



Mahout in Action

Mahout 实战

[美] Sean Owen Robin Anil 著
Ted Dunning Ellen Friedman
王斌 韩冀中 万吉 译

Apache基金会官方推荐

Mahout核心团队权威力作

大数据时代机器学习的实战经典



人民邮电出版社
POSTS & TELECOM PRESS

TURING

图灵程序设计丛书



Mahout in Action

Mahout 实战

[美] Séan Owen Robin Anil 著
Ted Dunning Ellen Friedman

王斌 韩冀中 万吉 译

人民邮电出版社
北京

图书在版编目 (C I P) 数据

Mahout 实战 / (美) 欧文 (Owen, S.) 等著 ; 王斌,
韩冀中, 万吉译. -- 北京 : 人民邮电出版社, 2014. 3
(图灵程序设计丛书)
书名原文: Mahout in action
ISBN 978-7-115-34722-0

I. ①M… II. ①欧… ②王… ③韩… ④万… III. ①
机器学习②电子计算机—算法理论 IV. ①TP181
②TP301.6

中国版本图书馆CIP数据核字 (2014) 第031399号

内 容 提 要

本书是 Mahout 领域的权威著作, 出自该项目核心成员之手, 立足实践, 全面介绍了基于 Apache Mahout 的机器学习技术。本书开篇从 Mahout 的故事讲起, 接着分三部分探讨了推荐系统、聚类和分类, 最后的附录涵盖 JVM 调优、Mahout 数学知识和相关资源。

本书适合所有数据分析和数据挖掘人员阅读, 需要有 Java 语言基础。

◆ 著 [美] Sean Owen Robin Anil Ted Dunning
Ellen Friedman

译 王 斌 韩冀中 万 吉

责任编辑 毛倩倩

执行编辑 刘 帅

责任印制 焦志炜

◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路11号

邮编 100164 电子邮件 315@ptpress.com.cn

网址 <http://www.ptpress.com.cn>

三河市海波印务有限公司印刷

◆ 开本: 800×1000 1/16

印张: 21.25

字数: 502千字

印数: 1—4 000册

2014年3月第1版

2014年3月河北第1次印刷

著作权合同登记号 图字: 01-2011-7805号

定价: 79.00元

读者服务热线: (010)51095186转600 印装质量热线: (010)81055316

反盗版热线: (010)81055315

广告经营许可证: 京崇工商广字第 0021 号



前言

追溯这本书的由来，就我个人而言，要从2005年说起。我的一个朋友当时正在创办一家公司，急需协同过滤技术。虽然当时可以找到成熟、开源的软件包，但是它们要么太过繁复，要么太学术化。所以，我决定从零开始，为这个朋友的创业公司开发了一个推荐系统的简单原型。遗憾的是，这家创业公司夭折了。然而，我却无法说服自己删除这个原型。它实在有趣，于是我对它进行整理并写了文档，用Taste这个名字将它发布为一个开源项目。

一年无声地过去了。我在业余时间为其增加了一些代码，并修复了一些问题。接着，有一两个用户出现，并提交了一些软件bug和补丁，然后又有了几个用户，再后来又增加了好一些。到了2008年，虽然小但却稳定的用户群形成了。后来，Apache Lucene的人把机器学习相关的部分剥离出来形成了Apache Mahout，他们建议把我们的两个项目进行合并。此后，本书在2009年晚些时候开始立项。而今，当看到这个项目滚雪球般地发展到2011年，并开始被大公司在生产系统中使用，我自己既惊讶又欣喜。

的确，我只是无心插柳。即便我已经是一个高级工程师，曾在谷歌工作过，也没有人会误认为我是这个领域的专家。我更像是一个博物馆的管理者，而不是一个画家，我将一个领域的伟大思想进行搜集、组织和打包，使之广为所用。这同样不失为一项有用的工作。

一些人在读过本书的初稿之后，说它是一本“通俗易懂”的机器学习书。这是一种盛誉，而我完全赞同。机器学习有其魔力所在，不过这个领域中有许多研究性的著作对于非专业人员而言就像天书，它们也与该技术的应用实践相去甚远。而本书旨在让读者易于理解，为爱好者揭示领悟的快乐，并为实践者节省工作时间。我希望你阅读本书时的惊喜比疑问多。

——Sean Owen

我对机器学习的兴趣可以回溯到2006年上大学的那段日子。那时，我作为实习生和一组人共同设计一个个性化的推荐引擎。这个小组后来成长为Minekey公司；我也被邀请加入，成为其核心开发人员。后来的四年，我一直从事机器学习技术的实现与试验。在此期间，我偶然间发现了Mahout，并开始作为一个Google Summer of Code的参赛学生加入这个项目。我记得，接下来的事情就是不断为它的代码库贡献算法和补丁，做性能调优，以及帮助邮件列表中的其他人。

由机器学习开发者、研究者和爱好者组成的社区非常出色，正在不断成长，而我有幸成为这个团队的一员。随着越来越多的公司采用Mahout，它正在成为机器学习的主流软件库。我衷心希望你在阅读本书时能够乐在其中。

——Robin Anil

我（Ted）在机器学习上是先做研究后做项目。我早期从事的是学术工作，后来参与了一些创业团队，从而得以将机器学习在实际中应用。

我（Ellen）以前在生物化学和分子生物学实验室工作。在研究大量数据的同时，我还写了许多技术文章。此番经历，让我痴迷于数据及其蕴含的意义。我努力把这种内在的东西写进本书里。

我们两个人一致认为开源有赖于一个有大量活跃用户参与的社区。Mahout的成功主要来自于那些使用这个软件的人，他们通过在邮件列表中展开讨论、修复bug，以及提供建议，把使用经验回馈到这个项目中。

为此，本书不仅给出代码的实用注解，而且引出了代码背后的一些概念。介绍隐藏于代码背后的框架，会使你更有效地加入到Mahout的讨论中，并从中获益。我们希望本书不仅能够帮助读者，而且能够使Mahout自身得到完善和发展。

——Ted Dunning和Ellen Friedman

致谢

本书的出版离不开众人的努力。作者对他们致以衷心的感谢，但限于篇幅，致谢名单只列出了其中一部分人，排名不分先后。

- ❑ 在机器学习领域发表核心文章的研究者，详见附录C。
- ❑ 花时间测试试用版软件的Mahout用户，他们寻找与解决bug，为软件打补丁乃至提出建议。
- ❑ Mahout提交者，他们致力于Mahout的发展、完善和提升。
- ❑ Manning出版社投入了大量时间和精力将本书出版并投入市场。特别感谢Katharine Osborne、Karen Tegtmeyer、Jeff Bleiel、Andy Carroll、Melody Dolab和Dottie Marsico，你看到的最终稿与他们的工作密不可分。
- ❑ 在本书写作过程中提供了宝贵反馈的审校者：Philipp K. Janert、Andrew Oswald、John Griffin、Justin Tyler Wiley、Deepak Vohra、Grant Ingersoll、Isabel Drost、Kenneth DeLong、Eric Raymond、David Grossman、Tom Morton，以及Rick Wagner。
- ❑ Alex Ott在本书印刷前一刻对全稿进行了细致的技术审核。
- ❑ 在作者在线论坛上发帖评价本书的MEAP读者^①。
- ❑ 每个在Mahout邮件列表中提问的人。
- ❑ 在本书长时间写作过程之中，给予我们无尽支持的家人和朋友！

^① MEAP，全称Manning Early Access Program。因为图书出版周期较长，为让读者可以时刻了解热门技术，Manning出版社推出了这一图书抢鲜阅读项目。参与其中的读者可以在图书未编辑完成之际，一章一章地阅读。——编者注

关于本书

你可能还有疑问：这本书是否适合我？

如果你正在寻找一本机器学习教材，答案是否定的。本书不会对诸多算法和技术的理论以及推导过程给出全面解释。如果你了解机器学习技术，并熟悉矩阵和向量等相关的数学概念，这有助于阅读本书，但并非必要条件。

如果你正在开发先进、智能的应用，答案则是肯定的。本书从实践而非理论入手来诠释这些技术，并给出完整的例子与解决方案。它在教授用Mahout解决问题的同时，带给你实践者的经验与领悟。

如果你是人工智能、机器学习及相关领域的研究人员，答案亦是肯定的。你所面对的最大障碍，很可能就是把新的算法应用到实践中。对于新的大规模算法的测试与部署，Mahout提供了一个成熟的框架、一系列模式以及现成的组件。本书是你在复杂的分布式计算框架上学习开发机器学习系统的“快车票”。

如果你正领导一个产品小组或创业团队，想利用机器学习创造一种竞争优势，那么本书同样适合你。通过实际示例，它让你了解这些技术的多种应用方式。它还会让小型技术团队变得高效，能够处理大量数据，而这在以前只有具有大量技术资源的组织机构才做得到。

路线图

本书分为三部分，分别介绍了Apache Mahout中的协同过滤、聚类和分类。

首先，第1章整体介绍Apache Mahout。这一章为你阅读后续各章奠定基础。

第一部分（第2章~第6章）由Sean Owen编写，主要介绍协同过滤与推荐。第2章基于Mahout构造推荐引擎并评估其性能。第3章探讨如何高效呈现推荐引擎使用的数据。第4章介绍可在Mahout中利用的所有推荐算法，并比较它们的优缺点。在此背景之下，第5章给出一个案例，将第4章介绍的推荐系统实现应用在该案例的真实问题中，配以某些特定属性的数据，从而建立一个可为生产环境所用的推荐引擎。第6章介绍Apache Hadoop，通过研究基于Hadoop的推荐引擎，首次为你展现分布式环境中的机器学习算法。

第二部分（第7章~第12章）探索Apache Mahout上的聚类算法。通过Robin Anil所做的技术说明，你可以把看起来类似的数据片段组织为一个集合或者说簇（cluster）。聚类有助于揭示大规模数据中有趣的信息组合。这部分从聚类中的简单问题开始介绍，并给出了Java示例。接下来，

作者引入更多实际示例，并展示如何让Apache Mahout以Hadoop作业的方式运行，从而轻而易举地聚类大量数据。

第三部分（第13章~第17章）由Ted Dunning和Ellen Friedman编写，探索如何用Mahout进行分类。首先，作者带你了解如何通过一组示例“教会”一个算法，从而建立和训练分类器模型。接下来，你会了解如何评估并微调分类器模型以得到更好的结果。这部分最后以一个分类实战案例结束。

代码约定及下载

本书源代码均采用等宽字体印刷，列为代码清单，并对重点进行注释。代码清单旨在简单明了，重点突出。它们通常不给出Java导入包、类声明、Java注释，以及其他对代码的讨论无关紧要的东西。

本书中的类名亦采用等宽字体，放于文本之间，以显示它们是在可以在Apache Mahout源代码中找到并研究的类名。例如，LogLikelihoodSimilarity是Mahout中的一个Java类。

一些代码清单中列出了可执行命令。它们是为Mac OS X和Linux发行版等类Unix环境而写的。如果使用了类Unix的Cygwin环境，它们也可以在微软Windows系统下运行。

本书关键代码清单中的源代码均可编译，且均可从www.manning.com/MahoutinAction下载^①。这些都是独立的Java源文件，并不包括编译脚本。为了方便起见，你可以把它们解压到Mahout源代码发布包的examples/src/java/main目录下。这样，Mahout的编译环境将会自动编译这些代码。

多媒体资料

四位作者均录制了音频和视频片段，与书中多数章中的特定节互相补充，为相应话题提供了附加信息。你可以从本书英文版电子书中看到或听到这些音视频片段，该电子书对英文纸质书的拥有者免费；你还可以从www.manning.com/MahoutinAction/extras免费获取。通过书中的音频和视频图标，你可以获知其所涉及的话题，以及发言者是谁。这些多媒体资料的清单详见“关于多媒体资料”。

作者在线

本书英文版读者可以免费访问Manning出版社专门维护的一个论坛，并可以发表评论、提出技术问题，并获得作者和其他论坛用户的帮助。你可以通过网页www.manning.com/MahoutinAction进入和订阅该论坛。完成注册后，你可以了解如何使用该论坛、该论坛所能提供的帮助，以及论坛的行为规范。

Manning出版社承诺为读者和作者提供一个进行深入对话的场所，但不对作者的参与程度做要求，他们对于该论坛的贡献是出于自愿且无报酬的。我们建议读者尽量向作者提一些具有挑战性的问题，让他们保持兴趣！

本书在印期间，读者均可访问作者在线论坛，并查看之前的讨论。

^① 亦可在图灵社区（iTuring.cn）本书页面免费注册下载。——编者注

关于多媒体资料

本书附带的多媒体资料可以在www.manning.com/MahoutinAction/extras/上免费收听或收看。
本书中空白处的音频或视频文件图标（如下所示），指出了书中哪些地方可参考附加资料。



Audio icon



Video icon

- No. 1 音频 p2
Sean介绍了Mahout项目以及他参与的事项。
- No. 2 音频 p19
Sean讨论了推荐系统的工作。
- No. 3 音频 p29
Sean阐述为什么他认为人们有可能过度“聆听”数据。
- No. 4 音频 p42
Sean谈论皮尔逊相关系数的实现。
- No. 5 音频 p63
Sean讨论了诠释性能指标的价值。
- No. 6 音频 p84
Sean解释了Mahout和Hadoop之间的关系。
- No. 7 音频 p108
Robin解释了如何为一个数据集选择正确的距离测度方法。
- No. 8 音频 p114
Robin扩展了苹果的类比例。
- No. 9 音频 p127
Robin解释了k-means聚类迭代过程。
- No. 10 音频 p165
Robin讨论改善聚类质量的策略。
- No. 11 音频 p179
Robin解释了如何改进大规模聚类的性能。

- No. 12 视频 p208
Ellen展示了如何训练一个模型使之逐步优化。
- No. 13 视频 p234
Ted和Ellen展示了Logistic回归的内部机制。
- No. 14 视频 p238
Ted比较了使用串行算法与并行算法的优势。
- No. 15 音频 p249
Ted和Ellen讨论了AUC评估方法。
- No. 16 音频 p252
Ted和Ellen讨论了为什么对数似然法意味着“永不说不”。

关于封面

封面上是“一个来自Rakov-Potok的男人”。Rakov-Potok是克罗地亚北方的一个村庄。该图取自克罗地亚19世纪中叶传统服饰影集的一个副本，作者为Nikola Arsenovic，由Ethnographic博物馆在2003年出版于克罗地亚的斯普利特。该图得自于乐于助人的Ethnographic博物馆馆员，这个博物馆位于该城镇在中世纪罗马时的核心位置，是公元304年左右罗马皇帝戴克里先的宫殿遗址。这本书包含来自克罗地亚不同地域的颜色精美的插图，附有服饰和日常生活的说明。

Rakov-Potok是一个风景如画的乡村，位于Samobor山脚下、萨瓦河土地肥沃的河谷中，距Zagreb城不远。它有着悠久的历史，在那里，你会与许多城堡、教堂和中世纪甚至罗马时期的遗迹不期而遇。封面上的人物身着白色羊毛长裤和白色羊毛外套，上面有着大量的红色和蓝色绣花——这是该地区山区居民的典型装束。

过去200年间，人们的着装和生活方式已经发生变化，曾经如此丰富的地域多样性已渐渐消失了。现在，各大洲的居民已经很难分辨，更遑论不同小村或距离只有几英里的人。也许我们用文化多样性换来的是更多样化的个人生活——必然是更为丰富和快节奏的技术生活。

Manning出版社在此类古老书籍的插图中取材，基于两个世纪前丰富多样的地域生活来制作图书封面，借此颂扬计算机行业的创造力和首创精神。

目 录

2.5	评估 GroupLens 数据集.....	19
2.5.1	提取推荐程序的输入.....	19
2.5.2	体验其他推荐程序.....	20
2.6	小结.....	20

3.1	偏好数据的表示	21
3.1.1	Preference 对象	21
3.1.2	PreferenceArray 及其实现	22
3.1.3	改善聚合的性能	23
3.1.4	FastByIDMap 和 FastIDSet	23
3.2	内存级 DataModel	24
3.2.1	GenericDataModel	24
3.2.2	基于文件的数据	25
3.2.3	可刷新组件	25
3.2.4	更新文件	26
3.2.5	基于数据库的数据	26
3.2.6	JDBC 和 MySQL	27
3.2.7	通过 JNDI 进行配置	27
3.2.8	利用程序进行配置	28
3.3	无偏好值的处理	29
3.3.1	何时忽略值	29
3.3.2	无偏好值时的内存级表示	30
3.3.3	选择兼容的实现	31
3.4	小结	33

4.1	理解基于用户的推荐	34
4.1.1	推荐何时会出错	34
4.1.2	推荐何时是正确的	35
4.2	探索基于用户的推荐程序	36

4.2.1 算法	36	5.2.1 基于用户的推荐程序	61
4.2.2 基于 GenericUserBased- Recommender 实现算法	36	5.2.2 基于物品的推荐程序	62
4.2.3 尝试 GroupLens 数据集	37	5.2.3 slope-one 推荐程序	63
4.2.4 探究用户邻域	38	5.2.4 评估查准率和查全率	63
4.2.5 固定大小的邻域	39	5.2.5 评估性能	64
4.2.6 基于阈值的邻域	39	5.3 引入特定域的信息	65
4.3 探索相似性度量	40	5.3.1 采用一个定制的物品相似性 度量	65
4.3.1 基于皮尔逊相关系数的相似度	40	5.3.2 基于内容进行推荐	66
4.3.2 皮尔逊相关系数存在的问题	42	5.3.3 利用 IDRescorer 修改推荐 结果	66
4.3.3 引入权重	42	5.3.4 在 IDRescorer 中引入性别	67
4.3.4 基于欧氏距离定义相似度	43	5.3.5 封装一个定制的推荐程序	69
4.3.5 采用余弦相似性度量	43	5.4 为匿名用户做推荐	71
4.3.6 采用斯皮尔曼相关系数基于相 对排名定义相似度	44	5.4.1 利用 PlusAnonymousUser- DataModel 处理临时用户	71
4.3.7 忽略偏好值基于谷本系数计算 相似度	45	5.4.2 聚合匿名用户	73
4.3.8 基于对数似然比更好地计算相 似度	46	5.5 创建一个支持 Web 访问的推荐程序	73
4.3.9 推测偏好值	47	5.5.1 封装 WAR 文件	74
4.4 基于物品的推荐	47	5.5.2 测试部署	74
4.4.1 算法	48	5.6 更新和监控推荐程序	75
4.4.2 探究基于物品的推荐程序	49	5.7 小结	76
4.5 Slope-one 推荐算法	50	第 6 章 分布式推荐	78
4.5.1 算法	50	6.1 分析 Wikipedia 数据集	78
4.5.2 Slope-one 实践	51	6.1.1 挑战规模	79
4.5.3 DiffStorage 和内存考虑	52	6.1.2 分布式计算的优缺点	80
4.5.4 离线计算量的分配	53	6.2 设计一个基于物品的分布式推荐算法	81
4.6 最新以及试验性质的推荐算法	53	6.2.1 构建共现矩阵	81
4.6.1 基于奇异值分解的推荐算法	53	6.2.2 计算用户向量	82
4.6.2 基于线性插值物品的推荐算法	54	6.2.3 生成推荐结果	82
4.6.3 基于聚类的推荐算法	55	6.2.4 解读结果	83
4.7 对比其他推荐算法	56	6.2.5 分布式实现	83
4.7.1 为 Mahout 引入基于内容的 技术	56	6.3 基于 MapReduce 实现分布式算法	83
4.7.2 深入理解基于内容的推荐算法	57	6.3.1 MapReduce 简介	84
4.8 对比基于模型的推荐算法	57	6.3.2 向 MapReduce 转换: 生成用 户向量	84
4.9 小结	57	6.3.3 向 MapReduce 转换: 计算共 现关系	85
第 5 章 让推荐程序实用化	59	6.3.4 向 MapReduce 转换: 重新思 考矩阵乘	87
5.1 分析来自约会网站的样本数据	59		
5.2 找到一个有效的推荐程序	61		

6.3.5 向 MapReduce 转换: 通过部分乘积计算矩阵乘	87	8.3 从文档中生成向量	119
6.3.6 向 MapReduce 转换: 形成推荐	90	8.4 基于归一化改善向量的质量	123
6.4 在 Hadoop 上运行 MapReduce	91	8.5 小结	124
6.4.1 安装 Hadoop	92	第 9 章 Mahout 中的聚类算法	125
6.4.2 在 Hadoop 上执行推荐	92	9.1 k-means 聚类	125
6.4.3 配置 mapper 和 reducer	94	9.1.1 关于 k-means 你需要了解的	126
6.5 伪分布式推荐程序	94	9.1.2 运行 k-means 聚类	127
6.6 深入理解推荐	95	9.1.3 通过 canopy 聚类寻找最佳 k 值	134
6.6.1 在云上运行程序	95	9.1.4 案例学习: 使用 k-means 对新闻聚类	138
6.6.2 考虑推荐的非传统用法	97	9.2 超越 k-means: 聚类技术概览	141
6.7 小结	97	9.2.1 不同类型的聚类问题	141
第 2 部分 聚类		9.2.2 不同的聚类方法	143
第 7 章 聚类介绍	100	9.3 模糊 k-means 聚类	145
7.1 聚类的基本概念	100	9.3.1 运行模糊 k-means 聚类	145
7.2 项目相似性度量	102	9.3.2 多模糊会过度吗	147
7.3 Hello World: 运行一个简单的聚类示例	103	9.3.3 案例学习: 用模糊 k-means 对新闻进行聚类	148
7.3.1 生成输入数据	103	9.4 基于模型的聚类	149
7.3.2 使用 Mahout 聚类	104	9.4.1 k-means 的不足	149
7.3.3 分析输出结果	107	9.4.2 狄利克雷聚类	150
7.4 探究距离测度	108	9.4.3 基于模型的聚类示例	151
7.4.1 欧氏距离测度	108	9.5 用 LDA 进行话题建模	154
7.4.2 平方欧氏距离测度	108	9.5.1 理解 LDA	155
7.4.3 曼哈顿距离测度	108	9.5.2 对比 TF-IDF 与 LDA	156
7.4.4 余弦距离测度	109	9.5.3 LDA 参数调优	156
7.4.5 谷本距离测度	110	9.5.4 案例学习: 寻找新闻文档中的话题	156
7.4.6 加权距离测度	110	9.5.5 话题模型的应用	158
7.5 在简单示例上使用各种距离测度	111	9.6 小结	158
7.6 小结	111	第 10 章 评估并改善聚类质量	160
第 8 章 聚类数据的表示	112	10.1 检查聚类输出	160
8.1 向量可视化	113	10.2 分析聚类输出	162
8.1.1 将数据转换为向量	113	10.2.1 距离测度与特征选择	163
8.1.2 准备 Mahout 所用的向量	115	10.2.2 簇间与簇内距离	163
8.2 将文本文档表示为向量	116	10.2.3 簇的混合与重叠	166
8.2.1 使用 TF-IDF 改进加权	117	10.3 改善聚类质量	166
8.2.2 通过 n-gram 搭配词考察单词的依赖性	118	10.3.1 改进文档向量生成过程	166

10.3.2 编写自定义距离测度	169	13.2.2 分类的应用	201
10.4 小结	171	13.3 分类的工作原理	202
第 11 章 将聚类用于生产环境	172	13.3.1 模型	203
11.1 Hadoop 下运行聚类算法的快速入门	172	13.3.2 训练、测试与生产	203
11.1.1 在本地 Hadoop 集群上运行 聚类算法	173	13.3.3 预测变量与目标变量	204
11.1.2 定制 Hadoop 配置	174	13.3.4 记录、字段和值	205
11.2 聚类性能调优	176	13.3.5 预测变量值的 4 种类型	205
11.2.1 在计算密集型操作中避免性 能缺陷	176	13.3.6 有监督学习与无监督学习	207
11.2.2 在 I/O 密集型操作中避免性 能缺陷	178	13.4 典型分类项目的工作流	207
11.3 批聚类及在线聚类	178	13.4.1 第一阶段工作流: 训练分类 模型	208
11.3.1 案例分析: 在线新闻聚类	179	13.4.2 第二阶段工作流: 评估分类 模型	212
11.3.2 案例分析: 对维基百科文章 聚类	180	13.4.3 第三阶段工作流: 在生产中 使用模型	212
11.4 小结	181	13.5 循序渐进的简单分类示例	213
第 12 章 聚类的实际应用	182	13.5.1 数据和挑战	213
12.1 发现 Twitter 上的相似用户	182	13.5.2 训练一个模型来寻找颜色填 充: 初步设想	214
12.1.1 数据预处理及特征加权	183	13.5.3 选择一个学习算法来训练 模型	215
12.1.2 避免特征选择中的常见陷阱	184	13.5.4 改进填充颜色分类器的 性能	217
12.2 为 Last.fm 上的艺术家推荐标签	187	13.6 小结	221
12.2.1 利用共现信息进行标签推荐	187	第 14 章 训练分类器	222
12.2.2 构建 Last.fm 艺术家词典	188	14.1 提取特征以构建分类器	222
12.2.3 将 Last.fm 标签转换成以艺 术家为特征的向量	190	14.2 原始数据的预处理	224
12.2.4 在 Last.fm 数据上运行 k-means 算法	191	14.2.1 原始数据的转换	224
12.3 分析 Stack Overflow 数据集	193	14.2.2 一个计算营销的例子	225
12.3.1 解析 Stack Overflow 数据集	193	14.3 将可分类数据转换为向量	226
12.3.2 在 Stack Overflow 中发现聚 类问题	193	14.3.1 用向量表示数据	226
12.4 小结	194	14.3.2 用 Mahout API 做特征散列	228
第三部分 分类		14.4 用 SGD 对 20 Newsgroups 数据集进 行分类	231
第 13 章 分类	198	14.4.1 开始: 数据集预览	231
13.1 为什么用 Mahout 做分类	198	14.4.2 20 Newsgroups 数据特征的 解析和词条化	234
13.2 分类系统基础	199	14.4.3 20 Newsgroups 数据的训练 代码	234
13.2.1 分类、推荐和聚类的区别	201		

14.5 选择训练分类器的算法.....238	16.2 确定规模和速度需求.....270
14.5.1 非并行但仍很强大的算法： SGD 和 SVM.....239	16.2.1 多大才算大.....270
14.5.2 朴素分类器的力量：朴素贝 叶斯及补充朴素贝叶斯.....239	16.2.2 在规模和速度之间折中.....272
14.5.3 精密结构的力量：随机森林 算法.....240	16.3 对大型系统构建训练流水线.....273
14.6 用朴素贝叶斯对 20 Newsgroups 数据 分类.....241	16.3.1 获取并保留大规模数据.....274
14.6.1 开始：为朴素贝叶斯提取 数据.....241	16.3.2 非规范化及下采样.....275
14.6.2 训练朴素贝叶斯分类器.....242	16.3.3 训练中的陷阱.....276
14.6.3 测试朴素贝叶斯模型.....242	16.3.4 快速读取数据并对其进行 编码.....278
14.7 小结.....244	16.4 集成 Mahout 分类器.....282
第 15 章 分类器评估及调优.....245	16.4.1 提前计划：集成中的关键 问题.....283
15.1 Mahout 中的分类器评估.....245	16.4.2 模型序列化.....287
15.1.1 获取即时反馈.....246	16.5 案例：一个基于 Thrift 的分类服 务器.....288
15.1.2 确定分类“好”的含义.....246	16.5.1 运行分类服务器.....292
15.1.3 认识不同的错误代价.....247	16.5.2 访问分类器服务.....294
15.2 分类器评估 API.....247	16.6 小结.....296
15.2.1 计算 AUC.....248	第 17 章 案例分析——Shop It To Me.....297
15.2.2 计算混淆矩阵和熵矩阵.....250	17.1 Shop It To Me 选择 Mahout 的原因.....297
15.2.3 计算平均对数似然.....252	17.1.1 Shop It To Me 公司简介.....298
15.2.4 模型剖析.....253	17.1.2 Shop It To Me 需要分类系 统的原因.....298
15.2.5 20 Newsgroups 语料上 SGD 分类器的性能指标计算.....254	17.1.3 对 Mahout 向外扩展.....298
15.3 分类器性能下降时的处理.....257	17.2 邮件交易系统的一般结构.....299
15.3.1 目标泄漏.....258	17.3 训练模型.....301
15.3.2 特征提取崩溃.....260	17.3.1 定义分类项目的目标.....301
15.4 分类器性能调优.....262	17.3.2 按时间划分.....303
15.4.1 问题调整.....262	17.3.3 避免目标泄漏.....303
15.4.2 分类器调优.....265	17.3.4 调整学习算法.....303
15.5 小结.....267	17.3.5 特征向量编码.....304
第 16 章 分类器部署.....268	17.4 加速分类过程.....306
16.1 巨型分类系统的部署过程.....268	17.4.1 特征向量的线性组合.....307
16.1.1 理解问题.....269	17.4.2 模型得分的线性扩展.....308
16.1.2 根据需要优化特征提取过程.....269	17.5 小结.....310
16.1.3 根据需要优化向量编码.....269	附录 A JVM 调优.....311
16.1.4 部署可扩展的分类器服务.....270	附录 B Mahout 数学基础.....313
	附录 C 相关资源.....318
	索引.....320

第 1 章

初识Mahout



本章内容

- Apache Mahout是什么？从哪里来？
- 现实中的推荐引擎、聚类和分类一览
- Mahout安装

大概你已经从书名中猜到了，本书主要讲解一个特殊的工具——Apache Mahout——在现实生活中的高效应用。它具备三个明显的特征。

首先，Mahout是一个来自Apache的、开源的机器学习（machine learning）软件库。它所实现的算法归属于机器学习或集体智慧（collective intelligence，也常常译为群体智慧或群体智能）这个广阔的领域。这意味着有许多事情可做，但对于此时此刻的Mahout，它主要关注于推荐引擎（协同过滤）、聚类和分类。

其次，Mahout是可扩展的。它旨在当所处理的数据规模远大于单机处理能力时成为一种可选的机器学习工具。在当前的Mahout系统中，这些可扩展的机器学习实现都是用Java来写的，而且有些部分是建立在Apache的Hadoop分布式计算项目之上的。

最后，它是一个Java软件库，并不提供用户接口、预装服务器（prepackaged server）或安装程序（installer）。它打算为开发者提供一个可用可改的工具框架。

出于阶段安排的需要，本章将通过一些常见的真实案例简要介绍一下推荐引擎、聚类和分类这几种机器学习手法，而Mahout通过它们帮助你处理数据。

若要在阅读本书时做到对Mahout随学随用，还要做一些必要的系统搭建与安装工作。

1.1 Mahout 的故事

首先来了解Mahout的背景知识。你可能还搞不清Mahout该如何发音：就是其通常的英语发音（[mə'haut]），它和trout押韵。它来自北印度语，意为驱象人，若要解释它，还有个小故事。

Mahout是2008年作为Apache Lucene的子项目出现的。Lucene项目推出了一个同名的著名开源搜索引擎，并给出了搜索、文本挖掘（text mining）和信息检索技术的先进实现方法。在计算机科学领域，这些术语和机器学习技术中的概念很相近，比如聚类（clustering），并在某种程度

上与分类（classification）相近。这样一来，某些Lucene贡献者的工作更多落入机器学习领域，从而逐渐脱离出来形成了独立的子项目。之后不久，Mahout吸纳了开源的协同过滤项目Taste。



No. 1

图1-1给出了Mahout在ASF（Apache Software Foundation，Apache软件基金会）中的部分传承关系。到2010年4月，Mahout已经成为了一个独立的顶级Apache项目，并发布了一个全新的驱象人徽标。

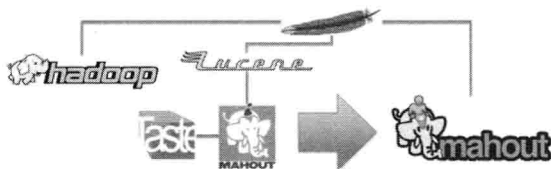


图1-1 Apache Mahout及其在ASF中的相关项目

Mahout所做的大量工作不仅体现在以高效和可扩展的方式实现这些经典算法，而且将部分算法进行转换使其可以在Hadoop上处理大规模的问题。Hadoop的吉祥物是一头象，Mahout项目的名字便由此而来！

从Mahout孵化出了许多技术和算法，其中有许多仍在开发或实验阶段（<https://cwiki.apache.org/confluence/display/MAHOUT/Algorithms>）。在该项目的早期，有3个明确的核心主题：推荐引擎（协同过滤）、聚类和分类。虽然它们绝非Mahout的全部，但在本书写作时，它们是最突出和最成熟的主题，也因此成为了本书的焦点。

也许你在阅读本书时已经了解了这三种技术的魅力，但为了不漏掉什么，请继续读下去。

1.2 Mahout 的机器学习主题

虽然Mahout项目在理论上可以实现所有类型的机器学习技术，但实际上当前它仅关注机器学习的三个主要领域，即推荐引擎（协同过滤）、聚类和分类。

1.2.1 推荐引擎

在目前采用的机器学习技术中，推荐引擎是最容易一眼就被认出来的。服务商或网站会根据你过去的行为向你推荐书籍、电影或文章。它们会推测你的品味与爱好，并找到某些你可能感兴趣的物品。

- 在部署了推荐系统的电子商务网站中，亚马逊大概是最有名的。亚马逊基于交易行为和网站记录为你推荐你可能感兴趣的书籍和其他物品（见图1-2）。
- 与之类似，Netflix为用户推荐其可能感兴趣的DVD，为了鼓励研究者改善其推荐质量，它给出了一份1 000 000美元的奖金，这使它颇具盛名。
- 像Libimseti这样的约会网站（稍后讨论）还能把一个人推荐给另一个人。