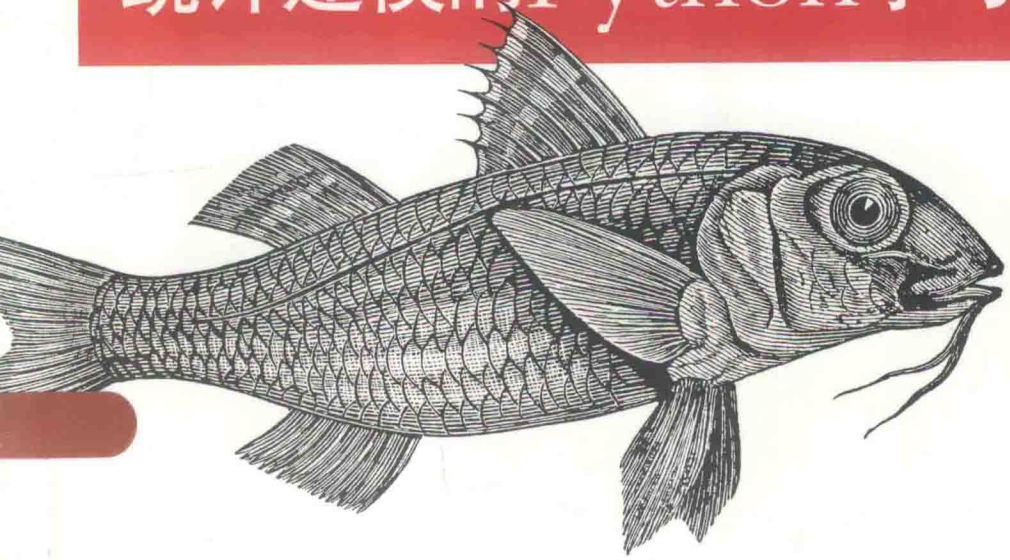
Think Bayes

贝叶斯思维：统计建模的Python学习法



[美] *Allen B. Downey* 著
许杨毅 译

O'REILLY®

人民邮电出版社
POSTS & TELECOM PRESS

贝叶斯思维：统计建模的 Python 学习法

[美] *Allen B. Downey* 著

许杨毅 译

O'REILLY®

Beijing • Cambridge • Farnham • Köln • Sebastopol • Tokyo

O'Reilly Media, Inc.授权人民邮电出版社出版

人 民 邮 电 出 版 社

北 京

图书在版编目（CIP）数据

贝叶斯思维 ：统计建模的Python学习法 / （美）唐尼（Downey, A. B.）著 ；许杨毅译. -- 北京 ：人民邮电出版社，2015.4
ISBN 978-7-115-38428-7

Ⅰ. ①贝… Ⅱ. ①唐… ②许… Ⅲ. ①贝叶斯统计量－统计模型－软件工具－程序设计 Ⅳ. ①O212.8 ②TP311.56

中国版本图书馆CIP数据核字(2015)第040520号

◆ 著　[美] Allen B. Downey
译　许杨毅
责任编辑　王峰松
责任印制　张佳莹　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：787×1000　1/16
印张：11.75
字数：232 千字　2015 年 4 月第 1 版
印数：1 - 3 500 册　2015 年 4 月河北第 1 次印刷

著作权合同登记号　图字：01-2013-8996 号

定价：49.00 元

读者服务热线：(010)81055410　印装质量热线：(010)81055316
反盗版热线：(010)81055315

内容提要

这本书旨在帮助那些希望用数学工具解决实际问题的人们，仅有的要求可能就是懂一点概率知识和程序设计。贝叶斯方法是一种常见的利用概率学知识去解决不确定性问题的数学方法，对于一个计算机专业人士，应当熟悉其在诸如机器翻译、语音识别、垃圾邮件检测等常见的计算机领域的应用。

本书实际上会扩大你的视野，即使不是一个计算机专业人士，你也可以看到在战争环境下（第二次世界大战德军坦克问题），法律问题上（肾肿瘤的假设验证），体育博彩领域中（棕熊队和加人队 NHL 比赛问题）贝叶斯方法的威力。怎么从有限的信息判断德军装甲部队的规模？你所支持的球队有多大可能赢得冠军？在《龙与地下城》勇士中，你应当对游戏角色属性的最大值有怎样的预期？甚至在普通的彩弹射击游戏中，拥有一些贝叶斯思维也能帮助你提高游戏水平。

除此以外，本书在共计 15 章的篇幅中讨论了怎样解决十几个现实生活中的实际问题。在这些问题的解决过程中，作者还潜移默化地帮助读者形成了建模决策的方法论，建模误差和数值误差怎么取舍，怎样为具体问题建立数学模型，如何抓住问题中的主要矛盾（模型中的关键参数），再一步一步地优化或者验证模型的有效性或者局限性。在这个意义上，这本书又是一本关于数学建模的成功样本。

推荐序

很多人把世界理解为基于简单的、确定的，非一即零、非黑即白的。但是真实的世界却是非常复杂的，不是一两个公式可以完美总结概括的。就像我们的高考成绩和我们的学习水平，确实有很大的联系，但是最后又会受到很多因素的影响（比如身体状况，是否休息好了，心情，天气等），进而使得我们的最终成绩在真实水平上下有很大的波动。这就像我们分析很多事情时，经常得到的结论，“既有必然性，又有偶然性”。

这个时候，基于概率和统计的方法给了我们很多的帮助。很多时候，我们不能给出每一个人、每一件事的确定结果。但是当我们观察大量的相同事件后，我们就会发现从一个集体的意义上的规律是存在的。而单个事件每次可能得到不同的结果，这些结果以最有可能的结果为中心，服从一定的概率分布。了解这些分布数据，使我们更加容易理解和预期真实世界的多边形。

回顾在进行计算机自然语言处理过程中走过的路，我们就会发现从研究规则到研究统计的转变。最初，研究人员都认为，语言是基于语法规则。这个也很容易理解，因为我们学习语言的时候，总是背单词，学语法，然后掌握语言。基于这种思维，自然语言处理经历了多年的发展后，遇到了巨大的挑战。那就是即便语法规则已经非常复杂，仍然不能处理大多数的语言情况。从结果上而言，自然语言处理的准确度远低于人类，不具有真正的使用价值。而后，有一批学者开始另辟蹊径，基于统计的思路进行探索。如果语言是根据人类沟通需求自然发生，然后才有总结出来的语法呢？基于这种思想，研究人员放弃语法规则，开始建立基于统计的模型。他们使用了大量的真实文本数据，分析每个词和它前后的词出现的统计关系，用贝叶斯方法以及马尔科夫过程，建立了新的自然语言处理模型。这一次，语言处理准确率有了巨大的提升，进而达到可以实用的要求。今天，当我们使用谷歌翻译、苹果的 Siri 语音服务的时候，后面都有基于统计的模型的功劳。

还有很多的真实世界的事情都是这样的，比如路上的交通是否阻塞、银行排队的时间、球赛的比赛结果，都是以一种概率的形式出现的。了解贝叶斯方法，也是了解真实世

界运行的一种有效途径。本书中也列举了很多的真实实例来告诉我们，贝叶斯方法和真实世界的联系。

另外，在我们正在经历的大数据时代，作为数据分析方法的一个巨大分支，基于贝叶斯的机器学习算法也在被广泛地使用，并产生很多实际意义。比如简单贝叶斯算法、贝叶斯信念网络等，被广泛地应用于分类和预测。对于海量数据的文本分类问题，例如，垃圾邮件的甄选和过滤，基于贝叶斯方法的算法取得了非常好的效果，并在很多公司中正在使用，帮助我们远离垃圾邮件的骚扰。

更加难能可贵的是，本书作者用相对简单的 Python 语言，对所涉及的实例进行了编程。对于有一定计算机基础的人来说，通过程序，可以进一步理解贝叶斯方法的应用，真正掌握并且可以利用这些程序达到举一反三的效果。

本书用简洁的语言，大量的实例和故事，辅之以简单的 Python 语言，把原本枯燥的概率理论讲得生动且容易理解。在学习到理论的同时，还了解了它的真实意义以及可以使用的地方。对于有追求的工程师和感兴趣的读者而言，这是一本提升自我的很好的图书。

酷我音乐 雷鸣[①]

① 雷鸣，现任酷我音乐董事长、CEO，国家千人计划特聘专家，百度创始七剑客之一，百度搜索引擎的早期设计者和技术负责人之一。获北京大学计算机科学硕士学位和斯坦福大学商学院 MBA 学位，曾任北京大学计算机系学生会主席和斯坦福大学中国学生学者联合会副主席。

前言

学习之道

这本书以及 Think 系列其他书籍的一个前提是：只要懂得编程，你就能用这个技能去学习其他的内容 。

绝大多数贝叶斯统计的书使用数学符号并以数学概念的形式表示数学思想，比如微积分。但本书使用了 Python 代码而不是数学，离散近似而不是连续数学。结果就是原本需要积分的地方变成了求和，概率分布的大多数操作变成了简单的循环。

我认为这样的表述是易于理解的，至少对于有编程经验的人们来说是这样的。当作建模选择时也非常实用，因为我们可以选取最合适的模型而不用担心偏离常规分析太多。

另外，这也提供了一个从简化模型到真实问题的平滑发展路线，第 3 章就是一个好示例。它由一个关于骰子的简单例子开始，那是基本概率的一个主题；紧接着谈到了一个我从 Mosteller《50 个挑战的统计学难题》（Fifty Challenging Problems in Probability）一书中借用的火车头问题；最后是德军坦克问题，这个第二次世界大战中成功的贝叶斯方法应用案例。

建模和近似

本书中多数章节的灵感都是由真实世界里的问题所激发的，所以涉及了一些建模知识，在应用贝叶斯方法（或者其他的分析方法）前，我们必须决定真实世界中的哪些部分可以被包括进模型，而哪些细节可以被抽象掉。

例如，第 7 章中那个预测冰球比赛获胜队伍的例子，我将进球得分建模为一个泊松过程，这预示着在比赛的任何时段进球机会都是相等的，这并不完全符合实际情况，但就大多数目的来说可能就够了。

第 12 章中，问题是对 SAT 得分进行解释（SAT 是用于全美大学的入学标准测试）。我以一个假设所有 SAT 试题难度相同的简化模型开始，但其实 SAT 的试题设计中既包括了相对容易，也包括了相对较难的试题。随后提出了第二个反映这一设计目的的模型，结果显出两个模型在最终效果上没有大的差别。

我认为在解决问题的过程中，明确建模过程作为其中一部分是重要的，因为这会提醒我们考虑建模误差（也就是建模当中简化和假设带来的误差）。

本书中的很多方法都基于离散分布，这让一些人担心数值误差，但对于真实世界的问题，数值误差几乎从来都小于建模误差。

再者，离散方法总能允许较好的建模选择，我宁愿要一个近似的良好的模型也不要一个精确但却糟糕的模型。

从另一个角度看，连续方法常在性能上有优势，比如能以常数时间复杂度的解法替换掉线性或者平方时间复杂度的解法。

总的来说，我推荐这些步骤的一个通用流程如下。

1. 当研究问题时，以一个简化模型开始，并以清晰、好理解、实证无误的代码实现它。注意力集中在好的建模决策而不是优化上。
2. 一旦简化模型有效，再找到最大的错误来源。这可能需要增加离散近似过程当中值的数量，或者增加蒙特卡洛方法中的迭代次数，或者增加模型细节。
3. 如果对你的应用而言性能就已经足够了，则没必要再优化。但如果要做，有两个方向可以考虑：评估你的代码以寻找优化空间，例如，如果你缓存了前面的计算结果，你也许能避免重复冗余的计算；或者可以去发现找到计算捷径的分析方法。

这一流程的好处是第一、第二步较快，所以你能在投入大量精力前研究多个可替代的模型。

另一个好处是在第三步，你可以从一个大体正确的可参考实现开始进行回归测试。也就是，检查优化后的代码是否得到了同样的结果，至少是近似的结果。

代码指南

本书中的很多例子使用了在 `thinkbayes.py` 当中定义的类和函数，可以从

http://thinkbayes.com/thinkbayes.py 下载这个模块。

本书大多数章节包括了可以从 *http://thinkbayes.com* 下载的代码，其中有一些依赖代码也需要下载，我建议你将这些文件全部放入同一个目录，这样代码间就可以彼此引用而无需变更 Python 的库文件搜索路径。

你可以在需要时再下载这些代码，或者一次性从 *http://thinkbayes.com/thinkbayes_code.zip* 下

载，这个文件也包括了某些程序使用的数据文件，当解压时，将创建名为 thinkbayes_code 的包括本书中所有代码的目录。

另外，如果是 Git 用户，你可以通过 fork 和 clone 来一次性获得这个仓库：*https://github.com/AllenDowney/ThinkBayes*。

我用到的模块之一是 thinkplot.py，它对 pyplot 中一些函数进行了封装，要使用它需要安装好 matplotlib，如果还没有，检查你的软件包管理器看看它是否存在，否则你可以从 *http://matplotlib.org* 得到下载指南。

最后，本书中一些程序使用了 NumPy 和 SciPy，可以从 http://numpy.org 和 http://scipy.org 获得。

编码风格

有经验的 Python 程序员会注意到本书中的代码没有符合 PEP 8 这一最通用的 Python 编码指南（*http://www.python.org/dev/peps/pep-0008/*）。

确切地说，PEP 8 使用带有词间下划线的小写函数名 like_this，而在本书中和实现的代码里，函数和方法名以大写开头并使用间隔式的大小写，LikeThis。

没有遵循 PEP 8 规范的原因是在我为书中内容准备代码时正在谷歌做访问学者，所以就遵循了谷歌的编码规范，它只在少数地方沿袭了 PEP 8，一用上了谷歌风格我就喜欢上了，现在要改太麻烦。

同样，在主题风格上，如在“Bayes's theorem”中，s 放在单引号后，在某些风格指南中倾向这样使用而在其他指南当中不是。我没有特别的偏好，但不得不选择其一，所以就是你们现在看到的这个。

最后一个排版上的注脚是：贯穿全书，我使用 PMF 和 CDF 表示概率密度函数或累积分布函数这些数学概念，而 Pmf 和 Cdf 是指我所表述的 Python 对象。

预备条件

还有几个出色的能在 Python 中进行贝叶斯统计的模块，包括 pymc 和 OpenBUGS，由于读者需要有相当多的背景知识才能开始使用这些模块，因此本书中我没有使用它们，而且我想使阅读本书的预备条件最小。如果你了解 Python 和一点点概率知识，就可以开始阅读本书。

第 1 章关于概率论和贝叶斯定理，没有程序代码。第 2 章介绍了 Pmf，一望而知是用来表示概率密度函数（PMF）的 Python 字典对象。然后第 3 章我介绍了 Suite，一个 Pmf 对象，也是一个能进行贝叶斯更新的框架，因而万事具备了。

好了，随后的章节中，我使用了高斯（正态）分布，二次和泊松分布，beta 分布等各种分析型的概率分布，在第 15 章，我介绍了不太常见的狄利克雷分布，不过接着也进行了解释。如果你不熟悉这类分布，可以从维基百科了解它们。也可以阅读本书的一本指南《统计思维》（Think Stats），或其他入门级的统计学书籍（不过，恐怕大多数类似书籍都会采取对实战没有太大帮助的数学方法来阐述）。

书中使用的惯例写法

本书中使用了下面的印刷惯例。

斜体（*Italic*）

表示新术语，URL，邮件地址，文件名和文件扩展名。

等宽（`Constant width`）

用于程序代码，也包括那些表示程序代码元素的段落，例如，变量和函数名，数据库，数据类型，环境变量，声明和关键字。

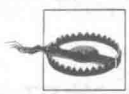
等宽粗体（**`Constant width bold`**）

命令或者其他由用户输入的文字。

等宽斜体（*`Constant width italic`*）

应该由用户输入值替换或者由上下文决定的文本。

这个图标表示这是一个提示、建议或者一般性的注记。



这个图标表示这是提醒或者警示。

我们的联系方式

如果你想就本书发表评论或有任何疑问，敬请联系出版社。

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

我们还为本书建立了一个网页，其中包含了勘误表、示例和其他额外的信息。你可以通过地址访问该网页：*http://oreil.ly/think-bayes*。

关于本书的技术性问题或建议，请发邮件到：bookquestions@oreilly.com。

欢迎登录我们的网站（*http://www.oreilly.com*），查看更多我们的书籍、课程、会议和最新动态等信息。

我们的其他联系方式如下。

Facebook：*http://facebook.com/oreilly*

Twitter：*http://twitter.com/oreillymedia*

YouTube：*http://www.youtube.com/oreillymedia*

贡献者列表

如果你发现本书有需要更正的地方或者其他建议，请发送电子邮件至 downey@allendowney.com。一旦根据你的反馈进行了修正，我会将你加入贡献者列表（除了要求不署名的情况）。

提供包含错误之处的段落部分，会让我更容易找到它们。只提供页和节数也可以，但还是不太容易找到错误之处。这里先致谢！

- 首先，我要感谢大卫·麦凯（David MacKay）的优秀作品《信息理论、推理和学习算法》（Information Theory, Inference, and Learning Algorithms），我从这本书里第一次理解了贝叶斯方法。他允许我使用他书中的几个问题来作为例子。
- 这本书也得益于我和圣乔恩·马哈的相互配合，2012 的秋天我在欧林学院审核了他的贝叶斯推理课程。
- 在参加波士顿 Python 用户组项目时，我在夜班时间完成了本书的部分内容，所以我也要感谢他们以及他们所提供的比萨。
- 乔纳森·爱德华兹提交了第一个拼写错误。
- 乔治·珀金斯发现了一个标记错误。
- 奥利维尔提出了几个有益的建议。
- 尤里·帕西奇尼克发现了几个错误。

- 克里斯托弗·欧霍特提交了一个更正和建议的清单。
- 罗伯特·马库斯发现了一个错误放置的小写 i。
- 麦克斯·黑尔珀林建议在第 1 章提供一个澄清章节。
- 马库斯·杜布勒指出“从碗中有放回的取出饼干”并不是一个真实的场景。
- 汤姆·波拉德和保罗 A. 吉安纳罗斯指出，在火车头案例中的某些数量有版本问题。
- 兰姆·林布发现了一个拼写错误，还建议了澄清章节。
- 2013 春天在我的《贝叶斯统计计算》课程上，学生们提出了许多有益的修正和建议，他们是：凯·奥斯汀，克莱尔·巴尼斯，卡里·本德尔，瑞秋·铂伊，凯特·门多萨，阿琼·伊耶，本·克罗普，内森·林之，凯尔·麦克康诺亥，亚历克·雷德福，布伦丹·里特，埃文·辛普森。
- 格雷戈·马拉和马特·艾提帮我澄清了“正确的价格”这个问题的一些讨论。
- 马库斯·奥格伦指出火车头问题的原有声明是有些含糊的。
- O'Reilly Media 的贾斯敏和丹在校对书的过程中也发现了许多可改进的地方。

目录