Python 贝叶斯深度学习

马特 • 贝纳坦(Matt Benatan)
[英] 约赫姆 • 吉特马(Jochem Gietema) 著
玛丽安 • 施耐德(Marian Schneider)
郭 涛 译

清華大學出版社
北 京

北京市版权局著作权合同登记号　图字：01-2023-5218

图书在版编目（CIP）数据

Python贝叶斯深度学习 / (英) 马特·贝纳坦 (Matt Benatan), (英) 约赫姆·吉特马 (Jochem Gietema), (英) 玛丽安·施耐德 (Marian Schneider) 著 ; 郭涛译. -- 北京 : 清华大学出版社, 2024. 10. -- ISBN 978-7-302-67216-6

Ⅰ. TP181

中国国家版本馆CIP数据核字第20247M9Q28号

责任编辑：王　军
装帧设计：孔祥峰
责任校对：马遥遥
责任印制：曹婉颖

出版发行：清华大学出版社
网　　址：https://www.tup.com.cn，https://www.wqxuetang.com
地　　址：北京清华大学学研大厦 A 座　　邮　　编：100084
社 总 机：010-83470000　　邮　　购：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质 量 反 馈：010-62772015，zhiliang@tup.tsinghua.edu.cn

印 装 者：大厂回族自治县彩虹印刷有限公司
经　　销：全国新华书店
开　　本：170mm×240mm　　**印　　张：**14.25　　**插　　页：**4　　**字　　数：**356 千字
版　　次：2024 年 10 月第 1 版　　**印　　次：**2024 年 10 月第 1 次印刷
定　　价：79.80 元

产品编号：104118-01

译 者 序

机器学习的大流派分别是符号主义、贝叶斯派、联结主义、行为主义、进化主义和行为类比主义。其中，贝叶斯派的主要核心思想是进行主观概率估计，其典型代表是贝叶斯推理，而联结主义则源于神经科学，最典型的代表是深度学习。本书主题是贝叶斯深度学习，顾名思义，它是贝叶斯推理和深度学习结合的产物。而支撑该主题的是这两者碰撞产生的新思想、新算法。经典的贝叶斯思想是统计推理——一种统计/概率范式。统计推理是贝叶斯推理框架中最重要的部分，也是概率机器学习的核心部分。这与频率主义思想完全不同，它并没有给出隐变量的确切值，而是保留模型具有的不确定性，最终得到隐变量的概率分布。

不言而喻，深度学习近十年得到了充分发展，取得了一系列成果。但对于深度学习这个强大的黑盒预测器而言，量化不确定性是一个具有挑战性且尚未解决的问题。贝叶斯神经网络能够学习权重分布，是目前对预测进行不确定性估计的最先进技术。而即使将深度学习与贝叶斯推理结合，形成贝叶斯深度学习新方法，依然存在训练过程难、计算成本高等问题。为了解决上述种种问题，深度集成方法应运而生。该方法借助集成学习理论基础而形成，可以快速对预测进行不确定性的估计。

实际上，贝叶斯神经网络思想早在 20 世纪 90 年代便由 Neal、Mackay 和 Bishop 等学者提出，但由于当时整个神经网络研究领域几乎停滞，并且训练神经网络的计算需求太大，因此贝叶斯深度学习仅仅停留在理论讨论起步阶段。而现代贝叶斯神经网络主要专注于对变分推理方法进行探索研究。

贝叶斯深度学习、深度集成学习、集成学习、贝叶斯推理等这些术语以及这些理论的推理过程，会让人望而生畏，畏葸不前。不过，本书并不是一本讨论贝叶斯深度学习理论的著作，而是将理论进行了简化，着重讲述其代码实现，以及贝叶斯深度学习工具集使用方面的实战技巧。本书内容共分三部分，第一部分(第 1~3 章)是基础概念和理论，主要介绍深度学习的发展历史和局限性，以及它与贝叶斯推理结合的时机、贝叶斯推理基础、深度学习基础；第二部分(第 4~7 章)主要介绍贝叶斯深度学习的基本思想、使用原则、标准工具集代码实现、实际考虑因素；第三部分(第 8~9 章)，讲述贝叶斯深度学习的应用和发展趋势。本书内容新颖、实战性强，填补了目前该领域的市场空白。此外，如果读者想进一步深入学习贝叶斯推理、集成学习等方面的主题，可参阅译者翻译的《概率图模型原理与应用(第 2 版)》《Python 贝叶斯建模与计算》和《集成学习实战》等图书。

在翻译本书的过程中，我查阅了大量的经典著(译)作，也得到了很多人的帮助。成都文理学院外国语学院的何静老师、刘晓骏博士参与了本书的审校工作，感谢她们所做的工作。此外，我还要感谢清华大学出版社的编辑、校对和排版人员，感谢他们为了保证本书质量所做的一切努力。

由于本书涉及内容广泛、深刻，加上译者翻译水平有限，书中难免存有不足之处，恳请各位读者不吝指正。

译者

2024 年 4 月

译 者 简 介

郭涛，主要从事人工智能、智能计算、概率与统计学、现代软件工程等前沿交叉研究。出版多部译作，包括《Python 贝叶斯建模与计算》《概率图模型原理与应用(第 2 版)》和《集成学习实战》。

作 者 简 介

Matt Benatan 博士是搜诺思(Sonos)的首席研究科学家，主要负责智能个性化系统的研究。他还获得了曼彻斯特大学的西蒙工业奖学金，并在那里合作开展了多个人工智能研究项目。Matt 在利兹大学获得了视听语音处理博士学位，之后进入工业界，在信号处理、材料发现和欺诈检测等多个领域开展机器学习研究。Matt 曾与他人合著了 Wiley 出版社出版的 *Deep learning for Physical Scientists* 一书，他目前的主要研究兴趣包括面向用户的人工智能、优化和不确定性估计。

Matt 不仅要对妻子 Rebecca 的关心、耐心和支持深表感激，也要对父母 Dan 和 Debby 的不懈热情、指导和鼓励深表感激。

Jochem Gietema 在阿姆斯特丹学习哲学和法律，毕业后转入机器学习领域。他目前在伦敦的 Onfido 公司担任应用科学家，在计算机视觉和异常检测领域开发并部署了多项专有的解决方案。Jochem 热衷于研究不确定性估计、交互式数据可视化以及用机器学习解决现实世界中的问题。

Marian Schneider 博士是机器学习和计算机视觉领域的应用科学家。他在马斯特里赫特大学获得了计算视觉神经科学博士学位。此后，他从学术界转入工业界，开发了一些机器学习解决方案并将其应用于多种产品，涵盖从大脑图像分割到不确定性估计，再到移动电话设备上更智能的图像获取等方面。

Marian 非常感谢他的伴侣 Undine，因为在本书的写作过程中 Undine 给予了他大力支持，尤其是在周末的宝贵时光里陪伴他，从而使本书的写作工作得以顺利进行。

关于审稿人

Neba Nfonsang 是一位数据科学家，也是丹佛大学数据科学与统计学专业的讲师，在丹佛大学获得了研究方法和统计学博士学位。Neba 在学术界和工业界都有丰富的工作经验，并教授过 12 门研究生课程，包括数据科学和统计学课程。他曾向全球多家公司传授高级分析技巧，并就设计生产就绪的统计和机器学习模型方面提供培训和最佳实践指导。

Avijit K 是一名出色的首席信息官(Chief Information Officer，CIO)，在数据科学和人工智能领域拥有超过 15 年的经验。他拥有一流大学的计算机科学博士学位，在机器学习、深度学习、计算机视觉、自然语言处理和相关技术领域完成了多个行业项目。Avi 目前在一家服务公司担任首席信息官，负责监管公司的技术运营，包括数据分析、基础设施和软件开发。

前　言

在过去的十年中，机器学习领域取得了长足的进步，并因此激发了公众的想象力。但我们必须记住，尽管这些算法令人印象深刻，但它们并非完美无缺。本书旨在通过平实的语言介绍如何在深度学习中利用贝叶斯推理，帮助读者掌握开发“知其所不知”模型的工具。这样，开发者就能开发出更鲁棒的深度学习系统，以便更好地满足现今基于机器学习的应用需求。

本书读者对象

本书面向从事机器学习算法开发和应用的研究人员、开发人员和工程师，以及希望开始使用不确定性感知深度学习模型的人员。

本书主要内容

第 1 章“深度学习时代的贝叶斯推理”介绍传统深度学习方法的用例和局限性。

第 2 章“贝叶斯推理基础”讨论贝叶斯建模和推理，同时探索了贝叶斯推理的黄金标准机器学习方法。

第 3 章“深度学习基础”介绍深度学习模型的主要构建模块。

第 4 章“贝叶斯深度学习介绍”结合第 2 章和第 3 章介绍的概念讨论贝叶斯深度学习。

第 5 章“贝叶斯深度学习原理方法”介绍贝叶斯神经网络近似的原理方法。

第 6 章“使用标准工具箱进行贝叶斯深度学习”介绍利用常见的深度学习方法推进模型不确定性估计。

第 7 章“贝叶斯深度学习的实际考虑因素”探讨和比较第 5 章和第 6 章介绍的方法的优缺点。

第 8 章“贝叶斯深度学习应用”概述贝叶斯深度学习的各种实际应用，如检测分布外数据或数据集漂移的鲁棒性。

第 9 章“贝叶斯深度学习的发展趋势”讨论贝叶斯深度学习的一些最新发展趋势。

如何充分利用本书

为了充分利用本书，你需要具备一定的机器学习和深度学习先验知识，并熟悉贝叶斯推理的相关概念。掌握一些使用 Python 和机器学习框架(如 TensorFlow 或 PyTorch)的实用知识也很

有价值，但并非必要。

建议使用 Python 3.8 或更高版本，因为本书所有代码都已经通过 Python 3.8 的测试。第 1 章将给出为本书中的示例代码设置环境的详细说明。

下载示例代码文件和彩色图片

本书的代码包也托管在 GitHub 上，网址是 https://github.com/PacktPublishing/Enhancing-Deep-Learning-with-Bayesian-Inference。如果代码有更新，将在现有的 GitHub 仓库中进行更新。读者也可以通过扫描封底的二维码下载本书的示例代码文件。

另外，我们还提供了一个 PDF 文件，其中包含本书所有截图/图表的彩色图片，通过扫描封底的二维码可下载该 PDF 文件。本书文前页中的“彩插”部分也列出了书中提到的部分彩色图片，供读者在阅读时方便查看。

目　录

第1章

深度学习时代的贝叶斯推理

在过去的十五年里，**机器学习(Machine Learning，ML)**从一个相对鲜为人知的领域变成了科技界的热门词汇，这在很大程度上要归功于**神经网络(Neural Network，NN)**取得的令人印象深刻的成就。**深度学习(Deep Learning，DL)**曾经是该领域的一个小众方向，但它在几乎所有可以想象到的应用领域取得的成就使其普及程度迅速上升。我们不再沉溺于其所提供的功能，而是期待它全面开花。从在社交网络应用程序中应用过滤器，到出国度假时依赖谷歌翻译，不可否认的是，深度学习现在已真正融入技术领域。

但是，尽管深度学习取得了令人瞩目的成就，提供了各种各样的产品和功能，但它还没有跨过最后的障碍。随着复杂的神经网络越来越多地应用于任务关键型和安全关键型应用中，围绕其鲁棒性出现的问题也越来越多。许多深度学习算法的黑箱性质让精通安全的解决方案架构师望而生畏，以至于许多人宁愿选择低于标准性能的系统，也不愿冒使用不透明系统带来的潜在风险。

那么，如何才能克服深度学习带来的忧虑，确保创建出更鲁棒、更值得信赖的模型呢？**可解释人工智能(Explainable Artificial Intelligence，XAI)**能够提供一些答案，而**贝叶斯深度学习(Bayesian Deep Learning，BDL)**领域则是一个重要的基石。在本书中，你将通过学习实际案例发现贝叶斯深度学习背后的基本原理，从而加深对该领域的理解，并掌握构建自己的贝叶斯深度学习模型所需的知识和工具。

不过，在开始之前，我们先深入探讨一下贝叶斯深度学习的合理性，以及为什么典型的深度学习方法可能并不像我们所想的那样鲁棒。在本章中，首先，我们将了解深度学习的一些成功和失败案例，以及贝叶斯深度学习如何帮助我们避免标准深度模型可能带来的悲剧性后果。其次，我们将概述本书其余部分的核心主题。最后，我们将介绍在实际示例中使用的库和数据。

本章主要内容：

- 深度学习时代的奇迹
- 了解深度学习的局限性
- 核心主题
- 设置工作环境

1.1 技术要求

所有代码都可以在本书的 GitHub 仓库中找到，网址为 https://github.com/PacktPublishing/Enhancing-Deep-Learning-with-Bayesian-Inference；也可以通过扫描本书封底的二维码进行下载。

1.2 深度学习时代的奇迹

在过去的 10~15 年中，由于深度学习取得了巨大成功，我们看到机器学习领域发生了巨大变化。深度学习的普遍应用给人留下的最深刻印象之一，或许就是它已影响了医学影像、制造业一直到翻译和内容创建工具等各个领域。

虽然深度学习在近几年才取得巨大成功，但其许多核心原理早已确立。研究人员使用神经网络已有一段时间了，事实上，可以说第一个神经网络(即感知器)早在 1957 年就已由 Frank Rosenblatt 提出！当然，这个早期的感知器并不像我们今天构建的模型那么复杂，但它是这些模型的一个重要组成部分，如图 1.1 所示。

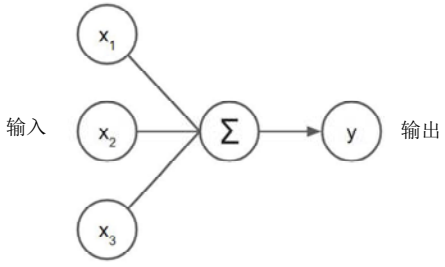


图 1.1 单个感知器示意图

20 世纪 80 年代，Kunihiko Fukushima 提出了**卷积神经网络(Convolutional Neural Network，CNN)**的概念，John Hopfield 于 1982 年开发了循环神经网络(Recurrent Neural Network，RNN)，许多如今耳熟能详的概念由此问世。20 世纪 80 年代和 90 年代，这些技术进一步成熟：1989 年，Yann LeCun 应用反向传播技术创建了一个能够识别手写数字的卷积神经网络，而 Hochreiter 和 Schmidhuber 则在 1997 年提出了长短期记忆循环神经网络的重要概念。

虽然我们在世纪之交以前就已经具备了构建当今强大模型的基础，但直到现代 GPU 的引入，这一领域才真正发展。有了 GPU 的加速训练和推理功能，我们才有可能开发出拥有数十(甚至数百)层的网络。这就为实现令人难以置信的复杂神经网络架构打开了大门，使其能够学习复杂、高维数据的紧凑特征表示。

AlexNet 是最早的极具影响力的网络架构之一，如图 1.2 所示。该网络由 Alex Krizhevsky、Ilya Sutskever 和 Geoffrey Hinton 开发，共有 11 层，能够将图像分为 1,000 个可能的类别。它在 2012 年举办的 ImageNet 大规模视觉识别挑战赛上取得了前所未有的优异成绩，展示了深度网络的强大威力。AlexNet 是第一个有影响力的神经网络架构，在随后的几年中，许多现在已经耳熟能详的架构相继问世，包括 VGG Net、Inception 架构、ResNet、EfficientNet、YOLO 等，

这样的例子不胜枚举！

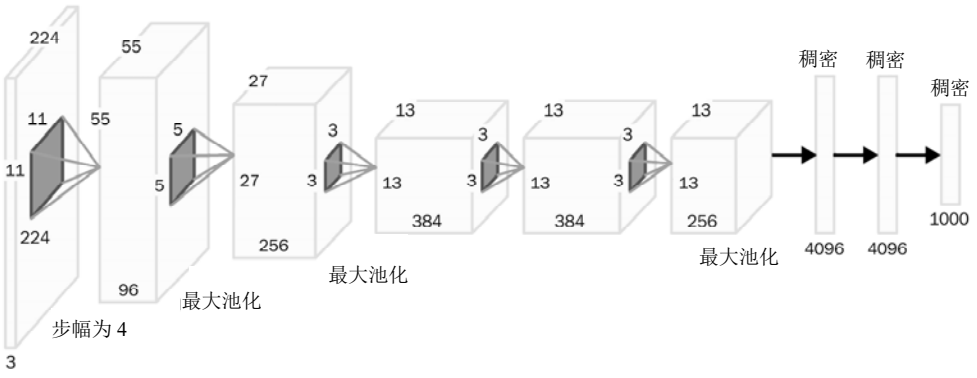


图 1.2　AlexNet 示意图

但是，神经网络不只是在计算机视觉应用中取得了成功。2014 年，Dzmitry Bahdanau、Kyunghyun Cho 和 Yoshua Bengio 的研究表明，端到端神经网络模型可以在机器翻译中获得最先进的结果。这是该领域的分水岭，大规模机器翻译服务迅速采用了这些端到端网络，从而推动了自然语言处理的进一步发展。时至今日，这些概念已经成熟，并产生了 **Transformer** 架构。这种架构通过自监督学习的能力学习丰富的特征嵌入，并对深度学习产生了巨大的影响。

由于各种架构赋予了神经网络令人印象深刻的灵活性，因此神经网络已在几乎所有可以想象到的领域的应用中实现了最先进的性能，它们现在已成为人们日常生活中熟悉的一部分。无论是我们在移动设备上使用的面部识别、谷歌翻译(Google Translate)等翻译服务，还是智能设备中所用的语音识别技术，这些网络显然不仅仅在图像分类挑战中具有竞争力，它们已经成为我们正在开发的技术的重要组成部分，甚至能够超越人类。

尽管有关深度学习模型超越人类专家的报道越来越频繁，但最著名的例子可能要数医学成像领域了。2020 年，伦敦帝国理工学院和谷歌健康公司的研究人员开发的一个网络，从乳房 X 光照片中检测乳腺癌方面的表现超过了六位放射科医生。几个月后，2021 年 2 月的一项研究表明，在诊断胆囊疾病方面，深度学习模型的表现优于两名人类专家。同年晚些时候发表的另一项研究表明，在从皮肤异常图像中检测黑色素瘤方面，卷积神经网络的表现优于 157 位皮肤科医生。

到目前为止，我们讨论的所有应用都是机器学习的监督应用，其中的模型都是针对分类或回归问题进行训练的。然而，深度学习最令人印象深刻的成就还体现在其他应用中，包括生成式模型和强化学习。后者最著名的例子之一可能就是 DeepMind 开发的强化学习模型 **AlphaGo**。顾名思义，该算法通过强化学习来训练下围棋。国际象棋等一些游戏可以通过使用相当简单的人工智能方法解决，而围棋则不同，从计算的角度来看，它的挑战性要大得多。这是由于围棋具有复杂性，许多可能的棋步组合对于更传统的方法来说是困难的。因此，当 AlphaGo 分别于 2015 年和 2016 年成功击败围棋冠军樊麾和李世石时，这可是个大新闻。

DeepMind 接着进一步完善了 AlphaGo，创建了一个通过与自己对弈来学习的版本，即 AlphaGo Zero。这个模型优于之前的任何模型，在围棋中取得了超越人类的表现。AlphaGo 成

功的核心算法 AlphaZero 在一系列其他游戏中也取得了超越人类的表现，证明了该算法在其他应用中具有的泛化能力。

在过去十年中，深度学习的另一个重要里程碑是**生成对抗网络(Generative Adversarial Network，GAN)**的问世。GAN 采用了两个网络：第一个网络的目标是生成与训练集具有相同统计质量的数据；第二个网络的目标是利用从数据集中学到的知识，对第一个网络的输出进行分类。由于第一个网络并没有直接在数据上训练，因此它不是简单地复制数据，而是有效地学会了欺骗第二个网络。这就是使用“对抗”一词的原因。通过这一过程，第一个网络能够学习哪种输出能成功欺骗第二个网络，从而生成与数据分布相匹配的内容。

GAN 可以生成特别令人印象深刻的输出结果。例如，图 1.3 就是由 StyleGAN2 模型生成的。

图 1.3　由 StyleGAN2 生成的人脸，源自 thispersondoesnotexist.com

但是，GAN 的强大之处不仅体现在生成逼真的人脸方面，在许多其他领域也有实际意义，例如，为药物发现提供分子组合建议。另外，它们还是通过数据增强来改进其他机器学习方法(使用 GAN 生成的数据)从而增强数据集的强大工具。

所有这些成功案例可能会让深度学习看起来无懈可击。尽管深度学习的成就令人印象深刻，但这并不能说明全部问题。在下一节中，我们将了解深度学习存在的一些不足，并开始了解未来贝叶斯方法如何帮助避免这些不足。

1.3　了解深度学习的局限性

正如你所见，深度学习已经取得了一些了不起的成就，不可否认，它正在彻底改变我们处理数据和预测建模的方式。但是，深度学习短暂的历史中也经历过一些黑暗时刻，这些故事为开发更鲁棒、更安全的系统提供了重要的经验教训。

在本节中，我们将介绍几个深度学习失败的关键案例，并从贝叶斯视角讨论它如何有助于产生更好的结果。

1.3.1　深度学习系统中的偏见

我们将从一个教科书式的**偏见**示例开始讲解，这是数据驱动方法面临的一个关键问题。该示例围绕亚马逊展开。作为一家家喻户晓的电子商务公司，亚马逊的出现开始彻底改变了图书

零售业，后来亚马逊成为几乎可以买到任何物品的一站式商店：从花园家具到新笔记本电脑，甚至家庭安全系统，只要你能想到的，都可能在亚马逊上买到。亚马逊在技术上也取得了长足进步，这通常是为了改善基础设施，从而实现业务扩张。从硬件基础设施到优化方法上的理论和技术飞跃，亚马逊从最初的电子商务公司发展成为技术领域的关键角色之一。

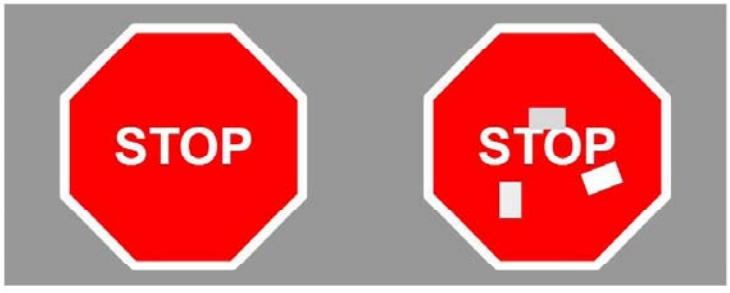
虽然这些技术飞跃往往为行业树立了标准，但这个案例却恰恰相反，它展示了数据驱动方法具有的一个关键弱点。我们所说的案例是亚马逊的人工智能招聘软件。自动化在亚马逊的成功中扮演着关键角色，因此将这种自动化扩展到审核简历上也是合情合理的。2014 年，亚马逊的机器学习工程师部署了一个工具来实现这一目标。该工具以过去 10 年的求职者为训练对象，旨在从公司庞大的求职者库中学习识别出有利的特质。然而，到了 2015 年，该工具在特征获取方面明显出现了一些弊端，导致产生了严重的不良行为。

该问题很大程度出现在基础数据上：由于当时科技行业的性质，亚马逊的简历数据集以男性求职者为主。这导致模型预测出现了极大不公平：它实际上学会了偏向男性，而对女性求职者有极大的偏见。该模型的歧视行为导致亚马逊放弃了该项目，现在它已成为人工智能界“偏见歧视”的一个重要范例。

在这里提出的问题中，需要考虑的一个重要因素是，这种偏见不仅仅是由**显性**信息驱动的，比如一个人的名字(这可能为性别提供线索)：算法会学习潜在信息，然后驱动偏见。这意味着该问题不能简单地通过匿名化来解决——工程师和科学家们要确保对偏见进行全面评估，从而使我们部署的算法是公平的。虽然贝叶斯方法无法让偏见消失，但它为我们提供了一系列有助于解决这些问题的工具。正如你在本书后面看到的，贝叶斯方法能够确定数据是在分布内还是在**分布外(Out-of-Distribution，OOD)**。在亚马逊这个示例中，可以利用贝叶斯方法提供的这一功能：分离分布外数据并对其进行分析，以了解为什么该数据是分布外的。它识别了一些相关信息，例如具有不合适经验的申请人？或识别了一些不相关的歧视性信息，例如申请人的性别？这可以帮助亚马逊的人工智能团队及早发现不良行为，从而制定出不偏不倚的解决方案。

1.3.2　过高置信预测导致危险

另一个被广泛引用的深度学习失败案例，出现在 Kevin Eykholt 等人的论文 *Robust Physical-World Attacks on Deep Learning Visual Classification*(https://arxiv.org/abs/1707.08945)中。这篇论文在强调深度学习模型的**对抗性攻击(adversarial attack)**问题上发挥了重要作用，该论文认为，对输入数据稍加修改，模型就会产生错误的预测。在该论文的一个重要例子中，作者在一个停车标志上贴上了白色和黑色贴纸，如图 1.4 所示。虽然对标志的修改很细微，但计算机视觉模型却将修改后的标志解释为“限速 45”标志。



分类：“停车”　　分类：“限速 45 MPH”

图 1.4　简单对抗攻击对解释停车标志的模型的影响说明

起初，这似乎无关紧要，但如果我们退一步，考虑到特斯拉、优步和其他公司对自动驾驶汽车所做的大量工作，就不难理解这种对抗性扰动会产生怎样的灾难性后果。在该停车标志的案例中，这种错误分类可能会导致自动驾驶汽车绕过“停止”标志，冲向十字路口的车流。这显然对乘客或其他行人不利。事实上，2016 年曾发生过一起与这里所描述的情况相似的事件，当时一辆特斯拉 Model S 在佛罗里达州北部与一辆卡车相撞(https://www.reuters.com/article/us-tesla-crash-idUSKBN19A2XC)。根据特斯拉公司的说法，特斯拉的自动驾驶系统没有检测到卡车后挂的拖车，因为它无法从拖车身后明亮的天空背景中分辨出拖车。司机也没有注意到拖车，最终导致了致命的撞击。但是，如果自动驾驶汽车所使用的决策过程更加复杂，我们是否就可以信任它呢？本书的一个重要主题就是利用机器学习系统做出鲁棒的决策，尤其是在任务关键型或安全关键型应用中。

虽然这个交通标志的例子直观地说明了错误分类具有危险性，但它也适用于众多其他场景，例如，从用于制造的机器人设备到自动外科手术等场景。

对置信度(或不确定性)有一定的了解，是提高这些系统的鲁棒性并确保持续安全行为的重要一步。就停车标志而言，只要有一个“知其所不知”的模型，就能避免造成潜在的悲剧性后果。正如你将在本书后面看到的，贝叶斯深度学习方法使我们能够通过对其进行不确定性估计来检测对抗性输入。在我们的自动驾驶汽车示例中，可将其纳入逻辑中，这样，如果模型不确定，汽车就会安全地停下来，切换到手动模式，让驾驶员安全地掌握驾驶情况。这就是不确定性感知模型带来的智慧：能够设计出了解自身局限性的模型，从而在意外情况下更加鲁棒。

1.3.3　变化趋势

我们的最后一个例子将探讨处理随时间变化的数据时所面临的挑战——这是实际应用中的一个常见问题。我们要考虑的第一个问题通常被称为**数据集漂移(dataset shift)**或协变量漂移(covariate shift)，当模型在推理时遇到的数据相对于模型所训练的数据发生变化时，就会出现这种情况。这通常是由实际问题具有动态性，以及训练集(甚至非常大的训练集)很少能表征其所代表现象的全部变化所导致的。这方面的一个重要例子可以在论文 *Systematic Review of Approaches to Preserve Machine Learning Performance in the Presence of Temporal Dataset Shift in Clinical Medicine* 中找到，Lin Lawrence Guo 等人在该论文中强调了数据集漂移方面存在的问题

(https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8410238/)。他们的研究表明，在临床环境下应用的机器学习模型中，解决数据集漂移相关问题的文献相对较少。由于临床数据是动态的，因此这个问题很难解决。接下来看一个例子。

在该例子中，有一个经过训练的模型，可以根据患者的症状自动为其开药。病人向医生讲述呼吸道症状，医生使用模型开药。根据其获得的数据，它开出了抗生素处方。这种方法对许多病人都有效。但随着时间的推移，情况发生了变化：一种新的疾病在人群中流行起来。这种新疾病的症状与之前流行的细菌感染非常相似，但它是由病毒引起的。由于模型无法适应数据集漂移，它会继续推荐使用抗生素，而这不仅对病人没有帮助，还可能导致这些病人对抗生素产生抗药性。

为了使真实世界数据的这些漂移变得更鲁棒，模型需要对数据集的漂移非常敏感。做到这一点的方法之一是使用贝叶斯方法，该方法可提供不确定性估计。如果将这一方法应用到我们的自动开处方例子中，当模型能够提供不确定性估计时，就会对数据中的微小变化变得敏感。例如，与新的病毒感染相关的症状可能存在细微差别，如不同类型的咳嗽。这将导致与模型预测相关的不确定性上升，表明模型需要根据新数据进行更新。

与此相关的一个问题被称为**灾难性遗忘(catastrophic forrgetting)**，它是由模型适应数据变化引起的。基于我们所给的例子，这听起来是件好事：如果模型能够适应数据的变化，那么它始终处于更新状态，对吗？但遗憾的是，事情并非如此简单。当模型学习新数据时，如果遗忘了过去的数据，就会发生灾难性遗忘。

例如，我们开发了一种机器学习算法来识别欺诈性文件。它一开始可能效果很好，但欺诈者很快就会发现，过去用来欺骗自动文件验证的方法不再有效，于是他们又开发了新的方法。虽然其中有一些方法可以成功，但模型(利用其不确定性估计)注意到，它需要适应新的数据。模型更新了数据集，重点关注当前流行的攻击方法，并进行了更多的训练迭代。它再一次成功地挫败了欺诈者，但令模型设计者大吃一惊的是，该模型已经开始允许一些较老、不太复杂的攻击通过，而过去这些攻击对模型来说很容易识别。

在对新数据进行训练时，模型的参数发生了变化。由于更新后的数据集中没有足够的旧数据支持，因此模型丢失了输入(文档)与其分类(是否欺诈)之间的旧关联信息。

虽然这个例子使用了不确定性估计来解决数据集漂移的问题，但它还可以进一步利用不确定性估计来确保数据集的平衡。这可以使用**不确定性采样(uncertainty sampling)**等方法来实现，即从不确定性区域进行采样，确保用于训练模型的数据集能够获取到当前和过去数据中的所有可用信息。

1.4　核心主题

本书旨在为读者提供开发自己的贝叶斯深度学习解决方案所需的工具和知识。虽然相信读者对统计学习和深度学习的概念有一定的了解，但本书中我们仍将对这些基本概念进行复习。

在第 2 章中，我们将复习贝叶斯推理的一些关键概念，包括概率和模型不确定性估计。在第 3 章中，我们将介绍深度学习的重要关键方面，包括通过反向传播进行学习，以及各种流行

的神经网络。介绍了这些基础知识后，我们将在第 4 章开始探讨贝叶斯深度学习。在第 5 章和第 6 章中，我们将深入探讨贝叶斯深度学习；首先学习原理方法，然后了解更实用的贝叶斯神经网络近似方法。

在第 7 章中，我们将探讨贝叶斯深度学习的一些实际注意事项，以便帮助我们了解如何将这些方法应用于实际问题。到第 8 章，待你对贝叶斯深度学习的核心方法有了深刻理解之后，我们将通过一些实际例子来巩固这一知识点。最后，第 9 章将概述贝叶斯深度学习领域当前面临的挑战，让你了解该技术的发展方向。

在本书的大部分内容中，理论将与实践案例相结合，让你通过亲自实现这些方法来加深理解。为了学习这些编码示例，需要在 Python 环境中设置必要的先决条件。接下来将对此详细介绍。

1.5 设置工作环境

要完成本书的实践内容，你需要一个具备必要先决条件的 Python 3.9 环境。我们推荐使用 conda，它是专为科学计算应用而设计的 Python 软件包管理器。要安装 conda，只需登录 https://conda.io/projects/conda/en/latest/user-guide/install/index.html，然后按照操作系统的说明进行操作即可。

安装好 conda 后，就可以设置本书使用的 conda 环境了：

```
1 conda create -n bdl python=3.9
```

按回车键执行命令后，系统会询问你是否要继续安装所需的软件包；只需键入“y”，然后按回车键。

现在，可以输入以下命令激活环境：

```
1 conda activate bdl
```

现在你会看到 shell 提示符包含 bdl，表明 conda 环境已激活。现在你可以安装本书所需的如下关键库了：

- **NumPy**：Numerical Python 或 NumPy 是 Python 数值编程的核心软件包。你可能已经非常熟悉了。
- **SciPy**：SciPy 或 Scientific Python，为科学计算应用提供了基础软件包。由 SciPy、matplotlib、NumPy 和其他库组成的完整科学计算栈通常被称为 SciPy 栈。
- **scikit-learn**：Python 核心机器学习库。它建立在 SciPy 栈基础上，为许多流行的机器学习方法提供了易用的实现。它还为数据加载和处理提供了大量辅助类和函数，我们将在全书中使用这些类和函数。
- **TensorFlow**：TensorFlow 与 PyTorch 和 JAX 一样，都是流行的 Python 深度学习框架之一。它提供了开发深度学习模型所需的工具，并将为本书中介绍的许多编程示例奠定基础。

- **TensorFlow Probability**：TensorFlow Probability 基于 TensorFlow 开发，提供了处理概率神经网络所需的工具。我们将在许多贝叶斯神经网络示例中使用它和 TensorFlow。

要安装本书所需的全部依赖项列表，并激活 conda 环境，请输入以下内容：

```
1 conda install -c conda-forge scipy sklearn matplotlib seaborn
2 tensorflow tensorflow-probability
```

下面总结一下本章所学到的知识。

1.6　小结

在本章中，我们重温了深度学习的成功之处，重新认识了它的巨大潜力，以及它在当今技术中无处不在的地位。我们还探讨了其不足之处的一些关键实例：深度学习的一些失败案例，这些案例揭示了其造成灾难性后果的可能性。虽然贝叶斯推理无法消除这些风险，但它可以构建更鲁棒的机器学习系统，其中既有深度学习的灵活性，也有贝叶斯推理的谨慎性。

在第 2 章中，我们将深入探讨贝叶斯推理和概率的一些核心概念，为进入贝叶斯深度学习做准备。

第2章

贝叶斯推理基础

在使用**深度神经网络(Deep Neural Network，DNN)**进行贝叶斯推理之前，我们应该花一些时间了解其基本原理。在本章中，我们将探讨贝叶斯建模的核心概念，并介绍用于贝叶斯推理的一些常用方法。到本章结束时，你应该能够很好地理解我们要使用概率建模的原因，以及要在原则性良好或条件良好的方法中寻找哪些属性。

本章主要内容：

- 重温贝叶斯建模知识
- 通过采样进行贝叶斯推理
- 探讨高斯过程

2.1 重温贝叶斯建模知识

贝叶斯建模关注的是在给定一些先验假设和观察结果的条件下，了解事件发生的概率。先验假设描述了我们对事件的初始信念或假设。例如，假设有两个六面骰子，我们想预测两个骰子的点数之和为 5 的概率。首先我们需要知道有多少种可能的结果，因为每个骰子有 6 个面，所以可能的结果数是 6×6=36。为了算出掷出骰子的点数总和为 5 的可能性，我们需要算出有多少种数值组合的总和是 5，如图 2.1 所示。

	1	2	3	4	5	6
1	2	3	4	5	6	7
2	3	4	5	6	7	8
3	4	5	6	7	8	9
4	5	6	7	8	9	10
5	6	7	8	9	10	11
6	7	8	9	10	11	12

图 2.1　掷两个六面骰子时数值之和为 5 的所有数值组合示意图

从图 2.1 可以看到，有 4 种数值组合的总和是 5，因此两个骰子点数总和为 5 的概率是 4/36，即 1/9。我们称这种初始信念为**先验(prior)**。现在，如果将观察到的信息结合起来，会发生什么情况呢？假设我们知道其中一个骰子的点数值是 3，这就将下一个可能的数值个数缩减到 6，因为我们只有剩下的一个骰子可以掷，而在骰子点数总和为 5 的情况下，就需要这个数值为 2。图 2.2 给出了掷出第一个骰子后与第一个值的和为 5 的下一个值。

	1	2	3	4	5	6
1	2	3	4	5	6	7
2	3	4	5	6	7	8
3	4	5	6	7	8	9
4	5	6	7	8	9	10
5	6	7	8	9	10	11
6	7	8	9	10	11	12

图 2.2　掷出第一个骰子后与第一个值的和为 5 的下一个值

因为假设骰子是均匀的，所以骰子点数总和为 5 的概率现在是 1/6。这个概率称为**后验概率(posterior)**，是利用我们观察的信息得到的。贝叶斯统计法的核心是贝叶斯法则，我们用它来确定给定先验知识的后验概率。贝叶斯法则的定义如下：

$$P(A|B) = \frac{P(B|A) \times P(A)}{P(B)} \tag{2.1}$$

我们可以将 $P(A|B)$定义为 $P(d_1+d_2=5|d_1=3)$，其中 d_1 和 d_2 分别代表骰子 1 和 2。可以用前面的例子来说明这一点。从**似然(likelihood)**值，即分子左边的项开始，我们可以看到：

$$P(B|A) = P(d_1 = 3|d_1 + d_2 = 5) = \frac{1}{4} \tag{2.2}$$

我们可以通过观察网格来验证这一点。移至分子的第二部分，即先验值，可以看到：

$$P(A) = P(d_1 + d_2 = 5) = \frac{4}{36} = \frac{1}{9} \tag{2.3}$$

在分母上，我们有一个**归一化常数**(normalization constant)(也称为**边际似然值**(marginal likelihood))，简单地说就是：

$$P(B) = P(d_1 = 3) = \frac{1}{6} \tag{2.4}$$

利用贝叶斯定理将其综合起来，可以得出结论：

$$P(d_1 + d_2 = 5|d_1 = 3) = \frac{\frac{1}{4} \times \frac{1}{9}}{\frac{1}{6}} = \frac{1}{6} \tag{2.5}$$

这里得到的是在知道一个骰子的点数值的情况下结果为 5 的概率。不过，在本书中，我们经常提到的是**不确定性(uncertainties)**，而不是概率——以及使用深度神经网络获取不确定性估计的学习方法。这些方法属于更广泛的**不确定性量化(uncertainty quantification)**范畴，旨在量化机器学习模型预测中存在的不确定性。也就是说，我们想要预测 $P(\hat{y}|\theta)$，其中 $\hat{y}$ 是来自模型的预测，θ 代表模型的参数。

从基本概率论得知，概率介于 0 和 1 之间。概率越接近 1，事情发生的可能性就越大，我们可以把不确定性看作从 1 中减去概率。在本例中，点数总和为 5 的概率是 $P(d_1+d_2=5|d_1=3)=1/6=0.166$。因此，不确定性就是 1−1/6=5/6=0.833，也就是说，结果不是 5 的概率大于 80%。在本书的学习过程中，我们将了解不确定性的不同来源，以及不确定性如何帮助读者开发更鲁棒的深度学习系统。

继续以骰子为例，以更好地理解模型的不确定性估计。许多常见的机器学习模型都以**最大似然估计(Maximum Likelihood Estimation)**或 MLE 为基础。也就是说，它们希望预测最有可能得到的值：在训练过程中调优参数，以便在给定输入 x 的情况下产生最有可能得到的结果 $\hat{y}$。举个简单的例子，假设我们想预测给定 d_1 值情况下产生的 d_1+d_2 值。可以简单地将其定义为以 d_1 为条件的 d_1+d_2 的**期望值(expectation)**：

$$\hat{y} = \mathbb{E}[d_1 + d_2|d_1] \tag{2.6}$$

即求出 d_1+d_2 可能值的平均值。

设 $d_1=3$，则 d_1+d_2 的可能值为{4,5,6,7,8,9}(如图 2.2 所示)，从而得出平均值：

$$\mu = \frac{1}{6}\sum_{i=1}^{6} a_i = \frac{4+5+6+7+8+9}{6} = 6.5 \tag{2.7}$$

这是从简单线性模型中得到的值，例如，由以下公式定义的线性回归：

$$\hat{y} = \beta x + \xi \tag{2.8}$$

在这种情况下，我们的回归系数和偏差值为 $\beta=1$，$\xi=3.5$。如果将 d_1 的值改为 1，就会看到这个平均值变为了 4.5，即 $d_1+d_2|\,d_1=1$ 可能值的平均值，换句话说就是{2,3,4,5,6,7}。从这个角度来看，模型预测非常重要：虽然这个例子非常简单，但其原理也适用于更复杂的模型和数据。通常在机器学习模型中看到的值是期望值，也就是所谓的平均值。大家可能都知道，平均值通常被称为**第一统计矩(first statistical moment)**，**第二统计矩(second statistical moment)**则是**方差(variance)**，方差使我们能够量化不确定性。

这个简单例子的方差定义如下：

$$\sigma^2 = \frac{\sum_{i=1}^{6}(a_i-\mu)^2}{n-1} \tag{2.9}$$

大家应该对这些统计矩很熟悉，也应该知道这里的方差表示为**标准差(standard deviation)**σ 的平方。在我们的示例中，假设 d_2 是一个理想状态下的均匀骰子，方差将始终保持不变：$\sigma^2=2.917$。也就是说，给定任何 d_1 值，就知道 d_2 都有同样的可能值，因此不确定性不会改变。但是，如果我们有一个非理想状态下的骰子 d_2，它有 50%的概率落在 6 上，而有 10%的概率落在其他数字上，会出现什么情况呢？这种不均匀的概率分布会改变平均值和方差。可以看看如

何将其表示为一组可能的值(换句话说，一个完美的骰子样本)——$d_1+d_2|d_1=1$ 的可能值集现在变成了{2,3,4,5,6,7,7,7,7,7}。新模型现在的偏差为ξ=4.5，从而实现了我们的预测：

$$\hat{y} = 1 \times 1 + 4.5 = 5.5 \tag{2.10}$$

可以看到，由于骰子 d_2 值的基本概率发生了变化，因此期望值也随之增加。然而，这里的重要区别在于方差值的变化：

$$\sigma^2 = \frac{\sum_{i=1}^{10}(a_i - \mu)^2}{n-1} = 3.25 \tag{2.11}$$

我们的方差增大了。从本质上讲，方差给出了每个可能值与平均值之间距离的平均值，因此这并不奇怪：与未加权骰子相比，加权骰子的结果更有可能偏离平均值，因此我们的方差也会增加。总之，从不确定性的角度来看：结果离平均值越远的可能性越大，不确定性就越大。

这对我们解释机器学习模型(以及更普遍的统计模型)预测结果的方式具有重要影响。如果我们的预测是对平均值的近似，而不确定性量化了结果偏离平均值的可能性，那么不确定性就会告诉我们**模型预测错误的可能性有多大**。因此，模型的不确定性可以决定何时应该相信预测，何时应该更加谨慎。

这里给出的例子非常简单，但应该有助于你了解我们希望通过模型不确定性量化来实现的目标。在学习贝叶斯推理的一些基准方法时，我们将继续探讨这些概念，学习如何把这些概念应用于更复杂的现实问题。我们将从贝叶斯推理最基本的方法——采样——开始介绍。

2.2 通过采样进行贝叶斯推理

在实际应用中，我们不可能确切知道某个结果是什么，同样，也不可能观察到所有可能的结果。在这种情况下，我们需要根据所掌握的证据做出最佳估计。证据由**样本**组成，即对可能结果的观察。广义上讲，机器学习的目的是学习能从数据子集中很好地泛化的模型。贝叶斯机器学习的目的是在做到这一点的同时，提供与模型预测相关的不确定性估计。在本节中，我们将了解如何使用采样来实现这一目标，同时将了解为什么采样可能不是最明智的方法。

2.2.1 近似分布

从根本来讲，采样就是关于近似分布的。假设我们想知道纽约人的身高分布，可以出去测量每个人的身高，但这需要测量 840 万人的身高！虽然这能给我们提供最准确的答案，但也是一种非常不切实际的方法。

相反，我们可以从总体中采样。一种基本的采样方法就是**蒙特卡洛采样(Monte Carlo sampling)**，即使用随机采样来提供数据，并从中近似分布。例如，给定一个纽约居民数据库，我们可以随机选择一个居民样本，并以此来近似所有居民的身高分布。对于随机采样和任何采样来说，近似值的准确率都取决于样本体的大小。我们希望得到的是一个**统计意义上**的子样本，这样才能对所得的近似值有置信。

为了更好地理解这一点，我们将从截断正态分布中生成 100,000 个数据点来模拟这个问题，

以近似 10 万人的身高分布。假设从统计样本中随机抽取 10 个样本。图 2.3 显示了我们的分布(右侧)与真实分布(左侧)的对比。

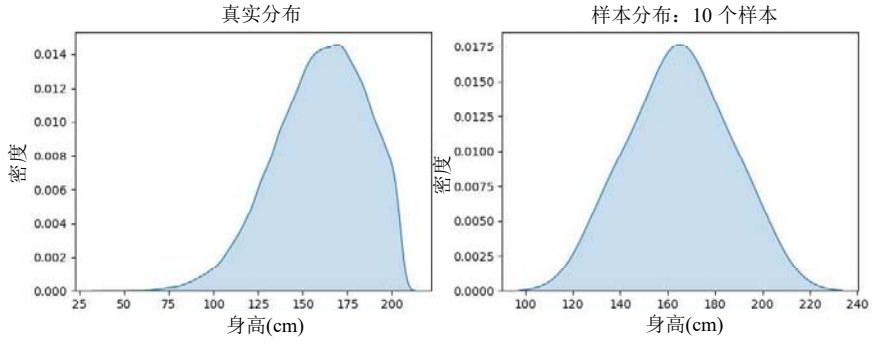


图 2.3　真实分布(左)与样本分布(右)的对比图

可以看到，这并不能很好地反映真实分布：这里看到的是更接近于三角形分布，而不是截断正态分布。如果仅根据这个分布来推理人群的身高，就会得出一些不准确的结论，比如缺少 200 厘米以上的截断点和分布左侧尾部的截断点。

我们可以通过增加样本量来获得更好的分布——尝试抽取 100 个样本。图 2.4 显示了样本量为 100 时真实分布与样本分布的对比。

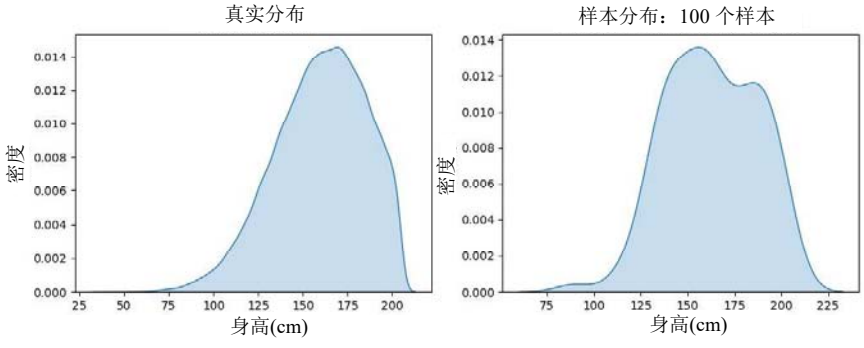


图 2.4　真实分布(左)与样本分布(右)的对比图

情况开始好转：我们可以看到左侧的一些尾部截断以及向 200 厘米的截断。然而，这个样本从某些区域采样的次数多于其他区域，从而导致了错误的表示：我们的平均值被拉低了，而且可以看到两个明显的峰值，而不是真实分布中的单个峰值。因此，我们把样本量再增加一个数量级，增加到 1,000 个样本。

这看起来好多了：虽然样本量只有真实统计人口的百分之一，但我们现在看到的分布与真实分布已非常接近了。这个例子说明，通过随机采样，可以用一个小得多的观察集来近似真实分布。但是，这个样本池仍然必须包含足够多的信息，才能很好地近似真实分布；若样本量太少，我们的子集在统计意义上就会不够充分，那么，对基本分布的近似度就会很低。图 2.5 显示了样本量为 1,000 时真实分布与样本分布的对比。

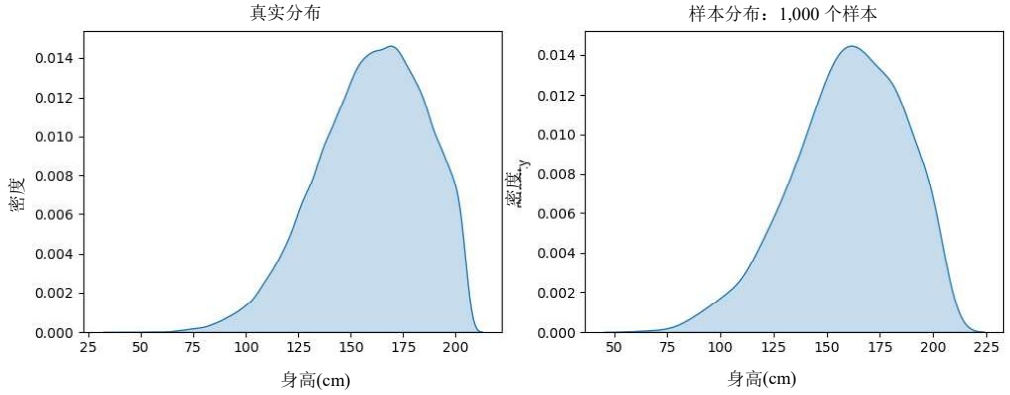


图 2.5　真实分布(左)与样本分布(右)的对比图

但简单的随机采样并不是近似分布的最实用方法。为此，我们转向使用**概率推理(probabilistic inference)**。在给定模型的情况下，概率推理提供了一种找到最能描述数据的模型参数的方法。为此，首先需要定义模型类型为先验模型。在我们的例子中，将使用截断高斯分布：这里的想法是，根据我们的直觉，假设人们的身高呈正态分布是合理的，但是很少有人的身高超过 6 英尺 5 英寸(≈2 米)。因此，我们将指定一个截断的高斯分布，其上限为 205 厘米，即略高于 6 英尺 5 英寸。由于是高斯分布，即$\mathcal{N}(\mu,\sigma)$，因此模型参数是$\theta=\{\mu,\sigma\}$。另外，还有一个约束条件，即我们的分布上限为b=205。

这就引出了一类基本算法：**马尔可夫链蒙特卡洛(Markov Chain Monte Carlo)**(或称**MCMC** 方法)。与简单的随机采样一样，这些方法使我们能够建立真实的基本分布图，但它们是按序列建立的，即每个样本都依赖于之前的样本。这种序列依赖性被称为**马尔可夫特性(Markov property)**，也被称为马尔可夫链。这种序列方法考虑了样本之间的概率相关性，使我们能够更好地近似概率密度。

MCMC 通过序列随机采样实现了这一点。就像我们熟悉的随机采样一样，MCMC 从分布中随机采样。但是，与简单的随机采样不同，MCMC 考虑的是成对的样本：之前的样本x_{t-1}和当前的样本x_t。对于每一对样本，我们都有一些标准来确定是否保留样本(这取决于 MCMC 的具体类型)。如果新值符合这个标准，比如说x_t“优于”我们之前使用的值x_{t-1}，那么样本就会被添加到链中，成为下一轮的x_t。如果样本不符合标准，就在下一轮使用当前的x_t。这样反复进行(通常是大量的)迭代，最终应该能得到一个很好的分布近似值。

这就是一种高效的采样方法，它能够近似于分布的真实参数。下面看看如何将其应用到我们的身高分布示例中。在 10 个样本中使用 MCMC，可以得到如图 2.6 所示的近似值。

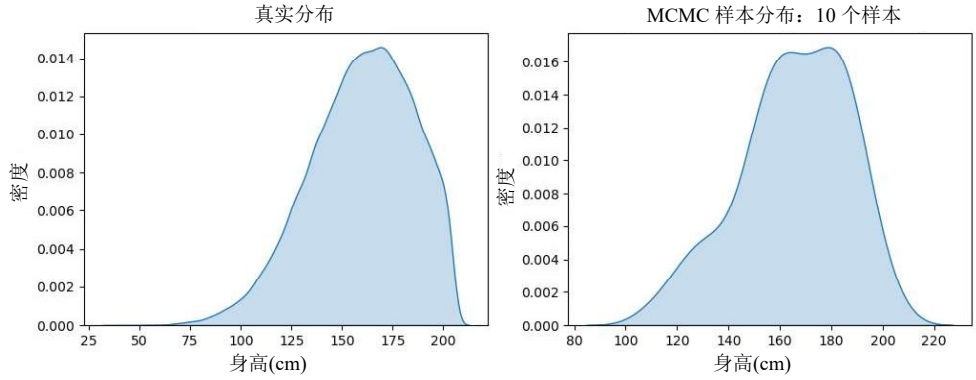


图 2.6　真实分布(左)与通过 MCMC 得出的近似分布(右)的对比图

10 个样本的结果还不错，当然比我们通过简单随机采样得出的三角形分布要好得多。下面看看 100 个样本的情况，得到的近似值如图 2.7 所示。

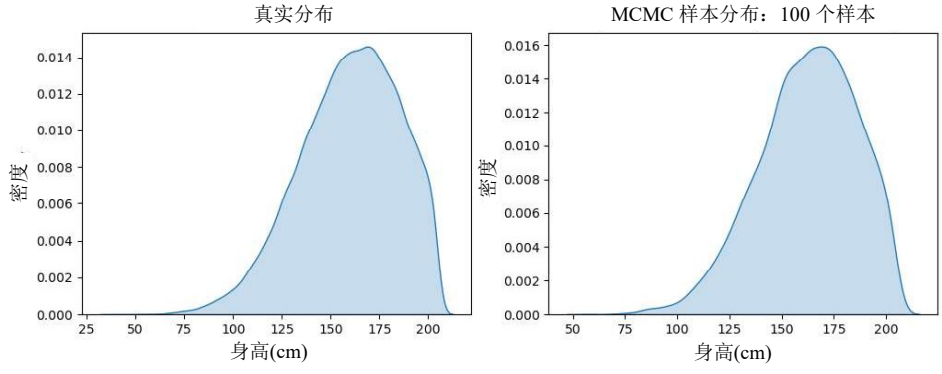


图 2.7　真实分布(左)与通过 MCMC 得出的近似分布(右)的对比图

这看起来相当不错——事实上，我们用 100 个 MCMC 样本得到的分布近似值比用 1,000 个简单随机样本得到的分布近似值还要准确。如果继续增加样本数量，就能得到越来越接近真实分布的近似值。但是，这个简单的例子并不能完全体现 MCMC 的优势：MCMC 的真正优势来自能够近似高维分布，并使其成为近似各种领域中难以解决的高维积分的宝贵技术。

在本书中，我们感兴趣的是如何估计机器学习模型参数的概率分布，这使我们能够估计与预测相关的不确定性。下一节通过将采样应用于贝叶斯线性回归，来了解如何切实做到这一点。

2.2.2　利用贝叶斯线性回归实现概率推理

在典型的线性回归中，我们希望使用线性函数 $f(x)$从输入 x 预测输出 $\hat{y}$，即 $\hat{y}=\beta x+\xi$。在贝叶斯线性回归中，我们采用概率方法，引入另一个参数 σ^2，这样回归公式就变成了：

$$\hat{y} = \mathcal{N}(x\beta + \xi, \sigma^2) \tag{2.12}$$

也就是说，$\hat{y}$ 遵循高斯分布。

这里，我们看到了熟悉的偏差项 ξ 和回归系数 β，并引入了方差参数 σ^2。为了拟合模型，需要定义这些参数的先验——就像我们在上一节的 MCMC 例子中所做的那样。将这些先验定义为：

$$\xi \approx \mathcal{N}(0,1) \tag{2.13}$$

$$\beta \approx \mathcal{N}(0,1) \tag{2.14}$$

$$\sigma^2 \approx |\mathcal{N}(0,1)| \tag{2.15}$$

注意，式 2.15 表示高斯分布的半正态分布(即零平均值高斯分布的正半部分，因为标准差不能为负)。我们将模型参数称为 $\theta=\beta, \xi, \sigma^2$，我们将使用采样来找到在给定数据条件下最大化这些参数似然性的值，换句话说，就是求出在给定数据 D 条件下参数的条件概率：$P(\theta|D)$。

我们可以使用多种 MCMC 采样方法来找到模型参数。一种常见的方法是使用 **Metropolis-Hastings** 算法。Metropolis-Hastings 算法尤其适用于从难以处理的分布中采样。它通过使用与真实分布成正比但不完全相等的提议分布 $Q(\theta'|\theta)$来实现。举例来说，这意味着如果在真实分布中，某个值 x_1 可能是另一个值 x_2 的两倍，那么我们的提议分布也将是如此。因为我们感兴趣的是观测值的概率，所以不需要知道真实分布中的确切值是多少，而只需要知道，在比例层面上，我们的提议分布等同于真实分布。

以下是贝叶斯线性回归中 Metropolis-Hastings 算法的关键步骤。

首先，我们根据每个参数的先验，从参数空间中抽取任意点 θ 进行初始化。使用以第一组参数 θ 为中心的高斯分布，选择一个新点 θ'。然后，在每次迭代 $t \in T$ 时，执行以下操作：

(1) 计算验收标准，定义为：

$$\alpha = \frac{P(\theta'|D)}{P(\theta|D)} \tag{2.16}$$

(2) 从均匀分布中生成一个随机数 $\epsilon \in [0, 1]$。如果 $\epsilon \leqslant \alpha$，则接受新的候选参数——将其添加到链中，使 $\theta=\theta'$。如果 $\epsilon > \alpha$，则保留当前的 θ 并绘制新值。

这个验收标准意味着，如果新的参数集比上一组参数集的似然性更高，就会看到 $\alpha>1$，在这种情况下，$\alpha>\epsilon$。这意味着，当根据更有可能给出数据的参数进行采样时，我们始终会接受这些参数。另一方面，如果 $\alpha<1$，我们有可能会拒绝这些参数，但也有可能会接受它们——这就允许我们探索似然性较低的区域。

Metropolis-Hastings 的这些机制产生的样本可以用来计算后验分布的高质量近似值。实际上，Metropolis-Hastings(以及更普遍的 MCMC 方法)需要一个“磨合”阶段，即用于摆脱低密度区域的初始采样阶段，在进行任意初始化的情况下通常会有这样一个阶段。

我们将其应用于一个简单的问题：为函数 $y=x^2+5+\eta$ 生成一些数据，其中 η 是一个噪声参数，其分布条件为 $\eta \approx \mathcal{N}(0, 5)$。使用 Metropolis-Hastings 拟合我们的贝叶斯线性回归器，就可以通过从函数中采样的点(用叉号表示)得到如图 2.8 所示的拟合结果。

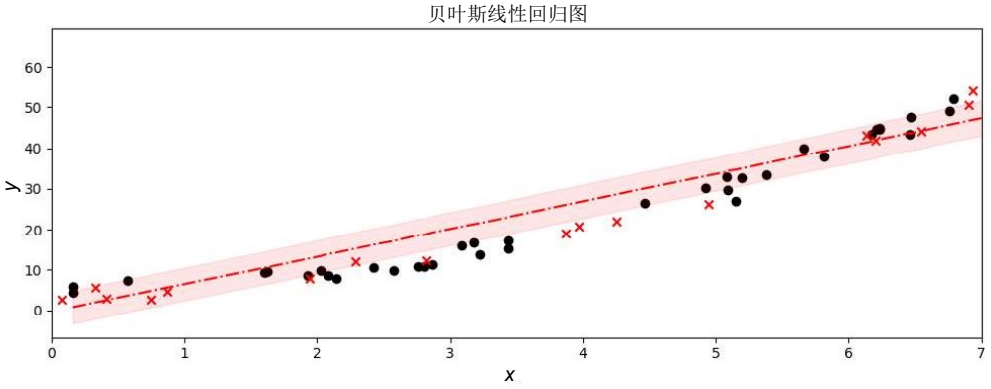


图 2.8　对生成的低方差数据进行贝叶斯线性回归

可以看到，我们的模型与标准线性回归的数据拟合方式相同。然而，与标准线性回归不同的是，我们的模型会产生预测不确定性：这由阴影区域表示。这种预测的不确定性可以了解基本数据的变化程度；使得这个模型比标准线性回归有用得多，因为现在我们可以了解数据的分布以及总体趋势。如果生成新数据并再次拟合，这次通过修改噪声分布使 $\eta \approx \mathcal{N}(0, 20)$来增加数据的分布范围，这样就可以看到数据的变化情况，如图 2.9 所示。

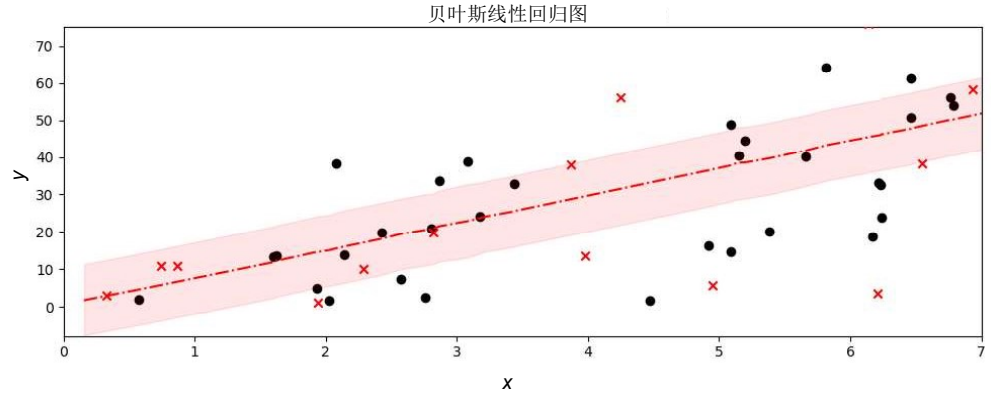


图 2.9　对生成的高方差数据进行贝叶斯线性回归

可以看到，预测的不确定性与数据的扩散成正比增加。这是不确定性感知方法的一个重要特性：当不确定性较小时，我们知道预测与数据拟合得很好；而当不确定性较大时，我们知道需谨慎看待预测，因为这表明模型与这一区域拟合得不是特别好。你将在下一节中看到一个更好的例子，它将继续说明数据较多或较少的区域如何影响模型的不确定性估计值。

这里可以看到我们的预测与数据拟合得很好。此外，还可以看到，σ^2 随不同区域的数据可用性而变化。这是一个有关**校准良好的不确定性**(也称为**高质量不确定性**)的很好例子。它是一个非常重要的概念，指的是在预测不准确的区域，其不确定性也很高。如果我们对预测不准确的置信度较高，或者对预测的准确性非常不确定，那么不确定性估计值的**校准就很差**。由于采样校准良好，因此采样通常被用作不确定性量化的基准。

遗憾的是，虽然采样在很多应用中都很有效，但由于每个参数都需要获取很多样本，这意味着对于高维度的参数来说，采样很快就会变得计算量过大而让人却步。例如，如果我们想对具有复杂、非线性关系的参数进行采样(如对神经网络的权重进行采样)，采样就不实用了。尽管如此，它在某些情况下仍然有用，稍后你将看到各种贝叶斯深度学习方法是如何利用采样的。

在下一节中，我们将探讨高斯过程，这是贝叶斯推理的另一种基本方法，它没有采样那样的计算开销。

2.3　探讨高斯过程

如上一节所述，采样很容易会因其计算成本过高而让人却步。为了解决这个问题，我们可以使用专为产生不确定性估计值而设计的机器学习模型(其中的黄金标准就是**高斯过程**)。

高斯过程(Gaussian Process，GP)已成为一种主要的概率机器学习模型，被广泛应用于从药理学到机器人学的各个领域。它的成功在很大程度上归功于它能够以一种良好的方式对其预测结果进行高质量的不确定性估计。那么，高斯过程具体是指什么呢？

从本质上讲，高斯过程是一种函数分布。为了便于理解，下面举一个典型的机器学习用例。我们想要学习某个函数 $f(x)$，它能将一系列输入 x 映射到一系列输出 y 上，这样我们就能通过 $\hat{y}=f(x)$来近似输出。在看到任何数据之前，我们对基本函数一无所知；有无数种潜在函数，如图 2.10 所示。

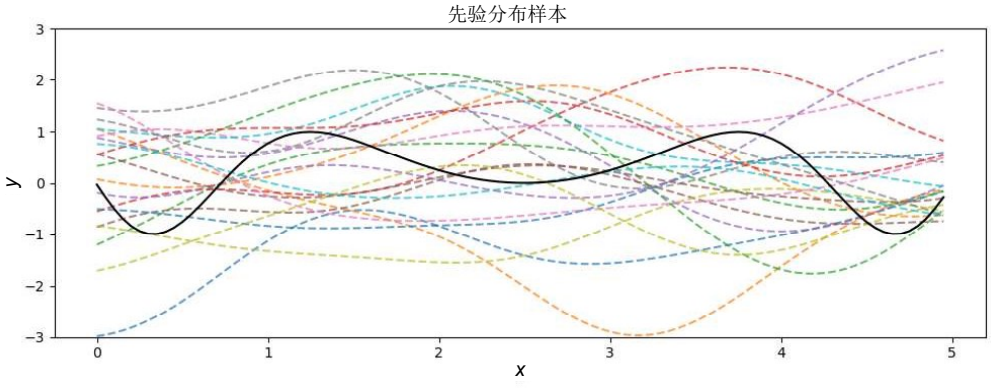


图 2.10　在看到数据之前潜在函数的空间示意图

这里，黑实线是我们希望学习的真实函数，而虚线则是在有数据(本例中没有数据)的情况下可能出现的函数。一旦我们观察到一些数据，就会发现潜在函数的数量变得更加有限，如图 2.11 所示。

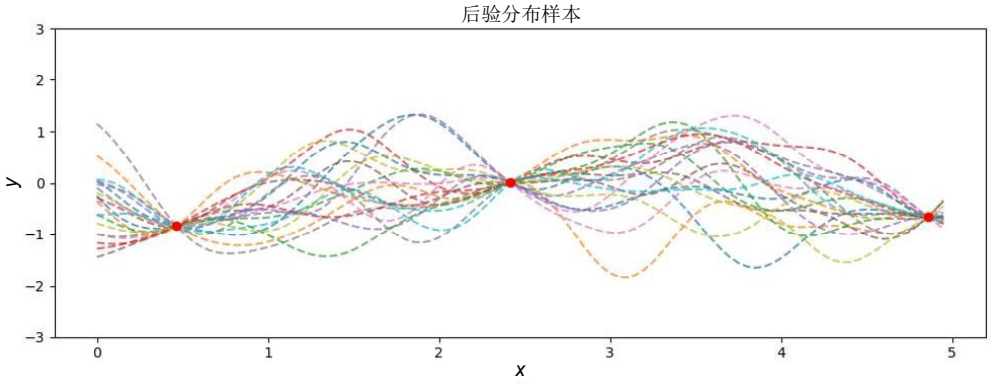


图 2.11　看到一些数据后的潜在函数空间示意图

在此可以看到，潜在函数都经过了我们观察到的数据点，但在这些数据点之外，函数却有一系列截然不同的取值范围。在简单的线性模型中，我们并不关心这些可能值的偏差：我们更乐于从一个数据点插值到另一个数据点，如图 2.12 所示。

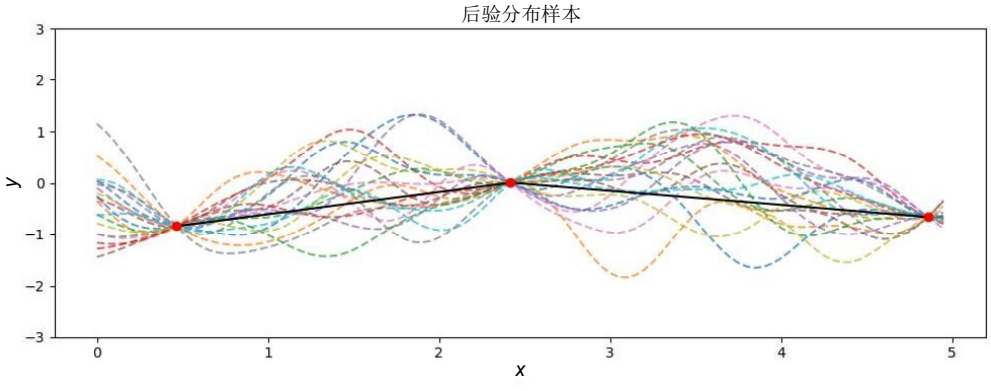


图 2.12　对观察数据进行线性插值的示意图

但是，这种内插法可能会导致预测非常不准确，而且无法计算与模型预测相关的不确定性程度。我们在没有数据点的区域所看到的偏差，正是我们想要用高斯过程来获取的。当我们的函数有多种可能取值时，就存在不确定性；而且通过获取不确定性的程度，就能估计出这些区域可能具有的变化。

从形式上看，高斯过程可以定义为一个函数：

$$f(x) \approx GP(m(x), k(x, x')) \tag{2.17}$$

这里，$m(x)$只是给定数据点 x 可能得到的函数值的平均值：

$$m(x) = \mathbb{E}[f(x)] \tag{2.18}$$

下一个项 $k(x, x')$是协方差函数或核函数。这是高斯过程的基本组成部分，因为它定义了我们对数据中不同点之间关系的建模方式。高斯过程使用平均值和协方差函数对可能的函数空间进行建模，从而得出预测结果及其相关的不确定性。

既然我们已经介绍了一些高层次的概念，那么下面再深入一点，了解高斯过程究竟是如何对可能的函数空间进行建模，从而估计不确定性的。为此，我们需要了解高斯过程先验。

2.3.1 用核定义先验信念

GP 核描述的是我们对数据的先验信念，因此你经常会看到它们被称为高斯过程先验。与式 2.3 中的先验告诉我们两个骰子掷出结果的概率一样，高斯过程先验也告诉我们有关从数据中预期的关系的一些重要信息。

虽然有从数据中推理先验的高级方法，但这不在本书的讨论范围之内。我们将重点讨论高斯过程的传统用法，即利用我们正在处理的数据来选择先验。

在你遇到的文献和任何实现中，你会发现高斯过程先验通常被称为**核(kernel)**或**协方差函数(covariance function)**(就像我们这里所说的一样)。这三个术语都可以互换，但为了与其他术语保持一致，我们今后将把它称为核。核仅仅是计算两个数据点之间距离的一种方法，用 $k(x, x')$ 表示，其中 x 和 x' 是数据点，$k(\ )$ 表示核函数。虽然核可以有多种形式，但在大部分高斯过程应用中只会用到少数基本核。

最常见的核可能是**平方指数(squared exponential)**或**径向基函数(Radial Basis Function，RBF)**核。这种核的形式如下：

$$k(x, x') = \sigma^2 \exp -\frac{(x - x')^2}{2l^2} \tag{2.19}$$

这里有几个常见的核参数：l 和 σ^2。输出方差参数 σ^2 只是一个缩放因子，用于控制函数与其平均值的距离。长度缩放参数 l 控制函数的平滑度，换句话说，就是函数在特定维度上的变化程度。该参数既可以是一个适用于所有输入维度的标量，也可以是一个为每个输入维度设置不同标量值的向量。后者通常是通过**自动相关性判断(Automatic Relevance Determination，ARD)**来实现的，它可以识别输入空间中的相关值。

高斯过程通过基于核的协方差矩阵进行预测，其本质是将新数据点与之前观察到的数据点进行比较。然而，与所有机器学习模型一样，高斯过程也需要经过训练，这就是长度缩放的作用所在。长度缩放构成了高斯过程的参数，通过训练，高斯过程可以学习长度缩放的最佳值，这通常是通过非线性优化器实现的，如 **Broyden-Fletcher-Goldfarb-Shanno(BFGS)**优化器。许多优化器都可供使用，包括你可能熟悉的深度学习优化器，如随机梯度下降及其变体。

下面来看看不同的核如何影响高斯过程预测。我们将从一个简单的例子开始讲解——一个简单的正弦波函数，如图 2.13 所示。

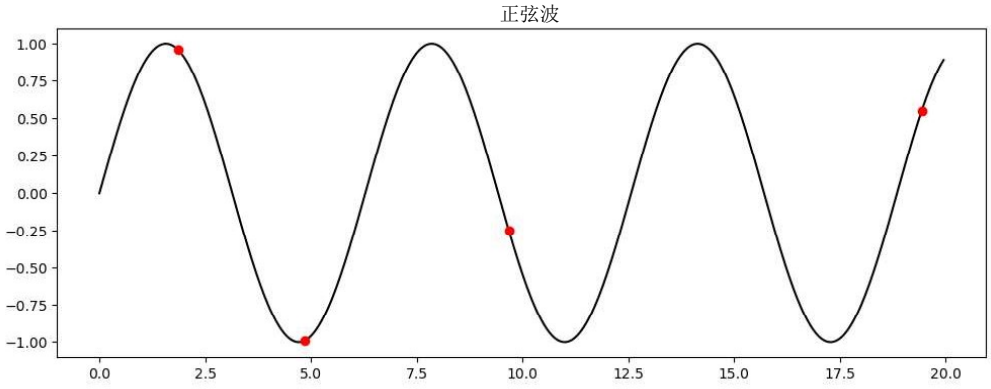


图 2.13　带有四个采样点的正弦波图

我们可以看到这里的函数图示，以及从该函数中采样的一些数据点。现在，对数据拟合一个具有周期核的高斯过程。周期核的定义如下：

$$k_{per}(x, x') = \sigma^2 \exp\left(\frac{2\sin^2(\pi|x - x'|/p)}{l^2}\right) \tag{2.20}$$

这里，我们看到了一个新参数 p，即周期函数的周期。设置 p=1，并在前面的例子中应用具有周期核的高斯过程，会得到如图 2.14 所示的结果。

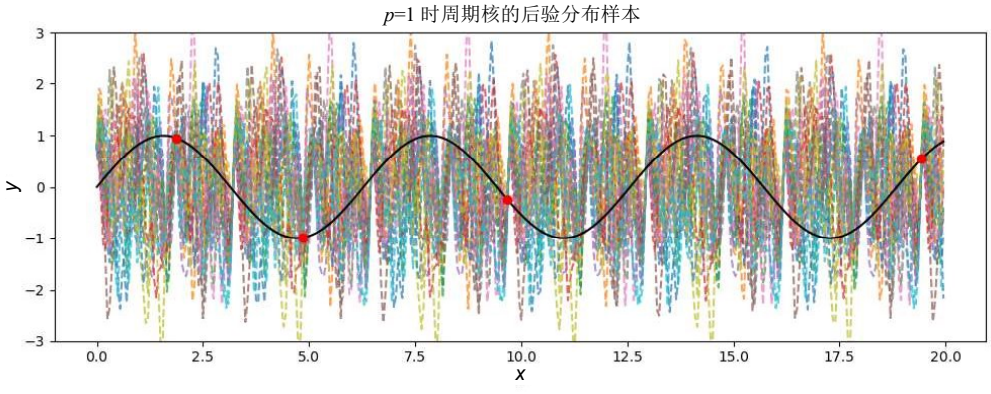


图 2.14　p=1 时周期核的后验预测图

这些样本看起来有噪声，但你应该可以看到后验产生的函数具有明显的周期性。造成噪声的原因有两个：缺乏数据和不准确的先验。如果数据有限，我们可以尝试通过改进先验来解决问题。在这种情况下，可以利用函数周期性的知识，通过设置 p=6 来改进我们的先验值，如图 2.15 所示。

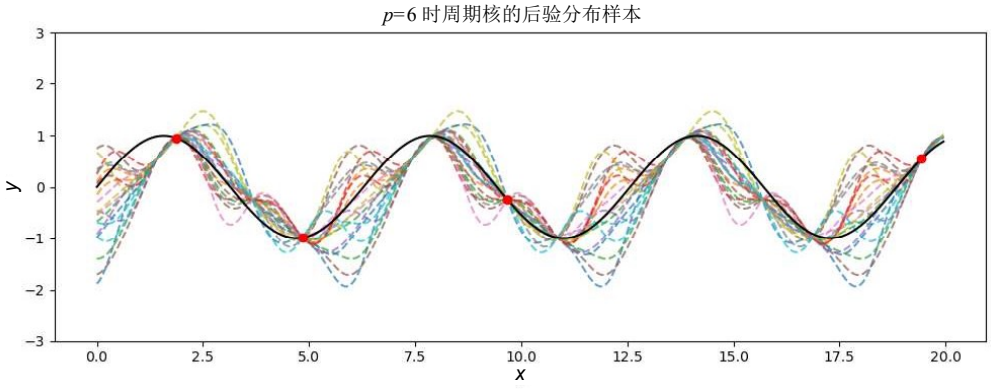


图 2.15　p=6 时周期核的后验预测图

可以看到，这与数据非常拟合：在数据较少的区域，仍然存在不确定性，但后验的周期性现在看起来是合理的。之所以能做到这一点，是因为我们使用了一个有信息量的先验；也就是说，这个先验包含了能全面描述数据的信息。这个先验由两个关键部分组成：

- 周期核
- 关于函数周期性的知识

如果我们将高斯过程修改为使用 RBF 核，就会发现这一点有多么重要，如图 2.16 所示。

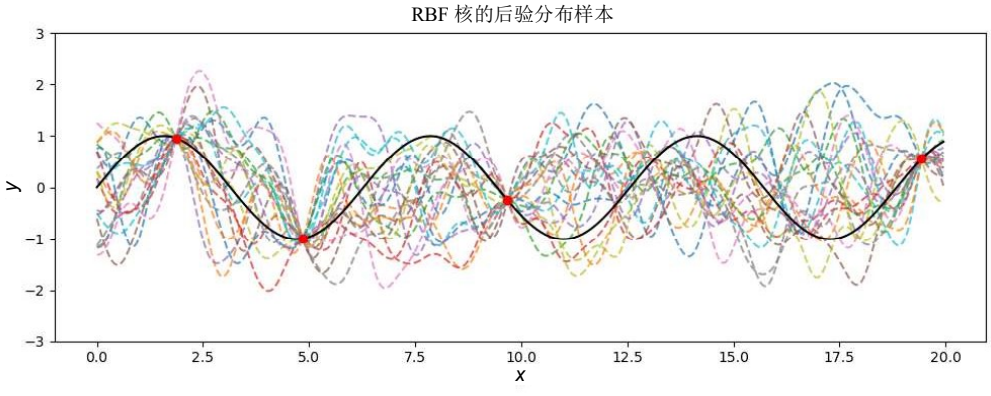


图 2.16　RBF 核的后验预测图

使用 RBF 核后，我们发现情况又变得非常混乱：由于数据有限且先验较差，无法对可能的函数空间进行适当限制，以拟合真实函数。在理想情况下，我们可以使用更合适的先验来解决这个问题，如图 2.15 所示，但这并不总是可行的。另一种解决方案是对更多数据进行采样。我们继续使用 RBF 核，从函数中抽取 10 个数据点，然后重新训练高斯过程，如图 2.17 所示。

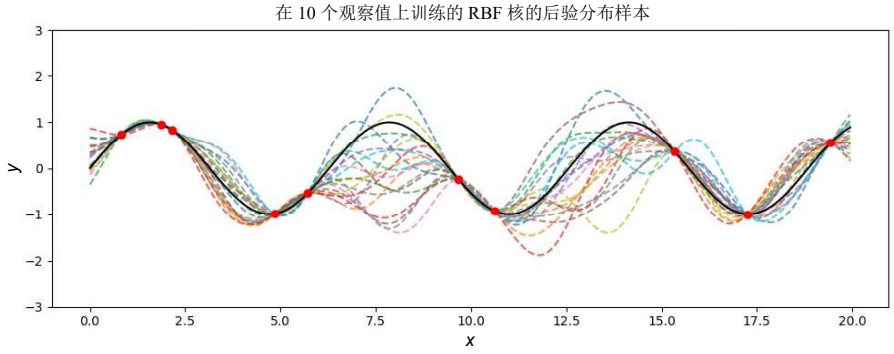


图 2.17　在 10 个观察值上训练的 RBF 核的后验预测图

结果看起来好多了——但如果我们有更多的数据和信息量更大的先验呢？图 2.18 显示了 p=6 时，以 10 个观察值为基础进行训练的后验预测图。

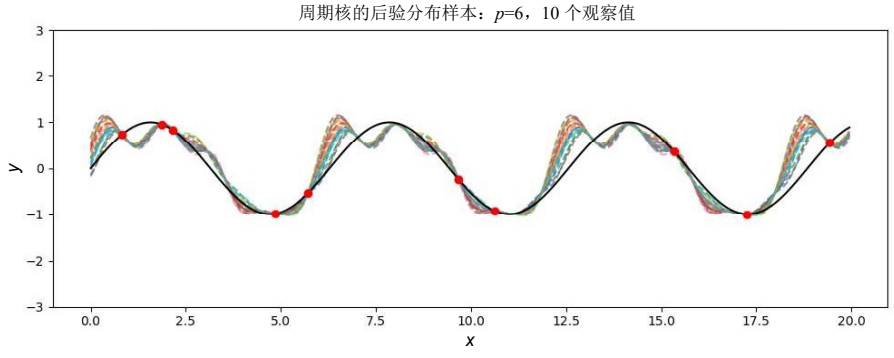


图 2.18　p=6 时周期核的后验预测图，以 10 个观察值为基础进行训练

现在的后验非常拟合我们的真实函数。因为我们的数据有限，所以仍有一些不确定的区域，但不确定性相对较小。

现在我们已经看到了一些核心原则，回到图 2.10～图 2.12 中所示的例子。下面快速回顾一下目标函数、后验样本以及我们之前看到的线性插值，如图 2.19 所示。

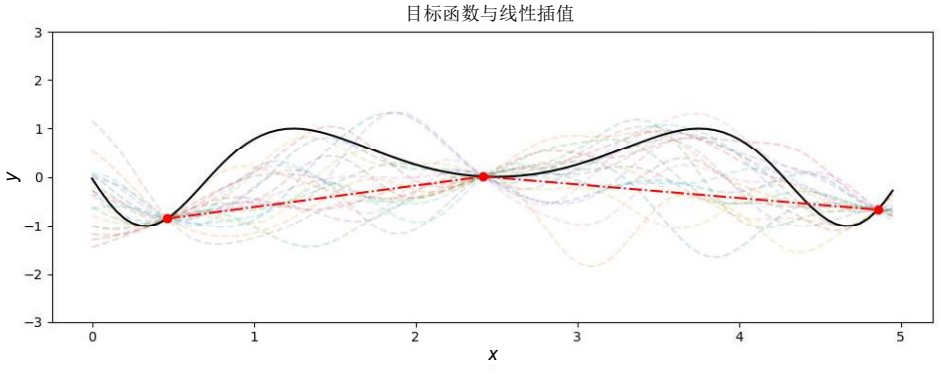


图 2.19　线性插值与真实函数的差异图

既然我们已经对高斯过程如何影响预测后验有了一定的了解，就不难发现线性插值与高斯过程的效果相差甚远。为了更清楚地说明这一点，我们来看看在给定三个样本的情况下，高斯过程对这个函数的预测结果，如图 2.20 所示。

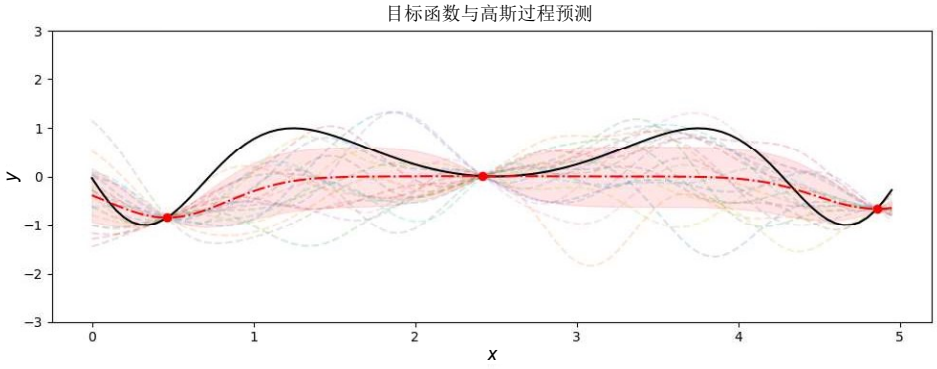


图 2.20　高斯过程预测与真实函数之间差异的示意图

这里，虚线是高斯过程预测的平均值(μ)，阴影部分是与这些预测相关的不确定性——平均值周围的标准差(σ)。将图 2.20 与图 2.19 进行对比。起初，两者之间的差别似乎很微妙，但我们可以清楚地看到，这不再是简单的线性插值：高斯过程的预测值被“拉”向我们的实际函数值。与前面的正弦波例子一样，高斯过程预测的行为受到两个关键因素的影响：先验(或核)和数据。

但图 2.20 中还显示了另一个关键细节：高斯过程预测的不确定性。我们看到，与许多典型的机器学习模型不同，高斯过程会给出与其预测相关的不确定性。这意味着我们可以更好地决定如何处理模型的预测结果，掌握这些信息将有助于确保我们的系统更加鲁棒。例如，如果不确定性太高，可以退回到手动系统。我们甚至可以跟踪那些预测不确定性较高的数据点，从而不断完善我们的模型。

就像前面的例子一样，可以通过增加一些观察数据来了解这种改进对预测的影响，如图 2.21 所示。

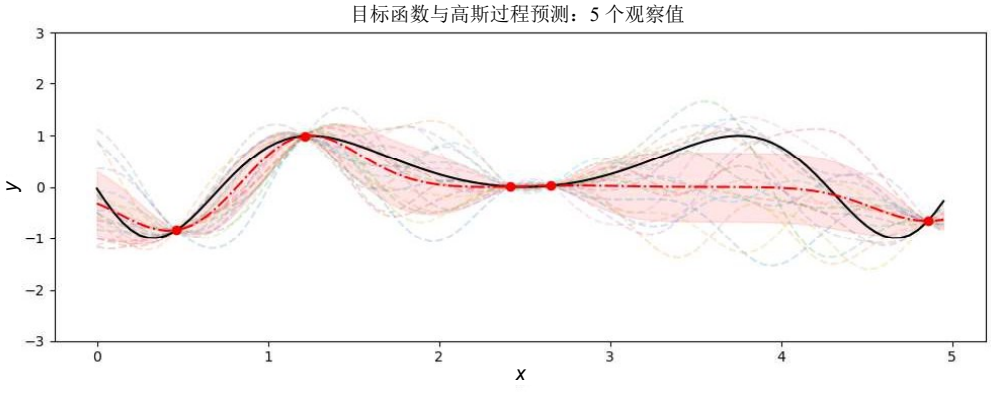


图 2.21　基于 5 个观察数据训练的高斯过程预测与真实函数之间的差异图

图 2.21 展示了不确定性在不同观察值区域的变化情况。可以看到，在 x=3 和 x=4 之间，不确定性相当高。这是可以理解的，因为我们也可以看到高斯过程的平均预测值与真实函数值有很大偏差。相反，如果观察 x=0.5 和 x=2 之间的区域，就会发现高斯过程预测相当接近真实函数，而且我们的模型对这些预测的置信度也更高，这一点可以从该区域较小的不确定性区间中看出。

我们在这里看到的是一个有关**校准良好的不确定性**(也称为**高质量不确定性**)的很好例子。它是一个非常重要的概念，指的是，在预测不准确的区域，其不确定性也很高。如果我们对预测不准确区域的置信度较高，或者对预测准确的区域非常不确定，那么不确定性估计值的**校准就很差**。

高斯过程是一种**理论性很强**的方法——这意味着它有坚实的数学基础，因此有很强的理论保证。这些保证之一就是它们经过了很好的校准，这也是高斯过程如此受欢迎的原因：如果使用高斯过程，就知道可以信赖它们的不确定性估计。

但遗憾的是，高斯过程并非没有局限性。我们将在下一节进一步了解这些局限性。

2.3.2　高斯过程的局限性

考虑到高斯过程原则性很强，并且能够产生高质量的不确定性估计，你可以认为它们是完美的不确定性感知机器学习模型。但在一些关键情况下，高斯过程却显得力不从心：

- 高维数据
- 海量数据
- 高度复杂的数据

前两点在很大程度上归因于高斯过程无法很好地得到扩展。要理解这一点，只需看看高斯过程的训练和推理过程。虽然本书对此无法详细介绍，但关键点在于掌握高斯过程训练所需的矩阵运算。

在训练过程中，需要对 $D\times D$ 矩阵求逆，其中 D 是数据的维度。正因为如此，高斯训练过程很快就会变得难以计算。通过使用 Cholesky 分解法(而不是直接对矩阵求逆)可以在一定程度上缓解这一问题。除了计算效率更高，Cholesky 分解在数值上也更加稳定。遗憾的是，Cholesky 分解也有其弱点：其计算复杂度为 $O(n^3)$。这意味着，随着数据集规模的增加，高斯过程的训练成本也会越来越高。

但受影响的不仅仅是训练：由于我们需要在推理时计算新数据点与所有观察数据点之间的协方差，因此高斯过程在推理时的计算复杂度为 $O(n^2)$。

除了计算成本，高斯过程所占用的内存也不小：因为我们需要存储协方差矩阵 $\boldsymbol{K}$，所以高斯过程的内存复杂度为 $O(n^2)$。因此，在包含大型数据集的情况下，即使我们拥有训练高斯过程所需的计算资源，但由于高斯过程对内存有较高的要求，在实际应用中使用高斯过程可能并不现实。

最后一点与数据的复杂性有关。大家可能都知道，我们也将在第 3 章“深度学习基础”中谈到，深度神经网络的主要优势之一就是能够通过非线性转换层来处理复杂的高维数据。虽然高斯过程功能强大，但它们也是相对简单的模型，无法像深度神经网络那样学习各种强大的特征

表征。

所有这些因素都意味着，虽然高斯过程是相对低维的数据和相当小的数据集的绝佳选择，但对于我们在机器学习中面临的许多复杂问题，它们并不实用。因此，我们转向了使用贝叶斯深度学习方法：这种方法具有深度学习的灵活性和可扩展性，同时还能产生模型的不确定性估计值。

2.4 小结

在本章中，我们介绍了与贝叶斯推理相关的一些基本概念和方法。首先，回顾了贝叶斯定理和概率论的基本原理，使我们能够理解不确定性的概念，以及如何将其应用于机器学习模型的预测。接下来，介绍了采样和一种重要的算法：马尔可夫链蒙特卡洛方法(简称 MCMC)。最后，介绍了高斯过程，并说明了校准良好的不确定性这一重要概念。这些关键主题将为后面的内容打下必要的基础，不过，我们鼓励大家研究推荐的阅读材料，以便更全面地理解本章介绍的主题。

在第 3 章中，我们将了解深度神经网络如何在过去十年中改变了机器学习的格局，探索深度学习带来的巨大优势，以及开发贝叶斯深度学习方法背后的动机。

2.5 延伸阅读

人们目前正在探索各种技术，以提高高斯过程的灵活性和可扩展性，例如，深度高斯过程或稀疏高斯过程。以下资源探讨了其中一些主题，并对本章涉及的内容进行了更深入的探讨：

- Martin 编写的 *Bayesian Analysis with Python*：本书全面涵盖了统计建模和概率编程的核心主题，包括各种采样方法的实际演练，以及与高斯过程和贝叶斯分析有关的其他各种核心技术的精彩概述。
- Rasmussen 和 Williams 编写的 *Gaussian Processes for Machine Learning*，这本书被认为是关于高斯过程的权威著作，对高斯过程的基础理论提供了非常详细的解释。对于任何认真研究贝叶斯推理的人来说，这都是一本重要文献。