Charu C. Aggarwal

离群分析

第二版

Springer

Outlier Analysis

Charu C. Aggarwal

离群分析

第二版

Springer

singola@gmx.com

Charu C. Aggarwal
IBM TJ Watson研究中心约克镇高
地，纽约，美国

ISBN 978-3-319-47577-6 ISBN 978-3-319-47578-3 (eBook)
DOI 10.1007/978-3-319-47578-3

国会图书馆控制号码：2016961247

© Springer International Publishing AG 2017
这项工作受版权保护。 发布者保留所有权利，无论是全部还是部分内容，特别是翻译，转载，重复使用插图，背诵，广播，缩微胶片或其他任何物理方式的复制权，以及传输或信息存储和检索，电子适应，计算机软件，或现在已知或以后开发的类似或不同的方法。
在本出版物中使用一般描述性名称，注册名称，商标，服务标记等，即使在没有具体声明的情况下，也不意味着此类名称不受相关保护性法律法规的约束，因此免于一般使用。

出版商，作者和编辑可以安全地假设本书中的建议和信息在出版之日被认为是真实和准确的。 出版商，作者或编辑均不对此处包含的材料或任何可能出现的错误或遗漏给予明示或暗示的保证。

印在无酸纸上

Springer的印记由Springer Nature出版
The registered company is Springer International Publishing AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

singola@gmx.com

我的妻子，我的女儿萨亚尼，
和我已故的父母Prem Sarup博士和Pushplata Aggarwal夫人。

singola@gmx.com

内容

singola@gmx.com

singola@gmx.com

singola@gmx.com

singola@gmx.com

singola@gmx.com

singola@gmx.com

singola@gmx.com

singola@gmx.com

singola@gmx.com

singola@gmx.com

singola@gmx.com

singola@gmx.com

singola@gmx.com

前言

“所有优秀的东西都非常罕见。” - Baruch Spinoza

第一版

大多数关于异常值检测的最早工作都是由统计界进行的。 虽然统计方法在数学上更精确，但它们有几个缺点，例如关于数据表示的简化假设，差的算法可扩展性以及对可解释性的低度关注。 随着用于数据收集的硬件技术的不断进步以及用于数据组织的软件技术（数据库）的进步，计算机科学家越来越多地参与该领域的最新进展。 计算机科学家根据他们在管理大量数据方面的实际经验来处理这个领域，并且假设数量少得多 - 数据可以是任何类型的，结构化的或非结构化的，并且可能非常大。 此外，计算机效率和数据直观分析等问题通常被计算机科学家认为比数学精度更重要，尽管后者也很重要。 这是数据挖掘领域的专业人士的方法，数据挖掘是大约20年前建立的计算机科学领域。 这导致了关于这一主题的多个学术团体的形成，这些学术团体仍然分离，部分原因在于技术风格的差异以及对不同问题的重要性和对该主题的方法的看法。 此时，与统计学家相比，数据挖掘专业人员（具有计算机科学背景）更积极地参与该领域。 这似乎是研究领域的一个重大变化。 本书从综合的角度介绍异常值检测，但重点是计算机科学专业人员。 特别强调将不同社区的方法相互联系起来。

在这个时间点写这本书的关键优势在于，计算机科学专业人员在过去二十年中所做的大量工作基本上没有受到关于这一主题的正式书籍的影响。 与异常值分析相关的经典书籍如下：

- P. Rousseeuw和A. Leroy。 稳健的回归和异常值检测，Wiley，2003。
- V. Barnett和T. Lewis。 统计数据中的异常值，Wiley，1994。
- D.霍金斯。 鉴于异常值，查普曼和霍尔，1980年。

singola@gmx.com

我们注意到这些书已经过时了，其中最新的是十年之久。 此外，这本（最新的）书真正关注的是回归和异常分析之间的关系，而不是后者。 异常值分析是一个更广泛的领域，其中回归分析只是一小部分。 其他书甚至更老，年龄在15到25岁之间。 它们专门针对统计社区。 这并不奇怪，因为数据挖掘（KDD）的第一个主流计算机科学会议是在1995年组织的。 数据挖掘社区的大部分工作都是在编写这些书之后进行的。 因此，这些书中没有涉及更广泛的数据挖掘社区感兴趣的许多关键主题。

本书的章节经过精心组织，以自然顺序广泛覆盖该区域。 重点放在简化内容上，以便学生和从业者也可以从书中获益。 虽然我们最初并不打算创建一个关于这一主题的教科书，但它在写作过程中发展成为一种既可以用作教学辅助工具也可以用作教材。 此外，它还可以用作参考书，因为每章都包含大量的书目记录。 因此，本书有双重目的，从多个角度全面阐述异常检测的主题。

第二版的附加说明

本书的第二版是对第一版的重大改进。 特别是，大多数章节都使用新材料和最新技术进行了升级。 在一些地方增加了更多解释，并且还增加了更新的技术。 关于异常集合的整章已经增加。 本书增加了许多新的主题，如特征选择，一类支持向量机，一类神经网络，矩阵分解，谱方法，小波变换和监督学习。 每章都更新了该主题的最新算法。

最后但并非最不重要的是，第一版由出版商作为专着进行分类，而第二版则正式分类为教科书。 书写风格得到了增强，学生可以轻松理解。 已经更详细地描述了许多算法，正如人们可能从教科书中期望的那样。 它还附有课堂教学的解决方案手册。

singola@gmx.com

Acknowledgments

第一版

我要感谢我的妻子和女儿在撰写本书时给予的爱和支持。 写一本书需要花费大量时间从家庭成员手中夺走。 这本书是他们在此期间耐心等待的结果。 我还欠我已故的父母，感谢他们向我灌输了对教育的热爱，这在我的书写作品中发挥了重要的鼓舞作用。

我还要感谢我的经理Nagui Halim为撰写本书提供了必要的巨大支持。 他的专业支持对我过去和现在的许多书籍都起到了重要作用。

多年来，我从众多合作者的见解中受益匪浅。 这些长期合作者按字母顺序列出的完整清单是Tarek F. Abdelzaher，Jiawei Han，Thomas S. Huang，Latifur Khan，Mohammad M. Masud，Spiros Papadimitriou，Guojun Qi和Philip S. Yu。 我要感谢他们多年来的合作和见解。

我还要特别感谢我的顾问詹姆斯·B·奥林（James B. Orlin）作为研究员在我早年的指导。 虽然我不再在同一领域工作，但我从他那里学到的遗产是我研究方法的重要组成部分。 特别是，他教会了我在研究过程中直觉和思想简单的重要性。 这些是研究中比公认的更重要的方面。 本书以简单直观的方式编写，旨在提高该领域对研究人员和从业人员的可访问性。

最后，我要感谢Lata Aggarwal帮助我完成了本书中使用PowerPoint图形创建的一些图形。

致第二版的致谢

在第二版的撰写过程中，我收到了各位同事的重要反馈。 特别是，我要感谢Leman Akoglu，Linh-Jen Lin，Saket Sathe，Jiliang Tang和Suhang Wang。 Leman和Saket提供了本书几个章节和章节的详细反馈。

singola@gmx.com

Author Biography

Charu C. Aggarwal 是IBM的杰出研究成员（DRSM）
纽约Yorktown Heights的TJ Watson研究中心。 他于1993年在Kan-pur的印度理工学院获得计算机科学研究生学位，并获得博士学位。 他于1996年从麻省理工学院毕业。 他在数据挖掘领域开展了广泛的工作。 他有一个出版社该技术获得了IBM杰出技术成就奖（2009年，2015年），分别用于数据流和高维数据。 在审稿会议和期刊上发表论文300余篇，撰写专利80多项。 他是15本书的作者或编辑，包括一本关于数据挖掘的教科书和一本关于异常值分析的综合书。 由于他的专利具有商业价值，他曾三次被指定为IBM的发明大师。 他因其在数据流中检测生物恐怖主义威胁的工作而获得IBM企业奖（2003年），他因其对隐私的科学贡献而获得IBM杰出创新奖（2008年）。
他因其基于缩合的隐私保护数据挖掘工作获得了2014年EDBT时间测试奖。 他还是IEEE ICDM研究贡献奖（2015年）的获得者，该奖项是数据挖掘领域中潜在研究贡献的两个最高奖项之一。

他曾担任IEEE大数据会议（2014年）的联合主席以及ACM CIKM会议（2015年），IEEE ICDM会议（2015年）和ACM KDD会议（2016年）的联合主席。 。 2004年至2008年，他担任IEEE知识与数据工程交易副主编。 他是来自Data的知识发现ACM交易的副主编，IEEE大数据交易的副主编，行动编辑数据挖掘和知识发现期刊，ACM SIGKDD探索的主编，以及知识和信息系统期刊的副主编。 他是Springer出版的社交网络讲义的顾问委员会成员。 他曾担任SIAM数据挖掘活动小组的副主席，并且是SIAM行业委员会的成员。 他是SIAM，ACM和IEEE的成员，负责“对知识发现和数据挖掘算法的贡献”。

singola@gmx.com

第1章

异常分析简介

> “永远不要把你不同的评论视为谴责，这可能是一种恭维。 这可能意味着你拥有独特的品质，就像最稀有的钻石一样.....是独一无二的。“ - Eugene Nathaniel Butler

1.1 介绍

异常值是一个与剩余数据明显不同的数据点。 霍金斯定义[249] 如下异常值：

> “异常值是一种观察结果，与其他观察结果偏离太多，以至于引起人们怀疑它是由不同的机制产生的。 ”

异常值在数据挖掘和统计文献中也被称为异常，不一致，偏差或异常。 在大多数应用程序中，数据由一个或多个生成过程创建，这些过程可以反映系统中的活动或收集的有关实体的观察结果。 当生成过程表现异常时，会导致异常值的产生。 因此，异常值通常包含有关影响数据生成过程的系统和实体的异常特征的有用信息。 对这些不寻常特征的识别提供了有用的应用特定见解。 一些例子如下：

- **入侵检测系统：**在许多计算机系统中，收集有关操作系统调用，网络流量或其他用户操作的不同类型的数据。 由于恶意活动，此数据可能会显示异常行为。 对此类活动的识别称为入侵检测。
- **信用卡欺诈：**信用卡欺诈越来越普遍，因为信用卡号等敏感信息可以更容易受到损害。 在许多情况下，未经授权使用信用卡可能会显示不同的模式，例如从特定地点购买特权或非常大的交易。 这些模式可用于检测信用卡交易数据中的异常值。

© Springer International Publishing AG 2017

C.C. Aggarwal, *Outlier Analysis*, DOI 10.1007/978-3-319-47578-3_1

singola@gmx.com

- **有趣的传感器事件：**传感器通常用于跟踪许多实际应用中的各种环境和位置参数。 基础模式的突然变化可能代表感兴趣的事件。 事件检测是传感器网络领域的主要激励应用之一。 正如本书后面所讨论的，事件检测是异常检测的重要时间版本。

- **医疗诊断：**在许多医疗应用中，数据来自各种设备，如磁共振成像（MRI）扫描，正电子发射断层扫描（PET）扫描或心电图（ECG）时间序列。 这些数据中不寻常的模式通常会反映疾病状况。

- **执法：**异常检测在执法中发现了许多应用，特别是在只能通过实体的多种行为随时间发现异常模式的情况下。 确定金融交易，交易活动或保险索赔中的欺诈通常需要识别由犯罪实体的行为产生的数据中的异常模式。

- **地球科学：**通过卫星或遥感等各种机制收集大量关于天气模式，气候变化或土地覆盖模式的时空数据。 此类数据中的异常情况可提供有关人类活动或环境趋势的重要见解，这可能是潜在原因。

在所有这些应用中，数据具有“正常”模型，并且异常被识别为与该正常模型的偏差。 正常数据点有时也称为内点。 在诸如入侵或欺诈检测的一些应用中，异常值对应于多个数据点的序列而不是单个数据点。 例如，欺诈事件通常可以反映特定序列中的个体的行为。 序列的特定城市与识别异常事件有关。 这种异常也被称为集体异常，因为它们只能从一组或一系列数据点集体推断出来。 这种集体异常往往是产生异常活动模式的异常事件的结果。 本书将解决这些不同类型的异常现象。

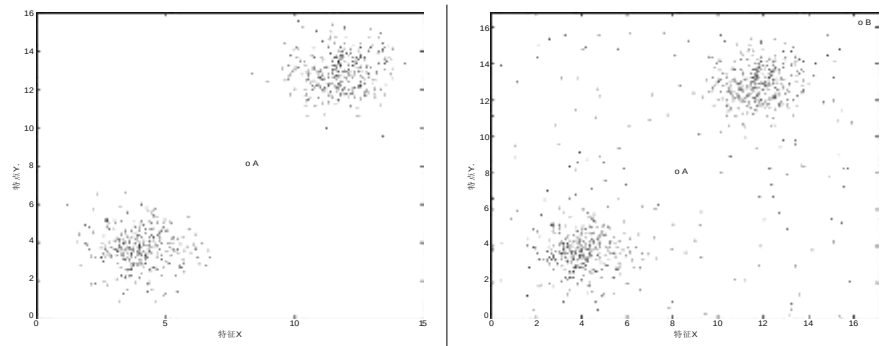
异常值检测算法的输出可以是以下两种类型之一：

- **异常值得分：**大多数异常值检测算法输出一个量化每个数据点“异常值”水平的分数。 该分数还可用于按其异常趋势的顺序对数据点进行排名。 这是一种非常通用的输出形式，它保留了特定算法提供的所有信息，但它没有提供应被视为异常值的少量数据点的简明摘要。

- **二进制标签：**第二种类型的输出是二进制标签，指示数据点是否是异常值。 虽然某些算法可能直接返回二进制标签，但异常值也可以转换为二进制标签。 这通常通过对异常值得分施加阈值来实现，并且基于得分的统计分布来选择阈值。 二进制标记包含的信息少于评分机制，但它是实际应用中决策制定通常需要的最终结果。

这通常是一种主观判断，关于什么构成了一个被认为是异常值的“足够”的偏差。 在实际应用中，数据可以嵌入到

singola@gmx.com



(a) 无噪音 （b）噪音图1.

1：噪音与异常之间的差异

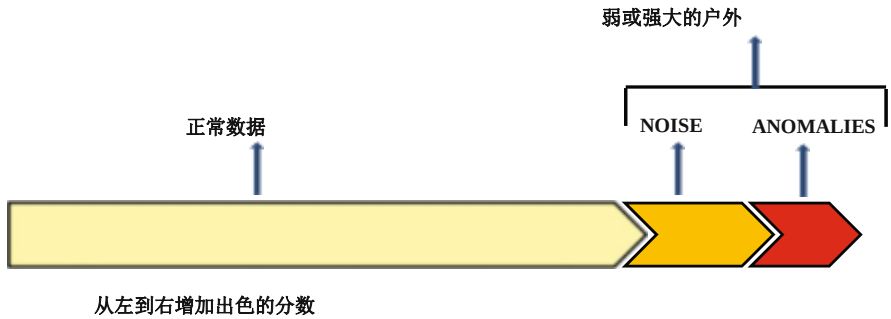


图1.2：从正常数据到异常值的频谱

显着的噪音，这种噪音可能不会对分析师有任何兴趣。 这通常是令人感兴趣的显着有趣的偏差。 为了说明这一点，请考虑图1.1（a）和（b）中所示的示例。 很明显，数据中的主要模式（或群集）在两种情况下都是相同的，尽管这些主要群集之外存在显着差异。 在图1.1（a）的情况下，单个数据点（标记为“A”）似乎与其余数据非常不同，因此非常明显是异常。 图1.1（b）中的情况更为主观。 而相应的数据点'A'如图1.1所示（b）也处于数据的稀疏区域，很难确切地说它表示与剩余数据集的真实偏差。 该数据点很可能代表数据中随机分布的噪声。 这是因为点“A”似乎是由其他随机分布点表示的模式。 因此，在本书中，术语“异常值”是指可以被视为异常或噪声的数据点，而“异常”是指分析人员感兴趣的特殊异常值。

在无监督的场景中，前面的有趣异常示例不可用，噪声代表正常数据和真实异常之间的语义边界 - 噪声通常被建模为弱点形式的异常值，并不总是满足数据所需的强标准。 要点被认为是有趣的或异常的。 例如，群集边界处的数据点通常可以被视为噪声。 Typ-

singola@gmx.com

实际上，大多数离群值检测算法使用数据点的离群值的一些量化测量，例如基础区域的稀疏性，基于最近邻居的距离，或者对基础数据分布的结果。 每个数据点都位于从正常数据到噪声的连续光谱上，最后是异常，如图1.2 所示。 该频谱的不同区域的分离通常不是精确定义的，并且是根据应用特定标准在临时基础上选择的。 此外，噪声和异常之间的分离并不是纯粹的，并且由嘈杂的生成过程产生的许多数据点可能具有足够的偏差，可以基于异常值得分被解释为异常。 因此，异常通常具有比噪声高得多的异常值，但这不是两者之间的区别因素。 相反，分析师的兴趣是调节噪声和异常之间的区别。

一些作者使用弱异常值和强异常值来区分噪声和异常[4,318]。 数据中的噪声检测具有其自身的许多应用。 例如，噪声的消除创建了更清晰的数据集，可用于其他数据挖掘算法。 尽管噪声本身可能并不令人感兴趣，但其去除和识别仍然是采矿目的的一个重要问题。 因此，噪声和异常检测问题在本书中都很重要。 在本书中，将确定与异常检测或噪声消除特定相关的方法。 然而，大部分离群值检测算法可以用于任何一个问题，因为它们之间的差异实际上是语义之一。

由于噪声和异常之间的语义区别是基于分析师的兴趣，因此发现此类异常并将其与噪声区分开的最佳方法是使用先前已知的异常示例的反馈。 在许多应用中经常出现这种情况，例如信用卡欺诈检测，其中可能存在先前有趣的异常示例。 这些可用于学习区分正常模式和异常数据的模型。 在许多特定应用场景中，监督异常值检测技术通常更有效，因为前面示例的特性可用于将搜索过程锐化为更相关的异常值。 这很重要，因为异常值可以在给定的数据集中以多种方式定义，其中大多数可能不是很有趣。 例如，在图中 在图1.1（a）和（b）中，先前的例子可能表明只有具有异常高的两个属性值的记录才应被视为异常。 在这种情况下，两个图中的点'A'应视为噪声，图1.1中的点'B'（b）应该被视为异常！ 这里要理解的关键点是，异常需要以一种有趣的方式变得不同寻常，监督过程会重新定义一个人可能会感兴趣的东西。 通常，无监督方法可用于噪声消除或异常检测，并且监督方法被设计用于应用特定异常检测。 非探测性方法通常用于探索性设置，其中将发现的异常值提供给分析人员，以进一步检查其应用特定的重要性。

在实际情况下，可以进行多级监督。 在完全监督的情景中，可以清楚地区分正常和异常数据的示例。 在某些情况下，可以使用异常值的示例，但“正常”数据的示例也可能包含某些（未知）比例的异常值。 这被称为具有正数和未标记数据的分类。 在其他半监督场景中，仅可以获得正常数据的示例或仅异常数据的示例。 因此，问题的变化的数量相当大，每个变化都需要相关但专用的技术集。

singola@gmx.com

最后，数据表示可能因应用程序而异。 例如，数据可以是纯多维的，点之间没有关系，或者数据可以按时间顺序顺序，或者可以以具有数据点之间的任意关系的网络的形式定义。 此外，数据中的属性可以是数字，分类或混合。 显然，异常值检测过程需要对基础数据中属性和关系的性质敏感。 实际上，关系本身通常可以以通常不一起出现的实体之间的连接的形式提供异常值检测标准。 这种异常值被称为上下文异常值。 一个典型的例子是社交网络分析中关联异常值的概念[17]。 在这种情况下，图中通常未连接在一起的实体（节点）可能显示彼此的异常连接。 因此，数据类型对异常检测过程的影响是显着的，并将在本书中仔细解决。

本章安排如下。 在1.2 节中，讨论了异常值分析中数据建模的重要性。 在1.3 节中，介绍了异常值检测的基本异常值模型。 异常集合在1.4 节中介绍。 第1.5 节讨论了用于分析的基本数据类型。 1.6 节介绍了用于数据分析的异常值的监督建模的概念。 评估异常值检测算法的方法将在1.7 节中讨论。 结论见第1.8节。

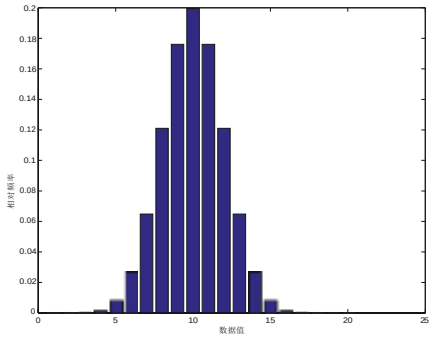
1.2 数据模型就是一切

实际上，所有异常值检测算法都会创建数据中正常模式的模型，然后根据与这些模式的偏差计算给定数据点的异常值。 例如，该数据模型可以是生成模型，例如高斯混合模型，基于回归的模型或基于接近度的模型。 所有这些模型都对数据的“正常”行为做出了不同的假设。 然后通过评估数据点和模型之间的质量来计算数据点的离群值。 在许多情况下，该模型可以在算法上定义。 例如，基于最近邻居的离群值检测算法根据其k的分布来模拟数据点的离群趋势- 最近邻距离。 因此，在这种情况下，假设异常值位于距大多数数据很远的位置。

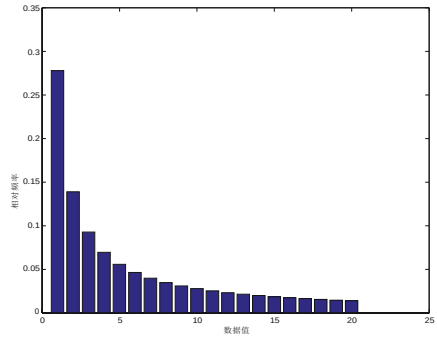
显然，数据模型的选择至关重要。 错误选择数据模型可能会导致结果不佳。 例如，如果数据不能满足模型的生成假设，或者如果没有足够的数据点来学习模型的参数，那么诸如高斯混合模型的完全生成模型可能效果不佳。 同样，如果底层数据是任意聚类的，那么基于线性回归的模型可能效果不佳。 在这种情况下，由于对模型的错误假设的不良影响，数据点可能被错误地报告为异常值。 不幸的是，异常检测在很大程度上是一个无监督的问题，其中不存在异常值的例子
学习[1] 特定数据集的最佳模型（以自动方式）。 异常值检测的这一方面往往使其比许多其他监督数据挖掘更具挑战性
像分类的问题，其中有标记的例子可用。 因此，在实践中，模型的选择通常由分析师对与应用程序相关的偏差类型的理解决定。 例如，在测量行为属性（例如位置特定温度）的空间应用中，假设空间局部中温度属性的异常偏差是一个是合理的。

[1]在分类等监督问题中，此过程称为模型选择。

singola@gmx.com



（a）正态分布



（b）Zipf分布图1.3：对Normal和Zipf分布应用Z值检验

异常指标。 另一方面，对于高维数据的情况，由于数据稀疏性，甚至数据局部性的定义也可能是不明确的。 因此，只有在仔细评估该域的相关建模属性之后，才能构建特定数据域的有效模型。

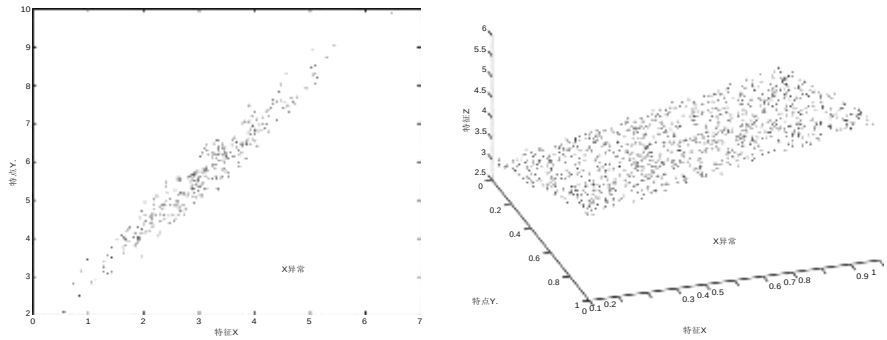
为了理解模型的影响，检查一个简单模型的使用是有益的，这个模型被称为异常值分析的Z值检验。 考虑一组1维定量数据观察，用$X_1 \ldots X_N$ 表示，平均值μ 和标准偏差σ。 数据点X_i 的Z值由Z_i 表示，其定义如下：

$$Z_i = \frac{|X_i - \mu|}{\sigma} \tag{1.1}$$

所述z-值测试计算的标准偏差的数量，通过该数据点是从平均遥远。 这为该点的离群值得分提供了良好的代理。 隐含的假设是数据是从正态分布建模的，因此Z值是从具有零均值和单位方差的标准正态分布中抽取的随机变量。 在可以准确估计分布的均值和标准差的情况下，良好的“经验法则”是使用Z_i 3作为异常的代理。 但是，在极少数样本可用的情况下，无法稳健估计基础分布的均值和标准差。 在这种情况下，结果来自Z.- 值测试需要使用（相关）学生的t分布而不是正态分布来更好地解释。 这个问题将在第2章中讨论。

从业者在建模过程中经常会忘记Z值检验隐含地假设基础数据的正态分布。 当这种近似差时，结果难以解释。 例如，考虑在图1.3中以1到20之间的值绘制的两个数据频率直方图。 在第一种情况中，直方图是从与（正态分布采样μ，σ）=（10，2），并且在第二种情况中，它是从一个齐普夫分布1取样/ I。 显而易见的是，大部分数据位于范围[10 2 3，10 + 2 3]为正态分布，以及位于该范围之外的所有数据点可以被认为是真正的异常。 就这样在这种情况下，Z值测试非常有效。 在Zipf分布的第二种情况下，异常并不十分清楚，尽管具有很高值（例如20）的数据点可能被认为是异常。 在这种情况下，数据的平均值和标准偏差分别为5.24和5.56。 因此，Z值测试不会将任何数据点声明为异常（对于a

singola@gmx.com



(a) 2-d数据 （b）3-d数据

图1.4：线性相关数据

3）的门槛，虽然确实接近了。 无论如何，至少从概率可解释性的角度来看，Zipf分布中Z值的重要性并不是很有意义。 这表明如果在建模阶段出现错误，则可能导致对数据的错误理解。 这些测试通常用作启发式算法，即使对于远离正态分布的数据集也能提供异常值分数的粗略概念，并且仔细解释这些分数很重要。

Z值验证即使作为启发式算法也不会起作用的一个例子是，它只是因为它的相对位置而不是它的极端位置而应用于一个异常值的数据点。 例如，如果将Z值测试应用于图1.1（a）中的单个维度，则测试将失败，因为点'A'将被视为最中心位置和正常数据点。 另一方面，测试仍然可以合理地应用于对应于每个点的k个最近邻距离的一组提取的1维值。 因此，模型的有效性取决于所用测试的选择以及如何应用。

模型的最佳选择通常是数据特定的。 这需要在选择模型之前很好地理解数据本身。 例如，基于回归的模型最适合于发现图1.4 的数据分布中的异常值，其中大多数数据沿线性相关平面分布。 另一方面，聚类模型更适合于图1.1中所示的情况。 给定数据集的模型选择不当可能会导致结果不佳。 因此，发现异常值的核心原则是基于对给定数据集中正常模式的结构的假设。 显然，“正常”模型的选择在很大程度上取决于分析师对该特定领域中自然数据模式的理解。 这意味着分析师对数据表示具有语义理解通常很有用，尽管在实际设置中这通常是不可能的。

有许多与模型选择相关的交易；具有太多参数的高度复杂模型很可能会超过数据，并且还会找到一种方法来处理异常值。 一个简单的模型，通过对数据的直观理解（也可能是对分析师正在寻找的内容的理解）构建，可能会带来更好的结果。 另一方面，过于简单的模型，可以很好地处理数据，可能会将正常模式声明为异常值。 选择数据模型的初始阶段可能是离群值分析中最关键的一个。 关于数据模型影响的主题将在整本书中重复，并附有具体的例子。

singola@gmx.com

1.2.1 与监督模型的连接

可以将异常值检测问题视为分类问题的变体，其中类标签（“正常”或“异常”）未被观察到。 因此，由于正常示例远远超过异常示例的事实，人们可以“假装”整个数据集包含正常类并且创建正常数据的（可能是有噪声的）模型。 与正常模型的偏差被视为异常值。 分类和离群检测之间的这种联系很重要，因为分类中的大部分理论和方法都归纳为异常值检测[32]]。 标记的未观察到的性质（或异常值得分）是异常检测方法被称为无监督的原因，而分类方法被称为监督。 在观察到异常标签的情况下，问题简化为数据分类的不平衡版本，并在第7章中详细讨论。

用于无监督异常值检测的正常数据模型可以被认为是分类中多类设置的一类模拟。 但是，从建模的角度来看，单类设置有时会更加微妙，因为在两个类的示例之间区分比预测特定实例是否匹配单个（普通）类的示例要容易得多。 当至少有两个类可用时，可以更容易地学习两个类之间的区别特征，以便提高模型的准确性。

在许多形式的预测学习中，例如分类和推荐，在基于实例的学习方法和显式泛化方法之间存在自然的二分法。 由于异常值检测方法需要设计正常数据模型以进行预测，因此这种二分法也适用于无监督域。 在基于实例的方法中，培训模型不是预先构建的。 相反，对于给定的测试实例，人们计算训练数据的最相关（即最接近）实例，并使用这些相关实例对测试实例进行预测。 基于实例的方法在分类领域被称为懒惰学习者[33]，在推荐系统领域也被称为基于记忆的方法[34]。

异常值分析中基于实例的学习方法的一个简单示例是使用数据点的1最近邻距离作为其异常值得分。 请注意，此方法不需要预先构建训练模型，因为在指定要预测的实例的身份（评分）之后，完成了确定最近邻居的所有工作。 1最近邻居离群值检测器可以被认为是监督域中1个最近邻居分类器的无监视模拟。 基于实例的模型在异常分析领域非常流行，因为它们简单，有效且直观。 事实上，许多最流行和最成功的异常检测方法，如k-最近邻探测器[58，456]和局部异常因子（LOF）[96]（参见第4章），是基于实例的方法。

基于实例的方法在异常值分析社区中的普及程度非常高，以至于其他监督方法的大量一类类似物经常被忽视。 原则上，几乎任何分类方法都可以重新设计以创建一类模拟。 这些方法中的大多数是显式泛化方法，其中包含概括模型
需要预先创建。 显式泛化方法对数据集D使用两步过程：

1. 使用原始数据集创建普通数据的单类模型。 例如，人们可能会学习描述图1.4（b）中的正常数据的线性超平面。 该超平面表示整个数据集的概括模型，因此表示数据集的显式概括。

singola@gmx.com

表1.1：异常值分析中的分类方法及其无监督类似物

监督模型	无监督模拟	类型
k-最近的邻居	k-NN distance, LOF, LOCI (Chapter 4)	Instance-based
线性回归	主成分分析 (Chapter 3)	明确的泛化
朴素贝叶斯	Expectation-maximization (Chapter 2)	明确的泛化
Rocchio	马哈拉诺比斯法（第3章） 聚类（第4章）	明确的泛化
决策树 随机森林	隔离树 隔离森林（章节5 和6）	明确的概括
Rule-based	FP-Outlier (Chapter 8)	明确的泛化
Support-vector 机	One-class support-vector 机器（第3章）	明确的概括
神经网络	复制器神经网络 (Chapter 3)	明确的概括
矩阵分解 （数据预测不完整）	主成分分析 矩阵分解（第3章）	明确的概括

2. 根据与此正常数据模型的偏差对每个点进行评分。 例如，如果我们在第一步中使用图1.4（b）的数据集学习线性超平面，那么我们可以将该超平面的欧几里德距离报告为异常值。

显式泛化方法的一个问题是相同的数据集用于训练和评分。 这是因为在评分过程中很难排除特定的测试点（如基于实例的方法）。 此外，与其中地面实况（标记）的存在或不存在自然地将数据划分为训练和测试部分的分类不同，在无监督问题中没有可用的标记。 因此，人们通常希望在无人监督的问题中使用整个数据集进行训练和测试，这会导致过度配置。然而，在实际环境中，过度配置的各个点的影响通常很小，因为显式泛化方法倾向于创建更大数据集的简明摘要（即，通用表示）。 由于相同的数据用于训练和测试，因此可以将异常值评分视为在假设“假装”所有训练数据点属于正常类别的情况下进行的训练数据错误。通常，减少过度配置的有效方法是以随机方式将数据重复划分为训练和测试集，并对来自各种模型的测试点的异常值进行平均。 这些方法将在后面的章节中讨论。

实际上，所有分类模型都可以通过使用适当的单类模拟推广到异常值检测。 此类模型的示例包括线性回归模型，主成分分析，概率期望最大化模型，聚类方法，一类支持向量机，矩阵分解模型和一类

singola@gmx.com

神经网络。 对于熟悉分类问题的读者，我们在表1.1中列出了各种分类模型及其相应的一类模拟，用于异常检测。 该表并不全面，旨在通过代表性示例提供有关监督和无监督设置之间联系的直觉。 有监督和无监督学习之间的联系非常深刻; 在第7章第7.7 节中，我们指出异常值检测和回归建模之间的另一个有用的联系。 这种特殊的连接具有以下优点：它能够使用数百个现成的回归模型进行无人监督的离群值检测，并且实现起来非常简单。

1.3 基本异常值检测模型

本节将概述文献中最重要的模型，并提供一些可能适用的设置。 后面的章节将详细讨论这些方法。 影响异常值模型选择的几个因素包括数据类型，数据大小，相关异常值示例的可用性以及模型中可解释性的需求。 这些标准中的最后一个值得进一步解释。

从分析人员的角度来看，离群值检测模型的可解释性非常重要。 通常需要确定为什么特定数据点应被视为异常值，因为它为分析人员提供了有关特定应用场景中所需诊断的进一步提示。 这个过程也被称为揭示关于异常值[318]或异常检测和描述[44]的内涵知识的过程[44]]。 不同的模型具有不同的可解释性。 通常，使用原始属性并对数据使用较少变换的模型（例如，主成分分析）具有更高的可解释性。 交易 - ff是数据转换通常以可解释性为代价增强异常值和正常数据点之间的对比度。 因此，在为异常值分析选择特定模型时，牢记这些因素至关重要。

1.3.1 异常检测中的特征选择

众所周知，由于离群值检测问题的无监督性质，在离群值检测中执行特征选择是很困难的。 与分类可以用作引导柱的分类不同，在无监督异常检测中学习特征如何与（未观察到的）基本事实相关是很困难的。 然而，一种常见的测量方法
一组单变量点x1 ... xN 的非均匀性是峰度测量。 第一步是计算这组值的平均μ 和标准偏差σ 并进行标准化
数据为零均值和单位方差如下：

$$z_i = \frac{x_i - \mu}{\sigma} \tag{1.2}$$

请注意，zi 的平方的平均值始终为1，因为zi 是如何定义的。 峰度测量计算zi的四次幂的平均值：

$$K(z_1 \ldots z_N) = \frac{\sum_{i=1}^{N} z_i^4}{N} \tag{1.3}$$

非均匀的特征分布显示出高水平的峰度。 例如，当数据包含一些极值时，峰度测量值将增加，因为

singola@gmx.com

使用第四种力量。 在子空间异常检测方法（见第5章）的背景下经常使用峰度测量[367]，其中在数据的低维投影中探索异常值。

Kurtosis测量的一个问题是，当它分别分析特征时，它不能很好地利用各种属性之间的相互作用。 也可以在较低维度的距离分布上使用峰度测量。 例如，可以在将数据投影到较低维度子空间S之后，计算所有数据点的N个Mahalanobis距离的集合到数据的质心的峰度测量 。 这种计算提供了子空间S的多维峰度，同时考虑了S的各个维度之间的相互作用。 马哈拉诺比斯距离在第2章中介绍。 可以将该计算与迭代地将特征迭代地添加到特征的候选子集S 的贪婪方法组合，以便构建具有最高多维峰度的维度的辨别子集。 第二种特征选择方法[429]是利用离群值检测问题的连接来监督学习。 基本思想是与所有其他特征无关的特征应该被认为是无关紧要的，因为异常值通常与违反正常数据依赖性模型相对应。 不相关的功能不能用于建模数据依赖性。 因此，如果使用回归模型来预测其他要素中的某个要素，并且平均误差过大，则应修剪此类要素。 将所有特征标准化为单位方差，并计算从其他特征预测第k个特征的均方根误差RMSEk 。 注意，如果是RMSEk如果大于1，则预测误差大于特征方差，因此应修剪第k个特征。 人们也可以使用这种方法来加权这些特征。 具体而言，第k个特征的权重由下式给出

最大{0，1 - RMSEk}。 有关此模型的详细信息，请参见第7章第7.7 节。

1.3.2 极值分析

离群检测的最基本形式是对一维数据的极值分析。 这些是非常特殊类型的异常值，其中假设过大或过小的值都是异常值。 在许多特定应用场景中，这种特殊类型的异常值也很重要。

关键是确定基础分布的统计尾部。 如前面图1.3所示，根据底层数据分布，尾部的性质可能会有很大差异。 正态分布是最容易分析的，因为大多数统计检验（例如Z值检验）可以直接用重要概率来解释。 然而，即使对于任意分布，这样的测试也提供了对数据点的异常分数的良好启发式概念，即使它们不能在统计上被解释。 在统计学文献中已经广泛研究了确定分布尾部的问题。 这些方法的细节将在第2章中讨论。

极值统计[437]不同于传统的异常值定义。 霍金斯提供的传统异常值定义通过它们的生成概率来定义这些对象，而不是它们的价值中的极端。 例如，在数据组1，2，2，50，98，98，1维值99，值1和99能，非常温和，被认为是极端值。 另一方面，值50是数据集的平均值，并且最明显不是极值。 但是，值50从其他大多数的数据值，其被分组为小范围，例如1的分离的，2，2和98，98，99。 因此，大多数基于概率和密度的模型都会将值50分类为数据中最强的异常值，这个结果也与Hawkins的一致性一致。

{ } { }

singola@gmx.com

异常值的定义。 极值分析和异常值分析之间的混淆很常见，尤其是在多变量数据的背景下。这种情况经常发生，因为许多极值模型也使用概率模型来量化数据点是极值的概率。

虽然极值分析自然是针对单变量（一维）数据设计的，但也可以通过确定数据的多维外围的点来将其推广到多变量数据。 重要的是要理解，即使在多变量情况下，这种异常值检测方法也适合于确定特定类型的异常值。 例如，图1.1（a）和（b）中的点“A”不会被这些方法视为极值，因为它不位于数据的外边界，即使它非常清楚图1.1（a）中的异常值 。另一方面，图1.1中的点'B'（b）可以被认为是一个极值，因为它位于多维数据集的郊区。

极值建模在大多数离群值检测算法中扮演着重要角色，这是最后一步。 这是因为大多数异常值建模算法以数字分数的形式量化数据点与正常模式的偏差。 极值分析通常需要作为这些建模偏差的最后一步，因为它们现在表示为单变量值，其中极值对应于异常值。 在许多多标准离群值检测算法中，可以获得离群值得分的向量（例如气象应用中的温度和压力的极值）。 在这种情况下，多变量极值方法可以帮助将这些异常值分数统一为单个值，并生成二进制标签输出。 因此，即使原始数据可能不是极值分析直接有用的形式，它仍然是异常值检测过程的一个组成部分。 此外，许多实际应用程序跟踪统计汇总，其中极值分析提供有关异常值的有用见解。

极值分析也可以扩展到多变量数据与使用的移动距离或深度为基础的方法[295，343，468]。 但是，这些方法仅适用于某些类型的特殊情况，其中已知异常值存在于数据边界。对多标准异常值分数的许多形式的后处理可以使用这样的方法。 另一方面，这种方法对于通用异常值分析不是非常有用，因为它们不能发现数据集的稀疏内部区域中的异常值。

1.3.3 概率论和统计模型

在概率和统计模型中，数据以闭合形式的概率分布的形式建模，并且学习该模型的参数。因此，这里的关键假设是关于执行建模的数据分布的具体选择。 例如，高斯混合模型假设数据是生成过程的输出，其中每个点属于k中的一个高斯聚类。 通过对观测数据使用期望最大化（EM）算法来学习这些高斯分布的参数，使得生成数据的过程的概率（或可能性）尽可能大。 该方法的关键输出是数据点到不同聚类的隶属概率，以及基于密度的模型分布。这提供了对异常值进行建模的自然方法，因为具有非常低的fi值的数据点可以被视为异常值。 在实践中，这些值的对数被用作离群值得分，因为利用对数值将异常值更好地表现为极值。 如前所述，极端值

singola@gmx.com

测试可以应用于这些值以识别异常值。

概率模型的一个主要优点是它们可以很容易地应用于几乎任何数据类型（或混合数据类型），只要每种混合物组分都有适当的生成模型即可。 例如，如果数据是分类的，那么可以使用离散的伯努利分布来模拟混合物的每个组分。 对于不同类型属性的混合，可以使用属性特定生成组件的乘积。 由于这些模型与概率一起工作，因此数据规范化的问题已经由生成假设来解释。 因此，概率模型提供了一个通用的基于EM的框架，它相对容易应用于任何特定的数据类型。 许多其他模型不一定如此。

概率模型的一个缺点是它们试图将数据转换为特定类型的分布，这有时可能不合适。此外，随着模型参数的数量增加，过度配置变得更加普遍。 在这种情况下，异常值可以包含正常数据的基础模型。 许多参数模型在内涵知识方面也难以解释，特别是当模型的参数无法根据基础属性直观地呈现给分析师时。 这可以打败异常检测的重要目的之一，即提供对异常数据生成过程的诊断理解。 第2章提供了概率方法的详细讨论，包括EM算法。

1.3.4 线性模型

这些方法使用线性相关来模拟沿低维子空间的数据[467]。 例如，在图1.4的情况下，数据沿着二维空间中的1维线对齐。 通过使用回归分析确定通过这些点的最佳线。 通常，最小二乘法用于确定最佳的低维超平面。 数据点与该超平面的距离用于量化异常值分数，因为它们量化了与正常数据模型的偏差。 可以对这些分数应用极值分析以确定异常值。 例如，在图1.4的二维示例中，数据点的线性模型（xi，yi），i 1 ... N 就两个系数而言，a 和 b 可以如下创建：

$$\text{义} = \text{一} \cdot \text{XI} + \text{b} + \text{SI} \quad \forall \text{我} \in \{1 \ldots \text{N}\} \qquad (1.4)$$

在此，SI 表示剩余，这是建模误差。 系数a 和b 需要从数据学习，以最小化最小二乘误差，这是由Σ表示$_{i=1}^{N}$ 小号$_i^2$.
这是一个凸的非线性规划问题，其解可以在闭合中获得形成。 平方残差提供异常值。 可以使用极值分析来识别异常大的偏差，这应该被视为异常值。

维度降低和主成分分析（PCA）的概念非常相似[296]，除了它使用非参数方法来模拟数据相关性。 PCA可以通过多变量回归分析得出，通过确定最小化超平面的最小平方误差（即距离）的超平面。 换句话说，它提供了较低维度的子空间，在投影后具有最小的重建误差。 异常值具有较大的重建错误，因为它们不符合数据中的聚合子空间模式。 因此，重建误差可以用作异常值。 此外，主成分分析可用于噪声校正[21]，其中数据点的属性被修改以减少噪声。 局外人

singola@gmx.com

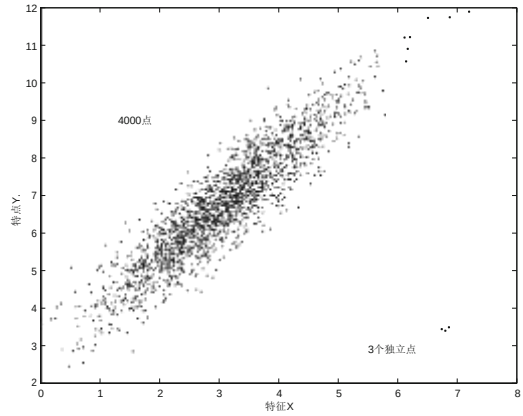


图1.5：小组异常可能对基于密度的方法构成挑战

积分可能比正常点更显着地纠正。 降维的方法是矩阵分解通用方法的特例；通用方法的主要优点是它甚至可以用于不完整的数据集。

维度降低和回归建模在原始属性方面特别难以解释。 这是因为子空间嵌入被定义为具有正或负系数的属性的线性组合。 根据数据属性的特定属性，这不容易直观地解释。 然而，某些形式的维数减少，例如非负矩阵分解，是高度可解释的。 第3 章讨论了异常值降低，回归分析和异常值检测的矩阵分解方法。 它们的自然非线性扩展，如内核PCA，内核SVM和神经网络，也在同一章中讨论。 此外，在章节中讨论了各种形式的非负矩阵分解8 和12。

1.3.4.1 光谱模型

许多矩阵分解方法（如PCA）也用于图形和网络的上下文中。 主要的不同之处在于如何为分解创建矩阵。 在某些类型的数据（例如图形和网络）中使用的这些方法的一些变体也称为光谱模型。 谱方法通常用于聚类图数据集，并且通常用于识别图的时间序列中的异常变化[280]。 谱方法与矩阵因子密切相关，也可用于此类设置[551]。 这样的模型将在本章节中讨论，3 4，图5，和12。

1.3.5 基于邻近的模型

基于邻近度的方法的想法是将异常值建模为基于相似性或距离函数与剩余数据隔离的点。基于邻近度的方法是异常值分析中最常用的方法之一。 基于邻近度的方法可以以三种方式之一应用，即聚类方法，基于密度的方法

singola@gmx.com

和最近邻居方法。 在聚类和其他基于密度的方法中，直接找到数据中的密集区域，并且将异常值定义为不在这些密集区域中的那些点。 或者，可以将异常值定义为远离密集区域的点。 聚类和基于密度的方法之间的主要差异是聚类方法对数据点进行分段，而基于密度的方法（如直方图）则对数据空间进行分段。 这是因为后一种情况的目标是估计数据空间中测试点的密度，这最好通过空间分割来实现。

在最近邻方法[317，456]中，每个数据的距离指向其k个最近邻居报告为离群其得分。通过选择$k> 1$ 的值，可以识别远离剩余数据集的小组紧密结合点并将其评分为异常值。将这样的数据点集合视为异常值是合理的，因为通常可以通过异常过程生成小的相关点集合。 例如，考虑图1.5中所示的情况，其中包含一个包含4000个数据点的大型集群，以及一小组孤立但三个间隔紧密且相关的异常。 这种情况非常普遍，因为由相同（罕见）过程引起的异常可能导致几组相同的小数据点。 在这种情况下，异常集内的点彼此接近，并且不能基于1最近邻距离来区分。 通过使用对数据的全局行为不敏感的某些类型的基于密度的算法，这种异常通常难以与噪声区分开。 另一方面，k最近邻方法有时可能是有效的。在图1.5的情况下，可以通过使用来识别这样的相关异常集k 3.第k个最近邻居得分提供数据集的离群值得分。 该方法通常在计算上是昂贵的，因为需要确定第k个最近邻居数据集中的每个点，对于包含的数据集需要$O（N^2）$个操作N 分。 但是，如果可以接受报告二进制标签而不是分数，这些距离计算中的许多可以被修剪，因为在少量距离计算之后可以将一些点显示为非异常值。 例如，如果计算的点的一小部分的距离到一个特定点之后“A”的k的“A” -nearest邻居距离比，所有的顶的下部r到目前为止发现的异常值，然后点'A'保证是非异常值。 因此，不需要执行到点'A'的进一步距离计算。 因此，异常值检测的二进制版本通常允许比问题的分数版本更快的算法。 后者总是与计算复杂度中的点数呈二次方相关。 在实际环境中，二次计算复杂度出乎意料地慢得多; 即使对于包含几十万个点的数据集，如果不利用某种形式的采样，通常也很难使用这些方法。

在聚类方法的情况下，第一步是使用聚类算法来确定数据集的密集区域。 在第二步中，使用对不同聚类的数据点的一些度量来计算数据点的离群值得分。 例如，在k-means聚类算法的情况下，数据点到其最近质心的距离可用于测量其作为异常值的倾向。 人们需要小心使用聚类方法，因为特定的数据分区（以及相应的异常值得分）可能与聚类方法的选择有显着差异。 因此，通常建议多次聚类数据并平均从不同运行中获得的分数[184，406]。这种方法的结果通常令人惊讶地强大。

基于密度的方法（如直方图）将数据空间划分为小区域，这些区域中的点数用于计算离群值得分。 基于密度

singola@gmx.com

当可以根据原始属性的组合呈现数据中的稀疏区域时，方法提供高水平的可解释性。 例如，考虑在以下属性子集上构造的稀疏区域：

$$\text{Age} \leq 20, \text{Salary} \geq 100,000$$

显然，这些约束定义了从语义角度高度可解释的数据空间的一部分。 它清楚地描述了为什么数据点应被视为异常值。 诸如核密度估计之类的一些其他方法不对数据空间进行划分，但是仍然关注于通过用更平滑的核函数替换空间分割来估计数据空间中的区域的密度。 第4章讨论了基于邻近的离群检测方法。

1.3.6 信息理论模型

许多上述用于离群值分析的模型使用各种形式的数据汇总，例如生成概率模型参数，聚类或低维表示超平面。 这些模型隐式地生成数据的小摘要，并且与此摘要的偏差被视为异常值。 信息理论措施也基于相同的原则，但是以间接的方式。 我们的想法是，外部人员会增加描述数据集所需的最小代码长度（即摘要的最小长度），因为它们表示与自然尝试汇总数据的偏差。 例如，请考虑以下两个字符串：

ABABABABABABABABABABABABABABABABAB ABABACABABABABABABABABAB
ABABABABAB

第二个字符串与第一个字符串具有相同的长度，并且仅在包含唯一符号“C”的单个位置处不同。 第一个字符串可以简洁地描述为“AB 17次”。 但是，第二个字符串具有对应于符号“C”的单个位置。 因此，第二个字符串不能再简洁地描述。 换句话说，字符串中某处存在符号“C”会增加其最小描述长度。 也很容易看出这个符号对应一个异常值。 信息理论模型与传统模型密切相关，因为两者都使用数据集的简洁表示作为比较基准。 例如，在多维数据集的情况下，两种类型的模型都使用以下简明描述：

- 概率模型根据生成模型参数描述数据集，例如高斯分布的混合或指数幂分布的混合[92]。
- 基于聚类或基于密度的摘要模型根据聚类描述，直方图或其他汇总表示描述数据集，以及最大误差容限[284]。
- PCA模型或光谱模型根据多维数据投影的低维子空间或网络的浓缩表示来描述数据[519]，其也被称为其潜在表示。
- 频繁模式挖掘方法根据频繁模式的底层代码簿来描述数据。 这些是用于信息理论异常检测[最常用的方法中42，151，497]。

singola@gmx.com

所有这些模型大致代表了代表总趋势的各个压缩组件的数据。 通常，异常值会根据这些压缩分量增加描述的长度，以达到相同的近似水平。 例如，具有异常值的数据集将需要更多数量的混合参数，聚类，基于PCA的子空间维度或频繁模式，以便实现相同的近似水平。 相应地，在信息理论方法中，关键思想是构建一个表示数据的代码簿，并将异常值定义为删除导致描述长度最大减少的点[151]，或最准确的汇总表示删除后的相同描述长度[284]。 术语“代码簿”在异常值分析中相当松散地定义，并且指的是数据的精简聚合分量，其中描述了数据。 编码的实际构造通常是启发式的，而有效的选择是该方法成功的关键。 一般而言，最小长度编码的判断为对于给定的数据集可以被用于表示目的[计算上难以解决的问题，并且因此各种启发式模型（或代码书）的42，151，284，497]。 在许多情况下，这些技术可以与用于异常值分析的常规数据汇总模型相关。 在一些情况下，在编码不是显式构造，但如熵或Kolmogorov复杂的措施，以便估计数据[的一个特定的C部的uneven-岬的水平用作替代352，312]。 可以选择性地探索具有更大不均匀性的区段以识别异常值。 这代表了信息理论模型的一个很好的用例，因为它在算法上比量化编码复杂度更简单，而不是实际构建编码。

传统模型通过直接将异常值定义为以固定压缩（例如，聚类或因式分解）以最不精确的方式表达（或偏离）的点，以互补的方式看待该问题。 另一方面，信息理论模型量化了在压缩固定误差（即，聚合偏差）时去除离群点的压缩尺寸的不同影响。 这两者显然是密切相关的，尽管前者是比后者更直接的得分方式。 由于信息-理论方法的措施是如何去连接的斯内德而言，从传统模式主要是双FF呃，他们经常用类似的方法与常规技术（例如，概率模型[92]，频繁模式挖掘[42，497]，直方图[284]或PCA [519]）来创建编码表示。 因此，大多数信息理论模型不能被视为与传统模型分离的单独族，并且将在本书的各个地方与传统模型一起讨论它们。 值得注意的是，由于其他点对总误差的影响，信息理论模型用于得分的间接方法有时会使分数变钝。 因此，信息理论模型通常不会胜过传统的同类模型。 因此，出现了最佳使用情况，其中量化编码成本在算法上比直接测量偏差更方便。

1.3.7 High-Dimensional Outlier Detection

对于离群检测，高维情况尤其具有挑战性。 这种行为的原因是许多维度可能是噪声并且与异常检测无关，这也可能增加成对距离变得更相似的倾向。 这里的关键点是不相关的属性对距离计算的准确性有稀释影响，因此得出的异常值得分也可能不准确。 当使用基于距离的算法来评估异常值时，人们经常观察到距离集中中弱相关和不相关属性的影响。 在高维空间，

singola@gmx.com

数据变得越来越稀疏，并且所有对数据点变得彼此[几乎等距离25，263]。 结果，异常值得分变得彼此不易区分。

在这种情况下，异常值最好在相关属性的低维局部子空间中强调。 这种方法被称为子空间异常值检测[4]，这是异常值分析领域中一类重要的算法。 子空间异常值检测中的假设是异常值通常隐藏在低维子空间的异常局部行为中，并且这种异常行为被全维分析掩盖。 因此，明确搜索最佳地强调点的异常行为的子空间通常是富有成效的。 该方法是（全维）聚类和（全数据）回归分析的推广。 它结合了局部数据模式分析和子空间分析，以挖掘重要的异常值。 这可能是一个巨大的挑战，因为在高维度上同时发现相关数据位置和子空间在计算上可能非常困难。 31，35]，这种技术只能通过识别多个相关的子空间，并从这些二FF erent子空间的预测组合被有意义地使用。 这种方法是密切相关的离群合奏[概念31，35]，这将在下一节以及在章中讨论6。 子空间方法对于解释异常值很有用，尤其是在根据原始属性描述子空间时。 在这种情况下，算法的输出提供了属性的特定组合以及与异常特征相关的数据局部性。 在需要从高维数据集中识别少量解释性属性的情况下，这种类型的可解释性是有用的。

第5章讨论了高维离群点检测的方法。

1.4 异常合奏

在许多数据挖掘问题中，例如聚类和分类，使用各种元算法来提高底层解决方案的稳健性。 这种元算法结合了多种算法的输出，被称为集合。 例如，在网络连接CLASSI阳离子共同集成方法包括装袋，子取样，升压和堆叠[11，33，176]。 同样，集合方法通常用于提高聚类的质量[23]。 因此，很自然地会问这种元算法是否也存在用于异常值检测。 答案是有效的，尽管与其他问题相比，异常值检测的元算法的工作相对较新，而分类已经很好地建立了分类和聚类。 正式确定这些问题的立场文件可以在[31]中找到，有关异常集合的书可以在[35]中找到。 近年来，在异常集合领域已经取得了显着的理论和算法进步[32]。 本章将对异常集合的领域进行广泛的概述，第6章将对此进行更详细的讨论。 。 异常值分析中有两种主要类型的集合：

- 在顺序集合中，顺序地应用给定算法或算法集，使得算法的未来应用受到先前应用的影响，无论是用于分析的基础数据的修改还是根据算法的特定选择。 最终结果是异常分析算法的最后应用的加权组合或最终结果。 例如，在分类问题的上下文中，可以将提升方法视为顺序集合的示例。

singola@gmx.com

Algorithm SequentialEnsemble(Data Set: D
Base Algorithms: $A_1 \ldots A_r$)
开始
$j = 1$;
重复
根据过去执行的结果选择算法 A_j;
根据过去执行的结果创建一个新的数据集 D_j = $f_j(D)$;
Apply A_j to D_j;
$j = j + 1$;
直到（终止）;
根据先前执行结果的组合**报告**异常值;
结束

图1.6：顺序集合框架

- 在独立的集合中，不同的算法或相同算法的不同实例应用于完整数据或数据部分。关于所应用的数据和算法的选择与从这些不同的算法执行获得的结果无关。 将不同算法执行的结果组合在一起，以获得更强大的异常值。

从根本上讲，异常集合在基础理论基础方面与分类集合并不是很不相同[32]。 即使异常值检测是一个无监督的问题，分类中的基本偏差 - 方差理论也可以通过将基础标记视为未观察到来适应异常值检测[32]。 结果，诸如装袋和子采样的许多自然集合方法可以容易地推广到具有微小变化的异常值检测。

1.4.1 顺序合奏

在顺序集合中，将一个或多个异常值检测算法顺序地应用于全部或部分数据。 该方法的核心原则是该算法的每个应用程序都能够通过修改算法或数据集实现更加精确的执行。 因此，取决于方法，可以在顺序执行中改变数据集或算法。 如果需要，这种方法既可以应用固定次数，也可以执行收敛。 图1.6提供了顺序集成算法的广泛框架。

在每次迭代中，基于先前执行的结果，对重新定义的数据使用连续重新定义的算法。函数fj（）用于创建数据的改进，其可以对应于数据子集选择，属性子集选择或通用数据转换方法。 上面的描述以非常一般的形式提供，并且可以从该框架实例化许多特殊情况。 例如，在实践中，只有单个算法可以用于数据的连续修改，因为数据随着时间的推移而被重新定义。 顺序集合可以应用于固定数量的迭代

singola@gmx.com

Algorithm IndependentEnsemble(Data Set: D
Base Algorithms: $A_1 \ldots A_r$)
开始
$j = 1$;
重复
选择算法A_j;
从D创建一个新的数据集fj（D）；
将 A_j 应用于fj（D）；
$j = j+1$;
直到（终止）；
根据先前执行结果的组合**报告**异常值;
结束

图1.7：独立集合框架

或收敛。 顺序集合的广泛原则是，通过连续算法执行获得更多的数据知识有助于关注可提供新见解的技术和数据部分。

在异常值分析文献中，序贯集合作为通用元算法尚未得到充分的探索。 然而，异常值的许多特定技术使用的方法可以被认为是连续集合的特殊情况。 一个典型的例子是使用两阶段算法来构建普通数据模型。 在第一阶段，使用异常检测算法来消除明显的外部因素。 在第二阶段，在去除这些明显的异常值之后构建更稳健的正态模型。 因此，第二阶段中的离群值分析更准确，因为已经去除了污染正常数据模型的许多训练点。 这种方法通常用于基于聚类的异常值分析（用于在后期构建更强大的聚类）[70]，或更强大的直方图构造和密度估计（见第4章）。

1.4.2 独立合奏团

在独立的集合中，算法的不同实例或数据的不同部分用于异常值分析。 或者，可以使用不同的初始化，参数集或随机种子来应用相同的算法。 可以组合这些不同算法执行的结果，以获得更稳健的异常值分数。 这种算法包括当今使用的绝大多数异常集合方法。 在图1.7的伪代码描述中提供了独立集合算法的通用描述。

独立集合的广泛原则是，不同的算法可能在数据的不同部分上执行得更好；因此，这些算法的结果组合可能会提供比任何单个集合组件更强大的结果。 因此，结果输出不再依赖于特定算法或数据集的特定伪像。 独立集合经常用于高维异常值检测，因为它们能够探索可能找到不同类型偏差的数据的不同子空间。 实际上，子空间的面积

singola@gmx.com

异常值检测与异常值集合分析密切相关（见第5章）。

有许多不同的方法可以利用不同的算法和训练数据集进行模型组合。 例如，在方法[31，32， 344，367]样品从基础数据的子空间，以独立地得分从每个这些执行的异常值。 然后，来自这些不同执行的分数被统一为单个点特定值。 类似地，如装袋和子采样，结合从CLASSI音响阳离子二FF erent训练数据集的结果的方法，也已推广到异常值检测[31，32]。 在某些情况下，通过在异常值评分算法中进行随机选择来构建随机模型[368]。 这些方法将在本章节中讨论5 和6。

1.5 分析的基本数据类型

我们上述讨论的大部分内容都集中在多维数值数据上。 此外，假设数据记录彼此独立。 但是，实际上，基础数据在属性类型和点对点依赖性方面可能更复杂。 本节将讨论此类实际数据类型的一些示例。

1.5.1 分类，文本和混合属性

实际应用程序中的许多数据集可能包含带有离散无序值的分类属性。 例如，人口统计数据可能包含种族，性别或邮政编码等属性。 这些属性值不是有序的，因此需要不同的分析技术。 混合属性数据包含数字和分类属性。 大多数现有模型可以扩展到这种情况。 在许多情况下，主要挑战是构建距离（或相似性）函数，该函数对于离散数据的情况保持语义上有意义。

当属性的可能值的数量不是太大时，基于回归的模型可以以有限的方式使用离散的属性值。 典型的方法是通过为每个分类值创建一个属性，将离散数据转换为二进制数据。 然后可以将诸如主成分分析的回归模型应用于该二进制数据集。 这些方法可以更容易地扩展到文本，其中在字频率之间存在固有的顺序。 在这种情况下，单词出现之间的相关性可用于创建回归模型。 事实上，一些最成功的文本去噪模型是基于潜在语义分析（LSA），这是一种线性回归分析[162]。]。 文本和分类数据的其他常用方法包括聚类[29]，基于接近度的方法[622]，概率模型[578]，并且基于频繁模式挖掘[方法42，253，497]。 第8章讨论了分类，文本和混合属性数据集中离群值检测的方法。

1.5.2 当数据值具有依赖关系时

本章中上述讨论的大部分内容都是关于常见的多维场景，其中假设数据记录可以彼此独立地处理。 实际上，不同的数据值可以在时间上，空间上或通过数据项之间的显式网络关系链接彼此相关。 即使在定义时，这种依赖性的存在也极大地改变了异常检测过程

singola@gmx.com

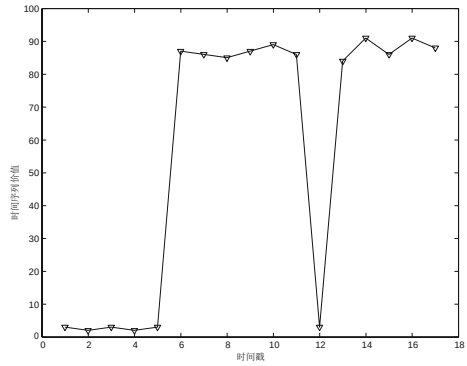


图1.8：时间序列示例

水平。 在这种情况下，数据项的预期值受其上下文依赖性的影响，因此异常值是基于这种上下文建模的偏差来定义的。 当单个数据项（例如，来自时间序列的值）由于其与其相关数据项的关系而被声明为异常时，它被称为上下文异常值或异常。 这是因为这种异常值只能在其与时间邻域中的项目的关系的背景下被理解。 这种异常值有时也被称为条件异常[503]。 例如，时间序列中的突然峰值是上下文异常，因为它与其最新时间戳的值非常不同;关键是要了解最新的时间戳是否定义了该系列的预期价值。 与预期值的偏差表示异常值。

当一组数据项被声明为一组点异常时，它被称为集体异常或异常值。 例如，股票代码值随时间的异常和快速振荡可以被认为是集体异常，并且它包括振荡中的所有数据项。 实际上，依赖于面向对象的数据中的所有异常都是上下文或集体异常，因为它们基于与相邻数据点的关系来计算期望值，以便确定意外模式。 此外，在此类数据集中，通常有多种方法可以模拟异常，具体取决于分析师可能正在寻找的内容。 本节介绍了此类数据域的一些示例。

1.5.2.1 时间序列数据和数据流

时间序列包含一组通常通过连续测量随时间生成的值。 因此，连续时间戳中的值不会发生非常显着的变化，也不会以平滑的方式变化。 在这种情况下，基础数据记录的突然变化可被视为异常事件。 因此，异常点的按照时间序列的发现密切相关异常事件检测的问题，并且这样的事件往往表现为对相关时间戳[要么上下文或集体异常9，19，315]。 事件通常由底层系统的突然变化产生，并且可能对分析师有相当大的兴趣。 例如，考虑连续时间戳上的以下时间序列值：

3, 2, 3, 2, 3, 87, 86, 85 87, 89, 86, 3, 84, 91, 86, 91, 88

时间序列如图1.8所示。 很明显，时间戳6处的数据值从3突然变化到87.这对应于异常值。随后，

singola@gmx.com

数据稳定在这个值，这成为新常态。 在时间戳12处，数据值再次下降到3.即使之前遇到过该数据值，由于连续数据值的突然变化，它仍然被认为是异常值。 因此，重要的是要理解在这种情况下，将数据值彼此独立地处理对于异常检测是没有帮助的，因为数据值高度受到数据点的相邻值的影响。 换句话说，时间背景很重要。 因此，时间序列数据中的离群值检测问题与变化检测问题高度相关，因为数据值的正常模型受时间顺序的相邻性高度控制。 当遇到全新的数据值时，它们被称为新奇数据[391，392，388]，虽然异常检测是相关的任何形式的突变，而不是只有新的数据值。

应该强调的是，变化分析和异常检测（在时间数据中）是密切相关的问题，但它们不一定相同。 时态数据集的变化可能以两种可能的方式之一发生：

- 在数据流中的值和趋势随时间缓慢变化，即称为概念漂移[现象390，10]。 在这种情况下，概念漂移只能通过较长时间的仔细分析来检测，并且在许多情况下并不是立即显而易见的。

- 数据流中的价值和趋势突然发生变化，从而立即引起人们对基础数据生成机制以某种方式发生变化的怀疑。

在这两种情形中，只有第二种情景可用于识别异常值。 同样很容易看出第二种情景与霍金斯对异常值的定义之间的相似之处[249]，这是本章开头介绍的。

这种情况下的常见挑战是在新数据值到达时实时执行异常值检测。 许多情况下的变化分析和临时数据中的异常检测过于紧密集成，无法单独处理。 在这样的设置中，一个的解决方案可以用于另一个，反之亦然。 另一方面，时间数据中的异常检测的建模公式非常多样化，并非所有这些都与变化检测直接相关。 通常，在线分析适用于改变检测，而o ffl ine分析可能会探索数据的其他不寻常方面。 一些例子如下：

- 当数据是时间序列（例如，传感器数据）的形式时，趋势的大的变化可以对应于异常。 使用基于窗口的分析可以发现这些与预测值的偏差。 在某些情况下，可能需要确定异常形状的时间序列子序列而不是数据中的变化点。

- 对于多维数据流，流数据的聚合分布的变化可以对应于异常事件。 例如，网络入侵事件可能会导致网络流中的聚合更改点。 另一方面，个别点新奇可能或可能不对应于总变化点。 后一种情况类似于多维异常检测，其具有流式方案的效率约束。

第9章讨论了时间序列数据和多维数据流中异常检测的方法。

singola@gmx.com

1.5.2.2 离散序列

许多离散的基于序列的应用，例如入侵检测和欺诈检测，本质上都是时间性的。 这种情况可以被认为是时间序列数据的分类或离散模拟，其中各个位置包含分类（符号）值。 离散序列本质上不一定是时间的，而是可以基于它们相对于彼此的相对位置。 一个例子是生物数据的情况，其中序列由它们的相对位置来定义。

与连续数据的自回归模型一样，可以使用（通常是马尔可夫）基于预测的技术来预测序列中单个位置的值。 与预测值的偏差被识别为上下文异常值。 通常希望在这些设置中实时执行预测。 在其他情况下，异常事件只能通过子序列在多个时间戳上显示的正常模式的变化来识别。 这类似于时间序列数据中异常形状检测的问题，它代表一组集体异常值。

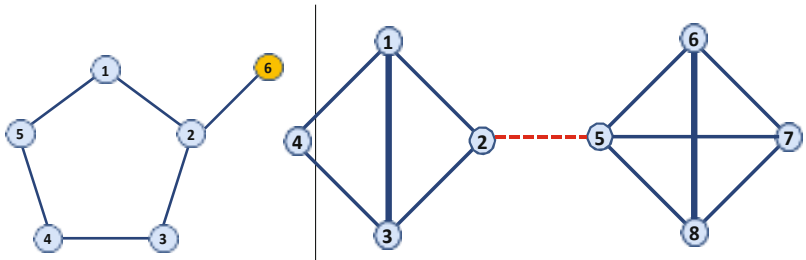
因此，离散序列类似于连续序列，除了各个位置中的临界值需要使用不同的相似性函数，表示数据结构和预测技术。 例如，离散序列预测需要（更复杂的）马尔可夫模型而不是（更简单）自回归技术。 然而，两种情况下的问题表述在概念上是相似的。 所使用的具体技术是不同的，因为数字时间序列值是有序的，并且在连续光谱中是可比较的，而离散值则不是。 由于这些差异，离散序列的情况已经在时间序列数据的不同章节中得到了解决。

离散数据在许多实际应用中很常见。 大多数生物序列是离散的，因此每个位置的值来自一组分类可能性。 同样，基于主机的入侵应用程序通常会导致离散数据，因为大量诊断事件是从一组离散的实例中提取的[126]。 第10章讨论了离散序列中的异常检测方法。

1.5.2.3 空间数据

在空间数据中，在空间位置处测量许多非空间属性（例如，温度，压力，图像像素颜色强度）。 这些值中不寻常的局部变化被视为异常值。 应该指出的是，时间数据中的异常检测与空间数据中的异常检测有一些相似之处[523]]。 两者通常都要求感兴趣的属性表现出一定程度的连续性。 例如，考虑测量可以与时间戳和空间坐标相关联的温度的测量。 正如预期连续时间戳的温度变化不太大（时间连续性）一样，也预期空间上接近的位置的温度变化不太大（空间连续性）。 事实上，海面温度和压力的这种不寻常的空间变化[523用于识别基础数据中的重要和异常时空事件（例如，形成旋风）。 时空数据是空间和时间数据的推广，任何领域中使用的方法通常可以推广到这些情景。 第11章讨论了在空间和时空数据中发现异常值的方法。

singola@gmx.com



(a) 节点异常值 （b）边缘异常值

图1.9：节点和边缘异常值的示例

1.5.2.4 网络和图形数据

在网络或图形数据中，数据值可以对应于网络中的节点，并且数据值之间的关系可以对应于网络中的边缘。 在这种情况下，异常值可以以不同的方式建模，这取决于节点在与其他节点的关系或边缘本身方面的不规则性。 例如，在其位置内显示其结构不规则性的节点可以被认为是异常值[41]。 类似地，连接的节点的完全不同的社区的边缘可被认为是关系或社区离群[17，214]。 在图1.9中，示出了网络中的异常值的两个示例。 图1.9（a）示出了节点异常值的示例，因为节点6具有与其他节点明显不同的异常位置结构。 在另一方面，所述边缘（2，5）在图1.9（b）可以被认为是异常值的关系或社区离群，因为它连接两个不同的社区。 因此，在诸如图之类的复杂数据中，异常值的定义具有更大的复杂性和灵活性。 也没有唯一的方法来定义异常值，它严重依赖于手头的应用领域。 一般而言，数据越复杂，分析师就先前推断出建模目的的正常情况就越多。

也可以将不同类型的依赖关系组合用于异常值建模。 例如，图表可能是暂时的。 在这种情况下，数据可能具有随时间变化和相互影响的结构和时间依赖性[17]。 因此，可以根据底层网络社区或距离结构的显着变化来定义异常值。 这些模型结合了网络分析和变化检测，以检测结构和时间异常值。 第12章提供了图中时间和非时间异常值检测方法的详细讨论。 有关检验在[可用的14，43，457]。

1.6 监督离群检测

在许多情况下，可以使用先前的异常值示例。 数据的子集可以被标记为异常，而剩余的数据可以被认为是正常的。 在这种情况下，异常识别过程被称为监督异常值检测，因为标签用于训练可以确定特定类型异常的模型。 因此，监督模型通常会从无监督的案例中提供非常不同的结果。 例如，请考虑以下时间序列：

3, 2, 3, 2, 3, 87, 2, 2, 3, 3, 3, 84, 91, 86, 91, 81

singola@gmx.com

在这种情况下，数据值的突然变化（在87和84处）可能被认为是无监督情景中的异常。 然而，在诸如信用卡交易等级的应用中，先前标记的时间序列示例可能表明数据的高连续值应该被认为是异常的。 在这种情况下，87的第一次出现不应被视为异常，而84的出现及其下列值应被视为（统称）异常。

作为一般规则，当标签可用时，应始终使用监督，因为它能够发现应用程序特定的感兴趣异常。 监督异常值检测是分类问题的一个特殊情况（特殊情况）。 该问题的主要特征是标签在相对存在方面极不平衡[132]。 由于异常远比正常点少，因此现有的分类器可以将所有测试点预测为正常点，并且仍然可以实现极佳的准确性。 然而，从实际的观点来看，这样的结果是没有用的。 因此，分类器被调整，因此异常类别的分类错误比大多数类别的分类错误更严重。 这个想法是，最好将负面类别预测为异常（误报），而不是错过真正的异常值（假阴性）。 与其他分类应用相比，这导致了误报和漏报之间的不同交易。 这些方法被称为成本敏感型学习，因为不同的差异成本适用于不同的类来规范这些交易。

受监督的设置还支持分类问题的其他几种变体，这些变体非常具有挑战性：

- 可以使用有限数量的正（异常）类的实例，而“正常”的例子可能包含未知比例的异常值[183]。 这被称为机器学习中的正无标记分类（PUC）问题。 除了分类模型需要更多地认识到负（未标记）类中的污染物之外，这种变化仍然与完全监督的稀有类情景非常相似。

- 正常和异常类的子集的唯一实例是可用的，但有些异常类可以从训练数据[丢失388，389，538]。 这种情况在诸如入侵检测等场景中非常常见，其中一些入侵可能是已知的，但随着时间的推移不断发现其他新类型的入侵。 这是用于离群检测的半监督设置。 在这种情况下可能需要使用监督和非监督方法的组合。

- 在主动学习中，标签获取的问题与学习过程配对[431]。 主要假设是获取异常值的示例是昂贵的，因此选择正确的标签示例以便使用最少数量的标签执行准确的分类是很重要的。

第7章讨论了用于异常检测的监督方法。

1.7 异常值评估技术

关键问题是如何评估异常值检测算法的有效性。 不幸的是，这通常是一项艰巨的任务，因为通过定义，异常值很少。 这意味着通常无法将数据点的地面实况标记为异常值或非异常值。 对于无监督算法尤其如此，因为如果

singola@gmx.com

确实可以获得地面实况，它可以用来创建一个更有效的监督算法。 在无监督的场景中（没有地面实况），通常很难以严格的方式判断底层算法的有效性。 因此，许多研究文献使用案例研究来提供对无监督情景中潜在异常值的直观和定性评估。

在数据聚类等其他无监督问题中，常见的方法是使用内部有效性度量，其中使用“优度”模型来度量算法的有效性。 例如，数据聚类中良好性的常用度量是聚类的均方半径。 这些措施的主要问题是它们只能提供“善”模型与学习模型的匹配程度。 毕竟，在无人监督的问题中，无法知道“正确”的善良模式; 矛盾的是，如果我们知道这个正确的模型，那么我们应该在算法中使用它而不是用于评估。 事实上，通过选择与善良模型相关的算法来比较这种内部有效性模型是相对容易的; 这个问题在集群领域是众所周知的[33]。 这也被称为内部评估中的过度补偿问题。 在异常值检测中，这个问题要严重得多，因为异常值标签的少量变化会极大地影响性能。 例如，基于距离的内部测量将优于基于距离的算法而不是线性（例如，基于PCA的）技术。 相反，内部有效性的线性模型有利于基于PCA的技术而不是基于距离的算法。 因此，内部有效性测量很少用于异常检测，这似乎是比数据聚类社区采用的更明智的方法。

在异常检测中，更合理（尽管不完美）的方法是使用外部有效性测量。 在某些情况下，数据集可以根据不平衡的分类问题进行调整，稀有标签可以用作地面实况异常值的替代品。 在这种情况下，出现了一个自然的问题，即地面实况如何用于评估有效性。 大多数异常检测算法输出异常值，并且该分数的阈值用于将分数转换为异常值标签。 如果选择阈值过于严格以最小化声明的异常值的数量，则算法将错过真正的异常点（假阴性）。 另一方面，如果算法将过多的数据点声明为异常值，则会导致过多的误报。

对于离群值得分上的任何给定阈值t ，声明的离群值集合由S（t）表示。 随着t的变化，S（t）的大小也会发生变化。 tt 表示数据集中异常值的真实集（地面实例集）。 然后，对于任何给定的阈值t，精度被定义为报告的异常值的百分比，其真实地证明是异常值。

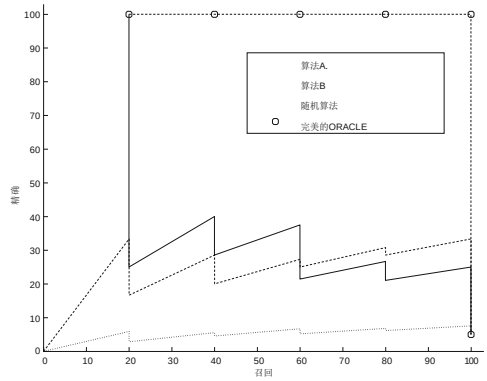
$$Precision(t) = 100 \cdot \frac{|S(t) \cap tt|}{|S(t)|} \tag{1.5}$$

精度（t）的值在t中不一定是单调的，因为分子和分母都可能随着t的变化而变化。 召回相应地定义为在阈值t处被报告为异常值的地面实况异常值的百分比。

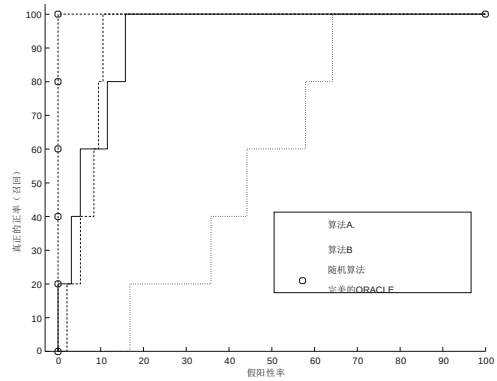
$$Recall(t) = 100 \cdot \frac{|S(t) \cap tt|}{|tt|} \tag{1.6}$$

通过改变参数t，可以绘制精度和召回之间的曲线。 这被称为精确回忆曲线。 该曲线不一定是单调的。

singola@gmx.com



(a) 精确调用



（b）接收器工作特性图1.10：精确调用和接收器工作特性曲线

算法	地面实况异常值排名
算法A.	1, 5, 8, 15, 20
算法B.	3, 7, 11, 13, 15
随机算法	17, 36, 45, 59, 66
完美的Oracle	1, 2, 3, 4, 5

表1.2：地面实况异常值的等级可用于构建精确召回曲线

对于更有效的算法，高精度值通常可能对应于较低的召回值，反之亦然。 当通过异常值得分排序时，还可以通过使用数据点的等级上的阈值来生成精确回忆（PR）曲线。 在异常值得分中没有联系，基于等级和基于分数的PR曲线将是相同的。

接收器工作特性曲线（ROC）与精确回忆曲线密切相关，但有时在视觉上更直观。在这种情况下，真实阳性率与虚假阳性率进行比较。 真正的正率TPR（t）的定义与召回的方式相同。 假阳性率FPR（t）是从地面实况阴性中错误报告的阳性的百分比。 换句话说，假阳性率是一种“坏”回忆，它报告错误报告为异常值的阴性百分比。因此，对于具有地面实况正数tt的数据集D，这些定义如下：

$$TPR(t) = Recall(t) = 100 \cdot \frac{|S(t) \cap tt|}{|tt|} \tag{1.7}$$

$$FPR(t) = BadRecall(t) = 100 \cdot \frac{|S(t) - tt|}{|D - tt|} \tag{1.8}$$

因此，ROC曲线绘制的“坏”召回（FPR（t在X轴）），和“好”召回（TPR（牛逼的））ÿ轴摆动。 注意，好的和坏的召回都会随着阈值t 的更宽松的值单调增加，在阈值t 处报告更多的异常值。 因此，也就ROC曲线的端点总是在（0，0）和（100，100），和随机方法有望沿着连接这些点的对角线表现出的性能。 在该对角线上方获得的升力提供了额外精度的概念

singola@gmx.com

随机方法的方法。 ROC曲线只是一种比精确回忆曲线更具特色的交易特征，尽管它具有单调且在升力特性方面更容易解释的优点。

为了说明从这些不同的图形表示中获得的见解，请考虑具有100个点的数据集的示例，其中五个点是异常值。 两个算法A 和B 应用于该数据集，其将所有数据点从1到100排名，较低的等级表示更大的异常倾向。 因此，可以通过确定5个基础事实异常值点的等级来生成精度和召回值。 见表1.2对于不同的算法，已经说明了5个基本事实异常值的一些假设排名。 此外，已经指出了随机算法的基础真实等级。 该算法输出每个点的随机异常值分数。 类似地，表中还示出了将正确的前5个点排列为异常值的“完美神谕”算法的等级。 此假设输出的异常值得分的相应PR曲线如图1.10所示（一个）。 除了oracle算法之外，所有trade-o ff曲线都是非单调的。 这是因为在等级阈值的任何特定放宽时发现新的异常值导致精度的峰值，这在较高的召回值下变得不那么明显。 相应的ROC曲线如图1.10（b）所示。 与PR曲线不同，此曲线显然是单调的。

这些曲线真正告诉我们什么？对于一条曲线严格支配另一条曲线的情况，两种算法之间的相对优势是明确的。 例如，很明显oracle算法优于所有算法，并且随机算法不如所有其他算法。 另一方面，算法 A 和B 在ROC曲线的不同部分表现出更好的性能。 在这种情况下，很难说一种算法是严格优越的。 从表1.2可以看出，算法A 非常高度地排列了三个正确的地面实况异常值，但其余两个异常值排名很差。 在算法B的情况下虽然所有五个异常值在等级阈值方面更早确定，但排名最高的异常值并不像算法A的情况那样好。 相应地，算法A 在PR（或ROC）曲线的早期部分占主导地位，而算法B 在后期部分占主导地位。 一些从业者使用ROC曲线下的区域作为算法整体有效性的代表。 梯形法则用于计算面积，其中阶梯状ROC曲线用更凸的近似值代替。

1.7.1 Interpreting the ROC AUC

ROC AUC具有以下简单的概率解释[246]：

定理1.7.1 给定一组点的排序或得分，按照它们倾向于异常值的顺序（较高的等级/分数表示更大的离群值），ROC AUC等于随机选择的异常值对的概率。 排名正确（或按正确顺序评分）。

换句话说，还可以通过计算数据集中所有异常值对的上述均值来定义ROC AUC：

$$\text{ROC AUC} = \text{MEAN}_{X_i \in G, X_j \in D-G} \begin{cases} 1 & \overline{\text{Xi}} \text{ 排名/得分高于 Xj} \\ 0。 & 5\ \text{Xi 排名/得分等于 Xj} \\ & \text{Xi 排名/得分低于 Xj} \end{cases} \tag{1.9}$$

这个定义的一个很好的特点是很容易直观地理解为什么随机算法会提供大约0的AUC。 这个定义也与此有关

singola@gmx.com

肯德尔秩相关COE FFI cient，在其中执行一个类似的计算在所有对对象的，而不是仅离群-内点对，并且该范围[2] 所述回报函数的从吸入{-1，0，+ 1}而不是{ 0，0。 5，1}。

应该非常小心地使用诸如AUC之类的措施，因为所有部分都是如此
对于不同的应用，ROC曲线可能不是同等重要的。 这是因为ROC曲线的初始部分通常更为重要。 例如，对于包含1000个点的数据集，无论异常数据点是排在第501位还是第601位，都几乎没有什么差别。 在另一方面，它使得大量的二FF erence离群值的数据点是否在1排列第一或101个位置。 ROC的AUC几乎没有区分这两种类型的错误。 从这个意义上讲，在许多情况下，精确度等测量有时会更加真实。 其他重量级的测量，如标准化折扣累积增益（NDCG），也可以从信息检索和推荐文献到异常值检测进行调整[34]。 该度量通过给予排在列表顶部的异常值更大的信用来计算排序列表的效用。

1.7.2基准测试中的常见错误

在异常值检测算法的基准测试期间出现的常见错误发生在算法依赖于一个或多个用户定义的参数的情况下。 例如，k-最近邻居异常值检测器基于其k-最近邻距离（其中k）对数据点进行评分是用户定义的参数。 通常重复运行算法以选择优化ROC AUC的最佳参数。 这种方法在离群值检测中是不可接受的，因为人们在选择参数时有效地使用了离群值标签的知识。 换句话说，该算法不再是无人监督的。 在像异常值检测这样的问题中，在一对算法之间进行基准测试的唯一正确方法是在“合理”的参数范围内运行并比较
两种算法使用一些中心估计器来表示运行结果。 例如，可以比较两种算法的中值AUC性能或箱形图性能[3] 而不是各种参数选择。 此外，不同的探测器可能需要识别完全不同的参数范围，这为比较目的带来了进一步的挑战。 例如，一类支持向量机
与最近邻检测器相比，可能具有完全不同的参数选择，这进一步使对其相对性能的正确理解变得复杂。 与监督设置不同，其中交叉验证可以为每个分类器选择近似最佳参数值，每个异常值检测器的合理范围通常基于数据集的简单元特征（例如其大小和维度）来设置。 。 显然，这些选择需要分析师对各种算法的一些理解和经验。 关于各种算法的正确参数选择，研究文献中仅提供有限的指导。

一个自然的问题是，是否可以使用每个候选算法的最佳参数选择来比较它们。 毕竟，这种方法似乎不支持任何特定算法而不是另一种算法，因为所有算法都用类似的方法播种

[2]在肯德尔等级相关系数的情况下，协议奖励为+1，而分歧则用1进行处罚。 中性值为0的抵消值。 异常值对或内部对总是属于中性类别。 结果，肯德尔系数的绝对量值对数据中异常值的比例更敏感。

[3]有关示例，请参见第6 章的图6.1 。 箱线图在部分正式出台2.2.2.3 的
Chapter 2.

singola@gmx.com

知识。 这样做有两个问题。 首先，在无监督问题中无法知道最佳参数选择，因此结果不会影响分析师在实际环境中的经验。 其次，这种评估将有利于更不稳定的算法，这种算法很容易过度进入特定的参数选择。 如果稳定算法在大多数情况下比不稳定算法表现更好，这将为分析师提供算法的偏斜视图，但不稳定算法在非常特定的参数选择下表现得非常好。 毕竟，在无人监督的环境中，正确能够猜测这种参数选择的可能性类似于在大海捞针中磕磕碰碰。 在比较一对探测器时避免参数化偏差的第二种方法[32]是在不同的设置上创建算法执行的整体平均值，并比较不同检测器的整体AUC。 这真实地提供了各种算法的最佳可能化身的相对性能的概念。 在一天结束时，比较两种算法在某种程度上是无监督设置中的艺术形式，并且必须至少依赖于分析师在进行适当的实验设计选择时的经验和良好判断。

1.8 结论和总结

异常值检测的问题在许多领域中都有应用，在这些领域中，需要确定底层生成过程中的有趣和不寻常的事件。 所有异常值检测方法的核心是创建表征正常数据的概率，统计或算法模型。 与该模型的偏差用于识别异常值。 对底层数据的良好的领域特定知识通常是设计简单而准确的模型的关键，这些模型不会覆盖基础数据。 当不同的数据点之间存在显着的关系时，异常值检测的问题变得尤其具有挑战性。 这是时间序列和网络数据的情况，其中数据点（无论是时间的还是结构的）之间的关系中的模式在定义异常值中起关键作用。 异常值分析具有进一步研究的巨大空间，特别是在结构和时间分析领域。

1.9 书目调查

已经就异常值分析问题撰写了大量书籍和调查。 经典的书籍[74，249，467]在这方面的大部分已在统计界的角度写的。 这些书中的大多数是在更广泛采用数据库技术之前编写的，因此不是从计算角度编写的。 最近，计算机科学界已经对这个问题进行了相当广泛的研究。 这些工作考虑了异常值检测的实际方面，对应于数据可能非常大的情况，或者可能具有非常高的维度。 还编写了许多调查，从不同的观点，方法或数据类型讨论异常值的概念[38]，77，125，126，313，388，389]。 这些调查研究离群点检测从视图二FF erent点，如神经网络的设置[388，389]或一类设置[313]。 其中，Chandola等人的调查。 [125]是最新的，可以说是最全面的。 这个优秀的评论涵盖了从多个社区的角度来看的异常检测。 的各种孤立点检测算法的详细实验比较可见[35，114，184，221，419]。 本章讨论的基本模型也得到了广泛的研究，并在文献中得到了广泛的研究。 这些方法的详细信息（以及相关的

singola@gmx.com

附录书目将在后面的章节中提供。 在这里，只涉及每个领域最重要的作品。 [467] 中介绍了基于回归的建模的关键统计技术。 1.2 节中讨论的Z值测试通常用于统计文献中，并且还可以获得有限样本量的许多变体，例如Grubb测试[225]和t值测试。 用于无监督数据集建模的基本EM算法首先在[164]中提出，并用于[578]中的异常值检测。 主成分分析（PCA）的非参数技术在章节中讨论
在[296]中很好地描述了1.2 。 PCA的核心技术扩展到文本（有一些微小的变化）作为潜在语义索引[162]。 了多种用于异常检测的基于距离的方法中提出了[317，456，533]和in[提出用于异常检测的基于密度的方法96]。 解释基于距离的异常值的方法首先在[318]中提出。 了多种用于孤立点检测信息理论方法在[讨论42，57，92，122，151，256，257，352，497]。

的高维的应用程序（诸如集群和最近邻搜索）行为差问题已在文献[几种现有作品已经观察到5，7， 8，25，263]。 在[4]中首次提出了高维离群点检测的问题。 为孤立点检测的子空间的方法在本文中被提出，并且最近的一些其它的方法都遵循的工作[的类似线308，327，402，403，404，406，604，
605, 606, 607, 619].

已经在不同的数据域的背景下广泛研究了异常值。 虽然数字数据是最常用的研究的情况下，众多的方法也被提出亲对分类和混合数据[38，578]。 文献语料库中无监督离群点检测的方法在[240]中提出。 在文献中也广泛研究了检测具有依赖性的异常值的问题。 用于检测按时间序列和流异常值和变化的方法中，提出了[9，17，19，29，310，311，312，315]。 新奇检测[388]是至异常值分析密切相关的区域，并且它往往是在监督模式，其中从数据流中的新的类实时[检测到的上下文中研究391，392]与使用的学习方法。 然而，在无监督场景中也经常研究新颖性检测，特别是在文本流中的主题检测和跟踪中的第一个故事检测的背景下[622]。 空间离群值[2，324，376，487，488，489，490与发现时间数据中的异常值的问题密切相关，因为这些数据也显示空间连续性，正如时间数据显示时间连续性一样。 空间数据的一些形式也有一个时间分量给他们，这需要时空离群[测定141，142]。 离散序列中的异常值检测与连续序列中的时间异常值检测问题有关。 对于离散序列，可以在[126]中找到一个很好的调查。 在各种类型的时间数据的异常检测一般调查可能在[发现231，232]。

用于与图形不寻常行为附近音响nding节点离群方法[提出41]，以及用于连接nding关系的异常值，子图异常值和社区离群技术在[提出17，214，416，452]。 所有这些方法的主要思想是网络中的异常区域是由边缘，子图和社区形式的异常关系引起的。 在颞网络进化网络分析的问题[进行了研究20，233，234，519]。 静态和动态网络异常检测的调查可以在[14，43，457]。

最近，已经提出了用于异常集合的方法。 [344]中的工作设计了在异常值检测方法中使用不同特征子集的方法，并将它们组合以提供更有效的结果。 在[工作402，403，404]显示了如何以提供一单向连接ED和更健壮的结果从由孤立点检测算法发现二FF erent子空间的得分相结合。 [367]中的工作提出了概念

singola@gmx.com

隔离森林是分类中随机森林成功概念的类似物。 最近，音响场已经通过定位现有的（正规）工作在离群合奏的音响场，并且还建立理论基础[形式化31，32，35]。

已经以罕见类检测的形式广泛研究了离群检测问题的监督版本。 为受监管的情况下，读者可以参考上CLASSI音响阳离子[A gen-全部擦除书176]，因为该问题本质上是一个成本敏感的变化就[132，182]上的标准CLASSI音响阳离子问题，其中，所述类分布非常不平衡。 特别地，读者可以参考[132，182]对成本敏感的学习从不平衡数据集的基础进行彻底解散cussion。 在[183]中讨论了许多用于从正数和未标记数据分类的方法]，本文中的参考文献也可以找到对该领域先前工作的一个很好的回顾。 在[工作431，618，619]第一个显示的人力监督是如何被用来显着地改善异常检测的电子FF ectiveness。 最后，新颖性检测的半监督方案已被广泛于[讨论388，389，538]。

异常值分析中的评估方法与信息检索，推荐系统和（监督）稀有学习中使用的技术相同。 实际上，最近的推荐系统书[34]的第7章中讨论的大多数评估方法也可用于异常值分析。 有关ROC曲线的详细讨论可以在[192]中找到。 虽然ROC和PR曲线是用于异常值评估的传统方法，但最近已经注意到[402]这些方法可能不一定提供不同类型分析所需的所有见解。 因此，工作在[402]已经提出了基于最佳可能排名和算法确定的排名之间的Spearman相关性的系数。

1.10 演习

1. 以下各点中的以下哪一点是异常值？为什么？

   - (1-dimensional) { 1, 3, 2, 1, 3, 2, 75, 1, 3, 2, 2, 1, 2, 3, 2, 1 }
   - (1-dimensional) { 1, 2, 3, 4, 2, 19, 9, 21, 20, 22 }
   - (2-dimensional) { (1, 9), (2, 9), (3, 9), (10, 10), (10, 3), (9, 1), (10, 2) }

2. 使用MATLAB或任何其他数学软件在练习1的不同情况下创建沿每个维度的数据分布的直方图。 您能直观地看到异常值吗？哪个？在哪种情况下异常值不清楚，为什么？

3. 对于练习1的二维情况，在二维平面上绘制数据点。 你能直观地看到异常值吗？哪个？

4. 对练习1中的每个案例应用Z值测试。 对于二维案例，将Z值测试应用于各个维度。 你发现了正确的异常值吗？

5. 对于练习1中的二维情况，构造函数$f(x_1, x_2) = |x_1 - x_2|$。将Z值验证应用于每个数据点上的$f(x_1, x_2)$。你获得了吗？
   如练习3中的视觉分析所示，纠正异常值？为什么？

singola@gmx.com

6. 确定练习1中的每个数据点的最近邻居。 哪些数据点具有最近邻居距离的最大值？它们是正确的异常值吗？

7. 对练习1中的每个案例应用k-means聚类算法，同时设置 k = 2.哪个数据点距离这两个方法最远？这些是正确的异常值吗？

8 考虑以下时间序列：

- 1, 2, 3, 3, 2, 1, 73, 1, 2, 3, 5
- 1, 2, 3, 4, 3, 2, 1, 3, 73, 72, 74, 73, 74, 1, 2, 3, 4, 2
- 1, 2, 3, 5, 6, 19, 11, 15, 17, 2, 17, 19 , 17, 18

您会考虑哪些数据点异常值？时间成分如何影响您选择的异常值？现在检查时间序列发生重大变化的点？这些点如何与异常值相关联？

9. 考虑从1到8索引的N个8个节点的无向网络tt =（N，A）。 让边集A 为（1,2），（1,3），（1,4），（1,5） }，（1,6），（1,7），（1,8）。 在纸上绘制网络{以使其可视化。 是否有任何节点，你会认为这是一个异常值？为什么？

- 现在删除边缘（1，7）。 这是否会改变您认为是异常值的节点集？为什么？

10. 考虑从1到8索引的N个8个节点的无向网络tt =（N，A）。 让边集A 为（1,2），（1,3），（1,4），（2,3） }，（2,4），（5,7），（4,7），（5,6），（6,8），（5,8），（6,7）。{在纸上绘制网络以使其可视化。 有没有边缘，你会认为是异常值？为什么？

11. 考虑三种算法A，B 和C，它们在具有100个点和5个异常值的数据集上运行。 三种算法的分数的异常值等级如下：A：1,3,5,8,11
B: 2, 5, 6, 7, 9
C: 2, 4, 6, 10, 13
绘制每个算法的PR曲线。 您是否会认为任何算法都严格优于其他任何算法？为什么？

singola@gmx.com

第2章

异常检测的概率统计模型

> “有了四个参数，我就可以找到一头大象，而且我可以让他摆动他的行李箱。
> ”- 约翰·冯·诺伊曼

2.1 介绍

最早的离群检测方法植根于概率和统计模型，可追溯到19世纪[180]。 这些方法是在计算机技术出现和普及之前提出的，因此设计时没有太多关注数据表示或计算效率等实际问题。 然而，基础数学模型非常有用，并且最终适用于各种计算场景。

异常值分析中流行的统计建模形式是检测极端单变量值。 在这种情况下，希望确定单变量分布的尾部的数据值以及相应的统计显着性水平。 尽管极端单变量值属于非常特殊的异常值类别，但它们有许多应用。 例如，几乎所有异常检测算法都使用数字分数来测量数据点的异常，这些算法的最后一步是确定这些分数的极值。 统计上显着的极值的识别有助于将异常值得分转换为二进制标记。 由不同类算法使用的异常值评分机制的一些示例如下：

- 在概率建模中，数据指向生成模型的可能性是异常值。
- 在基于邻近度的建模中，k最近邻距离，距离最近聚类质心的距离或局部密度值是异常值得分。
- 在线性建模中，数据点与数据的低维表示的剩余距离是异常值。

© Springer International Publishing AG 2017

C.C. Aggarwal, *Outlier Analysis*, DOI 10.1007/978-3-319-47578-3_2

singola@gmx.com

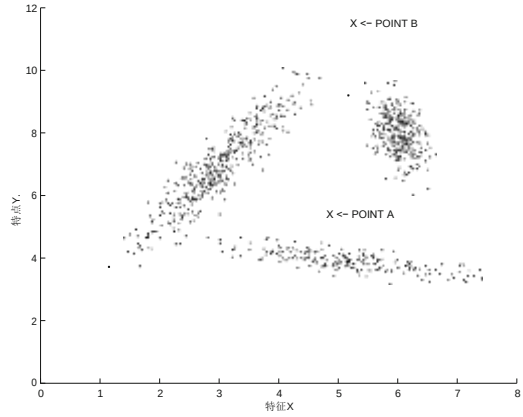


图2.1：多元极值和异常值之间的区别示例

- 在时间建模中，数据点与其预测值的偏差用于创建离群值得分。

因此，即使不能对原始数据进行极值建模，从一组离群值得分中有效确定极值的能力也构成了所有离群值检测算法的基石，作为最后一步。 因此，本章将广泛研究极值建模问题。

极值建模也可以轻松扩展到多变量数据。 位于数据的帕累托极值上的数据点被称为多变量极值。 例如图2.1，数据点'B'是多变量极值。 另一方面，数据点'A'是异常值，但不是多变量极值。 多变量极值分析方法有时也用于一般异常值分析。 这些技术有时在现实世界的异常分析应用中表现出色，但它们并不是一般的异常分析方法。 这种行为的原因主要在于这样一个事实，即现实世界的特征提取方法有时会创建表示异常值是由极值引起的表示。 例如，在信用卡欺诈检测应用程序中，通常提取与交易的大小和频率相对应的特征。 异常大或频繁的交易通常对应于异常值。 即使以这种方式提取特征子集，它也可以极大地提高多变量极值分析方法对异常值检测的有效性。 在一般情况下使用这种方法的缺点是数据点如图中的“A”这种方法错过了2.1 。 然而，尽管存在这种明显的缺点，但在实际应用中不应忽视这些方法。 在许多情况下，可以将这些技术添加为集合方法的一个或多个组件（参见第6章），以提高其准确性。 也可以使用概率建模来发现超出极值的一般异常值。 例如，在图2.1中，可以将数据集建模为三个高斯分量的混合，因此发现异常值'A'和'B'。 混合模型可以被认为是聚类算法的概率版本，可以将异常值发现为副产品。 这些方法的一个显着优点是，一旦确定了数据的生成模型，它们很容易推广到不同的数据格式甚至混合属性类型。 大多数概率模型假设每种混合分量（例如，高斯分布）的基础分布的特定形式，以模拟数据点的正常模式。 随后，学习该模型的参数以便观察

singola@gmx.com

数据具有由模型生成的最大可能性[164]。 因此，该模型是数据的生成模型，并且可以从该模型估计生成特定数据点的概率。 由模型生成的概率异常低的数据点被标识为异常值。混合模型是多元极值分析的自然概括; 例如，如果我们将混合物建模为包含单个高斯分量，则该方法专门研究一种最着名的多变量极值分析方法（参见2.3.4节中的马哈拉诺比斯方法）。

本章安排如下。 下一节将讨论单变量极值分析的统计模型。 第2.3节讨论了多变量数据中极值分析的方法。 2.4 节讨论了异常值的概率建模方法。 2.5 节讨论了异常值分析的概率模型的局限性。 第2.6 节介绍了结论和总结。

2.2 极值分析的统计方法

在本节中，我们将介绍单变量数据分布中极值分析的概率和统计方法。 概率分布中的极值统称为分布尾。 极值分析的统计方法量化了分布尾部的概率。 显然，尾部的概率值非常低，表明其中的数据值应该被认为是异常的。 在实际分布不可用的情况下，许多尾部不等式限制了这些概率。

2.2.1概率尾部不等式

可以使用尾部不等式来约束概率分布尾部的值应被视为异常的概率。 尾部不等式的强度取决于对基础随机变量的假设数量。 较少的假设会导致较弱的不等式，但这种不等式适用于较大类别的随机变量。 例如，Markov和Chebychev不等式是弱不等式，但它们适用于非常大类的随机变量。 另一方面，Cherno ff bound和Hoe ff ding不等式都是更强的不等式，但它们适用于受限类别的随机变量。

马尔可夫不等式是最基本的尾部不等式之一，它定义为仅采用非负值的分布。 令X为随机变量，概率分布为fX （x），均值为E[X]，方差为V ar[X]。

定理2.2.1（马尔可夫不等式）设X 是一个随机变量，只接受非负随机值。 然后，对于任何常数α 满足ë[X] <α，以下为真：

$$P(X > a) \leq E[X]/a \tag{2.1}$$

证明：设fX （x）表示随机变量X的密度函数。 然后，我们有：

$$E[X] = \int x \cdot f_X(x) \cdot \mathrm{d}x = \int_{0 \leq x \leq a} x f_X(x)\, \mathrm{d}x + \int_{x>a} x \cdot f_X(x) \cdot \mathrm{d}x$$
$$\geq \int_{x>a} x \cdot f_X(x) \cdot \mathrm{d}x \geq \int_{x>a} a \cdot f_X(x) \cdot \mathrm{d}x$$

singola@gmx.com

第一个不等式来自x的非负性，第二个不等式来自于积分仅在x > a的情况下定义的事实。此外，最后一个方程右边的项恰好等于aP （X>a）。 因此，以下情况属实：

$$E[X] \geq a \cdot P(X > a) \tag{2.2}$$

可以重新安排上述不等式以获得最终结果。 ■

马尔可夫不等式仅针对非负值的概率分布而定义，并且仅在上尾提供界限。 在实践中，通常希望绑定任意分布的两个尾部。 考虑X 是任意随机变量的情况，其不一定是非负的。 在这种情况下，不能直接使用马尔可夫不等式。 然而，（相关的）切比丘夫不等式在这种情况下非常有用。 Chebychev不等式是马尔可夫不等式直接应用于随机变量X的非负导数：

定理2.2.2（Chebychev不等式）设X 是任意随机变量。 然后，对于任何常数a，以下情况为真：

$$P(|X - E[X]| > a) \leq Var[X]/a^2 \tag{2.3}$$

证明：当且仅当（XE[X]）2 >a2时，不等式 XE[X] >a 为真。 通过将Y =（XE[X]）2定义为来自X的（非负）导数随机变量，很容易看出E[Y] = V ar[X]。 然后，定理陈述左侧的表达式与确定概率P （Y>a2）相同。 通过将马尔可夫不等式应用于随机变量Y.，人们可以获得所需的结果。

上述证明中使用的主要技巧是将马尔可夫不等式应用于随机变量的非负函数。 当X 的分布时，该技术通常对于证明其他类型的边界非常有用具有特定形式（例如伯努利随机变量的总和）。 在这种情况下，可以使用随机变量的参数化函数以获得参数化边界。 然后可以针对尽可能紧密的界限优化基础参数。 通过使用这种方法，可以推导出几个众所周知的界限，如Cherno ff界限和Hoe ff ding不等式。 马尔可夫和切比切夫不等式是相对较弱的不等式，并且通常不能提供足够严格的界限以在许多实际情况中有用。 这是因为这些不等式不对随机变量X的性质做出任何假设。 然而，当对随机变量使用更强的假设时，可以捕获许多实际情况。 在这种情况下，尾部分布的边界可能更紧密。 特定情况是随机变量X 可以表示为其他独立有界随机变量之和的情况。 ■

2.2.1.1 有界随机变量的和

许多以聚合形式定义的实际观察可以表示为有界随机变量的总和。 这种情况的一些例子如下：

例2.2.1（体育统计） 美国国家篮球协会（NBA）选秀小组可以获得不同候选球员的大学篮球统计数据。 对于每个玩家和每个游戏，一组定量值描述了他们在不同游戏中的各种得分统计数据。 例如，这些定量值可以对应于数字

singola@gmx.com

扣篮，助攻，篮板等等。 对于特定的统计数据，任何玩家的总体表现可以表示为他们在N种不同游戏中的统计数据的总和：

$$X = \sum_{i=1}^{N} X_i$$

Xi的所有值都在[l，u] 范围内。 玩家在不同游戏中的表现被假定为彼此独立。 习近平对所有参与者所代表的统计数据的长期全局均值为μ。 NBA选秀球队希望根据每个统计数据确定异常球员。

在此示例中，聚合统计量表示为有界随机变量的总和。 可以使用Hoe ff ding不等式来量化相应的尾部边界。

在许多情况下，聚合中的各个随机变量组件不仅是有界的，而且是二进制的。 因此，总统计量可以表示为伯努利随机变量的总和。

例2.2.2（杂货店购物）杂货店跟踪客户的数量（来自其频繁的购买者计划），他们在特定的一天经常光顾商店。 任何客户的长期概率我参加店在某一天被称为是 圆周率。 众所周知，不同客户的行为是彼此独立的。 在某一天，评估商店收到的概率超过η （频繁购买计划）的客户。

在第二个例子中，顾客数量可以表示为独立伯努利随机变量的总和。 相应的尾部分布可以用Cherno ff界限表示。 最后，我们提供了一种非常常见的聚合异常检测应用，即制造中的故障诊断。

例2.2.3（制造质量控制）公司使用制造装配线来生产产品，其中可能存在具有预定义（低）概率p的故障。 质量控制过程从装配线中采样N个产品，并仔细检查它们以计算有缺陷的产品数量。 对于给定数量的故障产品，评估装配线异常行为的概率。

样本大小N 通常很大，因此，可以使用中心极限定理来假设样本是正态分布的。 根据该定理，大量独立和相同的正态分布的总和收敛于正态分布。

本节将正式介绍不同类型的边界和近似值。 首先将讨论Cherno ff界限和Hoe ff ding不等式。 由于下尾部和上部尾部的表达略有不同，因此它们将单独处理。 下面介绍了Cherno ff下限。

定理2.2.3（下尾Cherno ff界）设X 是随机变量，可以表示为N个独立二元（伯努利）随机变量的总和，每个随机变量的概率为pi，取值为1 。

$$X = \sum_{i=1}^{N} X_i$$

singola@gmx.com

然后，对于任何δ ＆Element;（0，1），就可以显示以下内容：

$$P(X < (1-\delta)\cdot E[X]) < e^{-E[X]\cdot\delta^2/2} \tag{2.4}$$

其中e 是自然对数的基础。

证明：第一步是显示以下不等式：

$$P(X < (1-\delta)\cdot E[X]) < \frac{e^{-\delta}}{(1-\delta)^{(1-\delta)}}^{\Sigma_{E[X]}} \tag{2.5}$$

引入未知参数t > 0以创建参数化边界。 的尾下不等式X 被转换成上部尾不等式的幂
表达式$e^{-t\cdot X}$。 该随机表达式可以由马尔可夫不等式约束，并且它提供作为t的函数的约束。 可以优化t的这个函数，以便获得
最严格的约束。 通过在指数形式上使用马尔可夫不等式，可以得出以下结果：

$$P(X < (1-\delta)\cdot E[X]) \le \frac{E[e^{-t\cdot X}]}{e^{-t\cdot(1-\delta)\cdot E[X]}}$$

通过扩大$X = \Sigma^{\tilde{N}}_{i=1} X_i$ 在指数中，可以得到以下结果：

$$P(X < (1-\delta)\cdot E[X]) \le \frac{{}_i E[e^{-t\cdot X_i}]}{e^{-t\cdot(1-\delta)\cdot E[X]}} \tag{2.6}$$

上述简化使用了这样的事实：独立变量乘积的期望等于期望的乘积。 由于每个Xi 都是伯努利，因此可以显示以下内容：

$$E[e^{-t\cdot X_i}] = 1 + E[X_i]\cdot(e^{-t}-1) < e^{E[X_i]\cdot(e^{-t}-1)}$$

第二个不等式来自$eE^{[X_i]\cdot(e^{-t}-1)}$的多项式展开。 通过将该不等式代入等式2.6，并使用$E[X] = \Sigma_i E[X_i]$，可以获得以下结果：

$$P(X < (1-\delta)\cdot E[X]) \le \frac{e^{E[X]\cdot(e^{-t}-1)}}{e^{-t\cdot(1-\delta)\cdot E[X]}}$$

对于t> 0的任何值，右边的表达式都是正确的。 期望确定提供最紧密可能界限的t 的值。
这样的t 值可以通过获得
计算表达式相对于t 的导数并将其设置为0.可以证明，此优化过程的结果值$t = t^*$如下：

$$t^* = \ln(1/(1-\delta)) \tag{2.7}$$

通过在上述不等式中使用该t^* 值，可以证明它等效于等式2.5。 这样就完成了证明的第一部分。

在对数项的泰勒展开的第一个两个术语（1 - δ）·LN（1 - δ）可扩展到表明：（1 δ）（$1-\delta$）>电子$^{-\delta+\delta^2/2}$。 通过在等式2.5的分母中代入该不等式，获得期望的结果。

可以获得类似于上尾Cherno ff界限的结果，尽管形式略有不同。 ■

singola@gmx.com

定理2.2.4（上尾Cherno ff界）设X 是随机变量，它表示为N个独立二元（伯努利）随机变量的总和，每个变量的概率为pi，取值为1 。

$$X = \sum_{i=1}^{N} X_i$$

然后，对于任何δ ＆Element;（0，2·ë - 1）中，以下等式成立：

$$P(X > (1+\delta)\cdot E[X]) < e^{-E[X]\cdot\delta^2/4} \tag{2.8}$$

其中e 是自然对数的基础。

证明：第一步是显示以下不等式：

$$P(X > (1+\delta)\cdot E[X]) < \frac{e^{\delta}}{(1+\delta)^{(1+\delta)}}^{\Sigma_{E[X]}} \tag{2.9}$$

和以前一样，这可以通过引入未知参数来完成T> 0，以及将所述上部尾不等式上X 成上等'X 。 这可以通过马尔可夫不等式作为t的函数来限制。 t的这个功能可以优化，以获得
最严格的约束。

它可以通过代数简化的阳离子，在等式不等式被进一步示出2.9 提供的所有值所需的结果δ ∈（0，.2_ë 1）。 ■

接下来，将介绍Hoe ff ding不等式。 Hoe ff ding不等式是比Cherno ff约束更普遍的尾部不等式，因为它不需要从伯努利分布中提取基础数据值。 在这种情况下，需要从有界区间[li，ui] 绘制第i个数据值。 相应的概率界限用参数li 和ui表示。 因此，Cherno ff界限的场景是Hoe ff ding不等式的特例。 我们在下面陈述了Hoe ff ding不等式，其中上下尾部不等式都是相同的。

定理2.2.5（Hoe ff ding Inequality）设X 是一个随机变量，可以表示为N个独立随机变量的总和，每个变量的范围都在[li，ui]范围内。

$$X = \sum_{i=1}^{N} X_i$$

然后，对于任何θ> 0，可以显示以下内容：

$$P\left(X - E[X] > \theta\right) \le e^{-\frac{\Sigma 2\cdot\theta^2}{\sum_{i=1}^{N}(u_i-l_i)^2}} \tag{2.10}$$

$$P\left(E[X] - X > \theta\right) \le e^{-\frac{2\cdot\theta^2}{\Sigma_{i=1}^{N}(u_i-l_i)^2}} \tag{2.11}$$

证明：这里将描述上尾部分的证明。 下尾不等式的证明是相同的。 对于任何选择参数t≥0 ，以下情况均属实：

$$P(X - E[X] > \theta) = P(e^{t\cdot(X-E[X])} > e^{t\cdot\theta}) \tag{2.12}$$

马尔可夫不等式可用于表明右手概率最多

$E[e^{(X-E[X])}\cdot e^{-t\cdot\theta}$. The expression within $E[e^{(X-E[X])}]$ can be expanded in terms of the

singola@gmx.com

表2.1：用于绑定尾概率的不同方法的比较

结果	脚本	强度
Chebychev	任何随机变量	弱
马尔科夫	非负随机变量	弱
Hoeffding	独立有界的总和 随机变量	强（指数 减少样品）
Chernoff	iid伯努利的总和 随机变量	强（指数 减少样品）
CLT	**许多**iid变量的总和	几乎完全一样
广义CLT	**很多**独立的总和 和有界变量	几乎完全一样

个别成分Xi。 由于产品的期望等于独立随机变量预期的乘积，因此可以显示以下内容：

$$P(X - E[X] > \theta) \leq e^{-t\theta} \cdot \prod_i E[e^{t(X_i - E[X_i])}] \tag{2.13}$$

关键是要表明的值 ë[等'（X我-ë[X我]）]是至多等于 等2·（û我-升我）2/8。 这可以通过使用一个参数来证明，该参数使用指数函数et（（Xi-E[Xi]） 的凸性结合泰勒定理（参见练习12）。

因此，以下情况属实：

$$P(X - E[X] > \theta) \leq e^{-t\theta} \cdot \prod_i e^{t^2 \cdot (u_i - l_i)^2/8} \tag{2.14}$$

这种不等式适用于t的任何正值。 因此，为了找到最严格的界限，需要确定使上述等式的右侧最小化的t 的值。 t = t*的最佳值可以显示如下：

$$t^* = \frac{4 \cdot \theta}{\sum_{i=1}^{N} (u_i - l_i)^2} \tag{2.15}$$

通过代入t = t* 的值，可以获得期望的结果。 可以通过将上述步骤应用于P （E[X] -X>θ）而不是P （X - E[X] >θ）来导出下尾界。

因此，不同的不等式可能适用于不同的一般性情景，也可能适用有不同程度的力量。 表2.1列出了这些不同的情况。 ■

一个有趣的观察结果是Hoe ff ding尾部边界以θ2呈指数衰减，这正是正态分布的行为方式。 这并不奇怪，因为根据中心极限定理（CLT），大量独立有界随机变量的总和收敛于正态分布。 这种收敛是有用的，因为精确分布（或近似近似）提供的界限比任何上述尾部不等式更紧密。

定理2.2.6（中心极限定理）具有平均μ 和标准√偏差σ的大量N 个独立和相同分布随机变量的总和
收敛到正态分布，平均μ · N 和标准差σ · N 。

singola@gmx.com

CLT的更一般化形式也可以应用于独立变量的总和（不一定相同），其中变量在基础力矩测量方面是充分有界的。 CLT的这种概括的一个例子是Lyapunov CLT [88]。 基本思想是，大量独立（但不是相同分布）的随机变量之和的均值和方差可以分别用均值和方差的相应和来近似。 对于持有的条件，也对基础分布作出一些弱假设。 请参阅书目说明。

2.2.2 Statistical-Tail Confidence Tests

正态分布具有许多应用，例如统计尾部信息测试。 在统计尾部置信度测试中，识别出根据正态分布分布的一组数据值的极值。 正态分布的假设在实际领域中是普遍存在的。 这不仅适用于表示为随机样本总和的变量（如前一节所述），而是由不同随机过程生成的许多变量。 具有平均μ 和标准偏差σ 的正态分布的密度函数fX （x）定义如下：

$$f_X(x) = \frac{1}{\sigma \cdot \sqrt{2 \cdot \pi}} \cdot e^{\frac{-(x-\mu)^2}{2 \cdot \sigma^2}} \tag{2.16}$$

在某些设置中，假设建模分布的平均μ 和标准偏差σ 是已知的是合适的。 当大量数据样本可用于精确估计μ 和σ时就是这种情况。 在其他情况下，μ 和σ 可能来自领域知识。 然后，观测值xi 的Z值zi 可以如下计算：

$$z_i = (x_i - \mu)/\sigma \tag{2.17}$$

由于正态分布可以直接表示为Z值的函数（并且没有其他参数），因此点xi 的尾概率也可以表示为zi的函数。 实际上，Z值对应于缩放和平移的正态随机变量，也称为标准正态分布，均值为0和方差
因此，可以直接使用累积标准正态分布，以确定zi值处尾部概率的精确值。 从实际角度来看，由于这种分布不是以封闭形式提供的，因此使用正态分布表来将zi的不同值映射到概率。 这提供了显着性的统计水平，其可以直接解释为数据点是异常值的概率。 基本假设是数据是由正态分布生成的。

2.2.2.1 t-值测试

上述讨论假设可以从大量样本中非常精确地估计模型分布的均值和标准偏差。 但是，实际上，可用的数据集可能很小。 例如，对于具有20个数据点的样本，准确地模拟平均值和标准偏差要困难得多。 在这种情况下，我们如何准确地执行统计显着性测试？

学生的t分布提供了一种有效的方法来模拟这种情景中的异常。 这种分布由称为自由度ν的参数定义，该参数 由可用的样本大小密切定义。 所述吨-配送近似于

singola@gmx.com

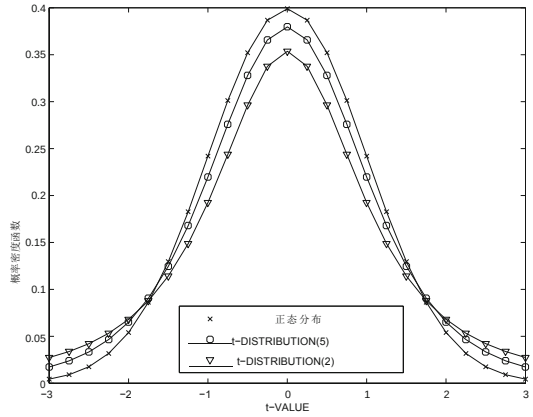


图2.2：不同自由度数的t分布（对应于不同的样本大小）

对于较大的自由度（> 1000），正态分布非常好，并且在它到达的极限内收敛到正态分布。 对于较小的自由度（或样本大小），t-分布具有与正态分布类似的钟形曲线，除了它具有更重的尾部。 这是非常直观的，因为较重的尾部导致由于无法从较少的样本精确估计建模（正态）分布的平均值和标准偏差而导致的统计显着性损失。

该吨-配送表示为几个独立同分布的标准正态分布的函数。 它有一个与数字对应的参数ν
自由度。 这规定了这种正常分布的数量
它表达了什么。 参数ν 设置为N -1，其中N 是可用样本的总数。 设U0 ...Uν 为ν + 1，独立且相同分布的正态
零均值和单位标准差的分布。 这种正态分布也称为标准正态分布。 然后，t分布定义如下：

$$T(\upsilon) = \frac{U_0}{\sqrt{\sum_{i=1}^{\upsilon} U_i^2)/\upsilon}} \tag{2.18}$$

使用t分布的直觉是分母明确地模拟估计基础正态分布的标准差的随机性
仅使用少量独立样本。 术语＆Sigma;ν
分母是具有参数ν的 x2分布 ，以及整个（缩放的）分母 ù 2中的 $\sum_{i=1}^{i}$
当ν $\Rightarrow \infty$ 时，收敛到1 。 因此，在极限情况下，当大量的时候
样本可用，分母贡献的随机性消失，t分布收敛到正态分布。 对于较小的ν （或样本大小）值，此分布具有较重的尾部。 图2.2中提供了不同ν 值的t分布示例。 显然，具有较少自由度的t分布具有较重的尾部。

使用少量样本x1 ... xN 进行极值检测的过程如下所示。 首先，估计样品的平均值和标准偏差。 然后，这用于直接从样本计算每个数据点的t值。 该ẗ-值
以与Z值相同的方式计算。 每个数据点的尾概率是根据t分布的累积密度函数计算的（N -1）
-degrees of

singola@gmx.com

自由。 与正态分布的情况一样，标准化表格可用于此目的。 从实际角度来看，如果有超过1000个样本可用，那么t分布（具有至少1000个自由度）非常接近正态分布，可以使用正态分布作为非常好的近似。

2.2.2.2 偏差平方和

异常值检测的一个常见情况是需要将沿着独立标准的偏差统一为单个异常值得分。 这些偏差中的每一个通常被建模为来自独立且相同分布的标准正态分布的Z值。 然后将总偏差度量计算为这些值的平方和。 对于d维数据集，这是具有d个自由度的x2分布。 一个具有d 自由度的x2分布被定义为d 独立的平方和
凹陷标准正态随机变量。 换句话说，考虑变量V ，即
表示为独立同分布的标准正态随机变量的平方和滋∽Ñ （0，1）：

$$V = \sum_{i=1}^{d} i^2$$

然后，V 是从具有d个自由度的x2分布绘制的随机变量。

$$V \sim x^2(d)$$

虽然这里省略了对x2分布特征的详细讨论，但其累积分布不能以封闭形式获得，但需要进行计算评估。 从实际角度来看，累积概率表通常是可用的
-
能够进行建模。 然后可以使用x2分布的累积概率表来确定该聚合偏差值的概率的概率水平。 当偏差被建模为时，这种方法特别有用
在统计上彼此独立。 正如我们将在第3章中看到的那样，这种情况可能出现在主成分分析等模型中，其中不同组件的误差通常被建模为独立的正态随机变量。

2.2.2.3 用箱形图可视化极值

可视化单变量极值的一种有趣方法是使用箱形图或箱形和晶须图。 这种方法在可视化异常值分数的背景下特别有用。 在箱形图中，单变量分布的统计数据按照五个数量进行汇总。 这五个量是“最小/最大”（胡须），上下四分位数（方框）和中位数（方框中间的线）。 我们在其中两个数量附近引用了引用，因为它们是以非标准方式定义的。 上四分位数和下四分位数之间的距离称为四分位数间距（IQR）。 “最小”和“最大”以（非标准）修剪方式定义，以便定义晶须的位置。 如果没有超过1的积分。 5 IQR高于最高四分位值（方框的上端），然后上方的晶须是真正的最大值。 否则，上部晶须设置为来自盒子上端的IQR的1.5倍。 一个完全类似的规则适用于较低的晶须，它从盒子的下端设置为1.5 IQR。 在正态分布数据的特殊情况下，比最高四分位数高1.5 IQR的值对应于标准差的2.7倍的距离

singola@gmx.com

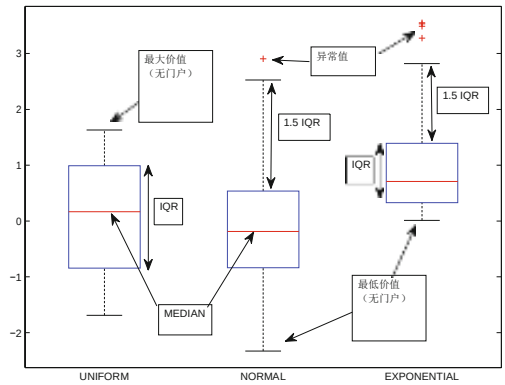


图2.3：用箱形图显示单变量极值

（从平均值）。 因此，晶须大致放置在类似于3位置σ
cut-o ff指向正态分布。

箱形图的一个例子如图2.3所示。 在这种情况下，我们已经显示了对应于（i）具有零均值和单位方差的均匀分布，（ii）标准正态分布，以及（iii）具有单位平均值的指数分布中的每一个的100个数据点。 请注意，前两个分布关于均值是对称的，而最后一个不是。 相应的箱形图如图2.3所示。 在每种情况下，上部和
框的下端表示[1]倍的上部和下部的四分位数。 在均匀分布的情况下，没有异常值，因此，上下胡须代表了
真实的最大值和最小值。 另一方面，在正常和指数分布的情况下，在上端存在异常值。 因此，在每种情况下，须将晶须置于盒子上端的1.5 IQR以上。

关于胡须的放置存在许多其他惯例，例如使用实际的最小值/最大值或使用特定百分位的数据分布。 本书中使用的特定惯例称为Tukey箱图。 除了可视化极值之外，这种类型的图对于可视化随机异常值检测算法的性能非常有用，并且通常用于异常值集合分析。 我们将在第6章第6.4 节重新讨论这个问题。

2.3 多元数据的极值分析

极值分析也可以以多种方式应用于多变量数据。 这些定义中的一些尝试明确地模拟基础分布，而其他定义基于更一般的统计分析，其不假设基础数据的任何特定统计分布。 在本节中，我们将讨论四种不同类型的方法，这些方法旨在在多变量数据的边界处找到数据点。这些类方法中的第一类（基于深度）不是统计或概率方法。 相反，它基于点几何的凸包分析。 但是，我们已经将它包含在本章中，因为它根据其发现的异常值的类型自然地与其他多变量极值方法相关。

[1]虽然使用四分位数无处不在，但可以改变方框的百分位数。

singola@gmx.com

算法FindDepthOutliers（数据集：，分数阈值：r）；
开始
$k = 1$;
重复
找出凸壳的角 D
集S ;将深度k 分配给S中的点;
$D = D - S$;
$k = k + 1$;
直到（D 为空）；
报告深度最多为r的点作为异常值;
结束

图2.4：用于发现基于深度的异常值的伪代码

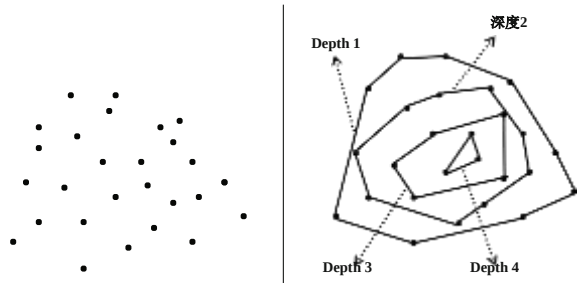
虽然本节中讨论的方法在发现数据空间外边界的异常值方面是有效的，但它们并不擅长在数据空间的内部区域内找到异常值。 对于图2.7中所示的情况，这些方法可以有效地找到异常值，但不是图2.1中所示的异常值“A”。 。 然而，确定这些异常值在许多特殊情况下都是有用的。 例如，在多个偏差值可能与记录相关联的情况下，多变量极值分析可能是有用的。 考虑一种天气应用，其中在不同的空间位置测量诸如温度和压力的多个属性，并且计算作为中间步骤的与预期值的局部空间偏差。 这些与不同属性的预期值的偏差可能需要转换为单个有意义的异常值得分。 第11章的第11.2.1.3 节说明了一个例子，其中计算空间数据的不同测量值的偏差。 通常，这些方法可用于后处理离群值得分的多维向量，其中每个离群值得分是使用不同且可能独立的标准导出的。 正如第1章所讨论的那样，将极值分析的方法与根据生成概率定义的一般离群值分析方法混淆是特别常见的。 但是，区分这两者很重要，因为使用这两种方法的特定应用场景是非常不同的。

2.3.1 基于深度的方法

在基于深度的方法中，使用凸包分析以找出异常值。 这个想法是数据外边界中的点位于凸包的角落。 这些点更可能是异常值。 基于深度的算法以迭代方式进行。 在第k次迭代中，从数据集中移除数据集的凸包角处的所有点。 这些点的深度为k。 重复这些步骤，直到数据集为空。 所有深度最多为r的点都被报告为异常值。 或者，可以将数据点的深度直接报告为异常值分数。 基于深度的方法的步骤如图2.4所示。

该算法也在图2.5中的样本数据集上进行了图解说明。 许多电子FFI cient方法音响nding基于深度的离群值已在[已经讨论295， 468]。 凸壳方法的计算复杂性随维度呈指数增长。 此外，随着维数的增加，更大比例的数据点位于

singola@gmx.com



(a) 数据点 （b）凸壳图2.5：

基于深度的异常值检测

凸壳的角落。 这是因为凸包角落处的点数可以与数据维度呈指数相关。 因此，这些方法不仅在计算上是不切实际的，而且由于异常值得分的增加，因此在更高维度上也越来越有效。 基于深度的方法通常与本章讨论的大多数概率和统计模型完全不同。 事实上，它们不能真正被视为概率或统计方法。 然而，由于它们与其他多变量极值方法的关系，它们在这里被呈现。 尽管在方法上不同，但这些方法具有许多共同的共同特征。 例如，它们仅在异常值位于数据空间边界的情况下才能正常工作，

2.3.2 基于偏差的方法

基于偏差的方法衡量异常值对数据方差的影响。 例如，[62]中提出的方法试图测量当删除特定数据点时底层数据的方差减少了多少。 由于基本假设是异常值位于数据的边界，因此预计删除这些数据点将显着减少方差。 这本质上是一种信息理论方法，因为它检查了数据点被删除时复杂性的降低。 相应地，一组数据点R 的平滑因子定义如下：

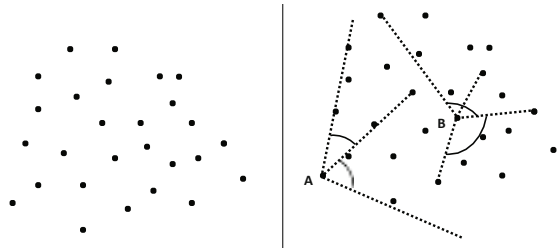
Deftnition 2.3.1 平滑因子SF （[R为一组）- [R 是在数据集的方差，当在设定点的还原- [R 被从数据中去除。

异常值被定义为异常集E ，使得它们的移除导致数据方差的最大减少。 换句话说，对于数据点R的任何子集，必须是这样的情况：

$$SF(E) \geq SF(R)$$

如果多个集合的方差减少相同，则较小的集合是首选。 这遵循标准信息理论原理，即在尽可能小的空间内尽可能多地增加数据描述长度的集合。 确定最优集合E 是一个非常困难的问题，因为对于包含N 个点的数据集存在$2^{N种}$可能性。 [62]中的工作使用了一些启发式算法，例如最佳搜索和随机抽样。 这种方法的一个很好的方面是它是独立于分布的，并且可以应用于任何类型的数据集，只要适当的定义

singola@gmx.com



(a) 数据点 （b）角度谱图2.6：基于角度的离群值检测

可以构造平滑因子。 在[62]的原始工作中，这种方法已应用于序列数据的情况。

2.3.3 Angle-Based Outlier Detection

尽管本书已将其重新分类为多变量极值分析方法，但该方法最初被提议作为一种通用异常值分析方法。 基于角度的方法的想法是，数据边界处的数据点可能将整个数据包围在较小的角度内，而内部的点可能在不同的角度处具有围绕它们的数据点。 例如，考虑图2.6中的两个数据点“A”和“B”，其中'A'是异常值，点'B'位于数据的内部。 很明显，所有数据点都位于以“A”为中心的有限角度内。 另一方面，数据点'B'不属于数据内部。 在这种情况下，不同点对之间的角度可以广泛变化。 实际上，数据点与剩余点的隔离程度越大，底层角度可能越小。 因此，具有较小角度光谱的数据点是异常值，而具有较大角度光谱的数据点不是异常值。

考虑三个数据点X，ÿ ，和ž。 然后，矢量Y之间的角度
当X 是异常值时 ，X 和ZX对于Y 和 Z的不同值不会有太大变化 。 此外，角度通过

点之间的距离反向加权。 相应的角度（加权余弦）定义如下：

$$WCos(\overline{Y}\_\ \overline{X}, \overline{Z}\ \_\ \overline{X}) = \frac{<(\overline{Y}-\overline{X}),(\overline{Z}-\overline{X})>}{||\overline{Y}-\overline{X}||^2 \cdot ||\overline{Z}-\overline{X}||^2}$$

在这里，|| · || 2表示$L_{2\text{范数}}$，< · > 表示标量积。 注意，这是加权余弦，因为分母包含$L_{2\text{范数}}$的平方。该
通过距离的反加权进一步减小了离群点的加权角度，这也对角度谱有影响。 然后，通过改变数据点Y 和Z来测量该角度的光谱的方差，同时保持X 的值固定。 相应地，数据点X 的基于角度的离群因子（ABOF） 定义如下：

$$ABOF(\overline{X}) = Var_{\{Y,Z\in D\}} WCos(\overline{Y}-\overline{X}, \overline{Z}-\overline{X})$$

作为异常值的数据点将具有较小的角度谱，因此将具有较低的基于角度的离群因子ABOF（X）的值。

singola@gmx.com

可以以多种方式计算不同数据点的基于角度的离群因子。 天真的方法是选择所有可能的数据点三元组并计算不同向量之间的O（N 3）角。 可以从这些值显式计算ABOF值。 然而，对于非常大的数据集，这种方法可能是不切实际的。 因此，已经提出了许多基于效率的优化。

为了加速该方法，自然的可能性是使用采样来近似基于角度的离群因子的该值。 可以使用k个数据点的样本 来近似数据点X的ABOF 。 一种可能性是使用无偏的样本。 然而，由于基于角度的离群因子通过距离反向加权，因此数据点的最近邻居对基于角度的离群因子具有最大贡献。 因此，k-nearest的邻居X 可以用来ectively近似异常因子得多多个EFF比所有数据点的无偏样品。 它也已在[325因为它们的ABOF的近似值太高，并且它们不可能是异常值，所以可以在近似计算的基础上滤除许多数据点。 仅针对一小组点计算ABOF的精确值，并且将具有最低ABOF值的点报告为异常值。 我们请读者[325]了解这些效率优化的详细信息。

由于距离的反加权，基于角度的离群值分析方法可以被认为是基于距离和基于角度的方法之间的混合。 如前面使用说明性示例所讨论的，后一因素主要被优化以发现数据中的多变量极值。 这些因素中的每一个的精确影响[2] 似乎不容易以统计上稳健的方式量化。 在大多数数据集中
如图2.1所示，异常值不仅存在于数据边界，还存在于数据内部。 与极端值不同，异常值由生成概率定义。 虽然距离因子可以对内部的异常值产生一些影响，但工作主要集中在角度测量的优势上，并在[325]中说明。]与角度因子相比，距离因子的影响程度较小。 这意味着由于较低的角度谱，数据边界上的异常值在总得分方面将非常受青睐。 因此，基于角度的方法以一种不同的方式处理内部和数据边界中具有相似生成概率的异常值，这对于一般异常值分析而言在统计上是不可取的。 具体而言，数据边界处的异常值更可能在异常值得分方面受到青睐。 这些方法可以有效地找到图2.7所示情况的异常值，但是图2.1 中所示的异常值“A”将受到更少的青睐。 因此，尽管这种方法最初是作为一种通用的离群值分析方法提出的，但它已经在本书的多变量极值分析方法一节中进行了分类。

在[325]中已经声称该方法由于其使用角度而不是距离而更适合于高维数据。 然而，早期的研究[455] 已经表明，由于余弦测量中的浓度效应，基于角度的测量不能免受维数诅咒的影响。 这种浓度效应也会影响角度的光谱，即使它们与距离相结合也是如此。 图2.6中角度光谱的变化很容易在二维数据中直观显示，但稀疏效应也会影响更高维度的角度光谱。 如果主要问题，如[325]，成对距离之间缺乏对比，然后使用角度而不是距离来解决这个问题。 在所有距离对相似的设置中，所有三角形都是等边的，因此所有（角度）的角度都是等边的

[2]当随机变量按因子a缩放时，其方差按比例缩放[2]。 但是，这里的缩放不是一个常数因素。

singola@gmx.com

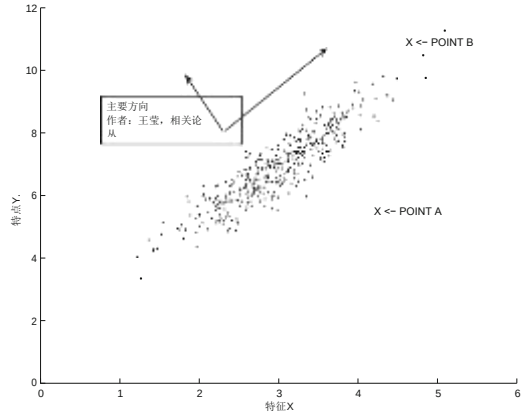


图2.7：具有马哈拉诺比斯距离的多变量数据的极值分析

将收敛到0.5。 实际上，余弦可以表示为欧几里德成对距离的直接函数：

$$\text{Cosine}(\overline{X}, \overline{Y}) = \frac{||X-0||^2 + ||Y-0||^2 - ||X-Y||^2}{2 \cdot ||X-0|| \cdot ||Y-0||} \tag{2.19}$$

如果距离保留关于相对对比的信息很少，则没有理由相信距离的间接函数（如余弦频谱）会做得更好。 为什么角度的频谱应是更健壮到高维度比距离A明确的解释并没有[3] 被在[提供325]。 更重要的是，这些方法没有解决本地无关属性[4]的问题，这是主要的
阻碍有效的离群值分析方法，增加维数。 需要注意的另一个要点是，多变量极值分析比高维度中的一般离群值分析简单得多，因为要探索的数据部分是近似已知的，因此分析是全局的而不是局部的。 过二FF erent尺寸证据可以通过使用一个非常简单的经典距离-分发方法[的累计343，493]。 在下一节中讨论的方法也适用于高维极值分析，因为它以不同的方式隐含地对数据中的全局相关和不相关的方向进行加权，并且在极端的概率可解释性方面具有统计上的合理性。 值。

2.3.4 基于距离分布的技术：马哈拉诺比斯方法

依赖于分布的方法是将整个数据集建模为通常以多元高斯分布的形式分布其均值。 设μ 是

[3]余弦函数在某些高维域（如文本）中的使用在后来的工作中被引用作为一个例子[326]。 在具有小的和变化的非零属性的域中，由于重要的归一化特性，余弦是优选的，而不是因为更大的维度抵抗性。 通过代入公式2.19中的X = Y = 1，很明显，如果所有点被归一化为位于单位球上，则余弦等于欧几里德距离。 即使对于独特的文本结构，余弦函数也不能抵抗维度诅咒[455]。 越来越多的非零属性，对更一般的分布，直接影响数据中心。 ||

singola@gmx.com

d-d d维数据集的平均（行）向量，Σ是其dd 协方差矩阵。 在这种情况下，协方差矩阵的第（i，j）个条目等于维度i 和j之间的协方差。 然后，d维（行向量）数据点X 的概率分布f （X）可以如下定义：

$$f(\overline{X}) = \frac{1}{\sqrt{|\Sigma| \cdot (2 \cdot \pi)^{(d/2)}}} \cdot \exp\left[-\frac{1}{2}(X - \mu)\Sigma^{-1}(X - \mu)^T\right] \quad (2.20)$$

Σ的值表示协方差矩阵的行列式。 我们注意到指数中的项是数据点X 与数据的质心μ 的平方Mahalanobis距离的（一半）。 该术语用作离群值，可直接计算如下：

$$\text{Mahalanobis}(\overline{X}, \overline{\mu}, \Sigma) = \sqrt{(\overline{X} - \overline{\mu})\Sigma^{-1}(\overline{X} - \overline{\mu})^T} \quad (2.21)$$

马哈拉诺比斯距离的计算需要协方差矩阵Σ的反演。 在矩阵Σ不可逆的情况下，可以使用具有dd 单位矩阵I的正则化。 基本思想是用公式2.21中的λ > 0的一些小值将Σ替换为Σ+ λI。 这里，λ > 0表示正则化参数。

一个点的马哈拉诺比斯距离类似于距数据质心的欧几里德距离，不同之处在于它基于属性间相关性对数据进行归一化。 例如，如果要将数据的轴系统旋转到主方向（如图2.7所示），那么数据将没有属性间相关性。 正如我们将在第3章第3.3 节中看到的那样，实际上可以通过使用主成分分析（PCA）来确定d维数据集中的这种相关方向。 马哈拉诺比斯距离简单地等于X 和 μ 之间的欧几里德距离在将每个变换的坐标值除以该方向的标准偏差之后，在这种变换的（轴旋转的）数据集中。 因此，也可以使用主成分分析来计算马哈拉诺比斯距离（见第3章第3.3.1 节）。

这种方法认识到不同的相关方向具有不同的方差，并且应该沿着这些方向以统计标准化的方式处理数据。 例如，在图2.7的情况下，根据数据中的自然相关性，数据点“A”可以更合理地被认为是数据点“B”的异常值。 另一方面，数据点'A'基于欧几里德距离更接近数据的质心（比数据点'B'），但不基于马哈拉诺比斯距离。 有趣的是，数据点'A'似乎也比数据点'B'具有更高的角度谱，至少从平均采样角度来看。 这意味着，至少在角度的主要标准的基础上，基于角度的方法将不正确地支持数据点“B”。 这是因为它无法解释不同方向的相对相关性，这个问题随着维度的增加而变得更加突出。 马哈拉诺比斯方法对增加维数具有鲁棒性，因为它使用协方差矩阵以统计有效的方式总结高维偏差。 值得注意的是，马哈拉诺比斯方法不应仅仅被视为极值方法。 事实上，作为部分3.3.1 表明，其相关敏感特性比其极值特征更强大。

我们还注意到，沿主关联方向的每个距离可以被建模为1维标准正态分布，其大致独立于其他正交相关方向。 如本章前面所讨论的，独立于标准正态分布绘制的d 变量的平方和将导致从具有d 自由度的x2分布绘制的变量。

singola@gmx.com

因此，可以使用x2分布的累积概率分布表来确定具有适当水平的异常值的异常值。

2.3.4.1 马哈拉诺比斯方法的优势

尽管Mahalanobis方法在第一眼看上似乎过于简单，但很容易忽略Mahalanobis方法以优雅的方式解释属性间依赖性的事实，这在高维数据集中变得尤为重要。 在精度，计算复杂性和参数化方面，这种简单的方法比基于距离的更复杂方法具有几个令人惊讶的优势：

1. 将Mahalanobis方法仅作为多变量极值分析方法观察是短视的，因为它的大部分功率在于它使用了属性间相关性。 协方差矩阵的使用确保在异常值检测过程中考虑属性间相关性。 事实上，正如第3章所讨论的，人们可以将Mahalanobis方法视为PCA的软版本。 虽然不是很明显，但是一些复杂的线性模型的优点
   例如，一类支持向量机（SVM）[4] 和矩阵分解被内置到该方法中。 从这个意义上讲，Mahalanobis方法使用了更多
   强大的模型比典型的多元极值分析方法。 第3章详细讨论了PCA与Mahalanobis方法的连接及其非线性扩展。 除了基于PCA的解释之外，它还具有自然概率解释作为下一节中讨论的EM方法的特例。

2. Mahalanobis方法是无参数的。 这在无监督问题中很重要，例如异常值检测，其中没有通过测试其在数据集上的性能来设置参数的有意义的方法。 这是因为参数调整无法提供地面实况。

3. 实际数据集中的特征通常以这样的方式提取，即极值可以暴露异常值，这对于Mahalanobis方法来说是一个简单的例子。 如果分析人员直观地了解（许多）特征值中的极端值表示异常值，那么有时可以使用Mahalanobis方法。 即使在所有特征都没有显示出这种特征的情况下，马哈拉诺比斯距离中的自然聚集效应也能够暴露异常值。 至少，人们可以利用马哈拉诺比斯方法作为集合方法的一个组成部分（参见第6章）利用对极值分析友好的特征子集。 与各种其他复杂检测器相比，最近邻检测器和马哈拉诺比斯方法的简单组合可以令人惊讶地稳健地执行。 将基于距离的分量添加到集合方法还可以确保不会完全错过图2.1 中数据点“A”之类的异常值。

4. 正如后面第4章所讨论的，大多数基于距离的方法对于包含N 个点的数据集需要O（N 2）时间。 即使对于包含几十万个点的数据集，计算异常值通常也在计算上具有挑战性

[4]有的一类支持向量机[538，539]学习的圆形隔板在转化的内核空间周围的数据，的质心包裹虽然。 正如下一章所讨论的，由于其基于PCA的解释，也可以对Mahalanobis方法进行核化。 此外，两种情况下的解决方案可以显示出密切相关（参见第3.4.3节）。

singola@gmx.com

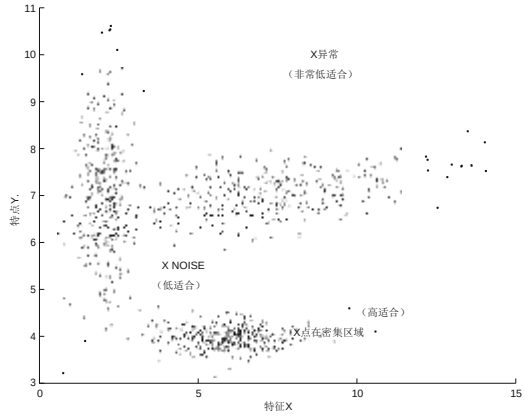


图2.8：将概率与异常行为联系起来

分数。 另一方面，马哈拉诺比斯方法在数据点的数量上是线性的，尽管在数据维度方面确实需要至少二次时间和空间。 然而，由于点的数量通常比维数的数量级大，因此马哈拉诺比斯方法在大多数实际环境中的计算时间方面具有显着优势。

因此，马哈拉诺比斯方法通常可以用作集合方法的附加组分，即使不希望在独立的基础上使用它。

2.4 异常分析的概率混合建模 - sis

上一节重点讨论了异常值模型的极值分析问题。 对于图2.7的示例，简单的Mahalanobis方法是有效的，因为整个数据集分布在一个大的集群中。 对于数据可能具有许多具有不同方向的不同聚类的情况，这种极值方法可能不具有效果。 图2.1说明了这种数据集的一个例子。 对于这种情况，需要更一般的基于分布的建模算法。

这种概括的关键思想是使用数据点的概率混合建模。 这些模型通常是生成模型，对于每个数据点，我们可以估计模型的生成概率（或概率）。 首先，我们假设生成模型的特定形式（例如，高斯混合），然后使用期望最大化（EM）算法估计该模型的参数。 估计参数以使观察到的数据具有生成模型的最大似然性。 给定此模型，我们然后估计基础数据点的生成概率（或fi t概率）。 分布的数据点将具有高的概率，而异常将具有非常低的概率。 2.8。

基于混合的生成模型的广义原则是假设数据是由k 分布的混合生成的，概率分布为1 ... k ，并重复应用以下随机过程：

G G

singola@gmx.com

- 选择r个的概率概率分布αR，其中[R ∈{1 ...， K}。
- 从G_r生成数据点。

我们用这个表示这个生成模型。 值αr 表示先验概率，并且直观地表示从混合分量r生成的数据的分数。 我们注意到αr的不同值，以及不同分布的参数r不是预先知道的，它们需要以数据驱动的方式学习。 在一些简化设置中，先验概率αr 的值可以固定为1/Gk，尽管在最一般情况下也需要从观察数据中学习这些值。 该
最典型的分布形式G.r 是高斯分布。 需要根据数据估计分布r 和先验概率 αr 的参数
数据具有生成的最大可能性。 因此，我们首先需要将数据集的概念定义为混合物的特定组分。
我们假设G_r 的密度函数由fr（·）给出。 由模型生成的数据点Xj 的概率（密度函数）由以下给出：

$$f^{point}(\overline{X_j}|M) = \sum_{i=1}^{k} a_i \cdot f^i(\overline{X_j}) \tag{2.22}$$

然后，对于包含N 的数据集由X1 ... XN 表示的记录的数据集，由模型生成的数据集的概率是相应的单个逐点概率（或概率密度）的乘积：

$$f^{data}(D|M) = \prod_{j=1}^{N} f^{point}(\overline{X_j}|M) \tag{2.23}$$

相对于数据集的对数似然性f t（）是上述表达式的对数，并且可以（更方便地）表示为不同数据点上的值之和。

$$L(D|M) = \log\left[\prod_{j=1}^{N} f^{point}(\overline{X_j}|M)\right] = \sum_{j=1}^{N} \log\left[\sum_{i=1}^{k} a_i \cdot f^i(X_j)\right] \tag{2.24}$$

需要优化该对数似然性以确定模型参数，并因此最大化数据点到生成模型的数量。 对数似然性因为它在不同数据点的附加性质及其数值方便性而优于似然性。

值得注意的是，如果我们（至少在概率上）知道哪个数据点是由混合物的哪个组分产生的，那么为混合物的每个组分分别确定最佳模型参数要容易得多。 同时，从不同组件生成这些不同数据点的可能性取决于这些最佳模型参数。 这种依赖性的圆度自然地建议了迭代EM算法，其中模型参数和概率数据点分配给组件被迭代地重新定义并彼此估计。 设Θ是表示描述混合模型的所有分量的整个参数集的矢量。 例如，在高斯混合模型的情况下，Θ将包含所有
组分混合物均值，方差，协方差，和参数α1...αK。 然后，EM算法以初始的Θ值集合开始（可能对应于随机值）
数据的分配指向混合物组分），并按如下方式进行：

singola@gmx.com

- **（E步骤）：**给定Θ中参数的当前值，确定后验概率 P（Xj| G,r, 由第r个混合分量产生 点XjΘ） 。 对所有点 - 分量对（Xj，Gr）执行该计算）。
- **（M步）：**给定数据点到簇的分配的当前概率，使用最大似然方法来确定所有参数Θ的值，其基于当前分配使对数似然性最大化。 因此，在高斯设置，所有簇手段，协方差矩阵，以及先验概率 α1 ...αK需要估计。

现在仍然需要解释E步骤和M步骤的细节。 E步骤简单地计算由混合物的每个组分生成的数据点Xj 的概率密度，然后计算每个组分的分数值。 这由贝叶斯后验概率定义，即数据点Xj 由分量r 生成（模型参数固定到当前的参数集合Θ）。 因此，我们有：

$$P(G_r|\overline{X_j},\Theta) = \frac{\alpha_r \cdot f^{r,\Theta}(\overline{X_j})}{\sum_{i=1}^{k} \alpha_i \cdot f^{i,\Theta}(\overline{X_j})} \tag{2.25}$$

由于某些符号的滥用，在概率密度函数中添加了上标Θ，以便表示在当前模型参数集合Θ下评估它们的事实。

接下来，我们描述M 的参数估计step，它最大化了可能性。 为了优化fi t，我们需要计算对数似然ft相对于相应模型参数的偏导数，并将它们设置为0以确定最佳值。 αr 的值易于估计，并且等于分配给每个聚类的点的预期分数，基于P 的当前值（r Xj, Θ）。 实际上，为了获得更小的数据集更强大的结果，增加了分子中每个簇的预期数据点数量。 1，分母中的总点数是N + k。 因此，估计的值 αR 是（1 +Σ N P （G- [R|XJ， Θ））/（k + Ñ ）。 这种方法是正规化的一种形式，它也被称为拉普拉斯平滑。 例如，如果N.非常小，那么一种方法将分配概率推向1/ k。 这代表了一种自然的先验关于聚类中点的分布的假设。

为了确定特定于混合物的特定分量r 的其他参数 ，我们简单地将P （ r Xj, Θ）的每个值 视为该分量中该数据点的权重，然后执行参数的最大似然估计。 那个组成部分。 这通常比必须同时处理混合物的所有组分简单得多。 精确的估算过程取决于手头的概率分布。 例如，考虑r的设置d 维中的第个高斯混合分量由以下分布表示：

$$f^{r,\Theta}(\overline{X_j}) = \frac{1}{\sqrt{|\Sigma_r| \cdot (2 \cdot \pi)^{(d/2)}}} \cdot \exp\left(-\frac{1}{2} \cdot (X_j - \mu_r) \Sigma_r^{-1} (X_j - \mu_r)^T\right) \tag{2.26}$$

在此，导率μr 为d维均值向量和Σ[R 是d × d 的非的广义高斯分布的协方差矩阵[R个分量。 Σ r的值表示协方差矩阵的行列式。 当混合分量的数量很大时，非对角线条目通常被设置为0，以便减少估计参数的数量。 在这种情况下，Σ的行列式r 简化的ES到沿各个尺寸的方差的乘积。

singola@gmx.com

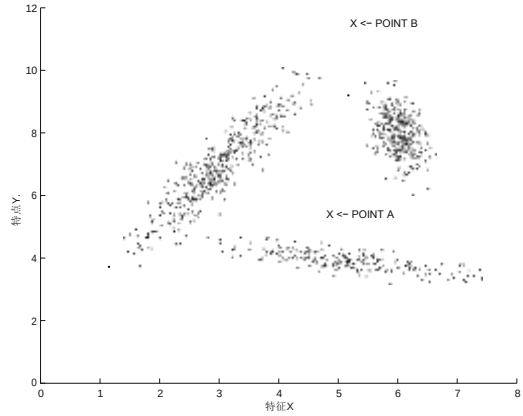


图2.9：EM算法可以确定具有任意相关性的聚类（重访图2.1）

可以示出的是的最大似然估计率μr 和$[\Sigma_{[R]}]_{IJ}$等于在该组件的数据点的（概率加权）指和共同变化。 回想一下，这些概率权重来自E步骤中的赋值概率。 因此，E步骤和M步骤彼此依赖并且可以概率地执行以收敛，以便确定最佳参数值Θ。

在过程结束时，我们有一个概率模型，将整个数据集描述为生成过程的观察输出。 该模型还以公式2.22的形式为每个数据点提供概率性值。 该值提供异常值。 因此，我们可以使用此方法对所有数据点进行排名，并确定最异常的数据点。 这个想法是远离数据密集区域的点（例如图2.8上部区域所示的点））将具有非常低的fi值。 这些点是数据中的异常。 如果需要，可以应用统计假设检验来识别具有异常低的fi t值的异常值。 但是，对于统计测试，应将对数函数应用于fi t值（即应使用对数似然值）以减小内点的相对方差（大的fi t值），以便具有非常低的fi t值的点将通过极值测试。

该方法需要混合组分的数量作为输入参数。 在某些情况下，可以使用有关数据的特定领域洞察力做出有意义的选择。 在没有这种见解的情况下，具有不同参数设置的混合模型集合是有用的[184]。 特别是，[184]中平均了在具有不同数量的混合成分的模型上获得的逐点对数似然得分。 此外，这些模型建立在数据集的不同样本上。 使用这种方法已经报道了优异的结果。

2.4.1 与聚类方法的关系

概率混合建模是聚类方法的随机版本，也可用于异常值检测（参见第4章）。 值得注意的是，高斯混合模型中的fi t值使用高斯指数中来自聚类质心的点的距离。 因此，单个高斯分布的对数似然性是马哈拉诺比斯距离，尽管来自多个高斯的加法因子不能简化为此。

singola@gmx.com

方式。 然而，最近的聚类的效果通常在数值中占主导地位。 在第4章的聚类模型中，只有到最近的聚类质心的距离才被直接用作离群值。 因此，第4 章的聚类技术可以被视为EM算法的硬版本，其中使用特定（最近）聚类来对点进行评分，而不是使用软概率组合中来自所有聚类的组合fi t值。

当聚类在不同的相关方向上具有细长的形状时，EM算法可以识别数据中任意定向的聚类。 这有助于更好地识别异常值。 例如，在图2.9的情况下，即使点'B'在绝对距离的基础上更靠近一个簇，点'A'的点也将低于点'B'的点。 这是因为高斯指数中的马哈拉诺比斯距离对数据中不同相关方向的距离进行了归一化。 实际上，数据点“A”更明显是一个异常值。

2.4.2 单一混合物组分的特殊情况

有趣的是，混合分布包含单个高斯分量的特殊情况（参见公式2.26）在实际环境中的效果令人惊讶。 这部分是因为使用单个高斯分量对应于2.3.4节的马哈拉诺比斯方法。 2.3.3.1节讨论了这种方法的具体优点。 正如我们将在第3章中看到的那样，这导致了主要组件分析（PCA）的软版本，由于其能够识别违反属性依赖性的数据点，因此已知其具有高效性。因此，从概率方法和线性模型的角度可以解释马哈拉诺比斯方法。

尽管单一混合物组分的使用似乎错过了真正的异常值（如图2.9中的异常值'A' ），但它的优点还在于混合物组分中没有一种能够覆盖一小块但紧密结合的异常值群。 当使用大量混合组件时，其中一个组件可能对应于一组紧密编织的异常值，如图中所示的异常值组2.10。 Mahalanobis方法将正确标记此群集中的点作为异常值，而混合模型（具有更多组件）会冒成将此小群集建模为合法混合组件的风险。 有趣的异常通常发生在小群集中，因为它们可能是由类似的潜在原因（例如，特定疾病或信用卡欺诈类型）引起的。 Mahalanobis方法能够将这些聚类识别为异常值，因为它们通常与数据的全局均值和协方差结构不一致。由于实际数据集的这些典型特征，像马哈拉诺比斯方法这样的非常简单的方法有时会胜过显着更复杂的模型。

正如下一章3.3.8 节所讨论的，人们还可以将Mahalanobis方法与内核方法结合起来，对更一般的分布进行建模。 例如，这些方法的一些变量可以正确识别图2.9中的异常值“A” 。这种方法的集合中心版本已被证明可以提供高质量的结果[35]。

2.4.3 利用EM模型的其他方法

上一节中讨论的EM模型将离群值得分量化为任何混合成分的点。 因此，假设所有混合成分都是正常类的实例。 一种不同的方法是可以获得关于正态类和异常类的分布差异的一些领域特定见解。 在这种情况下，不同的概率分布用于模拟

singola@gmx.com

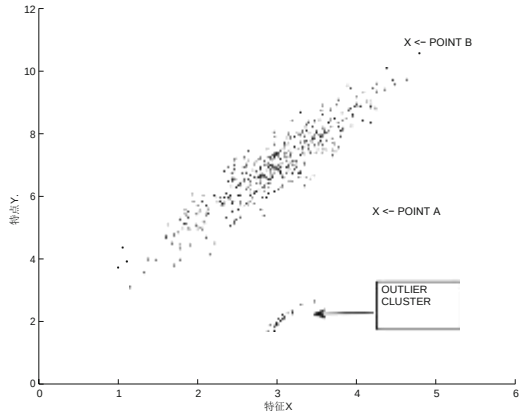


图2.10：单个混合物组分的使用对于小异常值簇的存在是稳健的

正常和异常的类。 一个点的离群值得分被量化为异常类别的ft，较大的得分表示异常。 在没有关于正常和异常类别分布差异的具体见解的情况下，这种方法通常很难使用。 例如，这种方法已用于识别特定类型的异常值，如噪声[110]。 在下一节中，我们将提供另一个设置示例，其中可以使用逼真（和不同）分布对正常类和异常类进行建模。

2.4.4 EM用于将得分转换为概率的应用

有趣的是，EM算法也可以用作许多此类异常值检测算法之后的最后一步，用于将分数转换为概率[213]。 注意，前一节的EM算法返回的fi t值（参见公式2.22）是概率密度值，不能解释为数值概率。 在数值概率方面表征异常值的能力是直觉和可解释性非常有用的一步。

这个想法是得分的分布可以被视为单变量数据集，然后可以用于概率生成模型。 在这种情况下，明确假定异常值点属于混合模型的一个组成部分（而不是简单地将它们视为具有低fi t值的点）。 请注意，只有在对异常值和非异常值类别的自然分布有一些额外的见解时，才能在此设置中区分异常值和非异常值类别。 因此，可以使用不同类型的分布来模拟异常值和非异常值组件。 在[工作]213]使用指数和高斯函数的双峰混合。 假设非离群点根据指数分布分布，而离群点根据高斯分布分布。 该假设是基于实际应用中的异常值得分的“典型”行为而做出的。 EM算法用于计算混合分布的每个分量的参数，以及相应的先验分配概率。 这些可用于通过使用贝叶斯规则将异常值得分转换为概率，因为它现在是

singola@gmx.com

可以计算数据点属于异常值分量的后验概率（见公式2.25）。 我们注意到，将混合分量分配给异常值类对于能够估计数据点是异常值的概率是至关重要的，这可以通过异常值分布的分布（高斯与指数）的差异来促进。 这两个班。

2.5 概率建模的局限性

参数方法非常容易受到基础数据中的噪声和过度拟合的影响。 混合模型总是假设数据的特定分布，然后尝试学习该分布的参数。 在这种分布的一般性和需要学习的参数数量之间存在自然交易。 如果未仔细校准此交易ff，则可能出现以下两种情况之一：

- 当模型的特定假设不准确时（例如，不恰当地使用高斯分布），数据不太可能很好地模拟模型。 结果，许多虚假数据点可能被报告为异常值。
- 当模型过于笼统时，描述模型的参数数量会增加。 例如，当使用不适当大量的混合物成分时，一个小但紧密结合的异常值簇可能会混合其中一种混合物成分。 图中说明了这种小型集群的一个例子2.10。 事实上，上一节将分数转换为概率的技术利用了单变量异常值得分通常在高斯分布中聚集在一起的事实。 不幸的是，没有办法将这种方法很容易地推广到多维数据集。 因此，当报告低点的点作为异常值（而不是特别建模的异常值类）时，由于小群异常值引起的过度拟合，总是可能错过真正的异常值。 减少过度拟合的一种可能性是将先验概率固定为1/ k，尽管这种假设有时可能导致欠缺。

正确选择简化假设总是很棘手。 例如，数据中的聚类可以是任意形状或方向，并且可以不具有简化的高斯假设，其中沿不同维度的数据值彼此独立。 这对应于Σ的非对角线项设置r在高斯情况下为0。 在实际数据集中，不同维度之间可能存在显着的相关性。 在这种情况下，人们不能假设矩阵Σ[R 是对角的，这将需要的学习Ô（d2）为每个群集参数。 当数据集中的点数较小时，这可能导致过度配置问题。 另一方面，在较大数据集的情况下，效率仍然是一个问题，特别是如果估计了更多数量的参数。 这是因为这些方法使用迭代EM算法，需要扫描整个算法

在E和M步骤的每次迭代中的数据步骤。 然而，这些方法仍然比许多基于点对点距离的方法更有效，这些方法对于包含N 个点的数据集需要O（N 2）时间。 这些方法将在第4章中讨论。

最后，可解释性问题仍然是许多参数方法的关注点。 例如，考虑广义高斯模型，该模型试图学习具有非零协方差的聚类。 在这种情况下，使用这些参数直观地解释聚类是很困难的。 相应地，定义简单直观的规则也很困难，这些规则提供了关于潜在异常值的批判性想法。 我们注意到这个问题

singola@gmx.com

可能不一定是所有参数方法的问题。 如果足够仔细地选择参数，则可以简单直观地描述最终模型。 例如，有时可以简单直观地根据数据的原始特征来描述没有协方差的高斯模型的简化版本。 另一方面，这种简化可能会导致不足和其他质的挑战。 然而，这些交易是几乎所有异常检测方法的特有，而不仅仅是概率模型。

2.6 结论和总结

在本章中，介绍了一些异常分析的基本概率和统计方法。 这些技术对于信心测试和极值分析非常有用。 还引入了许多用于极值分析的尾部不等式。 这些方法也可以推广到多变量场景。 极值分析作为将许多异常分析算法的分数转换为二进制标签的最后步骤具有极大的实用性。 在许多特定的应用中，即使对于一般的离群值分析，这种技术也非常有用。 用于异常值的概率混合建模的EM方法可以被视为Mahalanobis方法的推广。 该技术也可以被视为通常用于异常值检测的聚类方法之一。

2.7 书目调查

经典的不等式（例如，Markov，Chebychev，Cherno ff和Hoe ff ding）被广泛用于概率和统计中，用于限制基于聚合的统计的准确性。 这些不同方法的详细讨论可以在[407]中找到。 Hoe ff ding不等式的推广是McDiarmid的不等式[393]，它可以应用于Xi的不同值的更一般函数（超出线性可分和的总和）。 对此函数的主要限制是，如果函数的第i个参数（即Xi的值）更改为任何其他值，则函数的更改不能超过ci。

中心极限定理在概率和统计学中得到了广泛的研究[88]。 最初，该定理是针对独立和相同分布变量之和的情况而提出的。 随后，Aleksandr Lyapunov将其扩展到变量不一定相同分布的情况[88]，但确实需要独立。 对这些分布施加弱条件，确保总和不受少数组件的支配。 在这种情况下，变量之和也会收敛到正态分布。 因此，这是中心极限定理的广义版本。

统计假设测试已经以确定分布尾部[显着性的统计水平广泛使用在文献74，462]。 关于假设检验存在一个重要的文献，其中不仅可以测试单个数据点的异常性质，还可以测试数据点组的集体行为。 这些技术也用于在线分析处理场景，其中数据以数据立方体的形式组织。 通过使用假设检验来确定数据立方体的不同部分中的异常值通常是有用的[474]。

在方差[62]中首次提出了方差减少的偏差检测统计方法。 在[325]中提出了基于角度的多变量数据极值分析方法。 用于极值分析的多变量方法

singola@gmx.com

马氏距离在[提出343，493]。 当异常值位于簇之间的稀疏区域时，该技术不能很好地工作。 许多基于深度的方法已经在[已经提出了295，468]。 这些方法计算一组数据点的凸包，并逐渐剥离该船体角落处的点。 数据点的深度定义为剥离的凸包的顺序。 这些技术并没有得到太多的普及，因为它们具有与[[]的方法相同的缺点。 343]]用于发现内部定位的异常值。 此外，随着维数的增加，凸包计算非常昂贵。 此外，随着维数的增加，点的增加部分将位于最外侧的凸包上。 因此，这种方法实际上只能应用于二维或三维数据集。

应当注意的是，使用对孤立点检测概率方法是从在概率或不确定的数据[异常检测的问题不同的26，290，559]。 在前一种情况下，数据是不确定的，但这些方法是概率性的。 在后一种情况下，数据本身是概率性的。 关于EM算法的开创性讨论在[164]中提供。 当从指数分布族中提取混合物的组分时，该算法具有特别简单的形式。 [578]提出了一种在线混合学习算法，可以处理分类变量和数值变量。 EM算法的一个有趣的变化是将混合模型的一个组成部分作为异常组件处理[187]。 相应地，该组件来自均匀分布[187]，并且还被赋予低先验概率。 因此，这种方法不是确定不能很好地混合任何混合成分的异常点，而是试图确定混合物中这种特殊成分的点。 这种方法通常在建模噪声而不是异常方面更有效，因为混合模型中的特殊组件可能模拟噪声模式。 最后，最近还使用高斯混合模型来创建用于离群值检测的全局概率模型[583]。

EM算法也被用于从数据集中去除杂波[110]。 在这种情况下，通过将导出数据建模为泊松分布的混合，从数据集中去除噪声。 我们注意到[110]中的方法是为噪声检测而设计的，而不是真正异常的识别。 在[110]中显示，去除杂波（噪声）后数据质量的改善足以大大简化数据中相关特征的识别。 在[213] 中使用了使用混合物的特殊成分以将异常值得分的分布转换为概率的方法。 除了章节中讨论的方法2.4.4在[213]中讨论了使用逻辑S形函数的不同建模方法。 在[599]中已经讨论了在监督情景中将异常值得分转换为概率的方法。

EM方法的隐含假设是，一旦选择了混合成分，属性就有条件地彼此独立。 对属性相互依赖做出更强假设的概率方法是贝叶斯网络。 用于异常值检测的贝叶斯网络方法[66]利用现成的网络模拟属性之间的依赖关系，并使用这些依赖关系根据对这些依赖关系的违反情况将点评分为异常值。

2.8 演习

1. **[上尾Cherno ff界限]** 本章提供了上尾Cherno ff界限的证明草图，但不是完整的证明。 制定鞋面上的完整证据

singola@gmx.com

尾巴使用下尾证明作为指导。 你在哪里使用δ< 2·e - 1的事实？

2. 假设您使用“无偏”硬币100次。 您想调查硬币是否表现出异常行为（就未声称的“无偏见”而言）。 在无偏硬币的假设下，确定表示“尾部”数量的随机变量的均值和标准差。 通过使用（i）马尔可夫不等式来获得超过90个尾部的概率
（ii）Chebychev不等式（iii）Cherno ff Upper Tail Bound，（iv）Cherno ff Lower Tai l Bound和（v）Hoe ff ding Inequality。 [提示：可以使用上尾部或下尾部Cherno ff绑定，具体取决于您查看的随机变量。]

3. 重复练习2，当你知道硬币被操纵以在九个星期五中每八个显示“尾巴”时。

4. 使用中心极限定理通过正态分布来近似尾部的数量。 对于练习2和练习3的情况，使用累积正态分布来近似“尾部”的数量应该大于90的概率。

5. 制造过程产生小部件，每个小部件长100英尺，标准偏差为1英尺。 在正常操作下，这些长度彼此独立。

- 如果采样小部件长101.2英尺，使用正态分布假设来计算制造过程中某些异常情况发生的概率？
- 如果采样的小部件长96.3英尺，您的答案会如何变化？

6. 在上面的示例中，考虑对装配线中的10,000个小部件进行采样并发现其平均长度为1 00.05的情况。 在制造过程中发生异常的可能性是多少？

7. 使用MATLAB或任何其他数学软件绘制具有100个自由度的t分布。 在该图上叠加标准正态分布。 你能直观地看到差异吗？这告诉你什么？

8. 当协方差矩阵Σ的所有非对角元素都设置为零时，计算出EM算法的步骤，并且给定分量中的每个对角元素具有相同的值。 此外，先验分配概率等于1/ k，其中k 是混合分量的数量。 现在执行以下修改：

- 更改E步骤，以便确定性地将每个数据点分配给具有最高概率（硬分配）的群集，而不是软概率分配。 在什么基于距离的条件下，数据点被分配给群集？
- 该算法如何与k-means算法相关？
- 如果所有组件都受限制，那么你的答案会如何变化？
相同的聚类方差？

9. 使用从练习8得到的见解，制定出EM-算法与一组完整协方差的高斯混合模型如何矩阵$\Sigma_{[R}$，和一个固定设置先验，涉及一种广义k-means算法。 [提示：考虑马哈拉诺比斯距离计算的概念，用于k-means中的赋值。 如何定义先验概率？]

singola@gmx.com

10. 从UCI机器学习库中下载KDD Cup 1999数据集[203]。 从数据集中提取定量属性。 当非对角线元素设置为0时，将EM算法应用于20个混合分量。

- 确定每个数据点到学习分布的结果。 用最少的数据确定前10个点。 这些数据点是否与入侵攻击或普通数据相对应？
- 重复此过程，同时允许非零非对角元素。 你的答案如何改变？
- 从数据集中随机抽取990个点，然后添加上述第一个案例中找到的10个点。 在这个较小的数据集上重复此过程。 您是否在概率方面发现了重大异常？最低的fi t概率是否与上述第一种情况中的相同数据点相对应？
- 对上面的第二种情况重复相同的步骤。

11. 在UCI机器学习库的Ionosphere数据集上重复练习10的前两部分。 请注意，电离层数据集具有更高的维度（定量属性）和更少的记录。 在两种情况下，您是否确定了相同的前10个异常？什么是绝对的概率？这告诉您如何将这些算法应用于小型和高维数据集？

12. 让ž 是一个随机变量令人满意ë[Ž] = 0，并且ž ∈[A，B]。

- 表明Ë[等˙ž]≤ 等$^{2·（b-一）^2/8}$。
- 使用上述结果来完成Hoe ff ding不等式的证明。

singola@gmx.com

第3章

离群检测的线性模型

“我的本性是线性的，当我不是，我为自己感到骄傲。”- 辛西娅威尔

3.1 介绍

实际数据中的属性通常是高度相关的。 这种依赖性提供了彼此预测属性的能力。 预测和异常检测的概念密切相关。 毕竟，异常值是基于特定模型偏离预期（或预测）值的值。 线性模型专注于使用属性间依赖关系来实现此目标。 在经典统计文献中，该过程称为回归建模。

回归建模是相关分析的参数形式。 某些形式的相关分析试图从其他自变量中预测因变量，而其他形式则以潜在变量的形式汇总整个数据。 后者的一个例子是主成分分析方法。 两种形式的建模在异常分析的不同场景中非常有用。 前者是复杂的数据类型，如时间序列（见章节更加有用9 和11），而后者是现有的多维数据类型更有用。 对这两种形式的线性建模的单一讨论也为后面章节中的一些讨论奠定了基础。

线性模型的主要假设是（正常）数据嵌入在低维子空间中。 因此，不自然地使用该嵌入模型的数据点被视为异常值。 在基于接近度的方法的情况下，将在下一章中讨论，目标是确定空间的特定区域，其中异常点与其他点的行为非常不同。 另一方面，在线性方法中，目标是找到低维子空间，其中离群点的行为与其他点的行为非常不同。 这可以被视为聚类或最近邻方法的正交观点，其试图水平地（即，在行或数据值上）而不是垂直地（即，在列或维上）汇总数据。

© Springer International Publishing AG 2017

C.C. Aggarwal, *Outlier Analysis*, DOI 10.1007/978-3-319-47578-3_3

singola@gmx.com

可以将这些方法结合起来以获得更一般的局部子空间模型，这些模型可以通过以整体方式整合水平和垂直标准来识别异常值。

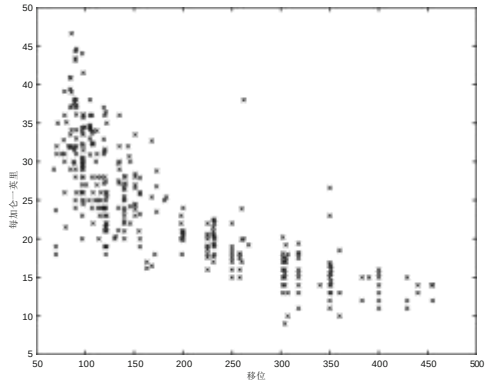
线性相关的假设是线性模型使用的信念的关键性飞跃。 对于特定数据集，这可能会或可能不会成立，这将对建模效率产生至关重要的影响。 为了解释这一点，我们将使用来自UCI机器学习库[203]的Autompg和Arrythmia数据集。 第一组数据包含描述各种汽车测量值和相应里程数（mpg）的功能。 第二组数据包含源自人类患者的心电图（ECG）读数的特征。

在图3.1（a）和（b）中，对于Autompg数据集，分别在每个位移和马力属性上显示了每加仑英里属性的相关性。 很明显，这些属性是高度相关的。 尽管在该特定数据集中也存在显着水平的噪声，但属性之间的线性相关性是显而易见的。 实际上，可以针对该数据集示出，随着维数的增加（通过从数据集中选择更多属性），数据可以沿着更低维度的平面对齐。 这在图3.1（e）的三维图中也很明显。 另一方面，当沿着心律失常数据集的三个测量维度的各种视图时（图3.1检查（c），（d）和（f）），显然数据分为两组，其中一组略大于另一组。 此外，将这种类型的数据分布嵌入到较低维的子空间中是相当困难的。 该数据集更适用于基于邻近度的分析，将在第4章中介绍。 引入这个例子的原因是重新审视第一章中关于选择正确数据正常模型的关键阶段所做选择的影响的观点。 在这种特殊情况下，很明显线性模型更适合数据集，其中数据自然地沿着低维超平面对齐。

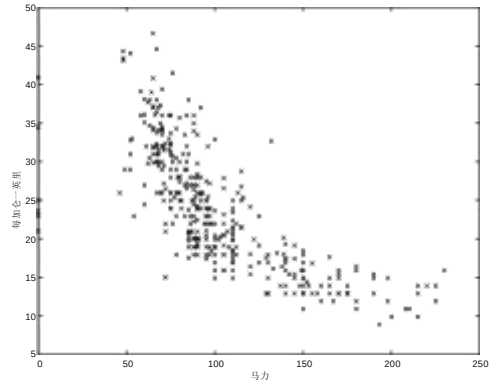
对于像离群检测这样的无监督问题，很难保证正常数据的特定模型是有效的。 例如，在一些数据集中，不同的属性子集可能适用于不同的模型。 使用第5章中讨论的子空间方法可以最好地解决这些数据集，可以结合行和列选择的功能进行异常值分析。 然而，在许多情况下，简化模型（如线性模型或基于接近度的模型）非常有效，而不会产生子空间方法的复杂性。 从模型选择的角度来看，数据的探索性和可视化分析在异常值检测的第一阶段非常关键，以便找出特定数据模型是否适合特定数据集。 在诸如异常检测之类的无监督问题的情况下尤其如此，其中为了测试各种模型的有效性而无法获得基础事实。

在本章中，将研究两类主要的线性模型。 第一类模型使用依赖变量和自变量之间的统计回归建模，以确定数据中特定类型的依赖关系。 当某些属性自然地预测其他属性时，这种形式的建模更有用。 例如，很自然地基于时间序列中的先前历史值的窗口来预测时间序列的最后值。 在这种情况下，通过创建依赖变量和独立变量的派生数据集并量化因变量的观测值与其预测值的偏差来利用面向依赖性的回归建模。 即使是多维数据，我们将显示在第7.7 章7对于该回归取向依赖性的模型可以被用于分解无监督异常值检测问题转换成d 为二FF erent回归建模问题d维数据。 第二类模型使用主成分分析来确定低维子空间

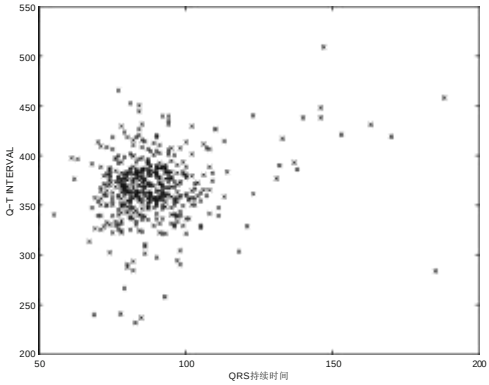
singola@gmx.com



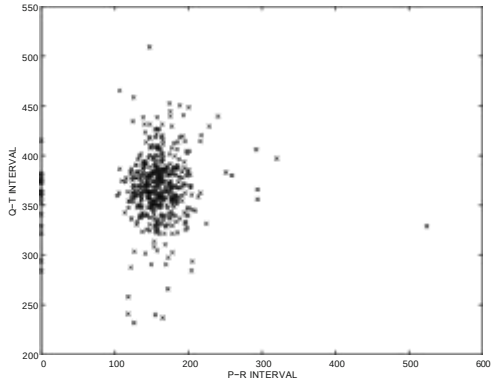
(a) 查看1（自动编译）



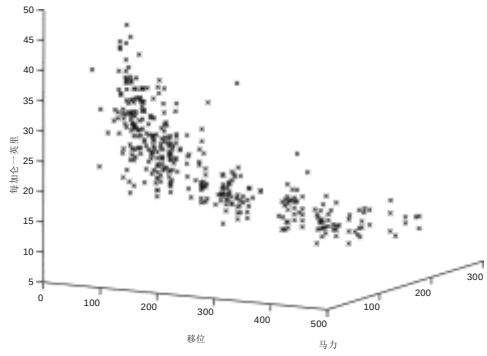
（b）查看2（自动编译）



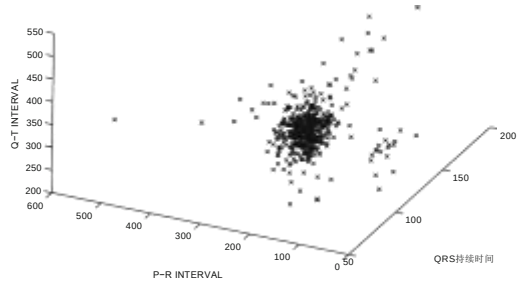
(c) View 1 (*Arrythmia*)



(d) View 2 (*Arrythmia*)



（e）3-d视图（Autompg）



（f）3-d视图（Arrythm

ia）图3.1：线性假设的有效性取决于数据集

singola@gmx.com

投射 这些模型对于传统的多维数据非常有用，因为所有属性都以同类方式处理，而不依赖于独立变量和自变量。 在技术和数学层面，两种形式的建模非常相似，并且使用非常类似的方法来导出最佳的低维表示。 主要的不同之处在于如何制定两种模型的目标函数。 本章中讨论的一些面向依赖关系的模型在制定后面章节中讨论的面向依赖关系的数据类型中的异常检测模型时也很有用。

应该强调的是，回归分析被广泛用于检测时间序列数据中的异常，本章中讨论的许多基本技术也适用于该场景。 这些技术将在第9章中详细讨论。 然而，由于问题的时间序列方面也基于时间上相邻的数据值的依赖性，因此在多维数据和时间序列数据的建模过程中存在许多微妙的差异。 因此，在本章中，将讨论更简单的多维异常值分析案例。 同时，讨论将是足够普遍的，以便引入将回归分析应用于时间序列场景（章节）所需的技术机制。 9

本章安排如下。 在3.2节中，将介绍异常值分析的基本线性回归模型。 在3.3节中，将讨论异常值分析的主成分方法。 这可以被认为是线性回归模型的一个重要特例，常用于离群值分析，它使用类似的优化模型。 因此，它在自己的部分给予专门的治疗。 我们讨论主成分分析的硬版本和软版本，并表明后者等同于前一章讨论的Mahalanobis方法。 此外，这种技术可以通过内核方法轻松扩展到非线性情况。 有关一类支持向量机的相关问题将在节中讨论。 神经网络在章节中介绍3.4。 线性模型的矩阵因式分解视图中部分讨论3.53.6。 3.7 节将研究异常值分析的线性模型的局限性。 第3.8 节包含结论和总结。

3.2 线性回归模型

在线性回归中，使用线性方程组对数据中的观测值进行建模。 具体而言，数据中的不同维度使用一组线性方程彼此相关，其中系数需要以数据驱动的方式学习。 由于观测值的数量通常远大于数据的维数，因此该方程组是过度确定的并且不能精确求解（即，零误差）。 因此，这些模型学习了最小化数据点偏离线性模型预测值的平方误差的系数。 错误函数的确切选择确定特定变量是否被特别处理（即，预测变量值的误差），或变量是否被均匀处理（即，估计的低维平面的误差距离）。 这些错误函数的不同选择不会导致相同的模型。 实际上，本章的讨论将表明模型在质量上可能非常不同，特别是在存在异常值的情况下。

回归分析通常被认为是统计领域中的一个重要应用。 在本申请的经典实例中，期望从一组独立变量中学习特定因变量的值。 这是时间序列分析中的常见情况，将在第9章中详细讨论。

singola@gmx.com

因此，特定变量由其他变量处理。 独立变量也称为解释变量。 这是具有上下文数据类型的共同主题，其中一些属性（例如，时间，空间位置或相邻系列值）被视为独立的，而其他属性（例如，温度或环境测量）被视为依赖的。 因此，本节中的大部分框架也将为后面章节中面向依赖关系的数据类型的分析奠定基础。 然而，对于简单的多维数据类型，所有维度都以均匀的方式处理，并且估计所有属性之间的最佳线性关系。 因此，在这两种不同设置的建模（即优化公式）中存在细微的差异。

考虑诸如时间和空间数据之类的域，其中属性被分为上下文和行为属性。 在这种情况下，特定的行为属性值通常被预测为其对话邻域中行为属性的线性函数，以确定与预期值的偏差。 这是通过从时间或空间数据构建多维数据集来实现的，其中特定行为属性值（例如，当前时间的温度）被视为因变量，以及其上下文邻域行为值（例如，先前窗口中的温度）） 被视为独立变量。 因此，预测因变量的重要性在估计偏差时是至关重要的，从而量化异常值。 在这种情况下，异常值是根据预测因变量的误差来定义的，而独立变量之间的异常关系被认为不太重要。 因此，优化过程的焦点在于最小化因变量的预测误差，以便创建正常数据的模型。 与此模型的偏差被视为异常值。

首先将研究具有因变量的回归分析的特例。 对此案例的讨论也为第9 章中的时间序列数据和第11章中的空间数据的案例提供了更详细的讨论。 此外，将使用这种技术将多维离群值检测问题分解为一组回归建模问题，将在第7章第7.7 节中进行研究。 在这种情况下识别异常值对于回归建模中的降噪也非常有用[467]，这本身就是一个重要的问题。

在后面的部分中，我们将介绍线性模型的更一般版本，其中不依赖于变量和独立变量。 这里的基本思想是假设（正常）数据位于特征空间中的低维超平面上。 来自该低维超平面的点（在垂直于超平面的方向上）的归一化偏差用于计算离群值得分。 正如我们稍后将展示的那样，通过使用能够将这些线性模型映射到更复杂的非线性模型的数据变换，可以极大地丰富这些简单的线性模型。 有趣的是，某些类型的转换甚至允许在更复杂的数据类型（例如时间序列数据，离散序列数据和图形数据）中使用这些线性模型进行离群值检测。 此外，

singola@gmx.com

3.2.1 依赖变量建模

变量y 可以建模为d 因变量（或维度）的线性函数，如下所示：

$$y = \sum_{i=1}^{d} w_i \cdot x_i + w_{d+1}$$

变量y 是响应变量或因变量，变量x1 ... xd 是独立变量或解释变量。 需要从训练数据中学习系数 w1 ... wd +1。数据集可能包含由X1 ... XN 表示的N个不同实例。 第j个点Xj 中的d 维表示为（xj1 ... xjd）。 第j个响应变量由yj表示。这N个实例对（Xj，yj）提供了如何实例
因变量与具有线性函数的d个自变量相关。 利用优化模型学习线性函数的参数，该模型最小化预测因变量的聚合平方误差。 这里的关键假设是因变量是手头应用程序的核心，因此错误仅针对此变量定义。 异常值被定义为从N个实例中学习的线性函数提供异常大的误差的那些数据点。 因此，响应变量的第j个实例与解释变量有关，如下所示：

$$y_j = \sum_{i=1}^{d} w_i \cdot x_{ji} + w_{d+1} + s_j \quad \forall j \in \{1, \ldots, N\} \tag{3.1}$$

这里，sj 表示对第j个实例建模时的错误，并且它提供异常值得分对（Xj，yj）。在最小二乘回归中，目标是确定回归 COE FFI cients 瓦特1 ... WD1，最大限度地减少错误Σ Ñ 小号2。
j=1 j
我们将首先介绍转换N个方程组所需的符号公式3.1 为矩阵形式。 所述Ñ ×（d + 1）矩阵，其Ĵ第i行是（XJ1 ... XJD，1）被表示为d，并且Ñ 的二FF erent值1矩阵（列向量）ý1 ... YN
由y表示。 值1包含在D的每一行中作为最终维度 ，以便解决常数项wd+1的效应。因此, D 的第一个d 维可以被认为是包含自变量的N个实例的d维数据集，y 是响应变量的相应的N 维列向量。 系数w1 ... wd +1 的（d + 1）维行向量用W 表示。 这创建了一个过度确定的方程组，对应于公式3.1的以下矩阵表示的行：

$$\overline{y} \approx D\overline{W}^T \tag{3.2}$$

注意使用“≈”，因为我们没有包括公式 3.1的误差项 sj 。 该
最小二乘误差Σ Ñ 小号预测响应变量的2通过最小化优化
j=1 j
平方误差 DW Ť 2为了学习COE FFI cient矢量 w ^ 。 通过应用
通过对目标函数的微积分，我们得到以下条件：

$$2D^T D\overline{W}^T - 2D^T \overline{y} = 0 \tag{3.3}$$

因此，这个最小化问题的最优系数由下面的等式提供：

$$\overline{W}^T = (D^T D)^{-1} D^T \overline{y} \tag{3.4}$$

注意，DT D 是（d +1）（d +1）矩阵，需要将其反转以求解该方程组。 在矩阵D 的秩小于（d + 1）的情况下，矩阵

singola@gmx.com

DT D 不可逆。 在这种情况下，公式3.2 表示一个未确定的系统，其中有很多解决方案，零误差（异常值得分）。 在这种情况下，由于缺乏数据，不可能以合理的方式对样本内数据点进行评分。 尽管如此，仍然可以通过使用学习的系数来对样本外的数据点进行评分。为了最小化过度配置，使用正则化。 鉴于正规化
参数α> 0，我们将术语αW2 加到目标函数中。 因此，新的目标函数J 如下：

$$J = ||D\overline{W}^T - y\dagger|^2 + \alpha||W||^2 \quad (3.5)$$

5）通过将J 的梯度相对于W设置为0，可以以类似的方式解决该优化问题。

$$2(D^T D + \alpha I)\overline{W}^T - 2D^T \bar{y} = 0 \quad (3.6)$$

这里，I 表示（d + 1）×（d + 1）单位矩阵。 正规化解决方案如下：

$$\overline{W}^T = (D^T D + \alpha I)^{-1} D^T \bar{y} \quad (3.7)$$

这个问题的封闭形式解决方案特别方便，是经典统计中回归分析的核心之一。 然后异常值得分对应

到N 维误差向量s = y - DW中 元素的绝对值。

但是，重要的是要注意使用不同的属性作为因变量
将提供不同的异常值分数。

要理解这一点，检查二维数据的特殊情况是有用的：

$$Y = w_1 \cdot X + w_2 \quad (3.8)$$

在这种情况下，系数w1 的估计具有特别简单的形式，并且可以表明对w1的最佳估计如下：

$$w_1 = \frac{Cov(X, Y)}{Var(X)}$$

这里，V ar（·）和Cov（·）对应于基础随机变量的方差和协方差。 一旦估计了w1 ，通过将X 和Y 的平均值插入线性相关性，可以进一步容易地估计值w2 。 通常，如果X 被回归
Y 而不是相反，人们会得到w1 = Cov（X，Y ）。 请注意
对于这些情况，回归依赖性将是不同的。

这组系数w1 ... wd+1定义了一个低维超平面，它尽可能地为数据提供数据，以便优化因变量中的误差。 对于相同的数据集，此超平面可能不同，具体取决于所选的变量

作为因变量。 为了解释这一点，让我们从UCI机器学习库[203]的Auto-Mpg数据集中检查两个属性的行为。

具体而言，Auto-Mpg数据集的第二和第三属性对应于与汽车相对应的一组记录中的Displacement和Horsepower属性。 这对属性的散点图如图3.2所示。 这个图中显示了三个回归平面，如下所示：

- 当马力（Y轴）取决于位移（X轴）时，绘制一个回归平面。 在这种情况下的残差是预测马力属性的误差。 优化了在各种数据点上的该残差的平方和。

singola@gmx.com

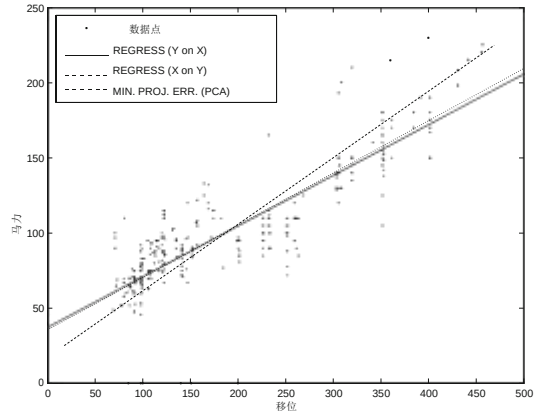


图3.2：最佳回归平面取决于优化的残差选择

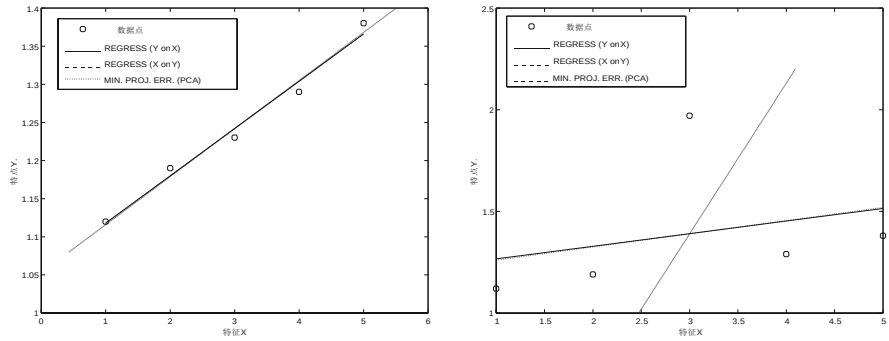
- 当位移（X轴）取决于马力（Y轴）时，绘制第二个回归平面。 在这种情况下的残差是Displacement属性的预测误差。
- 在最后一种情况下，目标是估计超平面，该超平面优化来自它的数据点的聚合均方距离（即，残差）。 因此，在这种情况下的残差是在超平面的法线方向上每个点到超平面的距离，并且它提供异常值分数。 这种超平面的计算将在后面的主成分分析（PCA）部分中讨论。

从图3.2 中可以明显看出，在这些不同的情况下，最佳超平面是非常不同的。 虽然均方投影距离的优化产生了一个超平面，它与Y- on-X 回归的情况有些相似，但两者并不相同。这是因为这些不同的情况对应于优化的残差上的不同误差选择，因此对应于不同的最佳配置超平面。 还值得注意的是，三个投影平面是共线的并且穿过数据集的平均值。

当数据很好地符合线性假设时，所有这些超平面都可能是相似的。 但是，噪声和异常值的存在有时会对建模过程产生严重的负面影响。 为了说明这一点，使用了来自[467] 的示例的变形。 在图3.3中，已经呈现了两组五个数据点的不同回归平面，对应于不同的因变量。 图3.3（a）和（b）中的两组五个数据点只有一个点不同，其中 Y 坐标在数据收集过程中失真。 因此，这一点并不能很好地解决剩余数据。

图3.3（a）中的原始数据集很好地符合线性假设。 因此，所有三个回归平面往往彼此非常相似。 然而，在单个数据点的扰动之后，所得到的投影平面被剧烈扰动。 特别是，Y-回归平面上的X 受到显着扰动，从而不再代表基础数据集中的真实趋势。 值得注意的是，最佳投影平面更接近两个回归模型的更稳定。 这是最佳投影平面的一般属性，因为它们以稳定的方式优化其方向

singola@gmx.com



(a) 回归没有值

（b）回归与一个异常扰动点

图3.3：异常值对回归分析质量的巨大影响

以便全面地处理数据。 这些飞机的确定将在下一节中讨论。

残差 sj 提供关于数据点j的离群值得分的有用信息 。 预计这些残差的平均值为0，并且可以直接从数据估计这些残差的方差。 这些残差的Z值也可以用作异常值。 回归建模中最常见的假设是假设误差项si 是正态分布，其以零为中心。 然后，第2 章讨论了t值测试可以直接用于不同的残差，随后可以删除偏远的观察结果。 残差的正常假设意味着系数的向量也正常地分布有均值和方差，如前所述。

该方法的一个问题是其存在异常值的不稳定性。 当异常值对回归有显着影响时，例如图3.3中X-on-Y 回归的情况（b），删除异常值可能会导致错误观察的消除，因为回归参数严重不正确。 具有讽刺意味的是，异常值的存在阻碍了异常值得分的正确建模。 解决这个问题的一种方法是使用集合方法，其中重复地对不同的点子集进行重复采样，以便对数据点进行不同的分数并对分数进行平均。 请注意，所有点都可以使用给定模型进行评分，因为样本外点也可以使用线性回归模型进行评分。 在训练点的数量很少的情况下，仅对于平均过程对样本外点进行评分。 换句话说，对一半的点进行采样以构建模型，并对剩余的一半进行评分。

3.2.1.1 因变量建模的应用

这些分数可用于与依赖于面向依赖的数据类型相关联的各种应用程序。 例如，在诸如空间和时间数据的上下文数据类型中，这些分数可用于发现异常的上下文异常值。 这些技术将在详细的章节中讨论9 和11。 这些分数还可用于删除那些对学习和回归建模应用程序有害的数据点。 例如，在回归建模应用程序中，可以移除具有大的训练错误的训练数据点，以便改进学习模型。

事实证明，即使对于无监督的异常检测，这种模型也是有用的。

singola@gmx.com

个人数据[429]。 基本思想是通过将其中一个属性作为因变量并将其余属性作为自变量进行复制来重复应用回归建模方法。 因此，对于d维数据集，可以创建总共d个回归模型。预测每个属性的平方误差以加权方式相加，以便定义数据点的异常值。 该模型将在第7章的7.7 节中详细讨论。

3.2.2 具有均方投影误差的线性建模

前一节讨论了特定变量（行为属性）被认为是特殊的情况，并且确定最佳平面以便最小化该变量的残差的均方误差。 回归建模的更一般形式是以类似方式处理所有变量的形式，并且确定最佳回归平面以最小化数据到平面的投影误差。 在下面的讨论中，假设包含N 个点的数据集是以中心为中心的，因此每个维度的平均值是0。

数据到平面的投影误差是点到它们投影到平面中的距离的平方的总和。 点'A'在平面上的投影是平面上距离'A'的最近点，并且使用在平面的法线方向上穿过'A'的线来计算。 该线与平面相交的点是
投影点。 因此，在这种情况下，让我们假设我们有一组变量x1 ... xd，相应的回归平面如下：

$$w_1 \cdot x_1 + \ldots + w_d \cdot x_d = 0 \tag{3.9}$$

假设数据是以均值为中心的，因此在这种情况下， 常数项wd+1正在丢失。 每个变量与系数相关联，并且在这种情况下缺少“特殊”（从属）变量y 。 考虑一个设置，其中我们有N个实例对应于d维数据点，用X1 ... XN表示。 如果分数
N 大于维数d，方程组是超定的并且全部
无论系数的矢量如何，数据中的点都可能不满足公式3.9
选择W = （w1 ... wd）。 因此，我们需要允许一个错误值 sj：

$$\overline{w} \wedge \cdot \overline{X}J = SJ \quad \forall \tilde{J} \in \{1 \ldots N\} \tag{3.10}$$

）因此，目标是确定的行矢量w ^ =（瓦特1 ... WD）COE FFI cients的，从而使
平方误差的总和Σ$\tilde{N}$ 小号2被最小化。 为了解决扩展问题
并且避免平凡解W = 0，假设归一化约束：

$$\sum_{i=1}^{d} ||W||^2 = w_i^2 \qquad w = 1 \tag{3.11}$$

注意，该缩放约束确保W · Xj = sj 的绝对值 恰好等于数学点Xj 距公式3.9中的超平面的距离。

如前所述，令D 为N × d 矩阵，其行中包含d维点，由X1 ... XN表示。 人们可以简洁地写出N 维列向量
不同数据点到该回归平面的距离为s = DW 根据
公式 3.10。 此列向量包含异常值分数。 的 大号范数 DW^T 2 的 2 || || 距离的列向量是目标函数，需要最小化

singola@gmx.com

确定最佳系数w1 ... wd。 由于公式3.11中的归一化约束，这是一个约束优化问题。人们可以将拉格朗日松弛的梯度设为DW^T 2 λ（w ^ ，以便导出关于最优参数向量的约束2 1）至0 w ^ 。这个约束结果是特征向量形式：

$$[D^T D]W^T = \lambda W^T \tag{3.12}$$

正半无限矩阵DT D的哪个特征向量提供最低目标函数值？在原始目标函数中用等式3.12 代入[DT D]W^T，将其评估为λW2 = λ。 沿特定特征向量的总平方误差等于特征值。因此，最佳矢量W 是DT D的最小特征向量。 注意，矩阵DT D 是以均值为中心的数据矩阵D的协方差矩阵Σ的缩放版本：

$$\Sigma = \frac{D^T D}{N} \tag{3.13}$$

矩阵的缩放不会影响其特征向量，但它会将特征值缩放为沿这些方向的方差而不是聚合误差。 因此假设数据由（d 1）维超平面紧凑地表示，该超平面垂直于协方差矩阵Σ的最小特征向量。 异常值分数对应于数据点距离其最近点（即，垂直于该超平面）的距离。 但是，维度（d 1）通常太大而无法发现足够的判别性异常值得分。 无论如何，上述解决方案为该问题的有效（和更一般）解决方案提供了第一步，该解决方案被称为主成分分析（PCA）。 PCA方法将上述解决方案概括为k维超平面，其中k 的值可以在1 ... d 1之间的任何值处变化。 由于其对异常值分析的重要性，该方法将在其自己的专用部分中与相应的应用程序一起讨论。

3.3 主成分分析

上一节的最小二乘公式简单地试图找到一个（d 1）维超平面，它对数据值具有最佳的fit，并且沿着剩余的正交方向计算得分。 主成分分析可用于解决此问题的一般版本。 具体而言，它可以找到任何维度的最佳表示超平面。 换句话说，PCA方法可以确定k维超平面（对于k <d的任何值），其使剩余（dk）维度上的平方投影误差最小化。 上一节中的优化解决方案是主成分分析的一个特例，它是通过设置k = 得到的d 1。

在主成分分析中，计算d维数据上的dd 协方差矩阵，其中第（i，j）个条目等于N个观察的集合的维度i和j 之间的协方差。 正如上一节中所讨论的，考虑一个多维数据集d 维数d 和尺寸N 。 所述N 的二FF erent行d 由X1 ... XN 表示。 每行都是数据中的d维实例。 第i行的各个维度由 Xi = [xi1 ... xid]表示，其中xij 是第i个实例Xi的第j维。 让我们用Σ，in表示数据集的dd 协方差矩阵 其中第（i，j）个条目是第i和 第j维之间的协方差。 由于数据集D 是以中心为中心的，因此协方差矩阵Σ可以表示如下：

$$\Sigma = \frac{D^T D}{N} \tag{3.14}$$

singola@gmx.com

该矩阵可以显示为对称且正半无限。 因此可以如下对角化：

$$\Sigma = P \Delta P^T$$

这里，Δ是对角矩阵，P 是矩阵，其列对应于Σ的（正交）特征向量。 对角矩阵Δ中的对应条目提供特征值。 在上述部分中，我们指出，到Σ的最小特征向量的正常超平面提供了（d 1）维超平面，其近似具有最小平方误差的数据。 说明这一点的另一种方式是作为（d 1）个最大特征向量的线性组合的子空间提供了一个轴系统，其中数据可以用非常小的损失近似表示。 可以用k个最大的特征向量推广该参数以定义相应的k- （近似）表示的维度子空间。 这里，k 可以是1 ... d− 1中的任何值，并且它可以不被设置为仅（d 1）。 异常值是该近似值的误差高的数据点。

因此，在主成分分析中，第一步是将数据转换为新的轴表示系统。 标准正交特征向量提供了应该投影数据的轴方向。 与异常值分析相关的主成分分析的关键属性如下：

属性3.3.1（PCA属性）主成分分析提供一组满足以下属性的特征向量：

- 如果数据被变换到对应于正交特征向量的轴系统，则沿每个轴（特征向量）的变换数据的方差等于对应的特征值。 此新表示中的变换数据的协方差为0。
- 由于沿着具有小特征值的特征向量的变换数据的方差较低，因此变换数据与沿这些方向的平均值的显着偏差可以表示异常值。

更多的细节和PCA的性质可在[找到33，296]。 PCA提供了比前一部分的一维最优解决方案更通用的解决方案，因为PCA解决方案根据参数k的选择提供任何维度的递归解决方案。 值得注意的是，PCA方法可能使用公式3.12的所有解，而不是仅使用最小的特征向量。

可以将数据变换为正交特征向量的新轴系统，其中变换的d维记录由$X_1^j \ldots X_N^j$表示。 这可以通过使用原始行向量表示Xi 和包含矩阵P 之间的乘积来实现
其列中的新轴（标准正交特征向量）：

$$\overline{Xi}^j = [x_{i1}^j \ldots x_{id}^j] = \overline{Xi}P \qquad (3.15)$$

令D^j为第i行包含变换点的变换数据矩阵$\overline{西}^j$。 可以表示去相关轴系统中的变换数据矩阵D^j
原始数据矩阵D的术语如下：

$$D^j = DP \qquad (3.16)$$

在这个新的表示中，新数据矩阵D^j中的属性间协方差为零，并且沿着各个属性的方差对应于沿着的坐标。

singola@gmx.com

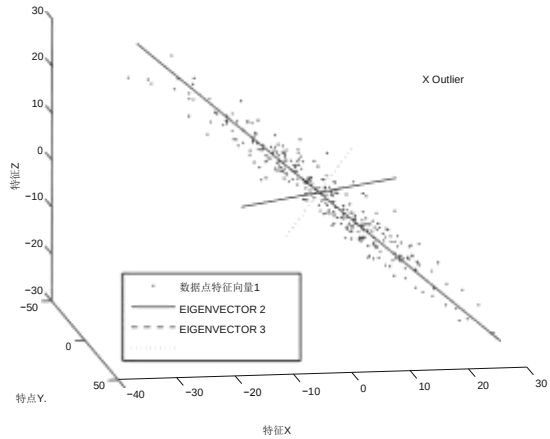


图3.4：特征向量对应于数据中的相关方向。 少量特征向量可以捕获数据中的大部分方差。

特征向量特征值。 例如，如果第j个特征值非常小，则此新变换表示中的x^j_{ij} 的值在不同的值上变化不大
的我和固定成本$\hat{J}$。 对于以中心为中心的数据，$x^j_{ij的}$这些值近似等于
平均值为0，除非数据点是异常值。

关于PCA的美妙部分是最大的特征向量一次性提供全局相关的关键方向。 这些方向也称为主要组成部分，它们是不相关的并且保留了大部分数据差异。 在实际设置中，大部分特征值通常非常小。 这意味着大多数数据沿着低得多的子空间排列。 从离群值分析的角度来看，这是非常方便的，因为不考虑这种对齐的观察可以假定为异常值。 例如，对于具有小特征值的特征向量j ，
第i个记录的x^j_{ij} 与$x^j_{kj的}$其他值的大偏差表示离群行为。 这是因为当j 被固定并且k 变化时，x^j_{kj} 的值变化不大。 因此，在这些设置中，值x^j_{ij} 可以被认为是不寻常的。

主成分分析在暴露来自下层数据集的异常值时的有效性可以用一个例子来说明。 考虑图3.4中所示的三维数据的散点图。 在这种情况下，相应的特征向量已经通过减小特征值（方差）来排序，尽管从这个二维透视图中的图中并不是立即显而易见的。 在这种情况下，沿第一特征向量的标准偏差是沿第二特征向量的标准偏差的三倍和沿第三特征向量的标准偏差的九倍。 因此，大多数方差将在由前两个特征向量形成的低维子空间中捕获，尽管通过仅选择第一特征向量也将捕获大量方差。 如果原始数据的距离指向与第一个特征向量相对应的1维线（并通过数据的平均值），则图中的数据点“X”将立即作为异常值暴露出来。 在高维数据的情况下，k维子空间。 然后可以通过检查通过数据点平均值的这个k维超平面的投影距离来计算数据点的残差。 距离超平面非常大的数据点可以被识别为异常值。 虽然是

singola@gmx.com

可以使用该距离作为离群值得分，可以通过如下归一化来进一步锐化得分。 到该超平面的平方欧几里德距离可以被分解为沿着最小特征向量的（dk）距离的平方和。 这些（dk）平方距离中的每一个应该除以相应的特征值（对于方差标准化），并且应该添加缩放值以提供最终结果。 这种缩放背后的直觉是沿着特征向量的方差是它的特征值，因此沿着较小的特征值的较大偏差应该被更大程度地奖励。

3.3.1 与马哈拉诺比斯方法的联系

在上述方法中，我们以硬方式去除k个最大的特征向量，并计算沿着剩余（dk）距离的这些平方距离的加权和作为离群值得分。 一个更简单的特殊情况是使用软方法来加权沿所有不同特征向量的距离，而不是选择一组特定的特征向量。 这一特殊情况与第2章第2.3.4节中的Mahalanobis方法相同。

通过沿每个主成分的方向评估数据点到质心的归一化距离来实现该计算。 令$\overline{e_j}$ 为第j个本征向量，沿该方向具有λ_j 的方差（特征值）。 整体归一化的异常值得分数据点的X，到质心[1] $\overline{\mu}$ 的数据由这些值的平方和给出：

$$\text{Score}(\overline{X}) = \sum_{j=1}^{d} \frac{|(\overline{X} - \overline{\mu}) \cdot \overline{e_j}|^2}{\lambda_j} \tag{3.17}$$

注意分母中存在λ_j，这提供了软加权。 计算$n \times d$ 数据矩阵D 的行的这些分数的简单算法如下：

1. 计算原始数据矩阵的协方差矩阵Σ d 和角化它作为$\Sigma = P \Delta P^T$ 。
2. 将数据D 转换为新的解相关轴系统，如$D^j = DP$ 。
3. 将D^j的每一列标准化为单位方差，将其除以标准偏差。
4. 对于D^j的每一行，从D^j的质心报告其（平方）欧几里德距离作为其异常值得分。

值得注意的是，当数据点沿着这些方向显着偏离时，公式3.17中对离群值得分的大部分贡献是由沿着主要分量的偏差提供的，具有小的λ_j值。 这也通过前述描述中的重要标准化步骤捕获。 此步骤认识到主要组件代表数据中的独立概念，并通过标准化实现转换维度的软加权形式，而不是以艰难的方式选择转换维度的子集。 从质心沿变换维度的点的欧几里德距离的平方和是a

具有d个 自由度的χ^2分布。 比较总残差的值

[1]虽然形心为平均值为中心的数据的来源，但也可以一旦协方差矩阵的特征向量已经计算转化非中心的数据。 公式3.17 的表达式继续适用于$\overline{\mu}$不一定为0的设置。

singola@gmx.com

为了确定异常水平的概率值，对于χ^2分布的累积分布。 上述方法首先在[493]中使用。

虽然可能不会立即显现，但上面计算的分数与第2 章第2.3.4 节中讨论的马哈拉诺比斯方法相同（参见练习11）。 具体而言，该部分的方程2.21 中计算的X 和μ 之间的马哈拉诺比斯距离值与上面的等式3.17中的得分完全相同，除了上面的特征向量分析更好地理解了该分数如何沿着不同的分解。 相关方向。 这种方法解释的另一个优点是它可以自然地扩展到数据分布在非线性流形上的设置，而不是图的线性设置。 3.4。 该扩展将在3.3.8节中讨论。

3.3.2 硬PCA与Soft PCA

Mahalanobis方法可以被视为一种软PCA，其中主成分是加权的而不是预先选择的。 PCA分解还允许选择忽略大的特征向量并仅使用最小的$\delta < d$ 特征向量，以计算异常值分数。 然而，这样做的好处尚不清楚，因为Mahalanobis方法已经通过特征值对每个特征向量的贡献的反加权进行了一种软修剪。 当所有属性都是相关数据集中的极值时，罕见值沿着长特征向量对齐并不罕见。 通过明确修剪长特征向量，可以忽略这些异常值。 因此，硬PCA仅侧重于发现依赖于面向依赖的异常值，而Mahalanobis方法（软PCA）可以发现依赖于面向依赖的异常值和极值。 从这个意义上讲，Mahalanobis方法可以被视为更加优雅的硬PCA概括。 Hard PCA专注于在低维空间中表示数据的重建错误，它引入了选择表示空间维度的附加参数。 在没有监督的设置中，引入参数总是导致数据特定的结果不可预测性，其中没有引导的方式来调整参数。

3.3.3 对噪音的敏感度

与因变量分析方法相比，主成分分析通常比少数异常值更稳定。 这是因为主成分分析计算了最优超平面的误差，而不是特定的变量。 当向数据添加更多异常值时，最佳超平面通常不会发生太大变化。 然而，在某些情况下，异常值的存在可能会带来挑战。 在这种情况下，存在几种用于执行稳健PCA的技术。 例如，可以使用此方法来确定第一阶段中明显的异常值。 在第二阶段，可以去除这些异常值，并且可以使用剩余数据更好地构建协方差矩阵。 然后用调整后的协方差矩阵重新计算得分。 这种方法也可以迭代应用。 在每次迭代中，移除明显的异常值，并构建更加重新定义的PCA模型。 最终异常值得分是最后一次迭代中的偏差水平。

singola@gmx.com

3.3.4 规范化问题

当不同尺寸的尺度非常不同时，PCA有时会产生不良结果。 例如，考虑包含Age和Salary等属性的人口统计数据集。 Salary属性的范围可以是数万，而Age属性几乎总是小于100。 PCA的使用将导致主要成分由高方差属性支配。 例如，对于仅包含Age和Salary的二维数据集，最大的特征向量几乎与Salary轴平行，而与Age和Salary属性之间的非常高的相关性无关。 这可以降低异常值检测过程的有效性。 因此，一种自然的解决方案是规范化数据，以使每个维度的方差为一个单位。 这是通过将每个维度除以其标准偏差来实现的。 这隐含地导致在主成分分析期间使用相关矩阵而不是协方差矩阵。 当然，这个问题并不是线性建模所特有的，并且通常建议对大多数离群值检测算法使用这种预处理。

3.3.5 正规化问题

当数据记录的数量N 很小时，不能非常准确地估计协方差矩阵，或者它可能是病态的。 例如，包含少量记录的数据集可能在某些维度上具有零方差，这可能低估了真实的可变性。 在这种情况下，希望使用正则化来避免过度配置。 这种类型的正则化直观地类似于第2章中针对EM方法讨论的拉普拉斯平滑的概念。 基本思想是通过向其添加αI 来调整协方差矩阵Σ ，其中 I 是d d 单位矩阵并且α>0是小正则化参数。 然后使用（Σ+ αI）的特征向量来计算得分。 这种修改的基本效果直观地等同于在计算协方差矩阵之前向每个维度添加少量具有方差α 的独立噪声。

或者，可以使用交叉验证进行评分。 数据被分成m个折叠，PCA模型仅在（m 1）个折叠处构建。 剩下的折叠得分。 每次折叠重复该过程。 以集合为中心的变体是对数据的子集进行采样以构建模型，并对所有点进行评分。 在多个样品上重复该过程并对得分进行平均。 这种方法的有效性见[35]。

3.3.6 应用于噪声校正

本书的大部分内容都致力于将异常值作为噪声去除或将异常值识别为异常。 但是，在某些应用程序中，纠正异常值中的错误以使它们更接近于数据中的广泛模式可能是有用的。 PCA是实现这一目标的自然方法，因为主要组成部分代表了广泛的模式。 基本思想是数据点在k上的投影对应于最大特征值（并且通过数据均值）的维超平面提供校正的表示。 显然，这种方法很可能比其他大多数正常数据点更显着地纠正异常点。 [21]中提供了一些关于为什么这种方法可能提高数据质量的理论和实验结果。

一种类似的方法，以PCA，称为潜在语义索引（LSI）也已经用于文本数据，以提高检索质量[162，425]。 文本表示本质上是

singola@gmx.com

嘈杂，因为同一个词可能意味着多个事物（同义词）或相同的概念可以用多个词（多义词）表示。 这导致几乎所有基于相似性的应用中的众多挑战。 特别是，在[425]中已经观察到，在文本数据中使用这种降维方法显着地提高了相似度计算的有效性，因为同义词和多义词的噪声效应减少了。 已经观察到[425]这种降维方法显着改善了文本中的相似度计算，因为同义词和多义词的噪声效应减少了。 LSI的技术[162]是PCA的一种变体，最初是为高效索引和检索而开发的。 然而，最终观察到相似度计算的质量也得到了改善[425]。 这一观察结果在[21]中得出了合理的结论，其中在理论上和实验上都显示了基于PCA的技术的显着降噪。

一种更有效的噪声校正方法是将异常值去除和重新插入与校正过程相结合。 第一步是执行PCA，并根据最佳表示平面的t检验去除顶部异常值。 随后，再次对此清洁数据集执行PCA，以便更准确地生成投影子空间。 然后可以在该校正的子空间上执行投影。 如果需要，该过程实际上可以迭代地重复，以便提供进一步的改进。 在[467]中提出了一些以稳健的方式执行回归分析和异常值去除的其他方法。

3.3.7 有多少特征向量？

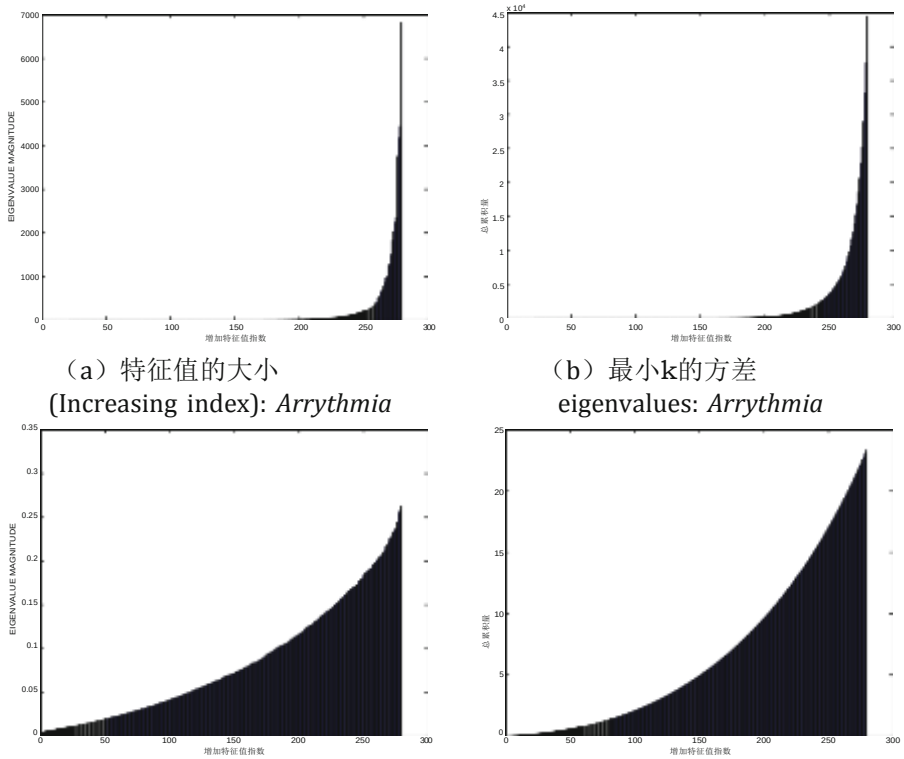
如前所述，具有最大方差的特征向量为数据表示，异常值分析和噪声校正提供了最丰富的子空间。 在某些情况下，没有必要以硬方式选择维度的子集，因为具有特征值的软逆加权是充分有效的。 然而，在诸如噪声校正的许多应用中，需要通过选择特定数量的特征向量将数据投影到较低维度的子空间中。 因此，关于如何确定投影子空间的维数k，出现了一个自然的问题。

大多数实际数据集中的一个观察结果是，大量的特征值相对较小，并且大多数方差集中在少数特征向量中。 图3.5 所示的一个例子显示了UCI机器学习库[203] 的Arrythmia数据集的279个特征向量的行为。 图3.5（a）显示了递增阶数的特征值的绝对值，而图3.5（b）显示了top-k 特征值中保留的总方差。 实质上，图3.5（b）是通过使用图3.5（a）中特征值的累积和得出的。 虽然它在一开始就有争议
关于Arrythmia数据集沿着许多维度微弱相关的章节，在成对的基础上，值得注意的是，仍然有可能找到少数方向的全局相关性，大多数方差都是如此。 保留。 实际上，可以证明，最小的215个特征值（279个中）累积地包含小于1%的数据集方差。

换句话说，大多数特征值非常小。 因此，相对于平均行为，保留对应于极大值的特征向量是值得的

[2]这部分是因为数据集相对较小，只有452条记录。 例如，图3.5（c）和（d）中的结果表明，即使对于相同大小的均匀分布的数据集，由于（不期望的）过度拟合，也可能在特征值中找到一些偏差。 另一方面，PCA的强度是即使弱相关的累积效应随着维数的增加而放大，并且可以找到包含信息预测的低得多的子空间。

singola@gmx.com



（a）特征值的大小 (Increasing index): *Arrythmia*

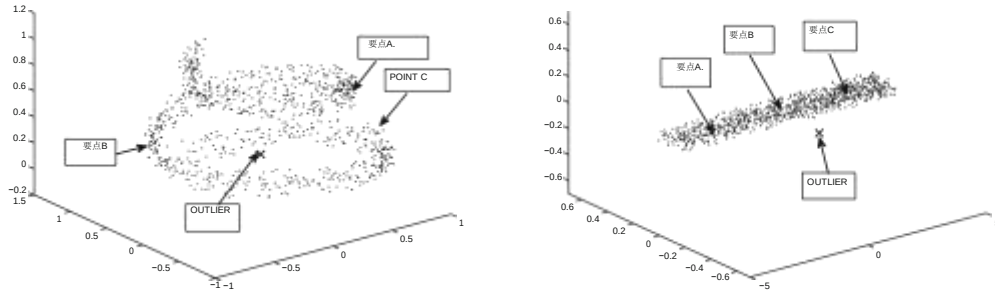
（b）最小k的方差 eigenvalues: *Arrythmia*

（c）特征值的大小 制服：仅限452条记录

（d）最小k个特征值的方差 统一：仅限452条记录

图3.5：一些特征值包含大部分方差（Arrythmia）

singola@gmx.com



（a）具有异常值的非线性流形 （b）（a）的变换表示

图3.6：非线性变换可以揭示线性模型难以发现的异常值

特征值。 如何确定什么是“非常大？”这是极值分析方法的经典案例，在第2章中介绍。 因此，每个特征值被视为数据样本，并且统计建模用于使用假设检验来确定大值。 在这种情况下的挑战是样本大小通常很小，除非数据集的维度非常高。 即使对于相对高维度的数据集（例如，50维数据集），可用于假设检验的样本数量（50个不同的特征值）相对较小。 因此，这是t值测试的良好候选者。 该T̃- 值测试可以与特定级别的重要性和适当的自由度结合使用，以确定所选择的特征向量的数量。

3.3.8 非线性数据分布的扩展

虽然PCA方法适用于线性对齐的数据分布，如图3.4所示，但在数据沿非线性流形对齐的情况下，PCA的vanilla版本并不有效。 这种歧管的一个例子如图3.6（a）所示。 在这种情况下，可以使用称为内核技巧的技术将PCA扩展到非线性设置。 非线性PCA的基本思想是使用数据变换，以便数据沿着变换空间中的线性超平面对齐。 换句话说，映射将非线性流形展开为线性嵌入。 图3.6中示出了这种嵌入的说明性示例（b）中。 在构建这样的嵌入之后，我们可以使用公式3.17的得分计算方法。

如何构建这样的嵌入？一种天真的方法是使用显式数据的多项式变换。 例如，二阶多项式变换将代表具有$d + d^2$维度的每个点。 换句话说，包括d 原始尺寸，并且对应于数据值的成对产品添加d^2个新尺寸。 随后，可以从协方差中提取所有非零特征向量这种新表现的矩阵。 请注意，在此设置中非零特征向量的数量可能大于d ，因为我们不知道变换空间的维数。 但是，非零特征向量的数量总是不大于数据点的数量。 公式3.17 可应用于此新表示以确定分数。 不幸的是，这不是一个实用的方法，因为它扩大了数量

singola@gmx.com

尺寸为$O(d^2)$，协方差矩阵的大小为$O(d^4)$。 从计算的角度来看，这种方法是不切实际的。 幸运的是，这些转换不需要明确地完成；相反，它们可以通过精心设计的隐式使用

所有数据点对之间的$N \times N$ 相似性矩阵。 该技术也被称为核技巧，尽管一般化的观点包括各种方法，例如频谱嵌入[297]和ISOMAP [542]。 本节描述的方法是使用核函数的Mahalanobis方法的直接非线性推广，并在[35]中作为集合中心自适应引入。 其中一些想法也与内核白化的概念有关[541]，其主张在应用任何学习方法之前将所有内核特征缩放到单位方差。

为了简化进一步的阐述，我们将首先描述从点积相似性矩阵而不是协方差矩阵中提取（线性）PCA的替代方法。 线性PCA的这种替代描述更容易推广到非线性情况。 在我们较早的PCA的描述中，我们首先构造的协方差矩阵$\Sigma = \frac{D^T D}{N}$ 与平均值为中心的矩阵D 和角化它以提取其$d \times d$ 特征向量矩阵P，

其中的列是新的基础向量。 随后，通过将这些基矢量上的数据投影为$D^j = DP$ 来给出变换的数据表示。 在确定新表示之后，离群值分数简单地等于距离
在将变换后的数据标准化为沿每个（变换的）维度的单位方差之后，从质心开始。 因此，关键是提取数据表示的新嵌入D^j以便提取异常值得分。 P 列中的基础表示仅提供用于计算嵌入D^j 的中间步骤，并且其本身对于计算离群值得分不是必需的。 PCA的相似性矩阵方法直接提取该嵌入D^j而不计算这些基矢量。

在PCA的相似性矩阵方法中，$N \times d$ $N \times N$ 点积矩阵$S = DD^T$ 被构造来代替$D^T D$ 协方差矩阵$d \times d$。 注意，S 是点积相似度矩阵，其中第（i，j）个条目等于第i个和第j个数据点之间的点积。 随后，正半有限相似矩阵S 被对角化如下：

$$S = DD^T = Q\Lambda^2 Q^T \tag{3.18}$$

在这里，Q 是$N \times N$ 矩阵，其列包含正交向量，虽然只有顶d 特征向量具有非零特征值，因为矩阵DD^T 最多有秩d。 Λ^2是包含正半有限矩阵的非负特征值的对角矩阵。 然后，在替代方案中的相似性矩阵的方法来PCA，可以示出的是，顶d 缩放的特征向量$Q\Lambda$（即，第一个d 的列$Q\Lambda$）得出的新$N \times d$ 数据表示[515]。 因此，这种替代方法[3]到PCA直接
从$N \times N$ N 点积的特征向量和特征值中提取嵌入
相似性矩阵小号= DD^T 没有计算基础矩阵P 。 这对于非线性维数降低特别有用，因为通常很难根据原始特征来解释转换后的基础。

非线性PCA的关键思想是，通过在上述方法中用精心选择的核相似性矩阵替换点积相似性矩阵$S = DD^T$，将非线性流形映射到线性超平面上，如图3.6中的映射（一）（数据矩阵d）到图3.6（b）（转化嵌入$Q\Lambda$）。 这些相似性矩阵将在3.3.8.1节中讨论。 这种方法是众所周知的技术的本质，称为内核技巧，相应的方法称为内核

[3]这种等价可以用矩阵D的SVD分解来显示。 实际上，DD^T和$D^T D$的非零特征值是相同的。

singola@gmx.com

PCA。 关键是假设S 的（i，j）值等于某些Φ（Xi）·Φ（Xj）
根据更高维度的一些未知变换Φ（Xi）变换空间 -
先进而精湛。 因此，相似度矩阵S 可用于计算马哈拉诺比斯分数，如下所示：

1. 使用off-the-shelf核函数或其他相似性矩阵计算（例如，在谱方法中使用的sparsi fi ed 和归一化相似性矩阵[378]）构建正半有限NN 核相似度函数S.

2. 对矩阵S = Q^ 2QT进行对角化，并将新的N × k 嵌入Dj设置为第一个 k 列Q对应最大特征值Λ。 默认假设是
设置k 使得包括S的所有非零特征向量。 注意，在非线性设置中k 可以大于d 。

3. 将Dj的每一列标准化为单位方差，将其除以标准偏差。

4. 对于Dj的每一行，从Dj的质心报告其（平方）欧几里德距离
作为其异常值得分。

为什么我们使用所有非零特征向量？在异常值检测等问题中，总体趋势并不充分，因为人们正在寻找异常。 事实上，大多数异常值都是沿着较小的特征向量强调的。 即使单个数据点沿着小的特征向量显着偏离对于异常值检测也很重要。 选择所有非零特征向量非常重要，因此不会错过重要的异常值。 假设Q 和Λ的列按特征值减小的顺序排序。 应该选择（唯一）值
的k 使得$\Lambda_{KK} > 0$，Λ_{k1}中，$k1 = 0$。在实践中，后者的值将不会精确为零，因为本征值计算过程中的数值误差的。 此外，这样计算错误放大了沿小特征向量得分的不准确性。 因此，一些保守的阈值小号上特征值（例如，10-8）可以用于选择cut-ÓFF点。 另一个警告是，由于过度配置，该方法可能会产生不良结果
当所有N个本征向量都非零时；在这种情况下，要么丢弃最大的特征向量，要么使用样本外实现（参见第3.3.8.2节）是有帮助的。

3.3.8.1 相似矩阵的选择

如何选择相似度矩阵S？ S的第（i，j）个条目由数据点Xi 和Xj之间的核相似度K（Xi，Xj）定义。 有许多öFF-常用于这样的设置中使用的现成的内核函数的，虽然一般的共识是有利于高斯核[的270，541]。 核函数S = [K（Xi，Xj）]的一些常见选择如下：

功能	形成
线性内核	$K(X_i, X_j) = X_i \cdot X_j$ （默认为PCA）
高斯径向基核	$K(X_i, X_j) = e^{-\|\|X_i - X_j\|\|/\sigma}$
多项式核	$K(X_i, X_j) = (X_i \cdot X_j + c)^h$
Sigmoid Kernel	$K(X_i, X_j) = \tanh(\kappa X_i \cdot X_j - \delta)$

singola@gmx.com

请注意，没有任何变换的线性PCA的特殊情况（第3.3节）对应于K（Xi，Xj）= Xi Xj，相应的内核也称为线性内核。 由于PCA在技术上是针对均值中心核矩阵定义的，因此可以对核矩阵进行中心[4] 。 但是，它在线性或非线性设置中都不是必需的，并且有时会在非线性设置中引起意外问题
特征向量非零。

许多这些内核函数都有与之相关的参数。 嵌入的形状取决于内核函数及其参数的选择。请注意，使用任意内核函数可能会导致嵌入与图3.6（b）稍微不同，这只是说明性的。 此外，不同的核函数或参数选择在暴露异常值时可能具有不同的有效性水平。 不幸的是，在无监督的方法中，没有选择这种内核的有意义的方法
功能或参数。 高斯[5] 内核的使用特别常见，因为它在不同的数据集中具有稳定性。 在这种情况下，参数 σ 应为
与点之间的成对距离相同的数量级。 较小的σ 值可以模拟复杂的分布，但也会导致过度拟合。 σ 产生的大值结果类似于线性马哈拉诺比斯方法。 对于小于1000点的小数据集，可能需要σ 值在点之间的中间成对距离的两倍和三倍之间[35]。

核函数仅提供计算相似性矩阵S的许多方式中的一种。 计算相似性矩阵的其他一些方法如下：

1. 人们可以通过仅保留每个数据点的相互k最近邻来稀疏化相似性矩阵。 相似度矩阵中的所有其他条目都设置为0.这种类型的相似性矩阵用于谱方法[378]，并且它倾向于支持局部嵌入。 此外，第（i，j）个条目可以通过行i 和行j 的和的几何平均值来归一化，以便提供局部归一化[297]。 这种局部归一化用于许多设置，例如第4章4.2.1节中讨论的LOF方法。

2. 通过使用ISOMAP的测地方法，可以使用点之间的成对距离计算。 随后，使用余弦定律将成对距离转换为成对相似性。 这种方法在[33]中讨论。 与光谱方法不同，这种方法倾向于支持全局嵌入。

通过选择特定类型的相似性矩阵，可以有效地控制将在各种应用中发现的异常值的类型。

3.3.8.2 实际问题

值得注意的是，对于包含N 个点的数据集，内核嵌入可以是N 维的。 在高斯核的情况下，嵌入维度通常大于输入数据集的维度。 每个特征向量定义嵌入的维度。 虽然大的特征向量对于像聚类这样的应用更为重要，但沿着小特征向量的变化对于异常值检测更为重要。

[4]内核矩阵可以是平均值为中心，通过使用更新S（IU / N）S（IU / N），其中I是单位矩阵，U是仅含有一个1秒NN矩阵。 ×

[5]这是值得注意的是2的因子从为了简单高斯核函数的指数的分母删去。–关于σ选择的所有讨论都符合这一惯例。

singola@gmx.com

将变换表示的维度标准化为单位方差强调了这些特征向量的重要性，如线性马哈拉诺比斯方法的情况。 保守的方法是使用所有非零特征向量（同时排除由数值误差引起的那些非零特征向量）。 然而，当几乎所有N个本征值都非零时，它有时是过度拟合的结果，结果可能很差。 一个
第二个问题是内核矩阵S 的大小是O（N 2），它可能非常大。 例如，对于包含十万个点的数据集，这种方法不是
实际的。

解决这两个问题的一种方法是使用基于可变子采样的集合中心自适应[35] [32]。 可以从50到1000点之间的大小为s 的数据重复绘制随机样本。 每个这样的样本用于（有效地）构建N 个点中的每个点的嵌入在其维度最多为s的空间中。

第一步是使用样本构建ss 相似性矩阵S. 随后，我们提取小号特征向量和特征值小号作为小号= QΛ2QT ，其中Q 和Λ都
s × s 矩阵。 然而，我们滴的零个向量 Q ，并且因此仅保留k̨<S 非零[6]的特征向量Q。对应的s × k 矩阵分别是Qk 和Λk。 因此，Qk的Λk̨ 提供的嵌入小号在-采样点。

对于剩余（N s个）外的采样点中，（N s个）小号内核相似所以被构造[7] 的外的采样点之间和在采样点。 换句话说，该矩阵中的第（i，j）个条目是第i个采样点和第j个采样点之间的核相似度 ，它也可以表示为之间的点积。
变换空间中的点。 我们将展示如何使用该矩阵来推导每个样本外数据点的k维嵌入，这与上述样本内点的嵌入一致。

令包含样本外点（在其行中）的k维嵌入的未知（N - s）× k 矩阵由Vk表示。 由于矩阵Vk 未知，目标是
使用所以和在样品中嵌入Qk的Λk̨ 以便估计它。 注意，在相似性矩阵中的每个条目所以可近似表示为成对点积的外的样本点（行VK点（行样本内）和Qk的Λk̨在转化的空间）。人们可以利用矩阵的产品作为表达这种关系所以= VK（Qk的Λk̨）T̂ 。 然后，通过
将该关系与QkΛ-1相乘后，很明显这些样本外点的（N - s）× k 嵌入矩阵由Vk = SoQkΛ-1 给出[481] 。 该小号× k̨ 矩阵Qk的Λk̨ （其是样品中的点的嵌入）k推叠与（Ñ - 小号）× k̨ 矩阵VK = SoQkΛ-1以创建单个Ñ × k̨ 嵌入矩阵Ë 所有的Ñ数据点：

$$\dot{E} = \begin{matrix} \Sigma \\ Q_k\Lambda_k \; S \\ {}_oQ_k\Lambda^{-1}{}_k \end{matrix} \qquad (3.19)$$

k

E的每列标准化为零均值和单位方差。 此嵌入用于一次性对所有数据点进行评分。 注意，由于标准化操作，标准化矩阵E 的平均行 向量是零向量。 将每行E 的平方距离与该行的平均值E （其为原点）报告为该行的离群值得分。 结果，每个点将获得一个离群值。 将得分除以E 的维数，以确保每个集合中所有点的平均得分

[6]由于数值误差，某些特征向量可能是非零或负的。 在这种情况下，最好是使用像10-非常保守的阈值[8]，以从那些由numer- iCal的错误区分真正的非零值。 这也很重要，因为由于重新缩放Dj，数值误差会更严重地损害沿着小特征值的计算。

[7]对于某些与数据相关的内核（如谱方法和ISOMAP），无法构造
尽管存在近似方法[79]，但是样本之外的相似性矩阵完全类似于So 。

singola@gmx.com

组件为1（因为标准化）。 使用不同的样本集重复整个过程m 次，以便每个点接收m个不同的异常值分数。 每个点的平均得分被报告为最终结果。

3.3.8.3 对任意数据类型的应用

该方法的一个美妙方面是，只要在一组N个对象之间存在正半场有限相似矩阵，就可以总是通过从相似性矩阵中提取多维嵌入来使用该方法。 主要的技术障碍是任何矩阵S，可以表示为DDT 以在变换空间中获得嵌入D ，可以显示为正半无限。 例如，如果我们使用一些域特定函数计算一组N个时间序列之间的NN 相似性矩阵，并且没有找到相似性矩阵S要是正半无限的，这意味着不存在嵌入，其点产品将为您提供所需的相似性。 有两种方法可以避免这种问题：

- 内核相似度函数已经设计用于各种数据类型，例如时间序列，字符串和图形。 这些相似性函数是特殊的，因为它们保证是正半无限的。 关于这些不同类型的核函数的讨论可以在[33]中找到。 请注意，并不总是可以使用内核，因为相似性矩阵可能以依赖于域的方式定义。

- 对于具有负特征值的任意相似性矩阵S ，如果它们的绝对幅度相对较小，我们可以丢弃具有负特征值的特征向量。 人们可以将这种方法视为稍微调整相似性矩阵的间接方式，以迫使其满足多维嵌入的要求。 可以显示相似性矩阵的近似水平取决于丢弃的特征值的绝对大小。 因此，期望负特征值的绝对量值小。 另一种方法是将S调整为半正有限矩阵S + αI，其中α > 0是最负的特征值的大小。 这种方法仅限于增加数据点的自相似性（规范）而不改变点间相似性。

第二种方法应该用于不能使用现成内核的情况，并且相似性函数是依赖于域的。 然而，特征向量的下降确实存在一些缺点，即可能会错过一些异常值。

3.4 一类支持向量机

一类支持向量机（SVM）可以被视为线性和逻辑回归模型的变体，其中边际的概念用于避免过度拟合，就像在回归模型中使用正则化一样。 这些误差惩罚是用松弛变量的概念计算的。 尽管可以使用平方损失（如回归），但在支持向量机中使用其他形式的损失函数（例如铰链损失）更为常见。 本节假设您熟悉支持向量机以进行分类。 不知情的读者可以参考[33]来了解支持向量机的基础知识。

使用支持向量机进行异常值检测的主要问题是该模型主要设计用于需要使用决策边界（及其相关的边缘超平面）分离的两个类。 但是，在异常值检测中，

singola@gmx.com

数据未被标记，因此假设所有提供的示例属于正常类，构造（可能是噪声的）模型。 为了解决这个问题，假设基于内核的变换表示的原点属于异常值类。 请注意，特定数据域中正常类和异常值之间的分离质量以及相应的转换将在很大程度上取决于该假设的有效性。

现在，我们假设使用未知函数Φ（）将数据点X 转换为Φ（X）。 这种转换不需要明确地执行，因为它是在使用内核相似性矩阵（如3.3.8节的非线性PCA方法）的优化问题中隐式执行的。 注意，Φ（X）本身是一些变换空间中的高维向量，并且Φ（X）的对应系数是向量W ，其具有与变换空间相同的维数。 将正常类与异常类分开的相应决策边界由以下给出：

$$\overline{W} \cdot \Phi(\overline{X}) - b = 0 \tag{3.20}$$

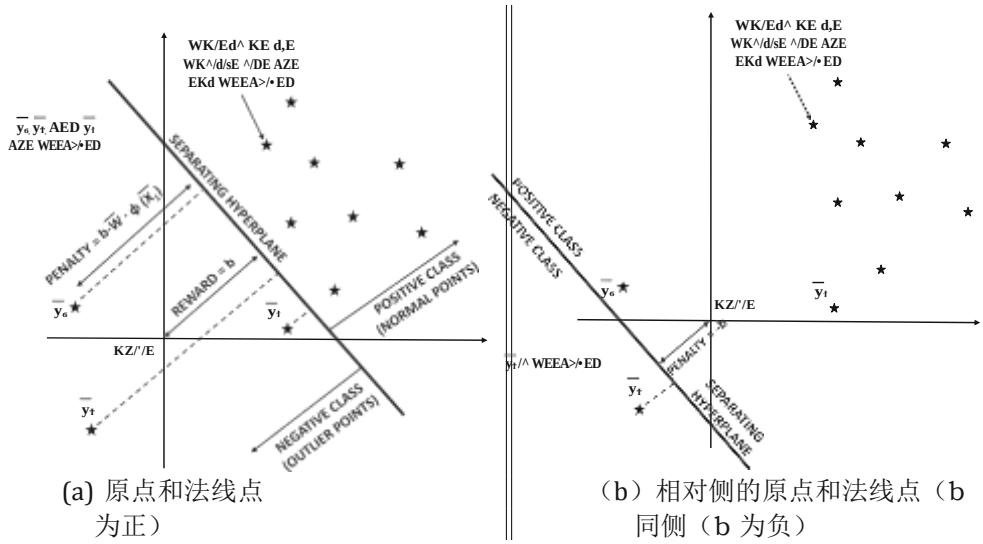
这里，b 是控制偏差的变量。 我们要制定的优化问题，这样的值W ^ Φ（X）b 为正尽可能多的训练例子越好，因为假设所有的训练样本属于正常（正）类。 因此，考虑到其中任何训练示例W Φ（X）b 是负的，我们施加的松弛惩罚最大 bW Φ（X），0。 在另一方面，原点奖励躺在该隔板的相对侧，并因此的负值W Φ（X）b 在Φ（X）= 0 的情况下是理想的。 这仅在b 为正时才有可能。 这种情况如图3.7（a）所示。 因此，为了在正常点的相反侧奖励尽可能远离分离器的原点，我们从目标函数公式中减去b 。 这具有将分离器尽可能远离原点推向正常点的效果。 此外，我们添加保证金正则项，即1 ||W ||2。 因此，总体目标函数如下：

$$\text{Minimize } J = \underbrace{\frac{1}{2}||W||^2}_{\text{Regularizer}} + \underbrace{\frac{1}{N\nu}\sum_{i=1}^{N} \max\{b - W \cdot \Phi(\overline{X}_i), 0\}}_{\text{培训数据惩罚}} - \underbrace{b}_{\text{Origin Reward}} \tag{3.21}$$

常数C> 1可以被视为与异常点相比的正常点的不同权重。 具体而言，可以将ν = 1/ C 视为训练集中的数据点是异常值的先验概率。 换句话说，C 的值调节了该模型中假阳性和假阴性之间的交易。 值得注意的是，W = 0且b = 0是具有零目标函数值的该优化问题的简并解。 因此，b 绝对不可能在最优性上严格为负，因为b的负值导致目标函数的严格正值J 。 该
具有负值b 的情况在图3.7（b）中示出，其相对于简并解决方案总是次优的。 值得注意的是，如果使用错误的[8] 种类型的核转换，其中原点不能与数据点线性分离，并且C 的值非常大，则可能导致退化解W 的不幸情况[9] = 0且b = 0。

[8]核心矩阵的均值中心可能导致退化解。 另一方面，任何非中心核矩阵（如未中心高斯核）都可以总是避免简并解，因为在变换空间中点之间的所有成对角都小于90°; 因此，原点总是线性的

singola@gmx.com



(a) 原点和法线点为正）

（b）相对侧的原点和法线点（b 同侧（b 为负）

图3.7：有用的解决方案对应于（a）中所示的情况。 对于简并解W = 0和b = 0，（b）中所示的情况总是次优的。 因此， b 在最优时永远不会是负的。 非负内核将所有点转换为具有线性可分离原点的单个orthant，因此总是存在一个非平凡的解。

内核表示中的线性分隔符的正侧对应于包含大多数点的输入表示中的任意形状的“小”区域。 处罚是由该地区以外的点造成的。 图3.7（a）的线性分隔符如何在原始图像中创建任意形状的区域的说明性示例
输入空间如图3.8所示。 注意，三个异常值X_1, X_2和X_3位于该区域之外，因为它们违反了变换空间中的线性决策边界。 增加 C 的值增加了包含内部的封闭区域的体积，也改变了其形状。

请注意，此优化公式是在未知的变换空间Φ（）中定义的。 此外，变换Φ（）通常根据点之间的成对（核）相似性间接定义。 而不是直接求解w ^ ，更常见的方法是预测的（优化的）值w ^ Φ（X）+ b 为测试点X 通过使用双制剂中的核技巧。 这是通过明确物化实现的（非负）松弛变量ξ1 ...ξN 为ñ 训练点：

$$\xi_i \geq b - \overline{W} \cdot \Phi(\overline{X_i}) \tag{3.22}$$

然后可以将该约束结合到优化公式中（连同对松弛变量的非负性约束），并且在目标函数 J 中用ξi 代替max {b - W ·Φ（Xi），0}。 这种约束公式允许基于方法
拉格朗日松弛，可以构建双重配方。 双重

可以从所有数据所在的orthant中分离出来，并且在软件执行中很少注意到这个问题。 然而，从一个级支持向量机的结果的质量倾向于是高度不可预测的[184，384]。 [9]原始论文[479]错误地指出大的训练数据处罚会导致负值
湾 事实上，这种方法对内核表示的敏感性是它的主要弱点。 一种可能退化问题的解决方案是施加约束|| W || = 1并摆脱正规化器。

singola@gmx.com

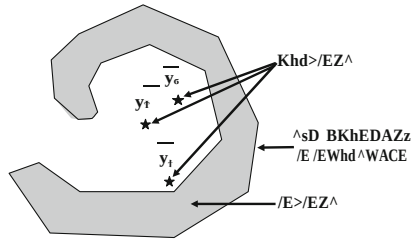


图3.8：原始输入空间中图3.7（a）的线性SVM边界的假设图示。 这只是一个说明性示例，并不反映实际计算。 封闭区域的形状对内核和参数C敏感。

制剂具有Ñ 变量α = [α1 ...αN¯]T̂，其中的每一个是对应于等式约束之一拉格朗日参数3.22 用于训练点。 有趣的是，双重公式可以用N × N 核相似性矩阵表示 S = [K（Xi，Xj）] = [Φ（Xi）Φ（Xj）]（而不是显式变换点），这是核心技巧的本质。 该相似性矩阵与3.3.8节中用于非线性PCA的相似性矩阵相同。 [480]中描述的双重表述如下：

$$\text{Minimize } \frac{1}{2} \alpha S \alpha^T$$

受制于：

$$0 \le a_i \le \frac{C}{N} \quad \forall i \in \{1 \dots N\}$$

$$\sum_{i=1}^{N} a_i = 1$$

虽然COE FFI cient矢量w ^ 未明确通过双重方法来计算，它仍然可以确定哪个边界数据点的决定的侧ý 通过使用下面的等同性，其在示出位于[128，479]：

$$\overline{W} \cdot \Phi(\overline{Y}) - b = \sum_{i=1}^{N} a_i \cdot K(\overline{Y}, \overline{X_i}) - b \tag{3.23}$$

右手侧均可使用，只要计算为α1 ...αN 和b 是已知的。 值 α1 ...αN 可以通过解决上述偶优化问题来计算

具有梯度下降（参见第3.4.1节）。 b 的值可以通过计算b =来学习 w ^ ·Φ（Ý ）=Σn 的αi · k（Y，僖）对于任何训练数据点 ý 其位于上分离超平面（即数据点是自由支持向量）。 可以显示任何数据点Xj，其中0 <αj<C / N 是自由支持向量。 我们可以平均计算所有这些数据点的b 值。

一旦训练了模型，就可以使用从公式3.23的决策边界导出的以下表达式对任何数据点X（包括未包括在训练数据中的样本外点）进行评分：

$$\text{Score}(\overline{X}) = \sum_{i=1}^{N} a_i \cdot K(\overline{X}, \overline{X_i}) - b \tag{3.24}$$

singola@gmx.com

从支持向量机的建模假设的角度来看，得分的负值表示数据点是异常值，而正值表示数据点是非离群值。 然而，人们也可以将计算值视为分数，尽管它对C的值敏感，C值调节异常值和内点之间的交易。

3.4.1 解决双重优化问题

可以使用梯度下降来解决双优化问题。 但是，在下降过程中，确保不违反约束是很重要的。 双目标函数的梯度，即1αSαT ，可以显示为N 维向量Sα。 因此，在将每个α i 初始化为1/ N 之后，梯度下降方法迭代地重复以下步骤：

1. $\overline{a} \Leftarrow \overline{a} - \eta \cdot S\overline{a}$
2. 设置在任何负值α 为0和的任何值的αi 大于C / N 到C / N 。
3. 缩放矢量α ，使得$\Sigma^{\bar{N}}_{i=1}$ 的αi = 1。

这里，η> 0是学习率。 重复这些步骤以收敛。 最后两个步骤被执行以迫使电流溶液至（大约）满足优化约束虽然约束可能不完全SATIS音响版在早期的迭代，因为在执行步骤3可能会再次导致的αi 超过C / N 。 在收敛时，通常会满足所有约束条件。 更多有关SVM培训的有效步骤可以在[128]中找到。

许多现成的库，如LIBSVM [128]和scikit-learn [629]提供了一流的支持向量机的实现。 scikit-learn代码提供了不同的接口，可以调用LIBSVM解算器。 值得注意的是，LIBSVM使用更复杂的坐标下降方法而不是上述简化的梯度下降方法。 与内核PCA方法一样，一个类支持向量机具有理想的属性，只要可以在数据对象之间定义内核相似性函数，它们就可以用于任意数据类型。

3.4.2 实际问题

[384]中提供了对文档数据中异常值检测的一些模型的深入评估。 特别是，与支持向量机相关的主要挑战是这些方法可能对内核的选择和与该方法相关的许多隐藏参数敏感，例如C 的值和与内核相关的参数。 [384]中的评估对一类支持向量机进行了以下观察：

> “然而，事实证明，对于特定的代表选择和内核以非常透明的方式非常敏感。 例如，该方法最适合二进制表示，而不是已知在其他方法中更优越的tf-idf或Hadamard表示。 此外，正确选择内核取决于二元向量中的特征数量。 由于基于这些选择，性能的差异是非常显着的，这意味着如果没有对这些表示问题的更深入理解，该方法就不健全。 “

singola@gmx.com

在最近对几个探测器进行的另一项实验评估[184]中，单级SVM是表现最差的探测器，并且经常提供比随机性差更差的探测器。

单类SVM假定原点是异常类的先验，即使在内核特征空间中也不是最优的。 例如，一个简单的操作，例如对核矩阵进行均值定中可能会产生不幸的影响。 作为一般规则，由于模型选择和参数调整中的自然困难，在无监督的问题设置中（与监督设置相比）更难使用内核。 例如，对于高斯核，点之间的距离成对的中值提供粗略近似的带宽σ，虽然值σ 也是对数据分布和尺寸敏感。 第二个问题是内核矩阵S的大小，它也决定了术语的数量在双目标函数中，是O（N 2）。 例如，对于包含十万个点的数据集，这种方法是不实际的。

使用变量子采样[32] 可以部分地解决这两个问题，这在不可预测的数据大小敏感参数的环境中特别有效。 基本思想是从数据中重复采样可变数量的训练点，这些训练点在nmin = 50和 nmax = 1000 之间变化。 因此，在每个训练模型中，核矩阵的大小最多为1 000 1000.随后，所有N根据该训练模型对得分进行评分，并将得分归一化为Z值。 这在内核支持向量机中是可能的，因为可以X使用学习的模型对样本外点进行评分。 事实上，由于过度配置较少，样本外的分数通常会更加稳健。 在从样本构建的每个训练模型中，可以使用不同的核函数和参数选择（在合理范围内）。 该过程可以根据需要重复多次。 最终的异常值得分可以报告为各种检测器的得分平均值。 这种方法是一种集合技术，将在第6章中详细讨论。

3.4.3 支持向量数据描述和其他内核模型的连接

单类SVM与许多其他内核模型密切相关。 支持向量数据描述（SVDD）[539]是一种不同的核SVM方法，其中数据被包含在变换特征空间中的半径为R 的超球面中（而不是来自原点的线性分离器）。 平方半径最小化，并且对违反保证金的处罚也是如此。 SVDD方法与3.4节的单类SVM 以及3.3.8节中讨论的内核Mahalanobis方法密切相关。

如果嵌入点具有以原点为中心的球形几何，则SVDD和线性SVM模型大约相当于[10] [539]。 最常见的例子是高斯核，它嵌入单位球上的所有点。 否则就是解决方案非常不同。 例如，即使使用以中心为中心的内核，在SVDD中也不会遇到3.4节中讨论的单类SVM的简并问题。 事实上，SVDD预测不会因核心矩阵的中心而发生变化，因为原点不是先验的。 另一方面，SVDD方法对多项式内核表现不佳，因为数据往往在特定方向上延长。

[10]我们用的术语“大致”，是因为SVDD优化制剂（与高斯内核）可以转化为像一个类SVM的制剂，其中的归一化约束W = 1被施加，而不是包括正则w ^ 2/ 2- 目 标 函 数[539]。 [184]中的实验结果表明，与具有高斯核的单类SVM相比，SVDD提供了稍好的解。

singola@gmx.com

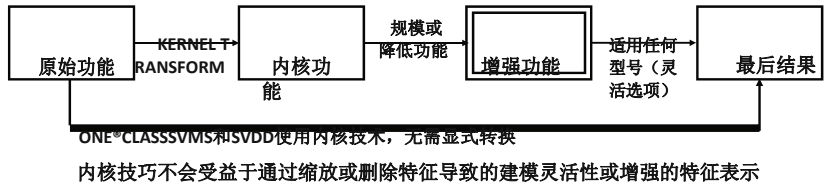


图3.9：显式和隐式内核转换之间的关系

改变了特征空间，并且围绕它的球体很差。

内核Mahalanobis方法与SVDD密切相关。 前者可被视为创建数据的新表示的一种方式，其中到变换数据的中心的距离直接提供异常值得分，并且不需要找到特定半径R 的硬边界。 使用优化模型。 内核Mahalanobis方法与其他一类方法之间的关键差异是使用所有变换维度的特征空间缩放到前者的单位方差。 这种类型的标准化特别有用，因为它可以防止高方差方向掩盖异常值。 归一化步骤可以提供额外的好处，这是基于双重内核技巧的原始单类SVM或SVDD解决方案中不可用的。 例如，可以使用3.3.8 节中的方法，非常轻松地将数据转换为特征空间中的单位方差方向然后在嵌入式表示中应用线性一类SVM以获得这些额外的好处。 实际上，在规范化的内核表示上使用圆形分隔符（如SVDD）对于多项式内核工作得相当好，多项式内核通常已知与SVDD一起工作不佳。 将3.3.8 节中的非线性变换视为（标准化的）数据特定的Mercer核映射是有帮助的，在此基础上许多简单的简单异常检测模型（例如，距质心的距离）变得极其有效。 此外，人们可以获得在提取的表示上使用任何其他离群值检测方法的灵活性。 例如，可以使用混合建模，基于距离的方法[475]或聚类（参见第4.2.1 节）第4章）关于内核表示。 尽管使用其他模型具有明显的灵活性，或者通过特征缩放/选择来增强底层表示，但是使用内核的显式特征映射非常不受重视。 例如，滴平小的特征向量导致聚类和二类别支持向量机[更好的精度33，481]，因为去噪ËFF学分的。 同样，由于异常增强效应，丢弃大的特征向量有时会导致单类SVM的更高精度; 这只是更柔和的功能缩放的硬版本。 此外，对于显式特征映射的样本外实现，无法保证使用对偶的计算优势（参见方程式）3.19）; 显式转换的唯一“缺点”是，一旦用于创建嵌入，它就会使可爱的内核技巧在SVM公式中变得冗余。 显式和隐式内核转换方法之间的关系如图3.9 所示。

单类支持向量机也可以被视为（粗略地）在内核特征空间中找到数据矩阵D 与其负集D （而不是将D 与原点分开）的最大余量分离的方法。 这个直观的视图提供了对两类SVM关系的更清晰的理解，它还有助于其他两类优化模型的适应，例如Fisher的线性判别式到一类设置。 该视图已被用于使核Fisher判别方法适应

singola@gmx.com

异常检测[466]。 核Fisher方法[466]与核SVM 之间的主要差异在于，前者确定了一个方向，它最大化了类间分离与类内分离的比率（在D 和 D之间 ），而后者确定了最大边际分离器在D 和D之间。 但是，两种方法在优化之前隐式地将这些数据转换为相同的内核特征表示。

在所有这些模型中，内核Mahalanobis方法的优点是需要最少数量的自由参数（仅对应于内核设置）。 其他方法试图在异常值和内点之间找到一个硬边界（参见图3.8），因此必须使用像C 这样的参数在建模早期规范异常值和内点之间的交易。 Mahalanobis方法是一种软方法，专注于发现得分和（适当地）将硬标签留到最后，当有更多关于得分分布的见解时。 在诸如异常检测之类的无监督问题中总是希望最小化用户驱动参数的数量，尤其是因为底层内核特征表示在语义上是不透明的。 在集合中心设置中对内核Mahalanobis方法的评估可以在[35]中找到。

3.5 线性模型的矩阵分解视图

PCA可以被视为一种矩阵分解。 为了理解这一点，从平均中心数据矩阵D考虑PCA 的秩k 表示。

$$D' = DP_k \tag{3.25}$$

在这里，d'是Ñk 缩减了的矩阵的表示d，和PK 是DK 含有最大矩阵k 的协方差矩阵Σ= 的（正交）的特征向量d^T d 在其列。
如前所述，d'仍然对应于Qk的Λk，其中所述秩k 点积的对角化（即，相似性）基质DDT 由下式给出Qk的Λ2 QT 。 这里，Qk 是N × k 矩阵 包含DDT 的标准正交特征向量。 因此，我们有：

$$D' = DP_k \approx Q_k\Lambda_k \tag{3.26}$$

有趣的是，它可以表明矩阵Qk的，Λk 和PK 可用于创建一个秩k 原始矩阵的因子分解d。 通过将公式3.26中的每个表达式与矩阵PT 相乘并设置PkPT = I，我们得到：

$$D \approx Q_k\Lambda_k P_k^T \tag{3.27}$$

该特定关系也称为数据矩阵D的秩k 奇异值分解（SVD）。 通过在一个因子中吸收对角矩阵，可以将D 表示为分解成两个矩阵，每个矩阵具有正交列。 特别是，我们去音响NE的Ñk 矩阵ü 作为Qk的Λk 和DK 矩阵V 作为PK。 因此，SVD可以表示为两个低秩矩阵的因子分解如下：

$$D \approx UV^T \tag{3.28}$$

有趣的是，矩阵U 和V 可以通过解决以下优化问题来学习：

$$\text{Minimize } ||D - UV^T||^2$$

受制于：

U的列是相互正交的V的列是相互正交的

singola@gmx.com

这里，$\|\cdot\|_F$表示误差矩阵（D UV T ）的弗罗贝尼乌斯范数。 Frobenius范数被定义为条目的平方和。 请注意，这种解释是
与PCA的均方误差优化一致。 值得注意的是，D UV T 的条目的绝对值提供了入口异常值分数，其比前面部分中讨论的行式异常分数更详细。 这些值表示压缩表示UV T 无法完全解释原始矩阵中的值，因为它们与基础相关结构的偏差。

这种SVD视图特别有用，因为它为使用其他类型的矩阵分解方法铺平了道路。 例如，可以以各种方式改变目标函数，可以添加正则化，并且可以修改约束以将特定（期望的）属性结合到因子矩阵U 和V中。 最简单的概括是去除U 和V 的正交性约束的概括;这导致无约束的矩阵分解。 或者，对于非负数据矩阵D.，我们可以对因素施加非负面约束，以创造更具可解释性的非负面表达。 这种分解在文本数据和网络数据中非常有用。 在第12章中，我们将提供这种分解的具体示例。 最后，这种类型的矩阵分解的最有用的方面是它可以用于不完整数据集中的异常检测，甚至提供关于导致异常的矩阵的特定条目的见解。 对于不完整的数据矩阵，PCA的简单版本无法做到这一点。 在下一节中，我们将提供不完整数据的无约束矩阵分解的示例。

3.5.1 不完整数据中的异常值检测

在本节中，我们将展示如何使用无约束矩阵分解来发现不完整数据矩阵的异常行或甚至异常条目。 后者可用于诸如推荐系统的应用中。 在推荐系统，我们可能有一个ñd 评级矩阵d 与ñ 用户和d项目。 可能需要发现异常评级，例如由推荐系统中的“shills”创建的虚假评级。 在此类应用程序中，每个列（项目）通常可能在数百万用户中包含少于100个评级。 在这种稀疏设置中，甚至协方差矩阵也不能根据PCA的需要准确估计。 在其他应用程序中，由于数据收集机制存在缺陷，许多数据值可能会丢失。 例如，在用户调查中，用户可能会选择离开许多现场银行。 在这样的设置中，可能希望发现数据矩阵的异常行。 所有这些复杂的设置都可以通过上述PCA模型的简单推广来解决。

为了以下讨论的目的，我们将假设D的第（i，j）个条目（当没有丢失时）由xij表示。 考虑一种设置，其中矩阵中指定（即，不缺失）的条目集由H表示。 换句话说，我们有以下内容：

$$H = \{(i,j) : x_{ij} \text{ is observed (not missing)}\} \tag{3.29}$$

然后，我们想将数据矩阵D 分解为表示UV T ，以便
U = [uis]是N ×k 矩阵，V = [vjs]是d×k 矩阵，k 是分解的秩。
一个条目（的预测值I，J）是$\Sigma^{k}_{s=1}$ uisvjs，我们希望将聚合最小化
关于观测值的误差。 但是，由于我们只知道数据矩阵中的条目H 的子集，因此我们必须仅针对指定的条目来制定优化问题。 此外，重要的是使用正则化来避免过度配置，因为指定条目的数量可能很小。 可以编写此优化问题（与...一起）

singola@gmx.com

正规化条款）如下：

$$\text{Minimize } J = \frac{1}{2} \sum_{(i,j) \in H} (x_{ij} - \sum_{k} u_{is} v_{js})^2 + \frac{a}{2} (||U||^2 + ||V||^2)$$

观察到的条目出错 Regularizer

受制于：

U 和V没有限制

这里，α > 0是正则化参数，因子矩阵上的平方Frobenius范数包含在目标函数中。 我们已经放弃了对U 和V 的约束来简化优化过程。 对于D中的观察条目，（i，j）的误差eij 条目是所观察到的值之间的二FF erence xij 和预测值$\Sigma^{k}_{s=1} u_{is} v_{js}$:

$$e_{ij} = x_{ij} - \sum_{s=1}^{k} u_{is} v_{js} \tag{3.30}$$

请注意，误差项仅针对H中观察到的条目定义。 因此，我们可以如下重写目标函数J ：

$$J = \frac{1}{2} \sum_{(i,j) \in H} e_{ij}^2 + \frac{a}{2} (||U||^2 + ||V||^2) \tag{3.31}$$

这个错误术语是关键; 求解优化问题得到（平方）误差项作为数据矩阵中各个条目的异常值。

可以使用梯度下降来解决优化问题。 因此，我们可以根据参数uis 和v_{js}计算目标函数的偏导数：

$$\frac{\partial J}{\partial u_{is}} = \sum_{j:(i,j) \in H} e_{ij}(-v_{js}) + a \cdot u_{is}$$

$$\frac{\partial J}{\partial v_{js}} = \sum_{i:(i,j) \in H} e_{ij}(-u_{is}) + a \cdot v_{js}$$

在梯度下降，我们创建了一个[（Ñ + d）k]维向量w ^ 的参数心病响应中的条目ù 和V 并进行更新w ^ $\Leftarrow$ w $\triangleq$ $\eta \nabla J$ 。 注意，J ∇ 由偏导数的（N + d）k维向量的整个向量定义计算如上。 可以用稀疏矩阵乘法等效地执行这些更新。 令E 为N × d 稀疏矩阵，其中仅指定的条目（即，H中的条目）采用eij的值。 缺少的条目采用零值。 请注意，这是有道理的
仅计算在所观察到的条目Ë 并使用稀疏数据结构来表示Ë。 上述梯度下降步骤可以显示为等效于以下矩阵式更新：

$$U \Leftarrow U(1 - a \cdot \eta) + \eta EV$$

$$V \Leftarrow V(1 - a \cdot \eta) + \eta E^T$$

U

singola@gmx.com

可以执行这些迭代以进行收敛。 η > 0 的值对应于学习速率。 选择学习率太大可能会导致过度流动。 另一方面，非常小的学习率将导致收敛太慢。 这种方法的一个变种速度是随机梯度下降，其中通过观察条目，我们循环xij 在^ h 随机顺序进行以下更新：

$$\text{uis} \Leftarrow \text{uis}(1 - \alpha \cdot \eta) + \eta\, \text{eijvjs} \quad \forall s \in \{1 \ldots k\}$$
$$\text{vjs} \Leftarrow \text{vjs}(1 - \alpha \cdot \eta) + \eta\, \text{eijuis} \quad \forall s \in \{1 \ldots k\}$$

产生的迭代也被执行以收敛。 更多细节可以在[34]中找到。

3.5.1.1 计算离群值

它仍然表明如何根据因素计算入口或行出异常值。 注意，近似的因式分解重建原始数据矩阵（如在SVD中）。 假设与重建值的偏差是异常值。 观察到的条目xij 的误差eij 根据公式3.30定义：

$$e_{ij} = x_{ij} - \sum_{s=1}^{k} u_{is} v_{js} \tag{3.32}$$

这些错误也称为残差。 具有大的正或负残差的条目往往是异常值，因为它们不符合低秩分解的正常模型。 因此，我们使用这些条目的平方作为异常值得分。

我们可以用类似的方式定义行式异常值得分。 对于数据矩阵的任何特定行，其异常值得分被定义为其观察到的条目中的平方残差的平均值。 因此，对于我第i行玺的d，我们去科幻NE的异常分数如下：

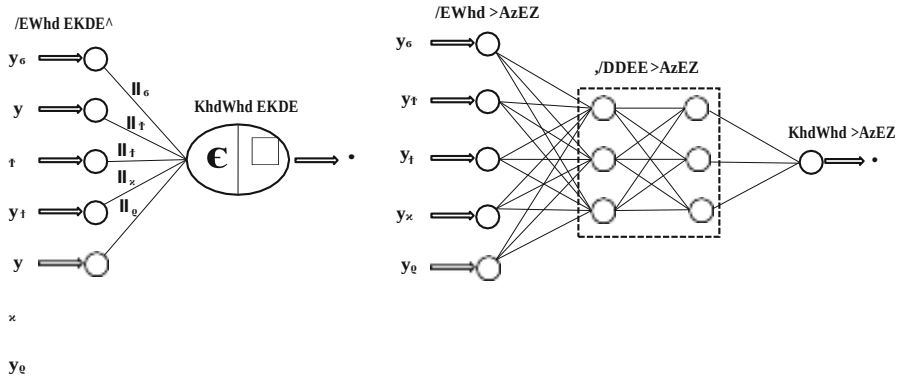
$$\text{Score}(\bar{X}_i) = \frac{\sum_{j:(i,j) \in H} e_{ij}^2}{n_i} \tag{3.33}$$

这里，ni 表示Xi中观察到的条目数。 有趣的是，它可以定义
列式异常值以完全相似的方式得分。 这在某些应用中很有用。 例如，在协作过滤应用程序中，商家可能对具有不寻常评级模式的项目（列）感兴趣。 矩阵分解是最常见的线性建模形式之一，它也可以在网络数据中找到应用。 其中一些应用将在第12章中讨论。

3.6 神经网络：从线性模型到深度模型学习

神经网络是模拟人类神经系统的计算学习模型。 在人类中，通过改变细胞之间的突触连接的强度来进行学习，所述细胞被称为神经元。 突触连接的强度根据人工刺激而改变。 在人工神经网络的情况下，各个节点被称为神经元。 神经元使用加权连接（或来自外部训练数据）从其他神经元接收输入，并且可以在对其输入执行计算之后将其输出传输到其他神经元。 就像在生物神经网络中学习，人工神经网络中的突触强度一样

singola@gmx.com



(a) 感知器 （b）多层图3.1

0：单层和多层神经网络

网络根据其输入改变其权重。 对神经元的最基本输入形式是训练数据的特征，其可以被认为是生物神经网络中使用的外部刺激的类比。

最简单的神经网络形式是感知器。 在最基本的形式中，感知器几乎与简单的线性回归模型或PCA /矩阵归因模型相同。 然而，它作为计算单元的整洁概念化允许我们将多个感知器放在多层网络中。 这种多层网络允许计算任何非线性函数。 出于这个原因，神经网络被称为通用函数逼近器。 感知器包含两层节点，对应于输入节点和单个输出节点。 in-的数量
put节点完全等于数据的维数d 。 设X =（x1 ... xd）为
d 输入，对应于数据记录的d 特征值。 一个输出
感知器被计算为基础输入的函数，其相关权重向量W =（w1 ... wd）：

$$z = \overline{W} \cdot \overline{X} = \sum_{i=1}^{d} w_i x_i \tag{3.34}$$

该功能也称为线性激活功能。 值得注意的是，激活函数中的权重与3.2 节中用于线性回归和秩-1 PCA的权重完全类似。 但是，在一般设置中，我们可以使用此线性模型的任意激活函数，并且还包含偏差项：

$$z = \Phi(\overline{W} \cdot \overline{X} + b) \tag{3.35}$$

正如我们稍后将讨论的，激活函数Φ通常是非线性函数，如sigmoid或tanh函数。 但是，就目前而言，我们将使用3.34 节中讨论的更简单的形式，因为它与本章讨论的其他模型相似。 感知器结构的图形表示如图3.10（a）所示。

神经网络可用于3.2节中讨论的两种设置中的任何一种; 回想一下，在一个设置中（参见第3.2.1节），因变量被视为特殊变量，而在另一个设置中（参见第3.2.2节），所有属性都以均匀方式处理，均方投影误差在超平面被最小化。 设置的第一个类型被用来创建复制器神经网络[160，250]，其中所述数据的各属性，使用其他属性预测，并且该预测的误差被用于量化离群值的分数。 这种类型的神经网络也称为自动编码器，将在3.6.2节中讨论。 第7章第7.7 节讨论了使用任何回归模型（而不仅仅是神经网络）的这种方法的概括。

singola@gmx.com

在本节中，我们将关注两种不同类型的神经网络。 首先，我们将关注一个我们试图创建一类神经网络的环境，其中尽管权重非零，网络的输出总是为零。 此设置不太常见，很少使用。 但是，我们将其包含在内，因为它与3.2.2节中讨论的优化方法直接相关。 在3.6.2节中，我们将讨论使用复制器神经网络和自动编码器进行离群检测。 尽管自动编码器框架更通用且可能更强大，但是可以在维数降低方法的框架内理解这两种类型的方法。 这种方法在异常值检测中更常用。

由于假设所有训练点都是单级设置中的法线点，因此公式3.34 的预测z 预计为0.请注意，这与线性建模的公式3.9中的假设完全相同，此处复制：

$$\sum_{i=1}^{d} w_i \cdot x_i \approx 0 \tag{3.36}$$

因此，假设由一类神经网络预测的z的任何非零值都是不符合正常数据模型的异常值的结果。 因此，对于单个实例Xi，其中神经网络预测zi，来自我们的一类假设的第i个点的平方误差如下：

$$J_i = z_i^2 = (\overline{W} \cdot \overline{X_i})^2 \tag{3.37}$$

因此，必须更新神经网络的权重以解决此错误。 这是通过梯度下降更新实现的。 此更新可能写成如下：

$$\overline{W} \Leftarrow \overline{W} - \eta \nabla J_i$$
$$= \overline{W} - \eta z_i \overline{X_i}$$

这里，η > 0是学习率。 而且，为了避免平凡解W = 0，更新的矢量W 被缩放到单位范数。 神经网络的训练阶段将d维记录X1 ... XN 逐个馈送到感知器，并对矢量W 执行上述更新，直到达到收敛。 整个过程是一个版本
随机梯度下降，结果与我们用PCA（参见第3.2.2节）或矩阵分解（参见第3.5节）得到的解决方案几乎相同，其等级设置为（d -1）。

为了对给定的数据点Xi进行评分，我们使用学习的模型来计算其异常值得分如下：

$$\text{Score}(\overline{X_i}) = (\overline{W} \cdot \overline{X_i})^2 \tag{3.38}$$

异常值将有更高的分数。 可以使用此模型对样本外点进行评分。 事实上，正如后面第3.6.3节所讨论的，样本外点数将具有更强大的异常值，因为它们不会超过训练数据。 感知器等效于使用具有秩（d 1）的矩阵分解或PCA，这相当于在将数据投影到顶部（d 1）主成分的空间中之后仅沿着最小特征向量对数据进行评分。 使用这种保守的评分方法，如果数据集的排名严格小于d，则所有点都可能具有0的异常值。 结果，许多真正的异常值将被遗漏。 这是过度配置的经典表现。 因此，更合理的方法是仅使用一半点训练神经网络，然后将剩余的一半评分为样本外测试点。 测试点的分数是

singola@gmx.com

标准化为零均值和单位方差。 在多个随机样本上重复该过程多次，并对得分进行平均。 稍后将讨论使用多个输出节点的替代方法。

3.6.1 非线性模型的推广

到目前为止，似乎没有一类神经网络能够实现与我们能够通过矩阵分解或PCA的特殊情况完成的任何不同的东西。 那么，整个练习的重点是什么？主要的一点是，将这些模型概念化为神经网络单元有助于我们将它们组合在一个多层神经网络架构中，该架构可以模拟底层数据中任意复杂的模式。 换句话说，感知器的概念化提供了一个黑盒框架，用于将这些更简单的模型“组合”到更复杂的模型中。 事实上，神经网络的普遍性甚至比3.3.8 节中讨论的非线性PCA方法还要大就这种技术可以建模的决策边界的类型而言。 从技术上讲，给定足够的数据，具有适度数量单位的多层神经网络几乎可以模拟任何一类数据分布，而无需对该分布的形状做出任何假设。 因此，神经网络有时也被称为“通用函数逼近器”。

除输入和输出层外，多层神经网络还有一个额外的隐藏层。 隐藏层本身可能以不同类型的拓扑连接。 常见类型的拓扑是其中隐藏层具有多个层，并且一个层中的节点向前馈送到下一层的节点中。 这被称为前馈网络。 图3.10（b）显示了具有两个隐藏层的前馈网络示例。 此外，不需要在任何层中使用线性函数。 使用各种激活函数Φ（）（基于等式3.35），例如tanh和sigmoid函数。

$$\Phi(z) = \frac{e^{2z} - 1}{e^{2z} + 1} \text{ (tanh function)}$$

$$\Phi(z) = \frac{1}{1 + e^{-z}} \text{ （sigmoid功能）}$$

$$\Phi(z) = \exp\left[-\frac{(z-\mu)^2}{2\sigma^2}\right] \text{ （高斯径向基函数）}$$

在单层神经网络中，训练过程很简单，因为已知神经元的预期输出为0.隐藏层的问题是我们不知道这些单元中神经元的输出应该是什么是。 我们只知道输出层的最终输出应为0.换句话说，从后期层到早期层需要某种类型的反馈来预期输出。 这是通过使用反向传播算法实现的。 反向传播算法具有前向阶段和后向阶段，其在每个实例的训练期间应用。 在前向阶段，激活功能首先应用于输入层，然后应用于隐藏层，直到输出传播到输出层。 然后计算错误，并且各种神经元的误差估计也向后传播。 然后将它们用于更新各种节点的权重。 与感知器的情况一样，在整个更新过程中将输出节点的权重缩放到单位范数是很重要的。 这样做是为了避免输出节点中所有权重都为0的简单解决方案。

此外，如果W 是任何特定隐藏对之间的权重的$h_1 \times h_2$矩阵

singola@gmx.com

在具有h_1和h_2节点（其中$h_2 \leq h_1$）的层中，我们需要施加约束$W^T W = I$.
避免隐藏层中每个节点对应的平凡变换
相同的转换或零输出。 这些额外的要求需要使用受约束的梯度下降方法，这通常要困难得多。

与感知器的情况一样，该过程适用于每个训练数据点；此外，我们循环遍历各种训练数据点，直到达到收敛。 [84]中提供了反向传播算法的详细讨论。 通过增加网络中的层数并使用不同类型的激活函数，我们可以模拟任意复杂的非线性模式。 学习具有大量层的神经网络的参数的过程需要许多专门的技巧;这类方法被称为深度学习[223]。

使用多个输出降低表示级别： 上述方法仅使用一个输出节点。 在感知器的情况下，这对应于等级（$d - 1$）的线性减少。 这相当于仅使用PCA中最小特征向量的分数。 在实践中，这是一个过大的表示级别，以获得有意义的减少。 减少表示等级的一种方法是使用多个（比如说
r）输出节点，以便每个节点的输出预期为零。 这种情况类似于沿着PCA中的r个最小特征向量进行评分（即，使用（$d-r$）的表示等级）。 因此，误差等于各个节点的输出的平方和。 此外，我们需要在输出层中对权重进行一些约束。 设W 是输出层中权重的h_r 矩阵，其中h_r 是最后一个隐藏层中的节点数。 为了确保权重向量的相互正交性和归一化，我们需要实施加附加约束$W^T W = I$.。 这种方法将需要输出层中权重的约束梯度下降。 这使得优化问题更具挑战性。 实现这些目标的更令人满意和无法解释的方法是使用复制器神经网络，这将在下一节中讨论。

异常值得分： 对于任何给定点X_i，如果多层网络具有单个输出节点输出z_i，异常值得分为z_i^2.如果神经网络有r输出$z_i(1) \ldots z_i(r)$，离群值得分是$\sum_{j=1}^{r} z_i(j)^2$。 请注意，这些分数的构造非常相似
方法到矩阵分解和PCA中使用的方法。

3.6.2 复制器神经网络和深度自动编码器

虽然上一节中的方法因其与传统神经网络中的监督学习的自然关系而具有直观的吸引力，但很少使用。 这是因为底层优化问题中的约束以及它不能用于导出数据的压缩表示的事实。 更常见的方法是使用自动编码器。 具有三个隐藏层的自动编码器示例如图3.11所示。 请注意，输出数量与输入数量相同
对于第i维，输入x_i 被重建为$\hat{x}_i$。 重建的总误差
$\sum_{i=1}^{d} (x_i - \hat{x}_i)^2$所有$d$ 维上的2对所有数据点求和，并最小化
神经网络训练。 重建的点SPECI音响C的误差，这是$\sum_{i=1}^{d} (X_i - \hat{X}_i)^2$，
提供该点的异常值。 在复制器神经中使用三个隐藏层
网络很常见，也用于[250]中讨论的方法。

自动编码器是离群值检测的自然选择，因为它们通常用于降低多维数据集的维数，作为PCA或矩阵分解的替代方案。 请注意，图3.11 中间隐藏层中的节点数远小于输入层中的节点数（这是非常典型的），并且

singola@gmx.com

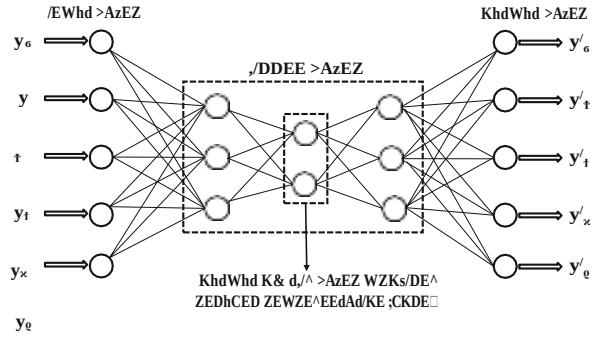


图3.11：自动编码器架构：将自动编码器架构与图3.10（b）中的自动编码器架构进行比较是有益的。

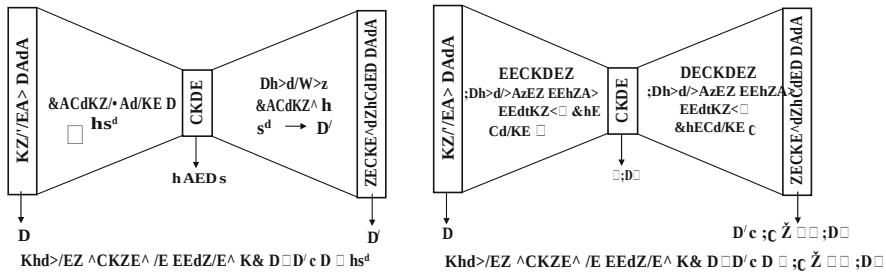
中间隐藏层中的节点的输出可以被视为数据的简化表示。 降维利用神经网络的在[讨论259，264]。 在这种情况下，已经表明[68]自动编码器的某些简化架构产生的维数减少与使用主成分分析获得的维数减少密切相关。 注意，中间隐藏层两侧的神经网络的体系结构通常（但不一定）是对称的。 实际上，可以将中间隐藏层两侧的自动编码器分成对应于编码器和解码器的两个部分，这提供了与矩阵分解和PCA非常相似的视点。

为了理解自动编码器为何对异常值检测有效，我们首先将传统的矩阵分解理解为一种编码和解码数据的方法。 分解D UV T 可以看作是一种压缩编码器的编码器数据矩阵d 成低秩因子ù 和V 。 此外，该产品UVŤ 提供重构的矩阵d^{j} = UVŤ 。 因此，U 和VT 的乘法可以是被视为一种解码器。 请注意，D^{j}与D 和绝对值不完全相同（DD^{j}）中的条目值提供了入口异常值分数。 矩阵分解的整个过程（在编码器 ¯ 解码器架构的背景下）在图中示出

Figure 3.12(a).

当使用PCA或矩阵分解时，人们假设线性压缩。 然而，多层神经网络架构提供了更一般类型的维数降低，其中使用神经网络模型可以进行任何类型的非线性降低。 实际上，通过在中间隐藏层的两侧分割图3.11 的复制器神经网络，我们在对应于编码器和解码器部分的中间隐藏层的每一侧获得多层神经网络。 这种情况如图3.12（b）所示。 网络的第一部分学习编码函数φ，神经网络的第二部分学习解码函数ψ。 因此，φ（D）表示数据集D的压缩表示（即，维数减少），并且将解码器函数ψ 应用于φ（D）产生重构数据D^{j} =（ψoφ）（D），它可能与D不完全相同。 异常值条目抵抗压缩，并且将显示D和D^{j}之间的最大变化。 残差矩阵（DD^{j}）中的条目的绝对值提供入口异常值分数。 与矩阵分解的情况一样，可以将这些入门分数转换为行方式分数或列分数。 矩阵分解的主要差异是自动编码器在表示任意数据分布方面的更大功率。 数量越多

singola@gmx.com



(a) 矩阵分解 （b）自动编码器

图3.12：自动编码器与降维/矩阵因子化方法之间的相似性

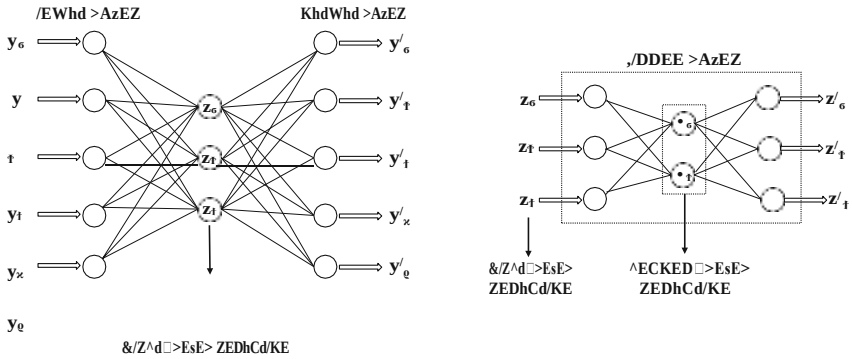
对于隐藏单元，可以获得任何数据分布的低维表示，无论它是否是线性的。 实际上，人们通常可以使用自动编码器来模拟比使用内核PCA更复杂的分布。

尽管将复制器神经网络分成编码器和解码器对于理解与维数降低的关系是有用的，但在实践中并不是必需的。 将复制器神经网络分成编码器 - 解码器对是允许创建压缩表示的特定类型的架构（称为n / p / n架构）。 如果没有明确需要减少数据的表示，则可以使用更通用的体系结构。 在复制器神经网络的原始工作[250]，使用了三个隐藏层。 tanh激活函数用于隐藏层，线性或sigmoid函数用于输出层。 中间隐藏层使用tanh函数的逐步变化。 输出层中的误差由重建误差给出，并且这些误差的反向传播用于训练神经网络。 方法的一般性和过度趋势之间的交易由隐藏层的数量和每个隐藏层中的单元数量来调节。 隐藏层的数量和每层中隐藏节点的数量可以通过使用验证集[160]凭经验确定]。 验证集对于确定培训过程的终止标准也很有用。 当验证集上的错误开始上升时，训练阶段终止。

该方法的成功源于其对复杂非线性分布进行建模的能力，尽管必须始终防止过度增加隐藏层的数量并导致过度配置。 如[264]所示，仔细的设计选择可以提供比具有神经网络的PCA更好的减少。 特别地，具有非常深的网络中，无监督方法中，被称为预训练[工作时80，459]对于实现有意义的维数减少至关重要。 在预训练中，通过首先学习外部隐藏层的权重然后学习内部隐藏层的权重，使用贪婪方法一次一层地训练网络。 得到的权重用作传统神经网络反向传播的最终阶段的起点，以便对它们进行精细调整。

训练前为图的网络的一个例子3.11 示于图3.13。 基本思想是假设两个（对称的）外部隐藏层包含较大维度的第一级缩减表示，而内部隐藏层包含较小维度的第二级缩减表示。 因此，第一步是使用图3.13（a）的简化网络学习与外部隐藏层相关的第一级缩减表示和相应的权重。 在这个网络中，

singola@gmx.com



（a）预训练第一级减少

（b）预训练第二级减少和外部重量和内部重量

图3.13：预训练图3.11的神经网络。

缺少中间隐藏层，两个对称的外隐藏层折叠为单个隐藏层。 假设两个外部隐藏层以对称方式彼此相关，如较小的复制器神经网络。 在第二步中，第一步中的缩减表示用于学习内隐藏层的二级缩减表示（和权重）。 因此，神经网络的内部部分本身被视为较小的复制器神经网络。 由于这些预训练子网中的每一个都小得多，因此可以在不过多配置的情况下学习权重。 然后使用该初始权重集来训练图3.11 的整个神经网络带有反向传播。 注意，对于包含任意数量隐藏层的深度神经网络，可以以分层方式执行该过程。 值得注意的是，对称自动编码器自然地构建了不同压缩级别的分层相关维数减少。

在没有预训练的情况下，该方法有可能获得微不足道的减少，因此重建总是返回训练数据的平均值[264]。 这是因为深层网络的大参数空间过度配置。 从参数优化的角度来看，可以将过度配置视为陷入局部最小值的过程。 通过选择良好的初始化点，预训练有助于将参数调节到更具吸引力的局部最小值。 这些技术是深度学习的最新成果，只能被描述为突破性进展[186]。 例如，它已在[264]可以使用深度自动编码器将784像素图像转换为仅6个实数。 PCA无法做到这一点。 正如深度自动编码器提供比矩阵分解和PCA更好的重建[264]，这种方法也可能更准确地用于异常检测。 在[本发明提供一种在数字二FF erent域神经网络的二FF erent实现的147，160，249，552]。

3.6.3 实际问题

使用神经网络存在两个问题。 第一个问题是神经网络训练缓慢。 由于计算复杂性是神经网络的固有问题，因此很难解决这个问题。 尽管如此，最近的硬件和算法进步使得神经网络（一般而言）和深度学习（特别是）更加可行。

神经网络的第二个问题是它们对噪声很敏感。 事实上，由于在训练阶段将异常值视为正常点，因此不可避免地会出现异常值

singola@gmx.com

是模型中的错误。 将异常值视为正常点的问题将表现为过度拟合。 在多层网络的情况下，这个问题尤其重要。 对于一个非常复杂的多层网络，即使使用预训练来避免过度训练，训练数据中的每个点都能够获得异常值0。 但是，抽样点数将继续具有更实际的分数，因为它们未包含在培训过程中。 因此，建议仅使用数据的随机子集来训练模型，并使用得到的模型对剩余点进行评分。 该随机子集的大小取决于所使用的多层网络的复杂性。 对于更复杂的网络，需要更大的训练数据集。 因此，应根据可用的训练数据量选择神经网络的复杂性。 一种自然的方法是从训练数据中重复采样点，创建模型，并使用生成的模型对剩余点进行评分。 然后对各个随机样本（其中对它们进行评分）的每个点的得分进行平均，以便提供最终结果。 使用这种方法的一个挑战是神经网络训练缓慢，因此训练多个模型有时在计算上是不可行的。 然而，最近的一些方法，如辍学训练[然后对各个随机样本（其中对它们进行评分）的每个点的得分进行平均，以便提供最终结果。 使用这种方法的一个挑战是神经网络训练缓慢，因此训练多个模型有时在计算上是不可行的。 然而，最近的一些方法，如辍学训练[然后对各个随机样本（其中对它们进行评分）的每个点的得分进行平均，以便提供最终结果。 使用这种方法的一个挑战是神经网络训练缓慢，因此训练多个模型有时在计算上是不可行的。 然而，最近的一些方法，如辍学训练[可以利用510]以高效的方式模拟集合性能，而无需明确地创建多个网络模型。

3.6.4 神经网络的广阔潜力

前面提到的部分显示了两种不同的方法，通过使用具有不同解释的输出来建模神经网络设置中的异常值。 在实践中，几乎有无数种方法可以使用神经网络来模拟异常值。 例如，通过适当地定义输出节点和相应的优化问题，神经网络可用于捕获各种无监督模型，例如自组织映射[593]，混合建模和聚类方法。 由于像聚类和混合物建模方法可以用于异常值检测（参见章节2 和4），人们也可以调整这些神经网络方法，将数据点评分为异常值。 鉴于深度学习的最新进展，这些方向代表了神经网络尚未开发的潜力。 然而，总是需要小心地以这样的方式设计模型，以避免过度配置神经网络的普遍问题。

3.7 线性建模的局限性

回归分析作为异常值检测的工具有一些限制。 本章最开始讨论了这些缺点中最重要的一点，其中探讨了回归分析的数据特性。 特别是，数据需要高度相关，并且沿着低维子空间对齐，以使回归分析技术变得有效。 当数据不相关但在某些区域高度聚集时，这些方法可能无法有效地工作。 在这种情况下，非线性模型和内核方法提供了有用的选择。 然而，这些方法是计算密集型的并且可能经常导致过度配置。

另一个相关问题是数据中的相关性可能不是全局性的。 最近的一些分析观察[7]表明子空间相关性特定于数据的特定位置。 在这种情况下，PCA发现的全局子空间对于异常值分析来说并不是最理想的。 因此，为了创建，有时将线性模型与接近模型（在下一章中讨论）结合起来是有用的

singola@gmx.com

更一般的局部子空间模型。 这将是高维和子空间离群检测的主题，将在第5章中详细讨论。

与任何基于模型的方法一样，当与一小组数据记录一起使用时，过度配置仍然是一个问题。 在这种情况下，记录数量与数据维度的关系很重要。 例如，如果数据点的数量小于维度，则可以找到方差为零的一个或多个方向。 即使对于数据大小与数据维度具有更大（但相似）幅度的情况，也可以观察到方差的相当大的偏差。 从图3.5的结果可以看出这一点（c）和（d），其中一小组均匀分布数据的特征值存在相当大的偏差。 随着数据大小的增加，这种偏斜会减少。 这是过度配置的经典案例，当数据集大小较小时，仔细解释结果非常重要。

基于回归的方法的可解释性相当低。 这些方法将数据投影到更低维度的子空间，子空间表示为原始特征空间的线性（正或负）组合。 在许多实际应用中，这不能简单地解释为物理意义。 这也有减少用户对特定应用的强烈知识的不利影响。 这是不合需要的，因为能够解释为什么数据点在原始数据空间的特征方面是异常值通常是有趣的。

最后，当数据的维度很大时，该方法的计算复杂性可能是一个问题。 当数据具有d的维数时，这导致 dd 协方差矩阵，其可能相当大。 此外，该矩阵的对角化将至少随着维数的增加而减慢地减速。 最近提出了许多技术，它们可以比二次维度更快地执行PCA [230]。 在线性方法的内核概括的情况下，计算问题尤其具有挑战性。 这些问题也可以通过集合方法得到改善[32]。 随着矩阵计算方法的进步和计算机硬件技术的日益增强，近年来这个问题已经不再是一个问题。 这种降维技术现在可以轻松应用于大型文本集合，其维数为几十万字。 事实上，近年来，神经网络和深度学习等方法已变得越来越可行。 如果可以克服与这些方法相关的计算挑战，它们通常可用于提供稳健的结果。

3.8 结论和总结

本章介绍了异常值检测的线性模型。 许多数据集显示了不同属性之间的显着相关性。 在这种情况下，线性建模可以提供一个有效的工具，用于从基础数据中删除异常值。 在大多数情况下，主成分分析为离群值去除提供了最有效的方法，因为它对数据中存在少量异常值更为鲁棒。 这些方法也可以扩展到非线性模型，尽管该方法计算复杂并且有时可能超过数据。 许多其他数学模型，如SVM，矩阵分解和神经网络，使用这些概念的不同变体。 多层神经网络可以模拟复杂和非线性模式，特别是使用深度学习方法。 除了神经网络，大多数这些模型都是全局模型，它们不能识别不同数据位置中不同的子空间和相关模式。 但是，它提供了一个通用框架，可用于广义局部线性模型

singola@gmx.com

在本章节中讨论4 和5。

3.9 书目调查

回归和异常值检测问题之间的关系已在文献中进行了广泛的探索[467]。 异常值分析通常被认为是对噪声效应的稳健回归的巨大挑战，并且这推动了整本关于该主题的书籍。 在许多情况下，异常值的存在可能导致回归分析方法的不稳定行为。 本章的图3.3（b）说明了这种情况的一个例子，其中单个异常值将回归斜率完全改变为不反映数据真实行为的回归斜率。 可以证明，在特定条件下，异常值对回归系数的估计具有任意大的影响。 这也被称为故障点[245，269]回归分析。 在异常值分析中这种情况是非常不可取的，因为可能会产生非常误导的结果。 随后，提出了许多具有更高分解点的估算器[467]。 在这种情况下，需要在数据中存在更高水平的污染以便发生故障。

主成分分析方法也经常用于经典文献[296]中，用于回归分析和降维。 它在文本域中的噪声校正应用首先在[425]中观察到，然后在理论上在[21]中建模。 结果表明，数据点投影到具有最大方差的超平面上提供了数据表示，由于从数据中去除了噪声，因此具有更高的相似性计算质量。 潜在语义索引[162，425]，PCA的一种变体，最初是在文本数据的背景下提出的，目的是减少检索的维数，而不是降低噪声。 然而，LSI多年的经验表明，检索质量实际上有所提高，正如[425]中明确指出的那样。 后来，这在理论上是为关系数据建模的[21]。 PCA和LSI是降维技术，通过发现维度之间的线性相关性来总结数据。

基于PCA的技术已经在顺序被用于检测在多种结构域的异常值，例如统计[116]，天文学[177]，生态数据[282]，网络入侵检测[334，493，544]，和许多种时间序列数据。 一些上述应用是暂时的，而其他应用则不是。 由于PCA与时间序列相关性分析之间的关系，这种回归方法的大部分应用已经到了时间域。 基于回归的方法将在第9章中重新访问，将讨论一些时间异常值分析的方法。 在时间数据的背景下，异常值分析问题与时间序列预测问题密切相关，其中时间序列中预测值的偏差被视为异常值。 在[22]中还讨论了各种基于回归的时间序列传感器数据流中的降噪和异常检测方法。 此外，已经使用了异常detec-和灰中的曲线图[许多类似于PCA的结构和时间版本的方法280，519]。 在这样的方法中，邻接矩阵的增强形式或相似性矩阵可以用于特征向量分析。 这些方法通常称为谱方法，将在第12章中讨论。 非线性降维的方法在[讨论481]，并应用到新颖性检测是在[讨论270，541]。 然而，[270]中的工作使用重建误差（硬核PCA）作为异常分数，而不是本书中讨论的软方法[35]。 核心Mahalanobis方法在[35]中被正式提出并作为一种独特的方法进行了测试。 ap-

singola@gmx.com

在[475]中还讨论了这种非线性降维对于离群检测的应用。 然而，[475]中的方法使用光谱方法而不是全局降维来增强还原过程的局部性质。 如果数据的不同部分显示出不同的流形结构，那么这种方法将更有效。

超越全球PCA的另一个通用模型是将数据建模为PCA的概率混合[549]。 这被称为概率PCA（PPCA）。 在混合建模过程中，这些方法很难在基础数据中产生噪声。 [161]中提出的方法通过以学生t分布的形式对下层噪声建模来增加PCA的稳健性。 基于PCA的聚类算法的异常值的影响是显着的。 [7]中的工作提供了一种方法，用于提供异常值作为聚类算法输出的副产品。 此外，在第5 章中将详细讨论在异常值分析中使用本地PCA的方法关于高维数据的离群分析。 在[591]中提供了一种在高维数据中使用维数降低方法的最新技术。 减少维度的一种补充方法是使用稀疏编码的RODS框架[178]。 稀疏编码方法将数据转换为高维和稀疏表示。 外部定义为包含字典原子的数据点，因此通常不会出现在其他数据点中，因此特定于异常。 在[3]中讨论了一个共同发现稀疏编码和异常值的框架。

内核的支持向量机已被经常用于新颖检测与使用的一类版本的模型[的480，384]。 关于支持向量机的一般讨论可以在[33]中找到。 [384]中的工作特别值得注意，因为它提供了对这些模型在不同数据集中的表现的有趣评估。 一类支持向量机可以对数据域，特征表示和内核函数的选择敏感。 在支持向量机的性能的变化和灵敏度在[讨论384，460]。 为支持向量机CLASSI音响阳离子其它一类的方法在所讨论的[52，113，303，384，460，538, 539].

原则上，任何矩阵分解技术都可用于异常值分析。 在[576]中讨论了使用矩阵分解的异常值分析方法的示例。 核心原则是降维方法提供了数据的近似表示以及相应的残差集。 这些残差可以用作异常值。 本章讨论的矩阵分解方法常用于推荐系统[34]。

神经网络在[84]中有详细讨论;[223]讨论了深度学习的问题。 一类神经网络在孤立点检测中的应用在[讨论268，388，389，529]。 [385]中提供了一类神经网络在文档分类中的具体应用。 另一类常用的神经网络是复制器神经网络[53，147，250，567，552]，用于将数据点评分为异常值。 复制器神经网络的最新实现可以在[160]中找到。 在[53]中探讨了使用深度学习自动编码器进行异常检测。

3.10 演习

1. 考虑以下观察的数据集：（1,1），（2,0.99），（3,2），（4,0.98），（5,0.97）。 使用Y 作为因变量执行回归。 然后使用X 作为因变量执行回归。 为什么回归线如此不同？应该删除哪一点以使回归线更相似？

singola@gmx.com

2. 对练习1的数据集执行主成分分析。 确定表示数据的最佳1维超平面。 哪个数据点离这个1维平面最远？

3. 删除练习2中找到的异常点，并对剩余的四个点执行PCA。 现在将异常点投影到最佳回归平面上。 校正点的价值是多少？

4. 假设您有包含数字数据的调查。 你知道参与者偶尔会在其中一个领域犯错误，因为它很难正确填写。 讨论如何检测这些异常值。

5. 如果参与者可能在任何一个领域而不仅仅是一个特定的领域犯了错误，你对前一个问题的回答会如何变化？

6. 从UCI机器学习库[203]下载KDD CUP 1999数据集，并对定量属性执行PCA。 表示（i）方差的80％，（ii）方差的95％，以及（iii）99％的方差所需的子空间的维度是多少。

7. 使用来自UCI机器学习库[203]的Arrythmia数据集重复练习6。

8. 在100维空间，其中从均匀分布中生成的每个维随机生成1000个数据点（0，1）。 使用此数据集重复练习6。 当您使用1,000,000个数据点而不是1000时，会发生什么？

9. 考虑具有变量X 和Y 的二维数据集。 假设V ar（X）
V ar（Y ）。 与Y- on-X 回归线的斜率相比，这如何影响X-on-Y 回归线的斜率。 这是否为您提供关于为什么在图回归线之一任何见解3.3相比，在图（B）转移显着地3.3（一）因为加了一个局外人的，？

10. 缩放Arrythmia数据集的每个维度，使每个维度的方差为1.使用缩放数据集重复练习7。 缩放过程是否会增加所需尺寸的数量，还是减少尺寸？为什么？您可以从中获得关于任意数据集的一般性推论吗？

11. 设Σ为数据集的协方差矩阵。 让Σ对角化如下：

$$\Sigma = PDP^T$$

这里，D 是包含特征值λi的对角矩阵，D^{-1}也是包含特征值的倒数的对角矩阵（即$^1/\lambda i$）

- Show that $\Sigma{-}1 = PD^{-1}PT$
- 对于给定的数据点X 从数据用平均值设定μ，表明马哈拉诺比斯距离的值（Xμ）Σ-1（Xμ）^{T}之间X 和平均μ降低到相同的表达如方程比分3.17。

singola@gmx.com

第4章

Proximity-Based Outlier Detection

“要带领管弦乐队，你必须转向人群。”- Max Lucado

4.1 Introduction

基于邻近度的技术在其地点（或邻近度）稀疏填充时将数据点定义为异常值。 数据点的接近度可以通过多种方式定义，这些方式彼此略有不同，但足够相似，值得在单个章节内进行统一处理。 定义离群值分析的最常见方法如下：

- **基于群集：**任何群集中的数据点的非成员资格，其与其他群集的距离，最近群集的大小或这些因素的组合用于量化异常值分数。 聚类问题与异常值检测问题具有互补关系，其中点要么属于聚类，要么应被视为异常值。
- **基于**距离：使用数据点到其k最近邻居（或其他变体）的距离以便确定接近度。 具有较大k近距离的数据点被定义为异常值。 基于距离的算法通常以比其他两种方法更详细的粒度执行分析。 另一方面，这种更大的粒度通常会带来显着的计算成本。
- **基于密度：**使用数据点的指定局部区域（网格区域或基于距离的区域）内的其他点的数量来定义局部密度。 这些局部密度值可以转换为离群值。 也可以使用其他基于核的方法或用于密度估计的统计方法。 聚类和基于密度的方法之间的主要差异在于聚类方法对数据点进行划分，而基于密度的方法对数据空间进行划分。

显然，所有这些技术都是密切相关的，因为它们基于一些概念
接近度（或相似度）。 主要差异在于如何接近的详细程度

© Springer International Publishing AG 2017

C.C. Aggarwal, *Outlier Analysis*, DOI 10.1007/978-3-319-47578-3_4

singola@gmx.com

被定义了。 这些不同的定义异常值的方法可能具有不同的优点和缺点。 在许多情况下，当使用[1] 个以上的概念来定义异常值得分时，这些不同类别的方法之间的区别变得模糊。 本章以统一的方式解决了这些问题。

基于距离和其他两类方法之间的一个主要差异在于执行分析的粒度级别。 在基于聚类和密度的方法中，通过划分点或空间，在异常值分析之前预先聚合数据。 将数据点与此预聚合数据中的分布进行比较以进行分析。 另一方面，在基于距离的方法中，计算到原始数据点（或类似变体）的k最近邻距离作为离群值得分。 因此，最近邻方法中的分析是在比聚类方法更详细的粒度级别上进行的。 相应地，这些方法提供了不同的交易
在不同大小的数据集的有效性和效率之间。 除非采用索引或修剪技术来加速计算，否则最近邻方法可能需要O（N 2）时间来计算具有N个记录的数据集的所有k最近邻距离。 但是，索引和修剪技术通常只能在一些受限制的设置中很好地工作，例如低维数据集。 此外，修剪不是为异常值而设计的
分数计算，它只能用于需要报告二进制标签（表示点是异常值）的设置。 尽管有这些缺点，最近邻方法仍然非常受欢迎。 这是因为这些方法通常可以提供更详细和准确的分析，尤其是对于无法进行稳健聚类或密度分析的较小数据集。 因此，模型的特定选择取决于数据的性质及其大小。

基于邻近的方法自然地设计用于检测噪声和异常，尽管不同的方法适合于这些不同类型的异常值。 例如，近端稀疏性的弱定义，例如群集中数据点的非隶属度，自然地设计用于检测弱离群值（或噪声），而基于密度或距离的定义的大量偏差或稀疏度可以还检测强异常值（或异常值）。 由于其直观的简单性和可解释性，这些方法非常受欢迎。 事实上，有许多方法可以直观地探索和解释异常值[318]基于以邻近为中心的定义。 由于底层方法的简单性，它们可以很容易地推广到几乎所有类型的数据，例如时间序列数据，序列数据或图形数据。

本章安排如下。 第4.2 节讨论了在异常值分析中使用聚类的方法。 第4.3 节讨论了基于距离的离群检测方法。 基于密度的方法将在4.4节中讨论。 基于接近度的离群值检测的局限性将在4.5节中讨论。 第4.6 节介绍了结论和总结。

4.2 集群和异常值：互补关系 - tionship

聚类和异常检测之间存在众所周知的互补关系。 一个简单的观点是每个数据点都是集群的成员或异常值。 在聚类中，目标是将点划分为密集子集，而在异常值检测中，目标是识别在这些点中看起来不自然的点。

[1]本章稍后将讨论，众所周知的LOF方法[96]可以解释为基于距离的方法或基于密度的方法，具体取决于它的呈现方式。

singola@gmx.com

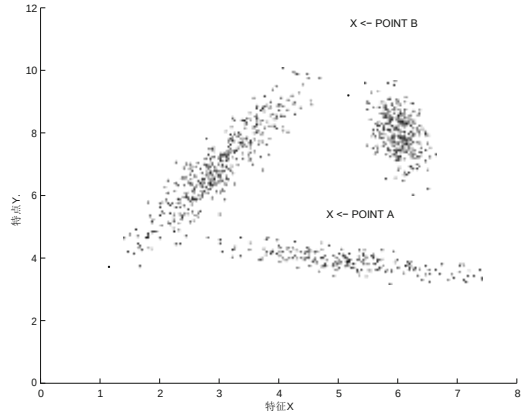


图4.1：重新审视图2.9 的示例：适当的距离计算可以检测到更好的异常值

密集的子集。 事实上，大多数聚类算法都会报告异常值作为其分析的副产品。

然而，重要的是要理解仅使用点到簇的互补关系以便定义异常值导致弱异常值或噪声的发现。 这是因为聚类中数据点的非成员资格是一种相当钝的锤子，用于测量数据点与正常模式的偏差水平。 例如，位于大型集群边缘的数据点与完全与所有其他集群隔离的数据点非常不同。 此外，非常小的聚类中的所有数据点有时也可被视为异常值。 因此，当使用聚类进行离群值检测时，使用更细微的方法（比非聚类成员的方法）来计算离群值得分。

可以通过使用数据点到聚类质心的距离来构造异常值得分的简单定义。 具体而言，数据点到其最近的聚类质心的距离可以用作数据点的离群值得分的代理。 由于聚类可能具有不同的形状和方向，因此使用的优秀距离测量是马哈拉诺比斯距离，其通过沿相关方向的局部聚类方差来缩放距离值。 考虑包含k个簇的数据集。 假定- [R个集群d维空间具有对应的d维行向量率μr 的属性明智装置，以及DD 共方差矩阵Σř。 该矩阵的第（i，j）个条目是本地的 —
该群集中维度i 和j 之间的协方差。 然后，马氏距离平方MB（X，导率μr，Σ[R）2中的数据点之间的X （表示为行向量），并与形心的簇分布率μr 和协方差矩阵Σ[R 是德音响定义如下：

$$MB(\overline{X}, \overline{\mu_r}, \Sigma_r)^2 = (\overline{X} - \overline{\mu_r})\Sigma_r^{-1}(\overline{X} - \overline{\mu_r})^T \tag{4.1}$$

在使用当地Mahalanobis距离对数据点进行评分之后，可以对这些得分应用任何形式的极值分析，以将它们转换为二进制标签。

人们还可以将马哈拉诺比斯距离视为在一些变换和缩放之后点与群集质心之间的经调整的欧几里德距离。 具体而言，点和质心被转换为由主要分量方向（群集点）定义的轴系统。随后，沿着由这些定义的每个新轴计算候选离群点和簇质心之间的平方距离。

singola@gmx.com

主成分，然后除以沿该成分的聚类点的方差。 所有组件上的这些缩放值的总和提供了平方马哈拉诺比斯距离。 Mahalanobis距离的效果是根据特定数据位置的特征提供统计标准化。 即使在群集方差较小的方向上的小距离在该数据局部性内也可能是统计上显着的。 类似地，沿着集群方差大的方向的大距离在该数据位置内可能在统计上不显着。 与欧几里德距离的全球使用相比，这种方法将产生更多重新定义的结果，因为它更适合于手头的数据局部。 从图4.1所示的示例可以看出这一点其中数据点'A'更明显地是数据点'B'的异常值，因为后者可能（弱）与细长簇之一相关。 然而，使用欧几里德距离不能检测到这种细微的区别，根据该距离，数据点'A'最接近最近的聚类质心。 值得注意的是，马哈拉诺比斯距离的使用实现了与本章后面讨论的一些其他基于局部密度的方法（如LOF和LOCI）所实现的局部归一化类似的目标。

离群值评分标准应始终与在聚类算法中优化的目标函数紧密相关。 当Mahalanobis距离用于评分时，它也应该在聚类过程中用于距离计算。 例如，Mahalanobis k-means算法[33]可用于聚类阶段。 因此，在每个分配迭代中，基于到各个群集质心的马哈拉诺比斯距离将数据点分配给群集。 因此，聚类过程将对底层聚类的不同形状和方向敏感（如图4.1所示）。 实际上，第2 章讨论了EM算法可以被认为是Mahalanobis k-means算法的软版本[23]。 注意，第2 章中概率模型的每个混合分量的高斯分布的指数中的项是（平方）Mahalanobis距离。 此外，由EM算法计算的fi t值通常由到最近的聚类质心的指数Mahalanobis距离支配。 Mahalanobis k-means算法将软概率转换为硬分配。 因此，基于聚类的异常值分析方法是第2章中介绍的（软）概率混合模型的硬化身。

除了基于距离的标准之外，通常使用群集基数作为异常值得分的组成部分。 例如，最近聚类中的点分数的负对数可以用作离群值得分的分量。 一个人可以创建两个单独的N.-基于距离和基数标准的分数维度向量，将每个向量标准化为单位方差，然后添加它们。 基数方法在基于直方图的方法中特别受欢迎，其中空间被划分为大小相等的区域。 值得注意的是，基于直方图的方法是聚类方法的变种。 将群集基数纳入分数有助于将小群聚类异常值与较大聚类中发生的正常点区分开来。 通过对每个簇中的数据点数量使用最小阈值，可以更有效地实现聚类异常的识别。 一个例子如图4.2所示其中使用阈值4足以识别三个孤立的数据点。 这种异常值在实际应用中很常见，因为相同（罕见）过程可能会多次生成这些异常值，尽管次数很少。 通常，聚类方法在处理聚类异常时比基于直方图的方法要好得多，因为它们以更灵活的方式划分数据点而不是数据空间; 因此，他们可以在分区过程中更容易地检测和调整这些类型的小簇。

基于聚类的方法自然具有很高的预测可变性，具体取决于

singola@gmx.com

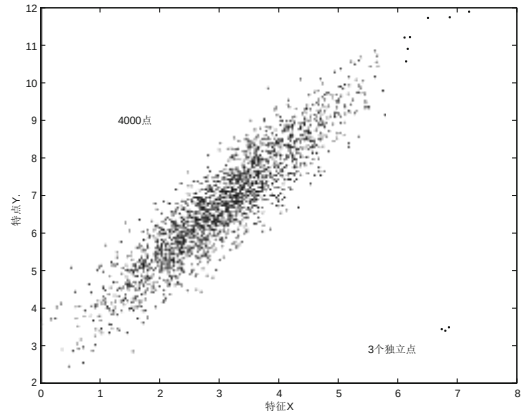


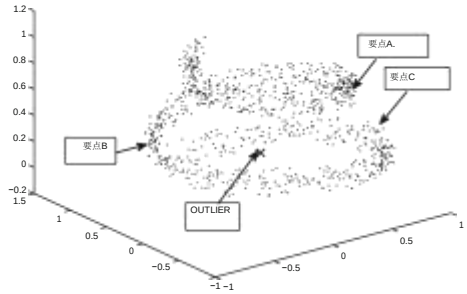
图4.2：重新审视图1.5 的示例：基于邻近度的方法中全局和局部分析的正确组合可以识别此类异常值

特定的模型选择，随机初始化或参数设置。 如第6章所述，这种类型的可变性是期望中次优检测器的理论指示。 为了提高性能，通常建议使用来自多个聚类（具有不同参数设置）的异常值分数的平均值来获得更好的结果。 即使在已知最佳参数设置的情况下，在随机聚类算法的不同运行中平均（标准化的）离群值得分也是有帮助的，以获得良好的结果。 确定性聚类算法可以通过在数据样本上运行，使用不同的初始化点，或明确随机化算法的特定步骤来适当地随机化。 通常，足够的随机化通常比基本聚类方法的质量更重要。 401]。 尽管来自单个聚类应用的得分通常不是最理想的，但是使用这种类型的集合方法提供了令人惊讶的良好结果。 第6章提供了有关此类方法的更多详细信息。 聚类集合的一个有趣特性是发现的异常值类型对所使用的聚类类型敏感。 例如，子空间聚类将产生子空间异常值（参见第5章第5.2.4 节））；相关敏感的聚类将产生相关敏感的异常值，而局部敏感的聚类将产生局部敏感的异常值。 通过组合不同的基本聚类方法，可以发现不同类型的异常值。 甚至可以将这种广泛的方法扩展到其他数据类型。 聚类集合已用于发现图数据中的边缘异常值[17]（参见12.3.2.3节）第12章）。

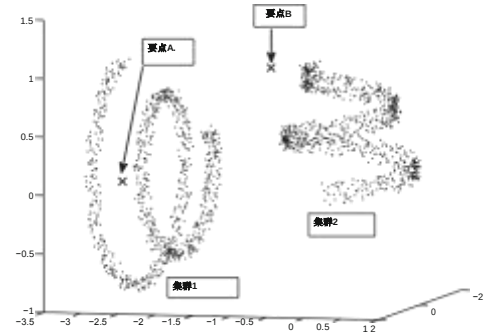
4.2.1 对任意形状群集的扩展

上一节中关于使用（局部）Mahalanobis距离的讨论表明，到最近的聚类质心的距离的计算应该对相应聚类的形状敏感。 虽然马哈拉诺比斯计算对于椭圆形（高斯）聚类的情况是有效的，但对于具有任意和非凸形状的聚类的情况，它并不是那么有效。 此类集群的示例如图4.3所示。 在图4.3（a）的情况下（参见第3章的图3.6（a）），整个数据被排列成单个螺旋状的全局流形。 图4.3（b）的情况

singola@gmx.com



(a) 全球数据分布



（b）本地数据分布

图4.3：不同的簇可能具有不同的形状。 相似矩阵的稀疏性对于创建嵌入在不同地方的不同聚类形状的嵌入是至关重要的。

因为数据的不同地点包含不同形状的集群，所以更具挑战性。 值得注意的是，图4.3（a）和（b）中的一些异常值实际上位于非凸簇的稀疏区域内。 这种异常值的一个例子是图4.3（b）中的簇1的点'A' 。 点'A'比集群1的许多其他点更接近集群1的质心。 显然，我们需要某种类型的数据转换，可以将点映射到暴露这些异常值的新空间。

图4.3（a）的情况比4.3（b）稍微简单一些 ，因为整个数据集在前者中排列为单个全局分布而在后者中不排列。 对于图4.3（a）的情况，它可以使用非线性主成分（或核主成分）分析等方法将点映射到一个新的空间，在该空间中可以有效地使用欧几里德距离。 这种方法将在第3章第3.3.8 节中讨论。 但是，这些方法需要一些修改来调整图4.3（b）中不同数据局部的影响。 在进一步阅读之前，建议读者重新阅读本节中讨论的非线性PCA方法第3章3.3.8 。

与非线性PCA的情况一样，我们可以使用N × N的最大特征向量
核相似度矩阵S = [sij]将数据嵌入到多维空间中
欧几里德距离函数可以有效地用于聚类。 S的 第（i，j）个条目等于第i个和第j个数据点之间的核相似性。 从这个意义上讲，第3 章3.3.8.1 节中描述的内核相似性函数提供了一个很好的起点。 通常优选高斯核，并且将带宽设置为采样的数据点对之间的中值距离。 但是，这些方法最适合于发现图4.3中的数据的全局嵌入（a）整个数据集以单一分布排列。 因此，在构建这些相似性矩阵时需要进行重要的修改，以便处理不同地区的数据变化分布（如图4.3（b）所示）。 这些MODI科幻阳离子，这是从谱聚类[概念衍生378]，是那些相似的矩阵sparsi音响阳离子和本地归一化[297，378]：

- **Sparsiftcation：**在这种情况下，我们保留在所计算的相似性小号的条目（I，J），如果我是间ķ的-nearest邻居ĵ，或者如果ĵ 是间ķ的-nearest邻居我。 否则，将S的这些条目设置为0.该步骤有助于降低相似性

singola@gmx.com

来自不同群集的点之间，尽管是以嘈杂的方式。 因此，它有助于创建嵌入，其中不同的聚类支配嵌入的不同维度。 在执行该步骤之后，矩阵S 变得稀疏，因为sij的大多数值是0。

- **局部归一化：**此步骤对于执行一种局部归一化非常有用，它可以帮助[2] 调整不同局部区域的密度。 但是，此步骤是可选的。 设ρi 是S的第i行中相似度的总和。 注意，ρi 表示第i个数据点附近的一种局部密度，因为它是由第i个数据点定义的它与邻居的相似之处。 然后将每个相似度值sij 除以ρi 和ρj 的几何平均值[297]。 换句话说，我们设置骶髂关节⇐ 骶髂关节/$\sqrt{\rho I \cdot \rho j}$。

随后，顶米矩阵的本征向量小号被提取并且被堆叠在的列n×m个矩阵d̂。 通常，m 的值非常小，例如10到15，
虽然它可能依赖于数据集。 矩阵Dj的每列被缩放到单位
规范。 该矩阵提供N个数据点的 m维表示 。 该
m的值应粗略设置为提取的簇数。 随后，使用该新表示上的k-means算法将Dj中的数据聚类成m个不同的聚类。 所有点都分配给它们最近的簇，即使它们看起来像异常值。
通过在变换的表示中聚类数据，可以发现任意形状的聚类。

每个点到其最近质心的距离被报告为离群值得分。 然而，在计算离群值得分时，使用大于m 的特征向量是很重要的，因为通常沿着小的本征向量强调异常值。 可以仅使用少量特征向量来执行聚类，但是通常（但不总是）沿着较小的特征向量隐藏异常。 因此，提取S的 所有非零特征向量（同时丢弃由明显数值引起的非零特征值）

错误）。 这导致 N × n 表示 Dn，其中 nm。 矩阵行
Dn 被划分为属于各种群集的点，这些点被发现
使用m维表示。 让数据矩阵包含n维
这些簇的表示由D(1) ... D(c) 表示。 每个D(j) 的列是

n n n

scaled3 to unit norm，以计算嵌入式中的局部Mahalanobis距离
空间。 上述聚类质心的n维表示是
通过计算每个聚类D(j) 中的n维点的平均值来构建（即使聚类本身是在m维空间中执行的）。 在这个n维空间中，每个点与其簇心的平方欧几里德距离报告为
outlier score.

使用这种方法的一个问题是相似性矩阵S的一些条目是有噪声的。 例如，来自不同群集的点之间具有高度相似性的条目是有噪声的。 这在原始表示中是可能的，因为很难使用依赖于原始空间中的欧氏距离的内核相似性函数（如高斯核）精确地计算相似性。 即使是少数这样的噪声条目也会严重损害频谱嵌入。 在[475]中讨论了一种方法，用于迭代地聚类点并使用聚类来校正S中的噪声条目。 重复该过程以收敛。 然而，[475]中的算法使用k-最近邻法对点进行评分而不是集群质心距离。

[2]虽然我们没有详细描述这种局部归一化的基本原理，但它与局部异常因子（LOF）算法的4.4 节中描述的类似。

[3]可能需要再次移除某些列，因为它们可能具有零（局部）方差。 因此，某些群集可能具有少于n个维度。

singola@gmx.com

通过使用簇集合方法可以获得稳健性，其中簇的数量，特征向量的数量，稀疏度水平和核带宽在基本方法的不同执行的适度范围内变化。 通过使用不同集合分量的平均分数来获得一个点的最终离群值。 还可以在数据点样本上构建聚类，以改善聚类过程的多样性和效率。

4.2.1.1 对任意数据类型的应用

这种光谱方法的一个重要优点是它们可以应用于任意数据类型，只要可以在对象之间定义相似性函数即可。 可以使用文献中任何可用的内核相似性方法来定义不同的数据类型，如时间序列，字符串和图形[33]。 一旦定义了相似度函数，就可以通过去除低权重的边来创建一个稀疏的相似性矩阵。 该矩阵的特征向量用于创建嵌入并执行聚类。 对于每个数据点，它与最近的聚类质心的最近距离用于定义其异常值。

4.2.2 聚类方法的优缺点

聚类方法的一个重要优点是，与（更常用的）基于距离的方法相比，它们相对较快。 基于距离的方法的运行时间在数据维度上是二次的。 另一方面，许多快速聚类算法存在于各种数据域中。 人们还可以使用本节前面讨论的频谱方法，通过定义适当的相似性函数，发现嵌入在任意形状的簇附近的异常值，或任意数据类型。

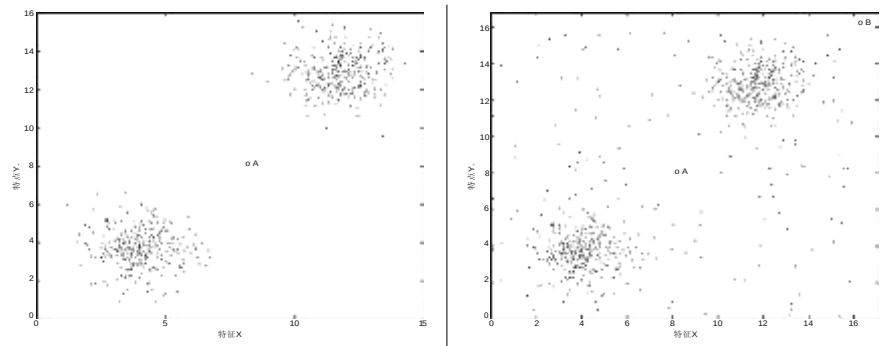
聚类方法的主要缺点是它们可能并不总是在较小数据集中提供所需细节级别的见解。当直接使用相对于原始数据点的距离计算而不是关于诸如聚类质心的聚合代表时，异常值分析方法的粒度通常更好。 因此，当可用数据点的数量足够大时，聚类方法是最有效的;在这种情况下，这些方法也具有效率优势。 聚类方法的另一个问题是得分在不同的随机执行或参数选择之间具有高度的可变性。 因此，平均不同执行的离群值得分（标准化）向量对于获得稳健的结果至关重要。

4.3 Distance-Based Outlier Analysis

基于距离的方法是跨越各种数据域的一类流行的检测算法，并且基于最近邻距离来定义异常值分数。 最简单的例子是将点的k最近邻距离报告为其离群值得分的情况。 由于此定义的简单性，通常很容易将此技术推广到其他数据类型。 虽然本章侧重于多维数值数据，但这些方法已经推广到几乎所有其他领域，如分类数据，文本数据，时间序列数据和序列数据。 本书后面的章节将介绍这些案例的基于距离的方法。

基于距离的方法以自然假设工作，即离群数据点的k最近邻距离远大于正常数据点。 一个主要的

singola@gmx.com



(a) 无噪音 （b）有噪音

图4.4：重新访问图1.1 的示例：由于更好的分析粒度，最近邻算法在嘈杂场景中可能比基于聚类的算法更有效

聚类和基于距离的方法之间的差异在于分析过程的粒度。 与基于聚类的方法相比，基于距离的方法通常具有更高的分析粒度。 基于距离的方法的这种特性可以使得能够更有效地区分噪声数据集中的弱异常值和强异常值。 例如，在图4.4的情况下，基于聚类的算法将不能容易地区分噪声和异常。 这是因为数据点'A'到最近的聚类质心的距离在图4.4（a）和（b）中将保持不变。 另一方面，一个k最近邻算法将区分这些情况，因为在距离评估中将包括噪声数据点。 另一方面，聚类方法将无法很好地区分这些情况，因为聚类质心对底层数据中的噪声相对不敏感。 当然，也可以修改基于簇的方法以包括噪声的影响。 在这些情况下，这两种方法融合到非常相似的方案中。 这是因为这两种方法密切相关。

基于距离的方法还能够识别密切相关的外包的孤立集群。 例如，为了识别包含k_0个数据点的小（异常）簇，需要在k最近邻算法中使用kk_0的值 。 虽然这种异常也可以通过聚类方法来识别，方法是设置阈值 $\geq$ 每个簇中的点数，这些点有时可能进入簇并偏置相应的簇质心。 这可能会以有害的方式影响异常值评分过程。

基于距离的方法的最一般输出是分数形式。 但是，如果需要每个数据点的异常值，那么（vanilla版本的）算法需要的操作与N^2完全成比例。 在识别数据点是否为异常值的二进制决策版本中，它是可能的使用各种类型的修剪和索引结构来大大加快方法。 在下面，我们将讨论两种输出类型的算法。

4.3.1 为基于距离的方法评分输出

基于距离的离群值得分基于其与剩余数据集的第k个最近邻距离。 这个评分机制有两个简单的变化

singola@gmx.com

响应确切的k最近邻和平均k最近邻检测器。 大多数最早的异常值检测方法都集中在使用精确的k-最近邻检测器。

定义4.3.1（精确k-最近邻分数）在数据库D = $X_1 \ldots X_N$中，任何数据点Xi 的异常值得分等于其与D- {Xi}中的点的第k个最近邻距离。

注意，得分点Xi 本身不包括在k个最近邻居中，以避免过度拟合。 例如，如果我们使用k = 1并允许在1个最近邻居中包含候选点，则每个点将是其自己的最近邻居，并且所有点的离群值得分将为0.排除候选点。 邻居之间避免了这种情况。

这个定义的主要问题是，很难知道任何特定数据点的k 的“正确”值。 在诸如离群检测之类的无监督问题中，通常无法使用交叉验证等方法进行参数调整，因为这些方法需要了解地面实况。 对k的变化选择更稳健的替代[58]是平均k-最近邻检测器，其也被称为加权k最近邻检测器。

Deftnition 4.3.2（平均k-最近邻分数） 考虑数据库D = $X_1 \ldots X_N$ 。 任何数据点Xi 的离群值得分等 于其与D- {Xi}中的k个最近邻居的平均距离 。

通常，如果我们知道k的“正确”值，则精确的k最近邻居倾向于给出比平均k-最近邻检测器的k 的最佳值给出的更好的结果。 然而，在诸如离群值检测的无监督问题中，不可能知道任何特定算法的k 的正确值，并且分析员可能使用k的值范围。 例如，分析人员可能会在[1，N /10]中使用等间隔k 值来测试算法。 在这种情况下，平均k-最近邻法对k的不同选择不太敏感，因为它可以在一系列不同的k值范围内平均得出k个最近邻点得分。 很少使用的相关方法是使用谐波平均值而不是算术平均值。

Deftnition 4.3.3（谐波k-最近邻分数）考虑数据库D = $X_1 \ldots X_N$ 。 任何数据点Xi 的离群值得分等于其与D- {Xi}中的k个最近邻居的距离的调和平均值。

必须注意从数据集中删除重复点以便强有力地使用此方法，因为包含0的任何数字集的调和平均值始终是
0.谐波平均值总是由较小的距离（与算术平均值相比）控制，因此使用较大的参数k 值可以提高稳健性。 事实上，在谐波平均的情况下可以设置k = N 并且仍然可以获得高质量的分数。 例如，如果我们在（算术上）平均的k-最近邻检测器中设置k = N ，那么将仅发现多变量极值，并且可以忽略孤立的中心点。 另一方面，在k = N时的谐波平均得分 也将能够发现数据中的孤立中心点，特别是对于大型数据集。 谐波平均的这种行为的原因在于其与基于密度的方法的关联（参见第4.4.4.1节）。 谐波平均值的主要优点是我们现在通过设置k = N 来设置无参数检测器，结果仍然是高质量的。 尽管谐波最近邻检测器在文献中尚未开发，但它们的潜力是显着的。

singola@gmx.com

计算所有数据点的分数通常是计算密集型的。 需要计算数据点之间的所有距离对。 因此，确实需要O（N 2）次操作。 当数据点的数量很大时，这可能非常昂贵。 例如，即使对于包含几十万个点的数据集，该方法也是如此可能无法在合理的时间内返回结果。 尽管经常声称可以使用索引结构在O（N log（N ））时间内有效地找到k个最近邻点，但该方法仅对低维数据集（即，维数小于10）有用。 这是因为在高维情况下修剪指数结构不是非常有效。 此外，这种索引结构具有大的恒定开销，这进一步损害了性能。

更有效的方法是预先选择数据点的样本。 在从样本中排除候选点（如果需要）之后，针对该样本对所有N个数据点进行评分。 结果可以在各种样本上取平均值，以便用整体来提高结果的质量[35]。 对于诸如谐波平均之类的较不稳定的变体尤其如此。

4.3.2 基于距离的方法的二进制输出

尽管基于分数的输出比二进制输出更通用，但它们在实际应用之外的应用有限，超出了发现哪些点是异常值的二元问题。 使用二进制输出的优点是可以修剪许多O（N 2）计算。 因此，只有得分最高的点被报告为异常值，我们不在乎关于非离群点的得分。 这可以通过在最近邻距离[317]（得分）上指定最小阈值或通过使用k最近邻距离的等级上的最大阈值来实现[456]。 前者参数化呈现在选择给分析员一个挑战[4] 的绝对距离阈值前面的（可能是不直观的）值。 基于距离的异常值的原始阈值定义[317]
基于使用分数f 和距离阈值β对其进行参数化：

Deftnition 4.3.4（分数阈值，基于距离离群值） 的对象Ô 在数据集是DB（女，β）离群值，如果至少部分˚F 对象谎言大于距离β 从Ò。 *D*

注意，基于分数的算法具有对应于第k个近邻的单个参数k ，而二进制阈值算法具有两个参数f 和β。 参数f 实际上等同于在原始定义中使用类似k 的参数。 我们可以通过设置k = N （1 f ）来使用精确的第k个最近邻距离，而不是使用分数f 。 为了整个章节的讨论的一致性，我们根据第k个最近邻距离重述这个定义：

| – |

Deftnition 4.3.5（基于分数阈值的距离异常值） 如果数据集D中的对象的精确第k个最近邻距离至少是β，则该对象是异常值。

第二解音响nition [456]是基于顶- [R 阈值，而不是分数的绝对值的阈值处理。 因此，这些点按照k最近邻距离的降序排列。 该顶[R 这样的数据点被报告为异常。 因此，阈值在距离等级而不是距离值上。

[4]可以计算数据点样本的离群值得分，并根据这些得分的均值和标准差设置估计值。

singola@gmx.com

Deftnition 4.3.6（等级阈值，基于距离离群值） 在数据集中的对象是异常值，如果其确切k个最近邻距离为所述顶之间ř 在数据组这样的值。

除了提供给用户的参数选择外，这两个定义几乎完全相同。 实际上，对于距离阈值β的每个选择，可以选择适当的r 值以便在两种情况下产生相同的结果。

请注意，解决此问题的最直接方法是首先使用嵌套循环方法计算所有成对k最近邻距离。随后，可以应用适当的阈值标准来将相关点报告为异常值。 然而，这种天真的方法在计算上是无效的。 毕竟，计算二进制输出而不是评分输出的主要优点是我们可以将异常值检测过程与修剪方法相结合，以使该方法更加高效。

在上述所有定义中，由于文献中这一定义的优势，我们使用了确切的k-最近邻距离。然而，所有上述定义和一些相关的修剪方法可以推广到平均k最近邻距离。 在下文中，我们将讨论针对精确的k-最近邻距离的各种修剪方法，并且还研究它们对平均k-最近邻距离的推广。

4.3.2.1 基于细胞的修剪

基于细胞的技术[317]基于定义4.3.5的分数阈值。 该方法是针对精确的k-最近邻距离而设计的，并且不能容易地推广到平均k-最近邻距离。 在基于单元的技术中，数据空间被划分为单元，其宽度是距离阈值β 和数据维度d的函数。 具体而言，每个维度被划分为最多宽度的单元格
$\beta/(2\cdot\sqrt{d})$。 选择宽度的奇数值以强制细胞中点的某些以距离为中心的特性，这些特性被用于修剪和高效处理。 该
方法最好在二维情况下解释。 考虑√二维情况，in
哪个连续的网格点最多为β/（2· 2）。 重要一点
请记住，网格单元的数量是基于数据空间的划分，
并且与数据点的数量无关。 这是低维数据方法效率的一个重要因素，其中网格单元的数量可能是适度的。 另一方面，这种方法不适用于更高维度的数据。

对于给定的细胞，其L_1邻居是通过跨越单个细胞 – 细胞边界而到达的细胞。 注意，在角落处接触的两个单元也是L_1个邻居。 在大号2个邻居是通过杂交2或3的边界获得的细胞。 标记为X的特定单元 以及其L_1和L_2邻居的集合如图4.5所示。 很明显，内部单元具有8个L_1邻居和40个L_2邻居。 然后，可以立即观察到以下属性。

1. 单元中的一对点之间的距离最多为β/2。
2. L_1邻居中的点与点之间的距离最多为β。
3. Lr 邻居中的点与点之间的距离（其中r> 2）至少为
   β.

唯一无法得出立即结论的单元格是L_2中的单元格。 这代表了特定单元格中数据点的不确定区域。 该方法中的所有距离计算都在该区域中的点对之间执行

singola@gmx.com

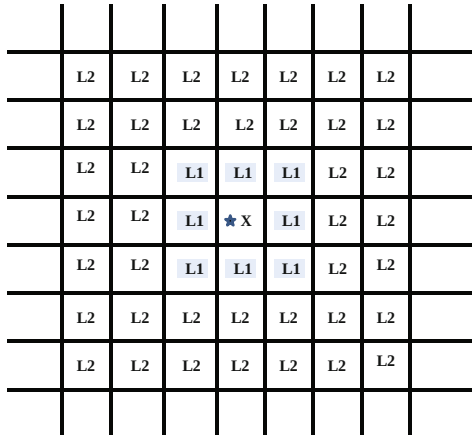


图4.5：基于单元的数据空间分区

不确定性。 然而，通过许多规则可以进一步修剪增益，这些规则能够有效地将一些点识别为异常值或非异常值，而不必实现所有这些距离计算。 这些如下：

1. 如果一个单元中包含多于k个数据点以及它的L_1个邻居，那么这些数据点都不是异常值。
2. 如果在单元格“A”中包含不超过k个数据点并且其L_1和L_2相邻，则单元格“A”中的所有点都是异常值。

该方法使用这些不同的属性和规则以高效的方式将点标记为异常值或非异常值。 第一步是直接将包含多于k 个点的单元格中的所有点标记为非异常值，因为第一个规则。 此外，这些小区的所有相邻小区仅包含非异常值。 为了获得第一规则的完全修剪能力，计算每个单元中的点之和及其L_1邻居。 如果
总数大于k，那么所有这些点也被标记为非异常值。

接下来，利用第二规则的修剪能力。 对于包含至少一个数据点的每个单元“A”，计算其中的点数与其L_1和L_2邻居中的数量之和 。 如果此数字不超过k，则单元格“A”中的所有点都标记为异常值。 此时，许多细胞可能被标记为异常值或非异常值。
这提供了主要的修剪收益。

未标记为异常值或非异常值的单元格中的数据点需要明确计算其k最近邻距离。 即使对于这样的数据点，也可以使用单元结构更快地计算k个最近邻距离。 考虑到目前为止尚未标记为纯异常值或纯非异常值单元格的单元格“A”。 这些细胞可能含有异常值和非异常值的混合物。 单元格“A”中数据点的主要不确定区域是单元格中的一组点
这个细胞'A'的L_2邻居。 不能知道'A' 的 L_2邻居中的点是否在单元'A'中的点的阈值距离β 内 。 明确的距离
需要计算以便确定单元“A”中的数据点的阈值β 内的点数。 L_1和L_2中不超过k 个点的那些数据点的距离小于β 被声明为异常值。 注意，距离计算需要仅从单元“A”中的点到L_2邻居中的点明确地执行

singola@gmx.com

细胞'A' 这是因为，在所有的点大号1个邻居已经已知在小于距离β 从“A”的任何点，且在所有点LR 为R>
2已经知道是
至少与'A'中任何一点的β 距离。 因此，在距离计算中实现了额外的节省水平。

上述描述是针对二维情况的。 方法也可以
扩展到更高的维度。 对于所述mai√n二FF erence d维的情况是在以下方面
单元格的宽度（现在是 β/（2· d））和L2邻居的定义。 在这种情况下
对于二维数据，小区的L2邻居定义为非邻居小区
距离最多三个细胞。 In√较高尺寸的一般情况，L2定义为
一组最多为| 2· d|的单元格 细胞离开但不是直接邻居。 所有其他步骤
该算法保持相同。 但是，对于高维度的情况，这种方法
变得越来越昂贵，因为细胞数量随着数据维数呈指数增长。 因此，该方法通常适用于低维数据。

在许多情况下，数据集可能在主内存中不可用，但可能存储在磁盘上。 因此，数据访问效率成为一个问题。 在[317]中已经显示了如何使用群集页面读取将此方法应用于磁盘驻留数据集。 该算法最多需要三遍数据。 更多细节见[317]。

4.3.2.2 基于采样的修剪

采样方法极其灵活，因为它们可以处理基于分数的阈值（定义4.3.5）或基于等级的阈值处理（定义4.3.6）。 此外，它们可以与精确的k-最近邻检测器或平均k-最近邻检测器一起使用。 它们非常适合修剪，也可以用作整体集合方法[32]（参见第6章）。 由于这些原因，采样方法应被视为基于距离的远离异常值检测的第一道攻击线。 在下面的讨论中，我们将使用基于秩的阈值定义和精确的k- 最近邻检测器。 然而，该方法对于任何其他阈值处理和检测组合的概括是直截了当的，因此被省略。

第一步是从数据中选择大小为S N S的样本，并计算样本中数据点与数据库中数据点之间的所有成对距离。 总共有N个这样的对。 该过程需要O（N s）O（N D）距离计算。 因此，对于每个采样点，k-最近邻距离已经是精确已知的。 顶部-.[R个样本中排名离群值被确定，其中[R 是要返回异常值的数目。 所述的S得分- [R个秩离群值提供了一个下界5 大号上ř在整个数据集中排名的异常值得分。 对于数据点S ，仅知道k个最近邻距离上的上界V k（X）。 这个上限是相等的
到样本 中每个点的k最近邻距离 D D − S S ⊂ D。 然而，
如果该上限Vk（X）不大于下限更大的大号已经确定的，则这样的数据点X 可以从进一步考虑中排除作为可能r 离群值。 通常，只要基础数据集很好地聚集，这将导致立即从大量异常候选中删除。 这是因为只要样本中包含每个聚类中的至少一个点，并且至少有r 个点，就会删除聚类中的大多数数据点。 位于有些稀疏的地区。 这通常可以通过实际数据集中的样本大小s的适度值来实现。
从D-步骤取出这些数据点之后，点的剩余的组为R⊆d - S的k-nearest邻居的方法可以应用到一个更小的候选集合R.

[5]请注意，较高的k-最近邻距离表示较大的离群值。

singola@gmx.com

该项[R 在排名离群 $R \cup S$ 返回作为网络最终输出。 根据已经实现的修剪水平，这可以显着减少计算时间，特别是当|R∪S| | d |。

早期终止技巧与嵌套循环

通过加快计算每个数据点的k个最近邻距离的第二阶段，可以进一步改进上一节中讨论的方法。

R。 这个想法是在的计算ķ任何数据点的-nearest邻距离X 不必通过终止之后一旦已确定X ∈ R 不可能是顶级之中[R 异常值。 在这种情况下，数据库中为计算的的扫描ķ的-nearest邻居X 可以提前终止。 D

注意，基于到样本的距离，已经具有每个X 的k个最近邻距离的估计（上限）V k（X）。 此外，ķ所述的-nearest neigh- BOR距离r日在最佳离群提供的下限，以使它的顶所需的“切割-邻FF”- [R 离群值。 该 S 下限由L表示。 这一估计Vķ（X的的）ķ的-nearest相邻距离X 被进一步拧紧（还原的）作为数据库扫描，并计算 $D - S$ X 的距离。 由于这种运行估计Vķ（X）总是在真正的上限ķ的-nearest相邻距离X，确定所述过程ķ的-nearest邻居X 可以尽快终止Vķ（X）低于下降已知的下界大号在顶- [R 离群值距离。 这被称为提前终止并提供显着的计算节省。 然后，可以处理下一个数据点。 在未实现提前终止的情况下，数据点X. 将 R 乎[6] 总是顶间- [R （电流）离群值。 因此，在这种情况下，下限L 也可以收紧（增加）到新的第r个最佳离群值得分。 当处理下一个数据点以确定时，这将导致更好的修剪它的k最近邻距离值。 为了最大化修剪的好处，不应以任意顺序处理数据点。 相反，它们应该以k最近邻距离（基于）的初始采样估计V k（）的递减顺序进行处理。 这确保了早期发现异常值，并且尽可能快地收紧界限L 以进行更好的修剪。 此外，在内环中，基于V k（Y ）的增加值，可以在相反方向上对数据点Y进行排序。 这样做可以确保k最近邻距离尽可能快地更新，并且提前终止的优势最大化。 嵌套循环方法也可以在没有采样的第一阶段[7] 的情况下实现，但是这种方法不具有正确排序处理的数据点的优点。 具有初始下限开始大号对- [R个从采样阶段中获得最好的离群值评分，嵌套循环执行如下：

[6]我们说“差不多”，因为X的最后一次距离计算可能会使V（X）低于L. 场景很不寻常，但偶尔会发生。

[7]文献中的大多数描述都省略了采样的第一阶段，这对于效率最大化非常重要。 时间序列分析[311]中的许多实现确实更仔细地对数据点进行排序，但不对采样进行排序。

singola@gmx.com

对于每个$\overline{X} \in R$ **做开始**
  对于每个$\overline{Y} \in D - S$ **就开始**
    更新当前k最近邻距离估计$V_k(\overline{X})$
      计算$\overline{Y}$到$\overline{X}$的距离;
    如果$V_k(\overline{X}) \leq L$ **则**终止内环;
  endfor
  if $V_k(\overline{X}) > L$ **then**
    将$\overline{X}$包含在当前r最佳离群值中，并将L更新为新的
      第r个最佳异常值得分;
endfor

注意，数据点$\overline{X}$的k个最近邻居不包括数据点本身。因此，必须注意嵌套循环结构，以忽略$\overline{X} = \overline{Y}$的平凡情况，同时更新k-最近邻距离。

4.3.2.3 基于索引的修剪

索引和聚类是数据本地化和访问的另外两种常见形式。因此，探索是否可以使用一些传统的聚类方法或索引结构来提高基于距离的计算的复杂性是很自然的。最初的方法[456]使用基于排序的阈值处理（定义4.3.6）和精确的k-最近邻检测器。

如在基于采样的修剪的情况下，候选数据点$X = (x_1 \ldots x_d)$的k最近邻距离的上界$V_k(X)$与修剪一起逐渐收紧。该方法使用点集的最小边界矩形来估计候选X到集合中任何点的距离边界。这些边界可用于修剪这些集合，而无需以所讨论的方式进行显式距离计算后来。令R为最小边界矩形，其中沿第i维度的下边界和上边界由$[r_i, r_i^j]$表示。然后，最小距离小的XI沿着我个尺寸在最小外接矩形的任何点- [R是潜在的0，如果$XI \in [RI, R_{我}^j]$。否则，最小距离为$\min_i \{|x_i - r_i|, |x_i - r_i^j|\}$。因此，通过沿着每个维度计算该最小值，可以估计总数$\sum_{i=1}^{d} \min\{|x_i - r_i|, |x_i - r_i^j|\}$。对应的总的最大值可估计为$\sum_{i=1}^{d} \max\{|x_i - r_i|, |x_i - r_i^j|\}$。分别沿着似地，最大距离最小维度到边界矩形R的X的$\max_i$由$\max\{|x_i - r_i|, |x_i - r_i^j|\}$给出最大值[2]。

上述边界可以与索引结构（例如R^*-Tree [78]）结合使用，用于估计数据点的k个最近邻距离。这是因为这样的索引结构使用最小边界矩形来表示数据节点。为了确定数据集中的异常值，逐个处理这些点以确定它们的k个最近邻距离。在算法的过程中动态地保持最高r这样的距离。在索引结构上使用分支定界修剪技术，以便有效地确定$V_k(X)$的值。当对边界矩形的最小距离估计值大于$V_k(\overline{X})$，那么边界矩形显然不包含任何可用于更新$V_k(X)$值的点。R^*-Tree的这些子树可以从考虑中完全修剪。

除了基于索引的修剪之外，还可以尽早考虑单个数据点。到目前为止发现的第r个最佳离群值的得分（即，k-最近邻距离）由Dmin表示。请注意，Dmin与使用的绑定L完全相同

singola@gmx.com

上一节抽样。 数据点X 的V k（X）的估计是单调的
随着算法进展而减少，因为找到了更好的最近邻居。 当该估计值低于Dmin时，可以不考虑点X.

已经针对不同的数据域提出了这种基本修剪技术的许多变体[311]。 通常，这种算法使用嵌套循环结构，其中候选数据点的异常值分数在启发式有序外循环中逐个计算，其近似于异常值分数的降低水平。 对于每个点，在内环中以近似于到候选点的距离增加的启发式排序来计算最近邻居。 当内环最近的近似距离小于目前发现的第r个最佳异常值时（V k（X）<Dmin），可以放弃内环。

外部和内部循环中的良好启发式排序可以确保可以提前考虑丢弃数据点。 不同的技巧可用于在各种应用程序设置中确定此启发式排序。 在许多情况下，在外循环中发现启发式排序的方法使用聚类和异常值检测问题的互补性，并根据包含它们的聚类的基数对数据点进行排序。 包含聚类的数据点首先检查很少（或一个）点。 通常，使用非常简单且高效的聚类过程来创建外循环排序。 用于在内循环中发现启发式排序的方法通常需要快速近似k最近邻排序，取决于具体的数据域或应用程序。 第9章讨论了这种方法对时间序列数据的改编[311] 。

基于分区的加速

如果上面讨论的约束估计过程不充分稳健，则上面讨论的方法可能需要对大量点的V k（X）进行合理精确的计算。 尽管修剪，这仍然是昂贵的。 在实践中，r 的值非常小，并且可以排除许多数据点X 而无需明确地估计V k（X）。 这是通过使用聚类来实现的[456]为了执行数据空间的分区，然后在这个粒度级别分析数据。 基于分区的方法用于以计算上有效的方式修剪那些不可能是异常值的数据点。 这是因为分区表示数据的不太精细的表示，其可以以较低的计算成本处理。 对于每个分区，计算所有包括的数据点的k个最近邻距离的下界和上界。 如果分区中任何点的k-最近邻距离估计的上述上限小于Dmin的当前值，然后可以考虑包含任何异常点来修剪整个分区。 基于分区的方法还为近似Dmin提供了更有效的方法。 首先，通过减小下限对分区进行排序。 的第一个升含有至少分区r̆ 点被确定。 第l个分区的下限提供Dmin的近似值。 使用包含点的索引结构中的节点的最小边界矩形来计算每个分区的上限和下限。 通过使用距每个（未通过的）候选数据点X 的距离这一事实可以获得更多的节省 不需要计算分区中的数据点，这些分区保证比点X （或其包含分区）的k最近邻距离的当前上限更远。

因此，该分析以较不详细的粒度级别执行。 这使得它
效率更接近基于聚类的方法。 实际上，分区本身就是

singola@gmx.com

使用诸如BIRCH的聚类算法生成[611]。 因此，这种方法修剪了许多数据点，然后使用一组更小的候选分区来进行分析。 这极大地提高了该方法的效率。 计算分区边界的确切细节使用上述关于不同分区的边界矩形的最小和最大距离的估计，并在[456] 中详细讨论。 由于基于距离和聚类方法之间的密切关系，使用聚类方法来改善k的近似是很自然的。 - 最近邻距离。 为了许多在文献中使用聚类其它技术以实现更好的修剪和加速比在基于距离的算法[58，219，533]。

4.3.3 数据相关的相似度量

所述ķ-nearest邻居方法可以与其他类型的相似性和距离函数配对。 与欧几里德距离不同，数据相关的相似性度量不仅取决于手头的点对，还取决于其他点的统计分布[33]。 这有助于将各种数据相关特征结合到相似性函数中，例如局部敏感性，相关敏感性或两者。 局部敏感性的一个直观例子是，在亚洲，同一对白种人被认为比在欧洲更相似[548]。

众所周知的局部敏感相似性度量是共享最近邻近相似性度量[287]，其中k个最近邻居集合的两个点的交叉点基数被用作相似度值。 这里的基本思想是，在密集区域中，两个点必须非常接近[8] 以便具有大量共同的最近邻居，而在稀疏区域中，相当远的点可能具有更大的数量。

共同邻居 共享最近邻测量的主要缺点是我们现在有两个参数用于最近邻居的数量，这可能是不同的; 一个参数用于相似度计算，另一个参数用于基于距离的异常值检测算法。 由于在最佳地设置参数方面存在困难，大量参数在无监督问题中产生不确定性。 共享最近邻措施的计算也很昂贵。 通过重复采样数据，计算测量值并对不同样本的测量值求平均值，可以以稳健且高效的方式计算这种相似性。 当使用采样数据计算相似度时，在所有点对之间计算共享的最近邻距离（无论它们是否被采样），

成对的马哈拉诺比斯距离，其适应马哈拉诺比斯方法来计算点对之间的距离，是相关敏感的距离测量。 此度量相当于在使用PCA将数据转换为不相关的方向并将转换后的数据中的每个维度归一化为单位方差后计算点对之间的欧几里德距离。 此度量的一个优点是它使用基础数据的相关结构。 如果在未变换的数据中沿着低方差方向而不是高方差方向对齐，则（未变换的）数据中的单位欧几里德距离处的两个数据点将具有更大的马哈拉诺比斯距离。

[8] 如4.4.1节所述，这种直觉直接用于局部敏感算法，如局部异常因子（LOF）方法。 因此，通过适当地改变精确k-最近邻居算法中的相似度函数，可以实现与诸如LOF的局部敏感算法类似的目标。

singola@gmx.com

基础数据分布[475]。

随机森林可以用于计算依赖于数据的相似性[99，491，555]的自己的能力[是-原因359]脱音响NE数据局部性以依赖于分布方式。 在无监督设置中，基本思想是为异常类生成合成数据，并将提供的数据集视为普通类。 如[491]，异常值在每个属性的最小和最大范围之间随机均匀生成。 在合成标记的数据上构建随机森林。 一对实例（例如，A和B）之间的相似性可以定义为（i）它们出现在同一叶节点中的随机森林中的树的数量，或（ii）公共的平均长度每个树中的两个实例A和B遍历的路径。 这种方法的主要优点是，一旦构建了随机森林，就可以非常有效地计算一对实例之间的相似性。

原则上，只要聚类适应于变化的局部密度和相关性，就可以使用保留其聚类结构的数据的任何分层表示来测量与上述方法的相似性。 创建随机树木其他无监督的方法包括使用随机分层聚类方法[401，555]和隔离森林[548]。 前者[401]还提供了创建包含分配了数据点的叶子和/或内部树节点的标识符的数据的“词袋”表示的选项。 在该表示上的汉明距离的计算几乎等同于随机森林相似性度量。 第8章第8.4.2 节讨论了分类数据的数据相关相似性度量。

4.3.4 ODIN：反向最近邻方法

大多数基于距离的方法直接使用k-最近邻分布来定义异常值。 一种不同的方法是使用反向k-最近邻的数量来定义异常值[248]。 因此，反向k最近邻的概念定义如下：

Deftnition 4.3.7 数据点p 是q的反向k最近邻，当且仅当q 是p的k个最近邻居之一时。

具有大的k个最近邻距离的数据点也将具有很少的反向邻域，因为它们将位于极少数据点的k个最近邻居之间。 因此，异常值被定义为反向k最近邻居的数量小于预定用户阈值的点。

根据潜在的k-最近邻图，也可以容易地理解反向最近邻居方法。 考虑一个图表，其中节点对应于数据点。 当且仅当q 在p的k- 最近邻居中时，有向边（p，q）被添加到图中。 因此，每个节点在该图中具有k 的outdegree 。 但是，节点的入度可以变化，并且等于反向k最近邻居的数量。 具有很少反向k-最近邻居的节点被声明为异常值。 或者，反向k的数量- 最近邻居可能被报告为离群值。 较小的值更能说明更大程度的离群值。 反向最近邻方法也称为使用度数（ODIN）的异常检测。 像前一节中的共享最近邻相似性，所有使用方法ķ-nearest相邻图表[100，248]是局部性敏感。

该方法需要确定每个节点的所有k个最近邻居。 此外，基于距离的修剪不再可能，因为最近的邻居

singola@gmx.com

表4.1：NHL球员统计中的异常值示例[318]

播放机名称	Short-手持目标	Power-玩目标	Game-赢得目标	Game-联系目标	游戏玩过
Mario Lemieux	31	8	8	0	70
Jaromir Jagr	20	1	12	1	82
约翰莱克莱尔	19	0	10	2	82
R. Brind'Amor	4	4	5	4	82

需要明确确定每个节点。 因此，该方法可能需要 $O(N^2)$ 时间来构建k-最近邻图。 该方法的另一个问题是许多数据点可能是反向最近邻居（离群值得分）数量的关系。 这两个问题都可以通过集合方法来解决[32]。 使用s 点的子样本重复评分数据点，其中s 被选择为a 恒定范围内的随机值。 将各种样品的平均分数报告为异常值。

4.3.5 基于距离的异常值的内涵知识

异常值分析中的一个重要问题是保留高水平的可解释性，以提供直观的解释和见解。 这在许多应用程序驱动的场景中非常重要。 [318]首先提出了基于距离的异常值的内涵知识的概念。 这个想法是根据属性的子集来解释对象的异常行为。 因此，在这种情况下，将显示属性子集上的最小边界框，以便解释数据点的异常行为。 例如，考虑国家冰球联盟（NHL）球员统计数据的情况，该数据首先在[318]中提出。 表4.1说明了一组示例统计数据。 样本输出来自[318]，解释这些异常值如下：

Mario Lemieux	一维空间中的异常值 权力发挥目标 短距离目标的二维空间中的异常值 比赛获胜的目标
R. Brind'Amor	一维空间中的异常值 游戏搭售目标。

在[318]中定义了几个概念，以便理解异常值的重要性：

1. 特定的属性集是存在异常值的最小属性集吗？

2. 数据中是否有其他异常值占主导地位的异常值？

内涵知识可以直接用细胞来表征，因为它们可以根据不同的属性定义边界矩形。 [318]中的工作提出了许多卷起和向下钻取方法，以便为内涵知识定义有趣的属性组合。 强弱异常值的概念也被定义。 通常考虑由最小属性组合定义的异常值

singola@gmx.com

从内心角度看更强。 应该强调的是，强弱异常值的这种定义特定于基于知识的内涵方法，并且与本书使用这些术语的更一般形式（作为对象的异常趋势）不同。

4.3.6 基于距离的方法探讨

基于距离的方法与基于聚类的技术相比具有许多定性优势，因为分析的粒度更加详细。 例如，基于距离的算法可以比基于群集的技术更好地区分噪声和异常。 此外，基于距离的方法也可以像聚类方法一样找到孤立的异常值组。 另一方面，聚类方法的优点在于它们可以提供关于用于定义距离的数据点的局部分布的见解。 例如，在图4.1的情况下，可以使用局部聚类结构来定义局部敏感的马哈拉诺比斯距离，这在识别异常值方面比对欧几里德度量的盲目应用更有效。 然而，也可以基于马哈拉诺比斯距离设计基于距离的算法，其也被称为实例特定的马哈拉诺比斯方法[33]。 这些对应于4.3.3节中讨论的与数据相关的相似性度量。

虽然本章后面解释的基于密度的方法确实包含了一些局部性概念，但它们仍然无法提供基于聚类和基于距离的方法的有效组合可以提供的详细的本地见解。 在这方面，一些最近的研究已经将本地聚类分析上市公司的基于距离的方法[7，153，475]。 此外，聚类方法的效率优势应纳入广义的基于距离的方法，以获得最佳结果。

4.4 基于密度的异常值

基于密度的方法使用空间的特定区域中的点数来确定异常值。 它们与聚类和基于距离的方法密切相关。 实际上，根据它们的呈现方式，这些算法中的一些可以更容易地被认为是聚类或基于距离的方法。 这是因为距离，聚类和密度的概念密切相关且相互依赖。 其中许多算法都会发现局部敏感的异常值。 本节讨论本地算法和全局算法。

在[96]中首次注意到基于距离的异常值对数据局部性的敏感性。 虽然图4.1 示出了在两个数据密度和簇取向普通E FF ECT数据局部性的，在工作中[96，97]具体来说地址变化的局部密度的问题。 为了理解这个特定的问题，请考虑图4.6中具有不同密度的数据集的特定示例。 该图包含两个标记为“A”和“B”的异常值。 此外，该图包含两个簇，其中一个比另一个更稀疏。 很明显，除非算法使用较小的距离阈值，否则基于距离的算法不能发现异常值“A”。 但是，如果使用较小的距离阈值，则较稀疏的群集中的许多数据点可能被错误地声明为异常值。 这也意味着当数据的局部分布存在显着的异质性时，基于距离的算法返回的排名是不正确的。 4.2节中也提到了这一观察结果（更一般地说），其中显示了图4.1 中的异常值对本地群集密度和群集方向（本地属性关联）都很敏感。 然而，基于密度的聚类的大部分工作一般都集中在一起

singola@gmx.com

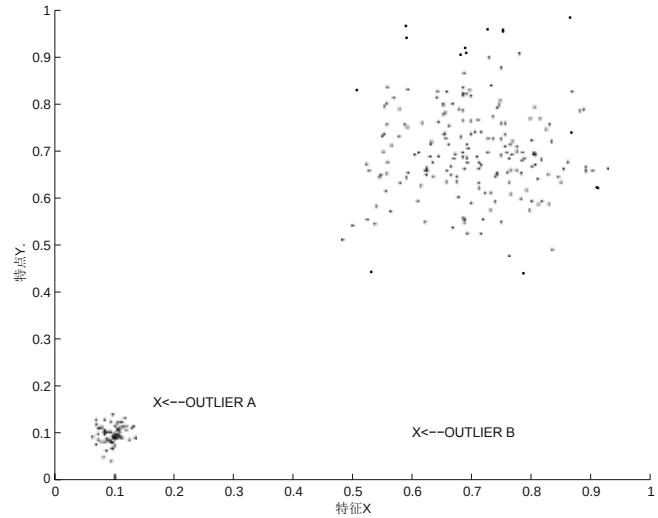


图4.6：局部密度对异常值的影响

关于不同数据密度而不是簇的不同形状和方向的问题。 所述k-nearest邻居算法可以与节的数据相关的相似性度量被配对4.3.3 解决引起的局部密度变化的问题。 然而，LOF方法[96]使用一种不同的方法，通过在本地定义异常值得分调整方式。

4.4.1 LOF：局部异常因子

局部异常因子（LOF）是数据点异常值的量化，能够根据不同局部密度的变化进行调整。对于给定的数据点X，让的Dk（X）是其与距离 k的-nearest邻居 X，并让 路（X）是内的点的集合k的-nearest相邻距离X。 需要注意的是路（X）通常包含k 点，但有时可能包含多于k 因为在联系点k- 最近邻距离。

然后，将可到达性距离RK（X，Y 的对象）X 相对于y 是定义为最大的去连接DIST（X，Y ）和k的-nearest相邻距离y ：

$$R_k(X, Y) = \max\{dist(X, Y), D^k(Y)\}$$

可达性距离在X 和Y 之间不对称。 直观地，当Y 在密集区域中并且X 和Y 之间的距离大时，X 相对于它的可达性距离等于真实距离dist（X，Y ）。 在另一方面，当之间的距离X 和y 是小的，那么可达距离由平滑k的-nearest相邻距离y 。 k 的值越大是的，平滑度越大。 相应地，相对于不同点的可达性距离也将变得更加相似。

然后，数据点X 的平均可达距离ARk（X）被定义为其到达邻域Lk（X）中的所有对象的可达性距离的平均值：

$$AR_k(X) = \text{MEAN}_{Y \in L_k(X)} R_k(X, Y)$$

singola@gmx.com

这里，MEAN函数简单地表示一组值的平均值。 [96]中的工作也将可达性密度定义为该值的倒数，尽管这个特定的假设省略了这一步骤，因为LOF值可以用平均可达距离ARk（X）更简单直观地表达。 。 本地异常因子是然后SIM-帘布层等于的平均比率ARK（X），以在所有点的相应值k的-neighborhood X：

$$LOF_k(\overline{X}) = \text{MEAN}_{\overline{Y} \in L_k(\overline{X})} \frac{AR_k(\overline{X})}{AR_k(\overline{Y})} \tag{4.2}$$

$$= AR_k(\overline{X}) \cdot \text{MEAN}_{\overline{Y} \in L_k(\overline{X})} \frac{1}{AR_k(\overline{Y})} \tag{4.3}$$

在定义中使用距离比率可确保在此定义中很好地考虑局部距离行为。 因此，当群集中的数据点均匀分布时，群集中对象的LOF值通常接近1。 例如，在图4.6的情况下，即使两个聚类的密度不同，两个聚类中数据点的LOF值也将接近1。 另一方面，两个边远点的LOF值将高得多，因为它们将根据与平均邻居可达性距离的比率来计算。 LOF分数也可以被视为一个点的归一化可达性距离，其中归一化因子是其位置中可达性距离的调和平均值。 例如，公式4.3 可以重写如下：

$$LOF_k(\overline{X}) = \frac{AR_k(\overline{X})}{\text{HMEAN}_{\overline{Y} \in L_k(\overline{X})} AR_k(\overline{Y})} \tag{4.4}$$

这里，HMEAN表示其位置中所有点的可达性距离的调和平均值。 原则上，我们可以在分母中使用任何类型的均值。 例如，如果我们在分母中使用了可达性距离的算术平均值，则结果将具有类似的解释。 关于LOF方法的一个观察是，尽管在文献中普遍理解为基于密度的方法，但是可以更简单地将其理解为具有平滑的相对基于距离的方法。 基于可达性距离的局部分布计算相对距离。 LOF方法最初在[96作为基于密度的方法，因为它能够适应不同密度的区域。 密度松散地定义为点的平滑可达性距离的倒数。 当然，这不是密度的精确定义，传统上是根据指定区域或体积内的数据点数量来定义的。 本章的介绍省略了这个中间密度变量，既简单又直接根据可达性距离来定义LOF。 LOF与数据密度的真正联系在于其通过使用相对距离来调整到不同数据密度的洞察力。 虽然本书也将这种方法分类为基于密度的方法，但它可以通过宽松的密度定义或距离来等效地理解。

1. 可以使用原始距离而不是使用可达性距离。

2. 可以简单地使用算术平均值，而不是使用公式4.4中的调和平均值。 另一种称为LDOF（基于局部距离的离群因子）[610]的变体使用Lk（X）中的点之间的平均成对距离而不是算术平均值的调和平均值。

singola@gmx.com

通过这种修改，与基于距离的方法的关系更加明显。 还可以使用与数据相关的相似性度量（参见4.3.3节）与精确的k-最近邻居异常值检测器相结合，以实现与LOF类似的目标。

4.4.1.1 处理重复点和稳定性问题

公式4.4 中的调和平均值的使用对算法的稳定性有一些有趣的结果，因为调和平均值通常不是一组值的非常稳定的中心代表。 在一组值中出现单个值0将导致调和均值为0.这在包含重复点（或密集区域中彼此非常接近的点）的数据集中产生影响。 例如，即使是公式4.4 的分母中的一个可达性值为0，整个表达式将设置为。 换句话说，重复点附近的所有点都有将其分数设置为的风险。 很明显，这样的结果是不可取的。 请注意，使用算术平均值而不是公式4.4中的调和平均值可以减少此问题。 另一种可能性是使用k-distinct-distance，这基本上等同于从模型构造的数据集中删除重复[96]。 但是，只有当这些重复数据是数据中的噪声或错误的结果时，删除重复项才是有益的。 如果重复表示真密度，则计算的LOF分数将有偏差。 因此，很难使用k的解决方案-distinct-distance，对数据集中的重复项没有更深入的语义理解。 最后，一种合理的方法是通过使用小值α > 0修改公式4.4 来使用正则化：

$$LOF_k(X) = \frac{a + AR_k(X)}{a + \text{HMEAN}_{Y \in L_k(X)} AR_k(Y)} \tag{4.5}$$

直观地，α 的值调节数据点是正常点的先验概率。 尽管可以使用上述方法有效地处理重复的问题，但是谐波归一化确实导致关于参数k的值的更稳定的稳定性问题。 例如，在[96]中建议在一系列不同的k 值范围内使用LOFk（X）的最大值作为离群值[96]。 但是，k的值很小，一组点可能偶然相互接近，这将增加其所在地点的异常值。 可以将此问题视为前面讨论的重复问题的软版本。 由于谐波归一化的不稳定性，即使是单个紧密编织组也可以增加异常值在给定地点的许多点。

这种类型的不稳定性倾向于使LOF检测器比其他检测器（例如平均 k-最近邻方法）对k 值更敏感。 因此，如果可以确定LOF 的k 的最佳值，则可以获得比使用k 的最佳值对于其他检测器（例如精确或平均k-最近邻方法）获得的更好的LOF结果。 然而，在解释这样的结果时必须小心，因为对于诸如离群值检测的无监督问题，不可能知道特定检测器中k的正确值。 此外，（相对不稳定的）LOF在特定值下的良好结果 k 可能仅仅是过度配置的表现。 在实践中，更好的性能测量是计算一系列参数选择的平均性能的各种度量。 在这种情况下，LOF通常是由较简单的检测器的表现优于如确切k-nearest邻居检测器和平均k-nearest邻居检测器[32，35]。 实际上，永远不应低估可信赖的旧k最近邻探测器。

singola@gmx.com

4.4.2 LOCI：局部相关积分

[426]中提出的一种有趣的方法使用基于局部密度的方法进行离群分析。 LOCI方法确实是基于密度的方法，因为它根据点周围预先指定的半径s 内的数据点的数量来定义数据点X的密度M （X，s）。 这被称为数据点X的计数邻域。 相应地，X 的δ-邻域中的平均密度AM （X，s，δ）被定义为距离最多δ的所有数据点的M （X，s）的平均值。 从X。 δ 的值也称为X的采样邻域，并且总是大于s。 Further-更多的价值小号总是被选择为一个固定比例δ，不管是什么值 δ 被使用。 δ 的值是分析中的关键参数，并且使用该参数的多个值以便在不同的粒度级别提供分析性见解。 X 附近的平均密度正式定义如下：

$$AM(\overline{X}, s, \delta) = \text{MEAN}_{[Y:dist(X,Y)\leq\delta]} M(\overline{Y}, s) \tag{4.6}$$

相应地，水平δ 的多粒度偏差因子MDEF （X，s，δ）是根据一个点的密度与其邻域的平均密度之比来表示的：

$$MDEF(\overline{X}, s, \delta) = 1 - \frac{\overline{M}(\overline{X}, s)}{AM(\overline{X}, s, \delta)} \tag{4.7}$$

在使用局部比率同时定义数据点的离群值得分时，请注意与LOF的相似性。 MDEF的值越大，异常值得分越大。 为了将MDEF分数转换为二进制标签，计算X 的采样邻域内的M （X，s）的不同值的偏差σ（X，s，δ）。

$$\sigma(\overline{X}, s, \delta) = \frac{STD_{[Y:dist(X,Y)\leq\delta]} \overline{M}(\overline{Y}, s)}{AM(\overline{X}, s, \delta)}$$

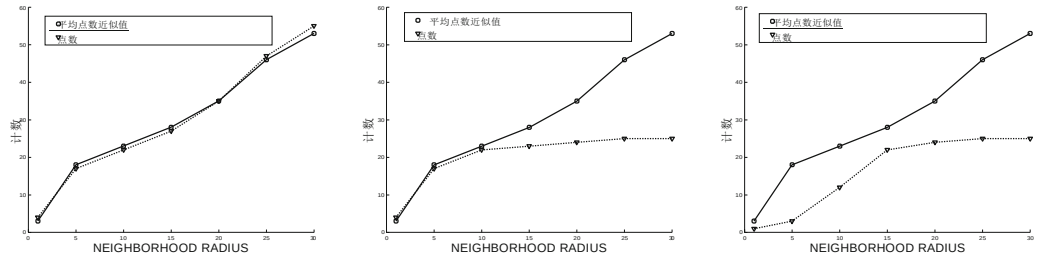
这里，函数STD 计算整个采样邻域的标准偏差。 分母解释了公式4.7 的MDEF值通过分母中的相同表达式进行缩放的事实。

s 的值总是选择为δ的一半，以便能够进行快速近似计算。 因此，在整个演示过程中，假设s 的值由δ的选择自动决定。 使用多个δ 值以便为异常值分析提供多粒度方法。 这些方法将采样半径从包含至少20个点的最小半径变为跨越大多数数据的最大半径。 如果在不同粒度级别计算的任何值中MDEF值异常大，则数据点是异常值。 具体来说，所述MDEF的值需要至少kσ（X，S，δ），其中k 被选择为3.这种k的选择在统计分析中使用正态分布假设是常见的。

通过以下修改可以提高算法效率：

- 只需要考虑一组有限的采样邻域。 特别是，如果采样或计数邻域不会因δ的微小变化而改变，则不需要考虑这些邻域。
- [426]中还提供了用于近似邻居计数的快速方法。 这为MDEF提供了高质量的近似值。它已被示于[426]，即数据的基于网格的划分的箱计数提供了一种快速的近似，当大号∞
  使用距离。 该近似也称为aLOCI算法。

singola@gmx.com



(a) 聚类点 （b）微聚类异常 （c）突出的异常

图4.7：没有显示范围的LOCI图的简化版本的说明性示例。 范围可以进一步帮助区分点的真密度何时低于其邻域的平均密度。 该图仅用于说明目的，仅显示在各种设置中可见的典型趋势类型。

4.4.2.1 LOCI情节

LOCI图压缩二维表示中有关数据点的信息，其中异常行为可从多粒度的视角中直观地解释。由于MDEF （X，s，δ）的值是通过检查M （X，s）和AM （X，s，δ）相对于δ的不同值的相对行为来构造的，因此可视化这些量中的每一个都是有意义的。 通过将它们分别绘制在采样邻域δ上。 因此，轨迹图显示的值δ 对X-轴对在以下基于计数的数量ÿ-轴：

- **点的密度行为：**的值中号（X，S）= 中号（$\overline{X}$，δ/2）绘制在ÿ 轴摆动。 这显示了数据点X 在不同的粒度水平下的实际密度行为。
- **附近的点的平均密度行为：**的值AM （X，S，δ）和 AM （X，S，δ）± STD$_{[\ddot{y}\,\in\,\overline{DIST(X,\,Y)\,\le\delta}]}$中号（Y，S）绘制在ÿ 轴摆动。 这显示了对于不同的粒度级别，X 的邻域密度行为（以及统计范围）。 范围有助于识别点的实际密度明显低于其邻域的真实密度的情况。

当数据点是异常值时，其密度行为通常会低于点邻域的密度行为，甚至可能低于邻域密度范围的下限。 LOCI图可以直观地了解数据点的偏差如何与不同粒度级别的极端偏差值相关联，并解释了为什么特定数据点可能具有较高的MDEF值。 使用不同的粒度级别有助于将算法调整到不同数据集的变幻莫测。 例如，在图4.2的情况下，任何基于距离的方法或LOF方法都需要选择k 的值（对于k）非常仔细地将这些数据点识别为异常值。 然而，LOCI图总是在视觉上实现正确的点特定级别的分析粒度。 在某些情况下，特定数据点可能仅在特定粒度级别显示为异常值，这也会显示在LOCI图中。 这种视觉洞察力有助于在无人监督的问题中建立直觉和可解释性像异常值检测。

LOCI图的说明性示例如图4.7所示。 在这里，我们已经展示了该图的简化版本，但未显示平均值的上下范围。

singola@gmx.com

年龄邻域密度。 换言之，我们已经表明只有邻域密度AM （X，S，δ）而不显示范围AM （X，S，δ）± STD[ŷ：DIST（X，Y）≤δ]中号（Y，S）。 这些范围进一步有助于区分点密度远低于邻近点的时间
硼浓度。 图4.7（a）显示了聚类点的典型行为，其中点的密度与半径的所有值上的邻域密度大致相似。 在群集异常的情况下，这种异常与少量点共同发生，两个密度只有在足够大的邻域大小（见图4.7（b））超出小群集的大小后才会发散。 由于小群体中出现异常值的倾向，因此群集异常往往是常见的。 在突出异常值的情况下，如图4.7所示（c），两个密度在一开始就有分歧。 因此，这些不同的图提供了关于为什么特定点应被视为异常值的见解。 值得注意的是，LOCI图是针对单个数据点的；因此，它通常仅在LOCI算法识别的一小组有趣点上使用。

4.4.3 基于直方图的技术

直方图使用空间分区方法进行基于密度的汇总。 在最简单的单变量数据的情况下，数据被离散化为最小值和最大值之间的相等宽度的区间，并且估计每个区间的频率。 位于频率非常低的箱中的数据点被报告为异常值。 在多变量数据的背景下，该方法可以以两种不同的方式推广：

- 针对每个维度单独计算离群值得分，然后可以聚合得分。
- 可以同时生成沿每个维度的离散化，并且可以构建网格结构。 可以使用网格结构中的点的分布以便创建稀疏区域的模型。 这些稀疏区域中的数据点是异常值。

在某些情况下，直方图仅根据点的样本构建（出于效率原因），但所有点都是根据点所在的箱的频率进行评分。 设 fı . . . fb 为b 单变量或多变量箱的（原始）频率。 这些频率代表这些箱内点的异常值。 较小的值更具指示性
异常性。 在某些情况下，通过将数据点减少1来调整数据点的频率计数（得分）。 这是因为在计数中包含数据点本身可以在极值分析期间掩盖其异常值。 如果使用数据样本来构建直方图，则此调整尤其重要，但是使用构建的直方图中的相关区间的频率对样本外点进行评分。 在这种情况下，仅采样点的得分减少1.因此，第j个点（属于具有频率fi的第i个bin ）的调整频率Fj 由以下给出：

$$F_j = f_i - I_j \tag{4.8}$$

这里，IJ ∈{0，1}是取决于是否指示变量Ĵ个点是样品中的点。 请注意，Fj 始终是非负的，因为包含样本内的bin
点的频率至少为1。

值得注意的是，调整后的频率Fj 的对数表示对数似然得分，这使得能够对得分到标签转换进行更好的极值分析。 为了规范化，我们使用$\log_2$（Fj +α）作为第j个点的离群值，其中α > 0是

singola@gmx.com

正则化参数。 要将分数转换为二进制标签，学生的t分布或正态分布可用于通过极值分析确定异常低的分数。 这些点被标记为异常值。 直方图与聚类方法非常相似，因为它们总结了数据的密集和稀疏区域，以便计算异常值; 主要的不同之处在于聚类方法对数据点进行划分，而直方图方法倾向于将数据空间划分为相同大小的区域。

基于直方图的技术面临的主要挑战是通常难以很好地确定最佳直方图宽度。 太宽或太窄的直方图不会在最佳检测异常值所需的粒度级别上对频率分布进行建模。 当容器太窄时，落在这些容器中的正常数据点将被识别为异常值。 另一方面，当箱太宽时，异常数据点可能落入高频箱中，因此不会被识别为异常值。 在这样的设置中，改变直方图宽度并获得相同数据点的多个分数是有意义的。 然后在不同的基础检测器上对这些（对数似然）分数取平均值，以获得最终结果。 像聚类一样 基于直方图的方法倾向于根据参数选择具有高度的预测可变性，这使得它们成为集合方法的理想候选者。 例如，RS-Hash方法（参见section 第5章的5.2.5 ）改变网格区域的维度和大小，以将得分作为异常值。 类似地，一些最近的集合方法，如隔离森林（参见第5.2.6节）可以被视为随机直方图，其中不同大小和形状的网格区域以分层和随机方式创建。 不是测量固定大小的网格区域中的点数，而是将隔离单个点所需的网格区域的预期大小的间接度量报告为异常值得分。 这种方法可以避免预先选择固定网格尺寸的问题。

使用基于直方图的技术的第二个（相关的）挑战是它们的空间划分方法使得它们对于聚类异常的存在是近视的。 例如，在图4.2的情况下 ，基于多元网格的方法可能无法将隔离的数据点组分类为异常值，除非仔细校准网格结构的分辨率。 这是因为网格的密度仅取决于其内部的数据点，并且当表示的粒度高时，一组孤立的点可以创建人工密集的网格单元。 也可以通过改变网格宽度和平均分数来部分解决此问题。

直方图方法在更高的维度上不能很好地工作，因为网格结构的维数随着维数的增加而增加，除非异常分数是根据精心选择的低维投影计算的。 例如，d维空间将包含至少2个^d 网格单元，因此，预期填充每个单元格的数据点的数量随着维数的增加呈指数减小。 如4.4.5节所述，使用一些技术，如旋转装袋[32]和子空间直方图（参见第5章第5.2.5 节））可以部分解决这个问题。 虽然基于直方图的tech- niques具有显着的电势，它们应该在结合使用高维子空间合奏（见第5 和6）以取得最佳效果。

4.4.4 核密度估计

在建立密度分布方面，核密度估计方法类似于直方图技术。 但是，通过使用内核函数而不是离散计数来构造更平滑的密度配置文件。 由于两类方法之间的相似性，人们倾向于在这两种情况下获得类似的结果，特别是如果使用集合方法来平滑实现和参数特定的影响。

singola@gmx.com

在核密度估计[496]中，也称为Parzen-Rosenblatt方法，在给定点处产生密度的连续估计。 给定点处的密度值估计为核函数K_h^j（）的平滑值之和
与数据集中的每个点相关联。 每个内核函数与调整平滑级别的内核宽度h 相关联。 基于N个数据点和核函数 K_h^j（·）的核密度估计值f（X）定义如下：

$$\hat{f}(\overline{X}) = \frac{1}{N} \sum_{i=1}^{N} K^j(h\overline{X} - \overline{X})^i \tag{4.9}$$

因此，数据集中的每个离散点 Xi 被连续函数K_h^j（·）代替，该函数 在Xi 处达到峰值并且具有由平滑参数h确定的方差。 一个
例如，这样的分布将是高斯核与宽度h 对于d维数据：

$$K_h^j(\overline{X} - \overline{X}_i) = \frac{1}{h\sqrt{2\pi}}^{\Sigma_d} \cdot e^{-\frac{||\overline{X}-\overline{X}_i||^2}{2h^2}} \tag{4.10}$$

估计误差由内核宽度h定义，内核宽度h以数据驱动的方式选择。 已经证明[496]对于大多数平滑函数K_h^j（），当数据点的数量达到最大值时，估计量f（X）渐近收敛到真密度。
函数f （X），条件是适当选择宽度h 。 对于包含N 个点的数据集，Silverman近似规则[496]建议带宽为h = 1。 06 $\hat{\sigma}N^{-1/5}$，其中的$\hat{\sigma}$是所估计的样本方差。 然而，这种选择只是一个经验法则；
通常，带宽的最佳选择是数据特定。 核密度估计在a
给定点是数据驱动的估计生成数据分布的概率密度函数的值。 [184]中使用[316]中提出的密度估计方法显示了良好的结果。 应该从公式4.9 的总和中排除测试点以避免过度拟合，这对于某些类型的核函数尤其重要。

可以将核密度估计视为由第2章中的期望最大化算法计算的fi t值的简单非参数替代 。 因此，核密度估计可以被视为异常值得分，其中较小的值表示更大程度的离群值。 具有异常低密度的数据点通过使用学生t分布或正态分布假设来声明异常值。 但是，对数函数应该应用于极值分析之前的分数，因为这种分析对对数似然分数更有效。

基于密度的方法面临与直方图技术类似的实现挑战。 正如网格宽度的选择导致直方图方法的两难选择一样，在核密度方法中正确选择带宽通常是数据分布特定的，这是先验不可知的。 此外，在局部密度存在很大差异的情况下，如图4.2 和4.6所示，使用全球带宽来估算密度可能效果不佳。 。 在某种意义上，带宽在核心密度估计中起到与直方图方法中的网格宽度类似的作用，因为它们都调节估计过程的局部性大小。 此外，这些方法对于更高的维度不是非常有效，因为密度估计过程的准确性随着维数的增加而降低。

4.4.4.1 与谐波k-最近邻检测器连接

定义4.3.3 的谐波k-最近邻检测器是基于密度的方法的一种特殊情况。 具体来说，核函数只是设置为与距离的反距离

singola@gmx.com

目标。 我们在这里复制公式4.9 ：

$$\hat{f}(X) = \frac{1}{N} \cdot \sum_{i=1}^{N} K^j \left(\frac{X - X_i}{h}\right)$$
$$\propto \frac{1}{N} \cdot \sum_{i=1} 1/||\overline{X} - \overline{X_i}||$$

注意，当在定义4.3.3中将k 的值设置为N 时，第二个表达式只是谐波分数的倒数。 删除任何与X 相同的训练点Xi 非常重要，以避免因过度训练而导致的分数。 请注意，此内核函数不使用带宽，因此它是一个无参数检测器。 人们也可以仅使用最近的k ＜N 个点来代替带宽进行估计，尽管通常在k = N时获得相当好的结果。 值得注意的是（算术上）平均k当k 设置为N 时，最近邻检测器不能发现数据分布中心附近的孤立点。 例如，在一个点环中心的单个异常值将在平均N 最近邻检测器中接收到最类似内部的分数。 然而，谐波N 最近邻检测器将在适当地对异常值进行评分方面做得非常出色，特别是如果N 的值很大。 这是因为使用更大的数据集总是可以更准确地估计内核密度。

随着点数的增加，这种方法确实将成为相对密度的良好启发式估计。 然而，在许多情况下，通过使用不同的数据样本重复估计f（X）并对分数求平均值，可以获得更好的结果
创建一个集合方法。 与所有基于密度的方法一样，在应用极值分析或集合方法之前，应将对数函数应用于f（X）（将其转换为对数似然）。

4.4.4.2 核方法的局部变化

通过单独或组合使用各种技巧，可以使内核方法对局部性更敏感：

1. 用于估计点处的密度的带宽可以通过使用其k最近邻距离而不是使用全局值来本地计算。

2. 可以使用其相邻点处的平均核密度来标准化一点处的核密度。

这两个技巧在[342]中结合使用。 此外，核函数的指数中的距离被替换为可达性距离（如LOF算法所定义）。

4.4.5 直方图和核方法的基于集合的实现

使用集合方法可以显着加强基于直方图和内核的技术。 基于直方图的技术的主要问题是在高维数据的背景下有效地使用它们。 在这种情况下，有可能设计出有效的实施方案。
使用旋转装袋[32]进行高维数据的思考，b√因为旋转
装袋大大减少了数据的维数从O（d）到O（ d）。 基础的

singola@gmx.com

想法是生成一个随机旋转的维度2+ |√d/2 |的子空间和项目
在应用基于多维数字直方图的方法之前，该子空间中的数据点
（或任何其他探测器）。 人们可能会选择甚至不到2 + | d /2 | 尺寸改善 -
评估高维度的影响。 描述了旋转过程的具体细节
在第5 章第5.3.3 节和第6章第6.4.4 节中。

这种方法使用不同的随机投影重复多次，并且对不同集合分量上的点的分数进行平均，以便计算最终的异常值分数。 旋转装袋的基于自然集合的方法隐含地构造了比由基础探测器构造的简单直方图更复杂的模型。 结果，可以大大提高精度。 请注意，旋转装袋被设计为[32]作为任何基础检测器的元算法，而不仅仅是基于直方图的方法。

一个相关的方法，反思为LODA [436]，显示了如何使用1维
具有d 个非零元素的随机投影，以便有效地对数据点进行评分。
直方图建立在包含d 非零矢量的一维投影上
元素，每个元素都是从标准正态分布中提取的。 直方图用于计算每个数据点的对数似然得分，因为它很容易从基于直方图和核密度的方法获得概率分数。 来自不同集合分量的每个点的对数似然得分被平均，以便计算最终离群值得分。 这里的基本思想是，即使一维直方图是非常弱的探测器，通常也可以在整体中组合这些弱探测器，以便创建非常强的离群值探测器。 LODA方法可视为旋转套袋的特殊情况[32]其中一维投影与基于直方图的检测器组合以获得高质量的结果。 RS-Hash方法对不同维度的轴平行子空间进行采样，以将数据点评分为异常值。 当在集合中心设置中使用直方图和核方法时，在对分数求平均值之前将对数函数应用于估计的密度是很重要的。

4.5 基于邻近检测的局限性

大多数基于接近度的方法使用距离来定义不同粒度级别的异常值。 通常，需要更高级别的粒度才能获得更高的准确性。 特别是，通过各种形式的摘要抽象数据的方法在真正的异常和低密度的噪声区域之间并没有很好地区分。 这种弱点体现在预测中具有不同类型的总结的高度可变性。 在这种情况下，总结过程中变幻莫测的预测变化需要通过集合方法来改善。 此外，这些方法需要仔细地结合全局和局部分析，以便找到数据中的真实异常值。 如图4.1 和图4所示，完全全局分析可能会遗漏重要的异常值4.6，而完全局部分析可能会错过离群值小集群组，如图所示4.2。 同时，增加分析的粒度会使算法效率低下。 在最坏的情况下，可能需要具有全粒度的基于距离的算法
包含N个记录的数据集中的O（N 2）距离计算。 虽然索引方法可以用于将修剪结合到异常值搜索中，但是有效性
由于数据稀疏性，修剪方法随着维数的增加而减少。

在高维数据的背景下，更加基本的限制不是效率，而是发现异常值的质量。 在高维的情况下，所有点变得几乎彼此等距的，因此在距离对比度丢失[25，263]。 这也被称为维度的诅咒，它出现了

singola@gmx.com

从数据稀疏性来看，它对许多高维应用产生了负面影响[8]。 随着维数的增加，大多数特征都不能用于异常值检测，并且这些维度的噪声影响将以非常消极的方式影响基于邻近度的方法。 在这种情况下，异常值可以被特征中的噪声掩盖，除非可以通过异常值检测方法明确地发现相关维度。 由于基于接近度的方法自然地设计为使用数据中的所有特征，因此随着维度的增加，它们的质量自然会降低。 确实存在一些方法用于通过子空间方法来提高这些方法在增加维数方面的有效性。 这些方法将在第5章中讨论。

4.6 结论和总结

本章概述了离群点分析的基于邻近度的关键技术。 所有这些方法都使用数据点之间的邻近信息来确定数据中的异常值。 这些方法与补充意义上的聚类技术密切相关；虽然前者在稀疏数据位置发现异常点，但后者试图确定密集数据位置。 因此，聚类本身是基于邻近的异常值分析中常用的方法。 通过正确选择聚类方法，还可以使该方法适应数据中任意形状的模式。 这些方法也可以扩展到任意数据类型。

由于易于实现和可解释性，基于邻近的方法在文献中广泛流行。 使用这种方法的主要挑战是它们通常是计算密集型的，并且大多数高质量方法在最坏情况下需要O（N 2）个距离计算。 可以使用各种修剪方法来改进
当需要二进制标签而不是分数时，它们的效率。 在各种修剪方法中，采样方法是最有效的方法之一。 局部归一化是用于改善各种异常值检测算法的有效性的原理。 两种常见的本地算法包括LOF和LOCI算法。 最后，还在文献中探索了基于直方图和核密度估计的技术。 虽然这些方法并没有得到很大的普及，但当它们与集合方法结合使用时，它们具有显着的潜力。

4.7 书目调查

多变量异常值的传统定义通常认为它们是聚类算法的副产品。 因此，异常值被定义为不会自然地进入任何聚类的数据点。 但是，群集中数据点的非成员资格无法区分噪声和异常。 二FF erent聚类algo- rithms及其对异常值分析使用的详细讨论可以在[发现283，307]。 许多聚类算法明确地将不在聚类内的点标记为异常值[594]。 然而，在一些稀疏的高维域中，例如事务数据，（子空间） - 聚类是用于识别异常值的少数可行方法之一[254]。

为了提高基于聚类的离群值分数的质量，可以使用数据点到聚类质心的距离，而不是仅使用聚类中数据点的成员资格。 [499]中的工作研究了许多用于聚类的确定性和概率方法，以便检测异常。 这些技术是在入侵检测的背景下设计的。 这种方法的一个挑战是防止聚类方法因数据中已经存在的噪声和异常而导致质量降低。 这是

singola@gmx.com

因为如果发现的星团已经被噪声和异常偏倚，它也会阻止异常值的有效识别。 这样的技术已在入侵检测的应用[上下文经常使用70，499]。 [70]中的工作使用第一阶段，通过使用与数据中的频繁模式匹配的数据点来识别正常数据。 随后使用这些正常数据以执行稳健的聚类。 然后将异常值确定为与这些簇位于显着距离的点。 这些方法可以被认为是一种顺序集合方法。

一些离群点检测方法也被建议用于在异常可以位于小簇[例188，253，291，420，445，446]。 这些技术中的许多技术通过使用距离阈值来工作以调节新簇的创建。 当数据点不在最近的群集质心的指定阈值距离内时，将创建包含单个实例的新群集。 这导致不同大小的聚类，因为一些新创建的聚类没有获得足够多的点添加到它们。 然后，数据点的离群值可以通过其簇中的点数以及其簇与其他簇的距离来确定。 一些索引技术也已在为了提出的速度向数据点的划分成群集[131，518]。 有偏见的采样[321]也被证明是一种有效且高效的基于聚类的异常值检测方法。 在[422]中提出了用于集成聚类和异常值检测的设施位置模型。 [475]中的工作展示了如何构建用于异常值检测的鲁棒频谱嵌入。 其中许多方法也可以扩展到任意数据类型。

基于距离的方法在文献中非常流行，因为它们能够以比聚类方法更高的粒度级别执行分析。 此外，这些方法直观且易于理解和实施。 [317]提出了第一种基于距离的方法。 这项工作的想法扩展到[318]中发现内涵知识。 随后，设计了索引方法以提高[456]中该方法的效率。 [58]中的工作使用线性化，其中多维空间填充有希尔伯特空间填充曲线。 这个1-d表示具有k的优点通过检查空间填充曲线上数据点的前身和后继，可以非常快速地确定最近邻居。 使用线性化表示上的k个最近邻距离的总和，以便生成数据对象的离群值得分。 而采用的总和的k̨-nearest相邻距离具有超过一定的优势k̨在人口稀少的数据和集群数据之间二FF erentiating -nearest相邻距离，它具有的缺点（有时）不能够检测的基孤立的异常现象如图4.2所示。 使用空间填充曲线的一个挑战是它们将数据映射到带有d 的超立方体
这个超立方体的尺寸和角数随着指数的增长呈指数增长
d。 在这种情况下，高维数据的稀疏性可能导致空间填充曲线的局部性行为的退化。 为了解决这个问题，[58]中的工作使用数据转换技术来改善局部性。 设计了一种迭代技术，需要对数据集进行d + 1次扫描。

[75]中的工作设计了一种简单的基于采样的修剪技术，以提高基于k最近邻的异常值检测技术的效率。 核心思想类似于[456]中使用的修剪规则。 这个想法是，如果一个对象的异常值得分小于第r个最佳异常值的k最近邻距离，那么该数据点不可能是异常值，并且可以从进一步的考虑中删除。 [75]中显示了这种简单的修剪规则，可以很好地处理随机数据。 随机化本身可以通过使用基于磁盘的shu ffling技术在线性时间内完成。 [572]为了改进，在较小的数据集样本上执行最近邻计算

singola@gmx.com

效率。 提供理论保证是为了限制采样过程导致的准确性损失。

修剪方法的有效性显然取决于能够以高效的方式在k-最近邻距离上产生良好界限的能力。 因此，[219]中的工作将数据划分为小集群。 所述ķ集群内的数据点的邻居-nearest距离，以便产生所述的上限使用ķ该点的邻居-nearest距离。 如果该上限小于已经找到的异常值集的分数，则可以从考虑中修剪该点。 [456]中的工作也使用聚类技术进行修剪。 [219]使用递归分层分区，以确保为每个群集分配相似数量的数据点。 使用沿最大方差的主分量的数据点的排序，以便提供k最近邻距离的快速估计。 在某些情况下，可以通过使用k-最近邻方法中的数据相关相似性度量来执行更准确的异常值检测。

的数据相关的相似函数的讨论提供于[设置33，491，548]。 对依赖于数据的相似性的方法包括共享近邻测量[287]，所述成对的全局/局部马哈拉诺比斯适应（参见第3章[33]），随机森林[99，491]，随机聚类树[401，555]和隔离森林[548]。 值得注意的是，隔离森林可被视为极端随机聚类森林（ERC-Forests）的改编[401]]在每个节点进行一次试验，并将树木生长到全高，叶子上有单个点。

在[190]中提出了一种基于分辨率的方法。 根据该方法，点是属于簇还是属于异常取决于距离阈值。 在最高分辨率级别，所有点都在各自的聚类中，因此是异常值。 随着分辨率逐渐降低，越来越多的数据点加入集群。 在每个步骤中，每个点从异常值变为群集。 基于此，在[190]中定义了分辨率离群因子（ROF）值。 这被证明可以为异常值分析提供有效的结果。

大多数基于距离的算法都是使用欧几里德距离设计的。 在实践中，欧几里得函数可能不是找出异常值的最佳函数。 事实上，对于许多其他数据领域，距离函数通常以相当复杂的方式定义，并且许多为欧几里德空间设计的修剪技术在任意空间中都不能很好地工作。 在这种情况下，一个高效的算法被设计用于任意度量空间中的离群值检测[533]，这需要最多三遍数据。

在[430]中提出了一种利用参考点来提高基于距离的算法的效率的方法。 这项工作的核心思想是根据相对于一组固定的R 参考点的相对密度来对数据点进行排序。 每个数据点基于它们到参考点的距离以R 可能的方式变换为1维空间。 对于这些R中的每一个在1维数据集中，计算每个数据点相对于对应参考点的相对密度。 数据点的总体相对密度程度被定义为所有参考点上的最小相对密度程度。 这种相对密度提供了一种对不同数据点进行排名的方法。 在[83]中提出了用于加速异常值检测的分布式算法。

可伸缩性是数据流环境中的一个重要问题。 通常，在数据流的情况下，使用过去的历史窗口来确定异常值。 数据中心，其ķ-nearest邻居值是在一个特定的ç滑动窗口的历史大声明异常值[60，322]。 [28]中的流聚类方法也可用于加速异常值分析过程。 这种方法已在[322]中讨论过。

singola@gmx.com

在异常值分析的上下文中局部密度的问题是第一个在[发给96，97]。 我们注意到，反向最近邻方法[100，248]在适应局部密度与使用反向最近邻法的条款本章股一些相似之处LOF呈现。 局部密度的变化可能导致基于全局距离的方法对异常值的排序较差。 因此，在[96]中提出了局部异常因子（LOF）的概念]。 这些方法基于局部密度调整对象的离群值。应该提到的是，密度的概念在LOF中实际上是松散定义的，是平均距离的倒数。 真正的密度定义应该真正计算特定体积中的数据点数量。 高密度局部区域的数据点具有较高的异常值，即使它们与其所在地的其他点略有隔离。 在[32]中提供了LOF算法与平均k-最近邻算法的比较。 结果表明，平均k最近邻算法更稳健。

随后提出了广泛的LOF方法的许多不同变体。 例如，在[工作293]提出顶的概念Ñ 本地异常值，其中，所述顶 Ñ 异常值测定密度的基础上。 修剪技术用于通过将数据划分为簇来计算运行时间，并计算每个簇中点的LOF值的界限。 因此，如果它们被保证仅包含具有比最弱的当前顶的下部LOF值的点的簇全部可以被修剪Ñ 离群值。 用于改善顶的电子FFectiveness其它方法Ñ 本地异常值检测与使用簇的修剪在[提出144]。

LOF的一个问题是，当不同密度的区域没有明确分开时，它有时会变得无效。 因此，[294]的INFLO技术提出了LOF的修改，它使用基于最近邻距离的对称最近邻关系以及反向最近邻距离来定义局部异常值。 在[534]中提出了基于连通性的离群因子（COF）的概念]，它还能够有效地找到低密度或任意形状区域的异常值。 COF和LOF之间的主要差异是COF定义数据点邻域的方式。 具体而言，通过将最近点添加到当前邻域集来递增地定义邻域。 这种方法基于单链接聚类的类似动机，因为它基于与单链接聚类完全相同的标准将邻域集（视为聚类）和点（视为单例聚类）合并。 因此，当点分布在数据的任意低维流形上时，它可以有效地定义任意形状的邻域。 LOF方法也与其他聚类技术相结合。 例如，[253，255]定义称为基于群集的局部异常因子（CBLOF）的分数，其中异常被定义为到附近群集的本地距离和数据点所属的群集的大小的组合。 与附近群集距离较远的小群集中的数据点会被视为异常值。

LOF方案也已扩展到具有非空间属性的空间数据的情况[523]。 例如，海面温度是空间位置背景下的非空间属性。 已知这样的数据表现出空间自相关，其中元素的值是由其直接邻居（例如，空间温度局部性）影响的。 此外，数据显示空间异方差性，其中数据点的方差基于其位置。 例如，“正常”温度变化显然基于地理位置。 我们注意到，从空间连续性的角度来看，空间数据与时间数据有一些相似之处，这类似于时间连续性。 相应地，[523]定义一个局部异常值测量，称为空间局部异常值测量（SLOM），特别适用于空间异常值检测。

singola@gmx.com

在[443]中讨论了LOF方法对流式方案的推广。

LOCI方法[426]也是一种局部敏感方法，它使用点周围的圆形邻域中的点数，而不是k的倒数。 - 局部密度计算的最近邻距离。 因此，从直观的角度来看，它确实是一种基于密度的方法。 此外，该方法在不同的粒度级别上进行测试，以减少参数选择，并在异常值检测过程中消除对某些输入参数的需要。 算法的近似版本可以在几乎线性的时间内实现。 这项工作的一个有趣的贡献是引入LOCI图，通过视觉图提供对数据中异常值的直观理解。 LOCI图提供了对不同大小的邻域如何对应于数据点的离群值得分的理解。

基于密度的异常值分析的传统方法涉及使用离散化，基于网格的方法和基于内核密度的方法。 的前两种属于在基于直方图的方法[一般类别260，288]。 基于直方图的技术在入侵检测技术中具有广泛的适用性，其中自然地构建基于频率的各种事件的概率。 基于直方图的方法的主要挑战是沿着每个维度的桶尺寸有时难以正确选取。 [476]提出了RS-Hash方法，通过在以集合为中心的方法中随机改变不同网格区域的大小和维度，在线性时间内执行子空间异常值检测。 该方法也已扩展到同一工作中的数据流。 一个密切相关的方法是内核密度估计的[262，496]。 这种方法的局部变化在[讨论342，483]。 核密度估计是基于网格的方法的连续变化，其中平滑核函数用于估计过程。 在[316]中提供了一种特别稳健的核密度估计形式，已经显示[184]以实现异常检测的良好结果。]。

4.8 演习

1. 考虑具有以下观察的数据集：{（1,3），（1.01,3.01），（0.99,3.01），（0.99,3），（0.99,2.99），（3,1）}。
   - 使用一维PCA对该数据集进行线性建模的结果是什么？
   - 在这种情况下，1-NN技术如何有效地发现异常值？
   - 讨论在这种情况下使用PCA的主要挑战。 你有什么办法吗？可以提高PCA的得分吗？
2. 考虑包含具有点{（1,1），（0,0），（2,2.1），（3,3.1），（4,4），（5.1,5）}的单个簇的数据集。
   - 考虑的两个点（6。 5，6。 5），和（1，2。 1）。 在一张纸上画点。 两个数据点中的哪一个看起来更像是一个异常值？
   - 1-NN算法在哪个点设置为欧几里德度量的最高异常值？
   - 1-NN算法在哪个点设置为欧几里德度量的最低异常值？
   - 当残差为异常值时，哪个数据点基于排名-1 PCA的算法设置为最高离群值？

singola@gmx.com

- 你会建议改变欧几里德元的距离函数吗？怎么样？

3. 从UCI机器学习库[203]下载Ionosphere数据集。

   - 当仅使用前3个特征向量时，在PCA方法中基于其残差分数对数据点进行排名。
   - 对于k的值，基于其k最近邻居分数对数据点进行排名从1到5不等。
   - 规范化数据，使每个维度的方差为1.根据k的最近邻居分数对数据点进行排序，对于k的值，范围从1到5。
   - 使用不同方法排名前5位的异常值中有多少数据点？
   - 现在使用投票方案，将不同方案中的异常值排列在一起。 现在排名前5位的离群值是多少？
   - 这种整体方法是否提供更强大的异常值？

4. 使用UCI机器学习库中的网络入侵数据集重复练习3。

5. 一家制造公司生产二维方形小部件，每侧通常分布有1米的长度，标准偏差为0.01米。

   - 从此分发中生成包含100,000个小部件的数据集。
   - 由于制造过程中存在缺陷，该公司生产了5个异常小部件。 每个这样的小部件的方形长度为0.1米，标准偏差为0.001米。 使用正态分布假设生成这5个异常点。
   - 1-NN方法是否会找到异常小部件？
   - 10-NN方法是否会找到异常小部件？

6. 对练习5中的数据集应用k-means聚类方法，其中使用了5个聚类质心。 作为后处理步骤，删除具有10个或更少数据点的任何群集。 根据数据点与最近的群集质心的距离对数据点进行评分。 哪些数据点的异常值得分最高？

7. 对练习5的情况应用反向1-NN算法。 哪些数据点具有最高的离群值？使用反向10-NN算法，哪些数据点具有最高的离群值？使用反向100-NN算法？

8. 使用LOF方法重复练习3和4，并确定异常值的排名。 在这种情况下异常值是否与练习3和4中的异常值相同？

9. 使用LOCI方法重复练习8。 异常值是否相同？

singola@gmx.com

第5章

高维异常值检测：子空间方法

> “鉴于我们在上述章节中所说的一切，我们似乎已经克服了许多障碍，是什么给我们的胜利庆祝蒙上阴影？它是维度的诅咒，是一种从早期就困扰科学家的诅咒。“ - 理查德·贝尔曼

5.1 介绍

许多真实的数据集都是非常高的维度。 在某些情况下，真实数据集可能包含数百或数千个维度。 随着维数的增加，许多传统的离群值检测方法无法有效地发挥作用。 这是众所周知的维度诅咒的人造物。 在高维空间中，数据变得稀疏，并且当在全维度中分析时，真正的异常值被多个不相关维度的噪声效应掩盖。

维数诅咒的一个主要原因是在高维情况下定义一个点的相关局部性的困难。 例如，基于接近度的方法通过在所有维度上使用距离函数来定义局部性。 另一方面，所有尺寸可能与特定测试点无关，这也影响了基础距离函数的质量[263]]。 例如，所有点对在高维空间中几乎是等距的。 这种现象被称为数据稀疏性或距离集中。 由于异常值被定义为稀疏区域中的数据点，因此这导致差的判别性情况，其中所有数据点在全维度中位于几乎相同的稀疏区域中。 维度诅咒带来的挑战并非特定于异常值检测。 众所周知，许多问题，如随着维[集群和相似性搜索体验质量的挑战5，7，121，263]。 实际上，有人提出几乎所有基于接近概念的算法都会在高维空间中定性地降级，因此需要

© Springer International Publishing AG 2017

C.C. Aggarwal, *Outlier Analysis*, DOI 10.1007/978-3-319-47578-3_5

singola@gmx.com

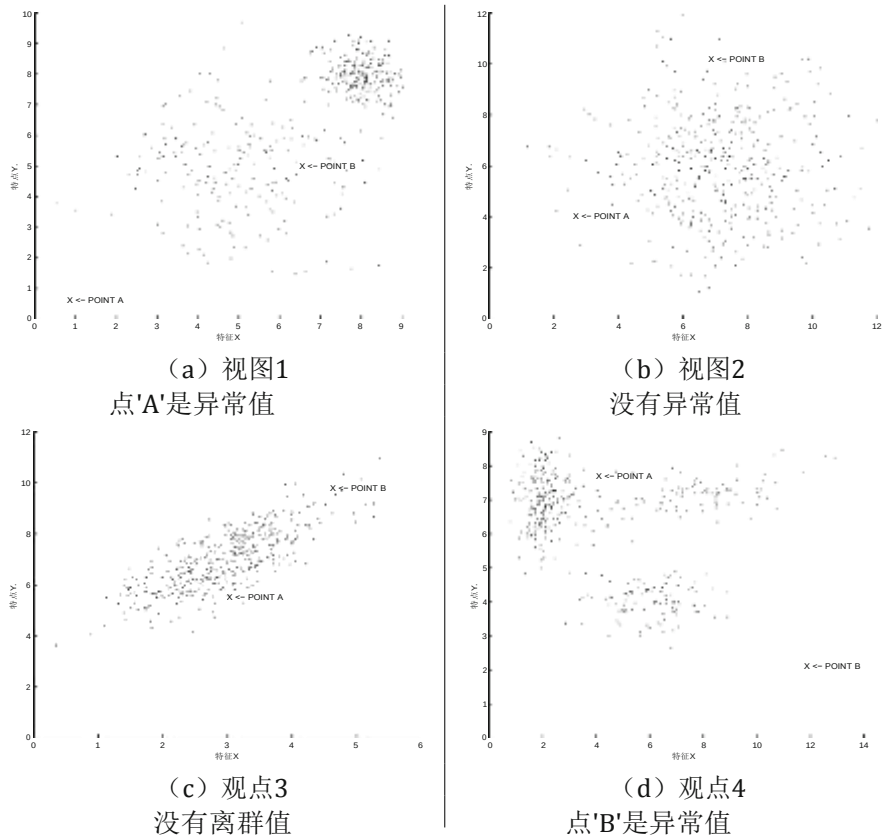


图5.1：异常行为被高维度的无关属性所掩盖。

以更有意义的方式重新定义[8]。 维数诅咒对离群值检测问题的影响首先在[4]中提到。

为了进一步解释全维异常值分析算法的无效性的原因，将提出一个激励性的例子。 在图5.1中，已经说明了假设数据集的四个不同的二维视图。 这些视图中的每一个对应于不相交的维度集。 很明显，点'A'在数据集的第一个视图中作为异常值暴露，而点'B'在数据集的第四个视图中作为异常值暴露。 但是，数据点“A”和“B”都不会在数据集的第二个和第三个视图中作为异常值公开。 因此，从测量“A”和“B”的异常值的角度来看，这些视图是嘈杂的。 在这种情况下，四个视图中的三个对于暴露任何特定的异常值“A”或“B”都是非信息性的并且噪声很大。 在这种情况下，当在全维度中执行距离测量时，异常值在这些视图内的随机分布中丢失。 这种情况通常会随着维度的增加而自然放大。 对于非常高维度的数据集，可能只有极小部分的视图可以为异常值分析过程提供信息。

上述图示说明如何告诉我们有关当地相关维度的问题？在实际情况下，对这种情况的物理解释非常直观。 物体可能具有多个测量量，并且该物体的显着异常行为可能仅在这些量的一小部分中反映出来。 对于前

singola@gmx.com

充分考虑一种飞机机械故障检测方案，其中不同测试的结果以不同的尺寸表示。 在同一平面上进行的数千次不同机身测试的结果可能大部分是正常的，有一些噪声变化，这些并不重要。 另一方面，一小部分测试中的一些偏差可能足以指示异常行为。 当来自测试的数据以完全维度表示时，异常数据点在几乎所有数据视图中将显示为正常，除了维度的很小一部分。 因此，聚合邻近度量不太可能暴露异常值，因为大量正常测试的嘈杂变化将掩盖异常值。 此外，当测试不同的对象（不同机身的实例）时，不同的测试（维度子集）可能与识别异常值有关。 换句话说，异常值通常嵌入在局部相关的子空间中。

这对于这种场景中的全维分析意味着什么？当使用全维距离来测量偏差时，大量“正常噪声”维度的稀释效应将使异常值的检测变得困难。 在大多数情况下，这将显示为距其他尺寸噪声的距离 - 浓度效应。 这可能使计算更加错误。 此外，存在于大量不同尺寸的噪声的附加效应将干扰实际偏差的检测。 简单来说，当使用全维分析时，由于全维计算中噪声的掩蔽和稀释效应，在低维子空间中会丢失异常值[4]。

对于其他基于距离的方法（例如聚类和相似性搜索）也经历了类似的效果。 对于这些问题，已经示出[5，7，263通过检查子空间中数据的行为，可以设计出更具意义的集群，这些集群是特定于所讨论的特定子空间的。 对于离群值检测问题，这种广泛的观察通常也是正确的。 由于异常值可能仅在数据的低维子空间中被发现，因此探索感兴趣偏差的较低维子空间是有意义的。 这种方法可以滤除大量尺寸的加性噪声效应，从而产生更强大的异常值。 一个有趣的观察是，即使在具有缺失属性值的数据集中，也经常可以识别这种低维投影。 这对于许多实际应用非常有用，其中特征提取是一个不同的过程，并且通常不存在完整的特征描述。 例如，在机身故障检测场景中，可能仅应用了一部分测试，因此仅维度子集中的值可用于离群值分析。 该模型被称为预计异常值检测，或者，子空间离群值检测[4]。

相关子空间的识别是一个极其具有挑战性的问题。 这是因为高维数据的可能投影的数量与数据的维度呈指数相关。 有效的异常值检测方法需要以集成的方式搜索数据点和维度，以便揭示最相关的异常值。 这是因为不同的维度子集可能与不同的异常值相关，如图5.1中的示例所示。 这进一步增加了计算复杂性。

一个重要的观察结果是，在异常检测问题的背景下子空间分析通常比在诸如聚类等问题的情况下更加困难。 这是因为像聚类这样的问题是基于聚合行为，而异常则是通过定义来实现的。 因此，在异常值分析的情况下，与基于聚类的问题（如聚类）相比，给定位置中各个维度的统计聚合通常为子空间探索过程提供非常弱的提示。 这样的时候

singola@gmx.com

弱提示导致省略相关维度，效果可能比包含不相关维度更加激烈，特别是在局部相关维度的数量是完整数据维度的一小部分的有趣情况下。 一个常见的错误是假设聚类和异常值分析之间的互补关系可以扩展到局部子空间选择的问题。 特别是，早期子空间聚类方法的维度选择方法的盲目适应，不知道子空间分析原则在不同问题上的细微差别，有时可能会遗漏重要的异常值。 在这种情况下，认识到识别异常分析的相关子空间的困难也是至关重要的。 一般来说，为每个数据点选择单个相关子空间可能会导致不可预测的结果，因此组合多个子空间的结果非常重要。 换句话说，子空间离群值检测固有地构成了以集合为中心的问题。

通常使用几类方法：

- 基于稀疏**度**：这些方法试图基于底层分布的稀有性来发现子空间。 这里的主要挑战是计算，因为稀有子空间的数量远远大于高维度中的密集子空间的数量。

- **无偏**：在这些方法中，子空间以无偏的方式进行采样，并且在采样的子空间中组合得分。 当从原始属性集中对子空间进行采样时，该方法称为特征装袋[344]。 在对任意定向的子空间进行采样的情况下，该方法称为旋转装袋[32]或旋转子空间采样。 尽管它们非常简单，但这些方法通常都能很好地工作。

- **基于聚合**：在这些方法中，聚合统计数据（如聚类统计，方差统计或数据的全局或局部子集的非均匀性统计）用于量化子空间的相关性。 与基于稀有度的统计不同，这些方法量化全局或局部参考点集的统计属性，而不是尝试直接识别很少填充的子空间。 由于这些方法仅提供用于识别相关子空间的弱（并且易于出错）的提示，因此多子空间采样是至关重要的。

本章安排如下。 子空间离群检测的轴并行方法在5.2节中进行了研究。 底层技术讨论了如何组合多个子空间来发现异常值。 在5.3节中讨论了在广义子空间（即任意定向的子空间）中识别异常值的问题。 本节还讨论了在非线性子空间中发现异常值的最新方法。 子空间分析的局限性在5.4节中讨论。 结论和摘要见5.5节。

5.2 轴平行子空间

子空间异常检测[4]的第一个工作提出了一个模型，其中异常值由轴平行子空间定义。 在这些方法中，异常值是在原始数据的特征子集中定义的。 显然，需要仔细量化来比较各种子空间的得分，特别是如果它们具有不同的维度并使用不同的参考尺度。 此外，需要方法来量化

singola@gmx.com

暴露异常值时各种子空间的有效性。 轴并联方法使用的方法有两个主要变化：

- 在一类方法中，逐个检查点，并识别它们相关的外围子空间。 这本质上是一种基于实例的方法。 这种类型的方法在计算上是昂贵的，因为可能需要大量的计算时间来确定每个点的离群子空间。 然而，该方法提供了更细粒度的分析，并且它对于提供内涵知识也很有用。 这种内涵知识对于描述为什么特定数据点是异常值很有用。

- 在第二类方法中，通过预先构建子空间模型来识别异常值。 每个点都根据模型进行评分。 在某些情况下，每个模型可以对应于单个子空间。 通常使用从不同模型获得的结果的集合分数对得分进行评分。 即使在模型中使用单个（全局）子空间对所有点进行评分的情况下，组合分数通常也会增强分数的局部子空间属性，因为集合方法能够减少代表性偏差[170]（cf 第6章第6.4.3 节。

对第一类方法的细粒度分析通常在计算上很昂贵。 由于该分析具有计算密集和细粒度的特性，因此通常更难以充分探索使用多个子空间进行分析。 这有时会对准确性产生不利影响。 第二类方法具有明显的计算优势。 可以利用该计算效率来探索更大数量的子空间并提供更稳健的结果。 属于第二类的许多方法，例如特征装袋，旋转装袋，子空间直方图和隔离林，是用于子空间异常值检测的更成功和准确的方法。

在子空间分析的背景下，基于集合的分析的优势非常显着[31]。 由于来自不同子空间的离群值得分可能非常不同，因此通常很难完全信任来自单个子空间的得分，并且得分的组合是至关重要的。 本章将探讨利用组合多个子空间的优点的几种方法。

5.2.1 异常检测的遗传算法

子空间离群检测的第一种方法[4]是遗传算法。 通过在低维空间中找到具有异常低密度的数据的局部区域来识别子空间异常值。 采用遗传算法来发现这样的局部子空间区域。 然后，异常值由其在这些地区的成员资格来定义。

5.2.1.1 去除异常的低维投影

为了识别异常的低维投影，重要的是提供异常低维投影的适当统计定义。 异常的低维投影是数据密度异常低于平均值的投影。 在这种情况下，第2 章介绍的极值分析方法很有用。

使用基于网格的方法来识别很少填充的本地子空间区域。 第一步是创建具有数据离散化的网格区域。 每个属性都是

singola@gmx.com

分为φ 范围。 这些范围是在等深度的基础上创建的。 因此，每个范围包含记录的分数f = 1/φ 。 使用等深度范围而不是等宽度范围的原因是数据的不同位置可能具有不同的密度。 因此，这种方法部分地调整了初始阶段期间数据密度的局部变化。 这些范围形成用于定义稀疏子空间区域的局部单位。

考虑通过从k个不同尺寸中选择网格范围而创建的k维立方体。 如果属性在统计上是独立的，则该k维区域中的记录的预期分数是fk。 当然，现实世界的数据通常远非统计独立，因此立方体中点的实际分布将与该预期值明显不同。 许多局部区域可能包含非常少的数据点，并且随着k值的增加，它们中的大多数将是空的。 在这些异常稀疏区域非空的情况下，它们内部的数据点可能是异常值。

假设数据库中的总点数由N 表示。 在上述独立性假设下，k维立方体中任何点的存在与否都是具有概率fk的伯努利随机变量。 然后，根据伯努利随机变量的性质，我们知道k维立方中点的预期分数和标准差分别由N fk 和N fk （1 fk ）给出。 此外，如果数据点数N.中心极限定理很大，可用于通过正态分布来近似立方体中的点数。 这样的假设可以帮助创建有意义的异常（稀疏性）度量 √ · · · − 立方体。 设n（D）为k维立方体中的点数D.稀疏系数

数据集D的S（D）可以如下计算：

$$S(D) = \frac{n(D) - N \cdot f^k}{\sqrt{N \cdot f^k \cdot (1 - f^k)}} \tag{5.1}$$

只有负的稀疏系数表示密度低于预期的局部投影区域。 由于 假设n（）为正态分布，因此正态分布表可用于量化其偏差的概率的概率水平。 尽管独立性假设从未真实存在，但它为估计点特异性异常提供了良好的实用启发式算法。

5.2.1.2 用于子空间搜索的Deftning遗传算子

由于指数计算复杂性，对所有子空间的详尽搜索是不切实际的。 因此，需要一种修剪大多数子空间的选择性搜索方法。 该问题的本质是在满足稀疏性条件的基于网格的子空间上没有向上或向下闭合的属性[1] 。 这些属性通常用于其他问题，如频繁模式挖掘[36]。 然而，不像

频繁的模式挖掘，其中一个人正在寻找高频率的模式，发现人口稀疏的维度子集的问题具有在大海捞针中找到针的概率。 此外，即使特定区域可以很好地填充在某些维度子集上，但是当这些维度组合时，它们可能非常稀疏地填充。 例如，在给定的数据集中，可能有大量的个体在20岁时聚集，并且适度数量的个体具有高度严重的阿尔茨海默病。 然而，非常罕见的人会满足

[1]向上闭合的模式是模式的所有超集也是有效模式的模式。 向下闭合的图案集是其中图案的所有子集也是该集合的成员的图案。

singola@gmx.com

这两个标准，因为疾病不会影响年轻人。 从离群检测的角度来看，一个20岁的早发性阿尔茨海默病是一个非常有趣的记录。 然而，图案的趣味性甚至没有通过其低维投影暗示。 因此，最好的投影通常是由未知的维度组合创建的，其低维投影可能包含非常少的用于指导子空间探索的提示。 一种解决方案是改变措施以强制更好的关闭或修剪属性；但是，强制选择由算法考虑因素驱动的度量通常会导致结果不佳。 通常，不可能预测在异常值得分上组合两组维度的效果。 因此，一个自然的选择是开发可以识别这种隐藏的维度组合的搜索方法。 一个重要的观察是，人们可以将发现稀疏子空间的问题视为最小化识别的子空间中的点数的计数的优化问题。 但是，由于子空间的数量随着数据维数呈指数增长，因此[4]使用遗传算法，也称为进化搜索方法。 这种优化方法在非结构化设置中特别有用，其中很少有硬性规则来指导搜索过程。

遗传算法，也称为进化算法[273]，是为了解决结构不良的优化问题而模拟有机进化过程的方法。 在进化方法中，优化问题的每个解决方案都可以表示为进化系统中的个体。 该“个体”的完美度量等于相应解的目标函数值。 在生物进化中，个体必须与作为优化问题的替代解决方案的其他个体竞争。 因此，人们总是在任何给定时间使用多种（即人口）解决方案，而不是单一解决方案。 此外，可以通过重组旧解决方案的特性来创建新的解决方案，这是生物再生产过程的类比。 因此，为了模拟重组和突变过程以完成模拟，定义了适当的操作。 突变可以被视为探索可能改进的密切相关解决方案的一种方式，就像在爬山方法中那样。

显然，为了模拟这种生物过程，我们需要对优化问题的解决方案进行某种简洁的表达。 该表示使得能够用于模拟重组和变异的算法过程的具体算法。 每个可行的解决方案都表示为一个字符串，可以将其视为解决方案的染色体表示。 将可行解决方案转换为字符串的过程称为其编码。 进化算法的有效性通常在很大程度上取决于编码的选择，因为它隐含地定义了用于搜索空间探索的所有操作。 通过固有函数评估字符串的完整性。 这相当于对优化问题的目标函数的评估。 因此，具有更好目标函数值的解决方案可以被视为生物环境中的个体的类比。 当进化算法模拟生物进化过程时，它通常会导致所有解决方案（人口）的平均目标函数得到改善，因为生物进化过程会随着时间的推移提高效率。 此外，由于对个体的永久（选择）偏见，解决方案群体的多样性丧失了。 这种多样性的丧失类似于收敛在其他类型的迭代优化算法中的工作方式。 德容[它通常会导致手头所有解决方案（人口）的平均目标函数得到改善，因为生物进化过程随着时间的推移会提高效率。 此外，由于对个体的永久（选择）偏见，解决方案群体的多样性丧失了。 这种多样性的丧失类似于收敛在其他类型的迭代优化算法中的工作方式。 德容[它通常会导致手头所有解决方案（人口）的平均目标函数得到改善，因为生物进化过程随着时间的推移会提高效率。 此外，由于对个体的永久（选择）偏见，解决方案群体的多样性丧失了。 这种多样性的丧失类似于收敛在其他类型的迭代优化算法中的工作方式。 德容[[163]将字符串中特定位置的收敛定义为95％的人口对该位置具有相同值的阶段。 当字符串表示中的所有位置都收敛时，据说人口已经收敛。

进化算法将子空间投影视为op-的可能解决方案

singola@gmx.com

定理问题。 这种投影可以很容易地表示为字符串。 由于数据被离散化为网格结构，我们可以假设任何维度中各种网格间隔的标识符范围从1到φ。 考虑一个d维数据点
d个不同尺寸的网格间隔用（m_1，... md）表示。 每个mi 的值可以取1到φ中的任何值，或者它可以取值*，表示“不关心”值。 因此，存在总共φ的1可能值英里。 考虑
一个4维数据集，其中φ = 10.然后，问题解决方案的一个可能的例子由字符串* 3 * 9 给出。 在这种情况下，第二维和第四维的范围被识别，而第一和第三维的范围被保留为“不关心”。 进化算法使用投影k 的维度作为输入参数。 因此，对于d维数据集，长度为d 的字符串将包含k个特定位置和（dk）“不关心”位置。 这表示k的字符串编码- 维子空间。 可以使用前面讨论的稀疏系数计算相应解的精度。 进化搜索技术从一群p 随机解开始，并迭代地使用选择，交叉和变异的过程，以便在可能的投影空间上执行爬山，解决方案重组和随机搜索的组合。 根据De Jong收敛标准[163]，该过程一直持续到人口收敛到全局最优。 在算法的每个阶段，m最佳投影解决方案（大多数负稀疏系数）以运行方式进行跟踪。 在算法结束时，这些解决方案被报告为数据中的最佳预测。 以下运算符用于选择，交叉和变异：

- **选择：**通过按等级对它们进行排序并将它们偏向群体来复制解决方案的副本，以支持更高排名的解决方案。 这被称为等级选择。

- **交叉：**交叉技术是算法成功的关键，因为它隐含地定义了子空间探索过程。 一种解决方案是使用均匀的两点交叉以产生重组儿童字符串。 两点交叉机制的工作原理是随机确定一个点中的一个点，称为交叉点，并将这些段交换到该点的右侧。 然而，这种盲重组过程可能经常产生差的解决方案。 因此，定义了优化的交叉机制。 在这种情况下，保证两个子解都对应于k- 作为父母的三维投影，孩子们通常具有高度的价值。 这是通过检查重组的不同可能性的子集并在其中选择最佳实现来实现的。 基本思路是从（至多）空间贪婪地选择k 维度
两个父母包括2 k个不同的维度。 [4]中提供了这种优化交叉过程的详细描述。

- **突变：**在这种情况下，字符串中的随机位置以预定义的突变概率进行。 必须注意确保在浮动过程之后投影的维数不会改变。

在终止时，算法之后是后处理阶段。 在后处理阶段，算法将包含异常投影的所有数据点报告为异常值。 该方法还提供了相关预测，这些预测提供了数据点异常行为的因果关系。 因此，这种方法具有高度的可解释性。

singola@gmx.com

5.2.2 寻找基于距离的外围子空间

在基本子空间异常值检测框架[4]的初步提议之后，沿着这条线的最早方法之一是HOS-Miner方法。 与HOS-矿工相关的更广泛的论述的几个二FF erent方面在[讨论605，606，607]。 在[605]中提出了关于HOS-Miner方法的第一次讨论。 根据这项工作，给定数据点X 的外围子空间的定义如下：

定义5.2.1 对于给定的数据点X，确定子空间集合，使得该子空间中其k个最近邻距离的总和至少为δ。

此方法不会使尺寸数量的距离标准化。 因此，随着维数的增加，子空间变得越来越可能。 该定义还展示了闭包属性，其中非外围子空间的任何子空间也不是外围的。 同样，边远子空间的每个超集也都是边远的。 显然，只有极小的子空间才是有趣的。 [605]中的方法使用这些闭包属性来修剪不相关或不感兴趣的子空间。 尽管上述定义具有理想的闭合特性，但使用固定阈值δ跨越不同维度的子空间似乎是不合理的。 选择基于算法方便性的定义通常会导致较差的结果。 正如年轻阿尔茨海默病患者的早期例子所示，真正的异常值通常隐藏在数据的子空间中，这不能从其低维或高维预测中推断出来。

使用X-Tree来执行索引以便在不同的子空间中有效地执行k-最近邻居查询。 为了进一步提高学习过程的效率，[605]中的工作使用数据的随机样本，以便在开始子空间探索过程之前了解子空间。 这是通过估算外围子空间的总储蓄系数（TSF）的数量来实现的。 这些用于规范特定查询点的搜索过程，并以有序的方式修剪不同的子空间。 此外，随着搜索的进行，不同子空间的TSF值会被动态更新。 它已在[605]中显示]这种方法可以用来有效地确定特定数据点的边远子空间。 使用二FF erent种的修剪性能和遗传算法的用于连接外围nding子空间许多方法是在[呈现606，607]。

5.2.3 特征Bagging：子空间采样视角

用于组合来自多个子空间的异常值的最简单方法是使用特征装袋[344]，这是一种集合方法。 整体的每个基本组件使用以下步骤：

- 从|中随机选择一个整数r d /2∫至（d - 1）。
- 在迭代t 中从基础数据集中随机选择r个特征（无替换），以便在第t次迭代中创建r维数据集Dt 。
- 将异常检测算法Ot 应用于数据集Dt ，以计算每个数据点的分数。

原则上，可以在每次迭代中使用不同的离群值检测算法，前提是在过程之后将得分归一化为Z值。 归一化也是必要的，以解释不同子空间样本包含不同数量的特征的事实。 但是，[344]中的工作使用LOF算法进行所有迭代。 自从

singola@gmx.com

LOF算法返回固有的归一化分数，这种归一化不是必需的。 在该过程结束时，来自不同算法的异常值得分以两种可能的方式之一组合：

- 广度 - 第一种方法：在这种方法中，算法的排名用于组合目的。 所有不同执行中排名靠前的异常值排在第一位，其次是排名第二的异常值（重复删除），依此类推。 由于特定等级内的异常值之间的打破关系，可能存在微小的变化。

- 累积和方法：总结了不同算法执行的异常值分数。 在此基础上报告排名最高的异常值。 人们还可以将此过程视为等效于集合方法中的平均组合函数（参见第6章）。

在[344]中通过实验证明，这些方法能够改善无关属性的影响。 在这种情况下，由于额外的噪声，全维算法无法将真实异常值与正常数据区分开来。

在第一眼看来，它似乎是随机子空间采样[344]并不试图优化相关子空间的发现。 尽管如此，它确实具有相对高效的样本子空间，因此可以对大量子空间进行采样以提高鲁棒性。即使每个探测器选择一个全局子空间，给定点的基于集合的组合分数也能够从局部优化的子空间中获得明显的好处。 这是因为不同的点可以在不同的子空间样本中获得有利的分数，并且整体组合总是能够识别在足够数量的子空间中有利的所有点。 这种现象也可以通过集合方法如何减少代表性偏差的概念来正式解释[170]（参见章节）第6章6.4.3.1 ）。 换句话说，集合方法提供了将全局子空间探索转换为局部子空间选择的隐式路径，因此本质上比其各个组件更强大。 第6.4.3.1节提供了以特征包装为中心的集合视角。

多个子空间采样产生的鲁棒性显然是非常理想的质量，只要最后的组合函数能够识别给定数据点的不同子空间样本的不同行为。 从某种意义上说，这种方法隐含地认识到检测相关和稀有子空间的困难，因此尽可能多地采样子空间以揭示罕见的行为。 从概念的角度来看，这种方法类似于利用许多弱学习者的力量在分类问题中创建一个强大的学习者。 已经证明该方法在[344]中显示了对于许多实际数据集的全维方法的一致性能改进]。 该方法也可以称为特征装袋方法或随机子空间集合方法。 尽管原始工作[344]使用LOF作为基础检测器，但平均k-最近邻检测器也已被证明有效[32]。

5.2.4 预计的聚类集合

预测的聚类方法将聚类定义为一组点以及这些点聚集良好的维数。 如第4章所述，聚类和异常值检测是互补问题。 因此，研究预测或子空间聚类方法是否也可用于异常值检测是很自然的。 虽然

singola@gmx.com

群集的相关子空间并不总是与异常值检测相关，两者之间仍然存在弱关系。 通过使用集合，可以加强使用此方法发现的异常值的类型。

如OutRank工作[406]所示，可以使用投影聚类算法[5]的集合进行子空间异常值检测。有鉴于此，[406]已经强调，使用多个投影聚类是必不可少的，因为使用单个投影聚类算法会产生非常差的结果。 OutRank的基本思想是重复使用以下过程：

- 在数据集上使用PROCLUS [5]等随机投影聚类方法来创建一组投影聚类。
- 根据每个点与其所属的集群的相似性来量化每个点的异常值。 相关分数的示例包括尺寸，维度，到群集质心的（投影）距离或这些因素的组合。 正确选择度量对所使用的特定聚类算法很敏感。

重复应用该过程，并对得分进行平均以产生最终结果。 在第一步中使用一个充分随机的聚类方法对于以这种以集合为中心的方法获得良好结果至关重要。

人们可以使用几种变体进行评分。 对于像PROCLUS这样的基于距离的算法，使用相同的距离度量来量化用于聚类的离群值得分是有意义的。 例如，在PROCLUS的情况下，可以使用距离其最近的聚类质心的数据点的曼哈顿节段距离。 通过首先计算点到其相关子空间中的簇的质心的曼哈顿距离，然后除以该子空间中的维数，来估计曼哈顿分段距离。 但是，该度量忽略了簇的点数和维数；此外，它不适用于具有重叠簇的基于模式的方法。 一种自然的方法是个人加权措施。 对于一点，计算其簇中的点数与最大簇大小的比例，并且还计算其簇子空间中的维数与最大簇子空间维度的比例。 一个简单的异常值得分是将这两个分数添加到出现该点的所有聚类上，然后除以数据中聚类的总数。 包含在许多大型和高维子空间簇中的点不太可能是异常值。 因此，分数较小的点被视为异常值。 本节将讨论二进制和分类数据的类似方法 一个简单的异常值得分是将这两个分数添加到出现该点的所有聚类上，然后除以数据中聚类的总数。 包含在许多大型和高维子空间簇中的点不太可能是异常值。 因此，分数较小的点被视为异常值。 本节将讨论二进制和分类数据的类似方法 一个简单的异常值得分是将这两个分数添加到出现该点的所有聚类上，然后除以数据中聚类的总数。 包含在许多大型和高维子空间簇中的点不太可能是异常值。 因此，分数较小的点被视为异常值。 本节将讨论二进制和分类数据的类似方法第8章8.5.1 。 在[406]中提出了另外两种称为聚类覆盖和子空间相似性的措施。 [406]中的工作尝试了许多不同的聚类算法，发现使用PROCLUS的多个随机运行产生了最好的结果。 这种变化称为Multiple-Proclus。 群集覆盖和子空间相似性度量用于实现这些结果，尽管通过个体加权度量也获得了合理的结果。

理解这种基于聚类的方法的关键点在于，发现的异常值类型对底层聚类很敏感，尽管以集合为中心的方法对于成功至关重要。 因此，局部敏感的聚类集合将使异常值局部敏感；子空间敏感的聚类集合将使离群子对子空间敏感，并且相关敏感的聚类集合将使异常值相关敏感。

5.2.5 线性时间内的子空间直方图

在[476]中提供了具有散列的子空间直方图的线性时间实现，并且被称为RS-Hash。 基本思想是在大小为s的数据样本上重复构建基于网格的直方图，并以集合为中心的方法组合分数。 每个直方图都是在随机选择的数据子空间上构建的。 子网空间的维数和网格区域的大小是其集合组件的特定。 在集合分量的测试阶段，基于其网格区域中的点数（来自训练样本）的对数，对数据中的所有N 个点进行评分。 用多个尺寸为s的样本重复该方法。 点特定分数在不同的集合分量上取平均值，以创建最终结果，这是非常稳健的。 在网格区域的维数和尺寸的变化被控制与上午整数维度参数- [R 和分数网格尺寸参数˚F （0，1），该随机变化过二FF erent合奏部件。 我们稍后将描述随机选择这些参数的过程。 目前，我们假设（为简单起见）这些参数的值是固定的。

尺寸为s N 的样本S 可以被视为单个整体分量的训练数据，并且通过在该训练样本上构建子空间直方图，在该分量中对每个N 个点进行评分。 首先，一组V 的ř 尺寸是随机从采样的 d 尺寸，并且所有的得分是在内置在该直方图进行ř维
子空间。 最小值minj 和最大值 MAXJ 所述的 Ĵ个尺寸从该样品来确定。 设xij 表示第i个点的第j维。 所有s · r
值xij 训练样本，使得在Ĵ ∈ V ， 被归一化， 如下所示：

$$x^j_{ij} \Leftarrow \frac{x_{ij} - min_j}{max_j - min_j} \tag{5.2}$$

甚至可以使用 $x^j_{ij} \Leftarrow xij / (maxj - minj)$ 来实现简单性。 在归一化时，我们还创建了以下r维三维离散化表示
训练点，其中对于每个所述的ř 在尺寸V ，我们使用宽度的网格尺寸˚F （0，1）。 此外，为了在整体组件之间引入分集，网格分区点的放置在整体组件之间变化。 在给定的整体组件中，网格分区点不是0，而是维度的值αj
j。 这可以通过将点i 和维j 的离散化标识符设置为（x^j_{ij} + αj）/ f来实现。 αj 的值在从（0，f ）ſ随机选择的值的前面的集合分量中固定。 这种r维三维离散化表示提供了
该点的r维边界框的标识。 哈希表维护训练样本中遇到的每个边界框的计数。 对于每个s训练点，通过对该r维离散表示进行散列来增加其边界框的计数，这需要恒定的时间。 在测试阶段，使用上述过程再次构建N 个点中的每个点的离散化表示，并且从训练样本上构建的哈希表中检索其计数。 让ni s 表示第i点的计数。 对于训练样本S中包含的点数，离群值得分第i点是$\log_2$（ni），而对于未包括在训练样本中的点
S，得分为$\log_2$（ni + 1）。 在多个集合组件上重复该过程 ≤
（通常为100），并返回每个点的平均分数。 低分表示异常值。

值得注意的是，在集合分量的测试阶段中使用的minj 和maxj 的值与从训练样本估计的值相同。 因此，训练阶段仅需要O（s）时间，并且s 的值通常是诸如1000的小常数。 每个集合分量的测试阶段需要O（N ）时间，并且整体

singola@gmx.com

算法需要O（N + s）时间。 常数因素也往往很小，而且方法非常快。

我们现在描述在每个√en中随机选择 f ， r和 αj的 过程。 _ _
集合组件。 从（1/ s，1 - 1/ s）随机均匀地选择f 的值。
随后，r 的值 被随机均匀地设置为1 + 0之间的整数。 5·
[logmax{2, 1/ F}（小号）]和logmax{2, 1/ F}（小号）。 每个αj 的值在运行时均匀选择
从（0，f ）开始。 网格大小（带f ），维度（带r）和位置的这种变化
网格区域（带有αj）提供了额外的多样性，这有助于整体的集合结果。 [476]中的工作还显示了该方法如何扩展到数据流。 该变体称为RS-Stream。 流式变体在散列表设计和维护中需要更高的复杂性，尽管整体方法非常相似。

5.2.6 隔离森林

[367]中的工作提出了一种称为隔离森林的模型，它与另一种被称为随机森林的集合技术具有一些直观的相似性。 随机森林是用于分类的最成功的模型之一，并且已知在各种问题领域中表现优于大多数分类[195]。 然而，构建隔离林的无监督方式是非常不同的，特别是在数据点如何评分方面。 如5.2.6.3节所述，隔离林也是极端随机化聚类林（ERC-Forest）用于聚类的特例[401]。

隔离林是一组隔离树的集合组合。 在隔离树中，数据在随机选择的属性中随机选择的分区点以轴平行切割进行递归分区，以便将实例隔离到具有越来越少实例的节点，直到这些点被隔离到包含一个实例的单个节点中。 在这种情况下，包含异常值的树枝明显不那么深，因为这些数据点位于稀疏区域。 因此，叶子到根的距离用作离群值得分。 通过平均隔离林的不同树中的数据点的路径长度来执行最终组合步骤。

隔离林与子空间异常值检测密切相关。 不同的分支对应于数据的不同本地子空间区域，具体取决于如何选择属性以进行拆分。 较小的路径对应于已经隔离异常值的子空间的较低维度[2] 。 孤立点所需的维数越小，该点的异常值就越强
成为。 换句话说，隔离森林在隐含的假设下工作，即它更有可能隔离由随机分裂创建的较低维度的子空间中的异常值。 例如，在我们的阿尔茨海默氏症患者前面的例子中，分割等的短序列年龄30，阿尔茨海默= 1很可能是一种罕见的个人与早发性阿尔茨海默病隔离。 ≤

隔离林的训练阶段构建多个隔离树，这些隔离树是决策树的无监督等价物。 每棵树都是二进制的，并且对于包含N 个点的数据集，最多具有N个叶子节点。 这是因为默认情况下每个叶节点只包含一个数据点，但通过参数化方法可以提前终止

[2]尽管可以沿相同的尺寸重复切割，但这在高维数据集中变得不太可能。 通常，路径长度与用于隔离的子空间的维度高度相关。 对于包含2^8 = 256个点（这是推荐的子样本大小）的数据集，树的平均深度将为8，但是异常值通常可以在少于三个或四个分割（维度）中被隔离。

singola@gmx.com

高度参数。 为了从包含N 个点的数据集构造隔离树，该方法创建包含所有点的根节点。 这是隔离树的初始状态。 将节点的候选列表（用于进一步拆分）初始化为包含根节点的单个列表。 然后，T 重复以下步骤以创建 C

隔离树T直到候选列表C为空：

1. 从C中随机选择一个节点R并从C中删除。
2. 选择随机属性i 并将R中的数据按照沿该属性选择的随机值a 分成两组R_1和R_2 。 因此，R_1中的所有数据点满足$x_i < a$，并且R_2中的所有数据点满足$x_i \geq a$。 在节点R中的数据点中的第i个属性的最小值和最大值之间随机均匀地选择随机值a 。 节点R_1和R_2是T中的R的子节点。
3. （对于每个执行步骤 我∈{1，2}）：如果日包含多于一个点然后将其添加到C.否则，指定该节点作为隔离树叶。

此过程将导致创建通常不平衡的二叉树。 异常节点往往比非异常节点更快地隔离。 例如，在所有维度上距离剩余数据非常远的点可能在第一次迭代中被分离为孤立的叶子。 因此，从根到叶子的路径长度被用作离群值得分。 请注意，从根到叶的路径定义了不同维度的子空间。 异常值通常可以在比正常点低得多的子空间中被隔离。 因此，从根到节点的边数等于其异常值。 较小的分数对应于异常值。 这种固有的随机方法重复多次，并对得分进行平均以产生最终结果。 类似集合的方法特别有效，它通常提供高质量的结果。 隔离树的基本版本在生长到全高时是无参数的。 在诸如异常值检测之类的无监督问题中，这种特性始终是一个显着的优势。 该方法的平均情况计算复杂度为对于每个隔离树，θ（N log（N ））和空间复杂度为O（N ）。

人们总是可以提高计算效率，并且通常可以通过子采样提高准确性。 子采样方法引发了进一步的多样性，并获得了异常集合中固有的一些优势[31]。 在这种情况下，在训练阶段使用子样本构建隔离树，尽管在测试阶段针对子样本对所有点进行评分。 训练阶段将树结构与拆分条件（例如$x_i < a$ 和$x_i \geq a$）一起返回）在每个节点。 通过使用在训练阶段期间计算的分割条件，在测试阶段对样本外点进行评分。 与决策树的测试阶段非常相似，通过使用分裂条件遍历从根到叶的适当路径来识别样本外点的适当叶节点，这是简单的单变量不等式。

子样本的使用导致更好的计算效率和更好的多样性。 在[367]中陈述了256的子样本大小在实践中很好地工作，尽管这个值可能随着手头的数据集而有所不同。 请注意，无论数据集大小如何，这都会导致构建树的计算和内存需求不断增加。 如果使用整个数据集，则测试阶段要求每个数据点的θ（log（N ））的平均情况复杂度。 另一方面，如果使用大小为256的子样本，则测试阶段需要每个点的恒定时间。 因此，如果使用具有恒定数量的样本和恒定数量的试验的子采样，则训练阶段的运行时间是恒定的，O（N ）用于测试阶段，空间复杂度也是恒定的。

singola@gmx.com

隔离林是一种有效的方法，考虑到大多数子空间方法是计算密集型的事实，这是值得注意的。

通过平均隔离林的不同树中的数据点的路径长度来执行最终组合步骤。 第6章第6.4.5节提供了从集合为中心的观点对这种方法的详细讨论。 Python库scikit-learn [630]和R库SourceForge [631]提供了这种方法的实际实现。

5.2.6.1 子空间选择的进一步增强

通过使用峰度测量来额外预选特征来增强该方法。 一组特征值$x_1 \ldots x_N$ 的峰度通过首先将它们标准化为$z_1 \ldots z_N$ ，零均值和单位标准偏差来计算：

$$z = \frac{x_i - \mu}{\sigma} \tag{5.3}$$

这里，μ 是平均值，σ 是$x_1 \ldots x_N$ 的标准偏差。 然后，峰值计算如下：

$$K(z_1 \ldots z_N) = \frac{\sum_{i=1}^{N} z_i^4}{N} \tag{5.4}$$

非常不均匀的特征将显示高水平的峰度。 因此，可以将库仑计算视为异常检测的特征选择度量。 [367]中的工作基于其单变量Kuritosis值的排名预先选择属性的子集，然后在（全局）丢弃这些特征之后构建随机森林。 请注意，这会导致全局子空间选择；尽管如此，随机分割方法仍然能够探索不同的局部子空间，尽管是随机的（如特征包装）。 Kurtosis措施的概括，称为多维Kurtosis，将在第1章第1.3.1 节中讨论。 该度量使用等式联合评估特征的子集5.4 关于该子空间中的点的Mahalanobis距离，而不是使用单变量峰值值对特征进行排序。 这种测量通常被认为比单变量峰值效应更有效，但它在计算上是昂贵的，因为它需要以结构化方式与特征子集探索相结合。 尽管已经在隔离林框架中使用了广义的Kurtosis度量，但它有可能在计算复杂性不那么重要的环境中应用。

5.2.6.2 提前终止

进一步的增强是树没有生长到全高。 一旦节点包含重复实例或超过某个阈值高度，节点的增长就会停止。 在[367] 的实验中，该阈值高度设定为10。 为了估计这些节点中的点的路径长度，需要分配额外的信用以考虑这些节点中的点尚未实现到全高的事实。 对于包含r个实例的节点，其附加信用$c(r)$被定义为具有r 个点[442]的二叉搜索树中的预期路径长度：

$$c(r) = \ln(r-1) - \frac{2(r-1)}{r} + 0.5772 \tag{5.5}$$

singola@gmx.com

请注意，此信用将从根添加到该节点的路径长度，以计算最终的异常值。 提前终止是一种以效率为中心的增强，如果需要，可以选择将树种长到全长。

5.2.6.3 与聚类集合和直方图的关系

可以将隔离林视为一种群集集合，如第5.2.4节中所述。 隔离树根据数据创建分层投影集群，其中集群由其边界框定义。 簇（节点）的边界框由从根到该节点的轴平行分裂序列定义。 然而，与大多数预测的聚类方法相比，隔离树极其随机化，因为它专注于以集合为中心的性能。 隔离林是一种基于决策树的聚类方法。 有趣的是，基本隔离林可以显示为早期聚类集合方法的变体，称为极随机聚类森林（ERC-Forests）[401]]。 主要的不同之处在于，ERC-Forest在每个节点使用多个tri-als来实现对类标签的少量监督；但是，通过将试验次数设置为1并将树长到全长，可以获得无监督的隔离树作为特殊情况。 由于ERC-Forests和隔离林中使用的空间划分（而不是点分区）方法，这些方法也与基于直方图和密度的方法具有一些直观的相似性。 隔离树创建分层和随机网格区域，每个分割的预期体积减少2倍。 因此，隔离树中的路径长度是包含单个数据点的最大网格区域的（分数）体积的负对数的粗略替代。 这类似于传统直方图中使用的对数似然密度的概念。 与传统的直方图不同，隔离林不是通过网格宽度参数化的，因此在处理不同密度的数据分布时更灵活。 此外，后者中直方图的灵活形状自然地定义了局部子空间区域。

利用预测聚类方法对异常值进行评分的措施[406]也与隔离森林有一些直观的相似之处。 [406]中的一个度量使用群集的子空间维度和群集中的点数之和作为异常值得分。 请注意，群集的子空间维度是隔离树中路径长度的粗略代理。 类似地，隔离树的一些变体，称为半空间树[532]，使用固定高度的树木。 在这些情况下，点的相关节点中的点数用于定义其异常值。 这类似于基于聚类的离群值检测方法，其中最近聚类中的点数通常用作离群值得分的重要组成部分。

5.2.7 选择高对比度子空间

5.2.3 节中讨论的特征装袋方法[344] 随机抽样子空间。 如果许多维度不相关，则每个子空间样本中可能包含至少一些维度。 同时，由于丢弃了许多维度，信息会丢失。 这些影响不利于方法的准确性。 因此，很自然地询问是否可以执行选择较少数量的高对比度子空间的预处理。 该方法也称为HiCS，因为它选择高对比度子空间。

在[308]中提出的工作中，仅在这些高对比度子空间中发现异常值，并且组合相应的分数。 因此，这种方法解耦了子

singola@gmx.com

空间搜索作为一种广义的预处理方法，来自各个数据点的异常值排序。 [308]中讨论的方法非常有趣，因为它的预处理方法可以找到相关的子空间，以减少不相关的子空间探索。尽管使用基于聚合的统计数据获得高对比度子空间，但这些统计数据仅用作提示，以便识别多个子空间以获得更强的鲁棒性。 这里的假设是，在存在显着的非均匀性和对比度的子空间中，罕见模式在统计上更可能发生。 最终的异常值得分结合了不同子空间的结果，以确保至少选择几个相关的子空间。 [308。]的工作见解]是将判别子空间选择与特征装袋的得分聚合相结合，以确定相关的离群值。 因此，特征装袋的唯一不同之处在于如何选择子空间; 算法在其他方面是相同的。 在[308]中已经表明，这种方法比特征装袋方法表现更好。因此，HiCS方法的概述如下：

1. 第一步是使用Apriori-like [37]探索选择判别子空间，本节末尾对此进行了描述。 这些是高对比度子空间。 此外，子空间也被修剪以解决它们之间的冗余。
   - 这种探索的一个重要部分是能够在勘探过程中评估候选子空间的质量。 这是通过量化子空间的对比度来实现的。
2. 一旦确定了子空间，就会使用与特征装袋方法完全相似的方法。 在将数据投影到这些子空间之后执行LOF算法，并且如5.2.3节中所讨论的那样组合得分。

因此，下面我们的描述将仅关注类Apriori探索的第一步和对比的相应量化。 我们将偏离呈现的自然顺序并首先描述对比度计算，因为它与正确理解类似Apriori的子空间探索过程密切相关。

考虑维度p的子空间，其中维度被索引为{1 ... p}（不失一般性）。 对于不相关数据的情况，属性值x_1 的条件概率$P(x_1|x_2 \dots x_p)$ 与其无条件概率$P(x_1)$ 相同。
由于数据分布的不均匀性，高对比度子空间可能违反此假设。 在我们早期的年轻阿尔茨海默病患者的例子中，这与青年和疾病相结合的意外罕见性相对应。 换句话说，P （阿尔茨海默氏症= 1）可能与P （阿尔茨海默氏症= 1 年龄 30）非常不同。 这个想法是具有这种意外非均匀性的子空间更可能包含异常值，尽管它仅被视为预先选择多个子空间之一的弱提示。 [308]中的方法使用类似Apriori的方法生成候选子空间[37]]在本节后面描述。 对于维度p的每个候选子空间（可能是
在Apriori式探索期间变化，它从数据中重复绘制成对的“样本”，以估计P （xi）和P （xi $x_1 \dots x_{i-1}, x_{i+1} \dots x_p$）和测试它们是否不同。 “样本”由（i）从1 ... p中选择特定属性i 来定义
用于测试，以及（ii）在p维空间中构造随机矩形区域用于测试。 由于用于测试的矩形区域的构造，每个xi 指的是
到的值（例如，一维范围年龄∈（10，在20））我个维度。 的值 P （XI）和 P （XI|$X_1 \dots X_{I-1}, X_{I1} \dots X_P$）计算在该随机矩形区域。 在绘制了M 对P （xi）和 P （xi|$x_1 \dots x_{i-1}, x_{i+1} \dots x_p$）的样本 之后 ，它是

singola@gmx.com

确定是否违反了假设检验的独立性假设。 基于学生t分布的各种测试可用于测量子空间与独立性基本假设的偏差。 这提供了子空间非均匀性的度量，因此提供了一种根据子空间包含异常值的倾向来测量子空间质量的方法。

自下而上的Apriori式[37]方法用于识别相关的预测。 在这种自下而上的方法中，子空间不断扩展到更高的维度，以进行非均匀性测试。 与Apriori一样，只有具有足够对比度的子空间才能作为潜在候选者进行非均匀性测试。 非均匀性测试如下进行。 对于维度p的每个候选子空间，以p 维生成随机矩形区域。 每个维度的随机范围的宽度是
选择使得1维范围包含$N\alpha^{(1/p)}$点，其中$\alpha < 1$。 因此，预期整个 p维区域包含 $N\alpha$ 个点。 第i个维度是
用于假设检验，其中i 的值从1 ... p 随机选择。 可以将索引i 视为测试维度。 让集合
沿着剩余（p -1）维度的范围用Si表示 。 位于尺寸i 范围的上限和下限内的Si 中的点的分数提供了P （$x_i|x_1 \ldots x_{i-1}, x_{i+1} \ldots x_d$）的估计 。 使用假设检验计算该值与P （$x_i$）的无条件值的统计标准化偏差
提供该子空间的偏差估计。 在不同的随机切片和测试尺寸上重复该过程多次; 然后，对不同测试的偏差值进行平均。 具有大偏差的子空间被识别为高对比度子空间。 在类似Apriori的阶段结束时，应用额外的修剪步骤来移除冗余子空间。 维数的子空间p 被去除，如果维数的子空间另一个（p + 1）的情况下（所报告的子空间中的）具有较高的对比度。

该方法将子空间识别与异常值检测分离，因此所有相关子空间都被预先识别为预处理步骤。 在识别子空间之后，在每个这样的子空间中使用LOF算法对点进行评分。 请注意，此步骤与功能包装非常相似，只是我们将自己限制在更精心选择的子空间。 然后，计算各个子空间中每个点的得分并进行平均，以提供每个数据点的统一得分。 原则上，可以使用诸如最大化的其他组合功能。 因此，可以适应特征装袋中使用的任何组合方法。 [308]中提供了用于选择相关子空间的算法的更多细节。

HiCS技术以直观的观点着称，即相关子空间的统计选择比选择随机子空间更有效。 主要挑战在于发现高对比度子空间，因为将Apriori类算法与基于样本的假设检验结合使用是计算密集型的。 存在许多用于发现高对比度子空间的直接替代方案，这可能值得探索。 例如，可以使用第1 章第1.3.1节中讨论的多维峰度测量法为了测试子空间与高维异常值检测的相关性。 该度量计算简单，并且由于使用了马哈拉诺比斯距离，因此也考虑了维度之间的相互作用。

5.2.8 子空间投影的局部选择

[402]中的工作使用相关子空间投影的局部统计选择来识别异常值。 换句话说，子空间投影的选择被优化为特定的数据点，因此给定数据点的位置在

singola@gmx.com

选择过程。 对于每个数据点X，识别一组子空间，这些子空间被考虑
从异常值检测的角度来看高对比度子空间。 然而，这个探索过程使用高对比度行为作为统计提示，以便探索多个子空间的鲁棒性，因为单个子空间可能无法完全捕获数据点的异常值。

OUTRES方法[402]检查较低维子空间的密度，以便识别相关的投影。 基本假设是对于给定的数据点 X，期望确定数据在其局部中非常均匀地分布的子空间。 为了表征数据点的局部性的分布，[402]中的工作计算子空间S中的数据点X 的局部密度，如下所示：

$$\text{den}(S, X) = |N(X, S)| = |\{\ddot{y} : \text{dists中}(X, Y) \leq \text{小号}\}| \quad (5.6)$$

这里，distS （X，Y ）表示子空间S中数据点X 和Y 之间的欧几里德距离。 这是密度的最简单的定义，尽管在OUTRES中使用其他更复杂的方法，如核密度估计[496]，以获得更多重新定义的结果。 核密度估计也在第4章中讨论。 这里的主要挑战是比较不同维度的子空间。 这是因为底层子空间的密度随着维数的增加而减小。 在[402]中已经表明，通过根据子空间的维度选择密度估计过程的带宽，可以获得跨越不同维度的子空间的可比较的密度估计。

此外，[402]中的工作使用统计技术，以便有意义地比较不同的子空间。 例如，如果数据均匀分布，则位于数据点距离s 内的数据点数量应由该子空间中数据的分数量来调节。具体而言，分数参数定义了二项式分布，表示该体积中的点数，如果该数据是均匀分布的话。 当然，人们对于从这种行为中显着偏离的子空间感兴趣。 使用统计测试来计算特定数据点X 的子空间的（局部）相关性。 这两个假设如下：

- 假设H_0：局部子空间邻域N（X，S）是均匀分布的。 假设 H_1：局部子空间邻域 （X，S）不是均匀分布的。
- N

Kolmogorov-Smirno ff优度检验[512]用于确定上述哪种假设是正确的。 重要的是要注意，该过程提供子空间有用性的概念，并且用于实现过滤条件，以从计算特定数据的异常值得分的过程中去除不相关的子空间。

点。 如果子空间通过假设条件H_1，则子空间被定义为相关的。 换句话说，使用必须满足的子空间的组合来计算异常值分数。

这个相关性标准。 该测试与有序子空间探索过程相结合，以确定相关的子空间$S_1 \ldots S_k$。稍后将详细描述该探索过程（参见图5.2）。

为了组合来自多个相关子空间的逐点分数，[402]中的工作使用从不同子空间获得的离群值得分的乘积。 因此，如果$S_1 \ldots S_k$ 是针对数据点X找到的不同的异常子空间，并且如果O（Si，X）是其在子空间Si中的异常值得分，则总异常值得分OS（X）定义如下：

$$OS(X) = \prod_i O(S_i, X) \quad (5.7)$$

singola@gmx.com

算法OUTRES（数据点：X，子空间：S）；
开始
对于每个属性我不是小号**做**
如果 Si ∉ {S} 我通过Kolmogorov-Smirno ff非均匀性测试**然后**
开始
使用公式5.6 或核密度估计计算den（Si，X）；使用公式5.8计算dev（Si，X）；
Compute O(Si, X) using Equation 5.9;
OS(X) = O(Si, X) · OS(X); OUTRES (X, Si);
结
束

图5.2：OUTRES算法

稍后将提供O（Si，X）的计算细节，尽管基本假设是低分表示更大的异常趋势。 在这种情况下使用产品而不是总和的优点是后者受到高分的支配，因此包含正常行为的一些子空间将主导总和。 另一方面，在产品的情况下，少量子空间中的异常行为将被大大放大。 这特别适用于异常检测问题。 值得注意的是，产品组合也可以被视为分数的对数之和。

为了确定离群值O（Si，X），只有当子空间的密度至少比平均值小两个标准差时，子空间才被认为对特定对象有意义。 这实质上是该子空间被认为是偏离的条件。 因此，子空间Si 中的数据点X 的偏差dev（Si，X）被定义为物体密度与X附近的平均密度的偏差除以两个标准偏差的比率。

$$dev(S_i, X) = \frac{\mu - den(S_i, X)}{2 \cdot \sigma} \tag{5.8}$$

μ 和σ 的值是在X 附近的数据点上计算的，因此该计算提供局部偏差值。 注意，异常子空间将具有dev（Si，X）> 1.子空间中的数据点的异常值得分是空间中的点的密度与其偏差的比率（如果它是异常子空间）。 否则，异常值得分被认为是1，并且它不影响公式5.7 中的乘积方式函数中的总异常值得分，用于组合来自不同子空间的数据点X 的得分。 因此，异常值得分为O（Si，X）定义如下：

$$O(S_i, X) = \begin{cases} \frac{den(S_i, X)}{dev(S_i, X)} & \text{if } dev(S_i, X) > 1 \\ 1 & \text{otherwise} \end{cases} \tag{5.9}$$

OUTRES算法的整个递归方法（参见图5.2）使用数据点X 作为输入，因此需要单独应用该过程以对每个候选数据点进行评分。 换句话说，这种方法本质上是一种基于实例的方法（如最近邻检测器），而不是选择子空间的方法之一

singola@gmx.com

以分离的方式预先（如功能装袋或HiCS）。 [402]中的观察结果是，非常低维度（例如，1维子空间）或非常高维度的子空间对于异常值检测不是非常有用的信息。 可以通过对候选子空间进行仔细的属性加法来识别信息性子空间。 在递归探索中，子空间中包含用于统计测试的附加属性。 当属性被添加到当前子空间Si时，利用非均匀性测试来确定是否应该使用该子空间。 如果它不相关，则丢弃子空间。 否则，异常值得分为O（Si，X）在该子空间中计算数据点，并通过将O（Si，X）与其相乘来更新异常值得分OS（X）的当前值。 由于不满足过滤条件的子空间的异常值得分被设置为1，因此它们不会影响这种乘法方法中的密度计算。 然后递归调用该过程以便探索下一个子空间。 因此，这样的过程可能探索指数个子空间，尽管由于修剪，实数可能非常适度。 特别地，非均匀性测试在探索期间修剪递归树的大部分。 给定数据点X的子空间探索的整体算法如图5.2所示 。 请注意，此伪代码假定整体异常值得分OS（X）类似于全局变量，可以由递归的所有级别访问，并且在第一次调用OUTRES之前将其初始化为1。 对OUTRES的初始调用使用空子空间作为S的参数值 。

5.2.9 基于距离的参考集

在[327]中提出了一种基于距离的方法，用于在数据的低维投影中发现异常值。 在这种方法中，不是试图在整个数据上找到异常低密度的局部子空间，而是为每个数据点提供特定的局部分析。 对于每个数据点X，识别出一组参考点 S（X）。 使用共享的最近邻距离[287]生成参考集S（X）作为候选者的top-k 最近点[287]（参见第4.3.3节）。

后该参考集合小号（X）一直identi音响版，用于相关子空间小号（X）被确定为集合Q（X维度），其中方差较小。 方差的特定阈值设置为S（X）中点的平均维度 - 特定方差的用户定义分数。 因此，该方法在子空间选择的关键步骤期间彼此独立地分析各个维度的统计。 X 的欧几里德距离被计算为由Q定义的子空间中的参考集S（X）的平均值。（X）。 这由tt（X）表示 。 tt（X）的值 受Q（X）中的维数的影响。 子空间离群度SOD（X数据点的）是通过归一化该距离定义德音响TT（X由维度中的数字）Q（X）：

$$SOD(X) = \frac{tt(X)}{|Q(X)|}$$

使用各个维度的方差来选择子空间集Q（X）的方法是从子空间聚类方法得到的相当天真的推广，并且是一个相当可疑的设计选择。 这是因为该方法完全忽略了各个维度之间的相互作用。 在许多情况下，例如前面讨论的年轻阿尔茨海默病患者的例子，不寻常的行为表现在维度之间的依赖性的违反，而不是个体维度的差异。 各个维度的差异告诉我们它们之间的依赖性很少。 许多

singola@gmx.com

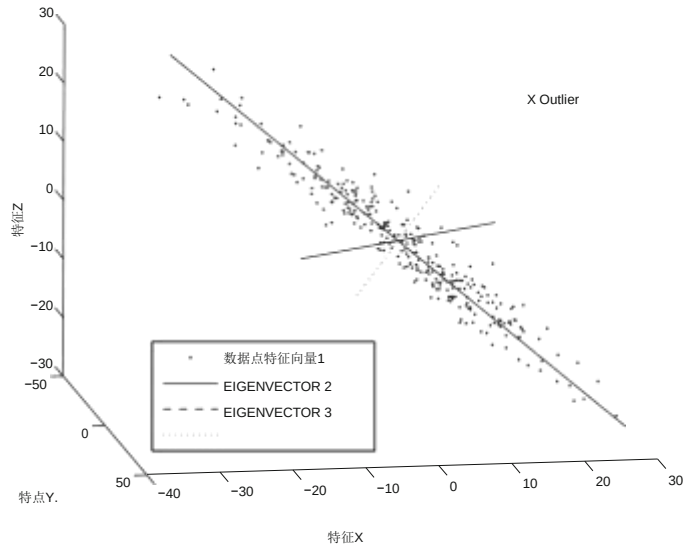


图5.3：重新访问图3.4 的示例：全局PCA可以在案例中发现异常值，其中整个数据沿着较低维度的流形排列。

其他见地技术如阀组[308，402]，它使用偏置子空间的选择，几乎总是使用尺寸之间的依赖关系作为关键选择标准。

另一个问题是使用单个子空间来获得异常值。 由于子空间选择标准忽略了依赖性，因此可以导致相关维度的删除。 在有趣的情况下，相关维度的数量有限，去除单个相关维度的负面影响可能比保持许多不相关的维度更加激烈。 使用单个子空间的特别成问题在于，如果在子空间选择中出现错误，则几乎没有机会从错误中恢复。 一般来说，这些类型的方法几乎总是胜过本章讨论的各种子空间集合方法（特征装袋[344]，旋转装袋[32]，RS-Hash [476]]和隔离森林[367]）。

5.3 广义子空间

虽然轴平行方法对于在大多数实际设置中发现异常值是有效的，但是在点沿着数据的任意低维流形排列的情况下，它们对于发现异常值并不是非常有用。 例如，在图5.4的情况下，来自二维数据的1维特征不能找到异常值。 另一方面，可以找到局部的1维相关子空间，使得大多数数据沿着这些局部的1维子空间对齐，并且剩余的偏差可以被分类为异常值。 尽管由于其低维度，这个特定数据集对于异常值检测似乎相对容易，但是随着维数的增加，该问题可能变得更具挑战性。

这些算法是以下两类算法的概括：

- 第3 章讨论的基于PCA的线性模型找到了数据中的全局相关区域。 例如，在图5.3的情况下，异常值可以有效

singola@gmx.com

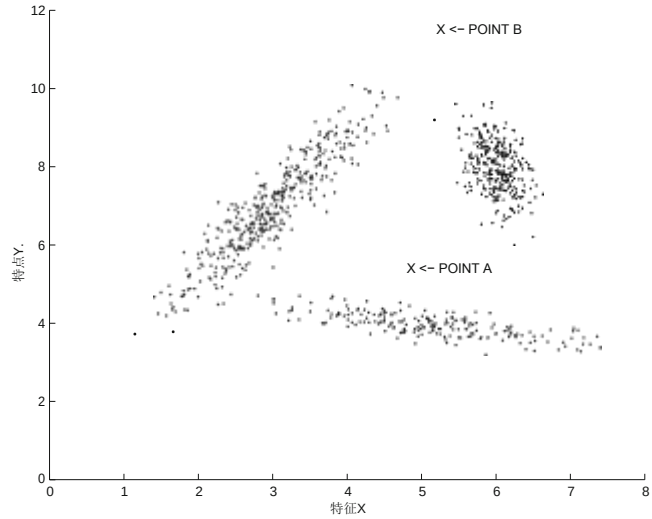


图5.4：重新审视图2.9 的示例：通过确定与基于PCA的本地群集的偏差，可以最好地发现异常值。 轴平行子空间异常值和全局PCA都不能捕获这样的簇。

通过确定这些全球相关方向来识别。 但是，没有这样的全球性
相关方向存在于图5.4中。

- 当数据沿低维轴平行子空间簇自然对齐时，本章前面讨论的轴平行子空间异常值可以找到偏差。 然而，在图5.4中并非如此，其中数据沿任意相关方向对齐。

广义子空间分析的目标是将这两种算法中的思想结合起来。 换句话说，期望与异常值发现同时确定任意定向的子空间。 在下文中，我们将讨论用于发现这种广义子空间的几种方法。 我们还将讨论一些发现非线性子空间的方法，其中数据沿局部非线性流形分布。

5.3.1 广义投影聚类方法

使用广义投影聚类方法可以部分解决这个问题，其中聚类在数据的任意对齐子空间中被识别[7]。 [7中讨论的方法]具有内置机制，以确定除群集之外的异常值。 这些异常值自然是不与簇对齐的数据点。 但是，该方法并未针对发现异常值进行特别优化，因为该方法的主要目的是确定群集。 发现异常值是聚类算法的副产品，而不是主要目标。 因此，该方法有时可能会发现较弱的异常值，这些异常值对应于数据中的噪声。 显然，需要通过使用离群值评分机制来区分不同点来正确区分强弱异常值的方法。 最简单的方法是计算每个候选异常值到每个聚类质心的局部马哈拉诺比斯距离。

singola@gmx.com

点到集群质心的距离仅使用该集群的均值和协方差矩阵。 第4 章第4.2 节（参见公式4.2）描述了局部马哈拉诺比斯距离的计算。 对于任何给定点，其最小马哈拉诺比斯距离（即到最近质心的距离）被报告为其离群值。

为了提高稳健性，应该使用随机方法以多种方式对数据进行聚类，并对不同模型中的逐点分数进行平均。 组合来自多个子空间的分数非常重要，因为使用子空间聚类发现的各个子空间并没有告诉我们关于异常值检测的相关子空间。 但是，使用聚类时发现的异常值通常会在使用集合方法时继承底层聚类的属性。 因此，在数据的子空间中使用簇会产生子空间敏感的异常值。[406]提供了轴并行设置的一个特定示例，其中显示了使用Multiple-Proclus方法的多个聚类的得分组合大大提高了聚类算法的单个应用程序的性能。

可以使用许多其他广义投影聚类方法，并且可以在[23]（第9章）中找到详细的调查。其中许多方法都非常有效。 这种方法的一个优点是，一旦预先确定了集群，评分过程就非常有效。 此外，模型构建过程通常需要少于O（N 2）的时间，这是大多数基于距离的检测器所需的。

5.3.2 利用Instance-Speci fi c参考集

为了确定针对特定数据点的局部性而优化的离群值，确定针对被评分的候选数据点X 优化的局部子空间是至关重要的。 确定这样的子空间是非常重要的，因为它通常不能从数据的本地聚合属性推断出来，用于检测稀有实例的行为。 最近在[328]中提出了一种方法用于使用参考集来发现广义子空间中的异常值。 早期广义子空间聚类方法的主要差异是局部参考集是特定于各种数据点的，而聚类提供了一组固定的参考集，用于对所有点进行评分。这种灵活性的代价是找到每个特定点的运行时间
参考集是O（N ）。 因此，该方法需要O（N 2）时间进行评分。

对于给定的数据点X，该方法连接的NDS全维度k-nearest的邻居 X。 这提供了一个参考集小号与平均向量μ。 将第3 章的PCA方法应用于局部参考集S 的协方差矩阵Σ（S），以便以递增的方差顺序确定关键特征向量e1 ... ed，具有相应的特征值
λ1 ≤ λ2 ... ≤ λD。 第3 章第3.3 节的讨论执行了相同的步骤[493]
除了它们是在全局基础上执行，而不是在本地参考集S上执行。
即使包含所有 d 维，也可以使用局部特征值缩放，将数据点X 的归一化异常值得分创建为数据的质心μ ，如第3章所述：

$$Score(\overline{X}) = \sum_{j=1}^{d} \frac{|(\overline{X} - \overline{\mu}) \cdot \overline{e_j}|^2}{\lambda_j} \tag{5.10}$$

如第2章第2.2.2.2 节所述，这可以近似建模为每个数据点具有d个自由度的x2分布，并且可以相互比较不同数据点的异常值。 使用这种方法
[493]在全球数据分析的背景下。 Chandola等人的调查报告。 [125]

singola@gmx.com

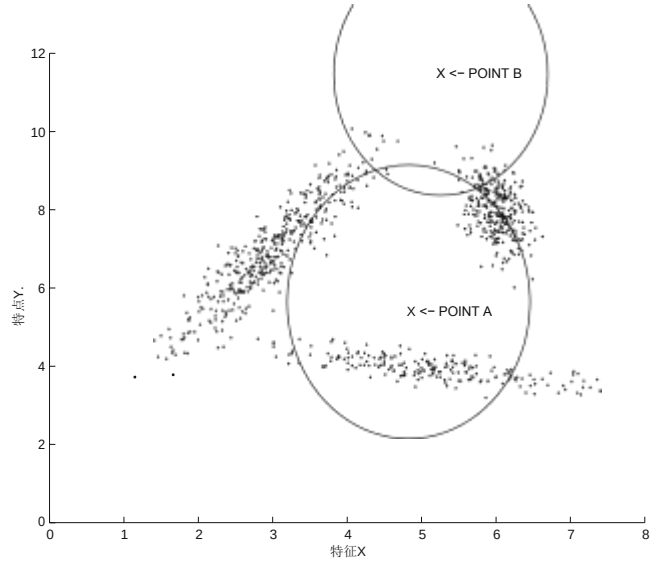


图5.5：本地参考集有时可能包含来自多个生成机制的点

提供了一个更简单的说明。 请注意，此方法可以视为软PCA的本地版本，并且不必使用子空间。

[328]中的工作与这种基本方法不同，因为它强调预先选择用于评分的特征向量的子集。 这主要是因为工作被定位为广义子空间离群值检测方法。 正如我们稍后将讨论的，这不一定是可取的。

选择具有最小特征值的δ 特征向量[3] 用于得分计算。 相应地，修剪得分德网络定义的基础上最小δ ≤ d 的特征向量：

$$Score(\overline{X}, \delta) = \sum_{j=1}^{\delta} \frac{|(\overline{X} - \mu) \cdot e_j|^2}{\lambda_j} \tag{5.11}$$

如何为特定数据点X固定δ 的值？得分是具有δ自由度的x2分布。 在[328]中观察到，通过将x2分布视为Γ分布的特殊情况，可以参数化δ 的值。

$$Score(\overline{X}, \delta) \sim \Gamma(\delta/2, 2)$$

通过选择δ 的值来为每个数据点特定地选择δ 的最佳值，以便基于该模型确定最大的不可能偏差。 这是通过使用上述分布的累积密度函数来完成的。 虽然这个值可以直接用作异常值，但也在[328]中显示，该分数如何转换为更直观的概率值。

然而，这种方法有几个缺点。 这些弱点源于使用单个子空间作为相关维度集的缺乏鲁棒性

[3] [328]中的工作使用δ作为最长特征向量的数量，这只是一个符号差异，但这里要注意以避免混淆。

singola@gmx.com

构建参考集，以及如何在分数计算中使用硬修剪方法。 我们将在下面详细讨论这些问题：

- 该方法使用单个子空间来使用本地参考集S来发现异常值。 如果未精确确定局部参考集S ，则这将不提供局部相关的正确方向。 使用单个子空间是有风险的，特别是使用基于弱聚合的提示时，因为通常可能无意中删除相关的子空间。 这可能会产生极大的影响。 使用多个子空间可以是在这样的场景中，例如在[提出的方法更相关的32，308，344，367，402，406]。

- 在使用全维k最近邻距离识别参考集时存在固有的圆形性，特别是如果距离在全维度中没有明确定义的话。 参考集中的点的选择和子空间的选择明显地以循环方式相互影响。 这是子空间分析中经典的“鸡与蛋”问题，这在[5]中首次提出]。 在这种情况下的分析需要同时而不是顺序。 众所周知，用于处理几乎所有问题域中的圆形度的最稳健的技术（例如，投影聚类方法）使用迭代方法，使得问题的点特定和维度特定方面能够彼此交互。 然而，这不是[328]中的情况，其中使用顺序分析。

  特别地，当使用全维距离时，可能发生在确定本地参考集期间可以使用许多局部不相关的特征。 因此，该集合可以包含来自多个生成机制的数据点，如图5.5所示。 当不相关要素的数量未知时，参考集中的特定点数将无法避免此问题。 使用较小的参考集大小可以在一定程度上减少这种情况发生的可能性，但是永远无法保证，特别是在使用许多不相关的特征时。 另一方面，减小参考集大小也可以产生相关超平面，其特征值统计超过了一组特别小的参考点。 事实上，这些问题的真正挑战在于正确选择参考集; 这种方法使这个问题变得微不足道。

- 由于公式5.10 的分母中的特征值已经对它们的相对重要性（或相关性）提供了软加权，因此不必以硬方式选择特定的一组特征向量。 例如，如果对于λi的较大值，数据点沿该方向显示甚至更大的偏差，则这样的异常值将被维度预选错过，或者将包括其他不太相关的维度。 一个例子是图5.5中的异常值B ，它沿着较长的特征向量对齐，因此最长的特征向量是关于其异常行为的最有用的信息。 特别是，选择δ的方法最小特征向量隐含地假设属性的相关性按特征值幅度排序。 虽然这对于基于聚合的聚类算法通常是正确的，但由于异常值的异常性质，在异常值分析中通常不是这样。 异常值沿长特征向量对齐的可能性根本不常见，因为两个高度相关的属性可能经常表现出类似相关性质的高度不正常行为。 这个例子还表明，离群值分析的罕见性质对于基于聚合的度量是多么脆弱。 这是因为稀有的原因不同，这在聚合统计中是无法完全捕获的。 这种观察证明了子集选择方法从聚类中直接推广的事实

singola@gmx.com

（基于聚合），通常不适合或优化（异常分析）异常值分析。 使用所有维度的一个优点是它可以降低到具有相同维度的局部马哈拉诺比斯距离，并且可以在不同异常值的分数中实现更好的可比性。 在这种情况下，可以更简单地从x2（d）分布导出直观概率值。

高维度的情况是极其困难的，并且可以理解的是，没有给定的方法能够完美地解决这些问题。

5.3.3 旋转子空间采样

旋转子空间采样方法最近在[32]中被提出作为一种集合方法，它改进了特征装袋[344]。 这种方法也称为旋转装袋。 正如特征装袋被设计用于发现轴平行子空间中的异常值一样，旋转装袋方法被设计用于发现广义子空间中的异常值。 与特征装袋的情况一样，这种方法是一种集合方法，它可以使用任何现成的异常值检测算法（例如，LOF）作为基本检测器。

旋转装袋的基本思想是在较低维空间中对随机旋转的子空间进行采样，并对该低维空间中的每个点进行评分。 得分来自
可以组合各种子空间以提供最终结果。 特别是这种方法
使用维度的子空间r = 2 + ⌊d /2⌋，这远低于典型的尺寸 -
特征装袋中使用的子空间的sionality。 这是因为轴旋转启用
从各个维度捕获不同程度的信息。 使用低维投影的能力对于诱导多样性并因此改善整体集合得分的质量也是有用的。 旋转的装袋算法的工作原理如下：

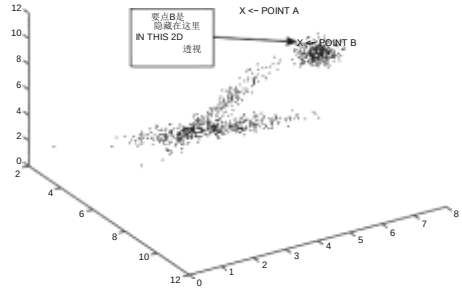
1. 确定数据中随机旋转的轴系统。
2. 从旋转轴系统采样$r = \lfloor 2 + \sqrt{d}/2 \rfloor$个方向。 沿着这些项目数据
   r directions.
3. 在投影数据上运行基础异常值检测器并存储每个点的分数。

组件得分可以通过（a）使用跨不同投影的点的平均得分，或（b）使用跨不同投影的点的最大得分来组合。 其他组合功能将在第6章中讨论。 使用很重要
分数标准化befo√re组合。

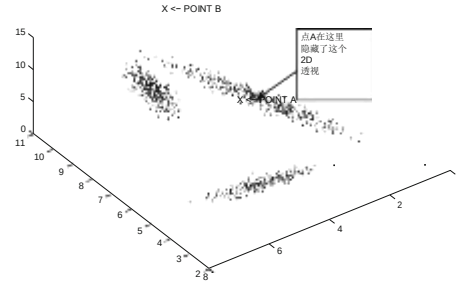
为了确定r = 2 + ⌊d /2⌋随机旋转相互正交的方向，
一d × r̆ 随机矩阵ÿ 产生，使得在矩阵中的每个值是均匀地解散在分布式在[-1，1]。 设Y的第t列用yt表示。 然后，使用如下的$y_1 \ldots y_r$ 的直接Gram-Schmidt正交化生成r个随机正交方向$e_1 \ldots e_r$：

1. $t = 1$; $e_1 = \frac{y_1}{|y_1|}$
2. $\overline{e_{t+1}} = \overline{y_{t+1}} - \sum_{j=1}^{t} (\overline{y_{t+1}} \cdot \overline{e_j})\overline{e_j}$
3. 将e_{t+1}标准化为单位范数。
4. $t = t + 1$
5. 如果t <r 转到第2步

singola@gmx.com



(a) 点'A'暴露，'B'是图中的这个2d视图中隐藏'A'



（b）暴露点'B'，隐藏在这个2d视

图5.6：在这个三维数据集中，点'A'和'B'暴露在不同的二维视图（投影）中。 但是，平均或最大化得分组合将暴露“A”和“B”。

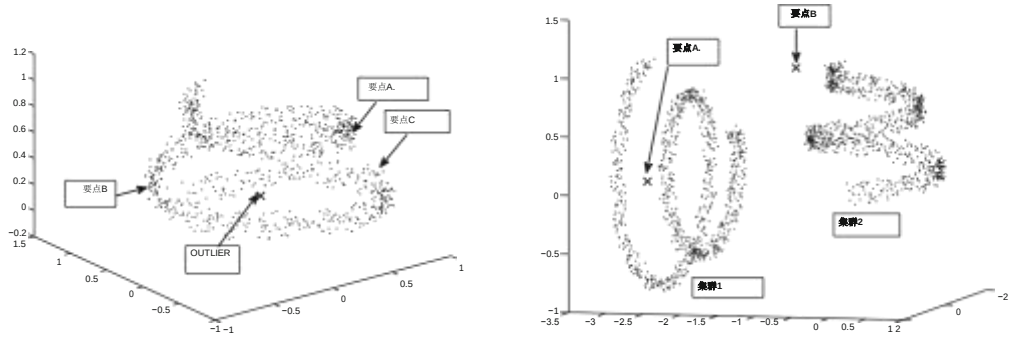
将得到的具有列e1 ..×. er的dr 矩阵用E表示。 所述Ñd 数据集d 进行变换和通过计算矩阵乘积投射到这些正交方向DE，这是一个N r个的矩阵[R维的点。 [32]中的结果显示该方法改进了功能包装。 通过×将其与其他整体技术相结合，可以获得进一步的改进。 旋转装袋使用更通用的模型来描述异常值而不是根据任意子空间进行特征装袋，因此使用集合更为重要。 发现与特定数据点相关的特定局部子空间通常是很困难的。 集合的强大功能在于，许多全局子空间选择的平均集合组合通常能够发现局部相关的子空间。 换句话说，整体本身比其个体成员更强大。 这一点可以从这类平均模型如何使用集合来减少代表性偏差的角度来解释（参见section第6章6.4.3.1 ）。 此外，就减少代表性偏差而言，最大化组合函数通常甚至更好。

图5.6说明了三维数据集的一个例子。 在这里，我们已经显示了不同的数据二维视图。很明显，在图5.6（a）的情况下，暴露出异常值“A”，而在图5.6（b）的情况下，暴露出异常值“B”。 但是，如果要使用通过在这两个视图上运行检测器获得的分数的平均值或最大化组合，则组合中的“A”和“B”点可能得分很高（因此被发现为异常值）。 这是集合方法如何克服各个探测器中的代表性偏差以提供更一般的模型的示例。 图中提供了用于克服代表性偏差的集合方法的这种能力的另一个例子第6章第6.4 节。 第6章第6.4.4 节还提供了以旋转装袋为中心的集合描述。

5.3.4 非线性子空间

子空间异常值检测的最常见情况发生在流形存在于任意形状的低维子空间中的情况。 这种模式的例子

singola@gmx.com



（a）全球数据分布 （b）本地数据分发

图5.7：任意形状的簇可以存在于低维流形中（重访第4章图4.3 ）

数据如图5.7所示。 该图与第4章的图4.3 相同。 事实上，4.2.1 节中提出的聚类方法提供了在这种非线性设置中发现异常值的方法之一。 由于本节中的讨论大致基于第4章第4.2.1 节中建立的框架，因此建议读者在继续进行之前重新审视该部分。

簇的任意形状提出了许多独特的挑战。 图5.7（a）和（b）中有几种情况，其中异常值位于非凸起图案的内部。 如果没有适当的局部非线性变换，发现这种异常值就变得更具挑战性。 这种提取是通过光谱方法实现的，可以将其视为一类特殊的核方法，可以发现任意形状的簇。

与4.2.1节的情况一样，提取数据的频谱嵌入。 这是通过构造k-最近邻图并计算其顶部特征向量来实现的。 由于构建频谱嵌入的方式，它已经使自身适应数据中的局部非线性模式，并且欧几里德距离可以用于变换的数据集。 与4.2.1 节的方法不同的 是，使用额外的步骤来降低频谱嵌入中的噪声[475]]。 频谱嵌入的主要问题是由邻域图中存在的坏边缘引起的，这些边缘连接不相交的簇。 因此，需要校正相似性图以提高嵌入的质量。 具体地，使用迭代方法来构建频谱嵌入，其中针对点对之间的错误计算校正相似性矩阵（用于构建嵌入）。 谱嵌入用于校正相似度矩阵，校正后的相似度矩阵用于构造谱嵌入。 这里的基本思想是来自相似性的光谱嵌入
矩阵S 本身将创建嵌入，从中可以构造新的相似性矩阵S^j。 在新的表示中，弱相关的点彼此远离
在相对基础上（如图5.7（a）中的点A和C ），强相关点相互移动。 因此，S 和S^j 的Hadamard乘积用于调整S：

$$S \Leftarrow S \circ S^j \tag{5.12}$$

这具有校正原始相似性矩阵 S 以减弱噪声的效果

singola@gmx.com

相对基础上的相似之处。 新的相似性矩阵用于再次创建新的嵌入。 重复这些步骤进行少量迭代，并使用最终嵌入来对点进行评分。 该过程在[475] 中有详细描述。 嵌入的每个新维度都标准化为单位方差。 甲k-nearest邻居检测器是为了返回连接NAL离群得分上嵌入执行。 通过在变换空间中使用k-最近邻检测器，可以有效地使用对原始空间中的非线性子空间模式敏感的依赖于数据的距离函数。 虽然工作在[475]没有讨论它，可以通过使用具有不同参数的多个聚类的平均分数来加强该方法。

5.3.5 回归建模技术

通过识别违反属性依赖性的点（例如，前面讨论的年轻阿尔茨海默病患者），也可以使用回归模型进行子空间离群值检测。 基本思想是将d维无监督异常值检测问题分解为一组d回归建模问题。 可以使用off-the-shelf回归模型使用剩余的（d 1）属性预测每个属性。 对于每个实例，聚合预测各种属性的平方误差以创建异常值分数。 第7章的7.7 节详细讨论了这种方法。 当使用诸如随机森林的某些类型的基础检测器时，该方法可以被解释为局部子空间异常值检测方法。 这些连接在7.7节中讨论。

5.4 子空间分析的讨论

在高维异常值分析中存在差异的主要原因是由于掩蔽了某些维度的局部噪声和无关性质的影响，这通常也表现在距离的集中。 在最近的调查[620]中已经声称，文献只关注距离集中问题和“弃权”讨论由不同生成机制引起的局部相关维度的影响（从而声称后者是一种新的洞察力）调查[620]）。 这是一个不正确且特别令人惊讶的断言，因为局部特征选择（相关性）和距离集中的各个方面已经在文献中进行了广泛的研究以及它们之间的相互联系。 在[原作4]（和几乎所有的随后的工作[308，344，402]）提供的图示（类似于图5.1）和不相关的属性如何（局部地）掩盖二FF erent特征SPECI音响Ç离群值相当详细的讨论数据视图。 在这种图示说明的背景下，[4]中阐述了关于本地不相关属性的内容：

> “因此，通过使用全尺寸距离测量，由于嘈杂和不相关的维度的平均行为，很难有效地确定异常值。 此外，不可能先验地修剪特定的特征，因为不同的点可能会显示不同类型的异常模式，每种模式都使用不同的特征或视图。“

全局特征选择在高维数据的INE FF ectiveness事实上形成用于子空间分析，这可以被认为是局部的特征选择方法，或者局部降维的方法[在激励原因7，121]。 在最早的一个作品的动机讨论中，详细讨论了局部子空间分析与高维数据中全局特征选择的无效性的这些联系。

singola@gmx.com

子空间分析[5，263]。 在这一点上，这些结果是众所周知的，并建立了[4]智慧。 虽然可以通过仔细调整来减少距离浓度的影响
分析信息维度的一部分，这种情况（通常）对子空间分析不感兴趣。

尽管嘈杂和无关的属性掩盖了外界，但最明显的是，观察肯定不是新的，距离集中和局部特征相关性这两个因素密切相关。 事实上，即使是调查中的实验模拟[620]表明浓度效应倾向于在存在太多不相关属性的环境中共同发生。 当然，这不是一个硬性规则；然而，对于基于距离的探测器来说，这是一个重要的问题。 子空间分析的有趣案例（通常）显示了两个属性的某些级别。 即使有限的距离集中度也会影响全维距离算法的有效性，因此这种影响对异常值分析的检验非常重要。 应该注意的是，噪声和无关属性更可能导致距离集中。 例如，对于均匀分布的数据，所有属性都有噪声，浓度效应极端，并且通过全维方法很难发现偏离相对较小维度的异常值。 在这种情况下，从全维度距离或基于密度的角度来看，所有数据点都具有几乎相同的异常分数，并且可以根据局部无关特征或距离集中效应等效地理解这一点。 当然，实际数据集不是均匀分布的，但是不同的数据集在不同的数据集中都存在不同的相关特征和集中效应。 子空间分析的一般假设是，添加更多维度通常不会成比例地增加特定异常值的更多信息。 具有挑战性的异常值通常由少数维度的行为决定，当点特定信息不随数据维度大幅增加时，即使适度的浓度效应也会对全维算法产生负面影响。 无关属性的数量越多，基于全维度距离的方法的计算可能就越错误。 在光谱的另一端的一个极端例子是，异常值表示每个维度中的信息和异常行为，因此异常值特征随着维度的增加而变得更强。 然而，在这种相当不感兴趣的情况下，由于异常值显示出许多相关特征并且通常也不符合剩余数据的距离集中行为，因此在大多数情况下，基于全尺寸距离的平凡算法将容易地发现它。 一般而言，信息维度也随着数据维度显着增加的情况对子空间分析不那么有趣，因为在这种更容易的情况下全维掩蔽行为变得不那么突出。 子空间分析不排除通过全维分析也可以找到更明显的偏差的可能性。

有许多高维数据集，可以通过更简单的全维算法执行有效的异常值检测。 特定算法对全维数据集的有效性取决于应用场景和提取特征的方式。 如果在考虑特定异常检测场景的情况下提取特征，则全维算法可能运行良好。 尽管如此，子空间分析通常会发现其他全维算法不易发现的异常值。 实际上，子空间分析不应该被视为一种独立的方法，而是作为一种集合方法来提高各种类型的性能

[4]一些最早的方法甚至将这些技术类别称为局部降维[121]，以强调增强的和不同的局部特征选择效应，这是由于不同的生成机制而产生的。

singola@gmx.com

基础探测器。 无论是开发用于高维异常值检测的特定基本方法（如子空间直方图），还是将其包裹在现有检测器周围（如特征和旋转套袋），将这一原理纳入异常值检测都有明显的好处。 此外，子空间分析提供了关于异常因果关系的有用见解；可以使用相关的局部子空间来提取对相关特征的特定组合的理解。 一个简单的例子是本章前面讨论的年轻阿尔茨海默病患者的情况。

异常值由于其非常罕见的特性，通常可以隐藏在高维数据集中的小维度组合中。 子空间分析对于这种情况特别有趣。 另一方面，当更多维度确实增加（显着）更多信息时，这将成为分析的一个简单案例，不再有趣。 在更多的情况下，绝大多数噪声维度使得从基于密度或数据稀疏性的角度区分数据点变得困难。 当异常值存在于适度数量的局部相关维度时，子空间分析在这些情况下特别有效。 这一观察结果也指出了一个特别困难的案例，其中局部无关维度的数量随着数据维度的增加而增加，而且很多维度也仍然缺乏相关性（但并不具有很强的相关性）。 这种情况经常发生在包含数千个维度的数据集中。 对于所有类的全维和子空间方法，这样的数据集仍然是未解决的情况。

总而言之，在设计子空间方法时需要牢记以下原则：

- 罕见的地区的直接勘探是PO小号锡布尔赫丁，虽然这是因为组合爆炸的计算挑战性[4]。
- 基于聚合的方法仅提供关于相关子空间的弱提示。 这些方法的主要功能在于通过组合不同子空间的结果，以集合为中心的设置。
- 整体的各个组成部分应考虑到效率考虑因素。 这是因为高效的组件能够实际使用更多的组件以获得更高的准确性。

一个有趣的观察是，即使使用弱基探测器，将它们组合通常也会产生非常强的结果。 功能包装，子空间直方图和旋转装袋等方法的成功基于这一事实。 注意，在每种情况下，底层基本检测器不是特别强；然而，最终的异常值检测结果非常有效。 高维离群检测和集合分析领域的最新进展非常重要。 尽管取得了这些进步，但许多高维数据集仍然是异常值分析的挑战。

5.5 结论和总结

用于离群值检测的子空间方法用于这样的情况，其中数据点的离群值趋势被大量本地无信息维度的噪声影响所稀释。 在这种情况下，通过搜索数据点与正常行为显着偏离的子空间，可以显着地锐化异常值分析过程。 最成功的方法为候选异常值识别多个相关子空间，然后组合来自不同子空间的结果，以便创建更健壮的基于集合的排名。

singola@gmx.com

异常值分析是所有子空间分析问题中最常见的问题。 这种差异源于异常值的罕见性质，这使得直接的统计分析更加困难。 由于子空间分析和局部特征选择是相关的，值得注意的是，即使对于全局特征选择，与聚类和分类算法相比，用于离群值分析的已知方法也很少。 原因很简单：足够的统计证据通常无法用于稀有特征的分析。 强大的统计数据是关于更多数据的，而异常值则是关于数据的减少以及与大多数数据的统计不一致！在高维子空间分析的特定情况下，包含统计一致性的区域和子空间告诉我们很少关于不一致的互补区域，因为后者的潜在区域远大于前者。 特别是，具有最大聚合一致性的局部子空间区域不一定揭示具有最大统计不一致性的稀有点的任何信息。

虽然许多最近用于子空间分析的集合方法已经取得了巨大的成功，但是特别困难的情况是大量维度与弱相关（但不是非常相关），甚至更多维度在本地无关紧要。 这些情况通常发生在包含数千个维度的数据集中，并且现有方法仍未解决。 虽然令人怀疑的是，问题的更多不同变化将完全解决，或者在所有情况下都能完全发挥作用，但目前可用的技术在许多重要场景中都有效。 能够设计这样的方法有许多优点，因为它们在识别异常原因方面可以提供许多见解。 主要的挑战是离群值分析是如此脆弱，通常不可能对汇总数据分析得出的推论做出有关断言。 效率问题似乎与高维异常值分析中的有效性密切相关。 这是因为异常值的搜索过程可能需要探索数据的多个本地子空间以确保稳健性。 随着现代计算机计算能力的不断提高，人们仍然希望这个领域变得越来越易于分析。

5.6 书目调查

在高维数据的情况下，研究两种不同的线的存在，其中一个调查高维异常检测[的电子FFI c iency58，219，557]，另一个调查高的电子FF ectiveness的更基本的问题尺寸离群检测[4]。 不幸的是，这两种作品之间的区别有时在文献中是模糊的，尽管这些作品显然是不同的作品，并且具有非常不同的动机。 应当指出的是，方法[讨论58，219，557]都是全维方法，因为异常值是根据它们的全维偏差定义的。 尽管[557]的方法使用投影进行索引，但这仅用作近似值以提高异常值检测过程的效率。

在高维情况下，（全维）离群值检测的效率也成为一个问题，因为大多数离群值检测方法需要在高维度上重复进行相似性搜索以确定最近邻居。 由于两个因素，这些方法的效率降低：（i）计算现在使用更大数量的维度，以及（ii）修剪方法和索引方法的有效性随着维数的增加而降低。 在广泛的相似性搜索文献中，这些问题的解决方案仍未得到解决。 因此，在高维的背景下，不太可能实现更有效的相似度计算。

singola@gmx.com

离群点检测，虽然取得了一些成功已经声称对改善高维异常检测的电子FFI ciency在[提出的方法58，219，557]。 总的来说，目前还不清楚这些方法如何与相似性搜索文献中用于索引高维数据的大量技术相比较。 本章根本没有研究效率问题，因为如果它甚至不能提供有意义的异常值，那么全维异常值检测技术的效率并不重要。 因此，本章的重点是在低维投影的背景下重新定义异常值检测问题的方法。

子空间异常值检测的问题首先在[4]中提出。 在本文中，提出了一种演化算法来发现可能存在异常值的低维子空间。 在[327]中提出了基于距离的离群点检测方法。 在[411]中提出了另一种基于距离的子空间离群检测方法。 一些方法也已通过ran- domly采样的子空间和组合从二FF erent子空间[得分提出了离群值分析344，367]。 特别是，[344试图结合来自这些不同子空间的结果，以便对异常值进行更稳健的评估。 这些基本上是基于集合的方法，试图通过对来自不同特征集的结果进行包装来提高检测稳健性。 这些方法的主要挑战是，在异常值隐藏在数据的特定子空间中的情况下，随机抽样可能无法很好地工作。 [308]中的工作可以被认为是[344]中广泛方法的推广，其中只有高对比度子空间被选择用于异常值检测的问题。 在[413]中讨论了用于偏置子空间选择的信息理论测量的使用。

上隔离的森林工作有关的早期工作使用随机森林和森林集群为集群，相似度计算，稀疏编码，离群去tection [99，401，491，555]。 特别是，[401]中的工作创建了极其随机的聚类森林（ERC-Forests），用于聚类和编码。 隔离林可以被视为ERC-Forest的一个特例，其中每个节点的试验次数设置为1，树木生长到全高。 隔离森林的许多变化[367]，例如半空间树[532]，已被提议。 隔离树和半空间树之间的主要差异在于后者是固定高度的完全平衡树，并且通过拾取每个属性的最小和最大范围之间的中间点来执行分割。 此外，每个属性的最小和最大范围以扰动的方式定义以诱导多样性。 测试实例的叶节点中的数据点的数量乘以叶节点的数量，以在单个集合组件中获得其异常值。 这些分数在不同的半空间树上取平均值。 在不同树木的深度不同的情况下，每个组分得分与叶节数的相乘是有帮助的[547]]。 最近，在[476]中提出了子空间直方图技术，称为RS-Hash。 该技术平均不同尺寸和形状的网格区域中的对数密度，以提供最终得分。 该方法使用随机散列来提高效率并且需要线性时间。 这些方法也可用于流数据。

网络连接的nding从SPECI音响Ç点外围子空间逆问题[进行了研究605，606，607]。 在这些方法中，为了加速边远子空间的搜索过程，提出了各种修剪和进化方法。 [59]中的工作也定义了外围物体的特殊性质，既包括整个种群（全球性质），也包括它所属的特定亚种群（当地财产）。 这两种方法都提供了关于底层数据的不同但有意义的见解。 在[606]中提供了用于在高维数据中发现外围子空间的遗传算法]。 为了加快完成功能评估，

singola@gmx.com

提出了一种方法，通过使用边界策略来加速k-最近邻距离的计算。 在[607]中提供了用于在高维数据中发现外围子空间的更广泛的框架。 在[582]中提出了一种使用双向搜索来发现偏离子空间的方法。 在该方法中，首先使用全维方法来确定异常值。 随后，检测并报告来自这些异常点的关键异常子空间。 在[405]中提出了一种使用规则来解释异常值对象的上下文的方法。

许多用于子空间离群勘探排名方法已经在[已经提出了402，403，404]。 在这些方法中，在数据的多个子空间中确定异常值。 不同的子空间可能提供有关不同异常值或相同异常值的信息。 因此，目标是以稳健的方式组合来自这些不同子空间的信息，以便报告最终的异常值集。 [402]中提出的OUTRES算法使用递归子空间探索来确定与特定数据点相关的所有子空间。 来自这些不同子空间的离群值得分被组合以提供最终值。 用于对子空间异常值进行排序的工具包在[403]。 在[406]中提出了一种使用多个数据视图进行子空间异常检测的更新方法。 在[64]中提出了多媒体数据库中子空间异常检测的方法。

用于子空间离群值检测的大多数方法在数据的轴并行子空间中执行探索。 这基于补充假设，即密集区域或簇隐藏在数据的轴平行子空间中。 然而，在最近的工作中已经表明，密集区域通常可以位于数据的任意方向的子空间中[7]。 这种聚类方法可以与5.2.4 节中讨论的方法结合使用，以发现异常值。 [328]中的另一项工作提供了基于本地参考集而不是集群的解决方案。 [32]提出了一种旋转的装袋方法; 这可以被视为任意导向案例的特征装袋方法的类比。 最后，在[475]中提出了一种在非线性子空间环境中发现异常值的方法。

最近，在动态数据和数据流的背景下也研究了异常值检测的问题。 在[604]中提出了SPOT方法，其能够确定来自高维数据流的预计异常值。 该方法采用基于窗口的时间模型和衰减单元摘要来从数据流中捕获统计数据。 最稀疏的子空间是通过各种有监督和无监督的学习过程获得的。 这些用于识别预计的异常值。 采用多目标遗传算法从训练数据中发现外围子空间。

高维异常值检测的问题也已扩展到其他应用特定场景，如天文数据[261]，不确定数据[26]，交易数据[255]和监督数据[619]。 在不确定的情景中，高维数据尤其具有挑战性，因为不确定情景中的噪声极大地增加了基础数据的稀疏性。 此外，可以获得不同属性的不确定性水平。 这有助于确定不同属性对异常检测目的的重要性。 在[26]中提出了用于不确定数据中离群点检测的子空间方法]。 在[619]中提出了用于高维离群点检测的监督方法。在这种情况下，识别出少量示例并呈现给用户。 然后使用它们来学习与对象的离群值相关的关键投影。 然后利用学习的信息来识别基础数据中的相关异常值。

singola@gmx.com