基于自寻优层次聚类的孤立点分析

Optimizing Hierarchical Clustering Based Outlier Analysis

温济川 （泸州职业技术学院教学质量监控中心 四川泸州 646000）

摘 要： 检测数据集中的孤立点经常需要用户设置一些参数，当用户没有相应的经验时，孤立点检测或者困难或者不完全。本文提出一种无需参数设置，自动查找最可能的孤立点的检测方法。主要贡献包括：提出的一种聚类评价函数和自寻优层次聚类算法，该算法首先产生聚类树，然后通过评价函数，从聚类树上选择最优聚类结果；提出一个孤立类的检测算法，从聚类结果中寻找孤立类，孤立类中的数据就是检测出的孤立点。实验证明了新方法的有效性。

关键字： 孤立点 层次聚类 自寻优

1 介绍

在数据分析中，经常存在一些数据对象与数据的一般行为或模型不一致，这样的数据对象称为孤立点。孤立点分析是数据挖掘的一个重要研究方向。在孤立点的分析中，有时将孤立点看作噪声，例如提高数据挖掘的精确度时，经常需要从数据集中去除孤立点；有时将孤立点看作感兴趣的模式而分析孤立点，例如银行的信用卡欺诈检测，入侵检测等。

孤立点分析是要从数据集中发现稀有或有趣的模式。对孤立点的分析已经有广泛的研究，主要包括四个部分：基于统计分布的方法、基于距离的孤立点分析、基于密度的孤立点分析和基于偏差的孤立点分析。传统的孤立点的定义都需要先验知识，即预设参数。下面是引用参考文献[1,2,3]中关于孤立点的三个定义。

(a) 设在数据集 S 中包含 n 个数据对象$\{o_1,..,o_n\}$。$o_i \in S$ 是孤立点，仅当 S 中至少有 k 个数据对象与 o_i 的距离大于 d。[1]

(b) 设在数据集 S 中包含 n 个数据对象$\{o_1,..,o_n\}$，每个数据对象 $o_i \in S$ 到它的第 k 个最近邻数据对象的距离为 d_i。数据对象 o_i 是孤立点，仅当 $d_i \in max_n\{d_1,..,d_n\}$。$max_n\{d_1,..,d_n\}$ 表示集合$\{d_1,..,d_n\}$ 中的最大的 n 个值。[2]

(c) 设在数据集 S 中包含 n 个数据对象$\{o_1,..,o_n\}$，每个数据对象 $o_i \in S$ 到它的第 k 个最近邻数据对象的平均距离为 d_i。数据对象 o_i 是孤立点，仅当 $d_i \in max_n\{d_1,..,d_n\}$。$max_n\{d_1,..,d_n\}$ 表示集合$\{d_1,..,d_n\}$ 中的最大的 n 个值。[3]

很显然当用户没有相应的经验时，对于定义(a)中参数 k、d，定义(b)、(c)中的参数 k、n 很难选择，则难以确认孤立点。

基于上述考虑，我们提出一种不需要用户设置参数的的孤立点分析方法。该方法基于层次聚类，首先使用自底向上的层次聚类方法产生一个聚类树，然后用聚类评价指标推荐最好的聚类结果。再从聚类结果中找出具有显著差异的类作为孤立类，孤立类中的数据就是孤立点。

本文余下部分组织如下：第二节对相关工作做了介绍；第三节将介绍自寻优层次聚类算法和通过划分孤立类来检测孤立点的算法。第四节通过实验验证了我们的新方法的有效性；第五节结论。

2 相关工作

对孤立点分析最早采用统计学的方法[5]，然而统计学方法要求关于数据集的参数知识，例如数据分布。在多数情况下，数据分布可能是未知的。当特效的检验尚未开发，或者观察到的分布不可能恰当地用任何标准的分布建模时，统计学方法不能确保所有孤立点能被检测到[4]。

为了解决统计学方法的局限，引入了基于距离的孤立点检测。如果数据集合 D 中对象至少有 pct 部分与对象 o 的距离大于 dmin，则称对象 o 是以 pct 和 dmin 为参数的基于距离的孤立点[4]。目前在该领域，研究人员提供了若干高效的基于距离的孤立点挖掘算法。比较有代表性的是基于索引的算法，基于嵌套－循环的算法和基于单元划分的算法[1,2,6,7]。

基于统计学和基于距离的孤立点检测都依赖于数据集 D 的全局分布。然而数据通常并不是均匀分布的。当分析分布密度相差很大的数据时，上面的方法将遇到困难，而需要基于密度的孤立点检测方法。基于密度的孤立点检测方法基本思想来源于密度聚类方法[8,9]。

基于偏差的孤立点检测不使用统计检验或基于距离的度量来识别异常对象。相反通过检查一组对象的主要特征来识别孤立点。背离这种描述的对象认为是孤立点。基于偏差的孤立点检测技术主要有序列孤立点检测技术[10]和 OLAP 数据立方检测技术两种[11]。

3　基于层次聚类的孤立点分析

3.1　自寻优层次聚类

聚类的算法有很多，在各种聚类算法中，需要设置一些参数，如聚类的个数，半径距离、密度阈值等。在文本聚类时，当用户没有这些参数选择经验时，产生聚类就较为困难。当我们把层次聚类用于孤立点分析时，并不知道这些聚类参数，对于孤立点我们没有先验知识。于是我们提出一种适合孤立点分析的自寻优自底向上的层次聚类方法来实现孤立点分析。

在我们的工作中，首先计算各数据对象间的距离然后建立相异度矩阵，在相异度矩阵上使用层次聚类算法，建立聚类树，然后按照提出一个聚类评价标。按照该标准，在聚类树上选择最优聚类结果。同时从聚类结果中，选择差异度最大的类别作为孤立点。

在建立相异度矩阵时，需要计算两个数据对象之间的距离。当数据对象是数值向量型的数据时，可以采用公式 1 的明氏距离公式；当数据对象是从文本中抽取出的特征向量时可以采用公式 2 的相似度计算公式。

$$d(A,B) = \left(\sum_{k=1}^{n} |A(x_k) - B(x_k)|^p\right)^{1/p} \quad (1)$$

$$Sin(v_1,v_2) = \frac{\sum_{k=1}^{n} W_{1k} \times W_{2k}}{\sqrt{\sum_{k=1}^{n} W_{1k}^2 \times \sum_{k=1}^{n} W_{2k}^2}} \quad (2)$$

当建立了相异度矩阵后，就可以使用自寻优层次聚类算法建立聚类树，然后找出最优聚类结果。这时需要一个对聚类结果进行评价的标准。

虽然在一些研究文献中提出过了聚类的评价函数，但是总有一些缺点或不适合作为对层次聚类的评价。例如，文献[1]提出了 Silhouette Coefficient 聚类评价标准。依据该标准的聚类评价值位于区间[0,1]，值越大表示聚类效果越好。然而可以发现，当每个样本作为一类时，可以得到最大的值1。因此 Silhouette Coefficient 并不适合作为层次聚类的聚类评价标准。

在我们的研究中，将两个样本特征向量的距离定义为两个特征向量的相异度；并提出了一个新的聚类评价函数 DistanceSum。

$$\text{DistanceSum} = \left(\alpha_{i=1}^{n} \frac{\text{innerSum}(C_i)}{|C_i|} + \text{interSum}\right) / |C_i| \quad (3)$$

$C = \{c_1, \ldots, c_n\}$ 是一个聚类结果；$c_i \in$ 是一个类；$|c_i|$ 是第 i 个类中样本的数目；$|C|$ 是集合 C 中类的数目。设一共有 n 个类，innerSum(C_i) 是第 i 个类内部各个样本之间的距离和；interSum 是各个类之间距离的和。我们定义两个类的距离是两个类中距离最远的两个样本的距离。

对于一个好的聚类结果，在类的内部的数据对象之间应该有更近的距离，而在两个类之间的数据对象则应该有着更远的距离。称之为有好的类内紧密度和类间分离度。函数 innerSum 和 interSum 通过求样本间距离和的方式类分别反映两个类的类内紧密度和类间分离度。

评价函数 DistanceSum 的构造表明，所有样本聚为一类和所有样本各为一类两种情况的 DistanceSum 值是一样的，均为最大值。当把一个类并入另一个类，新类的类内距离和将增加，但由于减少了一个类所以类间距离和将减小。如果两个类的合并是合理的则 DistanceSum 将减小，反之则将增加。

当聚类的 DistanceSum 的值越小时表示类别的划分越合理，聚类效果越好。因此在层次聚类中，我们选择 DistanceSum 最小的层次作为最后聚类的结果。这样的聚类不需要由用户设置距离阈值、聚类个数等参数，而能得到最优聚类结果。

算法 1 是一个凝聚层次聚类算法。把每个类作为树的节点，通过将最相似的两个节点合并形成一个聚类树。

算法 1：寻优层次聚类算法

输入：相异度矩阵 matrix

输出：最优聚类结果 P

步骤：

1 将每个数据对象作为一个类构建类集合 C，形成层次聚类树的最底层

2 使用评价函数 DistanceSum 来计算集合 C 的聚类质量

3 选择 C 中相异度最小的两个类合并，形成聚类树的一个新的层。

4 重复 2、3 步骤，直到|C| =1

5 从聚类树中选择 DistanceSum 值最小的一个层作为最终聚类结果 P

3.2 孤立类分析

当聚类结果中有 n 个类，我们需要确定那些类是孤立类，孤立类中的数据就是孤立点。因此，孤立点分析的任务就转换为从聚类结果中找孤立类的问题。我们根据每个类中数据个数的差异来判断孤立类。

观察 1：一个聚类结果中个数最多的类，不是孤立类。如果存在孤立类，则数据个数最少的类是孤立类。

由观察 1，聚类结果中如果存在孤立类，则孤立类和非孤立类在类内数据个数上有显著差异。因此可以采用基于偏差的孤立点分析方法[4]，找出在数据个数上有显著差异的类别，这些类别就是孤立类，而孤立类中的数据就是孤立点。算法 2 是孤立类检测的算法描述。

算法 2：孤立类检测算法

输入：聚类结果 P

输出：孤立类集合 S

步骤：

1 提取聚类结果中每个类的数据个数存入数组 vec；建立辅助数组 aid

2 计算 vec 中元素的方差，将值存入数组 Err

3 将 vec 中最大值删除，将该值对应的类的标号插入辅助数组 aid

4 重复 2、3 步骤，直到 vec 为空

5 从数组 Err 中查找最大方差变化（Err[i] – Err[i +1]）的值，得到 index = i +1

6 查找 aid 数组中第 index 到最后一个元素对应的聚类结果 P 中类。结果放入孤立类集合 S

4 实验

为了验证算法的性能我们用 JAVA 实现了上述算法。实验在 P4 CPU，512M 内存，Windows XP 计算机上进行。

实验采用两组数据集。第一组数据集 Dataset1 采用合成数据。Dataset1 中共有 60 组二维数据，如图 1 所示。运行我们的算法后，红圈所示的是孤立类，类内的数据就是孤立点。从图 1 中可以看出，我们的方法可以正确的将孤立点识别。

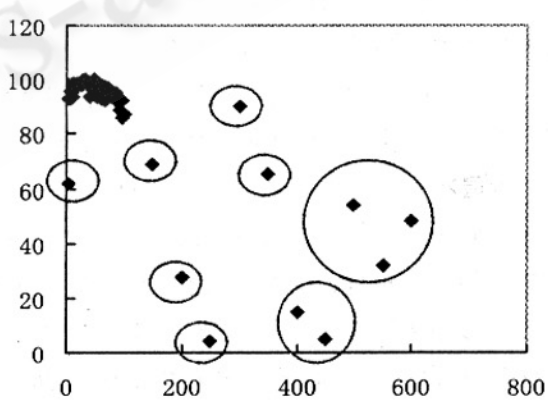

图 1　在数据集 Dataset1 上的孤立点检测

我们采用中文信息检索论坛[①]提供的中文网页训

① http://www.cwirf.org

练集 CCT2006,作为数据集 Dataset2。Dataset2 中的网页包含 8 个类别,每个类别有 150 篇网页。针对每个类,我们从其他类中选择 50 篇网页组成数据子集。这样共有 8 个数据子集,每个子集 200 篇网页。我们在每个数据子集上进行孤立点检测。表 1 是检测孤立点后的查准率和查全率。

从表 1 可以看出,我们的算法有较好的性能。如果采用传统的方法,除非用户知道孤立点的个数是 50,然后选择 Top 50 的孤立点可以达到最佳的选择,否则当用户设置选择小于 Top50 的数据时,孤立点的查全率只会比新方法低。

表 1 在数据集 Dataset1 上的孤立点检测

数据子集	Precision	Recall
1	0.97	0.90
2	0.92	0.81
3	0.99	0.94
4	0.98	0.95
5	0.95	0.89
6	0.97	0.97
7	0.98	0.99
8	0.93	0.87
AVG	0.96	0.91

5 结论

孤立点分析是数据挖掘的一个重要研究方向。本文提出一种新颖的孤立点检测算法,它不需要用户设置参数,可以自动从数据集中挖掘最可能的孤立点。该方法首先使用层次聚类算法建立聚类树,然后使用聚类评价函数从聚类树上选择最优聚类结果,再使用孤立类检测算法从聚类结果中检测孤立类,孤立类中的数据对象就是最终的孤立点。实验证明了新方法的有效性。

参考文献

1 EM Knorr, RT Ng, V Tucakov. Distance based Outlier: Algorithm and Application [J]. VLDB Journal 2000.

2 S Ramaswamy, R Rastogi, K Shim. Efficient Algorithm for Mining Outlier from Large Data Sets[C]. In the Proceedings of ACM SIGMOD, 2000.

3 F Angiulli, C Pizzuti. Fast Outlier Detection in High Dimensional Spaces[C]. In the Proceedings of Sixth European Conference on Data Mining and Knowledge Discovery, 2002.

4 Jiawei Han, Micheline Kamber. Data Mining Concepts and Techniques (Second Edition). China Machine Press, 2007.(韩家炜等. 数据挖掘 - 概念与技术. 第二版. 机械工业出版社. 2007)

5 BARNETT V, LEWIS T. Outliers in statistical data: 2nd [M]. NewYork: John Wiley & Sons, 1994.

6 EM Knorr, RT Ng. Finding intensional knowledge of distance - based outliers[J]. VLDB Journal 1999.

7 KNORR E, NG R. Algorithms for mining distance - based outliers in large datasets[C]. In the Proceedings of VLDB98.

8 A Hinneburg, DA Keim. An Efficient approach to clustering in large multimedia databases with noise [C]. In the Proceedings of KDD'98.

9 ESTER M, KRIEGEL H P, SANDER J, et al. A density based algorithm for discovering clusters in large spatial databases[C]. In the Proceedings of KDD'96.

10 ARNINGA, AGRAWAL R, RAGHAVAN P. A linear method for deviation detection in large databases [C]. In the Proceedings of KDD'96.

11 SARAWAGI S, AGRAWAL R, MEGIDDO N. Discovery driven exploration of OLAP data cubes[C]. In the Proceedings of EDBT'98.

12 Wen Jin Anthony K. H. Tung Jiawei Han. Mining Top n Local Outliers in Large Databases. In the proceedings of KDD 2001.