TURING 图灵程序设计丛书

[PACKT] PUBLISHING

[南非] Nick Pentreath 著　蔡立宇 黄章帅 周济民 译

Spark机器学习

Machine Learning with Spark

中国工信出版集团

人民邮电出版社
POSTS & TELECOM PRESS

TURING 图灵程序设计丛书

[南非] Nick Pentreath 著
蔡立宇 黄章帅 周济民 译

Spark机器学习

Machine Learning with Spark

人民邮电出版社
北 京

图书在版编目（CIP）数据

Spark机器学习 /（南非）彭特里思（Pentreath,N.）著 ；蔡立宇，黄章帅，周济民译. -- 北京 ：人民邮电出版社，2015.9
（图灵程序设计丛书）
ISBN 978-7-115-39983-0

Ⅰ. ①S… Ⅱ. ①彭… ②蔡… ③黄… ④周… Ⅲ. ①数据处理软件－机器学习 Ⅳ. ①TP274②TP181

中国版本图书馆CIP数据核字(2015)第176607号

内 容 提 要

本书每章都设计了案例研究，以机器学习算法为主线，结合实例探讨了 Spark 的实际应用。书中没有让人抓狂的数据公式，而是从准备和正确认识数据开始讲起，全面涵盖了推荐系统、回归、聚类、降维等经典的机器学习算法及其实际应用。

本书适合互联网公司从事数据分析的人员，以及高校数据挖掘相关专业的师生阅读参考。

◆ 著 [南非] Nick Pentreath
译 蔡立宇 黄章帅 周济民
责任编辑 李松峰
责任印制 杨林杰

◆ 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
北京鑫正大印刷有限公司印刷

◆ 开本：800×1000 1/16
印张：15
字数：355千字 2015年9月第1版
印数：1－4 000册 2015年9月北京第1次印刷
著作权合同登记号 图字：01-2015-2827号

定价：59.00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

版权声明

前　言

近年来，被收集、存储和分析的数据量呈爆炸式增长，特别是与网络、移动设备相关的数据，以及传感器产生的数据。大规模数据的存储、处理、分析和建模，以前只有Google、Yahoo!、Facebook和Twitter这样的大公司才涉及，而现在越来越多的机构都会面对处理海量数据的挑战。

面对如此量级的数据以及常见的实时利用该数据的需求，人工驱动的系统难以应对。这就催生了所谓的大数据和机器学习系统，它们从数据中学习并可自动决策。

为了能以低成本实现对大规模数据的支持，Google、Yahoo!、Amazon和Facebook涌现了大量开源技术。这些技术旨在通过在计算机集群上进行分布式数据存储和计算来简化大数据处理。

这些技术中最广为人知的是Apache Hadoop，它极大简化了海量数据的存储（通过Hadoop Distributed File System，即HDFS）和计算（通过Hadoop MapReduce，一种在集群里多个节点上进行并行计算的框架）流程，并降低了相应的成本。

然而，MapReduce有其严重的缺点，如启动任务时的高开销、对中间数据和计算结果写入磁盘的依赖。这些都使得Hadoop不适合迭代式或低延迟的任务。Apache Spark是一个新的分布式计算框架，从设计开始便注重对低延迟任务的优化，并将中间数据和结果保存在内存中。Spark提供简洁明了的函数式API，并完全兼容Hadoop生态系统。

不止如此，Spark还提供针对Scala、Java和Python语言的原生API。通过Scala和Python的API，Spark应用程序可充分利用Scala或Python语言的优势。这些优势包括使用相关的解释程序进行实时交互式的程序编写。Spark目前还自带一个分布式机器学习和数据挖掘工具包MLlib。经过重点开发，这个包中已经包括一些针对常见计算任务的高质量、可扩展的算法。本书会涉及其中的部分算法。

在大型数据集上进行机器学习颇具挑战性。这主要是因为常见的机器学习算法并非为并行架构而设计。大部分情况下，设计这样的算法并不容易。机器学习模型一般具有迭代式的特性，而这与Spark的设计目标一致。并行计算的框架有很多，但很少能在兼顾速度、可扩展性、内存处理和容错性的同时，还提供灵活、表达力丰富的API。Spark是其中为数不多的一个。

本书将关注机器学习技术的实际应用。我们会简要介绍机器学习算法的一些理论知识，但总的来说本书注重技术实践。具体来说，我们会通过示例程序和样例代码，举例说明如何借助Spark、MLlib以及其他常见的免费机器学习和数据分析套件来创建一个有用的机器学习系统。

本书内容

第1章“Spark的环境搭建与运行”，会讲到如何安装和搭建Spark框架的本地开发环境，以及怎样使用Amazon EC2在云端创建Spark集群。之后介绍Spark编程模型和API。最后分别用Scala、Java和Python语言创建一个简单的Spark应用。

第2章“设计机器学习系统”，会展示一个贴合实际的机器学习系统案例。随后会针对该案例设计一个基于Spark的智能系统所对应的高层架构。

第3章“Spark上数据的获取、处理与准备”，会详细介绍如何从各种免费的公开渠道获取用于机器学习系统的数据。我们将学到如何进行数据处理和清理，并通过可用的工具、库和Spark函数将它们转换为符合要求的数据，使之具备可用于机器学习模型的特征。

第4章“构建基于Spark的推荐引擎”，展示了如何创建一个基于协同过滤的推荐模型。该模型将用于向给定用户推荐物品，以及创建与给定物品相似的物品。这一章还会讲到如何使用标准指标来评估推荐模型的效果。

第5章“Spark构建分类模型”，阐述如何创建二元分类模型，以及如何利用标准的性能评估指标来评估分类效果。

第6章“Spark构建回归模型”，扩展了第5章中的分类模型以创建一个回归模型，并详细介绍回归模型的评估指标。

第7章“Spark构建聚类模型”，探索如何创建聚类模型以及相关评估方法的使用。你会学到如何分析和可视化聚类结果。

第8章“Spark应用于数据降维”，将通过多种方法从数据中提取其内在结构并降低其维度。你会学到一些常见的降维方法，以及如何对它们进行应用和分析。这里还会讲到如何将降维的结果作为其他机器学习模型的输入。

第9章“Spark高级文本处理技术”，介绍处理大规模文本数据的方法。这包括从文本提取特征以及处理文本数据常见的高维特征的方法。

第10章“Spark Streaming在实时机器学习上的应用”，对Spark Streaming进行综述，并介绍在流数据上的机器学习中它如何实现对在线和增量学习方法的支持。

预备知识

本书假设读者已有基本的Scala、Java或Python编程经验，以及机器学习、统计学和数据分析方面的基础知识。

本书目标

本书的预期读者是初中级数据科学研究者、数据分析师、软件工程师和对大规模环境下的机器学习或数据挖掘感兴趣的人。读者不需要熟悉Spark，但若具有统计、机器学习相关软件（比如MATLAB、scikit-learn、Mahout、R和Weka等）或分布式系统（如Hadoop）的实践经验，会很有帮助。

排版约定

在本书中，你会发现一些不同的文本样式，用以区别不同种类的信息。下面举例说明。

代码段的格式如下：

```
val conf = new SparkConf()
.setAppName("Test Spark App")
.setMaster("local[4]")
val sc = new SparkContext(conf)
```

所有的命令行输入或输出的格式如下：

```
>tar xfvz spark-1.2.0-bin-hadoop2.4.tgz

>cd spark-1.2.0-bin-hadoop2.4
```

新术语和重点词汇以楷体标示。屏幕、目录或对话框上的内容这样表示：“这些信息可以从AWS主页上依次点击‘**Account**’|‘**Security Credentials**’|‘**Access Credentials**’看到。”

这个图标表示警告或需要特别注意的内容。

这个图标表示提示或者技巧。

读者反馈

欢迎提出反馈。如果你对本书有任何想法，喜欢它什么，不喜欢它什么，请让我们知道。要写出真正对大家有帮助的书，了解读者的反馈很重要。

一般的反馈，请发送电子邮件至feedback@packtpub.com，并在邮件主题中包含书名。

如果你有某个主题的专业知识，并且有兴趣写成或帮助促成一本书，请参考我们的作者指南http://www.packtpub.com/authors。

客户支持

现在，你是一位自豪的Packt图书的拥有者，我们会尽全力帮你充分利用你手中的书。

下载示例代码

你可以用你的账户从http://www.packtpub.com下载所有已购买Packt图书的示例代码文件。如果你从其他地方购买本书，可以访问http://www.packtpub.com/support并注册，我们将通过电子邮件把文件发送给你。

勘误表

虽然我们已尽力确保本书内容正确，但出错仍旧在所难免。如果你在我们的书中发现错误，不管是文本还是代码，希望能告知我们，我们不胜感激。这样做可以减少其他读者的困扰，帮助我们改进本书的后续版本。如果你发现任何错误，请访问http://www.packtpub.com/submit-errata提交，选择你的书，点击勘误表提交表单的链接，并输入详细说明。勘误一经核实，你的提交将被接受，此勘误将上传到本公司网站或添加到现有勘误表。从http://www.packtpub.com/support选择书名就可以查看现有的勘误表。

侵权行为

互联网上的盗版是所有媒体都要面对的问题。Packt非常重视保护版权和许可证。如果你发现我们的作品在互联网上被非法复制，不管以什么形式，都请立即为我们提供位置地址或网站名称，以便我们可以寻求补救。

请把可疑盗版材料的链接发到copyright@packtpub.com。

非常感谢你帮助我们保护作者，以及保护我们给你带来有价值内容的能力。

问题

如果你对本书内容存有疑问，不管是哪个方面，都可以通过questions@packtpub.com联系我们，我们将尽最大努力来解决。

致　谢

过去一年里，本书的写作过程如同过山车一般跌宕起伏，伴随着熬夜和周末加班。对机器学习和Apache Spark的热爱让我受益良多，也希望本书能让读者有所收获。

非常感谢Packt出版团队在本书写作和编辑过程中提供的帮助，感谢Rebecca、Susmita、Sudhir、Amey、Neil、Vivek、Pankaj和所有为本书出过力的人。

同样感谢StumbleUpon公司的Debora Donato，她提供过数据和法律方面的协助。

写书的过程可能会让人感到孤立无援，因此审校人的反馈对保证本书的可读性，以及知晓还需要作出哪些调整十分有帮助。我深深地感谢Andrea Mostosi、Hao Ren和Krishna Sankar花费时间审阅本书，并提供细致且极为重要的反馈。

家人和朋友的不懈支持是本书得以写成的必要因素。特别是我的好妻子Tammy，感谢她在若干个夜晚和周末的陪伴与支持。谢谢你们所有人！

最后，谢谢你阅读这本书，希望它对你能有所帮助。

目　录

第1章 Spark的环境搭建与运行

Apache Spark是一个分布式计算框架，旨在简化运行于计算机集群上的并行程序的编写。该框架对资源调度，任务的提交、执行和跟踪，节点间的通信以及数据并行处理的内在底层操作都进行了抽象。它提供了一个更高级别的API用于处理分布式数据。从这方面说，它与Apache Hadoop等分布式处理框架类似。但在底层架构上，Spark与它们有所不同。

Spark起源于加利福利亚大学伯克利分校的一个研究项目。学校当时关注分布式机器学习算法的应用情况。因此，Spark从一开始便为应对迭代式应用的高性能需求而设计。在这类应用中，相同的数据会被多次访问。该设计主要靠利用数据集内存缓存以及启动任务时的低延迟和低系统开销来实现高性能。再加上其容错性、灵活的分布式数据结构和强大的函数式编程接口，Spark在各类基于机器学习和迭代分析的大规模数据处理任务上有广泛的应用，这也表明了其实用性。

关于Spark项目的更多背景信息，包括其开发的核心研究论文，可从项目的历史介绍页面中查到：http://spark.apache.org/community.html#history。

Spark支持四种运行模式。

- 本地单机模式：所有Spark进程都运行在同一个Java虚拟机（Java Vitural Machine，JVM）中。
- 集群单机模式：使用Spark自己内置的任务调度框架。
- 基于Mesos：Mesos是一个流行的开源集群计算框架。
- 基于YARN：即Hadoop 2，它是一个与Hadoop关联的集群计算和资源调度框架。

本章主要包括以下内容。

- 下载Spark二进制版本并搭建一个本地单机模式下的开发环境。各章的代码示例都在该环境下运行。
- 通过Spark的交互式终端来了解它的编程模型及其API。
- 分别用Scala、Java和Python语言来编写第一个Spark程序。
- 在Amazon的Elastic Cloud Compute（EC2）平台上架设一个Spark集群。相比本地模式，该集群可以应对数据量更大、计算更复杂的任务。

通过自定义脚本，Spark同样可以运行在Amazon的Elastic MapReduce服务上，但这不在本书讨论范围内。相关信息可参考http://aws.amazon.com/articles/4926593393724923；本书写作时，这篇文章是基于Spark 1.1.0写的。

如果读者曾构建过Spark环境并有Spark程序编写基础，可以跳过本章。

1.1　Spark的本地安装与配置

Spark能通过内置的单机集群调度器来在本地运行。此时，所有的Spark进程运行在同一个Java虚拟机中。这实际上构造了一个独立、多线程版本的Spark环境。本地模式很适合程序的原型设计、开发、调试及测试。同样，它也适应于在单机上进行多核并行计算的实际场景。

Spark的本地模式与集群模式完全兼容，本地编写和测试过的程序仅需增加少许设置便能在集群上运行。

本地构建Spark环境的第一步是下载其最新的版本包（本书写作时为1.2.0版）。各个版本的版本包及源代码的GitHub地址可从Spark项目的下载页面找到：http://spark.apache.org/downloads.html。

Spark的在线文档http://spark.apache.org/docs/latest/涵盖了进一步学习Spark所需的各种资料。强烈推荐读者浏览查阅。

为了访问**HDFS**（Hadoop Distributed File System，Hadoop分布式文件系统）以及标准或定制的Hadoop输入源，Spark的编译需要与Hadoop的版本对应。上述下载页面提供了针对Hadoop 1、CDH4（Cloudera的Hadoop发行版）、MapR的Hadoop发行版和Hadoop 2（YARN）的预编译二进制包。除非你想构建针对特定版本Hadoop的Spark，否则建议你通过如下链接从Apache镜像下载Hadoop 2.4预编译版本：http://www.apache.org/dyn/closer.cgi/spark/spark-1.2.0/spark-1.2.0-bin-hadoop2.4.tgz。

Spark的运行依赖Scala编程语言（本书写作时为2.10.4版）。好在预编译的二进制包中已包含Scala运行环境，我们不需要另外安装Scala便可运行Spark。但是，JRE（Java运行时环境）或JDK（Java开发套件）是要安装的（相应的安装指南可参见本书代码包中的软硬件列表）。

下载完上述版本包后，解压，并在终端进入解压时新建的主目录：

```
>tar xfvz spark-1.2.0-bin-hadoop2.4.tgz
>cd spark-1.2.0-bin-hadoop2.4
```

用户运行Spark的脚本在该目录的bin目录下。我们可以运行Spark附带的一个示例程序来测试是否一切正常：

```
>./bin/run-example org.apache.spark.examples.SparkPi
```

该命令将在本地单机模式下执行SparkPi这个示例。在该模式下，所有的Spark进程均运行于同一个JVM中，而并行处理则通过多线程来实现。默认情况下，该示例会启用与本地系统的CPU核心数目相同的线程。示例运行完，应可在输出的结尾看到类似如下的提示：

```
...
14/11/27 20:58:47 INFO SparkContext: Job finished: reduce at SparkPi.scala:35,
took 0.723269s
Pi is roughly 3.1465
...
```

要在本地模式下设置并行的级别，以`local[N]`的格式来指定一个`master`变量即可。上述参数中的`N`表示要使用的线程数目。比如只使用两个线程时，可输入如下命令：

```
>MASTER=local[2] ./bin/run-example org.apache.spark.examples.SparkPi
```

1.2 Spark 集群

Spark集群由两类程序构成：一个驱动程序和多个执行程序。本地模式时所有的处理都运行在同一个JVM内，而在集群模式时它们通常运行在不同的节点上。

举例来说，一个采用单机模式的Spark集群（即使用Spark内置的集群管理模块）通常包括：

- 一个运行Spark单机主进程和驱动程序的主节点；
- 各自运行一个执行程序进程的多个工作节点。

在本书中，我们将使用Spark的本地单机模式做概念讲解和举例说明，但所用的代码也可运行在Spark集群上。比如在一个Spark单机集群上运行上述示例，只需传入主节点的URL即可：

```
>MASTER=spark://IP:PORT ./bin/run-example org.apache.spark.examples.SparkPi
```

其中的`IP`和`PORT`分别是主节点IP地址和端口号。这是告诉Spark让示例程序运行在主节点所对应的集群上。

Spark集群管理和部署的完整方案不在本书的讨论范围内。但是，本章后面会对Amazon EC2集群的设置和使用做简要说明。

Spark集群部署的概要介绍可参见如下链接：

- http://spark.apache.org/docs/latest/cluster-overview.html
- http://spark.apache.org/docs/latest/submitting-applications.html

1.3　Spark编程模型

在对Spark的设计进行更全面的介绍前，我们先介绍SparkContext对象以及Spark shell。后面将通过它们来了解Spark编程模型的基础知识。

虽然这里会对Spark的使用进行简要介绍并提供示例，但要想了解更多，可参考下面这些资料。

- Spark快速入门：http://spark.apache.org/docs/latest/quick-start.html。
- 针对Scala、Java和Python的《Spark编程指南》：http://spark.apache.org/docs/latest/programming-guide.html。

1.3.1　SparkContext类与SparkConf类

任何Spark程序的编写都是从SparkContext（或用Java编写时的JavaSparkContext）开始的。SparkContext的初始化需要一个SparkConf对象，后者包含了Spark集群配置的各种参数（比如主节点的URL）。

初始化后，我们便可用SparkContext对象所包含的各种方法来创建和操作分布式数据集和共享变量。Spark shell（在Scala和Python下可以，但不支持Java）能自动完成上述初始化。若要用Scala代码来实现的话，可参照下面的代码：

```
val conf = new SparkConf()
.setAppName("Test Spark App")
.setMaster("local[4]")
val sc = new SparkContext(conf)
```

这段代码会创建一个4线程的SparkContext对象，并将其相应的任务命名为Test Spark APP。我们也可通过如下方式调用SparkContext的简单构造函数，以默认的参数值来创建相应的对象。其效果和上述的完全相同：

```
val sc = new SparkContext("local[4]", "Test Spark App")
```

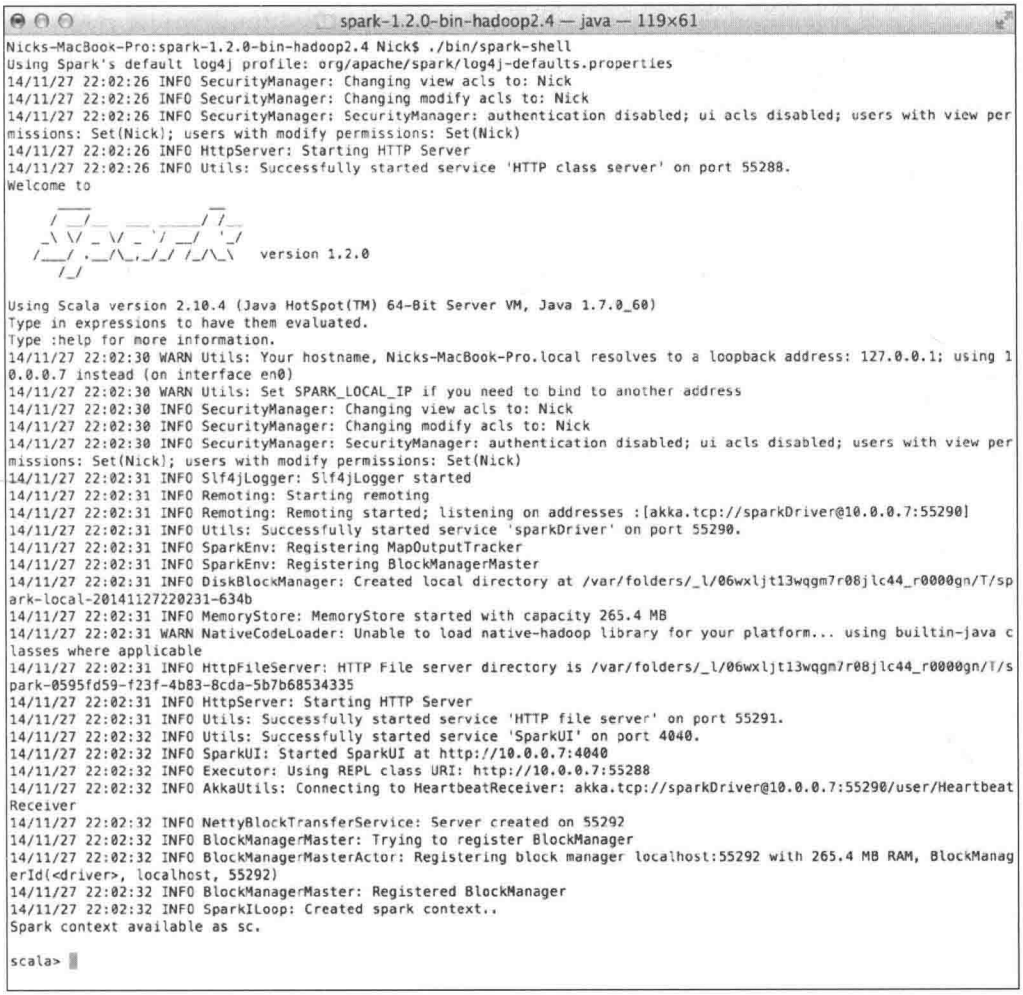
下载示例代码

你可从http://www.packtpub.com下载你账号购买过的Packt书籍所对应的示例代码。若书是从别处购买的，则可在https://www.packtpub.com/books/content/support注册，相应的代码会直接发送到你的电子邮箱。

1.3.2 Spark shell

Spark支持用Scala或Python REPL（Read-Eval-Print-Loop，即交互式shell）来进行交互式的程序编写。由于输入的代码会被立即计算，shell能在输入代码时给出实时反馈。在Scala shell里，命令执行结果的值与类型在代码执行完后也会显示出来。

要想通过Scala来使用Spark shell，只需从Spark的主目录执行./bin/spark-shell。它会启动Scala shell并初始化一个SparkContext对象。我们可以通过sc这个Scala值来调用这个对象。该命令的终端输出应该如下图所示：



要想在Python shell中使用Spark，直接运行./bin/pyspark命令即可。与Scala shell类似，Python下的SparkContext对象可以通过Python变量sc来调用。上述命令的终端输出应该如下图所示：