Understanding Machine Learning: From Theory to Algorithms

Shai Shalev-Shwartz and Shai Ben-David

UNDERSTANDING MACHINE LEARNING

FROM THEORY TO ALGORITHMS

理解机器学习

机器学习是计算机科学中增长最快的领域之一，具有深远的应用。本书的目的是以原理性的方式介绍机器学习及其提供的算法范式。本书提供了对机器学习基本思想及其数学推导的广泛理论阐述，这些推导将这些原理转化为实际算法。在介绍该领域的基础知识之后，本书涵盖了先前教科书未涉及的大量核心主题。这包括对学习计算复杂性和凸性与稳定性概念的讨论；包括随机梯度下降、神经网络和结构化输出学习等重要算法范式；以及如PAC-Bayes方法和基于压缩的界限等新兴理论概念。本书旨在为高级本科生或初级研究生课程设计，使机器学习的基本原理和算法对统计学、计算机科学、数学和工程领域的学者和非专业人士易于理解。

Shai Shalev-Shwartz是以色列希伯来大学计算机科学与工程学院的副教授。

Shai Ben-David 是加拿大滑铁卢大学计算机科学学院的教授。

理解机器学习

从理论到算法

Shai Shalev-Shwartz
The Hebrew University, Jerusalem

Shai Ben-David
University of Waterloo, Canada

CAMBRIDGE
UNIVERSITY PRESS

CAMBRIDGE
UNIVERSITY PRESS

32 美国大道，纽约，纽约州 10013-2473，美国 SA

剑桥大学出版社是剑桥大学的一部分。

它通过传播知识，在追求教育、学习和研究方面，进一步推动了大学使命，达到最高国际水平的卓越。

www.cambridge.org 关于此标题的信息：www.cambridge.org/9781107057135

首次发布 2014

美国印刷

A catalog record for this publication is available from the British Library

Library of Congress Cataloging in Publication Data

ISBN 978-1-107-05713-5 精装

Triple-S dedicates the book to triple-M

Preface

术语 $machine\ \ learning$ 指的是在数据中自动检测有意义模式的自动化检测。在过去几十年里，它已成为几乎任何需要从大型数据集中提取信息任务的常用工具。我们被基于机器学习技术的产品所包围：搜索引擎学习如何为我们提供最佳结果（同时投放盈利的广告），反垃圾邮件软件学习过滤我们的电子邮件，信用卡交易由学习如何检测欺诈的软件来保障。数码相机学习检测人脸，智能手机上的智能个人助理应用学习识别语音命令。汽车配备了使用机器学习算法构建的事故预防系统。机器学习在生物信息学、医学和天文学等科学应用中也得到了广泛应用。

所有这些应用的共同特征是，与计算机更传统的用途相比，在这些情况下，由于需要检测的模式的复杂性，人类程序员无法提供如何执行此类任务的明确、详细的规格说明。以智能生物为例，我们的许多技能是通过从我们的经验中$learning$获得或改进的（而不是遵循我们收到的明确指示）。机器学习工具关注的是赋予程序“学习”和适应的能力。

本书的第一个目标是提供一个严谨且易于理解的机器学习主要概念的介绍：什么是学习？机器如何学习？我们如何量化学习特定概念所需资源？学习总是可能的吗？我们能否知道学习过程是否成功或失败？

这本书的第二个目标是介绍几个关键的机器学习算法。我们选择介绍那些一方面在实践中成功应用，另一方面提供了广泛的学习技术谱系的算法。此外，我们还特别关注适合大规模学习（即“大数据”）的算法，因为近年来，我们的世界变得越来越“数字化”，可用于学习的数量急剧增加。因此，在许多应用中，数据量很大，计算时间是主要的瓶颈。因此，我们明确量化了学习给定概念所需的数据量和计算时间。

本书分为四部分。第一部分旨在对学习的根本问题给出一个初步的严谨答案。我们描述了Valiant的“可能近似正确”（PAC）学习模型的推广，这是对“什么是学习？”这一问题的第一个坚实回答。我们描述了经验风险最小化（ERM）、结构风险最小化（SRM）和最小描述长度（MDL）学习规则，展示了“机器如何学习”。我们使用ERM、SRM和MDL规则量化学习所需的数据量，并通过推导展示了学习可能失败的情况。

一个“没有免费午餐”定理。我们还讨论了学习所需的计算时间。本书的第二部分描述了各种学习算法。对于某些算法，我们首先提出一个更一般的原理，然后展示算法如何遵循该原理。虽然本书的前两部分侧重于PAC模型，但第三部分通过介绍更广泛的学习模型来扩展范围。最后，本书的最后部分致力于高级理论。

我们试图使这本书尽可能自成体系。然而，假设读者对概率、线性代数、分析和算法的基本概念感到舒适。本书的前三部分是为计算机科学、工程、数学或统计学的一年级研究生编写的。它也适合具有适当背景的本科生阅读。更高级的章节可供研究人员使用，以便获得更深入的理论理解。

Acknowledgements

本书基于希伯来大学Shai Shalev-Shwartz教授和滑铁卢大学Shai Ben-David教授讲授的机器学习课程。本书的第一稿源于2010-2013年间Shai Shalev-Shwartz在希伯来大学讲授的课程讲义。我们非常感谢Ohad Shamir的帮助，他在2010年担任该课程的助教，以及Alon Gonen的帮助，他在2011-2013年担任该课程的助教。Ohad和Alon准备了少量讲义和许多练习。Alon在整个书籍制作过程中给予我们帮助，我们对他的帮助感激不尽，他还准备了答案手册。

我们非常感谢Dana Rubinstein最宝贵的工作。Dana对稿件进行了科学校对和编辑，将其从基于讲座的章节转变为流畅且连贯的文本。

特别感谢Amit Daniely，他在仔细阅读书籍的高级部分的同时，还撰写了关于多类可学习性的高级章节。我们还要感谢耶路撒冷的一个读书俱乐部的成员，他们仔细阅读并建设性地批评了手稿的每一行。读书俱乐部的成员有：Maya Alroy，Yossi Arje-vani，Aharon Birnbaum，Alon Cohen，Alon Gonen，Roi Livni，Ofer Meshi，Dan Rosenbaum，Dana Rubinstein，Shahar Somin，Alon Vinnikov，和Yoav Wald。我们还要感谢Gal Elidan，Amir Globerson，Nika Haghtalab，Shie Mannor，Amnon Shashua，Nati Srebro，和Ruth Urner，因为他们有益的讨论。

Shai Shalev-Shwartz, 耶路撒冷，以色列
Shai Ben-David, 滑铁卢，加拿大

Contents

Preface *page* vii

1 Introduction 19
1.1 学习是什么？ 19 1.2 我们何时需要机器学习？ 21 1.3 学习类型 22 1.4 与其他领域的关联 24 1.5 如何阅读本书 25 1.5.1 基于本书的可能课程计划 26 1.6 符号 27

Part I Foundations 31

2 A Gentle Start 33
2.1 一个形式模型 – 统计学习框架 33 2.2 经验风险最小化 35 2.2.1 可能出现问题 – 过拟合 35 2.3 带有归纳偏好的经验风险最小化 36 2.3.1 有限假设类 37 2.4 练习 41

3 A Formal Learning Model 43
3.1 PAC学习 43 3.2 更通用的学习模型 44 3.2.1 放弃可实现性假设 - 不可知PAC学习 45 3.2.2 模型学习问题的范围 47

3.3 摘要 49 3.4 参考文献说明 50 3.5 练习 50

4 Learning via Uniform Convergence 54
4.1 一致收敛足以保证可学习性 54 4.2 有限类是无关的PAC可学习 55

4.3 摘要 58 4.4 参考文献说明 58 4.5 练习 58

5 The Bias-Complexity Tradeoff 60
5.1 无午餐定理 61 5.1.1 无午餐与先验知识 63 5.2 错误分解 64 5.3 概述 65 5.4 参考文献说明 66 5.5 练习 66

6 The VC-Dimension 67
6.1 无穷大小类可学习 67 6.2 VC维 68 6.3 例子 70 6.3.1 阈值函数 70 6.3.2 区间 71 6.3.3 与轴对齐的矩形 71 6.3.4 有限类 72 6.3.5 VC维与参数数量 72 6.4 PAC学习的基本定理 72 6.5 定理6.7的证明 73 6.5.1 Sauer引理与增长函数 73 6.5.2 小有效规模类的一致收敛 75 6.6 概述 78 6.7 参考文献注释 78 6.8 练习 78

7 Nonuniform Learnability 83
7.1 非均匀可学习性 83 7.1.1 非均匀可学习性的特征 84 7.2 结构风险最小化 85 7.3 最小描述长度与奥卡姆剃刀 89 7.3.1 奥卡姆剃刀 91 7.4 其他可学习性概念 – 一致性 92 7.5 讨论不同的可学习性概念 93 7.5.1 重新审视无免费午餐定理 95 7.6 总结 96 7.7 参考文献说明 97 7.8 练习 97

8 The Runtime of Learning 100
8.1 学习的计算复杂度 101

8.1.1 正式定义* 102 8.2 实现ERM规则 103 8.2.1 有限类 104 8.2.2 轴对齐矩形 105 8.2.3 布尔合取 106 8.2.4 学习3项DNF 107 8.3 高效可学习，但不是通过适当的ERM 107 8.4 学习的困难性* 108 8.5 概述 110 8.6 参考文献说明 110 8.7 练习 110

Part II From Theory to Algorithms 115

9 Linear Predictors 117
9.1 半空间 118 9.1.1 半空间类线性规划 119 9.1.2 半空间感知机 120 9.1.3 半空间的VC维数 122

9.2 线性回归 123 9.2.1 最小二乘法 124 9.2.2 多项式回归任务中的线性回归 125

9.3 逻辑回归 126 9.4 摘要 128 9.5 参考文献说明 128 9.6 练习 128

10 Boosting 130
10.1 弱可学习性 131 10.1.1 决策树桩的ERM高效实现 133 10.2 AdaBoost 134 10.3 基础假设的线性组合 137 10.3.1 $L(B, T)$的VC维数 139 10.4 AdaBoost在人脸识别中的应用 140 10.5 概述 141 10.6 参考文献说明 141 10.7 练习 142

11 Model Selection and Validation 144
11.1 使用SRM进行模型选择 145
11.2 验证 146 11.2.1 保留集 146 11.2.2 模型选择验证 147 11.2.3 模型选择曲线 148

11.2.4 k-倍交叉验证 149 11.2.5 训练-验证-测试划分 150 11.3 学习失败时该怎么办 151 11.4 概述 154 11.5 练习 154

12 Convex Learning Problems 156
12.1 凸性、Lipschitz性和光滑性 156 12.1.1 凸性 156 12.1.2 Lipschitz性 160 12.1.3 光滑性 162 12.2 凸学习问题 163 12.2.1 凸学习问题的可学习性 164 12.2.2 凸-Lipschitz/光滑有界学习问题 166 12.3 代理损失函数 167 12.4 概述 168 12.5 参考文献说明 169 12.6 练习 169

13 Regularization and Stability 171
13.1 正则化损失最小化 171 13.1.1 岭回归 172 13.2 稳定规则不会过拟合 173 13.3 Tikhonov 正则化作为稳定器 174 13.3.1 Lipschitz 损失 176 13.3.2 平滑且非负损失 177 13.4 控制拟合-稳定性权衡 178 13.5 概述 180 13.6 参考文献说明 180 13.7 练习 181

14 Stochastic Gradient Descent 184
14.1 梯度下降 185 14.1.1 凸-Lipschitz 函数的 GD 分析 186

14.2 子梯度 188 14.2.1 计算子梯度 189 14.2.2 Lipschitz 函数的子梯度 190 14.2.3 子梯度下降 190 14.3 随机梯度下降（SGD） 191 14.3.1 对于凸-Lipschitz-有界函数的 SGD 分析 191 14.4 变体 193 14.4.1 添加投影步骤 193 14.4.2 可变步长 194 14.4.3 其他平均技术 195

14.4.4 强凸函数* 195 1
4.5 使用SGD学习 196 14.5.1 SGD用于风险最小化 196 14.5.2 分析SGD在凸平滑学习问题中的应用 198 14.5.3 SGD用于正则化损失最小化 199

14.6 摘要 200 14.7 参考文献说明 200 14.8 练习 201

15 Support Vector Machines 202
15.1 边界和硬-SVM 202 15.1.1 同质情况 205 15.1.2 硬-SVM 的样本复杂度 205 15.2 软-SVM 和范数正则化 206 15.2.1 软-SVM 的样本复杂度 208 15.2.2 边界和基于范数的界限与维度 208 15.2.3 斜坡损失* 209 15.3 最优性条件和“支持向量”* 210 15.4 对偶* 211 15.5 使用 SGD 实现软-SVM 212 15.6 概述 213 15.7 参考文献说明 213 15.8 练习 214

16 Kernel Methods 215
16.1 将嵌入到特征空间中 215
16.2 内核技巧 217
16.2.1 作为表达先验知识的方式的核 221
16.2.2 描述核函数* 222
16.3 使用核实现软SVM 222
16.4 概述 224
16.5 参考文献说明 225
16.6 练习 225

17 Multiclass, Ranking, and Complex Prediction Problems 227
17.1 一对一与一对多 227 17.2 线性多类预测器 230 17.2.1 如何构建 Ψ 230 17.2.2 成本敏感分类 232 17.2.3 拓扑同伦方法 232 17.2.4 广义铰链损失 233 17.2.5 多类SVM和SGD 234 17.3 结构化输出预测 236 17.4 排序 238

17.4.1 排序的线性预测 240 17.5 二分排序和多变量性能度量 243 17.5.1 二分排序的线性预测 245 17.6 概述 247 17.7 参考文献说明 247 17.8 练习 248

18 Decision Trees 250
18.1 样本复杂度 251 18.2 决策树算法 252 18.2.1 收益度测量的实现 253 18.2.2 剪枝 254 18.2.3 实值特征的阈值分割规则 255 18.3 随机森林 255 18.4 概述 256 18.5 参考文献说明 256 18.6 练习 256

19 Nearest Neighbor 258
19.1 k 最近邻 258 19.2 分析 259 19.2.1 1-NN 规则的泛化界限 260 19.2.2 “维度诅咒” 263 19.3 高效实现* 264 19.4 概述 264 19.5 参考文献说明 264 19.6 练习 265

20 Neural Networks 268
20.1 前馈神经网络 269 20.2 学习神经网络 270 20.3 神经网络的表达能力 271 20.3.1 几何直觉 273 20.4 神经网络的样本复杂度 274 20.5 学习神经网络的运行时间 276 20.6 随机梯度下降和反向传播 277 20.7 概述 281 20.8 参考文献说明 281 20.9 练习 282

Part III Additional Learning Models 285

21 Online Learning 287
21.1 可行情况下的在线分类 288

21.1.1 在线可学习性 290 21.2 在不可实现情况下的在线分类 294 21.2.1 加权多数 295 21.3 在线凸优化 300 21.4 在线感知器算法 301 21.5 概述 304 21.6 参考文献说明 305 21.7 练习 305

22 Clustering 307
22.1 基于链接的聚类算法 310 22.2 ***k*-Means及其他成本最小化聚类** 311 22.2.1 k-Means算法 313 22.3 谱聚类 315 22.3.1 图割 315 22.3.2 图拉普拉斯和松弛图割 315 22.3.3 未归一化谱聚类 317 22.4 信息瓶颈* 317 22.5 聚类的概述 318 22.6 概述 320 22.7 参考文献说明 320 22.8 练习 320

23 Dimensionality Reduction 323
23.1 主成分分析（PCA）324 23.1.1 对 $d \gg m$ 的情况的一个更有效的解决方案326 23.1.2 实现和演示326 23.2 随机投影329 23.3 压缩感知330 23.3.1 证明*333 23.4 PCA 或压缩感知？ 338 23.5 概述338 23.6 参考文献说明339 23.7 练习339

24 Generative Models 342
24.1 最大似然估计器 343
24.1.1 连续随机变量的最大似然估计 344 24.1.2 最大似然与经验风险最小化 345 24.1.3 泛化分析 345

24.2 简单贝叶斯 347 24.3 线性判别分析 347 24.4 隐变量与EM算法 348

24.4.1 EM作为替代最大化算法 350 24.4.2 EM用于高斯混合（软k-均值） 352 24.5 贝叶斯推理 353 24.6 概述 355 24.7 参考文献说明 355 24.8 练习 356

25 Feature Selection and Generation 357
25.1 特征选择 358 25.1.1 过滤器 359 25.1.2 饥饿选择方法 360 25.1.3 稀疏性诱导范数 363 25.2 特征操作和归一化 365 25.2.1 特征变换示例 367 25.3 特征学习 368 25.3.1 使用自编码器的字典学习 368 25.4 概述 370 25.5 参考文献说明 371 25.6 练习 371

Part IV Advanced Theory 373

26 Rademacher Complexities 375
26.1 Rademacher复杂度 375 26.1.1 Rademacher微积分 379 26.2 线性类Rademacher复杂度 382 26.3 SVM泛化界限 383 26.4 低ℓ_1范数预测器的泛化界限 386 26.5 参考文献说明 386

27 Covering Numbers 388
27.1 覆盖 388 27.1.1 属性 388 27.2 通过链式从覆盖到Rademacher复杂性 389 27.3 参考文献说明 391

28 Proof of the Fundamental Theorem of Learning Theory 392
28.1 不可知情况的上界 392 28.2 不可知情况的下界 393 28.2.1 证明 $\boldsymbol{m(\epsilon,\delta) \geq 0.5\log(1/(4\delta))/\epsilon^2}$ 393 28.2.2 证明 $m(\epsilon, 1/8) \geq 8d/\epsilon^2$ 395 28.3 可实现情况的上界 398 28.3.1 从 ϵ-网到 PAC 可学习性 401

29 Multiclass Learnability 402
29.1 Natarajan 维度 402 29.2 多类基本定理 403 29.2.1 定理 29.3 的证明 403 29.3 计算Natarajan维度 404 29.3.1 一对一分类 404 29.3.2 通用多类到二类降维 405 29.3.3 线性多类预测器 405 29.4 关于好和坏的ERM 406 29.5 参考文献说明 408 29.6 练习 409

30 Compression Bounds 410
30.1 压缩界限 410 30.2 例子 412 30.2.1 轴对齐矩形 412 30.2.2 半空间 412 30.2.3 分离多项式 413 30.2.4 带边界的分离 414 30.3 参考文献说明 414

31 PAC-Bayes 415
31.1 PAC-Bayes 界限 415 31.2 参考文献说明 417 31.3 练习 417

Appendix A Technical Lemmas 419

Appendix B Measure Concentration 422

Appendix C Linear Algebra 430

Notes 435
References 437 *Index* 447

1 Introduction

本书的主题是自动学习，或者更常见地称为机器学习（ML）。也就是说，我们希望编程计算机，使它们能够“学习”它们可用的输入。粗略地说，学习是将经验转化为专业知识或知识的过程。学习算法的输入是训练数据，代表经验，输出是某种专业知识，这通常以另一种可以执行某些任务的计算机程序的形式出现。为了寻求对这个概念的形式-数学理解，我们必须更明确地说明我们所说的每个相关术语的含义：我们的程序将访问哪些训练数据？如何使学习过程自动化？我们如何评估这种过程的成功（即学习程序输出的质量）？

1.1 What Is Learning?

让我们从考虑自然发生的动物学习的一些例子开始。在ML中最基本的问题已经在我们熟悉的环境中出现了。

Bait Shyness – Rats Learning to Avoid Poisonous Baits: 當老鼠遇到外形或氣味新穎的食物時，它們首先會吃非常少量的食物，之後的喂食將取決於食物的味道和其生理作用。如果食物產生不良影響，新穎食物通常會與疾病相關聯，隨後，老鼠就不再吃這種食物。明顯地，這裡有一種學習機制在起作用——動物利用過去對某些食物的經驗來獲得識別這種食物安全性的專業知識。如果過去對食物的經驗被負面標籤，動物預測它在將來遇到時也會有負面影響。

受前面成功学习示例的启发，让我们演示一个典型的机器学习任务。假设我们想要编程一个机器学习如何过滤垃圾邮件。一个直观的解决方案似乎与老鼠学习如何避免有毒诱饵的方式相似。机器将简单地 *memorize* 所有被人类用户标记为垃圾邮件的先前电子邮件。当一封新邮件到达时，机器将在集合中搜索它。

之前的垃圾邮件。如果它与其中之一匹配，它将被删除。否则，它将被移至用户的收件箱文件夹。

虽然先前的“通过记忆学习”方法有时很有用，但它缺乏学习系统的一个重要方面——对未见电子邮件进行标记的能力。一个成功的学习者应该能够从个别例子进步到更广泛的 *generalization*。这也被称为 *inductive reasoning* 或 *inductive inference*。在之前提到的诱饵羞涩的例子中，老鼠在遇到某种类型食物的例子后，将它们对它的态度应用到新的、未见过的类似气味和味道的食物上。为了在垃圾邮件过滤任务中实现泛化，学习者可以扫描之前看到的电子邮件，并提取一组在电子邮件中出现可指示垃圾邮件的单词。然后，当收到一封新电子邮件时，机器可以检查是否有可疑的单词出现在其中，并据此预测其标签。这样的系统有可能正确预测未见电子邮件的标签。

然而，归纳推理可能会让我们得出错误的结论。为了说明这一点，让我们再次考虑一个来自动物学习的例子。

Pigeon Superstition: 在一个心理学家B. F. Skinner进行的实验中
他將一群饥饿的鸽子放在笼子里。笼子上已经安装了一个自动装置，定期向鸽子投放食物，与鸟类的行为没有任何关系。饥饿的鸽子在笼子里走来走去，当食物首次投放时，它发现每只鸽子都在进行某种活动（啄食、转动头部等）。食物的到来强化了每只鸟的特定行为，因此，每只鸟倾向于花更多的时间做同样的动作。这反过来又增加了下一次随机食物投放时，每只鸟再次进行该活动的可能性。结果是形成了一系列事件，强化了鸽子将食物投放与它们首次投放时进行的任何随机行为之间的联系。随后，它们继续勤奋地执行这些相同的动作。[1]

什么区分了导致迷信的学习机制和有用的学习？这个问题对于自动化学习者的开发至关重要。虽然人类学习者可以依靠常识来过滤掉随机无意义的结论，但一旦我们将学习任务出口到机器，我们必须提供明确的清晰原则，以保护程序不得出无意义或无用的结论。这种原则的发展是机器学习理论的核心目标。

那么，是什么使得老鼠的学习比鸽子的学习更成功？为了回答这个问题，让我们首先更深入地观察老鼠的诱饵恐惧现象。

Bait Shyness revisited – rats fail to acquire conditioning between food and electric shock or between sound and nausea: 鱼饵羞涩机制在

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

老鼠实际上比人们预期的要复杂。在Garcia（Garcia & Koelling 1996）进行的实验中，证明了如果食物消费后跟随的不愉快刺激被，比如说，电击（而不是恶心）所取代，那么就不会发生条件反射。即使在某些食物消费后反复进行的不愉快电击实验中，老鼠也不倾向于避免那种食物。当食物的特征（如味道或气味）暗示恶心时，用声音信号代替，也会出现类似的条件反射失败。老鼠似乎有一些“内置”的先验知识告诉它们，虽然食物和恶心的时空相关性可能是因果的，但食物消费和电击之间或声音和恶心之间不太可能存在因果关系。

我们得出结论，诱饵羞涩学习和鸽子的迷信之间的一个区别特征是引入了*prior knowledge*，这会偏斜学习机制。这也被称为*inductive bias*。实验中的鸽子愿意采用*any*解释食物的出现。然而，老鼠“知道”食物不会引起电击，并且噪音与某些食物的同时出现不太可能影响该食物的营养价值。老鼠的学习过程倾向于检测某种模式，而忽略事件之间的一些时间相关性。

结果表明，将先验知识纳入学习过程，对学习算法的成功是不可避免的（这一点在第五章中正式陈述并证明为“无免费午餐定理”）。开发用于表达领域专业知识、将其转化为学习偏差并量化这种偏差对学习成功的影响的工具，是机器学习理论的核心主题。简而言之，一个人在学习过程中开始时所拥有的先验知识（或先验假设）越强，从进一步示例中学习就越容易。然而，这些先验假设越强，学习就越不灵活——它事先就受制于对这些假设的承诺。我们将在第五章中明确讨论这些问题。

1.2 When Do We Need Machine Learning?

我们需要在何时使用机器学习而不是直接编程让计算机执行手头的任务？给定问题可能需要使用基于其“经验”学习和改进的程序的两个方面：问题的复杂性和对自适应性的需求。

Tasks That Are Too Complex to Program.

- *Tasks Performed by Animals/Humans:* 人类执行许多日常任务，但我们对这些任务如何进行的内省并不足够详细，以至于无法提取出良好的

定义程序。此类任务的例子包括驾驶、语音识别和图像理解。在这些任务中，最先进的机器学习程序，“从经验中学习”的程序，一旦接触到足够多的训练示例，就能取得相当满意的结果。

- *Tasks beyond Human Capabilities:* 另一大类受益于机器学习技术的任务是关于分析非常庞大和复杂的数据集：天文数据、将医学档案转化为医学知识、天气预报、基因组数据分析、网络搜索引擎和电子商务。随着越来越多的数字化记录数据可用，很明显，数据档案中蕴藏着大量有意义的 信息，这些信息对于人类来说太大、太复杂，难以理解。学习在大规模和复杂的数据集中检测有意义的模式是一个有希望的领域，其中计算机几乎无限的内存容量和不断增长的处理速度与学习程序的组合开辟了新的视野。

Adaptivity. 程序工具的一个限制性特征是其刚性——一旦程序被编写并安装，它就保持不变。然而，许多任务会随着时间的推移或用户的不同而发生变化。机器学习工具——其行为会适应其输入数据的程序——为这些问题提供了解决方案；它们本质上是对其所交互的环境变化具有适应性的。机器学习在解决此类问题中的典型成功应用包括解码手写文本的程序，其中固定程序可以适应不同用户手写的差异；自动适应垃圾邮件性质变化的垃圾邮件检测程序；以及语音识别程序。

1.3 Types of Learning

学习当然是一个非常广泛的领域。因此，机器学习的领域已经分化成几个子领域，分别处理不同类型的学习任务。我们提供了一个关于学习范式的粗略分类，旨在提供一些视角，了解本书的内容在机器学习广泛领域中的位置。

我们描述了四个参数，可以根据这些参数对学习范式进行分类。

Supervised versus Unsupervised 学习涉及学习者与环境之间的交互，因此可以根据这种交互的性质来划分学习任务。首先要注意的区别是监督学习和无监督学习之间的差异。作为一个

说明性示例，考虑学习检测垃圾邮件与异常检测的任务。对于垃圾邮件检测任务，我们考虑一个学习器接收训练电子邮件的设置，其中提供了标签 spam/not-spam。基于这样的训练，学习器应该找出一个规则来标记新到达的电子邮件消息。相比之下，对于异常检测任务，学习器所获得的训练只是一个大量的电子邮件消息（没有标签），学习器的任务是检测“不寻常”的消息。

更抽象地说，将学习视为“利用经验获得专业知识”的过程，监督学习描述了一种场景，其中“经验”，即训练示例，包含在未见的“测试示例”中缺失的显著信息（例如，spam/not-spam标签），而这些测试示例是应用所学专业知识的目标。在这种情况下，获得的专业知识旨在预测测试数据中缺失的信息。在这种情况下，我们可以将环境视为一个教师，通过提供额外的信息（标签）来“监督”学习者。然而，在无监督学习中，训练数据和测试数据之间没有区别。学习者处理输入数据的目标是得出一些总结或数据的压缩版本。将数据集聚类成相似对象的子集是此类任务的典型例子。

也存在一种中间学习设置，其中，虽然训练示例包含比测试示例更多的信息，但学习者需要为测试示例预测更多的信息。例如，可以尝试学习一个值函数，该函数描述了对于棋盘的每个设置，白方的位置比黑方更好的程度。然而，在训练时间学习者唯一可用的信息是实际棋局中发生的位置，并标记了谁最终赢得了那场比赛。这种学习框架主要在标题为 *reinforcement learning* 下进行研究。

Active versus Passive Learners 学习范式可以根据学习者的角色而有所不同。我们区分“主动”和“被动”学习者。主动学习者在训练时间与环境互动，例如，通过提出查询或进行实验，而被动学习者只观察环境（或教师）提供的信息，而不影响或指导它。请注意，垃圾邮件过滤器的学习者通常是被动的——等待用户标记发送给他们的电子邮件。在主动环境中，可以想象要求用户标记学习者选择的特定电子邮件，甚至由学习者自己编写的电子邮件，以增强其对垃圾邮件的理解。

Helpfulness of the Teacher 当人们思考人类学习时，无论是家中的婴儿还是学校的学生，这个过程通常涉及一位有帮助的老师，他试图向学习者提供最有用的信息，以{v*}

ful for achieving the learning goal. In contrast, when a scientist learns about n ature, the environment, playing the role of the teacher, can be best thought of as passive − apples drop, stars shine, and the rain falls without regard to the n eeds of the learner. We model such learning sce- narios by postulating that th e training data (or the learner's experience) is generated by some random pr ocess. This is the basic building block in the branch of "statistical learning. " Finally, learning also occurs when the learner's input is generated by an adversarial "teacher." This may be the case in the spam filtering example (if the spammer makes an effort to mislead the spam filtering designer) or in l earning to detect fraud. One also uses an adversarial teacher model as a worst -case scenario, when no milder setup can be safely assumed. If you can learn against an adversarial teacher, you are guaranteed to succeed interacting any odd teacher.

Online versus Batch Learning Protocol 最后我们提到的参数是学习者在整个学习过程中必须在线做出反应的情况与学习者有机会处理大量数据后才能运用所获得的专业知识的设置之间的区别。例如，股票经纪人必须根据迄今为止收集的经验做出每日决策。他可能随着时间的推移成为专家，但在过程中可能已经犯下了代价高昂的错误。相比之下，在许多数据挖掘设置中，学习者——数据挖掘者——在必须输出结论之前有大量的训练数据可供操作。

在这本书中，我们将仅讨论可能的学习范例的子集。我们的主要重点是带有被动学习者的监督统计批量学习（例如，尝试学习如何根据独立收集并已根据记录患者的命运进行标记的大量患者档案生成患者的预后）。我们还将简要讨论在线学习和批量无监督学习（特别是聚类）。

1.4 Relations to Other Fields

作为一个跨学科领域，机器学习与统计学、信息论、博弈论和优化等数学领域有着共同的主题。它自然是计算机科学的一个子领域，因为我们的目标是编程机器以便它们能够学习。从某种意义上说，机器学习可以被视为人工智能（Artificial Intelligence）的一个分支，因为毕竟，将经验转化为专业知识或检测复杂感官数据中的有意义模式是人类（和动物）智能的基石。然而，应该注意的是，与传统的AI相比，机器学习并不是试图构建智能行为的自动化模仿，而是利用优势

计算机补充人类智能的特殊能力，通常执行远远超出人类能力范围的任务。例如，扫描和处理大型数据库的能力使机器学习程序能够检测出人类感知范围之外的模式。

经验或训练在机器学习中的组成部分通常指的是随机生成的数据。学习者的任务是处理这些随机生成的示例，以得出适用于从中选取这些示例的环境的结论。这种对机器学习的描述突出了它与统计学的密切关系。事实上，这两个学科在目标和使用的技巧方面有很多共同之处。然而，在强调的几个重要差异中；如果一位医生提出了吸烟与心脏病之间存在相关性的假设，那么统计学家的作用就是查看患者样本并检查该假设的有效性（这是常见的统计任务，即假设检验）。相比之下，机器学习旨在使用从患者样本中收集的数据来描述心脏病的原因。希望自动化技术能够发现人类观察者可能错过的有意义模式（或假设）。

与传统的统计学相比，在机器学习领域，尤其是在本书中，算法考虑起着重要作用。机器学习是关于计算机执行学习的过程；因此，算法问题是至关重要的。我们开发算法来执行学习任务，并关注它们的计算效率。另一个区别是，虽然统计学通常对渐近行为（如随着样本量的增长到无穷大，基于样本的统计估计的收敛性）感兴趣，但机器学习理论主要关注有限样本界限。也就是说，给定可用的样本大小，机器学习理论旨在确定学习者基于此类样本可以期望的准确度。

这些两个学科之间还有进一步的差异，其中我们在这里只再提一个。在统计学中，通常假设某些预先订阅的数据模型（例如假设数据生成分布的常态性，或函数依赖的线性）进行工作，而在机器学习中，重点是在“无分布”的设置下工作，学习者尽可能少地假设数据分布的性质，并允许学习算法找出哪些模型最能近似数据生成过程。对这个问题的精确讨论需要一些技术预备知识，我们将在本书的后面部分回到这个问题，特别是在第5章中。

1.5 How to Read This Book

本书的第一部分提供了机器学习（ML）的基本理论原理。在某种意义上，这是其余部分的基础。

这本书的部分可以作为一个关于机器学习理论基础的迷你课程的基础。

本书的第二部分介绍了监督机器学习中最常用的算法方法。这些章节的子集也可用于向计算机科学、数学或工程学学生介绍机器学习。

本书的第三部分将讨论范围从统计分类扩展到其他学习模型。它涵盖了在线学习、无监督学习、降维、生成模型和特征学习。

本书的第四部分，高级理论，面向对研究感兴趣的读者，提供了用于分析和推动理论机器学习领域的更技术性的数学技术。

附录提供了书中使用的一些技术工具。特别是，我们列出了测度集中和线性代数的基本结果。

一些部分用星号标记，表示它们是为更高级的学生准备的。每个章节都以一个练习列表结束。课程网站上提供了答案手册。

1.5.1 Possible Course Plans Based on This Book

A 14 Week Introduction Course for Graduate Students:

1. 章节第2-4节。2. 第9章（不含VC计算）。3. 章节第5-6节（不含证明）。4. 第10章。5. 章节第7、11节（不含证明）。6. 章节第12、13节（包含一些较简单的证明）。7. 第14章（包含一些较简单的证明）。8. 第15章。9. 第16章。

10. 第18章。
11. 第22章。
12. 第23章（不含压缩感知的证明）。
13. 第24章。
1第4章 25节。

A 14 Week Advanced Course for Graduate Students:

1. 章节第26、27节。2. （续）3. 章节第6、28节。4. 章节第7节。5. 章节第31节。

6. 第30章。7. 第12
章、第13章。8. 第14
章。9. 第8章。10. 第
17章。11. 第29章。1
2. 第19章。13. 第20
章。14. 第21章。

1.6 Notation

本书中我们使用的符号大多数是标准的或现场定义的。在本节中，我们描述了我们的主要约定，并提供了一张总结我们符号的表格（表1.1）。鼓励读者在阅读本书时，如果遇到不清晰的符号，可以跳过本节，阅读后再回来查阅。

我们用小写字母表示标量和抽象对象（例如 x 和 λ）。通常，我们想强调某个对象是一个向量，然后我们使用粗体字母（例如 $\mathbf{x}$ 和 $\boldsymbol{\lambda}$）。向量 $\mathbf{x}$ 的 i 个元素表示为 x_i。我们用大写字母表示矩阵、集合和序列。含义应从上下文中清楚。正如我们马上将看到的，学习算法的输入是一系列训练示例。我们用 z 表示一个抽象示例，用 $S = z_1, \ldots, z_m$ 表示 m 个示例的序列。从历史上看，S 通常被称为训练 *set*；然而，我们将始终假设 S 是一个 *sequence* 而不是一个集合。m 个向量的序列表示为 $\mathbf{x}_1, \ldots, \mathbf{x}_m$。$\mathbf{x}_t$ 的 i 个元素表示为 $x_{t,i}$。

全书我们使用概率的基本概念。我们用 $\mathcal{D}$ 表示某个集合上的分布，例如，[2]。我们用 Z 表示 $z \sim \mathcal{D}$ 是根据 z 抽样的。给定一个随机变量 $f : Z \to \mathbb{R}$，其期望值表示为 $\mathbb{E}_{z\sim\mathcal{D}}[f(z)]$。当上下文中明确 $\mathbb{E}[f]$ 的依赖关系时，我们有时使用缩写。对于 $f : Z \to \{\texttt{true}, \texttt{false}\}$，我们也使用 $\mathbb{P}_{z\sim\mathcal{D}}[f(z)]$ 来表示 $\mathcal{D}(\{z : f(z) = \texttt{true}\})$。在下一章中，我们还将引入 $\mathcal{D}^m$ 的符号，表示由采样 $(z_1, \ldots, z_m)$ 诱导的 Z^m 上的概率，其中每个点 z_i 都独立于其他点从 $\mathcal{D}$ 中采样。

通常，我们努力避免使用渐近记号。然而，我们偶尔使用它来阐明主要结果。特别是，给定 $f : \mathbb{R} \to \mathbb{R}_+$ 和 $g : \mathbb{R} \to \mathbb{R}_+$，如果存在 $x_0, \alpha \in \mathbb{R}_+$ 使得对于所有 $x > x_0$ 我们都有 $f(x) \leq \alpha g(x)$，我们写 $f = O(g)$。如果对于每个 $\alpha > 0$ 都存在

[2] To be mathematically precise, $\mathcal{D}$ should be defined over some σ-algebra of subsets of Z. The user who is not familiar with measure theory can skip the few footnotes and remarks regarding more formal measurability definitions and assumptions.

Table 1.1 Summary of notation

symbol	meaning
$\mathbb{R}$	the set of real numbers
$\mathbb{R}^d$	the set of d-dimensional vectors over $\mathbb{R}$
$\mathbb{R}_+$	the set of non-negative real numbers
$\mathbb{N}$	the set of natural numbers
$O, o, \Theta, \omega, \Omega, \tilde{O}$	asymptotic notation (see text)
$\mathbb{1}_{[\text{Boolean expression}]}$	indicator function (equals 1 if expression is true and 0 o.w.)
$[a]_+$	$= \max\{0, a\}$
$[n]$	the set $\{1, \ldots, n\}$ (for $n \in \mathbb{N}$)
$\mathbf{x}, \mathbf{v}, \mathbf{w}$	(column) vectors
x_i, v_i, w_i	the ith element of a vector
$\langle \mathbf{x}, \mathbf{v} \rangle$	$= \sum_{i=1}^d x_i v_i$ (inner product)
$\|\mathbf{x}\|_2$ or $\|\mathbf{x}\|$	$= \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle}$ (the ℓ_2 norm of $\mathbf{x}$)
$\|\mathbf{x}\|_1$	$= \sum_{i=1}^d \|x_i\|$ (the ℓ_1 norm of $\mathbf{x}$)
$\|\mathbf{x}\|_\infty$	$= \max_i \|x_i\|$ (the ℓ_∞ norm of $\mathbf{x}$)
$\|\mathbf{x}\|_0$	the number of nonzero elements of $\mathbf{x}$
$A \in \mathbb{R}^{d,k}$	a $d \times k$ matrix over $\mathbb{R}$
$A^\top$	the transpose of A
$A_{i,j}$	the (i, j) element of A
$\mathbf{x}\,\mathbf{x}^\top$	the $d \times d$ matrix A s.t. $A_{i,j} = x_i x_j$ (where $\mathbf{x} \in \mathbb{R}^d$)
$\mathbf{x}_1, \ldots, \mathbf{x}_m$	a sequence of m vectors
$x_{i,j}$	the jth element of the ith vector in the sequence
$\mathbf{w}^{(1)}, \ldots, \mathbf{w}^{(T)}$	the values of a vector $\mathbf{w}$ during an iterative algorithm
$w_i^{(t)}$	the ith element of the vector $\mathbf{w}^{(t)}$
$\mathcal{X}$	instances domain (a set)
$\mathcal{Y}$	labels domain (a set)
Z	examples domain (a set)
$\mathcal{H}$	hypothesis class (a set)
$\ell : \mathcal{H} \times Z \to \mathbb{R}_+$	loss function
$\mathcal{D}$	a distribution over some set (usually over Z or over $\mathcal{X}$)
$\mathcal{D}(A)$	the probability of a set $A \subseteq Z$ according to $\mathcal{D}$
$z \sim \mathcal{D}$	sampling z according to $\mathcal{D}$
$S = z_1, \ldots, z_m$	a sequence of m examples
$S \sim \mathcal{D}^m$	sampling $S = z_1, \ldots, z_m$ i.i.d. according to $\mathcal{D}$
$\mathbb{P}, \mathbb{E}$	probability and expectation of a random variable
$\mathbb{P}_{z \sim \mathcal{D}}[f(z)]$	$= \mathcal{D}(\{z : f(z) = \texttt{true}\})$ for $f : Z \to \{\texttt{true}, \texttt{false}\}$
$\mathbb{E}_{z \sim \mathcal{D}}[f(z)]$	expectation of the random variable $f : Z \to \mathbb{R}$
$N(\boldsymbol{\mu}, C)$	Gaussian distribution with expectation $\boldsymbol{\mu}$ and covariance C
$f'(x)$	the derivative of a function $f : \mathbb{R} \to \mathbb{R}$ at x
$f''(x)$	the second derivative of a function $f : \mathbb{R} \to \mathbb{R}$ at x
$\frac{\partial f(\mathbf{w})}{\partial w_i}$	the partial derivative of a function $f : \mathbb{R}^d \to \mathbb{R}$ at $\mathbf{w}$ w.r.t. w_i
$\nabla f(\mathbf{w})$	the gradient of a function $f : \mathbb{R}^d \to \mathbb{R}$ at $\mathbf{w}$
$\partial f(\mathbf{w})$	the differential set of a function $f : \mathbb{R}^d \to \mathbb{R}$ at $\mathbf{w}$
$\min_{x \in C} f(x)$	$= \min\{f(x) : x \in C\}$ (minimal value of f over C)
$\max_{x \in C} f(x)$	$= \max\{f(x) : x \in C\}$ (maximal value of f over C)
$\operatorname{argmin}_{x \in C} f(x)$	the set $\{x \in C : f(x) = \min_{z \in C} f(z)\}$
$\operatorname{argmax}_{x \in C} f(x)$	the set $\{x \in C : f(x) = \max_{z \in C} f(z)\}$
log	the natural logarithm

$^{x}0$ 满足对所有 $x > x_0$ 我们有 $f(x) \leq \alpha g(x)$。我们写 $f = \Omega(g)$ 如果存在 $x_0, \alpha \in \mathbb{R}_+$ 使得对所有 $x > x_0$ 我们有 $f(x) \geq \alpha g(x)$。符号 $f = \omega(g)$ 定义类似。符号 $f = \Theta(g)$ 表示 $f = O(g)$ 和 $g = O(f)$。最后，符号 $f = \tilde{O}(g)$ 表示存在 $k \in \mathbb{N}$ 使得 $f(x) = O(g(x) \log^k(g(x)))$。

向量 $\mathbf{x}$ 和 $\mathbf{w}$ 的内积表示为 $\langle \mathbf{x}, \mathbf{w} \rangle$。当我们没有指定向量空间时，我们假设它是 d 维欧几里得空间，然后是 $\langle \mathbf{x}, \mathbf{w} \rangle = \sum_{i=1}^{d} x_i w_i$。向量 $\mathbf{w}$ 的欧几里得（或 ℓ_2）范数是 $\|\mathbf{w}\|_2 = \sqrt{\langle \mathbf{w}, \mathbf{w} \rangle}$。当从上下文中可以清楚地看出时，我们省略 ℓ_2 范数的下标。我们还使用其他 ℓ_p 范数，$\|\mathbf{w}\|_p = \left(\sum_i |w_i|^p\right)^{1/p}$，特别是 $\|\mathbf{w}\|_1 = \sum_i |w_i|$ 和 $\|\mathbf{w}\|_\infty = \max_i |w_i|$。

我们使用记号 $\min_{x \in C} f(x)$ 来表示集合 $\{f(x) : x \in C\}$ 的最小值。为了在数学上更加精确，当最小值不可达时，我们应该使用 $\inf_{x \in C} f(x)$。然而，在本书的语境中，下确界与最小值之间的区别通常并不重要。因此，为了简化表述，我们有时即使在 inf 更为合适的情况下也使用 min 记号。类似的说法也适用于 max 与 sup。

Part I

Foundations

2 A Gentle Start

让我们从展示如何在相对简化的环境中实现成功的学习开始我们的数学分析。想象你刚刚到达某个太平洋小岛。你很快发现木瓜是当地饮食中的重要成分。然而，你以前从未尝过木瓜。你必须学会预测你在市场上看到的木瓜是否美味。首先，你需要决定你的预测应该基于木瓜的哪些特征。基于你对其他水果的先前经验，你决定使用两个特征：木瓜的颜色，从深绿色到橙色和红色再到深棕色，以及木瓜的柔软度，从坚硬如石到泥状。你用来确定预测规则输入的是你检查过颜色和柔软度的木瓜样本，然后品尝并发现它们是否美味。让我们将这个任务分析为学习问题中涉及到的考虑因素的演示。

我们的第一步是描述一个旨在捕捉此类学习任务的正式模型。

2.1 A Formal Model – The Statistical Learning Framework

- **The learner's input:** 在基本统计学习设置中，学习器可以访问以下内容：
  - **Domain set:** 一个任意集合，$\mathcal{X}$。这是我们可能希望标记的对象集合。例如，在之前提到的木瓜学习问题中，域集将是所有木瓜的集合。通常，这些域点将由类似于木瓜的颜色和柔软度的*features* (向量表示。我们还将域点称为*instances*，将$\mathcal{X}$称为实例空间。
  - **Label set:** 对于我们的当前讨论，我们将标签集限制为两个元素的集合，通常是 $\{0, 1\}$ 或 $\{-1, +1\}$。让 $\mathcal{Y}$ 表示我们可能的标签集。在我们的木瓜例子中，让 $\mathcal{Y}$ 为 $\{0, 1\}$，其中 1 代表美味，0 表示不好吃。
  - **Training data:** $S = ((x_1, y1) \dots (x_m, y_m))$ 是 $\mathcal{X} \times \mathcal{Y}$ 中的有限序列对：也就是说，一个标记的域点序列。这是学习者可以访问的输入（就像一篮子已经

尝过它们的颜色、柔软度和美味。这样的标记示例通常被称为 *training examples*。有时我们也将 S 称为 *training set*。[1]

- **The learner's output:** 学习器被要求输出一个 *prediction rule*，$h\colon \mathcal{X} \to \mathcal{Y}$ 。此函数也被称为 *predictor*，*hypothesis* 或 *clas- sifier*。预测器可用于预测新领域点的标签。在我们的木瓜示例中，这是一条规则，我们的学习器将使用它来预测他在农贸市场检查的未来木瓜是否美味。我们使用符号 $A(S)$ 来表示学习算法 A 在接收到训练序列 S 后返回的假设。

- **A simple data-generation model** 我们现在解释如何生成训练数据。首先，我们假设实例（我们遇到的木瓜）是由某种概率分布（在这种情况下，代表环境）生成的。让我们用 $\mathcal{D}$ 表示 $\mathcal{X}$ 上的概率分布。需要注意的是，我们并不假设学习者了解这个分布。对于我们所讨论的学习任务类型，这可以是任何任意的概率分布。至于标签，在当前的讨论中，我们假设存在某种“正确”的标签函数 $f\colon \mathcal{X} \to \mathcal{Y}$，并且对于所有 i，有 $y_i = f(x_i)$。这个假设将在下一章中放宽。标签函数对学习者是未知的。事实上，这正是学习者试图弄清楚的事情。总之，训练数据中的每一对 S 都是通过首先根据 $\mathcal{D}$ 抽样一个点 x_i，然后通过 f 标记它来生成的。

- **Measures of success:** 我们定义 *error of a classifier* 为它不预测上述底层分布生成的随机数据点的正确标签的概率。也就是说，h 的错误是按照分布 $\mathcal{D}$ 抽取一个随机实例 x 的概率，使得 $h(x)$ 不等于 $f(x)$。

  形式上，给定一个域子集{v*}，概率分布{v*}分配一个数字{v*}，该数字决定了观察到点{v*}的可能性。在许多情况下，我们将{v*}称为事件，并使用函数{v*}：{v*}0{v*}1{v*}来表示它，即{v*}：{v*}1{v*}。在这种情况下，我们也使用符号{v*}[{v*}]来表示{v*}。

  我们定义预测规则h的错误，$\mathcal{X} \to \mathcal{Y}$，为

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

  这意味着此类 h 的错误是随机选择一个示例 x 的概率，使得 $h(x) \neq f(x)$ 。下标 $(\mathcal{D}, f)$ 表示错误是相对于概率分布 $\mathcal{D}$ 和

[1] Despite the "set" notation, S is a sequence. In particular, the same example may appear twice in S and some algorithms can take into account the order of examples in S.

[2] Strictly speaking, we should be more careful and require that A is a member of some σ-algebra of subsets of $\mathcal{X}$, over which $\mathcal{D}$ is defined. We will formally define our measurability assumptions in the next chapter.

正确的标记函数 f。当从上下文中可以明确时，我们省略这个下标。$L_{(\mathcal{D},f)}(h)$ 有几个同义词，如 *general- ization error*，*risk*，或 h 的 *true error*，本书中将交替使用这些名称。我们用字母 L 表示误差，因为我们把这种误差看作是学习者的 *loss*。我们还将讨论此类损失的其他可能公式。

- **A note about the information available to the learner** 学习者在世界分布 $\mathcal{D}$ 和标签函数 f 方面都是盲目的。在我们的木瓜例子中，我们刚刚到达一个新岛屿，我们对木瓜的分布以及如何预测它们的味道一无所知。学习者与环境互动的唯一方式是通过观察训练集。

在下一节中，我们描述了针对先前设置的简单学习范式并分析了其性能。

2.2 Empirical Risk Minimization

如前所述，学习算法接收一个训练集 S 作为输入，该训练集是从未知分布 $\mathcal{D}$ 中采样的，并由某个目标函数 f 标记，并且应该输出一个预测器 h_S：$\mathcal{X} \to \mathcal{Y}$ (下标 S 强调了输出预测器依赖于 S) 的这一事实。算法的目标是找到 h_S，以最小化与未知 $\mathcal{D}$ 和 f 相关的错误。

由于学习者不知道 $\mathcal{D}$ 和 f 是什么，因此真正的错误无法直接提供给学习者。学习者可以计算的有用错误概念是 *training error* ——分类器在训练样本上产生的错误：

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

在 $[m] = \{1, \ldots, m\}$。

empirical error 和 *empirical risk* 通常被互换用于表示此错误。

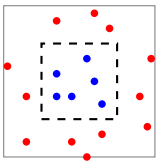
由于训练样本是学习者可获得的世界的快照，因此寻找在该数据上表现良好的解决方案是有意义的。这种学习范式——提出一个最小化$L_S(h)$的预测器h——被称为*Empirical Risk Minimization*或简称ERM。

2.2.1 Something May Go Wrong – Overfitting

尽管ERM规则看起来非常自然，如果不小心，这种方法可能会彻底失败。

为了演示这种失败，让我们回到学习到的问题。

预测基于其软度和颜色的木瓜口感。考虑以下所示样本：



假设概率分布 $\mathcal{D}$ 是这样的，实例在灰色正方形内均匀分布，标签函数 f 决定如果实例在内部蓝色正方形内，则标签为 1，否则为 0。图中灰色正方形的面积为 2，蓝色正方形的面积为 1。考虑以下预测器：

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

虽然这个预测因子可能看起来相当人为，但在练习1中，我们展示了使用多项式对其的自然表示。显然，无论样本是什么，$L_S(h_S) = 0$，因此这个预测因子可以通过ERM算法选择（它是经验最小成本假设之一；没有分类器可以具有更小的误差）。另一方面，任何仅在有限个实例上预测标签1的分类器的真实误差在这种情况下是1/2。因此，$L_{\mathcal{D}}(h_S) = 1/2$。我们找到了一个在训练集上的性能极好的预测器，但在真实“世界”上的性能非常差。这种现象被称为*overfitting*。直观地说，当我们的假设对训练数据“拟合得太好”时（也许就像日常经验中，一个人为他的每一个单独的行为提供完美的详细解释可能会引起怀疑），就会发生过度拟合。

2.3 Empirical Risk Minimization with Inductive Bias

我们刚刚证明了ERM规则可能会导致过拟合。我们不会放弃ERM范式，而是会寻找纠正它的方法。我们将寻找保证ERM不会过拟合的条件，即当ERM预测器在训练数据上表现良好时，它也很可能在底层数据分布上表现良好的条件。

一个常见的解决方案是在一个受限的搜索空间中应用ERM学习规则。形式上，学习者在看到数据之前应该预先选择一组预测因子。这个集合被称为*hypothesis class*，表示为$\mathcal{H}$。每个$h \in \mathcal{H}$是一个从$\mathcal{X}$到$\mathcal{Y}$的映射函数。对于给定的类别$\mathcal{H}$和一个训练样本S，ERM$_{\mathcal{H}}$学习者使用ERM规则选择一个预测因子$h \in \mathcal{H}$，

在可能的最小错误率下超过 S。形式上，

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

argmin表示在$\mathcal{H}$中达到$L_S(h)$在$\mathcal{H}$上最小值的假设集合。通过限制学习器从$\mathcal{H}$中选择预测器，我们将它$bias$引导到一组特定的预测器。这种限制通常被称为 $inductive\ bias$。由于这种限制的选择是在学习器看到训练数据之前确定的，因此理想情况下应基于对要学习的问题的一些先验知识。例如，对于木瓜味道预测问题，我们可能选择$\mathcal{H}$为通过轴对齐矩形（在由颜色和柔软度坐标确定的空间中）确定的预测器集合。我们将在以后证明，在这个类上$\mathrm{ERM}_{\mathcal{H}}$保证不会过拟合。另一方面，我们之前看到的过拟合示例表明，选择$\mathcal{H}$为包括将值1分配给有限域点的所有函数的预测器类，不足以保证$\mathrm{ERM}_{\mathcal{H}}$不会过拟合。

学习理论中的一个基本问题是，在哪些假设类中，$\mathrm{ERM}_{\mathcal{H}}$学习不会导致过拟合。我们将在本书后面研究这个问题。

直观上，选择一个更受限的假设类更好地保护我们免受过拟合的影响，但同时也可能给我们带来更强的归纳偏差。我们稍后会回到这个基本权衡。

2.3.1 Finite Hypothesis Classes

最简单的对类别的限制是对其大小（即 h 中的预测因子数量）施加上限。在本节中，我们表明，如果 $\mathcal{H}$ 是一个有限类别，那么只要基于足够大的训练样本（这个大小要求将取决于 $\mathcal{H}$ 的大小），$\mathrm{ERM}_{\mathcal{H}}$ 就不会过拟合。

限制学习器在某个有限假设类内的预测规则可能被视为一种相当温和的限制。例如，$\mathcal{H}$可以是可以通过最多10^9位代码编写的C++程序实现的全部预测器的集合。在我们的 Papayas 示例中，我们之前提到了轴对齐矩形的类别。虽然这是一个无限类别，如果我们通过使用64位浮点表示法对实数的表示进行离散化，假设类就变成了一个有限类别。

让我们现在分析在假设 $\mathcal{H}$ 是一个有限类的情况下，$\mathrm{ERM}_{\mathcal{H}}$ 学习规则的性能。对于一个训练样本 S，按照某些 $f : \mathcal{X} \to \mathcal{Y}$ 标记，让 h_S 表示将 ERM_H 应用到 S 的结果，即，

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

在这一章中，我们做出以下简化假设（将在下一章中放宽）。

DEFINITION 2.1（可实现性假设）存在 $h^\star \in \mathcal{H}$，使得 $L_{(\mathcal{D},f)}(h^\star) = 0$。请注意，这个假设意味着在随机样本的概率为1的情况下，S，其中 S 的实例根据 $\mathcal{D}$ 进行采样，并由 f 标记，我们有 $L_S(h^\star) = 0$。

可实现性假设意味着对于每个ERM假设，我们有[3] $L_S(h_S) = 0$。然而，我们感兴趣的*true*风险是h_S，$L_{(\mathcal{D},f)}(h_S)$的风险，而不是其经验风险。

显然，对于仅能访问样本 S 的算法，关于底层分布 $\mathcal{D}$ 的任何误差保证都应取决于 $\mathcal{D}$ 和 S 之间的关系。在统计机器学习中，常见的假设是训练样本 S 是通过从分布 $\mathcal{D}$ 中独立采样点生成的。形式上，

- **The i.i.d. assumption:** 训练集中的示例根据分布 $\mathcal{D}$ 独立同分布（i.i.d.）。也就是说，S 中的每个 x_i 都是按照 $\mathcal{D}$ 新鲜抽取的，然后根据标签函数 f 进行标记。我们用 $S \sim \mathcal{D}^m$ 表示这个假设，其中 m 是 S 的大小，$\mathcal{D}^m$ 表示通过将 $\mathcal{D}$ 应用到元组的每个元素，独立于元组的其他成员来诱导的 m-元组的概率。

  直观上，训练集 S 是一个窗口，通过这个窗口学习者可以获取关于世界分布 $\mathcal{D}$ 和标签函数 f 的部分信息。样本越大，就越有可能更准确地反映用于生成它的分布和标签。

由于 $L_{(\mathcal{D},f)}(h_S)$ 依赖于训练集 S，而该训练集是通过随机过程选择的，因此预测变量 h_S 的选择以及随之而来的风险 $L_{(\mathcal{D},f)}(h_S)$ 具有随机性。正式地说，我们称其为随机变量。期望 S 能够完全确定地引导学习器（从 $\mathcal{D}$ 的角度来看）指向一个好的分类器是不现实的，因为始终存在一些概率，即采样的训练数据可能恰好非常不具有代表性，对于潜在的 $\mathcal{D}$。如果我们回到木瓜品尝的例子，总有一些（小的）可能性，即我们恰好品尝到的所有木瓜都不美味，尽管比如说，我们岛屿上的 70% 的木瓜都是美味的。在这种情况下，$\text{ERM}_{\mathcal{H}}(S)$ 可能是恒等函数，将每个木瓜标记为“不好吃”（在岛屿上木瓜的真实分布上有 70% 的错误率）。因此，我们将解决 *probability* 以采样一个训练集，其中 $L_{(\mathcal{D},f)}(h_S)$ 不太大。通常，我们用 δ 表示得到非代表性样本的概率，并将 $(1-\delta)$ 称为我们的预测的 *confidence parameter*。

在之上，由于我们无法保证完美的标签预测，我们引入另一个参数来衡量预测质量，即 *accuracy parameter*。

3 从数学上讲，这以概率1成立。为了简化表述，我们有时省略“以概率1”的指定。

通常表示为 ϵ。我们将事件 $L_{(\mathcal{D},f)}(h_S) > \epsilon$ 解释为学习者的失败，而如果 $L_{(\mathcal{D},f)}(h_S) \leq \epsilon$，我们将算法的输出视为一个近似正确的预测器。因此（固定某些标记函数 $f: \mathcal{X} \to \mathcal{Y}$），我们感兴趣的是上界采样将导致学习者失败的实例 m 元组的概率。形式上，让 $S|_x = (x_1, \ldots, x_m)$ 为训练集的实例。我们希望上界

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

设 $\mathcal{H}_B$ 为“坏”假设的集合，即，

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

此外，设

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

误导样本集：即对于每一个 $S|_x \in M$，存在一个“坏”的假设 $h \in \mathcal{H}_B$，它在 $S|_x$ 上看起来像是一个“好”的假设。现在，回想一下我们想要约束事件 $L_{(\mathcal{D},f)}(h_S) > \epsilon$ 的概率。但是，由于可实现性假设意味着 $L_S(h_S) = 0$，因此可以得出结论，事件 $L_{(\mathcal{D},f)}(h_S) > \epsilon$ 只能在某些 $h \in \mathcal{H}_B$ 满足 $L_S(h) = 0$ 的情况下发生。换句话说，这个事件只有在我们的样本在误导样本集 M 中时才会发生。形式上，我们已经证明了

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

请注意，我们可以将 M 重写为

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

因此，

$$\mathcal{D}^m(\{S|_x : L_{(\mathcal{D},f)}(h_S) > \epsilon\}) \ \leq \ \mathcal{D}^m(M) = \mathcal{D}^m(\cup_{h \in \mathcal{H}_B} \{S|_x : L_S(h) = 0\}). \tag{2.6}$$

接下来，我们使用 *union bound* 对前述方程的右侧进行上界估计——这是概率的一个基本性质。

LEMMA 2.2（并集界） *For any two sets A, B and a distribution $\mathcal{D}$ we have*

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

应用并集界于方程（2.6）的右侧，得到

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

接下来，让我们将前一个不等式右侧的每一项进行界定。固定一些“坏”的假设 $h \in \mathcal{H}_B$。事件 $L_S(h) = 0$ 等价于

到事件 $\forall i, h(x_i) = f(x_i)$。由于训练集中的示例是独立同分布采样的，我们得到

$$\begin{aligned}\mathcal{D}^m(\{S|_x : L_S(h) = 0\}) &= \mathcal{D}^m(\{S|_x : \forall i, h(x_i) = f(x_i)\}) \\ &= \prod_{i=1}^{m} \mathcal{D}(\{x_i : h(x_i) = f(x_i)\}). \end{aligned} \tag{2.8}$$

对于训练集中每个元素的采样，我们有

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

在最后一个不等式由 $h \in \mathcal{H}_B$ 的事实得出。将前一个方程与方程（2.8）结合并使用不等式 $1 - \epsilon \leq e^{-\epsilon}$，我们得到对于每一个 $h \in \mathcal{H}_B$,

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

将此方程与方程（2.7）结合，我们得出结论

$$\mathcal{D}^m(\{S|_x : L_{(\mathcal{D},f)}(h_S) > \epsilon\}) \;\leq\; |\mathcal{H}_B|\, e^{-\epsilon m} \;\leq\; |\mathcal{H}|\, e^{-\epsilon\, m}.$$

图2.1给出了一个图形说明，解释了我们如何使用并集界。

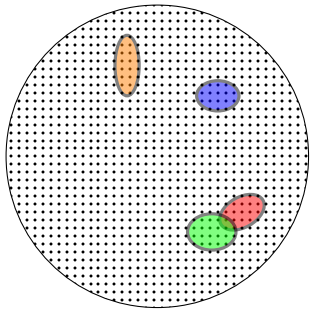


Figure 2.1 每个大圆中的点代表一个可能的 m 元组实例。每个彩色椭圆形代表某些“不良”预测器 $h \in \mathcal{H}_B$ 的“误导” m 元组实例集合。当 ERM 获得误导性训练集 S 时，它可能会过度拟合。也就是说，对于某些 $h \in \mathcal{H}_B$，我们有 $L_S(h) = 0$。

方程（2.9）保证对于每个单独的坏假设$h \in \mathcal{H}_B$，最多有$(1 - \epsilon)^m$的分数的训练集会是误导性的。特别是，m越大，这些彩色椭圆形的每个就越小。并集界限形式化了这样一个事实：表示相对于某些$h \in \mathcal{H}_B$ (的误导性训练集的区域，即M)中的训练集，至多是彩色椭圆形面积的加和。因此，它被限制在$|\mathcal{H}_B|$乘以彩色椭圆形最大尺寸。任何在彩色椭圆形之外的样本S都不能导致ERM规则过拟合。

COROLLARY 2.3 *Let $\mathcal{H}$ be a finite hypothesis class. Let $\delta \in (0,1)$ and $\epsilon > 0$*

and let m be an integer that satisfies

$$m \geq \frac{\log(|\mathcal{H}|/\delta)}{\epsilon}.$$

Then, for any labeling function, f, and for any distribution, $\mathcal{D}$, for which the realizability assumption holds (that is, for some $h \in \mathcal{H}$, $L_{(\mathcal{D},f)}(h) = 0$), with probability of at least $1-\delta$ over the choice of an i.i.d. sample S of size m, we have that for every ERM hypothesis, h_S, it holds that

$$L_{(\mathcal{D},f)}(h_S) \leq \epsilon.$$

前述推论告诉我们，对于足够大的 m，在有限假设类上的 ERM$_{\mathcal{H}}$ 规则将以 1 $-\delta$) *approximately* 的置信度 (正确到 ϵ) 的误差。在下一章中，我们将正式定义“可能近似正确”（PAC）学习模型。

2.4 Exercises

1. **Overfitting of polynomial matching:** 我们已经表明，在方程（2.3）中定义的预测器会导致过拟合。虽然这个预测器看起来非常不自然，但这个练习的目标是表明它可以被描述为一个阈值多项式。也就是说，要证明对于给定的训练集 $S = \{(\mathbf{x}_i, f(\mathbf{x}_i))\}_{i=1}^m \subseteq (\mathbb{R}^d \times \{0,1\})^m$，存在一个多项式 p_S，使得 $h_S(\mathbf{x}) = 1$ 当且仅当 $p_S(\mathbf{x}) \geq 0$，其中 h_S 如方程（2.3）中定义。因此，使用ERM规则学习所有阈值多项式的类别可能会导致过拟合。

2. 令 $\mathcal{H}$ 为一个在域 $\mathcal{X}$ 上的二元分类器类。令 $\mathcal{D}$ 为 $\mathcal{X}$ 上的一个未知分布，令 f 为 $\mathcal{H}$ 中的目标假设。固定一些 $h \in \mathcal{H}$。证明在 $S|_x$ 的选择下 $L_S(h)$ 的期望值等于 $L_{(\mathcal{D},f)}(h)$，即，

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

3. **Axis aligned rectangles:** 平面上的一个轴对齐矩形分类器是一种分类器，如果且仅当一个点位于某个矩形内，则将其值赋为1。形式上，对于实数 $a_1 \leq b_1, a_2 \leq b_2$，通过以下方式定义分类器 $h_{(a_1,b_1,a_2,b_2)}$：

$$h_{(a_1,b_1,a_2,b_2)}(x_1,x_2) = \begin{cases} 1 & \text{if } a_1 \leq x_1 \leq b_1 \text{ and } a_2 \leq x_2 \leq b_2 \\ 0 & \text{otherwise} \end{cases}. \tag{2.10}$$

所有平面中轴对齐矩形的类别定义为

$$\mathcal{H}^2_{\text{rec}} = \{h_{(a_1,b_1,a_2,b_2)} : a_1 \leq b_1, \text{ and } a_2 \leq b_2\}.$$

请注意，这是一个无限大小的假设类。在整个练习中，我们依赖于可实现性假设。

1. 令 A 为返回包含训练集中所有正例的最小矩形的算法。证明 A 是一个 ERM。
2. 证明如果 A 接收到大小为 $\geq \frac{4\log(4/\delta)}{\epsilon}$ 的训练集，那么，以至少 $1-\delta$ 的概率，它返回一个误差不超过 ϵ 的假设。

Hint: 修复一些分布 $\mathcal{D}$ 在 $\mathcal{X}$ 上，令 $R^* = R(a_1^*, b_1^*, a_2^*, b_2^*)$ 为生成标签的矩形，令 f 为相应的假设。令 $a_1 \geq a_1^*$ 为一个数，使得矩形 $R_1 = R(a_1^*, a_1, a_2^*, b_2^*)$ 的概率质量（相对于 $\mathcal{D}$）正好是 $\epsilon/4$。类似地，令 b_1, a_2, b_2 为一些数，使得矩形的概率质量 $R_2 = R(b_1, b_1^*, a_2^*, b_2^*), R_3 = R(a_1^*, b_1^*, a_2^*, a_2), R_4 = R(a_1^*, b_1^*, b_2, b_2^*)$ 都正好是 $\epsilon/4$。令 $R(S)$ 为 A 返回的矩形。参见图 2.2 的插图。

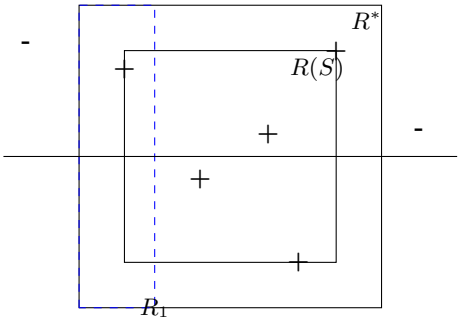


Figure 2.2 轴对齐矩形。

- 证明 $R(S) \subseteq R^*$。
- 证明如果 S 包含所有矩形 R_1, R_2, R_3, R_4 中的（正）示例，那么 A 返回的假设的错误最多为 ϵ。

- 对于每个 $i \in \{1, \ldots, 4\}$，上界化 S 不包含来自 R_i 的示例的概率。

- 使用并集界得出论点。3. 对$\mathbb{R}^d$类轴对齐矩形重复先前的提问。4. 证明先前提到的算法A的运行时间是关于d, $1/\epsilon$,和$\log(1/\delta)$的多项式。

3 A Formal Learning Model

在这一章中，我们定义了我们的主要形式学习模型——PAC学习模型及其扩展。我们将在第7章考虑其他可学习性的概念。

3.1 PAC Learning

在上一章中，我们已表明对于有限假设类，如果对该类应用足够大的训练样本（其大小与底层分布或标签函数无关）的ERM规则，则输出的假设将可能是近似正确的。更普遍地，我们现在定义*Probably Approximately Correct* (PAC)学习。

DEFINITION 3.1 (PAC可学习性) 一个假设类 $\mathcal{H}$ 是PAC可学习的，如果存在一个函数 $m_{\mathcal{H}} : (0,1)^2 \to \mathbb{N}$ 和一个具有以下性质的学习算法：对于每个 $\epsilon, \delta \in (0,1)$，对于每个在 $\mathcal{X}$ 上的分布 $\mathcal{D}$，以及对于每个标记函数 $f : \mathcal{X} \to \{0,1\}$，如果关于 $\mathcal{H}, \mathcal{D}, f$ 的可实现假设成立，那么当在由 $\mathcal{D}$ 生成并由 f 标记的 $m \geq m_{\mathcal{H}}(\epsilon, \delta)$ 独立同分布示例上运行学习算法时，算法返回一个假设 h，使得在至少 $1-\delta$ (的概率下，关于示例的选择)，$L_{(\mathcal{D},f)}(h) \leq \epsilon$。

PAC可学习性的定义包含两个近似参数。准确度参数 ϵ 决定了输出分类器可以偏离最优分类器多远（这对应于“近似正确”），以及一个置信度参数 δ，表示分类器满足该准确度要求的可能性有多大（对应于“PAC”中的“可能”部分）。在我们正在研究的数据访问模型下，这些近似是不可避免的。由于训练集是随机生成的，它可能总是有微小的机会是非信息性的（例如，总是有可能训练集只包含一个领域点，反复采样）。此外，即使我们足够幸运，得到了一个忠实代表 $\mathcal{D}$ 的训练样本，因为它只是一个有限样本，它可能总是有一些关于 $\mathcal{D}$ 的细微细节无法捕捉到。

反映。我们的准确度参数 ϵ 允许对学习者的分类器犯小错误“宽容”。

Sample Complexity
函数 $m_{\mathcal{H}} : (0,1)^2 \to \mathbb{N}$ 确定了学习 $\mathcal{H}$ 的 *sample complexity*：即需要多少个示例才能保证一个概率上近似正确的解。样本复杂度是准确度 (ϵ) 和置信度 (δ) 参数的函数。它还取决于假设类 $\mathcal{H}$ 的属性——例如，对于有限类，我们表明样本复杂度取决于 $\mathcal{H}$ 大小的对数。

注意，如果 $\mathcal{H}$ 是 PAC 可学习的，那么存在许多函数 $m_{\mathcal{H}}$ 满足 PAC 可学习性定义中给出的要求。因此，为了精确起见，我们将定义学习 $\mathcal{H}$ 的样本复杂度为“最小函数”，即对于任何 ϵ, δ，$m_{\mathcal{H}}(\epsilon, \delta)$ 是满足以准确率 ϵ 和置信度 δ 进行 PAC 学习的最小整数。

让我们现在回忆一下上一章中关于有限假设类分析的结论。它可以重新表述为：

COROLLARY 3.2 *Every finite hypothesis class is PAC learnable with sample complexity*

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

存在无限类也可以学习（例如，参见练习3）。稍后我们将展示，决定一个类PAC可学习性的不是其有限性，而是一个称为*VC dimension*的组合度量。

3.2 A More General Learning Model

该模型我们刚刚描述的可以很容易地进行推广，使其与更广泛的学习任务相关。我们在两个方面考虑推广：{v*}

Removing the Realizability Assumption
我们要求学习算法在满足可实现性假设的条件下，在数据分布 $\mathcal{D}$ 和标签函数 f 的对上成功。对于实际的学习任务，这个假设可能过于严格（我们真的能保证在颜色硬度空间中存在一个矩形，其中 *fully determines* 番荔枝是美味的吗？）。在下一个小节中，我们将描述一个 *agnostic PAC* 模型，其中这个可实现性假设被放弃。

Learning Problems beyond Binary Classification
我们迄今为止一直在讨论的学习任务与预测给定示例的二进制标签（如是否美味）有关。然而，许多学习任务具有不同的形式。例如，有人可能希望预测一个实数值（比如明天晚上9点的温度）或从有限标签集中选择的标签（比如明天报纸上主要故事的主题）。事实证明，通过允许各种损失函数，我们的学习分析可以很容易地扩展到这样的场景以及许多其他场景。我们将在3.2.2节中讨论这一点。

3.2.1 Releasing the Realizability Assumption – Agnostic PAC Learning

A More Realistic Model for the Data-Generating Distribution
回忆一下，可实现性假设要求存在 $h^\star \in \mathcal{H}$ 使得 $\mathbb{P}_{x\sim\mathcal{D}}[h^\star(x) = f(x)] = 1$。在许多实际问题中，这个假设并不成立。此外，可能更现实的是不假设标签完全由我们在输入元素上测量的特征决定（在木瓜的情况下，两个颜色和软度相同的木瓜可能味道不同）。在以下内容中，我们通过用更灵活的概念“数据标签生成分布”来代替“目标标签函数”来放宽可实现性假设。

形式上，从现在起，令 $\mathcal{D}$ 是 $\mathcal{X} \times \mathcal{Y}$ 上的一个概率分布，其中，如前所述，$\mathcal{X}$ 是我们的域集，$\mathcal{Y}$ 是一组标签（通常我们将考虑 $\mathcal{Y} = \{0, 1\}$）。也就是说，$\mathcal{D}$ 是域点和标签上的一个 *joint distribution*。可以将此类分布视为由两部分组成：一个分布 $\mathcal{D}_x$ 在未标记的域点上（有时称为 *marginal distribution*）和一个 *conditional* 概率，对于每个域点，$\mathcal{D}((x, y)|x)$。在木瓜示例中，$\mathcal{D}_x$ 确定了遇到颜色和硬度落在某些颜色-硬度值域内的木瓜的概率，条件概率是具有由 x 表示的颜色和硬度的木瓜是美味的水果的概率。实际上，这种建模允许两个颜色和硬度相同的木瓜属于不同的口味类别。

The empirical and the True Error Revised
对于一个概率分布 $\mathcal{D}$ 在 $\mathcal{X} \times \mathcal{Y}$ 上，可以衡量 h 在根据 $\mathcal{D}$ 随机抽取标记点时犯错的概率。我们重新定义预测规则 h 的真实错误（或风险）为

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

我们希望找到一个预测器，h，使得误差最小化。然而，学习器不知道生成数据$\mathcal{D}$。学习器所能访问的是训练数据，S。经验风险的定义

与之前相同，即，

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

给定 S，学习者可以计算任何函数 $h : X \to \{0, 1\}$ 的 $L_S(h)$。注意 $L_S(h) = L_{D(\text{uniform over } S)}(h)$。

The Goal
我们希望找到一些假设，$h : \mathcal{X} \to \mathcal{Y}$，它（可能近似地）最小化了真实风险，$L_D(h)$。

The Bayes Optimal Predictor.
给定任何在 $\mathcal{X} \times \{0, 1\}$ 上的概率分布 $\mathcal{D}$，从 $\mathcal{X}$ 到 $\{0, 1\}$ 的最佳标签预测函数将是

$$f_{\mathcal{D}}(x) = \begin{cases} 1 & \text{if } \mathbb{P}[y = 1|x] \geq 1/2 \\ 0 & \text{otherwise} \end{cases}$$

它很容易验证（参见练习7），对于每个概率分布{v*}，贝叶斯最优预测器{v*}是最优的，即没有其他分类器{v*}：{v*}0{v*}1{v*}具有更低的错误率。也就是说，对于每个分类器{v*}，{v*}。

很遗憾，由于我们不知道 $\mathcal{D}$，我们无法利用这个最优预测器 $f_{\mathcal{D}}$。学习者能够访问的是训练样本。现在我们可以提出无偏PAC学习能力的正式定义，这是PAC学习能力定义的自然扩展，适用于我们刚刚讨论的更现实、不可实现的设置。

显然，我们无法期望学习算法找到一个错误小于最小可能错误，即贝叶斯预测器的假设。

此外，正如我们将在后面证明的那样，一旦我们对数据生成分布没有任何先验假设，就无法保证任何算法能够找到与贝叶斯最优预测器一样好的预测器。相反，我们要求学习算法找到的预测器的误差不会比某些给定基准假设类中预测器的最佳可能误差大得多。当然，这种要求的力量取决于该假设类的选择。

DEFINITION 3.3（不可知PAC学习性）如果一个假设类$\mathcal{H}$是可不可知PAC学习的，当存在一个函数$m_{\mathcal{H}}$：$(0, 1)^2 \to \mathbb{N}$和一个具有以下特性的学习算法：对于每一个$\epsilon, \delta \in (0, 1)$和对于每一个在$\mathcal{X} \times \mathcal{Y}$上的分布$\mathcal{D}$，当在由$\mathcal{D}$生成的$m \geq m_{\mathcal{H}}(\epsilon, \delta)$独立同分布示例上运行学习算法时，算法返回一个假设$h$，使得，在至少$1 - \delta$（关于选择$m$训练示例的概率下，

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

显然，如果可实现性假设成立，无偏PAC学习提供了与PAC学习相同的保证。从这个意义上讲，无偏PAC学习推广了PAC学习的定义。当可实现性假设不成立时，没有任何学习器可以保证任意小的错误。然而，在无偏PAC学习的定义下，如果一个学习器的错误不是比从类别$\mathcal{H}$的预测器所能达到的最佳错误大得多，它仍然可以宣布成功。这与PAC学习形成对比，在PAC学习中，学习器被要求在绝对意义上而不是相对于假设类所能达到的最佳错误实现小错误。

3.2.2 The Scope of Learning Problems Modeled

我们接下来扩展我们的模型，使其能够应用于各种学习任务。让我们考虑一些不同学习任务的例子。

- **Multiclass Classification** 我们的分类不一定是二元的。以文档分类任务为例：我们希望设计一个程序，能够根据主题（例如，新闻、体育、生物学、医学）对给定的文档进行分类。此类任务的学习算法将能够访问正确分类的文档示例，并基于这些示例输出一个程序，该程序可以接受新文档作为输入，并输出该文档的主题分类。在这里，*domain set* 是所有潜在文档的集合。再次强调，我们通常用一组 *features* 来表示文档，这可以包括文档中不同关键词的计数，以及其他可能相关的特征，如文档的大小或其来源。在这个任务中，*label set* 将是可能的文档主题集合（因此 $\mathcal{Y}$ 将是某个大型有限集合）。一旦我们确定了我们的领域和标签集，我们框架的其他组件看起来与 papaya 品尝示例中的完全一样；我们的 *training sample* 将是（特征向量, 标签）对的有限序列，学习者的输出将是从领域集到标签集的函数，最后，为了衡量我们的成功，我们可以使用（文档，主题）对上预测器建议错误标签的事件的概率。

- **Regression** 在这个任务中，人们希望找到数据中的某些简单 *pattern* – 数据的 $\mathcal{X}$ 和 $\mathcal{Y}$ 成分之间的函数关系。例如，人们希望找到一个线性函数，根据婴儿头围、腹围和大腿长度的超声测量值来最好地预测婴儿的出生体重。在这里，我们的域集 $\mathcal{X}$ 是 $\mathbb{R}^3$ (的某个子集，即三个超声测量值)，而“标签”集 $\mathcal{Y}$ 是实数集（以克为单位的重量）。在这种情况下，更合适地称 $\mathcal{Y}$ 为 *target* 集合。我们的训练数据和学习者的输出与之前相同（有限序列的 (x, y) 对，以及从 $\mathcal{X}$ 到 $\mathcal{Y}$ 的函数）。然而，我们的成功度量是

不同的。我们可以通过真实标签和它们的预测值之间的
expected square difference 来评估假设函数 $h : \mathcal{X} \to \mathcal{Y}$ 的质量，即，

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

为了适应广泛的学习任务，我们如下泛化我们的成功度量形式化：

Generalized Loss Functions
给定任何集合 $\mathcal{H}$ (，它扮演我们的假设或模型) 的角色，或者某个域 Z，令 ℓ 是从 $\mathcal{H}\times Z$ 到非负实数集 ℓ 的任意函数，$\mathcal{H} \times Z \to \mathbb{R}_+$: *loss functions*。

我们称这样的函数为 *loss functions*。
注意，对于预测问题，我们有 $Z = \mathcal{X} \times \mathcal{Y}$。然而，我们的损失函数概念已经推广到预测任务之外，因此它允许 Z 是任何示例域（例如，在第22章中描述的无监督学习任务中，Z 不是实例域和标签域的乘积）。

我们现在定义 *risk function* 为一个分类器 $h \in \mathcal{H}$ 关于概率分布 D 在 Z 上的期望损失，即，

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

这是，我们考虑根据 $\mathcal{D}$ 随机选择的对象 z 上 h 的损失期望。同样，我们定义 *empirical risk* 为给定样本 $S = (z_1, \ldots, z_m) \in Z^m$ 上的期望损失，即，

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

分类和回归任务中使用的损失函数如下：

- **0–1 loss:** 这里，我们的随机变量 z 在对 $\mathcal{X} \times \mathcal{Y}$ 的集合上取值，损失函数是

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

  这是los s函数用于二进制或多类分类中 问题。
  应注意的是，对于随机变量 α，其取值为 $\{0, 1\}$，$\mathbb{E}_{\alpha\sim D}[\alpha] = \mathbb{P}_{\alpha\sim D}[\alpha = 1]$。因此，对于此损失函数，方程（3.3）和方程（3.1）中给出的 $L_{\mathcal{D}}(h)$ 的定义是一致的。
- **Square Loss:** 这里，我们的随机变量 z 在对 $\mathcal{X} \times \mathcal{Y}$ 的集合中取值，损失函数是

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

此损失函数用于回归问题。

我们将稍后看到更多有用损失函数实例化的例子。

总结来说，我们正式定义了针对一般损失函数的不可知PAC学习性。

DEFINITION 3.4（针对通用损失函数的不可知PAC学习性）一个假设类$\mathcal{H}$相对于集合Z和损失函数ℓ：$\mathcal{H} \times Z \to \mathbb{R}_+$是可不可知PAC学习的，如果存在一个函数$m_{\mathcal{H}}$：$(0,1)^2 \to \mathbb{N}$和一个具有以下性质的学习算法：对于每一个$\epsilon, \delta \in (0,1)$和对于每一个在$Z$上的分布$\mathcal{D}$，当在由$\mathcal{D}$生成的独立同分布示例$m \geq m_{\mathcal{H}}(\epsilon, \delta)$上运行学习算法时，算法返回$h \in \mathcal{H}$，使得，在至少$1-\delta$ (的概率下，关于选择m训练示例)，

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

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

Remark 3.1 (关于可测性的说明*) 在上述定义中，对于每一个 $h \in \mathcal{H}$，我们将函数 $\ell(h, \cdot) : Z \to \mathbb{R}_+$ 视为一个随机变量，并将 $L_{\mathcal{D}}(h)$ 定义为这个随机变量的期望值。为此，我们需要要求函数 $\ell(h, \cdot)$ 是可测的。形式上，我们假设存在一个 σ-代数，它是 Z 的子集的代数，在此代数上定义了概率 $\mathcal{D}$，并且每个初始段在 $\mathbb{R}_+$ 中的原像都包含在这个 σ-代数中。在具有 0−1 损失的二分类的特殊情况下，σ-代数是在 $\mathcal{X} \times \{0,1\}$ 上，并且我们对 ℓ 的假设等价于对每个 h，集合 $\{(x, h(x)) : x \in \mathcal{X}\}$ 包含在 σ-代数中的假设。

Remark 3.2 (适当的与表示无关的学习*) 在先前的定义中，我们要求算法从 $\mathcal{H}$ 返回一个假设。在某些情况下，$\mathcal{H}$ 是集合 $\mathcal{H}'$ 的子集，损失函数可以自然地扩展为从 $\mathcal{H}' \times Z$ 到实数的函数。在这种情况下，我们可能允许算法返回一个假设 $h' \in \mathcal{H}'$，只要它满足要求 $L_{\mathcal{D}}(h') \leq \min_{h \in \mathcal{H}} L_{\mathcal{D}}(h) + \epsilon$。允许算法从 $\mathcal{H}'$ 输出一个假设称为 *representation independent* 学习，而适当的算法必须从 $\mathcal{H}$ 输出一个假设时发生适当的算法。表示无关学习有时被称为“不适当的学习”，尽管在表示无关学习中没有任何不适当之处。

3.3 Summary

在这一章中，我们定义了我们的主要形式学习模型——PAC学习。基本模型依赖于可实现性假设，而不可知变体则

不对示例下的底层分布施加任何限制。我们还推广了PAC模型到任意损失函数。有时我们将最一般的模型简单地称为PAC学习，省略“无偏见”的前缀，让读者从上下文中推断出底层损失函数是什么。当我们想强调我们正在处理原始PAC设置时，我们会提到可实现性假设成立。在第7章中，我们将讨论其他可学习性的概念。

3.4 Bibliographic Remarks

我们关于具有通用损失函数的通用无监督PAC学习的最一般定义遵循了Vladimir Vapnik和Alexey Chervonenkis（Vapnik & Chervonenkis 1971）的工作。特别是，我们遵循了Vapnik关于学习的通用设置（Vapnik 1982，Vapnik 1992，Vapnik 1995，Vapnik 1998）。

PAC learning 由Valiant（1984年）提出。Valiant因引入PAC模型而获得2010年图灵奖。Valiant的定义要求样本复杂度在1/$/\epsilon$和1/$/\delta$以及类中假设的表示大小上是多项式的（参见Kearns & Vazirani（1994年））。正如我们在第6章中将要看到的，如果一个问题是PAC可学习的，那么样本复杂度将多项式地依赖于1/$/\epsilon$和log(1/$/\delta$)。Valiant的定义还要求学习算法的*runtime*将在这几个量上是多项式的。相比之下，我们选择区分学习的统计方面和计算方面。我们将在第8章中详细阐述计算方面，在那里我们引入了Valiant的完整PAC学习模型。出于说明的目的，即使我们忽略了学习的运行时方面，我们也使用PAC学习这个术语。最后，不可知PAC学习的形式化归功于Haussler（1992年）。

3.5 Exercises

1. **Monotonicity of Sample Complexity:** 设 $\mathcal{H}$ 为二元分类任务的假设类。假设 $\mathcal{H}$ 是 PAC 可学习的，其样本复杂度由 $m_{\mathcal{H}}(\cdot,\cdot)$ 给出。证明 $m_{\mathcal{H}}$ 在其每个参数上单调非增加。也就是说，证明对于给定的 $\delta \in (0,1)$，以及给定的 $0 < \epsilon_1 \leq \epsilon_2 < 1$，我们有 $m_{\mathcal{H}}(\epsilon_1,\delta) \geq m_{\mathcal{H}}(\epsilon_2,\delta)$。同样，证明对于给定的 $\epsilon \in (0,1)$，以及给定的 $0 < \delta_1 \leq \delta_2 < 1$，我们有 $m_{\mathcal{H}}(\epsilon,\delta_1) \geq m_{\mathcal{H}}(\epsilon,\delta_2)$。

2. 令 $\mathcal{X}$ 为一个离散域，并令 $\mathcal{H}_{\text{Singleton}} = \{h_z : z \in \mathcal{X}\} \cup \{h^-\}$，其中对于每个 $z \in \mathcal{X}$，h_z 是由 $h_z(x) =$ 定义的功能，1 如果 $x = z$ 和 $h_z(x) =$，0 如果 $x \neq z$。h^- 简单地是所有负假设，即 $\forall x \in X$，$h^-(x) = 0$。这里的可实现性假设意味着真实假设 f 标记域中所有示例为负，也许除了一个。

1. 描述一个实现可实现在线学习 $\mathcal{H}_{\text{Singleton}}$ 的 ERM 规则的算法。2. 证明 $\mathcal{H}_{\text{Singleton}}$ 是 PAC 可学习的。给出样本复杂度的上界。

3. 设 $\mathcal{X} = \mathbb{R}^2, \mathcal{Y} = \{0, 1\}$，并设 $\mathcal{H}$ 为平面上的同心圆类，即 $\mathcal{H} = \{h_r : r \in \mathbb{R}_+\}$，其中 $h_r(x) = 1_{[\|x\| \leq r]}$。证明 $\mathcal{H}$ 是 PAC 可学习的（假设可实现性），并且其样本复杂度是有界的

$$m_{\mathcal{H}}(\epsilon, \delta) \leq \left\lceil \frac{\log(1/\delta)}{\epsilon} \right\rceil .$$

4. 在这个问题中，我们研究如下定义的假设类 *Boolean conjunctions*。实例空间是 $\mathcal{X} = \{0, 1\}^d$，标签集是 $\mathcal{Y} = \{0, 1\}$。变量 $x_1, \ldots, x_d$ 上的一个文字是一个简单的布尔函数，形式为 $f(\mathbf{x}) = x_i$，其中某些 $i \in [d]$，或者 $f(\mathbf{x}) = 1 - x_i$，其中某些 $i \in [d]$。我们使用符号 $\bar{x}_i$ 作为 $1 - x_i$ 的缩写。合取是文字的任何乘积。在布尔逻辑中，乘积用 $\wedge$ 符号表示。例如，函数 $h(\mathbf{x}) = x_1 \cdot (1 - x_2)$ 写作 $x_1 \wedge \bar{x}_2$ 。

我们考虑所有关于 d 变量的合取命题的假设类。空合取被解释为所有正假设（即返回所有 $\mathbf{x}$ 的 $h(\mathbf{x}) = 1$ 的函数）。合取 $x_1 \wedge \bar{x}_1$ (以及类似地，任何涉及命题及其否定) 的合取都是允许的，并被解释为所有负假设（即返回所有 $\mathbf{x}$ 的 $h(\mathbf{x}) = 0$ 的合取）。我们假设可实现性：即我们假设存在一个布尔合取可以生成标签。因此，每个示例 $(\mathbf{x}, y) \in \mathcal{X} \times \mathcal{Y}$ 由对 d 布尔变量 $x_1, \ldots, x_d$ 的赋值及其真值（假为0，真为1）组成。

例如，设 $d = = 3$，并假设真实联合是 $x_1 \wedge \bar{x}_2$。那么，训练集 S 可能包含以下实例：

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

证明所有关于 d 变量的合取类的假设类是 PAC 可学习的，并对其样本复杂度进行限制。提出一个实现 ERM 规则的算法，其运行时间是 $d \cdot m$ 的多项式。

5. 令 $\mathcal{X}$ 为一个域，令 $\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_m$ 为在 $\mathcal{X}$ 上的分布序列。令 $\mathcal{H}$ 为在 $\mathcal{X}$ 上的有限类二分类器，并令 $f \in \mathcal{H}$。假设我们正在获取 m 个样本的样本 S，其中实例是独立的但不是同分布的；第 i 个实例是从 $\mathcal{D}_i$ 中抽取的，然后 y_i 被设置为 $f(\mathbf{x}_i)$。令 $\bar{D}_m$ 表示平均值，即 $\bar{\mathcal{D}}_m = (\mathcal{D}_1 + \cdots + \mathcal{D}_m)/m$。

修复精度参数 $\epsilon \in (0, 1)$。证明

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

*Hint*使用几何平均不等式。6. 设$\mathcal{H}$为二元分类器的假设类。证明如果$\mathcal{H}$是可识别的PAC可学习的，那么$\mathcal{H}$也是PAC可学习的。此外，如果A是$\mathcal{H}$的可识别PAC学习器，那么A也是$\mathcal{H}$的成功PAC学习器。7.
(*) The Bayes optimal predictor:证明对于每个概率分布$\mathcal{D}$，贝叶斯最优预测器$f_{\mathcal{D}}$是最佳的，即对于从$\mathcal{X}$到$\{0,1\}$的每个分类器g，$L_{\mathcal{D}}(f_{\mathcal{D}}) \leq L_{\mathcal{D}}(g)$。8. **(*)** 我们说，如果一个学习算法$A$ *is better than* B *with respect to*某个概率分布$\mathcal{D}$，那么A *is better than* B *with respect to*是可学习的。

$$L_{\mathcal{D}}(A(S)) \leq L_{\mathcal{D}}(B(S))$$

对于所有样本 $S \in (\mathcal{X} \times \{0,1\})^m$。我们称一个学习算法 A *is better than* B，如果它在所有关于 $\mathcal{X} \times \{0,1\}$ 的概率分布 $\mathcal{D}$ 上优于 B。

1. 一个概率标签预测器是一个函数，它将每个域点 x 分配一个概率值，$h(x) \in [0,1]$，该值确定预测标签 1 的概率。也就是说，给定这样的 h 和一个输入，x，通过抛一个偏向正面的硬币 $h(x)$ 来预测 x 的标签，如果硬币正面朝上则预测 1。形式上，我们定义一个概率标签预测器为一个函数，$h : \mathcal{X} \to [0,1]$。这样的 h 在一个例子 (x, y) 上的损失定义为 $|h(x) - y|$，这正好是预测 h 将不会等于 y 的概率。注意，如果 h 是确定性的，即返回 $\{0,1\}$ 的值，那么 $|h(x) - y| = 1_{[h(x) \neq y]}$。

证明对于每个在 $\mathcal{X} \times \{0,1\}$ 上生成的数据生成分布 $\mathcal{D}$，贝叶斯最优预测器具有最小的风险（相对于损失函数 $\ell(h,(x,y)) = |h(x) - y|$，在所有可能的标签预测器中，包括概率预测器）。2. 设 $\mathcal{X}$ 为一个域，$\{0,1\}$ 为一组标签。证明对于每个在 $\mathcal{X} \times \{0,1\}$ 上的分布 $\mathcal{D}$，存在一个学习算法 $A_{\mathcal{D}}$，它在 $\mathcal{D}$ 方面优于任何其他学习算法。3. 证明对于每个学习算法 A，存在一个概率分布 $\mathcal{D}$ 和一个学习算法 B，使得 A 在 $\mathcal{D}$ 方面不优于 B。

9. 考虑一种PAC模型的变体，其中有两个示例或acles：一个生成正例，一个生成负例，两者都根据底层分布 $\mathcal{D}$ 在 $\mathcal{X}$ 上。形式上，给定一个目标函数 f：$\mathcal{X} \to \{0,1\}$，令 $\mathcal{D}^+$ 是定义在 $\mathcal{X}^+ = \{x \in \mathcal{X} : f(x) = 1\}$ 上的分布 $\mathcal{D}^+(A) = \mathcal{D}(A)/\mathcal{D}(\mathcal{X}^+)$，对于每个 $A \subset \mathcal{X}^+$。同样，$\mathcal{D}^-$ 是由 $\mathcal{D}$ 诱导的 $\mathcal{X}^-$ 上的分布。

PAC学习性在双查询模型中的定义与PAC学习性的标准定义相同，只是在这里，学习器可以访问来自$\mathcal{D}^+$的$m_{\mathcal{H}}^+(\epsilon,\delta)$个独立同分布的示例和来自$\mathcal{D}^-$的$m^-(\epsilon,\delta)$个独立同分布的示例。学习器的目标是输出$h$，使得在至少$1-\delta$ (的概率下，选择

两个训练集，以及可能超过学习算法做出的非确定性决策），$L_{(D^+,f)}(h) \leq \epsilon$ 和$L_{(D-,f)}(h) \leq \epsilon$。

1. (*) 证明如果 $\mathcal{H}$ 是 PAC 可学习的（在标准单查询模型中），那么 $\mathcal{H}$ 在双查询模型中是 PAC 可学习的。2. (**) 定义 h^+ 为始终正假设，h^- 为始终负假设。假设 $h^+, h^- \in \mathcal{H}$。证明如果 $\mathcal{H}$ 在双查询模型中是 PAC 可学习的，那么 $\mathcal{H}$ 在标准单查询模型中是 PAC 可学习的。

4 Learning via Uniform Convergence

第一个我们讨论的正式学习模型是PAC模型。在第2章中，我们证明了在可实现性假设下，任何有限假设类都是PAC可学习的。在本章中，我们将开发一个通用工具*uniform convergence*，并将其应用于证明任何有限类在具有通用损失函数的不可知PAC模型中都是可学习的，只要范围损失函数是有界的。

4.1 Uniform Convergence Is Sufficient for Learnability

本章讨论的学习条件背后的思想非常简单。回想一下，给定一个假设类 $\mathcal{H}$，ERM 学习范式的工作方式如下：在接收到一个训练样本 S 后，学习器评估 $\mathcal{H}$ 中每个 h 在给定样本上的风险（或误差），并输出 $\mathcal{H}$ 中的一个成员，以最小化这种经验风险。希望一个最小化相对于 S 的经验风险的 h 也是一个风险最小化者（或风险接近最小值），相对于真实数据概率分布也是如此。为此，只需确保 $\mathcal{H}$ 中所有成员的经验风险都是其真实风险的良好近似。换句话说，我们需要在假设类中的所有假设上均匀地，经验风险将接近真实风险，如下面所形式化。

DEFINITION 4.1 (ϵ-代表性样本) 如果一个训练集 S 被称为 ϵ-代表性（相对于领域 Z，假设类 $\mathcal{H}$，损失函数 ℓ，和分布 $\mathcal{D}$），则

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

下一个简单引理表明，每当样本是($\epsilon/2$)-代表性时，ERM学习规则保证返回一个良好的假设。

LEMMA 4.2 *Assume that a training set S is $\frac{\epsilon}{2}$-representative (w.r.t. domain Z, hypothesis class $\mathcal{H}$, loss function ℓ, and distribution $\mathcal{D}$). Then, any output of* $\mathrm{ERM}_{\mathcal{H}}(S)$, *namely, any* $h_S \in \operatorname{argmin}_{h \in \mathcal{H}} L_S(h)$, *satisfies*

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

Proof 对于每个 $h \in \mathcal{H}$,

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

在第一个和第三个不等式中，由于假设 S 是 $\frac{\epsilon}{2}$ - 代表性（定义4.1），而第二个不等式成立，因为 h_S 是一个ERM预测器。

□

前述引理表明，为确保ERM规则是一个无偏PAC学习器，只需证明在随机选择训练集的情况下，至少有$1-\delta$ 的概率，它将是一个 ϵ -代表性的训练集。一致收敛条件形式化了这一要求。

DEFINITION 4.3（一致收敛）我们说一个假设类 $\mathcal{H}$ 在一个域 Z 和一个损失函数 ℓ) 上具有 *uniform convergence property* (，如果存在一个函数 $m_{\mathcal{H}}^{\text{UC}} : (0, 1)^2 \to \mathbb{N}$，使得对于每一个 $\epsilon, \delta \in (0, 1)$ 和每一个在 Z 上的概率分布 $\mathcal{D}$，如果 S 是根据 $\mathcal{D}$ 独立同分布抽取的 $m \geq m_{\mathcal{H}}^{\text{UC}}(\epsilon, \delta)$ 个样本，那么，以至少 $1-\delta$ 的概率，S 是 ϵ-代表性的。

类似于PAC学习样本复杂度的定义，函数$m_{\mathcal{H}}^{\text{UC}}$衡量获得一致收敛性质（最小）样本复杂度，即我们需要多少个示例来确保样本以至少$1-\delta$的概率是ϵ-代表性的。

该术语 *uniform* 指的是具有一个适用于 $\mathcal{H}$ 中所有成员以及在整个定义域上所有可能概率分布的固定样本大小。

以下推论直接由引理4.2和一致收敛的定义得出。

COROLLARY 4.4 *If a class $\mathcal{H}$ has the uniform convergence property with a function $m_{\mathcal{H}}^{UC}$ then the class is agnostically PAC learnable with the sample complexity $m_{\mathcal{H}}(\epsilon, \delta) \leq m_{\mathcal{H}}^{UC}(\epsilon/2, \delta)$. Furthermore, in that case, the* $\text{ERM}_{\mathcal{H}}$ *paradigm is a successful agnostic PAC learner for $\mathcal{H}$.*

4.2 Finite Classes Are Agnostic PAC Learnable

考虑到引理4.4，一旦我们证明有限假设类对于一致收敛成立，那么每个有限假设类都是不可知PAC可学习的这一说法将随之而来。

为了证明一致收敛成立，我们采用两步论证，类似于第2章中的推导。第一步应用并集不等式，第二步使用测度集中不等式。现在我们详细解释这两步。

修复一些 ϵ, δ。我们需要找到一个样本大小 m，以保证对于任何 $\mathcal{D}$，至少有 $1-\delta$ 的概率选择 $S = (z_1, \ldots, z_m)$ 样本

独立同分布来自 $\mathcal{D}$，对于所有 $h \in \mathcal{H}$，有 $|L_S(h) - L_\mathcal{D}(h)| \leq \epsilon$。也就是说，

$$\mathcal{D}^m(\{S : \forall h \in \mathcal{H}, |L_S(h) - L_\mathcal{D}(h)| \leq \epsilon\}) \geq 1 - \delta.$$

等效地，我们需要证明

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

写作

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

并且应用并集界（引理2.2），我们得到

$$\mathcal{D}^m(\{S : \exists h \in \mathcal{H}, |L_S(h) - L_\mathcal{D}(h)| > \epsilon\}) \leq \sum_{h \in \mathcal{H}} \mathcal{D}^m(\{S : |L_S(h) - L_\mathcal{D}(h)| > \epsilon\}). \tag{4.1}$$

我们的第二步将是论证这个不等式右侧的每一项都足够小（对于足够大的m）。也就是说，我们将表明，对于任何固定的假设h（在训练集采样之前预先选择），真实风险与经验风险之间的差距$|L_S(h) - L_\mathcal{D}(h)|$很可能是小的。

回忆 $L_\mathcal{D}(h) = \mathbb{E}_{z \sim \mathcal{D}}[\ell(h, z)]$ 以及 $L_S(h) = \frac{1}{m}\sum_{i=1}^m \ell(h, z_i)$。由于每个 z_i 都是从 $\mathcal{D}$ 中独立同分布采样的，随机变量 $\ell(h, z_i)$ 的期望值是 $L_\mathcal{D}(h)$。根据期望的线性性质，可以得出 $L_\mathcal{D}(h)$ 也是 $L_S(h)$ 的期望值。因此，量 $|L_\mathcal{D}(h) - L_S(h)|$ 是随机变量 $L_S(h)$ 与其期望值的偏差。因此，我们需要证明 $L_S(h)$ 在其期望值周围的测度是 *concentrated*。

一个基本的统计事实，*law of large numbers*，表明当m趋向于无穷大时，经验平均值收敛到它们的真实期望。这对于$L_S(h)$是正确的，因为它是一组独立同分布随机变量m的经验平均值。然而，由于大数定律仅是一个渐近结果，它不会提供关于任何给定、有限样本大小下经验估计误差与其真实值之间差距的信息。

相反，我们将使用Hoeffding的测度浓度不等式，该不等式量化了经验平均值与它们的期望值之间的差距。

LEMMA 4.5 (Höfding的不等式) *Let $\theta_1, \ldots, \theta_m$ be a sequence of i.i.d. random variables and assume that for all i, $\mathbb{E}[\theta_i] = \mu$ and $\mathbb{P}[a \leq \theta_i \leq b] = 1$. Then, for any $\epsilon > 0$*

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

证明可以在附录B中找到。

回到我们的问题，设 θ_i 为随机变量 $\ell(h, z_i)$。由于 h 是固定的，$z_1, \ldots, z_m$ 是独立同分布采样的，因此 $\theta_1, \ldots, \theta_m$ 也是独立同分布的随机变量。此外，$L_S(h) = \frac{1}{m}\sum_{i=1}^m \theta_i$ 和 $L_\mathcal{D}(h) = \mu$。让我们

进一步假设 ℓ 的范围为 [0, 1]，因此 $\theta_i \in [0,1]$。因此我们得到

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

将此与方程（4.1）结合得到

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

最后，如果我们选择

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

然后

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

COROLLARY 4.6 *Let $\mathcal{H}$ be a finite hypothesis class, let Z be a domain, and let $\ell : \mathcal{H} \times Z \to [0,1]$ be a loss function. Then, $\mathcal{H}$ enjoys the uniform convergence property with sample complexity*

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

Furthermore, the class is agnostically PAC learnable using the ERM algorithm with sample complexity

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

Remark 4.1（“离散化技巧”）虽然前面的推论仅适用于有限假设类，但有一个简单的技巧可以使我们得到无限假设类实际样本复杂度的一个非常好的估计。考虑一个由d个参数参数化的假设类。例如，设$\mathcal{X} = \mathbb{R}$，$\mathcal{Y} = \{\pm 1\}$，以及假设类$\mathcal{H}$都是形式为$h_\theta(x) = \text{sign}(x - \theta)$的函数。也就是说，每个假设由一个参数$\theta \in \mathbb{R}$参数化，假设对于所有大于$\theta$的实例输出1，对于小于$\theta$的实例输出$-1$。这是一个无限大小的假设类。然而，如果我们打算在实际中用计算机学习这个假设类，我们可能会使用64位浮点表示来维护实数。因此，在实践中，我们的假设类由可以用64位浮点数表示的标量集合参数化。这样的数最多有2^{64}个；因此，我们的假设类的实际大小最多为2^{64}。更一般地，如果我们的假设类由d个数字参数化，在实践中我们学习到的假设类的大小最多为2^{64d}。应用推论4.6，我们得到这样的样本复杂度为

类由 $\frac{128d+2\log(2/\delta)}{\epsilon^2}$ 限制。这个样本复杂度的上限存在一个缺点，即它依赖于我们机器使用的实数特定表示。在第6章中，我们将介绍一种严格分析无限大小假设类样本复杂度的方法。尽管如此，离散化技巧可以用于在许多实际情况下得到样本复杂度的大致估计。

4.3 Summary

如果对于假设类 $\mathcal{H}$ 统一收敛性质成立，那么在大多数情况下，$\mathcal{H}$ 中假设的经验风险将忠实代表它们的真实风险。统一收敛对于使用 ERM 规则的不可知 PAC 学习是充分的。我们已经证明有限假设类具有统一收敛性质，因此是可知 PAC 可学习的。

4.4 Bibliographic Remarks

函数类，其中一致收敛性质成立，也称为Glivenko-Cantelli类，以瓦列里·伊万诺维奇·Glivenko和弗朗切斯科·帕奥洛·Cantelli的名字命名，他们在20世纪30年代证明了第一个一致收敛结果。参见（Dudley, Gine & Zinn 1991）。Vapnik彻底研究了一致收敛与可学习性之间的关系——参见（Vapnik 1992, Vapnik 1995, Vapnik 1998）。实际上，正如我们在第6章中将要看到的，学习理论的基本定理表明，在二分类问题中，一致收敛不仅是可学习性的充分条件，也是必要条件。对于更一般的学习问题并非如此（参见（Shalev-Shwartz, Shamir, Srebro & Sridharan 2010））。

4.5 Exercises

1. 在这个练习中，我们表明，在PAC学习定义中关于误差收敛的 (ϵ, δ) 要求，实际上与一个看起来更简单的关于平均值（或期望）的要求非常接近。证明以下两个陈述对于任何学习算法 A、任何概率分布 $\mathcal{D}$ 和任何取值范围为 $[0, 1]$ 的损失函数是等价的：

1. 对于每个 $\epsilon, \delta > 0$，存在 $m(\epsilon, \delta)$ 使得 $\forall m \geq m(\epsilon, \delta)$

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

2.

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

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

2. **Bounded loss functions:** 在引理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

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

5 The Bias-Complexity Tradeoff

在第二章中，我们看到了如果不小心，训练数据可能会误导学习器，导致过拟合。为了克服这个问题，我们将搜索空间限制在某个假设类 $\mathcal{H}$ 中。这样的假设类可以看作是反映了学习器对任务的先验知识——一个信念，即类 $\mathcal{H}$ 的成员之一是任务的低误差模型。例如，在我们的木瓜口感问题中，基于我们之前对其他水果的经验，我们可能假设颜色-硬度平面上的某个矩形可以预测（至少是近似地）木瓜的口感。

这样的先验知识对于学习的成功真的必要吗？也许存在某种通用学习者，即对某个任务没有任何先验知识并且准备好接受任何任务挑战的学习者？让我们详细阐述这一点。一个特定的学习任务由一个未知的分布 $\mathcal{D}$ 在 $\mathcal{X} \times \mathcal{Y}$ 上定义，其中学习者的目标是找到一个预测器 $h : \mathcal{X} \to \mathcal{Y}$，其风险 $L_{\mathcal{D}}(h)$ 足够小。因此，问题是是否存在一个学习算法 A 和一个训练集大小 m，使得对于每个分布 $\mathcal{D}$，如果 A 从 $\mathcal{D}$ 接收 m 独立同分布的示例，那么它输出一个具有低风险的预测器 h 的可能性很高。

本章的第一部分正式回答了这个问题。无免费午餐定理表明，不存在这样的通用学习器。更精确地说，该定理表明，对于二分类预测任务，对于每个学习器，都存在一个分布，在该分布上它将失败。我们说学习器失败是指，当从该分布接收独立同分布的示例时，其输出的假设很可能具有很大的风险，例如，≥ 0.3，而对于相同的分布，存在另一个学习器将输出具有小风险的假设。换句话说，该定理表明，没有学习器可以在所有可学习任务上成功——每个学习器都有它失败的任务，而其他学习器则在这些任务上成功。

因此，当接近一个由某些分布 $\mathcal{D}$ 定义的学习问题时，我们应该对 $\mathcal{D}$ 有一些先验知识。这类先验知识的一种类型是 $\mathcal{D}$ 来自某些特定的参数分布族。我们将在第24章后面研究在这种假设下的学习。关于 $\mathcal{D}$ 的另一种先验知识，我们在定义 PAC 学习模型时假设了它，即存在 h 在某些预定义的假设类 $\mathcal{H}$ 中，使得 $L_{\mathcal{D}}(h) = 0$。关于 $\mathcal{D}$ 的另一种较软的先验知识是假设 $\min_{h \in \mathcal{H}} L_{\mathcal{D}}(h)$ 很小。在某种意义上，这种对 $\mathcal{D}$ 的较弱假设是使用

无神论PAC模型，其中我们要求输出假设的风险不会比$\min_{h\in\mathcal{H}} L_{\mathcal{D}}(h)$大得多。

在本书的第二部分，我们研究使用假设类作为形式化先验知识手段的益处和弊端。我们将ERM算法在类$\mathcal{H}$上的误差分解为两个组成部分。第一个组成部分反映了我们先验知识的质量，通过我们假设类中假设的最小风险$\min_{h\in\mathcal{H}} L_{\mathcal{D}}(h)$来衡量。这个组成部分也称为*approximation error*，或算法选择$\mathcal{H}$中假设的*bias*。第二个组成部分是过拟合的误差，它取决于类$\mathcal{H}$的大小或复杂性，被称为*estimation error*。这两个术语意味着在选择更复杂的$\mathcal{H}$（（可以减少偏差但增加过拟合的风险））或更简单的$\mathcal{H}$（（可能会增加偏差但减少潜在的过拟合））之间进行权衡。

5.1 The No-Free-Lunch Theorem

在这个部分中，我们证明不存在通用的学习器。我们通过展示没有任何学习器能够在所有学习任务上成功，正如以下定理所形式化的那样：

THEOREM 5.1 (无免费午餐) *Let A be any learning algorithm for the task of binary classification with respect to the $0-1$ loss over a domain $\mathcal{X}$. Let m be any number smaller than $|\mathcal{X}|/2$, representing a training set size. Then, there exists a distribution $\mathcal{D}$ over $\mathcal{X}\times\{0,1\}$ such that:*

1. There exists a function $f:\mathcal{X}\to\{0,1\}$ with $L_{\mathcal{D}}(f)=0$.
2. With probability of at least $1/7$ over the choice of $S\sim\mathcal{D}^m$ we have that $L_{\mathcal{D}}(A(S))\geq 1/8$.

这个定理表明，对于每个学习者，都存在一个任务它无法完成，尽管另一个学习者可以成功学习这个任务。实际上，在这种情况下，一个平凡的成功的学习者将是一个具有假设类 $\mathcal{H}=\{f\}$ 的 ERM 学习者，或者更一般地，是关于包含 f 且其大小满足方程 $m\geq 8\log(7|\mathcal{H}|/6)$ 的任何有限假设类的 ERM（参见推论 2.3）。

Proof 设 C 是 $\mathcal{X}$ 的一个大小为 $2m$ 的子集。证明的直觉是，任何只观察 C 中一半实例的学习算法对 C 中其余实例的标签没有任何信息。因此，存在一个“现实”，即某个目标函数 f，它将违反 $A(S)$ 在 C 中未观察到的实例上预测的标签。

请注意，从 C 到 $\{0,1\}$ 有 $T=2^{2m}$ 个可能的函数。用 $f_1,\ldots,f_T$ 表示这些函数。对于每个这样的函数，令 $\mathcal{D}_i$ 是一个分布。

$C \times \{0,1\}$ 由以下定义

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

这意味着选择一对(x,y)的概率是$1/|C|$，如果标签y确实是根据f_i的真实标签，而如果$y \neq f_i(x)$，则概率为0。显然，$L_{\mathcal{D}_i}(f_i) = 0$。

我们将证明对于每个算法 A，它从 $C \times \{0,1\}$ 接收一个包含 m 个示例的训练集，并返回一个函数 $A(S) : C \to \{0,1\}$，都成立。

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

显然，这意味着对于每个算法 A'，它从 $\mathcal{X} \times \{0,1\}$ 接收一个包含 m 个示例的训练集，存在一个函数 $f : \mathcal{X} \to \{0,1\}$ 和一个在 $\mathcal{X} \times \{0,1\}$ 上的分布 $\mathcal{D}$，使得 $L_{\mathcal{D}}(f) = 0$ 和

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

它很容易验证，前面的内容足以证明 $\mathbb{P}[L_{\mathcal{D}}(A'(S)) \geq 1/8] \geq 1/7$，这正是我们需要证明的（参见练习1）。

我们现在转向证明方程（5.1）成立。从C中m个示例的可能序列有$k = (2m)^m$个。用$S_1, \ldots, S_k$表示这些序列。此外，如果$S_j = (x_1, \ldots, x_m)$，我们用S_j^i表示包含在S_j中并由函数f_i标记的实例的序列，即 $S_j^i = ((x_1, f_i(x_1)), \ldots, (x_m, f_i(x_m)))$。如果分布是$\mathcal{D}_i$，那么可能的训练集$A$可以接收的是$S_1^i, \ldots, S_k^i$，并且所有这些训练集被采样的概率相同。因此，

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

使用“最大”大于“平均”以及“平均”大于“最小”的事实，我们有

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

接下来，修复一些 $j \in [k]$。定义 $S_j = (x_1, \ldots, x_m)$ 并让 $v_1, \ldots, v_p$ 成为 C 中不出现于 S_j 的例子。显然，$p \geq m$。因此，对于每一个

函数 $h : C \to \{0, 1\}$ 以及每个 i 我们都有

$$\begin{aligned} L_{\mathcal{D}_i}(h) &= \frac{1}{2m} \sum_{x \in C} \mathbb{1}_{[h(x) \neq f_i(x)]} \\ &\geq \frac{1}{2m} \sum_{r=1}^{p} \mathbb{1}_{[h(v_r) \neq f_i(v_r)]} \\ &\geq \frac{1}{2p} \sum_{r=1}^{p} \mathbb{1}_{[h(v_r) \neq f_i(v_r)]}. \end{aligned} \tag{5.5}$$

因此，

$$\begin{aligned} \frac{1}{T} \sum_{i=1}^{T} L_{\mathcal{D}_i}(A(S_j^i)) &\geq \frac{1}{T} \sum_{i=1}^{T} \frac{1}{2p} \sum_{r=1}^{p} \mathbb{1}_{[A(S_j^i)(v_r) \neq f_i(v_r)]} \\ &= \frac{1}{2p} \sum_{r=1}^{p} \frac{1}{T} \sum_{i=1}^{T} \mathbb{1}_{[A(S_j^i)(v_r) \neq f_i(v_r)]} \\ &\geq \frac{1}{2} \cdot \min_{r \in [p]} \frac{1}{T} \sum_{i=1}^{T} \mathbb{1}_{[A(S_j^i)(v_r) \neq f_i(v_r)]}. \end{aligned} \tag{5.6}$$

接下来，修复一些 $r \in [p]$。我们可以将 $f_1, \ldots, f_T$ 中的所有函数划分为 $T/2$ 个不相交的对，其中对于一对 $(f_i, f_{i'})$，我们有对于每一个 $c \in C$，$f_i(c) \neq f_{i'}(c)$ 当且仅当 $c = v_r$。由于对于这样的对我们必须有 $S_j^i = S_j^{i'}$，因此可以得出结论：

$$\mathbb{1}_{[A(S_j^i)(v_r) \neq f_i(v_r)]} + \mathbb{1}_{[A(S_j^{i'})(v_r) \neq f_{i'}(v_r)]} = 1,$$

这导致

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

将此与方程（5.6）、方程（5.4）和方程（5.3）相结合，我们得出方程（5.1）成立，从而得出我们的证明结论。 □

5.1.1 No-Free-Lunch and Prior Knowledge

如何将无免费午餐结果与先验知识的需求联系起来？让我们考虑一个在所有从 X 到 $\{0, 1\}$ 的函数 f 类别 $\mathcal{H}$ 上的 ERM 预测器。这个类别代表缺乏先验知识：将域到标签集的每个可能函数都视为一个好的候选者。根据无免费午餐定理，任何从 $\mathcal{H}$ 中的假设中选择输出的算法，特别是 ERM 预测器，都会在某些学习任务上失败。因此，这个类别不是 PAC 可学习的，如以下推论所形式化：

COROLLARY 5.2 *Let $\mathcal{X}$ be an infinite domain set and let $\mathcal{H}$ be the set of all functions from $\mathcal{X}$ to $\{0, 1\}$. Then, $\mathcal{H}$ is not PAC learnable.*

Proof 假设，通过反证法，该类是可学习的。选择一些 $\epsilon < 1/8$ 和 $\delta < 1/7$。根据PAC可学习性的定义，必须存在某种学习算法 A 和一个整数 $m = m(\epsilon, \delta)$，使得对于任何在 $\mathcal{X} \times \{0,1\}$ 上生成的数据生成分布，如果对于某个函数 $f : \mathcal{X} \to \{0,1\}$，$L_{\mathcal{D}}(f) = 0$，那么当 A 应用于由 $\mathcal{D}$ 生成的样本 S 的大小为 m 时，以大于 $1-\delta$ 的概率，$L_{\mathcal{D}}(A(S)) \leq \epsilon$。然而，应用无免费午餐定理，由于 $|\mathcal{X}| > 2m$，对于每个学习算法（特别是对于算法 A），存在一个分布 $\mathcal{D}$，使得以大于 $1/7 > \delta$ 的概率，$L_{\mathcal{D}}(A(S)) > 1/8 > \epsilon$，这导致了所需的矛盾。

□

我们如何防止此类故障？我们可以通过使用我们对特定学习任务的先验知识来规避由无免费午餐定理预见到的危险，以避免在学习该任务时导致我们失败的分发。这种先验知识可以通过限制我们的假设类来表示。

但是我们应该如何选择一个好的假设类呢？一方面，我们希望相信这个类包括没有任何错误的假设（在PAC设置中），或者至少从这个类中得到的假设的最小错误确实相当小（在不可知设置中）。另一方面，我们刚刚看到我们不能简单地选择最丰富的类——给定域上所有函数的类。这种权衡将在下一节中讨论。

5.2 Error Decomposition

为了回答这个问题，我们将ERM$_{\mathcal{H}}$预测器的误差分解为两个组成部分，如下所示。令h_S为一个ERM$_{\mathcal{H}}$假设。然后，我们可以写出

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

- **The Approximation Error** - 预测器在假设类中能够达到的最小风险。这个术语衡量了我们因为将自己限制在特定类别中而承担的风险有多大，即我们有多少*inductive bias*。近似误差不依赖于样本大小，由所选的假设类决定。扩大假设类可以减少近似误差。

  在可实现性假设下，近似误差为零。然而，在不可知情况下，近似误差可能很大.[1]

实际上，它始终包括贝叶斯最优预测器的误差（见第3章），这是最小但不可避免的误差，因为在这个模型中世界可能存在非确定性。有时在文献中，术语*approximation error*并不指min $_{h \in \mathcal{H}} L_{\mathcal{D}}(h)$，而是指超过贝叶斯最优预测器的误差，即min$_{h \in \mathcal{H}} L_{\mathcal{D}}(h) - \epsilon_{\text{Bayes}}$。

- **The Estimation Error** - 近似误差与ERM预测器所达到的误差之间的差异。估计误差产生是因为经验风险（即训练误差）只是真实风险的估计，因此最小化经验风险的预测器只是最小化真实风险的预测器的估计。

  这个估计的质量取决于训练集的大小以及假设类的大小或复杂性。正如我们所展示的，对于有限的假设类，ϵ_{est}随着$|\mathcal{H}|$（对数地）增加，随着m减少。我们可以将$\mathcal{H}$的大小视为其复杂性的一个度量。在未来的章节中，我们将定义假设类的其他复杂性度量。

我们的目标是使总风险最小化，因此我们面临一个权衡，称为*bias-complexity tradeoff*。一方面，选择$\mathcal{H}$为一个非常丰富的类别可以减少近似误差，但同时也可能增加估计误差，因为丰富的$\mathcal{H}$可能导致*overfitting*。另一方面，选择$\mathcal{H}$为一个非常小的集合可以减少估计误差，但可能增加近似误差，换句话说，可能导致*underfitting*。当然，对于$\mathcal{H}$的一个很好的选择是只包含一个分类器——贝叶斯最优分类器的类别。但贝叶斯最优分类器依赖于底层分布$\mathcal{D}$，而我们不知道（实际上，如果我们知道$\mathcal{D}$，学习就不再必要了）。

学习理论研究我们如何在保持合理的估计误差的同时使 $\mathcal{H}$ 变得更加丰富。在许多情况下，实证研究侧重于为某个领域设计良好的假设类。在这里，“良好”意味着那些近似误差不会过高 的类。想法是，尽管我们不是专家，不知道如何构建最优分类器，但我们仍然对所涉及的具体问题有一些先验知识，这使得我们能够设计出近似误差和估计误差都不太大的假设类。回到我们的木瓜例子，我们不知道木瓜的颜色和硬度是如何预测其味道的，但我们知道木瓜是一种水果，基于与其他水果的先前经验，我们推测颜色-硬度空间中的矩形可能是一个好的预测指标。

5.3 Summary

无免费午餐定理表明不存在通用的学习器。每个学习器都必须针对某些任务进行指定，并使用关于该任务的先验知识，才能成功。迄今为止，我们通过限制我们的输出假设属于所选假设类来建模我们的先验知识。在选择这个假设类时，我们面临权衡，即在更大的或更复杂的类（更有可能具有小的近似误差）和更受限制的类（将保证估计误差将

很小。在下一章中，我们将更详细地研究估计误差的行为。在第7章中，我们将讨论表达先验知识的替代方法。

5.4 Bibliographic Remarks

(Wolpert & Macready 1997)证明了几个关于优化的无免费午餐定理，但这些与我们在下面证明的定理有很大不同。我们在下面证明的定理与VC理论中的下界密切相关，正如我们在下一章将要研究的那样。

5.5 Exercises

1. 证明方程（5.2）足以证明 $\mathbb{P}[L_{\mathcal{D}}(A(S)) \geq 1/8] \geq 1/7$。设 θ 为一个在 [0, 1] 中取值的随机变量，其期望满足 $\mathbb{E}[\theta] \geq 1/4$。使用引理B.1来证明 $\mathbb{P}[\theta \geq 1/8] \geq 1/7$。2. 假设你被要求设计一个学习算法来预测患者是否会心脏病发作。算法可能访问的相关患者特征包括血压（BP）、体重指数（BMI）、年龄（A）、身体活动水平（P）和收入（I）。你必须在两个算法之间进行选择；第一个算法在由特征BP和BMI张成的二维空间中选择一个与轴对齐的矩形，而另一个算法在由所有先前特征张成的五维空间中选择一个与轴对齐的矩形。

1. 解释每个选择的优缺点。
2. 解释可用的标记训练样本数量将如何影响你的选择。

3. 证明对于正整数 {v*} 2，如果 $|\mathcal{X}| \geq km$，则我们可以将 No-Free-Lunch 定理中的下界 $1/4$ 替换为 $\frac{k-1}{2k} = \frac{1}{2} - \frac{1}{2k}$。即，设 A 为一个用于二分类任务的算法。设 m 为一个小于 $|\mathcal{X}|/k$ 的任何数，表示训练集的大小。那么，存在一个在 $\mathcal{X} \times \{0, 1\}$ 上的分布 $\mathcal{D}$，使得：

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

6 The VC-Dimension

在上一章中，我们将ERM$_{\mathcal{H}}$规则的误差分解为近似误差和估计误差。近似误差取决于我们的先验知识（如通过假设类$\mathcal{H}$的选择所反映）与潜在未知分布的拟合程度。相比之下，PAC学习性的定义要求估计误差在所有分布上都是有界统一的。

我们的当前目标是弄清楚哪些类 $\mathcal{H}$ 是 PAC 可学习的，并精确地描述学习给定假设类的样本复杂度。到目前为止，我们已经看到有限类是可学习的，但所有函数的类（在无限大小域上）不是。是什么使得一个类可学习而另一个不可学习？无限大小类是否可学习，如果是的话，什么决定了它们的样本复杂度?

我们本章首先表明无限类确实是可以学习的，因此，假设类的有限性不是学习性的必要条件。然后，我们以零一损失的二值分类设置中可学习类族为背景，提出了一个非常清晰的描述。这种描述最初由Vladimir Vapnik和Alexey Chervonenkis在1970年发现，并依赖于一个称为Vapnik-Chervonenkis维数（VC维数）的组合概念。我们正式定义VC维数，提供几个例子，然后陈述统计学习理论的基本定理，该定理综合了学习性、VC维数、ERM规则和一致收敛的概念。

6.1 Infinite-Size Classes Can Be Learnable

第四章中我们看到了有限类是可学习的，实际上，假设类样本复杂度被其大小的对数所上界。为了表明假设类的大小不是其样本复杂度的正确表征，我们首先提出一个可学习的无限大小假设类的简单例子。

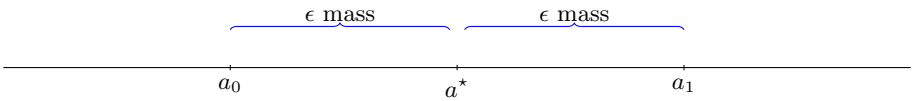
Example 6.1 设 $\mathcal{H}$ 为实线上的阈值函数集合，即 $\mathcal{H} = \{h_a : a \in \mathbb{R}\}$，其中 $h_a : \mathbb{R} \to \{0, 1\}$ 是一个满足 $h_a(x) = 1_{[x<a]}$ 的函数。为了提醒读者，$1_{[x<a]}$ 在 $x < a$ 时为 1，否则为 0。显然，$\mathcal{H}$ 是无限的

大小。尽管如此，以下引理表明，在PAC模型中使用ERM算法可以学习到$\mathcal{H}$。

Lemma 6.1 *Let $\mathcal{H}$ be the class of thresholds as defined earlier. Then, $\mathcal{H}$ is PAC learnable, using the ERM rule, with sample complexity of $m_{\mathcal{H}}(\epsilon, \delta) \leq$ ⌈对(2/δ)/ϵ⌉.取对数*

Proof 设 $a^\star$ 为一个阈值，使得假设 $h^\star(x) = 1_{[x<a^\star]}$ 实现了 $L_{\mathcal{D}}(h^\star) = 0$。设 $\mathcal{D}_x$ 为域 $\mathcal{X}$ 上的边缘分布，并设 $a_0 < a^\star < a_1$ 为以下内容

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



(如果 $\mathcal{D}_x(-\infty, a^\star) \leq \epsilon$ 我们设置 $a_0 = -\infty$，并且对于 a_1) 也是类似。给定一个训练集 S，令 $b_0 = \max\{x : (x, 1) \in S\}$ 和 $b_1 = \min\{x : (x, 0) \in S\}$ (如果 S 中没有正例，我们设置 $b_0 = -\infty$；如果 S 中没有负例，我们设置 $b_1 = \infty$)。令 b_S 为对应于 ERM 假设 h_S 的阈值，这暗示了 $b_S \in (b_0, b_1)$。因此，$L_{\mathcal{D}}(h_S) \leq \epsilon$ 的一个充分条件是 $b_0 \geq a_0$ 和 $b_1 \leq a_1$ 都成立。换句话说，

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

并且使用并集界我们可以将前面的结果限制为

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

事件 $b_0 < a_0$ 发生当且仅当 S 中的所有示例都不在区间 (a_0, a^*) 内，该区间的概率质量定义为 ϵ，即，

$$\mathbb{P}_{S\sim\mathcal{D}^m}[b_0 < a_0] = \mathbb{P}_{S\sim\mathcal{D}^m}[\forall(x, y) \in S,\ x \notin (a_0, a^\star)] = (1-\epsilon)^m \leq e^{-\epsilon m}.$$

自我们假设 $m >$ log(2/δ)/ϵ，因此方程至多为 $\delta/2$。同样，很容易看出 $\mathbb{P}_{S\sim\mathcal{D}^m}[b_1 > a_1] \leq \delta/2$。结合方程（6.1），我们得出证明。

□

6.2 The VC-Dimension

我们因此看到，虽然 $\mathcal{H}$ 的有限性是可学习性的充分条件，但它不是必要条件。正如我们将要展示的，一个称为假设类VC维度的属性给出了其可学习性的正确描述。为了激发VC维度的定义，让我们回顾一下无免费午餐定理（定理5.1）及其证明。在那里，我们已证明，如果没有

限制假设类，对于任何学习算法，攻击者可以构造一个分布，使得学习算法表现不佳，而另一个学习算法在相同的分布上会成功。为此，攻击者使用了一个有限集 $C \subset \mathcal{X}$，并考虑了一个集中在 C 元素上的分布族。每个分布都来自从 C 到 $\{0,1\}$ 的“真实”目标函数。为了使任何算法失败，攻击者利用了从 C 到 $\{0,1\}$ 的 all 个可能函数集中选择目标函数的能力。

当考虑假设类 $\mathcal{H}$ 的 PAC 可学习性时，对手被限制在构建某些假设 $h \in \mathcal{H}$ 实现零风险的分布。由于我们正在考虑集中在 C 元素上的分布，我们应该研究 $\mathcal{H}$ 在 C 上的行为，这导致以下定义。

DEFINITION 6.2 （$\mathcal{H}$ 到 C 的限制）设 $\mathcal{H}$ 是从 $\mathcal{X}$ 到 $\{0,1\}$ 的函数类，并设 $C = \{c_1, \ldots, c_m\} \subset \mathcal{X}$。$\mathcal{H}$ 到 C 的限制是从 C 到 $\{0,1\}$ 的函数集，这些函数可以由 $\mathcal{H}$ 推导出来。即，

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

在C到$\{0,1\}$的每个函数表示为$\{0,1\}^{|C|}$中的向量。

如果将 $\mathcal{H}$ 限制为 C 的集合是所有从 C 到 $\{0,1\}$ 的函数的集合，那么我们称 $\mathcal{H}$ $shatters$ 为集合 C。形式上：

DEFINITION 6.3（破碎）一个假设类 $\mathcal{H}$ 破碎一个有限集 $C \subset \mathcal{X}$，如果 $\mathcal{H}$ 在 C 上的限制是所有从 C 到 $\{0,1\}$ 的函数集。也就是说，$|\mathcal{H}_C| = 2^{|C|}$。

Example 6.2 设 $\mathcal{H}$ 为 $\mathbb{R}$ 上的阈值函数类。取一个集合 $C = \{c_1\}$。现在，如果我们取 $a = c_1 +$ 为 1，那么我们就有 $h_a(c_1) =$ 为 1，如果我们取 $a = c_1 -$ 为 1，那么我们就有 $h_a(c_1) =$ 为 0。因此，$\mathcal{H}_C$ 是从 C 到 $\{0,1\}$ 的所有函数的集合，并且 $\mathcal{H}$ 粉碎了 C。现在取一个集合 $C = \{c_1, c_2\}$，其中 $c_1 \leq c_2$。没有任何 $h \in \mathcal{H}$ 可以解释标签（0，1），因为任何将标签 0 分配给 c_1 的阈值必须也将标签 0 分配给 c_2。因此，不是所有从 C 到 $\{0,1\}$ 的函数都包含在 $\mathcal{H}_C$ 中；因此 C 不是由 $\mathcal{H}$ 粉碎的。

回到对抗分布的构建，正如在无免费午餐定理（定理5.1）的证明中，我们看到每当某个集合C被$\mathcal{H}$划分时，对手不受$\mathcal{H}$的限制，因为他们可以根据C到$\{0,1\}$的any目标函数构建C上的分布，同时仍然保持可实现性假设。这立即得出：

COROLLARY 6.4 *Let $\mathcal{H}$ be a hypothesis class of functions from $\mathcal{X}$ to $\{0,1\}$. Let m be a training set size. Assume that there exists a set $C \subset \mathcal{X}$ of size $2m$ that is shattered by $\mathcal{H}$. Then, for any learning algorithm, A, there exist a distribution $\mathcal{D}$ over $\mathcal{X} \times \{0,1\}$ and a predictor $h \in \mathcal{H}$ such that $L_{\mathcal{D}}(h) = 0$ but with probability of at least $1/7$ over the choice of $S \sim \mathcal{D}^m$ we have that $L_{\mathcal{D}}(A(S)) \geq 1/8$.*

推论6.4告诉我们，如果$\mathcal{H}$将大小为$2m$的集合C打碎，那么我们无法使用m个示例来学习$\mathcal{H}$。直观上，如果一个集合C被$\mathcal{H}$打碎，并且我们收到一个包含C一半实例的样本，这些实例的标签给我们关于C中其余实例标签的信息——其余实例的每一种可能的标签都可以由$\mathcal{H}$中的某个假设来解释。从哲学上讲，

If someone can explain every phenomenon, his explanations are worthless.

这直接引出了VC维的定义。

DEFINITION 6.5 (VC维) 假设类$\mathcal{H}$的VC维，记作VCdim($\mathcal{H}$)，是可以被$\mathcal{H}$打碎的最大集合$C \subset \mathcal{X}$的大小。如果$\mathcal{H}$可以打碎任意大型的集合，我们说$\mathcal{H}$具有无限VC维。

因此，命题6.4的直接后果是：

THEOREM 6.6 *Let $\mathcal{H}$ be a class of infinite VC-dimension. Then, $\mathcal{H}$ is not PAC learnable.*

Proof 由于 $\mathcal{H}$ 具有无穷大的 VC 维度，对于任何训练集大小 m，都存在一个大小为 $2m$ 的破碎集，根据推论 6.4，结论成立。 □

我们将在本章后面看到，逆命题也是正确的：有限的VC维保证了可学习性。因此，VC维表征了PAC可学习性。但在深入理论之前，我们首先展示几个例子。

6.3 Examples

在这一节中，我们计算几个假设类的VC维。为了证明VCdim($\mathcal{H}$) = d，我们需要证明

1. 存在一个大小为 d 的集合 C，它被 $\mathcal{H}$ 破碎。2. 每个大小为 $d+$ 的集合 C 1 都不被 $\mathcal{H}$ 破碎。

6.3.1 Threshold Functions

设 $\mathcal{H}$ 为 $\mathbb{R}$ 上的阈值函数类。回忆第 6.2 节的例子，我们已证明对于任意集合 $C = \{c_1\}$，$\mathcal{H}$ 粉碎 C；因此 VCdim($\mathcal{H}$) $\geq$ 1。我们还证明了对于任意集合 $C = \{c_1, c_2\}$，其中 $c_1 \leq c_2$，$\mathcal{H}$ 不粉碎 C。因此我们得出结论，VCdim($\mathcal{H}$) = 1。

6.3.2 Intervals

设 $\mathcal{H}$ 为 $\mathbb{R}$ 上的区间类，即 $\mathcal{H} = \{h_{a,b} : a, b \in \mathbb{R}, a < b\}$，其中 $h_{a,b} : \mathbb{R} \to \{0, 1\}$ 是一个满足 $h_{a,b}(x) = 1_{[x\in(a,b)]}$ 的函数。取集合 $C = \{1, 2\}$。然后，$\mathcal{H}$ 切分 C（确保你理解为什么）因此 VCdim($\mathcal{H}$) ≥ 2。现在取一个任意的集合 $C = \{c_1, c_2, c_3\}$ 并假设不失一般性 $c_1 \leq c_2 \leq c_3$。然后，标签 (1, 0, 1) 不能通过一个区间获得，因此 $\mathcal{H}$ 不切分 C。因此，我们得出结论 VCdim($\mathcal{H}$) $= 2$。

6.3.3 Axis Aligned Rectangles

让 $\mathcal{H}$ 表示轴对齐矩形的类，形式上：

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

哪里

$$h_{(a_1,a_2,b_1,b_2)}(x_1, x_2) = \begin{cases} 1 & \text{if } a_1 \leq x_1 \leq a_2 \text{ and } b_1 \leq x_2 \leq b_2 \\ 0 & \text{otherwise} \end{cases} \tag{6.2}$$

我们将以下展示VCdim($\mathcal{H}$) $= 4$。为了证明这一点，我们需要找到一个由$\mathcal{H}$打散的4点集，并展示没有5点集可以被$\mathcal{H}$打散。找到一个打散的4点集是容易的（参见图6.1）。现在，考虑任何5点集$C \subset \mathbb{R}^2$。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（第一个坐标最大），一个最低点（第二个坐标最小），以及一个最高点（第二个坐标最大）。不失一般性，记 $C = \{c_1, \ldots, c_5\}$，并让c_5为未被选中的点。现在，定义标签（1, 1, 1, 1, 0）。通过一个轴对齐的矩形无法获得这种标签。实际上，这样的矩形必须包含 $c_1, \ldots, c_4$；但在这个情况下，矩形还包含c_5，因为其坐标在由选定点定义的区间内。因此，C没有被H打散，因此VCdim(H) $= 4$。

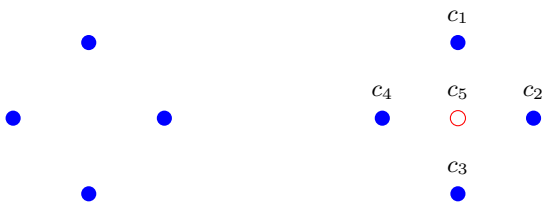


Figure 6.1 左侧：被轴对齐矩形分割的4个点。右侧：任何轴对齐矩形都不能用0标记c_5，其余点用1标记。

6.3.4 Finite Classes

设 $\mathcal{H}$ 为一个有限类。显然，对于任何集合 C，我们有 $|\mathcal{H}_C| \leq |\mathcal{H}|$，因此如果 $|\mathcal{H}| < 2^{|C|}$，则 C 不能被打碎。这表明 VCdim($\mathcal{H}$) $\leq \log_2(|\mathcal{H}|)$。这表明有限类的 PAC 可学习性源于具有有限 VC 维度的类的更一般 PAC 可学习性陈述，我们将在下一节中看到。然而，请注意，有限类 $\mathcal{H}$ 的 VC 维度可以显著小于 $\log_2(|\mathcal{H}|)$。例如，设 $\mathcal{X} = \{1, \ldots, k\}$，对于某个整数 k，考虑阈值函数类（如例 6.2 中定义的）。那么，$|\mathcal{H}| = k$ 但 VCdim($\mathcal{H}$) = 1。由于 k 可以任意大，$\log_2(|\mathcal{H}|)$ 和 VCdim($\mathcal{H}$) 之间的差距可以任意大。

6.3.5 VC-Dimension and the Number of Parameters

在先前的例子中，VC维恰好等于定义假设类参数的数量。虽然这通常是情况，但并不总是如此。考虑，例如，域$\mathcal{X} = \mathbb{R}$和假设类$\mathcal{H} = \{h_\theta \colon \theta \in \mathbb{R}\}$，其中$h_\theta \colon \mathcal{X} \to \{0,1\}$由$h_\theta(x) = \lceil 0.5 \sin(\theta x) \rceil$定义。可以证明VCdim($\mathcal{H}$) $= \infty$，即对于 *every* d，可以找到d个点被$\mathcal{H}$ (shattered by)（见练习8)）。

6.4 The Fundamental Theorem of PAC learning

我们已经证明了无限VC维度的某一类是不可学习的。逆命题也成立，从而导致了统计学习理论的基本定理：

THEOREM 6.7（统计学习的基本定理）*Let $\mathcal{H}$ be a hypothesis class of functions from a domain $\mathcal{X}$ to $\{0, 1\}$ and let the loss function be the $0-1$ loss. Then, the following are equivalent:*

1. *$\mathcal{H}$ has the uniform convergence property.*
2. *Any ERM rule is a successful agnostic PAC learner for $\mathcal{H}$.*
3. *$\mathcal{H}$ is agnostic PAC learnable.*
4. *$\mathcal{H}$ is PAC learnable.*
5. *Any ERM rule is a successful PAC learner for $\mathcal{H}$.*
6. *$\mathcal{H}$ has a finite VC-dimension.*

定理的证明在下一节给出。
不仅VC维数表征了PAC学习性；它甚至决定了样本复杂度。

THEOREM 6.8（统计学习的基本定理 - 定量版本）
Let $\mathcal{H}$ be a hypothesis class of functions from a domain $\mathcal{X}$ to $\{0, 1\}$ and let the loss function be the $0-1$ loss. Assume that VCdim($\mathcal{H}$) $= d < \infty$. *Then, there are absolute constants C_1, C_2 such that:*

1. $\mathcal{H}$ has the uniform convergence property with sample complexity

$$C_1 \frac{d + \log(1/\delta)}{\epsilon^2} \leq m^{UC}_{\mathcal{H}}(\epsilon, \delta) \leq C_2 \frac{d + \log(1/\delta)}{\epsilon^2}$$

2. $\mathcal{H}$ is agnostic PAC learnable with sample complexity

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

3. $\mathcal{H}$ is PAC learnable with sample complexity

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

该定理的证明在第28章给出。

Remark 6.3 我们提出了二元分类任务的根本定理。对于一些其他学习问题，如具有绝对损失或平方损失的回归，也有类似的结果。然而，该定理并不适用于所有学习任务。特别是，即使均匀收敛性质不成立，有时也可能实现可学习性（我们将在第13章习题2中看到一个例子）。此外，在某些情况下，ERM规则失败，但使用其他学习规则仍可实现可学习性。

6.5 Proof of Theorem 6.7

我们已经看到，在第4章中1 → 2。2 → 3和3 → 4的推论是平凡的，2 → 5也是如此。4 → 6和5 → 6的推论来自无免费午餐定理。困难的部分是证明6 → 1。证明基于两个主要论断：

- 如果 VCdim($\mathcal{H}$) = d，那么即使 $\mathcal{H}$ 可能是无限的，当将其限制为有限集合 $C \subset \mathcal{X}$时，其“有效”大小，$|\mathcal{H}_C|$，也只有 $O(|C|^d)$。也就是说，$\mathcal{H}_C$ 的大小以多项式而不是指数方式随着 $|C|$ 增长。这个论断通常被称为 *Sauer's lemma*，但它也被 Shelah 和 Perles 独立地陈述和证明。正式陈述在后面的第 6.5.1 节给出。
- 在第四章中，我们已证明有限假设类具有一致收敛性质。在后续的第六章6.5.2节中，我们推广了这一结果，并表明当假设类具有“小的有效规模”时，一致收敛始终成立。我们所说的“小的有效规模”是指那些$|\mathcal{H}_C|$随着$|C|$以多项式增长的门类。

6.5.1 Sauer's Lemma and the Growth Function

我们定义了*shattering*的概念，通过考虑$\mathcal{H}$对有限实例集的限制。增长函数衡量在m示例集上$\mathcal{H}$的最大“有效”大小。形式上：

DEFINITION 6.9（增长函数）设 $\mathcal{H}$ 为一个假设类。那么 $\mathcal{H}$ 的 *growth function*，记为 $\tau_{\mathcal{H}}$：$\mathbb{N} \to \mathbb{N}$，定义为

$$\tau_{\mathcal{H}}(m) \;=\; \max_{C \subset \mathcal{X}:|C|=m} \left|\mathcal{H}_C\right| .$$

用文字来说，$\tau_H(m)$ 是从大小为 m 的集合 C 到 $\{0,1\}$ 的不同函数的数量，这些函数可以通过将 $\mathcal{H}$ 限制为 C 来获得。

显然，如果VCdim($\mathcal{H}$) $= d$，那么对于任何 $m \leq d$，我们都有 $\tau_{\mathcal{H}}(m) = 2^m$。在这种情况下，$\mathcal{H}$ 诱导从 C 到 $\{0,1\}$ 的所有可能函数。以下由Sauer、Shelah和Perles独立提出的美丽引理表明，当 m 大于VC维度时，增长函数随着 m 的增加而不是指数增长而是多项式增长。

LEMMA 6.10 (Sauer-Shelah-Perles) *Let $\mathcal{H}$ be a hypothesis class with* VCdim($\mathcal{H}$) $\leq d < \infty$. *Then, for all m, $\tau_{\mathcal{H}}(m) \leq \sum_{i=0}^{d} \binom{m}{i}$. In particular, if $m > d+1$ then $\tau_{\mathcal{H}}(m) \leq (em/d)^d$.*

Proof of Sauer's Lemma *

为了证明引理，只需证明以下更强的命题：对于任意的 $C = \{c_1, \ldots, c_m\}$，我们有

$$\forall \mathcal{H}, \quad |\mathcal{H}_C| \;\leq\; |\{B \subseteq C : \mathcal{H} \text{ shatters } B\}|. \tag{6.3}$$

原因是方程（6.3）足以证明引理，因为如果VCdim($\mathcal{H}$) $\leq d$，那么没有大小超过d的集合能被$\mathcal{H}$打散，因此

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

当 $m > d+1$ 前一个表达式的右侧最多为 $(em/d)^d$（参见附录A中的引理A.5）。

我们剩下证明方程（6.3）的工作，我们使用归纳法来证明它。对于 $m = 1$，无论 $\mathcal{H}$ 是什么，方程（6.3）的两边要么都等于1，要么都等于2（空集总是被认为被 $\mathcal{H}$ 打碎）。假设方程（6.3）对于大小为 $k < m$ 的集合成立，让我们证明它对于大小为 m 的集合也成立。固定 $\mathcal{H}$ 和 $C = \{c_1, \ldots, c_m\}$。表示 $C' = \{c_2, \ldots, c_m\}$，此外，定义以下两个集合：

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

和

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

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

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

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

$$\begin{aligned}\mathcal{H}' &= \{h \in \mathcal{H} : \exists h' \in \mathcal{H} \text{ s.t. } (1 - h'(c_1), h'(c_2), \ldots, h'(c_m)) \\ &= (h(c_1), h(c_2), \ldots, h(c_m))\},\end{aligned}$$

即，$\mathcal{H}'$ 包含在 C' 上达成一致而在 c_1 上存在差异的假设对。根据这个定义，如果 $\mathcal{H}'$ 切割集合 $B \subseteq C'$，那么它也切割集合 $B \cup \{c_1\}$，反之亦然。结合这一事实以及使用归纳假设（现在应用于 $\mathcal{H}'$ 和 C'），我们得到:

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

总体而言，我们已经证明

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

这总结了我们的证明。

6.5.2 Uniform Convergence for Classes of Small Effective Size

在这一节中，我们证明如果 $\mathcal{H}$ 的有效尺寸较小，那么它具有一致收敛性质。形式上，

THEOREM 6.11 *Let $\mathcal{H}$ be a class and let $\tau_{\mathcal{H}}$ be its growth function. Then, for every $\mathcal{D}$ and every $\delta \in (0,1)$, with probability of at least $1-\delta$ over the choice of $S \sim \mathcal{D}^m$ we have*

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

在证明定理之前，让我们首先得出定理6.7的证明。

Proof of Theorem 6.7 只需证明如果VC维是有限的，则一致收敛性质成立。我们将证明

$$m^{\text{UC}}_{\mathcal{H}}(\epsilon, \delta) \leq 4\,\frac{16d}{(\delta\epsilon)^2}\,\log\left(\frac{16d}{(\delta\epsilon)^2}\right) + \frac{16\,d\log(2e/d)}{(\delta\epsilon)^2}.$$

从Sauer引理中，我们有对于$m > d$，$\tau_{\mathcal{H}}(2m) \leq (2em/d)^d$。结合定理6.11，我们得到至少以$1-\delta$的概率，

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

为了简单起见，假设 $\sqrt{d\log(2em/d)} \geq 4$；因此，

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

为确保前面的内容不超过 ϵ，我们需要

$$m \geq \frac{2d\log(m)}{(\delta\epsilon)^2} + \frac{2\,d\log(2e/d)}{(\delta\epsilon)^2}.$$

标准代数运算（参见附录A中的引理A.2）表明，上述条件成立的一个充分条件是

$$m \geq 4\,\frac{2d}{(\delta\epsilon)^2}\,\log\left(\frac{2d}{(\delta\epsilon)^2}\right) + \frac{4\,d\log(2e/d)}{(\delta\epsilon)^2}.$$

□

Remark 6.4 在证明定理6.7中我们推导出的 $m_{\mathcal{H}}^{\mathrm{UC}}$ 的上界不是最紧的。一个更紧的分析，可以得到定理6.8中给出的界限，可以在第28章中找到。

Proof of Theorem 6.11 *

我们将首先展示

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

由于随机变量 $\sup_{h\in\mathcal{H}}|L_{\mathcal{D}}(h) - L_S(h)|$ 是非负的，定理的证明可以直接从前面的内容中得出，使用马尔可夫不等式（见B.1节）。

要界定方程（6.4）的左侧，我们首先注意到对于每一个 $h \in \mathcal{H}$，我们可以重写 $L_{\mathcal{D}}(h) = \mathbb{E}_{S'\sim\mathcal{D}^m}[L_{S'}(h)]$，其中 $S' = z'_1, \ldots, z'_m$ 是一个额外的独立同分布样本。因此，

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

一个三角形不等式的推广得出

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

并且超母期望小于上确界期望的事实得出

$$\sup_{h\in\mathcal{H}}\mathop{\mathbb{E}}_{S'\sim\mathcal{D}^m}|L_{S'}(h) - L_S(h)| \leq \mathop{\mathbb{E}}_{S'\sim\mathcal{D}^m}\sup_{h\in\mathcal{H}}|L_{S'}(h) - L_S(h)|.$$

形式上，前两个不等式可由 Jensen 不等式得出。将所有这些结合起来，我们得到

$$\begin{aligned}\mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}\left[\sup_{h\in\mathcal{H}}|L_{\mathcal{D}}(h) - L_S(h)|\right] &\leq \mathop{\mathbb{E}}_{S,S'\sim\mathcal{D}^m}\left[\sup_{h\in\mathcal{H}}|L_{S'}(h) - L_S(h)|\right] \\ &= \mathop{\mathbb{E}}_{S,S'\sim\mathcal{D}^m}\left[\sup_{h\in\mathcal{H}}\frac{1}{m}\left|\sum_{i=1}^{m}(\ell(h,z'_i) - \ell(h,z_i))\right|\right].\end{aligned} \tag{6.5}$$

右侧的期望是在两个独立同分布样本 $S = z_1, \ldots, z_m$ 和 $S' = z'_1, \ldots, z'_m$ 的选择上。由于所有这些 $2m$ 向量都是独立同分布选择的，如果我们用随机向量 z'_i 的名称替换随机向量 z_i 的名称，那么什么都不会改变。如果我们这样做，那么在方程（6.5）中的项 $(\ell(h, z'_i) - \ell(h, z_i))$ 将被项 $-(\ell(h, z'_i) - \ell(h, z_i))$ 替换。因此，对于每个 $\boldsymbol{\sigma} \in \{\pm 1\}^m$，我们有方程（6.5）等于

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

因此，对于每个 $\boldsymbol{\sigma} \in \{\pm 1\}^m$ 都成立，如果我们从 $\{\pm 1\}$ 上的均匀分布中随机均匀采样 $\boldsymbol{\sigma}$ 的每个分量，记为 $U_{\pm}$，这也成立。因此，方程（6.5）也等于

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

并且根据期望的线性性质，它也等于

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

接下来，固定 S 和 S'，并令 C 为出现在 S 和 S' 中的实例。然后，我们只需在 $h \in \mathcal{H}_C$ 上取上确界。因此，

$$\begin{aligned} &\mathop{\mathbb{E}}_{\boldsymbol{\sigma} \sim U_{\pm}^m} \left[\sup_{h \in \mathcal{H}} \frac{1}{m} \left| \sum_{i=1}^{m} \sigma_i (\ell(h, z'_i) - \ell(h, z_i)) \right| \right] \\ &= \mathop{\mathbb{E}}_{\boldsymbol{\sigma} \sim U_{\pm}^m} \left[\max_{h \in \mathcal{H}_C} \frac{1}{m} \left| \sum_{i=1}^{m} \sigma_i (\ell(h, z'_i) - \ell(h, z_i)) \right| \right]. \end{aligned}$$

修复一些 $h \in \mathcal{H}_C$ 并表示 $\theta_h = \frac{1}{m} \sum_{i=1}^{m} \sigma_i (\ell(h, z'_i) - \ell(h, z_i))$。由于 $\mathbb{E}[\theta_h] = 0$ 且 θ_h 是独立变量的平均值，每个变量在 $[-1, 1]$ 中取值，根据 Hoeffding 不等式，对于每个 $\rho > 0$，我们有

$$\mathbb{P}[|\theta_h| > \rho] \leq 2 \exp\left(-2\, m\, \rho^2\right).$$

应用i 对于 $h \in \mathcal{H}_C$ 的并集界，我们得到对于任何 $\rho > 0$,

$$\mathbb{P}\left[\max_{h \in \mathcal{H}_C} |\theta_h| > \rho \right] \leq 2\, |\mathcal{H}_C| \exp\left(-2\, m\, \rho^2\right).$$

最后，附录A中的引理A.4告诉我们，前面的内容意味着

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

C将所有与$\tau_{\mathcal{H}}$的定义结合，我们已经证明t 帽子

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

6.6 Summary

学习理论的基本定理使用VC维来表征二分类器类别的PAC可学习性。一个类别的VC维是其组合性质，表示该类别能够打碎的最大样本大小。基本定理表明，一个类别是PAC可学习的当且仅当其VC维是有限的，并指定了PAC学习所需的样本复杂度。该定理还表明，如果一个问题在任何情况下都是可学习的，那么一致收敛成立，因此可以使用ERM规则来学习该问题。

6.7 Bibliographic remarks

VC维度的定义及其与学习能力和一致收敛的关系归功于Vapnik & Chervonenkis（1971）的开创性工作。与PAC学习性定义的关系归功于Blumer、Ehrenfeucht、Haussler和Warmuth（1989）。

已提出VC维度的几个推广。例如，肥大散度维度表征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），而Natarajan维度表征了一些多类学习问题的可学习性（Natarajan 1989）。然而，在一般情况下，可学习性与一致收敛性之间没有等价性。参见（Shalev-Shwartz, Shamir, Srebro & Sridharan 2010，Daniely, Sabato, Ben-David & Shalev-Shwartz 2011）。

Sauer的引理是由Sauer在回应Erdos（Sauer 1972）的一个问题后证明的。Shelah（与Perles）将其证明为一个对Shelah的稳定模型理论有用的引理（Shelah 1972）。Gil Kalai告诉我们，在某个稍后的时候，Benjy Weiss在遍历理论的背景下询问Perles关于这样的结果，Perles忘记了他曾经证明过这个引理，于是再次证明了它。Vapnik和Chervonenkis在统计学习理论的背景下证明了该引理。

6.8 Exercises

1. 展示VC维度的单调性属性：对于任意两个假设类，如果 $\mathcal{H}' \subseteq \mathcal{H}$，则 VCdim($\mathcal{H}'$) $\leq$ VCdim($\mathcal{H}$). 2. 给定一个有限的域集 $\mathcal{X}$ 和一个数 $k \leq |\mathcal{X}|$，找出以下每个类的VC维度（并证明你的结论）： 1. $\mathcal{H}^{\mathcal{X}}_{=k} = \{h \in \{0,1\}^{\mathcal{X}} : |\{x : h(x) = 1\}| = k\}$。即，所有将值1分配给 $\mathcal{X}$ 中恰好 k 个元素的函数的集合。

[1] http://gilkalai.wordpress.com/2008/09/28/extremal-combinatorics-iii-some-basic-theorems

2. $\mathcal{H}_{at-most-k} = \{h \in \{0,1\}^{\mathcal{X}} : |\{x : h(x) = 1\}| \leq k$ 或 $|\{x : h(x) = 0\}| \leq k\}$ 。3. 令 $\mathcal{X}$ 为布尔超立方体 $\{0,1\}^n$。对于集合 $I \subseteq \{1, 2, \ldots, n\}$，我们定义 *parity function* h_I 如下。在一个二进制向量 $\mathbf{x} = (x_1, x_2, \ldots, x_n) \in \{0,1\}^n$，

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

(这是，h_I计算I中位的奇偶性.)所有这样的奇偶函数类$\mathcal{H}_{n\text{-parity}} = \{h_I \colon I \subseteq \{1, 2, \ldots, n\}\}$的VC维是多少?

4. 我们通过证明对于每个有限VC维度的类 $\mathcal{H}$ 和域的每个子集 A，证明了Sauer引理。

$$|\mathcal{H}_A| \leq |\{B \subseteq A \ : \ \mathcal{H} \text{ shatters } B\}| \leq \sum_{i=0}^{d} \binom{|A|}{i} \, .$$

证明存在两种情况，前两个不等式是严格的（即，$\leq$ 可以被 $<$ 替换），以及它们可以被等式替换的情况。演示 $=$ 和 $<$ 的所有四种组合。

5. **VC-dimension of axis aligned rectangles in $\mathbb{R}^d$:** 设 $\mathcal{H}^d_{\text{rec}}$ 为 $\mathbb{R}^d$ 中轴对齐矩形的类。我们已经看到 VCdim($\mathcal{H}^2_{\text{rec}}$) = 4。证明在一般情况下，VCdim($\mathcal{H}^d_{\text{rec}}$) $= 2d$。

6. **VC-dimension of Boolean conjunctions:** 设 $\mathcal{H}^d_{con}$ 为变量 $x_1, \ldots, x_d$ ($d \geq 2$) 上的布尔合取类。我们已知这个类是有限的，因此是（无知的）PAC可学习的。在这个问题中，我们计算 VCdim($\mathcal{H}^d_{con}$)。

1. 证明 $|\mathcal{H}^d_{con}| \leq 3^d + 1$. 2. 推断 VCdim($\mathcal{H}$) $\leq d$ log 3. 3. 证明 $\mathcal{H}^d_{con}$ 将单位向量集 $\{\mathbf{e}_i : i \leq d\}$ 打碎。4. (**) 证明 VCdim($\mathcal{H}^d_{con}$) $\leq d$。提示：假设通过反证法存在一个集合 $C = \{c_1, \ldots, c_{d+1}$

}

这是被 $\mathcal{H}^d_{con}$ 破碎的。设 $h_1, \ldots, h_{d+1}$ 为 $\mathcal{H}^d_{con}$ 中的假设。 t
满足

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

对于每个 $i \in [d+1]$，h_i (或者更准确地说，与 h_i) 对应的合取包含一些字面 ℓ_i，它在 c_i 上为假，在 c_j 上为真，对于每个 $j \neq i$。使用鸽巢原理来证明必须存在一对 $i < j \leq d+1$，使得 ℓ_i 和 ℓ_j 使用相同的 x_k，并利用这一事实从合取 h_i, h_j 中推导出矛盾。

5. 考虑在 $\{0,1\}^d$ 上的单调布尔合取类 $\mathcal{H}^d_{mcon}$。这里的单调性意味着合取式不包含否定。

与 $\mathcal{H}^d_{con}$ 中一样，空合取被解释为全正假设。我们用全负假设 h^- 增强了 $\mathcal{H}^d_{mcon}$。证明 VCdim($\mathcal{H}^d_{mcon}$) = d。

7. 我们已经证明，对于有限假设类 $\mathcal{H}$，VCdim($\mathcal{H}$) $\leqslant \leq \lfloor \log(|\mathcal{H}|) \rfloor$。然而，这只是一个上界。一个类 的VC维可以远低于这个值：

1. 找到一个在实数区间 $\mathcal{X} = [0, 1]$ 上的函数类 $\mathcal{H}$ 的例子，使得 $\mathcal{H}$ 是无限的，而 VCdim($\mathcal{H}$) = 1。2. 给出一个在域 $\mathcal{X} = [0, 1]$ 上的有限假设类 $\mathcal{H}$ 的例子，其中 VCdim($\mathcal{H}$) = $\lfloor \log_2(|\mathcal{H}|) \rfloor$。

8. (*) 通常情况下，假设类中的VC维数等于（或可以由）定义该类中每个假设所需的参数数量。例如，如果 $\mathcal{H}$ 是 $\mathbb{R}^d$ 中轴对齐矩形的类，那么 VCdim($\mathcal{H}$) = 2d，这等于定义矩形所使用的参数数量。这里有一个例子表明这并不总是情况。我们将看到，一个假设类可能非常复杂，甚至可能无法学习，尽管它具有少量参数。

考虑域 $\mathcal{X} = \mathbb{R}$，以及假设类

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

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

*Hint*存在多种证明所需结果的方法。一种选择是应用以下引理：如果0 .$x_1 x_2 x_3 \ldots$是$x \in$ (0,1)的二进制展开，那么对于任何自然数m，$\lceil \sin(2^m \pi x) \rceil = (1 - x_m)$，前提是$\exists k \geq m$ s.t. $x_k = 1$。

9. 令 $\mathcal{H}$ 为有符号区间的类，即，
$\mathcal{H} = \{h_{a,b,s}\ a \leq b, s \in \{-1, 1\}\}$ 其中

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

计算 VCdim($\mathcal{H}$)。

10. 令 $\mathcal{H}$ 为从 $\mathcal{X}$ 到 {0, 1} 的函数类。
1. 证明如果 VCdim($\mathcal{H}$) $\geq d$，对于任意的 d，那么对于某个概率分布 $\mathcal{D}$ 在 $\mathcal{X} \times \{0, 1\}$ 上，对于每个样本大小，m，

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

*Hint*使用第5章的练习3。
2. 证明对于每一个可PAC学习的 {v*}，有 VCdim($\mathcal{H}$) $< \infty$。（注意，这是定理6.7中的3 $\rightarrow$ 6的蕴含。）

11. **VC of union:** 设 $\mathcal{H}_1, \ldots, \mathcal{H}_r$ 是某个固定域集 $\mathcal{X}$ 上的假设类。设 $d =$ = max$_i$ VCdim($\mathcal{H}_i$)，并假设为了简单起见 $d \geq$ = 3。

1. 证明以下公式：{v*}

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

Hint: 取一个由 k 个示例组成的集合，并假设它们被联合类别所分割。因此，联合类别可以生成这些示例上所有 2^k 种可能的标签化。使用 Sauer 的引理来证明联合类别不能生成超过 rk^d 种标签化。因此，$2^k < rk^d$。现在使用引理 A.2。

2. (*) 证明对于 $r = 2$ 成立

$$\mathrm{VCdim}\left(\mathcal{H}_1 \cup \mathcal{H}_2\right) \leq 2d + 1.$$

12. **Dudley classes:** 在这个问题中，我们讨论了在 $\mathbb{R}^n$ 上定义概念类的一个代数框架，并展示了此类类的VC维度与它们的代数性质之间的关系。对于函数 $f: \mathbb{R}^n \to \mathbb{R}$，我们定义相应的函数 $POS(f)(x) = 1_{[f(x)>0]}$。对于一个实值函数类 $\mathcal{F}$，我们定义一个相应的函数类 $POS(\mathcal{F}) = \{POS(f) : f \in \mathcal{F}\}$。我们说一个实值函数族 $\mathcal{F}$ 是 *linearly closed*，如果对于所有 $f, g \in \mathcal{F}$ 和 $r \in \mathbb{R}$，$(f + rg) \in \mathcal{F}$（，其中函数的加法和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，如果一个函数族是线性闭合的，那么我们可以将其视为实数域上的一个向量空间。对于一个函数 $g: \mathbb{R}^n \to \mathbb{R}$ 和一个函数族 $\mathcal{F}$，令 $\mathcal{F} + g \stackrel{\text{def}}{=} \{f + g : f \in \mathcal{F}\}$。具有某些函数向量空间 $\mathcal{F}$ 和某些函数 g 表示的假设类被称为 *Dudley classes*。

1. 证明对于每个 $g: \mathbb{R}^n \to \mathbb{R}$ 以及之前定义的每个函数向量空间 $\mathcal{F}$，有 VCdim ($POS(\mathcal{F} + g)$) = VCdim($POS(\mathcal{F})$)。

2. (**) 对于每一个实值函数的线性闭族 $\mathcal{F}$，相应的类 $POS(\mathcal{F})$ 的VC维等于作为向量空间的 $\mathcal{F}$（的线性维度）。*Hint:* 设 $f_1, \ldots, f_d$ 为向量空间 $\mathcal{F}$ 的一个基。考虑从 $\mathbb{R}^n$ 到 $\mathbb{R}^d$) 的映射 $x \mapsto (f_1(x), \ldots, f_d(x))$（。注意，这个映射在形式为 $POS(f)$ 的函数和 $\mathbb{R}^d$（中的齐次线性半空间之间诱导出一个匹配。齐次线性半空间类的VC维在第九章）中进行分析。

3. 证明以下每个类都可以表示为Dudley类：

1. 在 $\mathbb{R}^n$（上的半空间类 HS_n，见第9章。2. 在 $\mathbb{R}^n$（上的所有齐次半空间类 HHS_n，见第9章。3. 由 $\mathbb{R}^d$ 中的（开）球定义的所有函数类 B_d。使用Dudley表示法来确定此类VC维数。4. 令 P_n^d 表示由次数为 $\leq d$ 的多项式不等式定义的函数类，即 $P_n^d = \{h_p : p$ 是变量 $x_1, \ldots, x_n\}$, 中的次数为 $\leq d$ 的多项式

在 $\mathbf{x} = (x_1 \ldots, x_n)$，$h_p(\mathbf{x}) = 1_{[p(\mathbf{x})\geq 0]}$（多元多项式的次数是所有项中变量指数的最大和。例如，$p(\mathbf{x}) = 3x_1^3x_2^2 + 4x_3x_7^2$ 的次数是 5)。

1. 使用Dudley表示法来确定类别P_1^d的VC维数——所有在$\mathbb{R}$上的d次多项式的类别。2. 证明所有在$\mathbb{R}$上的多项式分类器的类别具有无限的VC维数。3. 使用Dudley表示法来确定类别P_n^d (作为d和n)的函数的VC维数。

7 Nonuniform Learnability

本书迄今为止讨论的PAC学习性概念允许样本大小依赖于准确性和置信参数，但它们在标记规则和潜在数据分布方面是一致的。因此，在这一点上可学习的类别是有限的（它们必须具有有限的VC维，如定理6.7所述）。在本章中，我们考虑更宽松、更弱的学习性概念。我们讨论这些概念的有用性，并提供了使用这些定义可学习的概念类特征。

我们通过定义一个“非均匀可学习性”的概念来开始这次讨论，该概念允许样本大小依赖于学习器所比较的假设。然后，我们提供了非均匀可学习性的特征描述，并表明非均匀可学习性是agnostic PAC可学习性的严格放宽。我们还表明，非均匀可学习性的一个充分条件是$\mathcal{H}$是假设类的一个可数并集，每个假设类都享有均匀收敛性质。这些结果将在第7.2节通过引入一个新的学习范式，称为结构风险最小化（SRM）来证明。在第7.3节中，我们为可数假设类指定SRM范式，从而得到最小描述长度（MDL）范式。MDL范式为归纳哲学原则奥卡姆剃刀提供了一个形式上的正当理由。接下来，在第7.4节中，我们引入*consistency*作为一个更弱的学习性概念。最后，我们讨论了不同学习性概念的意义和实用性。

7.1 Nonuniform Learnability

“非均匀可学习性”允许样本大小相对于学习者所竞争的不同假设是非均匀的。我们说一个假设 h 与另一个假设 h' 是 (ϵ, δ)-竞争的，如果，以高于 $(1-\delta)$ 的概率，

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

在PAC可学习性中，这个“竞争力”的概念并不很有用，因为我们正在寻找一个具有绝对低风险（在可实现的情况下）的假设或

与我们的类别中假设所达到的最小风险相比风险较低（在无知的情形下）。因此，样本大小仅取决于准确性和置信参数。然而，在非均匀可学习性中，我们允许样本大小具有$m_{\mathcal{H}}(\epsilon, \delta, h)$的形式；即，它还取决于我们与之竞争的$h$。形式上，

DEFINITION 7.1 一个假设类 $\mathcal{H}$ 是 *nonuniformly learnable*，如果存在一个学习算法 A 和一个函数 $m_{\mathcal{H}}^{\text{NUL}} : (0,1)^2 \times \mathcal{H} \to \mathbb{N}$，使得对于每一个 $\epsilon, \delta \in (0,1)$ 和每一个 $h \in \mathcal{H}$，如果 $m \geq m_{\mathcal{H}}^{\text{NUL}}(\epsilon, \delta, h)$，那么对于每一个分布 $\mathcal{D}$，在 $S \sim \mathcal{D}^m$ 的选择上至少有 $1-\delta$ 的概率，它成立。

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

在这个时候，回顾无监督PAC学习能力的定义（定义3.3）可能是有用的：

A hypothesis class $\mathcal{H}$ is agnostically PAC learnable if there exist a learning algorithm, A, and a function $m_{\mathcal{H}} : (0,1)^2 \to \mathbb{N}$ such that, for every $\epsilon, \delta \in (0,1)$ and for every distribution $\mathcal{D}$, if $m \geq m_{\mathcal{H}}(\epsilon, \delta)$, then with probability of at least $1-\delta$ over the choice of $S \sim \mathcal{D}^m$ it holds that

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

Note that this implies that for every $h \in \mathcal{H}$

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

在两种可学习性类型中，我们要求输出假设将与类中每个其他假设保持(ϵ, δ)-竞争。但这两个可学习性概念之间的区别是样本大小m是否可能依赖于与$A(S)$的错误进行比较的假设h的问题。请注意，非均匀可学习性是无关PAC可学习性的放宽。也就是说，如果一个类是无关PAC可学习的，那么它也是非均匀可学习的。

7.1.1 Characterizing Nonuniform Learnability

我们的目标是描述非均匀可学习性。在前一章中，我们通过证明一个二分类器类是如果且仅如果其VC维是有限的，则它是无偏PAC可学习的，从而找到了PAC可学习类的清晰描述。在以下定理中，我们找到了针对二分类任务的非均匀可学习类的不同描述。

THEOREM 7.2 *A hypothesis class $\mathcal{H}$ of binary classifiers is nonuniformly learnable if and only if it is a countable union of agnostic PAC learnable hypothesis classes.*

定理7.2的证明依赖于以下具有独立兴趣的结果：

THEOREM 7.3 *Let $\mathcal{H}$ be a hypothesis class that can be written as a countable union of hypothesis classes, $\mathcal{H} = \bigcup_{n\in\mathbb{N}} \mathcal{H}_n$, where each $\mathcal{H}_n$ enjoys the uniform convergence property. Then, $\mathcal{H}$ is nonuniformly learnable.*

回忆起在第4章中，我们已证明一致收敛对于无监督PAC学习是充分的。定理7.3将这一结果推广到非一致学习。本定理的证明将在下一节通过引入一个新的学习范式给出。我们现在转向证明定理7.2。

Proof of Theorem 7.2 首先假设 $\mathcal{H} = \bigcup_{n\in\mathbb{N}} \mathcal{H}_n$ 其中每个 $\mathcal{H}_n$ 是无偏PAC可学习的。使用统计学习的基本定理，可以得出每个 $\mathcal{H}_n$ 具有一致收敛性质。因此，使用定理7.3，我们得出 $\mathcal{H}$ 是非一致可学习的。

对于另一个方向，假设 $\mathcal{H}$ 可以使用某些算法 A 非均匀地学习。对于每个 $n \in \mathbb{N}$，令 $\mathcal{H}_n = \{h \in \mathcal{H} : m^{\text{NUL}}_{\mathcal{H}}(1/8, 1/7, h) \leq n\}$。显然，$\mathcal{H} = \cup_{n\in\mathbb{N}}\mathcal{H}_n$。此外，根据 $m^{\text{NUL}}_{\mathcal{H}}$ 的定义，我们知道对于任何满足关于 $\mathcal{H}_n$ 的可实现性假设的分布 $\mathcal{D}$，在 $S \sim \mathcal{D}^n$ 上至少有 6/7 的概率，我们有 $L_{\mathcal{D}}(A(S)) \leq 1/8$。使用统计学习的基本定理，这意味着 $\mathcal{H}_n$ 的 VC 维度必须是有限的，因此 $\mathcal{H}_n$ 是无偏的 PAC 可学习的。

□

以下示例表明，非均匀可学习性是agnostic PAC可学习性的严格放宽；即存在可非均匀学习的假设类，但不是agnostic PAC可学习的。

Example 7.1 考虑一个二分类问题，其实例域为$\mathcal{X} = \mathbb{R}$。对于每个$n \in \mathbb{N}$，让$\mathcal{H}_n$是度数为n的多项式分类器的类别；即，$\mathcal{H}_n$是所有形式为$h(x) = \text{sign}(p(x))$的分类器的集合，其中$p : \mathbb{R} \to \mathbb{R}$是度数为$n$的多项式。让$\mathcal{H} = \bigcup_{n\in\mathbb{N}} \mathcal{H}_n$。因此，$\mathcal{H}$是所有在$\mathbb{R}$上的多项式分类器的类别。容易验证VCdim($\mathcal{H}$) $= \infty$，而VCdim($\mathcal{H}_n$) $= n + 1$（参见练习12）。因此，$\mathcal{H}$不是PAC可学习的，而根据定理7.3，$\mathcal{H}$是非一致可学习的。

7.2 Structural Risk Minimization

到目前为止，我们通过指定一个假设类 $\mathcal{H}$ 来编码我们的先验知识，我们认为它包括了一个好的学习任务的预测器。另一种表达我们先验知识的方法是在 $\mathcal{H}$ 内部指定对假设的偏好。在结构风险最小化（SRM）范式下，我们首先假设 $\mathcal{H}$ 可以写成 $\mathcal{H} = \bigcup_{n\in\mathbb{N}} \mathcal{H}_n$，然后指定一个权重函数，$w : \mathbb{N} \to [0, 1]$，它为每个假设类 $\mathcal{H}_n$ 分配一个权重，使得更高的权重反映了更强的对假设类的偏好。在本节中，我们讨论如何利用这种先验知识进行学习。在下一节中，我们描述了几种重要的加权方案，包括最小描述长度。

具体来说，设 $\mathcal{H}$ 为一个可以写成 $\mathcal{H} = \bigcup_{n\in\mathbb{N}} \mathcal{H}_n$ 的假设类。例如，$\mathcal{H}$ 可能是所有多项式分类器的类，其中每个 $\mathcal{H}_n$ 是度数为 n (的多项式分类器的类，参见例7.1)。假设对于每个 n，类 $\mathcal{H}_n$ 具有均匀收敛性质（参见第4章定义4.3），并具有样本复杂度函数 $m^{\text{UC}}_{\mathcal{H}_n}(\epsilon, \delta)$。让我们还定义函数 $\epsilon_n \colon \mathbb{N} \times (0, 1) \to (0, 1)$。

$$\epsilon_n(m, \delta) = \min\{\epsilon \in (0, 1) : m^{\text{UC}}_{\mathcal{H}_n}(\epsilon, \delta) \leq m\}. \tag{7.1}$$

在文字上，我们有一个固定的样本大小 m，并且我们感兴趣的是使用 m 个示例所能实现的实证风险和真实风险之间可能的最小上界。

从一致收敛和 {v*} 的定义中可以得出，对于每一个 m 和 δ，在 $S \sim \mathcal{D}^m$ 的选择上至少有 $1 - \delta$ 的概率，我们有

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

让我们 $w : \mathbb{N} \to [0, 1]$ 是一个函数，使得 $\sum_{n=1}^{\infty} w(n) \leq 1$。我们称 w 为在假设类 $\mathcal{H}_1, \mathcal{H}_2, \ldots$ 上的 *weight function*。这种权重函数可以反映学习器对每个假设类的重视程度，或者不同假设类复杂性的某种度量。如果 $\mathcal{H}$ 是 N 假设类的有限并集，则可以简单地给所有假设类分配相同的权重 $1/N$。这种等权重对应于对任何假设类没有先验偏好。当然，如果一个人相信（作为先验知识）某个假设类更有可能包含正确的目标函数，那么应该给它分配更大的权重，以反映这种先验知识。当 $\mathcal{H}$ 是假设类的（可数的）无限并集时，不可能进行均匀加权，但许多其他加权方案可能有效。例如，可以选择 $w(n) = \frac{6}{\pi^2 n^2}$ 或 $w(n) = 2^{-n}$。在本章的后面，我们将提供另一种使用描述语言定义权重函数的便捷方法。

SRM规则遵循“边界最小化”方法。这意味着该范式的目标是找到一个假设，以最小化对真实风险的某个上界。SRM规则希望最小化的边界在以下定理中给出。

THEOREM 7.4 *Let $w : \mathbb{N} \to [0, 1]$ be a function such that $\sum_{n=1}^{\infty} w(n) \leq 1$. Let $\mathcal{H}$ be a hypothesis class that can be written as $\mathcal{H} = \bigcup_{n\in\mathbb{N}} \mathcal{H}_n$, where for each n, $\mathcal{H}_n$ satisfies the uniform convergence property with a sample complexity function $m^{UC}_{\mathcal{H}_n}$. Let ϵ_n be as defined in Equation (7.1). Then, for every $\delta \in (0, 1)$ and distribution $\mathcal{D}$, with probability of at least $1 - \delta$ over the choice of $S \sim \mathcal{D}^m$, the following bound holds (simultaneously) for every $n \in \mathbb{N}$ and $h \in \mathcal{H}_n$.*

$$|L_{\mathcal{D}}(h) - L_S(h)| \leq \epsilon_n(m, w(n) \cdot \delta).$$

Therefore, for every $\delta \in (0, 1)$ and distribution $\mathcal{D}$, with probability of at least

$1-\delta$ *it holds that*

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

Proof 对于每个 n 定义 $\delta_n = w(n)\delta$。应用假设对于所有 n 均匀收敛成立，其速率由方程（7.2）给出，我们得到，如果我们事先固定 n，那么在 $S \sim \mathcal{D}^m$ 的选择上，至少有 $1-\delta_n$ 的概率。

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

应用对 $n = 1, 2, \ldots$ 的并集界，我们得到，至少以概率 $1 - \sum_n \delta_n = 1 - \delta \sum_n w(n) \geq 1-\delta$，前面的结论对所有 n 都成立，这完成了我们的证明。

□

表示

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

然后方程（7.3）表明

$$L_{\mathcal{D}}(h) \leq L_S(h) + \epsilon_{n(h)}(m, w(n(h))\cdot\delta).$$

SRM范式寻找使此界限最小化的h，如下伪代码所示：

Structural Risk Minimization (SRM)
prior knowledge: $\mathcal{H} = \bigcup_n \mathcal{H}_n$ where $\mathcal{H}_n$ has uniform convergence with $m^{\text{UC}}_{\mathcal{H}_n}$ $w : \mathbb{N} \to [0,1]$ where $\sum_n w(n) \leq 1$ **define:** ϵ_n as in Equation (7.1) ; $n(h)$ as in Equation (7.4) **input:** training set $S \sim \mathcal{D}^m$, confidence δ **output:** $h \in \text{argmin}_{h\in\mathcal{H}} \left[L_S(h) + \epsilon_{n(h)}(m, w(n(h))\cdot\delta)\right]$

与前面章节中讨论的ERM范式不同，我们不再仅仅关注经验风险$L_S(h)$，而是愿意用对经验风险较小的类别的偏差来换取对低经验风险的偏差，以减小估计误差。

接下来，我们展示SRM范式可以用于每个类别的非均匀学习，这是一个均匀收敛假设类别的可数并集。

THEOREM 7.5 *Let $\mathcal{H}$ be a hypothesis class such that $\mathcal{H} = \bigcup_{n\in\mathbb{N}} \mathcal{H}_n$, where each $\mathcal{H}_n$ has the uniform convergence property with sample complexity $m^{UC}_{\mathcal{H}_n}$. Let $w : \mathbb{N} \to [0,1]$ be such that $w(n) = \frac{6}{n^2\pi^2}$. Then, $\mathcal{H}$ is nonuniformly learnable using the SRM rule with rate*

$$m^{NUL}_{\mathcal{H}}(\epsilon, \delta, h) \ \leq\ m^{UC}_{\mathcal{H}_{n(h)}}\left(\epsilon/2\,,\ \tfrac{6\delta}{(\pi n(h))^2}\right).$$

Proof 设 A 为关于加权函数 w 的 SRM 算法。对于每一个 $h \in \mathcal{H}$，ϵ 和 δ，令 $m \geq m^{\text{UC}}_{\mathcal{H}_{n(h)}}(\epsilon, w(n(h))\delta)$。利用 $\sum_n w(n) = 1$ 的性质，我们可以应用定理 7.4，得到在至少 $1-\delta$ 的概率下，关于 $S \sim \mathcal{D}^m$ 的选择，我们有对于每一个 $h' \in \mathcal{H}$，

$$L_{\mathcal{D}}(h') \leq L_S(h') + \epsilon_{n(h')}(m, w(n(h'))\delta).$$

前述内容特别适用于SRM规则返回的假设 $A(S)$。根据SRM的定义，我们得到：

$$L_{\mathcal{D}}(A(S)) \leq \min_{h'} \left[L_S(h') + \epsilon_{n(h')}(m, w(n(h'))\delta)\right] \leq L_S(h) + \epsilon_{n(h)}(m, w(n(h))\delta).$$

最后，如果 $m \geq m^{\text{UC}}_{\mathcal{H}_{n(h)}}(\epsilon/2, w(n(h))\delta)$，那么显然 $\epsilon_{n(h)}(m, w(n(h))\delta) \leq \epsilon/2$。此外，从每个 $\mathcal{H}_n$ 的均匀收敛性质，我们有超过 1 的概率 $-\delta$，

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

将所有前面的内容结合起来，我们得到 $L_{\mathcal{D}}(A(S)) \leq L_{\mathcal{D}}(h) + \epsilon$，这完成了我们的证明。 □

请注意，前一个定理也证明了定理7.3。

Remark 7.2 (非均匀可学习性午餐无价值) 我们已经证明，任何有限VC维度的类集合的可数并集都是非均匀可学习的。结果发现，对于任何无限领域集合$\mathcal{X}$，所有二元值函数的集合不是有限VC维度类集合的可数并集。我们将这个命题的证明留作一个（非平凡的）练习（参见练习5）。因此，从某种意义上说，无午餐定理对于非均匀学习也成立：即，当领域不是有限的时，不存在针对所有确定性二元分类器的非均匀学习器（尽管对于每个这样的分类器，都存在一个简单的学习算法——相对于只包含这个分类器的假设类进行ERM）。

有趣的是将定理7.5中给出的非均匀可学习性结果与分别对任何特定的$\mathcal{H}_n$进行无偏PAC学习任务进行比较。对于$\mathcal{H}$的非均匀学习者的先验知识或偏差较弱——它在整个类$\mathcal{H}$中寻找模型，而不是专注于一个特定的$\mathcal{H}_n$。这种先验知识弱化的代价是需要增加样本复杂度以与任何特定的$h \in \mathcal{H}_n$竞争。为了具体评估这个差距，考虑具有零一损失的二元分类任务。假设对于所有n，VCdim($\mathcal{H}_n$) $= n$。由于$m^{\text{UC}}_{\mathcal{H}_n}(\epsilon, \delta) = C\,\frac{n+\log(1/\delta)}{\epsilon^2}$ (其中C是定理6.8中出现的常数)，简单的计算表明

$$m^{\text{NUL}}_{\mathcal{H}}(\epsilon, \delta, h) - m^{\text{UC}}_{\mathcal{H}_n}(\epsilon/2, \delta) \leq 4C\frac{2\log(2n)}{\epsilon^2}.$$

这意味着，从包含目标 h 的特定 $\mathcal{H}_n$ 放松到类别的可数并集的成本取决于对数

该索引表示h所在的第一类。该成本随着类别的索引增加，可以解释为反映了在$\mathcal{H}$中了解良好优先级顺序的价值。

7.3 Minimum Description Length and Occam's Razor

设 $\mathcal{H}$ 为一个可数假设类。那么，我们可以将 $\mathcal{H}$ 写作单元素类的可数并集，即 $\mathcal{H} = \bigcup_{n\in\mathbb{N}}\{h_n\}$。根据 Hoeffding 不等式（引理 4.5），每个单元素类都具有以速率 $m^{\mathrm{UC}}(\epsilon,\delta) = \frac{\log(2/\delta)}{2\epsilon^2}$ 的均匀收敛性质。因此，方程（7.1）中给出的函数 ϵ_n 变为 $\epsilon_n(m,\delta) = \sqrt{\frac{\log(2/\delta)}{2m}}$，SRM 规则变为

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

等效地，我们可以将 w 视为一个从 $\mathcal{H}$ 到 [0, 1] 的函数，然后 SRM 规则变为

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

因此，在这种情况下，先验知识完全由我们分配给每个假设的权重决定。我们给我们认为更有可能正确的假设分配更高的权重，在学习算法中，我们更倾向于具有更高权重的假设。

在这个部分，我们讨论定义一个权重函数的特定方便方法，该函数基于对假设的描述长度。拥有一个假设类，人们可能会想知道我们如何描述或表示该类中的每个假设。我们自然地固定了一些描述语言。这可以是英语，编程语言，或一组数学公式。在这些语言中的任何一种，描述都由从某个固定字母表中抽取的有限符号（或字符）字符串组成。现在我们将正式化这些概念。

设 $\mathcal{H}$ 为我们希望描述的假设类。固定一些有限的符号集 Σ（或称为“字符集”），我们称之为字母表。为了具体化，我们设 $\Sigma = \{0,1\}$。字符串是来自 Σ 的有限符号序列；例如，$\boldsymbol{\sigma} = (0,1,1,1,0)$ 是长度为 5 的字符串。我们用 $|\boldsymbol{\sigma}|$ 表示字符串的长度。所有有限长度字符串的集合表示为 Σ^*。$\mathcal{H}$ 的描述语言是一个函数 $d : \mathcal{H} \to \Sigma^*$，将 $\mathcal{H}$ 的每个成员 h 映射到一个字符串 $d(h)$。$d(h)$ 被称为“h 的描述”，其长度表示为 $|h|$。

我们应要求描述语言是 *prefix-free*；也就是说，对于每个不同的 h, h'，$d(h)$ 不是 $d(h')$ 的前缀。也就是说，我们不允许任何字符串 $d(h)$ 完全等于任何更长字符串 $d(h')$ 的前 $|h|$ 个符号。字符串的无前缀集合具有以下组合性质：

LEMMA 7.6 （Kraft 不等式）*If $\mathcal{S} \subseteq \{0,1\}^*$ is a prefix-free set of strings, then*

$$\sum_{\boldsymbol{\sigma} \in \mathcal{S}} \frac{1}{2^{|\boldsymbol{\sigma}|}} \leq 1.$$

Proof 定义一个在$\mathcal{S}$成员上的概率分布如下：反复掷一个标有0和1的面无偏硬币，直到结果序列是$\mathcal{S}$的成员；此时停止。对于每个$\boldsymbol{\sigma} \in \mathcal{S}$，让$P(\boldsymbol{\sigma})$是这个过程生成字符串$\boldsymbol{\sigma}$的概率。注意，由于$\mathcal{S}$是无前缀的，对于每个$\boldsymbol{\sigma} \in \mathcal{S}$，如果掷硬币的结果跟随$\boldsymbol{\sigma}$的位，那么我们只有在结果序列等于$\boldsymbol{\sigma}$时才会停止。因此，对于每个$\boldsymbol{\sigma} \in \mathcal{S}$，我们得到$P(\boldsymbol{\sigma}) = \frac{1}{2^{|\boldsymbol{\sigma}|}}$。由于概率之和最多为1，我们的证明结束。

□

根据Kraft不等式，任何关于假设类$\mathcal{H}$的无前缀描述语言都会产生一个在该假设类上的权重函数w - 我们将简单地设置$w(h) = \frac{1}{2^{|h|}}$。这个观察立即得出以下结论：

THEOREM 7.7 *Let $\mathcal{H}$ be a hypothesis class and let $d : \mathcal{H} \to \{0,1\}^*$ be a prefix-free description language for $\mathcal{H}$. Then, for every sample size, m, every confidence parameter, $\delta > 0$, and every probability distribution, $\mathcal{D}$, with probability greater than $1-\delta$ over the choice of $S \sim \mathcal{D}^m$ we have that,*

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

where $|h|$ is the length of $d(h)$.

Proof 选择 $w(h) = 1/2^{|h|}$，应用定理7.4，并注意$\ln(2^{|h|}) = |h| \ln(2) < |h|$。

□

与定理7.4的情况一样，这个结果为$\mathcal{H}$提出了一种学习范式——给定一个训练集S，寻找一个最小化界限$L_S(h) + \sqrt{\frac{|h|+\ln(2/\delta)}{2m}}$的假设$h \in \mathcal{H}$。特别是，它建议以牺牲经验风险为代价来节省描述长度。这产生了最小描述长度学习范式。

Minimum Description Length (MDL)

prior knowledge:
$\mathcal{H}$ is a countable hypothesis class
$\mathcal{H}$ is described by a prefix-free language over $\{0,1\}$
For every $h \in \mathcal{H}$, $|h|$ is the length of the representation of h
input: A training set $S \sim \mathcal{D}^m$, confidence δ
output: $h \in \operatorname{argmin}_{h \in \mathcal{H}} \left[L_S(h) + \sqrt{\frac{|h|+\ln(2/\delta)}{2m}} \right]$

Example 7.3 让 $\mathcal{H}$ 成为所有可以使用某种编程语言（例如，C++）实现的预测器的类别。让我们用以下方式表示每个程序：

二进制字符串是通过运行gzip命令在程序上获得的（这产生了一个在字母表{0, 1 }上的前缀无关描述语言）。然后，$|h|$简单地是gzip在运行与h对应的C++程序时的输出长度（以比特为单位）。

7.3.1 Occam's Razor

定理7.7表明，当有两个假设具有相同的经验风险时，具有较短描述的那个假设的真实风险可以被一个较低值所界定。因此，这个结果可以被视为传达了一种哲学信息：

A short explanation (that is, a hypothesis that has a short length) tends to be more valid than a long explanation.

这是一个广为人知的原理，被称为奥卡姆剃刀，以14世纪英国逻辑学家威廉·奥卡姆的名字命名，据信他是第一个明确表述它的人。在这里，我们提供对这个原理的一种可能论证。定理7.7的不等式表明，一个假设h越复杂（在描述越长的意义上），它必须拟合的样本量就越大，以保证它具有较小的真实风险$L_{\mathcal{D}}(h)$。

在第二次审视时，我们的奥卡姆剃刀主张可能显得有些问题。在通常在科学中引用奥卡姆剃刀原则的背景下，衡量复杂性的语言是自然语言，而在这里我们可以考虑任何任意的抽象描述语言。假设我们有两个假设，其中 $|h'|$ 远小于 $|h|$。根据前面的结果，如果它们在给定的训练集上具有相同的误差 S，那么 h 的真实误差可能远高于 h' 的真实误差，因此应该选择 h' 而不是 h。然而，我们可能选择了不同的描述语言，比如说，将长度为 3 的字符串分配给 h，将长度为 100000 的字符串分配给 h'。突然看起来好像应该选择 h 而不是 h'。但这些都是我们在两句话前认为 h' 应该更可取的相同的 h 和 h'。这里的陷阱在哪里?

确实，假设之间没有固有的泛化差异。这里的关键方面是初始语言选择（或对假设的偏好）与训练集之间的依赖顺序。正如我们从基本的Hoeffding界（方程（4.2））所知，如果我们对任何假设 *be- fore* 承诺看到数据，那么我们保证有一个相当小的估计误差项 $L_{\mathcal{D}}(h) \leq L_S(h) + \sqrt{\frac{\ln(2/\delta)}{2m}}$。选择描述语言（或等价地，某些假设的加权）是对假设承诺的一种弱形式。我们不是对单个假设做出承诺，而是将我们的承诺分散到许多假设中。只要它与训练样本无关，我们的泛化界就成立。正如单个假设的选择可以通过样本进行任意选择一样，描述语言的选择也是如此。

7.4 Other Notions of Learnability – Consistency

学习性的概念可以通过允许所需的样本大小不仅依赖于 ϵ, δ 和 h，还依赖于用于生成训练样本和确定风险的底层数据生成概率分布 $\mathcal{D}$ (来进一步放宽。这种性能保证由学习规则的概念 *consistency*[1] 所捕捉。

DEFINITION 7.8（一致性）设Z为一个域集，设$\mathcal{P}$为在Z上的概率分布集，设$\mathcal{H}$为一个假设类。一个学习规则A相对于$\mathcal{H}$和$\mathcal{P}$是*consistent*，如果存在一个函数$m_{\mathcal{H}}^{\text{CON}}\colon (0,1)^2 \times \mathcal{H} \times \mathcal{P} \to \mathbb{N}$，使得对于每个$\epsilon, \delta \in (0,1)$，每个$h \in \mathcal{H}$，以及每个$\mathcal{D} \in \mathcal{P}$，如果$m \geq m_{\mathcal{H}}^{\text{NUL}}(\epsilon, \delta, h, \mathcal{D})$，那么以至少$1-\delta$的概率在$S \sim \mathcal{D}^m$的选择上成立。

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

如果 $\mathcal{P}$ 是所有分布的集合，[2] 我们称 A 是相对于 $\mathcal{H}$ 的 *universally consistent*。

一致性概念，当然是我们之前非均匀可学习性概念的放宽。显然，如果一个算法非均匀地学习一个类别 $\mathcal{H}$，那么它对该类别也是普遍一致的。这种放宽是严格的，因为在存在一致的学习规则，但它们不是成功的非均匀学习者。例如，在7.4节中定义的算法 `Memorize` 对所有在 $\mathbb{N}$ 上的二元分类器的类别是普遍一致的。然而，正如我们之前所论证的，这个类别不是非均匀可学习的。

Example 7.4 考虑如下定义的类别预测算法 `Memorize`。该算法记忆训练示例，并给定一个测试点 x，它预测训练样本中所有标记实例 x 中的多数标签（如果训练集中没有 x 的实例，则预测一些固定的默认标签）。可以证明（参见练习 6）该 `Memorize` 算法对于每个可计数的域 $\mathcal{X}$ 和有限标签集 $\mathcal{Y}$ (相对于零一损失) 是普遍一致的。

直观上，并不明显地将 `Memorize` 算法视为 *learner*，因为它缺乏泛化方面，即使用观察到的数据来预测未见示例的标签。因此，`Memorize` 是任何可数域集上所有函数类的一致算法，这因此引发了关于一致性保证有用性的怀疑。此外，细心的读者可能会注意到，我们在第二章中引入的“差劲的学习者”，

[1] In the literature, consistency is often defined using the notion of either convergence in probability (corresponding to weak consistency) or almost sure convergence (corresponding to strong consistency).

[2] Formally, we assume that Z is endowed with some sigma algebra of subsets Ω, and by "all distributions" we mean all probability distributions that have Ω contained in their associated family of measurable subsets.

这导致了过拟合，实际上是一种 Memorize 算法。在下一节中，我们将讨论不同可学习性概念的显著性，并基于不同可学习性的定义重新审视无免费午餐定理。

7.5 Discussing the Different Notions of Learnability

我们已经给出了三个可学习性的定义，现在我们讨论它们的实用性。通常情况下，数学定义的实用性取决于我们用它来做什么。因此，我们列出几个我们希望通过定义可学习性来实现的可能目标，并讨论在不同目标下不同定义的实用性。

What Is the Risk of the Learned Hypothesis?
第一个可能的目标是在学习算法上推导性能保证是限制输出预测器的风险。在这里，PAC学习和非均匀学习都为我们提供了基于其经验风险的学习假设的真实风险的界限。一致性保证不提供这样的界限。然而，总是可以使用验证集来估计输出预测器的风险（如第11章所述）。

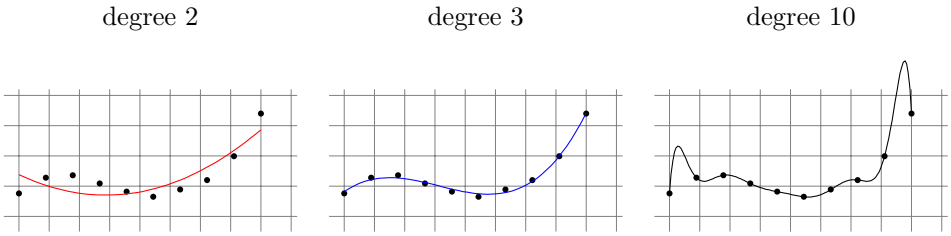
How Many Examples Are Required to Be as Good as the Best Hypothesis in $\mathcal{H}$?
当接近一个学习问题时，一个自然的问题是我们需要收集多少个样本才能学习它。在这里，PAC学习给出了一个明确的答案。然而，对于非均匀学习和一致性，我们事先不知道需要多少个样本来学习$\mathcal{H}$。在非均匀学习中，这个数字取决于$\mathcal{H}$中的最佳假设，而在一致性中，它也取决于潜在的分布。从这个意义上说，PAC学习是唯一有用的可学习性定义。另一方面，应该记住，即使我们学习的预测器的估计误差很小，如果$\mathcal{H}$有大的近似误差，其风险仍然可能很大。因此，对于“需要多少个样本才能与贝叶斯最优预测器一样好？”这个问题，即使PAC保证也不能给我们一个明确的答案。这反映了PAC学习的有用性依赖于我们先验知识的质量。

PAC保证还帮助我们理解，如果我们的学习算法返回一个风险很大的假设，我们应该做什么，因为我们可以限制由估计误差引起的错误部分，因此知道多少错误归因于近似误差。如果近似误差很大，我们知道我们应该使用不同的假设类。同样，如果一个非均匀算法失败，我们可以考虑在（假设的）子集上使用不同的加权函数。然而，当一个一致算法失败时，我们不知道这是否是因为估计误差还是近似误差。此外，即使我们确信我们有一个估计问题

误差项，我们不知道还需要多少更多示例才能使估计误差变小。

How to Learn? How to Express Prior Knowledge?
也许学习理论最有用的方面在于为“如何学习”的问题提供答案。PAC学习的定义产生了学习的限制（通过无免费午餐定理）和先验知识的必要性。它通过选择一个假设类来提供一种清晰的方式来编码先验知识，一旦做出这个选择，我们就有了通用的学习规则——ERM。非均匀学习性的定义也通过在$\mathcal{H}$的假设（子集）上指定权重来提供一种清晰的方式来编码先验知识。一旦做出这个选择，我们再次有了通用的学习规则——SRM。SRM规则在先验知识部分的情况下，在模型选择任务中也具有优势。我们在第11章中详细阐述了模型选择，这里我们给出一个简要的例子。

考虑将一维多项式拟合到数据的问题；也就是说，我们的目标是学习一个函数，$h : \mathbb{R} \to \mathbb{R}$，并且作为先验知识，我们考虑多项式的假设类。然而，我们可能不确定哪个次数 d 会给我们数据集带来最佳结果：较低的次数可能无法很好地拟合数据（即它将有较大的近似误差），而较高的次数可能导致过拟合（即它将有较大的估计误差）。在以下内容中，我们将展示将次数为2、3和10的多项式拟合到同一训练集的结果。



可以看出，随着我们增大度数，经验风险会降低。因此，如果我们选择 $\mathcal{H}$ 为所有最高次数为 10 的多项式类，那么针对这个类的 ERM 规则将输出一个 10 次多项式并会发生过拟合。另一方面，如果我们选择一个太小的假设类，比如最高次数为 2 的多项式，那么 ERM 将会遭受欠拟合（即大的近似误差）。相比之下，我们可以在所有多项式的集合上使用 SRM 规则，同时按照它们的度数对 $\mathcal{H}$ 的子集进行排序，这将产生一个 3 次多项式，因为其经验风险和估计误差的界限是最小的。换句话说，SRM 规则使我们能够根据数据本身选择正确的模型。我们为此灵活性付出的代价（除了相对于最优度数的 PAC 学习，估计误差略有增加之外）是我们不知道

需要提前多少个例子才能与$\mathcal{H}$中的最佳假设相竞争。

与PAC学习性和非一致性学习性不同，一致性的定义并没有产生一个自然的学习范式或编码先验知识的方法。事实上，在许多情况下，根本不需要先验知识。例如，我们看到即使是直观上不应被称为学习算法的Memorize算法，对于在可数域和有限标签集上定义的任何类别来说，也是一个一致算法。这表明一致性是一个非常弱的要求。

Which Learning Algorithm Should We Prefer?
可以争论说，尽管一致性是一个较弱的要求，但学习算法与从$\mathcal{X}$到$\mathcal{Y}$的所有函数集的一致性是可取的，这给我们一个保证，即对于足够的训练样本，我们总是能像贝叶斯最优预测器一样好。因此，如果我们有两个算法，其中一个是一致的，另一个则不是，我们应该更喜欢一致的算法。然而，这个论点有两个问题。首先，也许在大多数“自然”分布中，我们将在实践中观察到一致算法的样本复杂度会非常大，以至于在每种实际情况下，我们都不会获得足够的例子来享受这种保证。其次，使任何PAC或非均匀学习器与从$\mathcal{X}$到$\mathcal{Y}$的所有函数类一致并不难。具体来说，考虑一个可数域$\mathcal{X}$，一个有限的标签集$\mathcal{Y}$，以及一个从$\mathcal{X}$到$\mathcal{Y}$的函数类$\mathcal{H}$。我们可以使用以下简单技巧使任何非均匀学习器对于从$\mathcal{X}$到$\mathcal{Y}$的all分类器类一致：在接收到训练集后，我们首先在训练集上运行非均匀学习器，然后我们得到学习预测器的真实风险的界限。如果这个界限足够小，我们就完成了。否则，我们回到Memorize算法。这种简单的修改使算法对所有从$\mathcal{X}$到$\mathcal{Y}$的函数一致。由于使任何算法一致很容易，因此仅仅因为一致性考虑而更喜欢一个算法而不是另一个可能并不明智。

7.5.1 The No-Free-Lunch Theorem Revisited

回忆一下，无免费午餐定理（第5章第5.1定理）意味着没有任何算法可以在无限域上学习所有分类器。相比之下，在本章中我们看到了Memorize算法与可数无限域上所有分类器的类别一致。为了理解这两个陈述为什么不会相互矛盾，让我们首先回忆无免费午餐定理的正式陈述。

设 $\mathcal{X}$ 为一个可数无限域，并设 $\mathcal{Y} = \{\pm 1\}$。无免费午餐定理意味着以下结论：对于任何算法 A 和训练集大小 m，存在一个在 $\mathcal{X}$ 上的分布和一个函数 $h^\star : \mathcal{X} \to \mathcal{Y}$，使得如果 A

将获得一个由 m 独立同分布的训练样本，标记为 $h^\star$，然后 A 很可能返回一个误差更大的分类器。

`Memorize` 的一致性意味着以下内容：对于 $\mathcal{X}$ 上的每个分布和标签函数 $h^\star$: $\mathcal{X} \to \mathcal{Y}$，存在一个依赖于分布和 $h^\star$) 的训练集大小 m (，使得如果 `Memorize` 至少接收到 m 个示例，它很可能返回一个具有小误差的分类器。

我们在No-Free-Lunch定理中看到，我们首先固定训练集大小，然后找到一个对这种训练集大小有害的分布和标签函数。相比之下，在一致性保证中，我们首先固定分布和标签函数，然后才找到一个足以学习这种特定分布和标签函数的训练集大小。

7.6 Summary

我们引入了非均匀学习性作为PAC学习性的放松，以及一致性作为非均匀学习性的放松。这意味着即使无限VC维度的类也可以在某种较弱的学习性意义上被学习。我们讨论了不同学习性定义的有用性。

对于可数的假设类，我们可以应用最小描述长度方案，其中更短的描述的假设更受青睐，遵循奥卡姆剃刀原则。一个有趣的例子是所有我们可以用C++ (或任何其他编程语言)实现的预测器的假设类，我们可以使用MDL方案（非均匀地）来学习。

或许，我们可以在C++中实现的全部预测器类别是一个强大的函数类，并且可能包含我们在实践中希望学习到的一切。学习这个类的能力令人印象深刻，似乎这一章应该是这本书的最后一章。但这并非事实，因为学习的计算方面：即应用学习规则所需的运行时间。例如，为了针对所有C++程序实现MDL范式，我们需要对所有C++程序进行穷举搜索，这将永远无法完成。即使是针对描述长度最多为1000比特的所有C++程序实现ERM范式，也需要对2^{1000}个假设进行穷举搜索。虽然学习这个类的样本复杂度仅为$\frac{1000+\log(2/\delta)}{\epsilon^2}$，但运行时间是$\geq 2^{1000}$。这是一个巨大的数字——比可见宇宙中的原子数量还要大。在下一章中，我们将正式定义学习的计算复杂性。在这本书的第二部分，我们将研究可以高效实现ERM或SRM方案的假设类别。

7.7 Bibliographic Remarks

我们的非均匀可学习性定义与Blumer, Ehrenfeucht, Haussler & Warmuth (1987)中Occam算法的定义相关。SRM概念归功于(Vapnik & Chervonenkis 1974, Vapnik 1995)。MDL概念归功于(Rissanen 1978, Rissanen 1983)。SRM与MDL之间的关系在Vapnik (1995)中进行了讨论。这些概念也与$regularization$ (概念密切相关，例如Tikhonov (1943))。我们将在本书的第二部分详细阐述正则化。

一致性估计量的概念可以追溯到Fisher（1922年）。我们关于一致性的表述遵循Steinwart & Christmann（2008年），他们还推导出了一些无免费午餐定理。

7.8 Exercises

1. 证明对于任何有限类 $\mathcal{H}$，以及任何描述语言 d：$\mathcal{H} \to \{0,1\}^*$，$\mathcal{H}$ 的VC维数至多为 2 sup$\{|d(h)|: h \in \mathcal{H}\}$ - $\mathcal{H}$ 中预测器的最大描述长度。此外，如果 d 是一个无前缀的描述，那么 VCdim(H) $\leq$ sup$\{|d(h)|: h \in \mathcal{H}\}$。2. 设 $\mathcal{H} = \{h_n: n \in \mathbb{N}\}$ 是一个二分类的无限可数假设类。证明不可能为 $\mathcal{H}$ 中的假设分配权重，使得

- $\mathcal{H}$ 可以使用这些权重非均匀地学习。也就是说，加权函数 $w : \mathcal{H} \to [0,1]$ 应满足条件 $\sum_{h\in\mathcal{H}} w(h) \leq 1$。
- 权重将是单调不减的。也就是说，如果 $i < j$，那么 $w(h_i) \leq w(h_j)$。3。
- 考虑一个假设类 $\mathcal{H} = \bigcup_{n=1}^{\infty} \mathcal{H}_n$，其中对于每个 $n \in \mathbb{N}$，$\mathcal{H}_n$ 是有限的。找到一个权重函数 $w : \mathcal{H} \to [0,1]$，使得 $\sum_{h\in\mathcal{H}} w(h) \leq 1$，并且对于所有 $h \in \mathcal{H}$，$w(h)$ 由 $n(h) = \min\{n : h \in \mathcal{H}_n\}$ 和 $|\mathcal{H}_{n(h)}|$ 确定。
- (*) 定义当对于所有 n $\mathcal{H}_n$ 是可数（可能无限）时这样的函数 w。

4. 令 $\mathcal{H}$ 为某个假设类。对于任意的 $h \in \mathcal{H}$，令 $|h|$ 表示 h 的描述长度，根据某种固定的描述语言。考虑 MDL 学习范式，其中算法返回：

$$h_S \in \arg\min_{h\in\mathcal{H}} \left[L_S(h) + \sqrt{\frac{|h| + \ln(2/\delta)}{2m}} \right],$$

在 S 是大小为 m 的样本的情况下。对于任何 $B > 0$，令 $\mathcal{H}_B = \{h \in \mathcal{H} : |h| \leq B\}$，并定义

$$h_B^* = \arg\min_{h\in\mathcal{H}_B} L_{\mathcal{D}}(h).$$

证明 $L_{\mathcal{D}}(h_S) - L_{\mathcal{D}}(h_B^*)$ 的一个界限，以 B、置信参数 δ 和训练集大小 m 为条件。

- 注意：此类界限在文献中被称为 *oracle inequalities*：我们希望估计我们与参考分类器（或“神谕”）h_B^* 相比有多好。

5. 在这个问题中，我们希望展示非均匀学习能力的无免费午餐结果：即在任何无限域上，*all*函数类即使在放宽的非均匀学习变化下也是不可学习的。

回忆一下，一个算法A，*nonuniformly learns*一个假设类$\mathcal{H}$如果存在一个函数$m_{\mathcal{H}}^{\text{NUL}}$：$(0,1)^2 \times \mathcal{H} \to \mathbb{N}$，使得对于每一个$\epsilon, \delta \in (0,1)$和对于每一个$h \in \mathcal{H}$，如果$m \geq m_{\mathcal{H}}^{\text{NUL}}(\epsilon, \delta, h)$那么对于每一个分布$\mathcal{D}$，在$S \sim D^m$的选择上至少有1 – δ的概率，它成立

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

如果存在这样的算法，那么我们说 $\mathcal{H}$ 是 *nonuniformly learnable*。

1. 令 A 为一类 $\mathcal{H}$ 的非均匀学习器。对于每个 $n \in \mathbb{N}$ 定义 $\mathcal{H}_n^A = \{h \in \mathcal{H} : m^{\text{NUL}}(0.1, 0.1, h) \leq n\}$。证明每个此类 $\mathcal{H}_n$ 都具有有限的 VC 维度。

2. 证明如果类别 $\mathcal{H}$ 是非均匀可学习的，那么存在类别 $\mathcal{H}_n$，使得 $\mathcal{H} = \bigcup_{n \in \mathbb{N}} \mathcal{H}_n$，并且对于每个 $n \in \mathbb{N}$，VCdim($\mathcal{H}_n$) 是有限的。

3. 设 $\mathcal{H}$ 是一个能够分割无限集的类。那么，对于每一个类序列 $(\mathcal{H}_n : n \in \mathbb{N})$，使得 $\mathcal{H} = \bigcup_{n \in \mathbb{N}} \mathcal{H}_n$，存在某个 n，使得 VCdim($\mathcal{H}_n$) $= \infty$。

Hint: Given a class $\mathcal{H}$ that shatters some infinite set K, and a sequence of classes $(\mathcal{H}_n : n \in \mathbb{N})$, each having a finite VC-dimension, start by defining subsets $K_n \subseteq K$ such that, for all n, $|K_n| >$ VCdim($\mathcal{H}_n$) and for any $n \neq m$, $K_n \cap K_m = \emptyset$. Now, pick for each such K_n a function $f_n : K_n \to \{0,1\}$ so that no $h \in \mathcal{H}_n$ agrees with f_n on the domain K_n. Finally, define $f : X \to \{0,1\}$ by combining these f_n's and prove that $f \in \left(\mathcal{H} \setminus \bigcup_{n \in \mathbb{N}} \mathcal{H}_n\right)$.

4. 构造一个从单位区间 [0, 1] 到 $\{0,1\}$ 的函数类 $\mathcal{H}_1$，它是非均匀可学习的但不是 PAC 可学习的。

5. 构造一个从单位区间 [0, 1] 到 $\{0,1\}$ 的函数类 $\mathcal{H}_2$，该类不是非均匀可学习的。

6. 在这个问题中，我们希望证明算法 {v*} 对于任何可数域上的（二元值）函数的每一类都是一致的学习者。设 $\mathcal{X}$ 为一个可数域，设 $\mathcal{D}$ 为 $\mathcal{X}$ 上的一个概率分布。

1. 设 $\{x_i : i \in \mathbb{N}\}$ 为 $\mathcal{X}$ 的元素枚举，使得对于所有 $i \leq j$，$\mathcal{D}(\{x_i\}) \leq \mathcal{D}(\{x_j\})$。证明

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

2. 给定任意的 $\epsilon > 0$，证明存在 $\epsilon_D > 0$ 使得

$$\mathcal{D}(\{x \in \mathcal{X} : \mathcal{D}(\{x\}) < \epsilon_D\}) < \epsilon.$$

3. 证明对于每个 $\eta > 0$，如果 n 是这样的，使得 $\mathcal{D}(\{x_i\}) < \eta$ 对于所有 $i > n$ 成立，那么对于每个 $m \in \mathbb{N}$，

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

4. 结论：如果 $\mathcal{X}$ 是可数的，那么对于每个在 $\mathcal{X}$ 上的概率分布 $\mathcal{D}$，存在一个函数 $m_{\mathcal{D}}$：$(0,1)\times(0,1)\to\mathbb{N}$，使得对于每个 $\epsilon,\delta > 0$ 如果 $m > m_{\mathcal{D}}(\epsilon,\delta)$ 则

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

5. 证明 {v*} 对于任何可数域上的（二元值）函数的每一类都是一致学习器。

8 The Runtime of Learning

到目前为止，在本书中我们研究了学习的统计视角，即学习需要多少样本。换句话说，我们关注学习所需的信息量。然而，在考虑自动学习时，计算资源在确定任务复杂度方面也起着重要作用：即执行学习任务涉及多少*computation*。一旦学习者可获得足够的训练样本，就需要进行一些计算以提取假设或确定给定测试实例的标签。这些计算资源在机器学习的任何实际应用中都至关重要。我们将这两种资源称为*sample complexity*和*computational complexity*。在本章中，我们将注意力转向学习的计算复杂性。

学习计算的复杂性应在更广泛的通用算法任务计算复杂性的背景下进行考虑。这一领域已被广泛研究；例如，参见（Sipser 2006）。以下介绍性评论总结了与我们的讨论最相关的该一般理论的基本思想。

实际运行时间（以秒为单位）取决于算法实现的特定机器（例如，机器CPU的时钟频率）。为了避免对特定机器的依赖，通常以渐近意义分析算法的运行时间。例如，我们说归并排序算法的计算复杂度，该算法对一个包含n个元素的列表进行排序，是$O(n \log(n))$。这意味着我们可以在满足某些接受的抽象计算模型要求的任何机器上实现该算法，并且实际的运行时间（以秒为单位）将满足以下条件：存在常数c和n_0，这些常数可能依赖于实际机器，对于任何$n > n_0$的值，排序任何n个元素的运行时间将不超过$c\, n \log(n)$。对于运行时间对于某些多项式函数p为$O(p(n))$的算法可以执行的任务，通常使用术语*feasible*或*efficiently computable*。应该注意的是，这种分析取决于定义算法预期应用于任何实例的输入大小n。对于在常见的计算复杂性文献中讨论的“纯粹算法”任务，这个输入大小是明确定义的；算法得到一个输入实例，例如，要排序的列表或要计算的算术运算，它有一个明确的大小（例如，列表的长度或运算符的数量）。

表示中的位数）。对于机器学习任务，输入大小的概念并不那么明确。算法旨在检测数据集中的一些模式，并且只能访问该数据的随机样本。

我们本章首先讨论这个问题，并定义了学习的计算复杂度。对于高级学生，我们还提供了详细的正式定义。然后，我们继续考虑实现ERM规则的计算复杂度。我们首先给出几个可以高效实现ERM规则的假设类别的例子，然后考虑一些尽管类别确实可以高效学习，但ERM实现计算上很困难的情况。因此，实现ERM的困难并不意味着学习的困难。最后，我们简要讨论了如何证明给定学习任务的困难性，即没有任何学习算法可以高效地解决它。

8.1 Computational Complexity of Learning

回忆一下，一个学习算法可以访问一个示例域，Z，一个假设类，$\mathcal{H}$，一个损失函数，ℓ，以及从 Z 中采样的示例训练集，这些示例根据一个未知的分布 $\mathcal{D}$ 独立同分布地采样。给定参数 ϵ, δ，算法应该输出一个假设 h，使得至少以概率 $1 - \delta$，

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

如前所述，一个算法的实际运行时间（以秒为单位）取决于特定的机器。为了允许机器无关的分析，我们使用计算复杂度理论中的标准方法。首先，我们依赖于一个抽象机器的概念，例如图灵机（或实数上的图灵机（Blum, Shub & Smale 1989））。其次，我们以渐近意义分析运行时间，同时忽略常数因子，因此只要实现了抽象机器，具体的机器就不重要。通常，渐近线是与算法输入的大小有关的。例如，对于之前提到的归并排序算法，我们分析运行时间作为需要排序的项目数量的函数。

在学习算法的背景下，没有明确的“输入大小”概念。人们可能会将输入大小定义为算法接收到的训练集大小，但这相当没有意义。如果我们给算法一个非常大的示例数量，远大于学习问题的样本复杂度，算法可以简单地忽略额外的示例。因此，更大的训练集并不会使学习问题更难，因此，随着训练集大小的增加，学习算法的运行时间不应增加。同样，我们仍然可以将运行时间作为问题自然参数的函数来分析，例如目标精度、达到该精度的置信度以及问题的维数。

领域集，或与算法输出进行比较的假设类复杂性的某些度量。

为了说明这一点，考虑一个用于学习轴对齐矩形的算法。通过指定 ϵ、δ 和实例空间的维度，可以推导出学习轴对齐矩形的一个特定问题。我们可以通过固定 ϵ, δ 并改变维度为 $d = 2, 3, 4, \ldots$ 来定义一系列“矩形学习”问题。我们还可以通过固定 d, δ 并改变目标精度为 $\epsilon = \frac{1}{2}, \frac{1}{3}, \ldots$ 来定义另一系列“矩形学习”问题。当然，可以选择其他此类问题的序列。一旦固定了一个问题序列，就可以分析该序列变量的渐近运行时间。

在介绍正式定义之前，我们还需要解决一个细微的问题。基于前面的内容，一个学习算法可以通过将计算负担转移到输出假设上来“作弊”。例如，算法可以简单地定义输出假设为存储训练集在其内存中的函数，每当它接收到一个测试示例 x 时，它就在训练集上计算 ERM 假设并将其应用于 x。请注意，在这种情况下，我们的算法有一个固定的输出（即我们刚刚描述的函数）并且可以在常数时间内运行。然而，学习仍然很困难——困难现在在于实现输出分类器以获得标签预测。为了防止这种“作弊”，我们要求学习算法的输出必须应用于预测新示例的标签，其时间不超过训练时间（即从输入训练样本计算输出分类器的时间）。在下一小节中，高级读者可能会找到学习计算复杂度的正式定义。

8.1.1 Formal Definition*

以下定义依赖于一个底层抽象机器的概念，这通常是一个图灵机或实数上的图灵机。我们将使用算法所需的“操作”数量来衡量其计算复杂度，其中我们假设对于任何实现底层抽象机器的机器，存在一个常数 c，使得任何此类“操作”都可以在 c 秒内在机器上执行。

DEFINITION 8.1（学习算法的计算复杂性）我们定义学习复杂度为两步。首先，我们考虑一个固定学习问题的计算复杂性（由三元组$(Z, \mathcal{H}, \ell)$确定——一个领域集、一个基准假设类和损失函数）。然后，在第二步中，我们考虑这种复杂性沿着一系列此类任务的序列变化率。

1. 给定一个函数 $f : (0, 1)^2 \to \mathbb{N}$，一个学习任务 $(Z, \mathcal{H}, \ell)$ 和一个学习算法，$\mathcal{A}$，我们说 $\mathcal{A}$ 在时间 $O(f)$ 内解决了学习任务，如果存在某个常数 c，使得对于每个概率分布 $\mathcal{D}$

在 Z 上，并且输入 $\epsilon,\ \delta \in (0,1)$，当 $\mathcal{A}$ 可以访问由 $\mathcal{D}$ 独立同分布生成的样本时，

- $\mathcal{A}$ 执行最多 $cf(\epsilon,\ \delta)$ 次操作后终止
- $\mathcal{A}$的输出，表示为$h_{\mathcal{A}}$，可以在执行最多$cf(\epsilon,\ \delta)$次操作的情况下预测新示例的标签
- $\mathcal{A}$的输出可能大致正确；即，在随机样本$\mathcal{A}$接收到的)，$L_{\mathcal{D}}(h_{\mathcal{A}}) \leq \min_{h' \in \mathcal{H}} L_{\mathcal{D}}(h') + \epsilon$上，概率至少为$1-\delta$ (

2. 考虑一个学习问题序列，$(Z_n, \mathcal{H}_n, \ell_n)_{n=1}^{\infty}$，其中问题 n 由一个域 Z_n、一个假设类 $\mathcal{H}_n$ 和一个损失函数 ℓ_n 定义。

   设 $\mathcal{A}$ 为一种针对此类学习问题设计的算法。给定一个函数 $g : \mathbb{N} \times (0,1)^2 \to \mathbb{N}$，我们称 $\mathcal{A}$ 相对于前述序列的运行时间为 $O(g)$，如果对于所有 n，$\mathcal{A}$ 在时间 $O(f_n)$ 内解决了问题 $(Z_n, \mathcal{H}_n, \ell_n)$，其中 $f_n : (0,1)^2 \to \mathbb{N}$ 由 $f_n(\epsilon, \delta) = g(n, \epsilon, \delta)$ 定义。

我们称$\mathcal{A}$为相对于序列$(Z_n, \mathcal{H}_n, \ell_n)$的*efficient*算法，如果其运行时间是$O(p(n, 1/\epsilon, 1/\delta))$，对于某个多项式$p$。

从这个定义中我们可以看出，一个一般学习问题是否可以高效解决取决于它如何被分解为一系列具体学习问题。例如，考虑学习一个有限假设类的问题。正如我们在前几章所展示的，当训练样本的数量是 $m_{\mathcal{H}}(\epsilon, \delta) = \log(|\mathcal{H}|/\delta)/\epsilon^2$ 的阶时，ERM 规则对 $\mathcal{H}$ 的保证可以 (ϵ, δ)-学习 $\mathcal{H}$。假设对示例上的假设评估需要常数时间，可以通过对大小为 $m_{\mathcal{H}}(\epsilon, \delta)$ 的训练集进行穷举搜索来实现 ERM 规则在 $O(|\mathcal{H}|\, m_{\mathcal{H}}(\epsilon, \delta))$ 时间内。对于任何固定的有限 $\mathcal{H}$，穷举搜索算法在多项式时间内运行。此外，如果我们定义一个包含 $|\mathcal{H}_n| = n$ 的问题序列，那么穷举搜索仍然被认为是高效的。然而，如果我们定义一个包含 $|\mathcal{H}_n| = 2^n$ 的问题序列，那么样本复杂度仍然是 n 的多项式，但穷举搜索算法的计算复杂度随着 n (指数增长，因此变得低效。

8.2 Implementing the ERM Rule

给定一个假设类 $\mathcal{H}$，ERM$_{\mathcal{H}}$ 规则可能是最自然的学习范式。此外，对于二分类问题，我们观察到如果学习在任何情况下都是可能的，那么它可以通过 ERM 规则实现。在本节中，我们讨论实现 ERM 规则的几个假设类的计算复杂性。

给定一个假设类 $\mathcal{H}$，一个域集 Z 和一个损失函数 ℓ，相应的 ERM$_{\mathcal{H}}$ 规则可以定义为如下:

在有限输入样本 $S \in Z^m$ 上输出一些 $h \in \mathcal{H}$ 以最小化经验损失，
$L_S(h) = \frac{1}{|S|} \sum_{z \in S} \ell(h, z)$。

本节研究了实现 ERM 规则的运行时间，针对几个学习任务的示例进行了研究。

8.2.1 Finite Classes

限制假设类为有限类可能被视为一种合理的温和限制。例如，$\mathcal{H}$可以是所有可以通过最多10000位代码编写的C++程序实现的预测因子集合。其他有用的有限类示例是任何可以用有限数量的参数参数化的假设类，其中我们满足于使用有限数量的位来表示每个参数，例如，欧几里得空间中轴对齐矩形的类$\mathbb{R}^d$，当定义任何给定矩形的参数达到某种有限的精度时。

如前几章所示，学习有限类别的样本复杂度被上界为 $m_{\mathcal{H}}(\epsilon, \delta) = c \log(c|\mathcal{H}|/\delta)/\epsilon^c$，其中 $c = 1$ 在可实现的情形下，$c = 2$ 在不可实现的情形下。因此，样本复杂度对 $\mathcal{H}$ 的大小有轻微的依赖。在前面提到的 C++ 程序的例子中，假设的数量是 $2^{10,000}$，但样本复杂度仅为 $c(10,000 + \log(c/\delta))/\epsilon^c$。

一种在有限假设类上实现ERM规则的直接方法是进行穷举搜索。也就是说，对于每个$h \in \mathcal{H}$，我们计算经验风险$L_S(h)$，并返回一个使经验风险最小化的假设。假设在单个样本上评估$\ell(h, z)$所需的时间是常数k，则这种穷举搜索的运行时间变为$k|\mathcal{H}|m$，其中m是训练集的大小。如果我们让m成为样本复杂度的上界，那么运行时间变为$k|\mathcal{H}|c \log(c|\mathcal{H}|/\delta)/\epsilon^c$。

线性时间复杂度依赖于 {v*} 的大小，这使得这种方法对于大型类别来说效率低下（且不切实际）。形式上，如果我们定义一个问题序列 $(Z_n, \mathcal{H}_n, \ell_n)_{n=1}^{\infty}$，使得 $\log(|\mathcal{H}_n|) = n$，那么穷举搜索方法将产生指数级时间复杂度。在 C++ 程序的例子中，如果 $\mathcal{H}_n$ 是可以通过最多 n 位代码编写的 C++ 程序实现的函数集合，那么时间复杂度随着 n 的增长呈指数级增长，这意味着穷举搜索方法对于实际应用来说不切实际。事实上，这正是我们处理其他假设类别的理由之一，例如线性预测器类别，我们将在下一章中遇到，而不仅仅关注有限类别。

重要的是要认识到，一种算法方法（如穷举搜索）的低效并不意味着不存在有效的ERM实现。事实上，我们将展示一些例子，其中ERM规则可以有效地实现。

8.2.2 Axis Aligned Rectangles

设 $\mathcal{H}_n$ 为 $\mathbb{R}^n$ 中轴对齐矩形的类，即，

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

哪里

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

Efficiently Learnable in the Realizable Case
考虑在可实现情况下实施ERM规则。也就是说，我们给定一个示例训练集 $S = (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$，存在一个与轴对齐的矩形$h \in \mathcal{H}_n$，使得对于所有$i$，都有$h(\mathbf{x}_i) = y_i$。我们的目标是找到一个与轴对齐的矩形，具有零训练错误，即一个与S中所有标签一致的矩形。

我们稍后表明这可以在时间 $O(nm)$ 内完成。实际上，对于每个 $i \in [n]$，设置 $a_i = \min\{x_i : (\mathbf{x}, 1) \in S\}$ 和 $b_i = \max\{x_i : (\mathbf{x}, 1) \in S\}$。换句话说，我们将 a_i 视为 S 中正例的第 i 个坐标的最小值，将 b_i 视为 S 中正例的第 i 个坐标的最大值。很容易验证，得到的矩形具有零训练误差，并且找到每个 a_i 和 b_i 的运行时间是 $O(m)$。因此，此过程的总体运行时间是 $O(nm)$。

Not Efficiently Learnable in the Agnostic Case
在非确定情况下，我们不假设某些假设 h 完美地预测了训练集中所有示例的标签。因此，我们的目标是找到 h，以最小化 $y_i \neq h(\mathbf{x}_i)$ 的示例数量。结果发现，对于许多常见的假设类别，包括我们在此处考虑的轴对齐矩形类别，在非确定设置中解决ERM问题属于NP难题（并且，在大多数情况下，甚至难以找到一些 $h \in \mathcal{H}$，其误差不超过经验风险最小化器在 $\mathcal{H}$ 中误差的某个常数 $c > 1$ 倍）。也就是说，除非 P = NP，否则不存在运行时间在 m 和 n 上是多项式的算法，可以保证找到这些问题的ERM假设（Ben-David，Eiron & Long 2003）。

另一方面，值得注意的是，如果我们固定一个特定的假设类，比如在某些固定维度上的轴对齐矩形，n，那么这个类存在有效的学习算法。换句话说，存在在时间上以$1/\epsilon$和$1/\delta$ (的多项式时间内运行的成功的无偏见PAC学习器，但它们对维度n的依赖不是多项式的)。

要看到这一点，回忆一下我们为可实现情况提出的ERM规则的实施，由此可知，一个与轴对齐的矩形最多由$2n$个示例确定。因此，给定一个大小为m的训练集，我们可以对所有大小最多为$2n$个示例的训练集子集进行穷举搜索，并从每个这样的子集中构建一个矩形。然后，我们可以选择

具有最小训练误差的矩形。此过程保证找到ERM假设，该过程的运行时间为$m^{O(n)}$。因此，如果n是固定的，则运行时间与样本大小是多项式级别的。这并不与上述困难结果矛盾，因为在那时我们论证了除非P=NP，否则不能有一个算法其对维度n的依赖也是多项式的。

8.2.3 Boolean Conjunctions

布尔合取是一个从 $\mathcal{X} = \{0,1\}^n$ 到 $\mathcal{Y} = \{0,1\}$ 的映射，它可以表示为形式 $x_{i_1} \wedge \ldots \wedge x_{i_k} \wedge \neg x_{j_1} \wedge \ldots \wedge \neg x_{j_r}$ 的命题公式，对于某些索引 $i_1, \ldots, i_k, j_1, \ldots, j_r \in [n]$。这样一个命题公式定义的函数是

$$h(\mathbf{x}) = \begin{cases} 1 & \text{if } x_{i_1} = \cdots = x_{i_k} = 1 \text{ and } x_{j_1} = \cdots = x_{j_r} = 0 \\ 0 & \text{otherwise} \end{cases}$$

让 $\mathcal{H}_C^n$ 成为所有在 $\{0,1\}^n$ 上的布尔合取类的集合。$\mathcal{H}_C^n$ 的大小至多为 $3^n + 1$（因为在合取公式中，$\mathbf{x}$ 的每个元素要么出现，要么出现否定符号，要么根本不出现，我们还有所有负公式）。因此，使用 ERM 规则学习 $\mathcal{H}_C^n$ 的样本复杂度至多为 $n \log(3/\delta)/\epsilon$。

Efficiently Learnable in the Realizable Case

接下来，我们展示在时间多项式于 n 和 m 的情况下，可以解决 $\mathcal{H}_C^n$ 的 ERM 问题。思路是在假设合取中包含所有不与任何正标签示例矛盾的文字。令 $\mathbf{v}_1, \ldots, \mathbf{v}_{m^+}$ 为输入样本 S 中所有正标签实例。我们通过 $i \leq m^+$ 的归纳定义了一个假设（或合取）序列。令 h_0 为所有可能文字的合取。即，$h_0 = x_1 \wedge \neg x_1 \wedge x_2 \wedge \ldots \wedge x_n \wedge \neg x_n$。注意 h_0 将 $\mathcal{X}$ 的所有元素分配标签 0。我们通过从合取 h_i 中删除所有不被 $\mathbf{v}_{i+1}$ 满足的文字得到 h_{i+1}。算法输出假设 h_{m^+}。注意 h_{m^+} 标记了 S 中所有正标签示例为正。此外，对于每个 $i \leq m^+$，h_i 是标记 $\mathbf{v}_1, \ldots, \mathbf{v}_i$ 为正的最严格合取。现在，由于我们考虑在可实现的设置中学习，存在一个合取假设 $f \in \mathcal{H}_C^n$，它与 S 中的所有示例一致。由于 h_{m^+} 是标记 S 中所有正标签成员为正的最严格合取，因此任何被 f 标记为 0 的实例也被 h_{m^+} 标记为 0。因此，h_{m^+} 具有零训练误差（相对于 S），因此是一个合法的 ERM 假设。注意该算法的运行时间是 $O(mn)$。

Not Efficiently Learnable in the Agnostic Case
与轴对齐的矩形情况相同，除非 P = NP，否则不存在运行时间在 m 和 n 上是多项式的算法，可以保证在不可实现的情况下找到布尔合取类的一个ERM假设。

8.2.4 Learning 3-Term DNF

我们接下来表明，布尔合取类的一个轻微推广会导致在可实现的情形下求解ERM问题的不可解性。考虑3项析取范式公式（3项DNF）的类。实例空间是 $\mathcal{X} = \{0,1\}^n$，每个假设由形式为$h(\mathbf{x}) = A_1(\mathbf{x}) \vee A_2(\mathbf{x}) \vee A_3(\mathbf{x})$的布尔公式表示，其中每个$A_i(\mathbf{x})$是一个布尔合取（如前节定义）。如果$A_1(\mathbf{x})$或$A_2(\mathbf{x})$或$A_3(\mathbf{x})$中的任何一个输出标签1，则$h(\mathbf{x})$的输出为1。如果所有三个合取都输出标签0，则$h(\mathbf{x}) = 0$。

设 $\mathcal{H}^n_{3DNF}$ 为所有此类 3 项 DNF 公式的假设类。$\mathcal{H}^n_{3DNF}$ 的大小至多为 3^{3n}。因此，使用 ERM 规则学习 $\mathcal{H}^n_{3DNF}$ 的样本复杂度至多为 $3n \log(3/\delta)/\epsilon$。

然而，从计算的角度来看，这个问题很难。已经证明（参见（Pitt & Valiant 1988，Kearns 等人 1994）），除非 RP = NP，就没有多项式时间算法能够 *properly* 学习一个 3 项 DNF 学习问题序列，其中第 n 个问题的维度是 n。这里的“正确”意味着算法应该输出一个 3 项 DNF 公式的假设。特别是，由于 ERM$_{\mathcal{H}^n_{3DNF}}$ 输出 3 项 DNF 公式，它是一个正确的学习器，因此很难实现它。证明使用了将图 3-着色问题减少到 PAC 学习 3 项 DNF 问题的方法。详细技术见练习 3。参见（Kearns & Vazirani 1994，第 1.4 节）。

8.3 Efficiently Learnable, but Not by a Proper ERM

在上一节中，我们看到了对于3-DNF公式的类别$\mathcal{H}^n_{3DNF}$，无法有效地实现ERM规则。在本节中，我们展示了可以有效地学习这个类别，但使用的是相对于更大类别的ERM。

Representation Independent Learning Is Not Hard
接下来，我们展示可以有效地学习3项DNF公式。这与前节中提到的困难结果没有矛盾，因为我们现在允许“表示无关”的学习。也就是说，我们允许学习算法输出一个不是3项DNF公式的假设。基本思想是用一个更大的假设类替换原始的3项DNF公式假设类，以便新的类易于学习。学习

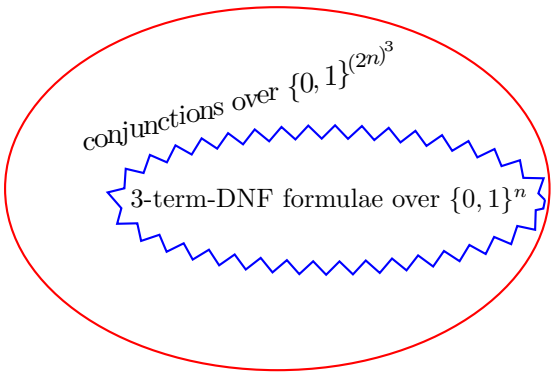
算法可能返回一个不属于原始假设类别的假设；因此得名“表示无关”学习。我们强调，在大多数情况下，返回具有良好预测能力的假设是我们真正感兴趣的事情。

我们首先注意到，因为 $\vee$ 在 $\wedge$ 上分配，所以每个三项DNF公式都可以重写为

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

接下来，让我们定义：$\psi : \{0,1\}^n \to \{0,1\}^{(2n)^3}$，使得对于每个文字三元组 u, v, w，都有一个变量在 ψ 的范围内指示 $u \vee v \vee w$ 是真还是假。因此，对于每个在 $\{0,1\}^n$ 上的 3-DNF 公式，都有一个在 $\{0,1\}^{(2n)^3}$ 上的合取，具有相同的真值表。由于我们假设数据是可实现的，我们可以针对 $\{0,1\}^{(2n)^3}$ 上的合取类解决 ERM 问题。此外，在学习更高维空间中合取类的样本复杂度至多为 $n^3 \log(1/\delta)/\epsilon$。因此，这种方法的总运行时间在 n 上是多项式的。

直观上，想法如下。我们从学习困难的一个假设类开始。我们切换到另一个表示，其中假设类比原始类更大，但结构更复杂，这允许更有效的ERM搜索。在新表示中，解决ERM问题很容易。



8.4 Hardness of Learning*

我们刚刚证明，实现ERM$_{\mathcal{H}}$的计算难度并不意味着此类类别$\mathcal{H}$不可学习。我们如何证明一个学习问题是计算上困难的?

一种方法依赖于密码学假设。在某种意义上，密码学与学习相反。在学习中，我们试图揭示我们所看到的例子背后的某些规则，而在密码学中，目标是确保即使有人能够访问，也无法发现某些秘密。

关于它的某些部分信息。在这个高级直观层面上，关于某些系统密码学安全性的结果转化为关于某些相应任务不可学习性的结果。遗憾的是，目前还没有一种方法可以证明一个密码协议是不可破解的。即使是常见的P $\neq$ NP假设也不足以证明这一点（尽管它可以被证明对于大多数常见的密码学场景是必要的）。证明密码协议安全性的常见方法是从某些 *cryp- tographic assumptions* 开始。这些被用作密码学基础的程度越高，我们对其真实性的信念就越强（或者至少，反驳它们的算法很难找到）。

我们现在简要描述如何从密码学假设中推导出学习难度硬度的基本思想。许多密码系统依赖于存在单射函数的假设。粗略地说，单射函数是一个函数 $f: \{0,1\}^n \to \{0,1\}^n$（更正式地说，它是一系列函数，每个维度 n）都有一个易于计算但难以逆转的函数。更正式地说，f 可以在时间 poly(n) 内计算，但对于任何随机多项式时间算法 A，以及对于每个多项式 $p(\cdot)$，

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

在随机选择 $\mathbf{x}$ 的均匀分布上对 $\{0,1\}^n$ 进行概率取值，以及 A 的随机性。

一个单射函数 f 被称为带陷阱门的单射函数，如果对于某个多项式函数 p，对于每个 n，都存在一个长度为 $\leq p(n)$ 的位串 s_n（，称为密钥），使得存在一个多项式时间算法，对于每个 n 和每个 $\mathbf{x} \in \{0,1\}^n$，在输入 $(f(\mathbf{x}), s_n)$ 上输出 $\mathbf{x}$。换句话说，尽管 f 难以求逆，一旦获得其密钥，求逆 f 就变得可行。这些函数由它们的密钥参数化。

现在，设F_n是定义在$\{0,1\}^n$上的一个门函数族，它可以被某个多项式时间算法计算。也就是说，我们固定一个算法，给定一个密钥（代表F_n中的一个函数）和一个输入向量，它可以在多项式时间内计算出与密钥对应的函数在输入向量上的值。考虑学习对应逆函数类$H_F^n = \{f^{-1} : f \in F_n\}$的任务。由于这个类中的每个函数都可以被某个大小为多项式的密钥s_n逆，因此这个类H_F^n可以由这些密钥参数化，其大小至多为$2^{p(n)}$。因此，其样本复杂度是多项式的。我们声称对于这个类不存在有效的学习器。如果存在这样的学习器L，那么通过在$\{0,1\}^n$中随机均匀采样多项式数量的字符串，并计算它们上的f，我们可以生成一个标记的训练样本对$(f(\mathbf{x}), \mathbf{x})$，这应该足以让我们的学习器找到$(\epsilon, \delta)$关于$f^{-1}$（相对于$f$）的均匀分布的近似，这将违反$f$的单向性质。

更详细的处理以及一个具体示例可以在（Kearns & Vazirani 1994，第6章）中找到。使用约简，他们还表明

函数类可以通过小布尔电路计算，即使在可实现的情形下，也不是高效可学习的。

8.5 Summary

学习算法的运行时间被作为学习问题不同参数的函数进行渐近分析，例如假设类的大小、我们的准确性度量、我们的置信度度量或领域集的大小。我们已经证明了ERM规则可以高效实现的案例。例如，我们在可实现性假设下，为布尔合取类和轴对齐矩形类推导了求解ERM问题的有效算法。然而，在非确定情况下实现这些类的ERM是NP难的。回想一下，从统计学的角度来看，可实现和非可实现情况之间没有区别（即，如果且仅如果一个类具有有限的VC维，则该类在这两种情况下都是可学习的）。相反，正如我们所看到的，从计算的角度来看，这种差异是巨大的。我们还展示了另一个例子，即3项DNF类，即使在可实现情况下实现ERM也很困难，但该类可以通过另一种算法高效地学习。

硬度实现ERM规则对于几个自然假设类激发了替代学习方法的开发，我们将在本书下一部分讨论这些方法。

8.6 Bibliographic Remarks

Valiant (1984) 介绍了高效的 PAC 学习模型，其中算法的运行时间要求在 $1/\epsilon$、$1/\delta$ 和类中假设表示大小上是多项式级的。Kearns & Vazirani (1994) 中给出了详细的讨论和详尽的文献注释。

8.7 Exercises

1. 令 $\mathcal{H}$ 为直线上的区间类（形式上等同于维度 $n = 1$ 中的轴对齐矩形）。提出一种实现 ERM$_{\mathcal{H}}$ 学习规则（在无偏见情况下）的方案，该方案在大小为 m 的训练集上运行，所需时间为 $O(m^2)$。*Hint*：使用动态规划。

2. 令 $\mathcal{H}_1, \mathcal{H}_2, \ldots$ 为二元分类的假设类序列。假设存在一个学习算法在可实现的情形下实现了ERM规则，使得算法对于每个类别 $\mathcal{H}_n$ 的输出假设仅依赖于训练集中的 $O(n)$ 个示例。此外，

假设在给定这些 $O(n)$ 个示例的时间 $O(n)$ 内可以计算出这样的假设，并且每个这样的假设的经验风险可以在时间 $O(mn)$ 内被评估。例如，如果 $\mathcal{H}_n$ 是 $\mathbb{R}^n$ 中轴对齐矩形的类别，我们看到了在可实现的情形下，可以找到一个最多由 $2n$ 个示例定义的 ERM 假设。证明在这种情况下，可以在时间 $O(mn\, m^{O(n)})$ 内找到一个不可实现的情形下的 $\mathcal{H}_n$ ERM 假设。

3. 在本练习中，我们提出了几个对于寻找ERM分类器来说计算上很困难的类别。首先，我们引入了n-维半空间类别HS_n，针对域$\mathcal{X} = \mathbb{R}^n$。这是所有形式为$h_{\mathbf{w},b}(\mathbf{x}) = \text{sign}(\langle \mathbf{w}, \mathbf{x} \rangle + b)$的函数的类别，其中$\mathbf{w}, \mathbf{x} \in \mathbb{R}^n$，$\langle \mathbf{w}, \mathbf{x} \rangle$是它们的内积，和$b \in \mathbb{R}$。详见第9章的详细描述。

1. 证明在由线性预测器类 $\mathcal{H} = HS_n$ 上的 ERM$_{\mathcal{H}}$ 计算上是困难的。更确切地说，我们考虑维度 n 线性增长且示例数量 m 被设置为某个常量乘以 n 的问题序列。

*Hint*您可以通过从以下问题进行归约来证明其难度：

Max FS: 给定一个线性不等式系统，$A\mathbf{x} > \mathbf{b}$，包含 $A \in R^{m \times n}$ 和 $\mathbf{b} \in \mathbb{R}^m$（，即一个在 n 变量上的 m 个线性不等式系统，找到包含尽可能多不等式的子系统，该子系统有解（这样的子系统称为 *feasible*）。

已证明（Sankaran 1993）问题 Max FS 是 NP-hard。证明任何找到任何训练样本 $S \in (\mathbb{R}^n \times \{+1, -1\})^m$ 的 ERM$_{HS_n}$ 假设的算法都可以用来解决大小为 m, n 的 Max FS 问题。*Hint*：定义一个映射，将 n 变量的线性不等式转换为 $\mathbb{R}^n$ 中的标记点，并定义一个映射，将 $\mathbb{R}^n$ 中的向量转换为半空间，使得向量 $\mathbf{w}$ 满足不等式 q 当且仅当与 q 对应的标记点被与 $\mathbf{w}$ 对应的半空间正确分类。得出结论，半空间经验风险最小化问题也是 NP-hard（也就是说，如果可以在样本大小 m 和欧几里得维度 n 的多项式时间内解决，那么类 NP 中的每个问题都可以在多项式时间内解决）。

2. 设 $\mathcal{X} = \mathbb{R}^n$，设 $\mathcal{H}_k^n$ 为 k 个线性半空间在 $\mathbb{R}^n$ 中所有交集的类。在本练习中，我们希望证明 ERM$_{\mathcal{H}_k^n}$ 对于每个 $k \geq$ 都是计算上困难的。3. 准确地说，我们考虑一系列问题，其中 $k \geq 3$ 是一个常数，n 线性增长。训练集大小 m 也随着 n 线性增长。

朝着这个目标，考虑图的颜色问题，定义为如下：

给定一个图 $G = (V, E)$，和一个数字 k，确定是否存在一个函数 $f : V \to \{1 \ldots k\}$，使得对于每一个 $(u, v) \in E$，$f(u) \neq f(v)$。

k-着色问题已知对于每个 $k \geq 3$ 都是 NP-hard（Karp 1972）。

我们希望将 k-着色问题减少到 $ERM_{\mathcal{H}_k^n}$：也就是说，要证明如果存在一个算法可以在 k、n 和样本大小 m 的多项式时间内解决 $ERM_{\mathcal{H}_k^n}$ 问题，那么就存在一个针对图 k-着色问题的多项式时间算法。

给定一个图 $G = (V, E)$，令 $\{v_1 \dots v_n\}$ 为 V 中的顶点。构建一个示例 $S(G) \in (\mathbb{R}^n \times \{\pm 1\})^m$，其中 $m = |V| + |E|$ 如下：

- 对于每个 $v_i \in V$，构建一个具有负标签的实例 $\mathbf{e}_i$。
- 对于每条边 $(v_i, v_j) \in E$，构建一个具有正标签的实例 $(\mathbf{e}_i + \mathbf{e}_j)/2$。1. 证明如果存在某个 $h \in \mathcal{H}_k^n$ 在 $S(G)$ 上具有零误差，那么 G 是 k-可着色的。*Hint:* 设 $h = \bigcap_{j=1}^{k} h_j$ 是 $\mathcal{H}_k^n$ 上的一个ERM分类器。通过将 $f(v_i)$ 设为满足 $h_j(\mathbf{e}_i) = -$ 的最小 j 来定义 V 的着色。利用半空间是凸集的事实来证明，通过边连接的两个顶点不可能具有相同的颜色。2. 证明如果 G 是 k-可着色的，那么存在某个 $h \in H_k^n$ 在 $S(G)$ 上具有零误差。*Hint:* 给定顶点的着色 f G，我们应该找到 k 超平面，$h_1 \dots h_k$ 其交集是一个完美的分类器 $S(G)$。对于所有这些超平面，设 $b = 0.6$，对于 $t \leq k$，设第 i 个超平面 t 的第 $w_{t,i}$ 个超平面的第 $-$ 个权重为 $f(v_i) = t$ 时为1，否则为0。3. 基于上述内容，证明对于任何 $k \geq 3$，ERM $_{\mathcal{H}_k^n}$ 问题是NP难的。

4. 在这个练习中，我们展示了解决ERM问题的难度与正确PAC学习的难度相当。回想一下，我们所说的“正确性”是指算法必须从假设类中输出一个假设。为了使这个陈述形式化，我们首先需要以下定义。

DEFINITION 8.2 随机多项式（RP）时间复杂度类是所有决策问题（即在任意实例中都必须找出答案是否为YES或NO的问题）的集合，对于这些问题存在一个概率算法（即算法在运行过程中允许抛掷随机硬币）具有以下性质：

- 在任何输入实例中，算法在输入大小上的运行时间都是多项式时间。
- 如果正确答案是NO，则算法必须返回NO。
- 如果正确答案是YES，则算法以概率 $a \geq 1/2$ 返回YES，以概率 $1 - a$.[1] 返回NO

显然，类RP包含类P。同时，已知RP包含在类NP中。目前尚不清楚这三个复杂度类之间是否存在任何等价性，但普遍认为NP是严格

[1] The constant 1/2 in the definition can be replaced by any constant in $(0, 1)$.

大于RP。特别是，人们认为NP难问题不能通过随机多项式时间算法解决。

- 证明如果一个类 $\mathcal{H}$ 可以被多项式时间算法 *properly* PAC 学习，那么 ERM$_{\mathcal{H}}$ 问题属于 RP 类。特别是，这表明每当 ERM$_{\mathcal{H}}$ 问题 NP-hard（例如，在前一个练习中讨论的半空间交集类），除非 NP $=$ RP，否则不存在多项式时间正确的 PAC 学习算法 $\mathcal{H}$。

  Hint: 假设你有一个算法 A，它可以正确地以某些类参数 n 的多项式时间以及 $1/\epsilon$ 和 $1/\delta$ 的多项式时间来 PAC 学习一个类 $\mathcal{H}$。你的目标是使用该算法作为子例程，在随机多项式时间内解决 ERM$_{\mathcal{H}}$ 问题。给定一个训练集 $S \in (\mathcal{X} \times \{\pm 1\}^m)$ 和一些 $h \in \mathcal{H}$，它在 S 上的误差为零，将 PAC 学习算法应用于 S 上的均匀分布，并运行它，以概率 ≥ 0.3 它找到一个函数 $h \in \mathcal{H}$，该函数相对于该均匀分布的误差小于 $\epsilon = 1/|S|$（。证明所描述的算法满足作为 ERM$_{\mathcal{H}}$ 的 RP 求解器的需求。

Part II

From Theory to Algorithms

9 Linear Predictors

在这一章中，我们将研究线性预测器的家族，这是假设类中最有用的家族之一。许多在实际中得到广泛应用的算法都依赖于线性预测器，首先是因为在许多情况下可以有效地学习它们。此外，线性预测器直观、易于解释，并且在许多自然学习问题中合理地拟合数据。

我们将介绍属于这个家族的几个假设类别——半空间、线性回归预测器和逻辑回归预测器——并介绍相关学习算法：半空间类别的线性规划和感知器算法，以及线性回归的最小二乘算法。本章专注于使用ERM方法学习线性预测器；然而，在后面的章节中，我们将看到学习这些假设类别的替代范式。

首先，我们定义仿射函数类为

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

哪里

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

它也将方便使用以下符号{v*}

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

L_d 是一组函数，其中每个函数由 $\mathbf{w} \in \mathbb{R}^d$ 和 $b \in \mathbb{R}$ 参数化，每个这样的函数将向量 x 作为输入，并将标量 $\langle \mathbf{w}, \mathbf{x} \rangle + b$ 作为输出。

不同线性预测器的假设类是函数 $\phi : \mathbb{R} \to \mathcal{Y}$ 在 L_d 上的组合。例如，在二分类中，我们可以选择 ϕ 为符号函数，而对于回归问题，其中 $\mathcal{Y} = \mathbb{R}$，ϕ 简单地是恒等函数。

可能更方便将称为 $bias$ 的 b 纳入 $\mathbf{w}$ 作为额外的坐标，并将值为 1 的额外坐标添加到所有 $\mathbf{x} \in \mathcal{X}$；即，设 $\mathbf{w}' = (b, w_1, w_2, \dots w_d) \in \mathbb{R}^{d+1}$ 和设 $\mathbf{x}' = (1, x_1, x_2, \dots, x_d) \in$

$\mathbb{R}^{d+}$1. 因此，

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

因此，$\mathbb{R}^d$中的每个仿射函数都可以重写为$\mathbb{R}^{d+1}$中的齐次线性函数，该函数应用于将常数1附加到每个输入向量的变换。因此，每当简化表示时，我们将省略偏差项，并将L_d称为形式为$h_{\mathbf{w}}(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle$的齐次线性函数类。

全书我们经常使用“线性函数”这一通用术语来指代仿射函数和（齐次）线性函数。

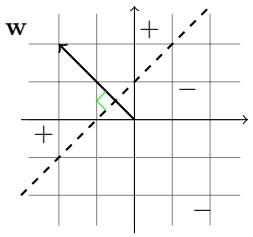
9.1 Halfspaces

第一个我们考虑的假设类是半空间类，用于二元分类问题，即$\mathcal{X} = \mathbb{R}^d$和$\mathcal{Y} = \{-1, +1\}$。半空间类定义为以下：

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

换句话说，HS_d中的每个半空间假设由$\mathbf{w} \in \mathbb{R}^d$和$b \in \mathbb{R}$参数化，在接收到向量$\mathbf{x}$后，假设返回标签符号$(\langle \mathbf{w}, \mathbf{x} \rangle + b)$。

为了在几何上说明这个假设类，考虑情况 $d = 2$ 是有益的。每个假设形成了一个与向量 $\mathbf{w}$ 垂直的超平面，并在点 $(0, -b/w_2)$ 处与垂直轴相交。位于超平面“上方”的实例，即与 $\mathbf{w}$ 共享锐角的实例，被标记为正。位于超平面“下方”的实例，即与 $\mathbf{w}$ 共享钝角的实例，被标记为负。



在9.1.3节中，我们将展示VCdim$(HS_d) = d + 1$。由此可知，只要样本大小是$\Omega\left(\frac{d+\log(1/\delta)}{\epsilon}\right)$，我们就可以使用ERM范式学习半空间。因此，我们现在讨论如何实现半空间的ERM过程。

以下是两种在可实现情况中找到ERM半空间的解决方案。在半空间的情况下，可实现情况通常被称为“可分离”情况，因为可以使用超平面将所有正例与所有负例分开。实现ERM规则

在不可分情况下（即无知的情形）已知是计算上困难的（Ben-David & Simon 2001）。学习不可分数据有几种方法。最流行的一种是使用
surrogate loss functions，即学习一个半空间，不一定最小化0 – 1 损失，而是相对于不同的损失函数。例如，在第9.3节中，我们将描述逻辑回归方法，即使在不可分情况下也可以有效地实现。我们将在第12章中更详细地研究代理损失函数。

9.1.1 Linear Programming for the Class of Halfspaces

线性规划（LP）是可以表示为在满足线性不等式条件下最大化线性函数的问题。也就是说，

$$\begin{aligned} \max_{\mathbf{w}\in\mathbb{R}^d} \quad & \langle \mathbf{u}, \mathbf{w} \rangle \\ \text{subject to} \quad & A\mathbf{w} \geq \mathbf{v} \end{aligned}$$

在 $\mathbf{w} \in \mathbb{R}^d$ 是我们希望确定的变量向量，A 是一个 $m \times d$ 矩阵，
$\mathbf{v} \in \mathbb{R}^m, \mathbf{u} \in \mathbb{R}^d$ 是向量。线性规划可以高效求解，[1]，并且还有公开可用的 LP 求解器实现。

我们将展示在可实现的情形下，半空间ERM问题可以表示为一个线性规划问题。为了简化，我们假设同质情形。令$S = \{(\mathbf{x}_i, y_i)\}_{i=1}^m$为一个大小为$m$的训练集。由于我们假设可实现的情形，一个ERM预测器应该在训练集上没有错误。也就是说，我们正在寻找某个向量$\mathbf{w} \in \mathbb{R}^d$，使得

$$\operatorname{sign}(\langle \mathbf{w}, \mathbf{x}_i \rangle) = y_i, \qquad \forall i = 1, \ldots, m.$$

等效地，我们正在寻找某个向量 $\mathbf{w}$，使得

$$y_i \langle \mathbf{w}, \mathbf{x}_i \rangle > 0, \qquad \forall i = 1, \ldots, m.$$

设 $\mathbf{w}^*$ 是满足此条件的向量（由于我们假设可实现性，因此它必须存在）。定义 $\gamma =$ 为 $_i(y_i \langle \mathbf{w}^*, \mathbf{x}_i \rangle)$ 的最小值，并令$\bar{\mathbf{w}} = \frac{\mathbf{w}^*}{\gamma}$。因此，对于所有 i，我们有

$$y_i \langle \bar{\mathbf{w}}, x_i \rangle = \frac{1}{\gamma} y_i \langle \mathbf{w}^*, \mathbf{x}_i \rangle \geq 1.$$

我们已经证明了存在一个向量满足

$$y_i \langle \mathbf{w}, \mathbf{x}_i \rangle \geq 1, \qquad \forall i = 1, \ldots, m. \tag{9.1}$$

显然，这样的向量是一个ERM预测器。
为了找到一个满足方程（9.1）的向量，我们可以依靠线性规划求解器如下。将A设为$m \times d$矩阵，其行是实例乘以

1 即，在 m、d 的时间多项式内，以及在实数表示大小内。

由 y_i。即 $A_{i,j} = y_i\, x_{i,j}$，其中 $x_{i,j}$ 是向量 $\mathbf{x}_i$ 的第 j 个元素。令 $\mathbf{v}$ 为向量 $(1, \ldots, 1) \in \mathbb{R}^m$。然后，方程 (9.1) 可以重写为

$$A\mathbf{w} \geq \mathbf{v}.$$

LP形式需要最大化目标，然而所有满足约束的$\mathbf{w}$都是作为输出假设的等价候选者。因此，我们设置了一个“虚拟”目标，$\mathbf{u} = (0, \ldots, 0) \in \mathbb{R}^d$。

9.1.2 Perceptron for Halfspaces

一个ERM规则的另一种实现是Rosenblatt的感知器算法（Rosenblatt 1958）。感知器是一个迭代算法，它构建一个向量序列 {v*}。最初，{v*} 被设置为全零向量。在第 {v*} 迭代中，感知器找到一个被 {v*} 错误标记的例子 {v*}，即一个满足 sign({v*}) {v*} 的例子。然后，感知器通过添加一个由标签 {v*} 缩放的实例 {v*} 来更新 {v*}。也就是说，{v*}。回想一下，我们的目标是让 {v*} 对于所有 {v*} 都等于0，并注意

$$y_i\langle \mathbf{w}^{(t+1)}, \mathbf{x}_i\rangle = y_i\langle \mathbf{w}^{(t)} + y_i\mathbf{x}_i, \mathbf{x}_i\rangle = y_i\langle \mathbf{w}^{(t)}, \mathbf{x}_i\rangle + \|\mathbf{x}_i\|^2.$$

因此，感知机的更新引导解决方案在 i 个示例上“更正确”。

Batch Perceptron

input: A training set $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$
initialize: $\mathbf{w}^{(1)} = (0, \ldots, 0)$
  for $t = 1, 2, \ldots$
    if $(\exists\ i \text{ s.t. } y_i\langle \mathbf{w}^{(t)}, \mathbf{x}_i\rangle \leq 0)$ then
      $\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + y_i\mathbf{x}_i$
    else
      output $\mathbf{w}^{(t)}$

以下定理保证了在可实现的情况下，算法停止时所有样本点都被正确分类。

THEOREM 9.1 *Assume that* $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$ *is separable, let* $B = \min\{\|\mathbf{w}\| : \forall i \in [m],\ \ y_i\langle \mathbf{w}, \mathbf{x}_i\rangle \geq 1\}$, *and let* $R = \max_i \|\mathbf{x}_i\|$. *Then, the Perceptron algorithm stops after at most* $(RB)^2$ *iterations, and when it stops it holds that* $\forall i \in [m],\ \ y_i\langle \mathbf{w}^{(t)}, \mathbf{x}_i\rangle > 0$.

Proof 根据停止条件的定义，如果感知机停止，则它必须已将所有示例分开。我们将证明，如果感知机运行了 T 次迭代，那么我们必定有 $T \leq (RB)^2$，这意味着感知机必须在最多 $(RB)^2$ 次迭代后停止。

设 $\mathbf{w}^\star$ 为在 B 的定义中达到最小值的向量。即，

$y_i\langle \mathbf{w}^\star, x_i\rangle \geq$ 对于所有 i，并且在满足这些约束的所有向量中，$\mathbf{w}^\star$ 的范数是最小的。

证明的想法是展示在执行 T 次迭代后，向量 $\mathbf{w}^\star$ 和 $\mathbf{w}^{(T+1)}$ 之间的余弦角至少为 $\frac{\sqrt{T}}{RB}$。也就是说，我们将展示

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

根据柯西-施瓦茨不等式，方程（9.2）的左边至多为1。因此，方程（9.2）将意味着

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

这将是我们的证明的结论。

为了证明方程（9.2）成立，我们首先证明 $\langle \mathbf{w}^\star, \mathbf{w}^{(T+1)}\rangle \geq T$。实际上，在第一次迭代时，$\mathbf{w}^{(1)} = (0, \ldots, 0)$，因此 $\langle \mathbf{w}^\star, \mathbf{w}^{(1)}\rangle = 0$，而在迭代 t 时，如果我们使用示例（$\mathbf{x}_i, y_i$）进行更新，则有

$$\begin{aligned}\langle \mathbf{w}^\star, \mathbf{w}^{(t+1)}\rangle - \langle \mathbf{w}^\star, \mathbf{w}^{(t)}\rangle &= \langle \mathbf{w}^\star, \mathbf{w}^{(t+1)} - \mathbf{w}^{(t)}\rangle \\ &= \langle \mathbf{w}^\star, y_i\mathbf{x}_i\rangle = y_i\langle \mathbf{w}^\star, \mathbf{x}_i\rangle \\ &\geq 1.\end{aligned}$$

因此，执行 T 次迭代后，我们得到:

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

如需。

接下来，我们上界 $\|\mathbf{w}^{(T+1)}\|$。对于每个迭代 t，我们有

$$\begin{aligned}\|\mathbf{w}^{(t+1)}\|^2 &= \|\mathbf{w}^{(t)} + y_i\mathbf{x}_i\|^2 \\ &= \|\mathbf{w}^{(t)}\|^2 + 2y_i\langle \mathbf{w}^{(t)}, \mathbf{x}_i\rangle + y_i^2\|\mathbf{x}_i\|^2 \\ &\leq \|\mathbf{w}^{(t)}\|^2 + R^2 \end{aligned} \tag{9.4}$$

在最后一个不等式是由于示例 i 必须满足 $y_i\langle \mathbf{w}^{(t)}, \mathbf{x}_i\rangle \leq \geqslant 0$ 的原因，并且 $\mathbf{x}_i$ 的范数不超过 R。现在，由于 $\|\mathbf{w}^{(1)}\|^2 = \geqslant 0$，如果我们对 T 次递归地使用方程 (9.4)，我们得到

$$\|\mathbf{w}^{(T+1)}\|^2 \leq TR^2 \quad \Rightarrow \quad \|\mathbf{w}^{(T+1)}\| \leq \sqrt{T}R. \tag{9.5}$$

结合方程（9.3）和方程（9.5），并利用事实 $\|\mathbf{w}^\star\| = B$，我们得到:

$$\frac{\langle \mathbf{w}^{(T+1)}, \mathbf{w}^\star\rangle}{\|\mathbf{w}^\star\|\,\|\mathbf{w}^{(T+1)}\|} \geq \frac{T}{B\sqrt{T}\,R} = \frac{\sqrt{T}}{B\,R}\,.$$

我们已有 th美證明方程式（9.2）成立，這結論了 你的证明。□

Remark 9.1 感知机易于实现且保证收敛。然而，收敛速度取决于参数 B，在某些情况下，d 中可能呈指数级增大。在这种情况下，最好通过解决线性规划来实现 ERM 问题，如前节所述。尽管如此，对于许多自然数据集，B 的大小并不太大，感知机收敛相当快。

9.1.3 The VC Dimension of Halfspaces

为了计算半空间的VC维数，我们从齐次情况开始。

THEOREM 9.2 *The VC dimension of the class of homogenous halfspaces in* $\mathbb{R}^d$ *is* d.

Proof 首先，考虑向量集 $\mathbf{e}_1, \ldots, \mathbf{e}_d$，其中对于每个 i，向量 $\mathbf{e}_i$ 是除了 i 坐标上的 1 以外全为零的向量。这个集合被齐次半空间类所打碎。确实，对于每个标记 $y_1, \ldots, y_d$，集合 $\mathbf{w} = (y_1, \ldots, y_d)$，然后对于所有 i 的 $\langle \mathbf{w}, \mathbf{e}_i \rangle = y_i$。

接下来，设 $\mathbf{x}_1, \ldots, \mathbf{x}_{d+1}$ 是 $d+$ 个 1 向量在 $\mathbb{R}^d$ 中的集合。那么，必须存在实数 $a_1, \ldots, a_{d+1}$，它们不全为零，使得 $\sum_{i=1}^{d+1} a_i \mathbf{x}_i = \mathbf{0}$。设 $I = \{i : a_i > 0\}$ 和 $J = \{j : a_j < 0\}$。I 或 J 至少有一个非空。我们首先假设它们都是非空的。那么，

$$\sum_{i \in I} a_i \mathbf{x}_i = \sum_{j \in J} |a_j| \mathbf{x}_j.$$

现在，假设 $\mathbf{x}_1, \ldots, \mathbf{x}_{d+1}$ 被同质类所破碎。那么，必须存在一个向量 $\mathbf{w}$，使得对于所有 $i \in I$，有 $\langle \mathbf{w}, \mathbf{x}_i \rangle > 0$，而对于每一个 $j \in J$，有 $\langle \mathbf{w}, \mathbf{x}_j \rangle < 0$。由此可得

$$0 < \sum_{i \in I} a_i \langle \mathbf{x}_i, \mathbf{w} \rangle = \left\langle \sum_{i \in I} a_i \mathbf{x}_i, \mathbf{w} \right\rangle = \left\langle \sum_{j \in J} |a_j| \mathbf{x}_j, \mathbf{w} \right\rangle = \sum_{j \in J} |a_j| \langle \mathbf{x}_j, \mathbf{w} \rangle < 0,$$

这导致矛盾。最后，如果 J (分别，I) 为空，则应将最右侧（分别，最左侧）的不等式替换为等式，这仍然导致矛盾。

□

THEOREM 9.3 *The VC dimension of the class of nonhomogenous halfspaces in* $\mathbb{R}^d$ *is* $d+1$.

Proof 首先，正如在定理9.2的证明中，很容易验证向量集$\mathbf{0}, \mathbf{e}_1, \ldots, \mathbf{e}_d$被非齐次半空间类所打碎。其次，假设向量$\mathbf{x}_1, \ldots, \mathbf{x}_{d+2}$被非齐次半空间类所打碎。但是，使用本章开头我们已展示的约简，可以得出结论，在$\mathbb{R}^{d+1}$中有$d+$个向量被齐次半空间类所打碎。但这与定理9.2矛盾。

□

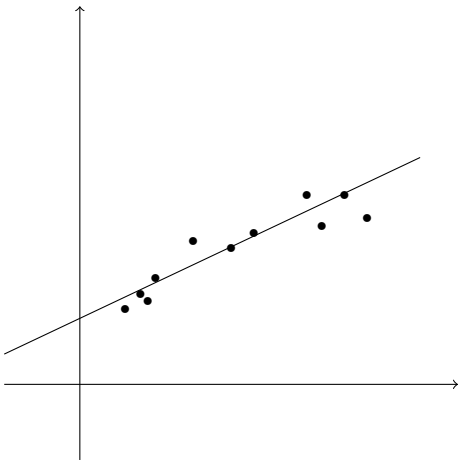


Figure 9.1 线性回归对于 $d = 1$. 例如，x 轴可能表示婴儿的年龄，y 轴表示她的体重。

9.2 Linear Regression

线性回归是一种常见的统计工具，用于建模某些“解释”变量与某些实值结果之间的关系。将其视为一个学习问题，域集 $\mathcal{X}$ 是 $\mathbb{R}^d$ 的子集，对于某些 d，标签集 $\mathcal{Y}$ 是实数集。我们希望学习一个线性函数 $h : \mathbb{R}^d \to \mathbb{R}$，它能最好地近似我们的变量之间的关系（例如，预测婴儿的体重作为其年龄和出生时体重的函数）。图9.1显示了 $d = 1$ 的一个线性回归预测器示例。

线性回归预测器的假设类仅仅是线性函数的集合，

$$\mathcal{H}_{reg} = L_d = \{\mathbf{x} \mapsto \langle \mathbf{w}, \mathbf{x} \rangle + b : \mathbf{w} \in \mathbb{R}^d,\ b \in \mathbb{R}\}.$$

接下来我们需要为回归定义一个损失函数。虽然在分类中损失的定义很简单，因为 $\ell(h, (\mathbf{x}, y))$ 只表明 $h(\mathbf{x})$ 是否正确预测了 y，但在回归中，如果婴儿的体重是 3 kg，那么预测值 3.00001 kg 和 4 kg 都是“错误的”，但我们显然更倾向于前者。因此，我们需要定义我们将在 $h(\mathbf{x})$ 和 y 之间的差异上受到多少“惩罚”。一种常见的方法是使用平方损失函数，即，

$$\ell(h, (\mathbf{x}, y)) = (h(\mathbf{x}) - y)^2.$$

对于这个损失函数，经验风险函数被称为均方误差，即，

$$L_S(h) = \frac{1}{m} \sum_{i=1}^{m} (h(\mathbf{x}_i) - y_i)^2.$$

在下一个小节中，我们将看到如何实现关于平方损失的线性回归的ERM规则。当然，还有许多其他损失函数可以使用，例如，绝对值损失函数$\ell(h,(\mathbf{x},y)) = |h(\mathbf{x}) - y|$。绝对值损失函数的ERM规则可以使用线性规划来实现（参见练习1）。

注意，由于线性回归不是一个二元预测任务，我们不能使用VC维来分析其样本复杂度。对线性回归样本复杂度的一种可能分析是依靠“离散化技巧”（参见第4章注释4.1）；也就是说，如果我们对使用有限数量的位（例如64位浮点表示）表示向量$\mathbf{w}$的每个元素和偏差b表示满意，那么假设类就变为有限的，其大小最多为$2^{64(d+1)}$。现在我们可以依靠第4章中描述的有限假设类的样本复杂度界限。然而，请注意，要应用第4章中的样本复杂度界限，我们还需要损失函数是有界的。本书后面我们将描述更严格的方法来分析回归问题的样本复杂度。

9.2.1 Least Squares

最小二乘法是解决关于线性回归预测器假设类平方损失的ERM问题的算法。对于此类，给定训练集S，并使用L_d的同质版本，ERM问题是要找到

$$\operatorname*{argmin}_{\mathbf{w}} L_S(h_{\mathbf{w}}) = \operatorname*{argmin}_{\mathbf{w}} \frac{1}{m}\sum_{i=1}^{m}(\langle \mathbf{w}, \mathbf{x}_i\rangle - y_i)^2.$$

为了解决这个问题，我们计算目标函数的梯度并将其与零进行比较。也就是说，我们需要解决

$$\frac{2}{m}\sum_{i=1}^{m}(\langle \mathbf{w}, \mathbf{x}_i\rangle - y_i)\mathbf{x}_i = 0.$$

我们可以将问题重新表述为问题 $A\mathbf{w} = \mathbf{b}$，其中

$$A = \left(\sum_{i=1}^{m} \mathbf{x}_i\,\mathbf{x}_i^\top\right) \quad \text{and} \quad \mathbf{b} = \sum_{i=1}^{m} y_i\mathbf{x}_i. \tag{9.6}$$

或者，以矩阵形式表示：

$$A = \begin{pmatrix} \vdots & & \vdots \\ \mathbf{x}_1 & \dots & \mathbf{x}_m \\ \vdots & & \vdots \end{pmatrix} \begin{pmatrix} \vdots & & \vdots \\ \mathbf{x}_1 & \dots & \mathbf{x}_m \\ \vdots & & \vdots \end{pmatrix}^\top , \tag{9.7}$$

$$\mathbf{b} = \begin{pmatrix} \vdots & & \vdots \\ \mathbf{x}_1 & \dots & \mathbf{x}_m \\ \vdots & & \vdots \end{pmatrix} \begin{pmatrix} y_1 \\ \vdots \\ y_m \end{pmatrix} . \tag{9.8}$$

如果 A 是可逆的，那么 ERM 问题的解是

$$\mathbf{w} = A^{-1}\,\mathbf{b}.$$

该情况下 A 不可逆需要线性代数的一些标准工具，这些工具在附录C中可以找到。可以很容易地证明，如果训练实例没有涵盖 $\mathbb{R}^d$ 的整个空间，那么 A 不可逆。尽管如此，由于 $\mathbf{b}$ 在 A 的范围内，我们总能找到系统 $A\mathbf{w} = \mathbf{b}$ 的一个解。实际上，由于 A 是对称的，我们可以用其特征值分解来表示它，即 $A = VDV^\top$，其中 D 是对角矩阵，V 是正交矩阵（即 $V^\top V$ 是单位 $d \times d$ 矩阵）。定义 D^+ 为对角矩阵，使得 $D^+_{i,i} = 0$ 如果 $D_{i,i} = 0$ 否则 $D^+_{i,i} = 1/D_{i,i}$。现在，定义

$$A^+ = VD^+V^\top \quad \text{and} \quad \hat{\mathbf{w}} = A^+\mathbf{b}.$$

让 $\mathbf{v}_i$ 表示 V 的 i 列。然后，我们有

$$A\hat{\mathbf{w}} = AA^+\mathbf{b} = VDV^\top VD^+V^\top\mathbf{b} = VDD^+V^\top\mathbf{b} = \sum_{i:D_{i,i}\neq 0} \mathbf{v}_i\mathbf{v}_i^\top\mathbf{b}.$$

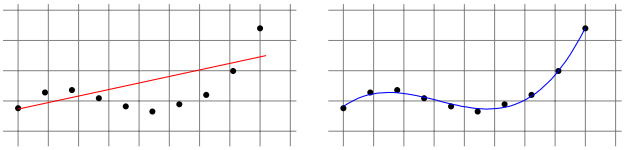
这意味着 $A\hat{\mathbf{w}}$ 是 $\mathbf{b}$ 在那些向量 $\mathbf{v}_i$ 的张量上的投影，其中 $D_{i,i} \neq 0$。由于 $\mathbf{x}_1, \dots, \mathbf{x}_m$ 的线性张量与那些 $\mathbf{v}_i$ 的线性张量相同，并且 $\mathbf{b}$ 在 $\mathbf{x}_i$ 的线性张量中，我们得出 $A\hat{\mathbf{w}} = \mathbf{b}$，这结束了我们的论证。

9.2.2 Linear Regression for Polynomial Regression Tasks

某些学习任务需要非线性预测器，例如多项式预测器。例如，一个一维的次数为 n 的多项式函数，即，

$$p(x) = a_0 + a_1x + a_2x^2 + \cdots + a_nx^n$$

在 $(a_0, \dots, a_n)$ 是一个大小为 $n +$ 的系数向量 1。在以下内容中，我们展示了一个训练集，使用三次多项式预测器比使用线性预测器拟合得更好。



我们将在此关注一类一维、n-次，多项式回归预测器，即，

$$\mathcal{H}^n_{poly} = \{x \mapsto p(x)\},$$

在 p 是一个一维多项式，其阶数为 n，由系数向量 $(a_0, \ldots, a_n)$ 参数化。请注意 $\mathcal{X} = \mathbb{R}$，因为这是一个一维多项式，以及 $\mathcal{Y} = \mathbb{R}$，因为这是一个回归问题。

一种学习此类的方法是将问题简化为线性回归问题，我们已展示了如何解决此问题。为了将多项式回归问题转换为线性回归问题，我们定义映射 ψ：$\mathbb{R} \to \mathbb{R}^{n+1}$ 使得 $\psi(x) = (1, x, x^2, \ldots, x^n)$。然后我们有

$$p(\psi(x)) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n = \langle \mathbf{a}, \psi(x) \rangle$$

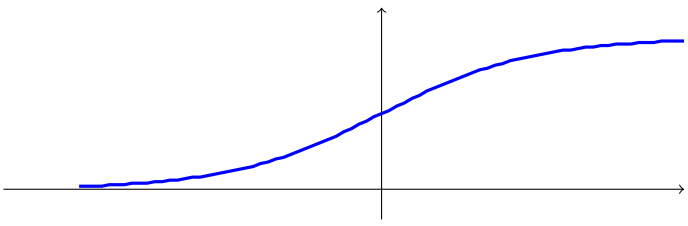
我们可以通过使用前面所示的最小二乘法找到最优系数向量 $\mathbf{a}$。

9.3 Logistic Regression

在逻辑回归中，我们从 $\mathbb{R}^d$ 学习一个从 h 到区间 [0, 1] 的函数族 {v*}。然而，逻辑回归用于分类任务：我们可以将 $h(\mathbf{x})$ 解释为 *probability* 的标签 $\mathbf{x}$ 为 1 的概率。与逻辑回归相关的假设类是sigmoid函数 ϕ_{sig}：$\mathbb{R} \to$ [0, 1] 在线性函数类 L_d 上的组合。特别是，逻辑回归中使用的sigmoid函数是 *logistic function*，定义为

$$\phi_{\text{sig}}(z) = \frac{1}{1 + \exp(-z)}. \tag{9.9}$$

“sigmoid”这个名字意味着“S形”，指的是该函数的图像，如图所示：



假设类因此是（为了简单起见，我们使用同质线性函数）：{v*}

$$H_{\mathrm{sig}} = \phi_{\mathrm{sig}} \circ L_d = \{\mathbf{x} \mapsto \phi_{\mathrm{sig}}(\langle \mathbf{w}, \mathbf{x} \rangle) : \mathbf{w} \in \mathbb{R}^d\}.$$

注意，当 $\langle \mathbf{w}, \mathbf{x} \rangle$ 非常大时，$\phi_{\mathrm{sig}}(\langle \mathbf{w}, \mathbf{x} \rangle)$ 接近 1，而如果 $\langle \mathbf{w}, \mathbf{x} \rangle$ 非常小，则 $\phi_{\mathrm{sig}}(\langle \mathbf{w}, \mathbf{x} \rangle)$ 接近 0。回想一下，对应于向量 $\mathbf{w}$ 的半空间预测是 sign($\langle \mathbf{w}, \mathbf{x} \rangle$)。因此，当 $|\langle \mathbf{w}, \mathbf{x} \rangle|$ 很大时，半空间假设和逻辑假设的预测非常相似。然而，当 $|\langle \mathbf{w}, \mathbf{x} \rangle|$ 接近 0 时，我们有 $\phi_{\mathrm{sig}}(\langle \mathbf{w}, \mathbf{x} \rangle) \approx \frac{1}{2}$。直观上，逻辑假设不确定标签的值，因此它猜测标签是 sign($\langle \mathbf{w}, \mathbf{x} \rangle$)，概率略大于 50%。相比之下，半空间假设始终输出一个确定性的预测，即 1 或 -1，即使 $|\langle \mathbf{w}, \mathbf{x} \rangle|$ 非常接近 0。

接下来，我们需要指定一个损失函数。也就是说，我们应该定义预测某些 $h_{\mathbf{w}}(\mathbf{x}) \in [0, 1]$ 在真实标签是 $y \in \{\pm 1\}$ 的情况下有多糟糕。显然，我们希望当 $y = 1$ 时，$h_{\mathbf{w}}(\mathbf{x})$ 会很大，即预测 -1 的概率会很大，如果 $y = -1$。注意，

$$1 - h_{\mathbf{w}}(\mathbf{x}) = 1 - \frac{1}{1 + \exp(-\langle \mathbf{w}, \mathbf{x} \rangle)} = \frac{\exp(-\langle \mathbf{w}, \mathbf{x} \rangle)}{1 + \exp(-\langle \mathbf{w}, \mathbf{x} \rangle)} = \frac{1}{1 + \exp(\langle \mathbf{w}, \mathbf{x} \rangle)}.$$

因此，任何合理的损失函数都会随着 $\frac{1}{1+\exp(y\langle \mathbf{w}, \mathbf{x} \rangle)}$ 单调增加，或者等价地，会随着 $1 + \exp(-y\langle \mathbf{w}, \mathbf{x} \rangle)$ 单调增加。逻辑回归中使用的逻辑损失函数根据 $1 + \exp(-y\langle \mathbf{w}, \mathbf{x} \rangle)$ 的对数来惩罚 $h_{\mathbf{w}}$（记住，对数是一个单调函数）。也就是说，

$$\ell(h_{\mathbf{w}}, (\mathbf{x}, y)) = \log\left(1 + \exp(-y\langle \mathbf{w}, \mathbf{x} \rangle)\right).$$

因此，给定一个训练集 $S = (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$，与逻辑回归相关的 ERM 问题为

$$\operatorname*{argmin}_{\mathbf{w} \in \mathbb{R}^d} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 + \exp(-y_i \langle \mathbf{w}, \mathbf{x}_i \rangle)\right). \tag{9.10}$$

对数损失函数的优势在于它相对于 $\mathbf{w}$ 是一个 *convex* 函数；因此可以使用标准方法有效地解决 ERM 问题。我们将在后面的章节中研究如何使用凸函数进行学习，并特别指定一个用于最小化凸函数的简单算法。

与逻辑回归（方程（9.10））相关的ERM问题与寻找最大似然估计量的问题相同，这是一个寻找最大化给定数据集联合概率的参数的已知统计方法，假设一个特定的参数概率函数。我们将在第24章研究最大似然方法。

9.4 Summary

线性预测器的家族是假设类中最有用的家族之一，许多在实践中广泛使用的学习算法都依赖于线性预测器。我们已经展示了在可分情况下相对于零一损失学习线性预测器的有效算法，以及在不可实现情况下相对于平方损失和对数损失的学习算法。在后面的章节中，我们将介绍损失函数的性质，这些性质使得有效的学习成为可能。

自然地，当我们假设某些线性预测器相对于潜在分布具有低风险时，线性预测器总是有效的。在下一章中，我们将展示如何通过在简单类别之上组合线性预测器来构建非线性预测器。这将使我们能够针对各种先验知识假设使用线性预测器。

9.5 Bibliographic Remarks

感知机算法可追溯至罗森布拉特（1958年）。其收敛速度的证明归功于（Agmon 1954，Novikoff 1962）。最小二乘回归可追溯至高斯（1795年）、勒让德（1805年）和阿德里安（1808年）。

9.6 Exercises

1. 展示如何将关于绝对值损失函数 $\ell(h,(\mathbf{x},y)) = |h(\mathbf{x}) - y|$ 的线性回归的 ERM 问题表示为线性规划；即展示如何编写该问题

$$\min_{\mathbf{w}} \sum_{i=1}^{m} |\langle \mathbf{w}, \mathbf{x}_i \rangle - y_i|$$

作为一个线性规划。
Hint: 开始证明对于任何 $c \in \mathbb{R}$,

$$|c| \;=\; \min_{a \geq 0} a \;\; \text{s.t.} \;\; c \leq a \text{ and } c \geq -a.$$

2. 证明方程（9.6）中定义的矩阵 A 可逆，当且仅当 $\mathbf{x}_1, \ldots, \mathbf{x}_m$ 张成 $\mathbb{R}^d$。

3. 证明定理9.1在以下意义上是紧的：对于任何正整数 m，存在一个向量 $\mathbf{w}^* \in \mathbb{R}^d$（（对于某些适当的 d））和一个示例序列 $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)\}$，使得以下条件成立：

- $R = \max_i \|\mathbf{x}_i\| \leq 1$.
- $\|\mathbf{w}^*\|2 = m$，并且对于所有 $i \leq m$，$y_i \langle \mathbf{x}_i, \mathbf{w}^* \rangle \geq 1$. 注意，使用定理9.1中的符号，因此我们得到

$$B = \min\{\|\mathbf{w}\| : \forall i \in [m], \;\; y_i \langle \mathbf{w}, x_i \rangle \geq 1\} \leq \sqrt{m}.$$

因此，$(BR)^2 \leq m$。

- 在运行感知机于这个示例序列时，它在收敛前进行了 m 次更新。

Hint: 选择 $d = m$，并对每个 i 选择 $\mathbf{x}_i = \mathbf{e}_i$。

4. (*) 对于任意一个数 m，找到一个标记示例序列 $((\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)) \in (\mathbb{R}^3 \times \{-1, +1\})^m$，使得定理9.1的上界等于 m，并且感知机算法必然犯 m 个错误。

Hint: 将每个 $\mathbf{x}_i$ 设置为形式为 (a, b, y_i) 的三维向量，其中 $a^2 + b^2 = R^2 - 1$。令 $\mathbf{w}^*$ 为向量 (0, 0, 1)。现在，回顾感知机上界（定理9.1）的证明，看看我们是在哪里使用了不等式（$\leq$）而不是等式（$=$），并找出不等式实际上等于等式的情况。

5. 假设我们按如下方式修改感知器算法：在更新步骤中，每次我们犯错误时，不是执行 $\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + y_i \mathbf{x}_i$，而是对某些 $\eta > 0$ 执行 $\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + \eta y_i \mathbf{x}_i$。证明修改后的感知器将执行与原始感知器相同数量的迭代次数，并且将收敛到一个与原始感知器输出指向相同方向的向量。

6. 在这个问题中，我们将得到 $\mathbb{R}^d$ 中（闭）球类的 VC 维度的界限，即，

$$\mathcal{B}_d = \{B_{\mathbf{v},r} : \mathbf{v} \in \mathbb{R}^d, r > 0\},$$

哪里

$$B_{\mathbf{v},r}(\mathbf{x}) = \begin{cases} 1 & \text{if } \|\mathbf{x} - \mathbf{v}\| \leq r \\ 0 & \text{otherwise} \end{cases}.$$

1. 考虑由 $\phi(\mathbf{x}) = (\mathbf{x}, \|\mathbf{x}\|^2)$ 定义的映射 $\phi : \mathbb{R}^d \to \mathbb{R}^{d+1}$。证明如果 $\mathbf{x}_1, \ldots, \mathbf{x}_m$ 被 $\mathcal{B}_d$ 破碎，那么 $\phi(\mathbf{x}_1), \ldots, \phi(\mathbf{x}_m)$ 被该问题中 $\mathbb{R}^{d+1}$ (中的半空间类所破碎。在此问题中，我们假设 sign(0) = 1)。这告诉我们关于 VCdim($\mathcal{B}_d$) 的什么?

2. (*) 找到一个由 $d+1$ 个点组成的集合 $\mathbb{R}^d$，该集合被 $\mathcal{B}_d$ 粉碎。得出结论：

$$d + 1 \leq \text{VCdim}(\mathcal{B}_d) \leq d + 2.$$

10 Boosting

提升是一种从理论问题中发展出来的算法范式，并成为了一种非常实用的机器学习工具。提升方法使用线性预测器的推广来解决本书中较早提出的前两个主要问题。第一个是偏差-复杂度权衡。我们在第5章中看到，ERM学习者的误差可以分解为*approximation error*和*estimation error*的和。学习者搜索的假设类越有表现力，近似误差就越小，但估计误差就越大。因此，学习者面临在考虑这两个因素之间选择一个好的权衡的问题。提升范式允许学习者在权衡中拥有平滑的控制。学习从基本类（可能具有较大的近似误差）开始，随着学习的进展，预测者可能属于的类越来越丰富。

第二个问题是提升方法解决的计算复杂性问题。如第8章所示，对于许多有趣的概念类，找到ERM假设的任务可能是计算上不可行的。提升算法放大了*weak learners*的准确性。直观上，可以将弱学习器视为一种使用简单的“经验法则”来输出来自易于学习的假设类并仅略好于随机猜测的算法。当弱学习器可以高效实现时，提升提供了一种将此类弱假设聚合起来，以近似更大、更难学习的类的好预测器的工具。

在这一章中，我们将描述和分析一个实际有用的提升算法，AdaBoost（自适应提升的简称）。AdaBoost算法输出一个假设，它是简单假设的线性组合。换句话说，AdaBoost依赖于通过在简单类之上组合线性预测器获得的假设类族。我们将展示AdaBoost使我们能够通过改变单个参数来控制逼近误差和估计误差之间的权衡。

AdaBoost展示了在本书后面会再次出现的主题，即通过将它们组合在其他函数之上来扩展线性预测器的表达能力。这将在第10.3节中详细阐述。

AdaBoost起源于关于是否可以将高效的弱学习器“提升”为高效的强学习器的理论问题。这个问题被提出

1988年由Kearns和Valiant提出，1990年由当时在麻省理工学院的罗伯特·沙皮尔解决。然而，提出的机制并不十分实用。1995年，罗伯特·沙皮尔和Yoav Freund提出了AdaBoost算法，这是第一个真正实用的提升算法。这个简单而优雅的算法变得非常流行，Freund和Schapire的工作获得了众多奖项的认可。

此外，提升是一种很好的学习理论实际影响的例子。虽然提升最初是一个纯粹的理论问题，但它导致了流行且广泛使用的算法。确实，正如我们将在本章后面所展示的，AdaBoost已被成功用于学习在图像中检测人脸。

10.1 Weak Learnability

回忆第3章中给出的PAC学习定义：如果一个假设类$\mathcal{H}$是PAC可学习的，那么存在$m_{\mathcal{H}}: (0,1)^2 \to \mathbb{N}$和一个具有以下性质的学习算法：对于每个$\epsilon, \delta \in (0,1)$，对于每个在$\mathcal{X}$上的分布$\mathcal{D}$，以及对于每个标签函数$f: \mathcal{X} \to \{\pm 1\}$，如果关于$\mathcal{H}, \mathcal{D}, f$的可实现假设成立，那么当在由$\mathcal{D}$生成并由$f$标记的独立同分布示例$m \geq m_{\mathcal{H}}(\epsilon, \delta)$上运行学习算法时，算法返回一个假设$h$，使得至少以$1-\delta$的概率$L_{(\mathcal{D},f)}(h) \leq \epsilon$。

此外，学习理论的基本定理（第6章第6.8定理）描述了可学习类族，并指出每个PAC可学习类都可以使用任何ERM算法进行学习。然而，PAC学习的定义和学习理论的基本定理忽略了学习的计算方面。实际上，正如我们在第8章所展示的，在某些情况下，实现ERM规则是计算上困难的（即使在可实现的情形下）。

然而，也许我们可以用计算难度换取对准确性的要求。给定一个分布 $\mathcal{D}$ 和一个目标标记函数 f，可能存在一个计算效率高的学习算法，其错误率仅略优于随机猜测？这促使我们提出以下定义。

DEFINITION 10.1 (γ-弱可学习性)

- 一个学习算法 A 是一个针对类别 $\mathcal{H}$ 的 γ-弱学习器，如果存在一个函数 $m_{\mathcal{H}}: (0,1) \to \mathbb{N}$，使得对于每个 $\delta \in (0,1)$，对于每个在 $\mathcal{X}$ 上的分布 $\mathcal{D}$，以及对于每个标签函数 $f: \mathcal{X} \to \{\pm 1\}$，如果关于 $\mathcal{H}, \mathcal{D}, f$ 的可实现假设成立，那么在 $m \geq m_{\mathcal{H}}(\delta)$ 上运行学习算法，对由 $\mathcal{D}$ 生成并由 f 标记的独立同分布示例进行学习时，算法返回一个假设 h，使得至少以概率 $1-\delta$，$L_{(\mathcal{D},f)}(h) \leq 1/2 - \gamma$。
- 一个假设类 $\mathcal{H}$ 是 γ-弱可学习的，如果存在一个 γ-弱学习器用于该类。

这个定义几乎与PAC学习的定义相同，在这里我们将其称为*strong learning*，但有一个关键的区别：强可学习意味着能够找到一个任意好的分类器（对于任意小的$\epsilon > 0$，错误率最多为ϵ）。然而，在弱可学习中，我们只需要输出一个错误率最多为$1/2 - \gamma$的假设，即错误率略好于随机标记给出的错误率。希望这可能比提出有效的（完整）PAC学习器更容易。

学习的基本定理（定理6.8）表明，如果一个假设类$\mathcal{H}$具有VC维d，那么PAC学习的样本复杂度$\mathcal{H}$满足$m_{\mathcal{H}}(\epsilon, \delta) \geq C_1 \frac{d+\log(1/\delta)}{\epsilon}$，其中$C_1$是一个常数。应用此定理，结合$\epsilon = 1/2 - \gamma$，我们立即得到，如果$d = \infty$，那么$\mathcal{H}$不是$\gamma$-弱可学习的。这表明从统计学的角度来看（即，如果我们忽略计算复杂度），弱可学习也由$\mathcal{H}$的VC维来表征，因此与PAC（强）学习一样困难。然而，当我们考虑计算复杂度时，弱学习的潜在优势可能是存在一个满足弱学习要求的算法，并且可以高效地实现。

一种可能的方法是采用一个“简单”的假设类，记为 B，并应用相对于 B 的 ERM 作为弱学习算法。为了使其工作，我们需要 B 满足两个要求：

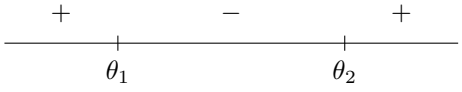
- ERM$_B$是高效可实现的。
- 对于每个由 $\mathcal{H}$ 中的某个假设标记的样本，任何 ERM$_B$ 假设的错误最多为 $1/2 - \gamma$。

然后，立即的问题是，我们能否将一个 *efficient* 弱学习器提升为 *efficient* 强学习器。在下一节中，我们将展示这确实是可能的，但在那之前，让我们举一个例子，说明使用基假设类 B 可以实现类别 $\mathcal{H}$ 的有效弱学习。

Example 10.1 (弱学习三段分类器使用决策树桩) 设 $\mathcal{X} = \mathbb{R}$ 和设 $\mathcal{H}$ 为三段分类器的类别，即 $\mathcal{H} = \{h_{\theta_1,\theta_2,b} : \theta_1, \theta_2 \in \mathbb{R}, \theta_1 < \theta_2, b \in \{\pm 1\}\}$，其中对于每个 x，

$$h_{\theta_1,\theta_2,b}(x) = \begin{cases} +b & \text{if } x < \theta_1 \text{ or } x > \theta_2 \\ -b & \text{if } \theta_1 \leq x \leq \theta_2 \end{cases}$$

一个示例假设（对于 $b = 1$）如下所示：



让 B 成为决策树桩的类别，即 $B = \{x \mapsto \text{sign}(x - \theta) \cdot b : \theta \in \mathbb{R}, b \in \{\pm 1\}\}$。在以下内容中，我们表明 ERM$_B$ 是 γ 弱学习器对于 $\mathcal{H}$，对于 $\gamma = 1/12$。

要看到这一点，我们首先证明对于与 $\mathcal{H}$ 一致的每个分布，都存在一个具有 $L_{\mathcal{D}}(h) \leq 1/3$ 的决策树。实际上，只需注意 $\mathcal{H}$ 中的每个分类器都由三个区域（两个无界射线和一个中心区间）组成，这些区域具有交替的标签。对于这样的两个区域，存在一个决策树与这两个组件的标签一致。注意，对于在 $\mathbb{R}$ 上的每个分布 $\mathcal{D}$ 和将直线划分为三个这样的区域的每个划分，这些区域中必须有一个区域的 $\mathcal{D}$-权重不超过 1/3。设 $h \in \mathcal{H}$ 为零错误假设。一个仅在这样区域上与 h 不一致的决策树具有最多 1/3 的错误。

最后，由于决策树桩的VC维数为2，如果样本大小大于$\Omega(\log(1/\delta)/\epsilon^2)$，那么以至少$1-\delta$的概率，ERM$_B$规则返回的假设错误不超过$1/3+\epsilon$。将$\epsilon = 1/12$，我们得到ERM$_B$的错误不超过$1/3 + 1/12 = 1/2 - 1/12$。

我们注意到 ERM$_B$ 是 γ 弱学习器对于 $\mathcal{H}$。接下来，我们展示如何有效地实现决策树桩的 ERM 规则。

10.1.1 Efficient Implementation of ERM for Decision Stumps

让 $\mathcal{X} = \mathbb{R}^d$，并考虑基于 $\mathbb{R}^d$ 的基假设类决策树，即，

$$\mathcal{H}_{\mathrm{DS}} = \{\mathbf{x} \mapsto \operatorname{sign}(\theta - x_i) \cdot b : \ \theta \in \mathbb{R}, i \in [d], b \in \{\pm 1\}\}.$$

为了简单起见，假设 $b = 1$；也就是说，我们关注 $\mathcal{H}_{\mathrm{DS}}$ 中形式为 sign($\theta - x_i$) 的所有假设。令 $S = ((\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m))$ 为一个训练集。我们将展示如何实现一个 ERM 规则，即如何找到一个最小化 $L_S(h)$ 的决策树。此外，由于在下一节中我们将展示 AdaBoost 需要找到一个相对于 S 上某个分布具有小风险的假设，因此我们将在此展示如何最小化此类风险函数。具体来说，令 $\mathbf{D}$ 为 $\mathbb{R}^m$ (中的一个概率向量，即 $\mathbf{D}$ 的所有元素都是非负的，且 $\sum_i D_i = 1$)。我们后面描述的弱学习器接收 $\mathbf{D}$ 和 S 并输出一个决策树 $h : \mathcal{X} \to \mathcal{Y}$，该决策树最小化相对于 $\mathbf{D}$ 的风险。

$$L_{\mathbf{D}}(h) = \sum_{i=1}^{m} D_i \mathbb{1}_{[h(\mathbf{x}_i) \neq y_i]}.$$

注意，如果 $\mathbf{D} = (1/m, \ldots, 1/m)$，那么 $L_{\mathbf{D}}(h) = L_S(h)$。

回忆一下，每个决策树桩由一个索引 $j \in [d]$ 和一个阈值 θ 参数化。因此，最小化 $L_{\mathbf{D}}(h)$ 等于解决以下问题

$$\min_{j \in [d]} \ \min_{\theta \in \mathbb{R}} \left(\sum_{i : y_i = 1} D_i \mathbb{1}_{[x_{i,j} > \theta]} + \sum_{i : y_i = -1} D_i \mathbb{1}_{[x_{i,j} \leq \theta]} \right). \tag{10.1}$$

修复 $j \in [d]$，并让我们对示例进行排序，以便 $x_{1,j} \leq x_{2,j} \leq \ldots \leq x_{m,j}$。定义 $\Theta_j = \{\frac{x_{i,j} + x_{i+1,j}}{2} : \ i \in [m-1]\} \cup \{(x_{1,j} - 1), (x_{m,j} + 1)\}$。请注意，对于任何 $\theta \in \mathbb{R}$，都存在 $\theta' \in \Theta_j$，它对样本 S 的预测与

阈值 θ。因此，我们可以在 $\theta \in \Theta_j$ 上进行最小化，而不是在 $\theta \in \mathbb{R}$ 上。

这已经给我们提供了一个有效的方法：选择使方程（10.1）的目标值最小的 $j \in [d]$ 和 $\theta \in \Theta_j$。对于每一个 j 和 $\theta \in \Theta_j$，我们都需要计算一个关于 m 个样本的总和；因此，这种方法的时间复杂度将是 $O(dm^2)$。接下来，我们展示一个简单的技巧，使我们能够在 $O(dm)$ 时间内最小化目标值。

观察如下。假设我们已经计算了 $\theta \in (x_{i-1,j}, x_{i,j})$ 的目标函数。令 $F(\theta)$ 为目标函数的值。然后，当我们考虑 $\theta' \in (x_{i,j}, x_{i+1,j})$ 时，我们有

$$F(\theta') = F(\theta) - D_i \mathbb{1}_{[y_i=1]} + D_i \mathbb{1}_{[y_i=-1]} = F(\theta) - y_i D_i.$$

因此，在给定前一个阈值处的目标函数 θ 的情况下，我们可以以恒定的时间计算出 θ' 处的目标函数。这意味着在预处理步骤中，我们根据每个坐标对示例进行排序后，最小化问题可以在时间 $O(dm)$ 内完成。这产生了以下伪代码。

ERM for Decision Stumps

input:
  training set $S = (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$
  distribution vector $\mathbf{D}$
goal: Find $j^\star, \theta^\star$ that solve Equation (10.1)
initialize: $F^\star = \infty$
for $j = 1, \ldots, d$
  sort S using the j'th coordinate, and denote
    $x_{1,j} \leq x_{2,j} \leq \cdots \leq x_{m,j} \leq x_{m+1,j} \stackrel{\text{def}}{=} x_{m,j} + 1$
  $F = \sum_{i:y_i=1} D_i$
  if $F < F^\star$
    $F^\star = F$, $\theta^\star = x_{1,j} - 1$, $j^\star = j$
  for $i = 1, \ldots, m$
    $F = F - y_i D_i$
    if $F < F^\star$ and $x_{i,j} \neq x_{i+1,j}$
      $F^\star = F$, $\theta^\star = \frac{1}{2}(x_{i,j} + x_{i+1,j})$, $j^\star = j$
output $j^\star, \theta^\star$

10.2 AdaBoost

AdaBoost（简称自适应提升）是一种算法，它能够访问弱学习器并找到一个具有低经验风险的假设。AdaBoost算法接收一个示例训练集 $S = (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$作为输入，其中对于每个$i$，存在一个标签函数$f$，使得$y_i = f(\mathbf{x}_i)$。提升过程按一系列连续的回合进行。在第$t$回合，提升器首先定义

一个在 S 上的分布，表示为 $\mathbf{D}^{(t)}$。也就是说，$\mathbf{D}^{(t)} \in \mathbb{R}_+^m$ 和 $\sum_{i=1}^m D_i^{(t)} = 1$。然后，增强器将分布 $\mathbf{D}^{(t)}$ 和样本 S 传递给弱学习器。（这样，弱学习器可以根据 $\mathbf{D}^{(t)}$ 和 f 构建独立同分布的示例。）假设弱学习器返回一个“弱”假设，h_t，其错误，

$$\epsilon_t \stackrel{\text{def}}{=} L_{\mathbf{D}^{(t)}}(h_t) \stackrel{\text{def}}{=} \sum_{i=1}^m D_i^{(t)} \mathbb{1}_{[h_t(\mathbf{x}_i) \neq y_i]},$$

最多为 $\frac{1}{2} - \gamma$ (，当然，弱学习器失败的概率最多为 δ。然后，AdaBoost 按如下方式为 h_t 分配权重：$w_t = \frac{1}{2} \log\left(\frac{1}{\epsilon_t} - 1\right)$。也就是说，$h_t$ 的权重与 h_t 的误差成反比。在回合结束时，AdaBoost 更新分布，使得 h_t 出错的示例将获得更高的概率质量，而 h_t 正确的示例将获得更低的风险质量。直观上，这将迫使弱学习者在下一轮中关注有问题的示例。AdaBoost 算法的输出是基于所有弱假设加权总和的“强”分类器。AdaBoost 的伪代码如下所示。

AdaBoost

input:
training set $S = (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$
weak learner WL
number of rounds T
initialize $\mathbf{D}^{(1)} = (\frac{1}{m}, \ldots, \frac{1}{m})$.
for $t = 1, \ldots, T$:
invoke weak learner $h_t = \mathrm{WL}(\mathbf{D}^{(t)}, S)$
compute $\epsilon_t = \sum_{i=1}^m D_i^{(t)} \mathbb{1}_{[y_i \neq h_t(\mathbf{x}_i)]}$
let $w_t = \frac{1}{2} \log\left(\frac{1}{\epsilon_t} - 1\right)$
update $D_i^{(t+1)} = \frac{D_i^{(t)} \exp(-w_t y_i h_t(\mathbf{x}_i))}{\sum_{j=1}^m D_j^{(t)} \exp(-w_t y_j h_t(\mathbf{x}_j))}$ for all $i = 1, \ldots, m$
output the hypothesis $h_s(\mathbf{x}) = \mathrm{sign}\left(\sum_{t=1}^T w_t h_t(\mathbf{x})\right)$.

以下定理表明，输出假设的训练误差随着提升轮数的增加而指数级快速下降。

THEOREM 10.2 *Let S be a training set and assume that at each iteration of AdaBoost, the weak learner returns a hypothesis for which $\epsilon_t \leq 1/2 - \gamma$. Then, the training error of the output hypothesis of AdaBoost is at most*

$$L_S(h_s) = \frac{1}{m} \sum_{i=1}^m \mathbb{1}_{[h_s(\mathbf{x}_i) \neq y_i]} \;\leq\; \exp(-2\,\gamma^2\,T)\ .$$

Proof 对于每个 t，表示 $f_t = \sum_{p \leq t} w_p h_p$。因此，AdaBoost 的输出

是 f_T。此外，表示

$$Z_t = \frac{1}{m}\sum_{i=1}^{m} e^{-y_i f_t(x_i)}.$$

注意，对于任何假设，我们都有 $1_{[h(x)\neq y]} \leq e^{-yh(x)}$。因此，$L_S(f_T) \leq Z_T$，所以只需证明 $Z_T \leq e^{-2\gamma^2 T}$。为了上界 Z_T，我们将其重写为

$$Z_T = \frac{Z_T}{Z_0} = \frac{Z_T}{Z_{T-1}} \cdot \frac{Z_{T-1}}{Z_{T-2}} \cdots \frac{Z_2}{Z_1} \cdot \frac{Z_1}{Z_0}, \tag{10.2}$$

在何处我们使用了事实 $Z_0 = 1$ 因为 $f_0 \equiv 0$。因此，只需证明对于每一轮 t，

$$\frac{Z_{t+1}}{Z_t} \leq e^{-2\gamma^2}. \tag{10.3}$$

为了做到这一点，我们首先注意到，通过一个简单的归纳论证，对于所有 t 和 i，

$$D_i^{(t+1)} = \frac{e^{-y_i f_t(x_i)}}{\sum_{j=1}^{m} e^{-y_j f_t(x_j)}}.$$

因此，

$$\begin{aligned}
\frac{Z_{t+1}}{Z_t} &= \frac{\sum_{i=1}^{m} e^{-y_i f_{t+1}(x_i)}}{\sum_{j=1}^{m} e^{-y_j f_t(x_j)}} \\
&= \frac{\sum_{i=1}^{m} e^{-y_i f_t(x_i)} e^{-y_i w_{t+1} h_{t+1}(x_i)}}{\sum_{j=1}^{m} e^{-y_j f_t(x_j)}} \\
&= \sum_{i=1}^{m} D_i^{(t+1)} e^{-y_i w_{t+1} h_{t+1}(x_i)} \\
&= e^{-w_{t+1}} \sum_{i:y_i h_{t+1}(x_i)=1} D_i^{(t+1)} + e^{w_{t+1}} \sum_{i:y_i h_{t+1}(x_i)=-1} D_i^{(t+1)} \\
&= e^{-w_{t+1}}(1-\epsilon_{t+1}) + e^{w_{t+1}}\epsilon_{t+1} \\
&= \frac{1}{\sqrt{1/\epsilon_{t+1}-1}}(1-\epsilon_{t+1}) + \sqrt{1/\epsilon_{t+1}-1}\;\epsilon_{t+1} \\
&= \sqrt{\frac{\epsilon_{t+1}}{1-\epsilon_{t+1}}}(1-\epsilon_{t+1}) + \sqrt{\frac{1-\epsilon_{t+1}}{\epsilon_{t+1}}}\epsilon_{t+1} \\
&= 2\sqrt{\epsilon_{t+1}(1-\epsilon_{t+1})}.
\end{aligned}$$

根据我们的假设，$\epsilon_{t+1} \leq \frac{1}{2} - \gamma$。由于函数 $g(a) = a(1-a)$ 在 $[0, 1/2]$ 上单调递增，我们得到{v*} 保持不变。

$$2\sqrt{\epsilon_{t+1}(1-\epsilon_{t+1})} \leq 2\sqrt{\left(\frac{1}{2}-\gamma\right)\left(\frac{1}{2}+\gamma\right)} = \sqrt{1-4\gamma^2}.$$

最后，使用不等式 $1-a \leq e^{-a}$，我们得到 $\sqrt{1-4\gamma^2} \leq e^{-4\gamma^2/2} = e^{-2\gamma^2}$。这表明方程（10.3）成立，从而得出我们的证明结论。 □

每次AdaBoost迭代涉及$O(m)$个操作以及一次对弱学习器的调用。因此，如果弱学习器可以高效实现（如决策树桩相对于ERM的情况），则整个训练过程将是高效的。

Remark 10.2 定理10.2假设在AdaBoost的每次迭代中，弱学习器返回的假设的加权样本误差最多为$1/2-\gamma$。根据弱学习器的定义，它有概率δ失败。使用并集界，弱学习器在所有迭代中都不会失败的概率至少为$1-\delta T$。正如我们在练习1中所示，样本复杂度对δ的依赖性总是可以以$1/\delta$的对数形式存在，因此使用非常小的δ作为弱学习器不是问题。因此，我们可以假设δT也很小。此外，由于弱学习器仅在训练集上的分布中应用，在许多情况下，我们可以实现弱学习器，使其具有零失败概率（即$\delta = 0$）。例如，在先前的章节中描述的寻找决策树最小值的弱学习器中就是这种情况。

定理10.2告诉我们，由AdaBoost构建的假设的经验风险随着T的增长趋于零。然而，我们真正关心的是输出假设的真实风险。为了论证真实风险，我们注意到AdaBoost的输出实际上是由弱学习器构建的T个弱假设的预测上的半空间组成。在下一节中，我们表明，如果弱假设来自低VC维度的基假设类，那么AdaBoost的估计误差将很小；也就是说，AdaBoost输出的真实风险将不会与其经验风险相差很远。

10.3 Linear Combinations of Base Hypotheses

如前所述，构建弱学习器的一种流行方法是针对基假设类应用ERM规则（例如，决策树桩上的ERM）。我们还看到，提升算法输出的是弱假设预测的半空间组合。因此，给定基假设类B (，例如决策树桩)，AdaBoost的输出将是以下类的一个成员：

$$L(B,T) = \left\{ x \mapsto \operatorname{sign}\left(\sum_{t=1}^{T} w_t h_t(x) \right) : \mathbf{w} \in \mathbb{R}^T,\ \forall t, \quad h_t \in B \right\}. \tag{10.4}$$

这意味着每个 $h \in L(B,T)$ 都由 T 个基假设从 B 和一个向量 $w \in \mathbb{R}^T$ 参数化。在实例 x 上对这样的 h 的预测是通过首先应用 T 个基假设来构建向量 $\psi(x) =$ 来获得的。

$(h_1(x), \ldots, h_T(x)) \in \mathbb{R}^T$，然后应用由 $\mathbf{w}$ 定义在 $\psi(x)$ 上的（齐次的）半空间。

本节中，我们通过将 $L(B,T)$ 的 VC 维度用 B 和 T 的 VC 维度来界定，分析了 $L(B,T)$ 的估计误差。我们将证明，在对数因子范围内，$L(B,T)$ 的 VC 维度被 T 倍的 B 的 VC 维度所界定。因此，Ad-aBoost 的估计误差随着 T 线性增长。另一方面，AdaBoost 的经验风险随着 T 减小。实际上，正如我们稍后所证明的，T 可以用来减少 $L(B,T)$ 的近似误差。因此，AdaBoost 的参数 T 使我们能够控制偏差-复杂度权衡。

为了展示 $L(B,T)$ 的表达能力如何随着 T 的增加而增强，考虑一个简单的例子，其中 $\mathcal{X} = \mathbb{R}$ 和基类是决策树桩，

$$\mathcal{H}_{\mathrm{DS1}} = \{x \mapsto \operatorname{sign}(x - \theta) \cdot b : \quad \theta \in \mathbb{R}, b \in \{\pm 1\}\}.$$

请注意，在这个一维情况下，$\mathcal{H}_{\mathrm{DS1}}$实际上等同于$\mathbb{R}$上的（非齐次）半空间。

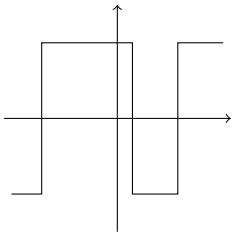
现在，令 $\mathcal{H}$ 为相对复杂的类（与线上的半空间相比），即分段常数函数类。令 g_r 为至多 r 段的分段常数函数；也就是说，存在阈值 $-\infty = \theta_0 < \theta_1 < \theta_2 < \cdots < \theta_r = \infty$ 使得

$$g_r(x) = \sum_{i=1}^{r} \alpha_i \mathbb{1}_{[x \in (\theta_{i-1}, \theta_i]]} \quad \forall i, \quad \alpha_i \in \{\pm 1\}.$$

用 $\mathcal{G}_r$ 表示所有至多具有 r 个片段的此类分段常数分类器的类。

在以下内容中，我们表明 $\mathcal{G}_T \subseteq L(\mathcal{H}_{\mathrm{DS1}}, T)$；即，在 T 上的半空间决策树生成所有最多有 T 个片段的分段常数分类器。

确实，不失一般性，考虑任意的 $g \in \mathcal{G}_T$，其中 $\alpha_t = (-1)^t$。这意味着如果 x 在区间 $(\theta_{t-1}, \theta_t]$ 中，那么 $g(x) = (-1)^t$。例如：



现在，函数

$$h(x) = \operatorname{sign}\left(\sum_{t=1}^{T} w_t \operatorname{sign}(x - \theta_{t-1})\right), \tag{10.5}$$

在 $w_1 = 0.5$ 且对于 $t > 1, w_t = (-1)^t$，它在 $L(\mathcal{H}_{\mathrm{DS1}}, T)$ 中，等于 g（参见练习 2）。

从本例中我们得到，$L(\mathcal{H}_{\mathrm{DS1}}, T)$可以破坏$T+1$个实例的任何集合在$\mathbb{R}$中；因此，$L(\mathcal{H}_{\mathrm{DS1}}, T)$的VC维至少为$T$+1。因此，$T$是一个可以控制偏差-复杂度权衡的参数：扩大$T$会产生一个更具表达力的假设类，但另一方面可能会增加估计误差。在下一个小节中，我们将正式为任何基类B的上界VC维$L(B, T)$。

10.3.1 The VC-Dimension of $L(B, T)$

以下引理告诉我们，$L(B, T)$ 的 VC 维度被$\tilde{O}$(VCdim(B) T) 上界限制（$\tilde{O}$ 符号忽略常数和对数因子）。

LEMMA 10.3 *Let B be a base class and let $L(B, T)$ be as defined in Equation (10.4). Assume that both T and* VCdim(B) *are at least* 3. *Then,*

$$\mathrm{VCdim}(L(B, T)) \leq T\,(\mathrm{VCdim}(B) + 1)\,(3\log(T\,(\mathrm{VCdim}(B) + 1)) + 2).$$

Proof 表示 d = VCdim(B)。令 $C = \{x_1, \ldots, x_m\}$ 为被 $L(B, T)$ 粉碎的集合。C 通过 $h \in L(B, T)$ 标记的每个标记都是通过首先选择 $h_1, \ldots, h_T \in B$ 然后在向量 $(h_1(x), \ldots, h_T(x))$ 上应用半空间假设获得的。根据 Sau er 定理，由 B 在 C 上诱导的最多有 $(em/d)^d$ 个不同的二分法（即标记）。因此，我们需要从最多 $(em/d)^d$ 个不同的假设中选择 T 个假设。最多有 $(em/d)^{dT}$ 种方法来做这件事。接下来，对于每个这样的选择，我们应用一个线性预测器，这会产生最多 $(em/T)^T$ 个二分法。因此，我们可以构建的二分法的总数最多被限制为

$$(em/d)^{dT}\,(em/T)^T \leq m^{(d+1)T},$$

在式中，我们使用了假设 d 和 T 至少为 3 的条件。由于我们假设 C 是破碎的，因此前一个值至少为 2^m，从而得到

$$2^m \leq m^{(d+1)T}.$$

因此，

$$m \leq \log(m)\frac{(d+1)T}{\log(2)}\ .$$

引理A.1在第A章中告诉我们，上述条件成立的一个必要条件是

$$m \leq 2\frac{(d+1)T}{\log(2)}\log\frac{(d+1)T}{\log(2)} \leq (d+1)T(3\log((d+1)T) + 2),$$

这总结了我们的证明。 □

在练习4中，我们表明对于某些基类，B，也成立VCdim($L(B, T)$) $\geq \Omega$(VCdim(B) T)。

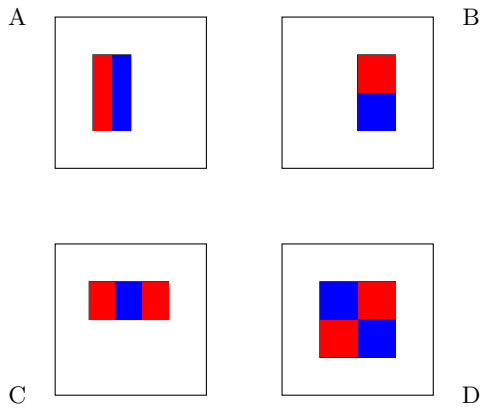


Figure 10.1 四种函数类型g，用于人脸识别的基本假设。类型A或B的g值是两个矩形区域内像素总和的差。这些区域具有相同的大小和形状，并且水平或垂直相邻。对于类型C，g的值是两个外部矩形内的总和减去中心矩形内的总和。对于类型D，我们计算矩形对角线对的差。

10.4 AdaBoost for Face Recognition

我们现在转向一个由Viola和Jones提出的基假设，用于人脸识别任务。在这个任务中，实例空间是图像，表示为像素灰度值的矩阵。具体来说，让我们考虑大小为24 $\times$ 24像素的图像，因此我们的实例空间是大小为24 $\times$ 24的实值矩阵集合。目标是学习一个分类器，$h : \mathcal{X} \to \{\pm 1\}$，给定一个图像作为输入，应该输出该图像是否为人脸。

每个基类中的假设都呈 $h(x) = f(g(x))$ 形式，其中 f 是一个决策树假设，$g : \mathbb{R}^{24,24} \to \mathbb{R}$ 是将图像映射到标量的函数。每个函数 g 都由参数化

- 一个与轴对齐的矩形 {v*}。由于每个图像的大小为 24 $\times$ 24，最多有 24^4 个与轴对齐的矩形。
- 一种类型，$t \in \{A, B, C, D\}$。每种类型对应一个掩码，如图10.1所示。

为了计算 g，我们将掩码 t 调整以适应矩形 R，然后计算位于红色矩形内的像素（即它们的灰度值之和）的总和，并将其从蓝色矩形中的像素总和减去。

自此类函数 g 的数量最多为 $24^4 \cdot 4$，我们可以通过首先计算 g 在每张图像上的所有可能输出，然后应用前一小节中描述的决策树弱学习器来实现基假设类的弱学习器。第一步可以非常

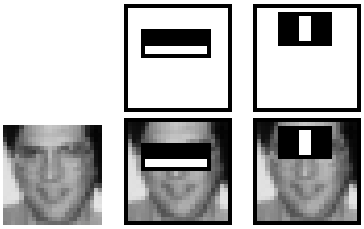


Figure 10.2 AdaBoost选出的第一和第二特征，由Viola和Jones实现。这两个特征显示在顶部行，然后在底部行的典型训练人脸上进行叠加。第一个特征测量眼睛区域与上脸颊区域之间的强度差异。该特征利用了眼睛区域通常比脸颊区域暗的观察结果。第二个特征比较眼睛区域的强度与鼻梁上的强度。

高效地通过一个预处理步骤，其中我们计算训练集中每个图像的积分图。有关详细信息，请参阅练习5。

在图10.2中，我们展示了使用Viola和Jones提出的基特征运行AdaBoost时选出的前两个特征。

10.5 Summary

提升是一种增强弱学习器准确性的方法。在本章中，我们描述了AdaBoost算法。我们已证明，经过T次AdaBoost迭代后，它从类别$L(B,T)$返回一个假设，该假设是通过在T个基类B的假设上组合线性分类器获得的。我们展示了参数T如何控制逼近误差和估计误差之间的权衡。在下一章中，我们将研究如何根据数据调整如T之类的参数。

10.6 Bibliographic Remarks

如前所述，提升算法起源于关于是否可以将高效的弱学习器“提升”为高效的强学习器的理论问题（Kearns & Valiant 1988），并由Schapire（1990）解决。AdaBoost算法由Freund和Schapire（1995）提出。

提升可以从许多角度来理解。在纯粹理论背景下，AdaBoost可以解释为一个负结果：如果一个假设类的强学习在计算上很困难，那么这个类的弱学习也是如此。这个负结果可以用来证明基于某些其他类 B 的 PAC 学习困难，只要

$\mathcal{H}$ 弱学习使用 B。例如，Klivans 和 Sherstov (2006) 已经表明，半空间交集类的 PAC 学习是困难的（即使在可实现的情形下）。这个困难结果可以用来证明单个半空间的不可知 PAC 学习也是计算上困难的（Shalev-Shwartz，Shamir 和 Sridharan 2010）。想法是证明单个半空间的不可知 PAC 学习者可以产生半空间交集类的弱学习器，由于这样的弱学习器可以被提升，我们将获得半空间交集类的强学习器。

AdaBoost还表明了弱学习者的存在与使用基假设预测的线性分类器对数据的可分性之间的等价性。此结果与冯·诺伊曼的最小-最大定理（冯·诺伊曼1928）密切相关，这是博弈论中的一个基本结果。

AdaBoost也与边缘的概念相关，我们将在第15章中稍后研究。它也可以被视为一种前向贪婪选择算法，这一主题将在第25章中介绍。Schapire & Freund（2012）最近的一本书从所有角度涵盖了提升，并提供了轻松访问该领域产生的丰富研究成果的途径。

10.7 Exercises

1. **Boosting the Confidence:** 设 A 是一个保证以下性质的算法：存在某些常数 $\delta_0 \in (0,1)$ 和一个函数 $m_{\mathcal{H}} : (0,1) \to \mathbb{N}$，使得对于每一个 $\epsilon \in (0,1)$，如果 $m \geq m_{\mathcal{H}}(\epsilon)$，那么对于每一个分布 $\mathcal{D}$，至少以概率 $1-\delta_0$，$L_{\mathcal{D}}(A(S)) \leq \min_{h\in\mathcal{H}} L_{\mathcal{D}}(h) + \epsilon$。

   建议一个依赖于 A 并在常规无偏PAC学习模型中学习 $\mathcal{H}$ 的过程，其样本复杂度为

$$m_{\mathcal{H}}(\epsilon,\delta) \leq k\, m_{\mathcal{H}}(\epsilon) + \left\lceil \frac{2\log(4k/\delta)}{\epsilon^2} \right\rceil,$$

   哪里

$$k = \lceil \log(\delta)/\log(\delta_0) \rceil.$$

   Hint: 数据分为 $k+$ 个 1 块，其中前 k 块的大小为 $m_{\mathcal{H}}(\epsilon)$ 个示例。使用 A 训练前 k 块。论证对于所有这些块，我们都有 $L_{\mathcal{D}}(A(S)) > \min_{h\in\mathcal{H}} L_{\mathcal{D}}(h)+\epsilon$ 的概率至多为 $\delta_0^k \leq \delta/2$。最后，使用最后一个块从由 k 块（通过依赖引理 4.6）生成的 k 个假设中选择。

2. 证明方程（10.5）中给出的函数 h 等于根据与 h 相同的阈值定义的分段常数函数。

3. 我们非正式地论证了AdaBoost算法使用加权机制来“强制”弱学习者在下一次迭代中关注有问题的示例。在这个问题中，我们将为这个论点找到一些严格的证明。

证明 h_t 关于分布 $\mathbf{D}^{(t+1)}$ 的误差正好是 $1/2$。也就是说，证明对于每一个 $t \in [T]$

$$\sum_{i=1}^{m} D_i^{(t+1)} \mathbb{1}_{[y_i \neq h_t(\mathbf{x}_i)]} = 1/2.$$

4. 在这个练习中，我们讨论形式为 $L(B,T)$ 的类的 VC 维度。我们证明了 $O(dT\log(dT))$ 的上界，其中 $d = \text{VCdim}(B)$。在这里，我们希望证明一个几乎匹配的下界。然而，这并不适用于所有类 B。

1. 注意对于每个类别 B 和每个数字 $T \geq 1$，$\text{VCdim}(B) \leq \text{VCdim}(L(B,T))$。找到一个类别 B，使得对于每个 $T \geq 1$，$\text{VCdim}(B) = \text{VCdim}(L(B,T))$。*Hint:* 将 $\mathcal{X}$ 视为一个有限集。2. 令 B_d 为 $\mathbb{R}^d$ 上的决策树类别。证明 $\log(d) \leq \text{VCdim}(B_d) \leq 5 + 2\log(d)$。*Hints:*

- 对于上界，依赖于练习11。
- 对于下界，假设 $d = 2^k$。令 A 为一个 $k \times d$ 矩阵，其列是 $\{\pm 1\}^k$ 中的所有 d 二进制向量。A 的行构成 R^d 中的一个 k 向量集。证明这个集合可以通过 $\mathbb{R}^d$ 上的决策树进行划分。3. 令 $T \geq 1$ 为任意整数。证明 $\text{VCdim}(L(B_d,T)) \geq 0.5\, T\log(d)$。*Hint:* 通过从上一个问题中的矩阵 A 中取行，以及从矩阵 $2A, 3A, 4A, \ldots, \frac{T}{2}A$ 中取行来构造一个 $\frac{T}{2}k$ 实例集。证明所得到的集合可以通过 $L(B_d,T)$ 进行划分。

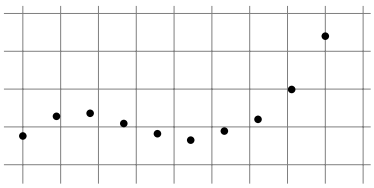
5. **Efficiently Calculating the Viola and Jones Features Using an Integral Image:** 设 A 为一个表示图像的 24×24 矩阵。A 的积分图像，记为 $I(A)$，是一个矩阵 B，使得 $B_{i,j} = \sum_{i' \leq i, j' \leq j} A_{i,j}$。

- 证明$I(A)$可以从A中计算出来，其时间复杂度与A的大小成线性关系。
- 展示如何以恒定的时间计算每个Viola和Jones特征从 $I(A)$（即，运行时间不依赖于定义特征的矩形大小）。

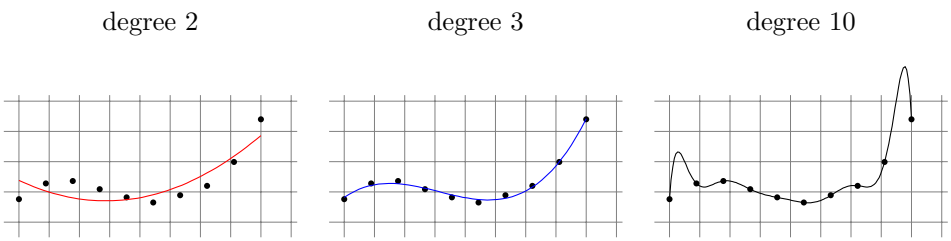
11 Model Selection and Validation

在上一章中，我们描述了AdaBoost算法，并展示了AdaBoost的参数T如何控制偏差-复杂度权衡。但是，我们如何在实践中设置T？更普遍地，当我们面对一些实际问题，我们通常可以想到几个可能产生良好解决方案的算法，每个算法可能有几个参数。我们如何为特定问题选择最佳算法？以及我们如何设置算法的参数？这项任务通常被称为*model selection*。

为了说明模型选择任务，考虑学习一维回归函数的问题，$h : \mathbb{R} \to \mathbb{R}$。假设我们获得了一个如图所示的训练集。



我们可以考虑将多项式拟合到数据中，如第9章所述。然而，我们可能不确定哪个次数 d 会给我们数据集带来最佳结果：较小的次数可能无法很好地拟合数据（即，它将有较大的近似误差），而较高的次数可能导致过拟合（即，它将有较大的估计误差）。在以下内容中，我们展示了拟合次数为2、3和10的多项式结果。很容易看出，随着我们增加次数，经验风险会降低。然而，观察图表，我们的直觉告诉我们，将次数设置为3可能比设置为10更好。因此，仅凭经验风险不足以进行模型选择。



在这一章中，我们将介绍两种模型选择方法。第一种方法基于我们在第7.2章中描述和分析的结构风险最小化（SRM）范式。当学习算法依赖于控制偏差-复杂度权衡的参数时（例如，在先前的例子中拟合多项式的度或AdaBoost中的参数T），SRM特别有用。第二种方法依赖于$validation$的概念。基本思想是将训练集分为两个集合。一个用于训练每个候选模型，另一个用于决定哪个模型能产生最佳结果。

在模型选择任务中，我们试图在近似误差和估计误差之间找到合适的平衡。更普遍地，如果我们的学习算法未能找到一个风险小的预测器，那么了解我们是过度拟合还是欠拟合就很重要。在第11.3节中，我们讨论了如何实现这一点。

11.1 Model Selection Using SRM

SRM 模式已在第 7.2 节中描述和分析。在此，我们展示了如何使用 SRM 来调整偏差和复杂度之间的权衡，而不需要在事先决定一个特定的假设类。考虑一个可数的假设类序列 {v*}。例如，在前面提到的多项式回归问题中，我们可以将 $\mathcal{H}_d$ 取为度数不超过 d 的多项式集合。另一个例子是将 $\mathcal{H}_d$ 取为前一章中描述的 AdaBoost 所使用的类 $L(B, d)$。

我们假设对于每个 d，类 $\mathcal{H}_d$ 具有形式为的样本复杂度函数的均匀收敛性质（参见第4章中的定义4.3）。

$$m^{\mathrm{UC}}_{\mathcal{H}_d}(\epsilon, \delta) \leq \frac{g(d)\log(1/\delta)}{\epsilon^2}, \tag{11.1}$$

在 $g : \mathbb{N} \to \mathbb{R}$ 是某个单调递增函数的情况下。例如，在二元分类问题的情况下，我们可以将 $g(d)$ 取为类 $\mathcal{H}_d$ 的 VC 维度乘以一个通用常数（出现在学习基本定理中的那个；参见定理 6.8）。对于 AdaBoost 使用的类 $L(B, d)$，函数 g 将简单地随着 d 增长。

回忆一下，SRM规则遵循“边界最小化”方法，在我们的情况下，边界如下：至少以$1 - \delta$的概率，对于每个 $d \in \mathbb{N}$ 和 $h \in \mathcal{H}_d$,

$$L_{\mathcal{D}}(h) \leq L_S(h) + \sqrt{\frac{g(d)(\log(1/\delta) + 2\log(d) + \log(\pi^2/6))}{m}}. \tag{11.2}$$

这个界限直接来自定理7.4，表明对于每个d和每个$h \in \mathcal{H}_d$，真实风险被两个项所限制——经验风险，$L_S(h)$,

并且一个依赖于 d 的复杂度项。SRM 规则将搜索 d 和 $h \in \mathcal{H}_d$ 以最小化方程 (11.2) 的右侧。

回到之前提到的多项式回归的例子，尽管10次多项式的经验风险小于3次多项式，我们仍然更喜欢3次多项式，因为它的复杂性（由函数$g(d)$的值反映）要小得多。

虽然SRM方法在某些情况下可能很有用，但在许多实际情况下，方程（11.2）给出的上界是悲观的。在下一节中，我们提出了一种更实用的方法。

11.2 Validation

我们通常会希望得到学习算法输出预测器真实风险的更好估计。到目前为止，我们已经推导出假设类估计误差的界限，这告诉我们，对于类中的 all 个假设，真实风险与经验风险相差不远。然而，这些界限可能过于宽松和悲观，因为它们适用于所有假设和所有可能的数据分布。可以通过使用一些训练数据作为验证集来获得对真实风险的更准确估计，在这个验证集上可以评估算法输出预测器的成功。这个过程被称为 $validation$。

当然，对真实风险的更好估计对于模型选择是有用的，正如我们将在第11.2.2节中描述的那样。

11.2.1 Hold Out Set

最简单估计预测器 h 真实误差的方法是通过采样一组额外的示例，这些示例独立于训练集，并使用此验证集上的经验误差作为我们的估计值。形式上，设 $V = (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_{m_v}, y_{m_v})$ 是一组根据 $\mathcal{D}$ (独立于训练集 m 中的 m_v 个新示例采样的示例。使用 Hoeffding 不等式（引理 4.5），我们得到以下:

THEOREM 11.1 *Let h be some predictor and assume that the loss function is in $[0, 1]$. Then, for every $\delta \in (0, 1)$, with probability of at least $1 - \delta$ over the choice of a validation set V of size m_v we have*

$$|L_V(h) - L_{\mathcal{D}}(h)| \leq \sqrt{\frac{\log(2/\delta)}{2\, m_v}}.$$

定理11.1中的界不依赖于构造h所使用的算法或训练集，并且比我们迄今为止看到的通常界更紧。这个界之所以紧，是因为它是基于一个独立于h生成方式的全新验证集的估计。为了说明这一点，假设h是通过应用ERM预测器获得的

关于VC维为d的假设类，在包含m个样本的训练集上。然后，根据学习的基本定理（定理6.8），我们得到如下界

$$L_{\mathcal{D}}(h) \leq L_S(h) + \sqrt{C\,\frac{d + \log(1/\delta)}{m}},$$

在定理6.8中出现的常数C，相比之下，从定理11.1中我们得到以下界限

$$L_{\mathcal{D}}(h) \leq L_V(h) + \sqrt{\frac{\log(2/\delta)}{2m_v}}.$$

因此，将 m_v 视为 m 的阶，我们得到一个更准确的估计，其准确度因子取决于VC维。另一方面，我们使用这种估计所付出的代价是，它需要在用于训练学习者的样本之外再增加一个样本。

从训练集和独立验证集进行采样相当于将我们的随机示例集随机划分为两部分，其中一部分用于训练，另一部分用于验证。因此，验证集通常被称为 *hold out*集。

11.2.2 Validation for Model Selection

验证可以自然地用于模型选择，如下所示。我们首先在给定的训练集上训练不同的算法（或具有不同参数的相同算法）。令$\mathcal{H} = \{h_1, \ldots, h_r\}$为不同算法的所有输出预测器的集合。例如，在训练多项式回归的情况下，我们会有每个h_r是度数为r的多项式回归的输出。现在，为了从$\mathcal{H}$中选择单个预测器，我们采样一个新的验证集，并选择在验证集上最小化误差的预测器。换句话说，我们在验证集上应用$\text{ERM}_{\mathcal{H}}$。

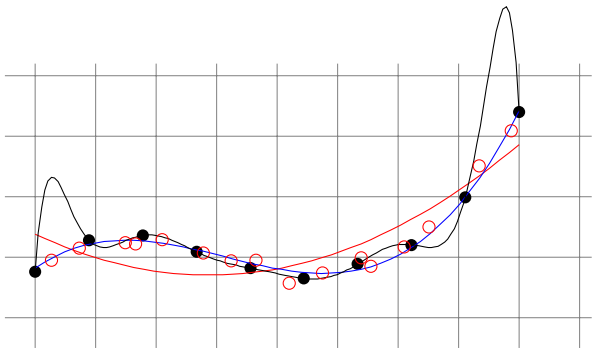
此过程与学习有限假设类非常相似。唯一的区别是 $\mathcal{H}$ 并非在事先固定，而是依赖于训练集。然而，由于验证集与训练集独立，我们得到它也独立于 $\mathcal{H}$，因此我们用于推导有限假设类界限的技术在这里同样适用。特别是，结合定理11.1与并集界限，我们得到：

THEOREM 11.2 *Let $\mathcal{H} = \{h_1, \ldots, h_r\}$ be an arbitrary set of predictors and assume that the loss function is in $[0, 1]$. Assume that a validation set V of size m_v is sampled independent of $\mathcal{H}$. Then, with probability of at least $1 - \delta$ over the choice of V we have*

$$\forall h \in \mathcal{H},\ |L_{\mathcal{D}}(h) - L_V(h)| \ \leq\ \sqrt{\frac{\log(2|\mathcal{H}|/\delta)}{2\,m_v}}.$$

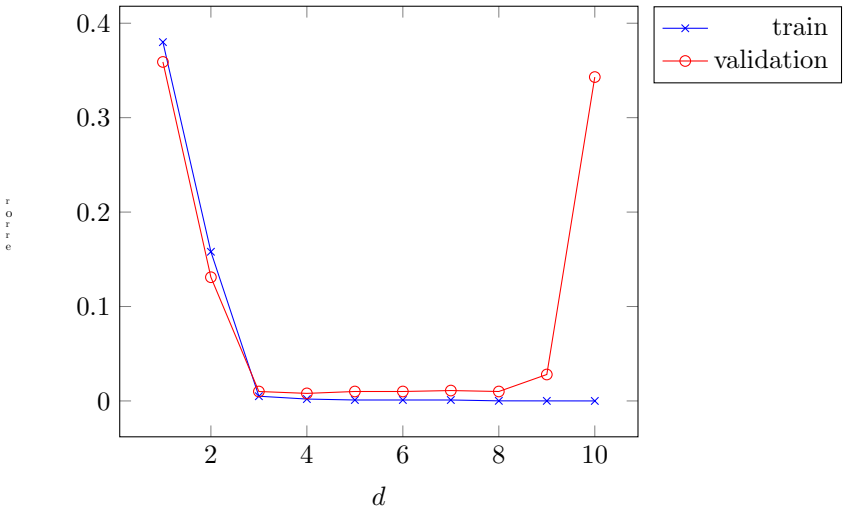
这个定理告诉我们，只要 $\mathcal{H}$ 不太大，验证集上的误差就近似于真实误差。然而，如果我们尝试太多的方法（导致 $|\mathcal{H}|$ 相对于验证集的大小过大），那么我们就有过拟合的风险。

为了说明验证在模型选择中的有用性，请再次考虑本章开头描述的拟合一维多项式的例子。在以下内容中，我们展示了相同的训练集，包括2次、3次和10次的ERM多项式，但这次我们还展示了额外的验证集（用红色、空心圆圈标记）。10次多项式具有最小的训练误差，而3次多项式具有最小的验证误差，因此它将被选为最佳模型。



11.2.3 The Model-Selection Curve

模型选择曲线显示了考虑的模型复杂度下的训练误差和验证误差。例如，对于之前提到的多项式拟合问题，曲线将看起来像：



随着我们增加多项式度（在我们的情况下，这是模型的复杂度），训练错误单调递减。另一方面，验证错误首先下降，然后开始上升，这表明我们开始遭受过拟合。

绘制此类曲线可以帮助我们了解我们是否正在搜索参数空间的正确区域。通常，可能存在多个需要调整的参数，每个参数可能取的值可能相当多。例如，在第13章中，我们描述了*regularization*的概念，其中学习算法的参数是一个实数。在这种情况下，我们从一个关于参数（s）的值的粗略网格开始，并绘制相应的模型选择曲线。基于曲线，我们将放大到正确的区域，并使用更细的网格进行搜索。验证我们是否处于相关区域非常重要。例如，在描述的多项式拟合问题中，如果我们从值集$\{1, 10, 20\}$开始搜索度数，并且不根据结果曲线使用更细的网格，我们将得到一个相当差的模型。

11.2.4 k-Fold Cross Validation

验证过程到目前为止所描述的假设数据丰富，并且我们有能力采样一个新的验证集。但在某些应用中，数据稀缺，我们不希望“浪费”数据在验证上。k-折交叉验证技术旨在在不浪费太多数据的情况下给出对真实错误的准确估计。

在 k-倍交叉验证中，原始训练集被划分为 k 个大小为 m/k (的子集（折）。为了简单起见，假设 m/k 是一个整数)。对于每个折，算法在其余折的并集上训练，然后使用该折来估计其输出的错误。最后，所有这些错误的平均值是

估计真实误差。当 m 是示例数量时，特殊情形 $k = m$ 被称为 *leave-one-out* (LOO)。

k-折叠交叉验证常用于模型选择（或参数调整），一旦选择了最佳参数，算法将使用此参数在整个训练集上重新训练。以下给出了用于模型选择的k-折叠交叉验证的伪代码。该过程接收一个训练集、S，一组可能的参数值、Θ，一个整数、k，表示折叠数，以及一个学习算法、A，该算法接收一个训练集以及一个参数$\theta \in \Theta$作为输入。它输出最佳参数以及在此参数下在整个训练集上训练的假设。

k-Fold Cross Validation for Model Selection

input:
  training set $S = (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$
  set of parameter values Θ
  learning algorithm A
  integer k
partition S into $S_1, S_2, \ldots, S_k$
foreach $\theta \in \Theta$
  for $i = 1 \ldots k$
    $h_{i,\theta} = A(S \setminus S_i; \theta)$
  $\text{error}(\theta) = \frac{1}{k} \sum_{i=1}^{k} L_{S_i}(h_{i,\theta})$
output
  $\theta^\star = \text{argmin}_\theta \left[\text{error}(\theta)\right]$
  $h_{\theta^\star} = A(S; \theta^\star)$

交叉验证方法在实践中通常效果很好。然而，有时可能会失败，正如练习1中给出的艺术示例所示。严格理解交叉验证的确切行为仍然是一个未解决的问题。罗杰斯和瓦格纳（罗杰斯 & 瓦格纳 1978）已经表明，对于 k 本地规则（例如，k 最近邻；见第19章）交叉验证过程给出了对真实错误的良好估计。其他论文表明，交叉验证适用于稳定的算法（我们将在第13章研究稳定性和其与学习性的关系）。

11.2.5 Train-Validation-Test Split

在大多数实际应用中，我们将可用的示例分为三个集合。第一个集合用于训练我们的算法，第二个集合用作模型选择的验证集。在选定了最佳模型后，我们在第三个集合上测试输出预测器的性能，这个集合通常被称为“测试集”。得到的数字被用作学习预测器真实错误的估计值。

11.3 What to Do If Learning Fails

考虑以下场景：你被分配了一个学习任务，并选择了假设类、学习算法和参数。你使用验证集调整参数，并在测试集上测试了学习到的预测器。遗憾的是，测试结果并不令人满意。那么，出了什么问题，接下来你应该做什么？

有许多元素可以被“固定”。主要方法如下列出：

- 获取更大的样本
- 通过以下方式更改假设类别：– 扩大它 – 减小它 – 完全更改它 – 更改您考虑的参数

- 更改数据的特征表示
- 更改应用学习规则所使用的优化算法

为了找到最好的补救措施，首先理解不良性能的原因是至关重要的。回想一下，在第5章中我们将学习预测器的真实误差分解为逼近误差和估计误差。逼近误差被定义为对于某些 $h^\star \in \text{argmin}_{h\in\mathcal{H}} L_{\mathcal{D}}(h)$，$L_{\mathcal{D}}(h^\star)$，而估计误差被定义为 $L_{\mathcal{D}}(h_S) - L_{\mathcal{D}}(h^\star)$，其中 h_S 是学习到的预测器（它基于训练集 S）。

类的近似误差不依赖于样本大小或所使用的算法。它只依赖于分布 $\mathcal{D}$ 和假设类 $\mathcal{H}$。因此，如果近似误差很大，增加训练集大小并不能帮助我们，同样减少假设类也没有意义。在这种情况下，扩大假设类或完全改变它（如果我们有一些以不同假设类形式存在的替代先验知识）可能会有所帮助。我们还可以考虑应用相同的假设类，但针对数据的不同特征表示（参见第25章）。

类别估计误差确实依赖于样本大小。因此，如果我们有一个大的估计误差，我们可以努力获取更多的训练示例。我们还可以考虑减少假设类别。然而，在那个情况下扩大假设类别是没有意义的。

Error Decomposition Using Validation
我们注意到，了解我们的问题是由于近似误差还是估计误差，对于找到最佳补救措施非常有用。在前一节中，我们看到了如何使用验证集上的经验风险来估计 $L_{\mathcal{D}}(h_S)$。然而，估计类别的近似误差更为困难。

相反，我们给出了一种不同的错误分解，这种分解可以从训练集和验证集中估计出来。

$$L_{\mathcal{D}}(h_S) = (L_{\mathcal{D}}(h_S) - L_V(h_S)) + (L_V(h_S) - L_S(h_S)) + L_S(h_S).$$

第一个项（$(L_{\mathcal{D}}(h_S) - L_V(h_S))$）可以使用定理11.1进行相当紧密的界定。直观上，当第二个项（$(L_V(h_S) - L_S(h_S))$）较大时，我们说我们的算法遭受了“过拟合”问题，而当经验风险项（$L_S(h_S)$）较大时，我们说我们的算法遭受了“欠拟合”问题。请注意，这两个项并不一定是估计和逼近误差的良好估计。为了说明这一点，考虑$\mathcal{H}$是一个VC维度d的类，$\mathcal{D}$是一个使得$\mathcal{H}$相对于$\mathcal{D}$的逼近误差为1/4的分布。只要我们的训练集大小小于d，我们就会对每个ERM假设有$L_S(h_S) = 0$。因此，训练风险$L_S(h_S)$和逼近误差$L_{\mathcal{D}}(h^\star)$可以显著不同。尽管如此，正如我们稍后所展示的，$L_S(h_S)$和$(L_V(h_S) - L_S(h_S))$的值仍然为我们提供了有用的信息。

首先考虑 $L_S(h_S)$ 较大的情况。我们可以写出

$$L_S(h_S) = (L_S(h_S) - L_S(h^\star)) + (L_S(h^\star) - L_{\mathcal{D}}(h^\star)) + L_{\mathcal{D}}(h^\star).$$

当 h_S 是 $\text{ERM}_{\mathcal{H}}$ 假设时，我们有 $L_S(h_S) - L_S(h^\star) \leq 0$。此外，由于 $h^\star$ 不依赖于 S，因此 $(L_S(h^\star) - L_{\mathcal{D}}(h^\star))$ 可以被相当紧密地界定（如定理11.1中所述）。最后一项是近似误差。因此，如果 $L_S(h_S)$ 很大，那么近似误差也会很大，我们应该相应地调整我们算法的补救措施（如前所述）。

Remark 11.1 可能是我们这一类的近似误差很小，但 $L_S(h_S)$ 的值很大。例如，也许我们在 ERM 实现中有一个错误，算法返回了一个不是 ERM 的假设 h_S。也可能找到 ERM 假设是计算上困难的，我们的算法应用了一些启发式方法来寻找近似的 ERM。在某些情况下，很难知道 h_S 相对于 ERM 假设有多好。但是，有时至少可以知道是否存在更好的假设。例如，在下一章中，我们将研究凸学习问题，其中存在可以检查以验证我们的优化算法是否收敛到 ERM 解的优化条件。在其他情况下，解可能取决于算法初始化中的随机性，因此我们可以尝试不同的随机选择的初始点，看看是否会出现更好的解。

接下来考虑$L_S(h_S)$较小的情况。正如我们之前所论证的，这并不一定意味着近似误差小。实际上，考虑两种情况，在这两种情况下，我们都在尝试使用ERM学习规则学习一个VC维数为d的假设类。在第一种情况下，我们有一个包含$m < d$个示例的训练集，该类别的近似误差较高。在第二种情况下，我们有一个包含$m > 2d$个示例的训练集和

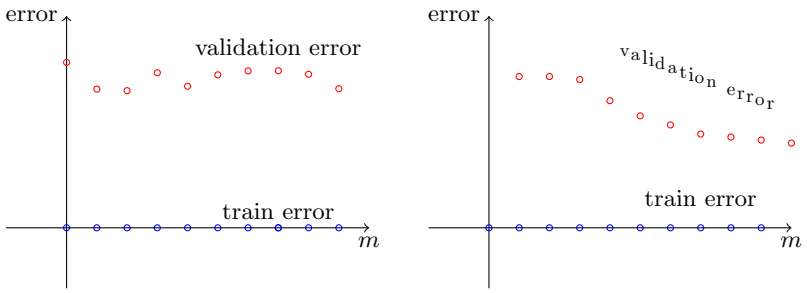


Figure 11.1 学习曲线示例。左侧：此学习曲线对应于示例数量始终小于类别的VC维度的场景。右侧：此学习曲线对应于近似误差为零且示例数量大于类别VC维度的场景。

类别的近似误差为零。在两种情况下 $L_S(h_S) = 0$。我们如何区分这两种情况?

Learning Curves

一种区分两种情况的可能方法是绘制 *learning curves*。为了生成学习曲线，我们在不断增加大小的数据前缀上训练算法。例如，我们首先可以在前10%的示例上训练算法，然后是20%，依此类推。对于每个前缀，我们计算训练错误（算法正在训练的前缀上的错误）和验证错误（在预定义的验证集上的错误）。这样的学习曲线可以帮助我们区分上述两种情况。在第一种情况下，我们预计验证错误将大约为 $1/2$，因为实际上我们没有学到任何东西。在第二种情况下，验证错误将开始为常数，但随后应该开始下降（一旦训练集大小大于VC维度，它必须开始下降）。图11.1给出了两种情况的示意图。

一般来说，只要近似误差大于零，我们预计训练误差会随着样本大小的增加而增长，因为更多的数据点使得为所有这些点提供解释变得更加困难。另一方面，验证误差往往随着样本大小的增加而降低。如果VC维是有限的，当样本大小趋于无穷大时，验证误差和训练误差会收敛到近似误差。因此，通过外推训练和验证曲线，我们可以尝试猜测近似误差的值，或者至少得到近似误差所在区间的粗略估计。

回到我们算法失败的最佳补救措施问题，如果我们观察到$L_S(h_S)$很小而验证误差很大，那么在任何情况下我们都知道我们的训练集大小不足以学习类别$\mathcal{H}$。然后我们可以绘制一个学习曲线。如果我们看到

验证错误开始下降时，最佳解决方案是增加示例数量（如果我们负担得起扩大数据）。另一个合理的解决方案是降低假设类的复杂性。另一方面，如果我们看到验证错误保持在1/2左右，那么我们没有证据表明$\mathcal{H}$的近似误差是好的。可能的情况是，增加训练集大小根本不能帮助我们。获取更多数据仍然可以帮助我们，因为在某个时候我们可以看到验证错误是否开始下降，或者训练错误是否开始增加。但是，如果更多数据很昂贵，可能首先尝试尝试降低假设类的复杂性会更好。

为了总结讨论，以下步骤应为一个 应用：

1. 如果学习涉及参数调整，绘制模型选择曲线以确保您适当地调整了参数（见第11.2.3节）。2. 如果训练误差过大，考虑扩大假设类、完全改变它或改变数据的特征表示。3. 如果训练误差很小，绘制学习曲线并尝试从中推断问题是否为估计误差或近似误差。4. 如果近似误差似乎足够小，尝试获取更多数据。如果这不可能，考虑降低假设类的复杂性。5. 如果近似误差似乎也很大，尝试完全改变假设类或数据的特征表示。

11.4 Summary

模型选择是根据数据本身选择适当学习任务的模型的任务。我们已经展示了如何使用SRM学习范式或使用更实用的验证方法来完成这项任务。如果我们的学习算法失败，应使用学习曲线对算法的错误进行分解，以便找到最佳的补救措施。

11.5 Exercises

1. **Failure of k-fold cross validation** 考虑一个情况，其中标签是随机选择的，根据 $\mathbb{P}[y = 1] = \mathbb{P}[y = 0] = 1/2$. 考虑一个学习算法，如果训练集中标签的奇偶性为 1，则输出常量预测器 $h(\mathbf{x}) = 1$，否则算法输出常量预测器 $h(\mathbf{x}) = 0$。证明在这种情况下，留一法估计值与真实误差之间的差总是 $1/2$。2. 设 $\mathcal{H}_1, \ldots, \mathcal{H}_k$ 为 k 假设类。假设你被给定了 m 独立同分布的训练样本，并且你想要学习类别 $\mathcal{H} = \cup_{i=1}^{k} \mathcal{H}_i$。考虑两种不同的方法：

- 学习 $\mathcal{H}$ 在 m 个示例中使用 ERM 规则

- 将 m 个示例分为大小为 $(1-\alpha)m$ 的训练集和大小为 αm 的验证集，对于某些 $\alpha \in (0,1)$。然后，使用验证方法进行模型选择。也就是说，首先使用关于 $\mathcal{H}_i$ 的 ERM 规则，在每个类别 $\mathcal{H}_i$ 上训练 $(1-\alpha)m$ 个训练示例，并让 $\hat{h}_1, \ldots, \hat{h}_k$ 成为结果假设。其次，在 αm 个验证示例上应用关于有限类别 $\{\hat{h}_1, \ldots, \hat{h}_k\}$ 的 ERM 规则。

描述在哪种情况下第一种方法比第二种方法更好，反之亦然。

12 Convex Learning Problems

在这一章中，我们介绍了 *convex learning problems*。凸学习包含了一类重要的学习问题，主要是因为我们大部分可以高效学习的内容都包含在其中。我们已经遇到了带有平方损失的线性回归和逻辑回归，它们是凸问题，并且确实可以高效地学习。我们还看到了非凸问题，例如带有 0-1 损失的半空间，这在不可实现的情况下已知计算上难以学习。

通常，凸学习问题是一个其假设类是凸集，并且对于每个示例其损失函数是凸函数的问题。我们以一些关于凸性的必要定义开始本章。除了凸性之外，我们还将定义Lipschitz性和光滑性，这些是损失函数的附加属性，有助于成功学习。接下来，我们转向定义凸学习问题，并证明进一步约束（如有界性和Lipschitz性或光滑性）的必要性。我们定义这些更受限制的学习问题家族，并声称凸-光滑/Lipschitz-有界问题是可学习的。这些论断将在下一章中得到证明，其中我们将介绍两种学习范式，它们可以成功学习所有凸-Lipschitz-有界或凸-光滑-有界的问题。

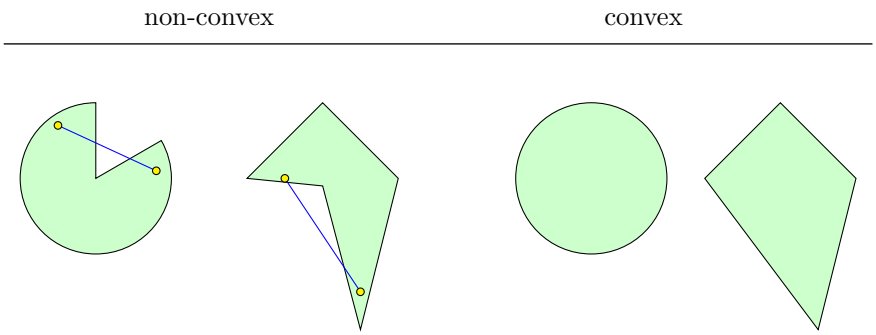
最后，在第12.3节中，我们展示了如何通过最小化凸的“代理”损失函数来处理一些非凸问题（而不是原始的非凸损失函数）。代理凸损失函数可以产生高效的解决方案，但可能会增加学习预测器的风险。

12.1 Convexity, Lipschitzness, and Smoothness

12.1.1 Convexity

DEFINITION 12.1（凸集）向量空间中的一个集合 C 是凸的，如果对于集合 C 中的任意两个向量 $\mathbf{u}, \mathbf{v}$，从 $\mathbf{u}$ 到 $\mathbf{v}$ 的线段包含在 C 中。也就是说，对于任意的 $\alpha \in [0, 1]$，我们有 $\alpha\mathbf{u} + (1 - \alpha)\mathbf{v} \in C$。

以下给出了$\mathbb{R}^2$中凸集和非凸集的例子。对于非凸集，我们描绘了集合中的两个点，使得这两个点之间的直线不包含在集合中。

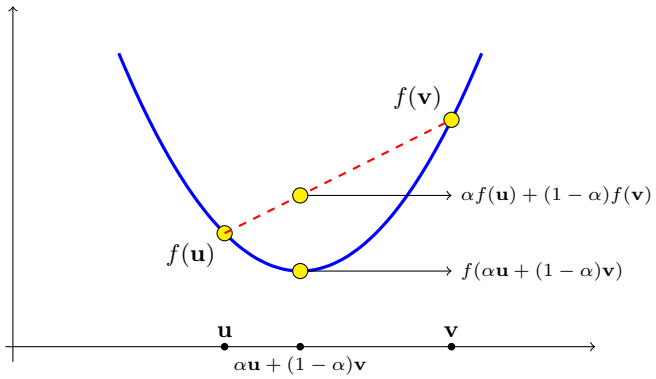


给定 $\alpha \in [0,1]$，点的组合 $\alpha\mathbf{u} + (1-\alpha)\mathbf{v}$ 被称为 $\mathbf{u}, \mathbf{v}$。

DEFINITION 12.2（凸函数）设C为一个凸集。若函数f: $C \to \mathbb{R}$对于任意的 $\mathbf{u}, \mathbf{v} \in C$和$\alpha \in [0,1]$，则$f$是凸函数。

$$f(\alpha\mathbf{u} + (1-\alpha)\mathbf{v}) \;\leq\; \alpha f(\mathbf{u}) + (1-\alpha) f(\mathbf{v}) \;.$$

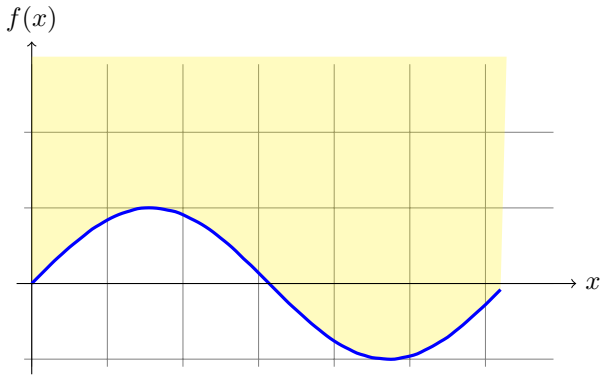
在文字上，如果对于任何 $\mathbf{u}, \mathbf{v}$，f 在 $\mathbf{u}$ 和 $\mathbf{v}$ 之间的图像位于连接 $f(\mathbf{u})$ 和 $f(\mathbf{v})$ 的线段下方，则 f 是凸的。以下展示了凸函数 $f : \mathbb{R} \to \mathbb{R}$ 的一个示意图。



函数 f 的 *epigraph* 是集合

$$\text{epigraph(f)} = \{(\mathbf{x}, \beta) : f(\mathbf{x}) \leq \beta\}. \tag{12.1}$$

一个函数 f 是凸的，当且仅当其凸包集是凸集。以下给出了一个非凸函数 f 的示例：$\mathbb{R} \to \mathbb{R}$，以及其凸包集。



凸函数的一个重要性质是，函数的每一个局部极小值也是全局极小值。形式上，设 {v*} 是以 {v*} 为中心、半径为 {v*} 的球。我们说 {v*} 是 {v*} 在 {v*} 处的局部极小值，如果存在某个 {v*} > 0，使得对于所有 {v*}，我们有 {v*}。由此可知，对于任何 {v*}（不一定在 {v*} 中），存在足够小的 {v*} > 0，使得 {v*}，因此

$$f(\mathbf{u}) \leq f(\mathbf{u} + \alpha(\mathbf{v} - \mathbf{u})) \ . \tag{12.2}$$

如果 f 是凸的，我们也有 {v*}

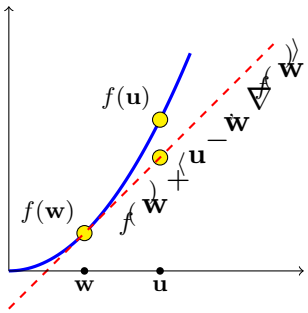
$$f(\mathbf{u} + \alpha(\mathbf{v} - \mathbf{u})) = f(\alpha\mathbf{v} + (1-\alpha)\mathbf{u}) \leq (1-\alpha)f(\mathbf{u}) + \alpha f(\mathbf{v}) \ . \tag{12.3}$$

这两个方程结合并重新排列项后，我们得出结论 $f(\mathbf{u}) \leq f(\mathbf{v})$。由于这对每个 $\mathbf{v}$ 都成立，因此 $f(\mathbf{u})$ 也是 f 的全局最小值。

凸函数的一个重要性质是，对于每个 $\mathbf{w}$，我们都可以在 $\mathbf{w}$ 处构造一个位于 f 之下的 f 的切线。如果 f 可微，则这条切线是线性函数 $l(\mathbf{u}) = f(\mathbf{w}) + \langle \nabla f(\mathbf{w}), \mathbf{u} - \mathbf{w} \rangle$，其中 $\nabla f(\mathbf{w})$ 是 f 在 $\mathbf{w}$ 处的梯度，即 f，$\nabla f(\mathbf{w}) =$，$\left(\frac{\partial f(\mathbf{w})}{\partial w_1}, \ldots, \frac{\partial f(\mathbf{w})}{\partial w_d} \right)$ 的偏导数向量。也就是说，对于凸可微函数，

$$\forall \mathbf{u}, \quad f(\mathbf{u}) \geq f(\mathbf{w}) + \langle \nabla f(\mathbf{w}), \mathbf{u} - \mathbf{w} \rangle. \tag{12.4}$$

第14章中，我们将将这个不等式推广到非可微函数。以下给出了方程（12.4）的示例。



如果 f 是一个可微的标量函数，有一个简单的方法来检查它是否是凸函数。

LEMMA 12.3 *Let $f:\mathbb{R}\to\mathbb{R}$ be a scalar twice differential function, and let f', f'' be its first and second derivatives, respectively. Then, the following are equivalent:*

1. *f is convex*
2. *f' is monotonically nondecreasing*
3. *f'' is nonnegative*

Example 12.1

- 标量函数 $f(x) = x^2$ 是凸函数。为了看到这一点，请注意 $f'(x) = 2x$ 和 $f''(x) = 2 > 0$。
- 标量函数 $f(x) = \log(1+\exp(x))$ 是凸函数。为了看到这一点，观察 $f'(x) = \frac{\exp(x)}{1+\exp(x)} = \frac{1}{\exp(-x)+1}$。这是一个单调递增函数，因为指数函数是单调递增函数。

以下声明表明，凸标量函数与线性函数的组合产生一个凸向量值函数。

CLAIM 12.4 *Assume that $f:\mathbb{R}^d\to\mathbb{R}$ can be written as $f(\mathbf{w}) = g(\langle\mathbf{w},\mathbf{x}\rangle + y)$, for some $\mathbf{x}\in\mathbb{R}^d$, $y\in\mathbb{R}$, and $g:\mathbb{R}\to\mathbb{R}$. Then, convexity of g implies the convexity of f.*

Proof 让 $\mathbf{w}_1, \mathbf{w}_2 \in \mathbb{R}^d$ 和 $\alpha \in [0,1]$。我们有

$$\begin{aligned} f(\alpha\mathbf{w}_1 + (1-\alpha)\mathbf{w}_2) &= g(\langle\alpha\mathbf{w}_1 + (1-\alpha)\mathbf{w}_2, \mathbf{x}\rangle + y) \\ &= g(\alpha\langle\mathbf{w}_1,\mathbf{x}\rangle + (1-\alpha)\langle\mathbf{w}_2,\mathbf{x}\rangle + y) \\ &= g(\alpha(\langle\mathbf{w}_1,\mathbf{x}\rangle + y) + (1-\alpha)(\langle\mathbf{w}_2,\mathbf{x}\rangle + y)) \\ &\le \alpha g(\langle\mathbf{w}_1,\mathbf{x}\rangle + y) + (1-\alpha)g(\langle\mathbf{w}_2,\mathbf{x}\rangle + y), \end{aligned}$$

最后的不等式由 g 的凸性得出。 □

Example 12.2

- 给定一些 $\mathbf{x} \in \mathbb{R}^d$ 和 $y \in \mathbb{R}$，令 $f: \mathbb{R}^d \to \mathbb{R}$ 被定义为 $f(\mathbf{w}) = (\langle \mathbf{w}, \mathbf{x} \rangle - y)^2$ 。然后，f 是函数 $g(a) = a^2$ 与线性函数的复合，因此 f 是一个凸函数。

- 给定一些 $\mathbf{x} \in \mathbb{R}^d$ 和 $y \in \{\pm 1\}$，令 $f: \mathbb{R}^d \to \mathbb{R}$ 被定义为 $f(\mathbf{w}) = \log(1 + \exp(-y\langle \mathbf{w}, \mathbf{x} \rangle))$。然后，$f$ 是函数 $g(a) = \log(1 + \exp(a))$ 与一个线性函数的复合，因此 f 是一个凸函数。

最后，以下引理表明凸函数的最大值是凸的，并且具有非负权重的凸函数的加权和也是凸的。

CLAIM 12.5 *For $i = 1, \ldots, r$, let $f_i: \mathbb{R}^d \to \mathbb{R}$ be a convex function. The following functions from $\mathbb{R}^d$ to $\mathbb{R}$ are also convex.*

- $g(x) = \max_{i \in [r]} f_i(x)$
- $g(x) = \sum_{i=1}^{r} w_i f_i(x)$, *where for all i, $w_i \geq 0$.*

Proof 第一条权利要求可由以下方式得出

$$\begin{aligned} g(\alpha u + (1-\alpha)v) &= \max_i f_i(\alpha u + (1-\alpha)v) \\ &\leq \max_i \left[\alpha f_i(u) + (1-\alpha) f_i(v)\right] \\ &\leq \alpha \max_i f_i(u) + (1-\alpha) \max_i f_i(v) \\ &= \alpha g(u) + (1-\alpha) g(v). \end{aligned}$$

对于第二个主张

$$\begin{aligned} g(\alpha u + (1-\alpha)v) &= \sum_i w_i f_i(\alpha u + (1-\alpha)v) \\ &\leq \sum_i w_i \left[\alpha f_i(u) + (1-\alpha) f_i(v)\right] \\ &= \alpha \sum_i w_i f_i(u) + (1-\alpha) \sum_i w_i f_i(v) \\ &= \alpha g(u) + (1-\alpha) g(v). \end{aligned}$$

□

Example 12.3 函数 $g(x) = |x|$ 是凸函数。为了看到这一点，请注意 $g(x) = \max\{x, -x\}$，并且函数 $f_1(x) = x$ 和 $f_2(x) = -x$ 都是凸函数。

12.1.2 Lipschitzness

以下关于Lipschitz性的定义是相对于$\mathbb{R}^d$上的欧几里得范数。然而，可以相对于任何范数定义Lipschitz性。

DEFINITION 12.6 (Lipschitzness) 设 $C \subset \mathbb{R}^d$。若函数 $f: \mathbb{R}^d \to \mathbb{R}^k$ 在 C 上是 ρ-Lipschitz，则对于每个 $\mathbf{w}_1, \mathbf{w}_2 \in C$，都有 $\|f(\mathbf{w}_1) - f(\mathbf{w}_2)\| \leq \rho \|\mathbf{w}_1 - \mathbf{w}_2\|$。

直观上，Lipschitz 函数不能变化得太快。注意，如果 $f:\mathbb{R}\to\mathbb{R}$ 可微，那么根据平均值定理，我们有

$$f(w_1)-f(w_2)=f'(u)(w_1-w_2)\ ,$$

在 u 是 w_1 和 w_2 之间的某一点。因此，如果 f 的导数在每处（绝对值）都受 ρ 的限制，那么该函数是 ρ-Lipschitz。

Example 12.4

- 函数 $f(x)=|x|$ 在 $\mathbb{R}$ 上是 1-Lipschitz。这遵循三角不等式：对于每个 x_1,x_2，

  $$|x_1|-|x_2|=|x_1-x_2+x_2|-|x_2|\le|x_1-x_2|+|x_2|-|x_2|=|x_1-x_2|.$$

  由于这对 x_1,x_2 和 x_2,x_1 都成立，我们得到 $||x_1|-|x_2||\le|x_1-x_2|$。

- 函数 $f(x)=\log(1+\exp(x))$ 在 $\mathbb{R}$ 上是 1-Lipschitz。为了看到这一点，观察如下：

  $$|f'(x)|=\left|\frac{\exp(x)}{1+\exp(x)}\right|=\left|\frac{1}{\exp(-x)+1}\right|\le 1.$$

- 函数 $f(x)=x^2$ 在 $\mathbb{R}$ 上对任何 ρ 都不是 ρ-Lipschitz。为了看到这一点，取 $x_1=0$ 和 $x_2=1+\rho$，然后

  $$f(x_2)-f(x_1)=(1+\rho)^2>\rho(1+\rho)=\rho|x_2-x_1|.$$

  然而，此函数在集合 $C=\{x$ 上是 ρ-Lipschitz：$|x|\le\rho/2\}$。确实，对于任何 $x_1,x_2\in C$，我们有

  $$|x_1^2-x_2^2|=|x_1+x_2|\ |x_1-x_2|\le 2(\rho/2)\,|x_1-x_2|=\rho|x_1-x_2|.$$

- 线性函数 $f:\mathbb{R}^d\to\mathbb{R}$ 由 $f(\mathbf{w})=\langle\mathbf{v},\mathbf{w}\rangle+b$ 定义，其中 $\mathbf{v}\in\mathbb{R}^d$ 是 $\|\mathbf{v}\|$-Lipschitz。实际上，使用柯西-施瓦茨不等式，

  $$|f(\mathbf{w}_1)-f(\mathbf{w}_2)|=|\langle\mathbf{v},\mathbf{w}_1-\mathbf{w}_2\rangle|\le\|\mathbf{v}\|\,\|\mathbf{w}_1-\mathbf{w}_2\|.$$

以下声明表明，Lipschitz函数的复合保持Lipschitz性质。

CLAIM 12.7 *Let $f(\mathbf{x})=g_1(g_2(\mathbf{x}))$, where g_1 is ρ_1-Lipschitz and g_2 is ρ_2-Lipschitz. Then, f is $(\rho_1\rho_2)$-Lipschitz. In particular, if g_2 is the linear function, $g_2(\mathbf{x})=\langle\mathbf{v},\mathbf{x}\rangle+b$, for some $\mathbf{v}\in\mathbb{R}^d, b\in\mathbb{R}$, then f is $(\rho_1\|\mathbf{v}\|)$-Lipschitz.*

Proof

$$\begin{aligned}|f(\mathbf{w}_1)-f(\mathbf{w}_2)|&=|g_1(g_2(\mathbf{w}_1))-g_1(g_2(\mathbf{w}_2))|\\&\le\rho_1\|g_2(\mathbf{w}_1)-g_2(\mathbf{w}_2)\|\\&\le\rho_1\,\rho_2\,\|\mathbf{w}_1-\mathbf{w}_2\|.\end{aligned}$$

□

12.1.3 Smoothness

定义光滑函数依赖于 *gradient* 的概念。回忆一下，可微函数 $f\colon \mathbb{R}^d \to \mathbb{R}$ 在 $\mathbf{w}$ 处的梯度，记为 $\nabla f(\mathbf{w})$，是 f 的偏导数向量，即 $\nabla f(\mathbf{w}) = \left(\frac{\partial f(\mathbf{w})}{\partial w_1}, \ldots, \frac{\partial f(\mathbf{w})}{\partial w_d}\right)$。

DEFINITION 12.8（平滑性）一个可微函数 {v*} : {v*} 是 {v*}-平滑的，如果其梯度是 {v*}-Lipschitz；即，对于所有 {v*}，我们有 {v*} {v*}。

可以证明平滑性意味着对于所有 $\mathbf{v}, \mathbf{w}$，我们有

$$f(\mathbf{v}) \leq f(\mathbf{w}) + \langle \nabla f(\mathbf{w}), \mathbf{v} - \mathbf{w} \rangle + \frac{\beta}{2}\|\mathbf{v} - \mathbf{w}\|^2 \ . \tag{12.5}$$

回忆起 f 的凸性意味着 $f(\mathbf{v}) \geq f(\mathbf{w}) + \langle \nabla f(\mathbf{w}), \mathbf{v} - \mathbf{w} \rangle$。因此，当一个函数既是凸函数又是光滑函数时，我们对该函数与其一阶近似之间的差异有上界和下界。

设置方程（12.5）右侧的 $\mathbf{v} = \mathbf{w} - \frac{1}{\beta}\nabla f(\mathbf{w})$ 并重新排列项，我们得到

$$\frac{1}{2\beta}\|\nabla f(\mathbf{w})\|^2 \leq f(\mathbf{w}) - f(\mathbf{v}).$$

如果我们进一步假设对于所有 $\mathbf{v}$，$f(\mathbf{v}) \geq 0$，我们得出以下结论：平滑性意味着以下：

$$\|\nabla f(\mathbf{w})\|^2 \leq 2\beta f(\mathbf{w}) \ . \tag{12.6}$$

一个满足此性质的函数也称为 *self-bounded* 函数。

Example 12.5

- 函数 $f(x) = x^2$ 是 2-平滑的。这直接从 $f'(x) = 2x$ 这一事实中得出。请注意，对于这个特定的函数，方程（12.5）和方程（12.6）成立且相等。
- 函数 $f(x) = \log(1 + \exp(x))$ 是 $(1/4)$-平滑的。确实，由于 $f'(x) = \frac{1}{1+\exp(-x)}$ 我们有

$$|f''(x)| = \frac{\exp(-x)}{(1+\exp(-x))^2} = \frac{1}{(1+\exp(-x))(1+\exp(x))} \leq 1/4.$$

因此，f' 是 $(1/4)$-Lipschitz。由于此函数非负，方程（12.6）也成立。

以下声明表明，在线性函数上对光滑标量函数的复合保持光滑性。

CLAIM 12.9 *Let $f(\mathbf{w}) = g(\langle \mathbf{w}, \mathbf{x} \rangle + b)$, where $g : \mathbb{R} \to \mathbb{R}$ is a β-smooth function, $\mathbf{x} \in \mathbb{R}^d$, and $b \in \mathbb{R}$. Then, f is $(\beta \|\mathbf{x}\|^2)$-smooth.*

Proof 通过链式法则，我们有 $\nabla f(\mathbf{w}) = g'(\langle \mathbf{w}, \mathbf{x} \rangle + b)\mathbf{x}$，其中 g' 是 g 的导数。利用 g 的光滑性和柯西-施瓦茨不等式，因此我们得到

$$\begin{aligned} f(\mathbf{v}) &= g(\langle \mathbf{v}, \mathbf{x} \rangle + b) \\ &\leq g(\langle \mathbf{w}, \mathbf{x} \rangle + b) + g'(\langle \mathbf{w}, \mathbf{x} \rangle + b)\langle \mathbf{v} - \mathbf{w}, \mathbf{x} \rangle + \frac{\beta}{2}(\langle \mathbf{v} - \mathbf{w}, \mathbf{x} \rangle)^2 \\ &\leq g(\langle \mathbf{w}, \mathbf{x} \rangle + b) + g'(\langle \mathbf{w}, \mathbf{x} \rangle + b)\langle \mathbf{v} - \mathbf{w}, \mathbf{x} \rangle + \frac{\beta}{2}\left(\|\mathbf{v} - \mathbf{w}\| \, \|\mathbf{x}\|\right)^2 \\ &= f(\mathbf{w}) + \langle \nabla f(\mathbf{w}), \mathbf{v} - \mathbf{w} \rangle + \frac{\beta \|\mathbf{x}\|^2}{2} \|\mathbf{v} - \mathbf{w}\|^2. \end{aligned}$$

□

Example 12.6

- 对于任意的 $\mathbf{x} \in \mathbb{R}^d$ 和 $y \in \mathbb{R}$，令 $f(\mathbf{w}) = (\langle \mathbf{w}, \mathbf{x} \rangle - y)^2$。然后，$f$ 是 $(2\|\mathbf{x}\|^2)$- 平滑的。
- 对于任意的 $\mathbf{x} \in \mathbb{R}^d$ 和 $y \in \{\pm 1\}$，令 $f(\mathbf{w}) = \log(1 + \exp(-y\langle \mathbf{w}, \mathbf{x} \rangle))$。然后，$f$ 是 $(\|\mathbf{x}\|^2/4)$-平滑的。

12.2 Convex Learning Problems

回忆起我们在第三章中关于学习的一般定义（定义3.4），我们有一个假设类$\mathcal{H}$，一组示例Z，以及一个损失函数ℓ：$\mathcal{H} \times Z \to \mathbb{R}_+$。到目前为止，在本书中，我们主要将$Z$视为实例空间和目标空间$Z = \mathcal{X} \times \mathcal{Y}$的乘积，$\mathcal{H}$是一组从$\mathcal{X}$到$\mathcal{Y}$的函数。然而，$\mathcal{H}$可以是任意集合。实际上，在本章中，我们考虑的假设类$\mathcal{H}$是欧几里得空间$\mathbb{R}^d$的子集。也就是说，每个假设都是某个实值向量。因此，我们将用$\mathbf{w}$表示$\mathcal{H}$中的假设。现在我们终于可以定义凸学习问题：

DEFINITION 12.10 (凸学习问题) 一个学习问题，$(\mathcal{H}, Z, \ell)$，被称为凸的，如果假设类 $\mathcal{H}$ 是一个凸集，并且对于所有 $z \in Z$，损失函数，$\ell(\cdot, z)$，是一个凸函数（其中，对于任何 z，$\ell(\cdot, z)$ 表示由 $f(\mathbf{w}) = \ell(\mathbf{w}, z)$ 定义的功能 $f : \mathcal{H} \to \mathbb{R}$）。

Example 12.7 (线性回归与平方损失) 回想一下，线性回归是用于建模某些“解释”变量与某些实值结果之间关系的一种工具（参见第9章）。域集$\mathcal{X}$是$\mathbb{R}^d$的子集，对于某些d，标签集$\mathcal{Y}$是实数集。我们希望学习一个线性函数h：$\mathbb{R}^d \to \mathbb{R}$，它能最好地近似我们的变量之间的关系。在第9章中，我们将假设类定义为同质线性函数的集合，$\mathcal{H} = \{\mathbf{x} \mapsto \langle \mathbf{w}, \mathbf{x} \rangle : \mathbf{w} \in \mathbb{R}^d\}$，并使用了平方损失函数$\ell(h, (\mathbf{x}, y)) = (h(\mathbf{x}) - y)^2$。然而，我们可以等效地将学习问题建模为一个凸学习问题，如下所示。

每个线性函数都由一个向量 $\mathbf{w} \in \mathbb{R}^d$ 参数化。因此，我们可以定义 $\mathcal{H}$ 为所有此类参数的集合，即 $\mathcal{H} = \mathbb{R}^d$。示例集合是 $Z = \mathcal{X} \times \mathcal{Y} = \mathbb{R}^d \times \mathbb{R} = \mathbb{R}^{d+1}$，损失函数是 $\ell(\mathbf{w}, (\mathbf{x}, y)) = (\langle \mathbf{w}, \mathbf{x} \rangle - y)^2$。显然，集合 $\mathcal{H}$ 是一个凸集。损失函数对其第一个参数也是凸的（参见示例12.2）。

LEMMA 12.11 *If ℓ is a convex loss function and the class $\mathcal{H}$ is convex, then the* ERM$_\mathcal{H}$ *problem, of minimizing the empirical loss over $\mathcal{H}$, is a convex optimization problem (that is, a problem of minimizing a convex function over a convex set).*

Proof 回忆一下，ERM$_\mathcal{H}$问题由以下定义：

$$\mathrm{ERM}_\mathcal{H}(S) = \operatorname*{argmin}_{\mathbf{w} \in \mathcal{H}} L_S(\mathbf{w}).$$

因此，对于样本 $S = z_1, \ldots, z_m$，对于每个 $\mathbf{w}$，$L_S(\mathbf{w}) = \frac{1}{m} \sum_{i=1}^m \ell(\mathbf{w}, z_i)$，12.5项表明 $L_S(\mathbf{w})$ 是一个凸函数。因此，ERM规则是在约束解应在凸集内的条件下，最小化凸函数的问题。

□

在温和条件下，这些问题可以使用通用优化算法有效地解决。特别是，在第14章中，我们将介绍一个用于最小化凸函数的非常简单的算法。

12.2.1 Learnability of Convex Learning Problems

我们已经论证，对于许多情况，对于凸学习问题实施ERM规则可以高效实现。但是凸性是问题可学习性的充分条件吗?

为了使问题更加具体：在VC理论中，我们看到了在{v*}维度的半空间是可学习的（可能效率不高）。我们还在第9章中通过使用“离散化技巧”论证，如果问题是{v*}个参数，它可以通过一个与{v*}相关的样本复杂度进行学习。也就是说，对于常数{v*}，问题应该是可学习的。那么，也许所有在{v*}上的凸学习问题都是可学习的?

示例12.8稍后显示，即使d较低，答案也是负的。并非所有在$\mathbb{R}^d$上的凸学习问题都是可学习的。这与VC理论没有矛盾，因为VC理论只处理二分类，而这里我们考虑的是一个广泛的问题家族。这也与“离散化技巧”没有矛盾，因为在那里我们假设损失函数是有界的，并且假设使用有限数量的位表示每个参数是足够的。正如我们稍后将要展示的，在许多实际场景中成立的某些附加限制条件下，凸问题是可学习的。

Example 12.8 (线性回归即使 $d = 1$) 令 $\mathcal{H} = \mathbb{R}$，损失为平方损失：$\ell(w, (x, y)) = (wx - y)^2$ (我们所指的是

同质情况）。设 A 为任何确定性算法.[1] 假设，通过反证法，A 是该问题的成功 PAC 学习者。也就是说，存在一个函数 $m(\cdot,\cdot)$，对于每个分布 $\mathcal{D}$ 和每个 ϵ,δ，如果 A 收到一个大小为 $m \geq m(\epsilon,\delta)$ 的训练集，它应该以至少 $1-\delta$ 的概率输出一个假设 $\hat{w} = A(S)$，使得 $L_{\mathcal{D}}(\hat{w}) - \min_w L_{\mathcal{D}}(w) \leq \epsilon$。

选择 $\epsilon = 1/100, \delta = 1/2$，令 $m \geq m(\epsilon,\delta)$，并设 $\mu = \frac{\log(100/99)}{2m}$。我们将定义两个分布，并展示 A 至少在一个分布上可能会失败。第一个分布 $\mathcal{D}_1$ 在两个例子 $z_1 = (1,0)$ 和 $z_2 = (\mu,-1)$ 上有支撑，其中第一个例子的概率质量为 μ，而第二个例子的概率质量为 $1-\mu$。第二个分布 $\mathcal{D}_2$ 完全在 z_2 上有支撑。

观察发现，对于这两种分布，训练集中所有示例都属于第二类的概率至少为99%。对于 $\mathcal{D}_2$，这一点显然成立，而对于 $\mathcal{D}_1$，这一事件的概率为

$$(1-\mu)^m \geq e^{-2\mu m} = 0.99.$$

自假设 A 是一个确定性算法，在接收到一个包含 m 个示例的训练集，每个示例为 $(\mu,-1)$ 后，该算法将输出一些 $\hat{w}$。现在，如果 $\hat{w} < -1/(2\mu)$，我们将设置分布为 $\mathcal{D}_1$。因此，

$$L_{\mathcal{D}_1}(\hat{w}) \geq \mu(\hat{w})^2 \geq 1/(4\mu).$$

然而，

$$\min_w L_{\mathcal{D}_1}(w) \leq L_{\mathcal{D}_1}(0) = (1-\mu).$$

因此，

$$L_{\mathcal{D}_1}(\hat{w}) - \min_w L_{\mathcal{D}_1}(w) \geq \frac{1}{4\mu} - (1-\mu) > \epsilon.$$

因此，此类算法 A 在 $\mathcal{D}_1$ 上失败。另一方面，如果 $\hat{w} \geq -1/(2\mu)$，则我们将分布设置为 $\mathcal{D}_2$。然后我们有 $L_{\mathcal{D}_2}(\hat{w}) \geq 1/4$，当 $\min_w L_{\mathcal{D}_2}(w) = 0$ 时，因此 A 在 $\mathcal{D}_2$ 上失败。总之，我们已经证明对于每个 A，都存在一个分布使得 A 失败，这意味着该问题不是 PAC 可学习的。

一个可能的解决方案是在假设类上添加另一个约束。除了凸性要求外，我们还需要$\mathcal{H}$将是*bounded*；也就是说，我们假设对于某个预定义的标量B，每个假设$\mathbf{w} \in \mathcal{H}$都满足$\|\mathbf{w}\| \leq B$。

有界性和凸性本身仍然不足以确保问题可学习，如下例所示。

Example 12.9 如示例12.8中所述，考虑一个具有平方损失的回归问题。然而，这次让 $\mathcal{H} = \{w : |w| \leq 1\} \subset \mathbb{R}$ 是一个有界的

[1] Namely, given S the output of A is determined. This requirement is for the sake of simplicity. A slightly more involved argument will show that nondeterministic algorithms will also fail to learn the problem.

假设类。容易验证 {v*} 是凸的。论证将与示例 12.8 中的相同，只是现在两个分布 $\mathcal{D}_1, \mathcal{D}_2$ 将在 $z_1 = (1/\mu, 0)$ 和 $z_2 = (1, -1)$ 上得到支持。如果算法 A 在接收到 m 个第二类示例后返回 $\hat{w} < -1/2$，那么我们将设置分布为 $\mathcal{D}_1$ 并有

$$L_{\mathcal{D}_1}(\hat{w}) - \min_w L_{\mathcal{D}_1}(w) \geq \mu(\hat{w}/\mu)^2 - L_{\mathcal{D}_1}(0) \geq 1/(4\mu) - (1-\mu) > \epsilon.$$

同样，如果 $\hat{w} \geq -1/2$ 我们将设置分布为 $\mathcal{D}_2$ 并保持那样

$$L_{\mathcal{D}_2}(\hat{w}) - \min_w L_{\mathcal{D}_2}(w) \geq (-1/2 + 1)^2 - 0 > \epsilon.$$

这个例子表明，我们需要对学习问题做出额外的假设，这次解决方案在于损失函数的Lipschitz连续性或平滑性。这促使我们定义了两类学习问题，凸-Lipschitz有界和凸-平滑有界，这些将在后面定义。

12.2.2 Convex-Lipschitz/Smooth-Bounded Learning Problems

DEFINITION 12.12 (凸-Lipschitz有界学习问题) 一个学习问题 $(\mathcal{H}, Z, \ell)$ 被称为具有参数 ρ, B 的凸-Lipschitz有界，如果以下条件成立：

- 假设类 $\mathcal{H}$ 是一个凸集，对于所有 $\mathbf{w} \in \mathcal{H}$，我们有 $\|\mathbf{w}\| \leq B$。
- 对于所有 $z \in Z$，损失函数 $\ell(\cdot, z)$ 是一个凸的且 ρ-Lipschitz 函数。

Example 12.10 让 $\mathcal{X} = \{\mathbf{x} \in \mathbb{R}^d : \|\mathbf{x}\| \leq \rho\}$ 和 $\mathcal{Y} = \mathbb{R}$。让 $\mathcal{H} = \{\mathbf{w} \in \mathbb{R}^d : \|\mathbf{w}\| \leq B\}$，并让损失函数为 $\ell(\mathbf{w}, (\mathbf{x}, y)) = |\langle \mathbf{w}, \mathbf{x} \rangle - y|$。这对应于一个具有绝对值损失的回归问题，其中我们假设实例位于半径为 ρ 的球体中，并将假设限制为由向量 $\mathbf{w}$ 定义的均匀线性函数，其范数受限于 B。然后，所得到的问题是具有参数 ρ, B 的凸-Lipschitz-Bounded。

DEFINITION 12.13 (凸-光滑-有界学习问题) 一个学习问题，$(\mathcal{H}, Z, \ell)$，如果满足以下条件，则称为具有参数 β, B 的凸-光滑-有界：

- 假设类 $\mathcal{H}$ 是一个凸集，对于所有 $\mathbf{w} \in \mathcal{H}$，我们有 $\|\mathbf{w}\| \leq B$。
- 对于所有 $z \in Z$，损失函数 $\ell(\cdot, z)$ 是一个凸的、非负的和 β-平滑的函数。

请注意，我们还要求损失函数非负。这是为了确保损失函数如前节所述是自限定的。

Example 12.11 让 $\mathcal{X} = \{\mathbf{x} \in \mathbb{R}^d : \|\mathbf{x}\| \leq \beta/\text{^}2\}$ 和 $\mathcal{Y} = \mathbb{R}$。让 $\mathcal{H} = \{\mathbf{w} \in \mathbb{R}^d : \|\mathbf{w}\| \leq B\}$，并让损失函数为 $\ell(\mathbf{w}, (\mathbf{x}, y)) = (\langle \mathbf{w}, \mathbf{x} \rangle - y)^2$。这对应于一个具有平方损失的回归问题，其中我们假设实例位于半径为 $\beta/\text{^}2$ 的球体中，并将假设限制为具有向量 $\mathbf{w}$ 的同质线性函数，其范数被限制在 B 内。然后，所得到的问题是具有参数 β, B 的凸-光滑-有界问题。

我们断言这两个学习问题族是可学习的。也就是说，损失函数的凸性、有界性和Lipschitz连续性或光滑性对于可学习性是足够的。我们将在下一章通过介绍成功学习这些问题的算法来证明这一主张。

12.3 Surrogate Loss Functions

如所述，并且我们将在下一章中看到，凸问题可以有效地学习。然而，在许多情况下，自然损失函数不是凸的，特别是实现ERM规则很困难。

作为一个例子，考虑关于$0-1$损失的半空间假设类学习问题。即，

$$\ell^{0-1}(\mathbf{w}, (\mathbf{x}, y)) = \mathbb{1}_{[y \neq \text{sign}(\langle \mathbf{w}, \mathbf{x} \rangle)]} = \mathbb{1}_{[y \langle \mathbf{w}, \mathbf{x} \rangle \leq 0]}.$$

此损失函数相对于 $\mathbf{w}$ 不是凸的，实际上，在尝试最小化相对于此损失函数的经验风险时，我们可能会遇到局部最小值（参见练习1）。此外，正如第8章所讨论的，在不可实现的情况下，解决关于$0-1$损失的ERM问题是已知的NP难题。

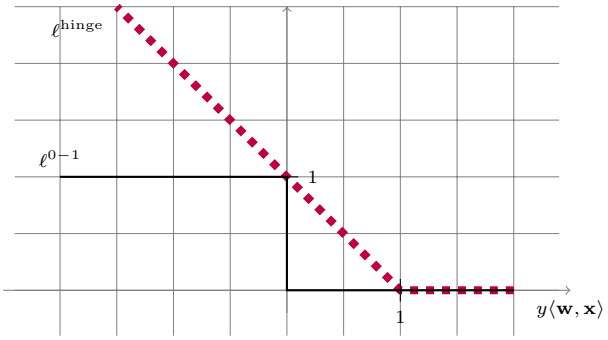
绕过硬度结果，一种流行的方法是将非凸损失函数通过凸代理损失函数进行上界。正如其名所示，凸代理损失的要求如下：

1. 它应该是凸的。2. 它应该上界原始损失
。

例如，在学习半空间的情况下，我们可以定义所谓的 *hinge* 损失作为 $0-1$ 损失的凸代理，如下所示：

$$\ell^{\text{hinge}}(\mathbf{w}, (\mathbf{x}, y)) \stackrel{\text{def}}{=} \max\{0, 1 - y\langle \mathbf{w}, \mathbf{x} \rangle\}.$$

显然，对于所有 $\mathbf{w}$ 和所有 $(\mathbf{x}, y)$，$\ell^{0-1}(\mathbf{w}, (\mathbf{x}, y)) \leq \ell^{\text{hinge}}(\mathbf{w}, (\mathbf{x}, y))$。此外，铰链损失的凸性直接由12.5命题得出。因此，铰链损失满足零一损失的凸代理损失函数的要求。以下给出了函数 ℓ^{0-1} 和 ℓ^{hinge} 的示意图。



一旦我们定义了代理凸损失，我们就可以根据它来学习问题。从铰链损失学习者的泛化要求将具有以下形式

$$L_{\mathcal{D}}^{\text{hinge}}(A(S)) \;\leq\; \min_{\mathbf{w}\in\mathcal{H}} L_{\mathcal{D}}^{\text{hinge}}(\mathbf{w}) + \epsilon,$$

在 $L_{\mathcal{D}}^{\text{hinge}}(\mathbf{w}) = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}[\ell^{\text{hinge}}(\mathbf{w},(\mathbf{x},y))]$。使用代理属性，我们可以将左侧的下界降低到 $L_{\mathcal{D}}^{0-1}(A(S))$，从而得到

$$L_{\mathcal{D}}^{0-1}(A(S)) \;\leq\; \min_{\mathbf{w}\in\mathcal{H}} L_{\mathcal{D}}^{\text{hinge}}(\mathbf{w}) + \epsilon.$$

我们可以进一步将上界重写如下：

$$L_{\mathcal{D}}^{0-1}(A(S)) \;\leq\; \min_{\mathbf{w}\in\mathcal{H}} L_{\mathcal{D}}^{0-1}(\mathbf{w}) + \left(\min_{\mathbf{w}\in\mathcal{H}} L_{\mathcal{D}}^{\text{hinge}}(\mathbf{w}) - \min_{\mathbf{w}\in\mathcal{H}} L_{\mathcal{D}}^{0-1}(\mathbf{w})\right) + \epsilon.$$

这意味着学习预测器的0−1误差被三个项的上界所限制：

- *Approximation error*这是术语min{v*}，它衡量假设类在分布上的表现好坏。我们已经在第五章中详细阐述了这一误差项。

- *Estimation error*这是由于我们只收到一个训练集而没有观察到分布 $\mathcal{D}$ 而产生的错误。我们已经在第五章中详细阐述了这一误差项。

- *Optimization error*这是衡量关于代理损失和原始损失的近似误差之间差异的术语 $\left(\min_{\mathbf{w}\in\mathcal{H}} L_{\mathcal{D}}^{\text{hinge}}(\mathbf{w}) - \min_{\mathbf{w}\in\mathcal{H}} L_{\mathcal{D}}^{0-1}(\mathbf{w})\right)$。优化误差是我们无法最小化关于原始损失的训练损失的结果。这个误差的大小取决于数据的特定分布以及我们使用的特定代理损失。

12.4 Summary

我们引入了两类学习问题：凸-利普希茨有界和凸-光滑有界。在接下来的两章中，我们将描述两种通用的

学习这些家族的学习算法。我们还引入了凸代理损失函数的概念，这使得我们也能够利用凸机制来解决非凸问题。

12.5 Bibliographic Remarks

存在几本关于凸分析和优化的优秀书籍（Boyd & Vandenberghe 2004，Borwein & Lewis 2006，Bertsekas 1999，Hiriart-Urruty & Lemaréchal 1996）。关于学习问题，凸-Lipschitz-有界问题族首先由Zinkevich（2003）在在线学习背景下研究，并由Shalev-Shwartz，Shamir，Sridharan和Srebro（2009）在PAC学习背景下研究。

12.6 Exercises

1. 构造一个示例，说明0−1损失函数可能存在局部最小值；即，构造一个训练样本$S \in (X \times \{\pm 1\})^m$（，对于$X = \mathbb{R}^2$），存在一个向量$\mathbf{w}$和一些$\epsilon > 0$，使得1. 对于任何$\mathbf{w}'$，使得$\|w - w'\| \leq \epsilon$，我们有$L_S(\mathbf{w}) \leq L_S(\mathbf{w}')$（，这里的损失是0−1损失）。这意味着$\mathbf{w}$是$L_S$的局部最小值。2. 存在一些$\mathbf{w}^*$使得$L_S(\mathbf{w}^*) < L_S(\mathbf{w})$。这意味着$\mathbf{w}$不是$L_S$的全局最小值。2. 考虑逻辑回归的学习问题：设$\mathcal{H} = \mathcal{X} = \{\mathbf{x} \in \mathbb{R}^d : \|\mathbf{x}\| \leq B\}$，对于某个标量$B > 0$，设$\mathcal{Y} = \{\pm 1\}$，并定义损失函数$\ell$为$\ell(\mathbf{w}, (\mathbf{x}, y)) = \log(1 + \exp(-y\langle \mathbf{w}, \mathbf{x}\rangle))$。证明所得到的 学习问题既是凸-Lipschitz有界的，也是凸-光滑有界的。指定Lipschitz性和光滑性的参数。3. 考虑使用铰链损失学习半空间的问题。我们将我们的域限制在半径为R的欧几里得球内。即，$\mathcal{X} = \{\mathbf{x} : \|\mathbf{x}\|_2 \leq R\}$。标签集是$\mathcal{Y} = \{\pm 1\}$，损失函数$\ell$定义为$\ell(\mathbf{w}, (\mathbf{x}, y)) = \max\{0, 1 - y\langle \mathbf{w}, \mathbf{x}\rangle\}$。我们已经知道损失函数是凸的。证明它是$R$-Lipschitz的。4.

(*) Convex-Lipschitz-Boundedness Is Not Sufficient for Computational Efficiency: 在下一章中，我们将从统计学的角度证明，所有凸-Lipschitz有界的问题都是可学习的（在无偏PAC模型中）。然而，我们学习这类问题的主要动机来自计算的角度——凸优化通常是有效可解的。然而，这个练习的目标是证明凸性本身不足以保证效率。我们证明，即使对于$d = 1$的情况，也存在一个凸-Lipschitz有界的问题，任何可计算的学习者都无法学习。

假设类为 $\mathcal{H} = [0, 1]$，并且让示例域，Z，为

所有图灵机的集合。定义损失函数如下。对于每个图灵机 $T \in Z$，令 $\ell(0,T) = 1$ 如果 T 在输入 0 上停止，并且 $\ell(0,T) = 0$ 如果 T 在输入 0 上不停止。同样，令 $\ell(1,T) = 0$ 如果 T 在输入 0 上停止，并且 $\ell(1,T) = 1$ 如果 T 在输入 0 上不停止。最后，对于 $h \in (0,1)$，令 $\ell(h,T) = h\ell(0,T) + (1-h)\ell(1,T)$。

1. 证明所得到的学习问题是凸-Lipschitz有界的。2. 证明没有可计算的算法可以学习这个问题。

13 Regularization and Stability

在上一章中，我们介绍了凸-Lipschitz有界和凸光滑有界学习问题的家族。在本节中，我们表明这两个家族中的所有学习问题都是可学习的。对于这类学习问题中的一些，可以证明一致收敛成立；因此，它们可以使用ERM规则进行学习。然而，这并不适用于所有这类学习问题。然而，我们将介绍另一种学习规则，并将证明它可以学习所有凸-Lipschitz有界和凸光滑有界学习问题。

新学习范式在本章中被称为 *Regularized Loss Minimization*，或简称RLM。在RLM中，我们最小化经验风险和正则化函数的总和。直观上，正则化函数衡量假设的复杂性。确实，正则化函数的一种解释是我们在第7章讨论的结构风险最小化范式。正则化的另一种观点是学习算法的 *stabilizer*。如果一个算法在输入略有变化时输出变化不大，则认为该算法是稳定的。我们将正式定义稳定性的概念（我们所说的“输入略有变化”和“输出变化不大”的含义）并证明其与学习能力的密切关系。最后，我们将展示使用平方 ℓ_2 范数作为正则化函数可以稳定所有凸-Lipschitz或凸-光滑学习问题。因此，RLM可以作为这些学习问题家族的一般学习规则。

13.1 Regularized Loss Minimization

Regularized Loss Minimization (RLM) 是一种学习规则，其中我们联合最小化经验风险和正则化函数。形式上，正则化函数是一个映射 $R : \mathbb{R}^d \to \mathbb{R}$，正则化损失最小化规则输出一个假设在

$$\operatorname*{argmin}_{\mathbf{w}} \left(L_S(\mathbf{w}) + R(\mathbf{w}) \right). \tag{13.1}$$

正则化损失最小化与最小描述长度算法和结构风险最小化相似（见第7章）。直观上，假设的“复杂性”通过正则化函数的值来衡量。

作用，并且算法在低经验风险和“更简单”或“更不复杂”的假设之间进行平衡。

存在许多可能的正则化函数可以选择，反映了关于问题的某些先验信念（类似于最小描述长度中的描述语言）。在本节中，我们将重点关注最简单的正则化函数之一：$R(\mathbf{w}) = \lambda\|\mathbf{w}\|^2$，其中$\lambda > 0$是一个标量，范数是$\ell_2$范数，$\|\mathbf{w}\| = \sqrt{\sum_{i=1}^{d} w_i^2}$。这产生了以下学习规则：

$$A(S) = \operatorname*{argmin}_{\mathbf{w}} \left(L_S(\mathbf{w}) + \lambda\|\mathbf{w}\|^2 \right). \tag{13.2}$$

这种正则化函数通常称为Tikhonov正则化。

如前所述，方程（13.2）的一种解释是使用结构风险最小化，其中$\mathbf{w}$的范数是其“复杂性”的度量。回想一下，在前一章中我们介绍了有界假设类概念。因此，我们可以定义一个假设类序列，$\mathcal{H}_1 \subset \mathcal{H}_2 \subset \mathcal{H}_3 \ldots$，其中$\mathcal{H}_i = \{\mathbf{w} : \|\mathbf{w}\|_2 \leq i\}$。如果每个$\mathcal{H}_i$的样本复杂度依赖于$i$，那么RLM规则与这个嵌套类序列的SRM规则相似。

正则化的另一种解释是作为稳定器。在下一节中，我们定义稳定性的概念并证明稳定的学习规则不会过拟合。但首先，让我们用平方损失来演示线性回归的RLM规则。

13.1.1 Ridge Regression

应用RLM规则和Tikhonov正则化到具有平方损失的线性回归中，我们得到以下学习规则：

$$\operatorname*{argmin}_{\mathbf{w}\in\mathbb{R}^d} \left(\lambda\|\mathbf{w}\|_2^2 + \frac{1}{m}\sum_{i=1}^{m} \frac{1}{2}(\langle \mathbf{w}, \mathbf{x}_i\rangle - y_i)^2 \right). \tag{13.3}$$

执行 线性回归使用方程（13.3）称为 *ridge reg ression.*

为了解方程（13.3），我们比较目标函数的梯度与零，并得到一组线性方程

$$(2\lambda m I + A)\mathbf{w} = \mathbf{b},$$

I 是 ID 实体矩阵和 $A, \mathbf{b}$ 如公式（9.6）中定义， 即，

$$A = \left(\sum_{i=1}^{m} \mathbf{x}_i \mathbf{x}_i^\top \right) \quad \text{and} \quad \mathbf{b} = \sum_{i=1}^{m} y_i \mathbf{x}_i \ . \tag{13.4}$$

由于 A 是一个正定矩阵，矩阵 $2\lambda m I + A$ 的所有特征值都由下界 $2\lambda m$ 界定。因此，这个矩阵是可逆的，岭回归的解变为

$$\mathbf{w} = (2\lambda m I + A)^{-1}\mathbf{b}. \tag{13.5}$$

在下一段中，我们正式展示正则化如何稳定算法并防止过拟合。特别是，下文（尤其是引理13.11）中提出的分析将得出：

THEOREM 13.1 *Let $\mathcal{D}$ be a distribution over $\mathcal{X} \times [-1, 1]$, where $\mathcal{X} = \{\mathbf{x} \in \mathbb{R}^d : \|\mathbf{x}\| \leq 1\}$. Let $\mathcal{H} = \{\mathbf{w} \in \mathbb{R}^d : \|\mathbf{w}\| \leq B\}$. For any $\epsilon \in (0, 1)$, let $m \geq 1$ 50 B^2/ϵ^2. Then, applying the ridge regression algorithm with parameter $\lambda = \epsilon/(3B^2)$ satisfies*

$$\mathop{\mathbb{E}}_{S \sim \mathcal{D}^m} [L_{\mathcal{D}}(A(S))] \;\leq\; \min_{\mathbf{w} \in \mathcal{H}} L_{\mathcal{D}}(\mathbf{w}) + \epsilon.$$

Remark 13.1 前述定理告诉我们需要多少个示例才能保证学习预测器的风险 *expected value* 被限制在类别近似误差加上 ϵ 的范围内。在通常的不可知PAC学习定义中，我们要求学习预测器的风险以至少$1 - \delta$ 的概率被限制。在练习1中，我们展示了如何使用具有有界期望风险的算法来构建不可知PAC学习器。

13.2 Stable Rules Do Not Overfit

直观上，如果一个学习算法对输入的小变化不产生很大的输出变化，那么这个算法是稳定的。当然，有许多方法来定义我们所说的“输入的小变化”以及“输出变化不大”的含义。在本节中，我们定义了一个特定的稳定性概念，并证明在这个定义下，稳定的规则不会过拟合。

设 A 为一个学习算法，设 $S = (z_1, \ldots, z_m)$ 为包含 m 个示例的训练集，并设 $A(S)$ 表示 A 的输出。如果算法 A 的输出真实风险 $L_{\mathcal{D}}(A(S))$ 与经验风险 $L_S(A(S))$ 之间的差异很大，则该算法会过拟合。如注释 13.1 所述，在本章中，我们关注这个量的期望（相对于 S 的选择），即 $\mathbb{E}_S[L_{\mathcal{D}}(A(S)) - L_S(A(S))]$。

我们接下来定义稳定性概念。给定训练集 S 和一个额外的示例 z'，令 $S^{(i)}$ 为通过将 S 中的第 i 个示例替换为 z' 而得到的训练集；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，“输入的小变化”意味着我们用 $S^{(i)}$ 替代 S 来输入 A。也就是说，我们只替换一个训练示例。我们通过比较假设 $A(S)$ 在 z_i 上的损失与假设 $A(S^{(i)})$ 在 z_i 上的损失来衡量这种输入小变化对 A 输出的影响。直观上，一个好的学习算法将使 $\ell(A(S^{(i)}), z_i) - \ell(A(S), z_i) \geq$ 为 0，因为在第一个项中学习算法没有观察到示例 z_i，而在第二个项中 z_i 确实被观察到。如果前面的差异非常大，我们怀疑学习算法可能过拟合。这是因为

学习算法在训练集中观察到 z_i 时，其预测会发生剧烈变化。这在下述定理中得到形式化。

THEOREM 13.2 *Let $\mathcal{D}$ be a distribution. Let $S = (z_1, \ldots, z_m)$ be an i.i.d. sequence of examples and let z' be another i.i.d. example. Let $U(m)$ be the uniform distribution over $[m]$. Then, for any learning algorithm,*

$$\mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}[L_{\mathcal{D}}(A(S)) - L_S(A(S))] = \mathop{\mathbb{E}}_{(S,z')\sim\mathcal{D}^{m+1},i\sim U(m)}[\ell(A(S^{(i)}, z_i)) - \ell(A(S), z_i)]. \tag{13.6}$$

Proof 由于 S 和 z' 都独立同分布地从 $\mathcal{D}$ 中抽取，因此对于每个 i，我们有

$$\mathop{\mathbb{E}}_{S}[L_{\mathcal{D}}(A(S))] = \mathop{\mathbb{E}}_{S,z'}[\ell(A(S), z')] = \mathop{\mathbb{E}}_{S,z'}[\ell(A(S^{(i)}), z_i)].$$

另一方面，我们可以写成

$$\mathop{\mathbb{E}}_{S}[L_S(A(S))] = \mathop{\mathbb{E}}_{S,i}[\ell(A(S), z_i)].$$

结合这两个方程，我们得出我们的证明。 □

当方程（13.6）的右侧很小时，我们称 A 是一个 *stable* 算法——改变训练集中单个示例不会导致显著变化。形式上，

DEFINITION 13.3 （平均替换一个稳定）设 $\epsilon : \mathbb{N} \to \mathbb{R}$ 是一个单调递减函数。我们称学习算法 A 在替换率为 $\epsilon(m)$ 时是平均替换一个稳定的，如果对于每个分布 $\mathcal{D}$

$$\mathop{\mathbb{E}}_{(S,z')\sim\mathcal{D}^{m+1},i\sim U(m)}[\ell(A(S^{(i)}, z_i)) - \ell(A(S), z_i)] \leq \epsilon(m).$$

定理13.2告诉我们，一个学习算法只有在平均替换一个稳定的情况下才不会过拟合。当然，一个不过拟合的学习算法不一定是好的学习算法——以一个总是输出相同假设的算法A为例。一个有用的算法应该找到一个假设，一方面适合训练集（即具有低经验风险），另一方面不会过拟合。或者，根据定理13.2，算法应该同时适合训练集并且是稳定的。正如我们将要看到的，RLM规则的参数λ在适合训练集和保持稳定之间取得平衡。

13.3 Tikhonov Regularization as a Stabilizer

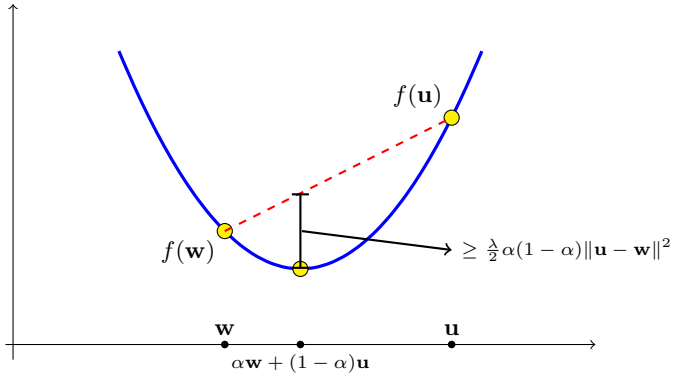
在上一节中，我们看到了稳定的规则不会过拟合。在本节中，我们展示了应用RLM规则与Tikhonov正则化$\lambda\|\mathbf{w}\|^2$，会导致一个稳定的算法。我们将假设损失函数是凸函数，并且它要么是Lipschitz连续的，要么是光滑的。

Tikhonov正则化的主要性质，我们依赖的是它使得RLM *strongly convex*的目标，如下定义。

DEFINITION 13.4 (强凸函数) 如果对于所有 $\mathbf{w}$, $\mathbf{u}$ 和 $\alpha \in (0,1)$，函数 f 是 λ-强凸的，则称其为 λ-强凸函数

$$f(\alpha\mathbf{w} + (1-\alpha)\mathbf{u}) \leq \alpha f(\mathbf{w}) + (1-\alpha)f(\mathbf{u}) - \frac{\lambda}{2}\alpha(1-\alpha)\|\mathbf{w}-\mathbf{u}\|^2.$$

显然，每个凸函数都是0-强凸的。以下图中给出了强凸性的示例。



以下引理表明RLM的目标是(2λ)-强凸的。此外，它强调了强凸性的一个重要性质。

LEMMA 13.5

1. *The function $f(\mathbf{w}) = \lambda\|\mathbf{w}\|2$ is 2λ-strongly convex.*
2. *If f is λ-strongly convex and g is convex, then $f+g$ is λ-strongly convex.*
3. *If f is λ-strongly convex and $\mathbf{u}$ is a minimizer of f, then, for any $\mathbf{w}$,*

$$f(\mathbf{w}) - f(\mathbf{u}) \geq \frac{\lambda}{2}\|\mathbf{w}-\mathbf{u}\|^2.$$

Proof 前两点直接由定义得出。为了证明最后一点，我们将强凸性的定义除以 α 并重新排列项，得到

$$\frac{f(\mathbf{u}+\alpha(\mathbf{w}-\mathbf{u})) - f(\mathbf{u})}{\alpha} \leq f(\mathbf{w}) - f(\mathbf{u}) - \frac{\lambda}{2}(1-\alpha)\|\mathbf{w}-\mathbf{u}\|^2.$$

取极限 $\alpha \to 0$，我们得到右侧收敛到 $f(\mathbf{w}) - f(\mathbf{u}) - \frac{\lambda}{2}\|\mathbf{w}-\mathbf{u}\|^2$。另一方面，左侧变为函数 $g(\alpha) = f(\mathbf{u}+\alpha(\mathbf{w}-\mathbf{u}))$ 在 $\alpha = 0$ 处的导数。由于 $\mathbf{u}$ 是 f 的极小值点，因此 $\alpha = 0$ 是 g 的极小值点，因此前面的左侧在极限 $\alpha \to 0$ 时趋于零，这完成了我们的证明。

□

我们现在转向证明RLM是稳定的。令$S = (z_1, \ldots, z_m)$为训练集，令z'为额外示例，并令$S^{(i)} = (z_1, \ldots, z_{i-1}, z', z_{i+1}, \ldots, z_m)$。令$A$为RLM规则，即，

$$A(S) = \operatorname*{argmin}_{\mathbf{w}} \left(L_S(\mathbf{w}) + \lambda\|\mathbf{w}\|^2 \right) .$$

表示 $f_S(\mathbf{w}) = L_S(\mathbf{w}) + \lambda\|\mathbf{w}\|^2$，根据引理13.5，我们知道 f_S 是 (2λ)-强凸的。根据引理的第3部分，对于任意的 $\mathbf{v}$，有

$$f_S(\mathbf{v}) - f_S(A(S)) \geq \lambda\|\mathbf{v} - A(S)\|^2. \tag{13.7}$$

另一方面，对于任意的 $\mathbf{v}$ 和 $\mathbf{u}$，以及所有的 i，我们有

$$\begin{aligned} f_S(\mathbf{v}) - f_S(\mathbf{u}) &= L_S(\mathbf{v}) + \lambda\|\mathbf{v}\|^2 - (L_S(\mathbf{u}) + \lambda\|\mathbf{u}\|^2) \\ &= L_{S^{(i)}}(\mathbf{v}) + \lambda\|\mathbf{v}\|^2 - (L_{S^{(i)}}(\mathbf{u}) + \lambda\|\mathbf{u}\|^2) \\ &\quad + \frac{\ell(\mathbf{v}, z_i) - \ell(\mathbf{u}, z_i)}{m} + \frac{\ell(\mathbf{u}, z') - \ell(\mathbf{v}, z')}{m}. \end{aligned} \tag{13.8}$$

特别地，选择 $\mathbf{v} = A(S^{(i)})$、$\mathbf{u} = A(S)$，并利用 $\mathbf{v}$ 最小化 $L_{S^{(i)}}(\mathbf{w}) + \lambda\|\mathbf{w}\|^2$ 的性质，我们得到：

$$f_S(A(S^{(i)})) - f_S(A(S)) \leq \frac{\ell(A(S^{(i)}), z_i) - \ell(A(S), z_i)}{m} + \frac{\ell(A(S), z') - \ell(A(S^{(i)}), z')}{m}. \tag{13.9}$$

将此与方程（13.7）结合，我们得到

$$\lambda\|A(S^{(i)}) - A(S)\|^2 \leq \frac{\ell(A(S^{(i)}), z_i) - \ell(A(S), z_i)}{m} + \frac{\ell(A(S), z') - \ell(A(S^{(i)}), z')}{m}. \tag{13.10}$$

接下来的两个小节继续对Lipschitz或光滑损失函数进行稳定性分析。对于这两个损失函数族，我们证明了RLM是稳定的，因此它不会过拟合。

13.3.1 Lipschitz Loss

如果损失函数 $\ell(\cdot, z_i)$ 是 ρ-Lipschitz，那么根据Lipschitz性的定义，

$$\ell(A(S^{(i)}), z_i) - \ell(A(S), z_i) \leq \rho\,\|A(S^{(i)}) - A(S)\|. \tag{13.11}$$

同样地，

$$\ell(A(S), z') - \ell(A(S^{(i)}), z') \leq \rho\,\|A(S^{(i)}) - A(S)\|.$$

将这些不等式代入方程（13.10）中，我们得到

$$\lambda\|A(S^{(i)}) - A(S)\|^2 \leq \frac{2\,\rho\,\|A(S^{(i)}) - A(S)\|}{m},$$

这导致

$$\|A(S^{(i)}) - A(S)\| \leq \frac{2\,\rho}{\lambda\,m}.$$

将前面的内容代入方程（13.11）中，我们得出结论

$$\ell(A(S^{(i)}), z_i) - \ell(A(S), z_i) \leq \frac{2\,\rho^2}{\lambda\,m}.$$

由于这对任何 S, z', i 都成立，我们立即得到：

COROLLARY 13.6 *Assume that the loss function is convex and ρ-Lipschitz. Then, the RLM rule with the regularizer $\lambda\|\mathbf{w}\|^2$ is on-average-replace-one-stable with rate $\frac{2\rho^2}{\lambda m}$. It follows (using Theorem 13.2) that*

$$\mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}[L_{\mathcal{D}}(A(S)) - L_S(A(S))] \leq \frac{2\,\rho^2}{\lambda\, m}.$$

13.3.2 Smooth and Nonnegative Loss

如果损失是 β-平滑且非负的，那么它也是自限定的（参见第12.1节）：

$$\|\nabla f(\mathbf{w})\|^2 \leq 2\beta f(\mathbf{w}). \tag{13.12}$$

我们进一步假设$\lambda \geq \frac{2\beta}{m}$，或者说，即$\beta \leq \lambda m/2$。根据平滑性假设，我们有

$$\ell(A(S^{(i)}), z_i) - \ell(A(S), z_i) \leq \langle \nabla \ell(A(S), z_i), A(S^{(i)}) - A(S)\rangle + \frac{\beta}{2}\|A(S^{(i)}) - A(S)\|^2 . \tag{13.13}$$

使用柯西-施瓦茨不等式和方程（12.6），我们进一步得到：{v*}

$$\begin{aligned}
&\ell(A(S^{(i)}), z_i) - \ell(A(S), z_i) \\
&\qquad \leq \|\nabla \ell(A(S), z_i)\|\, \|A(S^{(i)}) - A(S)\| + \frac{\beta}{2}\|A(S^{(i)}) - A(S)\|^2 \\
&\qquad \leq \sqrt{2\beta \ell(A(S), z_i)}\, \|A(S^{(i)}) - A(S)\| + \frac{\beta}{2}\|A(S^{(i)}) - A(S)\|^2 .
\end{aligned} \tag{13.14}$$

通过对称论证，可以得出以下结论，

$$\begin{aligned}
&\ell(A(S), z') - \ell(A(S^{(i)}), z') \\
&\qquad \leq \sqrt{2\beta \ell(A(S^{(i)}), z')}\, \|A(S^{(i)}) - A(S)\| + \frac{\beta}{2}\|A(S^{(i)}) - A(S)\|^2 .
\end{aligned}$$

将这些不等式代入方程（13.10）并重新排列项，我们得到

$$\|A(S^{(i)}) - A(S)\| \leq \frac{\sqrt{2\beta}}{(\lambda\, m - \beta)} \left(\sqrt{\ell(A(S), z_i)} + \sqrt{\ell(A(S^{(i)}), z')} \right).$$

C结合前面的假设 $\beta \leq \lambda m/2$ 产量

$$\|A(S^{(i)}) - A(S)\| \leq \frac{\sqrt{8\beta}}{\lambda\, m} \left(\sqrt{\ell(A(S), z_i)} + \sqrt{\ell(A(S^{(i)}), z')} \right).$$

将前面内容与方程（13.14）结合，再次使用假设 $\beta \leq \lambda m/\text{^}2$ 得到

$$\begin{aligned}
&\ell(A(S^{(i)}), z_i) - \ell(A(S), z_i) \\
&\quad \leq \sqrt{2\beta\ell(A(S), z_i)}\,\|A(S^{(i)}) - A(S)\| + \frac{\beta}{2}\|A(S^{(i)}) - A(S)\|^2 \\
&\quad \leq \left(\frac{4\beta}{\lambda m} + \frac{8\beta^2}{(\lambda m)^2}\right)\left(\sqrt{\ell(A(S), z_i)} + \sqrt{\ell(A(S^{(i)}), z')}\right)^2 \\
&\quad \leq \frac{8\beta}{\lambda m}\left(\sqrt{\ell(A(S), z_i)} + \sqrt{\ell(A(S^{(i)}), z')}\right)^2 \\
&\quad \leq \frac{24\beta}{\lambda m}\left(\ell(A(S), z_i) + \ell(A(S^{(i)}), z')\right),
\end{aligned}$$

在最后一步中我们使用了不等式 $(a+b)^2 \leq 3(a^2+b^2)$。对 $S,\ z',\ i$ 求期望，并注意到 $\mathbb{E}[\ell(A(S), z_i)] = \mathbb{E}[\ell(A(S^{(i)}),\ z')] = \mathbb{E}[L_S(A(S))]$，我们得出结论：

COROLLARY 13.7 *Assume that the loss function is β-smooth and nonnegative. Then, the RLM rule with the regularizer $\lambda\|\mathbf{w}\|^2$, where $\lambda \geq \frac{2\beta}{m}$, satisfies*

$$\mathbb{E}\left[\ell(A(S^{(i)}), z_i) - \ell(A(S), z_i)\right] \leq \frac{48\beta}{\lambda m}\,\mathbb{E}[L_S(A(S))].$$

注意，如果对于所有 z 我们有 $\ell(\mathbf{0},\ z) \leq C$，对于某个标量 $C > 0$，那么对于每一个 S，

$$L_S(A(S)) \leq L_S(A(S)) + \lambda\|A(S)\|^2 \leq L_S(\mathbf{0}) + \lambda\|\mathbf{0}\|^2 = L_S(\mathbf{0}) \leq C.$$

因此，推论13.7也意味着

$$\mathbb{E}\left[\ell(A(S^{(i)}), z_i) - \ell(A(S), z_i)\right] \leq \frac{48\,\beta\,C}{\lambda m}.$$

13.4 Controlling the Fitting-Stability Tradeoff

我们可以将学习算法的预期风险重写为

$$\mathop{\mathbb{E}}_{S}[L_{\mathcal{D}}(A(S))] = \mathop{\mathbb{E}}_{S}[L_S(A(S))] + \mathop{\mathbb{E}}_{S}[L_{\mathcal{D}}(A(S)) - L_S(A(S))]. \tag{13.15}$$

第一个项反映了$A(S)$与训练集拟合的好坏，而第二个项反映了真实风险与$A(S)$经验风险之间的差异。正如我们在定理13.2中所示，第二个项等价于A的稳定性。由于我们的目标是使算法的风险最小化，我们需要这两个项的和尽可能小。

在上一节中，我们已经界定了稳定性项。我们已经证明，随着正则化参数 λ 的增加，稳定性项会减少。另一方面，经验风险随着 λ 的增加而增加。因此，我们面临一个

权衡拟合和过拟合。这种权衡与我们之前在书中讨论的偏差-复杂度权衡相当相似。

我们现在推导RLM规则的经验风险项的界限。回忆RLM规则定义为$A(S) = \text{ar}$ $\text{gmin}_{\mathbf{w}} \left(L_S(\mathbf{w}) + \lambda\|\mathbf{w}\|^2\right)$。固定某个任意向量$\mathbf{w}^*$。我们有

$$L_S(A(S)) \leq L_S(A(S)) + \lambda\|A(S)\|^2 \leq L_S(\mathbf{w}^*) + \lambda\|\mathbf{w}^*\|^2.$$

对两边关于 S 求期望，并注意到 $\mathbb{E}_S[L_S(\mathbf{w}^*)] = L_{\mathcal{D}}(\mathbf{w}^*)$，我们得到

$$\mathop{\mathbb{E}}_{S}[L_S(A(S))] \leq L_{\mathcal{D}}(\mathbf{w}^*) + \lambda\|\mathbf{w}^*\|^2. \tag{13.16}$$

将此代入方程（13.15）中，我们得到

$$\mathop{\mathbb{E}}_{S}[L_{\mathcal{D}}(A(S))] \leq L_{\mathcal{D}}(\mathbf{w}^*) + \lambda\|\mathbf{w}^*\|^2 + \mathop{\mathbb{E}}_{S}[L_{\mathcal{D}}(A(S)) - L_S(A(S))].$$

结合前文与引理13.6，我们得出结论 de:

COROLLARY 13.8 *Assume that the loss function is convex and ρ-Lipschitz. Then, the RLM rule with the regularization function $\lambda\|\mathbf{w}\|^2$ satisfies*

$$\forall \mathbf{w}^*,\ \mathop{\mathbb{E}}_{S}[L_{\mathcal{D}}(A(S))] \leq L_{\mathcal{D}}(\mathbf{w}^*) + \lambda\|\mathbf{w}^*\|^2 + \frac{2\rho^2}{\lambda\, m}.$$

这个界限通常被称为一个 *oracle inequality* —— 如果我们将 $\mathbf{w}^*$ 视为一个低风险的假设，那么这个界限告诉我们需要多少个示例，以便 $A(S)$ 几乎和 $\mathbf{w}^*$ 一样好，就像我们已知 $\mathbf{w}^*$ 的范数一样。然而，在实践中，我们通常不知道 $\mathbf{w}^*$ 的范数。因此，我们通常根据第11章中描述的验证集来调整 λ。

我们也可以轻松地从引理13.8推导出类似于PAC的保证[1]，用于凸-李普希茨有界学习问题：

COROLLARY 13.9 *Let $(\mathcal{H}, Z, \ell)$ be a convex-Lipschitz-bounded learning problem with parameters ρ, B. For any training set size m, let $\lambda = \sqrt{\frac{2\rho^2}{B^2\, m}}$. Then, the RLM rule with the regularization function $\lambda\|\mathbf{w}\|^2$ satisfies*

$$\mathop{\mathbb{E}}_{S}[L_{\mathcal{D}}(A(S))] \leq \min_{\mathbf{w}\in\mathcal{H}} L_{\mathcal{D}}(\mathbf{w}) + \rho\, B\sqrt{\frac{8}{m}}.$$

In particular, for every $\epsilon > 0$, if $m \geq \frac{8\rho^2 B^2}{\epsilon^2}$ then for every distribution $\mathcal{D}$, $\mathbb{E}_S[L_{\mathcal{D}}(A(S))] \leq \min_{\mathbf{w}\in\mathcal{H}} L_{\mathcal{D}}(\mathbf{w}) + \epsilon$.

前述推论适用于Lipschitz损失函数。如果损失函数是光滑且非负的，那么我们可以将方程（13.16）与推论13.7相结合，得到：

1 再次，下面的界限是在预期风险上，但使用练习1，它可以用来推导出一个无监督的PAC学习保证。

COROLLARY 13.10 *Assume that the loss function is convex, β-smooth, and nonnegative. Then, the RLM rule with the regularization function $\lambda\|\mathbf{w}\|^2$, for $\lambda \geq \frac{2\beta}{m}$, satisfies the following for all $\mathbf{w}^*$:*

$$\mathop{\mathbb{E}}_{S}[L_{\mathcal{D}}(A(S))] \leq \left(1 + \frac{48\beta}{\lambda m}\right) \mathop{\mathbb{E}}_{S}[L_S(A(S))] \leq \left(1 + \frac{48\beta}{\lambda m}\right) \left(L_{\mathcal{D}}(\mathbf{w}^*) + \lambda\|\mathbf{w}^*\|^2\right).$$

例如，如果我们选择 $\lambda = \frac{48\beta}{m}$，从前面的内容中我们可以得到 $A(S)$ 的预期真实风险至多为 $A(S)$ 的预期经验风险的两倍。此外，对于这个 λ 的值，$A(S)$ 的预期经验风险至多为 $L_{\mathcal{D}}(\mathbf{w}^*) + \frac{48\beta}{m}\|\mathbf{w}^*\|^2$。

我们也可以根据引理13.10推导出凸平滑有界学习问题的可学习性保证。

COROLLARY 13.11 *Let $(\mathcal{H}, Z, \ell)$ be a convex-smooth-bounded learning problem with parameters β, B. Assume in addition that $\ell(\mathbf{0}, z) \leq 1$ for all $z \in Z$. For any $\epsilon \in (0,1)$ let $m \geq \frac{150\beta B^2}{\epsilon^2}$ and set $\lambda = \epsilon/(3B^2)$. Then, for every distribution $\mathcal{D}$,*

$$\mathop{\mathbb{E}}_{S}[L_{\mathcal{D}}(A(S))] \leq \min_{\mathbf{w}\in\mathcal{H}} L_{\mathcal{D}}(\mathbf{w}) + \epsilon\ .$$

13.5 Summary

我们引入了稳定性并表明如果一个算法是稳定的，那么它不会过拟合。此外，对于凸-Lipschitz有界或凸平滑有界问题，RLM规则与Tikhonov正则化相结合导致稳定的学习算法。我们讨论了正则化参数λ如何控制拟合和过拟合之间的权衡。最后，我们已证明所有来自凸-Lipschitz有界和凸平滑有界问题家族的学习问题都可以使用RLM规则进行学习。RLM范例是许多流行学习算法的基础，包括岭回归（我们在本章中讨论过）和支持向量机（将在第15章中讨论）。

在下章中，我们将介绍随机梯度下降，它为我们提供了一种非常实用的学习凸-Lipschitz有界和凸-光滑有界问题的替代方法，也可以用于有效地实现RLM规则。

13.6 Bibliographic Remarks

稳定性在许多数学情境中被广泛使用。例如，所谓的逆问题需要稳定性才能良好地提出，这一必要性首先由哈达玛德（1902年）认识到。正则化的概念及其与稳定性的关系通过蒂赫诺夫（1943年）和菲利普斯（1962年）的工作而广为人知。

在现代学习理论背景下，稳定性的使用可以追溯到至少Rogers & Wagner（1978）的工作，他们指出，关于样本小变化的敏感性控制了学习算法的留一法估计的方差。作者利用这一观察结果获得了k-最近邻算法的泛化界限（见第19章）。这些结果后来被扩展到其他“局部”学习算法（见Devroye, Györfi & Lugosi（1996）及其参考文献）。此外，还开发了将稳定性引入学习算法的实用方法，特别是Breiman（1996）引入的Bagging技术。

过去十年中，稳定性被研究为学习的一般条件。参见（Kearns & Ron 1999，Bousquet & Elisseeff 2002，Kutin & Niyogi 2002，Rakhlin，Mukherjee & Poggio 2005，Mukherjee，Niyogi，Poggio & Rifkin 2006）。我们的演示遵循Shalev-Shwartz，Shamir，Srebro & Sridhara-n（2010）的工作，他们证明了稳定性是学习的充分必要条件。他们还表明，所有凸-Lipschitz有界的学习问题都可以使用RLM进行学习，即使对于某些凸-Lipschitz有界的学习问题，在强意义上并不满足一致收敛。

13.7 Exercises

1. **From Bounded Expected Risk to Agnostic PAC Learning:** 设 A 是一个保证以下性质的算法：如果 $m \geq m_{\mathcal{H}}(\epsilon)$，那么对于每个分布 $\mathcal{D}$ 都有

$$\mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}[L_{\mathcal{D}}(A(S))] \leq \min_{h\in\mathcal{H}} L_{\mathcal{D}}(h) + \epsilon.$$

- 证明对于每个 $\delta \in (0,1)$，如果 $m \geq m_{\mathcal{H}}(\epsilon\,\delta)$，则至少以 $1-\delta$ 的概率有 $L_{\mathcal{D}}(A(S)) \leq \min_{h\in\mathcal{H}} L_{\mathcal{D}}(h) + \epsilon$ 成立。*Hint:* 注意到随机变量 $L_{\mathcal{D}}(A(S)) - \min_{h\in\mathcal{H}} L_{\mathcal{D}}(h)$ 是非负的，并依赖于马尔可夫不等式。

- 对于每个 $\delta \in (0,1)$ 让

$$m_{\mathcal{H}}(\epsilon,\delta) = m_{\mathcal{H}}(\epsilon/2)\lceil\log_2(1/\delta)\rceil + \left\lceil \frac{\log(4/\delta) + \log(\lceil\log_2(1/\delta)\rceil)}{\epsilon^2} \right\rceil.$$

  建议一个过程，使无偏PAC学习具有样本复杂度$m_{\mathcal{H}}(\epsilon,\delta)$的问题，假设损失函数被1所限制。

  Hint: 让 $k = \lceil\log_2(1/\delta)\rceil$。将数据分成 $k+1$ 块，其中前 k 块的大小为 $m_{\mathcal{H}}(\epsilon/2)$ 个示例。使用 A 训练前 k 块。基于前一个问题，论证对于所有这些块，我们都有 $L_{\mathcal{D}}(A(S)) > \min_{h\in\mathcal{H}} L_{\mathcal{D}}(h) + \epsilon$ 的概率至多为 $2^{-k} \leq \delta/2$。最后，使用最后一个块作为验证集。

2. **Learnability without Uniform Convergence:** 设 $\mathcal{B}$ 为单位球

$\mathbb{R}^d$, let $\mathcal{H} = \mathcal{B}$, let $Z = \mathcal{B} \times \{0,1\}^d$, and let $\ell : Z \times \mathcal{H} \to \mathbb{R}$ be defined as follows:

$$\ell(\mathbf{w}, (\mathbf{x}, \boldsymbol{\alpha})) = \sum_{i=1}^{d} \alpha_i (x_i - w_i)^2.$$

这个问题对应于一个 *unsupervised* 学习任务，意味着我们并不试图预测 $\mathbf{x}$ 的标签。相反，我们试图找到分布 $\mathcal{B}$ 的“质心”。然而，有一个转折，由向量 $\boldsymbol{\alpha}$ 模型。每个示例都是一个对 $(\mathbf{x}, \boldsymbol{\alpha})$，其中 $\mathbf{x}$ 是实例 $\mathbf{x}$，而 $\boldsymbol{\alpha}$ 表示 $\mathbf{x}$ 的哪些特征是“激活”的，哪些是“关闭”的。一个假设是一个向量 $\mathbf{w}$，表示分布的质心，损失函数是 $\mathbf{x}$ 和 $\mathbf{w}$ 之间的平方欧几里得距离，但仅针对 $\mathbf{x}$ 的“激活”元素。

- 证明使用RLM规则，该问题可学习，其样本复杂度不依赖于d。
- 考虑在Z上的分布$\mathcal{D}$如下：$\mathbf{x}$被固定为某个$\mathbf{x}_0$，$\boldsymbol{\alpha}$的每个元素以相等的概率采样为1或0。证明此问题的均匀收敛速率随着d增长。*Hint:*设m为训练集大小。证明如果$d \gg$属于^m，那么有很高的概率采样一组示例，对于其中某些$j \in [d]$，训练集中的所有示例都满足$\alpha_j =$等于1。证明这样的样本不能是ϵ表示的。得出结论，均匀收敛的样本复杂度必须随着log(d)增长。
- 结论是，如果我们将 d 取到无穷大，我们得到一个可学习的问题，但对于该问题，一致收敛性质不成立。与统计学习的基本定理进行比较。

3. **Stability and Asymptotic ERM Are Sufficient for Learnability:**

我们称一个学习规则 A 是一个具有速率 $\epsilon(m)$ 的 *AERM (Asymptotic Empirical Risk Minimizer)*，如果对于每个分布 $\mathcal{D}$ 都成立，那么

$$\mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}\left[L_S(A(S)) - \min_{h\in\mathcal{H}} L_S(h)\right] \le \epsilon(m).$$

我们说一个学习规则 A 以速率 $\epsilon(m)$ 学习一个类别 $\mathcal{H}$，如果对于每个分布 $\mathcal{D}$ 都成立，那么

$$\mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}\left[L_{\mathcal{D}}(A(S)) - \min_{h\in\mathcal{H}} L_{\mathcal{D}}(h)\right] \le \epsilon(m).$$

证明以下内容：

THEOREM 13.12 *If a learning algorithm A is on-average-replace-one-stable with rate $\epsilon_1(m)$ and is an AERM with rate $\epsilon_2(m)$, then it learns $\mathcal{H}$ with rate $\epsilon_1(m) + \epsilon_2(m)$.*

4. **Strong Convexity with Respect to General Norms:**
在整个章节中，我们使用了 ℓ_2 范数。在本练习中，我们将一些结果推广到一般范数。设 $\|\cdot\|$ 为某个任意范数，设 f 为相对于此范数的强凸函数（参见定义13.4）。

1. 证明命题13.5的第2-3项对每个范数都成立。
2. (*) 给出一个不满足引理13.5的第1项的范数的例子。

3. 令 $R(\mathbf{w})$ 为一个相对于某个范数 $\|\cdot\|$ 具有强凸性 (2λ) 的函数。令 A 为相对于 R 的 RLM 规则，即，

$$A(S) = \operatorname*{argmin}_{\mathbf{w}} \left(L_S(\mathbf{w}) + R(\mathbf{w}) \right) .$$

假设对于每个 z，损失函数 $\ell(\cdot, z)$ 在相同的范数下是 ρ-Lipschitz 连续的，即，

$$\forall z,\ \forall \mathbf{w}, \mathbf{v}, \quad \ell(\mathbf{w}, z) - \ell(\mathbf{v}, z) \leq \rho \|\mathbf{w} - \mathbf{v}\| .$$

证明 A 以 $\frac{2\rho^2}{\lambda m}$ 的比率平均替换稳定。

4. (*) 设 $q \in (1, 2)$ 并考虑 ℓ_q-范数

$$\|\mathbf{w}\|_q = \left(\sum_{i=1}^{d} \right)^{1/q}$$

可以证明（例如，参见Shalev-Shwartz [Shalev-Shwartz and Ben-David (2014)]）.该函数

$$R(\mathbf{w}) = \frac{1}{2(q-1)} \|\mathbf{w}\|_q^2$$

是相对于 $\|\mathbf{w}\|_q$ 的 1-强凸。证明如果 $q = \frac{\log(d)}{\log(d)-1}$，那么 $R(\mathbf{w})$ 在 $\mathbb{R}^d$ 上的 ℓ_1 范数下是 $\left(\frac{1}{3\log(d)}\right)$-强凸的。

14 Stochastic Gradient Descent

回忆一下，学习的目标是最小化风险函数，$L_{\mathcal{D}}(h) = \mathbb{E}_{z\sim\mathcal{D}}[\ell(h,z)]$。由于它依赖于未知的分布 $\mathcal{D}$，我们无法直接最小化风险函数。到目前为止，本书中我们讨论了依赖于经验风险的学习方法。也就是说，我们首先采样一个训练集 S 并定义经验风险函数 $L_S(h)$。然后，学习器根据 $L_S(h)$ 的值选择一个假设。例如，ERM 规则告诉我们从假设类 $\mathcal{H}$ 中选择最小化 $L_S(h)$ 的假设。或者，在前一章中，我们讨论了正则化风险最小化，其中我们选择一个假设，该假设同时最小化 $L_S(h)$ 和正则化函数 h。

在这一章中，我们描述并分析了一种相当不同的学习方法，称为 *Stochastic Gradient Descent* (SGD)。正如在第12章中，我们将重点关注凸学习问题的重要家族，并遵循该章节的符号，我们将把假设称为来自凸假设类$\mathcal{H}$的向量$\mathbf{w}$。在SGD中，我们尝试通过梯度下降过程直接最小化风险函数$L_{\mathcal{D}}(\mathbf{w})$。梯度下降是一种迭代优化过程，在每一步中，我们通过沿着当前点要最小化的函数梯度的负方向迈出一步来改进解。当然，在我们的情况下，我们最小化风险函数，由于我们不知道$\mathcal{D}$，因此也不知道$L_{\mathcal{D}}(\mathbf{w})$的梯度。SGD通过允许优化过程沿着一个随机方向迈出一步来绕过这个问题，只要该方向的期望值是梯度的负值。而且，正如我们将看到的，找到一个期望值对应于梯度的随机方向相当简单，即使我们不知道潜在的分布$\mathcal{D}$。

SGD在凸学习问题中的优势，与正则化风险最小化学习规则相比，在于SGD是一个高效的算法，可以在几行代码中实现，同时仍然享有与正则化风险最小化规则相同的样本复杂度。SGD的简单性还允许我们在无法应用基于经验风险的方法的情况下使用它，但这超出了本书的范围。

我们本章从基本的梯度下降算法开始，分析其在凸-Lipschitz函数上的收敛速度。接下来，我们引入子梯度概念，并证明梯度下降也可以应用于不可微函数。本章的核心是第14.3节，其中我们描述

随机梯度下降算法，以及几个有用的变体。我们表明，SGD具有与梯度下降速率相似的期望收敛速率。最后，我们转向SGD在学习问题中的应用。

14.1 Gradient Descent

在描述随机梯度下降法之前，我们想先描述标准梯度下降法以最小化可微凸函数 $f(\mathbf{w})$。

梯度函数 f 在 $\mathbb{R}^d \to \mathbb{R}$ 上关于 $\mathbf{w}$ 的梯度，记为 $\nabla f(\mathbf{w})$，是 f 的偏导数向量，即 $\nabla f(\mathbf{w}) = \left(\frac{\partial f(\mathbf{w})}{\partial w[1]}, \ldots, \frac{\partial f(\mathbf{w})}{\partial w[d]}\right)$。梯度下降是一种迭代算法。我们从初始值 $\mathbf{w}$ (，即 $\mathbf{w}^{(1)} = \mathbf{0}$) 开始。然后，在每次迭代中，我们在当前点的负梯度方向上迈出一步。也就是说，更新步骤是

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta \nabla f(\mathbf{w}^{(t)}), \tag{14.1}$$

在 $\eta > 0$ 是稍后要讨论的参数。直观上，因为梯度指向 f 在 $\mathbf{w}^{(t)}$ 附近增加最快的方向，算法在相反方向迈出小步，从而降低函数的值。最终，经过 T 次迭代后，算法输出平均向量，$\bar{\mathbf{w}} = \frac{1}{T}\sum_{t=1}^{T} \mathbf{w}^{(t)}$。输出也可以是最后一个向量，$\mathbf{w}^{(T)}$ ，或者性能最好的向量，$\operatorname{argmin}_{t\in[T]} f(\mathbf{w}^{(t)})$，但取平均值证明相当有用，尤其是在我们将梯度下降推广到非可微函数和随机情况时。

另一种激励梯度下降的方法是依赖于泰勒展开。在 f 处的 $\mathbf{w}$ 的梯度给出了 f 在 $\mathbf{w}$ 附近的 $f(\mathbf{u}) \approx f(\mathbf{w}) + \langle \mathbf{u} - \mathbf{w}, \nabla f(\mathbf{w}) \rangle$ 的一阶泰勒近似。当 f 是凸函数时 ，这个近似下界于 f，即，

$$f(\mathbf{u}) \geq f(\mathbf{w}) + \langle \mathbf{u} - \mathbf{w}, \nabla f(\mathbf{w}) \rangle.$$

因此，对于接近 $\mathbf{w}^{(t)}$ 的 $\mathbf{w}$，我们有 $f(\mathbf{w}) \approx f(\mathbf{w}^{(t)}) + \langle \mathbf{w} - \mathbf{w}^{(t)}, \nabla f(\mathbf{w}^{(t)}) \rangle$。因此 ，我们可以最小化 $f(\mathbf{w})$ 的近似。然而，对于远离 $\mathbf{w}^{(t)}$ 的 $\mathbf{w}$，近似可能会变得宽松。因此，我们希望联合最小化 $\mathbf{w}$ 和 $\mathbf{w}^{(t)}$ 之间的距离以及 $\mathbf{w}^{(t)}$ 附近的 f 的近似。如果参数 η 控制这两个项之间的权衡，我们得到以下更新规则

$$\mathbf{w}^{(t+1)} = \operatorname*{argmin}_{\mathbf{w}} \frac{1}{2}\|\mathbf{w} - \mathbf{w}^{(t)}\|^2 + \eta \left(f(\mathbf{w}^{(t)}) + \langle \mathbf{w} - \mathbf{w}^{(t)}, \nabla f(\mathbf{w}^{(t)}) \rangle \right).$$

通过求导数相对于 $\mathbf{w}$ 并将其与零比较来解前面的方程，得到与方程（14.1）相同的更新规则。

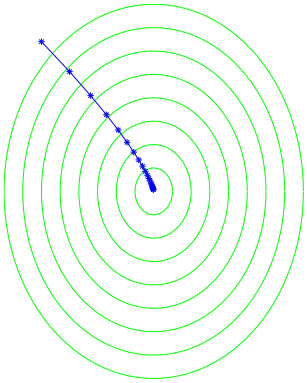


Figure 14.1 一个梯度下降算法的示意图。要最小化的函数是 $1.25(x_1+6)^2+(x_2-8)^2$。

14.1.1 Analysis of GD for Convex-Lipschitz Functions

为了分析GD算法的收敛速度，我们限制自己考虑凸-Lipschitz函数的情况（正如我们所见，许多问题很容易适应这种设置）。设$\mathbf{w}^\star$为任意向量，设B为$\|\mathbf{w}^\star\|$的上界。将$\mathbf{w}^\star$视为$f(\mathbf{w})$的最小值是方便的，但以下分析适用于每个$\mathbf{w}^\star$。

我们希望获得关于 $\mathbf{w}^\star$ 的解的次优性的上界，即 $f(\bar{\mathbf{w}})-f(\mathbf{w}^\star)$，其中 $\bar{\mathbf{w}}=\frac{1}{T}\sum_{t=1}^{T}\mathbf{w}^{(t)}$。根据$\bar{\mathbf{w}}$ 的定义，并使用 Jensen 不等式，我们有

$$\begin{aligned} f(\bar{\mathbf{w}})-f(\mathbf{w}^\star) &= f\left(\frac{1}{T}\sum_{t=1}^{T}\mathbf{w}^{(t)}\right)-f(\mathbf{w}^\star) \\ &\le \frac{1}{T}\sum_{t=1}^{T}\left(f(\mathbf{w}^{(t)})\right)-f(\mathbf{w}^\star) \\ &= \frac{1}{T}\sum_{t=1}^{T}\left(f(\mathbf{w}^{(t)})-f(\mathbf{w}^\star)\right). \end{aligned} \tag{14.2}$$

对于每个 t，由于 f 的凸性，我们有

$$f(\mathbf{w}^{(t)})-f(\mathbf{w}^\star) \le \langle \mathbf{w}^{(t)}-\mathbf{w}^\star, \nabla f(\mathbf{w}^{(t)})\rangle. \tag{14.3}$$

结合前面的内容，我们得到

$$f(\bar{\mathbf{w}})-f(\mathbf{w}^\star) \le \frac{1}{T}\sum_{t=1}^{T}\langle \mathbf{w}^{(t)}-\mathbf{w}^\star, \nabla f(\mathbf{w}^{(t)})\rangle.$$

为了界定右侧，我们依赖于以下引理：

LEMMA 14.1 *Let $\mathbf{v}_1, \ldots, \mathbf{v}_T$ be an arbitrary sequence of vectors. Any algorithm with an initialization $\mathbf{w}^{(1)} = 0$ and an update rule of the form*

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta \mathbf{v}_t \tag{14.4}$$

satisfies

$$\sum_{t=1}^{T} \langle \mathbf{w}^{(t)} - \mathbf{w}^\star, \mathbf{v}_t \rangle \leq \frac{\|\mathbf{w}^\star\|^2}{2\eta} + \frac{\eta}{2} \sum_{t=1}^{T} \|\mathbf{v}_t\|^2. \tag{14.5}$$

In particular, for every $B, \rho > 0$, if for all t we have that $\|\mathbf{v}_t\| \leq \rho$ and if we set $\eta = \sqrt{\frac{B^2}{\rho^2 T}}$, then for every $\mathbf{w}^\star$ with $\|\mathbf{w}^\star\| \leq B$ we have

$$\frac{1}{T} \sum_{t=1}^{T} \langle \mathbf{w}^{(t)} - \mathbf{w}^\star, \mathbf{v}_t \rangle \leq \frac{B\rho}{\sqrt{T}}.$$

Proof 使用代数运算（完成平方），我们得到：

$$\begin{aligned}
\langle \mathbf{w}^{(t)} - \mathbf{w}^\star, \mathbf{v}_t \rangle &= \frac{1}{\eta} \langle \mathbf{w}^{(t)} - \mathbf{w}^\star, \eta \mathbf{v}_t \rangle \\
&= \frac{1}{2\eta} (-\|\mathbf{w}^{(t)} - \mathbf{w}^\star - \eta \mathbf{v}_t\|^2 + \|\mathbf{w}^{(t)} - \mathbf{w}^\star\|^2 + \eta^2 \|\mathbf{v}_t\|^2) \\
&= \frac{1}{2\eta} (-\|\mathbf{w}^{(t+1)} - \mathbf{w}^\star\|^2 + \|\mathbf{w}^{(t)} - \mathbf{w}^\star\|^2) + \frac{\eta}{2} \|\mathbf{v}_t\|^2,
\end{aligned}$$

在最后一个等式由更新规则的定义得出。对等式在 t 上求和，我们得到

$$\sum_{t=1}^{T} \langle \mathbf{w}^{(t)} - \mathbf{w}^\star, \mathbf{v}_t \rangle = \frac{1}{2\eta} \sum_{t=1}^{T} \left(-\|\mathbf{w}^{(t+1)} - \mathbf{w}^\star\|^2 + \|\mathbf{w}^{(t)} - \mathbf{w}^\star\|^2 \right) + \frac{\eta}{2} \sum_{t=1}^{T} \|\mathbf{v}_t\|^2. \tag{14.6}$$

右侧的第一个和是一个望远镜和，它折叠成

$$\|\mathbf{w}^{(1)} - \mathbf{w}^\star\|^2 - \|\mathbf{w}^{(T+1)} - \mathbf{w}^\star\|^2.$$

将此代入方程（14.6），我们得到

$$\begin{aligned}
\sum_{t=1}^{T} \langle \mathbf{w}^{(t)} - \mathbf{w}^\star, \mathbf{v}_t \rangle &= \frac{1}{2\eta} (\|\mathbf{w}^{(1)} - \mathbf{w}^\star\|^2 - \|\mathbf{w}^{(T+1)} - \mathbf{w}^\star\|^2) + \frac{\eta}{2} \sum_{t=1}^{T} \|\mathbf{v}_t\|^2 \\
&\leq \frac{1}{2\eta} \|\mathbf{w}^{(1)} - \mathbf{w}^\star\|^2 + \frac{\eta}{2} \sum_{t=1}^{T} \|\mathbf{v}_t\|^2 \\
&= \frac{1}{2\eta} \|\mathbf{w}^\star\|^2 + \frac{\eta}{2} \sum_{t=1}^{T} \|\mathbf{v}_t\|^2,
\end{aligned}$$

在最后一个等式是由于定义 $\mathbf{w}^{(1)} = 0$。这证明了引理的第一部分（方程（14.5））。第二部分通过将 $\|\mathbf{w}^\star\|$ 上界为 B，$\|\mathbf{v}_t\|$ 上界为 ρ，除以 T，并插入 η 的值来得出。 □

引理14.1适用于GD算法中的$\mathbf{v}_t = \nabla f(\mathbf{w}^{(t)})$。正如我们将在引理14.7中所示，如果$f$是$\rho$-Lipschitz，那么$\|\nabla f(\mathbf{w}^{(t)})\| \leq \rho$。因此，我们满足引理的条件，并得到以下推论：

COROLLARY 14.2 *Let f be a convex, ρ-Lipschitz function, and let $\mathbf{w}^\star \in \operatorname{argmin}_{\{\mathbf{w}:\|\mathbf{w}\|\leq B\}} f(\mathbf{w})$. If we run the GD algorithm on f for T steps with $\eta = \sqrt{\frac{B^2}{\rho^2 T}}$, then the output vector $\bar{\mathbf{w}}$ satisfies*

$$f(\bar{\mathbf{w}}) - f(\mathbf{w}^\star) \;\leq\; \frac{B\,\rho}{\sqrt{T}}.$$

Furthermore, for every $\epsilon > 0$, to achieve $f(\bar{\mathbf{w}}) - f(\mathbf{w}^\star) \leq \epsilon$, it suffices to run the GD algorithm for a number of iterations that satisfies

$$T \geq \frac{B^2\rho^2}{\epsilon^2}.$$

14.2 Subgradients

GD算法要求函数f可微。我们现在将讨论推广到可微函数之外。我们将展示GD算法可以通过使用所谓的$f(\mathbf{w})$在$\mathbf{w}^{(t)}$处的子梯度，而不是梯度，应用于不可微函数。

为了激励子梯度定义，回忆一下，对于一个凸函数 f，其在 $\mathbf{w}$ 处的梯度定义了位于 f 之下的切线的斜率，即，

$$\forall \mathbf{u}, \quad f(\mathbf{u}) \geq f(\mathbf{w}) + \langle \mathbf{u} - \mathbf{w}, \nabla f(\mathbf{w}) \rangle. \tag{14.7}$$

图14.2的左侧给出了一个插图。

存在一个位于 f 之下的切线是凸函数的一个重要性质，这实际上是对凸性的另一种描述。

LEMMA 14.3 *Let S be an open convex set. A function $f : S \to \mathbb{R}$ is convex iff for every $\mathbf{w} \in S$ there exists $\mathbf{v}$ such that*

$$\forall \mathbf{u} \in S, \quad f(\mathbf{u}) \geq f(\mathbf{w}) + \langle \mathbf{u} - \mathbf{w}, \mathbf{v} \rangle. \tag{14.8}$$

这个引理的证明可以在许多凸分析教科书中找到（例如，（Borwein & Lewis 2006））。前面的不等式引导我们到子梯度的定义。

DEFINITION 14.4（子梯度）满足方程（14.8）的向量 $\mathbf{v}$ 被称为 f 在 $\mathbf{w}$ 处的 *subgradient*。f 在 $\mathbf{w}$ 处的子梯度集被称为 *differential set*，表示为 $\partial f(\mathbf{w})$。

右侧图14.2给出了子梯度的示意图。对于标量函数，凸函数 f 在 w 处的子梯度是接触 f 于 w 且在其他地方不高于 f 的直线的斜率。

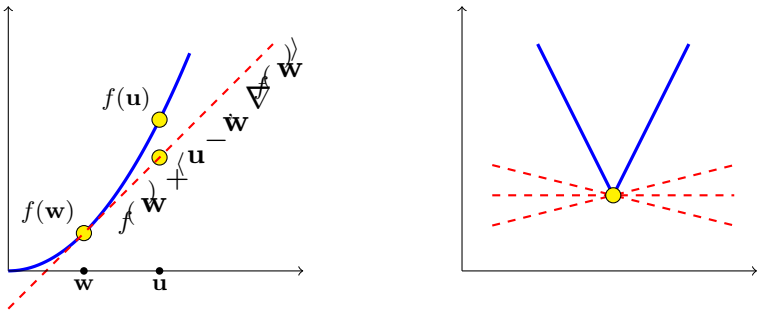


Figure 14.2 左侧：方程（14.7）的右侧是 f 在 $\mathbf{w}$ 处的切线。对于凸函数，切线下界 f。右侧：非可微凸函数几个子梯度的示意图。

14.2.1 Calculating Subgradients

如何构造给定凸函数的子梯度？如果一个函数在点 $\mathbf{w}$ 处可微，那么微分集是平凡的，如下所述。

CLAIM 14.5 *If f is differentiable at $\mathbf{w}$ then $\partial f(\mathbf{w})$ contains a single element – the gradient of f at $\mathbf{w}$, $\nabla f(\mathbf{w})$.*

Example 14.1 (绝对函数的微分集 考虑绝对值函数 $f(x) = |x|$。使用命题14.5，我们可以轻松构造 f 的可微部分的微分集，唯一需要特别注意的点为 $x_0 = 0$。在该点，很容易验证子微分是介于 -1 和 1 之间的所有数的集合。因此：

$$\partial f(x) = \begin{cases} \{1\} & \text{if } x > 0 \\ \{-1\} & \text{if } x < 0 \\ [-1, 1] & \text{if } x = 0 \end{cases}$$

对于许多实际应用，我们不需要在给定点计算整个子梯度集，因为该集合中的一个成员就足够了。以下声明展示了如何构造点值最大函数的子梯度。

CLAIM 14.6 *Let $g(\mathbf{w}) = \max_{i \in [r]} g_i(\mathbf{w})$ for r convex differentiable functions $g_1, \ldots, g_r$. Given some $\mathbf{w}$, let $j \in \operatorname{argmax}_i g_i(\mathbf{w})$. Then $\nabla g_j(\mathbf{w}) \in \partial g(\mathbf{w})$.*

Proof 由于 g_j 是凸的，因此对于所有 $\mathbf{u}$，我们有

$$g_j(\mathbf{u}) \geq g_j(\mathbf{w}) + \langle \mathbf{u} - \mathbf{w}, \nabla g_j(\mathbf{w}) \rangle.$$

自 $g(\mathbf{w}) = g_j(\mathbf{w})$ 和 $g(\mathbf{u}) \geq g_j(\mathbf{u})$ 以来，我们得到如下

$$g(\mathbf{u}) \geq g(\mathbf{w}) + \langle \mathbf{u} - \mathbf{w}, \nabla g_j(\mathbf{w}) \rangle,$$

这总结了我们的证明。 □

Example 14.2 (一个Hinge损失的子梯度)回忆第12.3节中的hinge损失函数$f(\mathbf{w}) = \max\{0, 1 - y\langle \mathbf{w}, \mathbf{x}\rangle\}$ 对于某个向量$\mathbf{x}$和标量y。为了计算某个$\mathbf{w}$处的hinge损失的子梯度，我们依赖于前面的断言，并得到以下定义的向量$\mathbf{v}$是$\mathbf{w}$处的hinge损失的子梯度：

$$\mathbf{v} = \begin{cases} \mathbf{0} & \text{if } 1 - y\langle \mathbf{w},\ \mathbf{x}\rangle \leq 0 \\ -y\mathbf{x} & \text{if } 1 - y\langle \mathbf{w},\ \mathbf{x}\rangle > 0 \end{cases}$$

14.2.2 Subgradients of Lipschitz Functions

回忆一下，一个函数 $f\colon A \to \mathbb{R}$ 是 ρ-Lipschitz 如果对于所有 $\mathbf{u}, \mathbf{v} \in A$

$$|f(\mathbf{u}) - f(\mathbf{v})| \leq \rho \|\mathbf{u} - \mathbf{v}\|.$$

以下引理给出了使用子梯度范数的等价定义。

LEMMA 14.7 *Let A be a convex open set and let $f\colon A \to \mathbb{R}$ be a convex function. Then, f is ρ-Lipschitz over A iff for all $\mathbf{w} \in A$ and $\mathbf{v} \in \partial f(\mathbf{w})$ we have that $\|\mathbf{v}\| \leq \rho$.*

Proof 假设对于所有 $\mathbf{v} \in \partial f(\mathbf{w})$，我们都有 $\|\mathbf{v}\| \leq \rho$。自从 $\mathbf{v} \in \partial f(\mathbf{w})$ 我们有

$$f(\mathbf{w}) - f(\mathbf{u}) \leq \langle \mathbf{v}, \mathbf{w} - \mathbf{u}\rangle.$$

使用柯西-施瓦茨不等式对右侧进行界定，我们得到

$$f(\mathbf{w}) - f(\mathbf{u}) \ \leq\ \langle \mathbf{v},\ \mathbf{w} - \mathbf{u}\rangle \leq \|\mathbf{v}\|\, \|\mathbf{w} - \mathbf{u}\| \leq \rho \|\mathbf{w} - \mathbf{u}\|.$$

一个类似的论据可以表明 $f(\mathbf{u}) - f(\mathbf{w}) \leq \rho \|\mathbf{w} - \mathbf{u}\|$。因此 f 是 ρ-Lipschitz。

现在假设 f 是 ρ-Lipschitz。选择一些 $\mathbf{w} \in A, \mathbf{v} \in \partial f(\mathbf{w})$。由于 A 是开集，存在 $\epsilon > 0$ 使得 $\mathbf{u} = \mathbf{w} + \epsilon \mathbf{v}/\|\mathbf{v}\|$ 属于 A。因此，$\langle \mathbf{u} - \mathbf{w}, \mathbf{v}\rangle = \epsilon \|\mathbf{v}\|$ 和 $\|\mathbf{u} - \mathbf{w}\| = \epsilon$。从子梯度的定义，

$$f(\mathbf{u}) - f(\mathbf{w}) \geq \langle \mathbf{v}, \mathbf{u} - \mathbf{w}\rangle = \epsilon \|\mathbf{v}\|.$$

另一方面，从 f 的Lipschitz性质我们有

$$\rho\, \epsilon = \rho \|\mathbf{u} - \mathbf{w}\| \geq f(\mathbf{u}) - f(\mathbf{w}).$$

结合这两个不等式，我们得出结论 $\|\mathbf{v}\| \leq \rho$。 □

14.2.3 Subgradient Descent

梯度下降算法可以通过使用 $f(\mathbf{w})$ 在 $\mathbf{w}^{(t)}$ 处的子梯度来推广到不可微函数，而不是梯度。收敛速度的分析保持不变：只需注意，方程（14.3）对子梯度同样成立。

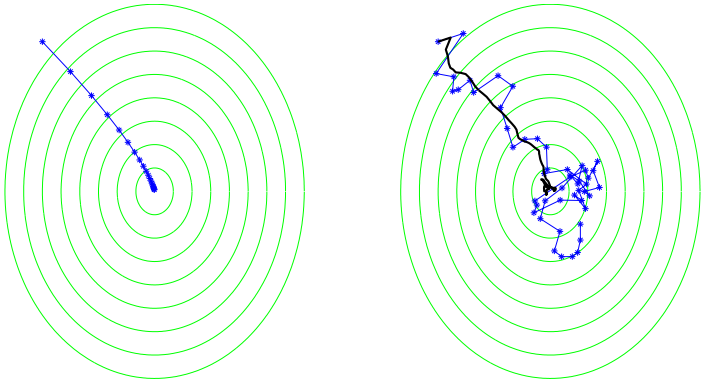


Figure 14.3 一个梯度下降算法（左侧）和随机梯度下降算法（右侧）的示意图。要最小化的函数是$1.25(x+6)^2+(y-8)^2$。对于随机情况，黑色线条表示$\mathbf{w}$的平均值。

14.3 Stochastic Gradient Descent (SGD)

在随机梯度下降中，我们不需要更新方向完全基于梯度。相反，我们允许方向是一个随机向量，并且只要求其在每次迭代的 *expected value* 将等于梯度方向。或者，更一般地说，我们要求随机向量的期望值将是当前向量处函数的子梯度。

Stochastic Gradient Descent (SGD) for minimizing $f(\mathbf{w})$

parameters: Scalar $\eta > 0$, integer $T > 0$
initialize: $\mathbf{w}^{(1)} = \mathbf{0}$
for $t = 1, 2, \ldots, T$
  choose $\mathbf{v}_t$ at random from a distribution such that $\mathbb{E}[\mathbf{v}_t \mid \mathbf{w}^{(t)}] \in \partial f(\mathbf{w}^{(t)})$
  update $\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta \mathbf{v}_t$
output $\bar{\mathbf{w}} = \frac{1}{T}\sum_{t=1}^{T} \mathbf{w}^{(t)}$

图14.3给出了随机梯度下降与梯度下降的示意图。正如我们在第14.5节中将要看到的，在学习问题背景下，很容易找到一个期望是风险函数子梯度的随机向量。

14.3.1 Analysis of SGD for Convex-Lipschitz-Bounded Functions

回忆我们在引理14.2中得到的GD算法的界限。对于随机情况，其中只有$\mathbf{v}_t$的期望值在$\partial f(\mathbf{w}^{(t)})$中，我们无法直接应用方程（14.3）。然而，由于$\mathbf{v}_t$的期望值是一个

子梯度 f 在 $\mathbf{w}^{(t)}$ 处，我们仍然可以推导出对随机梯度下降的 *expected* 输出的类似界限。这在下述定理中得到形式化。

THEOREM 14.8 *Let $B, \rho > 0$. Let f be a convex function and let $\mathbf{w}^\star \in \operatorname{argmin}_{\mathbf{w}:\|\mathbf{w}\|\leq B} f(\mathbf{w})$. Assume that SGD is run for T iterations with $\eta = \sqrt{\frac{B^2}{\rho^2 T}}$. Assume also that for all t, $\|\mathbf{v}_t\| \leq \rho$ with probability 1. Then,*

$$\mathbb{E}\left[f(\bar{\mathbf{w}})\right] - f(\mathbf{w}^\star) \leq \frac{B\,\rho}{\sqrt{T}}.$$

Therefore, for any $\epsilon > 0$, to achieve $\mathbb{E}[f(\bar{\mathbf{w}})] - f(\mathbf{w}^\star) \leq \epsilon$, it suffices to run the SGD algorithm for a number of iterations that satisfies

$$T \geq \frac{B^2\rho^2}{\epsilon^2}.$$

Proof 让我们引入记号 $\mathbf{v}_{1:t}$ 来表示序列 $\mathbf{v}_1, \ldots, \mathbf{v}_t$。对方程（14.2）取期望，我们得到

$$\mathop{\mathbb{E}}_{\mathbf{v}_{1:T}}\left[f(\bar{\mathbf{w}}) - f(\mathbf{w}^\star)\right] \leq \mathop{\mathbb{E}}_{\mathbf{v}_{1:T}}\left[\frac{1}{T}\sum_{t=1}^{T}(f(\mathbf{w}^{(t)}) - f(\mathbf{w}^\star))\right].$$

由于引理14.1对任何序列$\mathbf{v}_1, \mathbf{v}_2, \ldots \mathbf{v}_T$都成立，它也适用于SGD。通过对引理中的界取期望，我们得到

$$\mathop{\mathbb{E}}_{\mathbf{v}_{1:T}}\left[\frac{1}{T}\sum_{t=1}^{T}\langle \mathbf{w}^{(t)} - \mathbf{w}^\star, \mathbf{v}_t\rangle\right] \leq \frac{B\,\rho}{\sqrt{T}}. \tag{14.9}$$

它留给证明

$$\mathop{\mathbb{E}}_{\mathbf{v}_{1:T}}\left[\tfrac{1}{T}\sum_{t=1}^{T}(f(\mathbf{w}^{(t)}) - f(\mathbf{w}^\star))\right] \leq \mathop{\mathbb{E}}_{\mathbf{v}_{1:T}}\left[\tfrac{1}{T}\sum_{t=1}^{T}\langle \mathbf{w}^{(t)} - \mathbf{w}^\star, \mathbf{v}_t\rangle\right], \tag{14.10}$$

此处我们将予以证明。

使用期望的线性，我们有

$$\mathop{\mathbb{E}}_{\mathbf{v}_{1:T}}\left[\frac{1}{T}\sum_{t=1}^{T}\langle \mathbf{w}^{(t)} - \mathbf{w}^\star, \mathbf{v}_t\rangle\right] = \frac{1}{T}\sum_{t=1}^{T}\mathop{\mathbb{E}}_{\mathbf{v}_{1:T}}\left[\langle \mathbf{w}^{(t)} - \mathbf{w}^\star, \mathbf{v}_t\rangle\right].$$

接下来，我们回忆 *law of total expectation*：对于任意两个随机变量 α, β 和一个函数 g，$\mathbb{E}_\alpha[g(\alpha)] = \mathbb{E}_\beta\,\mathbb{E}_\alpha[g(\alpha)|\beta]$. 设置 $\alpha = \mathbf{v}_{1:t}$ 和 $\beta = \mathbf{v}_{1:t-1}$，我们得到

$$\begin{aligned}\mathop{\mathbb{E}}_{\mathbf{v}_{1:T}}\left[\langle \mathbf{w}^{(t)} - \mathbf{w}^\star, \mathbf{v}_t\rangle\right] &= \mathop{\mathbb{E}}_{\mathbf{v}_{1:t}}\left[\langle \mathbf{w}^{(t)} - \mathbf{w}^\star, \mathbf{v}_t\rangle\right] \\ &= \mathop{\mathbb{E}}_{\mathbf{v}_{1:t-1}}\mathop{\mathbb{E}}_{\mathbf{v}_{1:t}}\left[\langle \mathbf{w}^{(t)} - \mathbf{w}^\star, \mathbf{v}_t\rangle \,|\, \mathbf{v}_{1:t-1}\right].\end{aligned}$$

一旦我们知道 $\mathbf{v}_{1:t-1}$，$\mathbf{w}^{(t)}$ 的值就不再是随机的，因此

$$\mathop{\mathbb{E}}_{\mathbf{v}_{1:t-1}}\mathop{\mathbb{E}}_{\mathbf{v}_{1:t}}\left[\langle \mathbf{w}^{(t)} - \mathbf{w}^\star, \mathbf{v}_t\rangle \,|\, \mathbf{v}_{1:t-1}\right] = \mathop{\mathbb{E}}_{\mathbf{v}_{1:t-1}}\left\langle \mathbf{w}^{(t)} - \mathbf{w}^\star, \mathop{\mathbb{E}}_{\mathbf{v}_t}[\mathbf{v}_t \,|\, \mathbf{v}_{1:t-1}]\right\rangle.$$

因此，$\mathbf{w}^{(t)}$仅依赖于$\mathbf{v}_{1:t-1}$，SGD要求$\mathbb{E}_{\mathbf{v}_t}[\mathbf{v}_t \mid \mathbf{w}^{(t)}] \in \partial f(\mathbf{w}^{(t)})$，我们得到$\mathbb{E}_{\mathbf{v}_t}[\mathbf{v}_t \mid \mathbf{v}_{1:t-1}] \in \partial f(\mathbf{w}^{(t)})$。因此，

$$\mathop{\mathbb{E}}_{\mathbf{v}_{1:\,t-1}} \langle \mathbf{w}^{(t)} - \mathbf{w}^\star, \mathop{\mathbb{E}}_{\mathbf{v}_t}[\mathbf{v}_t \mid \mathbf{v}_{1:\,t-1}] \rangle \;\geq\; \mathop{\mathbb{E}}_{\mathbf{v}_{1:\,t-1}} [f(\mathbf{w}^{(t)}) - f(\mathbf{w}^\star)].$$

总体而言，我们已经证明

$$\begin{aligned}\mathop{\mathbb{E}}_{\mathbf{v}_{1:T}} [\langle \mathbf{w}^{(t)} - \mathbf{w}^\star, \mathbf{v}_t \rangle] &\geq \mathop{\mathbb{E}}_{\mathbf{v}_{1:t-1}} [f(\mathbf{w}^{(t)}) - f(\mathbf{w}^\star)] \\ &= \mathop{\mathbb{E}}_{\mathbf{v}_{1:T}} [f(\mathbf{w}^{(t)}) - f(\mathbf{w}^\star)] \,.\end{aligned}$$

对 t 求和，除以 T，并利用期望的线性，我们得到方程（14.10）成立，从而得出我们的证明。 □

14.4 Variants

本节中，我们描述了随机梯度下降的几种变体。

14.4.1 Adding a Projection Step

在GD和SGD算法的前期分析中，我们要求$\mathbf{w}^\star$的范数不超过B，这相当于要求$\mathbf{w}^\star$属于集合$\mathcal{H} = \{\mathbf{w}\colon \|\mathbf{w}\| \leq B\}$。从学习的角度来看，这意味着我们限制自己在一个$B$有界的假设类中。然而，我们向梯度（或其期望方向）相反的方向迈出的任何一步都可能使我们超出这个界限，甚至无法保证$\bar{\mathbf{w}}$满足它。在以下内容中，我们展示了如何在保持相同收敛速度的同时克服这个问题。

基本思路是添加一个 *projection step*；也就是说，我们现在将有一个两步更新规则，首先从 $\mathbf{w}$ 的当前值中减去一个子梯度，然后将得到的向量投影到 $\mathcal{H}$ 上。形式上，

1.. $\mathbf{w}^{(t+\frac{1}{2})} = \mathbf{w}^{(t)} - \eta \mathbf{v}_t$
2.. $\mathbf{w}^{(t+1)} = \operatorname{argmin}_{\mathbf{w} \in \mathcal{H}} \|\mathbf{w} - \mathbf{w}^{(t+\frac{1}{2})}\|$

步骤将当前值 $\mathbf{w}$ 替换为 $\mathcal{H}$ 中最接近它的向量。

显然，投影步骤保证了对于所有 t，有 $\mathbf{w}^{(t)} \in \mathcal{H}$。由于 $\mathcal{H}$ 是凸函数，这也意味着 $\bar{\mathbf{w}} \in \mathcal{H}$ 如所需。接下来，我们证明带有投影的 SGD 的分析保持不变。这是基于以下引理。

LEMMA 14.9 (投影引理) *Let $\mathcal{H}$ be a closed convex set and let $\mathbf{v}$ be the projection of $\mathbf{w}$ onto $\mathcal{H}$, namely,*

$$\mathbf{v} = \operatorname*{argmin}_{\mathbf{x} \in \mathcal{H}} \|\mathbf{x} - \mathbf{w}\|^2.$$

Then, for every $\mathbf{u} \in \mathcal{H}$,

$$\|\mathbf{w} - \mathbf{u}\|^2 - \|\mathbf{v} - \mathbf{u}\|^2 \geq 0.$$

Proof 通过 $\mathcal{H}$ 的凸性，对于每个 $\alpha \in (0,1)$，我们有 $\mathbf{v}+\alpha(\mathbf{u}-\mathbf{v}) \in \mathcal{H}$。因此，从 $\mathbf{v}$ 的最优性中，我们得到

$$\begin{aligned}\|\mathbf{v} - \mathbf{w}\|^2 &\leq \|\mathbf{v} + \alpha(\mathbf{u} - \mathbf{v}) - \mathbf{w}\|^2 \\ &= \|\mathbf{v} - \mathbf{w}\|^2 + 2\alpha\langle \mathbf{v} - \mathbf{w}, \mathbf{u} - \mathbf{v}\rangle + \alpha^2\|\mathbf{u} - \mathbf{v}\|^2.\end{aligned}$$

重新排列，我们得到

$$2\langle \mathbf{v} - \mathbf{w}, \mathbf{u} - \mathbf{v}\rangle \geq -\alpha \|\mathbf{u} - \mathbf{v}\|^2.$$

取极限 $\alpha \to 0$，我们得到

$$\langle \mathbf{v} - \mathbf{w}, \mathbf{u} - \mathbf{v}\rangle \geq 0.$$

因此，

$$\begin{aligned}\|\mathbf{w} - \mathbf{u}\|^2 &= \|\mathbf{w} - \mathbf{v} + \mathbf{v} - \mathbf{u}\|^2 \\ &= \|\mathbf{w} - \mathbf{v}\|^2 + \|\mathbf{v} - \mathbf{u}\|^2 + 2\langle \mathbf{v} - \mathbf{w},\ \mathbf{u} - \mathbf{v}\rangle \\ &\geq \|\mathbf{v} - \mathbf{u}\|^2.\end{aligned}$$

□

配备前面的引理，我们可以轻松地将SGD的分析适应到我们在一个封闭和凸集上添加投影步骤的情况。只需注意，对于每个 t，

$$\begin{aligned}&\|\mathbf{w}^{(t+1)} - \mathbf{w}^\star\|^2 - \|\mathbf{w}^{(t)} - \mathbf{w}^\star\|^2 \\ &\quad= \|\mathbf{w}^{(t+1)} - \mathbf{w}^\star\|^2 - \|\mathbf{w}^{(t+\frac{1}{2})} - \mathbf{w}^\star\|^2 + \|\mathbf{w}^{(t+\frac{1}{2})} - \mathbf{w}^\star\|^2 - \|\mathbf{w}^{(t)} - \mathbf{w}^\star\|^2 \\ &\quad\leq \|\mathbf{w}^{(t+\frac{1}{2})} - \mathbf{w}^\star\|^2 - \|\mathbf{w}^{(t)} - \mathbf{w}^\star\|^2.\end{aligned}$$

因此，当我们添加投影步骤时，引理14.1成立，因此其余的分析直接得出。

14.4.2 Variable Step Size

另一种SGD变体是按t函数减小步长。也就是说，我们不是用常数η来更新，而是使用η_t。例如，我们可以设置$\eta_t = \frac{B}{\rho\sqrt{t}}$并达到类似于定理14.8的界限。这种想法是，当我们接近函数的最小值时，我们更小心地迈步，以免“超过”最小值。

14.4.3 Other Averaging Techniques

我们已将输出向量设置为$\bar{\mathbf{w}} = \frac{1}{T}\sum_{t=1}^{T}\mathbf{w}^{(t)}$。有其他方法，例如对于某些随机$t \in [t]$输出$\mathbf{w}^{(t)}$，或者输出过去$\alpha T$次迭代的$\mathbf{w}^{(t)}$的平均值，对于某些$\alpha \in (0,1)$。还可以对最后几次迭代取加权平均值。这些更复杂的平均方案在某些情况下可以提高收敛速度，例如在以下定义的强凸函数的情况下。

14.4.4 Strongly Convex Functions*

在这一节中，我们展示了SGD的一个变体，该变体在目标函数强凸的问题上具有更快的收敛速度（参见前一章中关于强凸性的定义13.4）。我们依赖于以下命题，它推广了引理13.5。

CLAIM 14.10 *If f is λ-strongly convex then for every $\mathbf{w}, \mathbf{u}$ and $\mathbf{v} \in \partial f(\mathbf{w})$ we have*

$$\langle \mathbf{w} - \mathbf{u}, \mathbf{v} \rangle \;\geq\; f(\mathbf{w}) - f(\mathbf{u}) + \tfrac{\lambda}{2}\|\mathbf{w} - \mathbf{u}\|^2.$$

证明与引理13.5的证明类似，留作练习。

SGD for minimizing a λ-strongly convex function

Goal: 解决最小化$_{\mathbf{w}\in\mathcal{H}} f(\mathbf{w})$ $1, \ldots, T$选择一个随机向量
parameter: 使得$\mathbb{E}[\mathbf{v}_t|\mathbf{w}^{(t)}] \in \partial f(\mathbf{w}^{(t)})$ 设置 $\eta_t = 1/(\lambda\, t)$ 设置
initialize: $\mathbf{w}^{(1)} = \mathbf{0}$设置 $\mathbf{w}^{(t+1)} =$ 最小值
for $t =$ $_{\mathbf{w}\in\mathcal{H}} \|\mathbf{w} - \mathbf{w}^{(t+\frac{1}{2})}\|^2$ 的参数 **output:** $\bar{\mathbf{w}} = \frac{1}{T}\sum_{t=1}^{T}\mathbf{w}^{(t)}$

THEOREM 14.11 *Assume that f is λ-strongly convex and that $\mathbb{E}[\|\mathbf{v}_t\|^2] \leq \rho^2$. Let $\mathbf{w}^\star \in \operatorname{argmin}_{\mathbf{w}\in\mathcal{H}} f(\mathbf{w})$ be an optimal solution. Then,*

$$\mathbb{E}[f(\bar{\mathbf{w}})] - f(\mathbf{w}^\star) \leq \frac{\rho^2}{2\,\lambda\, T}(1 + \log(T)).$$

Proof Let {v*}[{v*}]. Since {v*} is strongly convex and {v*} is in the subgradient s et of {v*} at {v*} we have that

$$\langle \mathbf{w}^{(t)} - \mathbf{w}^\star, \nabla^{(t)} \rangle \;\geq\; f(\mathbf{w}^{(t)}) - f(\mathbf{w}^\star) + \tfrac{\lambda}{2}\|\mathbf{w}^{(t)} - \mathbf{w}^\star\|^2 \;. \tag{14.11}$$

接下来，我们证明

$$\langle \mathbf{w}^{(t)} - \mathbf{w}^\star, \nabla^{(t)} \rangle \;\leq\; \frac{\mathbb{E}[\|\mathbf{w}^{(t)} - \mathbf{w}^\star\|^2 - \|\mathbf{w}^{(t+1)} - \mathbf{w}^\star\|^2]}{2\,\eta_t} + \frac{\eta_t}{2}\rho^2. \tag{14.12}$$

由于 $\mathbf{w}^{(t+1)}$ 是 $\mathbf{w}^{(t+\frac{1}{2})}$ 在 $\mathcal{H}$ 上的投影，并且 $\mathbf{w}^\star \in \mathcal{H}$ 我们有 $\|\mathbf{w}^{(t+\frac{1}{2})} - \mathbf{w}^\star\|^2 \geq \|\mathbf{w}^{(t+1)} - \mathbf{w}^\star\|^2$。因此，

$$\begin{aligned}\|\mathbf{w}^{(t)} - \mathbf{w}^\star\|^2 - \|\mathbf{w}^{(t+1)} - \mathbf{w}^\star\|^2 &\geq \|\mathbf{w}^{(t)} - \mathbf{w}^\star\|^2 - \|\mathbf{w}^{(t+\frac{1}{2})} - \mathbf{w}^\star\|^2 \\ &= 2\eta_t \langle \mathbf{w}^{(t)} - \mathbf{w}^\star, \mathbf{v}_t \rangle - \eta_t^2 \|\mathbf{v}_t\|^2 \ .\end{aligned}$$

对两边取期望，重新排列，并使用假设 $\mathbb{E}[\|\mathbf{v}_t\|^2] \leq \rho^2$ 得到方程（14.12）。比较方程（14.11）和方程（14.12），并对 t 求和，我们得到

$$\begin{aligned}&\sum_{t=1}^{T} (\mathbb{E}[f(\mathbf{w}^{(t)})] - f(\mathbf{w}^\star)) \\ &\quad \leq \mathbb{E}\left[\sum_{t=1}^{T} \left(\frac{\|\mathbf{w}^{(t)} - \mathbf{w}^\star\|^2 - \|\mathbf{w}^{(t+1)} - \mathbf{w}^\star\|^2}{2\,\eta_t} - \tfrac{\lambda}{2}\|\mathbf{w}^{(t)} - \mathbf{w}^\star\|^2 \right)\right] + \frac{\rho^2}{2}\sum_{t=1}^{T} \eta_t.\end{aligned}$$

接下来，我们使用定义 $\eta_t = 1/(\lambda\, t)$ 并注意到等式右侧的第一个求和项简化为 $-\lambda T \|\mathbf{w}^{(T+1)} - \mathbf{w}^\star\|^2 \leq 0$。因此，

$$\sum_{t=1}^{T} (\mathbb{E}[f(\mathbf{w}^{(t)})] - f(\mathbf{w}^\star)) \leq \frac{\rho^2}{2\,\lambda} \sum_{t=1}^{T} \frac{1}{t} \leq \frac{\rho^2}{2\,\lambda}(1 + \log(T)).$$

该定理通过除以 T 并使用 Jensen 不等式从前面得出。

□

Remark 14.3 Rakhlin、Shamir与Sridharan（2012）推导出一个收敛率，其中消除了算法变体中T的对数项，该变体输出最后$T/2$次迭代的平均值，即 $\bar{\mathbf{w}} = \frac{2}{T}\sum_{t=T/2+1}^{T} \mathbf{w}^{(t)}$。Shamir与张（2013）已经证明，即使我们输出$\bar{\mathbf{w}} = \mathbf{w}^{(T)}$，定理14.11仍然成立。

14.5 Learning with SGD

我们迄今为止介绍了并分析了通用凸函数的SGD算法。现在我们将考虑其在学习任务中的应用。

14.5.1 SGD for Risk Minimization

回忆在学习中我们面临最小化风险函数的问题

$$L_{\mathcal{D}}(\mathbf{w}) = \mathop{\mathbb{E}}_{z\sim\mathcal{D}}[\ell(\mathbf{w}, z)].$$

我们看到了经验风险最小化的方法，其中我们最小化经验风险 $L_S(\mathbf{w})$，作为最小化 $L_{\mathcal{D}}(\mathbf{w})$ 的估计。SGD 允许我们采取不同的方法并直接最小化 $L_{\mathcal{D}}(\mathbf{w})$。由于我们不知道 $\mathcal{D}$，我们无法简单地计算 $\nabla L_{\mathcal{D}}(\mathbf{w}^{(t)})$ 并用 GD 方法最小化它。然而，使用 SGD，我们只需要找到梯度的无偏估计。

$L_{\mathcal{D}}(\mathbf{w})$，而 $\nabla L_{\mathcal{D}}(\mathbf{w}^{(t)})$

为了简单起见，我们首先考虑可微损失函数的情况。因此，风险函数 $L_{\mathcal{D}}$ 也是可微的。随机向量 $\mathbf{v}_t$ 的构造如下：首先，采样 $z \sim \mathcal{D}$。然后，定义 $\mathbf{v}_t$ 为函数 $\ell(\mathbf{w}, z)$ 关于 $\mathbf{w}$ 在点 $\mathbf{w}^{(t)}$ 处的梯度。然后，根据梯度的线性性质，我们有

$$\mathbb{E}[\mathbf{v}_t \mid \mathbf{w}^{(t)}] = \mathbb{E}_{z \sim \mathcal{D}}[\nabla \ell(\mathbf{w}^{(t)}, z)] = \nabla \mathbb{E}_{z \sim \mathcal{D}}[\ell(\mathbf{w}^{(t)}, z)] = \nabla L_{\mathcal{D}}(\mathbf{w}^{(t)}) \tag{14.3}$$

损失函数 $\ell(\mathbf{w}, z)$ 在 $\mathbf{w}^{(t)}$ 处的梯度因此是无偏估计的损失函数 $L_{\mathcal{D}}(\mathbf{w}^{(t)})$ 的梯度，并且可以通过在每个迭代 t 中采样一个单个的新示例 $z \sim \mathcal{D}$ 来轻松构建。

相同的论点适用于不可微的损失函数。我们只需让 $\mathbf{v}_t$ 成为 $\ell(\mathbf{w}, z)$ 在 $\mathbf{w}^{(t)}$ 处的一个子梯度。然后，对于每个 $\mathbf{u}$，我们有

$$\ell(\mathbf{u}, z) - \ell(\mathbf{w}^{(t)}, z) \geq \langle \mathbf{u} - \mathbf{w}^{(t)}, \mathbf{v}_t \rangle$$

对 $z \sim \mathcal{D}$ 进行期望取值，并在 $\mathbf{w}^{(t)}$ 的值下条件化，我们得到

$$\begin{aligned} L_{\mathcal{D}}(\mathbf{u}) - L_{\mathcal{D}}(\mathbf{w}^{(t)}) &= \mathbb{E}[\ell(\mathbf{u}, z) - \ell(\mathbf{w}^{(t)}, z) \mid \mathbf{w}^{(t)}] \\ &\geq \mathbb{E}[\langle \mathbf{u} - \mathbf{w}^{(t)}, \mathbf{v}_t \rangle \mid \mathbf{w}^{(t)}] \\ &= \langle \mathbf{u} - \mathbf{w}^{(t)}, \mathbb{E}[\mathbf{v}_t \mid \mathbf{w}^{(t)}] \rangle \end{aligned}$$

因此，$\mathbb{E}[\mathbf{v}_t \mid \mathbf{w}^{(t)}]$ 是 $L_{\mathcal{D}}(\mathbf{w})$ 在 $\mathbf{w}^{(t)}$ 处的子梯度。

为了总结，最小化风险的随机梯度下降框架如下。

Stochastic Gradient Descent (SGD) for minimizing $L_{\mathcal{D}}(\mathbf{w})$

parameters: 标量 $\eta > 0$，整数 $T > 0$

initialize: $\mathbf{w}^{(1)} = \mathbf{0}$

for $t = 1, 2, \ldots, T$

从 $\mathcal{D}$ 采样 z

选择 $\mathbf{v}_t \in \partial \ell(\mathbf{w}^{(t)}, z)$

更新 $\mathbf{w}^{(t+\frac{1}{2})} = \mathbf{w}^{(t)} - \eta \mathbf{v}_t$

output $\bar{\mathbf{w}} = \frac{1}{T} \sum_{t=1}^{T} \mathbf{w}^{(t)}$

我们现在将使用我们对SGD的分析来获得学习Lipschitz有界问题的样本复杂度分析。定理14.8得出以下结论：

COROLLARY 14.12 *Consider a convex-Lipschitz-bounded learning problem with parameters ρ, B. Then, for every $\epsilon > 0$, if we run the SGD method for minimizing*

$L_{\mathcal{D}}(\mathbf{w})$ with a number of iterations (i.e., number of examples)

$$T \geq \frac{B^2 \rho^2}{\epsilon^2}$$

and with $\eta = \sqrt{\frac{B^2}{T \rho^2}}$ then the output of SGD satisfies

$$\mathbb{E}\left[L_{\mathcal{D}}(\bar{\mathbf{w}})\right] \leq \min_{\mathbf{w} \in \mathcal{H}} L_{\mathcal{D}}(\mathbf{w}) + \epsilon.$$

有趣的是，所需的样本复杂度与我们所推导出的正则化损失最小化样本复杂度具有相同的数量级。事实上，SGD的样本复杂度甚至比我们所推导出的正则化损失最小化样本复杂度更好，高出8倍。

14.5.2 Analyzing SGD for Convex-Smooth Learning Problems

在前一章中，我们看到了正则化损失最小化规则也学习了凸平滑有界学习问题类别。现在我们展示SGD算法也可以用于此类问题。

THEOREM 14.13 *Assume that for all z, the loss function $\ell(\cdot, z)$ is convex, β-smooth, and nonnegative. Then, if we run the SGD algorithm for minimizing $L_{\mathcal{D}}(\mathbf{w})$ we have that for every $\mathbf{w}^\star$,*

$$\mathbb{E}[L_{\mathcal{D}}(\bar{\mathbf{w}})] \leq \frac{1}{1 - \eta\beta} \left(L_{\mathcal{D}}(\mathbf{w}^\star) + \frac{\|\mathbf{w}^\star\|^2}{2\eta T} \right).$$

Proof 回忆一下，如果一个函数是 β-平滑且非负的，那么它是自界的:

$$\|\nabla f(\mathbf{w})\|^2 \leq 2\beta f(\mathbf{w}).$$

为了分析凸平滑问题中的SGD，让我们定义$z_1, \ldots, z_T$为SGD算法的随机样本，令$f_t(\cdot) = \ell(\cdot, z_t)$，并注意$\mathbf{v}_t = \nabla f_t(\mathbf{w}^{(t)})$。对于所有$t$，$f_t$是一个凸函数，因此$f_t(\mathbf{w}^{(t)}) - f_t(\mathbf{w}^\star) \leq \langle \mathbf{v}_t, \mathbf{w}^{(t)} - \mathbf{w}^\star \rangle$。对$t$求和并使用引理14.1，我们得到

$$\sum_{t=1}^{T} (f_t(\mathbf{w}^{(t)}) - f_t(\mathbf{w}^\star)) \leq \sum_{t=1}^{T} \langle \mathbf{v}_t, \mathbf{w}^{(t)} - \mathbf{w}^\star \rangle \leq \frac{\|\mathbf{w}^\star\|^2}{2\eta} + \frac{\eta}{2} \sum_{t=1}^{T} \|\mathbf{v}_t\|^2.$$

将前面的内容与f_t的自约束性相结合，得到

$$\sum_{t=1}^{T} (f_t(\mathbf{w}^{(t)}) - f_t(\mathbf{w}^\star)) \leq \frac{\|\mathbf{w}^\star\|^2}{2\eta} + \eta\beta \sum_{t=1}^{T} f_t(\mathbf{w}^{(t)}).$$

除以T并重新排列，我们得到

$$\frac{1}{T} \sum_{t=1}^{T} f_t(\mathbf{w}^{(t)}) \leq \frac{1}{1 - \eta\beta} \left(\frac{1}{T} \sum_{t=1}^{T} f_t(\mathbf{w}^\star) + \frac{\|\mathbf{w}^\star\|^2}{2\eta T} \right).$$

接下来，我们对方程两边的期望值进行求导

到 $z_1, \ldots, z_T$。显然，$\mathbb{E}[f_t(\mathbf{w}^\star)] = L_{\mathcal{D}}(\mathbf{w}^\star)$。此外，使用与定理14.8证明中相同的论据，我们有

$$\mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T} f_t(\mathbf{w}^{(t)})\right] = \mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T} L_{\mathcal{D}}(\mathbf{w}^{(t)})\right] \geq \mathbb{E}[L_{\mathcal{D}}(\bar{\mathbf{w}})].$$

所有结合，我们得出我们的证明。 □

作为直接推论，我们得到：

COROLLARY 14.14 *Consider a convex-smooth-bounded learning problem with parameters β, B. Assume in addition that $\ell(\mathbf{0}, z) \leq 1$ for all $z \in Z$. For every $\epsilon > 0$, set $\eta = \frac{1}{\beta(1+3/\epsilon)}$. Then, running SGD with $T \geq 12B^2\beta/\epsilon^2$ yields*

$$\mathbb{E}[L_{\mathcal{D}}(\bar{\mathbf{w}})] \leq \min_{\mathbf{w}\in\mathcal{H}} L_{\mathcal{D}}(\mathbf{w}) + \epsilon.$$

14.5.3 SGD for Regularized Loss Minimization

我们已经表明，SGD 具有与正则化损失最小化相同的最大样本复杂度界限。然而，在某些分布上，正则化损失最小化可能得到更好的解。因此，在某些情况下，我们可能希望解决与正则化损失最小化相关的优化问题，即，[1]

$$\min_{\mathbf{w}} \left(\frac{\lambda}{2}\|\mathbf{w}\|^2 + L_S(\mathbf{w})\right). \tag{14.14}$$

由于我们处理的是损失函数为凸的凸学习问题，前述问题也是一个可以使用SGD解决的凸优化问题，正如我们将在本节中看到的那样。

定义 $f(\mathbf{w}) = \frac{\lambda}{2}\|\mathbf{w}\|^2 + L_S(\mathbf{w})$。注意 f 是一个 λ-强凸函数；因此，我们可以应用第14.4.4节中给出的 SGD 变体（带有 $\mathcal{H} = \mathbb{R}^d$）。为了应用此算法，我们只需要找到一种方法来构造 f 在 $\mathbf{w}^{(t)}$ 处的一个无偏估计的子梯度。这很容易做到，只需注意，如果我们从 S 中随机均匀地选择 z，然后选择 $\mathbf{v}_t \in \partial\ell(\mathbf{w}^{(t)}, z)$，那么 $\lambda\mathbf{w}^{(t)} + \mathbf{v}_t$ 的期望值就是 f 在 $\mathbf{w}^{(t)}$ 处的一个子梯度。

为了分析得到的算法，我们首先重写更新规则（假设

[1] We divided λ by 2 for convenience.

那 $\mathcal{H} = \mathbb{R}^d$ 因此投影步骤不重要) 如下

$$\begin{aligned}
\mathbf{w}^{(t+1)} &= \mathbf{w}^{(t)} - \frac{1}{\lambda t}\left(\lambda \mathbf{w}^{(t)} + \mathbf{v}_t\right) \\
&= \left(1 - \frac{1}{t}\right)\mathbf{w}^{(t)} - \frac{1}{\lambda t}\mathbf{v}_t \\
&= \frac{t-1}{t}\mathbf{w}^{(t)} - \frac{1}{\lambda t}\mathbf{v}_t \\
&= \frac{t-1}{t}\left(\frac{t-2}{t-1}\mathbf{w}^{(t-1)} - \frac{1}{\lambda(t-1)}\mathbf{v}_{t-1}\right) - \frac{1}{\lambda t}\mathbf{v}_t \\
&= -\frac{1}{\lambda t}\sum_{i=1}^{t}\mathbf{v}_i. \qquad (14.15)
\end{aligned}$$

如果我们假设损失函数是 ρ-Lipschitz 连续的，那么对于所有的 t，我们有 $\|\mathbf{v}_t\| \leq \rho$，因此 $\|\lambda \mathbf{w}^{(t)}\| \leq \rho$，这导致

$$\|\lambda \mathbf{w}^{(t)} + \mathbf{v}_t\| \leq 2\rho.$$

定理14.11因此告诉我们，在执行T次迭代后，我们有

$$\mathbb{E}[f(\bar{\mathbf{w}})] - f(\mathbf{w}^\star) \leq \frac{4\rho^2}{\lambda T}(1 + \log(T)).$$

14.6 Summary

我们介绍了梯度下降和随机梯度下降算法，以及它们的几种变体。我们分析了它们的收敛速度，并计算了保证预期目标函数不超过ϵ加上最优目标函数所需的迭代次数。最重要的是，我们展示了通过使用SGD可以直接最小化风险函数。我们通过从$\mathcal{D}$中独立同分布地采样一个点，并使用当前假设的损失$\mathbf{w}^{(t)}$在该点的子梯度作为风险函数梯度（或子梯度）的无偏估计来实现这一点。这意味着迭代次数的上界也给出了样本复杂度的上界。最后，我们还展示了如何将SGD方法应用于正则化风险最小化问题。在未来的章节中，我们将展示这如何产生与正则化风险最小化相关的某些优化问题的极其简单的求解器。

14.7 Bibliographic Remarks

SGD起源于Robbins & Monro（1951）。它在大规模机器学习问题中特别有效。例如，参见（Murata 1998，Le Cun 2004，Zhang 2004，Bottou & Bousquet 2008，Shalev-Shwartz，Singer & Srebro 2007，Shalev-Shwartz & Srebro 2008）。在优化领域，它已被研究

在 *stochastic optimization* 的背景下。例如，参见 (Nemirovski & Yudin 1978, Nesterov & Nesterov 2004, Nesterov 2005, Nemirovski, Juditsky, Lan & Shapiro 2009, Shapiro, Dentcheva & Ruszczyński 2009)。

我们推导出的强凸函数的界归功于 Hazan, Agarwal & Kale (2007)。如前所述，Rakhlin 等人 (2012) 获得了改进的界。

14.8 Exercises

1. 证明命题14.10。*Hint:*扩展引理13.5的证明。
2. 证明推论14.14。3. **Perceptron as a subgradient descent algorithm:** 设 $S = ((\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)) \in (\mathbb{R}^d \times \{\pm 1\})^m$。假设存在 $\mathbf{w} \in \mathbb{R}^d$，使得对于每一个 $i \in [m]$，我们有 $y_i \langle \mathbf{w}, \mathbf{x}_i \rangle \geq 1$，并令 $\mathbf{w}^\star$ 为满足上述要求的所有向量中范数最小的向量。令 $R = \max_i \|\mathbf{x}_i\|$。定义一个函数

$$f(\mathbf{w}) \;=\; \max_{i \in [m]} \left(1 - y_i \left\langle \mathbf{w}, \mathbf{x}_i \right\rangle\right).$$

- 证明$\min_{\mathbf{w}:\|\mathbf{w}\| \leq \|\mathbf{w}^\star\|} f(\mathbf{w}) =\geqslant 0$，并证明对于任何满足$f(\mathbf{w})$ <=1的$\mathbf{w}$，它们将S中的示例分开。
- 展示如何计算 f 的子梯度。
- 描述并分析此情况下的子梯度下降算法。将算法和分析与第9.1.2节中给出的批量感知机算法进行比较。

4. 证明具有可变步长的SGD的定理14.8的类似定理，$\eta_t = \frac{B}{\rho\sqrt{t}}$。

15 Support Vector Machines

在这一章和下一章中，我们讨论一个非常有用的机器学习工具：支持向量机范式（SVM）用于学习高维特征空间中的线性预测器。特征空间的高维性既提高了样本复杂度，也提高了计算复杂度。

SVM算法范式通过寻找“大间隔”分离器来解决样本复杂度挑战。粗略地说，如果一个半空间将训练集与一个大间隔分开，那么所有示例不仅位于分离超平面的正确一侧，而且离它很远。将算法限制为输出大间隔分离器可以在特征空间的维度很高（甚至无限）的情况下产生小的样本复杂度。我们引入了间隔的概念，并将其与正则化损失最小化范式以及感知器算法的收敛速度联系起来。

在下一章中，我们将使用 *kernels* 的想法来解决计算复杂度挑战。

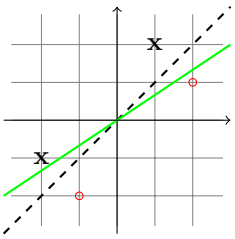
15.1 Margin and Hard-SVM

让 $S = (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$ 成为示例的训练集，其中每个 $\mathbf{x}_i \in \mathbb{R}^d$ 和 $y_i \in \{\pm 1\}$。我们说这个训练集是线性可分的，如果存在一个半空间，$(\mathbf{w}, b)$，使得 $y_i = \operatorname{sign}(\langle \mathbf{w}, \mathbf{x}_i \rangle + b)$ 对所有 i 成立。或者，这个条件可以重写为

$$\forall i \in [m], \quad y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) > 0.$$

所有满足此条件的半空间（$\mathbf{w}, b$）都是ERM假设（它们的0-1误差为零，这是可能的最小误差）。对于任何可分训练样本，存在许多ERM半空间。学习者应该选择哪一个?

例如，考虑图中描述的训练集。



当虚线黑色和实线绿色超平面都分隔了这四个例子时，我们的直觉可能让我们更倾向于黑色超平面而不是绿色超平面。一种形式化这种直觉的方法是使用*margin*的概念。

超平面相对于训练集的边界定义为训练集中一点与超平面之间的最小距离。如果一个超平面具有较大的边界，那么即使我们对每个实例进行轻微扰动，它仍然可以分离训练集。

我们将在后面看到，半空间的真实误差可以用其在训练样本上的间隔来界定（间隔越大，误差越小），无论这个半空间位于哪个欧几里得维度中。

Hard-SVM 这是我们在训练集中返回具有最大可能边界的ERM超平面的学习规则。为了正式定义Hard-SVM，我们首先使用定义半空间的参数来表示点$\mathbf{x}$到超平面的距离。

CLAIM 15.1 *The distance between a point* $\mathbf{x}$ *and the hyperplane defined by* $(\mathbf{w}, b)$ *where* $\|\mathbf{w}\| = 1$ *is* $|\langle \mathbf{w}, \mathbf{x} \rangle + b|$.

Proof 两点 $\mathbf{x}$ 和超平面之间的距离定义为

$$\min\{\|\mathbf{x} - \mathbf{v}\| : \langle \mathbf{w}, \mathbf{v} \rangle + b = 0\}.$$

取 $\mathbf{v} = \mathbf{x} - (\langle \mathbf{w}, \mathbf{x} \rangle + b)\mathbf{w}$，我们有

$$\langle \mathbf{w}, \mathbf{v} \rangle + b = \langle \mathbf{w}, \mathbf{x} \rangle - (\langle \mathbf{w}, \mathbf{x} \rangle + b)\|\mathbf{w}\|^2 + b = 0,$$

和

$$\|\mathbf{x} - \mathbf{v}\| = |\langle \mathbf{w}, \mathbf{x} \rangle + b| \, \|\mathbf{w}\| = |\langle \mathbf{w}, \mathbf{x} \rangle + b|.$$

因此，距离最多为 $|\langle \mathbf{w}, \mathbf{x} \rangle + b|$。接下来，在超平面上取另一点 $\mathbf{u}$，因此 $\langle \mathbf{w}, \mathbf{u} \rangle + b = 0$。我们有

$$\begin{aligned}
\|\mathbf{x} - \mathbf{u}\|^2 &= \|\mathbf{x} - \mathbf{v} + \mathbf{v} - \mathbf{u}\|^2 \\
&= \|\mathbf{x} - \mathbf{v}\|^2 + \|\mathbf{v} - \mathbf{u}\|^2 + 2\langle \mathbf{x} - \mathbf{v}, \mathbf{v} - \mathbf{u} \rangle \\
&\geq \|\mathbf{x} - \mathbf{v}\|^2 + 2\langle \mathbf{x} - \mathbf{v}, \mathbf{v} - \mathbf{u} \rangle \\
&= \|\mathbf{x} - \mathbf{v}\|^2 + 2(\langle \mathbf{w}, \mathbf{x} \rangle + b)\langle \mathbf{w}, \mathbf{v} - \mathbf{u} \rangle \\
&= \|\mathbf{x} - \mathbf{v}\|^2,
\end{aligned}$$

在最后一个等式成立是因为 $\langle \mathbf{w}, \mathbf{v} \rangle = \langle \mathbf{w}, \mathbf{u} \rangle = -b$。因此，距离

在 $\mathbf{x}$ 和 $\mathbf{u}$ 之间的距离至少是 $\mathbf{x}$ 和 $\mathbf{v}$ 之间的距离，这证明了我们的结论。

□

基于前面的主张，训练集中到分离超平面的最近点是 $\min_{i\in[m]} |\langle \mathbf{w}, \mathbf{x}_i\rangle + b|$。因此，硬SVM规则是

$$\operatorname*{argmax}_{(\mathbf{w},b):\|\mathbf{w}\|=1} \min_{i\in[m]} |\langle \mathbf{w}, \mathbf{x}_i\rangle + b| \quad \text{s.t.} \quad \forall i,\ y_i(\langle \mathbf{w}, \mathbf{x}_i\rangle + b) > 0.$$

无论何时存在前述问题的解（即，我们处于可分离情况），我们都可以将等价问题写成如下形式（参见练习1）：{v*}

$$\operatorname*{argmax}_{(\mathbf{w},b):\|\mathbf{w}\|=1} \min_{i\in[m]} y_i(\langle \mathbf{w}, \mathbf{x}_i\rangle + b). \tag{15.1}$$

接下来，我们给出Hard-SVM规则的另一个等价公式，将其作为二次优化问题来表示.[1]

Hard-SVM

input: $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$

solve:

$$(\mathbf{w}_0, b_0) = \operatorname*{argmin}_{(\mathbf{w},b)} \|\mathbf{w}\|^2 \text{ s.t. } \forall i,\ y_i(\langle \mathbf{w}, \mathbf{x}_i\rangle + b) \geq 1 \tag{15.2}$$

output: $\hat{\mathbf{w}} = \frac{\mathbf{w}_0}{\|\mathbf{w}_0\|}, \quad \hat{b} = \frac{b_0}{\|\mathbf{w}_0\|}$

以下词元表明，硬SVM的输出确实是具有最大间隔的分离超平面。直观上，硬SVM在所有分离数据的向量中搜索具有最小范数的$\mathbf{w}$，并且对于所有i，$|\langle \mathbf{w}, \mathbf{x}_i\rangle + b| \geq$都为1。换句话说，我们强制间隔为1，但现在我们测量间隔的单位与$\mathbf{w}$的范数成比例。因此，找到具有最大间隔的半空间归结为找到范数最小的$\mathbf{w}$。形式上：

LEMMA 15.2 *The output of Hard-SVM is a solution of Equation (15.1).*

Proof 设$(\mathbf{w}^\star, b^\star)$是方程(15.1)的解，并定义$(\mathbf{w}^\star, b^\star)$所获得的边界为$\gamma^\star = \min_{i\in[m]} y_i(\langle \mathbf{w}^\star, \mathbf{x}_i\rangle + b^\star)$。因此，对于所有$i$，我们有

$$y_i(\langle \mathbf{w}^\star, \mathbf{x}_i\rangle + b^\star) \geq \gamma^\star$$

或等价地

$$y_i(\langle \tfrac{\mathbf{w}^\star}{\gamma^\star}, \mathbf{x}_i\rangle + \tfrac{b^\star}{\gamma^\star}) \geq 1.$$

因此，对 $(\frac{\mathbf{w}^\star}{\gamma^\star}, \frac{b^\star}{\gamma^\star})$ 这对数满足二次优化的条件

一个二次优化问题是目标函数为凸二次函数且约束为线性不等式的优化问题。

问题在方程（15.2）中给出。因此，$\|\mathbf{w}_0\| \leq \|\frac{\mathbf{w}^\star}{\gamma^\star}\| = \frac{1}{\gamma^\star}$。由此可知，对于所有 i，

$$y_i(\langle \hat{\mathbf{w}}, \mathbf{x}_i\rangle + \hat{b}) = \frac{1}{\|\mathbf{w}_0\|} y_i(\langle \mathbf{w}_0, \mathbf{x}_i\rangle + b_0) \geq \frac{1}{\|\mathbf{w}_0\|} \geq \gamma^\star.$$

自 $\|\hat{\mathbf{w}}\| = 1$ 以来，我们得到 $(\hat{\mathbf{w}}, \hat{b})$ 是方程 (15.1) 的一个最优解。 □

15.1.1 The Homogenous Case

通常考虑同质半空间更为方便，即通过原点的半空间，因此由sign($\langle \mathbf{w}, \mathbf{x}\rangle$)定义，其中偏置项$b$被设置为零。对于同质半空间的硬SVM相当于求解

$$\min_{\mathbf{w}} \|\mathbf{w}\|^2 \quad \text{s.t.} \ \ \forall i, \ y_i\langle \mathbf{w}, \mathbf{x}_i\rangle \geq 1. \tag{15.3}$$

正如我们在第9章中讨论的那样，我们可以通过向每个$\mathbf{x}_i$的实例添加一个特征，将学习非齐次半空间的问题降低为学习齐次半空间的问题，从而将维度增加到$d+1$。

注意，然而，方程（15.2）中给出的优化问题没有正则化偏差项 b，而如果我们使用方程（15.3）在 $\mathbb{R}^{d+1}$ 中学习一个同质半空间，那么我们也会正则化偏差项（即权重向量的 $d+1$ 分量）。然而，正则化 b 通常不会对样本复杂度产生显著影响。

15.1.2 The Sample Complexity of Hard-SVM

回忆一下，$\mathbb{R}^d$ 中半空间的 VC 维度为 $d+1$。因此，学习半空间的样本复杂度随着问题维度的增加而增长。此外，学习的基本定理告诉我们，如果示例数量显著小于 d/ϵ，则没有任何算法可以学习到 ϵ 精度的半空间。当 d 非常大时，这会成为一个问题。

为了克服这个问题，我们将在潜在数据分布上做出一个额外的假设。特别是，我们将定义一个“带边距 γ 的可分性”假设，并表明如果数据是带边距 γ 可分的，那么样本复杂度被一个关于 $1/\gamma^2$ 的函数从上方有界。因此，即使维度非常大（甚至无限），只要数据遵循带边距可分性假设，我们仍然可以保持小的样本复杂度。这与学习基本定理给出的下界没有矛盾，因为我们现在对潜在数据分布做出了一个额外的假设。

在正式定义带边缘的分离性假设之前，我们需要解决一个缩放问题。假设一个训练集 $S = (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$ 在边缘 γ 下是可分离的，即方程（15.1）的最大目标值至少为 γ。然后，对于任何正标量 $\alpha > \geqslant 0$，训练集

$S' = (\alpha \mathbf{x}_1, y1), \ldots, (\alpha \mathbf{x}_m, y_m)$ 可以在 $\alpha\gamma$ 的间隔下分离。也就是说，对数据进行简单的缩放可以使它具有任意大的间隔。因此，为了给出有意义的间隔定义，我们必须考虑示例的规模。一种形式化的方法是使用以下定义。

DEFINITION 15.3 设 $\mathcal{D}$ 是 $\mathbb{R}^d \times \{\pm 1\}$ 上的一个分布。我们称 $\mathcal{D}$ 在 (γ, ρ)-margin 下可分，如果存在 $(\mathbf{w}^\star, b^\star)$ 使得 $\|\mathbf{w}^\star\| = 1$，并且以 1 的概率在 $(\mathbf{x}, y)$ 的选择下，我们有 $y(\langle \mathbf{w}^\star, \mathbf{x} \rangle + b^\star) \geq \gamma$ 和 $\|\mathbf{x}\| \leq \rho$。类似地，我们称 $\mathcal{D}$ 在使用同质半空间的情况下，在 (γ, ρ)-margin 下可分，如果上述条件在形式为 $(\mathbf{w}^\star, 0)$ 的半空间下成立。

在本书的高级部分（第26章），我们将证明Hard-SVM的样本复杂度依赖于$(\rho/\gamma)^2$，且与维度d无关。特别是，第26.3节中的定理26.13陈述如下：

THEOREM 15.4 *Let $\mathcal{D}$ be a distribution over $\mathbb{R}^d \times \{\pm 1\}$ that satisfies the (γ, ρ)-separability with margin assumption using a homogenous halfspace. Then, with probability of at least $1 - \delta$ over the choice of a training set of size m, the 0-1 error of the output of Hard-SVM is at most*

$$\sqrt{\frac{4\,(\rho/\gamma)^2}{m}} + \sqrt{\frac{2\log(2/\delta)}{m}}.$$

Remark 15.1 (边缘和感知机)在第9.1.2节中，我们描述并分析了感知机算法，用于寻找关于半空间类别的ERM假设。特别是，在第9.1定理9.1中，我们给出了感知机在给定训练集上可能进行的更新次数的上界。可以证明（参见练习2），上界正好是$(\rho/\gamma)^2$，其中ρ是示例半径，γ是边缘。

15.2 Soft-SVM and Norm Regularization

硬SVM公式的假设是训练集是线性可分的，这是一个相当强的假设。软SVM可以看作是对硬SVM规则的放宽，即使训练集不是线性可分的也可以应用。

方程（15.2）中的优化问题强制执行所有 i 的硬约束 $y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1$。一种自然的放松方法是允许训练集中的一些示例违反约束。这可以通过引入非负松弛变量 $\xi_1, \ldots, \xi_m$ 并将每个约束 $y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1$ 替换为约束 $y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1 - \xi_i$ 来建模。也就是说，ξ_i 衡量约束 $y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1$ 被违反的程度。软SVM同时最小化与边缘) 对应的 $\mathbf{w}$ (的范数以及与约束违反) 对应的 ξ_i (的平均值。两者之间的权衡

术语由参数 λ 控制。这导致软SVM优化问题：

Soft-SVM

input: $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$
parameter: $\lambda > 0$
solve:

$$\min_{\mathbf{w},b,\boldsymbol{\xi}} \left(\lambda \|\mathbf{w}\|^2 + \frac{1}{m} \sum_{i=1}^{m} \xi_i \right) \quad \text{s.t. } \forall i, \quad y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1 - \xi_i \quad \text{and} \quad \xi_i \geq 0 \tag{15.4}$$

output: $\mathbf{w}, b$

我们可以将方程（15.4）重写为一个正则化损失最小化问题。回忆一下铰链损失的定义：

$$\ell^{\text{hinge}}((\mathbf{w}, b), (\mathbf{x}, y)) = \max\{0, 1 - y(\langle \mathbf{w}, \mathbf{x} \rangle + b)\}.$$

给定 $(\mathbf{w}, b)$ 和一个训练集 S，S 上的平均铰链损失表示为 $L_S^{\text{hinge}}((\mathbf{w}, b))$。现在，考虑正则化损失最小化问题：

$$\min_{\mathbf{w},b} \left(\lambda \|\mathbf{w}\|^2 + L_S^{\text{hinge}}((\mathbf{w}, b)) \right). \tag{15.5}$$

CLAIM 15.5 *Equation (15.4) and Equation (15.5) are equivalent.*

Proof 修复一些 $\mathbf{w}, b$ 并考虑方程（15.4）中 $\boldsymbol{\xi}$ 的最小化。修复一些 i。由于 ξ_i 必须非负，当 $y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq$ 为 1 时，ξ_i 的最佳赋值为 0，否则为 1 $- y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b)$。换句话说，对于所有 i，有 $\xi_i = \ell^{\text{hinge}}((\mathbf{w}, b), (\mathbf{x}_i, y_i))$，从而得出该结论。 □

因此，我们看到软SVM属于我们在上一章中研究的正则化损失最小化范例。软SVM算法，即方程（15.5）的解，倾向于低范数分离器。在方程（15.5）中，我们旨在最小化的目标函数不仅惩罚训练错误，还惩罚大范数。

通常考虑软SVM来学习同质半空间更为方便，其中偏差项 b 被设置为0，从而得到以下优化问题：

$$\min_{\mathbf{w}} \left(\lambda \|\mathbf{w}\|^2 + L_S^{\text{hinge}}(\mathbf{w}) \right), \tag{15.6}$$

哪里

$$L_S^{\text{hinge}}(\mathbf{w}) = \frac{1}{m} \sum_{i=1}^{m} \max\{0, 1 - y\langle \mathbf{w}, \mathbf{x}_i \rangle\}.$$

15.2.1 The Sample Complexity of Soft-SVM

我们现在分析Soft-SVM在均匀半空间情况下的样本复杂度（即方程（15.6）的输出）。在引理13.8中，我们推导了一个正则化损失最小化框架的泛化界，假设损失函数是凸的和Lipschitz的。我们已经证明了hinge损失是凸的，因此只剩下分析hinge损失的Lipschitz性质。

CLAIM 15.6 *Let $f(\mathbf{w}) = \max\{0, 1 - y\langle \mathbf{w}, \mathbf{x}\rangle\}$. Then, f is $\|\mathbf{x}\|$-Lipschitz.*

Proof 它很容易验证，在 $\mathbf{w}$ 处 f 的任何子梯度都是形式 $\alpha\mathbf{x}$ 的，其中 $|\alpha| \leq 1$。现在这个结论可以从引理14.7得出。 □

推论13.8因此得出以下结论：

COROLLARY 15.7 *Let $\mathcal{D}$ be a distribution over $\mathcal{X} \times \{0, 1\}$, where $\mathcal{X} = \{\mathbf{x} : \|\mathbf{x}\| \leq \rho\}$. Consider running Soft-SVM (Equation (15.6)) on a training set $S \sim \mathcal{D}^m$ and let $A(S)$ be the solution of Soft-SVM. Then, for every $\mathbf{u}$,*

$$\mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}[L^{\text{hinge}}_{\mathcal{D}}(A(S))] \;\leq\; L^{\text{hinge}}_{\mathcal{D}}(\mathbf{u}) + \lambda\|\mathbf{u}\|^2 + \frac{2\rho^2}{\lambda m}.$$

Furthermore, since the hinge loss upper bounds the $0-1$ loss we also have

$$\mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}[L^{0-1}_{\mathcal{D}}(A(S))] \;\leq\; L^{\text{hinge}}_{\mathcal{D}}(\mathbf{u}) + \lambda\|\mathbf{u}\|^2 + \frac{2\rho^2}{\lambda m}.$$

Last, for every $B > 0$, if we set $\lambda = \sqrt{\frac{2\rho^2}{B^2 m}}$ then

$$\mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}[L^{0-1}_{\mathcal{D}}(A(S))] \;\leq\; \mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}[L^{\text{hinge}}_{\mathcal{D}}(A(S))] \;\leq\; \min_{\mathbf{w}:\|\mathbf{w}\|\leq B} L^{\text{hinge}}_{\mathcal{D}}(\mathbf{w}) + \sqrt{\frac{8\rho^2 B^2}{m}}.$$

因此，我们看到我们可以将学习半空间的学习样本复杂度控制为该半空间范数的函数，而与定义半空间的欧几里得空间维度无关。当我们通过嵌入到高维特征空间进行学习时，这变得非常重要，正如我们将在下一章中考虑的那样。

Remark 15.2 条件是 $\mathcal{X}$ 将包含具有有界范数的向量，这源于损失函数将是Lipschitz的要求。这不仅仅是一个技术细节。正如我们之前讨论的，如果不限制实例的规模，具有大间隔的分离是没有意义的。实际上，如果没有对规模的约束，我们总是可以通过将所有实例乘以一个大标量来扩大间隔。

15.2.2 Margin and Norm-Based Bounds versus Dimension

我们为Hard-SVM和Soft-SVM推导出的界限不依赖于实例空间的维度。相反，界限依赖于范数{v*}。

示例，ρ，半空间 B (的范数或等价地，边缘参数 γ)，以及在不可分情况下，边界还取决于所有范数为 $\leq B$ 的半空间的最小铰链损失。相比之下，同质半空间类的 VC 维度为 d，这意味着 ERM 假设的误差随着 $\sqrt{d/m}$ 的减小而减小。我们现在给出一个例子，其中 $\rho^2 B^2 \ll d$；因此，15.7 节推论中给出的界限比 VC 界限要好得多。

考虑学习如何根据主题对短文本文档进行分类的问题，例如，文档是否关于体育。我们首先需要将文档表示为向量。一种简单而有效的方法是使用 *bag-of-words* 表示。也就是说，我们定义一个单词字典，并将维度 d 设置为字典中单词的数量。给定一个文档，我们将其表示为向量 $\mathbf{x} \in \{0,1\}^d$，其中 $x_i = 1$ 如果字典中的 i’ th 个单词出现在文档中，否则为 $x_i = 0$。因此，对于这个问题，ρ^2 的值将是给定文档中不同单词的最大数量。

一个半空间为这个问题分配权重给单词。假设通过给几十个单词分配正负权重，我们可以以合理的准确性确定一个给定的文档是否关于体育，这是自然的。因此，对于这个问题，B^2的值可以设置为小于100。总的来说，可以说$B^2\rho^2$的值小于10,000是合理的。

另一方面，字典的典型大小远大于10,000。例如，英语中有超过100,000个不同的单词。因此，我们已经展示了一个问题，其中使用SVM规则学习半空间和使用vanilla ERM规则学习半空间之间可能存在数量级差异。

当然，可以构造出SVM界限比VC界限更差的问题。当我们使用SVM时，我们实际上引入了另一种归纳偏差——我们更喜欢大间隔半空间。虽然这种归纳偏差可以显著减少我们的估计误差，但它也可能扩大近似误差。

15.2.3 The Ramp Loss*

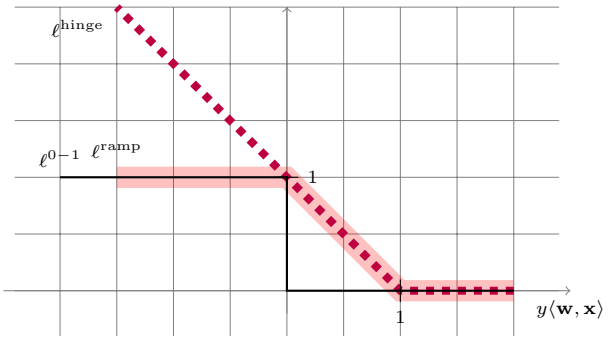
我们推导的15.7引理中的基于边界的界限依赖于我们最小化铰链损失的事实。正如我们在前一节中所示，项$\sqrt{\rho^2 B^2/m}$可以比VC界限中的对应项$\sqrt{d/m}$小得多。然而，15.7引理中的近似误差是以铰链损失来衡量的，而VC界限中的近似误差是以0−1损失来衡量的。由于铰链损失上界了0−1损失，因此相对于0−1损失的近似误差永远不会超过铰链损失的近似误差。

无法推导涉及估计误差项 $\sqrt{\rho^2 B^2/m}$ 的 0−1 损失的界限。这源于 0−1 损失是可缩放的

无感觉，因此当我们用0−1损失来衡量误差时，$\mathbf{w}$的范数或其边缘没有意义。然而，可以定义一个损失函数，一方面它是尺度敏感的，因此享有估计误差$\sqrt{\rho^2 B^2/m}$，另一方面它更类似于0−1损失。一个选项是*ramp loss*，定义为

$$\ell^{\text{ramp}}(\mathbf{w},(\mathbf{x},y)) = \min\{1, \ell^{\text{hinge}}(\mathbf{w},(\mathbf{x},y))\} = \min\{1\,,\, \max\{0, 1 - y\langle \mathbf{w},\mathbf{x}\rangle\}\}.$$

坡度损失以与0−1损失相同的方式惩罚错误，不对间隔较大的示例进行惩罚。坡度损失与0−1损失之间的区别仅在于正确分类但间隔不显著的示例。本书的高级部分给出了坡度损失的一般化界限（见附录26.3）。



SVM依赖于hinge损失而不是ramp损失的原因是hinge损失是凸的，因此从*computational*的角度来看，最小化hinge损失可以高效地执行。相比之下，最小化ramp损失的问题在计算上是不可行的。

15.3 Optimality Conditions and "Support Vectors"*

"支持向量机"这个名字来源于硬SVM的解$\mathbf{w}_0$是由（即位于）与分离超平面恰好距离$1/\|\mathbf{w}_0\|$的示例支持的。因此，这些向量被称为*support vectors*。为了理解这一点，我们依赖于**Fritz John optimality conditions**。

THEOREM 15.8 *Let $\mathbf{w}_0$ be as defined in Equation (15.3) and let $I = \{i : |\langle \mathbf{w}_0, \mathbf{x}_i\rangle| = 1\}$. Then, there exist coefficients $\alpha_1, \ldots, \alpha_m$ such that*

$$\mathbf{w}_0 = \sum_{i \in I} \alpha_i \mathbf{x}_i.$$

示例 $\{\mathbf{x}_i : i \in I\}$ 被称为 *support vectors*。
该定理的证明通过将以下引理应用于方程（15.3）得出。

LEMMA 15.9（弗里茨·约翰）*Suppose that*

$$\mathbf{w}^\star \in \operatorname*{argmin}_{\mathbf{w}} f(\mathbf{w}) \quad s.t. \ \ \forall i \in [m], \ g_i(\mathbf{w}) \leq 0,$$

where $f, g_1, \ldots, g_m$ *are differentiable. Then, there exists* $\boldsymbol{\alpha} \in \mathbb{R}^m$ *such that* $\nabla f(\mathbf{w}^\star) + \sum_{i \in I} \alpha_i \nabla g_i(\mathbf{w}^\star) = \mathbf{0}$*, where* $I = \{i : g_i(\mathbf{w}^\star) = 0\}$*.*

15.4 Duality*

历史上，SVM的许多属性是通过考虑方程（15.3）的*dual*获得的。我们对于SVM的表述不依赖于对偶性。为了完整性，我们以下将展示如何推导方程（15.3）的对偶。

我们首先将问题重新表述为等价形式，如下所示。考虑函数

$$g(\mathbf{w}) \ = \ \max_{\boldsymbol{\alpha} \in \mathbb{R}^m : \boldsymbol{\alpha} \geq \mathbf{0}} \sum_{i=1}^m \alpha_i (1 - y_i \langle \mathbf{w}, \mathbf{x}_i \rangle) \ = \ \begin{cases} 0 & \text{if } \forall i, y_i \langle \mathbf{w}, \mathbf{x}_i \rangle \geq 1 \\ \infty & \text{otherwise} \end{cases} .$$

我们可以因此将方程（15.3）重写为

$$\min_{\mathbf{w}} \left(\|\mathbf{w}\|^2 + g(\mathbf{w}) \right). \tag{15.7}$$

重排前面的内容，我们得到方程（15.3）可以重写为问题

$$\min_{\mathbf{w}} \max_{\boldsymbol{\alpha} \in \mathbb{R}^m : \boldsymbol{\alpha} \geq \mathbf{0}} \left(\frac{1}{2} \|\mathbf{w}\|^2 + \sum_{i=1}^m \alpha_i (1 - y_i \langle \mathbf{w}, \mathbf{x}_i \rangle) \right). \tag{15.8}$$

现在假设我们将上述方程中的 min 和 max 的顺序颠倒。这只会降低目标值（参见练习4），并且我们有

$$\begin{aligned} &\min_{\mathbf{w}} \max_{\boldsymbol{\alpha} \in \mathbb{R}^m : \boldsymbol{\alpha} \geq \mathbf{0}} \left(\frac{1}{2} \|\mathbf{w}\|^2 + \sum_{i=1}^m \alpha_i (1 - y_i \langle \mathbf{w}, \mathbf{x}_i \rangle) \right) \\ &\geq \max_{\boldsymbol{\alpha} \in \mathbb{R}^m : \boldsymbol{\alpha} \geq \mathbf{0}} \min_{\mathbf{w}} \left(\frac{1}{2} \|\mathbf{w}\|^2 + \sum_{i=1}^m \alpha_i (1 - y_i \langle \mathbf{w}, \mathbf{x}_i \rangle) \right). \end{aligned}$$

前述不等式称为 *weak duality*。结果，在我们的情况下，*strong duality* 也成立；也就是说，不等式成立时取等号。因此，*dual* 问题为

$$\max_{\boldsymbol{\alpha} \in \mathbb{R}^m : \boldsymbol{\alpha} \geq \mathbf{0}} \min_{\mathbf{w}} \left(\frac{1}{2} \|\mathbf{w}\|^2 + \sum_{i=1}^m \alpha_i (1 - y_i \langle \mathbf{w}, \mathbf{x}_i \rangle) \right). \tag{15.9}$$

我们可以通过注意到一旦 $\boldsymbol{\alpha}$ 被固定，优化

问题关于 $\mathbf{w}$ 是无约束的，并且目标函数可微；因此，在最优解处，梯度等于零：

$$\mathbf{w} - \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i \;=\; 0 \quad \Rightarrow \quad \mathbf{w} = \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i .$$

这表明解必须在示例的线性跨度中，这是一个我们将在后面用来推导具有核的SVM的事实。将前面的内容代入方程（15.9）中，我们得到对偶问题可以重写为

$$\max_{\boldsymbol{\alpha} \in \mathbb{R}^m : \boldsymbol{\alpha} \geq \mathbf{0}} \left(\frac{1}{2} \left\| \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i \right\|^2 + \sum_{i=1}^{m} \alpha_i \left(1 - y_i \left\langle \sum_j \alpha_j y_j \mathbf{x}_j , \mathbf{x}_i \right\rangle \right) \right) . \tag{15.10}$$

重新排列得到对偶问题

$$\max_{\boldsymbol{\alpha} \in \mathbb{R}^m : \boldsymbol{\alpha} \geq \mathbf{0}} \left(\sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \langle \mathbf{x}_j , \mathbf{x}_i \rangle \right) . \tag{15.11}$$

注意，对偶问题仅涉及实例之间的内积，不需要直接访问实例内部的特定元素。当实现具有核的SVM时，这一性质很重要，我们将在下一章中讨论。

15.5 Implementing Soft-SVM Using SGD

在这个部分，我们描述了一个非常简单的算法来解决软SVM的优化问题，即，

$$\min_{\mathbf{w}} \left(\frac{\lambda}{2} \|\mathbf{w}\|^2 + \frac{1}{m} \sum_{i=1}^{m} \max\{0, 1 - y \langle \mathbf{w}, \mathbf{x}_i \rangle\} \right) . \tag{15.12}$$

我们依赖于SGD框架来解决正则化损失最小化问题，如第14.5.3节所述。

回忆起，基于方程（14.15），我们可以将SGD的更新规则重写为

$$\mathbf{w}^{(t+1)} = -\frac{1}{\lambda t} \sum_{j=1}^{t} \mathbf{v}_j ,$$

在 $\mathbf{w}^{(j)}$ 上随机选择迭代 j 的示例处，$\mathbf{v}_j$ 是损失函数的子梯度。对于 hinge 损失，给定一个示例 $(\mathbf{x}, y)$，如果 $y \langle \mathbf{w}^{(j)}, \mathbf{x} \rangle \geq 1$，则可以选择 $\mathbf{v}_j$ 为 $\mathbf{0}$，否则为 $\mathbf{v}_j = -y\,\mathbf{x}$（参见示例 14.2）。记 $\boldsymbol{\theta}^{(t)} = -\sum_{j<t} \mathbf{v}_j$，我们得到以下过程。

SGD for Solving Soft-SVM

goal: Solve Equation (15.12)
parameter: T
initialize: $\boldsymbol{\theta}^{(1)} = \mathbf{0}$
for $t = 1, \ldots, T$
  Let $\mathbf{w}^{(t)} = \frac{1}{\lambda t}\boldsymbol{\theta}^{(t)}$
  Choose i uniformly at random from $[m]$
  If $(y_i \langle \mathbf{w}^{(t)}, \mathbf{x}_i \rangle < 1)$
    Set $\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} + y_i \mathbf{x}_i$
  Else
    Set $\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)}$
output: $\bar{\mathbf{w}} = \frac{1}{T}\sum_{t=1}^{T} \mathbf{w}^{(t)}$

15.6 Summary

SVM是一种学习半空间的算法，具有某种类型的前置知识，即对大间隔的偏好。硬SVM寻求具有最大间隔的半空间，以完美地分离数据，而软SVM不假设数据的可分性，并允许在一定程度上违反约束。这两种类型SVM的样本复杂度与直接半空间学习的样本复杂度不同，因为它不依赖于域的维度，而更多地依赖于如$\mathbf{x}$和$\mathbf{w}$的最大范数等参数。

下一章将实现维度无关样本复杂性的重要性，其中我们将讨论将给定域嵌入到某些高维特征空间中，作为丰富我们的假设类的方法。这样的程序引发了计算和样本复杂性问题。后者通过使用支持向量机（SVM）来解决，而前者可以通过使用具有核的SVM来解决，正如我们将在下一章中看到的。

15.7 Bibliographic Remarks

SVMs已在（Cortes & Vapnik 1995，Boser，Guyon & Vapnik 1992）中引入。关于SVM的理论和实践方面有许多优秀的书籍。例如，（Vapnik 1995，Cristianini & Shawe-Taylor 2000，Schölkopf & Smola 2002，Hsu，Chang & Lin 2003，Stein wart & Christmann 2008）。在Shalev-Shwartz等人（2007）中提出了使用SGD来解决软SVM。

15.8 Exercises

1. 证明硬SVM规则，即，

$$\operatorname*{argmax}_{(\mathbf{w},b):\|\mathbf{w}\|=1} \min_{i\in[m]} |\langle \mathbf{w}, \mathbf{x}_i\rangle + b| \quad \text{s.t.} \quad \forall i,\ y_i(\langle \mathbf{w}, \mathbf{x}_i\rangle + b) > 0,$$

等价于以下公式：{v*}

$$\operatorname*{argmax}_{(\mathbf{w},b):\|\mathbf{w}\|=1} \min_{i\in[m]} y_i(\langle \mathbf{w}, \mathbf{x}_i\rangle + b). \tag{15.13}$$

Hint: 定义 $\mathcal{G} = \{(\mathbf{w}, b) : \forall i, y_i(\langle \mathbf{w}, \mathbf{x}_i\rangle + b) > 0\}$。
1. 证明以下内容：

$$\operatorname*{argmax}_{(\mathbf{w},b):\|\mathbf{w}\|=1} \min_{i\in[m]} y_i(\langle \mathbf{w}, \mathbf{x}_i\rangle + b) \in \mathcal{G}$$

2. 证明 $\forall(\mathbf{w}, b) \in \mathcal{G}$,

$$\min_{i\in[m]} y_i(\langle \mathbf{w}, \mathbf{x}_i\rangle + b) = \min_{i\in[m]} |\langle \mathbf{w}, \mathbf{x}_i\rangle + b|$$

2. **Margin and the Perceptron** 考虑一个线性可分且具有边缘 γ 的训练集，并且所有实例都在半径为 ρ 的球体内。证明当在给定的 9.1.2 节中的批感知器算法上运行此训练集时，该算法将进行的最大更新次数是 $(\rho/\gamma)^2$。

3. **Hard versus soft SVM:** 证明或反驳以下论断：
There exists $\lambda > 0$ such that for every sample S of $m > 1$ examples, which is separable by the class of homogenous halfspaces, the hard-SVM and the soft-SVM (with parameter λ) learning rules return exactly the same weight vector.

4. **Weak duality:** 证明对于任意两个向量变量 $\mathbf{x} \in \mathcal{X}, \mathbf{y} \in \mathcal{Y}$ 的函数 f，都有

$$\min_{\mathbf{x}\in\mathcal{X}} \max_{\mathbf{y}\in\mathcal{Y}} f(\mathbf{x}, \mathbf{y}) \geq \max_{\mathbf{y}\in\mathcal{Y}} \min_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x}, \mathbf{y}).$$

16 Kernel Methods

在上一章中，我们描述了在高维特征空间中学习半空间的SVM范式。这使得我们能够通过首先将数据映射到高维特征空间，然后在其中学习线性预测器来丰富半空间的表达能力。这与AdaBoost算法类似，该算法学习在基假设上的半空间组合。虽然这种方法极大地扩展了半空间预测器的表达能力，但它同时提出了样本复杂性和计算复杂性的挑战。在上一章中，我们使用边缘的概念解决了样本复杂性问题。在这一章中，我们使用$kernels$方法来解决计算复杂性的挑战。

我们本章首先描述了将数据嵌入到高维特征空间中的想法。然后，我们引入了核的概念。核是一种实例之间的相似度度量。核相似性的特殊性质是，它们可以被视为某些希尔伯特空间（或某些高维的欧几里得空间）中的内积，而实例空间则被虚拟嵌入其中。我们介绍了“核技巧”，它使得学习算法的计算效率得到提高，而无需显式处理领域实例的高维表示。基于核的学习算法，特别是核-SVM，是非常有用且流行的机器学习工具。它们的成功可能归因于能够灵活地适应特定领域的先验知识，以及拥有一套高效实现算法。

16.1 Embeddings into Feature Spaces

半平面的表达能力相当有限 - 例如，以下训练集无法由半平面分离。

定义域为实数线；考虑域点 {v*}$10, -9, -8, \ldots, 0, 1, \ldots, 9, 10\}$，其中标签为 $+1$ 对于所有 x 满足 $|x| > 2$ 和 -1 否则。

为了使半空间类更具表达性，我们首先可以将原始实例空间映射到另一个空间（可能是一个更高维度的空间），然后在那个空间中学习一个半空间。例如，考虑之前提到的例子。不是在原始表示中学习一个半空间，让我们

首先定义一个映射 $\psi : \mathbb{R} \to \mathbb{R}^2$ 如下:

$$\psi(x) = (x, x^2).$$

我们使用术语 *feature space* 来表示 ψ 的范围。应用 ψ 后，数据可以很容易地用半空间 $h(x) =$ 符号$(\langle \mathbf{w}, \psi(x) \rangle - b)$来解释，其中 $\mathbf{w} = (0, 1)$ 和 $b = 5$。

基本范式如下:

1. 给定一些域集 $\mathcal{X}$ 和一个学习任务，选择一个映射 $\psi : \mathcal{X} \to \mathcal{F}$，对于某个 *feature space* $\mathcal{F}$，这通常将是 $\mathbb{R}^n$ 对于某个 n (然而，此类映射的范围可以是任何 *Hilbert space*，包括我们将在后面展示的无限维空间，)。2. 给定一个标记示例序列，$S = (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$，创建图像序列 $\hat{S} = (\psi(\mathbf{x}_1), y_1), \ldots, (\psi(\mathbf{x}_m), y_m)$。3. 在 $\hat{S}$ 上训练一个线性预测器 h。4. 预测测试点 $\mathbf{x}$ 的标签为 $h(\psi(\mathbf{x}))$。

注意，对于每个在$\mathcal{X} \times \mathcal{Y}$上的概率分布$\mathcal{D}$，我们可以通过设置每个子集 $A \subseteq \mathcal{F} \times \mathcal{Y}$的$\mathcal{D}^{\psi}(A) = \mathcal{D}(\psi^{-1}(A))$.[1]来轻松地定义其像概率分布$\mathcal{D}^{\psi}$在$\mathcal{F} \times \mathcal{Y}$上。由此可知，对于特征空间$L_{\mathcal{D}^{\psi}}(h) = L_{\mathcal{D}}(h \circ \psi)$上的每个预测器$h$，其中$h \circ \psi$是$h$到$\psi$的合成。

这个学习范式的成功取决于为给定的学习任务选择一个好的 ψ：也就是说，一个将使数据分布的图像（接近）在特征空间中线性可分 ψ，从而使结果算法成为给定任务的优秀学习器。选择这样的嵌入需要对该任务有先验知识。然而，通常使用一些通用的映射，这些映射使我们能够丰富半空间类并扩展其表达能力。一个值得注意的例子是多项式映射，它是我们在前一个例子中看到的 ψ 的一般化。

回忆一下，标准半空间分类器在实例 $\mathbf{x}$ 上的预测是基于线性映射 $\mathbf{x} \mapsto \langle \mathbf{w}, \mathbf{x} \rangle$。我们可以将线性映射推广到多项式映射 $\mathbf{x} \mapsto p(\mathbf{x})$，其中 p 是一个 k 次的多项式。为了简单起见，首先考虑 $\mathbf{x}$ 是一维的情况。在这种情况下，$p(x) = \sum_{j=0}^{k} w_j x^j$，其中 $\mathbf{w} \in \mathbb{R}^{k+1}$ 是我们需要学习的多项式的系数向量。我们可以重写 $p(x) = \langle \mathbf{w}, \psi(x) \rangle$，其中 $\psi : \mathbb{R} \to \mathbb{R}^{k+1}$ 是映射 $x \mapsto (1, x, x^2, x^3, \ldots, x^k)$。由此可知，在 $(k + 1)$ 维特征空间中学习一个 k 次多项式可以通过学习一个线性映射来完成。

更一般地，从 $\mathbb{R}^n$ 到 $\mathbb{R}$ 的一个 k 次多元多项式可以写成

$$p(\mathbf{x}) = \sum_{J \in [n]^r : r \leq k} w_J \prod_{i=1}^{r} x_{J_i}. \tag{16.1}$$

1 这对每个满足 $\psi^{-1}(A)$ 关于 $\mathcal{D}$ 可测的 A 都有定义。

与之前一样，我们可以重写 $p(\mathbf{x}) = \langle \mathbf{w}, \psi(\mathbf{x}) \rangle$，其中现在 $\psi : \mathbb{R}^n \to \mathbb{R}^d$ 是这样的，对于每个 $J \in [n]^r$，$r \leq k$，与 $\psi(\mathbf{x})$ 相关的坐标是单项式 $\prod_{i=1}^{r} x_{J_i}$。

自然地，基于多项式的分类器比半空间产生更丰富的假设类。我们在本章开头看到一个例子，其中训练集在其原始域（$\mathcal{X} = \mathbb{R}$）中不能被半空间分离，但在嵌入$x \mapsto (x,\ x^2)$之后可以完美分离。因此，虽然分类器在特征空间中总是线性的，但它可以在原始空间上表现出高度非线性行为，这是从其中采样实例的空间。

通常，我们可以选择任何将原始实例映射到某些 *Hilbert space* 的特征映射 ψ。[2] 欧几里得空间 $\mathbb{R}^d$ 是任何有限 d 的希尔伯特空间。但也有一些无限维的希尔伯特空间（我们将在本章后面看到）。

本讨论的底线是，我们首先通过应用一个非线性映射 ψ，将实例空间映射到某个特征空间，然后在那个特征空间中学习一个半空间，从而丰富半空间的类别。然而，如果 ψ 的范围是一个高维空间，我们面临两个问题。首先，$\mathbb{R}^n$ 中半空间的 VC 维度为 $n + 1$，因此，如果 ψ 的范围非常大，我们需要更多的样本才能在 ψ 的范围内学习一个半空间。其次，从计算的角度来看，在高维空间中进行计算可能成本过高。实际上，即使在特征空间中向量 $\mathbf{w}$ 的表示也可能是不切实际的。第一个问题可以通过使用大间隔（或低范数预测器）的范式来解决，正如我们在上一章中在 SVM 算法的背景下所讨论的那样。在下一节中，我们将解决计算问题。

16.2 The Kernel Trick

我们已经看到，将输入空间嵌入到某个高维特征空间可以使半空间学习更具表现力。然而，这种学习的计算复杂度可能仍然是一个严重的障碍——在非常高维数据上计算线性分离器可能非常昂贵。对此问题的常见解决方案是基于核的学习。在这个上下文中，“核”一词用于描述特征空间中的内积。给定某个域空间 $\mathcal{X}$ 到某个希尔伯特空间的一个嵌入 ψ，我们定义核函数 $K(\mathbf{x}, \mathbf{x}') = \langle \psi(\mathbf{x}), \psi(\mathbf{x}') \rangle$。可以将 K 视为指定实例之间的相似性，将嵌入 ψ 视为将域集映射到

[2] A Hilbert space is a vector space with an inner product, which is also complete. A space is complete if all Cauchy sequences in the space converge.
In our case, the norm $\|\mathbf{w}\|$ is defined by the inner product $\sqrt{\langle \mathbf{w}, \mathbf{w} \rangle}$. The reason we require the range of ψ to be in a Hilbert space is that projections in a Hilbert space are well defined. In particular, if M is a linear subspace of a Hilbert space, then every $\mathbf{x}$ in the Hilbert space can be written as a sum $\mathbf{x} = \mathbf{u} + \mathbf{v}$ where $\mathbf{u} \in M$ and $\langle \mathbf{v}, \mathbf{w} \rangle = 0$ for all $\mathbf{w} \in M$. We use this fact in the proof of the representer theorem given in the next section.

$\mathcal{X}$ 进入一个将这些相似性实现为内积的空间。结果发现，许多针对半空间的学习算法只需在域点对的核函数值的基础上进行即可。这类算法的主要优势是它们在不指定该空间中的点或显式表达嵌入 ψ 的情况下，在高维特征空间中实现线性分离器。本节的其余部分致力于构建此类算法。

在上一章中，我们看到了即使特征空间的维度很高，通过正则化 $\mathbf{w}$ 的范数也能得到小的样本复杂度。有趣的是，正如我们稍后所展示的，正则化 $\mathbf{w}$ 的范数也有助于克服计算问题。为了做到这一点，首先请注意，我们在上一章中推导出的所有 SVM 优化问题的版本都是以下一般问题的实例：

$$\min_{\mathbf{w}} \left(f\left(\langle \mathbf{w}, \psi(\mathbf{x}_1) \rangle, \ldots, \langle \mathbf{w}, \psi(\mathbf{x}_m) \rangle \right) + R(\|\mathbf{w}\|) \right), \tag{16.2}$$

在 $f : \mathbb{R}^m \to \mathbb{R}$ 是一个任意函数且 $R : \mathbb{R}_+ \to \mathbb{R}$ 是单调不减函数的情况下。例如，从方程（15.6）中可以推导出同质半空间的软SVM（Soft-SVM），通过让 $R(a) = \lambda a^2$ 和 $f(a_1, \ldots, a_m) = \frac{1}{m} \sum_i$ 取 max$\{0, 1 - y_i a_i\}$。同样，从方程（16.2）中可以推导出非同质半空间的硬SVM（Hard-SVM），通过让 $R(a) = a^2$ 和让 $f(a_1, \ldots, a_m)$ 在存在 b 使得对于所有 i 有 $y_i(a_i + b) \geq 1$ 时取 0，否则取 $f(a_1, \ldots, a_m) = \infty$。

以下定理表明，存在一个最优解位于 $\{\psi(\mathbf{x}_1), \ldots, \psi(\mathbf{x}_m)\}$ 的张成空间中，该最优解满足方程（16.2）。

THEOREM 16.1（表示定理）*Assume that ψ is a mapping from $\mathcal{X}$ to a Hilbert space. Then, there exists a vector $\boldsymbol{\alpha} \in \mathbb{R}^m$ such that $\mathbf{w} = \sum_{i=1}^m \alpha_i \psi(\mathbf{x}_i)$ is an optimal solution of Equation (16.2).*

Proof 设 $\mathbf{w}^\star$ 是方程（16.2）的一个最优解。因为 $\mathbf{w}^\star$ 是希尔伯特空间的一个元素，我们可以将 $\mathbf{w}^\star$ 重写为

$$\mathbf{w}^\star = \sum_{i=1}^m \alpha_i \psi(\mathbf{x}_i) + \mathbf{u},$$

在所有 i 中 $\langle \mathbf{u}, \psi(\mathbf{x}_i) \rangle =$ 为 0。设置 $\mathbf{w} = \mathbf{w}^\star - \mathbf{u}$。显然，$\|\mathbf{w}^\star\|^2 = \|\mathbf{w}\|^2 + \|\mathbf{u}\|^2$，因此 $\|\mathbf{w}\| \leq \|\mathbf{w}^\star\|$。由于 R 是非递减的，我们得到 $R(\|\mathbf{w}\|) \leq R(\|\mathbf{w}^\star\|)$。此外，对于所有 i，我们有

$$\langle \mathbf{w}, \psi(\mathbf{x}_i) \rangle = \langle \mathbf{w}^\star - \mathbf{u}, \psi(\mathbf{x}_i) \rangle = \langle \mathbf{w}^\star, \psi(\mathbf{x}_i) \rangle,$$

因此

$$f\left(\langle \mathbf{w}, \psi(\mathbf{x}_1) \rangle, \ldots, \langle \mathbf{w}, \psi(\mathbf{x}_m) \rangle \right) = f\left(\langle \mathbf{w}^\star, \psi(\mathbf{x}_1) \rangle, \ldots, \langle \mathbf{w}^\star, \psi(\mathbf{x}_m) \rangle \right).$$

我们已经证明，方程（16.2）在 $\mathbf{w}$ 处的目标值不能大于 $\mathbf{w}^\star$ 处的目标值，因此 $\mathbf{w}$ 也是一个最优解。自 $\mathbf{w} = \sum_{i=1}^m \alpha_i \psi(\mathbf{x}_i)$ 以来，我们得出结论。

□

基于表示定理，我们可以优化（16.2）式相对于系数 $\boldsymbol{\alpha}$ 而不是系数 $\mathbf{w}$，如下所示。将 $\mathbf{w} = \sum_{j=1}^{m} \alpha_j \psi(\mathbf{x}_j)$ 写作，对于所有 i，我们有

$$\langle \mathbf{w}, \psi(\mathbf{x}_i) \rangle = \left\langle \sum_j \alpha_j \psi(\mathbf{x}_j), \psi(\mathbf{x}_i) \right\rangle = \sum_{j=1}^{m} \alpha_j \langle \psi(\mathbf{x}_j), \psi(\mathbf{x}_i) \rangle.$$

同样地，

$$\|\mathbf{w}\|^2 = \left\langle \sum_j \alpha_j \psi(\mathbf{x}_j), \sum_j \alpha_j \psi(\mathbf{x}_j) \right\rangle = \sum_{i,j=1}^{m} \alpha_i \alpha_j \langle \psi(\mathbf{x}_i), \psi(\mathbf{x}_j) \rangle.$$

让 $K(\mathbf{x}, \mathbf{x}') = \langle \psi(\mathbf{x}), \psi(\mathbf{x}') \rangle$ 是一个实现相对于嵌入 ψ 的核函数的函数。我们不必求解方程（16.2），而是可以求解等价问题

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^m} f\left(\sum_{j=1}^{m} \alpha_j K(\mathbf{x}_j, \mathbf{x}_1), \ldots, \sum_{j=1}^{m} \alpha_j K(\mathbf{x}_j, \mathbf{x}_m) \right) + R\left(\sqrt{\sum_{i,j=1}^{m} \alpha_i \alpha_j K(\mathbf{x}_j, \mathbf{x}_i)} \right). \tag{16.3}$$

为了解决方程（16.3）中给出的优化问题，我们不需要直接访问特征空间中的元素。我们唯一需要知道的是如何计算特征空间中的内积，或者等价地，计算核函数。实际上，为了解决方程（16.3），我们只需要知道矩阵 $m \times m$ G 的值，使得 $G_{i,j} = K(\mathbf{x}_i, \mathbf{x}_j)$，这通常被称为 $Gram$ 矩阵。

特别地，指定给定的方程（15.6）中的Soft-SVM问题之前的内容，我们可以将问题重新表述为

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^m} \left(\lambda \boldsymbol{\alpha}^T G \boldsymbol{\alpha} + \frac{1}{m} \sum_{i=1}^{m} \max\left\{0, 1 - y_i (G\boldsymbol{\alpha})_i\right\} \right), \tag{16.4}$$

在$(G\boldsymbol{\alpha})_i$是通过对Gram矩阵G乘以向量$\boldsymbol{\alpha}$得到的向量中的i次元素。注意，方程(16.4)可以写成二次规划，因此可以有效地求解。在下一节中，我们描述了一个解决带核的Soft-SVM的更简单的算法。

一旦我们学习了系数 $\boldsymbol{\alpha}$，我们就可以通过以下方式计算新实例的预测：

$$\langle \mathbf{w}, \psi(\mathbf{x}) \rangle = \sum_{j=1}^{m} \alpha_j \langle \psi(\mathbf{x}_j), \psi(\mathbf{x}) \rangle = \sum_{j=1}^{m} \alpha_j K(\mathbf{x}_j, \mathbf{x}).$$

与直接在特征空间优化 $\mathbf{w}$ 相比，使用核的优势在于在某些情况下，特征空间的维度

实现内核函数非常简单，而其规模极其庞大。以下给出了一些示例。

Example 16.1 (多项式核) k次多项式核定义为

$$K(\mathbf{x}, \mathbf{x}') = (1 + \langle \mathbf{x}, \mathbf{x}' \rangle)^k.$$

现在我们将证明这确实是一个核函数。也就是说，我们将证明存在一个从原始空间到某个更高维空间的映射 ψ，使得 $K(\mathbf{x}, \mathbf{x}') = \langle \psi(\mathbf{x}), \psi(\mathbf{x}') \rangle$。为了简便起见，记 $x_0 = x_0' =$ 为 1。那么，我们有

$$\begin{aligned} K(\mathbf{x}, \mathbf{x}') = (1 + \langle \mathbf{x}, \mathbf{x}' \rangle)^k &= (1 + \langle \mathbf{x}, \mathbf{x}' \rangle) \cdot \cdots \cdot (1 + \langle \mathbf{x}, \mathbf{x}' \rangle) \\ &= \left(\sum_{j=0}^{n} x_j x_j' \right) \cdot \cdots \cdot \left(\sum_{j=0}^{n} x_j x_j' \right) \\ &= \sum_{J \in \{0,1,\ldots,n\}^k} \prod_{i=1}^{k} x_{J_i} x'_{J_i} \\ &= \sum_{J \in \{0,1,\ldots,n\}^k} \prod_{i=1}^{k} x_{J_i} \prod_{i=1}^{k} x'_{J_i}. \end{aligned}$$

现在，如果我们定义 $\psi : \mathbb{R}^n \to \mathbb{R}^{(n+1)^k}$ 使得对于 $J \in \{0, 1, \ldots, n\}^k$，存在 $\psi(\mathbf{x})$ 中的一个元素等于 $\prod_{i=1}^{k} x_{J_i}$，我们得到：

$$K(\mathbf{x}, \mathbf{x}') = \langle \psi(\mathbf{x}), \psi(\mathbf{x}') \rangle.$$

由于 ψ 包含所有直到 k 次的多项式，ψ 的范围上的半空间对应于原始空间上的 k 次多项式预测器。因此，使用 k 次多项式核学习半空间使我们能够在原始空间上学习 k 次多项式预测器。

请注意，在此处实现 K 的复杂度为 $O(n)$，而特征空间的维度约为 n^k。

Example 16.2 (高斯核) 设原始实例空间为 $\mathbb{R}$，考虑映射 ψ，其中对于每个非负整数 $n \geq 0$，存在一个元素 $\psi(x)_n$ 等于 $\frac{1}{\sqrt{n!}} e^{-\frac{x^2}{2}} x^n$。那么，

$$\begin{aligned} \langle \psi(x),\ \psi(x') \rangle &= \sum_{n=0}^{\infty} \left(\frac{1}{\sqrt{n!}} e^{-\frac{x^2}{2}} x^n \right) \left(\frac{1}{\sqrt{n!}} e^{-\frac{(x')^2}{2}} (x')^n \right) \\ &= e^{-\frac{x^2 + (x')^2}{2}} \sum_{n=0}^{\infty} \left(\frac{(xx')^n}{n!} \right) \\ &= e^{-\frac{\|x - x'\|^2}{2}}. \end{aligned}$$

这里特征空间是无限维的，而评估核非常

简单。更一般地，给定标量 $\sigma > 0$，高斯核被定义为

$$K(\mathbf{x}, \mathbf{x}') = e^{-\frac{\|\mathbf{x}-\mathbf{x}'\|^2}{2\sigma}}.$$

直观上，高斯核将特征空间中 $\mathbf{x}, \mathbf{x}'$ 之间的内积设置得接近零，如果实例彼此远离（在原始域中），如果它们彼此接近，则接近 1。σ 是一个控制尺度的参数，它决定了我们所说的“接近”的含义。很容易验证 K 在一个空间中实现了内积，在这个空间中，对于任何 n 和任何单项式

k 阶中存在一个 $\psi(\mathbf{x})$ 的元素等于$\frac{1}{\sqrt{n!}}$ e^{-} $\frac{\|\mathbf{x}\|^2}{2}\prod_{i=1}^{n} x_{J_i}$。因此，我们可以通过使用高斯核在原始空间上学习任何多项式预测器。

回忆一下，所有多项式预测器的VC维是无限的（参见练习12）。这并不矛盾，因为使用高斯核学习所需的样本复杂度取决于特征空间中的边界，如果我们幸运的话，这个边界会很大，但通常可以是任意小的。

高斯核也称为RBF核，即“径向基函数”。

16.2.1 Kernels as a Way to Express Prior Knowledge

如我们之前讨论的，一个特征映射ψ可以被视为将线性分类器的类别扩展到一个更丰富的类别（对应于特征空间上的线性分类器）。然而，正如书中至今所讨论的，任何假设类别对给定学习任务的适用性取决于该任务的本质。因此，可以将嵌入ψ视为表达和利用关于待解决问题先验知识的一种方式。例如，如果我们认为正例可以通过某个椭圆区分，我们可以定义ψ为所有最高阶为2的单项式，或者使用二次多项式核。

作为一个更现实的例子，考虑学习在文件中找到一个字符序列（“签名”）的任务，以指示它是否包含病毒。形式上，设$\mathcal{X}_d$为长度不超过d的某些字母表集合Σ中所有字符串的集合。一个人希望学习的假设类是$\mathcal{H} = \{h_v\colon\ v \in \mathcal{X}_d\}$，其中，对于字符串$x \in \mathcal{X}_d$，如果$v$是$x$ (和$h_v(x) = -$的子串，则$h_v(x)$为1，否则为)。让我们展示如何使用适当的嵌入，这个类可以通过结果特征空间上的线性分类器来实现。考虑一个映射ψ到空间$\mathbb{R}^s$，其中$s = |\mathcal{X}_d|$，使得$\psi(x)$的每个坐标对应于某个字符串v，并指示v是否是x (的子串，即对于每个$x \in \mathcal{X}_d$，$\psi(x)$是{中的向量,0}$^{|\mathcal{X}_d|}$)。请注意，这个特征空间的维度在d上是指数级的。不难看出，类$\mathcal{H}$的每个成员都可以通过在$\psi(x)$上组合线性分类器来实现，并且，此外，通过这样一个范数为1的半空间，它达到1的边界（见练习1）。此外，对于每个$x \in \mathcal{X}$，$\|\psi(x)\| = O(d)$。因此，总的来说，它可以通过使用SVM和样本来学习。

复杂度是关于 d 的多项式。然而，特征空间的维度是关于 d 的指数级，因此在特征空间上直接实现 SVM 是有问题的。幸运的是，可以在不将实例显式映射到特征空间的情况下轻松计算特征空间中的内积（即核函数）。实际上，$K(x, x')$ 是 x 和 x' 的公共子串的数量，这可以在关于 d 的多项式时间内轻松计算。

此示例还展示了特征映射如何使我们能够在非向量域中使用半空间。

16.2.2 Characterizing Kernel Functions*

如我们在上一节中讨论的那样，我们可以将核矩阵的指定视为表达先验知识的一种方式。考虑一个形式为 $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ 的给定相似度函数。这是一个有效的核函数吗？也就是说，它是否代表某些特征映射 ψ 之间的内积 $\psi(\mathbf{x})$ 和 $\psi(\mathbf{x}')$？以下引理给出了充分必要条件。

LEMMA 16.2 *A symmetric function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ implements an inner product in some Hilbert space if and only if it is positive semidefinite; namely, for all $\mathbf{x}_1, \ldots, \mathbf{x}_m$, the Gram matrix, $G_{i,j} = K(\mathbf{x}_i, \mathbf{x}_j)$, is a positive semidefinite matrix.*

Proof 它是显而易见的，如果 {v*} 在某个希尔伯特空间中实现内积，那么格拉姆矩阵是正半定的。对于另一个方向，定义在 $\mathcal{X}$ 上的函数空间为 $\mathbb{R}^{\mathcal{X}} = \{f: \mathcal{X} \to \mathbb{R}\}$。对于每个 $\mathbf{x} \in \mathcal{X}$，让 $\psi(\mathbf{x})$ 是函数 $\mathbf{x} \mapsto K(\cdot, \mathbf{x})$。通过取所有形式为 $K(\cdot, \mathbf{x})$ 的元素的线性组合来定义一个向量空间。定义这个向量空间上的内积为

$$\left\langle \sum_i \alpha_i K(\cdot, \mathbf{x}_i), \sum_j \beta_j K(\cdot, \mathbf{x}'_j) \right\rangle = \sum_{i,j} \alpha_i \beta_j K(\mathbf{x}_i, \mathbf{x}'_j).$$

这是一个有效的内积，因为它是对称的（因为 K 是对称的），它是线性的（直接的），并且它是正定的（很容易看出 $K(\mathbf{x}, \mathbf{x}) \geq 0$，只有当 $\psi(\mathbf{x})$ 是零函数时才取等号）。显然，

$$\langle \psi(\mathbf{x}), \psi(\mathbf{x}') \rangle = \langle K(\cdot, \mathbf{x}), K(\cdot, \mathbf{x}') \rangle = K(\mathbf{x}, \mathbf{x}'),$$

这总结了我们的证明。 □

16.3 Implementing Soft-SVM with Kernels

接下来，我们转向解决带有核的软SVM。虽然我们可以设计一个求解方程（16.4）的算法，但有一个更简单的方法，

直接解决特征空间中的软SVM优化问题，

$$\min_{\mathbf{w}} \left(\frac{\lambda}{2}\|\mathbf{w}\|^2 + \frac{1}{m}\sum_{i=1}^{m} \max\{0, 1 - y\langle \mathbf{w}, \psi(\mathbf{x}_i)\rangle\} \right), \tag{16.5}$$

当仅使用核评估时。基本观察是，我们在第15.5节中描述的SGD过程维护的向量$\mathbf{w}^{(t)}$始终位于$\{\psi(\mathbf{x}_1), \ldots, \psi(\mathbf{x}_m)\}$的线性跨度中。因此，我们不必维护$\mathbf{w}^{(t)}$，而是可以维护相应的系数$\boldsymbol{\alpha}$。

形式上，令K为核函数，即对于所有$\mathbf{x}, \mathbf{x}'$，$K(\mathbf{x}, \mathbf{x}') = \langle \psi(\mathbf{x}), \psi(\mathbf{x}')\rangle$。我们将在$\mathbb{R}^m$中维护两个向量，对应于第15.5节中SGD过程定义的两个向量$\boldsymbol{\theta}^{(t)}$和$\mathbf{w}^{(t)}$。也就是说，$\boldsymbol{\beta}^{(t)}$将是一个向量，使得

$$\boldsymbol{\theta}^{(t)} = \sum_{j=1}^{m} \beta_j^{(t)} \psi(\mathbf{x}_j) \tag{16.6}$$

并且$\boldsymbol{\alpha}^{(t)}$满足

$$\mathbf{w}^{(t)} = \sum_{j=1}^{m} \alpha_j^{(t)} \psi(\mathbf{x}_j). \tag{16.7}$$

向量$\boldsymbol{\beta}$和$\boldsymbol{\alpha}$按照以下程序进行更新。

SGD for Solving Soft-SVM with Kernels

Goal: Solve Equation (16.5)
parameter: T
Initialize: $\boldsymbol{\beta}^{(1)} = \mathbf{0}$
for $t = 1, \ldots, T$
  Let $\boldsymbol{\alpha}^{(t)} = \frac{1}{\lambda t}\boldsymbol{\beta}^{(t)}$
  Choose i uniformly at random from $[m]$
  For all $j \neq i$ set $\beta_j^{(t+1)} = \beta_j^{(t)}$
  If $(y_i \sum_{j=1}^{m} \alpha_j^{(t)} K(\mathbf{x}_j, \mathbf{x}_i) < 1)$
    Set $\beta_i^{(t+1)} = \beta_i^{(t)} + y_i$
  Else
    Set $\beta_i^{(t+1)} = \beta_i^{(t)}$
Output: $\bar{\mathbf{w}} = \sum_{j=1}^{m} \bar{\alpha}_j \psi(\mathbf{x}_j)$ where $\bar{\boldsymbol{\alpha}} = \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{\alpha}^{(t)}$

以下引理表明，前面的实现等价于在特征空间上运行第15.5节中描述的SGD过程。

LEMMA 16.3 *Let $\hat{\mathbf{w}}$ be the output of the SGD procedure described in Section 15.5, when applied on the feature space, and let $\bar{\mathbf{w}} = \sum_{j=1}^{m} \bar{\alpha}_j \psi(\mathbf{x}_j)$ be the output of applying SGD with kernels. Then $\bar{\mathbf{w}} = \hat{\mathbf{w}}$.*

Proof 我们将证明对于每个t方程（16.6）成立，其中$\boldsymbol{\theta}^{(t)}$是在特征中运行第15.5节所述的SGD过程的结果。

空间。根据 $\boldsymbol{\alpha}^{(t)} = \frac{1}{\lambda t}\boldsymbol{\beta}^{(t)}$ 和 $\mathbf{w}^{(t)} = \frac{1}{\lambda t}\boldsymbol{\theta}^{(t)}$ 的定义，这个命题意味着方程（16.7）也成立，我们的引理的证明将随后进行。为了证明方程（16.6）成立，我们使用一个简单的归纳论证。对于 $t = 1$，这个命题显然成立。假设它对 $t \geq 1$ 成立。那么，

$$y_i \left\langle \mathbf{w}^{(t)}, \psi(\mathbf{x}_i) \right\rangle = y_i \left\langle \sum_j \alpha_j^{(t)} \psi(\mathbf{x}_j), \psi(\mathbf{x}_i) \right\rangle = y_i \sum_{j=1}^{m} \alpha_j^{(t)} K(\mathbf{x}_j, \mathbf{x}_i).$$

因此，两个算法中的条件是等价的，如果我们更新 $\boldsymbol{\theta}$，则有

$$\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} + y_i \psi(\mathbf{x}_i) = \sum_{j=1}^{m} \beta_j^{(t)} \psi(\mathbf{x}_j) + y_i \psi(\mathbf{x}_i) = \sum_{j=1}^{m} \beta_j^{(t+1)} \psi(\mathbf{x}_j),$$

这总结了我们的证明。 □

16.4 Summary

从给定的域到某些高维空间的映射，在该空间上使用半空间预测器，可以非常强大。我们受益于一个丰富且复杂的假设类，但需要解决高样本和计算复杂性的问题。在第10章中，我们讨论了AdaBoost算法，该算法通过使用弱学习器来应对这些挑战：即使我们处于一个非常高维的空间中，我们也有一个“先知”，在每个迭代中赋予我们一个最佳的坐标来工作。在本章中，我们介绍了一种不同的方法，即核技巧。其想法是在高维空间中找到一个半空间预测器，我们不需要知道该空间中实例的表示，而是需要知道映射实例之间内积的值。在不使用该空间中实例的表示的情况下，通过核函数计算高维空间中实例之间的内积。我们还展示了如何使用核函数实现SGD算法。

特征映射和核技巧的理念使我们能够将半空间和线性预测器的框架应用于非向量数据。我们展示了如何使用核在字符串域上学习预测器。

我们介绍了核技巧在SVM中的应用。然而，核技巧可以应用于许多其他算法。一些例子作为练习给出。

本章结束了对线性预测和凸问题的章节系列。接下来的两章将处理完全不同类型的假设类。

16.5 Bibliographic Remarks

在SVM的背景下，核技巧在Boser等人（1992年）中被引入。参见Aizerman，Braverman和Rozonoer（1964年）。Scholkopf，Smola和Müller（1998年）首先提出了当算法仅依赖于内积时可以应用核技巧的观察。代表定理的证明在（Scholkopf，Herbrich，Smola和Williamson 2000年，Scholkopf，Herbrich和Smola 2001年）中给出。第16.2条引理中所述的条件是Mercer条件的简化。文献中已经为各种应用引入了许多有用的核函数。我们建议读者参考Scholkopf和Smola（2002年）。

16.6 Exercises

1. 考虑在文件中查找字符序列的任务，如第16.2.1节所述。证明类$\mathcal{H}$的每个成员都可以通过在$\psi(x)$上组合一个线性分类器来实现，其范数为1且达到1的间隔。

2. **Kernelized Perceptron:** 展示如何仅通过核函数访问实例来运行感知器算法。*Hint:* 推导过程与实现带有核的SGD的推导类似。

3. **Kernel Ridge Regression:** 岭回归问题，具有特征映射 ψ，是寻找一个向量 $\mathbf{w}$ 以最小化函数的问题

$$f(\mathbf{w}) = \lambda \|\mathbf{w}\|^2 + \frac{1}{2m}\sum_{i=1}^{m}(\langle \mathbf{w}, \psi(\mathbf{x}_i)\rangle - y_i)^2, \tag{16.8}$$

然后返回预测器

$$h(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x}\rangle.$$

显示 如何使用k实现岭回归算法 内核。

Hint: 表示定理告诉我们，存在一个向量 $\boldsymbol{\alpha} \in \mathbb{R}^m$，使得 $\sum_{i=1}^{m}\alpha_i\psi(\mathbf{x}_i)$ 是方程（16.8）的最小化值。

1. 令 G 为关于 S 和 K 的 Gram 矩阵。即，$G_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$。定义 g : $\mathbb{R}^m \to \mathbb{R}$ 为

$$g(\boldsymbol{\alpha}) = \lambda \cdot \boldsymbol{\alpha}^T G\boldsymbol{\alpha} + \frac{1}{2m}\sum_{i=1}^{m}(\langle \boldsymbol{\alpha}, G_{\cdot,i}\rangle - y_i)^2, \tag{16.9}$$

在 $G_{\cdot,i}$ 是 G 的 i 列的情况下。证明如果 $\boldsymbol{\alpha}^*$ 最小化方程 (16.9)，则 $\mathbf{w}^* = \sum_{i=1}^{m}\alpha_i^*\psi(\mathbf{x}_i)$ 是 f 的最小值点。

2. 找到 $\boldsymbol{\alpha}^*$ 的闭式表达式。

4. 设 N 为任意正整数。对于每一个 $x, x' \in \{1, \ldots, N\}$ 定义

$$K(x, x') = \min\{x, x'\}.$$

证明 K 是一个有效的核；即找到一个映射 $\psi : \{1, \ldots, N\} \to H$，其中 H 是某个希尔伯特空间，使得

$$\forall x, x' \in \{1, \ldots, N\},\ K(x, x') = \langle \psi(x), \psi(x') \rangle.$$

5. 一位超市经理希望根据购物车了解哪些顾客有婴儿。具体来说，他抽取了独立同分布的顾客样本，对于顾客 i，让 $x_i \subset \{1, \ldots, d\}$ 表示该顾客购买的商品子集，让 $y_i \in \{\pm 1\}$ 表示该顾客是否有婴儿的标签。作为先验知识，经理知道有 k 件商品，如果顾客至少购买其中 k 件，则标签被确定为 1。当然，这些 k 件商品的身份是未知的（否则就没有什么可学习的）。此外，根据商店规定，每位顾客最多可以购买 s 件商品。帮助经理设计一个学习算法，使其时间复杂度和样本复杂度都是 s, k 和 $1/\epsilon$ 的多项式。

6. 令 $\mathcal{X}$ 为一个实例集，令 ψ 为将 $\mathcal{X}$ 映射到某个希尔伯特特征空间 V 的特征映射。令 $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ 为在特征空间 V 中实现内积的核函数。

考虑二元分类算法，该算法根据最接近的平均值预测未见实例的标签。形式上，给定训练序列 $S = (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$，对于每个 $y \in \{\pm 1\}$，我们定义

$$c_y = \frac{1}{m_y} \sum_{i : y_i = y} \psi(\mathbf{x}_i).$$

在 $m_y = |\{i : y_i = y\}|$。我们假设 m_+ 和 m_- 都不为零。然后，算法输出以下决策规则：

$$h(\mathbf{x}) = \begin{cases} 1 & \|\psi(\mathbf{x}) - c_+\| \leq \|\psi(\mathbf{x}) - c_-\| \\ 0 & \text{otherwise.} \end{cases}$$

1. 设 $\mathbf{w} = c_+ - c_-$ 和设 $b = \frac{1}{2}(\|c_-\|^2 - \|c_+\|^2)$。证明

$$h(\mathbf{x}) = \text{sign}(\langle \mathbf{w}, \psi(\mathbf{x}) \rangle + b).$$

2. 展示如何在核函数的基础上表达 $h(\mathbf{x})$，且不访问 $\psi(\mathbf{x})$ 或 $\mathbf{w}$ 的单个条目。

17 Multiclass, Ranking, and Complex Prediction Problems

多类分类是将实例分类到多个可能的目标类别之一的问题。也就是说，我们的目标是学习一个预测器 $h : \mathcal{X} \to \mathcal{Y}$，其中 $\mathcal{Y}$ 是一个有限类别集。应用包括，例如，根据主题对文档进行分类（$\mathcal{X}$ 是文档集，$\mathcal{Y}$ 是可能的主题集）或确定给定图像中出现了哪个对象（$\mathcal{X}$ 是图像集，$\mathcal{Y}$ 是可能的对象集）。

多类学习问题的核心地位推动了各种解决该任务的途径的发展。可能最直接的方法是将多类分类问题简化为二分类问题。在第17.1节中，我们讨论了最常见的两种简化方法以及简化方法的主要缺点。

我们接着描述一个用于多类问题的线性预测器族。依赖于前几章中的RLM和SGD框架，我们描述了几个用于多类预测的实用算法。

在17.3节中，我们展示了如何使用多类机器来解决复杂预测问题，其中$\mathcal{Y}$可以非常大但具有某种结构。这项任务通常被称为*structured output learning*。特别是，我们展示了这种方法在识别手写单词的任务中的应用，其中$\mathcal{Y}$是所有可能长度有限（因此，$\mathcal{Y}$的大小与单词最大长度呈指数关系）的字符串的集合。

最后，在第17.4节和第17.5节中，我们讨论了排名问题，其中学习器应按照“相关性”对一组实例进行排序。一个典型应用是根据查询的相关性对搜索引擎的结果进行排序。我们描述了几个适合评估排名预测器性能的性能指标，并描述了如何有效地学习排名问题的线性预测器。

17.1 One-versus-All and All-Pairs

最简单的解决多类预测问题的方法是将其简化为二分类问题。回想一下，在多类预测中，我们希望学习一个函数 $h : \mathcal{X} \to \mathcal{Y}$。不失一般性，让我们用 $\mathcal{Y} = \{1, \ldots, k\}$ 表示。

在One-versus-All方法（也称为One-versus-Rest）中，我们训练k二分类器

数字，每个数字都区分一个类别和其余类别。也就是说，给定一个训练集 $S = (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$，其中每个 y_i 都在 $\mathcal{Y}$ 中，我们构建 k 个二进制训练集 $S_1, \ldots, S_k$，其中 $S_i = (\mathbf{x}_1, (-1)^{\mathbb{1}_{[y_1 \neq i]}}), \ldots, (\mathbf{x}_m, (-1)^{\mathbb{1}_{[y_m \neq i]}})$。换句话说，$S_i$ 是一个集合，如果其标签在 S 中为 i，则标记为 1，否则为 -1。对于每个 $i \in [k]$，我们基于 S_i 训练一个二进制预测器 $h_i : \mathcal{X} \to \{\pm 1\}$，希望 $h_i(\mathbf{x})$ 等于 1 当且仅当 $\mathbf{x}$ 属于类别 i。然后，给定 $h_1, \ldots, h_k$，我们使用以下规则构建一个多类别预测器

$$h(\mathbf{x}) \in \operatorname*{argmax}_{i \in [k]} h_i(\mathbf{x}). \tag{17.1}$$

当多个二进制假设预测“1”时，我们应该以某种方式决定预测哪个类别（例如，我们可以任意决定通过取$\operatorname{argmax}_i h_i(\mathbf{x})$的最小索引来打破平局）。当每个$h_i$都隐藏额外的信息时，可以将其解释为预测的置信度$y = i$，可以采用更好的方法。例如，在半空间中就是这样，实际的预测是$\operatorname{sign}(\langle \mathbf{w}, \mathbf{x} \rangle)$，但我们可以将$\langle \mathbf{w}, \mathbf{x} \rangle$解释为预测的置信度。在这种情况下，我们可以将方程（17.1）中给出的多类规则应用于实值预测。以下给出One-versus-All方法的伪代码。

One-versus-All

input:
  training set $S = (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$
  algorithm for binary classification A
foreach $i \in \mathcal{Y}$
  let $S_i = (\mathbf{x}_1, (-1)^{\mathbb{1}_{[y_1 \neq i]}}), \ldots, (\mathbf{x}_m, (-1)^{\mathbb{1}_{[y_m \neq i]}})$
  let $h_i = A(S_i)$
output:
  the multiclass hypothesis defined by $h(\mathbf{x}) \in \operatorname{argmax}_{i \in \mathcal{Y}} h_i(\mathbf{x})$

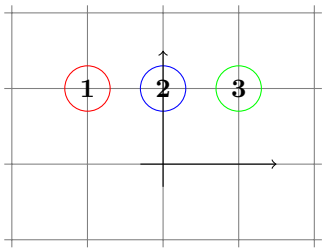
另一种流行的降维方法是 *All-Pairs* 方法，其中所有类别的成对比较。形式上，给定一个训练集 $S = (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$，其中每个 y_i 都在 $[k]$ 中，对于每个 $1 \leq i < j \leq k$，我们构建一个包含所有来自 S 的示例的二进制训练序列 $S_{i,j}$，这些示例的标签是 i 或 j。对于这样的每个示例，我们将 $S_{i,j}$ 中的二进制标签设置为 $+1$，如果多类标签 S 是 i，以及 -1，如果多类标签 S 是 j。接下来，我们基于每个 $S_{i,j}$ 训练一个二进制分类算法以获得 $h_{i,j}$。最后，我们通过预测具有最高“胜利”次数的类别来构建一个多类分类器。以下给出了 All-Pairs 方法的伪代码。

All-Pairs

input:
training set $S = (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$
algorithm for binary classification A
foreach $i, j \in \mathcal{Y}$ s.t. $i < j$
initialize $S_{i,j}$ to be the empty sequence
for $t = 1, \ldots, m$
If $y_t = i$ add $(\mathbf{x}_t, 1)$ to $S_{i,j}$
If $y_t = j$ add $(\mathbf{x}_t, -1)$ to $S_{i,j}$
let $h_{i,j} = A(S_{i,j})$
output:
the multiclass hypothesis defined by
$h(\mathbf{x}) \in \operatorname{argmax}_{i \in \mathcal{Y}} \left(\sum_{j \in \mathcal{Y}} \operatorname{sign}(j - i)\, h_{i,j}(\mathbf{x}) \right)$

尽管One-versus-All和All-Pairs等降维方法简单且易于从现有算法构建，但它们的简单性是有代价的。二元学习器没有意识到我们将使用其输出假设来构建多类预测器，这可能导致次优结果，如下例所示。

Example 17.1 考虑一个多类分类问题，其中实例空间是 $\mathcal{X} = \mathbb{R}^2$，标签集是 $\mathcal{Y} = \{1, 2, 3\}$。假设不同类别的实例位于非相交的球中，如下所示。



假设类别1, 2, 3的概率质量分别为40%, 20%, 和40%，分别考虑One-versus-All对此问题的应用，并假设One-versus-All使用的二分类算法是对半空间假设类别的ERM。观察发现，对于区分类别2与其他类别的判别问题，最优半空间将是所有负类分类器。因此，One-versus-All构建的多类预测器可能会在类别2的所有示例上出错（如果$h(\mathbf{x})$的定义中的平局被类别标签的数值打破，则这种情况会发生）。相比之下，如果我们选择$h_i(\mathbf{x}) = \langle \mathbf{w}_i, \mathbf{x} \rangle$，其中$\mathbf{w}_1 = \left(-\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}} \right)$，$\mathbf{w}_2 = (0, 1)$，和$\mathbf{w}_3 = \left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}} \right)$，那么由$h(\mathbf{x}) = \operatorname{argmax}_i h_i(\mathbf{x})$定义的分类器可以完美预测所有示例。我们看到

尽管形式为 $h(\mathbf{x}) = \text{argmax}_i \langle \mathbf{w}_i, \mathbf{x} \rangle$ 的预测器类别的近似误差为零，但 One-versus-All 方法可能无法从该类别中找到一个好的预测器。

17.2 Linear Multiclass Predictors

鉴于降维方法的不充分性，在本节中，我们研究了一种更直接的学习多类预测器的方法。我们描述了线性多类预测器的家族。为了激发这个家族的构建，回忆一下，二分类的线性预测器（即半空间）的形式为

$$h(\mathbf{x}) \;=\; \text{sign}(\langle \mathbf{w}, \mathbf{x} \rangle).$$

一个表达预测的等效方式如下：

$$h(\mathbf{x}) \;=\; \operatorname*{argmax}_{y \in \{\pm 1\}} \langle \mathbf{w}, y\mathbf{x} \rangle,$$

在 $\mathbf{x}$ 的每个元素乘以 y 后得到的向量 $y\mathbf{x}$。

这种表示自然地推广了半空间到多类问题，如下所示。令 $\Psi : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^d$ 为一个 *class-sensitive feature mapping*。也就是说，Ψ 以一对 $(\mathbf{x}, y)$ 作为输入并将其映射到一个 d 维特征向量。直观上，我们可以将 $\Psi(\mathbf{x}, y)$ 的元素视为评分函数，评估标签 y 与实例 $\mathbf{x}$ 的匹配程度。我们将在稍后详细说明 Ψ。给定 Ψ 和一个向量 $\mathbf{w} \in \mathbb{R}^d$，我们可以定义一个多类预测器，$h : \mathcal{X} \to \mathcal{Y}$，如下所示：

$$h(\mathbf{x}) \;=\; \operatorname*{argmax}_{y \in \mathcal{Y}} \langle \mathbf{w}, \Psi(\mathbf{x}, y) \rangle.$$

这是指，对于输入 $\mathbf{x}$ 的 h 的预测是获得最高加权分数的标签，其中加权是根据向量 $\mathbf{w}$ 进行的。

让 W 是 $\mathbb{R}^d$ 中的一些向量集，例如，$W = \{\mathbf{w} \in \mathbb{R}^d : \|\mathbf{w}\| \le B\}$，对于某个标量 $B > 0$。每一对 (Ψ, W) 定义了一个多类预测器的假设类：

$$\mathcal{H}_{\Psi,W} = \{\mathbf{x} \mapsto \operatorname*{argmax}_{y \in \mathcal{Y}} \langle \mathbf{w}, \Psi(\mathbf{x}, y) \rangle \;:\; \mathbf{w} \in W\}.$$

当然，我们将在后续讨论的立即问题是如何构建一个良好的 Ψ。请注意，如果 $\mathcal{Y} = \{\pm 1\}$ 并且我们设置 $\Psi(\mathbf{x}, y) = y\mathbf{x}$ 和 $W = \mathbb{R}^d$，那么 $\mathcal{H}_{\Psi,W}$ 就成为二元分类的均匀半空间预测器的假设类。

17.2.1 How to Construct Ψ

如前所述，我们可以将 $\Psi(\mathbf{x}, y)$ 的元素视为评估标签 y 与实例 $\mathbf{x}$ 匹配程度的得分函数。自然地，设计一个好的 Ψ 与设计一个好的特征映射问题类似（正如我们在第

第16章，我们将在第25章中更详细地讨论）。以下给出了两个有用的构造示例。

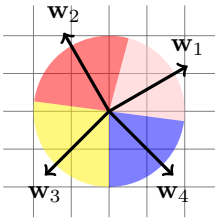
The Multivector Construction:
让我们 $\mathcal{Y} = \{1, \ldots, k\}$，并让 $\mathcal{X} = \mathbb{R}^n$。我们定义 $\Psi : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^d$，如下所示，其中 $d = nk$

$$\Psi(\mathbf{x},\ y) = [\ \underbrace{0, \ldots, 0}_{\in \mathbb{R}^{(y-1)n}}, \underbrace{x_1, \ldots, x_n}_{\in \mathbb{R}^n}, \underbrace{0, \ldots, 0}_{\in \mathbb{R}^{(k-y)n}}\ ]. \tag{17.2}$$

这是，$\Psi(\mathbf{x}, y)$ 由 k 个向量组成，每个向量的维度为 n，我们将所有向量设置为全零向量，除了第 y 个向量，它被设置为 $\mathbf{x}$。因此，我们可以将 $\mathbf{w} \in \mathbb{R}^{nk}$ 视为由 $\mathbb{R}^n$ 中的 k 个权重向量组成，即 $\mathbf{w} = [\mathbf{w}_1; \ldots; \mathbf{w}_k]$，因此得名 *multivector construction*。根据构造，我们有 $\langle \mathbf{w}, \Psi(\mathbf{x}, y) \rangle = \langle \mathbf{w}_y, \mathbf{x} \rangle$，因此多类预测变为

$$h(\mathbf{x}) = \operatorname*{argmax}_{y \in \mathcal{Y}} \ \langle \mathbf{w}_y, \mathbf{x} \rangle.$$

一个关于在 $\mathcal{X} = \mathbb{R}^2$ 上进行多类预测的几何说明如下所示。



TF-IDF:
前一个 $\Psi(\mathbf{x},\ y)$ 的定义没有结合任何关于问题的先验知识。接下来，我们描述一个特征函数 Ψ 的例子，该函数确实结合了先验知识。设 $\mathcal{X}$ 为一组文本文档，$\mathcal{Y}$ 为一组可能的主题。设 d 为单词字典的大小。对于字典中的每个单词，其对应的索引为 j，设 $TF(j, \mathbf{x})$ 为对应于 j 的单词在文档 $\mathbf{x}$ 中出现的次数。这个数量称为词频。此外，设 $DF(j, y)$ 为对应于 j 的单词在训练集中关于主题 y 的文档中出现的次数。这个数量称为文档频率，它衡量单词 j 在其他主题中是否频繁。现在，定义 Ψ：$\mathcal{X} \times \mathcal{Y} \to \mathbb{R}^d$ 为满足

$$\Psi_j(\mathbf{x}, y) = TF(j, \mathbf{x}) \log\left(\frac{m}{DF(j, y)}\right),$$

m 是我们训练集中文档的总数。前面的量被称为词频-逆文档频率或 TF-IDF。

简短地。直观上，如果与j对应的单词在文档$\mathbf{x}$中频繁出现，但在非主题文档y中完全不出现，那么$\Psi_j(\mathbf{x}, y)$应该很大。在这种情况下，我们倾向于认为文档$\mathbf{x}$是关于主题y的。请注意，与之前描述的多矢量构造不同，在当前构造中，Ψ的维度不依赖于主题数量（即$\mathcal{Y}$的大小）。

17.2.2 Cost-Sensitive Classification

截至目前，我们使用零一损失作为衡量 $h(\mathbf{x})$ 质量的性能指标。也就是说，假设 h 在一个例子 $(\mathbf{x}, y)$ 上的损失是，如果 $h(\mathbf{x}) \neq y$ 则为 1，否则为 0。在某些情况下，对不同级别的错误进行不同程度的惩罚更有意义。例如，在物体识别任务中，预测一张老虎的图片包含猫比预测包含鲸鱼要轻微得多。这可以通过指定一个损失函数 $\Delta : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$ 来建模，其中对于每一对标签 y', y，当正确标签是 y 时，预测标签 y' 的损失被定义为 $\Delta(y', y)$。我们假设 $\Delta(y, y) = 0$。请注意，可以通过将 $\Delta(y', y) =$ 设置为 $1_{[y' \neq y]}$ 来轻松地模拟零一损失。

17.2.3 ERM

我们已经定义了假设类 $\mathcal{H}_{\Psi,W}$ 并指定了一个损失函数 Δ。为了根据损失函数学习该类，我们可以应用关于此类的 ERM 规则。也就是说，我们寻找一个由向量 $\mathbf{w}$ 参数化的多类假设 $h \in \mathcal{H}_{\Psi,W}$，以最小化相对于 Δ 的经验风险。

$$L_S(h) = \frac{1}{m} \sum_{i=1}^{m} \Delta(h(\mathbf{x}_i), y_i).$$

我们现在表明，当 $W = \mathbb{R}^d$ 和我们处于可实现情况时，可以使用线性规划有效地解决 ERM 问题。实际上，在可实现情况下，我们需要找到一个向量 $\mathbf{w} \in \mathbb{R}^d$，它满足

$$\forall i \in [m], \quad y_i = \operatorname*{argmax}_{y \in \mathcal{Y}} \langle \mathbf{w}, \Psi(\mathbf{x}_i, y) \rangle.$$

等效地，我们需要 $\mathbf{w}$ 满足以下一组线性不等式

$$\forall i \in [m],\ \forall y \in \mathcal{Y} \setminus \{y_i\},\ \ \langle \mathbf{w}, \Psi(\mathbf{x}_i, y_i) \rangle > \langle \mathbf{w}, \Psi(\mathbf{x}_i, y) \rangle.$$

找到满足前面一组线性方程的 $\mathbf{w}$ 等于解决一个线性规划问题。

与二元分类的情况一样，也可以使用感知器算法的推广来解决ERM问题。参见练习2。

在不可行情况下，解决ERM问题通常计算上很困难。我们使用凸代理法来解决这个问题。

损失函数（见第12.3节）。特别是，我们将铰链损失推广到多类问题。

17.2.4 Generalized Hinge Loss

回忆在二分类中，铰链损失被定义为 $\max\{0, 1 - y\langle \mathbf{w}, \mathbf{x}\rangle\}$。我们现在将铰链损失推广到形式为的多类预测器

$$h_{\mathbf{w}}(\mathbf{x}) = \operatorname*{argmax}_{y' \in \mathcal{Y}} \langle \mathbf{w}, \Psi(\mathbf{x}, \mathbf{y}')\rangle.$$

回忆一下，一个代理凸损失应该上界原始非凸损失，在我们的情况下是 $\Delta(h_{\mathbf{w}}(\mathbf{x}), y)$。为了推导 $\Delta(h_{\mathbf{w}}(\mathbf{x}), y)$ 的上界，我们首先注意到 $h_{\mathbf{w}}(\mathbf{x})$ 的定义意味着

$$\langle \mathbf{w}, \Psi(\mathbf{x}, y)\rangle \leq \langle \mathbf{w}, \Psi(\mathbf{x}, h_{\mathbf{w}}(\mathbf{x}))\rangle.$$

因此，

$$\Delta(h_{\mathbf{w}}(\mathbf{x}), y) \;\leq\; \Delta(h_{\mathbf{w}}(\mathbf{x}), y) + \langle \mathbf{w}, \Psi(\mathbf{x}, h_{\mathbf{w}}(\mathbf{x})) - \Psi(\mathbf{x}, y)\rangle.$$

自 $h_{\mathbf{w}}(\mathbf{x}) \in \mathcal{Y}$ 以来，我们可以对前面的右侧进行上界估计

$$\max_{y' \in \mathcal{Y}} \left(\Delta(y', y) + \langle \mathbf{w}, \Psi(\mathbf{x}, y') - \Psi(\mathbf{x}, y)\rangle\right) \;\stackrel{\text{def}}{=}\; \ell(\mathbf{w}, (\mathbf{x}, y)). \tag{17.3}$$

我们使用术语“广义铰链损失”来表示前面的表达式。正如我们所展示的，$\ell(\mathbf{w}, (\mathbf{x}, y)) \geq \Delta(h_{\mathbf{w}}(\mathbf{x}), y)$。此外，当正确标签的分数大于任何其他标签的分数时，等式成立，y'，至少为$\Delta(y', y)$，即，

$$\forall y' \in \mathcal{Y} \setminus \{y\}, \quad \langle \mathbf{w}, \Psi(\mathbf{x}, \mathbf{y})\rangle \geq \langle \mathbf{w}, \Psi(\mathbf{x}, \mathbf{y}')\rangle + \Delta(y', y).$$

它也是显而易见的，$\ell(\mathbf{w}, (\mathbf{x}, y))$相对于$\mathbf{w}$是一个凸函数，因为它是在$\mathbf{w}$ (的线性函数上的最大值，参见第12章的12.5命题)，并且$\ell(\mathbf{w}, (\mathbf{x}, y))$相对于$\rho =$的最大值是$\rho$-Lipschitz。

Remark 17.2 我们使用“广义铰链损失”这个名称，因为在二元情况下，当 $\mathcal{Y} = \{\pm 1\}$，如果我们设置 $\Psi(\mathbf{x}, y) = \frac{y\mathbf{x}}{2}$，那么广义铰链损失就变成了二元分类的普通铰链损失。

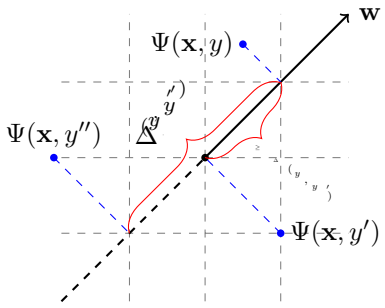
$$\ell(\mathbf{w}, (\mathbf{x}, y)) = \max\{0, 1 - y\langle \mathbf{w}, \mathbf{x}\rangle\}.$$

Geometric Intuition:
特征函数 $\Psi : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^d$ 将每个 $\mathbf{x}$ 映射到 $\mathbb{R}^d$ 中的 $|\mathcal{Y}|$ 向量。如果存在一个方向 $\mathbf{w}$，使得将 $|\mathcal{Y}|$ 向量投影到这个方向时，我们得到每个向量都表示为标量 $\langle \mathbf{w}, \Psi(\mathbf{x}, y)\rangle$，那么 $\ell(\mathbf{w}, (\mathbf{x}, y))$ 的值将为零，并且我们可以根据这些标量对这些不同的点进行排序，以便

- 该点对应正确的 y 是排名第一

- 对于每个 $y' \neq y$，$\langle \mathbf{w}, \Psi(\mathbf{x}, y) \rangle$ 和 $\langle \mathbf{w}, \Psi(\mathbf{x}, y') \rangle$ 之间的差异大于预测 y' 而不是 y 的损失。差异 $\langle \mathbf{w}, \Psi(\mathbf{x}, y) \rangle - \langle \mathbf{w}, \Psi(\mathbf{x}, y') \rangle$ 也被称为“边界”（见第15.1节）。

这是以下图示所说明的：



17.2.5 Multiclass SVM and SGD

一旦我们定义了广义铰链损失，我们就得到了一个凸-Lipschitz学习问题，并且我们可以应用我们解决此类问题的通用技术。特别是，我们在第13章中研究的RLM技术产生了多类SVM规则：

Multiclass SVM

input: $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$
parameters:
regularization parameter $\lambda > 0$
loss function $\Delta : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$
class-sensitive feature mapping $\Psi : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^d$
solve:

$$\min_{\mathbf{w} \in \mathbb{R}^d} \left(\lambda \|\mathbf{w}\|^2 + \frac{1}{m} \sum_{i=1}^{m} \max_{y' \in \mathcal{Y}} \left(\Delta(y', y_i) + \langle \mathbf{w}, \Psi(\mathbf{x}_i, y') - \Psi(\mathbf{x}_i, y_i) \rangle \right) \right)$$

output the predictor $h_{\mathbf{w}}(\mathbf{x}) = \operatorname{argmax}_{y \in \mathcal{Y}} \langle \mathbf{w}, \Psi(\mathbf{x}, y) \rangle$

我们可以使用通用的凸优化算法（或使用第15.5节中描述的方法）来解决与多类SVM相关的优化问题。让我们分析由此产生的假设的风险。分析无缝地遵循第13章中给出的凸-Lipschitz问题的通用分析。特别是，应用推论13.8并使用广义的hinge损失上界Δ损失的事实，我们立即获得推论15.7的类似物：

COROLLARY 17.1 *Let $\mathcal{D}$ be a distribution over $\mathcal{X} \times \mathcal{Y}$, let $\Psi : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^d$, and assume that for all $\mathbf{x} \in \mathcal{X}$ and $y \in \mathcal{Y}$ we have $\|\Psi(\mathbf{x}, y)\| \leq \rho/2$. Let $B > 0$.*

Consider running Multiclass SVM with $\lambda = \sqrt{\frac{2\rho^2}{B^2 m}}$ on a training set $S \sim \mathcal{D}^m$ and let $h_{\mathbf{w}}$ be the output of Multiclass SVM. Then,

$$\mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}[L^{\Delta}_{\mathcal{D}}(h_{\mathbf{w}})] \;\le\; \mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}[L^{\text{g-hinge}}_{\mathcal{D}}(\mathbf{w})] \;\le\; \min_{\mathbf{u}:\|\mathbf{u}\|\le B} L^{\text{g-hinge}}_{\mathcal{D}}(\mathbf{u}) + \sqrt{\frac{8\rho^2 B^2}{m}},$$

where $L^{\Delta}_{\mathcal{D}}(h) = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}[\Delta(h(\mathbf{x}), y)]$ and $L^{\text{g-hinge}}_{\mathcal{D}}(\mathbf{w}) = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}[\ell(\mathbf{w}, (\mathbf{x}, y))]$ with ℓ being the generalized hinge-loss as defined in Equation (17.3).

我们还可以应用第14章中描述的SGD学习框架来最小化$L^{\text{g-hinge}}_{\mathcal{D}}(\mathbf{w})$。回忆14.6命题，它处理了max函数的子梯度。鉴于这个命题，为了找到广义 hinge 损失的子梯度，我们只需要找到在广义 hinge 损失定义中达到最大值的$y \in \mathcal{Y}$。这产生了以下算法：

SGD for Multiclass Learning

parameters:
Scalar $\eta > 0$, integer $T > 0$
loss function $\Delta : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+$
class-sensitive feature mapping $\Psi : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^d$
initialize: $\mathbf{w}^{(1)} = \mathbf{0} \in \mathbb{R}^d$
for $t = 1, 2, \ldots, T$
sample $(\mathbf{x}, y) \sim \mathcal{D}$
find $\hat{y} \in \operatorname{argmax}_{y' \in \mathcal{Y}} \left(\Delta(y', y) + \langle \mathbf{w}^{(t)}, \Psi(\mathbf{x}, y') - \Psi(\mathbf{x}, y)\rangle\right)$
set $\mathbf{v}_t = \Psi(\mathbf{x}, \hat{y}) - \Psi(\mathbf{x}, y)$
update $\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta \mathbf{v}_t$
output $\bar{\mathbf{w}} = \frac{1}{T}\sum_{t=1}^{T} \mathbf{w}^{(t)}$

我们的关于SGD的一般分析，如第14.12引理所示，立即意味着：

COROLLARY 17.2 *Let $\mathcal{D}$ be a distribution over $\mathcal{X} \times \mathcal{Y}$, let $\Psi : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^d$, and assume that for all $\mathbf{x} \in \mathcal{X}$ and $y \in \mathcal{Y}$ we have $\|\Psi(\mathbf{x}, y)\| \le \rho/2$. Let $B > 0$. Then, for every $\epsilon > 0$, if we run SGD for multiclass learning with a number of iterations (i.e., number of examples)*

$$T \ge \frac{B^2\rho^2}{\epsilon^2}$$

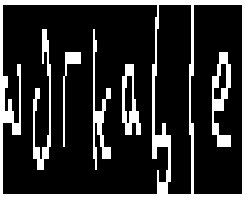
and with $\eta = \sqrt{\frac{B^2}{T\rho^2}}$ then the output of SGD satisfies

$$\mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}[L^{\Delta}_{\mathcal{D}}(h_{\bar{\mathbf{w}}})] \;\le\; \mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}[L^{\text{g-hinge}}_{\mathcal{D}}(\bar{\mathbf{w}})] \;\le\; \min_{\mathbf{u}:\|\mathbf{u}\|\le B} L^{\text{g-hinge}}_{\mathcal{D}}(\mathbf{u}) + \epsilon.$$

Remark 17.3 有趣的是，注意到第17.1引理和第17.2引理中给出的风险界限不明确依赖于标签集$\mathcal{Y}$的大小，这一点我们将在下一节中依赖。然而，这些界限可能通过$\Psi(\mathbf{x}, y)$的范数以及界限仅在存在某些向量$\mathbf{u}$、$\|\mathbf{u}\| \le B$，使得$L^{\text{g-hinge}}_{\mathcal{D}}(\mathbf{u})$不是过大时才有意义的方式，隐式地依赖于$\mathcal{Y}$的大小。

17.3 Structured Output Prediction

结构化输出预测问题是多类问题，其中 $\mathcal{Y}$ 非常大，但具有预定义的结构。结构在构建有效算法中起着关键作用。为了激发结构化学习问题，考虑光学字符识别（OCR）问题。假设我们收到一张某些手写单词的图像，并希望预测图像中写的是哪个单词。为了简化设置，假设我们知道如何将图像分割成一系列图像，每个图像包含与单个字母对应的图像块。因此，$\mathcal{X}$ 是图像序列的集合，$\mathcal{Y}$ 是字母序列的集合。请注意，$\mathcal{Y}$ 的大小随着单词最大长度的指数增长。以下给出了与标签 $\mathbf{y} =$ “workable” 对应的图像 $\mathbf{x}$ 的一个示例。



为了处理结构预测，我们可以依赖于上一节中描述的线性预测器族。特别是，我们需要为问题定义一个合理的损失函数 Δ，以及一个良好的类敏感特征映射 Ψ。这里的“良好”是指一个特征映射，它将导致对 Ψ 和 Δ 的线性预测器类具有低近似误差。一旦我们这样做，我们就可以依赖例如上一节中定义的 SGD 学习算法。

然而，$\mathcal{Y}$的巨大尺寸带来了几个挑战：

1. 要应用多类预测，我们需要在 $\mathcal{Y}$ 上解决一个最大化问题。当 $\mathcal{Y}$ 如此之大时，我们如何高效地预测？ 2. 我们如何高效地训练 $\mathbf{w}$？特别是，要应用 SGD 规则，我们再次需要在 $\mathcal{Y}$ 上解决一个最大化问题。3. 我们如何避免过拟合？

在上一节中，我们已经表明学习线性多类预测器的样本复杂度并不显式依赖于类别数量。我们只需确保Ψ的范围范数不要太大。这将解决过拟合问题。为了应对计算挑战，我们依赖于问题的结构，并定义函数Ψ和Δ，以便在$h_{\mathbf{w}}$的定义和SGD算法中的最大化问题可以高效地计算。在以下内容中，我们将展示实现这些目标的一种方法，用于之前提到的OCR任务。

为了简化展示，让我们假设 $\mathcal{Y}$ 中的所有单词长度为 r，并且我们字母表中的不同字母数量为 q。设 $\mathbf{y}$ 和 $\mathbf{y}'$ 为两个

单词（即字母序列）在 $\mathcal{Y}$ 中。我们定义函数 $\Delta(\mathbf{y}', \mathbf{y})$ 为 y' 和 y 中不同字母的平均数，即 $\frac{1}{r}\sum_{i=1}^{r} 1_{[y_i \neq y'_i]}$。

接下来，让我们定义一个类敏感的特征映射 {v*}。将 $\mathbf{x}$ 视为一个大小为 $n \times r$ 的矩阵将很方便，其中 n 是每张图像中的像素数，r 是序列中的图像数。$\mathbf{x}$ 的第 j 列对应于序列中的第 j 张图像（编码为像素灰度值的向量）。Ψ 的范围维度被设置为 $d = n\,q + q^2$。

第一个 nq 特征函数是“类型1”特征，其形式为：

$$\Psi_{i,j,1}(\mathbf{x}, \mathbf{y}) = \frac{1}{r} \sum_{t=1}^{r} x_{i,t}\, \mathbb{1}_{[y_t = j]}.$$

即，我们只对那些$\mathbf{y}$分配字母j的图像求i第个像素的值之和。三重索引$(i, j, 1)$表示我们正在处理类型1的特征(i, j)。直观上，这类特征可以捕捉到图像中灰度值指示某个字母的像素。第二种特征的形式是

$$\Psi_{i,j,2}(\mathbf{x}, \mathbf{y}) = \frac{1}{r} \sum_{t=2}^{r} \mathbb{1}_{[y_t = i]}\, \mathbb{1}_{[y_{t-1} = j]}.$$

这是，我们计算字母 i 后跟字母 j 的次数。直观上，这些特征可以捕捉到像“在单词中看到‘qu’这对字母的可能性很大”或“在单词中看到‘rz’这对字母的可能性很小”这样的规则。当然，其中一些特征可能不太有用，因此学习过程的目标是通过学习向量 $\mathbf{w}$ 为特征分配权重，以便加权分数能通过

$$h_{\mathbf{w}}(\mathbf{x}) = \operatorname*{argmax}_{\mathbf{y} \in \mathcal{Y}} \langle \mathbf{w}, \Psi(\mathbf{x}, \mathbf{y}) \rangle.$$

它留给展示如何有效地解决 $h_{\mathbf{w}}(\mathbf{x})$ 定义中的优化问题，以及如何在 SGD 算法中解决 $\hat{y}$ 定义中的优化问题。我们可以通过应用动态规划程序来完成这项工作。我们描述了求解 $h_{\mathbf{w}}$ 定义中最大化的程序，并将 SGD 算法中 $\hat{y}$ 定义中的最大化问题留作练习。

要推导动态规划过程，我们首先观察可以写出

$$\Psi(\mathbf{x}, \mathbf{y}) = \sum_{t=1}^{r} \boldsymbol{\phi}(\mathbf{x}, y_t, y_{t-1}),$$

对于一个适当的 $\boldsymbol{\phi} : \mathcal{X} \times [q] \times [q] \cup \{0\} \to \mathbb{R}^d$，并且为了简化，我们假设 y_0 总是等于 0。实际上，每个特征函数 $\Psi_{i,j,1}$ 可以用以下方式表示

$$\phi_{i,j,1}(\mathbf{x}, y_t, y_{t-1}) = x_{i,t}\, \mathbb{1}_{[y_t = j]},$$

当特征函数 $\Psi_{i,j,2}$ 可以用以下方式表示时

$$\phi_{i,j,2}(\mathbf{x}, y_t, y_{t-1}) = \mathbb{1}_{[y_t=i]}\,\mathbb{1}_{[y_{t-1}=j]}.$$

因此，预测可以写成

$$h_{\mathbf{w}}(\mathbf{x}) = \operatorname*{argmax}_{\mathbf{y}\in\mathcal{Y}} \sum_{t=1}^{r} \langle \mathbf{w}, \boldsymbol{\phi}(\mathbf{x}, y_t, y_{t-1})\rangle. \tag{17.4}$$

在以下内容中，我们推导出一个动态规划过程，该过程解决了形式如方程（17.4）给出的每个问题。该过程将维护一个矩阵 $M \in \mathbb{R}^{q,r}$，使得

$$M_{s,\tau} = \max_{(y_1,\ldots,y_\tau):y_\tau=s} \sum_{t=1}^{\tau} \langle \mathbf{w}, \boldsymbol{\phi}(\mathbf{x}, y_t, y_{t-1})\rangle.$$

显然，$\langle \mathbf{w}, \Psi(\mathbf{x}, \mathbf{y})\rangle$ 的最大值等于 $\max_s M_{s,r}$。此外，我们可以递归地计算 M:

$$M_{s,\tau} = \max_{s'} \left(M_{s',\tau-1} + \langle \mathbf{w}, \boldsymbol{\phi}(\mathbf{x}, s, s')\rangle \right). \tag{17.5}$$

这产生了以下步骤:

Dynamic Programming for Calculating $h_{\mathbf{w}}(\mathbf{x})$ as Given in Equation (17.4)

input: a matrix $\mathbf{x} \in \mathbb{R}^{n,r}$ and a vector $\mathbf{w}$
initialize:
  foreach $s \in [q]$
    $M_{s,1} = \langle \mathbf{w}, \boldsymbol{\phi}(\mathbf{x}, s, -1)\rangle$
for $\tau = 2, \ldots, r$
  foreach $s \in [q]$
    set $M_{s,\tau}$ as in Equation (17.5)
    set $I_{s,\tau}$ to be the s' that maximizes Equation (17.5)
set $y_t = \operatorname{argmax}_s M_{s,r}$
for $\tau = r,\ r-1, \ldots, 2$
  set $y_{\tau-1} = I_{y_\tau,\tau}$
output: $\mathbf{y} = (y_1, \ldots, y_r)$

17.4 Ranking

排名是根据其实际“相关性”对一组实例进行排序的问题。一个典型的应用是根据查询的相关性对搜索引擎的结果进行排序。另一个例子是监控系统电子交易，并应警告可能的欺诈交易。这样的系统应按可疑程度对交易进行排序。

形式上，令 $\mathcal{X}^* = \bigcup_{n=1}^{\infty} \mathcal{X}^n$ 为所有实例序列的集合

$\mathcal{X}$ 任意长度。一个排名假设，h，是一个接收实例序列 $\bar{\mathbf{x}} = (\mathbf{x}_1, \ldots, \mathbf{x}_r) \in \mathcal{X}^*$ 的函数，并返回 $[r]$ 的排列。让 h 的输出为一个向量 $\mathbf{y} \in \mathbb{R}^r$ 更为方便，通过排序 $\mathbf{y}$ 的元素，我们得到 $[r]$ 上的排列。我们用 $\pi(\mathbf{y})$ 表示由 $\mathbf{y}$ 诱导的 $[r]$ 上的排列。例如，对于 $r = 5$，向量 $\mathbf{y} = (2, 1, 6, -1, 0.5)$ 诱导排列 $\pi(\mathbf{y}) = (4, 3, 5, 1, 2)$。也就是说，如果我们按升序排序 $\mathbf{y}$，那么我们得到向量 $(-1, 0.5, 1, 2, 6)$。现在，$\pi(\mathbf{y})_i$ 是 y_i 在排序向量 $(-1, 0.5, 1, 2, 6)$ 中的位置。这种表示法反映了排名最高的实例是在 $\pi(\mathbf{y})$ 中取得最高值的实例。

在PAC学习模型的记法中，示例域是$Z = \bigcup_{r=1}^{\infty}(\mathcal{X}^r \times \mathbb{R}^r)$，而假设类$\mathcal{H}$是一组排序假设。接下来，我们描述排序的损失函数。定义此类损失函数有许多可能的方法，这里我们列出一些示例。在所有示例中，我们定义 $\ell(h, (\bar{\mathbf{x}}, \mathbf{y})) = \Delta(h(\bar{\mathbf{x}}), \mathbf{y})$，对于某个函数$\Delta$：$\bigcup_{r=1}^{\infty}(\mathbb{R}^r \times \mathbb{R}^r) \to \mathbb{R}_+$。

- **0–1 Ranking loss:** $\Delta(\mathbf{y}', \mathbf{y})$ 如果 $\mathbf{y}$ 和 $\mathbf{y}'$ 诱导出完全相同的排名，则该值为零；否则为 1。也就是说，$\Delta(\mathbf{y}', \mathbf{y}) = 1_{[\pi(\mathbf{y}') \neq \pi(\mathbf{y})]}$。这种损失函数在实践中的应用几乎从未使用过，因为它无法区分 $\pi(\mathbf{y}')$ 几乎等于 $\pi(\mathbf{y})$ 的情况和 $\pi(\mathbf{y}')$ 与 $\pi(\mathbf{y})$ 完全不同的情况。

- **Kendall-Tau Loss:** 我们计算两个排列中顺序不同的对（i, j）的数量。这可以写成

$$\Delta(\mathbf{y}', \mathbf{y}) = \frac{2}{r(r-1)} \sum_{i=1}^{r-1} \sum_{j=i+1}^{r} \mathbb{1}_{[\text{sign}(y'_i - y'_j) \neq \text{sign}(y_i - y_j)]}.$$

  这个损失函数比0-1损失更有用，因为它反映了两个排名之间的相似程度。

- **Normalized Discounted Cumulative Gain (NDCG):** 此措施通过使用单调不减的折扣函数 D：$\mathbb{N} \to \mathbb{R}_+$ 来强调列表中最正确的项。我们首先定义一个折扣累积收益度量：

$$G(\mathbf{y}', \mathbf{y}) = \sum_{i=1}^{r} D(\pi(\mathbf{y}')_i)\ y_i.$$

  在文字上，如果我们把 y_i 解释为项目 i 的“真实相关性”得分，那么我们取元素相关性的加权总和，而 y_i 的权重是在 i 在 $\pi(\mathbf{y}')$ 中的位置基础上确定的。假设 $\mathbf{y}$ 的所有元素都是非负的，很容易验证 $0 \leq G(\mathbf{y}', \mathbf{y}) \leq G(\mathbf{y},\ \mathbf{y})$。因此，我们可以通过比率 $G(\mathbf{y}',\ \mathbf{y})/G(\mathbf{y},\ \mathbf{y})$ 定义一个归一化的累积增益，相应的损失函数将是

$$\Delta(\mathbf{y}', \mathbf{y}) = 1 - \frac{G(\mathbf{y}', \mathbf{y})}{G(\mathbf{y}, \mathbf{y})} = \frac{1}{G(\mathbf{y}, \mathbf{y})} \sum_{i=1}^{r} \left(D(\pi(\mathbf{y})_i) - D(\pi(\mathbf{y}')_i)\right)\ y_i.$$

我们可以轻易地看出 $\Delta(\mathbf{y}', \mathbf{y}) \in [0, 1]$ 以及当 $\pi(\mathbf{y}') = \pi(\mathbf{y})$ 时 $\Delta(\mathbf{y}', \mathbf{y}) = 0$。

一种典型的定义折扣函数的方法是

$$D(i) = \begin{cases} \frac{1}{\log_2(r-i+2)} & \text{if } i \in \{r-k+1, \ldots, r\} \\ 0 & \text{otherwise} \end{cases}$$

$k < r$是一个参数。这意味着我们更关注排名更高的元素，并且完全忽略不在前-k排名的元素。NDCG度量通常用于评估搜索引擎的性能，因为在这样的应用中，完全忽略排名不在前列的元素是有意义的。

一旦我们有一个假设类和一个排序损失函数，我们可以使用ERM规则学习一个排序函数。然而，从计算的角度来看，得到的优化问题可能难以解决。接下来，我们将讨论如何学习排序的线性预测器。

17.4.1 Linear Predictors for Ranking

一种定义排名函数的自然方法是投影实例到某个向量 $\mathbf{w}$，然后将得到的标量作为排名函数的表示输出。也就是说，假设对于每个 $\mathcal{X} \subset \mathbb{R}^d$，我们定义一个排名函数 $\mathbf{w} \in \mathbb{R}^d$。

$$h_{\mathbf{w}}((\mathbf{x}_1, \ldots, \mathbf{x}_r)) = (\langle \mathbf{w}, \mathbf{x}_1 \rangle, \ldots, \langle \mathbf{w}, \mathbf{x}_r \rangle). \tag{17.6}$$

如我们在第16章中讨论的，我们还可以应用一种特征映射，将实例映射到某个特征空间，然后在特征空间中与$\mathbf{w}$取内积。为了简单起见，我们关注更简单的形式，如方程（17.6）所示。

给定一些 $W \subset \mathbb{R}^d$，我们现在可以定义假设类 $\mathcal{H}_W = \{h_{\mathbf{w}} \colon \mathbf{w} \in W\}$。一旦我们定义了这个假设类，并选择了一个排序损失函数，我们就可以应用 ERM 规则如下：给定一个训练集，$S = (\bar{\mathbf{x}}_1, \mathbf{y}_1), \ldots, (\bar{\mathbf{x}}_m, \mathbf{y}_m)$，其中每个 $(\bar{\mathbf{x}}_i, \mathbf{y}_i)$ 都在 $(\mathcal{X} \times \mathbb{R})^{r_i}$ 中，对于某个 $r_i \in \mathbb{N}$，我们应该寻找 $\mathbf{w} \in W$ 以最小化经验损失，$\sum_{i=1}^m \Delta(h_{\mathbf{w}}(\bar{\mathbf{x}}_i), \mathbf{y}_i)$。与二元分类的情况一样，对于许多损失函数，这个问题在计算上是困难的，因此我们转向描述凸代理损失函数。我们描述了 Kendall tau 损失和 NDCG 损失的代理。

A Hinge Loss for the Kendall Tau Loss Function:
我们可以将肯德尔tau损失视为每对的平均0−1损失。特别是，对于每个(i, j)，我们可以重新写为

$$\mathbb{1}_{[\text{sign}(y'_i - y'_j) \neq \text{sign}(y_i - y_j)]} = \mathbb{1}_{[\text{sign}(y_i - y_j)(y'_i - y'_j) \leq 0]}.$$

在我们的情况下，$y'_i - y'_j = \langle \mathbf{w}, \mathbf{x}_i - \mathbf{x}_j \rangle$。因此，我们可以使用如下 hinge 损失上界：

$$\mathbb{1}_{[\text{sign}(y_i - y_j)(y'_i - y'_j) \leq 0]} \;\leq\; \max\left\{0, 1 - \text{sign}\,(y_i - y_j)\,\langle \mathbf{w}, \mathbf{x}_i - \mathbf{x}_j \rangle\right\}.$$

对成对数据进行平均，我们得到以下Kendall tau损失函数的代理凸损失：

$$\Delta(h_{\mathbf{w}}(\bar{\mathbf{x}}), \mathbf{y}) \;\leq\; \frac{2}{r(r-1)} \sum_{i=1}^{r-1} \sum_{j=i+1}^{r} \max\left\{0, 1 - \text{sign}(y_i - y_j)\,\langle \mathbf{w}, \mathbf{x}_i - \mathbf{x}_j \rangle\right\}.$$

右侧相对于 $\mathbf{w}$ 是凸的，并且是 Kendall tau 损失的上界。它也是一个具有参数 $\rho \leq \max_{i,j} \|\mathbf{x}_i - \mathbf{x}_j\|$ 的 ρ-Lipschitz 函数。

A Hinge Loss for the NDCG Loss Function:
NDCG损失函数依赖于通过其诱导的排列预测的排名向量$\mathbf{y}' \in \mathbb{R}^r$。为了推导出代理损失函数，我们首先做出以下观察。设V为将$[r]$的所有排列编码为向量的集合；即，每个$\mathbf{v} \in V$是$[r]$中的向量，满足对于所有$i \neq j$，我们有$v_i \neq v_j$。然后（参见练习4），

$$\pi(\mathbf{y}') \;=\; \operatorname*{argmax}_{\mathbf{v} \in V} \sum_{i=1}^{r} v_i\, y'_i. \tag{17.7}$$

让我们表示 $\Psi(\bar{\mathbf{x}}, \mathbf{v}) = \sum_{i=1}^{r} v_i \mathbf{x}_i$；因此，

$$\begin{aligned}
\pi(h_{\mathbf{w}}(\bar{\mathbf{x}})) &= \operatorname*{argmax}_{\mathbf{v} \in V} \sum_{i=1}^{r} v_i \langle \mathbf{w}, \mathbf{x}_i \rangle \\
&= \operatorname*{argmax}_{\mathbf{v} \in V} \left\langle \mathbf{w}, \sum_{i=1}^{r} v_i \mathbf{x}_i \right\rangle \\
&= \operatorname*{argmax}_{\mathbf{v} \in V} \langle \mathbf{w}, \Psi(\bar{\mathbf{x}}, \mathbf{v}) \rangle.
\end{aligned}$$

基于这一观察，我们可以将广义铰链损失用于成本敏感的多类分类，作为NDCG损失的代理损失函数，如下所示：

$$\begin{aligned}
\Delta(h_{\mathbf{w}}(\bar{\mathbf{x}}), \mathbf{y}) \;&\leq\; \Delta(h_{\mathbf{w}}(\bar{\mathbf{x}}), \mathbf{y}) + \langle \mathbf{w}, \Psi(\bar{\mathbf{x}}, \pi(h_{\mathbf{w}}(\bar{\mathbf{x}}))) \rangle - \langle \mathbf{w}, \Psi(\bar{\mathbf{x}}, \pi(\mathbf{y})) \rangle \\
&\leq \max_{\mathbf{v} \in V} \; [\Delta(\mathbf{v}, \mathbf{y}) + \langle \mathbf{w}, \Psi(\bar{\mathbf{x}}, \mathbf{v}) \rangle - \langle \mathbf{w}, \Psi(\bar{\mathbf{x}}, \pi(\mathbf{y})) \rangle] \\
&= \max_{\mathbf{v} \in V} \; \left[\Delta(\mathbf{v}, \mathbf{y}) + \sum_{i=1}^{r} (v_i - \pi(\mathbf{y})_i) \langle \mathbf{w}, \mathbf{x}_i \rangle \right].
\end{aligned} \tag{17.8}$$

T他右侧是一个相对于 {v*} 的凸函数 $\mathbf{w}$.

我们现在可以使用SGD解决学习问题，如第17.2.5节所述。主要的计算瓶颈是计算损失函数的子梯度，这相当于找到使方程（17.8）达到最大值的$\mathbf{v}$（参见命题14.6）。使用NDCG损失的定义，这可以表示为

等同于解决问题

$$\operatorname*{argmin}_{\mathbf{v}\in V} \sum_{i=1}^{r} (\alpha_i v_i + \beta_i D(v_i)),$$

在 $\alpha_i = -\langle \mathbf{w}, \mathbf{x}_i \rangle$ 和 $\beta_i = y_i / G(\mathbf{y}, \mathbf{y})$ 的地方。我们可以通过定义一个矩阵 $A \in \mathbb{R}^{r,r}$ 来稍微不同地思考这个问题，其中

$$A_{i,j} = j\alpha_i + D(j)\,\beta_i.$$

现在，让我们将每个 j 视为一个“工人”，每个 i 视为一个“任务”，$A_{i,j}$ 视为将任务 i 分配给工人 j 的成本。从这个角度来看，找到 $\mathbf{v}$ 的问题变成了找到具有最小成本的任务分配问题。这个问题被称为“分配问题”，并且可以有效地解决。一个特定的算法是“匈牙利方法”（Kuhn 1955）。解决分配问题的另一种方法是使用线性规划。为此，我们首先将分配问题写成

$$\begin{aligned} \operatorname*{argmin}_{B\in\mathbb{R}_+^{r,r}} & \sum_{i,j=1}^{r} A_{i,j} B_{i,j} \\ \text{s.t.} \quad & \forall i \in [r], \ \sum_{j=1}^{r} B_{i,j} = 1 \\ & \forall j \in [r], \ \sum_{i=1}^{r} B_{i,j} = 1 \\ & \forall i, j, \quad B_{i,j} \in \{0, 1\} \end{aligned} \tag{17.9}$$

一个满足前述优化问题约束的矩阵 B 被称为置换矩阵。这是因为约束保证了每一行至多只有一个元素等于 1，每一列也至多只有一个元素等于 1。因此，矩阵 B 对应于由 $v_i = j$ 定义的单个索引 j 满足 $B_{i,j} = 1$ 的置换 $\mathbf{v} \in V$。

前述优化仍然不是一个线性规划，因为组合约束 $B_{i,j} \in \{0, 1\}$。然而，结果证明，这个约束是多余的——如果我们简单地省略组合约束来解决优化问题，我们仍然可以保证存在一个满足此约束的最优解。这一点将在后面形式化。

表示 $\langle A, B \rangle = \sum_{i,j} A_{i,j} B_{i,j}$。然后，方程（17.9）是使 B 成为置换矩阵的最小化 $\langle A, B \rangle$ 的问题。

一个矩阵 $B \in \mathbb{R}^{r,r}$ 被称为 *doubly stochastic*，如果 B 的所有元素都是非负的，B 的每一行的和为 1，以及 B 的每一列的和为 1。因此，在不加约束 $B_{i,j} \in \{0, 1\}$ 的情况下求解方程 (17.9) 是一个问题

$$\operatorname*{argmin}_{B\in\mathbb{R}^{r,r}} \langle A, B \rangle \ \text{ s.t. } \ B \ \text{ is a doubly stochastic matrix.} \tag{17.10}$$

以下声明指出，每个双随机矩阵都是置换矩阵的凸组合。

CLAIM 17.3 ((Birkhoff 1946, Von Neumann 1953)) *The set of doubly stochastic matrices in $\mathbb{R}^{r,r}$ is the convex hull of the set of permutation matrices in $\mathbb{R}^{r,r}$.*

基于该主张，我们很容易得到以下结果：

LEMMA 17.4 *There exists an optimal solution of Equation (17.10) that is also an optimal solution of Equation (17.9).*

Proof 设 B 是方程（17.10）的解。然后，根据17.3命题，我们可以写出 $B = \sum_i \gamma_i C_i$，其中每个 C_i 是一个置换矩阵，每个 $\gamma_i >$ 为0，$\sum_i \gamma_i =$ 为1。由于所有 C_i 都是双随机矩阵，我们显然有 $\langle A,\ B\rangle \leq \langle A,\ C_i\rangle$ 对于每个 i 都成立。我们断言存在某个 i，使得 $\langle A,\ B\rangle = \langle A,\ C_i\rangle$。这必须是真的，否则，如果对于每个 i $\langle A,\ B\rangle < \langle A,\ C_i\rangle$，我们都会有

$$\langle A, B\rangle = \left\langle A, \sum_i \gamma_i C_i \right\rangle = \sum_i \gamma_i \langle A, C_i\rangle > \sum_i \gamma_i \langle A, B\rangle = \langle A, B\rangle,$$

无法保持。因此，我们已经证明某些排列矩阵 C_i 满足 $\langle A, B\rangle = \langle A, C_i\rangle$。但是，由于对于每个其他排列矩阵 C，我们都有 $\langle A, B\rangle \leq \langle A, C\rangle$，我们得出结论，$C_i$ 是方程（17.9）和方程（17.10）的优化解。

□

17.5 Bipartite Ranking and Multivariate Performance Measures

在上一节中，我们描述了排序问题。我们使用向量 $\mathbf{y} \in \mathbb{R}^r$ 来表示元素 $\mathbf{x}_1, \ldots, \mathbf{x}_r$ 的顺序。如果 $\mathbf{y}$ 中的所有元素都彼此不同，那么 $\mathbf{y}$ 指定了一个对 $[r]$ 的全序。然而，如果 $\mathbf{y}$ 的两个元素达到相同的值，即 $y_i = y_j$ 对于 $i \neq j$，那么 $\mathbf{y}$ 只能指定 $[r]$ 的部分序。在这种情况下，我们说根据 $\mathbf{y}$，$\mathbf{x}_i$ 和 $\mathbf{x}_j$ 的相关性相等。在极端情况下，$\mathbf{y} \in \{\pm 1\}^r$，这意味着每个 $\mathbf{x}_i$ 要么是相关的，要么是不相关的。这种设置通常被称为“二分排序”。例如，在上一节中提到的欺诈检测应用中，每笔交易都被标记为欺诈 ($y_i = 1$) 或良性 ($y_i = -1$)。

表面上，我们可以通过学习一个二元分类器，将其应用于每个实例，并将正例放在排名列表的顶部来解决二分排序问题。然而，这可能会导致结果不佳，因为二元学习者的目标通常是最小化零一损失（或其代理），而排序器的目标可能显著不同。为了说明这一点，再次考虑欺诈检测问题。通常，大多数交易都是良性的（比如说99.9%）。因此，一个对所有交易都预测“良性”的二元分类器将具有0.1%的零一错误。虽然这个数字非常小，但这个预测对于欺诈检测应用来说毫无意义。这个问题的关键在于

问题源于零一损失的不足，而我们真正感兴趣的是。一个更合适的性能度量应该考虑整个实例集的预测。例如，在上一节中，我们定义了NDCG损失，它强调了排名靠前项的正确性。在本节中，我们描述了额外的损失函数，这些函数专门适用于二分排序问题。

与上一节一样，我们给定一个实例序列 $\bar{\mathbf{x}} = (\mathbf{x}_1, \ldots, \mathbf{x}_r)$，并预测一个排名向量 $\mathbf{y}' \in \mathbb{R}^r$。反馈向量为 $\mathbf{y} \in \{\pm 1\}^r$。我们定义一个依赖于 $\mathbf{y}'$ 和 $\mathbf{y}$ 的损失，并依赖于一个阈值 $\theta \in \mathbb{R}$。此阈值将向量 $\mathbf{y}' \in \mathbb{R}^r$ 转换为向量 $(\text{sign}(y'_i - \theta), \ldots, \text{sign}(y'_r - \theta)) \in \{\pm 1\}^r$。通常，$\theta$ 的值设置为 0。然而，正如我们将看到的，我们在考虑问题的额外约束时有时会设置 θ。

以下我们定义的损失函数依赖于以下4个数字：

$$\begin{aligned} \text{True positives:} \quad & a = |\{i : y_i = +1 \wedge \text{sign}(y'_i - \theta) = +1\}| \\ \text{False positives:} \quad & b = |\{i : y_i = -1 \wedge \text{sign}(y'_i - \theta) = +1\}| \\ \text{False negatives:} \quad & c = |\{i : y_i = +1 \wedge \text{sign}(y'_i - \theta) = -1\}| \\ \text{True negatives:} \quad & d = |\{i : y_i = -1 \wedge \text{sign}(y'_i - \theta) = -1\}| \end{aligned} \tag{17.11}$$

预测向量的 **recall**（，即 **sensitivity**），是真正例的分数“捕获”，即 $\frac{a}{a+c}$。**precision** 是我们预测的阳性标签中正确预测的分数，即 $\frac{a}{a+b}$。**specificity** 是我们预测器“捕获”的真正例的分数，即 $\frac{d}{d+b}$。

注意，当我们减小 θ 时，召回率会增加（当 $\theta = -\infty$ 时达到值 1）。另一方面，当我们减小 θ 时，精确度和特异性通常会降低。因此，精确度和召回率之间存在权衡，我们可以通过改变 θ 来控制它。以下定义的损失函数使用各种技术来结合精确度和召回率。

- **Averaging sensitivity and specificity**此度量是灵敏度和特异性的平均值，即，$\frac{1}{2}\left(\frac{a}{a+c} + \frac{d}{d+b}\right)$。这同样是正例上的准确率与负例上的准确率的平均值。在此，我们将θ =设为0，相应的损失函数为$\Delta(\mathbf{y}', \mathbf{y})$ =1 $- \frac{1}{2}\left(\frac{a}{a+c} + \frac{d}{d+b}\right)$。

- F_1**-score**：F_1得分是精确率和召回率的调和平均值：$\frac{2}{\frac{1}{\text{Precision}} + \frac{1}{\text{Recall}}}$。其最大值（为1）是在精确率和召回率都为1时获得，其最小值（为0）是在其中一个为0时获得（即使另一个为1）。F_1得分可以用以下数字a, b, c表示；$F_1 = \frac{2a}{2a+b+c}$。再次，我们将θ =设为0，损失函数变为$\Delta(\mathbf{y}', \mathbf{y})$ =1 $- F_1$。

- F_β**-score**它类似于 F_1 分数，但我们比精确度更重视召回率 β^2 倍，即 $\frac{1+\beta^2}{\frac{1}{\text{Precision}} + \beta^2 \frac{1}{\text{Recall}}}$。也可以写成

$F_\beta = \frac{(1+\beta^2)a}{(1+\beta^2)a+b+\beta^2 c}$。再次，我们将 $\theta =$ 设置为 0，损失函数变为 $\Delta(\mathbf{y}', \mathbf{y}) = 1 - F_\beta$。

- **Recall at** k我们在测量召回率时，预测必须包含最多 k 个正标签。也就是说，我们应该设置 θ 以便 $a + b \leq k$。这在应用欺诈检测系统时很方便，例如，银行员工只能处理少量可疑交易。

- **Precision at** k我们在测量精度时，预测必须至少包含 k 个正面标签。也就是说，我们应该设置 θ 以确保 $a + b \geq k$。

之前定义的措施通常被称为 *multivariate perfor- mance measures*。请注意，这些措施与平均零一损失高度不同，在先前的符号中等于 $\frac{b+d}{a+b+c+d}$。在之前提到的欺诈检测示例中，当99.9%的示例被标记为负标签时，预测所有示例都是负例的零一损失为0.1%。相比之下，这种预测的召回率为0，因此 F_1 分数也为0，这意味着相应的损失将是1。

17.5.1 Linear Predictors for Bipartite Ranking

我们接下来描述如何训练用于二分排序的线性预测器。与上一节一样，排序的线性预测器被定义为

$$h_{\mathbf{w}}(\bar{\mathbf{x}}) = (\langle \mathbf{w}, \mathbf{x}_1 \rangle, \ldots, \langle \mathbf{w}, \mathbf{x}_r \rangle).$$

相应的损失函数是之前描述的多变量性能度量之一。损失函数通过它诱导的二进制向量依赖于 $\mathbf{y}' = h_{\mathbf{w}}(\bar{\mathbf{x}})$，我们将其表示为

$$\mathbf{b}(\mathbf{y}') = (\text{sign}(y'_1 - \theta), \ldots, \text{sign}(y'_r - \theta)) \in \{\pm 1\}^r. \tag{17.12}$$

与上一节类似，为了便于高效算法，我们在 Δ 上推导出一个凸代理损失函数。其推导过程与上一节所述的 NDCG 排序损失的广义铰链损失函数的推导过程类似。

我们的第一个观察结果是，对于在之前定义的所有 θ 的值，存在某个 $V \subseteq \{\pm 1\}^r$，使得 $\mathbf{b}(\mathbf{y}')$ 可以重写为

$$\mathbf{b}(\mathbf{y}') = \operatorname*{argmax}_{\mathbf{v} \in V} \sum_{i=1}^{r} v_i y'_i. \tag{17.13}$$

这是对于选择 $V = \{\pm 1\}^r$ 的情况 $\theta = 0$ 明显是正确的。对于 θ 不取 0 的两个度量是 k 的精确度和 k 的召回率。对于 k 的精确度，我们可以将 V 取为包含 $\{\pm 1\}^r$ 中所有至少有 k 个 1 的向量的集合 $V_{\geq k}$。对于 k 的召回率，我们可以将 V 取为 $V_{\leq k}$，其定义方式类似。参见练习 5。

一旦我们定义了**b**如方程（17.13）所示，我们可以轻松地推导出一个凸代理损失，如下所示。假设$\mathbf{y} \in V$，我们有

$$\begin{aligned}\Delta(h_{\mathbf{w}}(\bar{\mathbf{x}}), \mathbf{y}) &= \Delta(\mathbf{b}(h_{\mathbf{w}}(\bar{\mathbf{x}})), \mathbf{y}) \\ &\leq \Delta(\mathbf{b}(h_{\mathbf{w}}(\bar{\mathbf{x}})), \mathbf{y}) + \sum_{i=1}^{r} (b_i(h_{\mathbf{w}}(\bar{\mathbf{x}})) - y_i)\langle \mathbf{w}, \mathbf{x}_i \rangle \\ &\leq \max_{\mathbf{v} \in V} \left[\Delta(\mathbf{v}, \mathbf{y}) + \sum_{i=1}^{r} (v_i - y_i) \langle \mathbf{w}, \mathbf{x}_i \rangle \right]. \qquad (17.14)\end{aligned}$$

右侧是关于 $\mathbf{w}$ 的凸函数。

我们现在可以使用SGD来解决学习问题，如第17.2.5节所述。主要的计算瓶颈是计算损失函数的子梯度，这相当于找到使方程（17.14）达到最大值的$\mathbf{v}$（参见命题14.6）。

在以下内容中，我们描述了如何高效地找到任何可以表示为方程（17.11）中给出的数字 $a,\ b,\ c,\ d$ 的函数的任何性能度量下的这个最大化器，并且对于集合 V 包含 $\{\pm 1\}^r$ 中所有满足某些约束的 a, b 值的元素。例如，对于“在 k 的召回率”，集合 V 是所有满足 $a + b \leq k$ 的向量。

以下为思路。对于任意的 $a, b \in [r]$，让

$$\bar{\mathcal{Y}}_{a,b} = \{\mathbf{v} \ : \ |\{i : v_i = 1 \wedge y_i = 1\}| = a \ \wedge \ |\{i : v_i = 1 \wedge y_i = -1\}| = b\}.$$

任何向量 $\mathbf{v} \in V$ 都属于 $\bar{\mathcal{Y}}_{a,b}$，对于某些 $a, b \in [r]$。此外，如果对于某些 $a, b \in [r]$，$\bar{\mathcal{Y}}_{a,b} \cap V$ 不为空，那么 $\bar{\mathcal{Y}}_{a,b} \cap V = \bar{\mathcal{Y}}_{a,b}$。因此，我们可以在与 V 非空交集的每个 $\bar{\mathcal{Y}}_{a,b}$ 中分别进行搜索，然后取最优值。关键观察是，一旦我们只搜索 $\bar{\mathcal{Y}}_{a,b}$，Δ 的值就固定了，所以我们只需要最大化表达式

$$\max_{\mathbf{v} \in \bar{\mathcal{Y}}_{a,b}} \sum_{i=1}^{r} v_i \langle \mathbf{w}, \mathbf{x}_i \rangle.$$

假设示例已按顺序排列，使得 $\langle \mathbf{w}, \mathbf{x}_1 \rangle \geq \cdots \geq \langle \mathbf{w}, \mathbf{x}_r \rangle$。然后，很容易验证我们希望将 v_i 设置为正数，对于最小的索引 i。这样做，在 a, b 的约束下，相当于将 $v_i =$ 设置为 1，对于 a 个排名最高的正例和 b 个排名最高的负例。这产生了以下过程。

Solving Equation (17.14)

input:
$(\mathbf{x}_1, \ldots, \mathbf{x}_r), (y_1, \ldots, y_r), \mathbf{w}, V, \Delta$
assumptions:
Δ is a function of $a,\ b,\ c,\ d$
V contains all vectors for which $f(a, b) = 1$ for some function f
initialize:
$P = |\{i : y_i = 1\}|$, $N = |\{i : y_i = -1\}|$
$\boldsymbol{\mu} = (\langle \mathbf{w}, \mathbf{x}_1 \rangle, \ldots, \langle \mathbf{w}, \mathbf{x}_r \rangle)$, $\alpha^\star = -\infty$
sort examples so that $\mu_1 \geq \mu_2 \geq \cdots \geq \mu_r$
let $i_1, \ldots, i_P$ be the (sorted) indices of the positive examples
let $j_1, \ldots, j_N$ be the (sorted) indices of the negative examples
for $a = 0, 1, \ldots, P$
$c = P - a$
for $b = 0, 1, \ldots, N$ such that $f(a, b) = 1$
$d = N - b$
calculate Δ using $a,\ b,\ c,\ d$
set $v_1, \ldots, v_r$ s.t. $v_{i_1} = \cdots = v_{i_a} = v_{j_1} = \cdots = v_{j_b} = 1$
and the rest of the elements of $\mathbf{v}$ equal -1
set $\alpha = \Delta + \sum_{i=1}^{r} v_i \mu_i$
if $\alpha \geq \alpha^\star$
$\alpha^\star = \alpha$, $\mathbf{v}^\star = \mathbf{v}$
output $\mathbf{v}^\star$

17.6 Summary

许多现实世界的监督学习问题可以表述为学习多类预测器。我们以介绍将多类学习简化为二类学习作为本章的开端。然后，我们描述并分析了多类学习中的线性预测器族。我们展示了即使类别数量极大，只要我们对问题有足够的结构，这个家族仍然可以使用。最后，我们描述了排序问题。在第29章中，我们将更详细地研究多类学习的样本复杂度。

17.7 Bibliographic Remarks

One-versus-All和All-Pairs方法简化已在错误校正输出码（ECOC）框架下统一（Dietterich & Bakiri 1995，Allwein, Schapire & Singer 2000）。还有其他类型的简化，例如基于树的分类器（例如，参见Beygelzimer, Langford & Ravikumar（2007））。已经研究了简化技术的局限性。

在（Daniely等人2011年，Daniely、Sabato和Shwartz 2012年）。参见第29章，其中我们分析了多类学习的样本复杂度。

直接使用线性预测器进行多类学习的途径已在（Vapnik 1998，Weston & Watkins 1999，Crammer & Singer 2001）中研究。特别是，多向量构造归功于Crammer & Singer（2001）。

柯林斯（2000）展示了如何将感知机算法应用于结构化输出问题。参见柯林斯（2002）。相关的方法是条件随机字段的判别学习；参见拉法蒂、麦克卡伦和佩雷拉（2001）。结构化输出SVM已在以下文献中研究：（韦斯顿、查佩勒、瓦普尼克、埃利谢夫和施沃尔科普夫2002，塔斯卡、古斯特林和科勒2003，索尚塔里迪斯、霍夫曼、乔奇姆斯和阿尔腾2004）。

动态过程，我们已提出用于计算结构化输出中的预测 $h_{\mathbf{w}}(\mathbf{x})$，与HMMs中Viterbi过程计算的向前-向后变量类似（例如，参见（Rabiner & Juang 1986））。更普遍地，解决结构化输出中的最大化问题与图形模型中的推理问题密切相关（例如，参见Koller & Friedman（2009））。

Chapelle, Le & Smola (2007) 提出使用结构化输出学习中的思想来学习与NDCG损失相关的排序函数。他们还观察到，在广义铰链损失定义中的最大化问题等价于分配问题。

Agarwal & Roth (2005)分析了二分排序的样本复杂度。Joachims (2005)研究了结构化输出SVM在具有多元性能度量的二分排序中的应用。

17.8 Exercises

1. 考虑一个集合 S，其中的示例在 $\mathbb{R}^n \times [k]$ 中，对于这些示例存在向量 $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k$，使得每个示例 $(\mathbf{x}, y) \in S$ 都落在以 $\boldsymbol{\mu}_y$ 为中心、半径为 $r \geq$ 的球内。1. 假设对于每个 $i \neq j$，$\|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\| \geq 4r$。考虑通过常数 1 连接每个实例，然后应用多向量构造，即，

$$\Psi(\mathbf{x}, y) = [\ \underbrace{0, \ldots, 0}_{\in \mathbb{R}^{(y-1)(n+1)}}, \underbrace{x_1, \ldots, x_n, 1}_{\in \mathbb{R}^{n+1}}, \underbrace{0, \ldots, 0}_{\in \mathbb{R}^{(k-y)(n+1)}}\ ].$$

证明存在一个向量 $\mathbf{w} \in \mathbb{R}^{k(n+1)}$，使得对于每一个 $(\mathbf{x}, y) \in S$，有 $\ell(\mathbf{w}, (\mathbf{x}, y)) = 0$。
Hint: 观察每个例子（$\mathbf{x}, y$）$\in S$，我们可以为某个$\|\mathbf{v}\| \leq r$写出$\mathbf{x} = \boldsymbol{\mu}_y + \mathbf{v}$。现在，取$\mathbf{w} = [\mathbf{w}_1, \ldots, \mathbf{w}_k]$，其中$\mathbf{w}_i = [\boldsymbol{\mu}_i\ , -\|\boldsymbol{\mu}_i\|^2/]$²。

2. **Multiclass Perceptron:** 考虑以下算法：

Multiclass Batch Perceptron

Input:
A training set $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$
A class-sensitive feature mapping $\Psi : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^d$
Initialize: $\mathbf{w}^{(1)} = (0, \ldots, 0) \in \mathbb{R}^d$
For $t = 1, 2, \ldots$
If ($\exists\ i$ and $y \neq y_i$ s.t. $\langle \mathbf{w}^{(t)}, \Psi(\mathbf{x}_i, y_i) \rangle \leq \langle \mathbf{w}^{(t)}, \Psi(\mathbf{x}_i, y) \rangle$) then
$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + \Psi(\mathbf{x}_i, y_i) - \Psi(\mathbf{x}_i, y)$
else
output $\mathbf{w}^{(t)}$

证明以下内容:

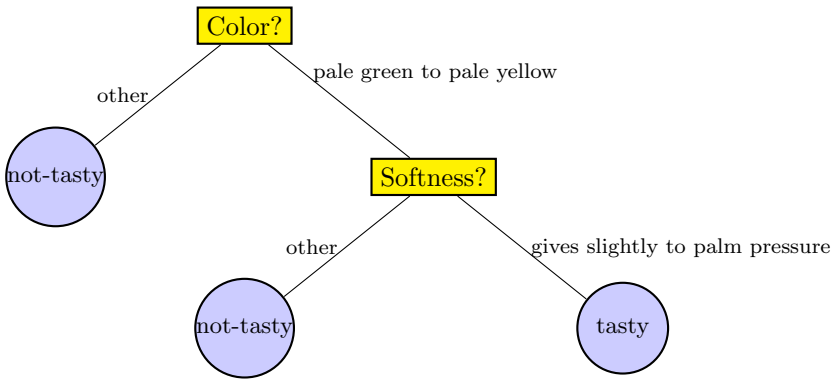
THEOREM 17.5 *Assume that there exists* $\mathbf{w}^\star$ *such that for all* i *and for all* $y \neq y_i$ *it holds that* $\langle \mathbf{w}^\star, \Psi(\mathbf{x}_i, y_i) \rangle \geq \langle \mathbf{w}^\star, \Psi(\mathbf{x}_i, y) \rangle + 1$. *Let* $R = \max_{i,y} \|\Psi(\mathbf{x}_i, y_i) - \Psi(\mathbf{x}_i, y)\|$. *Then, the multiclass Perceptron algorithm stops after at most* $(R\|\mathbf{w}^\star\|)^2$ *iterations, and when it stops it holds that* $\forall i \in [m],\ y_i = \operatorname{argmax}_y \langle \mathbf{w}^{(t)}, \Psi(\mathbf{x}_i, y) \rangle$.

3. 将第17.3节中给出的动态规划过程推广到解决多类预测SGD过程定义中$\hat{h}$的给定最大化问题。你可以假设对于某个任意函数δ，有$\Delta(\mathbf{y}', \mathbf{y}) = \sum_{t=1}^{r} \delta(y'_t, y_t)$。4. 证明方程（17.7）成立。

5. 证明根据方程（17.12）和方程（17.13）定义的 π 的两种定义对于所有多变量性能度量确实是等价的。

18 Decision Trees

决策树是一种预测器，$h : \mathcal{X} \to \mathcal{Y}$，通过从树的根节点到叶节点的路径预测与实例 $\mathbf{x}$ 相关的标签。为了简单起见，我们关注二分类设置，即 $\mathcal{Y} = \{0, 1\}$，但决策树也可以应用于其他预测问题。在根到叶路径上的每个节点，根据输入空间的分割选择后续子节点。通常，分割基于 $\mathbf{x}$ 的一个特征或一组预定义的分割规则。叶节点包含一个特定的标签。以下给出了用于木瓜示例（在第 2 章中描述）的决策树示例：



要检查一个给定木瓜是否美味，决策树首先检查木瓜的颜色。如果这个颜色不在浅绿色到浅黄色的范围内，那么树立即预测木瓜不好吃，无需额外测试。否则，树转向检查木瓜的柔软度。如果木瓜的柔软度使得它在手掌压力下略微变形，决策树预测木瓜是美味的。否则，预测结果是“不好吃”。前面的例子强调了决策树的主要优点之一——得到的分类器非常简单易懂。

18.1 Sample Complexity

一个在树内部节点上的流行分割规则是基于单个特征的阈值。也就是说，我们根据$1_{[x_i<\theta]}$移动到节点的右子节点或左子节点，其中$i \in [d]$ 是相关特征的索引，$\theta \in \mathbb{R}$ 是阈值。在这种情况下，我们可以将决策树视为将实例空间 $\mathcal{X} = \mathbb{R}^d$ 分割成单元格，其中树的每个叶子对应一个单元格。因此，具有 k 个叶子的树可以划分一组 k 个实例。因此，如果我们允许任意大小的决策树，我们得到一个无限 VC 维度的假设类。这种方法很容易导致过拟合。

为了避免过拟合，我们可以依靠第7章中描述的最小描述长度（MDL）原则，并旨在学习一个决策树，一方面它能很好地拟合数据，另一方面又不会太大。

为了简单起见，我们将假设 $\mathcal{X} = \{0,1\}^d$。换句话说，每个实例都是一个 d 位的向量。在这种情况下，对单个特征的阈值化对应于形式为 $1_{[x_i=1]}$ 的分割规则，其中 $i = [d]$。例如，我们可以通过假设一个木瓜由一个二维位向量 $\mathbf{x} \in \{0,1\}^2$ 参数化来模拟之前提到的“木瓜决策树”，其中位 x_1 表示颜色是否为浅绿色到浅黄色，位 x_2 表示软度是否对掌压有轻微的响应。在这种表示下，节点“颜色？”可以替换为 $1_{[x_1=1]}$，节点“软度？”可以替换为 $1_{[x_2=1]}$。虽然这是一个很大的简化，但我们将在以下部分提供的算法和分析可以扩展到更一般的情况。

在上述简化假设下，假设类变为有限，但仍然非常大。特别是，从 $\{0,1\}^d$ 到 $\{0,1\}$ 的任何分类器都可以用一个有 2^d 个叶子和深度为 $d+1$ 的决策树来表示（参见练习 1）。因此，该类的 VC 维度为 2^d，这意味着我们需要用于 PAC 学习假设类的样本数量随着 2^d 增长。除非 d 非常小，否则这是一个巨大的样本数量。

为了克服这个障碍，我们依赖于第7章中描述的MDL方案。基本先验知识是我们应该更喜欢较小的树而不是较大的树。为了使这种直觉形式化，我们首先需要定义一个决策树描述语言，该语言是无前缀的，并且对较小的决策树需要更少的位。这里有一种可能的方法：具有n个节点的树将在$n+$个1块中描述，每个块的大小为$\log_2(d+3)$位。第一个n块编码树的节点，按照深度优先顺序（先序），最后一个块标记代码的结束。每个块指示当前节点是：

- 内部节点形式为$1_{[x_i=1]}$，其中某些$i \in [d]$
- 一片值为1的叶子
- 一片值为0的叶子
- 代码结束

总体来说，有 $d+3$ 个选项，因此我们需要 $\log_2(d+3)$ 比特来描述每个块。

假设每个内部节点有两个子节点，[1]，不难证明这是一个无前缀编码的树，并且具有n个节点的树的描述长度是$(n+1)\log_2(d+3)$。

通过定理7.7，我们得到，在大小为m的样本中，至少以$1-\delta$的概率，对于每一个n和每一个具有n个节点的决策树$h \in \mathcal{H}$，都成立。

$$L_{\mathcal{D}}(h) \leq L_S(h) + \sqrt{\frac{(n+1)\log_2(d+3) + \log(2/\delta)}{2m}}. \tag{18.1}$$

这个界限进行权衡：一方面，我们期望更大的、更复杂的决策树具有更小的训练风险$L_S(h)$，但相应的n值将更大。另一方面，较小的决策树将具有较小的n值，但$L_S(h)$可能更大。我们的希望（或先验知识）是我们可以找到一个既具有低经验风险$L_S(h)$，又具有节点数n不太高的决策树。我们的界限表明，这样的树将具有低真实风险$L_{\mathcal{D}}(h)$。

18.2 Decision Tree Algorithms

方程（18.1）中给出的 $L_{\mathcal{D}}(h)$ 的界限暗示了一个决策树的学习规则——寻找一个使方程（18.1）右侧最小化的树。不幸的是，解决这个问题的计算量很大。因此，实际的决策树学习算法基于诸如贪婪方法之类的启发式算法，其中树是逐步构建的，并且在构建每个节点时做出局部最优决策。这样的算法不能保证返回全局最优决策树，但在实践中往往表现良好。

一个用于生长决策树的通用框架如下。我们从一棵只有一个叶子的树（根节点）开始，并根据训练集中所有标签的多数投票为这个叶子分配一个标签。现在我们进行一系列迭代。在每个迭代中，我们检查分割单个叶子的效果。我们定义一些“增益”度量，以量化由于这种分割而带来的改进。然后，在所有可能的分割中，我们选择最大化增益的那个进行分割，或者选择不分割叶子。

在以下内容中，我们提供了一个可能的实现。它基于一个名为“ID3”（代表“Iterative Dichotomizer 3”）的流行决策树算法。我们描述了该算法在二元特征情况下的应用，即，$\mathcal{X} = \{0,1\}^d$，

[1] We may assume this without loss of generality, because if a decision node has only one child, we can replace the node by its child without affecting the predictions of the decision tree.

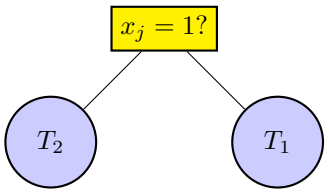
[2] More precisely, if NP≠P then no algorithm can solve Equation (18.1) in time polynomial in n, d, and m.

因此，所有分割规则的形式为 $1_{[x_i=1]}$，对于某些特征 $i \in [d]$。我们在第 18.2.3 节讨论实值特征的情况。

算法通过递归调用工作，初始调用为 $\texttt{ID3}(S, [d])$，并返回一个决策树。在下面的伪代码中，我们使用对过程 $\texttt{Gain}(S, i)$ 的调用，该过程接收一个训练集 S 和一个索引 i，并根据第 i 个特征评估树的分割增益。我们在第 18.2.1 节中描述了几个增益度量。

$\texttt{ID3}(S, A)$

INPUT: training set S, feature subset $A \subseteq [d]$
if all examples in S are labeled by 1, return a leaf 1
if all examples in S are labeled by 0, return a leaf 0
if $A = \emptyset$, return a leaf whose value = majority of labels in S
else :
  Let $j = \text{argmax}_{i \in A} \texttt{Gain}(S, i)$
  if all examples in S have the same label
    Return a leaf whose value = majority of labels in S
  else
    Let T_1 be the tree returned by $\texttt{ID3}(\{(\mathbf{x}, y) \in S : x_j = 1\}, A \setminus \{j\})$.
    Let T_2 be the tree returned by $\texttt{ID3}(\{(\mathbf{x}, y) \in S : x_j = 0\}, A \setminus \{j\})$.
    Return the tree:



18.2.1 Implementations of the Gain Measure

不同的算法使用 $\texttt{Gain}(S, i)$ 的不同实现。这里我们介绍了三种。我们使用符号 $\mathbb{P}_S[F]$ 来表示相对于 S 上的均匀分布，一个事件发生的概率。

Train Error: 增益的最简单定义是训练错误的减少。形式上，令 $C(a) = \min\{a, 1-a\}$。注意，在按特征 i 分割之前的训练错误是 $C(\mathbb{P}_S[y = 1])$，因为我们对标签进行了多数投票。同样，在按特征 i 分割之后的错误是

$$\mathbb{P}_S[x_i = 1]\, C(\mathbb{P}_S[y = 1 | x_i = 1]) + \mathbb{P}_S[x_i = 0] C(\mathbb{P}_S[y = 1 | x_i = 0]).$$

因此，我们可以定义 Gain 为这两个数的差，即，

$$\begin{aligned}\texttt{Gain}(S, i) := C(\mathbb{P}_S[y = 1]) \\ - \left(\mathbb{P}_S[x_i = 1]\, C(\mathbb{P}_S[y = 1 | x_i = 1]) + \mathbb{P}_S[x_i = 0] C(\mathbb{P}_S[y = 1 | x_i = 0])\right).\end{aligned}$$

Information Gain: 另一种在Quinlan（1993）的ID3和C4.5算法中使用的流行增益度量是信息增益。信息增益是分割前后标签熵的差异，通过将前一个表达式中的函数C替换为熵函数来实现，

$$C(a) = -a\log(a) - (1-a)\log(1-a).$$

Gini Index: 另一种增益的定义，该定义被Breiman、Friedman、Olshen和Stone（1984年）的CART算法所使用，是基尼指数，

$$C(a) = 2a(1-a).$$

信息增益和Gini指数都是训练误差的平滑凹上界。这些性质在某些情况下可能是有利的（例如，参见Kearns & Mansour（1996））。

18.2.2 Pruning

ID3算法之前描述的仍然存在一个大问题：返回的树通常非常大。这样的树可能具有较低的实证风险，但它们的真实风险往往会很高——根据我们的理论分析和实践。一种解决方案是限制ID3的迭代次数，导致具有有限节点数的树。另一种常见解决方案是在树构建后$prune$它，希望将其减小到更小的树，但仍然具有类似的实证误差。从理论上讲，根据方程（18.1）中的界限，如果我们能将n减小很多而不大幅增加$L_S(h)$，我们很可能会得到一个具有较小真实风险的决策树。

通常，剪枝是通过在树上的自下而上的遍历来执行的。每个节点可能被其子树之一或一个叶子节点替换，这取决于某些关于 $L_{\mathcal{D}}(h)$ (的界限或估计，例如，方程（18.1)）中的界限。以下给出了一个常见模板的伪代码。

Generic Tree Pruning Procedure

input:
  function $f(T, m)$ (bound/estimate for the generalization error of a decision tree T, based on a sample of size m),
  tree T.
foreach node j in a bottom-up walk on T (from leaves to root):
  find T' which minimizes $f(T', m)$, where T' is any of the following:
    the current tree after replacing node j with a leaf 1.
    the current tree after replacing node j with a leaf 0.
    the current tree after replacing node j with its left subtree.
    the current tree after replacing node j with its right subtree.
    the current tree.
  let $T := T'$.

18.2.3 Threshold-Based Splitting Rules for Real-Valued Features

在上一节中，我们描述了一个在假设特征是二进制且分割规则为形式$1_{[x_i=1]}$的情况下生长决策树的算法。现在，我们将这个结果扩展到实值特征和基于阈值的分割规则的情况，即$1_{[x_i<\theta]}$。这种分割规则产生决策树桩，我们在第10章中对其进行了研究。

基本思路是将问题简化为二元特征的情况，如下所示。令 $\mathbf{x}_1, \ldots, \mathbf{x}_m$ 为训练集的实例。对于每个实值特征 i，对实例进行排序，使得 $x_{1,i} \leq \cdots \leq x_{m,i}$。定义一组阈值 $\theta_{0,i}, \ldots, \theta_{m+1,i}$，使得 $\theta_{j,i} \in (x_{j,i}, x_{j+1,i})$ (，其中我们使用约定 $x_{0,i} = -\infty$ 和 $x_{m+1,i} = \infty$)。最后，对于每个 i 和 j，我们定义二元特征 $1_{[x_i<\theta_{j,i}]}$。一旦我们构建了这些二元特征，就可以运行上一节中描述的 ID3 程序。很容易验证，对于任何基于阈值的分割规则的原实值特征的决策树，都存在一个具有相同训练错误和相同节点数的基于构建的二元特征的决策树。

如果原始实值特征的个数为 d，样本个数为 m，则构建的二值特征的个数变为 dm。因此，计算每个特征的 `Gain` 可能需要 $O(dm^2)$ 次操作。然而，使用更巧妙的实现，运行时间可以减少到 $O(dm \log(m))$。这个想法与第10.1.1节中描述的ERM决策树桩的实现类似。

18.3 Random Forests

如前所述，任意大小的决策树类的VC维是无限的。因此，我们限制了决策树的大小。另一种减少过拟合危险的方法是通过构建树集合。特别是，在以下内容中，我们描述了Breiman（2001）引入的*random forests*方法。

随机森林是由一系列决策树组成的分类器，其中每棵树都是通过对训练集 S 应用算法 A 和一个额外的随机向量 θ 来构建的，其中 θ 是从某个分布中独立同分布采样的。随机森林的预测是通过在单个树的预测上进行多数投票获得的。

要指定特定的随机森林，我们需要定义算法 A 和 θ 上的分布。有许多方法可以做到这一点，这里我们描述一种特定选项。我们如下生成 θ。首先，我们从 S 中抽取一个有放回的随机子样本；也就是说，我们使用在 S 上的均匀分布来抽取一个大小为 m' 的新训练集 S'。其次，我们构建一个序列 $I_1, I_2, \ldots$，其中每个 I_t 是 $[d]$ 的一个大小为 k 的子集，该子集通过从 $[d]$ 中随机均匀抽取元素生成。所有这些随机变量形成向量 θ。然后，

算法 A 基于样本 S' 增长决策树（例如，使用 ID3 算法），在算法的每个分割阶段，算法被限制在从集合 I_t 中选择一个最大化增益的特征。直观上，如果 k 很小，这种限制可能会防止过拟合。

18.4 Summary

决策树是非常直观的预测器。通常，如果一个人工智能程序创建一个预测器，它将看起来像一棵决策树。我们已经证明了具有 k 叶子的决策树的VC维数为 k ，并提出了学习决策树的MDL范式。决策树的主要问题是它们难以从计算上学习；因此，我们描述了几个用于训练它们的启发式程序。

18.5 Bibliographic Remarks

许多学习决策树的算法（如ID3和C4.5）由Quinlan（1986）推导得出。CART算法归功于Breiman等人（1984）。随机森林由Breiman（2001）引入。关于进一步阅读，我们建议读者参考（Hastie，Tibshirani & Friedman 2001，Rokach 2007）。

证明训练决策树困难性的证明在Hyafil & Rivest（1976）中给出。

18.6 Exercises

1. 1. 证明任何二分类器 $h : \{0, 1\}^d \mapsto \{0, 1\}$ 可以实现为一个高度最多为 $d + 1$ 的决策树，其内部节点形式为 ($x_i = 0$?)，对于某些 $i \in \{1, \ldots, d\}$。

   2. 推断在域 $\{0, 1\}^d$ 上决策树类别的VC维数为 2^d。

2. **(Suboptimality of ID3)**
   考虑以下训练集，其中 $\mathcal{X} = \{0, 1\}^3$ 和 $\mathcal{Y} = \{0, 1\}$:

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

   假设我们希望使用这个训练集来构建一个深度为2的决策树（即，对于每个输入，我们允许提出两个形式为($x_i = 0$?)的问题，然后在决定标签之前）。

1. 假设我们将ID3算法运行到深度2（即根据算法选择根节点及其子节点，但不是继续递归，而是停止并根据每个子树中的多数标签选择叶子节点）。假设用于衡量每个特征质量的子程序基于熵函数（因此我们衡量 $information\ gain$），并且如果两个特征获得相同的分数，则任意选择其中一个。证明所得到的决策树的训练误差至少为$1/4$。

2. 找到一个深度为2的决策树，达到零训练错误。

19 Nearest Neighbor

最近邻算法是所有机器学习算法中最简单的之一。其思想是记住训练集，然后根据训练集中与其最近的邻居的标签来预测任何新实例的标签。这种方法背后的原理是基于这样的假设：用于描述领域点的特征与其标签相关，使得附近的点很可能具有相同的标签。此外，在某些情况下，即使训练集很大，找到最近邻也可以非常快（例如，当训练集是整个网络且距离基于链接时）。

请注意，与迄今为止我们讨论的算法范式（如ERM、SRM、MDL或RLM）不同，这些范式由某些假设类$\mathcal{H}$确定，最近邻方法无需在预定义的函数类中搜索预测器，就能在任何测试点上确定一个标签。

在这一章中，我们描述了用于分类和回归问题的最近邻方法。我们分析了它们在二分类简单情况下的性能，并讨论了实现这些方法的效率。

19.1 k Nearest Neighbors

在整个章节中，我们假设我们的实例域$\mathcal{X}$被赋予了一个度量函数ρ。也就是说，$\rho : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$是一个函数，它返回$\mathcal{X}$中任意两个元素之间的距离。例如，如果$\mathcal{X} = \mathbb{R}^d$，那么$\rho$可以是欧几里得距离，
$\rho(\mathbf{x}, \mathbf{x}') = \|\mathbf{x} - \mathbf{x}'\| = \sqrt{\sum_{i=1}^{d}(x_i - x'_i)^2}$。

让$S = (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$为一个训练示例序列。对于每个$\mathbf{x} \in \mathcal{X}$，让$\pi_1(\mathbf{x}), \ldots, \pi_m(\mathbf{x})$根据它们与$\mathbf{x}$、$\rho(\mathbf{x}, \mathbf{x}_i)$的距离对$\{1, \ldots, m\}$进行重新排序。也就是说，对于所有$i < m$，

$$\rho(\mathbf{x}, \mathbf{x}_{\pi_i(\mathbf{x})}) \leq \rho(\mathbf{x}, \mathbf{x}_{\pi_{i+1}(\mathbf{x})}).$$

对于一个数k，二分类的k-NN规则定义如下：

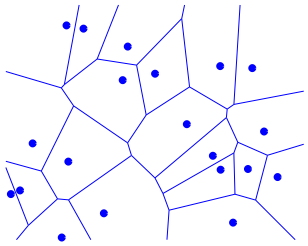


Figure 19.1 一个1-NN规则的决策边界的插图。图中表示的点为样本点，任何新点的预测标签将是它所属单元格中心样本点的标签。这些单元格被称为空间中的Voronoi划分。

k-NN

input: 每个点 $\mathbf{x} \in \mathcal{X}$ 的训练样本 $S = (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$ **output:**，返回 $\{y_{\pi_i(\mathbf{x})}$ 中的多数标签：$i \leq k\}$

当 $k = 1$ 时，我们有 1-NN 规则：

$$h_S(\mathbf{x}) \;=\; y_{\pi_1(\mathbf{x})}.$$

一个关于1-NN规则的几何说明如图19.1所示。

对于回归问题，即$\mathcal{Y} = \mathbb{R}$，可以定义预测为k个最近邻的平均目标。也就是说，$h_S(\mathbf{x}) = \frac{1}{k}\sum_{i=1}^{k} y_{\pi_i(\mathbf{x})}$。更一般地，对于某些函数$\phi$：$(\mathcal{X} \times \mathcal{Y})^k \to \mathcal{Y}$，关于$\phi$的$k$-NN规则是：

$$h_S(\mathbf{x}) \;=\; \phi\left((\mathbf{x}_{\pi_1(\mathbf{x})}, y_{\pi_1(\mathbf{x})}), \ldots, (\mathbf{x}_{\pi_k(\mathbf{x})}, y_{\pi_k(\mathbf{x})})\right). \tag{19.1}$$

它很容易验证，我们可以通过多数标签（对于分类）或通过平均目标（对于回归）来表示预测，如公式（19.1）所示，通过适当选择 ϕ。这种通用性可以导致其他规则；例如，如果 $\mathcal{Y} = \mathbb{R}$，我们可以根据与 $\mathbf{x}$ 的距离对目标进行加权平均：

$$h_S(\mathbf{x}) \;=\; \sum_{i=1}^{k} \frac{\rho(\mathbf{x}, \mathbf{x}_{\pi_i(\mathbf{x})})}{\sum_{j=1}^{k} \rho(\mathbf{x}, \mathbf{x}_{\pi_j(\mathbf{x})})}\, y_{\pi_i(\mathbf{x})}.$$

19.2 Analysis

由于NN规则是如此自然的学习方法，它们的泛化特性已经被广泛研究。大多数先前结果都是渐近一致性结果，分析了当样本大小为m时NN规则的性能。

趋向于无穷大，收敛速度取决于基础分布。正如我们在第7.4节所论证的，这种分析并不令人满意。人们希望从有限的训练样本中学习，并理解泛化性能作为有限训练集大小和数据分布的先验假设的函数。因此，我们提供对1-NN规则的有限样本分析，展示错误如何随着m的变化而减少，以及它如何依赖于分布的性质。我们还将解释如何将分析推广到任意k值的k-NN规则。特别是，分析指定了达到真实误差$2L_{\mathcal{D}}(h^\star) + \epsilon$所需的示例数量，其中$h^\star$是贝叶斯最优假设，假设标签规则“表现良好”（在稍后我们将定义的意义上）。

19.2.1 A Generalization Bound for the 1-NN Rule

我们现在分析二分类中1-NN规则的真正误差，即0-1损失下的$\mathcal{Y} = \{0, 1\}$和$\ell(h, (\mathbf{x}, y)) = 1_{[h(\mathbf{x}) \neq y]}$。在整个分析过程中，我们还假设$\mathcal{X} = [0, 1]^d$和$\rho$是欧几里得距离。

我们首先引入一些符号。令 $\mathcal{D}$ 是 $\mathcal{X} \times \mathcal{Y}$ 上的一个分布。令 $\mathcal{D}_{\mathcal{X}}$ 表示在 $\mathcal{X}$ 上诱导的边缘分布，并令 $\eta : \mathbb{R}^d \to \mathbb{R}$ 是标签上的条件概率 [1]，即，

$$\eta(\mathbf{x}) = \mathbb{P}[y = 1|\mathbf{x}].$$

回忆一下，贝叶斯最优规则（即最小化所有函数上的 $L_{\mathcal{D}}(h)$ 的假设）是

$$h^\star(\mathbf{x}) = \mathbb{1}_{[\eta(\mathbf{x}) > 1/2]}.$$

我们假设条件概率函数 η 对于某个 $c > 0$ 是 c-Lipschitz 的。也就是说，对于所有 $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$，$|\eta(\mathbf{x}) - \eta(\mathbf{x}')| \leq c \|\mathbf{x} - \mathbf{x}'\|$。换句话说，这个假设意味着如果两个向量彼此接近，那么它们的标签很可能相同。

以下引理将条件概率函数的Lipschitz性质应用于将1-NN规则的真误差上界化为每个测试实例与其在训练集中最近邻之间的期望距离的函数。

LEMMA 19.1 *Let $\mathcal{X} = [0, 1]^d, \mathcal{Y} = \{0, 1\}$, and $\mathcal{D}$ be a distribution over $\mathcal{X} \times \mathcal{Y}$ for which the conditional probability function, η, is a c-Lipschitz function. Let $S = (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$ be an i.i.d. sample and let h_S be its corresponding 1-NN hypothesis. Let $h^\star$ be the Bayes optimal rule for η. Then,*

$$\mathop{\mathbb{E}}_{S \sim \mathcal{D}^m}[L_{\mathcal{D}}(h_S)] \leq 2\, L_{\mathcal{D}}(h^\star) + c \mathop{\mathbb{E}}_{S \sim \mathcal{D}^m, \mathbf{x} \sim \mathcal{D}}[\|\mathbf{x} - \mathbf{x}_{\pi_1(\mathbf{x})}\|].$$

[1] Formally, $\mathbb{P}[y = 1|\mathbf{x}] = \lim_{\delta \to 0} \frac{\mathcal{D}(\{(\mathbf{x}', 1) : \mathbf{x}' \in B(\mathbf{x}, \delta)\})}{\mathcal{D}(\{(\mathbf{x}', y) : \mathbf{x}' \in B(\mathbf{x}, \delta), y \in \mathcal{Y}\})}$, where $B(\mathbf{x}, \delta)$ is a ball of radius δ centered around $\mathbf{x}$.

Proof 由于 $L_{\mathcal{D}}(h_S) = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}[1_{[h_S(\mathbf{x})\neq y]}]$，我们得到 $\mathbb{E}_S[L_{\mathcal{D}}(h_S)]$ 是采样训练集 S 和一个额外示例 $(\mathbf{x}, y)$ 的概率，使得 $\pi_1(\mathbf{x})$ 的标签与 y 不同。换句话说，我们首先根据 $\mathcal{D}_{\mathcal{X}}$ 采样 m 个未标记示例 $S_x = (\mathbf{x}_1, \ldots, \mathbf{x}_m)$，然后采样一个额外的未标记示例 $\mathbf{x} \sim \mathcal{D}_{\mathcal{X}}$，接着找到 S_x 中 $\mathbf{x}$ 的最近邻 $\pi_1(\mathbf{x})$，最后采样 $y \sim \eta(\mathbf{x})$ 和 $y_{\pi_1(\mathbf{x})} \sim \eta(\pi_1(\mathbf{x}))$。由此可得

$$\begin{aligned}\mathop{\mathbb{E}}_{S}[L_{\mathcal{D}}(h_S)] &= \mathop{\mathbb{E}}_{S_x\sim\mathcal{D}_{\mathcal{X}}^m,\mathbf{x}\sim\mathcal{D}_{\mathcal{X}},y\sim\eta(\mathbf{x}),y'\sim\eta(\pi_1(\mathbf{x}))}[\mathbb{1}_{[y\neq y']}] \\ &= \mathop{\mathbb{E}}_{S_x\sim\mathcal{D}_{\mathcal{X}}^m,\mathbf{x}\sim\mathcal{D}_{\mathcal{X}}}\left[\mathop{\mathbb{P}}_{y\sim\eta(\mathbf{x}),y'\sim\eta(\pi_1(\mathbf{x}))}[y\neq y']\right]. \qquad (19.2)\end{aligned}$$

我们接下来对任意两个域点 $\mathbf{x}, \mathbf{x}'$ 的 $\mathbb{P}_{y\sim\eta(\mathbf{x}),y'\sim\eta(\mathbf{x}')}[y \neq y']$ 进行上界估计：

$$\begin{aligned}\mathop{\mathbb{P}}_{y\sim\eta(\mathbf{x}),y'\sim\eta(\mathbf{x}')}[y\neq y'] &= \eta(\mathbf{x}')(1-\eta(\mathbf{x})) + (1-\eta(\mathbf{x}'))\eta(\mathbf{x}) \\ &= (\eta(\mathbf{x}) - \eta(\mathbf{x}) + \eta(\mathbf{x}'))(1-\eta(\mathbf{x})) \\ &\quad + (1-\eta(\mathbf{x}) + \eta(\mathbf{x}) - \eta(\mathbf{x}'))\eta(\mathbf{x}) \\ &= 2\eta(\mathbf{x})(1-\eta(\mathbf{x})) + (\eta(\mathbf{x}) - \eta(\mathbf{x}'))(2\eta(\mathbf{x}) - 1).\end{aligned}$$

使用 $|2\eta(\mathbf{x}) - 1| \leq 1$ 以及假设 η 是 c-Lipschitz，我们得到概率至多为：

$$\mathop{\mathbb{P}}_{y\sim\eta(\mathbf{x}),y'\sim\eta(\mathbf{x}')}[y\neq y'] \leq 2\eta(\mathbf{x})(1-\eta(\mathbf{x})) + c\,\|\mathbf{x}-\mathbf{x}'\|.$$

将此代入方程（19.2）我们得出结论

$$\mathop{\mathbb{E}}_{S}[L_{\mathcal{D}}(h_S)] \;\leq\; \mathop{\mathbb{E}}_{\mathbf{x}}[2\eta(\mathbf{x})(1-\eta(\mathbf{x}))] + c\;\mathop{\mathbb{E}}_{S,\mathbf{x}}[\|\mathbf{x}-\mathbf{x}_{\pi_1(\mathbf{x})}\|].$$

最后，贝叶斯最优分类器的误差是

$$L_{\mathcal{D}}(h^\star) \;=\; \mathop{\mathbb{E}}_{\mathbf{x}}[\min\{\eta(\mathbf{x}), 1-\eta(\mathbf{x})\}] \geq \mathop{\mathbb{E}}_{\mathbf{x}}[\eta(\mathbf{x})(1-\eta(\mathbf{x}))].$$

结合前两个不等式，我们完成了证明。 □

下一步是界定随机 $\mathbf{x}$ 与其最近元素在 S 之间的期望距离。我们首先需要以下一般概率引理。该引理将未受到随机样本影响的子集的概率权重界定为样本大小的函数。

LEMMA 19.2 *Let $C_1, \ldots, C_r$ be a collection of subsets of some domain set, $\mathcal{X}$. Let S be a sequence of m points sampled i.i.d. according to some probability distribution, $\mathcal{D}$ over $\mathcal{X}$. Then,*

$$\mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}\left[\sum_{i:C_i\cap S=\emptyset}\mathbb{P}[C_i]\right] \;\leq\; \frac{r}{m\,e}.$$

Proof 从期望的线性，我们可以重写：

$$\mathop{\mathbb{E}}_{S}\left[\sum_{i:C_i\cap S=\emptyset}\mathbb{P}[C_i]\right] \;=\; \sum_{i=1}^{r}\mathbb{P}[C_i]\mathop{\mathbb{E}}_{S}\left[\mathbb{1}_{[C_i\cap S=\emptyset]}\right].$$

接下来，对于每个 i 我们有

$$\mathop{\mathbb{E}}_{S}\left[\mathbb{1}_{[C_i\cap S=\emptyset]}\right] = \mathop{\mathbb{P}}_{S}[C_i\cap S=\emptyset] = (1-\mathbb{P}[C_i])^m \le e^{-\mathbb{P}[C_i]\,m}.$$

将前两个方程式相加，我们得到

$$\mathop{\mathbb{E}}_{S}\left[\sum_{i:C_i\cap S=\emptyset}\mathbb{P}[C_i]\right] \;\le\; \sum_{i=1}^{r}\mathbb{P}[C_i]\,e^{-\mathbb{P}[C_i]\,m} \;\le\; r\,\max_i \mathbb{P}[C_i]\,e^{-\mathbb{P}[C_i]\,m}.$$

最后，通过标准微积分，$\max_a ae^{-ma} \le \frac{1}{me}$，这就完成了证明。

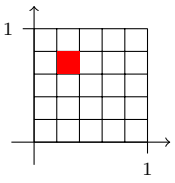
□

配备了前面的引理，我们现在可以陈述并证明本节的主要结果——1-NN学习规则的期望误差上界。

THEOREM 19.3 *Let $\mathcal{X}=[0,1]^d, \mathcal{Y}=\{0,1\}$, and $\mathcal{D}$ be a distribution over $\mathcal{X}\times\mathcal{Y}$ for which the conditional probability function, η, is a c-Lipschitz function. Let h_S denote the result of applying the 1-NN rule to a sample $S\sim\mathcal{D}^m$. Then,*

$$\mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}[L_{\mathcal{D}}(h_S)] \le 2\,L_{\mathcal{D}}(h^\star) + 4\,c\sqrt{d}\,m^{-\frac{1}{d+1}}.$$

Proof 修复一些 $\epsilon=1/T$，对于某些整数 T，设 $r=T^d$ 和设 $C_1,\ldots,C_r$ 是集合 $\mathcal{X}$ 的覆盖，使用长度为 ϵ 的箱子：即对于每一个 $(\alpha_1,\ldots,\alpha_d)\in[T]^d$，存在一个形如 $\{\mathbf{x}$ 的集合 C_i：$\forall j, x_j\in[(\alpha_j-1)/T,\alpha_j/T]\}$。以下给出了 $d=2$，$T=5$ 以及对应于 $\alpha=(2,4)$ 的集合的插图。



对于每个同一框中的 $\mathbf{x},\mathbf{x}'$，我们有 $\|\mathbf{x}-\mathbf{x}'\|\le\sqrt{d}\,\epsilon$。否则，$\|\mathbf{x}-\mathbf{x}'\|\le\sqrt{d}$。因此，

$$\mathop{\mathbb{E}}_{\mathbf{x},S}[\|\mathbf{x}-\mathbf{x}_{\pi_1(\mathbf{x})}\|] \le \mathop{\mathbb{E}}_{S}\left[\mathbb{P}\left[\bigcup_{i:C_i\cap S=\emptyset}C_i\right]\sqrt{d} + \mathbb{P}\left[\bigcup_{i:C_i\cap S\neq\emptyset}C_i\right]\epsilon\sqrt{d}\right],$$

并且通过将引理19.2与平凡的界限 $\mathbb{P}[\bigcup_{i:C_i\cap S\neq\emptyset}C_i]\le 1$ 结合，我们得到：{v*}

$$\mathop{\mathbb{E}}_{\mathbf{x},S}[\|\mathbf{x}-\mathbf{x}_{\pi_1(\mathbf{x})}\|] \le \sqrt{d}\left(\tfrac{r}{me}+\epsilon\right).$$

由于箱子的数量是 $r = (1/\epsilon)^d$，我们得到

$$\mathop{\mathbb{E}}_{S,\mathbf{x}}[\|\mathbf{x} - \mathbf{x}_{\pi_1(\mathbf{x})}\|] \leq \sqrt{d}\left(\frac{2^d\,\epsilon^{-d}}{m\,e} + \epsilon\right).$$

结合前面的内容与引理19.1，我们得到：{v*}

$$\mathop{\mathbb{E}}_{S}[L_{\mathcal{D}}(h_S)] \leq 2\,L_{\mathcal{D}}(h^\star) + c\sqrt{d}\left(\frac{2^d\,\epsilon^{-d}}{m\,e} + \epsilon\right).$$

最后，设置 $\epsilon = 2\,m^{-1/(d+1)}$ 并注意

$$\begin{aligned}\frac{2^d\,\epsilon^{-d}}{m\,e} + \epsilon &= \frac{2^d\,2^{-d}\,m^{d/(d+1)}}{m\,e} + 2\,m^{-1/(d+1)} \\ &= m^{-1/(d+1)}(1/e + 2) \leq 4m^{-1/(d+1)}\end{aligned}$$

我们得出证明。 □

定理表明，如果我们首先固定数据生成分布，然后让 m 趋向无穷大，那么 1-NN 规则的误差收敛到贝叶斯误差的两倍。该分析可以推广到更大的 k 值，表明 k-NN 规则的期望误差收敛到 $(1 + \sqrt{8/k})$ 倍的贝叶斯分类器误差。这已在定理 19.5 中形式化，其证明留作指导练习。

19.2.2 The "Curse of Dimensionality"

定理19.3中给出的上界随着 c (η) 的Lipschitz系数和 d，即域集 $\mathcal{X}$ 的欧几里得维度而增长。事实上，很容易看出，定理19.3中最后一项小于 ϵ 的一个必要条件是 $m \geq (4\,c\sqrt{d}/\epsilon)^{d+1}$。也就是说，训练集的大小应该随着维度的增加而指数增长。以下定理告诉我们，这不仅仅是我们上界的偶然现象，对于某些分布，这种数量的示例确实是使用NN规则进行学习所必需的。

THEOREM 19.4 *For any $c > 1$, and every learning rule, L, there exists a distribution over $[0,1]^d \times \{0,1\}$, such that $\eta(\mathbf{x})$ is c-Lipschitz, the Bayes error of the distribution is 0, but for sample sizes $m \leq (c+1)^d/2$, the true error of the rule L is greater than $1/4$.*

Proof 修复 c 和 d 的任何值。令 G_c^d 为 $[0,1]^d$ 上的网格，网格上各点之间的距离为 $1/c$。也就是说，网格上的每个点都是 $(a_1/c, \ldots, a_d/c)$ 的形式，其中 a_i 在 $\{0, \ldots, c-1, c\}$ 中。注意，由于该网格上任意两个不同的点至少相距 $1/c$，因此任何函数 $\eta : G_C^D \to [0,1]$ 都是 c-Lipschitz 函数。因此，所有 c-Lipschitz 函数的集合包含该域上的 *all* 二值函数的集合。因此，我们可以调用 No-Free-Lunch 结果（定理 5.1）来获得学习该类所需的样本大小的下界。网格上的点数为 $(c+1)^d$；因此，如果 $m < (c+1)^d/2$，定理 5.1 意味着我们所追求的下界。

□

指数依赖于维度的关系被称为$curse\ of\ di\text{-}\ mensionality$。正如我们所见，当示例数量小于$\Omega((c+1)^d)$时，1-NN规则可能会失败。因此，虽然1-NN规则没有将自己限制在预定义的假设集上，但它仍然依赖于某些先验知识——它的成功取决于假设，即基础分布的维度和Lipschitz常数η不是太高。

19.3 Efficient Implementation*

最近邻是一种通过记忆学习类型的规则。它需要存储整个训练数据集，并且在测试时，我们需要扫描整个数据集以找到邻居。因此，应用NN规则的时间是$\Theta(d\,m)$。这导致测试时的计算成本高昂。

当 d 较小时，计算几何领域的多个结果提出了数据结构，这些数据结构能够使 NN 规则在时间 $o(d^{O(1)}\log(m))$ 内应用。然而，这些数据结构所需的空间大约为 $m^{O(d)}$，这使得这些方法对于 d 较大的值来说不切实际。

为了克服这个问题，建议通过允许进行 $approximate$ 搜索来改进搜索方法。形式上，一个 r 近似搜索过程保证在距离最近邻最多 r 倍的距离内检索到一个点。NN 的三种流行近似算法是 kd 树、球树和局部敏感哈希（LSH）。例如，我们建议读者参考（Shakhnarovich，Darrell & Indyk 2006）。

19.4 Summary

k-NN规则是一个非常简单的学习算法，它依赖于“看起来相似的东西必须相似”的假设。我们使用条件概率的Lipschitz性质来形式化这种直觉。我们已经证明，在足够大的训练集下，1-NN的风险被限制在贝叶斯最优规则风险的两倍以内。我们还推导出一个下界，表明“维度诅咒”——所需的样本大小可能随着维度的增加而指数增长。因此，在实践中，NN通常在降维预处理步骤之后执行。我们将在第23章中讨论降维技术。

19.5 Bibliographic Remarks

Cover & Hart (1967) 对 1-NN 进行了首次分析，表明在温和条件下其风险收敛到贝叶斯最优错误的两倍。根据Stone (1977)的一个引理，Devroye & Gy{v*}orfi (1985) 已经证明了 {v*}0-NN 规则

一致性（相对于从$\mathbb{R}^d$到$\{0,1\}$的所有函数类假设）。Devroye等人（1996年）的书中给出了分析的良好表述。在此，我们给出了一个有限样本保证，该保证明确强调了分布的先验假设。参见第7.4节，讨论一致性结果。最后，Gottlieb、Kontorovich和Krauthgamer（2010年）推导出NN的另一个有限样本界限，该界限与VC界限更相似。

19.6 Exercises

在这个练习中，我们将证明 **k-NN** 规则下的以下定理。

THEOREM 19.5 *Let $\mathcal{X} = [0,1]^d, \mathcal{Y} = \{0,1\}$, and $\mathcal{D}$ be a distribution over $\mathcal{X} \times \mathcal{Y}$ for which the conditional probability function, η, is a c-Lipschitz function. Let h_S denote the result of applying the k-NN rule to a sample $S \sim \mathcal{D}^m$, where $k \geq 10$. Let $h^\star$ be the Bayes optimal hypothesis. Then,*

$$\mathop{\mathbb{E}}_{S}[L_{\mathcal{D}}(h_S)] \leq \left(1 + \sqrt{\frac{8}{k}}\right) L_{\mathcal{D}}(h^\star) + \left(6\, c\sqrt{d} + k\right) m^{-1/(d+1)}.$$

1. 证明以下引理。

LEMMA 19.6 *Let $C_1, \ldots, C_r$ be a collection of subsets of some domain set, $\mathcal{X}$. Let S be a sequence of m points sampled i.i.d. according to some probability distribution, $\mathcal{D}$ over $\mathcal{X}$. Then, for every $k \geq 2$,*

$$\mathop{\mathbb{E}}_{S \sim \mathcal{D}^m}\left[\sum_{i:|C_i \cap S|<k} \mathbb{P}[C_i]\right] \;\leq\; \frac{2rk}{m}.$$

Hints:

- 证明以下公式：

$$\mathop{\mathbb{E}}_{S}\left[\sum_{i:|C_i \cap S|<k} \mathbb{P}[C_i]\right] \;=\; \sum_{i=1}^{r} \mathbb{P}[C_i] \mathop{\mathbb{P}}_{S}\left[|C_i \cap S| < k\right].$$

- 修复一些 i 并假设 $k < \mathbb{P}[C_i]\, m/2$。使用切诺夫不等式来证明

$$\mathop{\mathbb{P}}_{S}\left[|C_i \cap S| < k\right] \leq \mathop{\mathbb{P}}_{S}\left[|C_i \cap S| < \mathbb{P}[C_i]m/2\right] \leq e^{-\mathbb{P}[C_i]\, m/8}.$$

- 使用不等式 $\max_a ae^{-ma} \leq \frac{1}{me}$ 来证明对于这样的 i，我们有

$$\mathbb{P}[C_i] \mathop{\mathbb{P}}_{S}\left[|C_i \cap S| < k\right] \leq \mathbb{P}[C_i] e^{-\mathbb{P}[C_i]\, m/8} \leq \frac{8}{me}.$$

- 结论通过使用以下事实得出：对于情况 $k \geq \mathbb{P}[C_i]\, m/2$，我们明显有：

$$\mathbb{P}[C_i] \mathop{\mathbb{P}}_{S}\left[|C_i \cap S| < k\right] \leq \mathbb{P}[C_i] \leq \frac{2k}{m}.$$

2. 我们使用符号 $y \sim p$ 作为“y 是一个期望值为 p 的伯努利随机变量”的简称。证明以下引理：

LEMMA 19.7 *Let $k \geq 10$ and let $Z_1, \ldots, Z_k$ be independent Bernoulli random variables with $\mathbb{P}[Z_i = 1] = p_i$. Denote $p = \frac{1}{k}\sum_i p_i$ and $p' = \frac{1}{k}\sum_{i=1}^{k} Z_i$. Show that*

$$\mathop{\mathbb{E}}_{Z_1,\ldots,Z_k} \mathop{\mathbb{P}}_{y\sim p}[y \neq \mathbb{1}_{[p'>1/2]}] \leq \left(1 + \sqrt{\frac{8}{k}}\right) \mathop{\mathbb{P}}_{y\sim p}[y \neq \mathbb{1}_{[p>1/2]}].$$

Hints:

W.l.o.g. 假设 $p \leq 1/2$。然后，$\mathbb{P}_{y\sim p}[y \neq 1_{[p>1/2]}] = p$。令 $y' = 1_{[p'>1/2]}$。

- 证明以下公式：

$$\mathop{\mathbb{E}}_{Z_1,\ldots,Z_k} \mathop{\mathbb{P}}_{y\sim p}[y \neq y'] - p = \mathop{\mathbb{P}}_{Z_1,\ldots,Z_k}[p' > 1/2](1-2p).$$

- 使用切诺夫界（引理B.3）来证明

$$\mathbb{P}[p' > 1/2] \leq e^{-k\,p\,h\left(\frac{1}{2p}-1\right)},$$

  哪里

$$h(a) = (1+a)\log(1+a) - a.$$

- 为了完成引理的证明，你可以依赖以下不等式（无需证明）：对于每个 $p \in [0, 1/2]$ 和 $k \geq 10$：

$$(1-2p)\, e^{-k\,p + \frac{k}{2}(\log(2p)+1)} \;\leq\; \sqrt{\frac{8}{k}}\, p.$$

3. 修复一些 $p, p' \in [0, 1]$ 和 $y' \in \{0, 1\}$。证明

$$\mathop{\mathbb{P}}_{y\sim p}[y \neq y'] \leq \mathop{\mathbb{P}}_{y\sim p'}[y \neq y'] + |p - p'|.$$

4. 根据以下步骤完成定理的证明：

- 如同定理19.3的证明中，取六个$\epsilon > 0$，并令$C_1, \ldots, C_r$为使用长度为ϵ的盒子覆盖集合$\mathcal{X}$。对于每个与$\mathbf{x}, \mathbf{x}'$在同一盒子中的元素，我们有 $\|\mathbf{x} - \mathbf{x}'\| \leq \sqrt{d}\,\epsilon$。否则，$\|\mathbf{x} - \mathbf{x}'\| \leq 2\sqrt{d}$。证明

$$\mathop{\mathbb{E}}_{S}[L_{\mathcal{D}}(h_S)] \leq \mathop{\mathbb{E}}_{S}\left[\sum_{i:|C_i\cap S|<k} \mathbb{P}[C_i]\right] + \max_i \mathop{\mathbb{P}}_{S,(\mathbf{x},y)}\left[h_S(\mathbf{x}) \neq y \mid \forall j \in [k],\ \|\mathbf{x} - \mathbf{x}_{\pi_j(\mathbf{x})}\| \leq \epsilon\sqrt{d}\right]. \quad (19.3)$$

- 使用引理19.6对第一个项进行界定。
- 要限制第二个和项，让我们固定 $S|_x$ 和 $\mathbf{x}$，使得 $S|_x$ 中 $\mathbf{x}$ 的所有 k 邻居与 $\mathbf{x}$ 的距离至多为 $\epsilon\sqrt{d}$。不失一般性，假设 k NN 是 $\mathbf{x}_1, \ldots, \mathbf{x}_k$。表示 $p_i = \eta(\mathbf{x}_i)$ 并让 $p = \frac{1}{k}\sum_i p_i$。使用练习 3 来证明

$$\mathop{\mathbb{E}}_{y_1,\ldots,y_j} \mathop{\mathbb{P}}_{y\sim\eta(\mathbf{x})}[h_S(\mathbf{x}) \neq y] \leq \mathop{\mathbb{E}}_{y_1,\ldots,y_j} \mathop{\mathbb{P}}_{y\sim p}[h_S(\mathbf{x}) \neq y] + |p - \eta(\mathbf{x})|.$$

W.l.o.g. 假设 $p \leq 1/2$。现在使用引理19.7来证明

$$\mathop{\mathbb{P}}_{y_1,\ldots,y_j} \mathop{\mathbb{P}}_{y\sim p}[h_S(\mathbf{x}) \neq y] \leq \left(1+\sqrt{\frac{8}{k}}\right) \mathop{\mathbb{P}}_{y\sim p}[\mathbb{1}_{[p>1/2]} \neq y].$$

- 证明以下公式：

$$\mathop{\mathbb{P}}_{y\sim p}[\mathbb{1}_{[p>1/2]} \neq y] = p = \min\{p, 1-p\} \leq \min\{\eta(\mathbf{x}), 1-\eta(\mathbf{x})\} + |p-\eta(\mathbf{x})|.$$

- 将所有前面的内容结合起来，可以得到方程（19.3）的第二项被以下不等式所限制

$$\left(1+\sqrt{\frac{8}{k}}\right) L_{\mathcal{D}}(h^\star) + 3\,c\,\epsilon\,\sqrt{d}.$$

- 使用 $r = (2/\epsilon)^d$ 可以得到：

$$\mathop{\mathbb{E}}_{S}[L_{\mathcal{D}}(h_S)] \leq \left(1+\sqrt{\frac{8}{k}}\right) L_{\mathcal{D}}(h^\star) + 3\,c\,\epsilon\,\sqrt{d} + \frac{2(2/\epsilon)^d\,k}{m}.$$

设置 $\epsilon = 2m^{-1/(d+1)}$ 并使用

$$6\,c\,m^{-1/(d+1)}\,\sqrt{d} + \frac{2k}{e} m^{-1/(d+1)} \leq \left(6c\sqrt{d}+k\right) m^{-1/(d+1)}$$

总结证明。

20 Neural Networks

一个人工神经网络是受大脑中神经网络结构启发的计算模型。在简化的脑模型中，它由大量基本计算设备（神经元）组成，这些设备通过复杂的通信网络相互连接，通过这些网络，大脑能够执行高度复杂的计算。人工神经网络是模仿这种计算范例的正式计算结构。

神经网络学习在20世纪中叶被提出。它产生了一种有效的学习范式，并且最近在多个学习任务上显示出达到尖端性能。

一个神经网络可以被描述为一个有向图，其节点对应神经元，边对应它们之间的连接。每个神经元接收其输入边连接的神经元的输出加权总和。我们关注不包含环的基本图 *feedforward* 网络。

在学习的背景下，我们可以定义一个由神经网络预测器组成的假设类，其中所有假设共享网络的基本图结构，但在边的权重上有所不同。正如我们在第20.3节中将要展示的，每个在时间$T(n)$内可以实现的、关于n变量的预测器也可以表示为一个大小为$O(T(n)^2)$的神经网络预测器，其中网络的大小是其中的节点数。因此，多项式大小的神经网络假设类族足以满足所有实际的学习任务，我们的目标是学习可以高效实现的预测器。此外，在第20.4节中，我们将展示学习此类假设类的样本复杂度也以网络的大小为界。因此，这似乎是我们想要适应的最终学习范式，因为它既具有多项式样本复杂度，又在所有由高效可实现的预测器组成的假设类中具有最小的近似误差。

训练此类神经网络预测器的假设类的问题在计算上是困难的。这将在第20.5节中形式化。用于训练神经网络的广泛使用的启发式方法依赖于我们在第14章中研究的SGD框架。在那里，我们已证明如果损失函数是凸的，则SGD是一个成功的学习者。在神经网络中，损失函数高度非凸。尽管如此，我们仍然可以实施SGD算法并

希望它能找到一个合理的解决方案（正如在几个实际任务中发生的那样）。在第20.6节中，我们描述了如何实现神经网络中的SGD。特别是，最复杂的操作是计算损失函数相对于网络参数的梯度。我们提出了*backpropagation*算法，该算法有效地计算梯度。

20.1 Feedforward Neural Networks

神经网络背后的思想是，许多神经元可以通过通信链路连接起来以执行复杂的计算。通常将神经网络的结构描述为一个图，其节点是神经元，图中的每条（有向）边将某个神经元的输出连接到另一个神经元的输入。我们将限制我们的注意力在正向网络结构上，其中基本图不包含循环。

前馈神经网络由一个有向无环图、$G = (V, E)$，以及边上的权重函数$w : E \to \mathbb{R}$描述。图的节点对应于神经元。每个单个神经元被建模为一个简单的标量函数，$\sigma : \mathbb{R} \to \mathbb{R}$。我们将关注$\sigma$的三个可能函数：符号函数，$\sigma(a) = \text{sign}(a)$，阈值函数，$\sigma(a) = 1_{[a>0]}$，以及Sigmoid函数，$\sigma(a) = 1/(1 + \exp(-a))$，它是阈值函数的平滑近似。我们称$\sigma$为神经元的“激活”函数。图中的每条边将某个神经元的输出链接到另一个神经元的输入。神经元的输入是通过取所有连接到它的神经元的输出的加权总和得到的，其中权重根据w确定。

为了简化网络执行的计算描述，我们进一步假设网络组织在 $layers$ 中。也就是说，节点集可以分解为（非空）不相交子集的并集，$V = \cup_{t=0}^{T} V_t$，使得 E 中的每条边将 V_{t-1} 中的某个节点与 V_t 中的某个节点连接起来，对于某些 $t \in [T]$。底层，V_0，被称为输入层。它包含 $n+$ 个神经元，其中 n 是输入空间的维度。对于每个 $i \in [n]$，V_0 中神经元 i 的输出简单地为 x_i。V_0 中的最后一个神经元是“常数”神经元，它始终输出 1。我们用 $v_{t,i}$ 表示 t 层的第 i 个神经元，用 $o_{t,i}(\mathbf{x})$ 表示当网络输入输入向量 $\mathbf{x}$ 时的 $v_{t,i}$ 的输出。因此，对于 $i \in [n]$ 我们有 $o_{0,i}(\mathbf{x}) = x_i$，对于 $i = n + 1$ 我们有 $o_{0,i}(\mathbf{x}) = 1$。我们现在按层逐层进行计算。假设我们已经计算了第 t 层中神经元的输出。然后，我们可以按照以下方式计算第 $t +$ 层 1 中神经元的输出。固定某个 $v_{t+1,j} \in V_{t+1}$。让 $a_{t+1,j}(\mathbf{x})$ 表示当网络输入输入向量 $\mathbf{x}$ 时 $v_{t+1,j}$ 的输入。然后，

$$a_{t+1,j}(\mathbf{x}) = \sum_{r:\,(v_{t,r},v_{t+1,j})\in E} w((v_{t,r}, v_{t+1,j}))\, o_{t,r}(\mathbf{x}),$$

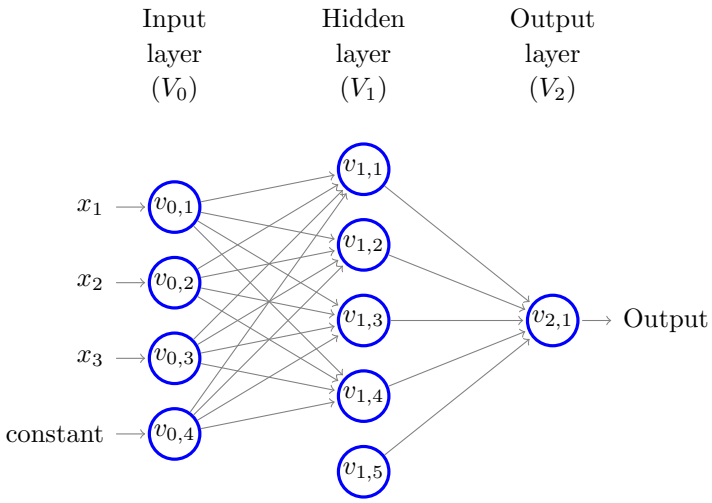
和

$$o_{t+1,j}(\mathbf{x}) = \sigma\left(a_{t+1,j}(\mathbf{x})\right).$$

这是指$v_{t+1,j}$的输入是连接到$v_{t+1,j}$的V_t中神经元的输出加权求和，其中权重根据w确定，而$v_{t+1,j}$的输出仅仅是将其输入应用激活函数σ。

层 $V_1, \ldots, V_{T-1}$ 通常被称为 *hidden layers*。顶层，V_T，被称为输出层。在简单的预测问题中，输出层包含一个神经元，其输出是网络的输出。

我们称 T 为网络中的层数（不包括 V_0），或网络的“深度”。网络的大小为 $|V|$。网络的“宽度”为 $\max_t |V_t|$。以下给出了一个深度为2、大小为10、宽度为5的分层前馈神经网络的示意图。请注意，在隐藏层中有一个没有输入边的神经元。该神经元将输出常数 $\sigma(0)$。



20.2 Learning Neural Networks

一旦我们通过(V, E, σ, w)指定了一个神经网络，我们就得到了一个函数 $h_{V,E,\sigma,w} : \mathbb{R}^{|V_0|-1} \to \mathbb{R}^{|V_T|}$。任何这样的函数集合都可以作为学习的一个假设类。通常，我们通过固定图(V, E)以及激活函数 σ，并让假设类成为某些 $w : E \to \mathbb{R}$ 的所有形式 $h_{V,E,\sigma,w}$ 的函数，来定义一个神经网络预测器的假设类。三元组(V, E, σ)通常被称为网络的 *architecture*。我们用

$$\mathcal{H}_{V,E,\sigma} = \{h_{V,E,\sigma,w} : w \text{ is a mapping from } E \text{ to } \mathbb{R}\}. \tag{20.1}$$

这意味着，指定假设类中假设的参数是网络边上的权重。

我们现在可以研究此类假设类的近似误差、估计误差和优化误差。在第20.3节中，我们通过研究$\mathcal{H}_{V,E,\sigma}$中假设函数能够实现何种类型的函数，从底层图的大小来研究$\mathcal{H}_{V,E,\sigma}$的近似误差。在第20.4节中，我们通过分析其VC维度来研究$\mathcal{H}_{V,E,\sigma}$的估计误差，对于二分类的情况（即$V_T = 1$和σ是符号函数）。最后，在第20.5节中，我们表明即使在底层图很小的情况下，学习类别$\mathcal{H}_{V,E,\sigma}$也是计算上困难的，而在第20.6节中，我们介绍了训练$\mathcal{H}_{V,E,\sigma}$最常用的启发式方法。

20.3 The Expressive Power of Neural Networks

本节我们研究神经网络的表达能力，即可以使用神经网络实现哪些类型的函数。更具体地说，我们将固定某些架构，V, E, σ，并研究$\mathcal{H}_{V,E,\sigma}$中的函数假设能实现哪些函数，这取决于V的大小。

我们首先通过研究哪种布尔函数（即从 $\{\pm 1\}^n$ 到 $\{\pm 1\}$ 的函数）可以通过 $\mathcal{H}_{V,E,\text{sign}}$ 实现。观察到一个计算机中，当使用 b 位存储实数时，每当我们在这样的计算机上计算一个函数 $f : \mathbb{R}^n \to \mathbb{R}$，我们实际上是在计算一个函数 $g : \{\pm 1\}^{nb} \to \{\pm 1\}^b$。因此，研究哪些布尔函数可以通过 $\mathcal{H}_{V,E,\text{sign}}$ 实现，可以告诉我们哪些函数可以在使用 b 位存储实数的计算机上实现。

我们从一条简单的断言开始，表明在不限制网络大小的情况下，每个布尔函数都可以使用深度为2的神经网络实现。

CLAIM 20.1 *For every n, there exists a graph (V, E) of depth 2, such that $\mathcal{H}_{V,E,sign}$ contains all functions from $\{\pm 1\}^n$ to $\{\pm 1\}$.*

Proof 我们构建一个图，包含 $|V_0| = n + 1, |V_1| = 2^n + 1$, 和 $|V_2| = 1$。令 E 为相邻层之间所有可能的边。现在，令 $f : \{\pm 1\}^n \to \{\pm 1\}$ 为某个布尔函数。我们需要证明我们可以调整权重，使得网络实现 f。令 $\mathbf{u}_1, \ldots, \mathbf{u}_k$ 为在 $\{\pm 1\}^n$ 上输出 1 的所有向量。观察发现，对于每一个 i 和每一个 $\mathbf{x} \in \{\pm 1\}^n$，如果 $\mathbf{x} \neq \mathbf{u}_i$ 则 $\langle \mathbf{x}, \mathbf{u}_i \rangle \leq n - 2$，如果 $\mathbf{x} = \mathbf{u}_i$ 则 $\langle \mathbf{x}, u_i \rangle = n$。因此，函数 $g_i(\mathbf{x}) = \text{sign}(\langle \mathbf{x}, \mathbf{u}_i \rangle - n + 1)$ 等于 1 当且仅当 $\mathbf{x} = \mathbf{u}_i$。因此，我们可以调整 V_0 和 V_1 之间的权重，使得对于每一个 $i \in [k]$，神经元 $v_{1,i}$ 实现函数 $g_i(\mathbf{x})$。接下来，我们观察到 $f(\mathbf{x})$ 是以下逻辑或的并集

函数 $g_i(\mathbf{x})$，因此可以写成

$$f(\mathbf{x}) \;=\; \operatorname{sign}\left(\sum_{i=1}^{k} g_i(\mathbf{x}) + k - 1\right),$$

这总结了我们的证明。

□

前述断言表明神经网络可以实现任何布尔函数。然而，这是一个非常弱属性，因为结果网络的规模可能呈指数级增大。在20.1断言证明中给出的构造中，隐藏层的节点数呈指数级增大。这并非我们证明的产物，如下定理所述。

THEOREM 20.2 *For every n, let $s(n)$ be the minimal integer such that there exists a graph (V, E) with $|V| = s(n)$ such that the hypothesis class $\mathcal{H}_{V,E,sign}$ contains all the functions from $\{0,1\}^n$ to $\{0,1$ $\}$. Then, $s(n)$ is exponential in n.*
Similar results hold for $\mathcal{H}_{V,E,\sigma}$ where σ is the sigmoid function.

Proof 假设对于某些(V, E)，我们有$\mathcal{H}_{V,E,\text{sign}}$包含从$\{0,1\}^n$到$\{0,1\}$的所有函数。因此，它可以破坏$m = 2^n$在$\{0,1\}^n$中的向量集，因此$\mathcal{H}_{V,E,\text{sign}}$的VC维数为$2^n$。另一方面，$\mathcal{H}_{V,E,\text{sign}}$的VC维数被$O(|E| \log(|E|)) \leq O(|V|^3)$所限制，正如我们将在下一节中展示的那样。这表明$|V| \geq \Omega(2^{n/3})$，这完成了我们对具有符号激活函数的网络的证明。sigmoid情况的证明是类似的。

□

Remark 20.1 对于任何 σ，只要我们限制权重，使得每个权重都可以用有限个比特数表示，就可以推导出与 $\mathcal{H}_{V,E,\sigma}$ 类似的定理。我们甚至可以考虑这样的假设类，其中不同的神经元可以采用不同的激活函数，只要允许的激活函数数量也是有限的。

哪些函数可以使用多项式大小的网络来表示？前面的论断告诉我们，使用多项式大小的网络表示所有布尔函数是不可能的。从积极的一面来看，在以下内容中，我们表明所有可以在时间 $O(T(n))$ 内计算的布尔函数也可以通过大小为 $O(T(n)^2)$ 的网络来表示。

THEOREM 20.3 *Let $T\colon \mathbb{N} \to \mathbb{N}$ and for every n, let $\mathcal{F}_n$ be the set of functions that can be implemented using a Turing machine using runtime of at most $T(n)$. Then, there exist constants $b, c \in \mathbb{R}_+$ such that for every n, there is a graph (V_n, E_n) of size at most $cT(n)^2 + b$ such that $\mathcal{H}_{V_n,E_n,sign}$ contains $\mathcal{F}_n$.*

这个定理的证明依赖于程序的时间复杂性和它们的电路复杂度之间的关系（例如，参见Sipser（2006））。简而言之，布尔电路是一种网络，其中包含单个神经元

实现它们的合取、析取和否定。电路复杂度衡量计算函数所需的布尔电路的大小。时间复杂度与电路复杂度之间的关系可以直观地如下所示。我们可以将计算机程序执行的每一步建模为其内存状态上的简单操作。因此，网络每层的神经元将反映计算机在相应时间的内存状态，而将转换到网络下一层涉及的是网络可以执行的计算。要将布尔电路与具有符号激活函数的网络相关联，我们需要证明我们可以使用符号激活函数实现合取、析取和否定操作。显然，我们可以使用符号激活函数实现否定运算符。以下引理表明，符号激活函数还可以实现其输入的合取和析取。

LEMMA 20.4 *Suppose that a neuron v, that implements the sign activation function, has k incoming edges, connecting it to neurons whose outputs are in $\{\pm 1\}$. Then, by adding one more edge, linking a "constant" neuron to v, and by adjusting the weights on the edges to v, the output of v can implement the conjunction or the disjunction of its inputs.*

Proof 只需观察，如果 $f : \{\pm 1\}^k \to \{\pm 1\}$ 是合取函数，$f(\mathbf{x}) = \wedge_i x_i$，则它可以写成 $f(\mathbf{x}) =$ 符号 $\left(1 - k + \sum_{i=1}^{k} x_i\right)$。同样，析取函数，$f(\mathbf{x}) = \vee_i x_i$，可以写成 $f(\mathbf{x}) =$ 符号 $\left(k - 1 + \sum_{i=1}^{k} x_i\right)$。

□

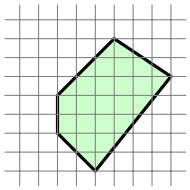
到目前为止，我们讨论了布尔函数。在练习1中，我们表明神经网络是 *universal approximators*。也就是说，对于每个固定的精度参数，$\epsilon > 0$，以及每个Lipschitz函数 $f : [-1, 1]^n \to [-1, 1]$，都存在一个网络，使得对于每个输入 $\mathbf{x} \in [-1, 1]^n$，网络输出一个介于 $f(\mathbf{x}) - \epsilon$ 和 $f(\mathbf{x}) + \epsilon$ 之间的数。然而，正如布尔函数的情况一样，这里的网络大小也不能在 n 中是多项式的。这在下述定理中得到了形式化，其证明是定理20.2的直接推论，留作练习。

THEOREM 20.5 *Fix some $\epsilon \in (0, 1)$. For every n, let $s(n)$ be the minimal integer such that there exists a graph (V, E) with $|V| = s(n)$ such that the hypothesis class $\mathcal{H}_{V,E,\sigma}$, with σ being the sigmoid function, can approximate, to within precision of ϵ, every 1-Lipschitz function $f : [-1, 1]^n \to [-1, 1]$. Then $s(n)$ is exponential in n.*

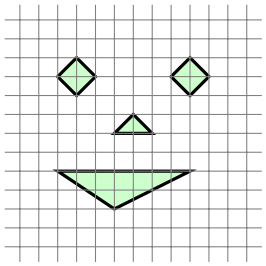
20.3.1 Geometric Intuition

我们接下来提供几个关于函数 $f : \mathbb{R}^2 \to \{\pm 1\}$ 的几何说明，并展示如何使用具有符号激活函数的神经网络来表示它们。

让我们从一个深度为2的网络开始，即一个具有单个隐藏层的网络。隐藏层中的每个神经元实现一个半空间预测器。然后，输出层中的单个神经元在隐藏层神经元的二进制输出上应用一个半空间。正如我们之前所展示的，半空间可以实现合取函数。因此，这样的网络包含所有是$k-1$个半空间交集的假设，其中k是隐藏层中的神经元数量；也就是说，它们可以表示所有具有$k-1$个面的凸多面体。以下给出了5个半空间交集的例子。



我们已经表明，层 V_2 中的一个神经元可以实现一个函数，该函数指示 $\mathbf{x}$ 是否在某个凸多面体中。通过添加一个额外的层，并让输出层中的神经元实现其输入的析取，我们得到一个计算多面体并集的网络。以下给出了这样一个函数的示意图。



20.4 The Sample Complexity of Neural Networks

接下来，我们讨论学习类别 $\mathcal{H}_{V,E,\sigma}$ 的样本复杂度。回忆一下，学习的基本定理告诉我们，学习一个二分类器的假设类别的样本复杂度取决于其VC维度。因此，我们专注于计算形式为 $\mathcal{H}_{V,E,\sigma}$ 的假设类别的VC维度，其中图的网络层包含一个神经元。

我们从符号激活函数开始，即，从 $\mathcal{H}_{V,E,\text{sign}}$ 开始。这个类别的VC维度是多少？直观上，因为我们学习了 $|E|$ 个参数，VC维度应该是 $|E|$ 的阶数。这确实如此，如下定理所形式化。

THEOREM 20.6 *The VC dimension of* $\mathcal{H}_{V,E,sign}$ *is* $O(|E| \log(|E|))$.

Proof 为简化证明中的符号，让我们用 $\mathcal{H}$ 表示假设类。回忆第 6.5.1 节中增长函数 $\tau_{\mathcal{H}}(m)$ 的定义。此函数衡量 $\max_{C\subset\mathcal{X}:|C|=m}|\mathcal{H}_C|$，其中 $\mathcal{H}_C$ 是 $\mathcal{H}$ 对从 C 到 $\{0,1\}$ 的限制。我们可以通过让 $\mathcal{H}_C$ 成为 $\mathcal{H}$ 对从 C 到 $\mathcal{Y}$ 的限制，并保持 $\tau_{\mathcal{H}}(m)$ 的定义不变，自然地扩展从 $\mathcal{X}$ 到某个有限集 $\mathcal{Y}$ 的函数集的定义。

我们的神经网络由一个分层图定义。令 $V_0,\ldots,V_T$ 为图的层。固定一些 $t\in[T$]。通过在 V_{t-1} 和 V_t 之间的边分配不同的权重，我们得到来自 $\mathbb{R}^{|V_{t-1}|}\to\{\pm1$ $\}^{|V_t|}$ 的不同函数。令 $\mathcal{H}^{(t)}$ 为所有可能的此类映射的集合 $\mathbb{R}^{|V_{t-1}|}\to\{\pm1\}^{|V_t|}$。然后，$\mathcal{H}$ 可以写成组合，$\mathcal{H}=\mathcal{H}^{(T)}\circ\ldots\circ\mathcal{H}^{(1)}$。在练习4中，我们表明假设类组合的增长函数被单个类增长函数的乘积所界定。因此，

$$\tau_{\mathcal{H}}(m)\le\prod_{t=1}^{T}\tau_{\mathcal{H}^{(t)}}(m).$$

此外，每个 $\mathcal{H}^{(t)}$ 可以写成函数类 $\mathcal{H}^{(t)}=\mathcal{H}^{(t,1)}\times\cdots\times\mathcal{H}^{(t,|V_t|)}$ 的乘积，其中每个 $\mathcal{H}^{(t,j)}$ 是层 $t-1$ 到 $\{\pm1\}$ 中第 j 个神经元可以实现的全部函数。在练习 3 中我们界定了乘积类，这导致

$$\tau_{\mathcal{H}^{(t)}}(m)\le\prod_{i=1}^{|V_t|}\tau_{\mathcal{H}^{(t,i)}}(m).$$

设 $d_{t,i}$ 为指向层 t 中第 i 个神经元的边的数量。由于该神经元是同质半空间假设，而同质半空间的VC维度是它们输入的维度，根据Sauer引理，我们有

$$\tau_{\mathcal{H}^{(t,i)}}(m)\le\left(\frac{em}{d_{t,i}}\right)^{d_{t,i}}\le(em)^{d_{t,i}}.$$

总体上，我们得到了以下结果

$$\tau_{\mathcal{H}}(m)\le(em)^{\sum_{t,i}d_{t,i}}=(em)^{|E|}.$$

现在，假设有 m 个破碎点。那么，我们必须有 $\tau_{\mathcal{H}}(m)=2^m$，从而我们得到

$$2^m\le(em)^{|E|}\quad\Rightarrow\quad m\le|E|\log(em)/\log(2).$$

该陈述由引理A.2得出。 □

接下来，我们考虑 $\mathcal{H}_{V,E,\sigma}$，其中 σ 是 Sigmoid 函数。令人惊讶的是，$\mathcal{H}_{V,E,\sigma}$ 的 VC 维度被下界为 $\Omega(|E|^2)$（参见练习 5）。也就是说，VC 维度是可调参数数量的平方。也有可能通过 $O(|V|^2\,|E|^2)$ 上界 VC 维度，但证明超出了本书的范围。无论如何，由于在实践中

我们只考虑权重可以用 $O(1)$ 位浮点数短表示的网络，通过使用离散化技巧，我们很容易得到这样的网络具有 $O(|E|)$ 的VC维度，即使我们使用sigmoid激活函数。

20.5 The Runtime of Learning Neural Networks

在前面几节中，我们已经表明，具有多项式大小底层图的神经网络类可以表达所有可以高效实现的函数，并且样本复杂性与网络的大小有利的依赖关系。在本节中，我们转向分析训练神经网络的时复杂度。

首先，我们证明即使在只有4个隐藏层神经元的单隐藏层网络中，根据 $\mathcal{H}_{V,E,\text{sign}}$ 实现ERM规则也是NP困难的。

THEOREM 20.7 *Let $k \geq 3$. For every n, let (V, E) be a layered graph with n input nodes, $k+1$ nodes at the (single) hidden layer, where one of them is the constant neuron, and a single output node. Then, it is NP hard to implement the ERM rule with respect to $\mathcal{H}_{V,E,sign}$.*

证明依赖于从 k-着色问题的约简，并留作练习6。

一种绕过先前硬度结果的方法可能是，为了学习，可能只需要找到一个具有低经验误差的预测器 $h \in \mathcal{H}$，而不一定是精确的 ERM。然而，结果证明，即使找到导致接近最小经验误差的权重的任务在计算上也是不可行的（参见（Bartlett & Ben-David 2002））。

也许人们也会想知道是否有可能改变网络的架构以规避困难结果。也就是说，也许相对于原始网络结构的ERM在计算上是困难的，但相对于某些其他更大的网络，ERM可能可以有效地实现（参见第8章中此类情况的例子）。另一种可能性是使用其他激活函数（例如S形函数或任何其他可以高效计算的激活函数）。有强有力的迹象表明，所有这些方法都注定要失败。事实上，在某个密码学假设下，已知学习半空间交集的问题甚至在表示无关的学习模型中也是困难的（参见Klivans & Sherstov（2006））。这意味着，在相同的密码学假设下，任何包含半空间交集的假设类都不能被有效地学习。

广泛用于训练神经网络的启发式方法依赖于我们在第14章研究过的SGD框架。在那里，我们已证明如果损失函数是凸的，则SGD是一个成功的学习者。在神经网络中，损失函数高度非凸。尽管如此，我们仍然可以实施SGD算法并

希望它能找到一个合理的解决方案（正如在几个实际任务中发生的那样）。

20.6 SGD and Backpropagation

寻找 $\mathcal{H}_{V,E,\sigma}$ 中的低风险假设问题等同于调整边权重的调优问题。在本节中，我们展示了如何使用 SGD 算法应用启发式搜索以获得良好的权重。在本节中，我们假设 σ 是 sigmoid 函数，$\sigma(a) = 1/(1+e^{-a})$，但对于任何可微标量函数，推导都是成立的。

由于 E 是一个有限集，我们可以将权重函数视为一个向量 $\mathbf{w} \in \mathbb{R}^{|E|}$。假设网络有 n 个输入神经元和 k 个输出神经元，用 $h_{\mathbf{w}} : \mathbb{R}^n \to \mathbb{R}^k$ 表示网络在权重函数由 $\mathbf{w}$ 定义时计算出的函数。让我们用 $\Delta(h_{\mathbf{w}}(\mathbf{x}), \mathbf{y})$ 表示预测 $h_{\mathbf{w}}(\mathbf{x})$ 时目标为 $\mathbf{y} \in \mathcal{Y}$ 的损失。为了具体化，我们将 Δ 取为平方损失 $\Delta(h_{\mathbf{w}}(\mathbf{x}), y) = \frac{1}{2}\|h_{\mathbf{w}}(\mathbf{x}) - \mathbf{y}\|^2$；然而，对于每个可微函数都可以得到类似的推导。最后，给定一个在示例域 $\mathbb{R}^n \times \mathbb{R}^k$ 上的分布 $\mathcal{D}$，$L_{\mathcal{D}}(\mathbf{w})$ 表示网络的损失，即，

$$L_{\mathcal{D}}(\mathbf{w}) = \mathop{\mathbb{E}}_{(\mathbf{x},\mathbf{y})\sim\mathcal{D}} \left[\Delta(h_{\mathbf{w}}(\mathbf{x}), \mathbf{y})\right].$$

回忆SGD算法以最小化风险函数{v*}。我们重复第14章中的伪代码，并进行一些修改，这些修改与神经网络应用相关，因为目标函数的非凸性。首先，在第14章中，我们将{v*}初始化为零向量，而在这里，我们将{v*}初始化为接近零的随机向量。这是因为使用零向量的初始化将导致所有隐藏神经元具有相同的权重（如果网络是全层网络）。此外，我们希望如果我们多次重复SGD过程，每次都使用一个新的随机向量初始化过程，那么其中一次运行将导致一个好的局部最小值。其次，虽然固定步长{v*}对于凸问题是有保证的，但在这里我们利用第14.4.2节中定义的可变步长{v*}。由于损失函数的非凸性，序列{v*}的选择更为重要，并且在实践中通过试错的方式进行调整。第三，我们在验证集上输出表现最好的向量。此外，有时在权重上添加正则化（参数{v*}）也是有益的。也就是说，我们尝试最小化{v*}。最后，梯度没有闭式解。相反，它使用反向传播算法实现，将在后续描述。

SGD for Neural Networks

parameters:
number of iterations τ
step size sequence $\eta_1, \eta_2, \ldots, \eta_\tau$
regularization parameter $\lambda > 0$
input:
layered graph (V, E)
differentiable activation function $\sigma : \mathbb{R} \to \mathbb{R}$
initialize:
choose $\mathbf{w}^{(1)} \in \mathbb{R}^{|E|}$ at random
(from a distribution s.t. $\mathbf{w}^{(1)}$ is close enough to $\mathbf{0}$)
for $i = 1, 2, \ldots, \tau$
sample $(\mathbf{x}, \mathbf{y}) \sim \mathcal{D}$
calculate gradient $\mathbf{v}_i = \texttt{backpropagation}(\mathbf{x}, \mathbf{y}, \mathbf{w}, (V, E), \sigma)$
update $\mathbf{w}^{(i+1)} = \mathbf{w}^{(i)} - \eta_i(\mathbf{v}_i + \lambda \mathbf{w}^{(i)})$
output:
$\bar{\mathbf{w}}$ is the best performing $\mathbf{w}^{(i)}$ on a validation set

Backpropagation

input:
example $(\mathbf{x}, \mathbf{y})$, weight vector $\mathbf{w}$, layered graph (V, E),
activation function $\sigma : \mathbb{R} \to \mathbb{R}$
initialize:
denote layers of the graph $V_0, \ldots, V_T$ where $V_t = \{v_{t,1}, \ldots, v_{t,k_t}\}$
define $W_{t,i,j}$ as the weight of $(v_{t,j}, v_{t+1,i})$
(where we set $W_{t,i,j} = 0$ if $(v_{t,j}, v_{t+1,i}) \notin E$)
forward:
set $\mathbf{o}_0 = \mathbf{x}$
for $t = 1, \ldots, T$
for $i = 1, \ldots, k_t$
set $a_{t,i} = \sum_{j=1}^{k_{t-1}} W_{t-1,i,j}\, o_{t-1,j}$
set $o_{t,i} = \sigma(a_{t,i})$
backward:
set $\boldsymbol{\delta}_T = \mathbf{o}_T - \mathbf{y}$
for $t = T-1, T-2, \ldots, 1$
for $i = 1, \ldots, k_t$
$\delta_{t,i} = \sum_{j=1}^{k_{t+1}} W_{t,j,i}\, \delta_{t+1,j}\, \sigma'(a_{t+1,j})$
output:
foreach edge $(v_{t-1,j}, v_{t,i}) \in E$
set the partial derivative to $\delta_{t,i}\, \sigma'(a_{t,i})\, o_{t-1,j}$

Explaining How Backpropagation Calculates the Gradient:
我们接下来解释反向传播算法如何计算损失函数相对于向量 $\mathbf{w}$ 在示例 $(\mathbf{x}, \mathbf{y})$ 上的梯度。让我们首先回忆一些向量微积分的定义。梯度的每个元素是相对于 $\mathbf{w}$ 中与网络的一条边相对应的变量的偏导数。回忆偏导数的定义。给定一个函数 $f : \mathbb{R}^n \to \mathbb{R}$，在 $\mathbf{w}$ 处相对于 ith 变量的偏导数是通过固定 $w_1, \ldots, w_{i-1}, w_{i+1}, w_n$ 的值得到的，这产生了一个标量函数 $g : \mathbb{R} \to \mathbb{R}$，定义为 $g(a) = f((w_1, \ldots, w_{i-1}, w_i + a, w_{i+1}, \ldots, w_n))$，然后对 g 在 0 处求导。对于具有多个输出的函数 $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^m$，$\mathbf{f}$ 在 $\mathbf{w} \in \mathbb{R}^n$ 处的 *Jacobian*，表示为 $J_{\mathbf{w}}(\mathbf{f})$，是一个 $m \times n$ 矩阵，其 i, j 元素是 $f_i : \mathbb{R}^n \to \mathbb{R}$ 相对于其 jth 变量在 $\mathbf{w}$ 处的偏导数。注意，如果 $m = 1$，那么雅可比矩阵是函数的梯度（表示为一个行向量）。以下是我们稍后将使用的两个雅可比矩阵计算的例子。

- 设 $\mathbf{f}(\mathbf{w}) = A\mathbf{w}$ 为 $A \in \mathbb{R}^{m,n}$。然后 $J_{\mathbf{w}}(\mathbf{f}) = A$。
- 对于每个 n，我们使用符号 $\boldsymbol{\sigma}$ 来表示从 $\mathbb{R}^n$ 到 $\mathbb{R}^n$ 的函数，该函数逐元素应用 sigmoid 函数。也就是说，$\boldsymbol{\alpha} = \boldsymbol{\sigma}(\boldsymbol{\theta})$ 表示对于每个 i，我们有 $\alpha_i = \sigma(\theta_i) = \frac{1}{1+\exp(-\theta_i)}$。很容易验证 $J_{\boldsymbol{\theta}}(\boldsymbol{\sigma})$ 是一个对角矩阵，其 (i, i) 项是 $\sigma'(\theta_i)$，其中 σ' 是（标量）sigmoid 函数的导数函数，即 $\sigma'(\theta_i) = \frac{1}{(1+\exp(\theta_i))(1+\exp(-\theta_i))}$。我们还使用符号 $\text{diag}(\boldsymbol{\sigma}'(\boldsymbol{\theta}))$ 来表示这个矩阵。

函数复合求导的 *chain rule* 可以用雅可比矩阵表示如下。给定两个函数 $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^m$ 和 $\mathbf{g} : \mathbb{R}^k \to \mathbb{R}^n$，我们有复合函数的雅可比矩阵，$(\mathbf{f} \circ \mathbf{g}) : \mathbb{R}^k \to \mathbb{R}^m$ 在 $\mathbf{w}$ 处是

$$J_{\mathbf{w}}(\mathbf{f} \circ \mathbf{g}) = J_{g(\mathbf{w})}(\mathbf{f}) J_{\mathbf{w}}(\mathbf{g}).$$

例如，对于 $\mathbf{g}(\mathbf{w}) = A\mathbf{w}$，其中 $A \in \mathbb{R}^{n,k}$，我们有

$$J_{\mathbf{w}}(\boldsymbol{\sigma} \circ \mathbf{g}) = \text{diag}(\boldsymbol{\sigma}'(A\mathbf{w}))\, A.$$

为了描述反向传播算法，让我们首先将 V 分解为图的层，$V = \cup_{t=0}^{T} V_t$。对于每个 t，让我们写出 $V_t = \{v_{t,1}, \ldots, v_{t,k_t}\}$，其中 $k_t = |V_t|$。此外，对于每个 t，表示 $W_t \in \mathbb{R}^{k_{t+1}, k_t}$ 为一个矩阵，它为 V_t 和 V_{t+1} 之间的每个潜在边分配权重。如果边存在于 E 中，则根据 $\mathbf{w}$ 将 $W_{t,i,j}$ 设置为边的权重 $(v_{t,j}, v_{t+1,i})$。否则，我们添加一个“幽灵”边并将其权重设置为零，$W_{t,i,j} = 0$。由于在计算关于某些边权重的偏导数时，我们固定所有其他权重，这些额外的“幽灵”边对现有边的偏导数没有影响。因此，我们可以假设，不失一般性，所有边都存在，即 $E = \cup_t (V_t \times V_{t+1})$。

接下来，我们讨论如何计算从 V_{t-1} 到 V_t 的边缘偏导数，即对 W_{t-1} 中的元素求偏导。由于我们固定了网络中所有其他权重，因此 V_{t-1} 中所有神经元的输出都是固定数字，不依赖于 W_{t-1} 中的权重。用 $\mathbf{o}_{t-1}$ 表示相应的向量。此外，让我们用 $\ell_t : \mathbb{R}^{k_t} \to \mathbb{R}$ 表示由层 $V_t, \dots, V_T$ 定义子网络的损失函数，它是 V_t 中神经元输出的函数。V_t 神经元的输入可以写成 $\mathbf{a}_t = W_{t-1}\mathbf{o}_{t-1}$，$V_t$ 神经元的输出是 $\mathbf{o}_t = \boldsymbol{\sigma}(\mathbf{a}_t)$。也就是说，对于每个 j，我们有 $o_{t,j} = \sigma(a_{t,j})$。我们得到，损失作为 W_{t-1} 的函数可以写成

$$g_t(W_{t-1}) = \ell_t(\mathbf{o}_t) = \ell_t(\boldsymbol{\sigma}(\mathbf{a}_t)) = \ell_t(\boldsymbol{\sigma}(W_{t-1}\mathbf{o}_{t-1})).$$

将此重写如下会更方便。设 $\mathbf{w}_{t-1} \in \mathbb{R}^{k_{t-1}k_t}$ 是通过连接 W_{t-1} 的行然后取所得长向量的转置得到的列向量。定义 O_{t-1} 为 $k_t \times (k_{t-1}k_t)$ 矩阵

$$O_{t-1} \;=\; \begin{pmatrix} \mathbf{o}_{t-1}^\top & 0 & \cdots & 0 \\ 0 & \mathbf{o}_{t-1}^\top & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{o}_{t-1}^\top \end{pmatrix}. \tag{20.2}$$

然后，$W_{t-1}\mathbf{o}_{t-1} = O_{t-1}\mathbf{w}_{t-1}$，因此我们也可以写成

$$g_t(\mathbf{w}_{t-1}) \;=\; \ell_t(\boldsymbol{\sigma}(O_{t-1}\,\mathbf{w}_{t-1})).$$

因此，应用链式法则，我们得到

$$J_{\mathbf{w}_{t-1}}(g_t) \;=\; J_{\boldsymbol{\sigma}(O_{t-1}\mathbf{w}_{t-1})}(\ell_t)\,\mathrm{diag}(\boldsymbol{\sigma}'(O_{t-1}\mathbf{w}_{t-1}))\,O_{t-1}.$$

使用我们的符号，我们有 $\mathbf{o}_t = \boldsymbol{\sigma}(O_{t-1}\mathbf{w}_{t-1})$ 和 $\mathbf{a}_t = O_{t-1}\mathbf{w}_{t-1}$，这得出

$$J_{\mathbf{w}_{t-1}}(g_t) = J_{\mathbf{o}_t}(\ell_t)\,\mathrm{diag}(\boldsymbol{\sigma}'(\mathbf{a}_t))\,O_{t-1}.$$

让我们也记作 $\boldsymbol{\delta}_t = J_{\mathbf{o}_t}(\ell_t)$。然后，我们可以进一步重写前面的内容为

$$J_{\mathbf{w}_{t-1}}(g_t) = \big(\delta_{t,1}\,\sigma'(a_{t,1})\,\mathbf{o}_{t-1}^\top \;,\; \dots \;,\; \delta_{t,k_t}\,\sigma'(a_{t,k_t})\,\mathbf{o}_{t-1}^\top\big)\,. \tag{20.3}$$

它留给计算每个 t 的向量 $\boldsymbol{\delta}_t = J_{\mathbf{o}_t}(\ell_t)$。这是 ℓ_t 在 $\mathbf{o}_t$ 处的梯度。我们以递归的方式计算这个值。首先观察，对于最后一层，我们有 $\ell_T(\mathbf{u}) = \Delta(\mathbf{u}, \mathbf{y})$，其中 Δ 是损失函数。由于我们假设 $\Delta(\mathbf{u}, \mathbf{y}) = \frac{1}{2}\|\mathbf{u}-\mathbf{y}\|^2$，我们得到 $J_{\mathbf{u}}(\ell_T) = (\mathbf{u}-\mathbf{y})$。特别是，$\boldsymbol{\delta}_T = J_{\mathbf{o}_T}(\ell_T) = (\mathbf{o}_T - \mathbf{y})$。接下来，注意

$$\ell_t(\mathbf{u}) = \ell_{t+1}(\boldsymbol{\sigma}(W_t\mathbf{u})).$$

因此，根据链式法则，

$$J_{\mathbf{u}}(\ell_t) = J_{\boldsymbol{\sigma}(W_t\mathbf{u})}(\ell_{t+1})\mathrm{diag}(\boldsymbol{\sigma}'(W_t\mathbf{u}))W_t.$$

特别地，

$$\begin{aligned}\boldsymbol{\delta}_t = J_{\mathbf{o}_t}(\ell_t) &= J_{\boldsymbol{\sigma}(W_t\mathbf{o}_t)}(\ell_{t+1})\mathrm{diag}(\boldsymbol{\sigma}'(W_t\mathbf{o}_t))W_t \\ &= J_{\mathbf{o}_{t+1}}(\ell_{t+1})\mathrm{diag}(\boldsymbol{\sigma}'(\mathbf{a}_{t+1}))W_t \\ &= \boldsymbol{\delta}_{t+1}\,\mathrm{diag}(\boldsymbol{\sigma}'(\mathbf{a}_{t+1}))W_t.\end{aligned}$$

总结来说，我们首先从网络的底部计算向量 $\{\mathbf{a}_t, \mathbf{o}_t\}$ 到其顶部。然后，我们计算从网络顶部到其底部的向量 $\{\boldsymbol{\delta}_t\}$。一旦我们有了所有这些向量，就可以使用方程（20.3）轻松地获得偏导数。因此，我们已经证明了反向传播的伪代码确实计算了梯度。

20.7 Summary

神经网络在大小为 $s(n)$ 的图上可以用来描述所有可以在运行时实现的预测器的假设类别。我们还表明，它们的样本复杂度与 $s(n)$ (多项式相关，具体来说，它取决于网络中的边数)。因此，神经网络假设类别似乎是一个很好的选择。遗憾的是，基于训练数据训练网络的问题在计算上是困难的。我们提出了 SGD 框架作为训练神经网络的启发式方法，并描述了反向传播算法，该算法有效地计算了相对于边的权重损失函数的梯度。

20.8 Bibliographic Remarks

神经网络在20世纪80年代和90年代初被广泛研究，但实证结果参差不齐。近年来，算法的进步、计算能力的提升和数据规模的增加导致了神经网络有效性的突破。特别是，“深度网络”（即超过2层的网络）在各种领域上表现出非常令人印象深刻的实际性能。一些例子包括卷积网络（Lecun & Bengio 1995）、受限玻尔兹曼机（Hinton, Osindero & Teh 2006）、自编码器（Ranzato, Huang, Boureau & Lecun 2007, Bengio & LeCun 2007, Collobert & Weston 2008, Lee, Grosse, Ranganath & Ng 2009, Le, Ranzato, Monga, Devin, Corrado, Chen, Dean & Ng 2012）和求和-积网络（Livni, Shalev-Shwartz & Shamir 2013, Poon & Domingos 2011）。参见（Bengio 2009）及其参考文献。

神经网络的表达能力和与电路复杂性的关系已在（Parberry 1994）中广泛研究。关于神经网络样本复杂性的分析，请参阅（Anthony & Bartlet 1999）。定理20.6的证明技术归功于Kakade和Tewari的讲义。

Klivans & Sherstov (2006) 已经证明，对于任何 $c > 0$，n^c 半空间在 $\{\pm 1\}^n$ 上的交集不是高效的可学习，即使我们允许表示无关的学习。这个困难结果依赖于密码学假设，即不存在多项式时间解的唯一最短向量问题。正如我们所论证的，这意味着即使允许更大的网络或可以高效实现的激活函数，也不可能存在训练神经网络的算法。

反向传播算法已在Rumelhart, Hinton & Williams (1986)中介绍。

20.9 Exercises

1. **Neural Networks are universal approximators:** 设 $f : [-1, 1]^n \to [-1, 1]$ 为一个 ρ-Lipschitz 函数。固定某个 $\epsilon > 0$。构造一个神经网络 $N : [-1, 1]^n \to [-1, 1]$，其激活函数为 sigmoid，使得对于每一个 $\mathbf{x} \in [-1, 1]^n$ 都满足 $|f(\mathbf{x}) - N(\mathbf{x})| \leq \epsilon$。

   Hint: 类似于定理19.3的证明，将 $[-1, 1]^n$ 分割成小盒子。利用 f 的Lipschitz性质，证明它在每个盒子中近似为常数。最后，证明神经网络可以先决定输入向量属于哪个盒子，然后预测该盒子中 f 的平均值。

2. 证明定理20.5。

   Hint: 对于每个 $f : \{-1, 1\}^n \to \{-1, 1\}$ 构造一个 1-Lipschitz 函数 $g : [-1, 1]^n \to [-1, 1]$，使得如果你可以逼近 g，那么你可以表示 f。

3. **Growth function of product:** 对于 $i = 1, 2$，设 $\mathcal{F}_i$ 是从 $\mathcal{X}$ 到 $\mathcal{Y}_i$ 的函数集合。定义 $\mathcal{H} = \mathcal{F}_1 \times \mathcal{F}_2$ 为笛卡尔积类。即对于每一个 $f_1 \in \mathcal{F}_1$ 和 $f_2 \in \mathcal{F}_2$，存在 $h \in \mathcal{H}$ 使得 $h(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}))$。证明 $\tau_{\mathcal{H}}(m) \leq \tau_{\mathcal{F}_1}(m)\, \tau_{\mathcal{F}_2}(m)$。

4. **Growth function of composition:** 设 $\mathcal{F}_1$ 是从 $\mathcal{X}$ 到 Z 的函数集，设 $\mathcal{F}_2$ 是从 Z 到 $\mathcal{Y}$ 的函数集。设 $\mathcal{H} = \mathcal{F}_2 \circ \mathcal{F}_1$ 为复合类。即对于每一个 $f_1 \in \mathcal{F}_1$ 和 $f_2 \in \mathcal{F}_2$，存在 $h \in \mathcal{H}$ 使得 $h(\mathbf{x}) = f_2(f_1(\mathbf{x}))$。证明 $\tau_{\mathcal{H}}(m) \leq \tau_{\mathcal{F}_2}(m)\tau_{\mathcal{F}_1}(m)$。

5. **VC of sigmoidal networks:** 在这个练习中，我们表明存在一个图 (V, E)，使得在这些图上具有sigmoid激活函数的神经网络的VC维是 $\Omega(|E|^2)$。请注意，对于每个 $\epsilon > 0$，sigmoid激活函数可以近似阈值激活函数，$1_{[\sum_i x_i]}$，精度达到 ϵ。为了简化演示，在整个练习中，我们假设我们可以精确地使用sigmoid激活函数实现激活函数 $1_{[\sum_i x_i > 0]}$。

   修复一些 n。

   1. 构建一个网络，N_1，具有 $O(n)$ 权重，该网络实现从 $\mathbb{R}$ 到 $\{0, 1\}^n$ 的函数，并满足以下性质。对于每个 $\mathbf{x} \in \{0, 1\}^n$，

如果我们将实数 $0.x_1x_2\ldots x_n$ 输入网络，那么网络的输出将是 $\mathbf{x}$。*Hint:* 表示 $\alpha = 0.x_1x_2\ldots x_n$ 并观察 $10^k\alpha - 0.5$ 至少是 0.5 如果 $x_k = 1$，并且如果 $x_k = -1$ 则至多是 -0.3。2. 构建一个具有 $O(n)$ 权重的网络 N_2，它将 $[n]$ 映射到 $\{0,1\}^n$，使得 $N_2(i) = \mathbf{e}_i$ 对于所有 i。也就是说，在接收到输入 i 后，网络输出除了第 i 个神经元外的所有零向量。3. 令 $\alpha_1, \ldots, \alpha_n$ 是 n 个实数，使得每个 α_i 都具有形式 $0.a_1^{(i)}a_2^{(i)}\ldots a_n^{(i)}$，其中 $a_j^{(i)} \in \{0,1\}$。构建一个具有 $O(n)$ 权重的网络 N_3，它将 $[n]$ 映射到 $\mathbb{R}$，并满足 $N_2(i) = \alpha_i$ 对于每个 $i \in [n]$。4. 结合 N_1, N_3 以获得一个接收 $i \in [n]$ 并输出 $\mathbf{a}^{(i)}$ 的网络。5. 构建一个接收 $(i,j) \in [n] \times [n]$ 并输出 $a_j^{(i)}$ 的网络 N_4。*Hint:* 观察 AND 函数在 $\{0,1\}^2$ 上可以使用 $O(1)$ 权重计算。6. 结论是存在一个具有 $O(n)$ 权重的图，使得结果假设类的 VC 维度为 n^2。

6. 证明定理20.7。*Hint:*证明类似于学习半步交的难度——参见第8章练习32。

Part III

Additional Learning Models

21 Online Learning

在这一章中，我们描述了一种不同的学习模型，称为*online*学习。在此之前，我们研究了PAC学习模型，其中学习器首先接收一批训练示例，使用训练集学习一个假设，并且只有在学习完成后才使用学到的假设来预测新示例的标签。在我们的 papayas 学习问题中，这意味着我们首先应该买一串 papayas 并尝遍它们。然后，我们使用所有这些信息来学习一个预测规则，该规则确定新 papayas 的味道。相比之下，在在线学习中，训练阶段和预测阶段之间没有分离。相反，每次我们买 papayas 时，它首先被视为一个 *test* 示例，因为我们应该预测它是否会尝起来好。然后，在咬了一口 papayas 之后，我们就知道了真实的标签，同一个 papayas 可以作为一个 *training* 示例，帮助我们改进对未来 papayas 的预测机制。

具体来说，在线学习发生在一系列连续的回合中。在每个在线回合中，学习者首先接收一个实例（学习者买了一个木瓜并知道它的形状和颜色，这构成了实例）。然后，学习者需要预测一个标签（这个木瓜是否美味？）。在回合结束时，学习者获得正确的标签（他尝了木瓜然后知道它是否美味）。最后，学习者使用这些信息来改进他未来的预测。

为了分析在线学习，我们遵循与我们对PAC学习研究相似的路线。我们从在线二分类问题开始。我们考虑可实现的情形，其中我们假设，作为先验知识，所有标签都是由给定假设类中的某个假设生成的，以及不可实现的情形，这对应于无知的PAC学习模型。特别是，我们介绍了一个重要的算法称为 *Weighted-Majority*。接下来，我们研究损失函数是凸的在线学习问题。最后，我们以*Perceptron*算法为例，展示了在在线学习模型中使用代理凸损失函数的应用。

21.1 Online Classification in the Realizable Case

在线学习是在一系列连续的回合中进行的，在第 t 轮中，学习器被给出一个来自实例域 $\mathcal{X}$ 的实例 $\mathbf{x}_t$，并要求提供其标签。我们用 p_t 表示预测的标签。在预测标签后，正确的标签 $y_t \in \{0, 1\}$ 被揭示给学习器。学习器的目标是尽可能少地在这一过程中犯预测错误。学习器试图从之前的回合中推断信息，以提高其在未来回合中的预测。

显然，如果过去和现在轮次之间没有相关性，学习就毫无希望。在本书之前，我们研究了PAC模型，其中我们假设过去和现在的例子是从同一分布源独立同分布采样的。在在线学习模型中，我们对示例序列的起源不做任何统计假设。序列可以是确定性的、随机的，甚至是对学习者的自身行为具有对抗性适应性（如在垃圾邮件过滤的情况下）。自然地，对手可以使我们的在线学习算法的预测错误数量任意大。例如，对手可以在每个在线轮次上呈现相同的实例，等待学习者的预测，并提供与正确标签相反的标签。

为了做出非平凡陈述，我们必须进一步限制问题。可实现性假设是一种可能的自然限制。在可实现的情况下，我们假设所有标签都是由某个假设生成的，$h^\star : \mathcal{X} \to \mathcal{Y}$。此外，$h^\star$来自一个学习者已知的假设类$\mathcal{H}$。这与我们在第3章研究的PAC学习模型类似。在这种对序列的限制下，学习者应该尽可能少犯错误，假设$h^\star$和实例序列可以被一个对手选择。对于一个在线学习算法A，我们用$M_A(\mathcal{H})$表示A在一个由某些$h^\star \in \mathcal{H}$标记的示例序列上可能犯的最大错误数。我们再次强调，$h^\star$和实例序列都可以被一个对手选择。$M_A(\mathcal{H})$的界限称为*mistake-bound*，我们将研究如何设计算法，使得$M_A(\mathcal{H})$最小。形式上：

DEFINITION 21.1（错误界限，在线可学习性）设$\mathcal{H}$为一个假设类，设A为一个在线学习算法。给定任何序列$S = (x_1, h^\star(y_1)), \ldots, (x_T, h^\star(y_T))$，其中$T$为任意整数，$h^\star \in \mathcal{H}$，设$M_A(S)$为$A$在序列$S$上犯的错误数。我们用$M_A(\mathcal{H})$表示上述形式的所有序列上$M_A(S)$的上确界。形式为$M_A(\mathcal{H}) \leq B < \infty$的界限称为*mistake bound*。我们说一个假设类$\mathcal{H}$是可在线学习的，如果存在一个算法A使得$M_A(\mathcal{H}) \leq B < \infty$。

我们的目标是研究在在线模型中哪些假设类是可学习的，特别是要找到给定假设类的好学习算法。

Remark 21.1 在整个本节和下一节中，我们忽略计算-

学习的一个方面，并不限制算法的效率。在第21.3节和第21.4节中，我们研究高效的在线学习算法。

为了简化展示，我们从有限假设类的情况开始，即$|\mathcal{H}| < \infty$。

在PAC学习中，我们识别出ERM是一个好的学习算法，从意义上讲，如果$\mathcal{H}$是可学习的，那么它可以通过$\text{ERM}_{\mathcal{H}}$规则进行学习。对于在线学习，一个自然的规则是在任何在线回合中使用任何ERM假设，即任何与所有过去示例一致的假设。

Consistent

input: A finite hypothesis class $\mathcal{H}$
initialize: $V_1 = \mathcal{H}$
for $t = 1, 2, \ldots$
  receive $\mathbf{x}_t$
  choose any $h \in V_t$
  predict $p_t = h(\mathbf{x}_t)$
  receive true label $y_t = h^\star(\mathbf{x}_t)$
  update $V_{t+1} = \{h \in V_t : h(\mathbf{x}_t) = y_t\}$

`Consistent` 算法维护一个集合 V_t，其中包含所有与 $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_{t-1}, y_{t-1})$ 一致的假设。这个集合通常被称为版本空间。然后，它从 V_t 中选择任何假设并根据这个假设进行预测。

显然，每当 `Consistent` 出现预测错误时，至少从 V_t 中移除一个假设。因此，在犯 M 个错误之后，我们得到 $|V_t| \leq |\mathcal{H}| - M$。由于 V_t 总是非空的（根据可实现性假设，它包含 $h^\star$），我们有 $1 \leq |V_t| \leq |\mathcal{H}| - M$。重新排列，我们得到以下：

COROLLARY 21.2 *Let $\mathcal{H}$ be a finite hypothesis class. The* `Consistent` *algorithm enjoys the mistake bound $M_{\texttt{Consistent}}(\mathcal{H}) \leq |\mathcal{H}| - 1$.*

构建一个假设类和一系列示例，在这些示例中 `Consistent` 确实会犯 $|\mathcal{H}| - 1$ 个错误相当容易（参见练习1）。因此，我们提出一个更好的算法，我们以一种更智能的方式选择 $h \in V_t$。我们将看到这个算法保证犯的错误数量将以指数级减少。

Halving

input: A finite hypothesis class $\mathcal{H}$
initialize: $V_1 = \mathcal{H}$
for $t = 1, 2, \ldots$
  receive $\mathbf{x}_t$
  predict $p_t = \operatorname{argmax}_{r \in \{0,1\}} |\{h \in V_t : h(\mathbf{x}_t) = r\}|$
    (in case of a tie predict $p_t = 1$)
  receive true label $y_t = h^\star(\mathbf{x}_t)$
  update $V_{t+1} = \{h \in V_t : h(\mathbf{x}_t) = y_t\}$

THEOREM 21.3 *Let $\mathcal{H}$ be a finite hypothesis class. The* `Halving` *algorithm enjoys the mistake bound $M_{\texttt{Halving}}(\mathcal{H}) \leq \log_2(|\mathcal{H}|)$.*

Proof 我们只需注意，每当算法出错时，我们就有 $|V_{t+1}| \leq |V_t|/2$，（因此得名“减半”）。因此，如果 M 是错误总数，我们就有

$$1 \leq |V_{T+1}| \leq |\mathcal{H}|\, 2^{-M}.$$

重新排列这个不等式，我们得出我们的证明。 □

当然，`Halving` 的错误界限比 `Consistent` 的错误界限要好得多。我们已看到在线学习与 PAC 学习不同——在 PAC 中，任何 ERM 假设都很好，而在在线学习中，选择任意的 ERM 假设远非最优。

21.1.1 Online Learnability

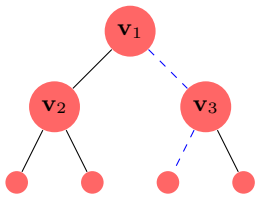
我们接下来采用更通用的方法，旨在描述在线学习能力。特别是，我们针对以下问题：对于给定的假设类 $\mathcal{H}$，最优的在线学习算法是什么？

我们提出一个表征最佳可达到错误界限的假设类维度。这个度量是由Nick Littlestone提出的，因此我们将其称为Ldim($\mathcal{H}$)。

为了激励Ldim的定义，将在线学习过程视为两个玩家之间的游戏是有帮助的：学习者与环境。在游戏的第t轮，环境选择一个实例$\mathbf{x}_t$，学习者预测一个标签$p_t \in \{0,1\}$，最后环境输出真实标签，$y_t \in \{0,1\}$。假设环境想要让学习者在游戏的头T轮犯错误。那么，它必须输出$y_t = 1 - p_t$，唯一的问题是如何选择实例$\mathbf{x}_t$以确保对于某些$h^\star \in \mathcal{H}$，对于所有$t \in [T]$都有$y_t = h^\star(\mathbf{x}_t)$。

一个对抗环境的策略可以形式化地描述为二叉树，如下所示。树的每个节点都与 $\mathcal{X}$ 中的一个实例相关联。最初，环境向学习者展示与树根相关联的实例。然后，如果学习者预测 $p_t = 1$，环境将宣布这是一个错误的预测（即，$y_t = 0$），并将遍历到当前节点的右子节点。如果学习者预测 $p_t = 0$，则环境将设置 $y_t = 1$ 并遍历到左子节点。这个过程将继续，并且在每个回合中，环境将展示与当前节点相关联的实例。

形式上，考虑一个深度为 T (的完全二叉树，我们定义树的深度为从根到叶子的路径中的边数)。这样的树中有 $2^{T+1} - 1$ 个节点，我们为每个节点附加一个实例。令 $\mathbf{v}_1, \ldots, \mathbf{v}_{2^{T+1}-1}$ 为这些实例。我们从树的根开始，并设置 $\mathbf{x}_1 = \mathbf{v}_1$。在第 t 轮，我们设置 $\mathbf{x}_t = \mathbf{v}_{i_t}$，其中 i_t 是当前节点。在结束时



	h_1	h_2	h_3	h_4
$\mathbf{v}_1$	0	0	1	1
$\mathbf{v}_2$	0	1	$*$	$*$
$\mathbf{v}_3$	$*$	$*$	0	1

Figure 21.1 一个深度为2的破碎树的插图。虚线路径对应于示例序列 $((\mathbf{v}_1, 1), (\mathbf{v}_3, 0))$ 。树被 $\mathcal{H} = \{h_1, h_2, h_3, h_4\}$ 破碎，其中 $\mathcal{H}$ 中每个假设在实例 $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$ 上的预测在表中给出（“$*$”标记表示 $h_j(\mathbf{v}_i)$ 可以是1或0）。

round t，我们进入 i_t 的左子节点，如果 $y_t =$ 为 0，或者进入右子节点，如果 $y_t =$ 为 1。也就是说，$i_{t+1} = 2i_t + y_t$。展开递归，我们得到 $i_t = 2^{t-1} + \sum_{j=1}^{t-1} y_j \, 2^{t-1-j}$。环境的前一策略只有在对于每一个$(y_1, \ldots, y_T)$都存在$h \in \mathcal{H}$使得$y_t = h(\mathbf{x}_t)$对所有$t \in [T]$成立时才成功。这导致以下定义。

DEFINITION 21.4 ($\mathcal{H}$ 破碎树) 深度为 d 的破碎树是在 $\mathcal{X}$ 中的一组实例序列 $\mathbf{v}_1, \ldots, \mathbf{v}_{2^d-1}$，对于每个标记 $(y_1, \ldots, y_d) \in \{0, 1\}^d$，存在 $h \in \mathcal{H}$ 使得对于所有 $t \in [d]$，我们有 $h(\mathbf{v}_{i_t}) = y_t$ 其中 $i_t = 2^{t-1} + \sum_{j=1}^{t-1} y_j \, 2^{t-1-j}$。

图21.1给出了深度为2的破碎树的示意图。

DEFINITION 21.5（Littlestone的维度（Ldim））Ldim($\mathcal{H}$)是满足存在一个深度为T的由$\mathcal{H}$破碎的破碎树的最大的整数T。

Ldim 的定义及上述讨论立即暗示以下内容：

LEMMA 21.6 *No algorithm can have a mistake bound strictly smaller than* Ldim($\mathcal{H}$)*; namely, for every algorithm, A, we have $M_A(\mathcal{H}) \geq$* Ldim($\mathcal{H}$)*.*

Proof 让 $T =$ Ldim($\mathcal{H}$)，并让 $\mathbf{v}_1, \ldots, \mathbf{v}_{2^T-1}$ 是满足 Ldim 定义中要求的序列。如果环境设置了 $\mathbf{x}_t = \mathbf{v}_{i_t}$ 和 $y_t = 1 - p_t$ 对于所有 $t \in [T]$，那么学习器会犯 T 个错误，而 Ldim 的定义暗示存在一个假设 $h \in \mathcal{H}$，使得对于所有 t，有 $y_t = h(\mathbf{x}_t)$。

□

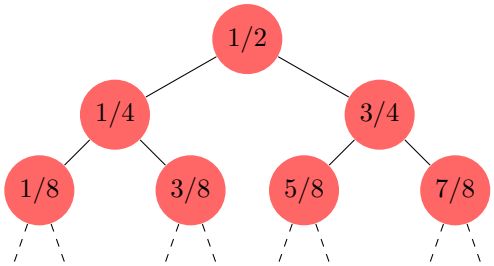
让我们现在给出几个例子。

Example 21.2 设 $\mathcal{H}$ 为一个有限假设类。显然，任何被 $\mathcal{H}$ 破坏的树的最大深度不超过 $\log_2(|\mathcal{H}|)$。因此，Ldim($\mathcal{H}$) $\leqslant \log_2(|\mathcal{H}|)$。另一种得出这个不等式的方法是将引理21.6与定理21.3结合起来。

Example 21.3 让 $\mathcal{X} = \{1, \ldots, d\}$ 和 $\mathcal{H} = \{h_1, \ldots, h_d\}$ 其中 $h_j(x) = 1$ 当且仅当

$x = j$. 然后很容易证明 $\text{Ldim}(\mathcal{H}) = 1$ 当 $|\mathcal{H}| = d$ 可以任意大时。因此，这个例子表明 $\text{Ldim}(\mathcal{H})$ 可以比 $\log_2(|\mathcal{H}|)$ 小得多。

Example 21.4 让 $\mathcal{X} = [0, 1]$ 和 $\mathcal{H} = \{x \mapsto 1_{[x<a]} : a \in [0, 1]\}$；即，$\mathcal{H}$ 是区间 $[0, 1]$ 上的阈值类。那么，$\text{Ldim}(\mathcal{H}) = \infty$。为了看到这一点，考虑以下树



这棵树被 $\mathcal{H}$ 破碎。并且，由于实数的密度，这棵树可以被制作得任意深。

引理21.6表明Ldim({v*})是任何算法错误界限的下界。有趣的是，存在一个标准算法，其错误界限与这个下界相匹配。该算法类似于Halving算法。回想一下，Halving的预测是根据与先前示例一致的假设的多数投票进行的。我们用V_t表示这个集合。换句话说，Halving将V_t分为两个集合：$V_t^+ = \{h \in V_t : h(\mathbf{x}_t) = 1\}$和$V_t^- = \{h \in V_t : h(\mathbf{x}_t) = 0\}$。然后根据两个组中较大的一个进行预测。这种预测背后的理由是，每当Halving犯错误时，它最终会得到$|V_{t+1}| \leq 0.5\,|V_t|$。

以下我们提出的最优算法采用了相同的思想，但不是根据更大的类别进行预测，而是根据具有更大Ldim的类别进行预测。

Standard Optimal Algorithm (SOA)

input: A hypothesis class $\mathcal{H}$
initialize: $V_1 = \mathcal{H}$
for $t = 1, 2, \ldots$
  receive $\mathbf{x}_t$
  for $r \in \{0, 1\}$ let $V_t^{(r)} = \{h \in V_t : h(\mathbf{x}_t) = r\}$
  predict $p_t = \text{argmax}_{r \in \{0,1\}} \text{Ldim}(V_t^{(r)})$
    (in case of a tie predict $p_t = 1$)
  receive true label y_t
  update $V_{t+1} = \{h \in V_t : h(\mathbf{x}_t) = y_t\}$

以下引理形式上确立了先前算法的最优性。

LEMMA 21.7 *SOA enjoys the mistake bound* $M_{SOA}(\mathcal{H}) \leq \text{Ldim}(\mathcal{H})$.

Proof 只需证明，每当算法做出预测错误时，我们有 $\text{Ldim}(V_{t+1}) \leq \text{Ldim}(V_t) - 1$。我们通过假设相反的情况来证明这个命题，即，$\text{Ldim}(V_{t+1}) = \text{Ldim}(V_t)$。如果这成立，那么 p_t 的定义意味着 $\text{Ldim}(V_t^{(r)}) = \text{Ldim}(V_t)$ 对于 $r = 1$ 和 $r = 0$ 都成立。但是，然后我们可以为类别 V_t 构造一个深度为 $\text{Ldim}(V_t) + 1$ 的破碎树，这导致所需的矛盾。

□

结合引理21.7和引理21.6，我们得到：

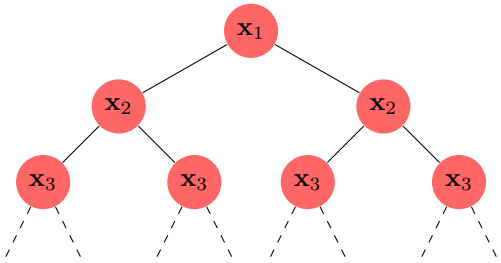
COROLLARY 21.8 *Let* $\mathcal{H}$ *be any hypothesis class. Then, the standard optimal algorithm enjoys the mistake bound* $M_{SOA}(\mathcal{H}) = \text{Ldim}(\mathcal{H})$ *and no other algorithm can have* $M_A(\mathcal{H}) < \text{Ldim}(\mathcal{H})$.

Comparison to VC Dimension

在PAC学习模型中，可学习性由类 {v*} 的VC维度来表征。回忆一下，类 {v*} 的VC维度是满足存在实例 {v*} 被分割的最大的数量 {v*}。也就是说，对于任何标签序列 ({v*}) {v*}0{v*} 1{v*}，存在一个假设 {v*} 能够给出恰好这个标签序列。以下定理将VC维度与Littlestone维度联系起来。

THEOREM 21.9 *For any class* $\mathcal{H}$, $\text{VCdim}(\mathcal{H}) \leq \text{Ldim}(\mathcal{H})$, *and there are classes for which strict inequality holds. Furthermore, the gap can be arbitrarily larger.*

Proof 我们首先证明$\text{VCdim}(\mathcal{H}) \leq \text{Ldim}(\mathcal{H})$。假设$\text{VCdim}(\mathcal{H}) = d$，并令 $\mathbf{x}_1, \ldots, \mathbf{x}_d$为一个被击碎的集合。我们现在构建一个实例的完全二叉树 $\mathbf{v}_1, \ldots, \mathbf{v}_{2^d-1}$，其中深度为$i$的所有节点都设置为$\mathbf{x}_i$ – 请参见以下插图：



现在，破碎集的定义清楚地表明我们得到了一个有效的深度为 d 的破碎树，我们得出结论 $\text{VCdim}(\mathcal{H}) \leq \text{Ldim}(\mathcal{H})$。为了表明差距可以任意大，只需注意，在例21.4中给出的类具有VC维数为1，而其Littlestone维数是无限的。

□

21.2 Online Classification in the Unrealizable Case

在上一节中，我们研究了可实现的在线可学习性。现在我们考虑不可实现的情况。与无偏PAC模型类似，我们不再假设所有标签都是由某个$h^\star \in \mathcal{H}$生成的，而是要求学习器与$\mathcal{H}$中的最佳固定预测器具有竞争力。这由算法的*regret*捕获，它衡量学习者在事后没有遵循某些假设$h \in \mathcal{H}$的预测时的“遗憾”。形式上，当算法A在一系列T示例上运行时相对于h的遗憾被定义为

$$\operatorname{Regret}_A(h, T) = \sup_{(x_1,y_1),\ldots,(x_T,y_T)} \left[\sum_{t=1}^{T} |p_t - y_t| - \sum_{t=1}^{T} |h(x_t) - y_t| \right], \tag{21.1}$$

相对于假设类 $\mathcal{H}$ 的算法的遗憾是

$$\operatorname{Regret}_A(\mathcal{H}, T) = \sup_{h \in \mathcal{H}} \operatorname{Regret}_A(h, T). \tag{21.2}$$

我们重申学习者的目标是在相对于 $\mathcal{H}$ 的最低可能遗憾。一个有趣的问题是，我们是否可以推导出一个具有低遗憾的算法，这意味着遗憾 $_A(\mathcal{H}, T)$ 随着回合数 T 的增加呈次线性增长，这表明学习者的 *error rate* 与 $\mathcal{H}$ 中的最佳假设之间的差异随着 T 趋向无穷大而趋于零。

我们首先表明这是一个不可能的任务——即使 $|\mathcal{H}| = 2$，没有任何算法可以获得亚线性遗憾界限。实际上，考虑 $\mathcal{H} = \{h_0, h_1\}$，其中 h_0 是一个总是返回 0 的函数，而 h_1 是一个总是返回 1 的函数。通过简单地等待学习者的预测并提供与真实标签相反的标签，对手可以使任何在线算法的错误次数等于 T。相比之下，对于任何真实标签序列 $y_1, \ldots, y_T$，令 b 为 $y_1, \ldots, y_T$ 中标签的大多数，那么 h_b 的错误次数最多为 $T/2$。因此，任何在线算法的遗憾可能至少为 $T - T/2 = T/2$，这在 T 上不是亚线性的。这个不可能的结果归因于 Cover（Cover 1965）。

为了绕过Cover的不可能性结果，我们必须进一步限制对抗环境的权力。我们通过允许学习器随机化他的预测来实现这一点。当然，仅凭这一点并不能规避Cover的不可能性结果，因为我们推导这个结果时对学习者的策略没有任何假设。为了使随机化有意义，我们迫使对抗环境在不知道学习者在第t轮投掷的随机硬币的情况下决定y_t。对手仍然可以知道学习者的预测策略，甚至知道之前轮次的随机硬币投掷，但它不知道学习者在第t轮实际使用的随机硬币投掷的值。通过这种（轻微的）游戏规则改变，我们分析了算法的*expected*个错误数量，这里的期望是针对学习者的随机化。也就是说，如果学习者在$\hat{y}_t$处输出$\mathbb{P}[\hat{y}_t = 1] = p_t$，那么他期望支付的成本

在圆 t 上

$$\mathbb{P}[\hat{y}_t \neq y_t] = |p_t - y_t|.$$

另一种说法是，不是让学习者的预测结果在 $\{0, 1\}$ 中，而是允许它们在 $[0, 1]$ 中，并将 $p_t \in [0, 1]$ 解释为在 t 轮次预测标签 1 的概率。

基于这个假设，可以推导出一个低遗憾算法。特别是，我们将证明以下定理。

THEOREM 21.10 *For every hypothesis class $\mathcal{H}$, there exists an algorithm for online classification, whose predictions come from* $[0, 1]$*, that enjoys the regret bound*

$$\forall h \in \mathcal{H},\ \sum_{t=1}^{T} |p_t - y_t| - \sum_{t=1}^{T} |h(\mathbf{x}_t) - y_t| \leq \sqrt{2\ \min\{\log(|\mathcal{H}|)\,,\, \mathrm{Ldim}(\mathcal{H}) \log(eT)\}\ T}.$$

Furthermore, no algorithm can achieve an expected regret bound smaller than $\Omega\left\{\mathrm{v}^*\right\} \mathrm{Ldim}(\mathcal{H})\ T\Big)$.

我们将提供前述定理上界部分的构造性证明。下界部分的证明可以在（Ben-David, Pal, & Shalev-Shwartz 2009）中找到。

证明定理21.10依赖于学习专家建议的*Weighted-Majority*算法。该算法本身很重要，我们将在下一小节中专门介绍它。

21.2.1 Weighted-Majority

加权多数是一种解决*prediction with expert ad- vice*问题的算法。在这个在线学习问题中，在第t轮，学习者必须选择d专家的建议。我们还允许学习者通过定义一个关于d专家的分布来随机化他的选择，即选择一个向量$\mathbf{w}^{(t)} \in [0, 1]^d$，其中$\sum_i w_i^{(t)} = 1$，并以概率$w_i^{(t)}$选择第$i$位专家。在学习者选择了一位专家后，它会收到一个成本向量，$\mathbf{v}_t \in [0, 1]^d$，其中$v_{t,i}$是遵循第$i$位专家建议的成本。如果学习者的预测是随机的，那么其损失被定义为平均成本，即$\sum_i w_i^{(t)} v_{t,i} = \langle \mathbf{w}^{(t)}, \mathbf{v}_t \rangle$。该算法假设轮数$T$是已知的。在练习4中，我们展示了如何使用*doubling trick*来消除这种依赖。

Weighted-Majority

input: number of experts, d ; number of rounds, T
parameter: $\eta = \sqrt{2\log(d)/T}$
initialize: $\tilde{\mathbf{w}}^{(1)} = (1, \ldots, 1)$
for $t = 1, 2, \ldots$
  set $\mathbf{w}^{(t)} = \tilde{\mathbf{w}}^{(t)}/Z_t$ where $Z_t = \sum_i \tilde{w}_i^{(t)}$
  choose expert i at random according to $\mathbb{P}[i] = w_i^{(t)}$
  receive costs of all experts $\mathbf{v}_t \in [0,1]^d$
  pay cost $\langle \mathbf{w}^{(t)}, \mathbf{v}_t \rangle$
  update rule $\forall i,\ \tilde{w}_i^{(t+1)} = \tilde{w}_i^{(t)} e^{-\eta v_{t,i}}$

以下定理对于分析加权多数的遗憾界限至关重要。

THEOREM 21.11 *Assuming that $T > 2\log(d)$, the* ***Weighted-Majority*** *algorithm enjoys the bound*

$$\sum_{t=1}^{T} \langle \mathbf{w}^{(t)}, \mathbf{v}_t \rangle - \min_{i \in [d]} \sum_{t=1}^{T} v_{t,i} \ \leq \ \sqrt{2\log(d)\, T}.$$

Proof 我们有:

$$\log \frac{Z_{t+1}}{Z_t} = \log \sum_i \frac{\tilde{w}_i^{(t)}}{Z_t} e^{-\eta v_{t,i}} = \log \sum_i w_i^{(t)} e^{-\eta v_{t,i}}.$$

使用不等式 $e^{-a} \leq 1 - a + a^2/2$，该不等式对所有 $a \in (0,1)$ 成立，以及 $\sum_i w_i^{(t)} = 1$ 的实际情况，我们得到

$$\begin{aligned} \log \frac{Z_{t+1}}{Z_t} &\leq \log \sum_i w_i^{(t)} \left(1 - \eta v_{t,i} + \eta^2 v_{t,i}^2/2\right) \\ &= \log\Big(1 - \underbrace{\sum_i w_i^{(t)} \left(\eta v_{t,i} - \eta^2 v_{t,i}^2/2\right)}_{\stackrel{\text{def}}{=} b}\Big). \end{aligned}$$

接下来，注意 $b \in (0,1)$。因此，对不等式 $1 - b \leq e^{-b}$ 的两边取对数，我们得到不等式 $\log(1-b) \leq -b$，它对所有 $b \leq 1$ 成立，并得到

$$\begin{aligned} \log \frac{Z_{t+1}}{Z_t} &\leq -\sum_i w_i^{(t)} \left(\eta v_{t,i} - \eta^2 v_{t,i}^2/2\right) \\ &= -\eta \left\langle \mathbf{w}^{(t)}, \mathbf{v}_t \right\rangle + \eta^2 \sum_i w_i^{(t)} v_{t,i}^2/2 \\ &\leq -\eta \left\langle \mathbf{w}^{(t)}, \mathbf{v}_t \right\rangle + \eta^2/2. \end{aligned}$$

这个不等式在 t 上求和，我们得到

$$\log(Z_{T+1}) - \log(Z_1) = \sum_{t=1}^{T} \log \frac{Z_{t+1}}{Z_t} \leq -\eta \sum_{t=1}^{T} \langle \mathbf{w}^{(t)}, \mathbf{v}_t \rangle + \frac{T\eta^2}{2}. \tag{21.3}$$

接下来，我们降低 Z_{T+1} 的下界。对于每个 i，我们可以重写 $\tilde{w}_i^{(T+1)} = e^{-\eta \sum_t v_{t,i}}$，我们得到

$$\log Z_{T+1} = \log \left(\sum_i e^{-\eta \sum_t v_{t,i}} \right) \geq \log \left(\max_i e^{-\eta \sum_t v_{t,i}} \right) = -\eta \min_i \sum_t v_{t,i}.$$

将前面的内容与方程（21.3）结合，并使用事实 $\log(Z_1) = \log(d)$，我们得到：

$$-\eta \min_i \sum_t v_{t,i} - \log(d) \;\leq\; -\eta \sum_{t=1}^{T} \langle \mathbf{w}^{(t)}, \mathbf{v}_t \rangle + \frac{T\eta^2}{2},$$

可以重新排列如下：

$$\sum_{t=1}^{T} \langle \mathbf{w}^{(t)}, \mathbf{v}_t \rangle - \min_i \sum_t v_{t,i} \;\leq\; \frac{\log(d)}{\eta} + \frac{\eta T}{2}.$$

将 η 的值代入方程式，我们得出了证明。 □

Proof of Theorem 21.10

配备了 `Weighted-Majority` 算法和定理 21.11，我们准备证明定理 21.10。我们从更简单的情况开始，其中 $\mathcal{H}$ 是一个有限类，并记为 $\mathcal{H} = \{h_1, \ldots, h_d\}$。在这种情况下，我们可以将每个假设 h_i 视为一个专家，其建议预测 $h_i(\mathbf{x}_t)$，其成本为 $v_{t,i} = |h_i(\mathbf{x}_t) - y_t|$。因此，算法的预测将是 $p_t = \sum_i w_i^{(t)} h_i(\mathbf{x}_t) \in [0, 1]$，损失是

$$|p_t - y_t| = \left| \sum_{i=1}^{d} w_i^{(t)} h_i(\mathbf{x}_t) - y_t \right| = \left| \sum_{i=1}^{d} w_i^{(t)} (h_i(\mathbf{x}_t) - y_t) \right|.$$

现在，如果 $y_t = 1$，那么对于所有 i，$h_i(\mathbf{x}_t) - y_t \leq 0$。因此，上述等于 $\sum_i w_i^{(t)} |h_i(\mathbf{x}_t) - y_t|$。如果 $y_t = 0$，那么对于所有 i，$h_i(\mathbf{x}_t) - y_t \geq 0$，上述也等于 $\sum_i w_i^{(t)} |h_i(\mathbf{x}_t) - y_t|$。总的来说，我们已经证明了

$$|p_t - y_t| = \sum_{i=1}^{d} w_i^{(t)} |h_i(\mathbf{x}_t) - y_t| = \langle \mathbf{w}^{(t)}, \mathbf{v}_t \rangle.$$

此外，对于每个 i，$\sum_t v_{t,i}$ 是假设 h_i 所犯错误的数量。应用定理21.11，我们得到

COROLLARY 21.12 *Let $\mathcal{H}$ be a finite hypothesis class. There exists an algorithm for online classification, whose predictions come from* $[0, 1]$ *, that enjoys the regret bound*

$$\sum_{t=1}^{T} |p_t - y_t| - \min_{h \in \mathcal{H}} \sum_{t=1}^{T} |h(\mathbf{x}_t) - y_t| \;\leq\; \sqrt{2 \log(|\mathcal{H}|)\, T}.$$

接下来，我们考虑一般假设类的情况。之前，我们为每个个体假设构建了一个专家。然而，如果 $\mathcal{H}$ 是无限的，这会导致一个空集界限。主要思想是以更复杂的方式构建一组专家。挑战在于如何定义一组专家，一方面，它不是过于庞大，另一方面，它包含能够给出准确预测的专家。

我们构造专家集，使得对于每个假设 $h \in \mathcal{H}$ 和每个实例序列 $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T$，集合中至少存在一个专家在这些实例上表现得与 h 完全相同。对于每个 $L \leq$ Ldim($\mathcal{H}$) 和每个序列 $1 \leq i_1 < i_2 < \cdots < i_L \leq T$，我们定义一个专家。该专家模拟在实例序列 $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T$ 上 SOA（在上一节中介绍）和环境之间的游戏，假设 SOA 在 $i_1, i_2, \ldots, i_L$ 轮次中恰好犯错误。专家由以下算法定义。

Expert$(i_1, i_2, \ldots, i_L)$

input 一个假设类 $\mathcal{H}$; 索引 $i_1 < i_2 < \cdots < i_L$ **initialize** ： $V_1 = \mathcal{H}$ $1, 2, \ldots, T$ 接收 $\mathbf{x}_t$ 对于 $r \in \{0, 1\}$ 令

for $t =$ $V_t^{(r)} = \{h \in V_t : h(\mathbf{x}_t) = r\}$ 定义 $\tilde{y}_t = \text{argmax}_r$ Ldim $\left(V_t^{(r)}\right)$ 在出现平局时 $\tilde{y}_t = 0$) 预测 $\hat{y}_t = 1 - \tilde{y}_t$ 预测 $\hat{y}_t = \tilde{y}_t$

(

更新 $V_{t+1} = V_t^{(\hat{y}_t)}$ **else**

请注意，每位此类专家可以在每一轮 t 都为我们提供预测，同时只观察实例 $\mathbf{x}_1, \ldots, \mathbf{x}_t$。我们的通用在线学习算法现在是将这些专家应用于 `Weighted-Majority` 算法的一个应用。

要分析该算法，我们首先注意到专家的数量是

$$d = \sum_{L=0}^{\text{Ldim}(\mathcal{H})} \binom{T}{L}. \tag{21.4}$$

可以证明，当 $T \geq$ Ldim($\mathcal{H}$) $+ 2$ 时，方程的右侧被 $(eT/\text{Ldim}(\mathcal{H}))^{\text{Ldim}(\mathcal{H})}$ (所界定。证明可以在引理A.5中找到)。

定理21.11告诉我们，Weighted-Majority的期望错误数至多为最佳专家的错误数加上$\sqrt{\text{^2 log}(d)\,T}$。接下来我们将证明最佳专家的错误数至多为$\mathcal{H}$中最佳假设的错误数。以下关键引理表明，对于任何实例序列，对于每个假设$h \in \mathcal{H}$，都存在具有相同行为的专家。

LEMMA 21.13 *Let $\mathcal{H}$ be any hypothesis class with* Ldim($\mathcal{H}$) $< \infty$. *Let $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T$ be any sequence of instances. For any $h \in \mathcal{H}$, there exists $L \leq$* Ldim($\mathcal{H}$) *and indices $1 \leq i_1 < i_2 < \cdots < i_L \leq T$ such that when running* 专家($i_1, i_2, \ldots, i_L$) *on the sequence $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T$, the expert predicts $h(\mathbf{x}_t)$ on each online round $t = 1, 2, \ldots, T$.*

Proof 修复 $h \in \mathcal{H}$ 和序列 $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T$。我们必须构建 L 和索引 $i_1, i_2, \ldots, i_L$。考虑在输入 $(\mathbf{x}_1, h(\mathbf{x}_1))$、$(\mathbf{x}_2, h(\mathbf{x}_2))$、…、$(\mathbf{x}_T, h(\mathbf{x}_T))$ 上运行 SOA。SOA 在此类输入上最多犯 Ldim($\mathcal{H}$) 个错误。我们定义 L 为 SOA 所犯错误数，并定义 $\{i_1, i_2, \ldots, i_L\}$ 为 SOA 犯错误的那轮集合。

现在，考虑在序列 $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T$ 上运行的专家($i_1, i_2, \ldots, i_L$)。根据构造，Expert($i_1, i_2, \ldots, i_L$) 维护的集合 V_t 等于 SOA 在序列 $(\mathbf{x}_1, h(\mathbf{x}_1)), \ldots, (\mathbf{x}_T, h(\mathbf{x}_T))$ 上运行时维护的集合 V_t。只有当轮次在 $\{i_1, i_2, \ldots, i_L\}$ 中时，SOA 的预测才与 h 的预测不同。由于 Expert($i_1, i_2, \ldots, i_L$) 在 t 不在 $\{i_1, i_2, \ldots, i_L\}$ 中时预测与 SOA 完全相同，而在 t 在 $\{i_1, i_2, \ldots, i_L\}$ 中时预测与 SOA 的预测相反，我们得出结论，专家的预测始终与 h 的预测相同。

□

前一个引理特别适用于在$\mathcal{H}$中的假设，该假设在示例序列上犯的错误最少，因此我们得到以下结果：

COROLLARY 21.14 *Let $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_T, y_T)$ be a sequence of examples and let $\mathcal{H}$ be a hypothesis class with* Ldim($\mathcal{H}$) $< \infty$. *There exists $L \leq$* Ldim($\mathcal{H}$) *and indices $1 \leq i_1 < i_2 < \cdots < i_L \leq T$, such that* 专家($i_1, i_2, \ldots, i_L$) *makes at most as many mistakes as the best $h \in \mathcal{H}$ does, namely,*

$$\min_{h \in \mathcal{H}} \sum_{t=1}^{T} |h(\mathbf{x}_t) - y_t|$$

mistakes on the sequence of examples.

与定理21.11一起，定理21.10的上界部分得到了证明。

21.3 Online Convex Optimization

在第12章中，我们研究了凸学习问题，并在无监督PAC学习框架中展示了这些问题的可学习性结果。在本节中，我们表明类似的可学习性结果也适用于在线学习框架中的凸问题。特别是，我们考虑以下问题。

Online Convex Optimization

definitions:
hypothesis class $\mathcal{H}$; domain Z ; loss function $\ell : \mathcal{H} \times Z \to \mathbb{R}$
assumptions:
$\mathcal{H}$ is convex
$\forall z \in Z$, $\ell(\cdot, z)$ is a convex function
for $t = 1, 2, \ldots, T$
learner predicts a vector $\mathbf{w}^{(t)} \in \mathcal{H}$
environment responds with $z_t \in Z$
learner suffers loss $\ell(\mathbf{w}^{(t)}, z_t)$

与在线分类问题一样，我们分析了算法的 *regret*。回想一下，关于一个竞争假设的在线算法的遗憾，这里将是一个向量 $\mathbf{w}^\star \in \mathcal{H}$，定义为

$$\text{Regret}_A(\mathbf{w}^\star, T) = \sum_{t=1}^{T} \ell(\mathbf{w}^{(t)}, z_t) - \sum_{t=1}^{T} \ell(\mathbf{w}^\star, z_t). \tag{21.5}$$

与之前一样，相对于一组竞争向量 $\mathcal{H}$，算法的遗憾定义为

$$\text{Regret}_A(\mathcal{H}, T) = \sup_{\mathbf{w}^\star \in \mathcal{H}} \text{Regret}_A(\mathbf{w}^\star, T).$$

第14章中，我们已证明随机梯度下降在无监督PAC模型中解决了凸学习问题。现在我们展示一个非常相似的算法，在线梯度下降，解决了在线凸学习问题。

Online Gradient Descent

parameter: $\eta > 0$
initialize: $\mathbf{w}^{(1)} = \mathbf{0}$
for $t = 1, 2, \ldots, T$
predict $\mathbf{w}^{(t)}$
receive z_t and let $f_t(\cdot) = \ell(\cdot, z_t)$
choose $\mathbf{v}_t \in \partial f_t(\mathbf{w}^{(t)})$
update:
1. $\mathbf{w}^{(t+\frac{1}{2})} = \mathbf{w}^{(t)} - \eta \mathbf{v}_t$
2. $\mathbf{w}^{(t+1)} = \text{argmin}_{\mathbf{w} \in \mathcal{H}} \|\mathbf{w} - \mathbf{w}^{(t+\frac{1}{2})}\|$

THEOREM 21.15 *The Online Gradient Descent algorithm enjoys the following regret bound for every* $\mathbf{w}^\star \in \mathcal{H}$,

$$\mathrm{Regret}_A(\mathbf{w}^\star, T) \leq \frac{\|\mathbf{w}^\star\|^2}{2\eta} + \frac{\eta}{2}\sum_{t=1}^{T}\|\mathbf{v}_t\|^2.$$

If we further assume that f_t *is* ρ*-Lipschitz for all* t*, then setting* $\eta = 1/\sqrt{T}$ *yields*

$$\mathrm{Regret}_A(\mathbf{w}^\star, T) \leq \frac{1}{2}(\|\mathbf{w}^\star\|^2 + \rho^2)\sqrt{T}.$$

If we further assume that $\mathcal{H}$ *is* B*-bounded and we set* $\eta = \frac{B}{\rho\sqrt{T}}$ *then*

$$\mathrm{Regret}_A(\mathcal{H}, T) \leq B\,\rho\,\sqrt{T}.$$

Proof 分析类似于具有投影的随机梯度下降分析。使用投影引理、$\mathbf{w}^{(t+\frac{1}{2})}$的定义以及子梯度定义，我们有对于每一个$t$,

$$\begin{aligned}
&\|\mathbf{w}^{(t+1)} - \mathbf{w}^\star\|^2 - \|\mathbf{w}^{(t)} - \mathbf{w}^\star\|^2 \\
&\quad = \|\mathbf{w}^{(t+1)} - \mathbf{w}^\star\|^2 - \|\mathbf{w}^{(t+\frac{1}{2})} - \mathbf{w}^\star\|^2 + \|\mathbf{w}^{(t+\frac{1}{2})} - \mathbf{w}^\star\|^2 - \|\mathbf{w}^{(t)} - \mathbf{w}^\star\|^2 \\
&\quad \leq \|\mathbf{w}^{(t+\frac{1}{2})} - \mathbf{w}^\star\|^2 - \|\mathbf{w}^{(t)} - \mathbf{w}^\star\|^2 \\
&\quad = \|\mathbf{w}^{(t)} - \eta\mathbf{v}_t - \mathbf{w}^\star\|^2 - \|\mathbf{w}^{(t)} - \mathbf{w}^\star\|^2 \\
&\quad = -2\eta\langle \mathbf{w}^{(t)} - \mathbf{w}^\star, \mathbf{v}_t\rangle + \eta^2\|\mathbf{v}_t\|^2 \\
&\quad \leq -2\eta(f_t(\mathbf{w}^{(t)}) - f_t(\mathbf{w}^\star)) + \eta^2\|\mathbf{v}_t\|^2.
\end{aligned}$$

对 t 求和，并观察到等式左边是一个望远镜求和，我们得到

$$\|\mathbf{w}^{(T+1)} - \mathbf{w}^\star\|^2 - \|\mathbf{w}^{(1)} - \mathbf{w}^\star\|^2 \leq -2\eta\sum_{t=1}^{T}(f_t(\mathbf{w}^{(t)}) - f_t(\mathbf{w}^\star)) + \eta^2\sum_{t=1}^{T}\|\mathbf{v}_t\|^2.$$

重新排列不等式并利用事实 $\mathbf{w}^{(1)} = \mathbf{0}$，我们得到

$$\begin{aligned}
\sum_{t=1}^{T}(f_t(\mathbf{w}^{(t)}) - f_t(\mathbf{w}^\star)) &\leq \frac{\|\mathbf{w}^{(1)} - \mathbf{w}^\star\|^2 - \|\mathbf{w}^{(T+1)} - \mathbf{w}^\star\|^2}{2\eta} + \frac{\eta}{2}\sum_{t=1}^{T}\|\mathbf{v}_t\|^2 \\
&\leq \frac{\|\mathbf{w}^\star\|^2}{2\eta} + \frac{\eta}{2}\sum_{t=1}^{T}\|\mathbf{v}_t\|^2.
\end{aligned}$$

这证明了定理中的第一个界限。第二个界限源于假设 f_t 是 ρ-Lipschitz，这意味着 $\|\mathbf{v}_t\| \leq \rho$。 □

21.4 The Online Perceptron Algorithm

感知机是一种经典的在线学习算法，用于二分类，其假设类为同质半空间，即 $\mathcal{H} = \{\mathbf{x} \mapsto \mathrm{sign}(\langle \mathbf{w}, \mathbf{x}\rangle)$:

$\mathbf{w} \in \mathbb{R}^d\}$在9.1.2节中，我们介绍了感知器的批量版本，旨在解决与$\mathcal{H}$相关的ERM问题。现在，我们介绍感知器算法的在线版本。

让 $\mathcal{X} = \mathbb{R}^d$，$\mathcal{Y} = \{-1, 1\}$。在第 t 轮中，学习者接收一个向量 $\mathbf{x}_t \in \mathbb{R}^d$。学习者维护一个权重向量 $\mathbf{w}^{(t)} \in \mathbb{R}^d$ 并预测 $p_t = \text{sign}(\langle \mathbf{w}^{(t)}, \mathbf{x}_t \rangle)$。然后，它接收 $y_t \in \mathcal{Y}$ 并在 $p_t \neq y_t$ 为 1 时支付 1，否则为 0。

学习者的目标是尽可能减少预测错误。在第21.1节中，我们描述了最优算法，并表明最佳可达到的错误界限取决于类的小石子维度。我们后来表明，如果 $d \geq 2$，则 Ldim($\mathcal{H}$) $= \infty$，这意味着我们无法期望做出很少的预测错误。确实，考虑树，其中 $\mathbf{v}_1 = (\frac{1}{2}, 1, 0, \ldots, 0)$，$\mathbf{v}_2 = (\frac{1}{4}, 1, 0, \ldots, 0)$，$\mathbf{v}_3 = (\frac{3}{4}, 1, 0, \ldots, 0$) 等。由于实数的密度，此树被包含所有由 $\mathbf{w}$ 参数化的形式为 $\mathbf{w} = (-1, a, 0$ $, \ldots, 0)$ 的假设的 $\mathcal{H}$ 子集所打碎，其中 $a \in [0, 1]$。我们得出结论，确实 Ldim($\mathcal{H}$) $= \infty$。

为了规避这一不可能性结果，感知机算法依赖于*surrogate convex losses* (（参见第12.3节）的技术。这也与我们在第15章研究过的*margin*概念密切相关。

一个权重向量 $\mathbf{w}$ 在一个例子 $(\mathbf{x}, y)$ 上犯错误，当且仅当 $\langle \mathbf{w}, \mathbf{x} \rangle$ 的符号不等于 y。因此，我们可以将 0−1 损失函数写成如下形式

$$\ell(\mathbf{w}, (\mathbf{x}, y)) = \mathbb{1}_{[y\langle \mathbf{w}, \mathbf{x} \rangle \leq 0]}.$$

在算法预测错误的回合中，我们将使用铰链损失作为代理凸损失函数

$$f_t(\mathbf{w}) = \max\{0, 1 - y_t \langle \mathbf{w}, \mathbf{x}_t \rangle\}.$$

铰链损失满足以下两个条件：

- f_t 这是一个凸函数
- 对于所有 $\mathbf{w}$，$f_t(\mathbf{w}) \geq \ell(\mathbf{w}, (\mathbf{x}_t, y_t))$。特别是，这适用于 $\mathbf{w}^{(t)}$。

在算法正确的回合中，我们将定义 $f_t(\mathbf{w}) =$ 为 0。显然，在这种情况下 f_t 也是凸的。此外，$f_t(\mathbf{w}^{(t)}) = \ell(\mathbf{w}^{(t)}, (\mathbf{x}_t, y_t)) =$ 为 0。

Remark 21.5 在12.3节中，我们为所有示例使用了相同的代理损失函数。在在线模型中，我们允许代理依赖于特定轮次。它甚至可以依赖于$\mathbf{w}^{(t)}$。我们能够使用特定轮次的代理的能力源于我们在在线学习中采用的worst-case类型分析。

让我们现在在函数序列 `Online Gradient Descent` 上运行 $f_1, \ldots, f_T$ 算法，假设类为 $\mathbb{R}^d$ (中的所有向量，因此投影步骤是空的)。回忆一下，该算法初始化 $\mathbf{w}^{(1)} = \mathbf{0}$，其更新规则是

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta \mathbf{v}_t$$

对于某些 $\mathbf{v}_t \in \partial f_t(\mathbf{w}^{(t)})$。在我们的情况下，如果 $y_t \langle \mathbf{w}^{(t)}, \mathbf{x}_t \rangle >$ 为 0，则 f_t 是零

函数，我们可以取 $\mathbf{v}_t = \mathbf{0}$。否则，容易验证 $\mathbf{v}_t = -y_t\mathbf{x}_t$ 在 $\partial f_t(\mathbf{w}^{(t)})$ 中。因此，我们得到更新规则

$$\mathbf{w}^{(t+1)} = \begin{cases} \mathbf{w}^{(t)} & \text{if } y_t\langle \mathbf{w}^{(t)}, \mathbf{x}_t\rangle > 0 \\ \mathbf{w}^{(t)} + \eta y_t \mathbf{x}_t & \text{otherwise} \end{cases}$$

用 $\mathcal{M}$ 表示包含 $\text{sign}(\langle \mathbf{w}^{(t)}, \mathbf{x}_t\rangle) \neq y_t$ 的轮次集合。注意，在第 t 轮，感知机的预测可以重写为

$$p_t = \text{sign}(\langle \mathbf{w}^{(t)}, \mathbf{x}_t\rangle) \;=\; \text{sign}\left(\eta \sum_{i\in\mathcal{M}:i<t} y_i \langle \mathbf{x}_i, \mathbf{x}_t\rangle\right) .$$

这个形式意味着，只要 $\eta >$ 不为 0，感知器算法的预测和集合 $\mathcal{M}$ 都不依赖于 η 的实际值。因此，我们得到了感知器算法：

Perceptron

initialize: $\mathbf{w}1 = \mathbf{0}$ $1, 2, \ldots, T$
接收 t $\mathbf{x}_t$ 预测 $p_t =$ 签名(
$\langle \mathbf{w}^{(t)}, \mathbf{x}_t\rangle$) 如果 $y_t\langle \mathbf{w}^{(t)}, \mathbf{x}_t\rangle \leq$
0 $\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + y_t\mathbf{x}_t$ 否则
$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)}$

为了分析感知机，我们依赖于上一节给出的在线梯度下降分析。在我们的情况下，感知机中使用的 f_t 的子梯度是 $\mathbf{v}_t = -1_{[y_t\langle \mathbf{w}^{(t)}, \mathbf{x}_t\rangle \leq 0]}\, y_t\, \mathbf{x}_t$。确实，感知机的更新是 $\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \mathbf{v}_t$，正如之前讨论的那样，这等价于对于每个 $\eta > 0$ 的 $\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta\mathbf{v}_t$。因此，定理21.15告诉我们

$$\sum_{t=1}^{T} f_t(\mathbf{w}^{(t)}) - \sum_{t=1}^{T} f_t(\mathbf{w}^\star) \leq \frac{1}{2\eta}\|\mathbf{w}^\star\|_2^2 + \frac{\eta}{2}\sum_{t=1}^{T}\|\mathbf{v}_t\|_2^2.$$

由于 $f_t(\mathbf{w}^{(t)})$ 是 0−1 损失的代理，我们知道 $\sum_{t=1}^{T} f_t(\mathbf{w}^{(t)}) \geq |\mathcal{M}|$。记 $R =$ 为 $\max_t \|\mathbf{x}_t\|$；然后我们得到

$$|\mathcal{M}| - \sum_{t=1}^{T} f_t(\mathbf{w}^\star) \leq \frac{1}{2\eta}\|\mathbf{w}^\star\|_2^2 + \frac{\eta}{2}|\mathcal{M}|\, R^2$$

设置 $\eta = \frac{\|\mathbf{w}^\star\|}{R\sqrt{|\mathcal{M}|}}$ 并重新排列，我们得到

$$|\mathcal{M}| - R\|\mathbf{w}^\star\|\sqrt{|\mathcal{M}|} - \sum_{t=1}^{T} f_t(\mathbf{w}^\star) \leq 0. \tag{21.6}$$

这个不等式意味着

THEOREM 21.16 *Suppose that the Perceptron algorithm runs on a sequence* $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_T, y_T)$ *and let* $R = \max_t \|\mathbf{x}_t\|$. *Let* $\mathcal{M}$ *be the rounds on which the Perceptron errs and let* $f_t(\mathbf{w}) = 1_{[t \in \mathcal{M}]} [1 - y_t \langle \mathbf{w}, \mathbf{x}_t \rangle]_+$. *Then, for every* $\mathbf{w}^\star$

$$|\mathcal{M}| \;\leq\; \sum_t f_t(\mathbf{w}^\star) + R \|\mathbf{w}^\star\| \sqrt{\sum_t f_t(\mathbf{w}^\star)} + R^2 \|\mathbf{w}^\star\|^2 \;.$$

In particular, if there exists $\mathbf{w}^\star$ *such that* $y_t \langle \mathbf{w}^\star, \mathbf{x}_t \rangle \geq 1$ *for all* t *then*

$$|\mathcal{M}| \leq R^2 \|\mathbf{w}^\star\|^2.$$

Proof 定理可从方程（21.6）和以下命题得出：给定 $x, b, c \in \mathbb{R}_+$，不等式 $x - b\sqrt{x} - c \leq 0$ 蕴含 $x \leq c + b^2 + b\sqrt{c}$。最后一个命题可以通过分析凸抛物线 $Q(y) = y^2 - by - c$ 的根轻松得出。

□

定理21.16的最后一个假设称为*separability with large margin*（，参见第15章）。也就是说，存在$\mathbf{w}^\star$不仅满足点$\mathbf{x}_t$位于半空间的正确一侧，还保证$\mathbf{x}_t$不会太靠近决策边界。更具体地说，$\mathbf{x}_t$到决策边界的距离至少为$\gamma = 1/\|\mathbf{w}^\star\|$，并且界限变为$(R/\gamma)^2$。

当可分性假设不成立时，界限涉及项$[1 - y_t \langle \mathbf{w}^\star, \mathbf{x}_t \rangle]_+$，该项衡量了满足边缘要求的可分性被违反的程度。

作为最后的评论，我们注意到存在一些情况，其中某些$\mathbf{w}^\star$在序列上不会犯错误，但感知器会犯很多错误。事实上，这是Ldim$(\mathcal{H}) = \infty$这一事实的直接后果。我们避开这种不可能性结果的方法是假设更多的例子序列——定理21.16中的界限只有在累积代理损失$\sum_t f_t(\mathbf{w}^\star)$不是过大时才有意义。

21.5 Summary

在这一章中，我们研究了在线学习模型。我们为PAC学习模型推导出的许多结果在在线模型中都有类似之处。首先，我们表明一个组合维度，即Littlestone维度，表征了在线可学习性。为了证明这一点，我们引入了SOA算法（对于可实现情况）和加权多数算法（对于不可实现情况）。我们还研究了在线凸优化，并表明当损失函数是凸的和Lipschitz时，在线梯度下降是一个成功的在线学习器。最后，我们介绍了在线感知器算法，将其作为在线梯度下降和代理凸损失函数概念的组合。

21.6 Bibliographic Remarks

标准最优算法是由Littlestone（1988）的开创性工作推导出来的。对不可实现情况的推广，以及其他如基于边界的Littlestone维度等变体，在（Ben-David等人，2009）中得到了推导。在（Abernethy，Bartlett，Rakhlin和Tewari 2008，Rakhlin，Sridharan和Tewari 2010，Daniely等人，2011）中获得了超越分类的在线学习性描述。加权多数算法归功于（Littlestone和Warmuth 1994）以及（Vovk 1990）。

“在线凸规划”这一术语由Zinkevich（2003）提出，但这一设置由Gordon（1999）在几年前就已经引入。感知机可以追溯到Rosenblatt（Rosenblatt 1958）。对于可实现情况（带有边界假设）的分析出现在（Agmon 1954，Minsky & Papert 1969）。Freund和Schapire（Freund & Schapire 1999）提出了一种基于将问题简化为可实现情况的分析，用于不可实现情况，并基于平方铰链损失。Gentile（Gentile 2003）给出了不可实现情况下的直接分析，其中使用了铰链损失。

对于更多信息，我们建议读者参考Cesa-Bianchi & Lugosi（2006）和Shalev-Shwartz（2011）。

21.7 Exercises

1. 找到一个假设类 $\mathcal{H}$ 和一个例子序列，使得 `Consistent` 在其上犯 $|\mathcal{H}| - 1$ 个错误。2. 找到一个假设类 $\mathcal{H}$ 和一个例子序列，使得 $Halving$ 算法的错误界限是紧的。3. 令 $d \geq = 2$，$\mathcal{X} = \{ = 1, \ldots, d\}$，并令 $\mathcal{H} = \{h_j : j \in [d]\}$，其中 $h_j(x) = = 1_{[x=j]}$。计算 $M_{\text{Halving}}(\mathcal{H})$ (，即推导 $M_{\text{Halving}}(\mathcal{H})$ 的上下界，并证明它们是相等的)。

4. **The Doubling Trick:**

在定理21.15中，参数η依赖于时间范围T。在本练习中，我们展示如何通过一个简单的技巧消除这种依赖。

考虑一个具有形式 $\alpha\sqrt{T}$ 的遗憾界限的算法，但其参数需要了解 T 的知识。以下描述的加倍技巧使我们能够将此类算法转换为不需要了解时间范围的算法。想法是将时间划分为逐渐增大的时间段，并在每个时间段上运行原始算法。

The Doubling Trick

input: algorithm A whose parameters depend on the time horizon
for $m = 0, 1, 2, \ldots$
  run A on the 2^m rounds $t = 2^m, \ldots, 2^{m+1} - 1$

证明如果 A 在每个 2^m 轮的每个周期上的遗憾不超过 $\alpha\sqrt{2^m}$，那么总遗憾不超过

$$\frac{\sqrt{2}}{\sqrt{2}-1}\,\alpha\,\sqrt{T}.$$

5. **Online-to-batch Conversions:** 在这个练习中，我们展示了如何使用一个成功的在线学习算法来推导出一个成功的PAC学习器。

考虑一个由实例域 $\mathcal{X}$ 和假设类 $\mathcal{H}$ 参数化的二分类 PAC 学习问题。假设存在一个在线学习算法 A，它具有一个错误界限 $M_A(\mathcal{H}) < \infty$。考虑在从实例空间 $\mathcal{X}$ 上的分布 $\mathcal{D}$ 中独立同分布采样的 T 个示例序列上运行此算法，并且这些示例被某些 $h^\star \in \mathcal{H}$ 标记。假设对于每一轮 t，算法的预测基于一个假设 $h_t\colon\ \mathcal{X} \to \{0,1\}$。证明

$$\mathbb{E}[L_{\mathcal{D}}(h_r)] \leq \frac{M_A(\mathcal{H})}{T},$$

期望是对实例的随机选择以及根据均匀分布选择 r 的随机选择。

Hint: 使用与定理14.8证明中出现的类似论据。

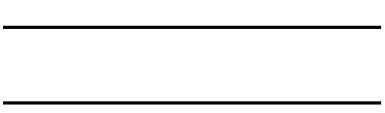
22 Clustering

聚类是探索性数据分析中最广泛使用的技术之一。在所有学科领域，从社会科学到生物学再到计算机科学，人们试图通过识别数据点中的有意义群体来获得对数据的初步直觉。例如，计算生物学家根据基因在不同实验中的表达相似性对基因进行聚类；零售商根据客户档案对客户进行聚类，以实现针对性营销；天文学家根据恒星的空间邻近性对恒星进行聚类。

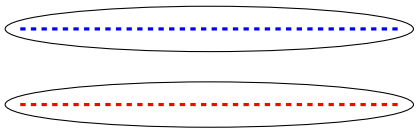
第一个需要明确的问题是，自然地，什么是聚类？直观上，聚类是将一组对象分组的过程，使得相似的对象最终落在同一个组中，而不同的对象被分到不同的组中。显然，这种描述相当不精确，可能存在歧义。相当令人惊讶的是，如何得出一个更严谨的定义并不清楚。

存在几个导致这种困难的原因。一个基本问题是，在早期声明中提到的两个目标在很多情况下可能相互矛盾。从数学的角度讲，相似性（或接近性）不是一个传递关系，而聚类共享是一个等价关系，特别是它是一个传递关系。更具体地说，可能存在一个长序列的对象，$x_1, \ldots, x_m$，其中每个x_i与其两个相邻对象x_{i-1}和x_{i+1}非常相似，但x_1和x_m非常不相似。如果我们希望确保每当两个元素相似时它们共享同一个聚类，那么我们必须将序列中的所有元素放入同一个聚类中。然而，在这种情况下，我们最终会得到不相似元素（x_1和x_m）共享一个聚类，从而违反第二个要求。

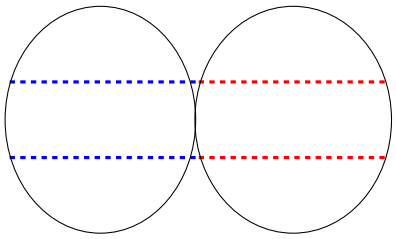
为了进一步说明这一点，假设我们想要将以下图片中的点聚集成两个簇。



一个强调不分离邻近点的聚类算法（例如，将在第22.1节中描述的单链接算法）将根据这两条线水平地分离输入数据来聚类：

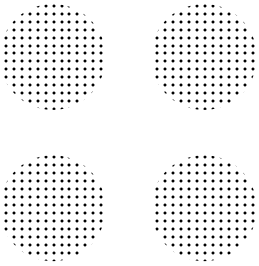


与强调没有远离的点共享同一簇的聚类方法（例如，将在第22.1节中描述的2-means算法）将通过将其垂直分为右手半部和左手半部来对相同的输入进行聚类：

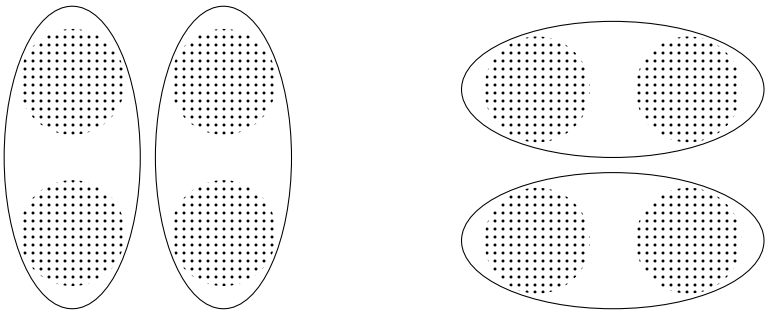


另一个基本问题是聚类缺乏“真实情况”，这是*unsupervised learning*中常见的问题。到目前为止，在本书中，我们主要处理*supervised*学习（例如，从标记的训练数据中学习分类器的问题）。监督学习的目标是明确的——我们希望学习一个分类器，它能尽可能准确地预测未来示例的标签。此外，监督学习器可以使用标记的训练数据通过计算经验损失来估计其假设的成功或风险。相比之下，聚类是一个*unsupervised learning*问题；也就是说，没有我们试图预测的标签。相反，我们希望以某种有意义的方式组织数据。因此，没有明确的聚类成功评估程序。事实上，即使在完全了解潜在数据分布的基础上，也不清楚对于该数据的“正确”聚类是什么，以及如何评估一个提出的聚类。

例如，考虑以下在 $\mathbb{R}^2$ 中的点集：



并且假设我们需要将它们聚为两个簇。我们有两个高度合理的解决方案：



这种现象不仅人工，而且在实际应用中发生。一组对象可以以各种不同的有意义的方式进行聚类。这可能是由于对象之间具有不同的隐含距离（或相似性）概念，例如，根据说话人的口音对语音录音进行聚类，而不是根据内容聚类；根据电影主题对电影评论进行聚类，而不是根据评论情感聚类；根据主题对画作进行聚类，而不是根据风格聚类，等等。

总结来说，对于给定的数据集，可能会有几种非常不同的可想象聚类解决方案。因此，存在多种聚类算法，在某些输入数据上，它们将输出非常不同的聚类。

A Clustering Model:
聚类任务在输入类型和预期计算的结果类型方面都可能有所不同。为了具体说明，我们将关注以下常见设置：

Input — 一组元素集，$\mathcal{X}$，以及其上的距离函数。即，一个对称的函数 d : $\mathcal{X} \times \mathcal{X} \to \mathbb{R}_+$，对所有 $x \in \mathcal{X}$ 满足 $d(x, x) = 0$，并且通常也满足三角不等式。或者，该函数可以是相似度函数 $s : \mathcal{X} \times \mathcal{X} \to [0, 1]$，它是对称的，并且对所有 $x \in \mathcal{X}$ 满足 $s(x, x) = 1$。此外，一些聚类算法还需要一个输入参数 k (，用于确定所需聚类的数量)。

Output — 域集合 $\mathcal{X}$ 的划分。也就是说，$C = (C_1, \dots C_k)$ 其中 $\bigcup_{i=1}^{k} C_i = \mathcal{X}$ 并且对于所有 $i \neq j$，$C_i \cap C_j = \emptyset$。在某些情况下，聚类是“软”的，即 $\mathcal{X}$ 划分为不同的聚类的概率，输出是一个函数，将向量 $(p_1(x), \dots, p_k(x))$ 分配给每个域点 $x \in \mathcal{X}$，其中 $p_i(x) = \mathbb{P}[\mathbf{x} \in C_i]$ 是 $\mathbf{x}$ 属于聚类 C_i 的概率。另一种可能的输出是从希腊语 dendron = 树，gramma = 绘制) 的聚类 *dendrogram* (，它是一个域子集的层次树，其叶子是单元素集合，其根是整个域。我们将在以下内容中更详细地讨论这种公式。

在以下内容中，我们概述了一些最流行的聚类方法。在本章的最后部分，我们回到对聚类是什么的高层次讨论。

22.1 Linkage-Based Clustering Algorithms

链接聚类可能是最简单、最直接的聚类范式。这些算法按顺序进行。它们从每个数据点作为一个单点聚类的平凡聚类开始。然后，这些算法反复合并前一个聚类中“最接近”的聚类。因此，每次这样的循环后，聚类的数量都会减少。如果继续进行，这些算法最终会导致所有领域点共享一个大型聚类的平凡聚类。因此，需要确定两个参数来明确定义此类算法。首先，我们必须决定如何衡量（或定义）聚类之间的距离，其次，我们必须确定何时停止合并。回想一下，聚类算法的输入是点间距离函数{v*}。有许多方法可以将{v*}扩展为领域子集（或聚类）之间的距离度量。最常见的方法是

1. 单链接聚类，其中簇间距离定义为两个簇成员之间的最小距离，即，

$$D(A,B) \stackrel{\text{def}}{=} \min\{d(x,y) : x \in A,\ y \in B\}$$

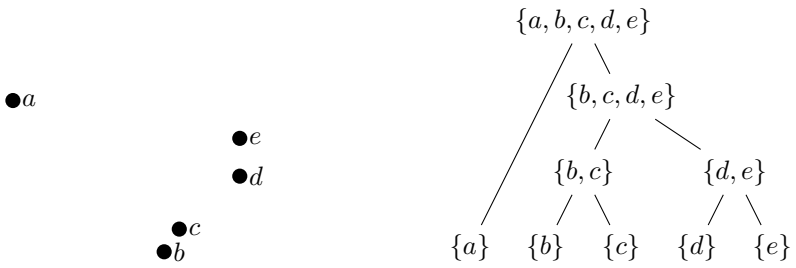
2. 平均链接聚类，其中两个簇之间的距离定义为其中一个簇中的一个点到另一个簇中的一个点的平均距离，即，

$$D(A,B) \stackrel{\text{def}}{=} \frac{1}{|A||B|} \sum_{x \in A,\ y \in B} d(x,y)$$

3. 最大链接聚类，其中两个簇之间的距离定义为它们元素之间的最大距离，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在意义上是 *agglomerative*，因为它们从完全碎片化的数据开始，随着过程的进行，不断构建更大和更大的簇。如果不采用停止规则，此类算法的结果可以描述为聚类 *dendrogram*：即一个域子集的树，其叶子节点是单元素集，根节点是整个域。例如，如果输入是左侧所示具有欧几里得距离的 $\mathcal{X} = \{a, b, c, d, e\} \subset \mathbb{R}^2$ 元素，那么得到的树状图是右侧所示的一个：



单链接算法与在加权图上寻找最小生成树的Kruskal算法密切相关。确实，考虑一个完整图，其顶点是$\mathcal{X}$的元素，边的权重（x, y）是距离（$d(x, y)$）。单链接算法执行的两个聚类的每次合并对应于上述图中的一条边的选择。还可以证明，单链接算法在其运行过程中选择的边的集合构成一个最小生成树。

如果要将树状图转换为空间划分（聚类），则需要采用一个 *stopping criterion*。常见的停止标准包括

- 固定数量的聚类 - 调整某些参数，k，一旦聚类数量达到 k 就停止合并聚类。
- 距离上界 - 固定一些 $r \in \mathbb{R}_+$。一旦所有簇间距离都大于 r，就停止合并。我们还可以将 r 设置为 $\alpha \max\{d(x, y) : x, y \in \mathcal{X}\}$ 对于某些 $\alpha < 1$。在这种情况下，停止标准被称为“缩放距离上界”。

22.2 k-Means and Other Cost Minimization Clusterings

另一种流行的聚类方法从定义一组可能的聚类参数集上的成本函数开始，聚类算法的目标是找到最小成本的分区（聚类）。在这种范式下，聚类任务被转化为一个优化问题。目标函数是从输入对（$\mathcal{X}, d$）和提出的聚类解决方案（$C = (C_1, \ldots, C_k)$）到正实数的函数。给定这样的目标函数，我们称之为G，聚类算法的*goal*被定义为对于给定的输入（$\mathcal{X}, d$），找到一个聚类（C）以使$G((\mathcal{X}, d), C)$最小化。为了达到这个目标，必须应用一些适当的搜索算法。

结果发现，大多数优化问题都是NP难的，有些甚至难以近似。因此，当人们谈论，比如说，k-均值聚类时，他们通常指的是某些特定的常见近似算法，而不是成本函数或最小化问题的对应精确解。

许多常见的目标函数需要聚类数量，k，作为一个

参数。在实际应用中，通常由聚类算法的用户选择最适合给定聚类问题的参数 k。

在以下内容中，我们描述了一些最常见的目标函数。

- **The k-means objective function** 是聚类目标中最受欢迎的之一。在 k-means 中，数据被划分为不相交的集合 $C_1, \ldots, C_k$，其中每个 C_i 都由一个质心 μ_i 表示。假设输入集 $\mathcal{X}$ 嵌入某个更大的度量空间 $(\mathcal{X}', d)$（因此 $\mathcal{X} \subseteq \mathcal{X}'$）并且质心是 $\mathcal{X}'$ 的成员。k-means 目标函数衡量每个点在 $\mathcal{X}$ 中到其簇质心的平方距离。C_i 的质心定义为

$$\mu_i(C_i) = \operatorname*{argmin}_{\mu \in \mathcal{X}'} \sum_{x \in C_i} d(x, \mu)^2.$$

然后，k-均值目标函数是

$$G_{\mathrm{k-means}}((\mathcal{X}, d), (C_1, \ldots, C_k)) = \sum_{i=1}^{k} \sum_{x \in C_i} d(x, \mu_i(C_i))^2.$$

这也可以写成

$$G_{\mathrm{k-means}}((\mathcal{X}, d), (C_1, \ldots, C_k)) = \min_{\mu_1, \ldots \mu_k \in \mathcal{X}'} \sum_{i=1}^{k} \sum_{x \in C_i} d(x, \mu_i)^2. \quad (22.1)$$

k-均方目标函数在数字通信任务中相关，例如，$\mathcal{X}$的成员可以被视为需要传输的信号集合。当$\mathcal{X}$可能是一个非常大的实值向量集合时，数字传输只允许为每个信号传输有限数量的比特。在这样约束下实现良好传输的一种方法是将$\mathcal{X}$的每个成员表示为某个有限集合$\mu_1, \ldots \mu_k$中的“接近”成员，并用传输最接近的μ_i的索引来代替传输任何$x \in \mathcal{X}$。k-均方目标可以被视为这种传输表示方案产生的失真的度量。

- **The k-medoids objective function** 与 k-means 目标相似，但要求聚类中心是输入集的成员。目标函数由以下定义：

$$G_{\mathrm{K-medoid}}((\mathcal{X}, d), (C_1, \ldots, C_k)) = \min_{\mu_1, \ldots \mu_k \in \mathcal{X}} \sum_{i=1}^{k} \sum_{x \in C_i} d(x, \mu_i)^2.$$

- **The k-median objective function** 与 k-聚类中心目标相当相似，只是数据点与其聚类中心之间的“扭曲”是通过距离而不是距离的平方来衡量的：

$$G_{\mathrm{K-median}}((\mathcal{X}, d), (C_1, \ldots, C_k)) = \min_{\mu_1, \ldots \mu_k \in \mathcal{X}} \sum_{i=1}^{k} \sum_{x \in C_i} d(x, \mu_i).$$

一个这样的目标有意义的例子是 *facility location* 问题。考虑在某个城市中定位 k 个消防站的任务。可以将房屋建模为数据点，并旨在放置消防站以最小化房屋与其最近的消防站之间的平均距离。

前一个例子都可以被视为 *center-based* 目标。此类聚类问题的解决方案由一系列聚类中心确定，聚类将每个实例分配到最近的中心。更普遍地，基于中心的目标准备通过选择某个单调函数 $f\colon \mathbb{R}_+ \to \mathbb{R}_+$，然后定义

$$G_f((\mathcal{X},d),(C_1,\ldots C_k)) = \min_{\mu_1,\ldots\mu_k\in\mathcal{X}'} \sum_{i=1}^{k}\sum_{x\in C_i} f(d(x,\mu_i)),$$

$\mathcal{X}'$ 是 $\mathcal{X}$ 或 $\mathcal{X}$ 的某个超集。

一些目标函数不是基于中心的。例如，*sum of in- cluster distances (SOD)*

$$G_{\mathrm{SOD}}((\mathcal{X},d),(C_1,\ldots C_k)) = \sum_{i=1}^{k}\sum_{x,y\in C_i} d(x,y)$$

并且我们在第22.3节中将要讨论的MinCut目标不是基于中心的。

22.2.1 The k-Means Algorithm

k-均方目标函数在聚类的实际应用中相当流行。然而，发现最优的k-均方解通常在计算上不可行（该问题是NP难的，甚至难以在某个常数范围内近似）。作为替代，以下简单的迭代算法经常被使用，如此频繁以至于在许多情况下，术语k-均方聚类指的是该算法的结果，而不是最小化k-均方目标成本的聚类。我们根据欧几里得距离函数$d(\mathbf{x},\mathbf{y}) = \|\mathbf{x}-\mathbf{y}\|$描述该算法。

k-Means

input: $\mathcal{X} \subset \mathbb{R}^n$; Number of clusters k
initialize: Randomly choose initial centroids $\boldsymbol{\mu}_1,\ldots,\boldsymbol{\mu}_k$
repeat until convergence
  $\forall i \in [k]$ set $C_i = \{\mathbf{x}\in\mathcal{X} : i = \operatorname{argmin}_j \|\mathbf{x}-\boldsymbol{\mu}_j\|\}$
  (break ties in some arbitrary manner)
  $\forall i \in [k]$ update $\boldsymbol{\mu}_i = \frac{1}{|C_i|}\sum_{\mathbf{x}\in C_i}\mathbf{x}$

LEMMA 22.1 *Each iteration of the k-means algorithm does not increase the k-means objective function (as given in Equation (22.1)).*

Proof 为了简化符号，让我们使用缩写 $G(C_1, \ldots, C_k)$ 来表示 k-均值目标，即
，

$$G(C_1, \ldots, C_k) = \min_{\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k \in \mathbb{R}^n} \sum_{i=1}^{k} \sum_{\mathbf{x} \in C_i} \|\mathbf{x} - \boldsymbol{\mu}_i\|^2. \tag{22.2}$$

它方便地定义 $\boldsymbol{\mu}(C_i) = \frac{1}{|C_i|} \sum_{\mathbf{x} \in C_i} \mathbf{x}$ 并注意 $\boldsymbol{\mu}(C_i) =$ 是 $_{\boldsymbol{\mu} \in \mathbb{R}^n} \sum_{\mathbf{x} \in C_i} \|\mathbf{x} -$ 的 argmin$\boldsymbol{\mu}\|^2$。因此
，我们可以将 k-means 目标函数重写为

$$G(C_1, \ldots, C_k) = \sum_{i=1}^{k} \sum_{\mathbf{x} \in C_i} \|\mathbf{x} - \boldsymbol{\mu}(C_i)\|^2. \tag{22.3}$$

考虑 t 迭代的 k-均值算法的更新。设 $C_1^{(t-1)}, \ldots, C_k^{(t-1)}$ 为前一个划分，设
$\boldsymbol{\mu}_i^{(t-1)} = \boldsymbol{\mu}(C_i^{(t-1)})$，设 $C_1^{(t)}, \ldots, C_k^{(t)}$ 为在迭代 t 时分配的新划分。使用第 22.2
节中给出的目标函数定义，我们可以清楚地看到:

$$G(C_1^{(t)}, \ldots, C_k^{(t)}) \leq \sum_{i=1}^{k} \sum_{\mathbf{x} \in C_i^{(t)}} \|\mathbf{x} - \boldsymbol{\mu}_i^{(t-1)}\|^2. \tag{22.4}$$

此外，新分区（$C_1^{(t)}, \ldots, C_k^{(t)}$）的定义意味着它在所有可能的分区（$C_1, \ldots, C_k$
）中最小化表达式$\sum_{i=1}^{k} \sum_{\mathbf{x} \in C_i} \|\mathbf{x} - \boldsymbol{\mu}_i^{(t-1)}\|^2$。因此，

$$\sum_{i=1}^{k} \sum_{\mathbf{x} \in C_i^{(t)}} \|\mathbf{x} - \boldsymbol{\mu}_i^{(t-1)}\|^2 \leq \sum_{i=1}^{k} \sum_{\mathbf{x} \in C_i^{(t-1)}} \|\mathbf{x} - \boldsymbol{\mu}_i^{(t-1)}\|^2. \tag{22.5}$$

使用方程（22.3），我们得到方程（22.5）的右侧等于 $G(C_1^{(t-1)}, \ldots, C_k^{(t-1)})$。
将此与方程（22.4）和方程（22.5）相结合，我们得到
$G(C_1^{(t)}, \ldots, C_k^{(t)}) \leq G(C_1^{(t-1)}, \ldots, C_k^{(t-1)})$，这完成了我们的证明。

□

虽然前面的引理告诉我们 k-均值目标函数是单调非增加的，但无法保证 k-均值算法达到收敛所需的迭代次数。此外，算法输出的 k-均值目标函数值与该目标函数可能的最小值之间的差距没有非平凡的下界。实际上，k-均值可能收敛到一个甚至不是局部最小值的位置（参见练习2）。为了提高 k-均值的性能，通常建议多次重复使用不同随机选择的初始质心进行程序（例如，我们可以选择初始质心为数据中的随机点）。

22.3 Spectral Clustering

通常，表示数据集 $\mathcal{X} = \{x_1, \ldots, x_m\}$ 中点之间关系的一种方便方法是使用 *similarity graph*；每个顶点代表一个数据点 x_i，并且每两个顶点之间通过一条边连接，其权重是它们的相似度，$W_{i,j} = s(x_i, x_j)$，其中 $W \in \mathbb{R}^{m,m}$。例如，我们可以设置 $W_{i,j} = \exp(-d(x_i, x_j)^2/\sigma^2)$，其中 $d(\cdot,\cdot)$ 是一个距离函数，σ 是一个参数。现在可以将聚类问题表述如下：我们希望找到一个图的划分，使得不同组之间的边权重低，而组内的边权重高。

在先前描述的聚类目标中，重点在于我们对聚类直观定义的一侧——确保同一聚类中的点相似。我们现在提出的目标则关注另一要求——被分离到不同聚类的点应该不相似。

22.3.1 Graph Cut

给定一个由相似性矩阵 W 表示的图，构建图划分的最简单和最直接的方法是解决最小割问题，该问题选择一个划分 $C_1, \ldots, C_k$ 以最小化目标

$$\operatorname{cut}(C_1, \ldots, C_k) = \sum_{i=1}^{k} \sum_{r \in C_i, s \notin C_i} W_{r,s}.$$

对于 $k = 2$，最小割问题可以有效地解决。然而，在实践中，它往往不会导致令人满意的分区。问题在于，在许多情况下，最小割的解决方案只是简单地从一个个体顶点将图的其他部分分离出来。当然，这并不是我们在聚类中想要实现的目标，因为簇应该是合理的大型点组。

一些解决方案已被提出。最简单的解决方案是对切割进行归一化，并如下定义归一化最小割目标：

$$\operatorname{RatioCut}(C_1, \ldots, C_k) = \sum_{i=1}^{k} \frac{1}{|C_i|} \sum_{r \in C_i, s \notin C_i} W_{r,s}.$$

如果簇不是太小，前面的目标函数会假设更小的值。不幸的是，引入这种平衡使得问题在计算上难以解决。谱聚类是一种放松最小化RatioCut问题的方法。

22.3.2 Graph Laplacian and Relaxed Graph Cuts

主数学对象是图拉普拉斯矩阵。文献中关于图拉普拉斯的几种不同定义，以下我们描述其中一个特定定义。

DEFINITION 22.2（未归一化的图拉普拉斯算子）*unnormalized graph Laplacian* 是一个 $m \times m$ 矩阵 $L = D - W$，其中 D 是一个对角矩阵，具有 $D_{i,i} = \sum_{j=1}^{m} W_{i,j}$。矩阵 D 被称为 *degree matrix*。

以下引理强调了RatioCut与拉普拉斯矩阵之间的关系。

LEMMA 22.3 *Let $C_1, \ldots, C_k$ be a clustering and let $H \in \mathbb{R}^{m,k}$ be the matrix such that*

$$H_{i,j} = \frac{1}{\sqrt{|C_j|}} \mathbb{1}_{[i \in C_j]}.$$

Then, the columns of H are orthonormal to each other and

$$RatioCut(C_1, \ldots, C_k) = \text{trace}(H^\top L H).$$

Proof 设 $\mathbf{h}_1, \ldots, \mathbf{h}_k$ 为 H 的列。这些向量是正交归一的事实直接从定义中得出。接下来，通过标准的代数操作，可以证明 $\text{trace}(H^\top L H) = \sum_{i=1}^{k} \mathbf{h}_i^\top L \mathbf{h}_i$，并且对于任何向量 $\mathbf{v}$，我们有

$$\mathbf{v}^\top L \mathbf{v} = \frac{1}{2} \left(\sum_r D_{r,r} v_r^2 - 2 \sum_{r,s} v_r v_s W_{r,s} + \sum_s D_{s,s} v_s^2 \right) = \frac{1}{2} \sum_{r,s} W_{r,s} (v_r - v_s)^2.$$

应用此公式于 $\mathbf{v} = \mathbf{h}_i$ 并注意到 $(h_{i,r} - h_{i,s})^2$ 仅在 $r \in C_i, s \notin C_i$ 或反之情况下不为零，我们得到

$$\mathbf{h}_i^\top L \mathbf{h}_i = \frac{1}{|C_i|} \sum_{r \in C_i, s \notin C_i} W_{r,s}.$$

□

因此，为了最小化RatioCut，我们可以寻找一个矩阵H，其列是正交归一的，并且每个$H_{i,j}$要么是0要么是$1/\sqrt{|C_j|}$。不幸的是，这是一个整数规划问题，我们无法有效地解决。相反，我们放宽了后者的要求，简单地搜索一个正交归一矩阵$H \in \mathbb{R}^{m,k}$，使其迹最小化。正如我们将在下一章关于PCA（特别是定理23.2的证明）中看到的那样，这个问题的解是将U设置为矩阵，其列是与L的最小特征值对应的特征向量。这个算法被称为未归一化谱聚类。

22.3.3 Unnormalized Spectral Clustering

Unnormalized Spectral Clustering

Input: $W \in \mathbb{R}^{m,m}$; Number of clusters k
Initialize: Compute the unnormalized graph Laplacian L
Let $U \in \mathbb{R}^{m,k}$ be the matrix whose columns are the eigenvectors of L corresponding to the k smallest eigenvalues
Let $\mathbf{v}_1, \ldots, \mathbf{v}_m$ be the rows of U
Cluster the points $\mathbf{v}_1, \ldots, \mathbf{v}_m$ using k-means
Output: Clusters $C_1, \ldots, C_K$ of the k-means algorithm

光谱聚类算法从找到对应图拉普拉斯矩阵最小特征值的 k 个特征向量组成的矩阵 H 开始。然后根据 H 的行来表示点。这种表示方式的改变之所以有用，是因为图拉普拉斯矩阵的性质。在许多情况下，这种表示方式的改变使得简单的 k -means 算法能够无缝地检测到簇。直观上，如果 H 如第 22.3 节引理中定义的那样，则在新表示中的每个点都是一个指示向量，其值仅在对应于它所属簇的元素上非零。

22.4 Information Bottleneck*

信息瓶颈方法是Tishby、Pereira和Bialek提出的一种聚类技术。它依赖于来自*information theory*的概念。为了说明该方法，考虑将文本文档聚类的例子，其中每个文档表示为一个词袋；即，每个文档是一个向量$\mathbf{x} = \{0,1\}^n$，其中n是词典的大小，$x_i = 1$ iff与索引i对应的单词出现在文档中。给定一组m文档，我们可以将m文档的词袋表示解释为关于随机变量x的联合概率，表示文档的身份（因此取值在[m]中），以及一个表示词典中单词身份的随机变量y（因此取值在[n]中）。

这个解释中，信息瓶颈指的是将聚类身份视为另一个随机变量，表示为C，它在[k]（其中k将由方法设置）中取值。一旦我们将x, y, C表示为随机变量，我们就可以使用信息论工具来表达聚类目标。特别是，信息瓶颈目标是

$$\min_{p(C|x)} I(x;C) - \beta I(C;y) \ ,$$

在 $I(\cdot;\ \cdot)$ 是两个随机变量之间的互信息，[1] β 是一个

1 即，给定一个概率函数，p 在对 (x, C) 上，

参数，最小化是在所有可能的将点分配给聚类的概率赋值上进行的。直观上，我们希望实现两个相互矛盾的目标。一方面，我们希望文档身份和聚类身份之间的互信息尽可能小。这反映了我们希望对原始数据进行强压缩的事实。另一方面，我们希望聚类变量和单词身份之间的互信息高，这反映了保留文档“相关”信息（如文档中出现的单词所反映的）的目标。这将参数统计中使用的经典最小充分统计量[2]的概念推广到任意分布。

解决与信息瓶颈原理相关的优化问题在一般情况下很困难。一些提出的方法类似于EM原理，我们将在第24章中讨论。

22.5 A High Level View of Clustering

到目前为止，我们主要列出了各种有用的聚类工具。然而，一些基本问题仍未得到解决。首先，什么是聚类？是什么使得一个 *clustering* 算法与任何任意输入空间并输出该空间划分的函数区分开来？是否存在任何与任何特定算法或任务无关的聚类基本属性？

一种解决此类问题的方法是采用公理化方法。已经尝试过提供几种聚类公理化定义。让我们通过展示Kleinberg（2003）所做尝试来演示这种方法。

考虑一个聚类函数 F，它接受任何有限域 $\mathcal{X}$ 作为输入，该域在其对上具有一个相似度函数 d，并返回 $\mathcal{X}$ 的一个划分。

考虑以下该函数的三个性质：

Scale Invariance (SI) 对于任何域集 $\mathcal{X}$，相似度函数 d，以及任何 $\alpha > 0$，以下应成立：$F(\mathcal{X}, d) = F(\mathcal{X}, \alpha d)$（其中 $(\alpha d)(x, y) \stackrel{\text{def}}{=} \alpha\, d(x, y)$）。

Richness (Ri) 对于任何有限 $\mathcal{X}$ 以及 X（的每个非空子集划分 $C = (C_1, \ldots C_k)$，存在某个在 $\mathcal{X}$ 上的相似度函数 d 使得 $F(\mathcal{X}, d) = C$。

$I(x; C) = \sum_a \sum_b p(a, b) \log\left(\frac{p(a,b)}{p(a)p(b)}\right)$，其中求和涵盖所有$x$可以取的值和所有$C$可以取的值。

[2]一个充分统计量是关于数据的一个函数，它具有相对于统计模型及其相关未知参数的充分性性质，这意味着“没有任何其他可以从同一样本计算出的统计量能提供关于参数值的任何额外信息。”例如，如果我们假设一个变量服从具有单位方差和未知期望的正态分布，那么平均函数就是一个充分统计量。

Consistency (Co) 如果 d 和 d' 是 $\mathcal{X}$ 上的相似度函数，使得对于每个 $x, y \in \mathcal{X}$ ，如果 x, y 属于 $F(\mathcal{X}, d)$ 中的同一簇，则 $d'(x, y) \leq d(x, y)$；如果 x, y 属于 $F(\mathcal{X}, d)$ 中的不同簇，则 $d'(x, y) \geq d(x, y)$，那么 $F(\mathcal{X}, d) = F(\mathcal{X}, d')$。

反思片刻可以发现，尺度不变性是一个非常自然的要求——如果聚类函数的结果取决于测量点间距离的单位，那就显得很奇怪。丰富性要求基本上表明，聚类函数的输出完全受函数 d 控制，这也是一个非常直观的特征。第三个要求，一致性，是唯一一个涉及聚类基本（非正式）定义的要求——我们希望相似点被聚在一起，不相似点被分到不同的簇中，因此，如果已经共享一个簇的点变得更加相似，而已经分离的点彼此之间的相似性变得更低，那么聚类函数应该对其之前的聚类决策有更强的“支持”。

然而，克莱因伯格（2003）已经证明了以下“不可能性”结果：

THEOREM 22.4 *There exists no function, F, that satisfies all the three properties: Scale Invariance, Richness, and Consistency.*

Proof 假设，通过反证法，某些 F 满足所有三个性质。选择一个至少有三个点的域集 $\mathcal{X}$。根据丰富性，必须存在某个 d_1 使得 $F(\mathcal{X}, d_1) = \{\{x\} : x \in \mathcal{X}\}$，并且也存在某个 d_2 使得 $F(\mathcal{X}, d_2) \neq F(\mathcal{X}, d_1)$。

设 $\alpha \in \mathbb{R}_+$ 满足对于每一个 $x, y \in \mathcal{X}$，$\alpha d_2(x, y) \geq d_1(x, y)$。设 $d_3 = \alpha d_2$。考虑 $F(\mathcal{X}, d_3)$。根据 F 的尺度不变性属性，我们应该有 $F(\mathcal{X}, d_3) = F(\mathcal{X}, d_2)$。另一方面，由于所有不同的 $x, y \in \mathcal{X}$ 在关于 $F(\mathcal{X}, d_1)$ 和 $d_3(x, y) \geq d_1(x, y)$ 的不同簇中，F 的一致性意味着 $F(\mathcal{X}, d_3) = F(\mathcal{X}, d_1)$。这与我们选择 d_1, d_2 以满足 $F(\mathcal{X}, d_2) \neq F(\mathcal{X}, d_1)$ 的条件相矛盾。

□

重要的是要注意，在这三个属性中没有任何一个“坏属性”。对于这三个公理中的每一对，都存在满足该对两个属性的自然聚类函数（甚至可以通过仅改变单链接聚类函数的停止标准来构造这样的例子）。另一方面，Kleinberg表明，任何最小化任何基于中心的客观函数的聚类算法不可避免地会失败一致性属性（然而，k-聚类内距离和最小化聚类确实满足一致性）。

克莱因伯格不可行性结果可以通过改变属性轻松规避。例如，如果一个人希望讨论具有固定聚类数量参数的聚类函数，那么用 k-Richness 替换 Richness 是自然的（即，每个将域划分为 k 个子集的划分都可通过聚类函数实现）。k-Richness、尺度不变性和一致性都适用于 k-means 聚类，因此它们是一致的。

另一种方法是放宽一致性属性。例如，可以说两个聚类 $C = (C_1, \ldots C_k)$ 和 $C' = (C'_1, \ldots C'_l)$ 是 *compatible*，如果对于每个聚类 $C_i \in C$ 和 $C'_j \in C'$，要么 $C_i \subseteq C'_j$ 要么 $C'_j \subseteq C_i$ 要么 $C_i \cap C'_j = \emptyset$ (值得注意的是，对于每个树状图，通过修剪该树状图获得的每个聚类都是兼容的)。“细化一致性”是在一致性属性假设下，新聚类 $F(\mathcal{X}, d')$ 与旧聚类 $F(\mathcal{X}, d)$ 兼容的要求。许多常见的聚类函数满足这一要求，以及尺度不变性和丰富性。此外，人们可以提出许多其他、不同的聚类函数属性，这些属性听起来直观且令人满意，并且一些常见的聚类函数也满足这些属性。

有许多方式来解释这些结果。我们建议将其视为表明不存在“理想”的聚类函数。每个聚类函数不可避免地会有些“不理想”的性质。因此，为任何给定任务选择聚类函数时，必须考虑该任务的具体性质。没有通用的聚类解决方案，就像没有能够学习每个可学习任务的分类算法（正如无免费午餐定理所示）。聚类，就像分类预测一样，必须考虑关于特定任务的一些先验知识。

22.6 Summary

聚类是一种无监督学习问题，我们希望将一组点划分为“有意义的”子集。我们介绍了包括基于链接的算法、k-means系列、谱聚类和信息瓶颈在内的几种聚类方法。我们讨论了将聚类的直观意义形式化的困难。

22.7 Bibliographic Remarks

k-均算法有时被称为Lloyd算法，以纪念Stuart Lloyd，他在1957年提出了该方法。关于谱聚类的更全面概述，我们建议读者参考Von Luxburg（2007）的优秀教程。信息瓶颈方法由Tishby、Pereira和Bialek（1999）引入。关于公理化方法的进一步讨论，请参阅Ackerman和Ben-David（2008）。

22.8 Exercises

1. **Suboptimality of k-Means:** 对于每个参数 $t > 1$，证明存在一个k-means问题实例，使得k-means算法

(可能找到一个k-means目标至少为$t \cdot$ OPT的解，其中OPT是最小k-means目标。

2. **k-Means Might Not Necessarily Converge to a Local Minimum:** 证明 k-均值算法可能收敛到一个不是局部最小值的位置。*Hint*: 假设 $k = 2$ 且样本点为 $\{1, 2, 3, 4\} \subset \mathbb{R}$，假设我们用中心 $k2\{ 4, 4\}$ 初始化 k-均值；并且假设我们在 C_i 的定义中通过将 i 分配给 $\text{argmin}_j \|\mathbf{x} - \boldsymbol{\mu}_j\|$ 的最小值来打破平局。

3. 给定一个度量空间$(\mathcal{X}, d)$，其中 $|\mathcal{X}| < \infty$，和 $k \in \mathbb{N}$，我们希望找到一个将 $\mathcal{X}$ 划分为 $C_1, \ldots, C_k$ 的划分，使得表达式最小化

$$G_{\text{k-diam}}((\mathcal{X}, d), (C_1, \ldots, C_k)) = \max_{j \in [d]} \text{diam}(C_j),$$

在 $\text{diam}(C_j) = \max_{x,x' \in C_j} d(x, x')$ (处，我们使用约定 $\text{diam}(C_j) = 0$ 如果 $|C_j| < 2$)。

与 k-均值目标类似，最小化 k- *diam* 目标是 NP-hard 的。幸运的是，我们有一个非常简单的近似算法：最初，我们选择一些 $x \in \mathcal{X}$ 并设置 $\mu_1 = x$。然后，算法迭代地设置

$$\forall j \in \{2, \ldots, k\},\ \mu_j = \operatorname*{argmax}_{x \in X} \min_{i \in [j-1]} d(x, \mu_i).$$

最后，我们设

$$\forall i \in [k],\ C_i = \{x \in X : i = \operatorname*{argmin}_{j \in [k]} d(x, \mu_j)\}.$$

证明所描述的算法是一个2-近似算法。也就是说，如果我们用$\hat{C}_1, \ldots, \hat{C}_k$表示它的输出，用$C_1^*, \ldots, C_k^*$表示最优解，那么，

$$G_{\text{k-diam}}((\mathcal{X}, d), (\hat{C}_1, \ldots, \hat{C}_k)) \leq 2 \cdot G_{\text{k-diam}}((\mathcal{X}, d), (C_1^*, \ldots, C_k^*)).$$

Hint: 考虑点 μ_{k+1} (，换句话说，如果我们想要 $k + 1$ 个簇)，我们将会选择的下一个中心，设 $r = \min_{j \in [k]} d(\mu_j, \mu_{k+1})$。证明以下不等式

$$G_{\text{k-diam}}((\mathcal{X}, d), (\hat{C}_1, \ldots, \hat{C}_k)) \leq 2r$$
$$G_{\text{k-diam}}((X, d), (C_1^*, \ldots, C_k^*)) \geq r.$$

4. 回想一下，一个聚类函数 F 被称为基于中心的聚类，如果对于某个单调函数 $f : \mathbb{R}_+ \to \mathbb{R}_+$，在每一个给定的输入 $(\mathcal{X}, d)$ 上，$F(\mathcal{X}, d)$ 是一个最小化目标函数的聚类

$$G_f((\mathcal{X}, d), (C_1, \ldots C_k)) = \min_{\mu_1, \ldots \mu_k \in \mathcal{X}'} \sum_{i=1}^{k} \sum_{x \in C_i} f(d(x, \mu_i)),$$

$\mathcal{X}'$ 是 $\mathcal{X}$ 或 $\mathcal{X}$ 的某个超集。

证明对于每个 $k > 1$，之前练习中定义的 k-diam 聚类函数不是一个基于中心的聚类函数。

Hint: 给定一个聚类输入$(\mathcal{X}, d)$，当$|\mathcal{X}| >$为2时，考虑向$\mathcal{X}$的一些（但不是全部）成员添加许多邻近点对k-diam聚类或任何基于中心的聚类的影响。

5. 回想我们讨论过的三种聚类“属性”：尺度不变性、丰富性和一致性。考虑单链接聚类算法。

1. 找出在固定数量聚类（任何非零固定数）停止规则下，单链接算法满足的三个属性中的哪一个。2. 找出在距离上界（任何非零固定上界）停止规则下，单链接算法满足的三个属性中的哪一个。3. 证明对于这些属性中的任何一对，都存在一个单链接聚类的停止标准，使得这两个公理得到满足。

6. 给定某个数 k，令 k-富集度是以下要求：
For any finite $\mathcal{X}$ and every partition $C = (C_1, \ldots C_k)$ of $\mathcal{X}$ (into nonempty subsets) there exists some dissimilarity function d over $\mathcal{X}$ such that $F(\mathcal{X}, d) = C$. 证明对于每一个数 k，存在一个满足以下三个属性的聚类函数：尺度不变性、k-富集度以及一致性。

23 Dimensionality Reduction

维度降低是将高维空间中的数据映射到新的空间的过程，该空间维度要小得多。这个过程与信息论中（有损）压缩的概念密切相关。降低数据维度有几个原因。首先，高维数据带来计算挑战。此外，在某些情况下，高维可能导致学习算法的泛化能力较差（例如，在最近邻分类器中，样本复杂度随着维度的增加而呈指数增长——见第19章）。最后，维度降低可用于数据的可解释性，用于发现数据的有意义结构，以及用于说明目的。

在这一章中，我们描述了降维的流行方法。在这些方法中，降维是通过将线性变换应用于原始数据来实现的。也就是说，如果原始数据在 $\mathbb{R}^d$ 中，而我们想将其嵌入到 $\mathbb{R}^n$ $(n < d)$ 中，那么我们希望找到一个矩阵 $W \in \mathbb{R}^{n,d}$ 来诱导映射 $\mathbf{x} \mapsto W\mathbf{x}$。选择 W 的一个自然标准是以一种能够合理恢复原始 $\mathbf{x}$ 的方式。不难证明，在一般情况下，从 $\mathbf{x}$ 中精确恢复 $W\mathbf{x}$ 是不可能的（参见练习 1）。

我们描述的第一种方法称为主成分分析（PCA）。在PCA中，压缩和恢复都通过线性变换来完成，该方法找到使恢复向量与原始向量之间的差异在最小二乘意义上最小的线性变换。

接下来，我们描述使用随机矩阵 W 的降维。我们推导出一个重要的引理，通常称为“Johnson-Lindenstrauss 引理”，该引理分析了这种随机降维技术造成的扭曲。

最后，我们展示了如何使用随机矩阵再次降低所有 *sparse* 向量的维度。这个过程被称为压缩感知。在这种情况下，恢复过程是非线性的，但仍然可以通过线性规划有效地实现。

我们通过强调PCA和压缩感知背后的“先验假设”来总结，这有助于我们理解两种方法的优点和缺点。

23.1 Principal Component Analysis (PCA)

设 $\mathbf{x}_1, \ldots, \mathbf{x}_m$ 是 m 个在 $\mathbb{R}^d$ 中的 m 向量。我们希望使用线性变换来降低这些向量的维度。一个矩阵 $W \in \mathbb{R}^{n,d}$，其中 $n < d$，诱导一个映射 $\mathbf{x} \mapsto W\mathbf{x}$，其中 $W\mathbf{x} \in \mathbb{R}^n$ 是 $\mathbf{x}$ 的低维表示。然后，可以使用第二个矩阵 $U \in \mathbb{R}^{d,n}$ 来（近似地）从其压缩版本恢复每个原始向量 $\mathbf{x}$。也就是说，对于一个压缩向量 $\mathbf{y} = W\mathbf{x}$，其中 $\mathbf{y}$ 在低维空间 $\mathbb{R}^n$ 中，我们可以构造 $\tilde{\mathbf{x}} = U\mathbf{y}$，使得 $\tilde{\mathbf{x}}$ 是 $\mathbf{x}$ 的恢复版本，并位于原始高维空间 $\mathbb{R}^d$ 中。

在PCA中，我们找到压缩矩阵 W 和恢复矩阵 U，使得原始向量和恢复向量之间的总平方距离最小；也就是说，我们旨在解决以下问题

$$\operatorname*{argmin}_{W \in \mathbb{R}^{n,d}, U \in \mathbb{R}^{d,n}} \sum_{i=1}^{m} \|\mathbf{x}_i - UW\mathbf{x}_i\|_2^2. \tag{23.1}$$

要解决这个问题，我们首先证明最优解具有特定的形式。

LEMMA 23.1 *Let (U, W) be a solution to Equation (23.1). Then the columns of U are orthonormal (namely, $U^\top U$ is the identity matrix of $\mathbb{R}^n$) and $W = U^\top$.*

Proof 修复任何 U, W 并考虑映射 $\mathbf{x} \mapsto UW\mathbf{x}$。此映射的范围，$R = \{UW\mathbf{x} : \mathbf{x} \in \mathbb{R}^d\}$，是 $\mathbb{R}^d$ 的一个 n 维线性子空间。设 $V \in \mathbb{R}^{d,n}$ 为一个矩阵，其列构成此子空间的正交基，即 V 的范围是 R 和 $V^\top V = I$。因此，R 中的每个向量都可以写成 $V\mathbf{y}$，其中 $\mathbf{y} \in \mathbb{R}^n$。对于每个 $\mathbf{x} \in \mathbb{R}^d$ 和 $\mathbf{y} \in \mathbb{R}^n$，我们有

$$\|\mathbf{x} - V\mathbf{y}\|_2^2 \;=\; \|\mathbf{x}\|^2 + \mathbf{y}^\top V^\top V\mathbf{y} - 2\mathbf{y}^\top V^\top \mathbf{x} \;=\; \|\mathbf{x}\|^2 + \|\mathbf{y}\|^2 - 2\mathbf{y}^\top (V^\top \mathbf{x}),$$

在式中，我们使用了 $V^\top V$ 是 $\mathbb{R}^n$ 的单位矩阵的事实。通过将前述表达式相对于 $\mathbf{y}$ 最小化，并比较相对于 $\mathbf{y}$ 的梯度与零，得到 $\mathbf{y} = V^\top \mathbf{x}$。因此，对于每个 $\mathbf{x}$，我们有

$$VV^\top \mathbf{x} = \operatorname*{argmin}_{\tilde{\mathbf{x}} \in R} \|\mathbf{x} - \tilde{\mathbf{x}}\|_2^2.$$

特别地，这对于 $\mathbf{x}_1, \ldots, \mathbf{x}_m$ 成立，因此我们可以用 $V, V^\top$ 代替 U, W，而不会增加目标函数

$$\sum_{i=1}^{m} \|\mathbf{x}_i - UW\mathbf{x}_i\|_2^2 \;\geq\; \sum_{i=1}^{m} \|\mathbf{x}_i - VV^\top \mathbf{x}_i\|_2^2.$$

由于这对每个 U, W 都成立，引理的证明随之而来。 □

基于前面的引理，我们可以将方程（23.1）中给出的优化问题重新写为如下：

$$\operatorname*{argmin}_{U \in \mathbb{R}^{d,n} : U^\top U = I} \sum_{i=1}^{m} \|\mathbf{x}_i - UU^\top \mathbf{x}_i\|_2^2. \tag{23.2}$$

我们进一步通过以下基本的代数操作简化优化问题。对于每个 $\mathbf{x} \in \mathbb{R}^d$ 和满足 $U^\top U = I$ 的矩阵 $U \in \mathbb{R}^{d,n}$，我们有

$$\begin{aligned}\|\mathbf{x} - UU^\top \mathbf{x}\|^2 &= \|\mathbf{x}\|^2 - 2\mathbf{x}^\top UU^\top \mathbf{x} + \mathbf{x}^\top UU^\top UU^\top \mathbf{x} \\ &= \|\mathbf{x}\|^2 - \mathbf{x}^\top UU^\top \mathbf{x} \\ &= \|\mathbf{x}\|^2 - \operatorname{trace}(U^\top \mathbf{x}\mathbf{x}^\top U), \end{aligned} \tag{23.3}$$

矩阵的迹是其对角线元素之和。由于迹是一个线性算子，这允许我们将方程（23.2）重写如下：

$$\operatorname*{argmax}_{U \in \mathbb{R}^{d,n}: U^\top U = I} \operatorname{trace}\left(U^\top \sum_{i=1}^m \mathbf{x}_i \mathbf{x}_i^\top U \right). \tag{23.4}$$

让 $A = \sum_{i=1}^m \mathbf{x}_i \mathbf{x}_i^\top$。矩阵 A 是对称的，因此它可以写成其谱分解形式 $A = VDV^\top$，其中 D 是对角线矩阵，$V^\top V = VV^\top = I$。在这里，D 的对角线元素是 A 的特征值，V的列是相应的特征向量。我们假设不失一般性 $D_{1,1} \geq D_{2,2} \geq \cdots \geq D_{d,d}$。由于 A 是正半定矩阵，因此也成立 $D_{d,d} \geq 0$。我们断言方程（23.4）的解是矩阵 U，其列是 A 对应于最大 n 特征值的 n 特征向量。

THEOREM 23.2 *Let $\mathbf{x}_1, \ldots, \mathbf{x}_m$ be arbitrary vectors in $\mathbb{R}^d$, let $A = \sum_{i=1}^m \mathbf{x}_i \mathbf{x}_i^\top$, and let $\mathbf{u}_1, \ldots, \mathbf{u}_n$ be n eigenvectors of the matrix A corresponding to the largest n eigenvalues of A. Then, the solution to the PCA optimization problem given in Equation (23.1) is to set U to be the matrix whose columns are $\mathbf{u}_1, \ldots, \mathbf{u}_n$ and to set $W = U^\top$.*

Proof 设 $VDV^\top$ 为 A 的谱分解。取一个具有正交列的矩阵 $U \in \mathbb{R}^{d,n}$ 并令 $B = V^\top U$。然后，$VB = VV^\top U = U$。由此可得

$$U^\top AU = B^\top V^\top VDV^\top VB = B^\top DB,$$

因此

$$\operatorname{trace}(U^\top AU) = \sum_{j=1}^d D_{j,j} \sum_{i=1}^n B_{j,i}^2.$$

注意 $B^\top B = U^\top VV^\top U = U^\top U = I$。因此，$B$ 的列也是正交归一的，这意味着 $\sum_{j=1}^d \sum_{i=1}^n B_{j,i}^2 = n$。此外，设 $\tilde{B} \in \mathbb{R}^{d,d}$ 为一个矩阵，使得其前 n 列是 B 的列，并且此外 $\tilde{B}^\top \tilde{B} = I$。那么，对于每个 j，我们有 $\sum_{i=1}^d \tilde{B}_{j,i}^2 = 1$，这意味着 $\sum_{i=1}^n B_{j,i}^2 \leq 1$。因此：

$$\operatorname{trace}(U^\top AU) \leq \max_{\boldsymbol{\beta} \in [0,1]^d : \|\boldsymbol{\beta}\|_1 \leq n} \sum_{j=1}^d D_{j,j} \beta_j .$$

不容易验证（见练习2）右边等于$\sum_{j=1}^{n} D_{j,j}$。因此，我们已经证明对于每个具有正交列的矩阵$U \in \mathbb{R}^{d,n}$，都有迹$(U^\top AU) \leq \sum_{j=1}^{n} D_{j,j}$。另一方面，如果我们设$U$为列是$A$的$n$个主特征向量的矩阵，我们得到迹$(U^\top AU) = \sum_{j=1}^{n} D_{j,j}$，这也就完成了我们的证明。

□

Remark 23.1 定理23.2的证明还告诉我们，方程（23.4）的目标值是$\sum_{i=1}^{n} D_{i,i}$。将此与方程（23.3）相结合，并注意到$\sum_{i=1}^{m} \|\mathbf{x}_i\|^2 = \text{trace}(A) = \sum_{i=1}^{d} D_{i,i}$，我们得到方程（23.1）的最优目标值是$\sum_{i=n+1}^{d} D_{i,i}$。

Remark 23.2 这是一个常见的做法，在应用PCA之前“中心化”示例。也就是说，我们首先计算$\boldsymbol{\mu} = \frac{1}{m}\sum_{i=1}^{m} \mathbf{x}_i$，然后对向量$(\mathbf{x}_1 - \boldsymbol{\mu}), \ldots, (\mathbf{x}_m - \boldsymbol{\mu})$应用PCA。这也与将PCA解释为方差最大化有关（参见练习4）。

23.1.1 A More Efficient Solution for the Case $d \gg m$

在某些情况下，数据的原始维度远大于示例数量m。如前所述，计算PCA解的计算复杂度为$O(d^3)$（计算特征值A）的复杂度加上$O(md^2)$（构建矩阵A）的复杂度。我们现在展示一个简单的技巧，使我们能够在$d \gg m$时更有效地计算PCA解。

回忆矩阵A被定义为$\sum_{i=1}^{m} \mathbf{x}_i \mathbf{x}_i^\top$。将$A = X^\top X$重写是方便的，其中$X \in \mathbb{R}^{m,d}$是一个矩阵，其$i$行是$\mathbf{x}_i^\top$。考虑矩阵$B = XX^\top$。也就是说，$B \in \mathbb{R}^{m,m}$是一个矩阵，其$i,j$元素等于$\langle \mathbf{x}_i, \mathbf{x}_j \rangle$。假设$\mathbf{u}$是$B$的一个特征向量：即，$B\mathbf{u} = \lambda\mathbf{u}$对于某个$\lambda \in \mathbb{R}$。将等式乘以$X^\top$并使用$B$的定义，我们得到$X^\top XX^\top \mathbf{u} = \lambda X^\top \mathbf{u}$。但是，使用$A$的定义，我们得到$A(X^\top \mathbf{u}) = \lambda(X^\top \mathbf{u})$。因此，$\frac{X^\top \mathbf{u}}{\|X^\top \mathbf{u}\|}$是$A$的一个特征向量，其特征值为$\lambda$。

我们可以因此通过计算B的特征值来计算PCA解，而不是A。计算特征值的复杂度是$O(m^3)$（，构建矩阵B）的复杂度是$m^2 d$（。

Remark 23.3 前一次讨论还暗示，为了计算主成分分析（PCA）解，我们只需要知道如何计算向量之间的内积。这使得我们能够在向量d非常大（甚至无限大）的情况下，通过核函数隐式地计算PCA，从而得到*kernel PCA*算法。

23.1.2 Implementation and Demonstration

A pseudocode of PCA is given in the following.

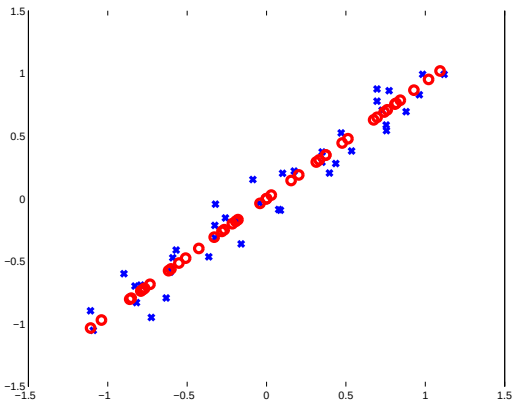


Figure 23.1 一组在 $\mathbb{R}^2$（蓝色 x 的向量及其使用 PCA（红色圆圈）降维到 $\mathbb{R}^1$ 后的重建。

PCA

input
A matrix of m examples $X \in \mathbb{R}^{m,d}$
number of components n
if $(m > d)$
$A = X^\top X$
Let $\mathbf{u}_1, \ldots, \mathbf{u}_n$ be the eigenvectors of A with largest eigenvalues
else
$B = XX^\top$
Let $\mathbf{v}_1, \ldots, \mathbf{v}_n$ be the eigenvectors of B with largest eigenvalues
for $i = 1, \ldots, n$ set $\mathbf{u}_i = \frac{1}{\|X^\top \mathbf{v}_i\|} X^\top \mathbf{v}_i$
output: $\mathbf{u}_1, \ldots, \mathbf{u}_n$

为了说明PCA是如何工作的，让我们在$\mathbb{R}^2$中生成一些向量，这些向量大致位于一条直线上，即在$\mathbb{R}^2$的一维子空间中。例如，假设每个示例的形式为$(x, x+y)$，其中x是从$[-1, 1]$中均匀随机选择的，而y是从均值为0、标准差为0.1的高斯分布中抽取的。假设我们对这些数据应用PCA。那么，对应于最大特征值的特征向量将接近向量$(1/\sqrt{2}, 1/\sqrt{2})$。当将一个点$(x, x+y)$投影到这个主成分上时，我们将获得标量$\frac{2x+y}{\sqrt{2}}$。原始向量的重建将是$((x+y/2), (x+y/2))$。在图23.1中，我们描绘了原始数据与重建数据。

接下来，我们展示了PCA在人脸数据集上的有效性。我们从耶鲁数据集（Georghiades，Belhumeur & Kriegman 2001）中提取了人脸图像。每张图像包含$50 \times 50 = 2500$个像素；因此原始维度非常高。

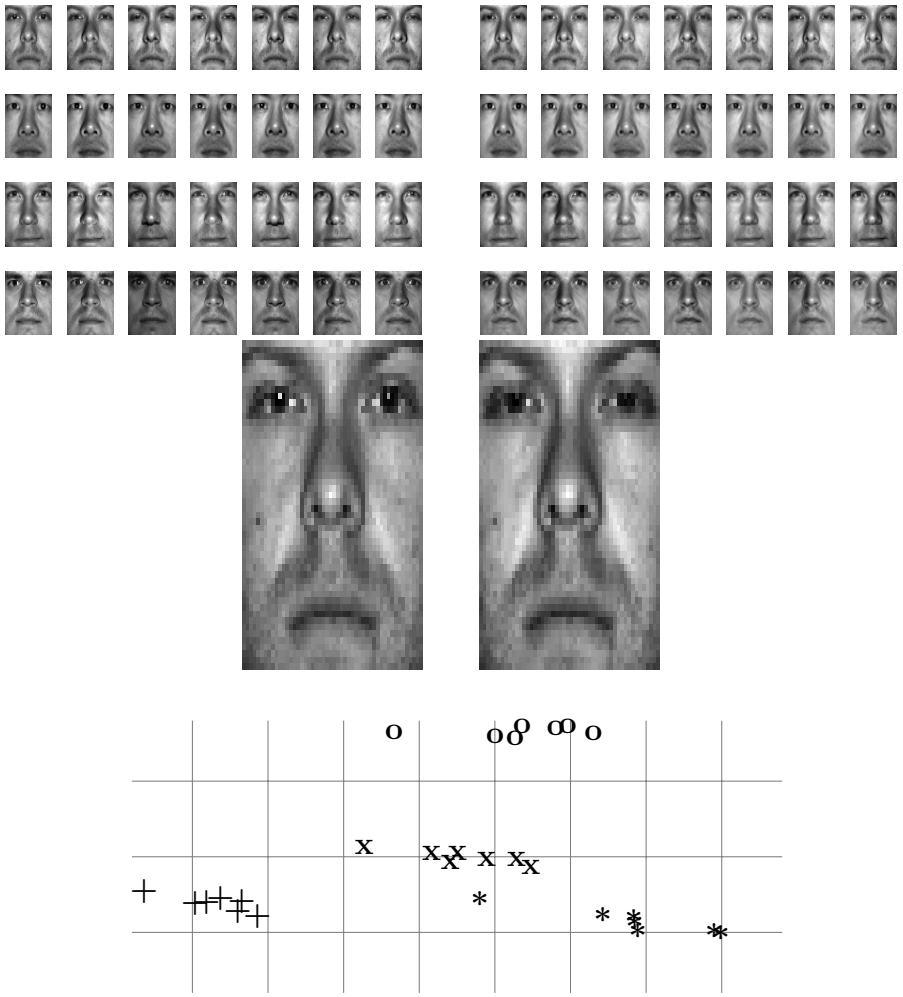


Figure 23.2 从耶鲁数据集中提取的人脸图像。左上：$\mathbb{R}^{50x50}$中的原始图像。右上：降维到$\mathbb{R}^{10}$和重建后的图像。中间行：PCA前后图像的放大版本。底部：降维到$\mathbb{R}^{2}$后的图像。不同的标记表示不同的人。

某些面部图像描绘在图23.2的左上角。使用PCA，我们将维度降低到$\mathbb{R}^{10}$，并重新构建回原始维度，即50^2。得到的重建图像描绘在图23.2的右上角。最后，在图23.2的底部，我们描绘了图像的二维表示。如图所示，即使从图像的二维表示中，我们仍然可以大致区分不同的个体。

23.2 Random Projections

在这个部分，我们展示了通过使用随机线性变换来降低维度会导致一个简单且出人意料地低失真的压缩方案。当 W 是一个随机矩阵时，变换 $\mathbf{x} \mapsto W\mathbf{x}$ 通常被称为随机投影。特别是，我们提供了一个由 Johnson 和 Lindenstrauss 提出的著名引理的变体，表明随机投影不会过多地扭曲欧几里得距离。

设 $\mathbf{x}_1, \mathbf{x}_2$ 是 $\mathbb{R}^d$ 中的两个向量。如果矩阵 W 不太扭曲 $\mathbf{x}_1$ 和 $\mathbf{x}_2$ 之间的距离，那么它们的比

$$\frac{\|W\mathbf{x}_1 - W\mathbf{x}_2\|}{\|\mathbf{x}_1 - \mathbf{x}_2\|}$$

接近1。换句话说，变换前后$\mathbf{x}_1$和$\mathbf{x}_2$之间的距离几乎相同。为了表明$\|W\mathbf{x}_1 - W\mathbf{x}_2\|$与$\|\mathbf{x}_1 - \mathbf{x}_2\|$的距离不远，只需表明$W$不会扭曲差分向量$\mathbf{x} = \mathbf{x}_1 - \mathbf{x}_2$的范数。因此，从现在起我们关注比率$\frac{\|W\mathbf{x}\|}{\|\mathbf{x}\|}$。
我们从分析对单个向量应用随机投影引起的失真开始。

LEMMA 23.3 *Fix some $\mathbf{x} \in \mathbb{R}^d$. Let $W \in \mathbb{R}^{n,d}$ be a random matrix such that each $W_{i,j}$ is an independent normal random variable. Then, for every $\epsilon \in (0, 3)$ we have*

$$\mathbb{P}\left[\left| \frac{\|(1/\sqrt{n})W\mathbf{x}\|^2}{\|\mathbf{x}\|^2} - 1 \right| > \epsilon \right] \leq 2\, e^{-\epsilon^2 n/6}.$$

Proof 无失真的情况下，我们可以假设 $\|\mathbf{x}\|^2 = 1$。因此，一个等价的不等式是

$$\mathbb{P}\left[(1-\epsilon)n \leq \|W\mathbf{x}\|^2 \leq (1+\epsilon)n\right] \geq 1 - 2e^{-\epsilon^2 n/6}.$$

设 $\mathbf{w}_i$ 为 W 的 i 行。随机变量 $\langle \mathbf{w}_i, \mathbf{x} \rangle$ 是 d 个独立正态随机变量的加权求和，因此它服从均值为零、方差为 $\sum_j x_j^2 = \|\mathbf{x}\|^2 = 1$ 的正态分布。因此，随机变量 $\|W\mathbf{x}\|^2 = \sum_{i=1}^n (\langle \mathbf{w}_i, \mathbf{x} \rangle)^2$ 具有分布 χ_n^2。现在，该陈述直接从第 B.7 节中给出的引理 B.12 所述的 χ^2 个随机变量的测度集中性质得出。

□

约翰逊-林德斯特拉斯引理可以通过简单的并集界论证得出。

LEMMA 23.4 (约翰逊-林德斯特拉斯引理) *Let Q be a finite set of vectors in $\mathbb{R}^d$. Let $\delta \in (0, 1)$ and n be an integer such that*

$$\epsilon = \sqrt{\frac{6\, \log(2|Q|/\delta)}{n}} \leq 3.$$

Then, with probability of at least $1-\delta$ *over a choice of a random matrix* $W \in \mathbb{R}^{n,d}$ *such that each element of* W *is distributed normally with zero mean and variance of* $1/n$ *we have*

$$\sup_{\mathbf{x}\in Q}\left|\frac{\|W\mathbf{x}\|^2}{\|\mathbf{x}\|^2}-1\right|<\epsilon.$$

Proof 结合引理23.3和并集界，我们有对于每个 $\epsilon \in (0,3)$：

$$\mathbb{P}\left[\sup_{\mathbf{x}\in Q}\left|\frac{\|W\mathbf{x}\|^2}{\|\mathbf{x}\|^2}-1\right|>\epsilon\right]\leq 2\,|Q|\,e^{-\epsilon^2 n/6}.$$

让 δ 表示不等式的右侧；因此我们得到

$$\epsilon=\sqrt{\frac{6\,\log(2|Q|/\delta)}{n}}.$$

□

有趣的是，命题23.4中给出的界不依赖于向量 $\mathbf{x}$ 的原始维度。事实上，即使 $\mathbf{x}$ 在一个无限维希尔伯特空间中，这个界仍然成立。

23.3 Compressed Sensing

压缩感知是一种降维技术，它利用一个先验假设，即原始向量在某些基中是稀疏的。为了激发压缩感知，考虑一个最多有 s 个非零元素的向量 $\mathbf{x} \in \mathbb{R}^d$。也就是说，

$$\|\mathbf{x}\|_0 \stackrel{\text{def}}{=} |\{i : x_i \neq 0\}| \leq s.$$

显然，我们可以通过使用索引，值对来表示 $\mathbf{x}$，从而压缩 $\mathbf{x}$。此外，这种压缩是无损的——我们可以从 s (索引，值) 对中精确地重建 $\mathbf{x}$。现在，让我们再向前迈一步，假设 $\mathbf{x} = U\boldsymbol{\alpha}$，其中 $\boldsymbol{\alpha}$ 是一个稀疏向量 $\|\boldsymbol{\alpha}\|_0 \leq s$，$U$ 是一个固定正交归一矩阵。也就是说，$\mathbf{x}$ 在另一个基中具有稀疏表示。实际上，许多自然向量在某种表示中（至少是近似地）是稀疏的。事实上，这个假设是许多现代压缩方案的基础。例如，JPEG-2000 图像压缩格式依赖于自然图像在小波基中是近似稀疏的事实。

我们还能将 $\mathbf{x}$ 压缩成大约 s 个数吗？嗯，一个简单的方法是将 $\mathbf{x}$ 乘以 $U^\top$，得到稀疏向量 $\boldsymbol{\alpha}$，然后通过其 s (索引，值) 对表示 $\boldsymbol{\alpha}$。然而，这需要我们首先“感知” $\mathbf{x}$，存储它，然后将其乘以 $U^\top$。这引发了一个非常自然的问题：为什么费这么大力气获取所有数据，而我们得到的大部分数据最终都会被丢弃？我们难道不能直接测量最终不会被丢弃的部分吗？

压缩感知是一种同时获取和压缩数据的技术。关键结果是随机线性变换可以在不丢失信息的情况下压缩 $\mathbf{x}$。所需的测量次数是 $s \log(d)$ 的数量级。也就是说，我们大致只获取关于信号的 重要信息。正如我们稍后将会看到的，我们付出的代价是更慢的重建阶段。在某些情况下，即使以牺牲更慢的重建为代价，节省压缩时间也是有意义的。例如，安全摄像头应该感应和压缩大量图像，而大多数时候我们根本不需要解码压缩数据。此外，在许多实际应用中，通过线性变换进行压缩是有利的，因为它可以在硬件中有效地执行。例如，由Baraniuk和Kelly领导的团队提出了一种采用数字微镜阵列来执行图像线性变换的光学计算的相机架构。在这种情况下，获取每个压缩测量值就像获取单个原始测量值一样简单。压缩感知的另一个重要应用是医学成像，其中减少测量次数意味着患者接受的辐射更少。

非正式地，压缩感知的主要前提是以下三个“令人惊讶”的结果：

1. 如果任何稀疏信号被$\mathbf{x} \mapsto W\mathbf{x}$压缩，则可以完全重建该信号，其中$W$是一个满足称为限制等周性质（RIP）的条件的矩阵。满足此性质的矩阵保证任何可稀疏表示的向量的范数具有低失真。

2. 通过求解线性规划，重建可以在多项式时间内计算。

3. 随机 $n \times d$ 矩阵很可能会满足RIP条件，前提是 n 大于 s 次方 $\log(d)$。

形式上，

DEFINITION 23.5 (RIP) 一个矩阵 $W \in \mathbb{R}^{n,d}$ 是 (ϵ, s)-RIP，如果对于所有 $\mathbf{x} \neq 0$ 满足 $\|\mathbf{x}\|_0 \leq s$，我们有

$$\left| \frac{\|W\mathbf{x}\|_2^2}{\|\mathbf{x}\|_2^2} - 1 \right| \leq \epsilon.$$

第一个定理确立了RIP矩阵为稀疏向量提供无损压缩方案。它还提供了一个（非有效）重建方案。

THEOREM 23.6 *Let $\epsilon < 1$ and let W be a $(\epsilon, 2s)$-RIP matrix. Let $\mathbf{x}$ be a vector s.t. $\|\mathbf{x}\|_0 \leq s$, let $\mathbf{y} = W\mathbf{x}$ be the compression of $\mathbf{x}$, and let*

$$\tilde{\mathbf{x}} \in \operatorname*{argmin}_{\mathbf{v}: W\mathbf{v} = \mathbf{y}} \|\mathbf{v}\|_0$$

be a reconstructed vector. Then, $\tilde{\mathbf{x}} = \mathbf{x}$.

Proof 我们假设，通过矛盾法，$\tilde{\mathbf{x}} \neq \mathbf{x}$。由于 $\mathbf{x}$ 满足定义 $\tilde{\mathbf{x}}$ 的优化问题中的约束，我们显然有 $\|\tilde{\mathbf{x}}\|_0 \leq \|\mathbf{x}\|_0 \leq s$。因此，$\|\mathbf{x} - \tilde{\mathbf{x}}\|_0 \leq 2s$，并且我们可以对向量 $\mathbf{x} - \tilde{\mathbf{x}}$ 应用 RIP 不等式。但是，由于 $W(\mathbf{x} - \tilde{\mathbf{x}}) = \mathbf{0}$，我们得到 $|0 - 1| \leq \epsilon$，这导致矛盾。

□

定理23.6中给出的重建方案似乎效率不高，因为我们需要最小化一个组合目标（$\mathbf{v}$的稀疏性）。令人惊讶的是，我们实际上可以用凸目标$\|\mathbf{v}\|_1$替换组合目标$\|\mathbf{v}\|_0$，这导致了一个可以高效解决的线性规划问题。这一点在以下定理中得到了正式陈述。

THEOREM 23.7 *Assume that the conditions of Theorem 23.6 holds and that* $\epsilon < \frac{1}{1+\sqrt{2}}$. *Then,*

$$\mathbf{x} = \operatorname*{argmin}_{\mathbf{v}:W\mathbf{v}=\mathbf{y}} \|\mathbf{v}\|_0 = \operatorname*{argmin}_{\mathbf{v}:W\mathbf{v}=\mathbf{y}} \|\mathbf{v}\|_1.$$

实际上，我们将证明一个更强的结果，即使 $\mathbf{x}$ 不是一个稀疏向量，这个结果也成立。

THEOREM 23.8 *Let* $\epsilon < \frac{1}{1+\sqrt{2}}$ *and let* W *be a* $(\epsilon, 2s)$*-RIP matrix. Let* $\mathbf{x}$ *be an arbitrary vector and denote*

$$\mathbf{x}_s \in \operatorname*{argmin}_{\mathbf{v}:\|\mathbf{v}\|_0 \leq s} \|\mathbf{x} - \mathbf{v}\|_1.$$

That is, $\mathbf{x}_s$ *is the vector which equals* $\mathbf{x}$ *on the* s *largest elements of* $\mathbf{x}$ *and equals* 0 *elsewhere. Let* $\mathbf{y} = W\mathbf{x}$ *be the compression of* $\mathbf{x}$ *and let*

$$\mathbf{x}^\star \in \operatorname*{argmin}_{\mathbf{v}:W\mathbf{v}=\mathbf{y}} \|\mathbf{v}\|_1$$

be the reconstructed vector. Then,

$$\|\mathbf{x}^\star - \mathbf{x}\|_2 \leq 2\frac{1+\rho}{1-\rho}\, s^{-1/2}\|\mathbf{x} - \mathbf{x}_s\|_1,$$

where $\rho = \sqrt{2}\epsilon/(1-\epsilon)$.

注意，在特殊情况下，当 $\mathbf{x} = \mathbf{x}_s$ 时，我们得到精确恢复 $\mathbf{x}^\star = \mathbf{x}$，因此定理23.7是定理23.8的特殊情况。定理23.8的证明见第23.3.1节。

最后，第三个结果告诉我们，具有 $n \geq \Omega(s \log(d))$ 的随机矩阵很可能是RIP。事实上，该定理表明，将随机矩阵乘以一个正交矩阵也提供了一个RIP矩阵。这对于压缩形式为 $\mathbf{x} = U\boldsymbol{\alpha}$ 的信号很重要，其中 $\mathbf{x}$ 不是稀疏的，而 $\boldsymbol{\alpha}$ 是稀疏的。在这种情况下，如果 W 是一个随机矩阵，并且我们使用 $\mathbf{y} = W\mathbf{x}$ 进行压缩，那么这相当于使用 $\mathbf{y} = (WU)\boldsymbol{\alpha}$ 压缩 $\boldsymbol{\alpha}$，由于 WU 也是RIP，我们可以从 $\mathbf{y}$ 中重建 $\boldsymbol{\alpha}$ (，从而也可以重建 $\mathbf{x}$)。

THEOREM 23.9 *Let U be an arbitrary fixed $d \times d$ orthonormal matrix, let ϵ, δ be scalars in $(0,1)$, let s be an integer in $[d]$, and let n be an integer that satisfies*

$$n \geq 100 \frac{s \log(40d/(\delta\epsilon))}{\epsilon^2}.$$

Let $W \in \mathbb{R}^{n,d}$ be a matrix s.t. each element of W is distributed normally with zero mean and variance of $1/n$. Then, with proabability of at least $1-\delta$ over the choice of W, the matrix WU is (ϵ, s)-RIP.

23.3.1 Proofs*

Proof of Theorem 23.8
我们遵循Candès（2008）的一个证明。

设 $\mathbf{h} = \mathbf{x}^\star - \mathbf{x}$。给定一个向量 $\mathbf{v}$ 和一个索引集 I，我们记 $\mathbf{v}_I$ 为一个向量，其 i 个元素是 v_i，如果 $i \in I$，否则为 0。

第一个技巧是将索引集 $[d] = \{1, \ldots, d\}$ 划分为大小为 s 的不相交子集。也就是说，我们将写成 $[d] = T_0 \cup T_1 \cup T_2 \ldots T_{d/s-1}$，其中对于所有 i，$|T_i| = s$，并且我们为了简单起见假设 d/s 是一个整数。我们定义划分如下。在 T_0 中，我们放入与 s 索引相对应的 s 最大元素，在 $\mathbf{x}$ (绝对值中的) 级别是任意打破的。让 $T_0^c = [d] \setminus T_0$。接下来，$T_1$ 将是与 s 索引相对应的 s 最大元素，在绝对值 $\mathbf{h}_{T_0^c}$ 中的。让 $T_{0,1} = T_0 \cup T_1$ 和 $T_{0,1}^c = [d] \setminus T_{0,1}$。接下来，$T_2$ 将对应于绝对值 $\mathbf{h}_{T_{0,1}^c}$ 中的 s 最大元素。并且，我们将以相同的方式构建 $T_3, T_4, \ldots$。

为了证明该定理，我们首先需要以下引理，它表明RIP也意味着近似正交性。

LEMMA 23.10 *Let W be an $(\epsilon, 2s)$-RIP matrix. Then, for any two disjoint sets I, J, both of size at most s, and for any vector $\mathbf{u}$ we have that $\langle W\mathbf{u}_I, W\mathbf{u}_J \rangle \leq \epsilon \|\mathbf{u}_I\|_2 \|\mathbf{u}_J\|_2$.*

Proof W.l.o.g. 假设 $\|\mathbf{u}_I\|_2 = \|\mathbf{u}_J\|_2 = 1$.

$$\langle W\mathbf{u}_I, W\mathbf{u}_J \rangle = \frac{\|W\mathbf{u}_I + W\mathbf{u}_J\|_2^2 - \|W\mathbf{u}_I - W\mathbf{u}_J\|_2^2}{4}.$$

但是，由于从RIP条件中得到 $|J \cup I| \leq 2s$ 我们得到 $\|W\mathbf{u}_I + W\mathbf{u}_J\|_2^2 \leq (1+\epsilon)(\|\mathbf{u}_I\|_2^2 + \|\mathbf{u}_J\|_2^2) = 2(1+\epsilon)$ 以及 $-\|W\mathbf{u}_I - W\mathbf{u}_J\|_2^2 \leq -(1-\epsilon)(\|\mathbf{u}_I\|_2^2 + \|\mathbf{u}_J\|_2^2) = -2(1-\epsilon)$，这证明了我们的结论。 □

我们现在准备好证明这个定理。显然，

$$\|\mathbf{h}\|_2 = \|\mathbf{h}_{T_{0,1}} + \mathbf{h}_{T_{0,1}^c}\|_2 \leq \|\mathbf{h}_{T_{0,1}}\|_2 + \|\mathbf{h}_{T_{0,1}^c}\|_2. \tag{23.5}$$

T为了证明该定理，我们将展示以下两个命题 ms:

Claim 1:. $\|\mathbf{h}_{T_{0,1}^c}\|_2 \leq \|\mathbf{h}_{T_0}\|_2 + 2s^{-1/2}\|\mathbf{x} - \mathbf{x}_s\|_1$

。 **Claim 2:.** $\|\mathbf{h}_{T_{0,1}}\|_2 \leq \frac{2\rho}{1-\rho} s^{-1/2}\|\mathbf{x} - \mathbf{x}_s\|_1$。

这两个主张与方程（23.5）相结合，我们得到

$$\begin{aligned}\|\mathbf{h}\|_2 &\le \|\mathbf{h}_{T_{0,1}}\|_2 + \|\mathbf{h}_{T_{0,1}^c}\|_2 \le 2\|\mathbf{h}_{T_{0,1}}\|_2 + 2s^{-1/2}\|\mathbf{x}-\mathbf{x}_s\|_1 \\ &\le 2\left(\tfrac{2\rho}{1-\rho}+1\right)s^{-1/2}\|\mathbf{x}-\mathbf{x}_s\|_1 \\ &= 2\frac{1+\rho}{1-\rho}\,s^{-1/2}\|\mathbf{x}-\mathbf{x}_s\|_1,\end{aligned}$$

并且这将是我们的证明的结论。

Proving Claim 1:
为了证明这个断言，我们根本不使用RIP条件，而只使用事实，即$\mathbf{x}^\star$最小化了ℓ_1范数。取$j>1$。对于每个$i\in T_j$和$i'\in T_{j-1}$，我们有$|h_i|\le|h_{i'}|$。因此，$\|\mathbf{h}_{T_j}\|_\infty \le \|\mathbf{h}_{T_{j-1}}\|_1/s$。因此，

$$\|\mathbf{h}_{T_j}\|_2 \le s^{1/2}\|\mathbf{h}_{T_j}\|_\infty \le s^{-1/2}\|\mathbf{h}_{T_{j-1}}\|_1.$$

对 $j=2,3,\ldots$ 求和，并使用三角不等式，我们得到

$$\|\mathbf{h}_{T_{0,1}^c}\|_2 \le \sum_{j\ge 2}\|\mathbf{h}_{T_j}\|_2 \le s^{-1/2}\|\mathbf{h}_{T_0^c}\|_1 \tag{23.6}$$

接下来，我们证明 $\|\mathbf{h}_{T_0^c}\|_1$ 不能很大。事实上，根据 $\mathbf{x}^\star$ 的定义，我们有 $\|\mathbf{x}\|_1 \ge \|\mathbf{x}^\star\|_1 = \|\mathbf{x}+\mathbf{h}\|_1$。因此，利用三角不等式，我们得到

$$\|\mathbf{x}\|_1 \ge \|\mathbf{x}+\mathbf{h}\|_1 = \sum_{i\in T_0}|x_i+h_i| + \sum_{i\in T_0^c}|x_i+h_i| \ge \|\mathbf{x}_{T_0}\|_1 - \|\mathbf{h}_{T_0}\|_1 + \|\mathbf{h}_{T_0^c}\|_1 - \|\mathbf{x}_{T_0^c}\|_1 \tag{23.7}$$

自 $\|\mathbf{x}_{T_0^c}\|_1 = \|\mathbf{x}-\mathbf{x}_s\|_1 = \|\mathbf{x}\|_1 - \|\mathbf{x}_{T_0}\|_1$ 以来，我们得到

$$\|\mathbf{h}_{T_0^c}\|_1 \le \|\mathbf{h}_{T_0}\|_1 + 2\|\mathbf{x}_{T_0^c}\|_1. \tag{23.8}$$

将此与方程（23.6）结合，我们得到

$$\|\mathbf{h}_{T_{0,1}^c}\|_2 \le s^{-1/2}\left(\|\mathbf{h}_{T_0}\|_1 + 2\|\mathbf{x}_{T_0^c}\|_1\right) \le \|\mathbf{h}_{T_0}\|_2 + 2s^{-1/2}\|\mathbf{x}_{T_0^c}\|_1,$$

这证明了主张1。

Proving Claim 2:
对于第二个主张，我们使用RIP条件得到：{v*}

$$(1-\epsilon)\|\mathbf{h}_{T_{0,1}}\|_2^2 \le \|W\mathbf{h}_{T_{0,1}}\|_2^2. \tag{23.9}$$

自 $W\mathbf{h}_{T_{0,1}} = W\mathbf{h} - \sum_{j\ge2}W\mathbf{h}_{T_j} = -\sum_{j\ge2}W\mathbf{h}_{T_j}$ 以来，我们有

$$\|W\mathbf{h}_{T_{0,1}}\|_2^2 = -\sum_{j\ge2}\langle W\mathbf{h}_{T_{0,1}}, W\mathbf{h}_{T_j}\rangle = -\sum_{j\ge2}\langle W\mathbf{h}_{T_0}+W\mathbf{h}_{T_1}, W\mathbf{h}_{T_j}\rangle.$$

从内积的RIP条件中，我们得到对于所有 $i\in\{1,2\}$ 和 $j\ge2$，我们有

$$|\langle W\mathbf{h}_{T_i}, W\mathbf{h}_{T_j}\rangle| \le \epsilon\|\mathbf{h}_{T_i}\|_2\|\mathbf{h}_{T_j}\|_2.$$

自 $\|\mathbf{h}_{T_0}\|_2 + \|\mathbf{h}_{T_1}\|_2 \leq \sqrt{\wedge}2\|\mathbf{h}_{T_{0,1}}\|_2$ 以来，因此我们得到

$$\|W\mathbf{h}_{T_{0,1}}\|_2^2 \;\leq\; \sqrt{2}\epsilon\|\mathbf{h}_{T_{0,1}}\|_2 \sum_{j\geq 2}\|\mathbf{h}_{T_j}\|_2.$$

将此与方程（23.6）和方程（23.9）结合，我们得到

$$(1-\epsilon)\|\mathbf{h}_{T_{0,1}}\|_2^2 \leq \sqrt{2}\epsilon\|\mathbf{h}_{T_{0,1}}\|_2 s^{-1/2}\|\mathbf{h}_{T_0^c}\|_1.$$

重新排列不等式得到

$$\|\mathbf{h}_{T_{0,1}}\|_2 \leq \frac{\sqrt{2}\epsilon}{1-\epsilon} s^{-1/2}\|\mathbf{h}_{T_0^c}\|_1.$$

最后，使用方程式（23.8）我们得到：

$$\|\mathbf{h}_{T_{0,1}}\|_2 \leq \rho s^{-1/2}\left(\|\mathbf{h}_{T_0}\|_1 + 2\|\mathbf{x}_{T_0^c}\|_1\right) \leq \rho\|\mathbf{h}_{T_0}\|_2 + 2\rho s^{-1/2}\|\mathbf{x}_{T_0^c}\|_1,$$

但是自 $\|\mathbf{h}_{T_0}\|_2 \leq \|\mathbf{h}_{T_{0,1}}\|_2$ 以来这表示

$$\|\mathbf{h}_{T_{0,1}}\|_2 \leq \frac{2\rho}{1-\rho}\, s^{-1/2}\|\mathbf{x}_{T_0^c}\|_1,$$

这证明了第二个命题的结论。

Proof of Theorem 23.9
为了证明该定理，我们遵循了（Baraniuk，Davenport，De-Vore & Wakin 2008）提出的方法。该思路是将Johnson-Lindenstrauss（JL）引理与一个简单的覆盖论证相结合。

我们从单位球的一个覆盖性质开始。

LEMMA 23.11 *Let $\epsilon \in (0,1)$. There exists a finite set $Q \subset \mathbb{R}^d$ of size $|Q| \leq \left(\frac{3}{\epsilon}\right)^d$ such that*

$$\sup_{\mathbf{x}:\|\mathbf{x}\|\leq 1} \min_{\mathbf{v}\in Q} \|\mathbf{x}-\mathbf{v}\| \;\leq\; \epsilon.$$

Proof 设 k 为一个整数，并设

$$Q' = \{\mathbf{x} \in \mathbb{R}^d : \forall j \in [d], \exists i \in \{-k, -k+1, \ldots, k\} \text{ s.t. } x_j = \tfrac{i}{k}\}.$$

显然，$|Q'| = (2k+1)^d$。我们将设 $Q = Q' \cap B_2(1)$，其中 $B_2(1)$ 是 $\mathbb{R}^d$ 的单位 ℓ_2 球。由于 Q' 中的点均匀分布在单位 ℓ_∞ 球上，Q 的大小是 Q' 的大小乘以单位 ℓ_2 和 ℓ_∞ 球体积的比值。ℓ_∞ 球的体积是 2^d，而 $B_2(1)$ 的体积是

$$\frac{\pi^{d/2}}{\Gamma(1+d/2)}.$$

为了简单起见，假设 d 是偶数，因此

$$\Gamma(1+d/2) \;=\; (d/2)! \;\geq\; \left(\tfrac{d/2}{e}\right)^{d/2},$$

在最后一个不等式中我们使用了斯特林近似。总体上我们得到

$$|Q| \leq (2k+1)^d \left(\pi/e\right)^{d/2} (d/2)^{-d/2} 2^{-d}. \tag{23.10}$$

现在让我们指定 k。对于每个 $\mathbf{x} \in B_2(1)$，让 $\mathbf{v} \in Q$ 是一个向量，其 i 个元素是 $\text{sign}(x_i)\, \lfloor |x_i|\, k \rfloor / k$。然后，对于每个元素，我们有 $|x_i - v_i| \leq 1/k$，因此

$$\|\mathbf{x} - \mathbf{v}\| \leq \frac{\sqrt{d}}{k}.$$

为确保右侧最多为 ϵ，我们应设置 $k = \lceil \sqrt{d}/\epsilon \rceil$。将此值代入方程（23.10）中，我们得出结论

$$|Q| \leq (3\sqrt{d}/(2\epsilon))^d \left(\pi/e\right)^{d/2} (d/2)^{-d/2} = \left(\frac{3}{\epsilon}\sqrt{\frac{\pi}{2e}}\right)^d \leq \left(\frac{3}{\epsilon}\right)^d.$$

□

设 $\mathbf{x}$ 为一个可以表示为 $\mathbf{x} = U\boldsymbol{\alpha}$ 的向量，其中 U 是某个正交矩阵，$\|\boldsymbol{\alpha}\|_0 \leq s$。结合前面的覆盖性质和 JL 引理（引理23.4），我们可以证明一个随机的 W 不会扭曲任何这样的 $\mathbf{x}$。

LEMMA 23.12 *Let U be an orthonormal $d \times d$ matrix and let $I \subset [d]$ be a set of indices of size $|I| = s$. Let S be the span of $\{U_i : i \in I\}$, where U_i is the ith column of U. Let $\delta \in (0,1)$, $\epsilon \in (0,1)$, and $n \in \mathbb{N}$ such that*

$$n \geq 24 \frac{\log(2/\delta) + s\,\log(12/\epsilon)}{\epsilon^2}.$$

Then, with probability of at least $1-\delta$ over a choice of a random matrix $W \in \mathbb{R}^{n,d}$ such that each element of W is independently distributed according to $N(0, 1/n)$, we have

$$\sup_{\mathbf{x} \in S} \left| \frac{\|W\mathbf{x}\|}{\|\mathbf{x}\|} - 1 \right| < \epsilon.$$

Proof 只需证明对所有具有 $\|\mathbf{x}\| =$ 的 $\mathbf{x} \in S$ 的引理。我们可以写出 $\mathbf{x} = U_I\boldsymbol{\alpha}$，其中 $\boldsymbol{\alpha} \in \mathbb{R}^s$，$\|\boldsymbol{\alpha}\|_2 = 1$，并且 U_I 是列由 $\{U_i : i \in I\}$ 组成的矩阵。使用引理23.11，我们知道存在一个大小为 $|Q| \leq (12/\epsilon)^s$ 的集合 Q，使得

$$\sup_{\boldsymbol{\alpha}:\|\boldsymbol{\alpha}\|=1} \min_{\mathbf{v} \in Q} \|\boldsymbol{\alpha} - \mathbf{v}\| \leq (\epsilon/4).$$

但是，由于 U 是正交的，我们还有以下结果

$$\sup_{\boldsymbol{\alpha}:\|\boldsymbol{\alpha}\|=1} \min_{\mathbf{v} \in Q} \|U_I\boldsymbol{\alpha} - U_I\mathbf{v}\| \leq (\epsilon/4).$$

应用引理23.4于集合$\{U_I\mathbf{v} : \mathbf{v} \in Q\}$，我们得到对于满足$n$的

在引理中给出的条件，以下以至少$1-\delta$的概率成立：

$$\sup_{\mathbf{v}\in Q}\left|\frac{\|WU_I\mathbf{v}\|^2}{\|U_I\mathbf{v}\|^2}-1\right|\leq\epsilon/2,$$

这也意味着

$$\sup_{\mathbf{v}\in Q}\left|\frac{\|WU_I\mathbf{v}\|}{\|U_I\mathbf{v}\|}-1\right|\leq\epsilon/2.$$

设a为满足以下条件的最小数

$$\forall\mathbf{x}\in S,\quad\frac{\|W\mathbf{x}\|}{\|\mathbf{x}\|}\leq 1+a.$$

显然$a<\infty$。我们的目标是证明$a\leq\epsilon$。这源于以下事实：对于任何单位范数的$\mathbf{x}\in S$，存在$\mathbf{v}\in Q$使得$\|\mathbf{x}-U_I\mathbf{v}\|\leq\epsilon/4$，因此

$$\|W\mathbf{x}\|\leq\|WU_I\mathbf{v}\|+\|W(\mathbf{x}-U_I\mathbf{v})\|\leq 1+\epsilon/2+(1+a)\epsilon/4.$$

Thus,Thus,

$$\forall\mathbf{x}\in S,\quad\frac{\|W\mathbf{x}\|}{\|\mathbf{x}\|}\leq 1+\left(\epsilon/2+(1+a)\epsilon/4\right).$$

但是，a的定义意味着

$$a\leq\epsilon/2+(1+a)\epsilon/4\quad\Rightarrow\quad a\leq\frac{\epsilon/2+\epsilon/4}{1-\epsilon/4}\leq\epsilon.$$

这证明了对于所有$\mathbf{x}\in S$，我们都有$\frac{\|W\mathbf{x}\|}{\|\mathbf{x}\|}-1\leq\epsilon$。另一方面也由此得出，因为

$$\|W\mathbf{x}\|\geq\|WU_I\mathbf{v}\|-\|W(\mathbf{x}-U_I\mathbf{v})\|\geq 1-\epsilon/2-(1+\epsilon)\epsilon/4\geq 1-\epsilon.$$

□

前述引理告诉我们，对于单位范数的{v*}，我们有

$$(1-\epsilon)\leq\|W\mathbf{x}\|\leq(1+\epsilon),$$

这表示

$$(1-2\,\epsilon)\leq\|W\mathbf{x}\|^2\leq(1+3\,\epsilon).$$

证明定理23.9的证明通过对所有I的选择进行并集界得出。

23.4 PCA or Compressed Sensing?

假设我们想要将降维技术应用于一组给定的示例。我们应该使用PCA还是压缩感知？在本节中，我们通过强调两种方法背后的基本假设来解决这个问题。

首先，了解每种方法何时可以保证完美恢复是有帮助的。PCA只要示例集包含在 $\mathbb{R}^d$ 的 n 维子空间中，就保证完美恢复。压缩感知只要示例集在某些基下是稀疏的，就保证完美恢复。基于这些观察，我们可以描述PCA将优于压缩感知的情况，反之亦然。

作为一个第一个例子，假设例子是 $\mathbb{R}^d$ 的标准基向量，即 $\mathbf{e}_1, \ldots, \mathbf{e}_d$，其中每个 $\mathbf{e}_i$ 是除了 i 坐标上的 1 以外都是全零向量。在这种情况下，例子是 1-稀疏的。因此，当 $n \geq \Omega(\log(d))$ 时，压缩感知将产生完美的恢复。另一方面，PCA 将导致性能不佳，因为只要 $n < d$，数据就远非处于一个 n 维子空间中。事实上，很容易验证在这种情况下，PCA 的平均恢复误差（即方程（23.1）的目标除以 m）将是 $(d-n)/d$，当 $n \leq d/2$ 时，它大于 $1/2$。

接下来，我们展示一个PCA优于压缩感知的案例。考虑m个恰好位于一个n维子空间上的例子。显然，在这种情况下，PCA将导致完美恢复。至于压缩感知，请注意，这些例子在任何以n个向量张成该子空间为第一n个向量的正交归一基中都是n-稀疏的。因此，如果我们把维度降低到$\Omega(n \log(d))$, 压缩感知也会有效。然而，在恰好n维的情况下，压缩感知可能会失败。PCA对某些类型的噪声也有更好的鲁棒性。参见（Chang，Weiss & Freeman 2009）以了解讨论。

23.5 Summary

我们引入了两种使用线性变换进行降维的方法：PCA和随机投影。我们已经证明，如果我们限制重建过程也是线性的，那么在平均平方重建误差的意义上，PCA是最佳的。然而，如果我们允许非线性重建，PCA不一定是最优过程。特别是对于稀疏数据，随机投影可以显著优于PCA。这一事实是压缩感知方法的核心。

23.6 Bibliographic Remarks

PCA 等同于使用奇异值分解（SVD）进行最佳子空间逼近。SVD 方法在附录 C 中描述。SVD 可追溯至 Eugenio Beltrami（1873 年）和 Camille Jordan（1874 年）。它已被多次重新发现。在统计文献中，它由 Pearson（1901 年）引入。除了 PCA 和 SVD 之外，还有其他一些名称指代相同的概念，并被用于不同的科学社区。一些例子包括 Eckart-Young 定理（以 Carl Eckart 和 Gale Young 为名，他们在 1936 年分析了该方法）、Schmidt-Mirsky 定理、因子分析和 Hotelling 变换。

压缩感知在Donoho (2006) 和 (Candes & Tao 2005) 中被引入。参见Candes (2006)。

23.7 Exercises

1. 在这个练习中，我们表明在一般情况下，线性压缩方案的精确恢复是不可能的。

   1. 令 $A \in \mathbb{R}^{n,d}$ 为一个任意的压缩矩阵，其中 $n \leq d-1$. 证明存在 $\mathbf{u}, \mathbf{v} \in \mathbb{R}^n$，$\mathbf{u} \neq \mathbf{v}$ 使得 $A\mathbf{u} = A\mathbf{v}$。2. 结论是线性压缩方案的精确恢复是不可能的。

2. 设 $\boldsymbol{\alpha} \in \mathbb{R}^d$ 使得 $\alpha_1 \geq \alpha_2 \geq \cdots \geq \alpha_d \geq 0$。证明

$$\max_{\boldsymbol{\beta} \in [0,1]^d : \|\boldsymbol{\beta}\|_1 \leq n} \sum_{j=1}^{d} \alpha_j \beta_j \quad = \quad \sum_{j=1}^{n} \alpha_j .$$

   Hint: 取每个向量 $\boldsymbol{\beta} \in [0,1]^d$，使得 $\|\boldsymbol{\beta}\|_1 \leq n$。令 i 为满足 $\beta_i < 1$ 的最小索引。如果 $i = n+1$，则已完成。否则，证明我们可以增加 β_i，同时可能降低某些 $j > i$ 的 β_j，从而获得更好的解。这将意味着最优解是将 $\beta_i =$ 设置为 1 对于 $i \leq n$ 和 $\beta_i =$，将 $i > n$ 设置为 0。

3. **Kernel PCA:** 在这个练习中，我们展示了如何利用核技巧在PCA的基础上构建非线性降维（参见第16章）。

   设 $\mathcal{X}$ 为某个实例空间，设 $S = \{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$ 为 $\mathcal{X}$ 中的点集。考虑一个特征映射 $\psi : \mathcal{X} \to V$，其中 V 是某个希尔伯特空间（可能是无限维的）。设 $K : \mathcal{X} \times \mathcal{X}$ 为核函数，即 $k(\mathbf{x}, \mathbf{x}') = \langle \psi(\mathbf{x}), \psi(\mathbf{x}') \rangle$。核PCA是将 S 中的元素映射到 V 的过程，使用 ψ，然后对 $\{\psi(\mathbf{x}_1), \ldots, \psi(\mathbf{x}_m)\}$ 应用PCA到 $\mathbb{R}^n$。此过程的输出是降维后的元素集。

   展示如何以多项式时间完成此过程，就 m 和 n 而言，假设每次对 $K(\cdot, \cdot)$ 的评估都可以在常数时间内完成。特别是，如果您的实现需要乘以两个矩阵 A 和 B，请验证它们的乘积可以计算。同样，

如果需要某个矩阵 C 的特征值分解，请验证这种分解可以计算。

4. {v²}

设 $\mathbf{x}_1, \ldots, \mathbf{x}_m$ 是 m 个在 $\mathbb{R}^d$ 中的向量，设 $\mathbf{x}$ 是一个在 $\mathbf{x}_1, \ldots, \mathbf{x}_m$ 上均匀分布的随机向量。假设 $\mathbb{E}[\mathbf{x}] = \mathbf{0}$。1. 考虑寻找一个单位向量 $\mathbf{w} \in \mathbb{R}^d$，使得随机变量 $\langle \mathbf{w}, \mathbf{x} \rangle$ 具有最大方差的问题。也就是说，我们希望解决以下问题

$$\operatorname*{argmax}_{\mathbf{w}:\|\mathbf{w}\|=1} \operatorname{Var}[\langle \mathbf{w}, \mathbf{x} \rangle] = \operatorname*{argmax}_{\mathbf{w}:\|\mathbf{w}\|=1} \frac{1}{m} \sum_{i=1}^{m} (\langle \mathbf{w}, \mathbf{x}_i \rangle)^2.$$

证明该问题的解是将 $\mathbf{w}$ 设置为 $\mathbf{x}_1, \ldots, \mathbf{x}_m$ 的第一个原向量。

2. 令 $\mathbf{w}_1$ 为前一个问题中的第一个主成分。现在，假设我们想要找到一个第二个单位向量 $\mathbf{w}_2 \in \mathbb{R}^d$，它最大化 $\langle \mathbf{w}_2, \mathbf{x} \rangle$ 的方差，但与 $\langle \mathbf{w}_1, \mathbf{x} \rangle$ 不相关。也就是说，我们想要解决:

$$\operatorname*{argmax}_{\mathbf{w}:\|\mathbf{w}\|=1,\ \mathbb{E}[(\langle \mathbf{w}_1, \mathbf{x} \rangle)(\langle \mathbf{w}, \mathbf{x} \rangle)]=0} \operatorname{Var}[\langle \mathbf{w}, \mathbf{x} \rangle].$$

证明此问题的解是将 $\mathbf{w}$ 设置为 $\mathbf{x}_1, \ldots, \mathbf{x}_m$ 的第二个主成分。

Hint: 请注意

$$\mathbb{E}[(\langle \mathbf{w}_1, \mathbf{x} \rangle)(\langle \mathbf{w}, \mathbf{x} \rangle)] = \mathbf{w}_1^\top \mathbb{E}[\mathbf{x}\mathbf{x}^\top]\mathbf{w} = m\mathbf{w}_1^\top A \mathbf{w},$$

在 $A = \sum_i \mathbf{x}_i \mathbf{x}_i^\top$. 由于 $\mathbf{w}$ 是 A 的特征向量，因此约束 $\mathbb{E}[(\langle \mathbf{w}_1, \mathbf{x} \rangle)(\langle \mathbf{w}, \mathbf{x} \rangle)] = = 0$ 等价于约束

$$\langle \mathbf{w}_1, \mathbf{w} \rangle = 0.$$

5. **The Relation between SVD and PCA:** 使用SVD定理（推论C.6）为定理23.2提供另一种证明方法。

6. **Random Projections Preserve Inner Products:** Johnson-Lindenstrauss引理告诉我们，随机投影可以保持有限向量集之间的距离。在这个练习中，你需要证明如果向量集在单位球内，那么不仅任何两个向量之间的距离被保持，而且内积也被保持。

设 Q 是 $\mathbb{R}^d$ 中的一组有限向量，并假设对于每个 $\mathbf{x} \in Q$，我们有 $\|\mathbf{x}\| \leq 1$。

1. 设 $\delta \in (0, 1)$ 和 n 为一个整数，使得

$$\epsilon = \sqrt{\frac{6 \log(|Q|^2/\delta)}{n}} \leq 3.$$

证明在至少$1 - \delta$ 的概率下，从随机选择

矩阵 $W \in \mathbb{R}^{n,d}$，其中 W 的每个元素独立地按照 $\mathcal{N}(0, 1/n)$ 分布，我们有

$$|\langle W\mathbf{u}, W\mathbf{v}\rangle - \langle \mathbf{u}, \mathbf{v}\rangle| \leq \epsilon$$

对于每个 $\mathbf{u}, \mathbf{v} \in Q$。

*Hint*使用 JL 来限制 $\frac{\|W(\mathbf{u}+\mathbf{v})\|}{\|\mathbf{u}+\mathbf{v}\|}$ 和 $\frac{\|W(\mathbf{u}-\mathbf{v})\|}{\|\mathbf{u}-\mathbf{v}\|}$。

2. (*) 设 $\mathbf{x}_1, \ldots, \mathbf{x}_m$ 是 $\mathbb{R}^d$ 中范数不超过 1 的向量集，并假设这些向量以 γ 为间隔线性可分。假设 $d \gg 1/\gamma^2$。证明存在一个常数 $c > 0$，使得如果我们随机将这些向量投影到 $\mathbb{R}^n$，对于 $n = c/\gamma^2$，那么至少有 99% 的概率，投影向量以 $\gamma/2$ 为间隔线性可分。

24 Generative Models

我们这本书以一个 *distribution free* 学习框架开始；也就是说，我们没有对数据的基本分布做出任何假设。此外，我们遵循了一个 *discriminative* 方法，我们的目标不是学习基本分布，而是学习一个准确的预测器。在本章中，我们描述了一个 *generative* 方法，其中假设数据的基本分布具有特定的参数形式，我们的目标是估计模型的参数。这个任务被称为 *parametric density estimation*。

判别方法的优势在于直接优化感兴趣的数量（预测精度），而不是学习潜在分布。这由Vladimir Vapnik在他的使用有限信息解决问题的原则中如此表述：

When solving a given problem, try to avoid a more general problem as an intermediate step.

当然，如果我们能够准确学习到潜在分布，我们就可以被认为是“专家”，因为我们可以使用贝叶斯最优分类器进行预测。问题是，通常学习潜在分布比学习一个准确的预测器更困难。然而，在某些情况下，采用生成学习方法是合理的。例如，有时（在计算上）估计模型的参数比学习一个判别预测器更容易。此外，在某些情况下，我们并没有具体任务在手，而是希望对数据进行建模，要么是为了在以后进行预测而不必重新训练预测器，要么是为了数据的可解释性。

我们从一种流行的用于估计数据参数的统计方法开始，这种方法被称为最大似然原理。接下来，我们描述了两个简化学习过程的生成假设。我们还描述了在有潜在变量的情况下计算最大似然度的EM算法。最后，我们对贝叶斯推理进行了简要描述。

24.1 Maximum Likelihood Estimator

让我们从一个简单的例子开始。一家制药公司开发了一种新药来治疗一些致命的疾病。我们希望估计使用该药物时的生存概率。为此，制药公司抽取了一个包含m人的训练集，并给他们提供了药物。令$S = (x_1, \ldots, x_m)$表示训练集，其中对于每个i，如果第i个人存活则为$x_i = 1$，否则为$x_i = 0$。我们可以使用单个参数$\theta \in [0, 1]$来表示潜在的分布，它表示生存的概率。

我们现在想根据训练集 S 估计参数 θ。一个自然的想法是使用 S 中 1 的平均数量作为估计值。即，

$$\hat{\theta} = \frac{1}{m} \sum_{i=1}^{m} x_i. \tag{24.1}$$

显然，$\mathbb{E}_S[\hat{\theta}] = \theta$。也就是说，$\hat{\theta}$是$\theta$的一个*unbiased estimator*。此外，由于$\hat{\theta}$是m独立同分布的二进制随机变量的平均值，我们可以使用Hoeffding不等式得到，在至少$1 - \delta$的概率下，对于S的选择，我们有

$$|\hat{\theta} - \theta| \ \leq \ \sqrt{\frac{\log(2/\delta)}{2\,m}}. \tag{24.2}$$

另一种对 $\hat{\theta}$ 的解释是作为 *Maximum Likelihood Estimator*，我们现在正式解释。我们首先写出生成样本 S 的概率：

$$\mathbb{P}[S = (x_1, \ldots, x_m)] = \prod_{i=1}^{m} \theta^{x_i} \, (1-\theta)^{1-x_i} = \theta^{\sum_i x_i} \, (1-\theta)^{\sum_i (1-x_i)}.$$

我们定义 *log likelihood* 为 S，给定参数 θ，作为前一个表达式的对数：

$$L(S; \theta) = \log \left(\mathbb{P}[S = (x_1, \ldots, x_m)] \right) = \log(\theta) \sum_i x_i + \log(1-\theta) \sum_i (1 - x_i).$$

最大似然估计是使似然性最大化的参数

$$\hat{\theta} \in \operatorname*{argmax}_{\theta} L(S; \theta). \tag{24.3}$$

接下来，我们证明在我们的情况下，方程（24.1）是一个最大似然估计量。为了证明这一点，我们取 $L(S;\ \theta)$ 关于 θ 的导数，并将其等于零：

$$\frac{\sum_i x_i}{\theta} - \frac{\sum_i (1 - x_i)}{1 - \theta} = 0.$$

求解方程 θ，我们得到方程（24.1）中给出的估计量。

24.1.1 Maximum Likelihood Estimation for Continuous Random Variables

设 X 为一个连续随机变量。那么，对于大多数 $x \in \mathbb{R}$，我们有 $\mathbb{P}[X = x] = 0$，因此之前给出的似然定义变得平凡。为了克服这个技术问题，我们将似然定义为概率 *density* 在 X 处的对数。也就是说，给定一个按密度分布 $\mathcal{P}_\theta$ 抽样的独立同分布训练集 $S = (x_1, \ldots, x_m)$，我们定义给定 θ 的 S 的似然为

$$L(S;\theta) = \log\left(\prod_{i=1}^{m} \mathcal{P}_\theta(x_i)\right) = \sum_{i=1}^{m} \log(\mathcal{P}_\theta(x_i)).$$

与之前一样，最大似然估计量是 $L(S$ 的最大化者；关于 θ 对 $\theta)$。

例如，考虑一个高斯随机变量，其中 X 的密度函数由 $\theta = (\mu, \sigma)$ 参数化，并定义为以下：

$$\mathcal{P}_\theta(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right).$$

我们可以将似然重写为

$$L(S;\theta) = -\frac{1}{2\sigma^2}\sum_{i=1}^{m}(x_i - \mu)^2 - m\log(\sigma\sqrt{2\pi}).$$

要找到一个参数 $\theta = (\mu, \sigma)$ 来优化这一点，我们取似然函数对 μ 和 σ 的导数，并将其与 0 比较。我们得到以下两个方程：

$$\frac{d}{d\mu}L(S;\theta) = \frac{1}{\sigma^2}\sum_{i=1}^{m}(x_i - \mu) = 0$$
$$\frac{d}{d\sigma}L(S;\theta) = \frac{1}{\sigma^3}\sum_{i=1}^{m}(x_i - \mu)^2 - \frac{m}{\sigma} = 0$$

求解前面的方程，我们得到最大似然估计：

$$\hat{\mu} = \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \text{and} \qquad \hat{\sigma} = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(x_i - \hat{\mu})^2}$$

注意，最大似然估计不总是无偏估计。例如，虽然 $\hat{\mu}$ 是无偏的，但可以证明方差的估计 $\hat{\sigma}$ 是有偏的（练习1）。

Simplifying Notation
为了简化我们的符号，我们在这章中使用 $\mathcal{P}[X = x]$ 来描述离散随机变量 $X = x$ (的概率以及连续变量 x (的分布密度。

24.1.2 Maximum Likelihood and Empirical Risk Minimization

最大似然估计器与经验风险最小化（ERM）原则有一些相似之处，我们在前几章中对其进行了深入研究。回想一下，在ERM原则中，我们有一个假设类 $\mathcal{H}$，我们使用训练集来选择一个假设 $h \in \mathcal{H}$ 以最小化经验风险。我们现在表明，最大似然估计器是针对特定损失函数的ERM。

给定一个参数 θ 和一个观测 x，我们定义 θ 在 x 上的损失为

$$\ell(\theta, x) = -\log(\mathcal{P}_\theta[x]). \tag{24.4}$$

这是，$\ell(\theta, x)$ 是观测 x 的对数似然的对立，假设数据按照 $\mathcal{P}_\theta$ 分布。这个损失函数通常被称为对数损失。基于这个定义，可以立即得出最大似然原理等价于最小化方程（24.4）中给出的损失函数的实证风险。即，

$$\operatorname*{argmin}_\theta \sum_{i=1}^{m} (-\log(\mathcal{P}_\theta[x_i])) = \operatorname*{argmax}_\theta \sum_{i=1}^{m} \log(\mathcal{P}_\theta[x_i]).$$

假设数据服从分布 $\mathcal{P}$ (，该分布不一定是我们所采用的参数形式)，则参数 θ 的真实风险变为

$$\begin{aligned} \mathop{\mathbb{E}}_x[\ell(\theta, x)] &= -\sum_x \mathcal{P}[x] \log(\mathcal{P}_\theta[x]) \\ &= \underbrace{\sum_x \mathcal{P}[x] \log\left(\frac{\mathcal{P}[x]}{\mathcal{P}_\theta[x]}\right)}_{D_{\mathrm{RE}}[\mathcal{P}||\mathcal{P}_\theta]} + \underbrace{\sum_x \mathcal{P}[x] \log\left(\frac{1}{\mathcal{P}[x]}\right)}_{H(\mathcal{P})}, \end{aligned} \tag{24.5}$$

在 D_{RE} 被称为 *relative entropy*，而 H 被称为 *entropy func- tion* 的地方。相对熵是两个概率之间的散度度量。对于离散变量，它总是非负的，并且只有在两个分布相同时才等于 0。因此，当 $\mathcal{P}_\theta = \mathcal{P}$ 时，真实风险是最小的。

表达式（24.5）所给出的内容强调了我们的生成假设如何影响我们的密度估计，即使在无限数据的极限情况下也是如此。它表明，如果底层分布确实是一种参数形式，那么通过选择正确的参数，我们可以使风险成为分布的熵。然而，如果分布不是假设的参数形式，即使是最好的参数也会导致一个较差的模型，而次优性是通过相对熵发散来衡量的。

24.1.3 Generalization Analysis

最大似然估计在从有限训练集中学习时有多好?

为了回答这个问题，我们需要定义我们如何评估密度估计问题的近似解的质量。与有明确“损失”概念的区别性学习不同，在生成学习中，有各种定义模型损失的方法。基于前一小节，一个自然的候选者是方程（24.5）中给出的期望对数损失。

在某些情况下，很容易证明最大似然原理也保证了低真实风险。例如，考虑估计单位方差高斯变量的均值的问题。我们之前看到，最大似然估计量是平均值：$\hat{\mu} = \frac{1}{m}\sum_i x_i$。设$\mu^\star$为最优参数。那么，

$$\begin{aligned}
\mathop{\mathbb{E}}_{x\sim N(\mu^\star,1)}[\ell(\hat{\mu},x) - \ell(\mu^\star,x)] &= \mathop{\mathbb{E}}_{x\sim N(\mu^\star,1)} \log\left(\frac{\mathcal{P}_{\mu^\star}[x]}{\mathcal{P}_{\hat{\mu}}[x]}\right) \\
&= \mathop{\mathbb{E}}_{x\sim N(\mu^\star,1)} \left(-\frac{1}{2}(x-\mu^\star)^2 + \frac{1}{2}(x-\hat{\mu})^2\right) \\
&= \frac{\hat{\mu}^2}{2} - \frac{(\mu^\star)^2}{2} + (\mu^\star - \hat{\mu}) \mathop{\mathbb{E}}_{x\sim N(\mu^\star,1)}[x] \\
&= \frac{\hat{\mu}^2}{2} - \frac{(\mu^\star)^2}{2} + (\mu^\star - \hat{\mu})\,\mu^\star \\
&= \frac{1}{2}(\hat{\mu} - \mu^\star)^2. \qquad (24.6)
\end{aligned}$$

接下来，我们注意到$\hat{\mu}$是m个高斯变量的平均值，因此它也是正态分布的，均值为$\mu^\star$，方差为$\sigma^\star/m$。从这个事实中，我们可以推导出以下形式的界限：至少以$-\delta$的概率，我们有$|\hat{\mu} - \mu^\star| \le \epsilon$，其中$\epsilon$依赖于$\sigma^\star/m$和$\delta$。

在某些情况下，最大似然估计明显过拟合。例如，考虑一个伯努利随机变量X并令$\mathcal{P}[X = 1] = \theta^\star$。正如我们之前所看到的，使用Hoeffding不等式，我们可以轻松地推导出一个关于$|\theta^\star - \hat{\theta}|$的保证，该保证以高概率成立（参见方程（24.2））。然而，如果我们的目标是获得方程（24.5）中定义的期望对数损失函数的小值，我们可能会失败。例如，假设$\theta^\star$非零但非常小。那么，样本大小为m的样本中没有任何元素为1的概率是$(1-\theta^\star)^m$，这大于$e^{-2\theta^\star m}$。因此，当$m \le \frac{\log(2)}{2\theta^\star}$时，样本全为零的概率至少为50%，在这种情况下，最大似然法则将$\hat{\theta} =$设置为0。但估计$\hat{\theta} = 0$的真实风险是

$$\begin{aligned}
\mathop{\mathbb{E}}_{x\sim\theta^\star}[\ell(\hat{\theta},x)] &= \theta^\star \ell(\hat{\theta},1) + (1-\theta^\star)\ell(\hat{\theta},0) \\
&= \theta^\star \log(1/\hat{\theta}) + (1-\theta^\star)\log(1/(1-\hat{\theta})) \\
&= \theta^\star \log(1/0) = \infty.
\end{aligned}$$

这个简单示例表明，我们在应用最大似然原理时应该小心。

为了克服过拟合，我们可以使用我们之前遇到的多种工具。

之前在书中。在练习2中概述了一种简单的正则化技术。

24.2 Naive Bayes

朴素贝叶斯分类器是生成假设和参数估计如何简化学习过程的经典示例。考虑基于特征向量 $\mathbf{x} = (x_1, \ldots, x_d)$ 预测标签 $y \in \{0, 1\}$ 的问题，其中我们假设每个 x_i 都在 $\{0, 1\}$ 中。回忆一下，贝叶斯最优分类器是

$$h_{\text{Bayes}}(\mathbf{x}) = \operatorname*{argmax}_{y \in \{0,1\}} \mathcal{P}[Y = y | X = \mathbf{x}].$$

要描述概率函数 $\mathcal{P}[Y = y | X = \mathbf{x}]$，我们需要 2^d 个参数，每个参数对应于 $\mathcal{P}[Y = 1 | X = \mathbf{x}]$ 对于 $\mathbf{x} \in \{0, 1\}^d$ 的某个值。这意味着我们需要的示例数量会随着特征数量的指数增长。

在朴素贝叶斯方法中，我们做出了（相当朴素）的生成假设：给定标签，特征之间相互独立。也就是说，

$$\mathcal{P}[X = \mathbf{x} | Y = y] = \prod_{i=1}^{d} \mathcal{P}[X_i = x_i | Y = y].$$

基于这个假设，使用贝叶斯定理，贝叶斯最优分类器可以进一步简化：

$$\begin{aligned} h_{\text{Bayes}}(\mathbf{x}) &= \operatorname*{argmax}_{y \in \{0,1\}} \mathcal{P}[Y = y | X = \mathbf{x}] \\ &= \operatorname*{argmax}_{y \in \{0,1\}} \mathcal{P}[Y = y] \mathcal{P}[X = \mathbf{x} | Y = y] \\ &= \operatorname*{argmax}_{y \in \{0,1\}} \mathcal{P}[Y = y] \prod_{i=1}^{d} \mathcal{P}[X_i = x_i | Y = y]. \end{aligned} \tag{24.7}$$

这意味着现在我们需要估计的参数数量仅为$2d + 1$。在这里，我们做出的生成假设显著减少了我们需要学习的参数数量。

当我们也使用最大似然原理估计参数时，得到的分类器被称为 *Naive Bayes* 分类器。

24.3 Linear Discriminant Analysis

线性判别分析（LDA）是生成假设如何简化学习过程的另一个示例。正如在朴素贝叶斯分类器中，我们再次考虑基于以下问题预测标签 $y \in \{0, 1\}$ 的问题

特征向量 {v*}。但现在生成假设如下。首先，我们假设 $\mathcal{P}[Y=1]=\mathcal{P}[Y=0]=1/2$。其次，我们假设给定 Y 的 X 的条件概率是高斯分布。最后，高斯分布的协方差矩阵对于标签的两个值都是相同的。形式上，设 $\boldsymbol{\mu}_0,\boldsymbol{\mu}_1\in\mathbb{R}^d$ 和设 Σ 为协方差矩阵。那么，密度分布由以下给出

$$\mathcal{P}[X=\mathbf{x}|Y=y]=\frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}}\exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_y)^T\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu}_y)\right).$$

正如我们在上一节中所示，使用贝叶斯定理，我们可以写出

$$h_{\text{Bayes}}(\mathbf{x})=\operatorname*{argmax}_{y\in\{0,1\}}\mathcal{P}[Y=y]\mathcal{P}[X=\mathbf{x}|Y=y].$$

这意味着我们将预测 $h_{\text{Bayes}}(\mathbf{x})=1$ 当且仅当

$$\log\left(\frac{\mathcal{P}[Y=1]\mathcal{P}[X=\mathbf{x}|Y=1]}{\mathcal{P}[Y=0]\mathcal{P}[X=\mathbf{x}|Y=0]}\right)>0.$$

这个比率通常被称为*log-likelihood ratio*。

在我们的情况下，对数似然比变为

$$\tfrac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_0)^T\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu}_0)-\tfrac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_1)^T\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu}_1)$$

我们可以将其重写为 $\langle\mathbf{w},\mathbf{x}\rangle+b$，其中

$$\mathbf{w}=(\boldsymbol{\mu}_1-\boldsymbol{\mu}_0)^T\Sigma^{-1}\quad\text{and}\quad b=\tfrac{1}{2}\left(\boldsymbol{\mu}_0^T\Sigma^{-1}\boldsymbol{\mu}_0-\boldsymbol{\mu}_1^T\Sigma^{-1}\boldsymbol{\mu}_1\right). \tag{24.8}$$

由于前面的推导，我们得到，在上述生成假设下，贝叶斯最优分类器是一个线性分类器。此外，可以通过从数据中估计参数 $\boldsymbol{\mu}_0,\boldsymbol{\mu}_1$ 和 Σ 来训练分类器，例如使用最大似然估计器。有了这些估计器，可以根据公式（24.8）计算出 $\mathbf{w}$ 和 b 的值。

24.4 Latent Variables and the EM Algorithm

在生成模型中，我们假设数据是通过从我们的实例空间 $\mathcal{X}$ 上的特定参数分布中进行采样而生成的。有时，使用潜在随机变量来表示这个分布是方便的。一个自然的例子是 k 个高斯分布的混合。也就是说，$\mathcal{X}=\mathbb{R}^d$，我们假设每个 $\mathbf{x}$ 都如下生成。首先，我们在 $\{1,\ldots,k\}$ 中选择一个随机数。让 Y 是对应这个选择的随机变量，并表示为 $\mathcal{P}[Y=y]=c_y$。其次，根据 Y 的值，我们根据高斯分布选择 $\mathbf{x}$。

$$\mathcal{P}[X=\mathbf{x}|Y=y]=\frac{1}{(2\pi)^{d/2}|\Sigma_y|^{1/2}}\exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_y)^T\Sigma_y^{-1}(\mathbf{x}-\boldsymbol{\mu}_y)\right). \tag{24.9}$$

因此，X的密度可以表示为：

$$\begin{aligned}\mathcal{P}[X=\mathbf{x}] &= \sum_{y=1}^{k}\mathcal{P}[Y=y]\mathcal{P}[X=\mathbf{x}|Y=y] \\ &= \sum_{y=1}^{k} c_y \frac{1}{(2\pi)^{d/2}|\Sigma_y|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_y)^T\Sigma_y^{-1}(\mathbf{x}-\boldsymbol{\mu}_y)\right).\end{aligned}$$

注意，Y是一个我们数据中未观察到的隐藏变量。尽管如此，我们引入 Y，因为它有助于我们描述 X 的概率的简单参数形式。

更一般地，设$\boldsymbol{\theta}$为X和Y的联合分布的参数(，例如，在前面的例子中，$\boldsymbol{\theta}$包括c_y、$\boldsymbol{\mu}_y$和Σ_y，对于所有$y=1,\ldots,k$)。那么，一个观测值$\mathbf{x}$的对数似然可以表示为

$$\log\left(\mathcal{P}_{\boldsymbol{\theta}}[X=\mathbf{x}]\right) = \log\left(\sum_{y=1}^{k}\mathcal{P}_{\boldsymbol{\theta}}[X=\mathbf{x},Y=y]\right).$$

给定一个独立同分布的样本 $S=(\mathbf{x}_1,\ldots,\mathbf{x}_m)$，我们希望找到 $\boldsymbol{\theta}$ 以最大化 S 的对数似然函数。

$$\begin{aligned}L(\boldsymbol{\theta}) &= \log\prod_{i=1}^{m}\mathcal{P}_{\boldsymbol{\theta}}[X=\mathbf{x}_i] \\ &= \sum_{i=1}^{m}\log\mathcal{P}_{\boldsymbol{\theta}}[X=\mathbf{x}_i] \\ &= \sum_{i=1}^{m}\log\left(\sum_{y=1}^{k}\mathcal{P}_{\boldsymbol{\theta}}[X=\mathbf{x}_i,Y=y]\right).\end{aligned}$$

最大似然估计量因此是最大化问题的解

$$\operatorname*{argmax}_{\boldsymbol{\theta}} L(\boldsymbol{\theta}) = \operatorname*{argmax}_{\boldsymbol{\theta}} \sum_{i=1}^{m}\log\left(\sum_{y=1}^{k}\mathcal{P}_{\boldsymbol{\theta}}[X=\mathbf{x}_i,Y=y]\right).$$

在许多情况下，对数内的求和使得前面的优化问题在计算上变得困难。Dempster、Laird和Rubin提出的*Expectation-Maximization*（EM)算法是用于搜索$L(\boldsymbol{\theta})$（局部）最大值的迭代过程。虽然EM算法不能保证找到全局最大值，但在实践中通常表现相当好。

EM是为那些情况设计的，如果我们知道潜在变量Y的值，那么最大似然优化问题将是可解决的。更精确地说，在$m\times k$矩阵和参数集$\boldsymbol{\theta}$上定义以下函数：

$$F(Q,\boldsymbol{\theta}) = \sum_{i=1}^{m}\sum_{y=1}^{k} Q_{i,y}\log\left(\mathcal{P}_{\boldsymbol{\theta}}[X=\mathbf{x}_i,Y=y]\right).$$

如果 Q 的每一行定义了在 $X = \mathbf{x}_i$ 的条件下关于 ith 隐含变量的概率，那么我们可以将 $F(Q, \boldsymbol{\theta})$ 解释为训练集 $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$ 的期望对数似然，这里的期望是针对 y_i 的选择，该选择基于 Q 的第 i 行。在 F 的定义中，求和是在对数之外进行的，我们假设这使关于 $\boldsymbol{\theta}$ 的优化问题变得可处理：

ASSUMPTION 24.1 对于任何矩阵 $Q \in [0, 1]^{m,k}$，使得 Q 的每一行之和为 1，优化问题

$$\operatorname*{argmax}_{\boldsymbol{\theta}} F(Q, \boldsymbol{\theta})$$

是可处理的。

EM的直观想法是我们有一个“鸡生蛋还是蛋生鸡”的问题。一方面，如果我们已知Q，那么根据我们的假设，找到最佳$\boldsymbol{\theta}$的优化问题是可解的。另一方面，如果我们已知参数$\boldsymbol{\theta}$，我们就可以将$Q_{i,y}$设置为在$X = \mathbf{x}_i$的条件下$Y = y$的概率。因此，EM算法在找到给定Q的$\boldsymbol{\theta}$和给定$\boldsymbol{\theta}$的Q之间交替。形式上，EM找到一系列解$(Q^{(1)}, \boldsymbol{\theta}^{(1)}), (Q^{(2)}, \boldsymbol{\theta}^{(2)}), \ldots$，其中在迭代$t$时，我们通过执行两个步骤来构建$(Q^{(t+1)}, \boldsymbol{\theta}^{(t+1)})$。

- **E**期望步骤：设置

$$Q_{i,y}^{(t+1)} = \mathcal{P}_{\boldsymbol{\theta}^{(t)}}[Y = y | X = \mathbf{x}_i]. \tag{24.10}$$

  这一步被称为期望步，因为它在潜在变量上产生一个新的概率，这定义了一个新的 *expected* 对 $\boldsymbol{\theta}$ 的对数似然函数。

- **M**优化步骤：将 $\boldsymbol{\theta}^{(t+1)}$ 设置为期望对数似然的最大值，其中期望值根据 $Q^{(t+1)}$ 计算：

$$\boldsymbol{\theta}^{(t+1)} = \operatorname*{argmax}_{\boldsymbol{\theta}} F(Q^{(t+1)}, \boldsymbol{\theta}). \tag{24.11}$$

  根据我们的假设，可以有效地解决这个问题优化问题。

初始值 $\boldsymbol{\theta}^{(1)}$ 和 $Q^{(1)}$ 通常随机选择，并在似然值改善不再显著后终止程序。

24.4.1 EM as an Alternate Maximization Algorithm

为了分析EM算法，我们首先将其视为一种交替最大化算法。定义以下目标函数

$$G(Q, \boldsymbol{\theta}) = F(Q, \boldsymbol{\theta}) - \sum_{i=1}^{m} \sum_{y=1}^{k} Q_{i,y} \log(Q_{i,y}).$$

第二项是Q的行*entropies*的和。令

$$\mathbb{Q} = \left\{ Q \in [0,1]^{m,k} : \forall i,\ \sum_{y=1}^{k} Q_{i,y} = 1 \right\}$$

矩阵的行定义了在 $[k]$ 上的概率的集合。以下引理表明，EM算法执行交替最大化迭代以最大化 G。

LEMMA 24.2 *The EM procedure can be rewritten as:*

$$\begin{aligned} Q^{(t+1)} &= \operatorname*{argmax}_{Q \in \mathbb{Q}} G(Q, \boldsymbol{\theta}^{(t)}) \\ \boldsymbol{\theta}^{(t+1)} &= \operatorname*{argmax}_{\boldsymbol{\theta}} G(Q^{(t+1)}, \boldsymbol{\theta}) \ . \end{aligned}$$

Furthermore, $G(Q^{(t+1)}, \boldsymbol{\theta}^{(t)}) = L(\boldsymbol{\theta}^{(t)})$.

Proof 给定 $Q^{(t+1)}$，我们显然有

$$\operatorname*{argmax}_{\boldsymbol{\theta}} G(Q^{(t+1)}, \boldsymbol{\theta}) = \operatorname*{argmax}_{\boldsymbol{\theta}} F(Q^{(t+1)}, \boldsymbol{\theta}).$$

因此，我们只需要证明对于任意的 $\boldsymbol{\theta}$，$\operatorname{argmax}_{Q \in \mathbb{Q}} G(Q, \boldsymbol{\theta})$ 的解是将 $Q_{i,y} = \mathcal{P}_{\boldsymbol{\theta}}[Y = y | X = \mathbf{x}_i]$ 设置为。事实上，根据 Jensen 不等式，对于任意的 $Q \in \mathbb{Q}$，我们有

$$\begin{aligned} G(Q, \boldsymbol{\theta}) &= \sum_{i=1}^{m} \left(\sum_{y=1}^{k} Q_{i,y} \log \left(\frac{\mathcal{P}_{\boldsymbol{\theta}}[X = \mathbf{x}_i, Y = y]}{Q_{i,y}} \right) \right) \\ &\leq \sum_{i=1}^{m} \left(\log \left(\sum_{y=1}^{k} Q_{i,y} \frac{\mathcal{P}_{\boldsymbol{\theta}}[X = \mathbf{x}_i, Y = y]}{Q_{i,y}} \right) \right) \\ &= \sum_{i=1}^{m} \log \left(\sum_{y=1}^{k} \mathcal{P}_{\boldsymbol{\theta}}[X = \mathbf{x}_i, Y = y] \right) \\ &= \sum_{i=1}^{m} \log \left(\mathcal{P}_{\boldsymbol{\theta}}[X = \mathbf{x}_i] \right) = L(\boldsymbol{\theta}), \end{aligned}$$

当对于 $Q_{i,y} = \mathcal{P}_{\boldsymbol{\theta}}[Y = y|X = \mathbf{x}_i]$ 我们有

$$\begin{aligned} G(Q, \boldsymbol{\theta}) &= \sum_{i=1}^{m} \left(\sum_{y=1}^{k} \mathcal{P}_{\boldsymbol{\theta}}[Y = y|X = \mathbf{x}_i] \log \left(\frac{\mathcal{P}_{\boldsymbol{\theta}}[X = \mathbf{x}_i, Y = y]}{\mathcal{P}_{\boldsymbol{\theta}}[Y = y|X = \mathbf{x}_i]} \right) \right) \\ &= \sum_{i=1}^{m} \sum_{y=1}^{k} \mathcal{P}_{\boldsymbol{\theta}}[Y = y|X = \mathbf{x}_i] \log \left(\mathcal{P}_{\boldsymbol{\theta}}[X = \mathbf{x}_i] \right) \\ &= \sum_{i=1}^{m} \log \left(\mathcal{P}_{\boldsymbol{\theta}}[X = \mathbf{x}_i] \right) \sum_{y=1}^{k} \mathcal{P}_{\boldsymbol{\theta}}[Y = y|X = \mathbf{x}_i] \\ &= \sum_{i=1}^{m} \log \left(\mathcal{P}_{\boldsymbol{\theta}}[X = \mathbf{x}_i] \right) = L(\boldsymbol{\theta}). \end{aligned}$$

这表明设置 $Q_{i,y} = \mathcal{P}_{\boldsymbol{\theta}}[Y = y|X = \mathbf{x}_i]$ 在 $Q \in \mathbb{Q}$ 上最大化 $G(Q, \boldsymbol{\theta})$ 并表明 $G(Q^{(t+1)}, \boldsymbol{\theta}^{(t)}) = L(\boldsymbol{\theta}^{(t)})$。 □

前述引理立即推出：

THEOREM 24.3 *The EM procedure never decreases the log-likelihood; namely, for all t,*

$$L(\boldsymbol{\theta}^{(t+1)}) \geq L(\boldsymbol{\theta}^{(t)}).$$

Proof 通过引理，我们有

$$L(\boldsymbol{\theta}^{(t+1)}) = G(Q^{(t+2)}, \boldsymbol{\theta}^{(t+1)}) \geq G(Q^{(t+1)}, \boldsymbol{\theta}^{(t)}) = L(\boldsymbol{\theta}^{(t)}).$$

□

24.4.2 EM for Mixture of Gaussians (Soft k-Means)

考虑一个由 k 个高斯分布组成的混合情况，其中 $\boldsymbol{\theta}$ 是一个三元组 $(\mathbf{c}, \{\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k\}, \{\Sigma_1, \ldots, \Sigma_k\})$，且 $\mathcal{P}_{\boldsymbol{\theta}}[Y = y] = c_y$ 和 $\mathcal{P}_{\boldsymbol{\theta}}[X = \mathbf{x}|Y = y]$ 如方程（24.9）所示。为了简化，我们假设 $\Sigma_1 = \Sigma_2 = \cdots = \Sigma_k = I$，其中 I 是单位矩阵。为这种情况指定 EM 算法，我们得到以下：

- **E**期望步骤：对于每个 $i \in [m]$ 和 $y \in [k]$，我们有

$$\begin{aligned} \mathcal{P}_{\boldsymbol{\theta}^{(t)}}[Y = y|X = \mathbf{x}_i] &= \frac{1}{Z_i} \mathcal{P}_{\boldsymbol{\theta}^{(t)}}[Y = y] \, \mathcal{P}_{\boldsymbol{\theta}^{(t)}}[X = \mathbf{x}_i|Y = y] \\ &= \frac{1}{Z_i} c_y^{(t)} \exp \left(-\frac{1}{2} \|\mathbf{x}_i - \boldsymbol{\mu}_y^{(t)}\|^2 \right), \end{aligned} \tag{24.12}$$

  在 Z_i 是一个归一化因子，它确保 $\sum_y \mathcal{P}_{\boldsymbol{\theta}^{(t)}}[Y = y|X = \mathbf{x}_i]$ 的和为 1。

- **M**优化步骤：我们需要将 $\boldsymbol{\theta}^{t+1}$ 设置为方程（24.11）的极大值。

在我们的情况下，这相当于最大化以下表达式相对于 $\mathbf{c}$ 和 $\boldsymbol{\mu}$ 的值：

$$\sum_{i=1}^{m}\sum_{y=1}^{k}\mathcal{P}_{\boldsymbol{\theta}^{(t)}}[Y=y|X=\mathbf{x}_i]\left(\log(c_y)-\frac{1}{2}\|\mathbf{x}_i-\boldsymbol{\mu}_y\|^2\right). \tag{24.13}$$

比较方程（24.13）关于 $\boldsymbol{\mu}_y$ 的导数与零相等，并重新排列项，我们得到：

$$\boldsymbol{\mu}_y=\frac{\sum_{i=1}^{m}\mathcal{P}_{\boldsymbol{\theta}^{(t)}}[Y=y|X=\mathbf{x}_i]\,\mathbf{x}_i}{\sum_{i=1}^{m}\mathcal{P}_{\boldsymbol{\theta}^{(t)}}[Y=y|X=\mathbf{x}_i]}.$$

这是，$\boldsymbol{\mu}_y$ 是 $\mathbf{x}_i$ 的加权平均值，权重根据在 E 步中计算的概率确定。为了找到最优的 $\mathbf{c}$，我们需要更加小心，因为我们必须确保 $\mathbf{c}$ 是一个概率向量。在练习 3 中，我们展示了该解为：

$$c_y \;=\; \frac{\sum_{i=1}^{m}\mathcal{P}_{\boldsymbol{\theta}^{(t)}}[Y=y|X=\mathbf{x}_i]}{\sum_{y'=1}^{k}\sum_{i=1}^{m}\mathcal{P}_{\boldsymbol{\theta}^{(t)}}[Y=y'|X=\mathbf{x}_i]}. \tag{24.14}$$

有趣的是将前述算法与第22章中描述的 k-means 算法进行比较。在 k-means 算法中，我们首先根据距离 $\|\mathbf{x}_i-\boldsymbol{\mu}_y\|$ 将每个示例分配到一个簇中。然后，我们根据分配到该簇的示例的平均值更新每个中心 $\boldsymbol{\mu}_y$。然而，在 EM 方法中，我们确定每个示例属于每个簇的概率。然后，我们根据整个样本的加权总和更新中心。因此，k-means 的 EM 方法有时被称为“软 k-means”。

24.5 Bayesian Reasoning

最大似然估计遵循频率主义方法。这意味着我们将参数 θ 视为固定参数，唯一的问题是我们的不知道其值。参数估计的不同方法被称为贝叶斯推理。在贝叶斯方法中，我们对 θ 的不确定性也使用概率理论进行建模。也就是说，我们将 θ 也视为一个随机变量，并将分布 $\mathcal{P}[\theta]$ 称为 *prior distribution*。正如其名称所示，先验分布应由学习者在观察数据之前定义。

作为一个例子，让我们再次考虑开发新药的制药公司。基于以往的经验，制药公司的统计学家认为，每当一种药物达到在人体上进行临床试验的水平时，它很可能有效。他们通过在θ上定义一个密度分布来模拟这种先验信念，使得

$$\mathcal{P}[\theta]=\begin{cases}0.8 & \text{if } \theta>0.5\\ 0.2 & \text{if } \theta\leq 0.5\end{cases} \tag{24.15}$$

与之前一样，给定一个特定的值 θ，假设条件概率 $\mathcal{P}[X = x|\theta]$ 是已知的。在制药公司示例中，X 在 $\{0, 1\}$ 中取值，并且 $\mathcal{P}[X = x|\theta] = \theta^x(1-\theta)^{1-x}$。

一旦定义了 θ 的先验分布和给定 θ 的 X 条件分布，我们再次完全了解 X 的分布。这是因为我们可以将 X 的概率写成边缘概率

$$\mathcal{P}[X = x] = \sum_{\theta} \mathcal{P}[X = x, \theta] = \sum_{\theta} \mathcal{P}[\theta]\mathcal{P}[X = x|\theta],$$

最后等式来源于条件概率的定义。如果 θ 是连续的，我们用密度函数替换 $\mathcal{P}[\theta]$，和变为积分：

$$\mathcal{P}[X = x] = \int_{\theta} \mathcal{P}[\theta]\mathcal{P}[X = x|\theta]\, d\theta.$$

表面上，一旦我们知道 $\mathcal{P}[X = x]$，训练集 $S = (x_1, \ldots, x_m)$ 就告诉我们不了什么，因为我们已经是知道新点 X 上分布的专家。然而，贝叶斯观点引入了 S 和 X 之间的依赖关系。这是因为我们现在将 θ 视为一个随机变量。新点 X 和 S 中的前一点在 *only* 条件下是独立的 θ。这与频率主义哲学不同，在频率主义哲学中，θ 是我们可能不知道的参数，但由于它只是分布的参数，新点 X 和前一点 S 总是独立的。

在贝叶斯框架中，由于 X 和 S 已不再独立，我们想要计算的是给定 S 的 X 的概率，根据链式法则，可以表示如下：

$$\mathcal{P}[X = x|S] = \sum_{\theta} \mathcal{P}[X = x|\theta, S]\, \mathcal{P}[\theta|S] = \sum_{\theta} \mathcal{P}[X = x|\theta]\, \mathcal{P}[\theta|S].$$

第二个不等式源于假设在条件 θ 下，X 和 S 是独立的。使用 *Bayes' rule*，我们有

$$\mathcal{P}[\theta|S] = \frac{\mathcal{P}[S|\theta]\, \mathcal{P}[\theta]}{\mathcal{P}[S]},$$

并且，根据点在 θ 条件下独立的假设，我们可以写出

$$\mathcal{P}[\theta|S] = \frac{\mathcal{P}[S|\theta]\, \mathcal{P}[\theta]}{\mathcal{P}[S]} = \frac{1}{\mathcal{P}[S]} \prod_{i=1}^{m} \mathcal{P}[X = x_i|\theta]\, \mathcal{P}[\theta].$$

因此，我们得到以下贝叶斯预测的表达式：{v*}

$$\mathcal{P}[X = x|S] = \frac{1}{\mathcal{P}[S]} \sum_{\theta} \mathcal{P}[X = x|\theta] \prod_{i=1}^{m} \mathcal{P}[X = x_i|\theta]\, \mathcal{P}[\theta]. \tag{24.16}$$

回到我们的制药公司例子，我们可以将 $\mathcal{P}[X = x|S]$ 重写为

$$\mathcal{P}[X = x|S] = \frac{1}{P[S]} \int \theta^{x+\sum_i x_i}(1-\theta)^{1-x+\sum_i(1-x_i)}\, \mathcal{P}[\theta]\, d\theta.$$

有趣的是，当 $\mathcal{P}[\theta]$ 是均匀的时，我们得到

$$\mathcal{P}[X = x|S] \propto \int \theta^{x+\sum_i x_i}(1-\theta)^{1-x+\sum_i(1-x_i)}\, d\theta.$$

求解前面的积分（使用分部积分法）我们得到

$$\mathcal{P}[X = 1|S] = \frac{(\sum_i x_i) + 1}{m + 2}.$$

回忆一下，在这种情况下根据最大似然原理进行的预测是 $\mathcal{P}[X = 1|\hat{\theta}] = \frac{\sum_i x_i}{m}$。具有均匀先验的贝叶斯预测与最大似然预测相当相似，除了它向训练集中添加“伪示例”，从而使预测偏向于均匀先验。

Maximum A Posteriori

在许多情况下，很难找到方程（24.16）中给出的积分的闭式解。可以使用几种数值方法来近似这个积分。另一种流行的解决方案是找到一个单值 θ，它最大化 $\mathcal{P}[\theta|S]$。最大化 $\mathcal{P}[\theta|S]$ 的 θ 值被称为 *Maximum A Poste- riori* 估计量。一旦找到这个值，我们就可以计算在最大 *a posteriori* 估计量和独立于 S 的条件下 $X = x$ 的概率。

24.6 Summary

在机器学习的生成方法中，我们旨在对数据的分布进行建模。特别是，在参数密度估计中，我们进一步假设数据的基本分布具有特定的参数形式，我们的目标是估计模型的参数。我们已描述了参数估计的几个原则，包括最大似然、贝叶斯估计和最大*a posteriori*。我们还描述了在基本数据分布的不同假设下实现最大似然的具体算法，特别是朴素贝叶斯、LDA和EM。

24.7 Bibliographic Remarks

20世纪初，罗纳德·费希尔研究了最大似然原理。贝叶斯统计学遵循贝叶斯定理，该定理以18世纪英国数学家托马斯·贝叶斯的名字命名。

有许多关于机器学习的生成式和贝叶斯方法的优秀书籍。例如，参见（Bishop 2006，Koller & Friedman 2009，MacKay 2003，Murphy 2012，Barber 2012）。

24.8 Exercises

1. 证明高斯变量方差的极大似然估计是有偏的。

2. 最大似然的正则化：考虑以下正则化损失最小化：

$$\frac{1}{m}\sum_{i=1}^{m}\log(1/\mathcal{P}_\theta[x_i]) + \frac{1}{m}\left(\log(1/\theta) + \log(1/(1-\theta))\right).$$

- 证明前述目标与我们在训练集中添加两个伪示例时的常规经验误差等价。结论是，正则化最大似然估计器将是

$$\hat{\theta} = \frac{1}{m+2}\left(1 + \sum_{i=1}^{m} x_i\right).$$

- 推导 $|\hat{\theta} - \theta^\star|$ 的高概率界限。*Hint*：将其重写为 $|\hat{\theta} - \mathbb{E}[\hat{\theta}] + \mathbb{E}[\hat{\theta}] - \theta^\star|$，然后使用三角不等式和 Hoeffding 不等式。
- 使用此方法来界定真实风险。提示：利用现在 $\hat{\theta} \geq \frac{1}{m+2}$ 的实际情况将 $|\hat{\theta} - \theta^\star|$ 与相对熵联系起来。

3. - 考虑一个一般优化问题，其形式如下：

$$\max_{\mathbf{c}} \sum_{y=1}^{k} \nu_y \log(c_y) \quad \text{s.t.} \quad c_y > 0, \sum_y c_y = 1\ ,$$

在 $\boldsymbol{\nu} \in \mathbb{R}_+^k$ 是一个非负权重向量的情况下。验证软 k-均值的 M 步涉及解决此类优化问题。

- 设 $\mathbf{c}^\star = \frac{1}{\sum_y \nu_y}\boldsymbol{\nu}$。证明 $\mathbf{c}^\star$ 是一个概率向量。
- 证明优化问题等价于以下问题：

$$\min_{\mathbf{c}} D_{\mathrm{RE}}(\mathbf{c}^\star||\mathbf{c}) \quad \text{s.t.} \quad c_y > 0, \sum_y c_y = 1\ .$$

- 使用相对熵的性质，得出 $\mathbf{c}^\star$ 是优化问题的解。

25 Feature Selection and Generation

在本书的开头，我们讨论了学习抽象模型，其中学习者利用的先验知识完全由假设类别的选择来编码。然而，还有一种建模选择，我们迄今为止已经忽略了：我们如何表示实例空间 {v*}？例如，在木瓜学习问题中，我们在软硬-颜色二维平面上提出了矩形的假设类别。也就是说，我们的第一个建模选择是将木瓜表示为对应其软硬和颜色的二维点。只有在那时，我们才选择了矩形的假设类别作为从平面到标签集的映射类别。将现实世界对象“木瓜”转换为表示其软硬或其颜色的标量称为 *feature function* 或简称为特征；也就是说，任何对现实世界对象的测量都可以被视为特征。如果 $\mathcal{X}$ 是向量空间的一个子集，那么每个 $x \in \mathcal{X}$ 有时被称为 *feature vector*。重要的是要理解，我们将现实世界对象编码为实例空间 $\mathcal{X}$ 的方式本身就是对问题的先验知识。

此外，即使我们已经有了一个实例空间 $\mathcal{X}$，它被表示为向量空间的一个子集，我们仍然可能想要将其转换为不同的表示，并在其上应用一个假设类。也就是说，我们可以在 $\mathcal{X}$ 上定义一个假设类，通过在将 $\mathcal{X}$ 映射到某个其他向量空间 $\mathcal{X}'$ 的特征函数上组合某个类 $\mathcal{H}$。我们已经遇到了这样的组合的例子——在第15章中，我们看到基于核的SVM学习了一个将 $\mathcal{X}$ 中的每个原始实例映射到某个希尔伯特空间的半空间类的组合。实际上，ψ 的选择是我们对问题施加的另一种先验知识形式。

在这一章中，我们研究了几种构建良好特征集的方法。我们从问题 *feature selection* 开始，其中我们有一个大量的特征池，我们的目标是选择少量将被我们的预测器使用的特征。接下来，我们讨论 *feature manipulations and normalization*。这包括我们对原始特征应用的一些简单转换。这些转换可能会降低我们学习算法的样本复杂度、偏差或计算复杂度。最后，我们讨论了 *feature learning* 的几种方法。在这些方法中，我们试图自动化特征构建的过程。

我们强调，尽管有一些常见的特征学习方法可以尝试，但无免费午餐定理意味着不存在终极的特征学习方法。任何特征学习算法可能在某些问题上失败。换句话说，每个特征学习者的成功（有时是隐含的）依赖于对数据分布的某种先验假设。此外，特征的相关质量高度依赖于我们后来将使用这些特征的 学习算法。以下例子说明了这一点。

Example 25.1 考虑一个回归问题，其中 $\mathcal{X} = \mathbb{R}^2$、$\mathcal{Y} = \mathbb{R}$ 和损失函数是平方损失。假设底层分布是这样的，一个示例 $(\mathbf{x}, y)$ 是如下生成的：首先，我们从 $[-1, 1]$ 上的均匀分布中采样 x_1。然后，我们确定地设置 $y = {x_1}^2$。最后，第二个特征被设置为 $x_2 = y + z$，其中 z 是从 $[-0.01, 0.01]$ 上的均匀分布中采样的。假设我们想要选择一个特征。直观上，第一个特征应该比第二个特征更受欢迎，因为目标可以根据第一个特征单独完美预测，而无法根据第二个特征完美预测。确实，如果我们稍后要应用至少2次的二次回归，选择第一个特征将是正确的选择。然而，如果学习者是一个线性回归器，那么我们应该更喜欢 $second$ 特征而不是第一个特征，因为基于第一个特征的优线性预测器的风险将大于基于第二个特征的优线性预测器的风险。

25.1 Feature Selection

在整个本节中，我们假设$\mathcal{X} = \mathbb{R}^d$。也就是说，每个实例都表示为一个包含$d$特征的向量。我们的目标是学习一个仅依赖于$k \ll d$特征的预测器。仅使用特征子集的预测器需要更小的内存占用，并且可以更快地应用。此外，在医疗诊断等应用中，获取每个可能的“特征”（例如，测试结果）可能成本高昂；因此，即使相对于使用更多特征的预测器性能略有下降，使用仅少量特征的预测器也是可取的。最后，将假设类限制为使用特征的小子集可以减少其估计误差，从而防止过拟合。

理想情况下，我们可以尝试从d特征中所有k子集，并选择导致性能最佳的预测器的子集。然而，这种穷举搜索通常在计算上是不可行的。以下我们描述了三种计算上可行的方法来进行特征选择。虽然这些方法不能保证找到最优子集，但在实践中它们通常表现良好。其中一些方法在特定假设下对所选子集的质量提供正式保证。我们在此不讨论这些保证。

25.1.1 Filters

可能最简单的特征选择方法是过滤法，其中我们根据某些质量指标独立于其他特征评估单个特征。然后我们可以选择得分最高的 k 个特征（或者根据它们的得分值决定选择特征的数量）。

许多关于特征的质最度量在文献中已被提出。可能最直接的方法是根据仅由该特征训练的预测器的错误率来设置特征的得分。

为了说明这一点，考虑一个具有平方损失的线性回归问题。设 $\mathbf{v} = (x_{1,j}, \ldots, x_{m,j}) \in \mathbb{R}^m$ 是一个向量，表示在 m 个示例的训练集上 j 个特征的值，设 $\mathbf{y} = (y_1, \ldots, y_m) \in \mathbb{R}^m$ 是相同 m 个示例上的目标值。仅使用 j 个特征的 ERM 线性预测器的经验平方损失将是

$$\min_{a,b\in\mathbb{R}} \frac{1}{m}\|a\mathbf{v} + b - \mathbf{y}\|^2,$$

在将标量 b 加到向量 $\mathbf{v}$ 上的意义是将 b 加到 $\mathbf{v}$ 的所有坐标上。为了解决这个问题，设 $\bar{v} = \frac{1}{m}\sum_{i=1}^m v_i$ 为特征的平均值，设 $\bar{y} = \frac{1}{m}\sum_{i=1}^m y_i$ 为目标的平均值。显然（参见练习 1），

$$\min_{a,b\in\mathbb{R}} \frac{1}{m}\|a\mathbf{v} + b - \mathbf{y}\|^2 = \min_{a,b\in\mathbb{R}} \frac{1}{m}\|a(\mathbf{v} - \bar{v}) + b - (\mathbf{y} - \bar{y})\|^2. \tag{25.1}$$

对右侧目标函数关于 b 求导并与零比较，我们得到 $b = = 0$。同样，在已知 $b = = 0$) 的情况下求解 a (，得到 $a = \langle \mathbf{v} - \bar{v}, \mathbf{y} - \bar{y}\rangle / \|\mathbf{v} - \bar{v}\|^2$。将此值代入目标函数，我们得到该值

$$\|\mathbf{y} - \bar{y}\|^2 - \frac{(\langle \mathbf{v} - \bar{v}, \mathbf{y} - \bar{y}\rangle)^2}{\|\mathbf{v} - \bar{v}\|^2}.$$

根据它们实现的最低损失对特征进行排名，等同于根据以下得分的绝对值对它们进行排名（现在得分越高，特征越好）：{v*}

$$\frac{\langle \mathbf{v} - \bar{v}, \mathbf{y} - \bar{y}\rangle}{\|\mathbf{v} - \bar{v}\|\,\|\mathbf{y} - \bar{y}\|} = \frac{\frac{1}{m}\langle \mathbf{v} - \bar{v}, \mathbf{y} - \bar{y}\rangle}{\sqrt{\frac{1}{m}\|\mathbf{v} - \bar{v}\|^2}\sqrt{\frac{1}{m}\|\mathbf{y} - \bar{y}\|^2}}. \tag{25.2}$$

前一个表达式被称为 *Pearson's correlation coefficient*。分子是第 j 个特征和目标值 $\mathbb{E}[(v - \mathbb{E}\,v)(y - \mathbb{E}\,y)]$ 的经验估计 *covariance*，而分母是第 j 个特征 $\mathbb{E}[($ $v - \mathbb{E}\,v)^2]$ 的经验估计的平方根，乘以目标值的方差。皮尔逊系数的范围从 -1 到 1，其中如果皮尔逊系数为 1 或 -1，则从 $\mathbf{v}$ 到 $\mathbf{y}$ 存在线性映射，且经验风险为零。

如果皮尔逊系数等于零，这意味着从 $\mathbf{v}$ 到 $\mathbf{y}$ 的最优线性函数是全零函数，这意味着 $\mathbf{v}$ *alone* 对预测 $\mathbf{y}$ 没有用。然而，这并不意味着 $\mathbf{v}$ 是一个不好的特征，因为它可能与其他特征 $\mathbf{v}$ 一起可以完美地预测 $\mathbf{y}$。确实，考虑一个简单的例子，其中目标是通过函数 $y = x_1 + 2x_2$ 生成的。假设 x_1 是从 $\{\pm 1\}$ 和 $x_2 = -\frac{1}{2}x_1 + \frac{1}{2}z$ 上的均匀分布生成的，而 z 也独立同分布地从 $\{\pm 1\}$ 上的均匀分布生成。那么，$\mathbb{E}[x_1] = \mathbb{E}[x_2] = \mathbb{E}[y] = 0$，并且我们还有

$$\mathbb{E}[yx_1] = \mathbb{E}[x_1^2] + 2\,\mathbb{E}[x_2 x_1] = \mathbb{E}[x_1^2] - \mathbb{E}[x_1^2] + \mathbb{E}[zx_1] = 0.$$

因此，对于足够大的训练集，第一个特征很可能具有接近零的皮尔逊相关系数，因此它最可能不会被选中。然而，不了解第一个特征，没有任何函数能够很好地预测目标值。

有许多其他评分函数可以被滤波方法使用。值得注意的例子是互信息估计或接收器操作特征（ROC）曲线下的面积。所有这些评分函数都存在与之前所展示的类似问题。我们建议读者参考Guyon & Elisseeff (2003)。

25.1.2 Greedy Selection Approaches

贪婪选择是特征选择中另一种流行的方法。与过滤方法不同，贪婪选择方法与底层学习算法相结合。贪婪选择的最简单实例是前向贪婪选择。我们从一组空的特征开始，然后逐渐将一个特征一次添加到选定的特征集中。鉴于我们当前选定的特征集是{v*27}，我们遍历所有{v*28}，并在特征集{v*29}上应用学习算法。每次这样的应用都会产生一个不同的预测器，我们选择添加产生最小风险（在训练集或验证集上）的特征。这个过程一直持续到我们选择了{v*30}个特征，其中{v*31}是预定义的允许特征预算，或者达到足够准确的预测器。

Example 25.2 (**Orthogonal Matching Pursuit**) 为了说明前向贪婪选择方法，我们将其指定为平方损失线性回归问题。设$X \in \mathbb{R}^{m,d}$为一个矩阵，其行是m个训练实例。设$\mathbf{y} \in \mathbb{R}^m$为$m$个标签的向量。对于每个$i \in [d]$，设$X_i$为$X$的第$i$列。给定一个集合$I \subset [d]$，我们用$X_I$表示其列是$\{X_i : i \in I\}$的矩阵。

前向贪婪选择法从 $I_0 = \emptyset$ 开始。在迭代 t 时，我们寻找特征索引 j_t，它在

$$\operatorname*{argmin}_j \min_{\mathbf{w} \in \mathbb{R}^t} \|X_{I_{t-1} \cup \{j\}}\mathbf{w} - \mathbf{y}\|^2.$$

然后，我们更新 $I_t = I_{t-1} \cup \{j_t\}$。

我们现在描述线性回归中前向贪婪选择方法的更有效实现，称为 *Orthogonal Matching Pursuit (OMP)*。其思路是保持到目前为止聚合的特征的正交基。设 V_t 为一个矩阵，其列构成 X_{I_t} 列的正交归一基。

显然，

$$\min_{\mathbf{w}} \|X_{I_t}\mathbf{w} - \mathbf{y}\|^2 = \min_{\boldsymbol{\theta} \in \mathbb{R}^t} \|V_t\boldsymbol{\theta} - \mathbf{y}\|^2.$$

我们将保持一个向量 $\boldsymbol{\theta}_t$，该向量使方程的右侧最小化。

最初，我们将 $I_0 = \emptyset$、$V_0 = \emptyset$ 和 $\boldsymbol{\theta}_1$ 设置为空向量。在第 t 轮中，对于每个 j，我们将 $X_j = \mathbf{v}_j + \mathbf{u}_j$ 分解，其中 $\mathbf{v}_j = V_{t-1}V_{t-1}^\top X_j$ 是 X_j 在由 V_{t-1} 张成的子空间上的投影，$\mathbf{u}_j$ 是 X_j 与 V_{t-1} (正交的部分，参见附录 C)。然后，

$$\begin{aligned}
&\min_{\boldsymbol{\theta},\alpha} \|V_{t-1}\boldsymbol{\theta} + \alpha\mathbf{u}_j - \mathbf{y}\|^2 \\
&\quad = \min_{\boldsymbol{\theta},\alpha} \left[\|V_{t-1}\boldsymbol{\theta} - \mathbf{y}\|^2 + \alpha^2\|\mathbf{u}_j\|^2 + 2\alpha\langle \mathbf{u}_j, V_{t-1}\boldsymbol{\theta} - \mathbf{y}\rangle\right] \\
&\quad = \min_{\boldsymbol{\theta},\alpha} \left[\|V_{t-1}\boldsymbol{\theta} - \mathbf{y}\|^2 + \alpha^2\|\mathbf{u}_j\|^2 + 2\alpha\langle \mathbf{u}_j, -\mathbf{y}\rangle\right] \\
&\quad = \min_{\boldsymbol{\theta}} \left[\|V_{t-1}\boldsymbol{\theta} - \mathbf{y}\|^2\right] + \min_{\alpha} \left[\alpha^2\|\mathbf{u}_j\|^2 - 2\alpha\langle \mathbf{u}_j, \mathbf{y}\rangle\right] \\
&\quad = \left[\|V_{t-1}\boldsymbol{\theta}_{t-1} - \mathbf{y}\|^2\right] + \min_{\alpha} \left[\alpha^2\|\mathbf{u}_j\|^2 - 2\alpha\langle \mathbf{u}_j, \mathbf{y}\rangle\right] \\
&\quad = \|V_{t-1}\boldsymbol{\theta}_{t-1} - \mathbf{y}\|^2 - \frac{(\langle \mathbf{u}_j, \mathbf{y}\rangle)^2}{\|\mathbf{u}_j\|^2}.
\end{aligned}$$

因此，我们应该选择特征

$$j_t = \operatorname*{argmax}_j \frac{(\langle \mathbf{u}_j, \mathbf{y}\rangle)^2}{\|\mathbf{u}_j\|^2}.$$

其余的更新是为了设置

$$V_t = \left[V_{t-1},\ \frac{\mathbf{u}_{j_t}}{\|\mathbf{u}_{j_t}\|^2}\right] \quad , \quad \boldsymbol{\theta}_t = \left[\boldsymbol{\theta}_{t-1}\ ;\ \frac{\langle \mathbf{u}_{j_t}, \mathbf{y}\rangle}{\|\mathbf{u}_{j_t}\|^2}\right].$$

OMP过程维护所选特征的正交归一基，其中在先前的描述中，通过类似于Gram-Schmidt正交化的过程获得正交归一性质。在实践中，Gram-Schmidt过程通常数值不稳定。在下面的伪代码中，我们使用SVD（见第C.4节）在每一轮结束时以数值稳定的方式获得正交归一基。

Orthogonal Matching Pursuit (OMP)

input:
data matrix $X \in \mathbb{R}^{m,d}$, labels vector $\mathbf{y} \in \mathbb{R}^m$,
budget of features T
initialize: $I_1 = \emptyset$
for $t = 1, \ldots, T$
use SVD to find an orthonormal basis $V \in \mathbb{R}^{m,t-1}$ of X_{I_t}
(for $t = 1$ set V to be the all zeros matrix)
foreach $j \in [d] \setminus I_t$ let $\mathbf{u}_j = X_j - VV^\top X_j$
let $j_t = \operatorname{argmax}_{j \notin I_t : \|\mathbf{u}_j\| > 0} \frac{(\langle \mathbf{u}_j, \mathbf{y} \rangle)^2}{\|\mathbf{u}_j\|^2}$
update $I_{t+1} = I_t \cup \{j_t\}$
output I_{T+1}

More Efficient Greedy Selection Criteria

让 $R(\mathbf{w})$ 成为向量 $\mathbf{w}$ 的经验风险。在每次前向贪婪选择方法的迭代中，对于每个可能的 j，我们应该在支持为 $I_{t-1} \cup \{j\}$ 的向量 $\mathbf{w}$ 上最小化 $R(\mathbf{w})$。这可能会很耗时。

一种更简单的方法是选择使 j_t 最小的 {v*}。

$$\operatorname*{argmin}_j \min_{\eta \in \mathbb{R}} R(\mathbf{w}_{t-1} + \eta \mathbf{e}_j),$$

$\mathbf{e}_j$ 是除了 jth 元素为 1 以外的全零向量。也就是说，我们保持先前选择的坐标的权重不变，只对新变量进行优化。因此，对于每个 j，我们需要解决一个关于单个变量的优化问题，这比优化 t 要容易得多。

一个更简单的方法是使用一个“简单”的函数来上界 $R(\mathbf{w})$，然后选择导致这个上界最大减少的特征。例如，如果 R 是一个 β-平滑函数（参见第12章方程（12.5））， 那么

$$R(\mathbf{w} + \eta \mathbf{e}_j) \leq R(\mathbf{w}) + \eta \frac{\partial R(\mathbf{w})}{\partial w_j} + \beta \eta^2 / 2.$$

最小化右侧关于 η 得到 $\eta = -\frac{\partial R(\mathbf{w})}{\partial w_j} \cdot \frac{1}{\beta}$，将此值代入上述公式中得到

$$R(\mathbf{w} + \eta \mathbf{e}_j) \leq R(\mathbf{w}) - \frac{1}{2\beta} \left(\frac{\partial R(\mathbf{w})}{\partial w_j} \right)^2 .$$

此值在 $R(\mathbf{w})$ 对 w_j 的偏导数最大时最小化。因此，我们可以选择 j_t 为 $R(\mathbf{w})$ 在 $\mathbf{w}$ 处梯度最大坐标的索引。

Remark 25.3 (AdaBoost作为前向贪婪选择过程)可以将第10章中的AdaBoost算法解释为前向贪婪

关于函数的选择程序

$$R(\mathbf{w}) = \log\left(\sum_{i=1}^{m} \exp\left(-y_i \sum_{j=1}^{d} w_j h_j(\mathbf{x}_i)\right)\right). \tag{25.3}$$

查看练习3。

Backward Elimination

另一种流行的贪婪选择方法是 *backward elimination*。在这里，我们从完整的特征集开始，然后逐渐从特征集中逐个移除一个特征。鉴于我们当前选定的特征集是 I，我们遍历所有 $i \in I$，并在特征集 $I \setminus \{i\}$ 上应用学习算法。每次这样的应用都会产生一个不同的预测器，我们选择移除具有最小风险的特性 i（在训练集或验证集上从 $I \setminus \{i\}$ 获得的预测器）。

自然，向后消除想法有许多可能的变体。也可以结合前向和后向贪婪步骤。

25.1.3 Sparsity-Inducing Norms

最小化经验风险的问题，在 k 个特征预算下可以表示为

$$\min_{\mathbf{w}} L_S(\mathbf{w}) \quad \text{s.t.} \quad \|\mathbf{w}\|_0 \leq k,$$

在 [1] 处

$$\|\mathbf{w}\|_0 = |\{i : w_i \neq 0\}|.$$

换句话说，我们希望$\mathbf{w}$是稀疏的，这意味着我们只需要测量$\mathbf{w}$的非零元素对应特征。

解决这个优化问题在计算上很困难（Natarajan 1995，Davis，Mallat & Avellaneda 1997）。一种可能的放松是，将非凸函数 $\|\mathbf{w}\|_0$ 替换为 ℓ_1 范数，$\|\mathbf{w}\|_1 = \sum_{i=1}^{d} |w_i|$，并解决该问题

$$\min_{\mathbf{w}} L_S(\mathbf{w}) \quad \text{s.t.} \quad \|\mathbf{w}\|_1 \leq k_1, \tag{25.4}$$

在 k_1 是一个参数的情况下。由于 ℓ_1 范数是一个凸函数，只要损失函数是凸的，这个问题就可以有效地解决。一个相关的问题是最小化 $L_S(\mathbf{w})$ 的和加上一个 ℓ_1 范数正则化项，

$$\min_{\mathbf{w}} \left(L_S(\mathbf{w}) + \lambda \|\mathbf{w}\|_1\right), \tag{25.5}$$

在 λ 是正则化参数的情况下。由于对于任何 k_1，都存在一个 λ 使得

1 函数 $\|\cdot\|_0$ 通常被称为 ℓ_0 范数。尽管使用了“范数”的符号，$\|\cdot\|_0$ 并非真正的范数；例如，它不满足范数的正齐次性性质，$\|a\mathbf{w}\|_0 \neq |a| \|\mathbf{w}\|_0$。

方程（25.4）和方程（25.5）得出相同的解，这两个问题在某种程度上是等价的。

ℓ_1正则化通常诱导稀疏解。为了说明这一点，让我们从一个简单的优化问题开始

$$\min_{w\in\mathbb{R}}\left(\frac{1}{2}w^2 - xw + \lambda|w|\right). \tag{25.6}$$

它很容易验证（参见练习2）这个问题的解是“软阈值”算子

$$w = \text{sign}(x)\left[|x| - \lambda\right]_+, \tag{25.7}$$

在 $[a]_+ \stackrel{\text{def}}{=} \max\{a, 0\}$。也就是说，只要 x 的绝对值小于 λ，最优解将是零。

接下来，考虑一个关于平方损失的单一维度回归问题：

$$\underset{w\in\mathbb{R}^m}{\text{argmin}}\left(\frac{1}{2m}\sum_{i=1}^{m}(x_i w - y_i)^2 + \lambda|w|\right).$$

我们可以将问题重写为

$$\underset{w\in\mathbb{R}^m}{\text{argmin}}\left(\frac{1}{2}\left(\frac{1}{m}\sum_i x_i^2\right)\ w^2 - \left(\frac{1}{m}\sum_{i=1}^{m} x_i y_i\right)\ w + \lambda|w|\right).$$

为了简单起见，我们假设$\frac{1}{m}\sum_i x_i^2 = = 1$，并记 $\langle \mathbf{x}, \mathbf{y}\rangle = \sum_{i=1}^m x_i y_i$；那么最优解是

$$w = \text{sign}(\langle \mathbf{x}, \mathbf{y}\rangle)\left[|\langle \mathbf{x}, \mathbf{y}\rangle|/m - \lambda\right]_+.$$

这意味着，除非特征 $\mathbf{x}$ 和标签向量 $\mathbf{y}$ 之间的相关性大于 λ，否则解将为零。

Remark 25.4 与ℓ_1范数不同，ℓ_2范数不会诱导稀疏解。事实上，考虑上述具有ℓ_2正则化的问题，即，

$$\underset{w\in\mathbb{R}^m}{\text{argmin}}\left(\frac{1}{2m}\sum_{i=1}^{m}(x_i w - y_i)^2 + \lambda w^2\right).$$

然后，最优解是

$$w = \frac{\langle \mathbf{x}, \mathbf{y}\rangle/m}{\|\mathbf{x}\|^2/m + 2\lambda}.$$

此解决方案即使 $\mathbf{x}$ 和 $\mathbf{y}$ 之间的相关性非常小，也将不为零。相比之下，正如我们之前所展示的，当使用 ℓ_1 正则化时，只有当 $\mathbf{x}$ 和 $\mathbf{y}$ 之间的相关性大于正则化参数 λ 时，w 才不为零。

向具有平方损失的线性回归问题添加 ℓ_1 正则化得到 LASSO 算法，定义为

$$\operatorname*{argmin}_{\mathbf{w}} \left(\frac{1}{2m} \|X\mathbf{w} - \mathbf{y}\|^2 + \lambda \|\mathbf{w}\|_1 \right). \tag{25.8}$$

在某些关于分布和正则化参数 λ 的假设下，LASSO 将找到稀疏解（例如，参见（Zhao & Yu 2006）及其参考文献）。ℓ_1 范数的另一个优点是，具有低 ℓ_1 范数的向量可以被“稀疏化”（例如，参见（Shalev-Shwartz, Zhang & Srebro 2010）及其参考文献）。

25.2 Feature Manipulation and Normalization

特征操作或归一化包括我们对每个原始特征应用的一些简单变换。这些变换可能会减少我们的假设类或估计误差，或者可以产生更快的算法。与特征选择问题类似，这里也没有绝对的“好”和“坏”变换，而是我们应用的每个变换都应该与我们将要应用于结果特征向量的学习算法以及我们对问题的先验假设相关。

为了激励 *normalization*，考虑一个具有平方损失的线性回归问题。设 $X \in \mathbb{R}^{m,d}$ 为行向量是实例向量的矩阵，设 $\mathbf{y} \in \mathbb{R}^m$ 为目标值向量。回忆起岭回归返回的向量

$$\operatorname*{argmin}_{\mathbf{w}} \left[\frac{1}{m} \|X\mathbf{w} - \mathbf{y}\|^2 + \lambda \|\mathbf{w}\|^2 \right] = (2\lambda m I + X^\top X)^{-1} X^\top \mathbf{y}.$$

假设 $d =$^2 和基础数据分布如下。首先，我们从 $\{\pm 1\}$ 中随机均匀地采样 y。然后，我们将 x_1 设置为 $y + 0.5\alpha$，其中 α 从 $\{\pm 1\}$ 中随机均匀地采样，并将 x_2 设置为 $0.0001y$。请注意，最优权重向量是 $\mathbf{w}^\star = [0; 10000]$，而 $L_{\mathcal{D}}(\mathbf{w}^\star) = 0$。然而，在 $\mathbf{w}^\star$ 的岭回归目标函数是 $\lambda 10^8$。相比之下，在 $\mathbf{w} = [1; 0]$ 的岭回归目标函数可能接近 $0.25 + \lambda$。因此，当 $\lambda > \frac{0.25}{10^8 - 1} \approx 0.25 \times 10^{-8}$ 时，岭回归在次优解 $\mathbf{w} = [1; 0]$ 处的目标函数更小。由于 λ 通常至少应该是 $1/m$（参见第 13 章的分析），因此，在上述示例中，如果示例数量小于 10^8，那么我们很可能输出一个次优解。

关键在于前一个例子中两个特征具有完全不同的尺度。特征归一化可以克服这个问题。有许多执行特征归一化的方法，其中最简单的方法就是确保每个特征接收到的值在 -1 和 1 之间。在前一个例子中，如果我们把每个特征除以它达到的最大值

我们将得到 $x_1 = \frac{y+0.5\alpha}{1.5}$ 和 $x_2 = y$。然后，对于 $\lambda \leq 10^{-3}$ 的岭回归解非常接近 $\mathbf{w}^\star$。

此外，我们在第13章中为正则化损失最小化推导出的泛化界限依赖于最优向量 $\mathbf{w}^\star$ 的范数以及实例向量最大范数 [2]。因此，在上述示例中，在我们归一化特征之前，我们有 $\|\mathbf{w}^\star\|^2 = 10^8$，而在归一化特征之后，我们有 $\|\mathbf{w}^\star\|^2 = 1$。实例向量的最大范数保持大致相同；因此，归一化大大提高了估计误差。

特征归一化也可以提高学习算法的运行时间。例如，在第14.5.3节中，我们展示了如何使用随机梯度下降（SGD）优化算法来解决正则化损失最小化问题。SGD收敛所需的迭代次数也取决于$\mathbf{w}^\star$的范数以及$\|\mathbf{x}\|$的最大范数。因此，与之前一样，使用归一化可以大大减少SGD的运行时间。

接下来，我们在以下内容中演示如何对特征进行简单的变换，例如 *clipping*，有时可以降低我们假设类的大致误差。再次考虑具有平方损失的线性回归。设 $a > 1$ 为一个大数，假设目标 y 从 $\{\pm 1\}$ 中随机均匀选择，然后单个特征 x 以概率 $(1 - 1/a)$ 设置为 y，以概率 $1/a$ 设置为 ay。也就是说，大多数时候我们的特征是有界的，但以非常小的概率它得到一个非常高的值。然后，对于任何 w，w 的期望平方损失是

$$\begin{aligned} L_{\mathcal{D}}(w) &= \mathbb{E}\,\frac{1}{2}(wx - y)^2 \\ &= \left(1 - \frac{1}{a}\right)\frac{1}{2}(wy - y)^2 + \frac{1}{a}\,\frac{1}{2}(awy - y)^2. \end{aligned}$$

求解 w，我们得到 $w^\star = \frac{2a-1}{a^2+a-1}$，当 a 趋向于无穷大时，$w^\star = \frac{2a-1}{a^2+a-1}$ 趋向于 0。因此，当 a 趋向于无穷大时，目标函数在 $w^\star$ 处趋向于 0.5。例如，对于 $a = 100$，我们将得到 $L_{\mathcal{D}}(w^\star) \geq 0.48$。接下来，假设我们应用一个“裁剪”变换；即，我们使用变换 $x \mapsto \text{sign}(x)\,\min\{1, |x|\}$。然后，应用这个变换后，$w^\star$ 变为 1，$L_{\mathcal{D}}(w^\star) =$ 变为 0。这个简单的例子表明，一个简单的变换可以对近似误差产生重大影响。

当然，思考一些例子并不困难，在这些例子中，相同的特征变换实际上会损害性能并增加逼近误差。这并不令人惊讶，因为我们已经论证了特征变换

[2] More precisely, the bounds we derived in Chapter 13 for regularized loss minimization depend on $\|\mathbf{w}^\star\|^2$ and on either the Lipschitzness or the smoothness of the loss function. For linear predictors and loss functions of the form $\ell(\mathbf{w}, (\mathbf{x}, y)) = \phi(\langle \mathbf{w}, \mathbf{x}\rangle, y)$, where ϕ is convex and either 1-Lipschitz or 1-smooth with respect to its first argument, we have that ℓ is either $\|\mathbf{x}\|$-Lipschitz or $\|\mathbf{x}\|^2$-smooth. For example, for the squared loss, $\phi(a, y) = \frac{1}{2}(a - y)^2$, and $\ell(\mathbf{w}, (\mathbf{x}, y)) = \frac{1}{2}(\langle \mathbf{w}, \mathbf{x}\rangle - y)^2$ is $\|\mathbf{x}\|^2$-smooth with respect to its first argument.

应依赖于我们对问题的先前假设。在上述例子中，一个可能使我们使用“裁剪”变换的先前假设是，那些超过预定义阈值值的特征不会给我们提供额外的有用信息，因此我们可以将它们裁剪到预定义的阈值。

25.2.1 Examples of Feature Transformations

我们现在列出几种常见的特征变换技术。通常，结合一些这些变换（例如，中心化 + 缩放）是有帮助的。在以下内容中，我们用 $\mathbf{f} = (f_1, \ldots, f_m) \in \mathbb{R}^m$ 表示特征 f 在 m 个训练样本上的值。此外，我们用 $\bar{f} = \frac{1}{m}\sum_{i=1}^{m} f_i$ 表示特征在所有样本上的经验均值。

Centering:
这个变换通过设置 $f_i \leftarrow f_i - \bar{f}$ 使特征具有零均值。

Unit Range:
这个变换使得每个特征的取值范围变为[0, 1]。形式上，设$f_{\max}$ =为$_i$ f_i的最大值和$f_{\min}$ =为$_i$ f_i的最小值。然后，我们设置$f_i \leftarrow \frac{f_i - f_{\min}}{f_{\max} - f_{\min}}$。同样，通过变换 $f_i \leftarrow 2\frac{f_i - f_{\min}}{f_{\max} - f_{\min}} - 1$，我们可以使每个特征的取值范围变为$[-1, 1]$。当然，很容易使取值范围变为$[0, b]$或$[-b, b]$，其中$b$是一个用户指定的参数。

Standardization:
这个变换使得所有特征具有零均值和单位方差。形式上，设 $\nu = \frac{1}{m}\sum_{i=1}^{m}(f_i - \bar{f})^2$ 为特征的样本方差。然后，我们设置 $f_i \leftarrow \frac{f_i - \bar{f}}{\sqrt{\nu}}$ 。

Clipping:
此变换裁剪特征的高值或低值。例如，$f_i \leftarrow \text{sign}(f_i)\max\{b, |f_i|\}$，其中 b 是用户指定的参数。

Sigmoidal Transformation:
如其名称所示，这种转换在特征上应用了一个Sigmoid函数。例如，
$f_i \leftarrow \frac{1}{1+\exp(b\, f_i)}$，其中$b$是一个用户指定的参数。这种转换可以被视为“软”版本的裁剪：它对接近零的值影响很小，并且在远离零的值上表现得类似于裁剪。

Logarithmic Transformation:
变换是 $f_i \leftarrow \log(b+f_i)$，其中 b 是用户指定的参数。这在特征是“计数”特征时被广泛使用。例如，假设该特征表示文本文档中某个词出现的次数。那么，该词零次出现和单次出现之间的差异比1000次出现和1001次出现之间的差异要重要得多。

Remark 25.5 在上述变换中，每个特征都是基于它在训练集上获得的值进行变换的，与其他特征的值无关。在某些情况下，我们希望基于其他特征来设置变换的参数。一个显著的例子是，对特征应用缩放，使得实例的某个范数的经验平均值变为1。

25.3 Feature Learning

截至目前，我们已讨论了特征选择和处理。在这些情况下，我们从一个预定义的向量空间 $\mathbb{R}^d$ 开始，代表我们的特征。然后，我们选择特征的一个子集（特征选择）或转换单个特征（特征转换）。在本节中，我们描述 *feature learning*，其中我们从一个实例空间 $\mathcal{X}$ 开始，并希望学习一个函数 $\psi : \mathcal{X} \to \mathbb{R}^d$，它将 $\mathcal{X}$ 中的实例映射为 d 维特征向量。

特征学习的想法是自动化寻找输入空间良好表示的过程。如前所述，无免费午餐定理告诉我们，为了构建良好的特征表示，我们必须在数据分布中引入一些先验知识。在本节中，我们介绍了几种特征学习方法，并展示了这些方法在哪些底层数据分布条件下可能是有用的。

全书我们已经看到了几个有用的特征构造。例如，在多项式回归的背景下，我们将原始实例映射到所有单变量的向量空间（参见第9章第9.2.2节）。在执行此映射后，我们在构建的特征上训练了一个 *linear* 预测器。自动化此过程就是学习一个变换 $\psi : \mathcal{X} \to \mathbb{R}^d$，使得在 ψ 上线性预测器的组合为当前任务提供良好的假设类。

在以下内容中，我们描述了一种称为 *dictio- nary learning* 的特征构建技术。

25.3.1 Dictionary Learning Using Auto-Encoders

词典学习的动机源于将文档表示为“词袋”的常用表示方法：给定一个包含单词的字典 $D = \{w_1, \ldots, w_k\}$，其中每个 w_i 是表示字典中单词的字符串，

并且给定一个文档，($p_1, \ldots, p_d$)，其中每个 p_i 是文档中的一个单词，我们将文档表示为一个向量 $\mathbf{x} \in \{0,1\}^k$，其中 x_i 如果对于某些 $j \in [d]$ 的 $w_i = p_j$ 为 1，否则为 $x_i = 0$。在许多文本处理任务中，通过经验观察到，当应用于这种表示时，线性预测器非常强大。直观上，我们可以将每个单词视为一个特征，它衡量文档的某些方面。给定标记的示例（例如，文档的主题），学习算法寻找一个线性预测器，以加权这些特征，使得单词出现的正确组合可以指示标签。

在文本处理中，词语和词典具有自然的意义，但在其他应用中，我们没有这样直观的实例表示。例如，考虑计算机视觉应用中的物体识别。在这里，实例是一张图像，目标是识别图像中出现的物体。对基于像素的图像表示应用线性预测器并不能得到一个好的分类器。我们希望有一个映射 ψ，它将基于像素的图像表示转换为“视觉词”的集合，表示图像的内容。例如，“视觉词”可以是“图像中有一个眼睛。”如果我们有这样的表示，我们就可以在这个表示之上应用线性预测器来训练一个分类器，例如用于人脸识别。因此，我们的问题是，我们如何学习一个“视觉词”的词典，使得图像的“词袋”表示有助于预测图像中出现的物体?

一种用于字典学习的初始朴素方法依赖于一个 *clustering* 算法（见第22章）。假设我们学习一个函数 $c : \mathcal{X} \to \{1, \ldots, k\}$，其中 $c(\mathbf{x})$ 是 $\mathbf{x}$ 所属的簇。然后，我们可以将簇视为“单词”，将实例视为“文档”，其中文档 $\mathbf{x}$ 映射到向量 $\psi(\mathbf{x}) \in \{0,1\}^k$，其中 $\psi(\mathbf{x})_i$ 为1当且仅当 $\mathbf{x}$ 属于第 i 个簇。现在，很明显，在 $\psi(\mathbf{x})$ 上应用线性预测器相当于将相同的目标值分配给属于同一簇的所有实例。此外，如果聚类是基于到类中心的距离（例如，k-means），则在 $\psi(\mathbf{x})$ 上的线性预测器产生一个分段常数预测器 $\mathbf{x}$。

两个 k-means 和 PCA 方法都可以被视为更一般字典学习方法的特例，这种方法被称为 *auto-encoders*。在自编码器中，我们学习一对函数：一个“编码器”函数，$\psi : \mathbb{R}^d \to \mathbb{R}^k$，和一个“解码器”函数，$\phi : \mathbb{R}^k \to \mathbb{R}^d$。学习过程的目标是找到一个函数对，使得重建误差，$\sum_i \|\mathbf{x}_i - \phi(\psi(\mathbf{x}_i))\|^2$，很小。当然，我们可以简单地设置 $k = d$ 和 ψ, ϕ 为恒等映射，从而得到完美的重建。因此，我们必须以某种方式限制 ψ 和 ϕ。在 PCA 中，我们限制 $k < d$ 并进一步限制 ψ 和 ϕ 为线性函数。在 k-means 中，k 不限制必须小于 d，但现在 ψ 和 ϕ 依赖于 k 聚类中心，$\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k$，并且 $\psi(\mathbf{x})$ 返回一个指示向量

在 $\{0,1\}^k$ 中表示最接近 $\mathbf{x}$ 的质心，而 ϕ 以指示向量作为输入并返回表示此向量的质心。

一个重要的性质是k-均值构造，这对于允许k大于d是关键的，即ψ将实例映射到$sparse$向量。事实上，在k-均值中，只有$\psi(\mathbf{x})$的一个坐标是非零的。因此，k-均值构造的一个直接扩展是限制ψ的范围为最多有s个非零元素的向量，其中s是一个小的整数。特别是，让ψ和ϕ是依赖于$\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k$的函数。函数$\psi$将一个实例向量$\mathbf{x}$映射到一个向量$\psi(\mathbf{x}) \in \mathbb{R}^k$，其中$\psi(\mathbf{x})$应该最多有$s$个非零元素。函数$\phi(\mathbf{v})$定义为$\sum_{i=1}^{k} v_i \boldsymbol{\mu}_i$。和之前一样，我们的目标是保持小的重建误差，因此我们可以定义

$$\psi(\mathbf{x}) = \operatorname*{argmin}_{\mathbf{v}} \|\mathbf{x} - \phi(\mathbf{v})\|^2 \;\; \text{s.t.} \;\; \|\mathbf{v}\|_0 \leq s,$$

在 $\|\mathbf{v}\|_0 = |\{j : v_j \neq 0\}|$。注意，当 $s = 1$ 且我们进一步限制 $\|\mathbf{v}\|_1 = 1$ 时，我们获得 k-means 编码函数；即，$\psi(\mathbf{x})$ 是距离 $\mathbf{x}$ 最近的质心的指示向量。对于 s 的较大值，先前提到的 ψ 定义中的优化问题变得计算困难。因此，在实践中，我们有时使用 ℓ_1 正则化代替稀疏性约束，并定义 ψ 为

$$\psi(\mathbf{x}) = \operatorname*{argmin}_{\mathbf{v}} \left[\|\mathbf{x} - \phi(\mathbf{v})\|^2 + \lambda \|\mathbf{v}\|_1\right],$$

在 $\lambda > 0$ 是一个正则化参数。无论如何，字典学习问题现在是要找到向量 $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k$，使得重建误差 $\sum_{i=1}^{m} \|\mathbf{x}_i - \phi(\psi(\mathbf{x}))\|^2$ 尽可能小。即使 ψ 使用 ℓ_1 正则化定义，这仍然是一个计算上困难的问题（类似于 k-means 问题）。然而，几种启发式搜索算法可能给出相当好的解决方案。这些算法超出了本书的范围。

25.4 Summary

许多机器学习算法都默认实例的特征表示。然而，表示的选择需要仔细关注。我们讨论了特征选择的方法，介绍了过滤器、贪婪选择算法和稀疏性诱导范数。接下来，我们展示了几个特征变换的例子，并证明了它们的有用性。最后，我们讨论了特征学习，特别是字典学习。我们已经表明，特征选择、操作和学习都依赖于对数据的某些先验知识。

25.5 Bibliographic Remarks

Guyon & Elisseeff (2003) 调查了多种特征选择过程，包括许多类型的过滤器。

前向贪婪选择程序，用于最小化受多面体约束的凸目标，其起源可追溯到Frank-Wolfe算法（Frank & Wolfe 1956）。许多作者研究了其与提升的关系，包括（Warmuth, Liao & Ratsch 2006，Warmuth, Glocer & Vishwanathan 2008，Shalev-Shwartz & Singer 2008）。匹配追踪在信号处理领域已被研究（Mallat & Zhang 1993）。几篇论文分析了在各种条件下贪婪选择方法。例如，参见Shalev-Shwartz, Zhang & Srebro（2010）及其参考文献。

使用ℓ_1-范数作为稀疏性的代理具有悠久的历史（例如，Tib- shirani（1996）及其参考文献），在理解ℓ_1-范数与稀疏性之间的关系方面也做了大量工作。它也与压缩感知密切相关（参见第23章）。将低ℓ_1范数预测器稀疏化的能力可以追溯到Maurey（Pisier 1980-1981）。在第26.4节中，我们还表明，低ℓ_1范数可以用来界定我们预测器的估计误差。

特征学习和字典学习最近在深度神经网络背景下得到了广泛研究。例如，参见（Lecun & Bengio 1995，Hinton等2006，Ranzato等2007，Collobert & Weston 2008，Lee等2009，Le等2012，Bengio 2009）及其参考文献。

25.6 Exercises

1. 证明方程（25.1）中给出的等式。*Hint*：设a^*, b^*为左侧的最小值。找到a, b，使得右侧的目标值小于左侧的目标值。对于另一个方向也做同样的事情。2. 证明方程（25.7）是方程（25.6）的解。3.
AdaBoost as a Forward Greedy Selection Algorithm: 回忆第10章中的Ad- aBoost算法。在本节中，我们给出AdaBoost作为前向贪婪选择算法的另一种解释。

- 给定一个由 m 实例 $\mathbf{x}_1, \ldots, \mathbf{x}_m$ 组成的集合，以及一个有限 VC 维度的假设类 $\mathcal{H}$，证明存在 d 和 $h_1, \ldots, h_d$，使得对于每一个 $h \in \mathcal{H}$，存在一个 $i \in [d]$，使得对于每一个 $j \in [m]$，都有 $h_i(\mathbf{x}_j) = h(\mathbf{x}_j)$。
- 设 $R(\mathbf{w})$ 如方程（25.3）中定义。给定某些 $\mathbf{w}$，定义 $f_{\mathbf{w}}$ 为函数

$$f_{\mathbf{w}}(\cdot) = \sum_{i=1}^{d} w_i h_i(\cdot).$$

设 $\mathbf{D}$ 为由 $[m]$ 定义的分发

$$D_i = \frac{\exp(-y_i f_{\mathbf{w}}(\mathbf{x}_i))}{Z},$$

在 Z 是一个确保 $\mathbf{D}$ 是概率向量的归一化因子的情况下。证明

$$\frac{\partial R(\mathbf{w})}{w_j} = -\sum_{i=1}^{m} D_i y_i h_j(\mathbf{x}_i).$$

此外，表示 $\epsilon_j = \sum_{i=1}^{m} D_i 1_{[h_j(\mathbf{x}_i) \neq y_i]}$，证明

$$\frac{\partial R(\mathbf{w})}{w_j} = 2\epsilon_j - 1.$$

结论是，如果 $\epsilon_j \leq 1/2 - \gamma$，那么 $\left|\frac{\partial R(\mathbf{w})}{w_j}\right| \geq \gamma/2$。

- 证明AdaBoost的更新保证了$R(\mathbf{w}^{(t+1)}) - R(\mathbf{w}^{(t)}) \leq \log(\sqrt{1-4\gamma^2})$。*Hint*：使用定理10.2的证明。

Part IV

Advanced Theory

26 Rademacher Complexities

第四章中，我们已证明一致收敛是可学习性的充分条件。在本章中，我们研究Rademacher复杂度，该复杂度衡量一致收敛的速度。我们将基于此度量提供泛化界限。

26.1 The Rademacher Complexity

回忆第4章中ϵ-代表性样本的定义，此处为方便起见重复列出。

DEFINITION 26.1 (ϵ-代表性样本) 如果一个训练集 S 被称为 ϵ-代表性（相对于领域 Z，假设类 $\mathcal{H}$，损失函数 ℓ，和分布 $\mathcal{D}$），则

$$\sup_{h\in\mathcal{H}} |L_{\mathcal{D}}(h) - L_S(h)| \leq \epsilon.$$

我们已经证明，如果 S 是一个 $\epsilon/2$ 代表性样本，那么 ERM 规则是一致的，即 $L_{\mathcal{D}}(\text{ERM}_{\mathcal{H}}(S)) \leq \min_{h\in\mathcal{H}} L_{\mathcal{D}}(h) + \epsilon$。

为了简化我们的符号，让我们表示

$$\mathcal{F} \stackrel{\text{def}}{=} \ell \circ \mathcal{H} \stackrel{\text{def}}{=} \{z \mapsto \ell(h,z) : h \in \mathcal{H}\},$$

并且给定 $f \in \mathcal{F}$，我们定义

$$L_{\mathcal{D}}(f) = \mathop{\mathbb{E}}_{z\sim\mathcal{D}}[f(z)] \quad , \quad L_S(f) = \frac{1}{m}\sum_{i=1}^{m} f(z_i).$$

我们定义 *representativeness* 为相对于 $\mathcal{F}$ 的 S，即函数 f 的真实误差与其经验误差之间的最大差距，即，

$$\text{Rep}_{\mathcal{D}}(\mathcal{F}, S) \stackrel{\text{def}}{=} \sup_{f\in\mathcal{F}} \big(L_{\mathcal{D}}(f) - L_S(f)\big). \tag{26.1}$$

现在，假设我们只想使用样本 S 来估计 S 的代表性。一个简单的方法是将 S 分成两个不相交的集合，$S = S_1 \cup S_2$；将 S_1 作为验证集，将 S_2 作为训练集。然后我们可以通过以下方式估计 S 的代表性：

$$\sup_{f\in\mathcal{F}} \big(L_{S_1}(f) - L_{S_2}(f)\big). \tag{26.2}$$

这可以通过定义 $\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_m) \in \{\pm 1\}^m$ 为一个向量来更紧凑地表示，使得 $S_1 = \{z_i : \sigma_i = 1\}$ 和 $S_2 = \{z_i : \sigma_i = -1\}$。然后，如果我们进一步假设 $|S_1| = |S_2|$，则方程（26.2）可以重写为

$$\frac{2}{m} \sup_{f \in \mathcal{F}} \sum_{i=1}^{m} \sigma_i f(z_i). \tag{26.3}$$

Rademacher复杂度度量通过考虑对随机选择的$\boldsymbol{\sigma}$的期望来捕捉这个想法。形式上，令$\mathcal{F} \circ S$为函数$f \in \mathcal{F}$在样本S上可以实现的全部可能评估的集合，即，

$$\mathcal{F} \circ S = \{(f(z_1), \ldots, f(z_m)) : f \in \mathcal{F}\}.$$

变量在 {v*} 中应独立同分布于 $\mathbb{P}[\sigma_i = 1] = \mathbb{P}[\sigma_i = -1] = \frac{1}{2}$。然后，定义 $\mathcal{F}$ 关于 S 的 Rademacher 复杂度为以下：

$$R(\mathcal{F} \circ S) \stackrel{\text{def}}{=} \frac{1}{m} \mathop{\mathbb{E}}_{\boldsymbol{\sigma} \sim \{\pm 1\}^m} \left[\sup_{f \in \mathcal{F}} \sum_{i=1}^{m} \sigma_i f(z_i) \right]. \tag{26.4}$$

更一般地，给定一组向量，$A \subset \mathbb{R}^m$，我们定义

$$R(A) \stackrel{\text{def}}{=} \frac{1}{m} \mathop{\mathbb{E}}_{\boldsymbol{\sigma}} \left[\sup_{\mathbf{a} \in A} \sum_{i=1}^{m} \sigma_i a_i \right]. \tag{26.5}$$

以下引理将 S 的代表性期望值限制为期望 Rademacher 复杂度的两倍。

LEMMA 26.2

$$\mathop{\mathbb{E}}_{S \sim \mathcal{D}^m} [\, \mathrm{Rep}_{\mathcal{D}}(\mathcal{F}, S)] \;\leq\; 2 \mathop{\mathbb{E}}_{S \sim \mathcal{D}^m} R(\mathcal{F} \circ S).$$

Proof 设 $S' = \{z'_1, \ldots, z'_m\}$ 为另一个独立同分布的样本。显然，对于所有 $f \in \mathcal{F}$，有 $L_{\mathcal{D}}(f) = \mathbb{E}_{S'}[L_{S'}(f)]$。因此，对于每个 $f \in \mathcal{F}$，我们有

$$L_{\mathcal{D}}(f) - L_S(f) \;=\; \mathop{\mathbb{E}}_{S'}[L_{S'}(f)] - L_S(f) \;=\; \mathop{\mathbb{E}}_{S'}[L_{S'}(f) - L_S(f)].$$

取两边的上确界，并利用上确界比期望的上确界小的性质，我们得到

$$\begin{aligned} \sup_{f \in \mathcal{F}} \big(L_{\mathcal{D}}(f) - L_S(f)\big) \;&=\; \sup_{f \in \mathcal{F}} \mathop{\mathbb{E}}_{S'}[L_{S'}(f) - L_S(f)] \\ &\leq\; \mathop{\mathbb{E}}_{S'} \left[\sup_{f \in \mathcal{F}} \big(L_{S'}(f) - L_S(f)\big) \right]. \end{aligned}$$

对 S 的期望值在等式两边取，我们得到

$$\begin{aligned} \mathop{\mathbb{E}}_{S} \left[\sup_{f \in \mathcal{F}} \big(L_{\mathcal{D}}(f) - L_S(f)\big) \right] &\leq \mathop{\mathbb{E}}_{S,S'} \left[\sup_{f \in \mathcal{F}} \big(L_{S'}(f) - L_S(f)\big) \right] \\ &= \frac{1}{m} \mathop{\mathbb{E}}_{S,S'} \left[\sup_{f \in \mathcal{F}} \sum_{i=1}^{m} (f(z'_i) - f(z_i)) \right]. \end{aligned} \tag{26.6}$$

接下来，我们注意到对于每个 j，z_j 和 z'_j 都是独立同分布的变量。因此，我们可以替换它们而不影响期望：

$$\begin{aligned}&\mathop{\mathbb{E}}_{S,S'}\left[\sup_{f\in\mathcal{F}}\left((f(z'_j)-f(z_j))+\sum_{i\neq j}(f(z'_i)-f(z_i))\right)\right] = \\ &\mathop{\mathbb{E}}_{S,S'}\left[\sup_{f\in\mathcal{F}}\left((f(z_j)-f(z'_j))+\sum_{i\neq j}(f(z'_i)-f(z_i))\right)\right].\end{aligned} \tag{26.7}$$

设 σ_j 为一个随机变量，满足 $\mathbb{P}[\sigma_j = 1] = \mathbb{P}[\sigma_j = -1] = 1/2$。从方程（26.7）中我们得到

$$\begin{aligned}&\mathop{\mathbb{E}}_{S,S',\sigma_j}\left[\sup_{f\in\mathcal{F}}\left(\sigma_j(f(z'_j)-f(z_j))+\sum_{i\neq j}(f(z'_i)-f(z_i))\right)\right] \\ &= \frac{1}{2}(\text{l.h.s. of Equation (26.7)}) + \frac{1}{2}(\text{r.h.s. of Equation (26.7)}) \\ &= \mathop{\mathbb{E}}_{S,S'}\left[\sup_{f\in\mathcal{F}}\left((f(z'_j)-f(z_j))+\sum_{i\neq j}(f(z'_i)-f(z_i))\right)\right].\end{aligned} \tag{26.8}$$

对此重复所有 j，我们得到

$$\mathop{\mathbb{E}}_{S,S'}\left[\sup_{f\in\mathcal{F}}\sum_{i=1}^{m}(f(z'_i)-f(z_i))\right] = \mathop{\mathbb{E}}_{S,S',\boldsymbol{\sigma}}\left[\sup_{f\in\mathcal{F}}\sum_{i=1}^{m}\sigma_i(f(z'_i)-f(z_i))\right]. \tag{26.9}$$

最后，

$$\sup_{f\in\mathcal{F}}\sum_i\sigma_i(f(z'_i)-f(z_i)) \le \sup_{f\in\mathcal{F}}\sum_i\sigma_i f(z'_i) + \sup_{f\in\mathcal{F}}\sum_i -\sigma_i f(z_i)$$

并且，由于 $\boldsymbol{\sigma}$ 的概率与 $-\boldsymbol{\sigma}$ 的概率相同，方程（26.9）的右侧可以被限制在

$$\begin{aligned}&\mathop{\mathbb{E}}_{S,S',\boldsymbol{\sigma}}\left[\sup_{f\in\mathcal{F}}\sum_i\sigma_i f(z'_i) + \sup_{f\in\mathcal{F}}\sum_i\sigma_i f(z_i)\right] \\ &\quad = m\mathop{\mathbb{E}}_{S'}[R(\mathcal{F}\circ S')] + m\mathop{\mathbb{E}}_{S}[R(\mathcal{F}\circ S)] = 2m\mathop{\mathbb{E}}_{S}[R(\mathcal{F}\circ S)].\end{aligned}$$

□

该词项立即得出结论，在期望值下，ERM规则找到的假设与$\mathcal{H}$中的最优假设非常接近。

THEOREM 26.3 *We have*

$$\mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}[L_{\mathcal{D}}(\mathrm{ERM}_{\mathcal{H}}(S)) - L_S(\mathrm{ERM}_{\mathcal{H}}(S))] \le 2\mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}R(\ell\circ\mathcal{H}\circ S).$$

Furthermore, for any $h^\star\in\mathcal{H}$

$$\mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}[L_{\mathcal{D}}(\mathrm{ERM}_{\mathcal{H}}(S)) - L_{\mathcal{D}}(h^\star)] \le 2\mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}R(\ell\circ\mathcal{H}\circ S).$$

Furthermore, if $h^\star = \text{argmin}_h L_{\mathcal{D}}(h)$ *then for each* $\delta \in (0,1)$ *with probability of at least* $1-\delta$ *over the choice of* S *we have*

$$L_{\mathcal{D}}(\text{ERM}_{\mathcal{H}}(S)) - L_{\mathcal{D}}(h^\star) \;\leq\; \frac{2\,\mathbb{E}_{S'\sim\mathcal{D}^m} R(\ell\circ\mathcal{H}\circ S')}{\delta}.$$

Proof 第一条不等式直接由引理26.2得出。第二条不等式成立，因为对于任何固定的 $h^\star$,

$$L_{\mathcal{D}}(h^\star) = \mathop{\mathbb{E}}_{S}[L_S(h^\star)] \geq \mathop{\mathbb{E}}_{S}[L_S(\text{ERM}_{\mathcal{H}}(S))].$$

第三不等式由前一个不等式通过依赖马尔可夫不等式（注意随机变量 $L_{\mathcal{D}}(\text{ERM}_{\mathcal{H}}(S)) - L_{\mathcal{D}}(h^\star)$ 是非负的）得出。

□

接下来，我们推导出类似于定理26.3中的界，并且对置信参数 δ 的依赖性更好。为了做到这一点，我们首先引入以下有界差异集中不等式。

LEMMA 26.4 (McDiarmid不等式) *Let* V *be some set and let* $f : V^m \to \mathbb{R}$ *be a function of* m *variables such that for some* $c > 0$*, for all* $i \in [m]$ *and for all* $x_1, \ldots, x_m, x_i' \in V$ *we have*

$$|f(x_1,\ldots,x_m) - f(x_1,\ldots,x_{i-1},x_i',x_{i+1},\ldots,x_m)| \leq c.$$

Let $X_1,\ldots,X_m$ *be* m *independent random variables taking values in* V*. Then, with probability of at least* $1-\delta$ *we have*

$$|f(X_1,\ldots,X_m) - \mathbb{E}[f(X_1,\ldots,X_m)]| \leq c\sqrt{\ln\left(\tfrac{2}{\delta}\right)\, m/2}.$$

基于McDiarmid不等式，我们可以推导出具有更好置信参数依赖性的泛化界限。

THEOREM 26.5 *Assume that for all* z *and* $h \in \mathcal{H}$ *we have that* $|\ell(h,z)| \leq c$*. Then,*

1. *With probability of at least* $1-\delta$*, for all* $h \in \mathcal{H}$*,*

$$L_{\mathcal{D}}(h) - L_S(h) \;\leq\; 2 \mathop{\mathbb{E}}_{S'\sim D^m} R(\ell\circ\mathcal{H}\circ S') + c\sqrt{\frac{2\,\ln(2/\delta)}{m}}.$$

   In particular, this holds for $h = \text{ERM}_{\mathcal{H}}(S)$*.*
2. *With probability of at least* $1-\delta$*, for all* $h \in \mathcal{H}$*,*

$$L_{\mathcal{D}}(h) - L_S(h) \;\leq\; 2\,R(\ell\circ\mathcal{H}\circ S) + 4\,c\sqrt{\frac{2\,\ln(4/\delta)}{m}}.$$

   In particular, this holds for $h = \text{ERM}_{\mathcal{H}}(S)$*.*
3. *For any* $h^\star$*, with probability of at least* $1-\delta$*,*

$$L_{\mathcal{D}}(\text{ERM}_{\mathcal{H}}(S)) - L_{\mathcal{D}}(h^\star) \;\leq\; 2\,R(\ell\circ\mathcal{H}\circ S) + 5\,c\sqrt{\frac{2\,\ln(8/\delta)}{m}}.$$

Proof 首先注意，随机变量 $\text{Rep}_{\mathcal{D}}(\mathcal{F}, S) = \sup_{h\in\mathcal{H}} (L_{\mathcal{D}}(h) - L_S(h))$ 满足引理26.4的有界差分条件，常数为 $2c/m$。将引理26.4中的界限与引理26.2结合，我们得到，至少以$1-\delta$的概率，

$$\text{Rep}_{\mathcal{D}}(\mathcal{F}, S) \le \mathbb{E}\,\text{Rep}_{\mathcal{D}}(\mathcal{F}, S) + c\sqrt{\frac{2\,\ln(2/\delta)}{m}} \le 2\,\mathop{\mathbb{E}}_{S'} R(\ell\circ\mathcal{H}\circ S') + c\sqrt{\frac{2\,\ln(2/\delta)}{m}}.$$

定理的第一个不等式由 $\text{Rep}_{\mathcal{D}}(\mathcal{F}, S)$ 的定义得出。对于第二个不等式，我们注意到随机变量 $R(\ell\circ\mathcal{H}\circ S)$ 也满足引理26.4中关于有界差异条件的常数$2c/m$。因此，第二个不等式由第一个不等式、引理26.4和并集不等式得出。最后，对于最后一个不等式，令 $h_S =$ 为 $\text{ERM}_{\mathcal{H}}(S)$，并注意到

$$\begin{aligned} &L_{\mathcal{D}}(h_S) - L_{\mathcal{D}}(h^\star) \\ &\quad = L_{\mathcal{D}}(h_S) - L_S(h_S) + L_S(h_S) - L_S(h^\star) + L_S(h^\star) - L_{\mathcal{D}}(h^\star) \\ &\quad \le (L_{\mathcal{D}}(h_S) - L_S(h_S)) + (L_S(h^\star) - L_{\mathcal{D}}(h^\star))\,. \end{aligned} \tag{26.10}$$

右侧的第一个项由定理的第二不等式所界定。对于第二个项，我们使用事实 $h^\star$ 不依赖于 S；因此，通过使用Hoeffding不等式，我们得到至少以概率$1-\delta/2$,

$$L_S(h^\star) - L_{\mathcal{D}}(h^\star) \;\le\; c\sqrt{\frac{\ln(4/\delta)}{2m}}. \tag{26.11}$$

结合这一点与并集界，我们得出我们的证明。 □

前述定理告诉我们，如果数量 {v*} 很小，则可以使用 ERM 规则学习类别 {v*}。重要的是强调，定理中给出的最后两个界限取决于特定的训练集 {v*}。也就是说，我们使用 {v*} 既是学习从 {v*} 中的假设，也是估计其质量。这种类型的界限称为 {v*}。

26.1.1 Rademacher Calculus

让我们现在讨论Rademacher复杂度测量的某些性质。这些性质将帮助我们为感兴趣的特定情况推导出$R(\ell\circ\mathcal{H}\circ S)$的一些简单界限。

以下引理直接来自定义。

LEMMA 26.6 *For any $A \subset \mathbb{R}^m$, scalar $c \in \mathbb{R}$, and vector $\mathbf{a}_0 \in \mathbb{R}^m$, we have*

$$R(\{c\,\mathbf{a} + \mathbf{a}_0 : \mathbf{a} \in A\}) \le |c|\, R(A).$$

以下引理告诉我们，A的凸包与A具有相同的复杂性。

LEMMA 26.7 *Let A be a subset of $\mathbb{R}^m$ and let $A' = \{\sum_{j=1}^N \alpha_j \mathbf{a}^{(j)} : N \in \mathbb{N}, \forall j, \mathbf{a}^{(j)} \in A, \alpha_j \geq 0, \|\boldsymbol{\alpha}\|_1 = 1\}$. Then, $R(A') = R(A)$.*

Proof 主要思想源于以下事实：对于任何向量 $\mathbf{v}$，我们有

$$\sup_{\boldsymbol{\alpha}\geq \mathbf{0}:\|\boldsymbol{\alpha}\|_1=1} \sum_{j=1}^{N} \alpha_j v_j \;=\; \max_j v_j.$$

因此，

$$\begin{aligned} m\,R(A') &= \mathop{\mathbb{E}}_{\boldsymbol{\sigma}} \sup_{\boldsymbol{\alpha}\geq \mathbf{0}:\|\boldsymbol{\alpha}\|_1=1} \sup_{\mathbf{a}^{(1)},\ldots,\mathbf{a}^{(N)}} \sum_{i=1}^{m} \sigma_i \sum_{j=1}^{N} \alpha_j a_i^{(j)} \\ &= \mathop{\mathbb{E}}_{\boldsymbol{\sigma}} \sup_{\boldsymbol{\alpha}\geq \mathbf{0}:\|\boldsymbol{\alpha}\|_1=1} \sum_{j=1}^{N} \alpha_j \sup_{\mathbf{a}^{(j)}} \sum_{i=1}^{m} \sigma_i a_i^{(j)} \\ &= \mathop{\mathbb{E}}_{\boldsymbol{\sigma}} \sup_{\mathbf{a}\in A} \sum_{i=1}^{m} \sigma_i a_i \\ &= m\,R(A), \end{aligned}$$

我们得出证明的结论。 □

下一个引理，归功于Massart，表明有限集的Rademacher复杂度随着集合大小的对数增长。

LEMMA 26.8 (Massart引理) *Let $A = \{\mathbf{a}_1, \ldots, \mathbf{a}_N\}$ be a finite set of vectors in $\mathbb{R}^m$. Define $\bar{\mathbf{a}} = \frac{1}{N}\sum_{i=1}^N \mathbf{a}_i$. Then,*

$$R(A) \;\leq\; \max_{\mathbf{a}\in A} \|\mathbf{a} - \bar{\mathbf{a}}\| \, \frac{\sqrt{2\,\log(N)}}{m}.$$

Proof 基于引理26.6，我们可以不失一般性地假设{v*}。设$\lambda >$为0，并设 $A' = \{\lambda \mathbf{a}_1, \ldots, \lambda \mathbf{a}_N\}$。我们如下上界Rademacher复杂度：

$$\begin{aligned} mR(A') = \mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\left[\max_{\mathbf{a}\in A'} \langle \boldsymbol{\sigma}, \mathbf{a}\rangle\right] \;&=\; \mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\left[\log\left(\max_{\mathbf{a}\in A'} e^{\langle \boldsymbol{\sigma}, \mathbf{a}\rangle}\right)\right] \\ &\leq \mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\left[\log\left(\sum_{\mathbf{a}\in A'} e^{\langle \boldsymbol{\sigma}, \mathbf{a}\rangle}\right)\right] \\ &\leq \log\left(\mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\left[\sum_{\mathbf{a}\in A'} e^{\langle \boldsymbol{\sigma}, \mathbf{a}\rangle}\right]\right) \qquad \text{// Jensen's inequality} \\ &= \log\left(\sum_{\mathbf{a}\in A'} \prod_{i=1}^{m} \mathop{\mathbb{E}}_{\sigma_i}\left[e^{\sigma_i a_i}\right]\right), \end{aligned}$$

在最后一个等式成立，因为Rademacher变量是独立的。接下来，使用引理A.6，我们有对于所有$a_i \in \mathbb{R}$，

$$\mathop{\mathbb{E}}_{\sigma_i} e^{\sigma_i a_i} = \frac{\exp(a_i) + \exp(-a_i)}{2} \leq \exp(a_i^2/2),$$

因此

$$mR(A') \le \log\left(\sum_{\mathbf{a}\in A'}\prod_{i=1}^{m}\exp\left(\frac{a_i^2}{2}\right)\right) = \log\left(\sum_{\mathbf{a}\in A'}\exp\left(\|\mathbf{a}\|^2/2\right)\right)$$
$$\le \log\left(|A'|\max_{\mathbf{a}\in A'}\exp\left(\|\mathbf{a}\|^2/2\right)\right) = \log(|A'|) + \max_{\mathbf{a}\in A'}(\|\mathbf{a}\|^2/2).$$

自 $R(A) = \frac{1}{\lambda}R(A')$ 我们从方程中得到

$$R(A) \le \frac{\log(|A|) + \lambda^2 \max_{\mathbf{a}\in A}(\|\mathbf{a}\|^2/2)}{\lambda m}.$$

设置 $\lambda = \sqrt{\text{^}2\log(|A|)/\max_{\mathbf{a}\in A}\|\mathbf{a}\|^2}$ 并重新排列项，我们得出我们的证明。

□

以下引理表明，将 A 与Lipschitz函数组合不会使Rademacher复杂度爆炸。证明归功于Kakade和Tewari。

LEMMA 26.9（紧缩引理）*For each $i \in [m]$, let $\phi_i : \mathbb{R} \to \mathbb{R}$ be a ρ-Lipschitz function, namely for all $\alpha, \beta \in \mathbb{R}$ we have $|\phi_i(\alpha) - \phi_i(\beta)| \le \rho\,|\alpha - \beta|$. For $\mathbf{a} \in \mathbb{R}^m$ let $\boldsymbol{\phi}(\mathbf{a})$ denote the vector $(\phi_1(a_1), \ldots, \phi_m(y_m))$. Let $\boldsymbol{\phi} \circ A = \{\boldsymbol{\phi}(\mathbf{a}) : a \in A\}$. Then,*

$$R(\boldsymbol{\phi} \circ A) \le \rho\, R(A).$$

Proof 为了简单起见，我们证明该引理对于 $\rho = 1$ 的情况。对于 $\rho \neq 1$ 的情况，通过定义 $\phi' = \frac{1}{\rho}\phi$ 并然后使用引理 26.6 将会得到。令 $A_i = \{(a_1, \ldots, a_{i-1}, \phi_i(a_i), a_{i+1}, \ldots, a_m) : \mathbf{a} \in A\}$。显然，只需证明对于任何集合 A 和所有 i，我们有 $R(A_i) \le R(A)$。不失一般性，我们将证明后一命题对于 $i = 1$，并且为了简化符号，我们省略了 ϕ_1 的下标。我们有

$$\begin{aligned}
mR(A_1) &= \mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\left[\sup_{\mathbf{a}\in A_1}\sum_{i=1}^{m}\sigma_i a_i\right] \\
&= \mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\left[\sup_{\mathbf{a}\in A}\sigma_1\phi(a_1) + \sum_{i=2}^{m}\sigma_i a_i\right] \\
&= \frac{1}{2}\mathop{\mathbb{E}}_{\sigma_2,\ldots,\sigma_m}\left[\sup_{\mathbf{a}\in A}\left(\phi(a_1) + \sum_{i=2}^{m}\sigma_i a_i\right) + \sup_{\mathbf{a}\in A}\left(-\phi(a_1) + \sum_{i=2}^{m}\sigma_i a_i\right)\right] \\
&= \frac{1}{2}\mathop{\mathbb{E}}_{\sigma_2,\ldots,\sigma_m}\left[\sup_{\mathbf{a},\mathbf{a}'\in A}\left(\phi(a_1) - \phi(a_1') + \sum_{i=2}^{m}\sigma_i a_i + \sum_{i=2}^{m}\sigma_i a_i'\right)\right] \\
&\le \frac{1}{2}\mathop{\mathbb{E}}_{\sigma_2,\ldots,\sigma_m}\left[\sup_{\mathbf{a},\mathbf{a}'\in A}\left(|a_1 - a_1'| + \sum_{i=2}^{m}\sigma_i a_i + \sum_{i=2}^{m}\sigma_i a_i'\right)\right], \qquad (26.12)
\end{aligned}$$

在最后一个不等式中，我们使用了 ϕ 是Lipschitz的假设。接下来，我们注意到，在前面的表达式中 $|a_1 - a_1'|$ 的绝对值可以

省略，因为 $\mathbf{a}$ 和 $\mathbf{a}'$ 都来自同一集合 A，并且上确界中的其余表达式在替换 $\mathbf{a}$ 和 $\mathbf{a}'$ 时不受影响。因此，

$$mR(A_1) \;\leq\; \frac{1}{2}\,\mathop{\mathbb{E}}_{\sigma_2,\ldots,\sigma_m}\left[\sup_{\mathbf{a},\mathbf{a}'\in A}\left(a_1 - a_1' + \sum_{i=2}^{m}\sigma_i a_i + \sum_{i=2}^{m}\sigma_i a_i'\right)\right]. \tag{26.13}$$

但是，使用与方程（26.12）相同的等式，很容易看出方程（26.13）的右侧正好等于 $m\,R(A)$，这也就完成了我们的证明。

□

26.2 Rademacher Complexity of Linear Classes

在这一节中，我们分析了线性类的Rademacher复杂度。为了简化推导，我们首先定义以下两个类：

$$\mathcal{H}_1 = \{\mathbf{x} \mapsto \langle \mathbf{w}, \mathbf{x}\rangle : \|\mathbf{w}\|_1 \leq 1\} \quad , \quad \mathcal{H}_2 = \{\mathbf{x} \mapsto \langle \mathbf{w}, \mathbf{x}\rangle : \|\mathbf{w}\|_2 \leq 1\}. \tag{26.14}$$

以下引理界定了 $\mathcal{H}_2$ 的 Rademacher 复杂度。我们允许 $\mathbf{x}_i$ 是任何希尔伯特空间中的向量（甚至是无限维的），并且这个界与希尔伯特空间的维度无关。当分析核方法时，这个性质变得很有用。

LEMMA 26.10 *Let $S = (\mathbf{x}_1, \ldots, \mathbf{x}_m)$ be vectors in a Hilbert space. Define: $\mathcal{H}_2 \circ S = \{(\langle \mathbf{w}, \mathbf{x}_1\rangle, \ldots, \langle \mathbf{w}, \mathbf{x}_m\rangle) : \|\mathbf{w}\|_2 \leq 1\}$. Then,*

$$R(\mathcal{H}_2 \circ S) \;\leq\; \frac{\max_i \|\mathbf{x}_i\|_2}{\sqrt{m}}.$$

Proof 使用柯西-施瓦茨不等式，我们知道对于任何向量 $\mathbf{w}, \mathbf{v}$，我们有 $\langle \mathbf{w}, \mathbf{v}\rangle \leq \|\mathbf{w}\|\,\|\mathbf{v}\|$。因此，

$$\begin{aligned} mR(\mathcal{H}_2 \circ S) &= \mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\left[\sup_{\mathbf{a}\in\mathcal{H}_2\circ S}\sum_{i=1}^{m}\sigma_i a_i\right] \\ &= \mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\left[\sup_{\mathbf{w}:\|\mathbf{w}\|\leq 1}\sum_{i=1}^{m}\sigma_i\langle \mathbf{w}, \mathbf{x}_i\rangle\right] \\ &= \mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\left[\sup_{\mathbf{w}:\|\mathbf{w}\|\leq 1}\langle \mathbf{w}, \sum_{i=1}^{m}\sigma_i\mathbf{x}_i\rangle\right] \\ &\leq \mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\left[\|\sum_{i=1}^{m}\sigma_i\mathbf{x}_i\|_2\right]. \end{aligned} \tag{26.15}$$

接下来，使用 Jensen 不等式，我们有

$$\mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\left[\left\|\sum_{i=1}^{m}\sigma_i\mathbf{x}_i\right\|_2\right] = \mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\left[\left(\left\|\sum_{i=1}^{m}\sigma_i\mathbf{x}_i\right\|_2^2\right)^{1/2}\right] \leq \left(\mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\left[\left\|\sum_{i=1}^{m}\sigma_i\mathbf{x}_i\right\|_2^2\right]\right)^{1/2} \tag{26.16}$$

最后，由于变量 $\sigma_1, \ldots, \sigma_m$ 是独立的，我们有

$$\begin{aligned}\mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\left[\|\sum_{i=1}^{m}\sigma_i\mathbf{x}_i\|_2^2\right] &= \mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\left[\sum_{i,j}\sigma_i\sigma_j\langle\mathbf{x}_i,\mathbf{x}_j\rangle\right] \\ &= \sum_{i\neq j}\langle\mathbf{x}_i,\mathbf{x}_j\rangle\mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\left[\sigma_i\sigma_j\right] + \sum_{i=1}^{m}\langle\mathbf{x}_i,\mathbf{x}_i\rangle\mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\left[\sigma_i^2\right] \\ &= \sum_{i=1}^{m}\|\mathbf{x}_i\|_2^2 \;\leq\; m\;\max_i\|\mathbf{x}_i\|_2^2.\end{aligned}$$

将此与方程（26.15）和方程（26.16）结合，我们得出证明。

□

接下来，我们界定了 $\mathcal{H}_1 \circ S$ 的 Rademacher 复杂度。

LEMMA 26.11 *Let $S = (\mathbf{x}_1, \ldots, \mathbf{x}_m)$ be vectors in $\mathbb{R}^n$. Then,*

$$R(\mathcal{H}_1 \circ S) \;\leq\; \max_i\|\mathbf{x}_i\|_\infty\sqrt{\frac{2\log(2n)}{m}}.$$

Proof 使用Holder不等式，我们知道对于任何向量$\mathbf{w}, \mathbf{v}$，我们有 $\langle\mathbf{w},\mathbf{v}\rangle \leq \|\mathbf{w}\|_1\|\mathbf{v}\|_\infty$。因此，

$$\begin{aligned}mR(\mathcal{H}_1 \circ S) &= \mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\left[\sup_{\mathbf{a}\in H_1\circ S}\sum_{i=1}^{m}\sigma_i a_i\right] \\ &= \mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\left[\sup_{\mathbf{w}:\|\mathbf{w}\|_1\leq 1}\sum_{i=1}^{m}\sigma_i\langle\mathbf{w},\mathbf{x}_i\rangle\right] \\ &= \mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\left[\sup_{\mathbf{w}:\|\mathbf{w}\|_1\leq 1}\langle\mathbf{w},\sum_{i=1}^{m}\sigma_i\mathbf{x}_i\rangle\right] \\ &\leq \mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\left[\|\sum_{i=1}^{m}\sigma_i\mathbf{x}_i\|_\infty\right]. \qquad (26.17)\end{aligned}$$

对于每个 $j \in [n]$，设 $\mathbf{v}_j = (x_{1,j}, \ldots, x_{m,j}) \in \mathbb{R}^m$。注意 $\|\mathbf{v}_j\|_2 \leq \sqrt{m}$ 最大 $_i\|\mathbf{x}_i\|_\infty$。设 $V = \{\mathbf{v}_1, \ldots, \mathbf{v}_n, -\mathbf{v}_1, \ldots, -\mathbf{v}_n\}$。方程（26.17）的右侧是 $m\,R(V)$。使用 Massart 引理（引理26.8）我们有

$$R(V) \leq \max_i\|\mathbf{x}_i\|_\infty\sqrt{2\,\log(2n)/m},$$

这总结了我们的证明。 □

26.3 Generalization Bounds for SVM

在本节中，我们使用Rademacher复杂度推导具有欧几里得范数约束的广义线性预测器的泛化界限。我们将展示这如何导致硬SVM和软SVM的泛化界限。

我们将考虑以下基于一般约束的公式化。设 $\mathcal{H} = \{\mathbf{w} : \|\mathbf{w}\|_2 \leq B\}$ 为我们的假设类，设 $Z = \mathcal{X} \times \mathcal{Y}$ 为示例域。假设损失函数 $\ell : \mathcal{H} \times Z \to \mathbb{R}$ 形式如下

$$\ell(\mathbf{w}, (\mathbf{x}, y)) = \phi(\langle \mathbf{w}, \mathbf{x} \rangle, y), \tag{26.18}$$

在 $\phi : \mathbb{R} \times \mathcal{Y} \to \mathbb{R}$ 是这样的，对于所有 $y \in \mathcal{Y}$，标量函数 $a \mapsto \phi(a, y)$ 是 ρ-Lipschitz。例如，铰链损失函数，$\ell(\mathbf{w}, (\mathbf{x}, y)) = \max\{0, 1 - y\langle \mathbf{w}, \mathbf{x} \rangle\}$，可以用 $\phi(a, y) = \max\{0, 1 - ya\}$ 写成，并注意 ϕ 对于所有 $y \in \{\pm 1\}$ 是 1-Lipschitz。另一个例子是绝对损失函数，$\ell(\mathbf{w}, (\mathbf{x}, y)) = |\langle \mathbf{w}, \mathbf{x} \rangle - y|$，可以用 $\phi(a, y) = |a - y|$ 写成，它对于所有 $y \in \mathbb{R}$ 也是 1-Lipschitz。

以下定理限制了 $\mathcal{H}$ 中所有预测器的泛化误差，使用它们的经验误差进行限制。

THEOREM 26.12 *Suppose that $\mathcal{D}$ is a distribution over $\mathcal{X} \times \mathcal{Y}$ such that with probability 1 we have that $\|\mathbf{x}\|_2 \leq R$. Let $\mathcal{H} = \{\mathbf{w} : \|\mathbf{w}\|_2 \leq B\}$ and let $\ell : \mathcal{H} \times Z \to \mathbb{R}$ be a loss function of the form given in Equation (26.18) such that for all $y \in \mathcal{Y}$, $a \mapsto \phi(a, y)$ is a ρ-Lipschitz function and such that $\max_{a \in [-BR, BR]} |\phi(a, y)| \leq c$. Then, for any $\delta \in (0, 1)$, with probability of at least $1 - \delta$ over the choice of an i.i.d. sample of size m,*

$$\forall \mathbf{w} \in \mathcal{H}, \quad L_{\mathcal{D}}(\mathbf{w}) \leq L_S(\mathbf{w}) + \frac{2\rho BR}{\sqrt{m}} + c\sqrt{\frac{2\ln(2/\delta)}{m}}.$$

Proof 让我们证明，以概率1，$R(F \circ S) \leq \rho BR/\sqrt{m}$，然后定理26.5将得出该定理。实际上，集合$F \circ S$可以写成

$$F \circ S = \{(\phi(\langle \mathbf{w}, \mathbf{x}_1 \rangle, y_1), \ldots, \phi(\langle \mathbf{w}, \mathbf{x}_m \rangle, y_m)) : \mathbf{w} \in \mathcal{H}\},$$

并且对 $R(F \circ S)$ 的约束可以通过结合引理26.9、引理26.10以及 $\|\mathbf{x}\|_2 \leq R$ 以概率1的假设直接得出。 □

我们接下来基于前一个定理推导出硬SVM的一般化界限。为了简化，我们不允许偏差项，并考虑硬SVM问题：

$$\operatorname*{argmin}_{\mathbf{w}} \|\mathbf{w}\|^2 \quad \text{s.t.} \quad \forall i,\ y_i \langle \mathbf{w}, \mathbf{x}_i \rangle \geq 1 \tag{26.19}$$

THEOREM 26.13 *Consider a distribution $\mathcal{D}$ over $\mathcal{X} \times \{\pm 1\}$ such that there exists some vector $\mathbf{w}^\star$ with $\mathbb{P}_{(\mathbf{x},y)\sim\mathcal{D}}[y\langle \mathbf{w}^\star, \mathbf{x} \rangle \geq 1] = 1$ and such that $\|\mathbf{x}\|_2 \leq R$ with probability 1. Let $\mathbf{w}_S$ be the output of Equation (26.19). Then, with probability of at least $1 - \delta$ over the choice of $S \sim \mathcal{D}^m$, we have that*

$$\mathbb{P}_{(\mathbf{x},y)\sim\mathcal{D}}[y \neq sign(\langle \mathbf{w}_S, \mathbf{x} \rangle)] \leq \frac{2\,R\,\|\mathbf{w}^\star\|}{\sqrt{m}} + (1 + R\,\|\mathbf{w}^\star\|)\sqrt{\frac{2\ln(2/\delta)}{m}}.$$

Proof 在整个证明过程中，令损失函数为斜坡损失（见第15.2.3节）。注意斜坡损失的范围是[0, 1]，并且它是一个1-Lipschitz函数。由于斜坡损失上界零一损失，因此我们有

$$\mathbb{P}_{(\mathbf{x},y)\sim\mathcal{D}}[y \neq \text{sign}(\langle \mathbf{w}_S, \mathbf{x}\rangle)] \leq L_{\mathcal{D}}(\mathbf{w}_S).$$

让 $B = \|\mathbf{w}^\star\|_2$ 并考虑集合 $\mathcal{H} = \{\mathbf{w}\colon \|\mathbf{w}\|_2 \leq B\}$。根据硬SVM的定义和我们对分布的假设，我们有 $\mathbf{w}_S \in \mathcal{H}$ 以概率1成立，且 $L_S(\mathbf{w}_S) =$ 为0。因此，根据定理26.12，我们得到

$$L_{\mathcal{D}}(\mathbf{w}_S) \leq L_S(\mathbf{w}_S) + \frac{2BR}{\sqrt{m}} + \sqrt{\frac{2\ln(2/\delta)}{m}}.$$

□

Remark 26.1 定理26.13表明，硬SVM的样本复杂度增长类似于 $\frac{R^2\|\mathbf{w}^\star\|^2}{\epsilon^2}$。通过更精细的分析和可分性假设，可以将界限提高到 $\frac{R^2\|\mathbf{w}^\star\|^2}{\epsilon}$ 的阶数。

前定理中的界依赖于 $\|\mathbf{w}^\star\|$，这是未知的。在以下内容中，我们推导出一个依赖于SVM输出范数的界；因此，它可以从训练集中本身计算得出。证明与结构风险最小化（SRM）界推导类似。

THEOREM 26.14 *Assume that the conditions of Theorem 26.13 hold. Then, with probability of at least* $1-\delta$ *over the choice of* $S \sim \mathcal{D}^m$*, we have that*

$$\mathbb{P}_{(\mathbf{x},y)\sim\mathcal{D}}[y \neq \mathit{sign}(\langle \mathbf{w}_S, \mathbf{x}\rangle)] \leq \frac{4R\|\mathbf{w}_S\|}{\sqrt{m}} + \sqrt{\frac{\ln(\frac{4\log_2(\|\mathbf{w}_S\|)}{\delta})}{m}}.$$

Proof 对于任何整数 {v*20}，令 {v*21} 2{v*22}，{v*23} : {v*24}，并令 {v*25} 。固定 {v*26}，然后使用定理26.12，我们有至少1 {v*27} 的概率

$$\forall \mathbf{w} \in \mathcal{H}_i, \quad L_{\mathcal{D}}(\mathbf{w}) \leq L_S(\mathbf{w}) + \frac{2B_iR}{\sqrt{m}} + \sqrt{\frac{2\ln(2/\delta_i)}{m}}$$

应用并集界并使用 $\sum_{i=1}^{\infty}\delta_i \leq \delta$，我们得到至少以概率 $1-\delta$ 这对所有 i 都成立。因此，对于所有 $\mathbf{w}$，如果我们让 $i = \lceil \log_2(\|\mathbf{w}\|)\rceil$，那么 $\mathbf{w} \in \mathcal{H}_i$，$B_i \leq 2\|\mathbf{w}\|$，和 $\frac{2}{\delta_i} = \frac{(2i)^2}{\delta} \leq \frac{(4\log_2(\|\mathbf{w}\|))^2}{\delta}$。因此，

$$\begin{aligned} L_{\mathcal{D}}(\mathbf{w}) &\leq L_S(\mathbf{w}) + \frac{2B_iR}{\sqrt{m}} + \sqrt{\frac{2\ln(2/\delta_i)}{m}} \\ &\leq L_S(\mathbf{w}) + \frac{4\|\mathbf{w}\|R}{\sqrt{m}} + \sqrt{\frac{4(\ln(4\log_2(\|\mathbf{w}\|)) + \ln(1/\delta))}{m}}. \end{aligned}$$

特别地，对于 $\mathbf{w}_S$ 成立，这结束了我们的证明 f. □

Remark 26.2 请注意，我们推导出的所有界限都不依赖于 $\mathbf{w}$ 的维度。当学习带有核的 SVM 时，利用这一性质，$\mathbf{w}$ 的维度可以非常大。

26.4 Generalization Bounds for Predictors with Low ℓ_1 Norm

在上一节中，我们推导了具有 ℓ_2-范数约束的线性预测器的泛化界限。在本节中，我们考虑以下一般的 ℓ_1-范数约束公式。设 $\mathcal{H} = \{\mathbf{w} : \|\mathbf{w}\|_1 \leq B\}$ 为我们的假设类，设 $Z = \mathcal{X} \times \mathcal{Y}$ 为示例域。假设损失函数 $\ell : \mathcal{H} \times Z \to \mathbb{R}$ 与方程（26.18）的形式相同，其中 $\phi : \mathbb{R} \times \mathcal{Y} \to \mathbb{R}$ 对其第一个参数是 ρ-Lipschitz。以下定理使用它们的经验误差对 $\mathcal{H}$ 中所有预测器的泛化误差进行界限。

THEOREM 26.15 *Suppose that $\mathcal{D}$ is a distribution over $\mathcal{X} \times \mathcal{Y}$ such that with probability 1 we have that $\|\mathbf{x}\|_\infty \leq R$. Let $\mathcal{H} = \{\mathbf{w} \in \mathbb{R}^d : \|\mathbf{w}\|_1 \leq B\}$ and let $\ell : \mathcal{H} \times Z \to \mathbb{R}$ be a loss function of the form given in Equation (26.18) such that for all $y \in \mathcal{Y}$, $a \mapsto \phi(a, y)$ is an ρ-Lipschitz function and such that $\max_{a \in [-BR, BR]} |\phi(a, y)| \leq c$. Then, for any $\delta \in (0, 1)$, with probability of at least $1 - \delta$ over the choice of an i.i.d. sample of size m,*

$$\forall \mathbf{w} \in \mathcal{H}, \quad L_{\mathcal{D}}(\mathbf{w}) \leq L_S(\mathbf{w}) + 2\rho BR\sqrt{\frac{2\log(2d)}{m}} + c\sqrt{\frac{2\ln(2/\delta)}{m}}.$$

Proof 证明与定理26.12的证明相同，而依赖于引理26.11而不是依赖于引理26.10。 □

有趣的是比较定理26.12和定理26.15中给出的两个界限。除了定理26.15中出现的额外log(d)因子外，这两个界限看起来很相似。然而，两个界限中的参数B, R有不同的含义。在定理26.12中，参数B对ℓ_2施加一个约束，而参数R捕捉实例的低ℓ_2-范数假设。相比之下，在定理26.15中，参数B对$\mathbf{w}$ (施加一个比ℓ_2约束更强的约束，而参数R捕捉实例的低ℓ_∞-范数假设（这比低ℓ_2-范数假设更弱）。因此，约束的选择应取决于我们对实例集的先验知识以及对良好预测器的先验假设。

26.5 Bibliographic Remarks

Rademacher复杂度用于界定一致收敛的使用归功于（Koltchinskii & Panchenko 2000，Bartlett & Mendelson 2001，Bartlett & Mendelson 2002）。有关进一步阅读，请参阅，例如，（Bousquet 2002，Boucheron, Bousquet & Lugosi 2005，Bartlett, Bousquet & Mendelson 2005）。

我们的集中引理证明归功于Kakade和Tewari的讲义。Kakade、Sridharan和Tewari（2008）为根据不同范数的假设推导线性类Rademacher复杂度的界限提供了一个统一的框架。

27 Covering Numbers

在这一章中，我们描述了另一种测量集合复杂性的方法，称为覆盖数。

27.1 Covering

DEFINITION 27.1（覆盖）设$A \subset \mathbb{R}^m$为一个向量集。我们称A在欧几里得度量下被集合A'以r方式覆盖，如果对于所有$\mathbf{a} \in A$，存在$\mathbf{a}' \in A'$满足$\|\mathbf{a} - \mathbf{a}'\| \leq r$。我们定义$N(r, A)$为覆盖$A$的最小$A'$的基数。

Example 27.1 (子空间) 假设$A \subset \mathbb{R}^m$，令$c = \max_{\mathbf{a} \in A} \|\mathbf{a}\|$，并假设$A$位于$\mathbb{R}^m$的$d$-维子空间中。那么，$N(r, A) \leq (2c\sqrt{d}/r)^d$。为了看到这一点，设$\mathbf{v}_1, \ldots, \mathbf{v}_d$为该子空间的一个正交基。那么，任何$\mathbf{a} \in A$都可以写成$\mathbf{a} = \sum_{i=1}^d \alpha_i \mathbf{v}_i$，其中$\|\boldsymbol{\alpha}\|_\infty \leq \|\boldsymbol{\alpha}\|_2 = \|\mathbf{a}\|_2 \leq c$。设$\epsilon \in \mathbb{R}$并考虑集合

$$A' = \left\{ \sum_{i=1}^d \alpha'_i \mathbf{v}_i : \forall i, \alpha'_i \in \{-c, -c+\epsilon, -c+2\epsilon, \ldots, c\} \right\}.$$

给定 $\mathbf{a} \in A$，使得 $\mathbf{a} = \sum_{i=1}^d \alpha_i \mathbf{v}_i$ 与 $\|\boldsymbol{\alpha}\|_\infty \leq c$，存在 $\mathbf{a}' \in A'$ 使得

$$\|\mathbf{a} - \mathbf{a}'\|^2 = \|\sum_i (\alpha'_i - \alpha_i)\mathbf{v}_i\|^2 \leq \epsilon^2 \sum_i \|\mathbf{v}_i\|^2 \leq \epsilon^2 \, d.$$

选择 $\epsilon = r/\sqrt{d}$；然后 $\|\mathbf{a} - \mathbf{a}'\| \leq r$，因此 A' 是 r 的一个覆盖。因此，

$$N(r, A) \leq |A'| = \left(\frac{2c}{\epsilon}\right)^d = \left(\frac{2c\sqrt{d}}{r}\right)^d.$$

27.1.1 Properties

以下引理直接来自定义。

LEMMA 27.2 *For any $A \subset \mathbb{R}^m$, scalar $c > 0$, and vector $\mathbf{a}_0 \in \mathbb{R}^m$, we have*

$$\forall r > 0, \quad N(r, \{c\,\mathbf{a} + \mathbf{a}_0 : \mathbf{a} \in A\}) \leq N(cr, A).$$

接下来，我们推导一个紧缩原理。

LEMMA 27.3 *For each $i \in [m]$, let $\phi_i : \mathbb{R} \to \mathbb{R}$ be a ρ-Lipschitz function; namely, for all $\alpha, \beta \in \mathbb{R}$ we have $|\phi_i(\alpha) - \phi_i(\beta)| \le \rho\,|\alpha - \beta|$. For $\mathbf{a} \in \mathbb{R}^m$ let $\boldsymbol{\phi}(\mathbf{a})$ denote the vector $(\phi_1(a_1), \ldots, \phi_m(a_m))$. Let $\boldsymbol{\phi} \circ A = \{\boldsymbol{\phi}(\mathbf{a}) : a \in A\}$. Then,*

$$N(\rho\, r, \boldsymbol{\phi} \circ A) \le N(r, A).$$

Proof 定义 $B = \boldsymbol{\phi} \circ A$。设 A' 是 r 的一个 A 覆盖，并定义 $B' = \boldsymbol{\phi} \circ A'$。然后，对于所有 $\mathbf{a} \in A$，存在 $\mathbf{a}' \in A'$ 满足 $\|\mathbf{a} - \mathbf{a}'\| \le r$。所以，

$$\|\boldsymbol{\phi}(\mathbf{a}) - \boldsymbol{\phi}(\mathbf{a}')\|^2 = \sum_i (\phi_i(a_i) - \phi_i(a_i'))^2 \le \rho^2 \sum_i (a_i - a_i')^2 \le (\rho r)^2.$$

因此，B' 是 B 的 $(\rho\, r)$-覆盖。 □

27.2 From Covering to Rademacher Complexity via Chaining

以下引理根据覆盖数 $N(r, A)$ 界定了 A 的 Rademacher 复杂度。这种技术称为 *Chaining*，归功于 Dudley。

LEMMA 27.4 *Let $c = \min_{\bar{\mathbf{a}}} \max_{\mathbf{a} \in A} \|\mathbf{a} - \bar{\mathbf{a}}\|$. Then, for any integer $M > 0$,*

$$R(A) \le \frac{c\, 2^{-M}}{\sqrt{m}} + \frac{6\, c}{m} \sum_{k=1}^{M} 2^{-k} \sqrt{\log(N(c\, 2^{-k}, A))}.$$

Proof 设 $\bar{\mathbf{a}}$ 是在 c 的定义中给出的目标函数的极小化器。基于引理26.6，我们可以在假设 $\bar{\mathbf{a}} = \mathbf{0}$ 的基础上分析Rademacher复杂度。

考虑集合 $B_0 = \{\mathbf{0}\}$ 并注意它是对 A 的 c-覆盖。设 $B_1, \ldots, B_M$ 为集合，使得每个 B_k 对应于 A 的最小 $(c\, 2^{-k})$-覆盖。设 $\mathbf{a}^* = = \text{argmax}_{\mathbf{a} \in A} \langle \boldsymbol{\sigma}, \mathbf{a} \rangle$ (，其中如果存在多个最大值，则以任意方式选择一个，如果不存在最大值，则选择 $\mathbf{a}^*$ 使得 $\langle \boldsymbol{\sigma}, \mathbf{a}^* \rangle$ 足够接近上确界)。注意 $\mathbf{a}^*$ 是 $\boldsymbol{\sigma}$ 的函数。对于每个 k，设 $\mathbf{b}^{(k)}$ 是 B_k (中 $\mathbf{a}^*$ 的最近邻，因此 $\mathbf{b}^{(k)}$ 也是 $\boldsymbol{\sigma}$) 的函数。使用三角不等式，

$$\|\mathbf{b}^{(k)} - \mathbf{b}^{(k-1)}\| \le \|\mathbf{b}^{(k)} - \mathbf{a}^*\| + \|\mathbf{a}^* - \mathbf{b}^{(k-1)}\| \le c\,(2^{-k} + 2^{-(k-1)}) = 3\, c\, 2^{-k}.$$

对于每个 k 定义集合

$$\hat{B}_k = \{(\mathbf{a} - \mathbf{a}') : \mathbf{a} \in B_k, \mathbf{a}' \in B_{k-1}, \|\mathbf{a} - \mathbf{a}'\| \le 3\, c\, 2^{-k}\}.$$

我们现在可以写出

$$\begin{aligned} R(A) &= \frac{1}{m}\,\mathbb{E}\langle \boldsymbol{\sigma}, \mathbf{a}^* \rangle \\ &= \frac{1}{m}\,\mathbb{E}\left[\langle \boldsymbol{\sigma}, \mathbf{a}^* - \mathbf{b}^{(M)} \rangle + \sum_{k=1}^{M} \langle \boldsymbol{\sigma}, \mathbf{b}^{(k)} - \mathbf{b}^{(k-1)} \rangle \right] \\ &\leq \frac{1}{m}\,\mathbb{E}\left[\|\sigma\|\, \|\mathbf{a}^* - \mathbf{b}^{(M)}\| \right] + \sum_{k=1}^{M} \frac{1}{m}\,\mathbb{E}\left[\sup_{\mathbf{a}\in \hat{B}_k} \langle \boldsymbol{\sigma}, \mathbf{a} \rangle \right]. \end{aligned}$$

由于 $\|\boldsymbol{\sigma}\| = \sqrt{m}$ 和 $\|\mathbf{a}^* - \mathbf{b}^{(M)}\| \leq c\,2^{-M}$，第一个加数最多为$\frac{c}{\sqrt{m}}\,2^{-M}$。此外，根据 Massart 引理,

$$\frac{1}{m}\,\mathbb{E} \sup_{\mathbf{a}\in \hat{B}_k} \langle \boldsymbol{\sigma}, \mathbf{a} \rangle \leq 3\,c\,2^{-k}\,\frac{\sqrt{2\log(N(c\,2^{-k}, A)^2)}}{m} = 6\,c\,2^{-k}\,\frac{\sqrt{\log(N(c\,2^{-k}, A))}}{m}.$$

因此,

$$R(A) \leq \frac{c\,2^{-M}}{\sqrt{m}} + \frac{6c}{m} \sum_{k=1}^{M} 2^{-k} \sqrt{\log(N(c2^{-k}, A))}.$$

□

作为推论，我们得到以下结果:

LEMMA 27.5 *Assume that there are $\alpha, \beta > 0$ such that for any $k \geq 1$ we have*

$$\sqrt{\log(N(c2^{-k}, A))} \leq \alpha + \beta k.$$

Then,

$$R(A) \leq \frac{6c}{m}\,(\alpha + 2\beta)\,.$$

Proof 从引理27.4得出，通过取 $M \to \infty$ 并注意到 $\sum_{k=1}^{\infty} 2^{-k} = 1$ 和 $\sum_{k=1}^{\infty} k2^{-k} = 2$。 □

Example 27.2 考虑一个位于 $\mathbb{R}^m$ 的 d 维子空间中的集合 A，并且满足 $c = \max_{\mathbf{a}\in A} \|\mathbf{a}\|$。我们已经证明了 $N(r, A) \leq \left(\frac{2c\sqrt{d}}{r}\right)^d$。因此，对于任何 k,

$$\begin{aligned} \sqrt{\log(N(c2^{-k}, A))} &\leq \sqrt{d \log\left(2^{k+1}\sqrt{d}\right)} \\ &\leq \sqrt{d\log(2\sqrt{d})} + \sqrt{k\,d} \\ &\leq \sqrt{d\log(2\sqrt{d})} + \sqrt{d}\,k. \end{aligned}$$

因此引理27.5得出

$$R(A) \;\leq\; \frac{6c}{m}\left(\sqrt{d\log(2\sqrt{d})} + 2\sqrt{d}\right) \;=\; O\left(\frac{c\sqrt{d\log(d)}}{m}\right).$$

27.3 Bibliographic Remarks

链式技术归功于Dudley（1987年）。对于覆盖数以及其他可以用来界定均匀收敛速率的复杂度度量的大量研究，我们建议读者参考（Anthony & Bartlet 1999）。

28 Proof of the Fundamental Theorem of Learning Theory

在这一章中，我们证明了第6章中的定理6.8。我们提醒读者定理的条件，这些条件将贯穿本章：$\mathcal{H}$ 是从域 $\mathcal{X}$ 到 $\{0,1\}$ 的函数假设类，损失函数是 $0-1$ 损失，并且 VCdim($\mathcal{H}$) $= d < \infty$。

我们将证明可实现情况和无偏情况的上界，并将证明无偏情况的下界。可实现情况的下界留作练习。

28.1 The Upper Bound for the Agnostic Case

对于上界，我们需要证明存在 C 使得 $\mathcal{H}$ 是具有样本复杂度的无偏PAC可学习的。

$$m_{\mathcal{H}}(\epsilon,\delta) \le C\frac{d+\ln(1/\delta)}{\epsilon^2}.$$

我们将证明略微宽松的界限：

$$m_{\mathcal{H}}(\epsilon,\delta) \le C\frac{d\log(d/\epsilon)+\ln(1/\delta)}{\epsilon^2}. \tag{28.1}$$

定理陈述中的更紧界需要更复杂的证明，其中应使用称为“链式”的技术对Rademacher复杂性进行更仔细的分析。这超出了本书的范围。

为了证明方程（28.1），只需证明应用具有样本大小 {v*} 的 ERM 即可。

$$m \ge 4\frac{32d}{\epsilon^2}\cdot\log\left(\frac{64d}{\epsilon^2}\right)+\frac{8}{\epsilon^2}\cdot(8d\log(e/d)+2\log(4/\delta))$$

产生一个 ϵ,δ-学习者用于 $\mathcal{H}$。我们基于定理26.5证明此结果。

设$(\mathbf{x}_1,y_1),\ldots,(\mathbf{x}_m,y_m)$为一个分类训练集。回忆Sauer-Shelah引理告诉我们，如果VCdim($\mathcal{H}$) $= d$，那么

$$|\{(h(\mathbf{x}_1),\ldots,h(\mathbf{x}_m)) : h\in\mathcal{H}\}| \;\le\; \left(\frac{e\,m}{d}\right)^d.$$

表示 $A=\{(1_{[h(\mathbf{x}_1)\neq y_1]},\ldots,1_{[h(\mathbf{x}_m)\neq y_m]}) : h\in\mathcal{H}\}$。这显然意味着

$$|A| \;\le\; \left(\frac{e\,m}{d}\right)^d.$$

结合26.8引理，我们得到Rademacher复杂度的以下界限：

$$R(A) \leq \sqrt{\frac{2d\log(em/d)}{m}}.$$

使用定理26.5，我们得到，至少以概率$1-\delta$，对于每个$h \in \mathcal{H}$，我们有

$$L_{\mathcal{D}}(h) - L_S(h) \leq \sqrt{\frac{8d\log(em/d)}{m}} + \sqrt{\frac{2\log(2/\delta)}{m}}.$$

重复之前的论点，对于零一损失取负，并应用并集界，我们得到至少以概率1 $-\delta$，对于每个$h \in \mathcal{H}$都成立。

$$\begin{aligned}|L_{\mathcal{D}}(h) - L_S(h)| &\leq \sqrt{\frac{8d\log(em/d)}{m}} + \sqrt{\frac{2\log(4/\delta)}{m}} \\ &\leq 2\sqrt{\frac{8d\log(em/d) + 2\log(4/\delta)}{m}}.\end{aligned}$$

为确保这小于ϵ，我们需要

$$m \geq \frac{4}{\epsilon^2} \cdot \left(8d\log(m) + 8d\log(e/d) + 2\log(4/\delta)\right).$$

使用引理A.2，不等式成立的一个充分条件是

$$m \geq 4\frac{32d}{\epsilon^2} \cdot \log\left(\frac{64d}{\epsilon^2}\right) + \frac{8}{\epsilon^2} \cdot \left(8d\log(e/d) + 2\log(4/\delta)\right).$$

28.2 The Lower Bound for the Agnostic Case

这里，我们证明存在C使得$\mathcal{H}$是具有样本复杂度的无监督PAC可学习的。

$$m_{\mathcal{H}}(\epsilon, \delta) \geq C\frac{d + \ln(1/\delta)}{\epsilon^2}.$$

我们将分两部分证明下界。首先，我们将证明$m(\epsilon, \delta) \geq 0.5\log(1/(4\delta))/\epsilon^2$，其次，我们将证明对于每一个$\delta \leq 1/8$，都有$m(\epsilon, \delta) \geq 8d/\epsilon^2$。这两个界限将完成证明。

28.2.1 Showing That $m(\epsilon, \delta) \geq 0.5\log(1/(4\delta))/\epsilon^2$

首先，我们证明对于任意的$\epsilon < 1/\sqrt{2}$和任意的$\delta \in (0, 1)$，我们有$m(\epsilon, \delta) \geq 0.5\log(1/(4\delta))/\epsilon^2$。为了做到这一点，我们证明对于$m \leq 0.5\log(1/(4\delta))/\epsilon^2$，$\mathcal{H}$是不可学习的。

选择一个被$\mathcal{H}$粉碎的例子。也就是说，让c成为这样一个例子

存在 $h_+, h_- \in \mathcal{H}$，使得 $h_+(c) = 1$ 和 $h_-(c) = -1$。定义两个分布，$\mathcal{D}_+$ 和 $\mathcal{D}_-$，使得对于 $b \in \{\pm 1\}$，我们有

$$\mathcal{D}_b(\{(x,y)\}) = \begin{cases} \frac{1+yb\epsilon}{2} & \text{if } x = c \\ 0 & \text{otherwise.} \end{cases}$$

这意味着所有分布质量都集中在两个例子$(c,1)$和$(c,-1)$上，其中(c,b)的概率是$\frac{1+b\epsilon}{2}$，$(c,-b)$的概率是$\frac{1-b\epsilon}{2}$。

设 A 为一个任意算法。从 $\mathcal{D}_b$ 中采样的任何训练集具有形式 $S = (c,y_1),\ldots,(c,y_m)$。因此，它完全由向量 $\mathbf{y} = (y_1,\ldots,y_m) \in \{\pm 1\}^m$ 描述。在接收到训练集 S 后，算法 A 返回一个假设 $h\colon \mathcal{X} \to \{\pm 1\}$。由于 A 关于 $\mathcal{D}_b$ 的错误仅取决于 $h(c)$，我们可以将 A 视为从 $\{\pm 1\}^m$ 到 $\{\pm 1\}$ 的映射。因此，我们用 $A(\mathbf{y})$ 表示 $\{\pm 1\}$ 中对应于 $h(c)$ 预测的值，其中 h 是 A 在接收到训练集 $S = (c,y_1),\ldots,(c,y_m)$ 后输出的假设。

请注意，对于任何假设 h，我们有

$$L_{\mathcal{D}_b}(h) = \frac{1 - h(c)b\epsilon}{2}.$$

特别地，贝叶斯最优假设是 h_b 并且

$$L_{\mathcal{D}_b}(A(\mathbf{y})) - L_{\mathcal{D}_b}(h_b) = \frac{1 - A(\mathbf{y})b\epsilon}{2} - \frac{1-\epsilon}{2} = \begin{cases} \epsilon & \text{if } A(\mathbf{y}) \neq b \\ 0 & \text{otherwise.} \end{cases}$$

修复 A。对于 $b \in \{\pm 1\}$，令 $Y^b = \{\mathbf{y} \in \{0,1\}^m : A(\mathbf{y}) \neq b\}$。分布 $\mathcal{D}_b$ 在 $\{\pm 1\}^m$ 上诱导一个概率 P_b。因此，

$$\mathbb{P}\left[L_{\mathcal{D}_b}(A(\mathbf{y})) - L_{\mathcal{D}_b}(h_b) = \epsilon\right] = \mathcal{D}_b(Y^b) = \sum_{\mathbf{y}} P_b[\mathbf{y}]\mathbb{1}_{[A(\mathbf{y})\neq b]}.$$

表示 $N^+ = \{\mathbf{y} : |\{i : y_i = 1\}| \geq m/2\}$ 和 $N^- = \{\pm 1\}^m \setminus N^+$。注意，对于任何 $\mathbf{y} \in N^+$，我们有 $P_+[\mathbf{y}] \geq P_-[\mathbf{y}]$，对于任何 $\mathbf{y} \in N^-$，我们有 $P_-[\mathbf{y}] \geq P_+[\mathbf{y}]$。

因此，

$$
\begin{aligned}
&\max_{b\in\{\pm 1\}} \mathbb{P}\left[L_{\mathcal{D}_b}(A(\mathbf{y})) - L_{\mathcal{D}_b}(h_b) = \epsilon\right] \\
&= \max_{b\in\{\pm 1\}} \sum_{\mathbf{y}} P_b[\mathbf{y}] \mathbb{1}_{[A(\mathbf{y})\neq b]} \\
&\geq \frac{1}{2}\sum_{\mathbf{y}} P_+[\mathbf{y}]\mathbb{1}_{[A(\mathbf{y})\neq +]} + \frac{1}{2}\sum_{\mathbf{y}} P_-[\mathbf{y}]\mathbb{1}_{[A(\mathbf{y})\neq -]} \\
&= \frac{1}{2}\sum_{\mathbf{y}\in N^+} (P_+[\mathbf{y}]\mathbb{1}_{[A(\mathbf{y})\neq +]} + P_-[\mathbf{y}]\mathbb{1}_{[A(\mathbf{y})\neq -]}) + \frac{1}{2}\sum_{\mathbf{y}\in N^-} (P_+[\mathbf{y}]\mathbb{1}_{[A(\mathbf{y})\neq +]} + P_-[\mathbf{y}]\mathbb{1}_{[A(\mathbf{y})\neq -]}) \\
&\geq \frac{1}{2}\sum_{\mathbf{y}\in N^+} (P_-[\mathbf{y}]\mathbb{1}_{[A(\mathbf{y})\neq +]} + P_-[\mathbf{y}]\mathbb{1}_{[A(\mathbf{y})\neq -]}) + \frac{1}{2}\sum_{\mathbf{y}\in N^-} (P_+[\mathbf{y}]\mathbb{1}_{[A(\mathbf{y})\neq +]} + P_+[\mathbf{y}]\mathbb{1}_{[A(\mathbf{y})\neq -]}) \\
&= \frac{1}{2}\sum_{\mathbf{y}\in N^+} P_-[\mathbf{y}] + \frac{1}{2}\sum_{\mathbf{y}\in N^-} P_+[\mathbf{y}] \ .
\end{aligned}
$$

下一条注意 $\sum_{\mathbf{y}\in N^+} P_-[\mathbf{y}] = \sum_{\mathbf{y}\in N^-} P_+[\mathbf{y}]$，并且这两个值都是二项式 $(m, (1 - \epsilon)/2)$ 随机变量值大于 $m/2$ 的概率。使用引理 B.11，这个概率的下限为

$$
\frac{1}{2}\left(1 - \sqrt{1 - \exp(-m\epsilon^2/(1-\epsilon^2))}\right) \ \geq \ \frac{1}{2}\left(1 - \sqrt{1 - \exp(-2m\epsilon^2)}\right),
$$

在本文中，我们使用了假设 $\epsilon^2 \leq 1/2$。因此，如果 $m \leq 0.5\log(1/(4\delta))/\epsilon^2$，那么存在 b 使得

$$
\begin{aligned}
&\mathbb{P}\left[L_{\mathcal{D}_b}(A(\mathbf{y})) - L_{\mathcal{D}_b}(h_b) = \epsilon\right] \\
&\geq \frac{1}{2}\left(1 - \sqrt{1 - \sqrt{4\delta}}\right) \geq \delta,
\end{aligned}
$$

在标准代数操作后得到最后一个不等式。这完成了我们的证明。

28.2.2 Showing That $m(\epsilon, 1/8) \geq 8d/\epsilon^2$

我们现在将证明对于每一个 $\epsilon < 1/(8\sqrt{2})$，我们都有 $m(\epsilon, \delta) \geq \frac{8d}{\epsilon^2}$。

让 $\rho = 8\epsilon$ 并注意 $\rho \in (0, 1/\sqrt{2})$。我们将按照以下方式构造一个分布族。首先，设$C = \{c_1, \ldots, c_d\}$为一个由$\mathcal{H}$划分的d实例集合。其次，对于每个向量$(b_1, \ldots, b_d) \in \{\pm 1\}^d$，定义一个分布$\mathcal{D}_b$，使得

$$
\mathcal{D}_b(\{(x, y)\}) = \begin{cases} \frac{1}{d}\cdot\frac{1+yb_i\rho}{2} & \text{if } \exists i : x = c_i \\ 0 & \text{otherwise.} \end{cases}
$$

这是，根据 $\mathcal{D}_b$ 抽样一个示例，我们首先随机均匀地抽样一个元素 $c_i \in C$，然后将标签设置为 b_i，概率为 $(1 + \rho)/\hat{}2$ 或 $-b_i$，概率为 $(1 - \rho)/\hat{}2$。

它很容易验证，对于 $\mathcal{D}_b$ 的贝叶斯最优预测器是假设

$h \in \mathcal{H}$ 如此，对于所有 $i \in [d]$，有 $h(c_i) = b_i$，其误差为 $\frac{1-\rho}{2}$。此外，对于任何其他函数 f: $\mathcal{X} \to \{\pm 1\}$，很容易验证

$$L_{\mathcal{D}_b}(f) = \frac{1+\rho}{2} \cdot \frac{|\{i \in [d] : f(c_i) \neq b_i\}|}{d} + \frac{1-\rho}{2} \cdot \frac{|\{i \in [d] : f(c_i) = b_i\}|}{d}.$$

因此，

$$L_{\mathcal{D}_b}(f) - \min_{h \in \mathcal{H}} L_{\mathcal{D}_b}(h) = \rho \cdot \frac{|\{i \in [d] : f(c_i) \neq b_i\}|}{d}. \tag{28.2}$$

接下来，修复一些学习算法 A。正如在“无免费午餐定理”的证明中，我们有

$$\max_{\mathcal{D}_b : b \in \{\pm 1\}^d} \mathop{\mathbb{E}}_{S \sim \mathcal{D}_b^m} \left[L_{\mathcal{D}_b}(A(S)) - \min_{h \in \mathcal{H}} L_{\mathcal{D}_b}(h) \right] \tag{28.3}$$

$$\geq \mathop{\mathbb{E}}_{\mathcal{D}_b : b \sim U(\{\pm 1\}^d)} \mathop{\mathbb{E}}_{S \sim \mathcal{D}_b^m} \left[L_{\mathcal{D}_b}(A(S)) - \min_{h \in \mathcal{H}} L_{\mathcal{D}_b}(h) \right] \tag{28.4}$$

$$= \mathop{\mathbb{E}}_{\mathcal{D}_b : b \sim U(\{\pm 1\}^d)} \mathop{\mathbb{E}}_{S \sim \mathcal{D}_b^m} \left[\rho \cdot \frac{|\{i \in [d] : A(S)(c_i) \neq b_i|}{d} \right] \tag{28.5}$$

$$= \frac{\rho}{d} \sum_{i=1}^{d} \mathop{\mathbb{E}}_{\mathcal{D}_b : b \sim U(\{\pm 1\}^d)} \mathop{\mathbb{E}}_{S \sim \mathcal{D}_b^m} \mathbb{1}_{[A(S)(c_i) \neq b_i]}, \tag{28.6}$$

在第一个等式由方程（28.2）得出。此外，使用 $\mathcal{D}_b$ 的定义，为了采样 $S \sim \mathcal{D}_b$，我们首先可以采样 $(j_1, \ldots, j_m) \sim U([d])^m$，设置 $x_r = c_{j_i}$，最后采样 y_r 使得 $\mathbb{P}[y_r = b_{j_i}] = (1+\rho)/2$。让我们简化符号，并使用 $y \sim b$ 表示根据 $\mathbb{P}[y = b] = (1+\rho)/2$ 进行采样。因此，方程（28.6）的右侧等于

$$\frac{\rho}{d} \sum_{i=1}^{d} \mathop{\mathbb{E}}_{j \sim U([d])^m} \mathop{\mathbb{E}}_{b \sim U(\{\pm 1\}^d)} \mathop{\mathbb{E}}_{\forall r, y_r \sim b_{j_r}} \mathbb{1}_{[A(S)(c_i) \neq b_i]}. \tag{28.7}$$

我们现在分两步进行。首先，我们证明在所有学习算法中，A，即最小化方程（28.7）（从而也最小化方程（28.4））的算法是最大似然学习规则，记为A_{ML}。形式上，对于每个i，$A_{ML}(S)(c_i)$是集合$\{y_r\colon\ r \in [m], x_r = c_i\}$中的多数投票。其次，我们为$A_{ML}$降低方程（28.7）的下界。

LEMMA 28.1 *Among all algorithms, Equation (28.4) is minimized for A being the Maximum-Likelihood algorithm, A_{ML}, defined as*

$$\forall i, \quad A_{ML}(S)(c_i) = sign\left(\sum_{r : x_r = c_i} y_r \right).$$

Proof 修复一些 $j \in [d]^m$。注意，给定 j 和 $y \in \{\pm 1\}^m$，训练集 S 是完全确定的。因此，我们可以写 $A(j, y)$ 而不是 $A(S)$。让我们也固定 $i \in [d]$。用 $b^{\neg i}$ 表示序列 $(b_1, \ldots, b_{i-1}, b_{i+1}, \ldots, b_m)$。此外，对于任何

$y \in \{\pm 1\}^m$，令 y^I 表示与索引 $j_r = i$ 对应的 y 的元素，令 $y^{\neg I}$ 为 y 的其余元素。我们有

$$\begin{aligned}
&\mathop{\mathbb{E}}_{b\sim U(\{\pm 1\}^d)} \mathop{\mathbb{E}}_{\forall r, y_r \sim b_{j_r}} \mathbb{1}_{[A(S)(c_i)\neq b_i]} \\
&= \frac{1}{2} \sum_{b_i \in \{\pm 1\}} \mathop{\mathbb{E}}_{b^{\neg i}\sim U(\{\pm 1\}^{d-1})} \sum_y P[y|b^{\neg i}, b_i] \mathbb{1}_{[A(j,y)(c_i)\neq b_i]} \\
&= \mathop{\mathbb{E}}_{b^{\neg i}\sim U(\{\pm 1\}^{d-1})} \sum_{y^{\neg I}} P[y^{\neg I}|b^{\neg i}] \frac{1}{2} \sum_{y^I} \left(\sum_{b_i \in \{\pm 1\}} P[y^I|b_i] \mathbb{1}_{[A(j,y)(c_i)\neq b_i]} \right).
\end{aligned}$$

括号内的和在 $A(j,y)(c_i)$ 是 $P[y^I|b_i]$ 在 $b_i \in \{\pm 1\}$ 上的最大化者时最小化，这正好是最大似然规则。对所有的 i 重复相同的论点，我们得出我们的证明。

□

修复 i。对于每个 j，令 $n_i(j) = \{|t : j_t = i|\}$ 为实例是 c_i 的实例数量。对于最大似然规则，我们有以下数量

$$\mathop{\mathbb{E}}_{b\sim U(\{\pm 1\}^d)} \mathop{\mathbb{E}}_{\forall r, y_r \sim b_{j_r}} \mathbb{1}_{[A_{ML}(S)(c_i)\neq b_i]}$$

这是二项式$(n_i(j), (1-\rho)/2)$随机变量大于$n_i(j)/2$的概率。使用引理B.11和假设 $\rho^2 \leq 1/2$，我们得到：

$$P[B \geq n_i(j)/2] \geq \frac{1}{2}\left(1 - \sqrt{1 - e^{-2n_i(j)\rho^2}}\right).$$

我们已经证明了

$$\begin{aligned}
&\frac{\rho}{d} \sum_{i=1}^{d} \mathop{\mathbb{E}}_{j\sim U([d])^m} \mathop{\mathbb{E}}_{b\sim U(\{\pm 1\}^d)} \mathop{\mathbb{E}}_{\forall r, y_r \sim b_{j_r}} \mathbb{1}_{[A(S)(c_i)\neq b_i]} \\
&\geq \frac{\rho}{2d} \sum_{i=1}^{d} \mathop{\mathbb{E}}_{j\sim U([d])^m} \left(1 - \sqrt{1 - e^{-2\rho^2 n_i(j)}}\right) \\
&\geq \frac{\rho}{2d} \sum_{i=1}^{d} \mathop{\mathbb{E}}_{j\sim U([d])^m} \left(1 - \sqrt{2\rho^2 n_i(j)}\right),
\end{aligned}$$

在最后一个不等式中，我们使用了不等式$1 - e^{-a} \leq a$。

由于平方根函数是凹函数，我们可以应用 Jensen 不等式，从而得到上述表达式有下界{v*}

$$\begin{aligned}
&\geq \frac{\rho}{2d} \sum_{i=1}^{d} \left(1 - \sqrt{2\rho^2 \mathop{\mathbb{E}}_{j\sim U([d])^m} n_i(j)}\right) \\
&= \frac{\rho}{2d} \sum_{i=1}^{d} \left(1 - \sqrt{2\rho^2 m/d}\right) \\
&= \frac{\rho}{2}\left(1 - \sqrt{2\rho^2 m/d}\right).
\end{aligned}$$

只要 $m < \frac{d}{8\rho^2}$，这个项就会大于 $\rho/4$。

总结来说，我们已经证明如果 $m < \frac{d}{8\rho^2}$，那么对于任何算法都存在一个分布，使得

$$\mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}\left[L_{\mathcal{D}}(A(S)) - \min_{h\in\mathcal{H}} L_{\mathcal{D}}(h)\right] \geq \rho/4.$$

最后，令 $\Delta = \frac{1}{\rho}(L_{\mathcal{D}}(A(S)) - \min_{h\in\mathcal{H}} L_{\mathcal{D}}(h))$ 并注意 $\Delta \in [0,1]$（参见方程（28.5））。因此，使用引理B.1，我们得到

$$\begin{aligned}\mathbb{P}[L_{\mathcal{D}}(A(S)) - \min_{h\in\mathcal{H}} L_{\mathcal{D}}(h) > \epsilon] = \mathbb{P}\left[\Delta > \frac{\epsilon}{\rho}\right] &\geq \mathbb{E}[\Delta] - \frac{\epsilon}{\rho} \\ &\geq \frac{1}{4} - \frac{\epsilon}{\rho}.\end{aligned}$$

选择 $\rho = 8\epsilon$，我们得出结论：如果 $m < \frac{d}{512\,\epsilon^2}$，那么至少以 $1/8$ 的概率我们将有 $L_{\mathcal{D}}(A(S)) - \min_{h\in\mathcal{H}} L_{\mathcal{D}}(h) \geq \epsilon$。

28.3 The Upper Bound for the Realizable Case

这里我们证明存在 C 使得 $\mathcal{H}$ 具有样本复杂度的 PAC 可学习性

$$m_{\mathcal{H}}(\epsilon,\delta) \leq C\frac{d\ln(1/\epsilon) + \ln(1/\delta)}{\epsilon}.$$

我们通过展示对于 $m \geq C\frac{d\ln(1/\epsilon)+\ln(1/\delta)}{\epsilon}$，$\mathcal{H}$ 可以使用 ERM 规则进行学习来这样做。我们基于 ϵ-网的概念来证明这个论断。

DEFINITION 28.2 ({v*}-net) 设 $\mathcal{X}$ 为一个域。$S \subset \mathcal{X}$ 是相对于在 $\mathcal{X}$ 上的分布 $\mathcal{D}$，对 $\mathcal{H} \subset 2^{\mathcal{X}}$ 的一个 ϵ-net，如果

$$\forall h \in \mathcal{H}:\ \ \mathcal{D}(h) \geq \epsilon \ \Rightarrow\ h \cap S \neq \emptyset.$$

THEOREM 28.3 *Let* $\mathcal{H} \subset 2^{\mathcal{X}}$ *with* VCdim($\mathcal{H}$) $= d$. *Fix* $\epsilon \in (0,1)$, $\delta \in (0,1/4)$ *and let*

$$m \geq \frac{8}{\epsilon}\left(2d\log\left(\frac{16e}{\epsilon}\right) + \log\left(\frac{2}{\delta}\right)\right).$$

Then, with probability of at least $1-\delta$ *over a choice of* $S \sim \mathcal{D}^m$ *we have that* S *is an* ϵ*-net for* $\mathcal{H}$.

Proof 让

$$B = \{S \subset \mathcal{X} : |S| = m,\ \exists h \in \mathcal{H}, \mathcal{D}(h) \geq \epsilon, h \cap S = \emptyset\}$$

集合是所有不是 ϵ-网集合的集合。我们需要界定 $\mathbb{P}[S \in B]$。定义

$$B' = \{(S,T) \subset \mathcal{X} : |S| = |T| = m,\ \exists h \in \mathcal{H}, \mathcal{D}(h) \geq \epsilon, h \cap S = \emptyset,\ |T \cap h| > \tfrac{\epsilon m}{2}\}.$$

Claim 1
$\mathbb{P}[S \in B] \leq 2\,\mathbb{P}[(S,T) \in B']$.
*Proof of Claim 1*由于 S 和 T 是独立选择的，因此我们可以写出

$$\mathbb{P}[(S,T) \in B'] = \mathop{\mathbb{E}}_{(S,T)\sim\mathcal{D}^{2m}}\left[\mathbb{1}_{[(S,T)\in B']}\right] = \mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}\left[\mathop{\mathbb{E}}_{T\sim\mathcal{D}^m}\left[\mathbb{1}_{[(S,T)\in B']}\right]\right].$$

注意 $(S,T) \in B'$ 意味着 $S \in B$，因此 $1_{[(S,T)\in B']} = 1_{[(S,T)\in B']}\,1_{[S\in B]}$，这给出

$$\begin{aligned}\mathbb{P}[(S,T) \in B'] &= \mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}\mathop{\mathbb{E}}_{T\sim\mathcal{D}^m}\mathbb{1}_{[(S,T)\in B']}\,\mathbb{1}_{[S\in B]} \\ &= \mathop{\mathbb{E}}_{S\sim\mathcal{D}^m}\mathbb{1}_{[S\in B]}\mathop{\mathbb{E}}_{T\sim\mathcal{D}^m}\mathbb{1}_{[(S,T)\in B']}.\end{aligned}$$

修复一些 S。然后，要么是 $1_{[S\in B]} = 0$ 或者 $S \in B$ 然后是 $\exists h_S$ 以便 $\mathcal{D}(h_S) \geq \epsilon$ 和 $|h_S \cap S| = 0$。因此，$(S,T) \in B'$ 的一个充分条件是 $|T \cap h_S| > \frac{\epsilon m}{2}$。因此，每当 $S \in B$ 我们都有

$$\mathop{\mathbb{E}}_{T\sim\mathcal{D}^m}\mathbb{1}_{[(S,T)\in B']} \;\geq\; \mathop{\mathbb{P}}_{T\sim\mathcal{D}^m}[|T \cap h_S| > \tfrac{\epsilon m}{2}].$$

但是，由于我们现在假设 $S \in B$，我们知道 $\mathcal{D}(h_S) = \rho \geq \epsilon$。因此，$|T \cap h_S|$ 是一个参数为 ρ (成功概率的单次尝试) 和 m (尝试次数) 的二项随机变量。切比雪夫不等式意味着

$$\mathbb{P}[|T\cap h_S| \leq \tfrac{\rho m}{2}] \leq e^{-\frac{2}{m\rho}(m\rho - m\rho/2)^2} = e^{-m\rho/2} \leq e^{-m\epsilon/2} \leq e^{-d\log(1/\delta)/2} = \delta^{d/2} \leq 1/2.$$

Thus,Thus,

$$\mathbb{P}[|T \cap h_S| > \tfrac{\epsilon m}{2}] \;=\; 1 - \mathbb{P}[|T \cap h_S| \leq \tfrac{\epsilon m}{2}] \;\geq\; 1 - \mathbb{P}[|T \cap h_S| \leq \tfrac{\rho m}{2}] \;\geq\; 1/2.$$

将所有前面的内容结合起来，我们得到 *conclude the proof of Claim 1*。

Claim 2 (Symmetrization):
$\mathbb{P}[(S,T) \in B'] \leq e^{-\epsilon m/4}\,\tau_{\mathcal{H}}(2m)$.
*Proof of Claim 2*为简化符号，令 $\alpha = m\epsilon/2$，对于序列 $A = (x_1,\ldots,x_{2m})$ 令 $A_0 = (x_1,\ldots,x_m)$。使用 B' 的定义，我们得到

$$\begin{aligned}\mathbb{P}[A \in B'] &= \mathop{\mathbb{E}}_{A\sim\mathcal{D}^{2m}}\max_{h\in\mathcal{H}}\mathbb{1}_{[\mathcal{D}(h)\geq\epsilon]}\,\mathbb{1}_{[|h\cap A_0|=0]}\,\mathbb{1}_{[|h\cap A|\geq\alpha]} \\ &\leq \mathop{\mathbb{E}}_{A\sim\mathcal{D}^{2m}}\max_{h\in\mathcal{H}}\mathbb{1}_{[|h\cap A_0|=0]}\,\mathbb{1}_{[|h\cap A|\geq\alpha]}.\end{aligned}$$

现在，让我们用 $\mathcal{H}_A$ 定义 A 上的有效不同假设的数量，即 $\mathcal{H}_A = \{h \cap A\colon h \in \mathcal{H}\}$。由此可得

$$\begin{aligned}\mathbb{P}[A \in B'] &\leq \mathop{\mathbb{E}}_{A\sim\mathcal{D}^{2m}}\max_{h\in\mathcal{H}_A}\mathbb{1}_{[|h\cap A_0|=0]}\,\mathbb{1}_{[|h\cap A|\geq\alpha]} \\ &\leq \mathop{\mathbb{E}}_{A\sim\mathcal{D}^{2m}}\sum_{h\in\mathcal{H}_A}\mathbb{1}_{[|h\cap A_0|=0]}\,\mathbb{1}_{[|h\cap A|\geq\alpha]}.\end{aligned}$$

让 $J = \{\mathbf{j} \subset [2m] : |\mathbf{j}| = m\}$。对于任意的 $\mathbf{j} \in J$ 和 $A = (x_1,\ldots,x_{2m})$ 定义 $A_{\mathbf{j}} = (x_{j_1},\ldots,x_{j_m})$。由于 A 的元素是独立同分布选择的，因此对于任意的 $\mathbf{j} \in J$ 和任意的函数 $f(A, A_0)$，都有 $\mathbb{E}_{A\sim\mathcal{D}^{2m}}[f(A,A_0)] =$

$\mathbb{E}_{A\sim\mathcal{D}2_m}[f(A, A_{\mathbf{j}})]$. 由于这对任何 $\mathbf{j}$ 都成立，因此也对从 J 中随机选择的 $\mathbf{j}$ 的期望成立。特别是，它对函数 $f(A, A_0) = \sum_{h\in\mathcal{H}_A} 1_{[|h\cap A_0|=0]}\, 1_{[|h\cap A|\geq\alpha]}$ 成立。因此，我们得到:

$$\begin{aligned}\mathbb{P}[A \in B'] &\leq \mathop{\mathbb{E}}_{A\sim\mathcal{D}^{2m}} \mathop{\mathbb{E}}_{j\sim J} \sum_{h\in\mathcal{H}_A} \mathbb{1}_{[|h\cap A_{\mathbf{j}}|=0]}\, \mathbb{1}_{[|h\cap A|\geq\alpha]} \\ &= \mathop{\mathbb{E}}_{A\sim\mathcal{D}^{2m}} \sum_{h\in\mathcal{H}_A} \mathbb{1}_{[|h\cap A|\geq\alpha]} \mathop{\mathbb{E}}_{j\sim J} \mathbb{1}_{[|h\cap A_{\mathbf{j}}|=0]}.\end{aligned}$$

现在，固定一些 A 使得 $|h\cap A| \geq \alpha$。然后，$\mathbb{E}_j\, 1_{[|h\cap A_{\mathbf{j}}|=0]}$ 是从至少有 α 个红球的袋子中取出 m 个球时，我们永远不会选择到一个红球的概率。这个概率至多为

$$(1-\alpha/(2m))^m = (1-\epsilon/4)^m \leq e^{-\epsilon m/4}.$$

因此我们得到

$$\mathbb{P}[A \in B'] \leq \mathop{\mathbb{E}}_{A\sim\mathcal{D}^{2m}} \sum_{h\in\mathcal{H}_A} e^{-\epsilon m/4} \leq e^{-\epsilon m/4} \mathop{\mathbb{E}}_{A\sim\mathcal{D}^{2m}} |\mathcal{H}_A|.$$

使用增长函数的定义，我们 *conclude the proof of Claim 2.*
Completing the Proof: 通过Sauer引理，我们知道 $\tau_{\mathcal{H}}(2m) \leq (2em/d)^d$。
结合这一点与两个主张，我们得到

$$\mathbb{P}[S \in B] \leq 2(2em/d)^d\, e^{-\epsilon m/4}.$$

我们希望不等式的右侧最多为 δ；也就是说，

$$2(2em/d)^d\, e^{-\epsilon m/4} \leq \delta.$$

重新排列，我们得到要求

$$m \geq \frac{4}{\epsilon}\left(d\log(2em/d) + \log(2/\delta)\right) = \frac{4d}{\epsilon}\log(m) + \frac{4}{\epsilon}(d\log(2e/d) + \log(2/\delta).$$

使用引理A.2，使前面的条件成立的一个充分条件是

$$m \geq \frac{16d}{\epsilon}\log\left(\frac{8d}{\epsilon}\right) + \frac{8}{\epsilon}(d\log(2e/d) + \log(2/\delta).$$

这是一个充分条件，即

$$\begin{aligned}m &\geq \frac{16d}{\epsilon}\log\left(\frac{8d}{\epsilon}\right) + \frac{16}{\epsilon}(d\log(2e/d) + \tfrac{1}{2}\log(2/\delta) \\ &= \frac{16d}{\epsilon}\left(\log\left(\frac{8d\,2e}{d\epsilon}\right)\right) + \frac{8}{\epsilon}\log(2/\delta) \\ &= \frac{8}{\epsilon}\left(2d\log\left(\frac{16e}{\epsilon}\right) + \log\left(\frac{2}{\delta}\right)\right).\end{aligned}$$

这结束了我们的证明。 □

28.3.1 From ϵ-Nets to PAC Learnability

THEOREM 28.4 *Let $\mathcal{H}$ be a hypothesis class over $\mathcal{X}$ with* VCdim($\mathcal{H}$) $= d$. *Let $\mathcal{D}$ be a distribution over $\mathcal{X}$ and let $c \in \mathcal{H}$ be a target hypothesis. Fix $\epsilon, \delta \in (0, 1)$ and let m be as defined in Theorem 28.3. Then, with probability of at least $1 - \delta$ over a choice of m i.i.d. instances from $\mathcal{X}$ with labels according to c we have that any ERM hypothesis has a true error of at most ϵ.*

Proof 定义类 $\mathcal{H}^c = \{c \bigtriangleup h : h \in \mathcal{H}\}$，其中 $c \bigtriangleup h = (h \setminus c) \cup (c \setminus h)$。容易验证，如果某些 $A \subset \mathcal{X}$ 被 $\mathcal{H}$ 破碎，那么它也被 $\mathcal{H}^c$ 破碎，反之亦然。因此，VCdim($\mathcal{H}$) = VCdim($\mathcal{H}^c$)。因此，使用定理28.3，我们知道至少以$1 - \delta$ 的概率，样本 S 是 ϵ -net 对于 $\mathcal{H}^c$。注意 $L_{\mathcal{D}}(h) = \mathcal{D}(h \bigtriangleup c)$。因此，对于任何 $h \in \mathcal{H}$ 与 $L_{\mathcal{D}}(h) \geq \epsilon$，我们有 $|(h \bigtriangleup c) \cap S| > 0$，这意味着 h 不能是一个ERM假设，这完成了我们的证明。

□

29 Multiclass Learnability

第17章我们介绍了多类分类问题，其目标是学习一个预测器 $h : \mathcal{X} \to [k]$。在本章中，我们讨论了关于0-1损失的多类预测器的PAC可学习性。与第6章一样，本章的主要目标是：

- 描述在（多类）PAC模型中哪些多类假设类是可学习的。
- 量化此类假设类的样本复杂度。

考虑到学习理论的基本定理（定理6.8），自然地寻求将VC维推广到多类假设类。在第29.1节中，我们展示了这种推广，称为*Natarajan dimension*，并基于Natarajan维数陈述了基本定理的推广。然后，我们演示了如何计算几个重要假设类的Natarajan维数。

回忆一下，学习理论基本定理的主要信息是，一个二分类器的假设类（相对于0-1损失）是可学习的，当且仅当它具有一致收敛性质，然后它可以通过任何ERM学习器进行学习。在第13章的练习2中，我们已表明这种等价性对于某个凸学习问题是不成立的。本章的最后部分致力于证明，即使在具有0-1损失的具有多类问题的可学习性与一致收敛性之间，这种等价性也会破裂，这些问题与二分类非常相似。事实上，我们构造了一个假设类，它可以被一个特定的ERM学习器学习，但对于其他ERM学习器可能失败，并且一致收敛性质不成立。

29.1 The Natarajan Dimension

本节中，我们定义了Natarajan维度，它是VC维度的多类预测器类别的推广。在本节中，令$\mathcal{H}$为多类预测器的假设类；即，每个$h \in \mathcal{H}$是从$\mathcal{X}$到$[k]$的函数。

为了定义Natarajan维度，我们首先推广了shat-tering的定义。

DEFINITION 29.1（破碎（多类版本））我们称一个集合 $C \subset \mathcal{X}$ 被 $\mathcal{H}$ 破碎，如果存在两个函数 $f_0, f_1 : C \to [k]$ 使得

- 对于每个 $x \in C$，$f_0(x) \neq f_1(x)$。
- 对于每个 $B \subset C$，存在一个函数 $h \in \mathcal{H}$ 使得

$$\forall x \in B, h(x) = f_0(x) \text{ and } \forall x \in C \setminus B, h(x) = f_1(x).$$

DEFINITION 29.2 (Natarajan维度) $\mathcal{H}$的Natarajan维度，表示为Ndim($\mathcal{H}$)，是最大碎集$C \subset \mathcal{X}$的大小。

不难看出，在恰好有两个类别的情形下，Ndim($\mathcal{H}$) = VCdim($\mathcal{H}$)。因此，Natarajan维度推广了VC维度。接下来，我们展示Natarajan维度使我们能够将统计学习的基本定理从二分类推广到多分类。

29.2 The Multiclass Fundamental Theorem

THEOREM 29.3（多类基本定理）*There exist absolute constants $C_1, C_2 > 0$ such that the following holds. For every hypothesis class $\mathcal{H}$ of functions from $\mathcal{X}$ to $[k]$, such that the Natarajan dimension of $\mathcal{H}$ is d, we have*

1. $\mathcal{H}$ has the uniform convergence property with sample complexity

$$C_1 \frac{d + \log(1/\delta)}{\epsilon^2} \leq m^{\text{UC}}_{\mathcal{H}}(\epsilon, \delta) \leq C_2 \frac{d \log(k) + \log(1/\delta)}{\epsilon^2}.$$

2. $\mathcal{H}$ is agnostic PAC learnable with sample complexity

$$C_1 \frac{d + \log(1/\delta)}{\epsilon^2} \leq m_{\mathcal{H}}(\epsilon, \delta) \leq C_2 \frac{d \log(k) + \log(1/\delta)}{\epsilon^2}.$$

3. $\mathcal{H}$ is PAC learnable (assuming realizability) with sample complexity

$$C_1 \frac{d + \log(1/\delta)}{\epsilon} \leq m_{\mathcal{H}}(\epsilon, \delta) \leq C_2 \frac{d \log\left(\frac{kd}{\epsilon}\right) + \log(1/\delta)}{\epsilon}.$$

29.2.1 On the Proof of Theorem 29.3

定理29.3的下界可以通过从二进制基本定理（参见练习5）进行归纳推导得出。

定理29.3中的上界可以沿着第28章（参见练习4）中给出的二分类基本定理的证明方法进行证明。该证明中唯一需要以非直接方式修改的成分是Sauer引理。它仅适用于二分类，因此必须被替换。一个合适的替代是Natarajan引理:

LEMMA 29.4 (Natarajan) $|\mathcal{H}| \leq |\mathcal{X}|^{\mathrm{Ndim}(\mathcal{H})} \cdot k^{2\mathrm{Ndim}(\mathcal{H})}$.

Natarajan引理的证明与Sauer引理的证明具有相同的风格，并将其留作练习（见练习3）。

29.3 Calculating the Natarajan Dimension

在这一节中，我们展示了如何计算（或估计）几个流行类别的Natarajan维度，其中一些在第17章中进行了研究。正如这些计算所示，Natarajan维度通常与定义假设所需的参数数量成正比。

29.3.1 One-versus-All Based Classes

第17章中，我们看到了将多类分类减少到二类分类的两种方法：一对一和全部对。在本节中，我们计算一对一方法的Natarajan维度。

回忆在One-versus-All中，我们为每个标签训练一个二元分类器，以区分该标签与其他标签。这自然地暗示考虑以下形式的多类假设类。设$\mathcal{H}_{\mathrm{bin}} \subset \{0,1\}^{\mathcal{X}}$为一个二元假设类。对于每个$\bar{h} = (h_1, \ldots, h_k) \in (\mathcal{H}_{\mathrm{bin}})^k$，定义$T(\bar{h})$：$\mathcal{X} \to [k]$为

$$T(\bar{h})(x) = \operatorname*{argmax}_{i \in [k]} h_i(x).$$

如果有两个标签最大化 $h_i(x)$，我们选择较小的那个。此外，让

$$\mathcal{H}_{\mathrm{bin}}^{\mathrm{OvA,k}} = \{T(\bar{h}) \ : \ \bar{h} \in (\mathcal{H}_{\mathrm{bin}})^k\}.$$

什么应该是 $\mathcal{H}_{\mathrm{bin}}^{\mathrm{OvA,k}}$ 的 Natarajan 维度？直观上，为了在 $\mathcal{H}_{\mathrm{bin}}$ 中指定一个假设，我们需要 $d = \mathrm{VCdim}(\mathcal{H}_{\mathrm{bin}})$ 个参数。为了在 $\mathcal{H}_{\mathrm{bin}}^{\mathrm{OvA,k}}$ 中指定一个假设，我们需要指定 k 个假设在 $\mathcal{H}_{\mathrm{bin}}$ 中。因此，kd 个参数应该足够。以下引理建立了这种直觉。

LEMMA 29.5 *If* $d = \mathrm{VCdim}(\mathcal{H}_{\mathrm{bin}})$ *then*

$$\mathrm{Ndim}(\mathcal{H}_{\mathrm{bin}}^{\mathrm{OvA,k}}) \leq 3kd \log(kd).$$

Proof 设 $C \subset \mathcal{X}$ 为一个破碎集。根据破碎集的定义（对于多类假设）

$$\left|\left(\mathcal{H}_{\mathrm{bin}}^{\mathrm{OvA,k}}\right)_C\right| \ \geq \ 2^{|C|}.$$

另一方面，$\mathcal{H}_{\mathrm{bin}}^{\mathrm{OvA,k}}$中的每个假设都是通过使用$\mathcal{H}_{\mathrm{bin}}$中的$k$个假设来确定的。因此，

$$\left|\left(\mathcal{H}_{\mathrm{bin}}^{\mathrm{OvA,k}}\right)_C\right| \ \leq \ |\,(\mathcal{H}_{\mathrm{bin}})_C\,|^k.$$

通过Sauer引理，$|(\mathcal{H}_{\text{bin}})_C| \leq |C|^d$。我们得出结论，

$$2^{|C|} \;\leq\; \left|\left(\mathcal{H}_{\text{bin}}^{\text{OvA,k}}\right)_C\right| \;\leq\; |C|^{dk}.$$

证明通过取对数并应用引理A.1得出。 □

如何紧致引理29.5? 不难看出，对于某些类，Ndim($\mathcal{H}_{\text{bin}}^{\text{OvA,k}}$)可以比$dk$ (小得多，参见练习1)。然而，对于一些自然的二进制类，$\mathcal{H}_{\text{bin}}$ (例如，半空间)，有Ndim($\mathcal{H}_{\text{bin}}^{\text{OvA,k}}$) = $\Omega(dk)$（参见练习6）。

29.3.2 General Multiclass-to-Binary Reductions

相同的推理可用于上界更一般的类别到二类归约的Natarajan维度。这些归约在数据上训练多个二分类器。然后，给定一个新实例，它们通过使用一些规则来预测其标签，该规则考虑了二分类器预测的标签。这些归约包括一对一和成对比较。

假设这种方法从二元类别 $\mathcal{H}_{\text{bin}}$ 中训练 l 个二元分类器，并且 $r : \{0,1\}^l \to [k]$ 是根据二元分类器的预测确定（多类别）标签的规则。对应于这种方法的理论类别可以定义为以下。对于每个 $\bar{h} = (h_1, \ldots, h_l) \in (\mathcal{H}_{\text{bin}})^l$，定义 $R(\bar{h}) : \mathcal{X} \to [k]$ 为

$$R(\bar{h})(x) = r(h_1(x), \ldots, h_l(x)).$$

最后，设 {v*}

$$\mathcal{H}_{\text{bin}}^r = \{R(\bar{h}) \;:\; \bar{h} \in (\mathcal{H}_{\text{bin}})^l\}.$$

与引理29.5类似，可以证明:

LEMMA 29.6 *If* $d = \text{VCdim}(\mathcal{H}_{\text{bin}})$ *then*

$$\text{Ndim}(\mathcal{H}_{\text{bin}}^r) \;\leq\; 3\,l\,d\,\log(l\,d)\,.$$

证明留作练习2。

29.3.3 Linear Multiclass Predictors

接下来，我们考虑线性多类预测器类别（参见第17.2节）。设 $\Psi : \mathcal{X} \times [k] \to \mathbb{R}^d$ 是某种类别敏感的特征映射，并设

$$\mathcal{H}_\Psi = \left\{ x \mapsto \underset{i \in [k]}{\text{argmax}} \langle \mathbf{w}, \Psi(x,i) \rangle \;:\; \mathbf{w} \in \mathbb{R}^d \right\}. \tag{29.1}$$

每个 $\mathcal{H}_\Psi$ 中的假设由 d 个参数确定，即向量 $\mathbf{w} \in \mathbb{R}^d$。因此，我们预计 Natarajan 维度将受限于 d。确实:

THEOREM 29.7 $\text{Ndim}(\mathcal{H}_\Psi) \leq d$.

Proof 设 $C \subset \mathcal{X}$ 为一个破碎集，设 $f_0, f_1 : C \to [k]$ 是见证破碎的两个函数。我们需要证明 $|C| \leq d$。对于每一个 $x \in C$，设 $\rho(x) = \Psi(x, f_0(x)) - \Psi(x, f_1(x))$。我们断言集合 $\rho(C) \stackrel{\text{def}}{=} \{\rho(x) : x \in C\}$ 由 $|C|$ 个元素组成（即 ρ 是一一对应的）并且被 $\mathbb{R}^d$ 上的同质线性分离器的二进制假设类破碎。

$$\mathcal{H} = \{\mathbf{x} \mapsto \text{sign}(\langle \mathbf{w}, \mathbf{x} \rangle) \ : \ \mathbf{w} \in \mathbb{R}^d\}.$$

由于 $\text{VCdim}(\mathcal{H}) = d$，因此将得出 $|C| = |\rho(C)| \leq d$，如需要求。

要确立我们的主张，只需证明 $|\mathcal{H}_{\rho(C)}| = 2^{|C|}$ 即可。确实，给定一个子集 $B \subset C$，根据分裂的定义，存在 $h_B \in \mathcal{H}_\Psi$ 使得

$$\forall x \in B, h_B(x) = f_0(x) \quad \text{and} \quad \forall x \in C \setminus B, h_B(x) = f_1(x).$$

设 $\mathbf{w}_B \in \mathbb{R}^d$ 为定义 h_B 的向量。对于每个 $x \in B$，我们有：

$$\langle \mathbf{w}, \Psi(x, f_0(x)) \rangle > \langle \mathbf{w}, \Psi(x, f_1(x)) \rangle \ \Rightarrow \ \langle \mathbf{w}, \rho(x) \rangle > 0.$$

同样，对于每个 $x \in C \setminus B$,

$$\langle \mathbf{w}, \rho(x) \rangle < 0.$$

因此，由相同的 $\mathbf{w} \in \mathbb{R}^d$ 标签定义的假设 $g_B \in \mathcal{H}$ 将 $\rho(B)$ 中的点标记为 1，将 $\rho(C \setminus B)$ 中的点标记为 0。由于这对每个 $B \subseteq C$ 都成立，我们得到 $|C| = |\rho(C)|$ 和 $|\mathcal{H}_{\rho(C)}| = 2^{|C|}$，这完成了我们的证明。

□

定理在以下意义上是紧的：存在映射 Ψ，使得 $\text{Ndim}(\mathcal{H}_\Psi) = \Omega(d)$。例如，这在多向量构造中是正确的（参见第17.2节和本章末尾的参考文献注释）。因此，我们得出结论：

COROLLARY 29.8 *Let $\mathcal{X} = \mathbb{R}^n$ and let $\Psi : \mathcal{X} \times [k] \to \mathbb{R}^{nk}$ be the class sensitive feature mapping for the multi-vector construction:*

$$\Psi(\mathbf{x}, y) = [\ \underbrace{0, \ldots, 0}_{\in \mathbb{R}^{(y-1)n}}, \underbrace{x_1, \ldots, x_n}_{\in \mathbb{R}^n}, \underbrace{0, \ldots, 0}_{\in \mathbb{R}^{(k-y)n}}\ ].$$

Let $\mathcal{H}_\Psi$ be as defined in Equation (29.1). Then, the Natarajan dimension of $\mathcal{H}_\Psi$ satisfies

$$(k-1)(n-1) \ \leq \ \text{Ndim}(\mathcal{H}_\Psi) \ \leq \ kn.$$

29.4 On Good and Bad ERMs

在这个部分，我们展示了具有以下性质的假设类的一个例子：该类中并非所有ERM都同样成功。此外，如果我们允许无限数量的标签，我们还将获得一个类的例子，该类是

可由某些ERM学习，但其他ERM将无法学习它。显然，这也意味着该类是可学习的，但它不具有一致收敛性质。为了简单起见，我们只考虑可实现的情况。

我们考虑的类如下定义。实例空间 {v*} 将是任何有限或可数集。令 $P_f(\mathcal{X})$ 为 $\mathcal{X}$ (的所有有限和可数子集的集合，即对于每个 $A \in P_f(\mathcal{X})$，A 或 $\mathcal{X} \setminus A$ 必须是有限的)。而不是 $[k]$，标签集是 $\mathcal{Y} = P_f(\mathcal{X}) \cup \{*\}$，其中 $*$ 是某个特殊标签。对于每个 $A \in P_f(\mathcal{X})$，定义 $h_A : \mathcal{X} \to \mathcal{Y}$ 为

$$h_A(x) = \begin{cases} A & x \in A \\ * & x \notin A \end{cases}$$

最后，我们选取的假设类是

$$\mathcal{H} = \{h_A \ : \ A \in P_f(\mathcal{X})\}.$$

设 $\mathcal{A}$ 是针对 $\mathcal{H}$ 的某个 ERM 算法。假设 $\mathcal{A}$ 在被 $h_A \in \mathcal{H}$ 标记的样本上操作。由于 h_A 是 $\mathcal{H}$ 中的 *only* 假设，可能会返回标签 A，如果 $\mathcal{A}$ 观察到标签 A，它“知道”学到的假设是 h_A，并且作为一个 ERM，必须返回它（注意在这种情况下返回的假设的错误为 0）。因此，为了指定一个 ERM，我们只需指定它在接收到形式为的样本时返回的假设即可

$$S = \{(x_1, *), \ldots, (x_m, *)\}.$$

我们考虑两种ERMs：第一种，$\mathcal{A}_{good}$，定义为

$$\mathcal{A}_{good}(S) = h_\emptyset;$$

这是指它输出预测每个 $x \in \mathcal{X}$ 为 ‘*’ 的假设。第二个 ERM，$\mathcal{A}_{bad}$，定义为

$$\mathcal{A}_{bad}(S) = h_{\{x_1, \ldots x_m\}^c}.$$

以下声明表明，$\mathcal{A}_{bad}$的样本复杂度大约是$|\mathcal{X}|$倍于$\mathcal{A}_{good}$的样本复杂度。这在不同ERM之间建立了一个差距。如果$\mathcal{X}$是无限的，我们甚至获得一个每个ERM都无法学习的可学习类别。

CLAIM 29.9

1. Let $\epsilon, \delta > 0$, $\mathcal{D}$ a distribution over $\mathcal{X}$ and $h_A \in \mathcal{H}$. Let S be an i.i.d. sample consisting of $m \geq \frac{1}{\epsilon} \log\left(\frac{1}{\delta}\right)$ examples, sampled according to $\mathcal{D}$ and labeled by h_A. Then, with probability of at least $1 - \delta$, the hypothesis returned by $\mathcal{A}_{good}$ will have an error of at most ϵ.

2. There exists $\mathcal{D}$ over $\mathcal{X}$ and $h_A \in \mathcal{H}$ such that the following holds. The hypothesis returned by $\mathcal{A}_{bad}$ upon receiving a sample of size $m \leq \frac{|X|-1}{6\epsilon}$, sampled according to $\mathcal{D}$ and labeled by h_A, will have error $\geq \epsilon$ with probability $\geq e^{-\frac{1}{6}}$.

Proof 设 $\mathcal{D}$ 是 $\mathcal{X}$ 上的一个分布，并假设正确的标签是 h_A。对于任何样本，$\mathcal{A}_{good}$ 返回 $h_\emptyset$ 或 h_A。如果它返回 h_A，则其真实错误为零。因此，只有当样本中的所有 m 个示例都来自 $\mathcal{X} \setminus A$，而 $h_\emptyset$，$L_{\mathcal{D}}(h_\emptyset) = \mathbb{P}_{\mathcal{D}}[A]$ 的错误是 $\geq \epsilon$ 时，它才返回一个具有 $\geq \epsilon$ 错误的假设。假设 $m \geq \frac{1}{\epsilon}\log(\frac{1}{\delta})$；那么后者的概率不超过 $(1-\epsilon)^m \leq e^{-\epsilon m} \leq \delta$。这确立了项目 1。

接下来我们证明第2项。我们将证明限制在$|\mathcal{X}| = d < \infty$的情况下。在无穷$\mathcal{X}$的情况下的证明类似。假设$\mathcal{X} = \{x_0, \ldots, x_{d-1}\}$。

让 $a > 0$ 足够小，使得对于每个 $\epsilon < a$，$1 - 2\epsilon \geq e^{-4\epsilon}$，并固定一些 $\epsilon < a$。在 $\mathcal{X}$ 上定义一个分布，通过设置 $\mathbb{P}[x_0] = 1 - 2\epsilon$，并且对于所有 $1 \leq i \leq d-1$，$\mathbb{P}[x_i] = \frac{2\epsilon}{d-1}$。假设正确的假设是 $h_\emptyset$，并让样本大小为 m。显然，$\mathcal{A}_{bad}$ 返回的假设将在样本外的所有 $\mathcal{X}$ 示例上出错。根据切诺夫不等式，如果 $m \leq \frac{d-1}{6\epsilon}$，那么以概率 $\geq e^{-\frac{1}{6}}$，样本将不会包含超过 $\frac{d-1}{2}$ 个来自 $\mathcal{X}$ 的示例。因此，返回的假设将具有 $\geq \epsilon$ 错误。

□

结论是，在多类分类中，不同ERMs的样本复杂度可能不同。对于*every*假设类，是否存在“好的”ERMs？以下猜想断言答案是肯定的。

CONJECTURE 29.10 *The realizable sample complexity of every hypothesis class* $\mathcal{H} \subset [k]^{\mathcal{X}}$ *is*

$$m_{\mathcal{H}}(\epsilon, \delta) = \tilde{O}\left(\frac{\mathrm{Ndim}(\mathcal{H})}{\epsilon}\right).$$

We emphasize that the $\tilde{O}$ *notation may hide only poly-log factors of* ϵ, δ, *and* Ndim($\mathcal{H}$), *but* 没有因子 *of* k.

29.5 Bibliographic Remarks

Natarajan 维度归功于 Natarajan（1989）。该论文还建立了 Natarajan 引理和基本定理的推广。在 Haussler & Long（1995）中研究了 Natarajan 引理的推广和更尖锐的版本。Ben-David、Cesa-Bianchi、Haussler & Long（1995）定义了一大家族维度概念，所有这些概念都推广了 VC 维度，可用于估计多类分类的样本复杂度。

Natarajan维度的计算，在此处提出，以及其他类别的计算，可在Daniely等人（2012年）的研究中找到。良好和不良ERMs的示例，以及猜想29.10，均来自Daniely等人（2011年）。

29.6 Exercises

1. 设 $d, k >$ = 0。证明存在一个具有 VC 维度 d 的二进制假设 $\mathcal{H}_{\text{bin}}$，使得 Ndim($\mathcal{H}_{\text{bin}}^{\text{OvA,k}}$) $= d$。
2. 证明引理29.6。3. 证明Nata
rajan引理。

Hint: 修复一些 $x_0 \in \mathcal{X}$。对于 $i, j \in [k]$，用 $\mathcal{H}_{ij}$ 表示所有可以扩展到 $\mathcal{H}$ 中的函数 $f : \mathcal{X} \setminus \{x_0\} \to [k]$，这些函数可以通过定义 $f(x_0) = i$ 和定义 $f(x_0) = j$ 来扩展。证明 $|\mathcal{H}| \leq |\mathcal{H}_{\mathcal{X}\setminus\{x_0\}}| + \sum_{i\neq j} |\mathcal{H}_{ij}|$ 并使用归纳法。

4. 将二进制基本定理和Natarajan引理的证明进行改编，以证明对于某个通用常数 $C > 0$ 和对于每个Natarajan维度的假设类 d，$\mathcal{H}$ 的无偏见样本复杂度是

$$m_{\mathcal{H}}(\epsilon, \delta) \leq C \frac{d \log\left(\frac{kd}{\epsilon}\right) + \log(1/\delta)}{\epsilon^2}.$$

5. 证明对于某个通用常数 $C > 0$ 和对于每个具有 Natarajan 维度的假设类 d，$\mathcal{H}$ 的无偏见样本复杂度是

$$m_{\mathcal{H}}(\epsilon, \delta) \geq C \frac{d + \log(1/\delta)}{\epsilon^2}.$$

Hint: 从二进制基本定理中推导它。
6. 令 $\mathcal{H}$ 为 $\mathbb{R}^d$ 中（非齐次）半空间的二进制假设类。本练习的目标是证明 Ndim($\mathcal{H}^{\text{OvA,k}}$) $\geq (d-1) \cdot (k-1)$。

1. 令 $\mathcal{H}_{\text{discrete}}$ 为所有函数 f 的类别 : $[k-1] \times [d-1] \to \{0, 1\}$，对于这些函数存在某个 i_0，使得对于每个 $j \in [d-1]$

$$\forall i < i_0, f(i, j) = 1 \text{ while } \forall i > i_0, f(i, j) = 0.$$

证明 Ndim($\mathcal{H}_{\text{discrete}}^{\text{OvA,k}}$) $= (d-1) \cdot (k-1)$。
2. 证明$\mathcal{H}_{\text{discrete}}$可以通过$\mathcal{H}$实现。也就是说，证明存在一个映射$\psi$：$[k-1] \times [d-1] \to \mathbb{R}^d$，使得

$$\mathcal{H}_{\text{discrete}} \subset \{h \circ \psi \ : \ h \in \mathcal{H}\} .$$

Hint: 您可以将 $\psi(i, j)$ 看作是一个向量，其 j 个坐标为 1，最后一个坐标为 i，其余为 0。
3. 推断 Ndim($\mathcal{H}^{\text{OvA,k}}$) $\geq (d-1) \cdot (k-1)$。

30 Compression Bounds

整本书中，我们尝试使用不同的方法来描述可学习性的概念。起初，我们展示了假设类的一致收敛性质保证了学习的成功。后来，我们引入了稳定性的概念，并证明了稳定的算法保证是好的学习者。然而，还有其他可能足以支持学习的性质，在本章及其后续章节中，我们将介绍两种处理这一问题的方法：压缩界限和PAC-Bayes方法。

在这一章中，我们研究压缩界限。粗略地说，我们将看到如果一个学习算法可以使用训练集的一个小子集来表示输出假设，那么该假设在其余示例上的错误估计了其真实错误。换句话说，一个能够“压缩”其输出的算法是一个好的学习器。

30.1 Compression Bounds

为了激励结果，让我们首先考虑以下学习协议。首先，我们采样一个由 k 个示例组成的序列，记为 T。在这些示例的基础上，我们构建一个假设，记为 h_T。现在我们想要估计 h_T 的性能，所以我们采样一个由 $m-k$ 个示例组成的新序列，记为 V，并计算 h_T 在 V 上的误差。由于 V 和 T 是独立的，我们可以立即从伯恩斯坦不等式（参见引理 B.10）得到以下结果。

LEMMA 30.1 *Assume that the range of the loss function is* $[0,1]$. *Then,*

$$\mathbb{P}\left[L_{\mathcal{D}}(h_T) - L_V(h_T) \geq \sqrt{\frac{2L_V(h_T)\log(1/\delta)}{|V|}} + \frac{4\log(1/\delta)}{|V|}\right] \leq \delta.$$

要推导这个界限，我们只需要 T 和 V 之间的独立性。因此，我们可以如下重新定义协议。首先，我们同意一个 k 索引序列 $I = (i_1, \ldots, i_k) \in [m]^k$。然后，我们采样一个 m 个例子的序列 $S = (z_1, \ldots, z_m)$。现在，定义 $T = S_I = (z_{i_1}, \ldots, z_{i_k})$ 并将 V 定义为 S 中剩余的例子。注意，此协议与我们之前定义的协议等价——因此引理30.1仍然成立。

应用对索引序列选择的并集约束，我们得到以下定理。

THEOREM 30.2 *Let k be an integer and let $B : Z^k \to \mathcal{H}$ be a mapping from sequences of k examples to the hypothesis class. Let $m \geq 2k$ be a training set size and let $A : Z^m \to \mathcal{H}$ be a learning rule that receives a training sequence S of size m and returns a hypothesis such that $A(S) = B(z_{i_1}, \ldots, z_{i_k})$ for some $(i_1, \ldots, i_k) \in [m]^k$. Let $V = \{z_j : j \notin (i_1, \ldots, i_k)\}$ be the set of examples which were not selected for defining $A(S)$. Then, with probability of at least $1 - \delta$ over the choice of S we have*

$$L_{\mathcal{D}}(A(S)) \;\leq\; L_V(A(S)) + \sqrt{L_V(A(S))\frac{4k\log(m/\delta)}{m}} + \frac{8k\log(m/\delta)}{m}.$$

Proof 对于任意的 {v*} [{v*}]{v*}，令 {v*}。令 {v*}。结合引理30.1与并集界，我们有

$$\begin{aligned}
&\mathbb{P}\left[\exists I \in [m]^k \text{ s.t. } L_{\mathcal{D}}(h_I) - L_V(h_I) \geq \sqrt{\frac{2L_V(h_I)\log(1/\delta)}{n}} + \frac{4\log(1/\delta)}{n}\right] \\
&\quad\leq \sum_{I\in[m]^k} \mathbb{P}\left[L_{\mathcal{D}}(h_I) - L_V(h_I) \geq \sqrt{\frac{2L_V(h_I)\log(1/\delta)}{n}} + \frac{4\log(1/\delta)}{n}\right] \\
&\quad\leq m^k\delta.
\end{aligned}$$

表示 $\delta' = m^k\delta$。使用假设 $k \leq m/\text{^}2$，它意味着 $n = m - k \geq m/\text{^}2$，上述推导表明，至少以概率 $1 - \delta'$，我们有

$$L_{\mathcal{D}}(A(S)) \;\leq\; L_V(A(S)) + \sqrt{L_V(A(S))\frac{4k\log(m/\delta')}{m}} + \frac{8k\log(m/\delta')}{m},$$

这总结了我们的证明。 □

作为直接推论，我们得到：

COROLLARY 30.3 *Assuming the conditions of Theorem 30.2, and further assuming that $L_V(A(S)) = 0$, then, with probability of at least $1 - \delta$ over the choice of S we have*

$$L_{\mathcal{D}}(A(S)) \;\leq\; \frac{8k\log(m/\delta)}{m}.$$

这些结果激励了以下定义：

DEFINITION 30.4 {v*} 设 $\mathcal{H}$ 是从 $\mathcal{X}$ 到 $\mathcal{Y}$ 的函数的假设类，设 k 是一个整数。我们说 $\mathcal{H}$ 具有大小为 k 的压缩方案，如果以下条件成立：

对于所有 m，存在 A：$Z^m \to [m]^k$ 和 B：$Z^k \to \mathcal{H}$，使得对于所有 $h \in \mathcal{H}$，如果我们将任何形式为 $(x_1, h(x_1)), \ldots, (x_m, h(x_m))$ 的训练集输入 A，然后输入 $(x_{i_1}, h(x_{i_1})), \ldots, (x_{i_k}, h(x_{i_k}))$ 到 B，其中 $(i_1, \ldots, i_k)$ 是 A 的输出，那么 B 的输出，记为 h'，满足 $L_S(h') = 0$。

它可以将不可实现序列的定义推广如下。

DEFINITION 30.5 {v*} 设 $\mathcal{H}$ 是从 $\mathcal{X}$ 到 $\mathcal{Y}$ 的函数的假设类，设 k 是一个整数。我们说 $\mathcal{H}$ 具有大小为 k 的压缩方案，如果以下条件成立：对于所有 m，存在 A ： $Z^m \to [m]^k$ 和 B： $Z^k \to \mathcal{H}$，使得对于所有 $h \in \mathcal{H}$，如果我们将任何形式为 ($x_1, y_1), \ldots, (x_m, y_m)$ 的训练集输入 A，然后输入 $(x_{i_1}, y_{i_1}), \ldots, (x_{i_k}, y_{i_k})$ 到 B，其中 $(i_1, \ldots, i_k)$ 是 A 的输出，那么 B 的输出，记为 h'，满足 $L_S(h') \leq L_S(h)$。

以下引理表明，对于可实现情况存在压缩方案也意味着对于不可实现情况存在压缩方案。

LEMMA 30.6 *Let $\mathcal{H}$ be a hypothesis class for binary classification, and assume it has a compression scheme of size k in the realizable case. Then, it has a compression scheme of size k for the unrealizable case as well.*

Proof 考虑以下方案：首先，找到一个ERM假设，用h表示。然后，丢弃所有h出错的示例。现在，对未删除的示例应用可实现的压缩方案。可实现的压缩方案的输出，用h'表示，必须在未删除的示例上正确。由于h在删除的示例上出错，因此h'的错误不能大于h的错误；因此h'也是一个ERM假设。

□

30.2 Examples

在以下示例中，我们为几个假设类提出了二分类的压缩方案。鉴于引理30.6，我们关注可实现的情况。因此，为了证明某个假设类具有压缩方案，必须证明存在$A, B,$和k，使得$L_S(h') = 0$。

30.2.1 Axis Aligned Rectangles

请注意，这是一个不可数的无限类。我们表明存在一个简单的压缩方案。考虑以下算法 A，其工作方式如下：对于每个维度，选择在该维度上具有极值的前两个正例。定义 B 为返回最小外接矩形的函数。然后，对于 $k = \in d$，在可实现的情况下，有 $L_S(B(A(S))) = = 0$。

30.2.2 Halfspaces

让 $\mathcal{X} = \mathbb{R}^d$ 并考虑同质半空间类，$\{\mathbf{x} \mapsto$ 符号$(\langle \mathbf{w}, \mathbf{x} \rangle)$： $\mathbf{w} \in \mathbb{R}^d\}$。

A Compression Scheme:
W.l.o.g. 假设所有标签都是正的（否则，将 $\mathbf{x}_i$ 替换为 $y_i\mathbf{x}_i$）。我们提出的压缩方案如下。首先，A 找到在 $\{\mathbf{x}_1,\ldots,\mathbf{x}_m\}$ 的凸包中且范数最小的向量 $\mathbf{w}$。然后，它将其表示为样本中 d 个点的凸组合（稍后将证明这总是可能的）。A 的输出是这些 d 个点。算法 B 接收这些 d 个点，并将 $\mathbf{w}$ 设置为它们凸包中范数最小的点。

接下来我们证明这确实是一个压缩方案。由于数据是线性可分的，$\{\mathbf{x}_1,\ldots,\mathbf{x}_m\}$的凸包不包含原点。考虑凸包中离原点最近的点$\mathbf{w}$。（这是一个唯一的点，是原点到这个凸包的欧几里得投影。）我们声称$\mathbf{w}$可以分离数据.[1]为了证明这一点，我们假设反证法，对于某些i，$\langle \mathbf{w},\mathbf{x}_i\rangle \leq 0$。取$\mathbf{w}' = (1 - \alpha)\mathbf{w} + \alpha\mathbf{x}_i$对于$\alpha = \frac{\|\mathbf{w}\|^2}{\|\mathbf{x}_i\|^2+\|\mathbf{w}\|^2} \in (0,1)$。那么$\mathbf{w}'$也在凸包中，

$$\begin{aligned}
\|\mathbf{w}'\|^2 &= (1-\alpha)^2\|\mathbf{w}\|^2 + \alpha^2\|\mathbf{x}_i\|^2 + 2\alpha(1-\alpha)\langle \mathbf{w},\mathbf{x}_i\rangle \\
&\leq (1-\alpha)^2\|\mathbf{w}\|^2 + \alpha^2\|\mathbf{x}_i\|^2 \\
&= \frac{\|\mathbf{x}_i\|^4\|\mathbf{w}\|^2 + \|\mathbf{x}_i\|^2\|\mathbf{w}\|^4}{(\|\mathbf{w}\|^2 + \|\mathbf{x}_i\|^2)^2} \\
&= \frac{\|\mathbf{x}_i\|^2\|\mathbf{w}\|^2}{\|\mathbf{w}\|^2 + \|\mathbf{x}_i\|^2} \\
&= \|\mathbf{w}\|^2 \cdot \frac{1}{\|\mathbf{w}\|^2/\|\mathbf{x}_i\|^2 + 1} \\
&< \|\mathbf{w}\|^2,
\end{aligned}$$

这导致矛盾。

我们已经表明 $\mathbf{w}$ 也是一个 ERM。最后，由于 $\mathbf{w}$ 在示例的凸包中，我们可以应用 Caratheodory 定理，从而得出 $\mathbf{w}$ 也位于 $d+$ 的一个子集的凸包中，该子集包含多边形的 1 个点。此外，$\mathbf{w}$ 的最小性意味着 $\mathbf{w}$ 必须位于多边形的一个面上，这表明它可以表示为 d 个点的凸组合。

它还需要证明 $\mathbf{w}$ 也是投影到由 d 点定义的多边形上。但这是必须成立的：一方面，较小的多边形是较大多边形的一个子集；因此，在较小多边形上的投影在范数上不能更小。另一方面，$\mathbf{w}$ 本身是一个有效解。投影的唯一性得出了我们的证明。

30.2.3 Separating Polynomials

让 $\mathcal{X} = \mathbb{R}^d$ 并考虑类 $\mathbf{x} \mapsto \text{sign}(p(x))$，其中 p 是一个次数 r 的多项式。

1 可证明 $\mathbf{w}$ 是最大间隔解的方向。

注意，$p(x)$ 可以重写为 $\langle \mathbf{w}, \psi(\mathbf{x}) \rangle$，其中 $\psi(x)$ 的元素是 $\mathbf{x}$ 的所有单项式，直到次数 r。因此，构造 $p(\mathbf{x})$ 的压缩方案的問題縮減為構造 $\mathbb{R}^{d'}$ 中半空间的压缩方案問題，其中 $d' = O(d^r)$。

30.2.4 Separation with Margin

假设一个训练集被与边缘 γ 分隔。感知机算法保证在收敛到一个在整个训练集上无错误的解之前，最多进行 $1/\gamma^2$ 次更新。因此，我们有一个大小为 $k \leq 1/\gamma^2$ 的压缩方案。

30.3 Bibliographic Remarks

压缩方案及其与学习的关系由Little-stone & Warmuth (1986) 提出。正如我们所展示的，如果一个类有压缩方案，那么它是可学习的。对于二元分类问题，根据学习的基本定理，该类具有有限的VC维。另一个方向，即是否每个有限VC维的假设类都有有限大小的压缩方案，是Manfred Warmuth提出的一个未解决的问题，至今仍未解决（参见Floyd 1989，Floyd & Warmuth 1995，Ben-David & Litman 1998，Livni & Simon 2013）。

31 PAC-Bayes

最小描述长度（MDL）和奥卡姆剃刀原则允许一个潜在非常大的假设类，但定义了假设的层次结构，并倾向于选择在层次结构中位置较高的假设。在本章中，我们描述了进一步泛化这一想法的PAC-Bayesian方法。在PAC-Bayesian方法中，通过定义假设类上的先验分布来表达先验知识。

31.1 PAC-Bayes Bounds

正如在MDL范式中，我们在类别$\mathcal{H}$中定义了一个假设的层次结构。现在，这个层次结构以$\mathcal{H}$上的先验分布的形式出现。也就是说，我们为每个$h \in \mathcal{H}$分配一个概率（如果$\mathcal{H}$是连续的，则为密度）$P(h) \geq 0$，并将$P(h)$称为h的先验得分。遵循贝叶斯推理方法，学习算法的输出不一定是单个假设。相反，学习过程定义了$\mathcal{H}$上的后验概率，我们用Q表示。在监督学习问题的背景下，其中$\mathcal{H}$包含从$\mathcal{X}$到$\mathcal{Y}$的函数，可以将Q视为定义如下随机预测规则：每当得到一个新的实例$\mathbf{x}$时，我们根据Q随机选择一个假设$h \in \mathcal{H}$并预测$h(\mathbf{x})$。我们定义Q在示例z上的损失为

$$\ell(Q, z) \stackrel{\text{def}}{=} \mathop{\mathbb{E}}_{h \sim Q}[\ell(h, z)].$$

根据期望的线性，Q 的一般化损失和训练损失可以表示为

$$L_{\mathcal{D}}(Q) \stackrel{\text{def}}{=} \mathop{\mathbb{E}}_{h \sim Q}[L_{\mathcal{D}}(h)] \quad \text{and} \quad L_S(Q) \stackrel{\text{def}}{=} \mathop{\mathbb{E}}_{h \sim Q}[L_S(h)].$$

以下定理告诉我们，后验 Q 的一般化损失与经验损失之间的差异被一个依赖于 Q 与先验分布 P 之间的 Kullback-Leibler 散度的表达式所界定。Kullback-Leibler 是两个分布之间距离的自然度量。该定理表明，如果我们想最小化 Q 的一般化损失，我们应该联合最小化 Q 的经验损失以及 Q 与先验分布之间的 Kullback-Leibler 距离。我们将

稍后展示在某些情况下这一想法如何导致正则化风险最小化原理。

THEOREM 31.1 *Let $\mathcal{D}$ be an arbitrary distribution over an example domain Z. Let $\mathcal{H}$ be a hypothesis class and let $\ell : \mathcal{H} \times Z \to [0, 1]$ be a loss function. Let P be a prior distribution over $\mathcal{H}$ and let $\delta \in (0, 1)$. Then, with probability of at least $1 - \delta$ over the choice of an i.i.d. training set $S = \{z_1, \ldots, z_m\}$ sampled according to $\mathcal{D}$, for all distributions Q over $\mathcal{H}$ (even such that depend on S), we have*

$$L_{\mathcal{D}}(Q) \leq L_S(Q) + \sqrt{\frac{D(Q||P) + \ln m/\delta}{2(m-1)}},$$

where

$$D(Q||P) \stackrel{\text{def}}{=} \mathop{\mathbb{E}}_{h\sim Q}[\ln(Q(h)/P(h))]$$

is the Kullback-Leibler divergence.

Proof 对于任何函数 $f(S)$，使用马尔可夫不等式：

$$\mathop{\mathbb{P}}_{S}[f(S) \geq \epsilon] = \mathop{\mathbb{P}}_{S}[e^{f(S)} \geq e^{\epsilon}] \leq \frac{\mathbb{E}_S[e^{f(S)}]}{e^{\epsilon}}. \tag{31.1}$$

让 $\Delta(h) = L_{\mathcal{D}}(h) - L_S(h)$我们将应用公式（31.1）与fu 函数

$$f(S) = \sup_{Q} \left(2(m-1) \mathop{\mathbb{E}}_{h\sim Q} (\Delta(h))^2 - D(Q||P) \right).$$

我们现在转向有界 $\mathbb{E}_S[e^{f(S)}]$。主要技巧是使用一个不依赖于 Q 但依赖于先验概率 P的表达式来上界 $f(S)$。为此，固定一些 S 并注意，根据 $D(Q||P)$ 的定义，我们得到对于所有 Q，

$$\begin{aligned} 2(m-1) \mathop{\mathbb{E}}_{h\sim Q} (\Delta(h))^2 - D(Q||P) &= \mathop{\mathbb{E}}_{h\sim Q}[\ln(e^{2(m-1)\Delta(h)^2} P(h)/Q(h))] \\ &\leq \ln \mathop{\mathbb{E}}_{h\sim Q}[e^{2(m-1)\Delta(h)^2} P(h)/Q(h)] \\ &= \ln \mathop{\mathbb{E}}_{h\sim P}[e^{2(m-1)\Delta(h)^2}], \end{aligned} \tag{31.2}$$

其中不等式由 Jensen 不等式和对数函数的凹性得出。因此，

$$\mathop{\mathbb{E}}_{S}[e^{f(S)}] \leq \mathop{\mathbb{E}}_{S} \mathop{\mathbb{E}}_{h\sim P}[e^{2(m-1)\Delta(h)^2}]. \tag{31.3}$$

右侧表达式的优势源于我们可以交换期望的顺序（因为 P是一个不依赖于 S 的先验），从而得到

$$\mathop{\mathbb{E}}_{S}[e^{f(S)}] \leq \mathop{\mathbb{E}}_{h\sim P} \mathop{\mathbb{E}}_{S}[e^{2(m-1)\Delta(h)^2}]. \tag{31.4}$$

接下来，我们声称对于所有的 h，我们有 $\mathbb{E}_S[e^{2(m-1)\Delta(h)^2}] \leq m$。为了做到这一点，回忆一下Hoeffding不等式告诉我们

$$\mathop{\mathbb{P}}_{S}[\Delta(h) \geq \epsilon] \leq e^{-2m\epsilon^2}.$$

这表明 $\mathbb{E}_S[e^{2(m-1)\Delta(h)^2}] \leq m$ (参见练习1)。将此与方程(31.4)结合并代入方程(31.1)，我们得到

$$\mathop{\mathbb{P}}_{S}[f(S) \geq \epsilon] \;\leq\; \frac{m}{e^{\epsilon}}. \tag{31.5}$$

表示上述 δ 的右侧，因此 $\epsilon = \ln(m/\delta)$，因此我们得到，至少以 $-\delta$ 的概率，对于所有 Q，我们有

$$2(m-1) \mathop{\mathbb{E}}_{h\sim Q}(\Delta(h))^2 - D(Q||P) \leq \epsilon = \ln(m/\delta).$$

重新排列不等式并再次使用Jensen不等式（函数 x^2 是凸函数），我们得出结论

$$\left(\mathop{\mathbb{E}}_{h\sim Q}\Delta(h)\right)^2 \leq \mathop{\mathbb{E}}_{h\sim Q}(\Delta(h))^2 \leq \frac{\ln(m/\delta) + D(Q||P)}{2(m-1)}. \tag{31.6}$$

□

Remark 31.1 (正则化) PAC-Bayes 界导致以下学习规则：

给定先验 P，返回一个后验 Q，该后验使函数最小化

$$L_S(Q) + \sqrt{\frac{D(Q||P) + \ln m/\delta}{2(m-1)}}. \tag{31.7}$$

此规则类似于 *regularized risk minimization* 原则。也就是说，我们共同最小化样本上 Q 的经验损失以及 Q 和 P 之间的 Kullback-Leibler “距离”。

31.2 Bibliographic Remarks

PAC-Bayes 界限首先由 McAllester (1998) 提出。另见 (McAllester 1999, McAllester 2003, Seeger 2003, Langford & Shawe-Taylor 2003, Langford 2006)。

31.3 Exercises

1. 令 X 为满足 $\mathbb{P}[X \geq \epsilon] \leq e^{-2m\epsilon^2}$ 的随机变量。证明 $\mathbb{E}[e^{2(m-1)X^2}] \leq m$。

2. • 假设 $\mathcal{H}$ 是一个有限假设类，将先验设置为在 $\mathcal{H}$ 上均匀，将后验设置为对于某些 h_S 和 $Q(h) =$ 为 $Q(h_S) = 1$，对于所有其他 $h \in \mathcal{H}$ 为 0。证明

$$L_{\mathcal{D}}(h_S) \leq L_S(h) + \sqrt{\frac{\ln(|\mathcal{H}|) + \ln(m/\delta)}{2(m-1)}}.$$

与使用一致收敛性推导出的界限进行比较。

- 推导类似于第7章中给出的Occam界限的界限，使用PAC-Bayes界限

Appendix A Technical Lemmas

LEMMA A.1 *Let $a > 0$. Then: $x \geq 2a\log(a) \Rightarrow\ x \geq a\log(x)$. It follows that a necessary condition for the inequality $x < a\log(x)$ to hold is that $x < 2a\log(a)$.*

Proof 首先注意，对于 $a \in (0, \sqrt{e}\,]$ 不等式 $x \geq a\log(x))$ 成立，因此该命题是平凡的。从现在开始，假设 $a > \sqrt{e}$。考虑函数 $f(x) = x - a\log(x)$。导数是 $f'(x) = 1 - a/x$。因此，对于 $x > a$ 导数是正的，函数是增加的。此外，

$$\begin{aligned} f(2a\log(a)) &= 2a\log(a) - a\log(2a\log(a)) \\ &= 2a\log(a) - a\log(a) - a\log(2\log(a)) \\ &= a\log(a) - a\log(2\log(a)). \end{aligned}$$

由于对于所有 $a > > 0$，有 $a - 2\log(a) > 0$，证明如下。 □

LEMMA A.2 *Let $a \geq 1$ and $b > 0$. Then: $x \geq 4a\log(2a)+2b\ \Rightarrow\ x \geq a\log(x)+b$.*

Proof 只需证明 $x\ \geq 4a\log(2a) + 2b$ 蕴含着 $x\ \geq 2a\log(x)$ 和 $x\ \geq 2b$。由于我们假设 $a\ \geq 1$，我们显然有 $x\ \geq 2b$。此外，由于 $b > 0$，我们有 $x \geq 4a\log(2a)$，根据引理A.1，这蕴含着 $x \geq 2a\log(x)$。这就完成了我们的证明。

□

LEMMA A.3 *Let X be a random variable and $x' \in \mathbb{R}$ be a scalar and assume that there exists $a > 0$ such that for all $t \geq 0$ we have $\mathbb{P}[|X - x'| > t] \leq 2e^{-t^2/a^2}$. Then, $\mathbb{E}[|X - x'|] \leq 4\,a$.*

Proof 对于所有 $i = 0, 1, 2, \ldots$ 表示 $t_i = a\,i$。由于 t_i 是单调递增的，因此我们有 $\mathbb{E}[|X - x'|]$ 至多为 $\sum_{i=1}^{\infty} t_i\,\mathbb{P}[|X - x'| > t_{i-1}]$。结合引理中的假设，我们得到 $\mathbb{E}[|X - x'|] \leqslant 2\,a\,\sum_{i=1}^{\infty} ie^{-(i-1)^2}$。证明现在从以下不等式得出

$$\sum_{i=1}^{\infty} ie^{-(i-1)^2} \leq\ \sum_{i=1}^{5} ie^{-(i-1)^2} + \int_{5}^{\infty} xe^{-(x-1)^2}dx <\ 1.8 + 10^{-7} < 2\ .$$

□

LEMMA A.4 *Let X be a random variable and $x' \in \mathbb{R}$ be a scalar and assume that there exists $a > 0$ and $b \geq e$ such that for all $t \geq 0$ we have $\mathbb{P}[|X - x'| > t] \leq 2b\,e^{-t^2/a^2}$. Then, $\mathbb{E}[|X - x'|] \leq\ a(2 + \sqrt{\log(b)})$.*

Proof 对于所有 $i = 0, 1, 2, \ldots$ 表示 $t_i = a\,(i+\sqrt{\log(b)})$。由于 t_i 是单调递增的，因此我们有

$$\mathbb{E}[|X - x'|] \leq a\sqrt{\log(b)} + \sum_{i=1}^{\infty} t_i\,\mathbb{P}[|X - x'| > t_{i-1}].$$

使用引理中的假设，我们有

$$\begin{aligned}
\sum_{i=1}^{\infty} t_i\,\mathbb{P}[|X - x'| > t_{i-1}] &\leq 2\,a\,b\,\sum_{i=1}^{\infty}(i+\sqrt{\log(b)})e^{-(i-1+\sqrt{\log(b)})^2} \\
&\leq 2\,a\,b\,\int_{1+\sqrt{\log(b)}}^{\infty} x e^{-(x-1)^2}dx \\
&= 2\,a\,b\,\int_{\sqrt{\log(b)}}^{\infty}(y+1)e^{-y^2}dy \\
&\leq 4\,a\,b\,\int_{\sqrt{\log(b)}}^{\infty} y e^{-y^2}dy \\
&= 2\,a\,b\,\left[-e^{-y^2}\right]_{\sqrt{\log(b)}}^{\infty} \\
&= 2\,a\,b/b = 2\,a.
\end{aligned}$$

C结合前面的不等式，我们得出结论：{v*}　　oof. □

LEMMA A.5 *Let m, d be two positive integers such that $d \leq m - 2$. Then,*

$$\sum_{k=0}^{d}\binom{m}{k} \leq \left(\frac{e\,m}{d}\right)^d.$$

Proof 我们通过归纳法证明这个命题。对于 $d = 1$，左边等于 $1 + m$，而右边等于 em；因此命题成立。假设命题对于 d 成立，让我们证明它对于 $d + 1$ 成立。根据归纳假设，我们有

$$\begin{aligned}
\sum_{k=0}^{d+1}\binom{m}{k} &\leq \left(\frac{e\,m}{d}\right)^d + \binom{m}{d+1} \\
&= \left(\frac{e\,m}{d}\right)^d\left(1 + \left(\frac{d}{e\,m}\right)^d\frac{m(m-1)(m-2)\cdots(m-d)}{(d+1)d!}\right) \\
&\leq \left(\frac{em}{d}\right)^d\left(1 + \left(\frac{d}{e}\right)^d\frac{(m-d)}{(d+1)d!}\right).
\end{aligned}$$

使用Stirling近似，我们进一步有：{v*}

$$\begin{aligned}
&\le \left(\frac{e\,m}{d}\right)^d \left(1 + \left(\frac{d}{e}\right)^d \frac{(m-d)}{(d+1)\sqrt{2\pi d}(d/e)^d}\right) \\
&= \left(\frac{e\,m}{d}\right)^d \left(1 + \frac{m-d}{\sqrt{2\pi d}(d+1)}\right) \\
&= \left(\frac{e\,m}{d}\right)^d \cdot \frac{d+1+(m-d)/\sqrt{2\pi d}}{d+1} \\
&\le \left(\frac{e\,m}{d}\right)^d \cdot \frac{d+1+(m-d)/2}{d+1} \\
&= \left(\frac{e\,m}{d}\right)^d \cdot \frac{d/2+1+m/2}{d+1} \\
&\le \left(\frac{e\,m}{d}\right)^d \cdot \frac{m}{d+1},
\end{aligned}$$

在最后一个不等式中，我们使用了假设 $d \le m-2$。另一方面，

$$\begin{aligned}
\left(\frac{e\,m}{d+1}\right)^{d+1} &= \left(\frac{e\,m}{d}\right)^d \cdot \frac{em}{d+1} \cdot \left(\frac{d}{d+1}\right)^d \\
&= \left(\frac{e\,m}{d}\right)^d \cdot \frac{em}{d+1} \cdot \frac{1}{(1+1/d)^d} \\
&\ge \left(\frac{e\,m}{d}\right)^d \cdot \frac{em}{d+1} \cdot \frac{1}{e} \\
&= \left(\frac{e\,m}{d}\right)^d \cdot \frac{m}{d+1},
\end{aligned}$$

这证明了我们的归纳论证。 □

LEMMA A.6 *For all $a \in \mathbb{R}$ we have*

$$\frac{e^a + e^{-a}}{2} \le e^{a^2/2}.$$

Proof 观察发现

$$e^a = \sum_{n=0}^{\infty} \frac{a^n}{n!}.$$

因此，

$$\frac{e^a + e^{-a}}{2} = \sum_{n=0}^{\infty} \frac{a^{2n}}{(2n)!},$$

和

$$e^{a^2/2} = \sum_{n=0}^{\infty} \frac{a^{2n}}{2^n\, n!}.$$

观察$(2n)! \ge 2^n\, n!$ 对于每个 $n \ge 0$，我们得出我们的证明。 □

Appendix B Measure Concentration

设 $Z_1, \ldots, Z_m$ 为独立同分布的随机变量序列，设 μ 为它们的均值。大数定律表明，当 m 趋向于无穷大时，经验平均数 $\frac{1}{m}\sum_{i=1}^{m} Z_i$ 以概率 1 收敛于期望值 μ。测度集中不等式量化了当 m 为有限时，经验平均数与期望值之间的偏差。

B.1 Markov's Inequality

我们从以下不等式开始，这个不等式被称为马尔可夫不等式。设 Z 为一个非负随机变量。Z 的期望可以表示如下：

$$\mathbb{E}[Z] = \int_{x=0}^{\infty} \mathbb{P}[Z \geq x] dx. \tag{B.1}$$

由于 $\mathbb{P}[Z \geq x]$ 是单调非增的，我们得到 n

$$\forall a \geq 0, \quad \mathbb{E}[Z] \geq \int_{x=0}^{a} \mathbb{P}[Z \geq x] dx \geq \int_{x=0}^{a} \mathbb{P}[Z \geq a] dx = a\, \mathbb{P}[Z \geq a]. \tag{B.2}$$

重排不等式得到马尔可夫不等式：

$$\forall a \geq 0, \quad \mathbb{P}[Z \geq a] \;\leq\; \frac{\mathbb{E}[Z]}{a}. \tag{B.3}$$

对于取值在[0, 1]中的随机变量，我们可以从马尔可夫不等式推导出以下结论。

LEMMA B.1 *Let Z be a random variable that takes values in $[0, 1]$. Assume that $\mathbb{E}[Z] = \mu$. Then, for any $a \in (0, 1)$,*

$$\mathbb{P}[Z > 1 - a] \geq \frac{\mu - (1 - a)}{a}.$$

This also implies that for every $a \in (0, 1)$,

$$\mathbb{P}[Z > a] \geq \frac{\mu - a}{1 - a} \geq \mu - a.$$

Proof 让 $Y = 1 - Z$。然后 Y 是一个非负随机变量，满足 $\mathbb{E}[Y] = 1 - \mathbb{E}[Z] = 1 - \mu$。在 Y 上应用马尔可夫不等式，我们得到

$$\mathbb{P}[Z \leq 1 - a] = \mathbb{P}[1 - Z \geq a] = \mathbb{P}[Y \geq a] \leq \frac{\mathbb{E}[Y]}{a} = \frac{1 - \mu}{a}.$$

因此，

$$\mathbb{P}[Z > 1 - a] \geq 1 - \frac{1-\mu}{a} = \frac{a+\mu-1}{a}.$$

□

B.2 Chebyshev's Inequality

应用马尔可夫不等式于随机变量$(Z - \mathbb{E}[Z])^2$，我们得到切比雪夫不等式：

$$\forall a > 0, \quad \mathbb{P}[|Z - \mathbb{E}[Z]| \geq a] = \mathbb{P}[(Z - \mathbb{E}[Z])^2 \geq a^2] \leq \frac{\mathrm{Var}[Z]}{a^2}, \tag{B.4}$$

在 $\mathrm{Var}[Z] = \mathbb{E}[(Z - \mathbb{E}[Z])^2]$ 是 Z 的方差。

考虑随机变量$\frac{1}{m}\sum_{i=1}^{m} Z_i$。由于 $Z_1, \ldots, Z_m$ 是独立同分布的，因此很容易验证

$$\mathrm{Var}\left[\frac{1}{m}\sum_{i=1}^{m} Z_i\right] = \frac{\mathrm{Var}[Z_1]}{m}.$$

应用切比雪夫不等式，我们得到以下结果：

LEMMA B.2 *Let $Z_1, \ldots, Z_m$ be a sequence of i.i.d. random variables and assume that $\mathbb{E}[Z_1] = \mu$ and $\mathrm{Var}[Z_1] \leq 1$. Then, for any $\delta \in (0, 1)$, with probability of at least $1 - \delta$ we have*

$$\left|\frac{1}{m}\sum_{i=1}^{m} Z_i - \mu\right| \leq \sqrt{\frac{1}{\delta\, m}}.$$

Proof 应用切比雪夫不等式，我们得到对于所有 $a > 0$

$$\mathbb{P}\left[\left|\frac{1}{m}\sum_{i=1}^{m} Z_i - \mu\right| > a\right] \leq \frac{\mathrm{Var}[Z_1]}{m\, a^2} \leq \frac{1}{m\, a^2}.$$

证明通过表示右侧 δ 并求解 a 来进行。 □

经验平均值与之前给出的平均值之间的偏差随着 m 多项式下降。可以获得显著更快的下降。在接下来的章节中，我们推导出指数级下降的界限。

B.3 Chernoff's Bounds

让 $Z_1, \ldots, Z_m$ 为独立的伯努利变量，其中对于每个 i，$\mathbb{P}[Z_i = 1] = p_i$ 并且 $\mathbb{P}[Z_i = 0] = 1 - p_i$。设 $p = \sum_{i=1}^{m} p_i$ 和设 $Z = \sum_{i=1}^{m} Z_i$。使用

单调性指数函数和马尔可夫不等式，我们有对于每一个 $t > 0$

$$\mathbb{P}[Z > (1+\delta)p] = \mathbb{P}[e^{tZ} > e^{t(1+\delta)p}] \leq \frac{\mathbb{E}[e^{tZ}]}{e^{(1+\delta)tp}}. \tag{B.5}$$

Next,

$$\begin{aligned}
\mathbb{E}[e^{tZ}] = \mathbb{E}[e^{t\sum_i Z_i}] &= \mathbb{E}[\prod_i e^{tZ_i}] \\
&= \prod_i \mathbb{E}[e^{tZ_i}] && \text{by independence} \\
&= \prod_i \left(p_i e^t + (1-p_i)e^0\right) \\
&= \prod_i \left(1 + p_i(e^t - 1)\right) \\
&\leq \prod_i e^{p_i(e^t-1)} && \text{using } 1 + x \leq e^x \\
&= e^{\sum_i p_i(e^t-1)} \\
&= e^{(e^t-1)p}.
\end{aligned}$$

将上述内容与方程（B.5）结合，并选择 $t = \log(1+\delta)$，我们得到

LEMMA B.3 *Let $Z_1, \ldots, Z_m$ be independent Bernoulli variables where for every i, $\mathbb{P}[Z_i = 1] = p_i$ and $\mathbb{P}[Z_i = 0] = 1 - p_i$. Let $p = \sum_{i=1}^m p_i$ and let $Z = \sum_{i=1}^m Z_i$. Then, for any $\delta > 0$,*

$$\mathbb{P}[Z > (1+\delta)p] \;\leq\; e^{-h(\delta)\,p},$$

where

$$h(\delta) = (1+\delta)\log(1+\delta) - \delta.$$

使用不等式 $h(a) \geq a^2/(\text{\^{}}2 + 2a/\text{\^{}}3)$ 我们得到

LEMMA B.4 *Using the notation of Lemma B.3 we also have*

$$\mathbb{P}[Z > (1+\delta)p] \;\leq\; e^{-p\,\frac{\delta^2}{2+2\delta/3}}.$$

对于另一个方向，我们应用类似的计算：

$$\mathbb{P}[Z < (1-\delta)p] = \mathbb{P}[-Z > -(1-\delta)p] = \mathbb{P}[e^{-tZ} > e^{-t(1-\delta)p}] \leq \frac{\mathbb{E}[e^{-tZ}]}{e^{-(1-\delta)tp}}, \tag{B.6}$$

并且,

$$\begin{aligned}\mathbb{E}[e^{-tZ}] = \mathbb{E}[e^{-t\sum_i Z_i}] &= \mathbb{E}[\prod_i e^{-tZ_i}] \\ &= \prod_i \mathbb{E}[e^{-tZ_i}] && \text{by independence} \\ &= \prod_i \left(1 + p_i(e^{-t} - 1)\right) \\ &\leq \prod_i e^{p_i(e^{-t}-1)} && \text{using } 1 + x \leq e^x \\ &= e^{(e^{-t}-1)p}.\end{aligned}$$

设置 $t = -\log(1-\delta)$ 得到

$$\mathbb{P}[Z < (1-\delta)p] \;\leq\; \frac{e^{-\delta p}}{e^{(1-\delta)\log(1-\delta)\,p}} = e^{-ph(-\delta)}.$$

它很容易验证 $h(-\delta) \geq h(\delta)$ 因此

LEMMA B.5 *Using the notation of Lemma B.3 we also have*

$$\mathbb{P}[Z < (1-\delta)p] \;\leq\; e^{-ph(-\delta)} \leq e^{-ph(\delta)} \leq e^{-p\,\frac{\delta^2}{2+2\delta/3}}.$$

B.4 Hoeffding's Inequality

LEMMA B.6 (Höfding的不等式) *Let $Z_1, \ldots, Z_m$ be a sequence of i.i.d. random variables and let $\bar{Z} = \frac{1}{m}\sum_{i=1}^m Z_i$. Assume that $\mathbb{E}[\bar{Z}] = \mu$ and $\mathbb{P}[a \leq Z_i \leq b] = 1$ for every i. Then, for any $\epsilon > 0$*

$$\mathbb{P}\left[\left|\frac{1}{m}\sum_{i=1}^m Z_i - \mu\right| > \epsilon\right] \;\leq\; 2\,\exp\left(-2\,m\,\epsilon^2/(b-a)^2\right).$$

Proof 表示 $X_i = Z_i - \mathbb{E}[Z_i]$ 和 $\bar{X} = \frac{1}{m}\sum_i X_i$。利用指数函数的单调性和马尔可夫不等式，我们有对于每一个 $\lambda > 0$ 和 $\epsilon > 0$,

$$\mathbb{P}[\bar{X} \geq \epsilon] = \mathbb{P}[e^{\lambda\bar{X}} \geq e^{\lambda\epsilon}] \leq e^{-\lambda\epsilon}\,\mathbb{E}[e^{\lambda\bar{X}}].$$

使用独立性假设，我们也有

$$\mathbb{E}[e^{\lambda\bar{X}}] = \mathbb{E}\left[\prod_i e^{\lambda X_i/m}\right] = \prod_i \mathbb{E}[e^{\lambda X_i/m}].$$

通过Höffding引理（稍后引理B.7），对于每个i，我们有

$$\mathbb{E}[e^{\lambda X_i/m}] \leq e^{\frac{\lambda^2(b-a)^2}{8m^2}}.$$

因此,

$$\mathbb{P}[\bar{X} \geq \epsilon] \leq e^{-\lambda\epsilon} \prod_i e^{\frac{\lambda^2(b-a)^2}{8m^2}} = e^{-\lambda\epsilon + \frac{\lambda^2(b-a)^2}{8m}}.$$

设置 $\lambda = 4m\epsilon/(b-a)^2$ 我们得到

$$\mathbb{P}[\bar{X} \geq \epsilon] \leq e^{-\frac{2m\epsilon^2}{(b-a)^2}}.$$

应用相同的论据于变量 $-\bar{X}$，我们得到 $\mathbb{P}[\bar{X} \leq -\epsilon] \leq e^{-\frac{2m\epsilon^2}{(b-a)^2}}$。通过在两种情况上应用并集界，得出该定理。 □

LEMMA B.7 (Höfding引理) *Let X be a random variable that takes values in the interval $[a, b]$ and such that $\mathbb{E}[X] = 0$. Then, for every $\lambda > 0$,*

$$\mathbb{E}[e^{\lambda X}] \leq e^{\frac{\lambda^2(b-a)^2}{8}}.$$

Proof 由于 $f(x) = e^{\lambda x}$ 是一个凸函数，因此对于每个 $\alpha \in (0, 1)$，我们有 $x \in [a, b]$。

$$f(x) \leq \alpha f(a) + (1-\alpha) f(b).$$

设置 $\alpha = \frac{b-x}{b-a} \in [0, 1]$ 得到

$$e^{\lambda x} \leq \frac{b-x}{b-a} e^{\lambda a} + \frac{x-a}{b-a} e^{\lambda b}.$$

对期望进行计算，我们得到

$$\mathbb{E}[e^{\lambda X}] \leq \frac{b - \mathbb{E}[X]}{b-a} e^{\lambda a} + \frac{\mathbb{E}[x] - a}{b-a} e^{\lambda b} = \frac{b}{b-a} e^{\lambda a} - \frac{a}{b-a} e^{\lambda b},$$

在何处我们使用了事实 $\mathbb{E}[X] = 0$。定义 $h = \lambda(b-a)$，$p = \frac{-a}{b-a}$，和 $L(h) = -hp + \log(1 - p + pe^h)$。然后，上述等式右侧的表达式可以重写为 $e^{L(h)}$。因此，为了完成我们的证明，只需证明 $L(h) \leq \frac{h^2}{8}$。这可以通过泰勒定理以及以下事实得出：$L(0) = L'(0) = 0$ 和 $L''(h) \leq 1/4$ 对于所有 h 成立。 □

B.5 Bennet's and Bernstein's Inequalities

贝内特和伯恩斯坦的不等式类似于切诺夫界，但它们适用于任何独立随机变量的序列。我们在此不证明这些不等式，它们可以在Cesa-Bianchi & Lugosi (2006)中找到。

LEMMA B.8 (本尼特不等式) *Let $Z_1, \ldots, Z_m$ be independent random variables with zero mean, and assume that $Z_i \leq 1$ with probability 1. Let*

$$\sigma^2 \geq \frac{1}{m} \sum_{i=1}^{m} \mathbb{E}[Z_i^2].$$

Then for all $\epsilon > 0$,

$$\mathbb{P}\left[\sum_{i=1}^{m} Z_i > \epsilon\right] \leq e^{-m\sigma^2 h\left(\frac{\epsilon}{m\sigma^2}\right)}.$$

where

$$h(a) = (1+a)\log(1+a) - a.$$

通过使用不等式 $h(a) \geq a^2/(\hat{}2 + 2a/\hat{}3)$，可以推导出以下结果：

LEMMA B.9 (伯恩斯坦不等式) *Let* $Z_1, \ldots, Z_m$ *be i.i.d. random variables with a zero mean. If for all* i, $\mathbb{P}(|Z_i| < M) = 1$, *then for all* $t > 0$ *:*

$$\mathbb{P}\left[\sum_{i=1}^{m} Z_i > t\right] \leq \exp\left(-\frac{t^2/2}{\sum \mathbb{E}\, Z_j^2 + Mt/3}\right).$$

B.5.1 Application

伯恩斯坦不等式可用于在我们在可实现情况下（第2章）推导出的速率$1/\epsilon$和我们在不可实现情况下（第4章）推导出的速率$1/\epsilon^2$之间进行插值。

LEMMA B.10 *Let* $\ell : \mathcal{H} \times Z \to [0,1]$ *be a loss function. Let* $\mathcal{D}$ *be an arbitrary distribution over* Z*. Fix some* h*. Then, for any* $\delta \in (0,1)$ *we have*

1. $$\mathop{\mathbb{P}}_{S\sim\mathcal{D}^m}\left[L_S(h) \geq L_D(h) + \sqrt{\frac{2L_D(h)\log(1/\delta)}{3\,m}} + \frac{2\log(1/\delta)}{m}\right] \leq \delta$$
2. $$\mathop{\mathbb{P}}_{S\sim\mathcal{D}^m}\left[L_D(h) \geq L_S(h) + \sqrt{\frac{2L_S(h)\log(1/\delta)}{m}} + \frac{4\log(1/\delta)}{m}\right] \leq \delta$$

Proof 定义随机变量 $\alpha_1, \ldots, \alpha_m$ 使得 $\alpha_i = \ell(h, z_i) - L_D(h)$。注意 $\mathbb{E}[\alpha_i] = 0$ 以及

$$\begin{aligned}
\mathbb{E}[\alpha_i^2] &= \mathbb{E}[\ell(h,z_i)^2] - 2L_{\mathcal{D}}(h)\,\mathbb{E}[\ell(h,z_i)] + L_{\mathcal{D}}(h)^2 \\
&= \mathbb{E}[\ell(h,z_i)^2] - L_{\mathcal{D}}(h)^2 \\
&\leq \mathbb{E}[\ell(h,z_i)^2] \\
&\leq \mathbb{E}[\ell(h,z_i)] = L_{\mathcal{D}}(h),
\end{aligned}$$

在最后一个不等式中，我们使用了事实 $\ell(h, z_i) \in [0,1]$，因此 $\ell(h,z_i)^2 \leq \ell(h,z_i)$。对 α_i 应用 Bernsein 不等式得出

$$\begin{aligned}
\mathbb{P}\left[\sum_{i=1}^{m} \alpha_i > t\right] &\leq \exp\left(-\frac{t^2/2}{\sum \mathbb{E}\,\alpha_j^2 + t/3}\right) \\
&\leq \exp\left(-\frac{t^2/2}{m\,L_{\mathcal{D}}(h) + t/3}\right) \stackrel{\text{def}}{=} \delta.
\end{aligned}$$

求解 t 得到

$$\begin{aligned}
&\frac{t^2/2}{m\,L_{\mathcal{D}}(h)+t/3} = \log(1/\delta) \\
&\quad\Rightarrow\quad t^2/2 - \frac{\log(1/\delta)}{3}\,t - \log(1/\delta)\,m\,L_{\mathcal{D}}(h) = 0 \\
&\quad\Rightarrow\quad t = \frac{\log(1/\delta)}{3} + \sqrt{\frac{\log^2(1/\delta)}{3^2} + 2\log(1/\delta)\,m\,L_{\mathcal{D}}(h)} \\
&\quad\leq 2\,\frac{\log(1/\delta)}{3} + \sqrt{2\log(1/\delta)\,m\,L_{\mathcal{D}}(h)}
\end{aligned}$$

自從 $\frac{1}{m}\sum_i \;\alpha_i = L_S(h) - L_{\mathcal{D}}(h)$, 至少 $1-\delta$,

$$L_S(h) - L_{\mathcal{D}}(h) \leq 2\,\frac{\log(1/\delta)}{3m} + \sqrt{\frac{2\log(1/\delta)\,L_{\mathcal{D}}(h)}{m}},$$

这证明了第一个不等式。引理的第二部分以类似的方式得出。

□

B.6 Slud's Inequality

设 X 为一个 (m,p) 二项变量。即 $X = \sum_{i=1}^{m} Z_i$，其中每个 Z_i 以概率 p 为 1，以概率 $1-p$ 为 0。假设 $p = (1-\epsilon)/2$。Slud 不等式（Slud 1977）告诉我们 $\mathbb{P}[X \geq m/2]$ 的下界是正态变量大于或等于 $\sqrt{m\epsilon^2/(1-\epsilon^2)}$ 的概率。以下引理可以通过正态分布的标准尾部界限得出。

LEMMA B.11 *Let X be a (m,p) binomial variable and assume that $p = (1-\epsilon)/2$. Then,*

$$\mathbb{P}[X \geq m/2] \geq \frac{1}{2}\left(1 - \sqrt{1 - \exp(-m\epsilon^2/(1-\epsilon^2))}\right).$$

B.7 Concentration of χ^2 Variables

设 $X_1, \ldots, X_k$ 为 k 个相互独立的正态分布随机变量。即对于所有 i，$X_i \sim N(0,1)$。随机变量 X_i^2 的分布称为 χ^2 (卡方) 分布，而随机变量 $Z = X_1^2 + \cdots + X_k^2$ 的分布称为具有 k 自由度的 χ_k^2 (卡方) 分布。显然，$\mathbb{E}[X_i^2] = 1$ 和 $\mathbb{E}[Z] = k$。以下引理表明 X_k^2 集中在其均值附近。

LEMMA B.12 *Let $Z \sim \chi_k^2$. Then, for all $\epsilon > 0$ we have*

$$\mathbb{P}[Z \leq (1-\epsilon)k] \leq e^{-\epsilon^2 k/6},$$

and for all $\epsilon \in (0,3)$ we have

$$\mathbb{P}[Z \geq (1+\epsilon)k] \leq e^{-\epsilon^2 k/6}.$$

Finally, for all $\epsilon \in (0\{v^2\}\ 3\{v^3\}$

$$\mathbb{P}\left[(1-\epsilon)k \leq Z \leq (1+\epsilon)k\right] \geq 1 - 2e^{-\epsilon^2 k/6}.$$

Proof 让我们写出 $Z = \sum_{i=1}^{k} X_i^2$，其中 $X_i \sim N(0,1)$。为了证明两个界限，我们使用切诺夫界法。对于第一个不等式，我们首先界定 $\mathbb{E}[e^{-\lambda X_1^2}]$，其中 $\lambda > 0$ 将在稍后指定。由于对于所有 $a \geq 0$，有 $e^{-a} \leq 1 - a + \frac{a^2}{2}$，因此我们得到

$$\mathbb{E}[e^{-\lambda X_1^2}] \ \leq \ 1 - \lambda\,\mathbb{E}[X_1^2] + \frac{\lambda^2}{2}\,\mathbb{E}[X_1^4].$$

使用已知的等式，$\mathbb{E}[X_1^2] = 1$ 和 $\mathbb{E}[X_1^4] = 3$，以及 $1 - a \leq e^{-a}$ 的事实，我们得到

$$\mathbb{E}[e^{-\lambda X_1^2}] \ \leq \ 1 - \lambda + \tfrac{3}{2}\lambda^2 \leq e^{-\lambda + \frac{3}{2}\lambda^2}.$$

现在，应用切诺夫界法，我们得到

$$\begin{aligned}
\mathbb{P}[-Z \geq -(1-\epsilon)k] &= \mathbb{P}\left[e^{-\lambda Z} \geq e^{-(1-\epsilon)k\lambda}\right] \\
&\leq e^{(1-\epsilon)k\lambda}\ \mathbb{E}\left[e^{-\lambda Z}\right] \\
&= e^{(1-\epsilon)k\lambda}\ \left(\mathbb{E}\left[e^{-\lambda X_1^2}\right]\right)^k \\
&\leq e^{(1-\epsilon)k\lambda}\, e^{-\lambda k + \frac{3}{2}\lambda^2 k} \\
&= e^{-\epsilon k\lambda + \frac{3}{2}k\lambda^2}.
\end{aligned}$$

选择 $\lambda = \epsilon/3$，我们得到引理中所述的第一个不等式。

对于第二个不等式，我们使用一个已知的闭式表达式来表示 χ_k^2 分布的随机变量的矩生成函数：

$$\forall \lambda < \tfrac{1}{2}, \quad \mathbb{E}\left[e^{\lambda Z^2}\right] \ = \ (1-2\lambda)^{-k/2}. \tag{B.7}$$

基于 方程式并使用切诺夫界法 我们有

$$\begin{aligned}
\mathbb{P}[Z \geq (1+\epsilon)k)] &= \mathbb{P}\left[e^{\lambda Z} \geq e^{(1+\epsilon)k\lambda}\right] \\
&\leq e^{-(1+\epsilon)k\lambda}\ \mathbb{E}\left[e^{\lambda Z}\right] \\
&= e^{-(1+\epsilon)k\lambda}\ (1-2\lambda)^{-k/2} \\
&\leq e^{-(1+\epsilon)k\lambda}\, e^{k\lambda} = e^{-\epsilon k\lambda},
\end{aligned}$$

在最后一个不等式发生的地方，因为（$1 - a$）$\leq e^{-a}$。设定$\lambda = \epsilon/6$（根据我们的假设在（$0, 1/2$）中）我们获得引理中所述的第二个不等式。

最后，最后一个不等式由前两个不等式和并集界得出。

□

Appendix C Linear Algebra

C.1 Basic Definitions

在这一章中，我们只讨论有限维欧几里得空间上的线性代数。我们将向量称为列向量。

给定两个 d 维向量 $\mathbf{u}, \mathbf{v} \in \mathbb{R}^d$，它们的内积是

$$\langle \mathbf{u}, \mathbf{v} \rangle = \sum_{i=1}^{d} u_i v_i.$$

欧几里得范数（又称ℓ_2范数）是$\|\mathbf{u}\| = \sqrt{\langle \mathbf{u}, \mathbf{u} \rangle}$。我们还使用$\ell_1$范数、$\|\mathbf{u}\|_1 = \sum_{i=1}^{d} |u_i|$和$\ell_\infty$范数$\|\mathbf{u}\|_\infty = \max_i |u_i|$。

一个 $\mathbb{R}^d$ 的子空间是 $\mathbb{R}^d$ 的一个子集，该子集在加法和数乘下是封闭的。向量集 $\mathbf{u}_1, \ldots, \mathbf{u}_k$ 的生成空间是包含所有形式为的向量的子空间

$$\sum_{i=1}^{k} \alpha_i \mathbf{u}_i$$

对于所有 i，$\alpha_i \in \mathbb{R}$。

一个向量集 {v*} 是独立的，如果对于每个 {v*}，{v*} 不在 {v*} 的张成空间中。我们说 {v*} 张成一个子空间 {v*}，如果 {v*} 是 {v*} 中向量的张成。我们说 {v*} 是 {v*} 的 {v*}，如果它既是独立的又张成 {v*}。{v*} 的维度是 {v*} 的基的大小，并且可以验证 {v*} 的所有基具有相同的大小 {v*}。我们说 {v*} 是一个正交集，如果对于所有 {v*}，{v*} 0。我们说 {v*} 是一个正交归一集，如果它是正交的，并且对于每个 {v*}，{v*} 1。

给定一个矩阵 $A \in \mathbb{R}^{n,d}$，A 的范围是其列的生成空间，A 的零空间是满足 $A\mathbf{u} = \mathbf{0}$ 的所有向量的子空间。A 的秩是其范围的维度。

矩阵 A 的转置，记作 $A^\top$，是一个矩阵，其 (i, j) 项等于 A 的 (j, i) 项。我们说 A 是对称的，如果 $A = A^\top$。

C.2 Eigenvalues and Eigenvectors

设 $A \in \mathbb{R}^{d,d}$ 为一个矩阵。一个非零向量 $\mathbf{u}$ 是 A 的特征向量，对应特征值 λ，如果

$$A\mathbf{u} = \lambda\mathbf{u}.$$

THEOREM C.1（光谱分解）*If $A \in \mathbb{R}^{d,d}$ is a symmetric matrix of rank k, then there exists an orthonormal basis of $\mathbb{R}^d$, $\mathbf{u}_1, \ldots, \mathbf{u}_d$, such that each $\mathbf{u}_i$ is an eigenvector of A. Furthermore, A can be written as $A = \sum_{i=1}^{d} \lambda_i \mathbf{u}_i \mathbf{u}_i^\top$, where each λ_i is the eigenvalue corresponding to the eigenvector $\mathbf{u}_i$. This can be written equivalently as $A = UDU^\top$, where the columns of U are the vectors $\mathbf{u}_1, \ldots, \mathbf{u}_d$, and D is a diagonal matrix with $D_{i,i} = \lambda_i$ and for $i \neq j$, $D_{i,j} = 0$. Finally, the number of λ_i which are nonzero is the rank of the matrix, the eigenvectors which correspond to the nonzero eigenvalues span the range of A, and the eigenvectors which correspond to zero eigenvalues span the null space of A.*

C.3 Positive definite matrices

一个对称矩阵 $A \in \mathbb{R}^{d,d}$ 是正定的，如果它的所有特征值都是正的。A 是半正定的，如果它的所有特征值都是非负的。

THEOREM C.2 *Let $A \in \mathbb{R}^{d,d}$ be a symmetric matrix. Then, the following are equivalent definitions of positive semidefiniteness of A:*

- *All the eigenvalues of A are nonnegative.*
- *For every vector $\mathbf{u}$, $\langle \mathbf{u}, A\mathbf{u} \rangle \geq 0$.*
- *There exists a matrix B such that $A = BB^\top$.*

C.4 Singular Value Decomposition (SVD)

设 $A \in \mathbb{R}^{m,n}$ 为秩为 r 的矩阵。当 $m \neq n$ 时，定理 C.1 中给出的特征值分解不能应用。我们将描述 A 的另一种分解，称为奇异值分解，简称 SVD。

单位向量 $\mathbf{v} \in \mathbb{R}^n$ 和 $\mathbf{u} \in \mathbb{R}^m$ 被称为 A 的右 *singular vectors* 和左 *singular vectors*，如果相应的 *singular value* $\sigma >$ 为 0。

$$A\mathbf{v} = \sigma\mathbf{u} \quad \text{and} \quad A^\top\mathbf{u} = \sigma\mathbf{v}.$$

我们首先证明，如果我们能找到具有正奇异值的 r 个正交奇异向量，那么我们可以分解 $A = UDV^\top$，其中 U 的列包含左奇异向量，V 的列包含右奇异向量，而 D 是一个对角 $r \times r$ 矩阵，其对角线上的元素是奇异值。

LEMMA C.3 *Let $A \in \mathbb{R}^{m,n}$ be a matrix of rank r. Assume that $\mathbf{v}_1, \ldots, \mathbf{v}_r$ is an orthonormal set of right singular vectors of A, $\mathbf{u}_1, \ldots, \mathbf{u}_r$ is an orthonormal set of corresponding left singular vectors of A, and $\sigma_1, \ldots, \sigma_r$ are the corresponding singular values. Then,*

$$A = \sum_{i=1}^{r} \sigma_i \mathbf{u}_i \mathbf{v}_i^\top.$$

It follows that if U is a matrix whose columns are the $\mathbf{u}_i$'s, V is a matrix whose columns are the $\mathbf{v}_i$'s, and D is a diagonal matrix with $D_{i,i} = \sigma_i$, then

$$A = UDV^\top.$$

Proof 任何 A 的右奇异向量必须在 $A^\top$ (的范围内，否则奇异值必须为零)。因此，$\mathbf{v}_1, \ldots, \mathbf{v}_r$ 是 A 范围的一个正交归一基。让我们通过添加向量 $\mathbf{v}_{r+1}, \ldots, \mathbf{v}_n$ 来将其补充为一个正交归一基 $\mathbb{R}^n$。定义 $B = \sum_{i=1}^{r} \sigma_i \mathbf{u}_i \mathbf{v}_i^\top$。只需证明对所有 i，$A\mathbf{v}_i = B\mathbf{v}_i$。显然，如果 $i > r$，那么 $A\mathbf{v}_i = 0$，并且 $B\mathbf{v}_i = 0$ 也一样。对于 $i \leq r$，我们有

$$B\mathbf{v}_i = \sum_{j=1}^{r} \sigma_j \mathbf{u}_j \mathbf{v}_j^\top \mathbf{v}_i = \sigma_i \mathbf{u}_i = A\mathbf{v}_i,$$

最后等式由定义得出。 □

下一个引理将 A 的奇异值与 $A^\top A$ 和 $AA^\top$ 的特征值相关联。

LEMMA C.4 *$\mathbf{v}, \mathbf{u}$ are right and left singular vectors of A with singular value σ iff $\mathbf{v}$ is an eigenvector of $A^\top A$ with corresponding eigenvalue σ^2 and $\mathbf{u} = \sigma^{-1} A\mathbf{v}$ is an eigenvector of $AA^\top$ with corresponding eigenvalue σ^2.*

Proof 假设 σ 是 A 的一个奇异值，而 $\mathbf{v} \in \mathbb{R}^n$ 是相应的右奇异向量。那么，

$$A^\top A\mathbf{v} = \sigma A^\top \mathbf{u} = \sigma^2 \mathbf{v}.$$

同样地，

$$AA^\top u = \sigma A\mathbf{v} = \sigma^2 \mathbf{u}.$$

对于另一个方向，如果 $\lambda \neq 0$ 是 $A^\top A$ 的特征值，且 $\mathbf{v}$ 是对应的特征向量，那么 $\lambda > 0$ 因为 $A^\top A$ 是正半定矩阵。设 $\sigma = \sqrt{\lambda}, \mathbf{u} = \sigma^{-1} A\mathbf{v}$。然后，

$$\sigma\mathbf{u} = \sqrt{\lambda} \frac{A\mathbf{v}}{\sqrt{\lambda}} = A\mathbf{v},$$

和

$$A^\top \mathbf{u} = \frac{1}{\sigma} A^\top A\mathbf{v} = \frac{\lambda}{\sigma} \mathbf{v} = \sigma\mathbf{v}.$$

□

最后，我们证明如果 A 的秩为 r，则它具有 r 个正交归一的特征向量。

LEMMA C.5 *Let $A \in \mathbb{R}^{m,n}$ with rank r. Define the following vectors:*

$$\begin{aligned}
\mathbf{v}_1 &= \operatorname*{argmax}_{\mathbf{v}\in\mathbb{R}^n:\|\mathbf{v}\|=1} \|A\mathbf{v}\| \\
\mathbf{v}_2 &= \operatorname*{argmax}_{\substack{\mathbf{v}\in\mathbb{R}^n:\|\mathbf{v}\|=1 \\ \langle\mathbf{v},\mathbf{v}_1\rangle=0}} \|A\mathbf{v}\| \\
&\vdots \\
\mathbf{v}_r &= \operatorname*{argmax}_{\substack{\mathbf{v}\in\mathbb{R}^n:\|\mathbf{v}\|=1 \\ \forall i<r,\ \langle\mathbf{v},\mathbf{v}_i\rangle=0}} \|A\mathbf{v}\|
\end{aligned}$$

Then, $\mathbf{v}_1, \ldots, \mathbf{v}_r$ is an orthonormal set of right singular vectors of A.

Proof 首先注意，由于 A 的秩是 r，A 的范围是维度为 r 的子空间，因此很容易验证对于所有 $i = 1, \ldots, r$，$\|A\mathbf{v}_i\| > 0$。设 $W \in \mathbb{R}^{n,n}$ 是通过 $A^\top A$ 的特征值分解得到的正交矩阵，即 $A^\top A = WDW^\top$，其中 D 是对角矩阵，且 $D_{1,1} \geq D_{2,2} \geq \cdots \geq 0$。我们将证明 $\mathbf{v}_1, \ldots, \mathbf{v}_r$ 是与非零特征值对应的 $A^\top A$ 的特征向量，因此，根据引理 C.4，这些也是 A 的右奇异向量。证明采用归纳法。对于归纳的基础，注意任何单位向量 $\mathbf{v}$ 可以写成 $\mathbf{v} = W\mathbf{x}$，对于 $\mathbf{x} = W^\top\mathbf{v}$，并注意 $\|\mathbf{x}\| = 1$。因此，

$$\|A\mathbf{v}\|^2 = \|AW\mathbf{x}\|^2 = \|WDW^\top W\mathbf{x}\|^2 = \|WD\mathbf{x}\|^2 = \|D\mathbf{x}\|^2 = \sum_{i=1}^n D_{i,i}^2 {x_i}^2.$$

因此，

$$\max_{\mathbf{v}:\|\mathbf{v}\|=1} \|A\mathbf{v}\|^2 = \max_{\mathbf{x}:\|\mathbf{x}\|=1} \sum_{i=1}^n D_{i,i}^2 {x_i}^2.$$

解的右侧是设置 $\mathbf{x} = (1, 0, \ldots, 0)$，这意味着 $\mathbf{v}_1$ 是 $A^\top A$ 的第一个特征向量。由于 $\|A\mathbf{v}_1\| > 0$，因此 $D_{1,1} > 0$ 如所需。对于归纳步骤，假设对于某个 $1 \leq t \leq r-1$ 的命题成立。那么，任何与 $\mathbf{v}_1, \ldots, \mathbf{v}_t$ 正交的 $\mathbf{v}$ 可以写成 $\mathbf{v} = W\mathbf{x}$，其中 $\mathbf{x}$ 的所有第一个 t 个元素为零。因此，有

$$\max_{\mathbf{v}:\|\mathbf{v}\|=1,\forall i\leq t,\mathbf{v}^\top\mathbf{v}_i=0} \|A\mathbf{v}\|^2 = \max_{\mathbf{x}:\|\mathbf{x}\|=1} \sum_{i=t+1}^n D_{i,i}^2 {x_i}^2.$$

解的右侧是除了 $x_{t+1} = 1$ 以外的所有零向量。这表明 $\mathbf{v}_{t+1}$ 是 W 的 $(t+1)$ 列。最后，由于 $\|A\mathbf{v}_{t+1}\| > 0$，因此 $D_{t+1,t+1} > 0$ 如所需。这完成了我们的证明。

□

COROLLARY C.6（奇异值分解定理）*Let $A \in \mathbb{R}^{m,n}$ with rank r. Then $A = UDV^\top$ where D is an $r \times r$ matrix with nonzero singular values of A and the columns of U, V are orthonormal left and right singular vectors of A. Furthermore, for all i, $D_{i,i}^2$ is an eigenvalue of $A^\top A$, the ith column of V is the corresponding eigenvector of $A^\top A$ and the ith column of U is the corresponding eigenvector of $AA^\top$.*

Notes

References

Abernethy, J., Bartlett, P. L., Rakhlin, A. & Tewari, A. (2008), Optimal strategies and minimax l ower bounds for online convex games, *in* ‘第十九届计算学习理论年度会议论文集’. Acke rman, M. & Ben-David, S. (2008), Measures of clustering quality: A working set of axioms for c lustering, *in* ‘神经信息处理系统（NIPS）会议论文集’，第121–128页。Agarwal, S. & R oth, D. (2005), Learnability of bipartite ranking functions, *in* ‘第十八届学习理论年度会议论 文集’，第16–31页。Agmon, S. (1954), ‘线性不等式的松弛法’，*Canadian Journal of Mathematics* **6**(3)，第382–392页。Aizerman, M. A., Braverman, E. M. & Rozonoer, L. I. (1964), ‘模式识别学习中的势函数方法的理论基础’，*Automation and Remote Control* **25**，第821–837页。Allwein, E. L., Schapire, R. & Singer, Y. (2000), ‘将 多类问题转化为二类问题：边缘分类器的统一方法’，*Journal of Machine Learning Research* **1**，第113–141页。Alon, N., Ben-David, S., Cesa-B ianchi, N. & Haussler, D. (1997), ‘尺度敏感维度、一致收敛和可学习性’，*Journal of the ACM* **44**(4)，第615–631页。Anthony, M. & Bartlet, P. (1999), *Neural Network Learning: Theoretical Foundations*，剑桥大学出版社。Baraniuk, R., Da venport, M., DeVore, R. & Wakin, M. (2008), ‘随机矩阵的受限等距性质的简单证明’，*Constructive Approximation* **28**(3)，第253–263页。Barber, D. (2012), *Bayesian reasoning and machine learning*，剑桥大学出版社。Bartlett, P.，Bousquet, O. & Mendelson, S. (2005), ‘局部Rademacher复杂性’，*Annals of Statistics* **33**(4)，第149 7–1537页。Bartlett, P. L. & Ben-David, S. (2002), ‘神经网络逼近问题的难度结果’，*Theor. Comput. Sci.* **284**(1)，第53–66页。Bartlett, P. L. & Long, P. M. & Williamson, R. C. (1994)，Fat-shattering and the learnability of real-valued functions，*in* ‘第七届计算学习 理论年度会议论文集’，ACM，第299–310页。Bartlett, P. L. & Mendelson, S. (2001)，Ra demacher and Gaussian complexities: Risk bounds and structural results，*in* ‘第十四届计算 学习理论年度会议，COLT 2001’，Springer，柏林，第224–240页。

Bartlett, P. L. & Mendelson, S. (2002), ‘Rademacher和Gaussian复杂性：风险界限和结构结果’, *Journal of Machine Learning Research* **3**, 463–482. Ben-David, S., Cesa-Bianchi, N., Haussler, D. & Long, P. (1995), ‘0{值函数类学习可辨识性的特征’, *Journal of Computer and System Sciences* **50**, 74–86. Ben-David, S., Eiron, N. & Long, P. (2003), ‘关于近似最大化一致性的难度’, *Journal of Computer and System Sciences* **66**(3), 496–514. Ben-David, S. & Litman, A. (1998), ‘Vapnik-Chervonenkis类的组合变异性及其在样本压缩方案中的应用’, *Discrete Applied Mathe- matics* **86**(1), 3–25. Ben-David, S., Pal, D., & Shalev-Shwartz, S. (2009), 无监督在线学习, *in*‘学习理论会议（COLT）’. Ben-David, S. & Simon, H. (2001), ‘线性感知器的有效学习’, *Advances in Neural Information Processing Systems* pp. 189–195. Bengio, Y. (2009), ‘为AI学习深度架构’, *Foundations and Trends in Machine Learning* **2**(1), 1–127. Bengio, Y. & LeCun, Y. (2007), ‘将学习算法扩展到AI’, *Large-Scale Kernel Machines* **34**. Bertseka s, D. (1999), *Nonlinear Programming*, Athena Scientific. Beygelzimer, A., Langford, J. & Ravikumar, P. (2007), ‘使用过滤器树的多元分类’, *Preprint, June*. Birkhoff, G. (1946) , ‘线性代数中的三个观察’, *Revi. Univ. Nac. Tucuman, ser A* **5**, 147–151. Bishop, C. M. (2006), *Pattern recognition and machine learning*, Vol. 1, springer New York. Bl um, L., Shub, M. & Smale, S. (1989), ‘关于实数计算和复杂性的理论：Np-完全性，递归函数和通用机器’, *Am. Math. Soc* **21**(1), 1–46. Blumer, A., Ehrenfeucht, A., Haussl er, D. & Warmuth, M. K. (1987), ‘Occam的剃刀’, *Information Processing Letters* **24**(6), 377–380. Blumer, A., Ehrenfeucht, A., Haussler, D. & Warmuth, M. K. (1989), ‘可学习性和Vapnik-Chervonenkis维度’, *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. & Va pnik, V. N. (1992), 最优边缘分类器的训练算法, *in*‘学习理论会议（COLT）’, pp. 14 4–152. Bottou, L. & Bousquet, O. (2008), 大规模学习的权衡, *in*‘NIPS’, pp. 161–168. Boucheron, S., Bousquet, O. & Lugosi, G. (2005), ‘分类理论：近期进展综述’, *ESAIM: Probability and Statistics* **9**, 323–375. Bousquet, O. (2002), 集中不等式和经验过程理论在学习算法分析中的应用, 博士论文, Ecole Polytechnique. Bousquet, O. & Elisse eff, A. (2002), ‘稳定性和泛化’, *Journal of Machine Learning Research* **2**, 499–526. Boyd, S. & Vandenberghe, L. (2004), *Convex Optimization*, Cambridge University Press.

布赖曼，L.（1996），偏差、方差和弧形分类器，技术报告460，加州大学伯克利分校统计学系。布赖曼，L.（2001），“随机森林”，*Machine learning* **45**(1)，5–32。布赖曼，L.，弗里德曼，J. H.，奥尔申，R. A. & 斯通，C. J.（1984），*Classification and Regression Trees*，Wadsworth & Brooks。坎代斯，E.（2008），“受限等距性质及其在压缩感知中的应用”，*Comptes Rendus Mathematique* **346**(9)，589–592。坎代斯，E. J.（2006），压缩采样，*in*“国际数学大会，西班牙马德里”。坎代斯，E. & 陶，T.（2005），“线性规划解码”，*IEEE Trans. on Information Theory* **51**，4203–4215。塞萨-比安奇，N. & 卢戈西，G.（2006），*Prediction, learning, and games*，剑桥大学出版社。张，H. S.，魏斯，Y. & 弗里曼，W. T.（2009），“信息感知”，*arXiv preprint arXiv:0901.4275*。查佩勒，O.，李，Q. & 斯莫拉，A.（2007），排序度量的大间隔优化，*in*“NIPS研讨会：网络搜索中的机器学习”。柯林斯，M.（2000），自然语言解析的判别性重排序，*in*“机器学习”。柯林斯，M.（2002），隐藏马尔可夫模型的判别性训练方法：感知器算法的理论和实验，*in*“自然语言处理经验方法会议”。科洛贝尔，R. & 威斯顿，J.（2008），自然语言处理的统一架构：具有多任务学习的深度神经网络，*in*“国际机器学习会议（ICML）”。科尔特斯，C. & 瓦普尼克，V.（1995），“支持向量机”，*Machine Learning* **20**(3)，273–297。科弗，T.（1965），“二元序列顺序预测者的行为”，*Trans. 4th Prague Conf. Information Theory Statistical Decision Functions, Random Processes* pp. 263–272。科弗，T. & 哈特，P.（1967），“最近邻模式分类”，*Information Theory, IEEE Transactions on* **13**(1)，21–27。克拉默，K. & 辛格，Y.（2001），“关于多类核向量机算法实现的探讨”，*Journal of Machine Learning Research* **2**，265–292。克里斯蒂安尼尼，N. & 肖特-泰勒，J.（2000），*An Introduction to Support Vector Machines*，剑桥大学出版社。丹尼利，A.，萨巴托，S.，本-戴维，S. & 莎莱夫-施瓦茨，S.（2011），多类可学习性和erm原理，*in*“学习理论会议（COLT）”。丹尼利，A.，萨巴托，S. & 莎莱夫-施瓦茨，S. S.（2012），多类学习方法：理论比较及其影响，*in*“NIPS”。戴维斯，G.，马尔拉特，S. & 阿夫兰达，M.（1997），“贪婪自适应逼近”，*Jour- nal of Constructive Approximation* **13**，57–98。德罗伊，L. & 吉奥尔菲，L.（1985），*Nonparametric Density Estimation: The L B1 S View*，Wiley。德罗伊，L.，吉奥尔菲，L. & 卢戈西，G.（1996），*A Probabilistic Theory of Pattern Recog- nition*，Springer。

Dietterich, T. G. & Bakiri, G. (1995), ‘通过错误纠正输出码解决多类学习问题’, *Journal of Artificial Intelligence Research* **2**, 263–286. Donoho, D. L. (2006), ‘压缩感知’, *Information Theory, IEEE Transactions on* **52**(4), 1289–1306. Dudley, R., Gine, E. & Zinn, J. (1991), ‘一致和通用的Glivenko-Cantelli类’, *Journal of Theoretical Probability* **4**(3), 485–510. Dudley, R. M. (1987), ‘通用Donsker类和度量熵’, *Annals of Prob- ability* **15**(4), 1306–1326. Fisher, R. A. (1922), ‘关于理论统计学的数学基础’, *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character* **222**, 309–368. Floyd, S. (1989), 空间限制学习与Vapnik-Chervonenkis维度, *in*‘学习理论会议（COLT）’, 第349–364页。Floyd, S. & Warmuth, M. (1995), ‘样本压缩、可学习和Vapnik-Chervonenkis维度’, *Machine Learning* **21**(3), 269–304. Frank, M. & Wolfe, P. (1956), ‘二次规划算法’, *Naval Res. Logist. Quart.* **3**, 95–110. Freund, Y. & Schapire, R. (1995), 决策理论在线学习的推广及其在提升中的应用, *in*‘欧洲计算学习理论会议（EuroCOLT）’, Springer-Verlag, 第23–37页。Freund, Y. & Schapire, R. E. (1999), ‘使用感知器算法进行大间隔分类’, *Machine Learning* **37**(3), 277–296. Garcia, J. & Koelling, R. (1996), ‘回避学习中线索与结果的关系’, *Foundations of animal behavior: classic papers with commentaries* **4**, 374. Gentile, C. (2003), ‘p-范数算法的鲁棒性’, *Machine Learning* **53**(3), 265–299. Georghiades, A., Belhumeur, P. & Kriegman, D. (2001), ‘从少到多：在可变光照和姿态下的人脸识别的照明锥模型’, *IEEE Trans. Pattern Anal. Mach. Intelligence* **23**(6), 643–660. Gordon, G. (1999), 预测问题的遗憾界限, *in*‘学习理论会议（COLT）’。Gottlieb, L.-A., Kontorovich, L. & Krauthgamer, R. (2010), 度量数据的有效分类, *in*‘第23届学习理论会议’, 第433–440页。Guyon, I. & Elisseeff, A. (2003), ‘变量和特征选择简介’, *Journal of Machine Learning Research, Special Issue on Variable and Feature Selection* **3**, 1157–1182. Hadamard, J. (1902), ‘关于偏导数问题和它们的物理意义’, *Princeton University Bulletin* **13**, 49–52. Hastie, T., Tibshirani, R. & Friedman, J. (2001), *The Elements of Statistical Learning*, Springer. Haussler, D. (1992), ‘决策理论对PAC模型在神经网络和其他学习应用中的推广’, *Information and Computation* **100**(1), 78–150. Haussler, D. & Long, P. M. (1995), ‘Saunders引理的推广’, *Journal of Combinatorial Theory, Series A* **71**(2), 219–240. Hazan, E., Agarwal, A. & Kale, S. (2007), ‘在线凸优化的对数遗憾算法’, *Machine Learning* **69**(2–3), 169–192.

Hinton, G. E., Osindero, S. & Teh, Y.-W. (2006), ‘深度信念网的快速学习算法’, *Neural Computation* **18**(7), 1527–1554. Hiriart-Urruty, J.-B. & Lemaréchal, C. (1996), *Convex Analysis and Minimization Al- gorithms: Part 1: Fundamentals*, 第1卷, Springer. Hsu, C.-W., Chang, C.-C. & Lin, C.-J. (2003), ‘支持向量分类的实用指南’. Hyafil, L. & Rivest, R. L. (1976), ‘构造最优二叉决策树是NP-完全的’, *Information Processing Letters* **5**(1), 15–17. Joachims, T. (2005), 多变量性能测量的支持向量方法, *in* ‘国际机器学习会议（ICML）论文集’. Kakade, S., Sridharan, K. & Tewari, A. (2008), 关于线性预测的复杂性：风险界限，边界界限和正则化, *in* ‘NIPS’. Karp, R. M. (1972), *Reducibility among combinatorial problems*, Springer. Kearns, M. J., Schapire, R. E. & Sellie, L. M. (1994), ‘走向高效的不可知学习’, *Machine Learning* **17**, 115–141. Kearns, M. & Mansour, Y. (1996), 关于自顶向下决策树学习算法的增强能力, *in* ‘ACM计算理论研讨会（STOC）’. Kearns, M. & Ron, D. (1999), ‘算法稳定性和留一法交叉验证的sanity-check界限’, *Neural Computation* **11**(6), 1427–1453. Kearns, M. & Valiant, L. G. (1988), 学习布尔公式或有限自动机与分解一样困难, 技术报告TR-14-88, 哈佛大学Aiken计算实验室. Kearns, M. & Vazirani, U. (1994), *An Introduction to Computational Learning Theory*, MIT Press. Kleinberg, J. (2003), ‘聚类的不可能性定理’, *Advances in Neural Information Processing Systems* 第463–470页. Klivans, A. R. & Sherstov, A. A. (2006), 学习半空间交集的密码学难度, *in* ‘FOCS’. Koller, D. & Friedman, N. (2009), *Probabilistic Graphical Models: Principles and Tech- niques*, MIT Press. Koltchinskii, V. & Panchenko, D. (2000), Rademacher过程和函数学习风险界限, *in* ‘高维概率II’, Springer, 第443–457页. Kuhn, H. W. (1955), ‘分配问题的匈牙利方法’, *Naval research logistics quarterly* **2**(1-2), 83–97. Kutin, S. & Niyogi, P. (2002), 几乎处处算法稳定性和泛化误差, *in* ‘第18届不确定性人工智能会议论文集’, 第275–282页. Lafferty, J., McCallum, A. & Pereira, F. (2001), 条件随机字段：用于分割和标记序列数据的概率模型, *in* ‘国际机器学习会议’, 第282–289页. Langford, J. (2006), ‘分类的实用预测理论教程’, *Journal of machine learning research* **6**(1), 第273页. Langford, J. & Shawe-Taylor, J. (2003), PAC-Bayes & margins, *in* ‘NIPS’, 第423–430页. Le Cun, L. (2004), 大规模在线学习, *in* ‘神经信息处理系统16：2003年会议论文集’, 第16卷, MIT Press, 第217页.

Le, Q. V., Ranzato, M.-A., Monga, R., Devin, M., Corrado, G., Chen, K., Dean, J. & Ng, A. Y. (2012), 使用大规模无监督学习构建高级特征, *in*《国际机器学习会议（ICML）》. Lecun, Y. & Bengio, Y. (1995), *Convolutional Networks for Images, Speech and Time Series*, 麻省理工学院出版社, 第255–258页。Lee, H., Grosse, R., Ranganath, R. & Ng, A. (2009), 用于可扩展无监督学习层次表示的卷积深度信念网络, *in*《国际机器学习会议（ICML）》. Littlestone, N. (1988), '当存在大量无关属性时快速学习：一种新的线性阈值算法', *Machine Learning* **2**, 第285–318页。Littlestone, N. & Warmuth, M. (1986), 将数据压缩与可学习性联系起来。未发表的手稿。Littlestone, N. & Warmuth, M. K. (1994), '加权多数算法', *Infor- mation and Computation* **108**, 第212–261页。Livni, R., Shalev-Shwartz, S. & Shamir, O. (2013), '训练深度网络的证明有效算法', *arXiv preprint arXiv:1304.7045*。Livni, R. & Simon, P. (2013), 诚实压缩及其在压缩方案中的应用, *in*《学习理论会议（COLT）》. MacKay, D. J. (2003), *Information theory, inference and learning algorithms*, 剑桥大学出版社。Mallat, S. & Zhang, Z. (1993), '使用时间-频率字典的匹配追求', *IEEE Transactions on Signal Processing* **41**, 第3397–3415页。McAllester, D. A. (1998), 一些PAC-Bayesian定理, *in*《学习理论会议（COLT）》. McAllester, D. A. (1999), PAC-Bayesian模型平均, *in*《学习理论会议（COLT）》, 第164–170页。McAllester, D. A. (2003), 简化的PAC-Bayesian边缘界限, *in*《学习理论会议（COLT）》, 第203–215页。Minsky, M. & Papert, S. (1969), *Perceptrons: An Introduction to Computational Geometry*, 麻省理工学院出版社。Mukherjee, S., Niyogi, P., Poggio, T. & Rifkin, R. (2006), '学习理论：稳定性对于泛化是充分的，对于经验风险最小化的一致性是必要且充分的', *Advances in Computational Mathematics* **25**(1-3), 第161–193页。Murata, N. (1998), '在线学习的统计研究', *Online Learning and Neural Networks. Cambridge University Press, Cambridge, UK*。Murphy, K. P. (2012), *Machine learning: a probabilistic perspective*, 麻省理工学院出版社。Natarajan, B. (1995), '线性系统的稀疏近似解', *SIAM J. Com- puting* **25**(2), 第227–234页。Natarajan, B. K. (1989), '关于学习和函数', *Mach. Learn.* **4**, 第67–97页。Nemirovski, A., Juditsky, A., Lan, G. & Shapiro, A. (2009), '稳健随机逼近方法在随机规划中的应用', *SIAM Journal on Optimization* **19**(4), 第1574–1609页。Nemirovski, A. & Yudin, D. (1978), *Problem complexity and method efficiency in opti- mization*, 莫斯科科学出版社。Nesterov, Y. (2005), 凸问题的原对偶子梯度方法, 技术报告, 运筹学与经济计量学中心（CORE）, 卢万天主教大学（UCL）。

Nesterov, Y. & Nesterov, I. (2004), *Introductory lectures on convex optimization: A basic course*, 第87卷, Springer荷兰。Novikoff, A. B. J. (1962), 关于感知器收敛证明, *in*《自动机数学理论研讨会论文集》, 第XII卷, 第615–622页。Parberry, I. (1994), *Circuit complexity and neural networks*, 麻省理工学院出版社。Pearson, K. (1901), '空间中点系统的最佳拟合线和面',
The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science **2**(11), 559–572。Phillips, D. L. (1962), '一种求解某些第一类积分方程的数值方法', *Journal of the ACM* **9**(1), 84–97。Pisier, G. (1980-1981), '关于B. maurey未发表结果的评论'。Pitt, L. & Valiant, L. (1988), '从示例学习中的计算限制',
Journal of the Association for Computing Machinery **35**(4), 965–984。Poon, H. & Domingos, P. (2011), 求和-积网络：一种新的深度架构, *in*《不确定性人工智能（UAI）会议》。Quinlan, J. R. (1986), '决策树的归纳', *Machine Learning* **1**, 81–106。Quinlan, J. R. (1993), *C4.5: Programs for Machine Learning*, Morgan Kaufmann。Rabiner, L. & Juang, B. (1986), '隐马尔可夫模型简介', *IEEE ASSP Magazine* **3**(1), 4–16。Rakhlin, A., Shamir, O. & Sridharan, K. (2012), 使梯度下降在强凸随机优化中达到最优, *in*《国际机器学习会议（ICML）》。Rakhlin, A., Sridharan, K. & Tewari, A. (2010), 在线学习：随机平均值, 组合参数和可学习性, *in*《NIPS》。Rakhlin, S., Mukherjee, S. & Poggio, T. (2005), '学习理论中的稳定性结果', *Analysis and Applications* **3**(4), 397–419。Ranzato, M., Huang, F., Boureau, Y. & Lecun, Y. (2007), 用于物体识别的无监督学习不变特征层次结构及其应用, *in*《2007年计算机视觉和模式识别, CVPR'07。IEEE会议》, IEEE, 第1–8页。Rissanen, J. (1978), '通过最短数据描述建模', *Automatica* **14**, 465–471。Rissanen, J. (1983), '整数的一个通用先验和最小描述长度估计',
The Annals of Statistics **11**(2), 416–431。Robbins, H. & Monro, S. (1951), '一种随机逼近方法', *The Annals of Mathematical Statistics*, 第400–407页。Rogers, W. & Wagner, T. (1978), '局部判别规则的有限样本无分布性能界限',
The Annals of Statistics **6**(3), 506–514。Rokach, L. (2007),
Data mining with decision trees: theory and applications, 第69卷, 世界科学。Rosenblatt, F. (1958), '感知器：大脑中信息存储和组织的一种概率模型',
Psychological Review **65**, 386–407。（在*Neurocomputing* (麻省理工学院出版社, 1988)再版）。Rumelhart, D. E., Hinton, G. E. & Williams, R. J. (1986), 通过误差传播学习内部表示, *in* D. E. Rumelhart & J. L. McClelland, 编, 《并行分布式处理 – 认知微观结构探索》, MIT Press, 第8章, 第318–362页。

Sankaran, J. K. (1993), ‘通过约束松弛解决线性规划不可行性的注释’, *Operations Research Letters* **13**(1), 19–20. Sauer, N. (1972), ‘关于集合族密度的讨论’, *Journal of Combinatorial Theory Series A* **13**, 145–147. Schapire, R. (1990), ‘弱学习能力的强度’, *Machine Learning* **5**(2), 197– 227. Schapire, R. E. & Freund, Y. (2012), *Boosting: Foundations and Algorithms*, MIT press. Schölkopf, B., Herbrich, R. & Smola, A. (2001), 广义重表示定理, *in* ‘计算学习理论’, pp. 416–426. Schölkopf, B., Herbrich, R., Smola, A. & Williamson, R. (2000), 广义重表示定理, *in* ‘NeuroCOLT’. Schölkopf, B. & Smola, A. J. (2002), *Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond*, MIT Press. Schölkopf, B., Smola, A. & Müller, K.-R. (1998), ‘非线性成分分析作为核特征值问题’, *Neural computation* **10**(5), 1299–1319. Seeger, M. (2003), ‘高斯过程分类的Pac-bayesian泛化误差界限’, *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), 在线学习：理论、算法与应用, 博士论文, 希伯来大学。 Shalev-Shwartz, S. (2011), ‘在线学习和在线凸优化’, *Founda- tions and Trends® in Machine Learning* **4**(2), 107–194. Shalev-Shwartz, S., Shamir, O., Srebro, N. & Sridharan, K. (2010), ‘可学习性、稳定性和一致收敛’, *The Journal of Machine Learning Research* **9999**, 2635–2670. Shalev-Shwartz, S., Shamir, O. & Sridharan, K. (2010), 使用零一损失学习基于核的半空间, *in* ‘学习理论会议（COLT）’. Shalev-Shwartz, S., Shamir, O., Sridharan, K. & Srebro, N. (2009), 随机凸优化, *in* ‘学习理论会议（COLT）’. Shalev-Shwartz, S. & Singer, Y. (2008), 关于弱学习能力和线性可分性的等价性：新的松弛和有效的提升算法, *in* ‘第十九次计算学习理论年会论文集’. Shalev-Shwartz, S., Singer, Y. & Srebro, N. (2007), Pegasos：SVM的原始估计子梯度求解器, *in* ‘国际机器学习会议’, pp. 807–814. Shalev-Shwartz, S. & Srebro, N. (2008), SVM优化：训练集大小对逆依赖性, *in* ‘国际机器学习会议’, pp. 928–935. Shalev-Shwartz, S., Zhang, T. & Srebro, N. (2010), ‘在稀疏约束优化问题中用稀疏性换取精度’, *Siam Journal on Optimization* **20**, 2807–2832. Shamir, O. & Zhang, T. (2013), 非光滑优化的随机梯度下降：收敛结果和最优平均方案, *in* ‘国际机器学习会议（ICML）’. Shapiro, A., Dentcheva, D. & Ruszczyński, A. (2009), *Lectures on stochastic program- ming: modeling and theory*, 第9卷, 工业与应用数学学会。

Shelah, S. (1972), ‘组合问题；无穷语言中模型和理论的稳定性与顺序’, *Pac. J. Math* **4**, 247–261. Sipser, M. (2006), *Introduction to the Theory of Computation*, Thomson Course Technology. Slud, E. V. (1977), ‘二项分布的分布不等式’, *The Annals of Probability* **5**(3), 404–412. Steinwart, I. & Christmann, A. (2008), *Support vector machines*, Springerverlag New York. Stone, C. (1977), ‘一致的非参数回归’, *The annals of statistics* **5**(4), 595–620. Taskar, B., Guestrin, C. & Koller, D. (2003), 最大边缘马尔可夫网络, *in* ‘NIPS’. Tibshirani, R. (1996), ‘通过lasso进行回归收缩和选择’, *J. Royal. Statist. Soc B.* **58**(1), 267–288. Tikhonov, A. N. (1943), ‘关于逆问题的稳定性’, *Dolk. Akad. Nauk SSSR* **39**(5), 195–198. Tishby, N., Pereira, F. & Bialek, W. (1999), 信息瓶颈方法, *in* ‘第37届Allerton通信、控制和计算会议’. Tsochantaridis, I., Hofmann, T., Joachims, T. & Altun, Y. (2004), 支持向量机学习用于相互依赖和结构化输出空间, *in* ‘第二十一届国际机器学习会议论文集’. Valiant, L. G. (1984), ‘可学习理论’, *Communications of the ACM* **27**(11), 1134–1142. Vapnik, V. (1992), 学习理论的风险最小化原理, J. E. Moody, S. J. Hanson & R. P. Lippmann, 编, 《神经网络信息处理系统4》, Morgan Kaufmann, 第831–838页。 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), ‘关于事件相对频率到其概率的均匀收敛’, *Theory of Probability and its applications* **XVI**(2), 264–280. Vapnik, V. N. & Chervonenkis, A. Y. (1974), *Theory of pattern recognition*, Nauka, 莫斯科。（俄语）。Von Luxburg, U. (2007), ‘关于谱聚类的教程’, *Statistics and computing* **17**(4), 395–416. von Neumann, J. (1928), ‘关于社交游戏理论（关于客厅游戏理论）’, *Math. Ann.* **100**, 295—320. Von Neumann, J. (1953), ‘一个与最优分配问题等价的双人零和游戏’, *Contributions to the Theory of Games* **2**, 5–12. Vovk, V. G. (1990), 聚合策略, *in* ‘学习理论会议（COLT）’, 第371–383页。Warmuth, M., Glocer, K. & Vishwanathan, S. (2008), 熵正则化的lpboost, *in* ‘算法学习理论（ALT）’. Warmuth, M., Liao, J. & Ratsch, G. (2006), 最大化边缘的完全纠正提升算法, *in* ‘第二十三届国际机器学习会议论文集’.

Weston, J., Chapelle, O., Vapnik, V., Elisseeff, A. & Schölkopf, B. (2002), 内核依赖估计, *in* 《神经网络信息处理系统进展》, 第 873– 880 页。Weston, J. & Watkins, C. (1999), 支持向量机在多类模式识别中的应用, *in* 《第七届欧洲人工神经网络研讨会论文集》。Wolpert, D. H. & Macready, W. G. (1997), 《优化中的免费午餐定理》, *Evolutionary Computation, IEEE Transactions on* **1**(1), 第 67–82 页。Zhang, T. (2004), 使用随机梯度下降算法解决大规模线性预测问题, *in* 《第二十一届国际机器学习会议论文集》。Zhao, P. & Yu, B. (2006), 《关于Lasso模型选择一致性的研究》, *Journal of Machine Learning Research* **7**, 第 2541–2567 页。Zinkevich, M. (2003), 在线凸规划和广义无穷小梯度上升, *in* 《机器学习国际会议》。

Index

3-term DNF, 107 F_1-得分，24
4 ℓ_1范数，183，332，363，3
86
准确度，38，43 激活函数，2
69 AdaBoost，130，**134**，36
2 全对，228，404 近似误差
，61，64 自编码器，368

反向传播，278向后消除，363词袋
，209基本假设，**137**贝叶斯最优，
46，52，260贝叶斯法则，354贝叶
斯推理，353Bennet不等式，426Ber
nstein不等式，426偏差，37，61，6
4偏差-复杂度权衡，65布尔合取，5
1，79，106提升，130提升置信度，
142有界性，165

C4.5, 254 CART, 254 连锁，389 切比
雪夫不等式，423 切比雪夫界，423 类
敏感特征映射，230 分类器，34 聚类
，307 光谱，315 压缩感知，330 压缩
界，410 压缩方案，411 计算复杂度，
100 置信度，38，43 一致性，92 一致
，289 收敛引理，381 凸，156 函数，
157

集合，156 强凸，174，195 凸-Lipschitz
有界学习，166 凸-光滑有界学习，166 覆
盖数，388 维度灾难，263 决策树桩，132
，133 决策树，250 系谱图，309，310 字
典学习，368 微分集，188 维度约简，323
离散化技巧，57 判别性，342 无分布，34
2 区域，33 样例域，48 双随机矩阵，242
对偶性，211 强对偶性，211 弱对偶性，2
11 Dudley类，81 有效可计算，100 EM，3
48 实验误差，35 实验风险，35，**48** 实验
风险最小化，*see* ERM 熵，345 相对熵，3
45 雅可比图，157 ERM，35 错误分解，64
，168 估计误差，61，64 期望最大化，*see*
EM 面部识别，*see* Viola-Jones 可行，100
特征，33 特征学习，368 特征归一化，36
5 特征选择，357，358 特征空间，215 特
征变换，367 滤波器，359

前向贪婪选择，360频率主义者，353增益，253 GD，*see*梯度下降泛化误差，35生成模型，342基尼指数，254 Glivenko-Cantelli，58梯度，158梯度下降，185格拉姆矩阵，219增长函数，73半空间，118同质，118，205不可分，119可分，118二分，289隐藏层，270希尔伯特空间，217Hoeffding不等式，56，425保留，146假设，34假设类，36独立同分布，38ID3，252不恰当，*see*表示独立归纳偏差，*see*偏差信息瓶颈，317信息增益，254实例，33实例空间，33积分图像，143Johnson-Lindenstrauss引理，329k-means，311，313软k-means，352k-median，312k-medoids，312Kendall tau，239核PCA，326核，215高斯核，220核技巧，217多项式核，220RBF核，220标签，33Lasso，365，386泛化界限，386潜在变量，348LDA，347Ldim，290，291学习曲线，153最小二乘法，124似然比，348线性判别分析，*see* LDA线性预测器，117

均质，118 线性规划，119 线性回归，122 链接，310 Lipschitzness，160，176，191 子梯度，190 Littlestone 维度，*see* Ldim 局部最小值，158 逻辑回归，126 损失，35 损失函数，**48** 0-1 损失，48，167 绝对值损失，124，128，166 凸损失，163 广义铰链损失，233 铰链损失，167 Lipschitz 损失，166 log-loss，345 逻辑损失，127 斜坡损失，209 平滑损失，166 平方损失，48 代理损失，167，302 边距，203 马尔可夫不等式，422 Massart 引理，380 最大链接，310 最大后验，355 最大似然，343 McDiarmid 不等式，378 MDL，89，90，251 测度集中，55，422 最小描述长度，*see* MDL 错误界限，288 高斯混合，348 模型选择，144，147 多类，47，227，402 成本敏感，232 线性预测器，230，405 多向量，231，406 Perceptron，248 抽象，227，405 SGD，235 SVM，234 多变量性能度量，243 Naive Bayes，347 Natarajan 维度，402 NDCG，239 最近邻，258 k-NN，258 神经网络，268 前馈网络，269 层叠网络，269 SGD，277 无免费午餐，61 非均匀学习，84

标准化累积折扣增益，*see* NDCG

奥卡姆剃刀，91 OMP，360 一对全，227 一对一，*see* 一对全 一对全，404 在线凸优化，300 在线梯度下降，300 在线学习，287 优化误差，168 神谕不等式，179 正交匹配追踪，*see* OMP 过拟合，35，65，152 PAC，43 无知 PAC，45，46 无知 PAC 对于一般损失，**49** PAC-Bayes，415 参数密度估计，342 PCA，324 皮尔逊相关系数，359 感知机，120 核感知机，225 多类，248 在线，301 排列矩阵，242 多项式回归，125 精度，244 预测器，34 前缀自由语言，89 主成分分析，*see* PCA 先验知识，63 大概正确，*see* PAC 投影，193 投影引理，193 合法的，49 剪枝，254 Rademacher 复杂度，375 随机森林，255 随机投影，329 排序，238 二部图，243 可实现性，37 召回，244 回归，47，122，172 正则化，171 Tikhonov，172，174 正则化损失最小化，*see* RLM 表示独立，49，107 代表样本，54，375 代表定理，218 岭回归，172 核岭回归，225 RIP，331 风险，35，45，**48** RLM，171，199

样本复杂度，44 萨乌尔引理，73 自限性，162 敏感性，244 SGD，190 粉碎，69，403 单链接，310奇异值分解，*see* SVD Slud的不等式，428 光滑性，162，177，198 SOA，292 诱导稀疏性的范数，363 特异性，244 谱聚类，315 SRM，85，145 稳定性，173 随机梯度下降，*see* SGD 强学习，132 结构风险最小化，*see* SRM 结构化输出预测，236 子梯度，188 支持向量机，*see* SVM SVD，431 SVM，202，383 对偶性，211 泛化界限，208，383 硬SVM，203，204 同质，205 核技巧，217 软SVM，206 支持向量，210 目标集，47 项频率，231 TF-IDF，231 训练误差，35 训练集，33 真实误差，35，45 欠拟合，65，152 一致收敛，54，**55** 并集界限，39 无监督学习，308 验证，144，146 交叉验证，149 训练-验证-测试划分，150 Vapnik-Chervonenkis 维度，*see* VC 维度 VC 维度，67，70 版本空间，289 Viola-Jones，139 弱学习，130，**131** 加权多数，295