云计算与大数据实验教材系列

Mahout 实验指南

主编　李琳　袁景凌　熊盛武

WUHAN UNIVERSITY PRESS
武汉大学出版社

云计算与大数据实验教材系列

Mahout 实验指南

主编 李琳 袁景凌 熊盛武

WUHAN UNIVERSITY PRESS
武汉大学出版社

图书在版编目(CIP)数据

Mahout 实验指南/李琳,袁景凌,熊盛武主编. —武汉:武汉大学出版社,2017.4

云计算与大数据实验教材系列

ISBN 978-7-307-12769-2

Ⅰ.M… Ⅱ.①李… ②袁… ③熊… Ⅲ.①机器学习—教材 ②电子计算机—算法理论—教材 Ⅳ.①TP181 ②TP301.6

中国版本图书馆 CIP 数据核字(2017)第 033057 号

责任编辑:叶玲利 责任校对:李孟潇 版式设计:马 佳

出版发行:**武汉大学出版社** (430072 武昌 珞珈山)

(电子邮件:cbs22@whu.edu.cn 网址:www.wdp.com.cn)

印刷:湖北民政印刷厂

开本:787×1092 1/16 印张:4.5 字数:108 千字 插页:1

版次:2017 年 4 月第 1 版 2017 年 4 月第 1 次印刷

ISBN 978-7-307-12769-2 定价:18.00 元

前　言

本书是一本数据挖掘和机器学习领域入门阶段的实验教材，每章由知识要点和实验两个部分组成。知识要点一是对学生实验过程中碰到的概念和算法进行简要的介绍和讨论；二是对数据挖掘和机器学习领域中常见知识点的理解给出较为完整的方法和思路。实验部分是典型应用的例子，使学生能够将理论内容和实际应用有机结合，解决学生学以致用的问题。本书最大的特点是知识点和实验的结合，基于 Mahout 工具包，针对每章的内容设计了大量的实验，帮助学生更好地理解、掌握理论内容。

Mahout 是 Apache Software Foundation(ASF) 旗下的一个开源项目，提供一些可扩展的机器学习领域经典算法的实现，旨在帮助开发人员更加方便快捷地创建智能应用程序。Mahout 包含许多实现，包括聚类、分类、推荐过滤、频繁项集挖掘。此外，通过使用 Apache Hadoop 库，Mahout 可以有效地扩展到云中。Mahout 是一个很强大的数据挖掘工具，是一个分布式机器学习算法的集合，最大的优点就是基于 hadoop 实现，把很多以前运行于单机上的算法，转化为 MapReduce 模式，这样大大提升了算法可处理的数据量和处理性能。

《Mahout 实验指南》与数据科学中的“数据挖掘”和“机器学习”理论内容相得益彰，理论内容为学生提供了算法原理和思想，实验指南提供在 Mahout 平台上运用理论内容和技术，配置和实现各种算法的步骤和方法。学生用知识要点的理论内容指导实验，反过来又通过实验来加深理解算法的基本概念、原理和流程，具有设计和运用算法的能力，了解算法应用的途径，达到理论和实践相互提升的教学目标。

本书由武汉理工大学计算机科学与技术学院的李琳、袁景凌、熊盛武和研究生林伟彬和晁朝辉共同编写，由李琳定稿。限于作者的水平，错误和不足之处在所难免，殷切希望使用本书的老师和学生批评和指正，也殷切希望读者能够就本书内容和叙述方式提出宝贵建议和意见，以便进一步完善。作者的 E-mail 地址为 cathylilin@ whut. edu. cn。

编者

2017 年 1 月

目　录

1　概述 …… 1
1.1　数据挖掘 …… 1
1.1.1　推荐系统 …… 1
1.1.2　聚类算法 …… 1
1.1.3　分类算法 …… 2
1.1.4　监督学习和无监督学习 …… 2
1.1.5　关联规则 …… 2
1.2　Mahout 使用说明 …… 3
1.2.1　关于 Mahout …… 3
1.2.2　配置 Mahout …… 3

2　推荐系统 …… 10
2.1　知识要点 …… 10
2.1.1　推荐系统定义 …… 10
2.1.2　查准率与查全率 …… 10
2.1.3　协同过滤 …… 11
2.1.4　相似度计算 …… 11
2.2　创建一个推荐程序 …… 13
2.2.1　创建输入 …… 13
2.2.2　运行推荐程序 …… 14
2.3　评估一个推荐程序 …… 15
2.4　基于用户的协同过滤 …… 16
2.4.1　算法思想 …… 16
2.4.2　基于欧几里得距离的 user-based 推荐程序 …… 16
2.5　基于商品的协同过滤 …… 17
2.5.1　算法思想 …… 17
2.5.2　基于欧几里得距离的 item-based 推荐程序 …… 18
2.6　Slope-one 推荐算法 …… 18
2.6.1　算法思想 …… 18
2.6.2　Slope-one 推荐程序 …… 19

3　聚类算法 …… 20
3.1　知识要点 …… 20
3.1.1　TFIDF 权重 …… 20
3.1.2　向量空间模型及距离度量 …… 21
3.1.3　k-means 聚类算法 …… 21
3.1.4　模糊 k-means 聚类算法 …… 24
3.2　聚类示例 …… 24
3.2.1　生成输入数据 …… 24
3.2.2　使用 Mahout 聚类 …… 25
3.3　使用各种距离度量 …… 28
3.3.1　欧氏距离测度 …… 28
3.3.2　平方欧氏距离测度 …… 28
3.3.3　曼哈顿距离测度 …… 29
3.3.4　余弦距离测度 …… 29
3.3.5　谷本距离测度 …… 29
3.4　数据向量化表示 …… 30
3.4.1　将数据转换为向量 …… 30
3.4.2　从文档中生成向量 …… 32
3.5　k-means 新闻聚类 …… 33
3.5.1　内存 k-means 聚类 …… 33
3.5.2　Hadoop 下的 k-means 新闻文本聚类 …… 34
3.6　模糊 k-means 新闻聚类 …… 36
3.6.1　内存模糊 k-means 聚类 …… 36
3.6.2　Hadoop 下的模糊 k-means 新闻文本聚类 …… 37

4　分类算法 …… 39
4.1　知识要点 …… 39
4.1.1　分类算法基本流程 …… 39
4.1.2　最近邻分类器 …… 40
4.1.3　逻辑回归分类算法 …… 41
4.1.4　SVM 分类算法 …… 42
4.1.5　朴素贝叶斯分类算法 …… 43
4.1.6　决策树 …… 44
4.1.7　随机森林分类算法 …… 45
4.1.8　人工神经网络分类器 …… 45
4.2　简单分类示例——填充颜色分类器 …… 46
4.2.1　查看数据 …… 47
4.2.2　训练模型 …… 47

4.3　文本分类算法准备工作…… 50
4.3.1　训练分类器流程…… 50
4.3.2　实现文本的词条化和向量化…… 50
4.4　逻辑回归新闻分类算法…… 52
4.4.1　准备数据集…… 52
4.4.2　模型建立与评估…… 53
4.4.3　部分运行过程…… 54
4.5　朴素贝叶斯新闻分类算法…… 55
4.6　隐马尔科夫模型…… 56

5　关联规则…… 58
5.1　知识要点…… 58
5.1.1　频繁项集发现…… 58
5.1.2　支持度和置信度…… 58
5.1.3　Apriori 关联规则挖掘算法…… 59
5.2　关联规则挖掘示例…… 59
5.2.1　发现频繁项集…… 59
5.2.2　产生关联规则…… 61

参考文献 …… 65

1 概　述

在数据工程和知识发现领域中，数据挖掘被公认为是一种有用的方法。数据挖掘从大量数据中提取有用的知识，主要目的是发现被隐藏的或者不是显而易见的信息。原始数据有着多种多样的形式，例如电子商务的交易数据、生物信息学研究领域中的基因表达等。近年来，关联规则挖掘、监督学习(分类算法)、无监督学习(聚类算法)等被深入地研究和广泛地应用。

1.1 数据挖掘

1.1.1 推荐系统

推荐系统的研究和竞赛如火如荼，推进了推荐方法的发展，使得基于海量数据的推荐方法得到了相应关注。用户的评分数据是推荐方法的主要处理对象，将用户、商品及评分表示成矩阵的数学表示形式，研究者提出了基于邻域的算法和基于模型的推荐算法。其中基于用户的协同过滤算法和基于物品的协同过滤算法是基于邻域算法中的两大重要算法。

基于用户的协同过滤算法的核心思想是通过计算用户与用户之间的相似性程度，并根据这些相似性找到与该用户最为相近的用户来预测。基于物品的协同过滤算法则是依据物品间的相似性程度来预测，较早的应用是亚马逊的商品推荐系统。此外，基于邻域的协同过滤算法中相似性的度量可以采用欧氏距离、余弦距离等，有相关研究通过改进相似度计算的方法来提高推荐质量。

1.1.2 聚类算法

聚类(clustering)，是将具有相同或类似属性的对象从数据集合中划分出来并形成簇或类，并且要求簇内的对象要尽可能保持较大的相似度，簇与簇之间的对象要尽可能具有较大的差异度。它是一种无监督的数据挖掘算法，训练数据集中并没有预定好的类标签。典型的聚类分析既可以作为一个单独的数据分析工具，也可以作为其他算法的预处理手段。

在 Mahout 中的聚类算法将聚类可视化为一个几何问题。聚类的核心是使用几何的技术表达不同距离的测量，找到一些重要的距离测量法和聚类的关系，平面上的聚类点与数据形成的类之间的相似性就可以表现出来。相关聚类算法主要由以下基本步骤组成。

(1)特征选择：选择任务相关信息，并具有最小的信息冗余度。特征工程往往是知识发现工作的基础。

(2)相似性度量：对两个特征向量，采取什么方式去计算相似性。

(3) 聚类标准：往往通过聚类函数或者是一些规则来表达。
(4) 聚类算法：选择一个合适的算法。
(5) 结果的校验：包括验证测试等。
(6) 对结果的解释：确定如何集成到具体应用当中。

1.1.3 分类算法

分类(classification)是预测类标签，而类标签是离散属性或者标称属性。分类通过具有类标签的训练数据集来建立模型并对新数据进行分类，是一种有监督的数据挖掘算法。比如通过图片对性别预测就是分类，判断一封邮件是垃圾邮件还是正常邮件也是分类。和分类比较相似的是数值预测，它是建立一个连续值函数来预测未知的或者是缺失的值。比如通过图片对年龄进行预测就是数值预测。或者建一个模型预测一只股票第二天的价格是多少，也是数值预测。分类和数值预测应用非常广泛，比如银行判断是否给这个客户发放信用卡，政府判断当前的交通情况下发生事故的风险是多少等。

分类的过程包括模型构建和模型使用。模型构建主要是建立用以描述预先定义好的类的分类器。在这里，每个样本都具有预先定义好的类标签，这些样本构成了模型训练的训练数据集，分类模型可以用分类规则、决策树或者是数学公式来表示。模型使用主要是对未知数据或者是未来的数据进行分类。在模型使用前，要评估模型的准确率。这要用到测试数据集。测试数据的类标签也是已知的，但测试数据集的类标签应该独立于训练数据集。如果评估后认为准确率是可以接受的，那就可以用来对新数据进行分类了。

1.1.4 监督学习和无监督学习

在前面的两个部分，引入了对标签数据的学习(监督学习或者分类)和未标记的数据学习(无监督学习或者聚类)两个算法。从直觉上来讲，大量的无标签数据是很容易获得的(例如，页面由搜索引擎爬取得到)，但是其中只有很小的一部分被打上了标签。研究者提出了半监督学习(或半监督式分类)，其目的是通过使用大量无标签数据以及解决问题，加上标记的数据，建立更好的分类方法。半监督式分类的方法中具有代表性的是自我训练、合作训练、衍生模型和基于图形的方法等。

1.1.5 关联规则

购物篮分析是频繁项集挖掘或关联分析的典型应用，目的是要通过大量顾客购买的数据发现，哪些商品经常被一起购买。对从超市 pos 终端收集到的大量顾客购买的商品数据进行分析中，一个很经典的例子就是啤酒与尿布。20 世纪 90 年代的美国沃尔玛超市，通过分析销售数据发现，“啤酒”与“尿布”会经常出现在同一个购物篮中。虽然这看上去有点奇怪，但利用这个规律，超市通过将啤酒与尿布摆放在相同的区域或一起促销等方法，很好地提高了这两件商品的销售收入。

能够支持类似应用的技术就是频繁模式挖掘或者关联分析，这也是数据挖掘领域最具影响力的技术之一。什么是频繁模式？频繁模式指的是数据集中频繁一起发生的模式，这里的模式可能是项集(比如放到购物车中的商品)、子序列(比如股票的一段价格走势)或

者子结构(比如社交网络中的人和相互关系)等。简单地说，关联分析旨在发现项集之间有趣的关联或相关性。

1.2 Mahout 使用说明

1.2.1 关于 Mahout

Mahout 是 Apache 旗下的开源机器学习库，本书主要给出 Mahout 在推荐引擎(协同过滤)、聚类、分类和关联规则挖掘四个方面的实验指南，通过现实中一些熟悉的案例简要介绍机器学习算法。

Mahout 也是可扩展的。Mahout 致力于成为需要处理的数据集非常大，也许已经大到远远超出单个机器能够处理的范围时的机器学习的工具。目前，Mahout 提供的机器学习的实现是用 Java 语言编写的，有一部分是基于 Apache Hadoop 分布式计算项目的。最后，Mahout 是一个 Java 库。它不提供用户接口、预先打包好的服务器或安装程序，它是一个将要被开发者使用和适应的工具框架。

在准备通过本书动手实战 Mahout 之时，要做一些必要的设置和安装。

1.2.2 配置 Mahout

为了使用接下来的章节提供的代码，需要安装一些工具，并假定使用 Mahout 的用户对 Java 开发已经有所掌握了。

Mahout 及其相关的框架都是基于 Java 的，因此它与平台无关，可以在任何一个可以运行 JVM 的平台上使用 Mahout。首先需要配置 Java 环境，请参考 http://www.cnblogs.com/xxx0624/p/4164744.html 获得相关文件。

➢ **配置 Java 环境**

1. 下载 JDK

(http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html)如：jdk-8u111-linux-i586.tar.gz。

注意：32 位/64 位系统，如果不符合，则在检验 JDK 是否安装成功的时候会报错(错误：无法执行二进制文件)。

2. 安装配置

1) $ sudo mkdir /usr/local/java

$ cd ~/.Downloads

$ sudo tar zxvf jdk-8u111-linux-i586.tar.gz -C /usr/local/java

2)设置环境变量

$ sudo gedit ~/.bashrc

```
# set java environment
export JAVA_HOME=/usr/local/java/jdk1.8.0_111
export JRE_HOME=${JAVA_HOME}/jre
export CLASSPATH=.:${JAVA_HOME}/lib:${JRE_HOME}/lib
export PATH=${JAVA_HOME}/bin:$PATH
```

sh ▾ Tab Width: 8 ▾

$ source ~/. bashrc

3)验证环境

$ java-version

```
wilben@wilben-virtual-machine:~/Downloads$ java -version
java version "1.8.0_111"
Java(TM) SE Runtime Environment (build 1.8.0_111-b14)
Java HotSpot(TM) Client VM (build 25.111-b14, mixed mode)
wilben@wilben-virtual-machine:~/Downloads$
```

图 1.1 验证 Java 环境

➢ Hadoop 1.2.1 安装教程

请参考 http：//www. cnblogs. com/xxx0624/p/4166095. html。

1)下载 Hadoop1. 2. 1

网址：https：//mirrors. tuna. tsinghua. edu. cn/apache/hadoop/common/hadoop-1. 2. 1/。

下载文件 hadoop-1. 2. 1. tar. gz。

2)安装配置

#cd /Downloads

#tar -zxvf hadoop-1. 2. 1. tar. gz -C /opt/

#gedit /opt/Hadoop-1. 2. 1/conf/Hadoop-env. sh

```
export JAVA_HOME=/usr/local/java/jdk1.8.0_111
export HADOOP_HOME=/opt/hadoop-1.2.1
export PATH=$PATH:/opt/hadoop-1.2.1/bin
```

#source/opt/Hadoop-1. 2. 1/conf/Hadoop-env. sh　　#配置文件生效

#gedit ~/. bashrc

```
# set hadoop environment
export HADOOP_HOME=/opt/hadoop-1.2.1
export HADOOP_CONF_DIR=$HADOOP_HOME/conf
export PATH=$PATH:$HADOOP_HOME/bin
export HADOOP_HOME_WARN_SUPPRESS=not_null
```

#source ~/. bashrc

3)验证环境

#hadoop version （注意：中间没有“-”）

```
wilben@wilben-virtual-machine:/opt/hadoop-1.2.1/bin$ hadoop version
Hadoop 1.2.1
Subversion https://svn.apache.org/repos/asf/hadoop/common/branches/branch-1.2 -r 1503152
Compiled by mattf on Mon Jul 22 15:23:09 PDT 2013
From source with checksum 6923c86528809c4e7e6f493b6b413a9a
This command was run using /opt/hadoop-1.2.1/hadoop-core-1.2.1.jar
```

图 1.2 验证 Hadoop 环境

➢ 配置 SSH 服务

1)安装 ssh

#sudo apt-get install openssh-server

#sudo apt-get install openssh-client

验证：#ps -e | grep sshd

```
wilben@wilben-virtual-machine:/opt/hadoop-1.2.1/conf$ sudo apt-get install opens
sh-client
Reading package lists... Done
Building dependency tree
Reading state information... Done
openssh-client is already the newest version (1:7.2p2-4ubuntu2.1).
openssh-client set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 408 not upgraded.
wilben@wilben-virtual-machine:/opt/hadoop-1.2.1/conf$ ps -e|grep sshd
 3736 ?        00:00:00 sshd
wilben@wilben-virtual-machine:/opt/hadoop-1.2.1/conf$
```

图 1.3 安装 SSH 环境

2)配置 SSH 本机免密登录

首先请确保防火墙都处于关闭状态，具体命令是 ufw disable。并确保安装 ssh openssh-server。

在主机 qiuchenl0 中执行以下命令：

①cd ~/. ssh （进入用户目录下的隐藏文件 . ssh）

②ssh-keygen -t rsa(用 rsa 生成密钥)

③cp id_rsa. pub authorized_keys(把公钥复制一份，并改名为 authorized_keys，这步执行完，应该 ssh localhost 就可以无密码登录本机了，可能第一次要密码)

④scp authorized_keysqiuchenl@ qiuchenl1：/home/qiuchenl/. ssh （把重命名后的公钥通过 ssh 提供的远程复制文件，复制到从机 qiuchenl1 上面)

⑤chmod 600 authorized_keys(更改公钥的权限，也需要在从机 qiuchenl1 中执行同样代码)

⑥ssh qiuchenl1(可以远程无密码登录 qiuchenl1 这台机子了，注意是 ssh 不是 sudo ssh。第一次需要密码，以后不再需要密码)

➢ 伪分布式模式配置

core-site. xml：Hadoop Core 的配置项，例如 HDFS 和 MapReduce 常用的 I/O 设置等。

hdfs-site. xml：Hadoop 守护进程的配置项，包括 namenode、辅助 namenode 和 datanode 等。

mapred-site. xml：MapReduce 守护进程的配置项，包括 jobtracker 和 tasktracker。

1）新建文件夹

```
#mkdir tmp
#mkdir hdfs
#mkdir hdfs/name
#mkdir hdfs/data
```

2）编辑文件

①core-site. xml

```
<configuration>
<property>
<name>fs. default. name</name>
<value>hdfs：//localhost：9000</value>
</property>
 <property>
<name>hadoop. tmp. dir</name>
<value>/opt/Hadoop-1. 2. 1/tmp</value>
</property>y>
</configuration>
```

②hdfs-site. xml

```
<configuration>
<property>
<name>dfs. replication</name>
<value>1</value>
</property>
<property>
<name>dfs. tmp. dir</name>
<value>/opt/hadoop-1. 2. 1/hdfs/name</value>
</property>
<property>
<name>dfs. data. dir</name>
<value>/opt/hadoop-1. 2. 1/hdfs/data</value>
</property>
</configuration>
```

③mapred-site. xml

<configuration>

<property>

<name>mapred. job. tracker</name>

<value>localhost：9001</value>

</property>

</configuration>

3)格式化 HDFS

$ hadoopnamenode -format

如果出现这种错误：

ERROR namenode. NameNode： java. io. IOException： Cannot create directory /home/xxx0624/hadoop/hdfs/name/current

则：将 hadoop 的目录权限设为当前用户可写 sudo chmod -R a+w /opt/Hadoop-1. 2. 1，授予 hadoop 目录的写权限

另外同时还需要更改 hdfs/data 文件夹的读写权限：

$ sudo chmod 755 data

4)启动 Hadoop

$ cd /opt/Hadoop-1. 2. 1/bin

$ ./start-all. sh

$ jps

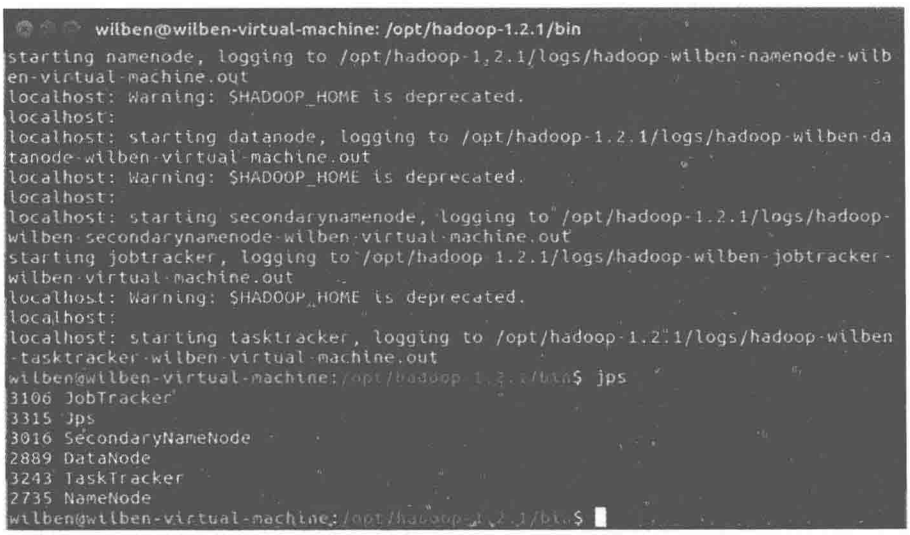

图 1.4 Hadoop 搭建

如图 1.4 所示，如果都列出来，说明搭建成功，漏一个都是有问题的。然后可以通过 firefox 浏览器查看运行状态。

http：//localhost：50030/ ——Hadoop 管理界面

http：//localhost：50060/ ——Hadoop Task Tracker 状态

http：//localhost：50070/　　——Hadoop DFS 状态

5)关闭 Hadoop

$ ./stop-all. sh

➢ **Mahout**

1)下载 Mahout 0. 6

地址：http：//archive. apache. org/dist/mahout/

mahout-distribution-0. 6. tar. gz

2)安装

#cd ~/. Downloads

#tar zxvf mahout-distribution-0. 6. tar. gz -C /opt/

3)配置

#gedit ~/. bashrc

```
# set mahout environment
export MAHOUT_HOME=/opt/mahout-distribution-0.6
export MAHOUT_CONF_DIR=$MAHOUT_HOME/conf
export PATH=$PATH:$MAHOUT_HOME/bin
```

#source ~/. bashrc

4)验证环境

$ mahout

```
wilben@wilben-virtual-machine: ~
wilben@wilben-virtual-machine:~$ mahout
MAHOUT_LOCAL is not set; adding HADOOP_CONF_DIR to classpath.
Running on hadoop, using HADOOP_HOME=/opt/hadoop-1.2.1
HADOOP_CONF_DIR=/opt/hadoop-1.2.1/conf
MAHOUT-JOB: /opt/mahout-distribution-0.6/mahout-examples-0.6-job.jar
An example program must be given as the first argument.
Valid program names are:
  arff.vector: : Generate Vectors from an ARFF file or directory
  baumwelch: : Baum-Welch algorithm for unsupervised HMM training
  canopy: : Canopy clustering
  cat: : Print a file or resource as the logistic regression models would see it
  cleansvd: : Cleanup and verification of SVD output
  clusterdump: : Dump cluster output to text
  clusterpp: : Groups Clustering Output In Clusters
  cmdump: : Dump confusion matrix in HTML or text formats
  cvb: : LDA via Collapsed Variation Bayes (0th deriv. approx)
  cvb0_local: : LDA via Collapsed Variation Bayes, in memory locally.
  dirichlet: : Dirichlet Clustering
  eigencuts: : Eigencuts spectral clustering
  evaluateFactorization: : compute RMSE and MAE of a rating matrix factorization
 against probes
  fkmeans: : Fuzzy K-means clustering
  fpg: : Frequent Pattern Growth
  hmmpredict: : Generate random sequence of observations by given HMM
```

图 1.5　验证 Mahout 环境

➢ 在 Eclipse 中创建 Mahout 工程

1)创建 Maven 工程

2)在 pom. xml 中引入 Mahout0. 6 相关的依赖 JAR 配置

```
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <mahout.version>0.6</mahout.version>
</properties>

<dependencies>
    <dependency>
        <groupId>junit</groupId>
        <artifactId>junit</artifactId>
        <version>3.8.1</version>
        <scope>test</scope>
    </dependency>

    <dependency>
        <groupId>org.apache.mahout</groupId>
        <artifactId>mahout-core</artifactId>
        <version>${mahout.version}</version>
    </dependency>
    <dependency>
        <groupId>org.apache.mahout</groupId>
        <artifactId>mahout-integration</artifactId>
        <version>${mahout.version}</version>
        <exclusions>
            <exclusion>
                <groupId>org.mortbay.jetty</groupId>
                <artifactId>jetty</artifactId>
            </exclusion>
            <exclusion>
                <groupId>org.apache.cassandra</groupId>
                <artifactId>cassandra-all</artifactId>
            </exclusion>
            <exclusion>
                <groupId>me.prettyprint</groupId>
                <artifactId>hector-core</artifactId>
            </exclusion>
        </exclusions>
    </dependency>
```

图 1.6　引入 Mahout0. 6 相关的依赖 JAR 配置

3)等待 JAR 下载完成即可

2 推荐系统

通过推荐系统的智能分析和挖掘，能够有效地帮助用户根据海量信息做出决策。本章的实验从简单推荐系统的实现入手，着重介绍了基于相似度计算的协同过滤方法和快速在线计算的 Slope-one 推荐方法，并对推荐系统性能进行了评估检验。

2.1 知识要点

2.1.1 推荐系统定义

推荐系统可以定义为一个软件代理，它能够智能地去分析用户的兴趣和喜好，同时根据用户的兴趣和喜好来进行推荐。一般来说，推荐系统的主要功能是预测。预测用户对一个没有买过的商品的购买可能性或者兴趣度。我们知道用户的喜好、用户的地域分布、年龄等各种用户属性信息，此外还知道用户对购买过的商品的打分情况以及商品的属性信息。推荐系统就会根据这些信息来形成最终的预测打分。主要方法有两大类：一类是基于内容的推荐，另外一类是基于协同过滤的推荐。第 1 章中提到的基于内存和基于模型的推荐都可以看做协同过滤方法。

基于内容的推荐主要根据用户以前买过的商品，预测哪个商品和用户以前买过的商品比较相似？换句话说就是“Show me more of the same what I've liked”。而协同过滤的思想是考虑朋友的兴趣或者购买历史。比如用户以前买过一些商品，同时用户的朋友也买过一些商品，协同过滤根据用户朋友购买的商品向用户进行推荐。还可以把各种推荐技术融合在一起，混合地进行推荐，我们称之为“混合推荐方法”。本章给出的实例以基于领域的协同过滤方法为主，围绕相似度(距离)度量、基于用户的协同过滤、基于商品的协同过滤和推荐评价四个方面展开学习。

2.1.2 查准率与查全率

对于一个给定的用户，评价推荐系统的质量或者推荐的准确度，可以采用查准率与查全率两个指标来衡量，它们也是信息检索领域常用的评价指标。信息检索和推荐系统都属于信息过滤，从大量信息中过滤得到和用户需求匹配的集合或者列表。

Precision(查准率、精度)衡量推荐系统或者检索系统得到的相关结果个数(商品，网页)与得到的结果总量的百分比。Recall(查全率，召回率)衡量推荐系统或者检索系统得到的相关结果个数(商品，网页)与系统中的相关结果总量的百分比。这样一个例子：某池塘有1 400条鲤鱼，300 只虾，300 只鳖。现在以捕鲤鱼为目的。撒一大网，逮着了 700