Machine Learning with Spark Second Edition

Spark机器学习（第2版）

[印] 拉结帝普·杜瓦 [印] 曼普利特·辛格·古特拉 [南非] 尼克·彭特里思 著
蔡立宇 黄章帅 周济民 译

- Spark项目管理委员会成员作品
- 注重技术实践，通过大量实例演示如何创建有用的机器学习系统

中国工信出版集团
人民邮电出版社
POSTS & TELECOM PRESS

计丛书

Machine Learning with Spark Second Edition

Spark机器学习（第2版）

[印] 拉结帝普·杜瓦 [印] 曼普利特·辛格·古特拉 [南非] 尼克·彭特里思 著
蔡立宇 黄章帅 周济民 译

人民邮电出版社
北 京

图书在版编目（CIP）数据

Spark机器学习 : 第2版 / (印) 拉结帝普·杜瓦, (印) 曼普利特·辛格·古特拉, (南非) 尼克·彭特里思著 ; 蔡立宇, 黄章帅, 周济民译. -- 北京 : 人民邮电出版社, 2018.11
(图灵程序设计丛书)
ISBN 978-7-115-49783-3

Ⅰ. ①S… Ⅱ. ①拉… ②曼… ③尼… ④蔡… ⑤黄… ⑥周… Ⅲ. ①数据处理软件 Ⅳ. ①TP274

中国版本图书馆CIP数据核字(2018)第243357号

内 容 提 要

本书结合案例研究讲解 Spark 在机器学习中的应用，并介绍如何从各种公开渠道获取用于机器学习系统的数据。内容涵盖推荐系统、回归、聚类、降维等经典机器学习算法及其实际应用。第 2 版新增了有关机器学习数学基础以及 Spark ML Pipeline API 的章节，内容更加系统、全面、与时俱进。

本书适合数据分析从业人员以及高校数据挖掘相关专业师生阅读参考。

◆ 著 [印] 拉结帝普·杜瓦 [印] 曼普利特·辛格·古特拉 [南非] 尼克·彭特里思
译 蔡立宇 黄章帅 周济民
责任编辑 温 雪
责任印制 周昇亮

◆ 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
三河市君旺印务有限公司印刷

◆ 开本：800×1000 1/16
印张：24.25
字数：573千字 2018年11月第1版
印数：1-3 500册 2018年11月河北第1次印刷
著作权合同登记号 图字：01-2017-8608号

定价：99.00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：京东工商广登字 20170147 号

站在巨人的肩上

Standing on Shoulders of Giants

iTuring.cn

站在巨人的肩上

Standing on Shoulders of Giants

iTuring.cn

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、Twitter 和 Salesforce 这样的大公司才会涉及，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数据的需求，人工驱动的系统就难以应对。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持续增长的大规模数据的支持，Google、Yahoo!、Amazon 和 Facebook 等公司推出了大量开源技术。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，它极大地简化了海量数据的存储（通过 HDFS，即 Hadoop distributed file system）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，从而弥补了 Hadoop 框架中的一些主要缺陷。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、Python 和 R 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 现在还自带了一个支持分布式机器学习和包含若干数据挖掘模型的工具包（1.6 版中为 Spark MLlib，2.0 版则对应 Spark ML）。该工具包正在重点开发中，但已包括多个针对常见机器学习任务的高质量、可扩展的算法。本书将会涉及部分此类任务。

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。多数情况下，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，但总体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；然后会介绍 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“机器学习的数学基础”会提供机器学习所需的基础数学知识。要理解算法，从而获得更好的建模效果，理解数学及其技巧十分重要。

第 3 章“机器学习系统设计”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 4 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 5 章“Spark 构建推荐引擎”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清单。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 6 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 7 章“Spark 构建回归模型”扩展了第 6 章中的分类模型以创建一个回归模型，并详细介绍了回归模型的评估指标。

第 8 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 9 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 10 章“Spark 高级文本处理技术”介绍了处理大规模文本数据的方法。这包括从文本中提取特征以及处理文本数据中常见的高维特征的方法。

第 11 章“Spark Streaming 实时机器学习”对 Spark Streaming 进行了综述，并介绍它如何在流数据上的机器学习中实现对在线和增量学习方法的支持。

第 12 章“Spark ML Pipeline API”在 Data Frames 的基础上提供了一套统一的接口（API），帮助用户创建和调试机器学习流程。

预备知识

本书假设读者已有基本的 Scala、Java、Python 或 R 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。

目标读者

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 或 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-2.1.0-bin-hadoop2.7.tgz
> cd spark-2.1.0-bin-hadoop2.7
```

新术语和**重点词汇**以黑体表示。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 https://www.packtpub.com/books/info/packt/authors。

客户支持

现在，你是一位令人自豪的 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 https://www.packtpub.com 下载所有已购买 Packt 图书的示例代码文件。[①]如果你从其他地方购买本书，可以访问 https://www.packtpub.com/books/content/support 并注册，我们将通过电子邮件把文件发送给你。

你可通过如下步骤下载本书示例代码：

(1) 在上述网站上，使用你自己的邮件地址和密码登录或注册；
(2) 将鼠标移动到顶部的 SUPPORT 标签页；
(3) 点击 Code Downloads & Errata；
(4) 在 Search 栏中输入本书名；
(5) 在搜索结果中选择要下载代码的书；
(6) 从下拉菜单中选择从何处购买的本书；
(7) 点击 Code Download 下载包含代码和部分数据的代码包。

代码包下载后，请使用如下软件的最新版本来解压缩或提取所含资料：

- Windows 上建议使用 WinRAR/7-Zip；
- Mac 上建议使用 Zipeg/iZip/UnRarX；
- Linux 上建议使用 7-Zip/PeaZip。

代码包在 GitHub 上也可获取，地址为 https://github.com/PacktPublishing/Machine-Learning-with-Spark-Second-Edition。https://github.com/PacktPublishing 上也列出了我们所出版的各类图书和视频的代码包，欢迎查看。

① 读者也可访问本书图灵社区页面（http://www.ituring.com.cn/book/2041）下载示例代码及彩图，并提交本书中文版勘误。——编者注

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 https://www.packtpub.com/books/info/packt/errata-submission-form-0 提交，选择你的书，点击勘误表提交表单的链接（Errata Submission Form），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 https://www.packtpub.com/books/content/support 选择书名，在图书页面的 Errata 区域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。

电子书

扫描如下二维码，即可购买本书电子版。

目　录

第 1 章 Spark 的环境搭建与运行

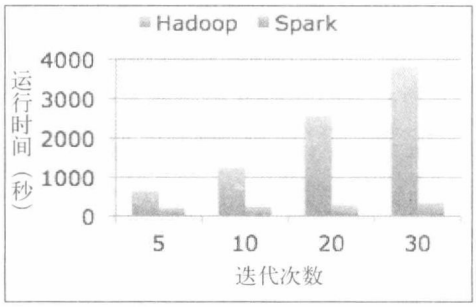
Apache Spark 是一个分布式计算框架，旨在简化运行于计算机集群或虚拟机上的并行程序的编写。该框架对资源调度，任务的提交、执行和跟踪，节点间的通信以及数据并行处理的内在底层操作都进行了抽象。它提供了一个更高级别的 API 来处理分布式数据。从这方面说，它与 Apache Hadoop 等分布式处理框架类似。但在底层架构上，Spark 与它们有所不同。

Spark 起源于加州大学伯克利分校 AMP 实验室的一个研究项目。该高校当时关注分布式机器学习算法的应用情况。因此，Spark 从一开始便是为应对迭代式应用的高性能需求而设计的。在这类应用中，相同的数据会被多次访问。该设计主要通过在内存中缓存数据集以及启动并行计算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 项目的更多信息，请参见：

- http://spark.apache.org/community.html
- http://spark.apache.org/community.html#history

从性能上说，Spark 在不同工作负载下的运行速度明显高于 Hadoop，如下图所示。



来源：https://amplab.cs.berkeley.edu/wp-content/uploads/2011/11/spark-lr.png

Spark 支持 4 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（JVM，Java virtual machine）进程中。
- 集群单机模式：使用 Spark 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 Hadoop YARN：YARN 常被称作 NextGen MapReduce。

本章主要包括以下内容。

- 下载 Spark 二进制版本，并搭建一个在本地单机模式下运行的开发环境。本书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 API。
- 分别用 Scala、Java、R 和 Python 语言来编写第一个 Spark 程序。
- 在 Amazon 的 EC2（Elastic Cloud Compute）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。
- 借助 Amazon Elastic Map Reduce 服务来构建一个 Spark 集群。

如果读者曾构建过 Spark 环境并熟悉有关 Spark 程序编写的基础知识，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在本地模式下运行。此时，所有的 Spark 进程都高效地运行在同一个 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。本地模式常用于原型设计、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，在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。各版本的版本包及源代码的 GitHub 地址可在 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 Hadoop 分布式文件系统（HDFS）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 Cloudera 的 Hadoop 发行版（CHD）、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 2.7 预编译版本：http://d3kbcqa49mib13.cloudfront.net/spark-2.0.2-bin-hadoop2.7.tgz。

Spark 的运行依赖 Scala 编程语言（写作本书时为 2.10.*x* 或 2.11.*x* 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，你需要先安装好 Java 运行时环境（JRE）或 Java 开发工具包（JDK）。

相应的安装指南可参见本书代码包中的软硬件列表。推荐使用 R 3.1 或以上版本。

下载完上述版本包后，在终端输入如下指令解压软件包并进入解压出的文件夹：

```
$ tar xfvz spark-2.0.0-bin-hadoop2.7.tgz
$ cd spark-2.0.0-bin-hadoop2.7
```

用户启动 Spark 所用的脚本在该目录的 bin 文件夹下。可通过如下命令运行 Spark 附带的一个示例程序来测试是否一切正常：

```
$ ./bin/run-example SparkPi 100
```

该命令将在本地单机模式下运行 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，该示例会启用的线程数与本地系统的 CPU 核心数目相同。示例运行完后，应可在输出的结尾看到如下的提示：

```
...
16/11/24 14:41:58 INFO Executor: Finished task 99.0 in stage 0.0
  (TID 99). 872 bytes result sent to driver
16/11/24 14:41:58 INFO TaskSetManager: Finished task 99.0 in stage
  0.0 (TID 99) in 59 ms on localhost (100/100)
16/11/24 14:41:58 INFO DAGScheduler: ResultStage 0 (reduce at
  SparkPi.scala:38) finished in 1.988 s
16/11/24 14:41:58 INFO TaskSchedulerImpl: Removed TaskSet 0.0,
  whose tasks have all completed, from pool
16/11/24 14:41:58 INFO DAGScheduler: Job 0 finished: reduce at
  SparkPi.scala:38, took 2.235920 s
Pi is roughly 3.1409527140952713
```

上述命令调用了 `org.apache.spark.examples.SparkPi` 类。

该类以 `local[N]`格式来接受输入参数，其中 `N` 表示要启用的线程数目。比如只使用两个线程时，便可使用如下命令：

```
$ ./bin/spark-submit --class org.apache.spark.examples.SparkPi
  --master local[2] ./examples/jars/spark-examples_2.11-2.0.0.jar 100
```

按惯例，将命令中的 `local[2]`改为 `local[*]`则会使用本机所有可用的核心。

1.2 Spark 集群

Spark 集群由两类进程构成：一个驱动程序和多个执行程序。在本地模式下，所有的进程都运行在同一个 JVM 内，而在集群模式下时，它们通常运行在不同的节点上。