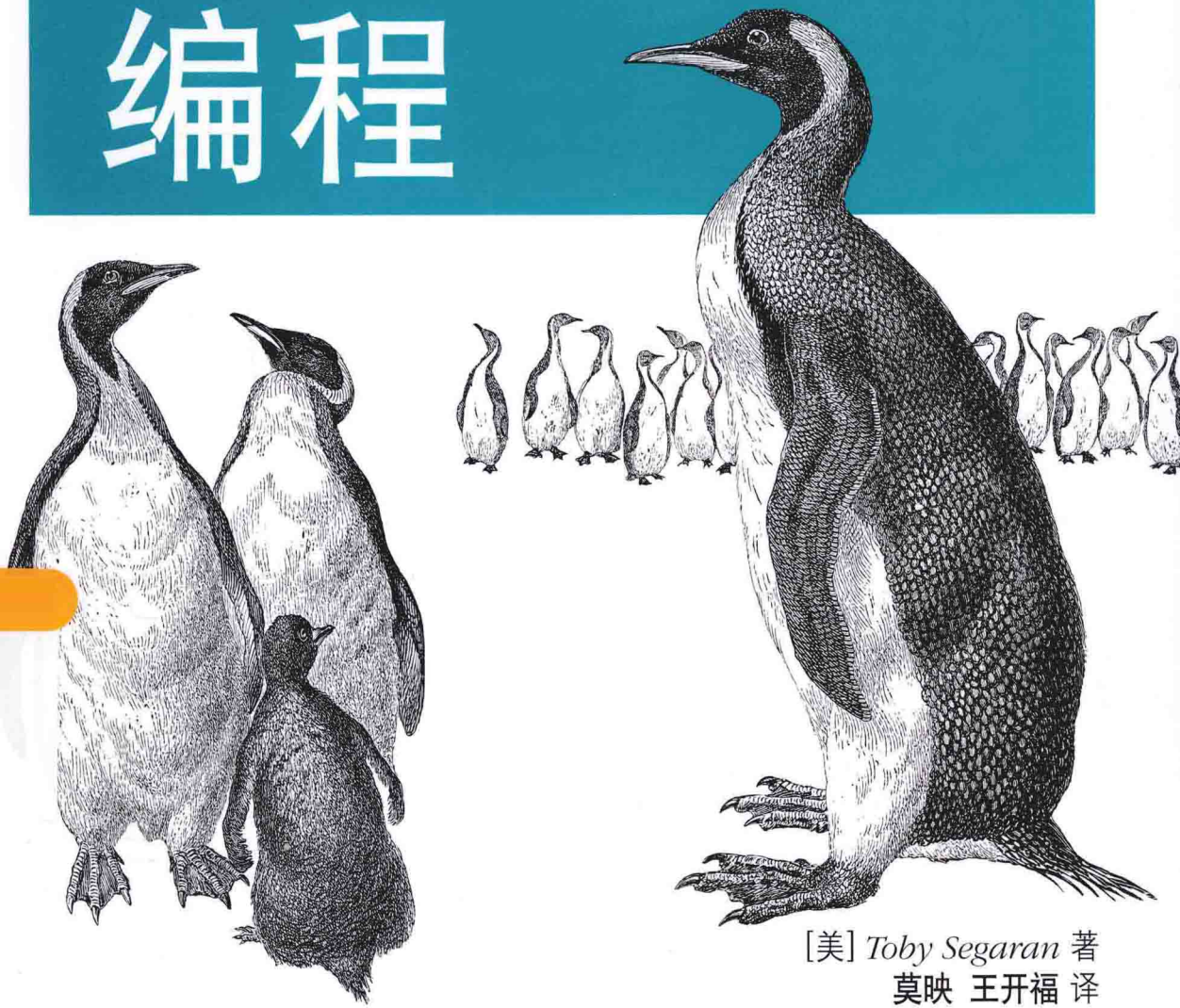
Programming Collective Intelligence

构建智能Web 2.0应用

集体智慧编程



[美] *Toby Segaran* 著

莫映 王开福 译

O'REILLY®

中国工信出版集团

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

集体智慧编程

Programming Collective Intelligence

[美] Toby Segaran 著

莫 映 王开福 译

電子工業出版社
Publishing House of Electronics Industry
北京•BEIJING

内 容 简 介

本书以机器学习与计算统计为主题背景，专门讲述如何挖掘和分析 Web 上的数据和资源，如何分析用户体验、市场营销、个人品味等诸多信息，并得出有用的结论，通过复杂的算法来从 Web 网站获取、收集并分析用户的数据和反馈信息，以便创造新的用户价值和商业价值。全书内容翔实，包括协作过滤技术（实现关联产品推荐功能）、集群数据分析（在大规模数据集中发掘相似的数据子集）、搜索引擎核心技术（爬虫、索引、查询引擎、PageRank 算法等）、搜索海量信息并进行分析统计得出结论的优化算法、贝叶斯过滤技术（垃圾邮件过滤、文本过滤）、用决策树技术实现预测和决策建模功能、社交网络的信息匹配技术、机器学习和人工智能应用等。

本书是 Web 开发者、架构师、应用工程师等的绝佳选择。

版权贸易合同登记号图字：01-2007-5378

图书在版编目（CIP）数据

集体智慧编程/（美）西格兰（Segaran,T.）著；莫映，王开福译. —北京：电子工业出版社，2015.3

书名原文: Programming collective intelligence

ISBN 978-7-121-25443-7

Ⅰ. ①集… Ⅱ. ①西… ②莫… ③王… Ⅲ. ①互联网络－程序设计 Ⅳ. ①TP393.4

中国版本图书馆 CIP 数据核字(2015)第 012312 号

策划编辑：张春雨
责任编辑：许　艳
封面设计：Karen Montgomery　张　健
印　　刷：北京中新伟业印刷有限公司
装　　订：三河市皇庄路通装订厂
出版发行：电子工业出版社
　　北京市海淀区万寿路 173 信箱　邮编 100036
开　　本：787×980　1/16　印张：22.25　字数：554 千字
版　　次：2015 年 3 月第 1 版
印　　次：2015 年 3 月第 1 次印刷
定　　价：79.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。服务热线：（010）88258888。

O'Reilly Media, Inc.介绍

O'Reilly Media 通过图书、杂志、在线服务、调查研究和会议等方式传播创新知识。自1978年开始，O'Reilly一直都是前沿发展的见证者和推动者。超级极客们正在开创着未来，而我们关注真正重要的技术趋势——通过放大那些“细微的信号”来刺激社会对新科技的应用。作为技术社区中活跃的参与者，O'Reilly的发展充满了对创新的倡导、创造和发扬光大。

O'Reilly为软件开发人员带来革命性的“动物书”；创建第一个商业网站（GNN）；组织了影响深远的开放源代码峰会，以至于开源软件运动以此命名；创立了Make杂志，从而成为DIY革命的主要先锋；公司一如既往地通过多种形式缔结信息与人的纽带。O'Reilly的会议和峰会集聚了众多超级极客和高瞻远瞩的商业领袖，共同描绘出开创新产业的革命性思想。作为技术人士获取信息的选择，O'Reilly现在还将先锋专家的知识传递给普通的计算机用户。无论是通过书籍出版、在线服务或者面授课程，每一项O'Reilly的产品都反映了公司不可动摇的理念——信息是激发创新的力量。

业界评论

“O'Reilly Radar博客有口皆碑。”

——Wired

“O'Reilly凭借一系列（真希望当初我也想到了）非凡想法建立了数百万美元的业务。”

——Business 2.0

“O'Reilly Conference是聚集关键思想领袖的绝对典范。”

——CRN

“一本O'Reilly的书就代表一个有用、有前途、需要学习的主题。”

——Irish Times

“Tim是位特立独行的商人，他不光放眼于最长远、最广阔的视野并且切实地按照Yogi Berra的建议去做了：‘如果你在路上遇到岔路口，走小路（岔路）。’回顾过去Tim似乎每一次都选择了小路，而且有几次都是一闪即逝的机会，尽管大路也不错。”

——Linux Journal

对本书的赞誉

Praise for *Programming Collective Intelligence*

“每年我都要审阅几本图书，自然而然地，在工作当中我阅读了大量的书籍。不得不承认，阅读本书让我获得了以前从未有过的、相当愉悦的阅读体验。太棒了！对于初学这些算法的开发者而言，我想不出比这本书更好的选择了，而对于像我这样学过 AI 的老朽而言，我也想不出还有什么更好的办法能够让自己重温这些知识的细节。”

——Dan Russell，Google 公司资深技术经理

“Toby 的这本书非常成功地将机器学习算法这一复杂的议题拆分成了一个个既实用又易懂的例子，我们可以直接利用这些例子来分析当前网络上的社会化交互作用。我要是早两年读过这本书，就会省去许多宝贵的时间，也不至于走那么多的弯路了。”

——Tim Wolters，Collective Intellect 公司 CTO

“本书获得了巨大的成功，它为大量相关数据的处理提供了非常丰富的计算方法。更重要的是，它将这些技术应用到了互联网上，而不是在一个个彼此孤立的数据孤岛中寻求价值。如果你是在为互联网开发应用，那么本书将是你的不二之选。”

——Paul Tyma，Google 公司高级软件工程师

译者序

还记得 20 世纪 90 年代，当大学校园里的学子们还在为能够通过对等网在不同计算机间相互发送消息而兴奋不已的时候，互联网就已经悄然兴起了。很快，人们就从 C/S 时代跨入了 B/S 时代。我们不必再担心每次都要安装复杂的客户端程序，只要有浏览器，就会有绚丽多彩的舞台。然而随着时间的推移，人们又开始有所回归，大家不时地抱怨：为什么不能让浏览器像客户端应用那样具有丰富的表现？为什么每次打开链接都要傻傻地等着空白页面消失？直到有一天，Tim O'Reilly 向世人宣告了一个新的概念——Web 2.0。于是，忽如一夜春风来，大大小小的 Web 2.0 应用如雨后春笋般不断涌现，互联网又迈向了一个新的时代。

Web 2.0 使互联网变得异彩纷呈：来自不同地域的人们可以随时修改别人写的文字，这就是维基；你有任何想法或观点都可以尽情地表达并欢迎别人评论，这就是博客；甚至连网页上出现的广告也都是与我们当前所关注的内容密切相关的，这就是 Google AdSense……所有这一切，都带给我们不同于以往的全新感受。但是，这些应用究竟是怎样实现的？隐藏在它们背后的原理到底是什么？怎样让我们的 Web 2.0 程序变得更加聪明，更加贴心呢？译者相信，本书必定能够为大家逐一解开萦绕在心中的这些谜团。

本书以 Web 2.0 的核心价值观——集体智慧作为出发点，探讨了各种能够让 Web 2.0 程序变得更为智能的算法及其应用。这些算法大多数来自机器学习和计算统计领域，其中的一些算法非常普及，而另一些则属于目前相当前沿的课题。它们包括了过滤器、聚类算法、支持向量机、遗传编程、优化技术，以及非常著名的 PageRank 算法，等等。将如此众多的优秀算法有效应用于互联网领域，并构造出具有智能特征的 Web 2.0 应用，应该是本书的一大亮点。同时，这也使本书有别于以往我们所见过的任何一本纯粹介绍 Web 2.0 技术与概念的书籍。不仅如此，本书还提供了大量可供运行的示例代码，这些代码具有很好的复用性，只要稍加修改就可以用于实际的应用系统之中。书中代码还大量使用了许多时下流行的开放 API，这些 API 来自于 Yahoo!、eBay、FaceBook 等众多热门的 Web 2.0 网站，这使得本书在保有实用价值的同时又不失时效性。

本书的英文版虽只有寥寥 300 多页，比起任何一本大部头的技术书籍都是不足道的，但作为一本为数不多的深入讲解蕴藏于智能 Web 2.0 应用背后的算法原理的书籍，其深度和内涵却远远超出了篇幅的局限。为了尽量将原书的思想内涵以中文形式尽数表达出来，作为译者的我们在本书翻译期间着实不敢懈怠。在将书稿提交给出版社编辑之前，我们对每一章的译文都进行了不少于两遍的仔细校对。作为补充，中文版还随附了翻译期间译者所用的中英文术语对照表，希望本书中文版能够得到诸位读者的认可。

这本译作的完成是团队协作努力的结果，这包括了参与翻译、审校，以及关注和支持本书翻译的所有人。感谢周筠老师对我们的信任，感谢本书的前后两位编辑王凡毓与王晓菲，尤其是晓菲，她为本书的后期审校与编辑加工付出了辛劳，我们的合作非常愉快。此外，还要感谢李唯一，她为本书的前期翻译提供了诸多帮助。

由于译者水平所限，译文难免有疏误之处，欢迎读者批评指正。

为了便于读者阅读理解，特在此附上本书翻译过程中整理提取的中英文术语对照表，参见表 0-1，表中所包含的多为专业领域的技术术语。其中部分术语在不同的文献中往往有不同的译法。本书为了统一，选择了比较常见的译法，如 clustering 可译作“聚类”或“聚集”，此处我们选择了“聚类”。类似的还有 k-nearest neighbors、cross-product、dot-product，等等。

另一部分术语，虽有固定译法，但我们结合上下文，采用了更为贴切的翻译。如 computationally intensive 常被译为“计算密集的”，而在此处，我们采用“计算量很大的”。类似的还有 data-intensive、solution、crawl，等等。

此外还有一部分术语，在当下的中文文献中并没有明确的公认译法，因而我们在书中给出了参考翻译，以供大家商榷。如 collective intelligence 被译为“集体智慧”，list comprehension 被译为“列表推导式”，等等。

表 0-1：中英文术语对照表

英　文	中　文	英　文	中　文
clustering	聚类	collective intelligence	集体智慧
computationally intensive	计算量很大的	crawl	（网页）检索
cross-product	叉乘	data-intensive	数据量很大的
dendrogram	树状图	dot-product	点积
groups	群组	inbound link，incoming link	外部回指链接
kernel methods，kernel tricks	核方法，核技法	K-Means	K-均值
k-nearest neighbors	k-最近邻	list comprehension	列表推导式
multidimensional scaling	多维缩放	observations	观测数据，观测值
pattern	模式	similarity	相似度，相似性
solution	（题）解	vertical search engine	垂直搜索引擎

莫　映　王开福

推荐序
Foreword

2006 年《时代周刊》将“You”评选为年度风云人物，证实了 Web 2.0 时代是用户创建内容的时代这一思想，也证实了维基百科、YouTube 和 Myspace 是 Web 2.0 变革的核心。而真相远比这复杂得多。我们看到的用户贡献给 Web 2.0 的内容只是冰山一角，还有 80%的重要内容深藏在水面之下，只是这些内容的贡献方式不是那么明显。

从很多方面来说，当 Google 公司发明 PageRank 技术后，Web 2.0 的革新才算开始。自此人们开始意识到万维网上的每一个链接都有其隐含的意义：每一个链接都是对这个网站重要性的一张投票。读懂了这些投票，以及站点的相对重要性，就能获得比仅研究网页本身更有用的搜索结果。正是这种突破性技术把 Google 推上了 21 世纪最重要技术公司的宝座。PageRank 现在是 Google 在决定采用哪些搜索结果时所使用的上百个隐性因子之一。

没有人会把 Google 定位为一家“用户创造内容”的公司，但是“用户创造内容”的的确确是 Web 2.0 的核心。这也是为什么我选择“是否利用了集体智慧”作为这场革新的判断标准。链接是用户创造的内容，但是 PageRank 是从内容中提取智慧的技术。Flickr 的“interestingness”算法，亚马逊的“购买此产品的用户还买了……”功能，Last.fm 的“相似艺术家的音乐”，eBay 的信用体系，以及 Google 的 Adsense，都使用了这样的技术。

我认为 Web 2.0 是“一套能利用人际网影响的体系设计，用的人越多，其本身就会变得越完善”。因此，第一步是让用户参与进来。然后，从这些用户身上学习，依据他们的行为和关注点来改良你的网站。

在《集体智慧编程》这本书中，Toby 介绍了从数据，包括用户数据，中提取有效信息的算法和技术。这些是 Web 2.0 程序员们需要掌握的知识。如果你想创建成功的网站，仅知道如何创建有数据库支持的网站是不够的，还需要了解如何从用户添加的数据中挖掘出有用的信息，这些数据包括他们直接添加的数据以及他们在你网站上的行为所产生的数据。

尽管自从 Web 2.0 这个词出现以来，很多人都写过关于它的书，但是本书是第一本教授 Web 2.0 应用编程的实用手册。

——Tim O'Reilly

前言
Preface

无论是有意还是无意，越来越多投身于互联网的人们已经制造出了相当多的数据，这给了我们无数潜在的机会来洞悉用户体验、商业营销、个人偏好和通常所谓的人类行为（human behavior）。本书向大家介绍了一个新兴的领域，称为**集体智慧**（collective intelligence）。这一领域涵盖了诸多方法，借助这些方法我们可以从众多 Web 站点处（这些站点的名字或许你曾经有所耳闻）提取到值得关注的重要数据；借助这些方法我们还可以从使用自己应用程序的用户那里搜集信息，并对我们所掌握的数据进行分析和理解。

本书的目的是要带领你超越以数据库为后端的简单应用系统，并告诉你如何利用自己和他人每天搜集到的信息来编写更为智能的程序。

先决条件
Prerequisites

本书的代码示例是用 Python 语言编写的，因此熟悉 Python 编程将会有助于你对算法的理解，不过笔者给出了所有算法的解释说明，所以其他语言的程序员也能看懂。对于已经了解了像 Ruby 或 Perl 这样高级语言的程序员，Python 代码应该是非常容易理解的。本书的定位不是要作为一本学习编程的指导书，因此尤为重要的一点在于，为了熟悉基本概念，最好已编写过足够多的代码。如果懂得递归和一点点函数式编程（functional programming）的基本概念，那么就会发觉书中的内容是很容易理解的。

本书并不假设你已经具备了任何有关数据分析、机器学习或统计学方面的知识。笔者在尝试以尽可能浅显易懂的方式来解释数学概念，不过具备一点三角学和统计学的基本知识将会对你理解算法有所助益。

示例风格
Style of Examples

本书每一章节的代码示例都是以一种教程式的风格编写而成的，它鼓励你以循序渐进的方式来构建应用程序，并对算法的工作原理有一个深入的理解和认识。大多数情况下，写完一个新的函数或方法之后，我们会在一个交互的会话环境里使用它，以此来理解算法的工作原理。通常算法是有简单的变体的，我们可以用多种方式对其进行扩展。通过示例讲解并以交互的方式对其进行测试，我们对算法将会有更为深入的理解，从而可以对其进行改造，以适应自己的应用程序。

为何选择 Python
Why Python?

尽管书中的算法是伴随着对相关公式的解释，以文字形式加以描述的，但是假如有针对算法和示例问题的实际代码，那将会是更有助益的（而且有可能更易于理解）。本书中的所有示例代码都是用 Python，一种优秀的高级语言编写而成的。之所以选择 Python 是因为它有如下特性。

简练

使用像 Python 这样的动态类型语言编写的代码往往比用其他主流语言编写而成的代码更加简短。这意味着，在完成示例的过程中会有更少的录入工作，而且这也意味着我们将更容易记住算法并真正领会算法的原理。

易于阅读

Python 不时被人们指为“可执行的伪代码”。虽然很明显这是一种夸大之词，但是它表明，大多数有经验的程序员可以读懂 Python 代码并领会代码所要表达的意图。Python 中一些不是很显见的语言要素将会在后面的“Python 技巧”一节中加以解释。

易于扩展

Python 随附了许多标准库，这些库涉及数学函数、XML（扩展标记语言）解析，以及网页下载。本书中用到的非标准库，如 RSS（Really Simple Syndication）解析器和 SQLite 接口，则是免费的，很容易下载、安装和使用。

交互性

在学习示例的过程中，可以尝试执行我们编好的函数，而无须为此专门编写额外的程序，这一点是非常有价值的。Python 可以直接从命令行运行程序，它还有交互提示，允许我们输入函数调用、创建对象，并以交互的形式来对包进行测试。

多范式

Python 支持面向对象、过程式和函数式编程风格。机器学习算法千差万别，最为清晰

的做法是针对不同算法采用不同的范式。有时将函数作为参数传入很有用处，而有时我们则需要在对象中捕获状态。对于这两种方式，Python 均予以支持。

多平台和免费

Python 有一个针对所有主流平台的单一参考实现，并且它对所有平台都是免费的。本书中所列代码可以运行于 Windows、Linux 和 Macintosh 环境。

Python 技巧
Python Tips

对于有兴趣学习 Python 编程的初学者而言，笔者推荐大家阅读由 Mark Lutz 与 David Ascher 合著的《Learning Python》（O'Reilly），该书对 Python 做了全面论述。为了更为直观地表达算法或基础性概念，笔者在整本书中使用了一些 Python 特有的语法，但是其他语言的程序员应该会发现，Python 的代码相对而言还是较为容易掌握的。下面是为非 Python 程序员提供的一份快速概览。

列表和字典的构造函数

Python 有一组不错的基本类型，其中有两种类型在本书中被大量使用，它们分别是**列表**和**字典**。列表是由一组任意类型的值构成的有序列表，它由方括号构造而成：

```
number_list=[1,2,3,4]
string_list=['a', 'b', 'c', 'd']
mixed_list=['a', 3, 'c', 8]
```

字典是由一组名值对构成的无序集合，类似于其他语言中的 hash map。它由大括号构造而成：

```
ages={'John':24,'Sarah':28,'Mike':31}
```

可以通过序列名后跟方括号的形式来访问列表和字典中的元素：

```
string_list[2]   # returns 'c'
ages['Sarah']    # returns 28
```

有意义的空白字符

与大多数语言有所不同，Python 实际上是利用代码的缩进来定义代码块的。请看下列代码片段：

```
if x==1:
  print 'x is 1'
  print 'Still in if block'
print 'outside if block'
```

因为代码是被缩进的，所以语法解释器知道前两个打印语句会在 x 为 1 的时候被执行。缩进可以是任意数量的空格，只要它是常量即可。本书使用的缩进是两个空格。在输入代码的时候，我们需要注意正确复制缩进。

列表推导式

列表推导式（list comprehension）是一种方便简洁的语法形式，我们可以利用它将一个列表经过滤后转换成另一个列表，也可以利用它将函数应用于列表中的元素。列表推导式以如下形式书写：

```
[表达式 for 变量 in 列表]
```

或者：

```
[表达式 for 变量 in 列表 if 条件]
```

例如，下列代码：

```
l1=[1,2,3,4,5,6,7,8,9]
print [v*10 for v in l1 if v>4]
```

将打印输出如下列表：

```
[50,60,70,80,90]
```

本书中频繁地使用了列表推导式，因为要将一个函数应用于整个列表，或是删除不需要的列表项时，这种表达方法非常简练。列表推导式的另一种常见用法是与 `dict` 构造函数结合在一起使用：

```
l1=[1,2,3,4,5,6,7,8,9]
timesten=dict([(v,v*10) for v in l1])
```

上述代码将会建立一个字典，以原先的列表作为键，以每个列表项乘以 10 作为值：

```
{1:10,2:20,3:30,4:40,5:50,6:60,7:70,8:80,9:90}
```

开放的 API
Open APIs

用于将集体智慧合成起来的算法需要来自许多用户的数据。除机器学习的算法外，本书还论及了许多开放的 Web API（应用编程接口）。这些 API 允许我们通过特殊的协议对来自相应 Web 站点的数据进行访问，我们可以编写程序将数据下载下来并加以处理。这些数据通常是由站点的使用者来提供的，我们可以从中挖掘出新的内涵来。有的时候，我们可以用现成的 Python 库来访问这些 API；而有时，如果没有现成的库，那么最为直接的做法莫过于创建自己的接口来访问数据，为此我们需要利用 Python 提供的内建库将数据下载下来，并对 XML 加以解析。

此处列出了一系列提供开放 API 的 Web 站点，我们将在本书中陆续接触到这些站点。

del.icio.us

一个社会型书签应用系统（social bookmarking application），其开放的 API 允许你根据标签（tag）或特定的用户来下载链接。

Kayak

一个提供 API 的旅游网站，你可以利用 API 在自己的程序中集成针对航班和旅馆的搜索。

eBay

一个提供 API 的在线交易站点，允许你查询当前正在出售的货品。

Hot or Not

一个评分与交友的网站，提供 API 对人员进行搜索，并获取其评分及个人资料。

Akismet

一种用于对协作型垃圾信息进行过滤的 API。

通过对来自单一源的数据进行处理，对来自多个源的数据进行组合，甚至通过将外部信息与自有系统的用户输入信息加以组合，我们可以构造出大量的潜在应用。对人们在不同网站以各种不同方式产生的数据加以充分利用的能力，便是构建集体智慧的一个基本要素。如果你想寻找更多的提供开放 API 的 Web 站点，不妨从访问 ProgrammableWeb 开始（*http://www.programmableweb.com*）。

各章概览
Overview of the Chapters

本书的每个算法都来源于某一现实的问题，希望这些问题能够很容易地被广大读者所理解。笔者将尝试尽量避开那些要求大量领域知识的问题，而将焦点集中在那些虽不失复杂性，但对大多数涉足者而言却又是简单易懂的问题上。

第 1 章，集体智慧导言

本章解释了蕴藏于机器学习背后的概念，并解释了如何将其应用于诸多不同的领域，以及如何利用它对从不同人群处搜集的数据进行分析，并从中得出新的结论。

第 2 章，提供推荐

本章介绍**协作型过滤**（collaborative filtering）技术，这项技术被许多在线零售商用来向顾客推荐商品或媒体。本章中有一节介绍了如何向一个社会型书签服务网站的用户提供推荐链接，还介绍了如何根据 MovieLens 所提供的数据集构筑一个影片推荐系统。

第 3 章，发现群组

本章基于第 2 章中给出的某些观点，介绍了两种不同的**聚类方法**，利用这些方法，我们可以在一个大数据集中自动找出具有相似特征的群组。本章还演示了如何利用聚类算法从一组颇受欢迎的博客之中寻找群组，以及利用聚类算法根据某个社会型网站的用户意愿去寻找群组。

第 4 章，搜索与排名

本章描述了构成一个搜索引擎的各个不同组成部分，它们包括：爬虫程序（crawler）、索引程序（indexer），以及查询引擎（query engine）。本章介绍了以来自外部网站的链接信息为依据给网页打分的 *PageRank* 算法，还向你展示了如何构建**神经网络**，借此获知与不同结果相关联的关键词。

第 5 章，优化

本章介绍了**优化算法**，设计这些算法的目的是，对问题的数百万个可能的题解进行搜索，并从中选出最优解。书中利用示例演示了这些算法的各种不同用法，包括：为一群去往相同地点的旅客寻找最佳航班，寻求为学生安排宿舍的最佳方案，以及给出交叉线数量最小的网络布局。

第 6 章，文档过滤

本章向读者演示了**贝叶斯过滤**，这一方法被广泛应用于许多免费的和商业的垃圾信息过滤系统中，用于根据单词类型及出现在文档中的其他特征对文档进行自动分类。我们将其应用于一组 RSS 搜索结果，以此来说明对内容项的自动分类过程。

第 7 章，决策树建模

本章介绍了**决策树**，我们不仅将它作为一种预测方法，而且还用它来为决策过程进行建模。本章中出现的第一棵决策树是根据假想的服务器日志数据构建而成的，我们利用它来预测用户是否有可能成为付费订户（premium subscriber）。本章的另一个例子则使用了来自真实 Web 站点的数据，用以对住房价格和来自 Hot or Not 网站的“热度（hotness）”评价进行建模。

第 8 章，构建价格模型

本章解决的是数值预测问题而非分类问题，期间用到了 **k-最近邻技术**，并且还用到了第 5 章中的优化算法。我们将这些方法与 eBay API 结合在一起构造出一个系统，能够根据拍卖品的一组属性预测出最终的拍卖价格。

第 9 章，高阶分类：核方法与 SVM

本章向读者介绍了如何利用**支持向量机**（support-vector machines）对在线约会网站的用户进行匹配，以及如何将其用于针对专业交友网站的好友信息搜索。支持向量机是一项非常高阶的技术，本章将之与其他方法进行了对比。

第 10 章，寻找独立特征

本章介绍了一种相对较新的技术，称为**非负矩阵因式分解**（non-negative matrix factorization），我们利用这项技术在数据集中寻找独立的特征。对许多数据集而言，其中所包含的内容都是可以借助一组独立特征的组合被重新构造出来的，而这些特征是我们事先不知道的，非负矩阵因式分解的思路便是要寻找这些特征。在本章中，我们利用一组新闻报道说明了该项技术的使用。期间，通过新闻故事来寻找其中的主题，一篇给定的新闻故事中会包含一个或多个这样的主题。

第 11 章，智能进化

本章介绍了**遗传编程**（genetic programming）的概念，这是一组非常复杂的技术，它超出了优化的范畴。并且，这项技术实际上借鉴了进化的思想，它是通过自动构造算法的方式来解决特定问题的。我们通过一个简单的游戏来说明这项技术的应用。在游戏中，计算机最初只是一个学艺不精的初级选手，但是随着游戏的不断进行，它会通过逐步改进其所拥有的代码来提升自己的技能。

第 12 章，算法总结

本章回顾了书中所讲述的所有机器学习算法及统计算法，并将它们与一组人为设计的问题做了对比。这将有助于我们理解算法的工作原理，并形象地说明每种算法划分数据的方法。

附录 A，第三方函数库

给出了有关本书所用的第三方库的信息，例如在哪里可以找到这些第三方库，以及如何进行安装。

附录 B，数学公式

包含了一部分数学公式及其说明，以及本书通篇引入的、以代码形式描述的诸多数学概念。

位于每章末尾处的练习，为读者提供了许多相关信息，借此我们可以对算法进行扩展并使其变得更加强大。

排版约定

Conventions

本书使用了下列排版约定。

普通字体

用于指示菜单标题、菜单选项、菜单按钮，以及键盘快捷键（诸如 Alt 和 Ctrl）。

斜体（*Italic*）

用于指示新的术语、URL、E-mail 地址、文件名、文件扩展名、路径名、目录，以及 UNIX 工具。

等宽字体（`Courier New`）

用于指示命令（command）、命令选项（option）、命令开关（switch）、变量（variable）、属性（attribute）、键值（key）、函数（function）、类型（type）、类（class）、名字空间（namespace）、方法（method）、模块（module）、成员属性（property）、参数（parameter）、值（value）、对象（object）、事件（event）、事件处理器（event handler）、XML 标签、HTML 标签、宏、文件内容或命令的输出结果。

等宽粗体（**`Courier New`**）

用于显示命令或其他应该由用户手工逐字输入的文本。

等宽斜体（*`Courier New`*）

用于显示可替换文本，这些文本应该被替换成用户提供的内容。

该图标代表了提示、建议，或者一般性注释。

使用示例代码
Using Code Examples

本书旨在帮助你完成手头的工作。一般而言，你可以在自己的程序和文档中随意使用书中代码。除非原样引用大量的代码，否则你无须征得我们的许可。例如，在编写程序时引用了本书的若干代码片段是无须许可的。而销售或发行 O'Reilly 图书的示例光盘则是需要许可的。通过引用书中内容及示例代码的方式来答疑解难是无须许可的。而将书中的大量示例代码加入到你的产品文档中则是需要许可的。

如果你在引用时注明出处，我们将不胜感激，但是这并非必须。引用通常包含了标题、作者、出版商，以及 ISBN 号。例如："Programming Collective Intelligence by Toby Segaran. Copyright 2007 Toby Segaran, 978-0-596-52932-1"。

如果你发现自己对示例代码的使用有失公允或违反了上述条款，敬请通过 *permissions@oreilly.com* 与我们取得联系。

如何联系我们
How to Contact Us

请将对本书的评价和存在的问题通过如下地址告知出版者：

美国：

O'Reilly Media, Inc.
1005 Gravenstein Highway North
Sebastopol, CA 95472

中国：

北京市西城区西直门南大街 2 号成铭大厦 C 座 807 室（100035）
奥莱利技术咨询（北京）有限公司

与本书有关的在线信息如下所示:

http://www.oreilly.com/catalog/9780596529321（原书）

如果你想就本书发表评论或提问技术问题，请发送 E-mail 至：

bookquestions@oreilly.com

致谢
Acknowledgments

我想对每一位曾经参与本书编写与出版的 O'Reilly 工作人员表达我的感谢之情。首先，我要感谢 Nat Torkington，是他告诉我写书这个想法值得一试，我还要感谢 Mike Hendrickson 和 Brian Jepson，是他们听了我的絮叨并令我兴致勃勃地撰写本书，我尤其要感谢 Mary O'Brien，他接手 Brian 的编辑工作，并且总是能够缓解我对工程浩大的担忧。

在出版团队中，我要感谢 Marlowe Shaeffer、Rob Romano、Jessamyn Read、Amy Thomson 和 Sarah Schneider，是他们将我的插图与书稿编辑成读者想要看到的形式。

感谢每一位参与本书审校工作的人，尤其是 Paul Tyma、Matthew Russell、Jeff Hammerbacher、Terry Camerlengo、Andreas Weigend、Daniel Russell 和 Tim Wolters。

感谢我的父母。

最后，万分感谢我的几位朋友，他们帮助我利用头脑风暴形成了有关本书的一些想法，并且总能对我无暇陪伴他们表示理解：Andrea Matthews、Jeff Beene、Laura Miyakawa、Neil Stroup 和 Brooke Blumenstein。如果没有你们的支持，写作本书会更为艰难，而且我必会漏掉一些更为有趣的示例。