，找出其中隐藏的规则（知识）。

教学系统中的数据挖掘大多利用关联分析或者分类分析，以发现一些大的模式。如通过分析学生的成绩，可能发现“高等数学”成绩好的学生，其“C 语言程序设计”课程的成绩也好。但关联规则在发现大的规则的同时也会忽略那些不经常出现的情况，有时这些例外情况更应该引起教育决策者的注意。孤立点挖掘就是专门为发现例外对象而开发。本文的主要思想是利用孤立点检测方法，找出教务管理信息中有趣的例外对象，这些对象往往容易被关联规

则挖掘和分类所忽略。

在教务管理系统中，可以将检测的数据类型分为相关性数据、随机性数和时间序列数据。

2.1 相关性数据

相关性数据是指在数据对象中，各个属性之间存在着某种联系。如对专业这个对象，显然它的专业类别、教师人数、学生人数、实验室数之间有着某种联系。一个文科专业有20名教师、2000名学生、2个实验室，可以认为是正常的，而一个理科专业在同样的条件下则可能是不合理的，因为它的实验室太少了。在教务管理系统中，相关性数据的孤立点检测可用于以下分析：(1)教师工作量分析。可以抽取教师的性别、年龄、学历、职称、职务，课时、科研工作量等特征，利用孤立点检测出异常的教师。(2)专业分析。选取在校生人数、新招生人数、毕业生签约人数、考研上线人数、教师人数、教授人数、博士人数等作为检测特征，以发现值得注意的专业，为制定招生计划、增加或者减少教师等提供决策支持。

2.2 随机性数据

随机性数据是指在数据对象中，对象的各个属性之间不存在联系，或者其联系可以忽略。如要对学生的选课情况进行分析时，选取他们的必修课、任选课、限选课、辅修课、综合素质课的选课数作为特征，这些特征之间并没有很大的关系，这类数据称为随机性数据。

2.3 时间序列数据

时间序列数据指在数据对象中，对象的各个属性之间存在着时间上的联系。考虑将学生四个学期所修的学分作为一个数据对象{14,16,11,17}，则这四个数据之间就存在着时间上的先后次序。时间序列类型数据的孤立点检测可以有以下的应用：(1)学生成绩分析。选取学生各个学期的平均绩点作为特征值，可以发现异常的学生，如进步较快的学生、成绩下降很明显的学生、成绩极不稳定的学生；若选取学生在校期间各个学期大学英语考试成绩和各次参加大学英语四六级考试的成绩作为特征，可以用来检测异常模式，如有作弊嫌疑的学生，借以反映考场纪律。(2)课程分析。用各个学期的选课人数作为特征值，以发现例外的课程，为课程设置、制定教学计划提供参考信息。

时间序列类型数据的孤立点检测还有很多应用，如教师工作量、科研经费的使用情况、教师教学质量、院系的奖励处分记录、资金流动等异常模式的发现。

3 应用实例与分析

以学生成绩分析为例，选取我校2000级2672名学生某个学期的成绩作为测试集，选取四个指标：选课门数、总成绩、总学分和平均学分绩点作为检测属性。

这是一个多变量孤立点检测问题，我们采用基于距离的孤立点挖掘方法。但无论是$DB(p,d)$孤立点定义[5]，还是D_n^k孤立点定义[8]都需要用户设置参数。一般地，用户没有关于这些参数的任何知识，实验也表明，检测算法对这些参数相当敏感。因此，我们提出了用对象间的距离和来判别孤立点的算法。

3.1 基于距离和(distance sum-based，DS)的孤立点检测算法

在D_n^k孤立点定义中，通过对象与其最近的k个邻居的平均距离来发现孤立点。算法需要设置参数k和查找每个对象的k个最近邻居。如果对每个数据对象，计算其与数据集中所有其它对象的距离之和，并以此来判别孤立点，则可以消除了用户设置参数k的要求。容易知道，为检测基于距离和的孤立点，算法将需要n^2次的数据对象间的距离计算，考虑到教务管理系统中的数据集一般较小，算法仍可以在较短的时间内返回结果。

设M为用户期望的孤立点个数，则距离之和最大的M个对象即为孤立点。基于距离和的孤立点检测算法如下：

```
procedure FindOutlier(db,M,O)
  //input:数据集 db;
  //output:O;孤立点集
  for i=1 to db.size
    read _ nextRecord(db,o);//将数据集中第 i 条记录读到 o 中
    //计算 o 与数据集中所有对象的距离
    for j=1 to db.size
      read _ nextRecord(db,q);//将数据集 sdb 中第 j 条记录读到 q 中
      pi=pi+distance(o,q);//计算 o 与其它对象的距离和
    next
  next
  getOutlier(O,M);//取得 pi 最大的 M 个对象
  return O
end
```

算法中距离的计算使用的是绝对距离，绝对距离又称曼哈顿距离，其定义如下：

$$d_{ij}=\sum_{k=1}^{m}|x_{ik}-x_{jk}|。$$

当然，也可以其它的距离量度函数，如欧氏距离或兰氏距离。

3.2 实验设计

为了验证DS算法的有效性，本文首先用文献[5]提出的循环一嵌套(Nested-Loop)算法(简称NL算法)进行孤立点检测，然后利用DS算法在同样的数据集上进行挖掘。

NL算法的基本思想是：将内存缓冲区分成大小相同的两块，第一块用来保存从没在该块保存过的数据块，同时把数据集划分为若干块。算法每次将一个数据块读到内存缓冲区的第二块中(第一次首先将第一个数据块读到第一块)，然后计算这两块中的每对对象间的距离。对第一块中的每个对象t，用一个变量count记录它的d距离邻居，一旦它的d距离邻居数超过M，则计数停止，开始处理下一个对象。如果计算完第二块中的对象后，t的count值仍然不大于M，则下一次将另一个数据块读进内存缓冲区的第二块后，继续用t去跟新读进的对象计算距离，并累计其count值。

用NL算法进行实验时，我们设计了两组实验，第一组固定参数p的值，改变d的值；另一组固定d值，改变p值。以此来察看p和d对检测结果的影响。实验结果如下：

表1 NL算法的检测结果

算法	p	d	孤立点(学号)	时间*(s)
NL	0.995	0.7	2000025045,2000030174 2000040010,2000060060 2000070223,2000075025 2000075120,2000095015 2000190064	288
NL	0.995	0.76	2000025045,2000040010 2000075025,2000095015 2000190064	302
NL	0.995	0.80	2000040010,2000095015 2000190064	295
NL	0.995	0.83	2000095015	306
NL	0.998	0.80	2000010008,2000010103 2000010154,2000030174 2000040010,2000050028 2000050085,2000075025 2000075120,2000095015 2000190064	276
NL	0.997	0.80	2000040010,2000095015 2000190064	324
NL	0.992	0.80	2000095015,2000190064	275

* 这里的时间不包括I/O时间。

用我们提出的DS算法时，取M=5，算法返回距离p_i值最大的5个学生，如下表所示。时间消耗是263s(不包括I/O时间)。

表2 DS算法的检测结果

学号	选课门数	总成绩	总学分	平均学分绩点
2000095015	13	6.7	14.0	1.06
2000190064	6	1.5	2.0	1.50
2000040010	35	6.7	13.0	1.22
2000075120	13	8.1	10.0	1.58
2000075025	18	8.5	13.0	1.56

3.3 实验结果分析

数据集中四个指标的均值分别为：46.43、127.48、123.83和2.756。2000095015、2000190064和2000075120都是孤立点，这容易理解，因为它们四个指标与相应的均值有很大的偏离，2000040010的选课门数虽然接近均值，但他的其它3个指标与均值的偏差也很大。

上面的实验结果表明，我们的定义与基于$DB(p,d)$的孤立点定义有着相似的结果，并且，DS算法还给出了孤立点的孤立程度的量的表示。同时也可以看到，$DB(p,d)$定义对p和d异常敏感，要求很高的精确度。为检测出孤立点，用户可能要经过大量的试探和失败。而在我们的算法里，总是返回值最大的M个对象。实际上，在计算完所有的后，用户可以任意指定M值，来察看每个对象的孤立程度。这正是我们对孤立点的定义的关键，因为检测的结果只是提供用户一个参考，只有用户才能最后确定真正的孤立点。

在时间复杂度上，DS算法稍微优于NL算法，这里还没有考虑NL算法为检测孤立点而进行的多次测试时间。

4 结束语

本文介绍了孤立点及其挖掘算法，探讨了孤立点检测在教务管理系统中的应用。在分析了现有的孤立点定义的缺点的基础上，给出了一个新的标识孤立点的定义和算法，并用实验验证了该定义的有效性。应该指出，我们的定义和算法也存在不足，比如对时间序列类型的数据，这样的定义就不适用了；另一方面，算法对检测出来的例外对象没有解释功能，更好的检测算法除了给出例外的程度以外，还应该可以向用户说明产生例外的原因。我们将在进一步的工作中继续研究这些问题。

(下转第162页)

型与流程原模型整合在一起，并提供对多层事务的支持，以使用户可以灵活定制事务的粒度。在运行态利用日志提供向后恢复的路径的半动态方法而建立起来的事务模型，该模型具有以下优点：

(1)效率高。克服传统事务模型中数据依赖和冲突关系的复杂性，从而严重降低系统并发执行性能的缺陷。利用改进的LOL确认机制，实现事务数据的有效分离，不会出现读脏数据的情况，把事务期间对数据的读写操作次数降到最低，大大降低复杂度，提高了效率；

(2)易于管理和控制。把原模型与补偿模型整合在一起，故障发生点自然获得，相关数据易于传递；

(3)可精确回滚。利用日志提供向后恢复的路径，即使事务中包含如循环等复杂的结构也不会造成干扰；

(4)支持嵌套事务，提高并行性和减小事务颗粒度，提供更加灵活的控制，满足用户多样化的需求。

在向后恢复的路径方面，其实并不是一定要严格按照执行的次序的逆序来进行的，因为有些活动并没有依赖关系，如果能使这些没有依赖关系的补偿活动能够并行执行，则效率会更高，如何实现并行的向后恢复是需要进一步研究的问题。

参考文献：

1 Georgakopoulos D, Hornick M, Sheth A. An overview of workflow management: From process modeling to workflow automation infrastructure. Distributed and Parallel Databases, 1995, 3: 119～153.

2 Sheth A, Rusinkiewicz M. On transactional workflows. In: Special Issue on Workflow and Extended Transaction Systems IEEE Computer Society, Washington D C, 1993.

3 Derks W, Dehnert J, Grefen P, et al. Customized atomicity specification for transactional workflows, cooperative database systems for advanced applications 2001. The Proceedings of the International Symposium, 2001, (s): 140～147.

4 Ding K, Jin B H, Wei J, et al. New model and scheduling protocol for transactional workflows, computer software and applications conference 2002. Proceedings 26th Annual International, 2002, (s): 920～927.

5 Mühlberger R, Orlowska M E, Kiepuszewski B. Backward Step: the right direction for production workflow systems. Distributed Systems Technology Centre Technical Report, 1998.

6 Kiepuszewski B, Mühlberger R, Orlowska M. FlowBack: providing backward recovery for workflow systems. Proceedings of 1998 ACM SIGMOD International Conference on Management of Data, 1998.

7 Liu C, Orlowska M, Lin X, et al. Improving backward recovery in workflow systems. In: 7th International Conference on Database Systyems for Advanced Applications, 2001.

8 Alonso G, Agrawal D, A El Abbadi, et al. Advanced transaction models in workflow contexts. In: 12th International Conference on Data Engineering, New Orleans, 1996.

(责任编辑：黎贞崇)

(上接第158页)

参考文献：

1 任承业. 校园信息系统中数据挖掘的研究与应用. 广州：暨南大学(硕士学位论文). 2003.

2 傅国强. 基于数据仓库的校园管理与决策支持系统的设计. 微机发展, 2003, 1: 82～84.

3 陶 兰, 王保迎, 吕建军. 数据挖掘技术在高等学校决策支持中的应用. 中国农业大学学报, 2003, 8(2): 39～41.

4 Jiawei Han, Micheline Kamber. 数据挖掘概念与技术. 范明、孟小峰, 等译. 北京：机械工业出版社, 2002. 223～259.

5 Knorr E M, Ng R Tucakov V. Distance-Based outliers: algorithms and applications. VLDB Journal Very Large Databases, 2000. 237～253.

6 Ramaswamy S, Rastogi R, Shim K. Efficient algorithms for mining outliers from large data sets. Proceedings of the ACM SIGMOD Conference, 2000. 473～438.

7 Bay S D, Schwabacher M. Mining Distance-Based Outliers in Near Linear Time with Randomization and a Simple Pruning Rule. Washington D C: SIGKDD'03, USA, 2003.

8 Breunig M M, Kriegel H P, Ng R T, et al. LOF: Identifying density-based local outliers. In: Proceedings of ACM SIGMOD International Conference on Management of Data, Dallas, Texas, USA, 2000. 93～104.

9 Aggarwal C C, Yu P S. Outlier detection for high dimensional data. In: Proceedings of the ACM SIGMOD International Conference on Management of data, 2001.

(责任编辑：黎贞崇)