计算机科学丛书

CAMBRIDGE

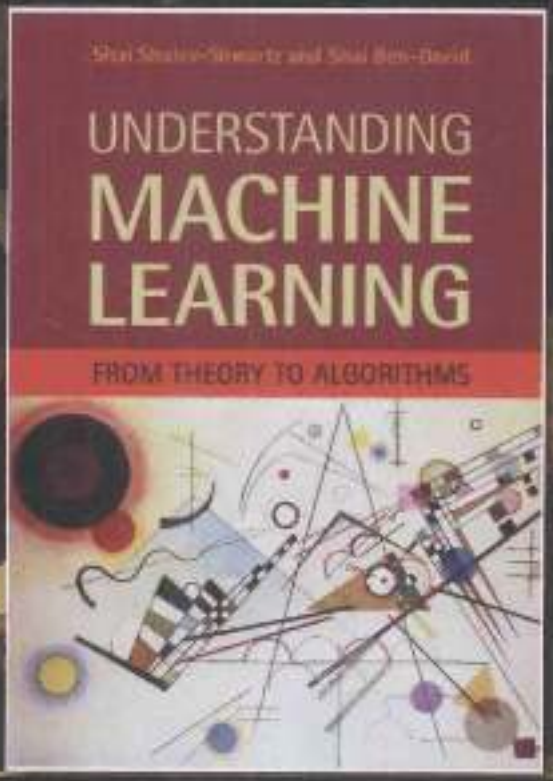
深入理解机器学习

从原理到算法

[以] 沙伊·沙莱夫-施瓦茨（Shai Shalev-Shwartz）
[加] 沙伊·本-戴维（Shai Ben-David） 著
张文生 等译

Understanding Machine Learning

From Theory to Algorithms



机械工业出版社
China Machine Press

深入理解机器学习 从原理到算法

Understanding Machine Learning From Theory to Algorithms

这部精心撰写的教材不仅讲引了严谨的理论，还涵盖了机器学习的实际应用。这是一本优秀的教材，适合所有想要了解如何在数据中寻找结构的读者。

——伯恩哈德·史科夫（Bernhard Schölkopf），马克斯·普朗克智能系统研究所

这本教材非常必要，对于想要建立机器学习的数学基础的读者来说，它同时兼顾深度和广度，内容严谨、直观而简洁。本书提供了不需的算法和分析技巧，经典而基础，还指出了最前沿的研究方向。机器学习是一项重要而迷人的领域，对于任何对其数学及计算基础感兴趣的人来说，这都是一本理想的书。

——艾弗瑞·布卢姆（Avrim Blum），卡内基-梅隆大学

机器学习是计算机科学中发展最快的领域之一，实际应用广泛。这本教材的目标是从理论角度提供机器学习的入门知识和相关算法范式。本书全面地介绍了机器学习背后的基本思想和理论依据，以及将这些理论转化为实际算法的数学推导。在介绍了机器学习的基本内容后，本书还覆盖了此前的教材中一系列从未涉及过的内容，其中包括对学习的计算复杂度、凸性和稳定性的概念的讨论，以及重要的算法范式的介绍（包括随机梯度下降、神经元网络以及结构化输出学习）。同时，本书引入了最新的理论概念，包括PAC-贝叶斯方法和压缩界。本书为高等院校本科高年级和研究生入门阶段而设计，不仅计算机、电子工程、数学统计专业学生能轻松理解机器学习的基础知识和算法，其他专业的读者也能读懂。

作者简介

沙伊·沙莱夫-施瓦茨（Shai Shalev-Shwartz） 以色列希伯来大学计算机及工程学院副教授，还在Mobileye公司研究自动驾驶，2009年之前他在芝加哥的丰田技术研究所工作。他的研究方向是机器学习算法。

沙伊·本-戴维（Shai Ben-David） 加拿大滑铁卢大学计算机科学学院教授，先后在以色列理工学院、澳大利亚国立大学和康奈尔大学任教。

CAMBRIDGE
UNIVERSITY PRESS
www.cambridge.org

书店网址：(010) 88379604
实店网址：(010) 88378991 88381076
图书网址：(010) 88378294 88379649 68995259

本店网址：www.hzbook.com
网上商店：www.china-pub.com
数字阅读：www.ertongbook.com.cn

ISBN 978-7-111-54302-2

9 787111 543022 >

定价：79.00元

计 算 机 科 学 丛 书

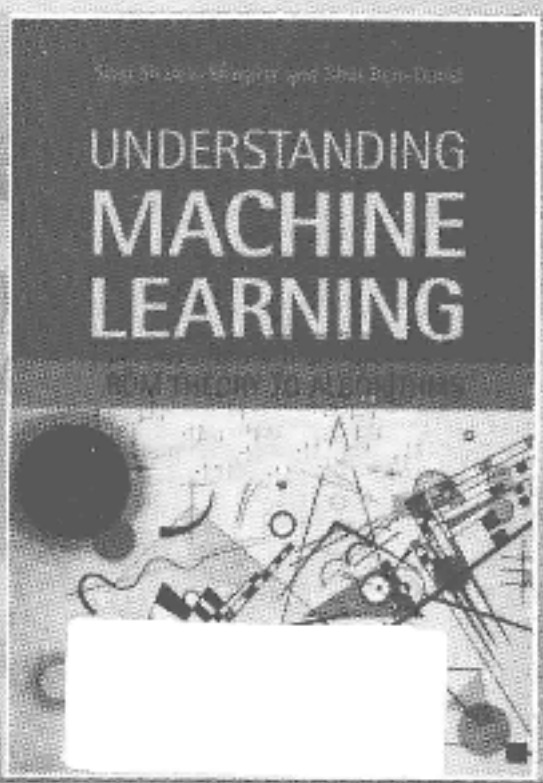
深入理解机器学习

从原理到算法

[以] 沙伊·沙莱夫-施瓦茨（Shai Shalev-Shwartz）
[加] 沙伊·本-戴维（Shai Ben-David） 著
张文生 等译

Understanding Machine Learning

From Theory to Algorithms



机械工业出版社
China Machine Press

图书在版编目（CIP）数据

深入理解机器学习：从原理到算法 / （以）沙伊·沙莱夫-施瓦茨（Shai Shalev-Shwartz）、（以）沙伊·本-戴维（Shai Ben-David）著；张文生等译．—北京：机械工业出版社，2016.7（2016.11 重印）
（计算机科学丛书）
书名原文：Understanding Machine Learning: From Theory to Algorithms

ISBN 978-7-111-54302-2

I. 深… II. 沙… 沙… 张… III. 机器学习 IV. TP181

中国版本图书馆 CIP 数据核字（2016）第 157549 号

本书版权登记号：图字：01-2016-3281

This is a Chinese Simplified edition of the following title published by Cambridge University Press: Shai Shalev-Shwartz and Shai Ben-David, Understanding Machine Learning: From Theory to Algorithms (ISBN 978-1-107-05713-5).

© Shai Shalev-Shwartz and Shai Ben-David 2014.

This Chinese Simplified edition for the People's Republic of China (excluding Hong Kong, Macau and Taiwan) is published by arrangement with the Press Syndicate of the University of Cambridge, Cambridge, United Kingdom.

© Cambridge University Press and China Machine Press in 2016.

This Chinese Simplified edition is authorised for sale in the People's Republic of China (excluding Hong Kong, Macau and Taiwan) only. Unauthorised export of this simplified Chinese is a violation of the Copyright Act. No part of this publication may be reproduced or distributed by any means, or stored in a database or retrieval system, without the prior written permission of Cambridge University Press and China Machine Press.

本书纸版由剑桥大学出版社与机械工业出版社合作出版。未经出版者预先书面许可，不得以任何方式复制或抄袭本书的任何部分。

此版本仅限在中华人民共和国境内（不包括香港、澳门特别行政区及台湾地区）销售。

本书讲述了机器学习的统一的计算理论和实用方法，讨论了学习的计算复杂性，介绍了凸优化方法、PAC-贝叶斯方法、维数等概念，介绍了一些重要的算法范式，包括随机梯度下降、神经网络及结构化输出。

全书讲解全面透彻，适合有一定基础的高年级本科生和研究生学习，也适合作为IT行业从事数据分析和挖掘工作的人员及研究人员的参考阅读。

出版发行：机械工业出版社（北京市西城区百万庄大街22号 邮政编码：100037）

责任编辑：关 敏　　责任校对：董纪丽

印　　刷：三河市宏图印务有限公司　　版　　次：2016年11月第1版第2次印刷

开　　本：185mm×260mm 1/16　　印　　张：20.25

书　　号：ISBN 978-7-111-54302-2　　定　　价：79.00元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

客服热线：（010）88379426 88361066　　建书热线：（010）88379804

购书热线：（010）68326294 88379649 68995259　　投稿热线：hzjsj@hzbook.com

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问：北京大成律师事务所 韩光/邹晓东

文艺复兴以来，源远流长的科学精神和逐步形成的学术规范，使西方国家在自然科学的各个领域取得了举世瞩目的优势；也正是这种的优势，使美国在信息技术发展的六十多年间名家辈出、独领风骚。在商业化的进程中，美国的产业界与教育界越来越紧密地结合，计算机学科中的许多泰山北斗同时身处科研和教学的最前线，由此而产生的经典科学著作，不仅奠定了研究的基础，还揭示了学术的源泉，既遵循学术规范，又自有学者个性，其价值并不会随时间流逝而减退。

近年，在全球信息化大潮的推动下，我国的计算机产业发展迅猛，对专业人才的需求日益迫切。这对计算机教育界和出版界都既是机遇，也是挑战；而专业教材的建设在教育战略上显得举足轻重。在我国信息技术发展时间较短的现状下，美国等发达国家在其计算机科学发展的几十年间积淀和发展的经典教材仍有许多值得借鉴之处。因此，引进一批国外优秀计算机教材将对我国计算机教育事业的发展起到积极的推动作用，也是与世界接轨、建设真正的世界一流大学的必由之路。

机械工业出版社华章公司较早意识到"出版要为教育服务"。自1998年开始，我们就将工作重点放在了翻译、移译国外优秀教材上，经过多年的不懈努力，我们与Pearson，McGraw-Hill，Elsevier，MIT，John Wiley & Sons，Cengage等世界著名出版公司建立了良好的合作关系，从他们现有的数百种教材中甄选出Andrew S. Tanenbaum，Bjarne Stroustrup，Brian W. Kernighan，Dennis Ritchie，Jim Gray，Afred V. Aho，John E. Hopcroft，Jeffrey D. Ullman，Abraham Silberschatz，William Stallings，Donald E. Knuth，John L. Hennessy，Larry L. Peterson等大师名家的一批经典作品，以"计算机科学丛书"为总称出版，供读者学习、研究及珍藏。大理石纹理的封面，也正体现了这套丛书的品位和格调。

"计算机科学丛书"的出版工作得到了国内外学者的鼎力相助，国内的专家不仅提供了中肯的选题指导，还不辞劳苦地担任了翻译和审校的工作；而原书的作者也相当关注其作品在中国的传播，有的还专门为其书的中译本作序。迄今，"计算机科学丛书"已经出版了近两百个品种，这些书籍在读者中树立了良好的口碑，并被许多高校采用为正式教材和参考书籍。其影印版"经典原版书库"作为原版引进被越来越多实施双语教学的学校所采用。

权威的作者、经典的教材、一流的译者、严格的审校、精细的编辑，这些因素使我们的图书有了质量的保证。随着计算机科学与技术专业学科建设的不断完善和教材改革的逐渐深化，教育界对国外计算机教材的需求和应用都将步入一个新的阶段，我们的目标是尽善尽美，而反馈的意见正是我们达到这一终极目标的重要帮助。华章公司欢迎老师和读者对我们的工作提出建议或给予指正，我们的联系方法如下：

华章网站：www.hzbook.com
电子邮件：hzjsj@hzbook.com
联系电话：(010)88379604
联系地址：北京市西城区百万庄南街1号
邮政编码：100037

华章教育

华章科技图书出版中心

以色列希伯来大学副教授 Shai Shalev-Shwartz 和加拿大滑铁卢大学教授 Shai Ben-David 的专著《Understanding Machine Learning: From Theory to Algorithms》是机器学习领域一部具有里程碑意义的著作。

近几年，机器学习是人工智能研究领域中最活跃的分支之一，已成为信息科学领域解决实际问题的重要方法，它的应用已遍及人工智能的各个应用领域。机器学习又是一个多学科的交叉领域，涉及数学、自动化、计算机科学、应用心理学、生物学和神经生理学等。这种学科交叉融合带来的良性互动，无疑促进了包括机器学习在内的诸学科的发展与繁荣。

本书内容十分丰富，作者以前所未有的广度和深度，介绍了目前机器学习中重要的理论和关键的算法。本书没有陷入"科普"式的堆砌材料的写作方式，由于作者是该领域的权威专家，因此在介绍各种理论和算法时，时刻不忘将不同理论、算法的对比与作者自身的研究成果传授给读者，使读者不至于对如此丰富的理论和算法无所适从。另外，特别值得指出的是，本书第一部分非常有特色，也是非常重要的一部分。这部分内容从更高的观点和更深的层次探讨机器学习的许多理论基础，引入对指导理论研究和实际应用都至关重要的概率近似正确(Probably Approximately Correct，PAC)学习理论。该理论旨在回答由机器学习得到的结果到底有多高的可信度与推广能力，从某种意义上来说，只有懂得了该部分，才可能透彻地理解和更好地运用其他章节的内容。国内关于 PAC 学习的资料非常少，在翻译过程中团队成员碰到了很大的困难，我们人工智能与机器学习研究团队为此进行了多方论证并多次召开专题讨论会。

本书主要面向人工智能、机器学习、模式识别、数据挖掘、计算机应用、生物信息学、数学和统计学等领域的研究生和相关领域的科技人员。翻译出版中译本的目的，是希望能为国内广大从事相关研究的学者和研究生提供一本全面、系统、权威的教科书和参考书。如果能做到这一点，译者将感到十分欣慰。

必须说明的是，本书的翻译是中国科学院自动化研究所人工智能与机器学习研究团队集体努力的结果，团队的成员杨当冰、王秋明、蒋琼娟、薛伟、魏波、李思民、张似衡、曾凡霞、于廷照、王巍、李涛、杨叶辉、胡文瑜、张志忠、唐永强、陈东杰、何泽文、张英华、李恒、李硕等参与了本书的翻译工作。李思民老师参与了全书的审校与修正。感谢机械工业出版社华章分社的大力协助，倘若没有他们的热情支持，本书的中译版难以如此迅速地与大家见面。另外，本书的翻译得到了国家自然科学基金委重点项目和面上项目(61472423、U1135005、61432008、61532006、61305028、61402481 等)的资助，特此感谢。

在翻译过程中，我们力求准确地反映原著内容，同时保留原著的风格。但由于译者水平有限，书中难免有不妥之处，恳请读者批评指正。

最后，谨把本书的中译版献给我的博士生导师王珏研究员！王珏老师生前对机器学习理论、算法和应用非常关注，对于PAC可学习理论也有着独到而深刻的理解，他启发并引领了我们研究团队对机器学习理论和算法的研究工作，使我们终身受益。

中国科学院自动化研究所
张文生
2016年4月于北京

"机器学习"旨在从数据中自动识别有意义的模式。过去几十年中，机器学习成为一项常用工具，几乎所有需要从大量数据集合中提取信息的任务都在使用它。我们身边的许多技术都以机器学习为基础：搜索引擎学习在带给我们最佳的搜索结果的同时，植入可以盈利的广告；屏蔽软件学习过滤垃圾邮件；用于保护信用卡业务的软件学习识别欺诈。数码相机学习人脸识别，智能电话上的个人智能助手学习识别语音命令。汽车配备了用机器学习算法搭建的交通事故预警系统。同时机器学习还被广泛应用于各个科学领域，例如生物信息学，医药以及天文学等。

这些应用领域的一个共同特点在于，与相对传统的计算机应用相比，所需识别的模式更复杂。在这些情景中，对于任务应该如何执行，人类程序员无法提供明确的、细节优化的具体指令。以智能生物为例，我们人类的许多技能都是通过从经验中学习而取得并逐步提高的(而非遵从别人给我们的具体指令)。机器学习工具关注的正是赋予程序"学习"和适应不同情况的能力。

本书的第一个目标是，提供一个准确而简明易懂的导论，介绍机器学习的基本概念：什么是学习？机器怎样学习？学习某概念时，如何量化所需资源？学习始终都是可能的吗？我们如何知道学习过程是成功或失败？

本书的第二个目标是，为机器学习提供几个关键的算法。我们提供的算法，一方面已经成功投入实际应用，另一方面广泛地考虑到不同的学习技术。此外，我们特别将注意力放到了大规模学习(即俗称的"大数据")上，因为近几年来，世界越来越"数字化"，需要学习的数据总量也在急剧增加。所以在许多应用中，数据量是充足的，而计算时间是主要瓶颈。因此，学习某一概念时，我们会明确量化数据量和计算时间这两个数值。

本书分为四部分。第一部分对于"学习"的基础性问题给出初步而准确的定义。我们会介绍 Valiant 提出的"概率近似正确(PAC)"可学习模型的通用形式，它将是对"何为学习"这一问题的第一个有力回答。我们还会介绍"经验风险最小化(ERM)""结构风险最小化(SRM)"和"最小描述长度(MDL)"这几个学习规则，展现"机器是如何学习的"。我们量化使用 ERM、SRM 和 MDL 规则学习时所需的数据总量，并用"没有免费的午餐"定理说明，什么情况下学习可能会失败。此外，我们还探讨了学习需要多少计算时间。本书第二部分介绍多种算法。对于一些算法，我们先说明其主要学习原则，再介绍该算法是如何依据其原则运作的。前两部分将重点放在 PAC 模型上，第三部分将范围扩展到更广、更丰富的学习模型。最后，第四部分讨论最前沿的理论。

我们尽量让本书能够自成一体，不过我们假设读者熟悉概率论、线性代数、数学分析和算法设计的基本概念。前三部分为计算机科学，工程学、数学和统计学研究生一年级学生设计，具有相关背景的本科生也可以使用。高级章节适用于想要对理论有更深入理解的研究者。

本书以“机器学习入门”课程为蓝本，这门课程由 Shai Shalev-Shwartz 和 Shai Ben-David 分别在希伯来大学和滑铁卢大学讲授。本书的初稿由 Shai Shalev-Shwartz 在 2010 至 2013 年间在希伯来大学所开课程的教案整理而成。感谢 2010 年的助教 Ohad Shamir 和 2011 至 2013 年的助教 Alon Gonen 的帮助，他们为课堂准备了一些教案以及许多课后练习。特别感谢 Alon 在全书编写过程中所做出的贡献，此外他还撰写了一册习题答案。

我们由衷地感谢 Dana Rubinstein 的辛勤工作。Dana 从科学的角度校对了书稿，对原稿进行了编辑，将它从章节教案的形式转换成连贯流畅的文本。

特别感谢 Amit Daniely，他仔细阅读了本书的高级部分，并撰写了多分类可学习性的章节。我们还要感谢耶路撒冷的一个阅读俱乐部的成员们，他们认真阅读了原稿的每一页，并提出了建设性的意见。他们是：Maya Alroy，Yossi Arjevani，Aharon Birnbaum，Alon Cohen，Alon Gonen，Roi Livni，Ofer Meshi，Dan Rosenbaum，Dana Rubinstein，Shahar Somin，Alon Vinnikov 和 Yoav Wald，还要感谢 Gal Elidan，Amir Globerson，Nika Haghtalab，Shie Mannor，Amnon Shashua，Nati Srebro 和 Ruth Urner 参与的有益讨论。

目 录

Understanding Machine Learning: From Theory to Algorithms

译者的话
译者序
前言
致谢

第1章 引论 …… 1
1.1 什么是学习 …… 1
1.2 什么时候需要机器学习 …… 2
1.3 学习的种类 …… 3
1.4 与其他领域的关系 …… 4
1.5 如何阅读本书 …… 4
1.6 符号 …… 6

第一部分 理论基础

第2章 简易入门 …… 10
2.1 一般模型——统计学习理论框架 …… 10
2.2 经验风险最小化 …… 11
2.3 考虑归纳偏置的经验风险最小化 …… 12
2.4 练习 …… 15

第3章 一般学习模型 …… 17
3.1 PAC学习理论 …… 17
3.2 更常见的学习模型 …… 18
3.2.1 放宽可实现假设——不可知PAC学习 …… 18
3.2.2 学习问题建模 …… 19
3.3 小结 …… 21
3.4 文献评注 …… 21
3.5 练习 …… 21

第4章 学习过程的一致收敛性 …… 24
4.1 一致收敛是可学习的充分条件 …… 24
4.2 有限类是不可知PAC可学习的 …… 25
4.3 小结 …… 26
4.4 文献评注 …… 27
4.5 练习 …… 27

第5章 偏差与复杂性权衡 …… 28
5.1 "没有免费的午餐"定理 …… 28
5.2 误差分解 …… 31
5.3 小结 …… 31
5.4 文献评注 …… 32
5.5 练习 …… 32

第6章 VC维 …… 33
6.1 无限的类也可学习 …… 33
6.2 VC维概述 …… 34
6.3 实例 …… 35
6.3.1 阈值函数 …… 35
6.3.2 区间 …… 35
6.3.3 平行于轴的矩形 …… 35
6.3.4 有限类 …… 36
6.3.5 VC维与参数个数 …… 36
6.4 PAC学习的基本定理 …… 36
6.5 定理6.7的证明 …… 37
6.5.1 Sauer引理及生长函数 …… 37
6.5.2 有小的有效规模的类的一致收敛性 …… 39
6.6 小结 …… 40
6.7 文献评注 …… 41
6.8 练习 …… 41

第7章 不一致可学习 …… 44
7.1 不一致可学习概述 …… 44
7.2 结构风险最小化 …… 45
7.3 最小描述长度和奥卡姆剃刀 …… 48
7.4 可学习的其他概念——一致收敛性 …… 50
7.5 探讨不同的可学习概念 …… 51

7.6 小结 …… 53
7.7 文献评注 …… 53
7.8 练习 …… 54

第 8 章 学习的运行时间 …… 56

8.1 机器学习的计算复杂度 …… 56
8.2 ERM 规则的实现 …… 58
8.2.1 有限集 …… 58
8.2.2 轴对齐矩形 …… 59
8.2.3 布尔合取式 …… 59
8.2.4 学习三项析取范式 …… 60
8.3 高效学习，而不通过合适的 ERM …… 60
8.4 学习的难度* …… 61
8.5 小结 …… 62
8.6 文献评注 …… 62
8.7 练习 …… 62

第二部分 从理论到算法

第 9 章 线性预测 …… 66

9.1 半空间 …… 66
9.1.1 半空间类线性规则 …… 67
9.1.2 半空间感知器 …… 68
9.1.3 半空间的 VC 维 …… 69
9.2 线性回归 …… 70
9.2.1 最小平方 …… 70
9.2.2 多项式线性回归 …… 71
9.3 逻辑斯谛回归 …… 72
9.4 小结 …… 73
9.5 文献评注 …… 73
9.6 练习 …… 73

第 10 章 boosting …… 75

10.1 弱可学习 …… 75
10.2 AdaBoost …… 78
10.3 基础假设类的线性组合 …… 80
10.4 AdaBoost 用于人脸识别 …… 82
10.5 小结 …… 83
10.6 文献评注 …… 83
10.7 练习 …… 84

第 11 章 模型选择与验证 …… 85

11.1 用结构风险最小化进行模型选择 …… 85
11.2 验证法 …… 86
11.2.1 留出的样本集 …… 86
11.2.2 模型选择的验证法 …… 87
11.2.3 模型选择曲线 …… 88
11.2.4 k 折交叉验证 …… 88
11.2.5 训练-验证-测试拆分 …… 89
11.3 如果学习失败了应该做什么 …… 89
11.4 小结 …… 92
11.5 练习 …… 92

第 12 章 凸学习问题 …… 93

12.1 凸性、利普希茨性和光滑性 …… 93
12.1.1 凸性 …… 93
12.1.2 利普希茨性 …… 96
12.1.3 光滑性 …… 97
12.2 凸学习问题概述 …… 98
12.2.1 凸学习问题的可学习性 …… 99
12.2.2 凸利普希茨/光滑有界学习问题 …… 100
12.3 替代损失函数 …… 101
12.4 小结 …… 102
12.5 文献评注 …… 102
12.6 练习 …… 102

第 13 章 正则化和稳定性 …… 104

13.1 正则损失最小化 …… 104
13.2 稳定规则不会过拟合 …… 105
13.3 Tikhonov 正则化作为稳定剂 …… 106
13.3.1 利普希茨损失 …… 108
13.3.2 光滑和非负损失 …… 108
13.4 控制适合与稳定性的权衡 …… 109
13.5 小结 …… 111
13.6 文献评注 …… 111
13.7 练习 …… 111

第 14 章 随机梯度下降 …… 114

14.1 梯度下降法 …… 114

14.2 次梯度 …… 116
14.2.1 计算次梯度 …… 117
14.2.2 利普希茨函数的次梯度 …… 118
14.2.3 次梯度下降 …… 118
14.3 随机梯度下降 …… 118
14.4 SGD的变型 …… 120
14.4.1 增加一个投影步 …… 120
14.4.2 变步长 …… 121
14.4.3 其他平均技巧 …… 121
14.4.4 强凸函数* …… 121
14.5 用SGD进行学习 …… 123
14.5.1 SGD求解风险极小化 …… 123
14.5.2 SGD求解凸光滑学习问题的分析 …… 124
14.5.3 SGD求解正则化损失极小化 …… 125
14.6 小结 …… 125
14.7 文献评注 …… 125
14.8 练习 …… 126

第15章 支持向量机 …… 127
15.1 间隔与硬SVM …… 127
15.1.1 另次情况 …… 129
15.1.2 硬SVM的样本复杂度 …… 129
15.2 软SVM与范数正则化 …… 130
15.2.1 软SVM的样本复杂度 …… 131
15.2.2 间隔、基于范数的界与维度 …… 131
15.2.3 斜坡损失* …… 132
15.3 最优化条件与"支持向量"* …… 133
15.4 对偶* …… 133
15.5 用随机梯度下降法实现软SVM …… 134
15.6 小结 …… 135
15.7 文献评注 …… 135
15.8 练习 …… 135

第16章 核方法 …… 136
16.1 特征空间映射 …… 136
16.2 核技巧 …… 137
16.2.1 核作为表达先验的一种形式 …… 140
16.2.2 核函数的特征* …… 141
16.3 软SVM应用核方法 …… 141
16.4 小结 …… 142
16.5 文献评注 …… 143
16.6 练习 …… 143

第17章 多分类、排序与复杂预测问题 …… 145
17.1 一对多和一对一 …… 145
17.2 线性多分类预测 …… 147
17.2.1 如何构建 $\mathbf{w}$ …… 147
17.2.2 对损失敏感的分类 …… 148
17.2.3 经验风险最小化 …… 149
17.2.4 泛化合页损失 …… 149
17.2.5 多分类SVM和SGD …… 150
17.3 结构化输出预测 …… 151
17.4 排序 …… 153
17.5 二分排序以及多变量性能测量 …… 157
17.6 小结 …… 160
17.7 文献评注 …… 160
17.8 练习 …… 161

第18章 决策树 …… 162
18.1 采样复杂度 …… 162
18.2 决策树算法 …… 163
18.2.1 增益测量的实现方式 …… 164
18.2.2 剪枝 …… 165
18.2.3 实值特征基于阈值的拆分规则 …… 165
18.3 随机森林 …… 165
18.4 小结 …… 166
18.5 文献评注 …… 166
18.6 练习 …… 166

第 19 章　最近邻 …… 167
19.1　k 近邻法 …… 167
19.2　分析 …… 168
19.2.1　1-NN 准则的泛化界 …… 168
19.2.2　"维数灾难" …… 170
19.3　效率实践* …… 171
19.4　小结 …… 171
19.5　文献评注 …… 171
19.6　练习 …… 171

第 20 章　神经元网络 …… 174
20.1　前馈神经网络 …… 174
20.2　神经网络学习 …… 175
20.3　神经网络的表达力 …… 176
20.4　神经网络样本复杂度 …… 178
20.5　学习神经网络的运行时 …… 179
20.6　SGD 和反向传播 …… 179
20.7　小结 …… 182
20.8　文献评注 …… 183
20.9　练习 …… 183

第三部分　其他学习模型

第 21 章　在线学习 …… 186
21.1　可实现情况下的在线分类 …… 186
21.2　不可实现情况下的在线识别 …… 191
21.3　在线凸优化 …… 195
21.4　在线感知器算法 …… 197
21.5　小结 …… 199
21.6　文献评注 …… 199
21.7　练习 …… 199

第 22 章　聚类 …… 201
22.1　基于链接的聚类算法 …… 203
22.2　k 均值算法和其他代价最小聚类 …… 203
22.3　谱聚类 …… 205
22.3.1　图割 …… 206
22.3.2　图拉普拉斯与松弛图割算法 …… 206
22.3.3　非归一化的谱聚类 …… 207
22.4　信息瓶颈* …… 208
22.5　聚类的进阶观点 …… 208
22.6　小结 …… 209
22.7　文献评注 …… 210
22.8　练习 …… 210

第 23 章　维度约简 …… 212
23.1　主成分分析 …… 212
23.1.1　当 $d \gg m$ 时一种更加有效的求解方法 …… 214
23.1.2　应用与说明 …… 214
23.2　随机投影 …… 215
23.3　压缩感知 …… 217
23.4　PCA 还是压缩感知 …… 223
23.5　小结 …… 223
23.6　文献评注 …… 223
23.7　练习 …… 223

第 24 章　生成模型 …… 226
24.1　极大似然估计 …… 226
24.1.1　连续随机变量的极大似然估计 …… 227
24.1.2　极大似然与经验风险最小化 …… 228
24.1.3　泛化分析 …… 228
24.2　朴素贝叶斯 …… 229
24.3　线性判别分析 …… 230
24.4　隐变量与 EM 算法 …… 230
24.4.1　EM 是交替最大化算法 …… 232
24.4.2　混合高斯模型参数估计的 EM 算法 …… 233
24.5　贝叶斯推理 …… 233
24.6　小结 …… 235
24.7　文献评注 …… 235
24.8　练习 …… 235

第 25 章　特征选择与特征生成 …… 237
25.1　特征选择 …… 237
25.1.1　滤波器 …… 238

25.1.2 贪婪选择方法 …… 239
25.1.3 稀疏诱导范数 …… 241
25.2 特征操作和归一化 …… 242
25.3 特征学习 …… 244
25.4 小结 …… 246
25.5 文献评注 …… 246
25.6 练习 …… 246

第四部分 高级理论

第 26 章 拉德马赫复杂度 …… 250
26.1 拉德马赫复杂度概述 …… 250
26.2 线性类的拉德马赫复杂度 …… 255
26.3 SVM 的泛化误差界 …… 256
26.4 低 ℓ_1 范数预测器的泛化误差界 …… 258
26.5 文献评注 …… 259

第 27 章 覆盖数 …… 260
27.1 覆盖 …… 260
27.2 通过链式反应从覆盖到拉德马赫复杂度 …… 261
27.3 文献评注 …… 262

第 28 章 学习理论基本定理的证明 …… 263
28.1 不可知情况的上界 …… 263
28.2 不可知情况的下界 …… 264
28.2.1 证明 $m(\epsilon, \delta) \geq 0.5\log(1/(4\delta))/\epsilon^2$ …… 264
28.2.2 证明 $m(\epsilon, 1/8) \geq 8d/\epsilon^2$ …… 265
28.3 可实现情况的上界 …… 267

第 29 章 多分类可学习性 …… 271
29.1 纳塔拉詹维 …… 271
29.2 多分类基本定理 …… 271
29.3 计算纳塔拉詹维 …… 272
29.3.1 基于类的一对多 …… 272
29.3.2 一般的多分类到二分类的简 …… 273
29.3.3 线性多分类预测器 …… 273
29.4 好的与坏的 ERM …… 274
29.5 文献评注 …… 275
29.6 练习 …… 276

第 30 章 压缩界 …… 277
30.1 压缩界概述 …… 277
30.2 例子 …… 278
30.2.1 平行于轴的矩形 …… 278
30.2.2 半空间 …… 279
30.2.3 可分多项式 …… 279
30.2.4 间隔可分的情况 …… 279
30.3 文献评注 …… 280

第 31 章 PAC-贝叶斯 …… 281
31.1 PAC-贝叶斯界 …… 281
31.2 文献评注 …… 282
31.3 练习 …… 282

附录 A 技术性引理 …… 284
附录 B 测度集中度 …… 287
附录 C 线性代数 …… 294
参考文献 …… 297
索引 …… 305

引　论

本书的主题是"自动学习"，后文中我们更经常称之为"机器学习"。机器学习的含义是，希望通过对计算机编程，使它能够根据已有的输入数据进行学习。粗略地说，学习是一个将经验转化为专业技能或知识的过程。输入学习算法的是代表经验的训练数据，而输出的则是知识。这种知识通常以一种可以被其他计算机程序执行任务时所用的形式存在。为寻求这一概念的形式化数学解释，我们必须更明确地了解其中涉及的每个术语的准确含义：程序获取的训练数据是什么？学习过程是如何自动进行的？如何评价这一学习过程的成败(即学习程序输出结果的质量)？

1.1　什么是学习

我们首先来看几个存在于大自然的动物学习的例子。从这些熟悉的例子中可以看出，机器学习的一些基本问题也存在于自然界。

怯饵效应——老鼠学习躲避毒饵：当老鼠遇到有新颖外观或气味的食物时，它们首先会少量进食，随后的进食量将取决于事物本身的风味及其生理作用。如果产生不良反应，那么新的食物往往会与这种不良后果相关联，随之，老鼠不再进食这种食物。很显然，这里有一个学习机制在起作用——动物通过经验来获取判断食物安全性的技能。如果对一种食物过去的经验是负标记的，那么动物会预测在未来遇到它时也会产生负面影响。

前文的示例解释了什么是学习成功，下面我们再举例说明什么是典型的机器学习任务。假设我们想对一台机器进行编程，使其学会如何过滤垃圾邮件。一个最简单的解决方案是仿照老鼠学习躲避毒饵的过程。机器只须记住所有以前被用户标记为垃圾的邮件。当一封新邮件到达时，机器将在先前垃圾邮件库中进行搜索。如果匹配其中之一，它会被丢弃。否则，它将被移动到用户的收件箱文件夹。

虽然上述"通过记忆进行学习"的方法时常是有用的，但是它缺乏一个学习系统的重要特性——标记未见邮件的能力。一个成功的学习器应该能够从个别例子过行泛化，这也称为归纳推理。在前面提到的"怯饵效应"例子中，老鼠遇到一种特定类型的食物后，它们会对新的、没见过的、有相似气味和口味的食物采取同样的态度。为了实现垃圾邮件过滤任务的泛化，学习器可以扫描以前见过的电子邮件，并提取那些垃圾邮件的指示性的词集；然后，当新电子邮件到达时，这台机器可以检查它是否含有可疑的单词，并相应地预测它的标签。这种系统应该有能力正确预测未见电子邮件的标签。

但是，归纳推理有可能推导出错误的结论。为了说明这一点，我们再来思考一个动物学习的例子。

鸽子迷信：心理学家 B. F. Skinner 进行过一项实验，他在笼子里放了一群饥饿的鸽子，笼子上附加了一个自动装置，不管鸽子当时处于什么行为状态，都会以固定的时间间隔为它们提供食物。饥饿的鸽子在笼子里走来走去，当食物第一次送达时，每只鸽子都在进行某项活动(啄食、转动头部等)。食物的到来强化了它们各自特定的行为，此后，每只鸟都倾向于花费更多的时间重复这种行为。接下来，随机的食物送达又增加了

每只鸟做出这种行为的机会。结果是，不管第一次食物送达时，每只鸟处于什么行为状态，这一连串的事件都增强了食物送达和这种行为之间的关联，进而，鸽子们也更频繁地做出这种行为[1]。

有用的学习机制与彩虹迷信的学习机制有何差别？这个问题对自动学习器的发展至关重要。尽管人类可以依靠常识来排除荒诞无意义的学习结论，但是一旦我们将学习任务付之于一台机器，就必须提供定义明确、清晰的规则，来防止程序得出无意义或无用的结论。发展这些规则是机器学习理论的一个核心目标。

是什么使老鼠的学习比鸽子更成功？作为回答这个问题的第一步，我们仔细看一下老鼠在"怯饵效应"实验中的心理现象。

事后审视"怯饵效应"——老鼠未能获得食物与电击或声音与反胃之间的关联：老鼠的怯饵效应机制可能比你想象中的更复杂。Garcia 进行的实验(Garcia & Koelling, 1966)表明，当进食后伴随的是不愉快的刺激时，比如说电击(不是反胃反应)，那么关联没有出现。即使将进食后电击的机制重复多次，老鼠仍然倾向于进食。同样，食物引起的反胃(口味或气味)与声音之间的关联实验也失败了。老鼠似乎有一些"内置的"先验知识，告诉它们，虽然食物和反胃存在因果相关，但是食物与电击或声音与反胃之间不太可能存在因果关系。

由此我们得出结论，怯饵效应和鸽子迷信的一个关键区别点是先验知识的引入使学习机制产生偏差，也称为"归纳偏置"。在实验中，鸽子愿意采取任何食物送达时发生的行为。然而，老鼠"知道"食物不能导致电击，也知道与食物同现的噪音不可能影响这种食物的营养价值。老鼠的学习过程偏向于发现某种模式，而忽略其他的关联。

事实证明，引入先验知识导致学习过程产生偏差，这对于学习算法的成功必不可少(正式陈述与证明参见第5章中的"没有免费的午餐"定理)。这种方法的发展，即能够表示领域知识，将其转化为一个学习偏置，并量化偏置对学习成功的影响，是机器学习理论的一个核心主题。粗略地讲，具有的先验知识(先验假设)越强，越容易从样本实例中进行学习。但是，先验假设越强，学习越不灵活——受先验假设限制。第5章将详细讨论这些问题。

1.2 什么时候需要机器学习

什么时候需要机器学习，而不是直接动手编程完成任务？在指定问题中，程序能否在"经验"的基础上自我学习和提高，有两方面的考虑：问题本身的复杂性和对自适应性的需要。

1. 过于复杂的编程任务

- 动物/人可执行的任务：虽然人类可以习惯性地执行很多任务，但是反思我们如何完成任务的内省机制还不够精细，无法从中提取一个定义良好的程序。汽车驾驶、语音识别和图像识别都属于此类任务。面对此类任务，只要接触到足够多的训练样本，目前最先进的机器学习程序，即能"从经验中学习"的程序，就可以达到比较满意的效果。
- 超出人类能力的任务：受益于机器学习技术，另一大系列任务都涉及对庞大且复杂的数据集进行分析：天文数据、医疗档案转化为医学知识，气象预报，基因组数据

[1] http://psychclassics.yorku.ca/Skinner/Pigeon/。

分析，网络搜索引擎和电子商务。随着越来越多的数字数据的出现，显而易见的是，隐含在数据里的有意义、有价值的信息过于庞大复杂，超出了人类的理解能力。学习在大量复杂数据中发现有意义的模式是一个有前途的领域，无限内存容量加上不断提高的处理速度，更为这一领域开辟了新的视野。

2. 自适应性

编程的局限之一是其刻板性——一旦程序的编写与安装完成，它将保持不变。但是，任务会随着时间的推移而改变，用户也会出现变更。机器学习方法——其行为自适应输入数据的程序——为这个难题提供了一个解决方案。机器学习方法天生具备自适应于互动环境变化的性质。机器学习典型的成功应用有：能够适应不同用户的手写体识别，自动适应变化的垃圾邮件检测，以及语音识别。

1.3 学习的种类

学习是一个非常广泛的领域，因此，机器学习根据学习任务的不同分为不同的子类。这里给出一个粗略的分类，旨在对本书中属于机器学习广泛领域的那部分内容提供一些视角。

下面给出四种分类方式。

监督与无监督：学习涉及学习器与环境之间的互动，那么可以根据这种互动的性质划分学习任务。首先需要关注的是监督学习与无监督学习之间的区别。下面以垃圾邮件检测和异常检测为例说明。对于垃圾邮件检测任务，学习器的训练数据是带标签的邮件（是/否垃圾邮件）。在这种训练的基础上，学习器应该找出标记新电子邮件的规则。相反，对于异常检测任务，学习器的训练数据是大量没有标签的电子邮件，学习器的任务是检测出“不寻常”的消息。

抽象一点来讲，如果我们把学习看做一个“利用经验获取技能”的过程，那么监督学习正是这样的一种场景：经验是包含显著信息（是/否垃圾邮件）的训练数据，“测试数据”缺少这些显著信息，但可从学到的“技能”中获取。此种情况下，获得的“技能”旨在预测测试数据的丢失信息，我们可以将环境看做通过提供额外信息（标签）来“监督”学习器的老师。然而，无监督学习的训练数据和测试数据之间没有区别。学习器处理输入数据的目标是提取概括信息（浓缩数据）。聚类（相似数据归为一类）是执行这样任务的一个典型例子。

还有一种中间情况，训练数据比测试数据包含更多的信息，也要求学习器预测更多信息。举个例子，当学习数值函数判断国际象棋游戏中白棋和黑棋谁更有利时，训练过程中提供给学习器的唯一信息是，谁在整个实际的棋牌类游戏中最终赢得那场比赛的标签。这种学习被称作“强化学习”。

主动学习器与被动学习器：学习可依据学习器扮演的角色不同分类为“主动”和“被动”学习器。主动学习器在训练时通过提问或实验的方式与环境交互，而被动学习器只观察环境（老师）所提供的信息而不影响或引导它。请注意，垃圾邮件过滤任务通常是被动学习——等待用户标记电子邮件。我们可以设想，在主动学习中，要求用户来标记学习器挑选的电子邮件，以提高学习器对“垃圾邮件是什么”的理解。

老师的帮助：人类的学习过程中（在家的幼儿或在校的学生）往往会有一个良师，他向学习者传输最有用的信息以实现学习目标，相比之下，科学家研究自然时，环境起到了老师的作用。环境的作用是消极的——苹果坠落、星星闪烁、雨点下落从不考虑学习者的需

求。在对这种学习情境建模时，我们假定训练数据（学习者的经验）是由随机过程产生的，这是统计机器学习的一个基本构成单元。此外，学习也发生在学习者的输入是由对立“老师”提供的。垃圾邮件过滤任务（如果垃圾邮件制作者尽力误导垃圾邮件过滤器设计者）和检测欺诈学习任务就是这种情况。当不存在更好的假设时，我们也会使用对立老师这一最坏方案。如果学习器能够从对立老师中学习，那么遇到任何老师都可以成功。

在线与批量：在线响应还是处理大量数据后才获得技能，是对学习器的另一种分类方式。举个例子，股票经纪人必须基于当时的经验信息做出日常决策。随着时间推移，他或许会成为专家，但是也会犯错并付出高昂的代价。相比之下，在大量的数据挖掘任务中，学习器，也就是数据挖掘器，往往是在处理大量训练数据之后才输出结论。

在本书中，我们只选取一部分机器学习技术进行讨论。重点是被动的、有监督的、统计批量学习（例如，基于大量独立收集的且带有病人最终结果标记的诊断记录，学习如何预测病人结果）。另外，本书也对在线学习和无监督批量学习（尤其是聚类）做了介绍。

5

1.4 与其他领域的关系

作为一门交叉学科，机器学习与统计学、信息论、博弈论、最优化等众多数学分支有着共同点。我们的最终目标是在计算机上编写程序，所以机器学习自然也是计算机科学的一个分支。在某种意义上，机器学习可以视为人工智能的一个分支，毕竟，要将经验转变成专业知识或从复杂感知数据中发现有意义的模式的能力是人类和动物智能的基石。但是，应该注意的是，与传统人工智能不同，机器学习并不是试图自动模仿智能行为，而是利用计算机的优势和特长与人类的智慧相得益彰。机器学习常用于执行远远超出人类能力的任务。例如，机器学习程序通过浏览和处理大型数据，能够检测到超出人类感知范围的模式。

机器学习（的经验）训练涉及的数据往往是随机生成的。机器学习的任务就是处理这些背景下的随机生成样本，得出与背景相符的结论。这样的描述强调了机器学习与统计学的密切关系。两个学科之间确实有很多共同点，尤其表现在目标和技术方面。但是，两者之间仍然存在显著的差别：如果一个医生提出吸烟与心脏病之间存在关联这一假设，这时应该由统计学家去查看病人样本并检验假设的正确性（这是常见的统计任务——假设检验）。相比之下，机器学习的任务是利用患者样本数据找出心脏病的原因。我们希望自动化技术能够发现被人类忽略的、有意义的模式（或假设）。

与传统统计学不同，算法在机器学习中（尤其在本书里）扮演了重要的角色。机器学习算法要靠计算机来执行，因此算法问题是关键。我们开发算法完成学习任务，同时关心算法的计算效率。两者的另外一个区别是，统计关心算法的渐近性（如随着样本量增长至无穷大，统计估计的收敛问题），机器学习理论侧重于有限样本，也就是说，给定有限可用样本，机器学习理论旨在分析学习器可达到的准确度。

6

机器学习与统计学之间还有很多差异，我们在此仅提到了少数。比如，在统计学中，常首先提出数据模型假设（生成数据呈正态分布或依赖函数为线性）；在机器学习中常考虑“非参数”背景，对数据分布的性质假设尽可能地少，学习算法自己找出最接近数据生成过程的模型。深入讨论这个问题需要更多的技术基础，详见第5章。

1.5 如何阅读本书

本书第一部分是机器学习的基本理论知识，从某种意义上讲，这是本书其余部分的基础。这部分应该作为机器学习理论入门课程的基础。

第二部分介绍了最常见的监督机器学习算法。部分内容可作为机器学习的介绍内容用于面向计算机科学、数学、工程类学生的人工智能课程。

第三部分讨论了统计分类等其他学习模型，包括在线学习、无监督学习、维数约简、生成模型和特征学习。

第四部分是高级理论，主要面向对机器学习方向有科研兴趣的读者。此部分涵盖了更多的数学方法，用于分析和推动机器学习理论发展。

附录给出了书中用到的一些数学方法，其中包括测度集中度理论和线性代数的基础结论。

标注星号的章节，更适合高年级的学生。大部分章的后面都有练习，课程网站上有解答。

建议的教学计划

A. 面向研究生的入门课程(14周)

1. 第2~4章
2. 第9章(略过VC计算)
3. 第5和6章(略过证明)
4. 第10章
5. 第7和11章(略过证明)
6. 第12和13章(可选取一些简单的证明)
7. 第14章(可选取一些简单的证明)
8. 第15章
9. 第16章
10. 第18章
11. 第22章
12. 第23章(略过压缩感知证明)
13. 第24章
14. 第25章

B. 面向研究生的高级课程(14周)

1. 第26和27章
2. (继续)
3. 第6和28章
4. 第7章
5. 第31章
6. 第30章
7. 第12和13章
8. 第14章
9. 第8章
10. 第17章
11. 第29章
12. 第19章
13. 第20章
14. 第21章

1.6 符号

本书所有使用的符号都是符合标准或提前定义的，本节给出一些约定（符号汇总见表1.1）。读者可跳过此部分，遇到符号定义不清楚时再返回本节。

表1.1 符号汇总表

符号	含义
$\mathbb{R}$	实数集
$\mathbb{R}^d$	$\mathbb{R}$上的d维向量集
$\mathbb{R}_+$	非负实数集
$\mathbb{N}$	自然数集
O, o, Θ, ω, Ω, $\tilde{O}$	渐近记号
$\mathbb{I}[\text{逻辑表达式}]$	指示函数（布尔表达式为真时等于1，否则等于0）
$[a]_+$	$=\max\{0, a\}$
$[n]$	集合$\{1, \cdots, n\}$（$n\in\mathbb{N}$）
$\boldsymbol{x}$, $\boldsymbol{v}$, $\boldsymbol{w}$	（列）向量
x_j, v_j, w_j	向量的第j个元素
$\langle \boldsymbol{x}, \boldsymbol{v}\rangle$	$=\sum_{j=1}^{d} x_j v_j$（内积）
$\|\boldsymbol{x}\|_2$或$\|\boldsymbol{x}\|$	$=\sqrt{\langle \boldsymbol{x}, \boldsymbol{x}\rangle}$（$\boldsymbol{x}$的$\ell_2$范数）
$\|\boldsymbol{x}\|_1$	$=\sum_{j=1}^{d} \|x_j\|$（$\boldsymbol{x}$的ℓ_1范数）
$\|\boldsymbol{x}\|_\infty$	$=\max_j \|x_j\|$（$\boldsymbol{x}$的ℓ_∞范数）
$\|\boldsymbol{x}\|_0$	$\boldsymbol{x}$中非零元素的个数
$A\in\mathbb{R}^{d\times k}$	$\mathbb{R}$上的$d\times k$矩阵
A^{T}	A的转置
$A(i, j)$	A的第(i, j)个元素
$\boldsymbol{x}\boldsymbol{x}^{\mathrm{T}}$	$d\times d$矩阵A，满足$A_{ij}=x_i x_j$（$\boldsymbol{x}\in\mathbb{R}^d$）
$\boldsymbol{x}_1, \cdots, \boldsymbol{x}_m$	包含m个向量的序列
x_{ij}	序列中第i个向量的第j个元素
$\boldsymbol{w}^{(1)}, \cdots, \boldsymbol{w}^{(T)}$	迭代算法中向量$\boldsymbol{w}$的值
$w_j^{(t)}$	向量$\boldsymbol{w}^{(t)}$的第j个元素
$\mathcal{X}$	实例空间（集合）
$\mathcal{Y}$	标签空间（集合）
$\mathcal{Z}$	样本空间（集合）
$\mathcal{H}$	假设空间（集合）
$\ell: \mathcal{H}\times\mathcal{Z}\to\mathbb{R}_+$	损失函数
$\mathcal{D}$	（$\mathcal{Z}$或$\mathcal{X}$上的）概率分布
$\mathcal{D}(A)$	在分布$\mathcal{D}$下，集合$A\subseteq\mathcal{Z}$上的概率
$z\sim\mathcal{D}$	基于分布$\mathcal{D}$的采样
$S=z_1, \cdots, z_m$	m个样本序列
$S\sim\mathcal{D}^m$	在分布$\mathcal{D}$下，独立同分布采样$S=z_1, \cdots, z_m$
$\mathbb{P}$, $\mathbb{E}$	随机变量的概率和期望
$\mathbb{P}_{z\sim\mathcal{D}}[f(z)]$	$=\mathcal{D}(\{z: f(z)=\text{真}\})$（$f: \mathcal{Z}\to\{\text{真}, \text{假}\}$）

（续）

符号	含义
$\mathbb{E}_{z\sim\mathcal{D}}[f(z)]$	随机变量 $f: Z\to\mathbb{R}$ 的期望
$N(\mu, C)$	高斯分布（期望 μ，协方差 C）
$f'(x)$	函数 $f:\mathbb{R}\to\mathbb{R}$ 在 x 处的一阶导数
$f''(x)$	函数 $f:\mathbb{R}\to\mathbb{R}$ 在 x 处的二阶导数
$\frac{\partial f(\mathbf{w})}{\partial w_i}$	函数 $f:\mathbb{R}^d\to\mathbb{R}$ 在 $\mathbf{w}$ 处的偏微分
$\nabla f(\mathbf{w})$	函数 $f:\mathbb{R}^d\to\mathbb{R}$ 在 $\mathbf{w}$ 处的梯度
$\partial f(\mathbf{w})$	函数 $f:\mathbb{R}^d\to\mathbb{R}$ 在 $\mathbf{w}$ 处的微分
$\min_{x\in C} f(x)$	$=\min\{f(x): x\in C\}$ 函数 f 在 C 上的最小值
$\max_{x\in C} f(x)$	$=\max\{f(x): x\in C\}$ 函数 f 在 C 上的最大值
$\operatorname{argmin}_{x\in C} f(x)$	集合 $\{x\in C: f(x)=\min_{x\in C} f(x)\}$
$\operatorname{argmax}_{x\in C} f(x)$	集合 $\{x\in C: f(x)=\max_{x\in C} f(x)\}$
$\log$	自然对数

我们使用小写字母表示标量和对象（例如 x，λ），强调对象是向量时使用黑体（例如 $\mathbf{x}$，$\boldsymbol{\lambda}$）。向量 $\mathbf{x}$ 的第 i 个元素表示为 x_i。使用大写字母表示矩阵、集合和序列。接下来我们看到，一个学习算法的输入是一个训练样本序列。用 z 表示样本，用 $S=z_1, \cdots, z_m$ 表示 m 个样本序列。通常，S 表示训练集合，本书中使用 S 表示序列。$\mathbf{x}_1, \cdots, \mathbf{x}_m$ 表示有 m 个向量的序列，其中向量 $\mathbf{x}_i$ 的第 j 个元素表示为 $x_{i,j}$。

本书使用概率论中的符号表示。$\mathcal{D}$ 表示集合上的分布[⊖]，$z\sim\mathcal{D}$ 表示分布 $\mathcal{D}$ 上的采样。随机变量 $f: Z\to\mathbb{R}$ 的期望是 $\mathbb{E}_{z\sim\mathcal{D}}[f(z)]$。当 z 在上下文中意思明确时简记为 $\mathbb{E}[f]$。当 $f: Z\to\mathbb{R}\{真, 假\}$ 时，分布 $\mathcal{D}(\{z: f(z)=真\})$ 也记为 $\mathbb{P}_{z\sim\mathcal{D}}[f(z)]$。另有，$Z^m=\{z_1, \cdots, z_m\}$ 的概率分布记为 $\mathcal{D}^m$，其中 Z^m 中的每个点 z_i 都是独立于其他点的 $\mathcal{D}$ 上的采样。

总体上，我们尽量避免使用渐近符号。为清晰重要结果，偶尔使用。特别地，有 $f: \mathbb{R}\to\mathbb{R}$ 和 $g: \mathbb{R}\to\mathbb{R}_+$，如果存在 $x_0, a\in\mathbb{R}_+$，对于所有的 $x>x_0$ 满足 $f(x)\leqslant ag(x)$，记为 $f=O(g)$；如果对于所有的 $a>0$，都存在 $x>x_0$ 满足 $f(x)\leqslant ag(x)$，记为 $f=o(g)$；如果存在 $x_0, a\in\mathbb{R}_+$，对于所有的 $x>x_0$ 满足 $f(x)\geqslant ag(x)$，记为 $f=\Omega(g)$；如果对于所有的 $a>0$，都存在 $x>x_0$ 满足 $f(x)\geqslant ag(x)$，记为 $f=\omega(g)$；当 $f=O(g)$，$g=O(f)$ 满足时，记为 $f=\Theta(g)$；如果存在 $k\in\mathbb{N}$，满足 $f(x)=O(g(x)\log^k(g(x)))$，记为 $f=\tilde{O}(g)$。

向量 $\mathbf{x}$ 和 $\mathbf{w}$ 默认为欧式空间上的 d 维向量，其内积表示为 $\langle \mathbf{x}, \mathbf{w}\rangle=\sum_{i=1}^{d} x_i w_i$。向量 $\mathbf{w}$ 的 ℓ_2 范数（欧式范数）表示为 $\|\mathbf{w}\|_2=\sqrt{\langle \mathbf{w}, \mathbf{w}\rangle}$，当上下文意义明确时，下标 2 省略。推广一下，$\ell_p$ 范数 $\|\mathbf{w}\|_p=\left(\sum_i |w_i|^p\right)^{1/p}$，其中 $\|\mathbf{w}\|_1=\sum_{i=1}^{d} |w_i|$，$\|\mathbf{w}\|_\infty=\max_i |w_i|$。

我们使用 $\min_{x\in C} f(x)$ 表示集合 $\{f(x): x\in C\}$ 上的最小值。从数学意义上准确地讲，当无法求出最小值时，我们应该使用 $\inf_{x\in C} f(x)$。然而，本书中用到的 inf 和 min 的区别很小，为了表述简单，虽然有些时候 inf 更准确，我们还是使用 min。最大值 max 和上确界 sup 也是同样情况。

⊖ 用准确的数学语言讲，$\mathcal{D}$ 应该在 Z 的子集的 σ-代数上定义。不熟悉测度论的读者可跳过与测度论中定义和假设相关的脚注。

第一部分

Understanding Machine Learning: From Theory to Algorithms

理论基础

简易入门

让我们从相对简单的设定开始，用数学分析展示如何取得成功的学习。假设你刚刚到达太平洋上的一个小岛，很快你发现木瓜是当地饮食中一个重要的组成部分，然而，你从来没有吃过木瓜。所以你必须学会如何判断市场上售卖的木瓜是否好吃。首先，你需要选择根据木瓜的哪些特征来给出判断。基于你之前选择其他水果的经验，你决定利用以下两个特征：木瓜的颜色（范围从暗绿色、橘黄色、红色到深棕色）、木瓜的软硬程度（范围从岩石般坚硬到浆糊般柔软）。为了获得对木瓜的判断，你的输入样本由以下属性决定：①通过观测获得木瓜的颜色和软硬程度；②通过亲口尝试确定这些木瓜到底好不好吃。下面让我们结合这个任务来分析并证明学习问题中需要考虑的因素。

我们的第一个步骤是描述一个能够刻画类似学习任务的形式化模型。

2.1 一般模型——统计学习理论框架

1. 学习器的输入

在基础的统计学设定中，学习器应该预先接触以下概念：

- **领域集**(domain set)：一个任意的集合$\mathcal{X}$。这个集合中的实例是我们希望能够为其贴上标签的。例如，在之前提到的木瓜学习问题中，领域集为所有木瓜的集合。这些领域集中的元素通常用一个能够表征其特征的向量表示（如木瓜的颜色和软硬程度）。我们也把领域中的元素称为实例，相应地，$\mathcal{X}$被称为实例空间。
- **标签集**(label set)：就目前讨论的内容来说，我们将标签集限定为一个二元集合，通常为$\{0, 1\}$或者$\{-1, +1\}$。令$\mathcal{Y}$为集合中可能的标签。对应于木瓜的例子，假定标签集$\mathcal{Y}$为$\{0, 1\}$，其中 1 代表木瓜好吃，0 表示木瓜难吃。
- **训练数据**(training data)：$S=((x_1, y_1), \cdots, (x_m, y_m))$为一个有限的序列，序列中的元素以$\mathcal{X}\times\mathcal{Y}$形式成对出现。也就是说，训练集是一个由带标签的领域集元素组成的序列。这个输入数据是学习器能够接触到的（例如有一堆木瓜，我们能够观测到它们颜色、软硬程度，同时也知道它们好不好吃）。这些带标签的样本通常称为训练样本，我们有时称S为训练集[⊖]。

[13]

2. 学习器的输出

要求学习器输出一个预测规则(prediction rule)，$h:\mathcal{X}\to\mathcal{Y}$。该函数也称为预测器(predictor)、假设(hypothesis)或分类器(classifier)。这个预测器可以用来预测一个新的领域元素的标签。在木瓜的例子中，学习器预测规则用来预测在农贸市场中我们将要检查的木瓜是否好吃。我们用$A(S)$来表示学习算法A在给定训练序列S的情况下返回的假设。

3. 一个简单的数据生成模型

下面介绍训练数据是如何产生的。首先假设实例（对应于木瓜）根据某些概率分布$\mathcal{D}$（对

⊖ 尽管这里用的是“集合”的概念，但是S是一个序列。尤其是当有两个相同的样本同时出现在S中时，某些算法可以利用样本在S中的顺序关系。

应于岛上的环境)采样获得。必须注意的是，我们并不需要学习器知道此概率分布的任何信息。对于我们讨论的学习任务来说，$\mathcal{D}$可以为任意的概率分布。在本章的讨论中，我们假设存在一些“正确”标记函数 $f:\mathcal{X}\to\mathcal{Y}$，使得对于任意的 i，$y_i=f(x_i)$。该假设将在下一章中被适当放松。对于学习器来说，该“正确”标记函数是未知的。实际上，指出每个样本的正确标签正是学习器需要完成的任务。综上，训练序列 S 中的每一对训练数据的产生过程是：首先根据概率分布$\mathcal{D}$采集样本点 x_i，然后利用“正确”标记函数 f 为其赋予标签。

4. 衡量成功

分类器误差定义为：未能成功预测随机数据点正确标签的概率(随机数据点是从之前提到的潜在分布中生成的)。也就是说，h 的误差是 $h(x)\neq f(x)$的概率，其中 x 是根据分布$\mathcal{D}$采集的随机样本。

形式上，给定一个领域子集(domain subset)[1] $A\subset\mathcal{X}$，概率分布$\mathcal{D}$，$\mathcal{D}(A)$决定了能够观测到 $x\in A$ 的概率。很多情况下，我们称 A 为一个事件，将其表达为一个函数 $\pi:\mathcal{X}\to\{0,1\}$，也就是说，$A=\{x\in\mathcal{X}:\pi(x)=1\}$。在这种情况下，我们也用 $\mathbb{P}_{x\sim\mathcal{D}}[\pi(x)]$来表示$\mathcal{D}(A)$。

预测准则($h:\mathcal{X}\to\mathcal{Y}$)的错误率定义为：

$$L_{(\mathcal{D},f)}(h)\stackrel{\text{def}}{=}\mathbb{P}_{x\sim\mathcal{D}}[h(x)\neq f(x)]\stackrel{\text{def}}{=}\mathcal{D}(\{x:h(x)\neq f(x)\}) \tag{2.1}$$

也就是说，h 的误差是随机选择一个样本 x，使得 $h(x)\neq f(x)$的概率。下标($\mathcal{D}$，f)说明误差的测量基于概率分布$\mathcal{D}$和正确标记函数 f。在以后的章节中我们将省略该下标。$L_{(\mathcal{D},f)}(h)$也称为泛化误差、损失或者 h 的真实误差。在本书中，我们将交叉地使用这些名称。因为泛化误差是与用户损失(loss)等价的，所以我们用字母 L 表示误差。在本书后面的内容中，我们也会讨论这种损失的其他可能的形式。 14

5. 注意事项：学习器可接触到的信息

对于分布$\mathcal{D}$和标记函数 f，学习器是未知的。在木瓜的例子中，我们刚刚到达一个新的小岛，对于木瓜的分布和如何预测木瓜味道一无所知。学习者与小岛中新环境接触的唯一方式就是通过观察训练集。

下一节将介绍一种开始算法设计和分析算法效果的简单范例。

2.2 经验风险最小化

之前提到，一个学习算法的输入是一个训练集 S，训练集从一个未知分布$\mathcal{D}$中采样获得，通过目标函数 f 对训练样本进行标记。我们需要输出一个预测器 $h_S:\mathcal{X}\to\mathcal{Y}$(下标 S 说明输出的预测器是基于训练集 S 的)。学习算法的目标是求出一个最小的预测器 h，使得关于未知分布$\mathcal{D}$和 f 的预测误差最小化。

由于学习器并不知道$\mathcal{D}$和 f 是什么样的，所以无法直接获知真实误差。学习器能够计算出来的一个有用的概念是训练误差——分类器在训练样本中导致的误差：

$$L_S(h)\stackrel{\text{def}}{=}\frac{|\{i\in[m]:h(x_i)\neq y_i\}|}{m} \tag{2.2}$$

其中$[m]=\{1,\cdots,m\}$。

术语经验误差或经验风险对于该误差通常可以互换使用。

---

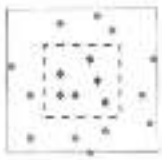
[1] 严格说来，我们应该要求在给定分布$\mathcal{D}$的情况下，A 属$\mathcal{X}$子集的某些 σ-代数中的一员。我们将在下面的章节给出可测性假设(measurability assumptions)的形式化定义。

对于学习者来说，训练样本是真实世界的一个缩影，因此利用训练集来寻找一个对于数据的可行解是合理的。这些学习范例——从预测器 h 出发到最小化 $L_S(h)$——称为经验风险最小化，或简称为 ERM。

可能出现的失误——过拟合

尽管 ERM 规则看起来顺理成章，但是如果不小心，这种方法可能惨遭失败。

举例说明这种失败，我们回到基于软硬程度和颜色预测木瓜味道的学习问题上。假设一个样本如下图所示：



15 假设概率分布$\mathcal{D}$使得实例(木瓜)在如上图所示灰色正方形中均匀分布。标记函数 f 决定了实例如果出现在正方形的内部，那么其标记为 1，否则标记为 0。图中灰色正方形的面积为 2，其中间的正方形面积为 1。思考如下预测器：

$$h_S(x)=\begin{cases} y_i & \text{如果 } \exists i\in[m] \text{ s.t. } x_i=x \\ 0 & \text{否则} \end{cases} \tag{2.3}$$

以上预测器看起来人工设计的痕迹太重，在练习 2.1 中，我们介绍如何利用多项式更自然地表示这个预测器。显然，无论样本是什么，$L_S(h_S)=0$，因此预测器可能会选择一种 ERM 算法(这是一种经验最小损失假设，没有分类器会比这种假设具有更小的误差)。另一方面，任务分类器通过有限个数的实例来预测标记 1 的真实误差，在本例中，为 1/2。于是 $L_{\mathcal{D}}(h_S)=1/2$。我们发现一个预测器在训练集上的效果非常优秀，但是在真实世界中的表现非常糟糕，这种现象称为过拟合。直观上，过拟合发生在当假设对于训练集契合地"太好了"(也许正如我们日常生活中的经验一样：一个人如果能对自己的每一个行为都能做出完美的解释，那么这个人是容易令人产生怀疑的)。

2.3 考虑归纳偏置的经验风险最小化

我们刚刚证明了 ERM 规则容易导致过拟合。相较于就此抛弃 ERM 范例，我们更倾向于寻找方法来修正它。我们将寻找保证 ERM 不会导致过拟合的条件。也就是说，在这样的条件下，ERM 预测器既能够在训练数据中获得不错的表现，也有较大可能性在潜在的数据分布下表现良好。

通常的解决方案是在一个受限的搜索空间中使用 ERM 学习准则。形式上，一个学习器应该提前选择(在接触到数据之前)一个预测器的集合。这个集合称为假设类，记为$\mathcal{H}$，每一个 $h\in\mathcal{H}$ 是从$\mathcal{X}$映射到$\mathcal{Y}$的一个函数。对于给定的假设类$\mathcal{H}$和一个训练样本集 S，$\mathrm{ERM}_{\mathcal{H}}$ 学习器根据在 S 上的最小化经验误差，利用 ERM 规则选择一个预测器 $h\in\mathcal{H}$。形式上如下

$$\mathrm{ERM}_{\mathcal{H}}(S)\in\operatorname*{argmin}_{h\in\mathcal{H}} L_S(h)$$

其中，argmin 表示从假设集合$\mathcal{H}$中选择使得 $L_S(h)$最小的假设 h。通过限制学习器从$\mathcal{H}$中选择预测器，我们的选择偏向于一个特别的预测器集合。这种限制通常称为归纳偏置。因为这种选择决定于学习器接触训练数据之前，所以它应该基于一些需要学习问题的先验知识。举例来说，对于木瓜味道的预测问题，我们或许可以选择由轴对称矩形(二维空间两

个维度分别颜色和软硬程度）所确定的预测器集合作为假设类$\mathcal{H}$。后面我们将说明基于此假设类的$\mathrm{ERM}_{\mathcal{H}}$为什么能够保证不过拟合。另一方面，我们之前看到的过拟合例子证明了：存在一类预测器，它们包括所有将有限集合中的元素设定为1的函数。选择此类预测器作为假设类$\mathcal{H}$是不足以防止$\mathrm{ERM}_{\mathcal{H}}$过拟合的。 16

在学习理论中，一个基本的问题是：选择哪种假设类$\mathrm{ERM}_{\mathcal{H}}$不会导致过拟合。本书接下来的章节将探讨这个问题。

直观上，选择一个更加严格受限的假设类能够更好地防止过拟合，但与此同时，也会带来更强的归纳偏置。我们后面会重新考虑这两者的权衡。

有限假设类

对于一个类来说，最简单的一种限制就是限定其势的上界（也就说$\mathcal{H}$中预测器h的个数）。本节说明了如果$\mathcal{H}$是有限类，$\mathrm{ERM}_{\mathcal{H}}$将不会过拟合，前提是拥有足够多的训练样本（其大小依赖于假设类$\mathcal{H}$的势）。

让学习器在有限假设类上选择预测规则是一种适度温和的限制。比如说，$\mathcal{H}$是一个预测器组成的集合，这些预测器可以通过C++程序利用至多10^9位的代码实现。在木瓜的例子中，我们提到由一系列轴对称地形组成的类。虽然这是一个无限类，但是如果我们离散化实数表示，例如，利用一个64位的浮点数表示，这个假设类将转化为一个有限类。

接下来我们分析$\mathrm{ERM}_{\mathcal{H}}$在有限假设类$\mathcal{H}$的前提下的表现。对于一个训练样本$S$，利用某些标记函数$f: \mathcal{X} \to \mathcal{Y}$为其贴上标签。设$h_S$为对$S$利用$\mathrm{ERM}_{\mathcal{H}}$得到的结果，也就是说

$$h_S \in \operatorname*{argmin}_{h \in \mathcal{H}} L_S(h) \tag{2.4}$$

在本章中，我们做出如下简化的假设（这些假设在下一章节中将放宽）。

定义2.1（可实现性假设） *存在$h^* \in \mathcal{H}$，使得$L_{\mathcal{D},f}(h^*)=0$。注意，这个假设意味着对于任意的随机样本集S（其中S中实例是根据分布$\mathcal{D}$随机采集的，其标签由f决定）以概率1使得$L_S(h^*)=0$。*

可实现性假设意味着对于每个ERM假设，我们有[⊖]$L_S(h_S)=0$。然而，相对于经验风险来说，我们更加感兴趣于h_S真实的风险$L_{\mathcal{D},f}(h_S)$。

显然，对于一个只能接触到样本集S的算法来说，关于潜在分布$\mathcal{D}$的任何误差保证都必须依赖于$\mathcal{D}$和S之间的关系。在统计机器学习中最通常的假设是S中训练样本是从$\mathcal{D}$中独立同分布地抽取的。形式上， 17

独立同分布（i.i.d.）假设：训练集中的样本根据分布$\mathcal{D}$独立同分布。也就是说，S中每一个x_i采样于$\mathcal{D}$，然后根据标记函数f确定其标签。记为$S \sim \mathcal{D}^m$，其中m为S的势。$\mathcal{D}^m$表示m-组（m-tuples）的概率，对于m-组中的每一个元素，都是独立于组中其他元素而从$\mathcal{D}$中独立抽取的。

直观上，训练集S是一个学习器从整体数据分布$\mathcal{D}$和标记函数f中获取的部分信息，是使得学习器能够接触到外部世界的一个窗口。训练样本越大，越能准确地反映数据分布和标记函数，从而利用其生成此分布和函数。

由于$L_{\mathcal{D},f}(h_S)$依赖于训练集S，而训练集通过一个随机过程采样，因此通过风险$L_{\mathcal{D},f}(h_S)$来选择预测器h_S也存在随机性。这就是所谓的随机变量。学习器试图通过完全

⊖ 从数学上说，这种情况以概率1成立，为了简化表述，我们通常省略"以概率1"这个说明符。

确定的 S 来确定一个好的分类器(从观测$\mathcal{D}$的角度来说)。这种想法是不实际的，因为总有一定的概率使得采样获得训练数据中有一些训练数据对于分布$\mathcal{D}$来说完全不具有代表性。回到木瓜的例子，即使岛上只有70%的木瓜是好吃的，但也有可能(尽管几率较小)我们尝到的所有木瓜全都是不好吃的。在这种情况下，$\mathrm{ERM}_{\mathcal{H}}(S)$会选择一个固定的函数标记所有的木瓜都是“不好吃”的(这种选择对于岛上木瓜的真实分布有70%的错误概率)。因此，在 $L_{\mathcal{D},f}(h_S)$不太大的情况下，我们将处理训练样本的采样概率。一般来说，我们将采样到非代表性样本的概率表示为 δ，同时 $1-\delta$ 在该预测中称为**置信参数**(confidence parameter)。

由于无法保证标签预测绝对准确，因此我们引入另外一个参数来评价预测的质量，称为**精度参数**(accuracy parameter)，记为 ε。如果 $L_{\mathcal{D},f}(h_S)>\varepsilon$，那么对于学习器来说，这是一个失败的预测。如果 $L_{\mathcal{D},f}(h_S)\leqslant\varepsilon$，我们认为该算法输出了一个近似正确的预测。因此(固定一些标记函数 $f:\mathcal{X}\to\mathcal{Y}$)，我们有意设定学习器对 m 组实例采样失败的概率上界，形式上，设 $S|_x=(x_1,\cdots,x_m)$为训练实例集，其上界为

$$\mathcal{D}^m(\{S|_x:L_{\mathcal{D},f}(h_S)>\varepsilon\})$$

设$\mathcal{H}_B$为“差”的假设集合，也就是

$$\mathcal{H}_B=\{h\in\mathcal{H}:L_{\mathcal{D},f}(h)>\varepsilon\}$$

此外，设

$$M=\{S|_x:\exists h\in\mathcal{H}_B,L_S(h)=0\}$$

为一个样本的误导集：对于所有的 $S|_x\subset M$，存在一个“差”的假设 $h\in\mathcal{H}_B$，使其看上去像一个“好”的假设。现在我们回顾对 $L_{\mathcal{D},f}(h_S)>\varepsilon$ 的概率限制，因为假设的可实现性意味着 $L_S(h_S)=0$，所以，只有当 $h\in\mathcal{H}_B$，$L_S(h_S)=0$ 时，$L_{\mathcal{D},f}(h_S)>\varepsilon$ 的情况才会出现。换句话说，这种情况发生的充分必要条件是我们的样本处于误导样本集 M 中。形式上，我们将其表示为

$$\{S|_x:L_{\mathcal{D},f}(h_S)>\varepsilon\}\subseteq M$$

注意，M 可以写为

$$M=\bigcup_{h\in\mathcal{H}_B}\{S|_x:L_S(h)=0\} \tag{2.5}$$

因此，

$$\mathcal{D}^m(\{S|_x:L_{\mathcal{D},f}(h_S)>\varepsilon\})\leqslant\mathcal{D}^m(M)=\mathcal{D}^m\Big(\bigcup_{h\in\mathcal{H}_B}\{S|_x:L_S(h)=0\}\Big) \tag{2.6}$$

接下来，我们利用联合界(概率学中一个基本的性质)对上式等号右边的公式进行上界限制。

引理 2.2(联合界)　对于任意的集合 A，B 以及分布$\mathcal{D}$，有

$$\mathcal{D}(A\cup B)\leqslant\mathcal{D}(A)+\mathcal{D}(B)$$

对式(2.6)利用联合界引理，得出

$$\mathcal{D}^m(\{S|_x:L_{\mathcal{D},f}(h_S)>\varepsilon\})\leqslant\sum_{h\in\mathcal{H}_B}\mathcal{D}^m(\{S|_x:L_S(h)=0\}) \tag{2.7}$$

下面，让我们限制上述不等式中右边的每个被加数。固定某“差”假设 $h\in\mathcal{H}_B$，$L_S(h)=0$等同于$\forall i$，$h(x_i)=f(x_i)$。由于训练集中的样本独立同分布采样，我们得到

$$\mathcal{D}^m(\{S|_x:L_S(h)=0\})=\mathcal{D}^m(\{S|_x:\forall i,h(x_i)=f(x_i)\})$$

18

$$= \prod_{i=1}^{m} \mathcal{D}(\{x_i : h(x_i) = f(x_i)\}) \tag{2.8}$$

对于训练集中每个独立的样本，有

$$\mathcal{D}(\{x_i : h(x_i) = y_i\}) = 1 - L_{\mathcal{D},f}(h) \leqslant 1 - \varepsilon$$

其中，最后一项不等式由 $h \in \mathcal{H}_B$ 而得。结合上个等式与式(2.8)，利用不等式 $1-\varepsilon \leqslant e^{-\varepsilon}$，可得，对于所有的 $h \in \mathcal{H}_B$

$$\mathcal{D}^m(\{S|_x : L_S(h) = 0\}) \leqslant (1-\varepsilon)^m \leqslant e^{-\varepsilon m} \tag{2.9}$$

结合此公式与式(2.7)，可得

$$\mathcal{D}^m(\{S|_x : L_{\mathcal{D},f}(h_S) > \varepsilon\}) \leqslant |\mathcal{H}_B| e^{-\varepsilon m} \leqslant |\mathcal{H}| e^{-\varepsilon m}$$

图2.1展示了我们是如何使用联合界的。

图2.1 大圆中的每个点表示 m-组中一个可能的实例。不同颜色的椭圆对应于"差"预测器 $h \in \mathcal{H}_B$ 的 m-组误导集。ERM规则在误导训练集 S 中可能会出现过拟合。也就是说，对于某些 $h \in \mathcal{H}_B$，我们有 $L_S(h)=0$。式(2.9)保证对于每个单独的差假设 $h \in \mathcal{H}_B$，至多训练集的 $(1-\varepsilon)^m$ 部分会被误导。尤其当 m 变大，这些带颜色的椭圆会变得越小，其面积表示训练集中被 $h \in \mathcal{H}_B$ 误导的训练集(即 M 中的训练集)大小。联合界指出，这些被误导训练数据的最大面积为这些椭圆面积之和。因此，其上界为 $|\mathcal{H}_B|$ 乘以带颜色椭圆的最大尺寸。带颜色椭圆外的任何样本集 S 都不会引起ERM规则的过拟合

推论2.3 设 $\mathcal{H}$ 为一个有限假设类，$\delta \in (0,1)$，$\varepsilon > 0$，m 为一个整数，以下不等式成立

$$m \geqslant \frac{\log(|\mathcal{H}|/\delta)}{\varepsilon}$$

19

从而对于任何的标记函数 f，任何的分布 $\mathcal{D}$，可实现性假设(即对于某些 $h \in \mathcal{H}$，$L_{\mathcal{D},f}(h)=0$)保证在独立同分布的样本集 S 上(S 的势为 m)至少以 $1-\delta$ 的概率，对于每个ERM假设 h_S，有以下不等式成立

$$L_{\mathcal{D},f}(h_S) \leqslant \varepsilon$$

上述推论告诉我们：对于足够大的 m 来说，由 $\mathrm{ERM}_{\mathcal{H}}$ 规则生成的有限假设类将会概率(置信度为 $1-\delta$)近似(误差上界为 ε)正确。在下一章中，我们正式定义概率近似正确(PAC)学习模型。

2.4 练习

2.1 **多项式匹配的过拟合**：我们看到式(2.3)中定义的预测器导致过拟合。虽然这些预测器看起来非常不自然，但是本练习的目的是展示它能够被多项式函数描述。即证明：

给定一个训练集 $S=\{(x_i, f(x_i))\}_{i=1}^{m}\subseteq(\mathbb{R}^d\times\{0,1\})^m$，存在一个多项式 p_S 使得 $h_S(x)=1$ 的充分必要条件是 $p_S(x)\geq 0$，其中 h_S 的定义同式(2.3)。同时说明利用 ERM 规则在所有多项式阈值类的学习可能导致过拟合。

2.2 设 $\mathcal{H}$ 是定义在领域集 $\mathcal{X}$ 上的二值分类器组成的假设类，$\mathcal{D}$ 为 $\mathcal{X}$ 上的未知分布，f 为 $\mathcal{H}$ 中的目标假设。固定某 $h\in\mathcal{H}$，证明 $L_S(h)$ 关于 $S\sim\mathcal{D}^m$ 的期望值等于 $L_{\mathcal{D},f}(h)$，即

$$\mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}\left[L_S(h)\right]=L_{\mathcal{D},f}(h)$$

[20] 2.3 **轴对称矩形**：平面上的一个轴对称矩形分类器将一个点预测为 1 的充分必要条件是该点落在一个特定的矩形中。形式上，给定实数 $a_1\leq b_1$，$a_2\leq b_2$，定义分类器 $h(a_1, b_1, a_2, b_2)$ 为

$$h_{(a_1,b_1,a_2,b_2)}(x_1,x_2)=\begin{cases}1 & \text{若 } a_1\leq x_1\leq b_1 \text{ 且 } a_2\leq x_2\leq b_2\\ 0 & \text{其他}\end{cases}\tag{2.10}$$

所有平面上的轴对称矩形形成的类定义为：

$$\mathcal{H}_{\text{rec}}=\{h_{(a_1,b_1,a_2,b_2)}: a_1\leq b_1 \text{ 且 } a_2\leq b_2\}$$

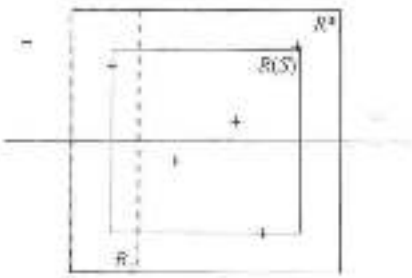


图 2.2 轴对称矩形

注意，这是一个有限假设类，该练习需要证明以下可实现条件。

1) 设 A 为一个算法，其返回训练样本形成的最小矩形，证明 A 是一个基于 ERM 的算法。

2) 证明如果 A 应用于一个样本个数 $\geq\frac{4\log(4/\delta)}{\varepsilon}$ 的训练集，其最小以 $1-\delta$ 的概率输出一个最大误差为 ε 的假设。

提示：固定 $\mathcal{X}$ 上的某分布 $\mathcal{D}$，令 $R^*=R(a_1^*, b_1^*, a_2^*, b_2^*)$ 为一个能够生成标签的矩形，f 为其对应的真实假设。$a_1\geq a_1^*$ 为一个使得矩形 $R_1=R(a_1^*, a_1, a_2^*, b_2^*)$ 的概率质量（关于 $\mathcal{D}$）恰好为 $\varepsilon/4$ 的值。同理，令 b_1，a_2，b_2 的值使得矩形 $R_2=R(b_1, b_1^*, a_2^*, b_2^*)$，$R_3=R(a_1^*, b_1^*, a_2^*, a_2)$，$R_4=R(a_1^*, b_1^*, b_2^*, b_2^*)$ 的概率质量恰好为 $\varepsilon/4$。设 $R(S)$ 为 A 返回的一个矩阵，如图 2.2 所示。

- 证明 $R(S)\subseteq R^*$。
- 证明如果 S 包含矩阵 R_1、R_2、R_3、R_4 中所有的(正)样本，那么 A 返回的假设最大的误差为 ε。
- 对于每个 $i\in\{1, \cdots, 4\}$，求 S 不包含 R_i 中任何样本的概率上界。
- 利用联合界总结论点。

3) 在 d 维空间 $\mathbb{R}^d$ 中重新证明上述几问。

[21] 4) 证明前述算法 A 能在基于 d，$1/\varepsilon$ 和 $\log(1/\delta)$ 的多项式时间内完成。

一般学习模型

本章定义一般学习模型——概率近似正确(PAC)学习模型及其延伸。第 7 章会介绍可学习性的其他概念。

3.1 PAC 学习理论

前面的章节已经给出，在经验风险最小化的规则下，对于一个有限假设类，如果有足够多的训练样本(训练样本的数量独立于潜在的分布，并且独立于标记函数)，那么输出的假设类是概率近似正确的。现在我们定义概率近似正确(PAC)学习。

定义 3.1(PAC 可学习) 若存在一个函数 $m_{\mathcal{H}}:(0,1)^2\to\mathbb{N}$ 和一个学习算法，使得对于任意 $\varepsilon,\delta\in(0,1)$ 和 $\mathcal{X}$ 上的任一分布 $\mathcal{D}$，任意的标号函数 $f:\mathcal{X}\to\{0,1\}$，如果在 $\mathcal{H}$，$\mathcal{D}$，f 下满足可实现的假设，那么当样本数量 $m\geq m_{\mathcal{H}}(\varepsilon,\delta)$ 时，其中样本由分布 $\mathcal{D}$ 独立同分布采样得到并且由函数 f 标记，算法将以不小于 $1-\delta$ 的概率返回一个假设类 h，使该假设类 h 满足 $L_{\mathcal{D},f}(h)\leq\varepsilon$。

概率近似正确可学习性的定义包含两个近似参数。准确度参数 ε 表征输出的分类器和最优分类器之间的距离(这对应于“PAC”的“近似正确”部分)，置信参数 δ 表征分类器达到准确要求的可能性(这对应于“PAC”的“概率”部分)。在我们研究的数据访问模型中，这些近似是不可避免的。由于训练集是随机生成的，因此始终有可能发生样本不提供信息的小概率事件的情况(例如，始终有可能出现这种情况：经过不断采样，训练集恰好只包含一个数据点)。更进一步，即使我们足够幸运得到一个训练样本，它能够很好地代表 $\mathcal{D}$，由于这是一个有限样本，因此 $\mathcal{D}$ 的很多细节依然不能被反映出来。准确度参数 ε，允许让学到的分类器出现小错误。

采样复杂度

函数 $m_{\mathcal{H}}:(0,1)^2\to\mathbb{N}$ 决定学习假设类 $\mathcal{H}$ 的采样复杂度：保证一个概率近似正确解所需的样本数量。采样复杂度是准确度参数 ε 和置信参数 δ 的一个函数。采样复杂度也依赖于假设类 $\mathcal{H}$ 的属性——比如，对于一个有限假设类，我们发现采样复杂度依赖于假设类 $\mathcal{H}$ 势的对数形式。

如果假设类 $\mathcal{H}$ 是 PAC 可学习的，有很多函数 $m_{\mathcal{H}}$ 满足 PAC 可学习定义给出的条件。因此，为了更加精确，我们定义假设类 $\mathcal{H}$ 的采样复杂度为最小函数，即对于任意的 ε 和 δ，$m_{\mathcal{H}}(\varepsilon,\delta)$ 是满足 PAC 可学习条件的最小整数。

回顾上一章介绍的有限假设类的分析及结论，可以重新表述为：

引理 3.2 任一有限假设类是 PAC 可学习的，其采样复杂度满足

$$m_{\mathcal{H}}(\varepsilon,\delta)\leq\left\lceil\frac{\log(|\mathcal{H}|)/\delta}{\varepsilon}\right\rceil$$

也存在无限假设类是可学习的(例如练习 3.3)。随后，我们会给出“决定一个类是否是 PAC 可学习的”不是假设类的势有限还是无限，而是根据一种名叫 VC 维的组合测度。

3.2 更常见的学习模型

前面给出的模型很容易加以推广，可以和更广的学习任务相关联。我们考虑两种形式的泛化：

1. 去掉可实现假设

我们的学习算法在分布为$\mathcal{D}$和标记函数为f上学习成功，是基于可实现假设的前提。对于实际的任务，这种假设可能太严格了（我们真的能保证存在一个矩形区域，它完全决定哪些木瓜是好吃的？）。下一节会给出不可知 PAC 模型，将可实现假设去掉。

2. 学习问题不只是二分类问题

到目前为止，我们讨论的还是给定一个样本预测二值标号（比如好吃还是不好吃）。然而，有许多其他形式的学习任务。例如，假设预测一个实值数字（明天晚上 9 点的气温）或从一个有限标号集里面选出一个标号（例如明天报纸头条的主题）。研究证明我们可以定义各种损失函数来将学习推广。这部分内容会在 3.2.2 节介绍。

23

3.2.1 放宽可实现假设——不可知 PAC 学习

1. 一种更实际的数据生成分布模型

可实现假设要求存在一个$h^* \in \mathcal{H}$使得$\mathbb{P}_{x\sim\mathcal{D}}[h^*(x)=f(x)]=1$。在很多实际问题中，这种假设并不成立。此外，最好不要假设标记完全由我们假定的特征决定（在木瓜的例子中，两个相同颜色相同软硬程度的木瓜味道有可能并不相同）。接下来，我们放宽可实现的假设，把“目标标记函数”替换为更灵活的概念——数据标记生成分布。

从现在起，在形式上，将$\mathcal{D}$定义为$\mathcal{X}\times\mathcal{Y}$上的概率分布，和之前一样，其中$\mathcal{X}$为定义域，$\mathcal{Y}$为标签集合（一般我们认为$\mathcal{Y}=\{0,1\}$）。那$\mathcal{D}$是定义域和标签集上的联合分布。我们可以将该分布分解为两部分：未标记定义域点的概率分布$\mathcal{D}_x$（也称为边缘分布）和每个定义域点标记的条件概率分布$\mathcal{D}((x,y)|x)$。在木瓜的例子中，$\mathcal{D}_x$决定碰到一个木瓜（其颜色和软硬程度落在某一范围内）的概率，条件概率表示x所表示的颜色和软硬程度对应的木瓜好吃的概率。在这种情况下，确实存在相同颜色和软硬程度的木瓜分属不同类的情况。

2. 改进后的经验误差和真实误差

对于$\mathcal{X}\times\mathcal{Y}$上的概率分布$\mathcal{D}$，根据分布$\mathcal{D}$随机生成的带标签的数据点，我们可以测量假设$h$犯错的可能性。我们重新定义预测器$h$的真实误差（或风险）为

$$L_{\mathcal{D}}(h) \stackrel{\text{def}}{=} \mathbb{P}_{(x,y)\sim\mathcal{D}}[h(x)\neq y] \stackrel{\text{def}}{=} \mathcal{D}(\{(x,y):h(x)\neq y\}) \tag{3.1}$$

我们想要找到一个预测器h，使得上述误差最小化。然而，学习器并不知道数据生成分布$\mathcal{D}$。学习器知道的是训练数据S。经验风险依旧是原来定义的形式，即

$$L_S(h) \stackrel{\text{def}}{=} \frac{|\{i\in[m]:h(x_i)\neq y_i\}|}{m}$$

给定S，对于任何的函数$h:\mathcal{X}\to\{0,1\}$，学习器都可以计算$L_S(h)$。注意，$L_S(h)=L_{\mathcal{D}((x_1,y_1),\ldots,(x_m,y_m))}(h)$。

3. 目标

24

我们想要找到假设$h:\mathcal{X}\to\mathcal{Y}$，使得真实风险$L_{\mathcal{D}}(h)$最小。

4. 贝叶斯最优预测器

给定$\mathcal{X}\times\{0,1\}$上的任意概率分布$\mathcal{D}$，将$\mathcal{X}$映射到$\{0,1\}$的最好的预测器是

$$f_D(x)=\begin{cases}1 & 若\quad (\mathbb{P}[y=1|x]\geq 1/2)\\ 0 & 其他\end{cases}$$

很容易验证(练习 3.7)，对于任意的概率分布$\mathcal{D}$，贝叶斯最优分类器 f_D是最优的，其他的分类器 $g:\mathcal{X}\to\{0,1\}$没有更低的错误率。即对任意的分类器 g，$L_D(f_D)\leq L_D(g)$。

可惜，我们不知道概率分布$\mathcal{D}$，不能使用这个最优分类器 f_D。学习器只能获取训练样本。我们现在给出不可知 PAC 可学习的正式定义。很自然地将 PAC 可学习推广到更现实的情况，就如之前讨论的，假定不可实现。

很明显，我们不能期望学习算法给出一个假设，其误差小于最小可能的误差，即贝叶斯分类器的误差。我们在之后会给出证明，如果对数据生成分布不做先验假设，没有算法能够保证找到一个和贝叶斯最优分类器一样好的预测器。我们希望学习算法能够找到一个预测器，其误差和给定假设类中最好预测器的误差相差不大。当然，这种要求的强度取决于假设类的选取。

定义 3.3(不可知 PAC 可学习)　若存在一个函数 $m_{\mathcal{H}}:(0,1)^2\to\mathbb{N}$ 和一个学习算法 A，使得对于任意 ϵ，$\delta\in(0,1)$和$\mathcal{X}\times\mathcal{Y}$上的任一分布$\mathcal{D}$，当样本数量 $m\geq m_{\mathcal{H}}(\epsilon,\delta)$时，其中样本由分布$\mathcal{D}$独立同分布采样得到，算法将以不小于 $1-\delta$ 的概率返回一个假设类 h，使该假设类 h 满足

$$L_{\mathcal{D}}(h)\leq \min_{h'\in\mathcal{H}}L_{\mathcal{D}}(h')+\epsilon$$

很明显，如果满足可实现假设，不可知 PAC 学习和 PAC 学习给出了相同的保证。这样看来，不可知 PAC 学习是 PAC 学习的一种泛化。当不满足可实现的假设时，学习器是不能保证任意小的误差的。然而，在不可知 PAC 学习的定义下，即使和假设类中最好的分类器有些许差距，学习器依然可以认为学习成功。而 PAC 学习要求学习器学到的分类器，其误差达到一个很小的绝对值，而且和假设类可达到的最小误差没有关系。

3.2.2　学习问题建模

接下来，我们将模型进一步拓展，使之能应用到更广的学习任务中。让我们来看一些其他学习任务。

- **多分类**　我们的分类问题不再是二分类问题。比如文本分类问题，我们希望设计一个程序，能够将文档按其主题进行分类(比如，新闻、体育、生物、医学)。对于这类任务，学习器需要根据已有的正确分类的文档，生成一个程序，对新文档给出其相应的主题。我们可以将文档用一系列的特征来表示，特征可以是文档中不同关键词出现的频数，或者其他相关的特征(比如文档的大小及来源)。在这个任务里，标签集是所有可能的主题的集合($\mathcal{Y}$可以是任意大的有限集)。一旦我们定义了定义域和标签集，主体框架的其他部分和木瓜例子看起来很相似：我们的训练样本是有限的序列(特征向量，标签)对，学习器输出一个从定义域到标签集的函数，最后，为了测试学习是否成功，我们可以用分类器给出错误标签的概率来表示。 [25]
- **回归问题**　在这类问题中，希望找到数据的简单模型——数据$\mathcal{X}$和$\mathcal{Y}$之间的关联函数。比如，希望找到一个线性函数，根据超声波检测到的婴儿头围、腹围和股骨长度，能够最好地预测出婴儿出生时的体重。在这里，定义域$\mathcal{X}$是 $\mathbb{R}^3$(三个超声波检测量)的一个子集，标签集$\mathcal{Y}$是实数集(以克为单位的体重)。在此语境下，称为目标集更为合适。这就是我们的训练数据和输出(有限序列(x,y)对，从$\mathcal{X}$到$\mathcal{Y}$的映射函

数)。然而，度量是否成功的标准不再和之前一样。我们可以用期望平方差来评估假设函数h：$\mathcal{X}\to\mathcal{Y}$给出的预测值和真实值之间的差异，即

$$L_{\mathcal{D}}(h)\stackrel{\text{def}}{=}\mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}}(h(x)-y)^2 \tag{3.2}$$

为了满足各式各样的学习任务，我们将学习是否成功的度量进行如下泛化：

广义损失函数

给定任意集合$\mathcal{H}$(相当于我们的假设类或模型)和定义域Z，令ℓ为$\mathcal{H}\times Z$到非负实数的一个映射函数，ℓ：$\mathcal{H}\times Z\to\mathbb{R}_+$，我们称这种函数为损失函数。

需要注意，对于预测问题，有$Z=\mathcal{X}\times\mathcal{Y}$。然而，我们定义的损失函数已经超出了预测任务的范畴，因此可以允许Z可以是任意形式的定义域(比如，在无监督学习问题中(例如第22章)，Z不再是实例空间和标签集的乘积形式)。

我们现在定义损失函数为分类器的期望损失，$h\in\mathcal{H}$，Z上的概率分布为$\mathcal{D}$，即

26

$$L_{\mathcal{D}}(h)\stackrel{\text{def}}{=}\mathop{\mathbb{E}}_{z\sim\mathcal{D}}[\ell(h,z)] \tag{3.3}$$

也就是说，目标z是从分布$\mathcal{D}$上随机采集到的，我们考虑假设类h在目标z的期望损失。与之类似，可以定义经验风险为给定数据集$S=(z_1,\cdots,z_m)\in Z^m$上的期望损失，即

$$L_S(h)\stackrel{\text{def}}{=}\frac{1}{m}\sum_{i=1}^{m}\ell(h,z_i) \tag{3.4}$$

前面的分类和回归问题的损失函数采用的是下述形式：

- **0－1损失**：在这里，随机变量z取值序列对集合$\mathcal{X}\times\mathcal{Y}$，损失函数为

$$\ell_{0\text{-}1}(h,(x,y))\stackrel{\text{def}}{=}\begin{cases}0 & \text{若 } h(x)=y\\ 1 & \text{若 } h(x)\neq y\end{cases}$$

这类损失函数用在二分类或多分类问题中。

需要注意的是，对于随机变量α，取值为$\{0,1\}$，$\mathbb{E}_{\alpha\sim\mathcal{D}}[\alpha]=\mathbb{P}_{\alpha\sim\mathcal{D}}[\alpha=1]$。因此，对于这类损失函数，式(3.1)和式(3.3)给出的$L_{\mathcal{D}}(h)$是一致的。

- **平方损失**：在这里，随机变量z取值序列对集合$\mathcal{X}\times\mathcal{Y}$，损失函数为

$$\ell_{sq}(h,(x,y))\stackrel{\text{def}}{=}(h(x)-y)^2$$

这类损失函数用在回归问题中。

我们会在后续章节看到很多这类损失函数的实例。

总结一下，我们正式定义广义损失函数下的不可知PAC学习。

定义3.4(广义损失函数下的不可知PAC可学习) 对于集合Z和损失函数ℓ：$\mathcal{H}\times Z\to\mathbb{R}_+$，若存在一个函数$m_{\mathcal{H}}$：$(0,1)^2\to\mathbb{N}$和一个学习算法，使得对于任意$\varepsilon,\delta\in(0,1)$，以及$Z$上的任一分布$\mathcal{D}$，当样本数量$m\geq m_{\mathcal{H}}(\varepsilon,\delta)$时，其中样本由分布$\mathcal{D}$独立同分布采样得到，算法将以不小于$1-\delta$的概率返回一个假设类$h$，使该假设类$h$满足

$$L_{\mathcal{D}}(h)\leq\min_{h'\in\mathcal{H}}L_{\mathcal{D}}(h')+\varepsilon$$

其中$L_{\mathcal{D}}(h)=\mathbb{E}_{z\sim\mathcal{D}}[\ell(h,z)]$。

评注(关于可测量性*) 在前面的定义中，对于任意的$h\in\mathcal{H}$，我们将$\ell(h,\cdot)$：$\mathcal{H}\times Z\to\mathbb{R}$视为随机变量，定义$L_{\mathcal{D}}(h)$为该随机变量的期望值。因此，我们需要要求$\ell(h,\cdot)$是可测量的。形式上，我们假定存在一个$Z$的$\sigma$-代数子集，以及其上的概率分布$\mathcal{D}$，$\mathbb{R}_+$的每个分割的原像在这个$\sigma$-代数里。在0－1损失的二分类情况下，$\sigma$-代数在$\mathcal{X}\times\{0,1\}$上，在$\ell$

27

上的假设相当于假设对于任意的h，集合$\{(x,h(x)):x\in\mathcal{X}\}$是$\sigma$-代数。

评注(完全与自主表示学习*)　在前面的定义中，我们要求算法从$\mathcal{H}$中返回一个假设。在某些情况下，$\mathcal{H}$是$\mathcal{H}'$的一个子集，损失函数可以拓展为一个从$\mathcal{H}'\times Z$到实数的函数。在这种情况下，我们允许算法返回一个假设$h'\in\mathcal{H}'$，只要它满足$L_{\mathcal{D}}(h')\leqslant\min_{h\in\mathcal{H}}L_{\mathcal{D}}(h)+\varepsilon$。允许算法从$\mathcal{H}'$返回一个假设，称为自主表示学习，完全学习要求算法必须从$\mathcal{H}$中返回一个假设。自主表示学习有时也称为"不完全学习"，尽管在自主学习中并不存在不恰当的情况。

3.3　小结

这一章定义了主要的正式学习模型——PAC 学习。基本模型基于可实现的假设，不可知 PAC 学习对样本分布不做限制。我们也将 PAC 模型推广到任意损失函数的情况。我们有时将最通用的模型简称为 PAC 学习，省略"不可知"这个前缀，让读者从上下文中体会潜在的损失函数是什么。再次强调最原始的 PAC 模型基于可实现的假设。第 7 章会探讨可学习的其他概念。

3.4　文献评注

对广义损失函数下的不可知 PAC 可学习的基本定义，参考了 Vladimir Vapnik 和 Alexey Chervonenkis 的著作(Vapnik 和 Chervonenkis 1971)。特别是，我们遵循了 Vapnik 关学习的一般设定(Vapnik 1982，Vapnik 1992，Vapnik 1995，Vapnik 1998)。

PAC 学习一词由 Valiant(1984)提出。Valiant 由于提出 PAC 模型，获得了 2010 年的图灵奖。在 Valiant 给出的定义中采样复杂度是关于 $1/\varepsilon$、$1/\delta$、假设类的势的多项式(也可以参考 Kearns 和 Vazirani(1994))。我们将会在第 6 章看到，如果一个问题是 PAC 可学习的，那么采样复杂度是关于 $1/\varepsilon$、$\log(1/\delta)$的多项式。Valiant 的定义还要求算法的运行时间是这些变量的多项式时间。相比之下，我们希望将学习的统计方面和计算方面分割开来。第 8 章会详细介绍计算方面的内容。最后，将不可知 PAC 学习规范化的工作应归功于 Haussler(1992)。

3.5　练习

3.1　**样本复杂度的单调性**：令$\mathcal{H}$为二分类任务的一个假设类。假定$\mathcal{H}$是 PAC 可学习的并且其样本复杂度由$m_{\mathcal{H}}(\cdot,\cdot)$给出。证明$m_{\mathcal{H}}$对其每个参数是单调非增的。即证明给定$\delta\in(0,1)$和$0<\varepsilon_1\leqslant\varepsilon_2<1$，有$m_{\mathcal{H}}(\varepsilon_1,\delta)\geqslant m_{\mathcal{H}}(\varepsilon_2,\delta)$。类似地，证明：给定$\varepsilon\in(0,1)$和$0<\delta_1\leqslant\delta_2<1$，有$m_{\mathcal{H}}(\varepsilon,\delta_1)\geqslant m_{\mathcal{H}}(\varepsilon,\delta_2)$。 [28]

3.2　令$\mathcal{X}$为离散域，$\mathcal{H}_{\text{Singleton}}=\{h_z:z\in\mathcal{X}\}\cup\{h^-\}$，其中对于每个$z\in\mathcal{X}$，$h_z$为一函数，定义为：如果$x=z$则$h_z(x)=1$，如果$x\neq z$则$h_z(x)=0$。$h^-$表示全负假设，即$\forall x\in\mathcal{X}$，有$h^-(x)=0$。在这里可实现假设表示正确假设$f$将定义域的所有样本都标记为负，有一个例外。

3.3　令$\mathcal{X}=\mathbb{R}^2$，$\mathcal{Y}=\{0,1\}$，令$\mathcal{H}$为平面上的同心圆假设类，即$\mathcal{H}=\{h_r:r\in\mathbb{R}_+\}$，其中$h_r(x)=\mathbb{1}_{[\|x\|\leqslant r]}$。证明：$\mathcal{H}$是 PAC 可学习的(假定可实现)，并且样本复杂度的上界为

$$m_{\mathcal{H}}(\varepsilon,\delta)\leqslant\left\lceil\frac{\log(1/\delta)}{\varepsilon}\right\rceil$$

3.4　本练习中，我们考虑布尔连词假设类问题。实例空间为$\mathcal{X}=\{0,1\}^d$，标签集为$\mathcal{Y}=\{0,1\}$。变量$x_1,\cdots,x_d$用布尔函数形式表示为：对某些$i\in[d]$，$f(x)=x_i$，对某些$i\in[d]$，$f(x)=1-x_i$。我们用符号$\bar{x}_i$来表示$1-x_i$。连接可以是任意的积形

式，在布尔逻辑中，积用∧来表示。例如，函数 $h(x)=x_1\cdot(1-x_2)$ 写为 $x_1\wedge\bar{x}_2$。

将假设类表示为 d 维变量的所有连接形式。空连接定义为全正假设(即，对于所有的 x，$h(x)=1$)。连接 $x_1\wedge\bar{x}_1$(相似地，字符与其取反形式相连)是可以存在的，并定义为全负假设(即，对于所有的 x，$h(x)=0$)。我们假定可实现：即，我们假定存在一个布尔连接可以正确生成上述标签。因此，每个样本 $(x, y)\in\mathcal{X}\times\mathcal{Y}$ 包含 d 维变量 $x_1, \cdots, x_d$ 的一种组合形式，以及其真实标签(0 为错误，1 为正确)。

例如，另 $d=3$，假定正确连接方式为 $x_1\wedge\bar{x}_2$。那么训练集 S 可以包括如下实例：

$$((1, 1, 1), 0), ((1, 0, 1), 1), ((0, 1, 0), 0), ((1, 0, 0), 1)$$

证明：d 维变量的所有连接形式组成的假设类是 PAC 可学习的并且给出其样本复杂度的上界。给出一种 ERM 规则下的实现方式，并要求时间复杂度是关于 $d\cdot m$ 的多项式。

3.5 令 $\mathcal{X}$ 为定义域，$\mathcal{D}_1, \mathcal{D}_2, \cdots, \mathcal{D}_m$ 为 $\mathcal{X}$ 上的一系列分布。令 $\mathcal{H}$ 为 $\mathcal{X}$ 上关于二分类器的有限假设类，其中 $f\in\mathcal{H}$。假定现有一样本集 S，有 m 个样本，实例是彼此独立但并非出自同一分布；第 i 个样本来自分布 $\mathcal{D}_i$，y_i 为 $f(x_i)$。令 $\mathcal{D}_m$ 表示平均值，即，$\mathcal{D}_m=(\mathcal{D}_1+\cdots+\mathcal{D}_m)/m$。

29

固定参数 $\epsilon\in(0, 1)$，证明：

$$\mathbb{P}[\exists h\in\mathcal{H} \text{ s.t. } L_{(\mathcal{D}_m, f)}(h)>\epsilon \text{ 且 } L_{(S, f)}(h)=0]\leq|\mathcal{H}|e^{-m\epsilon}$$

提示：使用均值不等式。

3.6 令 $\mathcal{H}$ 为 X 上关于二分类器的假设类。证明：如果 $\mathcal{H}$ 不可知 PAC 可学习，则 $\mathcal{H}$ 同样也是 PAC 可学习的。此外，对于 $\mathcal{H}$，A 是一种成功的不可知 PAC 学习算法，则对于 $\mathcal{H}$，A 也是一种成功的 PAC 学习算法。

3.7 * 贝叶斯最优预测器：证明对任意的概率分布 $\mathcal{D}$，贝叶斯最优预测器 $f_{\mathcal{D}}$ 是最优的，换言之，对从 $\mathcal{X}$ 到 $\{0, 1\}$ 的每个分类器 g，$L_{\mathcal{D}}(f_{\mathcal{D}})\leq L_{\mathcal{D}}(g)$。

3.8 * 在概率分布 $\mathcal{D}$ 下，对所有的 $S\subset(\mathcal{X}\times\{0, 1\})^m$，如果 $L_{\mathcal{D}}(A(S))\leq L_{\mathcal{D}}(B(S))$，则认为在该分布下学习算法 A 优于算法 B。如果对 $\mathcal{X}\times\{0, 1\}$ 上的所有概率分布 $\mathcal{D}$，都满足上式，则认为学习算法 A 优于算法 B。

1) 标签概率预测器是一映射函数，对于定义域的每个点 x 都给定一概率值，$h(x)\in[0, 1]$，即预测标签为 1 的概率值，换言之，给定一个 h 和一个输入 x，x 的标签可以通过模拟成抛硬币来预测，硬币正面朝上的概率为 $h(x)$，当且仅当硬币正面朝上时预测标签为 1。我们定义标签概率预测为一函数 $h: \mathcal{X}\to[0, 1]$，该假设函数 h 在样本 (x, y) 上的损失为 $|h(x)-y|$，也就是 h 的预测值不等于 y 的概率值。如果 h 已确定，返回值介于 $\{0, 1\}$，则 $|h(x)-y|=\mathbb{1}_{[h(x)\neq y]}$。证明：对于 $\mathcal{X}\times\{0, 1\}$ 上的任一数据生成分布 $\mathcal{D}$，贝叶斯最优预测器的风险最小(损失函数 $\ell(h, (x, y))=|h(x)-y|$，对所有可能的标签预测器，包括输出概率值的预测器)。

2) 令 $\mathcal{X}$ 为定义域，$\{0, 1\}$ 为标签集。证明：对 $\mathcal{X}\times\{0, 1\}$ 上的每个概率分布 $\mathcal{D}$，存在一学习算法 $A_{\mathcal{D}}$ 好于其他所有关于 $\mathcal{D}$ 的学习算法。

3) 证明：对任一学习算法 A，存在一个概率分布 $\mathcal{D}$ 和一个学习算法 B，使学习算法 B 不差于学习算法 A。

3.9 考虑 PAC 模型的一种变体，其中有两种样本神谕：根据 $\mathcal{X}$ 上的潜在分布 $\mathcal{D}$，一个生成正样本数据，另一个生成负样本数据。给定一目标函数 $f: \mathcal{X}\to\{0, 1\}$，令 $\mathcal{D}^+$ 表示

$\mathcal{X}^+=\{x\in\mathcal{X}: f(x)=1\}$上的概率分布，对任意的$A\subset\mathcal{X}^+$，有$\mathcal{D}^+(A)=\mathcal{D}(A)/\mathcal{D}(\mathcal{X}^+)$。同样，$\mathcal{D}^-$是由$\mathcal{D}$推导出的$\mathcal{X}^-$上的分布。

PAC可学习在两神谕模型中的定义和标准定义基本相同，不同于标准定义，此处学习算法可以从$\mathcal{D}^+$中获取$m_{\mathcal{H}}^+(\varepsilon,\delta)$个独立同分布的样本，从$\mathcal{D}^-$中获取$m_{\mathcal{H}}^-(\varepsilon,\delta)$个独立同分布的样本。学习算法的目标是输出$h$，约束条件是：以不小于$1-\delta$的概率（在两个训练集上的预测，但可以是学习算法给出的非确定性决策），$L_{\mathcal{D}^+,f}(h)\leqslant\varepsilon$并且$L_{\mathcal{D}^-,f}(h)\leqslant\varepsilon$。

* 1) 证明：当$\mathcal{H}$是PAC可学习的（在标准模型中），那么$\mathcal{H}$在二神谕模型中也是PAC可学习的。

** 2) 定义h^+为always-plus假设，h^-为always-minus假设。现假定$h^+,h^-\in\mathcal{H}$。证明：如果$\mathcal{H}$在两神谕模型中是PAC可学习的，那么$\mathcal{H}$在标准模型中也是PAC可学习的。

学习过程的一致收敛性

我们讨论过的第一个正式学习模型是 PAC 模型。第 2 章已经表明在可实现的假设下，任何有限的假设类都是 PAC 可学习的。在这一章中，我们将开发一个通用的工具——一致收敛，并用它来表明在有一般损失函数的不可知 PAC 模型中，只要距离损失函数是有界的，任何有限类都是可学习的。

4.1 一致收敛是可学习的充分条件

本章讨论的学习条件背后的思想很简单。回想一下，已知一个假设类$\mathcal{H}$，ERM 学习范式工作方式如下：一旦接收一个训练样本 S，学习器评估每一个$\mathcal{H}$中的 h 对于已知样本的损失(或误差)，并且输出$\mathcal{H}$中的一个最小化经验风险的元素。我们希望关于样本 S 的可以被小化经验风险的 h 也是一个关于真实数据概率分布的风险最小化(或者是风险接近最小化)。那么，它足以保证$\mathcal{H}$中的所有元素的经验风险是它们真实风险的一个很好的近似。换句话说，我们需要假设类中所有的假设都是一致的，经验风险将会接近真实风险，表达式如下所示。

定义 4.1(ε-代表性样本) 如果满足下列不等式：

$$\forall h \in \mathcal{H}, |L_S(h) - L_{\mathcal{D}}(h)| \leq \varepsilon$$

一个训练集 S 就称作ε-代表性样本(关于定义域 Z，假设类$\mathcal{H}$，损失函数 ℓ 和分布$\mathcal{D}$)。

下一个简单的引理说明只要样本是$\varepsilon/2$-代表性的，就可以保证 ERM 学习规则返回一个好的假设。

引理 4.2 假设一个训练集 S 是$\varepsilon/2$-代表性的(关于定义域 Z，假设类$\mathcal{H}$，损失函数 ℓ 和分布$\mathcal{D}$)。那么，任何一个 $\mathrm{ERM}_{\mathcal{H}}(S)$的输出，即任意 $h_S \in \operatorname{argmin}_{h\in\mathcal{H}} L_S(h)$都满足

$$L_{\mathcal{D}}(h_S) \leq \min_{h\in\mathcal{H}} L_{\mathcal{D}}(h) + \varepsilon$$

证明 对于所有的 $h \in \mathcal{H}$，

$$L_{\mathcal{D}}(h_S) \leq L_S(h_S) + \frac{\varepsilon}{2} \leq L_S(h) + \frac{\varepsilon}{2} \leq L_{\mathcal{D}}(h) + \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = L_{\mathcal{D}}(h) + \varepsilon$$

其中第一个和第三个不等式是由于 S 是$\varepsilon/2$-代表性的假设(定义 4.1)，第二个不等式成立因为 h_S 是 ERM 预测器的结果。 ■

上面的引理表明为了保证 ERM 规则是一个不可知 PAC 学习器，应该满足至少在概率 $1-\delta$ 下随机选择一个训练集，它将是 ε-代表性训练集。一致收敛条件形式化了这个要求。

定义 4.3(一致收敛) 如果一个假设类$\mathcal{H}$满足如下条件，那么它拥有一致收敛性质(关于定义域 Z 和损失函数 ℓ)：存在一个函数 $m_{\mathcal{H}}^{\mathrm{UC}}: (0,1)^2 \to \mathbb{N}$ 使得对于所有 $\varepsilon, \delta \in (0,1)$ 和在 Z 上的所有概率分布$\mathcal{D}$，如果 S 是从$\mathcal{D}$得到的一个独立同分布的满足 $m \geq m_{\mathcal{H}}^{\mathrm{UC}}(\varepsilon, \delta)$的样本，那么，至少在概率 $1-\delta$ 下，S 是ε-代表性的。

相似于 PAC 学习中样本复杂度的定义，函数 $m_{\mathcal{H}}^{\mathrm{UC}}$ 度量了获得一致收敛性质的(最小)

样本复杂度，即我们需要多少样本来确保至少在概率$1-\delta$下，样本是ε-代表性的。

一致性在这里指的是在定义域中所有可能的概率分布下，用于所有$\mathcal{H}$中的元素，有一个固定的样本大小。

下面的推论直接来自于引理4.2和一致收敛的定义。

推论4.4 如果类$\mathcal{H}$关于函数$m_{\mathcal{H}}^{UC}$有一致收敛的性质，那么这个类是样本复杂度为$m_{\mathcal{H}}(\varepsilon,\delta)\leqslant m_{\mathcal{H}}^{UC}(\varepsilon/2,\delta)$的不可知PAC可学习的。而且，在那种情况下，$\mathrm{ERM}_{\mathcal{H}}$范式是关于$\mathcal{H}$的成功的不可知PAC可学习的。

4.2 有限类是不可知PAC可学习的

鉴于推论4.4，只要我们确定对于一个有限假设类，一致收敛成立，那么每个有限假设类都是不可知PAC可学习的。

为了说明一致收敛成立，类似于第2章的推导，我们用两步的论证。第一步用联合界，第二步用测度集中度不等式。现在我们具体地解释这两步。

固定ε，δ。我们需要找到一个样本大小m可以保证下面的条件成立：对于任何$\mathcal{D}$，至少在概率$1-\delta$下，从$\mathcal{D}$中采样得到的独立同分布的样本的选择$S=(x_1,\dots,x_m)$，对于所有$h\in\mathcal{H}$，$|L_S(h)-L_{\mathcal{D}}(h)|\leqslant\varepsilon$成立。也就是，

$$\mathcal{D}^m(\{S:\forall h\in\mathcal{H},|L_S(h)-L_{\mathcal{D}}(h)|\leqslant\varepsilon\})\geqslant 1-\delta$$

同样，我们需要证明

$$\mathcal{D}^m(\{S:\exists h\in\mathcal{H},|L_S(h)-L_{\mathcal{D}}(h)|>\varepsilon\})<\delta$$

32

写出

$$\{S:\exists h\in\mathcal{H},|L_S(h)-L_{\mathcal{D}}(h)|>\varepsilon\}=\bigcup_{h\in\mathcal{H}}\{S:|L_S(h)-L_{\mathcal{D}}(h)|>\varepsilon\}$$

并且应用联合界(引理2.2)，我们得到：

$$\mathcal{D}^m(\{S:\exists h\in\mathcal{H},|L_S(h)-L_{\mathcal{D}}(h)|>\varepsilon\})\leqslant\sum_{h\in\mathcal{H}}\mathcal{D}^m(\{S:|L_S(h)-L_{\mathcal{D}}(h)|>\varepsilon\})\quad(4.1)$$

第二步是为了证明这个不等式右边的每个被加数都足够小(对于一个充分大的m)。也就是说，我们将要证明对于任意固定的类h(它是在训练集的采样之前要前选择的)，真实风险与经验风险之间的差距$|L_S(h)-L_{\mathcal{D}}(h)|$可能很小。

回想$L_{\mathcal{D}}(h)=\mathbb{E}_{z\sim\mathcal{D}}[\ell(h,z)]$和$L_S(h)=\frac{1}{m}\sum_{i=1}^{m}\ell(h,z_i)$。由于每个$z_i$都从$\mathcal{D}$中独立同分布采样得来，随机变量$\ell(h,z_i)$的期望值是$L_{\mathcal{D}}(h)$。由于期望的线性化，得出$L_S(h)$也是$L_S(h)$的期望值。因此，$|L_{\mathcal{D}}(h)-L_S(h)|$是随机变量$L_S(h)$与它的期望值之间的偏差。因此，我们需要证明$L_S(h)$的度量集中在它的期望值附近。

一个基本的统计事实——大数定理，说明了当m趋近于无穷大时，经验平均值收敛到它们的真实期望。这对于$L_S(h)$也是成立的，由于它是独立同分布的随机变量m的经验平均值。可是，由于大数定理仅仅是一个渐近结果，因此它对于任意给定的有限的样本大小的经验估计误差与其真实值之间的差距没有提供任何信息。

我们将用Hoeffding提出的一个测度集中度不等式来代替，它量化了经验平均值与它们期望值之间的差距。

引理4.5(Hoeffding不等式) 令$\theta_1,\dots,\theta_m$是一个独立同分布的随机变量的序列，假设对于所有的i，$\mathbb{E}[\theta_i]=\mu$而且$\Pr[a\leqslant\theta_i\leqslant b]=1$。那么，对于所有的$\varepsilon>0$

$$\mathbb{P}\left[\left|\frac{1}{m}\sum_{i=1}^{m}\theta_i-\mu\right|>\varepsilon\right]\leqslant 2\exp(-2m\varepsilon^2/(b-a)^2)$$

证明可以在附录B中找到。

回到我们的问题，令θ_i为随机变量$\ell(h,x_i)$。由于h是固定的，而且$x_1,\cdots,x_m$是独立同分布采样得到的，所以$\theta_1,\cdots,\theta_m$也是独立同分布的随机变量。而且，$L_S(h)=\frac{1}{m}\sum_{i=1}^{m}\theta_i$和$L_{\mathcal{D}}(h)=\mu$。让我们进一步假设$\ell$的范围是$[0,1]$，因此$\theta_i\in[0,1]$，因此得到

$$\mathcal{D}^m(\{S:|L_S(h)-L_{\mathcal{D}}(h)|>\varepsilon\})=\mathbb{P}\left[\left|\frac{1}{m}\sum_{i=1}^{m}\theta_i-\mu\right|>\varepsilon\right]\leqslant 2\exp(-2m\varepsilon^2) \tag{4.2}$$

把它和式(4.1)结合，得到

[33]

$$\begin{aligned}\mathcal{D}^m(\{S:\exists h\in\mathcal{H},|L_S(h)-L_{\mathcal{D}}(h)|>\varepsilon\})&\leqslant\sum_{h\in\mathcal{H}}2\exp(-2m\varepsilon^2)\\&=2|\mathcal{H}|\exp(-2m\varepsilon^2)\end{aligned}$$

最后，如果我们选择

$$m\geqslant\frac{\log(2|\mathcal{H}|/\delta)}{2\varepsilon^2}$$

那么

$$\mathcal{D}^m(\{S:\exists h\in\mathcal{H},|L_S(h)-L_{\mathcal{D}}(h)|>\varepsilon\})\leqslant\delta$$

推论4.6 *令$\mathcal{H}$是一个有限假设类，Z是一个定义域，并且令$\ell:\mathcal{H}\times Z\to[0,1]$是一个损失函数。那么，$\mathcal{H}$具有一致收敛性质，而且样本复杂度是*

$$m_{\mathcal{H}}^{UC}(\varepsilon,\delta)\leqslant\left\lceil\frac{\log(2|\mathcal{H}|/\delta)}{2\varepsilon^2}\right\rceil$$

而且，用ERM算法，这个类是不可知PAC可学习的，样本复杂度是

$$m_{\mathcal{H}}(\varepsilon,\delta)\leqslant m_{\mathcal{H}}^{UC}(\varepsilon/2,\delta)\leqslant\left\lceil\frac{2\log(2|\mathcal{H}|/\delta)}{\varepsilon^2}\right\rceil$$

评注（“离散化技巧”） 虽然之前的推论仅仅应用于有限假设类，但有一个简单的技巧可以让我们得到无限假设类的实际样本复杂度的一个很好的估计。考虑一个假设类由d个参数来参数化。比如，令$\mathcal{X}=\mathbb{R}$，$\mathcal{Y}=\{\pm1\}$，而且假设类$\mathcal{H}$是所有形式为$h_\theta(x)=\mathrm{sign}(x-\theta)$的函数，也就是说，每个假设由1个参数来参数化，$\theta\in\mathbb{R}$，而且对于所有大于$\theta$的实例，假设输出1；对于小于$\theta$的实例，假设输出$-1$。这就是一个有无限大小的假设类。然而，如果打算用计算机实际学习这个假设类，我们可能用浮点表示法来维持实数，也就是说64位。结果在实际中，假设类由可以用一个64位浮点数表达的标量集合来参数化。最多有2^{64}个这样的数，因此假设类的实际大小最多是2^{64}。更一般地，如果假设类由d个数来参数化，实际上我们学习到一个最大为2^{64d}的假设类。应用推论4.6，我们得到这样的类的样本复杂度以$\frac{128d-2\log(2/\delta)}{\varepsilon^2}$为界。样本复杂度的这个上界依赖于机器使用的实数的特定表达方式，这是它的缺点。第6章将会介绍一个分析无限大小的假设类的样本复杂度的严格方法。然而，在许多实际情况中，离散化技巧可以用来得到一个样本复杂度的粗略估计。

4.3 小结

如果假设类$\mathcal{H}$一致收敛，那么在大多数情况下$\mathcal{H}$中的假设的经验风险将会如实地表达它

们的真实风险。用 ERM 规则，一致收敛满足不可知 PAC 可学习的条件。我们已经表明有限假设类有一致收敛的性质，因此它也是不可知 PAC 可学习的。 34

4.4　文献评注

满足一致收敛性质的函数的类也叫做 Glivenko-Cantelli 类，这是以 Valery Ivanovich Glivenko 和 Francesco Paolo Cantelli 来命名的，他们在 20 世纪 30 年代首次证明了一致收敛的结果。可以参考 Dudley，Gine & Zinn(1991)。Vapnik 透彻地研究了一致收敛与可学习的关系，参考 Vapnik(1992)，Vapnik(1995)，Vapnik (1998)。实际上，就像我们将要在第 6 章看到的一样，学习理论的基本定理陈述了在二值分类问题中，一致收敛是可学习的充分必要条件。不过在更一般的学习问题中并非如此(参考 Shalev-Shwartz，Shamir，Srebro & Sridharan (2010))。

4.5　练习

4.1　在这个练习中，我们说明在 PAC 可学习的定义中，误差的收敛的(ε, δ)条件实际上非常接近关于平均(或者期望)的一个看起来更加简单的条件。证明：下列两种表述是等价的(对于任意学习算法 A，任意概率分布$\mathcal{D}$，范围在$[0, 1]$的任意损失函数)：

1) 关于所有的 $\varepsilon, \delta>0$，存在 $m(\varepsilon, \delta)$使得$\forall m \geq m(\varepsilon, \delta)$

$$\mathop{\mathbb{P}}_{S\sim\mathcal{D}^m}\left[L_{\mathcal{D}}(A(S)) > \varepsilon\right] < \delta$$

2)

$$\lim_{m\to\infty} \mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}\left[L_{\mathcal{D}}(A(S))\right] = 0$$

($\mathbb{E}_{S\sim\mathcal{D}^m}$表示大小为 m 的样本 S 的期望)。

4.2　**有界损失函数**：在推论 4.6 中，我们假设损失函数的范围是$[0, 1]$。证明：如果损失函数的范围是$[a, b]$，那么样本复杂度满足

$$m_{\mathcal{H}}(\varepsilon,\delta) \leq m_{\mathcal{H}}^{UC}(\varepsilon/2,\delta) \leq \left\lceil \frac{2\log(2|\mathcal{H}|/\delta)(b-a)^2}{\varepsilon^2} \right\rceil$$ 35

偏差与复杂性权衡

在第 2 章中我们看到，除非很小心，否则训练数据会误导学习器导致过拟合。为了克服这一问题，我们将搜索空间限制在某个假设类$\mathcal{H}$下。可以认为，这种假设类反映了学习器关于任务的先验知识，认为假设类$\mathcal{H}$中存在一个假设是低错误率模型。例如，在木瓜品尝问题中，以对于其他水果的经验为基础，我们可能限制在色度-硬度平面的某个矩形区域来预测木瓜的味道。

这样的先验知识对学习的成功是否必要？是否存在通用的学习器(一个没有特定任务先验知识的，并可挑战完成所有学习任务的学习器)？下面我们详细说明这点。一个特定的学习任务由$\mathcal{X}\times\mathcal{Y}$上的一个未知分布$\mathcal{D}$所定义，学习器的目标是寻找一个预测器$h:\mathcal{X}\to\mathcal{Y}$，使得损失$L_{\mathcal{D}}(h)$足够小。我们的问题是，如果$A$收到来自$\mathcal{D}$的$m$个独立同分布的样本，是否存在一个学习算法$A$和一个大小为$m$的训练集，使得对每一个分布$\mathcal{D}$，能以较大的几率输出一个具有较低风险的预测器$h$。

本章第一部分对此问题进行正式讨论。“没有免费的午餐”定理表明，不存在这样的通用学习器。更准确地说，这个定理阐述的是，对二分预测任务，每个学习器都存在一个使得学习失效的分布。如果学习器接收来自同一分布的独立同分布样本，其输出假设可能有$\geq 30\%$的较大风险，我们说学习失败；反之对同一分布，存在另一个学习器能输出一个具有较低风险的假设。换言之，这个定理说明，没有学习器能在所有可学习的任务上都学习成功——即每个学习器都有学习失败的任务，而这些任务对于其他学习器却能成功学习。

因此，解决一个由分布$\mathcal{D}$所定义的特定学习问题时，我们应该具备一些关于分布$\mathcal{D}$的先验知识。其中一类先验知识是限定$\mathcal{D}$来自具体的参数族分布。随后我们将在第 24 章研究这种假设的学习问题。关于$\mathcal{D}$的另一类先验知识是，当定义 PAC 学习模型时，在某个事先指定的假设类$\mathcal{H}$里存在假设h，使得$L_{\mathcal{D}}(h)=0$。关于$\mathcal{D}$一种较宽松的先验知识是假定$\min_{h\in\mathcal{H}}L_{\mathcal{D}}(h)$较小。一定程度上，这种弱假设是使用不可知 PAC 模型的先决条件，其中我们要求输出假设的风险不会超过$\min_{h\in\mathcal{H}}L_{\mathcal{D}}(h)$。

[36]

在本章第二部分我们采用一个假设类作为将先验知识标准化的方式，来研究其利弊性。我们将 ERM 算法在假设类$\mathcal{H}$上的误差分解为两部分。第一部分反映了先验知识的质量，由假设类具有的最小风险$\min_{h\in\mathcal{H}}L_{\mathcal{D}}(h)$所刻画。这部分也称为逼近误差，或是叫算法从$\mathcal{H}$选择一个假设所产生的偏差。第二部分是由过拟合引起的误差，取决于假设类的大小或复杂度，也称为估计误差。这两项意味着，在一个较为复杂的假设(可以减小偏差但会增加过拟合的风险)和一个简单的假设(可能会增大偏差但可以降低过拟合的风险)选择之间存在着一个权衡。

5.1 “没有免费的午餐”定理

在这部分我们证明不存在通用的学习器。通过证明没有学习器能在所有的任务上学习成功，我们将具体定理阐述如下：

定理 5.1(没有免费的午餐) 对实例空间$\mathcal{X}$上 0-1 损失的二分任务，令A表示任意的学

习算法，样本大小 m 表示小于 $|\mathcal{X}|/2$ 的任意数。则在 $\mathcal{X}\times\{0,1\}$ 上存在一个分布 $\mathcal{D}$，使得：

1. 存在一个函数 $f:\mathcal{X}\to\{0,1\}$ 满足 $L_{\mathcal{D}}(f)=0$。

2. 在样本集 $S\sim\mathcal{D}^m$ 上，以至少 $\frac{1}{7}$ 的概率满足 $L_{\mathcal{D}}(A(S))\geq\frac{1}{8}$。

这个定理陈述的是，对于每个学习器，都存在一个任务使其失败，即便这个任务能够被另一个学习器成功学习。实际上，一个平凡的学习器能在此类情况下学习成功，它将是关于假设类 $\mathcal{H}=\{f\}$ 的一个 ERM 学习器；或更广泛而言，其 ERM 是对任何包含 f 且样本大小满足 $m>8\log(7|\mathcal{H}|/6)$（见推论 2.3）的有限假设类而言的。

证明 令 C 是大小为 $2m$ 的集合 $\mathcal{X}$ 子集。直观的证据是，任何只观测到空间 C 中一半实例的学习算法，都不具有信息量来反映 C 中剩余实例的标签。因此，存在一个"事实"，即在 C 中未观测到的样本上，目标函数 f 赋的标签与 $A(S)$ 预测的标签不一致。 ■

注意，从 C 到 $\{0,1\}$ 有 $T=2^{2m}$ 个函数。这些函数表示为 $f_1,\cdots,f_T$。对每个这样的函数，令 $\mathcal{D}_i$ 表示定义在 $C\times\{0,1\}$ 上的分布：

$$\mathcal{D}_i(\{(x,y)\})=\begin{cases}1/|C| & \text{如果} \quad y=f_i(x)\\ 0 & \text{否则}\end{cases}$$

37

也就是，选择一对 (x,y)，标签 y 刚好对应 f_i 真实标签的概率是 $1/|C|$，而 $y\neq f_i(x)$ 的概率是 0。显然，$L_{\mathcal{D}_i}(f_i)=0$。

我们将证明，对每一个学习算法 A，其接收到来自 $C\times\{0,1\}$ 的 m 大小样本集，返回一个函数 $A(S):C\to\{0,1\}$，满足：

$$\max_{i\in[T]}\mathop{\mathbb{E}}_{S\sim\mathcal{D}_i^m}[L_{\mathcal{D}_i}(A(S))]\geq 1/4 \tag{5.1}$$

显然，这意味着对每一个学习算法 A'，其接收到来自 $C\times\{0,1\}$ 的 m 大小样本集，存在一个函数 $f:\mathcal{X}\to\{0,1\}$ 和 $\mathcal{X}\times\{0,1\}$ 上的一个分布 $\mathcal{D}$，使得 $L_{\mathcal{D}}(f)=0$ 且

$$\mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}[L_{\mathcal{D}}(A'(S))]\geq 1/4 \tag{5.2}$$

容易证明，以上是满足 $\mathbb{P}[L_{\mathcal{D}}(A'(S))\geq 1/8]\geq 1/7$ 的充分条件，这也是我们所要证明的（见练习 5.1）。

我们转为证明式(5.1)成立。对于来自 C 的 m 个样本，有 $k=(2m)^m$ 种可能的序列。将这些序列表示为 $S_1,\cdots,S_k$。同时，如果 $S_j=(x_1,\cdots,x_m)$，我们用 S_j^i 表示包含由函数 f_i 给实例 S_j 贴标签的序列，即 $S_j^i=((x_1,f_i(x_1)),\cdots,(x_m,f_i(x_m)))$。若分布是 $\mathcal{D}_i$，则 A 可能接收到的训练样本集是 $S_1^i,\cdots,S_k^i$，并且所有的训练集被采样的概率相等。因此

$$\mathop{\mathbb{E}}_{S\sim\mathcal{D}_i^m}[L_{\mathcal{D}_i}(A(S))]=\frac{1}{k}\sum_{j=1}^{k}L_{\mathcal{D}_i}(A(S_j^i)) \tag{5.3}$$

根据"最大值"大于"平均值"以及"平均值"大于"最小值"的事实，有

$$\begin{aligned}\max_{i\in[T]}\frac{1}{k}\sum_{j=1}^{k}L_{\mathcal{D}_i}(A(S_j^i))&\geq\frac{1}{T}\sum_{i=1}^{T}\frac{1}{k}\sum_{j=1}^{k}L_{\mathcal{D}_i}(A(S_j^i))\\&=\frac{1}{k}\sum_{j=1}^{k}\frac{1}{T}\sum_{i=1}^{T}L_{\mathcal{D}_i}(A(S_j^i))\\&\geq\min_{j\in[k]}\frac{1}{T}\sum_{i=1}^{T}L_{\mathcal{D}_i}(A(S_j^i))\end{aligned} \tag{5.4}$$

接下来，固定某个 $j\in[k]$，设 $S_j=(x_1,\cdots,x_m)$ 并令 $v_1,\cdots,v_p$ 是 C 中未出现在 S_j 的

38 样本。显然，$p \geq m$。因此，对每个函数 $h: C \to \{0, 1\}$ 和每个 i 有

$$L_{\mathcal{D}_i}(h) = \frac{1}{2m}\sum_{x \in C} \mathbb{1}_{[h(x) \neq f_i(x)]} \geq \frac{1}{2m}\sum_{r=1}^{p} \mathbb{1}_{[h(v_r) \neq f_i(v_r)]}$$

$$\geq \frac{1}{2p}\sum_{r=1}^{p} \mathbb{1}_{[h(v_r) \neq f_i(v_r)]} \tag{5.5}$$

因此，

$$\frac{1}{T}\sum_{i=1}^{T} L_{\mathcal{D}_i}(A(S_i)) \geq \frac{1}{T}\sum_{i=1}^{T} \frac{1}{2p}\sum_{r=1}^{p} \mathbb{1}_{[A(S_i)(v_r) \neq f_i(v_r)]}$$

$$= \frac{1}{2p}\sum_{r=1}^{p} \frac{1}{T}\sum_{i=1}^{T} \mathbb{1}_{[A(S_i)(v_r) \neq f_i(v_r)]}$$

$$\geq \frac{1}{2} \cdot \min_{r \in [p]} \frac{1}{T}\sum_{i=1}^{T} \mathbb{1}_{[A(S_i)(v_r) \neq f_i(v_r)]} \tag{5.6}$$

下面固定某个 $r \in [p]$。我们可以将 $f_1, \cdots, f_T$ 中所有的函数分为 $T/2$ 对不相交的函数，其中，对于每对 (f_i, f_j)，当且仅当 $c = v_r$ 时，对每个 $c \in C$ 满足 $f_i(c) \neq f_j(c)$。因为对于每对函数，一定有 $S_i = S_j$，且

$$\mathbb{1}_{[A(S_i)(v_r) \neq f_i(v_r)]} + \mathbb{1}_{[A(S_j)(v_r) \neq f_j(v_r)]} = 1$$

使得

$$\frac{1}{T}\sum_{i=1}^{T} \mathbb{1}_{[A(S_i)(v_r) \neq f_i(v_r)]} = \frac{1}{2}$$

结合上式和式(5.6)、式(5.4)以及式(5.3)，可知式(5.1)成立，证明完成。 ■

"没有免费的午餐"和先验知识

"没有免费的午餐"结论与对先验知识的必要与否有什么联系？考虑关于假设类$\mathcal{H}$上的一个 ERM 预测器，这个假设类由从 $\mathcal{X}$ 到$\{0, 1\}$的所有映射函数 f 构成。这个类代表先验知识的缺失：从域到标签集上的每个函数都能看成是一个好的候选。根据"没有免费的午餐"定理，从假设类$\mathcal{H}$中选择输出假设的任意算法，尤其是 ERM 预测器，都存在着某个任务使其学习失败。因此，下面的推论给出了形式化阐述，这个类不是 PAC 可学习的：

推论 5.2 令$\mathcal{X}$为一个无限定义域集，$\mathcal{H}$为从$\mathcal{X}$到$\{0, 1\}$上的所有映射集，则$\mathcal{H}$不是 PAC 可学习的。

39 **证明** 采用反证法，假设这个类是可学习的。选 $\varepsilon < 1/8$ 和 $\delta < 1/7$。由 PAC 可学习性的定义，一定存在学习算法 A 和一个整数 $m = m(\varepsilon, \delta)$，使得对于任意关于$\mathcal{X} \times \{0, 1\}$的生成数据分布，若对于某个函数 $f: \mathcal{X} \to \{0, 1\}$，有 $L_{\mathcal{D}}(f) = 0$，则当 A 应用于由$\mathcal{D}$产生的大小 m、独立同分布样本集 S 上，$L_{\mathcal{D}}(A(S)) < \varepsilon$ 以大于$(1-\delta)$的概率成立。然而，应用"没有免费的午餐"定理，由于$|\mathcal{X}| > 2m$，对每个学习算法(尤其是对算法 A)，存在一个分布$\mathcal{D}$使得以大于$1/7 > \delta$的概率，$L_{\mathcal{D}}(A(S)) > 1/8 > \varepsilon$ 成立，与假设矛盾。 ■

如何避免这样的失败？我们可以利用对于特定学习任务的先验知识，结合"没有免费的午餐"定理，来预见并脱离这样的困境，从而避免学习任务时会导致失败的分布。这样的先验知识可以通过限制假设类来表示。

但是我们如何选择一个好的假设类？一方面，我们希望这个类包含完全无误差(在 PAC 背景下)的假设，或至少包含的假设所能达到的最小误差实际上很小(在不可知背景下)。另

一方面，我们已经看到，不能只简单地选择最丰富的假设类——给定的域上所有函数的类。关于权衡的讨论见下一小节。

5.2 误差分解

为了回答本章的问题，我们将一个 $\mathrm{ERM}_{\mathcal{H}}$ 预测器的误差分解为两部分。令 h_S 为一个 $\mathrm{ERM}_{\mathcal{H}}$ 假设。则写作

$$L_{\mathcal{D}}(h_S) = \varepsilon_{app} + \varepsilon_{est} \quad 其中：\varepsilon_{app} = \min_{h \in \mathcal{H}} L_{\mathcal{D}}(h), \quad \varepsilon_{est} = L_{\mathcal{D}}(h_S) - \varepsilon_{app} \tag{5.7}$$

- **逼近误差**：假设类里预测器所取得的最小风险。这一项刻画由于限制到一个具体假设类所引起的风险，即所产生的归纳偏置。逼近误差不依赖于样本大小，取决于所选择的假设类。扩大假设类可以减小逼近误差。

在可实现性的假设下，逼近误差是零。然而，在不可知情况下，逼近误差可能很大[⊖]。

- **估计误差**：逼近误差与 ERM 预测器误差之间的差异。估计误差的产生是因为：经验风险（即训练误差）只是真实风险的一个估计，所以最小化经验风险预测器只是最小化真实风险预测器的一个估计。

预测器的估计好坏取决于样本集大小和假设类的大小或复杂度。如前所示，对一个有效假设类，ε_{est} 随 $\mathcal{H}$（以对数方式）递增，随 m 递减。我们可以将 $\mathcal{H}$ 的大小作为其复杂度的一种衡量。在后面的章节我们将定义一些其他的假设类复杂度衡量指标。 40

由于目标是将总风险最小化，因此我们面临着一个权衡，称为**偏差-复杂度权衡**。一方面，选择一个丰富的假设类作为 $\mathcal{H}$ 会导致过拟合，使得逼近误差减小的同时估计误差增大。另一方面，选择一个较小的假设类作为 $\mathcal{H}$，会导致估计误差减小的同时逼近误差增大，换言之会欠拟合。当然，关于 $\mathcal{H}$ 的一个好的选择是，假设类只包含一个分类器——贝叶斯最优分类器。但是贝叶斯最优分类器依赖于潜在分布 $\mathcal{D}$，而 $\mathcal{D}$ 却是未知的（事实上，事先知道分布就无需进行学习）。

学习理论研究的是我们如何使得 $\mathcal{H}$ 丰富的同时依然保持合理的估计误差。在很多情况中，经验研究着重于对某个域设计一个好的假设类。这里，“好”的假设类意味着其逼近误差不会过大。意思就是，虽然我们不是专家且不知道如何构造最优分类器，但是对面临的问题有一些先验知识，确保能够设计一个假设类。这个假设类的逼近误差和估计误差都不会太大。回到木瓜的例子，我们不知道如何根据木瓜的颜色和硬度预测其成熟的程度，但是我们知道颜色-硬度的二维矩形区域可能是一个好的预测器。

5.3 小结

“没有免费的午餐”定理说明不存在通用的学习器。每个学习器都有其特定的任务，为了学习成功要采用一些关于任务的先验知识。目前为止，通过限定输出假设为所选假设类中的一员，我们对先验知识进行建模。当选择这个假设类时，我们面临着一个权衡，是选择一个较大或是较复杂的假设类，保证有较小的逼近误差；还是选择一个有更多限制的假设类，保证较小的估计误差？关于估计误差的更多性质在下一章讨论。第7章将讨论其他表达先验知识的方法。

⊖ 实际中，逼近误差总是包含贝叶斯最优预测器（见第1章）的误差，由于模型中存在真实世界的不确定性，最小误差预测器总会产生不可避免的误差。有时在一些文献中，逼近误差定义的不是 $\min_{h \in \mathcal{H}} L_{\mathcal{D}}(h)$，而是指通过贝叶斯最优的预测器的误差，即 $\min_{h \in \mathcal{H}} L_{\mathcal{D}}(h) - \varepsilon_{Bayes}$。

5.4 文献评注

Wolpert & Macready (1997)从优化角度证明了几个"没有免费的午餐"定理，但是这与我们证明的理论有一些不同。本章所证明的定理与下一章的 VC 理论中的下确界有着密切关系。

5.5 练习

5.1 证明：式(5.2)是 $\mathbb{P}[L_{\mathcal{D}}(A(S))\geq 1/8]\geq 1/7$ 的充分条件。

提示：令 θ 是[0，1]区间上的随机变量，其期望满足 $\mathbb{E}[\theta]\geq 1/4$，根据引理 B.1，证明：$\mathbb{E}[\theta]\geq 1/8\geq 1/7$。

41

5.2 假如要求你设计一个学习算法来预测病人是否会有患心脏病的风险。算法所得的病人相关特征信息包括血压(BP)、体重指数(BMI)、年龄(A)、体育锻炼的频度(P)和收入(I)。你可以在两种算法中选择一种，一个算法选取由特征 BP 和 BMI 构成的二维空间矩形，另一个算法选取由以上所有五个维度特征所构成的超立方体。

1) 解释每个方案的优点和缺点；

2) 解释可提供标签的训练样本数对选择方案的影响。

5.3 证明：对正整数 $k\geq 2$，若 $|\mathcal{X}|\geq 2m$，则我们可将"没有免费的午餐"定理中的下界替换为 $\frac{k-1}{2k}-\frac{1}{2}-\frac{1}{2k}$。换言之，令 A 是针对二分任务的学习算法，令 m 是小于 $|\mathcal{X}|/k$ 的任意数，表示训练样本集大小。则存在 $\mathcal{X}\times\{0,1\}$ 上的一个分布，使得

- 存在一个函数 $f:\mathcal{X}\to\{0,1\}$，有 $L_{\mathcal{D}}(f)=0$
- $\mathbb{E}_{S\sim\mathcal{D}^m}[L_{\mathcal{D}}(A(S))]\geq\frac{1}{2}-\frac{1}{2k}$

42

VC 维

在之前的章节中，我们将 $\mathrm{ERM}_{\mathcal{H}}$ 规则产生的误差分解为逼近误差和估计误差。其中，逼近误差依赖于我们的先验知识（反映在对假设类$\mathcal{H}$的选择）与潜在的未知分布是否吻合。而 PAC 可学习性的定义要求，对于所有分布估计误差均有一个一致的界。

我们现在的目标是找到那些 PAC 可学习的假设类$\mathcal{H}$，并且精确地刻画学习给定假设类的样本复杂度。目前，我们已经知道有限的类是可学习的，而由所有函数组成的类（在一个无限规模的域）不是可学习的。那么，到底是什么使一个类可学习而使另一个类不可学习？无限规模的类是否可学习？如果可以，那是什么决定这种类的样本复杂度？

我们通过说明有些无限类确实可学习来开始这一章，基于此，说明了假设类的有限性不是可学习性的必要条件。之后，我们引入一种对可学习的假设类族非常新鲜的描述来建立那些采用 0—1 损失的二值分类问题。这个描述最早由 Vladimir Vapnik 和 Alexey Chervonenkis 于 1970 年发现，并借助于 VC 维的概念。我们将正式地定义 VC 维，给出一些 VC 维的例子，之后叙述统计机器学习理论的基本定理，该定理整合了可学习性、VC 维、ERM 规则以及一致收敛性的概念。

6.1 无限的类也可学习

在第 4 章中我们看到有限的类是可学习的，实际上，这种情况下，假设类的样本复杂度上界由假设类大小的对数决定。为了说明假设类的大小不是一个可用于描述样本复杂度的特征，我们首先举一个简单的例子，说明某些无限大小的假设类也是可学习的。

例 6.1 令$\mathcal{H}$是实线上阈值函数构成的集合，即，$\mathcal{H}=\{h_a : a\in\mathbb{R}\}$，其中，$h_a:\mathbb{R}\to\{0,1\}$是一个函数，使得 $h_a(x)=\mathbb{I}_{[x<a]}$。即如果 $x<a$，$\mathbb{I}_{[x<a]}$为 1，否则为 0。显然，$\mathcal{H}$是无限大小的。虽然如此，下面的引理表明$\mathcal{H}$在 PAC 模型下采用 ERM 算法是可学习的。 ◀

引理 6.1 *令$\mathcal{H}$为如之前定义的阈值函数类。那么，$\mathcal{H}$在采用 ERM 规则时是 PAC 可学习的，其样本复杂度 $m_{\mathcal{H}}(\epsilon,\delta)\le\lceil\log(2/\delta)/\epsilon\rceil$。*

证明 令 a^* 为阈值，则相应的假设 $h^*(x)=\mathbb{I}_{[x<a^*]}$可以使得 $L_{\mathcal{D}}(h^*)=0$。令$\mathcal{D}_x$ 为域$\mathcal{X}$上的边缘分布，令 $a_0<a^*<a_1$ 使得：

$$\mathop{\mathbb{P}}_{x\sim\mathcal{D}_x}[x\in(a_0,a^*)]=\mathop{\mathbb{P}}_{x\sim\mathcal{D}_x}[x\in(a^*,a_1)]=\epsilon$$

（如果$\mathcal{D}_x(-\infty,a^*)\le\epsilon$ 则令 $a_0=-\infty$。对于 a_1 也采用类似处理。）给定一个训练集 S，令 $b_0=\max\{x:(x,1)\in S\}$，$b_1=\min\{x:(x,0)\in S\}$（若在 S 中无正样本，令$b_0=-\infty$。同样，如果在 S 中无负样本，令 $b_1=\infty$）。令 b_S 为与 ERM 假设 h_S 相关的阈值，即 $b_S\in(b_0,b_1)$。因此，$L_{\mathcal{D}}(h_S)\le\epsilon$ 成立的充分条件是 $b_0\ge a_0$ 与 $b_1\le a_1$ 同时成立。换言之

$$\mathop{\mathbb{P}}_{S\sim\mathcal{D}^m}[L_{\mathcal{D}}(h_S)>\varepsilon]\leqslant \mathop{\mathbb{P}}_{S\sim\mathcal{D}^m}[b_0<a_0 \vee b_1>a_1]$$

采用联合界，上式变为

$$\mathop{\mathbb{P}}_{S\sim\mathcal{D}^m}[L_{\mathcal{D}}(h_S)>\varepsilon]\leqslant \mathop{\mathbb{P}}_{S\sim\mathcal{D}^m}[b_0<a_0]+\mathop{\mathbb{P}}_{S\sim\mathcal{D}^m}[b_1>a_1] \tag{6.1}$$

当且仅当 S 中所有样本均不在区间 (a_0, a^*) 中时，会出现 $b_0<a_0$ 的情况，将这种情况出现的概率定义为 ε，即

$$\mathop{\mathbb{P}}_{S\sim\mathcal{D}^m}[b_0<a_0]=\mathop{\mathbb{P}}_{S\sim\mathcal{D}^m}[\forall(x,y)\in S, x\notin(a_0,a^*)]=(1-\varepsilon)^m\leqslant e^{-\varepsilon m}$$

由于我们假定了 $m>\log(2/\delta)/\varepsilon$，则上式至多为 $\delta/2$。同样，可以容易得到 $\mathbb{P}_{S\sim\mathcal{D}^m}[b_1>a_1]\leqslant\delta/2$。联立式(6.1)，引理得证。 ■

6.2 VC 维概述

我们可以看到，虽然假设类$\mathcal{H}$的有限性是可学习性的充分条件，但它并不是一个必要条件。之后会看到，一个叫做 VC 维的性质能正确描述假设类的可学习性。为了引出 VC 维的定义，我们首先回顾“没有免费的午餐”定理(定理 5.1)及其证明。当时，我们已经说明如果不对假设类加以限制，任何学习算法总会遇到表现很差的情况，与此同时，总是有学习算法在此情况下表现很好。为了达到这样的情况，可以使用一个有限集 $C\subset\mathcal{X}$ 并且考虑在 C 上元素的分布族，其中每个分布由从 C 到$\{0,1\}$“真实的”目标函数产生。为了使任何学习算法失败，可以从由 C 到$\{0,1\}$所有可能的函数构成的集合中选择一个目标函数。

44

考虑到一个假设类$\mathcal{H}$的 PAC 可学习性，需构建一些分布使得某些假设 $h\in\mathcal{H}$ 达到零风险。由于考虑的是限制在 C 上元素的分布，我们需要学习完竟假设类$\mathcal{H}$在 C 上表现如何，于是引出了下面的定义。

定义 6.2(限制$\mathcal{H}$在 C 上) 令$\mathcal{H}$是从$\mathcal{X}$到$\{0,1\}$的一个函数类，并且令 $C=\{c_1,\cdots,c_m\}\subset\mathcal{X}$。限制$\mathcal{H}$在 C 上就是由来自$\mathcal{H}$从 C 到$\{0,1\}$的函数构成的集合，即

$$\mathcal{H}_C=\{(h(c_1),\cdots,h(c_m)):h\in\mathcal{H}\}$$

其中，我们将每个从 C 到$\{0,1\}$的函数表示为形如$\{0,1\}^{|C|}$的向量。

如果限制$\mathcal{H}$在 C 上是从 C 到$\{0,1\}$的所有函数的集合，那么我们称$\mathcal{H}$打散了集合 C。正式地：

定义 6.3(打散) 如果限制$\mathcal{H}$在 C 上是从 C 到$\{0,1\}$的所有函数的集合，则假设类$\mathcal{H}$打散了有限集 $C\subset\mathcal{X}$，此时 $|\mathcal{H}_C|=2^{|C|}$。

例 6.2 令$\mathcal{H}$是 $\mathbb{R}$ 上的阈值函数类，取一个集合 $C=\{c_1\}$。此时，如果取 $a=c_1+1$ 则有 $h_a(c_1)=1$，如果取 $a=c_1-1$，则有 $h_a(c_1)=0$。因此，$\mathcal{H}_C$ 是从 C 到$\{0,1\}$的所有函数的集合，故而$\mathcal{H}$打散了 C。此时如果取 $C=\{c_1,c_2\}$，其中 $c_1\leqslant c_2$，则不存在 $h\in\mathcal{H}$ 完成 C 到$\{0,1\}$所有可能的映射，因此任何阈值函数如果给 c_1 的标签是 0，则给 c_2 的标签一定也是 0。因此，$\mathcal{H}_C$ 没有包括所有从 C 到$\{0,1\}$的函数，故而此时 C 没有被$\mathcal{H}$打散。 ◀

回到之前所述“没有免费的午餐”定理(定理 5.1)中情况的构建上，我们可以看到当一些集合 C 被$\mathcal{H}$打散时，构建的分布便不局限于$\mathcal{H}$，因为可以根据从 C 到$\{0,1\}$的任意目标函数构建在 C 上的分布，并且同时保证可实现假设依然成立，这直接得到了：

推论 6.4 令$\mathcal{H}$是从$\mathcal{X}$到$\{0,1\}$的函数构成的假设类，令 m 是训练集的大小。假定存

在大小为 $2m$ 的集合 $C\subseteq\mathcal{X}$ 能被 $\mathcal{H}$ 打散。那么，对于任何学习算法 A，在 $\mathcal{X}\times\{0,1\}$ 上必定存在一个分布 $\mathcal{D}$ 和预测器 $h\in\mathcal{H}$ 使得 $L_{\mathcal{D}}(h)=0$，但是对于所选样本集 $S\sim\mathcal{D}^m$ 至少以 1/7 的概率有 $L_{\mathcal{D}}(A(S))\geq 1/8$。

推论 6.4 说明了如果假设类 $\mathcal{H}$ 打散了大小为 $2m$ 的集合 C，那么我们将无法通过 m 个样本来学习 $\mathcal{H}$。直观地讲，如果一个集合 C 被 $\mathcal{H}$ 打散，而我们只能得到 C 中一半样本构成的集合，那么这些样本的标签对于我们预测 C 中剩余样本标签的价值来说没有产生帮助，因为剩余样本的标签的每一种可能的组合都可以在假设类 $\mathcal{H}$ 中找到某些假设与之对应。从哲学上讲，如果有人可以解释每个现象，他的解释本身就是毫无意义的。 45

基于上述事实，现在我们可以引出 VC 维的定义：

定义 6.5(VC 维) 假设类 $\mathcal{H}$ 的 VC 维，记为 $\mathrm{VCdim}(\mathcal{H})$，是 $\mathcal{H}$ 可以打散的最大集合 $C\subseteq\mathcal{X}$ 的大小。如果 $\mathcal{H}$ 可以打散任意大的集合，我们说 $\mathcal{H}$ 的 VC 维是无穷的。

因此我们得到推论 6.4 的一个直接结果如下：

定理 6.6 令 $\mathcal{H}$ 是无穷 VC 维的假设类，那么 $\mathcal{H}$ 不是 PAC 可学习的。

证明 由于 $\mathcal{H}$ 有无穷 VC 维，故而对于任意 m 大小的训练集，总存在一个大小为 $2m$ 且被打散的集合，结合推论 6.4 定理得证。 ■

在本章后面，我们将会看到上述定理的逆命题也成立，即有限的 VC 维可以保证可学习性。因此，VC 维可以描述 PAC 可学习性。在深入研究理论之前，我们先看几个例子。

6.3 实例

这一小节将对几个假设类进行 VC 维的计算。为了证明 $\mathrm{VCdim}(\mathcal{H})=d$，需要证明：

1. 存在大小为 d 的集合 C 可以被 $\mathcal{H}$ 打散。
2. 每个大小为 $d+1$ 集合 C 都不能被 $\mathcal{H}$ 打散。

6.3.1 阈值函数

令 $\mathcal{H}$ 是 $\mathbb{R}$ 上的阈值函数类。在例 6.2 中，我们说明了任意形如 $C=\{c_1\}$ 的集合，都可以被 $\mathcal{H}$ 打散，因此 $\mathrm{VCdim}(\mathcal{H})\geq 1$。我们同时说明了任意形如 $C=\{c_1,c_2\}(c_1\leq c_2)$ 的集合，$\mathcal{H}$ 无法打散。因此，我们可以确定 $\mathrm{VCdim}(\mathcal{H})=1$。

6.3.2 区间

令 $\mathcal{H}$ 是 $\mathbb{R}$ 上的区间类，即 $\mathcal{H}=\{h_{a,b}:a,b\in\mathbb{R},a\leq b\}$，其中 $h_{a,b}:\mathbb{R}\to\{0,1\}$ 是一个函数，使得 $h_{a,b}(x)=\mathbf{1}_{x\in[a,b]}$。取集合 $C=\{1,2\}$，则 $\mathcal{H}$ 可以打散 C(请确保你知道原因)。因此 $\mathrm{VCdim}(\mathcal{H})\geq 2$。现在，取一个任意的集合 $C=\{c_1,c_2,c_3\}$ 并不失一般性地假定 $c_1\leq c_2\leq c_3$。那么，标签 $(1,0,1)$ 无法由一个区间获得，因此 $\mathcal{H}$ 没有打散这样的集合 C。所以我们可以得到 $\mathrm{VCdim}(\mathcal{H})=2$。

6.3.3 平行于轴的矩形

令 $\mathcal{H}$ 是平行于轴的矩形类，即

$$\mathcal{H}=\{h_{(a_1,a_2,b_1,b_2)}:a_1\leq a_2 \quad \text{且} \quad b_1\leq b_2\}$$ 46

其中

$$h_{(a_1,a_2,b_1,b_2)}(x_1,x_2)=\begin{cases}1 & 若\ a_1\leqslant x_1\leqslant a_2,b_1\leqslant x_2\leqslant b_2\\ 0 & 其他\end{cases}\tag{6.2}$$

接下来将证明 $\mathrm{VCdim}(\mathcal{H})=4$。为了证明这一点，我们需要找到一个由 4 个点组成的集合而且可以被$\mathcal{H}$打散，并且说明不存在由 5 个点组成的集合可以由$\mathcal{H}$打散。找到一个由 4 个点组成的集合可以被打散是容易的(见图 6.1)。现在，考虑任意 5 点构成的集合$C\subset\mathbb{R}^2$。在C中，取一个最左边的点(其第一个坐标在C中最小)，一个最右边的点(其第一个坐标最大)，一个最下面的点(第二个坐标最小)，以及一个最上面的点(第二个坐标最大)。不失一般性，记为$C=\{c_1,\cdots,c_5\}$，并令c_5为未被取到的点。现在，定义一种标签结果为$(1,1,1,1,0)$，则不可能由任何平行于轴的矩形得到这种标签结果。事实上，这样的矩形必须包含$c_1,\cdots,c_4$，但是在这种情况下，其必同时包含c_5，因为这个点的坐标在所选点的坐标区间内。因此，C没有被$\mathcal{H}$打散，故而$\mathrm{VCdim}(\mathcal{H})=4$。

图 6.1　左图：4 个点被平行于轴的矩形打散。右图：任何平行于轴的矩形都不能在将其他点标记为 1 的情况下将c_5标记为 0

6.3.4　有限类

令$\mathcal{H}$是一个有限类。那么，很显然对于任意集合C，有$|\mathcal{H}_C|\leqslant|\mathcal{H}|$，因此如果$|\mathcal{H}|\leqslant 2^{|C|}$，$C$将不会被打散。这意味着$\mathrm{VCdim}(\mathcal{H})\leqslant\log_2(|\mathcal{H}|)$。这就是说，有限 VC 维的 PAC 可学习性比有限类的 PAC 可学习性更为一般，我们将会在下一节详述。注意，有限类$\mathcal{H}$的 VC 维有可能比$\log_2(|\mathcal{H}|)$小得多。例如，令$\mathcal{X}=\{1,\cdots,k\}$，$k$为整数，考虑阈值函数类(如例 6.2 定义)，那么，$|\mathcal{H}|=k$但是$\mathrm{VCdim}(\mathcal{H})=1$。由于$k$可以变得任意大，故而$\log_2(|\mathcal{H}|)$和$\mathrm{VCdim}(\mathcal{H})$的差距也可以变得任意大。

6.3.5　VC 维与参数个数

在之前的例子中，我们发现 VC 维碰巧与定义假设类的参数的个数相等。虽然往往是这个情况，但要注意这不一定是永远正确的。下面考虑一种情况，设域为$\mathcal{X}=\mathbb{R}$，假设类为$\mathcal{H}=\{h_\theta:\theta\in\mathbb{R}\}$，其中$h_\theta:\mathcal{X}\to\{0,1\}$由$h_\theta(x)=\lceil 0.5\sin(\theta x)\rceil$定义。易证得$\mathrm{VCdim}(\mathcal{H})=\infty$，即，对于每个$d$，都能找到$d$个点被假设类$\mathcal{H}$打散(见练习 6.8)。

47

6.4　PAC 学习的基本定理

我们已经说明了 VC 维无限的类不是可学习的，其逆命题也是正确的，因此可以得到下述统计学习理论的基本定理：

定理 6.7(统计学习的基本定理)　令$\mathcal{H}$是一个由从$\mathcal{X}$到$\{0,1\}$的映射函数构成的假设类，且令损失函数为 0-1 损失。那么，下述陈述等价：

1. $\mathcal{H}$有一致收敛性。

2. 任何ERM规则都是对于$\mathcal{H}$成功的不可知PAC学习器。
3. $\mathcal{H}$是不可知PAC可学习的。
4. $\mathcal{H}$是PAC可学习的。
5. 任何ERM规则都是对于$\mathcal{H}$成功的PAC学习器。
6. $\mathcal{H}$的VC维有限。

该定理的证明将在下一小节给出。

VC维不仅可用于描述PAC可学习性，还可以决定样本复杂度。

定理6.8(统计学习的基本定理——定量形式) 令$\mathcal{H}$是一个由从$\mathcal{X}$到$\{0,1\}$的映射函数构成的假设类，且令损失函数为0—1损失。假定$\mathrm{VCdim}(\mathcal{H})=d<\infty$，那么，存在绝对常数$C_1$、$C_2$使得：

1. $\mathcal{H}$有一致收敛性，若其样本复杂度满足：

$$C_1\frac{d+\log(1/\delta)}{\varepsilon^2}\leqslant m_{\mathcal{H}}^{UC}(\varepsilon,\delta)\leqslant C_2\frac{d+\log(1/\delta)}{\varepsilon^2}$$

2. $\mathcal{H}$是不可知PAC可学习的，若其样本复杂度满足：

$$C_1\frac{d+\log(1/\delta)}{\varepsilon^2}\leqslant m_{\mathcal{H}}(\varepsilon,\delta)\leqslant C_2\frac{d+\log(1/\delta)}{\varepsilon^2}$$

3. $\mathcal{H}$是PAC可学习的，若其样本复杂度满足：

$$C_1\frac{d+\log(1/\delta)}{\varepsilon}\leqslant m_{\mathcal{H}}(\varepsilon,\delta)\leqslant C_2\frac{d\log(1/\varepsilon)+\log(1/\delta)}{\varepsilon}$$

上述定理的证明将会在第28章给出。

评注 我们所述的基本定理是针对二分类问题的。对于其他学习问题，如采用绝对值损失或者平方损失的回归问题也能得到类似结果。然而，该定理并不是对于所有的学习问题都成立。特别地，有些情况下即使满足一致收敛性不成立，也可能有可学习性成立(我们将在第13章和练习6.2中举例说明)。更进一步地，在某些情况下，ERM规则不成立但是可学习性可通过其他学习规则达到。 [48]

6.5 定理6.7的证明

在第4章中我们已经看到了1→2。这意味着2→3，3→4以及2→5是显然的。由“没有免费午餐”定理可知，4→6和5→6也是易得的。定理证明的难点在于6→1。证明过程主要基于下述两个论断：

- 如果$\mathrm{VCdim}(\mathcal{H})=d$，即使$\mathcal{H}$是无限的，当将其限制在一个有限集合$C\subset\mathcal{X}$时，其“有效”规模$|\mathcal{H}_C|$只有$O(|C|^d)$。即，$\mathcal{H}_C$随着$|C|$的增长呈现按多项式方式增长而不是按指数方式增长。该论断与Sauer引理有关，亦被Shelah和Perles提出和独立证明。之后在6.5.1节会正式地给出该论断。
- 5.4节说明了有限的假设类有着一致收敛性。之后在6.5.2节中会将这一论断推广并说明当假设类有一个“小的有效规模”时其一致收敛性成立。“小的有效规模”指的是$|\mathcal{H}_C|$随着$|C|$按多项式方式增长。

6.5.1 Sauer引理及生长函数

我们通过将假设类$\mathcal{H}$限制在由有限实例组成的集合上定义了打散的概念。所谓的生长函数就是度量$\mathcal{H}$在由m个样本构成的集合上的最大“有效”规模。正式地：

定义 6.9(生长函数) 令$\mathcal{H}$是假设类。$\mathcal{H}$的生长函数，记作$\tau_{\mathcal{H}}(m)$：$\mathbb{N}\to\mathbb{N}$，定义为：

$$\tau_{\mathcal{H}}(m)=\max_{C\subset\mathcal{X},\ |C|=m}|\mathcal{H}_C|$$

即，$\tau_{\mathcal{H}}(m)$就是从大小为m的集合C到$\{0,1\}$不同函数的个数，其可由限制$\mathcal{H}$在C上获得。

显然，如果$\mathrm{VCdim}(\mathcal{H})=d$，那么对于任意$m\leq d$，有$\tau_{\mathcal{H}}(m)=2^m$。在这种情况下，$\mathcal{H}$诱导了从$C$到$\{0,1\}$所有的函数。下述由Sauer、Shelah、Perles独立提出的美妙的引理，表明了当m变得比VC维大时，生长函数随着m按多项式方式增长而不是按指数方式增长。

引理 6.10(Sauer Shelah Perles) 令$\mathcal{H}$是一个假设类，且$\mathrm{VCdim}(\mathcal{H})\leq d<\infty$。那么对于所有的$m$，$\tau_{\mathcal{H}}(m)\leq\sum_{i=0}^{d}\binom{m}{i}$，特别地，如果$m>d-1$，那么$\tau_{\mathcal{H}}(m)\leq(em/d)^d$。

Sauer 引理的证明*

为了证明该引理，需要证明下述更严格的论断：对于任意$C=\{c_1,\cdots,c_m\}$，有

49

$$\forall\mathcal{H},|\mathcal{H}_C|\leq|\{B\subseteq C:\mathcal{H}\text{ 打散 }B\}| \tag{6.3}$$

式(6.3)对于证明引理是充分的，因为如果$\mathrm{VCdim}(\mathcal{H})\leq d$那么将不存在规模大于$d$且被$\mathcal{H}$打散的集合，因此

$$|\{B\subseteq C:\mathcal{H}\text{打散 }B\}|\leq\sum_{i=0}^{d}\binom{m}{i}$$

当$m>d-1$时上式右项至多为$(em/d)^d$(见附录A中引理A.5)。

因此现只须证明式(6.3)成立，我们采用归纳法，对于$m=1$的情况，无论$\mathcal{H}$是何种形式，式(6.3)两边或者都等于1或者都等于2(我们认为空集总是可被$\mathcal{H}$打散的)。下面假定对于集合规模$k<m$式(6.3)成立，现证明集合规模为m的情况。固定$\mathcal{H}$以及$C=\{c_1,\cdots,c_m\}$。另外，记$C'=\{c_2,\cdots,c_m\}$，并定义如下两个集合：

$$Y_0=\{(y_2,\cdots,y_m):(0,y_2,\cdots,y_m)\in\mathcal{H}_C\vee(1,y_2,\cdots,y_m)\in\mathcal{H}_C\}$$

$$Y_1=\{(y_2,\cdots,y_m):(0,y_2,\cdots,y_m)\in\mathcal{H}_C\wedge(1,y_2,\cdots,y_m)\in\mathcal{H}_C\}$$

很容易验证$|\mathcal{H}_C|=|Y_0|+|Y_1|$。另外，由于$Y_0=\mathcal{H}_{C'}$，考虑$\mathcal{H}$在$C'$的归纳假设(应用于$\mathcal{H}$和$C'$)，我们有

$$|Y_0|=|\mathcal{H}_{C'}|\leq|\{B\subseteq C':\mathcal{H}\text{ 打散 }B\}|=|\{B\subseteq C:c_1\notin B\wedge\mathcal{H}\text{ 打散 }B\}|$$

接下来，定义$\mathcal{H}'\subseteq\mathcal{H}$为：

$$\mathcal{H}'=\{h\in\mathcal{H}:\exists h'\in\mathcal{H}\ \text{s.t.}\ (1,h'(c_2),h'(c_3),\cdots,h'(c_m))=(h(c_1),h(c_2),\cdots,h(c_m))\}$$

即，$\mathcal{H}'$包含了那些在C'上适用但在c_1上不适用的假设。在这样的定义下，显然地，如果$\mathcal{H}'$打散了集合$B\subseteq C'$，那么它将同时打散集合$B\cup\{c_1\}$，反之亦成立。将$Y_1=\mathcal{H}'_{C'}$与上述事实联立，并考虑$\mathcal{H}'$在C'上的归纳假设，可得

$$\begin{aligned}|Y_1|=|\mathcal{H}'_{C'}|&\leq|\{B\subseteq C':\mathcal{H}'\text{ 打散 }B\}|=|\{B\subseteq C':\mathcal{H}'\text{ 打散 }B\cup\{c_1\}\}|\\&=|\{B\subseteq C:c_1\in B\wedge\mathcal{H}'\text{ 打散 }B\}|\leq|\{B\subseteq C:c_1\in B\wedge\mathcal{H}\text{ 打散 }B\}|\end{aligned}$$

综上所述，有

$$\begin{aligned}|\mathcal{H}_C|&=|Y_0|+|Y_1|\\&\leq|\{B\subseteq C:c_1\notin B\wedge\mathcal{H}\text{ 打散 }B\}|+|\{B\subseteq C:c_1\in B\wedge\mathcal{H}\text{ 打散 }B\}|\\&=|\{B\subseteq C:\mathcal{H}\text{ 打散 }B\}|\end{aligned}$$

■

6.5.2 有小的有效规模的类的一致收敛性

这一节中，我们要证明如果$\mathcal{H}$有小的有效规模，那么$\mathcal{H}$有一致收敛性，正式的表述如下： [50]

定理 6.11 令$\mathcal{H}$是一个类，令$\tau_{\mathcal{H}}$为其生长函数。那么，对于每个$\mathcal{D}$以及每个$\delta\in(0,1)$，对于任意$S\sim\mathcal{D}^m$，都以至少$1-\delta$的概率有下式成立：

$$|L_{\mathcal{D}}(h)-L_S(h)|\leqslant\frac{4+\sqrt{\log(\tau_{\mathcal{H}}(2m))}}{\delta\sqrt{2m}}$$

在证明该定理之前，我们首先完成定理 6.7 的证明。

定理 6.7 的证明

欲证有限 VC 维的假设类有着一致收敛性，须证：

$$m_{\mathcal{H}}^{UC}(\varepsilon,\delta)\leqslant 4\frac{16d}{(\delta\varepsilon)^2}\log\left(\frac{16d}{(\delta\varepsilon)^2}\right)+\frac{16d\log(2e/d)}{(\delta\varepsilon)^2}$$

由 Sauer 定理可得，对于$m>d$，有$\tau_{\mathcal{H}}(2m)\leqslant(2em/d)^d$，将该式与定理 6.11 联立可得以至少$1-\delta$的概率下式成立

$$|L_S(h)-L_{\mathcal{D}}(h)|\leqslant\frac{4+\sqrt{d\log(2em/d)}}{\delta\sqrt{2m}}$$

为了简化表达，假定$\sqrt{d\log(2em/d)}\geqslant 4$，因此有

$$|L_S(h)-L_{\mathcal{D}}(h)|\leqslant\frac{1}{\delta}\sqrt{\frac{2d\log(2em/d)}{m}}$$

为了保证上式至多为ε，我们需要得到

$$m\geqslant\frac{2d\log(m)}{(\delta\varepsilon)^2}+\frac{2d\log(2e/d)}{(\delta\varepsilon)^2}$$

标准的代数操作(见附录 A 中引理 A.2)表明上式成立的一个充分条件是

$$m\geqslant 4\frac{2d}{(\delta\varepsilon)^2}\log\left(\frac{2d}{(\delta\varepsilon)^2}\right)+\frac{4d\log(2e/d)}{(\delta\varepsilon)^2}$$ ■

评注 我们在定理 6.7 的证明中给出的$m_{\mathcal{H}}^{UC}$的上界可能不是最严格的。在第 28 章中将会给出一个满足定理 6.8 的界更严格的分析。

定理 6.11 的证明

我们由证明下式开始

$$\mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}\left[\sup_{h\in\mathcal{H}}|L_{\mathcal{D}}(h)-L_S(h)|\right]\leqslant\frac{4+\sqrt{\log(\tau_{\mathcal{H}}(2m))}}{\sqrt{2m}}\tag{6.4}$$

由于随机变量$\sup_{h\in\mathcal{H}}|L_{\mathcal{D}}(h)-L_S(h)|$是非负的，因此该定理可直接由马尔可夫不等式推出(见 B.1 节)。 [51]

为了给出式(6.4)左半部分的界，我们首先注意到对于每个$h\in\mathcal{H}$，我们可以重写$L_{\mathcal{D}}(h)=\mathbb{E}_{S'\sim\mathcal{D}^m}[L_{S'}(h)]$，其中$S'=z'_1,\ldots,z'_m$为新增的独立同分布样本。因此

$$\mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}\left[\sup_{h\in\mathcal{H}}|L_{\mathcal{D}}(h)-L_S(h)|\right]=\mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}\left[\sup_{h\in\mathcal{H}}\left|\mathop{\mathbb{E}}_{S'\sim\mathcal{D}^m}L_{S'}(h)-L_S(h)\right|\right]$$

利用三角不等式的一般形式可得

$$\left|\mathop{\mathbb{E}}_{S'\sim\mathcal{D}^m}[L_{S'}(h)-L_S(h)]\right|\leqslant\mathop{\mathbb{E}}_{S'\sim\mathcal{D}^m}|L_{S'}(h)-L_S(h)|$$

考虑到期望的上界小于上界的期望，故而

$$\sup_{h\in\mathcal{H}}\mathbb{E}_{S\sim\mathcal{D}^m}\left[L_{\mathcal{D}}(h)-L_S(h)\right]\leq\mathbb{E}_{S\sim\mathcal{D}^m}\left[\sup_{h\in\mathcal{H}}|L_{\mathcal{D}}(h)-L_S(h)|\right]$$

之前的两个不等式也可由 Jensen 不等式得到。联立上述各式，有

$$\mathbb{E}_{S\sim\mathcal{D}^m}\left[\sup_{h\in\mathcal{H}}|L_{\mathcal{D}}(h)-L_S(h)|\right]\leq\mathbb{E}_{S,S'\sim\mathcal{D}^m}\left[\sup_{h\in\mathcal{H}}|L_{S'}(h)-L_S(h)|\right]$$

$$=\mathbb{E}_{S,S'\sim\mathcal{D}^m}\left[\sup_{h\in\mathcal{H}}\frac{1}{m}\left|\sum_{i=1}^{m}(\ell(h,z_i')-\ell(h,z_i))\right|\right]\qquad(6.5)$$

等式右边的期望与两个独立同分布的样本 $S=z_1,\cdots,z_m$ 和 $S'=z_1',\cdots,z_m'$ 有关。由于所有 $2m$ 个向量都是独立同分布的，因此我们将随机变量 z_i 换名为 z_i' 不会产生任何变化，这样之后，式(6.5)中的项 $(\ell(h,z_i')-\ell(h,z_i))$ 将变为项 $-(\ell(h,z_i')-\ell(h,z_i))$。因此，对于每个 $\boldsymbol{\sigma}\in\{\pm1\}^m$，式(6.5)等价于：

$$\mathbb{E}_{S,S'\sim\mathcal{D}^m}\left[\sup_{h\in\mathcal{H}}\frac{1}{m}\left|\sum_{i=1}^{m}\sigma_i(\ell(h,z_i')-\ell(h,z_i))\right|\right]$$

由于该式对于每个 $\boldsymbol{\sigma}\in\{\pm1\}^m$ 成立，如果我们随机地对 $\boldsymbol{\sigma}$ 的每个分量按照在 $\{\pm1\}$ 上的均匀分布来采样，记作 U_m，该式也是成立的。因此，式(6.5)也等价于

$$\mathbb{E}_{\boldsymbol{\sigma}\sim U_m^m}\mathbb{E}_{S,S'\sim\mathcal{D}^m}\left[\sup_{h\in\mathcal{H}}\frac{1}{m}\left|\sum_{i=1}^{m}\sigma_i(\ell(h,z_i')-\ell(h,z_i))\right|\right]$$

由于期望是线性的，该式亦等价于

$$\mathbb{E}_{S,S'\sim\mathcal{D}^m}\mathbb{E}_{\boldsymbol{\sigma}\sim U_m^m}\left[\sup_{h\in\mathcal{H}}\frac{1}{m}\left|\sum_{i=1}^{m}\sigma_i(\ell(h,z_i')-\ell(h,z_i))\right|\right]$$

[32]　接下来，固定 S 与 S'，令 C 为在 S 与 S' 中同时出现的实例集。那么，我们可以取只在 $h\in\mathcal{H}_C$ 的上确界，因此

$$\mathbb{E}_{\boldsymbol{\sigma}\sim U_m^m}\left[\sup_{h\in\mathcal{H}}\frac{1}{m}\left|\sum_{i=1}^{m}\sigma_i(\ell(h,z_i')-\ell(h,z_i))\right|\right]$$

$$=\mathbb{E}_{\boldsymbol{\sigma}\sim U_m^m}\left[\max_{h\in\mathcal{H}_C}\frac{1}{m}\left|\sum_{i=1}^{m}\sigma_i(\ell(h,z_i')-\ell(h,z_i))\right|\right]$$

固定某个 $h\in\mathcal{H}_C$，并记 $\theta_h=\frac{1}{m}\sum_{i=1}^{m}\sigma_i(\ell(h,z_i')-\ell(h,z_i))$。由于 $\mathbb{E}[\theta_h]=0$ 且 θ_h 是在 $[-1,1]$ 取值的独立变量的平均值，因此由 Hoeffding 不等式，对于每个 $\rho>0$

$$\Pr[|\theta_h|>\rho]\leq2\exp(-2m\rho^2)$$

利用在 $h\in\mathcal{H}_C$ 上的联合界，可以得到，对于任意 $\rho>0$

$$\Pr\left[\max_{h\in\mathcal{H}_C}|\theta_h|>\rho\right]\leq2|\mathcal{H}_C|\exp(-2m\rho^2)$$

最后，由附录 A 中引理 A.2 可知，下式表明

$$\mathbb{E}\left[\max_{h\in\mathcal{H}_C}|\theta_h|\right]\leq\frac{4+\sqrt{\log(|\mathcal{H}_C|)}}{\sqrt{2m}}$$

联立上述各式与 $\tau_{\mathcal{H}}$ 的定义，可得

$$\mathbb{E}_{S\sim\mathcal{D}^m}\left[\sup_{h\in\mathcal{H}}|L_{\mathcal{D}}(h)-L_S(h)|\right]\leq\frac{4+\sqrt{\log(\tau_{\mathcal{H}}(2m))}}{\sqrt{2m}}$$

6.6　小结

学习理论的基本定理采用 VC 维的概念描述了二分类问题的 PAC 可学习性。一个类的

VC维是一种复杂的性质，描述了该类可打散样本的最大规模。基本定理表明了一个类是PAC可学习的当且仅当它的VC维是有限的，并且给出了PAC学习所需的样本复杂度。该定理也表明如果一个问题确实是可学习，那么该问题具有一致收敛性并且采用ERM规则可以实现可学习。

6.7 文献评注

VC维的定义以及它与可学习性和一致收敛性的联系源于Vapnik和Chervonenkis(1971)的工作。VC维与PAC可学习性的定义之间的联系源于Blumer、Ehrenfeucht、Haussler，以及Warmuth(1989)的工作。

自VC维的概念提出以来，陆续出现了一些对它的推广。例如，fat-shattering维度描述了一些回归问题的可学习性(Kearns，Schapire & Sellie 1994；Alon，Ben-David，Cesa-Bianchi & Haussler 1997；Bartlett，Long & Williamson 1994；Anthony & Bartlet 1999)，纳塔拉詹维度描述了一些多类学习问题的可学习性(Natarajan 1989)。然而，对于一般的情况，可学习性和一致收敛性不是等价的。详见(Shalev-Shwartz，Shamir，Srebro & Sridharan 2010；Daniely，Sabato，Ben-David & Shalev-Shwartz 2011)。

Sauer引理是Sauer为了解决Erdos问题而证明的(Sauer 1972)。Shelah和Perles证明了该引理对于Shelah的稳定模型理论很有用(Shelah 1972)。Gil Kalai还曾提到[⊖]，后来Benjy Weiss请Perles针对遍历理论证明该引理，Perles忘记了自己曾经证明过，于是又证明了一次。Vapnik和Chervonenkis在统计学习理论中也对该引理给出了证明。

6.8 练习

6.1 请说明下述关于VC维的单调性：对于任意两个假设类，如果$\mathcal{H}'\subseteq\mathcal{H}$，那么$\mathrm{VCdim}(\mathcal{H}')\leqslant\mathrm{VCdim}(\mathcal{H})$。

6.2 给定某个有限域$\mathcal{X}$，以及一个数$k\leqslant|\mathcal{X}|$，请指出下列几类的VC维并证明：

1) $\mathcal{H}^{\mathcal{X}}_{=k}=\{h\in\{0,1\}^{\mathcal{X}}:|\{x:h(x)=1\}|=k\}$：即所有将$\mathcal{X}$的$k$个元素赋值为1的函数组成的集合。

2) $\mathcal{H}_{\leqslant k\text{ or }\geqslant k}=\{h\in\{0,1\}^{\mathcal{X}}:|\{x:h(x)=1\}|\leqslant k$ 或 $|\{x:h(x)=0\}|\leqslant k\}$。

6.3 令$\mathcal{X}$是一个布尔超立方$\{0,1\}^n$。对于集合$I\subseteq\{1,2,\cdots,n\}$，我们定义一个奇偶函数$h_I$如下。对于一个二值向量$x=(x_1,x_2,\cdots,x_n)\in\{0,1\}^n$

$$h_I(x)=\left(\sum_{i\in I}x_i\right)\bmod 2$$

(即h_I计算了I的字节的奇偶性。)请问所有这种奇偶函数组成的假设类$(\mathcal{H}_{\text{parity}}=\{h_I:I\subseteq\{1,2,\cdots,n\}\})$的VC维是多少？

6.4 我们证明Sauer引理是通过证明对于每个有限VC维d的假设类$\mathcal{H}$，以及每个域内子集A有下式成立：$|\mathcal{H}_A|\leqslant|\{B\subseteq A:\mathcal{H}\text{ 打散 }B\}|\leqslant\sum_{i=0}^{d}\binom{|A|}{i}$

试说明上式中存在两个不等号严格成立的情况(即$\leqslant$可以换为$<$)，也存在两个不等号可以换为等号的情况。并证明所有这4种组合的情况。

6.5 在$\mathbb{R}^d$上平行于坐标轴矩形的VC维：令$\mathcal{H}^d_{\text{rec}}$是在$\mathbb{R}^d$上平行于坐标轴矩形类。我们已经说明了$\mathrm{VCdim}(\mathcal{H}^2_{\text{rec}})=4$，请证明对于一般的情况，$\mathrm{VCdim}(\mathcal{H}^d_{\text{rec}})=2d$。

⊖ http://gilkalai.wordpress.com/2008/09/28/extremal-combinatorics-iii-some-basic-theorems。

6.6 **布尔合取的 VC 维**：令$\mathcal{H}_{\text{conj}}^{d}$为在变量$x_1, \cdots x_d(d\geq 2)$上的布尔合取类。我们已经知道这个类是有限的，因此是(不可知)PAC 可学习的。现在我们来计算 $\text{VCdim}(\mathcal{H}_{\text{conj}}^{d})$：

1) 说明$|\mathcal{H}_{\text{conj}}^{d}|\leq 3^d+1$。

2) 推导 $\text{VCdim}(\mathcal{H})\leq d\log 3$。

54

3) 说明$\mathcal{H}_{\text{conj}}^{d}$打散了单位向量构成的集合$\{e_i: i\leq d\}$。

** 4) 说明 $\text{VCdim}(\mathcal{H}_{\text{conj}}^{d})\leq d$。

提示：假定上式不成立，即存在一个集合 $C=\{c_1, \cdots, c_{d+1}\}$能被$\mathcal{H}_{\text{conj}}^{d}$打散。令$h_1, \cdots, h_{d+1}$是$\mathcal{H}_{\text{conj}}^{d}$中的假设且满足：

$$\forall i,j\in[d+1], h_i(c_j)=\begin{cases}0 & i=j\\ 1 & \text{其他}\end{cases}$$

对于每个$i\in[d+1]$，h_i(更准确地说是与h_i有关的合取)包含了某个文字ℓ_i，该文字对于每个$j\neq i$在c_j上为假而在c_i上为真。根据鸽巢原理，必定存在一对$i<j\leq d+1$使得ℓ_i和ℓ_j使用了同样的变量x_k，而后基于这个事实与考虑h_i，h_j的合取时会得到矛盾。

5) 考虑在$\{0,1\}^d$上的单调布尔合取类$\mathcal{H}_{\text{mon-conj}}^{d}$。在这里单调性指的是合取式不包含负值。如在$\mathcal{H}_{\text{conj}}^{d}$中，空的合取可以解释为所有值均为正的假设。我们将所有值均为负的假设h^-加入到$\mathcal{H}_{\text{mon-conj}}^{d}$中。请说明 $\text{VCdim}(\mathcal{H}_{\text{mon-conj}}^{d})=d$。

6.7 我们已经说明对于有限的假设类$\mathcal{H}$，$\text{VCdim}(\mathcal{H})\leq\lfloor\log(|\mathcal{H}|)\rfloor$。然而这只是一个上界。一个类的 VC 维其实可以变得更小：

1) 找到一个例子说明定义在实区间$\mathcal{X}=[0,1]$上函数组成的假设类$\mathcal{H}$是无限的，但是其 $\text{VCdim}(\mathcal{H})=1$。

2) 给出一个例子，定义在实区间$\mathcal{X}=[0,1]$上函数组成的假设类$\mathcal{H}$是有限的，但是其 $\text{VCdim}(\mathcal{H})=\lfloor\log_2(|\mathcal{H}|)\rfloor$。

*6.8 我们经常会发现一个假设类的 VC 维等于定义假设类所需的参数个数(或可以此为界)。例如，如果$\mathcal{H}$是在$\mathbb{R}^2$上平行于坐标轴的矩形类，那么 $\text{VCdim}(\mathcal{H})=2d$，等于用来在$\mathbb{R}^2$上定义矩形所需的参数个数。这里给出一个例子来说明上述规律并不总是正确的。我们将会看到一个假设类可能是很复杂的甚至是不可学习的，但是其相关的参数却很少。

考虑域$\mathcal{X}=\mathbb{R}$，假设类为

$$\mathcal{H}=\{x\mapsto\lceil\sin(\theta x)\rceil: \theta\in\mathbb{R}\}$$

(这里，我们取$\lceil -1\rceil=0$)，证明 $\text{VCdim}(\mathcal{H})=\infty$。

提示：有很多方式可以证明待证结论。比如可以考虑下述引理：如果$0.x_1x_2x_3\cdots$是$x\in(0,1)$的二进制展开，那么对于任意自然数m，$\lceil\sin(2^m\pi x)\rceil=(1-x_m)$，继而$\exists k\geq m$ s.t. $x_k=1$。

6.9 令$\mathcal{H}$是带符号的区间类，即

$$\mathcal{H}=\{h_{a,b,s}: a\leq b, s\in\{-1,1\}\}$$

其中

$$h_{a,b,s}(x)=\begin{cases}s & \text{若 } x\in[a,b]\\ -s & \text{若 } x\notin[a,b]\end{cases}$$

请计算 $\text{VCdim}(\mathcal{H})$。

6.10 令$\mathcal{H}$是从$\mathcal{X}$到$\{0,1\}$的函数类。

1) 证明：如果 $\text{VCdim}(\mathcal{H})\geq d$，对于任意$d$，对于某个在$\mathcal{X}\times\{0,1\}$上的概率分布

$\mathcal{D}$，对于每个样本规模 m，有

$$\mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}\left[L_{\mathcal{D}}(A(S))\right]\geqslant \min_{h\in\mathcal{H}} L_{\mathcal{D}}(h)+\frac{d-m}{2d}$$

提示：利用第 5 章中的习题 5.3。

2) 证明：对于每个 PAC 可学习的 $\mathcal{H}$，有 $\mathrm{VCdim}(\mathcal{H})<\infty$（注意到这意味着定理 6.7 中 3→6 成立）。

6.11 组合 VC 维：令 $\mathcal{H}_1$，…，$\mathcal{H}_r$ 是在某个固定域 $\mathcal{X}$ 上的假设类，令 $d=\max_i \mathrm{VCdim}(\mathcal{H}_i)$，为了简化分析，假定 $d\geqslant 3$。

1) 证明：

$$\mathrm{VCdim}\left(\bigcup_{i=1}^{r}\mathcal{H}_i\right)\leqslant 4d\log(2d)+2\log(r)$$

提示：取一个由 k 个样本构成的集合并假定该集合可由组合的类打散。因此，组合的类可以在这些样本上产生所有 2^k 种可能的标签方式。利用 Sauer 引理可得组合类不会产生比 rk^d 种更多的标签方式，因此就得到了 $2^k\leqslant rk^d$。之后利用引理 A.2。

*2) 证明对于 $r=2$ 下式成立：

$$\mathrm{VCdim}(\mathcal{H}_1\cup\mathcal{H}_2)\leqslant 2d+1$$

6.12 Dudley 类：在本题中，我们讨论一种用来定义在 $\mathbb{R}^n$ 上概念类的代数框架并且说明这样的类的 VC 维与其代数性质之间的联系。给定一个函数 $f:\mathbb{R}^n\to\mathbb{R}$，我们定义其相关的函数，$\mathrm{POS}(f)(x)=\mathbb{1}_{[f(x)>0]}$。对于一个实值函数类 $\mathcal{F}$ 我们定义一个相关的函数类 $\mathrm{POS}(\mathcal{F})=\{\mathrm{POS}(f):f\in\mathcal{F}\}$。我们说一个实值函数族 $\mathcal{F}$ 是线性封闭的，如果对于所有的 f，$g\in\mathcal{F}$ 以及 $r\in\mathbb{R}$，都有 $(f+rg)\in\mathcal{F}$（其中函数的相加与标量乘法是逐点定义的，即对于所有 $x\in\mathbb{R}^n$，$(f+rg)(x)=f(x)+rg(x)$）。注意到如果一个函数族是线性封闭的那么我们可以将其当做实的向量空间。对于一个函数 $g:\mathbb{R}^n\to\mathbb{R}$ 和一个函数族 $\mathcal{F}$，令 $\mathcal{F}+g\overset{\mathrm{def}}{=}\{f+g:f\in\mathcal{F}\}$。对于某个向量空间 $\mathcal{F}$ 和某个函数 g 可以表示为 $\mathrm{POS}(\mathcal{F}+g)$ 的假设类被称为 Dudley 类。

1) 说明对于如之前定义的每个 $g:\mathbb{R}^n\to\mathbb{R}$ 和每个函数类 $\mathcal{F}$ 的向量空间，有 $\mathrm{VCdim}(\mathrm{POS}(\mathcal{F}+g))=\mathrm{VCdim}(\mathrm{POS}(\mathcal{F}))$。

**2) 对于每个线性封闭的实值函数类 $\mathcal{F}$，其相关类 $\mathrm{POS}(\mathcal{F})$ 的 VC 维等于 $\mathcal{F}$ 作为向量空间的线性维度。提示：令 f_1，…，f_d 是向量空间 $\mathcal{F}$ 上的基。考虑映射 $x\mapsto(f_1(x),\cdots,f_d(x))$（从 $\mathbb{R}^n$ 到 $\mathbb{R}^d$）。注意到该映射诱导了 $\mathbb{R}^n$ 上形如 $\mathrm{POS}(f)$ 的函数与 $\mathbb{R}^d$ 上平凡线性空间之间的对应（第 9 章中分析平凡线性空间类的 VC 维）。

3) 说明下列每个类都可以被表示为 Dudley 类：

(1) $\mathbb{R}^n$ 上的半空间类 HS_n（见第 9 章）。

(2) $\mathbb{R}^n$ 上的所有平凡半空间构成的类 THS_n（见第 9 章）。

(3) 由在 $\mathbb{R}^n$ 上（开）球定义的所有函数构成的类 B_n。利用 Dudley 表示指出该类的 VC 维。

(4) 令 P_n^d 表示由阶数 $\leqslant d$ 的多项式不等式定义的函数构成的类，即

$$P_n^d=\{h_p:p\text{ 是阶数}\leqslant d\text{ 的多项式，变量为 }x_1,\cdots,x_n\}$$

其中对于 $x=(x_1,\cdots,x_n)$，$h_p(x)=\mathbb{1}_{[p(x)>0]}$（多元变量多项式的阶就是其所有项中指数和的最大值。例如，$p(x)=3x_1^3x_2^2+4x_3x_4^2$ 的阶为 5）。

① 利用 Dudley 表示指出类 P_1^d（在 $\mathbb{R}$ 上所有 d 阶多项式类）的 VC 维。

② 证明 $\mathbb{R}$ 上所有多项式分类器构成的类具有无限的 VC 维。

③ 利用 Dudley 表示指出类 P_n^d（表示为 d 和 n 的函数）的 VC 维。

不一致可学习

目前为止，本书所讨论的PAC可学习的概念只考虑依赖精度和置信参数来决定样本数量，前提条件是，样本标签分布与内在的样本数据分布是一致的。类比，类别可学习是有条件的，样本必须具有有限的VC维（如定理6.7的说明）。在本章中，我们考虑更松的、更弱化约束条件下可学习的概念。我们将讨论这些概念的用途，以及提供在这种新定义下可学习概念类的特征描述。

首先，我们定义一个"不一致可学习"的概念，这个概念下允许样本数量依赖于学习器所在假设空间而变化。然后，我们描述"不一致可学习"的特征，指出"不一致可学习"是不可知PAC可学习的严格拓展。我们还论证了"不一致可学习"的一个充分条件是：$\mathcal{H}$是一个假设类别的可数并集，并且集合中的每个假设类都具有一致收敛属性。这个结论将在7.2节给出证明，证明过程中用到了结构风险最小化（SRM）机器学习范例。在7.3节我们具体描述了一种用于假设类可数并集的SRM法则，SRM范例是通过最小描述长度（MDL）方法实现的。MDL方法给出了一种奥卡姆剃刀哲学原理的形式化例证。然后，在7.4节，我们引入了一致性这种更加弱化的可学习概念，最后，分析了各种可学习概念的用途和意义。

7.1 不一致可学习概述

"不一致可学习"允许学习器针对所竞争的不同假设使用不同数量的样本。我们认为一个假设h以(ε, δ)可与另一个假设h'竞争，如果下式成立的概率不少于$(1-\delta)$，

$$L_{\mathcal{D}}(h) \leq L_{\mathcal{D}}(h') + \varepsilon$$

在PAC可学习中，没有用到"竞争力"的概念，当我们寻找具有绝对的最小风险的假设（在可选的情况下）或者寻找一个与最小风险差不多风险（在绝对最小风险不可知情况下）的假设，样本数量仅仅依赖于精度和置信度。然而，在不一致学习中，我们允许样本数量以$m_{\mathcal{H}}(\varepsilon, \delta, h)$的形式表示，也就是说，不一致可学习在表示形式上也依赖竞争力变量h。

定义7.1 若存在一个学习算法A和一个函数$m_{\mathcal{H}}^{\mathrm{NU}}:(0,1)^2\times\mathcal{H}\to\mathbb{N}$，使得对于任意的$\varepsilon, \delta\in(0,1)$和$h\in\mathcal{H}$，如果样本数量$m\geq m_{\mathcal{H}}^{\mathrm{NU}}(\varepsilon, \delta, h)$，那么对每个分布$\mathcal{D}$和所有的样本$S\sim\mathcal{D}^m$，下式成立的概率不少于$1-\delta$，

$$L_{\mathcal{D}}(A(S)) \leq L_{\mathcal{D}}(h) + \varepsilon$$

则假设类$\mathcal{H}$是不一致可学习的。

此时，回想下不可知条件下PAC可学习的定义（定义3.3）也许有帮助：

若存在一个学习算法A和一个函数$m_{\mathcal{H}}:(0,1)^2\to\mathbb{N}$，使得对于任意$\varepsilon, \delta\in(0,1)$和任一个分布$\mathcal{D}$，如果样本数量$m\geq m_{\mathcal{H}}(\varepsilon, \delta)$，那么对所有的样本$S\sim\mathcal{D}^m$，下式成立的概率不少于$1-\delta$，

$$L_{\mathcal{D}}(A(S)) \leq \min_{h'\in\mathcal{H}} L_{\mathcal{D}}(h') + \varepsilon$$

则假设$\mathcal{H}$是不可知条件下PAC可学习的。

注意，这就表示对于任一个 $h\in\mathcal{H}$，有下式成立

$$L_{\mathcal{D}}(A(S))\leqslant L_{\mathcal{D}}(h)+\varepsilon$$

在以上两类可学习中，我们要求输出假设与在假设类中的其他假设相比具有(ε,δ)竞争力。但是两类假设也有区别，那就是在不一致可学习中，样本数量 m 依赖于 $A(S)$错误对应的假设 h，而不知条件下PAC可学习不依赖于 h。同时，我们注意到不一致可学习比PAC可学习对假设条件要求更少，也就是说，如果一个假设类是不可知条件下PAC可学习，那么它也是不一致可学习的。

不一致可学习的特征

我们的目标是定义不一致可学习的特征。在之前的章节中，通过说明两类分类器假设类是不可知条件下PAC可学习的充要条件是它的VC维是有限的，我们已经找到PAC可学习类的简要特征。接下来的理论分析中，我们发现了在两类分类器上，不一致可学习与不可知PAC可学习不同的特征。

定理7.2 两类分类器的假设类$\mathcal{H}$是不一致可学习的当且仅当它是不可知PAC可学习假设类的可数并。

定理7.2的证明依赖于下面的定理7.3。

定理7.3 令一个假设类$\mathcal{H}$能够写成假设类的可数并，$\mathcal{H}=\bigcup_{n\in\mathbb{N}}\mathcal{H}_n$，如果$\mathcal{H}_n$是一致收敛的，那么$\mathcal{H}$是不一致可学习的。

59

回想下，在第4章中我们看到一致收敛是不可知条件下PAC可学习的充分条件，定理7.3将这个结论推广到不一致可学习。下一节中引入一个新的学习定理来证明定理7.3。现在我们证明定理7.2。

定理7.2的证明

充分性：假定$\mathcal{H}=\bigcup_{n\in\mathbb{N}}\mathcal{H}_n$，$\mathcal{H}_n$是不可知条件下PAC可学习的。应用统计学习的基本理论，每一个$\mathcal{H}_n$都遵循一致收敛属性。由定理7.3可知，$\mathcal{H}$是不一致可学习的。

必要性：假定$\mathcal{H}$是不一致可学习的并且使用算法 A。对于每一个 $n\in\mathbb{N}$，令$\mathcal{H}_n=\{h\in\mathcal{H}:m_{\mathcal{H}}^{NU}(1/8,1/7,h)\leqslant n\}$。显然，$\mathcal{H}=\bigcup_{n\in\mathbb{N}}\mathcal{H}_n$。此外，通过$m_{\mathcal{H}}^{NU}$的定义，我们知道对于任何关于$\mathcal{H}_n$满足可实现性假设的分布$\mathcal{D}$，选择样本 $S\sim\mathcal{D}^m$ 概率大于等于 6/7，则$L_{\mathcal{D}}(A(S))\leqslant 1/8$。由统计学习理论可知，$\mathcal{H}_n$的VC维一定是有限的，因此$\mathcal{H}_n$是不可知条件下PAC可学习的。 ■

下面的例子说明不一致可学习是不可知条件下PAC可学习的严格松弛，也就是说，存在假设类是不一致可学习的，但不是不可知条件下PAC可学习的。

例7.1 考虑一个二分类问题，样本在实际数域上取值。对于任意 $n\in\mathbb{N}$，$\mathcal{H}_n$是 n 次多项式分类器构成的假设类，也就是说，$\mathcal{H}_n$是形如$h(x)=\mathrm{sign}(p(x))$的所有分类器集合，这里 p 是$\mathbb{R}\to\mathbb{R}$的 n 次多项式。令$\mathcal{H}=\bigcup_{n\in\mathbb{N}}\mathcal{H}_n$，则$\mathcal{H}$是实数域$\mathbb{R}$上所有多项式构成的假设类。容易证明$\mathcal{H}$的VC维等于$\infty$，$\mathcal{H}_n$的VC维为 $n+1$(详见习题7.12)。因此，$\mathcal{H}$不是PAC可学习的，根据定理7.3，H是不一致可学习的。 ◀

7.2 结构风险最小化

目前为止，我们已经通过具体化一个假设类$\mathcal{H}$来利用先验知识，并且我们相信这样一个假设类中包含完成当前任务的有效预测器。然而，另一种表达先验知识的方式是将假设类$\mathcal{H}$上的偏好具体化。在结构风险最小化范例中，我们已经这样做了，首先假定$\mathcal{H}$能够写成$\mathcal{H}=\bigcup_{n\in\mathbb{N}}\mathcal{H}_n$，然后具体化一个权重函数$\omega:\mathbb{N}\to[0,1]$，这个权重函数给每个假设类赋予一个权重，高的权值表示对该假设类的强烈偏好。在这一节中，我们讨论如何学习这样的先验知识。在下一节中，我们描述一类重要的权重方法，包括最小描述长度。

具体来说，假设$\mathcal{H}$是一个能够写成形如$\mathcal{H}=\bigcup_{n\in\mathbb{N}}\mathcal{H}_n$的假设类。例如，$\mathcal{H}$可能是所有多项式分类器构成的类，$\mathcal{H}_n$表示$n$次多项式分类器构成的类(详见例7.1)。假定，对于任一个n，$\mathcal{H}_n$类满足一致收敛属性(详见第4章定义4.3)，且样本复杂度函数为$m_{\mathcal{H}_n}^{UC}(\varepsilon,\delta)$。通过下式定义函数$\varepsilon_n:\mathbb{N}\times(0,1)\to(0,1)$，

[60]

$$\varepsilon_n(m,\delta)=\min\{\varepsilon\in(0,1):m_{\mathcal{H}_n}^{UC}(\varepsilon,\delta)\leq m\}\tag{7.1}$$

总之我们有一个固定的样本数量m，我们感兴趣的是给定m个样本，经验风险和实际风险之差最小的概率上界。

从一致收敛的定义和ε_n，它遵循对于任一个m和δ，样本$S\sim\mathcal{D}^m$，下式成立的概率不少于$1-\delta$，

$$\forall h\in\mathcal{H}_n,|L_{\mathcal{D}}(h)-L_S(h)|\leq\varepsilon_n(m,\delta)\tag{7.2}$$

令$\omega:\mathbb{N}\to[0,1]$表示一个函数，满足$\sum_{n=1}^{\infty}\omega(n)\leq 1$，我们定义$\omega$是假设类$\mathcal{H}_1,\mathcal{H}_2,\cdots$的一个权重函数，这样一个权重函数可以反映每个假设类学习属性的重要性，或者不同假设类复杂性的度量。如果$\mathcal{H}$是N个假设类的有限并，我们也可以简单地对任一个假设类赋予$1/N$的权重，同等的权重意味着对任一假设类没有先验的偏好。当然，如果你认为某个假设类更有可能包含正确的目标函数，就可以给该假设类赋予较大的权重来反映这种先验知识。当$\mathcal{H}$是一个无穷(可数)假设的集合，虽然一致的权重假设是不可实现的，但是很多其他的权重设置可以使用。例如，你可以选择$\omega(n)=\frac{6}{\pi^2n^2}$或者$\omega(n)=2^{-n}$。在本章的后面部分，我们将使用描述语言介绍一种更加方便的定义权重函数的方法。

结构风险最小化是一种“最小化界”的方法。这就是说结构风险最小化是要寻找一个假设类来最小化真实风险的上确界。结构风险最小化原理期望最小化的界将在下面的定理中给出。

定理7.4 令$\omega:\mathbb{N}\to[0,1]$是一个权值函数，满足$\sum_{n=1}^{\infty}\omega(n)\leq 1$，$\mathcal{H}$是一个假设类可以写成$\mathcal{H}=\bigcup_{n\in\mathbb{N}}\mathcal{H}_n$，对于任一个$n$，$\mathcal{H}_n$满足一致收敛性，并且复杂度表示函数为$m_{\mathcal{H}_n}^{UC}$，令$\varepsilon_n$由方程(7.1)定义。然后，对于任一个$\delta\in(0,1)$，样本$S\sim\mathcal{D}^m$，对于任一个$n\in\mathbb{N}$和$h\in\mathcal{H}_n$，下式成立的概率不低于$1-\delta$，

$$|L_{\mathcal{D}}(h)-L_S(h)|\leq\varepsilon_n(m,\omega(n)\cdot\delta)$$

则对于任一个$\delta\in(0,1)$和分布$\mathcal{D}$，下式成立的概率不低于$1-\delta$，

$$\forall h\in\mathcal{H},L_{\mathcal{D}}(h)\leq L_S(h)+\min_{n:h\in\mathcal{H}_n}\varepsilon_n(m,\omega(n)\cdot\delta)\tag{7.3}$$

证明 对于任一个 n，定义 $\delta_n=\omega(n)\delta$。假定对于所有以方程(7.2)给出的 n 都满足一致收敛性，应用这个假设可得，若我们事先固定 n，在选择样本 $S\sim\mathcal{D}^m$ 的概率不低于 $1-\delta$ 条件下，

$$\forall h\in\mathcal{H}_n,\ |L_{\mathcal{D}}(h)-L_S(h)|\leqslant\varepsilon_n(m,\delta_n)$$

应用 $n=1,2,\cdots$ 的联合界，我们得到上述结论的概率不低于 $1-\sum_n\delta_n=1-\delta\sum_n\omega(n)\geqslant 1-\delta$。上述论证对所有的 n 都有效，也就完成了证明。 ■

令

$$n(h)=\min\{n:h\in\mathcal{H}_n\} \tag{7.4}$$

61

结合方程(7.3)，可得

$$L_{\mathcal{D}}(h)\leqslant L_S(h)+\varepsilon_{n(h)}(m,\omega(n(h))\cdot\delta)$$

结构风险最小化寻找假设 h 来最小化这个界，如下面伪代码形式化表示：

结构风险最小化(SRM)

先验：

$\mathcal{H}=\bigcup_{n\in\mathbb{N}}\mathcal{H}_n$，$\mathcal{H}_n$ 满足一致收敛，复杂度函数为 $m_{\mathcal{H}_n}^{\mathrm{UC}}$

$\omega:\mathbb{N}\to[0,1]$，其中 $\sum_n\omega(n)\leqslant 1$

定义： ε_n 由方程(7.1)定义，$n(h)$ 由方程(7.4)定义

输入： 训练集 $S\sim\mathcal{D}^m$，置信度 δ

输出： $h\in\operatorname{argmin}_{h\in\mathcal{H}}[L_S(h)+\varepsilon_{n(h)}(m,\omega(n(h))\cdot\delta)]$

与前面章节讨论的 ERM(经验风险最小化)不同，我们不仅关心经验风险 $L_S(h)$，而且为了最小化估计误差，更加关心在最小经验风险的偏置和 $\varepsilon_{n(h)}(m,\omega(n(h))\cdot\delta)$ 最小化之间取得一个平衡。

然后我们揭示了结构风险最小化能够用于每个类的不一致学习，这里的不一致学习指的是一致收敛假设类的可数并。

定理 7.5 令 $\mathcal{H}$ 是假设类，满足 $\mathcal{H}=\bigcup_{n\in\mathbb{N}}\mathcal{H}_n$，$\mathcal{H}_n$ 满足一致收敛性，并且复杂度表示函数为 $m_{\mathcal{H}_n}^{\mathrm{UC}}$，如果 $\omega:\mathbb{N}\to[0,1]$ 满足 $\omega(n)=\frac{6}{\pi^2n^2}$，那么，$\mathcal{H}$ 是不一致可学习的，结构风险最小化率为

$$m_{\mathcal{H}}^{\mathrm{NUL}}(\varepsilon,\delta,h)\leqslant m_{\mathcal{H}_{n(h)}}^{\mathrm{UC}}\left(\varepsilon/2,\frac{6\delta}{(\pi n(h))^2}\right)$$

证明 假定 A 是考虑权重函数 ω 的结构风险最小化算法。对于每一个 $h\in\mathcal{H}_n$，ε 和 δ，令 $m\geqslant m_{\mathcal{H}_{n(h)}}^{\mathrm{UC}}(\varepsilon,\omega(n(h))\delta)$。根据 $\sum_n\omega(n)=1$，选择样本 $S\sim\mathcal{D}^m$ 的概率不低于 $1-\delta$，应用定理 7.4，可以得到对于任一个 $h'\in\mathcal{H}_n$，下式成立，

$$L_{\mathcal{D}}(h')\leqslant L_S(h')+\varepsilon_{n(h')}(m,\omega(n(h'))\delta)$$

这个定理对于由结构风险最小化规则返回的假设 $A(S)$ 成立。通过结构风险最小化的定义可得

$$L_{\mathcal{D}}(A(S))\leqslant\min_{h'}[L_S(h')+\varepsilon_{n(h')}(m,\omega(n(h'))\delta)]\leqslant L_S(h)+\varepsilon_{n(h)}(m,\omega(n(h))\delta)$$

如果 $m \geq m_{\mathcal{H}_{\omega(n)}}^{UC}(\varepsilon/2, \omega(n(h))\delta)$，那么 $\varepsilon_{m,\omega}(m, \omega(n(h))\delta) \leq \varepsilon/2$ 成立。此外，从每个 $\mathcal{H}_n$ 的一致收敛属性，我们可得下式成立的概率大于 $1-\delta$，

$$L_S(h) \leq L_{\mathcal{D}}(h) + \varepsilon/2$$

综上所述，可得 $L_{\mathcal{D}}(A(S)) \leq L_{\mathcal{D}}(h) + \varepsilon$，定理得证。 ■

[62] 注意，前面的理论也证明了定理 7.3。

评注(不一致可学习的"没有免费的午餐"原理) 我们已经展示了任何可数的有限 VC 维所构成的类是不一致可学习。结论显示，对于无限域集合 $\mathcal{X}$，所有 $\mathcal{X}$ 上定义的二值函数所构成的类不是有限 VC 维的可数并。我们将这个结论的证明留作练习 7.5。接下来，在某种意义上，在不一致可学习中"没有免费的午餐"理论也是成立的。也就是说，当样本域无限时，不存在关于所有确定性二类分类器所构成的类的不一致学习器(尽管对于每一个分类器存在一个尝试算法能够学习包含这些分类器假设的结构风险最小化)。

单独比较 7.5 节理论所表述的不一致可学习和任何 $\mathcal{H}_n$ 的不可知条件下 PAC 可学习任务，是非常有意思的。先验知识、偏置和不一致学习器估计 $\mathcal{H}$ 是不够充分的，它需要在全空间上搜索一个模型，而不是在特定 $\mathcal{H}_n$ 上搜索一个模型。利用先验知识缺陷所带来的成本就是增加复杂度来与特定的 $h \in \mathcal{H}_n$ 相竞争。对于这种差异的简单估计就是，考虑到 0—1 损失的二值分类任务，假定对于所有 n，$\mathcal{H}_n$ 的 VC 维度为 n。因为 $m_{\mathcal{H}_n}^{UC}(\varepsilon, \delta) = C\frac{n+\log(1/\delta)}{\varepsilon^2}$($C$ 是定理 6.8 中所出现的数)，一个直接的计算表明

$$m_{\mathcal{H}}^{NU}(\varepsilon, \delta, h) - m_{\mathcal{H}_n}^{UC}(\varepsilon/2, \delta) \leq 4C\frac{2\log(2n)}{\varepsilon^2}$$

也就是说，从特定 $\mathcal{H}_n$ 中挖掘先验知识的成本，包含目标 h 来度量类的集合，这个类 $\mathcal{H}_n$ 包含标签 h 来建立一个假设类的可数并。这些类依赖于 h 所在的第一类的对数索引。代价增加了类的索引，可以解释为反映已知的假设类 $\mathcal{H}$ 的好的先验知识的排序值。

7.3 最小描述长度和奥卡姆剃刀

令 $\mathcal{H}$ 是可计算的假设类，那么，我们将 $\mathcal{H}$ 写成单个类的可数并，也就是 $\mathcal{H} = \bigcup_{n \in \mathbb{N}} \{h_n\}$。由 Hoeffding 不等式(引理 4.5)，每一个单类有一致收敛性，收敛速率 $m^{UC}(\varepsilon, \delta) = \frac{\log(2/\delta)}{2\varepsilon^2}$。因此，方程(7.1)所给出的函数 ε_n 变成 $\varepsilon_n(m, \delta) = \sqrt{\frac{\log(2/\delta)}{2m}}$，且结构风险最小化变成

$$\operatorname*{argmin}_{h \in \mathcal{H}} \left[L_S(h) + \sqrt{\frac{-\log(\omega(n)) + \log(2/\delta)}{2m}} \right]$$

等价地，我们可以认为 ω 是从 $\mathcal{H}$ 变换到 $[0, 1]$ 的函数，然后结构风险最小化变成

$$\operatorname*{argmin}_{h \in \mathcal{H}} \left[L_S(h) + \sqrt{\frac{-\log(\omega(h)) + \log(2/\delta)}{2m}} \right]$$

接下来假定前提条件，先验知识单纯由我们分配给每个假设类的权重决定。我们对可能正确的假设分配较高的权重，并且在机器学习中我们偏爱权值高的假设。 [63]

这节中我们讨论一种特别方便的方式定义 $\mathcal{H}$ 的权重函数，这个方法起源于假设的描述长度。有一个假设类，我们想知道如何描述和表示每一个类中的假设。自然地，我们聚焦在一些描述语言中想办法。这些语言可能是英语、编程语言或者一些数学公式。任何一种语言中，一个描述都是由一些特定的字母所组成符号的有限字符串构成。现在，我们形式

化这些概念。

令$\mathcal{H}$是我们要描述的假设类，定义有限符号集合Σ，我们称之为字母表。具体地说，我们令$\Sigma=\{0,1\}$，一个字符串是Σ中的有限符号序列。例如，$\boldsymbol{\sigma}=(0,1,1,1,0)$的字符串长度为5。我们用$|\boldsymbol{\sigma}|$表示字符串的长度。所有有限字符串的集合用$\Sigma^*$表示。对$\mathcal{H}$的描述语言用一个函数$d:\mathcal{H}\to\Sigma^*$，将$\mathcal{H}$中的每一个$h$假设映射为一个字符串$d(h)$，$d(h)$称为$h$的描述长度，并且$h$的描述长度用$|h|$表示。我们要求描述语言无前缀，也就是说不同的$h$，$h'$，$d(h)$不是$d(h')$的前缀。也就是，我们不允许任何一个字符串$d(h)$与另一个长字符串$d(h')$的前$|h|$个的符号完全一致。无前缀的字符串集合满足下面的组合属性：

引理7.6(Kraft不等式) 如果$S\subseteq\{0,1\}^*$是一个无前缀的字符串集合，则$\sum_{\boldsymbol{\sigma}\in S}\frac{1}{2^{|\boldsymbol{\sigma}|}}\leq 1$。

证明 定义成员S的一个概率分布如下：重复掷一个均匀的硬币，两个面分别用0和1表示，直到序列的结果是S的一个成员，此时停止掷硬币。对于任$\boldsymbol{\sigma}\in S$，$P(\boldsymbol{\sigma})$表示由上述过程产生字符串$\boldsymbol{\sigma}$的概率。注意到由于S无前缀，对于每一个$\boldsymbol{\sigma}\in S$，每一次投硬币的结果与$\boldsymbol{\sigma}$的比特位一致，当投硬币的输出序列等于$\boldsymbol{\sigma}$时停止投硬币。因此，我们可以得到对于每一个$\boldsymbol{\sigma}\in S$，$P(\boldsymbol{\sigma})=\frac{1}{2^{|\boldsymbol{\sigma}|}}$。由于概率最大之可能为1，因此结论得证。 ■

根据Kraft不等式，任何假设$\mathcal{H}$的无前缀描述语言都能给出假设类$\mathcal{H}$的权重函数w，我们可以简单地设置为$w(h)=\frac{1}{2^{|h|}}$。以上观察可以立即得到以下理论：

定理7.7 令$\mathcal{H}$是一个假设类，$d:\mathcal{H}\to\{0,1\}^*$是$\mathcal{H}$的一个无前缀描述语言。对于样本数量$m$，置信度参数$\delta>0$和概率分布$\mathcal{D}$，样本$S\sim\mathcal{D}^m$，下式成立的概率大于$1-\delta$，

$$\forall h\in\mathcal{H},L_{\mathcal{D}}(h)\leq L_S(h)+\sqrt{\frac{|h|-\ln(2/\delta)}{2m}}$$

这里$|h|$是指$d(h)$的长度。

证明 选择$w(h)=\frac{1}{2^{|h|}}$，应用定理7.4，$\varepsilon_w(m,\delta)=\sqrt{\frac{\ln(2/\delta)}{2m}}$。注意到，$\ln(2^{|h|})=|h|\ln(2)<|h|$。 ■

和定理7.4的情形一样，这个结果给出了对于训练集S，搜索假设$h\in\mathcal{H}$最小化界$L_S(h)+\sqrt{\frac{|h|-\ln(2/\delta)}{2m}}$的$\mathcal{H}$的一个学习范式。具体地说，这种方法折中考虑了经验风险和减少描述长度，这就得到了最小描述长度的学习范式。 | 64 |

最小描述长度(MDL)

先验：

$\mathcal{H}$是可计算的假设类

$\mathcal{H}$由定义在$\{0,1\}$上的无前缀语言描述

对于任一个$h\in\mathcal{H}$，$|h|$表示h的长度

输入：一个训练集$S\sim\mathcal{D}^m$，置信度为δ

输出：$h\in\operatorname{argmin}_{h\in\mathcal{H}}\left[L_S(h)+\sqrt{\frac{|h|+\ln(2/\delta)}{2m}}\right]$

例 7.2 令$\mathcal{H}$是所有预测器构成的类，这些预测器可以通过像 C++ 一样的编程语言来实现。用二进制串表示每一个程序，这些二进制串通过程序运行 gzip 命令而得到(这就得到了定义在$\{0,1\}$上的无前缀描述语言)。然后，当运行与h相关的 C++ 程序时，$|h|$仅仅是输出的比特位数。 ◀

奥卡姆剃刀

定理 7.7 指出，对于经验风险相同的两个假设，最小描述长度较小的假设，其真实风险的风险误差界更小。因此，这个结果表达了一种哲学理念：

短的解析(也就是长度短的假设)比长的解析更有效。

这是一个著名的原理，称之为奥卡姆剃刀，以 14 世纪的一个英国逻辑学家(威廉姆·奥卡姆)命名，威廉姆·奥卡姆被认为是第一个清晰地表述了这个原理。这里我们给出这个原理的一个可能的理由。根据定理 7.7 的不等式，假设h越复杂(在这里就是描述长度越长)，就需要更多的样本来保证真实风险$L_{\mathcal{D}}(h)$最小。

重新审视之下，奥卡姆剃刀看起来也有一些问题。在通常引用奥卡姆原则的情境中，自然语言是指经过复杂度度量的语言，而此处我们将一切任意抽象描述的语言纳入考虑。假定我们有两个假设，$|h'|$比$|h|$短得多。根据之前的结论，如果两个假设在训练集S上取得同样的错误，h的真实风险高于h'，因此我们应该倾向于偏好h'。然而，我们可以选择另一种描述语言，使得h的长度为 3，而h'的长度为 10 000，此时，看起来我们应该偏好h。但是h和h'与前文所述的选择偏好h'时相比并无差别。此处的陷阱在哪里？

65

的确，这里假设之间没有本质的普适性不同。重要的方面在于初始语言的选择(假设偏好的先验)和训练集的相关性顺序。根据方程(4.2)给出的基本 Hoeffding 界，若我们在没有数据之前先给定假设，则要使得估计错误表达式$L_{\mathcal{D}}(h)\leqslant L_S(h)+\sqrt{\frac{\ln(2/\delta)}{2m}}$相对较小。选择一种描述语言(或者等价地，给出假设的权值)是一种较弱的提出假设的形式。而不是提出一个假设，然后在众多的假设中传播这个假设。只要与训练样本无关，我们的泛化误差界就可以保证。就像选择单一假设用于估计样本一样，选择描述语言也可能是随机的。

7.4 可学习的其他概念——一致收敛性

学习的概念可以进一步松弛，允许所需样本数量不仅依赖于ε、δ和h，而且依赖产生数据所依据的概率分布$\mathcal{D}$(概率分布$\mathcal{D}$用于产生训练样本和决定风险)。这种类型的性能保证由一种一致收敛性的学习规则来给出。[1]

定义 7.8(一致收敛性) 令Z表示一种域的集合，$\mathcal{P}$表示Z上的概率分布，$\mathcal{H}$表示假设类。若存在一个函数$m_{\mathcal{H}}^{UN}:(0,1)^2\times\mathcal{H}\times\mathcal{P}\to\mathbb{N}$使得对于任意一个$h\in\mathcal{H}$，$\mathcal{D}\in\mathcal{P}$，$\varepsilon,\delta\in(0,1)$，如果$m\geqslant m_{\mathcal{H}}^{UN}(\varepsilon,\delta,h,\mathcal{D})$，样本$S\sim\mathcal{D}^m$，下式成立的概率不低于$1-\delta$，

$$L_{\mathcal{D}}(A(S))\leqslant L_{\mathcal{D}}(h)+\varepsilon$$

我们就认为一个学习规则A关于$\mathcal{H}$和$\mathcal{P}$一致收敛[2]。如果$\mathcal{P}$是所有分布的集合，则我们说A全局收敛到$\mathcal{H}$。

---

[1] 历史上，一致收敛性定义时通常用到概率意义上收敛(对应弱一致收敛性)或者几乎确定收敛(对应强一致收敛性)。

[2] 形式上，我们假定Z被以σ-algebra 代数子集Ω，以及在相关子集中包含Ω度量子集的具有的分布。

当然，一致收敛性的概念是我们之前提到的不一致可学习概念的进一步松弛。显然，如果一个算法能不一致可学习一个类$\mathcal{H}$，那么它一定全局一致收敛到类$\mathcal{H}$。这种松弛是严格的，在这个意义下说一个算法是一致收敛学习，它不一定是不一致可学习的。例如，后面的例7.3所定义的Memorize算法对于$\mathcal{X}$上所有两类分类器构成的类是全局一致收敛的。但是，从之前讨论的结论，它不是不一致可学习的。

例7.3 考虑如下定义的分类预测算法Memorize。这个算法记忆训练样本，给定一个测试样本x，它在所有训练集存在的样本标签中，预测概率最大的样本标签（也可以是一些默认的固定标签，尽管没有x的实例样本出现在训练集）。如习题7.6所揭示的，Memorize算法对于每一个可计算的域$\mathcal{X}$和有限的标签$\mathcal{Y}$（0-1损失），是全局一致收敛的。◀ 56

乍看之下，Memorize算法作为一种学习器并不明显，因为它缺乏泛化方面，即使用观测数据去预测在训练样本集中没有出现的标签的能力。事实上，Memorize算法对于任何可数域集合上所有函数的构成的类都是一致收敛算法，因此我们对一致收敛性保证的用途产生了怀疑。敏锐的读者可能注意到在第2章中我们所介绍的"不良学习器"，可能导致过拟合的学习器，事实上也是Memorize算法。下一节中我们探讨不同的可学习概念的重要性，并再次应用"没有免费的午餐"理论进行分析。

7.5 探讨不同的可学习概念

我们已经给出三种可学习的概念，现在来讨论它们的用途。通常，一个数学定义的用途取决于我们为什么需要这样一个定义。因此我们列出几个可能的通过定义可学习而期望取得的目标，然后讨论应用不同可学习的定义来实现这些目标。

1. 学习假设的风险是什么？

第一个可能的目标来自于保证一个学习算法的输出预测风险界。这里，PAC可学习和不一致可学习都基于经验风险给出了学习假设的真实风险上界。一致收敛性没有提供这样一个界，但是通常可以使用验证集来估计输出预测器的风险（这将在第11章中描述）。

2. 得到$\mathcal{H}$中的最好假设需要多少样本？

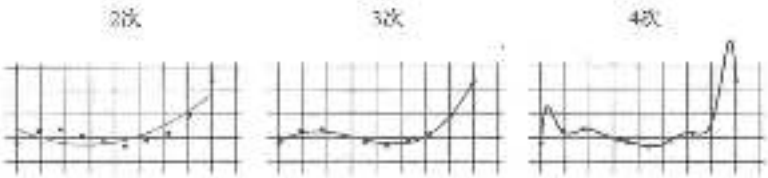
解决学习问题时，一个实际的问题就是我们需要收集多少个样本。对此，PAC学习给出了直接的答案。然而，不一致可学习和一致收敛性事先没有给出需要多少个样本来学习$\mathcal{H}$。在不一致学习中，样本数量依赖于$\mathcal{H}$中最好的假设，在一致收敛性中，样本数量还依赖于数据潜在的分布。从这个意义上来说，PAC学习是对可学习性唯一有用的定义。另一方面，我们应该记住，即使估计出预测器的错误很小，如果$\mathcal{H}$有很大的近似错误，这个风险也可能很大。因此，对于"需要多少样本来获得贝叶斯最优预测器"这个问题，即使是PAC也不能保证一个干脆的答案。这反映出一个事实，即就是应用PAC学习依赖于先验知识的质量。

PAC保证也能帮助我们理解，当学习算法返回一个大风险假设时我们下一步应该怎么做，这是因为我们对部分错误建立一个界，这个界来源于对误差的估计，因此知道有多少错误造成了近似误差。如果一个假设的误差很大，我们知道应该使用一个不同的假设类。同样地，如果一个不一致学习算法失败，我们可以考虑在假设类上使用不同的权重函数。然而，当一个一致收敛算法失败，我们不知道这是由估计误差还是近似误差造成的，甚至，即使我们确定问题是由估计误差造成的，我们也不能确定需要多少样本可以使得估计误差变小。 67

3. 如何学习？如何表达先验？

学习理论最有用的方面是为“如何学习”提供了答案。PAC学习的定义突破了学习的限制(通过“没有免费的午餐”理论)和必要的先验知识。PAC学习通过假设类的选择，给出了应用先验知识的直接方式。一旦假设类选定，我们就有了一个通用的学习规则——经验风险最小化。不一致可学习也提供一种应用先验知识的直接方式，那就是在假设类$\mathcal{H}$或它的子集上定义权重，一旦权值确定，我们也有了一个通用的学习规则——结构风险最小化。当先验知识是有偏的，结构风险最小化在模型选择上也有优势。我们在第11章中精心设计了模型选择，在这里只给出一个简要的范例。

考虑给定数据的一维多项式拟合问题，也就是说，我们的目标是学习一个函数$h:\mathbb{R}\to\mathbb{R}$，根据先验知识，考虑假设类是多项式，但是我们并不知道次数d为多少时能够在数据集上给出最好的结果。次数太低不能很好地拟合数据(比如大的拟合误差)，次数太高则可能会出现过拟合(比如大的估计误差)。接下来我们描述分别用2次，3次，10次多项式来拟合同样的数据集所取得的结果。



不难看出，当多项式次数增加，经验风险下降。因此，我们选择所有次数不大于10的多项式构成类$\mathcal{H}$，然后依据这个类的经验风险最小化原则输出一个10次多项式，而这会出现过拟合。另一方面，如果我们选择次数太小的多项式类，比如不大于2的多项式类，由于欠拟合，经验风险将会很大(比如大的近似错误)。相比较而言，我们可以在所有多项式集合中，使用结构风险最小化原则，同时强制子集$\mathcal{H}$依赖于各自的多项式次数，这将会得到3次多项式，因为同时考虑了经验风险和估计误差界最小化。换句话说，经验风险最小化使我们根据数据本身选择一个正确的模型。获得这种好处的代价是(除稍微增加PAC学习相关误差估计最优多项式次数之外)我们事先不知道需要多少个样本来确定$\mathcal{H}$中的最优假设。

[58]

与PAC学习和不一致可学习的概念不同，一致收敛的定义没有自然的学习范式或者编码利用先验的方式。事实上，大多数情况下都根本不需要先验知识。例如，我们看到即使Memorize算法对于定义在可计算的域和有限的标签集上的任何类都是一致收敛算法，它本质上还不能称为学习算法，这意味着一致收敛是一种非常弱的要求。

4. 我们偏好什么样的学习算法？

有人认为，即使一致收敛是弱条件，学习算法也最好要与所有$\mathcal{X}$到$\mathcal{Y}$的函数集合保持一致，这样可以保证只要有足够的训练样本，我们总可以得到贝叶斯最优估计。因此我们有两种算法，一种是一致收敛的，另一种不是一致收敛的，我们应该偏好一致收敛算法。然而，这种说法是有问题的，有以下两个原因：第一，可能在大多数自然分布情况下，一致收敛算法要求的样本过大，在现实中不可能每次都有足够的样本来满足这种保证。第二，构造PAC或者不一致可学习来获得关于$\mathcal{X}$到$\mathcal{Y}$的所有函数构成的类一致收敛也不是太难的事情。具体来说，考虑一个可计算的域$\mathcal{X}$、一个有限的标签集$\mathcal{Y}$和一个从$\mathcal{X}$到$\mathcal{Y}$的假设函数类$\mathcal{H}$，我们可以使用以下技巧，设计关于$\mathcal{X}$到$\mathcal{Y}$的所有分类器构成的类$\mathcal{H}$的任何不一致学习器来实现一致收敛。技巧如下：在一个接收的训练集上，我们首先运行不一致学习

器，学到预测器真实风险的一个界，如果界足够小，我们已经达到目的；否则，我们重新使用 Memorize 算法。这个简单的修改将使得我们的算法对于$\mathcal{X}$到$\mathcal{Y}$的所有函数一致收敛。由于让任何算法一致收敛是很容易的事情，就没有必要仅从一致收敛出发来确定偏好一种算法，而不是另一种算法。

重提"没有免费的午餐"理论

回想第5章5.1节"没有免费的午餐"理论，它是说没有算法能够在无限域上学习所有分类器构成的类。相比较而言，本章我们看到 Memorize 算法在无限域上的所有分类器构成的类是一致收敛的。要理解这两个说法是没有矛盾的，我们首先回想下"没有免费的午餐"理论的标准表述。

令$\mathcal{X}$表示可计算的无限域，$\mathcal{Y}=\{\pm 1\}$，"没有免费的午餐"理论如下：对于任何算法 A 和一个训练样本数量 m，存在一个分布$\mathcal{X}$和一个函数 $h^*:\mathcal{X}\to\mathcal{Y}$，如果 A 只是获得 m 个独立同分布样本其中的一个，对应的标签为 h^*，那么 A 很可能返回一个有大的误差的分类器。

Memorize 的一致收敛性如下：对于任何的分布$\mathcal{X}$和标签函数 $h^*:\mathcal{X}\to\mathcal{Y}$，存在一个训练集 m(依赖于分布和 h^*)使得 Memorize 算法至少需要 m 个样本来获得较小的误差。 [69]

7.6 小结

我们引入了不一致可学习作为 PAC 可学习的松弛，一致收敛性作为不一致可学习的松弛。这就意味着，在一些较弱的可学习概念里，即使 VC 维是无穷，也是可以学习的。我们讨论了不同可学习定义的用途。

对于可以计算的假设类，我们应用最小描述长度原理，根据奥卡姆剃刀原理，拥有较短的描述长度的假设更受偏爱。一个有意思的假设类是，所有的预测器都可以用C++（或者其他任何一种编程语言）来执行，我们可以通过最小描述长度原理来实现不一致学习。

有争议的是，所有的预测器（可以用 C++ 来执行）构成的类是一个强大的类，这个类中包含所有我们期望学习得到的东西。这种学习能力让人印象深刻，表面上看，这一章应当是本书的最后一章。事实并非如此，原因是学习的计算性：即应用学习规则所需的运行时间。例如，为了执行最小描述长度算法相关的 C++ 程序，我们需要穷尽搜索所有的 C++ 程序，这将永远也达不到。即使是执行经验风险最小化原则相关的所有最小描述长度的 C++ 程序至多 1000 个比特，也需要穷尽搜索 2^{1000} 个假设。但是学习这个类的样本复杂度仅仅只有 $\frac{1000+\log(2/\delta)}{\epsilon^2}$，运行时间 $\geq 2^{1000}$。这是一个很大的数，比可见的全宇宙原子数量都大。在下一章中，我们将正式定义学习复杂度。本书的第二部分将研究假设类，使得经验风险最小和结构风险最小化原理能够高效地执行。

7.7 文献评注

我们定义不一致可学习与奥卡姆剃刀算法相关(Blumer，Ehrenfeucht，Haussler 和 Warmuth，1987)。结构风险最小化的概念起源于 Vapnik & Chervonenkis(1974)，Vapnik(1995)。最小描述长度概念源于 Rissanen(1978)，Rissanen(1983)。结构风险最小化和最小描述长度的关系由 Vapnik(1995)探讨。这些概念与正则项密切相关(Tikhonov 1943)，我们在本书的第二部分对正则项展开讨论。

一致收敛的概率可以追溯到 Fisher(1922)，我们表述的一致收敛性遵循 Steinwart 和 Christmann(2008)，他们也发展了“没有免费午餐”理论。

70

7.8 练习

7.1 证明对于任何有限类$\mathcal{H}$和任何描述语言$d:\mathcal{H}\to\{0,1\}^*$，$\mathcal{H}$类的 VC 维至多是$2\sup\{|d(h)|:h\in\mathcal{H}\}$，$\mathcal{H}$的一个预测器的最大描述长度。甚至，如果$d$是一个无前缀描述，那么$\mathrm{VCdim}(\mathcal{H})\leq\sup\{|d(h)|:h\in\mathcal{H}\}$。

7.2 令$\mathcal{H}=\{h_n:n\in\mathbb{N}\}$表示一个无限可计算的二类分类器的假设类。证明不可能对$\mathcal{H}$的假设类赋予权重使得：

1) $\mathcal{H}$能够使用这些权重进行不一致地学习。即，权重函数$\omega:\mathcal{H}\to[0,1]$应该满足条件$\sum_{h\in\mathcal{H}}\omega(h)\leq 1$。

2) 权重是单调不下降的。即，如果$i<j$，那么$\omega(h_i)\leq\omega(h_j)$。

7.3 1) 考虑一个假设类$\mathcal{H}=\bigcup_{n\in\mathbb{N}}\mathcal{H}_n$，对于每个$n\in\mathbb{N}$，$\mathcal{H}_n$是有限的。找到一个权重函数$\omega:\mathcal{H}\to[0,1]$使得$\sum_{h\in\mathcal{H}}\omega(h)\leq 1$，则对于所有的$h\in\mathcal{H}$，$\omega(h)$由$n(h)=\min\{n:h\in\mathcal{H}_n\}$和$|\mathcal{H}_{n(h)}|$决定。

*2) 定义一个权重函数ω，当所有的n，$\mathcal{H}_n$是可计算的(可能是无限的)。

7.4 令$\mathcal{H}$表示假设类，对任一$h\in\mathcal{H}$，$|h|$表示h的描述长度(根据一些固定的描述语言)，考虑 MDL 的最小描述长度学习原理，在这个算法里返回：

$$h_S\in\operatorname*{argmin}_{h\in\mathcal{H}}\left[L_S(h)+\sqrt{\frac{|h|+\ln(2/\delta)}{2m}}\right]$$

这里S表示样本数量m，对于任一$B>0$，令$\mathcal{H}_B=\{h\in\mathcal{H}:|h|\leq B\}$，定义

$$h_B^*=\operatorname*{argmin}_{h\in\mathcal{H}_B}L_{\mathcal{D}}(h)$$

证明：$L_{\mathcal{D}}(h_S)-L_{\mathcal{D}}(h_B^*)$是关于$B$的一个界，置信参数$\delta$，训练样本集的数量$m$。

注意：这样的界在历史上被称为神谕不等式：我们期望估计参考分类器(或者神谕)h_B^*是否足够好。

7.5 在这个问题中我们将要揭示不一致可学习的“没有免费的午餐”的一个结果。即在无限域上，即使是在松弛的不一致变化可学习，也不是所有函数构成的类是可学习的。

回想算法A，一个假设类$\mathcal{H}$的不一致学习，如只存在一个函数$m_{\mathcal{H}}^{\mathrm{UL}}:(0,1)^2\times\mathcal{H}\to\mathbb{N}$使得对于每一个$\varepsilon,\delta\in(0,1)$，$h\in\mathcal{H}$，如果$m\geq m_{\mathcal{H}}^{\mathrm{NL}}(\varepsilon,\delta,h)$，那么对于任一个分布$\mathcal{D}$，样本$S\sim\mathcal{D}^m$，下式成立的概率不低于$1-\delta$，

$$L_{\mathcal{D}}(A(S))\leq L_{\mathcal{D}}(h)+\varepsilon$$

如果存在这样一个算法，那么我们说$\mathcal{H}$是不一致可学习的。

1) 令A是类$\mathcal{H}$的不一致学习器，对于任一$n\in\mathbb{N}$，定义$\mathcal{H}_n=\{h\in\mathcal{H}:m^{\mathrm{NL}}(0.1,0.1,h)\leq n\}$。证明：每个$\mathcal{H}_n$有一个有限的 VC 维。

2) 证明如果类$\mathcal{H}$是不一致可学习的，那么有类$\mathcal{H}_n$，使得$\mathcal{H}=\bigcup_{n\in\mathbb{N}}\mathcal{H}_n$，对于任一个$n\in\mathbb{N}$，$\mathrm{VCdim}(\mathcal{H}_n)$是有限的。

3) 令$\mathcal{H}$表示一个类散落在无限数据集上，然后，对于类的序列$(\mathcal{H}_n:n\in\mathbb{N})$，$\mathcal{H}=\bigcup_{n\in\mathbb{N}}\mathcal{H}_n$，存在$n$使得$\mathrm{VCdim}(\mathcal{H}_n)=\infty$。

提示：给定一个类$\mathcal{H}$散落在一些无限的数据集K上和一个序列类$(\mathcal{H}_n: n\in\mathbb{N})$，每个都有一个有限的VC维，开始定义一个子集$K_n\subseteq K$，使得对于所有的$n$，$|K_n|>\mathrm{VCdim}(\mathcal{H}_n)$，并且对于$n\neq m$，$K_n\cap K_m=\varnothing$。对于每个这样的$K_n$的挑选一个函数$f_n: K_n\to\{0,1\}$，使得没有$h\in\mathcal{H}$在域$K_n$上遵循$f_n$。最后，通过连接这些$f_n$定义$f: X\to\{0,1\}$，证明$f\in(\mathcal{H}\setminus\bigcup_{n\in\mathbb{N}}\mathcal{H}_n)$。 [71]

4) 构建一个从$[0,1]$到$\{0,1\}$的函数类$\mathcal{H}_1$，这是不一致可学习的，但不是PAC可学习的。

5) 构建一个从$[0,1]$到$\{0,1\}$的函数类$\mathcal{H}_2$，这不是不一致可学习的。

7.6 在这个问题中，我们期望揭示Memorize算法对于任一个定义在任意可计算域上函数类是一致可学习的。令$\mathcal{X}$表示一个可计算的域，并且$\mathcal{D}$是$\mathcal{X}$的概率分布。

1) 令$\{x_i: i\in\mathbb{N}\}$表示元素$\mathcal{X}$的一个枚举对象，使得对于所有的$i\leqslant j$，$\mathcal{D}(\{x_i\})\leqslant\mathcal{D}(\{x_j\})$，证明：

$$\lim_{n\to\infty}\sum_{i>n}(\mathcal{D}(\{x_i\}))=0$$

2) 给定任一$\varepsilon>0$，证明存在$\varepsilon_0>0$使得

$$\mathcal{D}(\{x\in\mathcal{X}: \mathcal{D}(\{x\})\leqslant\varepsilon_0\})<\varepsilon$$

3) 证明对于任一$\eta>0$，如果n使得对于所有的$i>n$，$\mathcal{D}(\{x_i\})<\eta$，那么对于任一$m\in\mathbb{N}$，下式成立：

$$\mathop{\mathbb{P}}_{S\sim\mathcal{D}^m}[\exists x_i: (\mathcal{D}(\{x_i\})>\eta \text{ 且 } x_i\notin S)]\leqslant ne^{-\eta m}$$

4) 推断，如果$\mathcal{X}$是可计算的，那么对于$\mathcal{X}$的任一个概率分布$\mathcal{D}$，存在一个函数$m_{\mathcal{D}}: (0,1)\times(0,1)\to\mathbb{N}$，使得对于任一$\varepsilon$，$\delta>0$，如果$m>m_{\mathcal{D}}(\varepsilon,\delta)$，那么

$$\mathop{\mathbb{P}}_{S\sim\mathcal{D}^m}[\mathcal{D}(\{x: x\notin S\})>\varepsilon]<\delta$$

5) 证明Memorize算法在可计算的域上，对于所有二值函数构成的类是一致收敛学习算法。 [72]

学习的运行时间

到目前为止，本书从统计的视角研究了学习，即学习需要多少样本；换句话说，我们关注学习需要的信息量。但是如果是自动学习，任务的复杂程度主要由计算资源决定，即执行任务时需要多少计算。一旦学习者有一个充分的训练样本，花费一些计算量就可以从中提取假设或者对给定的测试样例加标注。这些计算资源对任何一个机器学习的应用都是至关重要的，我们将之分成两类，一类是样本复杂度，一类是计算复杂度。在本章，我们将转而关注机器学习的计算复杂度。

机器学习的复杂度应该在更一般的通用算法的层面上考虑。这个领域已经有了大量的研究，例如 Sipser(2006)。接下来的介绍性说明总结了这套通用理论的基本理念，这也是和我们的讨论最相关的内容。

一个算法的实际运行时间(以秒为衡量单位)取决于具体实现的机器(例如，一个机器 CPU 的时钟速度)。为了避免对具体机器的依赖，一般讨论的是渐近意义下的算法运行时间，例如，归并排序算法的计算复杂度为 $O(n\log n)$。这意味着，在任意机器上实现这个算法，都可以满足可接受的抽象计算模型的要求，同时其实际运行时间满足如下条件：存在一个依赖于实际机器的常数 c 和 n_0，使得对于任意的 $n>n_0$，对 n 个元素排序的实际运行时间至多为 $cn\log n$。通常用可行或者计算有效性来指称复杂度为 $O(p(n))$(其中 p 为多项式)的算法。注意到这类讨论取决于对具体应用的问题所定义的输入规模 n。正如一般的讨论计算复杂度的文献提到的，纯算法领域中的输入规模有清晰的定义：算法得到一个输入，也就是一个等待排序的列表，或者一些等待计算的代数操作，这些都有良好的规模定义(也就是能用比特衡量其表现形式的规模)。但是对于机器学习的任务而言，输入规模的记号有些模糊，因为一个机器学习算法旨在提取数据集的模式，通常只能得到数据集的随机代表。

73

我们从这个问题和定义机器学习的算法复杂度入手。我们面为水平较高的学生提供了一个详细的规范定义。然后转向实现 ERM 规则的计算复杂度。我们首先给出了一些使 ERM 规则能够保持计算有效性的假设集，随之考虑一些 ERM 计算困难的案例，尽管这些案例实际上是可以有效学习的。接下来说明，ERM 实现的困难并不意味学习上的困难。最后，我们简短地论证了如何展示给定学习任务的困难度，即表明没有学习算法能够高效地解决它。

8.1 机器学习的计算复杂度

试想一个机器学习算法：它使用了域集 Z，假设集 $\mathcal{H}$，损失函数 ℓ，从 Z 中依照未知分布 $\mathcal{D}$ 独立同分布地抽取的训练集。在给定参数 ϵ，δ 的条件下，算法输出一个假设 h，它至少以概率 $1-\delta$ 满足下式：

$$L_{\mathcal{D}}(h) \leq \min_{h'\in\mathcal{H}} L_{\mathcal{D}}(h') + \epsilon$$

如上文提到的，算法的实际运行的时间取决于具体的机器。为了得到独立于机器的分析，我们运用标准的计算复杂性理论的方法。首先，我们依赖于一个抽象概念上的机器，

如图灵机(或实际机器上的图灵机(Blum，Shub 和 Smale，1989))。其次，我们从渐近的意义分析运行时间，而忽略常量因子；因此，只要实现了抽象机器，具体的机器并不重要。通常，渐近是相对于算法的输入规模而言的。例如，在之前提到的归并排序算法，运行时间是等待排序的元素个数的函数。

从学习算法的角度来讲，"输入规模"并没有明确的定义。有人也许把算法所接收的训练集的规模定义为输入规模，但这可能毫无意义。如果我们将大量样本输入算法，远远超过问题的样本复杂度，算法可能只会忽略多余的样本。因此，训练集的增大并不意味着学习问题变得更难，所以用于学习问题的运行时间不应该因为训练集的规模增大而增加。同理，可以将运行时间作为问题的参数的函数，包括：目标的精确度，该精确度的置信度，域的维度以及与算法输出进行比较的假设集的复杂度度量。

为了阐述这个问题，以一个学习轴对称矩形的学习算法为例。给定具体的 ε，δ 和实 74
例空间的具体维数，从而得到具体的轴对称矩形的学习任务。我们可以固定 ε，δ，使维数从 $d=2$，3，4，…变化，从而定义一连串的"学习矩形"的问题。我们也可以固定 d，δ，使目标精确度从 $\varepsilon=\frac{1}{2}$，$\frac{1}{3}$，…变化。当然也可以选取其他的问题序列。一旦一个序列确定，就分析运行时间关于这个序列的变量的渐进函数。

在引入正式的定义之前，还有一个细节需要讨论。在前面的基础上，学习理论可以通过将计算的负担转嫁到输出假设集上进行"欺骗"。比如，算法可以简单地将输出假设集定义为存储了训练集的函数，从而每当接受一个测试样本 x，算法在训练集上执行 ERM 算法并且用之于 x。注意在这种情况下，算法有一个固定的输出(也就是我们刚才描述的函数)并且可以在常数时间内运行完毕。但是，学习依旧是困难的——现在难处在于用输出的分类器去得到标签的预测。为了防止这种"欺骗"，我们需要规定用学习算法的输出，来为一些新的样本做标签预测所花费的时间不应该超过学习(也就是从训练样例中计算得到输出分类器)的运行时间。高水平的读者会在下一小节找到计算复杂度的正式定义。

正式的定义*

接下来的定义基于在底层抽象机器上的记号，这些记号可以用于图灵机或者基于实际机器上的图灵机。我们用算法实施的操作数量来衡量其计算复杂度，前提是：对不同的底层抽象机器的实现机器都存在常数 c，使得这些操作都可以在 c 秒钟完成。

定义 8.1(机器学习算法的计算复杂度) 我们分两步定义机器学习算法的复杂度。首先对一个固定学习问题(由三元组$(Z, \mathcal{H}, \ell)$——域集、假设集和损失函数决定)定义计算复杂度。第二步，我们考虑一系列相似任务的复杂度的变化情况。

1. 考虑函数 $f:(0,1)^2\to\mathbb{N}$，学习任务$(Z, \mathcal{H}, \ell)$，学习算法A。称A在 $O(f)$时间内是可学习的，如果存在常数 c，使得对任意在 Z 上的概率分布$\mathcal{D}$，给定输入 ε，$\delta\in(0,1)$，当A从$\mathcal{D}$中独立同分布地获取样本后，

- A在执行了至多 $cf(\varepsilon, \delta)$操作后终止；
- 记A的输出为 h_A，用它对新的样本进行标注的时候至多需要 $cf(\varepsilon, \delta)$操作； 75
- A的输出是概率意义上的精确：也就是说，在A接收的样本集上，至少以概率 $1-\delta$ 使得 $L_{\mathcal{D}}(h_A)\leq\min_{h'\in\mathcal{H}}L_{\mathcal{D}}(h')+\varepsilon$。

2. 考虑一系列的学习问题$(Z_n, \mathcal{H}_n, \ell_n)_{n=1}^{\infty}$，其中问题 n 由域集 Z_n、假设集$\mathcal{H}_n$、损失函数ℓ_n 定义。A表示解决这一系列机器学习问题的算法。给定函数 $g:\mathbb{N}\times(0,1)^2\to\mathbb{N}$，我

们认为A的运行时间是$O(f_n)$。如果对任意的n，A能够在$O(f_n)$时间内解决问题$(Z_n, \mathcal{H}_n, \ell_n)$，其中$f_n:\{0,1\}^n \to \mathbb{R}$定义为$f_n(\varepsilon, \delta)=g(n, \varepsilon, \delta)$。

我们称A学习有效，如果对于序列$(Z_n, \mathcal{H}_n, \ell_n)$，它的运行时间是$O(p(n, 1/\varepsilon, 1/\delta))$，其中$p$是多项式。

从定义中可以看到，一个通用学习问题能不能高效地解决，取决于它是不是能够分解成为特定的学习问题序列。例如，考虑学习有界假设集的问题。在前面的章节中，训练样本规模$m_{\mathcal{H}}(\varepsilon, \delta)=\log(|\mathcal{H}|/\delta)/\varepsilon^2$保证了ERM规则是$(\varepsilon, \delta)$-可学习的。假设在一个样本上验证一个假设占用常数时间，那么在$O(|\mathcal{H}|m_{\mathcal{H}}(\varepsilon, \delta))$时间对$\mathcal{H}$的穷举来实现ERM规则是可行的。对任意固定有限的$\mathcal{H}$，穷举算法花费一个多项式量级的时间。更进一步，如果我们定义$|\mathcal{H}_n|=n$的问题序列，那么遍历搜索也会是高效的。但是，如果我们定义$|\mathcal{H}_n|=2^n$的问题序列，样本复杂度仍然是关于n的多项式，但是遍历算法计算复杂度却是随n呈几何级数增长(因此认为是低效的)。

8.2 ERM规则的实现

给定假设集$\mathcal{H}$，$\mathrm{ERM}_{\mathcal{H}}$规则是最自然的学习样式。此外，对可学习的二分类问题，ERM规则均行之有效。本节我们在几个假设集上讨论实现ERM规则的计算复杂度。

给定域集Z，假设集$\mathcal{H}$，损失函数ℓ，相应的$\mathrm{ERM}_{\mathcal{H}}$规则如下定义：

在有限输入样本集$S\in Z^m$上，输出$h\in\mathcal{H}$满足经验风险的最小化：

$$L_S(h)=\frac{1}{|S|}\sum_{z\in S}\ell(h,z)$$

76 这节主要研究在几个学习任务的样本集上ERM规则的运行时间。

8.2.1 有限集

将假设集限制在有限集上是一个合理的轻微限制。例如，$\mathcal{H}$可能是所有用C++程序在至多10 000比特编码下实现的预测器的集合。其他有用的有限集是可以用有限参数表征的假设，每个参数的表现形式都可以用有限比特来完成，例如$\mathbb{R}^d$空间中轴对称矩形的坐标集合，这些限定矩形的参数在限制精度下是可以具体化的。

从前面的章节我们知道，学习有限集的机器学习问题中，抽样复杂度有一个上界：$m_{\mathcal{H}}(\varepsilon, \delta)=c\log(c|\mathcal{H}|/\delta)/\varepsilon^2$，其中$c=1$的情况是可实现的，而$c=2$的情况是不可实现的。因此抽样复杂度轻度依赖于$\mathcal{H}$的规模。在前面提到的C++程序中，假设集元素有$2^{10\,000}$个，但是抽样复杂度仅为$c(10\,000-\log(c/\delta))/\varepsilon^2$。

在有限假设集上实现ERM的一个直接方式就是实施遍历。也就是说，对每一个假设$h\in\mathcal{H}$，我们计算一个经验风险$L_S(h)$，然后返回一个使经验风险最小的假设。假定在单个样本上评估$\ell(h, z)$花费一个常数时间k，则遍历花费时间为$k|\mathcal{H}|m$，其中m是训练集的规模。如果令m代表抽样复杂度的上界，那么运行时间即为$k|\mathcal{H}|c\log(c|\mathcal{H}|/\delta)/\varepsilon^2$。

运行时间随假设集的规模大小线性增长，这使得遍历的方法在大规模集合上变得低效(不现实)。我们将序列问题正式定义为$(Z_n, \mathcal{H}_n, \ell_n)_{n=1}^{\infty}$，其中$\log|\mathcal{H}_n|=n$，那么遍历的算法需要指数运行时间。在C++程序样例中，如果$\mathcal{H}_n$是在C++程序中用n比特代码表示的函数集，则运行时间随n指数增长，这意味着遍历在实际运用是不现实的。事实上，这是我们为什么处理其他假设集(比如线性预测器)的原因，我们将在接下来的章节关注这些假设集，而不仅仅是有限假设集。

重要的是，要认识到一种算法（比如遍历）的低效并不意味没有高效的 ERM 实现方式。事实上，我们将会给出 ERM 能够高效实施的范例。

8.2.2　轴对称矩形

令$\mathcal{H}$是高维空间$\mathbb{R}^n$里矩形的坐标，也就是说，

$$\mathcal{H}_n = \{h_{(a_1,\dots,a_n,b_1,\dots,b_n)} : \forall i, a_i \le b_i\}$$

其中，

$$h_{(a_1,\dots,a_n,b_1,\dots,b_n)}(x,y) = \begin{cases} 1 & \text{若 } \forall i, x_i \in [a_i, b_i] \\ 0 & \text{其他} \end{cases} \tag{8.1}$$

77

1. 可高效学习的可行情况

试想实现 ERM 的现实情况。我们给定训练集 $S=(x_1, y_1), \dots, (x_m, y_m)$，存在一个轴对称的矩形 $h \in \mathcal{H}_n$，其中对任意 i 均有 $h(x_i)=y_i$。我们的目标是找到一个零训练误差的矩形，也就是说，一个符合 S 上所有标签的矩形。

我们将证明这在 $O(mn)$ 时间内可行。事实上，对每个 $i \in [n]$，令 $a_i = \min\{x_i : (x, 1) \in S\}$ 和 $b_i = \max\{x_i : (x, 1) \in S\}$。总之，我们让 a_i 是 S 的正样例中第 i 个分量的最小值，而 b_i 为最大值。验证这样的矩形是零训练误差的并不困难，并且找到 a_i 和 b_i 的运行时间是 $O(m)$。因此，总运行时间为 $O(nm)$。

2. 不可高效学习的未知情况

在未知的情况下，我们并不假定某些假设 h 完美地预测了所有训练集上的样本的标签。我们的目标是找到假设 h 来最小化 $y_i \ne h(x_i)$。对于很多普通的假设集，包括这里考虑的轴对称矩阵的假设集，在未知情况下实施 ERM 规则是 NP 难的问题（在大多数情况下，找到某个 $h \in \mathcal{H}$ 使其误差不大于常数 $c>0$ 乘以$\mathcal{H}$中经验风险最小化的误差，甚至是 NP 难的）。也就是说，除非 P=NP，否则没有以 m，n 为参数的多项式时间算法来保证找到一个 ERM 假设来解决这些问题（Ben David，Eiron & Long 2003）。

另一方面，值得注意的是，如果我们固定特定的假设集，比如给定维数 n 的轴对称矩形，那么存在针对这类假设集的高效算法。也就是说，成功的未知情况下的 PAC 学习器能够在关于 $1/\varepsilon$，$1/\delta$ 的多项式时间内运行结束（但是其关于维数 n 的依赖却不是多项式的）。

为了论述这个结论，回忆我们在可行情况下 ERM 规则的实现：为了确定轴对称矩形至多需要 $2n$ 个样例。因此，考虑规模 m 的训练集，我们在每个规模最大为 $2n$ 的子集上建立一个矩形。然后选择其中最小化训练误差的矩形。这个流程保证可以找到最小风险假设，并且运行时间是 $m^{O(n)}$。因此可以得出结论：如果 n 是固定的，运行时间就是关于样本规模的多项式。这和上述的困难性的结果并不相悖，因为可以定论：除非 P=NP，否则没有依赖于维数 n 的多项式算法。

8.2.3　布尔合取式

从$\mathcal{X}=\{0,1\}^n$映射到$\mathcal{Y}=\{0,1\}$的布尔合取式可以表达为命题形式 $x_{i_1} \wedge \dots \wedge x_{i_s} \wedge \neg x_{j_1} \wedge \dots \wedge \neg x_{j_r}$，其中 $i_1, \dots, i_s, j_1, \dots, j_r \in [n]$。这样一个表达式定义的函数为：

78

$$h(x) = \begin{cases} 1 & \text{若 } x_{i_1} = \dots = x_{i_s} = 1 \text{ 且 } x_{j_1} = \dots = x_{j_r} = 0 \\ 0 & \text{其他} \end{cases}$$

令$\mathcal{H}_n^c$表示$\{0,1\}^n$上所有可能的布尔合取式集合，其规模至多为 3^n+1（因为在合取式

中，x 的每个元素或者出现，或者结合负标记出现，或者根本不出现，并且我们有全部负标记公式）。因此，运用 ERM 规则学习$\mathcal{H}_C$的样本复杂度至多为$d\log(3/\delta)/\epsilon$。

1. 可高效学习的可行情况

接下来，我们将论述对$\mathcal{H}_C$施用 ERM 规则可以在 n，m 的多项式时间内实现。采取的方法是定义包含了所有元素中样本不相悖的合取式。令 v_1，…，v_{m^+} 表示所有的正标记样本 S。我们定义一半假设序列，下标为 $i\leq m^+$。令 h_0 表示所有可能元素的合取式，即$h_0=x_1\wedge\neg x_1\wedge x_2\wedge\neg x_2\wedge\cdots\wedge x_n\wedge\neg x_n$。注意到，这里 h_0 对$\mathcal{X}$中的样本标记为 0。$h_i=1$ 是从合取式 h_i 中删除所有不满足 v_{i+1} 的元素。这个算法输出假设 h_{m^+}。注意到 h_{m^+} 将样本 S 中所有的正样本标记为正。不仅如此，对任意的 $i\leq m^+$，h_i 是将 v_1，…，v_i 等标记为正的最严格的合取式。由于我们考虑的学习任务在可实现的前提下，那么存在一个合取式假设 $f\in\mathcal{H}_C$，其与 S 中所有样本是一致的。由于 h_{m^+} 是将正样本标记为正的最严格的合取式，任何被 f 标记为 0 的样例将会被 h_{m^+} 标记为 0。结论就是 h_{m^+}（关于 S 的）训练误差为 0，从而是一个合理的 ERM 假设。注意到算法运行时间为 $O(mn)$。

2. 不可高效学习的未知情况

同轴对称矩形的情况一样，除非 P=NP，没有一种算法，既符合运行时间是关于 m，n 的多项式，又能够保证找到不可知情况下的布尔合取式备选集中的最小风险假设。

8.2.4 学习三项析取范式

接下来我们论述，即使是在可实现的情况中，布尔合取式的轻微泛化也会导致 ERM 问题难以解决。取一个三项析取范式（3 项 DNF）的集合。实例空间为$\mathcal{X}=\{0,1\}^n$。每一个假设都可以用布尔逻辑式表达为 $h(x)=A_1(x)\vee A_2(x)\vee A_3(x)$，其中每个 $A_i(x)$都是布尔合取式（如前面一小节所定义的）。当 $A_1(x)$或者 $A_2(x)$或者 $A_3(x)$取值为 1，$h(x)$的输出是 1。如果三个合取式的输出都是 0，那么 $h(x)=0$。

令$\mathcal{H}_{3DNF}$表示 3 项 DNF 表达式的假设集，其规模最多为 3^{3n}。因此，运用 ERM 规则学习$\mathcal{H}_{3DNF}$的样本复杂度至多为 $3n\log(3/\delta)/\epsilon$。

但是，从计算的角度来看，这个学习问题是困难的。已经证明（参考 Pitt & Valiant 1988，Kearns，Schapire & Sellie 1994），除非 RP=NP，否则不存在多项式时间算法能够"合适"地学习 3 项 DNF 序列问题——其中第 n 个问题的维数为 n。我们说"合适"，就暗示了算法必须输出一个形式为 3 项 DNF 的假设。具体而言，因为 $\mathrm{ERM}_{\mathcal{H}_{3DNF}}$ 输出了一个三项 DNF，所以是一个合适的学习器，因此实现上是困难的。其证明运用了三色图问题的约简。细节的技术在习题 8.4 给出，同时也可以参考文献（Kearns 和 Vazirani 1994，1.4 节）。

79

8.3 高效学习，而不通过合适的 ERM

在前面的小节我们提到，不可能在 3 项析取式的备选集$\mathcal{H}_{3DNF}$上高效地使用 ERM 规则。在这一节我们将论证这个集合是可能被高效学习的，但需要在一个更大的集合上使用 ERM 规则。

表示独立学习是不难的

接下来我们论述高效学习 3 项 DNF 的可能性。其与前面小结所述的困难性结论并无矛盾，因为我们在这里允许"表示独立的"学习。也就是说，我们允许学习算法输出一个并不是 3 项 DNF 的假设。原始的想法是用一个更大的可以方便学习的假设集取代原先的 3 项 DNF 公式的假设集。学习算法可能返回一个并不属于原来的假设集的假设，因此名字

叫做"表示独立学习"。我们强调，在绝大多数的情况下，得到一个泛化能力强的假设才是我们在实践中真正感兴趣的。

首先注意因为∨分布在∧之中，每一个3项DNF可以写成：

$$A_1 \vee A_2 \vee A_3 = \bigwedge_{u \in A_1, v \in A_2, w \in A_3} (u \vee v \vee w)$$

接下来，我们定义：$\phi: \{0,1\}^n \to \{0,1\}^{(2n)^3}$，使得对所有的元素三元组 u、v、w，ϕ 存在一个变量表示 $u \vee v \vee w$ 是真还是假，所以，每个 $\{0,1\}^n$ 上的3项DNF对应 $\{0,1\}^{(2n)^3}$ 上的合取式，更进一步，在更高维度的空间里学习合取式集合的样本复杂度至多为 $n^4 \log(1/\delta)/\varepsilon$。因此，该方法的运行时间是关于 n 的多项式。

其直观的原理如下：我们从一个难以学习的假设集开始，转换到另一种假设集更大但是有更多结构信息的表示上，其允许一个更高效的ERM遍历算法。在新的表示上，解决这个ERM问题是简单的。

90

8.4 学习的难度*

我们刚刚论证了 $\mathrm{ERM}_{\mathcal{H}}$ 上计算难以实现并不意味着 $\mathcal{H}$ 是不能学习的。那么我们如何证明一个学习问题是计算难的呢？

一个途径是依赖密码学的假设。从某种意义上来说，密码和机器学习是对立的概念。在机器学习中，我们试图从观测样例中发现背后的某些规律，而在密码学中，目标是确保即使有人获取到一些局部信息，也无法成功解密。从高度直观的角度来看，对某些系统的安全加密导致相应的任务不可学习的性质。遗憾的是，目前没人能证明密码协议是不可破译的，甚至通常假设P≠NP也不足以证明(虽然这个假设已经被证明对于一般的密码学理论是必要的)。通常证明密码协议安全的方法是通过密码学假设。越多地使用这些假设作为密码学基础，我们就更加坚信其正确性(至少违背它们的算法很难找到)。

下面简单地阐述如何从密码学假设推导机器学习难度的基本原理。很多加密系统依赖于单向函数的假设。简单来讲，单向函数是映射 $f: \{0,1\}^n \to \{0,1\}^n$(更加严谨地来讲，它是一系列对应于每个维度 n 的函数)，这个映射计算上简单，然而其逆运算很难。更加正式地讲，f 能够在关于 n 的多项式时间内计算，但是对任意随机多项式时间算法 A 和任意的多项式 $p(\cdot)$，

$$\mathbb{P}[f(A(f(x))) = f(x)] < \frac{1}{p(n)}$$

其中概率是依据 $\{0,1\}^n$ 上的均匀分布和算法 A 的随机性选取 x 的概率。

一个单向函数 f 称为陷门单向函数，如果对一些多项式 p，对任意 n 存在一个长度小于或者等于 $p(n)$ 的比特串 s(称为秘钥)，使得存在一个多项式时间算法可以对任意的 n

和$x\in\{0,1\}^n$，输入$(f(x),s_f)$可以输出x。换言之，尽管f是难以逆运算的，一旦有了它的秘钥，逆运算就是可行的。这些函数就可以用它们的秘钥进行表征。

81

现在，令F_n表示$\{0,1\}^n$上的陷门函数族，其元素能在多项式时间内计算。即，我们固定一个算法，给定一个秘钥(表征F_n中一个函数)和一个输入向量，其在多项式时间内可以计算出对应于秘钥和输入向量的值。

取学习其逆函数的族，$\mathcal{H}_F^n=\{f^{-1}:f\in F_n\}$。因为族中的每个函数可以通过规模关于$n$的多项式的秘钥$s_f$来逆运算，那么族$\mathcal{H}_F^n$也可以通过这些秘钥表征且其规模最多为$2^{p(n)}$。因此其样本复杂度是关于$n$的多项式。我们所定没有高效的学习器可以学习这样的备选集。如果存在这样一个学习器L，那么在$\{0,1\}^n$上均匀地随机抽取多项式规模的比特串，在其上计算f可以得到$(f(x),x)$的标记样本，这些样本足以使学习器得到一个满足(ϵ,δ)近似的f^{-1}(由于在f上的随机分布)，这和f的单向特性相悖。

在Kearns和Vazirani(1994)的书中第6章对此有更细致的讨论，并给出了具体的例子。运用约简的方式，该书还证明了可以用布尔电路计算的函数族并不能高效学习，即使是在可实现的情况下。

8.5 小结

可以将机器学习算法的运行时间视作学习问题的不同参数的函数来渐进地分析，参数包括假设集的规模、精确性的度量方式、置信度的度量方式、域集的规模。我们论证了ERM规则能够高效实现的案例，例如，在可实现的假设下，推导了解决布尔合取式和轴对称矩形的高效算法。但是，在不可知的情况下，对这些假设集实现ERM规则是NP困难的。回到统计的观角，可实现和不可知的情况没有差别(即，一个备选集在这两种情况下能不能被学习当且仅当其VC维是有限的)。相反如我们所见，从计算的视角来看，这种差别却相当大。同时另一个学习3项DNF的例子说明，即使在可实现的前提下，实现ERM规则依然是困难的，但是这个备选集可以被其他的算法有效学习。

在一些自然的假设集上实现ERM规则的困难性驱使一些代替的学习算法的发展，我们将在以后的章节讨论这些学习算法。

8.6 文献评注

Valiant(1984)引入了高效的PAC学习模型，限定了算法的运行时间是关于$1/\epsilon$、$1/\delta$和假设集的表示规模的多项式。细致的讨论和详尽的参考文献要点参见Kearns和Vazirani(1994)。

82

8.7 练习

8.1 令$\mathcal{H}$表示直线上的区间(相当于1维的轴对称矩形)，在不可知的情况下，提出实现$\mathrm{ERM}_{\mathcal{H}}$的方式，且给定训练集规模为$m$，学习时间为$O(m^2)$。

提示：采用动态规划。

8.2 令$\mathcal{H}_0$，$\mathcal{H}_1$，…表示二分类问题的假设集序列。假设存在一个在可实现情况下实现ERM规则的学习算法，且其对每个$\mathcal{H}_n$输出的假设仅仅取决于训练集上$O(n)$个学习样本。更进一步，假定输出的假设能在$O(n)$时间内从这$O(n)$个样例中得到，并且每个假设的经验风险可以在$O(mn)$时间计算得到。例如，如果$\mathcal{H}_n$是$\mathbb{R}^n$上轴对称矩形的假设集，那么在可实现的情况下可以学习得到一个由至多$2n$个样例定义的

ERM 假设。证明在这种情况下，对$\mathcal{H}_n$在不可实现的情况下可以在$O(mnm^{O(n)})$找到 ERM 假设。

8.3 在这个练习中，我们展示几个备选集，在其上建立 ERM 分类器是计算困难的。首先，我们引入 n 维半空间备选集 HS_n，样本集为$\mathcal{X}=\mathbb{R}^n$。这个备选集是具有如下形式函数的集合：$h_{w,b}(x)=\mathrm{sign}(\langle w, x\rangle+b)$，其中 $w, x\in\mathbb{R}^n$，并且$\langle w, x\rangle$是内积，$b\in\mathbb{R}$。在第9章中有更详尽的讨论。

1) 论述在备选集$\mathcal{H}=HS_n$上实现 $\mathrm{ERM}_{\mathcal{H}}$ 的线性分类器是计算难的。更加精确地讲，我们考虑随着维数 n 线性增长的问题序列，其样例的数量 m 是 n 的常数倍。

提示：可以通过如下问题的约简来证明其难度：

最大 FS：给定线性不等式系统 $Ax>b$，其中 $A\in\mathbb{R}^{m\times n}$，$b\in\mathbb{R}^m$（也就是说，这个系统包括了 m 个 n 元线性不等式，$x=(x_1, \dots, x_n)$），找到其会有尽可能多的有解（称这样的子系统为可行的）的不等式的子系统。

已经证明最大 FS 问题是 NP 难的（Sankaran 1993）。

论述在训练集 $S\in(\mathbb{R}^n\times\{+1, -1\})^m$，任何学习 ERM_{HS_n} 假设的算法都可以用来解决规模为 m，n 的最大 FS 问题。

提示：定义一个映射变换线性不等式的 n 个变量到 $\mathbb{R}^n$ 标记点上，另一个映射变换 $\mathbb{R}^n$ 的矢量到半空间，使得向量 w 满足不等式 q 当且仅当标记点对应的 q 位是由对应于 w 的半空间分类的。证明：对于半空间的经验风险最小化的问题是 NP 难的（即，如果它可以在关于样本大小 m 和欧氏维数 n 的多项式时间内解决，则每一个 NP 类问题可以在多项式时间内解决）。

2) 令$\mathcal{X}=\mathbb{R}^n$，$\mathcal{H}_k^n$表示 k 个线性半空间的交点。在这个练习中，我们希望证明 $\mathrm{ERM}_{\mathcal{H}_k^n}$ 对任意 $k\geq 3$ 是计算困难的。精确来讲，考虑问题序列，其中 $k\geq 3$ 是常数且 n 线性增长。训练集规模 m 随 n 线性增长。为了证明结论，考虑如下定义的图的 k-着色问题：

给定图 $G=(V, E)$，常数 k，推断是否存在函数 $f: V\to\{1\cdots k\}$，使得对任意的$(u, v)\in E$，$f(u)\neq f(v)$。

k-着色问题对任意 $k\geq 3$ 是 NP 难的（Karp 1972）。我们希望约简 k-着色问题到 $\mathrm{ERM}_{\mathcal{H}_k^n}$：即证明如果有一个算法在关于 k，n 和采样规模 m 的多项式时间内解决，那么有一个多项式算法可以解决图 k-着色问题。

给定图 $G=(V, E)$，令$\{v_1, \dots, v_n\}$表示 V 中的顶点。建立样例 $S(G)\in(\mathbb{R}^n\times\{+1\})^m$，其中 $m=|V|+|E|$，同时：

第一，对任意 $v_i\in V$，建立负标记的样例 e_i；

第二，对任意边$(v_i, v_j)\in E$，建立正标记的样例$(e_i+e_j)/2$。

①证明如果存在 $h\in\mathcal{H}_k^n$在 $S(G)$是零训练误差的，那么 G 是可以 k-着色的。

②证明如果 G 是可以 k-着色的，那么存在 $h\in\mathcal{H}_k^n$在 $S(G)$是零训练误差的。

③在前述的基础上，证明对任意的 $k\geq 3$，任意的 $\mathrm{ERM}_{\mathcal{H}_k^n}$是 NP 难的。

8.4 在此练习中，我们表明，解决 ERM 的难度相当于合适的 PAC 学习的难度。回想一下，我们称算法"合适"意味着它必须从假设类输出一个假设。形式化这种说法，我们首先需要以下定义：

定义 8.2 复杂性类随机多项式（RP）时间是所有有左概率算法（即，算法运行时允许进行抛硬币）的决策问题（即，问题的任何实例都要求回答是或者否）的集合，且必

83

须满足如下特性：

第一，对任意输入实例，算法运行时间是输入规模的多项式时间；

第二，如果正确的回答为否，算法返回否；

第三，如果正确的回答为是，算法以概率 $a\geq 1/2$ 返回是，以概率 $1-a$ 返回否[⊖]。

明显，RP 类包含 P 类，RP 类包含于 NP 类。但是并不清楚三者中间是不是存在任何等量关系。大家普遍承认的是 NP 类严格大于 RP 类，即 NP 难问题没有随机多项式时间算法。

证明如果一个假设类$\mathcal{H}$是可以被多项式时间算法合适 PAC 学习的，那么 $\mathrm{ERM}_{\mathcal{H}}$ 问题是 RP 类问题。特别地，如果一个 $\mathrm{ERM}_{\mathcal{H}}$ 问题是 NP 难的（例如上一个题目讨论的半空间的交点），那么除非 NP=RP，否则不存在$\mathcal{H}$的多项式时间算法的合适的 PAC 学习器。

64 ? 85

⊖ 定义中的分数 1/2 可以被(0，1)中的任意常数代替。

第二部分

Understanding Machine Learning: From Theory to Algorithms

从理论到算法

第9章

Understanding Machine Learning: From Theory to Algorithms

线性预测

本章我们将学习线性预测，它是假设类中最重要的成员。许多广泛应用的学习算法都基于线性预测，最重要的原因是它能在许多情形下有效地学习。此外，线性预测具有直观性，易于理解，在许多天然的学习问题中对数据拟合良好。

我们将介绍一些属于线性预测的假设类：半空间法(halfspace)，线性回归预测，逻辑斯蒂回归预测等，同时也介绍相关的学习算法：线性规划，半空间中的感知器算法和线性回归中的最小均方算法。本章通过经验风险最小化方法(ERM 方法)研究线性预测。同时，在接下来的几章，我们也通过另外的范例来学习这些假设类。

首先，我们定义仿射函数类：

$$L_d = \{h_{w,b} : w \in \mathbb{R}^d, b \in \mathbb{R}\}$$

其中，

$$h_{w,b}(x) = \langle w, x\rangle + b = \left(\sum_{i=1}^{d} w_i x_i\right) + b$$

使用这个记号将很方便：

$$L_d = \{x \mapsto \langle w, x\rangle + b : w \in \mathbb{R}^d, b \in \mathbb{R}\}$$

可以这样解读：L_d 是函数集合，其中每个函数被 $w \in \mathbb{R}^d$ 和 $b \in \mathbb{R}$ 参数化，并以向量 x 作为输入，以标量 $\langle w, x\rangle + b$ 作为输出。

线性预测中另一类不同的假设类是由 L_d 中的函数 $\phi: \mathbb{R} \to \mathcal{Y}$ 组成的。例如，在二分类中，我们将 ϕ 选取为符号函数，在回归问题中 $\mathcal{Y} = \mathbb{R}$，ϕ 可以是恒等函数。

将偏移量 b 包含在 w 中将更为方便，只需在 w 内加入一维并将 $x \in \mathcal{X}$ 对应加入全为 1 的一维，即，令 $w' = (b, w_1, w_2, \cdots, w_d) \in \mathbb{R}^{d+1}$ 且 $x' = (1, x_1, x_2, \cdots, x_d) \in \mathbb{R}^{d+1}$，从而

$$h_{w,b}(x) = \langle w, x\rangle + b = \langle w', x'\rangle$$

一个推论是使用类似的变换且在输入向量中加入为 1 的常量，$\mathbb{R}^d$ 中的任何一个仿射函数均可写成 $\mathbb{R}^{d+1}$ 中的齐次线性函数。因此，当这种表示可以化简时，我们将忽略偏移量，认为 L_d 是一类形式为 $h_w(x) = \langle w, x\rangle$ 的齐次线性函数。

全书中我们使用“线性函数”代表仿射函数和(齐次)线性函数。

9.1 半空间

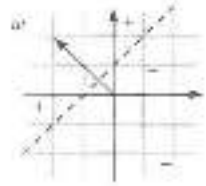
我们考虑的第一个假设类是半空间类，它为二分类而设计。即 $\mathcal{X} = \mathbb{R}^d$，$\mathcal{Y} = \{-1, +1\}$。半空间类的定义如下：

$$HS_d = \operatorname{sign} \circ L_d = \{x \mapsto \operatorname{sign}(h_{w,b}(x)) : h_{w,b} \in L_d\}$$

换言之，每一个 HS_d 半空间假设均被 $w \in \mathbb{R}^d$ 和 $b \in \mathbb{R}$ 参数化，当输入一个向量 x 时，假设返回一个标签 $\operatorname{sign}(\langle w, x\rangle + b)$。

为几何化的阐述这类假设，我们可以选取 $d = 2$ 的情形。每一个假设形成一个与向量 w 垂直的超平面，并且与纵轴相交于点 $(0, -b/w_2)$。那些在超平面上方的，即与 w 成锐

角的样本，被标记为正样本；那些在超平面下方的，即与 w 成钝角的样本，被标记为负样本。



在 9.1.3 节中，我们将给出 $\mathrm{VCdim}(HS_d)=d+1$。这意味着只要样本量为 $\Omega\left(\frac{d+\log(1/\delta)}{\epsilon}\right)$，我们就可以通过 ERM 范式学习半空间。因此，我们现在讨论半空间 ERM 方法。

接下来我们介绍两种方法寻找可行的 ERM 半空间。在半空间的概念中，可行被认为是"可分"的，因为使用超平面完全区分正负样本是可能的。在不可分情形(例如未知情况)中使用 ERM 法则是难于计算的(Ben-David，Simon，2001)。有许多方法可以学习不可分数据，最流行的是使用替代损失函数(surrogate loss function)，即不必使用 0—1 损失最小化经验风险来学习半空间，而可以使用不同的损失函数。例如，我们将在 9.3 节中描述逻辑斯谛回归方法，它能在不可分情形中有效执行。我们会在第 12 章中详细学习替代损失函数。 [90]

9.1.1 半空间类线性规划

线性规划问题可以表述为在线性不等式约束下最大化线性函数，即：

$$\max_{w\in\mathbb{R}^d}\langle u,w\rangle$$
$$\text{s.t. } Aw\geq v$$

其中 $w\in\mathbb{R}^d$ 是我们希望求解的参数向量，A 是 $m\times d$ 维矩阵，$v\in\mathbb{R}^m$，$u\in\mathbb{R}^d$ 为向量。线性规划能被有效地求解[⊖]，此外，有公开的线性规划求解程序。

我们将证明，可分情形的半空间 ERM 问题可以表述成线性规划问题。不失一般性，我们假定为齐次情形。令 $S=(x_1,y_1)\ldots$ 为 m 维训练集。因为我们假定样本可分，训练集上的 ERM 预测是 0 误差的。即，我们可以寻找向量 $w\in\mathbb{R}^d$ 满足

$$\mathrm{sign}(\langle w,x_i\rangle)=y_i,\quad \forall i=1,\cdots,m$$

同样，我们可以找到向量 w 满足

$$y_i\langle w,x_i\rangle>0,\quad \forall i=1,\cdots,m$$

令 w^* 满足该条件(因为我们假定可分，因此它一定存在)。定义 $\gamma=\min_i(y_i\langle w^*,x_i\rangle)$ 并令 $\overline{w}=\frac{w^*}{\gamma}$。因此，对于所有的 i，我们有

$$y_i\langle\overline{w},x_i\rangle=\frac{1}{\gamma}y_i\langle w^*,x_i\rangle\geq 1$$

因此我们可以证明存在向量满足

$$y_i\langle w,x_i\rangle\geq 1,\quad \forall i=1,\cdots,m \tag{9.1}$$

⊖ 即在 m，d 的多项式时间内，以及在实数的表示长度上。

显然，向量为 ERM 预测。

为找到向量满足式(9.1)，我们可以依最线性规划求解。集合 A 为 $m\times d$ 维矩阵，它的行样本乘 y_i。即 $A_{i,j}=y_i x_{i,j}$，其中 $x_{i,j}$ 是 x_i 的 j 阶元素。令 v 为 $(1,\cdots,1)\in\mathbb{R}^d$ 向量，那么式(9.1)可以写成

$$Aw\geq v$$

线性规划形式需要最大化目标，但所有满足该约束的 w 均为假设输出的候选。因此，我们设定一个“虚拟”的目标，$u=(0,\cdots,0)\in\mathbb{R}^d$。

[91]

9.1.2　半空间感知器

另一个 ERM 法则的计算方法是感知器算法(Rosenblatt，1958)，感知器算法是迭代式的，它构建一系列的向量 $w^{(1)}$，$w^{(2)}$，…初始的 $w^{(1)}$ 设置为 0 向量。在第 t 次迭代，感知器找到被 $w^{(t)}$ 错分的样本 i，即，该样本使 $\mathrm{sign}(\langle w^{(t)},x_i\rangle)\neq y_i$。因此，通过将样本 x_i 乘比例系数 y_i 加入向量，感知器更新 $w^{(t)}$，使 $w^{(t+1)}=w^{(t)}+y_i x_i$。我们的目标是使对所有的 i 有 $y_i\langle w,x_i\rangle>0$，且

$$y_i\langle w^{(t+1)},x_i\rangle=y_i\langle w^{(t)}+y_i x_i,x_i\rangle=y_i\langle w^{(t)},x_i\rangle+\|x_i\|^2$$

因此，感知器的更新使其解对第 i 个样本变得“更加正确”。

感知器批处理算法

输入：训练集 (x_1,y_1)，…，(x_m,y_m)

初始化：$w^{(1)}=(0,\cdots,0)$

循环：$t=1,2,\cdots$

　如果($\exists i$ s.t. $y_i\langle w^{(t)},x_i\rangle\leq 0$)，**那么**

　　$w^{(t+1)}=w^{(t)}+y_i x_i$

　否则

　　输出 $w^{(t)}$

下面定理保证在可分情形时，该算法终止时所有样本均被正确分类。

定理 9.1　假定 (x_1,y_1)，…，(x_m,y_m) 是可分的，令 $B=\min\{\|w\|:\forall i\in[m],y_i\langle w,x_i\rangle\geq 1\}$，同时 $R=\max_i\|x_i\|$。那么，感知器算法最多在 $(RB)^2$ 次迭代终止，且终止时满足 $\forall i\in[m]$，$y_i\langle w^{(t)},x_i\rangle>0$。

证明　根据终止条件的定义，感知器终止时所有样本均被划分，我们将证明算法迭代次数 T 满足 $T\leq(RB)^2$，这意味着感知器最多运行 $(RB)^2$ 次迭代。

设 w^* 为 B 定义下最小的向量，即对所有的 i，有 $y_i\langle w^*,x_i\rangle\geq 1$，在所有满足这个约束的向量中，$w^*$ 具有最小范数。

证明的思想是在 T 次迭代后，w^* 与 $w^{(T+1)}$ 夹角余弦至少为 $\frac{\sqrt{T}}{RB}$。即

$$\frac{\langle w^*,w^{(T+1)}\rangle}{\|w^*\|\,\|w^{(T+1)}\|}\geq\frac{\sqrt{T}}{RB}\tag{9.2}$$

根据柯西-施瓦茨不等式，式(9.2)左侧最大为 1，因此，式(9.2)意味着

$$1\geq\frac{\sqrt{T}}{RB}\Rightarrow T\leq(RB)^2$$

我们需要证明它。 [92]

为证明式(9.2)成立，我们首先证明$\langle w^*, w^{(T+1)}\rangle \geq T$。显然，在第一步迭代$w^{(1)}=(0, \cdots, 0)$，有$\langle w^*, w^{(1)}\rangle=0$。在第$t$步迭代，如果我们使用样本$(x_i, y_i)$更新，将有

$$\begin{aligned}\langle w^*, w^{(t+1)}\rangle-\langle w^*, w^{(t)}\rangle &= \langle w^*, w^{(t+1)}-w^{(t)}\rangle \\ &= \langle w^*, y_i x_i\rangle = y_i\langle w^*, x_i\rangle \geq 1\end{aligned}$$

因此，在T次迭代之后，我们而有

$$\langle w^*, w^{(T+1)}\rangle = \sum_{t=1}^{T}(\langle w^*, w^{(t+1)}\rangle-\langle w^*, w^{(t)}\rangle) \geq T \tag{9.3}$$

下面，我们找到$\|w^{(T+1)}\|$的上界。对每步迭代t，我们有

$$\begin{aligned}\|w^{(t+1)}\|^2 &= \|w^{(t)}+y_i x_i\|^2 \\ &= \|w^{(t)}\|^2+2y_i\langle w^{(t)}, x_i\rangle+y_i^2\|x_i\|^2 \\ &\leq \|w^{(t)}\|^2+R^2\end{aligned} \tag{9.4}$$

其中，最后的不等式是因为样本i是使$y_i\langle w^{(t)}, x_i\rangle \leq 0$而必要的，且$x_i$的范数最大为$R$。现在，因为$\|w^{(1)}\|^2=0$，如果我们使用式(9.4)递归计算$T$次迭代，将有

$$\|w^{(T+1)}\|^2 \leq TR^2 \Rightarrow \|w^{(T+1)}\| \leq \sqrt{T}R \tag{9.5}$$

将式(9.3)代入式(9.5)中，并利用$\|w^*\|=B$，我们得到

$$\frac{\langle w^{(T+1)}, w^*\rangle}{\|w^*\|\|w^{(T+1)}\|} \geq \frac{T}{B\sqrt{T}R}=\frac{\sqrt{T}}{RB}$$

从而式(9.2)成立，证毕。 ■

评论 9.1 感知器方法简单并保证收敛，但收敛速率取决于参数B，它在某些情况下是随着d指数爆炸的。在这种情况，采用前文所述的线性规划解决ERM问题将更合适。然而，对于大部分天然的数据集，B将不会太大，感知器收敛还是相当快的。

9.1.3 半空间的VC维

我们从齐次情况出发，来完善半空间VC维理论。

定理 9.2 齐次半空间$\mathbb{R}^d$的VC维是d。

证明 首先我们考虑向量集合$e_1, \cdots, e_d$，其中e_i的第i个元素为1其余元素为零，这个集合被半空间类打散。显然，对于$y_1, \cdots, y_d$中每一个标签，给定$w=(y_1, \cdots, y_d)$有$\langle w, e_i\rangle=y_i(\forall i)$。 [93]

而后，令$x_1, \cdots, x_{d+1}$为$\mathbb{R}^d$中的$d+1$个向量的集合。那么，一定有非全部为零的实数$a_1, \cdots, a_{d+1}$，满足$\sum_{i=1}^{d+1}a_i x_i=\mathbf{0}$。令$I=\{i: a_i>0\}$且$J=\{j: a_j<0\}$，$I$，$J$不全是空的。首先我们假设它们均非空，即

$$\sum_{i\in I}a_i x_i=\sum_{j\in J}|a_j|x_j$$

现在假定$x_1, \cdots, x_{d+1}$被齐次类打散。那么必有向量w对于所有的$i\in I$满足$\langle w, x_i\rangle>0$，对$j\in J$有$\langle w, x_j\rangle<0$，进而有

$$0<\sum_{i\in I}a_i\langle x_i, w\rangle=\langle\sum_{i\in I}a_i x_i, w\rangle=\langle\sum_{j\in J}|a_j|x_j, w\rangle=\sum_{j\in J}|a_j|\langle x_j, w\rangle<0$$

这是个矛盾式。最后，如果J(或是I)为空，那么上式的右侧(或左侧)的不等号也会矛盾。 ■

定理 9.3　非齐次半空间 $\mathbb{R}^d$ 的 VC 维是 $d+1$。

证明　首先，就像证明定理 9.2 一样，容易知道向量集合 $0, e_1, \cdots, e_d$ 被非齐次半空间类打散。然后，假设向量 $x_1, \cdots, x_{d+2}$ 被非齐次半空间打散。但是，使用本章开始介绍的降维方法，$\mathbb{R}^{d+1}$ 空间中能被齐次半空间打散的向量有 $d+2$ 个，这与定理 9.2 矛盾。■

9.2　线性回归

线性回归是常用的统计工具，用来建立"解释性"变量与观测值之间的关系。从机器学习的角度说明，定义域 $\mathcal{X}$ 是 $\mathbb{R}^d$ 的 d 维子集，标签集 $\mathcal{Y}$ 是实数集。我们可以试图寻找一个线性函数 $h: \mathbb{R}^d \to \mathbb{R}$ 使参数之间的关系拟合最好（比如，通过儿童的年龄与出生重量的关系预测体重）。图 9.1 给出了 $d=1$ 时的线性回归。

线性回归假设类是线性函数的集合：

$$\begin{aligned}\mathcal{H}_{reg} &= L_d \\ &= \{x \mapsto \langle w, x\rangle + b : w \in \mathbb{R}^d, b \in \mathbb{R}\}\end{aligned}$$

接下来我们需定义回归的损失函数。在分类问题中损失函数的定义是显而易见的，即 $\ell(h,(x,y))$ 表明 $h(x)$ 是否对 y 正确分类。在回归中，如果儿童的体重是 3kg，那么预测为 3.00001kg 或 4kg 都是错的，但我们显然更倾向于前者。因此我们需要定义对 $h(x)$ 与 y 之间差异的惩罚力度。一个常用形式为平方损失函数，即

$$\ell(h,(x,y)) = (h(x)-y)^2$$

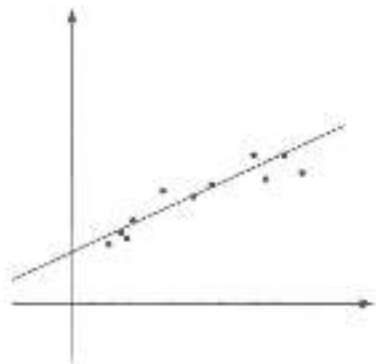


图 9.1　$d=1$ 时的线性回归。此时 x 轴代表儿童年龄，y 轴代表其体重

对应该惩罚函数，其经验风险函数叫均方误差，即

$$L_S(h) = \frac{1}{m}\sum_{i=1}^{m}(h(x_i)-y_i)^2$$

在下面部分，我们将看到如何使用平方损失执行线性回归的 ERM 准则。当然，有很多其他的损失函数可以使用，例如绝对值损失函数 $\ell(h,(x,y)) = |h(x)-y|$。使用绝对值损失函数的 ERM 准则可以采用线性规划方法（见练习 9.1）。

需要注意的是线性回归不是二分类问题，我们不能使用 VC 维理论分析其样本复杂性。一个可行的办法是使用离散化方法（见第 4 章注 4.1）。如果我们愿意将向量 w 和偏移量 b 用有限位数（比如 64 位的浮点数）表示，那么该假设类将为有限的，且其最大样本量为 $2^{O(d)}$。我们可以依靠第 4 章中样本复杂度确界分析假设类。然而需要注意的是，为应用第 4 章中的样本复杂度定界方法，损失函数也需要有界。本书的后续章节会讨论线性回归问题样本复杂度的更严格方法。

9.2.1　最小平方

95

最小平方算法是根据平方损失来求解线性回归假设类的 ERM 问题。这类 ERM 问题是给定训练集 S，使用齐次的 L_d 来找到

$$\operatorname*{argmin}_{w} L_S(h_w) = \operatorname*{argmin}_{w} \frac{1}{m}\sum_{i=1}^{m}(\langle w, x_i\rangle - y_i)^2$$

为解决这一问题，我们计算目标函数的梯度并将其与 0 比较。即，我们需求解

$$\frac{2}{m}\sum_{i=1}^{m}(\langle \boldsymbol{w},\boldsymbol{x}_i\rangle - y_i)\,\boldsymbol{x}_i = \mathbf{0}$$

我们可以将该问题重新表述为 $A\boldsymbol{w}=\boldsymbol{b}$，其中

$$A=\left(\sum_{i=1}^{m}\boldsymbol{x}_i\boldsymbol{x}_i^{\mathrm{T}}\right) \quad 且 \quad \boldsymbol{b}=\sum_{i=1}^{m}y_i\boldsymbol{x}_i \tag{9.6}$$

或以矩阵的形式

$$A=\begin{bmatrix}\vdots & & \vdots\\ \boldsymbol{x}_1 & \cdots & \boldsymbol{x}_m\\ \vdots & & \vdots\end{bmatrix}\begin{bmatrix}\vdots & & \vdots\\ \boldsymbol{x}_1 & \cdots & \boldsymbol{x}_m\\ \vdots & & \vdots\end{bmatrix}^{\mathrm{T}} \tag{9.7}$$

$$\boldsymbol{b}=\begin{bmatrix}\vdots & & \vdots\\ \boldsymbol{x}_1 & \cdots & \boldsymbol{x}_m\\ \vdots & & \vdots\end{bmatrix}\begin{bmatrix}y_1\\ \vdots\\ y_m\end{bmatrix} \tag{9.8}$$

如果 A 可逆，那么 ERM 问题的解为

$$\boldsymbol{w}=A^{-1}\boldsymbol{b}$$

在 A 不可逆的情形，求解则需要线性代数的一些标准工具，它们在附录 C 中给出。容易证明，如果训练样本不是充满整个 $\mathbb{R}^d$ 空间，那么 A 将是不可逆的。然而，我们总能找到系统 $A\boldsymbol{w}=\boldsymbol{b}$ 的解，因为 $\boldsymbol{b}$ 在 A 的范围内。事实上，因为 A 是对称阵，我们可以用特征值分解来表示它：$A=VDV^{\mathrm{T}}$，其中 D 是对角阵，V 是标准正交矩阵（即 $V^{\mathrm{T}}V$ 是 $d\times d$ 的单位阵）。定义 D^{+} 为对角阵，满足当 $D_{j,j}=0$ 时 $D^{+}_{j,j}=0$，否则 $D^{+}_{j,j}=1/D_{j,j}$。现在，定义

$$A^{+}=VD^{+}V^{\mathrm{T}} \quad 且 \quad \hat{\boldsymbol{w}}=A^{+}\boldsymbol{b}$$

令 $\boldsymbol{v}_j$ 为 V 的第 j 列，那么我们有

$$A\hat{\boldsymbol{w}}=AA^{+}\boldsymbol{b}=VDV^{\mathrm{T}}VD^{+}V^{\mathrm{T}}\boldsymbol{b}=VDD^{+}V^{\mathrm{T}}\boldsymbol{b}=\sum_{j:D_{j,j}\neq 0}\boldsymbol{v}_j\boldsymbol{v}_j^{\mathrm{T}}\boldsymbol{b}$$

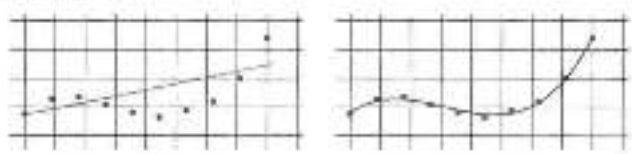
即，$A\hat{\boldsymbol{w}}$ 是 $\boldsymbol{b}$ 在那些满足 $D_{j,j}\neq 0$ 的向量 $\boldsymbol{v}_j$ 上的投影。因为 $\boldsymbol{x}_1,\cdots,\boldsymbol{x}_m$ 张成的线性空间与 $\{\boldsymbol{v}_j\}$ 张成的空间一致且 $\boldsymbol{b}$ 由 $\boldsymbol{x}_i$ 线性张成，我们有 $A\hat{\boldsymbol{w}}=\boldsymbol{b}$，它证明了我们的观点。 96

9.2.2 多项式线性回归

一些学习问题需要非线性预测，比如多项式预测。一个 n 阶一维多项式函数的例子是

$$p(x)=a_0+a_1x+a_2x^2+\cdots+a_nx^n$$

其中 $(a_1,\cdots,a_n)$ 是长度为 $n+1$ 的向量系数。下面我们表述一个训练集，使用 3 阶多项式拟合效果要优于线性预测。



我们这里关注 n 阶一维多项式回归类，即

$$\mathcal{H}^{n}_{\mathrm{poly}}=\{x\mapsto p(x)\}$$

其中 p 是 n 阶一维多项式，以系数向量 $(a_1,\cdots,a_n)$ 参数化。需要注意它是一个一维多项式回归问题，当 $\mathcal{X}=\mathbb{R}$，有 $\mathcal{Y}=\mathbb{R}$。

对于这类的一种学习方法是化简为我们已经介绍的线性回归问题。为将多项式回归转化为线性回归，我们定义映射 $\varphi:\mathbb{R}\to\mathbb{R}^{n+1}$ 使得 $\varphi(x)=(1,x,x^2,\cdots,x^n)$，那么我们有

$$p(\psi(x)) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n = \langle a, \varphi(x) \rangle$$

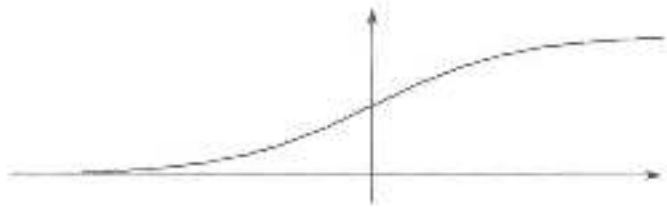
这样我们能够通过上文的最小平方算法找到系数向量 a 的最优解。

9.3　逻辑斯谛回归

在逻辑斯谛回归中，我们学习一簇函数 h 将 $\mathbb{R}^d$ 映射到 $[0, 1]$ 区间。然而，逻辑斯谛回归被用于分类任务：我们可以将 $h(x)$ 解读为 x 标签为 1 的概率。逻辑斯谛回归的假设类由 sigmoid 函数 $\phi_{sig}: \mathbb{R} \to [0, 1]$ 组成，而不是线性函数 L_d。特别地，逻辑斯谛回归中的 sigmoid 函数是逻辑斯谛函数，它定义为

[97]

$$\phi_{sig}(z) = \frac{1}{1 + \exp(-z)} \tag{9.9}$$



sigmoid 这个名字意味 S 形状，指上图所示的函数形状。因此假设类为（此处为简便，我们使用齐次线性函数）：

$$H_{sig} = \phi_{sig} \circ L_d = \{x \mapsto \phi_{sig}(\langle w, x \rangle) : w \in \mathbb{R}^d\}$$

需要注意的是，当 $\langle w, x \rangle$ 非常大则 $\phi_{sig}(\langle w, x \rangle)$ 趋近于 1，而 $\langle w, x \rangle$ 非常小则 $\phi_{sig}(\langle w, x \rangle)$ 趋近于 0。回顾半空间假设预测，其中与 w 相一致的符号为 $\langle w, x \rangle$。因此，当 $|\langle w, x \rangle|$ 很大时，半空间假设和逻辑斯谛假设的预测是相似的。然而，当 $|\langle w, x \rangle|$ 接近 0 时，我们有 $\phi_{sig}(\langle w, x \rangle) \approx \frac{1}{2}$。直观上，逻辑斯谛假设不确信标签的值，所以它猜想标签为符号 $(\langle w, x \rangle)$ 的概率稍大于 50%。相比之下，半空间假设总是输出确定的 1 或 -1，即使 $|\langle w, x \rangle|$ 非常接近 0。

下面，我们确定损失函数，即，我们应该定义给定 $y \in \{\pm 1\}$ 时，使用 $h_w(x) \in [0, 1]$ 预测的损失程度。显然，我们希望如果 $y = 1$ 时，h_w 尽可能大；$y = -1$ 时，$1 - h_w$（即预测 -1 的概率）尽可能大。注意，

$$1 - h_w(x) = 1 - \frac{1}{1 + \exp(-\langle w, x \rangle)} = \frac{\exp(-\langle w, x \rangle)}{1 + \exp(-\langle w, x \rangle)} = \frac{1}{1 + \exp(\langle w, x \rangle)}$$

因此，任何合理的损失函数都应随 $\frac{1}{1 + \exp(y\langle w, x \rangle)}$ 单调增，或者等价地，随 $1 + \exp(-y\langle w, x \rangle)$ 单调增。逻辑斯谛中惩罚 h_w 的损失函数基于 $1 + \exp(-y\langle w, x \rangle)$ 的对数（对数是单调函数），即

$$\ell(h_w, (x, y)) = \log(1 + \exp(-y\langle w, x \rangle))$$

因此，给定训练集 $S = (x_1, y_1), \cdots, (x_m, y_m)$，逻辑斯谛回归的 ERM 问题为

$$\operatorname*{argmin}_{w \in \mathcal{X}} \frac{1}{m} \sum_{i=1}^{m} \log(1 + \exp(-y_i \langle w, x_i \rangle)) \tag{9.10}$$

逻辑斯谛损失函数的一个优点是它是关于 w 的凸函数。所以 ERM 问题可以使用标准方法有效求解。我们将在后续章节中研究如何利用凸函数学习，在特殊情况下用简单的算法最小化凸函数。

逻辑斯蒂回归的 ERM 问题(式 9.10)与最大似然估计问题相同，后者是一个在给定数据集和具体的参数化概率函数上寻找联合概率最大化的著名统计学方法。我们将在第 24 章中学习最大似然方法。 98

9.4 小结

线性预测是假设类中最有用的部分，许多广泛应用的学习算法都是基于线性预测。对于线性预测可分情形中的 0—1 损失，不可分情形的平方损失以及逻辑斯蒂损失，我们给出了有效的学习算法。在后面的章节，我们将展示这些损失函数之所以能够有效进行学习的性质。

自然地，当我们先验假设一些线性预测在特定分布上具有低风险时，线性预测是有效的。下一章中我们将看到如何在简单类上使用线性预测构建非线性预测，这将使我们利用线性预测解决多种先验假设。

9.5 文献评注

感知器算法源于 Rosenblatt(1958)，其收敛率的证明来自于 Agmon(1954)和 Novikoff(1962)。最小平方回归源于 Gauss(1795)，Legendre(1805)和 Adrain(1808)。

9.6 练习

9.1 说明如何使用线性回归的绝对值损失函数解决 ERM 问题，$\ell(h,(x,y))=|h(x)-y|$，即证明如何将

$$\min_{w}\sum_{i=1}^{m}|\langle w,x_i\rangle-y_i|$$

写成线性规划。

提示：证明任意的 $c\in\mathbb{R}$，

$$|c|=\min_{a\geq 0}a \quad \text{s.t.} \quad c\leq a \quad \text{且} \quad c\geq -a$$

9.2 证明式(9.6)中的 A 矩阵可逆当且仅当 $x_1,\cdots,x_n$ 张成 $\mathbb{R}^d$。

9.3 证明定理 9.1 在下面情形下的严格性：对任意的正整数 m，有向量 $w^*\in\mathbb{R}^d$(对于一些合适的 d)和序列样本 $(x_1,y_1),\cdots,(x_m,y_m)$ 使下列条件满足：

1) $R=\max_i|x_i|\leq 1$

2) $\|w^*\|^2=m$ 且对于所有的 $i\leq m$，$y_i\langle x_i,w^*\rangle\geq 1$。注意，使用定理 9.1 中的记号，我们可以得到

$$B=\min\{\|w\|:\forall i\in[m],y_i\langle w,x_i\rangle\geq 1\}\leq\sqrt{m}$$

因此 $(BR)^2\leq m$。 99

3) 对序列样本使用感知器方法时，将在 m 步后收敛。

提示：选取 $d=m$，且对于每个 i 选取 $x_i=e_i$。

9.4 给定任意 m，找到序列中有标签样本 $(x_1,y_1),\cdots,(x_m,y_m)\in(\mathbb{R}^2\times\{-1,-1\})^m$ 的样本，使定理 9.1 中的上界为 m 且感知器算法出现 m 个错分。

提示：设定每个 x_i 为一个形如 (a,b,y_i) 的 3 维向量，其中 $a^2+b^2=R^2-1$。令 $w^*=(0,0,1)$。回顾感知器上界的证明(定理 9.1)，找到我们使用不等号($\leq$)而不是等号(=)的地方，思考什么情形下等号成立。

9.5　假设我们改进感知器算法：在更新步骤，当分类错误时我们使用$w^{(t+1)}=w^{(t)}+\eta y_i x_i$（$\eta>0$）而不是$w^{(t+1)}=w^{(t)}+y_i x_i$。证明改进感知器算法的迭代步数与原始感知器一样，并且收敛时向量所指向的方向也相同。

9.6　本题中，我们将考虑$\mathbb{R}^d$球空间类的VC维。

$$\mathcal{B}_d=\{B_{v,r}: v\in\mathbb{R}^d, r>0\}$$

其中

$$B_{v,r}(x)=\begin{cases}1 & 若\quad \|x-v\|\leqslant r\\ 0 & 其他\end{cases}$$

1）考虑映射$\phi:\mathbb{R}^d\to\mathbb{R}^{d+1}$，其中$\phi(x)=(x,\|x\|^2)$。证明如果$x_1,\cdots,x_m$被$\mathcal{B}_d$打散，则$\phi(x_1),\cdots,\phi(x_m)$被$\mathbb{R}^{d+1}$半空间打散（在这个方程中我们假定$\mathrm{sign}(0)=1$）。关于$\mathrm{VCdim}(\mathcal{B}_d)$，这告诉了我们什么？

*2）在$\mathbb{R}^d$中找到被$\mathcal{B}_d$打散的$d+1$个点的集合，证明

[100]

$$d+1\leqslant \mathrm{VCdim}(\mathcal{B}_d)\leqslant d+2$$

boosting

boosting 作为一种源自理论问题的算法范式，已经发展成为一种非常实用的机器学习工具。boosting 算法泛化了线性预测器，并由此处理本书前面提及的两个主要问题。第一个问题是偏差-复杂度权衡。（在第 5 章）我们已经看到，根据经验风险最小化（ERM）原则得到的学习器的误差可以拆分为逼近误差和估计误差二者之和。我们要搜索的学习器的假设类表达能力越强，那么它的逼近误差也就越小，但是估计误差则相应变大。因此，任何一种学习器都会面临如何更好地权衡二者之间关系这样一个问题。boosting 算法使得学习器可以对这二者的权衡有一个平滑的控制。算法首先从一个最基本的假设（可能会有较大的逼近误差）开始，随着算法的进行，预测器所属的假设就变得越来越完善。

boosting 涉及的第二个问题就是算法学习的计算复杂度。正如第 8 章所述，对于一些我们感兴趣的假设类，寻求对应的 ERM 假设可能从计算上来说是不可行的。boosting 算法则可提高弱学习器的精度。直观地说，我们可以认为弱学习器就是根据经验法则，从一组易于学习的假设空间中获取一种假设的算法。对于这类学习器，它们的效果仅需略优于随机猜测。如果弱学习器是易于实现的，boosting 则相当于一种工具，它可以将这些弱学习器聚合得到近似最优的预测器，而这些预测器可以适用于比较大且难于学习的假设。

本章中，我们将描述分析一种实用且有效的 boosting 算法——AdaBoost（Adaptive Boosting）。AdaBoost 算法可以得到一个假设，而这个假设是一些基本假设的线性组合。也就是说，AdaBoost 依赖于假设类族，而这些假设类则是通过一些简单类的线性组合而得。后面我们会说明 AdaBoost 仅仅通过调整一个参数即可控制逼近误差与估计误差的权衡。

AdaBoost 揭示了通过其他函数的组合可以提高线性预测器的表示能力这样一个主题，本书后续会提到这点。10.3 节详细介绍了这一问题。

AdaBoost 源自于能否由高效的弱学习器聚合为高效的强学习器这样一个理论问题。这个问题最早由 Kearns 和 Valiant 于 1988 年提出，随后 Robert Schapire 于 1990 年解决了此问题，而后麻省理工学院的一名研究生也对其进行了研究。然而，当时提出的这种方法并不实用。1995 年，Robert Schapire 和 Yoav Freund 提出 AdaBoost 算法，这是第一个真正实用且易于实现的 boosting 算法。这个简单而严谨的算法很快广为流传，Freund 和 Schapire 的这一工作也荣获了很多的奖项。

进一步说，boosting 是冲击可学习理论实用性的一个很好的例证。尽管 boosting 源自于纯理论问题，但它已经产生了许多广为流传的应用算法。事实上，正如本章前面所提，AdaBoost 已经成功应用于人脸图像检测。

10.1 弱可学习

先回顾一下第 3 章 PAC 可学习的定义：称一个假设类$\mathcal{H}$是 PAC 可学习的，如果存在样本复杂度$m_{\mathcal{H}}:(0,1)^2\to\mathbb{N}$以及具有如下性质的学习算法：对于任意的$\epsilon,\delta\in(0,1)$，任意$\mathcal{X}$上的分布$\mathcal{D}$，以及任意标签函数$f:\mathcal{X}\to\{\pm1\}$，如果可实现假设对于$\mathcal{H}$，$\mathcal{D}$，$f$成立，

那么，当学习算法作用于分布$\mathcal{D}$产生的、由标签函数f标定的$m\geq m_{\mathcal{H}}(\varepsilon,\delta)$个独立同分布的样本时，会返回一个假设$h$使得$L_{(\mathcal{D},f)}(h)\leq\varepsilon$的概率至少为$1-\delta$。

进一步说，可学习理论基本定理(第6章定理6.8)描述了可学习类族并指出任意PAC可学习类均可由ERM算法学得。然而，PAC可学习的定义及可学习理论基本定理均忽略了学习的计算复杂度。事实上，正如第8章所述，在有些情况下，使用ERM准则从计算上来说是十分困难的(尽管可实现)。

然而，或许我们可以通过降低精度来降低计算复杂度。给定分布$\mathcal{D}$以及目标标签函数f，是否存在一种误差略优于随机猜测但可高效计算的学习算法？这就引出了如下定义。

定义 10.1(γ-弱可学习)

- 我们称学习算法A是类$\mathcal{H}$的γ-弱可学习器，如果存在函数$m_{\mathcal{H}}:(0,1)\to\mathbb{N}$使得对任意的$\delta\in(0,1)$，任意$\mathcal{X}$上的分布$\mathcal{D}$，以及任意标签函数$f:\mathcal{X}\to\{\pm1\}$，可实现假设对于$\mathcal{H}$，$\mathcal{D}$，$f$成立，那么，当学习算法作用于分布$\mathcal{D}$产生的、由标签函数$f$标定的$m\geq m_{\mathcal{H}}(\delta)$个独立同分布的样本时，会返回一个假设$h$使得$L_{(\mathcal{D},f)}(h)\leq 1/2-\gamma$的概率至少为$1-\delta$。
- 对于假设类$\mathcal{H}$存在一个γ-弱可学习的学习器，那么就称假设类$\mathcal{H}$是γ-弱可学习的。

在这里我们称PAC可学习为强可学习，这个定义与PAC可学习的定义几乎相同，最主要的一点不同就是：强可学习强调能够找到一个任意精度的分类器的能力(对于任意小的$\varepsilon>0$，误差最大为ε)。然而在弱可学习中，我们仅仅需要得到一个误差最大为$1/2-\gamma$的假设，也就是说，误差仅需优于随机猜测。我们希望的是寻求一个高效的弱学习器比得到一个高效的强学习器容易。

102

可学习理论基本定理(定理6.8)指出，如果假设类$\mathcal{H}$的VC维为d，$\mathcal{H}$PAC可学习的采样复杂度满足$m_{\mathcal{H}}(\varepsilon,\delta)\geq C_1\dfrac{d+\log(1/\delta)}{\varepsilon}$，其中$C_1$为常数。将$\varepsilon=1/2-\gamma$代入，我们很快可以得出如果$d=\infty$，那么$\mathcal{H}$就不是$\gamma$-弱可学习的。这就表明，从统计的角度看(如果我们忽略计算复杂度)，弱可学习也由假设$\mathcal{H}$的VC维刻画，因此它与PAC(强)可学习一样困难。然而，当我们考虑计算复杂度时，弱可学习潜在的优势在于或许存在一种算法满足弱学习器的要求，并且是易于实现的。

可行的方法就是选取一个简单的假设类，记作B，利用ERM准则将B作为弱学习算法。为了实现这一目的，我们需要B满足如下两个条件：

- ERM_B可以高效地实现。
- 对于每个通过$\mathcal{H}$中的假设类标记的样本，任意的ERM_B误差最大为$1/2-\gamma$。

接下来的问题就转变为是否可以将高效的弱学习器集成为高效的强学习器。下一节我们会说明这确实是可行的，但在此之前，先来看一个例子。这个例子表明，用基础假设类B可以得到类$\mathcal{H}$的高效弱可学习器。

例 10.1 (用决策桩得到3段分类器弱可学习性) 设$\mathcal{X}=\mathbb{R}$，$\mathcal{H}$是3段分类器类，即，$\mathcal{H}=\{h_{\theta_1,\theta_2,b}:\theta_1,\theta_2\in\mathbb{R},\theta_1<\theta_2,b\in\{\pm1\}\}$，对于任意$x$

$$h_{\theta_1,\theta_2,b}(x)=\begin{cases}+b & \text{如果 } x<\theta_1 \text{ 或者 } x>\theta_2\\ -b & \text{如果 } \theta_1\leq x\leq\theta_2\end{cases}$$

一个例子假设$(b=1)$描述如下：

设 B 为决策桩类，也就是说，$B=\{x\mapsto \mathrm{sign}(x-\theta)\cdot b:\theta\in\mathbb{R},\ b\in\{\pm1\}\}$。下面我们说明$\mathrm{ERM}_B$对$\mathcal{H}$是$\gamma$-弱可学习的，其中 $\gamma=1/12$。

为了说明这点，我们首先明确，对于与$\mathcal{H}$一致的每一个分布，都有一个决策桩使得 $L_{\mathcal{D}}(h)\leqslant 1/3$。事实上，$\mathcal{H}$中的每个分类器都包含三块区域(两个无界的射线区域和一个中心区域)，每块区域都有可变的标签。对于任意的一对区域，都有一个决策桩与这两部分的标签一致。对于实数域上任意的分布$\mathcal{D}$，对于任一将这条直线划分为三块的划分，三块区域当中必有一块区域对应于$\mathcal{D}$的权重最大为 1/3。设$h\in\mathcal{H}$是一个零误差的假设。决策桩仅仅在这一区域与 h 不一致，误差最大为 1/3。

103

最后，由于决策桩的 VC 维是 2，如果样本大小稍大于 $\Omega(\log(1/\delta)/\varepsilon^2)$，$\mathrm{ERM}_B$规则返回一个误差最大为 $1/3-\varepsilon$ 的假设的概率至少为 $1-\delta$。设 $\varepsilon=1/12$，我们可以得到ERM_B误差最大为 $1/3-1/12=1/2-1/12$。

我们可以看到ERM_B对$\mathcal{H}$是一个γ-弱可学习器。接下来我们将说明如何将 ERM 准则有效地应用到决策桩中。

有效应用 ERM 准则于决策桩

设$\mathcal{X}=\mathbb{R}^d$，考虑$\mathbb{R}^d$上的基本假设类决策桩，也就是说，

$$\mathcal{H}_{\mathrm{DS}}=\{x\mapsto \mathrm{sign}(\theta-x_i)\cdot b:\theta\in\mathbb{R},i\in[d],b\in\{\pm1\}\}$$

简单起见，设 $b=1$；也就是说，我们考虑$\mathcal{H}_{\mathrm{DS}}$中具有 $\mathrm{sign}(\theta-x_i)$形式的所有假设。设 $S=((x_1,y_1),\cdots,(x_m,y_m))$为训练集。接下来我们将说明如何应用 ERM 规则，也就是怎样找到一个决策桩使得$L_S(h)$最小。更进一步，由于在下一节说明 AdaBoost 需要寻求一个与 S 上的分布相关并且风险最小化的假设，这里我们会说明如何最小化这种风险函数。更精确地说，设 $\boldsymbol{D}$ 是$\mathbb{R}^m$里的一个概率向量(也就是说，$\boldsymbol{D}$ 里所有元素值非负并且 $\sum_i D_i=1$)。后面我们描述的弱学习器输入 $\boldsymbol{D}$ 和 S，输出一个决策桩 $h:\mathcal{X}\to\mathcal{Y}$最小化关于$\mathcal{D}$的风险

$$L_{\boldsymbol{D}}(h)=\sum_{i=1}^{m}D_i\mathbb{1}_{[h(x_i)\neq y_i]}$$

注意到如果 $\boldsymbol{D}=(1/m,\cdots,1/m)$，那么$L_{\boldsymbol{D}}(h)=L_S(h)$。

我们知道每个决策桩由索引 $j\in[d]$和阈值 θ 决定。因此，最小化$L_{\boldsymbol{D}}(h)$等价于解决

$$\min_{j\in[d]}\min_{\theta\in\mathbb{R}}\Big(\sum_{i:y_i=1}D_i\mathbb{1}_{[x_{i,j}>\theta]}+\sum_{i:y_i=-1}D_i\mathbb{1}_{[x_{i,j}\leqslant\theta]}\Big)\tag{10.1}$$

固定 $j\in[d]$ 并对样本进行排序，使得 $x_{1,j}\leqslant x_{2,j}\leqslant\cdots\leqslant x_{m,j}$。定义 $\Theta_j=\left\{\frac{x_{i,j}+x_{i+1,j}}{2}:i\in[m-1]\right\}\cup\{(x_{1,j}-1),(x_{m,j}+1)\}$。对于任意的 $\theta\in\mathbb{R}$，必存在$\theta'\in\Theta_j$，对于样本 S 有相同的预测结果。因此，我们可以在 $\theta\in\Theta_j$上而不是$\theta\in\mathbb{R}$上最小化目标函数。

这已经给我们提供了一种函数的算法：选择 $j\in[d]$和 $\theta\in\Theta_j$使得公式(10.1)的目标函数值最小。对于每一个 j 和$\theta\in\Theta_j$我们必须计算 m 个样本的总和；因此，这种方法的运行时间是 $O(dm^2)$。接下来我们会介绍一个简单的技巧使得最小化目标函数的运行时间为$O(dm)$。

算法流程如下。假定我们已经计算得到对于 $\theta\in(x_{i-1,j},x_{i,j})$的目标函数值，并假定

$F(\theta)$为此目标函数值。当我们考虑$\theta' \in (x_{i,j}, x_{i+1,j})$时有

[104]

$$F(\theta') = F(\theta) - D_i \mathbb{1}_{y_i=-1} + D_i \mathbb{1}_{y_i=1} = F(\theta) - y_i D_i$$

因此，给定在先前的阈值θ处的目标函数值，我们可以在一个常数时间内计算目标函数在θ'处的值，也就是说，经过对样本的每个坐标进行排序这样一个预处理，最小化问题就可以在$O(dm)$时间内解决。伪代码如下：

决策桩经验风险最小化

输入：

训练集 $S=((x_1, y_1), \cdots, (x_m, y_m))$

分布向量 $\boldsymbol{D}$

目标： 寻找j^*，θ^*满足等式(10.1)

初始化： $F^* = \infty$

for $j=1, \cdots, d$

根据第j维坐标对S排序，并记

$x_{1,j} \leq x_{2,j} \leq \cdots \leq x_{i,j} \leq x_{i+1,j} \leq \cdots \leq x_{m,j}$，$x_{m+1,j} = x_{m,j}+1$

$F = \sum_{i:y_i=1} D_i$

if $F < F^*$

$F^* = F$，$\theta^* = x_{1,j}-1$，$j^* = j$

for $i=1, \cdots, m$

$F = F - y_i D_i$

if $F < F^*$ 并且 $x_{i,j} \neq x_{i+1,j}$

$F^* = F$，$\theta^* = \frac{1}{2}(x_{i,j} + x_{i+1,j})$，$j^* = j$

输出： j^*，θ^*

10.2 AdaBoost

AdaBoost 是一种可以获得弱学习器并寻求经验风险最小的算法。AdaBoost 算法的输入为样本训练集 $S=((x_1, y_1), \cdots, (x_m, y_m))$，对于每一个$i$，$y_i = f(x_i)$对应于标签函数$f$。boosting 算法就是一个连续迭代的过程。在第t次迭代中，booster 首先定义样本集S上的分布，以$\boldsymbol{D}^{(t)}$表示。也就是说$\boldsymbol{D}^{(t)} \in \mathbb{R}_+^m$并且$\sum_{i=1}^m D_i^{(t)} = 1$。然后，booster 将分布$\boldsymbol{D}^{(t)}$和样本集$S$传递给弱学习器。(在这种方式下，弱学习器可以根据$\boldsymbol{D}^{(t)}$和$f$构建独立同分布的样本。)弱学习器将会返回一个"弱"的假设h_t，其误差

$$\varepsilon_t \stackrel{\text{def}}{=} L_{\boldsymbol{D}^{(t)}}(h_t) \stackrel{\text{def}}{=} \sum_{i=1}^m D_i^{(t)} \mathbb{1}_{[h_t(x_i) \neq y_i]}$$

最大为$\frac{1}{2}-\gamma$(当然，弱学习器也会有不超过δ的概率是失败的)。然后，AdaBoost 分配给h_t一个权重$w_t = \frac{1}{2}\log\left(\frac{1}{\varepsilon_t}-1\right)$。也就是说，$h_t$的权重与$h_t$的误差成反比。在迭代过程的最后，AdaBoost 更新样本分布，使得h_t分错的样本概率更大而分正确的样本概率更小。直

观地说，这会强制弱学习器在下一次迭代中更加关注上一次分错的样本。AdaBoost 算法的输出是一个基于所有弱假设空间加权和的“强”分类器。AdaBoost 伪代码如下： 105

AdaBoost

输入：

训练集 $S=((x_1, y_1), \cdots, (x_m, y_m))$

弱学习器 WL

迭代次数 T

初始化： $\boldsymbol{D}^{(1)}=\left(\frac{1}{m}, \cdots, \frac{1}{m}\right)$

for $t=1, \cdots, T$

调用弱学习器 $h_t=\mathrm{WL}(\boldsymbol{D}^{(t)}, S)$

计算 $\varepsilon_t=\sum_{i=1}^{m} D_i^{(t)} \mathbb{1}_{[y_i \neq h_t(x_i)]}$

设 $w_t=\frac{1}{2}\log\left(\frac{1}{\varepsilon_t}-1\right)$

更新 $D_i^{(t+1)}=\dfrac{D_i^{(t)} \exp(-w_t y_i h_t(x_i))}{\sum_{j=1}^{m} D_j^{(t)} \exp(-w_t y_j h_t(x_j))}$，对任意的 $i=1, \cdots, m$

输出： 假设 $h_S(x)=\operatorname{sign}\left(\sum_{t=1}^{T} w_t h_t(x)\right)$

接下来的定理表明输出假设的训练误差随着 boosting 迭代次数的增加呈指数下降。

定理 10.2 假定 S 为训练集，并且 AdaBoost 每次迭代之后的弱学习器都会返回一个假设使得 $\varepsilon_t \leqslant \frac{1}{2}-\gamma$。AdaBoost 给出假设的训练误差最大为

$$L_S(h_S)=\frac{1}{m}\sum_{i=1}^{m} \mathbb{1}_{[h_S(x_i) \neq y_i]} \leqslant \exp(-2\gamma^2 T)$$

证明 对于任意 t，记 $f_t=\sum_{k=1}^{t} w_k h_k$。因此，AdaBoost 的输出为 f_T。另外，记

$$Z_t=\frac{1}{m}\sum_{i=1}^{m} \mathrm{e}^{-y_i f_t(x_i)}$$

对于任意的假设，我们有 $\mathbb{1}_{[h(x) \neq y]} \leqslant \mathrm{e}^{-y h(x)}$。因此，$L_S(f_T) \leqslant Z_T$，因此只需证明 $Z_T \leqslant \mathrm{e}^{-2\gamma^2 T}$。为了得到 Z_T 的上界，我们将其重写为

$$Z_T=\frac{Z_T}{Z_0}=\frac{Z_T}{Z_{T-1}} \cdot \frac{Z_{T-1}}{Z_{T-2}} \cdots \frac{Z_2}{Z_1} \cdot \frac{Z_1}{Z_0} \tag{10.2}$$

这里我们利用了 $Z_0=1$，因为 $f_0=0$。因此，只需证明对于每一次迭代 t，

$$\frac{Z_{t+1}}{Z_t} \leqslant \mathrm{e}^{-2\gamma^2} \tag{10.3}$$

为证明上式，我们首先说明，利用一个简单的归纳证明，对于所有的 t 和 i，

$$D_i^{(t+1)}=\frac{\mathrm{e}^{-y_i f_t(x_i)}}{\sum_{j=1}^{m} \mathrm{e}^{-y_j f_t(x_j)}}$$

106

因此，

$$
\begin{aligned}
\frac{Z_{t+1}}{Z_t} &= \frac{\sum_{i=1}^{m} e^{-y_i f_{t+1}(x_i)}}{\sum_{i=1}^{m} e^{-y_i f_t(x_i)}} = \frac{\sum_{i=1}^{m} e^{-y_i f_t(x_i)} e^{-y_i h_{t+1}(x_i)}}{\sum_{i=1}^{m} e^{-y_i f_t(x_i)}} = \sum_{i=1}^{m} D_i^{(t)} e^{-y_i h_{t+1}(x_i)} \\
&= e^{-\eta_{t+1}} \sum_{i:y_i h_{t+1}(x_i)=1} D_i^{(t+1)} + e^{\eta_{t+1}} \sum_{i:y_i h_{t+1}(x_i)=-1} D_i^{(t+1)} \\
&= e^{-\eta_{t+1}}(1-\varepsilon_{t+1}) + e^{\eta_{t+1}} \varepsilon_{t+1} \\
&= \frac{1}{\sqrt{\frac{1}{\varepsilon_{t+1}}-1}}(1-\varepsilon_{t+1}) + \sqrt{\frac{1}{\varepsilon_{t+1}}-1}\,\varepsilon_{t+1} \\
&= \sqrt{\frac{\varepsilon_{t+1}}{1-\varepsilon_{t+1}}}(1-\varepsilon_{t+1}) + \sqrt{\frac{1-\varepsilon_{t+1}}{\varepsilon_{t+1}}}\,\varepsilon_{t+1} = 2\sqrt{\varepsilon_{t+1}(1-\varepsilon_{t+1})}
\end{aligned}
$$

根据假设，$\varepsilon_{t+1} \leq \frac{1}{2}-\gamma$，并且函数 $g(a)=a(1-a)$ 在 $[0, 1/2]$ 上是单调递增的，我们可以得到

$$
2\sqrt{\varepsilon_{t+1}(1-\varepsilon_{t+1})} \leq 2\sqrt{\left(\frac{1}{2}-\gamma\right)\left(\frac{1}{2}+\gamma\right)} = \sqrt{1-4\gamma^2}
$$

最后，根据不等式 $1-a \leq e^{-a}$，我们可以得到 $\sqrt{1-4\gamma^2} \leq e^{-\frac{4\gamma^2}{2}} = e^{-2\gamma^2}$。这就证明了等式(10.3)成立，也就证明了我们的结论。 ■

AdaBoost 每次迭代有 $O(m)$ 步操作以及一个调用弱学习器操作。因此，如果弱学习器可以高效地应用(正如决策桩利用 ERM 准则)，那么总的训练过程将会是高效的。

评注 定理 10.2 假定 AdaBoost 每次迭代弱学习器都会返回一个假设，其加权样本误差最大为 $\frac{1}{2}-\gamma$。根据弱学习器的定义，也会有 δ 的概率失败。根据一致界定理，弱学习器在所有的迭代过程中不失败的概率至少为 $1-\delta T$。练习 10.1 中可以看到，采样复杂度与失败概率 δ 总是对数关系的，因此，对于弱学习来说引入一个非常小的 δ 并不困难。因此我们可以假设 δT 也是很小的。进一步说，因为弱学习器只是应用在训练集上的分布，很多情况下我们可以实现弱学习器使之失败的概率为零(即 $\delta=0$)。一个例子就是弱学习器采用决策桩寻求 $L_D(h)$ 最小的情况，前面部分已经对其做了描述。

107

定理 10.2 告诉我们 AdaBoost 构建的假设的经验风险随着 T 的增加而趋近于零。然而，我们真正关心的是输出的真实误差。为了说明真实误差，我们首先明确 AdaBoost 的输出事实上是半空间的组合，而这些半空间是由弱学习器构建的 T 个弱假设。下一节我们会说明如果弱假设来自一组低 VC 维的假设类，那么 AdaBoost 的估计误差就很小；也就是说，AdaBoost 输出的真实风险与经验风险差别不会太大。

10.3 基础假设类的线性组合

正如前面提及的，主流的算法构建弱学习器时在某一个基础假设类应用经验风险最小准则(例如，在决策桩上利用 ERM 准则)。我们也知道 AdaBoost 的输出事实上是半空间的组合。因此，给定一个基础假设类 B(例如决策桩)，AdaBoost 的输出将会是下列当中的一个：

$$L(B,T)=\{x\mapsto \operatorname{sign}\Big(\sum_{t=1}^{T} w_t h_t(x)\Big): w\in \mathbb{R}^T, \forall t, h_t\in B\} \tag{10.4}$$

也就是说，每个 $h\in L(B,T)$ 都以 B 里的 T 个基础假设和一个向量 $w\in\mathbb{R}^T$ 为参数。这样的一个 h 作用于实例 x 上的输出可以通过如下得到，首先利用 T 个基础假设构建向量 $\psi(x)=(h_1(x),\cdots,h_T(x))\in\mathbb{R}^T$，然后，将 w 定义的半空间作用于 $\psi(x)$。

本节我们分析 VC 维固定的情况下 $L(B,T)$ 的估计误差，而 $L(B,T)$ 的 VC 维与 B 的 VC 维及 T 有关。接下来我们会看到，最大为对数，$L(B,T)$ 的 VC 维以 T 倍的 B 的 VC 维为界。也就是 AdaBoost 的估计误差随着 T 线性增加。另一方面，AdaBoost 的经验风险随 T 递减。事实上，我们后面将会说明，T 可以用来降低 $L(B,T)$ 的逼近误差。因此，AdaBoost 的参数 T 使得我们可以控制偏差-复杂度的权衡。

为了说明 $L(B,T)$ 的表示能力是如何随着 T 而增加的，考虑一个简单的例子，$\mathcal{X}=\mathbb{R}$，基础类为决策桩，

$$\mathcal{H}_{\mathrm{DS1}}=\{x\mapsto \operatorname{sign}(x-\theta)\cdot b: \theta\in\mathbb{R}, b\in\{\pm 1\}\}$$

在这个一维的例子中，$\mathcal{H}_{\mathrm{DS1}}$ 事实上等价于 $\mathbb{R}$ 上的(非齐次)半空间。

假设 $\mathcal{H}$ 为更加复杂的(相对于直线上的半空间来说)分段常值函数类。设 g 是一个最大为 r 段的分段常值函数，也就是说，存在阈值 $-\infty=\theta_0<\theta_1<\theta_2<\cdots<\theta_r=\infty$ 使得

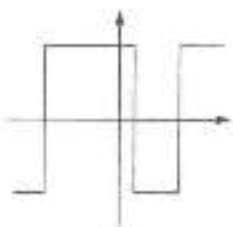
$$g_r(x)=\sum_{i=1}^{r} a_i \mathbb{1}_{[x\in(\theta_{i-1},\theta_i]}, \quad \forall i, a_i\in\{\pm 1\}$$

108

定义 $\mathcal{G}_r$ 为最多为 r 段的所有分段常值分类器的类。

接下来我们说明 $\mathcal{G}_T\subseteq L(\mathcal{H}_{\mathrm{DS1}},T)$；也就是说，$T$ 个决策桩的半空间类等价于最大 T 段的分段常值分类器。

事实上，不失一般性，考虑任意的 $g\in\mathcal{G}_T$ 并且 $a_1=(-1)^t$。这就表明，如果 x 在区间 $(\theta_{t-1},\theta_t]$ 里，那么 $g(x)=(-1)^t$。例如：



那么，函数

$$h(x)=\operatorname{sign}\Big(\sum_{t=1}^{T} w_t \operatorname{sign}(x-\theta_{t-1})\Big) \tag{10.5}$$

其中 $w_1=0.5$，并且对 $t>1$，$w_t=(-1)^t$ 属于 $L(\mathcal{H}_{\mathrm{DS1}},T)$ 并且等于 g(参见练习 10.2)。

从这个例子我们可以看出，$L(\mathcal{H}_{\mathrm{DS1}},T)$ 可以打散实数域 $\mathbb{R}$ 上任意的 $T+1$ 个实例组成的集合；也就是说，$L(\mathcal{H}_{\mathrm{DS1}},T)$ 的 VC 维至少为 $T+1$。因此，T 是控制偏差-复杂度权衡的一个参数：增大 T 可以得到一个表达能力更强的假设类，但另一方面，又可能增加估计误差。在下一小节，我们会正式给出对于任意基类 B，$L(B,T)$ 的 VC 维上界。

$L(B,T)$ 的 VC 维

下面的定理表明 $L(B,T)$ 的 VC 维以 $\tilde{O}(\mathrm{VCdim}(B)T)$ 为上界($\tilde{O}$ 符号忽略了常数及对数因子)。

引理 10.3　设 B 为基类，$L(B, T)$ 定义如等式(10.4)，假定 T 和 $\mathrm{VCdim}(B)$ 均至少为 3，那么，

$$\mathrm{VCdim}(L(B,T)) \leq T(\mathrm{VCdim}(B)+1)(3\log(T(\mathrm{VCdim}(B)+1))+2)$$

证明　设 $d=\mathrm{VCdim}(B)$，$C=\{x_1, \ldots, x_m\}$ 可由 $L(B, T)$ 打散，由 $h \in L(B, T)$ 确定 C 的标签。首先选择 $h_1, \ldots, h_T \in B$，然后对向量 $(h_1(x), \ldots, h_T(x))$ 应用半空间假设。由 Sauer 引理，B 在 C 上最多有 $(em/d)^d$ 种不同的二分法(标签)。因此，我们要从 $(em/d)^d$ 个不同的假设中选出 T 个，而这样的选择最多有 $(em/d)^{dT}$ 种。接下来，对于每一个选择，我们应用一个线性预测器，也就得到了最多 $(em/T)^T$ 种二分法。因此，我们所能构建的二分法的总数最多为

[129]

$$(em/d)^{dT}(em/T)^T \leq m^{(d+1)T}$$

这里我们应用了 d 和 T 均至少为 3 的假设。由于假定 C 可被打散，我们必须使前半部分不小于 2^m，因此

$$2^m \leq m^{(d+1)T}$$

因此，

$$m \leq \log(m)\frac{(d+1)T}{\log(2)}$$

附录 A 中的引理 A.1 表明使前半部分成立的必要条件是

$$m \leq 2\frac{(d+1)T}{\log(2)}\log\frac{(d+1)T}{\log(2)} \leq (d-1)T(3\log((d+1)T)+2)$$

以上定理得证。 ■

在练习 10.4 中我们会看到，对于一些基类 B，$\mathrm{VCdim}(L(B, T)) \geq \Omega(\mathrm{VCdim}(B)T)$ 也成立。

10.4　AdaBoost 用于人脸识别

现在我们转向由 Viola 和 Jones 提出的用于人脸识别的一个基础假设。在这个任务里，实例空间是图像，图像由像素灰度值矩阵表示。为简单起见，假设图片大小为 24×24(像素)，也就是说我们的实例空间就是大小为 24×24 的实值矩阵的集合。我们的目的是学习一个分类器 $h: \mathcal{X} \to \{\pm 1\}$，使得输入一幅给定的图像，输出结果为图片中是否包含人脸。

基类的每个假设都具有 $h(x)=f(g(x))$ 的形式，其中 f 是决策桩，$g: \mathbb{R}^{24\times 24} \to \mathbb{R}$ 将一幅图像映射为一个实数。每个函数 g 由以下信息参数化：

- 轴对齐的矩形 R。由于每幅图像大小为 24×24，最多有 24^4 个轴对齐的矩形。
- 一种类型，$t \in \{A, B, C, D\}$。每种类型对应一个掩模，见图 10.1。

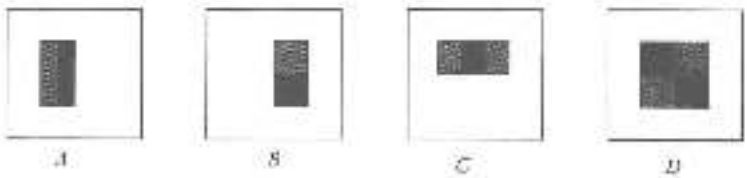


图 10.1　基于人脸识别的四种函数类型 g。类型 A 和 B 的 g 值是两个矩形区域各自像素值和的差，这些区域大小形状相同，水平或垂直相连。对于类型 C，g 的值为中间矩形像素值和减去两侧矩形区域像素值之和。对于类型 D，我们计算对角线成对矩形的差值

为了计算 g，我们将掩模 t 拉伸以适用矩形 R，然后将内矩形像素之和减去在外矩形

内的像素之和(也就是灰度值的总和)。

由于这样的函数 g 最多有 $24^2\times4$ 个，所以我们在对基假设类应用弱学习器的时候，就可以通过首先计算 g 对于每一幅图像的所有可能输出，然后再应用前面描述的决策桩弱学习器。通过计算训练集里每幅图像的积分这样一组预处理，使得第一步可以高效地完成，详见练习 10.5。

图 10.2 描述了当运行由 Viola 和 Jones 提出的基特征时，通过 AdaBoost 选出的前两个特征。 [110]

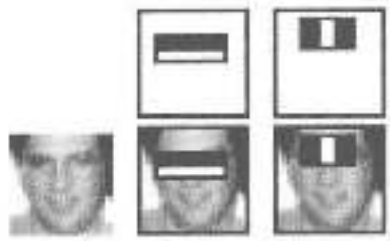


图 10.2 Viola 和 Jones 应用的由 AdaBoost 选出的第一个和第二个特征。第一行是两种特征，第二行是特征覆盖在一幅经典人脸图像的效果。第一个特征描述眼睛区域与整个上部脸颊区域光强的差值，而通常眼睛区域要比脸颊区域暗。第二个特征比较眼睛区域及穿过鼻梁区域的光强

10.5 小结

boosting 是放大弱学习器精度的一种方法。本章我们描述了 AdaBoost 算法，指出了经过 T 次迭代，AdaBoost 会返回类 $L(B,T)$ 的一个假设，而这是通过基类 B 的 T 个假设的线性组合得到的。我们也说明了参数 T 如何控制逼近误差与估计误差的权衡。下一章，我们会研究如何在数据集上调整参数(如 T)。

10.6 文献评注

正如前面提及的，boosting 源自一组高效的弱学习器是否可提升为一个高效的强学习器这样的一个理论问题(Kearns & Valiant 1998)，并由 Schapire 解决(1990)。AdaBoost 算法由 Freund 和 Schapire 提出(1995)。 [111]

boosting 可以从很多方面来描述。纯粹从理论上讲，AdaBoost 可被解释为一种反面的效果：如果假设类的强学习计算困难，那么它的弱学习也同样如此。这可以很好地说明，只要 $\mathcal{H}$ 可通过 B 弱学习，如果某些假设 $\mathcal{H}$ 是 PAC 学习难以实现的，那么类 B 的不可知 PAC 可学习也是难以实现的。例如，Klivans 和 Sherstov(2006)指出半空间交叉类的 PAC 学习是困难的(即使在可实现的情况下)。这个结果可用于说明，单个半空间的不一致 PAC 可学习是计算困难的(Shalev-Shwartz, Shamir & Sridharan 2010)。其目的是为了说明由一个半空间的不可知 PAC 学习器可以得到交叉半空间的弱学习器，由于弱学习器可以提升，我们就可以得到交叉半空间的一个强学习器。

AdaBoost 也证明了弱学习的存在性与在基假设类上用线性分类器的数据可分性二者等价。这与博弈论中的基本定理 von Neumann 极小极大定理(von Neumann 1928)是非常相关的。

AdaBoost 也与我们第 15 章中讲述的 margin 相关；它也可看做我们第 25 章中讲述的前向贪心选择算法。Schapire 和 Freund 最新的书(2012)涵盖了 boosting 的所有方面，这

使得我们更易于接触到此领域的宝贵财富。

10.7　练习

10.1　**boosting 置信度**：设算法 A 确保存在常数$\delta_0\in(0,1)$及函数$m_{\mathcal{H}}:(0,1)\to\mathbb{N}$，使得对任意的 $\varepsilon\in(0,1)$，如果$m\geq m_{\mathcal{H}}$，那么对于任意的分布$\mathcal{D}$满足$L_{\mathcal{D}}(A(S))\leq\min_{h\in\mathcal{H}}L_{\mathcal{D}}(h)+\varepsilon$的概率至少为$1-\delta_0$。

证明依赖于算法 A 及假设$\mathcal{H}$的程序满足一般的不可知 PAC 可学习模型，并且采样复杂度

$$m_{\mathcal{H}}(\varepsilon,\delta)\leq k\,m_{\mathcal{H}}(\varepsilon)+\left\lceil\frac{2\log(4k/\delta)}{\varepsilon^2}\right\rceil$$

其中，

$$k=\lceil\log(\delta)/\log(\delta_0)\rceil$$

提示：将数据分为$k+1$组，其中前k组样本大小为$m_{\mathcal{H}}(\varepsilon)$，并用算法 A 学习前k组。说明对于所有的组，均有$L_{\mathcal{D}}(A(S))\leq\min_{h\in\mathcal{H}}L_{\mathcal{D}}(h)+\varepsilon$的概率最大为$\delta_0\leq\delta/2$。最后，利用最后一组从算法 A 获得的k个假设里面进行选择(依赖于推论 4.6)。

10.2　证明等式(10.3)给出的函数 h 等价于利用h_t的阈值定义的分段常值函数。

10.3　我们用并不正式的方式说明了 AdaBoost 算法利用权重机制，"迫使"弱学习器在下一次迭代中聚焦于出问题的样本。本题我们要对这个说明进行严格的证明。证明h_t对于分布$\mathcal{D}^{(t+1)}$的误差为 1/2，也就是说，证明对于任意的$t\in[T]$

[112]

$$\sum_{i=1}^{m}D_i^{(t+1)}\mathbf{1}_{[y_i\neq h_t(x_i)]}=1/2$$

10.4　本题讨论$L(B,T)$的 VC 维。我们已经证明了一个上界$(O(dT\log(dT))$，其中$d=\mathrm{VCdim}(B)$。这里我们希望证明一个几乎匹配的更低的界。然而，这并不是对所有的类 B 都成立。

1) 我们知道对任意的类 B 及任意迭代次数$T\geq1$，$\mathrm{VCdim}(B)\leq\mathrm{VCdim}(L(B,T))$。寻找一个类 B 使得对于任意$T\geq1$有$\mathrm{VCdim}(B)=\mathrm{VCdim}(L(B,T))$。

提示：$\mathcal{X}$是有限集。

2) 假设B_d是$\mathbb{R}^d$上的决策桩类，证明$\log(d)\leq\mathrm{VCdim}(B_d)\leq5+2\log(d)$。

提示：对于上界，参考练习 10.11。对于下界，假定$d=2^k$，设 A 是$k\times d$的矩阵，它的列为$\{\pm1\}^k$内所有长度为d的二值向量，A 的行是$\mathbb{R}^d$中的k个向量的集合。证明这个集合可以由$\mathbb{R}^d$上的决策桩打散。

3) 设$T\geq1$是任意整数，证明$\mathrm{VCdim}(L(B_d,T))\geq0.5T\log(d)$。

提示：根据上一问题的矩阵 A 的行构建$\frac{Tk}{2}$个实例集合，并且这些矩阵的行为$2A,3A,4A,\ldots,\frac{T}{2}A$。证明这个集合可以被$L(B_d,T)$打散。

10.5　**用积分图像快速计算 Viola 和 Jones 提出的特征**：设 A 是表示一幅图像的24×24的矩阵，记 A 的积分图$I(A)$为 B 使得$B_{i,j}=\sum_{(i'\leq i,j'\leq j)}A_{i',j'}$。

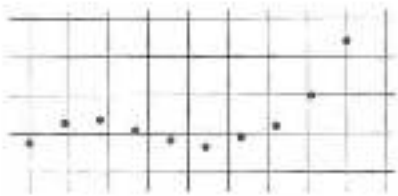
- 证明$I(A)$可由 A 在线性于 A 大小的时间内计算而得。
- 证明何种情况下，Viola 和 Jones 特征可由$I(A)$在常数时间内计算而得(也就是说，运行时间不依赖于定义特征的矩形的大小)。

[113]

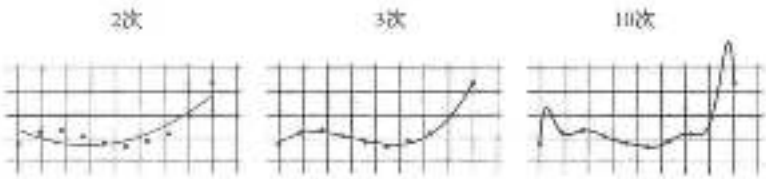
模型选择与验证

在之前的章节中，我们描述了 AdaBoost 算法，并且揭示了 AdaBoost 算法中参数 T 如何控制偏差-复杂度权衡。但是实际问题中我们如何设置参数 T？一般情况下，当面对实际问题，我们通常可以想出几种可能取得好结果的算法，每一种算法可能有几个参数。我们如何为解决发生在身边的问题选择一种最佳算法？如何设置算法参数？这就是通称的模型选择问题。

为了说明模型选择任务，考虑一维回归函数的训练问题，$h: \mathbb{R} \to \mathbb{R}$，假定我们获得如图所示的训练样本集。



我们可以用多项式来拟合这些数据，如第 3 章描述的那样。然而，我们不确定多项式次数 d 为多少会得到最好的结果。多项式次数太低不能很好地拟合数据（比如大的拟合误差），次数太高则可能会出现过拟合（比如大的估计误差）。接下来我们描述分别用 2 次，3 次，10 次多项式来拟合同样的数据集所取得的结果。不难看出，经验风险随着多项式次数增加而减少。然而，图可以直观地告诉我们设置多项式次数为 3 要比设置多项式次数为 10 更好。也就是说仅用经验风险来进行模型选择是不够的。 [114]



在本章中我们介绍两种模型选择的方法。第一种方法建立在 7.2 节所描述的结构风险最小化原则之上，结构风险最小化在学习算法依赖于某一个参数控制偏差-复杂度权衡考虑时非常有用（比如前面例子中拟合多项式的次数或者 AdaBoost 算法中的参数 T）。第二种方法建立在验证的概念之上，基本想法就是将训练集拆分成两个集合，一个用于训练候选的模型，另一个用于确定那一个模型会取得最好的结果。

在模型选择任务中，我们尽力寻找逼近误差和估计误差的平衡点。通常，如果我们的学习算法不能找到一个风险很小的预测器，弄清误差是由过拟合还是欠拟合造成的是很重要的。在 11.3 节中我们将讨论如何做到这一点。

11.1 用结构风险最小化进行模型选择

7.2 节中描述并分析了结构风险最小化原理。这里我们只讨论在事先没有设定特定假

设时，如何应用结构风险最小化原理来调整偏差和复杂度的权衡。取一个可计算的假设类序列$\mathcal{H}_1$，$\mathcal{H}_2$，$\mathcal{H}_3$…例如，已遇到的多项式回归问题，我们用$\mathcal{H}_d$表示次数至多为d的多项式构成的集合。另一个例子，如先前章节描述的那样，用$\mathcal{H}_d$表示AdaBoost所使用的类$L(B, d)$。

我们假定对于任意d，类$\mathcal{H}_d$满足一致收敛属性(见第4章4.3定义)，样本复杂度函数具有以下形式：

$$m_{\mathcal{H}_d}^{UC}(\varepsilon,\delta) \leq \frac{g(d)\log(1/\delta)}{\varepsilon^2} \tag{11.1}$$

这里$g:\mathbb{N}\to\mathbb{R}$是单调递增函数。例如，对于二分类问题，我们可以用$g(d)$乘上一个全局常数(这个常数出现在学习的基本理论里，详见定理6.8)表示$\mathcal{H}_d$类的VC维。对于AdaBoost所使用的类$L(B, d)$，函数g只是简单地随着d增加。

回想结构风险最小化规则遵循"最小化界"方法，在这个例子中，对于$d\in\mathbb{N}$和$h\in\mathcal{H}_d$，最小误差界为下式成立的概率不低于$1-\delta$

115

$$L_{\mathcal{D}}(h) \leq L_S(h) + \sqrt{\frac{g(d)(\log(1/\delta) + 2\log(d) + \log(\pi^2/6))}{m}} \tag{11.2}$$

这个界直接来源于定理7.4，它揭示了：对于任意d和$h\in\mathcal{H}_d$，真实风险界取决于以下两项：经验风险$L_S(h)$和依赖于d的复杂度表达形式。结构风险最小化规则搜索d和$h\in\mathcal{H}_d$，来最小化方程(11.2)。

回到前面描述的多项式回归例子，尽管10次多项式的经验风险小于3次多项式的经验风险，我们仍然偏好次数为3的多项式，因为3次多项式的复杂度比10次多项式复杂度低(复杂度由函数$g(d)$的值反映)。

结构风险最小化在多数情形下都非常有用，但是在很多实际情况下方程(11.2)给出的上界过于悲观。在下一小节中我们提出一个更实用的方法。

11.2 验证法

通常我们希望能更好地估计学习算法所对应输出预测器的真实风险。到目前为止，我们根据一个假设类的估计误差独立界，证明对于一个类中所有假设，真实风险偏离经验风险不远。尽管，这些界是松弛的、悲观的，但是它可以反映所有假设和所有可能的数据分布。通过使用一部分训练数据作为验证集，我们能够得到真实风险的更精确估计，在验证集上可以估计算法输出预测器的有效性，这个过程就称为验证法。

自然地，真实风险的一个更好的估计对于模型选择是非常有用的，我们将在11.2.2节中描述。

11.2.1 留出的样本集

估计预测器h的真实误差最简单的方式就是对附加的样本集采样，独立于训练集，使用验证集上的经验错误作为估计器。形式上，使用$V=(x_1, y_1), \ldots, (x_{m_v}, y_{m_v})$表示新的$m_v$样本的集合，这些样本从分布$\mathcal{D}$上采样得到(训练集$S$的$m$个样本独立)。使用引理4.5的Hoeffding不等式可得：

定理11.1 令h表示预测器，假定损失函数在$[0, 1]$上取值，则对于任一个$\delta\in(0, 1)$，选择一个样本容量为m_v的验证集V的概率不低于$1-\delta$，可得

$$|L_V(h)-L_{\mathcal{D}}(h)|\leqslant\sqrt{\frac{\log(2/\delta)}{2m_v}}$$

定理 11.1 的界不依赖于算法或用于构建 h 的训练误差集，并且比我们目前为止的通常界更紧。界更紧的原因是在新的验证集的估计方面，新的验证集独立于 h 产生的方式。为了说明这一点，假定通过在有 m 个样本的训练集上应用 VC 维 d 的假设类的经验风险最小化预测器得到 h。然后，从定理 6.8 描述的学习基本理论，我们可以获得下面的界 [116]

$$L_{\mathcal{D}}(h)\leqslant L_S(h)+\sqrt{C\frac{d+\log(1/\delta)}{m}}$$

这里 C 是定理 6.8 中出现的常数。相比较而言，从定理 11.1，我们可以得到下面的界

$$L_{\mathcal{D}}(h)\leqslant L_V(h)+\sqrt{C\frac{\log(2/\delta)}{2m_v}}$$

因此，m_v 是 m 的顺序，我们可以通过依赖于 VC 维的因子得到更精确的估计器。而代价是，它要求在训练学习器所用样本之上生成一个附加的样本。

对训练集、独立验证集进行采样，这等同于随机将我们的样本集合分为两部分，一部分用于训练，另一部分用于验证。因此，验证集通常称为留出的样本集。

11.2.2　模型选择的验证法

验证法可以自然地用于模型选择。首先，我们在训练集上训练不同的算法（或者同一个算法，不同的参数），令 $\mathcal{H}=\{h_1,\cdots,h_r\}$ 表示所有不同算法输出预测器的集合。例如，训练多项式回归器的例子，我们用 h_r 表示 r 次多形式回归的输出。从 $\mathcal{H}$ 中选择一个预测器，采样一个独立于训练集的验证集，最终选择一个在验证集上误差最小的预测器。换句话说，我们在验证集上应用经验风险最小化。

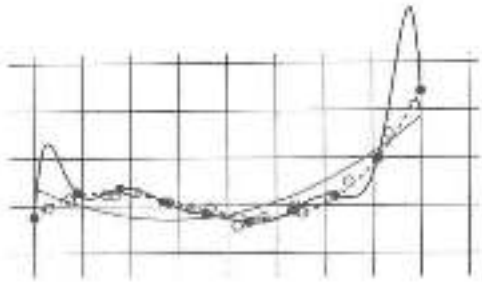
这个过程与学习一个有限假设类非常相似。唯一的不同就是 $\mathcal{H}$ 不是事先固定的，且更依赖于训练集。尽管，由于验证集独立于训练集，我们得到验证集也独立于 $\mathcal{H}$，因此同样的技术（我们用于设计有限假设类的界）也成立。特别地，结合定理 11.1，我们得到一个联合的界：

定理 11.2　令 $\mathcal{H}=\{h_1,\cdots,h_r\}$ 表示一个预测器的特定集合，假定损失函数在 $[0,1]$。假定一个样本数量为 m_v 的验证集 V 与 $\mathcal{H}$ 采样独立。那么，选择 V 的概率不低于 $1-\delta$，我们得到

$$\forall h\in\mathcal{H},|L_{\mathcal{D}}(h)-L_V(h)|\leqslant+\sqrt{\frac{\log(2|\mathcal{H}|/\delta)}{2m_v}}$$

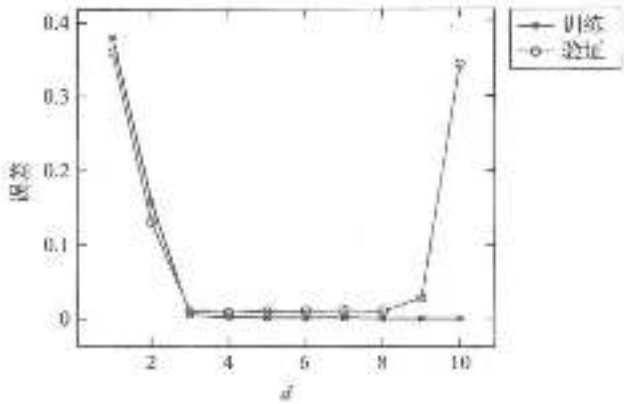
这个理论告诉我们，只要 $\mathcal{H}$ 不太大，验证集的错误就会近似真实误差。但是，如果我们尝试更多的方法（结果是 $|\mathcal{H}|$ 与验证集的样本数量强相关），就会有过拟合的风险。 [117]

为了说明验证如何对模型选择起作用，重新思考本章开头描述的一维多项式样本拟合。接下来描述同样的训练集，次数为 2，3，10 的经验风险，但是这次我们也描述一个附加的验证集（用空心圆标记）。10 次多项式有最小的训练误差，然而 3 次多项式有最小的验证误差，因此 3 次多项式被选为最佳模型。



11.2.3 模型选择曲线

模型选择曲线显示训练误差和验证误差作为一个模型考虑的复杂度函数。例如，对于先前提出的多项式拟合，曲线看起来如下：



从上图可以看出，随着多项式次数的增加，训练误差单调下降(此例中表示样本复杂度)。另一方面，验证错误先下降后上升，这表示模型开始被过拟合损害。

画出曲线图能帮助我们知道所搜索的参数空间机制是否正确。通常不只一个参数要优化，参数的取值也可能非常大。比如，在第13章中我们描述了正则项的概念，其中学习算法的参数是实数。在这种情况下，一种粗略的网格搜索参数 S 的值，绘制相应的模型选择曲线。我们将基本曲线缩放到正确的尺度，然后采用更好的网格搜索。验证我们使用的机制是否正确非常重要。比如，在多项式拟合问题的描述，如果我们开始搜索多项式次数 $\{1, 10, 20\}$ 集合，但是没有采用一个基于正确结果曲线的网格搜索，最终会得到一个较差的模型。

118

11.2.4 k 折交叉验证

到目前为止，所描述的验证程序假定数据是足够大的，并且我们有能力对一个新的验证集采样。但是在一些应用中数据很少，我们不想将数据浪费在验证集上。k 折交叉验证技术正是为在不浪费太多数据的情况下，精确估计真实误差而设计。

在 k 折交叉验证中，将原训练集拆分为样本数量为 m/k 的 k 折样本子集(简单起见，假定 m/k 是一个整数)。对于每一折样本，这个算法是在其他折样本的联合样本上训练，然后由这一折的样本上估计输出的错误。最终，所有误差的平均即为真实误差的估计。特殊情形 $k=m$，这里 m 表示样本数量，这种方法称为留一验证法(LOO)。

k 折交叉验证经常用于模型选择(或参数优化)，并且一旦选择了最好的参数，这个算法被限制使用这组最优的参数在整个训练集上。k 折模型选择交叉验证的伪代码给出如下。这个过程输入训练集 S，可能的参数集合 Θ，整数 k(表示折数)，以及一个学习算法 A(A 输入一个训练集和参数 $\theta \in \Theta$)，它输出整个训练集上由此参数训练过的最佳的参数和假设。

k 折交叉验证用于模型选择

输入：

　训练集 $S=(x_1, y_1), \cdots, (x_m, y_m)$

　参数值集合 Θ

　学习算法 A

　整数 k

拆分： 将 S 拆分为 $S_1, S_2, \cdots, S_k$

对于每一个 $\theta \in \Theta$

　循环 $i=1, \cdots, k$

　　$h_{i,\theta}=A(S/S_i; \theta)$

　$\text{error}(\theta)=\frac{1}{k}\sum_{i=1}^{k} L_{S_i}(h_{i,\theta})$

输出：

　$\theta^*=\text{argmin}_\theta[\text{error}(\theta)]$

　$h_{\theta^*}=A(S; \theta^*)$

实践中，交叉验证方法通常取得很好的效果，尽管它也有可能失败，像练习 11.1 所示的人工训练一样。严格来说，理解交叉验证的精确行为仍是一个有争议的问题。1978 年，Rogers 和 Wagner 的研究显示 k 个局部规则(比如 19 章的 k 近邻)，交叉验证程序给出真实错误的好的估计。一些研究显示交叉验证对稳定算法非常有效(在 13 章中，我们将学习稳定性和相关的可学习性)。 [119]

11.2.5　训练-验证-测试拆分

大多数实际应用中，我们将可利用的样本拆分成 3 个集合。第一个集合用于训练我们的算法，第二个集合用于模型选择的验证数据集。选择最优模型后，我们在第三个数据集上测试输出预测器的性能。第三个数据集我们称之为测试数据集。测试集上的测试结果被用于估计学习预测器的真实错误。

11.3　如果学习失败了应该做什么

试想下面的场景：当你接到一个学习任务，需要选择一个假设类、一个学习算法和参数来想办法解决它。你使用一个验证集来优化参数并在测试数据集上测试学习预测器。不幸的是，测试结果并不令人满意。那么问题在哪里，我们接下来应该怎么做呢？

很多因素是已知的。主要方法如下：

- 增大样本集
- 改变假设类

- 扩大假设类
- 缩减假设类
- 彻底改变它
- 改变参数
- 改变数据的特征表示
- 应用学习规则改变优化算法

为了找到最好的改进策略，首先，理解损坏性能的原因非常重要。回想第5章我们将学习预测器的真实误差分解为近似误差和估计误差。对于$h^* \in \underset{h \in \mathcal{H}}{\arg\min} L_{\mathcal{D}}(h)$，近似误差定义为$L_{\mathcal{D}}(h^*)$，估计误差定义为$L_{\mathcal{D}}(h_S)-L_{\mathcal{D}}(h^*)$，这里$h_S$表示学习预测器(建立在训练集$S$之上)。

类的近似错误不依赖于样本数量或所使用的算法。它只依赖于分布$\mathcal{D}$和假设类$\mathcal{H}$。因此，如果近似误差太大，它将不会帮助我们扩大训练样本数量，而且对于降低假设类没有意义。在这种情况下，扩大假设类或将其彻底改变是有用的(如果我们通过不同的假设类形式有一些可选的先验知识)。我们能考虑应用同样的假设类，但是应用数据的不同的特征表示(详见第25章)。

130

类的错误估计强烈依赖于样本数量。因此，如果有大的估计错误，我们可以努力获取更多的训练样本。我们也可以考虑减少假设类，但是，在这种情况下，它对于扩大假设类没有什么意义。

1. 使用验证分解误差

弄清问题是近似误差还是估计误差，对于找到最好的改进策略是非常重要的。在先前的章节中我们看到如何通过在验证集上使用经验风险估计$L_{\mathcal{D}}(h_S)$。但是，估计类的近似错误更加困难。替代的方法是，我们给出一个不同的误差分解，这种分解可以从训练集和测试集估计得到。

$$L_{\mathcal{D}}(h_S) = (L_{\mathcal{D}}(h_S) - L_V(h_S)) + (L_V(h_S) - L_S(h_S)) + L_S(h_S)$$

第一项$L_{\mathcal{D}}(h_S)-L_V(h_S)$可以使用定理11.1进行一个很紧的界。简单而言，当第二项$L_V(h_S)-L_S(h_S)$较大时，我们称算法由于过拟合而损害了效果，当经验风险较大时，我们说算法由于欠学习而受损。注意到这两项不是必要的估计误差和近似误差的好的估计。为了说明这一点，考虑这种情况，$\mathcal{H}$类有VC维d，$\mathcal{D}$是一个分布，$\mathcal{H}$关于$\mathcal{D}$的近似误差是1/4。只要训练样本的数量小于d，对于每个经验误差最小化假设，我们将得到$L_S(h_S)=0$。因此，训练风险$L_S(h_S)$和近似误差$L_{\mathcal{D}}(h^*)$有本质的不同。但是，像我们后面将看到的，$L_S(h_S)$和$L_V(h_S)-L_S(h_S)$将提供有用的信息。

首先考虑$L_S(h_S)$很大，我们将$L_S(h_S)$写成

$$L_S(h_S) = (L_S(h_S) - L_S(h^*)) - (L_{\mathcal{D}}(h^*) - L_{\mathcal{D}}(h^*)) + L_{\mathcal{D}}(h^*)$$

当h_S是经验风险最小化假设，我们有$L_S(h_S)-L_S(h^*)\leq 0$。此外，由于h^*不依赖于S，$L_S(h^*)-L_{\mathcal{D}}(h^*)$项能够得到一个更紧的界(如定理11.1所示)。最后一项是近似误差。接下来如果$L_S(h_S)$很大，则近似误差会很大，失败算法的改进应该有相应的设计(像先前讨论的那样)。

评注　类的近似误差可能很小，但是$L_S(h_S)$值很大。例如，我们在执行经验风险最小化时可能碰到困难，这个算法返回假设h_S它不是经验误差最小化。经常会出现经验误差最小化计算困难，我们的算法使用一些启发式算法尝试找到一个近似的经验误差最小化。在一些情况下，很难知道h_S与经验风险最小化假设的接近程度。但至少我们知道它

们是否是好的假设。例如，在下一章，我们将研究凸学习问题，优化条件就是优化算法是否能够优化到经验风险最小化。在其他情况下，这种解决方案依赖于算法的随机初始值，所以我们随机选择不同的初始值以确实是否能发现更好的解决方案。

下面考虑 $L_S(h_S)$很小的情况。根据我们之前讨论的，近似误差小不是必要的。的确，考虑这两个场景，我们都使用经验风险最小化学习规则，尽力学习 VC 维为 d 的假设类。在第一个场景，我们有一个样本数量为 $m<d$ 训练集，并且类的近似误差很大。在第二个场景，我们有一个样本数量为 $m>2d$ 训练集，并且类的近似误差为 0。在两个场景中 $L_S(h_S)=0$，我们怎样区分这两个场景呢？ 121

2 学习曲线

区分这两种场景的一个可行方式是绘制学习曲线。为了获得学习曲线，我们让样本在数量不断增加的无前缀数据集上训练算法。例如，首先，我们训练样本集的前 10%，然后训练样本集的 20%，以此类推。对每一个前缀，我们计算训练错误(在这个前缀上训练算法)和验证错误(在一个提前定义的验证集上)。这样的学习曲线能帮我们区分先前提到的这两个场景。在第一个场景中，我们期望验证集错误近似所有的前缀的 1/2，而我们没有真正学到什么东西。在第二个场景中，验证错误将从常数开始，然后开始下降(当训练样本数量大于 VC 维时，误差开始下降)。这两种情况可用图 11.1 说明。

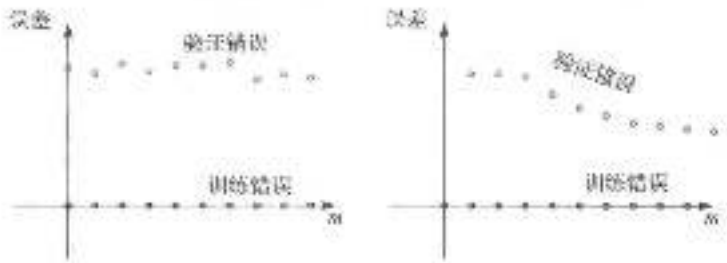


图 11.1 学习曲线的例子。左图：学习曲线与场景相关，在这些场景中，样本数量远常比类的 VC 维小。右图：学习曲线与场景相关，在这些场景中，近似错误是 0，样本数量比类 VC 维大

通常情况下，只要近似误差大于 0，训练误差就会随样本数量而增加，但数据量过大使得很难对这些给出一个解析。因此数据量越大，越难给出完整的解析。另一方面，验证错误随着样本数量增加逐渐减小。如果 VC 维是有限的，当样本数量达到无限时，验证误差和训练误差收敛到一个近似误差。因此，通过推断训练和验证曲线我们尽力猜测近似误差的值，或者至少得到一个近似误差大概区间的估计。

回到为失败算法寻找最好改进方法的问题，如果 $L_S(h_S)$很小，但是验证误差很大，那么在这种情况下类$\mathcal{H}$的训练误差集是不够的。此时可以画出学习曲线。如果验证误差开始下降，那么最好的解决方案是增加样本数量(如果我们可以扩大数据)。另一个合理的解决方案就是减少假设类的复杂度。另一方面，如果验证集错误保持在 1/2 左右，那么我们没有证据表明$\mathcal{H}$的近似错误已经足够好。此时增大训练集可能根本没有帮助，获得更多的数据仍能帮助我们，因为在这个点上，我们可以看到验证错误是否开始下降，训练误差是否开始增加。但是，如果获得更多数据代价很昂贵，最好首先尽力降低假设类的复杂度。 122

总结以上的讨论，应该采取以下步骤：

1) 如果学习包括参数优化，画出模型选择曲线来确认你已经近似优化参数(详见 11.2.3 节)。

2) 如果扩大假设类，训练误差特别大，那就彻底改变它，或者改变数据的特征表示

方法。

3）如果训练误差很小，画出学习曲线，尽力推断误差是来源于估计误差还是近似误差。

4）如果近似误差看起来足够小，尝试获得更多的数据。如果这不太可能，我们则考虑减少假设类的复杂度。

5）如果近似误差很大，尝试改变假设类或者彻底改变特征表示方式。

11.4　小结

模型选择的任务就是基于数据本身选择一个近似学习模型。我们揭示了如何使用结构风险最小化原理或者更实用的验证方法做到这一点。如果学习算法失败了，应该使用学习曲线来分解算法的误差，以便找到最佳改进方法。

11.5　练习

11.1　**k 折交叉验证失败**　试想根据 $\mathbb{P}[y=1]=\mathbb{P}[y=0]=1/2$ 随机选择标签的情形。取一个学习算法，如果训练集标签是 1，则输出常数预测值 $h(x)=1$；其他情况下算法输出的常数预测 $h(x)=0$。证明：在这种情况下，留一验证估计误差和真实误差之差总是 1/2。

11.2　令 $\mathcal{H}_1, \ldots, \mathcal{H}_k$ 是 k 个假设类。假定给你 m 个独立同分布的训练样本，并且你想学习类 $\mathcal{H}=\bigcup_{i=1}^{k}\mathcal{H}_i$，考虑两个可选的方法：

- 使用经验风险最小化规则，在 m 个样本上学习 $\mathcal{H}$。
- 将 m 个样本拆分为样本数量为 $(1-\alpha)m$ 的训练样本和样本数量为 αm 的验证集，$\alpha\in(0,1)$。然后，应用基于验证的模型选择方法，即，首先使用关于 $\mathcal{H}_i$ 的经验风险最小化规则，在 $(1-\alpha)m$ 个训练样本上训练类 $\mathcal{H}_i$。令 $\hat{h}_1, \ldots, \hat{h}_k$ 表示结果假设。然后，在 αm 个验证样本集上，在有限类 $\{\hat{h}_1, \ldots, \hat{h}_k\}$ 上应用经验风险最小化原则。

描述第一个方法优于第二个方法的情景，并描述第二个方法优于第一个方法的情景。

[123]

凸学习问题

本章主要介绍凸学习问题。绝大多数可以有效学习的问题属于凸学习的范畴，所以凸学习问题包含着一系列重要的学习问题。例如，我们已经遇到的具有平方损失和逻辑斯谛回归的线性回归问题都是凸问题，并且这些问题的确可以被有效地学习。此外，我们也看到一些非凸的问题，如半空间的 0—1 损失问题，在不可分的情形中，在无法实现的情况下，计算学习该问题是比较困难的。

通常，一个凸学习问题的假设类是一个凸集，并且对于每一个样本而言，它的损失函数是一个凸函数。本章从凸性的一些必要的定义讲起。除了凸性，还将定义损失函数的其他性质，如利普希茨性、光滑性，这些性质能帮助我们成功地学习。接下来，定义凸学习问题，并说明进一步约束的必要性，如有界性、利普希茨性或光滑性。我们定义这些更加受限的学习问题，并断言凸光滑/利普希茨有界的学习问题是可学习的。我们将在后面的两章证明这些断言，并给出两种学习范例，这两种范例可以成功地学习所有的凸利普希茨有界问题或凸光滑有界问题。

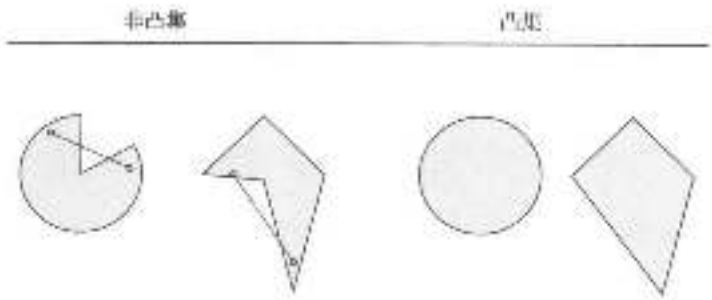
最后，我们将在 12.3 节中说明如何通过极小化凸替代损失函数来处理一些非凸的问题，即原本的损失函数是非凸的。替代凸损失函数可以得到有效的解，但是可能会增加学习到的预测器的风险。

12.1 凸性、利普希茨性和光滑性

12.1.1 凸性

定义 12.1（凸集） 设 C 是向量空间的一个集合，若对 C 中任意两点 u 和 v，连接它们的线段仍在 C 中，那么集合 C 是一个凸集；换言之，对任一实数 $\alpha\in[0,1]$，都有 $\alpha u+(1-\alpha)v\in C$。 [124]

下图给出的是 $\mathbb{R}^2$ 中凸集和非凸集的几个例子。对于非凸集而言，连接两点的线段不在集合中。



给定 $\alpha\in[0,1]$，$\alpha u+(1-\alpha)v$ 称为 u 和 v 的凸组合。

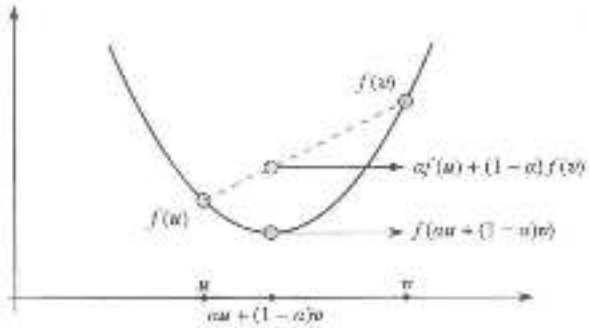
定义 12.2（凸函数） 设 C 是一个凸集，如果对任意的 $u,v\in C$ 及 $\alpha\in[0,1]$，函数

$f: C \to \mathbb{R}$ 满足

$$f(\alpha u+(1-\alpha)v) \leqslant \alpha f(u)+(1-\alpha)f(v)$$

则称 f 为 C 上的凸函数。

换句话说，对于任意的 u 和 v，如果函数 f 在 u 和 v 之间的图形位于连接 $f(u)$ 和 $f(v)$ 的线段的下方，那么 f 是凸函数。下图给出了凸函数 $f: \mathbb{R} \to \mathbb{R}$ 的几何解释。

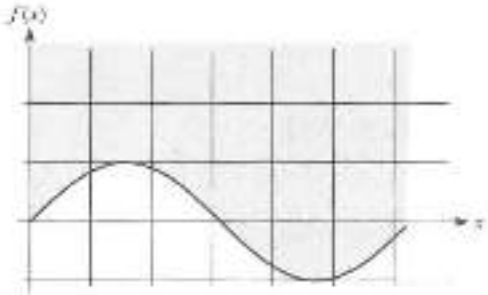


函数 f 的上境图(epigraph)是集合

$$\text{epigraph}(f)=\{(x,\beta): f(x) \leqslant \beta\} \tag{12.1}$$

容易证明函数 f 是凸的当且仅当它的上境图是一个凸集。下图给出的是非凸函数 $f: \mathbb{R} \to \mathbb{R}$ 以及它的上境图。

[125]



凸函数的一个重要性质是它的每一个局部极小值也是全局极小值。设 $B(u, r)=\{v: |v-u| \leqslant r\}$ 是一个以 u 为球心 r 为半径的球，如果存在某个 $r>0$ 使得对于任意的 $v \in B(u, r)$ 都有 $f(v) \geqslant f(u)$，那么我们说 $f(u)$ 是 f 在 u 处的一个局部极小值。于是，对于任意的 v(不一定在 B 中)，存在一个充分小的 $\alpha>0$ 使得 $u+\alpha(v-u) \in B(u, r)$，并且成立

$$f(u) \leqslant f(u+\alpha(v-u)) \tag{12.2}$$

如果 f 是凸的，那么

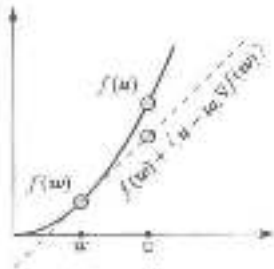
$$f(u+\alpha(v-u))=f(\alpha v+(1-\alpha)u) \leqslant (1-\alpha)f(u)+\alpha f(v) \tag{12.3}$$

由式(12.2)和式(12.3)可得 $f(u) \leqslant f(v)$。由于该式对每一个 v 都成立，所以 $f(u)$ 是 f 的一个全局极小值。

凸函数另一个重要的性质是对每一个 w，我们可以构造 f 在 w 处的切线，该切线始终位于函数 f 的下方。如果 f 是可微的，那么该切线是一个线性函数 $l(u)=f(w)+\langle \nabla f(w), u-w\rangle$，其中 $\nabla f(w)$ 表示 f 在 w 处的梯度，即 f 的偏导数向量 $\nabla f(w)=\left(\frac{\partial f(w)}{\partial w_1}, \ldots, \frac{\partial f(w)}{\partial w_d}\right)$，也就是说，对凸可微函数而言，

$$\forall u, f(u) \geq f(w) + \langle \nabla f(w), u - w \rangle \tag{12.4}$$

在第 14 章中，我们将把该不等式推广至不可微函数。下面给出了(12.4)式的图解说明。



如果 f 是一个可微的标量函数，那么可以验证它也是一个凸函数。

引理 12.3　设 $f: \mathbb{R} \to \mathbb{R}$ 是一个二阶可微的标量函数，f' 和 f'' 分别表示函数 f 的一阶导和二阶导函数，那么下面的命题是等价的：

1. f 是凸的；
2. f' 是单调不减的；
3. f'' 是非负的。

[126]

例 12.1

- 标量函数 $f(x) = x^2$ 是凸的。注意到需知 $f'(x) = 2x$ 和 $f''(x) = 2 \geq 0$。
- 标量函数 $f(x) = \log(1 + \exp(x))$ 是凸的。注意到需知 $f'(x) = \frac{\exp(x)}{1+\exp(x)} = \frac{1}{\exp(-x)+1}$。因为指数函数是一个单调递增的函数，所以 $f'(x)$ 是一个单调递增的函数。 ◀

下面的断言表明一个凸标量函数和一个线性函数的组合得到的是一个凸向量值函数。

论断 12.4　假设对于某个 $x \in \mathbb{R}^d$，$y \in \mathbb{R}$ 和 $g: \mathbb{R}^n \to \mathbb{R}$，函数 $f: \mathbb{R}^d \to \mathbb{R}$ 可以写成 $f(w) = g(\langle w, x \rangle + y)$。那么 g 的凸性蕴含着 f 的凸性。

证明　设 $w_1, w_2 \in \mathbb{R}^d$，$\alpha \in [0, 1]$，则

$$\begin{aligned} f(\alpha w_1 + (1-\alpha) w_2) &= g(\langle \alpha w_1 + (1-\alpha) w_2, x \rangle + y) \\ &= g(\alpha \langle w_1, x \rangle + (1-\alpha) \langle w_2, x \rangle + y) \\ &= g(\alpha(\langle w_1, x \rangle + y) + (1-\alpha)(\langle w_2, x \rangle + y)) \\ &\leq \alpha g(\langle w_1, x \rangle + y) + (1-\alpha) g(\langle w_2, x \rangle + y) \end{aligned}$$

其中最后一个不等式由 g 的凸性得到。 ■

例 12.2

- 给定 $x \in \mathbb{R}^d$ 和 $y \in \mathbb{R}$，设 $f(w) = (\langle w, x \rangle - y)^2$ 是定义在 $\mathbb{R}^d$ 上的实函数，那么 f 是函数 $g(a) = a^2$ 在线性函数上的一个组合，且 f 是凸的。
- 给定 $x \in \mathbb{R}^d$ 和 $y \in \{\pm 1\}$，设 $f(w) = \log(1 + \exp(-y\langle w, x \rangle))$ 是定义在 $\mathbb{R}^d$ 上的实函数，那么 f 是函数 $g(a) = \log(1 + \exp(a))$ 在线性函数上的一个组合，且 f 是凸的。

下面的论断表明凸函数的最大化是凸的；加权的凸函数的和也是凸的。

论断 12.5 设 $f_i: \mathbb{R}^n \to \mathbb{R}$ 是凸函数，$i=1, \cdots, r$，那么下面定义在 $\mathbb{R}^n$ 上的实函数也都是凸函数。

- $g(x)=\max_{i\in[r]} f_i(x)$
- $g(x)=\sum_{i=1}^{r} w_i f_i(x)$，其中对于任意的 i，$w_i \geq 0$。

127

证明

(1)

$$\begin{aligned} g(\alpha u+(1-\alpha)v) &= \max_i f_i(\alpha u+(1-\alpha)v) \\ &\leq \max_i [\alpha f_i(u)+(1-\alpha)f_i(v)] \\ &\leq \alpha \max_i f_i(u)+(1-\alpha)\max_i f_i(v) \\ &\leq \alpha g(u)+(1-\alpha)g(v) \end{aligned}$$

(2)

$$\begin{aligned} g(\alpha u+(1-\alpha)v) &= \sum_i w_i f_i(\alpha u+(1-\alpha)v) \\ &\leq \sum_i w_i [\alpha f_i(u)+(1-\alpha)f_i(v)] \\ &\leq \alpha \sum_i w_i f_i(u)+(1-\alpha)\sum_i w_i f_i(v) \\ &\leq \alpha g(u)+(1-\alpha)g(v) \end{aligned}$$

■

例 12.3 函数 $g(x)=|x|$ 是凸的。注意到需知 $g(x)=\max\{x, -x\}$，且函数 $f_1(x)=x$ 和 $f_2(x)=-x$ 都是凸的。 ◀

12.1.2 利普希茨性

利普希茨性的定义是在 $\mathbb{R}^n$ 空间上对欧氏范数而言的。然而，我们可以定义关于任意范数的利普希茨性。

定义 12.6（利普希茨性） 设 $C \subset \mathbb{R}^n$，$f: \mathbb{R}^n \to \mathbb{R}^m$，如果对于任意的 $w_1, w_2 \in C$，有 $\|f(w_1)-f(w_2)\| \leq \rho \|w_1-w_2\|$，那么 f 是 ρ-利普希茨。

直观地说，一个利普希茨函数不会变化太快。如果函数 $f: \mathbb{R} \to \mathbb{R}$ 是可微的，那么由中值定理可知

$$f(w_1)-f(w_2)=f'(u)(w_1-w_2)$$

其中 u 位于 w_1 和 w_2 之间。由此断定，如果 f 的导数按绝对值处处以 ρ 为界，那么函数 f 是 ρ-利普希茨。

例 12.4

- 函数 $f(x)=|x|$ 在 $\mathbb{R}$ 上是 1-利普希茨的。这个可以由三角不等式推得：对于每个 x_1, x_2，有
  $$|x_1|-|x_2|=|x_1-x_2+x_2|-|x_2| \leq |x_1-x_2|+|x_2|-|x_2|=|x_1-x_2|$$
  进一步，得 $||x_1|-|x_2|| \leq |x_1-x_2|$。
- 函数 $f(x)=\log(1+\exp(x))$ 在 $\mathbb{R}$ 上是 1-利普希茨的。注意到

$$|f'(x)| = \left|\frac{\exp(x)}{1+\exp(x)}\right| = \left|\frac{1}{\exp(-x)+1}\right| \leqslant 1$$ 128

- 对于任意的 ρ，函数 $f=x^2$ 在 $\mathbb{R}$ 上不是 ρ-利普希茨。令 $x_1=0$，$x_2=1+\rho$，那么
$$f(x_2)-f(x_1) = (1+\rho)^2 \geqslant \rho(1+\rho) = \rho|x_2-x_1|$$
然而，该函数在集合 $C=\{x: |x|\leqslant\rho/2\}$ 上是 ρ-利普希茨。注意到，对于任意的 $x_1, x_2\in C$，有
$$|x_1^2-x_2^2| = |x_1-x_2||x_1+x_2| \leqslant 2(\rho/2)|x_1-x_2| = \rho|x_1-x_2|$$
- f 是 $\mathbb{R}^d$ 上的实线性函数，定义为 $f(w)=\langle v, w\rangle+b$，其中 $v\in\mathbb{R}^d$ 是 $\|v\|$ 利普希茨的。由柯西-施瓦茨不等式，得
$$|f(w_1)-f(w_2)| = |\langle v, w_1-w_2\rangle| \leqslant \|v\|\,\|w_1-w_2\|$$
下面的论断表明利普希茨函数的组合仍具有利普希茨性。 ◀

论断 12.7　设 $f(x)=g_1(g_2(x))$，其中 g_1 是 ρ_1-利普希茨，g_2 是 ρ_2-利普希茨，那么 f 是 $(\rho_1\rho_2)$-利普希茨。特别地，如果 g_2 是线性函数，对于 $v\in\mathbb{R}^d$，$b\in\mathbb{R}$，$g_2(x)=\langle v, x\rangle+b$，那么 f 是 $(\rho_1\|v\|)$-利普希茨。

证明

$$\begin{aligned}|f(w_1)-f(w_2)| &= |g_1(g_2(w_1))-g_1(g_2(w_2))|\\ &\leqslant \rho_1|g_2(w_1)-g_2(w_2)|\\ &\leqslant \rho_1\rho_2|w_1-w_2|\end{aligned}$$ ■

12.1.3　光滑性

光滑函数的定义依赖于梯度的概念。可微函数 $f: \mathbb{R}^d\to\mathbb{R}$ 在 w 处的梯度是 f 的偏导数，记为 $\nabla f(w)=\left(\frac{\partial f(w)}{\partial w_1}, \ldots, \frac{\partial f(w)}{\partial w_d}\right)$。

定义 12.8（光滑性）　如果可微函数 $f: \mathbb{R}^d\to\mathbb{R}$ 的梯度是 β-利普希茨，即对于所有的 v，w，满足 $\|\nabla f(v)-\nabla f(w)\|\leqslant\beta|v-w|$，那么 f 是 β-光滑。

可以看出光滑性意味着对于所有的 v，w，有

$$f(v)\leqslant f(w)+\langle\nabla f(w), v-w\rangle+\frac{\beta}{2}\|v-w\|^2 \tag{12.5}$$

注意到函数 f 的凸性意味着 $f(v)\geqslant f(w)+\langle\nabla f(w), v-w\rangle$。所以，当一个函数既凸又光滑的时候，我们可以同时得到函数与其一阶近似差值的上下界。

令式(12.5)右端的 $v=w-\frac{1}{\beta}\nabla f(w)$，可得

$$\frac{1}{2\beta}\|\nabla f(w)\|^2\leqslant f(w)-f(v)$$ 129

进一步，假设对于所有的 v 有 $f(v)\geqslant 0$，那么可以推断光滑性也意味着

$$\|\nabla f(w)\|^2\leqslant 2\beta f(w) \tag{12.6}$$

满足这个性质的函数也称为自有界(self-bounded)函数。

例 12.5

- 函数 $f(x)=x^2$ 是 2-光滑。注意到 $f'(x)=2x$，且对这个特殊的函数，式(12.5)与式(12.6)都以等式成立。

- 函数 $f(x)=\log(1+\exp(x))$ 是(1/4)-光滑。注意到 $f'(x)=\frac{1}{1+\exp(x)}$，于是

$$|f''(x)|=\frac{\exp(-x)}{(1+\exp(-x))^2}=\frac{1}{(1+\exp(-x))(1-\exp(x))}\leqslant 1/4$$

因此，f' 是(1/4)-利普希茨。由于该函数非负，所以式(12.6)也成立。 ◀

下面的论断表明一个光滑的标量函数在线性函数上的组合仍具有光滑性。

论断 12.9　设 $f(w)=g(\langle w,x\rangle+b)$，其中函数 $g:\mathbb{R}\to\mathbb{R}$ 是 β-光滑，$x\in\mathbb{R}^d$，$b\in\mathbb{R}$，那么 f 是 $(\beta\|x\|^2)$-光滑。

证明　由链式规则得 $\nabla f(w)=g'(\langle w,x\rangle+b)x$，其中 g' 是 g 的导数。利用 g 的光滑性和柯西-施瓦茨不等式，得

$$\begin{aligned}
&f(v)-g(\langle v,x\rangle+b)\\
&\leqslant g(\langle w,x\rangle+b)-g'(\langle w,x\rangle+b)\langle v-w,x\rangle+\frac{\beta}{2}(\langle v-w,x\rangle)^2\\
&\leqslant g(\langle w,x\rangle+b)-g'(\langle w,x\rangle+b)\langle v-w,x\rangle+\frac{\beta}{2}(\|v-w\|\|x\|)^2\\
&\leqslant f(w)+\langle\nabla f(w),v-w\rangle+\frac{\beta\|x\|^2}{2}\|v-w\|^2
\end{aligned}$$

■

例 12.6

- 对于任意的 $x\in\mathbb{R}^d$，$y\in\mathbb{R}$，设 $f(w)=(\langle w,x\rangle-y)^2$，于是 f 是 $(2\|x\|^2)$-光滑。
- 对于任意的 $x\in\mathbb{R}^d$，$y\in\{\pm1\}$，设 $f(w)=\log(1+\exp(-y\langle w,x\rangle))$，于是 f 是 $(\|x\|^2/4)$-光滑。 ◀

12.2　凸学习问题概述

注意到学习的一般性定义(第3章定义3.4)包含着三个要素：假设类$\mathcal{H}$，样本集 Z 和损失函数 $\ell:\mathcal{H}\times Z\to\mathbb{R}_+$。到目前为止，本书主要考虑 Z 是一个实例空间和一个目标空间的乘积，即 $Z=\mathcal{X}\times\mathcal{Y}$，$\mathcal{H}$是从$\mathcal{X}$到$\mathcal{Y}$的函数集合。然而，$\mathcal{H}$可以是任意的集合。在这一章中，我们考虑$\mathcal{H}$都是欧几里得空间 $\mathbb{R}^d$ 的子集。也就是说，每个假设是某个实值向量。所以，我们可以将$\mathcal{H}$记为 w。现在，我们终于可以定义凸学习问题了。

130

定义 12.10(凸学习问题)　如果假设类$\mathcal{H}$是凸集，且对于任意的 $z\in Z$，损失函数 $\ell(\cdot,z)$是凸函数，那么学习问题$(\mathcal{H},Z,\ell)$是凸的。这里，对于任意的 z，$\ell(\cdot,z)$表示由 $f(w)=\ell(w,z)$定义的函数 $f:\mathcal{H}\to\mathbb{R}$。

例 12.7(具有平方损失的线性回归)　注意到线性回归是一个可以模拟"解释性"变量与实值输出(参考第9章)之间关系的工具。定义域$\mathcal{X}$是 $\mathbb{R}^d$ 的一个子集，标签集$\mathcal{Y}$是由一些实数构成的集合。我们的目标是学习出一个能最好地近似变量之间关系的线性函数 $h:\mathbb{R}^d\to\mathbb{R}$。在第9章中，我们把假设类定义为由齐次线性函数组成的集合$\mathcal{H}=\{x\mapsto\langle w,x\rangle:w\in\mathbb{R}^d\}$，并使用平方损失函数 $\ell(h,(x,y))=(h(x)-y)^2$。然而，我们可以将学习问题等价地描述为一个凸学习问题。每个线性函数均由向量 $w\in\mathbb{R}^d$ 进行参数化。样本集 $Z=\mathcal{X}\times\mathcal{Y}=\mathbb{R}^d\times\mathbb{R}=\mathbb{R}^{d+1}$，损失函数 $\ell(w,(x,y))=(\langle w,x\rangle-y)^2$。显然，$\mathcal{H}$是一个凸集。损失函数关于它的第一个变量($w$)也是凸的(参考例12.2)。 ◀

引理 12.11　如果损失函数 ℓ 是凸函数，假设类$\mathcal{H}$是凸集，那么ERM$_{\mathcal{H}}$问题(在$\mathcal{H}$上极

小化经验损失)是一个凸优化问题；也就是相当于在一个凸集上极小化一个凸函数。

证明　$\mathrm{ERM}_{\mathcal{H}}$ 问题定义为

$$\mathrm{ERM}_{\mathcal{H}}(S)=\underset{w\in\mathcal{H}}{\operatorname{argmin}}L_S(w)$$

又 $S=z_1,\cdots,z_m$，对于每个 w，$L_S(w)=\frac{1}{m}\sum_{i=1}^{m}\ell(w,z_i)$。论断 12.5 意味着 $L_S(w)$ 是一个凸函数。因此，ERM 规则就是极小化一个凸函数并使求得的解包含在一个凸集之中。■

在适当的条件下，这样的问题可以通过一般的优化算法进行求解。特别地，我们将在第 14 章中给出一个非常简单的极小化凸函数的算法。

12.2.1　凸学习问题的可学习性

对于很多情形，利用 ERM 规则可以有效地求解凸学习问题。但是，凸性是否是问题可学习性的充分条件呢？

为了使问题更加具体：在 VC 维中，我们知道 d 维的半空间是可学习的(或许效率比较低)。在第 9 章我们说如果问题含有 d 个参数，那么使用"离散技巧"，问题是可学习的，此时采样复杂度是一个关于 d 的函数。也就是说，对于一个常数 d 而言，问题应该是可学习的。那么，是不是 $\mathbb{R}^d$ 上所有的凸学习问题都是可学习的呢？ 131

后面的例 12.8 表明即使在 d 很低的情况下，答案也是否定的。不是 $\mathbb{R}^d$ 上所有的凸学习问题都是可学习的。这和 VC 维理论并不矛盾，因为 VC 维理论只解决二分类问题，而这里我们考虑的是一类更广泛的问题。这和"离散技巧"也不矛盾，因为我们假设损失是有界的，同时假设用有限数量的位来表示每个参数就足够了。正如我们后面将要说明的，在许多实际情况中，如果添加一些额外的约束条件，那么凸问题是可学习的。

例 12.8　(线性回归的不可学习性，包括 $d=1$ 的情形)　设 $\mathcal{H}=\mathbb{R}$，损失函数为平方损失：$\ell(w,(x,y))=(wx-y)^2$(我们指的是齐次的情况)。设 A 是任意一个确定的算法[⊖]。利用反证法，假设对于该问题来讲，A 是一个成功的 PAC 学习器，也就是说，存在一个函数 $m(\cdot,\cdot)$，使得对于每个分布 $\mathcal{D}$，ε，δ，如果 A 收到一个大小为 $m\geq m(\varepsilon,\delta)$ 的训练集，那么它至少以 $1-\delta$ 的概率输出假设 $\hat{w}=A(S)$，使得 $L_{\mathcal{D}}(\hat{w})-\min_{w}L_{\mathcal{D}}(w)\leq\varepsilon$。

令 $\varepsilon=1/100$，$\delta=1/2$，$m\geq m(\varepsilon,\delta)$，$\mu=\frac{\log(100/99)}{2m}$。我们将定义两种分布，并说明 A 有可能至少在其中一个分布上失效。第一个分布 $\mathcal{D}_1$ 由两个样本 $z_1=(1,0)$ 和 $z_2=(\mu,-1)$ 支撑，第一个样本的概率质量函数是 μ，第二个样本的概率质量函数是 $1-\mu$。第二个分布 $\mathcal{D}_2$ 完全由样本 z_2 支撑。◀

注意到对于两个分布来讲，训练集的所有样本属于第二类的概率至少是 99%。对分布 $\mathcal{D}_2$ 而言，这是显然的。而对 $\mathcal{D}_1$ 而言，该事件的概率是

$$(1-\mu)^m\geq e^{-2\mu m}=0.99$$

既然我们假设 A 是一个确定的算法，当 A 接收到一个由 m 个样本组成的训练集时，其中每个样本都是 $(\mu,-1)$，算法会输出某个 $\hat{w}$。此时，如果 $\hat{w}<-1/(2\mu)$，我们令分布为 $\mathcal{D}_1$。因此

⊖ 在给定 S 的前提下，输出 A 是确定的。这只是为了方便起见。此外，不确定性算法是不可以用来学习的。

$$L_{\mathcal{D}_1}(\hat{w}) \geq \mu(\hat{w})^2 \geq 1/(4\mu)$$

又因为

$$\min_w L_{\mathcal{D}_1}(w) \leq L_{\mathcal{D}_1}(0) = (1-\mu)$$

于是

$$L_{\mathcal{D}_1}(\hat{w}) - \min_w L_{\mathcal{D}_1}(w) \geq \frac{1}{4\mu} - (1-\mu) > \epsilon$$

所以，这样的一个算法 A 在分布 $\mathcal{D}_1$ 上是无效的。另一方面，如果 $\hat{w} \geq 1/(2\mu)$，那么我们将令分布为 $\mathcal{D}_2$。于是当 $\min_w L_{\mathcal{D}_2}(w)=0$ 时，我们有 $L_{\mathcal{D}_2}(\hat{w}) \geq 1/4$，因此算法 A 在分布 $\mathcal{D}_2$ 上是无效的。总的来说，我们说明了对于每一个 A 都存在一个分布使得 A 在该分布上是无效的，这就意味着该问题不是 PAC 可学的。

[132]

一个可能的解决方法是在假设类上添加其他的约束条件。除了凸性，我们还要求 $\mathcal{H}$ 是有界的，即假定对于某个预先给定的标量 B，每个假设 $w \in \mathcal{H}$ 都满足 $\|w\| \leq B$。

下面，举例说明有界性和凸性仍不能保证问题是可学习的。

例 12.9 在例 12.8 中，考虑平方损失的回归问题。然而，这次我们令 $\mathcal{H}=\{w: |w| \leq 1\}$ 是一个有界的假设类。不难证明 $\mathcal{H}$ 是凸的。现在，除了分布 $\mathcal{D}_1$ 和 $\mathcal{D}_2$ 分别是由 $z_1=(1/\mu, 0)$ 和 $z=(1, -1)$ 支撑外，参数和例 12.8 中是一样的。如果算法 A 收到第一类中的 m 个样本时返回 $\hat{w}<1/2$，那么我们将分布设为 $\mathcal{D}_1$，并且有

$$L_{\mathcal{D}_1}(\hat{w}) - \min_w L_{\mathcal{D}_1}(w) \geq \mu(\hat{w}/\mu) - L_{\mathcal{D}_1}(0) \geq 1/(4\mu) - (1-\mu) > \epsilon$$

类似地，如果 $\hat{w} \geq 1/2$，我们将分布设为 $\mathcal{D}_2$，且有

$$L_{\mathcal{D}_2}(\hat{w}) - \min_w L_{\mathcal{D}_2}(w) \geq (-1/2-1)^2 - 0 > \epsilon$$

这个例子说明对于学习问题我们需要其他的一些假设条件。这次的解决方法是假设损失函数具有利普希茨性或光滑性。这就促使我们给出两类学习问题的定义：凸利普希茨有界和凸光滑有界，这两个定义将在下面给出。 ◀

12.2.2　凸利普希茨/光滑有界学习问题

定义 12.12（凸利普希茨有界学习问题）　如果假设类 $\mathcal{H}$ 是一个凸集，且对于所有的 $w \in \mathcal{H}$ 都成立 $\|w\| \leq B$；对于所有的 $z \in Z$，损失函数 $\ell(\cdot, z)$ 是凸的且是 ρ-利普希茨，则称学习问题 $(\mathcal{H}, Z, \ell)$ 是凸利普希茨有界的，其中 ρ，B 是参数。

例 12.10 令 $\mathcal{X}=\{x \in \mathbb{R}^d: \|x\| \leq \rho\}$，$\mathcal{Y}=\mathbb{R}$。设假设类 $\mathcal{H}=\{w \in \mathbb{R}^d: \|w\| \leq B\}$，损失函数为 $\ell(w, (x, y))=|\langle w, x\rangle - y|$。这对应于具有绝对损失的回归问题。这里我们假设样本在一个以 ρ 为半径的球内，且限制假设和由向量 w 定义的线性函数同质，$\|w\| \leq B$。然后，得到的问题便是一个以 ρ 和 B 为参数的凸利普希茨有界的学习问题。 ◀

定义 12.13（凸光滑有界学习问题）　如果假设类 $\mathcal{H}$ 是一个凸集且对于所有的 $w \in \mathcal{H}$ 都成立 $\|w\| \leq B$；对于所有的 $z \in Z$，损失函数 $\ell(\cdot, z)$ 是凸的、非负的且是 β-光滑，那么称学习问题 $(\mathcal{H}, Z, \ell)$ 是凸光滑有界的，其中 β，B 是参数。

[133]

注意到我们要求损失函数是非负的，这是为了保证损失是自有界的。

例 12.11 令 $\mathcal{X}=\{x \in \mathbb{R}^d: \|x\| \leq \rho/2\}$，$\mathcal{Y}=\mathbb{R}$。设假设类 $\mathcal{H}=\{w \in \mathbb{R}^d: \|w\| \leq B\}$，损失函数为 $\ell(w, (x, y))=(\langle w, x\rangle - y)^2$。这对应于具有平方损失的回归问题。这里我们

假设样本在一个以 $\rho/2$ 为半径的球内，且限制假设和由向量 w 定义的线性函数同质，$\|w\|\leq B$。然后，得到的问题便是一个以 ρ 和 B 为参数的凸光滑有界的学习问题。◀

我们断言这两类学习问题是可学习的。也就是说，损失函数具有凸性、有界性和利普希茨性或光滑性是可学习的充分性。在下一章中，我们将通过能成功学习这些问题的算法来证明这个论断。

12.3　替代损失函数

正如前面所提到的，我们将在下面的章节看到凸学习问题可以被有效地求解。然而，在许多情况下，自然的损失函数不是凸的，特别地，实施 ERM 准则是困难的。

举个例子，考虑学习半空间上关于 0—1 损失的假设类问题，即

$$\ell^{0-1}(w,(x,y))=\mathbb{1}_{[y\neq \mathrm{sign}(\langle w,x\rangle)]}=\mathbb{1}_{[y\langle w,x\rangle\leq 0]}$$

这个损失函数关于 w 是非凸的，极小化该损失函数的经验风险的时候，我们得到的是一个局部极小值(见练习 12.1)。而且，如第 8 章讨论的一样，在无法实现的情况下，求解关于 0—1 损失的 ERM 问题是 NP 难的。

为了避免这个困难的结果，一个流行的方法是通过一个凸的替代损失函数来定义非凸损失函数的上界。正如这个名字所指示的，一个凸替代损失需要满足：

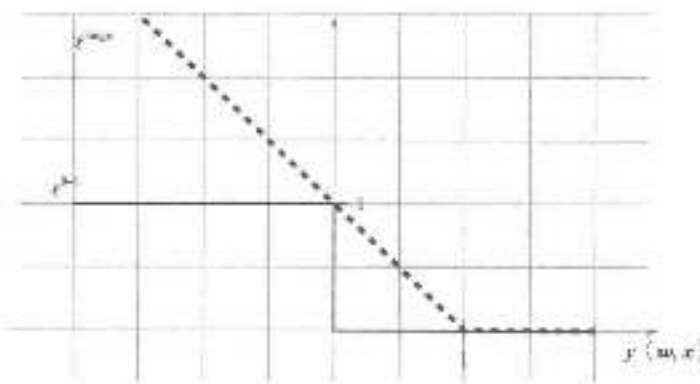
1) 它是凸的。

2) 它是原来损失函数的一个上界。

例如，在学习半空间的情况下，我们可以定义所谓的合页(hinge)损失作为 0—1 损失的凸替代，合页损失定义如下：

$$\ell^{\mathrm{hinge}}(w,(x,y))\stackrel{\mathrm{def}}{=}\max\{0,1-y\langle w,x\rangle\}$$

显然，对于所有的 w 和 (x,y)，$\ell^{0-1}(w,(x,y))\leq\ell^{\mathrm{hinge}}(w,(x,y))$。此外，合页损失函数的凸性可以直接由论断 12.5 得到。因此，对于 0—1 损失而言，合页损失函数满足凸替代损失函数的要求。函数 ℓ^{0-1} 和 ℓ^{hinge} 的示意图如下。 134



一旦我们定义了替代凸损失，关于它我们就可以学习问题了。从合页损失学习的一般要求可知

$$L_{\mathcal{D}}^{\mathrm{hinge}}(A(S))\leq\min_{w\in\mathcal{H}}L_{\mathcal{D}}^{\mathrm{hinge}}(w)+\epsilon$$

其中 $L_{\mathcal{D}}^{\mathrm{hinge}}(w)=\mathbb{E}_{(x,y)\sim\mathcal{D}}[\ell^{\mathrm{hinge}}(w,(x,y))]$。使用替代性质，由 $L_{\mathcal{D}}^{0-1}(A(S))$ 我们可以给出左端的下界，得

$$L_{\mathcal{D}}^{0-1}(A(S)) \leq \min_{w \in \mathcal{H}} L_{\mathcal{D}}^{h_w^*}(w) + \varepsilon$$

进一步，我们将上界重新写成：

$$L_{\mathcal{D}}^{0-1}(A(S)) \leq \min_{w \in \mathcal{H}} L_{\mathcal{D}}^{0-1}(w) + \left(\min_{w \in \mathcal{H}} L_{\mathcal{D}}^{h_w^*}(w) - \min_{w \in \mathcal{H}} L_{\mathcal{D}}^{0-1}(w)\right) + \varepsilon$$

也就是说，学到的预测器的 0—1 误差的上界由三部分构成：

- 逼近误差：$\min_{w \in \mathcal{H}} L_{\mathcal{D}}^{0-1}(w)$，它衡量的是在分布上假设类的表现有多么的好。在第 5 章中我们已经对这个误差项做了详细的描述。
- 估计误差：我们没能观测到分布$\mathcal{D}$，而是只接收到了一个训练集，这项误差就是基于这样一个事实得到的。同样地，在第 5 章中我们也已经详细描述了这个误差项。
- 优化误差：$\min_{w \in \mathcal{H}} L_{\mathcal{D}}^{h_w^*}(w) - \min_{w \in \mathcal{H}} L_{\mathcal{D}}^{0-1}(w)$，它衡量的是关于替代损失的逼近误差和关于原始损失的逼近误差两者之间的差异。优化误差可以认为是我们极小化关于原始损失的训练损失能力的一个结果。这个误差的大小依赖于我们所使用数据的特定分布和特定的替代损失。

12.4　小结

[136]　我们介绍了两类学习问题：凸利普希茨有界问题和凸光滑有界问题。在接下来的两章中，我们将描述两种对这两类问题而言通用的学习算法。我们还介绍了凸替代损失函数的概念，这使得我们可以用凸机制来解决非凸的问题。

12.5　文献评注

一些关于凸分析和优化的优秀书籍，如 Boyd 和 Vandenberghe(2004)，Borwein 和 Lewis(2006)，Bertsekas(1999)，Hiriart-Urruty 和 Lemarechal(1993)。Zinkevich(2003) 在在线学习的背景下第一个研究了凸利普希茨有界问题，而 Shalev-Shwartz，Shamir，Sridharan 和 Srebro(2009)则在 PAC 学习的背景下第一个研究了凸利普希茨有界问题。

12.6　练习

12.1　构造一个例子说明 0—1 损失可能得到局部极小值：即构造一个训练集 $S \in (\mathcal{X} \times \{\pm 1\})^m$(假设 $\mathcal{X} = \mathbb{R}^d$)，存在一个向量 w 和某个 $\varepsilon > 0$ 使得

1) 对任何使得 $\|w - w'\| \leq \varepsilon$ 的 w' 我们有 $L_S(w) \leq L_S(w')$，其中损失为 0—1 损失，这就意味着 w 是 L_S 的局部极小值。

2) 存在某个 w^* 使得 $L_S(w^*) < L_S(w)$。这就意味着 w 不是 L_S 的全局极小值。

12.2　考虑逻辑斯谛回归问题：设$\mathcal{H} = \mathcal{X} = \{x \in \mathbb{R}^d : |x| \leq B\}$，标量 $B > 0$，令$\mathcal{Y} = \{\pm 1\}$，损失函数 ℓ 定义为 $\ell(w, (x, y)) = \log(1 + \exp(-y\langle w, x\rangle))$。说明该问题既是凸利普希茨有界又是凸光滑有界的，并指出利普希茨性和光滑性的参数。

12.3　考虑合页损失的半空间学习问题。我们将定义域限制在半径为 R 的欧几里得球上。也就是说，$\mathcal{X} = \{x : |x| \leq R\}$。令标签集$\mathcal{Y} = \{\pm 1\}$，损失函数 ℓ 定义为 $\ell(w, (x, y)) = \max(0, 1 - y\langle w, x\rangle)$。我们已经知道该损失函数是凸的，请说明它是 R 利普希茨。

* 12.4　**凸利普希茨有界性不是计算效率的充分条件**：在下一章中，我们从统计的角度说明所有的凸利普希茨有界的问题(在不可知 PAC 模型下)是可学习的。然而，我们学习这样问题的动机是源自可计算的角度——凸问题通常可以被有效地求解。但是，

这个练习的目的是说明只有凸性不是可有效计算的充分条件。我们说明甚至在 $d=1$ 的情况下，存在凸利普希茨有界问题是不能被学习的。

设假设类为 $\mathcal{H}=[0,1]$，样本的定义域为所有的图灵机 $\mathcal{Z}$。定义如下的损失函数。对于每一个图灵机 $T\in\mathcal{Z}$，如果 T 在输入 0 处出错暂停，则令 $\ell(0,T)=1$；如果 T 在输入 0 处不出错暂停，那么令 $\ell(0,T)=0$。类似地，如果 T 在输入 0 处出错暂停，则令 $\ell(1,T)=0$，如果 T 在输入 0 处不出错暂停，那么令 $\ell(1,T)=1$。最后，对于 $h\in(0,1)$，令 $\ell(h,T)=h\ell(0,T)+(1-h)\ell(1,T)$。

1) 说明该学习问题是凸利普希茨有界的。

2) 说明没有可计算的算法能够学习该问题。

138

正则化和稳定性

在上一章中，我们介绍了一族凸利普希茨有界和凸光滑有界的学习问题。本章，我们来说明这两个族中的所有学习问题都是可学习的。对于这种形式的一些学习问题，证明一致收敛满足是可能的；因此用 ERM 准则它们是可学习的。然而，对于这种形式的所有的学习问题并不都是真的。但是，我们将要介绍另一种学习规则，并且表明它可以学习所有的凸利普希茨有界和凸光滑有界的学习问题。

在这一章中我们介绍新的学习范式，即正则损失最小化，或者简写成 RLM。在 RLM 中我们最小化经验风险和一个正则化函数的和。直观地来讲，正则化函数描述了假设的复杂度。事实上，一个正则化函数的解释是在第 7 章曾经讨论过的结构风险最小化范式。对于正则化的另一个认识是学习算法的稳定剂。如果一个算法的输入的一个小的变化不会太多地改变输出，这个算法就被看作是稳定的。我们将会在形式上定义稳定性的概念（我们所说的“输入的小的变化”和“不会太多地改变输出”分别是什么意思）并且证明它与可学习的紧密关系。最终，我们将会说明用平方 ℓ_2 范数作为正则化函数，可以让所有的凸利普希茨有界和凸光滑有界的学习问题都是稳定的。因此，对于这些学习问题的族，RLM 可以被用来作为一个一般学习规则。

13.1 正则损失最小化

正则损失最小化（RLM）是一个同时最小化经验风险和一个正则化函数的学习规则。形式上，一个正则化函数是一个映射 $R: \mathbb{R}^d \to \mathbb{R}$，正则损失最小化规则输出一个假设：

$$\operatorname*{argmin}_{w}(L_S(w) + R(w)) \tag{13.1}$$

正则损失最小化共有最小描述长度算法和结构风险最小化（参考第 7 章）的相似性。直观地讲，假设的“复杂性”用正则化函数的值来描述，而且算法平衡了低经验风险与“更简单”或者“不那么复杂”的假设。

137

我们可以用很多可能的正则化函数，它们反映了一些问题的先验知识（类似于在最小描述长度中的描述语言）。在本节中，我们将聚焦一个最常见的正则化函数：$R(w) = \lambda \|w\|^2$，其中 $\lambda > 0$ 是一个标量而且范数是 ℓ_2 范数，$\|w\| = \sqrt{\sum_i w_i^2}$。这就产生了学习规则：

$$A(S) = \operatorname*{argmin}_{w}(L_S(w) + \lambda \|w\|^2) \tag{13.2}$$

这种形式的正则化函数通常叫做 Tikhonov 正则化。

就像之前提到的一样，等式（13.2）可以用结构风险最小化来解释，其中 w 的范数是它的“复杂度”的一种度量。回忆在上一章中，我们介绍了有界假设类的概念。因此，我们可以定义假设类的一个序列，$\mathcal{H}_1 \subset \mathcal{H}_2 \subset \mathcal{H}_3 \subset \cdots$，其中 $\mathcal{H}_i = \{w : \|w\|_2 \le i\}$。如果每个 $\mathcal{H}_i$ 的样本复杂度依赖于 i，那么对于这个嵌套类的序列，RLM 规则类似于 SRM 规则。

正则化的一个不同的解释是稳定剂。在下一节中，我们定义稳定性的概念，并且证明稳定的学习规则不会过拟合。但是让我们首先解释对于有平方损失的线性回归的 RLM 规则。

岭回归

把有 Tikhonov 正则化的 RLM 规则用到有平方损失的线性回归中，我们得到下面的学习规则：

$$\underset{w\in\mathbb{R}^d}{\operatorname{argmin}}\left(\lambda\|w\|_2^2+\frac{1}{m}\sum_{i=1}^{m}\frac{1}{2}(\langle w,x_i\rangle-y_i)^2\right) \tag{13.3}$$

用等式(13.3)实现线性回归被称作岭回归。

为了求解等式(13.3)我们将目标函数的梯度比作零，就得到一组线性等式

$$(2\lambda mI+A)w=b$$

其中 I 是单位矩阵，A，b 在等式(9.6)中定义，即

$$A=\left(\sum_{i=1}^{m}x_ix_i^T\right)\quad 和\quad b=\sum_{i=1}^{m}y_ix_i \tag{13.4}$$

由于 A 是半正定矩阵，矩阵 $2\lambda mI+A$ 的所有特征值的边界都在 $2\lambda m$ 以下。因此，这个矩阵是可逆的，岭回归的解变成

$$w=(2\lambda mI+A)^{-1}b \tag{13.5}$$

在下一节中，我们将正式说明正则化如何让算法稳定并且抑制过拟合的发生。特别是，下一节中出现的分析(特别是推论 13.11)将会产生：

定理 13.1　令 $\mathcal{D}$ 是一个在 $\mathcal{X}\times[-1,1]$ 上的分布，其中 $\mathcal{X}=\{x\in\mathbb{R}^d:\|x\|\leq 1\}$。令 $\mathcal{H}=\{w\in\mathbb{R}^d:\|w\|\leq B\}$。对于任何 $\varepsilon\in(0,1)$，令 $m\geq 150B^2/\varepsilon^2$。那么，用参数为 $\lambda=\varepsilon/(3B^2)$ 的岭回归算法满足

$$\mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}[L_{\mathcal{D}}(A(S))]\leq\min_{w\in\mathcal{H}}L_{\mathcal{D}}(w)+\varepsilon$$

评注　上面的定理告诉我们要保证学习到的预测器的风险的平均值以类的近似误差加上 ε 为边界，需要的样本个数。在通常的不可知 PAC 可学习的定义中，我们需要学习到的预测器的风险以至少 $1-\delta$ 的概率边界化。在练习 13.1 中，我们表明有边界的期望风险的算法怎么用于构建一个不可知 PAC 可学习器。

13.2　稳定规则不会过拟合

直观上来看，如果算法的输入的一个小的变化不会太多地改变算法的输出，这个算法就是稳定的。当然，有许多方法来定义我们所说的“输入的一个小的变化”和“不会太多地改变输出”。在这一节中，我们定义稳定性的一个具体的概念，并证明在这个定义下稳定规则不会过拟合。

令 A 是我们的学习算法，$S=(z_1,\ldots,z_m)$ 是 m 个样本的训练集合，$A(S)$ 表示 A 的输出。如果输出的真实风险 $L_{\mathcal{D}}(A(S))$ 和输出的经验风险 $L_S(A(S))$ 之间的差别很大，这个算法 A 就是过拟合的，就像评注指到的一样。这一章中我们集中于量的期望值(关于 S 的选择)，即 $\mathbb{E}_S(L_{\mathcal{D}}(A(S))-L_S(A(S)))$。

下面我们定义稳定性的概念。已知一个训练集合 S 和一个附加的样本 z'，令 $S^{(i)}$ 表示用 z' 代替 S 中的第 i 个样本得到的训练集：即 $S^{(i)}=(z_1,\ldots,z_{i-1},z',z_{i+1},\ldots,z_m)$。在我们对稳定性的定义中，“输入的一个小的变化”指的是将算法 A 用到 $S^{(i)}$(代替 S)中。也就是说，我们仅仅替代了一个训练样本。我们通过比较假设 $A(S)$ 在 z_i 上的损失与假设

[138]

$A(S^{(i)})$在z_i上的损失，来描述算法A的输入的小的变化对于输出的影响。直观地说，一个好的学习算法有$\ell(A(S^{(i)}),z_i)-\ell(A(S),z_i)\geq 0$，因为第一项中学习算法没有观察到样本$z_i$但是在第二项中观察到样本$z_i$。如果上面的差别是很大的，我们怀疑学习算法可能过拟合。这是因为如果在训练集中观察到了这个样本，学习算法会大幅度地改变它的预测。这些可以用下面的定理形式化表示。

定理 13.2　令$\mathcal{D}$是一个分布。令$S=(z_1,\ldots,z_m)$是一个独立同分布的样本的序列，z'是另一个独立同分布的样本。令$U(m)$是一个在$[m]$上的均匀分布。那么，对于任何学习算法，

$$\mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}\left[L_{\mathcal{D}}(A(S))-L_S(A(S))\right]=\mathop{\mathbb{E}}_{(S,z')\sim\mathcal{D}^{m+1},i\sim U(m)}\left[\ell(A(S^{(i)}),z_i)-\ell(A(S),z_i)\right]\tag{13.6}$$

证明　由于S和z'都是从$\mathcal{D}$中得到的独立同分布的样本集或样本，对于所有的i，

[139]

$$\mathop{\mathbb{E}}_{S}\left[L_{\mathcal{D}}(A(S))\right]=\mathop{\mathbb{E}}_{S,z'}\left[\ell(A(S),z')\right]=\mathop{\mathbb{E}}_{S,z'}\left[\ell(A(S^{(i)}),z_i)\right]$$

另一方面，我们可以写下

$$\mathop{\mathbb{E}}_{S}\left[L_S(A(S))\right]=\mathop{\mathbb{E}}_{S,i}\left[\ell(A(S),z_i)\right]$$

结合这两个等式，我们可以推出结论。■

当等式(13.6)右边是非常小的时候，我们说A是稳定的算法——训练集中改变一个样本不会引起很大的变化。正式表示如下。

定义 13.3(on average replace one stable)　令$\varepsilon:\mathbb{N}\to\mathbb{R}$是一个单调递减函数。我们说如果对于所有的分布$\mathcal{D}$下式成立，一个学习算法$A$就是在比率$\varepsilon(m)$下的 on-average-replace one stable：

$$\mathop{\mathbb{E}}_{(S,z')\sim\mathcal{D}^{m+1},i\sim U(m)}\left[\ell(A(S^{(i)}),z_i)-\ell(A(S),z_i)\right]\leq\varepsilon(m)$$

定理 13.2 告诉我们当且仅当一个算法是 on-average-replace one stable，它就不会过拟合。当然，一个不会过拟合的学习算法也不一定是一个好的学习算法——比如说一个总是输出相同假设的算法A。一个有用的算法应该找到一个既适合训练集(也就是有一个低的经验风险)又不过拟合的假设。或者，根据定理 13.2，算法应该在适合训练集的同时，也是稳定的。正如我们将看到的一样，RLM 规则中的参数λ平衡了适合训练集与稳定性。

13.3　Tikhonov 正则化作为稳定剂

在上一节中，我们看到了稳定规则不会过拟合。在这一节中，我们说明用有 Tikhonov 正则化$\lambda\|w\|^2$的 RLM 规则可以得到一个稳定的算法。我们假设损失函数是凸的，而且它是利普希茨的或是光滑的。

我们所依赖的 Tikhonov 正则化的主要性质是它能够让 RLM 的目标函数是强凸的，下面会给出定义。

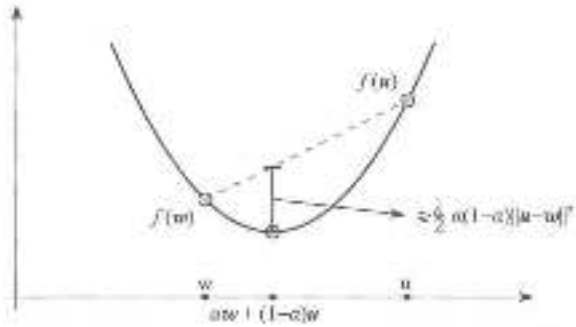
定义 13.4(强凸函数)　如果对于所有的w，u和$\alpha\in(0,1)$都有下列不等式成立，我们就说这个函数f是λ-强凸的：

[140]

$$f(\alpha w+(1-\alpha)u)\leq\alpha f(w)+(1-\alpha)f(u)-\frac{\lambda}{2}\alpha(1-\alpha)\|w-u\|^2$$

显而易见，每个凸函数都是 0-强凸的。强凸的一个说明如下图所示。

下面的引理说明 RLM 的目标函数是2λ-强凸的。另外，它强调了强凸的一个重要性质。



引理 13.5

1. 函数 $f(w)=\lambda\|w\|^2$ 是 2λ-强凸的。
2. 如果 f 是 λ-强凸的而且 g 是凸的，那么 $f+g$ 是 λ-强凸的。
3. 如果 f 是 λ-强凸的而且 u 是 f 的一个极小值，那么，对于任何 w，

$$f(w)-f(u)\geq\frac{\lambda}{2}\|w-u\|^2$$

证明 前两点可以直接从定义推导得到。为了证明最后一点，我们将强凸的定义除以 α 并且调换每项的位置，可以得到

$$\frac{f(u+\alpha(w-u))-f(u)}{\alpha}\leq f(w)-f(u)-\frac{\lambda}{2}(1-\alpha)\|w-u\|^2$$

取极限 $\alpha\to 0$，不等式右边收敛到 $f(w)-f(u)-\frac{\lambda}{2}\|w-u\|^2$。另一方面，不等式左边变成函数 $g(\alpha)=f(u+\alpha(w-u))$ 在 $\alpha=0$ 处的导数。由于 u 是 f 的一个极小值，$\alpha=0$ 就是 g 的一个极小值。因此，在极限 $\alpha\to 0$ 时，不等式的左边趋近于 0，这就结束了我们的证明。■

现在我们转向去证明 RLM 是稳定的。令 $S=(z_1,\ldots,z_m)$ 是一个训练集，z' 是一个额外的样本，$S^{(i)}=(z_1,\ldots,z_{i-1},z',z_{i+1},\ldots,z_m)$。令 A 是 RLM 规则，即

$$A(S)=\underset{w}{\operatorname{argmin}}(L_S(w)+\lambda\|w\|^2)$$

令 $f_S(w)=L_S(w)+\lambda\|w\|^2$，而且基于引理 13.5 我们知道 f_S 是 (2λ)-强凸的。按照引理的第三部分，对于任何 v，

$$f_S(v)-f_S(A(S))\geq\lambda\|v-A(S)\|^2 \tag{13.7}$$

141

另一方面，对于任何 v 和 u，对于所有的 i，我们有

$$\begin{aligned}f_S(v)-f_S(u)&=L_S(v)+\lambda\|v\|^2-(L_S(u)+\lambda\|u\|^2)\\&=L_{S^{(i)}}(v)+\lambda\|v\|^2-(L_{S^{(i)}}(u)+\lambda\|u\|^2)\\&\quad+\frac{\ell(v,z_i)-\ell(u,z_i)}{m}+\frac{\ell(u,z')-\ell(v,z')}{m}\end{aligned} \tag{13.8}$$

特别地，选择 $v=A(S^{(i)})$，$u=A(S)$，并且用 v 最小化 $L_{S^{(i)}}(w)+\lambda\|w\|^2$ 的事实，我们得到

$$\begin{aligned}f_S(A(S^{(i)}))-f_S(A(S))&\leq\frac{\ell(A(S^{(i)}),z_i)-\ell(A(S),z_i)}{m}\\&\quad+\frac{\ell(A(S),z')-\ell(A(S^{(i)}),z')}{m}\end{aligned} \tag{13.9}$$

结合它和等式(13.7)我们得到

$$\lambda \|A(S^{(i)})-A(S)\|^2 \leqslant \frac{\ell(A(S^{(i)}),z_i)-\ell(A(S),z_i)}{m} + \frac{\ell(A(S),z')-\ell(A(S^{(i)}),z')}{m} \tag{13.10}$$

下面的两节继续进行利普希茨或者光滑损失函数的稳定性分析。对于这两个损失函数的族，我们表明 RLM 是稳定的，所以它不会过拟合。

13.3.1　利普希茨损失

如果损失函数 $\ell(\cdot,z_i)$ 是 ρ-利普希茨的，那么根据利普希茨的定义，

$$\ell(A(S^{(i)}),z_i)-\ell(A(S),z_i) \leqslant \rho\|A(S^{(i)})-A(S)\| \tag{13.11}$$

相似地，

$$\ell(A(S),z')-\ell(A(S^{(i)}),z') \leqslant \rho\|A(S^{(i)})-A(S)\|$$

把这些不等式代入到等式(13.10)中得到

$$\lambda \|A(S^{(i)})-A(S)\|^2 \leqslant \frac{2\rho\|A(S^{(i)})-A(S)\|}{m}$$

上式可以变成

$$\|A(S^{(i)})-A(S)\| \leqslant \frac{2\rho}{\lambda m}$$

把上面的不等式再代入不等式(13.11)中，我们最终可以得到

$$\ell(A(S^{(i)}),z_i)-\ell(A(S),z_i) \leqslant \frac{2\rho^2}{\lambda m}$$

由于它对于任何 S，z'，i 都成立，我们可以得到：

推论 13.6　假设损失函数是凸的和 ρ-利普希茨的。那么，正则化项为 $\lambda\|w\|^2$ 的 RLM 规则是比率为 $\frac{2\rho^2}{\lambda m}$ 的 on-average-replace-one-stable。

142

因此(用定理 13.2)

$$\mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}\left[L_{\mathcal{D}}(A(S))-L_S(A(S))\right] \leqslant \frac{2\rho^2}{\lambda m}$$

13.3.2　光滑和非负损失

如果损失是 β-光滑的和非负的，那么它也是自有界的(参考 12.1 节)：

$$\|\nabla f(w)\|^2 \leqslant 2\beta f(w) \tag{13.12}$$

我们进一步假设 $\lambda \geqslant \frac{2\beta}{m}$，也就是 $\beta \leqslant \lambda m/2$。根据光滑性的假设，我们有

$$\ell(A(S^{(i)}),z_i)-\ell(A(S),z_i) \leqslant \langle \nabla\ell(A(S),z_i),A(S^{(i)})-A(S)\rangle + \frac{\beta}{2}\|A(S^{(i)})-A(S)\|^2 \tag{13.13}$$

用柯西-施瓦茨不等式和式(12.5)，我们可以进一步得到

$$\begin{aligned}\ell(A(S^{(i)}),z_i)-\ell(A(S),z_i) &\leqslant \|\nabla\ell(A(S),z_i)\|\|A(S^{(i)})-A(S)\| + \frac{\beta}{2}\|A(S^{(i)})-A(S)\|^2 \\ &\leqslant \sqrt{2\beta\ell(A(S),z_i)}\|A(S^{(i)})-A(S)\| \\ &\quad + \frac{\beta}{2}\|A(S^{(i)})-A(S)\|^2\end{aligned} \tag{13.14}$$

用对称的说法，它也满足

$$\ell(A(S),z')-\ell(A(S^{(i)}),z')\leqslant\sqrt{2\beta\ell(A(S^{(i)}),z')}\,\|A(S^{(i)})-A(S)\|-\frac{\beta}{2}\|A(S^{(i)})-A(S)\|^2$$

把这些不等式带入式子(13.10)中，并且改变每一项的位置，我们可以得到

$$\|A(S^{(i)})-A(S)\|\leqslant\frac{\sqrt{2\beta}}{(\lambda m-\beta)}\left(\sqrt{\ell(A(S),z_i)}+\sqrt{\ell(A(S^{(i)}),z')}\right)$$

结合上式和假设 $\beta\leqslant\lambda m/2$，可以得到

$$\|A(S^{(i)})-A(S)\|\leqslant\frac{\sqrt{8\beta}}{\lambda m}\left(\sqrt{\ell(A(S),z_i)}-\sqrt{\ell(A(S^{(i)}),z')}\right)$$ [143]

结合上式和式(13.14)，并且再次用假设 $\beta\leqslant\lambda m/2$，可以得到

$$\begin{aligned}&\ell(A(S^{(i)}),z_i)-\ell(A(S),z_i)\\&\leqslant\sqrt{2\beta\ell(A(S),z_i)}\,\|A(S^{(i)})-A(S)\|+\frac{\beta}{2}\|A(S^{(i)})-A(S)\|^2\\&\leqslant\left(\frac{4\beta}{\lambda m}-\frac{8\beta^2}{(\lambda m)^2}\right)\left(\sqrt{\ell(A(S),z_i)}-\sqrt{\ell(A(S^{(i)}),z')}\right)^2\\&\leqslant\frac{8\beta}{\lambda m}\left(\sqrt{\ell(A(S),z_i)}+\sqrt{\ell(A(S^{(i)}),z')}\right)^2\\&\leqslant\frac{24\beta}{\lambda m}\left(\ell(A(S),z_i)+\ell(A(S^{(i)}),z')\right)\end{aligned}$$

其中，在最后一步中，我们用到了不等式 $(a+b)^2\leqslant 3(a^2+b^2)$。关于 S，z'，i 取期望，并且注意到 $\mathbb{E}[\ell(A(S),z_i)]=\mathbb{E}[\ell(A(S^{(i)}),z')]=\mathbb{E}[L_S(A(S))]$，我们可得以下推论：

推论 13.7　假设损失函数是 β-光滑的和非负的。那么，正则化项为 $\lambda\|w\|^2$ 的 RLM 规则满足下式成立，其中 $\lambda\geqslant\frac{2\beta}{m}$：

$$\mathbb{E}[\ell(A(S^{(i)}),z_i)-\ell(A(S),z_i)]\leqslant\frac{48\beta}{\lambda m}\mathbb{E}[L_S(A(S))]$$

注意如果对于所有的 z，对于一些标量 $C>0$，我们都有 $\ell(\mathbf{0},z)\leqslant C$，那么对于所有 S，

$$L_S(A(S))\leqslant L_S(A(S))+\lambda\|A(S)\|^2\leqslant L_S(\mathbf{0})+\lambda\|\mathbf{0}\|^2=L_S(\mathbf{0})\leqslant C$$

因此，推论 13.7 也意味着

$$\mathbb{E}[\ell(A(S^{(i)}),z_i)-\ell(A(S),z_i)]\leqslant\frac{48\beta C}{\lambda m}$$

13.4　控制适合与稳定性的权衡

我们重写一个学习算法的期望风险如下所示：

$$\mathbb{E}_S[L_D(A(S))]=\mathbb{E}_S[L_S(A(S))]+\mathbb{E}_S[L_D(A(S))-L_S(A(S))]\qquad(13.15)$$

第一项反映了 $A(S)$ 适合训练数据的程度，第二项反映了 $A(S)$ 的真实风险与经验风险之间的差别。就像我们在定理 13.2 中说明的一样，第二项等价于 A 的稳定性。由于目标是最小化算法的风险，我们需要两项的和是小的。

在上一节中，我们给稳定性项加了边界。我们已经说明随着正则化参数 λ 的增加，稳定性项减少。另一方面，经验风险随着 λ 的增加而增加。因此，我们面临着适合与过拟合之间的权衡。这个权衡与本书中之前讨论的偏差-复杂度权衡非常相似。 [144]

现在我们推导 RLM 规则的经验风险项的边界。回想 RLM 规则定义为 $A(S)=\underset{w}{\operatorname{argmin}}(L_S(w)+\lambda\|w\|^2)$。固定任意向量 w^*，我们有

$$L_S(A(S))\leq L_S(A(S))+\lambda\|A(S)\|^2\leq L_S(w^*)+\lambda\|w^*\|^2$$

将式子的两边都关于 S 取期望，并且注意到 $\mathbb{E}_S[L_S(w^*)]=L_{\mathcal{D}}(w^*)$，我们得到

$$\mathop{\mathbb{E}}_{S}[L_S(A(S))]\leq L_{\mathcal{D}}(w^*)+\lambda\|w^*\|^2 \tag{13.16}$$

将它带入到式(13.15)可以得到

$$\mathop{\mathbb{E}}_{S}[L_{\mathcal{D}}(A(S))]\leq L_{\mathcal{D}}(w^*)+\lambda\|w^*\|^2+\mathop{\mathbb{E}}_{S}[L_{\mathcal{D}}(A(S))-L_S(A(S))]$$

结合上式和推论 13.6，我们可以下推论：

推论 13.8　*假设损失函数是凸的和 ρ-利普希茨的。那么，正则化项为 $\lambda\|w\|^2$ 的 RLM 规则满足：*

$$\forall\ w^*,\ \mathop{\mathbb{E}}_{S}[L_{\mathcal{D}}(A(S))]\leq L_{\mathcal{D}}(w^*)+\lambda\|w^*\|^2+\frac{2\rho^2}{\lambda m}$$

这个边界通常被称作种瑞不等式——如果我们把 w^* 看作是低风险的假设，这个边界告诉我们需要多少样本就可以实现 $A(S)$ 和 w^* 几乎一样，如果我们知道 w^* 的范数。然而实际上，我们通常不知道 w^* 的范数。因此，我们经常像第 11 章描述的一样，在一个验证集的基础上调整 λ。

对于凸利普希茨有界学习问题，我们也可以从推论 13.8 中导出一个 PAC 类似的保证[⊖]：

推论 13.9　*令 $(\mathcal{H},Z,\ell)$ 是一个参数为 ρ，B 的凸利普希茨有界学习问题。对于任何训练集的大小 m，令 $\lambda=\sqrt{\frac{2\rho^2}{B^2m}}$。那么，正则化项为 $\lambda\|w\|^2$ 的 RLM 规则满足：*

$$\mathop{\mathbb{E}}_{S}[L_{\mathcal{D}}(A(S))]\leq\min_{w\in\mathcal{H}}L_{\mathcal{D}}(w)+\rho B\sqrt{\frac{8}{m}}$$

特别地，对于所有的 $\varepsilon>0$，如果 $m\geq\frac{8\rho^2B^2}{\varepsilon^2}$，那么对于所有的分布 $\mathcal{D}$，$\mathbb{E}_S[L_{\mathcal{D}}(A(S))]\leq\min_{w\in\mathcal{H}}L_{\mathcal{D}}(w)+\varepsilon$。

上面的推论对于利普希茨损失函数成立。如果替换成光滑的和非负的损失函数，那么我们可以结合式(13.16)和推论 13.7 得到：

推论 13.10　*假设损失函数是凸的、β-光滑的和非负的。那么，对于所有 w^*，正则化项为 $\lambda\|w\|^2$ 的 RLM 规则满足下式成立，其中 $\lambda\geq\frac{2\beta}{m}$：*

145

$$\mathop{\mathbb{E}}_{S}[L_{\mathcal{D}}(A(S))]\leq\left(2+\frac{48\beta}{\lambda m}\right)\mathop{\mathbb{E}}_{S}[L_S(A(S))]\leq\left(1+\frac{48\beta}{\lambda m}\right)(L_{\mathcal{D}}(w^*)+\lambda\|w^*\|^2)$$

比如，如果我们选择 $\lambda=\frac{48\beta}{m}$，可以从上式得到 $A(S)$ 的期望真实风险接近于 $A(S)$ 的期望经验风险的两倍。而且，对于 λ 的这个取值，$A(S)$ 的期望经验风险接近于 $L_{\mathcal{D}}(w^*)+\frac{48\beta}{m}\|w^*\|^2$。

⊖　此外，下面的边界是关于期望风险的，但是用练习 13.1，它可以用来推导一个不可知 PAC 可学习的保证。

对于凸光滑有界学习问题，我们也可以从推论 13.10 中导出一个可学习的保证。

推论 13.11　令$(\mathcal{H}, Z, \ell)$是一个参数为β，B的凸光滑有界学习问题。另外假设对于所有的$z\in Z$，$\ell(\mathbf{0}, z)\leq 1$。对于任何$\varepsilon\in(0, 1)$，令$m\geq\frac{150\beta B^2}{\varepsilon^2}$，并且设$\lambda=\varepsilon/(3B^2)$。那么，对于所有的分布$\mathcal{D}$，

$$\mathop{\mathbb{E}}_{S}[L_{\mathcal{D}}(A(S))]\leq\min_{w\in\mathcal{H}}L_{\mathcal{D}}(w)+\varepsilon$$

13.5　小结

我们介绍了稳定性，并且说明如果一个算法是稳定的，那么它就不会过拟合。而且，对于凸利普希茨边界和凸光滑边界问题，有 Tikhonov 正则化的 RLM 规则生成一个稳定的学习问题。我们讨论了正则化参数λ如何控制适合与过拟合之间的权衡。最终，我们表明，用 RLM 规则的所有来自凸利普希茨有界和凸光滑有界族的学习问题都是可学习的。RLM 范式是许多流行学习算法的基础，包括岭回归(在本章中讨论过)和支持向量机(将会在第 15 章讨论)。

下一章中，我们将介绍随机梯度下降，它为我们学习凸利普希茨有界和凸光滑有界问题提供了一个可供选择的方法，而且还可以用来有效地实现 RLM 规则。

13.6　文献评注

稳定性被广泛地用于许多数学环境下。比如，对于所谓逆问题应该很好地提出稳定的必要性第一次被 Hadamard(1902)认识到。正则化的思想和它与稳定性的关系通过 Tikhonov(1943)和 Phillips(1962)的工作变得为大家所熟知。在现代学习理论的内容中，稳定性的应用可以至少追溯到 Rogers 和 Wager(1978)的工作，他们注意到一个学习算法关于样本中的小的变化的敏感性控制了留一估计的方差。作者用这个观察得到了k-邻近算法的泛化边界(参考第 19 章)。这些结果后来扩展到了其他"局部的"学习算法(参考 Devroye，Györfi 和 Lugosi(1996)以及其中的引用)。另外，已经发展出实际的方法可以把稳定性引入学习算法，特别是 Breiman(1996)介绍的 Bagging 技术。 146

在过去的十年中，稳定性被当做一个可学习的一般条件来研究。参考 Kearns & Ron(1999)，Bousquet & Elisseeff(2002)，Kutin & Niyogi(2002)，Rakhlin、Mukherjee & Poggio(2005)，Mukherjee、Niyogi、Poggio & Rifkin(2006)。我们的介绍跟随了 Shalev-Shwartz，Shamir，Srebo 和 Sridharan (2010)的工作，他们说明了稳定性是可学习的充要条件。他们也说明用 RLM 规则，所有的凸利普希茨有界学习问题都是可学习的，即使在强的语文下一些凸利普希茨有界学习问题的一致收敛不满足。

13.7　练习

13.1　从有界期望风险到不可知 PAC 可学习：令A是可以保证下面的条件成立的一个算法：如果$m\geq m_{\mathcal{H}}(\varepsilon)$，那么对于所有分布$\mathcal{D}$都满足

$$\mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}[L_{\mathcal{D}}(A(S))]\leq\min_{h\in\mathcal{H}}L_{\mathcal{D}}(h)+\varepsilon$$

- 说明对于所有$\delta\in(0, 1)$，如果$m\geq m_{\mathcal{H}}(\varepsilon\delta)$，那么至少在概率$1-\delta$下，$L_{\mathcal{D}}(A(S))\leq\min_{h\in\mathcal{H}}L_{\mathcal{D}}(h)+\varepsilon$成立。

提示：观察随机变量$L_{\mathcal{D}}(A(S))-\min_{h\in\mathcal{H}}L_{\mathcal{D}}(h)$是非负并且依赖马尔可夫不等式。

- 对于所有的$\delta\in(0,1)$，令

$$m_{\mathcal{H}}(\varepsilon,\delta)=m_{\mathcal{H}}(\varepsilon/2)\lceil\log_2(1/\delta)\rceil+\left\lceil\frac{\log(4/\delta)+\log(\lceil\log_2(1/\delta)\rceil)}{\varepsilon^2}\right\rceil$$

提出一个样本复杂度为$m_{\mathcal{H}}(\varepsilon,\delta)$的不可知PAC学习这个问题的步骤，假设损失函数的界限为1。

提示：令$k=\lceil\log_2(1/\delta)\rceil$。把数据分成$k+1$组，其中每个前$k$组的样本大小为$m_{\mathcal{H}}(\varepsilon/2)$。用$A$训练前$k$个组。在上一个问题的基础上，讨论对于所有组都有$L_{\mathcal{D}}(A(S))>\min_{h\in\mathcal{H}}L_{\mathcal{D}}(h)+\varepsilon$成立的概率最多为$2^{-k}\leq\delta/2$。最终，用最后一组作为验证集。

13.2 **没有一致收敛性的可学习**：令$\mathcal{B}$是一个取值为$\mathbb{R}^d$的单位球。令$\mathcal{H}=\mathcal{B}$，$Z=\mathcal{B}\times\{0,1\}^d$，而且令$\ell:Z\times\mathcal{H}\to\mathbb{R}$定义如下：

$$\ell(w,(x,\boldsymbol{\alpha}))=\sum_{i=1}^{d}\alpha_i(x_i-w_i)^2$$

这个问题相当于一个非监督学习任务，意味着我们不去预测x的标签。相反，我们要做的是找到在$\mathcal{B}$上分布的"团的中心"。但是，有一个用向量$\boldsymbol{\alpha}$建模的扭曲：每个样本都是一个$(x,\boldsymbol{\alpha})$对，其中x是实例x，$\boldsymbol{\alpha}$表明x中的哪些特征是"激活的"，哪些是"关掉的"。一个假设就是一个表示分布的团的中心的向量w，而且损失函数是x与w之间的平方欧式距离，不过仅仅关于x中"激活的"元素。

147

- 说明用RLM规则，这个问题是可学习的，而且样本复杂度不依赖于d。
- 考虑一个Z上的分布$\mathcal{D}$如下：x固定成一些x_0，$\boldsymbol{\alpha}$的每一个元素以相等的概率采样为0或者1。说明这个问题的一致收敛的比率随着d增长。

  提示：令m是一个训练集的大小。说明如果$d\gg2^m$，就有很大的概率采样一个样本集使得存在一些$j\in[d]$对于训练集中所有的样本$\alpha_j=1$。说明这样的一个样本不是ε-可表示的。得出一致收敛的样本复杂度一定随着$\log(d)$增长的结论。
- 推导结论，如果我们把d取为无限大，就可以得到一个可学习的但是一致收敛性不成立的问题。比较它和统计学习的基本理论。

13.3 **稳定性和渐近的EPM足以满足可学习的条件**：如果对于分布$\mathcal{D}$，下式成立，我们就说一个学习规则A是在比率$\varepsilon(m)$下的渐近的经验风险最小化(AERM)：

$$\mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}\left[L_S(A(S))-\min_{h\in\mathcal{H}}L_S(h)\right]\leq\varepsilon(m)$$

如果对于分布$\mathcal{D}$，下式成立，我们就说一个学习规则A是在比率$\varepsilon(m)$下学习了一个类$\mathcal{H}$：

$$\mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}\left[L_{\mathcal{D}}(A(S))-\min_{h\in\mathcal{H}}L_{\mathcal{D}}(h)\right]\leq\varepsilon(m)$$

证明下述定理：

定理13.12 如果一个学习算法A是在比率$\varepsilon_1(m)$下的on-average-replace-one-stable，也是在比率$\varepsilon_2(m)$下的渐近经验风险最小化，那么它在比率$\varepsilon_1(m)+\varepsilon_2(m)$下学习$\mathcal{H}$。

13.4 **关于一般范数的强凸**：在这一节中我们都用ℓ_2范数。在这个习题中，我们把一些结果推广到一般范数。令$\|\cdot\|$表示任意范数，且f是一个关于这个范数的强凸函数(参考定义13.4)。

1) 说明引理13.5中的2~3项对于所有的范数都成立。

*2) 给出引理 13.5 中的第一项的范数不成立的一个例子。

3) 令 $R(w)$ 是一个关于一些范数 $\|\cdot\|$ 的 (2λ)-强凸函数，令 A 是一个关于 R 的 RLM 规则，即

$$A(S) = \operatorname*{argmin}_{w}(L_S(w) + R(w))$$

假设对于所有的 z，损失函数 $\ell(\cdot, z)$ 是关于相同范数的 ρ 利普希茨的，即

$$\forall z, \forall w, v, \quad \ell(w, z) - \ell(v, z) \leq \rho \|w - v\|$$

证明 A 是在比率 $\frac{2\rho^2}{\lambda m}$ 下的 on-average-replace-one-stable。

*4) 令 $q \in (1, 2)$ 而且考虑 ℓ_q-范数

$$\|w\|_q = \left(\sum_{i=1}^{d} |w_i|^q\right)^{1/q}$$

可以看出(比如，参考 Shalev Shwartz(2007))函数

$$R(w) = \frac{1}{2(q-1)} \|w\|_q^2$$

是关于 $\|w\|_q$ 的 1-强凸的。说明如果 $q = \frac{\log(d)}{\log(d)-1}$，那么 $R(w)$ 是关于在 $\mathbb{R}^d$ 上的 ℓ_1 范数的 $\frac{1}{3\log(d)}$-强凸的。

148
?
149

随机梯度下降

回想一下，学习的目的是极小化风险函数，$L_{\mathcal{D}}(h)=\mathbb{E}_{z\sim\mathcal{D}}[\ell(h,z)]$。由于它依赖的分布$\mathcal{D}$是未知的，所以不能直接极小化风险函数。在本书中，到目前为止我们已经讨论了依赖经验风险的学习方法。首先，我们采样一个训练集 S 并定义经验风险函数 $L_S(h)$。然后，学习者根据 $L_S(h)$ 的值选择一个假设。例如，ERM 准则告诉我们在假设类$\mathcal{H}$上选择极小化 $L_S(h)$ 的那个假设。或者像在之前的章节中，我们讨论正则化风险极小化。在正则化风险极小化中，我们选择一个联合极小化 $L_S(h)$ 和正则化函数的假设 h。

本章我们将描述并分析一个相当不同的学习方法，称之为随机梯度下降(Stochastic Gradient Descent，SGD)。如第 12 章我们关注的一类重要的凸学习问题，给定符号后，我们把假设看成是凸假设类$\mathcal{H}$中的向量 w。在 SGD 中，我们试图利用梯度下降策略去直接极小化风险函数 $L_{\mathcal{D}}(w)$。梯度下降是一个迭代优化策略，通过取沿着函数当前迭代点的负梯度方向的步长来提高解的精度。当然，在我们这种情况下，极小化风险函数，并不知道分布$\mathcal{D}$，也不知道 $L_{\mathcal{D}}(w)$ 的梯度。通过取一个随机方向的步长，SGD 可以避开这个问题，该方向的期望就是负梯度。正如我们将看到的，尽管我们不知道潜在的分布$\mathcal{D}$，但是寻找这样的随机方向(期望对应着梯度)却是比较容易的。

在凸学习的环境中，SGD 相对于正则化风险极小化学习准则的优势是它是一个有效的算法，可以仅由几行代码实现，并且和正则化风险极小化学习准则有相同的样本复杂度。这种简易性使得我们可以在不能使用基于经验风险方法的情况下使用 SGD 方法，由于这部分内容超出了本书讨论的范围，在此不做过多描述。

下面，我们先介绍基本的梯度下降算法并分析它求解凸利普希茨函数的收敛速度。然后介绍次梯度的符号并说明梯度下降也可以用于不可微函数。本章的核心是 14.3 节，在这一节中我们将描述随机梯度下降算法，以及它的一些变型，并说明 SGD 的期望收敛速度和梯度下降的收敛速度相似。最后，我们探讨 SGD 求解学习问题的能力。

14.1 梯度下降法

在描述随机梯度下降方法之前，我们先介绍极小化可微凸函数 $f(w)$ 的标准梯度下降方法。

可微函数 $f:\mathbb{R}^d\to\mathbb{R}$ 在 w 处的梯度是 f 的偏导数，记作$\nabla f(w)$，$\nabla f(w)=\left(\frac{\partial f(w)}{\partial w[1]},\ldots,\frac{\partial f(w)}{\partial w[d]}\right)$。梯度下降是一个迭代算法。给定 w 的初始点($w^{(1)}=0$)，然后在每次迭代的时候，沿着当前迭代点的负梯度方向取步长，步长更新如下

$$w^{(t+1)}=w^{(t)}-\eta\nabla f(w^{(t)}) \tag{14.1}$$

其中 $\eta>0$，稍后将对 η 做讨论。直观地说，梯度点是函数 f 在$w^{(t)}$附近上升速度最快的方向，算法取的是反方向上的一个小步长，因此可以降低函数值。最后，经过 T 次迭代，算法输出一个平均值 $\bar{w}=\frac{1}{T}\sum_{t=1}^{T}w(t)$。这个输出也可能是最后一个向量$w^{(T)}$或者是性能最好

的向量 $\arg\min_{t\in[T]} f(w^{(t)})$，但是取平均确实是有用的，特别是当我们把梯度下降推广到不可微函数和随机的情形。

另一种激励梯度下降方法的方式是依模泰勒近似。由 f 在 w 的梯度可以得到 f 在 w 附近的一阶泰勒近似 $f(u)\approx f(w)+\langle u-w, \nabla f(w)\rangle$。当 f 为凸时，这个近似可以给出 f 的下界，即

$$f(u)\geqslant f(w)+\langle u-w, \nabla f(w)\rangle$$

所以，对靠近 $w^{(t)}$ 的 w 我们有 $f(w)\approx f(w^{(t)})+\langle w-w^{(t)}, \nabla f(w^{(t)})\rangle$。因此，我们可以极小化 $f(w)$ 的近似，不过，这个近似对于离 $w^{(t)}$ 很远的 w 可能会失效。所以，我们打算联合极小化 w 与 $w^{(t)}$ 之间的距离和 f 在 $w^{(t)}$ 附近的近似。如果参数 η 控制着两项之间的权衡，那么我们有下面的更新规则

$$w^{(t+1)}=\arg\min_{w}\frac{1}{2}\|w-w^{(t)}\|^2+\eta(f(w^{(t)})+\langle w-w^{(t)}, \nabla f(w^{(t)})\rangle)$$

对 w 求导并令结果等于 0 可以得到和(14.1)式一样的结果。

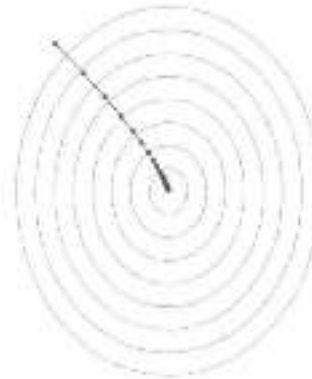


图 14.1　梯度下降法的一个图解。求解的函数是 $1.25(x_1+6)^2+(x_2-8)^2$

[151]

梯度下降法求解凸利普希茨函数的分析

为了分析梯度法的收敛速度，在这里只考虑凸利普希茨函数的情形(正如我们所看到的，许多问题可以很容易表达成这样的形式)。设 w^* 是任一向量，B 是 $\|w^*\|$ 的一个上界。很容易想到让 w^* 作为 $f(w)$ 的最小值，但是下面的分析适用于每个 w^*。

关于 w^*，我们想要得到关于解的次优性的一个上界，即，$f(\overline{w})-f(w^*)$，其中 $\overline{w}=\frac{1}{T}\sum_{t=1}^{T}w(t)$。由 $\overline{w}$ 的定义和詹生不等式，得

$$\begin{aligned}f(\overline{w})-f(w^*)&=f\Big(\frac{1}{T}\sum_{t=1}^{T}w^{(t)}\Big)-f(w^*)\\&\leqslant\frac{1}{T}\sum_{t=1}^{T}(f(w^{(t)}))-f(w^*)\\&=\frac{1}{T}\sum_{t=1}^{T}(f(w^{(t)}))-f(w^*)\end{aligned}\tag{14.2}$$

对每个 t，由于 f 的凸性，我们有

$$f(w^{(t)})-f(w^*)\leqslant\langle w^{(t)}-w^*, \nabla f(w^{(t)})\rangle\tag{14.3}$$

结合前面得

$$f(w)-f(w^*)\leqslant\frac{1}{T}\sum_{t=1}^{T}\langle w^{(t)}-w^*, \nabla f(w^{(t)})\rangle$$

为了给出右端的界，我们先给出下面的引理：

引理 14.1　设 $v_1, \cdots, v_T$ 是任一的向量序列，任意一个初始点为 $w^{(1)}=0$，迭代准则为

$$w^{(t+1)}=w^{(t)}-\eta v_t\tag{14.4}$$

的算法满足 [152]

$$\sum_{t=1}^{T}\langle w^{(t)}-w^*,v_t\rangle \leqslant \frac{\|w^*\|^2}{2\eta}+\frac{\eta}{2}\sum_{t=1}^{T}\|v_t\|^2 \tag{14.5}$$

特别地，对每个 $B,\rho>0$，如果对所有的 t 都成立 $\|v_t\|\leqslant\rho$，令 $\eta=\sqrt{\frac{B^2}{\rho^2 T}}$，那么对每个满足 $\|w^*\|\leqslant B$ 的 w^* 有

$$\frac{1}{T}\sum_{t=1}^{T}\langle w^{(t)}-w^*,v_t\rangle \leqslant \frac{B\rho}{\sqrt{T}}$$

证明　利用代数方法(完全平方)得

$$\begin{aligned}\langle w^{(t)}-w^*,v_t\rangle &= \frac{1}{\eta}\langle w^{(t)}-w^*,\eta v_t\rangle\\ &= \frac{1}{2\eta}(-\|w^{(t)}-w^*-\eta v_t\|^2+\|w^{(t)}-w^*\|^2+\eta^2\|v_t\|^2)\\ &= \frac{1}{2\eta}(-\|w^{(t+1)}-w^*\|^2+\|w^{(t)}-w^*\|^2)+\frac{\eta}{2}\|v_t\|^2\end{aligned}$$

其中最后一个不等式由更新准则的定义得到。对等式在 t 上求和，得

$$\sum_{t=1}^{T}\langle w^{(t)}-w^*,v_t\rangle=\frac{1}{2\eta}\sum_{t=1}^{T}(-\|w^{(t+1)}-w^*\|^2+\|w^{(t)}-w^*\|^2)+\frac{\eta}{2}\sum_{t=1}^{T}\|v_t\|^2 \tag{14.6}$$

右端第一个求和项是伸缩和，为

$$\|w^{(1)}-w^*\|^2-\|w^{(T+1)}-w^*\|^2$$

代入(14.6)式，得

$$\begin{aligned}\sum_{t=1}^{T}\langle w^{(t)}-w^*,v_t\rangle &= \frac{1}{2\eta}(\|w^{(1)}-w^*\|^2-\|w^{(T+1)}-w^*\|^2)+\frac{\eta}{2}\sum_{t=1}^{T}\|v_t\|^2\\ &\leqslant \frac{1}{2\eta}\|w^{(1)}-w^*\|^2+\frac{\eta}{2}\sum_{t=1}^{T}\|v_t\|^2\\ &\leqslant \frac{1}{2\eta}\|w^*\|^2+\frac{\eta}{2}\sum_{t=1}^{T}\|v_t\|^2\end{aligned}$$

其中最后一个不等式是因为 $w^{(1)}=0$。这证明了引理(式(14.5))的第一部分。通过 $\|w^*\|$ 的上界 B，$\|v_t\|$ 的上界 ρ，除以 T 并代入 η 可得第二项。 ■

将引理 14.1 应用于(GD)算法，令 $v_t=\nabla f(w^{(t)})$。在引理 14.7 中我们将说明，如果 f 是 ρ-利普希茨，那么 $\|\nabla f(w^{(t)})\|\leqslant\rho$。所以，引理的条件得到满足，并且有下面的推论：

133

推论 14.2　设 f 是一个凸 ρ-利普希茨函数，$w^*\in\operatorname*{argmin}_{w:\|w\|\leqslant B} f(w)$。如果对 f 实施 T 步 GD 算法，且令 $\eta=\sqrt{\frac{B^2}{\rho^2 T}}$，那么输出向量 $\overline{w}$ 满足

$$f(\overline{w})-f(w^*)\leqslant\frac{B\rho}{\sqrt{T}}$$

进一步，对每个 $\varepsilon>0$，为了达到 $f(\overline{w})-f(w^*)\leqslant\varepsilon$，只要运行(GD)算法多次满足 $T\geqslant\frac{B^2\rho^2}{\varepsilon^2}$ 即可。

14.2　次梯度

(GD)算法要求函数 f 是可微的。我们现在超出可微函数的范畴来进行讨论。通过使用

f 在 $w^{(t)}$ 处的次梯度代替梯度，我们将看到 GD 算法也可以应用于不可微函数。

为了给出次梯度的定义，先回顾一下对凸函数 f 而言，f 在 w 处的梯度定义了位于 f 下方的切线的斜率，即

$$\forall u, f(u) \geqslant f(w) + \langle u - w, \nabla f(w) \rangle \tag{14.7}$$

图 14.2 的左侧是关于梯度的图解说明。

对凸函数而言，位于 f 下方的切线的存在是一个很重要的性质。事实上，这也是凸性的另一种刻画。

引理 14.3　*设 S 是一个开凸集，函数 $f: S \to \mathbb{R}$ 是凸的当且仅当对每个 $w \in S$，存在 v 使得*

$$\forall u \in S, f(u) \geqslant f(w) + \langle u - w, v \rangle \tag{14.8}$$

该引理的证明可以在许多凸分析的教材中找到(如 Borwein 和 Lewis 2006)。该不等式让我们有了次梯度的定义。

定义 14.4(次梯度)　*满足(14.8)式的向量 v 称为 f 在 w 处的次梯度。f 在 w 处的次梯度的集合称为微分集，记作 $\partial f(w)$。*

图 14.2 的右侧是关于次梯度的图解说明。对于标量函数，凸函数 f 在 w 处的次梯度是与 f 在 w 相接的一条线的斜率，而不是其他在 f 之上的线。

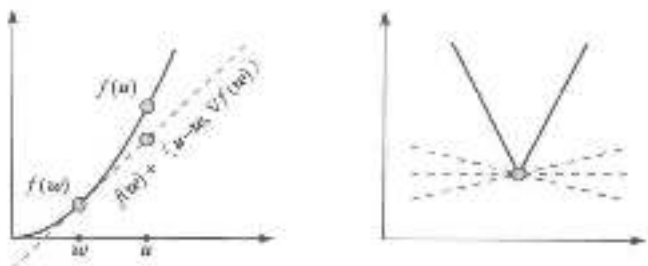


图 14.2　左图：式(14.7)的右边是 f 在 w 处的切线。对凸函数而言，这条切线是 f 的下界。右图：不可微凸函数的几个次梯度图解

14.2.1　计算次梯度

对于一个给定的函数，如何构造它的次梯度？正如下面的论断所说，如果函数在 w 处是可微的，那么微分集是平凡的。

[154]

论断 14.5　*如果 f 在 w 处可微，那么 $\partial f(w)$ 中只含有一个元素，即 f 在 w 处的梯度 $\nabla f(w)$。*

例 14.1　(绝对值函数的微分集)　考虑绝对值函数 $f(x) = |x|$。由论断 14.5 我们可以构造出 f 可微部分的微分集。只有一个点需要特别关注，即 $x_0 = 0$。在那个点，容易验证次梯度是由 -1 和 1 之间的所有数构成的集合。因此：

$$\partial f(x) = \begin{cases} \{1\} & \text{如果 } x > 0 \\ -1 & \text{如果 } x < 0 \\ [-1, 1] & \text{如果 } x = 0 \end{cases}$$

对于许多的实际应用，我们并不需要计算在给定点处的全部次梯度集，因为只要有集合里

的一个元素就足够了。下面的论断说明如何构造逐点最大函数的一个次梯度。 ◀

论断 14.6　对 r 个凸可微函数 $g_1, \cdots, g_r$，令 $g(w)=\max_{i\in[r]} g_i(w)$。给定某个 w，设 $j\in\arg\min_i g_i(w)$，那么 $\nabla g_j(w)\in\partial g(w)$。

证明　由于 g_j 是凸的，对于所有的 u 有

$$g_j(u)\geqslant g_j(w)+\langle u-w,\nabla g_j(w)\rangle$$

又 $g(w)=g_j(w)$，$g(u)\geqslant g_j(u)$，于是有

$$g(u)\geqslant g(w)+\langle u-w,\nabla g_j(w)\rangle$$

这就证明了我们结论。 ■

例 14.2（合页损失的次梯度）　回顾第 12.3 节中的合页损失函数，对于某个向量 x，标量 y，$f(w)=\max\{0,1-y\langle w,x\rangle\}$。为了计算合页损失在某个 w 处的次梯度，利用前面的论断所得到的如下定义的向量 v 是合页损失在 w 处的次梯度：

$$v=\begin{cases}\mathbf{0} & 若\quad 1-y\langle w,x\rangle\leqslant 0\\ -yx & 若\quad 1-y\langle w,x\rangle>0\end{cases}$$ ◀

155

14.2.2　利普希茨函数的次梯度

回顾一下，如果对于所有的 u，$v\in A$，成立

$$|f(u)-f(v)|\leqslant\rho|u-v|$$

那么函数 $f: A\to R$ 是 ρ-利普希茨的。下面的引理利用次梯度的范数给出了一个等价的定义。

引理 14.7　设 A 是一个开凸集，$f: A\to R$ 是一个凸函数。那么 f 在 A 上是 ρ-利普希茨当且仅当对于所有的 $w\in A$ 和 $v\in\partial f(w)$，有 $|v|\leqslant\rho$。

证明　假设对所有的 $v\in\partial f(w)$ 成立 $|v|\leqslant\rho$。由于 $v\in\partial f(w)$，所以

$$f(w)-f(u)\leqslant\langle v,w-u\rangle$$

利用柯西-施瓦茨不等式对右端取界得

$$f(w)-f(u)\leqslant\langle v,w-u\rangle\leqslant\|v\|\,\|w-u\|\leqslant\rho|w-u|$$

一个类似的观点可以表明 $f(w)-f(u)\leqslant\rho\|w-u\|$。因此，$f$ 是 ρ-利普希茨。

假设 f 是 ρ-利普希茨。选择某个 $w\in A$，$v\in\partial f(w)$。由于 A 是开集，故存在 $\varepsilon>0$ 使得 $u=w+\varepsilon v/|v|$ 属于 A。所以 $\langle u-w,v\rangle=\varepsilon|v|$，$|u-w|=\varepsilon$。由次梯度的定义得

$$f(u)-f(w)\geqslant\langle v,u-w\rangle=\varepsilon|v|$$

另一方面，由 f 的利普希茨性得

$$\rho\varepsilon=\rho|u-w|\geqslant f(u)-f(w)$$

结合这两个不等式，得 $|v|\leqslant\rho$。 ■

14.2.3　次梯度下降

利用 $f(w)$ 在 $w^{(t)}$ 处的次梯度代替梯度，可以将梯度下降法推广到不可微函数。对于次梯度收敛速度的分析仍保持不变；可以看到式(14.3)对次梯度同样成立。

14.3　随机梯度下降

在随机梯度下降中，我们不要求基于精确的梯度值来更新迭代方向，而是允许迭代方

向是一个随机向量，并且只要求在每次迭代的时候该方向的期望值和梯度方向是相等的。或者，更一般地，我们要求随机向量的期望值是函数在当前向量处的次梯度。 [156]

图 14.3 给出了随机梯度下降与梯度下降的图解比较。正如我们将在 14.5 节看到的，在学习问题的环境中，容易找到一个随机向量，该向量的期望是风险函数的次梯度。

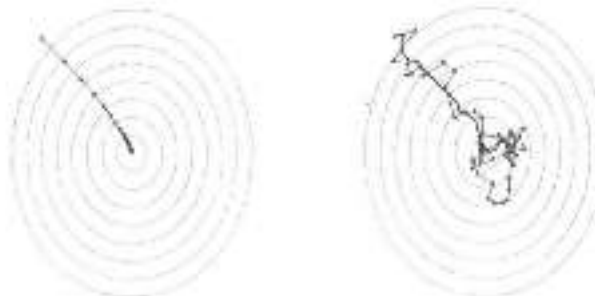


图 14.3　梯度下降法(左)和随机梯度下降法(右)的图解。极小化的函数是 $1.25(x-6)^2+(y-8)^2$。对于随机的情形，实线描述的是 w 的平均值

极小化 $f(w)$ 的随机梯度下降法(SGD)

参数：标量 $\eta>0$，整数 $T>0$

初始化：$w^{(1)}=\mathbf{0}$

for $t=1,2,\dots,T$

　　以一个分布随机选择 v_t，使得 $\mathbb{E}[v_t|w^{(t)}]\in\partial f(w^{(t)})$

　　更新 $w^{(t+1)}=w^{(t)}-\eta v_t$

输出 $\overline{w}=\frac{1}{T}\sum_{t=1}^{T}w^{(t)}$

SGD 求解凸利普希茨有界函数的分析

回顾一下推论 14.2 中得到的 GD 算法的界。对于随机的情形，只有 v_t 的期望属于 $\partial f(w^{(t)})$，故不能直接应用式(14.3)。然而，由于 v_t 的期望是 f 在 $w^{(t)}$ 的一个次梯度，所以我们还是可以得到一个类似的界，该界是关于随机梯度下降的期望输出，这可以表述为下面的定理。

定理 14.8　设 $B,\rho>0$，f 是一个凸函数，$w^*\in\arg\min_{w:\|w\|\le B}f(w)$。假设 SGD 运行 T 次，$\eta=\sqrt{\frac{B^2}{\rho^2T}}$，且对于所有的 t，以概率 1 成立 $\|v_t\|\le\rho$。那么有

$$\mathbb{E}[f(\overline{w})]-f(w^*)\le\frac{B\rho}{\sqrt{T}}$$ [157]

所以，对于任意的 $\varepsilon>0$，要达到 $\mathbb{E}[f(\overline{w})]-f(w^*)\le\varepsilon$，只要运行 SGD 的次数满足 $T\ge\frac{B^2\rho^2}{\varepsilon^2}$ 即可。

证明　首先引入一个符号 $v_{1:t}$，它表示序列 $v_1,\dots,v_t$。对式(14.2)两边取期望，得

$$\mathop{\mathbb{E}}_{v_{1:T}}[f(\bar{w})-f(w^*)]\leqslant \mathop{\mathbb{E}}_{v_{1:T}}\left[\frac{1}{T}\sum_{t=1}^{T}(f(w^{(t)}))-f(w^*)\right]$$

既然引理 14.1 对于任意的序列 $v_1, v_2, \cdots, v_T$ 都成立，那么它也可以用于 SGD。对该引理中的界取期望，得

$$\mathop{\mathbb{E}}_{v_{1:T}}\left[\frac{1}{T}\sum_{t=1}^{T}\langle w^{(t)}-w^*, v_t\rangle\right]\leqslant \frac{B\rho}{\sqrt{T}} \tag{14.9}$$

还需证明

$$\mathop{\mathbb{E}}_{v_{1:T}}\left[\frac{1}{T}\sum_{t=1}^{T}(f(w^{(t)})-f(w^*))\right]\leqslant \mathop{\mathbb{E}}_{v_{1:T}}\left[\frac{1}{T}\sum_{t=1}^{T}\langle w^{(t)}-w^*, v_t\rangle\right] \tag{14.10}$$

关于这个结论我们在此给出证明。

利用期望的线性，得

$$\mathop{\mathbb{E}}_{v_{1:T}}\left[\frac{1}{T}\sum_{t=1}^{T}\langle w^{(t)}-w^*, v_t\rangle\right]=\frac{1}{T}\sum_{t=1}^{T}\mathop{\mathbb{E}}_{v_{1:T}}[\langle w^{(t)}-w^*, v_t\rangle]$$

接下来，先回顾一下全期望法则：对每两个随机变量 α、β 和函数 g，$\mathbb{E}_\alpha[g(\alpha)]=\mathbb{E}_\beta\mathbb{E}_\alpha[g(\alpha)|\beta]$。令 $\alpha=v_{1:t}$，$\beta=v_{1:t-1}$，得

$$\begin{aligned}\mathop{\mathbb{E}}_{v_{1:T}}[\langle w^{(t)}-w^*, v_t\rangle]&=\mathop{\mathbb{E}}_{v_{1:t}}[\langle w^{(t)}-w^*, v_t\rangle]\\&=\mathop{\mathbb{E}}_{v_{1:t-1}}\mathop{\mathbb{E}}_{v_t}[\langle w^{(t)}-w^*, v_t\rangle\,|\,v_{1:t-1}]\end{aligned}$$

一旦我们知道了 $v_{1:t-1}$，$w^{(t)}$ 的值就不再是随机的了，所以

$$\mathop{\mathbb{E}}_{v_{1:t-1}}\mathop{\mathbb{E}}_{v_t}[\langle w^{(t)}-w^*, v_t\rangle\,|\,v_{1:t-1}]=\mathop{\mathbb{E}}_{v_{1:t-1}}\langle w^{(t)}-w^*, \mathop{\mathbb{E}}_{v_t}[v_t\,|\,v_{1:t-1}]\rangle$$

由于 $w^{(t)}$ 只依赖于 $v_{1:t-1}$ 且 SGD 要求 $\mathop{\mathbb{E}}_{v_t}[v_t\,|\,w^{(t)}]\in\partial f(w^{(t)})$，于是有 $\mathop{\mathbb{E}}_{v_t}[v_t\,|\,v_{1:t-1}]\in\partial f(w^{(t)})$。因此，

$$\mathop{\mathbb{E}}_{v_{1:t-1}}\langle w^{(t)}-w^*, \mathop{\mathbb{E}}_{v_t}[v_t\,|\,v_{1:t-1}]\rangle\geqslant \mathop{\mathbb{E}}_{v_{1:t-1}}[f(w^{(t)})-f(w^*)]$$

总的来说，我们证明了

[158]

$$\mathop{\mathbb{E}}_{v_{1:T}}[\langle w^{(t)}-w^*, v_t\rangle]\geqslant \mathop{\mathbb{E}}_{v_{1:t-1}}[f(w^{(t)})-f(w^*)]=\mathop{\mathbb{E}}_{v_{1:T}}[f(w^{(t)})-f(w^*)]$$

对 t 求和，除以 T，再使用期望的线性，可知式(14.10)成立。

14.4　SGD 的变型

在这一节中我们介绍几个随机梯度下降的几种变型。

14.4.1　增加一个投影步

前面对 GD 和 SGD 的分析中要求 w^* 的范数至多为 B，也就是要求 w^* 属于集合 $\mathcal{H}=\{w: \|w\|\leqslant B\}$。从学习的角度说，这意味着我们要将考虑的范围限制到一个以 B 为界的假设类中。然而，在与梯度相反的方向(或者它的期望方向)上取的每一个步长都可能导致走出这个界，甚至不能保证 $\bar{w}$ 满足这个条件。下面，我们说明如何在保持相同收敛速度的同时克服这一问题。

基本的想法是增加一个投影步：也就是说，我们采用一个两步更新准则，首先减去当前 w 处的值的次梯度，然后将得到的向量投影到 $\mathcal{H}$ 上，形式上可表示为：

1. $w^{(t+\frac{1}{2})}=w^{(t)}-\eta v_t$

2. $w^{(t+1)}=\operatorname{argmin}_{w\in\mathcal{H}}\|w-w^{(t+\frac{1}{2})}\|$

这个投影步通过$\mathcal{H}$中与 w 最近的那个向量来替代当前的w。

显然，投影步保证了对于所有的 t 都有$w^{(t)}\in\mathcal{H}$。由于$\mathcal{H}$是凸的，这也意味着 $\overline{w}\in\mathcal{H}$。下面，我们说明对具有投影兼略的 SGD 的分析依然不变。该分析基于下面的引理。

引理 14.9(投影引理) 设$\mathcal{H}$是一个闭凸集，v 是 w 在$\mathcal{H}$上的投影，即

$$v=\operatorname*{argmin}_{x\in\mathcal{H}}\|x-w\|^2$$

然后，对每个 $u\in\mathcal{H}$，

$$\|w-u\|^2-\|v-u\|^2\geq 0$$

证明 由$\mathcal{H}$的凸性知，对每个 $\alpha\in(0,1)$有 $v-\alpha(u-v)\in\mathcal{H}$。所以，由 v 的最优性得

$$\begin{aligned}\|v-w\|^2&\leq\|v+\alpha(u-v)-w\|^2\\&=\|v-w\|^2+2\alpha\langle v-w,u-v\rangle+\alpha^2\|u-v\|^2\end{aligned}$$

移项得

$$2\langle v-w,u-v\rangle\geq-\alpha\|u-v\|^2$$

当$\alpha\to 0$有

$$\langle v-w,u-v\rangle\geq 0$$

159

于是

$$\begin{aligned}&\|w-u\|^2=\|w-v+v-u\|^2\\&=\|w-v\|^2+\|v-u\|^2+2\langle v-w,u-v\rangle\geq\|v-u\|^2\end{aligned}$$

结合前面的引理，我们可以很容易将对 SGD 的分析适用于此情况，即在一个闭凸集上加了投影步。只需要注意对每个 t 有

$$\begin{aligned}&\|w^{(t+1)}-w^*\|^2-\|w^{(t)}-w^*\|^2\\&=\|w^{(t+1)}-w^*\|^2-\|w^{(t+\frac{1}{2})}-w^*\|^2+\|w^{(t+\frac{1}{2})}-w^*\|^2-\|w^{(t)}-w^*\|^2\\&\leq\|w^{(t+\frac{1}{2})}-w^*\|^2-\|w^{(t)}-w^*\|^2\end{aligned}$$

所以，当我们增加了投影步后引理 14.1 也是成立的，剩余的分析部分便可以直接得到。■

14.4.2 变步长

SGD的另一个变化形式是把对步长的减小看成是关于 t 的一个函数，就是说，不再使用一个常数步长 η，而是用 η_t。例如，我们可以令 $\eta_t=\frac{B}{\rho\sqrt{t}}$，并且可以达到一个与定理 14.8 类似的界。思想是当我们非常靠近函数极小值的时候，选择步长就要更仔细，以免超过极小值。

14.4.3 其他平均技巧

我们令输出的向量是 $\overline{w}=\frac{1}{T}\sum_{t=1}^{T}w^{(t)}$。还有其他的输出方法，如随机输出一个$w^{(t)}$，$t\in[T]$，或输出过去$\alpha T$次迭代的$w^{(t)}$的平均值，$\alpha\in(0,1)$。还可以取最近几次迭代的加权平均。在某些情况下，这些更复杂的平均策略可以提升收敛速度，如下面定义的强凸函数。

14.4.4 强凸函数*

这一节我们说明 SGD 的一个变型，当问题的目标函数是强凸(见前一章关于强凸的定义 13.4)时，该方法有更快的收敛速度。这依赖下面的论断，它是引理 13.5 的推广。

论断 14.10 如果 f 是 λ-强凸的，那么对于每个 w，u 和 $v\in\partial f(w)$ 成立

$$\langle w-u,v\rangle\geq f(w)-f(u)+\frac{\lambda}{2}\|w-u\|^2$$

[160] 证明过程和引理 13.5 的证明类似，我们将它留作练习。

极小化 λ-强凸函数的随机梯度下降

目的：求解 $\min\limits_{w\in\mathcal{H}} f(w)$

参数：T

初始化：$w^{(1)}=0$

for $t=1,\cdots,T$

　随机选择一个向量 v_t，使得 $\mathbb{E}[v_t|w^{(t)}]\in\partial f(w^{(t)})$

　令 $\eta_t=1/(\lambda t)$

　令 $w^{(t+\frac{1}{2})}=w^{(t)}-\eta_t v_t$

　令 $w^{(t+1)}=\operatorname{argmin}\limits_{w\in\mathcal{H}}\|w-w^{(t+\frac{1}{2})}\|^2$

输出 $\overline{w}=\frac{1}{T}\sum\limits_{t=1}^{T}w^{(t)}$

定理 14.11 假设 f 是 λ-强凸的，$\mathbb{E}[\|v_t\|^2]\leq\rho^2$，令 $w^*\in\operatorname{argmin}\limits_{w\in\mathcal{H}}f(w)$ 是一个最优解，那么

$$\mathbb{E}[f(\overline{w})]-f(w^*)\leq\frac{\rho^2}{2\lambda T}(1+\log(T))$$

证明 令 $\nabla^{(t)}=\mathbb{E}[v_t|w^{(t)}]$。由于 f 是强凸的，$\nabla^{(t)}$ 在 f 在 $w^{(t)}$ 处的次梯度集中，从而有

$$\langle w^{(t)}-w^*,\nabla^{(t)}\rangle\geq f(w^{(t)})-f(w^*)+\frac{\lambda}{2}\|w^{(t)}-w^*\|^2 \tag{14.11}$$

接下来，我们证明

$$\langle w^{(t)}-w^*,\nabla^{(t)}\rangle\leq\frac{\mathbb{E}[\|w^{(t)}-w^*\|^2-\|w^{(t+1)}-w^*\|^2]}{2\eta_t}+\frac{\eta_t}{2}\rho^2 \tag{14.12}$$

因为 $w^{(t+1)}$ 是 $w^{(t+\frac{1}{2})}$ 在 $\mathcal{H}$ 上的投影，$w^*\in\mathcal{H}$，所以 $\|w^{(t+\frac{1}{2})}-w^*\|^2\geq\|w^{(t+1)}-w^*\|^2$。所以，

$$\begin{aligned}\|w^{(t)}-w^*\|^2-\|w^{(t+1)}-w^*\|^2&\geq\|w^{(t)}-w^*\|^2-\|w^{(t+\frac{1}{2})}-w^*\|^2\\&\quad-2\eta_t\langle w^{(t)}-w^*,v_t\rangle-\eta_t^2\|v_t\|^2\end{aligned}$$

对上式两边取期望，重新整理，再结合假设 $\mathbb{E}[\|v_t\|^2]\leq\rho^2$ 可以得式(14.12)。对比式(14.11)和式(14.12)，对 t 求和得

$$\begin{aligned}&\sum_{t=1}^{T}(\mathbb{E}[f(w^{(t)})]-f(w^*))\\&\leq\mathbb{E}\Big[\sum_{t=1}^{T}\Big(\frac{\|w^{(t)}-w^*\|^2-\|w^{(t+1)}-w^*\|^2}{2\eta_t}-\frac{\lambda}{2}\|w^{(t)}-w^*\|^2\Big)\Big]+\frac{\rho^2}{2}\sum_{t=1}^{T}\eta_t\end{aligned}$$

接下来，利用定义 $\eta_t=1/(\lambda t)$ 且注意到右端第一项求和可缩为 $-\lambda T\|w^{(T+1)}-w^*\|^2\leq 0$。因此，

[161] $$\sum_{t=1}^{T}(\mathbb{E}[f(w^{(t)})]-f(w^*))\leq\frac{\rho^2}{2\lambda}\sum_{t=1}^{T}\frac{1}{t}\leq\frac{\rho^2}{2\lambda}(1+\log(T))$$

两边除以 T，再利用詹生不等式即得结论。 ■

评注　Rakhlin，Shamir 和 Sridharan(2012)得到一个收敛速度，其中 $\log(T)$ 对一个变型的算法而言消失了。在该算法中，输出的是最近 $T/2$ 次迭代的平均，即 $\overline{w}=\frac{2}{T}\sum_{t=T/2+1}^{T} w^{(t)}$。Shamir 和 Zhang(2013)证明了如果输出 $\overline{w}=w^{(T)}$，定理 14.11 也是成立的。

14.5　用 SGD 进行学习

我们已经介绍和分析了 SGD 方法求解一般的凸函数。下面我们将考虑其学习任务的能力。

14.5.1　SGD 求解风险极小化

回顾一下在学习中我们面临的问题是极小化风险函数

$$L_{\mathcal{D}}(w)=\mathop{\mathbb{E}}_{z\sim\mathcal{D}}[\ell(w,z)]$$

我们看到经验风险极小化方法极小化的是经验风险 $L_S(w)$，是对极小化 $L_{\mathcal{D}}(w)$ 的一个估计。SGD 允许我们采用不同的方法可以直接极小化 $L_{\mathcal{D}}(w)$。因为我们不知道 $\mathcal{D}$，所以不能简单地计算 $\nabla L_{\mathcal{D}}(w)$，也不能通过 GD 方法来极小化 $L_{\mathcal{D}}(w)$。而用 SGD，我们需要做的是找到 $L_{\mathcal{D}}(w)$ 梯度的一个无偏估计，即条件期望值为 $\nabla L_{\mathcal{D}}(w^{(t)})$ 的一个随机向量。现在，我们将看到怎样的一个估计能够简单地构造出来。

为简单起见，我们首先考虑目标函数是可微的情形。因此，风险函数 $L_{\mathcal{D}}$ 也是可微的。随机向量 v_t 的构造如下：首先，采样 $z\sim\mathcal{D}$，再定义 v_t 为关于 w 的损失函数 $\ell(w,z)$ 在 $w^{(t)}$ 处的梯度。然后，由梯度的线性得

$$\mathbb{E}[v_t|w^{(t)}]=\mathop{\mathbb{E}}_{z\sim\mathcal{D}}[\nabla\ell(w^{(t)},z)]=\nabla\mathop{\mathbb{E}}_{z\sim\mathcal{D}}[\ell(w^{(t)},z)]=\nabla L_{\mathcal{D}}(w^{(t)}) \tag{14.13}$$

所以，损失函数 $\ell(w,z)$ 在 $w^{(t)}$ 处的梯度是风险函数 $L_{\mathcal{D}}(w^{(t)})$ 梯度的一个无偏估计，并且这个梯度是可以通过在 t 次迭代时采样一个新的样本 $z\sim\mathcal{D}$ 来构造。

同样的论点对不可微的损失函数也是成立的。令 v_t 是 $\ell(w,z)$ 在 $w^{(t)}$ 处的次梯度。那么，对于每个 u 有

$$\ell(u,z)-\ell(w^{(t)},z)\geq\langle u-w^{(t)},v_t\rangle$$

两边关于 $z\sim\mathcal{D}$ 取期望，关于 $w^{(t)}$ 取条件，得

$$\begin{aligned}L_{\mathcal{D}}(u)-L_{\mathcal{D}}(w^{(t)})&=\mathbb{E}[\ell(u,z)-\ell(w^{(t)},z)\mid w^{(t)}]\geq\mathbb{E}[\langle u-w^{(t)},v_t\rangle\mid w^{(t)}]\\&=\langle u-w^{(t)},\mathbb{E}[v_t\mid w^{(t)}]\rangle\end{aligned}$$

从而 $\mathbb{E}[v_t|w^{(t)}]$ 是 $L_{\mathcal{D}}(w)$ 在 $w^{(t)}$ 处的次梯度。

简言之，极小化风险函数的随机梯度下降框架如下。

极小化 $L_{\mathcal{D}}(w)$ 的随机梯度下降(SGD)

参数：标量 $\eta>0$，整数 $T>0$

初始化：$w^{(1)}=0$

for $t=1,2,\cdots,T$

　采样 $z\sim\mathcal{D}$

　选择 $v_t\in\partial\ell(w^{(t)},z)$

　更新 $w^{(t+1)}=w^{(t)}-\eta v_t$

输出 $\overline{w}=\frac{1}{T}\sum_{t=1}^{T}w^{(t)}$

162

我们将利用对 SGD 的分析来得到凸利普希茨有界学习问题的样本复杂度。由定理 14.8 可得下面的推论。

推论 14.12　考虑带有参数 ρ 和 B 的凸利普希茨有界学习问题。对每个 $\varepsilon>0$，如果我们运行极小化 $L_{\mathcal{D}}(w)$ 的 SGD 方法的迭代次数（即样本的个数）$T\geq\frac{B^2\rho^2}{\varepsilon^2}$ 且 $\eta=\sqrt{\frac{B^2}{\rho^2 T}}$，那么 SGD 的输出满足

$$\mathbb{E}[L_{\mathcal{D}}(\overline{w})]\leq\min_{w\in\mathcal{H}}L_{\mathcal{D}}(w)+\varepsilon$$

有趣的是所需的样本复杂度和对正则化损失极小化得到的样本复杂度保证是一个量级的。实际上，SGD 的样本复杂度甚至比我们从正则化损失极小化得到的因子为 8 的样本复杂度要好。

14.5.2　SGD 求解凸光滑学习问题的分析

在前一章中我们看到正则化损失极小化准则也可以学习一类凸光滑有界的学习问题。现在我们来证明 SGD 算法也可以用来求解这类问题。

定理 14.13　假设对于所有的 z，损失函数 $\ell(\cdot,z)$ 是凸的，β-光滑的且非负，那么如果利用 SGD 求解 $L_{\mathcal{D}}(w)$，对于每个 w^* 有

$$\mathbb{E}[L_{\mathcal{D}}(\overline{w})]\leq\frac{1}{1-\eta\beta}\left(L_{\mathcal{D}}(w^*)+\frac{\|w^*\|^2}{2\eta T}\right)$$

证明　注意到如果一个函数是 β-光滑的且非负，那么它是自有界的，即有

[163]

$$\|\nabla f(w)\|^2\leq 2\beta f(w)$$

为了分析 SGD 求解凸光滑问题，我们定义 SGD 算法的随机样本 $z_1,\ldots,z_T$，令 $f_t(\cdot)=\ell(\cdot,z_t)$，$v_t=\nabla f_t(w^{(t)})$。对所有的 t，f_t 是凸函数，所以 $f_t(w^{(t)})-f_t(w^*)\leq\langle v_t,w^{(t)}-w^*\rangle$。对 t 求和，再利用引理 14.1 得

$$\sum_{t=1}^{T}(f_t(w^{(t)})-f_t(w^*))\leq\sum_{t=1}^{T}\langle v_t,w^{(t)}-w^*\rangle\leq\frac{\|w^*\|^2}{2\eta}+\frac{\eta}{2}\sum_{t=1}^{T}\|v_t\|^2$$

结合 f_t 的自有界性得

$$\sum_{t=1}^{T}(f_t(w^{(t)})-f_t(w^*))\leq\frac{\|w^*\|^2}{2\eta}+\eta\beta\sum_{t=1}^{T}f_t(w^{(t)})$$

除以 T 再重新排列得

$$\frac{1}{T}\sum_{t=1}^{T}f_t(w^{(t)})\leq\frac{1}{1-\eta\beta}\left(\frac{1}{T}\sum_{t=1}^{T}f_t(w^*)+\frac{\|w^*\|^2}{2\eta T}\right)$$

接下来，对前一个式子的两边关于 $z_1,\ldots,z_T$ 取期望。显然，$\mathbb{E}[f_t(w^*)]=L_{\mathcal{D}}(w^*)$。此外，利用证明定理 14.8 相同的论据可得

$$\mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T}f_t(w^{(t)})\right]=\mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T}L_{\mathcal{D}}(w^{(t)})\right]\geq\mathbb{E}[L_{\mathcal{D}}(\overline{w})]$$

综上结论得证。 ■

可以直接得到如下推论：

推论 14.14　考虑一个带有参数 β 和 B 的凸光滑有界的学习问题。假设对所有的 $z\in Z$ 有 $\ell(0,z)\leq 1$。对每一个 $\varepsilon>0$，令 $\eta=\frac{1}{\beta(1+3/\varepsilon)}$，那么运行 SGD 算法 $T\geq 12B^2\beta/\varepsilon^2$ 次成立

$$\mathbb{E}[L_{\mathcal{D}}(\overline{w})] \leq \min_{w \in \mathcal{H}} L_{\mathcal{D}}(w) + \varepsilon$$

14.5.3 SGD 求解正则化损失极小化

我们已经证明了在最坏的情况下 SGD 也有着和正则损失极小化相同的样本复杂度。然而，在某些分布上，正则损失极小化可能会产生更好的解。所以，在某些情况下我们还是想要求解与正则化损失极小化相关联的优化问题，即，[⊖]

$$\min_{w}\left(\frac{\lambda}{2}\|w\|^2 + L_S(w)\right) \tag{14.14}$$

由于我们处理的是凸学习问题，其中损失函数是凸的，正如我们在本节将看到的，上面的问题也是凸优化问题，同样可以用 SGD 来求解。 [154]

定义 $f(w) = \frac{\lambda}{2}\|w\|^2 + L_S(w)$。注意到 f 是 λ-强凸函数；所以，我们可以利用 14.4.4 节($\mathcal{H}=\mathbb{R}^d$)中给出的 SGD 的变型进行求解。为了应用该算法，我们只需找到一种方式构造 f 在 $w^{(t)}$ 处次梯度的无偏估计。注意到，如果我们从 S 中均匀地选择 z，且选择 $v_t \in \partial\ell(w^{(t)}, z)$，那么 $\lambda w^{(t)} + v_t$ 的期望值就是 f 在 $w^{(t)}$ 处的次梯度。

为了分析得到的算法，我们先将更新准则(假设 $\mathcal{H}=\mathbb{R}^d$，所以投影步就不重要了)重写如下：

$$\begin{aligned} w^{(t+1)} &= w^{(t)} - \frac{1}{\lambda t}(\lambda w^{(t)} + v_t) = \left(1 - \frac{1}{t}\right)w^{(t)} - \frac{1}{\lambda t} v_t = \frac{t-1}{t} w^{(t)} - \frac{1}{\lambda t} v_t \\ &= \frac{t-1}{t}\left(\frac{t-2}{t-1} w^{(t-1)} - \frac{1}{\lambda(t-1)} v_{t-1}\right) - \frac{1}{\lambda t} v_t = -\frac{1}{\lambda t}\sum_{i=1}^{t} v_i \end{aligned} \tag{14.15}$$

如果假设损失函数是 ρ-利普希茨，从而对于所有的 t 有 $\|v_t\| \leq \rho$，所以 $\|\lambda w^{(t)}\| \leq \rho$，进一步得

$$\lambda\|w^{(t)} + v_t\| \leq 2\rho$$

所以定理 14.11 告诉我们执行 T 次迭代后有

$$\mathbb{E}[f(\overline{w})] - f(w^*) \leq \frac{4\rho^2}{\lambda T}(1 + \log(T))$$

14.6 小结

我们介绍了梯度下降和随机梯度下降算法，连同它们的一些变化形式。分析了它们的收敛速度，计算了可以确保期望目标至多是 ε 加上最优目标的迭代次数。最重要的是我们证明了使用 SGD 可以直接极小化风险函数。这是通过从 $\mathcal{D}$ 中独立同分布地采样得到一个点，并使用损失函数在当前假设 $w^{(t)}$ 处的次梯度作为风险函数梯度(或次梯度)的无偏估计来实现的。这意味着迭代次数的界也能得到样本复杂度的界。最后，我们说明了如何将 SGD 应用到正则化风险极小化中。在下面的章节中，我们将说明 SGD 如何得到求解与正则化风险极小化相关联的优化问题的非常简单的算法。 [165]

14.7 文献评注

SGD 可以追溯到文献 Robbins 和 Monro(1951)。在大规模机器学习问题中，SGD 方法是特别有效的，可参考文献 Murata(1998)，Le Cun(2004)，Zhang(2004)，Bottou 和 Bousquet(2008)，Shalev-Shwartz，Singer 和 Srebro(2007)，Shalev-Shwartz 和 Srebro

---

⊖ λ 除以 2 是为了方便计算。

(2008)，在优化领域，它是在随机优化的背景下被研究的，可参考文献 Nemirovski 和 Yudin(1978)，Nesterov 和 Nesterov(2004)，Nesterov(2005)，Nemirovski、Juditsky、Lan 和 Shapiro(2009)，Shapiro、Dentcheva 和 Ruszczyński(2009)。

我们所得到的求解强凸函数的界要归因于 Hazan、Agarwal 和 Kale (2007)，正如前面提到的，改进的界可以参考文献 Rakhlin、Shamir 和 Sridharan(2012)。

14.8　练习

14.1　证明论断 14.10。(提示：扩展引理 13.5 的证明。)

14.2　证明推论 14.14。

14.3　感知器作为次梯度下降算法：令 $S=((x_1, y_1), \cdots, (x_m, y_m))\in(\mathbb{R}^d\times\{\pm1\})^m$。假设存在 $w\in\mathbb{R}^d$ 使得对每个 $i\in[m]$ 都有 $y_i\langle w, x_i\rangle\geq1$，令 w^* 是满足前面要求的所有向量中范数最小的一个。设 $R=\max_i\|x_i\|$，定义函数

$$f(w)=\max_{i\in[m]}(1-y_i\langle w,x_i\rangle)$$

- 说明 $\min_{w:\|w\|\leq\|w^*\|} f(w)=0$，且使得 $f(w)<1$ 的任意的 w 能分离 S 中的样本。
- 说明如何计算 f 的次梯度。
- 描述并分析这种情况下的次梯度下降算法，将该算法和 9.1.2 节中的批处理感知器算法进行比较和分析。

166 * 14.4　变步长：证明一个与定理 14.8 的类似的结论，即 SGD 取一个变步长 $\eta=\frac{B}{\rho\sqrt{t}}$。

支持向量机

本章以及下一章中我们要讨论一种非常有用的机器学习工具：在高维特征空间学习线性预测器的支持向量机(SVM)，在高维特征空间中，同时要面临样本复杂度和计算复杂度的挑战。

SVM 算法通过搜索“大间隔”分类器来应对样本复杂度的挑战。粗略地说，如果所有的样本不仅被分类超平面正确分开并且远离分类超平面，那么我们就说一个半空间用大间隔分开了训练样本集。该算法要求输出一个有着大间隔的分类器甚至可以在特征空间维度很高(甚至无穷)的情况下得到一个小的样本复杂度。我们将介绍间隔的概念并将其与正则损失最小化以及感知器算法的收敛速率联系起来。

在下一章中我们将用核的概念来应对计算复杂度的挑战。

15.1 间隔与硬 SVM

令 $S=(x_1, y_1), \ldots, (x_m, y_m)$ 是训练样本集，其中每个 $x_i \in \mathbb{R}^d$，$y_i \in \{\pm 1\}$。如果存在一个半空间 (w, b)，使得对于所有 i，有 $y_i = \mathrm{sign}(\langle w, x_i \rangle + b)$，我们就说该训练集是线性可分的。这个条件也可以写为

$$\forall i \in [m], y_i(\langle w, x_i \rangle + b) > 0$$

所有满足该条件的半空间 (w, b) 都是 ERM 假设(它们的 0-1 误差为 0，为最小可能的误差)。对于任何可分的训练样本，存在着很多 ERM 半空间。那么，学习器会在它们之中挑选哪个作为最终输出呢？

比如考虑训练集如下图所示：

当虚线与实线均分开了这 4 个样本，我们直观地会选择虚线而不是实线。一种将这种直观形式化的方式就是用间隔的概念。

定义超平面在训练集上的间隔为训练集中的点到超平面的最短距离。如果一个超平面有大的间隔的话，尽管每个样本有小的扰动，该超平面仍将分开训练集。

我们之后将会看到半空间的误差可以由其在训练样本上的间隔来界定(间隔越大，误差越小)，而与该半空间的欧几里得维度无关。

硬 SVM 是一种学习规则，在这种规则下我们可以得到一个用最大可能间隔分开训练集的 ERM 超平面。为了正式地定义硬 SVM，我们首先用定义半空间的参数来表示一个点 x 到超平面的距离。

论断 15.1 一个点 x 到由 (w, b) 定义的超平面的距离为 $|\langle w, x \rangle + b|$，其中，

$\|w\|=1$。

证明　定义一个点 x 到超平面的距离为

$$\min\{\|x-v\|:\langle w,v\rangle+b=0\}$$

取 $v=x-(\langle w,x\rangle+b)w$，可得

$$\langle w,v\rangle+b=\langle w,x\rangle-(\langle w,x\rangle+b)\|w\|^2+b=0$$

以及

$$\|x-v\|=|\langle w,x\rangle+b|\,\|w\|=|\langle w,x\rangle+b|$$

因此，该距离至多为 $|\langle w,x\rangle+b|$。接下来，取超平面上另外一个点 u，因此有 $\langle w,u\rangle+b=0$。则

$$\begin{aligned}\|x-u\|^2&=\|x-v+v-u\|^2\\&=\|x-v\|^2+\|v-u\|^2+2\langle x-v,v-u\rangle\\&\geq\|x-v\|^2+2\langle x-v,v-u\rangle\\&=\|x-v\|^2+2(\langle w,x\rangle+b)\langle w,v-u\rangle\\&=\|x-v\|^2\end{aligned}$$

其中最后一个等式成立的原因是 $\langle w,v\rangle=\langle w,u\rangle=-b$。因此，$x$ 与 u 的距离至少为 x 与 v 的距离，证毕。■

358

上述论断的基础是认为训练集到分类超平面的最近点是 $\min_{i\in[m]}|\langle w,x_i\rangle+b|$。因此，硬 SVM 规则为：

$$\operatorname*{argmax}_{(w,b):\|w\|=1}\ \min_{i\in[m]}|\langle w,x_i\rangle+b|\quad\text{s.t.}\quad\forall i,\ y_i(\langle w,x_i\rangle+b)>0$$

当上述问题有解（即可分情况），我们可以写成如下等价问题（见练习 15.1）：

$$\operatorname*{argmax}_{(w,b):\|w\|=1}\ \min_{i\in[m]}y_i(\langle w,x_i\rangle+b)\tag{15.1}$$

接下来，我们用二次优化问题[⊖]的形式给出硬 SVM 的另一种等价形式：

硬 SVM

输入：$(x_1,y_1),\cdots(x_m,y_m)$

求解：

$$(w_0,b_0)=\operatorname*{argmin}_{(w,b)}\|w\|^2\quad\text{s.t.}\quad\forall i,\ y_i(\langle w,x_i\rangle+b)\geq1\tag{15.2}$$

输出：$\hat{w}=\dfrac{w_0}{\|w_0\|}$，$\hat{b}=\dfrac{b_0}{\|w_0\|}$

接下来的引理将说明硬 SVM 的输出确实是最大间隔的分类超平面。直观上讲，硬 SVM 是在搜索这样的 w，即在所有向量中有着最小范数分开了原数据并且对于所有 i，$|\langle w,x_i\rangle+b|\geq1$。换言之，我们可以强制间隔就是 1，但现在要通过 w 的范数来度量间隔的大小。因此，找有最大间隔半空间的问题就变成了找有着最小范数 w 的问题。正式地：

引理 15.2　硬 SVM 的输出是式(15.1)的一个解。

证明　令 (w^*,b^*) 是式(15.1)的一个解，定义由 (w^*,b^*) 得到的间隔为 $\gamma^*=\min_{i\in[m]}y_i(\langle w^*,x_i\rangle+b^*)$。因此，对于所有的 i，有

⊖　二次优化问题就是目标函数为凸二次函数且限制条件为线性不等式的优化问题。

$$y_i(\langle w^*, x_i\rangle + b^*) \geqslant \gamma^*$$

也即

$$y_i\left(\left\langle \frac{w^*}{\gamma^*}, x_i\right\rangle + \frac{b^*}{\gamma^*}\right) \geqslant 1$$

因此，这样的对$\left(\frac{w^*}{\gamma^*}, \frac{b^*}{\gamma^*}\right)$满足式(15.2)给出的二次优化问题的条件。因此，$\|w_0\| \leqslant \left\|\frac{w^*}{\gamma^*}\right\| = \frac{1}{\gamma^*}$。则对于所有的$i$，

$$y_i(\langle \hat{w}, x_i\rangle + \hat{b}) = \frac{1}{\|w_0\|} y_i(\langle w_0, x_i\rangle + b_0) \geqslant \frac{1}{\|w_0\|} \geqslant \gamma^*$$

由于$\|\hat{w}\| = 1$，可得$(\hat{w}, \hat{b})$就是式(15.1)的最优解。 ■ 169

15.1.1 齐次情况

考虑齐次半空间往往更为方便，即，半空间是通过原点的，因此可以被定义为$\mathrm{sign}(\langle w, x\rangle)$，其中偏差项$b$为0。硬SVM在齐次半空间条件下就是求解下式：

$$\min_w \|w\|^2 \quad \text{s.t.} \quad \forall i, y_i\langle w, x_i\rangle \geqslant 1 \tag{15.3}$$

如同我们在第9章已经讨论过的，我们可以将一个非齐次半空间的学习问题退化成一个齐次半空间的学习问题，只需给每个实例x_i增加一维特征，即将特征维度增加为$d+1$。

注意，显然式(15.2)中给出的优化问题不约束偏差项b，但是如果我们用式(15.3)在$\mathbb{R}^{d+1}$上学习半空间，那我们也将会约束偏差项(即，权重向量的第$d+1$个分量)。然而，对b的正则通常对于样本复杂度不会产生一个显著的影响。

15.1.2 硬SVM的样本复杂度

回想在$\mathbb{R}^d$上半空间的VC维是$d+1$。则学习半空间的样本复杂度随着问题的维度而增长。更进一步地，学习的基本定理告诉我们如果样本数明显小于d/ε，将没有能学习到ε-精确的半空间的算法。这个问题在d很大时尤其显著。

为了解决该问题，我们将对潜在的数据分布作一个附加的假设。特别地，我们将定义一个"用间隔γ可分"的假设，并且说明如果数据可由间隔γ分开，那么上述问题的样本复杂度将由$1/\gamma^2$的函数界定。这就是说哪怕维度很大(甚至是无限)的，只要相关的数据是在某个间隔下可分的，我们仍可得到一个小的样本复杂度。这与学习的基本定理给出的下界是不矛盾的，因为我们此时对潜在的数据分布作了一个额外的假设。

在我们正式地定义间隔可分假设之前，需要考虑一个尺度问题。设想一个训练集$S=(x_1, y_1), \cdots, (x_m, y_m)$用间隔$\gamma$可分，即式(15.1)的最大目标函数值至少为$\gamma$。那么，对于任意正的尺度因子$\alpha>0$，训练集$S'=(\alpha x_1, y_1), \cdots, (\alpha x_m, y_m)$将由间隔$\alpha\gamma$分开。这就是说，一个对数据简单的尺度变化可以使得训练集由任意大的间隔分开。所以为了给间隔一个有意义的定义，我们必须同时考虑样本的尺度。一种将上述想法形式化的方式就是考虑如下定义。

定义 15.3 令$\mathcal{D}$是在$\mathbb{R}^d \times \{\pm 1\}$上的分布。我们说$\mathcal{D}$由$(\gamma, \rho)$-间隔可分，如果存在$(w^*, b^*)$使得$\|w^*\|=1$，且以1的概率在$(x, y)\sim\mathcal{D}$的选择下有$y(\langle w^*, x\rangle - b^*) \geqslant \gamma$以及$\|x\| \leqslant \rho$成立。类似地，我们说$\mathcal{D}$由$(\gamma, \rho)$-间隔用齐次半空间可分，如果上述成立并且半空间取$(w^*, 0)$的形式。 170

在本书的进阶部分(第26章)，我们将会证明硬SVM的复杂度由$(\rho/\gamma)^2$决定并且与维度d无关。特别地，第26.3节中定理26.13表述如下：

定理15.4　令$\mathcal{D}$是在$\mathbb{R}^d\times\{\pm1\}$上的分布且满足采用齐次半空间下的$(\gamma,\rho)$-间隔可分假设。那么，在选择大小为$m$的训练集后，以至少$1-\delta$的概率有硬SVM输出的0-1误差最多为

$$\sqrt{\frac{4(\rho/\gamma)^2}{m}}+\sqrt{\frac{2\log(2/\delta)}{m}}$$

评注(间隔与感知器)　在第9.1.2节中，我们已经描述并分析了用感知器算法来找到关于半空间类的ERM假设。特别地，在定理9.1中我们给出了感知器在一个给定训练上可能需要迭代次数的上界。在练习15.2中可以说明这个上界确切地是$(\rho/\gamma)^2$，其中ρ是样本的半径，γ是间隔。

15.2　软SVM与范数正则化

硬SVM的形式假定了训练集是线性可分的，这其实是一个很强的假设。软SVM可以认为是对硬SVM规则的一种放松，因此可以在训练集不是线性可分时应用。

在式(15.2)的优化问题中，有一个很强的限制，即对于所有i，有$y_i(\langle w,x_i\rangle+b)\geq1$。一个很自然的放松就是允许该约束变成对于训练集中的一些样本不成立。即引入一些非负松弛变量$\xi_1,\dots,\xi_m$，用约束$y_i(\langle w,x_i\rangle+b)\geq1-\xi_i$替代约束$y_i(\langle w,x_i\rangle+b)\geq1$。这就是说，$\xi_i$度量了约束$y_i(\langle w,x_i\rangle+b)\geq1$不满足的程度。软SVM联合最小化$w$的范数(有间隔相关)与$\xi$的平均(与约束不满足的程度有关)。二者的权衡用参数λ来控制。因此，软SVM优化问题如下：

软SVM

输入：$(x_1,y_1),\dots(x_m,y_m)$

参数：$\lambda>0$

求解：

$$\min_{w,b,\xi}\left(\lambda\|w\|^2+\frac{1}{m}\sum_{i=1}^{m}\xi_i\right)\tag{15.4}$$

$$\text{s.t.}\quad\forall i,\ y_i(\langle w,x_i\rangle+b)\geq1-\xi_i\ \text{且}\ \xi_i\geq0$$

输出：w,b

171

我们可以将式(15.4)重写为正则损失最小化问题的形式。回想之前定义的合页损失：

$$\ell^{\text{hinge}}((w,b),(x,y))=\max\{0,1-y(\langle w,x\rangle+b)\}$$

给定(w,b)以及训练集S，S上平均的合页损失记作$L_S^{\text{hinge}}((w,b))$。现在，考虑一个正则损失最小化问题：

$$\min_{w,b}\left(\lambda\|w\|^2+L_S^{\text{hinge}}((w,b))\right)\tag{15.5}$$

论断15.5　*式(15.4)与式(15.5)是等价的。*

证明　固定某个w,b，考虑式(15.4)中在ξ下的最小化。固定某个i，由于ξ_i一定非负，如果$y_i(\langle w,x_i\rangle+b)\geq1$，$\xi_i$最优赋值为0，否则，最优赋值为$1-y_i(\langle w,x_i\rangle+b)$。换言之，对于所有$i$，$\xi_i=\ell^{\text{hinge}}((w,b),(x_i,y_i))$，故论断成立。■

因此，我们看到软 SVM 本质上是之前章节学习过的正则损失最小化。一个软 SVM 算法，即式(15.5)的解，倾向于选择范数低的分类器。式(15.5)试图最小化的目标函数不仅对训练误差有惩罚，还对大的范数有惩罚。

同样地，考虑软 SVM 学习一个齐次半空间往往更加方便，即偏差项 b 为 0。这就是下述优化问题：

$$\min_{w}(\lambda\|w\|^2+L_S^{\text{hinge}}(w)) \tag{15.6}$$

其中

$$L_S^{\text{hinge}}(w)=\frac{1}{m}\sum_{i=1}^{m}\max\{0,1-y\langle w,x_i\rangle\}$$

15.2.1　软 SVM 的样本复杂度

我们现在分析对于齐次半空间的软 SVM(即式(15.6)的输出)的样本复杂度。在推论 13.8 中，我们得到了在假定损失函数凸利普希茨的情况下正则损失最小化框架的泛化界。我们已经说明合页损失是凸的，所以现在只剩下分析合页损失的利普希茨性。

论断 15.6　令 $f(w)=\max\{0,1-y\langle w,x\rangle\}$，则，$f$ 是 $\|x\|$-利普希茨的。

证明　很容易验证 f 在 w 上的任意次梯度都是 αx 的形式，其中 $|\alpha|\leqslant 1$。由引理 14.7，论断得证。 ■

因此，由推论 13.8 可得：

推论 15.7　令 $\mathcal{D}$ 是在 $\mathcal{X}\times\{\pm 1\}$ 上的分布，其中 $\mathcal{X}=\{x:\|x\|\leqslant\rho\}$。考虑在训练集 $S\sim\mathcal{D}^m$ 上运行软 SVM 算法(式 15.5)，令 $A(S)$ 是软 SVM 的解。那么，对于每个 u，

$$\mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}[L_{\mathcal{D}}^{\text{hinge}}(A(S))]\leqslant L_{\mathcal{D}}^{\text{hinge}}(u)+\lambda\|u\|^2+\frac{2\rho^2}{\lambda m} \tag{172}$$

更进一步，由于合页损失是 0—1 损失的上界，故而

$$\mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}[L_{\mathcal{D}}^{0\text{-}1}(A(S))]\leqslant L_{\mathcal{D}}^{\text{hinge}}(u)+\lambda\|u\|^2+\frac{2\rho^2}{\lambda m}$$

最后，对于每个 $B>0$，如果我们取 $\lambda=\sqrt{\frac{2\rho^2}{B^2 m}}$，那么

$$\mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}[L_{\mathcal{D}}^{0\text{-}1}(A(S))]\leqslant \mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}[L_{\mathcal{D}}^{\text{hinge}}(A(S))]\leqslant \min_{w:\|w\|\leqslant B}L_{\mathcal{D}}^{\text{hinge}}(w)+\sqrt{\frac{8\rho^2B^2}{m}}$$

因此，我们可以看到可以通过半空间参数的范数的函数来控制学习一个半空间所需的样本复杂度，而与定义半空间的欧几里得维度无关。这对于高维特征空间的学习尤其显著，这一点我们将在之后的章节讨论。

评注　$\mathcal{X}$ 包含范数约束的向量这个条件源于损失函数是利普希茨的要求。这不仅仅是一个技术性要求。如我们之前所讨论，如果不对样本的尺度做限制，用大的间隔可分将没有任何意义。事实上，如果对尺度不做限制，我们总是可以通过对所有样本乘以一个尺度因子使得间隔变得无限大。

15.2.2　间隔、基于范数的界与维度

我们针对硬 SVM 与软 SVM 提出的界不依赖于实例空间的维度。事实上，这些界依赖于样本的范数 ρ，半空间的范数 B(或者是间隔的参数 γ)，以及在不可分的情况下，所有

范数小于等于 B 的半空间的最小合页损失。另一方面，齐次半空间的 VC 维是 d，这意味着 ERM 假设的误差随 $\sqrt{d/m}$ 减小。我们现在给出一个例子，$\rho^2 B^2 \ll d$，由此来说明由推论 15.7给出的界要比 VC 界好得多。

考虑一个根据主题学习短文本的分类问题，即，判断某个文本是否是关于体育的。我们首先需要将文本表示为向量。一个简单而有效的方式就是采用文字包(bag-of-words)的表示，即，我们定义一个文字的字典，并令其维度 d 为字典中文字的个数。给定一个文本，我们将其表示为一个向量 $x \in \{0,1\}^d$，其中当字典中的第 i 个文字在文本中出现时 $x_i=1$，否则 $x_i=0$。因此，对于这个问题，ρ^2 的值就是在给定的文本中有区别的文字的最大个数。

对于这个问题，待求的半空间给文字分配了权重。我们假定在给一些文字分配正或者负的权重后能够以足够精确度来判定文本是否是关于体育的。因此，对于这个问题，B^2 可以设定为小于 100。总之，认为 $B^2\rho^2$ 的值小于 10 000 是合理的。

另一方面，一般的字典包含的文字明显是大于 10 000 的。例如，英语中至少有 100 000个有区别的文字。因此我们可以看出来这个问题中，采用 SVM 规则学习一个半空间和采用一个合适的 ERM 规则学习一个半空间的区别不是一个量级的。

173

当然，构造一个问题使得 SVM 界比 VC 界差得多也是可能的。当使用 SVM 时，我们其实引入了另外一种形式的归纳偏置——我们选择了大间隔的半空间。这个归纳偏置可能明显降低估计误差，也可能增大逼近误差。

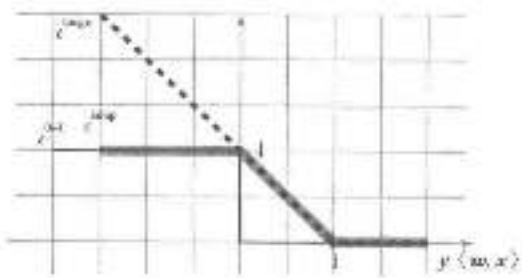
15.2.3　斜坡损失*

推论 15.7 中给出的基于间隔的界依赖于我们最小化的是合页损失。如之前的小节中看到的，项 $\sqrt{\rho^2 B^2/m}$ 有可能比 VC 界相关的项 $\sqrt{d/m}$ 要小得多。然而，推论 15.7 的逼近误差与合页损失有关，而 VC 界与 0—1 损失有关。由于合页损失是0—1损失的上界，因此由0—1损失得到的逼近误差永远不会超过由合页损失得到的逼近误差。

对于 0—1 损失，不可能得到包含估计误差项 $\sqrt{\rho^2 B^2/m}$ 的界。这是由于 0—1 损失是尺度不敏感的，导致了当我们度量 0—1 损失下的误差时，考虑 w 的范数或者相应的间隔是没有意义的。然而，还是有可能定义一种损失函数，在这种损失函数下，首先它是尺度敏感的，因而可以在估计误差中包含项 $\sqrt{\rho^2 B^2/m}$。与此同时，它还与 0—1 损失更为接近。其中一种满足上述条件的就是斜坡损失(ramp loss)，定义如下：

$$\ell^{\text{ramp}}(w,(x,y)) = \min\{1, \ell^{\text{hinge}}(w,(x,y))\}$$
$$= \min\{1, \max\{0, 1 - y\langle w, x\rangle\}\}$$

斜坡损失如 0—1 损失一样惩罚错误，并且对间隔分开的样本不做惩罚。斜坡损失与0—1损失的区别仅仅在于那些被正确分类但是没有一个明显间隔的样本上。在本书的进阶部分给出了斜坡损失的泛化界(见 26.3 节)。



174

SVM 采用合页损失而不是斜坡损失的原因在于合页损失是凸的，因此从计算角度而言最小化合页损失要更为易行。与此同时，最小化斜坡损失的问题是计算困难的。

15.3　最优化条件与“支持向量”*

“支持向量机”的名字来源于硬 SVM 的求解过程，即由那些与分类超平面距离确实是 $1/\|w_0\|$ 的样本来“支持”（即线性张成）。因此这些向量被称作支持向量。为了看清这一点，我们采用 Fritz John 最优化条件。

定理 15.8　令 w_0 如式(15.3)中定义，$I=\{i: |\langle w_0, x_i\rangle|=1\}$。则存在系数 $\alpha_1, \ldots, \alpha_m$ 使得

$$w_0=\sum_{i\in I}\alpha_i x_i$$

样本 $\{x_i: i\in I\}$ 称为支持向量。

将下述引理与式(15.3)联立可证得上述定理。

引理 15.9(Fritz John)　假定

$$w^* \in \operatorname*{argmin}_{w} f(w) \quad \text{s.t.} \quad \forall i\in[m], g_i(w)\le 0$$

其中 $f, g_1, \ldots, g_m$ 可导。那么，存在 $\alpha\in\mathbb{R}^m$ 使得 $\nabla f(w^*)+\sum_{i\in I}\alpha_i\nabla g_i(w^*)=\mathbf{0}$，其中 $I=\{i: g_i(w^*)=0\}$。

15.4　对偶*

SVM 最早提出来的时候，许多作页是通过考虑式(15.3)的对偶形式获得的。我们之前对 SVM 的描述是没有依赖对偶的。为了内容的完整性，我们下面将介绍如何得到式(15.3)的对偶形式。

首先我们对式(15.3)重写出一个等价问题形式如下。考虑如下函数：

$$g(w)=\max_{\alpha\in\mathbb{R}^m,\alpha\ge 0}\sum_{i=1}^{m}\alpha_i(1-y_i\langle w,x_i\rangle)=\begin{cases}0 & \text{若 } \forall i, y_i\langle w,x_i\rangle\ge 1\\ +\infty & \text{其他}\end{cases}$$

因此我们可以将式(15.3)重写为

$$\min_{w}(\|w\|^2+g(w)) \tag{15.7}$$

重新排列上式中项的顺序，我们可以将式(15.3)重写为如下问题：

$$\min_{w}\max_{\alpha\in\mathbb{R}^m,\alpha\ge 0}\left(\frac{1}{2}\|w\|^2-\sum_{i=1}^{m}\alpha_i(1-y_i\langle w,x_i\rangle)\right) \tag{15.8}$$

175

现在我们将等式中求最大与求最小的顺序交换，这只会使目标函数值减小（见练习 15.4），之后有

$$\min_{w}\max_{\alpha\in\mathbb{R}^m,\alpha\ge 0}\left(\frac{1}{2}\|w\|^2+\sum_{i=1}^{m}\alpha_i(1-y_i\langle w,x_i\rangle)\right)$$

$$\ge \max_{\alpha\in\mathbb{R}^m,\alpha\ge 0}\min_{w}\left(\frac{1}{2}\|w\|^2+\sum_{i=1}^{m}\alpha_i(1-y_i\langle w,x_i\rangle)\right)$$

上述不等式称作弱对偶性(weak duality)。已证明对于我们这种情况，强对偶性也是成立的，即上述不等式可以取等号。因此，对偶问题为

$$\max_{\alpha\in\mathbb{R}^m,\alpha\ge 0}\min_{w}\left(\frac{1}{2}\|w\|^2+\sum_{i=1}^{m}\alpha_i(1-y_i\langle w,x_i\rangle)\right) \tag{15.9}$$

下面我们对上述问题进行简化，固定 $\boldsymbol{\alpha}$ 之后，与 $\boldsymbol{w}$ 相关的优化问题就是无约束的并且目标函数是可导的，因此，取最优时，梯度值为 0：

$$\boldsymbol{w}-\sum_{i=1}^{m}\alpha_i y_i \boldsymbol{x}_i=0 \Rightarrow \boldsymbol{w}=\sum_{i=1}^{m}\alpha_i y_i \boldsymbol{x}_i$$

上式告诉我们最优解由样本线性张成，并且启示我们之后可以用核来得到 SVM。将上式代入到式(15.9)中可以将对偶问题重写为

$$\max_{\boldsymbol{\alpha}\in\mathbb{R}^m,\boldsymbol{\alpha}\geq 0}\left(\frac{1}{2}\left|\sum_{i=1}^{m}\alpha_i y_i \boldsymbol{x}_i\right|^2-\sum_{i=1}^{m}\alpha_i\left(1-y_i\left(\sum_j \alpha_j y_j \langle \boldsymbol{x}_j,\boldsymbol{x}_i\rangle\right)\right)\right) \tag{15.10}$$

重拍上式中项的顺序可得对偶问题为

$$\max_{\boldsymbol{\alpha}\in\mathbb{R}^m,\boldsymbol{\alpha}\geq 0}\left(\sum_{i=1}^{m}\alpha_i-\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j \langle \boldsymbol{x}_j,\boldsymbol{x}_i\rangle\right) \tag{15.11}$$

注意到上述对偶问题只与样本间的内积有关而不需要直接访问单个特定样本。这个性质在用核来实现 SVM 时是非常重要的，我们在下一章中会详细讨论。

15.5 用随机梯度下降法实现软 SVM

本节中我们要介绍一种非常简单的算法求解软 SVM 优化问题，即

$$\min_{\boldsymbol{w}}\left(\frac{\lambda}{2}\|\boldsymbol{w}\|^2+\frac{1}{m}\sum_{i=1}^{m}\max\{0,1-y\langle \boldsymbol{w},\boldsymbol{x}_i\rangle\}\right) \tag{15.12}$$

[176] 我们根据随机梯度下降法的框架来解该正则损失最小化问题，如 14.5.3 小节中所述。

在式(14.15)中，我们可以将随机梯度下降法的更新规则重写如下：

$$\boldsymbol{w}^{(t+1)}=-\frac{1}{\lambda t}\sum_{j=1}^{t}\boldsymbol{v}_j$$

其中 $\boldsymbol{v}_j$ 是损失函数于第 j 步迭代随机选择样本后在 $\boldsymbol{w}^{(j)}$ 的次梯度。对于合页损失，给定一个样本 $(\boldsymbol{x},y)$，若 $y\langle \boldsymbol{w}^{(j)},\boldsymbol{x}\rangle\geq 1$，我们可以选 $\boldsymbol{v}_j$ 为 $\boldsymbol{0}$，否则，选 $\boldsymbol{v}_j=-y\boldsymbol{x}$（见练习 14.2）。记 $\boldsymbol{\theta}^{(t)}=-\sum_{j<t}\boldsymbol{v}_j$，可得如下程序：

解软 SVM 的随机梯度下降法

求解：式(15.12)
参数：T
初始化：$\boldsymbol{\theta}^{(1)}=0$
for $t=1,\cdots,T$
　令 $\boldsymbol{w}^{(t)}=\frac{1}{\lambda t}\boldsymbol{\theta}^{(t)}$
　随机从 $[m]$ 中均匀地选择 i
　如果 $(y_i\langle \boldsymbol{w}^{(t)},\boldsymbol{x}_i\rangle<1)$
　令 $\boldsymbol{\theta}^{(t+1)}=\boldsymbol{\theta}^{(t)}+y_i\boldsymbol{x}_i$
　否则 $\boldsymbol{\theta}^{(t+1)}=\boldsymbol{\theta}^{(t)}$
输出：$\overline{\boldsymbol{w}}=\frac{1}{T}\sum_{t=1}^{T}\boldsymbol{w}^{(t)}$

15.6 小结

SVM 是在给定先验知识形式(即选择大的间隔)下学习半空间的一个算法。硬 SVM 寻找用大间隔完美分开样本的半空间，而软 SVM 不对数据可分性作假设，而是允许限制条件有一定程度放松。两种形式的 SVM 的样本复杂度与直接学习半空间的样本复杂度是不同的，这是由于其不依赖域的维度而是依赖参数(如 x 的最大范数或者 w 等)。

在下一章将会看到不依赖于维度的样本复杂度是非常重要的，我们将会讨论把给定的域嵌入到高维特征空间作为扩充假设类的方式。这样的扩充面临着计算复杂度和样本复杂度的问题。后者可用 SVM 来解决，前者可以用核的 SVM 来解决，这一点我们将在下一章详述。

15.7 文献评注

Cortes 和 Vapnik(1992)，Boser、Guyor 和 Vapnik(1992)介绍过 SVM。在关于 SVM 的理论和应用方面有很多好书。例如，Vapnik(1995)，Cristianini & Shawe-Taylor (2000)，Schölkopf & Smola(2002)，Hsu 等(2003)，Steinwart 和 Christmann(2008)。采用随机梯度下降来解软 SVM 由 Shalev-Shwartz 等在 2007 年提出。 [177]

15.8 练习

15.1 请说明硬 SVM 规则，即

$$\underset{(w,b):\|w\|=1}{\operatorname{argmax}} \min_{i\in[m]} |\langle w,x_i\rangle + b| \quad \text{s.t.} \quad \forall i, y_i(\langle w,x_i\rangle + b) > 0$$

等价于下式：

$$\underset{(w,b):\|w\|=1}{\operatorname{argmax}} \min_{i\in[m]} y_i(\langle w,x_i\rangle + b) \tag{15.13}$$

提示：定义 $\mathcal{G}=\{(w, b): \forall i, y_i(\langle w, x_i\rangle + b)>0\}$。

1) 说明

$$\underset{(w,b):\|w\|=1}{\operatorname{argmax}} \min_{i\in[m]} y_i(\langle w,x_i\rangle + b) \in \mathcal{G}$$

2) 说明，$\forall (w, b)\in\mathcal{G}$，

$$\min_{i\in[m]} y_i(\langle w,x_i\rangle + b) = \min_{i\in[m]} |\langle w,x_i\rangle + b|$$

15.2 **间隔与感知器** 考虑一个由间隔 γ 线性可分的训练集，因此所有的样本都在一个半径为 ρ 的球中。请证明在 9.1.2 小节中给出的批量感知器算法在该训练集上运行将会作出的最大迭代数是 $(\rho/\gamma)^2$。

15.3 **硬和软 SVM** 证明或推翻如下论断：

存在 $\lambda>0$ 使得对于每个由 $m>1$ 个样本组成的样本集 S(S 可由齐次半空间分开)，硬 SVM 和软 SVM(参数为 λ)学习规则给出的权重向量相同。

15.4 **弱对偶性** 请证明对于任何关于两个向量 $x\in\mathcal{X}$，$y\in\mathcal{Y}$ 的函数 f，不式成立：

$$\min_{x\in\mathcal{X}} \max_{y\in\mathcal{Y}} f(x,y) \geq \max_{y\in\mathcal{Y}} \min_{x\in\mathcal{X}} f(x,y)$$

[178]

核 方 法

前一章我们叙述了 SVM 模型用于学习高维特征空间中的半空间。通过首先将数据映射到高维特征空间，然后在此空间中学习线性预测器，使得半空间的表达能力更加丰富。这与在基空间中学习半空间的线性组合的 AdaBoost 算法类似。尽管这种方式使得半空间预测器的表达能力得到了极大的提升，它同时也带来了样本复杂度及计算复杂度的挑战。前一章我们通过间隔(margin)的概念解决了样本复杂度的问题。本章中，我们将通过核方法解决计算复杂度带来的挑战。

本章我们以数据到高维特征空间映射的思想为开端，进而介绍核的思想。核是样本相似性的一种度量。核相似性的特点在于它可以看作样本映射到的虚拟空间希尔伯特空间(或者高维欧式空间)的内积。我们会介绍使得学习算法计算高效执行，而不必直接处理样本高维空间表示的“核技巧”。基于核的学习算法，尤其是核支持向量机(kernel SVM)，是非常有效且流行的机器学习工具。它们的成功归因于灵活易得的领域先验知识，以及成型的高效快速执行算法。

16.1 特征空间映射

半空间的表达能力非常受限。例如，以下训练集对于半空间是不可分的。

假设定义域为实线：考虑定义点$\{-10, -9, -8, \ldots, 0, 1, \ldots, 9, 10\}$，其中$|x|>2$的$x$的标签为$+1$，其余的为$-1$。

为了使半空间类描述能力更强，我们首先将原始实例空间映射到另一空间(可能是一个高维空间)并且在此空间中学习一个半空间。例如，考虑前面提到的样本。我们首先定义一个映射$\psi: \mathbb{R}\to\mathbb{R}^2$，而不在原始表示下学习半空间，其中

170

$$\psi(x) = (x, x^2)$$

我们用特征空间来表示ψ的值域。应用ψ之后，数据就可以很容易地利用半空间$h(x)=\mathrm{sign}(\langle w, \psi(x)\rangle - b)$来解释，其中$w=(0, 1)$，$b=5$。

基本范式描述如下：

1. 给定定义域$\mathcal{X}$及学习任务，选择映射$\psi: \mathcal{X}\to\mathcal{F}$，特征空间$\mathcal{F}$通常是关于$n$的$n$维实数空间$\mathbb{R}^n$(但是，映射的值域可以是任意希尔伯特空间，包括无限空间，后面我们将会说明)。

2. 给定已标记的样本集，$S=(x_1, y_1), \ldots, (x_m, y_m)$，建立映射序列$\hat{S}=(\psi(x_1), y_1), \ldots, (\psi(x_m), y_m)$。

3. 在$\hat{S}$上训练线性预测器h。

4. 预测测试样本x的标签$h(\psi(x))$。

需要指出的是，对于任意$\mathcal{X}\times\mathcal{Y}$上的概率分布$\mathcal{D}$，通过设定对任意子集$A\subseteq\mathcal{F}\times\mathcal{Y}$，$\mathcal{D}^\psi(A)=\mathcal{D}(\psi^{-1}(A))$，我们可以定义它在$\mathcal{F}\times\mathcal{Y}$上的映射概率分布$\mathcal{D}^\psi$。[1]接下来就可以得到

[1] 这个定义对对任意的A，使得$\psi^{-1}(A)$对于$\mathcal{D}$是可测的。

特征空间上的每一个预测器 h，$L_{P\psi}(h)=L_P(h\circ\psi)$，其中 $h\circ\psi$ 表示 ψ 上 h 的集合。

这个范式的有效性取决于对于给定任务选取好的映射 ψ：也就是，ψ 使得在特征空间中(近乎)线性可分的数据分布的映射，对于给定任务算法是一个好的学习器的映射。这样的一个映射的选取依赖于给定任务的先验，然而，通常应用一些通用的可提高半空间类表达能力的映射。值得一提的一个例子就是多项式映射，它是前面我们看到的 ψ 的推广。

我们知道，对于实例 x，标准的半空间分类器的预测依基于线性映射 $x\mapsto\langle w, x\rangle$。我们可以将线性映射泛化为多项式映射 $x\mapsto p(x)$，其中 p 是 k 阶多元多项式。简单起见，考虑 x 是一维的情况。在此情况下，$p(x)=\sum_{j=0}^{k} w_j x^j$，其中 $w\in\mathbb{R}^{k+1}$ 是我们要学习的多项式系数向量。将 $p(x)$ 重新记为 $p(x)=\langle w, \psi(x)\rangle$，其中 $\psi:\mathbb{R}\to\mathbb{R}^{k+1}$ 是映射 $x\mapsto(1, x, x^2, x^3, \cdots, x^k)$。也就是说，在 $\mathbb{R}$ 上学习一个 k 阶多项式可以通过在 $k+1$ 维特征空间中学习一个线性映射实现。

更加一般地，从 $\mathbb{R}^n$ 到 $\mathbb{R}$ 的一个 k 阶多元多项式可记作

$$p(x)=\sum_{J:|J|\le k} w_J \prod_{i=1}^{n} x_i^{J_i} \tag{16.1}$$

跟前面一样，我们可以将 $p(x)$ 重新记为 $p(x)=\langle w, \psi(x)\rangle$，而现在 $\psi:\mathbb{R}^n\to\mathbb{R}^r$ 使得对于任意的 $J\in[n]^r$，$r\le k$，与 J 关联的 $\psi(x)$ 的坐标是单项式 $\prod_{i=1}^{r} x_{J_i}$。 182

当然，基于多项式的分类器假设类比半空间丰富。本章的开始我们已经看过这样一个例子，训练集在原始空间($\mathcal{X}=\mathbb{R}$)由半空间是不可分的，但经过映射 $x\mapsto(x, x^2)$ 之后，则是完全可分的。因此，尽管分类器在特征空间里总是线性的，但在样本采样的原始空间却有极强的非线性。

一般情况下，我们可以选取任意的特征映射 ψ 使得原始样本映射到某些希尔伯特空间。[⊖] 对任意有限的 d，欧式空间 $\mathbb{R}^d$ 是希尔伯特空间。但也有无穷维希尔伯特空间(本章后面我们会看到)。

这里我们要讨论的主旨就是通过首先应用非线性映射 ψ，将样本空间映射到特征空间，然后在这个特征空间里学习一个半空间，使得半空间的表达能力得到提升。然而，如果映射 ψ 的值域为高维空间，我们就会遇到两个问题。首先，n 维空间 $\mathbb{R}^n$ 里半空间的 VC 维为 $n+1$，因此，如果映射 ψ 的值域非常大，我们就需要非常多的样本来学习 ψ 值域里的半空间。其次，从计算量的角度看，高维空间里进行运算可能代价非常高。事实上，向量 w 在特征空间里可能是不可表示的。第一个问题可以通过应用最大间隔(或者小范数预测器)来解决，我们在前一章 SVM 算法的内容里面已经讨论过。接下来，我们就考虑计算复杂度的问题。

16.2 核技巧

我们已经看到将输入空间映射到高维特征空间可使得半空间学习的表达能力更强。然

⊖ 希尔伯特空间是一个具有内积的向量空间，它是一个完备空间。如果空间中所有的柯西序列收敛，那么称这个空间是完备的。在我们的例子中，范数 $\|w\|$ 由内积 $\sqrt{\langle w, w\rangle}$ 定义。我们之所以要求映射 ψ 的值域是希尔伯特空间是因为希尔伯特空间的映射已有完美的定义。更特殊一点，如果 M 是一个线性希尔伯特空间，那么希尔伯特空间中的任意一个 x 都可记作 $x=w+u$，其中 $u\in M$，并且对于任意的 $w\in M$，$\langle u, w\rangle=0$。下一节证明表示定理的时候，我们会用到这个事实。

面，学习的计算复杂度的问题可能还是一个严重障碍——在非常高维数的空间里学习线性分类器可能计算量非常大。这个问题通用的解决方案就是基于核的学习算法。这里"核"的概念用于描述特征空间的内积。给定由定义域$\mathcal{X}$到希尔伯特空间的特征映射ψ，定义核函数为$K(x, x')=\langle\psi(x), \psi(x')\rangle$。我们也可以将$K$看作度量样本相似性的特殊形式，也可以将$\psi$看作从定义域$\mathcal{X}$到由内积实现相似性的空间的映射。实际上，许多半空间学习算法可以仅仅通过定义域上点对的核函数值来完成。这些算法最主要的优势在于它们可以在高维特征空间中实现线性分类器而不必知道样本点在特征空间中的具体形式或者映射的表
[181] 达式。本节接下来的部分致力于构建这样的算法。

在前一章我们看到尽管特征空间维数非常高，正则化w的范数可以降低样本复杂度。更有趣的是，正如后面我们会说明的，正则化w的范数对解决计算复杂度问题也是非常有帮助的。首先，前一章我们提出，所有版本的SVM优化都解决以下这样一个通用的问题：

$$\min_{w}\left(f(\langle w, \psi(x_1)\rangle, \ldots, \langle w, \psi(x_m)\rangle)+R(\|w\|)\right) \tag{16.2}$$

其中，$f:\mathbb{R}^m\to\mathbb{R}$是任意的函数，$R:\mathbb{R}_+\to\mathbb{R}$是单调不减函数。例如，对于齐次半空间软SVM（等式(15.6)）可以通过使等式(16.2)的$R(a)=\lambda a^2$且$f(a_1, \ldots, a_m)=\frac{1}{m}\sum_i \max\{0, 1-y_i a_i\}$而得。同样，非齐次半空间（等式(15.2)）可通过使等式(16.2)的$R(a)=a^2$且对于任意的i如果存在b使得$y_i(a_i-b)\geq 1$那么$f(a_1, \ldots, a_m)$等于0，反之$f(a_1, \ldots, a_m)=\infty$。

下面的定理证明了在展开空间$\{\psi(x_1), \ldots, \psi(x_m)\}$中存在等式(16.2)的最优解。

定理16.1（表示定理）　假定ψ是由$\mathcal{X}$到希尔伯特空间的映射。那么，存在向量$\boldsymbol{\alpha}\in\mathbb{R}^m$使得$w=\sum_{i=1}^m \alpha_i\psi(x_i)$是等式(16.2)的最优解。

证明　设w^*是等式(16.2)的最优解。由于w^*是希尔伯特空间的元素，我们可以将其写为

$$w^*=\sum_{i=1}^m \alpha_i\psi(x_i)+u$$

其中对于任意的i有$\langle u, \psi(x_i)\rangle=0$。设$w=w^*-u$。显然，$\|w^*\|^2=\|w\|^2+\|u\|^2$。因此，$\|w\|\leq\|w^*\|$。由于$R$是不减的，我们有$R(\|w\|)\leq R(\|w^*\|)$。另外，对于任意的$i$有

$$y_i\langle w, \psi(x_i)\rangle=y_i\langle w^*-u, \psi(x_i)\rangle=y_i\langle w^*, \psi(x_i)\rangle$$

因此，

$$f(y_1\langle w, \psi(x_1)\rangle, \ldots, y_m\langle w, \psi(x_m)\rangle)=f(y_1\langle w^*, \psi(x_1)\rangle, \ldots, y_m\langle w^*, \psi(x_m)\rangle)$$

我们已经说明目标函数式(16.2)在w处的值不大于其在w^*处的值，因此，w也是一个最优解。由于$w=\sum_{i=1}^m \alpha_i\psi(x_i)$，我们就证明了以上定理。 ■

在表示定理的基础之上，我们就可以按照如下以$\boldsymbol{\alpha}$的系数向量w最优化等式(16.2)。记$w=\sum_{i=1}^m \alpha_i\psi(x_i)$，对于任意的$j$我们有

[182]
$$\langle w, \psi(x_j)\rangle=\langle\sum_i \alpha_i\psi(x_i), \psi(x_j)\rangle=\sum_{i=1}^m \alpha_i\langle\psi(x_i), \psi(x_j)\rangle$$

同样，

$$\|w\|^2=\left(\sum_i \alpha_i\phi(x_i),\sum_j \alpha_j\phi(x_j)\right)=\sum_{i,j=1}^{m}\alpha_i\alpha_j(\phi(x_i),\phi(x_j))$$

设 $K(x,x')=(\phi(x),\phi(x'))$ 是特征映射 ϕ 应用的核函数。我们可以解下面这个等价问题而不是等式(16.2)

$$\min_{\alpha\in\mathbb{R}^m} f\left(\sum_{i=1}^{m}\alpha_i K(x_i,x_1),\cdots,\sum_{i=1}^{m}\alpha_i K(x_i,x_m)\right)+R\left(\sqrt{\sum_{i,j=1}^{m}\alpha_i\alpha_j K(x_j,x_i)}\right) \tag{16.3}$$

要解等式(16.3)的优化问题，我们不必用到特征空间里的元素。唯一需要知道的就是怎样计算特征空间的内积，或者说，计算核函数。事实上，要解等式(16.3)，我们只需知道一个 $m\times m$ 的矩阵 G 的值，使得 $G_{i,j}=K(x_i,x_j)$，G 通常称作 Gram 矩阵。

特殊地，具体到前面等式(15.6)给出的软 SVM 问题，可以将问题重新记为

$$\min_{\alpha\in\mathbb{R}^m}\left(\lambda\boldsymbol{\alpha}^\top G\boldsymbol{\alpha}+\frac{1}{m}\sum_{i=1}^{m}\max\{0,1-y_i(G\boldsymbol{\alpha})_i\}\right) \tag{16.4}$$

其中 $(G\boldsymbol{\alpha})_i$ 是 Gram 矩阵与向量 $\boldsymbol{\alpha}$ 乘积向量的第 i 个元素。注意到等式(16.4)可以写成二次规划的形式，因此可以高效快速地解决。下一节我们会介绍一个用核解决软 SVM 的更加简单的算法。

只要学到系数 $\boldsymbol{\alpha}$，我们就可以对一个新样本进行预测

$$\langle w,\phi(x)\rangle=\sum_{i=1}^{m}\alpha_i(\phi(x_i),\phi(x))=\sum_{i=1}^{m}\alpha_i K(x_i,x)$$

利用核而不是直接在特征空间里优化 w 的优势就在于某些情况下，特征空间的维数是非常高的，而利用核函数则非常简单。下面给出了几个例子。

例 16.1（多项式核） k 阶多项式核定义为

$$K(x,x')=(1+\langle x,x'\rangle)^k$$

现在我们说明这确实是一个核函数。也就是说，我们要说明，存在一个由原始空间到高维空间的映射 ϕ 使得 $K(x,x')=(\phi(x),\phi(x'))$。简单起见，设 $x_0=x'_0=1$，那么我们有 183

$$\begin{aligned}K(x,x')&=(1+\langle x,x'\rangle)^k=(1+\langle x,x'\rangle)\cdots\cdots(1+\langle x,x'\rangle)\\&=\left(\sum_{j=0}^{n}x_jx'_j\right)\cdots\cdots\left(\sum_{j=0}^{n}x_jx'_j\right)\\&=\sum_{J\in\{0,1,\cdots,n\}^k}\prod_{j=1}^{k}x_{J_j}x'_{J_j}=\sum_{J\in\{0,1,\cdots,n\}^k}\prod_{j=1}^{k}x_{J_j}\prod_{j=1}^{k}x'_{J_j}\end{aligned}$$

如果我们定义 $\phi:\mathbb{R}^n\to\mathbb{R}^{(n+1)^k}$ 使得对于 $J\in\{0,1,\cdots,n\}^k$，$\phi(x)$ 里面有元素为 $\prod_{j=1}^{k}x_{J_j}$，我们就可得到

$$K(x,x')=(\phi(x),\phi(x'))$$

由于 ϕ 包含所有 k 阶单项式，原始空间里的 k 阶多项式就对应于映射 ϕ 空间里的半空间。因此，利用 k 阶多项式核学习半空间就使得我们可以在原空间里学习一个 k 阶多项式预测器。 ◀

这里我们需要注意的是应用核函数 K 的复杂度为 $O(n)$，而特征空间的维数大约为 n^k。

例 16.2（高斯核） 设原空间为 $\mathbb{R}$，考虑这样的一个映射 ϕ：对于任意非负整数 $n\geq 0$，存在元素 $\phi(x)_n=\frac{1}{\sqrt{n!}}e^{-\frac{x^2}{2}}x^n$。那么，

$$\langle\psi(x),\psi(x')\rangle=\sum_{n=1}^{\infty}\left(\frac{1}{\sqrt{n!}}e^{-\frac{\|x\|^2}{2}}x^n\right)\left(\frac{1}{\sqrt{n!}}e^{-\frac{\|x'\|^2}{2}}(x')^n\right)$$
$$=e^{-\frac{\|x\|^2+\|x'\|^2}{2}}\sum_{n=1}^{\infty}\left(\frac{(x,x')^n}{n!}\right)=e^{-\frac{\|x-x'\|^2}{2}}$$

这里特征空间是无穷维，但应用核是非常简单地。更一般地，给定标量 $\sigma>0$，高斯核定义为

$$K(x,x')=e^{-\frac{\|x-x'\|^2}{\sigma}}$$

直观地，如果两个样本 x，x' 彼此距离远（原始空间），那么高斯核使得特征空间里二者的内积接近于 0，相反如果原空间距离近，内积则接近于 1。σ 是用来决定怎样意味着"近"的参数。很容易证明 K 是某一空间的内积，在这个空间里，对于任意的 n 及任意的 k 阶单项式都存在元素 $\psi(x)$ 等于 $\frac{1}{\sqrt{n!}}e^{-\frac{\|x\|^2}{2}}\prod_{i=1}^{k}x_{j_i}$。因此，通过利用高斯核，我们可以学得在原空间的任意多项式预测器。 ◀

[184]

我们知道所有多项式预测器类的 VC 维是有限的（见练习 16.12），由于学习高斯核所需的样本复杂度依赖于特征空间的间隔，如果够幸运的话间隔会比较大，但通常 margin 会比较小，然而这二者并不矛盾。

高斯核也称 RBF 核，即"Radial Basis Functions"。

16.2.1 核作为表达先验的一种形式

正如前面我们所讨论的，特征映射 ψ 可以看作线性分类器到表达能力更加丰富的类（对应于特征空间里的线性分类器）的扩展。然而，到目前本书讨论的内容为止，给定任务的任意假设类的有效性取决于任务的本身特性，因此我们也可以将映射 ψ 看作对当前问题表达利用先验知识的一种方式。例如，如果我们相信正样本可以由一些椭圆形区分，就可以定义 ψ 是所有二阶单项式或者 2 阶多项式核。

举一个更实际的例子，考虑学习找到文件里的序列字符（"签字"）用于指示其是否含有病毒。一般地，设 $\mathcal{X}$ 为字母集 Σ 里的所有有限字符串组成的集合，并且 $\mathcal{X}_d$ 是所有长度最大为 d 的字符串的集合。我们期望学到的假设为 $\mathcal{H}=\{h_v:v\in\mathcal{X}_d\}$，使得对于字符串 $x\in\mathcal{X}$，当且仅当 v 是 x 的子串时，$h_v(x)=1$（反之 $h_v(x)=-1$）。接下来我们说明，如何应用一个合适的映射，使得这个假设可以通过在特征空间里学习一个线性分类器完成。考虑到特征空间里的映射 ψ，其中 $s=|\mathcal{X}_d|$，因此，$\psi(x)$ 的每个坐标与字符串 v 对应，并且表明 v 是否是 x 的一个子串（也就是说，对于任意的 $x\in\mathcal{X}$，$\psi(x)$ 是 $\{0,1\}^{|\mathcal{X}_d|}$ 里的一个向量）。需要指出的是，特征空间的维数与 d 呈指数关系。不难看出，类 $\mathcal{H}$ 的每一个元素都可以通过 $\psi(x)$ 上的线性分类器组合而得，此外，可以通过范数为 1 的半空间而得，这样可使得间隔为 1（见练习 16.1）。进一步说，对于任意的 $x\in\mathcal{X}$，$\|\psi(x)\|=O(\sqrt{d})$。总的来说，样本复杂度为与 d 相关的多项式时，用 SVM 是可学习的。然而，特征空间的维数与 d 呈指数关系，因此直接在特征空间里应用 SVM 是不切实际的。幸运的是，计算特征空间的内积（例如核函数）是容易的，而不必知道特征映射后样本的具体形式。事实上，$K(x,x')$ 就是 x 和 x' 公共子串的数目，这可以在与 d 相关的多项式时间内很容易地计算出。

这个例子也说明了特征映射是怎样使得我们可以在非矢量域里应用半空间。

16.2.2 核函数的特征*

正如前一节讨论的，我们可以将核函数看作表达先验知识的形式。考虑给定的相似性函数 $K: \mathcal{X}\times\mathcal{X}\to\mathbb{R}$，它是否是一个合法的核函数？也就是说，对于特征映射 ψ 它是否表示 $\psi(x)$ 与 $\psi(x')$ 的内积？以下定理给出了充分必要条件。 185

定理 16.2 一个对称的函数 $K: \mathcal{X}\times\mathcal{X}\to\mathbb{R}$ 对应希尔伯特空间的内积，当且仅当它是半正定的；也就是说，对于所有的 $x_1, \cdots, x_m$，Gram 矩阵 $G_{i,j}=K(x_i, x_j)$ 是一个半正定矩阵。

证明 显然如果 K 是希尔伯特空间的内积，那么 Gram 矩阵就是半正定的。反过来，我们首先定义 $\mathcal{X}$ 上的函数空间 $\mathbb{R}^{\mathcal{X}}=\{f: \mathcal{X}\to\mathbb{R}\}$。对于任意的 $x\in\mathcal{X}$，设 $\psi(x)$ 是函数 $x\mapsto K(\cdot, x)$。通过所有具有 $K(\cdot, x)$ 形式元素的线性组合，我们可以定义一个向量空间。定义这个向量空间的内积为

$$\langle \sum_i \alpha_i K(\cdot, x_i), \sum_j \beta_j K(\cdot, x_j') \rangle = \sum_{i,j} \alpha_i \beta_j K(x_i, x_j')$$

由于它是对称的（K 是对称的），因此它是一个合法的内积，它是线性的（显然）。并且半正定（容易看出 $K(x, x)\geq 0$，仅有 $\psi(x)=0$ 时等于 0）。显然，

$$\langle \psi(x), \psi(x') \rangle = \langle K(\cdot, x), K(\cdot, x') \rangle = K(x, x')$$

以上定理得证。 ■

16.3 软 SVM 应用核方法

接下来我们用核方法处理软 SVM。尽管我们已经设计算法解决等式(16.4)的问题，仍仍有更简单的方法直接在特征空间里解决软 SVM 的优化问题。

$$\min_{w}\left(\frac{\lambda}{2}\|w\|^2+\frac{1}{m}\sum_{i=1}^{m}\max\{0, 1-y\langle w, \psi(x_i)\rangle\}\right) \tag{16.5}$$

并且只利用核演化。基础就是我们在 15.5 节介绍的 SGD 得到的向量 w^t 总是存在于 $\{\psi(x_1), \cdots, \psi(x_m)\}$ 的线性展开空间。因此，我们可以计算对应系数 $\boldsymbol{\alpha}$ 而不是 $w^{(t)}$。

正式地，设 K 是核函数，也就是说，对所有的 x, x'，$K(x, x')=\langle \psi(x), \psi(x') \rangle$。我们要考虑 $\mathbb{R}^m$ 里的两个向量，对应于 15.5 节里 SGD 的 $\boldsymbol{\theta}^{(t)}$ 和 $w^{(t)}$。也就是说，$\boldsymbol{\beta}^{(t)}$ 是一个向量，使得

$$\boldsymbol{\theta}^{(t)}=\sum_{j=1}^{m}\beta_j^{(t)}\psi(x_j) \tag{16.6}$$

向量 $\boldsymbol{\alpha}^{(t)}$ 使得

$$w^{(t)}=\sum_{j=1}^{m}\alpha_j^{(t)}\psi(x_j) \tag{16.7}$$

向量 $\boldsymbol{\alpha}$ 和 $\boldsymbol{\beta}$ 的更新按照以下流程。 186

SGD 解带核函数的软 SVM

目标：解等式(16.5)
参数：T
初始化：$\boldsymbol{\beta}^{(1)}=\mathbf{0}$

for $t=1, \dots, T$

设 $\boldsymbol{\alpha}^{(t)}=\frac{1}{\lambda t}\boldsymbol{\beta}^{(t)}$

从 $[m]$ 里随机均匀选取 i

对所有的 $j\neq i$，设 $\beta_j^{(t+1)}=\beta_j^{(t)}$

If $y_i\sum_{j=1}^{m}\alpha_j^{(t)}K(\boldsymbol{x}_j,\boldsymbol{x}_i)<1$

　　设 $\beta_i^{(t+1)}=\beta_i^{(t)}+y_i$

Else

　　设 $\beta_i^{(t+1)}=\beta_i^{(t)}$

输出：$\overline{\boldsymbol{w}}=\sum_{j=1}^{m}\overline{\alpha}_j\psi(\boldsymbol{x}_j)$，其中 $\overline{\boldsymbol{\alpha}}=\frac{1}{T}\sum_{t=1}^{T}\boldsymbol{\alpha}^{(t)}$

下面的引理说明前面的这个实现等价于 15.5 节描述的在特征空间里运行 SGD 的流程。

引理 16.3　设 $\hat{\boldsymbol{w}}$ 是 15.5 节描述的在特征空间里应用 SGD 流程的输出，$\overline{\boldsymbol{w}}=\sum_{j=1}^{m}\overline{\alpha}_j\psi(\boldsymbol{x}_j)$ 是应用核函数 SGD 流程的输出，那么 $\overline{\boldsymbol{w}}=\hat{\boldsymbol{w}}$。

证明　我们会说明对于任意的 t 等式(16.6)成立，其中 $\boldsymbol{\theta}^{(t)}$ 是在特征空间里应用 SGD 算法输出的结果。根据定义，$\boldsymbol{\alpha}^{(t)}=\frac{1}{\lambda t}\boldsymbol{\beta}^{(t)}$，$\boldsymbol{w}^{(t)}=\frac{1}{\lambda t}\boldsymbol{\theta}^{(t)}$，这就说明等式(16.7)是成立的。接下来继续我们的证明。为了证明等式(16.6)成立，我们用一个简单的归纳证明。对于 $t=1$，等式显然成立。假设当 $t\geq 1$ 时等式成立，那么

$$y_i\langle \boldsymbol{w}^{(t)},\psi(\boldsymbol{x}_i)\rangle=y_i\langle\sum_j\alpha_j^{(t)}\psi(\boldsymbol{x}_j),\psi(\boldsymbol{x}_i)\rangle=y_i\sum_{j=1}^{m}\alpha_j^{(t)}K(\boldsymbol{x}_j,\boldsymbol{x}_i)$$

因此，两个算法的条件是等价的，如果我们更新 $\boldsymbol{\theta}$，那么有

$$\boldsymbol{\theta}^{(t+1)}=\boldsymbol{\theta}^{(t)}+y_i\psi(\boldsymbol{x}_i)=\sum_{j=1}^{m}\beta_j^{(t)}\psi(\boldsymbol{x}_j)+y_i\psi(\boldsymbol{x}_i)=\sum_{j=1}^{m}\beta_j^{(t+1)}\psi(\boldsymbol{x}_j)$$ ■

16.4　小结

从定义域映射到高维空间，在高维空间里应用的半空间预测器具有很强的表达能力。一方面我们受益于丰富且复杂的假设类，但也要解决样本复杂度和计算复杂度带来的困难。在第 10 章，我们讨论了 AdaBoost 算法，它应用弱学习器的同时也面临许多挑战；尽管在高维空间中处理问题，但在每次迭代中我们都会学得一个效果比较好的坐标。本章我们介绍了一种不同的方法，核技巧。想法是，为在高维空间中学得一个半空间预测器，我们不必知道样本在此空间的具体表达形式，而只需知道样本映射之后内积的值。通过核函数，计算高维空间里样本之间的内积就不需要知道样本的具体表达形式。我们也介绍了如何将核函数应用到 SGD 算法中。

[167]

特征映射及核技巧的思想使得我们可以对非向量数据应用半空间及线性预测器的框架。我们也介绍了如何利用核函数在字符串域里学习预测器。

我们说明了核技巧对于SVM的有效性。当然，核技巧可以应用于其他很多算法，练习中给出了一些例子。

线性预测器及凸问题这一系列的章节以本章为结尾。接下来的两章将会介绍完全不同类型的假设类。

16.5 文献评注

在SVM的背景下，核技巧由Boser等人引入(1992)，也可参见Aizerman等人(1964)。Scholkopf等人最早提出核技巧可以用于任何仅依赖于内积的算法(1998)。表示定理的证明由Schölkopf等人(2000)和Scholkopf等人(2001)给出。引理16.2的条件是Mercer定理的简单形式。许多文献提出了各种各样应用的核函数。读者可参阅Scholkopf和Smola(2002)。

16.6 练习

16.1 考虑16.2.1节描述的在文件里寻找字符串的任务。证明类$\mathcal{H}$里的每个元素都可由$\psi(x)$上的线性分类器组合而成，并且它们的范数为1，间隔为1。

16.2 核化感知器：说明仅知道经过核函数之后的样本时，如何运行感知器算法。

提示：将生算法等同于核函数应用到SGD算法中。

16.3 核岭回归：带有特征映射ψ的岭回归问题就是要寻找一个向量w使得以下函数值最小

$$f(w)=\lambda\|w\|^2+\frac{1}{2m}\sum_{i=1}^{m}(\langle w,\psi(x_i)\rangle-y_i)^2 \tag{16.8}$$

然后返回预测器

$$h(x)=\langle w,x\rangle$$

说明如何将核函数应用到岭回归算法中。

提示：表示定理告诉我们存在向量$\boldsymbol{\alpha}\in\mathbb{R}^m$使得$\sum_{i=1}^{m}\alpha_i\psi(x_i)$是等式(16.8)的解。 188

1) 设G是关于S和K的Gram矩阵，也就是说，$G_{i,j}=K(x_i,x_j)$。定义$g:\mathbb{R}^m\to\mathbb{R}$

$$g(\boldsymbol{\alpha})=\lambda\cdot\boldsymbol{\alpha}^{\mathrm{T}}G\boldsymbol{\alpha}+\frac{1}{2m}\sum_{i=1}^{m}(\langle\boldsymbol{\alpha},G_{:,i}\rangle-y_i)^2 \tag{16.9}$$

其中$G_{:,i}$是G的第i列。证明如果$\boldsymbol{\alpha}^*$使得等式(16.9)最小，那么$w^*=\sum_{i=1}^{m}\alpha_i^*\psi(x_i)$是$f$的解。

2) 寻找$\boldsymbol{\alpha}^*$的封闭表达式。

16.4 设N是任意正整数。对任意的$x,x'\in\{1,\dots,N\}$定义

$$K(x,x')=\min\{x,x'\}$$

证明K是合法核函数。也就是说，寻找映射$\psi:\{1,\dots,N\}\to H$，其中H是希尔伯特空间，使得

$$\forall x,x'\in\{1,\dots,N\},K(x,x')=\langle\psi(x),\psi(x')\rangle$$

16.5 超市管理员想要根据顾客的购物清单学习预测该顾客是否有小孩。特殊地，他独立同分布地采集了一些顾客样本，对于顾客i，设$x_i\subset\{1,\dots,d\}$表示顾客商品子集，设$y_i\in\{\pm1\}$是表明顾客是否有小孩的标签。作为先验知识，管理员知道一共有k个商品，当且仅当顾客购买了这k个商品里至少一个时，标签为1。当然，这k个

商品具体是什么并不知道(否则，就没有必要学习了)。另外，根据超市规定，每个顾客最多可以购买 s 个商品。帮助管理员设计一种学习算法使得时间复杂度与样本复杂度都是关于 s，k，$1/\varepsilon$ 的多项式。

16.6　设$\mathcal{X}$是样本集，ψ 是将$\mathcal{X}$映射到希尔伯特空间 V 的特征映射。设 K：$\mathcal{X}\times\mathcal{X}\to\mathbb{R}$ 是在特征空间 V 里应用内积的核函数。

考虑根据平均最近的类预测未知样本的二分类算法。正式地，给定训练序列 $S=(x_1, y_1), \cdots, (x_m, y_m)$，对任意的 $y\in\{\pm1\}$，定义

$$c_y=\frac{1}{m_y}\sum_{i:y_i=y}\psi(x_i)$$

其中，$m_y=|\{i: y_i=y\}|$。假定m_+和m_-不为零。那么算法的输出按照如下决策规则：

$$h(x)=\begin{cases}1 & \|\psi(x)-c_+\|\leq\|\psi(x)-c_-\| \\ -1 & \text{其他}\end{cases}$$

1) 设 $w=c_+-c_-$，$b=\frac{1}{2}\left(\|c_-\|^2-\|c_+\|^2\right)$。证明

$$h(x)=\operatorname{sign}(\langle w,\psi(x)\rangle+b)$$

189　2) 说明在不知道 $\psi(x)$ 或 w 元素情况下，如何在核函数的基础上描述 $h(x)$。

多分类、排序与复杂预测问题

多分类是如何将待分类点划归到几个目标类别之中的问题。这就是说，我们的目标是学习一个预测器 $h:\mathcal{X}\to\mathcal{Y}$，其中的$\mathcal{Y}$是一个类别的有限集合。这种分类的应用包括，比如根据文件主题进行相应的分配($\mathcal{X}$是文件的集合，$\mathcal{Y}$是一个可能的文件主题的集合)，或者识别那一个目标产生了相应的图片($\mathcal{X}$是图片的集合而$\mathcal{Y}$是可能的产生目标的集合)。

多分类的中心任务就是多类别机器学习，这也刺激产生了一大批旨在解决该任务的方法。也许最直接的方法就是将多任务的分类转化为二分类问题。在 17.1 节，我们将探讨最普通的两种简化方法，以及它们的主要缺点。

我们之后将描述一个针对多分类的线性预测器问题。利用之前很多章节介绍的 RLM 和 SGD 架构，我们描述了几个实用的针对多分类预测的算法。

在 17.3 节，我们将展示如何使用多分类学习机去处理复杂的预测问题，在这些问题中，$\mathcal{Y}$集合可能非常巨大，但是，可能具有一些结构可以利用。这种学习任务经常被称为结构化输出学习。举一个特别的例子，就识别手写文字的任务来说，这种情况下，$\mathcal{Y}$的集合是特定字段边界长度数值作为变量所产生的所有组合情况(所以$\mathcal{Y}$的大小是根据最大字段长度而指数变化的)。

最后，在 17.4 节和 17.5 节中，我们讨论了在一定情况下，学习者需要将样本集中的点根据它们的“关联性”进行排序的问题。一个典型的应用就是根据搜索问句请求，按照与搜索内容的相关性对搜索结果进行排序的问题。我们描述了几种根据其学习结果相关性来评价预测器的测量标准，并且对于如何利用线性预测器来有效率地解决排序问题进行了介绍。

17.1 一对多和一对一

解决多分类预测的一个最简单的方法就是将其简化为一个二分类问题。回想一下多分类预测，我们所想学习的函数是 $h:\mathcal{X}\to\mathcal{Y}$，没有泛化的损失，我们则标注为$\mathcal{Y}=\{1,\dots,k\}$。在一对多(One-versus-All)(也称为一对其他剩余)的方法中，我们训练 k 个二元的分类器，每一个都产生一类和其他剩余类之间的划分界限。即，给定一个训练集：$S=\{(x_1,y_1),\dots,(x_m,y_m)\}$，其中的每一个 y_i 都在集合$\mathcal{Y}$中，我们建立 k 个二元的训练子集 $S_1,\dots,S_k$，其中$S_i=(x_1,(-1)^{\mathbf{1}_{y_1\neq i}}),\dots,(x_m,(-1)^{\mathbf{1}_{y_m\neq i}})$。用文字表述的话，$S_i$ 是那些在 S 集合中标签为 i 而标注标签值为 1(否则为 -1)的样本点的集合。对于每一个 $i\in[k]$，我们训练一个基于 S_i 集合的二元的预测器 $h_i:\mathcal{X}\to\{\pm1\}$，希望当且仅当 x 属于类别 i 时，$h_i(x)$的输出等于 1。那么，给定 $h_1,\dots,h_k$，我们建立了一个多类别的分类器，应用的规则是

$$h(x)\in\operatorname*{argmax}_{i\in[k]}h_i(x) \tag{17.1}$$

当超过一个二元假设的预测是“1”的时候，某种程度上我们应当决定预测的类别(比如，我们随意地做一个决定，通过选择在$\operatorname*{argmax}_i h_i(x)$)中序数最小的来打断相互连接)，一个更好的方法是，不论哪一个 h_i 隐含了另外的附加信息，都可以在 $y=i$ 的预测中被赏

信。例如，在一个半空间划分的情况下，实际的预测结果是 $\mathrm{sign}(\langle w, x\rangle)$，但是我们可以把 $\langle w, x\rangle$ 当做预测中的置信。在这种情况下，我们可以应用在公式(17.1)中给出的多类别分类，进行实际值的预测。一对多方法的伪代码在下面给出。

一对多

输入：

训练集 $S=(x_1, y_1), \ldots, (x_m, y_m)$

二分类算法 A

for 每个 $i\in\mathcal{Y}$

令 $S_i=(x_1, (-1)^{\mathbf{1}[y_1=i]}), \ldots, (x_m, (-1)^{\mathbf{1}[y_m=i]})$

$h_i=A(S_i)$

输出：

多类假设定义为 $h(x)\in\operatorname{argmax}_{i\in\mathcal{Y}} h_i(x)$

另一个流行的简化是一对一(All-Pairs)的方法，就是把类别的全部成对组合进行相互的比较。正式地，给定一个训练集 $S=(x_1, y_1), \ldots, (x_m, y_m)$，其中的每一个 y_t 都在 $[k]$ 中，对于每一个 $1\le i<j\le k$ 我们建立一个二元的训练序列 $S_{i,j}$，包含来自 S 的全部样本点，标签是 i 或者 j。对于每一个样本，如果多分类中的标签是 i，我们设定在 $S_{i,j}$ 中的标签值是 $+1$，而如果对应的是 j 则标注为 -1。接下来，我们在每一个 $S_{i,j}$ 上训练一个二分类算法来得到 $h_{i,j}$。最后，我们建立一个通过获得最多数量"wins"的类别作为预测的多类别的分类器。一对一方法的伪代码在下面给出。

191

一对一

输入：

训练集 $S=(x_1, y_1), \ldots, (x_m, y_m)$

二分类算法 A

for 各个 $i, j\in\mathcal{Y}$ 且满足 $i<j$

初始化 $S_{i,j}$ 为空序列

for $t=1, \ldots, m$

若 $y_t=i$，加 $(x_t, 1)$ 到 $S_{i,j}$

若 $y_t=j$，加 $(x_t, -1)$ 到 $S_{i,j}$

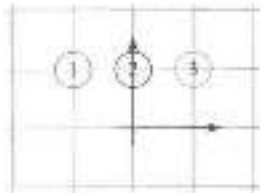
令 $h_{i,j}=A(S_{i,j})$

输出：

多类别假设定义为 $h(x)\in\operatorname{argmax}_{i\in\mathcal{Y}}\left(\sum_{j\in\mathcal{Y}}\mathrm{sign}(j-i)h_{i,j}(x)\right)$

虽然简化的方法如一对多和一对一可以根据现有的算法简单地进行构建。二元的学习器并不能意识到实际上我们准备使用它的假设输出来构建一个多分类的预测器，而这也许会导致并不令人满意的结果，正如在下面的例子中说明的一样。

例 17.1 考虑一个多分类问题，其对应的 $\mathcal{X}=\mathbb{R}^2$，标签集 $\mathcal{Y}=\{1, 2, 3\}$。假设这些不同类别的样本被安放在下边描述的不相交的球形中。



假设属于类别 1，2，3 的大致概率分别是 40%，20%和 40%。考虑应用一对多的方法，并且假设利用该方法进行的二分类是关于假设类别半空间划分的 ERM。观察一下这个问题对于类别 2 和其他类别的分界线，理想的半空间应该是全部标记负的分类器。所以，根据一对多方法的多分类器可能在类别 2 的划分问题上出错(如若对于 $h(x)$ 定义的连接由于数值的类别标签被打破，这种情况就会发生)。对比来说，如果我们选择

$$h_i(x)=\langle w_i,x\rangle,\text{其中 } w_1=\left(-\frac{1}{\sqrt{2}},\frac{1}{\sqrt{2}}\right),w_2=(0,1),w_3=\left(\frac{1}{\sqrt{2}},\frac{1}{\sqrt{2}}\right)$$

那么根据 $h(x)=\operatorname{argmax}_i h_i(x)$ 定义的分类器将完美地预测所有样本，我们可以看到即便来自于 $h(x)=\operatorname{argmax}_i\langle w_i,x\rangle$ 的预测器相对误差近乎 0，一对多方法可能并不能成功地找到一个好的类别预测器。

192

17.2　线性多分类预测

由于简化方法的不完备性，我们在这一节将会学习一个更加直接的多分类预测器。我们将介绍线性多分类预测器族，对于激发我们建立这部分新的方法的原动力，回想一个线性分类器进行二分类(即半空间)的假设由来：$h(x)=\operatorname{sign}(\langle w,x\rangle)$。

下面是对于这个预测的等价描述：

$$h(x)=\operatorname*{argmax}_{y\in\{\pm1\}}\langle w,yx\rangle$$

其中 yx 是把向量 x 中的每个元素乘以 y 得到的新向量。

这种表述将多分类的半空间问题进行了很自然的一般化。令 $\Psi:\mathcal{X}\times\mathcal{Y}\to\mathbb{R}^d$ 为一个类敏感的特征映射。具体来说，Ψ 将一对 (x,y) 作为输入，并将其映射到一个 d 维的特征向量中。直觉地讲，$\Psi(x,y)$ 被看做一个评分函数，可以衡量标签 y 有多么适合样本 x。我们之后将会进一步介绍它。给定 Ψ 和一个向量 $w\in\mathbb{R}^d$，我们可以定义一个多分类的预测器，$h:\mathcal{X}\to\mathcal{Y}$，如下所示：

$$h(x)=\operatorname*{argmax}_{y\in\mathcal{Y}}\langle w,\Psi(x,y)\rangle$$

具体来说，h 根据输入 x 的预测标签获得了最高的权重得分，而这种权重是根据向量 w 来定义的。

令 W 为 $\mathbb{R}^d$ 向量空间中的一些子集，例如，$W=\{w\in\mathbb{R}^d:\|w\|\le B\}$，其中 $B>0$。每一组 (Ψ,W) 定义一个多分类预测器中的假设类：

$$\mathcal{H}_{\Psi,W}=\{x\mapsto\operatorname*{argmax}_{y\in\mathcal{Y}}\langle w,\Psi(x,y)\rangle:w\in W\}$$

当然，我们会很快产生一个待解决的问题，这也是接下来的探讨，如何去构建一个好的 Ψ？注意，如果 $\mathcal{Y}=\{\pm1\}$ 并且有 $\Psi(x,y)=yx$ 和 $W=\mathbb{R}^d$，那么 $\mathcal{H}_{\Psi,W}$ 变成对于二分类的齐次半空间划分假设类。

17.2.1　如何构建 Ψ

正如先前提到的，我们可以把 $\Psi(x,y)$ 看做是一个评价标签 y 是否适合 x 的评分函

数。自然地，设计一个好的 Ψ 就正如设计一个好的特征映射(这和我们在第 16 章所述的，以及在接下来的第 25 章将会进一步讨论的相似)。下面我们将给出两个有效的构建。

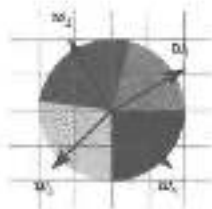
1. 多重矢量构建

令$\mathcal{Y}=\{1,\cdots,k\}$且$\mathcal{X}=\mathbb{R}^n$。我们定义 $\Psi:\mathcal{X}\times\mathcal{Y}\to\mathbb{R}^d$，其中 $d=nk$，具体形式如下：

$$\Psi(x,y)=[\underbrace{0,\cdots,0}_{\in\mathbb{R}^{n(y-1)}},\underbrace{x_1,\cdots,x_n}_{\in\mathbb{R}^n},\underbrace{0,\cdots,0}_{\in\mathbb{R}^{n(k-y)}}] \tag{17.2}$$

193

也就是说，$\Psi(x,y)$由 k 个向量组成，其中的每一个都是 n 维，除了第 y 个向量设为 x，我们把其他全部都定义为零向量。它允许我们把 $w\in\mathbb{R}^d$ 看作是 $\mathbb{R}^n$ 中 k 权重组成的向量，即，$w=[w_1;\cdots;w_k]$，因此称为多重矢量构建。通过构建，我们有$\langle w,\Psi(x,y)\rangle=\langle w_y,x\rangle$，并且多分类预测变成 $h(x)=\underset{y\in\mathcal{Y}}{\operatorname{argmax}}\langle w_y,x\rangle$。多分类预测在$\mathcal{X}=\mathbb{R}^2$上的几何表示如下图所示。



2. TF-IDF

之前对于 $\Psi(x,y)$的定义并没有完全利用关于问题的先验知识。我们接下来将描述一个并不具体表现出先验知识的特征函数 Ψ。令$\mathcal{X}$为关于文件的集合，而$\mathcal{Y}$是其可能的主题的集合。令 d 为对应字的词典的大小。对于字典中的每一个单词，相应的序数为 j，令 $TF(j,x)$为一个词符合序数 j 在文件 x 中所出现的次数。这种量化指标被称为词项频率(Term-Frequency)。另外，规定 $DF(j,y)$是序号为 j 的词在关于文件的训练集中与主题 y 不符合的数量，这种量化指标被称为文档频率(Document-Frequency)，并且衡量序数为 j 的词是否在其他主题中出现频繁。现在，我们定义 $\Psi:\mathcal{X}\times\mathcal{Y}\to\mathbb{R}^d$ 使得 $\Psi_j(x,y)=TF(j,x)\log\left(\frac{m}{DF(j,y)}\right)$，其中 m 是训练集中全部文件的数目。上述量化标准被称为词频逆文档频率，或者简写为 TF-IDF。直觉地讲，如果对应序数 j 的词在许多份文件 x 中出现，$\Psi_j(x,y)$应当得到一个大的结果，而不会出现在不属于主题 y 的全部文件之中。需要注意的是，和之前多重矢量的构建不同，现在的构建中 Ψ 的维度并不取决于主题的数目(换言之，不取决于集合$\mathcal{Y}$的大小)。

17.2.2　对损失敏感的分类

目前为止，我们使用 0—1 损失函数作为 $h(x)$表现的衡量标准。这就是说，假设 h 对于一个样本点(x,y)所产生的损失在 $h(x)\neq y$ 的时候值为 1，反之则为 0。在一些情况中，损失函数应当要对不同的错误有不同的惩罚敏感度。举例来说，在目标识别的任务中，将一张含有老虎的图片识别为猫要比识别为鲸鱼的错误严重程度低一些。这可以通过设计一个具体的损失函数来实现，$\Delta:\mathcal{Y}\times\mathcal{Y}\to\mathbb{R}_+$，其中对于每一组标签 y 和 y'，预测标签 y'而真实标签为 y 所带来的损失可以被定义为 $\Delta(y',y)$。根据这样的假设，$\Delta(y,y)=0$。其实，0—1 损失可以简单地通过设定 $\Delta(y',y)=\mathbb{1}[y'\neq y]$表示。

194

17.2.3 经验风险最小化

我们之前已经定义了假设类$\mathcal{H}_{\Psi,W}$，规定了损失函数Δ。为了学习关于损失函数的类，我们可以应用关于这个类的ERM准则，也就是说，我们可以寻找一个多分类假设$h\in\mathcal{H}_{\Psi,W}$，该多分类假设可以通过一个向量w来表示其参数，这样就可以最小化损失函数Δ的经验风险，

$$L_S(h)=\frac{1}{m}\sum_{i=1}^{m}\Delta(h(x_i),y_i)$$

现在我们要证明的是当$W=\mathbb{R}^d$并且在可实现情况下，利用线性回归算法极有可能有效地解决ERM问题。确实，在可实现情况下，我们确实需要找到向量$w\in\mathbb{R}^d$使其满足

$$\forall i\in[m],y_i=\operatorname*{argmax}_{y\in\mathcal{Y}}\langle w,\Psi(x_i,y)\rangle$$

等价地，w也需要满足以下一组线性不等式：

$$\forall i\in[m],\forall y\in\mathcal{Y}\setminus\{y_i\},\langle w,\Psi(x_i,y_i)\rangle>\langle w,\Psi(x_i,y)\rangle$$

找到满足之前线性等式的w等同于解决了一个线性算法问题。

正如二分类问题一样，我们也可以使用一种通用的感知算法来解决ERM问题，见练习17.2。

在不可实现的情况下，解决ERM问题一般都是计算很难的。我们利用凸优化方法代替损失函数(见12.3节)来解决这个困难。尤其是将合页损失(hinge loss)泛化到多分类问题。

17.2.4 泛化合页损失

回忆一下，在二分类问题中，合页损失被定义为$\max\{0,1-y\langle w,x\rangle\}$。现在我们将合页损失推广到多分类预测，将其表示成以下形式

$$h_w(x)=\operatorname*{argmax}_{y'\in\mathcal{Y}}\langle w,\Psi(x,y')\rangle$$

一个替代的凸损失应该限定原来非凸损失$\Delta(h_w(x),y)$的上界。为了推导关于$\Delta(h_w(x),y)$的上界，我们首先需要注意的是$h_w(x)$的定义意味着

$$\langle w,\Psi(x,y)\rangle\leqslant\langle w,\Psi(x,h_w(x))\rangle$$

因此，

$$\Delta(h_w(x),y)\leqslant\Delta(h_w(x),y)+\langle w,\Psi(x,h_w(x))-\Psi(x,y)\rangle$$

由于$h_w(x)\in\mathcal{Y}$，我们可以通过下面的公式(17.3)来限定右边部分上界

$$\max_{y'\in\mathcal{Y}}(\Delta(y',y)+\langle w,\Psi(x,y')-\Psi(x,y)\rangle)\overset{\triangleq}{=}\ell(w,(x,y)) \tag{17.3}$$

我们用术语"泛化合页损失"来表示之前的叙述。正如之前所说的，$\ell(w,(x,y))\geqslant\Delta(h_w(x),y)$，而且等号表示在任何情况下，正确标记的得分都要比任何其他标签y'的得分大至少$\Delta(y',y)$，也就是说，

$$\forall y'\in\mathcal{Y}\setminus\{y\},\langle w,\Psi(x,y)\rangle\geqslant\langle w,\Psi(x,y')\rangle+\Delta(y',y)$$

很显然，由于$\ell(w,(x,y))$是线性函数w的最大值，所以$\ell(w,(x,y))$是一个关于w的凸函数(见第12章的论断12.5)。$\ell(w,(x,y))$是ρ-利普希茨函数，其中$\rho=\max_{y'\in\mathcal{Y}}\|\Psi(x,y')-\Psi(x,y)\|$。

评注 既然在二分类情况下，我们使用"泛化合页损失"这个名称，当$\mathcal{Y}=\{\pm1\}$时，如果设定$\Psi(x,y)=\frac{yx}{2}$，那么泛化合页损失变成了二分类问题的普通合页损失。

196

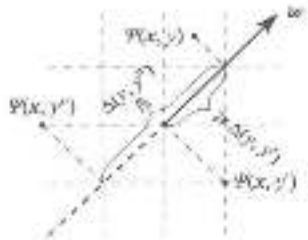
$$\ell(w,(x,y)) = \max\{0, 1-\gamma(w,x)\}$$

直观的几何说明

特征函数 $\Psi: \mathcal{X}\times\mathcal{Y}\to\mathbb{R}^d$ 在 $\mathbb{R}^d$ 维空间中将每个 x 映射到 $|\mathcal{Y}|$ 向量。如果存在一个方向 w，当将 $|\mathcal{Y}|$ 向量映射到这个方向时，每一个向量可以用标量 $\langle w, \Psi(x, y)\rangle$ 来表示，$\ell(w, (x, y))$ 将为零。我们可以基于这些标量来排序不同的点，使得

- 对应正确的 y 的点排在前面。
- 对于每一个 $y'\neq y$，$\langle w, \Psi(x, y)\rangle$ 和 $\langle w, \Psi(x, y')\rangle$ 之间的偏差比用 y' 替代 y 的损失更大。$\langle w, \Psi(x, y)\rangle - \langle w, \Psi(x, y')\rangle$ 也指"间隔"(margin)(见 15.1 节)。

可以通过下图表来说明：



17.2.5　多分类 SVM 和 SGD

一旦定义了泛化合页损失，我们就得到了一个凸利普希茨可学习问题。我们就可以利用通用的方法来解决这样的问题。尤其是利用在第 13 章中学过的 RLM 方法得出多分类 SVM 准则：

[196]

多分类 SVM

输入：$(x_1, y_1), \ldots, (x_m, y_m)$

参数：

正则化参数 $\lambda>0$

损失函数 $\Delta: \mathcal{Y}\times\mathcal{Y}\to\mathbb{R}$

类敏感特征映射 $\Psi: \mathcal{X}\times\mathcal{Y}\to\mathbb{R}^d$

求解：

$$\min_{w\in\mathbb{R}^d}\left(\lambda\|w\|^2 + \frac{1}{m}\sum_{i=1}^{m}\max_{y'\in\mathcal{Y}}\left(\Delta(y', y_i) + \langle w, \Psi(x_i, y') - \Psi(x_i, y_i)\rangle\right)\right)$$

输出：预测器 $h_w(x) = \operatorname{argmax}_{y\in\mathcal{Y}}\langle w, \Psi(x, y)\rangle$

我们可以使用一般的凸优化算法(或者是使用 15.5 节中描述的方法)来解决多分类 SVM 的优化问题。让我们分析结果假设的风险。分析可以完美无偏差地遵从第 13 章中对凸利普希茨问题的一般分析。尤其是，应用推论 13.8 和泛化合页损失限定 Δ 损失的上界这一事实，很容易得到一个类似推论 15.7 的推论。

推论 17.1　令 $\mathcal{D}$ 服从 $\mathcal{X}\times\mathcal{Y}$ 分布，$\Psi: \mathcal{X}\times\mathcal{Y}\to\mathbb{R}^d$，假设对于所有的 $x\in\mathcal{X}$，$y\in\mathcal{Y}$，满足 $\|\Psi(x, y)\|\leq\rho/2$，令 $B>0$，在训练集 $S\sim\mathcal{D}^m$ 上用参数 $\lambda=\sqrt{\frac{2\rho^2}{B^2 m}}$ 运行多分类 SVM，h_w

是多分类 SVM 的输出。那么，

$$\mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}\left[L_{\mathcal{D}}^{0\text{-}1}(h_{\mathbf{w}})\right]\leqslant \mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}\left[L_{\mathcal{D}}^{\text{gen}}(\mathbf{w})\right]\leqslant \min_{\mathbf{u}:\|\mathbf{u}\|\leqslant B} L_{\mathcal{D}}^{\text{hinge}}(\mathbf{u})+\sqrt{\frac{8\rho^2B^2}{m}}$$

其中，$L_{\mathcal{D}}^{0\text{-}1}(h)=\mathbb{E}_{(x,y)\sim\mathcal{D}}[\Delta(h(x)),\ y]$，$L_{\mathcal{D}}^{\text{hinge}}(\mathbf{w})=\mathbb{E}_{(x,y)\sim\mathcal{D}}[\ell(\mathbf{w},\ (x,\ y))]$，$\ell$是等式(17.3)中定义的泛化合页损失。

我们也可以应用 SGD 学习框架来最小化第 14 章提到的 $L_{\mathcal{D}}^{\text{gen}}(\mathbf{w})$。回顾一下论断 14.6，说明了处理最大化函数的次梯度。根据论断 14.6，为了找到泛化合页损失的次梯度，我们要找到 $y\in\mathcal{Y}$，使其能够实现泛化合页损失定义中的最大化。它遵从以下算法： [137]

多分类学习的 SGD

参数：

标量 $\eta>0$，参数 $T>0$

损失函数 Δ：$\mathcal{Y}\times\mathcal{Y}\to\mathbb{R}_+$

类敏感特征映射 Ψ：$\mathcal{X}\times\mathcal{Y}\to\mathbb{R}^d$

初始化： $\mathbf{w}^{(1)}=\mathbf{0}\in\mathbb{R}^d$

对于 $t=1,\ 2,\ \cdots,\ T$

样本 $(x,\ y)\sim\mathcal{D}$

找到 $\hat{y}\in\operatorname{argmax}_{y'\in\mathcal{Y}}(\Delta(y',\ y)+\langle\mathbf{w}^{(t)},\ \Psi(x,\ y')-\Psi(x,\ y)\rangle)$

令 $\mathbf{v}_t=\Psi(x,\ \hat{y})-\Psi(x,\ y)$

更新 $\mathbf{w}^{(t+1)}=\mathbf{w}^{(t)}-\eta\mathbf{v}_t$

输出： $\overline{\mathbf{w}}=\frac{1}{T}\sum_{t=1}^{T}\mathbf{w}^{(t)}$

我们对推论 14.12 中给定的 SGD 进行一般地分析可以很快得出下列推论：

推论 17.2　令$\mathcal{D}$服从$\mathcal{X}\times\mathcal{Y}$分布，$\Psi$：$\mathcal{X}\times\mathcal{Y}\to\mathbb{R}^d$，假设对于所有的 $x\in\mathcal{X}$，$y\in\mathcal{Y}$，满足 $\|\Psi(x,\ y)\|\leqslant\rho/2$，令 $B>0$，那么对于每一个 $\epsilon>0$，通过若 T 次迭代运算(也就是样本数量)来执行 SGD 进行多分类学习

$$T\geqslant\frac{B^2\rho^2}{\epsilon^2}$$

与 $\eta=\sqrt{\frac{B^2}{\rho^2T}}$时，SGD 的输出满足

$$\mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}\left[L_{\mathcal{D}}^{0\text{-}1}(h_{\overline{\mathbf{w}}})\right]\leqslant \mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}\left[L_{\mathcal{D}}^{\text{gen}}(\overline{\mathbf{w}})\right]\leqslant \min_{\mathbf{u}:\|\mathbf{u}\|\leqslant B} L_{\mathcal{D}}^{\text{hinge}}(\mathbf{u})+\epsilon$$

评注　推论 17.1 和推论 17.2 中给定的风险边界并不直接地依赖于标签集$\mathcal{Y}$的大小，我们下一节将依赖于这个事实。然而，边界有可能通过 $\Psi(x,\ y)$的形式间接地依赖于标签集$\mathcal{Y}$的大小。只有存在一些向量 $\mathbf{u}$，$\|\mathbf{u}\|\leqslant B$，使得 $L_{\mathcal{D}}^{\text{hinge}}(\mathbf{u})$不是特别大的情况下，边界才有意义。

17.3　结构化输出预测

结构化输出预测问题是一个多分类问题。$\mathcal{Y}$是一个非常大的假设类，但是被赋予一个

预定义的结构。此结构在构建有效的算法中扮演着重要角色。为了促进解决结构学习问题，考虑光学字符识别问题。假设我们接收一幅手写字图像，并要预测图像中是哪些字。为了简化背景，假设我们知道如何把图像分割成序列，这些图像每一幅都包含了一个对应于单个字符的补丁图像。因此，$\mathcal{X}$是一组图像序列，$\mathcal{Y}$是一组字母序列。注意，$\mathcal{Y}$的大小随最大长度的增加呈指数增长。对应于标签 y = "可行" 的图像 x 举例如下。

198

为了解决结构预测，可以利用前面章节描述的线性预测器的函数族。特别地，我们需要定义一个针对该问题的合理的损失函数 Δ，也需要一个对类敏感的好的特征映射 Ψ。说"好"，意味着一个特征映射对关于 Ψ 和 Δ 的线性预测的类，将会带来一个低的近似误差。一旦如此定义，就可以利用前一节定义的 SGD 学习算法。

但是，$\mathcal{Y}$的庞大规模带来一些挑战：

1）为了运用多分类预测，我们需要解决关于$\mathcal{Y}$的最大化问题。当$\mathcal{Y}$非常大时，我们如何有效地预测？

2）我们如何有效地训练 w？特别地，为了运用 SGD 规则，我们再次需要解决关于$\mathcal{Y}$的最大化问题。

3）如何才能避免过拟合？

在上一节中已经说明，一个多分类的线性预测器的样本复杂度并不是明确地依赖于类的个数。我们只需确保 Ψ 的值域的范数不是太大。这将会解决过拟合问题。为了解决计算上的挑战，我们基于这个问题的结构并定义函数 Ψ 和 Δ，以便于在定义 h_w 和 SGD 算法下有效地计算最大问题。接下来，我们展示一种方法来实现之前提到的 OCR 任务。

为了简化表示，我们假设在$\mathcal{Y}$中的所有单词的长度为 r，字母表中不同字母的数目为 q。令 y 和 y' 是$\mathcal{Y}$中两个不同的单词（即字母序列）。定义函数 $\Delta(y, y')$ 是在 y 和 y' 中不同字母的平均数，即 $\frac{1}{r}\sum_{i=1}^{r}\mathbb{1}_{[y_i \neq y'_i]}$。

接着定义一个对类敏感的特征映射 $\Psi(x, y)$，x 是 $n \times r$ 的矩阵，n 是每幅图像的像素，r 是图像序列中图像的数目。x 中的第 j 列对应于序列中第 j 幅图像（被编码为一个像素灰度级的序列）。Ψ 的维度幅度设为 $d = nq + q^2$。

第一个 nq 特征函数是"类型 1"特征，函数形式如下：

$$\Psi_{i,j,1}(x,y) = \frac{1}{r}\sum_{t=1}^{r} x_{i,t}\,\mathbb{1}_{[y_t=j]}$$

即，只将所有类别为 j 的图像中的第 i 个位置的像素值累加起来。三维坐标$(i, j, 1)$表明处理类型 1 的特征(i, j)。直观地，这些特征可以捕获图像中的像素，这些图像的灰度级由一个确定的字母表示。第二种类型的特征形式是：$\Psi_{i,j,2}(x,y) = \frac{1}{r}\sum_{t=2}^{r}\mathbb{1}_{[y_t=i]}\mathbb{1}_{[y_{t-1}=j]}$。

199

即，累加图像矩阵中前一个图像类别为 j 后一个类别为 i 的情况出现次数。直观地，这些特征可以捕捉到这样的规则：在一个词中可能见到"qu"或者在一个词中不太可能见到"zx"。当然，其中的一些特征可能没有多大用处，因此学习过程的目标是通过学习向量 w

分配给特征一定的权重。因此，带权重的结果通过以下函数将给出一个良好的预测：

$$h_w(x) = \operatorname*{argmax}_{y \in \mathcal{Y}} \langle w, \Psi(x, y) \rangle$$

接下来说明如何有效地解决目标函数 $h_w(x)$ 的优化问题，也就是如何通过 SGD 算法求解上述优化问题以获得最优解 $\tilde{y}$。这个问题可以利用动态规划过程来解决。我们描述了在 h_w 的定义中解决最大化问题的过程，并且把 SGD 算法中 $\tilde{y}$ 的定义的一个最大化问题留作练习。

为了导出动态规划，首先观察并写出 $\Psi(x, y) = \sum_{t=1}^{r} \phi(x, y_t, y_{t-1})$，对于一个合适的 $\phi: \mathcal{X} \times [q] \times [q] \cup \{0\} \to \mathbb{R}^d$。为了简化，我们假设 y_0 恒等于 0。事实上，每个特征函数 $\Psi_{i,j,1}$ 可以被写成 $\phi_{i,j,1}(x, y_t, y_{t-1}) = x_{i,t} \mathbf{1}_{[y_t = j]}$，而特征函数 $\Psi_{i,j,2}$ 可以被写成 $\phi_{i,j,2}(x, y_t, y_{t-1}) = \mathbf{1}_{[y_t = j]} \mathbf{1}_{[y_{t-1} = i]}$。因此，预测函数可以被写成

$$h_w(x) = \operatorname*{argmax}_{y \in \mathcal{Y}} \sum_{t=1}^{r} \langle w, \phi(x, y_t, y_{t-1}) \rangle \tag{17.4}$$

接下来我们导出一个动态程序，解决方程(17.4)中给出形式的每一个问题。程序将会令矩阵 $M \in \mathbb{R}^{r \times q}$，满足

$$M_{t,s} = \max_{(y_1, \ldots, y_t): y_t = s} \sum_{t=1}^{t} \langle w, \phi(x, y_t, y_{t-1}) \rangle$$

很清楚地，内积 $\langle w, \Psi(x, y) \rangle$ 等于 $\max_s M_{r,s}$。此外，我们可以用一个递归的方式计算 M：

$$M_{t,s} = \max_{s'} (M_{t-1,s'} + \langle w, \phi(x, s, s') \rangle) \tag{17.5}$$

200

得出以下算法：

计算 $h_w(x)$（由方程(17.4)给出）的动态规划

输出：矩阵 $x \in \mathbb{R}^{d \times r}$ 和向量 w

初始化：

　对每个 $s \in [q]$

　　$M_{1,s} = \langle w, \phi(x, s, -1) \rangle$

对 $t = 2, \ldots, r$

　对每个 $s \in [q]$

　　令 $M_{t,s}$ 如方程(17.5)中所述

　　令 $I_{t,s}$ 为 s'，最大化方程(17.5)

　　令 $y_t = \operatorname*{argmax}_s M_{r,s}$

对 $t = r, r-1, \ldots, 2$

　令 $y_{t-1} = I_{t,y_t}$

输出：$y = (y_1, \ldots, y_r)$

17.4　排序

排序是根据实例之间的“关联”排列这些实例的问题。一个典型的应用是排列一个搜索引擎的结果（根据这些结果与查询的关联）。另一个例子是监控电子事务处理的系统，对可能的欺诈交易报警。这个系统会根据交易的可疑程度筛选交易。

正式地，令$\mathcal{X}^*=\bigcup_{r=1}^{\infty}\mathcal{X}^r$是任意长度的$\mathcal{X}$的所有实例序列集。一个排序假设类$h$是一个接受实例$\bar{x}=(x_1,\cdots,x_r)\in\mathcal{X}^*$的序列的函数，返回一个$[r]$的排列。更方便地，令$h$的输出是一个向量$y\in\mathbb{R}^r$，对$y$的元素进行排序，我们获得了$[r]$上的排列。我们用$\pi(y)$来表示$[r]$的排列。比如，$r=5$，向量$y=(2,1,6,-1,0.5)$引出$\pi(y)=(4,3,5,1,2)$。也就是说，如果我们对$y$进行一个升序排序，那么得到向量$(-1,0.5,1,2,6)$。现在$\pi(y)_i$是在排序向量$(-1,0.5,1,2,6)$中$y_i$的位置。这个符号反映了排序最高的实例在$\pi(y)$中取得最高值。

在PAC学习模型的符号中，实例的定义域是$Z=\bigcup_{r=1}^{\infty}(\mathcal{X}^r\times\mathbb{R}^r)$，假设类$\mathcal{H}$是一些排序假设类的集合。下面我们描述排序问题的损失函数。有许多方法可以定义这样的损失函数，我们列出了几个例子。对于全部例子而言，定义$\ell(h(\bar{x},y))=\Delta(h(\bar{x}),y)$，损失函数$\Delta:\bigcup_{r=1}^{\infty}(\mathbb{R}^r\times\mathbb{R}^r)\to\mathbb{R}_+$。

- **0—1排序损失**：如果y'和y引出完全一样的排序，则$\Delta(y',y)=0$；否则$\Delta(y',y)=1$。也就是说，$\Delta(y',y)=\mathbb{I}_{[\pi(y')\neq\pi(y)]}$。这样一个损失函数几乎从不用于实际中，因为它不能区分$\pi(y)$和$\pi(y')$几乎相等以及$\pi(y)$和$\pi(y')$完全不同的情况。

251

- **Kendall-Tau损失**：计算在两个排列中排序不同的(i,j)对的数目，这可以被写成

$$\Delta(y',y)=\frac{2}{r(r-1)}\sum_{i=1}^{r-1}\sum_{j=i+1}^{r}\mathbb{I}_{[(y'_i-y'_j)\neq(y_i-y_j)]}$$

这个损失函数比0—1函数更有用，因为它反映出两个排列的相似度。

- **归一化折扣累积增益(NDCG)**：这个测量强调用一个单调递增的折扣函数$D:\mathbb{N}\to\mathbb{R}_+$。我们首先定义一个折扣累积增益测度：$G(y',y)=\sum_{i=1}^{r}D(\pi(y')_i)y_i$。通俗地讲，如果我们把$y_i$解释为目标$i$的"正确的关联"的评分，那么取要素间关联的加权和，而y_i的权重取决于在$\pi(y')$中的目标i的位置。假设y中的所有元素都是非负的，容易证明$0\leq G(y',y)\leq G(y,y)$。因此我们可以根据比率$G(y',y)/G(y,y)$定义一个NDCG，对应的损失函数是

$$\Delta(y',y)=1-\frac{G(y',y)}{G(y,y)}=\frac{1}{G(y,y)}\sum_{i=1}^{r}(D(\pi(y)_i)-D(\pi(y')_i))y_i$$

容易看出$\Delta(y',y)\in[0,1]$，且当$\pi(y)=\pi(y')$时$\Delta(y',y)=0$。

定义折扣函数的一个典型方法是：

$$D(i)=\begin{cases}\dfrac{1}{\log_2(r-i+2)} & 若\quad i\in\{r-k+1,\cdots,r\}\\ 0 & 其他\end{cases}$$

其中$k<r$，这意味着我们更多地关心为更高排序的元素，并完全忽略了不在排序前k的其他元素。NDCG测度被用来估算搜索引擎的性能，因为在这样的应用中忽略不在排序前k的元素完全是有道理的。

一旦有一个假设类和一个排序损失函数，我们就可以用ERM准则学习出一个排序函数。然而，从计算的角度来看，得出理想的结果似乎有点难。我们接下来讨论如何学习排序问题的线性预测器。

排序线性预测器

排序函数的一个简单定义是将样本投影到某个向量$w\in\mathbb{R}^d$，然后将输出的标量结果当

做排序函数的表示。即，假设$\mathcal{X}\subset\mathbb{R}^d$，对每个 $w\in\mathbb{R}^d$ 定义一个排序函数 [202]

$$h_w((x_1,\cdots,x_r))=(\langle w,x_1\rangle,\cdots,\langle w,x_r\rangle) \tag{17.6}$$

正如第 16 章中的讨论，我们也可以首先进行特征映射，即将样本映射到某个特征空间，然后在特征空间中计算与 w 的内积。简单起见，我们只考虑式(17.6)这样的简化形式。

给定某个 $W\subset\mathbb{R}^d$，定义假设类$\mathcal{H}_W=\{h_w:w\in W\}$。一旦定义好这个假设类，选好排序损失函数，我们可以运用 ERM 准则如下：给定训练集 $S=(\bar{x}_1,y_1),\cdots,(\bar{x}_m,y_m)$，这里每个$(\bar{x}_i,y_i)$都在$(\mathcal{X}\times\mathbb{R})^r$内，其中 $r_i\in\mathbb{N}$，我们需要搜索 $w\in W$ 以最小化经验风险 $\sum_{i=1}^{m}\Delta(h_w(\bar{x}_i),y_i)$。正如二分类的情形，对很多损失函数来说，这个问题的计算都是困难的，因此我们转而采用凸替代损失函数。我们将说明 Kendall tau 损失和 NDCG 损失的替代。

1. Kendall Tau 损失函数的合页损失

我们可以将 Kendall Tau 损失看做每一对样本间的 0—1 损失的平均。特别地，对于每对(i,j)我们重写为

$$\mathbb{1}_{[\mathrm{sign}(x_i'-x_j')\neq\mathrm{sign}(y_i-y_j)]}=\mathbb{1}_{[\mathrm{sign}(y_i-y_j)(x_i'-x_j')\leq 0]}$$

在这里，$(x_i'-x_j')=\langle w,x_i-x_j\rangle$，由此我们可以采用合页损失作为上界，方法如下：

$$\mathbb{1}_{[\mathrm{sign}(y_i-y_j)(x_i'-x_j')\leq 0]}\leq\max(0,1-\mathrm{sign}(y_i-y_j)\langle w,x_i-x_j\rangle)$$

在所有样本对上取平均，可以得到如下的 Kendall tau 损失函数的凸替代损失：

$$\Delta(h_w(x),y)\leq\frac{2}{r(r-1)}\sum_{i=1}^{r-1}\sum_{j=i+1}^{r}\max(0,1-\mathrm{sign}(y_i-y_j)\langle w,x_i-x_j\rangle)$$

上式右边关于 w 是凸的，并且也是 Kendall tau 的上界，它也是参数为 $\rho\leq\max_{i,j}\|x_i-x_j\|$ 的ρ-利普希茨函数。

2. NDCG 损失函数的合页损失

NDCG 损失函数依赖于预测排序向量 $y'\in\mathbb{R}^r$（由它引出的排列）。为引出替代损失函数，我们首先观察到如下事实：令 V 为所有在$[r]$上的表示为向量的排列的集合；即，每个$v\in V$是在$[r]^r$中的向量，并且满足对所有 $i\neq j$ 有 $v_i\neq v_j$。那么(参看练习 17.4)，

$$\pi(y')=\mathrm{argmax}_{v\in V}\sum_{i=1}^{r}v_iy_i' \tag{17.7}$$ [203]

令 $\Psi(\bar{x},v)=\sum_{i=1}^{r}v_ix_i$；由此

$$\begin{aligned}\pi(h_w(\bar{x}))&=\mathrm{argmax}_{v\in V}\sum_{i=1}^{r}v_i\langle w,x_i\rangle=\mathrm{argmax}_{v\in V}\langle w,\sum_{i=1}^{r}v_ix_i\rangle\\&=\mathrm{argmax}_{v\in V}\langle w,\Psi(\bar{x},v)\rangle\end{aligned}$$

基于以上事实，我们可以采用针对代价敏感的多分类问题的泛化合页损失，把它作为 NDCG 损失的替代损失函数：

$$\begin{aligned}\Delta(h_w(x),y)&\leq\Delta(h_w(x),y)+\langle w,\Psi(\bar{x},\pi(h_w(\bar{x})))\rangle-\langle w,\Psi(\bar{x},\pi(y))\rangle\\&\leq\max_{v\in V}[\Delta(v,y)+\langle w,\Psi(\bar{x},v)\rangle-\langle w,\Psi(\bar{x},\pi(y))\rangle]\\&=\max_{v\in V}[\Delta(v,y)-\sum_{i=1}^{r}(v_i-\pi(y)_i)\langle w,x_i\rangle]\end{aligned} \tag{17.8}$$

上式右边关于 w 是凸的。

我们现在可以采用 17.2.5 节阐述的 SGD 方法来解决这个学习问题。主要的计算瓶颈在于计算损失函数的次梯度，这等价于搜索使得式(17.8)达到最大的 v(参看论断 14.6)。采用 NDCG 损失的定义，这等价于解决以下问题

$$\operatorname*{argmin}_{v\in V}\sum_{j=1}^{r}(\alpha_j v_j+\beta_j D(v_j))$$

其中 $\alpha_j=-\langle w, x_j\rangle$ 且 $\beta_j=y_j/G(\mathbf{y},\mathbf{y})$。我们可以稍微换个视角考虑这个问题，定义矩阵 $A\in\mathbb{R}^{r\times r}$，其中

$$A_{i,j}=\alpha_j v_i+D(j)\beta_i$$

现在，将每个 j 视为"工人"，每个 i 视为"工作"，且 $A_{i,j}$ 为将工作 i 指派给工人 j 完成所需要的花费。在这个视角下，搜索 v 的问题被转换为寻找花费最少的指派方式。该问题被称为"指派问题"，能被有效地加以解决。一种特别的算法是"匈牙利算法"(Kuhn1955)。另一种解决指派问题的方法是线性规划。首先将指派问题重写为

[204]

$$\operatorname*{argmin}_{B\in\mathbb{R}^{r\times r}}\sum_{i,j=1}^{r}A_{i,j}B_{i,j} \tag{17.9}$$

$$\text{s.t. } \forall i\in[r],\ \sum_{j=1}^{r}B_{i,j}=1$$

$$\forall j\in[r],\ \sum_{i=1}^{r}B_{i,j}=1$$

$$\forall i,j,\ B_{i,j}\in\{0,1\}$$

满足前述优化问题中限制条件的矩阵 B 被称为置换矩阵。这是因为限制条件保证矩阵每行每列均至多有一项为 1。因此，矩阵 B 与置换向量 $v\in V$ 一一对应，满足对于 $v_i=j$ 有唯一的 j 使得 $B_{i,j}=1$。

由于组合限制 $B_{i,j}\in\{0,1\}$ 的存在，前述优化问题仍然不是一个线性规划问题。然而事实上，这个限制是多余的——如果忽略该组合限制而直接求解优化问题，我们仍然能保证存在最优解满足该组合限制。之后将给出正规的结论。

令 $\langle A,B\rangle=\sum_{i,j}A_{i,j}B_{i,j}$。那么，式(17.9)是一个使得 B 为置换矩阵的优化问题。

矩阵 $B\in\mathbb{R}^{r\times r}$ 被称作双随机矩阵，如果 B 的所有元素非负，且 B 的每行每列的和均为 1。因此忽略 $B_{i,j}\in\{0,1\}$ 的限制来求解式(17.9)即为如下问题：

$$\operatorname*{argmin}_{B\in\mathbb{R}^{r\times r}}\langle A,B\rangle \quad \text{s.t.} \quad B\text{ 是双随机矩阵} \tag{17.10}$$

下述论断说明每个双随机矩阵都是置换矩阵的凸组合。

论断 17.3(Birkhoff 1946，Von Neumann 1953)　*$\mathbb{R}^{r\times r}$ 中双随机矩阵的集合是 $\mathbb{R}^{r\times r}$ 中置换矩阵集合的凸包。*

在该论断的基础上，易得如下引理：

引理 17.4　*存在式(17.10)的最优解，它也是式(17.9)的最优解。*

证明　令 B 为式(17.10)的解。那么根据论断 17.3，有 $B=\sum_i\gamma_i C_i$，其中每个 C_i 都是置换矩阵，每个 $\gamma_i\geq 0$ 且满足 $\sum_i\gamma_i=1$。既然所有的 C_i 都是双随机矩阵，显然对所有 i

都有$\langle A, B\rangle \leq \langle A, C_i\rangle$。我们断言存在某个 i 使得$\langle A, B\rangle = \langle A, C_i\rangle$。该断言必定成立，否则，对每个 i 都有$\langle A, B\rangle < \langle A, C_i\rangle$，我们将得到

$$\langle A,B\rangle = \langle A,\sum_i \gamma_i C_i\rangle = \sum_i \gamma_i \langle A,C_i\rangle > \sum_i \gamma_i \langle A,B\rangle = \langle A,B\rangle$$

205

上式矛盾。因此存在某个置换矩阵 C_i，满足$\langle A, B\rangle = \langle A, C_i\rangle$。但是，因为对其他的置换矩阵 C 均有$\langle A, B\rangle \leq \langle A, C_i\rangle$，所以可以得到结论：$C_i$ 是式(17.9)和式(17.10)的最优解。 ■

17.5　二分排序以及多变量性能测量

在之前的章节我们已经描述了排序问题，用向量 $y \in \mathbb{R}^r$ 来表示成员 $x_1, \cdots, x_r$ 的顺序。如果 y 中的所有成员都是不同的，那么 y 就是一个全序。可是，如果 y 中的两个成员的值是相同的，$y_i = y_j$(对于 $i \neq j$)，那么 y 就只是一个部分的序。在这种情况下，我们说 x_i 和 x_j 在 y 上是平等关系的。极端情况下，设 $y \in \{\pm 1\}^r$，意味着每一个 x_i 或者相关或者不相关。这种情况经常被称作“二分排序”。例如，在前面提到的欺诈检测中，每一个交易都被标记为欺诈性($y_i = 1$)或者良性($y_i = -1$)。

这样看来，通过学习一个二分类器，将其应用在每一个成员上，并将正值排序在前，我们可以解决这个二分排序问题。可是，这种排序策略采用的是二分类的方法，而二分类的优化目标往往是与排序目标不同的 0—1 损失，所以效果并不好。为了说明这一点，我们再次考虑欺诈检测问题。通常情况下，大多数交易都是良性的(99%)。因此，一个二分类器如果将所有的交易都预测为良性，那么得到的 0—1 误差也仅仅只有 0.1%。这是一个非常小的数字，但是对于欺诈检测问题却是毫无意义的。这个问题的症结来自于 0—1 损失对于排序问题的不准确性。我们需要考虑一种在全体实例上更准确有效的测量方法。例如，在先前的部分中我们已经定义了 NDCG 损失，它更注重排序前列实例的准确性。在本节，我们将介绍对于二分排序问题更准确的一些损失函数。

正如前面章节提到的，假设我们得到一个实例序列 $\bar{x} = (x_1, \cdots, x_r)$，并且预测一个排序向量 $y' \in \mathbb{R}^r$。反馈向量是 $y \in \{\pm 1\}^r$。我们定义一个依赖于 y 和 y' 以及阈值 $\theta \in \mathbb{R}$ 的损失。这个阈值就可以将向量 y' 转变为向量$(\mathrm{sign}(y_1' - \theta), \cdots, \mathrm{sign}(y_r' - \theta)) \in \{\pm 1\}^r$。通常情况下，$\theta$ 的值一般设为 0。可是，正如所看见的那样，我们设置阈值 θ 的时候需要考虑一些额外的限制。

接下来定义的损失函数取决于以下四个参数：

$$\begin{aligned}
&\text{真阳性}: a = |\{i : y_i = +1 \wedge \mathrm{sign}(y_i' - \theta) = +1\}| \\
&\text{假阳性}: b = |\{i : y_i = -1 \wedge \mathrm{sign}(y_i' - \theta) = +1\}| \\
&\text{假阴性}: c = |\{i : y_i = +1 \wedge \mathrm{sign}(y_i' - \theta) = -1\}| \\
&\text{真阴性}: d = |\{i : y_i = -1 \wedge \mathrm{sign}(y_i' - \theta) = -1\}|
\end{aligned} \tag{17.11}$$

预测矢量的召回率(又叫敏感率)是真阳性 y' “捕获”的比值，即$\frac{a}{a+c}$。准确率是指所有正样本中预测的正确样本与正样本的比值，即$\frac{a}{a+b}$。特异率是指预测器“捕获”的真阴性的比值，即$\frac{d}{d+b}$。

206

注意到当我们减少 θ 时，召回却会增加(当 $\theta = -\infty$ 时，达到值 1)。另一方面，当减少 θ 时准确率和特异率通常会降低。因此，在准确率和召回率之间需要做一个权衡，而我们

可以通过改变θ的取值进行控制。接下来定义的损失函数运用不同技巧来结合准确率和召回率。

- **平均敏感率和特异率**：这种测量方法平均了敏感率和特异率，即$\frac{1}{2}\left(\frac{a}{a+c}+\frac{d}{d+b}\right)$。这同样也是准确率在正样本与负样本之间的平均。这里，我们令$\theta=0$，相应的损失函数是$\Delta(y', y)=1-\frac{1}{2}\left(\frac{a}{a+c}+\frac{d}{d+b}\right)$。
- **F_1-得分**：F_1得分方法是准确率和召回率的调和平均：$\frac{2}{\frac{1}{\text{准确率}}+\frac{1}{\text{召回率}}}$。当召回率和准确率同时取1时得到它的最大值(取1)，当召回率和准确率有任意一个取0(即便另外一个取1)时得到它的最小值(取0)。F_1得分还可以利用参数a，b，c写作：$F_1=\frac{2a}{2a+b+c}$。同样，我们令$\theta=0$，得到的损失函数是$\Delta(y', y)=1-F_1$。
- **F_β-得分**：F_β得分方法类似F_1得分方法，只是在召回率项增加了β^2的权重，即$\frac{1+\beta^2}{\frac{1}{\text{准确率}}+\beta^2\frac{1}{\text{召回率}}}$。它同样可以写作$F_\beta=\frac{(1+\beta^2)a}{(1+\beta^2)a+b+\beta^2 c}$。同样，我们令$\theta=0$，得到的损失函数是$\Delta(y', y)=1-F_\beta$。
- **k处召回率**：我们测量当预测中至多包含k个正样本时的召回率。这意味着，需要设置θ的取值使其满足$a+b\leq k$。这样做是很便捷的，例如在欺诈检测系统中，银行职员仅需处理很少的有嫌疑的交易。
- **k处准确率**：我们测量当预测中至少包含k个正样本时的准确率。这意味着，需要设置θ的取值使其满足$a+b\geq k$。

前面介绍的方法通常被称作多变量性能测量。注意到这些方法是与均衡0—1损失极度不同的，即在前面符号的表示中等于$\frac{b+d}{a+b+c+d}$。在前面所述的欺诈检测例子中，当99.9%的实例都被标记为负样本，那么预测所有实例都为负样本的0—1损失仅仅0.1%。相反，这样的预测的召回率是0，由此可得F_1得分也是0，故相应的损失函数将会是1。

二分排序线性预测器

我们接下来介绍如何针对二分排序训练线性预测器。如前面章节所述，一个线性预测器对于排序问题定义为：

$$h_w(\bar{x})=(\langle w, x_1\rangle, \cdots, \langle w, x_r\rangle)$$

相应的损失函数是之前介绍的多变量性能测量的一种。损失函数通过它引导的二值向量取决于$y'=h_w(\bar{x})$，记作

207

$$b(y')=(\operatorname{sign}(y'_1-\theta), \cdots, \operatorname{sign}(y'_r-\theta))\in\{\pm 1\}^r \tag{17.12}$$

如前面章节，为了使算法高效，我们得到在Δ上的一个凸损失替代函数。这与之前对于NDCG排序损失的泛化合页损失相类似。

我们应该首先注意到对于所有之前定义的θ值，存在$V\subseteq\{\pm 1\}^r$，使得$b(y')$可以被改写为

$$b(y')=\operatorname*{argmax}_{v\in V}\sum_{i=1}^{r} v_i y'_i \tag{17.13}$$

如果我们选择 $V \subseteq \{\pm 1\}^r$，在 $\theta=0$ 情况下上式显然正确。而 k 处准确率及 k 处召回率这两种方法并不是将 θ 设为 0。对于 k 处准确率，我们可以将集合 V 设为 $V_{\geq k}$，其中 $V_{\geq k}$ 是指所有在 $\{\pm 1\}^r$ 向量中元素为 1 的个数至少是 k 的向量集合。对于 k 处召回率，我们可以类似地将集合 V 设为 $V_{\leq k}$。详见练习 17.3。

一旦我们用等式(17.13)定义了 b，可以很容易得到一个如下的凸替代损失。假设 $y \in V$，我们有

$$\begin{aligned}\Delta(h_w(\overline{x}), y) &= \Delta(b(h_w(\overline{x})), y) \\ &\leq \Delta(b(h_w(\overline{x})), y) + \sum_{i=1}^{r} (b_i(h_w(\overline{x})) - y_i)\langle w, x_i\rangle \\ &\leq \max_{v \in V}\left[\Delta(v, y) + \sum_{i=1}^{r} (v_i - y_i)\langle w, x_i\rangle\right] \qquad (17.14)\end{aligned}$$

等式的右边是一个关于 w 的凸替代损失。

现在我们可以运用 17.2.5 节描述的 SGD 准则来解决这个学习问题。主要的计算瓶颈在于计算损失函数的次梯度，它等价于寻找 v 使得等式(17.14)达到最大值(见论断 14.6)。

接下来，我们将要介绍对于任何一个可以被写为等式(17.11)所给出参数 a，b，c，d 的函数的性能测量，并且对于集合 V 包含 $\{\pm 1\}^r$ 中的所有成员(参数 a，b 满足一些限制)，如何有效的寻找这个最大值。例如，对于"k 处召回率"，集合 V 就是满足 $a+b \leq k$ 的所有向量。

方法如下所示，对于所有的 a，$b \in [r]$，令

$$\overline{\mathcal{Y}}_{a,b} = \{v : |\{i : v_i = 1 \wedge y_i = 1\}| = a \wedge |\{i : v_i = 1 \wedge y_i = -1\}| = b\}$$

任一向量 $v \in V$ 都会落入对于特定的 a，$b \in [r]$ 的集合 $\overline{\mathcal{Y}}_{a,b}$ 中。更进一步，如果对于某些 a，$b \in [r]$，$\overline{\mathcal{Y}}_{a,b} \cap V$ 非空，那么 $\overline{\mathcal{Y}}_{a,b} \cap V = \overline{\mathcal{Y}}_{a,b}$。因此，我们可以搜索每一个与 V 有非空交集的 $\overline{\mathcal{Y}}_{a,b}$，然后取最优值。其中我们应当注意到一旦搜索 $\overline{\mathcal{Y}}_{a,b}$ 内的元素，Δ 的值是固定的，所以我们只需最大化表达式：

$$\max_{v \in \overline{\mathcal{Y}}_{a,b}} \sum_{i=1}^{r} v_i \langle w, x_i\rangle$$

假设实例已经被排好顺序，那么 $\langle w, x_1\rangle \geq \cdots \geq \langle w, x_r\rangle$。很容易验证我们想将具有最小下标 i 的元素 v_i 标记为正样本。这样做，在 a，b 的限制下，这意味着对于那些 a 排在前列的正样本和对于那些 b 排在前列的负样本置 $v_i=-1$。算法如下：

208

解方程(17.14)

输入：

$(x_1, \cdots, x_r)$，$(y_1, \cdots, y_r)$，w，V，Δ

假设：

Δ 是由 a，b，c，d 表示的函数

对于函数 f，V 包含所有满足 $f(a, b)=1$ 的全体向量

初始化：

$P=|\{i : y_i=1\}|$，$N=|\{i : y_i=-1\}|$

$\mu=(\langle w, x_1\rangle, \cdots, \langle w, x_r\rangle)$，$a^*=-\infty$

```
    排序实例使得 p_1 ≥ p_2 ≥ … ≥ p_P
    令 i_1, …, i_P 是正样本的下标(已排序)
    令 j_1, …, j_N 是负样本的下标(已排序)
对于 a=0, 1, …, P
    c=P-a
    对于 b=0, 1, …, N 满足 f(a, b)=1
      d=N-b
      用 a, b, c, d 计算 Δ
      设定 v_1, …, v_P 满足 v_{i_1}=…=v_{i_a}=v_{j_1}=…=v_{j_b}=1, 且 v 中剩下的元素为-1
      令 α=Δ+Σ_{i=1}^{P} v_i p_i
      如果 α>α*
        α*=α, v*=v
输出 v*
```

17.6 小结

现实中的许多监督学习问题可以被看作学习一个对多分类预测器。我们通过介绍从多分类到二分类的约简开始本章的学习。然后描述并分析了多分类学习的线性预测器大家族。我们展示了这个预测器家族如何使用，即使类的数目极其巨大，只要我们有一个关于问题的足够大的结构。最后，我们描述了排序问题。在 29 章我们将会更详细地研究多分类学习的样本复杂度。

229

17.7 文献评注

一对多和一对一约简方法已经在纠错输出编码(ECOC)(Dietterich & Bakiri 1995, Allwein, Schapire & Singer 2000)的框架下被统一。还有其他约简类型，比如基于分类树的分类(参考，例如 Beygelzimer, Langford & Ravikumar(2007))，约简技术的局限性已经被研究过(Daniely 等 2011, Daniely 等 2012)。也可以见 29 章中分析多分类学习中的样本复杂度。

多分类学习的线性预测器的直接方法已经被研究过(Vapnik 1998, Weston & Watkins 1999, Crammer & Singer 2001)。特别地，多向量构建是由 Crammer 和 Singer (2001)提出的。

Collins(2000)已经展示了对于结构化输出问题如何应用感知器算法。也可以参考 Collins(2002)。一个相关的方法是对于条件随机场的有区别的学习，见 Lafferty 等(2001)。结构化输出 SVM 方法已经被研究了，见文献 Weston 等(2002)，Collins(2002)，Taskar 等(2003)，Tsochantaridis 等(2004)。

在结构输出章节中我们提出的计算预测器 $h_w(x)$ 的动态过程，类似于 HMMs 中向前向后变量计算的 Viterbi 过程(例如 Rabiner & Juang (1986))。更一般地，在结构输出中解决最大值问题与图模型中推断问题是极其相关的(见 Koller & Friedman (2009a))。

Chapelle, Le 和 Smola (2007) 提出了运用 NDCG 损失学习排序函数的方法，这个思

起来自于结构输出学习。他们同样发现定义泛化合页损失的最大化问题等价于指派问题。

Agarwal 和 Roth(2005)分析了二分排序的样本复杂度。Joachims(2005)研究了运用多变量性能测量解决二分排序的 SVM 结构化输出的适用性。

17.8　练习

17.1　考虑 $\mathbb{R}^n\times[k]$ 中的样本集 S，存在向量 $\mu_1,\dots,\mu_k$ 使得每个样本 $(x,y)\in S$ 属于以 μ_y 为中心，半径 $r\geq 1$ 的球中。同时假设对每对 $i\neq j$，有 $\|\mu_i-\mu_j\|\leq 4r$。考虑将每个样本用常数 1 连接在一起，然后应用多向量的构建方式，即，

$$\Psi(x,y)=[\underbrace{0,\dots,0}_{\in\mathbb{R}^{(y-1)n+(y-1)}},\underbrace{x_1,\dots,x_n,1}_{\in\mathbb{R}^{n+1}},\underbrace{0,\dots,0}_{\in\mathbb{R}^{(k-y)n+(k-y)}}]$$

试证存在向量 $w\in\mathbb{R}^{k(n+1)}$ 使得对每个 $(x,y)\in S$ 满足 $\ell(w,(x,y))=0$。

提示：观察到对每个样本 $(x,y)\in S$，可以对某个 $\|v\|\leq r$ 重写 $x=\mu_y+v$。现在，令 $w=[w_1,\dots,w_k]$，其中 $w_i=[\mu_i,-\|\mu_i\|^2/2]$。

17.2　多分类感知器：考虑以下算法：

> **多分类批量感知器**
>
> **输入：**
>
> 一个训练集 $(x_1,y_1),\dots,(x_m,y_m)$
>
> 一个类别敏感的特征映射 $\Psi:\mathcal{X}\times\mathcal{Y}\to\mathbb{R}^d$
>
> **初始化：** $w^{(1)}=(0,\dots,0)\in\mathbb{R}^d$
>
> 对于 $t=1,2,\dots$
>
> 若(存在 i 和 $y\neq y_i$，使得 $\langle w^{(t)},\Psi(x_i,y_i)\rangle\leq\langle w^{(t)},\Psi(x_i,y)\rangle$)那么
>
> $w^{(t+1)}=w^{(t)}+\Psi(x_i,y_i)-\Psi(x_i,y)$
>
> 否则
>
> 输出 $w^{(t)}$

试证如下定理：

定理 17.5　假设存在 w^* 使得对所有 i 和所有 $y\neq y_i$，$\langle w^*,\Psi(x_i,y_i)\rangle\geq\langle w^*,\Psi(x_i,y)\rangle+1$ 成立。令 $R=\max_{i,y}\|\Psi(x_i,y_i)-\Psi(x_i,y)\|$。那么，多分类感知器算法在经历至多 $(R\|w^*\|)^2$ 次迭代后终止，且终止时对任何 $i\in[m]$，满足 $y_i=\operatorname{argmax}_y\langle w^{(t)},\Psi(x_i,y)\rangle$。

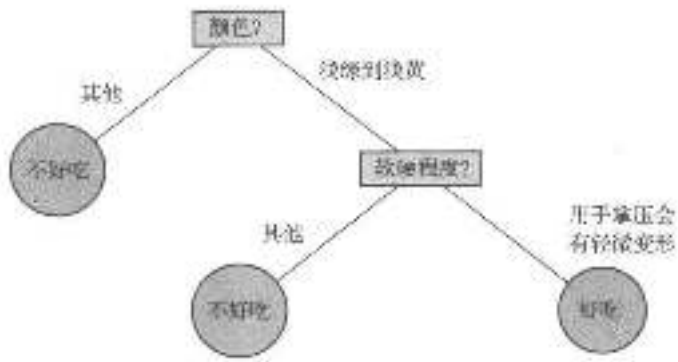
17.3　在多分类预测的 SGD 步骤中，由 $\hat{s}$ 的定义给出的最大化问题，试推广 17.3 节给出的动态规划步骤来解决它。你可以假设对某个函数 δ 有 $\Delta(y',y)=\sum_{i=1}^{t}\delta(y'_i,y_i)$。

17.4　证明式(17.7)成立。

17.5　证明式(17.12)和式(17.13)中定义的 π 在所有多变量性能测量上是等价的。

决 策 树

决策树是一种 $h:\mathcal{X}\to\mathcal{Y}$ 形式的预测器。从根结点开始，对实例 x 的某一特征进行测试，根据测试结果将实例分配到子结点，直至达到叶子结点。预测叶子结点所属的类即实例 x 的标签。为了简单起见，我们先考虑二分类情况，即 $\mathcal{Y}=\{0,1\}$。但是决策树也可以被应用到其他预测问题。从根结点到叶子结点路径上的每个结点，其后继结点是输入空间的一种拆分。通常，拆分是根据实例 x 的某一特征或是预先设定的拆分规则。每个叶子结点都对应一个特定的标签。下面给出水瓜一例（在第 2 章中论述）的一种决策树：



为了判断一个给定的水瓜好吃还是不好吃，决策树先测试水瓜的颜色，如果颜色不在浅绿到浅黄的范围之内，决策树不用做其他测试，直接预测该水瓜是不好吃的。否则，决策树转而测试水瓜的软硬程度。如果用手掌压水瓜产生轻微的变形，则决策树预测该水瓜是好吃的。否则，决策树预测该水瓜是不好吃的。前面所述的例子凸显了决策树的一个主要的优点——分类器的结果很容易理解和解释。

18.1 采样复杂度

一种流行的决策树中间结点拆分规则是对单个特征二值化。我们向左子结点还是右子结点移动基于 $\mathbb{1}_{[x_i<\theta]}$，其中 $i\in[d]$ 是特征的索引，$\theta\in\mathbb{R}$ 是阈值。在这种情况下，我们可以把决策树看作是将实例空间 $\mathcal{X}=\mathbb{R}^d$ 拆分成一系列单元，每个叶子结点对应一个单元。由此得出结论，一棵有 k 个叶子的树能够打散一个包含 k 个实例的集合。因此，如果我们允许决策树是任意大小的，将得到一个 VC 维无穷大的假设类。这种方法很容易造成过拟合。

为了防止过拟合，我们可以利用第 7 章所述的最小描述长度准则（MDL），学习一棵决策树，使之一方面能很好地拟合数据，另一方面树的规模不会太大。

为了简单起见，我们假定 $\mathcal{X}=\{0,1\}^d$。换言之，每个实例是一个 d 维的向量。这样，将特征二值化相当于对一些 $i\in[d]$ 使用 $\mathbb{1}_{[x_i=1]}$ 形式的拆分规则。比如，我们在构建“水瓜决策树”之前假定水瓜用二维位向量 $x\in\{0,1\}^2$ 表示，x_1 表示水瓜的颜色是否在浅绿到浅黄的范围之内，x_2 表示用手掌压水瓜时是否会产生轻微的变形。用这种表示方法，结点“颜色?”可以用 $\mathbb{1}_{[x_1=1]}$ 来代替，结点“软硬程度?”可以用 $\mathbb{1}_{[x_2=1]}$ 来表示。虽然做了极

大的简化，但是接下来我们给出的算法和分析可以扩展到更一般的形式。

基于前面的简化假设，假设类变成了有限假设类，但是数量依然很大。任何从$\{0,1\}^d$到$\{0,1\}$的分类问题都可以用有2^d个叶子结点深度为$d+1$的决策树表示出来。因此，其VC维是2^d，也就是说PAC学习一个假设类需要的样本数量随2^d增长。除非d很小，否则需要大量的训练样本。

为了解决这个问题，我们需要利用第7章所述的MDL方案。根据潜在的先验知识，相比于规模大的决策树，我们更倾向于规模小的决策树。为了形式化地表示这种直觉，我们需要先给决策树定义一种描述语言，这种描述语言是无前缀的，并且对于规模小的决策树其描述长度要短。这里给出一种可能的方法：有n个结点的树用$n-1$块组成，每一块用$\log_2(d+3)$位来表示。前n块以深度优先的方式编码树的结点，最后一块标记编码的结束。每一块表明当前的结点是否为：

- 某一特征$\mathbf{1}_{[x_i=1]}$形式的中间结点
- 值为1的叶子结点
- 值为0的叶子结点
- 代码终止

总共有$d+3$种选项，因此需要用$\log_2(d+3)$位来表示每一块。 213

假定每个中间结点有两个子结点[1]，不难证明这是树的一种无前缀编码，有n个结点树的描述长度是$(n-1)\log_2(d+3)$。

通过定理7.7，我们可以得出，样本数量为m，对于任意的n和任意的有n个结点的决策树$h\in\mathcal{H}$，下式以不小于$1-\delta$的概率成立：

$$L_{\mathcal{D}}(h)\le L_S(h)+\sqrt{\frac{(n+1)\log_2(d+3)+\log(2/\delta)}{2m}} \tag{18.1}$$

这个上界存在一个折中：一方面我们希望更大规模更复杂的决策树减小训练误差$L_S(h)$，但是相应的n的值会变大。另一方面，规模小的决策树对应的n值较小，但是训练误差$L_S(h)$会变大。我们希望能找到一个决策树训练误差$L_S(h)$较小，同时结点数n也不至于太大，这样才能获得较低的真实风险$L_{\mathcal{D}}(h)$。

18.2 决策树算法

公式(18.1)中$L_{\mathcal{D}}(h)$的界给出了决策树的一种学习规则——使公式(18.1)右边最小的树即为所求的决策树。可惜，已经证明解该问题是计算难的[2]。因此，实际的决策树学习算法是基于启发式思想比如贪婪方法，逐步构建决策树，在每个结点采用局部最优策略。这种算法不能保证返回全局最优的决策树，但是在实践中取得不错效果。

决策树生长过程的总体框架如下。一棵树从单叶子结点(根结点)开始，将实例数最多的类作为该叶子结点的类标记。我们现在做一系列的迭代，每次迭代，测试拆分一个叶子结点的效果。我们定义一些"增益"指标来量化由此拆分带来的提升效果。然后，在所有可能的拆分中，或者选择最大化增益的拆分方式或者选择不拆分。

接下来，我们提供一种可能的实现方式。介绍一种叫ID3(Iterative Dichotomizer 3)的决策树算法。在描述算法时，我们假定实例是二值特征，即$\mathcal{X}=\{0,1\}^d$，因此所有的拆分

---

[1] 我们可以不失一般性地假设，因为如果决策结点只有一个子结点，可以用子结点代替它，而不会影响决策树的规模。

[2] 更确切地，如果NP≠P，那么没有算法可以在n，d，m的多项式时间内求解方程(18.1)。

规则都是$\mathbb{I}_{\{x_i=1\}}$形式，其中$i\in[d]$。我们会在18.2.3节讨论实值特征的情况。

算法是一个递归调用的方法，最开始调用ID3(S，$[d]$)并返回一棵决策树。在后面的伪代码中，我们会调用程序Gain(S，i)，该程序输入训练集S和索引i，评估根据第i个特征拆分之后的增益。我们会在18.2.1节介绍几种增益测量方法。

[211]

ID3(S，A)

输入：训练数据S，特征子集$A\subseteq[d]$

如果S中的所有样本都标号为1，返回一个叶子1

如果S中的所有样本都标号为0，返回一个叶子0

如果$A=\emptyset$，返回一个叶子节点，将S中标号最多的类作为该节点的类标号

　否则：

　　令$j=\operatorname{argmax}_{i\in A}\mathrm{Gain}(S,i)$

　　如果S中的所有样本标号相同

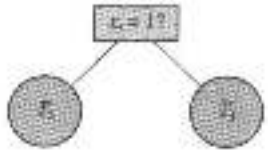
　　返回叶子节点，其值为S中标号最多的类

　　否则

　　T_1为ID3($\{(x,y)\in S:x_j=1\}$，$A\setminus\{j\}$)返回的树

　　T_2为ID3($\{(x,y)\in S:x_j=0\}$，$A\setminus\{j\}$)返回的树

　　返回树：



18.2.1　增益测量的实现方式

不同的算法采用不同的增益测量方式Gain(S，i)。在这里，我们给出三种方式。我们用符号$\mathbb{P}_S[F]$来表示在S上采用均匀分布事件发生的概率。

训练误差：增益的最简单定义是训练误差的减少量。设$C(a)=\min\{a,1-a\}$。根据第i个特征拆分之前的训练误差是$C(\mathbb{P}_S[y=1])$，因为我们采用投票数多的标签。同样，第i个特征拆分之后的错误率是

$$\mathbb{P}_S[x_i=1]C(\mathbb{P}_S[y=1\mid x_i=1])+\mathbb{P}_S[x_i=0]C(\mathbb{P}_S[y=1\mid x_i=0])$$

因此，我们可以定义增益为两者的差，即

$$\begin{aligned}\mathrm{Gain}(S,i)&:=C(\mathbb{P}_S[y=1])\\&\quad-(\mathbb{P}_S[x_i=1]C(\mathbb{P}_S[y=1\mid x_i=1])+\mathbb{P}_S[x_i=0]C(\mathbb{P}_S[y=1\mid x_i=0]))\end{aligned}$$

信息增益：另一种流行的增益测量方法是信息增益，被Quinlan用在ID3和C4.5算法中。信息增益是结点拆分前后熵的差值，实现方式是将前面表达式中的函数C替换为熵函数

[215]

$$C(a)=-a\log(a)-(1-a)\log(1-a)$$

基尼系数：另一种增益的定义是基尼系数，由Breiman，Friedman，Olshen和Stone(1984)在CART算法中使用。

$$C(a)=2a(1-a)$$

信息增益和基尼系数都是平滑的凹函数，对训练误差有上界。这些特性在特定情形下有很大的优点（参考 Kearns&Mansour(1996)）。

18.2.2 剪枝

前面所述的 ID3 算法存在很大的问题：返回的树规模很大。这样的树可能经验风险很低，但是它们的真实风险往往比较高——不论是根据理论分析还是在实际操作中。一种解决方法是限制 ID3 算法的迭代次数，使树的结点有上限。另一种常用的方法是在树构建完成之后进行剪枝，希望使树的规模变小，同时能保持近似的经验误差。理论上讲，根据公式(18.1)给出的界，如果将 n 变小同时不怎么增加 $L_S(h)$ 的值，我们有可能得到一棵真实风险较小的决策树。

通常，剪枝是一个自下而上的过程，根据一些界或者 $L_S(h)$ 的估计值，可以将结点由其子树或单个叶子结点替代。下面给出了剪枝的一个常用模板伪代码。

一般剪枝过程

输入：

函数 $f(T, m)$（样本规模为 m，决策树广义误差的界或估计），树 T

对于树 T 叶子节点到根节点上任意的节点 j

找到使 $f(T', m)$ 最小的 T'，T' 是下列情况的一种：

将节点 j 替换为标号为叶子节点 1 后的树

将节点 j 替换为标号为叶子节点 0 后的树

将节点 j 替换为其左子树后的树

将节点 j 替换为其右子树后的树

当前树

$T \leftarrow T'$

18.2.3 实值特征基于阈值的拆分规则

在之前的章节，我们假定特征是二进制且拆分规则是 $\mathbb{1}_{[i]=1}$ 形式时，如何生成一棵决策树。现在，我们将前面的结论拓展到特征是实数，拆分规则是 $\mathbb{1}_{[i]<\theta}$ 的情况。这种拆分规则可以看做是决策桩，我们已经在第 10 章介绍过。 216

基本的思路是将问题简化为二值特征的情况。设 $x_1, \ldots, x_m$ 是训练集中的实例。对于每一个实值特征 i，将实例按第 i 个特征从小到大排序 $x_{1,i} \le \cdots \le x_{m,i}$。定义一系列阈值 $\theta_{0,i}, \ldots, \theta_{m-1,i}$，其中 $\theta_{j,i} \in (x_{j,i}, x_{j+1,i})$（在这里我们约定 $x_{0,i} = -\infty$，$x_{m+1,i} = +\infty$）。最后，对于每个 i 和 j 我们定义二值特征 $\mathbb{1}_{[x_i<\theta_{j,i}]}$。在构建完这些二值特征之后，就可以运行前一节所述的 ID3 程序了。很容易验证，对于任意一个原始特征是实值，采用基于阈值的拆分规则构建得到的决策树，我们能找到一个相同训练误差相同结点数目的基于二值特征构建得到的决策树。

如果实值特征维数是 d，样本数目是 m，构建得到的二值特征数目是 dm 个。计算每个特征的增益需要 $O(dm^2)$ 次运算。如果我们采用一种更聪明的方法，可以将运行时间降低到 $O(dm\log(m))$。这种思路类似于 10.1.1 节里实现决策桩 ERM 时采用的方法。

18.3 随机森林

前面已经提及，由任意规模的决策树构成的类，其 VC 维是无限的。因此我们要限制

决策树的规模。另一种降低过拟合风险的方法是将树进行集成。接下来我们将介绍由Breiman(2001)提出的随机森林方法。

一个随机森林是由一系列决策树构成的分类器，每棵树都是将算法 A 作用到训练集 S 和随机变量 θ 上，其中 θ 是从某一独立同分布采样得到的。随机森林的预测值是由每个树进行多数投票得到的。

为了明确一个特定的随机森林，我们需要定义算法 A 和作用于 θ 上的分布。有许多方法可以实现，在这里我们介绍一种情形。我们由以下方式生成 θ。首先，从训练集 S 随机采样一个子样本，即在训练集 S 上采用均匀分布采样到一个样本数目为 m' 的新的训练集 S'。第二，我们构建一个序列 $I_1, I_2, \cdots$，每个 I_i 是 d 的一个大小为 k 的子集。所有这些变量组成向量 θ。然后使用算法 A 根据训练集 S' 生成一棵决策树(例如使用 ID3 算法)，在拆分时，要求在子集 I_i 中选择使增益最大的特征。直观上讲，如果 k 很小，这种限制会防止过拟合。

18.4　小结

决策树是一种非常直观的分类器。如果程序员设计一个预测器，那么它很像一个决策树。一个有 k 个叶子结点的树其 VC 维是 k，我们给出了用 MDL 来学习一棵决策树的范例。决策树最主要的问题是该问题是计算难的，因此我们给出了几种启发式训练方法。

[217]

18.5　文献评注

Quinlan(1986)推导了许多决策树学习算法(比如 ID3 和 C4.5)。CART 算法是由 Breiman，Friedman，Olshen 和 Stone 提出(1984)。随机森林是由 Breiman(2001)提出。读者可以参考 Hastie、Tibshirani & Friedman(2001)和 Rokach(2007)做进一步阅读。

训练决策树的计算难度证明由 Hyafil 和 Rivest(1976)给出。

18.6　练习

18.1　1) 证明：任意的二分类器 $h:\{0,1\}^d \mapsto \{0,1\}$ 可以由至多 $d+1$ 层的决策树实现，决策树的中间节点 $i \in \{0, \cdots, d\}$ 可以表示为 $(x_i = 0?)$ 形式。

2) 推导如下结论：定义域为 $\{0,1\}^d$ 的决策树假设类，其 VC 维为 2^d。

18.2　(ID3 的次优性)

考虑如下训练集，其中 $\mathcal{X} = \{0,1\}^3$，$\mathcal{Y} = \{0,1\}$：

$$((1,1,1),1)$$
$$((1,0,0),1)$$
$$((1,1,0),0)$$
$$((0,0,1),0)$$

假定用该训练集来训练一深度为 2 的决策树(即，对每个输入，在判别标号之前，我们可以问形如 $(x_i = 0?)$ 的两个问题)。

1) 假定运行 ID3 算法得到一决策树，其深度至多为 2(即根据算法选择根节点及其子节点，根据每个子树的多数标签终止算法和选择叶子节点，而不是继续递归循环)。假定使用熵函数(因此我们测量信息增益)来评估每个特征的质量，如果两个特征得到相同的信息增益，随机挑选其中一个，证明得到的决策树其训练误差至少为 1/4。

[218]

2) 寻找一深度为 2 的决策树，使其训练误差为 0。

最 近 邻

最近邻算法是最简单的机器学习算法。其思想是先存储训练集，然后以训练集中新实例最近邻的标签来预测新实例标签。这种方法的合理性基于一种假设，这种假设认为用于描述域点的特征与其标签相关，邻近的点之间可能具有相同的标签。更进一步说，在某些情况下，即使训练集庞大，完成一个最近邻的寻找会非常快(例如，当训练集是所有网页，以链接作为距离)。

注意到，相比我们目前讨论过的算法范例(ERM、SRM、MDL 或 RLM 都是由一些假设类$\mathcal{H}$决定的)，最近邻方法无需在指定的函数类里搜索预测器，就可以找出任何测试点的标签。

在本章，我们描述关于分类和回归问题的最近邻方法。我们就简单的二分问题分析其性能，并讨论执行这些方法的效率。

19.1 k 近邻法

本章，假设域上给定一个度量函数 ρ。也就是说，$\rho:\mathcal{X}\times\mathcal{X}\to\mathbb{R}$ 是一个返回任意两点距离的函数。例如，如果$\mathcal{X}=\mathbb{R}^d$，那么 ρ 可以是欧氏距离

$$\rho(x,x')=\|x-x'\|=\sqrt{\sum_{i=1}^{d}(x_i-x'_i)^2}$$

令 $S=(x_1,y_1),\dots,(x_m,y_m)$是一个训练样本序列。对每一个 $x\in\mathcal{X}$，令 $\pi_1(x),\dots,\pi_m(x)$是按样本与 x 的距离对$(1,\dots,m)$的重排序列。也就是说，对所有的 $i<m$，有

$$\rho(x,x_{\pi_i(x)})\le\rho(x,x_{\pi_{i+1}(x)})$$

219

对于一个数 k，k-NN 准则对二分问题的定义如下：

k-NN

输入：一个训练样本集 $S=(x_1,y_1),\dots,(x_m,y_m)$

输出：对每个点 $x\in\mathcal{X}$

返回$\{y_{\pi_i(x)}:i\le k\}$中的投票最多的标签

当 $k=1$ 时，我们有 1-NN 准则：

$$h_S(x)=y_{\pi_1(x)}$$

关于 1-NN 准则的几何说明见图 19.1。

对于回归问题，即$\mathcal{Y}=\mathbb{R}$，你可以定义预测为 k 个近邻的平均目标值，也就是，$h_S(x)=\frac{1}{k}\sum_{i=1}^{k}y_{\pi_i(x)}$。更一般地，对某个函数 ϕ：$\bigcup_{i=1}^{\infty}(\mathcal{X}\times\mathcal{Y})^i\to\mathcal{Y}$，关于 ϕ 的 k-NN 准则是：

$$h_S(x)=\phi((x_{\pi_1(x)},y_{\pi_1(x)}),\dots,(x_{\pi_k(x)},y_{\pi_k(x)})) \tag{19.1}$$

容易验证，我们可以通过最多的投票(对于分类)或平均目标值(对于回归)来预测，按式(19.1)选择一个合适的 ϕ。这种推广可以引申到其他准则，比如，若$\mathcal{Y}=\mathbb{R}$，我们可以根据

样本与测试点 x 的距离，采用一个目标值的加权平均作为预测：

$$h_S(x)=\sum_{i=1}^{k}\frac{\rho(x,x_{\pi_i(x)})}{\sum_{j=1}^{k}\rho(x,x_{\pi_j(x)})}y_{\pi_i(x)}$$

图 19.1　1-NN 准则的决策边界说明。画出的点是样本点，任何新实例点的预测标签与其所属的单元中心样本点的标签一样。这些单元被称为 Voronoi Tessellation 空间

19.2　分析

由于最近邻准则是如此自然的学习方法，其泛化性已有大量研究。此前大部分研究的结论是渐进相容性结果，分析了当样本大小 m 趋于无穷时最近邻准则的性能和依赖于潜在分布的收敛速度。正如我们在 7.4 小节里讨论的一样，仅有这类的分析还是不够的。我们想要从有限训练样本里学习并理解泛化性，这种泛化性是关于有限训练集大小和清晰的数据分布先验的函数。因此我们提供了一个关于 1-NN 准则的有限样本分析，说明误差是随 m 递减的函数且依赖于分布的性质。我们也将解释，这种分析能推广到任意 k 时的 k-NN 准则，特别地，分析指定了达到真实误差 $2L_{\mathcal{D}}(h^*)+\varepsilon$ 时所需要的样本数，其中 h^* 是贝叶斯最优假设，假设这种标记准则是"表现良好的"（我们将在后面定义）。

[220]

19.2.1　1-NN 准则的泛化界

我们现在分析 0-1 损失下二分问题的 1-NN 准则的真实误差，也就是说，$\mathcal{Y}=\{0,1\}$ 且 $\ell(h,(x,y))=\mathbb{1}_{[h(x)\neq y]}$。假设在整个分析中 $\mathcal{X}=[0,1]^d$ 且 ρ 是欧氏距离。

我们先介绍一些概念。令 $\mathcal{D}$ 是关于 $\mathcal{X}\times\mathcal{Y}$ 的一个分布。用 $\mathcal{D}_{\mathcal{X}}$ 表示关于 $\mathcal{X}$ 的一个边缘分布，$\eta:\mathbb{R}^d\to\mathbb{R}$ 表示关于标签的条件概率，即

$$\eta(x)=\mathbb{P}[y=1\mid x]$$

回顾贝叶斯最优准则（即，所有函数中使得 $L_{\mathcal{D}}(h)$ 最小的假设）：

$$h^*(x)=\mathbb{1}_{[\eta(x)\geq 1/2]}$$

我们假设条件概率函数 η 对任意 $c>0$ 都满足 c-利普希茨性，即对所有的 $x,x'\in\mathcal{X}$，$|\eta(x)-\eta(x')|\leq c\|x-x'\|$。换言之，这种假设意味着，如果两个向量彼此相近，那么它们的标签更可能一致。

下面的引理将条件概率函数的利普希茨性应用到 1-NN 准则的真实误差上界分析中，其中这个误差是由每个测试样例与其在近邻点之间的距离所确定的函数。

引理 19.1　令 $\mathcal{X}=[0,1]^d$，$\mathcal{Y}=\{0,1\}$，且 $\mathcal{D}$ 是 $\mathcal{X}\times\mathcal{Y}$ 上的一个分布，在这个分布上条件概率函数 η 是一个 c-利普希茨性函数。令 $S=(x_1,y_1),\ldots,(x_m,y_m)$ 是独立同分布样本，并令 h_S 是其对应的 1-NN 假设。用 h^* 表示关于 η 的贝叶斯最优准则。则有

$$\mathbb{E}_{S\sim\mathcal{D}^m}[L_{\mathcal{D}}(h_S)]\leq 2L_{\mathcal{D}}(h^*)+c\,\mathbb{E}_{x\sim\mathcal{D}^m,x\sim\mathcal{D}}[\|x-x_{\pi_1(x)}\|]$$

证明 由于$\ell_D(h_S)=\mathbb{E}_{(x,y)\sim D}\mathbb{1}_{[h_S(x)\neq y]}$，我们可得$\mathbb{E}_S[L_D(h_S)]$表示采样到一个训练集$S$和一个额外样本$(x,y)$的概率，其中$x_1(x)$与$y$不同。换言之，我们可以根据$\mathcal{D}_x$先采样到$m$个无标签样本$S_x=(x_1,\dots,x_m)$，以及一个额外的无标签样本$x\sim\mathcal{D}_x$，然后找到$x$在 221
S_x中最近邻$x_1(x)$，最后采样$y\sim\eta(x)$和$y_{x_1(x)}\sim\eta(x_1(x))$。由此得

$$\begin{aligned}\mathbb{E}_S[L_D(h_S)]&=\mathop{\mathbb{E}}_{S_x\sim\mathcal{D}_x^m,x\sim\mathcal{D}_x,y\sim\eta(x),y'\sim\eta(x_1(x))}[\mathbb{1}_{[y\neq y']}]\\&=\mathop{\mathbb{E}}_{S_x\sim\mathcal{D}_x^m,x\sim\mathcal{D}_x}\Big[\mathop{\mathbb{P}}_{y\sim\eta(x),y'\sim\eta(x_1(x))}[y\neq y']\Big]\end{aligned}\tag{19.2}$$

对于任意两个域点x，x'，我们接下来确定$\mathbb{P}_{y\sim\eta(x),y'\sim\eta(x')}[y\neq y']$的上界：

$$\begin{aligned}\mathop{\mathbb{P}}_{y\sim\eta(x),y'\sim\eta(x')}[y\neq y']&=\eta(x')(1-\eta(x))+(1-\eta(x'))\eta(x)\\&=(\eta(x)-\eta(x)-\eta(x'))(1-\eta(x))\\&\quad+(1-\eta(x)+\eta(x)-\eta(x'))\eta(x)\\&=2\eta(x)(1-\eta(x))+(\eta(x)-\eta(x'))(2\eta(x)-1)\end{aligned}$$

利用$|2\eta(x)-1|\leqslant 1$和η是c-利普希茨性的假设，我们可得最大概率：

$$\mathop{\mathbb{P}}_{y\sim\eta(x),y'\sim\eta(x')}[y\neq y']\leqslant 2\eta(x)(1-\eta(x))+c\|x-x'\|$$

将上式与式(19.2)相加，我们推断出

$$\mathop{\mathbb{E}}_S[L_D(h_S)]\leqslant\mathop{\mathbb{E}}_x[2\eta(x)(1-\eta(x))]+c\mathop{\mathbb{E}}_{S,x}[\|x-x_{1(x)}\|]$$

最终，贝叶斯最优分类器误差是

$$L_D(h^*)=\mathop{\mathbb{E}}_x[\min\{\eta(x),1-\eta(x)\}]\geqslant\mathop{\mathbb{E}}_x[\eta(x)(1-\eta(x))]$$

结合上面两个不等式，定理得证。 ■

接下来，对于一个随机样本，确定其与S中最近邻之间距离期望的界。我们首先需要下面的概率引理。这个引理确定了了集不被随机采样的概率权重，这种概率权重是样本大小的一个函数。

引理 19.2 令$C_1,\dots,C_r$是某个域$\mathcal{X}$上子集的集合，根据$\mathcal{X}$的某个概率分布$\mathcal{D}$，令S是采样得到一组m个独立同分布样本的序列。则有

$$\mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}\Big[\sum_{i:C_i\cap S=\emptyset}\mathbb{P}[C_i]\Big]\leqslant\frac{r}{me}$$

证明 从期望的线性性质出发，我们可改写：

$$\mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}\Big[\sum_{i:C_i\cap S=\emptyset}\mathbb{P}[C_i]\Big]=\sum_{i=1}^r\mathbb{P}[C_i]\mathop{\mathbb{E}}_S[\mathbb{1}_{[C_i\cap S=\emptyset]}]$$

然后，对每一i我们有

$$\mathop{\mathbb{E}}_S[\mathbb{1}_{[C_i\cap S=\emptyset]}]=\mathop{\mathbb{P}}_S[C_i\cap S=\emptyset]=(1-\mathbb{P}[C_i])^m\leqslant e^{-m\mathbb{P}[C_i]}$$ 222

结合前面两等式，我们得

$$\mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}\Big[\sum_{i:C_i\cap S=\emptyset}\mathbb{P}[C_i]\Big]\leqslant\sum_{i=1}^r\mathbb{P}[C_i]e^{-m\mathbb{P}[C_i]}\leqslant r\max_i\mathbb{P}[C_i]e^{-m\mathbb{P}[C_i]}$$

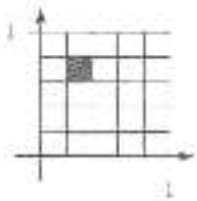
最终，计算得$\max_a ae^{-ma}\leqslant\frac{1}{me}$，定理得证。 ■

用前面的引理，我们可以容易地陈述和证明这一小节的主要结论——1-NN学习准则的期望误差上界。

引理 19.3　令$\mathcal{X}=[0,1]^d$，$\mathcal{Y}=\{0,1\}$，且$\mathcal{D}$是$\mathcal{X}\times\mathcal{Y}$上的一个分布，在这个分布上条件概率函数$\eta$是$c$-利普希茨函数。用$h_S$表示采样$S\sim\mathcal{D}^m$时，应用1-NN准则得到的输出假设，则有

$$\mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}[L_{\mathcal{D}}(h_S)]\leqslant 2L_{\mathcal{D}}(h^*)+4c\sqrt{d}m^{-\frac{1}{d+1}}$$

证明　对某个T，固定$\varepsilon=1/T$，令$r=T^d$，$C_1,\cdots,C_r$是长为ε的格，构成对集合的覆盖：即，对每个$(\alpha_1,\cdots,\alpha_d)\in[T]^d$，存在一个形式为$\{x:\forall j,\ x_j\in[(\alpha_j-1)/T,\alpha_j/T]\}$的集合$C_i$。当$d=2$，$T=5$时，对应$\alpha=(2,4)$的集合说明图如下：



对同一个框里的每对x，x'，我们有$|x-x'|\leqslant\sqrt{d}\varepsilon$。否则，$|x-x'|\leqslant\sqrt{d}$。因此，

$$\mathop{\mathbb{E}}_{x\sim\mathcal{D}}[\|x-x_{\eta(x)}\|]\leqslant\mathop{\mathbb{E}}_{x\sim\mathcal{D}}\Big[\mathbb{P}\Big[\bigcup_{i:C_i\cap S=\emptyset}C_i\Big]\sqrt{d}+\mathbb{P}\Big[\bigcup_{i:C_i\cap S\neq\emptyset}C_i\Big]\varepsilon\sqrt{d}\Big]$$

并结合引理19.2和$\mathbb{P}\Big[\bigcup_{i:C_i\cap S=\emptyset}C_i\Big]\leqslant 1$，我们得到

$$\mathop{\mathbb{E}}_{x\sim\mathcal{D}}[\|x-x_{\eta(x)}\|]\leqslant\sqrt{d}\Big(\frac{r}{m\varepsilon}+\varepsilon\Big)$$

由于任的数目是$r=(1/\varepsilon)^d$，我们有

$$\mathop{\mathbb{E}}_{x\sim\mathcal{D}}[\|x-x_{\eta(x)}\|]\leqslant\sqrt{d}\Big(\frac{2^d\varepsilon^{-d}}{m\varepsilon}+\varepsilon\Big)$$

结合前面的引理19.1我们得到

223

$$\mathop{\mathbb{E}}_{S}[L_{\mathcal{D}}(h_S)]\leqslant 2L_{\mathcal{D}}(h^*)+c\sqrt{d}\Big(\frac{2^d\varepsilon^{-d}}{mc}+\varepsilon\Big)$$

最后，设置$\varepsilon=2m^{-1/(d+1)}$并有

$$\frac{2^d\varepsilon^{-d}}{m\varepsilon}+\varepsilon=\frac{2^d2^{-d}m^{d/(d+1)}}{mc}+2m^{-1/(d+1)}=m^{-1/(d+1)}(1/c+2)\leqslant 4m^{-1/(d+1)}$$

证明完成。 ■

这个定理表明，如果我们先固定生成数据的分布，让m趋于无穷，则1-NN准则的误差收敛于两倍贝叶斯误差。这个分析可以推广到较大值的k，并说明k-NN准则的期望误差收敛于$(1+\sqrt{8/k})$倍贝叶斯分类器误差。这一结论以定理19.5给出，其证明留作练习。

19.2.2　"维数灾难"

定理19.3中给出的上界随c（η的利普希茨系数）和域集的欧氏维数d增长。实际上易得，定理19.3中最后一项小于ε的必要条件是$m\geqslant(4c\sqrt{d}/\varepsilon)^{d+1}$。即，训练集的大小应该随着维数的增加呈指数递增。以下的定理说明，这不仅是一个构造的上界，对某些分布而言，样本的数量确实是NN准则学习所必需的。

定理19.4　对任意$c>1$和每个学习准则L，存在一个分布$[0,1]^d\times\{0,1\}$，使得

$\eta(x)$是c-利普希茨的，且分布的贝叶斯误差是0，但对样本数$m<(c+1)^d/2$，准则L的真实误差大于1/4。

证明 固定任意的值c和d。令G_c^d是$[0,1]^d$上距离$1/c$的点构成的网格，即，网格上的每个点形式为$(a_1/c,\cdots,a_d/c)$，其中a_i属于$\{0,\cdots,c-1,c\}$。注意到，由于网格上任意不同两点至少距离$1/c$，任意函数$\eta:G_c^d\to[0,1]$是c-利普希茨函数。因此，在G_c^d中的c-利普希茨函数集合包含该域上所有二值函数。因此我们可以运用"没有免费的午餐"定理(定理5.1)，得到对学习假设类所需样本大小的下界。在网格上的点数为$(c+1)^d$，如果$m<(c+1)^d/2$，定理5.1给出了我们所要确定的下界。 ■

这种对维数的指数依赖性被称为"维数灾难"。我们看到，如果样本数小于$\Omega((c+1)^d)$，1-NN准则可能会失败。因此，尽管1-NN准则没有限制在一个指定的假设集里，但它仍然依赖于先验知识——学习的成功仅依赖于维数和潜在分布的利普希茨常数η不会太大。 224

19.3 效率实施*

最近邻法是一种记忆学习类的准则，这种方法要求存储整个训练数据集，在测试时为了找到最近邻，需要浏览所有样本，应用NN准则的时间因此是$\Theta(dm)$。这导致测试计算的开销很大。

当d很小，计算几何领域中的一些结论已经提出，结合相关数据结构能够确保应用NN准则能在$o(d^{O(1)}\log(m))$时间内完成。然而，数据结构要求的空间为$m^{O(d)}$，这对较大的d值是不现实的。

为了克服这一问题，可考虑一个近似搜索来改善搜索方法。形式上，r-近似搜索程序确保检索到的点距离测试样本至多是最近邻样本距离测试样本的r倍。三种流行的关于NN的近似算法是kd-树、球树和局部敏感哈希(LSH)。具体可以参考Shakhnarovich, Darrell & Indyk(2006)。

19.4 小结

k-NN准则是一种简单的学习算法，这种学习算法依赖于假设"看起来相似的事物一定是相似的"。我们利用条件概率函数的利普希茨性阐述了这一(直觉)判断。我们已经表明，对于一个足够大的训练集，1-NN的风险上界是两倍贝叶斯最优准则风险。我们也推导了一个下界，说明了"维数灾难"——所需要的样本大小最终随维数的增加呈指数增长。因此，实际中NN通常在维数约简预处理步骤之后执行。我们后面将在23章讨论维数约简技术。

19.5 文献评注

Cover和Hart(1967)给出了1-NN的最早分析，显示了在一定情况下其风险收敛于两倍贝叶斯最优误差。Stone(1977)给出了引理，Devroye和Györfi(1985)证明了k-NN准则是渐进收敛的(关于所有从$\mathbb{R}^d$到$\{0,1\}$的函数的假设类)。一个较全面的分析在Devroye等(1996)的书中给出。这里，我们给出一个有限样本集，确保明确地强调了关于分布的先验假设。关于渐进收敛性结论的讨论见小节7.4。最后，Gottlieb, Kontorovich和Krauthgamer(2010)推导了另一个关于NN的有限样本界，这个界与VC界类似。

19.6 练习

在练习中，我们将证明以下关于k-NN准则的定理。

定理 19.5　令$\mathcal{X}=[0,1]^d$，$\mathcal{Y}=\{0,1\}$，且$\mathcal{D}$是$\mathcal{X}\times\mathcal{Y}$上的一个分布，在这个分布上条件概率函数$\eta$是$c$-利普希茨函数。并令$h_S$表示选择$S\sim\mathcal{D}^m$时，应用$k$-NN准则的输出假设，其中$k\geq 10$，用$h^*$表示关于$\eta$的贝叶斯最优准则。则有

[225]

$$\mathop{\mathbb{E}}_{S}[L_{\mathcal{D}}(h_S)]\leq\left(1+\sqrt{\frac{8}{k}}\right)L_{\mathcal{D}}(h^*)+(4c\sqrt{d}+k)m^{-1/(d+1)}$$

19.1　证明了以下引理。

引理 19.6　令$C_1,\ldots,C_r$是某个域$\mathcal{X}$上子集的集合。令S是根据关于$\mathcal{X}$的某个概率分布$\mathcal{D}$采样得到的一个长为m个独立同分布样本序列。则有

$$\mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}\Big[\sum_{i:|C_i\cap S|<k}\mathbb{P}[C_i]\Big]\leq\frac{2rk}{m}$$

提示：

- 证明

$$\mathop{\mathbb{E}}_{S}\Big[\sum_{i:|C_i\cap S|<k}\mathbb{P}[C_i]\Big]=\sum_{i=1}^{r}\mathbb{P}[C_i]\mathop{\mathbb{P}}_{S}[|C_i\cap S|<k]$$

- 固定某个i并假设$k<\mathbb{P}[C_i]m/2$，利用切比雪夫界说明

$$\mathop{\mathbb{P}}_{S}[|C_i\cap S|<k]\leq\mathop{\mathbb{P}}_{S}[|C_i\cap S|<\mathbb{P}[C_i]m/2]\leq e^{-\mathbb{P}[C_i]m/8}$$

- 利用不等式$\max_{a\in\mathbb{R}^+}ae^{-a}\leq\frac{1}{me}$说明对这样的$i$，我们有

$$\mathbb{P}[C_i]\mathop{\mathbb{P}}_{S}[|C_i\cap S|<k]\leq\mathbb{P}[C_i]e^{-\mathbb{P}[C_i]m/8}\leq\frac{8}{me}$$

- 证明可借助以下事实：对$k\geq\mathbb{P}[C_i]m/2$的情况，我们显然有

$$\mathbb{P}[C_i]\mathop{\mathbb{P}}_{S}[|C_i\cap S|<k]\leq\mathbb{P}[C_i]\leq\frac{2k}{m}$$

19.2　我们标记$y\sim p$作为"y是一个期望为p的伯努利随机变量"。证明以下引理：

引理 19.7　令$k\geq 10$，并令$Z_1,\ldots,Z_k$是满足$\mathbb{P}[Z_i=1]=p_i$的独立伯努利随机变量。表示$p=\frac{1}{k}\sum_i p_i$且$p'=\frac{1}{k}\sum_{i=1}^{k}Z_i$。证明：

$$\mathop{\mathbb{E}}_{Z_1,\ldots,Z_k,y\sim p}[|y\neq\mathbb{1}_{[p'>1/2]}|]\leq\left(1+\sqrt{\frac{8}{k}}\right)\mathop{\mathbb{P}}_{y\sim p}[|y\neq\mathbb{1}_{[p>1/2]}|]$$

提示：

不失一般性假设$p\leq 1/2$，而$\mathbb{P}_{y\sim p}[y\neq\mathbb{1}_{[p>1/2]}]=p$。令$y'=\mathbb{1}_{[p'>1/2]}$。

- 证明

$$\mathop{\mathbb{E}}_{Z_1,\ldots,Z_k,y\sim p}[|y\neq y'|]=p+\mathop{\mathbb{E}}_{Z_1,\ldots,Z_k}[|p'>1/2|](1-2p)$$

- 利用切比雪夫界（引理 B.3）证明：

$$\mathbb{P}[p'>1/2]\leq e^{-k\cdot h(1/2p-1)}$$

其中

[226]

$$h(a)=(1+a)\log(1+a)-a$$

- 读者可以通过以下不等式（无需证明）完成证明：对每个$p\in[0,1/2]$和$k\geq 10$，

$$(1-2p)e^{-k\cdot h(1/2p-1)}\leq\sqrt{\frac{8}{k}}p$$

19.3 固定某个 p，$p'\in[0,1]$和 $y'\in\{0,1\}$。证明：

$$\mathop{\mathbb{P}}_{y\sim p}[y\neq y']\leqslant \mathop{\mathbb{P}}_{y\sim p'}[y\neq y']+|p-p'|$$

19.4 根据以下步骤完成定理的证明：

- 同定理 19.3 的证明，固定某个 $\varepsilon>0$ 并令 $C_1,\cdots,C_r$ 是长为 ε 的框，构成对集合 $\mathcal{X}$ 的覆盖。对同一个框里的每对 x、x'，我们有 $|x-x'|\leqslant\varepsilon\sqrt{d}$。否则，$|x-x'|\leqslant 2\sqrt{d}$。证明：

$$\mathop{\mathbb{E}}_{S}[L_{\mathcal{D}}(h_S)]\leqslant \mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}\Big[\sum_{i:|C_i\cap S|<k}\mathbb{P}[C_i]\Big]$$
$$+\max_i \mathop{\mathbb{P}}_{S,(x,y)}\Big[h_S(x)\neq y \mid \forall j\in[k],|x-x_{\pi_j(x)}|\leqslant\varepsilon\sqrt{d}\Big] \qquad (19.3)$$

- 利用引理 19.6 确定第一个加数的界。
- 为确定第二个加数的界，我们固定 $S|_x$ 和 x 使得 x 在 $S|_x$ 中的所有 k 个近邻与 x 的最远距离不超过 $\varepsilon\sqrt{d}$。不失一般性，假设 kNN 是 $x_1,\cdots,x_k$。表示 $p_i=\eta(x_i)$ 且令 $p=\frac{1}{k}\sum_i p_i$。运用练习 19.3 证明：

$$\mathop{\mathbb{E}}_{y_1,\cdots,y_k}\mathop{\mathbb{P}}_{y\sim\mathcal{D}(x)}[h_S(x)\neq y]\leqslant \mathop{\mathbb{E}}_{y_1,\cdots,y_k}\mathop{\mathbb{P}}_{y\sim p}[h_S(x)\neq y]+|p-\eta(x)|$$

不失一般性地假设 $p\leqslant 1/2$。利用引理 19.7 证明

$$\mathop{\mathbb{P}}_{y_1,\cdots,y_k}\mathop{\mathbb{P}}_{y\sim p}[h_S(x)\neq y]\leqslant\Big(1+\sqrt{\frac{8}{k}}\Big)\mathop{\mathbb{P}}_{y\sim p}[\mathbf{1}_{[p>1/2]}\neq y]$$

- 证明

$$\mathop{\mathbb{P}}_{y\sim p}[\mathbf{1}_{[p>1/2]}\neq y]=p=\min\{p,1-p\}\leqslant\min\{\eta(x),1-\eta(x)\}+|p-\eta(x)|$$

- 结合前面的结论可得到等式(19.3)中第二个加数的界为

$$\Big(1+\sqrt{\frac{8}{k}}\Big)L_{\mathcal{D}}(h^*)+3c\varepsilon\sqrt{d}$$

- 运用 $r=(2/\varepsilon)^d$ 可得：

$$\mathop{\mathbb{E}}_{S}[L_{\mathcal{D}}(h_S)]\leqslant\Big(1+\sqrt{\frac{8}{k}}\Big)L_{\mathcal{D}}(h^*)+3c\varepsilon\sqrt{d}+\frac{2(2/\varepsilon)^d k}{m}$$

设 $\varepsilon=2m^{-1/(d+1)}$ 并运用

$$6cm^{-1/(d+1)}\sqrt{d}+\frac{2k}{c}m^{-1/(d+1)}\leqslant(6c\sqrt{d}+k)m^{-1/(d+1)}$$

证毕。

227

神经元网络

人工神经网络是一种计算模型，它以大脑的神经网络结构为原型。在大脑的简化模型中，它包含了大量的基本计算器件(神经元)，以复杂的通信网络形式彼此互联，通过它们，大脑能够执行高度复杂的计算。人工神经网络是规则的计算结构，它模仿大脑的计算框架建模。

神经网络学习的提出始于 20 世纪中期，由它产生的有效学习框架，现在被证明在某些问题上具有领先的性能。

神经网络可以被描述为有向图，其中节点是神经元，边是它们之间的连接。每个神经元的输入是与其输入边相连的神经元输出的加权和。我们关注前馈网络，其中不包含环。

在机器学习的背景下，我们定义含有神经网络预测的假设类，其中所有的假设类共享网络结构，但边上的权重不同。我们将在 20.3 节中说明，每个超过 n 个变量的能够在 $T(n)$ 时间内执行的预测都能表示成复杂度为 $O(T(n)^2)$ 的神经网络预测，其中网络的大小是内部的节点数。因此多项式规模的神经网络假设类足够完成所有实际的学习任务。更进一步，在 20.4 节我们将看到该类学习问题的样本复杂度相对于网络大小也是有界的。因此，这似乎是我们最希望适应的学习框架，因为它在所有可以高效执行的假设类中，具有多项式的样本复杂度和最小的近似误差。

[228] 需要提醒的是，训练神经网络假设类的计算代价是相当大的，这将在 20.5 节中给出形式化说明。一个广泛应用的神经网络启发式方法基于 14 章中介绍的 SGD 框架。我们已经证明在凸损失函数的条件下，SGD 是有效的方法。在神经网络中，损失函数是高度非凸的。然而，我们仍可以采用 SGD 算法并希望它能够找到可行解(就像许多实际学习任务中所表现的)。在 20.6 节中，我们说明如何在神经网络上应用 SGD 方法。特别地，最复杂的操作是计算损失函数关于网络参数的梯度。我们给出有效计算梯度的反向传播算法。

20.1 前馈神经网络

神经网络隐含的思路是许多神经元能够通过信息互联共同执行复杂计算。将神经网络结构表达成图形是常见的，其中节点是神经元，图中的每个(直接相连的)边连接一些神经元的输出和另外神经元的输入。我们将把精力集中在前馈神经网络结构上，这些图中不含环。

前馈神经网络被描述为有向无环图 $G=(V, E)$，同时边上具有权重函数 $w: E \to \mathbb{R}$。图中的节点是神经元。每个单独的神经元被建模为简单标量函数 $\sigma: \mathbb{R} \to \mathbb{R}$。我们关注三个可能的 σ 函数：符号函数 $\sigma(a)=\operatorname{sign}(a)$，阈值函数 $\sigma(a)=\mathbb{1}_{[a>0]}$ 和 sigmoid 函数 $\sigma(a)=1/(1+\exp(-a))$(它是阈值函数的平滑估计)。我们将 σ 称为神经元的激活函数。图中的每个边连接一些神经元的输出和另一些神经元的输入。神经元的输入是所有与其相连的神经元输出的加权和，权重由 w 给定。

为了简化神经网络计算的描述，我们进一步假设网络是由层(layer)组织的。即网络的节点集合能够分解为独立非空的单元子集 $V=\cup_{j=0}^{T} V_j$。因此 E 中的每个边都连接着 V_j 与

V_{i+1} 中的节点($i\in[T]$)。最底层 V_0 叫做输入层，它包含 $n+1$ 个神经元，其中 n 是输入空间的维度。对于每个 $i\in[n]$，V_0 层中神经元 i 的输出为 x_i。V_0 层中最后一个神经元为常数神经元，它总是输出 1。我们用 $v_{i,j}$ 表示 i 层中第 j 个神经元，用 $o_{i,j}(x)$ 表示 $v_{i,j}$ 的输出，其中 x 为网络的输入向量。因此对于 $i\in[n]$ 有 $o_{0,i}(x)=x_i$ 且对于 $i=n+1$ 有 $o_{0,i}(x)=1$。我们现在开始逐层计算。假设已经计算出 i 层的神经元输出，然后，就可以计算出第 $i+1$ 层神经元的输出。固定 $v_{i+1,j}\in V_{i+1}$。网络输入向量为 x 时，令 $a_{i+1,j}(x)$ 表示 $v_{i+1,j}$ 的输入，那么

$$a_{i+1,j}(x)=\sum_{v_{i,l}\in V_i,\ (v_{i,l},v_{i+1,j})\in E}\omega((v_{i,l},v_{i+1,j}))o_{i,l}(x)$$

且

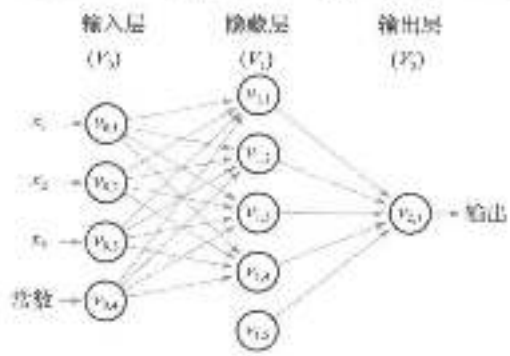
$$o_{i+1,j}(x)=\sigma(a_{i+1,j}(x))$$

229

即，$v_{i+1,j}$ 的输入是 V_i 中与 $v_{i+1,j}$ 相连的神经元输出的加权和，其中权重由 ω 给出，而 $v_{i+1,j}$ 的输出是激活函数作用在输入上的结果。

V_1，…，V_{T-1} 称为隐藏层。最顶层 V_T 称为输出层。在简单的预测问题中，输出层只包含一个神经元，它的输出为整个网络的输出。

T 是网络中的层数(不包含 V_0)，或者称为网络的深度。网络的体积(size)为 $|V|$。网络的宽度(width)是 $\max_i|V_i|$。图中的前馈神经网络深度为 2，体积为 10，宽度为 5。注意到隐藏层中的神经元没有输入边。这个神经元的输出恒为 $\sigma(0)$。



20.2 神经网络学习

一旦通过 (V,E,σ,ω) 指定了一个神经网络，我们将得到函数 $h_{V,E,\sigma,\omega}:\mathbb{R}^{|V_0|-1}\to\mathbb{R}^{|V_T|}$。该集合中的任何函数都可以作为学习的假设类。通常，我们通过设定图 (V,E) 和激活函数 σ 来定义神经网络预测的假设类，并使假设类为所有形如 $h_{V,E,\sigma,\omega}$ 的函数，其中 $\omega:E\to\mathbb{R}$。三元组 (V,E,σ) 称为网络的结构。我们认为假设类为

$$\mathcal{H}_{V,E,\sigma}=\{h_{V,E,\sigma,\omega}:\omega\ E\ 到\ \mathbb{R}\ 的映射\}\tag{20.1}$$

即指定假设类的参数是网络边上的权重。

我们现在可以研究假设类的逼近误差、估计误差和优化误差。在 20.3 节中，根据图的体积，我们通过分析 $\mathcal{H}_{V,E,\sigma}$ 中所采用的函数类型研究其逼近误差。在 20.4 节中，我们通过分析 VC 维研究 $\mathcal{H}_{V,E,\sigma}$ 在二分类问题(即 $V_T=1$，σ 为符号函数)中的估计误差。最后在 20.5 节中，我们说明即使图很小，$\mathcal{H}_{V,E,\sigma}$ 类学习也具有计算复杂性。在 20.6 节中我们给出最常用的启发式 $\mathcal{H}_{V,E,\sigma}$ 训练方法。

230

20.3　神经网络的表达力

本节中我们研究神经网络的表达力，即什么类型的函数能够应用在神经网络里，更具体地，我们将给定一些结构 V，E，σ，并研究$\mathcal{H}_{V,E,\sigma}$中能够实现什么样的体积为 V 的函数假设。

我们从$\mathcal{H}_{V,E,\sigma}$能够实现何种布尔函数(即函数映射从$\{\pm1\}^n$到$\{\pm1\}$)开始讨论。注意到对于任何以 b 个比特存储实数的计算机，当计算函数 $f:\mathbb{R}^n\to\mathbb{R}$ 时，我们实际上是计算 $g:\{\pm1\}^{nb}\to\{\pm1\}^b$。因此，研究$\mathcal{H}_{V,E,\mathrm{sign}}$能够实现何种布尔函数能够让我们知道计算机能够完成 b 比特实数的何种函数。

我们从一个简单的论断开始：在不约束网络体积时，深度为 2 的神经网络能够实现任何布尔函数。

论断 20.1　*对任意的 n，存在深度为 2 的图(V,E)，使得$\mathcal{H}_{V,E,\mathrm{sign}}$包含所有的$\{\pm1\}^n$到$\{\pm1\}$函数映射。*

证明　我们构建一个图，其中 $|V_0|=n+1$，$|V_1|=2^n+1$，$|V_2|=1$。令 E 为相邻层的所有可能的边。现在，令 $f:\{\pm1\}^n\to\{\pm1\}$为布尔函数。我们需要证明能够通过调节权重实现 f。令 $u_1,\cdots,u_k$ 为$\{\pm1\}^n$中所有使 f 输出为 1 的向量。观察到对于每个 i 和 $x\in\{\pm1\}^n$，如果 $x\neq u_i$ 那么$\langle x,u_i\rangle\leq n-2$，如果 $x=u_i$ 那么$\langle x,u_i\rangle=n$。可以推导出 $g_i(x)=\mathrm{sign}(\langle x,u_i\rangle-n+1)$为 1 当且仅当 $x=u_i$。进而我们可以调节 V_0 和 V_1 之间的权重使对于每个 $i\in[k]$，神经元 $v_{1,i}$实现函数 $g_i(x)$。下面，我们注意到 $f(x)$是对 $g_i(x)$的析取，因此可以写成

$$f(x)=\mathrm{sign}\Big(\sum_{i=1}^{k}g_i(x)+k-1\Big)$$

证毕。　■

之前的论断表明神经网络能够实现任意的布尔函数。但是这是一个很弱的性质，因为网络的体积将指数爆炸式增长。在论断 20.1 的证明的构造中，隐藏层的节点数呈指数爆炸。这不是我们证明制造的，下面的定理给出说明。

定理 20.2　*对于任意的 n，令 $s(n)$为最小整数，满足存在一个图(V,E)，有$|V|=s(n)$，从而假设类$\mathcal{H}_{V,E,\mathrm{sign}}$包含了所有$\{0,1\}^n$到$\{0,1\}$的函数，那么 $s(n)$是关于 n 的指数。σ 是 sigmoid 函数时$\mathcal{H}_{V,E,\sigma}$也存在类似的结论。*

证明　假定对于(V,E)我们有$\mathcal{H}_{V,E,\mathrm{sign}}$包含了所有$\{0,1\}^n$到$\{0,1\}$的函数。可以推论它能打散$\{0,1\}^n$集合中的 $m=2^n$ 个向量，因此$\mathcal{H}_{V,E,\mathrm{sign}}$的 VC 维是 2^n。另一方面，$\mathcal{H}_{V,E,\mathrm{sign}}$的 VC 维以$O(|E|\log(|E|))\leq O(|V|^2)$为界，这将在下一节说明。这意味着$|V|\geq\Omega(2^{n/2})$，这证明了以符合函数为激活函数的网络情形，sigmoid 函数情形也是类似的。　■

231

评注　对$\mathcal{H}_{V,E,\sigma}$中任何σ推导类似的理论是可行的，只要我们约束权重以使每个以公共常数为界的多比特权重都能表达。我们也可以考虑不同神经元使用不同激活函数的假设类，只要激活函数的数目是有限的。

多项式体积的网络能够表示什么样的函数？之前的论断告诉我们它不能表示所有的布尔函数。积极的一面是，接下来我们将看到所有能在$O(T(n))$时间内计算的布尔函数都能由一个体积为$O(T(n)^2)$的网络表示。

定理 20.3 令 $T:\mathbb{N}\to\mathbb{N}$ 且对于每个 n，令 $\mathcal{F}_n$ 为一个函数集，它能使用图灵机在 $T(n)$ 时间内执行。那么，存在常数 $b, c\in\mathbb{R}_+$，对每个 n，有体积最大为 $cT(n)^2+b$ 的图 (V_n, E_n) 使 $\mathcal{H}_{V_n,E_n,\mathrm{sgn}}$ 包含 $\mathcal{F}_n$。

这个定理的证明依赖程序时间复杂度与电路复杂度(见 Sipser 2006)的关系。简单讲，布尔电路是由单个神经元组成的网络，实现输入的合取、析取和否定。电路复杂度衡量完成所需函数计算的布尔电路体积。时间复杂度和电路复杂度的关系可以从下面的内容直观看到。我们认为计算机程序的每一步执行都是对内存的一次简单操作。因此，网络中各层的神经元将反应计算机相应时间的内存状态。而向网络下一层的传导则包含了网络执行的一次简单计算。为了将布尔电路与符号激活函数的网络相联系，我们需要证明使用符号激活函数能够完成合取、析取和否定运算。显然，我们能够使用符号函数执行否定运算，下面的引理将表明符号激活函数也能完成合取和析取。

引理 20.4 假设神经元 v 使用符号激活函数，具有 k 个输入边，与其连接的神经元输出为 $\{\pm 1\}$。那么，通过加入一个新的边，将一个常数神经元连接至 v 并调节其权重，则 v 能实现对其输入的合取或者析取运算。

证明 仅注意如果 $f:\{\pm 1\}^k\to\{\pm 1\}$ 是合取函数 $f(x)=\bigwedge_i x_i$，那么它可以写成 $f(x)=\mathrm{sign}\left(1-k+\sum_{i=1}^{k}x_i\right)$。类似，析取函数 $f(x)=\bigvee_i x_i$ 可以写成 $f(x)=\mathrm{sign}\left(k-1+\sum_{i=1}^{k}x_i\right)$。■ 232

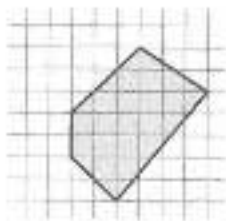
目前我们讨论了布尔函数。在练习 20.1 中，我们将看到神经网络是通用拟合器，即对任意给定的精度 $\varepsilon>0$ 和利普希茨函数 $f:[-1,1]^n\to[-1,1]$，可以构建一个网络，满足对于任意的输入向量 $x\in[-1,1]^n$，网络的输出在 $f(x)-\varepsilon$ 和 $f(x)+\varepsilon$ 之间。但是，就像布尔函数的情形一样，网络的体积也不能保证是 n 阶多项式的。下面定理给出了形式化证明，它的证明是定理 20.2 的推论，并留作练习。

定理 20.5 给定 $\varepsilon\in(0,1)$，对于每个 n，令 $s(n)$ 为最小整数，满足存在一个图 (V,E)，有 $|V|=s(n)$，从而 g 为 sigmoid 函数的假设类 $\mathcal{H}_{V,E,g}$ 能精度 ε 近似每个 1-利普希茨函数 $f:[-1,1]^n\to[-1,1]$，那么 $s(n)$ 是关于 n 的指数形式。

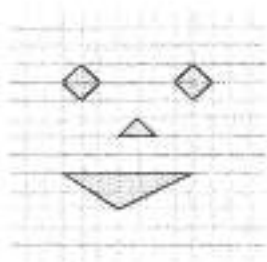
几何直观

下面我们给出一些函数 $f:\mathbb{R}^2\to\{\pm 1\}$ 的几何直观表示，并说明如何使用符号激活函数的神经网络对其表示。

我们以深度为 2 的网络开始，即网络只有一个隐藏层。隐藏层中的每个神经元都是半空间预测器。那么，输出层的单一神经元对隐藏层所有神经元的二值输出应用半空间预测。正如我们之前看到的，半空间能够实现合取函数。因此，这样的网络包含了所有 $k-1$ 个半空间公共部分的所有假设，其中 k 是隐藏层中神经元的个数。它们可以表示所有 $k-1$ 面的凸多面体，下面给出一个 5 个半空间相交的例子。



我们已经证明 V_2 层中的神经元能够实现指示 x 是否为凸多面体的函数。通过加入一层或更多层，并使输出层的神经元实现其输入的析取，我们将得到计算联合多面体的网络。下面是一个该函数的例子：



[233]

20.4　神经网络样本复杂度

下面我们讨论学习 $\mathcal{H}_{V,E,\sigma}$ 类的样本复杂度。回顾基本学习理论，我们知道学习二分类假设类的样本复杂度依赖于 VC 维。因此，我们关注形如 $\mathcal{H}_{V,E,\sigma}$ 的假设类的 VC 维计算，其中图的输出层只包含一个神经元。

我们从符号激活函数出发，即 $\mathcal{H}_{V,E,\mathrm{sign}}$，这一类的 VC 是多少呢？直观上，由于我们学习参数 $|E|$，VC 维应该由 $|E|$ 决定。确实如此，下面的定理说明的这一点。

定理 20.6　$\mathcal{H}_{V,E,\mathrm{sign}}$ 的 VC 维是 $O(|E|\log(|E|))$。

证明　为简化证明过程中的表示，我们记假设类为 $\mathcal{H}$。回顾 6.5.1 节中生长函数 $\tau_{\mathcal{H}}(m)$ 的定义。该函数度量 $\max_{C\subseteq\mathcal{X}:|C|=m}|\mathcal{H}_C|$，其中 $\mathcal{H}_C$ 是 $\mathcal{H}$ 对函数从 C 到 $\{0,1\}$ 的限制。我们自然地在函数集合上拓展这一定义为从 $\mathcal{X}$ 到有限集 $\mathcal{Y}$，使 $\mathcal{H}_C$ 为函数 $\mathcal{H}$ 对 C 到 $\mathcal{Y}$ 上的限制，并保留 $\tau_{\mathcal{H}}(m)$ 的定义不变。

神经网络由层状图定义。令 $V_0,\dots,V_T$ 为图中各层，固定 $t\in[T]$，通过设定 V_{t-1} 与 V_t 层间边上不同的权重，我们可获得 $\mathbb{R}^{|V_{t-1}|}\to\{\pm1\}^{|V_t|}$ 上的不同函数。令 $\mathcal{H}^{(t)}$ 为所有从 $\mathbb{R}^{|V_{t-1}|}\to\{\pm1\}^{|V_t|}$ 上可能的映射，那么 $\mathcal{H}$ 可以写成复合形式 $\mathcal{H}=\mathcal{H}^{(T)}\circ\dots\circ\mathcal{H}^{(1)}$。在练习 20.4 中我们给出假设类的复合生长函数是以各个类的生长函数的积为界的。因此

$$\tau_{\mathcal{H}}(m)\le\prod_{t=1}^{T}\tau_{\mathcal{H}^{(t)}}(m)$$

此外，每个 $\mathcal{H}^{(t)}$ 都能写成函数类的积 $\mathcal{H}^{(t)}=\mathcal{H}^{(t,1)}\times\dots\times\mathcal{H}^{(t,|V_t|)}$，其中 $\mathcal{H}^{(t,j)}$ 为 t 层的第 j 个神经元能够执行的从 $t-1$ 层到 $\{\pm1\}$ 的所有函数。练习 20.3 中，我们限制了乘积类，这使得

$$\tau_{\mathcal{H}^{(t)}}(m)\le\prod_{j=1}^{|V_t|}\tau_{\mathcal{H}^{(t,j)}}(m)$$

令 $d_{t,j}$ 为 t 层中指向第 j 个神经元的边的数量。由于神经元是齐次半空间假设且齐次半空间的 VC 维是其输入维度，根据 Sauer 定理有

$$\tau_{\mathcal{H}^{(t,j)}}(m)\le\left(\frac{em}{d_{t,j}}\right)^{d_{t,j}}\le(em)^{d_{t,j}}$$

最终，我们有

[234]

$$\tau_{\mathcal{H}}(m)\le(em)^{\sum_{t,j}d_{t,j}}=(em)^{|E|}$$

现在，我们假设有 m 个被打散的点，那么，我们需有 $\tau_{\mathcal{H}}(m)=2^m$，从中得到

$$2^m \leqslant (cm)^{|E|} \Rightarrow m \leqslant |E|\log(cm)/\log(2)$$

这可由引理 A.2 推得。 ■

下面我们考虑$\mathcal{H}_{V,E,\sigma}$，其中 σ 为 sigmoid 函数。惊奇的是$\mathcal{H}_{V,E,\sigma}$的 VC 维要低于边界 $\Omega(|E|^2)$(见练习 20.5)，这意味着，VC 维是可调参数的平方。将 VC 维的上界视为 $O(|V|^4|E|^2)$也是可以的，但证明不在本书范围内。在任何情况下，由于在实际中我们只考虑这样的神经网络，即权重由复杂度为 $O(1)$比特的浮点数简短表示，通过使用分离化技巧，我们能够轻松地知道该种网络的 VC 维是 $O(|E|)$，即使我们使用 sigmoid 函数。

20.5 学习神经网络的运行时

在之前的章节，我们已经揭示具有多项式容量的神经网络类能够表达所有能够高效执行的函数，同时，样本复杂度依赖于网络的体积。本节中，我们研究训练神经网络的时间复杂度。

我们首先说明在$\mathcal{H}_{V,E,\mathrm{sign}}$上执行 ERM 法则是 NP 难问题，即使该网络只有包含 4 个神经元的单一隐藏层。

定理 20.7 令 $k\geqslant 3$，对于任意的 n，令(V, E)为具有 n 个输入的层状网络。(唯一)的隐藏层具有 $k+1$ 个节点，其中一个是常数神经元，且网络只有一个输出节点。那么，对$\mathcal{H}_{V,E,\mathrm{sign}}$执行 ERM 法则是 NP 难问题。

证明依赖于 k 着色问题的推导，留作练习 20.6。

之前给出的困难性结果也许是由于学习的目的带来的，它可能找到一个具有低经验误差的预测 $h\in\mathcal{H}$，而不是准确的 ERM。但是，结果证明即使找到权重使其接近经验误差也是在计算上不可行的(见 Bartlett&Ben David 2002)。

那是否可以通过改变网络的结构以回避结果的困难？也就是说，也许 ERM 对于原始网络结构是困难的，但对于其他的大的网络就能高效实现(见第 8 章的例子)。另一个思路是改变激活函数(比如 sigmoid，或者其他能够高效计算的激活函数)。这些方法注定会失败。确实，在一些假设中，学习半空间的交叉区域是困难的，即使在表示独立模型的学习中(见 Klivans &Sherstov 2006)。这意味着在同样的假设下，半空间交叉区域的学习是不高效的。 235

一个广泛使用的启发式神经网络学习方法是我们在 14 章中讨论的 SGD。我们证明了 SGD 是凸损失函数下最优的学习法。在神经网络中，损失函数是高度非凸的。然而，我们仍然可以使用 SGD 算法，并希望其得到可行的解(就像一些实际应用中表现出的那样)。

20.6 SGD 和反向传播

在$\mathcal{H}_{V,E,\sigma}$中以低的风险找到假设的问题等价于找到边的权重的问题。本节我们给出如何用 SGD 算法启发式搜索最优解。我们假设 σ 是 sigmoid 函数 $\sigma(a)=1/(1+e^{-a})$。但推导适用于任何可微标量函数。

由于 E 是有限集，我们可以认为权重函数是一个向量 $w\in\mathbb{R}^{|E|}$。假设网络有 n 个输入神经元和 k 个输出神经元，记为 $h_w:\mathbb{R}^n\to\mathbb{R}^k$，在给定 w 表示的权值函数后它有网络计算。当目标是 $y\in\mathcal{Y}$时，$h_w(x)$的损失函数记为 $\Delta(h_w(x), y)$。具体地，我们将认为 Δ 为平方损失 $\Delta(h_w(x), y)=\frac{1}{2}\|h_w(x)-y\|^2$；然而，类似的推导对所有可微函数成立。最终，给定样本域 $\mathcal{X}\times\mathcal{Y}$ 上的分布D，令 $L_D(w)$为网络的风险，即

$$L_{\mathcal{D}}(w)=\mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}}[\Delta(h_w(x),y)]$$

回顾最小化风险函数 $L_{\mathcal{D}}(w)$ 的 SGD 算法。我们在 14 章的伪代码上加一些改进，使之与神经网络的非线性目标函数相关。首先，在 14 章中初始化 w 为零向量，这里我们初始化 w 为接近零的随机向量。因为全部为零的初始化会使隐藏层的权重相同(如果网络是包含完整层的)。此外，我们希望如果反复执行 SGD 算法，并且每次都用新的随机向量初始化，将有一个达到局部最优。第二，固定的步长 η 足够保证凸函数问题的效果，这里我们使用变步长 η_t，如 14.4.2 小节中定义的。由于损失函数的凸特性，η_t 序列的选取是至关重要的，这需要反复地尝试。第三，我们在验证集上获得最好的输出。此外，有时对权重加入正则系数 λ 是有效的。即，我们尝试最小化 $L_{\mathcal{D}}(w)+\frac{\lambda}{2}\|w\|^2$。最后，梯度不具有解析解，取而代之，使用反向传播算法，如下文所述：

236

神经网络 SGD 算法

参数：

迭代次数 τ

步长序列 $\eta_1, \eta_2, \ldots, \eta_\tau$

正则参数 $\lambda>0$

输入：

分层图 (V, E)

可微激活函数：$\sigma:\mathbb{R}\to\mathbb{R}$

初始化：

随机选取 $w^{(1)}\in\mathbb{R}^{|V|}$

(分布使 $w^{(1)}$ 趋于 0)

循环： $t=1, 2, \ldots, \tau$

样本 $(x, y)\sim\mathcal{D}$

计算梯度 $v_t=\text{backpropagation}(x, y, w(V, E), \sigma)$

更新 $w^{(t+1)}=w(t)-\eta_t(v_t+\lambda w^{(t)})$

输出：

w 为验证集上最优的 $w^{(t)}$

反向传播算法

输入：

样本 (x, y)，权重向量 w，分层图 (V, E)

激活函数：$\sigma:\mathbb{R}\to\mathbb{R}$

初始化：

记图中各层 $V_0, \ldots, V_T$，其中 $V_i=\{v_{i,1}, \ldots, v_{i,k_i}\}$

定义 $W_{i,j,l}$ 为 $(v_{i,j}, v_{i+1,l})$ 的权重

(如果 $(v_{i,j}v_{i+1,l})\notin E$，设 $W_{i,j,l}=0$)

前向：

设 $o_0=x$

循环　$t=1, \cdots, T$

　循环　$i=1, \cdots, k_t$

　　设 $a_{t,i}=\sum_{j=1}^{k_{t-1}} W_{t-1,j,i} o_{t-1,j}$

　　设 $o_{t,i}=\sigma(a_{t,i})$

后向：

　设 $\boldsymbol{\delta}_T=\boldsymbol{\sigma}_T-y$

　循环　$t=T-1, T-2, \cdots, 1$

　　循环　$i=1, \cdots, k_t$

　　$\delta_{t,i}=\sum_{j=1}^{k_{t+1}} W_{t,i,j} \delta_{t+1,j} \sigma'(a_{t+1,j})$

输出：

　对于每个边 $(v_{t-1,j}, v_{t,i}) \subset E$

　　设定偏微分 $\delta_{t,i} \sigma'(a_{t,i}) o_{t-1,j}$

反向传播如何计算梯度

下面我们解释反向传播算法如何计算给定向量 w，在样本 (x, y) 上的损失函数梯度。我们首先回顾向量微积分的定义。梯度中每一个元素都是网络各边上 V，E，σ 参数的偏微分。回顾偏微分的定义，给定函数 $f: \mathbb{R}^n \to \mathbb{R}$，其关于 w 中第 i 个变量的偏微分通过 $w_1, \cdots, w_{i-1}, w_{i+1}, \cdots, w_n$ 给出，这产生了标量函数 $g: \mathbb{R} \to \mathbb{R}$，由 $g(\alpha)=f((w_1, \cdots, w_{i-1}, w_i+\alpha, w_{i+1}, \cdots, w_n))$ 定义，然取 $g=0$ 的微分。对于多输出函数 $f: \mathbb{R}^n \to \mathbb{R}^m$，$f$ 在 $w \in \mathbb{R}^n$ 的雅克比矩阵记为 $J_w(f)$，它是 $m \times n$ 阶矩阵，第 i 行 j 列元素为 $f_i: \mathbb{R}^n \to \mathbb{R}$ 的偏微分。如果 $m=1$，那么雅克比矩阵是函数的梯度（表示为行向量）。下面是两个雅克比矩阵的性质，我们将在后面使用。

237

- 令 $f(w)=Aw$，对于 $A \in \mathbb{R}^{m \times n}$，那么 $J_w(f)=A$。
- 对于任意的 n，σ 表示 $\mathbb{R}^n$ 到 $\mathbb{R}^n$ 的函数，采用 sigmoid 函数。即 $\boldsymbol{\alpha}=\boldsymbol{\sigma}(\boldsymbol{\theta})$ 表示对任意的 i，我们有 $\alpha_i=\sigma(\theta_i)=\frac{1}{1+\exp(-\theta_i)}$。容易验证 $J_{\boldsymbol{\theta}}(\boldsymbol{\sigma})$ 是对角矩阵，其 (i, i) 元素是 $\sigma'(\theta_i)$，其中 σ' 为标量 sigmoid 函数的衍生函数，即 $\sigma'(\theta_i)=\frac{1}{(1+\exp(\theta_i))(1+\exp(-\theta_i))}$，我们也用 $\mathrm{diag}(\boldsymbol{\sigma}'(\boldsymbol{\theta}))$ 表示这个矩阵。

复合函数微分链式法则也能用于雅克比矩阵。给定两个函数 $f: \mathbb{R}^m \to \mathbb{R}^n$ 和 $g: \mathbb{R}^l \to \mathbb{R}^m$，我们有在 w 处的复合函数 $(f \circ g): \mathbb{R}^l \to \mathbb{R}^n$ 的雅克比矩阵：

$$J_w(f \circ g)=J_{g(w)}(f) J_w(g)$$

例如，$g(w)=Aw$，其中 $A \in \mathbb{R}^{m \times l}$，我们有

$$J_w(f \circ g)=\mathrm{diag}(\boldsymbol{\sigma}'(Aw)) A$$

为了描述反向传播算法，让我们将 V 分解到图中各层，$V=\cup_{t=1}^{T} V_t$。对于任意的 t，我们记 $V_t=\{v_{t,1}, \cdots, v_{t,k_t}\}$，其中 $k_t=|V_t|$。此外，对于任意的 t 满足 $W_t \subset \mathbb{R}^{(k_t+1) \times k_{t+1}}$ 是一个矩阵，其给出了 V_t 与 V_{t+1} 各个边的权重。如果 E 中存在这条边，那么根据 w 我们将 $W_{t,i,j}$ 设为边 $(v_{t,i}, v_{t+1,j})$ 的权重。否则，我们加入"虚拟的"边并将权重设为 0，即

$W_{i,j,l}=0$。由于计算各个边权重的偏微分时，我们会固定其他的权重，这些“虚拟”加入的边对计算已有权重的偏微分不会有影响。不失一般性，我们可以认为所有的边均存在，即 $E=\bigcup_i(V_i\times V_{i+1})$。

下面，我们讨论如何计算 V_{i-1} 到 V_i 的各个边的，关于 W_{i-1} 中元素的偏微分。由于固定了网络中其他的权重，可以推出所有 V_{i-1} 中的神经元输出固定不受权重 W_{i-1} 影响，可以用 $\boldsymbol{o}_{i-1}$ 表示。此外，我们用 $\ell_i:\mathbb{R}^{k_i}\to\mathbb{R}$ 表示各个子网的损失函数，子网由 $V_i,\cdots,V_r$ 层定义，以 V_i 神经元的输出为函数。V_i 中神经元的输入可以写成 $\boldsymbol{a}_i=W_{i-1}\boldsymbol{o}_{i-1}$，输出可以写成 $\boldsymbol{o}_i=\boldsymbol{\sigma}(\boldsymbol{a}_i)$，即，对任意 j，$o_{i,j}=\sigma(a_{i,j})$。我们得到的损失是 W_{i-1} 的函数，能够写成

238

$$g_i(W_{i-1})=\ell_i(\boldsymbol{o}_i)=\ell_i(\boldsymbol{\sigma}(\boldsymbol{a}_i))=\ell_i(\boldsymbol{\sigma}(W_{i-1}\boldsymbol{o}_{i-1}))$$

按照以下的方法重写上述内容将是方便的。令 $w_{i-1}\in\mathbb{R}^{k_{i-1}k_i}$ 为 W_{i-1} 中各行连接并转置而成的列向量。定义 O_{i-1} 为如下 $k_i\times(k_{i-1}k_i)$ 矩阵：

$$O_{i-1}=\begin{pmatrix}\boldsymbol{o}_{i-1}^{\mathrm{T}} & 0 & \cdots & 0\\ 0 & \boldsymbol{o}_{i-1}^{\mathrm{T}} & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \cdots & \boldsymbol{o}_{i-1}^{\mathrm{T}}\end{pmatrix} \tag{20.2}$$

那么，$W_{i-1}\boldsymbol{o}_{i-1}=O_{i-1}w_{i-1}$，所以我们有

$$g_i(w_{i-1})=\ell_i(\boldsymbol{\sigma}(O_{i-1}w_{i-1}))$$

因此，使用链式法则，我们得到

$$J_{w_{i-1}}(g_i)=J_{\boldsymbol{\sigma}(O_{i-1}w_{i-1})}(\ell_i)\mathrm{diag}(\boldsymbol{\sigma}'(O_{i-1}w_{i-1}))O_{i-1}$$

使用我们的符号，有 $\boldsymbol{o}_i=\boldsymbol{\sigma}(O_{i-1}w_{i-1})$ 且 $\boldsymbol{a}_i=O_{i-1}w_{i-1}$，这推出

$$J_{w_{i-1}}(g_i)=J_{\boldsymbol{o}_i}(\ell_i)\mathrm{diag}(\boldsymbol{\sigma}'(\boldsymbol{a}_i))O_{i-1}$$

令 $\boldsymbol{\delta}_i=J_{\boldsymbol{o}_i}(\ell_i)$，那么，我们将上式进一步写成

$$J_{w_{i-1}}(g_i)=(\delta_{i,1}\sigma'(a_{i,1})\boldsymbol{o}_{i-1}^{\mathrm{T}},\cdots,\delta_{i,k_i}\sigma'(a_{i,k_i})\boldsymbol{o}_{i-1}^{\mathrm{T}}) \tag{20.3}$$

它被用于计算向量 $\boldsymbol{\delta}_i=J_{\boldsymbol{o}_i}(\ell_i)$，这是 $\boldsymbol{o}_i$ 处 ℓ_i 的梯度。我们用迭代方法求解它。首先注意到，对于最后一层，我们有 $\ell_r(u)=\Delta(u,y)$，其中 Δ 为损失函数。由于我们假定 $\Delta(u,y)=\frac{1}{2}|u-y|^2$，得到 $J_u(\ell_r)=(u-y)$。特别地，$\boldsymbol{\delta}_r=J_{\boldsymbol{o}_r}(\ell_r)=(\boldsymbol{o}_r-y)$。下面，因为

$$\ell_i(u)=\ell_{i+1}(\boldsymbol{\sigma}(W_iu))$$

因此，根据链式法则有

$$J_u(\ell_i)=J_{\boldsymbol{\sigma}(W_iu)}(\ell_{i+1})\mathrm{diag}(\boldsymbol{\sigma}'(W_iu))W_i$$

特别地，

$$\begin{aligned}\boldsymbol{\delta}_i=J_{\boldsymbol{o}_i}(\ell_i)&=J_{\boldsymbol{\sigma}(W_i\boldsymbol{o}_i)}(\ell_{i+1})\mathrm{diag}(\boldsymbol{\sigma}'(W_i\boldsymbol{o}_i))W_i\\&=J_{\boldsymbol{a}_{i+1}}(\ell_{i+1})\mathrm{diag}(\boldsymbol{\sigma}'(\boldsymbol{a}_{i+1}))W_i\\&=\boldsymbol{\delta}_{i+1}\mathrm{diag}(\boldsymbol{\sigma}'(\boldsymbol{a}_{i+1}))W_i\end{aligned}$$

总之，我们可以首先从网络的底部向上计算向量 $\{\boldsymbol{a}_i,\boldsymbol{o}_i\}$。而后，从网络的顶部向下计算向量 $\{\boldsymbol{\delta}_i\}$。一旦计算完成所有向量，就可以通过式(20.3)轻松计算偏微分。我们已经

239

展现了反向传播计算梯度的伪代码。

20.7 小结

医体积为 $s(n)$ 的神经网络能够描述所有运行时间为 $O(\sqrt{s(n)})$ 的假设类预测。我们也

证明了它的样本复杂度为 $s(n)$ 的多项式(特别地，它依赖于网络边的个数)。因此，神经网络假设类似乎是个不错的选择。遗憾的是，基于样本集训练神经网络是难于计算的。我们给出了 SGD 框架作为神经网络训练的启发式方法，并给出了反向传播算法，它能高效计算各个边上权重损失函数的梯度。

20.8　文献评注

神经网络在 19 世纪 80 年代和 90 年代初期得到广泛研究，但只有经验上的成功。最近，算法上的进步以及数据体积和计算能力上的提升，使神经网络的有效性获得突破。特别是深度网络(即 2 层以上网络)已经在许多领域表现出优异的性能。例子包括卷积网络(LeCun & Bengio 1995)，受限玻尔兹曼机(Hinton，Osindero & Teh 2006)，自动编码(Ranzato 等 2007，Bengio & LeCun 2007，Collobert & Weston 2008，Lee 等 2009，Le 等 2012)和卷积网络(Livni，Shalev-Shwartz & Shamir 2013，Poon & Domingos 2011)。同时也有(Bengio 2009)和其他的资料。

神经网络的表达力和其与电路复杂度的关系也得到了研究(Parberry 1994)。为研究神经网络的复杂度，我们建议参考 Anthony & Bartlet (1999)。我们证明定理 20.6 的技巧参考 Kakade 和 Tewari 的演讲笔记。

Klivans 和 Sherstov(2006)已经证明对于任意的 $c>0$，n^c 半空间在 $\{\pm1\}^n$ 上的相交部分不能通过 PCA 有效学习，即使我们允许表示独立学习。这一困难是出于以下假设，即唯一最短向量问题不存在多项式时间的解。我们认为，它暗示着神经网络的训练不能找到高效算法，即使我们允许更大的网络或其他的能够高效执行的激活函数。

反向传播算法由 Rumelhart，Hinton 和 Williams(1986)提出。

20.9　练习

20.1　**神经网络是通用拟合器**：假定 $f:[-1,1]^n\to[-1,1]$ 是 ρ-利普希茨函数。给定 $\varepsilon\in(0,1)$。构造一个神经网络 $N:[-1,1]^n\to[-1,1]$，以 sigmoid 函数为激励函数，满足对于每个 $x\in[-1,1]^n$ 有 $|f(x)-N(x)|\le\varepsilon$。

提示：类似于定理 19.3 的证明，将 $[-1,1]^n$ 划分为若干个小矩形。利用利普希茨性质证明函数 f 在每个小矩形上都近似于常数。最后，证明神经网络根据输入向量所在的小矩形，预测网络的输出值为 f 在该小矩形上的平均值。

290

20.2　证明定理 20.5。

提示：对于任意函数 $f:[-1,1]^n\to[-1,1]$，构造一个 1-利普希茨函数 $g:[-1,1]^n\to[-1,1]$，证明可以逼近 g，从而可以逼近 f。

20.3　**乘积的生长函数**：对于 $i=1,2$，令 $\mathcal{F}_i$ 为从 $\mathcal{X}$ 到 $\mathcal{Y}_i$ 的函数集。定义 $\mathcal{H}=\mathcal{F}_1\times\mathcal{F}_2$ 为 Cartesian 乘积类。即对任意的 $f_1\in\mathcal{F}_1$ 和 $f_2\in\mathcal{F}_2$，存在 $h\in\mathcal{H}$ 使得 $h(x)=(f_1(x),f_2(x))$。求证 $\tau_{\mathcal{H}}(m)\le\tau_{\mathcal{F}_1}(m)\tau_{\mathcal{F}_2}(m)$。

20.4　**复合的生长函数**：令 $\mathcal{F}_1$ 为从 $\mathcal{X}$ 到 $\mathcal{Z}$ 的函数集，$\mathcal{F}_2$ 为从 $\mathcal{Z}$ 到 $\mathcal{Y}$ 的函数集。定义 $\mathcal{H}=\mathcal{F}_2\circ\mathcal{F}_1$ 为复合类。即对任意的 $f_1\in\mathcal{F}_1$ 和 $f_2\in\mathcal{F}_2$，存在 $h\in\mathcal{H}$ 使得 $h(x)=f_2(f_1(x))$。求证 $\tau_{\mathcal{H}}(m)\le\tau_{\mathcal{F}_2}(m)\tau_{\mathcal{F}_1}(m)$。

20.5　**Sigmoid 网络的 VC 维**：本题中，我们将证明存在一个图 (V,E)，使得该图上以 Sigmoid 函数为激励函数的神经网络的 VC 维是 $\Omega(|E|^2)$。注意到对于任意的

$\varepsilon>0$，Sigmoid 函数都能以 ε 的精度逼近阈值激励函数 $\mathbf{1}_{\{\sum_i x_i>0\}}$。为简化证明，在本题中，我们假设 Sigmoid 函数能够精确实现阈值激励函数 $\mathbf{1}_{\{\sum_i x_i>0\}}$。

固定某个 n。

1) 构造网络 N_1，其权重数为 $O(n)$，将 $\mathbb{R}$ 映射到 $\{0,1\}^n$ 且满足如下性质：对于任意 $\boldsymbol{x}\in\{0,1\}^n$，如果网络的输入是实数 $0.x_1x_2\cdots x_n$，则网络的输出为 $\boldsymbol{x}$。
   提示：记 $a=0.x_1x_2\cdots x_n$，注意到当 $x_2=1$ 时，$10^6a-0.5$ 不小于 0.5，当 $x_2=-1$ 时，$10^6a-0.5$ 不大于 -0.5。

2) 构造网络 N_2，其权重数为 $O(n)$，将 $[n]$ 映射到 $\{0,1\}^n$ 且对所有的 i 有 $N_2(i)=e_i$。也就是说，对于输入值 i，网络的输出在第 i 个节点值为 1，在其余节点值均为 0。

3) 令 $a_1,\cdots,a_n$ 为 n 个实数，且每个 a_i 均可写为 $0.a_i^{(1)}a_i^{(2)}\cdots a_i^{(n)}$，其中 $a_i^{(j)}\in\{0,1\}$。构造网络 N_3，其权重数为 $O(n)$，将 $[n]$ 映射到 $\mathbb{R}$，且对任意 $i\in[n]$ 有 $N_3(i)=a_i$。

4) 联合 N_1 和 N_2，得到一个新的网络，其输入为 $i\in[n]$，输出为 $\boldsymbol{a}^{(i)}$。

5) 构造网络 N_4 使其输入为 $(i,j)\in[n]\times[n]$，输出为 $a_i^{(j)}$。
   提示：$\{0,1\}^3$ 上的与函数(AND function)可由 $O(1)$ 权重表示。

6) 证明结论：存在一个 $O(n)$ 权重的图，使其对应的假设类的 VC 维是 n^2。

20.6　证明定理 20.7。

241　提示：可参考平空间的相交部分的学习难题，见第 8 章练习 32。

第三部分

Understanding Machine Learning: From Theory to Algorithms

其他学习模型

在线学习

本章我们讨论另一种学习模型——在线学习。在前面的章节中，我们已经讲过了PAC学习模型：首先学习器接受一批训练样本，从训练集中学习假设，并且只有在完成该过程之后才运用学习到的假设来判别新样本的类别。在前述的木瓜学习问题中，这意味着我们需要首先购买一堆木瓜并全部试吃。然后，我们运用所有信息来学习决定新木瓜口味的预测规则。相比之下，在线学习的过程在训练和预测环节中间并无分隔。相反，每次我们购买一个木瓜，因为需要先预测其口味是否良好，它首先被考虑作为一个测试样例。咬一口木瓜之后，我们获知其真实标签，随后这个木瓜作为训练样本用于提高我们对未来的木瓜的预测机制。

具体来说，在线学习发生在一序列连续的回合里。在每一个在线的回合，学习器首先接受一个新实例（比如学习器购买一个木瓜并且获知构成该实例的形状和颜色）。然后学习器需要预测其标签（比如该木瓜是不是美味？）。在回合的结尾，学习器得到其正确的标签（已经尝试该木瓜并且获知是不是美味）。最后，学习器运用这些信息来提高未来的预测。

为了分析在线学习，我们遵循与PAC学习相似的研究路线。我们从二值化分类问题的在线学习开始。既考虑可实现的情况，在这种情况下，先验知识是假定所有标签从给定假设集合上的某些假设生成；同时也考虑不可实现的情况，这对应于不可知PAC学习器。特别地，我们介绍一个重要算法，称为加权投票。其次，我们研究损失函数是凸函数的在线学习问题。最后，我们介绍感知器算法作为在线学习中运用替代凸损失函数的一个例子。

21.1 可实现情况下的在线分类

在线学习在一序列连续的回合里执行：在回合 t，学习器接收从实例域$\mathcal{X}$获取的实例 x_t，并且需要提供其标签，记预测标签为 p_t。在预测标签之后，学习器获知其真实的标签 $y_t \in \{0, 1\}$。学习器的目标是在这个过程中尽可能少地预测错误。学习器尝试从先前的回合中推断有效信息，从而在未来的回合中提高预测性能。

显然，如果在过去和现在的回合之中没有任何相关，学习是没有希望的。在本书前面部分，我们研究PAC模型的时候假定了过去和现在的样本独立同分布地采样于相同的源分布。在线学习模型中，我们对样本序列的来源不做统计上的假设。样本序列可以是确定性的，也可以是随机的，或者甚至是敌对式地根据学习器自身行为进行调整的（比如垃圾邮件的过滤）。自然而然地，一个对手可以让在线学习算法的预测错误任意地大。例如，这个对手可以在每个回合展示相同的实例，等待学习器的预测，然后提供一个相反的标签作为正确标签。

为了做出不平凡的陈述，我们需要进一步限定这个问题。可实现性假设是一个可能且自然的限制。在可实现的情况下，我们假定所有的标签从假设 $h^*:\mathcal{X}\to y$ 生成。更进一步，h^* 是从学习器已知的假设集$\mathcal{H}$中获取的。这和我们在第3章中研究PAC学习模型是相似

的。有了这个限制，假设h^*和实例序列可以由对手来选定，学习器也应该尽可能少地犯错误。对于在线学习算法A，记A在标记$h^*\in\mathcal{H}$的样例序列上可能犯的错误的个数最多为$M_A(\mathcal{H})$。我们再次强调，h^*和实例序列可以由对手来选定。$M_A(\mathcal{H})$的界称为误差界，我们将会研究如何设计算法使得$M_A(\mathcal{H})$最小。正式地，

定义21.1（误差界，在线可学习性） 令$\mathcal{H}$表示假设集，A表示在线学习算法，考虑序列$S=(x_1,h^*(y_1)),\dots,(x_T,h^*(y_T))$，其中$T$是任意整数，$h^*\in\mathcal{H}$，令$M_A(S)$表示在序列$S$上$A$预测的错误个数。记$M_A(\mathcal{H})$为上述格式的序列上的$M_A(S)$的上确界。形如$M_A(\mathcal{H})\leqslant B<\infty$的界称为误差界。称假设集$\mathcal{H}$是可在线学习的，如果存在一个算法使得$M_A(\mathcal{H})\leqslant B<\infty$。

我们的目标是研究哪些假设集在在线学习的模型下是可学习的，尤其是为给定假设集找到一个好的学习算法。

评注 在本小节和下一小节中，我们忽略学习问题的计算复杂性，并且不限定算法必须高效。在21.3和21.4节，我们研究高效的在线学习算法。

为了简化表示，我们从有限的假设集开始，换言之，$|\mathcal{H}|<\infty$。 246

在PAC学习中，如果$\mathcal{H}$是可学习的则可以由$\mathrm{ERM}_{\mathcal{H}}$规则学习，在这种意义下我们界定ERM是一种好的学习算法。在线学习的一种自然学习规则是利用（在任何一个在线的回合中的）任一ERM规则，也就是说，采用任何与过去的样例相一致的假设。

一致性算法

输入：有限假设类$\mathcal{H}$
初始化：$V_1=\mathcal{H}$
对于$t=1,2,\dots$
　接收x_t
　选择$h\in V_t$
　预测$p_t=h(x_t)$
　接收真实标签$y_t=h^*(x_t)$
　更新$V_{t+1}=\{h\in V_t:\ h(x_t)=y_t\}$

一致性（Consistent）算法维护一个集合V_t，其中包含所有和$(x_1,y_1),\dots,(x_{t-1},y_{t-1})$一致的假设。这个集合通常被称为可行域，学习器从$V_t$中选取任一假设，根据它进行预测。

显然，无论何时一致性算法犯了预测错误，至少一个假设将从V_t移除。因此，在做出M个错误之后$|V_t|\leqslant|\mathcal{H}|-M$。由于$V_t$非空（由可实现假设，其包含$h^*$），我们有$1\leqslant|V_t|\leqslant|\mathcal{H}|-M$。整理可得如下结论：

推论21.2 令$\mathcal{H}$表示有限假设集，一致性算法的误差界为$M_{\text{Consistent}}(\mathcal{H})\leqslant|\mathcal{H}|-1$。

很容易构造一个假设集和样例序列使得一致性算法实际上将做出$|\mathcal{H}|-1$个错误（参见练习21.1）。因此，我们介绍一个更好的算法，其中通过一种更聪明的方式选取$h\in V_t$。我们将看到这个算法可以保证指数式地犯更少的错误。

二分算法

输入：有限假设类$\mathcal{H}$
初始化：$V_1=\mathcal{H}$
对于 $t=1, 2, \ldots$
　接收 x_t
　预测 $p_t=\underset{r\in\{0,1\}}{\operatorname{argmax}}|\{h\in V_t: h(x_t)=r\}|$
　（约束预测 $p_t=1$）
　接收真实标签 $y_t=h^*(x_t)$
　更新 $V_{t+1}=\{h\in V_t: h(x_t)=y_t\}$

定理 21.3　令$\mathcal{H}$表示有限假设类。二分（Halving）算法的误差界为 $M_{\text{Halving}}(\mathcal{H})\leq \log_2(|\mathcal{H}|)$。

[247]

证明　我们只要注意到：无论算法误差为何，始终成立$|V_{t+1}|\leq|V_t|/2$（因此称为二分）。因此，如果总误差数为M，则有

$$1\leq|V_{T+1}|\leq|\mathcal{H}|2^{-M}$$

整理这个不等式可以得到所要证明的结论。 ■

二分算法的误差界当然比一致性算法的误差界好得多。我们已经看到在线学习和PAC学习的不同之处——在PAC下，任一ERM的假设都是好的，而在线学习下，选择任意的ERM假设远未达到最优。

在线可学习性

接下来我们采用更普遍的方法来刻画在线可学习性。特别地，我们关注下述问题：给定假设集$\mathcal{H}$，什么是最优的在线学习算法？

我们引入假设集的维数概念来刻画可达到的最优误差界。这种度量由Nick Littlestone提出，因而我们记之为$\operatorname{Ldim}(\mathcal{H})$。

为了引出对Ldim的定义，将在线学习的过程视为两个玩家之间的游戏，分别是学习器及其环境。在游戏的第t个回合，环境挑出实例x_t，学习器预测标签$p_t\in\{0, 1\}$。假定环境想让学习器在游戏的第T个回合出错，它必须输出$y_t=1-p_t$。唯一的问题是它如何选择实例x_t使得对于某些假设$h^*\in\mathcal{H}$而言，对任意$t\in[T]$，$y_t=h^*(x_t)$成立。

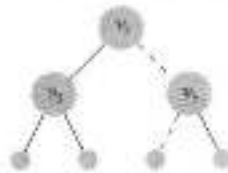
一个敌对式的环境的策略可以正式地描述为下述的二叉树：其中的每个节点和$\mathcal{X}$中的一个实例相关联。最初，环境向学习器展示和根节点相关联的实例。接下来，如果学习器预测$p_t=1$，环境将声明这是一个错误的预测（也就是说，$y_t=0$）并且向当前节点的右子树遍历。如果学习器预测$p_t=0$，则环境将设$y_t=1$并向左子树遍历。持续这个过程，在每个回合，环境将展示当前节点的关联实例。

正式地，考虑一棵深度为T的完全二叉树（定义树的深度为：从根节点到叶子节点的路径上的数目）。这棵树上有$2^{T+1}-1$个节点，每个节点都附加一个实例，记实例为$v_1, \ldots, v_{2^{T+1}-1}$。我们从树的根节点开始，令$x_1=v_1$。在第$t$个回合，令$x_t=v_{i_t}$，其中$i_t$是当前节点。在第$t$个回合的结束阶段，如果$y_t=0$我们转向$i_t$的左子树，如果$y_t=1$则转向其右子树。这意味着，$i_{t+1}=2i_t+y_t$。解该递推式可以得到$i_t=2^{t-1}+\sum_{j=1}^{t-1}y_j2^{t-j}$。

前述的环境策略成功当且仅当对任何$(y_1, \cdots, y_T)$，对所有的$t \in [T]$，存在$h \in \mathcal{H}$使得$y_t = h(x_t)$成立。这引出了下面的定义。

定义 21.4（$\mathcal{H}$打散树） 一棵深度为d的打散树，是$\mathcal{X}$上的一个实例序列$v_1, \cdots, v_{2^d-1}$，对应于每一种标签信息$(y_1, \cdots, y_d) \in \{0, 1\}^d$，存在$h \in \mathcal{H}$，使得对任意的 248
$t \in [d]$，有$h(v_{i_t}) = y_t$，其中$i_t = 2^{t-1} + \sum_{j=1}^{t-1} y_j 2^{t-1-j}$。

一个深度为2的打散树的示例在图21.1中给出。



	h_1	h_2	h_3	h_4
v_1	0	0	1	1
v_2	0	1	*	*
v_3	*	*	0	1

图 21.1

定义 21.5（Littlestone 维（Ldim）） Ldim($\mathcal{H}$)是满足下述条件的最大整数T：存在深度为T的被$\mathcal{H}$所打散的打散树。

由Ldim的定义和前述的讨论立得：

引理 21.6 不存在误差界严格小于Ldim($\mathcal{H}$)的算法。换言之，对任意算法A，我们有$M_A(\mathcal{H}) \geq \mathrm{Ldim}(\mathcal{H})$。

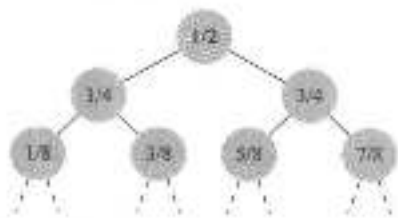
证明 令$T = \mathrm{Ldim}(\mathcal{H})$，并令$v_1, \cdots, v_{2^T-1}$为满足Ldim定义的序列。对所有$t \in [T]$，如果环境设置$x_t = v_{i_t}$与$y_t = 1 - p_t$，那么学习器会犯$T$个错误，而Ldim的定义蕴含着存在一个假设$h \in \mathcal{H}$，使得对所有的$t$有$y_t = h(x_t)$。 ■

现在给出几个例子。

例 21.1 令$\mathcal{H}$表示有限假设集。显然，任何被$\mathcal{H}$打散的树深度最大为$\log_2(|\mathcal{H}|)$。因此，$\mathrm{Ldim}(\mathcal{H}) \leq \log_2(|\mathcal{H}|)$。这个不等式也可以结合引理21.6和定理21.3推出。 ◀

例 21.2 令$\mathcal{X} = \{1, \cdots, d\}$，$\mathcal{H} = \{h_1, \cdots, h_d\}$，其中$h_j(x) = 1$当且仅当$x = j$。其次，容易证明$\mathrm{Ldim}(\mathcal{H}) = 1$，而$|\mathcal{H}| = d$可以任意大。因此，这个例子说明，Ldim($\mathcal{H}$)可以远小于$\log_2(|\mathcal{H}|)$。 ◀

例 21.3 令$\mathcal{X} = [0, 1]$，$\mathcal{H} = \{x \mapsto \mathbf{1}_{x \leq a} : a \in [0, 1]\}$；也就是说，$\mathcal{H}$是$[0, 1]$区间上的阈值的类。因此$\mathrm{Ldim}(\mathcal{H}) = \infty$。为了说明这一点，考虑树：



249

这棵树被$\mathcal{H}$打散，且由于实数的稠密性，这棵树可以构造出任意深度。 ◀

引理21.6说明了Ldim($\mathcal{H}$)是任何算法误差界的下界。有趣的是，有一个标准算法，其误差界正好匹配这个下界。这个算法和二分算法相似。回顾二分算法，其预测过程是根

据和以往样例相一致的假设的多数投票决定的。我们记这样的假设集合为 V_t。换句话说，二分算法将 V_t 分割为两个集合：$V_t^{+}=\{h\in V_t: h(x_t)=1\}$ 和 $V_t^{-}=\{h\in V_t: h(x_t)=0\}$，然后根据两组之中较大的进行预测。这样预测的理论依据是无论何时二分算法犯了一个错误，它都会结束于 $|V_{t+1}|\leq 0.5|V_t|$。

下述的最优算法采用了相同的思路，但并非依据较大的集合，而是依据具有较大的 Ldim 的集合进行预测。

标准最优算法(SOA)

输入：假设类 $\mathcal{H}$
初始化：$V_1=\mathcal{H}$
对于 $t=1, 2, \cdots$
　接收 x_t
　对于 $r\in\{0, 1\}$，令 $V_t^{(r)}=\{h\in V_t: h(x_t)=r\}$
　预测 $p_t=\underset{r\in\{0,1\}}{\operatorname{argmax}}\ \mathrm{Ldim}(V_t^{(r)})$
（约束预测 $p_t=1$）
　接收真实标签 y_t
　更新 $V_{t+1}=\{h\in V_t: h(x_t)=y_t\}$

下述引理正式表述了上述算法的最优性。

引理 21.7　SOA 算法的误差界为 $M_{\mathrm{SOA}}(\mathcal{H})\leq \mathrm{Ldim}(\mathcal{H})$。

证明　只需要证明：无论何时算法犯了一个预测错误，都成立 $\mathrm{Ldim}(V_{t+1})\leq \mathrm{Ldim}(V_t)-1$。我们通过反证法，假设 $\mathrm{Ldim}(V_{t+1})=\mathrm{Ldim}(V_t)$。如果假设成立，$p_t$ 的定义则意味着，对 $r=1$ 和 $r=0$，$\mathrm{Ldim}(V_t^{(r)})=\mathrm{Ldim}(V_t)$。但是，那么我们就能构造一棵打散树，对应于集合 V_t，其深度为 $\mathrm{Ldim}(V_t)+1$，这就产生了矛盾。 ■

结合引理 21.6 和引理 21.7 可以得到：

推论 21.8　令 $\mathcal{H}$ 表示假设集。标准最优算法的误差界为 $M_{\mathrm{SOA}}(\mathcal{H})=\mathrm{Ldim}(\mathcal{H})$，并且没有其他算法可以实现 $M_A(\mathcal{H})<\mathrm{Ldim}(\mathcal{H})$。

和 VC 维的比较

250　在 PAC 学习模型下，可学习性是通过 $\mathcal{H}$ 集合的 VC 维来刻画的。回顾一下，集合 $\mathcal{H}$ 的 VC 维是 $\mathcal{H}$ 所打散的实例 $x_1, \cdots, x_d$ 的最大基数 d。也就是说，对任意标签序列 $(y_1, \cdots, y_d)\in\{0, 1\}^d$，存在假设 $h\in\mathcal{H}$ 给出这个确切的标签序列。下述定理说明了 VC 维和 Littlestone 维之间的关系。

定理 21.9　对任意假设集 $\mathcal{H}$，$\mathrm{VCdim}(\mathcal{H})\leq \mathrm{Ldim}(\mathcal{H})$，并且存在集合使得不等号严格成立。更进一步，二者的差距可以任意大。

证明　我们首先证明 $\mathrm{VCdim}(\mathcal{H})\leq \mathrm{Ldim}(\mathcal{H})$。假定 $\mathrm{VCdim}(\mathcal{H})=d$，记 $x_1, \cdots, x_d$ 为打散的集合。现在，构造一棵实例为 $v_1, \cdots, v_{2^d-1}$ 的完全二叉树，其深度为 i 的所有节点都是 x_i，如下图所示：

打散集的定义明确地告诉我们已经得到了一棵有效的深度为 d 的打散树，因此得到 $\mathrm{VCdim}(\mathcal{H})\leq\mathrm{Ldim}(\mathcal{H})$。为了证明二者的差距可以任意大，只需要注意到例 21.4 中的假设集，其 VC 维是 1，但是其 Littlestonw 维是无穷大。 ■

21.2 不可实现情况下的在线识别

在上一小节我们研究了可实现情况下的在线可学习性。现在我们考虑不可实现的情况。同不可知的 PAC 模型相似，我们不再假设所有的标签是由某个 $h^*\in\mathcal{H}$ 生成的，而是要求学习器与 $\mathcal{H}$ 中固定的最优预测器进行竞争。这被称为算法的缺憾度，其度量了学习器没有跟随某些假设的预测 $h\in\mathcal{H}$ 导致的不同。正式地，在 T 个样本的序列上，对比于 $\mathcal{H}$ 而言，算法 A 的缺憾度可以定义为

$$\mathrm{Regret}_A(h,T)=\sup_{(x_1,y_1),\cdots,(x_T,y_T)}\left[\sum_{t=1}^{T}|p_t-y_t|-\sum_{t=1}^{T}|h(x_t)-y_t|\right] \tag{21.1}$$

对比于假设集 $\mathcal{H}$ 而言，算法的缺憾度为

$$\mathrm{Regret}_A(\mathcal{H},T)=\sup_{h\in\mathcal{H}}\mathrm{Regret}_A(h,T) \tag{21.2}$$

我们重申学习器的目标是对于 $\mathcal{H}$ 拥有可能的最小缺憾值。一个有趣的问题是，是否能够推导出一种缺憾度低的算法，也就是说 $\mathrm{Regret}_A(\mathcal{H},T)$ 随着回合数 T 呈次线性增长，这意味着学习器的误差率和 $\mathcal{H}$ 中最好的假设之间的差距随着 T 趋于无穷而趋向于 0。

首先我们证明这是一个不可能的任务——即使 $|\mathcal{H}|=2$ 也没有缺憾值的界为次线性的算法。事实上，考虑 $|\mathcal{H}|=\{h_0,h_1\}$，$h_0$ 函数永远返回 0 值而 h_1 返回 1。对于可以简单地等待学习器的预测然后提供相反的标签作为正确标签，就可以让在线学习算法的错误个数等于 T。相反，对任意正确标签的序列 $y_1,\cdots,y_T$，令 b 表示 $y_1,\cdots,y_T$ 中占大多数的标签，那么 h_b 犯的错误至多为 $T/2$。因此，任一在线学习算法的缺憾度应该为至少 $T-T/2=T/2$，这并不是 T 的次线性函数。这个不可能性由 Cover(1995)提出。 231

为了回避 Cover 的不可能性，我们必须进一步限制对抗的环境的力量。我们通过允许学习器随机生成其预测来做到这一点。当然，它自身不足以避免 Cover 不可能性，因为在推导的过程中我们没有对学习器的策略做任何假设。为了令随机性有意义，我们强制对抗的环境在决定 y_t 时，并不知道学习器在第 t 个回合时随机投出的硬币的结果，对抗的环境依然可以知道学习器的预测策略，甚至包括之前回合的随机投出的硬币的结果，但它不会知道学习器在第 t 个回合采用的随机投出硬币的真实值。在对策的这个(轻微)变化下，我们分析算法的期望犯错数，其中期望对学习器自身的随机性求取。也就是说，如果学习器以 $\mathbb{P}[\hat{y}_t=1]=p_t$ 输出 $\hat{y}_t$，那么在第 t 个回合它付出的期望损失为

$$\mathbb{P}[\hat{y}_t\neq y_t]=|p_t-y_t|$$

从另外一种角度理解，不认为学习器的预测落在 $\{0,1\}$ 内，而是允许其取值在 $[0,1]$ 中，并且将 $p_t\in[0,1]$ 理解为第 t 个回合预测标签为 1 的概率。

在这种假设下，可以推出一种低缺憾度算法。特别地，我们将证明以下定理。

定理 21.10 对每一个假设集$\mathcal{H}$，存在一个在线识别的算法，其预测来自于$[0,1]$，其缺憾度满足

$$\forall h\in\mathcal{H},\ \sum_{t=1}^{T}|p_t-y_t|-\sum_{t=1}^{T}|h(x_t)-y_t|\leq\sqrt{2\min\{\log(|\mathcal{H}|),1+\mathrm{Ldim}(\mathcal{H})\log(eT)\}T}$$

更进一步，没有算法能够达到比$\Omega(\sqrt{\mathrm{Ldim}(\mathcal{H})T})$小的期望缺憾值的界。

我们将会对定理的上界提供一个构造性的证明。下界的证明参见 Ben-David，Pal 和 Shalev-Shwartz(2009)。

定理 21.10 的证明依赖于带专家建议的学习的加权投票算法。这个算法很重要，下一小节我们来研究它。

加权投票

加权投票是解决带专家意见的预测问题的算法。在线学习问题中，学习器在第t个回合需要从d个给定的专家中选择。我们也允许学习器通过在d个专家上定义一个分布，从而随机地进行选择，也就是说，选择一个向量$w^{(t)}\in[0,1]^d$，其中$\sum_i w_i^{(t)}=1$，按照概率$w_i^{(t)}$选择第i个专家的意见。学习器选择一个专家之后，它接收一个代价向量$v_t\in[0,1]^d$，其中$v_{t,i}$表示听从第i个专家的代价。如果学习器的预测是随机的，那么它的损失定义为平均代价，即$\sum_i w_i^{(t)}v_{t,i}=\langle w^{(t)},v_t\rangle$。算法假设已经给定回合数目$T$。在练习 21.4 中，我们介绍如何利用倍增技巧摆脱这个依赖。

262

加权投票

输入：专家数d，回合数T

参数：$\eta=\sqrt{2\log(d)/T}$

初始化：$\tilde{w}^{(1)}=(1,\dots,1)$

对于$t=1,2,\dots$

令$w^{(t)}=\tilde{w}^{(t)}/Z_t$，其中$Z_t=\sum_i\tilde{w}_i^{(t)}$

根据$\mathbb{P}[i]=w_i^{(t)}$随机选择专家i

接收所有专家的代价$v_t\in[0,1]^d$

付出代价$\langle w^{(t)},v_t\rangle$

更新规则$\forall i$，$\tilde{w}_i^{(t+1)}=\tilde{w}_i^{(t)}e^{-\eta v_{t,i}}$

下述定理是分析加权投票算法的缺憾度界的关键。

定理 21.11 假定$T\geq 2\log(d)$，加权投票算法的界为

$$\sum_{t=1}^{T}\langle w^{(t)},v_t\rangle-\min_{i\in[d]}\sum_{t=1}^{T}v_{t,i}\leq\sqrt{2\log(d)T}$$

证明 我们有

$$\log\frac{Z_{T+1}}{Z_1}=\log\sum_i\frac{\tilde{w}_i^{(1)}}{Z_1}e^{-\eta v_{t,i}}=\log\sum_i w_i^{(1)}e^{-\eta v_{t,i}}$$

运用不等式：对所有$a\in(0,1)$，$e^{-a}\leq 1-a+\frac{a^2}{2}$，并且运用事实$\sum_i w_i^{(1)}=1$，可以得到

$$\log\frac{Z_{t+1}}{Z_t}\leqslant\log\sum_i w_i^{(t)}\left(1-\eta v_{i,t}+\frac{\eta^2 v_{i,t}^2}{2}\right)$$
$$=\log\Bigg(1-\underbrace{\sum_i w_i^{(t)}\left(\eta v_{i,t}-\frac{\eta^2 v_{i,t}^2}{2}\right)}_{\triangleq b}\Bigg)$$

接下来，注意到 $b\in(0,1)$，因此，对不等式 $1-b\leqslant e^{-b}$ 两边取对数可以得到不等式 $\log(1-b)\leqslant -b$，其对于所有 $b\leqslant 1$ 成立。可以得到 253

$$\log\frac{Z_{t+1}}{Z_t}\leqslant-\sum_i w_i^{(t)}\left(\eta v_{i,t}-\frac{\eta^2 v_{i,t}^2}{2}\right)$$
$$=-\eta\langle \boldsymbol{w}^{(t)},\boldsymbol{v}_t\rangle+\frac{\eta^2\sum_i w_i^{(t)}v_{i,t}^2}{2}$$
$$\leqslant-\eta\langle \boldsymbol{w}^{(t)},\boldsymbol{v}_t\rangle+\frac{\eta^2}{2}$$

在 t 上对该不等式求和，得到

$$\log(Z_{T+1})-\log(Z_1)=\sum_{t=1}^{T}\log\frac{Z_{t+1}}{Z_t}\leqslant-\eta\sum_{t=1}^{T}\langle \boldsymbol{w}^{(t)},\boldsymbol{v}_t\rangle+\frac{T\eta^2}{2}\tag{21.3}$$

接下来，我们寻找 Z_{T+1} 的下界。对任意 i，可以写成 $w_i^{(T+1)}=e^{-\eta\sum_t v_{i,t}}$，并且我们得到

$$\log Z_{T+1}=\log\Big(\sum_i e^{-\eta\sum_t v_{i,t}}\Big)\geqslant\log\Big(\max_i e^{-\eta\sum_t v_{i,t}}\Big)=-\eta\min_i\sum_t v_{i,t}$$

结合上式和公式(21.3)，并且运用事实 $\log Z_1=\log d$，得到

$$-\eta\min_i\sum_t v_{i,t}-\log(d)\leqslant-\eta\sum_{t=1}^{T}\langle \boldsymbol{w}^{(t)},\boldsymbol{v}_t\rangle+\frac{T\eta^2}{2}$$

重新整理得到

$$\sum_{t=1}^{T}\langle \boldsymbol{w}^{(t)},\boldsymbol{v}_t\rangle-\min_i\sum_t v_{i,t}\leqslant\frac{\log(d)}{\eta}+\frac{\eta T}{2}$$

将 η 的值代入方程可以得到我们的结论。 ■

定理 21.10 的证明

有了加权投票算法和定理 21.11，我们现在可以证明定理 21.10。我们从较为简单的情形开始，即$\mathcal{H}$是有限集，并且记$\mathcal{H}=\{h_1,\ldots,h_d\}$。在这个情况下，我们视每个假设 h_i 为一个专家，其建议即预测 $h_i(\boldsymbol{x}_t)$，其代价为 $v_{i,t}=|h_i(\boldsymbol{x}_t)-y_t|$。因此算法的预测为 $p_t=\sum_i w_i^{(t)}h_i(\boldsymbol{x}_t)\in[0,1]$，损失为

$$|p_t-y_t|=\left|\sum_{i=1}^{d}w_i^{(t)}h_i(\boldsymbol{x}_t)-y_t\right|=\left|\sum_{i=1}^{d}w_i^{(t)}(h_i(\boldsymbol{x}_t)-y_t)\right|$$

如果 $y_t=1$，那么对所有的 i，$h_i(\boldsymbol{x}_t)-y_t\leqslant 0$。因此，上式等价于 $\sum_i w_i^{(t)}|h_i(\boldsymbol{x}_t)-y_t|$。如果 $y_t=0$，那么对所有的 i，$h_i(\boldsymbol{x}_t)-y_t\geqslant 0$。上述也等价于 $\sum_i w_i^{(t)}|h_i(\boldsymbol{x}_t)-y_t|$。总之，我们证明了 254

$$|p_t-y_t|=\sum_{i=1}^{d}w_i^{(t)}|h_i(\boldsymbol{x}_t)-y_t|=\langle \boldsymbol{w}^{(t)},\boldsymbol{v}_t\rangle$$

更进一步，对任一 i，$\sum_t v_{i,t}$ 就是假设 h_i 所犯的错误的数量。应用定理 21.11，可以得

到以下推论。

推论 21.12　令$\mathcal{H}$表示有限假设集。存在一个在线的识别学习算法，其预测在[0，1]上，其缺憾度的界为

$$\sum_{t=1}^{T}|p_t-h_t|-\min_{h\in\mathcal{H}}\sum_{t=1}^{T}|h_t(x_t)-h_t|\leqslant\sqrt{2\log(|\mathcal{H}|)T}$$

下面，我们考虑一般的假设集的情况。先前，我们对每一个单独的假设构造一个专家。但是，如果$\mathcal{H}$是无限的，将导致一个无意义的界。应对的主要思路是以更复杂的方法构造一个专家集合，挑战在于如何定义专家集合，使其一方面不太大，另一方面包含给出精确预测的专家。

我们构造这样的专家集合，其对于任一假设$h\in\mathcal{H}$和每一个实例序列x_1，x_2，…，x_T，集合里存在至少一个专家去预测和h在这些实例上的一致，对任一$L\leqslant\mathrm{Ldim}(\mathcal{H})$和任一序列$1\leqslant i_1<i_2<\cdots<i_L\leqslant T$，我们定义一个专家。这个专家模拟SOA算法（参见前面的小节）和环境在实例序列x_1，x_2，…，x_T上的游戏，假设SOA在回合i_1，i_2，…，i_L上精确犯错。专家可以用下述算法定义。

Expert(i_1，i_2，…，i_L)

输入：假设类$\mathcal{H}$，下标$i_1<i_2<\cdots<i_L$

初始化：$V_1=\mathcal{H}$

对于$t=1$，2，…，T

　接收x_t

　对于$r\in\{0,1\}$，令$V_t^{(r)}=\{h\in V_t:h(x_t)=r\}$

　定义$\hat{y}_t=\arg\max_r \mathrm{Ldim}(V_t^{(r)})$

　　（约定预测$\hat{y}_t=0$）

　若$t\in\{i_1,i_2,\cdots,i_L\}$

　　预测$\hat{y}_t=1-\hat{y}_t$

　否则预测$\hat{y}_t=\hat{y}_t$

　更新$V_{t+1}=V_t^{(\hat{y}_t)}$

注意到，每一个这样的专家在每回合t只观测实例x_1，x_2，…，x_t给出预测，适用在线学习算法或在异这些专家的加权投票算法。

为了分析算法，我们首先注意到专家的数量为

$$d=\sum_{L=0}^{\mathrm{Ldim}(\mathcal{H})}\binom{T}{L} \tag{21.4}$$

可以证明，当$T\geqslant\mathrm{Ldim}(\mathcal{H})+2$，等式右边的的界为$(eT/\mathrm{Ldim}(\mathcal{H}))^{\mathrm{Ldim}(\mathcal{H})}$（证明参见引理A.5）。

定理21.11告诉我们，加权投票的期望犯错数最大为最好的专家犯错数乘以$\sqrt{2\log(d)T}$。我们接下来将证明最好的专家的犯错数最多为$\mathcal{H}$中最好的假设的犯错数。下述关键引理证明，在任一实例序列上，对任一假设$h\in\mathcal{H}$存在一个专家表现相同。

引理21.13　令$\mathcal{H}$表示任一$\mathrm{Ldim}(\mathcal{H})<\infty$的假设集。令$x_1$，$x_2$，…，$x_T$表示实例序列。对任一$h\in\mathcal{H}$，存在$L\leqslant\mathrm{Ldim}(\mathcal{H})$和下标$1\leqslant i_1<i_2<\cdots<i_L\leqslant T$，使得在$x_1$，$x_2$，…，

x_T 上运行专家算法$(i_1, i_2, \cdots, i_L)$，对于在线学习的每一个回合 $t=1, 2, \cdots, T$，其预测为$h(x_t)$。

证明 固定$h\in\mathcal{H}$和序列 $x_1, x_2, \cdots, x_T$。我们需要构造 L 和下标$i_1, i_2, \cdots, i_L$。考虑在输入$(x_1, h(x_1)), (x_2, h(x_2)), \cdots, (x_T, h(x_T))$上运行 SOA 算法。SOA 在每个输入上犯错至多为 $\mathrm{Ldim}(\mathcal{H})$。令 L 表示 SOA 犯错的个数，并令$\{i_1, \cdots, i_L\}$为犯错的回合的集合。

现在，考虑在 $x_1, x_2, \cdots, x_T$ 上运行专家算法$(i_1, i_2, \cdots, i_L)$。根据构造，专家$(i_1, i_2, \cdots, i_L)$维护的集合 V_t 等价于运行在序列$(x_1, h(x_1)), (x_2, h(x_2)), \cdots, (x_T, h(x_T))$上由 SOA 维护的集合。SOA 的预测和 h 的预测不同当且仅当该回合在$\{i_1, \cdots, i_L\}$中。因为专家$(i_1, i_2, \cdots, i_L)$当 t 不属于$\{i_1, \cdots, i_L\}$时预测同 SOA 一样，如果 t 属于$\{i_1, \cdots, i_L\}$则预测与 SOA 的预测相反，我们推论：专家的预测和 h 的预测永远是一致的。 ■

特别地，对于$\mathcal{H}$中在样本序列上犯错最少的假设，上述引理成立，因此我们得到下述推论。

推论 21.14 令表示样本序列，$\mathcal{H}$表示 $\mathrm{Ldim}(\mathcal{H})<\infty$的假设集。存在 $L\leq\mathrm{Ldim}(\mathcal{H})$和下标 $1\leq i_1<i_2<\cdots<i_L\leq T$，使得专家$(i_1, i_2, \cdots, i_L)$最多犯错数和最好的假设 $h\in\mathcal{H}$一致，即在样本序列上犯错数为

$$\min_{h\in\mathcal{H}}\sum_{t=1}^{T}|h(x_t)-y_t|$$

结合定理 21.11，定理 21.10 的上界部分得证。 [256]

21.3 在线凸优化

在第 13 章中我们研究了凸学习问题，并且在不可知 PAC 学习框架下介绍了这些问题的可学习性。本节我们介绍在线学习框架下凸问题可学习性的类似结果。特别地，我们考虑下述问题：

在线凸优化

定义：假设类$\mathcal{H}$，域 Z，损失函数ℓ：$H\times Z\to\mathbb{R}$

假定：$\mathcal{H}$为凸；$\forall z\in Z$，$\ell(\cdot, z)$是凸函数

对于 $t=1, 2, \cdots, T$

　学习器预测一个向量 $w^{(t)}\in\mathcal{H}$

　环境响应 $z_t\in Z$

　学习器遭受损失$\ell(w^{(t)}, z_t)$

和在线识别问题一样，我们分析算法的缺憾度。回顾在考虑竞争性假设的条件下，在线学习算法的缺憾度如下定义，此处将假设写成权重向量 $w^*\in\mathcal{H}$：

$$\mathrm{Regret}_A(w^*, T)=\sum_{t=1}^{T}\ell(w^{(t)}, z_t)-\sum_{t=1}^{T}\ell(w^*, z_t) \tag{21.5}$$

与前面类似，相对于一系列竞争向量的集合$\mathcal{H}$，算法的缺憾度定义为

$$\mathrm{Regret}_A(\mathcal{H}, T)=\sup_{w^*\in\mathcal{H}}\mathrm{Regret}_A(w^*, T)$$

在第 14 章，我们介绍了不可知 PAC 模型下的随机梯度下降方法，用于解决凸学习问

题。我们现在介绍一个非常相似的算法——在线梯度下降，用于解决在线凸学习问题。

在线梯度下降

参数：$\eta>0$

初始化：$w^{(1)}=0$

对于 $t=1,2,\cdots,T$

　预测 $w^{(t)}$

　获取 z_t 并令 $f_t(\cdot)=\ell(\cdot,z_t)$

　选择 $v_t\in\partial f_t(w^{(t)})$

　更新：

　　1) $w^{(t+\frac{1}{2})}=w^{(t)}-\eta v_t$

　　2) $w^{(t+1)}=\underset{w\in\mathcal{H}}{\arg\min}\|w-w^{(t+\frac{1}{2})}\|$

267

定理 21.15　对任意 $w^*\in\mathcal{H}$，在线梯度下降算法的缺憾度的界如下：

$$\mathrm{Regret}_A(w^*,T)\leq\frac{\|w^*\|^2}{2\eta}+\frac{\eta}{2}\sum_{t=1}^{T}\|v_t\|^2$$

如果我们进一步假设 f_t 对任意 t 具有 ρ-利普希茨性，令 $\eta=\frac{1}{\sqrt{T}}$，得到

$$\mathrm{Regret}_A(w^*,T)\leq\frac{1}{2}(\|w^*\|^2+\rho^2)\sqrt{T}$$

如果我们进一步假设 $\mathcal{H}$ 是 B-有界的，并令 $\eta=\frac{B}{\rho\sqrt{T}}$，则

$$\mathrm{Regret}_A(\mathcal{H},T)\leq B\rho\sqrt{T}$$

证明　下述分析和有投影的随机梯度下降的分析相似。运用投影引理、$w^{(t+\frac{1}{2})}$ 的定义和子梯度的定义，得到对任意的 t，

$$\begin{aligned}
&\|w^{(t+1)}-w^*\|^2-\|w^{(t)}-w^*\|^2\\
&=\|w^{(t+1)}-w^*\|^2-\|w^{(t+\frac{1}{2})}-w^*\|^2+\|w^{(t+\frac{1}{2})}-w^*\|^2-\|w^{(t)}-w^*\|^2\\
&\leq\|w^{(t+\frac{1}{2})}-w^*\|^2-\|w^{(t)}-w^*\|^2\\
&=\|w^{(t)}-\eta v_t-w^*\|^2-\|w^{(t)}-w^*\|^2\\
&=-2\eta\langle w^{(t)}-w^*,v_t\rangle+\eta^2\|v_t\|^2\\
&\leq-2\eta(f_t(w^{(t)})-f_t(w^*))+\eta^2\|v_t\|^2
\end{aligned}$$

对所有的 t 求和，注意到左侧是一个伸缩和，可以得到

$$\|w^{(T+1)}-w^*\|^2-\|w^{(1)}-w^*\|^2\leq-2\eta\sum_{t=1}^{T}(f_t(w^{(t)})-f_t(w^*))+\eta^2\sum_{t=1}^{T}\|v_t\|^2$$

整理不等式，运用 $w^{(1)}=0$，得到：

$$\begin{aligned}
\sum_{t=1}^{T}(f_t(w^{(t)})-f_t(w^*))&\leq\frac{\|w^{(1)}-w^*\|^2-\|w^{(T+1)}-w^*\|^2}{2\eta}+\frac{\eta}{2}\sum_{t=1}^{T}\|v_t\|^2\\
&\leq\frac{\|w^*\|^2}{2\eta}+\frac{\eta}{2}\sum_{t=1}^{T}\|v_t\|^2
\end{aligned}$$

这证明了定理的第一个界。而第二个界来自于由假设 f_t 是 ρ-利普希茨的，也就是说

$\|v_t\| \leqslant \rho$。 ■

21.4 在线感知器算法

感知器是一个经典的二值分类问题的在线学习算法，其假设集为齐次半空间，即$\mathcal{H}=\{x \mapsto \operatorname{sign}(\langle w, x\rangle): w \in \mathbb{R}^d\}$。在9.1.2节我们介绍了感知器的批处理版本，其目标是解决$\mathcal{H}$上的ERM问题。现在介绍感知器算法的在线版本。 [258]

令$\mathcal{X}=\mathbb{R}^d$，$\mathcal{Y}=\{-1, 1\}$。在第t个回合，学习器接受向量$x_t \in \mathbb{R}^d$。学习器维护一个权重向量$w^{(t)} \in \mathbb{R}^d$，并且预测$p_t=\operatorname{sign}(\langle w^{(t)}, x_t\rangle)$，然后接收$y_t \in \mathcal{Y}$，如果$y_t \neq p_t$则付出代价1，反之代价为0。

学习器的目标是尽可能少犯预测错误。在21.1节中我们描绘了最优化算法，并且证明了可实现的最优误差界取决于假设集的Littlestone维。随后我们论述，如果$d \geqslant 2$，则$\operatorname{Ldim}(\mathcal{H})=\infty$，这意味着我们不可能只犯较少的错误。事实上，考虑树：$v_1=\left(\frac{1}{2}, 1, 0, \cdots, 0\right)$，$v_2=\left(\frac{1}{4}, 1, 0, \cdots, 0\right)$，$v_3=\left(\frac{3}{4}, 1, 0, \cdots, 0\right)$，等等。由于实数的稠密性，打散这棵树的集合是$\mathcal{H}$的子集，其中$\mathcal{H}$包含所有以形如$w=(-1, a, 0, \cdots, 0)(a \in [0, 1])$的参数所表示的假设，可以得出结论：$\operatorname{Ldim}(\mathcal{H})=\infty$。

为了回避这个不可能性，感知器算法运用替代凸损失函数的技巧(参见12.3节)。这也和我们在第15章所研究的间隔的概念相关。

只要$\langle w, x\rangle$的符号和y不相等，权重向量w预测样例(x, y)发生错误。因此，我们可以重写0-1损失函数如下：

$$\ell(w,(x,y))=\mathbb{1}_{[y\langle w,x\rangle\leqslant 0]}$$

在每次算法预测错误的回合，我们可以利用合页(hinge)损失作为替代凸损失函数

$$f_t(w)=\max\{0,1-y_t\langle w,x_t\rangle\}$$

合页损失函数满足下列两个条件：

- f_t是一个凸函数；
- 对所有的w，$f_t(w) \geqslant \ell(w, (x_t, y_t))$。特别地，对$w^{(t)}$也成立。

在每次算法预测正确的回合，我们定义$f_t(w)=0$。显然在这种情况下f_t也是凸函数，而且$f_t(w)=\ell(w, (x_t, y_t))=0$。

评注 在12.3节中我们对所有的训练样例运用了相同的替代损失函数。在线学习的模型中，我们允许依据特殊的回合甚至可以依据$w^{(t)}$来使用替代函数。我们能使用专门的替代函数基于在线学习中对于最坏情况的分析。

现在，在函数序列$f_1, \cdots, f_T$和全体的$\mathbb{R}^d$中向量构成的假设集上，运行在线梯度下降算法。回顾，算法初始化$w^{(1)}=\mathbf{0}$，且更新公式为

$$w^{(t+1)}=w^{(t)}-\eta v_t$$

其中$v_t \in \partial f_t(w_t)$。在这里，如果$y_t\langle w^{(t)}, x_t\rangle>0$，则$f_t$是零函数，所以$v_t=\mathbf{0}$。否则，容易验证$v_t=-y_t x_t$在$\partial f_t(w^{(t)})$之中。因此可以得到更新公式为 [259]

$$w^{(t+1)}=\begin{cases} w^{(t)} & \text{若 } y_t\langle w^{(t)}, x_t\rangle>0 \\ w^{(t)}+\eta y_t x_t, & \text{否则} \end{cases}$$

记$\operatorname{sign}(\langle w^{(t)}, x_t\rangle) \neq y_t$的回合集合为$\mathcal{M}$。注意到在第$t$个回合，感知器的预测可以写成

$$p_t=\operatorname{sign}(\langle w^{(t)}, x_t\rangle)=\operatorname{sign}\left(\eta \sum_{i \in \mathcal{M}, i<t} y_i\langle x_i, x_t\rangle\right)$$

这个形式意味着感知器算法的预测和集合$\mathcal{M}$不依赖于η的真实值，只要$\eta>0$即可。因此我们得到感知器算法如下：

感知器

初始化：$w_1=\mathbf{0}$
对于$t=1, 2, \cdots, T$
　接收x_t
　预测$p_t=\mathrm{sign}(\langle w^{(t)}, x_t\rangle)$
　若$y_t\langle w^{(t)}, x_t\rangle\leqslant 0$
　　$w^{(t+1)}=w^{(t)}+y_t x_t$
　否则$w^{(t+1)}=w^{(t)}$

为了分析感知器算法，我们运用前一部分给出的在线梯度下降算法。在这里，感知器算法中用到的f_t的子梯度为$v_t=-\mathbb{1}_{[y_t\langle w^{(t)}, x_t\rangle\leqslant 0]}y_t x_t$。实际上，感知器算法的更新公式为$w^{(t+1)}=w^{(t)}-v_t$，并且如前讨论的，这等价于对任意$\eta>0$，$w^{(t+1)}=w^{(t)}-\eta v_t$。因此，由定理21.15可以知道

$$\sum_{t=1}^{T} f_t(w^{(t)})-\sum_{t=1}^{T} f_t(w^*)\leqslant \frac{1}{2\eta}\|w^*\|_2^2+\frac{\eta}{2}\sum_{t=1}^{T}\|v_t\|_2^2$$

由于$f_t(w^{(t)})$是关于0-1损失的替代，可知$\sum_{t=1}^{T} f_t(w^{(t)})\geqslant|\mathcal{M}|$。记$R=\max_t\|x_t\|$，可得

$$|\mathcal{M}|-\sum_{t=1}^{T} f_t(w^*)\leqslant \frac{1}{2\eta}\|w^*\|_2^2+\frac{\eta}{2}|\mathcal{M}|R^2$$

令$\eta=\frac{\|w^*\|}{R\sqrt{|\mathcal{M}|}}$，整理得到

$$\mathcal{M}-R\|w^*\|\sqrt{|\mathcal{M}|}-\sum_{t=1}^{T} f_t(w^*)\leqslant 0 \tag{21.6}$$

260

这个不等式推出以下定理。

定理21.16　假定感知器算法在序列$(x_1, y_1), \cdots, (x_T, y_T)$上运行并且令$R=\max_t\|x_t\|$，记$\mathcal{M}$为感知器算法预测错误的回合，令$f_t(w)=\mathbb{1}_{[t\in\mathcal{M}]}[1-y_t\langle w, x_t\rangle]_+$。则对任意的$w^*$，有

$$|\mathcal{M}|\leqslant\sum_t f_t(w^*)+R\|w^*\|\sqrt{\sum_t f_t(w^*)}+R^2\|w^*\|^2$$

特别地，如果存在w^*，使得对所有t，成立$y_t\langle w^*, x_t\rangle\geqslant 1$，则

$$|\mathcal{M}|\leqslant R^2\|w^*\|^2$$

证明　这个定理可以从公式(21.6)和下述论断得到：给定$x, b, c\in\mathbb{R}_+$，不等式$x-b\sqrt{x}-c\leqslant 0$意味着$x\leqslant c+b^2+b\sqrt{c}$。这个论断可以简单地从分析凸抛物线$Q(y)=y^2-by-c$的根得到。■

定理21.16的最后一个假设称为最大间隔可分离性（参见第15章）。也就是说，存在w^*不仅满足将样本点分在半空间正确的一侧，而且保证其不太靠近决策面。更严谨地说，

到决策面的距离至少为 $\gamma=\frac{1}{\|w^*\|}$，并且上述定理中M的界可以写为$\left(\frac{R}{\gamma}\right)^2$。

如果可分离性假设不成立，则界受项$[1-y_t\langle w, x_t\rangle]_+$的影响。该项度量了可分离性所需边界被破坏的程度。

作为最后的注解，我们注意到存在某些 w^* 在序列上零错误，但是感知器将会犯很多错误。实际上，这是$\mathrm{Ldim}(\mathcal{H})=\infty$的直接结果。为了避免这种不可能性，我们对样例序列做出更多假设——定理21.16的界只有在替代损失的累积 $\sum_t f_t(w^*)$不太大的情况下才有意义。

21.5 小结

在本章我们研究了在线学习模型。许多在PAC模型中推导出来的结果在在线学习模型中都有相似的对应。首先，我们介绍了一个组合的维度，即Littlestone维，来刻画在线可学习性。为了证明这一点，我们介绍了(可实现情况下的)SOA算法和(不可实现情况下的)加权投票算法。同时研究了在线凸优化问题，并且证明了，只要损失函数是凸的、具有利普希茨性的，则在线梯度下降算法是成功的学习算法。最后作为在线梯度下降和替代凸损失函数的结合，我们介绍了在线的感知器算法。

21.6 文献评注

标准最优化算法是从Littlestone(1988)的基础工作推导得出的。对不可实现情况的推广，以及其他变式包括基于间隔的Littlestone维，由Ben-David等人(2009)提出。除了识别之外，在线可学习性的描绘由Abernethy、Bartlett、Rakhlin & Tewari (2008)、Rakhlin、Sridharan及Tewari(2010)、Daniely等人(2011)得到。加权投票算法由Littlestone、warmuth(1994)和Vovk(1990)提出。 291

在线凸优化的概念由Zinkevich(2003)提出，但是这一系列出现于早些年的Gordon(1999)。感知器则要追溯到Rosenblatt(1958)。可实现情况(包括间隔假设)的分析出现在Agmon(1954)、Minsky和Papert(1969)、Freund和Schapire(1999)基于可实现情况的归约，展示了不可实现的情况下的平方合页误差的分析。不可实现情况下的合页误差的直接分析由Gentile(2003)给出。

更多的信息我们推荐阅读Cesa-Bianchi和Lugosi(2006)以及Shalev-Shwartz(2011)的著作。

21.7 练习

21.1 寻找假设集$\mathcal{H}$和一个样本序列，使得一致性算法犯$|H|-1$个错误。

21.2 寻找假设集$\mathcal{H}$和一个样本序列，使得二分算法的误差界是紧的。

21.3 令$d\geq 2$，$\mathcal{X}=\{1, \cdots, d\}$，$\mathcal{H}=\{h_i: i\in[d]\}$，其中$h_i(x)=\mathbb{1}_{x=i}$。计算$M_{\mathrm{Halving}}(\mathcal{H})$(也就是说，推导其下界和上界，然后证明二者相等)。

21.4 倍增技巧：

定理21.15中，参数η取决于时间范围T。在这个练习中，我们介绍如何通过一个简单的技巧解除这种依赖关系。

考虑族遗憾度的界形如$a\sqrt{T}$的算法，但是其参数需要关于T的知识。倍增技巧如下所述，它允许我们转化这样的算法为不需要知道时间范围的算法。其思路是分

期时间为大小递增的时期，并且在每个时期上运行原算法。

倍增技巧

输入：参数依赖时间范围的算法 A

对于 $m=0, 1, 2, \cdots$

　在 2^m 个回合上运行 A，$t=2^m, \cdots, 2^{m+1}-1$

证明，如果 A 在 2^m 回合中每个时期的缺憾度至多为 $\alpha\sqrt{2^m}$，则总的缺憾度至多为

$$\frac{\sqrt{2}}{\sqrt{2}-1}\alpha\sqrt{T}$$

21.5　**在线到批处理的转化**：在这个练习中，我们论证成功的在线学习算法如何用于推导成功的 PAC 学习器。

考虑一个二值分类的 PAC 学习问题，其实例域为 $\mathcal{X}$，假设集为 $\mathcal{H}$。假设存在一个在线学习算法 A，其误差界 $M_A(\mathcal{H})<\infty$。考虑在 T 个样本的序列上运行该算法，样本独立同分布地采样于实例空间 $\mathcal{X}$ 的分布 $\mathcal{D}$，并且被 $h^*\in\mathcal{H}$ 标注。假定在每个回合 t，算法的预测基于假设 $h_t: \mathcal{X}\to\{0, 1\}$。证明：

$$\mathbb{E}_{t\sim[T]}\left[I_{\mathcal{D}}(h_t)\right]\leq\frac{M_A(\mathcal{H})}{T}$$

其中，期望是对实例的随机选择和 $[T]$ 上服从均匀分布的 t 的随机选择同时求取的。

262
:
263

提示：采用定理 14.8 的证明过程中相似的论据。

聚　类

聚类是一种运用广泛的探索性数据分析技术。纵观所有的学科，从社会学到生物学再到计算机科学，人们对数据产生第一直觉往往是通过对数据进行有意义的分组。例如，计算机生物学家根据在不同实验中基因表达的相似性对基因进行聚类；零售商根据顾客概况对客户聚类，来定向进行市场营销；天文学家根据星星的空间距离对其聚类。

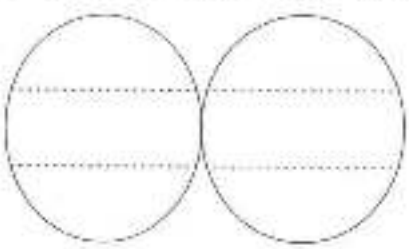
很自然，首先需要弄清聚类是什么？直观上讲，聚类是将对象进行分组的一项任务，使相似的对象归为一类，不相似的对象归为不同类。很明显，这种描述是非常模糊而且不准确的。然而令人惊奇的是，很难提出一种更为严格的定义。

造成这种困难的原因有很多。一个最基本的问题是上述涉及的两个目标在很多情况下是互相冲突的。从数学上讲，虽然聚类共享具有等价关系甚至传递关系，但是相似性(或距离)不具有传递关系。具体而言，假定有一对象序列，x_1，…，x_m，x_i 与其邻元素 x_{i-1} 和 x_{i+1} 非常相似，但是 x_1 和 x_m 非常不相似。如果认定不论什么时候，相似的两个元素必须在相同的聚类中，那么我们要将这一序列的所有元素放在同一个聚类。若如此，不相似的元素 x_1 和 x_m 将共享同一聚类，因此违背第二条要求。

为了进一步说明，假定我们希望将下图中的点聚为两类。

一类聚类算法强调不要将紧邻的点分离开来(例如，将在 22.1 节中讲述的单链接算法)，这类算法会将这种输入划分成两条平行线：

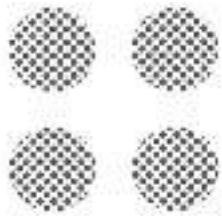
与此相反，另一类聚类算法强调同一聚类的点彼此不能远离(例如，将在 22.1 节中讲述的 2-均值算法)，这类算法会用一条垂直的线将输入分为左右两部分：



另一个基本问题是聚类缺乏实际情况。这是无监督学习的共同问题。本书到目前为止，我们主要处理的是监督学习(例如，从已标记的训练数据中学习一个分类器)。监督学习的目标很明确——我们希望学到一个分类器，使之预测未来样本的标号尽可能准确。更

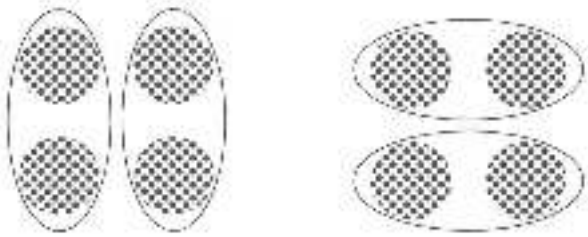
重要的是，监督学习可以用已标注数据计算经验风险来评估学习是否成功或假设的风险。相反，聚类是一种无监督学习问题：即我们不预测标号。我们希望将数据进行有意义的整合。因此，对于聚类，并没有明确的成功评估过程。事实上，即使已知数据分布的全部知识，我们也并不清楚什么是数据的正确聚类，或者如何评估聚类效果。

如下图所示，考虑$\mathbb{R}^2$上的点集：



255

假定现要求将其聚为两类。我们有两种非常合理的解决方案：



这种现象不是人为设定的，而是确实会出现在实际应用中。一个给定的对象集合，可以有多种有意义的划分方式。这可能是因为对象间的距离（或相似性）有多种隐式的定义，例如，将演讲者的录音根据演讲者的口音聚类或根据内容聚类，将影评根据影片主题聚类或根据评论情感聚类，将图画根据主题聚类或根据类型聚类，等等。

总而言之，给定一个数据集，有多种不同的聚类解决方案，因此多种聚类算法，对相同输入数据，产生的聚类存在很大差异。

一种聚类模型

聚类任务随着输入类型和期望输出类型的变化而改变。具体而言，我们会重点关注下述情形：

输入——元素集$\mathcal{X}$和距离函数。即，函数$d: \mathcal{X}\times\mathcal{X}\to\mathbb{R}$是对称的，对所有的$x\in\mathcal{X}$满足$d(x, x)=0$，而且一般会满足三角不等式。距离函数可以用相似性函数替代，相似性函数$s: \mathcal{X}\times\mathcal{X}\to[0, 1]$是对称的，并且对所有的$x\in\mathcal{X}$满足$s(x, x)=1$。此外，一些聚类算法需要指定输入参数$k$（决定聚类的数目）。

输出——域集合$\mathcal{X}$的一种划分。即，$C=(C_1, \cdots, C_k)$，其中$\bigcup_{i=1}^{k} C_i=\mathcal{X}$，并且对所有的$i\neq j$有$C_i\cap C_j=\emptyset$。一些情况中，聚类是“柔软的”，即将$\mathcal{X}$按概率形式划分到不同的聚类，对定义域内的点$x\in\mathcal{X}$，给出一个向量$(p_1(x), \cdots, p_k(x))$，其中$p_i(x)=\mathbb{P}[x\in C_i]$是$x$属于类$C_i$的概率。另一种可能的输出形式是聚类系统树图（dendrogram，来源于希腊语，dengron=tree，gramma=drawing），这是一种域子集的分层树，其叶子节点对应单元素集，根节点表示全域。我们会在下文中详述。

下面，我们总结一些最常用的聚类方法。在本章的最后一节，我们会在更高层次上讨论聚类是什么这一问题。

22.1 基于链接的聚类算法

基于链接的聚类算法可能是最简单最直接的聚类形式。这类算法一般需要一系列循环。这类算法从一些琐碎的聚类开始，将每个数据点作为一个单点聚类，然后，这类算法循环将前一阶段中最近的两个聚类合并。因此，聚类数目随着循环过程逐渐减少。如果一直进行下去，这类算法会将所有的定义域数据点归为一个大类。为了将该类算法定义清楚，需要确定两个参数。第一，我们需要决定怎样测量(或定义)类间距离，第二，我们需要确定什么时候终止合并。聚类算法的输入需要指点两个点之间的距离函数 d。有许多方法可以将 d 进行扩展，来测量两个聚类或域子集之间的距离。最常用的方法有 266

1. 单链接聚类，类间距离定义为两类元素间的最短距离，即，

$$D(A,B) \stackrel{\text{def}}{=} \min\{d(x,y): x \in A, y \in B\}$$

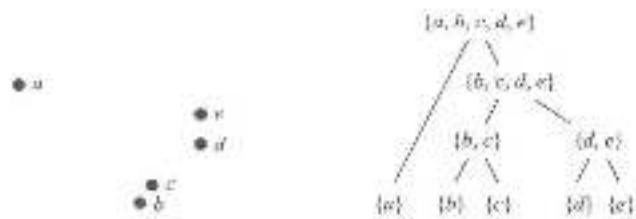
2. 平均链接聚类，类间距离定义为两类元素间距离的平均值，即，

$$D(A,B) \stackrel{\text{def}}{=} \frac{1}{|A|\,|B|} \sum_{x \in A, y \in B} d(x,y)$$

3. 最大链接聚类，类间距离定义为两类元素间的最大距离，即，

$$D(A,B) \stackrel{\text{def}}{=} \max\{d(x,y): x \in A, y \in B\}$$

基于链接的聚类算法是凝聚式的，一开始，数据完全是碎片化的，然后逐步构建越来越大的聚类。如果没有加入停止规则，这类算法的结果可以用聚类系统树图来描述：即，一个域子集构成的树，其叶子节点是单元素集，根节点为全域。例如，如下左图所示，输入元素是 $\mathcal{X}=\{a, b, c, d, e\}\subset\mathbb{R}^2$，采用的距离函数是欧几里得距离，如下右图为生成的系统树图：



单链接算法和 Kruskal 算法很相似，目的是在加权图上找到一个最小生成树。试想一幅图，图的顶点是 $\mathcal{X}$ 中元素，边 (x, y) 的权重是距离 $d(x, y)$。每次单链接算法将两个聚类进行合并，相当于在上图中添加一条边。单链接算法得到的边集合和最小生成树是一致的。 267

如果想将一个系统树图转化为一个空间(聚类)划分，则需要设定停止准则。常用的停止准则包括：

- 固定类的数量　固定参数 k，当聚类数目为 k 时停止聚类。
- 设定距离上限　固定 $r\in\mathbb{R}_+$，当所有的组间距离都超过 r 时停止聚类。我们也可以设定 r 为 $\alpha\max\{d(x, y): x, y\in\mathcal{X}\}$，其中 $\alpha<1$。这种情况下，停止准则被称为"折合距离上限"。

22.2 k 均值算法和其他代价最小聚类

另一种流行的聚类算法是首先对可能的聚类定义一个代价函数，聚类算法的目标是

寻找一种使代价最小的划分。在这类范例中，聚类任务转化为一个优化问题。目标函数是一个从输入$(\mathcal{X}, d)$和聚类方案$C=(C_1, \cdots, C_k)$映射到正实数的函数。给定一个这样的目标函数，我们将其表示为G，对于给定的一个输入$(\mathcal{X}, d)$，聚类算法的目标被定义为寻找一种聚类C使$G((\mathcal{X}, d), C)$最小。为了达到上述目标，需要运用一些合适的搜索算法。

事实证明，大多数的聚类优化问题是NP难问题，甚至有些问题的近似也是NP难问题。因此，当人们谈论k均值算法，一般是指一些特殊的近似算法，而不是最小化问题的损失函数或精确解。

许多常见的目标函数要求指定参数聚类数目k。实际上，这通常需要算法的使用者根据给定的聚类问题来选定k值。

下面，我们会介绍几种最常用的目标函数。

k均值算法目标函数是最流行的聚类目标。在k均值算法中，数据被划分到不相交的集合$C_1, \cdots, C_k$中，其中每个C_i由其中心点μ_i代表。假定输入集$\mathcal{X}$被嵌入到更广的测度空间$(\mathcal{X}', d)$(因此$\mathcal{X} \subseteq \mathcal{X}'$)，中心点是$\mathcal{X}'$的元素。$k$均值的目标函数测量$\mathcal{X}$中各点与其对应聚类中心点的平方距离。$C_i$中心点被定义为

$$\mu_i(C_i) = \operatorname*{argmin}_{\mu \in \mathcal{X}'} \sum_{x \in C_i} d(x, \mu)^2$$

那么，k均值算法的目标是

$$G_{k\text{-means}}((\mathcal{X}, d), (C_1, \cdots, C_k)) = \sum_{i=1}^{k} \sum_{x \in C_i} d(x, \mu_i(C_i))^2$$

268

也可以写成如下形式

$$G_{k\text{-means}}((\mathcal{X}, d), (C_1, \cdots, C_k)) = \min_{\mu_1, \cdots, \mu_k \in \mathcal{X}'} \sum_{i=1}^{k} \sum_{x \in C_i} d(x, \mu_i)^2 \tag{22.1}$$

k均值算法的目标函数目的明确，例如，在数字通讯任务中，可以将$\mathcal{X}$视为传输信号的集合。尽管$\mathcal{X}$可能是一个包含实值向量的大集合，数字传输只允许对每个信号每次传输有限位。在这种限制下，实现较好传输的一种方法是将$\mathcal{X}$的每个成员用有限集$\mu_1, \cdots, \mu_k \in \mathcal{X}'$中一个"近似的"成员代替。$k$均值算法的目标可以看成这种传输表示方案失真程度的一种测量。

k中心点算法目标函数和k均值算法的目标函数相似，不同处是k中心点算法要求聚类中心点是输入集的成员。目标函数定义为

$$G_{k\text{-medoid}}((\mathcal{X}, d), (C_1, \cdots, C_k)) = \min_{\mu_1, \cdots, \mu_k \in \mathcal{X}} \sum_{i=1}^{k} \sum_{x \in C_i} d(x, \mu_i)^2$$

k中位数算法目标函数和k中心点算法的目标函数十分相似，不同处是k中位数算法中，数据点和聚类中心点的"失真"是用距离测量，而不是距离的平方：

$$G_{k\text{-median}}((\mathcal{X}, d), (C_1, \cdots, C_k)) = \min_{\mu_1, \cdots, \mu_k \in \mathcal{X}} \sum_{i=1}^{k} \sum_{x \in C_i} d(x, \mu_i)$$

一个使用该目标函数的例子是工厂选址问题。考虑一项任务，该任务要求在城市设定k个消防站。在该例中可以将房屋作为数据点，目标是设定位置使得房屋与其最近消防站距离平均值最小。

前面的例子可以被统称为基于中心的目标。这类聚类问题的解决方案，由一系列中心点决定，聚类算法将每个实例分配给与之最近的类中心。更一般的情况，基于中心的目标

由一些单调函数 $f:\mathbb{R}_+\to\mathbb{R}_+$ 决定，并定义

$$G_f((\mathcal{X},d),(C_1,\cdots,C_k))=\min_{\mu_1,\cdots,\mu_k\in\mathcal{X}'}\sum_{i=1}^{k}\sum_{x\in C_i}f(d(x,\mu_i))$$

其中$\mathcal{X}'$或者为$\mathcal{X}$或者为$\mathcal{X}$的超集。

有一些目标函数并不是基于中心的。例如类内距离总和(SOD)

$$G_{SOD}((\mathcal{X},d),(C_1,\cdots,C_k))=\sum_{i=1}^{k}\sum_{x,y\in C_i}d(x,y)$$

[269]

和我们将在 22.3 节介绍的最小割目标都不是基于中心的目标。

k 均值算法

k 均值目标函数在实际的聚类应用中很常见。然而，事实证明寻找 k 均值(k-means)算法的最优解通常是计算不可行的(问题是 NP 难的，甚至接近常数近似解的求解是 NP 难的)，通常用下面这种简单的迭代算法作为替代方法，多数情况下，k 均值聚类指的是这种算法的结果而不是最小化 k 均值目标函数的结果。我们以欧几里得距离 $d(x,y)=|x-y|$ 为例描述该算法。

k 均值

输入：$\mathcal{X}\subset\mathbb{R}^n$；聚类数目 k

初始化：随机初始化中心点 $\mu_1,\cdots,\mu_k$

重复直到收敛

$\forall i\in[k]$设定 $C_i=\{x\in\mathcal{X}:i=\operatorname{argmin}_i|x-\mu_i|\}$

(配合使用任意方式的中断)

$\forall i\in[k]$更新 $\mu_i=\frac{1}{|C_i|}\sum_{x\in C_i}x$

引理 22.1 k 均值算法的每次迭代都不会使 k 均值目标函数增加(由公式(22.1)给出)。

证明 为了数学表示简单，我们使用 $G(C_1,\cdots,C_k)$来表示 k 均值算法目标，即，

$$G(C_1,\cdots,C_k)=\min_{\mu_1,\cdots,\mu_k\in\mathcal{X}'}\sum_{i=1}^{k}\sum_{x\in C_i}\|x-\mu_i\|^2 \quad (22.2)$$

很容易定义 $\mu(C_i)=\frac{1}{|C_i|}\sum_{x\in C_i}x$，记 $\mu(C_i)=\operatorname{argmin}_{\mu\in\mathcal{X}'}\sum_{x\in C_i}|x-\mu|^2$。因此，我们重写 k 均值的目标为

$$G(C_1,\cdots,C_k)=\sum_{i=1}^{k}\sum_{x\in C_i}|x-\mu(C_i)|^2 \quad (22.3)$$

考虑 k 均值算法的第 t 次迭代时的更新过程。设 $C_1^{(t-1)},\cdots,C_k^{(t-1)}$ 是前一次的划分情况，令 $\mu_i^{(t-1)}=\mu(C_i^{(t-1)})$，令 $C_1^{(t)},\cdots,C_k^{(t)}$ 为第 t 次迭代时的划分，使用公式(22.2)给出的目标函数定义，我们有

$$G(C_1^{(t)},\cdots,C_k^{(t)})\leqslant\sum_{i=1}^{k}\sum_{x\in C_i^{(t)}}|x-\mu_i^{(t-1)}|^2 \quad (22.4)$$

除此之外，新划分$(C_1^{(t)},\cdots,C_k^{(t)})$的定义意味着在所有可能的划分$(C_1,\cdots,C_k)$中，新划

270 分使 $\sum_{i=1}^{k}\sum_{x\in C_i}\|x-\mu_i^{(t-1)}\|^2$ 最小。因此，

$$\sum_{i=1}^{k}\sum_{x\in C_i^{(t)}}\|x-\mu_i^{(t-1)}\|^2\leqslant\sum_{i=1}^{k}\sum_{x\in C_i^{(t-1)}}\|x-\mu_i^{(t-1)}\|^2 \tag{22.5}$$

利用公式(22.3)，公式(22.5)等号右边部分等于 $G(C_1^{(t-1)},\cdots,C_k^{(t-1)})$。将这与公式(22.4)和公式(22.5)结合，得到 $G(C_1^{(t)},\cdots,C_k^{(t)})\leqslant G(C_1^{(t-1)},\cdots,C_k^{(t-1)})$，进而得到我们的证明。 ■

虽然上述引理告诉我们 k 均值目标是单调非增的，然而对于 k 均值算法达到收敛的迭代次数并没有给出保证。此外，算法给出的 k 均值目标函数输出值和目标函数的最小可能值之差，并没有非平凡下界。实际上，k 均值可能会收敛到局部最小值(练习 22.2)。为了提高 k 均值的结果，通常使用不同的随机初始中心点，将该程序运行多次(比如，输入数据的任意一点都可以选为初始中心点)。

22.3 谱聚类

表示数据集 $\mathcal{X}=\{x_1,\cdots,x_n\}$ 中点与点关系的常用便捷方式是相似图：每个顶点代表一个数据点 x_i，两个顶点由一条边相连，边的权重对应数据点之间的相似性，$W_{i,j}=s(x_i,x_j)$，其中 $W\in\mathbb{R}^{n\times n}$。例如，我们可以设 $W_{i,j}=\exp(-d(x_i,x_j)^2/\sigma^2)$，其中 $d(\cdot,\cdot)$ 为距离函数，σ 为参数。聚类问题现在可以表述如下：我们希望找到一种图的划分，使不同组的组间边有较低权重，使相同组的组内边有较高权重。

在前述的聚类目标中，我们给出了聚类的一种直观定义——确保同类中的点相似。我们现在给出另一种要求——在不同类中的点应当不相似。

22.3.1 图割

给定一个有相似矩阵 W 表示的图，对图进行划分的最简单和最直接方式是求解最小割问题，选取划分 $C_1,\cdots,C_k$ 使下列目标最小化

$$\mathrm{cut}(C_1,\cdots,C_k)=\sum_{i=1}^{k}\sum_{r\in C_i,s\notin C_i}W_{r,s}$$

对 $k=2$ 的情况，最小割问题可以有效地解决。但是，实际中经常不能实现满意的划分。很多情况中会出现，最小割方法简单地将单个顶点与其他顶点分离开来。当然，这不是我们想达到的聚类效果，因为聚类应当是合理地将一组点归为一类。 271

对这种问题，有多种解决方案。最简单的方式是将分割正则化，这里定义正则化后的最小割目标为：

$$\mathrm{RatioCut}(C_1,\cdots,C_k)=\sum_{i=1}^{k}\frac{1}{|C_i|}\sum_{r\in C_i,s\notin C_i}W_{r,s}$$

前面的目标假设聚类不是太小的时候取得更小的值。不幸的是，引入这项平衡让问题变得难以计算。谱聚类是一种最小比例割的松弛解法。

22.3.2 图拉普拉斯与松弛图割算法

谱聚类的主要数学对象是图拉普拉斯矩阵。在文献中有多种图拉普拉斯定义，下面我们介绍一种最流行的定义。

定义 22.2(非归一化的图拉普拉斯) 非归一化的图拉普拉斯是一个 $m\times m$ 的矩阵 $L=D-W$，其中 D 是一个对角阵，$D_{i,i}=\sum_{j=1}^{m}W_{i,j}$。矩阵 D 被称为度矩阵。

下面的引理强调了比例割和拉普拉斯矩阵之间的关系。

引理 22.3 令 $C_1,\cdots,C_k$ 为一个聚类，$H\in\mathbb{R}^{m,k}$ 为一个矩阵，则

$$H_{i,j}=\frac{1}{\sqrt{|C_j|}}\mathbb{1}_{[i\in C_j]}$$

H 矩阵的列是彼此正交的，并且

$$\mathrm{RatioCut}(C_1,\cdots,C_k)=\mathrm{trace}(H^TLH)$$

证明 令 $\boldsymbol{h}_1,\cdots,\boldsymbol{h}_k$ 为 H 的列。从定义中我们已知这些向量是彼此正交的。接下来，按照标准代数操作，可以得出 $\mathrm{trace}(H^TLH)=\sum_{i=1}^{k}\boldsymbol{h}_i^TL\boldsymbol{h}_i$。对于任意的向量 $\boldsymbol{v}$ 我们有

$$\boldsymbol{v}^TL\boldsymbol{v}=\left(\frac{1}{2}\left(\sum_r D_{r,r}v_r^2-2\sum_{r,s}v_rv_sW_{r,s}+\sum_s D_{s,s}v_s^2\right)\right)=\frac{1}{2}\sum_{r,s}W_{r,s}(v_r-v_s)^2$$

将该式与 $\boldsymbol{v}=\boldsymbol{h}_i$ 结合，注意当且仅当 $r\in C_i$，$s\notin C_i$ 时 $(h_{i,r}-h_{i,s})^2$ 不为零，反过来，我们得到

$$\boldsymbol{h}_i^TL\boldsymbol{h}_i=\frac{1}{|C_i|}\sum_{r\in C_i,s\notin C_i}W_{r,s}$$ ■

因此，为了最小化比例割，我们可以寻找一个矩阵 H，其列是正交的，$H_{i,j}$ 或为 0 或为 $1/\sqrt{|C_j|}$。不幸的是，这是一个整数规划问题，我们不能有效求解。作为替代，我们松弛后一项的要求，寻找一个正交矩阵 $H\in\mathbb{R}^{m,k}$，最小化 $\mathrm{trace}(H^TLH)$。这类问题的一种有效解决途径是令矩阵 U 的列为 L 矩阵最小的 k 个特征值对应的特征向量，如我们将要在下一章看到的 PCA 部分(特别是定理 23.2 的证明)。这种算法称作非归一化的谱聚类。 272

22.3.3 非归一化的谱聚类

非归一化的谱聚类

输入：$W\in\mathbb{R}^{m,m}$；聚类数目 k

初始化：计算非归一化的图拉普拉斯 L

令 矩阵 $U\in\mathbb{R}^{m,k}$ 的列为 L 矩阵最小的 k 个特征值对应的特征向量

令 $\boldsymbol{v}_1,\cdots,\boldsymbol{v}_m$ 为 U 的列

使用 k 均值算法对 $\boldsymbol{v}_1,\cdots,\boldsymbol{v}_m$ 聚类

输出：k 均值算法输出聚类 $C_1,\cdots,C_k$

谱聚类算法首先寻找矩阵 H，其列为图拉普拉斯矩阵最小的 k 个特征值对应的特征向量。然后将矩阵 H 的每一行作为一个数据点。根据图拉普拉斯性质，这种表示是有效的。在许多情况下，这种表示方式的改变使得即使采用简单的 k 均值算法同样能无疑地找到合理聚类。直观上讲，如果矩阵 H 按照引理 22.3 形式进行定义，在新表示中每个点是一个指示向量，那么只有元素与其所属类对应时，向量值不为零。

22.4　信息瓶颈*

信息瓶颈方法是由 Tishby，Pereira 和 Bialek 提出的聚类技术，其概念来源于信息论。为了举例说明该方法，考虑文本聚类问题，每个文本表示为一个词袋；即，每个文档都是一个向量 $x=\{0,1\}^n$，其中 n 是字典的长度，$x_i=1$ 当且仅当第 i 个词在文档中出现。给定一个有 m 个文档的集合，我们可以将 m 个文档的词袋表示理解为随机变量 x 的联合概率分布，指示文档的身份(因此在 $[m]$ 中取值)，以及一个随机变量 y，指示单词在词典中的身份(因此在 $[n]$ 中取值)。

根据这种解释，信息瓶颈是指将聚类属性表示为另一个随机变量 C，在 $[k]$ 中取值(其中 k 同样是由方法确定)。一旦将 x，y，C 表述为随机变量，我们可以使用信息论中的方法来表示聚类目标。信息瓶颈的目标是

[273]

$$\min_{p(C|x)} I(x;C)-\beta I(C;y)$$

其中 $I(\cdot,\cdot)$ 是两个随机变量的互信息[⊖]，β 是参数，在每个点分属聚类的所有可能概率分布中求取极小值。直观上讲，我们希望达到两个矛盾的目标。一方面，我们希望文档属性和聚类属性的互信息尽可能小，这反映了我们希望对原数据进行强压缩。另一方面，我们希望聚类变量和词属性的互信息尽可能大，这反映了保留文档关联信息(用词在文档中出现来表示)的目标。将参数统计中的最小充分统计量[⊜]推广到了任意分布。

解信息瓶颈准则下的最优化问题通常是非常困难的。有些解决方法类似于将要在第 24 章讨论的 EM 准则。

22.5　聚类的进阶观点

到目前为止，我们罗列了许多有用的聚类算法。然而，还有一些基本问题尚未解决。首先也是最重要的，聚类是什么？聚类算法和输入一个空间输出一个空间分布的任意函数的区别是什么？聚类有没有一些基本性质是独立于具体算法或任务的？

回答这些问题的一种方式是公理化方法。很多人尝试对聚类提出一个公理化的定义，让我们展示 Kleinberg(2003)给出的尝试方法。

考虑一个聚类函数 F，将任意有限域 $\mathcal{X}$ 及不相似函数 d 作为输入，返回 $\mathcal{X}$ 的一个划分。

考虑这类函数的三种特性：

尺度不变性[SI]　对任意的域集 $\mathcal{X}$，不相似函数 d，以及任意的 $\alpha>0$，下式成立：$F(\mathcal{X},d)=F(\mathcal{X},\alpha d)$(其中 $(\alpha d)(x,y)\overset{\text{def}}{=}\alpha d(x,y)$)。

丰富性[Ri]　对任意的有限集 $\mathcal{X}$ 和划分 $C=(C_1,\cdots,C_k)$(划分到非空子集)，存在多种不相似函数 d 使得 $F(\mathcal{X},d)=C$。

一致性[Co]　如果 d 和 d' 都是 $\mathcal{X}$ 上的不相似函数，对任一 $x,y\in\mathcal{X}$，根据 $F(\mathcal{X},d)$，如果 x，y 属于同一类，则 $d'(x,y)\leqslant d(x,y)$，x，y 属于不同类，则 $d'(x,y)\geqslant d(x,y)$，

[274]

那么 $F(\mathcal{X},d)=F(\mathcal{X},d')$。

⊖ 给定 (x,C) 上的概率函数 p，$I(x;C)=\sum_c\sum_b p(a,b)\log\left(\frac{p(a,b)}{p(a)p(b)}\right)$，其中求和部分从对所有可能的 x 和 C。

⊜ 充分统计量是关于输入数据的一个函数，充分性是对统计模型及相关的未知参数而言，表示“没有其他的统计量可以提供样本和参数的额外信息”。例如，如果我们假定一个变量是正态分布，方差为单位方差，期望未知，则平均值函数是一个充分统计量。

尺度不变性是一种非常自然的要求——如果聚类函数输出的结果依赖于测量点之间的距离测度单元，那将显得十分奇怪。丰富性要求主要想说明聚类函数的输出是由函数 d 全权决定，也是一种非常直观的特征。一致性要求是和聚类基本（非正式）定义相关的要求——我们希望相似的点聚到一类，不相似的点分属不同类，因此共享同类的点更相似，已经分离的点不相似，聚类函数应当对之前的聚类决策有很强的"支撑"作用。

然而，Kleinberg(2003)已经给出了下述"不可能"结论：

定理 22.4 不存在一个函数 F 同时满足上述三种属性：尺度不变性，丰富性，一致性。

证明 根据反证法，假设存在函数 F 满足上述三种属性。取一个至少有三个点的域集 $\mathcal{X}$。根据丰富性，肯定存在 d_1 使得 $F(\mathcal{X}, d_1)=\{\{x\}: x\in\mathcal{X}\}$，存在 d_2 使得 $F(\mathcal{X}, d_2)\neq F(\mathcal{X}, d_1)$。

令 $\alpha\in\mathbb{R}_+$，则对任一 $x, y\in\mathcal{X}$ 有 $\alpha d_2(x, y)\geqslant d_1(x, y)$。令 $d_3=\alpha d_2$，考虑 $F(\mathcal{X}, d_3)$。根据函数 F 的尺度不变属性，我们有 $F(\mathcal{X}, d_3)=F(\mathcal{X}, d_2)$。另一方面，因为所有不同的 $x, y\in\mathcal{X}$ 分属不同类，关于 $F(\mathcal{X}, d_1)$ 和 $d_3(x, y)\geqslant d_1(x, y)$，函数 F 的一致性属性表明 $F(\mathcal{X}, d_3)=F(\mathcal{X}, d_1)$。这里产生矛盾，因为我们选取的 d_1 和 d_2 使 $F(\mathcal{X}, d_2)\neq F(\mathcal{X}, d_1)$。 ■

要注意的是，在这三条属性中没有"坏公理"和"坏属性"。对于三条属性中的每一对，存在自然的聚类函数满足这对属性（对于单链接聚类函数，读者仅仅通过设定不同的终止准则就可以构建这样的例子）。另一方面，Kleinberg 给出结论，对于最小化基于中心点的目标函数，任意的聚类算法都不可避免地违背了一致性属性（然而，k-sum-of-in-cluster-distances 最小化聚类确实满足一致性）。

Kleinberg 的"不可能"结论可以通过改变属性来规避。例如，如果讨论含固定数量参数的聚类函数，很自然地将丰富性改为 k-丰富性（即，将域划分到 k 个子集是可以实现的），k 均值聚类满足 k-丰富性、尺度不变性和一致性，因此能够达到一致。或者可以放松一致性属性。例如，如果对任一类 $C_i\in C$ 和 $C'_j\in C'$，有 $C_i\subseteq C'_j$ 或者 $C'_j\subseteq C_i$ 或者 $C_i\cap C'_j=\varnothing$，我们就说两个聚类 $C=(C_1, \cdots, C_k)$ 和 $C'=(C'_1, \cdots, C'_l)$ 是兼容的（这是值得做的，因为对每个系统树图，根据剪边得到的两个聚类，其系统树图是兼容的）。"精致一致性"是要求，在一致性属性的假设下，新聚类 $F(\mathcal{X}, d')$ 和旧聚类 $F(\mathcal{X}, d)$ 是兼容的。许多聚类函数满足这项要求的同时也满足尺度不变性和丰富性。进一步，可以提出许多其他不同的、很直观的、令人满意的并且已知一些聚类函数满足的聚类函数属性。 [275]

解释这些结果的方法有很多。我们建议将其视为没有"理想的"聚类函数。每个聚类函数都不可避免地有一些"不良的"属性。给定一项任务，聚类函数的选取必须考虑该任务的特定属性。没有统一的聚类解决方案，就像没有一种分类算法能够对每一项可学习任务都能学习（就如"没有免费午餐"定理所示）。和其他分类预测一样，聚类必须考虑特定任务的先验知识。

22.6 小结

聚类是一个无监督学习问题，希望将点集划分到"有意义的"多个子集。我们给出了几种聚类手段，包括基于链接的算法、k 均值家族、谱聚类和信息瓶颈。我们讨论了将聚类的直观含义进行形式化表示的困难。

22.7 文献评注

k 均值算法有时也会被称为 Lloyd 算法，根据 Stuart Lloyd 命名，他于 1957 年提出了该算法。为了更全面地了解谱聚类相关知识，我们建议读者去阅读 Von Luxburg(2007)的优秀教材。信息瓶颈方法是由 Tishby，Pereira 和 Bialek(1999)提出。公理化方法可以参考 Ackerman 和 Ben-David(2008)的工作。

22.8 练习

22.1 **k 均值算法的次优性**：对任一参数 $t>1$，证明：存在 k 均值问题的一个实例，k 均值算法(可能)找到一种解决方案，使其 k 均值目标至少为 $t\cdot\mathrm{OPT}$，其中 OPT 为 k 均值目标的最小值。

22.2 **k 均值算法不一定收敛于局部极小值**：证明：k 均值算法可能收敛于某点，而该点并不是局部极小值。

提示：假定 $k=2$，样本点为 $\{1,2,3,4\}\subset\mathbb{R}$，并假定初始化 k 均值中心点为 $\{2,4\}$；同时改变 C_i 定义中的赋值关系，将 i 分配给 $\mathrm{argmin}_i|x-\mu_i|$ 的最小值。

22.3 给定一测度空间 $(\mathcal{X},d)$，其中 $|\mathcal{X}|<\infty$，$k\in\mathbb{N}$，寻找一种划分方式将 $\mathcal{X}$ 划归到 $C_1,\cdots,C_k$，使下述表达式取得极小值

$$G_{k\text{-diam}}((\mathcal{X},d),(C_1,\cdots,C_k))=\max_{j\in[k]}\mathrm{diam}(C_j)$$

其中 $\mathrm{diam}(C_j)=\max_{x,x'\in C_j}d(x,x')$(这里约定，如果 $|C_j|<2$，则 $\mathrm{diam}(C_j)=0$)。

类似于 k 均值算法的目标函数，最小化 k-diam 目标是 NP 难问题。幸运的是，我们有一种简单的近似算法：开始选取某个 $x\in\mathcal{X}$，并令 $\mu_1=x$。接下来，算法进行迭代，令

[276]

$$\forall j\in\{2,\cdots,k\},\mu_j=\mathrm{argmax}_{x\in\mathcal{X}}\min_{i\in[j-1]}d(x,\mu_i)$$

最后，令

$$\forall i\in[k],C_i=\{x\in\mathcal{X}:i=\mathrm{argmin}_{j\in[k]}d(x,\mu_j)\}$$

证明：刚才描述的算法是 2-近似算法。即，如果我们将输出结果用 $\hat{C}_1,\cdots,\hat{C}_k$ 表示，最优解决方案用 $C_1^*,\cdots,C_k^*$ 表示，那么

$$G_{k\text{-diam}}((\mathcal{X},d),(\hat{C}_1,\cdots,\hat{C}_k))\leq 2G_{k\text{-diam}}((\mathcal{X},d),(C_1^*,\cdots,C_k^*))$$

提示：考虑点 μ_{k+1}(换言之，如果想要 $k+1$ 个聚类，我们将要选取的下一个中心点)。令 $r=\min_{i\in[k]}d(\mu_i,\mu_{k+1})$，证明下列不等式

$$G_{k\text{-diam}}((\mathcal{X},d),(\hat{C}_1,\cdots,\hat{C}_k))\leq 2r,\quad G_{k\text{-diam}}((\mathcal{X},d),(C_1^*,\cdots,C_k^*))\geq r$$

22.4 对于某个单调函数 $f:\mathbb{R}_+\to\mathbb{R}_+$，在每个给定的输入 $(\mathcal{X},d)$，$F(\mathcal{X},d)$ 通过最小化目标函数

$$G_f((\mathcal{X},d),(C_1,\cdots,C_k))=\min_{\mu_1,\cdots,\mu_k\in\mathcal{X}'}\sum_{i=1}^{k}\sum_{x\in C_i}f(d(x,\mu_i))$$

进行聚类，其中 $\mathcal{X}'$ 为 $\mathcal{X}$ 或 $\mathcal{X}$ 的超集。我们将这类聚类算法称为基于中心的聚类。

证明：对任一 $k>1$，上一题中的 k-diam 算法不属于基于中心的聚类算法。

提示：给定一个聚类输入 $(\mathcal{X},d)$，其中 $|\mathcal{X}|>2$，在 $\mathcal{X}$ 中的某些(不是所有)成员添加一些临近点，考虑该情况对 k-diam 算法和基于中心的聚类算法的作用是否相同。

22.5 我们讨论了聚类的三种“特性”：尺度不变性、丰富性和一致性，考虑单链接聚类

算法。

1) 对于终止准则为达到固定聚类数 Π（任意固定非零数字）的单链接聚类算法，找出该类算法满足三条特征中的哪条特性。

2) 对于终止准则为类内距离达到上限（任意固定非零上限）的单链接聚类算法，找出该类算法满足三条特征中的哪条特性。

3) 证明对于三条特性中的任意一对，都存在一种终止准则，使单链接聚类算法满足这两种特性。

22.6 给定某个数 k，令 k-丰富性满足如下条件：

对任意的有限集 $\mathcal{X}$ 和 $\mathcal{X}$ 的任意一种划分 $C=(C_1, \cdots, C_k)$（划分为非空子集），存在一些 X 上的距离函数 d 使 $F(\mathcal{X}, d)=C$。

证明，对于任意一个 k，存在聚类函数满足三条特性：尺度不变性、k-丰富性和一致性。

维度约简

降维是将高维数据映射到低维空间的过程。该过程与信息论中的(有损)压缩概念密切相关。降维的原因通常有以下几个。首先，高维数据增加了运算的难度。其次，高维使得学习算法的泛化能力变弱(例如，在最近邻分类器中，样本复杂度随着维度成指数增长，参考第 19 章)。最后，降维能够增加数据的可读性，利用发掘数据的有意义的结构。

在本章，我们介绍了一些比较流行的降维方法。在这些方法中，降维是通过对原始数据的线性变换实现的。即，如果数据是 d 维的，我们想将其约简到 n 维($n<d$)，则需要找到一个矩阵 $W\in\mathbb{R}^{n,d}$ 使得映射 $x\mapsto Wx$。选择 W 的一个最自然的准则是在降维的同时能够复原原始的数据 x。通常这是比较困难的，而且从 Wx 中准确复原 x 是不可能的(见练习 23.1)。

第一种方法称之为主成分分析(PCA)。在 PCA 中，降维和复原都是通过线性变换实现，而且复原的信号与原始的信号保持均方差最小。

接下来，我们介绍如何利用随机矩阵进行降维。我们推导出一个重要的引理，称为"Johnson-Lindenstrauss 引理"，该引理分析了随机降维技术的失真情况。

最后，我们介绍如何利用随机矩阵对稀疏向量进行降维。该过程被称为压缩感知。在这种情况下，复原是非线性的，但可通过线性规划有效实现。

[278]

在小结部分，我们指出 PCA 和压缩感知背后的先验假设，这有利于我们理解两种方法的优缺点。

23.1 主成分分析

令 $x_1,\ldots,x_m$ 为 m 个 d 维向量。我们想利用线性变换对这些向量进行降维。给定矩阵 $W\in\mathbb{R}^{n,d}$($n<d$)，则存在映射 $x\mapsto Wx$，其中 $Wx\in\mathbb{R}^n$ 是 x 的低维表示。另外，矩阵 $U\in\mathbb{R}^{d,n}$ 能够将压缩后的信号(近似)复原为原始的信号。即，对于压缩向量 $y=Wx$，其中 y 在低维空间 $\mathbb{R}^n$ 中，我们能够构建 $\tilde{x}=Uy$，使得 $\tilde{x}$ 是 x 的复原版本，处在原始的高维空间 $\mathbb{R}^d$ 中。

在 PCA 中，我们要找的压缩矩阵 W 和复原矩阵 U 使得原始信号和复原信号在平方距离上最小：即，我们需要求解如下问题

$$\underset{W\in\mathbb{R}^{n,d},U\in\mathbb{R}^{d,n}}{\operatorname{argmin}}\sum_{i=1}^{m}\|x_i-UWx_i\|_2^2 \tag{23.1}$$

为了求解上述问题，我们首先显示该最优解具有特别的形式。

引理 23.1 令(U,W)是式(23.1)的一个解，则 U 的列是单位正交的(即，U^TU 是 $\mathbb{R}^n$ 上的单位矩阵)以及 $W=U^T$。

证明 给定任何的 U，W，考虑映射 $x\mapsto UWx$。该映射的值域 $R=\{UWx:x\in\mathbb{R}^d\}$ 是 $\mathbb{R}^d$ 中的一个 n 维线性子空间。令 $V\in\mathbb{R}^{d,n}$ 为一个单位正交矩阵，且它的列构成了上述子空间的一组正交基，即，V 的值域为 R 且 $V^TV=I$。因此，R 中的每一个列向量可表示为 Vy，其中 $y\in\mathbb{R}^n$。对于每一个 $x\in\mathbb{R}^d$ 和 $y\in\mathbb{R}^n$，我们有

$$\|x-Vy\|_2^2=\|x\|^2+y^TV^TVy-2y^TV^Tx=\|x\|^2+\|y\|^2-2y^T(V^Tx)$$

其中，我们利用了 $V^{\mathrm{T}}V$ 是 $\mathbb{R}^n$ 中的单位矩阵。对上式关于 y 求最小，即使关于 y 的梯度为 0，有 $y=V^{\mathrm{T}}x$。因此，对于每一个 x 我们有

$$VV^{\mathrm{T}}x=\underset{\tilde{x}\in R}{\operatorname{argmin}}\|x-\tilde{x}\|_2^2$$

特别地，上式对于 x_1，…，x_m 均成立。因此，我们用 V，V^{T} 来取代 U，W，由此产生如下不等关系

$$\sum_{i=1}^{m}\|x_i-UWx_i\|_2^2\geqslant\sum_{i=1}^{m}\|x_i-VV^{\mathrm{T}}x_i\|_2^2$$

因为对于每一个 U，W，上式都成立，所以引理成立。 ■ 279

在上述引理的基础上，我们可以重写优化问题式(23.1)为

$$\underset{U\in\mathbb{R}^{d\times n},U^{\mathrm{T}}U=I}{\operatorname{argmin}}\sum_{i=1}^{m}\|x_i-UU^{\mathrm{T}}x_i\|_2^2 \tag{23.2}$$

我们利用下面基本的代数操作来进一步简化该优化问题。对于每一个 $x\in\mathbb{R}^d$ 和矩阵 $U\in\mathbb{R}^{d\times n}$ 且 $U^{\mathrm{T}}U=I$，我们有

$$\begin{aligned}\|x-UU^{\mathrm{T}}x\|^2&=\|x\|^2-2x^{\mathrm{T}}UU^{\mathrm{T}}x+x^{\mathrm{T}}UU^{\mathrm{T}}UU^{\mathrm{T}}x\\&=\|x\|^2-x^{\mathrm{T}}UU^{\mathrm{T}}x\\&=\|x\|^2-\operatorname{trace}(U^{\mathrm{T}}xx^{\mathrm{T}}U)\end{aligned} \tag{23.3}$$

其中，矩阵的迹(trace)为矩阵的对角元素之和。因为迹是一个线性算子，这允许我们将式(23.2)重写为

$$\underset{U\in\mathbb{R}^{d\times n},U^{\mathrm{T}}U=I}{\operatorname{argmin}}\operatorname{trace}\left(U^{\mathrm{T}}\sum_{i=1}^{m}x_ix_i^{\mathrm{T}}U\right) \tag{23.4}$$

令 $A=\sum_{i=1}^{m}x_ix_i^{\mathrm{T}}$。矩阵 A 是对称的且可进行谱分解 $A=VDV^{\mathrm{T}}$，其中 D 是对角矩阵以及 $V^{\mathrm{T}}V=VV^{\mathrm{T}}=I$。这里，$D$ 的对角元素是 A 的特征值，V 的列对应 A 的特征向量。不失一般性，我们假设 $D_{1,1}\geqslant D_{2,2}\geqslant\cdots\geqslant D_{d,d}$。因为 A 是半正定的，所以 $D_{d,d}\geqslant 0$。我们得出式(23.4)的解为 A 的最大的 n 个特征值对应的特征向量所构成的矩阵 U。

定理 23.2 令 x_1，…，x_m 是 $\mathbb{R}^d$ 中的任意向量，$A=\sum_{i=1}^{m}x_ix_i^{\mathrm{T}}$，以及 u_1，…，u_n 是 A 中最大的 n 个特征值对应的特征向量。那么，如式(23.1)所示的 PCA 优化问题的解为：令 U 的列等于 u_1，…，u_n 以及 $W=U^{\mathrm{T}}$。

证明 令 VDV^{T} 为 A 的谱分解。给定列正交矩阵 $U\in\mathbb{R}^{d\times n}$，令 $B=V^{\mathrm{T}}U$。则，$VB=VV^{\mathrm{T}}U=U$。进而有

$$U^{\mathrm{T}}AU=B^{\mathrm{T}}V^{\mathrm{T}}VDV^{\mathrm{T}}VB=B^{\mathrm{T}}DB$$

以及

$$\operatorname{trace}(U^{\mathrm{T}}AU)=\sum_{j=1}^{d}D_{j,j}\sum_{i=1}^{n}B_{j,i}^2$$

注意 $B^{\mathrm{T}}B=U^{\mathrm{T}}VV^{\mathrm{T}}U=U^{\mathrm{T}}U=I$。因此，$B$ 也是列正交的，这意味着 $\sum_{j=1}^{d}\sum_{i=1}^{n}B_{j,i}^2=n$。另外，令 $\tilde{B}\in\mathbb{R}^{d\times d}$ 为一个前 n 列与 B 相同并且 $\tilde{B}^{\mathrm{T}}\tilde{B}=I$ 的矩阵。则对于每一个 j，我们有 $\sum_{i=1}^{d}\tilde{B}_{j,i}^2=1$，这意味着 $\sum_{i=1}^{n}B_{j,i}^2\leqslant 1$。进而 280

$$\operatorname{trace}(U^{\mathrm{T}}AU) \leqslant \max_{\|\boldsymbol{\alpha}\|_2=1,\ \boldsymbol{\alpha}\in\mathbb{R}^d} \sum_{j=1}^{d} D_{j,j}\alpha_j^2$$

不难验证(见 23.2)上式右边等于 $\sum_{j=1}^{n} D_{j,j}$。因此对于每一个列正交矩阵 $U\in\mathbb{R}^{d\times n}$，$\operatorname{trace}(U^{\mathrm{T}}AU)\leqslant\sum_{j=1}^{n} D_{j,j}$ 均成立。在另一方面，如果我们令 U 的列为 A 的前 n 个特征向量，我们有 $\operatorname{trace}(U^{\mathrm{T}}AU)=\sum_{j=1}^{n} D_{j,j}$，由此定理得证。 ■

评注　定理 23.2 的证明同样告诉我们式(23.4)的目标值是 $\sum_{j=1}^{n} D_{j,j}$。这与式(23.3)结合以及 $\sum_{i=1}^{m}\|\boldsymbol{x}_i\|^2=\operatorname{trace}(A)=\sum_{j=1}^{d} D_{j,j}$，我们可以得到式(23.1)的最优值是 $\sum_{j=n+1}^{d} D_{j,j}$。

评注　在实际应用中，在运用 PCA 之前需要对样本进行"中心化"。即，我们首先计算 $\boldsymbol{\mu}=\frac{1}{m}\sum_{i=1}^{m}\boldsymbol{x}_i$，然后用 PCA 作用于 $(\boldsymbol{x}_1-\boldsymbol{\mu}),\ldots,(\boldsymbol{x}_m-\boldsymbol{\mu})$。这也与 PCA 作为方差最大化的解释相关(见练习 23.4)。

23.1.1　当 $d\gg m$ 时一种更加有效的求解方法

在一些情况下，数据的原始维度远远大于样本的个数 m。按照如前所述的方法求解 PCA 的计算复杂度是 $O(d^3)$(用于计算 A 的特征值)再加上 $O(md^2)$(用于构建矩阵 A)。我们现在介绍在 $d\gg m$ 的情形下更加有效求解 PCA 的一个简单技巧。

回忆一下 A 可以被描述为 $\sum_{i=1}^{m}\boldsymbol{x}_i\boldsymbol{x}_i^{\mathrm{T}}$。因此，重写为 $A=X^{\mathrm{T}}X$，其中 $X\in\mathbb{R}^{m\times d}$ 的第 i 行为 $\boldsymbol{x}_i^{\mathrm{T}}$。考虑矩阵 $B=XX^{\mathrm{T}}$，则 $B\in\mathbb{R}^{m\times m}$ 的第 i,j 个元素为 $\langle\boldsymbol{x}_i,\boldsymbol{x}_j\rangle$。假设 $\boldsymbol{u}$ 是 B 的一个特征向量，有 $\lambda\in\mathbb{R}$，$B\boldsymbol{u}=\lambda\boldsymbol{u}$。等式两边左乘 X^{T} 并利用 B 的定义可得 $X^{\mathrm{T}}XX^{\mathrm{T}}\boldsymbol{u}=\lambda X^{\mathrm{T}}\boldsymbol{u}$。但是，利用 A 的定义，有 $A(X^{\mathrm{T}}\boldsymbol{u})=\lambda(X^{\mathrm{T}}\boldsymbol{u})$。因此，$\frac{X^{\mathrm{T}}\boldsymbol{u}}{\|X^{\mathrm{T}}\boldsymbol{u}\|}$ 是 A 的一个特征向量，对应的特征值为 λ。

所以，我们可以通过计算 B 的特征值来取代 A 去求解 PCA。该过程的复杂度是 $O(m^3)$(用于计算 B 的特征值)以及 $O(m^2d)$(用于构建 A)。

评注　之前的讨论同样意味着我们仅仅只需要知道如何去计算向量的内积去求解 PCA。这使得我们在 d 非常大(甚至无限)的时候能够利用核去隐性地求解 PCA，从而产生了核 PCA 算法。

23.1.2　应用与说明

[281]　下面给出了 PCA 的一个伪代码。

PCA

输入

包含 m 条数据样本的矩阵 $X\in\mathbb{R}^{m\times d}$

成分个数 n

如果($m>d$)

$A=X^{\mathrm{T}}X$

令 $u_1, \cdots, u_n$ 为 A 的前 n 个最大特征值对应的特征向量

否则

$B=XX^{\mathrm{T}}$

令 $v_1, \cdots, v_n$ 为 B 的前 n 个最大特征值对应的特征向量

对 $i=1, \cdots, n$，令 $u_i=\frac{1}{\|X^{\mathrm{T}}v_i\|}X^{\mathrm{T}}v_i$

输出：$u_1, \cdots, u_n$

为了说明 PCA 是如何运作的，我们首先生成一些散落在一条直线附近的二维向量，即，处于二维空间的一维子空间。例如，每一个样本具有$(x, x+y)$的形式，其中 x 从$[-1, 1]$中均匀随机选取，而 y 从一个均值为 0、标准方差为 0.1 的高斯分布中随机采样。我们现在将 PCA 应用于这些数据。那么，对应着最大特征值的特征向量近似于向量$(1/\sqrt{2}, 1/\sqrt{2})$。将点$(x, x+y)$投影到该主成分上，我们将得到标量$\frac{2x+y}{\sqrt{2}}$。原始向量的重构为$(x+y/2, x+y/2)$。在图 23.1 中，我们画出了原始和重构的数据。

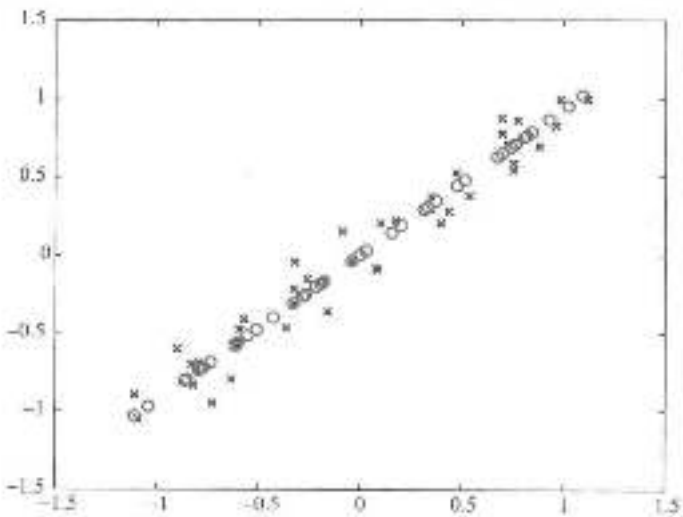


图 23.1 二维空间的一个向量集(×)以及利用 PCA 降维到一维后的重构结果(○)

[282]

接下来，我们阐述 PCA 在一个人脸数据集上的有效性。我们从 Yale 数据集(Georghiades, Belhumeur & Kriegman 2001)选取部分人脸图像。每幅图像有 $50\times50=2500$ 像素；因此原始的维度非常高。

一些人脸图像展示在图 23.2 的左上部分。利用 PCA，我们约简维度到 $\mathbb{R}^{10}$，然后重构回原始的维度($\mathbb{R}^{50\times50}$)。重构的结果展示在图 23.2 的右上部分。最后，在图 23.2 的底部，我们画出了图像的二维表达。可以看到，即使从图像的二维表达中我们仍然能够粗略地分离不同的个体。

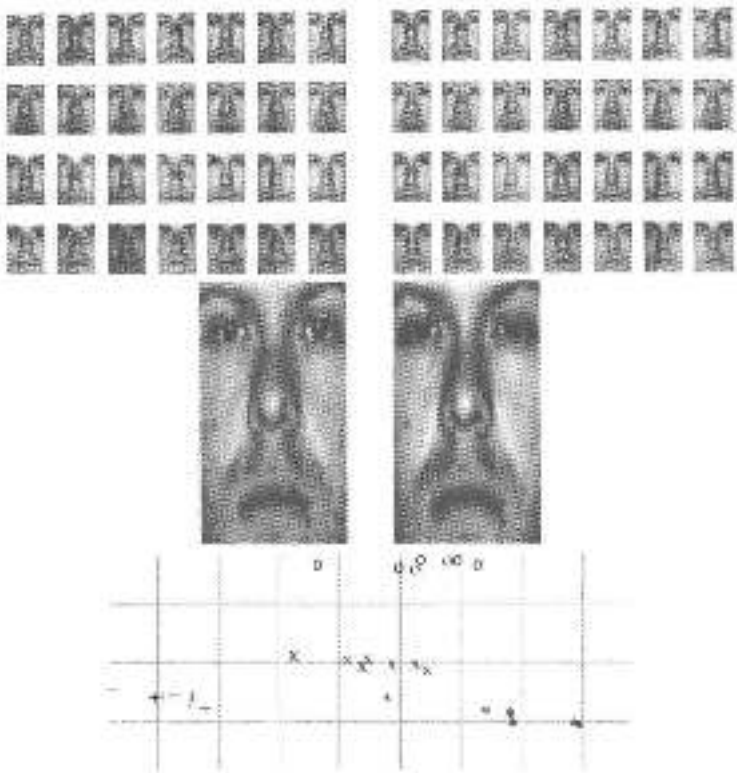


图 23.2　从 Yale 数据集中选取的人脸图像。左上：原始图像（$\mathbb{R}^{n'}$）。右上：降维到 $\mathbb{R}^n$ 后重构的图像。中间一行：PCA 处理前后的一幅图像的放大版本。底部：降维到 $\mathbb{R}^n$ 后的图像。不同的标记代表不同的个人。

23.2　随机投影

本节我们介绍利用随机线性投影进行降维，这导致了一种具有很低失真的压缩策略。变换 $x \mapsto Wx$，其中 W 是一个随机矩阵，因此又称之为随机投影。我们通过提供一个由 Johnson 和 Lindenstrauss 给出的著名引理的变种，从而显示随机投影不会扭曲欧氏距离太多。

[283]

令 x_1，x_2 为 $\mathbb{R}^n$ 上的两个向量，如果

$$\frac{|Wx_1 - Wx_2|}{|x_1 - x_2|}$$

接近 1，则矩阵 W 没有扭曲 x_1 和 x_2 之间的距离太多，或具有保距特性。换句话说，x_1 和 x_2 之间的距离在变换前后基本一致。为了显示 $|Wx_1 - Wx_2|$ 没有太过偏离 $|x_1 - x_2|$，只要证明 W 没有扭曲差分向量 $x = x_1 - x_2$ 的范数，因此，从现在开始我们关注 $\frac{|Wx|}{|x|}$。

我们首先分析由于应用随机投影产生的失真。

引理 23.3　固定某个 $x \in \mathbb{R}^n$。令 $W \in \mathbb{R}^{n' \times n}$ 为随机矩阵使得每个 W_{ij} 是独立正态随机变量。那么，对于 $\varepsilon \in (0, 3)$ 我们有

$$\Pr\left[\left|\frac{|(1/\sqrt{n}Wx)|^2}{|x|^2} - 1\right| > \varepsilon\right] \leq 2e^{-\varepsilon^2 n/6}$$

证明 不失一般性，我们假设$\|x\|^2=1$。因此有等价不等式

$$\mathbb{P}[(1-\varepsilon)n \leqslant \|Wx\|^2 \leqslant (1+\varepsilon)n] \geqslant 1-2e^{-\varepsilon^2 n/6}$$

令w_i为W的第i行。随机变量$\langle w_i, x\rangle$是d个独立正态随机变量的一个加权和，因而均值为0，方差为$\sum_j x_j^2=\|x\|^2=1$。因此，随机变量$\|Wx\|^2=\sum_{i=1}^{n}(\langle w_i, x\rangle)^2$具有一个$\chi^2$分布。最终的结论可由B.7节中的引理B.12所述的χ^2随机变量的测量集的特性直接得出。 ■

Johnson-Lindenstrauss引理可用一个简单的联合有界声明得出。

引理23.4（Johnson-Lindenstrauss引理） 令Q为$\mathbb{R}^d$上的一个有限向量集合。另令$\delta\in(0,1)$且n为一个整数，使得

$$\varepsilon=\sqrt{\frac{6\log(2|Q|/\delta)}{n}}\leqslant 3$$

那么，选择一个随机矩阵$W\in\mathbb{R}^{n\times d}$，$W$中的每一个元素满足零均值和$1/n$方差，则有接近$1-\delta$的概率

$$\sup_{x\in Q}\left|\frac{\|Wx\|^2}{\|x\|^2}-1\right|<\varepsilon$$

证明 结合引理23.3和联合界，对于每一个$\varepsilon\in(0,3)$我们有 284

$$\mathbb{P}\left[\sup_{x\in Q}\left|\frac{\|Wx\|^2}{\|x\|^2}-1\right|>\varepsilon\right]\leqslant 2|Q|e^{-\varepsilon^2 n/6}$$

令δ表示上面不等式的右边；因此我们可以得到

$$\varepsilon=\sqrt{\frac{6\log(2|Q|/\delta)}{n}}$$ ■

有趣的是，引理23.4中的界不依赖于x的原始维度。事实上，即使x在无限维的希尔伯特空间上，该界依然有效。

23.3 压缩感知

压缩感知利用原始信号在某一个基上表示稀疏这一先验假设进行降维。考虑向量$x\in\mathbb{R}^d$，它具有至多s个非零元素。即，

$$\|x\|_0\stackrel{\text{def}}{=}|\{i:x_i\neq 0\}|\leqslant s$$

显然，我们可以通过利用s个（索引，值）对表示x从而对x进行压缩。而且，这种压缩是无损的——我们可以从这s个（索引，值）对中准确重构x。现在，进一步假设$x=U\alpha$，其中α是一个稀疏向量，$\|\alpha\|_0\leqslant s$，以及U是一个确定的正交矩阵。即，x在另一个基上具有稀疏表达。事实上，很多自然信号在某一表达上是稀疏或近似稀疏的。该假设应用在很多现代压缩方法中，例如，图像压缩中的JPEG 2000格式便是基于自然图像在小波基上的近似稀疏性。

我们怎么将x压缩成s个元素呢？一种简单的方法是将x乘以U^T，则得到稀疏向量α，然后利用s个（索引，值）对表示α。然而，这首先需要我们去"感知"x，然后存储它，进而才能乘以U^T。这产生了一个非常自然的问题：既然需要压缩的信号的大部分内容是要抛弃的，我们为什么要花费代价去获得所有的数据呢？我们能不能直接获取那些最终不被抛弃的信号内容呢？

压缩感知是一种同时获取和压缩数据的技术。关键结果是一个随机线性变换可以在不损失信息的前提下对 x 进行压缩。需要测量的数目是 $s\log(d)$ 阶的，即我们仅仅只需要信号的这些重要信号。在后面，我们可以看到，我们需要付出的代价是一个比较复杂的重建阶段。在一些情况下，在压缩阶段节约时间而在重构阶段花费更多的时间是有意义的。例如，一个安全监控摄像头应该感知和压缩大量的图像，而大部分时间我们是不需要对这些压缩数据进行解码的。而且在很多实际应用中，利用线性变换进行压缩的优势在于可以在硬件上表现得高效。例如，Baraniuk 和 Kelly 带领的团队已经提出了一个摄像头结构，该结构利用一个数字微镜阵列进行图像线性变换的光计算。在这种情况下，获取每一个压缩测量与获取单个原测量一样简单。压缩感知的另一个重要应用是医学成像，在该领域，需要更少的测量从而对病人产生更少的辐射。

[285]

非正式地，压缩感知的主要前提来自于如下三个"惊人"的结果：

1. 如果一个信号通过 $x\mapsto Wx$ 压缩，其中 W 是一个满足约束等距特性(RIP)的矩阵，满足该特性的矩阵能够保证任一稀疏表达向量范数的低失真。

2. 通过求解一个线性规划问题，重构可在多项式时间内完成。

3. 当 n 大于 $s\log(d)$ 阶次时，一个随机的 $n\times d$ 矩阵很有可能满足 RIP 条件。

正式地有如下定义：

定义 23.5(RIP) 如果对所有的 $x\neq 0$ 且 $\|x\|_0\leq s$ 有下式成立，那么一个矩阵 $W\in\mathbb{R}^{n\times d}$ 是 (ϵ,s)-RIP 的：

$$\left|\frac{\|Wx\|_2^2}{\|x\|_2^2}-1\right|\leq\epsilon$$

第一个定理说明了 RIP 矩阵能够产生一个稀疏向量的无损压缩策略。它同样提供了一个(不高效的)重构策略。

定理 23.6 令 $\epsilon<1$ 以及 W 为 $(\epsilon,2s)$-RIP 矩阵，x 是一个满足 $\|x\|_0\leq s$ 的向量，$y=Wx$ 是 x 的压缩，并令

$$\bar{x}\in\underset{v:Wv=y}{\operatorname{argmin}}\|v\|_0$$

为重构向量，则 $\bar{x}=x$。

证明 利用反证法，我们假设 $\bar{x}\neq x$，因为 $\|\bar{x}\|_0\leq\|x\|_0\leq s$，所以 $\|x-\bar{x}\|_0\leq 2s$。我们将 RIP 不等式应用到 $x-\bar{x}$。但是，因为 $W(x-\bar{x})=0$，我们有 $|0-1|\leq\epsilon$，这与假设矛盾。 ■

定理 23.6 给出的重建方法似乎依旧低效，因为我们需要最小化一个组合目标(v 的稀疏性导致)。神奇的是，我们可以将组合目标 $\|v\|_0$ 替换为凸的目标 $\|v\|_1$，从而转化为线性规划问题并能高效求解。以下定理对此做出正式阐述。

定理 23.7 假设定理 23.6 的条件成立以及 $\epsilon<\frac{1}{1+\sqrt{2}}$，则，

[286]

$$x=\underset{v:Wv=y}{\operatorname{argmin}}\|v\|_0=\underset{v:Wv=y}{\operatorname{argmin}}\|v\|_1$$

事实上，我们将证明一个更强的结果，该结果即使在 x 不是稀疏向量的条件下依然成立。

定理 23.8 令 $\epsilon<\frac{1}{1+\sqrt{2}}$ 以及 W 是 $(\epsilon,2s)$-RIP 矩阵。令 x 为任意向量并定义

$$x_s \in \underset{v:\|v\|_0 \le s}{\operatorname{argmin}} \|x - v\|_1$$

即，x_s 是一个 s 个最大元素与 x 相等且其他元素为 0 的向量。令 $y = Wx$ 为 x 的压缩以及

$$x^* \in \underset{v:Wv = y}{\operatorname{argmin}} \|v\|_1$$

为重构向量。则有，

$$\|x^* - x\|_2 \le 2\frac{1-\rho_s^{-1/2}}{1-\rho}\|x - x_s\|_1$$

其中 $\rho = \sqrt{2}\varepsilon/(1-\varepsilon)$。

注意在 $x = x_s$ 这种情况下，我们能够获得一个准确复原，$x^* = x$，所以定理 23.7 是定理 23.8 的个例。定理 23.8 的证明在 23.3.1 节。

最后，第三个结果告诉我们，$n \ge \Omega(s\log(d))$ 的随机矩阵很有可能是 RIP 的。事实上，这个理论显示一个随机矩阵乘以一个正交矩阵同样是一个 RIP 矩阵。这对压缩信号 $x = U\alpha$（x 不稀疏，α 稀疏）很重要。在这种情况下，如果 W 是随机矩阵且用 $y = Wx$ 压缩，这类似用 $y = (WU)\alpha$ 压缩 α，而且由于 WU 是 RIP 的，所以我们能够从 y 中重构 α（以及 x）。

定理 23.9 令 U 是一个任意的 $d \times d$ 正交矩阵，ε 和 δ 是在 $(0, 1)$ 的标量，s 是在 $[d]$ 上的整数，n 是满足下列条件的整数

$$n \ge 100\frac{s\log(40d/(\delta\varepsilon))}{\varepsilon^2}$$

令 $W \in \mathbb{R}^{n,d}$ 的每一个元素满足均值为 0，方差为 $1/n$ 的正态分布。那么，当选择好 W 后，矩阵 WU 以至少 $1-\delta$ 的概率是 (ε, s)-RIP 的。

以下证明内容属于高级部分。

定理 23.8 的证明

我们从 Candès(2008)的工作中得出以下证明。

令 $h = x^* - x$，给定一个向量 v 和一个索引集合 I，我们用 v_I 表示一个向量，如果 $i \in I$，则该向量的第 i 个元素是 v_i，否则为 0。

我们使用的第一个技巧是将索引集合 $[d] = \{1, \cdots, d\}$ 划分为大小为 s 的不相交的集合。即，对于所有的 i，有 $[d] = T_0 \cup T_1 \cup T_2 \cdots T_{d/s-1}$，$|T_i| = s$。为了证明的简易性，我们假设 d/s 是整数，我们规定：在 T_0 中，包含 x 的绝对值中的最大的 s 个元素对应的 s 个的索引。令 $T_0^c = [d] \setminus T_0$。接下来，$T_1$ 包含 $h_{T_0^c}$ 的绝对值值中最大的 s 个元素所对应的索引。令 $T_{0,1} = T_0 \cup T_1$ 和 $T_{0,1}^c = [d] \setminus T_{0,1}$。接下来，$T_2$ 对应着 $h_{T_{0,1}^c}$ 绝对值中最大 s 个元素。我们利用相同的方法构建 T_3，T_4，… [287]

为了证明这个定理，我们首先需要如下引理，该引理显示 RIP 同样预示着近正交性。

引理 23.10 令 W 是一个 $(\varepsilon, 2s)$-RIP 矩阵。那么对任意两个大小至多为 s 的不相交集合 I，J，以及对任意向量 u，都有 $\langle Wu_I, Wu_J\rangle \le \varepsilon\|u_I\|_2\|u_J\|_2$。

证明 不失一般性，假设 $\|u_I\|_2 = \|u_J\|_2 = 1$，则有

$$\langle Wu_I, Wu_J\rangle = \frac{\|Wu_I + Wu_J\|_2^2 - \|Wu_I - Wu_J\|_2^2}{4}$$

但是，因为 $|I \cup J| \le 2s$，从 RIP 条件可以得到，$\|Wu_I - Wu_J\|_2^2 \le (1+\varepsilon)(\|u_I\|_2^2 - \|u_J\|_2^2) -$

$2(1+\varepsilon)$以及$-\|Wu_i-Wu_j\|_2^2\leqslant-(1-\varepsilon)(\|u_i\|_2^2+\|u_j\|_2^2)=-2(1-\varepsilon)$，由此结论得证。 ■

现在我们准备证明之前的定理23.8。显然，

$$\|h\|_2=\|h_{T_{0,1}}+h_{T_{0,1}^c}\|_2\leqslant\|h_{T_{0,1}}\|_2+\|h_{T_{0,1}^c}\|_2 \tag{23.5}$$

为了证明定理，我们将使用如下两个论断：

论断1：$\|h_{T_{0,1}^c}\|_2\leqslant\|h_{T_0}\|_2+2s^{-1/2}\|x-x_s\|_1$

论断2：$\|h_{T_{0,1}}\|_2\leqslant\frac{2\rho}{1-\rho}s^{-1/2}\|x-x_s\|_1$

将这两个论断和方程(23.5)结合在一起，我们得到

$$\begin{aligned}\|h\|_2&\leqslant\|h_{T_{0,1}}\|_2+\|h_{T_{0,1}^c}\|_2\leqslant2\|h_{T_{0,1}}\|_2+2s^{-1/2}\|x-x_s\|_1\\&\leqslant2\left(\frac{2\rho}{1-\rho}+1\right)s^{-1/2}\|x-x_s\|_1\\&=2\frac{1+\rho}{1-\rho}s^{-1/2}\|x-x_s\|_1\end{aligned}$$

由此结论得证。

证明论断1：

证明这个论断，我们根本无需用到RIP条件，而只需要注意到一个事实：x^*最小化ℓ_1范数。令$j\geqslant1$。对每个$i\in T_j$和$i'\in T_{j-1}$我们有$|h_i|\leqslant|h_{i'}|$。所以，$\|h_{T_j}\|_\infty\leqslant\|h_{T_{j-1}}\|_1/s$。从而，

$$\|h_{T_j}\|_2\leqslant s^{1/2}\|h_{T_j}\|_\infty\leqslant s^{-1/2}\|h_{T_{j-1}}\|_1$$

将此式在$j=2,3,\cdots$上求和，再利用三角不等式，可以得到

288

$$\|h_{T_{0,1}^c}\|_2\leqslant\sum_{j\geqslant2}\|h_{T_j}\|_2\leqslant s^{-1/2}\|h_{T_0^c}\|_1 \tag{23.6}$$

其次，我们说明$\|h_{T_0^c}\|_1$不可能很大。事实上，从x^*的定义可以得出$\|x\|_1\geqslant\|x^*\|_1=\|x+h\|_1$。因此，利用三角不等式，可以得到

$$\|x\|_1\geqslant\|x+h\|_1=\sum_{i\in T_0}|x_i+h_i|+\sum_{i\in T_0^c}|x_i+h_i|\geqslant\|x_{T_0}\|_1-\|h_{T_0}\|_1+\|h_{T_0^c}\|_1-\|x_{T_0^c}\|_1 \tag{23.7}$$

又因为$\|x_{T_0^c}\|_1=\|x-x_s\|_1=\|x\|_1-\|x_{T_0}\|_1$，则有

$$\|h_{T_0^c}\|_1\leqslant\|h_{T_0}\|_1+2\|x_{T_0^c}\|_1 \tag{23.8}$$

将此式与方程(23.6)结合，就能得到

$$\|h_{T_{0,1}^c}\|_2\leqslant s^{-1/2}(\|h_{T_0}\|_1+2\|x_{T_0^c}\|_1)\leqslant\|h_{T_0}\|_2+2s^{-1/2}\|x_{T_0^c}\|_1$$

由此论断1得证。

证明论断2：

对于第2个论断，我们利用RIP条件得出

$$(1-\varepsilon)\|h_{T_{0,1}}\|_2^2\leqslant\|Wh_{T_{0,1}}\|_2^2 \tag{23.9}$$

因为$Wh_{T_{0,1}}=Wh-\sum_{j\geqslant2}Wh_{T_j}=-\sum_{j\geqslant2}Wh_{T_j}$，则有

$$\|Wh_{T_{0,1}}\|_2^2=-\sum_{j\geqslant2}\langle Wh_{T_{0,1}},Wh_{T_j}\rangle=-\sum_{j\geqslant2}\langle Wh_{T_0}+Wh_{T_1},Wh_{T_j}\rangle$$

将RIP条件用到内积上，可以得到对于所有的$i\in\{1,2\}$和$j\geqslant2$有

$$|\langle Wh_{T_i},Wh_{T_j}\rangle|\leqslant\varepsilon\|h_{T_i}\|_2\|h_{T_j}\|_2$$

因为$\|h_{T_0}\|_2+\|h_{T_1}\|_2\leqslant\sqrt{2}\|h_{T_{0,1}}\|_2$，所以

$$\|Wh_{T_{0,1}}\|_2^2 \leqslant \sqrt{2}\varepsilon\|h_{T_{0,1}}\|_2 \sum_{l\geqslant 2}\|h_{T_l}\|_2$$

将此式与方程(23.6)和方程(23.9)结合能够得出

$$(1-\varepsilon)\|h_{T_{0,1}}\|_2^2 \leqslant \sqrt{2}\varepsilon\|h_{T_{0,1}}\|_2 s^{-1/2}\|h_{T_0^c}\|_1$$

重排以上不等式则有

$$\|h_{T_{0,1}}\|_2 \leqslant \frac{\sqrt{2}\varepsilon}{1-\varepsilon}s^{-1/2}\|h_{T_0^c}\|_1$$

最后，利用方程(23.8)，我们推出

$$\|h_{T_{0,1}}\|_2 \leqslant \rho s^{-1/2}(\|h_{T_0}\|_1 + 2\|x_{T_0^c}\|_1) \leqslant \rho\|h_{T_0}\|_2 + 2\rho s^{-1/2}\|x_{T_0^c}\|_1$$

但是因为$\|h_{T_0}\|_2 \leqslant \|h_{T_{0,1}}\|_2$，这意味着

$$\|h_{T_{0,1}}\|_2 \leqslant \frac{2\rho}{1-\rho}s^{-1/2}\|x_{T_0^c}\|_1$$

由此论断2得证。

289

定理23.9的证明

为了证明这个定理，我们采用来自Baraniuk，Davenport，DeVore和Wakin(2008)的方法。该策略是将Johnson-Lindenstrauss(JL)引理和对覆盖的简单讨论结合在一起。

我们从单位球的覆盖性质谈起。

引理23.11 令$\varepsilon\in(0,1)$。存在一个有限集$Q\subset\mathbb{R}^d$，其大小$|Q|\leqslant\left(\frac{3}{\varepsilon}\right)^d$，使得下式成立：

$$\sup_{x:\|x\|\leqslant 1}\min_{v\in Q}\|x-v\|\leqslant\varepsilon$$

证明 令k为一个整数，再令

$$Q'=\left\{x\in\mathbb{R}^d:\forall j\in[d],\exists i\in\left\{-k,-k+1,\cdots,k\right\}\text{ s.t. }x_j=\frac{i}{k}\right\}$$

显然，$|Q'|=(2k+1)^d$。我们令$Q=Q'\cap B_2(1)$，其中$B_2(1)$是$\mathbb{R}^d$中的单位ℓ_2球。因为Q'中的点在单位ℓ_∞球，则Q的大小是Q'的大小乘以单位ℓ_2球和单位ℓ_∞球的体积比。ℓ_∞的体积是2^d，而$B_2(1)$的体积是

$$\frac{\pi^{d/2}}{\Gamma(1+d/2)}$$

简单起见，假设d是偶数。因此

$$\Gamma(1+d/2)=(d/2)!\geqslant\left(\frac{d/2}{e}\right)^{d/2}$$

其中，最后一个不等式中，我们采用了斯特林近似。总的来说，我们可以推出

$$|Q|\leqslant(2k+1)^d(\pi/e)^{d/2}(d/2)^{-d/2}2^{-d} \tag{23.10}$$

现在固定k。对每个$x\in B_2(1)$，令向量$v\in Q$，其第i个元素是$\operatorname{sign}(x_i)\lfloor|x_i|k\rfloor/k$。那么，对每个元素都有$|x_i-v_i|\leqslant 1/k$。所以，

$$\|x-v\|\leqslant\frac{\sqrt{d}}{k}$$

为保证上式右端项至多为ε，我们令$k=\lceil\sqrt{d}/\varepsilon\rceil$，将该值带入方程(23.10)，可以推出

$$|Q|\leqslant(3\sqrt{d}/(2\varepsilon))^d(\pi/e)^{d/2}(d/2)^{-d/2}=\left(\frac{3}{\varepsilon}\sqrt{\frac{\pi}{2e}}\right)^d\leqslant\left(\frac{3}{\varepsilon}\right)^d$$ ■

令向量 x 能被写为 $x=U\alpha$，其中 U 为正交矩阵而 $\|\alpha\|_0\leqslant s$。将之前的覆盖性质和 JL 引理(引理 23.4)相结合，我们将看到一个随机矩阵 W 不会使得任何 x 失真(在随机映射意义下不失真，详见引理 23.4)。

引理 23.12　令 U 为 $d\times d$ 的正交矩阵，$I\subseteq[d]$ 为指标集合，大小 $|I|=s$。令 S 为 $\{U_i:i\in I\}$ 的线性展开，其中 U_i 是 U 的第 i 列。再令 $\delta\in(0,1)$，$\varepsilon\in(0,1)$ 和 $n\in\mathbb{N}$，且

[290]

$$n\geqslant 24\frac{\log(2/\delta)+s\log(12/\varepsilon)}{\varepsilon^2}$$

那么，以至少 $1-\delta$ 的概率，对随机矩阵 $W\in\mathbb{R}^{n\times d}$，其中 W 的每个元素独立，均服从 $N(0,1/n)$ 分布，则有

$$\sup_{x\in S}\left|\frac{\|Wx\|}{\|x\|}-1\right|\leqslant\varepsilon$$

证明　只需要对所有 $x\in S$ 且 $\|x\|=1$ 证明引理即可。我们可以重写 x 为 $x=U_I\alpha$，其中 $\alpha\in\mathbb{R}^s$，且 $\|\alpha\|_2=1$，而矩阵 U_I 的列为 $\{U_i:i\in I\}$。采用引理 23.11，我们知道存在集合 Q，其大小满足 $|Q|\leqslant(12/\varepsilon)^s$，那么

$$\sup_{\alpha:\|\alpha\|=1}\min_{v\in Q}\|\alpha-v\|\leqslant(\varepsilon/4)$$

但这为 U 是正交的，所以也能得到

$$\sup_{\alpha:\|\alpha\|=1}\min_{v\in Q}\|U_I\alpha-U_Iv\|\leqslant(\varepsilon/4)$$

将引理 23.4 用于集合 $\{U_Iv:v\in Q\}$，可以得到满足引理中条件的 n，那么下式以至少 $1-\delta$ 的概率成立：

$$\sup_{v\in Q}\left|\frac{\|WU_Iv\|^2}{\|U_Iv\|^2}-1\right|\leqslant\varepsilon/2$$

这也意味着

$$\sup_{v\in Q}\left|\frac{\|WU_Iv\|}{\|U_Iv\|}-1\right|\leqslant\varepsilon/2$$

令 a 为满足下式的最小数：

$$\forall x\in S,\frac{\|Wx\|}{\|x\|}\leqslant 1-a$$

显然 $a<1$，我们的目标是让 $a\leqslant\varepsilon$。注意到对任意单位范数的 $x\in S$ 都存在 $v\in Q$ 使得 $\|x-U_Iv\|\leqslant\varepsilon/4$，所以

$$\|Wx\|\leqslant\|WU_Iv\|+\|W(x-U_Iv)\|\leqslant 1+\varepsilon/2+(1-a)\varepsilon/4$$

因此，

$$\forall x\in S,\frac{\|Wx\|}{\|x\|}\leqslant 1-(\varepsilon/2-(1+a)\varepsilon/4)$$

但 a 的定义意味着

$$a\leqslant\varepsilon/2+(1+a)\varepsilon/4\Rightarrow a\leqslant\frac{\varepsilon/2+\varepsilon/4}{1-\varepsilon/4}\leqslant\varepsilon$$

由此证明对所有 $x\in S$，我们有 $\frac{\|Wx\|}{\|x\|}-1\leqslant\varepsilon$，不等式的另一半也可以由此得到，因为

$$\|Wx\|\geqslant\|WU_Iv\|-\|W(x-U_Iv)\|\geqslant 1-\varepsilon/2-(1-\varepsilon)\varepsilon/4\geqslant 1-\varepsilon$$ ■

之前的引理告诉我们对任意单位范数的 $x\in S$，都有

[291]

$$(1-\varepsilon)\leqslant\|Wx\|\leqslant(1+\varepsilon)$$

这意味着

$$(1-2\varepsilon)\leqslant\|Wx\|^2\leqslant(1+3\varepsilon)$$

23.9 的证明由在所有可能的 I 上的联合界得到。

23.4 PCA 还是压缩感知

假设我们将对某个给定数据集合进行降维，应该使用 PCA 还是压缩感知呢？在这一节我们将通过强调并理解两种方法背后的假设，来解决这个问题。

首先需要理解的是每种方法在何种情况下能保证完美的数据恢复。当数据集合包含在 $\mathbb{R}^d$ 中的 n 维子空间时，PCA 能保证完美的恢复。而当数据集合是稀疏(在某些基上)的时候，压缩感知能保证完美的恢复。基于以上事实，我们将阐述在何种情况下 PCA 比压缩感知更有效，反之亦然。

第一个例子假设所有数据是 $\mathbb{R}^d$ 的标准基向量，即 e_1，…，e_d，其中每个向量 e_i 的第 i 位为 1，其余全为 0。这种情况下，所有的数据都是 1-稀疏的。因此，只要数据个数满足 $m\geqslant\Omega(\log(d))$，压缩感知就能保证完美恢复。另一方面，PCA 在这个数据集合的性能就很差，因为只要 $n<d$，这些数据就远远不能称其为 n 维子空间。事实上，很容易验证在此种情况下，PCA 的恢复误差(也即是，方程(23.1)的目标除以 m)为 $(d-n)/d$，当 $n\leqslant d/2$ 时这就比 1/2 大了。

接下来我们给出一个 PCA 效果比压缩感知好的例子。考虑正好在 n 维子空间中的 m 条数据。显然在此种情况下，PCA 将得到完美恢复。而对于压缩感知，注意到所有的数据在任何正交基(其中前 n 个基向量线性展开为这个子空间)下为 n-稀疏的。所以，如果我们将维数降低到 $\Omega(n\log(d))$，压缩感知也能起作用。然而，对 n 维的情况，压缩感知就失效了。PCA 对此种噪声有更强的恢复能力。详见 Chang，Weiss & Freeman(2009)中的讨论。

23.5 小结

我们介绍了两种采用线性变换的降维方法：PCA 和随机映射。如果将重构过程限制在线性操作下，PCA 在均方重构误差的意义下式最佳的。然而，如果允许采用非线性重构，PCA 不一定最优。特别地，对于稀疏数据，随机映射表现显著超过了 PCA。这个事实就是压缩感知方法的核心。

23.6 文献评注

PCA 等价于采用奇异值分解(SVD)进行最佳子空间近似。附录 C 详细描述了 SVD 方法。SVD 追溯到 Eugenio Beltrami(1873)和 Camille Jordan(1874)，又多次被重新发现。在统计学文献中，它由 Pearson(1901)首次提出。除了 PCA 和 SVD，还有其他一些名字在不同的科学领域也表示相同的意思。例如 Eckart-Young 定理(在 Carl Eckart 和 Gale Young 于 1936 年研究了该方法之后)，还有 Schmidt-Mirsky 定理，因子分析，以及 Hotelling 变换。 292

压缩感知在 Donoho(2006)和 Candes & Tao(2005)中被引入。Candes(2006)也有提及。

23.7 练习

23.1 本练习中将显示，通常情况下，基于线性压缩方案的精确恢复是不可能的。

1) 令 $A\in\mathbb{R}^{n\times d}$ 为任意满足 $n\leq d-1$ 的压缩矩阵。试证存在 u，$v\in\mathbb{R}^d$，$u\neq v$ 使得 $Au=Av$。

2) 试证基于线性压缩方案的精确恢复是不可能的。

23.2 令 $\boldsymbol{\alpha}\in\mathbb{R}^d$ 使得 $\alpha_1\geq\alpha_2\geq\cdots\geq\alpha_d\geq 0$。试证

$$\max_{\boldsymbol{\beta}\in[0,1]^d:\|\boldsymbol{\beta}\|_1\leq n}\sum_{j=1}^{d}\alpha_j\beta_j-\sum_{i=1}^{n}\alpha_i$$

提示：考虑每个向量 $\boldsymbol{\beta}\in[0,1]^d$ 且 $\|\boldsymbol{\beta}\|_1\leq n$。令 i 为使得 $\beta_i\leq 1$ 的最小的下标。如果 $i=n+1$，那么证明完成，否则，可以找到对某个 $j>i$，增加 β_j，减少 β_i。然后可以得到一个优的解。这意味着最优的解是令对 $i\leq n$ 有 $\beta_i=1$，而对 $i>n$ 有 $\beta_i=0$。

23.3 **核 PCA**：本练习中我们将展示采用核方法（参看第 16 章），如何将 PCA 用于构造非线性降维。

令 $\mathcal{X}$ 为样本空间，$S=\{x_1,\cdots,x_m\}$ 为包含 $\mathcal{X}$ 中点的集合。考虑特征映射 $\psi:\mathcal{X}\to V$，其中 V 为希尔伯特空间（可能无限维）。令 $K:\mathcal{X}\times\mathcal{X}$ 为核函数，即 $k(x,x')=\langle\psi(x),\psi(x')\rangle$。核 PCA 是利用 ψ 将 S 中的元素映射到 V 中，然后利用 PCA 将 $\{\psi(x_1),\cdots,\psi(x_m)\}$ 映射到 $\mathbb{R}^d$ 中的过程。该过程的输出就是被降维的元素。

假设每次 $K(\cdot,\cdot)$ 的计算复杂度为常数时间，试证该降维过程的计算复杂度在基于 m 和 n 的多项式时间。特别地，如果你的算法实现要求计算两个矩阵 A 和 B 的乘积，则需要验证乘积是否能被计算（矩阵维度符合乘积要求）。同样地，如果需要对某个矩阵 C 执行 SVD，则需要验证分解能否被执行。

23.4 **方差最大化的 PCA 解释**

令 $x_1,\cdots,x_m$ 为 $\mathbb{R}^d$ 中的 m 个向量，再令随机向量 x 的分布和 $x_1,\cdots,x_m$ 上的正态分布一致。假设 $\mathbb{E}[x]=\mathbf{0}$。

1) 考虑寻找单位向量 $w\in\mathbb{R}^d$，使得随机变量 $\langle w,x\rangle$ 方差最大。也就是说，需要求解以下问题：

[253]

$$\operatorname*{argmax}_{w:\|w\|=1}\operatorname{Var}[\langle w,x\rangle]=\operatorname*{argmax}_{w:\|w\|=1}\frac{1}{m}\sum_{i=1}^{m}(\langle w,x_i\rangle)^2$$

试证该问题的解就是令 w 为 $x_1,\cdots,x_m$ 的第一主成分。

2) 令 w_1 为前一个问题中的第一主向量。现在，假设我们需要寻找第二个单位向量，$w_2\in\mathbb{R}^d$，使得随机变量 $\langle w_2,x\rangle$ 方差最大，但与 $\langle w_1,x\rangle$ 不相关。也就是说，需要求解下列问题：

$$\operatorname*{argmax}_{w:\|w\|=1:\mathbb{E}[\langle w,x\rangle\langle w_1,x\rangle]=0}\operatorname{Var}[\langle w,x\rangle]$$

试证该问题的解就是令 w 为 $x_1,\cdots,x_m$ 的第二主成分。

提示：注意到

$$\mathbb{E}[(\langle w_1,x\rangle)(\langle w,x\rangle)]=w_1^{\mathrm{T}}\mathbb{E}[xx^{\mathrm{T}}]w=\alpha w_1^{\mathrm{T}}Aw$$

其中 $A=\sum_i x_ix_i^{\mathrm{T}}$。因为 w 为 A 的特征向量，因此限制条件 $\mathbb{E}[(\langle w_1,x\rangle)(\langle w,x\rangle)]=0$ 等价于 $\langle w_1,w\rangle=0$。

23.5 **SVD 和 PCA 之间的关系**：采用 SVD 的定理（推论 C.6）来证明定理 23.2。

23.6 **保内积的随机映射**：Johnson-Lindenstrauss 引理告诉我们，随机映射保持有限向量集合间的距离。本练习中，你需要证明如果向量集合在单位球内，那么不仅两两向量间的距离被保持，而且内积也被保持。

令 Q 为 $\mathbb{R}^d$ 中的有限向量集合，再假设对每个 $x\in Q$ 都有 $|x|\leqslant 1$。

1) 令 $\delta\in(0,1)$，且 n 为某整数，使得

$$\varepsilon=\sqrt{\frac{6\log(|Q|^2/\delta)}{n}}\leqslant 3$$

试证以至少 $1-\delta$ 的概率，对随机矩阵 $W\subset\mathbb{R}^{n\times d}$，其中 W 的每个元素独立，均服从 $N(0,1/n)$ 分布，则下式

$$|\langle Wu,Wv\rangle-\langle u,v\rangle|\leqslant\varepsilon$$

对每个 $u,v\in Q$ 均满足。

*2) 令 $x_1,\cdots,x_n$ 为 $\mathbb{R}^d$ 中的单位至多为 1 的向量，再假设这些向量以间隔 γ 线性可分。假设 $d\gg 1/\gamma^2$。试证存在常数 $c>0$，使得对 $n=c/\gamma^2$，将 $\mathbb{R}^d$ 中这些向量随机投影到 $\mathbb{R}^n$，则以至少 99% 的概率，投射向量以间隔 $\gamma/2$ 线性可分。

234

生成模型

本书的开始介绍了一个与数据分布无关的学习框架：也就是，无需对数据的潜在分布做出任何假设。进一步，我们采用判别式的学习方法，以得到一个高精度的预测器，而不是刻画数据的潜在分布。本章将介绍生成式的学习方法，即对数据的潜在分布的参数形式作出假设，并估计其模型参数。这个任务通常被称为参数概率密度估计。

判别式学习的一个显著优点是，它直接对目标量（预测精度）进行优化，而不是对潜在分布进行学习。Vladimir Vapnik 在有限数量信息解决问题的基本原则中，强调：

在解决一个给定问题时，要设法避免把解决一个更为一般的问题作为中间步骤。

当然，如果我们能够成功地对数据的潜在分布进行学习，那么就可以使用贝叶斯预测最优分类，在这个意义上，我们就可以被认为"专家"。困难在于，对数据的潜在分布的学习，通常比预测器的训练更为困难。然而，在某些情况下，采用生成式学习是合理的。例如，有时，对模型的参数估计比训练预测器更容易（计算量更小）。此外，有些情况下，当学习的任务不明确时，我们可以对数据进行建模，用于今后的预测任务，或是对数据本身进行理解和分析。

本章首先介绍估计数据参数的一个常见的统计方法，即极大似然准则。然后，将描述两个生成假设，这将极大地简化学习过程。接下来，讨论含有隐变量的概率模型参数的极大似然估计法，即 EM 算法。在本章结尾，我们将简单介绍贝叶斯推理。

24.1 极大似然估计

我们举一个简单的例子。假定某制药公司开发了一种新的药物来治疗一种致命的疾病。为了估计服药患者的存活概率分布，制药公司采集了服用该药物的 m 个患者的信息。令 $S=(x_1, \cdots, x_m)$ 表示 m 个患者构成的训练集，其中，如果患者 i 存活，则记 $x_i=1$，否则记 $x_i=0$。我们可以使用存活率 $\theta\in[0, 1]$ 来刻画数据的概率分布。

295

我们希望在给定训练集 S 的基础上，对参数 θ 进行估计。一个直观的想法就是，将训练集 S 中 1 的平均出现频率，作为参数 θ 的估计。即

$$\hat{\theta}=\frac{1}{m}\sum_{i=1}^{m}x_i \tag{24.1}$$

显然，$\mathbb{E}_S(\hat{\theta})=\theta$，即 $\hat{\theta}$ 是参数 θ 的无偏估计。此外，由于 $\hat{\theta}$ 是 m 个独立同分布的随机变量的均值，由 Hoeffding 不等式知

$$|\hat{\theta}-\theta|\leq\sqrt{\frac{\log(2/\delta)}{2m}} \tag{24.2}$$

成立的概率不低于 $1-\delta$。

事实上，关于 $\hat{\theta}$ 的另一个解释是，它是参数 θ 的极大似然估计。我们首先写出样本集 S 的生成概率

$$\mathbb{P}[S=(x_1,\cdots,x_m)]=\prod_{i=1}^{m}\theta^{x_i}(1-\theta)^{1-x_i}=\theta^{\sum_i x_i}(1-\theta)^{\sum_i(1-x_i)}$$

对上式取对数，就是给定参数 θ 时，样本集 S 的对数似然函数

$$L(S;\theta)=\log(\mathrm{P}[S=(x_1,\cdots,x_m)])=\log(\theta)\sum_i x_i+\log(1-\theta)\sum_i(1-x_i)$$

如果我们把对数似然度看作参数 θ 的函数，则极大似然估计就是使得似然程度最大的那个点

$$\hat{\theta}=\underset{\theta}{\operatorname{argmax}}L(S;\theta) \tag{24.3}$$

对于我们的例子，公式(24.1)给出了存活率 θ 的极大似然估计，因为，令 $L(S,\theta)$ 关于 θ 的导数为 0，有

$$\frac{\sum_i x_i}{\theta}-\frac{\sum_i(1-x_i)}{1-\theta}=0$$

对等式求解，就得到了公式(24.1)给出的存活率 θ 的估计。

24.1.1 连续随机变量的极大似然估计

假设 X 是一个连续型随机变量。那么，对于所有的 $x\in\mathbb{R}$，有 $\mathrm{P}[X=x]=0$。于是之前给出的似然度的定义，对于连续性随机变量来说，似乎不太合理。为了克服这个技术困难，我们可以定义似然度为随机变量 X 的概率密度函数在 x 点的对数值。具体地，由分布 $\mathcal{P}_\theta$ 采样得到的一个独立同分布训练集 $S=(x_1,\cdots x_m)$，我们定义 S 关于参数 θ 的似然函数为

$$L(S;\theta)=\log\left(\prod_{i=1}^{m}\mathcal{P}_\theta(x_i)\right)=\sum_{i=1}^{m}\log(\mathcal{P}_\theta(x_i))$$

295

参数 θ 的极大似然估计就是函数 $L(S;\theta)$ 关于 θ 的极大值点。

现以一个正态分布的随机变量为例来说明求极大似然估计的过程。设 X 服从以 $\theta=(\mu,\sigma)$ 为参数的正态分布：

$$\mathcal{P}_\theta(x)=\frac{1}{\sigma\sqrt{2\pi}}\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

则似然函数为

$$L(S;\theta)=-\frac{1}{2\sigma^2}\sum_{i=1}^{m}(x_i-\mu)^2-m\log(\sigma\sqrt{2\pi})$$

为使似然函数达到最大，分别令其关于 μ 和 σ 的偏导数为 0，可以得到如下方程组：

$$\frac{\mathrm{d}}{\mathrm{d}\mu}L(S;\theta)=\frac{1}{\sigma^2}\sum_{i=1}^{m}(x_i-\mu)=0$$

$$\frac{\mathrm{d}}{\mathrm{d}\sigma}L(S;\theta)=\frac{1}{\sigma^3}\sum_{i=1}^{m}(x_i-\mu)^2-\frac{m}{\sigma}=0$$

对方程组进行求解，得到极大似然估计：

$$\hat{\mu}=\frac{1}{m}\sum_{i=1}^{m}x_i \quad 和 \quad \hat{\sigma}=\sqrt{\frac{1}{m}\sum_{i=1}^{m}(x_i-\hat{\mu})^2}$$

值得注意的是，极大似然估计不总是无偏的。例如，本例中，均值的估计 $\hat{\mu}$ 是无偏的，但标准差的估计 $\hat{\sigma}$ 就不是无偏的(见练习 24.1)。

符号简化

为了简化符号，本章中统一用 $\mathcal{P}[X=x]$ 描述下面两种情况：离散随机变量 $X=x$ 的概率，或连续变量 X 在 x 点的概率密度。

24.1.2 极大似然与经验风险最小化

极大似然估计与我们在前面的章节中广泛研究的经验风险最小化(ERM)原则，是具有一定相似性的。在经验风险最小化原则中，有一个假设集$\mathcal{H}$，利用训练集进行学习，选取假设$h\in\mathcal{H}$，实现使得经验风险最小化。本小节将证明，极大似然估计是对于特定的损失函数的经验风险最小化。

对于给定的参数θ和观测样本x，定义损失函数为

297

$$\ell(\theta,x)=-\log(\mathcal{P}_\theta[x]) \tag{24.4}$$

也就是说，假设观测样本X服从分布$\mathcal{P}_\theta$，损失函数$\ell(\theta,x)$与x的对数似然函数相差一个负号。该损失函数通常被称为对数损失，在此基础上，很容易验证，极大似然准则等价于(24.4)式定义的损失函数的经验风险最小化：即

$$\operatorname*{argmin}_\theta\sum_{i=1}^{m}(-\log(\mathcal{P}_\theta[x_i]))=\operatorname*{argmax}_\theta\sum_{i=1}^{m}\log(\mathcal{P}_\theta[x_i])$$

数据服从的潜在分布为$\mathcal{P}$(不必满足参数化形式)，参数θ的真实风险为

$$\begin{aligned}\mathbf{E}_{x\sim\mathcal{P}}[\ell(\theta,x)]&=-\sum_x\mathcal{P}[x]\log(\mathcal{P}_\theta[x])\\&=\underbrace{\sum_x\mathcal{P}[x]\log\left(\frac{\mathcal{P}[x]}{\mathcal{P}_\theta[x]}\right)}_{D_{\mathrm{KL}}(\mathcal{P}\|\mathcal{P}_\theta)}-\underbrace{\sum_x\mathcal{P}[x]\log\left(\frac{1}{\mathcal{P}[x]}\right)}_{H(\mathcal{P})}\end{aligned} \tag{24.5}$$

其中，D_{KL}称为相对熵，H称为熵函数。相对熵是描述两个概率分布的差异的一种度量，对于离散分布，相对熵总是非负的，并且等于0当且仅当两个分布是相同的。由此可知，当$\mathcal{P}_\theta=\mathcal{P}$时，真实风险达到最小值。

公式(24.5)刻画了生成式的假设对于密度估计的影响，即使是在无穷多样本的极限情况下，该影响依然存在。该式还表明，如果潜在分布具有参数化形式，那么可以通过选择合适的参数，使风险降为潜在分布的熵。然而，如果潜在分布不满足假设的参数化形式，那么由最优参数所确定的模型也可能是较差的，模型的优劣是用相对熵刻画的。

24.1.3 泛化分析

对于给定的有限训练集，如何评价极大似然估计的优劣？

为了回答这一问题，需要针对概率密度估计问题，定义其近似解的优良性准则。在判别式学习中，能够清晰地确定“损失函数”；而对于生成式学习，模型的损失函数的定义是有多种可能的。由上节知，公式(24.5)列出的期望对数损失是一种最自然的损失定义。

在某些情况下，很容易验证，极大似然准则确保了真实风险的最小化。例如，假定某正态分布的方差为1，对其均值进行估计。由前面的小节知，均值的极大似然估计就是样

298

本的均值$\hat{\mu}=\frac{1}{m}\sum_i x_i$。设$\mu^*$是最优的参数估计值，则有

$$\begin{aligned}\mathbf{E}_{x\sim\mathcal{N}(\mu^*,1)}[\ell(\hat{\mu},x)-\ell(\mu^*,x)]&=\mathbf{E}_{x\sim\mathcal{N}(\mu^*,1)}\log\left(\frac{\mathcal{P}_{\mu^*}[x]}{\mathcal{P}_{\hat{\mu}}[x]}\right)\\&=\mathbf{E}_{x\sim\mathcal{N}(\mu^*,1)}\left(-\frac{1}{2}(x-\mu^*)^2+\frac{1}{2}(x-\hat{\mu})^2\right)\\&=\frac{\hat{\mu}^2}{2}-\frac{(\mu^*)^2}{2}+(\mu^*-\hat{\mu})\mathbf{E}_{x\sim\mathcal{N}(\mu^*,1)}[x]\end{aligned}$$

$$
\begin{aligned}
&= \frac{\hat{\mu}^2}{2} - \frac{(\mu^*)^2}{2} + (\mu^* - \hat{\mu})\mu^* \\
&= \frac{1}{2}(\hat{\mu} - \mu^*)^2
\end{aligned} \tag{24.6}
$$

注意到，$\hat{\mu}$作为m个正态随机变量的均值，也服从正态分布，其均值为μ^*，标准差为σ^*/m。于是，我们以至少$1-\delta$的概率保证$|\hat{\mu}-\mu^*|\leqslant\varepsilon$成立，其中，$\varepsilon$由$\sigma^*/m$和$\delta$确定。

在某些情况下，极大似然估计会出现过拟合的问题。如，考察伯努利随机变量X，设$\mathcal{P}[X=1]=\theta^*$。由Hoeffding不等式，很容易得知：$|\theta^*-\hat{\theta}|$以较大概率成立（详见公式(24.2)）。然而，由公式(24.5)定义的期望对数损失函数却未必能足够小。为此说明这一事实，假定θ^*非零，充分小。那么，m个样本中全部都是0的概率为$(1-\theta^*)^m$，是大于$e^{-m\theta^*}$的。由此知，当$m\leqslant\frac{\log(2)}{2\theta^*}$时，样本中全部为0的概率至少是50%，此时由极大似然准则，有$\hat{\theta}=0$。此时的真实风险为

$$
\begin{aligned}
\mathop{\mathbb{E}}_{x\sim\mathcal{P}^*}[\ell(\hat{\theta},x)] &= \theta^*\,\ell(\hat{\theta},1) + (1-\theta^*)\,\ell(\hat{\theta},0) \\
&= \theta^*\log(1/\hat{\theta}) + (1-\theta^*)\log(1/(1-\hat{\theta})) \\
&= \theta^*\log(1/0) = \infty
\end{aligned}
$$

这个简单的例子说明，我们使用极大似然准则时要慎重。

为克服过拟合问题，我们可以采用以前遇到的各种处理手段。练习24.2介绍了一个简单的正则化技术。

24.2 朴素贝叶斯

朴素贝叶斯分类器是利用生成假设和参数估计来简化学习过程的经典范例。对于给定特征向量$x=(x_1,\cdots,x_d)$，我们的目的是预测样本的标签$y\in\{0,1\}$。假定每个x_i都属于$\{0,1\}$。回想一下，贝叶斯最优分类器是

$$
h_{\text{Bayes}}(x) = \operatorname*{argmax}_{y\in\{0,1\}} \mathcal{P}[Y=y \mid X=x]
$$

289

为了描述概率函数$\mathcal{P}[Y=y\mid X=x]$，我们需要2^d个参数，每个对应于给定一个$x\in\{0,1\}^d$时概率函数$\mathcal{P}[Y=1\mid X=x]$的值。这意味着，我们所需的样本数量随特征个数呈指数型增长。在朴素贝叶斯方法，我们给出的(朴素的)生成假设是，对于给定的标签，各特征之间是彼此独立的。即

$$
\mathcal{P}[X=x \mid Y=y] = \prod_{i=1}^{d} \mathcal{P}[X_i = x_i \mid Y=y]
$$

有了这个假设，并使用贝叶斯法则，贝叶斯最优分类可进一步简化为：

$$
\begin{aligned}
h_{\text{Bayes}}(x) &= \operatorname*{argmax}_{y\in\{0,1\}} \mathcal{P}[Y=y \mid X=x] \\
&= \operatorname*{argmax}_{y\in\{0,1\}} \mathcal{P}[Y=y]\,\mathcal{P}[X=x \mid Y=y] \\
&= \operatorname*{argmax}_{y\in\{0,1\}} \mathcal{P}[Y=y]\prod_{i=1}^{d} \mathcal{P}[X_i=x_i \mid Y=y]
\end{aligned} \tag{24.7}
$$

也就是说，现在我们需要估计的参数个数只有$2d+1$。这里，所述的生成假设，帮助我们显著地减少了需要学习的参数数量。当我们使用最大似然原则进行估计参数时，得到的分类被称为朴素贝叶斯分类器。

24.3 线性判别分析

线性判别分析(Linear Discriminant Analysis，LDA)是借助生成假设简化学习过程的另一个范例。我们再次考虑给定特征矢量 $x=(x_1, \cdots, x_d)$ 的基础上，预测对应的标签 $y\in\{0, 1\}$ 问题。但现在的生成假设如下：首先，我们假设 $\mathcal{P}[Y=1]=\mathcal{P}[Y=1]=1/2$；第二，我们假定，给定 Y 时 X 服从高斯分布；最后，对于标签的两个不同值，假定其对应的高斯分布的协方差矩阵是相同的。从形式上看，令 $\boldsymbol{\mu}_0, \boldsymbol{\mu}_1\in\mathbb{R}^d$ 且 Σ 是一个协方差矩阵，则密度分布由下式给出

$$\mathcal{P}[X=x \mid Y=y]=\frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}}\exp\left(-\frac{1}{2}(x-\boldsymbol{\mu}_y)^{\mathrm{T}}\Sigma^{-1}(x-\boldsymbol{\mu}_y)\right)$$

我们在上一节已经证明了，使用贝叶斯法则，有

$$h_{\mathrm{Bayes}}(x)=\operatorname*{argmax}_{y\in\{0,1\}}\mathcal{P}[Y=y]\,\mathcal{P}[X=x \mid Y=y]$$

这意味着，我们将预测 $h_{\mathrm{Bayes}}(x)=1$ 当且仅当

320

$$\log\left(\frac{\mathcal{P}[Y=1]\,\mathcal{P}[X=x \mid Y=1]}{\mathcal{P}[Y=0]\,\mathcal{P}[X=x \mid Y=0]}\right)>0$$

这个比例通常被称为对数似然比。

在我们的假设下，对数似然比为

$$\frac{1}{2}(x-\boldsymbol{\mu}_0)^{\mathrm{T}}\Sigma^{-1}(x-\boldsymbol{\mu}_0)-\frac{1}{2}(x-\boldsymbol{\mu}_1)^{\mathrm{T}}\Sigma^{-1}(x-\boldsymbol{\mu}_1)$$

我们可以将上式改写为 $\langle w, x\rangle+b$，其中

$$w=(\boldsymbol{\mu}_1-\boldsymbol{\mu}_0)^{\mathrm{T}}\Sigma^{-1} \quad 和 \quad b=\frac{1}{2}\left(\boldsymbol{\mu}_0^{\mathrm{T}}\Sigma^{-1}\boldsymbol{\mu}_0-\boldsymbol{\mu}_1^{\mathrm{T}}\Sigma^{-1}\boldsymbol{\mu}_1\right) \tag{24.8}$$

由前述推导的结果可知，在上述生成假设下，贝叶斯最优分类器就是线性分类器。此外，人们可以通过最大似然估计等方法，利用训练数据来估计参数 $\boldsymbol{\mu}_0$，$\boldsymbol{\mu}_1$ 和 Σ，从而实现分类器的训练。事实上，有了参数 $\boldsymbol{\mu}_0$，$\boldsymbol{\mu}_1$ 的估计值，可以通过公式(24.8)计算 w 和 b 的值。

24.4 隐变量与 EM 算法

在生成模型中，我们假设数据通过在实例空间中依据一个特定参数分布采样生成。有时，借助隐变量可以很方便地表达这个分布。一个自然例子是混合高斯分布。即实例空间 $\mathcal{X}=\mathbb{R}^d$ 并假定每个 x 按照如下方法产生：首先，我们在 $\{1, \cdots, k\}$ 中选择一个随机数，令 Y 为对应的随机变量，记 $\mathcal{P}[Y=y]=c_y$；第二，我们根据 Y 的取值，依照高斯分布生成 x

$$\mathcal{P}[X=x \mid Y=y]=\frac{1}{(2\pi)^{d/2}|\Sigma_y|^{1/2}}\exp\left(-\frac{1}{2}(x-\boldsymbol{\mu}_y)^{\mathrm{T}}\Sigma_y^{-1}(x-\boldsymbol{\mu}_y)\right) \tag{24.9}$$

于是，X 的密度函数可以写为

$$\begin{aligned}\mathcal{P}[X=x]&=\sum_{y=1}^{k}\mathcal{P}[Y=y]\,\mathcal{P}[X=x \mid Y=y]\\&=\sum_{y=1}^{k}c_y\frac{1}{(2\pi)^{d/2}|\Sigma_y|^{1/2}}\exp\left(-\frac{1}{2}(x-\boldsymbol{\mu}_y)^{\mathrm{T}}\Sigma_y^{-1}(x-\boldsymbol{\mu}_y)\right)\end{aligned}$$

需要注意的是，Y 是无法从数据中观测到的一个隐藏的变量。尽管如此，我们仍然引入 Y，因为它有助于我们将 X 的概率描述为一个简单的参数形式。

更一般地，令 $\boldsymbol{\theta}$ 是 X 和 Y 的联合分布的参数(例如，在前面的例子中，$\boldsymbol{\theta}$ 由 c_y，$\boldsymbol{\mu}_y$ 和

Σ_y 组成，其中 y 取遍 $1, \cdots, k$ 中的所有值）。然后，可以将观察数据 x 的对数似然写为

$$\log(\mathcal{P}_\theta[X=x]) = \log\Big(\sum_{y=1}^{k} \mathcal{P}_\theta[X=x, Y=y]\Big)$$

[301]

给定一个独立同分布样本集 $S=(x_1, \cdots, x_n)$，我们想找到最优的 $\boldsymbol{\theta}$，使 S 的对数似然最大化

$$\begin{aligned} L(\boldsymbol{\theta}) &= \log \prod_{i=1}^{n} \mathcal{P}_\theta[X=x_i] \\ &= \sum_{i=1}^{n} \log \mathcal{P}_\theta[X=x_i] \\ &= \sum_{i=1}^{n} \log\Big(\sum_{y=1}^{k} \mathcal{P}_\theta[X=x_i, Y=y]\Big) \end{aligned}$$

因此，最大似然估计是如下最大化问题的解：

$$\operatorname*{argmax}_{\theta} L(\boldsymbol{\theta}) = \operatorname*{argmax}_{\theta} \sum_{i=1}^{n} \log\Big(\sum_{y=1}^{k} \mathcal{P}_\theta[X=x_i, Y=y]\Big)$$

在许多情况下，在对数函数内的求和，使前述优化问题很难计算。期望最大化（EM）算法（由 Dempster，Laird 和 Rubin 提出）是一个反复搜索 $L(\boldsymbol{\theta})$ 的局部最大值的算法。虽然 EM 不能保证找到全局最大值，但它在实践中取得了很好的应用。

EM 特别适用于这样的情形，如果我们能够确定隐变量 Y 的取值，则最大似然优化问题是非常易于处理的。更精确地说，定义关于 $m \times k$ 矩阵和参数 $\boldsymbol{\theta}$ 的函数如下：

$$F(Q, \boldsymbol{\theta}) = \sum_{i=1}^{m} \sum_{y=1}^{k} Q_{i,y} \log(\mathcal{P}_\theta[X=x_i, Y=y])$$

如果 Q 的每一行定义给出 $X=x_i$ 时第 i 个潜变量的概率，那么我们将 $F(Q, \boldsymbol{\theta})$ 解释为训练集 $(x_1, y_1), \cdots, (x_m, y_m)$ 的预期对数似然，其中，期望是相对于每个 y_i 的近似分布的，即矩阵 Q 的第 i 行所确定的 y_i 的近似分布。在 F 的定义中，求和在对数函数之外，并且我们假设这使得关于 $\boldsymbol{\theta}$ 的优化问题容易处理：

假设 24.1 对于任意 $Q \in [0, 1]^{m \times k}$，如果 Q 的每一行求和都是 1，那么优化问题

$$\operatorname*{argmax}_{\theta} F(Q, \boldsymbol{\theta})$$

易解。

EM 的直观的想法是，我们用“鸡生蛋，蛋生鸡”的思路来解决问题：一方面，如果我们已知 Q，那么由假设，找到最优化问题的最优解 $\boldsymbol{\theta}$ 是容易求解的。另一方面，如果我们已知参数 $\boldsymbol{\theta}$，可以令 $Q_{i,y}$ 是给定 $X=x_i$ 时 $Y=y$ 的分布概率。因此，EM 算法在给定 Q 求解 $\boldsymbol{\theta}$ 和给定 $\boldsymbol{\theta}$ 求解 Q 之间交替。从形式上看，EM 算法找到解的序列 $(Q^{(1)}, \boldsymbol{\theta}^{(1)}), (Q^{(2)}, \boldsymbol{\theta}^{(2)}), \cdots$，其中在 t 次迭代时，我们通过如下两个步骤构造 $(Q^{(t+1)}, \boldsymbol{\theta}^{(t+1)})$。 [302]

- 期望步骤（E Step）：令

$$Q_{i,y}^{(t+1)} = \mathcal{P}_{\theta^{(t)}}[Y=y \mid X=x_i] \tag{24.10}$$

  这个步骤被称为期望步骤（expectation step），因为它产生隐变量的一个新概率分布，从而定义了关于 $\boldsymbol{\theta}$ 的一个新的预期似然函数。

- 极大步骤（M Step）：令 $\boldsymbol{\theta}^{(t+1)}$ 是预期似然函数的极大值点，这里的期望是关于 $Q^{(t+1)}$ 的：

$$\boldsymbol{\theta}^{(t+1)} = \operatorname*{argmax}_{\theta} F(Q^{(t+1)}, \boldsymbol{\theta}) \tag{24.11}$$

由我们的假设，可以有效地解决这个最优化问题。

$\boldsymbol{\theta}$ 和 Q 的初始值 $\boldsymbol{\theta}^{(1)}$ 和 $Q^{(1)}$ 通常是随机选取的，并且当样本集的对数似然度不再显著增加时，算法的迭代过程终止。

24.4.1　EM 是交替最大化算法

为分析 EM 算法，我们首先把它看作一个交替最大化算法。定义如下目标函数

$$G(Q,\boldsymbol{\theta}) = F(Q,\boldsymbol{\theta}) - \sum_{i=1}^{m}\sum_{y=1}^{k} Q_{i,y}\log(Q_{i,y})$$

其中，第二项是 Q 的每一行的熵的总和。令

$$\mathcal{Q} = \left\{Q \in [0,1]^{m\times k} : \forall i, \sum_{y=1}^{k} Q_{i,y} = 1\right\}$$

是定义集合 $[k]$ 上的概率分布的矩阵的全体。下面的引理表明，EM 对 G 交替进行最大化迭代。

引理 24.2　EM 迭代过程可以重写为

$$Q^{(t+1)} = \operatorname*{argmax}_{Q\in\mathcal{Q}} G(Q,\boldsymbol{\theta}^{(t)})$$

$$\boldsymbol{\theta}^{(t+1)} = \operatorname*{argmax}_{\boldsymbol{\theta}} G(Q^{(t+1)},\boldsymbol{\theta})$$

进一步，$G(Q^{(t+1)},\boldsymbol{\theta}^{(t)}) = L(\boldsymbol{\theta}^{(t)})$。

证明　给定 $Q^{(t+1)}$，显然有

[333]

$$\operatorname*{argmax}_{\boldsymbol{\theta}} G(Q^{(t+1)},\boldsymbol{\theta}) = \operatorname*{argmax}_{\boldsymbol{\theta}} F(Q^{(t+1)},\boldsymbol{\theta})$$

因此，我们只需要表明，对于任何 $\boldsymbol{\theta}$，$\operatorname*{argmax}_{Q\in\mathcal{Q}} G(Q,\boldsymbol{\theta})$ 的解是 $Q_{i,y} = \mathcal{P}_{\boldsymbol{\theta}}[Y=y \mid X=x_i]$。事实上，由詹生不等式，对任何 $Q\in\mathcal{Q}$，我们有

$$\begin{aligned}
G(Q,\boldsymbol{\theta}) &= \sum_{i=1}^{m}\left(\sum_{y=1}^{k} Q_{i,y}\log\left(\frac{\mathcal{P}_{\boldsymbol{\theta}}[X=x_i, Y=y]}{Q_{i,y}}\right)\right)\\
&\leqslant \sum_{i=1}^{m}\left(\log\left(\sum_{y=1}^{k} Q_{i,y}\frac{\mathcal{P}_{\boldsymbol{\theta}}[X=x_i, Y=y]}{Q_{i,y}}\right)\right)\\
&= \sum_{i=1}^{m}\log\left(\sum_{y=1}^{k}\mathcal{P}_{\boldsymbol{\theta}}[X=x_i, Y=y]\right)\\
&= \sum_{i=1}^{m}(\log\mathcal{P}_{\boldsymbol{\theta}}[X=x_i]) = L(\boldsymbol{\theta})
\end{aligned}$$

而对于 $Q_{i,y} = \mathcal{P}_{\boldsymbol{\theta}}[Y=y \mid X=x_i]$，我们有

$$\begin{aligned}
G(Q,\boldsymbol{\theta}) &= \sum_{i=1}^{m}\left(\sum_{y=1}^{k}\mathcal{P}_{\boldsymbol{\theta}}[Y=y \mid X=x_i]\log\left(\frac{\mathcal{P}_{\boldsymbol{\theta}}[X=x_i, Y=y]}{\mathcal{P}_{\boldsymbol{\theta}}[Y=y \mid X=x_i]}\right)\right)\\
&= \sum_{i=1}^{m}\sum_{y=1}^{k}\mathcal{P}_{\boldsymbol{\theta}}[Y=y \mid X=x_i]\log(\mathcal{P}_{\boldsymbol{\theta}}[X=x_i])\\
&= \sum_{i=1}^{m}\log(\mathcal{P}_{\boldsymbol{\theta}}[X=x_i])\sum_{y=1}^{k}\mathcal{P}_{\boldsymbol{\theta}}[Y=y \mid X=x_i]\\
&= \sum_{i=1}^{m}\log(\mathcal{P}_{\boldsymbol{\theta}}[X=x_i]) = L(\boldsymbol{\theta})
\end{aligned}$$

这表明 $Q_{i,y} = \mathcal{P}_{\boldsymbol{\theta}}[Y=y \mid X=x_i]$ 是 $G(Q,\boldsymbol{\theta})$ 在 $Q\in\mathcal{Q}$ 上的极大值点，同时也表明 $G(Q^{(t+1)},$

$\boldsymbol{\theta}^{(t)}) = L(\boldsymbol{\theta}^{(t)})$。 ■

前面的引理直接表明：

定理 24.3 在 EM 算法中，似然函数是单调递增的；也就是，对任意 t

$$L(\boldsymbol{\theta}^{(t+1)}) \geqslant L(\boldsymbol{\theta}^{(t)})$$

证明 由前面的引理有

$$L(\boldsymbol{\theta}^{(t+1)}) = G(Q^{(t+1)}, \boldsymbol{\theta}^{(t+1)}) \geqslant G(Q^{(t+1)}, \boldsymbol{\theta}^{(t)}) = L(\boldsymbol{\theta}^{(t)})$$ ■

24.4.2 混合高斯模型参数估计的 EM 算法

考虑由 k 个高斯分布构成的高斯混合分布，参数 $\boldsymbol{\theta}$ 是三元组 $(\boldsymbol{c}, \{\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_k\}, \{\Sigma_1, \cdots, \Sigma_k\})$，其中 $\mathcal{P}_{\boldsymbol{\theta}}[Y=y]=c_y$ 且 $\mathcal{P}_{\boldsymbol{\theta}}[X=\boldsymbol{x} \mid Y=y]$ 如公式(24.9)所示。为简便，我们假设 $\Sigma_1 = \Sigma_2 = \cdots = \Sigma_k = I$，其中 I 为单位矩阵。在这样的情况下，EM 算法如下所示： 304

E 步：对每个 $i \in [m]$ 和 $y \in [k]$，我们有

$$\begin{aligned}\mathcal{P}_{\boldsymbol{\theta}^{(t)}}[Y=y \mid X=\boldsymbol{x}_i] &= \frac{1}{Z_i}\mathcal{P}_{\boldsymbol{\theta}^{(t)}}[Y=y]\,\mathcal{P}_{\boldsymbol{\theta}^{(t)}}[X=\boldsymbol{x}_i \mid Y=y] \\ &= \frac{1}{Z_i}c_y^{(t)}\exp\left(-\frac{1}{2}\|\boldsymbol{x}_i-\boldsymbol{\mu}_y^{(t)}\|^2\right)\end{aligned} \tag{24.12}$$

其中 Z_i 是归一化因子，使得求和式 $\sum_y \mathcal{P}_{\boldsymbol{\theta}^{(t)}}[Y=y \mid X=\boldsymbol{x}_i]$ 为 1。

M 步：我们需要取公式(24.11)的最大值点 $\boldsymbol{\theta}^{(t+1)}$，也就是使得下式最大化的参数 $\boldsymbol{c}$ 和 $\boldsymbol{\mu}$：

$$\sum_{i=1}^{m}\sum_{y=1}^{k}\mathcal{P}_{\boldsymbol{\theta}^{(t)}}[Y=y \mid X=\boldsymbol{x}_i]\left(\log(c_y)-\frac{1}{2}\|\boldsymbol{x}_i-\boldsymbol{\mu}_y\|^2\right) \tag{24.13}$$

置公式(24.13)关于 $\boldsymbol{\mu}_y$ 的偏导数为 0，经整理得到：

$$\boldsymbol{\mu}_y = \sum_{i=1}^{m}\mathcal{P}_{\boldsymbol{\theta}^{(t)}}[Y=y \mid X=\boldsymbol{x}_i]\,\boldsymbol{x}_i$$

也就是，$\boldsymbol{\mu}_y$ 是 $\boldsymbol{x}_i$ 的加权平均值，其中权重为 E 步中得到的后验分布。为找到最优的 $\boldsymbol{c}$，我们必须仔细以保证 $\boldsymbol{c}$ 是一个概率分布向量。在练习 24.3 中，我们证明了最优解为

$$c_y = \frac{\sum_{i=1}^{m}\mathcal{P}_{\boldsymbol{\theta}^{(t)}}[Y=y \mid X=\boldsymbol{x}_i]}{\sum_{y'=1}^{k}\sum_{i=1}^{m}\mathcal{P}_{\boldsymbol{\theta}^{(t)}}[Y=y' \mid X=\boldsymbol{x}_i]} \tag{24.14}$$

将上述算法和第 22 章中的 k 均值算法相比较，是很有意思的。在 k 均值算法中，我们首先根据距离 $|\boldsymbol{x}_i-\boldsymbol{\mu}_y|$ 将每个样本点分配到某一个类中，然后，我们将类中心 $\boldsymbol{\mu}_y$ 更新为分配给该类的样本的平均值。然而，在 EM 算法中，我们首先确定每个样本属于每个类的概率，然后，我们将类中心更新为所有样本的加权平均值。出于这个原因，借助 EM 算法的 k 均值方法有时被称为"soft k-means"。

24.5 贝叶斯推理

最大似然估计是遵循频率论的一种方法，这意味着我们假定参数 θ 为固定参数，只是不知道它的取值。还有一种参数估计方法是贝叶斯推理。在贝叶斯方法中，关于参数 θ 的不确定性也用概率来描述，也就是说，我们认为参数 θ 也是随机变量，其先验分布为 $\mathcal{P}[\theta]$。正如它的名字所表示的，先验分布应该由学习者在观测到数据之前就确定。 305

作为一个例子，让我们再来考虑研发了一种新药物的制药公司的例子。根据过去的经验，制药公司的统计学家认为，当一种药物已到了临床实验的阶段，它应该是比较有效的。他们将关于θ的先验信念定义为如下的分布：

$$\mathcal{P}[\theta]=\begin{cases}0.8 & 若\ \theta>0.5\\ 0.2 & 若\ \theta\leqslant 0.5\end{cases} \tag{24.15}$$

和前面一样，给定θ的取值时，假定条件概率$\mathcal{P}[X=x|\theta]$是已知的。在制药公司的例子中，X取值于$\{0,1\}$并且$\mathcal{P}[X=x|\theta]=\theta^{x}(1-\theta)^{1-x}$。

一旦确定了参数θ的先验分布和给定θ时X的条件分布，我们就得到了关于X分布的全部知识。这是因为我们可以将X分布表示为边缘概率

$$\mathcal{P}[X=x]=\sum_{\theta}\mathcal{P}[X=x,\theta]=\sum_{\theta}\mathcal{P}[\theta]\mathcal{P}[X=x|\theta]$$

其中最后一个等式是根据条件概率的定义得到。如果参数θ是连续的，我们就可以将$\mathcal{P}[\theta]$替换为密度函数，并且用积分代替求和

$$\mathcal{P}[X=x]=\int_{\theta}\mathcal{P}[\theta]\mathcal{P}[X=x|\theta]\mathrm{d}\theta$$

表面上看，一旦我们知道$\mathcal{P}[X=x]$，训练集$S=(x_1,\cdots,x_m)$并不能给我们带来任何新的知识，因为我们已经是知道新的样本X的分布的专家了。然而，贝叶斯观点引入S和X之间的依赖关系，这是因为我们将参数θ看作随机变量。一个新样本点X和前面训练集S关于参数θ是条件独立的。这是与频率派不同的，频率派认为θ仅仅是分布的参数，而新样本点X和前面训练集S始终是独立的。

在贝叶斯框架中，由于X和S不再独立，我们想计算的是给定S时X的概率，由链式法则，可以写为如下：

$$\mathcal{P}[X=x|S]=\sum_{\theta}\mathcal{P}[X=x|\theta,S]\mathcal{P}[\theta|S]=\sum_{\theta}\mathcal{P}[X=x|\theta]\mathcal{P}[\theta|S]$$

第二个等式成立的原因是：X和S关于参数θ是条件独立的。由贝叶斯准则，我们有

$$\mathcal{P}[\theta|S]=\frac{\mathcal{P}[S|\theta]\mathcal{P}[\theta]}{\mathcal{P}[S]}$$

根据假设，所有的样本关于θ是条件独立的，我们有

306

$$\mathcal{P}[\theta|S]=\frac{\mathcal{P}[S|\theta]\mathcal{P}[\theta]}{\mathcal{P}[S]}=\frac{1}{\mathcal{P}[S]}\prod_{i=1}^{m}\mathcal{P}[X=x_i|\theta]\mathcal{P}[\theta]$$

因此得到如下的贝叶斯预测：

$$\mathcal{P}[X=x|S]=\frac{1}{\mathcal{P}[S]}\sum_{\theta}\mathcal{P}[X=x|\theta]\prod_{i=1}^{m}\mathcal{P}[X=x_i|\theta]\mathcal{P}[\theta] \tag{24.16}$$

回到制药公司的例子中，我们可以将$\mathcal{P}[X=x|S]$重写为

$$\mathcal{P}[X=x|S]=\frac{1}{\mathcal{P}[S]}\int\theta^{x+\sum_i x_i}(1-\theta)^{1-x+\sum_i(1-x_i)}\mathcal{P}[\theta]\mathrm{d}\theta$$

有趣的是，如果$\mathcal{P}[\theta]$是均匀分布的，我们有

$$\mathcal{P}[X=x|S]\propto\int\theta^{x+\sum_i x_i}(1-\theta)^{1-x+\sum_i(1-x_i)}\mathrm{d}\theta$$

求解上面的积分（分布积分法）我们有

$$\mathcal{P}[X=1|S]=\frac{(\sum_i x_i)+1}{m+2}$$

回忆当使用极大似然准则时，得到的预测为 $\mathcal{P}[X=1|\hat{\theta}]=\frac{\sum_i x_i}{m}$。而在均匀分布的先验下，得到的贝叶斯预测与极大似然预测是十分相似的，区别是它加入了“伪例”，利用均匀的先验对预测进行了调整。

最大后验法

在许多情况下，对于等式(24.16)所给出的积分，很难找到其封闭形式的解。有很多数值方法可用于近似此积分。另一种流行的解决办法是寻找一个使得 $\mathcal{P}[\theta|S]$ 最大的 θ 值。使得 $\mathcal{P}[\theta|S]$ 最大的于 θ 值被称为最大后验估计。一旦这确定了最大后验分布值，我们根据 X 和 S 的条件独立性计算出 $X=x$ 的概率。

24.6 小结

在机器学习的生成方法中，我们的目标是模拟数据的分布。特别是，在参数密度估计中，进一步假设数据的潜在分配具有特定的参数形式，我们的目标是估计分布的参数。我们已经描述了几个参数估计准则，包括最大似然、贝叶斯估计和最大后验。我们还描述了几个针对潜在分布的不同的假设下的极大似然估计的具体算法，具体有朴素贝叶斯、线性判别分析和 EM 算法。

24.7 文献评注

20 世纪初，统计学家 Ronald Fisher 开始研究极大似然准则。贝叶斯学派源于贝叶斯准则，是以 18 世纪的英国数学家 Thomas Bayes 命名的。 337

关于机器学习中的生成学习和贝叶斯方法，有很多优秀的专著，如 Bishop(2006)，Koller & Friedman(2009a)，MacKay(2003)，Murphy(2012)，Barber(2012)。

24.8 练习

24.1 证明：关于高斯变量的方差的极大似然估计是有偏的。

24.2 极大似然估计的正则化：考察下面的正则损失

$$\frac{1}{m}\sum_{i=1}^{m}\log(1/\mathcal{P}_\theta[x_i])+\frac{1}{m}(\log(1/\theta)+\log(1/(1-\theta)))$$

- 证明上述的优化目标等价于将训练集中加入两个“伪例”。验证上述的正则的极大似然估计的解为

$$\hat{\theta}=\frac{1}{m+2}\left(1+\sum_{i=1}^{m}x_i\right)$$

- 推导 $|\hat{\theta}-\theta^*|$ 的概率下界。提示：将此式重写为 $|\hat{\theta}-\mathbb{E}(\hat{\theta})+\mathbb{E}(\hat{\theta})-\theta^*|$ 并利用三角不等式和 Hoeffding 不等式。
- 用此概率界来估计真实损失。提示：借助事实 $\hat{\theta}\geq\frac{1}{m+2}$ 将 $|\hat{\theta}-\theta^*|$ 与相对熵联系起来。

24.3 考虑具有如下形式的一般的优化问题：

$$\max_c \sum_{y=1}^{k} v_y \log(c_y) \quad \text{s.t.} \quad c_y \geq 0, \sum_y c_y = 1$$

其中 $v\in\mathbb{R}^k$ 是非负权重向量。

- 验证 soft k-means 中的 M 步中求解了这一优化问题。
- 令 $c^* = \frac{1}{\sum_y \nu_y} v$，证明 c^* 是一个概率向量。
- 证明这个优化问题等价于

$$\min_c D_{KL}(c^* \| c) \text{ s.t. } c_y > 0, \sum_y c_y = 1$$

308

- 利用相对熵的性质，验证 c^* 是优化问题的最优解。

特征选择与特征生成

在本书的开头，我们讨论了学习的抽象模型，在这个抽象模型中，完全是通过对假设类的选择来编码利用先验知识，然而，还有另一种目前没有讨论的模型选择方法：如何表示样本空间$\mathcal{X}$？例如，在木瓜学习问题中，我们提出了光滑性和颜色两个维度的矩阵假设类，也就是说，第一个假设就是，在完成从平面到标签集映射的矩形假设类之后，就可以用二维平面上的点来对应表示木瓜的光滑性和颜色。这种从现实世界的木瓜到用数值来表示它的光滑性和颜色的变换，我们称之为特征函数，简称特征。也就是说，任何对现实世界物体的度量可以认为是一种特征。如果$\mathcal{X}$是向量空间的一个子集，每一个$x\in\mathcal{X}$有时候称之为特征向量。理解我们如何利用问题相关的先验知识，将现实世界物体编码为输入空间$\mathcal{X}$的方式是非常重要的。

更深入地说，即使已经将输入空间$\mathcal{X}$表示成一个向量空间的子集，我们可能还是想改变它的表示形式，然后在新的表示形式上应用假设类。也就是说，我们可能定义一个$\mathcal{X}$的假设类，通过在某种能将向量空间$\mathcal{X}$映射为$\mathcal{X}'$的特征函数上定义假设类来实现。我们已经碰到过这样的例子，在第 15 章中，可以看到基于核的支持向量机算法，通过在源空间样本$\mathcal{X}$到希尔伯特空间的特征映射ψ上学习每一个两类分类器来实现。确实，选择ψ是另一种应用先验知识来处理问题的方法。

在这一章中，我们研究构建一个好的特征集的几种方法。首先讨论特征选择问题，特征选择就是从特征池大量特征中选择少量用于构建预测器的特征。然后，讨论针对特征的操作和特征归一化，这些特征变换将降低学习算法的样本复杂度，预测器的偏置以及计算复杂度。最后，我们讨论几种特征学习的方法，在这几种方法中，我们尝试自动完成特征构建的过程。

我们强调，虽然有很多可以尝试的共同的特征学习方法，但是“没有免费的午餐”理论指出不存在一种能处理所有问题的极端特征学习器，任何特征学习器都有可能在某些问题上失败，换句话说，每一个特征学习器的成功依赖于数据分布的某种先验假设形式(有时候这种先验假设可能不那么明显)。进一步说，特征的质量高度依赖我们后期所使用的学习算法。下面的例子给出了说明。

例 25.1 考虑一个回归问题，$\mathcal{X}=\mathbb{R}^2$，$\mathcal{Y}=\mathbb{R}$，损失函数使用平方损失。给出样本的潜在分布，样本(x, y)是这样产生的：首先，从$[-1, 1]$的均匀分布上采样x_1，然后，确切地令$y=x_1^2$。最后，第二个特征集是$x_2=y-x$，这里x是从$[-0.01, 0.01]$的均匀分布上采样得到。假定我们只想选用一个特征。直观上，单独使用第一个特征要比单独使用第二个特征更好。的确，如果我们应用次数大于 2 的多项式回归方法，那么第一个特征是正确的选择，然而，如果使用线性回归器，我们应当更倾向于选择第二个特征，这是因为：对于最优的线性分类器，使用第一个特征的风险大于使用第二个特征的风险。 ◀

25.1 特征选择

在这一节中，我们假定$\mathcal{X}=\mathbb{R}^d$。也就是说，每一个样本可以用d个特征的向量表示。

我们的目标是学习一个仅依赖于 k 个特征的预测器，k 远小于 d。使用少量的特征的预测器要求更少的内存空间，并且计算速度更快。并且，在像医疗诊断这样的应用中，获取可能的特征（例如测试结果）费用昂贵。因此，即使在性能上与使用大量特征相比有些退化，使用少量特征的预测器也是有需求的。最后，约束假设类使用少量特征构成的子集能降低估计误差，防止过拟合。

理想情况下，我们尝试 d 个特征的所有 k 个特征组合，然后选择最优预测结果对应的特征子集。然而，这种穷尽搜索方法通常在计算上是不可行的。接下来，我们描述三种计算可行的特征选择方法。虽然这些方法不能保证一定找到最优的特征子集，但是它们通常都能在实践中取得相当好的结果。一些特征选择方法还可以在某些假设条件下形式化保证特征选择子集的质量。我们在这里不讨论这些质量保证的内容。

25.1.1　滤波器

滤波方法可能是最简单的特征方法，在滤波方法中，我们将某些特征看做独立于其他特征，然后根据一些质量度量标准来估计这些独立特征。我们选择 k 个获得最高评分的特征（此外，也可以依据最好的评分确定特征的数量）。

[312]

文献中已经提出了很多质量评价的方法。最直接的方法可能是依据预测器的错误率来获得特征的评分，这些预测器是通过待评估特征单独训练而得到的。

为了说明这个，我们考虑采用平方损失的线性回归问题。令 $v=(x_{1,j},\ldots,x_{m,j})\in\mathbb{R}^m$ 表示 m 个训练样本第 j 个特征值形成的向量，令 $y=(y_1,\ldots,y_m)\in\mathbb{R}^m$ 表示 m 个样本的目标值。仅使用第 j 个特征的经验风险最小化线性预测器的经验平方损失是

$$\min_{a,b\in\mathbb{R}}\frac{1}{m}\|av+b-y\|^2$$

在这里加 b 的含义是：v 的所有维度的值都加上 b。为了求解这个最小化问题，令 $\bar{v}=\frac{1}{m}\sum_{i=1}^{m}v_i$ 表示特征的平均值，令 $\bar{y}=\frac{1}{m}\sum_{i=1}^{m}y_i$ 表示目标的平均值。显然（见练习 25.1），

$$\min_{a,b\in\mathbb{R}}\frac{1}{m}\|av+b-y\|^2=\min_{a,b\in\mathbb{R}}\frac{1}{m}\|a(v-\bar{v})+b-(y-\bar{y})\|^2 \tag{25.1}$$

等式右边对 b 求导，令导数等于 0，我们得到 $b=0$。同样，对 a 求导，当 $b=0$ 时，我们得到 $a=\langle v-\bar{v},y-\bar{y}\rangle/\|v-\bar{v}\|^2$，将 a，b 的值代入目标函数，我们得到

$$\|y-\bar{y}\|^2-\frac{(\langle v-\bar{v},y-\bar{y}\rangle)^2}{\|v-\bar{v}\|^2}$$

依据最小平方损失对特征排序，等同于依据下面评分的绝对值进行排序（这里高分表示好的特征）：

$$\frac{\langle v-\bar{v},y-\bar{y}\rangle}{\|v-\bar{v}\|\ \|y-\bar{y}\|}=\frac{\frac{1}{m}\langle v-\bar{v},y-\bar{y}\rangle}{\sqrt{\frac{1}{m}\|v-\bar{v}\|^2}\sqrt{\frac{1}{m}\|y-\bar{y}\|^2}} \tag{25.2}$$

上面的表达式被称为皮尔森相关系数。分子表示第 j 个特征和目标值方差（即 $\mathbb{E}[(v-\mathbb{E}v)(y-\mathbb{E}y)]$）的经验估计，而分母表示第 j 个特征方差（$\mathbb{E}[(v-\mathbb{E}v)^2]$）乘上目标值所得方差经验估计的均方根。皮尔森相关系数的取值范围为 -1 到 1，这里如果皮尔森相关系数等于 1 或 -1，表示 v 和 y 之间有线性映射关系，且经验风险等于 0。

如果皮尔森相关系数等于 0，表示 v 到 y 的最优线性映射为各个维度都等于 0，这就

是说单独只用 v 不足以预测 y。但是这并不意味着 v 是一个坏的特征，比如可能出现这种情况，v 和其他特征组合起来能很好地预测 y。的确，考虑一个简单的例子，目标通过函数 $y=x_1+2x_2$ 来产生。假定 x_1 是由 $\{\pm 1\}$ 上的均匀分布产生，而 $x_2=-\frac{1}{2}x_1+\frac{1}{2}z$，这里 z 也是由 $\{\pm 1\}$ 上的均匀分布产生。那么，$\mathbb{E}[x_1]=\mathbb{E}[x_2]=\mathbb{E}[y]=0$，我们可以得到 311

$$\mathbb{E}[yx_1]=\mathbb{E}[x_1^2]+2\mathbb{E}[x_2x_1]=\mathbb{E}[x_1^2]-\mathbb{E}[x_1^2]+\mathbb{E}[zx_1]=0$$

因此，对于足够大的训练集，第一个特征的皮尔森相关系数很可能等于 0，因此它很可能不被选择。然而，如果不知道第一个特征，没有函数能够很好地预测目标值。

还有很多其他的评分函数可以用于滤波方法。著名的评分函数的例子是互信息估计或者接受操作特征(ROC)曲线的面积。所有这些评分函数都受先前说明例子类似的制约。我们推荐读者阅读 Guyon 和 Elisseeff(2003)。

25.1.2 贪婪选择方法

贪婪选择是另一个盛行的特征选择方法。和滤波方法不同，贪婪选择方法伴随着学习算法。最简单的贪婪选择的例子是前向贪婪选择方法。我们从一个空集开始，然后逐步每次添加一个特征到选择的特征集。给定当前选择的特征集 I，我们遍历所有的 $i\notin I$，然后在 $I\cup\{i\}$ 特征集上应用学习算法。每一个这样的应用取得一个不同的预测器，我们选择添加特征使得预测器的风险最小(在训练集或者验证集)。持续这个过程直到我们选择了 k 个特征，这里 k 表示预先定义的可以承担的特征数，或者得到一个足够精度的预测器。

例 25.2 (正交匹配追踪) 为了说明前向贪婪选择方法，我们具体化到使用平方损失的线性回归问题。令 $X\in\mathbb{R}^{m\times d}$ 是一个有 m 个训练样本行的矩阵。令 $y\in\mathbb{R}^m$ 表示 m 个标签构成的向量。对于每一个 $i\in[d]$，令 X_i 表示 X 的第 i 列。给定一个集合 $I\subset[d]$，我们用 X_I 表示列为 $\{X_i:i\in I\}$ 的矩阵。

前向贪婪选择方法从 $I_0=\varnothing$ 开始，在第 t 次迭代，我们寻找第 j_t 个特征，使得

$$\arg\min_j \min_{w\in\mathbb{R}^t}|X_{I_{t-1}\cup\{j\}}w-y|^2$$

那么，我们更新 $I_t=I_{t-1}\cup\{j_t\}$。 ◀

现在，我们描述一种针对线性回归问题更加高效的执行前向贪婪选择的方法，称之为正交匹配追踪。保持正交的想法基于到目前为止的特征集合。令 V_t 表示基于 X_{I_t} 的列正交形成的矩阵。显然，

$$\min_w|X_{I_t}w-y|^2=\min_{\theta\in\mathbb{R}^t}|V_t\theta-y|^2$$

我们将证明一个 θ_t 使得等式右边最小化。

首先，令 $I_0=\varnothing$，$V_0=\varnothing$，θ_0 表示空向量。在 t 个循环，对于每一个 j，我们分解 $X_j=v_j+u_j$，这里 $v_j=V_{t-1}V_{t-1}^TX_j$ 是 X_j 在 V_{t-1} 张成的子空间上的投影，u_j 表示 X_j 与 V_{t-1} 正交的部分。那么， 312

$$\begin{aligned}
&\min_{\theta,\alpha}|V_{t-1}\theta+\alpha u_j-y|^2\\
&=\min_{\theta,\alpha}[|V_{t-1}\theta-y|^2+\alpha^2|u_j|^2+2\alpha\langle u_j,V_{t-1}\theta-y\rangle]\\
&=\min_{\theta,\alpha}[|V_{t-1}\theta-y|^2+\alpha^2|u_j|^2+2\alpha\langle u_j,-y\rangle]\\
&=\min_{\theta}[|V_{t-1}\theta-y|^2]+\min_{\alpha}[\alpha^2|u_j|^2-2\alpha\langle u_j,y\rangle]
\end{aligned}$$

$$=[\|V_{t-1}\boldsymbol{\theta}_{t-1}-\boldsymbol{y}\|^2]+\min_{\alpha}[\alpha^2\|\boldsymbol{u}_j\|^2-2\alpha\langle\boldsymbol{u}_j,\boldsymbol{y}\rangle]$$

$$=\|V_{t-1}\boldsymbol{\theta}_{t-1}-\boldsymbol{y}\|^2-\frac{(\langle\boldsymbol{u}_j,\boldsymbol{y}\rangle)^2}{\|\boldsymbol{u}_j\|^2}$$

也就是，我们应当选择特征，使得

$$j_t=\operatorname*{argmax}_j\frac{(\langle\boldsymbol{u}_j,\boldsymbol{y}\rangle)^2}{\|\boldsymbol{u}_j\|^2}$$

其余的更新如下：

$$V_t=\left[V_{t-1},\frac{\boldsymbol{u}_{j_t}}{\|\boldsymbol{u}_{j_t}\|^2}\right],\boldsymbol{\theta}_t=\left[\boldsymbol{\theta}_{t-1},\frac{\langle\boldsymbol{u}_{j_t},\boldsymbol{y}\rangle}{\|\boldsymbol{u}_{j_t}\|^2}\right]$$

正交匹配追踪算法过程中明了一组被选特征的正交基，通过过程描述可以看出，正交属性是通过一种类似于施密特正交化方法获得的。应用中，施密特正交化方法通常数值不稳定。在下面的伪代码中，我们使用 SVD(见 C.4 节)在每一个循环的结束之前以数值稳定方式获得一组正交基。

正交匹配追踪算法(OMP)

输入：

　数据矩阵 $X\in\mathbb{R}^{n\times d}$，标签向量 $\boldsymbol{y}\in\mathbb{R}^n$

　最大特征数量 T

初始化： $I_1=\emptyset$

对于 $t=1,\cdots,T$

　使用 SVD 方法找到 X_{I_t} 的一组正交基 $V\in\mathbb{R}^{n\times|I_t|}$

　(当 $t=1$ 时，设置 V 为零矩阵)

　对于每一个 $j\in[d]\setminus I_t$，令 $\boldsymbol{u}_j=X_j-VV^TX_j$

　令 $j_t=\operatorname{argmax}_{j\in[d]\setminus I_t}\frac{(\langle\boldsymbol{u}_j,\boldsymbol{y}\rangle)^2}{\|\boldsymbol{u}_j\|^2}$

　更新 $I_{t+1}=I_t\cup\{j_t\}$

输出： I_{T+1}

313

1. 更高效的贪婪选择准则

令 $R(\boldsymbol{w})$ 表示向量 $\boldsymbol{w}$ 的经验风险。在前向贪婪选择方法的每一个循环，对于每一个可能的 j，我们应当最小化 $R(\boldsymbol{w})$ 关于由 $I_{t-1}\cup\{j\}$ 支持的 $\boldsymbol{w}$，这种方法可能计算上很费时。

一个简单的近似方法是选择 j_t 最小化下面的式子：

$$\operatorname*{argmin}_j\min_{\eta\in\mathbb{R}}R(\boldsymbol{w}_{t-1}+\eta\boldsymbol{e}_j)$$

这里，$\boldsymbol{e}_j$ 表示除第 j 个元素为 1 之外，其他元素都为 0 的向量。也就是说，我们保持先前选择的维度不变，仅仅优化新加入的变量。因此，对于每一个 j 我们需要求解关于一个单变量的优化问题，这会比优化 t 容易得多。

一个更加简单的方法是由一个简单函数表示上界，选择新加入特征使得上界有最大的数值下降。例如，如果 R 是 β-光滑函数(见 12 章方程(12.5))，那么

$$R(\boldsymbol{w}+\eta\boldsymbol{e}_j)\leqslant R(\boldsymbol{w})+\eta\frac{\partial R(\boldsymbol{w})}{\partial w_j}+\beta\eta^2/2$$

关于 η 最小化不等式的右边，可得 $\eta=-\frac{\partial R(\boldsymbol{w})}{\partial w_j}\cdot\frac{1}{\beta}$，将 η 代入不等式，可得

$$R(w+\eta e_j) \leqslant R(w) - \frac{1}{2\beta}\left(\frac{\partial R(w)}{\partial w_j}\right)^2$$

当 $R(w)$ 关于 w_j 的偏导数取得最大时，上式取得最小值。因此，我们可以选择 j，使得 $R(w)$ 关于 w 的梯度最大的维度。

评注(AdaBoost 作为一个前向贪婪选择过程) 有可能将第 10 章介绍的 AdaBoost 算法解析为一种关于以下函数的前向贪婪选择过程：

$$R(w) = \log\left(\sum_{i=1}^{m}\exp\left(-y_i\sum_{j=1}^{d}w_j h_j(x_i)\right)\right) \tag{25.3}$$

见练习 25.3。

2. 反向终止算法

另一个盛行的贪婪选择方法是反向终止算法。这里，我们从全部特征组成的集合开始，然后逐步从特征集合中一次减少一个特征。给定我们当前选择的特征集 $I \setminus \{i\}$，每一个这样的做法取得一个不同的预测器，然后我们选择去掉特征 i 使得预测器从特征集 $I \setminus \{i\}$ 得到最小的风险(在训练集或者验证集上)。

本质上，有很多可能的反向终止算法的变形方法，结合前向贪婪和反向贪婪步骤也是有可能做到的。

314

25.1.3 稀疏诱导范数

最大特征数量为 k 的经验风险最小化问题可以写成

$$\min_{w} L_S(w) \quad \text{s.t.} \quad \|w\|_0 \leqslant k$$

这里，[⊖]

$$\|w\|_0 = |\{i: w_i \neq 0\}|$$

换句话说，我们希望 w 是稀疏的，也就是只需要特征权重 w 不为 0 的特征。

求解这个优化问题是计算困难的(Natarajan 1995，Davis，Mallat & Avellaneda 1997)。一个可能的松弛方法是使用 ℓ_1 范数替代 $\|w\|_0$，$\|w\|_1 = \sum_{i=1}^{d}|w_i|$，然后求解下面的问题：

$$\min_{w} L_S(w) \quad \text{s.t.} \quad \|w\|_1 \leqslant k_1 \tag{25.4}$$

这里 k_1 是一个参数。由于 ℓ_1 范数是凸函数，只要损失函数是凸的，这个问题能够高效地求解。另一个相关的问题是最小化 $L_S(w)$ 和 ℓ_1 范数正则项之和，

$$\min_{w}(L_S(w) + \lambda\|w\|_1) \tag{25.5}$$

这里 λ 是正则项参数。由于对于任意的 k_1 存在 λ 使得方程(25.4)与方程(25.5)取得同样的优化结果，在某种意义上说这两个问题是等价的。

ℓ_1 正则项通常导致稀疏的优化结果。为了说明这一点，让我们处理如下简单的优化问题，

$$\min_{w\in\mathbb{R}}\left(\frac{1}{2}w^2 - xw + \lambda|w|\right) \tag{25.6}$$

容易验证(见练习 25.2)，可以使用软阈值方法优化这个问题。

⊖ 函数 $\|\cdot\|_0$ 通常称呼是 ℓ_0 范数，尽管使用范数描述，$\|\cdot\|_0$ 不是真正的范数。例如，它不满足范数的正同质属性，$\|aw\|_0 \neq |a|\,\|w\|_0$。

$$w = \mathrm{sign}(x)\,[\,|x| - \lambda\,]_+ \tag{25.7}$$

这里$[a]_+ \stackrel{\text{def}}{=} \max(a, 0)$。也就是说，只要$x$的绝对值小于$\lambda$，最优结果将会使得$w$的值置为0。

然后，考虑使用平方损失的一维回归问题：

$$\underset{w\in\mathbb{R}}{\mathrm{argmin}}\left(\frac{1}{2m}\sum_{i=1}^{m}(x_i w - y_i)^2 + \lambda|w|\right)$$

我们可以重写这个问题为

[315]

$$\underset{w\in\mathbb{R}}{\mathrm{argmin}}\left(\frac{1}{2}\left(\frac{1}{m}\sum_i x_i^2\right)w^2 - \left(\frac{1}{m}\sum_{i=1}^{m} x_i y_i\right)w + \lambda|w|\right)$$

为了简化，我们假定$\frac{1}{m}\sum_i x_i^2 = 1$，表示$\langle \boldsymbol{x}, \boldsymbol{y}\rangle = \sum_{i=1}^{m} x_i y_i$，那么最优解为

$$w = \mathrm{sign}(\langle \boldsymbol{x}, \boldsymbol{y}\rangle)\,[\,|\langle \boldsymbol{x}, \boldsymbol{y}\rangle|/m - \lambda\,]_+$$

也就是说，结果将等于0，除非特征$\boldsymbol{x}$和标签向量$\boldsymbol{y}$的相关系数大于λ。

评注 和ℓ_1范数不同，ℓ_2范数不会导致稀疏解。的确，考虑下面的ℓ_2范数问题：

$$\underset{w\in\mathbb{R}}{\mathrm{argmin}}\left(\frac{1}{2m}\sum_{i=1}^{m}(x_i w - y_i)^2 + \lambda w^2\right)$$

那么，最优解为

$$w = \frac{\langle \boldsymbol{x}, \boldsymbol{y}\rangle/m}{|\boldsymbol{x}|^2/m + 2\lambda}$$

这个解将不会等于0，即使$\boldsymbol{x}$和$\boldsymbol{y}$的相关系数很小。相比而言，和我们先前描述的一样，当使用ℓ_1范数，只有在$\boldsymbol{x}$和$\boldsymbol{y}$的相关系数大于λ时，w才会非0。

$$\underset{w}{\mathrm{argmin}}\left(\frac{1}{2m}|Xw - \boldsymbol{y}|^2 + \lambda|w|_1\right) \tag{25.8}$$

在关于分布和正则项参数λ的一些假设之下，LASSO将会得到系数解(参见，例如Zhao & Yu(2006)和其中的参考文献)。ℓ_1范数的另一个优势就是使用ℓ_1范数的向量将被稀疏化(参见，例如Shalev-Shwartz，Zhang和Srebro(2010)和其中的参考文献)。

25.2 特征操作和归一化

特征操作或归一化包括在每一个原特征上的简单变换。这些变换可能使得我们假设类的近似误差或估计误差更低或者能够得到一个更快的算法。与特征选择的问题相似，这里没有绝对好或绝对不好的变换，更可能的是每一个特征变换与在这些特征矢量上的学习算法以及这个问题相关的先验假设密切相关。

从归一化的动机，考虑使用平方损失的线性回归问题。令$X\in\mathbb{R}^{m\times d}$是一个行为样本向量的矩阵，令$\boldsymbol{y}\in\mathbb{R}^m$表示目标值向量。回想下岭回归返回的向量：

[316]

$$\underset{w}{\mathrm{argmin}}\left[\frac{1}{m}|Xw - \boldsymbol{y}|^2 + \lambda\|w\|^2\right] = (2\lambda m I + X^\top X)^{-1}X^\top \boldsymbol{y}$$

假定$d=2$，并且潜在的数据分布如下。首先，从$\{\pm 1\}$上均匀随机采样y，然后，设置$x_1 = y + 0.5\alpha$，这里α是$[-1,1]$上均匀随机采样，我们令$x_2 = 0.001y$。注意最优权值向量$w^* = [0, 10\,000]$，并且$L_{\mathcal{D}}(w^*) = 0$。然而，岭回归在w^*的目标值为$\lambda\,10^8$。相比而言，岭回归的目标函数值在$w = [1, 0]$的值可能是$0.25 + \lambda$。也就是任何时候$\lambda > \frac{0.25}{10^8 - 1} \approx 0.25\times 10^{-8}$，

岭回归的目标函数值小于子优化 $w=[1;0]$。由于典型的 λ 应当不小于 $1/m$(见第13章的分析)，在如下的例子中，如果样本数小于 10^4，那么我们很可能输出子优化的解。

上面例子的关键是两个特征有完全不同的尺度。特征归一化能够克服这个问题。有很多方法可以实现特征归一化，最简单的方法是使每一个特征得到的值都在-1到1范围内。在上面的例子中，如果我们将每一个特征除以最大值，将获得 $x_1=\frac{y+0.5\varepsilon}{1.5}$ 和 $x_2=y$，那么对于 $\lambda\leq 10^{-4}$，岭回归的解与 w^* 非常接近。

更进一步，我们在第13章对于正则最小损失获得的泛化误差界依赖于最优向量 w^* 的范数和样本向量的最大范数[⊖]。因此，对于前面的例子，在归一化特征之前，我们得到 $\|w^*\|^2=10^4$，归一化特征之后，我们得到 $\|w^*\|^2=1$。样本向量的极大范数归一化前后大致差不多。但是归一化极大改善了估计误差。

特征归一化也能改善学习算法的运行时间。例如，在14.5.3节，我们已经讨论了如何应用随机梯度下降优化算法来解决正则损失最小问题。SGD算法迭代到收敛所需的迭代次数依赖于 w^* 的范数和极大范数 $\|x\|$。因此，和之前表述的一样，使用特征归一化能大大降低SGD算法的运行时间。

接下来，我们展示一个类似于裁剪这样的简单特征变换如何降低假设类的近似误差。考虑平方损失的线性回归问题。令 $a>1$，且 a 是一个很大的数，假定目标值 y 是 $\{\pm 1\}$ 上随机均匀采样得到的，并且单个特征 x 到 y 的概率是 $(1-1/a)$，x 到 ay 的概率是 $1/a$，也就是说，大多数时候我们的特征是有界的，但是有小概率取得很大的值。那么，对于任意 w，w 的均方损失的期望是

$$
\begin{aligned}
L_{\mathcal{D}}(w) &= \mathbb{E}\,\frac{1}{2}(wx-y)^2 \\
&= \left(1-\frac{1}{a}\right)\frac{1}{2}(wy-y)^2+\frac{1}{a}\,\frac{1}{2}(awy-y)^2
\end{aligned}
$$

317

求解 w，我们得到 $w^*=\frac{2a-1}{a^2+a-1}$，当 a 趋于无穷时，w^* 趋于0。因此，当 a 趋于无穷时，在 w^* 点的目标函数值趋于0.5。例如，当 $a=100$ 时，我们得到 $L_{\mathcal{D}}(w^*)\approx 0.48$。接下来，我们假定应用一个裁剪变换。也就是说，使用变换 $x\mapsto \mathrm{sign}(x)\min\{1,|x|\}$。那么，跟随这个变换，$w^*$ 变为1，$L_{\mathcal{D}}(w^*)=0$。这个简单的例子显示一个简单的变换会对近似误差造成很大影响。

当然，不难想到这样的例子，同样的特征变换实际上损坏性能且增加近似误差。这并不奇怪，就像我们之前已经申明的特征变换应当依赖对问题的先验假设。在前面的例子中，一个先验假设是：值大于预定义的阈值不会提供任何有用的信息，因此我们可以将它们裁剪到预定义的阈值。这个先验假设导致我们使用裁剪变换。

特征变换的例子

我们现在列出几种特征变换的通用技术。通常，合并几种特征变换也是有帮助的(比

⊖ 更加精确地说，在第13章描述的正则损失最小化的界依赖于 $\|w^*\|^2$ 和利普希茨性或损失函数的光滑性。对于线性预测器和损失函数的形式 $\ell(w,(x,y))=\phi(\langle w,x\rangle,y)$，这里 ϕ 是凸的，关于第一个变量1-利普希茨或1-光滑，ℓ 或者是 $\|x\|$-利普希茨，或者 $\|x\|^2$-光滑。例如，对于均方损失，$\phi(a,y)=\frac{1}{2}(a-y)^2$，且 $\ell(w,(x,y))=\frac{1}{2}(\langle w,x\rangle-y)^2$ 是关于第一个变量 $\|x\|^2$-光滑。

如，中心化+尺度变换）。接下来，我们用 $f=(f_1, \cdots, f_m)\in\mathbb{R}^m$ 表示在 m 个训练样本上的特征 f。同样，我们用 $\bar{f}=\frac{1}{m}\sum_{i=1}^{m} f_i$ 表示所有样本特征的经验均值。

中心化：

通过变换 $f_i \leftarrow f_i-\bar{f}$，这个变换使得特征有 0 均值。

归一化范围：

这个变换使每一个特征的范围都是[0，1]。形式上，令 $f_{\max}=\max_i f_i$ 并且 $f_{\min}=\min_i f_i$。那么，我们设置 $f_i \leftarrow \frac{f_i-f_{\min}}{f_{\max}-f_{\min}}$。类似地，我们可以使每一个特征的范围为[−1，1]，通过变换 $f_i \leftarrow 2\frac{f_i-f_{\min}}{f_{\max}-f_{\min}}-1$。当然，很容易将范围变换为[0，$b$]或[−$b$，$b$]，其中 b 是用户自定义的参数。

标准化：

这个变换使所有特征有 0 均值和 1 方差。形式上，令 $v=\frac{1}{m}\sum_{i=1}^{m}(f_i-\bar{f})^2$ 表示特征的经验方差，那么设置 $f_i \leftarrow \frac{f_i-\bar{f}}{\sqrt{v}}$。

裁剪变换：

这个变换裁剪特征的高值或者低值。例如 $f_i \leftarrow \mathrm{sign}(f_i)\max\{b, |f_i|\}$，这里 b 是用户自定义参数。

Sigmoidal 变换：

像名字表示的那样，这个变换在特征上用到了 sigmoid 函数。例如 $f_i \leftarrow \frac{1}{1+\exp(bf_i)}$，这里 b 是用户自定义参数。这个变换可以认为是一种软版本的裁剪变换。它对接近于 0 的值有一些作用，并且与远离 0 的裁剪变换很相似。

[318]

对数变换：

这个变换是 $f_i \leftarrow \log(b+f_i)$，这里 b 是用户自定义参数。这个变换广泛地用于当特征是计数型特征的情况。例如，假定特征表示在一个文档中某个词出现的次数。那么，某个词出现一次与没有出现的区别，要比一个词出现 1000 次还是 1001 次更为重要。

评注　在先前的变换中，每一个特征的变换是基于在训练集获得的值，独立于其他特征值。在某些情况下，我们也想基于其他特征值来设置变换的参数。一个著名的例子就是在特征上应用尺度变换，这个尺度变换使得样本的范数的经验平均值为 1。

25.3　特征学习

到目前为止，我们已经讨论了特征选择和特征操作。在这些情况下，我们用预定义的向量空间 $\mathbb{R}^d$ 表示特征。那么，我们选择一个特征子集（特征选择）或者单个特征的变换（特征变换）。在这一节中，我们描述特征学习，从一些样本空间 $\mathcal{X}$ 开始，学习一个函数 $\phi:\mathcal{X}\rightarrow\mathbb{R}^d$，这个映射函数将样本 x 映射到 d 维特征向量。

特征学习的概念就是自动地找到输入空间的好的表示的过程。像上面描述的那样，“没有免费的午餐”理论告诉我们必须在数据分布上应用先验知识，从而建立好的特征表示。在这一节中，我们介绍一些特征学习算法，并且说明这些方法能够应用的潜在数据分布条件。

目前，我们已经介绍了几种有用的特征构建方法。例如，在多项式回归中，我们将源特征映射到所有单项式构成的向量空间(见第 9 章 9.2.2 节)。执行特征映射之后，我们在构建特征上训练一个线性预测器。这个过程的自动化可以学到一个变换 $\psi:\mathcal{X}\to\mathbb{R}^d$，这个变换由定义在 ψ 上的线性预测器假设类组成，得到一个好的假设类用于实际问题处理。

接下来，我们描述一种特征构建的方法，称为字典学习。

用自编码实现字典学习

字典学习的动因来源于用于文档表示的“词袋”方法：给定许多字构成的字典 $D=\{w_1,\dots,w_d\}$，这里每一个 w_i 是一个字符串，表示一个字典里的字。给定一个文档 $(p_1,\dots,p_l)$，这里每一个 p_i 表示文档里的一个字，我们将文档表示成一个向量 $x\in\{0,1\}^d$，这里如果对于某些 $j\in[d]$，$w_j=p_i$，那么 x_j 为 1，其他情况 x_j 为 0。根据在很多文档处理任务的经验观察，线性预测器用在这些表示上非常有效。直观上，我们可以想象每一个字就是度量文档某一方面的一个特征。给定标签的例子(比如文档的主题)，一个学习算法搜索一个线性预测器，这个线性预测器给这些特征赋予权重使得表示标签里出现的每个字都能有正确的连接。 319

当进行文本处理时，词或字典都有自然的含义，在一些其他应用中，我们就没有这样直观的实例表示。例如，计算机视觉的目标识别问题。这里，样本是图像，目标是识别图像中出现的物体。在一个基于像素的图像表示上，应用一个线性预测器就不能取得一个好的分类效果。那么我们如果找到一个映射 ψ 使得能够使用基于像素表示的图像，输出一个“视觉词”构成的袋来表示图像中的内容。例如，一个“视觉字“可以看成是“在图像中有一只眼睛。”如果有这样的表示，我们就可以在这个表示上应用线性预测器来训练一个分类器，比如人脸识别。因此，我们的问题是如何学习字典里的“视觉词”，使得图像的词袋表示能够有助于预测图像里面出现的物体？

首先，字典学习的一个粗略方法依赖于聚类算法(见第 22 章)。假定我们学习到一个函数 $c:\mathcal{X}\to\{1,\dots,k\}$，这里 $c(x)$ 是关于 x 的聚类。那么，我们可以认为这些聚类就是“词”，实例就是文档，这里文档 x 映射到向量 $\psi(x)\in\{0,1\}^k$，这里 $\psi(x)_i$ 是 1 当且仅当 x 属于第 i 个聚类。现在，我们可以明确地看到在 $\psi(x)$ 上应用线性预测器等同于对同一个类的样本赋予同样的目标值。更进一步，如果聚类是基于类中心距离，那么在 $\psi(x)$ 上应用线性预测器将会得到 x 的分段常数预测器。

k 均值和 PCA 方法都可以认为是更通用的字典学习的特例，这个字典学习称为自编码器。在自编码器中，我们学习一对函数：一个编码函数 $\psi:\mathbb{R}^d\to\mathbb{R}^k$，以及一个解码函数 $\varphi:\mathbb{R}^k\to\mathbb{R}^d$。学习过程的目标就是找到一对函数使得构建误差 $\sum_i\|x_i-\varphi(\psi(x_i))\|^2$ 较小。当然，我们可以尝试 $k=d$ 且 ψ，φ 是同一个映射，这个映射使得完美实现重构。因此我们必须通过某种方式限制 ψ 和 φ。在 PCA 算法中，我们限制 $k<d$ 并约束 ψ 和 φ 是线性函数。在 k 均值算法中，k 没有限制小于 d，但是 ψ 和 φ 依赖于 k 个中心 $\boldsymbol{\mu}_1,\dots,\boldsymbol{\mu}_k$ 和 $\psi(x)$ 返回一个 $\{0,1\}^k$ 中的指示向量，这个向量表示最近的中心到 x 的距离，而 φ 作为输入指示向量，返回表示向量的中心。

k 均值构建的一个重要属性(允许 k 大于 d)就是，ψ 将样本映射到一个稀疏向量。事实上，在 k 均值中，只有 $\psi(x)$ 的单个坐标不等于 0。k 均值构建的一个扩展方法就是约束 ψ 是一个至多有 s 个非零元素的向量，这里 s 是一个小的整数。特别地，令 ψ 和 φ 是关于 320

$\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_k$的函数。函数$\phi$把样本向量$\boldsymbol{x}$映射到向量$\phi(\boldsymbol{x}) \in \mathbb{R}^k$，其中$\phi(\boldsymbol{x})$应至多有$s$个非零元素。函数$\varphi(\boldsymbol{v})$定义为$\sum_{i=1}^{k} v_i \boldsymbol{\mu}_i$。和之前表述的一样，我们的目标就是获得一个较小的重构误差，因此定义

$$\phi(\boldsymbol{x}) = \operatorname{argmin}_{\boldsymbol{v}} \|\boldsymbol{x} - \varphi(\boldsymbol{v})\|^2 \text{ s.t. } \|\boldsymbol{v}\|_0 \leqslant s$$

这里。注意当$s=1$，并且$\|\boldsymbol{v}\|_1 = 1$时，我们得到k均值编码函数，也就是$\phi(\boldsymbol{x})$表示与$\boldsymbol{x}$最接近的中心的指示向量。对于s取大值，先前定义关于φ的优化问题变得计算困难。因此，应用上，有时候我们用ℓ_1范数代替稀疏约束，定义ϕ如下：

$$\phi(\boldsymbol{x}) = \operatorname{argmin}_{\boldsymbol{v}} \left[\|\boldsymbol{x} - \varphi(\boldsymbol{v})\|^2 + \lambda \|\boldsymbol{v}\|_1\right]$$

这里$\lambda > 0$是一个正则参数。总之，字典学习问题就是找到向量$\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_k$使得重构误差$\sum_i \|\boldsymbol{x}_i - \varphi(\phi(\boldsymbol{x}_i))\|^2$尽可能小。即使$\phi$是用$\ell_1$范数来定义的，这仍然是计算困难问题（与$k$均值方法问题相似）。然而，一些启发式算法可能给出相当好的结果，这些算法超出了本书讨论的范围。

25.4　小结

很多特征学习算法想当然地应用样本的特征表示，然而特征表示的选择问题需要格外小心。我们讨论了特征选择的方法，介绍了滤波、贪婪选择算法以及稀疏诱导范数。接下来，我们给出了几种特征变换的实例，介绍了它们的用途。最后，我们讨论了特征学习，并且特别介绍了字典学习。我们说明了特征选择、特征操作和特征学习都依赖于数据的先验知识。

25.5　文献评注

2003年，Guyon和Elisseeff综述了几种特征选择的方法，包括很多滤波类型的方法。前向贪婪选择方法用于最小化多面体约束下的凸目标优化问题起源于Frank Wolfe算法（Frank & Wolfe 1956），好几位学者研究了前向贪婪选择方法与boosting方法的关系问题，包括Warmuth、Liao & Ratsch(2006)，Warmuth、Glocer & Vishwanathan(2008)，Shalev Shwartz & Singer(2008)。正交匹配追踪算法已经用于信号处理领域（Mallat & Zhang 1993）。一些文献分析了各种不同条件下的贪婪选择方法。例如，Shalev-Shwartz、Zhang和Srebro(2010)，以及这些文章的一些参考文献。

821　使用ℓ_1范数来近似稀疏表示有很长的历史（比如Tibshirani(1996)以及其中的参考文献），很多工作是在研究理解ℓ_1范数与稀疏的关系问题。这也与压缩感知非常接近（见第23章）。稀疏低ℓ_1范数的能力来源于Maurey(Pisier 1980—1981)。在26.4节，我们也可以看到低ℓ_1范数能够用于预测期的估计误差界。

特征学习和字典学习已经被扩展应用于深度神经网络学习上。比如LeCun & Bengio(1995)，Hinton等(2006)，Ranzato等(2007)，Collobert & Weston(2008)，Lee等(2009)，Le等(2012)，Bengio(2009)，以及其中的参考文献。

25.6　练习

25.1　证明方程(25.1)给出的等式。提示：令a^*，b^*是最小化左边式子时的取值。找到a，b使得右边的函数值小于左边的目标函数值。找到另一组a，b使得左边的函数

使小于右边的目标函数值。

25.2　证明方程(25.7)是方程(25.6)的解。

25.3　**AdaBoost 是一种前向贪婪选择算法**：回想第 10 章的 AdaBoost 算法，在这一节中我们给出了 AdaBoost 的另一种解析，那就是作为一种前向贪婪选择算法。

- 给定 m 个样本的集合 $x_1, \ldots, x_m$，以及一个 VC 维有限的假设类 $\mathcal{H}$，证明存在 d 以及 $h_1, \ldots, h_d$ 使得对于每一个 $h \in \mathcal{H}$，存在 $i \in [d]$，对于每一个 $j \in [m]$，$h_i(x_j) = h(x_j)$。
- 令 $R(\boldsymbol{w})$ 如方程(25.3)定义。给定一些 $\boldsymbol{w}$，定义 $f_{\boldsymbol{w}}$ 函数如下：

$$f_{\boldsymbol{w}}(\cdot) = \sum_{i=1}^{d} w_i h_i(\cdot)$$

  令 $\boldsymbol{D}$ 表示 $[m]$ 上的分布，定义如下：

$$D_i = \frac{\exp(-y_i f_{\boldsymbol{w}}(x_i))}{Z}$$

  这里 Z 是归一化因子，使得 $\boldsymbol{D}$ 是概率向量，证明：

$$\frac{\partial R(\boldsymbol{w})}{w_j} = -\sum_{i=1}^{m} D_i y_i h_j(x_i)$$

  更进一步，表示 $\epsilon_j = \sum_{i=1}^{m} D_i \mathbb{1}_{[h_j(x_i) \neq y_i]}$，证明

$$\frac{\partial R(\boldsymbol{w})}{w_j} = 2\epsilon_j - 1$$

  得出结论，如果 $\epsilon_j \leq 1/2 - \gamma$，那么 $\left|\frac{\partial R(\boldsymbol{w})}{w_j}\right| \geq \gamma/2$。
- 试说明由 AdaBoost 算法迭代可以得到 $R(\boldsymbol{w}^{(t+1)}) - R(\boldsymbol{w}^{(t)}) \leq \log(\sqrt{1-4\gamma^2})$。提示：利用定理 10.2 的证明。

322

第四部分

Understanding Machine Learning: From Theory to Algorithms

高级理论

拉德马赫复杂度

在第 4 章中，我们已经证明了一致收敛性是可学习的充分条件。那么在本章我们介绍用来测量一致收敛的速率的拉德马赫复杂度的相关知识。最后给出基于这种测量方法的一些泛化误差。

26.1 拉德马赫复杂度概述

回忆第 4 章关于 ε-代表性样本的定义。我们在这里为了方便，重复列于下方。

定义 26.1（ε-代表性样本） 一个训练集被称作 ε-代表性样本（定义在域 Z、假设类 $\mathcal{H}$、损失函数 ℓ 和分布 $\mathcal{D}$），如果满足

$$\sup_{h\in\mathcal{H}}|L_{\mathcal{D}}(h)-L_S(h)|\leqslant\varepsilon$$

我们已经证明了如果 S 是 $\varepsilon/2$-代表性样本，那么在 ERM 准则下它是 ε-一致的，即：$L_{\mathcal{D}}(\mathrm{ERM}_{\mathcal{H}}(S))\leqslant\min_{h\in\mathcal{H}}L_{\mathcal{D}}(h)+\varepsilon$。

为了简化记号，让我们表示如下：

$$\mathcal{F}=\ell\circ\mathcal{H}\stackrel{\text{def}}{=}\{z\mapsto\ell(h,z):h\in\mathcal{H}\}$$

并且给定 $f\in\mathcal{F}$，我们定义

$$L_{\mathcal{D}}(f)=\mathop{\mathbb{E}}_{z\sim\mathcal{D}}[f(z)],\quad L_S(f)=\frac{1}{m}\sum_{i=1}^{m}f(z_i)$$

我们定义集合 S 在 $\mathcal{F}$ 上的代表性为一个函数 f 的真实误差和它的经验误差的上确界。即

$$\mathrm{Rep}_{\mathcal{D}}(\mathcal{F},S)\stackrel{\text{def}}{=}\sup_{f\in\mathcal{F}}(L_{\mathcal{D}}(f)-L_S(f)) \tag{26.1}$$

现在，假设我们想要仅依靠样本集 S 本身来估计它的代表性。一个简单的想法就是将 S 分为两个不相交的子集，$S=S_1\cup S_2$。将 S_1 作为验证集、S_2 作为训练集。我们就可以测得 S 的样本集代表性：

325

$$\sup_{f\in\mathcal{F}}(L_{S_2}(f)-L_{S_1}(f)) \tag{26.2}$$

如果我们设 $\boldsymbol{\sigma}=(\sigma_1,\cdots,\sigma_m)\in\{\pm1\}^m$ 为一个向量，并且使得 $S_1=\{z_i:\sigma_i=1\}$，$S_2=\{z_i:\sigma_i=-1\}$，那么等式(26.2)就可以被更简洁地表示为

$$\frac{2}{m}\sup_{f\in\mathcal{F}}\sum_{i=1}^{m}\sigma_i f(z_i) \tag{26.3}$$

拉德马赫复杂度采用了这个想法，它考虑在随机选择 $\boldsymbol{\sigma}$ 的情况下等式(26.3)的期望。形式上，令 $\mathcal{F}\circ S$ 是一个函数 $f\in\mathcal{F}$ 在样本集 S 上所取得的函数值的全体，即

$$\mathcal{F}\circ S=\{(f(z_1),\cdots,f(z_m)):f\in\mathcal{F}\}$$

依据 $\Pr[\sigma_i=1]=\Pr[\sigma_i=-1]=\frac{1}{2}$，设 $\boldsymbol{\sigma}$ 中的变量是独立同分布的。那么定义在样本集 S 上的 $\mathcal{F}$ 的拉德马赫复杂度按如下定义：

$$R(\mathcal{F}\circ S)\stackrel{\text{def}}{=}\frac{1}{m}\mathop{\mathbb{E}}_{\sigma\sim\{\pm1\}^m}\left[\sup_{f\in\mathcal{F}}\sum_{i=1}^{m}\sigma_i f(z_i)\right] \tag{26.4}$$

更其一般性，给定一个向量集合 $A\in\mathbb{R}^m$，我们定义

$$R(A)\stackrel{\text{def}}{=}\frac{1}{m}\mathop{\mathbb{E}}_{\sigma}\left[\sup_{a\in A}\sum_{i=1}^{m}\sigma_i a_i\right] \tag{26.5}$$

接下来的引理表明集合 S 的代表性的期望不大于 2 倍的拉德马赫复杂度的期望。

引理 26.2

$$\mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}[\mathrm{Rep}_{\mathcal{D}}(\mathcal{F},S)]\leqslant 2\mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}R(\mathcal{F}\circ S)$$

证明 设 $S'=\{z'_1,\cdots,z'_m\}$ 是另外一个独立同分布样本。显然，对于所有的 $f\in\mathcal{F}$，$L_{\mathcal{D}}(f)=\mathop{\mathbb{E}}_{S'}[L_{S'}(f)]$，因此，对于每一个 $f\in\mathcal{F}$，我们有

$$L_{\mathcal{D}}(f)-L_S(f)=\mathop{\mathbb{E}}_{S'}[L_{S'}(f)]-L_S(f)=\mathop{\mathbb{E}}_{S'}[L_{S'}(f)-L_S(f)]$$

在等式两边对 $f\in\mathcal{F}$ 取上确界，并且依据期望的上确界小于上确界的期望这一事实，我们可以得到

$$\begin{aligned}\sup_{f\in\mathcal{F}}(L_{\mathcal{D}}(f)-L_S(f))&=\sup_{f\in\mathcal{F}}\mathop{\mathbb{E}}_{S'}[L_{S'}(f)-L_S(f)]\\&\leqslant\mathop{\mathbb{E}}_{S'}\left[\sup_{f\in\mathcal{F}}(L_{S'}(f)-L_S(f))\right]\end{aligned}$$

[326]

在等式两边同时对 S 取期望，我们有

$$\begin{aligned}\mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}\left[\sup_{f\in\mathcal{F}}(L_{\mathcal{D}}(f)-L_S(f))\right]&\leqslant\mathop{\mathbb{E}}_{S,S'}\left[\sup_{f\in\mathcal{F}}(L_{S'}(f)-L_S(f))\right]\\&=\frac{1}{m}\mathop{\mathbb{E}}_{S,S'}\left[\sup_{f\in\mathcal{F}}\sum_{i=1}^{m}(f(z'_i)-f(z_i))\right]\end{aligned} \tag{26.6}$$

接下来，我们注意到 z_i，z'_i 和 z_i 都是独立同分布变量，因此，可以互换它们且并不影响期望：

$$\begin{aligned}&\mathop{\mathbb{E}}_{S,S'}\left[\sup_{f\in\mathcal{F}}\left(f(z'_j)-f(z_j)+\sum_{i\neq j}(f(z'_i)-f(z_i))\right)\right]\\&=\mathop{\mathbb{E}}_{S,S'}\left[\sup_{f\in\mathcal{F}}\left((f(z_j)-f(z'_j))+\sum_{i\neq j}(f(z'_i)-f(z_i))\right)\right]\end{aligned} \tag{26.7}$$

令 σ_j 是一个随机变量，且满足 $\mathbb{P}[\sigma_j=1]=\mathbb{P}[\sigma_j=-1]=\frac{1}{2}$，从等式(26.7)我们可以得到

$$\begin{aligned}&\mathop{\mathbb{E}}_{S,S',\sigma_j}\left[\sup_{f\in\mathcal{F}}\left(\sigma_j(f(z'_j)-f(z_j))+\sum_{i\neq j}(f(z'_i)-f(z_i))\right)\right]\\&=\frac{1}{2}\text{（方程(26.7)左边）}+\frac{1}{2}\text{（方程(26.7)右边）}\\&=\mathop{\mathbb{E}}_{S,S'}\left[\sup_{f\in\mathcal{F}}\left((f(z'_j)-f(z_j))+\sum_{i\neq j}(f(z'_i)-f(z_i))\right)\right]\end{aligned} \tag{26.8}$$

对于所有 j 重复上述步骤，我们有

$$\mathop{\mathbb{E}}_{S,S'}\left[\sup_{f\in\mathcal{F}}\sum_{i=1}^{m}(f(z'_i)-f(z_i))\right]=\mathop{\mathbb{E}}_{S,S',\sigma}\left[\sup_{f\in\mathcal{F}}\sum_{i=1}^{m}\sigma_i(f(z'_i)-f(z_i))\right] \tag{26.9}$$

最终

$$\sup_{f\in\mathcal{F}}\sum_i\sigma_i(f(z'_i)-f(z_i))\leqslant\sup_{f\in\mathcal{F}}\sum_i\sigma_i f(z'_i)+\sup_{f\in\mathcal{F}}\sum_i-\sigma_i f(z_i)$$

并且由于 σ 的概率等于 $-\sigma$ 的概率，等式(26.9)的右边可以重写为

$$\mathbb{E}_{\sigma\sim\mathcal{U}_m}\left[\sup_{f\in\mathcal{F}}\sum_i \sigma_i f(x'_i) - \sup_{f\in\mathcal{F}}\sum_i \sigma_i f(x_i)\right]$$

$$= m\,\mathbb{E}_{S'}[R(\mathcal{F}, S')] + m\,\mathbb{E}_{S}[R(\mathcal{F}, S)] = 2m\,\mathbb{E}_{S}[R(\mathcal{F}, S)] \qquad \blacksquare$$

这个引理直接告诉我们，在期望意义下，依据 ERM 准则找到的假设类非常接近 $\mathcal{H}$ 中的最优假设类。

327

定理 26.3 我们有

$$\mathbb{E}_{S\sim\mathcal{D}^m}\left[L_{\mathcal{D}}(\mathrm{ERM}_{\mathcal{H}}(S)) - L_S(\mathrm{ERM}_{\mathcal{H}}(S))\right] \le 2\,\mathbb{E}_{S\sim\mathcal{D}^m} R(\ell\circ\mathcal{H}, S)$$

而对于所有的 $h^*\in\mathcal{H}$，有

$$\mathbb{E}_{S\sim\mathcal{D}^m}\left[L_{\mathcal{D}}(\mathrm{ERM}_{\mathcal{H}}(S)) - L_{\mathcal{D}}(h^*)\right] \le 2\,\mathbb{E}_{S\sim\mathcal{D}^m} R(\ell\circ\mathcal{H}, S)$$

更进一步，如果 $h^* = \arg\min_h L_{\mathcal{D}}(h)$，那么对于任意的 $\delta\in(0,1)$，我们至少以 $1-\delta$ 的概率在样本集 S 上有

$$L_{\mathcal{D}}(\mathrm{ERM}_{\mathcal{H}}(S)) - L_{\mathcal{D}}(h^*) \le \frac{2\,\mathbb{E}_{S\sim\mathcal{D}^m} R(\ell\circ\mathcal{H}, S)}{\delta}$$

证明 第一个不等式可以直接由引理 26.2 获得。第二个不等式则根据对于任意固定的 h^*，有

$$L_{\mathcal{D}}(h^*) = \mathbb{E}_S[L_S(h^*)] \ge \mathbb{E}_S[L_S(\mathrm{ERM}_{\mathcal{H}}(S))]$$

第三个不等式可以由之前的不等式以及马尔可夫不等式(注意随机变量 $L_{\mathcal{D}}(\mathrm{ERM}_{\mathcal{H}}(S)) - L_{\mathcal{D}}(h^*)$ 是非负的)获得。 $\blacksquare$

接下来，我们可以从定理 26.3 得到一个更独立依赖置信参数 δ 的界。为了得到这个界，我们首先介绍下面的有界偏差集中不等式。

引理 26.4(麦克迪尔米德不等式) 设 V 是某一集合，且设 $f: V^m\to\mathbb{R}$ 是一个 m 个自变量的函数，对于某个 $c>0$，对于所有的 $i\in[m]$ 且对于所有的 $x_1, \ldots, x_m, x_i'\in V$，我们有

$$|f(x_1,\ldots,x_m) - f(x_1,\ldots,x_{i-1},x_i',x_{i+1},\ldots,x_m)| \le c$$

设 $X_1, \ldots, X_m$ 是取自 V 中的 m 个独立随机变量，那么，我们至少以 $1-\delta$ 的概率有

$$|f(X_1,\ldots,X_m) - \mathbb{E}[f(X_1,\ldots,X_m)]| \le c\sqrt{\ln\left(\frac{2}{\delta}\right)m/2}$$

在麦克迪尔米德不等式的基础上我们可以得到一个更独立于置信参数 δ 的泛化误差界。

定理 26.5 假设对于所有的 z 和 $h\in\mathcal{H}$，我们有 $\ell(h, z) \le c$，那么

1. 至少以 $1-\delta$ 的概率，对于所有的 $h\in\mathcal{H}$ 有

$$L_{\mathcal{D}}(h) - L_S(h) \le 2\,\mathbb{E}_{S\sim\mathcal{D}^m} R(\ell\circ\mathcal{H}, S) + c\sqrt{\frac{2\ln(2/\delta)}{m}}$$

特别地，不等式对于 $h=\mathrm{ERM}_{\mathcal{H}}(S)$ 也成立。

2. 至少以 $1-\delta$ 的概率，对于所有的 $h\in\mathcal{H}$ 有

$$L_{\mathcal{D}}(h) - L_S(h) \le 2R(\ell\circ\mathcal{H}, S) + 4c\sqrt{\frac{2\ln(4/\delta)}{m}}$$

328

特别地，不等式对于 $h=\mathrm{ERM}_{\mathcal{H}}(S)$ 也成立。

3. 对于任意的 h^*，至少以 $1-\delta$ 的概率有

$$L_{\mathcal{D}}(\mathrm{ERM}_{\mathcal{H}}(S)) - L_{\mathcal{D}}(h^*) \leqslant 2R(\ell \circ \mathcal{H} \circ S) + 5c\sqrt{\frac{2\ln(8/\delta)}{m}}$$

证明　首先注意到随机变量 $\mathrm{Rep}_{\mathcal{D}}(\mathcal{F}, S) = \sup_{h\in\mathcal{H}}(L_{\mathcal{D}}(h) - L_S(h))$ 满足引理 26.4 在常数 $2c/m$ 下的有界偏差条件，结合引理 26.4 和引理 26.2 的结论，我们可以以至少以 $1-\delta$ 的概率有

$$\mathrm{Rep}_{\mathcal{D}}(\mathcal{F}, S) \leqslant \mathbb{E}\,\mathrm{Rep}_{\mathcal{D}}(\mathcal{F}, S) + c\sqrt{\frac{2\ln(2/\delta)}{m}} \leqslant 2\,\mathop{\mathbb{E}}_{S} R(\ell \circ \mathcal{H} \circ S') + c\sqrt{\frac{2\ln(2/\delta)}{m}}$$

第一个不等式可以直接从 $\mathrm{Rep}_{\mathcal{D}}(\mathcal{F}, S)$ 的定义中得到，而对于第二个不等式，我们注意到随机变量 $R(\ell \circ \mathcal{H} \circ S)$ 同样满足在常数 $2c/m$ 下的有界偏差条件，所以第二个不等式就可以直接由引理 26.4，第一个不等式以及联合边界得到。最后，对于第三个不等式，记 $h_S = \mathrm{ERM}_{\mathcal{H}}(S)$，并且注意到

$$\begin{aligned} L_{\mathcal{D}}(h_S) - L_{\mathcal{D}}(h^*) &= L_{\mathcal{D}}(h_S) - L_S(h_S) + L_S(h_S) - L_S(h^*) + L_S(h^*) - L_{\mathcal{D}}(h^*) \\ &\leqslant (L_{\mathcal{D}}(h_S) - L_S(h_S)) + (L_S(h^*) - L_{\mathcal{D}}(h^*)) \end{aligned} \tag{26.10}$$

等式右边的第一个加项的上界由第二个不等式就可以得到，第二个加项需注意到 h^* 是不依赖于样本集 S 的，运用霍夫丁不等式，我们至少以 $1-\delta/2$ 的概率有

$$L_S(h^*) - L_{\mathcal{D}}(h^*) \leqslant c\sqrt{\frac{\ln(4/\delta)}{2m}} \tag{26.11}$$

结合联合边界我们证明了结论。 ■

之前的定理告诉我们，如果 $R(\ell \circ \mathcal{H} \circ S)$ 很小，那么很有可能利用 ERM 准则可以学习到假设类 $\mathcal{H}$。值得强调的是上述定理给出的最后两个界是依赖于特定的训练集 S 的，这意味着，我们利用训练集 S 去学习假设类并且用它去衡量这个假设类的好坏。这种类型的界我们称之为数据相关界。

拉德马赫积分

现在，让我们讨论拉德马赫复杂度的一些性质，这些性质便于我们获得针对某些特殊情况下的 $R(\ell \circ \mathcal{H} \circ S)$ 的简单的界。

下面的引理可以直接由定义获得。

引理 26.6　对于任意的 $A \in \mathbb{R}^m$，标量 $c \in \mathbb{R}$ 和向量 $a_0 \in \mathbb{R}^m$，我们有

$$R(\{ca + a_0 : a \in A\}) \leqslant |c| R(A)$$

下面的引理告诉我们 A 的凸包与 A 有一样的复杂度。

引理 26.7　设 A 是 $\mathbb{R}^m$ 的一个子集，且 $A' = \{\sum_{j=1}^{N} \alpha_j a^{(j)} : N \in \mathbb{N}, \forall j, a^{(j)} \in A, \alpha_j \geqslant 0, \|\alpha\|_1 = 1\}$，那么 $R(A') = R(A)$。

证明　对于任意向量我们有

$$\sup_{\alpha \geqslant 0, \|\alpha\|_1 = 1} \sum_{i=1}^{N} \alpha_i v_i = \max_i v_i$$

因此

$$mR(A') = \mathop{\mathbb{E}}_{\sigma} \sup_{\alpha \geqslant 0, \|\alpha\|_1 = 1} \sup_{a^{(1)}, \ldots, a^{(N)} \in A} \sum_{i=1}^{m} \sigma_i \sum_{j=1}^{N} \alpha_j a_i^{(j)}$$

$$
\begin{aligned}
&= \mathop{\mathbb{E}}_{\sigma} \sup_{\|a\|_1 \le 1} \sum_{j=1}^{N} a_j \sup_{\sigma^{(j)}} \sum_{i=1}^{m} \sigma_i a_i^{(j)} \\
&= \mathop{\mathbb{E}}_{\sigma} \sup_{a \in A} \sum_{i=1}^{m} \sigma_i a_i \\
&= mR(A)
\end{aligned}
$$

我们证明了该结论。 ■

下面的引理(来自于马萨特)说明了一个有限集合的拉德马赫复杂度随集合的大小而对数增长。

引理 26.8(马萨特引理) 令 $A=\{a_1, \cdots, a_N\}$ 是 $\mathbb{R}^m$ 中的一个有限集合。定义 $\bar{a}=\frac{1}{N}\sum_{i=1}^{N} a_i$，那么

$$
R(A) \le \max_{a \in A} \|a-\bar{a}\| \frac{\sqrt{2\log(N)}}{m}
$$

证明 在引理 26.6 的基础上，我们可以不失一般性地假设 $\bar{a}=\mathbf{0}$，令 $\lambda>0$ 且 $A'=\{\lambda a_1, \cdots, \lambda a_N\}$，我们得到它的拉德马赫上界：

$$
\begin{aligned}
mR(A') &= \mathop{\mathbb{E}}_{\sigma}\Big[\max_{a \in A'} \langle \sigma, a \rangle\Big] = \mathop{\mathbb{E}}_{\sigma}\Big[\log\Big(\max_{a \in A'} e^{\langle \sigma, a \rangle}\Big)\Big] \\
&\le \mathop{\mathbb{E}}_{\sigma}\Big[\log\Big(\sum_{a \in A'} e^{\langle \sigma, a \rangle}\Big)\Big] \\
&\le \log\Big(\mathop{\mathbb{E}}_{\sigma}\Big[\sum_{a \in A'} e^{\langle \sigma, a \rangle}\Big]\Big) \qquad \text{// 詹生不等式} \\
&= \log\Big(\sum_{a \in A'} \prod_{i=1}^{m} \mathop{\mathbb{E}}_{\sigma_i}\big[e^{\sigma_i a_i}\big]\Big)
\end{aligned}
$$

最后一个等式成立是由于拉德马赫变量之间是相互独立的。

接下来，利用引理 A.6，对于所有的 $a_i \in \mathbb{R}^m$，我们有

330

$$
\mathop{\mathbb{E}}_{\sigma_i} e^{\sigma_i a_i} = \frac{\exp(a_i)+\exp(-a_i)}{2} \le \exp(a_i^2/2)
$$

因此

$$
\begin{aligned}
mR(A') &\le \log\Big(\sum_{a \in A'} \prod_{i=1}^{m} \exp\Big(\frac{a_i^2}{2}\Big)\Big) = \log\Big(\sum_{a \in A'} \exp\big(\|a\|^2/2\big)\Big) \\
&\le \log\Big(|A'| \max_{a \in A'} \exp\big(\|a\|^2/2\big)\Big) = \log(|A'|) + \max_{a \in A'}(\|a\|^2/2)
\end{aligned}
$$

由于 $R(A)=\frac{1}{\lambda}R(A')$，我们得到等式

$$
R(A) \le \frac{\log(|A|)+\lambda^2 \max_{a \in A}(\|a\|^2/2)}{\lambda m}
$$

令 $\lambda=\sqrt{2\log(|A|)/\max_{a \in A}\|a\|^2}$，重新排列式子，我们就得到了结论。 ■

下面的引理告诉我们如果给集合 A 作用一个利普希茨函数，并不会增大它的拉德马赫复杂度。这个证明主要是来自 Kakade 和 Tewari。

引理 26.9(压缩引理) 对每一个 $i \in [m]$，令 $\varphi_i: \mathbb{R} \to \mathbb{R}$ 是一个 ρ-利普希茨函数：即，

对于所有的 α，$\beta\in\mathbb{R}$，我们有 $|\varphi_i(\alpha)-\varphi_i(\beta)|\leqslant\rho|\alpha-\beta|$。对于 $a\in\mathbb{R}^n$，令 $\boldsymbol{\varphi}(a)$ 表示向量 $(\varphi_1(a_1),\cdots,\varphi_n(a_n))\leqslant\rho|a-\beta|$。令 $\boldsymbol{\varphi}\circ A=\{\boldsymbol{\varphi}(a):a\in A\}$。那么

$$R(\boldsymbol{\varphi}\circ A)\leqslant\rho(A)$$

证明　为了简单起见，我们只证明 $\rho=1$ 的情形，而当 $\rho\neq1$ 时，我们可以定义 $\varphi'=\frac{1}{\rho}\varphi$，并利用引理 26.6 即可。设 $A_i=\{(a_1,\cdots,a_{i-1},\varphi_i(a_i),a_{i+1},\cdots,a_n):a\in A\}$，这足够证明对于任意集合 A 和所有的 i，我们有 $R(A_i)\leqslant R(A)$。不失一般性，我们将在下面的证明中只证明 $i=1$ 的情形，并且为了简单起见，省略了 φ_1 的下标。我们有

$$\begin{aligned}
mR(A_1)&=\mathbb{E}_\sigma\Big[\sup_{a=(a_1,\cdots,a_m)\in A_1}\sum_{i=1}^m\sigma_ia_i\Big]\\
&=\mathbb{E}_\sigma\Big[\sup_{a\in A}\sigma_1\varphi(a_1)+\sum_{i=2}^m\sigma_ia_i\Big]\\
&=\frac{1}{2}\mathbb{E}_{\sigma_2,\cdots,\sigma_m}\Big[\sup_{a\in A}\Big(\varphi(a_1)+\sum_{i=2}^m\sigma_ia_i\Big)+\sup_{a\in A}\Big(-\varphi(a_1)+\sum_{i=2}^m\sigma_ia_i\Big)\Big]\\
&=\frac{1}{2}\mathbb{E}_{\sigma_2,\cdots,\sigma_m}\Big[\sup_{a,a'\in A}\Big(\varphi(a_1)-\varphi(a_1')+\sum_{i=2}^m\sigma_ia_i+\sum_{i=2}^m\sigma_ia_i'\Big)\Big]\\
&\leqslant\frac{1}{2}\mathbb{E}_{\sigma_2,\cdots,\sigma_m}\Big[\sup_{a,a'\in A}\Big(|a_1-a_1'|+\sum_{i=2}^m\sigma_ia_i+\sum_{i=2}^m\sigma_ia_i'\Big)\Big]
\end{aligned}\tag{26.12}$$

其中，在最后的一个不等式中我们利用了假设，即 φ 是利普希茨函数。我们注意到最后的表达式的前半部分，绝对值 $|a-a'|$ 是可以省略的，这是因为 a，a' 都是来自于同一个集合 A，并且，表达式的后半部分的上确界并不受 a，a' 的替换的影响，因此 [337]

$$mR(A_1)\leqslant\frac{1}{2}\mathbb{E}_{\sigma_2,\cdots,\sigma_m}\Big[\sup_{a,a'\in A}\Big(a_1-a_1'+\sum_{i=2}^m\sigma_ia_i+\sum_{i=2}^m\sigma_ia_i'\Big)\Big]\tag{26.13}$$

但是，用不等式(26.12)中的不等关系，不难发现不等式(26.13)的右边等于 $mR(A)$，这就证明了我们的结论。 ■

26.2　线性类的拉德马赫复杂度

本节我们分析线性类的拉德马赫复杂度。为了简化推导，首先定义以下两类：

$$\mathcal{H}_1=\{x\mapsto\langle w,x\rangle:\|w\|_1\leqslant1\},\quad\mathcal{H}_2=\{x\mapsto\langle w,x\rangle:\|w\|_2\leqslant1\}\tag{26.14}$$

下面的引理给出了假设类 $\mathcal{H}_2$ 的拉德马赫复杂度的界，我们允许 x_i 可以是任意希尔伯特空间的向量(甚至是无穷维空间)。这个性质当我们分析核方法时是非常有用的。

引理 26.10　令 $S=(x_1,\cdots,x_m)$ 是希尔伯特空间的一个向量，定义 $\mathcal{H}_2\circ S=\{(\langle w,x_1\rangle,\cdots,\langle w,x_m\rangle):\|w\|_2\leqslant1\}$，那么

$$R(\mathcal{H}_2\circ S)\leqslant\frac{\max_i\|x_i\|_2}{\sqrt{m}}$$

证明　利用柯西-施瓦茨不等式，我们知道对于任意两个向量 w，v，我们有 $\langle w,v\rangle\leqslant\|w\|\|v\|$，因此

$$mR(\mathcal{H}_2\circ S)=\mathbb{E}_\sigma\Big[\sup_{w\in\mathcal{H}_2\circ S}\sum_{i=1}^m\sigma_ia_i\Big]$$

$$
\begin{aligned}
&= \mathbb{E}_{\sigma}\left[\left[\sup_{w:\|w\|\le\rho}\sum_{i=1}^{m}\sigma_i\langle w, x_i\rangle\right]\right] \\
&= \mathbb{E}_{\sigma}\left[\sup_{w:\|w\|\le\rho}\left\langle w, \sum_{i=1}^{m}\sigma_i x_i\right\rangle\right] \\
&\le \mathbb{E}_{\sigma}\left[\left\|\sum_{i=1}^{m}\sigma_i x_i\right\|_2\right]
\end{aligned}
\tag{26.15}
$$

利用恰生不等式，我们有

352

$$
\mathbb{E}_{\sigma}\left[\left\|\sum_{i=1}^{m}\sigma_i x_i\right\|_2\right] = \mathbb{E}_{\sigma}\left[\left(\left\|\sum_{i=1}^{m}\sigma_i x_i\right\|_2^2\right)^{1/2}\right] \le \left(\mathbb{E}_{\sigma}\left[\left\|\sum_{i=1}^{m}\sigma_i x_i\right\|_2^2\right]\right)^{1/2} \tag{26.16}
$$

最后，由于变量 $\sigma_1, \cdots, \sigma_m$ 是相互独立的，所以

$$
\begin{aligned}
\mathbb{E}_{\sigma}\left[\left\|\sum_{i=1}^{m}\sigma_i x_i\right\|_2^2\right] &= \mathbb{E}_{\sigma}\left[\sum_{i,j}\sigma_i\sigma_j\langle x_i, x_j\rangle\right] \\
&= \sum_{i\ne j}\langle x_i, x_j\rangle\,\mathbb{E}_{\sigma}[\sigma_i\sigma_j] + \sum_{i=1}^{m}\langle x_i, x_i\rangle\,\mathbb{E}_{\sigma}[\sigma_i^2] \\
&= \sum_{i=1}^{m}\|x_i\|_2^2 \le m\max_i\|x_i\|_2^2
\end{aligned}
$$

结合等式(26.15)和等式(26.16)，我们证明了结论。 ■

接下来我们证明 $\mathcal{H}_1 \circ S$ 的拉德马赫复杂度。

引理 26.11　令 $S=(x_1, \cdots, x_m)$ 是 $\mathbb{R}^n$ 空间中的的一个向量，那么

$$
R(\mathcal{H}_1 \circ S) \le \max_i\|x_i\|_\infty\sqrt{\frac{2\log(2n)}{m}}
$$

证明　利用 Holder 不等式，我们知道对于任意两个向量 w，v，我们有 $\langle w, v\rangle \le \|w\|_1\|v\|_\infty$。因此

$$
\begin{aligned}
mR(\mathcal{H}_1 \circ S) &= \mathbb{E}_{\sigma}\left[\sup_{h\in\mathcal{H}_1\circ S}\sum_{i=1}^{m}\sigma h_i\right] \\
&= \mathbb{E}_{\sigma}\left[\sup_{w:\|w\|_1\le 1}\sum_{i=1}^{m}\sigma_i\langle w, x_i\rangle\right] \\
&= \mathbb{E}_{\sigma}\left[\sup_{w:\|w\|_1\le 1}\left\langle w, \sum_{i=1}^{m}\sigma_i x_i\right\rangle\right] \\
&\le \mathbb{E}_{\sigma}\left[\left\|\sum_{i=1}^{m}\sigma_i x_i\right\|_\infty\right]
\end{aligned}
\tag{26.17}
$$

对于每一个 $j\in[n]$，令 $v_j=(x_{1,j}, \cdots, x_{m,j})\in\mathbb{R}^m$，注意到 $\|v_j\|_2\le\sqrt{m}\max_i\|x_i\|_\infty$。令 $V=(v_1, \cdots, v_n, -v_1, \cdots, -v_n)$。不等式(26.17)的右边是 $mR(V)$。使用马萨特引理(引理 26.8)，我们有

$$
R(V) \le \max_i\|x_i\|_\infty\sqrt{2\log(2n)/m}
$$

这就证明了我们的结论。 ■

26.3　SVM 的泛化误差界

本节我们利用拉德马赫复杂度得到在欧几里得范数限制下的广义线性预测器的泛化误

差界。我们将展示如何推导出硬-SVM 和软-SVM 的泛化误差界。 333

我们将考虑基于以下约束的一般形式，令$\mathcal{H}=\{w:\|w\|_2\leq B\}$作为我们的假设类，令$Z=\mathcal{X}\times\mathcal{Y}$作为样本空间，假设损失函数$\ell:\mathcal{H}\times Z\to\mathbb{R}$具有如下形式

$$\ell(w,(x,y))=\varphi(\langle w,x\rangle,y) \tag{26.18}$$

这里的$\varphi:\mathbb{R}\times\mathcal{Y}\to\mathbb{R}$是指对于所有的$y\in\mathcal{Y}$，标量函数$a\mapsto\varphi(a,y)$是$\rho$-利普希茨函数。例如，铰链损失 hinge-loss 函数$\ell(w,(x,y))=\max\{0,1-y\langle w,x\rangle\}$可以写作等式(26.18)的形式，只需使$\varphi(a,y)=\max\{0,1-ya\}$，并且应注意到$\varphi$对于所有的$y\in\{\pm1\}$都满足 1-利普希茨条件。另一个例子是绝对损失函数，$\ell(w,(x,y))=|\langle w,x\rangle-y|$，它也可以写作等式(26.18)的形式，只需使$\varphi(a,y)=|a-y|$，它也是对于所有的$y\in\mathbb{R}$满足 1-利普希茨条件。

下面的定理给出了$\mathcal{H}$中所有预测的泛化误差界，这个界是利用经验误差给出的。

引理 26.12　*假设$\mathcal{D}$是在$\mathcal{X}\times\mathcal{Y}$上的一个概率分布，我们以概率 1 有$\|x\|_2\leq R$。令$\mathcal{H}=\{w:\|w\|_2\leq B\}$，$\ell:\mathcal{H}\times Z\to\mathbb{R}$是形如等式(26.18)中的一个损失函数，因此，对于所有的$y\in\mathcal{Y}$，$a\mapsto\varphi(a,y)$是一个ρ-利普希茨函数且$\max_{a\in[-BR,BR]}|\varphi(a,y)|\leq c$。那么，对于任意的$\delta\in(0,1)$，至少以$1-\delta$的概率选择一个大小为$m$的独立同分布样本*

$$\forall w\in\mathcal{H},L_{\mathcal{D}}(w)\leq L_S(w)+\frac{2\rho BR}{\sqrt{m}}+c\sqrt{\frac{2\ln(2/\delta)}{m}}$$

证明　令$F=\{(x,y)\mapsto\varphi(\langle w,x\rangle,y):w\in\mathcal{H}\}$，我们将说明以概率 1 有$R(F\circ S)\leq\rho BR/\sqrt{m}$。然后，该引理可由引理(26.5)得出。实际上，集合$F\circ S$可以被写为

$$F\circ S=\{(\varphi(\langle w,x_1\rangle,y_1),\dots,\varphi(\langle w,x_m\rangle,y_m)):w\in\mathcal{H}\}$$

$R(F\circ S)$的界可以直接由引理 26.9、引理 26.10 和假设(以概率 1 有$\|x\|_2\leq R$)得出。 ■

接下来我们引出一个基于先验理论的硬 SVM 的泛化界。简单来说，我们不容许有偏差项，考虑硬 SVM 问题：

$$\operatorname*{argmin}_w\|w\|^2\ \text{s.t.}\ \forall i,y_i\langle w,x_i\rangle\geq1$$

引理 26.13　*考虑一个在$\mathcal{X}\times\{\pm1\}$分布$\mathcal{D}$，存在一些向量$w^*$满足$\mathbb{P}_{(x,y)\sim\mathcal{D}}[y\langle w^*,x\rangle\geq1]=1$，并且以概率 1 有$\|x\|_2\leq R$。令$w_S$是等式(26.19)的输出。那么我们至少以$1-\delta$的概率，在样本$S\sim\mathcal{D}^m$的情况下，有*

$$\mathbb{P}_{(x,y)\sim\mathcal{D}}[y\neq\operatorname{sign}(\langle w_S,x\rangle)]\leq\frac{2R\|w^*\|}{\sqrt{m}}+(1+R\|w^*\|)\sqrt{\frac{2\ln(2/\delta)}{m}}$$

证明　在证明的全部过程中，假设损失函数是斜坡损失(见 15.2.3 节)。注意到，斜坡损失的范围是[0，1]，是一个 1-利普希茨函数。由于斜坡损失由 0-1 损失界定，我们有

$$\mathbb{P}_{(x,y)\sim\mathcal{D}}[y\neq\operatorname{sign}(\langle w_S,x\rangle)]\leq L_{\mathcal{D}}(w_S)$$ 334

令$B=\|w^*\|_2$，考虑集合$\mathcal{H}=\{w:\|w\|_2\leq B\}$。由硬 SVM 的定义和关于分布的假设，我们以概率 1 有$w_S\in\mathcal{H}$，并且$L_S(w_S)=0$。因此，根据引理 26.12 我们有

$$L_{\mathcal{D}}(w_S)\leq L_S(w_S)+\frac{2BR}{\sqrt{m}}+\sqrt{\frac{2\ln(2/\delta)}{m}}$$ ■

评注　这个引理意味着硬 SVM 问题的样本复杂度以$\frac{R^2\|w^*\|^2}{\varepsilon^2}$增长。用一个更加精

巧的分析和可分离性假设，样本复杂度可能以$\frac{R^2\|w^*\|^2}{\varepsilon^2}$增长。

在前面的定理中，误差界依赖于$\|w^*\|$，但它并不可知。接下来我们引出一个界，它依赖于SVM的输出的范数，因此可以被训练集计算出来。该证明与结构风险最小化中的界的证明类似。

定理 26.14 假设定理26.13的条件存在，那么，我们至少以$1-\delta$的概率，在样本$S\sim\mathcal{D}^m$的情况下，有

$$\mathop{\mathbb{P}}_{(x,y)\sim\mathcal{D}}[y\neq\operatorname{sign}(\langle w_S,x\rangle)]\leqslant\frac{4R\|w_S\|}{\sqrt{m}}+\sqrt{\frac{\ln\left(\frac{4\log(\|w_S\|)}{\delta}\right)}{m}}$$

证明 对于任意整数i，令$B_i=2^i$，$\mathcal{H}_i=\{w:\|w\|\leqslant B_i\}$，且令$\delta_i=\frac{\delta}{2i^2}$。固定$i$，使用定理26.12，我们至少以$1-\delta$的概率有

$$\forall w\in\mathcal{H}_i,L_{\mathcal{D}}(w)\leqslant L_S(w)+\frac{2B_iR}{\sqrt{m}}+\sqrt{\frac{2\ln(2/\delta_i)}{m}}$$

利用联合界以及$\sum_{i=1}^{\infty}\delta_i\leqslant\delta$，我们至少以$1-\delta$的概率对于所有$i$有上式成立。因此，对于所有的$w$，如果我们令$i=\lceil\log_2(\|w\|)\rceil$，那么$w\in\mathcal{H}_i$，$B_i\leqslant2\|w\|$，且$\frac{2}{\delta_i}=\frac{(2i)^2}{\delta}\leqslant\left(\frac{4\log_2(\|w\|)}{\delta}\right)^2$。因此

$$\begin{aligned}L_{\mathcal{D}}(w)&\leqslant L_S(w)+\frac{2B_iR}{\sqrt{m}}+\sqrt{\frac{2\ln(2/\delta_i)}{m}}\\&\leqslant L_S(w)+\frac{4\|w\|R}{\sqrt{m}}+\sqrt{\frac{4(\ln(4\log_2(\|w\|))+\ln(1/\delta))}{m}}\end{aligned}$$

特别地，它对于w_S成立，这就证明了我们的结论。 ■

标注 26.2 注意到我们已经证明的所有的误差界都不依赖w的维数。这个性质使得我们学习SVM的核函数时，w的维数可以很大。

26.4 低ℓ_1范数预测器的泛化误差界

在之前的章节中，我们导出了ℓ_2范数约束的线性预测器的泛化上界。本节中，我们考虑如下在ℓ_1范数约束下的一般形式。令$\mathcal{H}=\{w:\|w\|_1\leqslant B\}$为假设类集合，且$Z=\mathcal{X}\times\mathcal{Y}$为样本空间。假设损失函数，$\ell:\mathcal{H}\times Z\to\mathbb{R}$与等式(26.18)具有相同形式，即$\varphi:\mathbb{R}\times\mathcal{Y}\to\mathbb{R}$是定义在第一个变量上的$\rho$-利普希茨函数。下面的定理根据$\mathcal{H}$中所有预测器的经验损失定义了它们的泛化误差界。

[335]

定理 26.15 假设$\mathcal{D}$是定义在$\mathcal{X}\times\mathcal{Y}$上的一个分布，使得我们以概率1有$\|x\|_\infty\leqslant R$。令$\mathcal{H}=\{w\in\mathbb{R}^d:\|w\|_1\leqslant B\}$且$\ell:\mathcal{H}\times Z\to\mathbb{R}$是与等式(26.18)形式相同一个损失函数，使得对所有$y\in\mathcal{Y}$，$a\mapsto\varphi(a,y)$是一个ρ-利普希茨函数且使得$\max_{a\in[-BR,BR]}|\varphi(a,y)|\leqslant c$。那么，对任意$\delta\in(0,1)$，在$m$个独立同分布的样本上，我们以至少$1-\delta$的概率有

$$\forall w\in\mathcal{H},L_{\mathcal{D}}(w)\leqslant L_S(w)+2\rho BR\sqrt{\frac{2\log(2d)}{m}}+c\sqrt{\frac{2\ln(2/\delta)}{m}}$$

证明 证明过程与定理 26.12 的过程完全相同，除了根据引理 26.11 引出而不是引理 26.10。 ■

通过比较定理 26.12 和定理 26.15 的两个界，可以发现一些有意思的事情。抛开定理 26.15 额外的 $\log(d)$ 因子，两种界看起来十分相似。然而，参数 B 和 R 在两个界中有不同的含义。在定理 26.12 中，参数 B 对 w 施加了 ℓ_2 范数约束，而参数 R 给定了样本上一种较弱的 ℓ_2 范数假设。相反，定理 26.15 中，参数 B 对 w 施加 ℓ_1 范数约束（较 ℓ_2 限制更强），而参数 R 给定了样本上的 ℓ_∞ 范数假设（比 ℓ_2 范数假设更弱）。因此，对约束的选择应该根据样本集分布的先验以及合适的预测器的假设先验而定。

26.5 文献评注

使用拉德马赫理论来界定一致收敛来自于 Koltchinskii & Panchenko(2000)，Bartlett & Mendelson (2001)，Bartlett & Mendelson (2002)。另外，例如 Bousquet (2002)，Boucheron、Bousquet & Lugosi(2005)，Bartlett、Bousquet & Mendelson(2005)中也有阐述。我们关于压缩引理的证明来自 Kakade 和 Tewari 的课程笔记，Kakade、Sridharan 和 Tewari(2008)为不同的范数假设下的线性假设类的拉德马赫复杂度的界的推导定义了一个统一框架。

[336]

覆 盖 数

在这一章，我们用另一种方式来度量集合的复杂度，这种方式叫做覆盖数。

27.1 覆盖

定义 27.1（覆盖） 假设 $A\subset\mathbb{R}^m$ 是维度为 m 的向量集，如果对于所有的 $a\in A$，在 A' 中均存在 a' 满足 $|a-a'|\leqslant r$，我们就说在欧几里得度量空间中，集合 A' 覆盖集合 A。我们将能够 r 覆盖集合 A 的最小集合 A' 定义为集的势，并用 $N(r, A)$ 表示。

例 27.1（子空间） 假设 $A\subset\mathbb{R}^m$，令 $c=\max_{a\in A}|a|$，假设 A 可以映射到 $\mathbb{R}^m$ 的 d 维子空间中，那么 $N(r, A)\leqslant(2c\sqrt{d}/r)^d$。要证明等式成立，假设 $v_1, \cdots, v_d$ 是子空间的正交向量基。那么，对于任意的 $a\in A$，均能表示成 $a=\sum_{i=1}^{d}a_iv_i$，其中 $\|a\|_\infty\leqslant|a|_2=|a|_2\leqslant c$。令 $\varepsilon\in\mathbb{R}$，考虑集合

$$A'=\left\{\sum_{i=1}^{d}a'_iv_i:\forall i,a'_i\in\{-c, -c+\varepsilon, -c+2\varepsilon,\cdots,c\}\right\}$$

给定 $a\in A$，其中 a 满足约束条件：$a=\sum_{i=1}^{d}a_iv_i$，$|a|_\infty\leqslant c$，那么存在 $a'\in A'$，使得

$$\|a-a'\|^2=\|\sum_i(a'_i-a_i)v_i\|^2\leqslant\varepsilon^2\sum_i\|v_i\|^2\leqslant\varepsilon^2d$$

令 $\varepsilon=r/\sqrt{d}$，则 $|a-a'|\leqslant r$，因此 A' 是 A 的 r 覆盖，进而可以推导出：

$$N(r,A)\leqslant|A'|=\left(\frac{2c}{\varepsilon}\right)^d=\left(\frac{2c\sqrt{d}}{r}\right)^d$$ ◀

性质

从定义中很容易得出以下引理。

引理 27.2 对于任意的 $A\subset\mathbb{R}^m$，标量 $c>0$，向量 $a_0\in\mathbb{R}^m$，我们可以得出

$$\forall r>0, N(r,\{ca+a_0:a\in A\})\leqslant N(cr,A)$$

下面，我们将推导一个收敛准则。

引理 27.3 对于每个 $i\in[m]$，令 $\varphi_i:\mathbb{R}\to\mathbb{R}$ 是一个 ρ 利普希茨函数，也就是对于所有 $\alpha,\beta\in\mathbb{R}$，满足 $|\varphi_i(\alpha)-\varphi_i(\beta)|\leqslant\rho|\alpha-\beta|$。对于 $a\in\mathbb{R}^m$，令 $\boldsymbol{\varphi}(a)$ 表示向量 $(\varphi_1(a_1), \cdots, \varphi_m(a_m))$。令 $\boldsymbol{\varphi}\circ A=\{\boldsymbol{\varphi}(a):a\in A\}$，那么

$$N(\rho r,\boldsymbol{\varphi}\circ A)\leqslant N(r,A)$$

证明 定义 $B=\boldsymbol{\varphi}\circ A$，令 A' 是 A 的 r 覆盖，定义 $B'=\boldsymbol{\varphi}\circ A'$，那么，对于所有的 $a\in A$，存在 $a'\in A'$，使得 $|a-a'|\leqslant r$，因此，

$$|\boldsymbol{\varphi}(a)-\boldsymbol{\varphi}(a')|^2=\sum_i(\varphi_i(a_i)-\varphi_i(a'_i))^2\leqslant\rho^2\sum_i(a_i-a'_i)^2\leqslant(\rho r)^2$$

因此，B'是B的ρr覆盖。 ■

27.2 通过链式反应从覆盖到拉德马赫复杂度

下面的引理基于覆盖数$N(r, A)$给出了集合A拉德马赫复杂度的边界，这种由Dudley最早提出的技巧称为链式反应。

引理27.4 令$c=\min_{a}\max_{a\in A}|a-\bar{a}|$，那么，对于任意整数$M\geq 0$，

$$R(A)\leq \frac{c2^{-M}}{\sqrt{m}}+\frac{6c}{m}\sum_{k=1}^{M}2^{-k}\sqrt{\log(N(c2^{-k},A))}$$

证明 令$\bar{a}$是给定的目标函数c的最小值，基于引理26.6，假设$\bar{a}=0$，我们能够分析拉德马赫复杂度。

考虑集合$B_0=\{\mathbf{0}\}$，并且注意它是集合A的c覆盖。令集合B_1，…，B_M分别是集合A的最小$c2^{-k}$覆盖。令$a^*=\operatorname{argmax}_{a\in A}\langle\boldsymbol{\sigma}, a\rangle$(如果存在多个最大值，那么任意选择一个，如果不存在最大值，那么我们选择使$\langle\boldsymbol{\sigma}, a^*\rangle$接近于最大值的$a^*$)。注意$a^*$是$\boldsymbol{\sigma}$的一个函数，对于每一个$k$，令$b^{(k)}$是$B_k$中$a^*$的最近邻(因此$b^{(k)}$也是$\boldsymbol{\sigma}$的一个函数)。使用三角不等式：

$$\|b^{(k)}-b^{(k-1)}\|\leq\|b^{(k)}-a^*\|+\|a^*-b^{(k-1)}\|\leq c(2^{-k}-2^{-(k-1)})=3c2^{-k}$$

对于每一个k，定义集合

$$\hat{B}_k=\{(a-a'): a\in B_k, a'\in B_{k-1}, \|a-a'\|\leq 3c2^{-k}\}$$ [338]

可得：

$$\begin{aligned}R(A)&=\frac{1}{m}\mathbb{E}\langle\boldsymbol{\sigma}, a^*\rangle\\&=\frac{1}{m}\mathbb{E}\Big[\langle\boldsymbol{\sigma}, a^*-b^{(M)}\rangle+\sum_{k=1}^{M}\langle\boldsymbol{\sigma}, b^{(k)}-b^{(k-1)}\rangle\Big]\\&\leq\frac{1}{m}\mathbb{E}\Big[\|\boldsymbol{\sigma}\|\,\|a^*-b^{(M)}\|\Big]+\sum_{k=1}^{M}\frac{1}{m}\mathbb{E}\Big[\sup_{a\in\hat{B}_k}\langle\boldsymbol{\sigma}, a\rangle\Big]\end{aligned}$$

由于$\|\boldsymbol{\sigma}\|=\sqrt{m}$，$\|a^*-b^{(M)}\|\leq c2^{-M}$，因此第一项最大值是$\frac{c}{\sqrt{m}}2^{-M}$，另外，通过马萨特引理我们有

$$\frac{1}{m}E\sup_{a\in\hat{B}_k}\langle\boldsymbol{\sigma}, a\rangle\leq 3c2^{-k}\frac{\sqrt{2\log(N(c2^{-k},A)^2)}}{m}=6c2^{-k}\frac{\sqrt{\log(N(c2^{-k},A))}}{m}$$

因此，

$$R(A)\leq\frac{c2^{-M}}{\sqrt{m}}+\frac{6c}{m}\sum_{k=1}^{M}2^{-k}\sqrt{\log(N(c2^{-k},A))}$$ ■

我们可以得到以下推论：

引理27.5 假设存在α，$\beta>0$，对于任意的$k\geq 1$，我们有

$$\sqrt{\log(N(c2^{-k},A))}\leq\alpha+\beta k$$

那么

$$R(A)\leq\frac{6c}{m}(\alpha+2\beta)$$

证明　当$M\to\infty$时，边界满足引理27.4，注意$\sum_{k=1}^{\infty}2^{-k}=1$，$\sum_{k=1}^{\infty}k2^{-k}=2$。

例27.2 集合A存在于$\mathbb{R}^n$的d维子空间，且有$c=\max_{a\in A}\|a\|$，我们已经证明$N(r,A)\leqslant\left(\frac{2c\sqrt{d}}{r}\right)^d$，所以，对于任意$k$，

$$\begin{aligned}\sqrt{\log(N(c2^{-k},A))}&\leqslant\sqrt{d\log(2^{k+1}\sqrt{d})}\\&\leqslant\sqrt{d\log(2\sqrt{d})}+\sqrt{kd}\\&\leqslant\sqrt{d\log(2\sqrt{d})}+\sqrt{dk}\end{aligned}$$

因此，引理27.5得出

$$R(A)\leqslant\frac{6c}{m}\left(\sqrt{d\log(2\sqrt{d})}+2\sqrt{d}\right)=O\left(\frac{c\sqrt{d\log(d)}}{m}\right)$$ ◀

27.3　文献评注

335 ~ 340

链式反应技术由Dudley(1987)最早提出。如果想要进一步学习覆盖数以及其他用来界定一致收敛速率的复杂度测量方法，可以参考阅读Anthony & Bartlet(1999)。

学习理论基本定理的证明

本章我们证明第 6 章的定理 6.8。该定理的条件是：$\mathcal{H}$是由从定义域$\mathcal{X}$映射到值域$\{0,1\}$的函数组成的假设类，损失函数是 0-1 损失，并且 $\mathrm{VCdim}(\mathcal{H})=d<\infty$。

我们会证明可实现和不可知两种情况下的上界，以及不可知情况的下界。可实现情况下界的证明则留作练习。

28.1 不可知情况的上界

关于上界，我们需要证明的是：存在常数 C 使得$\mathcal{H}$为不可知 PAC 可学习，并且样本复杂度满足：

$$m_{\mathcal{H}}(\varepsilon,\delta)\leq C\frac{d+\ln(1/\delta)}{\varepsilon^2}$$

我们将证明一个略为松弛的界：

$$m_{\mathcal{H}}(\varepsilon,\delta)\leq C\frac{d\log(d/\varepsilon)+\ln(1/\delta)}{\varepsilon^2} \tag{28.1}$$

而定理中更为紧的界则需要更为复杂的证明，必须对拉德马赫复杂度做更仔细地分析，其中用到了一种称为“链”的技术。这些内容超出了本书范畴。

为了证明式(28.1)，只需要说明，对样本大小

$$m\geq 4\frac{32d}{\varepsilon^2}\cdot\log\left(\frac{64d}{\varepsilon^2}\right)+\frac{8}{\varepsilon^2}\cdot(8d\log(e/d)+2\log(4/\delta))$$

采用 ERM 准则能够得到关于$\mathcal{H}$的 ε，δ-学习器。我们将在定理 26.5 的基础上证明该结果。

令$(x_1,y_1),\ldots,(x_m,y_m)$为用于分类的训练集。回顾 Sauer-Shelah 引理，如果 $\mathrm{VCdim}(\mathcal{H})=d$，那么

$$|\{(h(x_1),\ldots,h(x_m)):h\in\mathcal{H}\}|\leq\left(\frac{em}{d}\right)^d$$

构造 $A=\{(\mathbb{1}_{[h(x_1)\neq y_1]},\ldots,\mathbb{1}_{[h(x_m)\neq y_m]}):h\in\mathcal{H}\}$。该式清晰地表示出

$$|A|\leq\left(\frac{em}{d}\right)^d$$

将此式与引理 26.8 结合起来我们可以得到如下关于拉德马赫复杂度的界：

$$R(A)\leq\sqrt{\frac{2d\log(em/d)}{m}}$$

再利用定理 26.5 我们得到，以至少 $1-\delta$ 的概率，对每个 $h\in\mathcal{H}$能推出

$$L_{\mathcal{D}}(h)-L_S(h)\leq\sqrt{\frac{8d\log(em/d)}{m}}+\sqrt{\frac{2\log(2/\delta)}{m}}$$

重复之前有关减去 0-1 损失的讨论，然后运用联合界，我们得到，以至少 $1-\delta$ 的概率，对每个 $h\in\mathcal{H}$有下式成立：

$$|L_{\mathcal{D}}(h)-L_S(h)|\leq\sqrt{\frac{8d\log(em/d)}{m}}+\sqrt{\frac{2\log(4/\delta)}{m}}$$

$$\leqslant 2\sqrt{\frac{8d\log(em/d)+2\log(4/\delta)}{m}}$$

为保证其小于ε，我们需要

$$m \geqslant \frac{4}{\varepsilon^2}\cdot(8d\log_2(m)+8d\log(e/d)+2\log(4/\delta))$$

使用引理A.2，使得上一不等式成立的充分条件是

$$m \geqslant 4\frac{32d}{\varepsilon^2}\cdot\log\left(\frac{64d}{\varepsilon^2}\right)+\frac{8}{\varepsilon^2}\cdot(8d\log(e/d)-2\log(4/\delta))$$

28.2　不可知情况的下界

这里我们需要证明，存在常数C使得$\mathcal{H}$为不可知PAC可学习，并且样本复杂度满足

$$m_{\mathcal{H}}(\varepsilon,\delta) \geqslant C\frac{d+\ln(1/\delta)}{\varepsilon^2}$$

分两步来证明该下界。第一，我们证明$m(\varepsilon,\delta)\geqslant 0.5\log(1/(4\delta))/\varepsilon^2$。第二，我们证明对每个$\delta\leqslant 1/8$有$m(\varepsilon,\delta)\geqslant 8d/\varepsilon^2$，由这两个界即得证。

28.2.1　证明$m(\varepsilon,\delta)\geqslant 0.5\log(1/(4\delta))/\varepsilon^2$

我们首先证明对于每个$\varepsilon<1/\sqrt{2}$和$\delta\in(0,1)$，有$m(\varepsilon,\delta)\geqslant 0.5\log(1/(4\delta))/\varepsilon^2$。为得到此结论，我们说明对于$m\leqslant 0.5\log(1/(4\delta))/\varepsilon^2$，$\mathcal{H}$是不可学习的。

挑选被$\mathcal{H}$打散的一个样本。即是，令c为一个样本，使得存在h^+，$h^-\in\mathcal{H}$，其中$h^+(c)=1$和$h^-(c)=-1$。定义两个分布$\mathcal{D}_+$和$\mathcal{D}_-$，对于$b\in\{\pm1\}$有

[342]

$$\mathcal{D}_b(\{(x,y)\})=\begin{cases}\dfrac{1+yb\varepsilon}{2} & 若\ x=c\\ 0 & 其他\end{cases}$$

即是，分布集中在两个样本$(c,1)$和$(c,-1)$，其中(c,b)的概率是$\frac{1+b\varepsilon}{2}$，$(c,-b)$的概率是$\frac{1-b\varepsilon}{2}$。

令A为任意算法。从$\mathcal{D}_b$采样得到的训练集的样式是$S=(c,y_1),\cdots,(c,y_m)$。因此，训练集被向量$y=(y_1,\cdots,y_m)\in\{\pm1\}^m$完全刻画。一旦收到训练集$S$，算法$A$将返回假设$h:\mathcal{X}\to\{\pm1\}$。既然定义在$\mathcal{D}_b$上的$A$的误差取决于$h(c)$，我们可以将$A$看作从$\{\pm1\}^m$到$\{\pm1\}$的映射，因此，我们用取值于$\{\pm1\}$的$A(y)$代表对$h(c)$的预测值，其中$h$是算法$A$在收到训练集$S=(c,y_1),\cdots,(c,y_m)$后输出的假设。

注意到对任意假设h有

$$L_{\mathcal{D}_b}(h)=\frac{1-h(c)b\varepsilon}{2}$$

特别地，用h_b表示贝叶斯最优假设，那么

$$L_{\mathcal{D}_b}(A(y))-L_{\mathcal{D}_b}(h_b)=\frac{1-A(y)b\varepsilon}{2}-\frac{1-\varepsilon}{2}=\begin{cases}\varepsilon & 若\ A(y)\neq b\\ 0 & 其他\end{cases}$$

固定A。对于$b\in\{\pm1\}$，令$Y^b=\{y\in\{0,1\}^m:A(y)\neq b\}$，那么分布$\mathcal{D}_b$引出在$\{\pm1\}^m$上的分布$P_b$。因此，

$$\mathbb{E}[L_{\mathcal{D}_b}(A(y))-L_{\mathcal{D}_b}(h_b)-\varepsilon]-\mathcal{D}_b(Y^b)=\sum_y P_b[y]\mathbb{1}_{[A(y)\neq b]}$$

构造 $N^+=\{y:|\{i:y_i=1\}|\geq m/2\}$ 和 $N^-=\{\pm1\}^m\setminus N^+$。注意到对任意 $y\in N^+$ 有 $P_-[y]\geq P_+[y]$，并且对任意 $y\in N^-$ 有 $P_+[y]\geq P_-[y]$。因此，

$$\begin{aligned}
&\max_{b\in\{\pm1\}}\mathbb{P}[L_{\mathcal{D}_b}(A(y))-L_{\mathcal{D}_b}(h_b)-\varepsilon]\\
&=\max_{b\in\{\pm1\}}\sum_y P_b[y]\mathbb{1}_{[A(y)\neq b]}\\
&\geq\frac{1}{2}\sum_y P_+[y]\mathbb{1}_{[A(y)\neq-1]}+\frac{1}{2}\sum_y P_-[y]\mathbb{1}_{[A(y)\neq1]}\\
&=\frac{1}{2}\sum_{y\in N^+}(P_+[y]\mathbb{1}_{[A(y)\neq+1]}+P_-[y]\mathbb{1}_{[A(y)\neq-1]})\\
&\quad+\frac{1}{2}\sum_{y\in N^-}(P_+[y]\mathbb{1}_{[A(y)\neq+1]}+P_-[y]\mathbb{1}_{[A(y)\neq-1]})\\
&\geq\frac{1}{2}\sum_{y\in N^+}(P_-[y]\mathbb{1}_{[A(y)\neq+1]}+P_-[y]\mathbb{1}_{[A(y)\neq-1]})\\
&\quad+\frac{1}{2}\sum_{y\in N^-}(P_+[y]\mathbb{1}_{[A(y)\neq+1]}+P_+[y]\mathbb{1}_{[A(y)\neq-1]})\\
&=\frac{1}{2}\sum_{y\in N^+}P_-[y]+\frac{1}{2}\sum_{y\in N^-}P_+[y]
\end{aligned}$$

343

接下来注意到 $\sum_{y\in N^+}P_-[y]=\sum_{y\in N^-}P_+[y]$，其值均为服从二项分布$(m,(1-\varepsilon)/2)$的随机变量的值大于 $m/2$ 的概率。使用引理 B.11，该概率的下界为

$$\frac{1}{2}\left(1-\sqrt{1-\exp(-m\varepsilon^2/(1-\varepsilon^2))}\right)\geq\frac{1}{2}\left(1-\sqrt{1-\exp(-2m\varepsilon^2)}\right)$$

该不等式中我们利用了假设 $\varepsilon^2<1/2$。由此如果 $m\leq0.5\log(1/(4\delta))/\varepsilon^2$ 那么存在 b 使得

$$\mathbb{P}[L_{\mathcal{D}_b}(A(y))-L_{\mathcal{D}_b}(h_b)-\varepsilon]\geq\frac{1}{2}\left(1-\sqrt{1-\sqrt{4\delta}}\right)\geq\delta$$

遵从标准的代数操作，可以得到最后一个不等式。以上推出了需要证明的结论。

28.2.2 证明 $m(\varepsilon,1/8)\geq8d/\varepsilon^2$

我们将证明对每个 $\varepsilon<1/(8\sqrt{2})$ 有 $m(\varepsilon,\delta)\geq8d/\varepsilon^2$。

令 $\rho=8\varepsilon$，可以看到 $\rho\in(0,1/\sqrt{2})$。我们将构造如下分布族。首先，令 $C=\{c_1,\cdots,c_d\}$ 为被$\mathcal{H}$打散的 d 个样例的集合。第二，对每个向量 $(b_1,\cdots b_d)\in\{\pm1\}^d$，定义分布$\mathcal{D}_b$ 满足

$$\mathcal{D}_b(\{x,y\})=\begin{cases}\frac{1}{d}\cdot\frac{1+yb_i\rho}{2} & \text{若 }\exists i,x=c_i\\ 0 & \text{其他}\end{cases}$$

即是，为了从$\mathcal{D}_b$ 中采样，我们首先从集合 C 中等概率地随机挑选一个元素 c_i，然后以概率$(1+\rho)/2$ 将标志设置为 b_i，以概率$(1-\rho)/2$ 设置为 $-b_i$。

很容易验证对于分布$\mathcal{D}_b$ 的贝叶斯最优预测器是对所有 $i\in[d]$满足 $h(c_i)=b_i$ 的假设 $h\in\mathcal{H}$，其误差为$(1-\rho)/2$。此外，对任意函数 $f:\mathcal{X}\to\{\pm1\}$，容易验证

$$L_{\mathcal{D}_b}(f)=\frac{1+\rho}{2}\cdot\frac{|\{i\in[d]:f(c_i)\neq b_i\}|}{d}+\frac{1-\rho}{2}\cdot\frac{|\{i\in[d]:f(c_i)=b_i\}|}{d}$$

因此，

[344]

$$L_{\mathcal{D}_0}(f) = \min_{h\in\mathcal{H}} L_{\mathcal{D}_0}(h) - \rho\cdot\frac{|\{i\in[d]: f(c_i)\neq b_i\}|}{d} \tag{28.2}$$

接下来，固定某个学习算法 A。如同证明“没有免费的午餐”定理中那样，可以得到

$$\max_{\mathcal{D}_0: A\in\{\pm1\}^d} \mathbb{E}_{S\sim\mathcal{D}_0^m}\left[L_{\mathcal{D}_0}(A(S)) - \min_{h\in\mathcal{H}} L_{\mathcal{D}_0}(h)\right] \tag{28.3}$$

$$\geq \mathbb{E}_{\mathcal{D}_0\sim U(\{\pm1\}^d)} \mathbb{E}_{S\sim\mathcal{D}_0^m}\left[L_{\mathcal{D}_0}(A(S)) - \min_{h\in\mathcal{H}} L_{\mathcal{D}_0}(h)\right] \tag{28.4}$$

$$= \mathbb{E}_{\mathcal{D}_0\sim U(\{\pm1\}^d)} \mathbb{E}_{S\sim\mathcal{D}_0^m}\left[\rho\cdot\frac{|\{i\in[d]: A(S)(c_i)\neq b_i\}|}{d}\right] \tag{28.5}$$

$$= \frac{\rho}{d}\sum_{i=1}^{d} \mathbb{E}_{\mathcal{D}_0\sim U(\{\pm1\}^d)} \mathbb{E}_{S\sim\mathcal{D}_0^m} \mathbb{1}_{[A(S)(c_i)\neq b_i]} \tag{28.6}$$

其中第一个不等式来自式(28.2)。此外，根据分布$\mathcal{D}_0$的定义，我们可以首先从分布 $S\sim\mathcal{D}_0$ 中采样$(j_1, \ldots, j_m)\sim U([d])^m$，令 $x_t=c_{j_t}$，最后根据 $\mathbb{P}[y_t=b_{j_t}]=(1+\rho)/2$ 采样 y_t。简化符号，使用 $y\sim b$ 来代表根据 $\mathbb{P}[y=b]=(1+\rho)/2$ 采样。因此，式(28.6)右边等价于

$$\frac{\rho}{d}\sum_{i=1}^{d} \mathbb{E}_{j\sim U([d])^m} \mathbb{E}_{b\sim U(\{\pm1\}^d)} \mathbb{E}_{\forall t, y_t\sim b_{j_t}} \mathbb{1}_{[A(S)(c_i)\neq b_i]}. \tag{28.7}$$

现在分两步来推进。首先，我们证明在所有的学习算法中，最小化式(28.7)(因此也最小化式(28.4))的 A 就是最大似然学习规则，用 A_{ML} 来表示。正式地，对每个 i，$A_{ML}(S)(c_i)$ 等价于在集合$\{y_r: r\in[m], x_r=c_i\}$上进行多数票决。第二步，我们针对 A_{ML} 算法降低式(28.7)的界。

引理 28.1 *在所有的算法中，最小化式(28.4)的算法 A 即为最大似然算法 A_{ML}，其被定义为*

$$\forall i,\quad A_{ML}(S)(c_i) = \operatorname{sign}\Big(\sum_{r: x_r=c_i} y_r\Big)$$

证明 固定某个 $j\in[d]^m$，注意到给定 j 和 $y\in\{\pm1\}^m$，那么训练集 S 被完全确定。因此我们用 $A(j, y)$来代替 $A(S)$。同时固定 $i\in[d]$，用 b^{-i} 代表序列$(b_1, \ldots, b_{i-1}, b_{i+1}, \ldots, b_m)$。同时，对任意 $y\in\{\pm1\}^m$，令 y^i 代表使得 $j_r=i$ 的指标对应的 y 中的元素，y^{-i} 则代表 y 中其他元素。可以得到

$$\begin{aligned}
&\mathbb{E}_{b\sim U(\{\pm1\}^d)} \mathbb{E}_{\forall t, y_t\sim b_{j_t}} \mathbb{1}_{[A(S)(c_i)\neq b_i]} \\
&= \frac{1}{2}\sum_{b_i\in\{\pm1\}} \mathbb{E}_{b^{-i}\sim U(\{\pm1\}^{m-1})} \sum_{y} P[y^{-i}, b_i]\, \mathbb{1}_{[A(j, y)(c_i)\neq b_i]} \\
&= \mathbb{E}_{b^{-i}\sim U(\{\pm1\}^{m-1})} \sum_{y^{-i}} P[y^{-i}|b^{-i}]\, \frac{1}{2}\sum_{j}\Big(\sum_{b_i\in\{\pm1\}} P[y^{i}|b_i]\, \mathbb{1}_{[A(j, y)(c_i)\neq b_i]}\Big)
\end{aligned}$$

[345]

当 $A(j, y)(c_i)$使得在 $b_i\in\{\pm1\}$上的 $P[y^i|b_i]$最大时，上式括号中的和也达到最小。此时，$A(j, y)(c_i)$也正是最大似然规则，以此类推，对所有的 i 都成立。结论得证。 ■

固定 i，对每个 j，令 $n_i(j)=|\{t: j_t=i\}|$代表样例为 c_i 的样例个数，采用最大似然规则，可以知道下式

$$\mathbb{E}_{b\sim U(\{\pm1\}^d)} \mathbb{E}_{\forall t, y_t\sim b_{j_t}} \mathbb{1}_{[A_{ML}(S)(c_i)\neq b_i]}$$

正好是满足二项分布$(n_i(j),\ (1-\rho)/2)$的随机变量的值大于 $n_i(j)/2$ 的概率。利用引理 B.11，再加上假设 $\rho^2\leqslant 1/2$，得到

$$P[B\geqslant n_i(j)/2]\leqslant \frac{1}{2}\left(1-\sqrt{1-e^{-2n_i(j)\rho^2}}\right)$$

因此可以推出

$$\begin{aligned}
&\frac{\rho}{d}\sum_{i=1}^{d}\ \mathop{\mathbb{E}}_{j\sim U(\{0,1\}^n)}\ \mathop{\mathbb{E}}_{h\sim U(\pm 1)^d}\ \mathop{\mathbb{E}}_{\forall i\in[d]\setminus\{j\}}\ \mathbb{1}_{[A(S)(i)\neq h_i]}\\
\geqslant&\frac{\rho}{2d}\sum_{i=1}^{d}\ \mathop{\mathbb{E}}_{j\sim U(\{0,1\}^n)}\left(1-\sqrt{1-e^{-2\rho^2 n_i(j)}}\right)\\
\geqslant&\frac{\rho}{2d}\sum_{i=1}^{d}\ \mathop{\mathbb{E}}_{j\sim U(\{0,1\}^n)}\left(1-\sqrt{2\rho^2 n_i(j)}\right)
\end{aligned}$$

最后一个不等式中我们使用了不等式 $1-e^{-a}\leqslant a$。

因为平方根函数是凹的，应用詹生不等式，可以得到上式的下界为

$$\begin{aligned}
&\geqslant\frac{\rho}{2d}\sum_{i=1}^{d}\left(1-\sqrt{2\rho^2\ \mathop{\mathbb{E}}_{j\sim U(\{0,1\}^n)}\ n_i(j)}\right)\\
&=\frac{\rho}{2d}\sum_{i=1}^{d}\left(1-\sqrt{2\rho^2 m/d}\right)\\
&=\frac{\rho}{2}\left(1-\sqrt{2\rho^2 m/d}\right)
\end{aligned}$$

只要 $m<d/8\rho^2$，这一项就会比 $\rho/4$ 大。

总的来说，我们已经证明了，如果 $m<d/8\rho^2$，那么对于任何算法，都存在一个分布使得

$$\mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}\left[L_{\mathcal{D}}(A(S))-\min_{h\in\mathcal{H}}L_{\mathcal{D}}(h)\right]\geqslant\rho/4$$

[346]

最后，令 $\Delta=\frac{1}{\rho}(L_{\mathcal{D}}(A(S))-\min_{h\in\mathcal{H}}L_{\mathcal{D}}(h))$，又注意到 $\Delta\in[0,1]$（参见式(28.5)）。因此，利用引理 B.1，得到

$$\begin{aligned}
\mathbb{P}\left[L_{\mathcal{D}}(A(S))-\min_{h\in\mathcal{H}}L_{\mathcal{D}}(h)>\varepsilon\right]&=\mathbb{P}\left[\Delta>\frac{\varepsilon}{\rho}\right]\geqslant\mathbb{E}[\Delta]-\frac{\varepsilon}{\rho}\\
&\geqslant\frac{1}{4}-\frac{\varepsilon}{\rho}
\end{aligned}$$

选择 $\rho=8\varepsilon$，我们得到结论：如果 $m(\varepsilon,\delta)<8d/\varepsilon^2$，那么以至少 1/8 的概率得到 $L_{\mathcal{D}}(A(S))-\min_{h\in\mathcal{H}}L_{\mathcal{D}}(h)\geqslant\varepsilon$。

28.3　可实现情况的上界

这里我们需要证明存在常数 C 使得 $\mathcal{H}$ 为 PAC 可学习，并且样本复杂度满足

$$m_{\mathcal{H}}(\varepsilon,\delta)\leqslant C\frac{d\ln(1/\varepsilon)+\ln(1/\delta)}{\varepsilon}$$

我们将证明对于 $m\geqslant C\frac{d\ln(1/\varepsilon)+\ln(1/\delta)}{\varepsilon}$，采用 ERM 准则，$\mathcal{H}$ 是可学习的。我们将在 ε-网的概念下证明上述论断。

定义 28.2（ε-网）　令 $\mathcal{X}$ 为定义域，$S\subset\mathcal{X}$ 在分布 $\mathcal{D}$ 上对于 $\mathcal{H}\subset 2^{\mathcal{X}}$ 是一个 ε-网，如果下式成立

$$\forall h \in \mathcal{H}:\ \mathcal{D}(h) \geq \varepsilon \Rightarrow h \cap S \neq \varnothing$$

定理 28.3 令$\mathcal{H} \subset 2^{\mathcal{X}}$，且$\mathrm{VCdim}(\mathcal{H}) = d$。固定$\varepsilon \in (0, 1)$，$\delta \in (0, 1/4)$并且令

$$m \geq \frac{8}{\varepsilon}\left(2d\log\left(\frac{16e}{\varepsilon}\right) + \log\left(\frac{2}{\delta}\right)\right)$$

那么，对于样本$S \sim \mathcal{D}^m$，以至少$1-\delta$的概率有S关于$\mathcal{H}$是一个ε-网。

证明 令

$$B = \{S \subset \mathcal{X} \mid |S| = m, \exists h \in \mathcal{H}, \mathcal{D}(h) \geq \varepsilon, h \cap S = \varnothing\}$$

为不是ε-网的集合所构成的集合。我们需要为$\mathbb{P}[S \in B]$定界。定义

$$B' = \{(S, T) \subset \mathcal{X} : |S| = |T| = m, \exists h \in \mathcal{H}, \mathcal{D}(h) \geq \varepsilon, h \cap S = \varnothing, |T \cap h| > \frac{\varepsilon m}{2}\}$$

论断 1

$$\mathbb{P}[S \in B] \leq 2\,\mathbb{P}[(S, T) \in B']$$

论断 1 的证明 既然S和T都是独立选取的，那么我们可以推出

[347]

$$\mathbb{P}[(S, T) \in B'] = \mathop{\mathbb{E}}_{(S,T)\sim\mathcal{D}^{2m}}[\mathbf{1}_{[(S,T)\in B']}] = \mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}\left[\mathop{\mathbb{E}}_{T\sim\mathcal{D}^m}[\mathbf{1}_{[(S,T)\in B']}]\right]$$

注意到$(S, T) \in B'$暗示了$S \in B$。因此$\mathbf{1}_{[(S,T)\in B']} = \mathbf{1}_{[(S,T)\in B']}\mathbf{1}_{[S\in B]}$，并得到

$$\begin{aligned}\mathbb{P}[(S, T) \in B'] &= \mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}\mathop{\mathbb{E}}_{T\sim\mathcal{D}^m}\mathbf{1}_{[(S,T)\in B']}\mathbf{1}_{[S\in B]} \\ &= \mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}[\mathbf{1}_{[S\in B]}\mathop{\mathbb{E}}_{T\sim\mathcal{D}^m}\mathbf{1}_{[(S,T)\in B']}]\end{aligned}$$

固定S。那么，无论是$\mathbf{1}_{[S\in B]} = 0$还是$S \in B$，都存在$h_S$使得$\mathcal{D}(h_S) \geq \varepsilon$并且$|h_S \cap S| = 0$。这个结论来自于$(S, T) \in B'$的充分条件是$|T \cap h_S| > \frac{\varepsilon m}{2}$。因此，只要$S \in B$都有

$$\mathop{\mathbb{E}}_{T\sim\mathcal{D}^m}\mathbf{1}_{[(S,T)\in B']} \geq \mathop{\mathbb{P}}_{T\sim\mathcal{D}^m}\left[|T \cap h_S| > \frac{\varepsilon m}{2}\right]$$

但是，因为现在假设$S \in B$，所以有$\mathcal{D}(h_S) = p \geq \varepsilon$。因此，$|T \cap h_S|$是满足参数为$p$(单次试验的成功概率)和$m$(试验总次数)的二项分布的随机变量。切尔诺夫不等式表明

$$\mathbb{P}\left[|T \cap h_S| \leq \frac{pm}{2}\right] \leq e^{-2(p-p/2)^2/m} = e^{-mp/2} \leq e^{-m\varepsilon/2} \leq e^{-d\log(16/\varepsilon)/2} = \delta^{d/2} \leq 1/2$$

因此，

$$\mathbb{P}\left[|T \cap h_S| > \frac{\varepsilon m}{2}\right] = 1 - \mathbb{P}\left[|T \cap h_S| \leq \frac{\varepsilon m}{2}\right] \geq 1 - \mathbb{P}\left[|T \cap h_S| \leq \frac{pm}{2}\right] \geq 1/2$$

结合前述所有，我们完成论断 1 的证明。

论断 2(对称性)

$$\mathbb{P}[(S, T) \in B'] \leq e^{-m/8}\,\mathrm{De}(2m)$$

论断 2 的证明 为了简化符号，令$\alpha = m\varepsilon/2$，且对于序列$A = (x_1, \cdots, x_{2m})$，令$A_1 = (x_1, \cdots, x_m)$。利用$B'$的定义，我们得到

$$\begin{aligned}\mathbb{P}[A \in B'] &= \mathop{\mathbb{E}}_{A\sim\mathcal{D}^{2m}}\max_{h\in\mathcal{H}}\mathbf{1}_{[\mathcal{D}(h)\geq\varepsilon]}\mathbf{1}_{[|h\cap A_1|=0]}\mathbf{1}_{[|h\cap A|\geq\alpha]} \\ &\leq \mathop{\mathbb{E}}_{A\sim\mathcal{D}^{2m}}\max_{h\in\mathcal{H}}\mathbf{1}_{[|h\cap A_1|=0]}\mathbf{1}_{[|h\cap A|\geq\alpha]}\end{aligned}$$

现在，通过$\mathcal{H}_A$来定义在A上的不同假设的有效数字，即，$\mathcal{H}_A = \{h \cap A : h \in \mathcal{H}\}$。由此，

$$\mathbb{P}[A \in B'] = \mathop{\mathbb{E}}_{A\sim\mathcal{D}^{2m}}\max_{h\in\mathcal{H}_A}\mathbf{1}_{[|h\cap A_1|=0]}\mathbf{1}_{[|h\cap A|\geq\alpha]}$$

$$\leqslant \mathop{\mathbb{E}}_{A\sim\mathcal{D}^{2m}} \max_{h\in\mathcal{H}} \mathbb{1}_{[|h\cap A_1|=0]} \mathbb{1}_{[|h\cap A|\geqslant\alpha]}$$

令 $J=\{j\subset[2m]:|j|=m\}$。对任意 $j\in J$ 和 $A=(x_1,\cdots,x_{2m})$ 定义 $A_j=(x_{j_1},\cdots,x_{j_m})$。因为 A 中元素是独立同分布选取出来的，那么对任意 $j\in J$ 和任意函数 $f(A,A_1)$ 都满足 $\mathbb{E}_{A\sim\mathcal{D}^{2m}}[f(A,A_1)]=\mathbb{E}_{A\sim\mathcal{D}^{2m}}[f(A,A_j)]$。既然该等式对任意 j 都成立，那么它对从 J 中随机选取的 j 的期望也成立。特别地，它对函数 $f(A,A_1)=\sum_{h\in\mathcal{H}_A}\mathbb{1}_{[|h\cap A_1|=0]}\mathbb{1}_{[|h\cap A|\geqslant\alpha]}$ 也成立。因此可以得到

348

$$\begin{aligned}\mathbb{P}[A\in B'] &\leqslant \mathop{\mathbb{E}}_{A\sim\mathcal{D}^{2m}} \mathop{\mathbb{E}}_{j\sim J} \max_{h\in\mathcal{H}_A} \mathbb{1}_{[|h\cap A_j|=0]} \mathbb{1}_{[|h\cap A|\geqslant\alpha]} \\ &= \mathop{\mathbb{E}}_{A\sim\mathcal{D}^{2m}} \max_{h\in\mathcal{H}_A} \mathbb{1}_{[|h\cap A|\geqslant\alpha]} \mathop{\mathbb{E}}_{j\sim J} \mathbb{1}_{[|h\cap A_j|=0]}\end{aligned}$$

现在，固定算法 A 使得 $|h\cap A|\geqslant\alpha$。那么，$\mathbb{E}_j\,\mathbb{1}_{[|h\cap A_j|=0]}$ 是从一个至少有 α 个红球的包里拿出的 m 个球中无一为红球的概率。这个概率至多为

$$(1-\alpha/(2m))^m=(1-\varepsilon/4)^m\leqslant e^{-\varepsilon m/4}$$

因此可以得到

$$\mathbb{P}[A\in B']\leqslant \mathop{\mathbb{E}}_{A\sim\mathcal{D}^{2m}} \sum_{h\in\mathcal{H}_A} e^{-\varepsilon m/4} \leqslant e^{-\varepsilon m/4} \mathop{\mathbb{E}}_{A\sim\mathcal{D}^{2m}}[\mathcal{H}_A]$$

采用生长函数的定义，我们完成论断 2 的证明。

完成证明　基于 Sauer 的引理我们知道 $\tau_{\mathcal{H}}(2m)\leqslant(2em/d)^d$。将该式与前述两论断结合，可以得到

$$\mathbb{P}[S\in B]\leqslant 2(2em/d)^d e^{-\varepsilon m/4}$$

我们希望不等式右侧至多为 δ，即

$$2(2em/d)^d e^{-\varepsilon m/4}\leqslant\delta$$

重新整理一下，可以得到对 m 的要求是

$$m\geqslant\frac{4}{\varepsilon}(d\log(2em/d)+\log(2/\delta))=\frac{4d}{\varepsilon}\log(m)+\frac{4}{\varepsilon}(d\log(2e/d)+\log(2/\delta))$$

利用引理 A.2，使得前述式子成立的充分条件是

$$m\geqslant\frac{16d}{\varepsilon}\log\left(\frac{8d}{\varepsilon}\right)+\frac{8}{\varepsilon}(d\log(2e/d)+\log(2/\delta))$$

进一步地，上式的充分条件是

$$\begin{aligned}m &\geqslant\frac{16d}{\varepsilon}\log\left(\frac{8d}{\varepsilon}\right)+\frac{16}{\varepsilon}\left(d\log(2e/d)+\frac{1}{2}\log(2/\delta)\right) \\ &=\frac{16d}{\varepsilon}\left(\log\left(\frac{8d2e}{d\varepsilon}\right)\right)+\frac{8}{\varepsilon}\log(2/\delta) \\ &=\frac{8}{\varepsilon}\left(2d\log\left(\frac{16e}{\varepsilon}\right)+\log\left(\frac{2}{\delta}\right)\right)\end{aligned}$$

证明完成。 ■

从 ε-网到 PAC 可学习

定理 28.4　*令 $\mathcal{H}$ 是在 $\mathcal{X}$ 上的假设类，且 $\mathrm{VCdim}(\mathcal{H})=d$。令 $\mathcal{D}$ 是 $\mathcal{X}$ 上的分布，$c\in\mathcal{H}$ 是目标假设。固定 $\varepsilon,\delta\in(0,1)$，且令 m 如定理 28.3 中定义的那样。那么，在从 $\mathcal{X}$ 中选出的 m 个独立同分布的样例，且各自符号根据 c 来决定，将至少以 $1-\delta$ 的概率，任何 ERM 假*

349

设的真实误差至多为 ϵ。

证明 定义类 $\mathcal{H}'=\{c\triangle h: h\in\mathcal{H}\}$，其中 $c\triangle h=(h\setminus c)\cup(c\setminus h)$。容易验证如果某个 $A\subset\mathcal{X}$ 能被 $\mathcal{H}$ 打散，那么它也能被 $\mathcal{H}'$ 打散，反之亦然。因此，$\mathrm{VCdim}(\mathcal{H})=\mathrm{VCdim}(\mathcal{H}')$。因此，根据定理 28.3，可以知道以至少 $1-\delta$ 的概率，样本 S 是 $\mathcal{H}'$ 的 ϵ-网。注意到 $L_{\mathcal{D}}(h)=\mathcal{D}(h\triangle c)$。因此，对任意满足 $L_{\mathcal{D}}(h)\geq\epsilon$ 的 $h\in\mathcal{H}$，有 $|(h\triangle c)\cap S|>0$。这暗示了 h 不可能是 ERM 准则。定理得证。 ■

350

多分类可学习性

在第 17 章中我们介绍了多分类问题，目标是学习一个预测器 $h:\mathcal{X}\to[k]$。在本章中我们提出基于 0—1 损失函数下 PAC 算法的多分类预测器的可学习性。正如第 6 章中讲述的，这一章的主要目标是：

- 描述多分类问题的假设类的特性在(多分类)PAC 模型下是可学习的。
- 量化此类假设类的样本复杂度。

基于学习理论(定理 6.8)中的基本定理，我们很自然地想探索多分类假设类中 VC 维的泛化。在 29.1 节中我们将要展示这样一个泛化，称为纳塔拉詹维(Natarajan dimension)，并且陈述基于纳塔拉詹维的基本定理的一个泛化。然后我们论证如何计算若干重要的假设类的纳塔拉詹维。

回想学习理论中基本定理的主要信息，就是说一个二分类假设类是可学习的(基于 0—1 损失)，当且仅当它有一致收敛特性时，假设类在任意的 ERM(经验风险最小化)准则下是可学习的。在第 13 章中的练习 29.2，我们展示了这个类学习问题可以等值分解成一个确定的凸集学习问题。本章的最后一节致力于说明介于学习性和一致收敛性上的等值分解，甚至在多分类问题亦如此。实际上，我们构造了这样一个假设类，它可以被一个特定的 ERM 准则学习，但是对于其他的 ERM 评价准则也许就会失效，且不保持一致的收敛特性。

29.1 纳塔拉詹维

在这一章中我们定义纳塔拉詹维，它是多分类预测器所属类的 VC 维的一个泛化的概念。贯穿本节中，$\mathcal{H}$ 是一个多分类预测器的假设类；每一个 $h\in\mathcal{H}$ 是一个从定义域(样本集)$\mathcal{X}$ 到值域(类)$[k]$ 的一个函数。 [351]

要想定义纳塔拉詹维，我们首先泛化“打散”的定义。

定义 29.1(打散(多分类角度)) 我们说一个集合 $C\subset\mathcal{X}$ 被 $\mathcal{H}$ 打散，如果存在两个函数 $f_0, f_1:C\to[k]$ 使得

- 对每一个 $x\in C$，$f_0(x)\neq f_1(x)$。
- 对每一个 $B\subset C$，存在一个函数 $h\in\mathcal{H}$，使得 $\forall x\in B$，$h(x)=f_0(x)$ 并且 $\forall x\in C\setminus B$，$h(x)=f_1(x)$。

定义 29.2(纳塔拉詹维) 假设类 $\mathcal{H}$ 的纳塔拉詹维表示为 $\mathrm{Ndim}(\mathcal{H})$，是被打散的集合 $C\subset\mathcal{X}$ 的最大尺寸。

在这种情况下很容易看出，确切地说对于二分类，$\mathrm{Ndim}(\mathcal{H})=\mathrm{VCdim}(\mathcal{H})$。因此，纳塔拉詹维(Na 维)泛化了 VC 维。接下来我们展示纳塔拉詹维(Na 维)允许我们泛化针对从二分类到多分类的统计学习的基础理论。

29.2 多分类基本定理

定理 29.3(多分类基本定理) 存在绝对常量 $C_1, C_2>0$ 使得以下成立。对于从 $\mathcal{X}$ 到

$[k]$的函数的任意假设类$\mathcal{H}$，$\mathcal{H}$的 Na 维是 d。我们有

1. $\mathcal{H}$有一致收敛性，样本复杂度为

$$C_1 \frac{d+\log\left(\frac{1}{\delta}\right)}{\varepsilon^2} \leq m_{\mathcal{H}}^{UC}(\varepsilon,\delta) \leq C_2 \frac{d\log(k)+\log\left(\frac{1}{\delta}\right)}{\varepsilon^2}$$

2. $\mathcal{H}$是不可知 PAC 可学习的，样本复杂度为

$$C_1 \frac{d+\log\left(\frac{1}{\delta}\right)}{\varepsilon^2} \leq m_{\mathcal{H}}(\varepsilon,\delta) \leq C_2 \frac{d\log(k)+\log\left(\frac{1}{\delta}\right)}{\varepsilon^2}$$

3. $\mathcal{H}$是 PAC 可学习的(假设可实现性)，样本复杂度为

$$C_1 \frac{d+\log\left(\frac{1}{\delta}\right)}{\varepsilon} \leq m_{\mathcal{H}}(\varepsilon,\delta) \leq C_2 \frac{d\log\left(\frac{kd}{\varepsilon}\right)+\log\left(\frac{1}{\delta}\right)}{\varepsilon}$$

定理 29.3 的证明

在定理 29.3 中，下界可以被二分类基础理论中的约简推断出来(见练习 29.5)。

在 28 章中给出了二分类基础理论的证明过程，沿着这个证明主线，定理 29.3 中的上界可以被确定下来(见练习 29.4)。证明中唯一需要以复杂方式修改的要素是 Sauer 引理。它只应用在二分类问题，因此必须被取代。能够合适地替代它的就是纳塔拉詹引理：

引理 29.4(纳塔拉詹) $|\mathcal{H}| \leq |\mathcal{X}|^{\mathrm{Ndim}(\mathcal{H})} \cdot k^{2\mathrm{Ndim}(\mathcal{H})}$

352

纳塔拉詹定理的证明采用了 Sauer 引理的证明精髓，留作练习(见练习 29.3)。

29.3 计算纳塔拉詹维

在这一章中我们讨论如何计算(或估计)几个著名的类的纳塔拉詹维 $\mathrm{Ndim}(\mathcal{H})$，其中的几个我们已经在 17 章中讲过。正如这些计算表明，纳塔拉詹维总是和所要求定义的假设中的参数的个数成比例的。

29.3.1 基于类的一对多

在第 17 章中我们已经目睹了多分类问题到二分类问题的两种约简方法：一对多和一对一。在这一小节中我们计算一对多方法的纳塔拉詹维。

回想我们训练过的一对多方法：对于每一个标签，二分类器区别开正确标签和其他的标签。在接下来的形式中，很自然地建议考虑多分类假设类。令$\mathcal{H}_{\mathrm{bin}} \subseteq \{0,1\}^{\mathcal{X}}$是一个二分类假设类。每一个$\bar{h}=(h_1,\cdots,h_k)\in(\mathcal{H}_{\mathrm{bin}})^k$，定义$T(\bar{h}):\mathcal{X}\to[k]$为

$$T(\bar{h})(x)=\underset{i\in[k]}{\arg\max}\, h_i(x)$$

如果存在两个标签最大化$h_i(x)$，选择较小的那个标签。另外，令$\mathcal{H}_{\mathrm{bin}}^{\mathrm{OVA},k}=\{T(\bar{h}):\bar{h}\in(\mathcal{H}_{\mathrm{bin}})^k\}$。什么应该是$\mathcal{H}_{\mathrm{bin}}^{\mathrm{OVA},k}$的 Na 维？直观地讲，要想详细列举$\mathcal{H}_{\mathrm{bin}}$的假设类，我们需要$d=\mathrm{VCdim}(\mathcal{H}_{\mathrm{bin}})$的参数。要想详细列举$\mathcal{H}_{\mathrm{bin}}^{\mathrm{OVA},k}$的假设类，我们需要$k$个$\mathcal{H}_{\mathrm{bin}}$中的假设类。因此，$kd$个参数就可以满足要求。接下来的引理确定了这一直觉上的猜想。

引理 29.5 如果$d=\mathrm{VCdim}(\mathcal{H}_{\mathrm{bin}})$，则

$$\mathrm{Ndim}(\mathcal{H}_{\mathrm{bin}}^{\mathrm{OVA},k}) \leq 3kd\log(kd)$$

证明　令 $C\subset\mathcal{X}$ 是一个被打散的集合。由打散的定义(对于多分类假设)可得，

$$|(\mathcal{H}_{\text{bin}}^{\text{OvA},k})_C|\geq 2^{|C|}$$

另一方面，$\mathcal{H}_{\text{bin}}$ 中的 k 类假设类决定了 $\mathcal{H}_{\text{bin}}^{\text{OvA},k}$ 中每一个假设类。因此，

$$|(\mathcal{H}_{\text{bin}}^{\text{OvA},k})_C|\leq|(\mathcal{H}_{\text{bin}})_C|^k$$

由 Sauer 引理知，

$$|(\mathcal{H}_{\text{bin}})_C|\leq|C|^d$$

推导出

$$2^{|C|}\leq|(\mathcal{H}_{\text{bin}}^{\text{OvA},k})_C|\leq|C|^{dk}$$

接下来证明采用对数并且应用了引理 A.1。 ■ 353

引理 29.5 是否严格成立？不难得出，对于某些假设类，$\text{Ndim}(\mathcal{H}_{\text{bin}}^{\text{OvA},k})$ 比 dk 更小(见练习 29.1)。然而，存在若干天然的二分类问题，$\mathcal{H}_{\text{bin}}$(例如半空间)的 $\text{Ndim}(\mathcal{H}_{\text{bin}}^{\text{OvA},k})=\Omega(dk)$(见练习 29.6)。

29.3.2　一般的多分类到二分类约简

对于更多一般意义上的多分类和二分类之间的转化中的约简问题，可以用支撑引理 29.5 成立的理由来约束其 Na 维。这些约简方法基于数据训练了若干个二分类器。然后给出一个新实例，它们将一些二分类器预测出的标签考虑进去，通过这样的规则预测出了新标签。这些约简包括一对多和一对一。

假设有这样一个方法，它可以从一个二分类假设类 $\mathcal{H}_{\text{bin}}$ 训练 l 个二分类器，并且存在一个规则 $r:\{0,1\}^l\to[k]$，根据二分类器的预测，这条规则决定了(多类)标签。按照这个方法假设类可以定义如下：对于每一个 $\bar{h}=(h_1,\cdots,h_l)\in(\mathcal{H}_{\text{bin}})^l$，定义 $R(\bar{h}):\mathcal{X}\to[k]$ 为

$$R(\bar{h})(x)=r(h_1(x),\cdots,h_l(x))$$

最后，令

$$\mathcal{H}_{\text{co}}=\{R(\bar{h}):\bar{h}\in(\mathcal{H}_{\text{bin}})^l\}$$

与引理 29.5 相似，可以证明如下引理 29.6：

引理 29.6　如果 $d=\text{VCdim}(\mathcal{H}_{\text{bin}})$，则

$$\text{Ndim}(\mathcal{H}_{\text{co}})\leq 3ld\log(ld)。$$

证明留给读者，见练习 29.2。

29.3.3　线性多分类预测器

接下来，我们考虑线性多分类预测器的假设类(见 17.2 节)。令 $\Psi:\mathcal{H}\times[k]\to\mathbb{R}^d$ 是一些敏感类的特征映射，并且

$$\mathcal{H}_\Psi=\{x\mapsto\operatorname*{argmax}_{i\in[k]}\langle w,\Psi(x,i)\rangle:w\in\mathbb{R}^d\}\tag{29.1}$$

在 $\mathcal{H}_\Psi$ 中的每一个假设取决于 d 个参数，即，向量 $w\in\mathbb{R}^d$。因此，我们希望纳塔拉詹维有上界 d。事实上有：

定理 29.7　$\text{Ndim}(\mathcal{H}_\Psi)\leq d$。

证明　打散集合 $C\subset\mathcal{X}$，令 $f_0,f_1:C\to[k]$ 是证明打散集合的两个函数。我们需要证明 $|C|\leq d$。对每一个 $x\in C$，令 $\rho(x)=\Psi(x,f_0(x))-\Psi(x,f_1(x))$。我们声明，集合 $\rho(C)=\{\rho(x):x\in C\}$ 在 $\mathbb{R}^d$(d 维欧式空间)上由 $|C|$ 个元素(例如 ρ 是一对一的映射)构成并

且被齐次线性分离器的二分类假设类打散，数学表示为

354

$$\mathcal{H}=\{x\mapsto \operatorname{sign}(\langle w,x\rangle): w\in \mathbb{R}^d\}$$

因为 $\mathrm{VCdim}(\mathcal{H})=d$，得出 $|C|=|\rho(C)|\leq d$，正如所要求证明的。

为达到我们的要求，需要证明 $|\mathcal{H}_{\rho(C)}|=2^{|C|}$。事实上，给出一个 C 的子集 B，基于打散的定义，存在 $h_B\in\mathcal{H}_\varphi$ 使得

$$\forall x\in B, h_B(x)=f_0(x),\quad \forall x\in C\setminus B, h_B(x)=f_1(x)$$

令 $w_B\in\mathbb{R}^d$ 是一个定义 h_B 的向量。我们有，对于任意一个 $x\in B$，

$$\langle w,\Psi(x,f_0(x))\rangle>\langle w,\Psi(x,f_1(x))\rangle\Rightarrow\langle w,\rho(x)\rangle>0$$

相似地，对于任意一个 $x\in C\setminus B$，

$$\langle w,\rho(x)\rangle<0$$

由此可得同样被 $w\in\mathbb{R}^d$ 定义的假设类 $h_B\in\mathcal{H}$ 把向量 $\rho(B)$ 中的样本点标记为 1，向量 $\rho(C\setminus B)$ 中的样本点标记为 0。因为这样的标记适用于每一个 $B\subset C$，得出 $|C|=|\rho(C)|$，$|\mathcal{H}_{\rho(C)}|=2^{|C|}$，证明了我们的结论。 ■

定理严格成立的条件是，存在一个映射 Ψ 使得 $\mathrm{Ndim}(\mathcal{H}_\varphi)=\Omega(d)$。例如，该定理满足多向量(多重向量)的构造(见 17.2 章和本章末尾的文献评注)。我们因此得出结论：

推论 29.8 令 $\mathcal{X}=\mathbb{R}^n$ 并且 $\Psi:\mathcal{H}\times[k]\to\mathbb{R}^n$ 是欧感特征的类构造多向量的映射：

$$\Psi(x,y)=[\underbrace{0\cdots0}_{\in\mathbb{R}^{(y-1)n}},\underbrace{x_1\cdots x_n}_{\in\mathbb{R}^n},\underbrace{0\cdots0}_{\in\mathbb{R}^{(k-y)n}}]$$

令 $\mathcal{H}_\varphi$ 如等式(29.1)中定义。则 $\mathcal{H}_\varphi$ 的纳塔拉詹维满足

$$(k-1)(n-1)\leq \mathrm{Ndim}(\mathcal{H}_\varphi)\leq kn$$

29.4 好的与坏的 ERM

本节我们提出一个假设类的例子，该假设类具有的特性不是所有的 ERM 评判准则都能适合的。此外，如果允许无限多的标签，我们仍然可以获得一个被 ERM 准则学习出的一个假设类，但是其他 ERM 准则也许会学习失败。很明显地，这说明了假设类是可以学习的，但是它们没有一致的收敛性。为简单起见，我们只考虑可实现的情况。我们考虑的假设类定义如下。实例空间 $\mathcal{X}$ 是任何有限集或可数集。令 $P_f(\mathcal{X})$ 是 $\mathcal{X}$ 的所有子集的集合，包括有限子集和余有限子集(也就是说，对于每一个 $A\in P_f(\mathcal{X})$，无论是 A 还是 $\mathcal{X}\setminus A$ 一定是有限的集合)。标签的集合是 $\mathcal{Y}=P_f(\mathcal{X})\cup\{*\}$ 而不是 $[k]$，其中 $*$ 是一些特殊的标签。对每一个 $A\in P_f(\mathcal{X})$，定义 $h_A:\mathcal{X}\to\mathcal{Y}$ 如下：

$$h_A(x)=\begin{cases}A & x\in A\\ * & x\notin A\end{cases}$$

最后，我们采用的假设类是

355

$$\mathcal{H}=\{h_A: A\subset P_f(\mathcal{X})\}$$

令 A 是 $\mathcal{H}$ 的 ERM 算法。假设 A 在一个被 $h_A\in\mathcal{H}$ 所标签的样本集上操作。因为 h_A 是 $\mathcal{H}$ 中唯一可能返回标签 A 的假设类，如果 A 观察标签 A，它"知道"已学习的假设类是 h_A，并且，作为 ERM 准则一定要返回(注意在这种情况下假设类返回误差是 0)。因此，指定一个 ERM，我们应该只指定返回的假设类接收一个如下形式的样本：

$$S=((x_1,*),\cdots,(x_m,*))$$

我们考虑两个 ERM 准则：第一个，A_{good} 定义如下：

$$A_{\mathrm{good}}(S)=h_\emptyset$$

也就是说，它输出一些假设类，这些假设类对于每一个 $x\in\mathcal{X}$ 预测出 "*"。第二个 ERM 是 A_{bad}，定义如下：

$$A_{\text{bad}}(S)=h_{\langle x_1,\dots,x_m\rangle}$$

接下来的声明展示了 A_{bad} 的样本复杂度大约是 A_{good} 的样本复杂度的 $|\mathcal{X}|$ 倍数。这形成了不同的 ERM 准则之间的裂口。如果 $\mathcal{X}$ 是有限集，我们甚至可以获得一个可学习的类，然而并非是对于任何的 ERM 准则都可学习的。

论断 29.9

1. 令 $\varepsilon,\delta>0$，$\mathcal{D}$ 是样本集 $\mathcal{X}$ 的分布，且 $h_A\in\mathcal{H}$。令 S 是由 $m\geq\frac{1}{\varepsilon}\log\left(\frac{1}{\delta}\right)$ 个独立同分布样本构成的，根据 $\mathcal{D}$ 采样并且标签是 h_A。然后，在 A_{good} 状态下，假设类以至少 $1-\delta$ 的概率返回，错误率至多是 ε。

2. 存在一个常量 $a>0$，使得对于每一个 $0<\varepsilon<a$，存在一个 $\mathcal{X}$ 上的分布 $\mathcal{D}$ 并且 $h_A\in\mathcal{H}$，有下述成立。在 A_{bad} 状态下，接收样本集大小为 $m\leq\frac{|X|-1}{6\varepsilon}$，根据 $\mathcal{D}$ 采样并且标签是 h_A，返回的假设类将会以 $e^{-\frac{1}{6}}$ 的概率出现一个不小于 ε 的错误率。

证明　$\mathcal{D}$ 是 $\mathcal{X}$ 的分布，假设正确的标签是 h_A。对于任意的样本，A_{bad} 返回 $h_{\mathcal{X}}$ 或者 h_A。如果返回 h_A，那么函数的真实误差是 0。因此，返回一个错误率不小于 ε 的假设类当且仅当样本中全部 m 个例子来源于 $\mathcal{X}\setminus A$，同时 $L_{\mathcal{D}}(h_{\emptyset})=\mathbb{P}_{\mathcal{D}}[A]$ 中 $h_{\emptyset}$ 的错误率不小于 ε。假定 $m\geq\frac{1}{\varepsilon}\log\left(\frac{1}{\delta}\right)$；后者的概率仅仅是 $(1-\varepsilon)^m\leq e^{-m\varepsilon}\leq\delta$。这证明了第 1 条。

接下来证明第 2 条，我们限定 $|\mathcal{X}|=d<\infty$ 成立。对于无限的 $\mathcal{X}$ 的证明是相似的，假设 $\mathcal{X}=\{x_1,\dots,x_{d-1}\}$。

令 $a>0$ 足够小使之对于任意的 $\varepsilon<a$ 有 $1-2\varepsilon\geq e^{-\frac{1}{6}}$。定义一个 $\mathcal{X}$ 上的分布，令 $\mathbb{P}[x_1]=1-2\varepsilon$。对于所有的 $1\leq i\leq d-1$，$\mathbb{P}[x_i]=\frac{2\varepsilon}{d-1}$。假设正确的假设类是 $h_{\emptyset}$，样本集大小是 m。明显地，被 A_{bad} 返回的假设类将会误导一些不在样本集 $\mathcal{X}$ 中的样本。根据切尔诺夫界，如果 $m\leq\frac{d-1}{6\varepsilon}$，那么样本集以大于 $e^{-\frac{1}{6}}$ 的概率仅仅包括 $\mathcal{X}$ 中的 $\frac{d-1}{2}$ 个样本。因此返回的假设类会有不小于 ε 的错误率。 [356]

从例子中得出结论：在多分类中，样本复杂度随着不同的 ERM 评判准则可能会不同。对于任意的假设类是否存在"好"的 ERM 评判准则？接下来的猜想给出了问题的答案是：存在。

猜想 29.10　每个假设类 $\mathcal{H}\subset[k]^{\mathcal{X}}$ 的可实现的样本复杂度是

$$m_{\mathcal{H}}(\varepsilon,\delta)=\tilde{O}\left(\frac{\text{Ndim}(\mathcal{H})}{\varepsilon}\right)$$

我们强调符号 $\tilde{O}$ 可能只隐藏了 ε、δ 和 $\text{Ndim}(\mathcal{H})$ 的广义对数因子，没有隐藏 k 的因子。

29.5　文献评注

纳塔拉詹维来自于纳塔拉詹 1989 年发表的一篇文章，并且那篇文章确立了纳塔拉詹引理和泛化的基本定理。关于纳塔拉詹引理的泛化和更明晰的版本是 Haussler 和 Long 于 1995 年研究提出的。Ben-David，Cesa-Bianchi，Haussler 和 Long 在 1995 年定义了关于

维度概念的大家族，这个维度定义泛化了 VC 维并且可能被用来估计多分类问题的样本复杂度。

纳塔拉詹维和其他假设类的计算可以在 Daniely 等(2012)中找到，基于好的和坏的类上的 ERM，以及猜想 29.10，都是来源于 Daniely 等(2012)。

29.6 练习

29.1 令 $d, k>0$。证明：存在 VC 维是 d 的二类假设函数集 $\mathcal{H}_{\text{bin}}$，使得 $\text{Ndim}(\mathcal{H}_{\text{bin}}^{m,k})=d$。

29.2 证明引理 29.6。

29.3 证明纳塔拉詹引理。

提示：固定样本点 $x_0\in\mathcal{X}$。对于 $i, j\in[k]$，所有的函数 $f:\mathcal{X}\setminus\{x_0\}\to[k]$ 记作 $\mathcal{H}_{ij}$，并且可以通过定义 $f(x_0)=i$ 和 $f(x_0)=j$ 扩展到 $\mathcal{H}$ 中的一个函数。用归纳法证明 $|\mathcal{H}|\le|\mathcal{H}_{\mathcal{X}\setminus\{x_0\}}|+\sum_{i<j}|\mathcal{H}_{ij}|$。

29.4 根据二分类基础定理和纳塔拉詹引理的证明，证明对于全局常量 $C>0$，并且对于每一个纳塔拉詹维是 d 的假设类，$\mathcal{H}$ 的不可知的样本复杂度是 $m_{\mathcal{H}}(\varepsilon,\delta)\le C\frac{d\log(\frac{kd}{\varepsilon})+\log(1/\delta)}{\varepsilon^2}$。

29.5 证明：对于全局常量 $C>0$，并且对于每一个纳塔拉詹维是 d 的假设类，$\mathcal{H}$ 的不可知的样本复杂度是 $m_{\mathcal{H}}(\varepsilon,\delta)\ge C\frac{d+\log(1/\delta)}{\varepsilon^2}$。

提示：从二分类基础定理推断。

29.6 令 $\mathcal{H}$ 是 $\mathbb{R}^d$ 中(非齐次的)半空间的二分类假设类，此练习的目标是证明

$$\text{Ndim}(\mathcal{H}^{m,k})\ge(d-1)\cdot(k-1)$$

1) 令 $\mathcal{H}_{\text{dim}}$ 是满足 $f:[k-1]\times[d-1]\to\{0,1\}$ 的所有函数的类。存在 i_0 对于任意的 $j\in[d-1]$，有以下结论成立：当 $\forall i>i_0$，$f(i,j)=0$，有 $\forall i<i_0$，$f(i,j)=1$ 成立。

证明 $\text{Ndim}(\mathcal{H}_{\text{dim}}^{m,k})=(d-1)\cdot(k-1)$。

2) 证明：$\mathcal{H}_{\text{dim}}$ 可以被 $\mathcal{H}$ 解释。也就是说，证明存在一个映射 $\psi:[k-1]\times[d-1]\to\mathbb{R}^d$，使得 $\mathcal{H}_{\text{dim}}\subseteq\{h\circ\psi:h\in\mathcal{H}\}$。

提示：可以把 $\psi(i,j)$ 看作是一个向量，该向量的第 j 个元素是 1，最后一个元素是 i，其余元素全部是 0。

3) 推导：$\text{Ndim}(\mathcal{H}_{\text{dim}}^{m,k})\ge(d-1)\cdot(k-1)$。

357 ~ 358

压 缩 界

本书中，我们已尝试采用不同的方式来描述“可学习性”这个概念。从最开始的假设类一致收敛性保证成功学习，到后来引入稳定性来反映稳定的算法可以保证得到好的学习器。然而，对于学习而言，还有一些其他的充分条件，本章以及下一章将介绍两种新的条件：压缩界和 PAC-贝叶斯法。

本章将重点介绍压缩界。简单地说，如果一个学习算法可以用训练集的一个小子集来表达输出假设，那将可用此假设在其余样本上的误差来估计全体样本的真实误差。换言之，一个可以“压缩”其输出的算法就是一个好的学习器。

30.1 压缩界概述

为了引出压缩界，首先考虑如下学习机制。我们先采样得到包含 k 个样本的序列，记作 T。基于这些样本，构建一个假设 h_T。现在我们想要估计 h_T 的效果，因此我们采样得到另一个包含 $m-k$ 个样本的序列，记作 V，而后计算 h_T 在 V 上的误差。由于 V 和 T 是独立的，根据 Bernstein 不等式可得如下引理。

引理 30.1 假设损失函数在 $[0,1]$ 取值，那么，

$$\mathbb{P}\left[L_{\mathcal{D}}(h_T)-L_V(h_T)\geq\sqrt{\frac{2L_V(h_T)\log(1/\delta)}{|V|}}+\frac{4\log(1/\delta)}{|V|}\right]\leq\delta$$

为了得到这个界，我们仅需保证 V 和 T 的独立性。因此，我们可以重新定义之前的学习机制如下。首先，在长度为 k 的序列 $I=(i_1,\dots,i_k)\in[m]^k$ 上学习，之后，采样得到 m 个样本的序列 $S=(z_1,\dots,z_m)$，定义 $T=S_I=(z_{i_1},\dots,z_{i_k})$，$V$ 是 S 中剩余的样本。由于这个机制与之前的机制是等价的，因此引理 30.1 依然成立。

359

利用对于任意长度为 k 的序列的联合界，我们得到如下定理：

定理 30.2 k 为整数，$B:Z^k\to\mathcal{H}$ 是长度为 k 的样本序列到假设类的映射，训练集规模 $m\geq 2k$，$A:Z^m\to\mathcal{H}$ 是由大小为 m 的训练样本序列导得假设的学习规则，使得对于某些 $(i_1,\dots,i_k)\in[m]^k$，$A(S)=B(z_{i_1},\dots,z_{i_k})$。令 $V=\{z_j:j\notin(i_1,\dots,i_k)\}$ 是没有被来定义 $A(S)$ 的样本集。那么，对于任意样本集 S，以至少 $1-\delta$ 的概率下式成立：

$$L_{\mathcal{D}}(A(S))\leq L_V(A(S))+\sqrt{L_V(A(S))\frac{4k\log(m/\delta)}{m}}+\frac{8k\log(m/\delta)}{m}$$

证明 对于任意 $I\in[m]^k$，令 $h_I=B(z_{i_1},\dots,z_{i_k})$，$n=m-k$。由引理 30.1 和联合界可得

$$\mathbb{P}\left[\exists I\in[m]^k\text{ s.t. }L_{\mathcal{D}}(h_I)-L_V(h_I)\geq\sqrt{\frac{2L_V(h_I)\log(1/\delta)}{n}}+\frac{4\log(1/\delta)}{n}\right]$$

$$\leq\sum_{I\in[m]^k}\mathbb{P}\left[L_{\mathcal{D}}(h_I)-L_V(h_I)\geq\sqrt{\frac{2L_V(h_I)\log(1/\delta)}{n}}+\frac{4\log(1/\delta)}{n}\right]\leq m^k\delta$$

记 $\delta'=m^k\delta$，假定 $k\leq m/2$，即 $n=m-k\geq m/2$，由上式可得以至少 $1-\delta'$ 的概率有下

面的式子成立：

$$L_{\mathcal{D}}(A(S)) \leq L_V(A(S)) + \sqrt{L_V(A(S))\frac{4k\log(m/\delta')}{m}} + \frac{8k\log(m/\delta')}{m}$$

证毕。 ■

有上述定理可直接得到下面的推论。

推论 30.3 假设定理 30.2 的条件成立，并假定 $L_V(A(S))=0$。那么，对于任意 S，以至少 $1-\delta$ 的概率下式成立：

$$L_{\mathcal{D}}(A(S)) \leq \frac{8k\log(m/\delta)}{m}$$

由上述结论可以引出如下定义：

定义 30.4（压缩机制） 令$\mathcal{H}$是一个从$\mathcal{X}$到$\mathcal{Y}$的假设函数类，k 为整数。如果下面的条件成立，我们就说$\mathcal{H}$有一个大小为 k 的压缩机制：

对于所有 m，存在 $A:Z^m \to [m]^k$ 以及 $B:Z^k \to \mathcal{H}$使得对于所有 $h \in \mathcal{H}$，如果我们将任意形如$(x_1, h(x_1)), \dots, (x_m, h(x_m))$的训练集输入到 A，形如$(x_{i_1}, h(x_{i_1})), \dots, (x_{i_k}, h(x_{i_k}))$的训练集输入到 B，其中$(i_1, \dots, i_k)$是 A 的输出，用 h'表示 B 的输出，那么满足$L_S(h')=0$。

360

对于不可实现序列的情况，可以容易地进行如下推广。

定义 30.5（不可实现序列的压缩机制） 令$\mathcal{H}$是一个从$\mathcal{X}$到$\mathcal{Y}$的假设函数类，k 为整数。如果下面的条件成立，我们就说$\mathcal{H}$有一个大小为 k 的压缩机制：

对于所有 m，存在 $A:Z^m \to [m]^k$ 以及 $B:Z^k \to \mathcal{H}$使得对于所有 $h \in \mathcal{H}$，如果我们将任意形如$(x_1, y_1), \dots, (x_m, y_m)$的训练集输入到 A，形如$(x_{i_1}, y_{i_1}), \dots, (x_{i_k}, y_{i_k})$的训练集输入到 B，其中$(i_1, \dots, i_k)$是 A 的输出，用 h'表示 B 的输出，那么满足 $L_S(h') \leq L_S(h)$。

下面的引理表面可实现情况压缩机制的存在意味着不可实现情况压缩机制的存在。

引理 30.6 令$\mathcal{H}$是一个二分类问题的假设类，并假定在可实现情况下$\mathcal{H}$有一个大小为 k 的压缩机制，则$\mathcal{H}$在不可实现的情况下同样有一个大小为 k 的压缩机制。

证明 考虑如下机制：首先，找到一个满足 ERM 的假设并记为 h。然后，丢掉所有 h 错分的样本，之后在未被丢弃的样本上应用可实现情况的压缩机制，将该机制的输出记为 h'，则必定在未被丢弃的样本上分类正确。由于 h 在丢弃的样本上均分类错误，因此 h'在那些被丢弃样本上的错误不会比 h 更多。故而 h'也是一个满足 ERM 的假设。 ■

30.2 例子

在如下例子中，我们将介绍一些对于二分类问题的假设类的压缩机制。引理 30.6 已表明我们此时只需关注可实现情况即可。因此，为了表明一个确定的假设类有压缩机制，需要找到相应的 A，B 和 k 使得 $L_S(h')=0$。

30.2.1 平行于轴的矩形

注意到这个假设类是不可数无穷多的。不过该假设类有一个简单的压缩机制。考虑一个算法 A：对于每一维，选择在这一维度有极值的两个正样本。定义 B 是一个函数：（根据找

到的样本)返回一个最小包络矩形。那么对于 $k=2d$，在可实现情况下，$L_S(B(A(S)))=0$。

30.2.2 半空间

令 $\mathcal{X}=\mathbb{R}^d$，并只考虑齐次半空间，即 $\{x\mapsto \mathrm{sign}(\langle w, x\rangle): w\in\mathbb{R}^d\}$。

压缩机制：

不失一般性，假定所有标签都是正的(否则，用 $y_i x_i$ 替换 x_i)。首先，算法 A 在 $\{x_1, \cdots, x_n\}$ 的凸包中找到一个范数最小的向量 w，此向量可由样本中的 d 个点的凸组合表出(之后会看到这种表出总是可行)。A 的输出就是这 d 个点。而后算法 B 根据这 d 个点得到 w，并将 w 作为样本凸包中有着最小范数的点。 [361]

接下来我们证明这的确是一个压缩机制。由于样本是线性可分的，$\{x_1, \cdots, x_n\}$ 将不会包括原点。现在考虑凸包中距离原点最近的点 w。(这个点是唯一的，因为该点是原点在凸包上的欧氏投影。)我们认为 w 分开了原数据[⊖]。为了说明这一点，采用反证法，假定对于某些 i，有 $\langle w, x_i\rangle\leq 0$。对于 $\alpha=\frac{\|w\|^2}{\|x_i\|^2+\|w\|^2}\in(0, 1)$，取 $w'=(1-\alpha)w+\alpha x_i$，则 w' 也在凸包中且

$$
\begin{aligned}
\|w'\|^2 &= (1-\alpha)^2\|w\|^2+\alpha^2\|x_i\|^2+2\alpha(1-\alpha)\langle w, x_i\rangle \\
&\leq (1-\alpha)^2\|w\|^2+\alpha^2\|x_i\|^2 \\
&= \frac{\|x_i\|^4\|w\|^2+\|x_i\|^2\|w\|^4}{(\|w\|^2+\|x_i\|^2)^2} \\
&= \frac{\|x_i\|^2\|w\|^2}{\|w\|^2+\|x_i\|^2} \\
&= \|w\|^2\cdot\frac{1}{\|w\|^2/\|x_i\|^2+1} \\
&< \|w\|^2
\end{aligned}
$$

这样就产生了矛盾。

因此我们可以说明这样的 w 也是满足 ERM 的。最后，由于 w 在样本凸包中，根据 Caratheodory 定理可得 w 也在该多边形(译者注：指样本凸包)中由 $d+1$ 个点构成子集的凸包中。更进一步，w 的最小性要求 w 必须在多边形的表面，意味着它可以由 d 个点的凸组合表出。

接下来还需要说明 w 也是原点在由 d 个点定义的多边形上的投影，这是必然的：一方面，小的多边形是大的多边形的子集，因此原点在小的多边形上的投影在范数上不会变更小；另一方面，w 本身是一个有效解。由于投影是唯一的，命题得证。

30.2.3 可分多项式

令 $\mathcal{X}=\mathbb{R}^d$ 并考虑类：$x\mapsto \mathrm{sign}(p(x))$，其中 p 是 r 阶多项式。

注意到 $p(x)$ 可以写作 $\langle w, \psi(x)\rangle$，$\psi(x)$ 表示所有阶数不超过 r 的单项式。因此，为 $p(x)$ 构建压缩机制的问题约简成为 $\mathbb{R}^{d'}$ 上的半空间构建压缩机制，其中 $d'=O(d^r)$。

[362]

30.2.4 间隔可分的情况

假定一个训练集可以由间隔 γ 分开，感知器算法保证了收敛到一个在整个训练集都不

⊖ 可以证明 w 就是最大间隔解的方向。

发生错误的解至多需要$\frac{1}{\gamma}$次迭代。因此，我们可以得到一个大小为$k \leq 1/\gamma^2$的压缩机制。

30.3 文献评注

压缩机制及其与学习的关系由 Littlestone 和 Warmuth(1986)引出。如我们之前所述，如果一个类有压缩机制那么它便是可学习的，对于二分类问题，结合学习的基本定理可知，这样的类有一个有限的 VC 维。另一个方面，是否每个有限 VC 维的类都有有限大小的压缩机制还是一个未知的问题，该问题由 Manfred Warmuth 提出并且到现在仍未解(见 Floyd(1989)，Floyd&Warmuth(1995)，Ben-David&Litman(1998)，Livni&Simon (2013))。

[363]

PAC-贝叶斯

最小描述长度(MDL)和奥卡姆剃刀原则虽然允许存在一个可能很大的假设类，但定义了假设分层并且倾向选择在分层中出现在更高层的假设。在这一章，我们将阐述 PAC-贝叶斯法来将上述概念推广。在 PAC-贝叶斯中，先验知识通过给假设类定义先验分布来表达。

31.1 PAC-贝叶斯界

在 MDL 机制中，我们在类$\mathcal{H}$的假设中定义一种分层。现在，分层以假设类$\mathcal{H}$上的先验分布的形式来表达。这就是说，我们为每一个假设$h\in\mathcal{H}$分配了一个概率(如果$\mathcal{H}$连续便是概率密度)$P(h)\geq 0$，并且将 $P(h)$作为 h 的先验得分。根据贝叶斯推理，学习算法的输出不一定是一个单一假设，而可以是给假设类$\mathcal{H}$输出一个后验概率分布，记为 Q。在监督学习问题中，假设类$\mathcal{H}$包含了从$\mathcal{X}$到$\mathcal{Y}$的函数，那么 Q 可以被认为是定义了如下随机预测的规则。一旦得到一个新的样本 x，我们根据 Q 随机地抽选假设$h\in\mathcal{H}$并预测得到 $h(x)$。我们将 Q 在一个样本 z 上的损失定义如下：

$$\ell(Q,z)\stackrel{\text{def}}{=}\mathop{\mathbb{E}}_{h\sim Q}[\ell(h,z)]$$

由于期望是线性的，Q 的泛化误差和训练误差可以写作：

$$L_{\mathcal{D}}(Q)\stackrel{\text{def}}{=}\mathop{\mathbb{E}}_{h\sim Q}[L_{\mathcal{D}}(h)]\quad 和\quad L_S(Q)\stackrel{\text{def}}{=}\mathop{\mathbb{E}}_{h\sim Q}[L_S(h)]$$

下面的定理根据 Q 和先验分布 P 之间的 K-L(Kullback-Leibler)散度告诉我们 Q 的泛化误差和经验误差的差异是可以用界来约束的。K-L 散度是描述两个分布的差异的一种自然的度量。该定理表明如果我们想要最小化 Q 的泛化误差，应该同时最小化 Q 的经验误差以及 Q 与先验分布之间的 K-L 距离。我们之后将说明在某些情况下这个想法是如何导出正则风险最小化原则的。 [364]

定理 31.1 令$\mathcal{D}$为样本域 Z 上的任意分布。令$\mathcal{H}$是一个假设类，$\ell:\mathcal{H}\times Z\to[0,1]$为损失函数。令 P 是$\mathcal{H}$上的先验分布，$\delta\in(0,1)$。则对于根据分布$\mathcal{D}$采样得到独立同分布的训练集 $S=\{z_1,\ldots,z_m\}$，对于所有在$\mathcal{H}$的分布 Q(尽管与 S 有关)，以至少 $1-\delta$ 的概率有下式成立：

$$L_{\mathcal{D}}(Q)\leq L_S(Q)+\sqrt{\frac{D(Q\|P)+\ln m/\delta}{2(m-1)}}$$

其中

$$D(Q\|P)\stackrel{\text{def}}{=}\mathop{\mathbb{E}}_{h\sim Q}[\ln(Q(h)/P(h))]$$

为 Kullback-Leibler 散度。

证明 对任意函数 $f(S)$，根据马尔可夫不等式：

$$\mathop{\mathbb{P}}_{S}[f(S)\geq\epsilon]=\mathop{\mathbb{P}}_{S}[e^{f(S)}\geq e^{\epsilon}]\leq\frac{\mathop{\mathbb{E}}_{S}[e^{f(S)}]}{e^{\epsilon}}\tag{31.1}$$

令 $\Delta(h)=L_{\mathcal{D}}(h)-L_S(h)$，接下来我们将利用式(31.1)并选择如下函数：

$$f(S)=\sup_{Q}\Big(2(m-1)\mathop{\mathbb{E}}_{h\sim Q}(\Delta(h))^2-D(Q\|P)\Big)$$

接下来，我们界定 $\mathbb{E}_S[e^{f(S)}]$。主要的技巧在于用一个不依赖 Q 但是依赖先验概率 P 的表达式来求 $f(S)$ 的上界。即固定 S 并注意到由 $D(Q\|P)$ 的定义可以得到，对于所有的 Q 有下式成立：

$$\begin{aligned}2(m-1)\mathop{\mathbb{E}}_{h\sim Q}(\Delta(h))^2-D(Q\|P)&=\mathop{\mathbb{E}}_{h\sim Q}\left[\ln\left(e^{2(m-1)(\Delta(h))^2}P(h)/Q(h)\right)\right]\\&\leqslant\ln\mathop{\mathbb{E}}_{h\sim Q}\left[e^{2(m-1)(\Delta(h))^2}P(h)/Q(h)\right]\\&=\ln\mathop{\mathbb{E}}_{h\sim P}\left[e^{2(m-1)(\Delta(h))^2}\right]\end{aligned}\tag{31.2}$$

其中不等式部分根据期望不等式以及 log 函数的凹性得出。因此，

$$\mathop{\mathbb{E}}_{S}\left[e^{f(S)}\right]\leqslant\mathop{\mathbb{E}}_{S}\mathop{\mathbb{E}}_{h\sim P}\left[e^{2(m-1)(\Delta(h))^2}\right]\tag{31.3}$$

这个表达式的右半部分使我们可以调换两个求期望的顺序(因为 P 是一个先验分布，且不依赖于样本 S)，因此，

365

$$\mathop{\mathbb{E}}_{S}\left[e^{f(S)}\right]\leqslant\mathop{\mathbb{E}}_{h\sim P}\mathop{\mathbb{E}}_{S}\left[e^{2(m-1)(\Delta(h))^2}\right]\tag{31.4}$$

接下来，我们断言对于所有 h，有 $\mathop{\mathbb{E}}_{S}\left[e^{2(m-1)(\Delta(h))^2}\right]\leqslant m$。为了说明这点，我们考虑 Hoeffding 不等式，即

$$\mathop{\mathbb{P}}_{S}[\Delta(h)\geqslant\varepsilon]\leqslant e^{-2m\varepsilon^2}$$

这意味着 $\mathbb{E}_S\left[e^{2(m-1)(\Delta(h))^2}\right]\leqslant m$(见练习 31.1)。将该式与式(31.4)代入式(31.1)可以得到

$$\mathop{\mathbb{P}}_{S}[f(S)\geqslant\varepsilon]\leqslant\frac{m}{e^{\varepsilon}}\tag{31.5}$$

将上式右半部分与 δ 对应，令 $\varepsilon=\ln(m/\delta)$，故而我们得到，对于所有 Q，以至少 $1-\delta$ 的概率有

$$2(m-1)\mathop{\mathbb{E}}_{h\sim Q}(\Delta(h))^2-D(Q\|P)\leqslant\varepsilon=\ln(m/\delta)$$

重新排列不等式的顺序并再次利用詹牛不等式(注意到函数 x^2 是凸的)我们最终得到

$$\Big(\mathop{\mathbb{E}}_{h\sim Q}\Delta(h)\Big)^2\leqslant\mathop{\mathbb{E}}_{h\sim Q}(\Delta(h))^2\leqslant\frac{\ln(m/\delta)+D(Q\|P)}{2(m-1)}\tag{31.6}$$ ■

评注(正则化) PAC-贝叶斯界引出如下学习规则：

给定一个先验 P，返回一个后验 Q，并最小化如下函数：

$$L_S(Q)+\sqrt{\frac{D(Q\|P)+\ln m/\delta}{2(m-1)}}\tag{31.7}$$

这个规则与正则风险最小化原则很像，即，我们需要最小化 Q 在样本上的经验损失以及 Q 与 P 之间的 K-L 距离。

31.2 文献评注

PAC-贝叶斯界最早由 McAllester 于 1988 年提出，亦见 McAllester(1999)，McAllester(2003)，Seeger(2003)，Langford & Shawe-Taylor(2003)，Langford(2006)。

31.3 练习

366

31.1 令 X 是一个随机变量，满足 $\mathbb{P}[X\geqslant\varepsilon]\leqslant e^{-2m\varepsilon^2}$，证明 $\mathbb{E}[e^{2(m-1)X^2}]\leqslant m$。

31.2 1) 假定$\mathcal{H}$是一个有限假设类，并认为$\mathcal{H}$上的假设服从均匀分布，设后验为对于某些假设h_S，$Q(h_S)=1$，对于其他假设$h\in\mathcal{H}$，$Q(h)=0$，试证下式成立：

$$L_{\mathcal{D}}(h_S)\leq L_S(h)+\sqrt{\frac{\ln(|\mathcal{H}|)+\ln(m/\delta)}{2(m-1)}}$$

并将该式与我们用一致收敛得到的界做比较。

2) 用PAC-贝叶斯界得到一个界，它类似第7章给出的奥卡姆界。 367

技术引理

引理 A.1 令 $a>0$，则有 $x\geq 2a\log(a)\Rightarrow x\geq a\log(x)$。进一步，$x<a\log(x)$ 的必要条件是 $x<2a\log(a)$。

证明 首先注意，对于 $a\in(0,\sqrt{e}]$，$x\geq a\log(x)$ 显然成立，从而引理的结论成立。对于 $a>\sqrt{e}$，考察函数 $f(x)=x-a\log(x)$。其导数为 $f'(x)=1-a/x$。于是对于 $x>a$，导数是正定且递增的。此外，

$$\begin{aligned} f(2a\log(a)) &= 2a\log(a)-a\log(2a\log(a)) \\ &= 2a\log(a)-a\log(a)-a\log(2\log(a)) \\ &= a\log(a)-a\log(2\log(a)) \end{aligned}$$

可由 $a-2\log(a)>0$ 对所有的 $a>0$ 均成立，从而引理得证。 ■

引理 A.2 令 $a\geq 1$ 且 $b>0$，则 $x\geq 4a\log(2a)+2b\Rightarrow x\geq a\log(x)+b$。

证明 只须证明 $x\geq 4a\log(2a)+2b$ 意味着 $x\geq 2a\log(x)$ 且 $x\geq 2b$。由于假定 $a\geq 1$，容易得到 $x\geq 2b$。此外，由 $b>0$ 可知 $x\geq 4a\log(2a)$，再由引理 A.1 可知 $x\geq 2a\log(x)$。引理得证。 ■

引理 A.3 令 X 是一个随机变量，$x'\in\mathbb{R}$ 是一个标量。假定存在 $a>0$ 使得对于所有 $t>0$，有 $\mathbb{P}[|X-x'|>t]\leq 2e^{-t^2/a^2}$。那么，$\mathbb{E}[|X-x'|]\leq 4a$。

证明 对于所有的 $i=1,2,\ldots$，令 $t_i=ai$。由于 t_i 是单调递增的，可知 $\sum_{i=1}^{\infty}t_i\mathbb{P}[|X-x'|>t_i]$ 是 $\mathbb{E}[|X-x'|]$ 的上界。由此及引理的假设，有 $\mathbb{E}[|X-x'|]\leq 2a\sum_{i=1}^{\infty}ie^{-(i-1)^2}$。引理结论由如下不等式得到

[369]

$$\sum_{i=1}^{\infty}ie^{-(i-1)^2}\leq\sum_{i=1}^{3}ie^{-(i-1)^2}+\int_{2}^{\infty}xe^{-(x-1)^2}dx<1.8+10^{-7}<2$$ ■

引理 A.4 令 X 是一个随机变量，$x'\in\mathbb{R}$ 是一个标量。假定存在 $a>0$ 和 $b\geq e$ 使得对于所有 $t>0$，有 $\mathbb{P}[|X-x'|>t]\leq 2be^{-t^2/a^2}$。那么，$\mathbb{E}[|X-x'|]\leq a(2+\sqrt{\log(b)})$。

证明 对于所有的 $i=1,2,\ldots$，令 $t_i=a(i+\sqrt{\log(b)})$。由 t_i 是单调递增的，可知

$$\mathbb{E}[|X-x'|]\leq a\sqrt{\log(b)}+\sum_{i=1}^{\infty}t_i\mathbb{P}[|X-x'|>t_{i-1}]$$

由引理中假设，有

$$\begin{aligned} \sum_{i=1}^{\infty}t_i\mathbb{P}[|X-x'|>t_{i-1}] &\leq 2ab\sum_{i=1}^{\infty}(i+\sqrt{\log(b)})e^{-(i+\sqrt{\log(b)})^2} \\ &\leq 2ab\int_{1+\sqrt{\log(b)}}^{\infty}xe^{-(x-1)^2}dx \end{aligned}$$

$$
\begin{aligned}
&= 2ab\int_{\sqrt{\log(x)}}^{\infty}(y+1)e^{-y^2}\,dy \\
&\leqslant 4ab\int_{\sqrt{\log(x)}}^{\infty}ye^{-y^2}\,dy \\
&= 2ab\left[-e^{-y^2}\right]_{\sqrt{\log(x)}}^{\infty} \\
&= 2ab/b = 2a
\end{aligned}
$$

结合上面两个不等式，引理得证。 ■

引理 A.5 假设 m，d 是两个正整数且 $d\leqslant m-2$，那么

$$\sum_{k=0}^{d}\binom{m}{k}\leqslant\left(\frac{em}{d}\right)^{d}$$

证明 我们将采用归纳法来证明本引理。对于 $d=1$，等式左边等于 $1+m$，等式右边等于 em，于是结论成立。假定结论对于 d 成立，我们现在来证明结论对于 $d+1$ 成立。由归纳假设，有

$$
\begin{aligned}
\sum_{k=0}^{d+1}\binom{m}{k} &\leqslant \left(\frac{em}{d}\right)^{d}+\binom{m}{d+1} \\
&= \left(\frac{em}{d}\right)^{d}\left(1+\left(\frac{d}{em}\right)^{d}\frac{m(m-1)(m-2)\cdots(m-d)}{(d+1)d!}\right) \\
&\leqslant \left(\frac{em}{d}\right)^{d}\left(1+\left(\frac{d}{e}\right)^{d}\frac{(m-d)}{(d+1)d!}\right)
\end{aligned}
$$

370

根据 Stirling 不等式，进一步有

$$
\begin{aligned}
&\leqslant \left(\frac{em}{d}\right)^{d}\left(1+\left(\frac{d}{e}\right)^{d}\frac{(m-d)}{(d+1)\sqrt{2\pi d}\,(d/e)^{d}}\right) \\
&= \left(\frac{em}{d}\right)^{d}\left(1+\frac{m-d}{\sqrt{2\pi d}(d-1)}\right) \\
&= \left(\frac{em}{d}\right)^{d}\cdot\frac{d+1+(m-d)/\sqrt{2\pi d}}{d+1} \\
&\leqslant \left(\frac{em}{d}\right)^{d}\cdot\frac{d+1+(m-d)/2}{d+1} \\
&= \left(\frac{em}{d}\right)^{d}\cdot\frac{d/2+1-m/2}{d+1} \\
&\leqslant \left(\frac{em}{d}\right)^{d}\cdot\frac{m}{d+1}
\end{aligned}
$$

其中，最后一个不等式由假设 $d\leqslant m-2$ 得到。另一方面，

$$
\begin{aligned}
\left(\frac{em}{d+1}\right)^{d+1} &= \left(\frac{em}{d}\right)^{d}\cdot\frac{em}{d+1}\cdot\left(\frac{d}{d+1}\right)^{d} \\
&= \left(\frac{em}{d}\right)^{d}\cdot\frac{em}{d+1}\cdot\frac{1}{(1+1/d)^{d}} \\
&\geqslant \left(\frac{em}{d}\right)^{d}\cdot\frac{em}{d+1}\cdot\frac{1}{e} \\
&= \left(\frac{em}{d}\right)^{d}\cdot\frac{m}{d+1}
\end{aligned}
$$

这证明了我们归纳的结论。 ■

引理 A.6 对于所有 $a \in \mathbb{R}$，有

$$\frac{e^a - e^{-a}}{2} \leqslant e^{a^2/2}$$

证明 因为

$$e^a = \sum_{n=0}^{\infty} \frac{a^n}{n!}$$

于是，有

$$\frac{e^a + e^{-a}}{2} = \sum_{n=0}^{\infty} \frac{a^{2n}}{(2n)!}$$

且

$$e^{a^2/2} = \sum_{n=0}^{\infty} \frac{a^{2n}}{2^n n!}$$

371 注意，$(2n)! \geqslant 2^n n!$ 对所有 $n \geqslant 0$ 成立，从而证明完成。 ■

测度集中度

假设 $Z_1, \cdots, Z_m$ 为一列独立同分布随机变量，均值为 μ。强大数定律表明，当 m 趋于无穷时，经验平均值 $\frac{1}{m}\sum_{i=1}^{m} Z_i$ 以概率 1 收敛于期望值 μ。测度集中度不等式量化了当 m 为有限值时，经验平均值相对于期望值的偏差。

B.1 马尔可夫不等式

首先介绍马尔可夫不等式。假定 Z 是一个非负随机变量，Z 的期望可以写为

$$\mathbb{E}[Z] = \int_{x=0}^{\infty} \mathbb{P}[Z \geq x]\mathrm{d}x \tag{B.1}$$

由 $\mathbb{P}[Z \geq x]$ 单调非增，得到

$$\forall a \geq 0, \quad \mathbb{E}[Z] \geq \int_{x=0}^{a} \mathbb{P}[Z \geq x]\mathrm{d}x \geq \int_{x=0}^{a} \mathbb{P}[Z \geq a]\mathrm{d}x = a\,\mathbb{P}[Z \geq a] \tag{B.2}$$

整理上式便得到马尔可夫不等式：

$$\forall a \geq 0, \quad \mathbb{P}[Z \geq a] \leq \frac{\mathbb{E}[Z]}{a} \tag{B.3}$$

对于取值于 $[0, 1]$ 的随机变量，其马尔可夫不等式如下：

引理 B.1 设 Z 是一个取值于 $[0, 1]$ 的随机变量。假定 $\mathbb{E}[Z] = \mu$，那么对于任意的 $a \in (0, 1)$，有

$$\mathbb{P}[Z > 1 - a] \geq \frac{\mu - (1 - a)}{a}$$

这表明对于任意的 $a \in (0, 1)$，有

$$\mathbb{P}[Z > a] \geq \frac{\mu - a}{1 - a} \geq \mu - a$$ [372]

证明 令 $Y = 1 - Z$，则 Y 是非负随机变量，且 $\mathbb{E}[Y] = 1 - \mathbb{E}[Z] = 1 - \mu$。由关于 Y 的马尔可夫不等式，有

$$\mathbb{P}[Z \leq 1 - a] = \mathbb{P}[1 - Z \geq a] = \mathbb{P}[Y \geq a] \leq \frac{\mathbb{E}[Y]}{a} = \frac{1 - \mu}{a}$$

所以，

$$\mathbb{P}[Z > 1 - a] \geq 1 - \frac{1 - \mu}{a} = \frac{a + \mu - 1}{a}$$ ■

B.2 切比雪夫不等式

对随机变量 $(Z - \mathbb{E}(Z))^2$ 应用马尔可夫不等式，就得到了切比雪夫不等式：

$$\forall a > 0, \quad \mathbb{P}[|Z - \mathbb{E}[Z]| \geq a] = \mathbb{P}[(Z - \mathbb{E}[Z])^2 \geq a^2] \leq \frac{\mathrm{Var}[Z]}{a^2} \tag{B.4}$$

其中 $\mathrm{Var}[Z] = \mathbb{E}[(Z - \mathbb{E}(Z))^2]$ 是 Z 的方差。

考察随机变量$\frac{1}{m}\sum_{i=1}^{m} Z_i$。由于$Z_1, \cdots, Z_m$独立同分布，容易验证

$$\mathrm{Var}\left[\frac{1}{m}\sum_{i=1}^{m} Z_i\right] = \frac{\mathrm{Var}[Z_1]}{m}$$

应用切比雪夫不等式，有如下结论：

引理 B.2　设$Z_1, \cdots, Z_m$是独立同分布随机变量，假定$\mathbb{E}[Z_1]=\mu$且$\mathrm{Var}[Z_1]\leqslant 1$，那么对于任意的$\delta\in(0,1)$，

$$\left|\frac{1}{m}\sum_{i=1}^{m} Z_i - \mu\right| \leqslant \sqrt{\frac{1}{\delta m}}$$

成立的概率大于$1-\delta$。

证明　由切比雪夫不等式，对于所有的$a>0$，有

$$\mathbb{P}\left[\left|\frac{1}{m}\sum_{i=1}^{m} Z_i - \mu\right| > a\right] \leqslant \frac{\mathrm{Var}[Z_1]}{ma^2} \leqslant \frac{1}{ma^2}$$

令上式右边为δ求解a，便得到引理结论。 ■

我们已经看到，经验平均值和对于期望值的偏差是随着m多项式下降的。试图获得更快的下降速度，是有可能实现的。事实上，在下面几节中，我们推导出偏差的上界是呈指数下降的。

B.3　切尔诺夫界

假设$Z_1, \cdots, Z_m$是独立的伯努利变量，其中对任意i，$\mathbb{P}[Z_i=1]=p_i$且$\mathbb{P}[Z_i=0]=1-p_i$。令$p=\sum_{i=1}^{m} p_i$且$Z=\sum_{i=1}^{m} Z_i$。利用指数函数的单调性和马尔可夫不等式，对于任意

[379] $t>0$，有

$$\mathbb{P}[Z>(1+\delta)p] = \mathbb{P}[e^{tZ} > e^{t(1+\delta)p}] \leqslant \frac{\mathbb{E}[e^{tZ}]}{e^{t(1+\delta)p}} \tag{B.5}$$

接下来，

$$\begin{aligned}
\mathbb{E}[e^{tZ}] &= \mathbb{E}[e^{t\sum_i Z_i}] = \mathbb{E}\left[\prod_i e^{tZ_i}\right] \\
&= \prod_i \mathbb{E}[e^{tZ_i}] && \text{由独立性} \\
&= \prod_i (p_i e^t + (1-p_i)e^0) \\
&= \prod_i (1+p_i(e^t-1)) \\
&\leqslant \prod_i e^{p_i(e^t-1)} && \text{利用 } 1+x\leqslant e^x \\
&= e^{\sum_i p_i(e^t-1)} \\
&= e^{(e^t-1)p}
\end{aligned}$$

结合上式及式(B.5)并选取$t=\log(1+\delta)$，有

引理 B.3　假设$Z_1, \cdots, Z_m$是独立的伯努利变量，其中对任意i，$\mathbb{P}[Z_i=1]=p_i$且$\mathbb{P}[Z_i=0]=1-p_i$。令$p=\sum_{i=1}^{m} p_i$且$Z=\sum_{i=1}^{m} Z_i$。那么对于任意的$\delta>0$，有

$$\mathbb{P}[Z > (1-\delta)p] \leqslant \mathrm{e}^{-h(\delta)p}$$

其中，

$$h(\delta) = (1-\delta)\log(1+\delta) - \delta$$

利用不等式 $h(a) \geqslant a^2/(2+2a/3)$，有

引理 B.4 在引理 B.3 的假设下，还能得到如下结论：

$$\mathbb{P}[Z > (1+\delta)p] \leqslant \mathrm{e}^{-p\frac{\delta^2}{2+2\delta/3}}$$

另一方面，类似可得

$$\mathbb{P}[Z < (1-\delta)p] = \mathbb{P}[-Z < -(1-\delta)p] = \mathbb{P}[\mathrm{e}^{-\alpha Z} < \mathrm{e}^{-\alpha(1-\delta)p}] \leqslant \frac{\mathbb{E}(\mathrm{e}^{-\alpha Z})}{\mathrm{e}^{-\alpha(1-\delta)p}} \quad (B.8)$$ 374

且

$$\begin{aligned}
\mathbb{E}[\mathrm{e}^{-\alpha Z}] &= \mathbb{E}[\mathrm{e}^{-\alpha\sum_i X_i}] = \mathbb{E}[\prod_i \mathrm{e}^{-\alpha X_i}] \\
&= \prod_i \mathbb{E}[\mathrm{e}^{-\alpha X_i}] && \text{由独立性} \\
&= \prod_i (1 + p_i(\mathrm{e}^{-\alpha}-1)) \\
&\leqslant \prod_i \mathrm{e}^{p_i(\mathrm{e}^{-\alpha}-1)} && \text{利用 } 1-x \leqslant \mathrm{e}^{-x} \\
&= \mathrm{e}^{(\mathrm{e}^{-\alpha}-1)p}
\end{aligned}$$

令 $t = -\log(1-\delta)$，有

$$\mathbb{P}[Z < (1-\delta)p] \leqslant \frac{\mathrm{e}^{-\delta p}}{\mathrm{e}^{(1-\delta)\log(1-\delta)p}} = \mathrm{e}^{-h(-\delta)p}$$

容易验证 $h(-\delta) \geqslant h(\delta)$，从而得到

引理 B.5 在引理 B.3 的假设下，还能得到如下结论：

$$\mathbb{P}[Z < (1-\delta)p] \leqslant \mathrm{e}^{-h(-\delta)p} \leqslant \mathrm{e}^{-h(\delta)p} \leqslant \mathrm{e}^{-p\frac{\delta^2}{2+2\delta/3}}$$

B.4 Hoeffding 不等式

引理 B.6（Hoeffding 不等式） 假设 $Z_1, \cdots, Z_m$ 是一列独立同分布随机变量，令 $Z = \frac{1}{m}\sum_{i=1}^{m} Z_i$。假定 $\mathbb{E}[Z] = \mu$ 且 $P[a \leqslant Z_i \leqslant b] = 1$ 对所有 i 成立。那么对任意 $\varepsilon > 0$ 有

$$\mathbb{P}\left[\left|\frac{1}{m}\sum_{i=1}^{m} Z_i - \mu\right| > \varepsilon\right] \leqslant 2\exp\left(-2m\varepsilon^2/(b-a)^2\right)$$

证明 记 $X_i = Z_i - \mathbb{E}[Z_i]$ 且 $\overline{X} = \frac{1}{m}\sum_{i=1}^{m} X_i$。由指数函数的单调性及马尔可夫不等式，对任意的 $\lambda > 0$ 和 $\varepsilon > 0$，有

$$\mathbb{P}[\overline{X} \geqslant \varepsilon] = \mathbb{P}[\mathrm{e}^{\lambda\overline{X}} \geqslant \mathrm{e}^{\lambda\varepsilon}] \leqslant \mathrm{e}^{-\lambda\varepsilon}\,\mathbb{E}[\mathrm{e}^{\lambda\overline{X}}]$$

由独立性假设有

$$\mathbb{E}[\mathrm{e}^{\lambda\overline{X}}] = \mathbb{E}\left[\prod_i \mathrm{e}^{\lambda X_i/m}\right] = \prod_i \mathbb{E}[\mathrm{e}^{\lambda X_i/m}]$$

再由 Hoeffding 引理（引理 B.7），对任意 i 有

$$\mathbb{E}[e^{\lambda(X_i-m)}]\leqslant e^{\frac{\lambda^2(b-a)^2}{8n^2}}$$

因此，

$$\mathbb{P}[\overline{X}\geqslant\varepsilon]\leqslant e^{-\lambda\varepsilon}\prod_i e^{\frac{\lambda^2(b-a)^2}{8n^2}}=e^{-\lambda\varepsilon+\frac{\lambda^2(b-a)^2}{8n}}$$

375

令 $\lambda=4n\varepsilon/(b-a)^2$，则

$$\mathbb{P}[\overline{X}\geqslant\varepsilon]\leqslant e^{-\frac{2n\varepsilon^2}{(b-a)^2}}$$

类似地，对变量 $-\overline{X}$ 进行讨论，可以得到 $P[\overline{X}\leqslant-\varepsilon]\leqslant e^{-\frac{2n\varepsilon^2}{(b-a)^2}}$，联合上述两个结论，引理得证。 ■

引理 B.7（Hoeffding 引理）　设 X 是一个随机变量，取值于区间 $[a,b]$ 且满足 $\mathbb{E}[X]=0$，那么，对于任意的 $\lambda>0$，有

$$\mathbb{E}[e^{\lambda X}]\leqslant e^{\lambda^2(b-a)^2/8}$$

证明　由于函数 $f(x)=e^{\lambda x}$ 是凸函数，我们知道对于任意 $\alpha\in(0,1)$ 和 $x\in[a,b]$，有

$$f(x)\leqslant\alpha f(a)+(1-\alpha)f(b)$$

令 $\alpha=\frac{b-x}{b-a}\in[0,1]$，则

$$e^{\lambda x}\leqslant\frac{b-x}{b-a}e^{\lambda a}+\frac{x-a}{b-a}e^{\lambda b}$$

对上式取期望，又由于 $\mathbb{E}[X]=0$，我们得到

$$\mathbb{E}[e^{\lambda X}]\leqslant\frac{b-\mathbb{E}[X]}{b-a}e^{\lambda a}+\frac{\mathbb{E}[X]-a}{b-a}e^{\lambda b}=\frac{b}{b-a}e^{\lambda a}-\frac{a}{b-a}e^{\lambda b}$$

记 $h=\lambda(b-a)$，$p=\frac{-a}{b-a}$ 且 $L(h)=-hp+\log(1-p+pe^h)$，则上式右边可以写为 $e^{L(h)}$。因此，为证明命题，只须证明 $L(h)\leqslant\frac{h^2}{8}$。根据泰勒公式，又 $L(0)=L'(0)=0$ 和 $L''(h)\leqslant1$ 对所有 h 均成立，可以得证。 ■

B.5　Bennet 和 Bernsein 不等式

Bennet 和 Bernsein 不等式与切尔诺夫界是相似的，但是它们对于任意独立随机变量序列都成立。这里只给出结论，略去证明过程，有兴趣的读者，可以查阅 Casa Bianchi 和 Lugosi (2006)。

引理 B.8（Bennet 不等式）　假设 $Z_1,\cdots,Z_n$ 是一列独立随机变量，均值为 0，且 $Z_i\leqslant1$ 的概率为 1。令

$$\sigma^2\geqslant\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}[Z_i^2]$$

那么对任意的 $\varepsilon>0$，有

$$\mathbb{P}\left[\sum_{i=1}^{n}Z_i>\varepsilon\right]\leqslant e^{-n\sigma^2h(\varepsilon/n\sigma^2)}$$

其中

376

$$h(a)=(1+a)\log(1+a)-a$$

根据 $h(a)\geq a^2/(2+2a/3)$，容易推导出：

引理 B.9(Bernsein 不等式) 假设 $Z_1, \cdots, Z_m$ 是一列独立随机变量，均值为 0。如果对所有的 i，$\mathbb{P}(Z_i\leq M)=1$ 成立，那么对于任意的 $t>0$，有

$$\mathbb{P}\left[\sum_{i=1}^{m} Z_i > t\right] \leq \exp\left(-\frac{t^2/}{\sum \mathbb{E}Z_i^2 + Mt/3}\right)$$

应用

可实现假设 PAC 理论中样本复杂度的下界正比于 $1/\varepsilon$(第 2 章)，不可实现假设 PAC 理论中样本复杂度的下界正比于 $1/\varepsilon^2$(第 4 章)，而 Bernsein 不等式可以用来描述这两个比率之间的情形。

引理 B.10 设 $\ell:\mathcal{H}\times Z\to[0,1]$ 为损失函数，令 $\mathcal{D}$ 为 Z 上任意一个分布。固定 h，对任意的 $\delta\in(0,1)$，我们有

1. $\mathop{\mathbb{P}}_{S\sim\mathcal{D}^m}\left[L_S(h)\geq L_{\mathcal{D}}(h)+\sqrt{\frac{2L_{\mathcal{D}}(h)\log(1/\delta)}{3m}}+\frac{2\log(1/\delta)}{m}\right]\leq\delta$
2. $\mathop{\mathbb{P}}_{S\sim\mathcal{D}^m}\left[L_{\mathcal{D}}(h)\geq L_S(h)+\sqrt{\frac{2L_S(h)\log(1/\delta)}{m}}+\frac{4\log(1/\delta)}{m}\right]\leq\delta$

证明 定义随机变量 $\alpha_i, \cdots, \alpha_m$ 满足 $\alpha_i=\ell(h, z_i)-L_{\mathcal{D}}(h)$。注意到 $\mathbb{E}[\alpha_i]=0$ 且

$$\begin{aligned}
\mathbb{E}[\alpha_i^2] &= \mathbb{E}[\ell(h,z_i)^2]-2L_{\mathcal{D}}(h)\,\mathbb{E}[\ell(h,z_i)]+L_{\mathcal{D}}(h)^2\\
&= \mathbb{E}[\ell(h,z_i)^2]-L_{\mathcal{D}}(h)^2\\
&\leq \mathbb{E}[\ell(h,z_i)^2]\\
&\leq \mathbb{E}[\ell(h,z_i)]=L_{\mathcal{D}}(h)
\end{aligned}$$

在最后一个不等式中我们用到了事实 $\ell(h, z_i)\in[0, 1]$，于是 $\ell(h, z_i)^2\leq\ell(h, z_i)$。对 $\alpha_1, \cdots, \alpha_m$ 使用 Bernsein 不等式，有

$$\begin{aligned}
\mathbb{P}\left[\sum_{i=1}^{m}\alpha_i > t\right] &\leq \exp\left(-\frac{t^2/2}{\sum \mathbb{E}\alpha_i^2+t/3}\right)\\
&\leq \exp\left(-\frac{t^2/2}{mL_{\mathcal{D}}(h)+t/3}\right)\overset{\text{令}}{=}\delta
\end{aligned}$$

377

求解出 t，得到

$$\begin{aligned}
&\frac{t^2/2}{mL_{\mathcal{D}}(h)+t/3}=\log(1/\delta)\\
&\Rightarrow t^2/2-\frac{\log(1/\delta)}{3}t-\log(1/\delta)mL_{\mathcal{D}}(h)=0\\
&\Rightarrow t=\frac{\log(1/\delta)}{3}+\sqrt{\frac{\log^2(1/\delta)}{3^2}+2\log(1/\delta)mL_{\mathcal{D}}(h)}\\
&\leq 2\frac{\log(1/\delta)}{3}+\sqrt{2\log(1/\delta)mL_{\mathcal{D}}(h)}
\end{aligned}$$

由 $\frac{1}{m}\sum_i\alpha_i=L_S(h)-L_{\mathcal{D}}(h)$ 可知，在不小于 $1-\delta$ 的概率下，

$$L_S(h)-L_{\mathcal{D}}(h)\leq 2\frac{\log(1/\delta)}{3m}+\sqrt{\frac{2\log(1/\delta)L_{\mathcal{D}}(h)}{m}}$$

这就证明了第一个不等式。定理第二部分的证明是类似的，这里略去。 ■

B.6　Slud 不等式

令 X 是 (m, p) 二项随机变量，也就是，$X=\sum_{i=1}^{m} Z_i$，其中每个 Z_i 等于 1 的概率为 p，等于 0 的概率为 $1-p$。假定 $p=(1-\varepsilon)/2$。Slud 不等式（Slud，1977 年）表明，$\mathbb{P}[X\geq m/2]$ 的一个下界是正态随机变量大于或者等于 $\sqrt{m\varepsilon^2/(1-\varepsilon^2)}$ 的概率。下面的引理可由正态分布的标准尾边界得到。

引理 B.11　令 X 是 (m, p) 二项随机变量，且 $p=(1-\varepsilon)/2$，那么，

$$\mathbb{P}[X\geq m/2]\geq \frac{1}{2}\left(1-\sqrt{1-\exp(-m\varepsilon^2/(1-\varepsilon^2))}\right)$$

B.7　χ^2 随机变量的集中度

令 $X_1, \cdots, X_k$ 是 k 个独立同分布的正态随机变量；也就是对于每个 i，$X_i\sim N(0, 1)$，随机变量 X_i^2 的分布称为 χ^2 分布，随机变量 $Z=X_1^2+\cdots+X_k^2$ 的分布称为 χ_k^2 分布（自由度为 k），显然有 $\mathbb{E}[X_i^2]=1$ 且 $\mathbb{E}[Z]=k$。下面的引理说明，χ_k^2 分布是集中在其均值附近的。

引理 B.12　令 $Z\sim\chi_k^2$，则对于任意的 $\varepsilon>0$ 有

$$\mathbb{P}[Z\leq(1-\varepsilon)k]\leq e^{-\varepsilon^2 k/6}$$

此外，对于任意的 $\varepsilon\in(0, 3)$，我们有

$$\mathbb{P}[Z\geq(1+\varepsilon)k]\leq e^{-\varepsilon^2 k/6}$$

综上，对于任意的 $\varepsilon\in(0, 3)$，有

$$\mathbb{P}[(1-\varepsilon)k\leq Z\leq(1+\varepsilon)k]\geq 1-2e^{-\varepsilon^2 k/6}$$

[378]

证明　令 $Z=\sum_{i=1}^{k} X_i^2$，其中 $X_i\sim N(0, 1)$。为了证明引理，我们采用切比雪夫界估计方法。对于第一个不等式，我们首先估计 $\mathbb{E}[e^{-\lambda X_1^2}]$ 的界，其中 $\lambda>0$ 待定。由于 $e^{-a}\leq 1-a+\frac{a^2}{2}$ 对所有的 $a>0$ 成立，我们有

$$\mathbb{E}[e^{-\lambda X_1^2}]\leq 1-\lambda\,\mathbb{E}[X_1^2]+\frac{\lambda^2}{2}\,\mathbb{E}[X_1^4]$$

利用常见不等式，$\mathbb{E}[X_1^2]=1$ 且 $\mathbb{E}[X_1^4]=3$，及事实 $1-a\leq e^{-a}$，我们有

$$\mathbb{E}[e^{-\lambda X_1^2}]\leq 1-\lambda+\frac{3}{2}\lambda^2\leq e^{-\lambda+\frac{3}{2}\lambda^2}$$

再由切比雪夫界估计方法，我们得到

$$\begin{aligned}\mathbb{P}[Z\leq(1-\varepsilon)k]&=\mathbb{P}[e^{-\lambda Z}\geq e^{-\lambda(1-\varepsilon)k}]\\&\leq e^{\lambda(1-\varepsilon)k}\,\mathbb{E}[e^{-\lambda Z}]\\&=e^{\lambda(1-\varepsilon)k}\left(\mathbb{E}[e^{-\lambda X_1^2}]\right)^k\\&\leq e^{\lambda(1-\varepsilon)k}e^{-\lambda k+\frac{3}{2}\lambda^2 k}\\&=e^{-\lambda\varepsilon k+\frac{3}{2}\lambda^2 k}\end{aligned}$$

令 $\lambda=\varepsilon/3$，则引理的第一个不等式得证。

对于第二个不等式，我们知道 χ_d^2 分布的矩母函数(moment generating function)具有如下表示形式：

$$\forall \lambda < \frac{1}{2}, \quad \mathbb{E}[e^{\lambda Z^2}] = (1-2\lambda)^{-d/2} \tag{B.7}$$

由此及切比雪夫界估计方法，我们有

$$\begin{aligned}\mathbb{P}[Z \geq (1-\varepsilon)k] &= \mathbb{P}[e^{\lambda Z} \geq e^{(1-\varepsilon)\lambda k}] \\ &\leq e^{-(1-\varepsilon)\lambda k}\,\mathbb{E}[e^{\lambda Z}] \\ &= e^{-(1-\varepsilon)\lambda k}(1-2\lambda)^{-k/2} \\ &\leq e^{-(1-\varepsilon)\lambda k} e^{\lambda k} = e^{-\varepsilon\lambda k}\end{aligned}$$

其中，最后一个不等式成立是因为 $1-a \leq e^{a}$。令 $\lambda = \varepsilon/6$(即 $\lambda \in (0, 1/2)$)，我们就证明了引理的第二个不等式。

最后，将引理的前两个不等式结合起来，就得到了引理的最后一个不等式。 ■ 379

线性代数

C.1 基本概念

本章，我们只考虑有限维欧氏空间上的线性代数，向量是指列向量。

给定两个 d 维向量 $\boldsymbol{u}$，$\boldsymbol{v}\in\mathbb{R}^d$，定义它们的内积为

$$\langle \boldsymbol{u},\boldsymbol{v}\rangle=\sum_{i=1}^{d}u_iv_i$$

欧氏范数(即ℓ_2 范数)定义为$\|\boldsymbol{u}\|=\sqrt{\langle \boldsymbol{u},\boldsymbol{u}\rangle}$。我们也会用到$\ell_1$ 范数$\|\boldsymbol{u}\|_1=\sum_{i=1}^{d}|u_i|$，和$\ell_\infty$范数$\|\boldsymbol{u}\|_\infty=\max_i|u_i|$。

欧氏空间$\mathbb{R}^d$ 的子空间是关于加法运算和数乘运算封闭的$\mathbb{R}^d$ 子集。由向量 $\boldsymbol{u}_1$，…，$\boldsymbol{u}_k$ 张成的子空间是具有如下形式的向量的全体：

$$\sum_{i=1}^{k}\alpha_i\boldsymbol{u}_i$$

其中对于所有的 i，$\alpha_i\in\mathbb{R}$。

称向量集 $U=\{\boldsymbol{u}_1,\cdots,\boldsymbol{u}_k\}$是独立的，如果对于任意的 i，$\boldsymbol{u}_i$ 都不在 $\boldsymbol{u}_1$，…，$\boldsymbol{u}_{i-1}$，$\boldsymbol{u}_{i+1}$，…，$\boldsymbol{u}_k$ 张成的子空间内。我们称 U 张成子空间 V，如果 V 是由 U 中的向量张成的。V 的维数被定义为空间 V 的基的数量(可以验证 V 的所有基的数量都是相同的)。我们称 U 是正交集，如果对于任意的 $i\neq j$，都有$\langle \boldsymbol{u}_i,\boldsymbol{u}_j\rangle=0$；进一步，称 U 是标准正交集，如果对于任意的 i，都有$\|\boldsymbol{u}_i\|=1$。

给定矩阵 $A\in\mathbb{R}^{m\times d}$，$A$ 的域定义为其列向量张成的空间，而 A 的零空间定义为满足 $A\boldsymbol{u}=0$ 的所有向量构成的子空间。A 的秩即是 A 的域空间的维数。

矩阵 A 的转置矩阵，记为 A^{T}，其第(i,j)元素等于矩阵矩阵 A 的第(j,i)元素。如果 $A=A^{\mathrm{T}}$，则称 A 是对称矩阵。

C.2 特征值与特征向量

对于矩阵 $A\in\mathbb{R}^{d\times d}$，称非零向量 $\boldsymbol{u}$ 为 A 对应于特征值 λ 的特征向量，如果满足

$$A\boldsymbol{u}=\lambda\boldsymbol{u}$$

定理 C.1(谱分解) 如果 $A\in\mathbb{R}^{d\times d}$是对称矩阵，秩为 k，那么存在$\mathbb{R}^d$ 的一组标准正交基，$\boldsymbol{u}_1$，…，$\boldsymbol{u}_d$，使每个 $\boldsymbol{u}_i$ 都是 A 的特征向量。进一步，A 可以表示为 $A=\sum_{i=1}^{d}\lambda_i\boldsymbol{u}_i\boldsymbol{u}_i^{\mathrm{T}}$，其中 λ_i 是与$\boldsymbol{u}_i$ 相对应的特征值。这等价于 $A=UDU^{\mathrm{T}}$，其中，U 的列向量为 $\boldsymbol{u}_1$，…，$\boldsymbol{u}_d$，矩阵 D 是对角矩阵，满足 $D_{i,i}=\lambda_i$ 且对于 $i\neq j$ 有 $D_{i,j}=0$。此外，非零特征值的个数与矩阵的秩是相等的，非零特征值对应的特征向量张成的空间是 A 的域，零特征值对应的特征向量张成的空间是 A 的零空间。

C.3 正定矩阵

我们说对称矩阵 $A\in\mathbb{R}^{n\times n}$ 是正定的，如果 A 的特征值都是正的。如果 A 的特征值都是非负的，则称 A 是半正定的。

定理 C.2 令 $A\in\mathbb{R}^{n\times n}$ 是对称矩阵，下面是半正定性的等价定义：

- A 的特征值都是非负的。
- 对于任意向量 u，都有 $\langle u, Au\rangle\geq 0$。
- 存在矩阵 B 使得 $A=BB^{\top}$。

C.4 奇异值分解

假定矩阵 $A\in\mathbb{R}^{m\times n}$ 的秩为 r。当 $m\neq n$ 时，定理 C.1 给出的特征值分解不再适用。我们将给出矩阵的另一种分解方式，即奇异值分解(Single Value Decomposition，SVD)。

对于单位向量 $v\in\mathbb{R}^n$ 和 $u\in\mathbb{R}^m$，分别称它们为矩阵 A 对应于奇异值 $\sigma>0$ 的右奇异向量和左奇异向量，如果满足

$$Av=\sigma u,\quad A^{\top}u=\sigma v$$

我们首先证明：如果存在 r 个正奇异值对应的单位正交奇异向量，那么矩阵可分解为 $A=UDV^{\top}$，其中 U 和 V 的列向量分别是左奇异向量和右奇异向量，D 是一个 $r\times r$ 对角矩阵且对角线上的元素为奇异值。

引理 C.3 矩阵 $A\in\mathbb{R}^{m\times n}$ 的秩为 r。假定 $v_1,\cdots,v_r$ 是由 A 的右奇异向量组成的单位正交集，$u_1,\cdots,u_r$ 是由 A 的左奇异向量组成的单位正交集，$\sigma_1,\cdots,\sigma_r$ 是对应的奇异值。那么， [381]

$$A=\sum_{i=1}^{r}\sigma_i u_i v_i^{\top}$$

令 U 是以全体 u_i 为列向量的矩阵，V 是以全体 v_i 为列向量的矩阵，D 是对角矩阵满足 $D_{ii}=\sigma_i$，则

$$A=UDV^{\top}$$

证明 矩阵 A 的右奇异向量属于 $A^{\top}$ 的域空间(否则，对应的奇异值一定是 0)。因此，$v_1,\cdots,v_r$ 构成了 A 的域空间的一组单位正交基。可以找到向量 $v_{r+1},\cdots,v_n$ 使得 $v_1,\cdots,v_n$ 构成了 $\mathbb{R}^n$ 的单位正交基。定义 $B=\sum_{i=1}^{r}\sigma_i u_i v_i^{\top}$，则只需证明 $Av_i=Bv_i$。显然，当 $i>r$ 时，$Av_i=0$ 和 $Bv_i=0$ 均成立；对于 $i\leq r$，我们有

$$Bv_i=\sum_{j=1}^{r}\sigma_j u_j v_j^{\top}v_i=\sigma_i u_i=Av_i$$

其中最后一个等式是由奇异向量的定义得到的。 ■

下面的引理考察了 A 的奇异值与 $A^{\top}A$ 和 $AA^{\top}$ 的特征值之间的关系。

引理 C.4 v 和 u 分别是 A 的右奇异向量和左奇异向量，对应的奇异值是 σ；这等价于，v 是 $A^{\top}A$ 的特征向量，对应的特征值是 σ^2，而 $u=\sigma^{-1}Av$ 是 $AA^{\top}$ 的特征向量，对应的特征值也是 σ^2。

证明 假定 σ 是 A 的奇异值，对应的右奇异向量是 v，那么

$$A^{\top}Av=\sigma A^{\top}u=\sigma^2 v$$

类似地，

$$AA^{\mathrm{T}}u=\sigma Av=\sigma^2 u$$

另一方面，如果$\lambda\neq 0$是$A^{\mathrm{T}}A$的特征值，对应的特征向量是v，则由$A^{\mathrm{T}}A$是半正定的可知$\lambda>0$。令$\sigma=\sqrt{\lambda}$，$u=\sigma^{-1}Av$，则有

$$\sigma u=\sqrt{\lambda}\frac{Av}{\sqrt{\lambda}}=Av$$

且

$$A^{\mathrm{T}}u=\frac{1}{\sigma}A^{\mathrm{T}}Av=\frac{\lambda}{\sigma}v=\sigma v$$

■

[382] 最后，我们来证明如果A的秩是r，则它有r个单位正交奇异向量。

引理 C.5 矩阵$A\in\mathbb{R}^{m\times n}$的秩为$r$，定义

$$v_1=\underset{v\in\mathbb{R}^n,\|v\|=1}{\operatorname{argmax}}\|Av\|$$

$$v_2=\underset{\substack{v\in\mathbb{R}^n,\|v\|=1\\ v\cdot v_1=0}}{\operatorname{argmax}}\|Av\|$$

$$\vdots$$

$$v_r=\underset{\substack{v\in\mathbb{R}^n,\|v\|=1\\ \forall i<r,v\cdot v_i=0}}{\operatorname{argmax}}\|Av\|$$

则$v_1,\cdots,v_r$是由A的右奇异向量组成的单位正交集。

证明 首先注意到A的秩是r，则A的域空间是一个r维的子空间，从而容易验证对于所有的$i=1,\cdots,r$，$\|Av_i\|>0$成立。令$W\in\mathbb{R}^{n\times n}$是由$A^{\mathrm{T}}A$的特征值分解所确定的单位正交矩阵，即$A^{\mathrm{T}}A=WDW^{\mathrm{T}}$，其中$D$是对角矩阵，满足$D_{1,1}\geqslant D_{2,2}\geqslant\cdots\geqslant 0$。我们将证明$v_1,\cdots,v_r$是$A^{\mathrm{T}}A$的对应于非零特征值的特征向量，再由引理C.4可知，是A的右奇异向量。采用数学归纳法进行证明。注意到任意的单位向量v都可以表示为$v=Wx$，其中，$x=W^{\mathrm{T}}v$且$\|x\|=1$。那么

$$\|Av\|^2=\|AWx\|^2=\|WDW^{\mathrm{T}}Wx\|^2=\|WDx\|^2=\|Dx\|^2=\sum_{i=1}^{n}D_{i,i}^2x_i^2$$

因此，

$$\max_{v,\|v\|=1}\|Av\|^2=\max_{x,\|x\|=1}\sum_{i=1}^{n}D_{i,i}^2x_i^2$$

右式的解可以设定为$x=(1,0,\cdots,0)$，这表明v_1是$A^{\mathrm{T}}A$的最大特征值。由$\|Av_1\|>0$可知$D_{1,1}>0$满足归纳假设。现假定结论对于$1\leqslant i\leqslant r-1$成立。则任意正交于$v_1,\cdots,v_i$的向量$v$都可以表示为$v=Wx$，其中$x$的前$i$个元素为0。由此，

$$\max_{v,\|v\|=1,\forall j\leqslant i,v^{\mathrm{T}}v_j=0}\|Av\|^2=\max_{x,\|x\|=1,x_1=\cdots=x_i=0}\sum_{i=1}^{n}D_{i,i}^2x_i^2$$

右式的解为满足$x_{i+1}=1$且其余分量均为0的向量，这表明v_{i+1}是矩阵W的第$(i+1)$列。最后，再由$\|Av_{i+1}\|>0$可知$D_{i+1,i+1}>0$，引理得证。 ■

推论 C.6(SVD分解) 设矩阵$A\in\mathbb{R}^{m\times n}$的秩为$r$，则$A=UDV^{\mathrm{T}}$，其中，$D$是以$A$的非零奇异值为对角元素的$r\times r$对角矩阵，矩阵$U$，$V$的列向量分别是$A$的左奇异向量和右奇异向量。此外，对于所有的$i$，$D_{i,i}^2$是$A^{\mathrm{T}}A$的特征值，矩阵$V$的第$i$列是对应的$A^{\mathrm{T}}A$的特征向量，矩阵$U$的第$i$列是对应的$AA^{\mathrm{T}}$的特征向量。

[383]

参考文献

Abernethy, J., Bartlett, P. L., Rakhlin, A. & Tewari, A. (2008), "Optimal strategies and minimax lower bounds for online convex games," in *Proceedings of the nineteenth annual conference on computational learning theory*.

Ackerman, M. & Ben-David, S. (2008), "Measures of clustering quality: A working set of axioms for clustering," in *Proceedings of Neural Information Processing Systems* (NIPS), pp. 121–128.

Agarwal, S. & Roth, D. (2005), "Learnability of bipartite ranking functions," in *Proceedings of the 18th annual conference on learning theory*, pp. 16–31.

Agmon, S. (1954), "The relaxation method for linear inequalities," *Canadian Journal of Mathematics* **6**(3), 382–392.

Aizerman, M. A., Braverman, E. M. & Rozonoer, L. I. (1964), "Theoretical foundations of the potential function method in pattern recognition learning," *Automation and Remote Control* **25**, 821–837.

Allwein, E. L., Schapire, R. & Singer, Y. (2000), "Reducing multiclass to binary: A unifying approach for margin classifiers," *Journal of Machine Learning Research* **1**, 113–141.

Alon, N., Ben-David, S., Cesa-Bianchi, N. & Haussler, D. (1997), "Scale-sensitive dimensions, uniform convergence, and learnability," *Journal of the ACM* **44**(4), 615–631.

Anthony, M. & Bartlet, P. (1999), *Neural Network Learning: Theoretical Foundations*, Cambridge University Press.

Baraniuk, R., Davenport, M., DeVore, R. & Wakin, M. (2008), "A simple proof of the restricted isometry property for random matrices," *Constructive Approximation* **28**(3), 253–263.

Barber, D. (2012), *Bayesian reasoning and machine learning*, Cambridge University Press.

Bartlett, P., Bousquet, O. & Mendelson, S. (2005), "Local rademacher complexities," *Annals of Statistics* **33**(4), 1497–1537.

Bartlett, P. L. & Ben-David, S. (2002), "Hardness results for neural network approximation problems," *Theor. Comput. Sci.* **284**(1), 53–66.

Bartlett, P. L., Long, P. M. & Williamson, R. C. (1994), "Fat-shattering and the learnability of real-valued functions," in *Proceedings of the seventh annual conference on computational learning theory*, (ACM), pp. 299–310.

Bartlett, P. L. & Mendelson, S. (2001), "Rademacher and Gaussian complexities: Risk bounds and structural results," in *14th Annual Conference on Computational Learning Theory* (COLT) *2001*, Vol. 2111, Springer, Berlin, pp. 224–240.

Bartlett, P. L. & Mendelson, S. (2002), "Rademacher and Gaussian complexities: Risk bounds and structural results," *Journal of Machine Learning Research* **3**, 463–482.

Ben-David, S., Cesa-Bianchi, N., Haussler, D. & Long, P. (1995), "Characterizations of learnability for classes of $[0, \ldots, n]$-valued functions," *Journal of Computer and System Sciences* **50**, 74–86.

Ben-David, S., Eiron, N. & Long, P. (2003), "On the difficulty of approximately maximizing agreements," *Journal of Computer and System Sciences* **66**(3), 496–514.

Ben-David, S. & Litman, A. (1998), "Combinatorial variability of vapnik-chervonenkis classes with applications to sample compression schemes," *Discrete Applied Mathematics* **86**(1), 3–25.

Ben-David, S., Pal, D., & Shalev-Shwartz, S. (2009), "Agnostic online learning," in Conference on Learning Theory (COLT).

Ben-David, S. & Simon, H. (2001), "Efficient learning of linear perceptrons," *Advances in Neural Information Processing Systems*, pp. 189–195.

Bengio, Y. (2009), "Learning deep architectures for AI," *Foundations and Trends in Machine Learning* **2**(1), 1–127.

Bengio, Y. & LeCun, Y. (2007), "Scaling learning algorithms towards AI," *Large Scale Kernel Machines* **34**.

Bertsekas, D. (1999), *Nonlinear programming*, Athena Scientific.

Beygelzimer, A., Langford, J. & Ravikumar, P. (2007), "Multiclass classification with filter trees," *Preprint, June* .

Birkhoff, G. (1946), "Three observations on linear algebra," *Revi. Univ. Nac. Tucuman, ser. A* **5**, 147–151.

Bishop, C. M. (2006), *Pattern recognition and machine learning*, Vol. 1, Springer: New York.

Blum, L., Shub, M. & Smale, S. (1989), "On a theory of computation and complexity over the real numbers: Np-completeness, recursive functions and universal machines," *Am. Math. Soc.* **21**(1), 1–46.

Blumer, A., Ehrenfeucht, A., Haussler, D. & Warmuth, M. K. (1987), "Occam's razor," *Information Processing Letters* **24**(6), 377–380.

Blumer, A., Ehrenfeucht, A., Haussler, D. & Warmuth, M. K. (1989), "Learnability and the Vapnik-Chervonenkis dimension," *Journal of the Association for Computing Machinery* **36**(4), 929–965.

Borwein, J. & Lewis, A. (2006), *Convex analysis and nonlinear optimization*, Springer.

Boser, B. E., Guyon, I. M. & Vapnik, V. N. (1992), "A training algorithm for optimal margin classifiers," in COLT, pp. 144–152.

Bottou, L. & Bousquet, O. (2008), "The tradeoffs of large scale learning," in NIPS, pp. 161–168.

Boucheron, S., Bousquet, O. & Lugosi, G. (2005), "Theory of classification: A survey of recent advances," *ESAIM: Probability and Statistics* **9**, 323–375.

Bousquet, O. (2002), Concentration Inequalities and Empirical Processes Theory Applied to the Analysis of Learning Algorithms, PhD thesis, Ecole Polytechnique.

Bousquet, O. & Elisseeff, A. (2002), "Stability and generalization," *Journal of Machine Learning Research* **2**, 499–526.

Boyd, S. & Vandenberghe, L. (2004), *Convex optimization*, Cambridge University Press.

Breiman, L. (1996), Bias, variance, and arcing classifiers, Technical Report 460, Statistics Department, University of California at Berkeley.

Breiman, L. (2001), "Random forests," *Machine Learning* **45**(1), 5–32.

Breiman, L., Friedman, J. H., Olshen, R. A. & Stone, C. J. (1984), *Classification and regression trees*, Wadsworth & Brooks.

Candès, E. (2008), "The restricted isometry property and its implications for compressed sensing," *Comptes Rendus Mathematique* **346**(9), 589–592.

Candes, E. J. (2006), "Compressive sampling," in *Proc. of the int. congress of math.*, Madrid, Spain.

Candes, E. & Tao, T. (2005), "Decoding by linear programming," *IEEE Trans. on Information Theory* **51**, 4203–4215.

Cesa-Bianchi, N. & Lugosi, G. (2006), *Prediction, learning, and games*, Cambridge University Press.

Chang, H. S., Weiss, Y. & Freeman, W. T. (2009), "Informative sensing," *arXiv preprint arXiv:0901.4275*.

Chapelle, O., Le, Q. & Smola, A. (2007), "Large margin optimization of ranking measures," in *NIPS workshop: Machine learning for Web search* (Machine Learning).

Collins, M. (2000), "Discriminative reranking for natural language parsing," in Machine Learning.

Collins, M. (2002), "Discriminative training methods for hidden Markov models: Theory and experiments with perceptron algorithms," in *Conference on Empirical Methods in Natural Language Processing*.

Collobert, R. & Weston, J. (2008), "A unified architecture for natural language processing: deep neural networks with multitask learning," in International Conference on Machine Learning (ICML).

Cortes, C. & Vapnik, V. (1995), "Support-vector networks," *Machine Learning* **20**(3), 273–297.

Cover, T. (1965), "Behavior of sequential predictors of binary sequences," *Trans. 4th Prague conf. information theory statistical decision functions, random processes*, pp. 263–272.

Cover, T. & Hart, P. (1967), "Nearest neighbor pattern classification," *Information Theory, IEEE Transactions on* **13**(1), 21–27.

Crammer, K. & Singer, Y. (2001), "On the algorithmic implementation of multiclass kernel-based vector machines," *Journal of Machine Learning Research* **2**, 265–292.

Cristianini, N. & Shawe-Taylor, J. (2000), *An introduction to support vector machines*, Cambridge University Press.

Daniely, A., Sabato, S., Ben-David, S. & Shalev-Shwartz, S. (2011), "Multiclass learnability and the erm principle," in COLT.

Daniely, A., Sabato, S. & Shwartz, S. S. (2012), "Multiclass learning approaches: A theoretical comparison with implications," in NIPS.

Davis, G., Mallat, S. & Avellaneda, M. (1997), "Greedy adaptive approximation," *Journal of Constructive Approximation* **13**, 57–98.

Devroye, L. & Györfi, L. (1985), *Nonparametric density estimation: The L BI S view*, Wiley.

Devroye, L., Györfi, L. & Lugosi, G. (1996), *A probabilistic theory of pattern recognition*, Springer.

Dietterich, T. G. & Bakiri, G. (1995), "Solving multiclass learning problems via error-correcting output codes," *Journal of Artificial Intelligence Research* **2**, 263–286.

Donoho, D. L. (2006), "Compressed sensing," *Information Theory, IEEE Transactions* **52**(4), 1289–1306.

Dudley, R., Gine, E. & Zinn, J. (1991), "Uniform and universal glivenko-cantelli classes," *Journal of Theoretical Probability* **4**(3), 485–510.

Dudley, R. M. (1987), "Universal Donsker classes and metric entropy," *Annals of Probability* **15**(4), 1306–1326.

Fisher, R. A. (1922), "On the mathematical foundations of theoretical statistics," *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character* **222**, 309–368.

Floyd, S. (1989), "Space-bounded learning and the Vapnik-Chervonenkis dimension," in COLT, pp. 349–364.

Floyd, S. & Warmuth, M. (1995), "Sample compression, learnability, and the Vapnik-Chervonenkis dimension," *Machine Learning* **21**(3), 269–304.

Frank, M. & Wolfe, P. (1956), "An algorithm for quadratic programming," *Naval Res. Logist. Quart.* **3**, 95–110.

Freund, Y. & Schapire, R. (1995), "A decision-theoretic generalization of on-line learning and an application to boosting," in European Conference on Computational Learning Theory (EuroCOLT), Springer-Verlag, pp. 23–37.

Freund, Y. & Schapire, R. E. (1999), "Large margin classification using the perceptron algorithm," *Machine Learning* **37**(3), 277–296.

Garcia, J. & Koelling, R. (1996), "Relation of cue to consequence in avoidance learning," *Foundations of animal behavior: classic papers with commentaries* **4**, 374.

Gentile, C. (2003), "The robustness of the p-norm algorithms," *Machine Learning* **53**(3), 265–299.

Georghiades, A., Belhumeur, P. & Kriegman, D. (2001), "From few to many: Illumination cone models for face recognition under variable lighting and pose," *IEEE Trans. Pattern Anal. Mach. Intelligence* **23**(6), 643–660.

Gordon, G. (1999), "Regret bounds for prediction problems," in Conference on Learning Theory (COLT).

Gottlieb, L.-A., Kontorovich, L. & Krauthgamer, R. (2010), "Efficient classification for metric data," in *23rd conference on learning theory*, pp. 433–440.

Guyon, I. & Elisseeff, A. (2003), "An introduction to variable and feature selection," *Journal of Machine Learning Research, Special Issue on Variable and Feature Selection* **3**, 1157–1182.

Hadamard, J. (1902), "Sur les problèmes aux dérivées partielles et leur signification physique," *Princeton University Bulletin* **13**, 49–52.

Hastie, T., Tibshirani, R. & Friedman, J. (2001), *The elements of statistical learning*, Springer.

Haussler, D. (1992), "Decision theoretic generalizations of the PAC model for neural net and other learning applications," *Information and Computation* **100**(1), 78–150.

Haussler, D. & Long, P. M. (1995), "A generalization of sauer's lemma," *Journal of Combinatorial Theory, Series A* **71**(2), 219–240.

Hazan, E., Agarwal, A. & Kale, S. (2007), "Logarithmic regret algorithms for online convex optimization," *Machine Learning* **69**(2–3), 169–192.

Hinton, G. E., Osindero, S. & Teh, Y.-W. (2006), "A fast learning algorithm for deep belief nets," *Neural Computation* **18**(7), 1527–1554.

Hiriart-Urruty, J.-B. & Lemaréchal, C. (1993), *Convex analysis and minimization algorithms*, Springer.

Hsu, C.-W., Chang, C.-C., & Lin, C.-J. (2003), "A practical guide to support vector classification."

Hyafil, L. & Rivest, R. L. (1976), "Constructing optimal binary decision trees is NP-complete," *Information Processing Letters* **5**(1), 15–17.

Joachims, T. (2005), "A support vector method for multivariate performance measures," in *Proceedings of the international conference on machine learning* (ICML).

Kakade, S., Sridharan, K. & Tewari, A. (2008), "On the complexity of linear prediction: Risk bounds, margin bounds, and regularization," in NIPS.

Karp, R. M. (1972), *Reducibility among combinatorial problems*, Springer.

Kearns, M. & Mansour, Y. (1996), "On the boosting ability of top-down decision tree learning algorithms," in ACM Symposium on the Theory of Computing (STOC).

Kearns, M. & Ron, D. (1999), "Algorithmic stability and sanity-check bounds for leave-one-out cross-validation," *Neural Computation* **11**(6), 1427–1453.

Kearns, M. & Valiant, L. G. (1988), "Learning Boolean formulae or finite automata is as hard as factoring, Technical Report TR-14-88, Harvard University, Aiken Computation Laboratory.

Kearns, M. & Vazirani, U. (1994), *An Introduction to Computational Learning Theory*, MIT Press.

Kearns, M. J., Schapire, R. E. & Sellie, L. M. (1994), "Toward efficient agnostic learning," *Machine Learning* **17**, 115–141.

Kleinberg, J. (2003), "An impossibility theorem for clustering," NIPS, pp. 463–470.

Klivans, A. R. & Sherstov, A. A. (2006), Cryptographic hardness for learning intersections of halfspaces, in FOCS.

Koller, D. & Friedman, N. (2009), *Probabilistic graphical models: Principles and techniques*, MIT Press.

Koltchinskii, V. & Panchenko, D. (2000), "Rademacher processes and bounding the risk of function learning," in *High Dimensional Probability II*, Springer, pp. 443–457.

Kuhn, H. W. (1955), "The hungarian method for the assignment problem," *Naval Research Logistics Quarterly* **2**(1–2), 83–97.

Kutin, S. & Niyogi, P. (2002), "Almost-everywhere algorithmic stability and generalization error," in *Proceedings of the 18th conference in uncertainty in artificial intelligence*, pp. 275–282.

Lafferty, J., McCallum, A. & Pereira, F. (2001), "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *International conference on machine learning*, pp. 282–289.

Langford, J. (2006), "Tutorial on practical prediction theory for classification," *Journal of machine learning research* **6**(1), 273.

Langford, J. & Shawe-Taylor, J. (2003), "PAC-Bayes & margins," in NIPS, pp. 423–430.

Le, Q. V., Ranzato, M.-A., Monga, R., Devin, M., Corrado, G., Chen, K., Dean, J. & Ng, A. Y. (2012), "Building high-level features using large scale unsupervised learning," in ICML.

Le Cun, L. (2004), "Large scale online learning," in *Advances in neural information processing systems 16: Proceedings of the 2003 conference*, Vol. 16, MIT Press, p. 217.

LeCun, Y. & Bengio, Y. (1995), "Convolutional networks for images, speech, and time series," in *The handbook of brain theory and neural networks*, The MIT Press.

Lee, H., Grosse, R., Ranganath, R. & Ng, A. (2009), "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," in ICML.

Littlestone, N. (1988), "Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm," *Machine Learning* **2**, 285–318.

Littlestone, N. & Warmuth, M. (1986), Relating data compression and learnability, Unpublished manuscript.

Littlestone, N. & Warmuth, M. K. (1994), "The weighted majority algorithm," *Information and Computation* **108**, 212–261.

Livni, R., Shalev-Shwartz, S. & Shamir, O. (2013), "A provably efficient algorithm for training deep networks," *arXiv preprint arXiv:1304.7045* .

Livni, R. & Simon, P. (2013), "Honest compressions and their application to compression schemes," in COLT.

MacKay, D. J. (2003), *Information theory, inference and learning algorithms*, Cambridge University Press.

Mallat, S. & Zhang, Z. (1993), "Matching pursuits with time-frequency dictionaries," *IEEE Transactions on Signal Processing* **41**, 3397–3415.

McAllester, D. A. (1998), "Some PAC-Bayesian theorems," in COLT.

McAllester, D. A. (1999), "PAC-Bayesian model averaging," in COLT, pp. 164–170.

McAllester, D. A. (2003), "Simplified PAC-Bayesian margin bounds," in COLT, pp. 203–215.

Minsky, M. & Papert, S. (1969), *Perceptrons: An introduction to computational geometry*, The MIT Press.

Mukherjee, S., Niyogi, P., Poggio, T. & Rifkin, R. (2006), "Learning theory: stability is sufficient for generalization and necessary and sufficient for consistency of empirical risk minimization," *Advances in Computational Mathematics* **25**(1–3), 161–193.

Murata, N. (1998), "A statistical study of on-line learning," *Online Learning and Neural Networks, Cambridge University Press*.

Murphy, K. P. (2012), *Machine learning: a probabilistic perspective*, The MIT Press.

Natarajan, B. (1995), "Sparse approximate solutions to linear systems," *SIAM J. Computing* **25**(2), 227–234.

Natarajan, B. K. (1989), "On learning sets and functions," *Mach. Learn.* **4**, 67–97.

Nemirovski, A., Juditsky, A., Lan, G. & Shapiro, A. (2009), "Robust stochastic approximation approach to stochastic programming," *SIAM Journal on Optimization* **19**(4), 1574–1609.

Nemirovski, A. & Yudin, D. (1978), *Problem complexity and method efficiency in optimization*, Nauka, Moscow.

Nesterov, Y. (2005), Primal-dual subgradient methods for convex problems, Technical report, Center for Operations Research and Econometrics (CORE), Catholic University of Louvain (UCL).

Nesterov, Y. & Nesterov, I. (2004), *Introductory lectures on convex optimization: A basic course*, Vol. 87, Springer, Netherlands.

Novikoff, A. B. J. (1962), "On convergence proofs on perceptrons," in *Proceedings of the symposium on the mathematical theory of automata*, Vol. XII, pp. 615–622.

Parberry, I. (1994), *Circuit complexity and neural networks*, The MIT press.

Pearson, K. (1901), "On lines and planes of closest fit to systems of points in space," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* **2**(11), 559–572.

Phillips, D. L. (1962), "A technique for the numerical solution of certain integral equations of the first kind," *Journal of the ACM* **9**(1), 84–97.

Pisier, G. (1980–1981), "Remarques sur un résultat non publié de B. maurey."

Pitt, L. & Valiant, L. (1988), "Computational limitations on learning from examples," *Journal of the Association for Computing Machinery* **35**(4), 965–984.

Poon, H. & Domingos, P. (2011), "Sum-product networks: A new deep architecture," in Conference on Uncertainty in Artificial Intelligence (UAI).

Quinlan, J. R. (1986), "Induction of decision trees," *Machine Learning* **1**, 81–106.

Quinlan, J. R. (1993), *C4.5: Programs for machine learning*, Morgan Kaufmann.

Rabiner, L. & Juang, B. (1986), "An introduction to hidden markov models," *IEEE ASSP Magazine* **3**(1), 4–16.

Rakhlin, A., Shamir, O. & Sridharan, K. (2012), "Making gradient descent optimal for strongly convex stochastic optimization," in ICML.

Rakhlin, A., Sridharan, K. & Tewari, A. (2010), "Online learning: Random averages, combinatorial parameters, and learnability," in NIPS.

Rakhlin, S., Mukherjee, S. & Poggio, T. (2005), "Stability results in learning theory," *Analysis and Applications* **3**(4), 397–419.

Ranzato, M., Huang, F., Boureau, Y. & Lecun, Y. (2007), "Unsupervised learning of invariant feature hierarchies with applications to object recognition," in Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on, IEEE, pp. 1–8.

Rissanen, J. (1978), "Modeling by shortest data description," *Automatica* **14**, 465–471.

Rissanen, J. (1983), "A universal prior for integers and estimation by minimum description length," *The Annals of Statistics* **11**(2), 416–431.

Robbins, H. & Monro, S. (1951), "A stochastic approximation method," *The Annals of Mathematical Statistics*, pp. 400–407.

Rogers, W. & Wagner, T. (1978), "A finite sample distribution-free performance bound for local discrimination rules," *The Annals of Statistics* **6**(3), 506–514.

Rokach, L. (2007), *Data mining with decision trees: Theory and applications*, Vol. 69, World Scientific.

Rosenblatt, F. (1958), "The perceptron: A probabilistic model for information storage and organization in the brain," *Psychological Review* **65**, 386–407. (Reprinted in *Neurocomputing*, MIT Press, 1988).

Rumelhart, D. E., Hinton, G. E. & Williams, R. J. (1986), "Learning internal representations by error propagation," in D. E. Rumelhart & J. L. McClelland, eds, *Parallel distributed processing – explorations in the microstructure of cognition*, MIT Press, chapter 8, pp. 318–362.

Sankaran, J. K. (1993), "A note on resolving infeasibility in linear programs by constraint relaxation," *Operations Research Letters* **13**(1), 19–20.

Sauer, N. (1972), "On the density of families of sets," *Journal of Combinatorial Theory Series A* **13**, 145–147.

Schapire, R. (1990), "The strength of weak learnability," *Machine Learning* **5**(2), 197–227.

Schapire, R. E. & Freund, Y. (2012), *Boosting: Foundations and algorithms*, MIT Press.

Schölkopf, B. & Smola, A. J. (2002), *Learning with kernels: Support vector machines, regularization, optimization and beyond*, MIT Press.

Schölkopf, B., Herbrich, R. & Smola, A. (2001), "A generalized representer theorem," in *Computational learning theory*, pp. 416–426.

Schölkopf, B., Herbrich, R., Smola, A. & Williamson, R. (2000), "A generalized representer theorem," in *NeuroCOLT*.

Schölkopf, B., Smola, A. & Müller, K.-R. (1998), 'Nonlinear component analysis as a kernel eigenvalue problem', *Neural computation* **10**(5), 1299–1319.

Seeger, M. (2003), "Pac-bayesian generalisation error bounds for gaussian process classification," *The Journal of Machine Learning Research* **3**, 233–269.

Shakhnarovich, G., Darrell, T. & Indyk, P. (2006), *Nearest-neighbor methods in learning and vision: Theory and practice*, MIT Press.

Shalev-Shwartz, S. (2007), Online Learning: Theory, Algorithms, and Applications, PhD thesis, The Hebrew University.

Shalev-Shwartz, S. (2011), "Online learning and online convex optimization," *Foundations and Trends® in Machine Learning* **4**(2), 107–194.

Shalev-Shwartz, S., Shamir, O., Srebro, N. & Sridharan, K. (2010), "Learnability, stability and uniform convergence," *The Journal of Machine Learning Research* **9999**, 2635–2670.

Shalev-Shwartz, S., Shamir, O. & Sridharan, K. (2010), "Learning kernel-based halfspaces with the zero-one loss," in COLT.

Shalev-Shwartz, S., Shamir, O., Sridharan, K. & Srebro, N. (2009), "Stochastic convex optimization," in COLT.

Shalev-Shwartz, S. & Singer, Y. (2008), "On the equivalence of weak learnability and linear separability: New relaxations and efficient boosting algorithms," in *Proceedings of the nineteenth annual conference on computational learning theory*.

Shalev-Shwartz, S., Singer, Y. & Srebro, N. (2007), "Pegasos: Primal Estimated sub-GrAdient SOlver for SVM," in *International conference on machine learning*, pp. 807–814.

Shalev-Shwartz, S. & Srebro, N. (2008), "SVM optimization: Inverse dependence on training set size," in *International conference on machine learning* ICML, pp. 928–935.

Shalev-Shwartz, S., Zhang, T. & Srebro, N. (2010), "Trading accuracy for sparsity in optimization problems with sparsity constraints," *Siam Journal on Optimization* **20**, 2807–2832.

Shamir, O. & Zhang, T. (2013), "Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes," in ICML.

Shapiro, A., Dentcheva, D. & Ruszczyński, A. (2009), *Lectures on stochastic programming: modeling and theory*, Vol. 9, Society for Industrial and Applied Mathematics.

Shelah, S. (1972), "A combinatorial problem: stability and order for models and theories in infinitary languages," *Pac. J. Math* **4**, 247–261.

Sipser, M. (2006), *Introduction to the Theory of Computation*, Thomson Course Technology.

Slud, E. V. (1977), "Distribution inequalities for the binomial law," *The Annals of Probability* **5**(3), 404–412.

Steinwart, I. & Christmann, A. (2008), *Support vector machines*, Springerverlag, New York.

Stone, C. (1977), "Consistent nonparametric regression," *The Annals of Statistics* **5**(4), 595–620.

Taskar, B., Guestrin, C. & Koller, D. (2003), "Max-margin markov networks," in NIPS.

Tibshirani, R. (1996), "Regression shrinkage and selection via the lasso," *J. Royal. Statist. Soc B.* **58**(1), 267–288.

Tikhonov, A. N. (1943), "On the stability of inverse problems," *Dolk. Akad. Nauk SSSR* **39**(5), 195–198.

Tishby, N., Pereira, F. & Bialek, W. (1999), "The information bottleneck method," in *The 37'th Allerton conference on communication, control, and computing*.

Tsochantaridis, I., Hofmann, T., Joachims, T. & Altun, Y. (2004), "Support vector machine learning for interdependent and structured output spaces," in *Proceedings of the twenty-first international conference on machine learning*.

Valiant, L. G. (1984), "A theory of the learnable," *Communications of the ACM* **27**(11), 1134–1142.

Vapnik, V. (1992), "Principles of risk minimization for learning theory," in J. E. Moody, S. J. Hanson & R. P. Lippmann, eds., *Advances in Neural Information Processing Systems 4*, Morgan Kaufmann, pp. 831–838.

Vapnik, V. (1995), *The Nature of Statistical Learning Theory*, Springer.

Vapnik, V. N. (1982), *Estimation of Dependences Based on Empirical Data*, Springer-Verlag.

Vapnik, V. N. (1998), *Statistical Learning Theory*, Wiley.

Vapnik, V. N. & Chervonenkis, A. Y. (1971), "On the uniform convergence of relative frequencies of events to their probabilities," *Theory of Probability and Its Applications* **XVI**(2), 264–280.

Vapnik, V. N. & Chervonenkis, A. Y. (1974), *Theory of pattern recognition*, Nauka, Moscow (In Russian).

Von Luxburg, U. (2007), "A tutorial on spectral clustering," *Statistics and Computing* **17**(4), 395–416.

von Neumann, J. (1928), "Zur theorie der gesellschaftsspiele (on the theory of parlor games)," *Math. Ann.* **100**, 295—320.

Von Neumann, J. (1953), "A certain zero-sum two-person game equivalent to the optimal assignment problem," *Contributions to the Theory of Games* **2**, 5–12.

Vovk, V. G. (1990), "Aggregating strategies," in COLT, pp. 371–383.

Warmuth, M., Glocer, K. & Vishwanathan, S. (2008), "Entropy regularized lpboost," in *Algorithmic Learning Theory* (ALT).

Warmuth, M., Liao, J. & Ratsch, G. (2006), "Totally corrective boosting algorithms that maximize the margin," in *Proceedings of the 23rd international conference on machine learning*.

Weston, J., Chapelle, O., Vapnik, V., Elisseeff, A. & Schölkopf, B. (2002), "Kernel dependency estimation," in Advances in neural information processing systems, pp. 873–880.

Weston, J. & Watkins, C. (1999), "Support vector machines for multi-class pattern recognition," in *Proceedings of the seventh european symposium on artificial neural networks*.

Wolpert, D. H. & Macready, W. G. (1997), "No free lunch theorems for optimization," *Evolutionary Computation, IEEE Transactions on* **1**(1), 67–82.

Zhang, T. (2004), "Solving large scale linear prediction problems using stochastic gradient descent algorithms," in *Proceedings of the twenty-first international conference on machine learning*.

Zhao, P. & Yu, B. (2006), "On model selection consistency of Lasso," *Journal of Machine Learning Research* **7**, 2541–2567.

Zinkevich, M. (2003), "Online convex programming and generalized infinitesimal gradient ascent," in *International conference on machine learning*.

索　引

索引中的页码为英文原书页码，与书中页边标注的页码一致。

3-term DNF(3 项析取范式)，79
F_1-score(F_1 得分)，207
L_0 norm(L_0 范数)，149，286，315，355

A

accuracy(精确度)，18，22
activation function(激活函数)，228
AdaBoost(AdaBoost 算法)，101，**105**，314
all-pairs(一对一)，191，353
approximation error(逼近误差)，37，40
auto-encoders(自编码器)，315

B

backpropagation(反向传播)，237
backward elimination(反向终止算法)，314
bag of words(词袋模型)，173
base hypothesis(基础假设类)，**108**
Bayes optimal(贝叶斯最优)，24，30，221
Bayes rule(贝叶斯准则)，306
Bayesian reasoning(贝叶斯推理)，305
Bennet's inequality(贝内特不等式)，376
Bernstein's inequality(伯恩斯坦不等式)，376
bias(偏置)，16，37，42
bias-complexity tradeoff(偏差-复杂度权衡)，41
Boolean conjunctions(布尔合取式)，29，54，78
boosting (boosting 算法)，101
boosting the confidence(置信度 boosting)，112
boundedness(有界性)，133

C

C4.5(C4.5 算法)，215
CART(CART 算法)，216
chaining(链式反应)，338
Chebyshev's inequality(切比雪夫不等式)，375
Chernoff bounds(切尔诺夫界)，373
class-sensitive feature mapping(类敏感的特征映射)，163
classifier(分类器)，14
clustering(聚类)，264
　spectral(谱聚类)，271
compressed sensing(压缩感知)，285
compression bounds(压缩界)，359
compression scheme(压缩方法)，360
computational complexity(计算复杂度)，73
confidence(置信度)，18，22
consistency(一致性)，65
Consistent(一致性算法)，247
contraction lemma(压缩引理)，321
convex(凸性)，124
　function(凸函数)，125
　set(凸集)，124
　strongly convex(强凸性)，140，160
convex Lipschitz bounded learning(凸利普希茨有界学习)，133
convex-smooth-bounded learning(凸光滑有界学习)，133
covering numbers(覆盖数)，337
curse of dimensionality(维数灾难)，224

D

decision stumps(决策桩)，103，104
decision trees(决策树)212
dendrogram(系统树图)266，267
dictionary learning(字典学习)，319
differential set(微分集)，184
dimensionality reduction(维数约简)，278
discretization trick(离散化技巧)，34
discriminative(判别式的)，295
distribution free(分布无关)，295
domain(域)，13
domain of examples(样本域)，26
doubly stochastic matrix(双随机矩阵)，205
duality(对偶性)，175
　strong duality(强对偶性)，175

weak duality(弱对偶性)，178
Dudley classes(Dudley 类)，56

E

efficient computable(可高效计算)，73
EM(期望最大化算法)，301
Empirical Risk Minimization，see ERM(经验风险最小化，参见 ERM)
empirical error(经验误差)，15
empirical risk(经验风险)，15，27
entropy(熵)，298
  relative entropy(相关熵)，298
epigraph(上境图)，125
ERM(经验风险最小化)，15
error decomposition(误差分解)，40，135
estimation error(估计误差)，37，40
Expectation-Maximization，see EM(期望最大化，参见 EM)

F

face recognition，see Viola-Jones(人脸识别，参见 Viola-Jones)
feasible(可行的)，73
feature(特征)，75
feature learning(特征学习)，319
feature normalization(特征归一化)，316
feature selection(特征选择)，309，310
feature space(特征空间)，179
feature transformation(特征变换)，319
filters(滤波器)，310
forward greedy selection(前向贪婪选择)，312
frequentist(频率学派)，305

G

gain 增益 215
GD，see gradient descent(GD，参见梯度下降)
generalization error(泛化误差)，14
generative models(生成模型)，295
Gini index(基尼系数)，215
Glivenko-Cantelli(Glivenko-Cantelli 类)，35
gradient(梯度)，125
gradient descent(梯度下降)，151
Gram matrix(Gram 矩阵)，183
growth function(生长函数)，45

H

halfspace(半空间)，59
  homogeneous(齐次的)，91，170
  nonseparable(不可分的)，90
  separable(可分的)，90
Halving(二分算法)，247
hidden layers(隐含层)，230
Hilbert space(希尔伯特空间)，181
Hoeffding's inequality(Hoeffding 不等式)，33，375
holdout(留出)，115
hypothesis(假设)，14
hypothesis class(假设类)，16

I

i.i.d.(独立同分布)，18
ID3(ID3 算法)，214
improper，see representation independent(不适当的，参见独立表示)
inductive bias，see bias(归纳偏置，参见偏置)
information bottleneck(信息瓶颈)，272
information gain(信息增益)，215
instance(实例)，12
  instance space(实例空间)，13
integral image(积分图像)，113

J

Johnson-Lindenstrauss lemma (Johnson-Lindenstrauss 引理)，284

K

k-means(k 均值算法)，268，270
  soft k means(软 k 均值算法)，304
k-median(k 中位数算法)，269
k-medoids(k 中心点算法)，269
Kendall tau(Kendall tau 损失)，221
kernel PCA(核 PCA 算法)，291
kernels(核)，179
  Gaussian kernel(高斯核)，184
  kernel trick(核技巧)，181
  polynomial kernel(多项式核)，183
  RBF kernel(径向基函数核)，184

L

label(标签)，13

Lasso(lasso 算法)，318，335
generalization bounds(泛化界)，335
latent variables(隐变量)，301
LDA(线性判别分析)，300
Ldim(LittleStone 维)，248，249
learning curves(学习曲线)，122
least squares(最小平方)，90
likelihood ratio(似然函数比)，201
linear discriminant analysis，see LDA(线性判别分析，参见 LDA)
linear predictor(线性分类器)，89
homogeneous(齐次线性分类器)，90
linear programming(线性规划)，91
linear regression(线性回归)，84
linkage(链接)，296
Lipschitzness(利普希茨性)，128，142，157
subgradient(子梯度)，165
Littlestone dimension，see Ldim(Littlestone 维，参见 Ldim)
local minimum(局部极小)，126
Logistic regression(逻辑斯蒂回归)，97
loss(损失)，15
loss function(损失函数)，26
0-1 loss 0-1(损失函数)，27，134
absolute value loss(绝对值损失函数)，85，89，132
convex loss(凸损失函数)，131
generalized hinge loss(泛化 hinge 损失)，195
hinge loss(hinge 损失)，134
Lipschitz loss(利普希茨损失函数)，133
log-loss(对数损失函数)，298
logistic loss(逻辑斯蒂损失函数)，98
ramp loss(斜坡损失函数)，174
smooth loss(光滑损失函数)，135
square loss(平方损失函数)，27
surrogate loss(代理损失函数)，134，359

M

margin(间隔)，158
Markov's inequality(马尔可夫不等式)，372
Massart lemma(马萨特引理)，330
max linkage(最大链接)，297
maximum a posterior(最大化后验)，307
maximum likelihood(极大似然法)，295
McDiarmid's inequality(麦克迪尔米德不等式)，328
MDL(最小描述长度)，63，65，213
measure concentration(测度集中度)，32，372
Minimum Description Length，see MDL(最小描述长度，参见 MDL)
mistake bound(误差界)，246
mixture of Gaussians(高斯混合模型)，301
model selection(模型选择)，114，117
multiclass(多分类)，25，190，351
cost-sensitive(损失敏感的)，194
linear predictors(线性分类器)，193，354
multivector(多向量)，193，355
Perceptron(感知器)，211
reduction(约简)，196，354
SGD(随机梯度下降)，198
SVM(支持向量机)，197
multivariate performance measures(多变量性能度量)，206

N

Naive Bayes(朴素贝叶斯)，259
Natarajan dimension(纳塔拉詹维)，351
NDCG(归一化折扣累积增益)，202
Nearest Neighbor(最近邻)，219
k-NN(k 近邻)，220
neural networks(神经网络)，228
feedforward networks(前馈神经网络)，229
layered networks(层次网络)，229
SGD(随机梯度下降)，236
No-Free-Lunch("没有免费的午餐")，37
nonuniform learning(非一致学习)，59
Normalized Discounted Cumulative Gain，see NDCG(归一化折扣累积增益，参见 NDCG)

O

Occam's razor(奥卡姆剃刀)，65
OMP(正交匹配追踪)，312
one-versus-all(一对多)，191，353
one versus rest，see one-versus-all(一对剩余，参见一对多)
online convex optimization(在线凸优化)，257
online gradient descent(在线梯度下降)，257
online learning(在线学习)，245
optimization error(优化误差)，135
oracle inequality(神谕不等式)，145

orthogonal matching pursuit, see OMP(正交匹配追踪，参见 OMP)
overfitting(过拟合)，15，41，121

P

PAC(概率近似正确)，22
agnostic PAC(不可知 PAC)，23，25
agnostic PAC for general loss(广义损失的不可知 PAC)，27
PAC-Bayes(PAC 贝叶斯)，364
parametric density estimation(参数密度估计)，299
PCA(主成分分析)，279
Pearson's correlation coefficient(皮尔森相关系数)，311
Perceptron(感知器)，92
kernelizedPerceptron(核化感知器)，188
multiclass(多类别)，211
online(在线)，259
permutation matrix(置换矩阵)，205
polynomial regression(多项式回归)，96
precision(精确度)，206
predictor(预测器)，14
prefix free language(无前缀语言)，64
Principal Component Analysis, see PCA(主成分分析，参见 PCA)
prior knowledge(先验知识)，39
Probably Approximately Correct, see PAC(概率近似正确，参见 PAC)
projection(投影)，159
projection lemma(投影引理)，159
proper(完全)，28
pruning(剪枝)，216

R

Rademacher complexity(拉德马赫复杂度)，325
random forests(随机森林)，217
random projections(随机投影)，283
ranking(排序)，201
bipartite(二分)，206
realizability(可实现性)，17
recall(召回)，206
regression(回归)，26，94，138
regularization(正则化)，137
Tikhonov(Tikhonov 正则化)，138，140
regularized loss minimization, see RLM(最小化正则损失，参见 RLM)
representation independent(独立表示)，28，80
representative sample(代表性样本)，31，325
representer theorem(表示定理)，182
ridge regression(岭回归)，138
kernel ridge regression(核岭回归)，188
RIP(有限等距约束)，286
risk(风险)，14，24，26
RLM(最小化正则损失)，137，164

S

sample complexity(样本复杂度)，22
Sauer's lemma(Sauer 引理)，45
self-boundedness(自有界性)，133
sensitivity(敏感度)，206
SGD(奇异值分解)，156
shattering(打散)，45，352
single linkage(单连接)，267
Singular Value Decomposition, see SVD(奇异值分解，参见 SVD)
Sluf's inequality(Sluf 不等式)，378
smoothness(光滑性)，129，143，163
SOA(标准优化算法)，250
sparsity-inducing norms(稀疏诱导范数)，315
specificity(具体性)，206
spectral clustering(谱聚类)，271
SRM(最小化结构风险)，60，115
stability(稳定性)，139
Stochastic Gradient Descent, see SGD(随机梯度下降，参见 SGD)
strong learning(强学习)，102
Structural Risk Minimization, see SRM(最小化结构风险，参见 SRM)
structured output prediction(结构输出预测)，198
subgradient(次梯度)，154
Support Vector Machines, see SVM(支持向量机，参见 SVM)
SVD(奇异值分解)，381
SVM(支持向量机)，167，333
duality(对偶性)，175
generalization bounds(泛化边界)，172，333
hard-SVM(硬 SVM)，168，169
homogeneous(齐次情况)，170
kernel trick(核函数技巧)，181
soft-SVM(软 SVM)，171

support vectors(支持向量)，175

T

target set(目标集合)，26
term frequency(词项频率)，194
TF-IDF(词频逆文档频率)，194
training error(训练误差)，15
training set(训练集)，13
true error(真实误差)，14，24

U

underfitting(欠拟合)，41，121
uniform convergence(一致收敛性)，31，32
union bound(联合界)，19
unsupervised learning(无监督学习)，265

V

validation(验证)，114，115
cross validation(交叉验证)，119
train-validation-test split(训练验证测试拆分)，120
Vapnik-Chervonenkis dimension，see(VC维)
dimension(维度)
VC dimension(VC维)，43，46
version space(可行域)，247
Viola-Jones(Viola-Jones 基假设)，118

W

weak learning(弱学习)，101，102
Weighted Majority(加权投票)，252

推荐阅读

神经网络与机器学习（原书第3版）

作者：Simon Haykin ISBN：978-7-111-32412-3 定价：79.00元

机器学习

作者：Tom Mitchell ISBN：978-7-111-10993-5 定价：35.00元

数据挖掘：实用机器学习工具与技术（原书第3版）

作者：Ian H. Witten 等 ISBN：978-7-111-40381-9 定价：79.00元

模式分类（原书第2版）

作者：Richard O. Duda 等 ISBN：978-7-111-12148-0 定价：59.00元

推荐阅读

机器学习与R语言实战

作者：丘祐玮（Yu-Wei Chiu） 译者：杨冬等
ISBN：978-7-111-63696-9 定价：69.00元

机器学习与R语言

作者：Brett Lantz 译者：李洪成等
ISBN：978-7-111-48157-6 定价：69.00元

机器学习导论（原书第3版）

作者：埃塞姆·阿培丁 译者：龙明
ISBN：978-7-111-52194-8 定价：79.00元

机器学习：实用案例解析

作者：Drew Conway 等 译者：张天江等
ISBN：978-7-111-41731-6 定价：69.00元

推荐阅读

大数据分析：数据驱动的企业绩效优化、过程管理和运营决策

作者：Thomas H. Davenport ISBN：978-7-111-49184-2 定价：39.00元

统计学习导论——基于R应用

作者：加雷斯·詹姆斯 等 ISBN：978-7-111-49771-4 定价：79.00元

数据科学：理论、方法与R语言实践

作者：比姆·朱姆尔马 ISBN：978-7-111-52926-2 定价：69.00元

商务智能：数据分析的管理视角（原书第3版）

作者：拉姆什·沙尔达 等 ISBN：978-7-111-49439-3 定价：69.00元