矩阵法

在数据挖掘和模式中

承认

算法基础

主编：曼彻斯特大学Nicholas J.Higham

暹罗的算法基础系列是一个短期的面向用户的书籍集的艺术状态的数字方法。由专家撰写的这本书为读者提供了足够的知识，使他们能够为应用程序选择合适的方法，并了解该方法的优点和局限性。这些书涵盖了从数值分析和科学计算中抽取的一系列主题。目标受众是研究人员和实践者，他们使用数学、工程和计算科学的方法和高水平本科生。

本系列书籍不仅为解决特定问题所用的一种方法或一类方法提供了数学背景，还解释了如何将该方法开发成算法并翻译成软件。这些书描述了一种方法的适用范围，并对解决问题和解释结果提供了指导。这一理论是在从业者能够接触到的层面上提出的。Matlab®软件是代码的首选语言，因为它可以在各种平台上使用，并且是原型设计、测试和问题解决的良好环境。

该系列旨在为易于访问的数字算法提供指南，包含其他地方不易找到的实用建议，并包括实现算法的可理解代码。

编辑部

|  |  |
| --- | --- |
| 彼得·本纳  切姆尼茨理工大学  约翰·R·吉尔伯特  加州大学圣巴巴拉分校  迈克尔·T·希思  伊利诺伊大学香槟分校  C.T.凯利  北卡罗来纳州立大学  克里夫·莫勒尔  数学作品  詹姆斯·纳吉  埃默里大学 | 戴安娜·奥利里  马里兰大学  罗伯特罗素  西蒙弗雷泽大学  罗伯特·D·斯克尔  普渡大学  丹尼·索伦森·赖斯大学  安德鲁·J·沃特恩牛津大学  亨利·沃尔科维奇  滑铁卢大学 |

系列卷

Eld\_n，L.，数据挖掘和模式识别中的矩阵方法

Hansen，P.C.、Nagy，J.G.和O'Leary，D.P.，去模糊图像：矩阵、光谱和过滤

Davis，T.A.，《稀疏线性系统的直接方法》

Kelley，C.T.，用牛顿法求解非线性方程组

拉尔斯·埃尔登

瑞典林平大学林平

数据挖掘中的矩阵方法与模式

承认

工业与应用数学学会

费城

版权所有©2007由工业和应用数学学会出版。

10 9 8 7 6 5 4 3 2 1

版权所有。在美利坚合众国印刷。未经出版商书面许可，不得以任何方式复制、储存或传播本书的任何部分。有关信息，请写信给工业和应用数学学会，地址：3600 University City Science Center，Philadelphia，PA 19104-2688。

本书可使用商标名称，但不包括商标符号。这些名称仅在编辑上下文中使用；不打算侵犯商标。

谷歌是谷歌公司的商标。

Matlab是Mathworks，Inc.的注册商标。有关Matlab产品信息，请联系Mathworks，Inc.，3 Apple Hill Drive，Natick，MA 01760-2098 USA，508-647-7000，传真：508-647-7101，info@mathworks.com，www.mathworks.com。

图6.2、10.1、10.7、10.9、10.11、11.1和11.3来自L.Eld\_n，《数据挖掘中的数值线性代数》，Acta Numer.，15:327–384，2006。经剑桥大学批准转载

按。

图14.1、14.3和14.4由作者根据出现在P.N.Belhumur、J.P.Hespanha和D.J.Kriegman中的图像构建，特征面与Fisherfaces：使用类特定线性投影的识别，IEEE Trans。模式分析。机器。《国际贸易法》，19:711–7201997年。

国会图书馆出版资料编目

Eld\_n，Lars，1944年-

数据挖掘和模式识别中的矩阵方法。

每厘米-（算法基础；04）包括参考书目和索引。

ISBN 978-0-898716-26-9（PBK.：alk.纸）

1。数据挖掘。2。模式识别系统数学模型。三。代数，线性。一、职务。

QA76.9.D343E52 2007年

05.74-DC20 2006041348

是注册商标。

目录

前言九

线性代数概念和矩阵分解

1. 数据挖掘和模式识别中的向量和矩阵3
   1. 数据挖掘和模式识别。……………….. 3
   2. 向量和矩阵。……………………………4
   3. 本书的目的。……………………………7
   4. 编程环境。………………………8
   5. 浮点计算。………………………8
   6. 符号和约定。……………………….. 11
2. 向量和矩阵13
   1. 矩阵向量乘法。………………………13
   2. 矩阵矩阵乘法。…………………….. 15
   3. 内积和向量范数。………………….. 17
   4. 矩阵规范。…………………………………18
   5. 线性独立性：基础。………………………20
   6. 矩阵的秩。……………………………21
3. 线性系统和最小二乘法23
   1. LU分解。………………………………23
   2. 对称正定矩阵。……………….. 25
   3. 微扰理论和条件数。………….. 26
   4. 高斯消去中的舍入误差。…………….. 27
   5. 带状矩阵。……………………………….. 29
   6. 最小二乘问题。………………………31
4. 正交性37
   1. 正交向量和矩阵。………………….. 38
   2. 初等正交矩阵。……………………40
   3. 浮点运算的数目。……………….. 45
   4. 浮点运算中的正交变换。.. 46

V

六、内容

1. QR分解47
   1. 正交变换为三角形。……….. 47
   2. 解最小二乘问题。…………………51
   3. 是否计算Q。………………….. 52
   4. QR因子分解的失败计数。………………….. 53
   5. 最小二乘问题解的误差。…….. 53
   6. 更新最小二乘问题的解。………54
2. 奇异值分解57
   1. 分解。…………………………….. 57
   2. 基本子空间。………………………….. 61
   3. 矩阵近似。………………………….. 63
   4. 主成分分析。…………………….. 66
   5. 解最小二乘问题。……………………66
   6. 最小二乘条件数与微扰理论

问题。…………………………………….. 69

* 1. 等级不足和不确定的系统。…………70
  2. 计算SVD。……………………………72
  3. 完成正交分解。……………….. 72

1. 降秩最小二乘模型75
   1. 截断SVD：主成分回归。………77
   2. 一种Krylov子空间方法。………………………80
2. 张量分解91
   1. 引言。………………………………….. 91
   2. 基本张量概念。………………………….. 92
   3. 张量SVD。…………………………………94
   4. 用hospd逼近张量。……………….. 96
3. 聚类与非负矩阵分解101
   1. k-均值算法。…………………………102
   2. 非负矩阵分解。……………………一百零六

二、数据挖掘应用

1. 手写数字分类113
   1. 手写数字和一个简单的算法。……………一百一十三
   2. 使用SVD基进行分类。……………………115
   3. 切线距离。………………………………122
2. 文本挖掘129
   1. 预处理文档和查询。…………….130 11.2矢量空间模型。…………………………131
   2. 潜在语义索引。…………………………135
   3. 聚类分析。……………………………………139

目录七

* 1. 非负矩阵分解。……………………一百四十一
  2. LGK标准化。……………………………一百四十二
  3. 平均性能。……………………………145

1. 网页搜索引擎的网页排名147
   1. 页面排名。………………………………………一百四十七
   2. 随机行走和马尔可夫链。……………………一百五十
   3. pagerank计算的幂方法。…………154
   4. 击打。…………………………………………一百五十九
2. 自动关键字和关键字句子提取161
   1. 显著性得分。…………………………………161
   2. 从k阶近似中提取关键句。……一百六十五
3. 张量SVD 169人脸识别
   1. 张量表示。……………………………一百六十九
   2. 面部识别。……………………………….172 14.3带软管压缩的人脸识别。……………一百七十五

三、计算矩阵分解

1. 计算特征值和奇异值179
   1. 微扰理论。……………………………180
   2. 幂法和逆迭代法。………………一百八十五
   3. 三对角形式的相似性约简。………………一百八十七
   4. 对称三对角矩阵的QR算法。……一百八十九
   5. 计算SVD。……………………………196
   6. 非对称特征值问题。………………197
   7. 稀疏矩阵。…………………………………一百九十八
   8. Arnoldi和Lanczos方法。…………………200。
   9. 软件。………………………………………二百零七

书目209

索引217

前言

这本书的第一个版本是由瑞典国家科学计算研究生院（NGSSC）组织的一个关于数据挖掘和科学技术应用的研究生课程的一套讲义。从那时起，该材料已被用于并进一步发展为林科-平大学的一门关于数据挖掘和信息技术数值算法的本科课程。这是计算机科学学生的第二门科学计算课程。

这本书主要是为那些以前参加过科学计算/数值分析入门课程的本科生编写的。它也可能对不同数据挖掘和模式识别领域的早期研究生有用，他们需要线性代数技术的介绍。

这本书的目的是证明有几个非常强大的数值线性代数技术来解决数据挖掘和模式识别的不同领域的问题。为了实现这一目标，有必要提供超出瑞典大学科学计算（数值分析）第一门课程通常所涵盖范围的材料。另一方面，由于这本书是面向应用的，因此不可能对所用线性代数算法的数学和数值方面进行全面的处理。

这本书由三部分组成。在简要介绍了数据挖掘和模式识别的几个领域之后，提出了线性代数概念和矩阵分解。我希望这足够学生在Matlabr等解决问题的环境中使用矩阵分解。给出了一些数学证明，但重点是矩阵分解的存在性和性质，而不是它们的计算方法。在第二部分中，线性代数技术应用于数据挖掘问题。当然，数据挖掘和模式识别的知识非常有限：我选择了非常适合线性代数技术的问题领域。为了智能地使用Matlab等强大的计算矩阵分解软件，需要对底层算法有一些了解。第三部分简要介绍了特征值和奇异值算法。

我没有雄心壮志写一本食谱书：“给定一个问题，这里有一个算法来解决它。”这将是困难的，因为这个领域太多样化，无法给出明确而简单的解决方案。相反，我的目的是给学生一套可以尝试的工具，但更可能的是，这些工具需要修改，以便对特定的应用程序有用。书中的一些方法是用matlab脚本描述的。他们不应该

九

X前言

被认为是严肃的算法，而不是为了说明目的而给出的伪代码。

该书的网页www.siam.org/book s/fa04上提供了一系列练习和电脑作业。

感谢NGSSC对制作原始讲稿的支持。讲稿已经被几个同事用过了。感谢Gene Golub和Saara Hyvo–nen提供的有用意见。我的几个学生通过指出不一致之处和提问帮助我改进了演示。我感谢BerkantSavas让我使用他在第10章的硕士论文的结果。三位匿名推荐人阅读了这本书的早期版本，并提出了改进建议。最后，我要感谢暹罗的系列编辑尼克·海厄姆仔细阅读了手稿。他深思熟虑的建议帮助我大大改进了内容和演示。

Lars Eld'en Linko–Ping，2006年10月

第一部分

线性代数概念和矩阵分解

第一章

数据挖掘和模式中的向量和矩阵

承认

1.1数据挖掘和模式识别

在现代社会，大量的数据被收集并存储在计算机中，以便以后提取有用的信息。在收集数据时，通常不知道以后会请求什么数据，因此数据库的设计并不是为了提取任何特定的信息，而是在很大程度上是非结构化的。从大型数据集中提取有用信息的科学通常被称为“数据挖掘”，有时还加上“知识发现”。

模式识别通常被认为是一种独立于数据挖掘的技术，但其定义是相关的：“接受原始数据并根据模式的‘类别’进行操作的行为”[31]。在这本书中，我们不会强调概念之间的差异。

数据挖掘有许多应用领域，从电子商务[10，69]到生物信息学[6]，从科学应用如金星火山分类[21]到信息检索[3]和互联网搜索引擎[11]。

数据挖掘是一门真正的跨学科科学，其中使用了计算机科学、统计学和数据分析、线性代数和优化等技术，通常以相当折衷的方式使用。由于申请的实际重要性，该地区现在有许多书籍和调查[24、25、31、35、45、46、47、49、108]。

一点也不夸张，那就是我们的日常生活充满了各种各样的情况，在这种情况下，我们往往在不知情的情况下，依靠先进的数学方法进行数据挖掘。线性代数和数据分析等方法是许多数据挖掘技术的基本组成部分。本书介绍了数学和数值方法及其在数据挖掘和模式识别中的应用。

三

1.2向量和矩阵

下面的例子说明了向量和矩阵在数据挖掘中的使用。这些例子展示了本书中讨论的主要数据挖掘领域，它们将在第二部分中进行更详细的描述。

在许多应用中，矩阵只是数据的矩形数组，元素是标量实数：

.

要用数学方法处理数据，必须添加一些数学结构。在最简单的情况下，矩阵的列被视为rm中的向量。

例1.1.术语文档矩阵用于信息检索。考虑以下五个文件的选择。关键词，我们称之为术语，用黑体字标记。

|  |  |
| --- | --- |
| 文件1： | GoogleTM Matrix P是互联网的一个模型。 |
| 文件2： | 如果从网页j到i有链接，则pij不是零。 |
| 文件3： | 谷歌矩阵用于对所有网页进行排名。 |
| 文件4： | 排序是通过求解矩阵特征值问题来完成的。 |
| 文件5： | 英格兰队在国际足联前十名中失利 |

排名。

如果我们计算每个文档中术语的频率，我们会得到以下结果：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 术语 | DOC 1 | DOC 2 | DOC 3 | DOC 4 | DOC 5 |
| 特征值 | 零 | 零 | 零 | 一 | 零 |
| 英格兰 | 零 | 零 | 零 | 零 | 一 |
| 国际足联 | 零 | 零 | 零 | 零 | 一 |
| 谷歌 | 一 | 零 | 一 | 零 | 零 |
| 互联网 | 一 | 零 | 零 | 零 | 零 |
| 链接 | 零 | 一 | 零 | 零 | 零 |
| 矩阵 | 一 | 零 | 一 | 一 | 零 |
| 页 | 零 | 一 | 一 | 零 | 零 |
| 等级 | 零 | 零 | 一 | 一 | 一 |
| 网状物 | 零 | 一 | 一 | 零 | 零 |

1.2.向量和矩阵

因此，在R10中，每个文档都由一个向量或点表示，我们可以将所有文档组织成一个术语文档矩阵：

.

现在假设我们要找到与查询“网页排名”相关的所有文档。这是由一个查询向量表示的，其构造方式类似于术语文档矩阵：

γ

零

0 Q=00

\_\_0\_

零

零

一

一

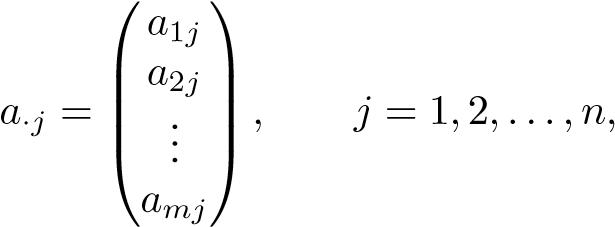
一

因此，查询本身被视为一个文档。信息检索任务现在可以表述为一个数学问题：找到靠近向量Q的A列。要解决这个问题，我们必须在R10中使用一些距离测量。

在信息检索应用中，通常情况下，尺寸m较大，为106级。此外，由于大多数文档只包含一小部分术语，因此矩阵中的大多数元素都等于零。这样的矩阵称为稀疏矩阵。

一些信息检索方法使用线性代数技术（例如奇异值分解（SVD））进行数据压缩和检索增强。第11章介绍了信息检索的向量空间方法。

通常，不仅将矩阵视为数字数组或向量集，而且将其视为线性运算符也是有用的。表示a的列



然后写

.

然后定义线性变换

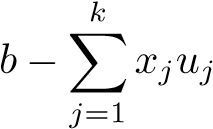
.

例1.2。手写数字的分类是模式识别中的一个模型问题。这里，向量用于表示数字。一位数字的图像是一个16×16的数字矩阵，代表灰度。它也可以通过叠加矩阵的列来表示为r256中的向量。一组n位数（比如手写的3）可以用矩阵a∈r256×n来表示，一个跨度a子空间的列可以用r256来表示。我们可以使用svd a=u∑v t计算该子空间的近似基。“3-子空间”的三个基向量如图1.1所示。



图1.1.美国邮政服务数据库中的手写数字[47]，以及3的基向量（底部）。

设b为表示未知数字的向量。我们现在想（自动地，通过计算机）将未知数字分类为0-9中的一个。给定一组3's，u1，u2，…，uk的近似基向量，我们可以通过检查是否存在基向量的线性组合来确定b是否为3，这样



1.3.本书的目的

很小。因此，我们在这里计算b的坐标，以uj kj=1为基础。

在第10章中，我们讨论了手写数字的分类方法。

数据挖掘的基本思想是从大数据集（通常是非结构化的）中提取有用的信息。因此，所采用的方法必须是有效的，并且通常是针对大问题专门设计的。在一些数据挖掘应用程序中，会出现巨大的矩阵。

例1.3。从互联网上所有可用网页中提取信息的任务是由搜索引擎完成的。谷歌搜索引擎的核心是矩阵计算，可能是常规执行的最大的计算[71]。google matrix p有几十亿个订单，即接近互联网上的网页总数。该矩阵是基于网络的链接结构构造的，如果从网页j到i有链接，那么元素pij是非零的。

下面的小链接图说明了一组带有大纲和inlink的网页：

1

2

3

4

5

6

构造了相应的链接图矩阵，使列和行表示网页，而列j中的非零元素表示网页j的大纲。

.

为了使搜索引擎有用，它必须使用网页质量的度量。谷歌矩阵用于对所有页面进行排名。排序是通过求解p的特征值问题来完成的；见第12章。

1.3本书的目的

本书的主要目的不是数字线性代数的教科书，而是对现代线性代数中一些技术的面向应用的介绍，重点是数据挖掘和模式识别。它在很大程度上取决于一个易于使用的编程环境的可用性，该环境实现了我们将要呈现的算法。因此，我们不再详细描述算法，而是提供足够的数学理论和数字背景信息，以便读者能够理解和使用嵌入在类似matlab[68]的软件包中的强大软件。

关于本书中使用的矩阵分解的数值和算法方面的更全面的介绍，请参阅任何最近的教科书[29、42、50、92、93、97]。大系统和稀疏系统的线性系统和特征值问题的解在[4，5]中进行了详细讨论。对于那些想要研究数值线性代数算法的详细实现的人来说，Fortran、C和C++的软件可以通过互联网免费获得[1 ]。

假设读者学习过线性代数和科学计算（数值分析）的入门课程。熟悉matlab等面向矩阵的编程语言的基础知识有助于理解演示。

1.4编程环境

在本书中，我们使用matlab[68]来演示概念和算法。我们的代码不应被视为软件；相反，它们旨在演示基本原则，我们强调的是简单性，而不是效率和健壮性。这些代码只能用于小型实验，而不能用于生产计算。

即使我们使用matlab，我们也要强调，可以使用任何实现现代矩阵计算的编程环境，例如mathematica[112]或统计数据包。

1.5浮点计算

### 1.5.1故障计数

不同算法的执行时间有时可以通过计算浮点运算的次数来比较，即使用浮点数的算术运算。在本书中，我们遵循标准程序[42]并分别对每个操作进行计数，我们对一个操作使用术语flop。因此，语句y=y+a\*x，其中变量是scalars，计为两个flop。

习惯上只计算最高订货期。我们强调，失败计数通常是效率和计算时间的非常粗略的度量，在某些情况下甚至可能产生误导。在现代计算机上，数据访问模式是非常重要的，因为它们总是具有内存层次结构。因此，在某些情况下，具有相同触发器计数的算法的执行时间可以按数量级变化。

1.5.浮点计算

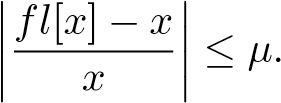
### 1.5.2浮点舍入误差

算法的误差分析将不是本书的主要部分，但我们将引用一些没有证据的结果。我们假设计算是按照IEEE浮点标准[2]进行的，因此，以下模型是有效的。

通常，实数x不能在浮点系统中精确表示。设fl[x]为代表x的浮点数，然后

（1.1）

对于一些人来说，令人满意的是，其中μ是浮点系统的单位舍入。从（1.1）我们可以看出，任何实数x的浮点表示中的相对误差满足



在IEEE双精度算法（Matlab中的标准浮点格式）中，单位舍入满足μ≈10−16。在IEEE单精度中，我们有μ≈10−7。

]是浮点运算的结果，其中

表示+、-、和/中的任意一个。那么，如果=0，

（1.2）

或者，等价地，

（1.3）

对于一些人来说，令人满意的是，其中μ是浮点系统的单位舍入。

当我们像（1.2）中那样估计浮点运算的计算结果中的误差时，我们可以将其视为正向误差。或者，我们可以将（1.3）改写为



对于一些满足的数e和f

| E≤μX，F≤μY。

换句话说，]是对微扰数据进行操作的精确结果。这是一个反向误差分析的例子。

可以用IEEE双精度表示的最小和最大的正实数约为10−308和10308（对应于负数）。如果计算结果给出了一个浮点数量级

v

w

图1.2.gjk算法中的向量。

小于10−308，则会发生一个称为下溢的浮点异常。同样，计算大于10308的浮点数也会导致溢出。

示例1.4（计算机图形中的浮点计算）。检测两个三维物体之间的碰撞是将图形应用于计算机游戏、动画和模拟的标准问题[101]。早期的不动点算法被用于计算机图形学，但现在这种计算通常是在浮点算法中进行的。这一领域的一个重要子问题是计算最接近原点的凸体上的点。这个问题可以用吉尔伯特-约翰逊-基尔西（gjk）算法来解决，它是迭代的。算法采用停止准则



对于迭代，向量如图1.2所示。当接近解决方案时，向量非常接近。在[101，pp.142–145]中，描述了在浮点算术中计算s（v，w）时可能出现的数值困难。这里，我们简单解释了当v和w是标量，s=v2-vw时的计算，它显示了与向量情况完全相同的问题。

假设数据不准确（它们是先前计算的结果；在任何情况下，它们都会出现表示错误（1.1）），

，

其中v和w相对较小，通常为μ量级。从（1.2）我们可以看到，每个算术运算都会产生一个相对误差（1.3），因此

，

1.6.符号和约定

我们假设的地方。计算出的数量的相对误差可以用

.

我们看到，如果是大的、近的，那么相对误差可能是大的。例如，当v=100，w=99.999时，我们得到

计算机图形应用程序，则]中的相对误差可能太大，以至于永远无法满足终止条件，迭代也永远不会停止。在gjk算法中，除上述情况外，还存在其他情况，当浮点舍入误差可能导致终止标准不可靠时，必须特别小心；见[101]。

在前面的例子中出现的问题称为取消：当我们用误差减去两个几乎相等的数字时，结果的有效数字更少，相对误差更大。有关IEEE标准和浮点计算中舍入错误的更多详细信息，请参见，例如[34，第2章]。线性代数算法的广泛舍入误差分析在[50]中给出。

1.6符号和惯例

我们将考虑带实数分量的向量和矩阵。通常向量用小写斜体罗马字母表示，矩阵用大写斜体罗马字母或希腊字母表示：

x∈rn，a=（aij）∈rm×n.

张量，即具有三个或更多索引的实数数组，将用书法字体表示。例如，

S=（sijk）∈rn1×n2×n3。

我们将用rm表示实场上维m的向量空间，用rm×n表示m×n矩阵的空间。

符号

，

其中1位于位置i，用于“规范”单位向量。从上下文来看，维度通常是显而易见的。

单位矩阵用i表示，有时我们强调尺寸，用i k表示k×k单位矩阵。符号diag（d1，…，dn）表示对角矩阵。例如，i=diag（1,1，…，1）。

第二章

向量和矩阵

我们假设读者知道线性代数的基本概念。为了完整起见，这里将简要介绍一些内容。

2.1矩阵向量乘法

如何定义线性代数中的基本运算是很重要的，因为它会影响一个人对抽象概念的心理形象。有时人们会认为，操作应该按一定的顺序进行，而这样的定义不要求排序。设A为m×n矩阵。考虑的定义

# 矩阵向量乘法：

（2.1）可以用符号来说明定义。

（2.2）

显然，向量y的不同分量的计算完全独立，可以按任意顺序进行。但是，定义可能会让人认为应该按行访问矩阵，如（2.2）和以下matlab代码所示：

十三

对于i=1:m y（i）=0；对于j=1:n

Y（i）=Y（i）+A（i，j）\*X（j）；结束

结束

或者，我们可以用以下方式编写操作。设a·j为a的列向量，然后我们可以写出

.

这可以用符号来说明：

（2.3）

在这里，向量是按列访问的。在matlab中，可以编写这个版本

对于i=1:m y（i）=0；

j=1:n时结束，i=1:m y（i）=y（i）+a（i，j）\*x（j）；

结束

结束或等效地，使用matlab的向量运算，

y（1:m）=0；对于j=1:n y（1:m）=y（1:m）+a（1:m，j）\*x（j）；

结束

因此，执行矩阵向量乘法的两种方法对应于更改代码中循环的顺序。这种书写方式还强调A的列向量作为基向量，X的分量作为相对于基的坐标的视图。

2.2.矩阵乘法

2.2矩阵乘法

矩阵乘法可以用几种方式进行，每种方式表示矩阵的不同访问模式。设A∈Rm×K和B∈Rk×N，矩阵乘法的定义是

（2.4）

与矩阵向量乘法（2.1）的定义相比，我们发现在矩阵乘法中，B中的每列向量都乘以A。我们可以将（2.4）表示为矩阵乘法代码。

对于i=1:m

对于j=1:n

对于s=1:k

C（I，J）=C（I，J）+A（I，S）\*B（S，J）端

结束

这是矩阵乘法的内积版本，在下面的等效代码中强调了这一点：

对于i=1:m

对于j=1:n

C（i，j）=A（i，1:k）\*B（1:k，j）端

结束

可以立即看到循环变量可以在3中排列！=6种不同的方法，我们可以编写一个通用矩阵乘法代码：

用于…用于…用于…

C（I，J）=C（I，J）+A（I，S）\*B（S，J）端

结束

中给出了面向列（或SAXPY）的版本

对于j=1:n

对于s=1:k

C（1:m，j）=C（1:m，j）+A（1:m，s）\*B（s，j）端

矩阵A由列访问，B由标量访问。此访问模式可以是

图示为

在另一个排列中，我们让s环是最外面的：

对于s=1:k

对于j=1:n

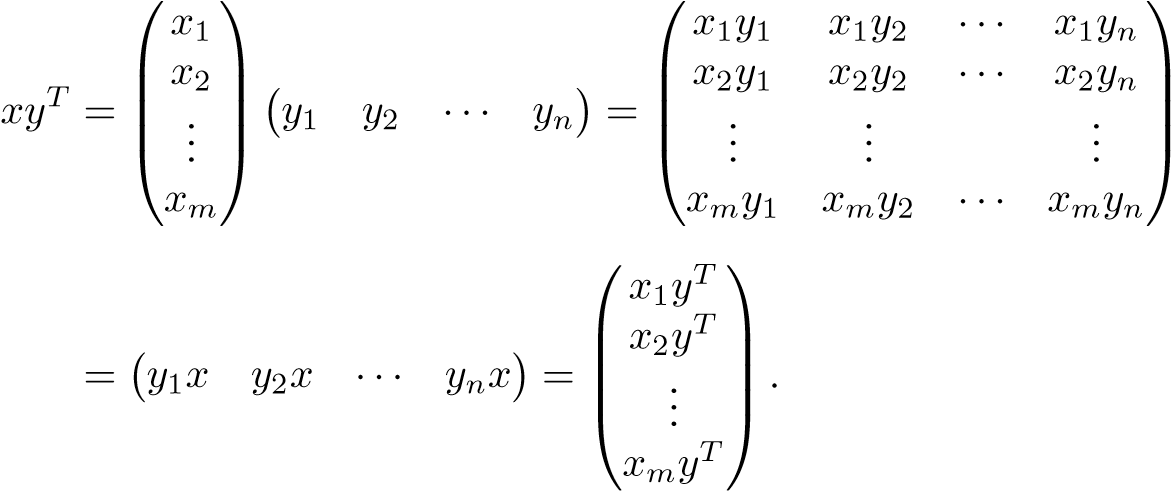
C（1:m，j）=C（1:m，j）+A（1:m，s）\*B（s，j）端

结束

这可以解释为如下。设a·k表示a的列向量，设b的行向量，则矩阵乘法可写成

（2.5）

这是矩阵乘法的外积形式。记住，外积遵循矩阵乘法的标准定义：让x和y分别是rm和rn中的列向量；然后



在外积形式（2.5）中写入矩阵c=ab可被视为简单矩阵中c的展开式。稍后我们将看到这样的矩阵的秩等于1。

2.3.内积和向量范数

2.3内积和向量范数

在本节中，我们将简要讨论如何测量向量的“大小”。最常见的向量规范是

# 1-标准，

，欧几里得范数（2-范数），最大范数。

欧几里得向量范数是r3到rn中标准欧几里得距离的推广。这里定义的三个规范都是p-norm的特殊情况：

.

与欧几里得向量范数相关联的是RN中两个向量x和y之间的内积，定义为

（x，y）=xty。

一般来说，向量范数是一个具有以下性质的映射rn→r

0代表所有X，

=0，如果且仅当x=0时，

，

三角不等式。

通过规范，我们可以引入向量近似中连续性和误差的概念。假设x是向量x的近似值。对于任何给定的向量范数，我们定义绝对误差。



相对误差（假设

.

在有限维向量空间中，所有向量范数在以下意义上都是等价的：

对于任意两个规范，存在常数m和m，使得

，（2.6）

其中m和m不依赖于x。例如，用x∈rn，

.

这种等价性意味着，如果一个向量序列（在一个范数中收敛到x，

，

然后它在所有规范中收敛到相同的极限。

在数据挖掘应用中，通常使用两个向量之间角度的余弦作为距离度量：

.

用这个方法，如果余弦接近一，两个向量就很接近。同样，如果x和y之间的夹角为π/2，即xty=0，则x和y是正交的。

2.4矩阵规范

对于任何向量范数，我们都可以定义相应的算子范数。设·为向量范数。相应的矩阵范数定义为

.

可以证明这样的矩阵范数满足（对于α∈R）

0代表所有A，

=0，如果且仅当a=0，

，

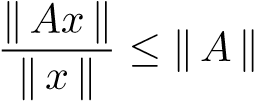
三角不等式。

对于上面定义的矩阵范数，以下基本不等式成立。

提案2.1.让·表示一个向量范数和相应的矩阵范数。然后

.

证据。根据我们的定义



2.4.矩阵规范

对于all=0，这给出了第一个不等式。第二个通过使用前两个for来证明。

我们可以证明2-范数满足

，

即矩阵ata最大特征值的平方根。因此，对于给定的矩阵（中等或大尺寸的矩阵），要获得的计算量相对较大。计算矩阵无穷范数（对于a∈rm×n）要容易得多。

，

矩阵1-范数

.

在第6.1节中，我们将看到矩阵的2-范数有一个关于a的奇异值的显式表达式。

设A∈Rm×N。在某些情况下，我们将把矩阵不当作一个线性算子，而是当作一个空间中的点，即，RmN。然后我们可以使用frobinius矩阵范数，它由

（2.7）

有时用等价形式写出弗罗贝尼乌斯范数是可行的。

，（2.8）

其中矩阵b的迹∈rn×n是其对角元素的和，

N.

i＝1

弗罗贝尼乌斯范数不对应于向量范数，因此在这个意义上它不是一个算子范数。这个范数的优点是比2范数更容易计算。弗罗贝尼乌斯矩阵范数实际上与欧几里得向量范数密切相关，即当矩阵用RMN中的元素标识时，它是矩阵Rm×N（线性空间）上的欧几里得向量范数。

2.5线性独立性：基础

给定一组向量（，考虑一组线性组合

跨距（跨度）

对于任意系数αj，向量（当

=0，如果且仅当αj=0，对于j=1,2，…，n。

Rm中的一组m线性无关向量称为Rm中的基：Rm中的任何向量都可以表示为基向量的线性组合。

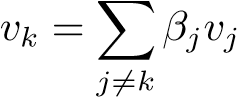
提案2.2.假设向量是线性相关的。然后一些vk可以写成其余的线性组合。证据。存在系数αj，其中一些=0，这样

.

取an=0并写

，

这和



βj=−αj/αk。

如果我们有一组线性相关向量，那么我们可以保持一个线性无关的子集，并用线性无关的子集来表示其余的部分。因此，我们可以将线性无关向量的个数作为集合信息内容的度量，并相应地压缩集合：以线性无关向量作为集合的代表（基向量），并根据基计算其余向量的坐标。然而，在实际应用中，我们很少有精确的线性相关向量，而是几乎是线性相关向量。结果表明，为了使这种数据约简过程实用且数值稳定，我们需要基向量不仅是线性无关的，而且是正交的。我们将在第4章中回到这一点。

2.6.矩阵的秩

2.6矩阵的秩

矩阵的秩定义为线性无关列向量的最大数目。线性无关列向量的个数等于线性无关行的个数是线性代数的标准结果。

向量。

稍后我们将看到，任何矩阵都可以表示为秩1的展开式。

矩阵。

提案2.3.外积矩阵xyt，其中x和y是rn中的向量，具有秩1。

证明。

.

因此，XYT的所有列（行）都是线性相关的。

一个秩为n的平方矩阵a∈rn×n称为非奇异矩阵，其逆a−1满足

a a−1=a−1a=i。

如果我们用一个非奇异矩阵乘以线性无关向量，那么这些向量就保持线性无关。

提案2.4.假设向量v1，…，vp是线性无关的。对于任何非奇异矩阵t，向量tv1，…，tvp是线性无关的。

证据。显然，如果且仅当=0时=0（因为我们可以用t或t−1将任何方程相乘）。因此，声明如下。

第三章

线性系统和最小

正方形

在本章中，我们简要回顾了关于线性方程组解的一些事实，

ax=b，（3.1）

式中a∈rn×n为正方形且非奇异。线性系统（3.1）可以用高斯消元和部分旋转来求解，这相当于将矩阵分解为三角形矩阵的乘积。

我们还将考虑超定线性系统，其中矩阵A∈Rm×N是M>N的矩形，并使用最小二乘法求解。因为我们只提供结果作为背景，所以我们大多是在没有证据的情况下陈述结果。有关求解线性方程组的矩阵分解理论的详细介绍，请参见，例如[42，92]。

在讨论矩阵分解之前，我们陈述了关于（3.1）唯一解存在条件的基本结果。

提案3.1.设a∈rn×n，假设a是非奇异的。然后对于任何右手边b，线性系统ax=b有一个唯一的解。

证据。结果是非奇异矩阵的列向量线性无关的直接结果。

3.1 LU分解

利用高斯变换可以方便地描述高斯消元，这些变换是高斯消元与LU分解等价的关键。关于高斯变换的更多细节可以在任何一本数字线性代数教科书中找到；参见，例如[42，p.94]。在部分旋转的高斯消元中，行的重新排序是通过

二十三

排列矩阵，是具有重新排序行的标识矩阵；参见，例如[42，第3.4.1节]。

考虑一个n×n矩阵a。在部分旋转高斯消元的第一步中，我们重新排列矩阵的行，以便将第一列中最大数量的元素移动到（1,1）位置。这相当于从左边乘以置换矩阵p1。然后用相乘的方法进行消除，即在对角线下方第一列中元素的归零。

A（1）：=L−1 1p1a，（3.2）

其中，l1是高斯变换

.

部分旋转高斯消元第一步的结果是

.

然后，高斯消除算法通过将第二列的元素归零（将最大的元素移动到对角线位置之后），依此类推。

从（3.2）我们可以看出，带有部分旋转的高斯消元的第一步可以表示为矩阵因子分解。整个过程也是如此。

定理3.2（lu分解）。任何非奇异的n×n矩阵a都可以分解为

pa=lu，

其中p是置换矩阵，l是下三角矩阵，主对角线上有一个，u是上三角矩阵。

证明（草图）。这个定理可以用归纳法证明。从（3.2）开始

p1a=l1a（1）。

定义（n-1）×（n-1）矩阵

γ

A（1）22…A（1）2N\_\_

B=…。

n2…安（1）a（1）

3.2.对称正定矩阵

通过归纳假设，B可以分解为

pbb=lbub，

然后我们看到，pa=lu，在哪里

U=11 2b，L=b b，P=p1，a在1 01 0

0 U P M1 L0铅

而at2=（a12 a13…A1N）。

计算LU分解的工作量大约是2n3/3次。在高斯消元的第k步中，一个操作在（n−k+1）×（n−k+1）子矩阵上，对于该子矩阵中的每个元素，执行一次乘法和一次加法。因此，总的失败次数

是

，

大约。

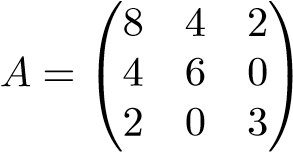
3.2对称正定矩阵

对称正定矩阵A的LU分解总是可以不用旋转来计算。此外，可以利用对称性，使分解也变得对称，并且需要的工作量是一般情况下的一半。

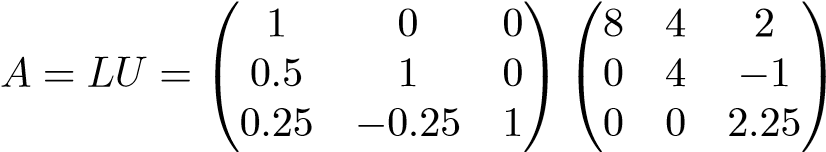
定理3.3（ldlt分解）。任何对称正定矩阵A都有分解

A=低密度脂蛋白，

其中，L是下三角形，主对角线上有一个，D是一个对角矩阵，带有正对角元素。例3.4.正定矩阵



有LU分解



以及LDLT分解

.

D中的对角线元素是正的，因此我们可以

，

然后我们得到

a=ldlt=（ld1/2）（d1/2lt）=utu，

其中u是上三角矩阵。这种LDLT分解的变体称为Cholesky分解。

由于a是对称的，所以只需要存储主对角线及其上的元素，即n（n+1）/2个矩阵元素。LDLT和Cholesky分解所需的存储量完全相同。也可以看出，由于只需要计算普通LU分解中一半的元素，因此工作量也减半，大约为n3/3次失败。当计算LDLT分解时，不需要首先计算LU分解，但是可以直接计算L和D中的元素。

3.3扰动理论和条件数

非奇异矩阵a的条件数定义为

，

其中·表示任何操作员规范。如果我们使用一个特殊的矩阵范数，例如2范数，那么我们就写

（3.3）

条件数用于量化矩阵和右侧受到少量扰动时，线性系统ax=b的解可以改变多少。

定理3.5。假设a是非奇异的

.

那么矩阵a+δa是非奇异的，并且

.

3.4.高斯消去中的舍入误差

扰动系统的解

（a+δa）y=b+δb

满足

.

例如，有关证明，请参见[42，定理2.7.2]或[50，定理7.2]

条件数较大的矩阵称为病态矩阵。定理3.5表明，具有病态矩阵的线性系统对数据中的扰动（即矩阵和右手边）敏感。

3.4高斯消去中的舍入误差

从第1.5.2节，关于浮点运算中的舍入误差，我们知道任何实数（在浮点系统中可表示）都用不超过单位舍入μ的相对误差表示。这个事实也可以说



当在浮点系统中表示矩阵A和向量B的元素时，会出现错误：



类似于b。因此，我们可以写

fl[a]=a+δa，fl[b]=b+δb，

哪里

.

目前，如果我们假设在系统ax=b的求解过程中没有进一步的舍入误差，我们可以看到x满足

这是反向误差分析的一个例子：计算出的解x是精确的

扰动问题的解。

利用微扰理论，我们可以从定理3.5中估计x的误差。

我们得到

（前提是

我们还可以分析高斯消元中的舍入误差对结果的影响。以下定理成立。（有关高斯消元的详细误差分析，请参阅[50，第9章]或[42，第3.3、3.4章]。）

定理3.6。假设我们使用单位舍入的浮点系统

是从高斯消去得到的三角形因子

部分旋转，应用于矩阵A。此外，假设x是用

前后替换：

L

那么x是系统的精确解。

哪里

，

k（n）是一个三次多项式，是在消除过程的步骤k-1中计算出的元素。

我们观察到gn依赖于高斯消去过程中矩阵元素的增长，而不是明确依赖于乘数的大小。可以计算gn，这样就可以得到舍入误差的后验估计。

先验（预先），我们可以证明gn≤2n−1，并且可以在元素增长如此严重的地方构建矩阵（注意，g31=230≈109）。然而，在实际中，高斯消元部分旋转时，gn很少大于8。

重要的是要注意，在高斯消去过程中，有些矩阵类没有元素增长，即gn=1，即使没有进行旋转。这是真的，例如，如果a是对称的和正定的。

在几乎所有的情况下，定理中的估计对于三次多项式k（n）都过于悲观。为了达到相等，所有舍入误差必须最大化，且其累积效应必须最大化不利。

3.5.带状矩阵

我们要强调的是，这种先验误差分析的主要目的不是给出线性系统解的误差估计，而是揭示算法的潜在不稳定性，并为比较不同算法提供依据。因此，定理3.6证明了高斯变换与我们将在第4章中介绍的正交变换相比的主要弱点：它们会导致矩阵元素的大增长，进而导致舍入误差。

3.5带状矩阵

在许多情况下，例如，常微分方程和偏微分方程的边值问题，当很大一部分元素等于零时，就会出现矩阵。如果非零元素集中在主对角线周围，则该矩阵称为带矩阵。更准确地说，如果存在自然数p和q，那么矩阵a被称为带矩阵。

如果j−i>p或i−j>q，aij=0。

例3.7.设q=2，p=1。设a为维度6的带矩阵：

.

w=q+p+1被称为矩阵的带宽。从这个例子来看，我们

请注意，w是a中任何一行中非零元素的最大数目。

存储带矩阵时，我们不存储带外的元素。同样，当线性方程组求解时，可以利用带结构来减少运算次数。

我们首先考虑P=Q=1的情况。这样的带矩阵称为三对角矩阵。

让

α1β1 2γ32α3 Nβ31 n n 1 Nγαβ2

A=…………

γαβ

−−−1

γαN

矩阵可以存储在三个向量中。在一个三对角系统ax=b的解中，很容易利用这个结构；我们首先假设a是对角占优的，因此不需要旋转。

%三对角矩阵的Lu分解。对于k=1:n-1

γ（k+1）=γ（k+1）/α（k）；α（k+1）=α（k+1）\*β（k）；

结束

%y=b.y（1）=b（1）解的正代换；k=2:n

y（k）=b（k）-γ（k）\*y（k-1）；

结束

%ux=y.x（n）=y（n）/alpha（n）解的反替换；k=n-1:-1:1

x（k）=（y（k）-β（k）\*x（k+1））/α（k）；

结束

运算（乘法和加法）的数目约为3N，除法的数目为2N。

在部分旋转高斯消元中，上三角矩阵的带宽增大。如果a的带宽w=q+p+1（主对角线下的q对角线和p上的p），那么在部分旋转的情况下，系数u的带宽wu=p+q+1。很容易看出，在L中不会创建新的非零元素。

带矩阵A的Lu分解中的因子L和U是带

矩阵。

### 例3.8。让4 2 T

2 5 5

A=25.2。

2 5 5

2 5

a具有cholesky分解a=u，其中2 1

2 1

U=2 1.

2 1

二

相反的是

，

密度很大。

结果表明，带矩阵的逆矩阵通常是稠密矩阵。因此，在大多数情况下，不应显式计算带矩阵的逆矩阵。

3.6最小二乘问题

在这一节中，我们将介绍最小二乘法和使用法方程的线性最小二乘问题的解决方案。基于正交变换的最小二乘问题的其他解决方法将在第5章和第6章中介绍。我们还将给出第6.6节中最小二乘问题的扰动结果。有关线性最小二乘问题的现代数值方法的广泛处理，请参见[14]。

例3.9.假设我们想通过附加不同的重量和测量弹簧的长度来确定弹簧的弹性特性。根据胡克定律，我们知道长度l取决于力f，根据

E+κF=L，

其中e和k是常数，有待确定。假设我们进行了一项实验，并获得了以下数据：

F 1 2 3 4 5

.

L 7.97 10.2 14.2 16.0 21.2

数据如图3.1所示。由于测量值存在误差，我们希望使用所有数据，以尽量减少误差的影响。因此，我们得到了一个比未知数据更多的系统，一个超定系统，

，

或者，以矩阵形式，

.

我们将用最小二乘法确定弹簧弹性常数的近似值。

+

+

+

+

+

5

10

15

20

*l*

1 2 3 4 5楼

图3.1.弹簧试验中的测量数据。

设a∈rm×n，m>n，系统

AX＝B

被称为过度确定：它有比未知更多的方程。一般来说，这样的系统没有解决方案。这可以通过让m=3和n=2从几何上看，也就是说，我们考虑了r3中的两个向量a·1和a·2。我们想找到向量的线性组合，这样

x1a·1+x2a·2=b。

在图3.2中，我们看到通常这样的问题没有解决方案。这两个矢量跨越一个平面，如果右侧B不在平面内，则A·1和A·2没有线性组合，因此x1a·1+x2a·2=B。

在这种情况下，“求解线性系统”的一个明显的替代方法是使向量r=b−x1a·1−x2a·2=b−ax尽可能小。b−ax称为剩余矢量，如图3.2所示。

问题的解决取决于我们如何测量剩余向量的长度。在最小二乘法中，我们使用标准欧几里得距离。因此，我们要找到一个向量x∈rn来解决最小化问题。

（3.4）

由于未知x在（3.4）中呈线性出现，这也被称为线性最小二乘问题。

在这个例子中，我们从我们在r3中的距离的知识中立即知道，如果我们选择平面中向量的线性组合，使得残差

*b*

*r*

*a*

·

1

*a*

·

2

图3.2.最小二乘问题，m=3，n=2。剩余向量b-ax是点。

矢量与平面正交。由于矩阵A的列跨越了平面，因此我们通过使r与a的列正交得到了解。这种几何直觉在一般情况下也是有效的：

rta·j=0，j=1,2，…，n.

（见第2.3节中正交性的定义）同样，我们可以

.

然后，使用r=b−ax，我们得到了正态方程（名称现在很明显）

ATAX=ATB

用于确定x中的系数。

定理3.10。如果A的列向量是线性无关的，则法向方程

ATAX=ATB

是非奇异的，有一个独特的解决方案。

证据。我们首先证明了ata是正定的。设x为任意非零矢量。然后，从线性独立的定义来看，我们得到=0。当y=ax时，我们有

，

相当于ata是正定的。因此，ata是非奇异的，

正态方程有一个唯一的解，我们表示所有r=b-a x的x。我们可以写信

然后，我们证明x是最小二乘问题的解，即：

和

.

因为=0，中间的两个项等于0，我们得到

Tr+（x−x）Tata（x−x）=r 22+a（x−x）22≥，

这是需要证明的。

例3.11。我们现在可以解决本章开头给出的示例。我们有

.

然后我们用matlab得到

>>C=A'\*A%法向方程

C=5.15

15 55

>>X=C\（A'\*B）

X=4.2360

三点二二六零

使用法向方程求解线性最小二乘问题有两个显著缺点：

1。形成ATA会导致信息丢失。2。条件数ata是a的平方：

κ（ata）=（κ（a））2.

我们用几个例子来说明这些要点。

例3.12。设为小，并定义矩阵

.

接下来是

.

如果非常小，1+2的浮点表示满足]=1，则在浮点运算中，法方程变为奇异方程。因此，在形成ATA时会丢失A中存在的重要信息。

使用a的奇异值分解定义了矩形矩阵a的条件数。我们将在第6.6节中陈述关于最小二乘问题条件的结果。

例3.13。我们使用matlab计算示例3.9中矩阵的条件数：

A=1 1 2

1 3

1 4

1 5

康德（A）=8.3657康德（A'\*A）=69.9857

然后我们假设我们有一个线性模型

L（x）=c0+c1x

对于数据向量x=（101102103104105）t，这给出了一个大条件数的数据矩阵：

A=1 101 1 102

1 103

1 104

1 105

Cond（A）=7.5038E+03 Cond（A'\*A）=5.6307E+07如果我们使用该模型

L（x）=b0+b1（x−103）

相应的法向方程变成对角线，并且条件更好（证明这一点）。

经常会出现一个矩阵相同的最小二乘问题序列，



有解决方案

X=（ATA）-1ATBI，I＝1，2，…，P。

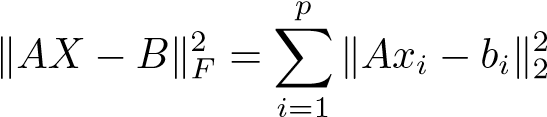
定义，我们可以用矩阵形式写

（3.5）

有了这个解决方案

x=（ata）−1安培。

这源于身份



（3.5）中的p子问题是独立的。

第四章

正交性

即使求解线性方程组和正态方程组的高斯消去法是一种广泛应用于许多应用中的标准算法，但在需要将最重要的信息与不重要的信息分开的情况下，这种方法仍然不够。（噪音）。“数据质量”的典型线性代数公式是对“好基向量和坏基向量”的概念进行量化；不严格地说，好基向量是那些“非常线性无关”的向量，即接近正交的向量。在同样的情况下，几乎线性相关的向量是坏基向量。在这一章中，我们将介绍一些理论和正交向量计算算法。第6章给出了一组向量“质量”的更完整的量化。

例4.1.在示例3.13中，我们看到，在最小二乘问题中选择不合适的基向量会导致病态正态方程。沿着相似的线，定义两个矩阵

，

其列如图4.1所示。可以看出，这两个矩阵的列向量在r3中跨同一平面。从图中可以清楚地看出，B的柱是正交的，它比A的柱确定平面要好得多，A的柱是非常接近的。

从多个角度来看，在向量空间中使用正交向量作为基向量是有利的。在本章中，我们将列出正交向量集和正交矩阵的一些重要性质。我们假设向量在rm中，m≥n。

三十七

0

0.2

0.4

0.6

0.8

1

1.2

1.4

0

0.2

0.4

0.6

0.8

1

−1

−0.5

0

0.5

1

图4.1.在r3中跨越一个平面的三个矢量。

4.1正交向量和矩阵

我们首先回顾两个非零向量x和y，如果xty=0（即cosθ（x，y）=0，则称为正交向量。

提案4.2.设qj，j=1,2，…，n，为正交，即。然后它们是线性无关的。

证据。假设它们是线性相关的。那么，从命题2.2，存在一个qk，这样

.

用这个方程乘以qkt，我们得到

，

因为向量是正交的。这是一个矛盾。

使Rm中的正交向量组qj，j=1,2，…，m归一化，

.

然后它们被称为正交，在rm中它们构成正交基。

方阵

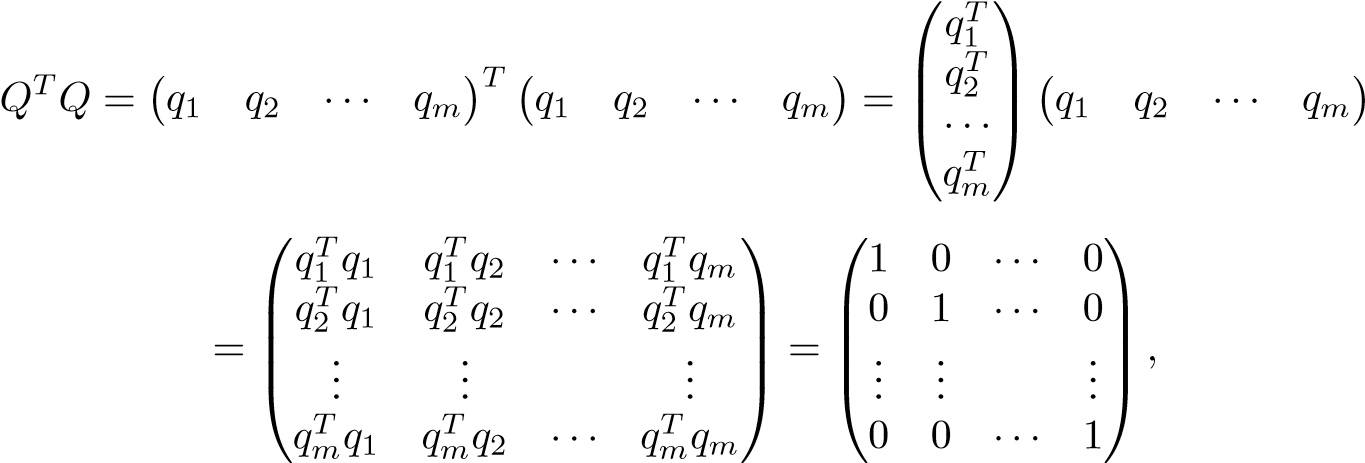


4.1.正交向量和矩阵

其列是正交的称为正交矩阵。正交矩阵满足我们在一系列命题中列出的一些重要性质。

提案4.3.正交矩阵q满足qtq=i。

证明。



由于正态性。

其列的正交性意味着正交矩阵具有满秩，求逆矩阵的难度很小。

提案4.4.正交矩阵q∈rm×m具有秩m，且由于qtq=i，其逆矩阵等于q−1=qt。

提案4.5.正交矩阵的行是正交的，即qqt=i。

证据。设x为任意向量。我们将证明qqtx=x。给定x有一个唯一确定的向量y，这样qy=x，因为q-1存在。然后

qqtx=qqtqy=qy=x。

因为x是任意的，所以qqt=i。

提案4.6.两个正交矩阵的乘积是正交的。

证据。设q和p为正交，x=pq。然后

xtx=（pq）tpq=qtppq=qtq=i。

任意Rm子空间的正交基都可以扩大到整个空间的正交基。下一个命题用矩阵表示。

提案4.7.给定一个矩阵q1∈rm×k，有正交列，存在一个矩阵q2∈rm×k（m−k），使得q=（q1q2）是一个正交矩阵。

这个命题是线性代数的标准结果。稍后我们将演示如何计算q。

正交矩阵的一个最重要的性质是保持向量的长度。

提案4.8.向量的欧氏长度在正交变换q下是不变的。

证据。.

相应的矩阵范数和弗罗贝尼乌斯范数在正交变换下是不变的。

提案4.9.设u∈rm×m与v∈rn×n为正交。那么对于任何

A∈Rm×N，

.

证据。第一个等式很容易用命题4.8证明。第二个是用弗罗贝尼乌斯范数的替代表达式（2.8）和同一性Tr（bc）=Tr（cb）证明的。

4.2初等正交矩阵

我们将使用初等正交矩阵将矩阵简化为紧致形式。例如，我们将矩阵A∈Rm×N，m≥N转换成三角形。

### 4.2.1平面旋转

2×2平面旋转矩阵

，

是正交的。向量x乘以g，以顺时针方向将向量旋转一个角度θ，其中c=cosθ。平面旋转可通过选择以下选项使向量x的第二个元素归零：

.

通过在较大的单位矩阵中嵌入二维旋转，可以操纵任意维的向量和矩阵。

4.2.初等正交矩阵

例4.10。我们可以选择C和S

\_\_1 0 0 0\_\_

0摄氏度0秒

g=

0 0 1 0

0−S 0 C

在matlab脚本中，我们通过平面（2，4）中的旋转使向量x∈r4中的元素4归零。执行

x=[1；2；3；4]；sq=sqrt（x（2）^2+x（4）^2）；c=x（2）/sq；s=x（4）/sq；

g=[1 0 0 0；0 c 0 s；0 0 1 0；0-s 0 c]；y=g\*x给出结果

Y=1.0000

四点四七二一

三

零

使用平面旋转序列，我们现在可以将任意向量转换为单位向量的倍数。这可以通过几种方式实现。我们在下面的示例中演示一个。

例4.11.g3在平面（3given a vector，4）中，我们将最后一个元素x∈r4归零，我们将其转换为：κe1。首先，通过旋转

.

然后，通过平面（2,3）中的旋转g2，我们将位置3中的元素归零：

.

最后，第二个元素被旋转g1湮灭：

.

根据命题4.8，欧几里得长度被保留，因此我们知道。

我们总结这些转变。我们有

κe1=g1（g2（g3x））=（g1g2g3）x。

因为正交矩阵的乘积是正交的（命题4.6），所以矩阵p=g1g2g3是正交的，总的结果是px=κe1。

平面旋转非常灵活，可以有效地用于稀疏结构的问题，例如带矩阵。另一方面，对于密集矩阵，它们需要比户主转换更多的触发器；见第4.3节。

例4.12。在本节前面的matlab示例中，我们显式地将2×2嵌入到更大尺寸的矩阵中。这是浪费操作，因为计算机执行代码时没有考虑到矩阵只有两行被更改的事实。相反，执行整个矩阵乘法，在维度n矩阵的情况下需要2n3个触发器。下面两个matlab函数说明了如何执行旋转以保存操作（和存储）：

函数[c，s]=rot（x，y）；

%构造一个平面旋转，使向量[x；y]中的第二个%分量归零（x和y是标量）sq=sqrt（x^2+y^2）；c=x/sq；s=y/sq；

函数x=近似值（c，s，i，j，x）；

%在平面（i，j）中应用平面（平面）旋转

%到矩阵X

x（[i，j]，：）=[c s；-s c]\*x（[i，j]，：）；

下面的脚本将向量x减少到标准基向量e1的倍数：

x=[1；2；3；4]；对于i=3:-1:1

[c，s]=rot（x（i），x（i+1））；x=近似值（c，s，i，i+1，x）；

结束

>>X=5.4772

零

零

零

减去后，x的第一个分量等于。

4.2.初等正交矩阵

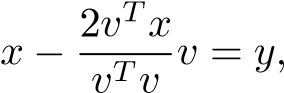
### 4.2.2户主改造

设=0为任意向量，并将

；

P是对称和正交的（通过简单的计算来验证这一点！）这种矩阵称为反射矩阵或户主变换。让x和y被赋予相同长度的向量，然后问问题，“我们能确定

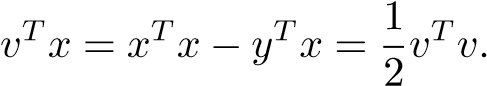
一个户主转换p，使得px=y？“公式px=y可以写成



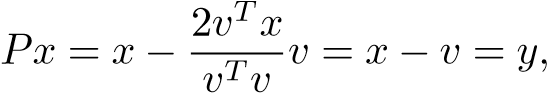
它的形式是βv=x-y。由于v以β因子抵消的方式进入p，我们可以选择β=1。当v=x-y时，我们得到

vtv=xtx+yty−2xty=2（xtx−xty）

因为XTX=Yty。此外，



所以我们有

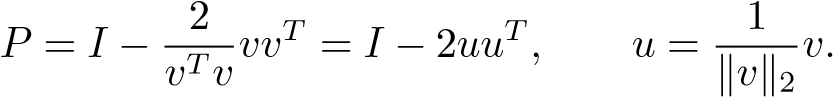


如我们所愿。在矩阵计算中，我们通常希望向量中的元素为零，现在我们选择y=κe1，其中，和。这个

向量v应等于

V=x−κe1。

为了避免取消（即两个近浮点数的减法），我们选择符号（k）=-符号（x1）。现在我们已经计算了v，我们可以简化并编写



因此，户主向量u的长度为1。户主向量的计算可在以下matlab代码中实现：

函数u=househ（x）

%计算户主向量u，使

%（i-2 u\*u’）x=k\*e\_1，其中

%| k等于x%的欧几里得范数，e\_1是第一个单位向量n=长度（x）；%x中的分量数

kap=范数（x）；v=零（n，1）；v（1）=x（1）+符号（x（1））\*kap；v（2:n）=x（2:n）；u=（1/范数（v））\*v；

在大多数情况下，应避免明确形成户主矩阵P，因为它可以更紧凑地由向量U表示。应根据p x=x−（2utx）U进行p乘，其中矩阵向量乘需要4N个触发器（如果p是形式，则不是o（n2））。D明确）。矩阵乘法px完成

Px=A−2U（UTX）。（4.1）

通过户主转换的乘法在以下代码中实现：

函数y=apphouse（u，x）；

%将矩阵x乘以户主矩阵

%y=（i-2\*u\*u'）\*x

y=x-2\*u\*（u'\*x）；

例4.13。矢量的前三个元素由以下matlab语句序列归零：

>>X=[1；2；3；4]；>>U=房屋H（X）；

>>Y=应用程序库（U，X）

Y=-5.4772 0

零

零

由于平面旋转可以嵌入到单元矩阵中，为了结构化地应用转换，我们同样可以嵌入户主转换。例如，假设我们已经将矩阵中的第一列转换为单位向量，然后我们希望将主对角线下面第二列中的所有元素归零。因此，在一个5×4矩阵的例子中，我们要计算转换

4.3.浮点运算数

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ×××。  2a（1）=p2\_\_\_×000×××\_\_\_\_\_\_×00  P××X=0  ×××。  0××0  对（1）的第二列进行如下分区： | ×  ×0  零  零 | ×  ×  ×  ×  × | ×\_\_=：A  ×（2）  ××。  ×  × | （4.2） |

，

其中是一个标量。我们知道如何转换为单位向量；让p\_2成为一个这样做的户主转换。然后，通过将p\_2嵌入到单位矩阵中来实现转换（4.2）：

（4.3）

很明显，p2使a（1）的第一行保持不变，并计算转换（4.2）。另外，很容易看到第一列中新创建的零没有被销毁。

与平面旋转的情况类似，不应将户主变换显式嵌入更大维度的单位矩阵中。相反，应该将它应用于在转换过程中受影响的行（在从左边进行乘法的情况下）。

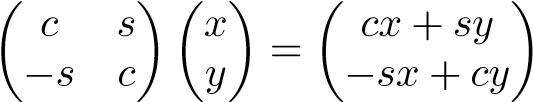
例4.14.（4.2）中的转换通过以下语句完成。

u=房屋h（a（2:m，2））；a（2:m，2:n）=apphouse（u，a（2:m，2:n））；

|  |  |  |  |
| --- | --- | --- | --- |
| >>A=-0.8992 | -0.6708个 | -0.7788个 | -0.9400元 |
| -0.0000元 | 零点三二九九 | 零点七四零零 | 零点三八九一 |
| -0.0000元 | 零 | -0.1422个 | -0.6159个 |
| -0.0000元 | -0.0000元 | 零点七五七六 | 零点一六三二 |
| -0.0000元 | -0.0000元 | 零点三零五三 | 零点四六八零 |

4.3浮点运算次数

我们将比较使用平面变换和户主变换将m×n矩阵a的第一列转换为单位向量k e1的倍数的触发器的数目。考虑第一个平面旋转。显然，计算



需要四个乘法和两个加法，即六个浮点运算。将这种转换应用于m×n矩阵需要6n个触发器。为了使矩阵第一列中除一个元素以外的所有元素归零，我们应用m-1旋转。因此，总故障计数为6（m−1）n≈6mn。

如果按照（4.1）中的方式通过户主转换执行相应的操作，那么只需要4mn的触发器。（请注意，乘2不是一个触发器，因为它是由编译器作为移位实现的，这比

触发器；或者，可以用√2缩放向量u。）

4.4浮点正交变换

算术

正交变换在浮点运算中非常稳定。例如，可以证明[50，第367页]浮点数p\_中计算得出的近似于精确p的户主转换满足

，

其中，μ是浮点系统的四舍五入单位。我们还有向后误差的结果

.

因此，浮点结果等于精确正交矩阵和数据矩阵的乘积，数据矩阵受到非常小的扰动。类似的结果适用于平面旋转。

第五章

QR分解

这本书的一个主要主题是通过正交变换将矩阵分解成紧凑的形式（例如三角形或对角线）。我们现在将介绍第一个这样的分解，qr分解，它是矩阵a在正交矩阵和三角形矩阵的乘积中的分解。这比计算LU分解更雄心勃勃，因为这两个因素都只需要是三角形的。

5.1正交变换为三角形

通过一系列的户主变换，我们可以变换任意矩阵a∈rm×n，m≥n，

，

其中r为上三角，q∈r m×m为正交。使用小尺寸矩阵可以方便地说明该过程。设A∈R5×4。在第一步中，我们将第一列中主对角线以下的元素归零，

，

其中，+表示在转换中发生变化的元素。正交矩阵h1等于户主变换。在第二步中，我们使用嵌入的户主转换，如（4.3）所示，将元素归零。

四十七

在第二列主对角线下方：

.

同样，在右侧，+'表示在转换过程中已更改的元素，+'表示在当前转换过程中未更改的元素。

在第三步中，我们将第三列对角线下的元素湮灭。-

Un:

.

在第四步之后，我们计算了上三角矩阵r，总结了变换序列。

.

注意，矩阵hi具有以下结构（这里假设a∈rm×n）：

（5.1）

，

因此，我们在身份矩阵中嵌入了连续较小维的户主变换，并且在每个步骤中向量ui都变短了。很容易看出矩阵hi也是户主变换。例如，

.

三角形式的变换等价于矩阵A的分解。

5.1正交变换为三角形

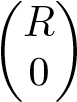
定理5.1（qr分解）。任何矩阵a∈rm×n，m≥n都可以用正交矩阵变换成上三角形式。这个变换等价于一个分解

，

其中q∈r m×m为正交，r∈r n×n为上三角。如果a的列是线性无关的，那么r是非奇异的。

证据。前面例子中概述的构造过程可以很容易地适应一般情况，前提是，如果要转换为单位矢量的矢量是零矢量，那么正交变换的选择等于恒等式。

柱的线性独立性



根据提案2.4。由于r是上三角形，线性独立性意味着它的对角元素不是零。（如果一列的对角线上有一个零，那么它将是它左边那些的线性组合。）因此，r的行列式是非零的，这意味着r是非奇异的。

我们在图5.1中象征性地说明了QR分解。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | *A* | = | *Q* | | 0  *R* |

m×n m×m×n

图5.1.QR分解的符号说明。

通常，以另一种方式编写分解是很方便的，其中保留的q的唯一部分对应于a列的正交化；见图5.2。

这种薄的qr分解可以通过划分q=（q1 q2），其中q1∈rm×n，并注意在乘法中，块q2乘以零：

（5.2）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | *A* | = | *Q*1 | | *R* |

m×n m×n×n

图5.2.薄qr分解a=q1r。

从这个方程可以看出，r（a）=r（q1）；因此，我们现在已经计算了距离空间r（a）的正交基。此外，如果我们在（5.2）中写出j列，

，

我们看到R中的J列在正交基础上保持AJ的坐标。

例5.2。我们在matlab中对qr分解的计算进行了简单的数值说明：

A=1 1 1 2 4

1 3 3

1 4 4

>>[Q，R]=QR（A）

|  |  |  |
| --- | --- | --- |
| Q=-0.5000 | 0.6708 0.5000 | 零点二二三六 |
| -0.5000元 | 0.2236-0.5000 | -0.6708个 |
| -0.5000元 | -0.2236-0.5000 | 零点六七零八 |
| -0.5000元 | -0.6708 0.5000 | -0.2236个 |
| R=-2.0000 | -5.0000至15.0000 |  |
| 零 | -2.2361-11.1803年 |  |
| 零 | 0.2万 |  |
| 零 | 0 0 |  |

通过命令qr（a，0）获得薄qr分解：

>>[Q，R]=QR（A，0）

1. =-0.5000 0.6708 0.5000-0.5000 0.2236-0.5000-0.5000-0.2236-0.5000

-0.5000-0.6708 0.5000

5.2.求解最小二乘问题

1. =-2.0000-5.0000-15.0000 0-2.2361-11.1803

0 0 2.0000

5.2求解最小二乘问题

利用QR分解，我们可以解决最小二乘问题。

，（5.3）

式中a∈rm×n，m≥n，不形成正规方程。为此，我们利用欧几里得向量范数在正交变换下不变的事实（命题4.8）：

.

在残差向量中引入A的二维分解，得到

.

然后我们对q=（q1 q2），其中q1∈rm×n，并表示

.

现在我们可以写信了

（5.4）

假设A的列是线性无关的，我们可以求解

Rx=b1

使（5.4）中的第一项等于零，从而最小化。我们现在已经证明了以下定理。

定理5.3（QR分解的最小二乘解）。假设矩阵A∈Rm×N具有全列秩和薄qr分解a=q1r，则最小二乘问题具有唯一解。



例5.4.作为一个例子，我们从第3.6节开始解决最小二乘问题。矩阵和右手边是

|  |  |
| --- | --- |
| A＝1 | 1 B=7.9700 |
| 一 | 2000年10月2日 |
| 一 | 3.14.2000年 |
| 一 | 4 16.0000元 |
| 一 | 2000年5月21日 |

具有二维分解和最小二乘解

>>[q1，r]=qr（a，0）%薄qr

|  |  |
| --- | --- |
| 问题1=-0.4472 | -0.6325个 |
| -0.4472个 | -0.3162个 |
| -0.4472个 | 零 |
| -0.4472个 | 零点三一六二 |
| -0.4472个 | 零点六三二五 |
| R=-2.2361 | -6.7082个 |
| 零 | 三点一六二三 |

>>x=r\（q1'\*b）

X=4.2360

三点二二六零

注意，matlab语句x=a\b使用完全相同的算法给出了相同的结果。

5.3计算与否Q

通过将变换应用于单位矩阵，可以同时计算正交矩阵q和r。同样，薄qr分解中的q1可以通过将变换应用于部分恒等矩阵来计算。

.

然而，在许多情况下，我们不需要明确的Q。相反，应用相同的户主转换序列可能就足够了。由于（5.1）中所示的嵌入式户主矩阵的结构，用于构造户主转换的向量（将矩阵A简化为上三角形式）可以存储在主对角线下的a中，位置等于零。然后需要一个额外的向量来存储r对角线上的元素。

在求解最小二乘问题（5.3）时，根本不需要计算q。通过将右侧连接到矩阵，我们计算

，

5.4.QR因子分解的失败计数

通过求解得到最小二乘解。我们也看到了

.

因此，最优残差范数是作为三角化过程的副产品得到的。

5.4 QR因子分解的失败计数

如第4.3节所示，将户主转换应用于m×n矩阵，使对角线下第一列的元素归零，需要大约4mn的浮点运算。在下面的转换中，只有行2到m和列2到n发生了更改（请参见（5.1）），并且受转换影响的子矩阵的维数在每个步骤中都减少了一个。因此，用于计算r的触发器的数量大约是

.

然后矩阵Q以因子形式提供，作为户主转换的产物。如果我们显式地计算全矩阵q，那么在步骤k+1中我们需要

4（m−k）m触发器，总共导致

.

可以利用q累积中的结构来稍微减少触发器计数[42，第5.1.6节]。

5.5最小二乘问题解的误差

如第4.4节所述，户主转换和平面旋转在浮点舍入误差方面具有良好的特性。这里我们给出一个定理，其证明可在[50，定理19.3]中找到。

定理5.5。假设a∈rm×n，m≥n具有全列秩，并通过户主变换用qr因子分解法求解最小二乘问题。那么计算出的解x\_是

，

哪里

，

c1和c2是小常数。

由此可见，从向后误差的意义上来说，该解是尽可能好的（即，该方法是向后稳定的）。利用第6.6节中的摄动理论，可以估计计算解中的正向误差。在第3.6节中，我们建议用于求解最小二乘问题的法方程法在浮点运算中具有不太令人满意的特性。结果表明，除非矩阵A条件良好，否则该方法是不可逆稳定的。这两种方法的优缺点在[50，p.399]中得到了很好的总结。这里我们给出一个例子，尽管有些极端，但证明对于某些最小二乘问题，用正态方程法给出的解比用qr分解得到的解要精确得多；参见例3.12。

例5.6。让我们来考虑矩阵

.

A的条件编号为107号命令。以下matlab脚本

x=[1；1]；b=a\*x；

xq=a\b；%qr分解xn=（a'\*a）\（a'\*b）；%正态方程

[XQ XN]号

给出了结果

1.0000000000000 1.01123595505618

1.0000000000000 0.98876404494382，说明正态方程法存在矩阵ata的条件数是a的条件数的平方这一事实。

5.6更新最小二乘法的解

问题

在某些应用中，A的行和B的相应元素是实时测量的。让我们把B的元素称为观察。每次观测到，都要计算一个新的最小二乘解。如果我们从头开始重新计算这个解决方案，每一次新的观察都会花费O（mn2）触发器。这在大多数情况下都太昂贵了，而且计算量也不必要太大，因为每次有新的观测结果时，可以通过更新o（n2）触发器中的qr分解来计算最小二乘解。此外，更新算法不要求我们保存正交矩阵Q！

假设我们已经减少了矩阵和右手边

，（5.5）

5.6.更新最小二乘问题的解决方案

从中可以很容易地得到最小二乘解。假设我们没有

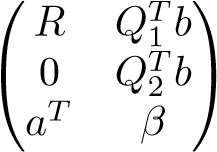
标量。然后，我们要找到增广最小二乘问题的解，然后用一个新的观察值来表示（在β），其中a∈rn和β是a。

（5.6）

对于新的矩阵，我们可以将约简式（5.5）写成

.

因此，如果我们减少，我们可以找到增广最小二乘问题（5.6）的解。



通过一系列正交变换形成三角形。矢量在这个约简中不起作用，因此我们从推导中排除它。

现在我们将演示如何使用平面旋转序列将其还原为三角形。算法的思想将用一个n=4的小例子来说明。我们从

.

通过在（1，n+1）平面上的旋转，我们将底行向量的第一个元素归零。结果是

++++++×+×+×+

×××，

××

0××

其中，+表示在当前转换中已更改的元素。然后，底部向量的第二个元素通过（2，n+1）中的旋转归零：

\_

×××。

××

0××

请注意，上一步中引入的零不会被销毁。经过三个类似的步骤，最终结果是：

.

总共需要n个旋转来计算减少量。（5.6）的最小二乘解现在通过求解rx∙∙b1得到。

第六章

奇异值分解

即使QR分解对于解决最小二乘问题非常有用，并且具有良好的稳定性，它也有一个缺点，即它对矩阵的行和列进行不同的处理：它只为列空间提供基础。奇异值分解（SVD）以对称的方式处理行和列，因此它提供了关于矩阵的更多信息。它还对矩阵中包含的信息进行“排序”，这样，不严格地说，“支配部分”就变得可见。这使得SVD在数据挖掘和许多其他领域非常有用。

6.1分解

定理6.1（SVD）。任何m×n矩阵a，m≥n，都可以分解。

，（6.1）

其中u∈rm×m和v∈rn×n是正交的，∑rn×n是对角的，

∑=diag（σ1，σ2，…，σn）、

σ1≥σ2≥··········≥σn≥0。

证据。假设m≥n不受限制：在另一种情况下，只需将定理应用于at。我们在[42]中给出了一个证明。考虑最大化问题

.

由于我们在闭集上求连续函数的上确界，因此对某些向量x求上确界。将ax=σ1y，其中=1和

五十七

（根据定义）。利用命题4.7，我们可以构造正交

矩阵

z1=（y z／2）∈rm×m，w1=（xw／2）∈rn×n。

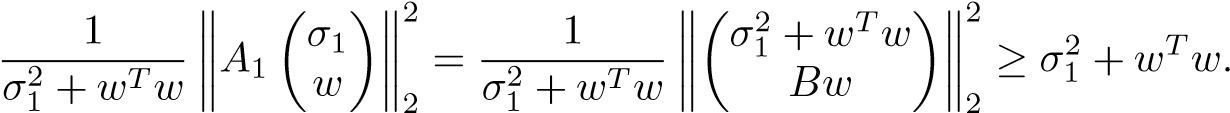
然后

，

因为Ytax=σ1，并且=0。放

.

然后



但是，因此w=0必须保持不变。因此，我们朝着a的对角化迈出了一步。证明现在是通过归纳法完成的。假设

=diag（σ2，…，σn）。

然后我们有了

.

因此，通过定义

，

证明了这个定理。

u和v列称为奇异向量，对角元素σi为奇异值。

我们在这一点上强调，这不仅是一个重要的理论结果，而且还有非常有效和精确的计算SVD的算法；见第6.8节。

SVD以不同的名称出现在其他科学领域。在统计和数据分析中，奇异向量与主分量密切相关（见第6.4节），在图像处理中，SVD的名称为karhunen–

Loewe扩展。

我们用符号说明了SVD：

6.1.分解

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | *A* | = | *U* | | 0  0 | |  | | --- | | *V T* | |

m×n m×m×n

用分区u=（u1 u2），其中u1∈rm×n，我们得到了薄的svd，

A=U1∑V t，

象征性地说明，

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | *A* | = | *U* | | 0  0 | |  | | --- | | *V T* | |

m×n m×n×n

如果我们写出矩阵方程

av=u1∑，atu1=v∑

一列接一列，我们得到等效方程

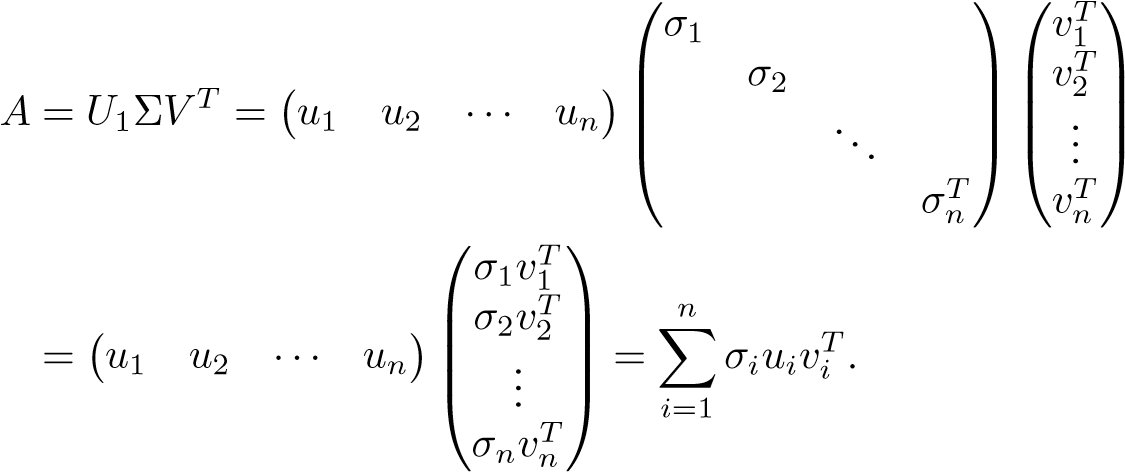
avi=σiui，atui=σivi，i=1,2，…，n.

SVD也可以写成矩阵的扩展：

（6.2）

这通常被称为外部产品形式，它是从

精简版：



SVD的外部产品形式如图所示：

++···我是说。

例6.2。我们计算全列秩矩阵的SVD：

A=1 1

1 2

1 3

1 4

>>[U，S，V]=SVD（A）

|  |  |  |
| --- | --- | --- |
| U=0.2195-0.8073 | 零点零二三六 | 零点五四七二 |
| 0.3833至0.3912 | -0.4393个 | -0.7120个 |
| 0.5472 0.0249 | 零点八零七九 | -0.2176个 |
| 0.7110 0.4410  S=5.7794 0  0 0.7738  0 0  0 0  V=0.3220-0.9467  0.9467 0.3220  SVD的精简版本为>>[U，S，V]=SVD（A，0）  U=0.2195-0.8073  0.3833至0.3912  0.5472 0.0249  0.7110 0.4410  S=5.7794 0  0.7738 V=0.3220-0.9467 | -0.3921个 | 零点三八二四 |
| 0.9467 0.3220 |  |  |

第2.4节定义了矩阵2-范数。从定理6.1的证明我们已经知道了。这是一个非常重要的事实，值得单独提出一个建议。

6.2.基本子空间

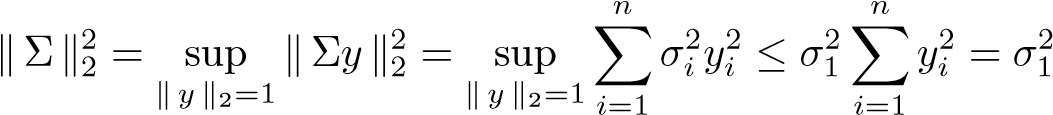
提案6.3.矩阵的2-范数由

.

证据。以下是另一种证明。在不失去一般性的前提下，假设a∈rm×n，m≥n，并使a的svd为a=u∑v t，在正交变换下范数是不变的，因此

.

现在的结果如下，因为对角矩阵的2-范数等于最大对角元素的绝对值：



y=e1相等。

6.2基本子空间

SVD给出了矩阵的四个基本子空间的正交基。矩阵A的范围是线性子空间。

r（a）=y y=a x，任意x。

假设a的等级为r：

σ1≥·····································

然后，使用外部产品形式，我们有

.

矩阵A的零空间是线性子空间。

n（a）=x a x=0。

因为，我们看到任何向量都在空值中-

## 空间：

.

经过类似的论证，我们得到了下面的定理。

定理6.4（基本子空间）。

1. 奇异向量u1，u2，…，ur是r（a）和

等级（a）=dim（r（a））=r。

1. 奇异向量vr+1，vr+2，…，vn是n（a）和

尺寸（n（a））=n−r。

1. 奇异向量v1，v2，…，vr是r（at）中的正交基。
2. 奇异向量ur+1，ur+2，…，um是n（at）中的正交基。

例6.5。我们通过在前面的例子中构造第三列作为列1和列2的线性组合来创建一个缺秩矩阵：

>>A（：，3）=A（：，1）+0.5\*A（：，2）

|  |  |  |
| --- | --- | --- |
| A=1.0000 1.0000 | | 一点五零零零 |
| 1.0000 2.0000 | | 二 |
| 1.0000 3.0000 | | 二点五零零零 |
| 1.0000 4.0000  >>[U，S，V]=SVD（A，0） | | 三 |
| U=0.2612 | -0.7948个 | -0.5000元 |
| 零点四零三二 | -0.3708个 | 零点八三三三 |
| 零点五四五一 | 零点零五三三 | -0.1667个 |
| 零点六八七一 | 零点四七七四 | -0.1667个 |
| S=7.3944 | 零 | 零 |
| 零 | 零点九零七二 | 零 |
| 零 | 零 | 零 |
| V=0.2565 | -0.6998 | 零点六六六七 |
| 零点七三七二 | 零点五八七七 | 零点三三三三 |
| 零点六二五一 | -0.4060个 | -0.6667个 |

第三奇异值等于零，矩阵是秩亏矩阵。显然，v的第三列是n（a）中的基向量：

>>A\*V（：，3）

ANS＝

1.0E-15型\*

零

-0.2220个

-0.2220个

零

6.3.矩阵近似

0

2

4

6

8

10

12

14

16

18

20

10

−1

10

0

10

1

singular values

index

图6.1.秩10加噪声的矩阵的奇异值。

6.3矩阵近似

假设a是一个低阶矩阵加上噪声：a=a0+n，其中噪声n比a0小。通常情况下，a的奇异值的行为如图6.1所示。在这种情况下，如果噪声的大小足够小，则大的奇异值的数量通常被称为矩阵的数值等级。如果我们知道a0的正确秩，或者可以估计它，例如通过检查奇异值，那么我们可以通过用正确秩的矩阵近似a来“去除噪声”。最明显的方法就是简单地截断奇异值展开（6.2）。假设数值等级等于k，则我们近似

.

截短的SVD非常重要，不仅用于消除噪声，还用于压缩数据（见第11章），以及稳定极端病态问题的解决方案。

结果表明，截断SVD是一个近似问题的解，即人们希望用一个较低的秩来近似一个给定的矩阵。我们将在两个规范中考虑矩阵A的低阶近似。首先给出了矩阵2-范数的定理。

定理6.6.假设矩阵A∈Rm×N具有秩r>k，矩阵逼近问题

最小Z级（

有解决方案

Z=ak：=uk∑kvkt，

其中，uk=（u1，…，uk），vk=（v1，…，vk）和∑k=diag（σ1，…，σk）。最小值是

.

这一定理的证明可以在[42，第2.5.5节]中找到。接下来回顾弗罗贝尼乌斯矩阵范数的定义（2.7）

.

结果表明，这种情况下的近似结果是相同的。

定理6.7。假设矩阵A∈Rm×N具有秩r>k，frobenius范数矩阵逼近问题

最低F级（

有解决方案

，

其中，uk=（u1，…，uk），vk=（v1，…，vk）和∑k=diag（σ1，…，σk）。最小值是

，

其中p=最小值（m，n）。

为了证明这个定理，我们需要一个引理。

引理6.8。考虑具有内积的mn维向量空间Rm×n

（6.3）

和标准

.

6.3.矩阵近似

设a∈rm×n，SVD a=u∑v t，然后求矩阵

uivjt，i=1,2，…，m，j=1,2，…，n，（6.4）

是Rm×N中的正态基。

证据。使用身份）我们得到

，

这表明矩阵是正交的。因为有mn这样的矩阵，它们以rm×n构成一个基。

证明（定理6.7）。这一证据是基于[41]中的证据。根据基（6.4）写出矩阵Z∈Rm×N，

，

选择系数的位置。在这个证明中，我们用σij表示∑的元素。由于基的正态性，我们有

.

显然，我们可以选择第二项为等于零。然后，z的表达式如下：

.

因为z的秩等于这个和中的项数，我们看到约束秩（z）=k意味着在和中应该正好有k个非零项。为了最小化目标函数，我们选择

ζi i=σii，i=1,2，…，k，

这就产生了预期的结果。

矩阵的低阶近似表示为

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | *A* | ≈ |  | |  |

=英国∑kvkt。

6.4主成分分析

SVD的近似性质可以用来说明SVD与主成分分析（PCA）的等价性。假设x∈rm×n是一个数据矩阵，其中每列都是一个均值为零的实值随机向量的观测值。假设矩阵居中，即每列的平均值等于零。设x的svd为x=u∑v t，右奇异向量vi称为x的主分量方向[47，p.62]。矢量

z1=xv1=σ1u1

在x列的所有标准化线性组合中，样本方差最大：

.

使用线性代数术语，求最大方差的向量，等于最大化瑞利商：

T T

=argmax.

归一化变量u1=（1/σ1）xv1称为x的归一化第一主成分。

在确定了最大样本方差的向量之后，我们通常要继续寻找与第一个样本方差正交的第二大样本方差的向量。这是通过计算缩减数据矩阵最大样本方差的向量来实现的。继续这个过程，我们可以按顺序确定所有的主要成分，也就是说，我们计算奇异向量。在程序的一般步骤中，随后的主分量被定义为最大方差的向量，受其与前一个方差正交的约束。

那些。

例6.9.PCA如图6.2所示。从相关正态分布中生成500个数据点，并收集在数据矩阵X∈R3×500中。数据点和主要组件在图的顶部图表中进行了说明。然后，我们对数据矩阵x1：=x−σ1u1v1t进行放气；对应于x1的数据点在底部图中给出。

6.5解决最小二乘问题

利用支持向量机可以解决最小二乘问题。假设我们有一个超定系统ax～b，其中矩阵a有全列秩。写

SVD

，

6.5.求解最小二乘问题

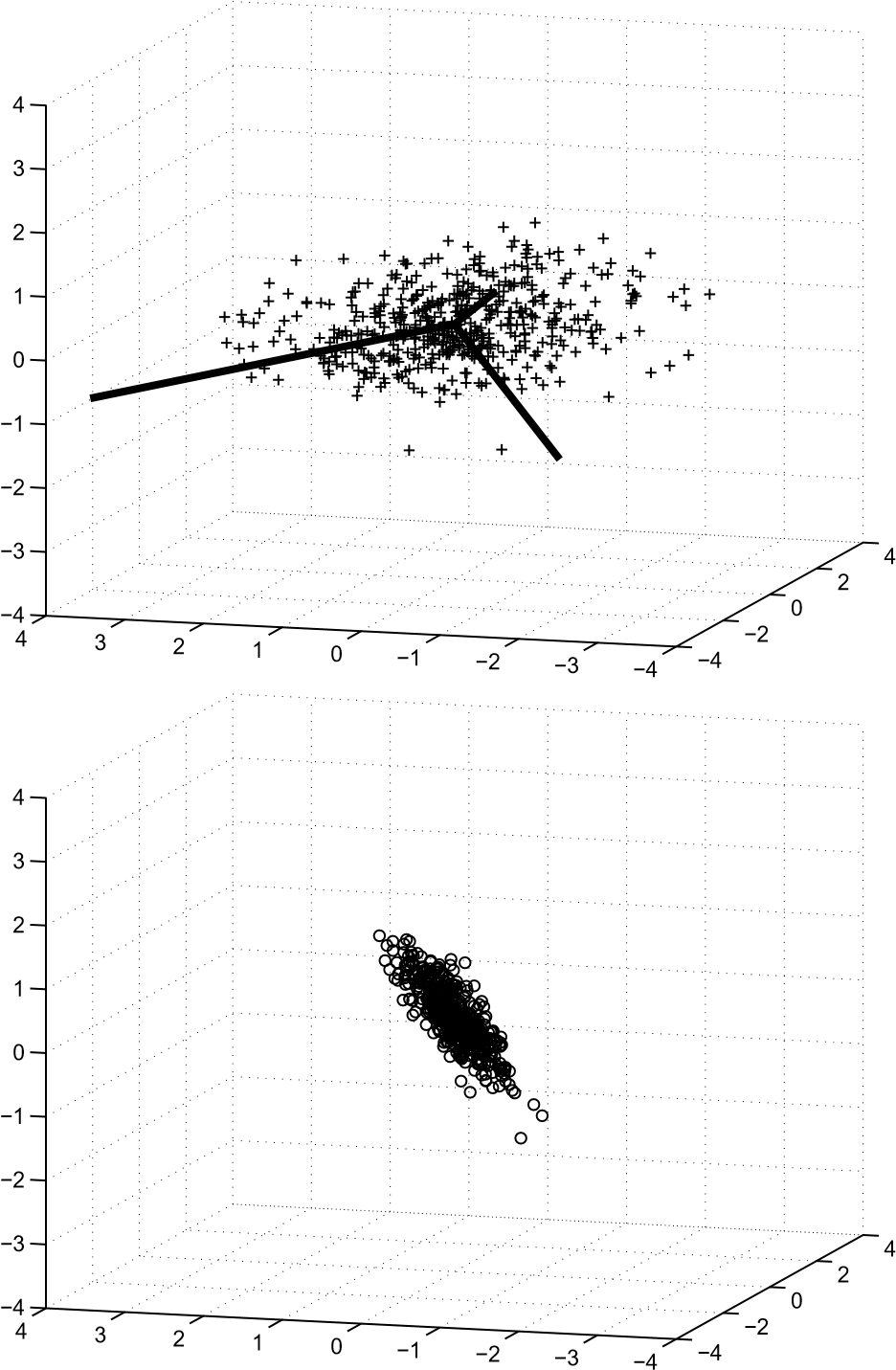


图6.2.顶部：R3中具有（缩放）主要组件的点群。底部：与第一个主成分的贡献相同的数据被缩减。

其中U1∈Rm×N.利用SVD和范数在正交变换下不变的事实，我们得到

，

其中，bi=uitb，y=v tx。因此

.

现在我们可以通过输入y=∑−1b1来最小化。最小二乘解由

（6.5）

记住∑是对角线，

∑1=diag，

因此，也可以编写解决方案

.

假设a具有完整的列秩，意味着所有的奇异值都是非零的：σi>0，i=1,2，…，n。我们还发现在这种情况下，解是唯一的。

定理6.10（SVD最小二乘解）。假设矩阵A∈Rm×N具有全列秩和细SVD A=U1∑V t，则最小二乘问题具有唯一解。

.

例6.11。作为一个例子，我们解决了第3.6章开头给出的最小二乘问题。矩阵和右手边是

|  |  |
| --- | --- |
| A=1 1 | B=7.9700 |
| 1 2 | 2000年10月 |
| 1 3 | 2000年14月 |
| 1 4 | 16.0000元 |
| 1 5  >>[U1，S，V]=SVD（A，0）  U1=0.1600-0.7579  0.2853-0.4675 0.4106-0.1772  0.5359 0.1131 0.6612 0.4035  S=7.6912 0  0.9194美元  V=0.2669-0.9637  0.9637 0.2669 | 2000年2月21日 |

6.6.最小二乘问题的条件数和扰动理论

A中的两列向量是线性无关的，因为奇异值都是非零的。使用（6.5）解决最小二乘问题：

>>X=V\*（S \（U1'\*B））

X=4.2360

三点二二六零

6.6最小二乘问题的条件数和扰动理论

用支持向量机定义了矩形矩阵的条件数。让a有秩r，即其奇异值满足

σ1≥·····································

其中p=最小值（m，n）。然后定义条件编号

.

注意，在一个正方形非奇异矩阵的情况下，这简化为定义。

（3.3）。

Wedin[106]证明了以下微扰定理。

定理6.12。假设矩阵A∈Rm×N，其中m≥n具有全列秩，设x为最小二乘问题的解。设δa

δb是这样的扰动

.

然后扰动矩阵a+δa具有满秩，解δx的扰动满足

，

其中r是剩余的r=b−ax。

这里至少要做两个重要的观察：

* 1. 数字k决定最小二乘问题的条件，如果m=n，则剩余r等于零，不等式成为线性方程组的扰动结果；参见定理3.5。
  2. 在超定情况下，残差通常不等于零。那么条件取决于κ2。如果残差的范数很大，这种依赖性可能很重要。

6.7等级不足和不确定系统

假设a是秩亏，即秩（a）=r<m i n（m，n）。最小二乘问题仍然可以解决，但解决方案不再是唯一的。在这种情况下，我们编写SVD

，（6.6）

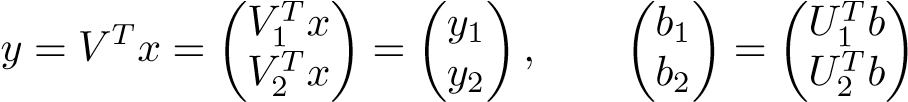
哪里

，（6.7）

并且∑1的对角线元素都是非零的。残差的范数现在可以写了。

.

放



利用正交变换下范数的不变性

变成

.

因此，我们可以通过选择y1=∑−1 1b1来最小化残差。事实上，

，

其中y2是任意的，解决了最小二乘问题。因此，最小二乘问题的解是不唯一的，并且，由于v2的列跨越了a的空空间，所以它在这个空空间中，不确定度在这里。我们可以写信

，

因此，我们选择y2=0得到最小范数的解。我们将推导归纳为一个定理。

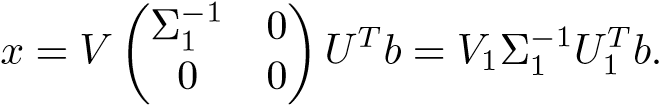
### 定理6.13（最小范数解）。假设矩阵A是秩

SVD不足（6.6），（6.7）。那么最小二乘问题就没有唯一的解了。然而，问题是

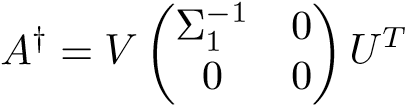
，

6.7.等级不足和不确定系统

有独特的解决方案



矩阵



称为a的伪逆矩阵。它是为任意非零矩阵定义的。

尺寸。

SVD也可用于求解不确定线性系统，即未知量比方程多的系统。设a∈rm×n，m<n。A的SVD是

（6.8）

|  |  |
| --- | --- |
| 显然，如果且只有∑是非奇异的，则a具有整行秩。我们提出一个关于线性系统解的定理。 |  |
| AX＝B | （6.9） |

对于A具有完整行排名的情况。

定理6.14（欠定线性系统的解）。设a∈rm×n与svd（6.8）具有全行秩。那么线性系统（6.9）总是有一个解，然而，这个解是不唯一的。问题

，（6.10）

|  |  |
| --- | --- |
| 有独特的解决方案 |  |
| x=v1∑−1utb。 | （6.11） |

证据。使用SVD（6.8），我们可以编写

.

因为∑是非奇异的，我们可以看到，对于任何右侧，（6.11）都是线性系统的解。但是，我们可以在的空空间中添加任意的解决方案组件，并且我们仍然有一个解决方案。最小范数解，

即（6.10）的解由（6.11）给出。

根据右手边的不同，秩亏情况可能有解，也可能没有解，这种情况可以很容易地用定理6.13来处理。

6.8计算SVD

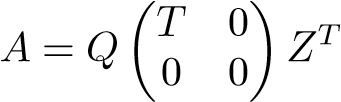
svd在matlab中由语句[u，s，v]=svd（a）计算。此语句是Lapack[1]中算法的实现。（SVD的双精度高级驱动算法称为dgesvd。）在该算法中，首先通过从左到右的一系列户主变换将矩阵简化为双对角形式。然后使用QR算法的变种将双对角矩阵迭代地简化为对角形式；见第15章。

稠密（全）矩阵的支持向量密度可以用O（mn2）触发器计算。根据计算出的量，常数的顺序为5–25。

在matlab中，通过语句[u，s，v]=svds（a，k），计算大型稀疏矩阵的部分svd。此语句基于来自arpack的lanczos方法。我们在第15章中简要描述了Lanczos算法。有关更全面的治疗，请参阅[4]。

6.9完全正交分解

在矩阵秩亏的情况下，计算SVD是确定秩的最可靠的方法。但是，它的缺点是计算成本相对较高，并且更新成本较高（添加新行和/或列时）。这两个问题可能都很关键，例如在实时应用程序中。因此，已经开发出了一种近似SVD的方法，即所谓的完全正交分解法，这种方法在无噪声的情况下和在精确的算法中都可以写成



对于正交q和z以及三角形t∈r r×r，当a具有秩r时，显然，svd是完全正交分解的特例。

在本节中，我们假设矩阵a∈r m×n，m≥n具有精确的或数值的秩r（回顾第63页数值秩的定义）。

### 6.9.1带立柱旋转的QR

获得完全正交分解的第一步是在QR分解的计算中执行柱旋转[22]。在算法开始之前考虑矩阵：计算每列的2个范数，并将最大范数的列移到最左边的位置。这相当于用右边的置换矩阵p1乘以矩阵。在还原为三角形的第一步中，应用了户主转换，以消除第一列中的元素：

.

然后在下一步中，找到b中具有最大范数的列，对这些列进行排列，以便将具有最大范数的列移动到a中的位置2（这仅涉及

6.9.完全正交分解

当然，第2列到第n列），并减少b的第一列：

.

可见r11≥r22。

经过n个步骤后，我们计算出

.

置换矩阵的乘积本身就是置换矩阵。

提案6.15。假设a的秩为r，那么，在精确的算法中，列旋转的qr分解由下式给出：

，

并且R11的对角线元素是非零的（R11是非奇异的）。

证据。显然，过程中出现的对角线元素是不增加的：r11≥r22≥····。

假设r r+1，r+1>0。这意味着QR分解中r的秩大于r，这是一个矛盾，因为r和a必须具有相同的秩。

例6.16。下面的matlab脚本执行QR分解，其中列以构造为缺乏秩的矩阵为轴：

[u，ru]=qr（randn（3））；[v，rv]=qr（randn（3））；

d=diag（[1 0.5 0]）；a=u\*d\*v'；

[q，r，p]=qr（a）；%qr，柱旋转

R=-0.8540 0.4311-0.0642 0 0.4961-0.2910

0 0 0.0000

>>R（3,3）=2.7756E-17

在许多情况下，带有旋转的QR分解提供了关于矩阵数值等级的合理准确信息。我们通过添加噪声来修改前面脚本中的矩阵：

[u，ru]=qr（randn（3））；[v，rv]=qr（randn（3））；

d=diag（[1 0.5 0]）；a=u\*d\*v'+1e-4\*randn（3）；

[q，r，p]=qr（a）；

>>R=0.8172-0.4698-0.1018 0-0.5758 0.1400

0 0 0.000 1

最小对角线元素的数量级与最小对角线元素的数量级相同。

奇异值：

>>SVD（A）=1.0000

0.4999 0.000一

然而，事实证明，人们不能完全依靠这个过程来提供关于可能的等级缺陷的正确信息。我们给出了卡汉的一个例子；见[50，第8.3节]。

例6.17。设c2+s2=1。对于n足够大，三角形矩阵

tn（c）=diag（1，s，s2，…，sn−1）……\_\_\_1−1c−cc·······−cc 1−1c

是非常病态的。对于n=200和c=0.2，我们有

.

因此，在IEEE双精度中，矩阵是奇异的。三角形矩阵的列都有长度1。因此，由于对角线右边每一行的元素是相等的，因此以柱为轴的QR分解不会引入任何柱交换，上三角矩阵r等于tn（c）。但是，底部对角线元素等于s199≈0.0172，因此对于此矩阵qr（具有柱枢转），没有提供任何关于病态条件的信息。

第七章

降秩最小二乘模型

考虑一个线性模型

b=ax+η，a∈rm×n，

式中，η为随机噪声，给出了a和b。如果选择通过最小化残差b-a x的欧几里得范数来确定x，那么就有一个线性最小二乘问题。

（7.1）

在某些情况下，实际的解决方案x本身并不是感兴趣的主要对象，而是一个辅助的中间变量。这是一种情况，例如，在某些分类方法中，当剩余量的范数是有趣的数量时；见第10章。

最小二乘预测是另一个领域，其中的解决方案X是一个中间量，本身并不有趣（除了它应该是稳健和可靠的数字意义）。A列由解释变量的观察结果组成，用于解释因变量B的变化。在这种情况下，必须用近似解\_x来解释B的变化，即相对残差应该是购物中心。给定\_x和解释变量观察的新行向量，可以预测因变量的对应值：

B预测（7.2）

在这种情况下，通常不需要，甚至不希望找到实际将（7.1）中的残余量最小化的解决方案。例如，在预测中，通常有几个解释变量（几乎）是线性相关的。因此，矩阵A往往是非常病态的，最小二乘解受测量误差和浮点舍入误差的影响很大。

七十五

例7.1。matlab脚本

A=[1 0 1 1 1

1 1；

B=[A A\*[1；0.5]+1e-7\*randn（3,1）]；B=B\*[1；1；1]+1e-4\*randn（3,1）；X=B\b

创建矩阵B，其第三列几乎是其他两列的线性组合。该矩阵是非常病态的：条件数为κ2（b）≈5.969·107。脚本给出了最小二乘法的解

X=-805.95

-402.47美元

八百零七点九五

这个近似解很好地解释了因变量的变化，因为残差很小：

resn=标准（b\*x-b）/标准（b）=1.9725e-14

然而，由于解的分量很大，在（7.2）的评估中会有相当大的抵消（见第1.5节），这会导致数值误差。此外，在应用程序中，可能很难解释这种解决方案。

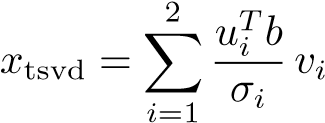
由于矩阵的数值秩为2，最小二乘解与用于构造右手边的向量存在较大偏差。（奇异值为3.1705、0.6691和8.4425·10−8。）鉴于此，接受近似解是合理的。

xt=0.7776

零点八八九一

一点二二二二

使用截断的SVD获得



（参见第7.1节）。该候选解的剩余范数为1.2·10−5，与右侧扰动的数量级相同。由于（7.2）的评估中的取消要小得多或完全消除（取决于新的向量），这样的解向量可能更适合预测。

为了减少问题的不适条件，从而降低解对数据扰动的敏感度，有时会在解X所在的RN中引入一个近似的低维正交基。让基础

7.1.截断SVD：主成分回归

向量是，对于k的某个（小）值，然后根据近似基确定解的坐标，我们在最小二乘问题中使ansatz x=zky并求解

（7.3）

这是一个对应于降秩模型的最小二乘问题。在下面的两个部分中，我们描述了确定这种基向量矩阵的两种方法。第一种方法是基于数据矩阵A的SVD，第二种方法是Krylov子空间方法，其中右侧影响基的选择。

7.1截断SVD：主成分回归

假设数据矩阵具有SVD

，

其中r是a的秩（注意，我们允许m≥n或m≤n）。最小二乘问题（7.1）的最小范数解（见第6.7节）是

（7.4）

当数据矩阵A的SVD“有序变化”从主方向开始时，我们看到和（7.4）中的项也是这样组织的：第一项是沿着数据矩阵主方向的解分量，第二项是沿着主方向的分量。G第二个最主要的方向，等等。

因此，如果我们更喜欢使用SVD诱导的次序，那么我们应该选择（7.3）中的矩阵zk等于a的第一个k右奇异向量（我们假设k≤r）：

.

利用这个事实

，

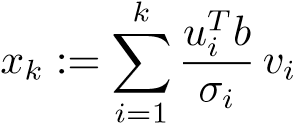
式中，我们得到

.

我们看到最小二乘问题min有解。

，

相当于



作为（7.1）的近似解。这通常被称为截断的SVD解决方案。

通常人们希望找到k的低值，这样剩余值的减少就足够大了。程序可以被表述为一种算法，有时也被称为主成分回归。

### 主成分回归（截断SVD）

1. 求k的最小值，这样。
2. 放

.

参数tol是一个预定义的公差。

例7.2。我们使用示例1.1中的矩阵。让

γ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A=1101000 | 零  零  一  零  一  零  一 | 一  零  零  一  一  一  一 | 零  零  零  一  零  一  零 | 0000010\_\_\_\_\_\_\_ |

0 0 0 1 0\_

γ

0 0 0 0 1 0 0 0 0 1

7.1.截断SVD：主成分回归

0

1

2

3

4

5

0.4

0.5

0.6

0.7

0.8

0.9

1

Truncation index

Relative residual

图7.1.作为截断索引k函数的查询向量q1（实线）和q2（虚线）的残差的相对范数。

回想一下，每一列对应一个文档（这里是一个句子）。我们想知道两个查询向量q1和q2如何用解的奇异值展开（7.4）的前几个项表示，即我们将解决最小二乘问题。

，

对于不同的k值，这两个向量是

，

分别对应单词rank、page、web和english、fifa。在图7.1中，我们绘制了作为k函数的两个向量的残差的相对范数。从示例1.1中，我们发现文档的主要内容与使用google矩阵对网页的排名有关，这反映在主要的奇异向量中。由于q1“包含google术语”，它可以很好地用前几个奇异向量表示。另一方面，第二季度的术语只与“足球文件”有关，因此，作为k的函数，预计第一季度的剩余衰减速度比第二季度快。

q1和q2的前五个左奇异向量的坐标是

|  |  |
| --- | --- |
| u'\*[q1 q2]=1.2132 | 零点一五七四 |
| -0.5474个 | 零点五二一五 |
| 零点七六九八 | 零点七六九八 |
| -0.1817年 | 零点七八三九 |
| 零点三九八一 | -0.3352个 |

向量q1在第一个左奇异向量u1中有一个实质性的分量，因此对于k=1，残差会大大减少。由于q2在u1方面的分量很小，所以在第一步中，残差只会略有减少。

如果我们想将这个例子中的相对残差减少到0.7以下，那么我们应该选择k=2作为q1，k=4作为q2。

7.2 Krylov子空间法

当我们将截断SVD（主成分回归）用于降秩模型时，右侧完全不影响基向量Zi的选择。在实施例7.2中，这一点的影响是明显的，其中向量q2的剩余衰减率比q1慢得多。

在许多情况下，我们希望有一个剩余的快速衰减，作为任何右手边的基向量数的函数。因此，有必要让右侧影响基向量的选择。这是在数值线性代数领域中一种称为Lanczos–Golub–Kahan（LGK）双向化的算法中完成的。在化学计量学和其他领域，一种密切相关的方法被称为偏最小二乘法或潜在结构投影法（PLS）。它是一种从大量krylov子空间方法中提取出来的算法，通常用于稀疏线性系统的求解；参见例如[42，第9-10章]、[80]或关于特征值-奇异值计算，参见第15.8节。

Krylov子空间方法是递归的，但是在我们的推导中，我们将从使用householder转换将矩阵简化为双对角形式开始。本节介绍主要受[15]的影响。

### 7.2.1使用户主转换的双向化

稠密矩阵c∈的SVD算法的第一步

rm×（n+1）是通过户主转换将其简化为上双联形式。

从左到右。我们假设m>n。结果是

，（7.5）

其中p和w是正交的，b\_是上双对角的。分解本身也可用于其他目的。例如，它通常用于最小二乘问题的近似解，既有稠密问题，也有稀疏问题。

我们用一个小例子来说明户主的双向化过程，其中c∈r6×5。首先，第一列中的所有子目录元素都通过从左到右的转换归零（转换中更改的元素用表示）：

.

然后，通过从右到右的不同的户主变换w1，我们在第一行从位置3到N零元素。

，

其中，z1是户主转换。由于这种转换不会改变第一列中的元素，所以我们刚在第一列中引入的零保持不变。第一步的结果是

.

现在，我们以类似的方式继续，通过从左到右的转换，使第二列中对角线以下的所有元素归零。矩阵p2的构造使得它不会改变c1第一行中的元素，即p2具有

结构

，

式中，P∙2∈R5×5为户主转换。我们得到

.

然后，通过从右派的转变，

，

我们在不破坏新引入的零的情况下消灭第二行元素：p1tc1w2=×0×0×0 0 0=：c2。

零

00 0

0 00\_\_\_\_

我们以类似的方式继续，最终获得

（7.6）

一般情况下，

P=p1p2·············································

是户主转型的产物，以及

γ

β1α1\_

B\_=2…..N.∈R（N+1）×N+1）βα2

βαN

βN+ 1

是上双音。

由于正交矩阵的构造方式，它们有一个特殊的结构，将在本章的其余部分中使用。

提案7.3.用p i，i=1,2，…，m表示双相分解（7.6）中p的列，然后

，

其中，c1是c的第一列，z∈rn×n是正交的。

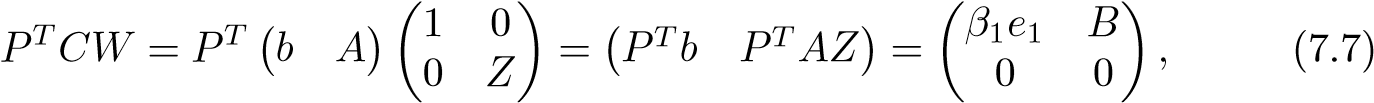
证据。第一个关系式紧接着是ptc1=β1e1。第二个原因是，所有WI都具有这种结构

，

式中，i i∈ri×i为同一矩阵，zi为正交矩阵。

通过户主转换减少双年型需要4mn2−4n3/3次失败。如果，那么首先将a简化为上三角形式，然后将r因子双向化是更有效的方法。

假设现在我们要解决的是最小二乘问题min，其中a∈rm×n，如果我们在双向化过程中选择，那么我们得到一个等价的双向最小二乘问题。利用（7.6）和7.3，我们得到



哪里

.

然后，定义y=ztx，我们就可以写出余数的范数，

（7.8）

如果我们使用平面旋转序列将b简化为上双对角形式（见下文），那么双对角最小二乘问题min可以在o（n）触发器中解决。

### 7.2.2 LGK标准化

现在，我们将给出前一节的bidialogization过程的另一种描述，它允许我们以递归的方式计算分解（7.7）。这是LGK的双向化。（7.7）的最后一个方程的一部分可以写出来。

，

这意味着

.

将两边的i列（i≥2）相等，我们得到

atpi=βizi−1+αizi，

可以写的

αIzi=ATPI−βIzi−1.（7.9）

同样，将第i列等同于

，

我们得到

azi=αipi+βi+1pi+1，

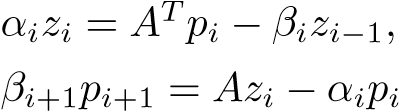
可以写的

βi+1pi+1=azi−αipi。（7.10）

现在，通过从命题7.3，方程（7.9）和（7.10）中编译起始方程β1p1=b，我们得出了一个递归式：

**LGKBidiagonalization**

1. β1p1=b，z0=0
2. 对于i=1:n



1. 结束

确定系数αi−1和βi，以便

如果任何αi或βi等于零，递归就会中断。可以证明（参见，例如[15，第7.2节]）在最小二乘问题的解决中，这些事件在对应于定义明确的特殊情况的意义上是无害的。

递归的bidialogization过程给出了精确的算法，与住户bidialogization的结果相同，因此生成的向量（和（满足pti pj=0和）。然而，在浮点运算中，向量随着递归的进行而失去正交性；见第7.2.7节。

### 7.2.3最小二乘问题的近似解

定义矩阵，以及

.

同样，我们可以把关系az=pb和atp=zbt写成递归，现在我们可以把递归的前k步写成矩阵方程。

azk=pk+1bk+1.（7.11）

考虑最小平方问题。注意，列向量zi是Rn中的正交向量，解决方案x就在这里。假设我们想在向量z1，z2，…，zk所跨越的子空间中找到最佳近似解。这相当于求解最小二乘问题。

，

0.4

0.5

0.6

0.7

0.8

0.9

1

0.4

0.5

0.6

0.7

0.8

0.9

1

0 1 2 3 4 5 0 1 2 3 4 5 K

图7.2.查询向量q1（左）和q2（右）作为子空间维数k的函数的残差的相对范数。截短的SVD解的残差曲线是实心的，对于双化解的残差曲线是虚线。

式中Y∈Rk。从（7.11）我们可以看出，这与解决

，

利用的正交性，我们可以重写

，

因为b=β1p1。接下来是

，（7.12）

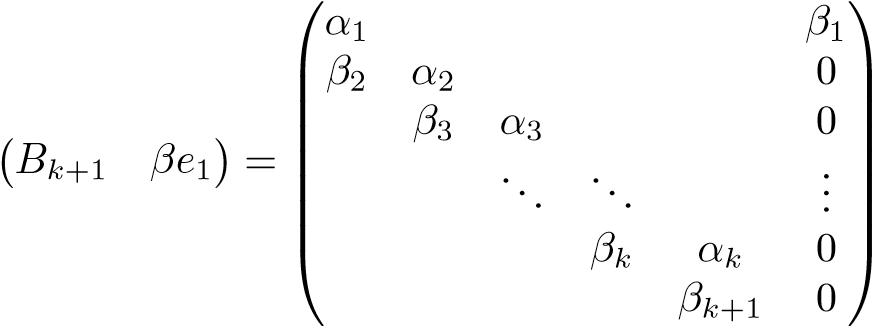
由于采用双对角结构，我们可以求解O（N）触发器；见下文。

例7.4。使用双标准化，我们计算了与例7.2相同的最小二乘问题的近似解。残差的相对范数，

在图7.2中，对于截短的SVD解和双向化过程，绘制为k的函数。结果表明，在这两种情况下，基于双标准化的方法比截断的SVD解更快地衰减残差。因此，在这个例子中，我们让基向量zi受右手边q1和q2的影响，这导致更小维的降秩模型。如果我们想将相对残差降低到0.7以下，那么在这两种情况下，我们都可以用双标准化方法选择k=1。

具有双对角结构的最小二乘问题（7.12）可以用

一系列平面旋转。考虑减少



上三角形。我们现在将证明残差的范数可以很容易地计算出来。在第一步中，我们通过在（1，2）平面上的旋转使β2归零，用余弦和正弦c1和s1。结果是

，

其中已更改的矩阵元素用HAT标记，新的非零元素用+标记。在下一步中，我们用余弦和正弦旋转使β3归零。

.

k步后的最终结果是

，

哪里。如果我们定义

平面旋转的乘积为正交矩阵qk+1∈r（k+1）×k+1，我们得到了qr分解

（7.13）

和

（7.14）

使用QR分解我们可以写

，

最小二乘问题的残差范数等于γk=β1s1···sk。由此可知，当我们生成标量系数αi和βi时，残差范数可以递归计算，因此可以监测残差的衰减。

### 7.2.4矩阵近似

双标准化过程也给出了矩阵A的低阶近似值。在这里，考虑矩阵at进行推导更为方便。假设我们要使用zk列作为RN中的近似基向量。然后，我们可以根据这个基础，通过解最小二乘问题来确定at列的坐标。

（7.15）

引理7.5。给定矩阵a∈Rm×n和矩阵zk∈Rn×k与或-

对于法向列，最小二乘问题（7.15）有解决方案。

sk=pk+1bk+1。

证据。由于ZK的柱是正交的，最小二乘问题有了解。

skt=zkt，

其中（7.11）与sk=pk+1bk+1相同。

从引理我们可以看出，在≈zk（pk+1bk+1）t处有一个最小二乘近似，或者，等价地，

a≈pk+1bk+1zkt。

然而，这不是一个“适当”的秩k近似，因为pk+1∈rm×（k+1）和

bk+1∈r（k+1）×k.现在，用bk+1的qr分解（7.13），我们得到

，

其中wk定义为pk+1qk+1的前k列。现在我们有了适当的秩k近似值a：

（7.16）

a的低阶近似表示为

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | *A* | ≈ |  | |  |

.

如前所述，我们可以将低阶近似解释为如下。wk列是rm子空间的基础。在此基础上，A列的J坐标在YKT的J列给出。

### 7.2.5 Krylov子空间

在lgk双标准化中，我们创建了两组基向量，即pi和zi。它还需要证明它们跨越了哪些子空间。从递归中我们看到z1是atb的倍数，p2是b和aatb的线性组合。

通过简单的归纳证明，我们可以证明

pk∈span b，aatb，（aat）2b，…，（aat）k−1b，zk∈span atb，（ata）atb，…，（ata）k−1atb

对于k=1,2，…。表示

kk（c，b）=span b，cb，c2，…，ck−1b。

这是一个叫做krylov的子空间。我们有以下结果。

提案7.6.pk的柱是kk的正交基（aat，b），zk的柱是kk的正交基（ata，atb）。

### 7.2.6偏最小二乘法

偏最小二乘（PLS）[109，111]是一种计算近似最小二乘解的递归算法，常用于化学计量学。算法存在不同的变体，其中最常见的可能是所谓的nipals公式。

NIPALS PLS算法

1. a0= a
2. 对于i=1,2，…，k

（d）ai=ai−1−u∙iv∙it

该算法与LGK的双向化算法有一些显著的区别，最重要的是一旦计算出一对新的向量（？ui，v？i），就会对数据矩阵进行放气。然而，事实证明[32，110]PLS算法在数学上等同于一个LGK双向化的变体，该变体通过选择不是p1而是α1z1=atb开始。这意味着向量（wi）ki=1在kk（ata，atb）中形成正态基，在kk（aat，aatb）中形成正态基。

### 7.2.7计算双向化

递归版本的双向化存在生成向量失去正交性的缺点。这可以通过使用Gram-Schmidt过程重新调整向量的方法来解决。另一方面，户主bidialogization生成的向量与浮点运算中预期的一样正交；参见第4.4节。因此，对于中等维的稠密矩阵A，应该使用这个变量。

对于大型稀疏矩阵或其他结构矩阵，通常需要使用递归变量。这是因为户主算法通过正交变换修改矩阵，从而破坏结构。请注意，对于这些问题，PLS算法也有同样的缺点，因为它简化了上述算法中的矩阵（步骤（d））。

开发了一个避免存储所有向量pi和zi的lgk双向化版本[75]。

第八章

张量分解

8.1引言

到目前为止，在这本书中我们已经考虑了线性代数，其中的主要对象是向量和矩阵。这些可以分别被认为是一维和二维数据数组。例如，在术语文档矩阵中，每个元素与一个术语和一个文档相关联。在许多应用程序中，数据通常按照两个以上的类别进行组织。相应的数学对象通常称为张量，处理张量的数学领域是多线性代数。这里，为了简单起见，我们把自己局限于张量a=（aijk）∈rl×m×n，这是一组带有三个下标的数据；这样的张量可以用符号表示为

A

=

例8.1。在手写数字分类中，训练集是一组图像，手工分为10个班。每一类都是一组一类的数字，可以认为是张量；见图8.1。如果每个数字都表示为一个16×16的表示灰度的数字矩阵，那么一组n个数字就可以组织成张量A∈R16×16×N。

我们将使用[60]的术语，将张量a∈rl×m×n称为3模阵列，即阵列的不同“尺寸”称为模式。张量a∈rl×m×n的维数是l、m和n。在这个术语中，矩阵是一个双模数组。

九十一

16

16

digits

3

图8.1.一个数字的图像是一个16×16的矩阵，一组数字是张量。

在本章中，我们将矩阵svd推广到3模张量，然后在第14章中，我们将描述如何将其用于人脸识别。对n模张量的进一步推广很容易，可以在[60]中找到。事实上，人脸识别应用程序需要5模式阵列。

20世纪60年代，心理测量学和化学计量学的研究人员率先在数据分析应用中使用张量；参见例如[91]。

8.2张量的基本概念

首先定义两个张量的内积：

（8.1）

相应的规范是

（8.2）

如果我们专门定义矩阵（2模张量），我们会发现这相当于矩阵frobenius范数；见第2.4节。

接下来我们用矩阵定义张量的i模乘法。张量a∈rl×m×n与矩阵u∈rl0×l的单模积，用a×1u表示，是一个l0×m×n张量，其中项由下式给出。

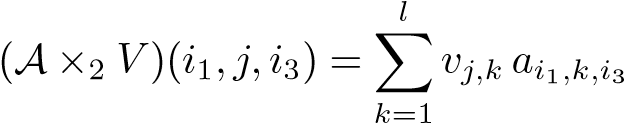
（8.3）

8.2基本张量概念

要进行比较，请考虑矩阵乘法

（8.4）

我们从第2.2节回忆起，矩阵乘法等于将a中的每一列乘以矩阵u。比较（8.3）和（8.4），我们发现张量矩阵乘法的相应性质为真：在1-模积中，3-模数组中的所有列向量都是乘法。用矩阵u表示d。同样，用矩阵v表示张量的2模乘法。



意味着张量的所有行向量都乘以v。注意，矩阵的2模乘v等于矩阵的右乘v t，

A×2V=a v t；

3模乘法是类似的。

有时把张量展开成矩阵很方便。张量A沿三种模式展开的定义（使用（半）matlab符号；对于一般定义，见[60]）如下：

展开，展开，展开。

可以看出，沿模式i展开使该模式成为矩阵a（i）的第一个模式，其他模式是循环处理的。例如，a（j）的第i行包含a的所有元素，其中的jth索引等于i。下面是另一种放置方法。

1. A的列向量是A（1）的列向量。
2. A的行向量是A（2）的列向量。
3. A的3模式向量是A（3）的列向量。

a的1展开相当于将张量分为片a（：，i，：）（矩阵）并将片安排在长矩阵a（1）中。

例8.2。设b∈r3×3×3为张量，在matlab中定义为

B（：，：，1）=B（：，：，2）=B（：，：，3）=

1 2 3 11 12 13 21 22 23

4 5 6 14 15 16 24 25 26

7 8 9 17 18 19 27 28 29

然后沿第三种模式展开

>>B3=展开（B，3）

b3=1 2 3 4 5 6 7 8 9

11 12 13 14 15 16 17 18 19

21 22 23 24 25 26 27 28 29

展开操作的逆操作被写入

折叠（展开（a））=a。

要想很好地定义折叠操作，必须提供有关目标张量的信息。在我们稍微非正式的陈述中，我们抑制了这一点。

利用展开折叠运算，我们现在可以建立一个与i模张量乘法等价的矩阵乘法：

a×i u=折叠i（u展开i（a））=折叠i（ua（i））（8.5）

根据定义，i模式和j模式乘法在以下情况下立即通勤：

（a×i f）×j g=（a×j g）×i f=a×i f×j g。

两个i模乘法满足该恒等式

（a×i f）×i g=a×i（gf）。

这很容易用（8.5）证明：

（a×i f）×i g=（foldi（f（unfoldi（a）））×i g

=foldi（g（展开i（foldi（f（展开i（a）））））=foldi（gf展开i（a））=a×i（gf）。

8.3张量SVD

矩阵支持向量机可以用不同的方法推广到张量。我们提出了一个类似于近似主成分分析的推广。它通常被称为高阶SVD（hospd）[60]。

8.3.张量SVD

定理8.3（hospd）。张量a∈rl×m×n可写成

A=S×1U（1）×2U（2）×3U（3），（8.6）

其中u（1）∈rl×l，u（2）∈rm×m，u（3）∈rn×n为正交矩阵。S是与A尺寸相同的张量；它具有所有正交性的性质：S的任意两片片在标量积（8.1）的意义上是正交的：



为了。单模奇异值的定义如下：



他们被命令

（8.7）

其他模式中的奇异值及其顺序类似。

证据。我们只给出了计算正交因子和张量s的公式；有关完整的证明，请参见[60]。计算SVD，

a（i）=u（i）∑（i）（v（i））t，i=1,2,3，（8.8）

然后放

S=A×1（U（1））T×2（U（2））T×3（U（3））T。

仍然需要证明S的切片是正交的，I型奇异值是递减的。

全正交张量S通常称为核心张量。软管如图8.2所示。

|  |  |  |  |
| --- | --- | --- | --- |
| A  = | |  | | --- | | *U*(1) | | S  *U*  (2)  *U*  (3) |

图8.2.软管的可视化。

方程式（8.6）也可写

，

其解释如下：元素spqs反映了奇异向量u（1）p、u（2）q和u（3）s组合的变化。

hosvd的计算很简单，通过以下matlab代码实现，尽管效率有点低：

函数[u1，u2，u3，s，s1，s2，s3]=svd3（a）；%计算三向张量a的hospd

[U1，S1，V]=SVD（展开（A，1））；

[u2，s2，v]=svd（展开（a，2））；

[U3，S3，V]=SVD（展开（A，3））；

S=tmul（tmul（a，u1'，1），u2'，2），u3'，3）；

假设函数tmul（a，x，i）在模式i，a×i x中用矩阵x乘以张量a。

设v为与ui相同维度的正交；然后从恒等式[60]

s×i u（i）=s×i（u（i）v v t）=（s×i v t）（×i u（i）v），

可能出现软管不是唯一的。然而，这种变换破坏了i模奇异值的有序性。因此，hosvd本质上是唯一的；例外情况是沿任何模式都有相同的奇异值。（这与矩阵SVD中出现的不唯一性类型相同。）

在某些应用中，一个模式的尺寸比其他模式的尺寸的乘积大。例如，假设

a∈rl×m×n，L>mn。然后可以证明核心张量S满足

s（i，：，：）=0，i>mn，

我们可以省略核心的零部分，将（8.6）重写为一个薄的hospd，

，（8.9）

在哪里和。

8.4用hospd近似张量

矩阵可以用SVD写成秩1项的和；见（6.2）。张量的类似展开式可以用张量矩阵的定义来推导。

8.4.用hospd近似张量

乘法：张量a∈rl×m×n可以表示为矩阵乘以奇异向量的和：

，（8.10）

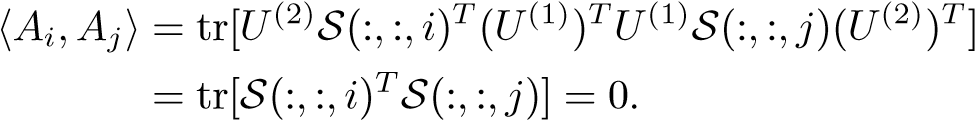
其中u（3）i是u（3）中的列向量。人工智能将被识别为rm×n中的矩阵和rm×n×1中的张量。膨胀（8.10）如图所示：

|  |  |  |
| --- | --- | --- |
| A1 | + | A2 |

+···我是说。

这种扩展与其他模式类似。

很容易证明ai矩阵在标量积（8.1）意义上是正交的：



（在这里，我们用矩阵标识了切片s（：，：，i），并使用了标识tr（ab）=tr（ba）。

现在可以看出，展开式（8.10）可以解释如下。沿张量A第三模的每一片都可以（精确地）用正交基（其中r3是a的正3模奇异值的个数：

，（8.11）

其中，zi（j）是u（3）i的第j个分量。此外，我们还有人工智能的同时正交因式分解，

ai=s（：，：，i）×1u（1）×2u（2），

由于不同j的所有j模奇异值的次序（8.7），其性质是每个s（：，：，i）的“质量”集中在左上角。

角落。

我们将在下面的示例中说明hospd。

例8.4.给出131个手写数字，其中每个数字是16×16矩阵，我们计算了16×16×131张量的hospd。在图8.3中，我们绘制了

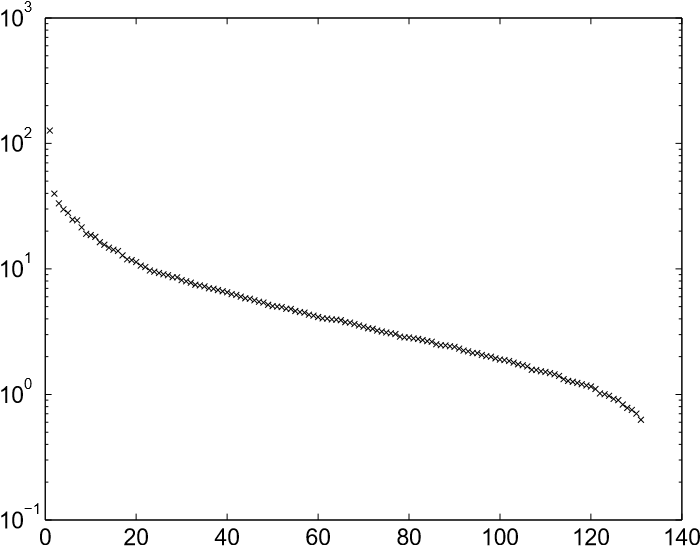


图8.3.数字（第三）模式下的奇异值。

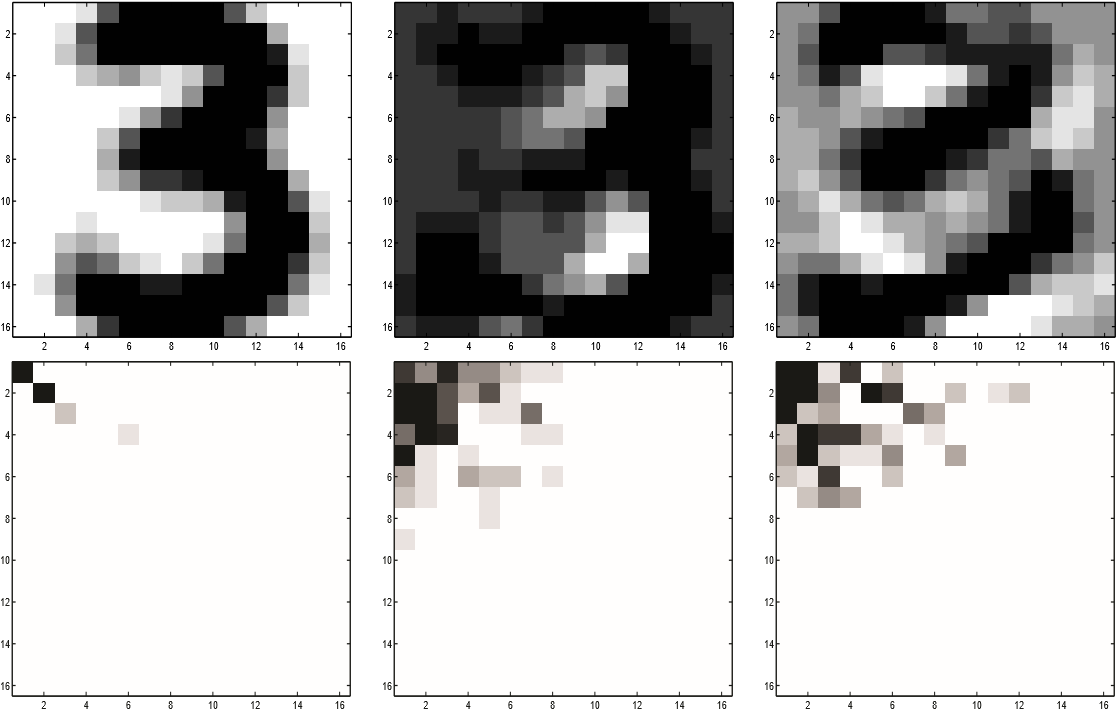


图8.4.顶行显示三个矩阵a1、a2和a3，底行显示核心张量s（：、：、1）、s（：、：、2）和s（：、：、3）（分量的绝对值）的三个切片。

沿第三种模式的奇异值（不同的数字）；可以看出，前20个奇异值占了相当大比例的数字变化（注意对数刻度）。事实上，

.

前三个矩阵a1、a2和a3如图8.4所示。可以看出，第一个矩阵看起来像是不同3的平均值；这是131位数字的主要“方向”，当被视为r256中的点时。接下来的两幅图像代表不同数字之间“平均值”的主要变化方向。

8.4.用hospd近似张量

在图8.4的下一行中，我们绘制了核心张量的三个部分的绝对值：s（：，：，1），s（：，：，2）和s（：，：，3）。可以看出，这些矩阵的质量集中在左上角。

如果我们截断扩展（8.10），

，

对于一些k，那么我们有一个张量的近似值（这里是在第三种模式下），用正交基表示。我们在（8.11）中看到，a的每一个3模切片a（：，：，j）可以写成正交基矩阵aj的线性组合。在手写数字分类中（参见第10章），人们可能需要根据正交基计算未知数字的坐标。由于基的正交性，这很容易做到。

例8.5。让z表示一个未知数字。为了分类的目的，我们要根据前面例子中3的基础来计算z的坐标。这是通过解决最小二乘问题来实现的。

，

其中范数是矩阵Frobenius范数。放

.

因为基与标量积是正交的，

=0用于

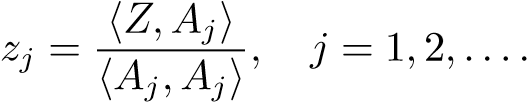
我们可以重写

.

为了求最小值，我们计算了关于zj的偏导数，并把它们等于零，

，

它给出了最小二乘问题的解



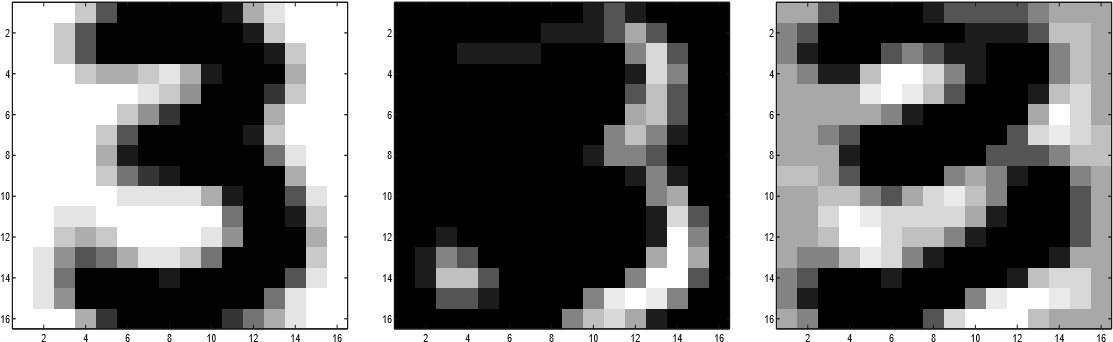


图8.5.手写体3的压缩基矩阵。

由于核心张量的质量集中在三个指数的小值上，因此可以通过hospd在所有三种模式下同时进行数据压缩。这里我们假设在模式i.let中压缩到ki列，并且s\_=s（1:k1,1:k2,1:k3）。然后考虑近似值

.

我们说明如下：

A

≈

*U*

(1)

*k*

1

ˆ

S

*U*

(2)

*k*

2

*U*

(3)

*k*

3

例8.6.我们从示例8.4压缩了3的基矩阵aj。在图8.5中，我们说明了压缩基矩阵

.

参见图8.4中相应的全基矩阵。注意，新的基矩阵a\_j不再是正交的。

第九章

聚类与非负矩阵分解

数据压缩和分类的一个重要方法是在集群中组织数据点。簇是在某些距离测量中紧密相连的一组数据点的子集。我们可以分别计算每个集群的平均值，并使用这些平均值作为集群的代表。等价地，该方法可用作基向量，并且所有数据点都用它们相对于该基的坐标表示。

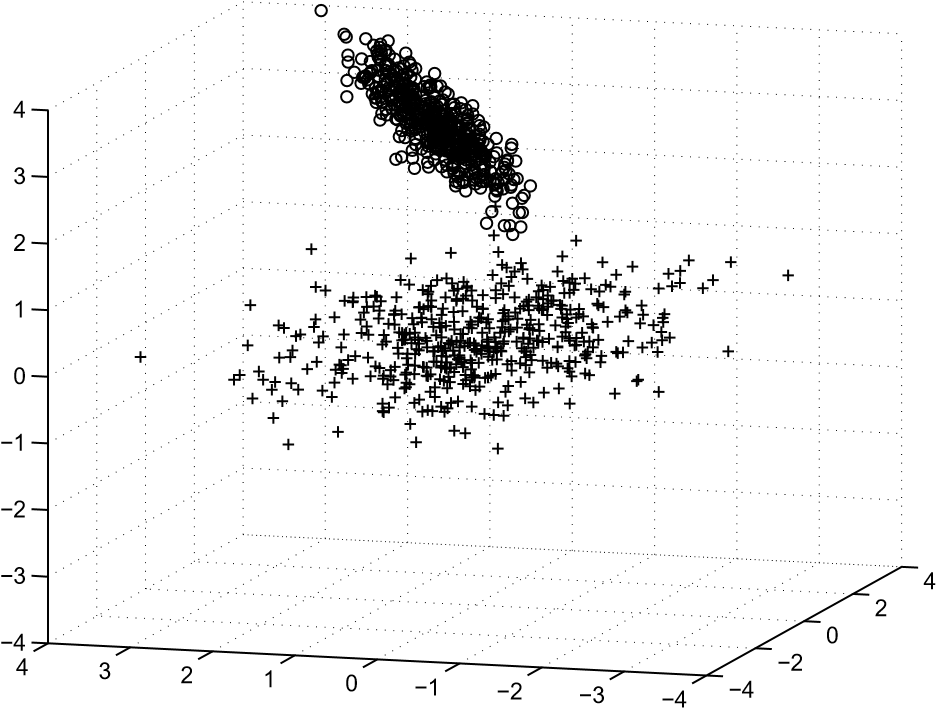


图9.1.r3中的两个簇。

例9.1。在图9.1中，我们展示了r3中的一组数据点，这些数据点是由两个相关的正态分布生成的。假设我们知道有两个集群，我们可以很容易地从视觉上确定哪些点属于哪个类。一种聚类算法采用一组完整的点，并使用一些距离度量对它们进行分类。

一百零一

计算集群有几种方法。其中最重要的是k均值算法。我们在第9.1节中对此进行了描述。

在数据挖掘应用程序中，矩阵通常是非负的。如果我们使用SVD计算矩阵的低阶近似值，那么由于奇异向量的正交性，我们很可能得到带有负元素的因子。用负元素的低阶近似来近似非负矩阵似乎有些不自然。相反，人们通常希望用非负因子计算低阶近似值：

a≈w h，w，h≥0。（9.1）

在许多应用中，非负因子分解有助于解释应用概念中的低阶近似。

在第11章中，我们将对非负矩阵应用一种聚类算法，并将聚类中心作为基向量，即作为（9.1）中矩阵w中的列。然而，这并不能保证h也是非负的。近年来，提出了几种非负矩阵分解算法，并在不同的应用中得到了成功的应用。我们在第9.2节中描述了这种算法。

9.1 k均值算法

假设我们有n个数据点（，我们将其组织为矩阵A∈Rm×N中的列。让=（表示将向量a1，a1，…，an划分为k个簇：

πj=νaν属于簇j。假设群的平均值或质心是

，

式中，nj是πj中的元素数。我们将描述基于欧几里得距离测量的k-均值算法。

簇πj的紧密性或相干性可以用和来度量。

.

矢量越接近质心，qj值越小。聚类的质量可以用整体一致性来衡量：

.

在k-均值算法中，我们寻求一个具有最佳一致性的分区，即它是最小化问题的解。

.

9.1.k-均值算法

算法的基本思想很简单：假设有一个临时分区，就可以计算质心。然后，对于特定集群中的每个数据点，检查是否有另一个形心比当前集群形心更近。如果是这样的话，就需要重新分配。

### k-均值算法

1. 从初始分区\_（0）开始，计算相应的质心向量，（m（0）j）kj=1。计算q（\_（0））。将t=1。
2. 对于每个向量ai，找到最近的质心。如果最近的向量是，则指定。
3. 计算（新分区的）质心。
4. 如果Q（（t−1））−Q（（t））<TOL，则停止；否则，将T增加1并转到步骤2。

初始分区通常是随机选择的。该算法通常具有较快的收敛速度，但不能保证算法找到全局最小值。

例9.2。从乳腺癌诊断研究[66]中选取了一个聚类分析的标准例子。基质A∈R9×683含有乳腺细胞学检查的数据。在683项检查中，444项诊断为良性，239项诊断为恶性。我们在k均值算法中迭代k=2，直到函数q（\_）的相对差小于10−10。使用随机初始分区，迭代分六步收敛（参见图9.2），我们给出目标函数的值。但是需要注意的是，收敛不是单调的：第3步之后的目标函数比第6步之后的小。结果表明，在许多情况下，该算法只给出局部极小值。

由于测试数据是人工分类的，所以我们知道哪些病人患有良性肿瘤和哪些恶性肿瘤，我们可以检查算法给出的聚类。结果见表9.1。在239名恶性肿瘤患者中，K均值算法正确地将222名患者分类，但错误地将17名患者分类。

在第11章和下面的示例中，我们使用集群进行信息检索或文本挖掘。以质心向量作为基向量，用它们的坐标表示文档的基向量。

1

2

3

4

5

6

100

120

140

160

180

200

220

240

260

Iteration

Q(

Π

)

图9.2.乳腺癌数据K均值算法中的目标函数。

表9.1.用k均值算法对癌症数据进行分类。B代表良性，M代表恶性。

|  |  |  |
| --- | --- | --- |
|  | K-表示 | |
|  | 米 | 乙 |
| 米 | 二百二十二 | 十七 |
| 乙 | 九 | 四百三十五 |

例9.3。考虑示例1.1中的术语文档矩阵，

γ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A=1101000 | 零  零  一  零  一  零  一 | 一  零  零  一  一  一  一 | 零  零  零  一  零  一  零 | 0000010\_\_\_\_\_\_\_， |

0 0 0 1 0\_

γ

0 0 0 0 1 0 0 0 0 1

回想一下，前四个文档涉及谷歌和网页排名，而第五个文档涉及足球。利用这一知识，我们可以将前四列向量的平均值作为该簇和第五列向量的质心。

9.1.k-均值算法

作为第二个质心，即我们使用归一化基向量

.

根据这个近似基础，A的柱坐标通过求解

.

考虑到薄qr分解c=qr，这个最小二乘问题的解h=r-1qta

.

我们看到前两列的第二基向量的坐标为负。这在术语文档设置中很难解释。例如，它意味着第一列a1近似于

.

目前还不清楚这一“近似文件”中“英格兰”和“国际足联”这两个词的含义。

最后，我们注意到，为供以后参考，相对近似误差相当高：

（9.2）

在下一节中，我们将在前面的示例中近似矩阵，确保基向量和坐标都是非负的。

9.2非负矩阵分解

给定一个数据矩阵a∈rm×n，我们要计算一个秩k近似，它被约束为具有非负因子。因此，假设w∈rm×k和h∈rk×n，我们要求解

（9.3）

同时考虑到W和H的一个优化问题，该问题是非线性的。然而，如果已知其中一个未知矩阵，比如w，那么计算h的问题将是一个标准的、非负约束的、矩阵右侧的最小二乘问题。因此，最常见的解决方法（9.3）是使用交替最小二乘法（ALS）程序[73]：

### 交替非负最小二乘法

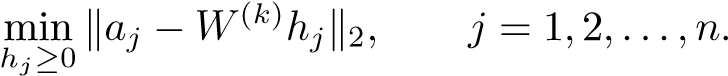
1. 猜测初始值w（1）。
2. 对于k=1,2，…直到收敛
   1. 求解min，得到h（k）。
   2. 求解min，给出w（k+1）。

然而，因式分解wh不是唯一的：我们可以引入任何带正对角元素的对角矩阵d及其因子之间的逆矩阵。

wh=（wd）（d−1h）。

为了避免一个因素的增长和另一个因素的衰退，我们需要在每次迭代中规范化其中一个因素。一种常见的规范化方法是缩放w的列，使每列中的最大元素等于1。

让aj和hj是a和h的列。逐个写出列，我们看到矩阵最小二乘问题等价于n个独立向量最小二乘问题：



这些问题可以通过[61，第23章]中的主动集算法来解决。通过变换矩阵，将确定w的最小二乘问题转化为m独立向量最小二乘问题。因此，ALS算法的核心可以用伪Matlab编写：

9.2.非负矩阵分解

while（not converged）[w]=规范化（w）；对于i=1:n

h（：，i）=lsqnonneg（w，a（：，i））；end for i=1:m w=lsqnonneg（h'，a（i，：）'）；w（i，：）=w'；end

结束

非负矩阵分解算法有很多变种。上述算法存在非负最小二乘主动集算法耗时的缺点。作为一种更便宜的选择，考虑到薄qr分解w=qr，我们可以采用无约束最小二乘法。

H=R−1qta，

然后将h中的所有负元素设为零，同样在算法的另一步中。在[13]中描述了增强稀疏性的改进。

[63]中给出了一个乘法算法：

while（不收敛）

w=w.%（w>=0）；

h=h.%（w'\*v）./（（w'\*w）\*h+epsilon）；

h=h.%（h>=0）；

w=w.%（v\*h’）./（w\*（h\*h’）+epsilon）；

[W，H]=正火（W，H）；结束

（变量epsilon应该被赋予一个小值，并用于避免被零除。）带运算符的矩阵运算。\*和./等价于componentwise语句

.

该算法可以看作是一种梯度下降法。

由于非负矩阵分解法有着许多重要的应用，因此算法的发展是一个活跃的研究领域。例如，为迭代找到终止标准的问题似乎没有找到一个好的解决方案。[13]对不同算法进行了调查。

非负因子分解a≈wh可用于聚类：如果h i j是h的j列中最大的元素，则将数据向量aj赋给聚类i。

[20，37]。

非负矩阵分解法被广泛应用于：

文档聚类和电子邮件监控[85，8]，音乐转录[90]，生物信息学[20，37]和光谱分析[78]，等等。

### 9.2.1初始化

非负矩阵分解的几种算法的一个问题是不能保证收敛到全局最小值。通常情况下，收敛速度很慢，达到次优近似。计算良好初始近似值的有效程序可以基于a[18]的SVD。我们知道第一个k奇异三元组（给出了frobinius范数中a的最佳秩k近似值）。很容易看出，如果a是非负矩阵，那么u1和v1是非负矩阵（参见第6.4节）。因此，如果a=u∑v t是a的svd，我们可以把第一个奇异向量u1作为w（1）中的第一列（如果算法需要，可以把它作为初始近似h（1）中的第一行；下面我们只处理w（1）的近似值）。

由于正交性，下一个最好的向量u2很可能有负分量。但是，如果我们计算矩阵并用零替换所有的负元素，得到非负矩阵，那么我们就知道这个矩阵的第一个奇异向量是非负的。此外，我们希望它是一个相当好的u2近似值，所以我们可以把它作为w（1）的第二列。

该过程可以通过以下稍微简化的matlab脚本来实现：

[u，s，v]=svds（a，k）；%仅计算k最大奇异值

%值和相应的向量

w（：，1）=u（：，1）；对于j=2:k

c=u（：，j）\*v（：，j）'；

c=c.%（c>=0）；

[U，S，V]=SVD（C，1）；

W（：，J）=U；结束

matlab[u，s，v]=svds（a，k）仅使用lanczos方法计算k最大奇异值和相应的奇异向量；见第15.8.3节。标准的SVD函数SVD（A）计算完全分解，通常速度较慢，特别是当矩阵较大且稀疏时。

例9.4.我们使用随机初始化和基于SVD的初始化计算了示例9.3中矩阵A的秩2非负因子分解。使用随机初始化时，收敛速度较慢（参见图9.3），经过10次迭代后，它没有收敛。算法的相对逼近误差

9.2.非负矩阵分解

1

2

3

4

5

6

7

8

9

10

0.58

0.59

0.6

0.61

0.62

0.63

0.64

0.65

0.66

0.67

Iteration

Error

图9.3.非负矩阵因式分解中作为迭代数函数的相对近似误差。上曲线为随机初始化曲线，下曲线为基于SVD的初始化曲线。

SVD初始化为0.574（与K均值算法（9.2）的错误0.596相比）。在某些运行中，具有随机初始化的算法收敛到局部次优最小值。SVD初始化的因子分解是

.

现在可以解释分解。前四个文档由基本向量很好地表示，这些基本向量具有与Google相关的关键字的大组件。相比之下，第五个文档仅由第一个基向量表示，但其坐标比前四个面向Google的文档的坐标小。这样，排名2的近似值就突出了与Google相关的内容，而“足球文档”则被淡化了。在第11章中，我们将看到其他低阶近似，例如基于SVD的近似，具有类似的

效果。

另一方面，如果我们计算一个秩3近似值，那么我们得到

.

我们看到，现在w中的第三个向量实质上是一个“足球”基向量，而其他两个向量则表示与google相关的文档。

第二部分

数据挖掘应用程序

第十章

手写数字分类

手写数字的计算机分类是模式识别中的一个标准问题。典型的应用是自动读取信封上的邮政编码。[62]对不同的算法进行了全面的回顾。

10.1手写数字和简单算法

在图10.1中，我们将在本章的示例中说明手写数字。



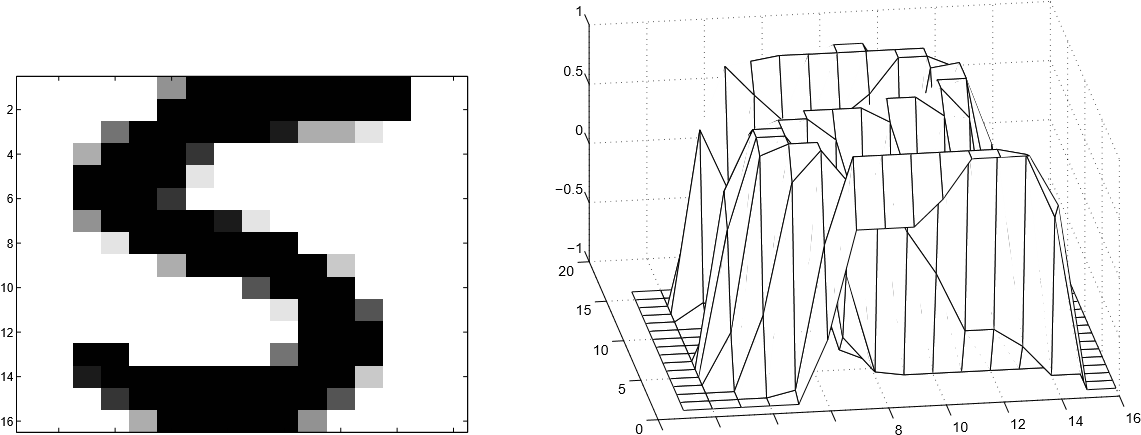
图10.1.来自美国邮政服务数据库的手写数字；参见，例如[47]。

我们将以三种不同但等效的格式处理这些数字：

1. 为16×16灰度图像，如图10.1所示；
2. 作为两个变量的函数，s=s（x，y），如图10.2所示；
3. 作为r256中的向量。

在未知数字的分类中，我们需要计算到已知数字的距离。可以使用不同的距离度量，也许最自然的度量是欧几里得距离：将图像的列堆叠到

一百一十三



2 4 6 8 10 12 14 16 0 2 4 6

图10.2.作为两个变量的函数的数字。

矢量并在r256中将每个数字标识为矢量。然后定义距离函数

.

另一个距离函数是两个向量之间的余弦。

在手写数字分类的实际应用中，例如邮政编码读取，必须考虑硬件和实时因素。在这一章中，我们描述一个理想化的环境。问题如下：

给定一组人工分类的数字（训练集），对一组未知数字（测试集）进行分类。

在美国邮政服务数据库中，培训集包含7291个手写数字。这里我们将使用1707位数字的子集，在0和9之间相对平均分布。测试集有2007个数字。

如果我们把训练集的数字看作向量或点，那么可以合理地假设一类的所有数字在欧几里得256维向量空间中形成一组点。理想情况下，集群是完全分离的（否则对未知数字进行分类的任务将非常困难），集群之间的分离取决于训练数字写得有多好。

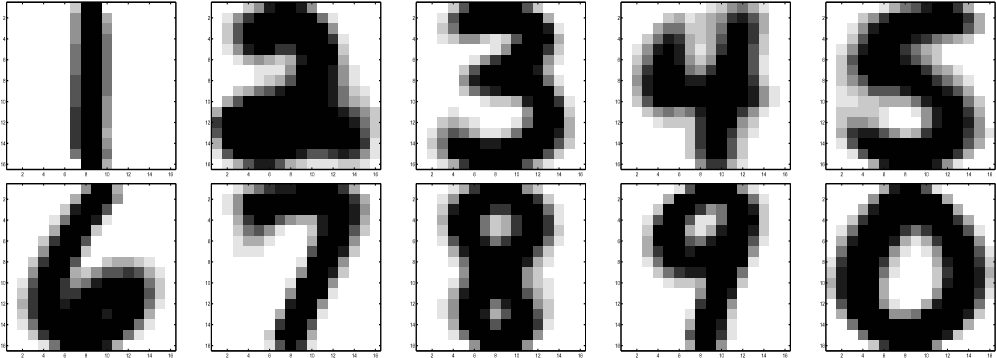


图10.3.训练集中所有数字的平均值（质心）。

在图10.3中，我们说明了训练集中数字的平均值（质心）。从图中我们可以看出大多数数字都写得很好。（如果有许多写得不好的数字，则平均值将非常分散。）这表明簇之间的分隔相当好。因此，计算每个未知数字到平均值的距离的简单分类算法似乎是相当准确的。

### 一种简单的分类算法

训练：给定手动分类训练集，计算所有10个班的平均值（质心）mi，i=0，…，9。

分类：对于测试集中的每个数字，如果mk是最接近的平均值，则将其分类为k。

结果表明，对于我们的测试集，该算法的成功率在75%左右，这还不够好。这种性能相对较差的原因是，该算法不使用任何有关每类数字变化的信息。

10.2使用SVD基础进行分类

我们现在将描述一种分类算法，该算法基于使用SVD计算的正交基向量对每个数字类内的变化进行建模。这可以看作是基于降秩模型的最小二乘算法；参见第7章。

如果我们将图像视为16×16矩阵，那么数据是多维的；参见图10.4。将每幅图像的所有列叠加在一起可以得到一个矩阵。假设a∈rm×n，m=256，是由一类所有训练数字组成的矩阵，比如3。一个跨度的列是Rm的线性子空间。但是，不能期望这个子空间有大的维度，因为如果它有，那么不同类型数字的子空间就会相交（记住，我们考虑的是r256的子空间）。

现在的想法是用子空间的正交基对一类训练（和测试）数字集内的变化进行“建模”。可以使用SVD计算正交基，任何矩阵A都是秩1矩阵的和：

++···（10.1）

A中的每一列代表一个数字3的图像，因此左奇异向量ui是“3的图像空间”中的正交基。我们将左奇异向量称为“奇异图像”。从（10.1）开始，A的第j列等于

到

，

16

16

digits

3

|  |
| --- |
| 一 |

二百五十六

数字

图10.4.一个数字的图像是一个矩阵，一类图像的集合形成张量。在图的下半部分，每个数字（一种）都由矩阵中的一列表示。

我们看到，根据这个基础，图像j在a中的坐标是σivij。根据SVD的矩阵逼近性质（定理6.6和6.7），我们知道第一个奇异向量代表数据矩阵的“支配”方向。因此，如果我们将向量ui折叠成图像，我们期望第一个奇异向量看起来像3，并且下面的奇异图像应该表示围绕第一个奇异图像的训练集的主要变化。在图10.5中，我们说明了训练集3的奇异值和前三个奇异图像。在中间的图中，我们根据前三个奇异向量绘制了131位数字中每个数字的坐标。我们看到所有的数字都有第一个奇异图像的很大一部分（在0.05和0.1之间），事实上，这看起来非常像图10.3中3的平均值。然后我们看到坐标中第二个和第三个单数的变化很大。

图像。

SVD基础分类算法将基于以下假设：

1. 每个数字（在训练集和测试集中）都有其自身类型的第一个奇异图像的特征。“少”的更确切意义应该在实验中研究。
2. 根据前几个奇异图像展开，可以很好地区分不同类别的数字。

三

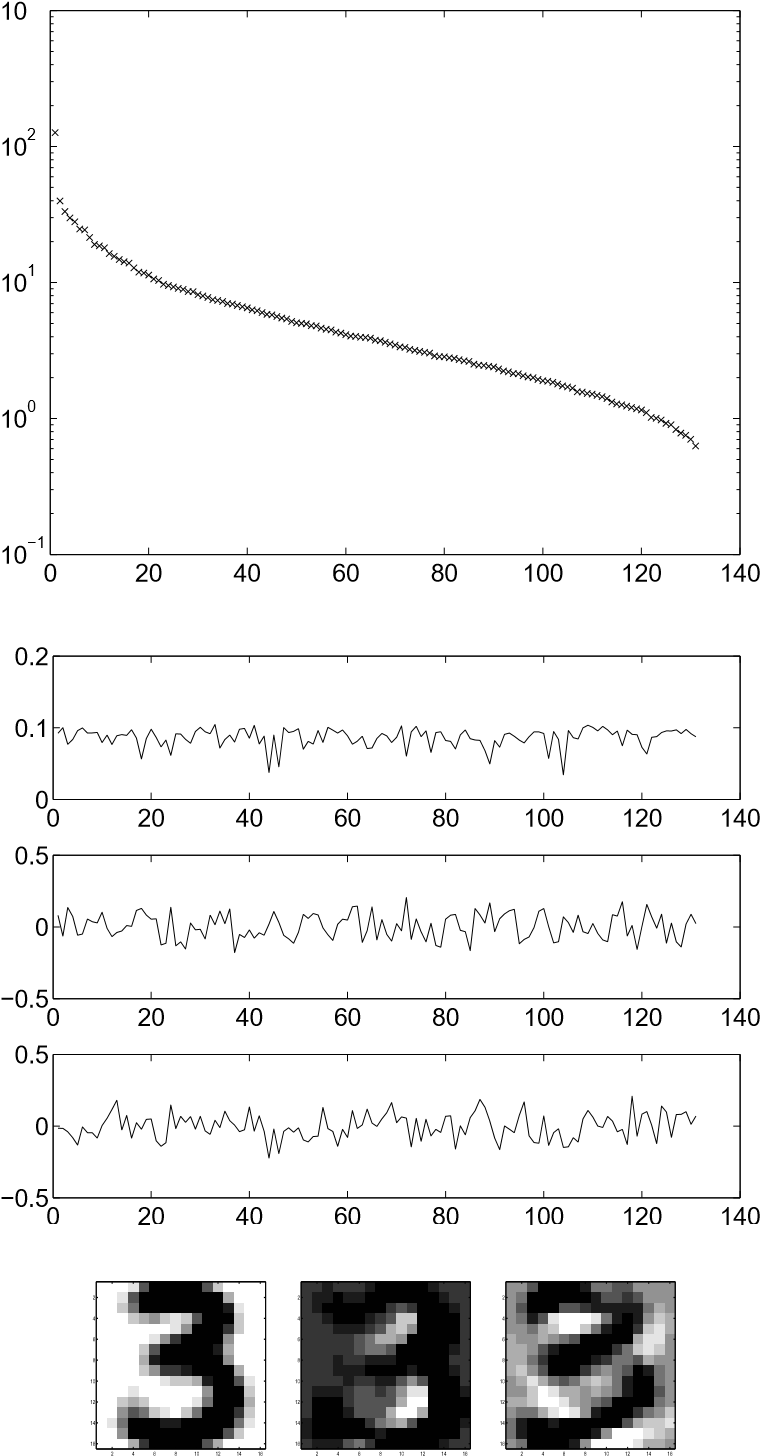


图10.5.奇异值（顶部）、前三个右奇异向量vi（中间）和前三个奇异图像（底部）中131个测试数字的坐标。

1. 如果一个未知数字在一个特定的奇异图像基（即3的基）中可以比在其他类的基中更好地近似，那么该未知数字很可能是3。

因此，我们应该计算一个未知数字在10个不同的基数中的表示能力。这可以通过计算类型的最小二乘问题中的剩余向量来实现。

，

其中z表示未知数字，ui表示奇异图像。我们可以

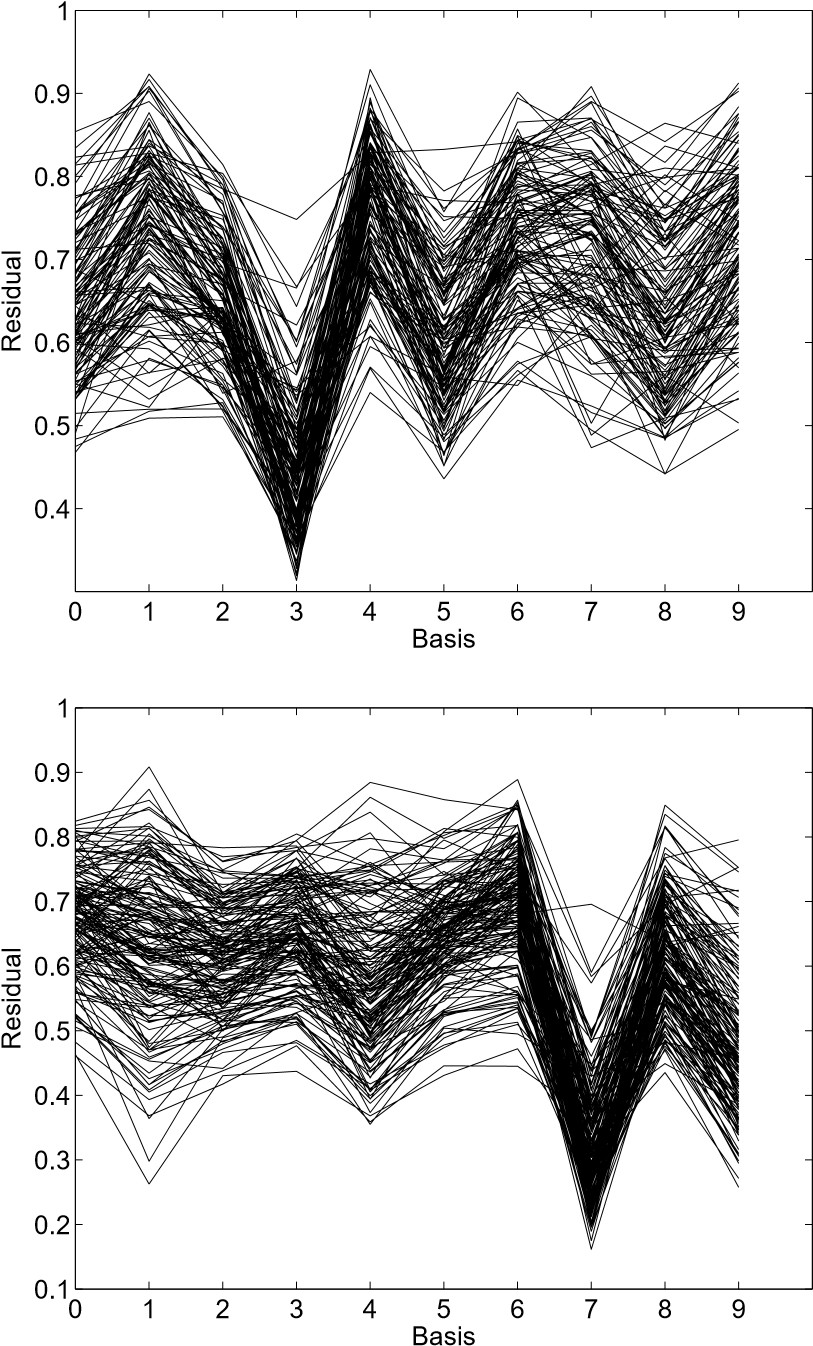


图10.6.所有试验3（顶部）和7（底部）在所有基础上的相对残差。每个类使用10个基向量。

把这个问题写在表格里

，

哪里。由于UK列是正交的，用α=UKTZ求出了该问题的解，最小二乘问题的残差向量范数为

，（10.2）即未知数字在与跨度正交的子空间上的投影范数（UK）。

为了证明上述假设是合理的，我们在图10.6中说明了所有测试3和7在所有10个基础上的相对剩余范数。

图10.7.未知数字（尼斯3）和近似使用1，3，5，7和9项在3基（顶部）。相对残差最小二乘问题（下）。

在这两个数字中，每个未知数字都有一条曲线，自然不可能看到单独的曲线。然而，我们可以看到，大多数的测试3和7是根据他们自己的基础最好的近似。这些图表还提供了关于哪些分类错误比其他分类错误更可能发生的信息。（例如，3’s和5’s是相似的，而3’s和4’s是完全不同的；当然，这只证实了我们已经知道的。）

还有一个有趣的现象，那就是残差是如何依赖于基数中的项数的。在图10.7中，我们用3基图像的不同数目来说明一个写得很好的3的近似值。在图10.8和10.9中，我们给出了3基中丑陋的3和5基中漂亮的3的近似值。

从图10.7和10.9中，我们可以看到nice 3在3-基中的相对残差比5-基中的相对残差小得多。我们还从图10.8中看到，丑陋的3在3个基础上没有很好地表示。因此，自然地，如果数字绘制得很糟糕，那么我们就不能期望得到基于SVD基的清晰分类。

基于支持向量机扩展模型，可以设计出几种分类算法。下面我们给出一个简单的变量。

图10.8.未知数字（丑陋的3）和近似值，使用1、3、5、7和9项作为3基数（顶部）。最小二乘问题中的相对残差

（底部）。

### 一种SVD基分类算法

训练：对于已知数字的训练集，计算每种数字的SVD。

分类：对于给定的测试数字，计算其在所有10个碱基中的相对残差。如果一个残差明显小于所有其他残差，则按此分类。否则放弃。

该算法的工作总结如下：

训练：计算10个M2×NI矩阵的SVD。

每个数字都是m×m的数字化图像。ni：训练位数i。

测试：计算10个最小二乘残差（10.2）。

图10.9.未知数字（尼斯3）和近似使用1，3，5，7和9项在5个基础（顶部）。最小二乘问题中的相对残差

（底部）。

表10.1.将分类作为基础图像数量的函数进行更正（针对每个类）。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| #基础图像 | 一 | 二 | 四 | 六 | 八 | 十 |
| 正确率（%） | 八十 | 八十六 | 九十 | 九十点五 | 九十二 | 九十三 |

因此测试阶段比较快，该算法适合于实时计算。该算法与simca方法有关[89]。

接下来，我们给出一些（来自[82]）美国邮政服务数据库的测试结果，这里有7291个培训数字和2007个测试数字[47]。在表10.1中，我们给出了分类结果，作为每类基础图像数量的函数。

即使与仅使用形心图像的方法相比，性能有了非常显著的改善，但由于最佳算法达到97%左右的正确分类，结果仍然不够好。培训和测试包含一些很难分类的数字；我们在图10.10中给出了一些示例。这样写得不好的数字很难自动处理。

二

四

六

八

十

十二

十四

十六

2 4 6 8 10 12 14 16 2 4 6 8 10 12 14 14 2 4 6 8 10 12 14 16 2 6 8 10 12 14 16

图10.10.美国邮政数据库中的丑陋数字。

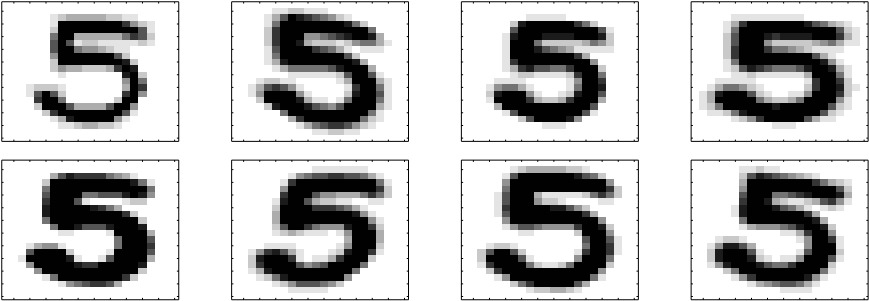
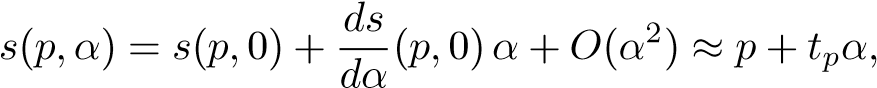


图10.11.数字（左）和可接受的转换（右）。从左到右按列排列，数字（1）用一支更细、更粗的笔书写，（2）对角拉伸，（3）垂直压缩和拉长，以及（4）旋转。

10.3切线距离

一个好的分类算法应该能够对写得相当好但在欧几里得距离上仍与理想数字有很大偏差的未知数字进行分类。有些偏差是人类很容易处理的，而且是很常见和可以接受的。我们在图10.11中说明了一些这样的变化。这种转换对人类读者来说并不困难，理想情况下，在自动数字识别中，它们应该非常容易处理。[86，87]中描述了在小的此类变换下不变的距离测量，即切线距离。

16×16图像可以解释为r256中的点。设p为图像中的固定模式。我们首先考虑只允许一个转换的情况，比如说，在X方向转换模式（数字）。这种转换可以被认为是沿着r256中的曲线移动模式。将曲线参数化为实参数α，使曲线由s（p，α）给出，并以s（p，0）=p的方式给出。一般来说，曲线是非线性的，可以用泰勒展开式中的前两项来近似。



其中0）是r256中的向量。通过在0附近稍微改变α，我们沿着曲线上点P的切线做一个小的图案移动。假设我们有另一个模式e，近似于：

s（e，α）≈e+teα。

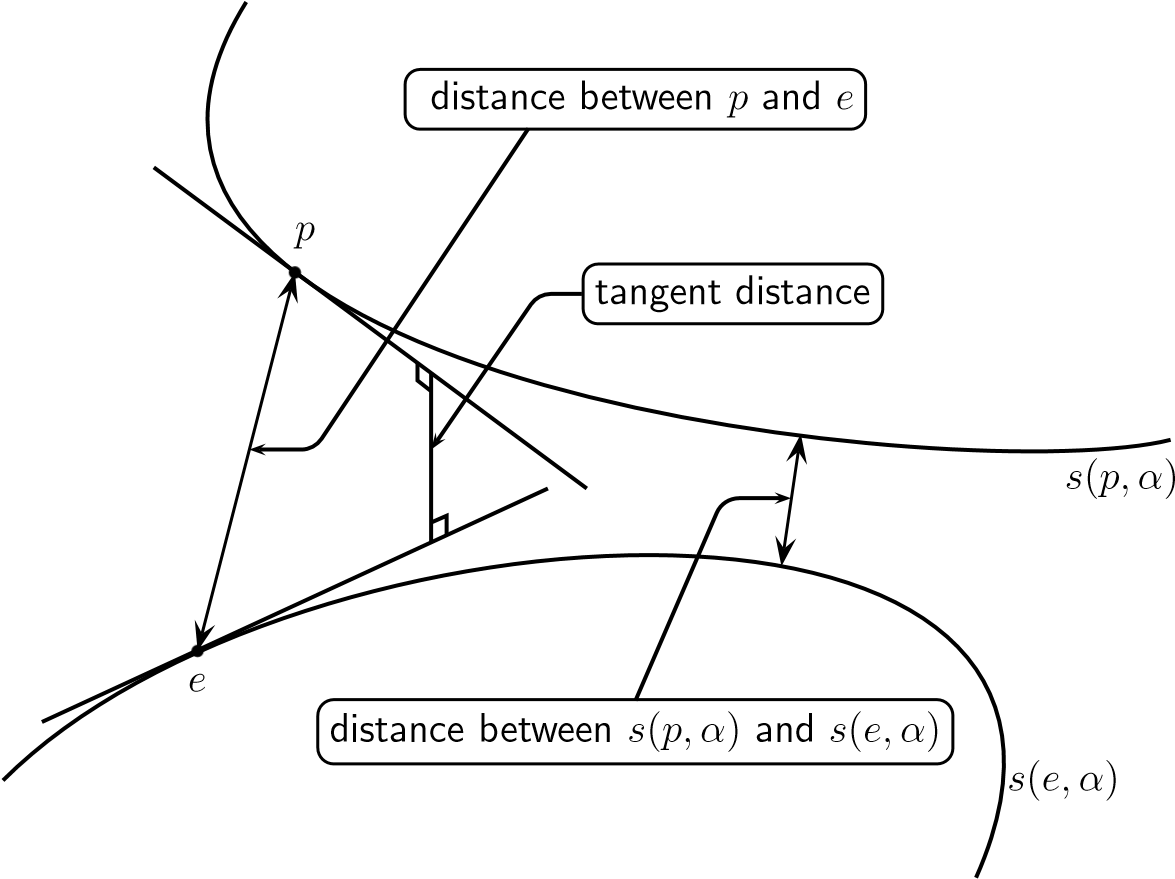


图10.12.点P和E之间的距离，以及切线距离。

因为我们在允许的情况下考虑沿着曲线的小运动，所以这样的小运动不应影响距离函数。因此，理想情况下，我们希望将p和e之间的紧密性度量定义为两条曲线之间的最近距离；见图10.12（参见[87]）。

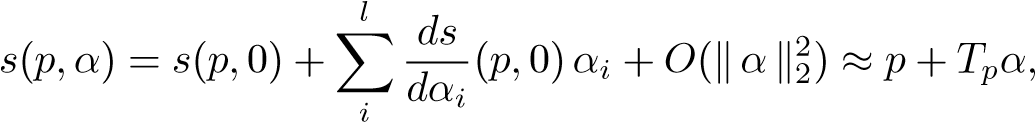
然而，由于一般我们不能计算曲线之间的距离，我们可以使用一阶近似值，并计算点P和E中两条切线之间的最近距离。因此，我们将沿着它们各自的切线独立移动图案，直到找到最远的距离。如果我们用通常的欧几里得范数来测量这个距离，我们就求解最小二乘问题。-

莱姆

.

现在考虑一下这样的情况，当我们允许沿着l在r256中的不同曲线移动模式p时，用α=（α1···αl）t参数化。这相当于在r256中的l维表面（流形）上移动模式。假设我们有两个模式，p和e，每个模式都允许在其允许转换的表面上移动。理想情况下，我们希望找到曲面之间最近的距离，但是，由于这是不可能计算的，我们现在定义一个距离测量，在这里我们计算点p和e中曲面的两个相切平面之间的距离。

如前所述，切线平面由函数s（p，α）的泰勒展开式中的前两项给出：



其中tp是矩阵

，

导数都在点（p，0）中进行了计算。

因此，点P和点E之间的切线距离被定义为最小二乘问题中可能的最小残差，

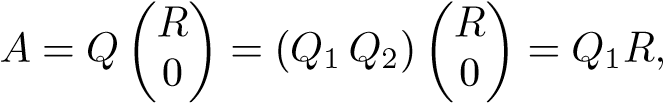
.

可以解决最小二乘问题，例如，使用的SVD。

请注意，我们对解决方案本身不感兴趣，只对残差的范数感兴趣。把最小二乘问题写在表格中

.

如果我们使用QR分解



残差的范数由下式给出

.

当矩阵A碰巧没有完整的列秩时，可以使用SVD轻松处理；参见第6.7节。当两个模式接近时，相切矩阵的列几乎成线性依赖关系的概率很高。

这个距离函数最重要的特性是它在切线平面上图案的运动下是不变的。例如，如果我们在一个图案的X方向做一个小的平移，那么用这个度量，它移动的距离等于零。

### 10.3.1改造

在这里，我们把图像模式看作是两个变量的函数，p=p（x，y），我们证明了每个变换的导数可以表示为微分算子，它是导数px=dxdp和

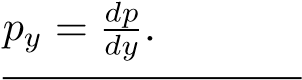


图10.13.一个模式，它的x-导数和该模式的x-翻译。

翻译。最简单的转换是模式在x方向上由αx转换，即：

s（p，αx）（x，y）=p（x+αx，y）。

显然，使用链式法则，

.

在图10.13中，我们给出了一个模式及其x-导数。然后我们证明，通过添加一个小的导数倍数，模式可以被转换为左和右。

类似地，对于y-翻译，我们得到

.

旋转。用点（x，y）中的p值替换点（x，y）中的p值，以此角度αr旋转图案。

.

因此我们定义了函数

s（p，αr）（x，y）=p（x cosαr+y sinαr，−xsinαr+y cosαr），

我们得到了导数

.

设置αr=0，我们有

，

其中衍生产品的评估值为（x，y）。

图10.14给出了旋转变换的示例。

图10.14.一个图案，它的旋转导数，以及图案的旋转。

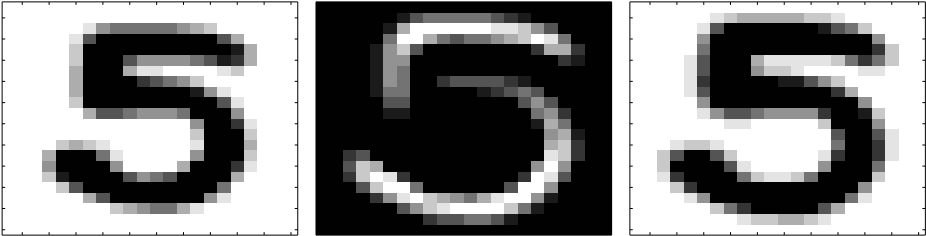


图10.15.模式及其缩放导数，以及模式的“向上缩放”。

缩放。通过定义

s（p，αs）（x，y）=p（（1+αs）x，（1+αs）y），

我们得到了导数

.

比例转换如图10.15所示。

通过定义

s（p，αp）（x，y）=p（（1+αp）x，（1-αp）y），

我们可以把图案拉伸到与轴线平行的地方。衍生产品是

.

在图10.16中，我们展示了平行双曲线变换。

对角双曲变换。通过定义

s（p，αh）（x，y）=p（x+αhy，y+αhx）

我们可以沿着对角线拉伸图案。衍生产品是

.

在图10.17中，我们展示了对角双曲变换。

图10.16.一个图案，它的平行双曲线导数和两个拉伸图案。

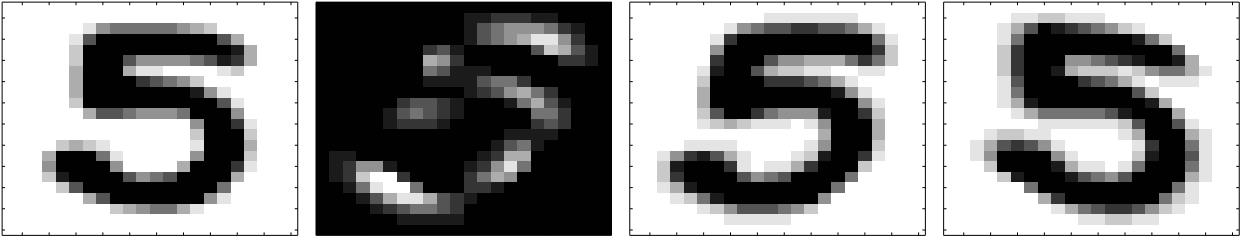


图10.17.一个图案，它的对角双曲导数，和两个拉伸图案。

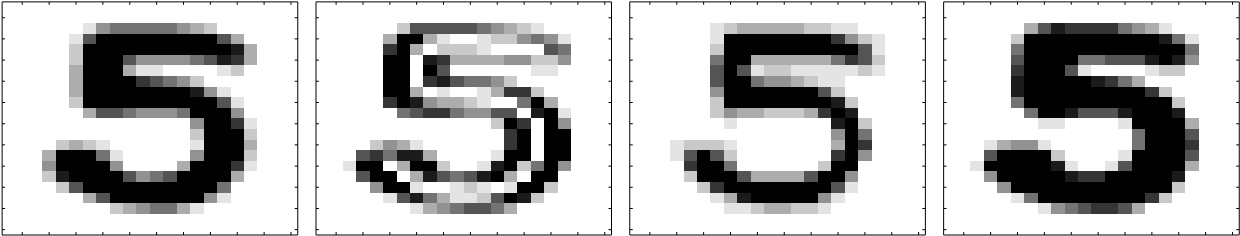


图10.18.一个图案，它的增厚衍生物，一个更薄的图案，和一个更厚的图案。

加厚。使用类似技术可以使图案变薄或变厚；有关详细信息，请参见[87]。“增厚”衍生物是

（px）2+（py）2.

增稠和减薄如图10.18所示。

切线距离分类算法

训练：对于训练集中的每个数字，计算其切线矩阵tp。

分类：对于每个测试数字，

* 计算其正切矩阵；
* 计算所有训练位数的切线距离，并将测试位数分类为最近的训练位数。

虽然该算法在分类性能方面相当好（美国邮政服务数据库[82]的正确分类率为96.9%），但它非常昂贵，因为每个测试数字都要与所有训练数字进行比较。为了具有竞争力，它必须与其他一些减少切线距离比较数量的算法相结合。

在本章结尾，我们指出有必要以不同的方式对数字进行预处理，以增强分类；见[62]。例如，如果图像平滑（与高斯核卷积）会提高性能[87]。在[82]中，导数px和py通过有限差分进行数值计算。

第十一章

文本挖掘

通过文本挖掘，我们指的是从大量且通常是非结构化的文本集合中提取有用信息的方法。另一个密切相关的术语是信息检索。一个典型的应用是搜索科学论文摘要数据库。例如，在医学应用中，人们可能希望找到数据库中处理特定综合征的所有摘要。因此，我们将一个搜索短语，一个查询，以及与综合症相关的关键字放在一起。然后利用检索系统将查询结果与数据库中的文档进行匹配，并向用户展示相关的文档，并根据相关度进行排序。

例11.1。以下是医疗集合中搜索的典型查询

# 摘要：

9。在心脏外科、神经外科、头部损伤和传染病中使用诱导性低温。

这个查询是从一个名为medline的测试集合中获取的，我们将这个查询称为q9。

库目录是文本挖掘应用程序的另一个示例。

例11.2。为了说明信息检索中的一个问题，我们在linko–ping大学图书馆期刊目录中进行了搜索：

|  |  |
| --- | --- |
| 搜索短语 | 结果 |
| 计算机科学工程 | 找不到任何东西 |
| 计算机科学工程 | IEEE：计算输入  科学与工程 |

当然，我们希望系统对用户的小错误不敏感。任何人都可以看到IEEE日志接近查询。从

一百二十九

这个例子我们得出结论，在许多情况下，直接的词匹配是不够好的。

Web搜索引擎是一个非常著名的文本挖掘领域，它的搜索短语通常很短，而且常常有许多相关文档，因此将它们全部呈现给用户是不可能的。在该应用程序中，搜索结果的排名对于搜索引擎的效率至关重要。我们将在第12章中讨论这个问题。

有关信息检索的概述，请参见，例如[43]。在本章中，我们将简要介绍一种最常用的文本挖掘方法，即向量空间模型[81]。本文简要介绍了向量空间模型和一些变体：使用termdocument矩阵的SVD的潜在语义索引（LSI）、基于聚类的方法、非负矩阵分解和lgk双向化。有关与向量空间模型相关的不同技术的更详细说明，请参见[12]。

11.1文件和查询的预处理

在本节中，我们将讨论在建立特定文档集合的向量空间模型之前对文本进行的预处理。

在信息检索中，携带有关文档内容信息的关键字称为术语。信息检索的基本步骤是创建文档集合中所有术语的列表，即所谓的索引。对于每个术语，将存储包含该特定术语的所有文档的列表。这被称为反向索引。

但是在建立索引之前，必须完成两个预处理步骤：清除所有停止词和词干。

停止词实际上可以在任何文档中找到。因此，在文档中出现这样一个词并不能将此文档与其他文档区分开来。以下是一个特定停止列表的开头：

a，a，ability，about，above，accordinate，across，actually，after，after，again，again s t，ain，all，allow，allow，allow，already，already，already，already，虽然，总是，在，在，在，在，在，一，和，另一，任何人，任何人，任何人，任何东西，无论如何，任何地方，apaRT，出现，欣赏，恰当，是，不是，周围，作为，旁白，问…。

词干化是将每个共轭词或词干有一个后缀的词还原的过程。显然，从信息检索的角度来看，在以下简化中不会丢失任何信息：

11.2.向量空间模型

### 可计算计算计算

公共域词干算法在互联网上可用。

表11.1.medline集合的索引的开头。使用波特-斯坦默和GTP解析器[38]。

|  |  |
| --- | --- |
| 无堵塞 | 有堵塞 |
| 行动 | 行动 |
| 动作激活 | 活性剂 |
| 积极主动活动活动实际行动 | 实际的 |
| 实际上敏锐 | 阿奎蒂 |
| 严重的 | 阿库特 |
| 广告 | 广告 |
| 适应 | 适应 |
| 自适应添加 | 添加 |
| 附加附加 | 栓剂 |

例11.3。我们分析了medline集合中的1063个文档（实际上是30个查询和1033个文档），不管是否有词干，在这两种情况下都删除了停止词。为了保持一致性，必须对停止列表执行相同的词干。在第一种情况下，术语数为5839，在第二种情况下为4281。我们在表11.1中列出了部分术语列表。

11.2矢量空间模型

向量空间模型的主要思想是创建一个术语文档矩阵，其中每个文档都由列向量表示。该列在与文档中可以找到的术语相对应的位置上有非零条目。因此，每一行代表一个术语，并且在与可以找到该术语的文档相对应的位置上具有非零条目；参见第11.1节中的倒排索引。

第1章给出了术语文档矩阵的简化示例。在这里，我们手动计算术语的频率。对于实际问题，我们使用文本解析器来创建术语文档矩阵。[38113]中描述了两个公共域解析器。除非另有说明，我们已经将[113]中的例子用于本章中更大的示例。用于信息检索的文本分析器通常包括词干分析器和删除停止词的选项。此外，还有一些过滤器，例如用于删除文档中的格式代码，例如HTML或XML。

通常，不仅要计算文档中出现的术语，还要应用术语加权方案，其中a的元素根据文档集合的特性进行加权。同样，通常会进行文档加权。在[12，第3.2.1节]中描述了许多方案。例如，可以通过定义

aij=fij对数（n/ni），（11.1）

其中fij是术语频率，术语i出现在文档j中的次数，ni是包含术语i的文档数（反向文档频率）。如果一个术语经常出现在少数几个文档中，那么这两个因素都很大。在这种情况下，该术语在不同的文档组之间有很好的区别，并且（11.1）中的日志因子在出现的文档中赋予了很大的权重。

通常，术语文档矩阵是稀疏的：大多数矩阵元素等于零。然后，当然，避免存储所有的零，而是使用稀疏矩阵存储方案；参见第15.7节。

例11.4。对于使用gtp[38]解析的stemmed medline集合，矩阵（包括30个查询列）为4163×1063，其中48263个非零元素，即约1%。矩阵的前500行和前500列如图11.1所示。

### 11.2.1查询匹配与性能建模

查询匹配是查找与特定查询q相关的文档的过程。这通常使用余弦距离度量来完成：如果查询q和aj之间的角度足够小，则认为文档aj是相关的。同样，如果

，

其中tol是一个预定义的公差。如果减小了公差，则返回更多的文档，其中许多文档可能与查询相关。但在

11.2.向量空间模型

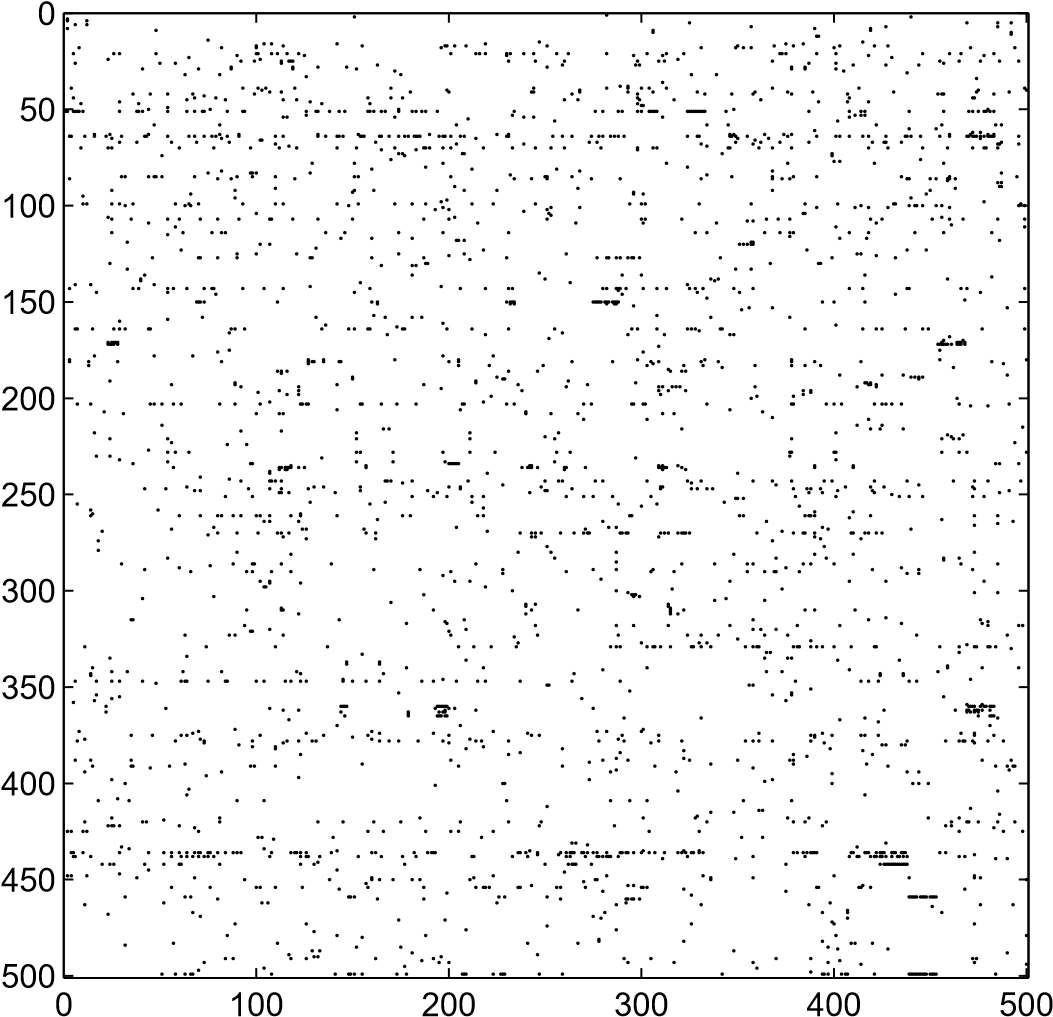


图11.1.medline矩阵的前500行和列。每个点代表一个非零元素。

同时，存在这样的风险：当公差降低时，也会返回越来越多与之无关的文档。

例11.5。我们在stemmed medline集合中对查询q9进行了查询匹配。对于cosine测量，tol=0.19，只有文件409被认为是相关的。当公差降低到0.17时，还检索到文件415和467。

我们在与图11.2中的两个公差值匹配的查询中说明了不同类别的文档。当返回的两组文档与相关文档的交集尽可能大，返回的无关文档数较少时，查询匹配会产生良好的结果。对于高公差值，检索到的文件可能是相关的（图11.2中的小圆圈）。当余弦公差降低时，交集增加，但同时返回更多无关文档。

在信息检索的性能建模中，我们定义了以下度量：

精度：，（11.2）

其中dr是检索到的相关文档数，dt是总数

RELEVANT

DOCUMENTS

RETURNED

DOCUMENTS

ALL DOCUMENTS

图11.2.两个公差值的检索和相关文件。虚线圆圈表示为高余弦公差值检索到的文档。

检索到的文档数，以及

召回：，（11.3）

其中nr是数据库中相关文档的总数。由于cosine测量的tol值很大，因此我们希望具有较高的精度，但召回率较低。对于一个小的TOL值，我们将有高召回，但低精度。

在评估不同的信息检索方法和模型时，通常会使用一些查询。出于测试目的，所有文档都已由人工读取，并且与某个查询相关的文档都已标记。这使得绘制回忆与精确的图表成为可能，以说明某一信息检索方法的性能。

例11.6。我们使用cosine度量对medline集合（stemmed）中的查询q9进行了查询匹配。对于公差的特定值，我们从（11.2）和（11.3）中计算出相应的召回和精度。通过将公差从接近1变为0，我们得到了召回和精度的向量，这些向量给出了有关此查询的检索方法质量的信息。在不同方法的比较中，如图11.3所示，说明了绘制召回与精确图。理想情况下，一种方法在精度较高的同时具有较高的召回率。因此，曲线越靠近右上角，检索质量越高。

在本例和以下示例中，使用术语频率和反向文档频率加权（11.1）计算矩阵元素。

11.3.潜在语义索引

0

20

40

60

80

100

0

10

20

30

40

50

60

70

80

90

100

Recall (%)

Precision (%)

图11.3.使用向量空间方法查询Q9的匹配。回忆与精确。

11.3潜在语义索引

潜在语义索引（LSI）[28，9]是基于数据中存在一些潜在的潜在语义结构的假设。它被广泛使用的字所破坏，并且可以通过使用SVD将数据（术语文档矩阵和查询）投影到低维空间来发现和增强这种语义结构。

设a=u∑v t为术语文档矩阵的svd，并用秩k的矩阵进行近似：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | *A* | ≈ |  | |  |

.

UK列位于文档空间中，是我们用来近似文档的正交基础。根据列向量写出hk，hk=（h1，h2，…，hn）。从a≈ukhk我们得到a j≈ukhj，这意味着hk的j列在正交基上保持了j号文件的坐标。使用秩k近似，术语文档矩阵表示为ak=ukhk，在查询匹配中，我们计算qtak=qtukhk=（uktq）thk。因此，我们根据新的文档基础计算查询的坐标，并根据

（11.4）

这意味着查询匹配是在k维空间中执行的。

例11.7。我们在medline集合中对q9进行了查询匹配，使用秩100的截断SVD近似矩阵。

0

20

40

60

80

100

0

10

20

30

40

50

60

70

80

90

100

Recall (%)

Precision (%)

图11.4.查询Q9的匹配。全向量空间模型（实线）和秩100近似（虚线和菱形）的回忆与精度。

召回精度曲线如图11.4所示。可以看到，对于这个查询，LSI提高了检索性能。在图11.5中，我们还证明了许多术语文档矩阵所共有的一个事实：它是相当好的条件，并且在奇异值序列中没有间隙。因此，我们无法通过检验奇异值来找到合适的LSI近似等级，必须通过检索实验来确定。

另一个值得注意的事实是，当k=100时，矩阵近似中的近似误差，

，

是大的，而且我们仍然得到改进的检索性能。鉴于术语文档矩阵的截断SVD近似存在较大的近似误差，

11.3.潜在语义索引

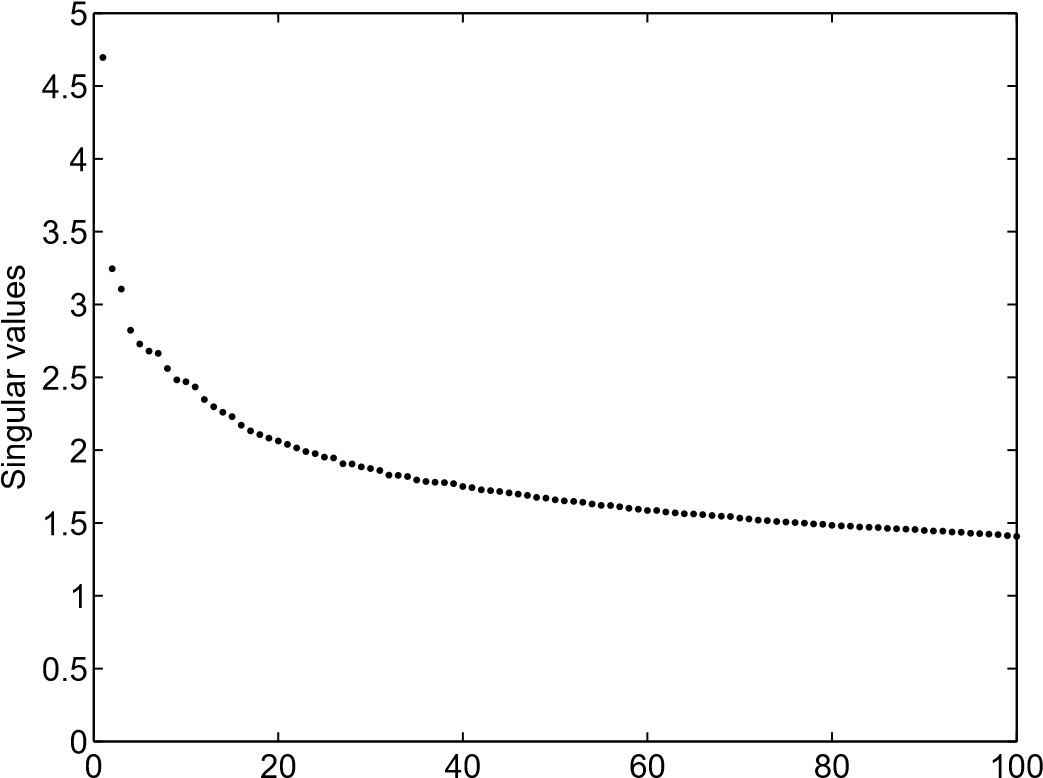


图11.5.medline（stemmed）矩阵的前100个奇异值。矩阵列按欧几里得长度单位进行缩放。

人们可能会质疑“最佳”奇异向量是否构成表示术语文档矩阵的最佳基础。另一方面，由于我们得到了如此好的结果，一个更自然的结论可能是，frobenius规范不是术语文档矩阵中信息内容的良好度量。

同样有趣的是，看看数据中最重要的“方向”是什么。从定理6.6我们知道，前几个左奇异向量是文档空间中的主要方向，它们的最大分量应该指示这些方向是什么。matlab语句find（abs（u（：，k））>0.13）结合术语索引中的查找，得出k=1,2的以下结果：

|  |  |
| --- | --- |
| U（：，1） | U（：，2） |
| 细胞 | 案例 |
| 生长 | 细胞 |
| 激素 | 儿童 |
| 病人 | 缺陷DNA生长患者心室 |

在第13章中，我们将回到从文本中提取关键字的问题。

应该说，LSI并没有为medline集合中的所有查询提供显著更好的结果：有些LSI提供的结果与全向量模型相当，有些LSI提供的性能较差。然而，重要的往往是平均性能。

在[52]中对LSI的不同方面进行了系统研究。结果表明，LSI对降秩k的极小值具有较好的检索性能，同时相对矩阵逼近误差较大。很可能无法证明任何通用的结果来解释LSI可以以何种方式和为哪些数据改进检索性能。相反，我们给出了一个人工的例子（在[12]中使用类似的思想作为相应的例子构造），给出了部分解释。

例11.8。考虑示例1.1中的术语文档矩阵和查询“网页排名”。显然，文档1-4与查询相关，而文档5完全无关。但是，我们得到查询的余弦值和原始数据

，

表示文档5（足球文档）与查询的相关性与文档4相同。此外，由于文档1中没有查询词，因此此文档与查询是正交的。

然后我们计算术语文档矩阵的SVD，并使用秩2近似。投影到二维子空间后，根据（11.4）计算的余弦为：

.

原来，文档1被认为与原始表示形式中的查询完全无关，现在它具有高度相关性。此外，相关文件2-4的余弦值也得到了加强。同时，文件5的余弦值已显著降低。因此，在这个人工例子中，维度约简提高了检索性能。

在图11.6中，我们绘制了前两个左奇异向量坐标系中的五个文档和查询。显然，在这个表示中，第一个文档比文档5更接近查询。前两个左奇异向量是

，

奇异值是∑=diag（2.8546，1.8823，1.7321，1.2603，0.8483）。a中的前四列通过google、matrix等词进行强耦合，并且

11.4.聚类

0.5

1

1.5

2

2.5

−1.5

−1

−0.5

0

0.5

1

1

2

3

4

5

q

u

1

u

2

图11.6.五个文档和查询投影到前两个左奇异向量的坐标系中。

这些词是文档集的主要内容（参见单数值）。这显示在U1的组成中。因此，即使查询中没有一个词与文档1匹配，该文档也与主导方向紧密相关，因此它在简化表示中变得相关。

11.4聚类

在文档集合的情况下，很自然地假定存在内容相似的文档组。如果我们将文档看作是RM中的点，那么我们可以将组可视化为集群。通过平均值即质心来表示每个簇，我们可以根据质心来压缩数据。因此，例如，使用k-均值算法的聚类是术语文档矩阵的低阶近似的另一种方法。集群在信息检索中的应用在[30，76，77]中进行了描述。

与LSI类似，质心（归一化但非正交）的矩阵ck∈rm×k可作为“文档空间”中的一个近似基，查询匹配则需要在此基础上确定所有文档的坐标。这可以通过求解矩阵最小二乘问题来实现。

.

然而，首先对C的柱进行正交化比较方便，即计算C的二维分解。

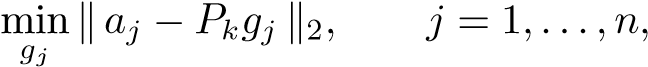
ck=pkr，

解决

pk∈rm×k，r∈rk×k，

（11.5）

单独编写−pkgk的每一列，我们看到这个矩阵最小二乘问题等价于n个独立的标准最小二乘问题。



其中，gj是gk中的j列。由于pk有正态柱，我们得到gj=pktaj，（11.5）的解为



为了匹配查询q，我们计算产品

，

其中qk=pktq。因此，低维近似中的余弦是

.

例11.9。我们对medline集合的q9进行了查询匹配。在计算聚类之前，我们将列标准化为相等的欧几里得长度。我们使用正交归一化的质心将矩阵从一个聚类中近似为50个聚类。召回精度图如图11.7所示。我们发现，对于高回忆值，质心法和等级加倍的LSI法一样好；见图11.4。

对于等级50，质心法中的近似误差，

，

甚至比100级的LSI还要高。

改进后的性能可以用与LSI类似的方式来解释。作为集群的“平均文档”，质心捕获集群中主要文档之间的主要链接。通过用形心表达所有文档，强调了主要的链接。

当我们测试Medline集合中的所有30个查询时，我们发现秩等于50的质心方法与秩100的LSI具有相似的性能：有些查询的全向量空间模型更好，但也有一些查询的质心方法更好。好多了。

11.5.非负矩阵分解

0

20

40

60

80

100

0

10

20

30

40

50

60

70

80

90

100

Recall (%)

Precision (%)

图11.7.查询Q9的匹配。全矢量空间模型（实线）和秩50质心近似（实线和圆）的回忆与精度。

11.5非负矩阵分解

假设我们已经计算出了术语文档矩阵的近似非负矩阵分解，

a≈w h，w≥0，h≥0，

式中，h的w∈rm×k和h∈rk×n。h的j列在由w列组成的近似非正交基中保持j文件的坐标。我们首先要通过求解最小二乘问题min来确定在同一基中查询向量q的表示。然后，在这里在此基础上，我们计算查询和所有文档向量之间的角度。考虑到w的薄qr分解，

W=qr，P∈Rm×K，R∈Rk×K，

缩减基础中的查询是

Q\_=R−1qtq，

文件j的余弦是

.

0

20

40

60

80

100

0

10

20

30

40

50

60

70

80

90

100

Recall (%)

Precision (%)

图11.8.查询Q9的匹配。全向量空间模型（实线）和秩50非负矩阵近似（虚线和×）的回忆与精度。

例11.10。我们使用中描述的100次乘法算法迭代计算了medline termdocument矩阵的秩50近似值。

第9.2条相对近似误差约为0.89。查询q9的调用精度曲线如图11.8所示。

11.6 LGK标准化

到目前为止，在本章中，我们已经描述了三种改进信息检索向量空间方法的方法，即通过基于SVD、聚类和非负矩阵分解的低阶近似表示术语文档矩阵A。这三种方法有一个共同的缺点：当添加或删除新文档时，更新低阶近似值的成本很高。在第7章中，我们描述了一种计算与最小二乘问题有关的低阶近似值的方法，在lgk双向化中使用右手边作为起始向量。在这里，我们将把这种方法应用于文本挖掘问题，这样就可以为每个查询计算一个新的低阶近似值。因此，在更改文档收集时不需要额外的计算成本。另一方面，每个查询匹配的工作量会变大。本节的灵感来源于[16]。

给定一个查询向量Q，我们应用递归的LGK双向化算法（或PLS）。

11.6.LGK标准化

### 查询Q的lgk bidialogization

1. β1p1=q，z0=0
2. 当i=1:kαizi=atpi−βizi−1βi+1pi+1=azi−αipi时

### 三。结束

确定系数αi−1和βi，以便

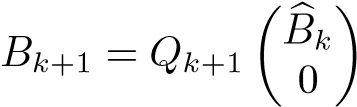
向量pi和zi收集在矩阵中，并且

. 在这个过程的k个步骤之后，我们生成了一个k级

近似值；见（7.16）。我们在下面的命题中总结了推导过程。

提案11.11.让azk=pk+1bk+1是lgk的k步的结果。

递归，并让



是双矩阵bk+1的qr分解。那么我们有一个秩k近似值

，（11.6）

哪里

.

使用低秩近似（11.6）进行查询匹配的方法与LSI中使用SVD低秩近似的方法大致相同。然而，如果使用以下方法，结果表明[16]获得了更好的性能。

wk的列向量是接近查询q的文档的正交近似基础。我们现在选择根据该基础计算查询的投影，而不是根据该基础计算a的列的坐标：

.

然后我们用余弦度量

，（11.7）

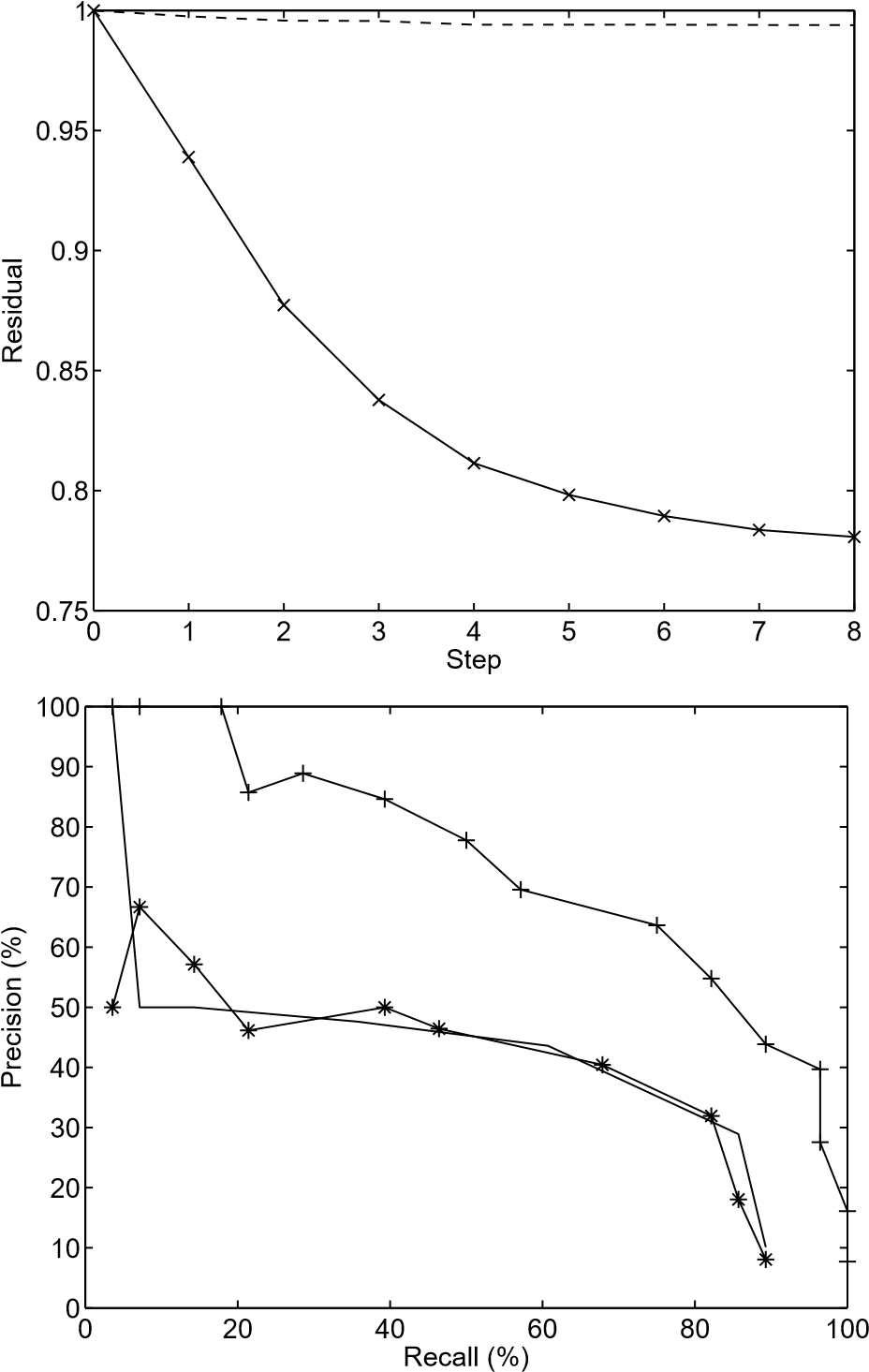


图11.9.查询Q9的匹配。上图显示了相对残差，它是LGK双向化（实线和×）步骤数的函数。作为比较，给出了基于第一奇异向量（主成分回归）的残差（虚线）。在底图中，我们给出了全矢量空间模型（实线）的回忆与精度，以及两步（实线和+s）和八步（实线和s）的二元化。

也就是说，我们计算投影查询和原始文档之间角度的余弦。

例11.12。我们以q9作为起始向量进行了lgk双向化。结果表明，相对残差下降相当缓慢，略低于0.8；见图11.9。不过，经过两个步骤，该方法已经给出了

11.7.平均性能

比全矢量空间模型要好得多。也可以看出，八个步骤的双诊断结果更差。

实例11.12表明，由LGK双标准化得到的低阶近似在降噪方面具有与LSI和基于形心的方法相似的性质。值得注意的是，当查询向量用于影响第一个基向量时，较低的秩（在本例中为2）会给出几乎相同的检索结果。另一方面，当步数增加时，精度变差，接近全矢量空间模型。这是自然的，因为低阶近似逐渐变得更好，经过大约8个步骤后，它几乎代表了termdocument矩阵中与查询相关的所有信息，从全向量空间的意义上来说。

模型。

为了确定应该执行多少个递归步骤，可以监视最小二乘残差：在哪里

；

参见（7.14）。当残差范数大幅减少时，我们可以假设查询是由文档的线性组合来表示的。然后我们可以期望性能与全向量空间模型的性能大致相同。因此，为了比全向量空间模型有更好的性能，应该在残差曲线开始变平之前停止。

11.7平均性能

应在多个测试集上进行比较不同信息检索方法的实验。此外，一个人不仅应该使用一个查询，还应该使用一系列查询。

例11.13。我们测试了medline集合中的30个查询，并计算了本章介绍的五种方法的平均精度召回曲线。为了计算要比较的方法的平均精度，有必要

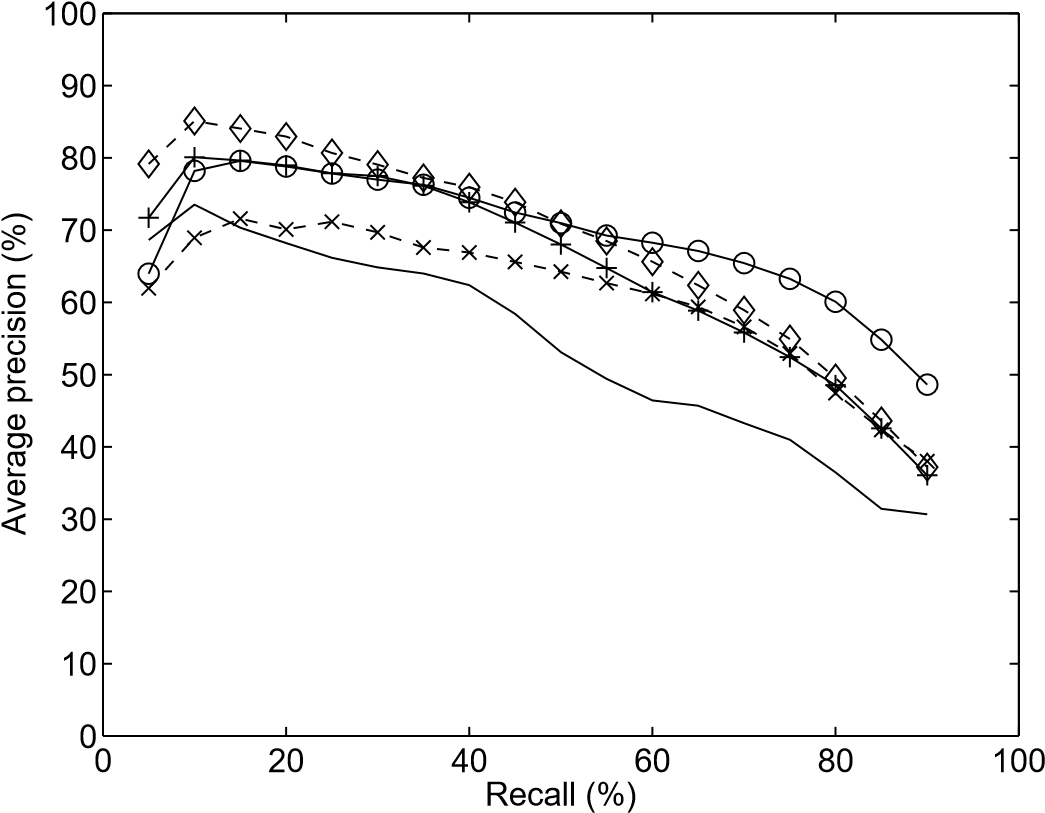


图11.10.查询匹配Medline集合中的所有30个查询。所采用的方法有：全矢量空间法（实线）、100阶LSI（虚线和菱形）、50阶形心近似（实线和圆）、50阶非负矩阵分解（虚线和×s）、LGK二级化（实线和＋s）。

以特定的召回值，比如5、10、15、…，90%对其进行评估。我们通过线性插值得到这些。

结果如图11.10所示。结果表明，与全向量空间模型相比，100阶LSI、50阶形心近似、非负矩阵因子分解和LGK二值化两个步骤的平均精度都有较大提高。

从示例11.13中我们看到，对于medline测试集合，基于术语文档矩阵的低阶近似的方法都比全向量空间模型表现得更好。当然，要支付的价格是更多的计算。在LSI、形心近似和非负矩阵分解的情况下，额外的计算可以离线进行，即与查询匹配分离。如果将文档添加到集合中，则必须重新计算近似值，这可能会很昂贵。另一方面，该方法基于LGK双向化，执行与查询匹配相关的额外计算。因此，它可以在术语文档矩阵经常发生变化的情况下有效地使用。

其他测试集合也得到了类似的结果；参见，例如[11]。然而，文本文档的结构起着作用。例如，在[52]中，表明LSI对医学文摘的性能比《时代》杂志的文章要好得多。

第十二章

网页搜索引擎的网页排名

当使用搜索引擎在互联网上进行搜索时，首先有一个传统的文本处理部分，其目的是查找包含查询词的所有网页。由于网络规模巨大，点击量可能太大，无法使用。因此，需要一些质量度量来过滤掉那些不那么有趣的页面。

当使用Web搜索引擎时，搜索短语通常是未指定的。

例12.1。2005年9月29日，谷歌使用“大学”这一搜索词组进行了一次搜索，结果链接到以下知名大学：哈佛大学、斯坦福大学、剑桥大学、耶鲁大学、康奈尔大学、牛津大学。与搜索短语相关的网页总数超过20亿。

显然，谷歌使用了一种算法来对所有网页进行排名，这种算法相当符合常识性的质量标准。有点令人惊讶的是，排名过程不是基于人类的判断，而是基于网络的链接结构。不严格地说，如果一个网页有来自其他高排名网页的链接，谷歌会给它分配一个高排名。我们将看到这个自引用语句可以用数学公式表示为某个矩阵的特征值方程。

12.1页面排名

当然，要定义一个对搜索引擎的所有用户都可以接受的普遍有效的相关性度量是不可能的。谷歌使用pagerank的概念作为网页质量的衡量标准。它基于这样一个假设，即一个页面的链接数量提供了关于页面重要性的信息。我们将主要基于[74]和[33]对pagerank进行描述。关于谷歌，请参见[19]。

一百四十七

让所有网页从1到n排序，让我成为一个特定的网页。然后oi将表示我链接到的一组页面，即大纲。大纲的数量表示为ni= oi。inlink的集合（表示为i i）是大纲为i的页面。

*i*

*I*

*i*

*O*

*i*

一般来说，我可以认为一个页面越重要，它的链接越多。然而，仅基于链接数量的排名系统很容易操作：当您设计一个网页时（例如，出于商业原因），您希望被尽可能多的用户看到，您可以简单地创建大量（无信息和不重要）页面，这些页面具有u链接到i。为了阻止这种情况，我们定义了i的等级，这样，如果一个高等级的页面j有一个大纲，这就增加了i的重要性，如下所示：页面i的等级是大纲到i的页面等级的加权和。权重是这样的，即第J页在其大纲中平均划分。把这个转化成数学，我们得到

（12.1）

这个初步定义是递归的，因此不能直接计算pageranks。相反，可以使用定点迭代。猜测一个初始排名向量r0。然后迭代

（12.2）

这种迭代有一些问题：如果一个页面没有大纲，那么在迭代过程中，它只通过其inlink累积排名，但是这个排名永远不会进一步分布。因此，不清楚迭代是否收敛。我们稍后再讨论这个问题。

如果我们将（12.1）重新表述为表示互联网图形的矩阵的特征值问题，将获得更多的见解。设q为n维的平方矩阵。定义

qij=1美元/nj，如果有从j到i的链接，则为0。

12.1.页面排名

这意味着第i行在与i的inlink对应的位置上有非零元素。同样，第j列在与j的outlines对应的位置上也有非零元素等于nj，并且，如果页面有outlines，第j列中所有元素的和等于1。在矩阵q的以下符号图中，非零元素表示为：

J

链接

大纲

例12.2。下面的链接图说明了一组带有大纲和inlink的网页：

1

2

3

4

5

6

相应的矩阵变成

.

由于第4页没有大纲，相应的列等于零。

显然，定义（12.1）等价于第一行的标量积和向量r，后者拥有所有页面的排名。我们可以用矩阵形式写出方程，

λr=qr，λ=1，（12.3），即r是特征值为λ=1的q的特征向量。现在很容易看出迭代（12.2）等于

r（k+1）=qr（k），k=0,1，…，

这是计算特征向量的幂次方法。然而，目前还不清楚pagerank是否定义得很好，因为我们不知道是否存在一个等于1的特征值。结果表明，马尔可夫链理论在

分析。

12.2随机行走和马尔可夫链

对pagerank概念有一个随机的遍历解释。假设一个浏览网页的浏览者在相同概率的大纲中选择下一个网页。然后，随机游动产生一个马尔可夫链（参见，例如[70]）。马尔可夫链是一个随机过程，其中下一个状态完全由当前状态决定；该过程没有内存。马尔可夫链的转移矩阵是qt。（注意，我们使用的符号与随机过程理论中常见的符号稍有不同。）

随机冲浪者不应该被卡住。换句话说，我们的随机行走模型应该没有没有没有大纲的页面。（这样一个页面对应于q中的零列），因此模型被修改，以便在所有位置用一个常量值替换零列。这意味着进入任何其他互联网页面的概率是相等的。定义向量

如果nj=0，dj=1，

否则为0，

对于j=1，…，n和

（12.4）

定义了修改后的矩阵

（12.5）

通过这种修改，矩阵p是一个合适的列随机矩阵：它有非负元素，每列元素的总和为1。前面的陈述可以重新表述如下。

12.2.随机游动和马尔可夫链

提案12.3.一列随机矩阵p满足

ETP=ET（12.6）

其中e由（12.4）定义。

例12.4.上一个示例中的矩阵修改为

.

与（12.3）类似，我们将pagerank向量定义为特征值为1的p的唯一特征向量，

pr=r。

过渡矩阵的特征向量对应于马尔可夫链的平稳概率分布。位置i，ri中的元素是经过大量步骤后，随机步行者在网页i上的概率。但是，特征值为1的唯一特征值的存在仍然没有保证。为了确保唯一性，矩阵必须是不可约的；参见[53]。

定义12.5.如果存在置换矩阵p，则称为可约矩阵a。

，（12.7）

其中x和z都是正方形。否则，矩阵称为不可约矩阵。

例12.6。为了说明可约性的概念，我们给出了一个对应于可约矩阵的链接图的例子：

1

4

5

2

3

6

一个进入链接图左侧的随机步行者永远不会离开它，同样也会被困在右侧。对应的矩阵是

，（12.8）

其形式为（12.7）。实际上，这个矩阵有两个特征值等于1，一个等于−1；参见示例12.10。

对应于不可约矩阵的有向图是强连通的：在图中任意两个节点（ni，nj），都存在一条从ni到nj的路径。

不可约正矩阵的最大特征值的唯一性是由佩伦-弗罗贝尼乌斯定理所保证的；我们在这里讨论的特殊情况下说明它。不等式a>0被理解为a的所有元素都是严格正的。用显性特征值表示最大的特征值，即λ1。

定理12.7。设A为不可约列随机矩阵。主特征值λ1等于1。有一个唯一的对应特征向量r满足r>0；这是唯一非负的特征向量。如果a>0，则i<1，i=2,3，…，n。

证据。因为a是列随机的，所以我们有eta=et，这意味着1是a的特征值。其余的陈述可以用Perron-Frobenius理论证明[70，第8章]。

考虑到互联网的规模，我们可以确定链接矩阵p是可约化的，这意味着p的pagerank特征向量没有很好的定义。为了确保不可约性，也就是说，为了使随机步行者不可能被困在子图中，人们人为地添加了从每个网页到所有其他网页的链接。在矩阵项中，这可以通过取p和秩1矩阵的凸组合来实现。

，（12.9）

某些α满足0≤α≤1。很容易看出矩阵A是列随机的：

.

12.2.随机游动和马尔可夫链

附加排名1项的随机行走解释是，在每个时间步中，浏览页面的冲浪者将跳到概率为1−α的随机页面（有时称为远程传输）。

我们现在看到矩阵A的pagerank向量定义得很好。

提案12.8.（12.9）中定义的柱随机矩阵a是不可约的（因为a>0），并且具有显性特征值λ1=1。相应的特征向量r满足r>0。

对于数值特征值算法的收敛性，必须知道P的特征值是如何被秩1修正（12.9）所改变的。

定理12.9.假设柱随机矩阵p的特征值为1，λ2，λ3…，λn。那么a=αp+（1-α）n1 eet的特征值为1，αλ2，αλ3，…，αλn。

证据。将\_e定义为e归一化为欧几里得长度1，并让u1∈rn×（n−1）是正交的。那么，既然\_etp=e\_t，

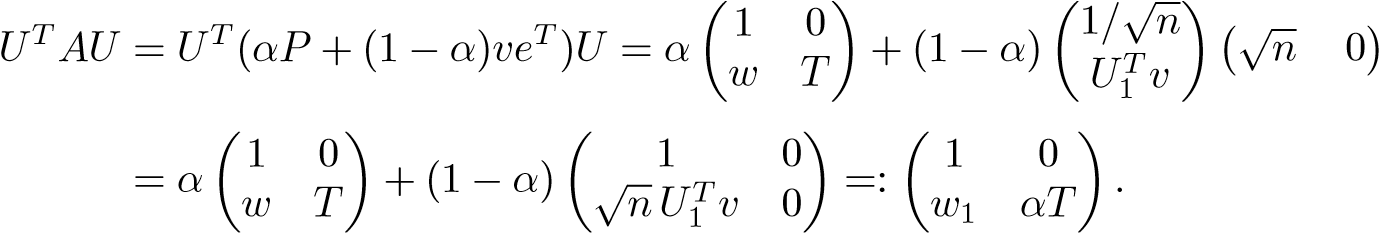
（12.10）

哪里。因为我们做了一个相似变换-

另外，矩阵t的特征值为λ2，λ3，…，λn。

.

因此，



现在声明立即生效。

定理12.9表明，即使p的多重特征值等于1（谷歌矩阵实际上就是这样），a值的第二大特征值始终等于α。

示例12.10。我们用（12.8）中的p和α=0.85计算矩阵的特征值和特征向量。matlab代码lp=eig（p）'；e=one（6,1）；

a=0.85\*p+0.15/6\*e\*e'；

[r，l]=eig（a）得出以下结果：

lp=-0.5 1.0-0.5 1.0-1.0 0 0

R=0.447-0.365-0.354 0.000 0.817 0.101

0.430-0.365 0.354-0.000-0.408-0.752

0.430-0.365 0.354 0.000-0.408 0.651

0.057-0.000-0.707 0.000 0.000-0.000

0.469 0.548-0.000-0.707 0.000 0.000

0.456 0.548 0.354 0.707-0.000-0.000

诊断（L）=1.0 0 0.85-0.0-0.85-0.425-0.425

可以看出，第一个特征向量（对应于特征值1）是唯一的非负特征向量，如定理12.7所述。

而不是修改（12.9），我们可以定义

A=αp+（1−α）vet，

其中v是一个非负向量，其值为1，可以选择使搜索偏向于某些类型的网页。因此，它有时被称为个性化向量[74，48]。矢量V还可以用来避免被所谓的链路农场操纵[57，59]。

12.3 pagerank计算的幂法

我们要解决特征值问题

ar=r，

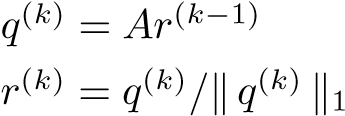
其中，r标准化=1。在本节中，我们用T1表示所寻求的特征向量。对于概率分布的随机矩阵和向量，自然使用向量的1-范数（第2.3节）。由于a的稀疏性和维数（几十亿阶），使用第15章中描述的密集矩阵的任何标准方法计算特征向量都是不可能的，因为这些方法是基于对矩阵应用正交变换。目前唯一可行的方法是幂法。

假设给出了初始近似值r（0）。下面的算法给出了幂法。

12.3.pagerank计算的幂法

### ar=λr的幂法

对于k=1,2，…直到收敛



归一化向量（使其1-范数等于1）的目的是避免向量变得非常大或非常小，从而在浮点系统中不可呈现。稍后我们将看到，在pagerank计算中不需要规范化。在这种情况下，不需要计算特征值近似值，因为已知所寻求的特征值等于一。

功率法的收敛性取决于特征值的分布。为了使表示更简单，我们假设a是可对角化的，即存在特征向量的非奇异矩阵t，t−1at=diag（λ1，…，λn）。特征值λi的阶数为1=λ1>λ2≥·····≥λn。根据特征向量展开初始近似值r（0）。

r（0）=c1t1+c2t2+····+cntn，

其中，假设=0，r=T1是所寻求的特征向量。然后我们有了

.

显然，因为j=2,3，…我们有λj\_<1，第二项趋于零，幂法收敛到特征向量r=T1。收敛速度由λ2确定。如果接近1，则迭代非常慢。幸运的是，谷歌矩阵并非如此；参见定理12.9及以下。

利用特征值问题的残差向量，给出了幂次迭代的停止准则。设λ\_为特征值的计算近似值，\_r为相应的近似特征向量。然后可以看出[94]，[4，p.229]最佳误差矩阵e，其中

（a+e）r\_=λ\_r，\_

没错，满足

，

其中s=ar\_−λ\_r\_。这意味着，如果残差很小，则计算出的近似特征向量\_r是接近a的矩阵a+e的精确特征向量。因为在pagerank计算中，我们处理的是一个正矩阵，其所有列加起来都是一个，所以使用1-范数是很自然的。[55。因为1-范数和欧几里得范数是等价的（参见（2.6）），所以这没有什么区别。

在通常的幂法公式中，向量被归一化以避免下溢或溢出。我们现在证明，当矩阵是列随机的时候，这是不必要的。

提案12.11.假设向量z满足

矩阵A是列随机的。然后

（12.11）

证据。输入y=az。然后



因为a是随机列（eta=et）。

鉴于google矩阵的巨大维度，计算

矩阵向量积y=az，其中。回想一下，p是根据实际的链接矩阵q构造的

，

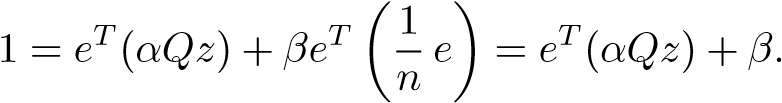
其中，行向量d在所有与没有大纲的网页相对应的位置都有元素1（见（12.5））。这意味着要形成p，我们要在q中插入大量的全向量，每个向量的维数都与总网页数相同。因此，我们负担不起显式存储p。让我们更详细地看一下乘法y=az：

（12.12）

哪里

β=αdtz+（1−α）etz。

我们不需要用这个方程来计算β。相反，我们可以将（12.11）与（12.12）结合使用：



因此，我们有。另外一个额外的好处是我们根本不使用向量d，也就是说，我们不需要知道哪些页面缺少大纲。

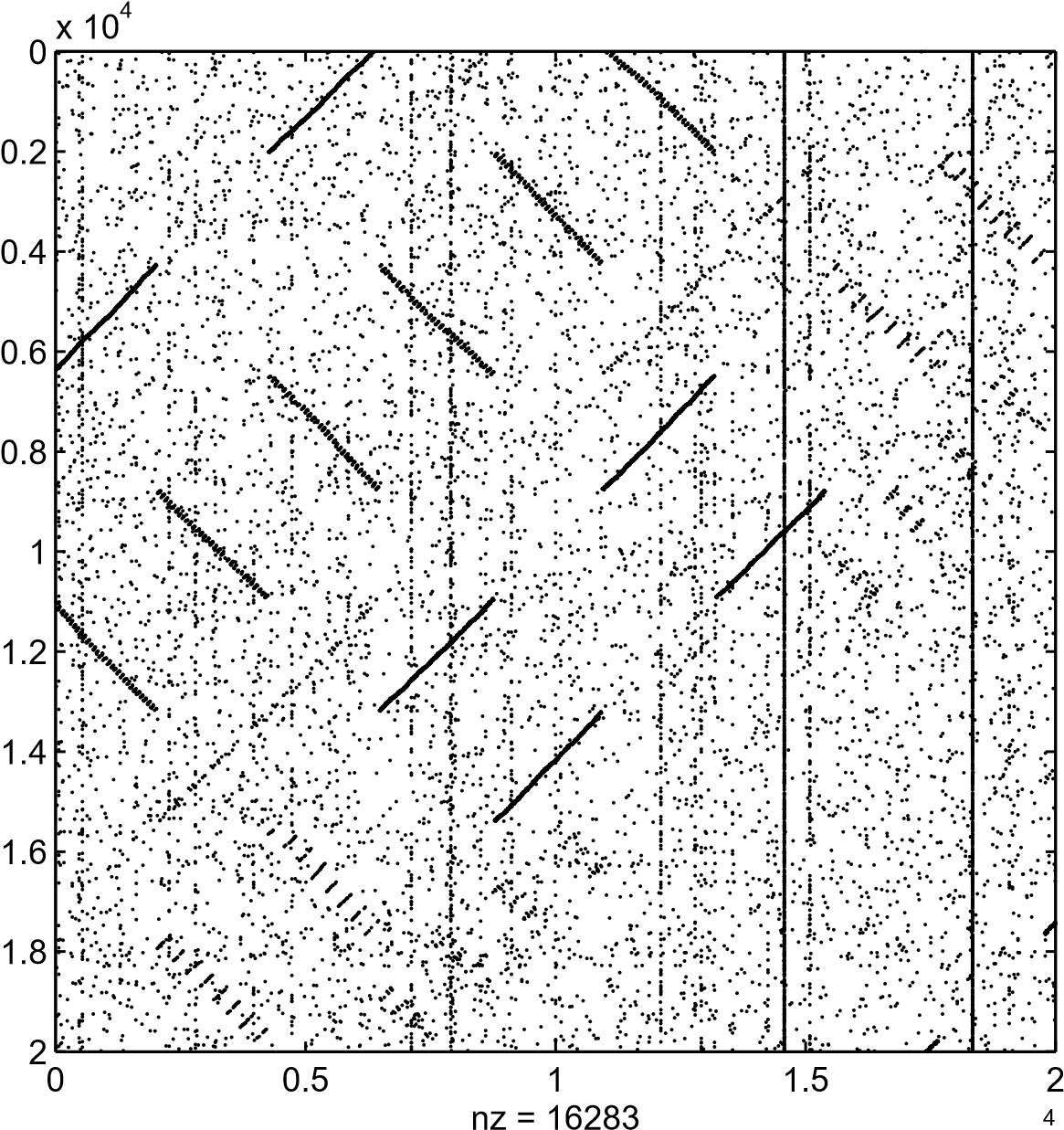
以下matlab代码实现矩阵向量乘法：

12.3.pagerank计算的幂法

yhat=alpha\*q\*z；beta=1-范数（yhat，1）；y=yhat+beta\*v；residual=norm（y-z，1）；

这里v=（1/n）e，或者一个个性化的远程传送向量；参见第154页。为了节省内存，我们甚至应该避免使用额外的向量yhat并用y替换它。

根据定理12.9，我们知道google矩阵的第二特征值满足λ2=α。α的典型值为0.85。大约需要K=57次迭代，使系数0.85K等于10-4。据报道[57]接近谷歌使用的迭代次数。



X 10

图12.1.斯坦福.edu矩阵的20000×20000子矩阵。

例12.12。作为一个例子，我们使用了从stanford.edu域获得的矩阵p。页数为281903页，链接总数为2312497。矩阵的一部分如图12.1所示。我们计算了pagerank

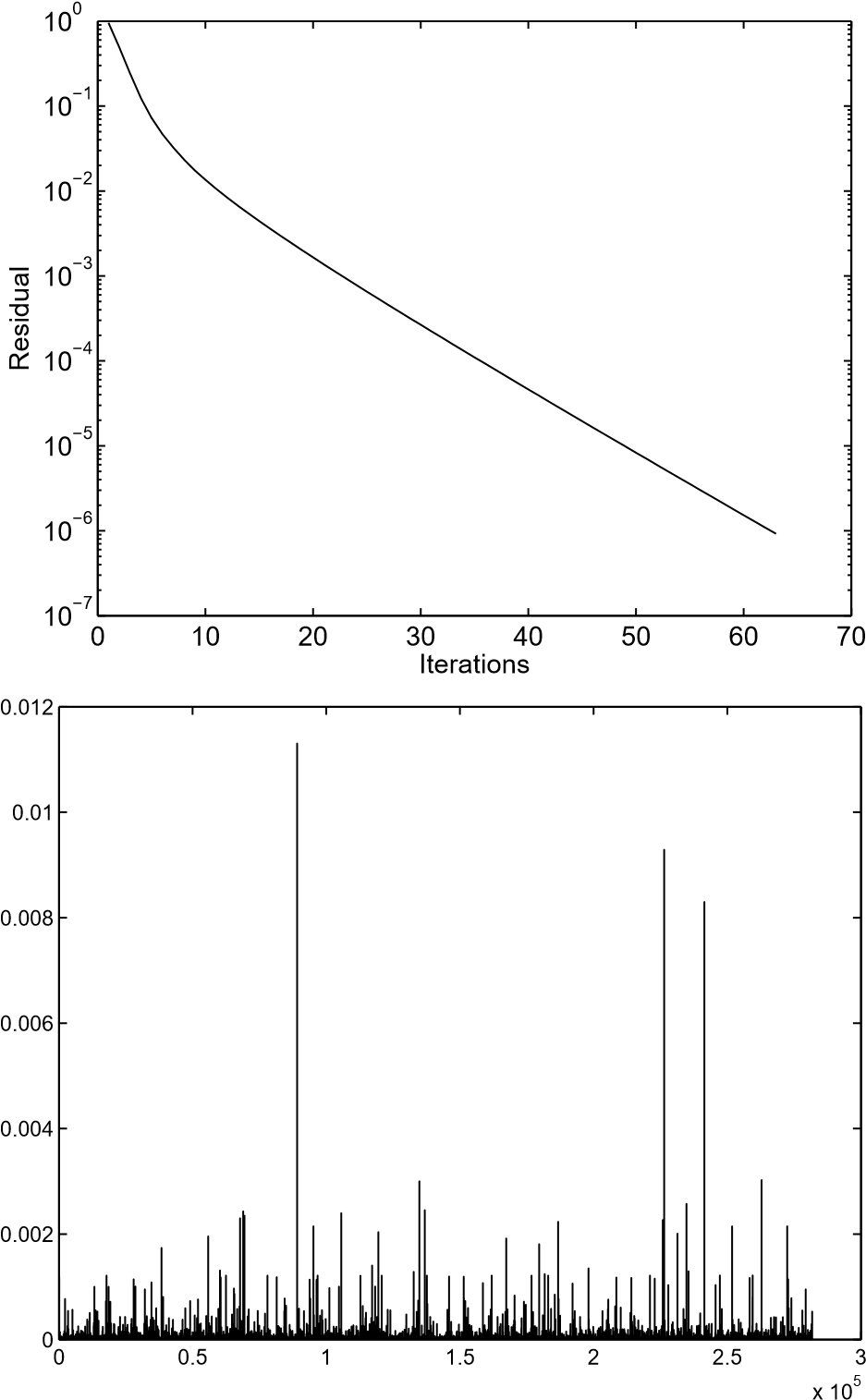


图12.2.stanford.edu矩阵的幂迭代（top）和pagerank向量（bottom）中的余数。

向量使用功率法，α=0.85，迭代63次，直到残差的1-范数小于10-6。残差和最终pagerank向量如图12.2所示。

由于一个pagerank计算可能需要几天时间，因此已经提出了迭代过程的一些改进。在[53]中，描述了一种自适应方法，该方法检查pagerank向量组件的收敛性，并避免对这些组件执行幂迭代。据报道，加速率高达30%。在[54]中使用了Web的块结构，并且报告了系数2的加速。基于Aitken-Extrapo的加速方法-

12.4.点击量

相关描述见[55]。Langville和Meyer的几篇论文以及[51]中讨论了聚合方法。

当计算互联网子集的pagerank时，例如，一个特定的域，矩阵p可以是一个维度，对于该维度，可以使用除幂法以外的方法，例如阿诺迪方法；见[40]和第15.8.3节。甚至可以使用matlab函数eigs来计算少量的特征值和稀疏矩阵的相应特征向量，使用arnoldi方法重新启动。

在[44]中提出了pagerank的变体。pagerank矩阵的其他属性在[84]中给出。

12.4点击量

另一种基于Web链接结构的方法与pagerank[56]同时介绍。它被称为hits（超文本诱导的主题搜索），基于权限和中心的概念。权威是一个有许多链接的网页，而中心有许多大纲。基本思想是好的中心指向好的权威，好的权威指向好的中心。每个网页都被分配一个中心分数Y和一个权威分数X。

设l为有向网络图的邻接矩阵。然后给出两个方程式，根据基本思想，用数学方法定义两个分数之间的关系：

x=lty，y=lx.（12.13）

计算分数的算法是幂法，它收敛到L的最大奇异值对应的左右奇异向量，在hits的实现中，使用的邻接矩阵不是整个web的邻接矩阵，而是与查询相关的所有页面的邻接矩阵。

现在有大量关于pagerank、hits和其他排名方法的文献。有关概述，请参见[7、58、59]。在[65]中提出了点击率和pagerank的组合。

显然，pagerank和hits的基础思想不仅限于Web应用程序，而且可以应用于其他网络分析。最近，一项关于最高法院判例的研究中使用了hits方法的变体[36]。hits在[17]中被概括，它也处理同义词提取。在[72]中，genarnk（基于pagerank概念）被用于微阵列实验的分析。

第十三章

自动关键字和关键字句子提取

由于文本信息量的激增，需要开发自动的文本摘要程序。一种典型的情况是，Web搜索引擎从每个文档中显示少量与特定查询匹配的文本。另一个相关领域是新闻文章的总结。

自动文本摘要是一个活跃的研究领域，它与信息检索、自然语言处理和机器学习等多个领域有着密切的联系。非正式地，文本摘要的目标是从文本文档中提取内容，并以简明的形式和对用户或应用程序的需要敏感的方式向用户呈现最重要的内容[67]。在本章中，我们将有一个不那么雄心勃勃的目标：我们将介绍一种从文本中自动提取关键词和关键句子的方法。信息检索中的向量空间模型和pagerank概念之间存在联系。我们还将使用非负矩阵分解。本演示基于[114]。使用QR分解的文本摘要如[26，83]所述。

13.1显著性得分

考虑从中提取关键词和关键句子的文本。我们将以本书第12章为例。作为预处理步骤之一，应该执行词干化，这样具有不同词尾的相同词干只由一个标记表示。停止词（参见第11章）经常出现在文本中，但由于它们不区分不同的句子，所以应该删除它们。同样，如果文本带有特殊符号，例如数学或标记语言标记（HTML、LaTex），则可能需要删除这些符号。

由于我们想比较不同句子中的词频，我们必须将每个句子视为单独的文档（在信息检索术语中）。预处理完成后，我们使用与信息检索中相同类型的解析器来解析文本。这样，术语文档矩阵就是

一百六十一

准备好了，在本章中我们将把它称为一个词-句矩阵。因此，我们有一个矩阵a∈rm×n，其中m表示不同项的个数，n表示句子的个数。元素aij被定义为句子j中词条i的频率。

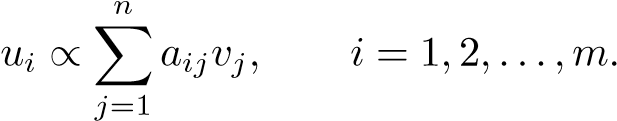
列向量（a1j a2j…在与句子j中出现的术语相对应的位置上，t不是零。同样，行向量（ai1 ai2…ain）在与包含i项的句子相对应的位置处为非零。

[114]中程序的基础是术语和句子同时但独立的排序。因此，术语i给出非负显著性得分，表示ui。显著性得分越高，这个词就越重要。句子j的显著性得分表示vj。

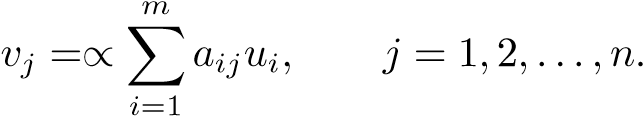
显著性得分的分配基于相互强化原则[114]：

如果一个词出现在许多句子中，并且有很高的显著性得分，那么它应该有很高的显著性得分。如果一个句子中包含了许多具有高显著性得分的单词，那么它应该具有高显著性得分。

更准确地说，我们认为第一项的显著性得分与出现它的句子的得分之和成正比；此外，每一项都由相应的矩阵元素加权，



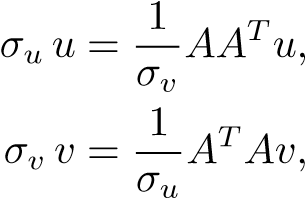
同样，句子j的显著性得分被定义为与其词的得分成正比，并由相应的aij加权。



收集两个向量u∈rm和v∈rn的显著性得分，这两个方程可以写成

|  |  |
| --- | --- |
| σuu=平均值， | （13.1） |
| σvv=atu， | （13.2） |

其中，σu和σv是比例常数。实际上，常数必须相等。把一个方程插入另一个方程，我们得到



13.1.显著性得分

结果表明，U和V分别是AAT和ATA的特征向量，具有相同的特征值。因此，u和v是对应于相同奇异值的奇异向量。

如果我们选择最大的奇异值，那么我们就可以保证u和v的分量是非负的。

总之，术语的显著性得分被定义为u1的组成部分，句子的显著性得分是v1的组成部分。

0

50

100

150

200

250

300

350

400

−0.1

0

0.1

0.2

0.3

0.4

0

2

0

4

0

60

80

1

00

12

0

14

0

1

60

1

80

2

00

−0.2

0

0.2

0.4

0.6

图13.1.第12章的显著性得分：学期得分（上）和句子得分（下）。

例13.1。我们根据第12章创建了一个术语句矩阵。由于文本是使用LaTex编写的，所以我们首先必须删除所有LaTex排版命令。这是使用一个名为detex.35的词汇扫描仪完成的，然后对文本进行词干处理并删除停止词。使用文本解析器tmg[113]构造了一个术语句矩阵A：183个句子中有388个术语。第一个奇异向量在matlab中计算，[u，s，v]=svds（a，1）。（矩阵是稀疏的，所以我们对稀疏矩阵使用SVD函数。）奇异向量如图13.1所示。

通过定位U1的10个最大组件并使用文本解析器生成的字典，我们发现以下按重要性排序的单词在本章中最重要：

|  |
| --- |
| 一 |

γ

图13.2.秩1近似的符号说明：.

网页，搜索，大学，网络，谷歌，排名，大纲，链接，数字，相等

以下是六个最重要的句子，顺序如下：

1. 2005年9月29日，谷歌使用搜索短语大学进行了一次搜索，结果链接到以下知名大学：哈佛大学、斯坦福大学、剑桥大学、耶鲁大学、康奈尔大学、牛津大学。
2. 当使用搜索引擎在互联网上进行搜索时，首先是传统的文本处理部分，目的是找到包含查询词的所有网页。
3. 不严格地说，如果一个网页有来自其他高排名网页的链接，谷歌会给它分配一个高排名。
4. 假设一个浏览网页的浏览者从相同概率的大纲中选择下一个网页。
5. 同样，J列的非零元素在与J的大纲对应的位置上等于NJ，并且，如果页面有大纲，则J列中所有元素的总和等于1。
6. 附加的排名1项的随机行走解释是，在每一个时间步骤中，访问页面的冲浪者将跳转到概率为1−α的随机页面（有时称为远程传输）。

显然，这种方法更喜欢长句。另一方面，这些句子无疑是课文的关键句子。

上述方法也可被视为术语句矩阵A的秩1近似值，如图13.2所示。

在这个解释中，向量u1是a列所跨越的子空间的基向量，行向量根据这个基持有a列的坐标。现在我们看到基于显著性得分的方法有一个缺点：如果有两个包含相同显著性项的“最高级句子”，那么它们的坐标将大致相同，并且两个句子将被提取为关键句子。这是不必要的，因为它们非常相似。接下来我们将看到，如果我们将关键句子提取建立在秩k近似的基础上，就可以避免这种情况。

|  |  |  |
| --- | --- | --- |
| 一 | γ |  |

13.2.从秩k近似中提取关键句子

图13.3.低阶近似的符号说明：a≈cd。

13.2从K级中提取关键句子

近似值

假设我们已经计算出了一个很好的条件句矩阵的秩k近似值，

a≈c d，c∈rm×k，d∈rk×n，（13.3）

如图13.3所示。这种近似可以基于SVD、聚类[114]或非负矩阵因式分解。选择的维度k大于或等于要提取的关键句子数。c是基向量的秩k矩阵，d的每一列根据基向量保持a中相应列的坐标。

现在回想一下，C中的基向量表示“句子空间”中最重要的方向，即列空间。然而，低阶近似并不能立即给出最重要的句子。如果我们首先确定A列的基础是“最重的”，即D列的最大2-范数，就可以找到这些句子。这定义了一个新的基向量。然后我们继续确定d列，它是剩余k-1基向量中最重的列，依此类推。

为了推导该方法，我们注意到在近似等式（13.3）中，我们可以在c和d之间引入任何非奇异矩阵t及其逆矩阵，并且我们可以在不改变关系的情况下，将该关系与右边的任何置换p相乘：

ap≈cdp=（ct）（t−1dp）

式中t∈Rk×k。

从近似值（13.3）开始，我们首先找到d中最大范数的列，并将其按p1排列到第一列，同时将a的对应列移动到第一位置。然后我们确定一个户主转换q1，它将元素（1,1）下方第一列中的元素归零，并将转换应用于c和d：

.

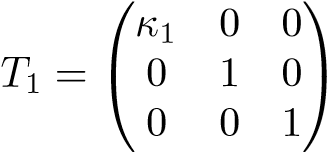
事实上，这是以d为支点的QR分解的第一步，我们以m=6，n=5，k=3的例子继续讨论。

为了说明该程序处理两个或多个非常相似的重要句子的问题（参见第164页），我们还假设d的第4列与移动到第一个位置的列具有几乎相同的坐标。

在第一步之后，矩阵具有结构

，

其中，κ是DP1第一列的欧几里德长度。由于第4列与现在位于位置1的列类似，所以在第2行和第3行中有小的条目。然后我们引入对角矩阵



各因素之间：

.

这只改变左因子的第1列和右因子的第1列。

（标有）。从关系中

ap1≈c1d1，

我们现在看到ap1中的第一列大约等于c1中的第一列。记住原矩阵C的列是矩阵A的主方向，我们现在已经确定了A的“主列”。

在继续之前，我们进行以下观察。如果D的一列与第一列（示例中的第4列）相似，那么它现在将在第一行下面具有小元素，并且在选择第二个最主要的文档时不会起到作用。因此，如果有两个或两个以上的重要句子的关键字或多或少相同，则只选择其中一个。

接下来，我们确定A的第二个最主要的列。为此，我们计算d1列的规范，不包括第一行（因为该行根据c1的第一列保存坐标）。列和

13.2.从秩k近似中提取关键句子

最大的标准移动到位置2，并通过类似于上述方式的户主转换减少。在这一步之后，我们

.

因此第二列

ap1p2≈c2d2

拥有第二大主宰地位。

继续这个过程，最终结果是

，

其中r是上三角形，p是排列的产物。现在，ap的前k列包含矩阵的支配列，这些列的秩k近似值在ck中。如果我们写的话这就更清楚了

，（13.4）

哪里

是原始C的旋转和缩放版本（即Rm中跨过相同子空间的列），它仍然保持A的主导方向。假设ai1、ai2、…、a i k是ap的前k列。则（13.4）等于

aij≈c\_j，j=1,2，…，k.

这意味着由C列给出的控制方向，

与a中的k列直接关联。

上述算法相当于通过柱旋转计算QR分解（见第6.9.1节）。

，

其中q是正交的，r是上三角形。注意，如果我们只想找到前k个句子，我们不需要对基向量矩阵进行任何转换，找到前k个句子的算法可以在matlab中实现，如下所示：

%c\*d是a的秩k近似值

[q，rs，p]=qr（d）；p=[1:n]\*p；

p k=p（1:k）；%ap前k列的索引

例13.2。我们使用第9.2节的乘法算法计算了例13.1中的术语句矩阵的非负矩阵因子分解。然后我们用上面描述的方法确定了六个最前面的句子。选择了实施例13.1中的句子1、2、3和5，以及以下内容

二：

1. 由于a的稀疏性和维数（几十亿阶），使用第15章中描述的密集矩阵的任何标准方法计算特征向量都是不可能的，因为这些方法是基于对矩阵应用正交变换。
2. 在[53]中，描述了一种自适应方法，该方法检查pagerank向量各分量的收敛性，并避免对这些分量执行幂迭代。

当使用SVD计算低阶近似值时，得到了相同的结果。

第十四章

张量支持向量机人脸识别

即使面部表情、光照、视角等发生变化，人类也非常善于识别面部。开发针对不同条件的鲁棒人脸识别自动程序是一个具有挑战性的研究问题，已使用几种不同的方法进行了研究。主成分分析（即SVD）是一种常用的技术，通常被称为“特征面”[23，88，100]。然而，当所有的照片都是在相似的条件下拍摄的时候，这种方法是最好的，当几个环境因素不同的时候，这种方法也不能很好地发挥作用。还研究了更通用的双线性模型；参见，例如[95]。

最近[102，103，104，105]研究了图像合成的多行分析方法。特别是人脸识别问题，采用张量模型，张量面方法。通过让张量的模式代表不同的观察条件，例如照明或面部表情，与PCA方法相比，可以提高识别算法的精度。

在本章中，我们将描述一种与张量面相关的人脸识别张量方法。由于我们处理的图像通常存储为m×n数组，m和n的顺序为100–500，因此要识别的每个面的计算量相当大。我们将讨论张量SVD（hosvd）如何也可用于降维以减少触发器计数。

14.1张量表示

假设我们有一组NP人员的图像，其中每个图像都是一个mi1×mi2数组，mi1mi2=ni。我们假设图像的列被叠加，这样每一个图像都由RN中的向量表示。进一步假设每个人的面部表情都不同。通常情况下，一个人的ni≥5000，而且通常ni比ne和np大得多。这个

一百六十九

图像集合存储为张量，

A∈RNI×NE×NP.（14.1）

我们分别将不同的模式称为图像模式、表达模式和人模式。

例如，如果我们也有不同照明度、视角等每个人的照片，那么我们可以用更高阶张量来表示图像采集[104]。为了简单起见，这里我们只考虑3模张量的情况。高阶张量的推广是直接的。

例14.1。我们对耶鲁大学人脸数据库中的10人图像进行了预处理，将每幅图像剪切并分解为112×78像素，存储在8736的矢量中。图14.1显示了五幅图像。



图14.1.有五种不同表达的人1（来自耶鲁大学面部数据库）。

每个人都有11种不同的表情。

当然，模式的顺序是任意的；为了明确和说明目的，我们假设（14.1）的顺序。然而，为了（某种程度上）强调顺序的任意性，将使用符号×e沿表达式模式将张量乘上矩阵，其他模式也同样如此。

我们现在假设并写出薄软管（见定理8.3和（8.9）），

A=S×I F×E G×P H，（14.2）

其中s∈renep×ne×n p是核心张量，p p f∈rni×nenp有正交柱，g∈rn×n和h∈rn×n是正交的。

例14.2。我们计算了10个人的面部图像张量，每个人有10种不同的表情。奇异值如图14.2所示。表达式和人模式中的10个奇异值都是显著的，这意味着表达式和人之间应该相对容易区分。

根据软管的用途，可以用不同的方式解释软管。我们首先说明这种关系

A=D×E g×P h，

14.1张量表示

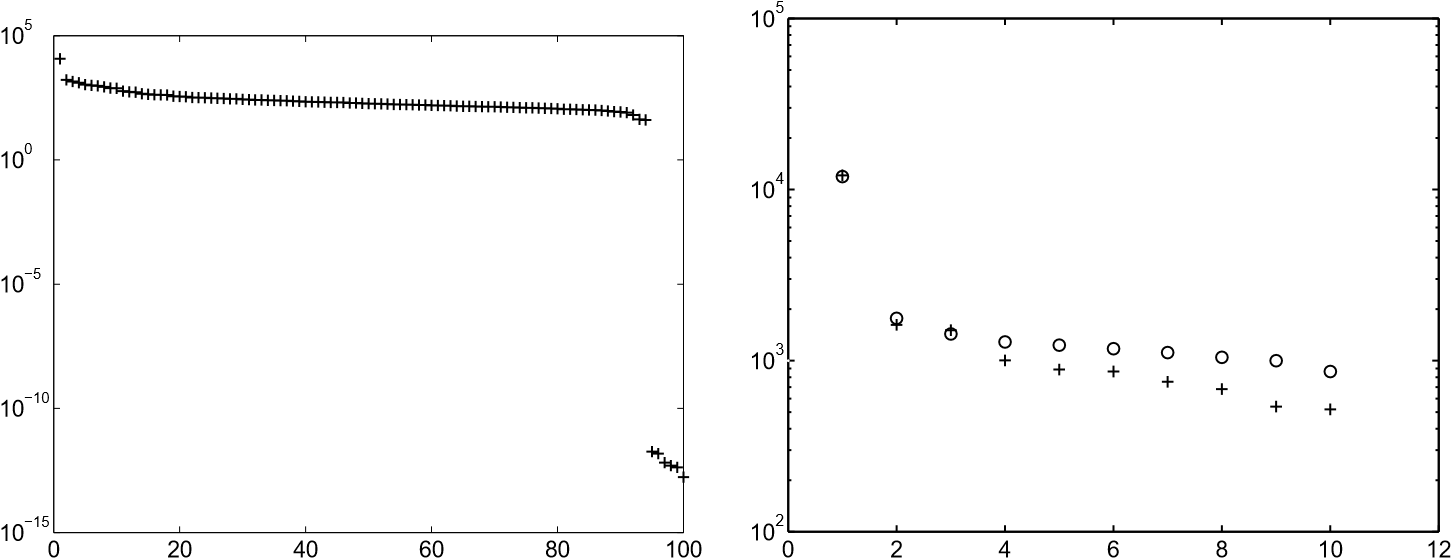


图14.2.图像模式（左）、表达式模式（右，+）和人员模式（右，圆）中的奇异值。

其中d=s×i f：

|  |
| --- |
| G |

我=

e

现在，让我们重述张量矩阵乘法的定义（第8.2节）。对于确定性，我们考虑2-模式，即这里的e-模式，乘法：

.

我们看到，固定表达式参数的一个特定值，也就是说，将j=e0对应于只使用g的e0行。通过在person模式中进行类似的选择，我们得到

a（：，e0，p0）=d×e ge0×p hp0，（14.3）

其中ge0表示g的e0行向量，hp0表示h的p0行向量。我们在下图中说明（14.3）：

A（：，e0，p0）

我们用文字概括如下：

通过将张量d与hp0和ge0在各自的模式下相乘，可以合成e0表达中p0人的图像。因此，人p0的独特特征是行向量hp0，表达e0的独特特征是ge0，通过双线性形式。

D×E G×P H.

例14.3。matlab代码

a=tmul（tmul（d，ue（4，：），2），向上（6，：），3）；

在表达式4中给出第6个人（快乐）；见图14.3。回想一下，函数tmul（a，x，i）将张量a乘以模式i中的矩阵x。



图14.3.表情4中的人6（快乐）。

14.2人脸识别

我们现在将考虑如下分类问题：

给定一个未知人的图像，用RNI中的向量表示，确定它代表的NP人中的哪一个，或者确定未知人不在数据库中。

14.2.人脸识别

对于分类，我们将hospd（14.2）写成以下形式：

A=C×P H，C=S×I F×E G.（14.4）

对于一个特殊的表达，我们有

A（：，E，：）=C（：，E，：）×P H.（14.5）

显然，我们可以用矩阵来识别张量a（：，e，：）和c（：，e，：），我们表示ae和ce。因此，对于所有的表达式，我们都有线性关系

ae=ceht，e=1,2，…，ne.（14.6）

注意，相同的（正交）矩阵h出现在所有的ne关系中。使用ht=，可以写入（14.6）的P列

（14.7）

我们可以解释（14.6）和（14.7）如下：

ae的p列包含表达式e中的person p的图像。ce的列是表达式e的基向量，h的p行（即hp）在此基础上保持person p图像的坐标。此外，同一个HP在所有表达式库中保存着人P的图像坐标。

接下来假设z∈rni是一个未知表达的未知人的图像（不在ne中），我们想对它进行分类。我们称Z为测试图像。显然，如果它是表达式e中人p的图像，那么z在这个基础上的坐标等于hp。因此，我们可以通过计算z在所有表达式基中的坐标，并检查每个表达式的z坐标是否与H的任何行的元素重合（或几乎重合），来对z进行分类。

通过求解最小二乘问题，可以求出表达式基e中z的坐标。

（14.8）

算法总结如下：

### 分类算法（初版）

|  |  |
| --- | --- |
| %Z是一个测试图像。对于e=1,2，…，ne  求解最小值 | . |
| 对于p=1,2，…，np  如果  结束 | 托尔，然后归类为P人，停下来。 |

该算法的工作量很大：对于每个测试图像z，我们必须用ce∈rni×np求解ne最小二乘问题（14.8）。

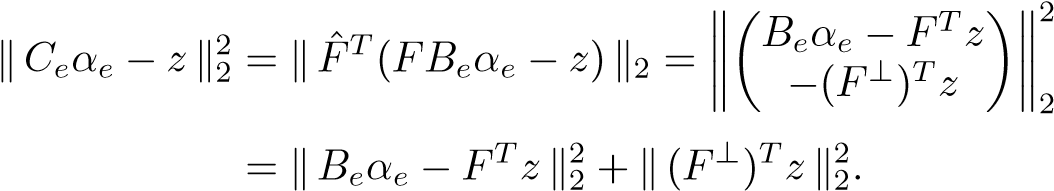
然而，从（14.4）中回忆，c=s×i f×e g，这意味着

ce=fbe，

其中bi e∈e prnenp×np是用（s×e g）（：，e，：）标识的矩阵。注意f∈rn×n n；我们假设ni比nenp大得多。然后，仅用于分析，放大矩阵，使其变为方形和正交：



现在在标准中插入f\_t：



因此，我们可以先计算ftz，然后求解ne最小二乘问题。

（14.9）

矩阵be的维数为nenp×np，因此求解（14.9）比（14.8）便宜得多。也可以预先计算每个矩阵的QR分解，以进一步减少工作量。因此，我们得出了以下算法。

### 分类算法

预处理步骤。计算并保存所有be矩阵的qr分解，be=qere，e=1,2，…，ne。

%Z是一个测试图像。

计算\_z=ftz。对于e=1,2，…，ne

求αe的reαe=qte z\_。

对于p=1,2，…，np

如果是TOL，则归类为P人并停止。

### 结束

14.3.带软管压缩的人脸识别

在一个典型的应用程序中，即使测试图像是数据库中某个人的图像，也可能使用另一个未在数据库中表示的表达式来获取。然而，上述算法在这种情况下工作良好，如[104]所述。

例14.4.在耶鲁大学的数据库中，每10个人都有一个眨眼的图像。我们将这些图像作为测试图像，并使用上面的算法计算出数据库中最接近的图像。在任何情况下



图14.4.上面一行显示要分类的图像，下面一行显示数据库中对应的最近的图像。

14.3带软管压缩的人脸识别

由于堆芯的有序性，对于不同的模式（定理8.3），我们可以截断堆芯，这样截断的hospd仍然是A的一个很好的近似值。对于我们假设的一些k值，定义fk=f（：，1:k），它比ni小得多，但比ni大得多。比NP。然后，仅用于分析，放大矩阵，使其变为方形和正交：



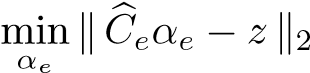
然后类似地截断核心张量，即

（14.10）

由定理8.3可知，E-模中G的乘法不影响I-模中的hospd排序性质。

.

因此，如果图像模式奇异值的衰减速度足够快，则尽管存在压缩，但仍有可能获得良好的识别精度。因此，如果我们在前面的算法中使用c，我们将不得不解决最小二乘问题。



具有明显的定义。现在，从（14.10）开始，我们在哪里。在范数符号内乘我们得到的

.

在识别算法的“压缩”变体中，操作\_z=ftz替换为\_，并且循环中的最小二乘问题是

更小。

例14.5。我们使用了与前一个示例相同的数据，但在图像模式中将正交基截断为k。当k=10时，所有测试图像都被正确分类，但当k=5时，10个图像中的2个被错误分类。因此，在本例中，在不牺牲分类精度的情况下，可以大幅降低等级（从100降到10）。

在我们的示例中，人数和不同的表达式太少，不需要进一步压缩数据。然而，在实际应用中，为了在合理的时间内对图像进行分类，可以在表达式和人称模式中截断核心张量，从而解决比未压缩情况下更小的最小二乘问题。

第三部分

计算矩阵

分解

第十五章

计算特征值和奇异值

在Matlab和其他现代编程环境中，特征值和奇异值是使用高级函数获得的，例如，eig（a）和svd（a）。这些函数实现来自lapack子程序库[1]的算法。为了确定高阶函数背后的方向，在本章中，我们简要介绍了计算稠密矩阵和大稀疏矩阵特征值和奇异值的一些方法。有关更广泛的治疗，请参见例如[4，42，79]。

函数eig和svd用于密集矩阵，即大多数元素为非零的矩阵。稠密矩阵的特征值算法有两个阶段：

1. 矩阵简化为紧致形式：对称情况下的三对角矩阵和非对称情况下的海森堡矩阵。这个相位由有限的正交变换序列组成。
2. 迭代还原为对角形式（对称情况）或三角形形式（非对称情况）。这是使用QR算法完成的。

对于大型稀疏矩阵，通常不可能（甚至有趣）计算所有特征值。这里有一些利用稀疏性的特殊方法。利用特征值算法的变化来计算奇异值。

作为背景材料，我们给出了关于特征值问题摄动理论的一些理论结果。此外，本文还简要介绍了计算特征值的幂次法及其相应的逆迭代法。

在线性代数教材中，常把矩阵a∈rn×n的特征值问题作为多项式方程的解引入。

Det（a−λi）=0。

在一般问题的计算解中，这种方法是无用的，有两个原因：（1）对于有意义的维数矩阵，计算行列式太昂贵；（2）即使行列式和多项式可以计算，特征值对pertu非常敏感。系数中的

一百七十九

多项式。相反，特征值数值计算的基本工具是正交相似变换。设V为正交矩阵。然后进行转换（对应于基础的更改）

A−→V TAV（15.1）

很明显，在这种变换下，特征值是保留的：

ax=λx v tav y=λy，（15.2）

其中y=v tx。

15.1扰动理论

计算特征值的QR算法基于正交相似变换（15.1），它计算一系列变换，最终结果是对角线（对于对称A）或三角形（对于非对称A）。由于该算法是迭代的，因此有必要确定浮点数何时足够小，以便在数值上被视为零。为了有一个良好的理论基础，这一决定，必须知道特征值和特征向量对数据的小扰动，即矩阵系数有多敏感。

关于特征值和奇异值灵敏度的知识也很有用，这是一个更基本的原因：矩阵元素通常是测量值，容易出错。灵敏度理论给出了在这种情况下我们能信任特征值等的信息。

在这一节中，我们给出了一些没有证明的摄动结果，首先对对称矩阵a∈rn×n，假设n×n矩阵的特征值为

命令

λ1≥λ2≥·······≥λn。

我们考虑一个扰动矩阵a+e，并询问a+e的特征值和特征向量与a的特征值和特征向量之间的距离。

### 例15.1。让

.

这是三对角矩阵QR算法的一个典型情况：通过一系列正交相似变换，使三对角矩阵收敛到一个对角矩阵。我们什么时候可以考虑一个小的非对角浮点数为零？A和A+E的特征值能偏离多少？

15.1.扰动理论

定理15.2。设a∈rn×n和a+e为对称矩阵。则λk（a）+λn（e）≤λk（a+e）≤λk（a）+λ1（e），k=1,2，…，n，

和



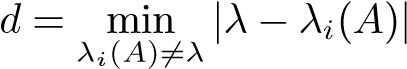
从这个定理我们可以看出，不严格地说，如果我们用扰动矩阵元素，那么特征值也会受到扰动）。例如，在示例15.1中，矩阵e的特征值为±10-15和。因此，两个矩阵的特征值最多相差10-15。

特征向量的灵敏度取决于特征值的分离。

定理15.3。设[λ，q]为对称矩阵A的特征值特征向量对，假设特征值简单。形成正交矩阵并对矩阵qtaq和qteq进行划分，

.

定义



然后假设。然后存在a+e的特征向量q\_，这样q和q\_之间的距离（测量为矢量之间角度的正弦）是有界的：

.

这个定理只有在特征值简单的情况下才有意义。结果表明，特征值相近的特征向量对扰动较为敏感，计算精度较高。

例15.4。例15.1中矩阵A的特征值为

0.8820、1.0000、2.0000、3.1180。

与最小特征值相对应的A和A+E的特征向量之间的偏差可通过以下公式估计：

.

由于特征值被很好地分离，因此该矩阵的特征向量对数据中的扰动相当不敏感。

为了给出非对称矩阵的摄动结果，我们首先引入了上拟三角矩阵的概念：如果上拟三角矩阵的形式为

，

其中，每个RII要么是一个标量，要么是一个具有复共轭特征值的2×2矩阵。R的特征值等于对角块Rii的特征值（也就是说，如果Rii是一个标量，那么它就是R的特征值）。

定理15.5（实舒尔分解）。对于任意（对称或非对称）矩阵a∈rn×n，存在一个正交矩阵u，使得

Utau=R，（15.3）

其中R是上准三角形。

分区

，

其中Uk∈Rn×K和Rk∈Rk×K，从（15.3）我们得到

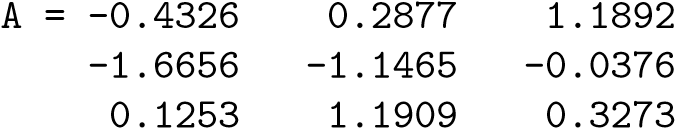
AUK=乌克兰（15.4）

这意味着r（auk）r（uk），其中r（uk）表示英国的范围。因此，UK被称为a的不变子空间或特征空间，分解（15.4）被称为局部舒尔分解。

如果a是对称的，那么r是对角的，schur分解与特征值分解utau=d相同，其中d是对角的。如果A是非对称的，那么它的部分或全部特征值可能是复杂的。

例15.6。舒尔分解是matlab中的标准函数。如果矩阵是实数，则r是上准三角形：

>>A=兰登（3）



>>[U，R]=舒尔（A）

U=0.2827 0.2924 0.9136

0.8191-0.5691-0.0713-0.4991-0.7685 0.4004

15.1.扰动理论

R=-1.6984 0.2644-1.2548 0 0.2233 0.7223

0-1.4713 0.2233

如果我们计算特征值分解，我们得到

>>[X，D]=EIG（A）

|  |  |
| --- | --- |
| X=0.2827 | 0.4094-0.3992i 0.4094+0.3992i |
| 零点八一九一 | -0.0950+0.5569i-0.0950-0.5569i |
| -0.4991美元 | 0.5948 0.5948 |
| D=-1.6984 | 0 0 |
| 零 | 0.2233+1.0309i 0 |
| 零 | 0 0.2233-1.0309i |

非对称矩阵的特征向量不是正交的。

非对称矩阵特征值的灵敏度取决于舒尔分解中严格上三角部分R的范数。为了方便起见，我们这里使用复杂版本的装饰来表示结果。-

位置。

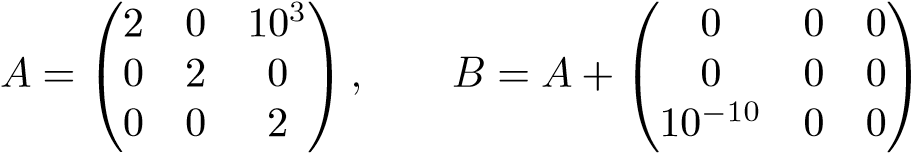
定理15.7。设uhau=r=d+n为a的复Schur分解，其中u为单位，r为上三角，d为对角，τ表示扰动矩阵a+e的特征值。此外，设p为最小整数，使得np=0。然后

，

哪里

.

该定理表明，高度非对称矩阵的特征值比对称矩阵的特征值对扰动更为敏感；参见定理15.2。例15.8。矩阵



具有特征值

2，2，2，

和

分别为2.00031622776602、1.99968377223398、2.0000000000000。扰动的相关量为η1/2≈3.164·10−04。

定理15.3的非对称版本是相似的：特征向量之间的角度又取决于特征值的分离。我们在下面给出了一个简化的陈述，我们忽略了复杂特征值的可能性。

定理15.9。设[λ，q]为a的特征值特征向量对，并假定该特征值是简单的。形成正交矩阵和划分

矩阵

.

定义

假设d>0。如果扰动e足够小，则存在a+e的特征向量q\_，以向量之间角度的正弦测量的q和q\_之间的距离受

.

这个定理本质上说，如果我们扰动，那么特征向量就受到/d的扰动。

例15.10。将三对角矩阵定义为

.

对于n=100，其最小特征值约为0.01098771。以下matlab脚本计算定理15.9中的数量d：

%xn是对应于

%最小特征值

[q，r]=qr（xn）；

h=q'\*a\*q；lam=h（1,1）；a2=h（2:n，2:n）；

d=最小值（svd（a2 lam\*眼（尺寸（a2）））；

我们得到d=1.6207·10−4。因此，如果我们将矩阵扰动10-10，这可能会使特征向量大约改变4·10-6。

15.2.幂法和逆迭代

15.2幂法和逆迭代

幂法是计算最大特征值和相应特征向量的经典迭代方法。它的收敛速度很慢，这取决于特征值的分布。因此，它不应该用于密集矩阵。对于稀疏矩阵，通常应使用Lanczos方法或Jacobi–Davidson方法的变体；见[4]和第15.8.3节。然而，在某些应用中，问题的维度是如此之大，以至于没有其他方法是可行的；见第12章。

尽管在实际问题上的实用性有限，但从理论上讲，幂法是很重要的。此外，还有一种变幂法，逆迭代，这是非常重要的现实意义。

在这一节中，我们给出了比第12章更一般的幂法公式，并回顾了它的一些性质。

### 计算最大特征值的幂法

%k=1的初始近似值x：最大y=a\*x；lambda=y'\*x；如果范数（y-lambda\*x）<tol\*abs（lambda）

break%停止迭代结束x=1/norm（y）\*y；

结束

幂法的收敛性取决于矩阵A特征值的分布，假设最大特征值的大小是简单的，且λi是有序的λ1>λ2≥····≥λn。收敛速度由比值λ2/λ1确定。如果这个比率接近1，那么迭代就非常慢。

根据特征值问题的残差向量，可以建立幂次迭代的停止准则：如果残差r=a x\_−λ\_x\_的范数很小，则特征值近似较好。

例15.11。再考虑一下三对角矩阵

.

A20的两个最大特征值约为3.9677和3.9016。作为初始近似，我们选择了一个随机向量。在图15.1中，我们绘制了迭代期间的不同错误度量：相对残差表示

零

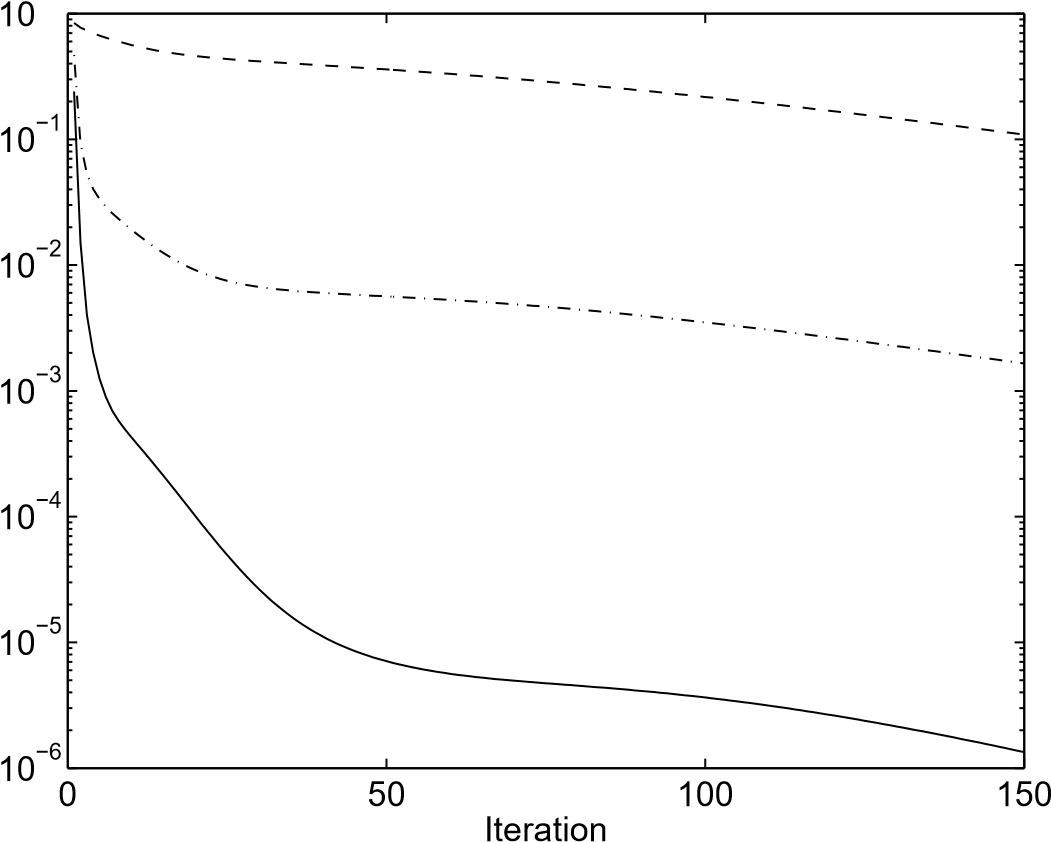


图15.1.A20的功率迭代。相对残差

（实线），特征值近似中的绝对误差（破折号-

虚线），以及精确特征向量和近似值（虚线）之间的角度（弧度）。

第k次迭代中的λ1近似值）、特征值近似值的误差以及精确和近似特征向量之间的角度。经过150次迭代后，计算出的特征值近似值的相对误差为0.0032。

我们得到λ2（a20）/λ1（a20）=0.9833。接下来是

0.9833150≈0.0802，

这表明收敛非常慢，如图15.1所示。这与150次迭代期间精确和近似特征向量之间的角度减小相当：从1.2847弧度到0.0306弧度。

如果我们用幂法中的−1迭代，

X（K）=A−1X（K−1）、

然后，由于a−1的特征值为1/λi，特征值近似序列收敛到1/λmin，其中λmin是最小绝对值的特征值。更好的是，如果我们有足够好的特征值之一τ≈λk的近似值，那么移位矩阵a−τi的特征值最小。因此，“逆幂法”的收敛速度很快，这种方法称为逆迭代。

15.3.三对角形式的相似性减少

**Inverseiteration**

%初始近似x和特征值近似tau

[l，u]=lu（a-tau\*i）；k=1时：max y=u（l\x）；theta=y'\*x；if norm（y-theta\*x）<tol\*abs（theta）

break%停止迭代结束x=1/norm（y）\*y；

结束

lambda=tau+1/theta；x=1/norm（y）\*y；

例15.12。实施例15.10中矩阵A100的最小特征值为λ100=0.01098771187192至14位小数精度。如果我们使用近似值λ100≈τ=0.011并应用逆迭代，我们得到快速收敛；见图15.2。在这个例子中，收敛因子是

，

这意味着经过四次迭代后，误差减小了3·10−10的系数。

为了提高效率，逆迭代要求我们有很好的特征值近似。此外，我们必须能够廉价地求解线性系统（a−τi）y=x（对于y）。如果A是带矩阵，那么可以很容易地得到LU分解，并且在每次迭代中，可以通过前向替换和后向替换（如上面的代码所示）来求解系统。如果可以计算稀疏LU分解而不需要太多填充，那么其他稀疏矩阵也可以使用相同的方法。

15.3三对角形式的相似性降低

我们将在第15.4节中介绍的QR算法是一种迭代算法，其中在每个步骤中都会计算一个QR分解。如果将其应用于稠密矩阵a∈rn×n，则阶跃成本为o（n3）。通过首先将矩阵变换为紧凑形式，通过正交相似变换（15.1），可以大大降低这种高昂的成本。

A-→V TAV，

对于正交矩阵V。我们已经在（15.2）中看到了特征值在这种变换下是保持不变的，

ax=λx v tav y=λy，

其中y=v tx。

1

2

3

4

5

6

7

10

−14

10

−12

10

−10

10

−8

10

−6

10

−4

10

−2

Iteration

图15.2.a10的逆迭代，τ=0.011。相对残差（实线）、特征值近似（虚线）中的绝对误差以及精确特征向量和近似（虚线）之间的角度（弧度）。

它可以简化为三对角形式。我们用一个例子来说明这个过程，让a∈rn×n是对称的。通过一系列的户主转变

n=6。首先，当我们从左乘a时，我们构造一个从第1列位置3到n的元素归零的变换：a=h1\_××××××\_\_\_=\_\_\_\_×0000××××\_\_\_

××××××

h1\_\_\_××××××\_。

××××××

××××××

××××××

在转换中更改的元素由第一行的元素表示，不会更改。在正交相似变换中。注意，

我们将乘以从右边转置的同一矩阵。因为在左乘法中，第一行没有被触摸，所以第一列将保持不变：

.

由于对称性，第一行的元素3到n等于零。

在下一步中，我们将第二列中位置4到N的元素归零。因为这只影响第3行到第n行以及相应的列，所以这不会破坏我们在第一步中创建的零。结果是

.

在n-2这样的相似变换之后，矩阵是三对角形式：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| \_×0  T× | ×  ×  × | 零  ×  × | 零  零  × | 0 0  零 | 0\_  0 0 |

V av=，

000 000×0×0×0×0×0×0\_

0倍

其中v=h1th2t···································

总之，我们已经演示了如何通过N-2户主转换序列将对称矩阵简化为三对角形式：

a−→t=v tav，v=h1h2···hn−2，（15.5）

由于三对角矩阵T是通过相似变换进行约简的，所以它具有与A相同的特征值。

如果要利用对称性，将其简化为三对角形式需要4n3/3个触发器。在QR分解的情况下，户主转换可以存储在A的次极化部分。如果显式计算V，则需要4n3/3额外的触发器。

15.4对称三对角的QR算法

### 矩阵

现在我们将给出对称三对角矩阵的QR算法的草图。我们强调，我们的matlab代码被大大简化，仅用于演示算法的基本思想。实际软件（在LAPACK中）包含许多效率、鲁棒性和数值稳定性方面的特性。

我们所描述的过程可以看作是相似性约简（15.5）的延续，但现在我们将矩阵t简化为对角形式：

t−→=qttq，q=q1q2···（15.6）

式中∧=diag（λ1λ2…，λn）。矩阵qi将是正交的，但在这里它们将使用平面旋转来构造。然而，（15.5）和（15.6）之间最重要的区别是没有计算∧的有限算法。我们计算矩阵序列，

t0：=t，t i=qti−1qi，i=1,2，…（15.7）

这样它就收敛到一个对角矩阵，

lim ti=∧。

I→∞

我们将在数值例子中证明收敛速度非常快，因此在浮点运算中，算法实际上可以被视为有限的。由于（15.7）中的所有变换都是相似变换，因此∧的对角元素是t的特征值。

我们现在给出对称三对角矩阵t∈rn×n的qr算法的第一版。

对称t的qr迭代：底部特征值

对于i=1：最大百分比临时简化

mu=wilkshift（t（n-1:n，n-1:n））；[q，r]=qr（t-mu\*i）；

t=r\*q+mu\*i端

函数mu=wilkshift（t）；%计算wilkinson shift l=eig（t）；如果abs（l（1）-t（2,2））<abs（l（2）-t（2,2））

mu=l（1）；否则mu=l（2）；

结束

我们看到移位矩阵qr=t−τi的qr分解被计算出来，然后将移位加回t：=rq+τi。移位是右下角最接近tnn的2×2子矩阵的特征值。这叫做威尔金森变换。

我们把算法应用于矩阵

（15.8）

在第一步之后，结果被（为了可读性稍微编辑了一下）

t=1.0000 0.7071 0 0 0 0 0

0.7071 2.0000 1.2247 0 0 0 0 1.2247 2.3333-0.9428 0 0 0-0.9428 1.6667 0.8660 0

0 0 0 0.8660 2.0000-0.5000

0 0 0 0-0.5000 3.0000

我们首先看到三角形结构被保留下来，右下角的非对角元素变小了。我们还执行了三个步骤，并更仔细地观察了该子矩阵：

2.36530292572181-0.02609619264716

|  |  |
| --- | --- |
| -0.02609619264716 | 3.24632297453998 |
| 2.59270689576885 | 0.00000366571479号 |
| 0.00000366571479号 | 3.24697960370634号 |
| 2.77097818052654 | 0.00000亿 |
| -0.00000亿 | 3.24697960371747 |

因此，经过四次迭代后，非对角元素的工作精度变为零，因此我们在右下角有一个特征值。

当特征值被发现后，我们可以消除这个问题，继续使用上面的（n-1）×（n-1）子矩阵，它现在是

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 零点四三七四 | 零点三一七六 | 零 | 零 | 零 |
| 零点三一七六 | 零点七九六一 | -0.4395美元 | 零 | 零 |
| 零 | -0.4395美元 | 一点三一九八 | 零点二九二二 | 零 |
| 零 | 零 | 零点二九二二 | 三点四二八八 | 零点五九零二 |
| 零 | 零 | 零 | 零点五九零二 | 二点七七一零 |

我们现在对这个矩阵应用相同的算法。在接下来的三个步骤中跟踪它的右下子矩阵

3.74629910763238-0.01184028941948

|  |  |
| --- | --- |
| -0.01184028941948 | 2.44513898239641 |
| 3.68352336882524 | 0.0000009405188元 |
| 0.0000009405188元 | 2.44504186791263 |
| 3.54823766699472 | 0.00000亿 |
| -0.00000亿 | 2.44504186791263 |

在这三次迭代之后，我们又在右下角有了一个特征值。现在，该算法通过将该特征值缩减并将活动矩阵的维数减少一个来进行计算。下面给出了该算法的初步实现。

**QRiterationforsymmetric**

T

函数[d，it]=qrtrid（t）；

%计算对称三对角的特征值

%矩阵采用二维显式算法

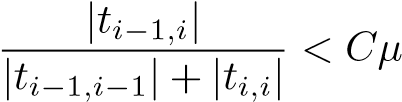
%威尔金森位移n=尺寸（t，1）；it=0；对于i=n:-1:3

而abs（t（i-1，i））>…

（abs（t（i，i））+abs（t（i-1，i-1））\*c\*eps it=it+1；mu=wilkshift（t（i-1:i，i-1:i））；[q，r]=qr（t（1:i，1:i）-mu\*眼（i））；

t=r\*q+mu\*眼（i）；末端d（i）=t（i，i）；末端d（1:2）=eig（t（1:2,1:2））'；

对于给定的子矩阵t（1:i，1:i），qr步数迭代到停止标准



满足，其中c是一个小常数，μ是单位四舍五入。从定理15.2我们可以看出，考虑到这样一个小元素作为一个数值零，会导致特征值的扰动非常小（并且可以接受）。在实际软件中，使用了稍微复杂一点的停止标准。

当应用于矩阵t100（参见（15.8））时，值为c=5204 qr步骤，即每个特征值约2步。计算的特征值与Matlab特征值函数计算的最大偏差为2.9·10−15。

当然，使用matlab函数qr来计算三对角矩阵的qr分解是一种效率很低的方法，qr是一种基于户主的算法。相反，应使用O（N）触发器中的n-1平面旋转计算分解。我们用一个小例子来说明这个过程，其中三对角矩阵t=t（0）是6×6。（1,2）平面中的第一个子垂线元素（从顶部）通过从左侧旋转归零，然后第二个子垂线通过（2,3）中的旋转归零。象征性地，

.

注意创建的填充（新的非零元素，表示为+）。在n-1步之后，我们得到一个具有三个非零对角线的上三角矩阵：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| =GTN 1 GT1（t（0）×0  R−····−τi）= | ×  ×0 | +  ×  ×0 | +  ×  ×0 | +  ×  ×0 | ××+。 |

然后，我们应用右旋，rg1······gn-1，即，我们从涉及前两列的转换开始。然后进行涉及第二列和第三列的旋转。两步之后的结果是

.

我们看到，在从左到右的变换中，我们在对角线下面引入的零被系统地填充了。经过n-1步后，我们

.

但是我们做了一个相似变换：当q=g1g2······································

（1）t（0）t（0）

|  |  |
| --- | --- |
| t=rq+τi=q（t−τi）q+τi=q t q，  所以我们知道t（1）是对称的，  ××\_\_\_\_  （1）=××  T×××。  ×××。  ×××。  ××  因此，我们显示了以下结果。 | （15.9） |

提案15.13。三对角矩阵的qr步骤

qr=t（k）−τki，t（k+1）=rq+τki，

是相似变换

t（k+1）=qtt（k）q，（15.10）

保留了三对角结构。转换可以用O（N）触发器中的平面旋转来计算。

从（15.9）可以看出，这种转变似乎不起重要作用。然而，它决定了QR步骤中正交变换的值。实际上，移位策略对于算法的有效性是绝对必要的：如果不执行移位，那么QR算法通常会非常缓慢地收敛，实际上与幂法一样缓慢；参见第15.2节。另一方面，可以证明[107]（如[93，第3章]）移位的QR算法具有很快的收敛性。

提案15.14.具有威尔金森位移的对称QR算法三次收敛于特征值分解。

在QR算法的实际软件中，我们上面概述的算法有几个改进。例如，算法检查所有的非对角元素（如果它们很小）：当发现一个可以忽略的非对角元素时，问题可以分为两部分。QR算法还有一个分而治之的变体。关于广泛治疗，见[42，第8章]。

15.4.1隐性转变

QR算法的一个重要方面是移位可以隐式执行。这对于该算法在支持向量机和非对称特征问题中的应用尤其有用。这个变量是基于隐式Q定理的，我们在这里用稍微简化的形式给出。

定理15.15。假设a是对称的，假设q和v是正交矩阵，qtaq和v tav都是三对角矩阵。那么，如果q和v的第一列相等，q1=v1，那么q和v基本上相等：q i=±vi，i=2,3，…，n。

有关证据，请参见[42，第8章]。

这个定理的一个结果是，如果我们在t−τi的qr分解中确定并应用第一个转换，并且如果我们以这样一种方式构造其余的转换，我们最终得到一个三对角矩阵，那么我们已经执行了一个移位的qr步骤，正如在命题中一样。15.13.本程序执行如下。

让第一个平面旋转确定为

，（15.11）

其中α1和β1是t的上对角线和次对角线元素。定义

，

并将旋转应用于t。左乘在第一行引入一个新的非零元素，相应地，右乘在第一列引入一个新的非零元素：

γ

×××。

1. TG1=\_\_\_×+×+\_\_\_\_

GT×××，

×××。

××××年

其中+表示新的非零元素。接下来，我们确定（2,3）-平面上的旋转，该平面消灭新的非零，同时进一步引入新的非零：

×××。+

1. GT1 TG1B2=\_\_\_\_×0×+0\_\_\_\_

燃气轮机×××。

×××。

×××。

××

以一种类似的方式，我们“追逐膨胀”，直到我们

，

通过最后的旋转，我们可以使凸起归零，同时恢复三对角形状。

注意，只有在确定第一个旋转（15.11）时才使用换档。旋转仅适用于非位移三对角矩阵。由于隐式qr定理，定理15.15，这相当于命题15.13中给出的移位qr步。

15.4.2特征向量

计算对称矩阵特征值的QR算法（包括三对角形式的约简）只需计算特征值，就需要4n3/3次方波。计算特征向量矩阵的正交变换的累积，大约需要另一个9n3触发器。

如果所有n个特征值都需要，但只有少数特征向量需要，那么使用逆迭代（第15.2节）计算这些特征向量更便宜，计算出的特征值λ\_i作为偏移：

（a−λ\_ii）x（k）=x（k−1），k=1,2，…。

QR算法产生的特征值非常接近精确的特征值（见下文），因此通常只需要逆迭代的一步就可以得到一个非常好的特征向量，即使对特征向量的初始猜测是随机的。

从数值稳定性的角度来看，QR算法是理想的。存在一个正正交矩阵q和一个扰动，使得特征值d\_的计算对角矩阵完全满足

qt（a+e）q=d\_

其中μ是浮点系统的四舍五入单位。

然后，根据定理15.2，我们知道计算出的特征值λ\_i与精确的特征值相差很小。

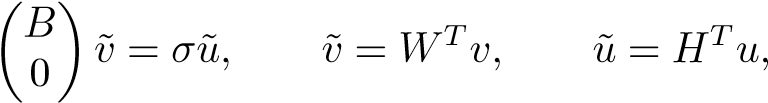
### 15.5计算SVD

由于矩阵A的奇异值是ata和aat的特征值平方，很明显，计算svd的问题可以用类似于对称特征值问题的算法来解决。然而，重要的是避免形成矩阵ata和aat，因为这将导致信息丢失（参见第54页的最小二乘法示例）。

假设a为m×n，m≥n，计算稠密矩阵a的SVD的第一步是通过户主从左到右的变换将其简化为上双对角形式。

（15.12）

有关此减少的说明，请参见第7.2.1节。因为我们在这个约简中使用了正交变换，所以矩阵b的奇异值与a相同。假设σ是a的奇异值，其奇异向量为u和v。那么av=σu等于



从（15.12）。

15.6.非对称特征值问题

很容易看出矩阵btb是三对角的。计算b的奇异值的选择方法是矩阵btb采用隐式移位的三对角qr算法，而不显式形成。

设a∈rm×n，式中m≥n，在6mn2+20n3触发器中可计算出薄SVD a=u1∑v t（参见第6.1节）。

### 15.6非对称特征值问题

如果我们对非对称矩阵执行与第15.3节中相同的步骤，那么由于非对称性，对角线上方的任何元素都不会归零。因此，最终结果是Hessenberg矩阵：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | ×  ×  ×0  零  零 | ×  ×  ×  ×0  零 | ×  ×  ×  ×  ×0 | ×  ×  ×  ×  ×  × | ×  \_\_\_\_  ×  ×  ×  × |

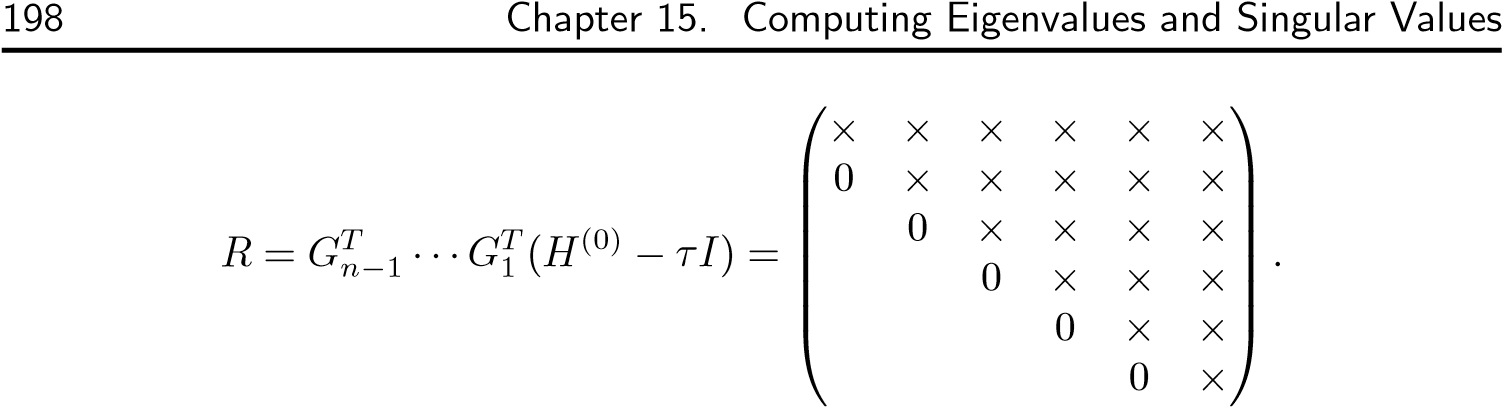
使用户主转换将Hessenberg形式简化为10n3/3触发器。

15.6.1非对称矩阵的QR算法

第15.4节中给出的三对角矩阵的“未定义”QR算法同样适用于Hessenberg矩阵，其结果是上三角矩阵，即schur分解中的r因子。为了提高效率，就像在对称情况下一样，算法的每个步骤中的qr分解都是使用平面旋转来计算的，但是这里的转换应用于更多的元素。

我们用一个小例子来说明这个过程。假设矩阵H∈R6×6为上Hessenberg，假设从右下角的2×2矩阵计算出威尔金森位移τ。为了简单起见，我们假设这种转变是真实的。表示h（0）：=h。h−τi中的第一个次对角元素（从顶部）通过（1,2）平面中从左侧旋转归零，然后通过（2,3）中的旋转归零。象征性地，

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| \_\_\_\_\_\_\_\_9116;\_\_\_\_\_\_×0×0  经过n-1步，我们得到一个上三角矩阵： | ×  ×  ×  × | ×  ×  ×  ×  × | ×  ×  ×  ×  ×  × | ×  \_\_\_\_  ×  ×  ×  × |



然后，我们应用右旋，rg1······gn-1，即，我们从涉及前两列的转换开始。然后进行涉及第二列和第三列的旋转。两步之后的结果是

.

我们看到我们在从左到右的变换中引入的零被系统地填充了。经过n-1步后，我们

.

但是我们做了一个相似变换：当q=g1g2······································

h（1）=rq+τi=qt（h（0）−τi）q+τi=qth（0）q，（15.13）

我们知道h（1）与h（0）具有相同的特征值。

非对称QR算法的收敛性几乎与对称QR算法的收敛性相同[93，第2章]。

提案15.16。具有威尔金森位移的非对称QR算法向舒尔分解四次收敛。

正如在对称情况下，上述算法有许多改进；参见，例如[42，第7章]，[93，第2章]。特别是，人们通常使用隐式双移位来避免复杂的算术运算。

在给定特征值的情况下，选取的特征向量可以用上Hessenberg矩阵进行逆迭代计算，并将计算出的特征值作为位移。

### 15.7稀疏矩阵

在许多应用中，矩阵元素的一小部分是非零的。然后矩阵被称为稀疏矩阵。很常见的是

15.7.稀疏矩阵

矩阵元素不是零。

在稀疏矩阵特征值问题的数值求解中，通常采用迭代法。这是因为第15.3节中描述的紧凑形式的转换将完全破坏稀疏性，从而导致过度的存储需求。此外，将其简化为紧凑形式的计算复杂度往往过高。

在第15.2节和第15.8节中，我们描述了数值求解大型稀疏矩阵特征值（和奇异值）问题的几种方法。这里我们简要描述了一种存储稀疏矩阵的可能方法。

为了利用矩阵的稀疏性，只能存储非零元素。我们简要描述了一种稀疏矩阵的存储方案，即压缩行存储。

例15.17。让

γ

0.6667 0 0 0.2887\_

0 0.7071 0.4082 0.2887

A=。

0.3333 0 0.4082 0.2887\_\_

0.6667 0 0 0 0

在压缩行存储中，非零项存储在一个称为val的向量中（我们对表中的元素进行四舍五入以节省此处的空间），以及等长向量colind中相应的列索引：

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 瓦尔 | 零点六七 | 零点二九 | 零点七一 | 零点四一 | 零点二九 | 零点三三 | 零点四一 | 零点二九 | 零点六七 |
| 科林德 | 一 | 四 | 二 | 三 | 四 | 一 | 三 | 四 | 一 |
| 罗普特 | 一 | 三 | 六 | 九 | 十 |  |  |  |  |

向量rowptr指向val中每行第一个元素所占据的位置。

压缩行存储方案便于将y=ax相乘。rowptr向量中在val向量结束后指向（不存在）位置的额外条目用于简化y=ax的乘法代码。

稀疏A的乘法y=ax

函数y=ax（val、colind、rowptr、x）

%计算y=a\*x，其中压缩后的行存储m=length（rowptr）-1；对于i=1:m

a=val（rowptr（i）：rowptr（i+1）-1）；y（i）=a\*x（colind（rowptr（i）：rowptr（i+1）-1））；

端部Y=Y'；

可以看出，压缩行存储对于y=atz的乘法是不方便的。然而，有一种类似的压缩列存储方案，自然非常适合这种情况。

稀疏矩阵的压缩行（列）存储与FORTRAN和C等编程语言有关，程序员必须显式处理稀疏存储[27]。matlab有一个内置的稀疏矩阵存储方案，矩阵操作过载。例如，对于稀疏矩阵A，matlab语句y=a\*x实现稀疏矩阵向量乘法，并且matlab在内部执行类似于上面的代码。

在特定的应用中，根据矩阵的结构，不同的稀疏矩阵存储方案会影响矩阵运算的性能。在[39]中，比较了用于信息检索的稀疏矩阵算法。

### 15.8 Arnoldi和Lanczos方法

QR方法可以用来计算中等尺寸矩阵的特征值和奇异值分解。（中型矩阵的大小取决于可用的计算能力。）通常在数据挖掘和模式识别中，矩阵非常大且稀疏。然而，稀疏矩阵的特征值、奇异值和舒尔分解通常是密集的：几乎所有元素都是非零的。

例15.18。考试中链接图矩阵的舒尔分解-

例1.3，

，

在matlab中计算得出：【u，r】=schur（a），结果如下：

U =

-0.0000-0.4680-0.0722-0.0530 0.8792-0.0000-0.0000-0.4680-0.0722-0.3576-0.2766-0.7559-0.5394 0.0161 0.3910 0.6378 0.0791-0.3780-0.1434-0.6458-0.3765 0.3934-0.3509 0.3780

-0.3960-0.2741 0.6232-0.4708-0.1231 0.3780

-0.7292 0.2639-0.5537-0.2934 0.0773-0.0000

r＝

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 零点九二零七 | 零点二二三九 | -0.2840个 | 零点零一四八 | -0.1078个 | 零点三三三四 |
| 零 | 零点三三三三 | 零点一四九五 | 零点三七四六 | -0.3139个 | 零点零三七一 |
| 零 | 零 | -0.6361个 | -0.5327个 | -0.0181个 | -0.0960 |
| 零 | 零 | 零 | -0.3333个 | -0.1850年 | 零点一七五一 |
| 零 | 零 | 零 | 零 | -0.2846个 | -0.2642个 |
| 零 | 零 | 零 | 零 | 零 | 零 |

我们发现，正交矩阵的几乎所有元素都是非零的。

因此，由于存储要求变得禁止，使用QR方法通常是不可能的。相反，我们使用的方法不转换矩阵本身，而是将其用作运算符，即计算矩阵向量积y=ax。我们已经在第15.2节中描述了一种这样的方法，即功率法，可以用来计算最大特征值和相应的特征向量。本质上，在幂法中，我们计算一个向量序列，ax0，a2x0，a3x0，…，它向特征向量收敛。然而，一旦我们计算出一个新的功率，即，我们已经从yk 1=ak-1X到yk=akx，我们就扔掉yk-1和所有的信息。

包含在特征向量的早期近似中。

Krylov子空间方法的思想是使用矢量X0、Ax0、A2X0、…、AK−1的序列中的信息，这些信息组织在子空间中，即Krylov子空间，

kk（a，x0）=span x0，ax0，a2x0，…，ak−1x0，

从这个子空间中提取尽可能好的特征向量近似值。在第7章中，我们已经描述了Lanczos双标准化方法，这是一种Krylov子空间方法，可用于求解近似最小二乘问题。在第15.8.3节中，我们将证明它也可用于计算矩阵的某些奇异值和向量的近似值。但首先，我们提出了阿诺迪方法及其在计算大稀疏矩阵的部分舒尔分解问题中的应用。

15.8.1 Arnoldi方法和Schur分解

计算舒尔分解（定理15.5），假设a∈rn×n是大的、稀疏的、非对称的，并且我们希望toa=urut，其中u是或-

T形和R形是上三角形。我们对Arnoldi方法的推导将类似于lgk双标准化方法的第7章。因此，我们将从正交相似度约简到上赫森堡形式（这里n=6）：t\_\_\_\_\_\_

×0×x×x×x

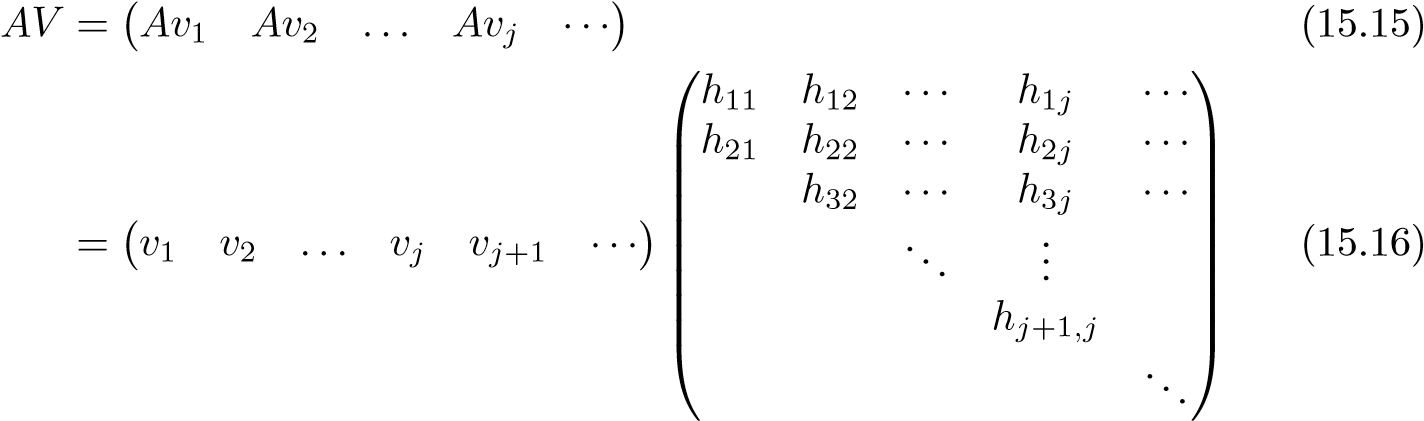
V av=H=0×××××××××（15.14）

零

0 0 0×0×X×X

0 0 0

原则上，这可以使用第15.3节中的户主转换来计算，但是由于a是稀疏的，这将导致填充零元素。相反，我们将展示V和H列可以以递归的方式计算，只使用矩阵向量积（就像在lgk双向化方法中一样）。在表格中重写（15.14）



一列一列地读，我们发现第一列是

av1=h11v1+h21v2，

它可以写在表格里

h21v2=av1−h11v1。

因此，因为v1和v2是正交的，所以h21是根据v2具有欧几里德长度1的要求确定的。同样地，（15.15）–（15.16）的第j列是

，

可以写的

（15.17）

现在，有了v1，v2，…，vj，我们可以从（15.17）计算vj+1，如果我们规定它与前面的向量是正交的。这给出了方程

hij=vitavj，i=1,2，…，j.

元素hj+1，j是根据vj+1的长度为1的要求得到的。

因此，我们可以使用以下递归计算v和h的列：

#### 阿诺迪法

1。起始向量v1，满足2。对于j=1,2，…

.

#### 三。结束

显然，在步骤j中只需要一个矩阵向量积avj。

对于一个大型稀疏矩阵来说，在递归中执行许多步骤是不可能的，主要是出于存储的原因。假设已经执行了k个步骤，其中，和定义

.

现在我们可以用矩阵形式编写递归的前k步：

，（15.18）

哪里。这叫做阿诺迪分解。

在递归的k步之后，我们执行了k矩阵向量乘法。下面的命题表明，我们保留了这些步骤中产生的所有信息（与幂法相比）。

提案15.19。向量v1，v2，…，vk是krylov子空间k（a，v1）=span v1，av1，…，ak−1v1中的正交基。

证据。矢量的正交性通过构造（或直接计算）来验证。第二部分可以通过归纳法加以证明。

现在的问题是，我们如何能从krylov子空间近似特征值和特征向量。注意，如果zk是一个特征空间（见（15.4）），那么对于一些矩阵m∈Rk×k，我们将得到azk=zkm。因此，为了观察vk偏离特征空间的程度，我们可以检查对于一些矩阵m，剩余的avk−vkm有多大。幸运的是，有一个选择最优m的方法任何给定的VK。

定理15.20。设vk∈rn×k有正态列，并定义r（m）=avk−vkm，其中m∈rk×k。



解决方案m=vktavk。

证据。参见，例如[93，定理4.2.6]。

从Arnoldi分解（15.18），我们立即得到最佳MA-

特里克斯

m=vkt（vkhk+hk+1，kvk+1etk）=hk，

因为vk+1与前面的向量是正交的。根据阿诺迪分解，最佳残差由下式给出：

，

所以剩余范数在Arnoldi递归中是免费的。

假设vk是一个足够好的特征空间近似值，我们如何计算一个近似的部分schur分解auk=ukrk？让

是的舒尔分解，我们得到了

.

得出的特征值是A特征值的近似值。

例15.21。我们使用幂法和Arnoldi法计算了例15.10中定义的矩阵A100的最大特征值。图15.3给出了特征值近似值的误差。结果表明，Krylov子空间的特征值信息比幂法中唯一的向量所携带的特征值信息多得多。

上述基本Arnoldi方法有两个问题，这两个问题都可以有效地解决：

•在精确算法中，VJ向量是正交的，但在浮点算法中，正交性随着迭代的进行而丢失。通过明确地重新调整向量的方向性来修复正交性。这可以在算法的每个步骤中完成，也可以在检测到非正交性时选择性地完成。•工作量和存储需求随着迭代的进行而增加，在计算出足够好的近似值之前，可能会耗尽内存。这可以通过重启Arnoldi来解决。

零

0

5

10

15

10

−3

10

−2

10

−1

10

Iteration

图15.3.计算A100最大特征值的幂和Arnoldi方法。功率法（虚线）和阿诺迪法（虚线）特征值近似的相对误差。

程序。本文发展了一种方法，在给定尺寸k的Arnoldi分解的情况下，将其简化为较小尺寸k0的Arnoldi分解，并在这种简化中清除不需要的特征值。这个隐式重启的arnoldi方法[64]已经在matlab函数eigs中实现。

15.8.2 Lanczos三对角化

如果将Arnoldi程序应用于对称矩阵A，则由于对称性，上Hessenberg矩阵hk变为三对角矩阵。从A的正交三对角化（15.5）开始，我们可以得出一个更经济的对称算法版本，我们将其写成

.

通过识别左右两侧的J列并重新排列方程，我们得到

βjvj+1=avj−αjvj−βj−1vj−1，

我们可以把它用于方程a v=v t的递归重构，系数αj和βj是根据向量正交和归一化的要求确定的。下面我们给出了生成Lanczos分解的Lanczos三对角化方法的基本版本，

avk=vktk+βkvk+1etk，

其中t k由t的第一行和第三列k组成。

#### 兰佐斯三对角化

1。把β0=0和v0=0，选择一个起始向量v1，满足2。对于j=1,2，…

.

#### 三。结束

同样，在递归矩阵中，a不被转换，但只用于矩阵向量乘法，并且在每次迭代中只需要计算一个矩阵向量积。这种基本的Lanczos三对角化程序与基本的Arnoldi程序有相同的缺陷，可以用同样的方法来解决问题。

matlab函数eigs检查矩阵是否对称，如果是这样，则使用隐式重新启动的lanczos三对角化方法。

15.8.3计算稀疏SVD

LGK Bidialogization方法最初是为计算SVD而制定的[41]。它可用于计算部分双向化（7.11）。

azk=pk+1bk+1，

其中bk+1为双向，zk、pk+1为正交。基于这种分解，奇异值和奇异向量的近似计算方法与前一节中的三对角化方法类似。事实上，可以证明（如[4，第6.3.3章]）LGK

15.9.软件

双对角化过程等价于将Lanczos三对角化应用于对称矩阵。

，（15.19）

使用特定的启动向量，因此可以应用隐式重新启动。

matlab函数svds实现了矩阵的Lanczos三对角化方法（15.19），隐式重启。

### 15.9软件

在许多计算领域，一个相当常见的错误是低估了开发软件的成本。因此，不利用现有的软件是非常不明智的，尤其是当它是由世界专家开发并免费提供时。

15.9.1拉帕克

lapack是一个线性代数包，可以从netlib访问和下载，网址为http://www.netlib.org/lapack/。

我们引用网页描述：

Lapack用Fortran 77编写，并提供求解联立线性方程组、线性方程组的最小二乘解、特征值问题和奇异值问题的例程。还提供了相关的矩阵分解（Lu、Cholesky、QR、SVD、Schur、广义Schur），以及相关的计算，如Schur分解的重新排序和估计条件数。密集矩阵和带状矩阵被处理，但不是一般稀疏矩阵。在所有领域，实矩阵和复矩阵都提供了类似的功能，包括单精度和双精度。

lapack例程的编写使得尽可能多的计算是通过调用基本线性代数子程序（blas）来执行的。许多现代高性能计算机都可以使用高效的机器专用BLAS实现。或者，用户可以下载Atlas以自动为体系结构生成优化的BLAS库。

在LAPACK的基础上建立了matlab中的基本稠密矩阵函数。

LAPACK（或LAPACK的转换/转换）的可选语言接口可在FORTRAN 95、C、C++和Java中使用。

scalapack是lapack的一个并行版本，也可以从netlib获得，网址为http://www.netlib.org/scalapack/。这个包是为消息传递并行计算机而设计的，可以在任何支持MPI的系统上使用。

15.9.2稀疏矩阵软件

如前所述，matlab中的特征值和奇异值函数基于lanczos和arnoldi方法，并隐式重启[64]。这些算法摘自http://www.caam.rice.edu/software/arpack/。

从网页：

该包被设计用来计算一般n乘n矩阵a的几个特征值和相应的特征向量。它最适合于大型稀疏或结构化矩阵a，其中结构化意味着矩阵向量积w←av需要n阶，而不是通常的n2浮点数。int操作。

本书[4]概述了特征值和奇异值计算的算法和软件。

有关密集和稀疏矩阵计算的其他软件，请访问http://www.netlib.org/linalg/。

15.9.3编程环境

我们在这本书中使用了matlab作为描述算法的工具。在其他商用软件系统中，我们要提到Mathematica和统计软件包，如SASR和SPSS，它们具有矩阵计算、数据和文本挖掘的功能。

41http://www.wolfram.com/、http://www.sas.com/和http://www.spss.com/。

参考文献

1. E.Anderson、Z.Bai、C.Bischoff、L.S.Blackford、J.Demmel、J.Dongarra、J.Du Croz、A.Greenbaum、S.Hammarling、A.McKenney和D.C.Sorensen。《拉帕克用户指南》，第3版，暹罗，费城，1999年。
2. ANSI/IEEE 754标准。二进制浮点运算。IEEE，纽约，1985年。
3. R.Baeza Yates和B.Ribeiro Neto。现代信息检索。ACM出版社，Addison-Wesley，纽约，1999年。
4. Z.Bai，J.Demmel，J.Dongarra，A.Ruhe，H.van der Vorst，eds.代数特征值问题求解模板：实用指南。暹罗，费城，2000年。
5. R.Barrett、M.Berry、T.F.Chan、J.Demmel、J.Donato、J.Dongarra、V.Eijkhout、R.Pozo、C.Romine和H.van der Vorst。线性系统解的模板：迭代方法的构建块。暹罗，费城，1994年。
6. 卑尔根。生物信息学计算。普伦蒂斯-霍尔，纽约，2002年。
7. P.柏金。关于pagerank计算的调查。互联网数学，2:73–1202005。
8. M.Berry和M.Browne。使用非负矩阵因子分解的电子邮件监视。计算。数学。组织理论，11:249–264，2005。
9. M.W.Berry、S.T.Dumais和G.W.O'Brien。利用线性代数进行智能信息检索。暹罗版本，1995年37:573–595。
10. M.J.A.Berry和G.Linoff。掌握数据挖掘。客户关系管理的艺术与科学。约翰·威利，纽约，2000年。
11. M.W.Berry，ed.计算信息检索。暹罗，费城，2001年。
12. M.W.贝瑞和M.布朗。了解搜索引擎。数学建模与文本检索，第2版，暹罗，费城，2005年。

二百零九

1. M.W.Berry、M.Browne、A.Langville、V.P.Pauca和R.J.Plemmons。近似非负矩阵分解的算法和应用。技术报告，田纳西大学计算机科学系，2006年。
2. 美国BJ公司最小二乘问题的数值方法。暹罗，费城，1996年。
3. 美国BJ公司最小二乘问题的计算。第13:1–512004年。
4. K.Blom和A.Ruhe。一种用于信息检索的Krylov子空间方法。暹罗J.矩阵分析申请，26:566–5822005。
5. V.D.布朗德尔、A.加加多、M.海曼、P.塞内拉特和P.范门。图顶点之间的相似性度量：同义词提取和网页搜索的应用。暹罗版本，46:647–6662004。
6. C.Boutsidis和E.Gallopoulos。基于SVD的非负矩阵分解初始化。技术报告HPCLAB-SCG-6/08-05，帕特拉斯大学，希腊，2005年。
7. S.Brin和L.Page。对大型超文本网络搜索引擎的剖析。计算。网络ISDN系统，30:107–117，1998年。
8. J.-P.Brunet、P.Tamayo、T.R.Golub和J.P.Mesirov。利用矩阵分解发现基因和分子模式。PNAS，101:4164–41692004年。
9. M.C.Burl、L.Asker、P.Smyth、U.Fayyad、P.Perona、L.Crumpler和J.Aubele。学会认识金星上的火山。机器学习，30:165–1951998。
10. P.A.Businger和G.H.Golub。户主变换的线性最小二乘解。数字。数学，1965年7:269–276。
11. R.Chelappa、C.L.Wilson和S.Sirohey。人脸和机器识别：一项调查。PROCIEEE，83:705–740，1995年。
12. N.Christianini和J.Shawe Taylor。支持向量机简介。剑桥大学出版社，伦敦，2000年。
13. K.J.CIOS、W.Pedrycz和R.W.Swiniarski。数据挖掘。知识发现方法。Kluwer，波士顿，1998年。
14. J.M.Conroy、J.D.Schlesinger、D.P.O'Leary和J.Goldstein。回到基础：2006年经典。在2006年第02届会议记录中。网址：http://duc.nist.gov/pubs.html。
15. T.A.戴维斯。稀疏线性系统的直接方法。算法基础2.暹罗，费城，2006年。
16. S.Deerwester、S.Dumais、G.Furnas、T.Landauer和R.Harsman。通过潜在语义分析进行索引。J.阿米尔。SOC。通知。科学，41:391–407，1990年。
17. J.W.德梅尔。应用数值线性代数。暹罗，费城，1997年。
18. I.S.Dhillon和D.S.Modha。使用聚类对大型稀疏文本数据进行概念分解。机器学习，42:143–1752001。
19. R.O.Duda、P.E.Hart和D.G.Storck。模式分类，第2版，威利国际科学，纽约，2001年。
20. L.Eld'en.公司偏最小二乘与Lanczos双标准化的比较I：多元回归投影法的分析。计算。统计学家。数据分析，46:11–31，2004年。
21. L.Eld'en.公司数据挖掘中的数值线性代数。活动编号：15:327–3842006。
22. L.Eld'en、L.Wittmeyer Koch和H.Bruun Nielsen。介绍数值计算分析和matlab说明。学生：Literatur，Lund，2004年。
23. M.Fayyad，G.Piatetsky Shapiro，P.Smyth，R.Uhurusamy，eds.知识发现和数据挖掘进展。AAAI出版社/麻省理工学院出版社，加利福尼亚州门罗公园，1996年。
24. J.H.福勒和S.杰恩。最高法院判例的权威：网络分析。技术报告，加州大学政治科学系，戴维斯，2005年。
25. Y.Gao和G.Church。通过稀疏非负矩阵因子分解改进分子癌症分类发现。生物信息学，21:3970–39752005。
26. J.T.Giles、L.Wo和M.W.Berry。用于文本挖掘的GTP（通用文本分析器）软件。《统计数据挖掘与知识发现》，H.Bozdogan，ed.，CRC出版社，佛罗里达州博卡拉顿，2003年，第455-471页。
27. N.Goharian、A.Jain和Q.Sun。稀疏矩阵信息检索算法的比较分析。j.系统。赛博内。通知，2003年1月。
28. G.H.Golub和C.Greif。用于计算pagerank的Arnoldi类型算法。比特，46:759–771，2006。
29. G.Golub和W.Kahan。计算矩阵的奇异值和伪逆。暹罗J.num.肛门的Ser。B，1965年2:205–224。
30. G.H.Golub和C.F.Van Loan。矩阵计算，第3版，约翰霍普金斯出版社，巴尔的摩，1996年。
31. D.格罗斯曼和O.弗里德。信息检索：算法和启发式。Kluwer，波士顿，1998年。
32. Z.Gyo–ngyi、H.Garcia Molina和J.Pedersen。与trustrank抗击网络垃圾邮件。第30届超大数据库国际会议，摩根考夫曼，2004年，第576-587页。
33. J.Han和M.Kamber。数据挖掘：概念和技术。摩根考夫曼，旧金山，2001。
34. D.Hand、H.Mannela和P.Smyth。数据挖掘原理。麻省理工学院出版社，马萨诸塞州剑桥，2001年。
35. T.Hastie、R.Tibshirani和J.Friedman。统计学习的要素。数据挖掘、推理和预测。斯普林格，纽约，2001年。
36. T.H.Haveliwala和S.D.Kamvar。个性化页面排名方法的分析比较。技术报告，斯坦福大学计算机科学系，加利福尼亚州斯坦福，2003年。
37. M.Hegland公司。数据挖掘技术。活动编号，10:313–355，2001。
38. N.J.海厄姆。《数值算法的准确性和稳定性》，第2版，暹罗，费城，2002年。
39. I.C.F.Ipsen和S.Kirkland。Langville和Meyer提出的一种pagerank更新算法的收敛性分析。暹罗J.矩阵分析申请日期：2006年27:952–967。
40. E.R.杰塞普和J.H.马丁。对信息检索的潜在语义分析方法进行了新的探讨。《计算信息检索》，M.W.Berry，ed.，暹罗，费城，2001年，第121-144页。
41. S.D.Kamvar、T.H.Haveliwala和G.H.Golub。用于计算pagerank的自适应方法。线性代数应用，386:51–652003。
42. S.D.Kamvar、T.H.Haveliwala、C.D.Manning和G.H.Golub。利用Web的块结构计算pagerank。技术报告，斯坦福大学计算机科学系，加利福尼亚州斯坦福，2003年。
43. S.D.Kamvar、T.H.Haveliwala、C.D.Manning和G.H.Golub。用于加速pagerank计算的外推方法。第12届国际万维网会议，布达佩斯，2003年，第261-270页。
44. J.M.Kleinberg。超链接环境中的权威源。J.Assoc.计算。马赫数，46:604–632，1999年。
45. A.N.Langville和C.D.Meyer。更深入的页面排名。互联网数学，1:335–3802005。
46. A.N.Langville和C.D.Meyer。网络信息检索特征向量方法综述。暹罗版本，47:135–1612005。
47. A.N.Langville和C.D.Meyer。谷歌的网页排名及更高：搜索引擎排名的科学。普林斯顿大学出版社，新泽西州普林斯顿，2006年。
48. L.de Lathaower、B.de Moor和J.Vandewalle。多行奇异值分解。暹罗J.矩阵分析申请，21:1253–12782000.
49. C·L·劳森和R·J·汉森。解最小二乘问题。应用程序中的经典。数学。15。暹罗，费城，1995年。1974年由普伦蒂斯-霍尔首次出版的作品的修订再版。
50. Y.Lecun、L.Bottou、Y.Bengio和P.Haffner。基于梯度学习的文档识别方法。PROCIEEE，86:2278–23241998年11月。
51. D.Lee和H.Seung。用非负矩阵分解法学习物体的各个部分。《自然》，401:788–7911999年10月。
52. R.B.Lehoucq、D.C.Sorensen和C.Yang。arpack用户指南：使用隐式重新启动的arnoldi方法解决大规模特征值问题。暹罗，费城，1998年。
53. R.Lempel和S.Moran。链接结构分析的随机方法。ACM传输通知。系统，19:131–1602001。
54. O.Mangasarian和W.Wolberg。通过线性规划进行癌症诊断。暹罗新闻，1990年23:1,18。
55. i.马尼。自动汇总。John Benjamins，阿姆斯特丹，2001年。
56. matlab用户指南。Mathworks，Inc.，Natick，MA，1996年。
57. 中东和北非地区数据挖掘您的网站。数字出版社，波士顿，1999年。
58. C.D.迈耶。矩阵分析和应用线性代数。暹罗，费城，2000年。
59. C.莫勒。世界上最大的矩阵计算。Matlab新闻与注释，2002年10月，第12-13页。
60. J.L.Morrison、R.Breitling、D.J.Higham和D.R.Gilbert。Generank：使用搜索引擎技术对微阵列实验进行分析。BMC生物信息，2005年6:233。
61. P.Paatero和U.Tapper。正矩阵因式分解：一种非负因子模型，能最大限度地利用数据值的误差估计。环境测量学，5:111–126，1994年。
62. L.Page、S.Brin、R.Motwani和T.Winograd。pagerank引文排名：为网络带来秩序。斯坦福数字图书馆工作论文，斯坦福，加州，1998年。
63. C.C.Paige和M.Saunders。LSQR：稀疏线性方程和稀疏最小二乘法的一种算法。ACM传输数学。软件，8:43–711982。
64. H.Park、M.Jeon和J.Ben Rosen。基于向量空间的信息检索中文本数据的低维表示。《计算信息检索》，M.W.Berry，ed.，暹罗，费城，2001年，第3-23页。
65. H.Park、M.Jeon和J.B.Rosen。基于形心和最小二乘的文本数据的低维表示。位，43:427–448，2003。
66. V.P.Pauca、J.Piper和R.Plemmons。光谱数据分析的非负矩阵分解。线性代数应用，416:29–472006。
67. 萨阿德。大特征值问题的数值方法。曼彻斯特大学出版社，英国曼彻斯特，1992年。
68. Y.Saad.稀疏线性系统的迭代方法，第2版，暹罗，

费城，2003年。

1. G.Salton、C.Yang和A.Wong。用于自动索引的向量空间模型。通信协会计算机马赫，18:613–6201975年。
2. 萨瓦。手写数字算法的分析与测试。林科大学数学系硕士论文，2002年。
3. J.D.Schlesinger、J.M.Conroy、M.E.Okurowski、H.T.Wilson、D.P.O'Leary、A.Taylor和J.Hobbs。了解机器性能在人类性能环境下的多文档摘要。在2002年的Duc02会议记录中。网址：http://duc.nist.gov/pubs.html。
4. S.Serra Capizzano.公司谷歌矩阵的标准形式：对pagerank计算的潜在贡献。暹罗J.矩阵分析申请日期：2005年27:305–312。
5. F.Shahnaz、M.Berry、P.Pauca和R.Plemmons。使用非负矩阵分解来记录聚类。j.通知。PROC管理，42:373–386，2006年。
6. P.Simard、Y.Le Cun和J.S.Denker。利用新的变换距离进行有效的模式识别。在神经信息处理系统5的进展中，J. D. Cowan，S. J. Hanson，C. L. Giles，EDS，摩根Kaffman，旧金山，1993，第50 - 58页。
7. P.Y.Simard、Y.A.Le Cun、J.S.Denker和B.Victorri。模式识别切线距离和切线传播的变换不变性。实习生。J.成像系统技术，11:181–194，2001.
8. L.Sirovich和M.Kirby。人脸特征化的低维程序。J.光学SoC.埃默。A，1987年4:519–524。
9. M.Sjo–str–om和S.Wold。基于主成分模型的模式识别方法。在实践中的模式识别，E.S.Gelsema和L.N.Kanal，eds.，北荷兰，阿姆斯特丹，1980年，第351-359页。
10. P.Smaragdis和J.Brown。复调音乐转录的非负矩阵分解。美国电气与电子工程师协会（IEEE）音频和声学信号处理应用研讨会，2003年，第177-180页。
11. A.Smilde、R.Bro和P.Geladi。多途径分析：在化学科学中的应用。约翰·威利，纽约，2004年。
12. G.W.斯图尔特。矩阵算法：基本分解。暹罗，费城，1998年。
13. G.W.斯图尔特。矩阵算法第二卷：特征系统。暹罗，费城，2001年。
14. G.W.Stewart和J.-G.Sun。矩阵摄动理论。学术出版社，波士顿，1990年。
15. 特南鲍姆和弗里曼。用双线性模型分离样式和内容。神经计算，12:1247–12832000。
16. M.Totty和M.Mangalindan。随着谷歌成为网络的守门人，网站争相进入。《华尔街日报》，39，2003年2月26日。
17. L.N.Trefethen和D.B.Bau，III.数值线性代数，暹罗，

费城，1997年。

1. L.R.塔克。因子分析在三维矩阵中的推广。《对数学心理学的贡献》，H.Gulliksen和N.Frederiksen，eds.，Holt，Rinehart和Winston，纽约，1964年，第109-127页。
2. L.R.塔克。关于三模因子分析的一些数学注记。《心理计量学》，1966年，31:279–311。
3. M.A.Turk和A.P.Pentland。特征用于识别。《认知神经科学杂志》，3:71–86，1991年。
4. 范登伯根。交互式三维环境中的碰撞检测。摩根考夫曼，旧金山，2004。
5. 瓦西里斯库先生。人体运动特征：分析、合成、识别。2002年，加拿大魁北克市，模式识别国际会议（ICPR'02）。
6. M.A.O.Vasilescu和D.Terzopoulos。图像集合的多行分析：张量面。第7届欧洲计算机视觉会议（ECCV'02），哥本哈根，丹麦，计算机科学讲义2350，斯普林格出版社，纽约，2002年，第447-460页。
7. M.A.O.Vasilescu和D.Terzopoulos。用于面部识别的多行图像分析。模式识别国际会议（ICPR'02），加拿大魁北克市，2002年，第511-514页。
8. M.A.O.Vasilescu和D.Terzopoulos。图像合成的多行子空间分析。美国电气与电子工程师协会计算机视觉与模式识别会议（CVPR'03），威斯康星州麦迪逊，2003年，第93-99页。
9. P.˚A.韦丁。伪逆扰动理论。比特，13:344–3541973。
10. J.H.威尔金森。具有原点偏移的三对角QR算法的全局收敛性。线性代数应用，1:409–420，1968。
11. I.H.Witten和E.Frank。数据挖掘。使用Java实现的实用机器学习工具和技术。摩根考夫曼，旧金山，2000。
12. H.沃尔德。潜变量软建模：非线性迭代偏最小二乘法。在概率论和统计学方面，为纪念M.S.Bartlett，J.Gani，ed.的论文，伦敦学术出版社，1975年。
13. S.Wold，A.Ruhe，H.Wold和W.J.Dunn，III.线性回归中的共线性问题。偏最小二乘（PLS）方法的广义逆。暹罗科学杂志。统计计算，1984年5:735–743。
14. S.Wold、M.Sjo–str–om和L.Eriksson。PLS回归：化学计量学的基本工具。化学计量学综合。实验室系统，58:109–1302001。
15. S.Wolfram。《数学书》，第4版，剑桥大学出版社，伦敦，1999年。
16. D.Zeimpekis和E.Gallopoulos。一个用于生成termdocument矩阵的matlab工具箱的设计。《聚类高维数据及其应用研讨会》正在进行，I.S.Dhillon，J.Kogan和J.Ghosh，eds.，加利福尼亚州纽波特海滩，2005年，第38-48页。
17. 哈扎。使用互补原理和句子聚类进行一般性总结和关键词提取。第25届国际ACM SIGIR信息检索研究与发展会议，坦佩雷，芬兰，2002年，第113-120页。

索引

1-范数，17分析，27矩阵，19稳定性，54向量，17带矩阵，29

2-范数，17，61带宽，29矩阵，19基，20，37向量，17矩阵，99

3模阵列，91正交，50

正交，38，65

绝对误差，17矢量，14，165，173

邻接矩阵，159双对角矩阵，81，196

艾特肯外推，159双向化

代数，多行，91户主，81

所有正交性，95 Lanczos–Golub–Kahan，80、84、85，ALS，见交替最小二乘142、146、201、206交替最小二乘，106偏最小二乘，206

角度，18双线性形式，172

动画，10生物信息学，3，108近似BLAS，14，207

低级，63，89，109，135，139，乳腺癌诊断，103

145，168凸起，195

164级

K级，135，165取消，11，43，76

阿诺迪癌症，103

分解，203质心，102，114，139方法，159，203，204，208近似，146

隐式重启，205化学计量学，92，94

递归，203 Cholesky分解，26，30，207

Arpack，72，208分类，75，114，120，127，172-

阵列174

n-模式，91簇，101，114，139 N-路，91相干，102阿特拉斯，207簇，139，165权威，159相干，簇，102

分数，159列旋转，72165

柱随机矩阵，150

反向完全正交分解，72误差，9，46，54压缩柱存储，200

二百一十七

压缩行存储，199个计算机游戏，10个计算机图形，10个概念向量，139

条件编号，26、34、35、69

坐标，14，50，89，103，165，173

核心张量，95，170

余弦距离，18，114，132，136，140，

一百四十一

数据

压缩，63，100，175

矩阵，66放气，66

质量，减少37，减少20

分解胆石，26，30，207

特征值，182，200

低密度脂蛋白，25

卢，24，207

三对角，30 qr，49，161，207柱旋转，72，165

薄，49

Schur，182、197、200、207综合楼，183部分，182、204

真的，182

奇异值，57、116、135、207

瘦，59

致密基质，31、42、179、185

因变量，75

行列式，179

对角线矩阵，12位，手写，6、91、97、113-

128距离

余弦，18，114，132，136，140，141欧几里得，17，113，122正切，122，124

文件

聚类，108，139加权，132，162

主特征值，152电子商务，3个特征值，169

Eigenspace，182年

特征值，150分解，182，200

优势，152

扰动，181183

问题，7灵敏度，180

相似变换，180

特征向量，150，196

摄动，181184

电子邮件监控，108方程，多项式，179

等效向量范数，17

错误

绝对，17向后，9，46，54

逆向分析，27

浮点，9，46，53

前进，9

相对，9，17

欧几里德

距离，17，113，122标准，123

向量范数，17，19

解释变量，75

人脸识别，172

国际足联，4，79，105

有限算法，190

浮点算术，9，46，155，190，196错误，9，46，53运算，8

溢出，10，156

标准（IEEE），9

下溢，10，156

失败，8

计数，45，53，195197

足球，4，80，110向前错误，9频率，期限，132

Frobenius Norm，19、40、64、92、99基本子空间，62

高斯变换，23高斯消去，23 Generank，159

Gilbert–Johnson–Keerthi算法，10

给定旋转，参见平面旋转

谷歌，4，7，79，104，109，147矩阵，153

克-施密特，90

图表

互联网，148

链接，7149

强连接，152 GTP，见文本分析器

手写数字，6，91，97，113–128分类，6，91

美国邮政数据库，6，

97、113、114、121、122、128

Hessenberg矩阵，197

点击，参见超文本诱导主题搜索

胡克定律，31

软管，94，170薄，96，170截短，175

户主双诊断，81矩阵，43转换，43，46，47，53，80，

188、196、197 HTML、132、161集线器、159

分数，159

超文本诱导的主题搜索，159

IEEE算术，9双精度，9浮点标准，9单精度，9

病态矩阵，27隐Q定理，194隐移位，194197

索引，130，137反转，130

无穷范数矩阵，19信息检索，3，4，103，129，

133，161，200初始化，SVD，108

inlink，7，148，159页

内部产品，15、17、92

互联网，3，4，7，147，164

图表，148

不变子空间，182

逆文件频率，132逆迭代，186，196，198逆矩阵，21，31逆索引，130不可约矩阵，151

k-均值算法，102，139 Kahan矩阵，74

Karhunen–Loewe扩建，58

Krylov子空间，80、89、201、203

Lanczos方法，72，208

双诊断，84，85

三对角化，206207

隐式重新启动，206207

Lanczos–Golub–Kahan Bidialogization，80、84、85、142、146，

201，206拉帕克，14，72，179，189，207潜在语义分析，135潜在语义索引，130，135，146乳胶，161，163

LDLT分解，25最小二乘，31，85交替，106

方法，32非阴性，106

正态方程，33，54

扰动，69

预测，75

问题，32、51、66、85、117解决方案

最小范数，70 qr分解，51

SVD，68岁

词汇扫描仪，163

图书馆目录，129线性独立性，20线性运算符，5线性系统，23

过度确定，23、31、32、51、66

微扰理论，26

不确定，71

链接，4148

农场，154图，7149

矩阵，7200

低质

近似值，63，89，109，135，

139，145，168矩阵，63

LSA，见潜在语义分析LSI，见潜在语义索引

LU分解，24207

三对角线，30

机器学习，161多功能，123标记语言，161

马尔可夫链，150

数学，8，208

matlab，8个

矩阵，4 2-范数，19邻接，159

近似值，63–65，116波段，29基，99

Bidiagnal，81，196年

柱随机，150密，31，42，179，185对角线，12因子非负，102，106，141，146，

161165168年

谷歌，153

海森堡，197

户主，43个病态，27个倒数，21，31个不可约，151

Kahan，74链接图，7200

低级，63

乘法，15，93

外部产品，16

非奇异，21

空白，61

正交，39

排列，24，72，165

阳性，152

正定，25

范围，61，182

等级，21，等级1，21，152

秩亏，70矩形，23可约，151

反射，43

旋转，40、47、55、197

稀疏，5132，163，185，198，200，

208存储，199，200

对称，25条款文件，4，91，104，130，

131，135有期徒刑，162

过渡，150三角形，23三对角，29，188，197，205

上准三角形，182上三角形，47

矩阵范数，18

1-标准，19

2-标准，61

弗罗贝尼乌斯，19，40，64

无穷范数，19

[matrix-vector multiplication, 13](#_Toc559255)

[max-norm, 17](#_Toc559256)

[vector, 17](#_Toc559257)

[medical abstracts, 129](#_Toc559258)

medline，129，136，140，142，144，145微阵列，159

模式，91型，降级，77，115 MPI，207

多线性代数，91乘法i模，92矩阵，15，93

矩阵向量，13张量矩阵，92

音乐转录，108相互强化原理，162

n路数组，91自然语言处理，161 netlib，207

网络分析，159噪声

还原，145去除，63

非负最小二乘，106非负矩阵因式分解，102，

106，141，146，161，165，168非奇异矩阵，21

定额

1-标准，17欧几里得，123

矩阵，18

1-标准，19

2-标准，61

弗罗贝尼乌斯，19，40，64

无穷大，最大值19，运算符17，p-范数18，17

张量，92

弗罗贝尼乌斯，92，99

矢量，17

欧几里得方程，17，19正态方程，33，54零空间，61数值秩，63，72，76

算子范数，18正交

基础，50分解，完整，72

矩阵，39相似变换，180，187

转换，浮点，46个向量，18，38

正交

基准，38，65

向量，38

外部产品，16，59 outlinek，7，148，159 overdetermined system，23，31，32，51，

66溢出，10，156

p-norm，vector，17 pagerank，147–159161解析器，文本，132，161，163

偏最小二乘，见PLS偏轴，23，30模式识别，6

PCA，见主成分分析

性能建模，133排列矩阵，24，72，165

Perron–Frobenius定理，152

个性化向量，154

扰动

特征值，181，183特征向量，181，184

最小二乘法，69

理论，26，28，180

平面旋转，40、46、47、55、197

请参见对潜在结构的投影

多项式方程，179波特-斯坦默，131正定矩阵，25正矩阵，152

功率法，150，154，185，201，204精度，133预测，75预处理，130主成分，58分析，66169

回归，78，144

对潜在结构的投影，80、89，

142伪逆，71心理测量学，92，94

QR算法，179180收敛，194198

不对称，197

对称，190，192

QR分解，49、161、207

柱枢转，72，165

精简，49个更新，54个

QR功能，50查询，5，79，129–147，159

匹配，132–146

随机的

冲浪者，150步行，150范围，61，182

排名，21

数字，63、72、76

秩-1

近似值，164

矩阵，21，152

秩亏矩阵，70秩k近似，135，165

排名，4147148159

矢量，148

回忆，134个矩形矩阵，23个降阶模型，77，115个可降阶矩阵，151

反射矩阵，43回归，主成分，78，

144相对

误差，9，17剩余，75

再拉伸，90，204残余

相对，75矢量，32，117

旋转

吉文斯，40飞机，40，46，47，55，197

旋转矩阵，55，197

舍入误差，9

显著性得分，162

SAS，208号

萨克斯比，14，15岁

斯卡拉帕克，207

舒尔分解，182197200，

207部分，182，204

搜索引擎，3，7，130，147，161语义结构，135移位，186

隐式，194197

威尔金森，190西姆卡，121

相似变换，正交，

180，187单数

图像，116值，58，163

I型，95张量，95

向量，58，163

奇异值分解，57，94，116，130，135，163，165，168，

169，200，206，207计算，72，196扩展，59

Lanczos–Golub–Kahan方法，108，

206外积形式，59张量，94

瘦，59

截断，63、78、136

（张量的）切片，93软件，207

稀疏矩阵，5132，163，185，198，

200，208存储，199，200光谱分析，108弹簧常数，31 SPSS，208

波特·斯坦默，131岁

堵塞，130161

停止字，130161

强连通图，152个子空间

基本，62不变，182

Krylov，80，89，201，203摘要，案文，161–168最高法院判例，159冲浪者，随机，150

SVD，见奇异值分解

SVD功能，60，72 SVDS功能，72，108，207

对称矩阵，25

同义词提取，159

标记，161切线

距离，122，124平面，123

传送，153164张量，11，91–100，169–176

核心，95，170

SVD，94岁

展开，93张紧面，169

期限，130162

频率，132加权，132，162

术语文件矩阵，4，91，104，130，

131，135句话矩阵，162

测试集，114–116文本挖掘，103，129–146文本分析器，132，161

GTP，131号

TMG，163号

文本检索会议，145文本摘要，161–168定理

隐式Q，194

Perron–Frobenius，152薄

软管，96、170

QR，49

SVD，59 TMG，参见文本解析器跟踪，19，92训练集，91，114–116，120，127转换对角线双曲线，126

高斯，23岁

户主，43，46，47，53，80，

188196197年

正交，浮点，46平行双曲线，126

旋转，125缩放，126相似，正交，180，187

增厚，127

翻译，125

过渡矩阵，150

TREC, *see* Text Retrieval Conference

triangle inequality, 17, 18

triangular matrix, 23

tridiagonal matrix, 29, 188, 197, 205

truncated HOSVD, 175 truncated SVD, 63, 78, 136 Tucker model, 94

underdetermined system, 71 underflow, 10, 156

unfolding, 93

unit roundoff, 9, 27, 46, 192, 196 updating QR decomposition, 54

upper quasi-triangular matrix, 182

upper triangular matrix, 47

U.S. Postal Service database, 6, 97,

113, 114, 121, 122, 128

variable

dependent, 75 explanatory, 75

vector

basis, 14, 173

concept, 139 norm, 17

1-norm, 17 2-norm, 17 equivalence, 17 Euclidean, 17, 19

max-norm, 17

personalization, 154

ranking, 148 residual, 117 singular, 163

vector space model, 130, 146, 161 vectors

orthogonal, 18, 38 orthonormal, 38

volcanos on Venus, 3

Web page, 4

Web search engine, *see* search engine

weighting

document, 132, 162 term, 132, 162

Wilkinson shift, 190

XML, 132

Yale Face Database, 170

zip code, 113