

模式识别与机器学习

马春鹏

October 26, 2014

目录

1 绪论	9
1.1 例子：多项式曲线拟合	10
1.2 概率论	16
1.2.1 概率密度	20
1.2.2 期望和协方差	21
1.2.3 贝叶斯概率	22
1.2.4 高斯分布	24
1.2.5 重新考察曲线拟合问题	26
1.2.6 贝叶斯曲线拟合	28
1.3 模型选择	29
1.4 维度灾难	30
1.5 决策论	33
1.5.1 最小化错误分类率	34
1.5.2 最小化期望损失	35
1.5.3 拒绝选项	35
1.5.4 推断和决策	36
1.5.5 回归问题的损失函数	38
1.6 信息论	39
1.6.1 相对熵和互信息	44
1.7 练习	46
2 概率分布	52
2.1 二元变量	52
2.1.1 Beta分布	54
2.2 多项式变量	56
2.2.1 狄利克雷分布	58
2.3 高斯分布	59
2.3.1 条件高斯分布	63
2.3.2 边缘高斯分布	65
2.3.3 高斯变量的贝叶斯定理	67
2.3.4 高斯分布的最大似然估计	69
2.3.5 顺序估计	69
2.3.6 高斯分布的贝叶斯推断	71
2.3.7 学生t分布	75
2.3.8 周期变量	77
2.3.9 混合高斯模型	81
2.4 指数族分布	83
2.4.1 最大似然与充分统计量	86
2.4.2 共轭先验	87
2.4.3 无信息先验	87
2.5 非参数化方法	89
2.5.1 核密度估计	90
2.5.2 近邻方法	92
2.6 练习	94
3 回归的线性模型	101
3.1 线性基函数模型	101
3.1.1 最大似然与最小平方	102
3.1.2 最小平方的几何描述	105
3.1.3 顺序学习	105

3.1.4	正则化最小平方	105
3.1.5	多个输出	106
3.2	偏置-方差分解	108
3.3	贝叶斯线性回归	111
3.3.1	参数分布	111
3.3.2	预测分布	113
3.3.3	等价核	116
3.4	贝叶斯模型比较	118
3.5	证据近似	121
3.5.1	计算证据函数	121
3.5.2	最大化证据函数	123
3.5.3	参数的有效数量	124
3.6	固定基函数的局限性	126
3.7	练习	126
4	分类的线性模型	130
4.1	判别函数	131
4.1.1	二分类	131
4.1.2	多分类	132
4.1.3	用于分类的最小平方方法	133
4.1.4	Fisher线性判别函数	135
4.1.5	与最小平方的关系	137
4.1.6	多分类的Fisher判别函数	138
4.1.7	感知器算法	139
4.2	概率生成式模型	141
4.2.1	连续输入	143
4.2.2	最大似然解	144
4.2.3	离散特征	146
4.2.4	指数族分布	146
4.3	概率判别式模型	147
4.3.1	固定基函数	147
4.3.2	logistic回归	148
4.3.3	迭代重加权最小平方	149
4.3.4	多类logistic回归	150
4.3.5	probit回归	151
4.3.6	标准链接函数	152
4.4	拉普拉斯近似	154
4.4.1	模型比较和BIC	155
4.5	贝叶斯logistic回归	156
4.5.1	拉普拉斯近似	156
4.5.2	预测分布	157
4.6	练习	158
5	神经网络	161
5.1	前馈神经网络	161
5.1.1	权空间对称性	165
5.2	网络训练	165
5.2.1	参数最优化	168
5.2.2	局部二次近似	169
5.2.3	使用梯度信息	170
5.2.4	梯度下降最优化	170
5.3	误差反向传播	171

5.3.1	误差函数导数的计算	172
5.3.2	一个简单的例子	174
5.3.3	反向传播的效率	175
5.3.4	Jacobian矩阵	175
5.4	Hessian矩阵	177
5.4.1	对角近似	177
5.4.2	外积近似	178
5.4.3	Hessian矩阵的逆矩阵	178
5.4.4	有限差	179
5.4.5	Hessian矩阵的精确计算	179
5.4.6	Hessian矩阵的快速乘法	180
5.5	神经网络的正则化	182
5.5.1	相容的高斯先验	183
5.5.2	早停止	185
5.5.3	不变性	186
5.5.4	切线传播	187
5.5.5	用变换后的数据训练	189
5.5.6	卷积神经网络	190
5.5.7	软权值共享	191
5.6	混合密度网络	193
5.7	贝叶斯神经网络	197
5.7.1	后验参数分布	198
5.7.2	超参数最优化	199
5.7.3	用于分类的贝叶斯神经网络	200
5.8	练习	202
6	核方法	206
6.1	对偶表示	206
6.2	构造核	207
6.3	径向基函数网络	211
6.3.1	Nadaraya-Watson模型	212
6.4	高斯过程	214
6.4.1	重新考虑线性回归问题	214
6.4.2	用于回归的高斯过程	216
6.4.3	学习超参数	219
6.4.4	自动相关性确定	220
6.4.5	用于分类的高斯过程	221
6.4.6	拉普拉斯近似	222
6.4.7	与神经网络的联系	225
6.5	练习	225
7	稀疏核机	228
7.1	最大边缘分类器	228
7.1.1	重叠类分布	231
7.1.2	与logistic回归的关系	235
7.1.3	多类SVM	236
7.1.4	回归问题的SVM	237
7.1.5	计算学习理论	240
7.2	相关向量机	241
7.2.1	用于回归的RVM	241
7.2.2	稀疏性分析	244
7.2.3	RVM用于分类	247

7.3	练习	249
8	图模型	251
8.1	贝叶斯网络	251
8.1.1	例子：多项式回归	253
8.1.2	生成式模型	255
8.1.3	离散变量	255
8.1.4	线性高斯模型	257
8.2	条件独立	259
8.2.1	图的三个例子	260
8.2.2	d-划分	264
8.3	马尔科夫随机场	266
8.3.1	条件独立性质	267
8.3.2	分解性质	268
8.3.3	例子：图像去噪	269
8.3.4	与有向图的关系	271
8.4	图模型中的推断	274
8.4.1	链推断	274
8.4.2	树	277
8.4.3	因子图	277
8.4.4	加和-乘积算法	279
8.4.5	最大加和算法	285
8.4.6	一般图的精确推断	289
8.4.7	循环置信传播	289
8.4.8	学习图结构	290
8.5	练习	290
9	混合模型和EM	293
9.1	K 均值聚类	293
9.1.1	图像分割与压缩	296
9.2	混合高斯	297
9.2.1	最大似然	298
9.2.2	用于高斯混合模型的EM	300
9.3	EM的另一种观点	303
9.3.1	重新考察高斯混合模型	304
9.3.2	与 K 均值的关系	305
9.3.3	伯努利分布的混合	306
9.3.4	贝叶斯线性回归的EM算法	309
9.4	一般形式的EM算法	310
9.5	练习	313
10	近似推断	316
10.1	变分推断	316
10.1.1	分解概率分布	317
10.1.2	分解近似的性质	319
10.1.3	例子：一元高斯分布	321
10.1.4	模型比较	324
10.2	例子：高斯的变分混合	324
10.2.1	变分分布	325
10.2.2	变分下界	329
10.2.3	预测概率密度	330
10.2.4	确定分量的数量	331

10.2.5 诱导分解	332
10.3 变分线性回归	332
10.3.1 变分分布	333
10.3.2 预测分布	334
10.3.3 下界	335
10.4 指数族分布	335
10.4.1 变分信息传递	337
10.5 局部变分方法	337
10.6 变分logistic回归	341
10.6.1 变分后验概率分布	341
10.6.2 最优化变分参数	343
10.6.3 超参数的推断	344
10.7 期望传播	346
10.7.1 例子：聚类问题	350
10.7.2 图的期望传播	352
10.8 练习	355
 11 采样方法	358
11.1 基本采样算法	359
11.1.1 标准概率分布	359
11.1.2 拒绝采样	361
11.1.3 可调节的拒绝采样	362
11.1.4 重要采样	363
11.1.5 采样-重要性-重采样	365
11.1.6 采样与EM算法	366
11.2 马尔科夫链蒙特卡罗	367
11.2.1 马尔科夫链	368
11.2.2 Metropolis-Hastings算法	370
11.3 吉布斯采样	370
11.4 切片采样	373
11.5 混合蒙特卡罗算法	374
11.5.1 动态系统	374
11.5.2 混合蒙特卡罗方法	376
11.6 估计划分函数	378
11.7 练习	379
 12 连续潜在变量	381
12.1 主成分分析	381
12.1.1 最大方差形式	382
12.1.2 最小误差形式	383
12.1.3 PCA的应用	385
12.1.4 高维数据的PCA	388
12.2 概率PCA	388
12.2.1 最大似然PCA	391
12.2.2 用于PCA的EM算法	393
12.2.3 贝叶斯PCA	395
12.2.4 因子分析	397
12.3 核PCA	399
12.4 非线性隐含变量模型	402
12.4.1 独立成分分析	402
12.4.2 自关联网络	403
12.4.3 对非线性流形建模	405

12.5 练习	407
13 顺序数据	410
13.1 马尔科夫模型	410
13.2 隐马尔科夫模型	413
13.2.1 用于HMM的最大似然法	417
13.2.2 前向后向算法	418
13.2.3 用于HMM的加和-乘积算法	423
13.2.4 缩放因子	425
13.2.5 维特比算法	426
13.2.6 隐马尔科夫模型的扩展	427
13.3 线性动态系统	430
13.3.1 LDS中的推断	432
13.3.2 LDS中的学习	434
13.3.3 LDS的推广	436
13.3.4 粒子滤波	437
13.4 练习	438
14 组合模型	441
14.1 贝叶斯模型平均	441
14.2 委员会	442
14.3 提升方法	443
14.3.1 最小化指数误差	444
14.3.2 提升方法的误差函数	446
14.4 基于树的模型	447
14.5 条件混合模型	449
14.5.1 线性回归模型的混合	449
14.6 logistic模型的混合	452
14.6.1 专家混合	453
14.7 练习	454
A 附录A. 数据集	456
A.1 手写数字	456
A.2 石油流	456
A.3 老忠实间歇喷泉	458
A.4 人工生成数据	459
B 附录B. 概率分布	460
B.1 伯努利分布	460
B.2 Beta分布	460
B.3 二项分布	461
B.4 狄利克雷分布	461
B.5 Gamma分布	462
B.6 高斯分布	462
B.7 高斯-Gamma分布	463
B.8 高斯-Wishart分布	464
B.9 多项式分布	464
B.10 正态分布	464
B.11 学生t分布	465
B.12 均匀分布	465
B.13 Von Mises分布	465
B.14 Wishart分布	466

C 附录C. 矩阵的性质	467
C.1 矩阵的基本性质	467
C.2 迹和行列式	467
C.3 矩阵的导数	468
C.4 特征向量方程	469
D 附录D. 变分法	472
E 附录E. 拉格朗日乘数法	474

1 绪论

寻找数据中模式的问题是一个基本的问题，有着很长的很成功的历史。例如，16世纪Tycho Brahe的大量的观测使得Johannes Kepler发现行星运行的经验性规律，这反过来给经典力学的发展提供了跳板。类似地，原子光谱的规律的发现在20世纪初期对于量子力学的发展和证明有着重要的作用。模式识别领域关注的是利用计算机算法自动发现数据中的规律，以及使用这些规律采取将数据分类等行动。

考虑手写数字识别的例子，如图1.1所示。每个数字对应一个 28×28 像素的图像，因此可以表示为一个由784个实数组成的向量 x 。目标是建立一个机器，能够以这样的向量 x 作为输入，以数字0到9为输出。这不是一个简单的问题，因为手写体变化多端。这个问题可以使用人工编写的规则解决，或者依据笔画的形状启发式地区分数字，但是实际中这样的方法导致了规则数量的激增，以及不符合规则的例外等等，并且始终给出较差的结果。

使用机器学习的方法可以得到好得多的结果。这个方法中，一个由 N 个数字 $\{x_1, \dots, x_N\}$ 组成的大集合被叫做训练集 (training set)，用来调节模型的参数。训练集中数字的类别实现已知，通常是被独立考察、人工标注的。我们可以使用目标向量 (target vector) t 来表示数字的类别，它代表对应数字的标签。使用向量来表示类别的合适的技术将在后面讨论。注意对于每个数字图像 x 只有一个目标向量 t 。

运行机器学习算法的结果可以被表示为一个函数 $y(x)$ ，它以一个新的数字的图像 x 为输入，产生向量 y ，与目标向量的形式相同。函数 $y(x)$ 的精确形式在训练 (training) 阶段被确定，这个阶段也被称为学习 (learning) 阶段，以训练数据为基础。一旦模型被训练出来，它就能确定新的数字的图像集合中图像的标签。这些新的数字的图像集合组成了测试集 (test set)。正确分类与训练集不同的新样本的能力叫做泛化 (generalization)。在实际应用中，输入向量的变化性是相当大的，以至于训练数据只所有可能的输入向量中相当小得一部分，所以泛化是模式识别的一个中心问题。

对于大部分实际应用，原始输入向量通常被预处理 (pre-processed)，变换到新的变量空间。人们期望在新的变量空间中模式识别问题可以更容易地被解决。例如，在数字识别的问题中，数字的图像通常被转化缩放，使得每个数字能够被包含到一个固定大小的盒子中。这极大地减少了每个数字类别的变化性，因为现在所有数字的位置和大小现在相同，这使得后续的区分不同类别的模式识别算法变得更加容易。这个预处理阶段有时被叫做特征抽取 (feature extraction)。注意新的测试集必须使用与训练集相同的方法进行预处理。

为了加快计算速度，也可能进行预处理。例如，如果目标是高清视频中得实时人脸检测，计算机每秒钟必须处理大量的像素。将这些像素直接传递给一个复杂的模式识别算法在计算上是不可行的。相反，目标是找到可以快速计算的有用的特征，这些特征还能够保存有用的信息使得人脸和非人脸可以被区分开。这些特征之后被用作模式识别算法的输入。例如，一个矩形小区域内图像灰度的平均值可以被快速计算 (Viola and Jones, 2014)，并且一组这样的特征被证明在快速人脸检测中很有效。由于这样的特征的数量小于像素的数量，因此这种预处理代表了一种形式的维数降低。必须注意，由于在预处理阶段信息通常被丢弃，因此如果信息对于问题的解决很重要的话，系统整体的精度会下降。

训练数据的样本包含输入向量以及对应的目标向量的应用叫做有监督学习 (supervised learning) 问题。数字识别就是这个问题的一个例子，它的目标是给每个输入向量分配到有限数

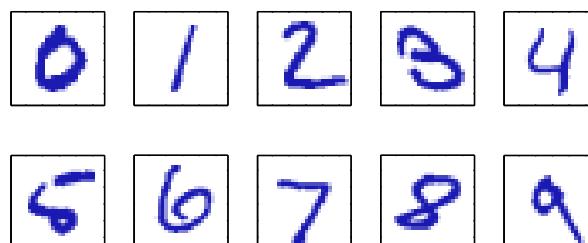


图 1.1: 来自美国邮政编码的手写数字的例子

量离散标签中的一个，被称为分类（classification）问题。如果要求的输出由一个或者多个连续变量组成，那么这个任务被称为回归（regression）。回归问题的一个例子是化学药品制造过程中产量的预测。在这个问题中，输入由反应物、温度、压力组成。

在其他的模式识别问题中，训练数据由一组输入向量 x 组成，没有任何对应的目标值。在这样的无监督学习（unsupervised learning）问题中，目标可能是发现数据中相似样本的分组，这被称为聚类（clustering），或者决定输入空间中数据的分布，这被称为密度估计（density estimation），或者把数据从高维空间投影到二维或者三维空间，为了数据可视化（visualization）。

最后，反馈学习（reinforcement learning）（Sutton and Barto, 1998）技术关注的问题是在给定的条件下，找到合适的动作，使得奖励达到最大值。这里，学习问题没有给定最优输出的用例。这些用例必须在一系列的实验和错误中被发现。这与有监督学习相反。通常，有一个状态和动作的序列，其中学习算法与环境交互。在许多情况下，当前动作不仅影响直接的奖励，也对所有后续时刻的奖励有影响。例如，通过使用合适的反馈学习技术，一个神经网络可以学会backgammon游戏的玩法，并且玩得很好（Tesauro, 1994）。这里神经网络必须学习把一大组位置信息、骰子投掷的结果作为输入，产生一个移动的方式作为输出。通过让神经网络自己和自己玩一百万局，这个目的就可以达到。一个主要的挑战是backgammon游戏会涉及到相当多次的移动，但是只有在游戏结束的时候才能给出奖励（以胜利的形式）。奖励必须被合理地分配给所有引起胜利的移动步骤。这些移动中，有些移动很好，其他的移动不是那么好。这是信用分配（credit assignment）问题的一个例子。反馈学习的一个通用的特征是探索（exploration）和利用（exploitation）的折中。“探索”是指系统尝试新类型的动作，“利用”是指系统使用已知能产生较高奖励的动作。过分地集中于探索或者利用都会产生较差的结果。反馈学习继续是机器学习研究中得一个活跃的领域。然而，详细讨论反馈学习不在本书的范围内。

虽然这些任务中每一个都需要自己的工具和技术，但是在这些任务背后的许多关键思想都是相通的。本章的主要目标是以一种相对非正式的形式介绍最重要的概念，并且使用简单的例子来说明。稍后在本书中，我们将看到同样的思想以更加复杂的模型的形式重新出现，这些模型能够应用于真实世界中模式识别的应用中。本章也将介绍将自始至终在本书中使用的三个重要工具：概率论、决策论、信息论。虽然这些东西听起来让人感觉害怕，但是实际上它们非常直观。并且，在实际应用中，如果想让机器学习技术发挥最大作用的话，清楚地理解它们是必须的。

1.1 例子：多项式曲线拟合

我们以一个简单的回归问题开始。本章中，我们将以这个问题为例，说明许多关键的概念。假设我们观察到一个实值输入变量 x ，我们想使用这个观察来预测实值目标变量 t 的值。对于这个目的，一个很好的方法是考虑一个使用已知的产生方式人工制造出的例子，因为这样我们就知道生成数据的精确过程，从而能够和我们学到得模型进行比较。这个例子的数据由函数 $\sin(2\pi x)$ 产生，目标变量带有随机的噪声。详细的描述见附录A。

现在假设给定一个训练集。这个训练集由 x 的 N 次观测组成，写作 $\mathbf{x} \equiv (x_1, \dots, x_N)^T$ ，伴随这对应的 t 的观测值，记作 $\mathbf{t} \equiv (t_1, \dots, t_N)^T$ 。图1.2展示了由 $N = 10$ 个数据点组成的图像。图1.2中的输入数据集合 \mathbf{x} 通过选择 $x_n (n = 1, \dots, N)$ 的值来生成。这些 x_n 均匀分布在区间 $[0, 1]$ ，目标数据集 \mathbf{t} 的获得方式是：首先计算函数 $\sin(2\pi x)$ 的对应的值，然后给每个点增加一个小的符合高斯分布的随机噪声（高斯分布将在1.2.4节讨论），从而得到对应的 t_n 的值。通过使用这种方式产生数据，我们利用了许多真实数据集合的一个性质，即它们拥有一个内在的规律，这个规律是我们想要学习的，但是独自的观察被随机噪声干扰。这种噪声可能由一个本质上随机的过程产生，例如放射性衰变。但是更典型的情况是由于存在没有被观察到的具有变化性的噪声源。

我们的目标是利用这个训练集预测对于输入变量的新值 \hat{x} 的目标变量的值 \hat{t} 。正如我们将要看到的那样，这涉及到隐式地发现内在的函数 $\sin(2\pi x)$ 。这本质上是一个困难的问题，因为我们不得不从有限的数据中生成。并且观察到得数据被噪声干扰，因此对于一个给定的 \hat{x} ，合适的 \hat{t} 值具有不确定性。概率论（在1.2节讨论）提供了一个框架，用来以精确的数学的形式描述这种不确定性。决策论（在1.5节讨论）让我们能够根据合适的标准，利用这种概率的表示，进行最优的预测。

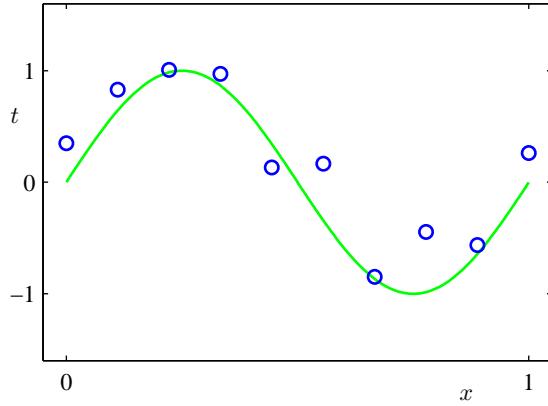


图 1.2: 由 $N = 10$ 个数据点组成的训练集的图像, 用蓝色圆圈标记。每个数据点由输入变量 x 的观测以及对应的目标变量 t 组成。绿色曲线给出了用来生成数据的 $\sin(2\pi x)$ 函数。我们的目标是对于某些新的 x 值, 预测 t 的值, 而无需知道绿色曲线。

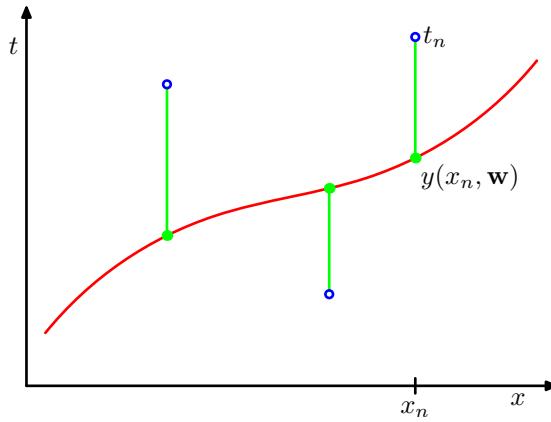


图 1.3: 误差函数 (1.2) 对应于每个数据点与函数 $y(x, \mathbf{w})$ 之间位移 (绿色垂直线) 的平方和 (的一半)。

但是现在, 我们要用一种相当非正式的、相当简单的方式来进行曲线拟合。特别地, 我们将使用下面形式的多项式函数来拟合数据:

$$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \dots + w_M x^M = \sum_{j=0}^M w_j x^j \quad (1.1)$$

其中 M 是多项式的阶数 (order), x^j 表示 x 的 j 次幂。多项式系数 w_0, \dots, w_M 整体记作向量 \mathbf{w} 。注意, 虽然多项式函数 $y(x, \mathbf{w})$ 是 x 的一个非线性函数, 它是系数 \mathbf{w} 的一个线性函数。类似多项式函数的这种关于未知参数满足线性关系的函数有着重要的性质, 被叫做线性模型, 将在第3章和第4章充分讨论。

系数的值可以通过调整多项式函数拟合训练数据的方式确定。这可以通过最小化误差函数 (error function) 的方法实现。误差函数衡量了对于任意给定的 \mathbf{w} 值, 函数 $y(x, \mathbf{w})$ 与训练集数据的差别。一个简单的应用广泛的误差函数是每个数据点 x_n 的预测值 $y(x_n, \mathbf{w})$ 与目标值 t_n 的平方和。所以我们最小化

$$E(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^N \{y(x_n, \mathbf{w}) - t_n\}^2 \quad (1.2)$$

其中, 因子 $\frac{1}{2}$ 是为了后续运算方便而加入的。我们将在后续章节中讨论选择这个误差函数的原因。现在, 我们只是简单地注意一下它是一个非负的量, 并且当且仅当函数 $y(x, \mathbf{w})$ 对所有的训练数据点均做出正确预测时, 误差函数为零。平方和误差函数的几何表示见图1.3。

我们可以通过选择使得 $E(\mathbf{w})$ 尽量小的 \mathbf{w} 来解决曲线拟合问题。由于误差函数是系数 \mathbf{w} 的二

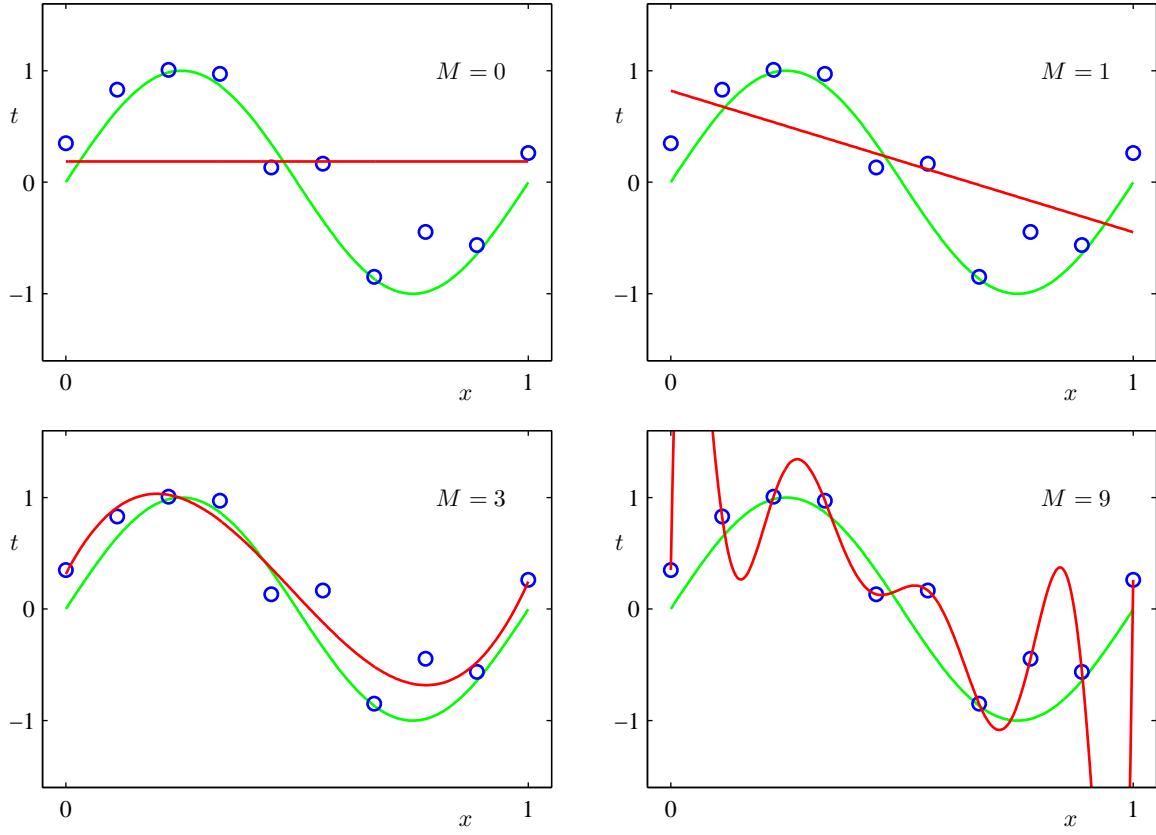


图 1.4: 不同阶数的多项式曲线, 用红色曲线表示, 拟合了图1.2中的数据集。

次函数, 因此它关于系数的导数是 \mathbf{w} 的线性函数, 所以误差函数的最小值有一个唯一解, 记作 \mathbf{w}^* , 可以用解析的方式求出。最终的多项式函数由函数 $y(x, \mathbf{w}^*)$ 给出。

选择多项式的阶数 M 也是一个问题。正如我们即将看到的那样, 这是一个被称为模型对比 (model comparison) 或者模型选择 (model selection) 的重要问题的一个特例。在图1.4中, 我们给出了4个拟合多项式的结果。多项式的阶数分别为 $M = 0, 1, 3, 9$, 数据集是图1.2所示的数据。

我们注意到常数 ($M = 0$) 和一阶 ($M = 1$) 多项式对于数据的拟合效果相当差, 很难代表函数 $\sin(2\pi x)$ 。对于图1.4中给出的例子, 三阶 ($M = 3$) 多项式似乎给出了对函数 $\sin(2\pi x)$ 的最好的拟合。当我们达到更高阶的多项式 ($M = 9$), 我们得到了对于训练数据的一个完美的拟合。事实上, 多项式函数精确地通过了每一个数据点, $E(\mathbf{w}^*) = 0$ 。然而, 拟合的曲线剧烈震荡, 就表达函数 $\sin(2\pi x)$ 而言表现很差。这种行为叫做过拟合 (over-fitting)。

正如我们之前提到的那样, 目标是通过对新数据的预测实现良好的泛化性。我们可以定量考察模型的泛化性与 M 的关系。考察的方式为: 考虑一个额外的测试集, 这个测试集由100个数据点组成, 这100个数据点的生成方式与训练集的生成方式完全相同, 但是在目标值中包含的随机噪声的值不同。对于每个 M 的选择, 我们之后可以用公式 (1.2) 计算训练集的 $E(\mathbf{w}^*)$, 也可以计算测试集的 $E(\mathbf{w}^*)$ 。有时候使用根均方 (RMS) 误差更方便。这个误差由下式定义:

$$E_{RMS} = \sqrt{2E(\mathbf{w}^*)/N} \quad (1.3)$$

其中, 除以 N 让我们能够以相同的基础对比不同大小的数据集, 平方根确保了 E_{RMS} 与目标变量 t 使用相同的规模和单位进行度量。图1.5展示了对于不同的 M 值, 训练数据和测试数据的 RMS 误差。测试集的误差衡量了对于新观察到的数据 x , 我们预测 t 的值的效果的好坏。根据图1.5, 我们看到小的 M 值会造成较大的测试集误差, 这可以归因于对应的多项式函数相当不灵活, 不能够反映出 $\sin(2\pi x)$ 的震荡。当 M 的取值为 $3 \leq M \leq 8$ 时, 测试误差较小, 对于生成函数 $\sin(2\pi x)$ 也能给出合理的模拟。对于 $M = 3$ 的情形, 可以从图1.4中看出。

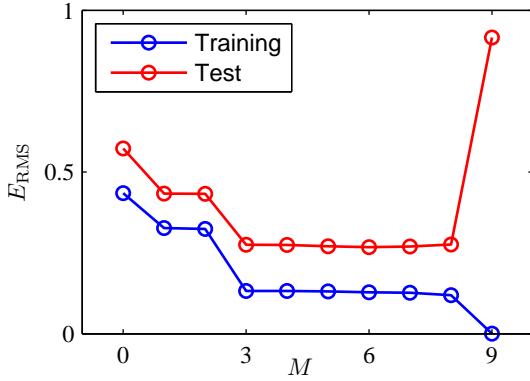


图 1.5: 公式 (1.3) 定义的根均方误差的图像，分别在训练数据集上和独立的测试数据集上对于不同的 M 进行了计算。

	$M = 0$	$M = 1$	$M = 3$	$M = 9$
w_0^*	0.19	0.82	0.31	0.35
w_1^*		-1.27	7.99	232.37
w_2^*			-25.43	-5321.83
w_3^*			17.37	48568.31
w_4^*				-231639.30
w_5^*				640042.26
w_6^*				-1061800.52
w_7^*				1042400.18
w_8^*				-557682.99
w_9^*				125201.43

表 1.1: 不同阶数的多项式的系数 w^* 的值。观察随着多项式阶数的增加，系数的大小是如何剧烈增大的。

对于 $M = 9$ 的情形，训练集的误差为 0，这符合我们的预期，因为此时的多项式函数有 10 个自由度，对应于 10 个系数 w_0, \dots, w_9 ，所以可以调节模型的参数，使得模型与训练集中的 10 个数据点精确匹配。然而，正如我们在图 1.4 中看到的那样，测试集误差变得非常大，对应的函数 $y(x, \mathbf{w}^*)$ 表现出剧烈的震荡。

这可能看起来很矛盾，因为给定阶数的多项式包含了所有低阶的多项式函数作为特殊情况。 $M = 9$ 的多项式因此能够产生至少与 $M = 3$ 一样好的结果。并且，我们可以猜想，对于新数据最好的预测是 $\sin(2\pi x)$ ，这是生成数据所使用的函数（我们稍后将会看到确实是这样）。我们知道函数 $\sin(2\pi x)$ 的幂级数展开包含所有阶数的项，所以我们可能会以为结果会随着 M 的增大而单调地变好。

我们可以更深刻地思考这个问题，通过考察不同阶数多项式的系数 w^* 的值，如表 1.1 所示。我们看到随着 M 的增大，系数的大小通常会变大。对于 $M = 9$ 的多项式，通过调节系数，让系数取相当大的正数或者负数，多项式函数可以精确地与数据匹配，但是对于数据之间的点（尤其是临近区间端点处的点），从图 1.4 可以看到函数表现出剧烈的震荡。直觉上讲，发生了这样的事情：有着更大的 M 值的更灵活的多项式被过分地调参，使得多项式被调节成了与目标值的随机噪声相符。

考察给定模型的行为随着数据集规模的变化情况也很有趣，如图 1.6 所示。我们可以看到，对已一个给定的模型复杂度，当数据集的规模增加时，过拟合问题变得不那么严重。另一种表述方式是，数据集规模越大，我们能够用来拟合数据的模型就越复杂（即越灵活）。一个粗略的启发是，数据点的数量不应该小于模型的可调节参数的数量的若干倍（比如 5 或 10）。然而，正如我们将在第 3 章看到的那样，参数的数量对于模型复杂度的大部分合理的度量来说都不是必要的。

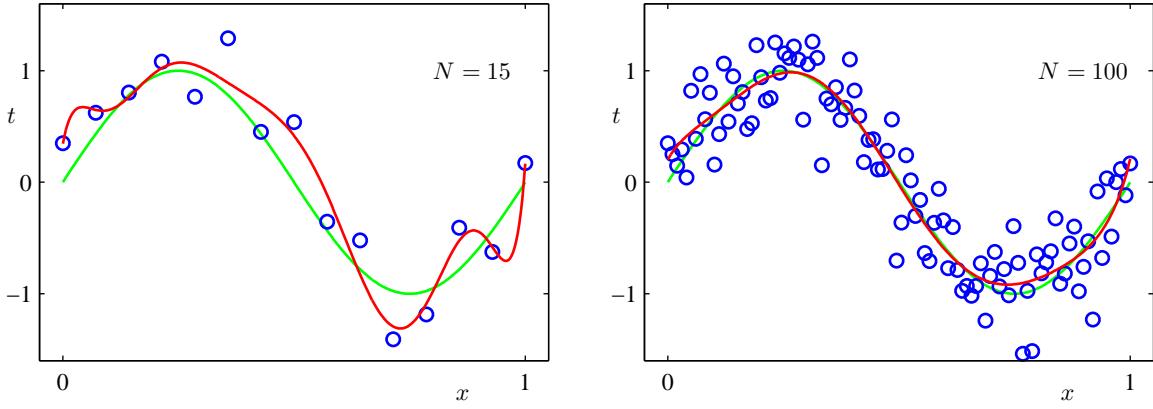


图 1.6: 使用 $M = 9$ 的多项式对 $M = 15$ 个数据点 (左图) 和 $N = 100$ 个数据点 (右图) 通过最小化平方和误差函数的方法得到的解。我们看到增大数据集的规模会减小过拟合问题。

并且, 令人无法满意的一点是, 不得不根据可得到的训练集的规模限制参数的数量。似乎更加合理的是, 根据待解决的问题的复杂性来选择模型的复杂性。我们将会看到, 寻找模型参数的最小平方方法代表了最大似然 (maximum likelihood) (将在1.2.5节讨论) 的一种特殊情形, 并且过拟合问题可以被理解为最大似然的一个通用属性。通过使用一种贝叶斯 (Bayesian) 方法, 过拟合问题可以被避免。我们将会看到, 从贝叶斯的观点来看, 对于模型参数的数量超过数据点数量的情形, 没有任何难解之处。实际上, 一个贝叶斯模型中, 参数的有效 (effective) 数量会自动根据数据集的规模调节。

但是现在, 继续使用当前的方法还是很有用的。并且考虑在实际中我们可以如何应用有限规模的数据集也是很有意义的。在这种情况下, 我们可能期望建立相对复杂和灵活的模型。经常用来控制过拟合现象的一种技术是正则化 (regularization)。这种技术涉及到给误差函数 (1.2) 增加一个惩罚项, 使得系数不会达到很大的值。这种惩罚项最简单的形式采用所有系数的平方和的形式。这推导出了误差函数的修改后的形式:

$$\tilde{E}(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^N \{y(x_n, \mathbf{w}) - t_n\}^2 + \frac{\lambda}{2} \|\mathbf{w}\|^2 \quad (1.4)$$

其中 $\|\mathbf{w}\|^2 \equiv \mathbf{w}^T \mathbf{w} = w_0^2 + w_1^2 + \dots + w_M^2$, 系数 λ 控制了正则化项相对于平方和误差项的重要性。注意, 通常系数 w_0 从正则化项中省略, 因为包含 w_0 会使得结果依赖于目标变量原点的选择 (Hastie et al., 2001)。 w_0 也可以被包含在正则化项中, 但是必须有自己的正则化系数 (我们将在5.5.1节详细讨论这个问题)。公式 (1.4) 中的误差函数也可以用解析的形式求出最小值。像这样的技术在统计学的文献中被叫做收缩 (shrinkage) 方法, 因为这种方法减小了系数的值。二次正则项的一个特殊情况被称为山脊回归 (ridge regression) (Hoerl and Kennard, 1970)。在神经网络的情形中, 这种方法被叫做权值衰减 (weight decay)。

图1.7展示了在 $M = 9$ 的情况下用与之前相同的数据拟合多项式的结果。这次使用的是公式 (1.4) 的正则化误差函数。我们看到, 对于 $\ln \lambda = -18$, 过拟合现象被压制, 我们可以得到关于本质函数 $\sin(2\pi x)$ 的一个更好的模拟。但是如果我们将 λ 选择的过大, 我们又得到了一个不好的结果, 如图1.7所示的 $\ln \lambda = 0$ 的情形。拟合的多项式的对应的系数在表1.2中给出, 表明正则化在减小系数的值方面产生了预期的效果。

正则化项对于泛化错误的影响可以从图1.8看出。图1.8给出了训练集和测试集的RMS误差与 $\ln \lambda$ 的关系。我们看到, 在效果上, λ 控制了模型的复杂性, 因此决定了过拟合的程度。

模型复杂度是一个重要的话题, 将在1.3节详细讨论。这里我们简单地说一下, 如果我们试着用最小化误差函数的方法解决一个实际的应用问题, 那么我们不得不寻找一种方式来确定模型复杂度的合适值。上面的结果给出了一种完成这一目标的简单方式, 即通过把给定的数据中的一部分从测试集中分离出, 来确定系数 \mathbf{w} 。这个分离出来的验证集 (validation set), 也被称为拿出集 (hold-out set), 用来最优化模型的复杂度 (M 或者 λ)。但是在许多情况下, 这太浪费有价值的训练数据了, 我们不得不寻找更高级的方法。

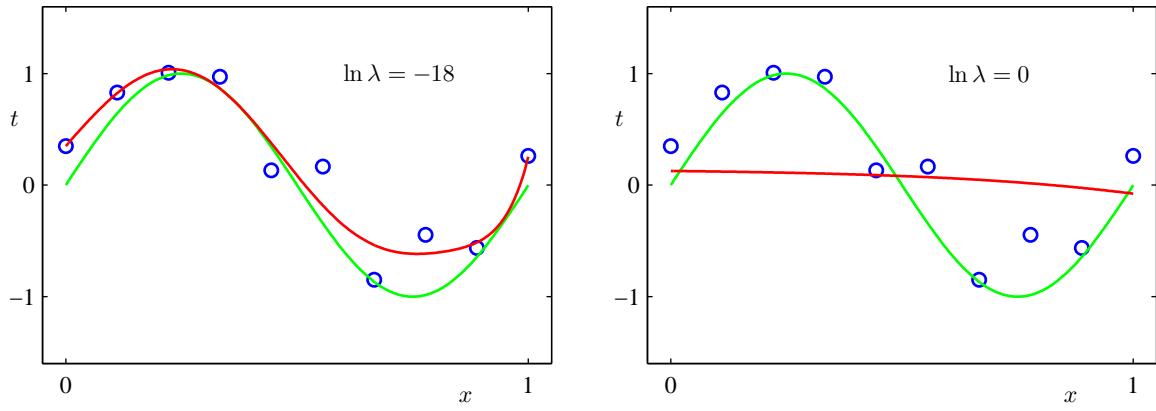


图 1.7: 使用正则化的误差函数 (1.4)，用 $M = 9$ 的多项式拟合图 1.2 中的数据集。其中正则化参数 λ 选择了两个值，分别对应于 $\ln \lambda = -18$ 和 $\ln \lambda = 0$ 。没有正则化项的情形，即 $\lambda = 0$ ，对应于 $\ln \lambda = -\infty$ ，在图 1.4 的右下角给出。

	$\ln \lambda = -\infty$	$\ln \lambda = -18$	$\ln \lambda = 0$
w_0^*	0.35	0.35	0.13
w_1^*	232.37	4.74	-0.05
w_2^*	-5321.83	-0.77	-0.06
w_3^*	48568.31	-31.97	-0.05
w_4^*	-231639.30	-3.89	-0.03
w_5^*	640042.26	55.28	-0.02
w_6^*	-1061800.52	41.32	-0.01
w_7^*	1042400.18	-45.95	-0.00
w_8^*	-557682.99	-91.53	0.00
w_9^*	125201.43	72.68	0.01

表 1.2: 不同的正则化参数 λ 下， $M = 9$ 的多项式的系数 w^* 的值。注意， $\ln \lambda = -\infty$ 对应于没有正则化的模型，即图 1.4 右下角的模型。我们看到，随着 λ 的增大，系数的大小逐渐变小。

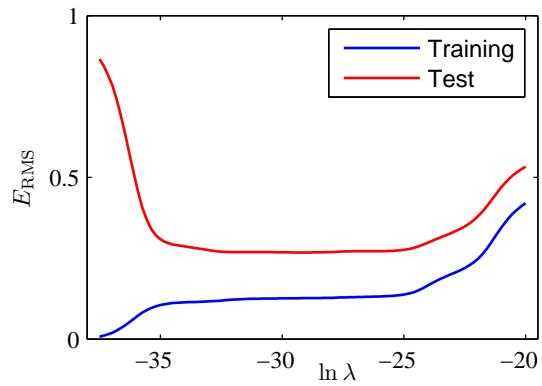


图 1.8: 对于 $M = 9$ 的多项式，均方根误差 (1.3) 与 $\ln \lambda$ 的关系。

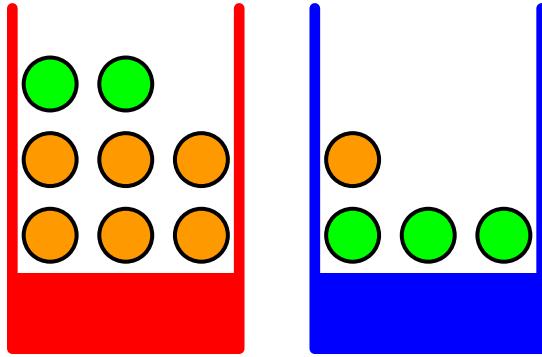


图 1.9: 我们使用一个简单的例子来说明概率论的基本思想。有两个不同颜色的盒子，每个盒子中都有水果，苹果用绿色表示，橘子用橙色表示。

目前我们关于多项式拟合的讨论大量地依赖于直觉。我们现在寻找一个更加形式化的方法解决模式识别中的问题。我们要使用概率论的方法。概率论不仅提供了本书后续几乎所有章节的基础，它也能让我们更深刻地理解本章中我们通过多项式拟合的问题引出的重要概念，能让我们把这些概念扩展到更复杂的情况。

1.2 概率论

在模式识别领域的一个关键概念是不确定性的概念。它可以由测量的误差引起，也可以由数据集的有限大小引起。概率论提供了一个合理的框架，用来对不确定性进行量化和计算。概率论还构成了模式识别的一个中心基础。当与决策论（1.5节讨论）结合，概率论让我们能够根据所有能得到的信息做出最优的预测，即使信息可能是不完全的或者是含糊的。

我们将通过一个简单的例子介绍概率论的基本概念。假设我们由两个盒子，一个红色的，一个蓝色的，红盒子中有2个苹果和6个橘子，蓝盒子中有3个苹果和1个橘子（如图1.9所示）。现在假定我们随机选择一个盒子，从这个盒子中我们随机选择一个水果，观察一下选择了哪种水果，然后放回盒子中。假设我们重复这个过程很多次。假设我们在40%的时间中选择红盒子，在60%的时间中选择蓝盒子，并且我们选择盒子中的水果时是等可能选择的。

在这个例子中，我们要选择的盒子的颜色是一个随机变量，记作 B 。这个随机变量可以取两个值中的一个，即 r （对应红盒子）或 b （对应蓝盒子）。类似地，水果的种类也是一个随机变量，记作 F 。它可以取 a （苹果）或者 o （橘子）。

开始阶段，我们把一个事件的概率定义为事件发生的次数与试验总数的比值，假设总试验次数趋于无穷。因此选择红盒子的概率为 $\frac{4}{10}$ ，选择蓝盒子的概率为 $\frac{6}{10}$ 。我们把这些概率分布记作 $p(B = r) = \frac{4}{10}$ 和 $p(B = b) = \frac{6}{10}$ 。注意，根据定义，概率一定位于区间 $[0, 1]$ 内。并且，如果事件是相互独立的，并且包含所有可能的输出（例如在这个例子中，盒子一定要么是红色，要么是蓝色），那么我们看到那些事件的概率的和一定等于1。

我们现在可以问这样的问题：选择到苹果的整体概率是多少？或者，假设我们选择了橘子，我们选择的盒子是蓝盒子的概率是多少？我们可以回答这种问题，事实上也可以回答与模式识别相关的比这些复杂得多的问题。前提是掌握概率论的两个基本规则：加和规则（sum rule）、乘积规则（product rule）。获得了这些规则之后，我们将重新回到我们的水果盒子的例子中。

为了推导概率的规则，考虑图1.10所示的稍微一般一些的情形。这个例子涉及到两个随机变量 X 和 Y （例如可以是上面例子中“盒子”和“水果”的随机变量）。我们假设 X 可以取任意的 x_i ，其中 $i = 1, \dots, M$ ，并且 Y 可以取任意的 y_j ，其中 $j = 1, \dots, L$ 。考虑 N 次试验，其中我们对 X 和 Y 都进行取样，把 $X = x_i$ 且 $Y = y_j$ 的试验的数量记作 n_{ij} 。并且，把 X 取值 x_i （与 Y 的取值无关）的试验的数量记作 c_i ，类似地，把 Y 取值 y_j 的试验的数量记作 r_j 。

X 取值 x_i 且 Y 取值 y_j 的概率被记作 $p(X = x_i, Y = y_j)$ ，被称为 $X = x_i$ 和 $Y = y_j$ 的联合概率（joint probability）。它的计算方法为落在单元格 i, j 的点的数量与点的总数的比值，即：

$$p(X = x_i, Y = y_j) = \frac{n_{ij}}{N} \quad (1.5)$$

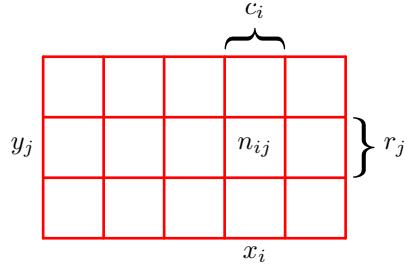


图 1.10: 我们可以这样推导概率的加和规则和乘积规则: 考虑两个随机变量, X , 取值为 $\{x_i\}$, 其中 $i = 1, \dots, M$, 和 Y , 取值为 $\{y_j\}$, 其中 $j = 1, \dots, L$ 。在这个例子中, 我们取 $M = 5$ 和 $L = 3$ 。如果我们考虑这些变量的总计 N 个实例, 那么我们将 $X = x_i$ 且 $Y = y_j$ 的实例的数量记作 n_{ij} , 它是对应的单元格中点的数量。列 i 中的点的数量, 对应于 $X = x_i$, 被记作 c_i , 行 j 中的点的数量, 对应于 $Y = y_j$, 被记作 r_j 。

这里我们隐式地考虑极限 $N \rightarrow \infty$ 。类似地, X 取值 x_i (与 Y 取值无关) 的概率被记作 $p(X = x_i)$, 计算方法为落在列 i 上的点的数量与点的总数的比值, 即:

$$p(X = x_i) = \frac{c_i}{N} \quad (1.6)$$

由于图 1.10 中列 i 上的实例总数就是这列的所有单元格中实例的数量之和, 我们有 $c_i = \sum_j n_{ij}$, 因此根据公式 (1.5) 和公式 (1.6), 我们有:

$$p(X = x_i) = \sum_{j=1}^L p(X = x_i, Y = y_j) \quad (1.7)$$

这是概率的加和规则 (sum rule)。注意, $p(X = x_i)$ 有时被称为边缘概率 (marginal probability), 因为它通过把其他变量 (本例中的 Y) 边缘化或者加和得到。

如果我们只考虑那些 $X = x_i$ 的实例, 那么这些实例中 $Y = y_j$ 的实例所占的比例被写成 $p(Y = y_j | X = x_i)$, 被称为给定 $X = x_i$ 的 $Y = y_j$ 的条件概率 (conditional probability)。它的计算方式为: 计算落在单元格 i, j 的点的数量与列 i 的点的数量的比值, 即:

$$p(Y = y_j | X = x_i) = \frac{n_{ij}}{c_i} \quad (1.8)$$

从公式 (1.5)、公式 (1.6) 和 (1.8), 我们可以推导出下面的关系:

$$p(X = x_i, Y = y_j) = \frac{n_{ij}}{N} = \frac{n_{ij}}{c_i} \cdot \frac{c_i}{N} = p(Y = y_j | X = x_i)p(X = x_i) \quad (1.9)$$

这被称为概率的乘积规则 (product rule)。

到现在为止, 我们相当仔细地区分随机变量 (例如水果例子中的盒子 B) 和随机变量可以取的值 (例如盒子是红色时取值为 r)。因此 B 取值为 r 的概率被记作 $p(B = r)$ 。虽然这种记法避免了歧义性, 这种记号相当笨拙, 并且在很多情况下没有必要。相反, 我们简单地用 $p(B)$ 表示随机变量 B 的分布, $p(r)$ 表示这个分布对于特定的值 r 的估计, 假定这种表达方式在给定上下文的情况下不会造成歧义。

使用这种简洁的记法, 我们可以用下面的形式表示概率论的两条基本规则:

$$\textbf{sum rule } p(X) = \sum_Y p(X, Y) \quad (1.10)$$

$$\textbf{product rule } p(X, Y) = p(Y | X)p(X) \quad (1.11)$$

这里 $p(X, Y)$ 是联合概率, 可以表述为“ X 且 Y 的概率”。类似地, $p(Y | X)$ 是条件概率, 可以表述为“给定 X 的条件下 Y 的概率”, $p(X)$ 是边缘概率, 可以简单地表述为“ X 的概率”。这两个简单的规则组成了我们在全书中使用的全部概率推导的基础。

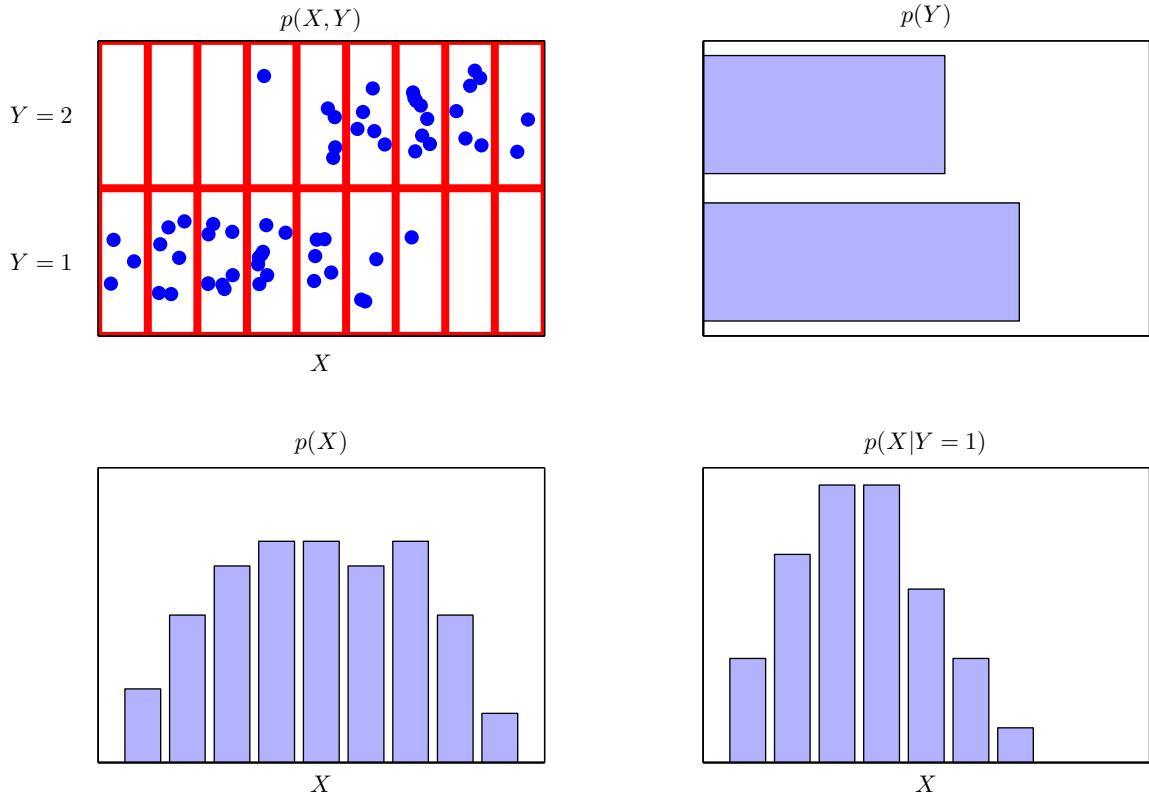


图 1.11: 两个变量 X 和 Y 上的概率分布的一个例子。 X 可以取 9 个可能的值，而 Y 可以去 2 个可能的值。左上图给出了从这两个变量的联合概率分布中抽取的 60 个样本点。剩下的图给出了估计边缘概率分布 $p(X)$ 和 $p(Y)$ 的直方图，以及条件概率分布 $p(X | Y = 1)$ 的直方图，这个条件概率分布对应于左上图的下面一行。

根据乘积规则，以及对称性 $p(X, Y) = p(Y, X)$ ，我们立即得到了下面的两个条件概率之间的关系：

$$p(Y | X) = \frac{p(X | Y)p(Y)}{p(X)} \quad (1.12)$$

这被称为贝叶斯定理 (Bayes' theorem)，在模式识别和机器学习领域扮演者中心角色。使用加和规则，贝叶斯定理中的分母可以用出现在分子中的项表示：

$$p(X) = \sum_Y p(X | Y)p(Y) \quad (1.13)$$

我们可以把贝叶斯定理的分母看做归一化常数，用来确保公式 (1.12) 左侧的条件概率对于所有的 Y 的取值之和为 1。

在图 1.11 中，我们给出了一个简单的涉及到两个变量的联合分布的例子，来说明边缘分布和条件分布的概念。这里我们从联合分布中抽取了有限数量 $N = 60$ 的样本，展示在了左上角。在右上角是数据点取两种 Y 值的比例的直方图。根据概率的定义，这些比例在 $N \rightarrow \infty$ 时将会等于对应的概率 $p(Y)$ 。我们可以把直方图看成在给定有限数量的数据点的情形下，对概率分布建模的一种简单的方式。使用数据对概率分布建模是统计模式识别的核心，在本书中将会详细介绍。图 1.11 中剩下的两张图分别给出了估计 $p(X)$ 和 $p(X | Y = 1)$ 的直方图。

现在让我们回到水果盒子的例子。现在我们将再一次清楚地区分随机变量和它的实例。我们看到选择红盒子或者蓝盒子的概率分别由下式给出：

$$p(B = r) = \frac{4}{10} \quad (1.14)$$

$$p(B = b) = \frac{6}{10} \quad (1.15)$$

注意，这两个式子满足 $p(B = r) + p(B = b) = 1$ 。

现在假设我们随机选择一个盒子，结果发现是蓝盒子。然后我们选择苹果的概率就是蓝盒子中苹果的比例（等于 $\frac{3}{4}$ ），因此 $p(F = a | B = b) = \frac{3}{4}$ 。实际上，我们可以写出给定盒子种类的条件下水果种类的全部四个概率：

$$p(F = a | B = r) = \frac{1}{4} \quad (1.16)$$

$$p(F = o | B = r) = \frac{3}{4} \quad (1.17)$$

$$p(F = a | B = b) = \frac{3}{4} \quad (1.18)$$

$$p(F = o | B = b) = \frac{1}{4} \quad (1.19)$$

还要注意，这些概率是归一化的，所以

$$p(F = a | B = r) + p(F = o | B = r) = 1 \quad (1.20)$$

类似地

$$p(F = a | B = b) + p(F = o | B = b) = 1 \quad (1.21)$$

我们现在使用加和规则和乘积规则来计算选择一个苹果的整体概率：

$$\begin{aligned} p(F = a) &= p(F = a | B = r)p(B = r) + p(F = a | B = b)p(B = b) \\ &= \frac{1}{4} \times \frac{4}{10} + \frac{3}{4} \times \frac{6}{10} = \frac{11}{20} \end{aligned} \quad (1.22)$$

使用加和规则，可以计算出 $p(F = o) = 1 - \frac{11}{20} = \frac{9}{20}$ 。

反过来，假设我们知道被选择的水果是橘子，我们想知道它来自于哪个盒子。这需要我们在给定水果种类的条件下估计盒子的概率分布，然而公式 (1.16) 至公式 (1.19) 给出的是在已知盒子颜色的情形下水果的概率分布。我们可以使用贝叶斯定理来解决这种逆转的条件概率问题：

$$p(B = r | F = o) = \frac{p(F = o | B = r)p(B = r)}{p(F = o)} = \frac{3}{4} \times \frac{4}{10} \times \frac{20}{9} = \frac{2}{3} \quad (1.23)$$

根据加和规则，我们可以计算出 $p(B = b | F = o) = 1 - \frac{2}{3} = \frac{1}{3}$ 。

我们可以按照下面的方式表述贝叶斯定理。如果在我们知道水果的种类之前，有人问我们哪个盒子被选中，那么我们能够得到的最多的信息就是概率 $p(B)$ 。我们把这个叫做先验概率 (prior probability)，因为它是在我们观察到水果种类之前就能够得到的概率。一旦我们知道水果是橘子，我们就能够使用贝叶斯定理来计算概率 $p(B | F)$ 。这个被称为后验概率 (posterior probability)，因为它是观察到 F 之后的概率。注意，在这个例子中，选择红盒子的先验概率是 $\frac{4}{10}$ ，所以与红盒子相比，我们更有可能选择蓝盒子。然而，一旦我们观察到选择的水果是橘子，我们发现红盒子的后验概率现在是 $\frac{2}{3}$ ，因此现在实际上更可能选择的是红盒子。这个结果与我们的直觉相符，因为红盒子中橘子的比例比蓝盒子高得多，因此观察到水果是橘子这件事提供给我们更强的证据来选择红盒子。事实上，这个证据相当强，已经超过了先验的假设，使得红盒子被选择的可能性大于蓝盒子。

最后，如果两个变量的联合分布可以分解成两个边缘分布的乘积，即 $p(X, Y) = p(X)p(Y)$ ，那么我们说 X 和 Y 相互独立 (independent)。根据乘积规则，我们可以得到 $p(Y | X) = p(Y)$ ，因此对于给定 X 的条件下的 Y 的条件分布实际上独立于 X 的值。例如，在我们的水果盒子的例子中，如果每个盒子包含同样比例的苹果和橘子，那么 $p(F | B) = P(F)$ ，从而选择苹果的概率就与选择了哪个盒子无关。

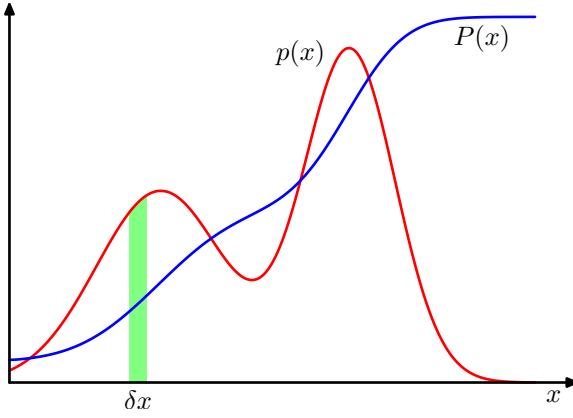


图 1.12: 离散变量的概率的概念可以扩展到连续变量上的概率分布 $p(x)$ 。 x 位于区间 $(x, x + \delta x)$ 的概率为 $p(x)\delta x$, 其中 $\delta x \rightarrow 0$ 。概率密度可以表示为累计密度函数 $P(x)$ 的导数。

1.2.1 概率密度

既然考虑了定义在离散事件集合上的概率, 我们也希望考虑与连续变量相关的概率。我们会把我们的讨论限制在一个相对非正式的形式上。如果一个实值变量 x 的概率落在区间 $(x, x + \delta x)$ 的概率由 $p(x)\delta x$ 给出 ($\delta x \rightarrow 0$), 那么 $p(x)$ 叫做 x 的概率密度 (probability density)。图 1.12 说明了这个概念。 x 位于区间 (a, b) 的概率由下式给出:

$$p(x \in (a, b)) = \int_a^b p(x) dx \quad (1.24)$$

由于概率是非负的, 并且 x 的值一定位于实数轴上得某个位置, 因此概率密度一定满足下面两个条件:

$$p(x) \geq 0 \quad (1.25)$$

$$\int_{-\infty}^{\infty} p(x) dx = 1 \quad (1.26)$$

在变量以非线性的形式变化的情况下, 概率密度函数通过 Jacobian 因子变换为与简单的函数不同的形式。例如, 假设我们考虑一个变量的变化 $x = g(y)$, 那么函数 $f(x)$ 就变成了 $\tilde{f}(y) = f(g(y))$ 。现在让我们考虑一个概率密度函数 $p_x(x)$, 它对应于一个关于新变量 y 的密度函数 $p_y(y)$, 其中下标的不同表明了 $p_x(x)$ 和 $p_y(y)$ 是不同的密度函数这一事实。对于很小的 δx 的值, 落在区间 $(x, x + \delta x)$ 内的观测会被变换到区间 $(y, y + \delta y)$ 中。其中 $p_x(x)\delta x \simeq p_y(y)\delta y$, 因此

$$p_y(y) = p_x(x) \left| \frac{dx}{dy} \right| = p_x(g(y)) |g'(y)| \quad (1.27)$$

这个性质的一个结果就是, 概率密度最大值的概念取决于变量的选择。

位于区间 $(-\infty, z)$ 的 x 的概率由累积分布函数 (cumulative distribution function) 给出。定义为:

$$P(z) = \int_{-\infty}^z p(x) dx \quad (1.28)$$

这满足 $P'(x) = p(x)$, 如图 1.12 所示。

如果我们有几个连续变量 x_1, \dots, x_D , 整体记作向量 \mathbf{x} , 那么我们可以定义联合概率密度 $p(\mathbf{x}) = p(x_1, \dots, x_D)$, 使得 \mathbf{x} 落在包含点 \mathbf{x} 的无穷小体积 $\delta \mathbf{x}$ 的概率由 $p(\mathbf{x})\delta \mathbf{x}$ 给出。多变量概率密度必须满足

$$p(\mathbf{x}) \geq 0 \quad (1.29)$$

$$\int p(\mathbf{x}) d\mathbf{x} = 1 \quad (1.30)$$

其中，积分必须在整个 x 空间上进行。我们也可以考虑离散变量和连续变量相结合的联合概率分布。

注意，如果 x 是一个离散变量，那么 $p(x)$ 有时被叫做概率质量函数（probability mass function），因为它可以被看做集中在合法的 x 值处的“概率质量”的集合。

概率的加和规则和乘积规则以及贝叶斯规则，同样可以应用于概率密度函数的情形，也可以应用于离散变量与连续变量相结合的情形。例如，如果 x 和 y 是两个实数变量，那么加和规则和乘积规则的形式为

$$p(x) = \int p(x, y) dy \quad (1.31)$$

$$p(x, y) = p(y | x)p(x) \quad (1.32)$$

形式化地证明连续变量的加和规则和乘积规则（Feller, 1966）需要一个被称为测度论（measure theory）的数学分支，不在本书的讨论范围内。然而，它的正确性可以非形式化地观察出来。把每个实数变量除以区间的宽度 Δ ，然后考虑这些区间上的概率分布。取极限 $\Delta \rightarrow 0$ ，把求和转化为积分，就得到了预期的结果。

1.2.2 期望和协方差

涉及到概率的一个重要的操作是寻找函数的加权平均值。在概率分布 $p(x)$ 下，函数 $f(x)$ 的平均值被称为 $f(x)$ 的期望（expectation），记作 $\mathbb{E}[f]$ 。对于一个离散变量，它的定义为

$$\mathbb{E}[f] = \sum_x p(x)f(x) \quad (1.33)$$

因此平均值根据 x 的不同值的相对概率加权。在连续变量的情形下，期望以对应的概率密度的积分的形式表示

$$\mathbb{E}[f] = \int p(x)f(x) dx \quad (1.34)$$

两种情形下，如果我们给定有限数量的 N 个点，这些点满足某个概率分布或者概率密度函数，那么期望可以通过求和的方式估计

$$\mathbb{E}[f] \simeq \frac{1}{N} \sum_{n=1}^N f(x_n) \quad (1.35)$$

在第11章讨论取样方法时，我们将会经常用到这个结果。当 $N \rightarrow \infty$ 时，公式（1.35）的估计就会变得精确。

有时，我们会考虑多变量函数的期望。这种情形下，我们可以使用下标来表明被平均的是哪个变量，例如

$$\mathbb{E}_x[f(x, y)] \quad (1.36)$$

表示函数 $f(x, y)$ 关于 x 的分布的平均。注意， $\mathbb{E}_x[f(x, y)]$ 是 y 的一个函数。

我们也可以考虑关于一个条件分布的条件期望（conditional expectation），即

$$\mathbb{E}_x[f | y] = \sum_x p(x | y)f(x) \quad (1.37)$$

连续变量情形下的定义与此类似。

$f(x)$ 的方差（variance）被定义为

$$\text{var}[f] = \mathbb{E}[(f(x) - \mathbb{E}[f(x)])^2] \quad (1.38)$$

它度量了 $f(x)$ 在均值 $\mathbb{E}[f(x)]$ 附近变化性的大小。把平方项展开，我们看到方差也可以写成 $f(x)$ 和 $f(x)^2$ 的期望的形式

$$\text{var}[f] = \mathbb{E}[f(x)^2] - \mathbb{E}[f(x)]^2 \quad (1.39)$$

特别地，我们可以考虑变量 x 自身的方差，它由下式给出：

$$\text{var}[x] = \mathbb{E}[x^2] - \mathbb{E}[x]^2 \quad (1.40)$$

对于两个随机变量 x 和 y ，协方差（covariance）被定义为

$$\text{cov}[x, y] = \mathbb{E}_{x,y}[\{x - \mathbb{E}[x]\}\{y - \mathbb{E}[y]\}] = \mathbb{E}_{x,y}[xy] - \mathbb{E}[x]\mathbb{E}[y] \quad (1.41)$$

它表示在多大程度上 x 和 y 会共同变化。如果 x 和 y 相互独立，那么它们的协方差为0。

在两个随机向量 \mathbf{x} 和 \mathbf{y} 的情形下，协方差是一个矩阵

$$\text{cov}[\mathbf{x}, \mathbf{y}] = \mathbb{E}_{\mathbf{x}, \mathbf{y}}[\{\mathbf{x} - \mathbb{E}[\mathbf{x}]\}\{\mathbf{y}^T - \mathbb{E}[\mathbf{y}^T]\}] = \mathbb{E}_{\mathbf{x}, \mathbf{y}}[\mathbf{x}\mathbf{y}^T] - \mathbb{E}[\mathbf{x}]\mathbb{E}[\mathbf{y}^T] \quad (1.42)$$

如果我们考虑向量 \mathbf{x} 各个分量之间的协方差，那么我们可以将记号稍微简化一下： $\text{cov}[\mathbf{x}] \equiv \text{cov}[\mathbf{x}, \mathbf{x}]$

1.2.3 贝叶斯概率

本章目前为止，我们根据随机重复事件的频率来考察概率。我们把这个叫做经典的(classical)或者频率学家(frequentist)的关于概率的观点。现在我们转向更加通用的贝叶斯(Bayesian)观点。这种观点中，频率提供了不确定性的一个定量化描述。

考虑一个不确定性事件，例如月球是否曾经处于围绕太阳的自己的轨道上，或者本世纪末北极冰盖是否会消失。这些事件无法重复多次，因此我们无法像之前水果盒子那样定义概率。但是，我们通常会有一些想法，例如，北极冰盖融化的速度等等。如果我们获得了新鲜的证据，例如人造卫星收集到了一些新的修正信息，我们可能就会修正我们对于冰盖融化速度的观点。我们估计冰盖融化速度会影响我们采取的措施，例如我们会努力减少温室气体的排放。在这样的情况下，我们可能希望能够定量地描述不确定性，并且根据少量新的证据对不确定性进行精确的修改，对接下来将要采取的动作进行修改，或者对最终的决策进行修改。这可以通过一种优雅的通用的贝叶斯概率观点来实现。

然而，在作出合理的推断时，如果我们想要尊重常识，那么使用概率论来表达不确定性不是可选的，而是不可避免的。例如，Cox (1946) 证明，如果用数值来表示置信的程度，那么编码了这种置信度中符合常识的一组简单的公理能够唯一地推导出一组规则来操控置信的程度，这组规则等价于概率的加和规则和乘积规则。这首次含糊地证明了概率论能够被当做布尔逻辑在涉及到不确定性的问题时的扩展(Jaynes, 2003)。许多其他学者也发表了不同的性质集合或者公理集合，这些性质或公理是不确定性的度量应该满足的(Ramsey, 1931; Good, 1950; Savage, 1961; deFinetti, 1970; Lindley, 1982)。在这些情形下，结果的数值量的行为精确地符合概率的规则。因此把这些量看成(贝叶斯观点的)概率就很自然了。

在模式识别领域，对概率有一个更加通用的观点同样是很帮助的。考虑1.1节讨论过的多项式曲线拟合的例子。对于观察到的变量 t_n 这一随机值的概率，应用频率学家的观点似乎是很合理的。然而，我们想针对模型参数 \mathbf{w} 的合适选择进行强调和定量化。我们将会看到，从贝叶斯的观点来看，我们能够使用概率论来描述模型参数(例如 \mathbf{w})的不确定性，或者模型本身的选择。

贝叶斯定理现在有了一个新的意义。回忆一下，在水果盒子的例子中，水果种类的观察提供了相关的信息，改变了选择了红盒子的概率。在那个例子中，贝叶斯定理通过将观察到的数据融合，来把先验概率转化为后验概率。正如我们将看到的，在我们对数量(例如多项式曲线拟合例子中的参数 \mathbf{w})进行推断时，我们可以采用一个类似的方法。在观察到数据之前，我们有一些关于参数 \mathbf{w} 的假设，这以先验概率 $p(\mathbf{w})$ 的形式给出。观测数据 $\mathcal{D} = \{t_1, \dots, t_N\}$ 的效果可以通过条件概率 $p(\mathcal{D} | \mathbf{w})$ 表达，我们将在1.2.5节看到这个如何被显式地表达出来。贝叶斯定理的形式为

$$p(\mathbf{w} | \mathcal{D}) = \frac{p(\mathcal{D} | \mathbf{w})p(\mathbf{w})}{p(\mathcal{D})} \quad (1.43)$$

它让我们能够通过后验概率 $p(\mathbf{w} | \mathcal{D})$ ，在观测到 \mathcal{D} 之后估计 \mathbf{w} 的不确定性。

贝叶斯定理右侧的量 $p(\mathcal{D} | \mathbf{w})$ 由观测数据集 \mathcal{D} 来估计，可以被看成参数向量 \mathbf{w} 的函数，被称为似然函数（likelihood function）。它表达了在不同的参数向量 \mathbf{w} 下，观测数据出现的可能性的大小。注意，似然函数不是 \mathbf{w} 的概率分布，并且它关于 \mathbf{w} 的积分并不（一定）等于1。

给定似然函数的定义，我们可以用自然语言表述贝叶斯定理

$$\text{posterior} \propto \text{likelihood} \times \text{prior} \quad (1.44)$$

其中所有的量都可以看成 \mathbf{w} 的函数。公式 (1.43) 的分母是一个归一化常数，确保了左侧的后验概率分布是一个合理的概率密度，积分为1。实际上，对公式 (1.43) 的两侧关于 \mathbf{w} 进行积分，我们可以用后验概率分布和似然函数来表达贝叶斯定理的分母

$$p(\mathcal{D}) = \int p(\mathcal{D} | \mathbf{w}) p(\mathbf{w}) d\mathbf{w} \quad (1.45)$$

在贝叶斯观点和频率学家观点中，似然函数 $p(\mathcal{D} | \mathbf{w})$ 都起着重要的作用。然而，在两种观点中，使用的方式有着本质的不同。在频率学家的观点中， \mathbf{w} 被认为是一个固定的参数，它的值由某种形式的“估计”来确定，这个估计的误差通过考察可能的数据集 \mathcal{D} 的概率分布来得到。相反，从贝叶斯的观点来看，只有一个数据集 \mathcal{D} （即实际观测到的数据集），参数的不确定性通过 \mathbf{w} 的概率分布来表达。

频率学家广泛使用的一个估计是最大似然（maximum likelihood）估计，其中 \mathbf{w} 的值是使似然函数 $p(\mathcal{D} | \mathbf{w})$ 达到最大值的 \mathbf{w} 值。这对应于选择使观察到的数据集出现概率最大的 \mathbf{w} 的值。在机器学习的文献中，似然函数的负对数被叫做误差函数（error function）。由于负对数是单调递减的函数，最大化似然函数等价于最小化误差函数。

一种决定频率学家的误差的方法是自助法（bootstrap）（Efron, 1979; Hastie et al., 2001）。这种方法中，多个数据集使用下面的方式创造。假设我们的原始数据集由 N 个数据点 $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ 组成。我们可以通过随机从 \mathbf{X} 中抽取 N 个点的方式，创造一个新的数据集 \mathbf{X}_B 。抽取时可以有重复，因此某些 \mathbf{X} 中的数据点可能在 \mathbf{X}_B 中有重复，而其他的在 \mathbf{X} 中的点会在 \mathbf{X}_B 中缺失。这个过程可以重复 L 次，生成 L 个数据集，每个数据集的大小都是 N ，每个数据集是通过对原始数据集 \mathbf{X} 采样得到的。参数估计的统计准确性之后就可以通过考察不同的自助数据集之间的预测的变化性来进行评估。

贝叶斯观点的一个优点是对先验概率的包含是很自然的事情。例如，假定投掷一枚普通的硬币3次，每次都是正面朝上。一个经典的最大似然模型在估计硬币正面朝上的概率时，结果会是1，表示所有未来的投掷都会是正面朝上！相反，一个带有任意的合理的先验的贝叶斯的方法将不会得出这么极端的结论。

关于频率学家的观点和贝叶斯的观点的相对优势有很多争论。事实上并没有纯粹的频率学家观点或者贝叶斯的观点。例如，针对贝叶斯方法的一种广泛的批评就是先验概率的选择通常是为了计算的方便而不是为了反映出任何先验的知识。某些人甚至把贝叶斯观点中结论对于先验选择的依赖性的本质看成困难的来源。减少对于先验的依赖性是所谓无信息（noninformative）先验的一个研究动机。然而，这会导致比较不同模型时的困难，并且实际上当先验选择不好的时候，贝叶斯方法有很大的可能性会给出错误的结果。频率学家估计方法在一定程度上避免了这一问题，并且例如交叉验证的技术在模型比较等方面也很有用。

本书着重强调贝叶斯观点，这反映出过去几年贝叶斯方法在实际应用中重要性的逐渐增长。本书也会在必要的时候讨论有用的频率学家观点下的概念。

虽然贝叶斯的框架起源于18世纪，但是贝叶斯方法的实际应用在很长时间内都被执行完整的贝叶斯步骤的困难性所限制，尤其是需要在整个参数空间求和或者求积分，这在做预测或者比较不同的模型时必须进行。取样方法的发展，例如马尔科夫链蒙特卡罗（在第11章讨论），以及计算机速度和存储容量的巨大提升，打开了在相当多的问题中使用贝叶斯技术的大门。蒙特卡罗方法非常灵活，可以应用于许多种类的模型。然而，它们在计算上很复杂，主要应用于小规模问题。

最近，许多高效的判别式方法被提出来，例如变种贝叶斯（variational Bayes）和期望传播（expectation propagation）。这些提供了一种可选的补充的取样方法，让贝叶斯方法能够应用于大规模的应用中（Blei et al., 2003）。

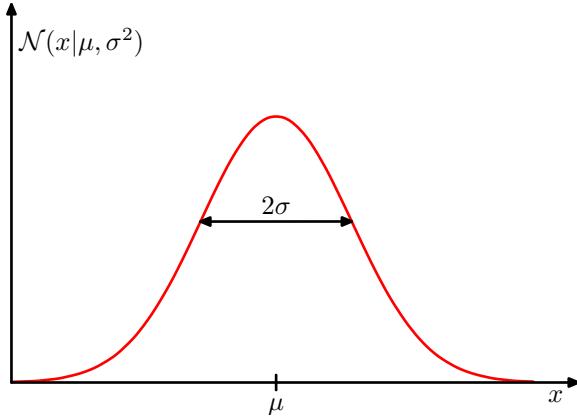


图 1.13: 一元高斯分布的图像，给出了均值 μ 和标准差 σ 。

1.2.4 高斯分布

我们将用整个第二章来研究各种各样的概率分布以及它们的性质。然而，在这里介绍连续变量一种最重要的概率分布是很方便的。这种分布就是正态分布（normal distribution）或者高斯分布（Gaussian distribution）。在其余章节中（事实上在整本书中），我们将会经常用到这种分布。

对于一元实值变量 x ，高斯分布被定义为

$$\mathcal{N}(x | \mu, \sigma^2) = \frac{1}{(2\pi\sigma^2)^{\frac{1}{2}}} \exp\left\{-\frac{1}{2\sigma^2}(x - \mu)^2\right\} \quad (1.46)$$

它由两个参数控制： μ ，被叫做均值（mean），以及 σ^2 ，被叫做方差（variance）。方差的平方根，由 σ 给定，被叫做标准差（standard deviation）。方差的倒数，记作 $\beta = \frac{1}{\sigma^2}$ ，被叫做精度（precision）。我们稍后将看到这些项的意义。图1.13给出了高斯分布的图像。

根据公式 (1.46)，我们看到高斯分布满足

$$\mathcal{N}(x | \mu, \sigma^2) > 0 \quad (1.47)$$

并且很容易证明高斯分布是归一化的，因此

$$\int_{-\infty}^{\infty} \mathcal{N}(x | \mu, \sigma^2) dx = 1 \quad (1.48)$$

因此公式 (1.46) 满足合理的概率密度函数的两个要求。

我们已经能够找到关于 x 的函数在高斯分布下的期望。特别地， x 的平均值为

$$\mathbb{E}[x] = \int_{-\infty}^{\infty} \mathcal{N}(x | \mu, \sigma^2) x dx = \mu \quad (1.49)$$

由于参数 μ 表示在分布下的 x 的平均值，它通常被叫做均值。类似地，二阶矩为

$$\mathbb{E}[x^2] = \int_{-\infty}^{\infty} \mathcal{N}(x | \mu, \sigma^2) x^2 dx = \mu^2 + \sigma^2 \quad (1.50)$$

根据公式 (1.49) 和公式 (1.50)， x 的方差被定义为

$$\text{var}[x] = \mathbb{E}[x^2] - \mathbb{E}[x]^2 = \sigma^2 \quad (1.51)$$

因此 σ^2 也被叫做方差参数。分布的最大值被叫做众数。对于高斯分布，众数与均值恰好相等。

我们也对 D 维向量 x 的高斯分布也感兴趣，定义为

$$\mathcal{N}(x | \mu, \Sigma) = \frac{1}{(2\pi)^{\frac{D}{2}}} \frac{1}{|\Sigma|^{\frac{1}{2}}} \exp\left\{-\frac{1}{2}(x - \mu)^T \Sigma^{-1}(x - \mu)\right\} \quad (1.52)$$

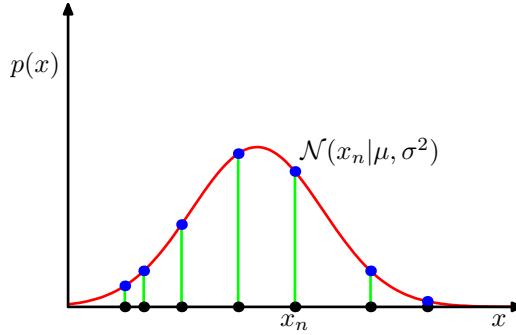


图 1.14: 高斯概率分布的似然函数, 由红色曲线表示。这里, 黑点表示数据集 $\{x_n\}$ 的值, 公式 (1.53) 给出的似然函数对应于蓝色值的乘积。最大化似然函数涉及到调节高斯分布的均值和方差, 使得这个乘积最大。

其中 D 维向量 μ 被称为均值, $D \times D$ 的矩阵 Σ 被称为协方差, $|\Sigma|$ 表示 Σ 的行列式。我们将在本章中简短地使用多变量高斯分布, 详细的性质将在2.3节讨论。

现在假定我们有一个观测的数据集 $\mathbf{x} = (x_1, \dots, x_N)^T$, 表示标量变量 x 的 N 次观测。注意, 我们使用了一个字体不同的 \mathbf{x} 来和向量变量 $(x_1, \dots, x_D)^T$ 作区分, 后者记作 \mathbf{x} 。我们假定各次观测是独立地从高斯分布中抽取的, 分布的均值 μ 和方差 σ^2 未知, 我们想根据数据集来确定这些参数。独立地从相同的数据点中抽取的数据点被称为独立同分布 (independent and identically distributed), 通常缩写成i.i.d.。我们已经看到两个独立事件的联合概率可以由各个事件的边缘概率的乘积得到。由于我们的数据集 \mathbf{x} 是独立同分布的, 因此给定 μ 和 σ^2 , 我们可以给出数据集的概率

$$p(\mathbf{x} | \mu, \sigma^2) = \prod_{n=1}^N \mathcal{N}(x_n | \mu, \sigma^2) \quad (1.53)$$

当我们把它看成 μ 和 σ^2 的时候, 这就是高斯分布的似然函数, 图像如图1.14所示。

使用一个观测数据集来决定概率分布的参数的一个通用的标准是寻找使似然函数取得最大值的参数值。这个标准看起来可能很奇怪, 因为从我们之前对于概率论的讨论来看, 似乎在给定数据集的情况下最大化概率的参数 (而不是在给定参数的情况下最大化数据集出现的概率) 是更加自然的。事实上, 这两个标准是相关的。我们后面将使用曲线拟合的例子来说明这一点。

但是现在, 我们要通过最大化似然函数 (1.53) 来确定高斯分布中未知的参数 μ 和 σ^2 。实际应用中, 考虑似然函数的对数值更方便。由于对数函数是一个单调递增函数, 最大化某个函数的对数等价于最大化这个函数。取对数不仅简化了后续数学分析, 也有助于数值计算, 因为大量小概率的乘积很容易下溢, 这可以通过计算对数概率的和的方式来解决。根据公式 (1.46) 和公式 (1.53), 对数似然函数可以写成

$$\ln p(\mathbf{x} | \mu, \sigma^2) = -\frac{1}{2\sigma^2} \sum_{n=1}^N (x_n - \mu)^2 - \frac{N}{2} \ln \sigma^2 - \frac{N}{2} \ln(2\pi) \quad (1.54)$$

关于 μ , 最大化函数 (1.54), 我们可以得到最大似然解

$$\mu_{ML} = \frac{1}{N} \sum_{n=1}^N x_n \quad (1.55)$$

这是样本均值 (sample mean), 即观测值 $\{x_n\}$ 的均值。类似地, 关于 σ^2 最大化函数 (1.54), 我们得到了方差的最大似然解

$$\sigma_{ML}^2 = \frac{1}{N} \sum_{n=1}^N (x_n - \mu_{ML})^2 \quad (1.56)$$

这是关于样本均值 μ_{ML} 的样本方差 (sample variance)。注意, 我们要同时关于 μ 和 σ^2 来最大化函数 (1.54), 但是在高斯分布的情况下, μ 的解和 σ^2 无关, 因此我们可以首先估计公式 (1.55) 然后使用这个结果来估计公式 (1.56)。

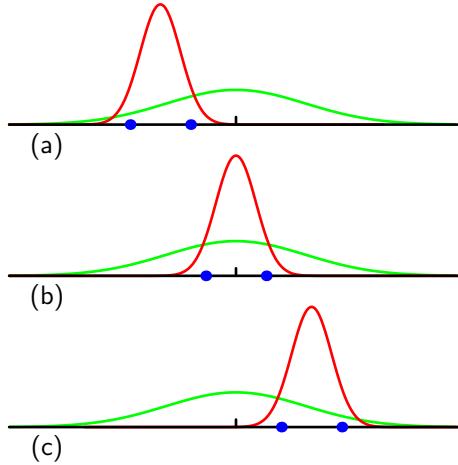


图 1.15: 这幅图说明了使用最大似然方法确定高斯分布的方差时，偏移是如何产生的。绿色曲线表示真实的高斯分布，数据点是根据这个概率分布生成的。三条红色曲线表示对三个数据集拟合得到的高斯概率分布，每个数据集包含两个蓝色的数据点，使用公式 (1.55) 和公式 (1.56) 给出的最大似然的结果进行拟合。对三个数据集求平均，均值是正确的，但是方差被系统性地低估了，因为它是相对样本均值进行测量的，而不是相对真实均值进行测量。

稍后在本章中，以及在后续的章节中，我们要强调最大似然方法的极大的局限性。这里，我们通过考察我们给出的一元高斯分布的最大似然参数解，来稍微说明一下这个问题。特别地，我们会看到，最大似然方法系统化地低估了分布的方差。这是一种叫做偏移 (bias) 的现象的例子，与多项式曲线拟合问题中遇到的过拟合问题相关。我们首先注意到，最大似然解 μ_{ML} 和 σ_{ML}^2 都是数据集 x_1, \dots, x_N 的函数。考虑这些量关于数据集的期望。数据集里面的点来自参数为 μ 和 σ^2 的高斯分布。很容易证明

$$\mathbb{E}[\mu_{ML}] = \mu \quad (1.57)$$

$$\mathbb{E}[\sigma_{ML}^2] = \left(\frac{N-1}{N}\right)\sigma^2 \quad (1.58)$$

因此，最大似然估计的平均值将会得到正确的均值，但是将会低估方差，因子为 $\frac{N-1}{N}$ 。这背后的直觉在图 1.15 中说明。

根据公式 (1.58)，下面的对于方差参数的估计是无偏的。

$$\hat{\sigma}^2 = \frac{N}{N-1}\sigma_{ML}^2 = \frac{1}{N-1}\sum_{n=1}^N(x_n - \mu_{ML})^2 \quad (1.59)$$

注意，当数据点的数量 N 增大时，最大似然解的偏移会变得不太严重，并且在极限 $N \rightarrow \infty$ 的情况下，方差的最大似然解与产生数据的分布的真实方差相等。在实际应用中，只要 N 的值不太小，那么偏移的现象不是个大问题。然而，在本书中，我们感兴趣的是带有很多参数的复杂模型。这些模型中，最大似然的偏移问题会更加严重。实际上，我们会看到，最大似然的偏移问题是我们在多项式曲线拟合问题中遇到的过拟合问题的核心。

1.2.5 重新考察曲线拟合问题

我们已经看到，多项式曲线拟合的问题可以通过误差最小化问题来表示。这里我们回到曲线拟合的问题，从概率的角度来考察它，并且可以更深刻地认识误差函数和正则化，并且能够让我们完全从贝叶斯的角度来看待这个问题。

曲线拟合问题的目标是能够根据 N 个输入 $\mathbf{x} = (x_1, \dots, x_N)^T$ 组成的数据集和它们对应的目标值 $\mathbf{t} = (t_1, \dots, t_N)^T$ ，在给出输入变量 x 的新值的情况下，对目标变量 t 进行预测。我们可以使用

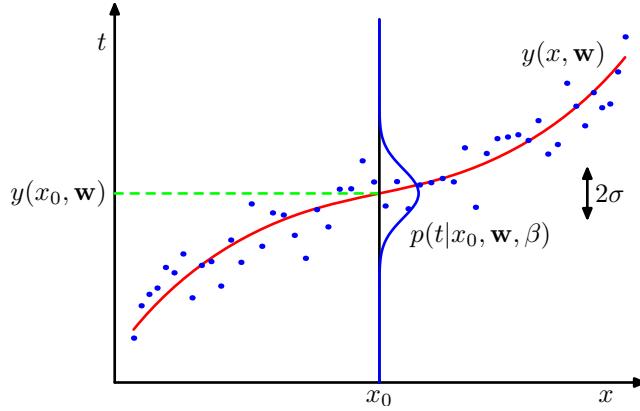


图 1.16: 用图形说明了公式 (1.60) 给出的给定 x 的条件下 t 的高斯条件概率分布, 其中均值为多项式函数 $y(x, \mathbf{w})$, 精度由参数 β 给出, 它与方差的关系为 $\beta^{-1} = \sigma^2$ 。

概率分布来表达关于目标变量的值的不确定性。为了达到这个目的, 我们要假定, 给定 x 的值, 对应的 t 值服从高斯分布, 分布的均值为 $y(x, \mathbf{w})$, 由公式 (1.1) 给出。因此, 我们有

$$p(t | x, \mathbf{w}, \beta) = \mathcal{N}(t | y(x, \mathbf{w}), \beta^{-1}) \quad (1.60)$$

其中, 为了和后续章节中的记号相同, 我们定义了精度参数 β , 它对应于分布方差的倒数。图 1.16 给出了图形化表示。

我们现在用训练数据 $\{\mathbf{x}, \mathbf{t}\}$, 通过最大似然方法, 来决定未知参数 \mathbf{w} 和 β 的值。如果数据假定从分布 (1.60) 中抽取, 那么似然函数为

$$p(\mathbf{t} | \mathbf{x}, \mathbf{w}, \beta) = \prod_{n=1}^N \mathcal{N}(t_n | y(x_n, \mathbf{w}), \beta^{-1}) \quad (1.61)$$

与我们之前处理简单高斯分布时的做法一样, 最大化对数似然函数是很方便的。用公式 (1.46) 给出的高斯分布的形式来替换, 我们可以得到对数似然函数

$$\ln p(\mathbf{t} | \mathbf{x}, \mathbf{w}, \beta) = -\frac{\beta}{2} \sum_{n=1}^N \{y(x_n, \mathbf{w}) - t_n\}^2 + \frac{N}{2} \ln \beta - \frac{N}{2} \ln(2\pi) \quad (1.62)$$

首先考虑确定多项式系数的最大似然解 (记作 \mathbf{w}_{ML})。这些由公式 (1.62) 关于 \mathbf{w} 来确定。为了达到这个目的, 我们可以省略公式 (1.62) 右侧的最后两项, 因为他们不依赖于 \mathbf{w} 。并且, 我们注意到, 使用一个正的常数系数来缩放对数似然函数并不会改变关于 \mathbf{w} 的最大值的位置, 因此我们可以用 $\frac{1}{2}$ 来代替系数 $\frac{\beta}{2}$ 。最后, 我们不去最大化似然函数, 而是等价地去最小化负对数似然函数。于是我们看到, 目前为止对于确定 \mathbf{w} 的问题来说, 最大化似然函数等价于最小化由公式 (1.2) 定义的平方和误差函数。因此, 在高斯噪声的假设下, 平方和误差函数是最大化似然函数的一个自然结果。

我们也可以使用最大似然方法来确定高斯条件分布的精度参数 β 。关于 β 来最大化函数 (1.62), 我们有

$$\frac{1}{\beta_{ML}} = \frac{1}{N} \sum_{n=1}^N \{y(x_n, \mathbf{w}_{ML}) - t_n\}^2 \quad (1.63)$$

我们又一次首先确定控制均值的参数向量 \mathbf{w}_{ML} , 然后使用这个结果来寻找精度 β_{ML} 。这与简单高斯分布时的情形相同。

已经确定了参数 \mathbf{w} 和 β , 我么现在可以对新的 x 的值进行预测。由于我们现在有一个概率模型, 预测可以通过给出 t 的概率分布的预测分布 (predictive distribution) 来表示 (而不仅仅是一个点的估计)。预测分布通过把最大似然参数代入公式 (1.60) 给出。

$$p(t | x, \mathbf{w}_{ML}, \beta_{ML}) = \mathcal{N}(t | y(x, \mathbf{w}_{ML}), \beta_{ML}^{-1}) \quad (1.64)$$

现在让我们朝着贝叶斯的方法前进一步，引入在多项式系数 \mathbf{w} 上的先验分布。简单起见，我们考虑下面形式的高斯分布

$$p(\mathbf{w} | \alpha) = \mathcal{N}(\mathbf{w} | \mathbf{0}, \alpha^{-1} \mathbf{I}) = \left(\frac{\alpha}{2\pi} \right)^{\frac{M+1}{2}} \exp \left\{ -\frac{\alpha}{2} \mathbf{w}^T \mathbf{w} \right\} \quad (1.65)$$

其中 α 是分布的精度， $M + 1$ 是对于 M 阶多项式的向量 \mathbf{w} 的元素的总数。像 α 这样控制模型参数分布的参数，被称为超参数（hyperparameters）。使用贝叶斯定理， \mathbf{w} 的后验概率正比于先验分布和似然函数的乘积。

$$p(\mathbf{w} | \mathbf{x}, \mathbf{t}, \alpha, \beta) \propto p(\mathbf{t} | \mathbf{x}, \mathbf{w}, \beta) p(\mathbf{w} | \alpha) \quad (1.66)$$

给定数据集，我们现在通过寻找最可能的 \mathbf{w} 值（即最大化后验概率）来确定 \mathbf{w} 。这种技术被称为最大后验（maximum posterior），简称MAP。取公式 (1.66) 的负对数，结合公式 (1.62) 和公式 (1.65)，我们可以看到，最大化后验概率就是最小化下式：

$$\frac{\beta}{2} \sum_{n=1}^N \{y(x_n, \mathbf{w}) - t_n\}^2 + \frac{\alpha}{2} \mathbf{w}^T \mathbf{w} \quad (1.67)$$

因此我们看到最大化后验概率等价于最小化正则化的平方和误差函数（之前在公式 (1.4) 中提到），正则化参数为 $\lambda = \frac{\alpha}{\beta}$ 。

1.2.6 贝叶斯曲线拟合

虽然我们已经谈到了先验分布 $p(\mathbf{w} | \alpha)$ ，但是我们目前仍然在进行 \mathbf{w} 的点估计，这并不是贝叶斯观点。在一个纯粹的贝叶斯方法中，我们应该自始至终地应用概率的加和规则和乘积规则。我们稍后会看到，这需要对所有 \mathbf{w} 值进行积分。对于模式识别来说，这种积分是贝叶斯方法的核心。

在曲线拟合问题中，我们知道训练数据 \mathbf{x} 和 \mathbf{t} ，以及一个新的测试点 x ，我们的目标是预测 t 的值。因此我们想估计预测分布 $p(t | x, \mathbf{x}, \mathbf{t})$ 。这里我们要假设参数 α 和 β 是固定的，事先知道的（后续章节中我们会讨论这种参数如何通过贝叶斯方法从数据中推断出来）。

简单地说，贝叶斯方法就是自始至终地使用概率的加和规则和乘积规则。因此预测概率可以写成下面的形式

$$p(t | x, \mathbf{x}, \mathbf{t}) = \int p(t | x, \mathbf{w}) p(\mathbf{w} | \mathbf{x}, \mathbf{t}) d\mathbf{w} \quad (1.68)$$

这里， $p(t | x, \mathbf{w})$ 由公式 (1.60) 给出，并且我们省略了对于 α 和 β 的依赖，简化记号。这里， $p(\mathbf{w} | \mathbf{x}, \mathbf{t})$ 是参数的后验分布，可以通过对公式 (1.66) 归一化得到。我们在3.3节将看到，对于曲线拟合这样的问题，后验分布是一个高斯分布，可以解析地求出。类似地，公式 (1.68) 中的积分也可以解析地求解。因此，预测分布由高斯的形式给出：

$$p(t | x, \mathbf{x}, \mathbf{t}) = \mathcal{N}(t | m(x), s^2(x)) \quad (1.69)$$

其中，均值和方差分别为

$$m(x) = \beta \phi(x)^T \mathbf{S} \sum_{n=1}^N \phi(x_n) t_n \quad (1.70)$$

$$s^2(x) = \beta^{-1} + \phi(x)^T \mathbf{S} \phi(x) \quad (1.71)$$

这里，矩阵 \mathbf{S} 由下式给出

$$\mathbf{S}^{-1} = \alpha \mathbf{I} + \beta \sum_{n=1}^N \phi(x_n) \phi(x_n)^T \quad (1.72)$$

其中， \mathbf{I} 是单位矩阵，向量 $\phi(x)$ 被定义为 $\phi_i(x) = x^i (i = 0, \dots, M)$ 。

我们看到，公式 (1.69) 的预测分布的均值和方差依赖于 x 。公式 (1.71) 的第一项表示预测值 t 的不确定性，这种不确定性由目标变量上的噪声造成。在最大似然的预测分布 (1.64) 中，这种不确定性通过 β_{ML}^{-1} 表达。然而，第二项也对参数 \mathbf{w} 的不确定性有影响。这是贝叶斯方法的结果。图1.17说明了正弦曲线的回归问题。

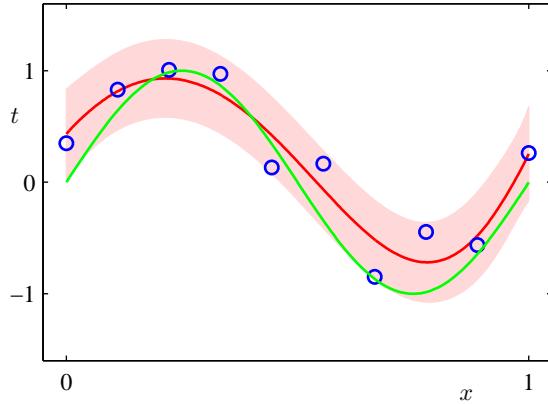


图 1.17: 用贝叶斯方法处理多项式曲线拟合问题得到的预测分布的结果。使用的多项式为 $M = 9$, 超参数被固定为 $\alpha = 5 \times 10^{-3}$ 和 $\beta = 11.1$ (对应于已知的噪声方差)。其中, 红色曲线表示预测概率分布的均值, 红色区域对应于均值周围 ± 1 标准差的范围。

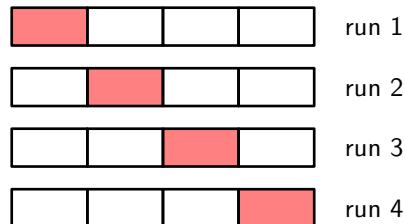


图 1.18: 参数为 S 的交叉验证方法, 这里说明了 $S = 4$ 的情形。以能够得到的数据为输入, 将其划分为 S 组 (最简单的情况下, 等于数据的个数)。然后, $S - 1$ 组数据被用于训练一组模型, 然后在剩余的一组上进行评估。然后对于所有 S 的可能选择重复进行这一步骤, 使用剩余的一组进行评估, 这里用红色标记出来。之后, 对 S 轮运行结果的表现得分求平均值。

1.3 模型选择

在我们使用最小平方拟合多项式曲线的例子中, 我们看到, 存在一个最优的多项式阶数, 能够给出最好的结果。多项式的阶数控制了模型的自由参数的个数, 因此控制了模型的复杂度。通过正则化的最小平方, 正则化系数 λ 也控制了我们的模型复杂度。而对于更复杂的模型, 例如混合分布或者神经网络, 可能存在多个控制模型复杂度的参数。在实际应用中, 我们需要确定这些参数的值, 这么做的主要目的通常是为了在新数据上能做出最好的预测。此外, 除了找到模型中复杂度参数的合适的值之外, 我们可能还希望找到一个可选的模型的范围, 以便能够找到对于特定应用的最好的模型。

我们已经看到, 在最大似然方法中, 由于过拟合现象, 模型在训练集上的表现并不能很好地表示模型对于未知数据的预测能力。如果数据量很大, 那么模型选择很简单。使用一部分可得到的数据, 可以训练出一系列的模型, 也可以得到某个给定模型的一系列复杂度的参数值。之后在独立数据上 (有时被称为验证集) 比较它们, 选择预测表现最好的模型即可。如果模型的设计使用有限规模的数据集迭代很多次, 那么对于验证数据会发生一定程度的过拟合, 因此保留一个第三方的测试集是很有必要的。这个测试集用来最终评估选择的模型的表现。

但是在许多实际应用中, 训练数据和测试数据都是很有限的。为了建立好的模型, 我们想使用尽可能多的可得到的数据进行训练。然而, 如果验证机很小, 它对预测表现的估计就会有一定的噪声。解决这种困境的一种方法是使用交叉验证 (cross validation), 如图 1.18 所示。这种方法能够让可得到数据的 $\frac{S-1}{S}$ 用于训练, 同时使用所有的数据来评估表现。当数据相当稀疏的时候, 考虑 $S = N$ 的情况很合适, 其中 N 是数据点的总数。这种技术叫做“留一法” (leave-one-out)。

交叉验证的一个主要的缺点是需要进行的训练的次数随着 S 而增加, 这对于训练本身很耗时的问题来说是个大问题。对于像交叉验证这种使用分开的数据来评估模型表现的方法来说, 还有一个问题: 对于一个单一的模型, 我们可能有多个复杂度参数 (例如可能有若干个正则化参

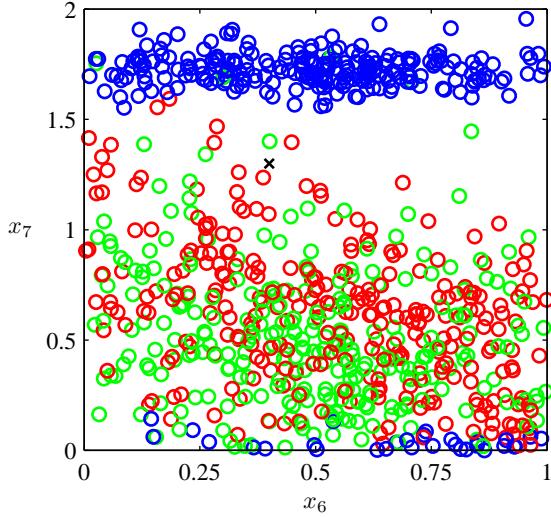


图 1.19: 石油流数据的输入变量 x_6 和 x_7 的散点图, 其中红色表示“同质状”类别, 绿色表示“环状”类别, 蓝色表示“薄片状”类别。我们的目标是分类新的数据点, 记作“ \times ”。

数)。在最坏的情况下, 探索这些参数的组合所需的训练次数可能是参数个数的指数函数。很显然, 我们需要一种更好的方法。理想情况下, 模型的选择应该只依赖于训练数据, 并且应该允许在一轮训练中对比多个超参数以及模型类型。因此我们需要找到一种模型表现的度量, 它只依赖于训练数据, 并且不会由于过拟合产生偏移的问题。

历史上各种各样的“信息准则”被提出来。这些“信息准则”尝试修正最大似然的偏差。修正的方法是增加一个惩罚项来补偿过于复杂的模型造成的过拟合。例如, 赤池信息准则 (Akaike information criterion), 或者简称为AIC (Akaike, 1974), 选择下面使这个量最大的模型:

$$\ln p(\mathcal{D} \mid \mathbf{w}_{ML}) - M \quad (1.73)$$

这里, $p(\mathcal{D} \mid \mathbf{w}_{ML})$ 是最合适的对数似然函数, M 是模型中可调节参数的数量。这个量的一种变体, 被称为贝叶斯信息准则 (Bayesian information criterion), 或者简称为BIC, 将会在4.4.1节讨论。但是, 这种准则没有考虑模型参数的不确定性, 在实际应用中它们倾向于选择过于简单的模型。因此, 我们会在3.4节中讨论完整的贝叶斯方法。我们会看到, 这种方法中, 复杂度的惩罚性是如何自然地得出。

1.4 维度灾难

在多项式曲线拟合的例子中, 我们只有一个输入变量 x 。但是对于模式识别的实际应用来说, 我们不得不处理由许多输入变量组成的高维空间。正如我们现在讨论的那样, 这个问题是个很大的挑战, 也是影响模式识别技术设计的重要因素。

为了说明这个问题, 我们考虑一个人工合成的数据集。这个数据集中的数据表示一个管道中石油、水、天然气各自所占的比例 (Bishop and James, 1993)。这三种物质在管道中的几何形状有三种不同的配置, 被称为“同质状”、“环状”和“薄片状”。三种物质各自的比例也会变化。每个数据点由一个12维的输入向量组成。输入向量是伽马射线密度计的读数, 度量了一窄束伽马射线穿过管道后强度的衰减。数据集的详细描述见附录A。图1.19给出了数据集里的100个点, 每个点只画出了两个分量 x_6 和 x_7 (为了说明的方便, 剩余的10个分量被忽略)。每个数据点根据它属于的三种几何类别之一被标记。我们的目标是使用这个数据作为训练集, 训练一个模型, 能够对于一个新的 (x_6, x_7) 的观测 (例如图1.19中标记为“叉”的点) 进行分类。我们观察到, 标记为“叉”的点周围由许多红色的点, 因此我们可以猜想它属于红色类别。然而, 它附近也有很多绿色的点, 因此我们也可以猜想它属于绿色类别。似乎它不太可能属于蓝色类别。直观看来, 标记为“叉”的点的类别应该与训练集中它附近的点强烈相关, 与距离比较远的点的相关性比较弱。事实上, 这种直观的想法是合理的, 将会在后续章节中详细证明。我们如何把这

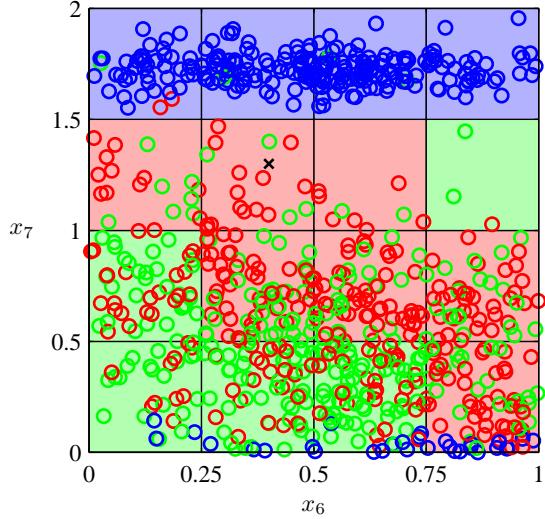


图 1.20: 分类问题的一种简单的解法，其中输入空间被划分为单元格，任何新的测试数据点被划分到同一单元格内具有最多数据的类别。正如我们将看到的那样，这种简单的方法有许多严重的缺点。

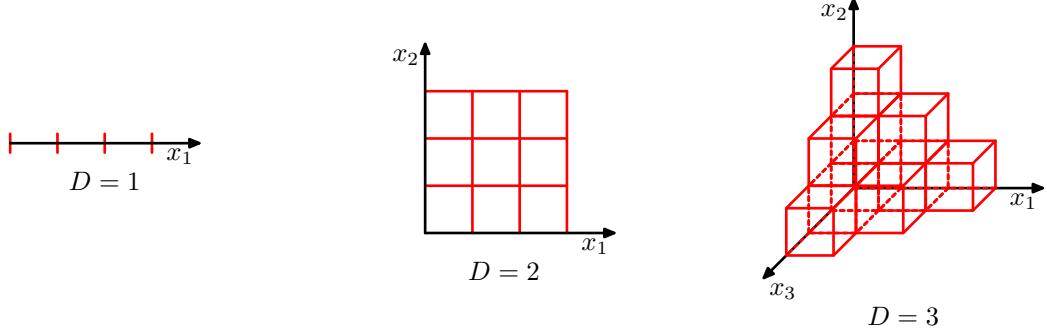


图 1.21: 维度灾难的例子，展示了单元格的数量随着空间的维度 D 指数增长。为了清晰起见， $D = 3$ 的情形中只给出了立方体区域的一个子集。

种直观想法转化为学习算法呢？一种简单的方式是把输入空间划分成小的单元格，如图1.20所示。当给出测试点，我们要预测类别的时候，我们首先判断它属于哪个单元格，然后我们寻找训练集中落在同一个单元格中的训练数据点。测试点的类别就是测试点所在的单元格中数量最多的训练数据点的类别。

这种朴素的观点有很多问题。当需要处理的问题有很多输入数据，并且对应于高维的输入空间时，有一个问题就变得尤为突出。问题的来源如图1.21所示。图1.21表明，如果我们把空间的区域分割成一个个的单元格，那么这些单元格的数量会随着空间的维数以指数的形式增大。当单元格的数量指数增大时，为了保证单元格不为空，我们就不得不需要指数量级的训练数据。很明显，我们只能在变量数量相当少的情况下才能使用这种方法，因此我们需要寻找一些更高级的方法。

我们可以更深刻地讨论一下高维空间中出现的问题。让我们回到多项式拟合的问题，考虑一下我们如何把上面的方法推广到输入空间有多个变量的情形。如果我们有 D 个输入变量，那么一个三阶多项式就可以写成如下的形式

$$y(\mathbf{x}, \mathbf{w}) = w_0 + \sum_{i=1}^D w_i x_i + \sum_{i=1}^D w_{ij} x_i x_j + \sum_{i=1}^D \sum_{j=1}^D \sum_{k=1}^D w_{ijk} x_i x_j x_k \quad (1.74)$$

随着 D 的增加，独立的系数的数量（并非所有的系数都独立，因为变量 x 之间的互换对称性）的增长速度正比于 D^3 。在实际应用中，为了描述数据中复杂的依存关系，我们可能需要使用高阶多项式。对于一个 M 阶多项式，系数数量的增长速度类似于 D^M 。虽然增长速度是一个幂函

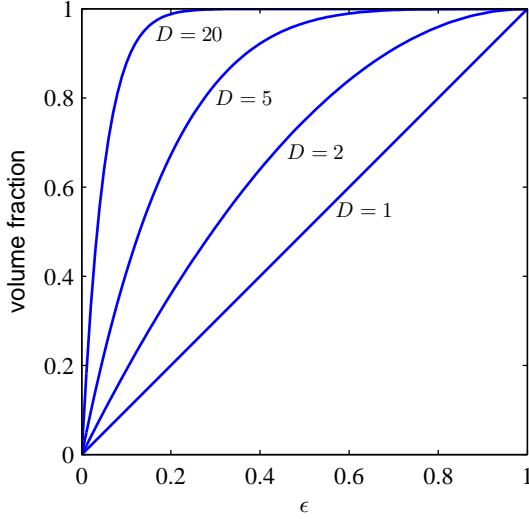


图 1.22: 对于不同的 D , 位于 $r = 1 - \epsilon$ 和 $r = 1$ 之间的部分与球的体积比。

数, 而不是指数函数, 但是这仍然说明了, 这种方法会迅速变得很笨重, 因此在实际应用中很受限。

我们在三维空间中建立的几何直觉会在考虑高维空间时不起作用。例如, 考虑 D 维空间的一个半径 $r = 1$ 的球体, 请问, 位于半径 $r = 1 - \epsilon$ 和半径 $r = 1$ 之间的部分占球的总体积的百分比是多少? 我们注意到, D 维空间的半径为 r 的球体的体积一定是 r^D 的倍数, 因此我们有

$$V_D(r) = K_D r^D \quad (1.75)$$

其中常数 K_D 值依赖于 D 。因此我们要求解的体积比就是

$$\frac{V_D(1) - V_D(1 - \epsilon)}{V_D(1)} = 1 - (1 - \epsilon)^D \quad (1.76)$$

图1.22给出了不同 D 值下, 上式与 ϵ 的关系。我们看到, 对于较大的 D , 这个体积比趋近于1, 即使对于小的 ϵ 也是这样。因此, 在高维空间中, 一个球体的大部分体积都聚集在表面附近的薄球壳上!

再举一个和模式识别直接相关的例子。考虑高维空间的高斯分布的行为。如果我们从笛卡尔坐标系变换到极坐标系, 然后把方向变量积分出来, 我们就得到了一个概率密度的表达式 $p(r)$, 这个表达式是关于距离原点的半径 r 的函数。因此 $p(r)\delta r$ 就是位于半径 r 处厚度为 δr 的薄球壳内部的概率质量。对于不同的 D 值, 这个概率分布的图像如图1.23所示。我们看到, 对于大的 D 值, 高斯分布的概率质量集中在薄球壳处。

高维空间产生的这种困难有时被称为维度灾难 (curse of dimensionality) (Bellman, 1961)。本书中, 我们会频繁使用一维或者二维空间中的例子来说明问题, 因为这使得方法可以很容易地通过图形展示出来。但是读者需要注意, 不是所有在低维空间的直觉都可以推广到高维空间。

虽然维度灾难在模式识别应用中是一个重要的问题, 但是它并不能阻止我们寻找应用于高维空间的有效技术。原因有两方面。第一, 真实的数据经常被限制在有着较低的有效维度的空间区域中, 特别地, 在目标值会发生重要变化的方向上也会有这种限制。第二, 真实数据通常比较光滑 (至少局部上比较光滑), 因此大多数情况下, 对于输入变量的微小改变, 目标值的改变也很小, 因此对于新的输入变量, 我们可以通过局部的类似于插值的技术来进行预测。成功的模式识别技术利用上述的两个性质中的一个, 或者都用。例如, 考虑制造业中的一个应用。这个应用中, 照相机拍摄了传送带上的相同的平面物体, 目标是判断它们的方向。每一张图片都是高维空间中的一个点。高维空间的维数由像素的数量决定。由于物体会出现在图片的不同位置, 并且方向不同, 因此图像之间有3个自由度, 并且一组图片将会处在高维空间的一个三维

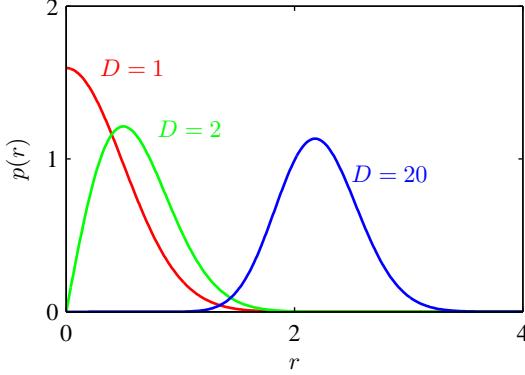


图 1.23: 不同的维度 D 中的高斯分布的概率密度关于半径 r 的关系。在高维空间中，高斯分布的大部分概率质量位于某个半径上的一个薄球壳上。

流形中。由于物体的位置或方向与像素灰度值的关系很复杂，因此流形一定是高度非线性的。如果目标是学习一个模型，这个模型能够以图片作为输入，然后输出物体的方向，与位置无关，那么这个流形中就只有一个自由度了。这很有意义。

1.5 决策论

在1.2节中，我们已经看到了概率论是如何提供给我们一个自始至终的数学框架来量化和计算不确定性。这里我们将要转而讨论决策论。当决策论与概率论结合的时候，我们能够在涉及到不确定性的情况下做出最优的决策。这在模式识别中经常遇到。

假设我们有一个输入向量 x 和对应的目标值向量 t ，我们的目标是对于一个新的 x 值，预测 t 。对于回归问题， t 由连续变量组成，而对于分类问题， t 表示类别标签。联合概率分布 $p(x, t)$ 完整地总结了与这些变量相关的不确定性。从训练数据集中确定 $p(x, t)$ 是推断（inference）问题的一个例子，并且通常是一个非常难的问题。对这种问题的解答是本书大部分内容的主题。但是在一个实际应用中，我们经常必须对 t 的值做出具体的预测，或者更一般地，根据我们对于 t 的可能取值的理解，采取一个具体的动作。这一方面就是决策论的主题。

例如，考虑一个医疗诊断问题。在这个问题中，我们给一个病人拍了X光片，我们想判断病人是否得了癌症。在这种情形下，输入向量 x 是X光片的像素的灰度值集合，输出变量 t 表示病人患有癌症（记作类 C_1 ）或者不患癌症（记作类 C_2 ）。例如，我们可以选择 $t = 0$ 表示类 C_1 ，选择 $t = 1$ 表示类 C_2 。我们稍后会看到，这种标签值的选择对于概率模型特别方便。一般的推断问题就变成了确定联合分布 $p(x, C_k)$ ，或者等价地 $p(x, t)$ 。它给出了最完整的概率描述。虽然这个量很有用，很有信息量，但是最后我们必须确定是否对病人进行治疗，并且我们希望这种选择在某些情况下是最优的（Duda and Hart, 1973）。这是决策步骤，是决策论的主题，告诉我们在给定合适的概率的前提下，如何进行最优的决策。我们会看到，一旦我们解决了推断问题，那么决策阶段通常就变得非常简单，甚至不值得一提。

这里我们简要介绍一下决策论的关键思想，以满足本书剩余部分的要求。更多的背景以及更详细的讨论可以参考 Berger (1985) 和 Bather (2000)。

在给出一个更详细的分析之前，让我们首先非形式化地考虑一下概率论如何在做决策时起作用。当我们得到一个新病人的X光片 x 时，我们的目标是判断这个X光片属于两类中的哪一类。我们感兴趣的是在给定这个图像的前提下，两个类的概率，即 $p(C_k | x)$ 。使用贝叶斯定理，这些概率可以用下面的形式表示

$$p(C_k | x) = \frac{p(x | C_k)p(C_k)}{p(x)} \quad (1.77)$$

注意，出现在贝叶斯定理中的任意一个量都可以从联合分布 $p(x, C_k)$ 中得到，要么通过积分的方式，要么通过关于某个合适的变量求条件概率。我们现在把 $p(C_k)$ 称为类 C_k 的先验概率，把 $p(C_k | x)$ 称为对应的后验概率。因此 $p(C_1)$ 表示在我们拍X光之前，一个人患癌症的概率。类似地， $p(C_1 | x)$ 表示使用X光中包含的信息通过贝叶斯定理修改之后的对应的后验概率。如果我

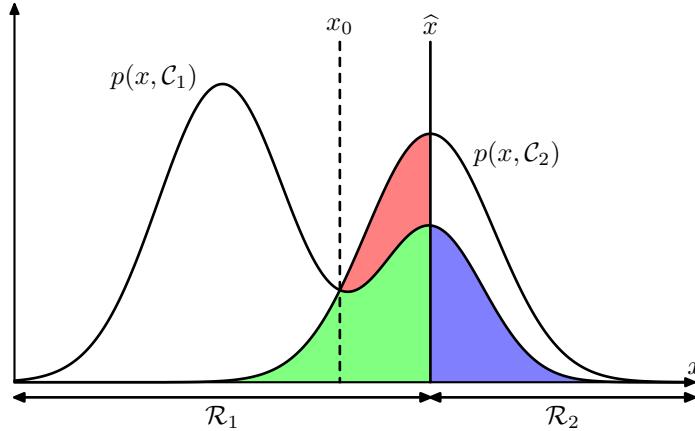


图 1.24: 两个类别的联合概率分布 $p(x, \mathcal{C}_k)$ 与 x 的关系, 以及决策边界 $x = \hat{x}$ 。 $x \geq \hat{x}$ 的值被分类为 \mathcal{C}_2 , 因此属于决策区域 \mathcal{R}_2 , 而 $x < \hat{x}$ 的值被分类为 \mathcal{C}_1 , 属于区域 \mathcal{R}_1 。错误出现在蓝色、绿色和红色区域, 从而对于 $x < \hat{x}$, 错误的来源是将属于类别 \mathcal{C}_2 的点错分到类别 \mathcal{C}_1 (表示为红色区域与绿色区域的总和), 相反对于 $x \geq \hat{x}$ 的点, 错误的来源是将属于类别 \mathcal{C}_1 的点错分到类别 \mathcal{C}_2 (表示为蓝色区域)。当我们改变决策区域的位置 \hat{x} 时, 绿色区域和蓝色区域的总面积是一个常数, 而红色区域的面积发生改变。 \hat{x} 的最优选择是 $p(x, \mathcal{C}_1)$ 的曲线与 $p(x, \mathcal{C}_2)$ 的曲线相交, 对应于 $\hat{x} = x_0$, 因为此时红色区域消失。这等价于最小化错误分类率的决策规则, 这个规则将 x 分配到具有最大的后验概率 $p(\mathcal{C}_k | x)$ 的区域中。

我们的目标是最小化把 x 分到错误类别中的可能性, 那么根据直觉, 我们要选择有最大后验概率的类别。我们现在要证明, 这种直觉是正确的, 并且我们还会讨论进行决策的更加通用的标准。

1.5.1 最小化错误分类率

假定我们的目标很简单, 即尽可能少地作出错误分类。我们需要一个规则来把每个 x 的值分到一个合适的类别。这种规则将会把输入空间切分成不同的区域 \mathcal{R}_k , 这种区域被称为决策区域 (decision region)。每个类别都有一个决策区域, 区域 \mathcal{R}_k 中的所有点都被分到 \mathcal{C}_k 类。决策区域间的边界被叫做决策边界 (decision boundary) 或者决策面 (decision surface)。注意, 每一个决策区域未必是连续的, 可以由若干个分离的区域组成。在后续的章节中, 我们会给出决策边界和决策区域的例子。为了找到最优的决策规则, 首先考虑两类的情形, 就像癌症问题的例子中那样。如果我们把属于 \mathcal{C}_1 类的输入向量分到了 \mathcal{C}_2 类 (或者相反), 那么我们就犯了一个错误。这种事情发生的概率为

$$\begin{aligned} p(\text{mistake}) &= p(\mathbf{x} \in \mathcal{R}_1, \mathcal{C}_2) + p(\mathbf{x} \in \mathcal{R}_2, \mathcal{C}_1) \\ &= \int_{\mathcal{R}_1} p(\mathbf{x}, \mathcal{C}_2) d\mathbf{x} + \int_{\mathcal{R}_2} p(\mathbf{x}, \mathcal{C}_1) d\mathbf{x} \end{aligned} \quad (1.78)$$

我们可以随意选择把点 x 分到两类中的某一类的决策规则。很明显, 为了最小化 $p(\text{mistake})$, 我们对于 x 的分类结果应该让公式 (1.78) 的被积函数尽量小。因此, 如果对于给定的 x 值, 如果 $p(\mathbf{x}, \mathcal{C}_1) > p(\mathbf{x}, \mathcal{C}_2)$, 那么我们就把 x 分到类别 \mathcal{C}_1 中。根据概率的乘积规则, 我们有 $p(\mathbf{x}, \mathcal{C}_k) = p(\mathcal{C}_k | \mathbf{x})p(\mathbf{x})$ 。由于因子 $p(\mathbf{x})$ 对于两项都相同, 因此我们可以这样表述: 如果我们把每个 x 分配到后验概率 $p(\mathcal{C}_k | x)$ 最大的类别中, 那么我们分类错误的概率就会最小。对于一元输入变量 x 的二分类问题, 结果如图1.24所示。

对于更一般的 K 类的情形, 最大化正确率会稍微简单一些, 即最大化下式

$$p(\text{correct}) = \sum_{k=1}^K p(\mathbf{x} \in \mathcal{R}_k, \mathcal{C}_k) = \sum_{k=1}^K \int_{\mathcal{R}_k} p(\mathbf{x}, \mathcal{C}_k) d\mathbf{x} \quad (1.79)$$

当区域 \mathcal{R}_k 的选择使得每个 x 都被分到使 $p(\mathbf{x}, \mathcal{C}_k)$ 最大的类别中时, 上式取得最大值。再一次使用乘积规则 $p(\mathbf{x}, \mathcal{C}_k) = p(\mathcal{C}_k | \mathbf{x})p(\mathbf{x})$, 并且注意到因子 $p(\mathbf{x})$ 对于所有项都相同, 我们可以看到每个 x 都应该被分到有着最大后验概率 $p(\mathcal{C}_k | x)$ 的类别中。

	癌症	正常
癌症	0	1000
正常	1	0

图 1.25: 癌症诊断问题的损失矩阵的例子，矩阵的元素为 L_{kj} 。行对应于真实的类别，而列对应于我们的决策准则做出的分类。

1.5.2 最小化期望损失

对于许多应用，我们的目标要比单纯地最小化错误分类的数量更加复杂。让我们再次考虑医疗诊断的问题。我们注意到，如果已给没有患癌症的病人被错误地诊断为患病，结果可能给病人带来一些压力，并且病人可能需要进一步确诊。相反，如果患癌症的病人被诊断为健康，结果可能会因为缺少治疗而使病人过早死亡。因此这两种错误的结果是相当不同的。很明显，对于第二种错误，我们最好少犯，甚至由于少犯第二种错误会导致第一种错误增加也没关系。

我们可以通过损失函数 (loss function) 来形式化地描述这个问题。损失函数也被称为代价函数 (cost function)，是对于所有可能的决策或者动作可能产生的损失的一种整体的度量。我们的目标是最小化整体的损失。注意，有些学者不考虑损失函数，而是考虑效用函数 (utility function)，并且要最大化这个函数。如果我们让效用函数等于损失函数的相反数的话，那么这些概念是等价的，因此整本书中我们都将使用损失函数这个概念。假设对于新的 \mathbf{x} 的值，真实的类别为 C_k ，我们把 \mathbf{x} 分类为 C_j （其中 j 可能与 k 相等，也可能不相等）。这样做的结果是，我们会造成某种程度的损失，记作 L_{kj} ，它可以看成损失矩阵 (loss matrix) 的第 k, j 个元素。例如，在癌症的例子中，我们可能有图1.25所示的损失矩阵。这个特别的损失矩阵表明，如果我们做出了正确的决策，那么不会造成损失。如果健康人被诊断为患有癌症，那么损失为1。但是如果一个患有癌症的病人被诊断为健康，那么损失为1000。

最优解是使损失函数最小的解。但是，损失函数依赖于真实的类别，这是未知的。对于一个给定的输入向量 \mathbf{x} ，我们对于真实类别的不确定性通过联合概率分布 $p(\mathbf{x}, C_k)$ 表示。因此，我们转而去最小化平均损失。平均损失根据这个联合概率分布计算，定义为

$$\mathbb{E}[L] = \sum_k \sum_j \int_{\mathcal{R}_j} L_{kj} p(\mathbf{x}, C_k) d\mathbf{x} \quad (1.80)$$

每一个 \mathbf{x} 可以被独立地分到决策区域 \mathcal{R}_j 中。我们的目标是选择区域 \mathcal{R}_j ，来最小化期望损失 (1.80)。这表明，对于每个 \mathbf{x} ，我们要最小化 $\sum_k L_{kj} p(\mathbf{x}, C_k)$ 。和之前一样，我们可以使用乘积规则 $p(\mathbf{x}, C_k) = p(C_k | \mathbf{x})p(\mathbf{x})$ 来消除共同因子 $p(\mathbf{x})$ 。因此，最小化期望损失的决策规则是对于每个新的 \mathbf{x} ，把它分到能使下式取得最小值的第 j 类：

$$\sum_k L_{kj} p(C_k | \mathbf{x}) \quad (1.81)$$

一旦我们知道了类的后验概率 $p(C_k | \mathbf{x})$ 之后，这件事就很容易做了。

1.5.3 拒绝选项

我们已经看到，在发生分类错误的输入空间中，后验概率 $p(C_k | \mathbf{x})$ 通常远小于1，或者等价地，不同类别的联合分布 $p(\mathbf{x}, C_k)$ 有着可比的值。这些区域中，类别的归属相对不确定。在某些应用中，对于这种困难的情况，避免做出决策是更合适的选择。这样会使得模型的分类错误率降低。这被称为拒绝选项 (reject option)。例如，在我们假想的医疗例子中，一种合适的做法是，使用自动化的系统来对那些几乎没有疑问的X光片进行分类，然后把不容易分类的X光片留给人类的专家。我们可以用这种方式来达到这个目的：引入一个阈值 θ ，拒绝后验概率 $p(C_k | \mathbf{x})$ 的最大值小于等于 θ 的那些输入 \mathbf{x} 。图1.26说明了一元输入变量 x 的二分类问题的情

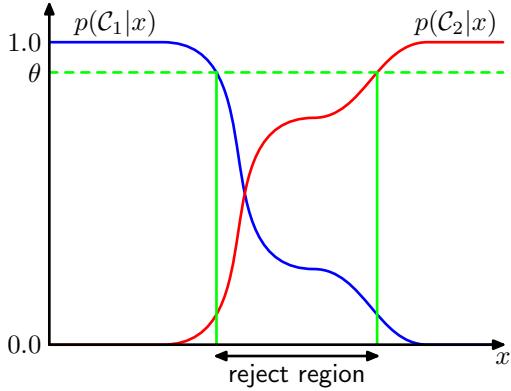


图 1.26: 拒绝选项的例子。如果输入 x 使得两个后验概率分布中较大的那个概率分布小于或等于某个阈值 θ ，那么 x 会被拒绝识别。

形。注意，令 $\theta = 1$ 会使所有的样本都被拒绝，而如果有 K 个类别，那么令 $\theta < \frac{1}{K}$ 将会确保没有样本被拒绝。因此被拒绝的样本比例由 θ 的值控制。

我们可以很容易地把拒绝准则推广到最小化期望损失的情形。那种情形下，我们已知一个损失矩阵，这个矩阵考虑了拒绝决策所带来的损失。

1.5.4 推断和决策

我们已经把分类问题划分成了两个阶段：推断（inference）阶段和决策（decision）阶段。在推断阶段，我们使用训练数据学习 $p(\mathcal{C}_k | \mathbf{x})$ 的模型。在接下来的决策阶段，我们使用这些后验概率来进行最优的分类。另一种可能的方法是，同时解决两个问题，即简单地学习一个函数，将输入 \mathbf{x} 直接映射为决策。这样的函数被称为判别函数（discriminant function）。

事实上，我们可以区分出三种不同的方法来解决决策问题，这三种方法都已经在实际应用问题中被使用。这三种方法按照复杂度降低的顺序给出：

(a) 首先对于每个类别 \mathcal{C}_k ，独立地确定类条件密度 $p(\mathbf{x} | \mathcal{C}_k)$ 。这是一个推断问题。然后，推断先验类概率 $p(\mathcal{C}_k)$ 。之后，使用贝叶斯定理

$$p(\mathcal{C}_k | \mathbf{x}) = \frac{p(\mathbf{x} | \mathcal{C}_k)p(\mathcal{C}_k)}{p(\mathbf{x})} \quad (1.82)$$

求出后验类概率 $p(\mathcal{C}_k | \mathbf{x})$ 。和往常一样，贝叶斯定理的分母可以用分子中出现的项表示，因为

$$p(\mathbf{x}) = \sum_k p(\mathbf{x} | \mathcal{C}_k)p(\mathcal{C}_k) \quad (1.83)$$

等价地，我们可以直接对联合概率分布 $p(\mathbf{x}, \mathcal{C}_k)$ 建模，然后归一化，得到后验概率。得到后验概率之后，我们可以使用决策论来确定每个新的输入 \mathbf{x} 的类别。显式地或者隐式地对输入以及输出进行建模的方法被称为生成式模型（generative model），因为通过取样，可以用来人工生成出输入空间的数据点。

(b) 首先解决确定后验类密度 $p(\mathcal{C}_k | \mathbf{x})$ 这一推断问题，接下来使用决策论来对新的输入 \mathbf{x} 进行分类。这种直接对后验概率建模的方法被称为判别式模型（discriminative models）。

(c) 找到一个函数 $f(\mathbf{x})$ ，被称为判别函数。这个函数把每个输入 \mathbf{x} 直接映射为类别标签。例如，在二分类问题中， $f(\cdot)$ 可能是一个二元的数值， $f = 0$ 表示类别 \mathcal{C}_1 ， $f = 1$ 表示类别 \mathcal{C}_2 。这种情况下，概率不起作用。

让我们考虑一下这三种方法的相对优势。方法(a)需要求解的东西最多，因为它涉及到寻找在 \mathbf{x} 和 \mathcal{C}_k 上的联合概率分布。对于许多应用， \mathbf{x} 的维度很高，这会导致我们需要大量的训练数据才能在合理的精度下确定类条件概率密度。注意，先验概率 $p(\mathcal{C}_k)$ 经常能够根据训练数据集里的每个类别的数据点所占的比例简单地估计出来。但是，方法(a)的一个优点是，它能够通过公式 (1.83) 求出数据的边缘概率密度 $p(\mathbf{x})$ 。这对于检测模型中具有低概率的新数据点很有用，对于

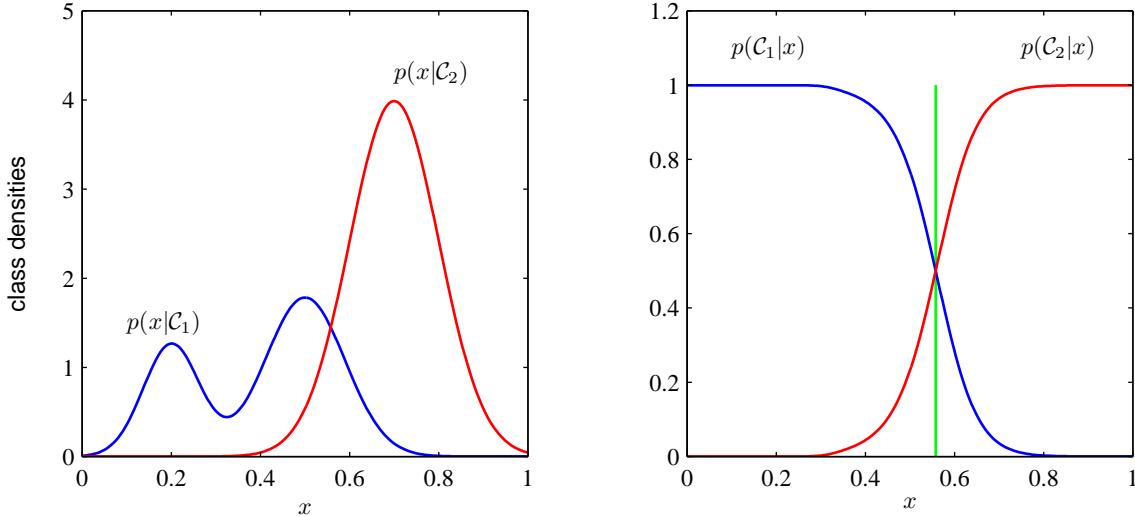


图 1.27: 具有一元输入变量 x 的两个类别的类条件概率密度（左图）以及对应的后验概率密度（右图）。注意，左图中，蓝色曲线表示类条件概率密度 $p(x | C_1)$ ，它的峰值对于后验概率分布没有影响。右图中的垂直绿色直线表示给出最小误分类率的 x 的决策边界。我们假设先验概率分布 $p(C_1)$ 和 $p(C_2)$ 是相等的。

这些点，模型的预测准确率可能会很低。这种技术被称为离群点检测（outlier detection）或者异常检测（novelty detection）（Bishop, 1994; Tarassenko, 1995）。

然而，如果我们只想进行分类的决策，那么这种方法会浪费计算资源。并且，实际上我们只是想求出后验概率 $p(C_k | x)$ （可以直接通过方法(b)求出），但是为了求出它，这种方法需要大量的数据来寻找联合概率 $p(x, C_k)$ 。事实上，类条件密度可能包含很多对于后验概率几乎没有影响的结构，如图1.27所示。关于机器学习中的生成式方法和判别式方法的相对优势，以及如何将两者结合，有很多研究成果（Jebara, 2004; Lasserre et al., 2006）。

一种更简单的方法是方法(c)。这种方法中，我们使用训练数据来寻找将每个 x 直接映射为类别标签的判别函数 $f(x)$ 。这样，我们就把推断阶段和决策阶段结合到一个学习问题中了。在图 1.27 给出的例子中，这对应于绿色竖直线给出的 x 的值，因为这是给出最小错误分类概率的决策边界。

但是，使用方法(c)，我们不在能够接触到后验概率 $p(C_k | x)$ 。有很多强烈的理由需要计算后验概率，即使我们接下来要使用后验概率来进行决策。这些理由包括：

- 最小化风险。考虑这样一个问题，问题中损失矩阵的元素时时刻刻都被修改（例如金融应用中可能出现的情况）。如果我们知道后验概率，我们只需要恰当地修改公式 (1.81) 所定义的最小风险决策准则即可。如果我们只有一个判别准则，那么损失矩阵的任何改变都需要我们返回训练数据，重新解决分类问题。
- 拒绝选项。如果给定被拒绝的数据点所占的比例，后验概率让我们能够确定最小化误分类率的拒绝标准，或者在更一般的情况下确定最小化期望损失的拒绝标准。
- 补偿类先验概率。重新考虑我们的医疗 X 光问题。假定我们已经从普通人群中收集了大量的 X 光片，用作训练数据，用来建立一个自动诊断系统。由于癌症在普通人群中是很少见的，我们可能发现 1000 个样本中只有一个对应癌症。如果我们使用这样的数据集来训练一个模型，由于癌症类别所占的比例很小，我们会遇到很困难的问题。例如，一个将所有的点都判定为正常类别的分类器就已经能够达到 99.9% 的精度。避免这种平凡解是很困难的。并且，即使是一个大的数据集，只有很少的 X 光片对应着癌症，因此学习算法不会接收到很多这种 X 光片，因此不太可能具有很好的泛化性。一个平衡的数据集里，我们已经从每个类别中选择了相等数量的样本，这让我们能够找到一个更加准确的模型。然而，我们之后就必须补偿修改训练数据所造成的影响。假设我们已经使用这种修改后的数据，找到了后验概率的模型。根据公式 (1.82) 的贝叶斯定理，我们看到后验概率正比于先验概

率，而先验概率可以表示为每个类别的数据点所占的比例。因此我们可以把从人造的平衡数据中得到的后验概率除以数据集里的类比例，再乘以我们想要应用模型的目标人群中类别的比例即可。最后，我们需要归一化来保证新的后验概率之和等于1。注意，如果我们直接学习一个判别函数而不确定后验概率，这个步骤就无法进行。

- 组合模型。对于复杂的应用来说，我们可能希望把问题分解成若干个小的子问题，每个子问题都可以通过一个独立的模型解决。例如，在我们假想的医疗诊断问题中，我们可能有来自血液检查的数据，以及X光片。我们不把所有的这种同样类型的信息集中到一个巨大的输入空间中，而是建立一个系统来表示X光片而另一个系统来表示血液数据。这样做效率更高。只要两个模型都给出类别的后验概率，我们就能够使用概率的规则系统化地结合输出。完成这个目标的一个简单的方式是假设对于每个类别，X光片的输入的分布（记作 \mathbf{x}_I ）和血液数据的输入的分布（记作 \mathbf{x}_B ）是独立的，因此

$$p(\mathbf{x}_I, \mathbf{x}_B | \mathcal{C}_k) = p(\mathbf{x}_I | \mathcal{C}_k)p(\mathbf{x}_B | \mathcal{C}_k) \quad (1.84)$$

这是条件独立（conditional independence）的一个例子，因为当分布以类别 \mathcal{C}_k 为条件时满足独立性。同时给出X光片和血液数据，后验概率为

$$\begin{aligned} p(\mathcal{C}_k | \mathbf{x}_I, \mathbf{x}_B) &\propto p(\mathbf{x}_I, \mathbf{x}_B | \mathcal{C}_k)p(\mathcal{C}_k) \\ &\propto p(\mathbf{x}_I | \mathcal{C}_k)p(\mathbf{x}_B | \mathcal{C}_k)p(\mathcal{C}_k) \\ &\propto \frac{p(\mathcal{C}_k | \mathbf{x}_I)p(\mathcal{C}_k | \mathbf{x}_B)}{p(\mathcal{C}_k)} \end{aligned} \quad (1.85)$$

因此我们需要求出类先验概率 $p(\mathcal{C}_k)$ ，这可以通过估计每个类别的数据点所占的比例很容易地得到。之后我们需要对后验概率归一化，使得后验概率之和等于1。公式(1.84)的独立性假设是朴素贝叶斯模型（naive Bayes model）的一个例子。注意，联合边缘分布 $p(\mathbf{x}_I, \mathbf{x}_B)$ 在这个模型下通常不会被分解。在后续章节中，我们会看到如何不依赖公式(1.84)的独立性假设来建立组合数据的模型。

1.5.5 回归问题的损失函数

目前为止，我们以分类问题为例，讨论了决策论。我们现在考虑回归问题，例如之前讨论过的曲线拟合问题。决策阶段包括对于每个输入 \mathbf{x} ，选择一个对于 t 值的具体的估计 $y(\mathbf{x})$ 。假设这样做之后，我们造成了一个损失 $L(t, y(\mathbf{x}))$ 。平均损失（或者说期望损失）就是

$$\mathbb{E}[L] = \iint L(t, y(\mathbf{x}))p(\mathbf{x}, t) \, d\mathbf{x} \, dt \quad (1.86)$$

回归问题中，损失函数的一个通常的选择是平方损失，定义为 $L(t, y(\mathbf{x})) = \{y(\mathbf{x}) - t\}^2$ 。这种情况下，期望损失函数可以写成

$$\mathbb{E}[L] = \iint \{y(\mathbf{x}) - t\}^2 p(\mathbf{x}, t) \, d\mathbf{x} \, dt \quad (1.87)$$

我们的目标是选择 $y(\mathbf{x})$ 来最小化 $\mathbb{E}[L]$ 。如果我们假设一个完全任意的函数 $y(\mathbf{x})$ ，我们能够形式化地使用变分法求解：

$$\frac{\delta \mathbb{E}[L]}{\delta y(\mathbf{x})} = 2 \int \{y(\mathbf{x}) - t\} p(\mathbf{x}, t) \, dt = 0 \quad (1.88)$$

求解 $y(\mathbf{x})$ ，使用概率的加和规则和乘积规则，我们得到

$$y(\mathbf{x}) = \frac{\int t p(\mathbf{x}, t) \, dt}{p(\mathbf{x})} = \int t p(t | \mathbf{x}) \, dt = \mathbb{E}_t[t | \mathbf{x}] \quad (1.89)$$

这是在 \mathbf{x} 的条件下 t 的条件均值，被称为回归函数（regression function）。结果如图1.28所示。这个结果可以扩展到多个目标变量（用向量 t ）的情形。这种情况下，最优解是条件均值 $\mathbf{y}(\mathbf{x}) = \mathbb{E}_t[\mathbf{t} | \mathbf{x}]$ 。

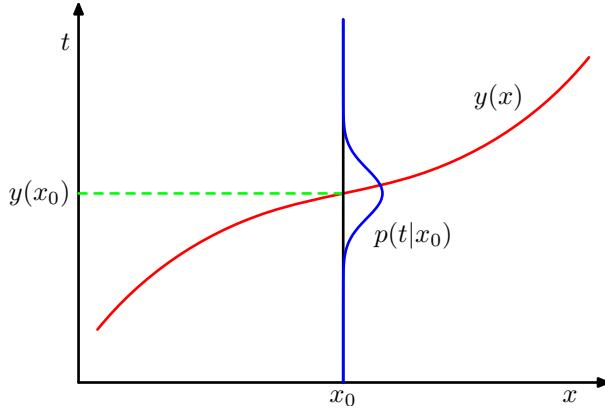


图 1.28: 最小化了期望平方损失的回归函数 $y(x)$ 由条件概率分布 $p(t | x)$ 的均值给出。

我们也可以使用一种稍微不同的方式推导出这个结果，这也将透露出回归问题的本质。已经知道了最优解是条件期望，我们可以把平方项按照下面的方式展开：

$$\begin{aligned}\{y(\mathbf{x}) - t\}^2 &= \{y(\mathbf{x}) - \mathbb{E}[t | \mathbf{x}] + \mathbb{E}[t | \mathbf{x}] - t\}^2 \\ &= \{y(\mathbf{x}) - \mathbb{E}[t | \mathbf{x}]\}^2 + 2\{y(\mathbf{x}) - \mathbb{E}[t | \mathbf{x}]\}\{\mathbb{E}[t | \mathbf{x}] - t\} \\ &\quad + \{\mathbb{E}[t | \mathbf{x}] - t\}^2\end{aligned}$$

其中，为了不让符号过于复杂，我们使用 $\mathbb{E}[t | \mathbf{x}]$ 来表示 $\mathbb{E}_t[t | \mathbf{x}]$ 。代入损失函数中，对 t 进行积分，我们看到交叉项消失，因而得到下面形式的损失函数

$$\mathbb{E}[L] = \int \{y(\mathbf{x}) - \mathbb{E}[t | \mathbf{x}]\}^2 p(\mathbf{x}) d\mathbf{x} + \int \text{var}[t | \mathbf{x}] p(\mathbf{x}) d\mathbf{x} \quad (1.90)$$

我们寻找的函数 $y(x)$ 只出现在第一项中。当 $y(x)$ 等于 $\mathbb{E}[t | \mathbf{x}]$ 时第一项取得最小值，这时第一项会被消去。这正是我们之前推导的结果，表明最优的最小平方预测由条件均值给出。第二项是 t 的分布的方差，在 x 上进行了平均。它表示目标数据内在的变化性，可以被看成噪声。由于它与 $y(x)$ 无关，因此它表示损失函数的不可减小的最小值。

与分类问题相同，我们可以确定合适的概率然后使用这些概率做出最优的决策，或者我们可以建立直接决策的模型。实际上，我们可以区分出三种解决回归问题的方法，按照复杂度降低的顺序，依次为：

- (a) 首先解决确定联合概率密度 $p(\mathbf{x}, t)$ 的推断问题。之后，计算条件概率密度 $p(t | \mathbf{x})$ 。最后，使用公式 (1.89) 积分，求出条件均值。
- (b) 首先解决确定条件概率密度 $p(t | \mathbf{x})$ 的推断问题。之后使用公式 (1.89) 计算条件均值。
- (c) 直接从训练数据中寻找一个回归函数 $y(x)$ 。

这三种方法的相对优势和之前所述的分类问题的情形很相似。

平方损失函数不是回归问题中损失函数的唯一选择。实际上，有些情况下，平方损失函数会导致非常差的结果，这时我们就需要更复杂的方法。这种情况的一个重要的例子就是条件分布 $p(t | \mathbf{x})$ 有多个峰值，这在解决反演问题时经常出现。这里我们简要介绍一下平方损失函数的一种推广，叫做闵可夫斯基损失函数 (Minkowski loss)，它的期望为

$$\mathbb{E}[L_q] = \iint |y(\mathbf{x}) - t|^q p(\mathbf{x}, t) d\mathbf{x} dt \quad (1.91)$$

当 $q = 2$ 时，这个函数就变成了平方损失函数的期望。图1.29给出了不同 q 值下，函数 $|y - t|^q$ 关于 $y - t$ 的图像。当 $q = 2$ 时， $\mathbb{E}[L_q]$ 的最小值是条件均值。当 $q = 1$ 时， $\mathbb{E}[L_q]$ 的最小值是条件中位数。当 $q \rightarrow 0$ 时， $\mathbb{E}[L_q]$ 的最小值是条件众数。

1.6 信息论

从概率论到决策论，本章中我们讨论了一系列的概念。这些概念将会组成本书后续章节中讨论的基础。在本章的最后一节，我们要介绍信息论领域的一些概念。这些概念对于模式识别

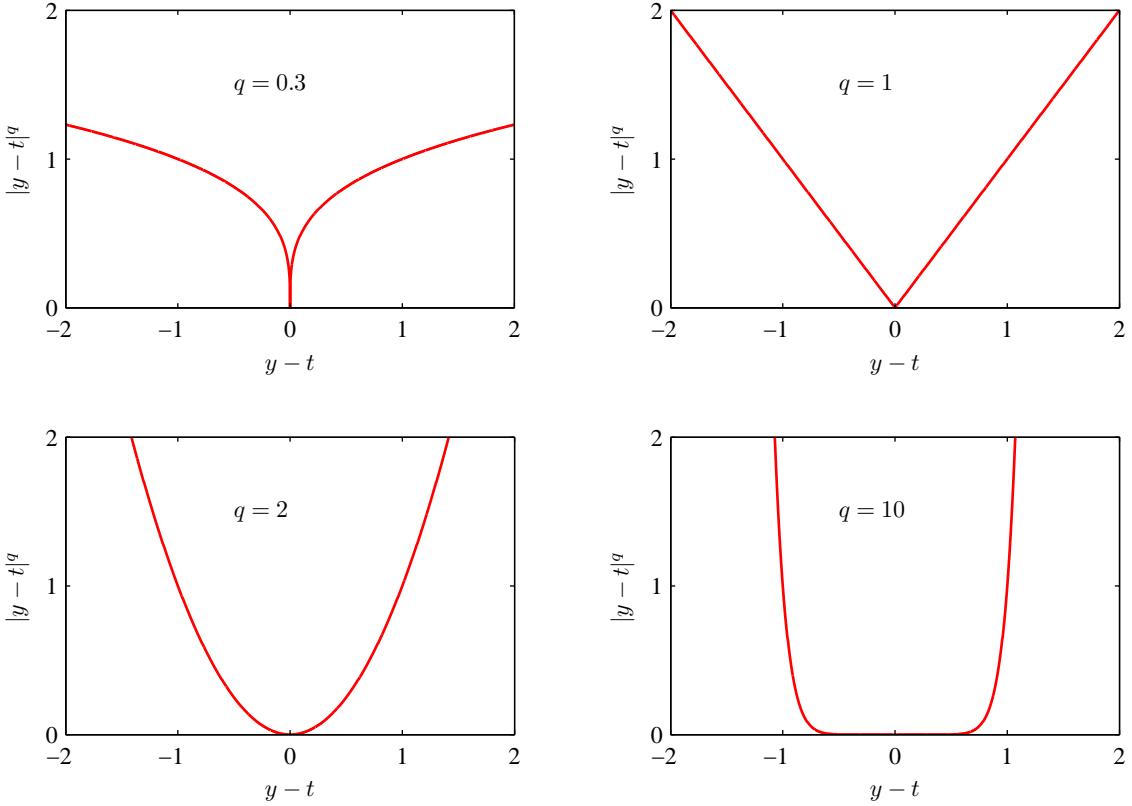


图 1.29: 对于不同的 q 值, $L_q = |y - t|^q$ 的图像。

和机器学习技术的发展也是很有用的。再强调一次, 我们只关注关键的概念。关于更加详细的讨论, 读者可以参考其他资料 (Viterbi and Omura, 1979; Cover and Thomas, 1991; MacKay, 2003)。

首先, 我们考虑一个离散的随机变量 x 。当我们观察到这个变量的一个具体值的时候, 我们接收到了多少信息呢? 信息量可以被看成在学习 x 的值的时候的“惊讶程度”。如果有人告诉我们一个相当不可能的时间发生了, 我们收到的信息要多于我们被告知某个很可能发生的事件发生时收到的信息。如果我们知道某件事情一定会发生, 那么我们就不会接收到信息。于是, 我们对于信息内容的度量将依赖于概率分布 $p(x)$, 因此我们想要寻找一个函数 $h(x)$, 它是概率 $p(x)$ 的单调递增函数, 表达了信息的内容。 $h(\cdot)$ 的形式可以这样寻找: 如果我们有两个不相关的事件 x 和 y , 那么我们观察到两个事件同时发生时获得的信息应该等于观察到事件各自发生时获得的信息之和, 即 $h(x, y) = h(x) + h(y)$ 。两个不相关事件是统计独立的, 因此 $p(x, y) = p(x)p(y)$ 。根据这两个关系, 很容易看出 $h(x)$ 一定与 $p(x)$ 的对数有关。因此, 我们有

$$h(x) = -\log_2 p(x) \quad (1.92)$$

其中, 负号确保了信息一定是正数或者是零。注意, 低概率事件 x 对应于高的信息量。对数的底的选择是任意的。现在我们将遵循信息论的普遍传统, 使用2作为对数的底。在这种情形下, 正如我们稍后会看到的那样, $h(x)$ 的单位是比特 (bit, binary digit)。

现在假设一个发送者想传输一个随机变量的值给接收者。这个过程中, 他们传输的平均信息量可以通过求公式 (1.92) 关于概率分布 $p(x)$ 的期望得到。这个期望值为

$$H[x] = - \sum_x p(x) \log_2 p(x) \quad (1.93)$$

这个重要的量被叫做随机变量 x 的熵 (entropy)。注意, $\lim_{p \rightarrow 0} p \log_2 p = 0$, 因此只要我们遇到一个 x 使得 $p(x) = 0$, 那么我们就应该令 $p(x) \log_2 p(x) = 0$ 。

目前为止，对于公式 (1.92) 的信息的定义以及公式 (1.93) 的熵的定义，我们已经有了一种启发式的动机。我们现在要说明，这些定义确实有着有用的性质。考虑一个随机变量 x 。这个随机变量有8种可能的状态，每个状态都是等可能的。为了把 x 的值传给接收者，我们需要传输一个3比特的消息。注意，这个变量的熵由下式给出

$$H[x] = -8 \times \frac{1}{8} \log_2 \frac{1}{8} = 3 \text{ bits}$$

现在考虑一个具有8种可能状态 $\{a, b, c, d, e, f, g, h\}$ 的随机变量，每个状态各自的概率为 $(\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{64}, \frac{1}{64}, \frac{1}{64}, \frac{1}{64})$ (Cover and Thomas, 1991)。这种情形下的熵为

$$H[x] = -\frac{1}{2} \log_2 \frac{1}{2} - \frac{1}{4} \log_2 \frac{1}{4} - \frac{1}{8} \log_2 \frac{1}{8} - \frac{1}{16} \log_2 \frac{1}{16} - \frac{4}{64} \log_2 \frac{1}{64} = 2 \text{ bits}$$

我们看到，非均匀分布比均匀分布的熵要小。后面当我们根据无序程度来讨论熵的概念时，我们会获得一些更深刻的认识。现在，让我们考虑如何把变量状态的类别传递给接收者。与之前一样，我们可以使用一个3比特的数字来完成这件事情。然而，我们可以利用非均匀分布这个特点，使用更短的编码来描述更可能的事件，使用更长的编码来描述不太可能的事件。我们希望这样做能够得到一个更短的平均编码长度。我们可以使用下面的编码串：0、10、110、1110、111100、111101、111110、111111来表示状态 $\{a, b, c, d, e, f, g, h\}$ 。传输的编码的平均长度就是

$$\text{average code length} = \frac{1}{2} \times 1 + \frac{1}{4} \times 2 + \frac{1}{8} \times 3 + \frac{1}{16} \times 4 + 4 \times \frac{1}{64} \times 6 = 2 \text{ bits}$$

这个值又一次与随机变量的熵相等。注意，我们不能使用更短的编码串，因为必须能够从多个这种字符串的拼接中分割出各个独立的字符串。例如，11001110唯一地编码了状态序列 c, a, d 。

熵和最短编码长度的这种关系是一种普遍的情形。无噪声编码定理 (noiseless coding theorem) (Shannon, 1948) 表明，熵是传输一个随机变量状态值所需的比特位的下界。

现在开始，我们会把熵的定义中的对数变成自然对数，因为这样做会使得熵的概念与本书后续章节中的思想结合起来比较方便。这种情况下，熵的度量的单位是nat，而不是bit。两者的差别是一个 $\ln 2$ 的因子。

我们已经通过具体化随机变量的状态所需的平均信息量介绍了熵的概念。事实上，熵的概念最早起源于物理学，是在热力学平衡的背景中介绍的。后来，熵成为描述统计力学中的无序程度的度量。我们可以这样理解熵的这种含义：考虑一个集合，包含 N 个完全相同的物体，这些物体要被分到若干个箱子中，使得第 i 个箱子中有 n_i 个物体。考虑把物体分配到箱子中的不同方案的数量。有 N 种方式选择第一个物体，有 $(N - 1)$ 种方式选择第二个物体，以此类推。因此总共有 $N!$ 种方式把 N 个物体分配到箱子中，其中 $N!$ 表示乘积 $N \times (N - 1) \times \dots \times 2 \times 1$ 。然而，我们不想区分每个箱子内部物体的重新排列。在第 i 个箱子中，有 $n_i!$ 种方式对物体重新排序，因此把 N 个物体分配到箱子中的总方案数量为

$$W = \frac{N!}{\prod_i n_i!} \quad (1.94)$$

这被称为乘数 (multiplicity)。熵被定义为通过适当的参数放缩后的对数乘数，即

$$H = \frac{1}{N} \ln W = \frac{1}{N} \ln N! - \frac{1}{N} \sum_i \ln n_i! \quad (1.95)$$

我们现在考虑极限 $N \rightarrow \infty$ ，并且保持比值 $\frac{n_i}{N}$ 固定，使用 Stirling 的估计

$$\ln N! \simeq N \ln N - N \quad (1.96)$$

可以得到

$$H = -\lim_{N \rightarrow \infty} \sum_i \left(\frac{n_i}{N} \right) \ln \left(\frac{n_i}{N} \right) = -\sum_i p_i \ln p_i \quad (1.97)$$

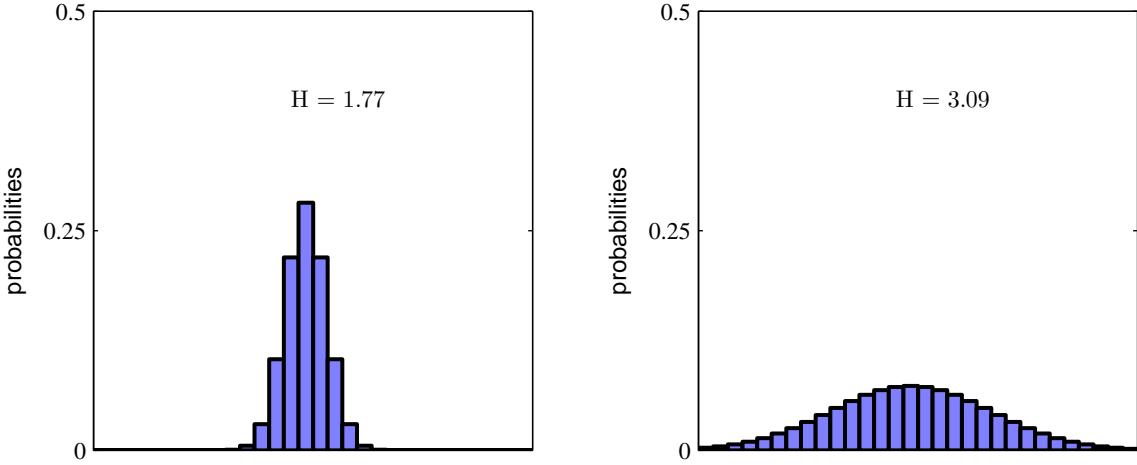


图 1.30: 两个概率分布在30个箱子上的直方图, 表明熵值越大, H 越宽。最大的熵值产生于均匀分布, 此时的熵值为 $H = -\ln(1/30) = 3.40$ 。

推导时我们使用了 $\sum_i n_i = N$ 。这里, $p_i = \lim_{N \rightarrow \infty} (\frac{n_i}{N})$ 是一个物体被分配到第*i*个箱子的概率。使用物理学的术语, 箱子中物体的具体分配方案被称为微观状态 (microstate), 整体的占领数的分布, 表示为比值 $\frac{n_i}{N}$, 被称为宏观状态 (macrostate)。乘数 W 也被称为宏观状态的权重 (weight)。

我们可以把箱子表述成离散随机变量 X 的状态 x_i , 其中 $p(X = x_i) = p_i$ 。这样, 随机变量 X 的熵就是

$$H[p] = - \sum_i p(x_i) \ln p(x_i) \quad (1.98)$$

如果分布 $p(x_i)$ 在几个值周围有尖锐的峰值, 熵就会相对较低。如果分布 $p(x_i)$ 相对平衡地跨过许多值, 那么熵就会相对较高, 如图1.30所示。由于 $0 \leq p_i \leq 1$, 因此熵是非负的。当 $p_i = 1$ 且所有其他的 $p_{j \neq i} = 0$ 时, 熵取得最小值0。在概率归一化的限制下, 使用拉格朗日乘数法可以找到熵的最大值。因此, 我们要最大化

$$\tilde{H} = - \sum_i p(x_i) \ln p(x_i) + \lambda \left(\sum_i p(x_i) - 1 \right) \quad (1.99)$$

可以证明, 当所有的 $p(x_i)$ 都相等, 且值为 $p(x_i) = \frac{1}{M}$ 时, 熵取得最大值。其中, M 是状态 x_i 的总数。此时对应的熵值为 $H = \ln M$ 。这个结果也可以通过Jensen不等式推导出来 (稍后会简短讨论一下)。为了证明驻点确实是最大值, 我们可以求熵的二阶导数, 即

$$\frac{\partial^2 \tilde{H}}{\partial p(x_i) \partial p(x_j)} = -I_{ij} \frac{1}{p_i} \quad (1.100)$$

其中 I_{ij} 是单位矩阵的元素。

我们可以把熵的定义扩展到连续变量 x 的概率分布 $p(x)$, 方法如下。首先把 x 切分成宽度为 Δ 的箱子。然后假设 $p(x)$ 是连续的。均值定理 (mean value theorem) (Weisstein, 1999) 告诉我们, 对于每个这样的箱子, 一定存在一个值 x_i 使得

$$\int_{i\Delta}^{(i+1)\Delta} p(x) dx = p(x_i) \Delta \quad (1.101)$$

我们现在可以这样量化连续变量 x : 只要 x 落在第*i*个箱子中, 我们就把 x 赋值为 x_i 。因此观察到值 x_i 的概率为 $p(x_i)\Delta$ 。这就变成了离散的分布, 这种情形下熵的形式为

$$H_\Delta = - \sum_i p(x_i) \Delta \ln(p(x_i) \Delta) = - \sum_i p(x_i) \Delta \ln p(x_i) - \ln \Delta \quad (1.102)$$

推导时我们使用了 $\sum_i p(x_i)\Delta = 1$, 这可以由公式 (1.101) 得出。我们现在省略公式 (1.102) 右侧的第二项 $-\ln \Delta$, 然后考虑极限 $\Delta \rightarrow 0$ 。在这种极限下, 公式 (1.102) 右侧的第一项就变成了 $p(x) \ln p(x)$ 的积分, 因此

$$\lim_{\Delta \rightarrow 0} \left\{ - \sum_i p(x_i) \Delta \ln p(x_i) \right\} = - \int p(x) \ln p(x) \, dx \quad (1.103)$$

其中, 右侧的量被称为微分熵 (differential entropy)。我们看到, 熵的离散形式与连续形式的差是 $\ln \Delta$, 这在极限 $\Delta \rightarrow 0$ 的情形下发散。这反映出一个事实: 具体化一个连续变量需要大量的比特位。对于定义在多元连续变量 (联合起来记作向量 \mathbf{x}) 上的概率密度, 微分熵为

$$H[\mathbf{x}] = - \int p(\mathbf{x}) \ln p(\mathbf{x}) \, d\mathbf{x} \quad (1.104)$$

在离散分布的情况下, 我们看到最大熵对应于变量的所有可能状态的均匀分布。现在让我们考虑连续变量的最大熵。为了让这个最大值有一个合理的定义, 有必要限制 $p(x)$ 的一阶矩和二阶矩, 同时还要保留归一化的限制。因此我们最大化微分熵的时候要遵循下面三个限制

$$\int_{-\infty}^{\infty} p(x) \, dx = 1 \quad (1.105)$$

$$\int_{-\infty}^{\infty} xp(x) \, dx = \mu \quad (1.106)$$

$$\int_{-\infty}^{\infty} (x - \mu)^2 p(x) \, dx = \sigma^2 \quad (1.107)$$

带有限制条件的最大化问题可以使用拉格朗日乘数法求解, 因此我们要最优化下面的关于 $p(x)$ 的函数

$$\begin{aligned} & - \int_{-\infty}^{\infty} p(x) \ln p(x) \, dx + \lambda_1 \left(\int_{-\infty}^{\infty} p(x) \, dx - 1 \right) \\ & + \lambda_2 \left(\int_{-\infty}^{\infty} xp(x) \, dx - \mu \right) + \lambda_3 \left(\int_{-\infty}^{\infty} (x - \mu)^2 p(x) \, dx - \sigma^2 \right) \end{aligned}$$

使用变分法, 令这个函数的导数等于零, 我们有

$$p(x) = \exp \left\{ -1 + \lambda_1 + \lambda_2 x + \lambda_3 (x - \mu)^2 \right\} \quad (1.108)$$

将这个结果代入三个限制方程中, 即可求出拉格朗日乘数, 最终的结果为

$$p(x) = \frac{1}{(2\pi\sigma^2)^{\frac{1}{2}}} \exp \left\{ -\frac{(x - \mu)^2}{2\sigma^2} \right\} \quad (1.109)$$

因此最大化微分熵的分布是高斯分布。注意, 在最大化熵的时候, 我们没有限制概率分布非负。但是, 由于求出的分布确实是非负的, 我们可以得出结论: 这种限制是不必要的。

如果我们求高斯分布的微分熵, 我们会得到

$$H[\mathbf{x}] = \frac{1}{2} \{ 1 + \ln(2\pi\sigma^2) \} \quad (1.110)$$

因此我们看到熵随着分布宽度 (即 σ^2) 的增加而增加。这个结果也表明, 与离散熵不同, 微分熵可以为负, 因为对于公式 (1.110), 当 $\sigma^2 < \frac{1}{2\pi e}$ 时, $H(x) < 0$ 。

假设我们有一个联合概率分布 $p(\mathbf{x}, \mathbf{y})$ 。我们从这个概率分布中抽取了一对 \mathbf{x} 和 \mathbf{y} 。如果 \mathbf{x} 的值已知, 那么需要确定对应的 \mathbf{y} 值所需的信息就是 $-\ln p(\mathbf{y} | \mathbf{x})$ 。因此, 用来确定 \mathbf{y} 值的平均附加信息可以写成

$$H[\mathbf{y} | \mathbf{x}] = - \iint p(\mathbf{y}, \mathbf{x}) \ln p(\mathbf{y} | \mathbf{x}) \, dy \, dx \quad (1.111)$$

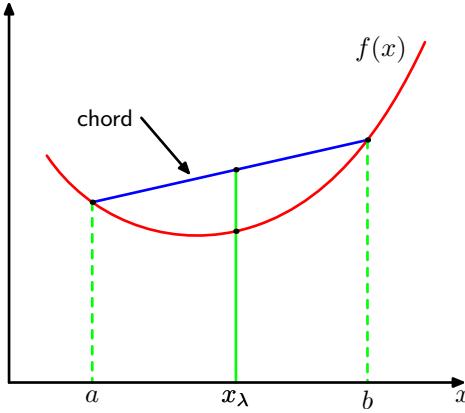


图 1.31: 凸函数 $f(x)$ 的每条弦（蓝色表示）位于函数上或函数上方，函数用红色曲线表示。

这被称为给定 x 的情况下， y 的条件熵。使用乘积规则，很容易看出，条件熵满足下面的关系

$$H[x, y] = H[y | x] + H[x] \quad (1.112)$$

其中， $H[x, y]$ 是 $p(x, y)$ 的微分熵， $H[x]$ 是边缘分布 $p(x)$ 的微分熵。因此，描述 x 和 y 所需的信息是描述 x 自己所需的信息，加上给定 x 的情况下具体化 y 所需的额外信息。

1.6.1 相对熵和互信息

本节目前为止，我们已经介绍了信息论的许多概念，包括熵的关键思想。我们现在开始把这些思想关联到模式识别的问题中。考虑某个未知的分布 $p(x)$ ，假定我们已经使用一个近似的分布 $q(x)$ 对它进行了建模。如果我们使用 $q(x)$ 来建立一个编码体系，用来把 x 的值传给接收者，那么，由于我们使用了 $q(x)$ 而不是真实分布 $p(x)$ ，因此在具体化 x 的值（假定我们选择了一个高效的编码系统）时，我们需要一些附加的信息。我们需要的平均的附加信息量（单位是 nat）为

$$\begin{aligned} \text{KL}(p \| q) &= - \int p(\mathbf{x}) \ln q(\mathbf{x}) d\mathbf{x} - \left(- \int p(\mathbf{x}) \ln p(\mathbf{x}) d\mathbf{x} \right) \\ &= - \int p(\mathbf{x}) \ln \left\{ \frac{q(\mathbf{x})}{p(\mathbf{x})} \right\} d\mathbf{x} \end{aligned} \quad (1.113)$$

这被称为分布 $p(x)$ 和分布 $q(x)$ 之间的相对熵（relative entropy）或者 Kullback-Leibler 散度（Kullback-Leibler divergence），或者 KL 散度（Kullback and Leibler, 1951）。注意这不是一个对称量，即 $\text{KL}(p \| q) \neq \text{KL}(q \| p)$ 。

我们现在要证明，Kullback-Leibler 散度满足 $\text{KL}(p \| q) \geq 0$ ，并且当且仅当 $p(\mathbf{x}) = q(\mathbf{x})$ 时等号成立。为了证明这一点，我们首先介绍凸函数（convex function）的概念。如果一个函数具有如下性质：每条弦都位于函数图像或其上方（如图 1.31 所示），那么我们说这个函数是凸函数。位于 $x = a$ 到 $x = b$ 之间的任何一个 x 值都可以写成 $\lambda a + (1 - \lambda)b$ 的形式，其中 $0 \leq \lambda \leq 1$ 。弦上的对应点可以写成 $\lambda f(a) + (1 - \lambda)f(b)$ ，函数的对应值为 $f(\lambda a + (1 - \lambda)b)$ 。这样，凸函数的性质就可以表示为

$$f(\lambda a + (1 - \lambda)b) \leq \lambda f(a) + (1 - \lambda)f(b) \quad (1.114)$$

这等价于要求函数的二阶导数处处为正。凸函数的例子有 $x \ln x$ ($x > 0$) 和 x^2 。如果等号只在 $\lambda = 0$ 和 $\lambda = 1$ 处取得，我们就说这个函数是严格凸函数（strictly convex function）。如果一个函数具有相反的性质，即每条弦都位于函数图像或其下方，那么这个函数被称为凹函数（concave function）。对应地，也有严格凹函数（strictly concave function）的定义。如果 $f(x)$ 是凸函数，那么 $-f(x)$ 就是凹函数。

使用归纳法，我们可以根据公式 (1.114) 证明凸函数 $f(x)$ 满足

$$f \left(\sum_{i=1}^M \lambda_i x_i \right) \leq \sum_{i=1}^M \lambda_i f(x_i) \quad (1.115)$$

其中，对于任意点集 $\{x_i\}$ ，都有 $\lambda_i \geq 0$ 且 $\sum_i \lambda_i = 1$ 。公式 (1.115) 的结果被称为Jensen不等式 (Jensen's inequality)。如果我们把 λ_i 看成取值为 $\{x_i\}$ 的离散变量 x 的概率分布，那么公式 (1.115) 就可以写成

$$f(\mathbb{E}[x]) \leq \mathbb{E}[f(x)] \quad (1.116)$$

其中， $\mathbb{E}[\cdot]$ 表示期望。对于连续变量，Jensen不等式的形式为

$$f\left(\int x p(x) dx\right) \leq \int f(x) p(x) dx \quad (1.117)$$

我们把公式 (1.117) 形式的Jensen不等式应用于公式 (1.113) 给出的Kullback-Leibler散度，可得

$$\text{KL}(p \parallel q) = - \int p(x) \ln \left\{ \frac{q(x)}{p(x)} \right\} dx \geq - \ln \int q(x) dx = 0 \quad (1.118)$$

推导过程中，我们使用了 $-\ln x$ 是凸函数的事实，以及归一化条件 $\int q(x) dx = 1$ 。实际上， $-\ln x$ 是严格凸函数，因此只有 $q(x) = p(x)$ 对于所有 x 都成立时，等号才成立。因此我们可以把Kullback-Leibler散度看做两个分布 $p(x)$ 和 $q(x)$ 之间不相似程度的度量。

我们看到，在数据压缩和密度估计（即对未知概率分布建模）之间有一种隐含的关系，因为当我们知道真实的概率分布之后，我们可以给出最有效的压缩。如果我们使用了不同于真实分布的概率分布，那么我们一定会损失编码效率，并且在传输时增加的平均额外信息量至少等于两个分布之间的Kullback-Leibler散度。

假设数据通过未知分布 $p(x)$ 生成，我们想要对 $p(x)$ 建模。我们可以试着使用一些参数分布 $q(x | \theta)$ 来近似这个分布。 $q(x | \theta)$ 由可调节的参数 θ 控制（例如一个多元高斯分布）。一种确定 θ 的方式是最小化 $p(x)$ 和 $q(x | \theta)$ 之间关于 θ 的Kullback-Leibler散度。我们不能直接这么做，因为我们不知道 $p(x)$ 。但是，假设我们已经观察到了服从分布 $p(x)$ 的有限数量的训练点 x_n ，其中 $n = 1, \dots, N$ 。那么，关于 $p(x)$ 的期望就可以通过这些点的有限加和，使用公式 (1.35) 来近似，即

$$\text{KL}(p \parallel q) \simeq \frac{1}{N} \sum_{n=1}^N \{-\ln q(x_n | \theta) + \ln p(x_n)\} \quad (1.119)$$

公式 (1.119) 右侧的第二项与 θ 无关，第一项是使用训练集估计的分布 $q(x | \theta)$ 下的 θ 的负对数似然函数。因此我们看到，最小化Kullback-Leibler散度等价于最大化似然函数。

现在考虑由 $p(x, y)$ 给出的两个变量 x 和 y 组成的数据集。如果变量的集合是独立的，那么他们的联合分布可以分解为边缘分布的乘积 $p(x, y) = p(x)p(y)$ 。如果变量不是独立的，那么我们可以通过考察联合概率分布与边缘概率分布乘积之间的Kullback-Leibler散度来判断它们是否“接近”于相互独立。此时，Kullback-Leibler散度为

$$\begin{aligned} I[x, y] &\equiv \text{KL}(p(x, y) \parallel p(x)p(y)) \\ &= - \iint p(x, y) \ln \left(\frac{p(x)p(y)}{p(x, y)} \right) dx dy \end{aligned} \quad (1.120)$$

这被称为变量 x 和变量 y 之间的互信息 (mutual information)。根据Kullback-Leibler散度的性质，我们看到 $I[x, y] \geq 0$ ，当且仅当 x 和 y 相互独立时等号成立。使用概率的加和规则和乘积规则，我们看到互信息和条件熵之间的关系为

$$I[x, y] = H[x] - H[x | y] = H[y] - H[y | x] \quad (1.121)$$

因此我们可以把互信息看成由于知道 y 值而造成的 x 的不确定性的减小（反之亦然）。从贝叶斯的观点来看，我们可以把 $p(x)$ 看成 x 的先验概率分布，把 $p(x | y)$ 看成我们观察到新数据 y 之后的后验概率分布。因此互信息表示一个新的观测 y 造成的 x 的不确定性的减小。

1.7 练习

(1.1) (*) 考虑公式 (1.2) 给出的平方和误差函数，其中函数 $y(x, \mathbf{w})$ 由公式 (1.1) 给出。证明最小化误差函数的系数 $\mathbf{w} = \{w_i\}$ 由下列线性方程的集合给出

$$\sum_{j=0}^M A_{ij} w_j = T_i \quad (1.122)$$

其中

$$A_{ij} = \sum_{n=1}^N (x_n)^{i+j}, \quad T_i = \sum_{n=1}^N (x_n)^i t_n \quad (1.123)$$

这里， i, j 表示元素的下标，而 $(x)^i$ 表示 x 的 i 次幂。

(1.2) (*) 写下能够使由公式 (1.4) 给出的正则化的平方和误差函数取得最小值的系数 w_i 应该满足的与公式 (1.122) 类似的一组线性方程。

(1.3) (***) 假设我们有三个彩色的盒子： r （红色）、 b （蓝色）、 g （绿色）。盒子 r 里有 3 个苹果，4 个橘子，3 个酸橙；盒子 b 里有 1 个苹果，1 个橘子，0 个酸橙；盒子 g 里有 3 个苹果，3 个橘子和 4 个酸橙。如果盒子随机被选中的概率为 $p(r) = 0.2$, $p(b) = 0.2$, $p(g) = 0.6$ 。选择一个水果从盒子中拿走（盒子中选择任何水果的概率都相同），那么选择苹果的概率是多少？如果我们观察到选择的水果实际上是橘子，那么它来自绿色盒子的概率是多少？

(1.4) (**) 考虑一个定义在连续变量 x 上的概率密度 $p_x(x)$ ，假设我们使用 $x = g(y)$ 做了一个非线性变量变换，从而概率密度变换由公式 (1.27) 给出。通过对公式 (1.27) 取微分，请证明，由于 Jacobian 因子的原因， y 的概率密度最大的位置 \hat{y} 与 x 的概率密度最大的位置 \hat{x} 的关系通常不是简单的函数关系 $\hat{x} = g(\hat{y})$ 。这说明概率密度（与简单的函数不同）的最大值取决于变量的选择。请证明，在线性变换的情况下，最大值位置的变换方式与变量本身的变换方式相同。

(1.5) (*) 使用定义 (1.38) 证明 $\text{var}[f(x)]$ 满足公式 (1.39)。

(1.6) (*) 请证明，如果两个变量 x 和 y 是独立的，那么它们的协方差为零。

(1.7) (**) 在本练习中，我们证明公式 (1.48) 给出的一元高斯分布的归一化条件。为了证明这一点，我们考虑下面的积分

$$I = \int_{-\infty}^{\infty} \exp\left(-\frac{1}{2\sigma^2}x^2\right) dx \quad (1.124)$$

这个积分可以这样计算：首先将它的平方写成下面的形式

$$I^2 = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \exp\left(-\frac{1}{2\sigma^2}x^2 - \frac{1}{2\sigma^2}y^2\right) dx dy \quad (1.125)$$

现在使用笛卡尔坐标 (x, y) 到极坐标 (r, θ) 的坐标变换，然后替换 $u = r^2$ 。请证明，通过对 θ 和 u 积分，然后两边取平方根，我们可以得到

$$I = (2\pi\sigma^2)^{\frac{1}{2}} \quad (1.126)$$

最后，使用这个结果，证明高斯分布 $\mathcal{N}(x | \mu, \sigma^2)$ 是归一化的。

(1.8) (**) 通过使用变量替换，证明由公式 (1.46) 给出的一元高斯分布满足公式 (1.49)。接下来，通过对下面的归一化条件

$$\int_{-\infty}^{\infty} \mathcal{N}(x | \mu, \sigma^2) dx = 1 \quad (1.127)$$

两侧关于 σ^2 求微分，证明高斯分布满足公式 (1.50)。最后，证明公式 (1.51) 成立。

(1.9) (*) 证明由公式 (1.46) 给出的高斯分布的众数（即最大值）为 μ 。类似地，证明由公式 (1.52) 给出的多元高斯分布的众数为 μ 。

(1.10) (*) 假设两个变量 x 和 z 是统计独立的。证明它们的和的均值和方差满足

$$\mathbb{E}[x + z] = \mathbb{E}[x] + \mathbb{E}[z] \quad (1.128)$$

$$\text{var}[x + z] = \text{var}[x] + \text{var}[z] \quad (1.129)$$

(1.11) (*) 通过令对数似然函数 (1.54) 关于 μ 和 σ^2 的导数等于零, 证明公式 (1.55) 和公式 (1.56)。

(1.12) (**) 使用公式 (1.49) 和公式 (1.50) 的结果, 证明

$$\mathbb{E}[x_n x_m] = \mu^2 + I_{nm} \sigma^2 \quad (1.130)$$

其中 x_n 和 x_m 表示从均值为 μ 方差为 σ^2 的高斯分布中采样的数据点。当 $n = m$ 时, $I_{nm} = 1$, 否则 $I_{nm} = 0$ 。从而证明了公式 (1.57) 和公式 (1.58) 的结果。

(1.13) (*) 假设高斯分布的方差由公式 (1.56) 进行估计, 但是估计时将均值的最大似然估计 μ_{ML} 替换为真实的均值 μ 。证明, 此时对于方差的估计的期望等于真实的方差。

(1.14) (**) 证明任意的方阵的元素 w_{ij} 都可以写成 $w_{ij} = w_{ij}^S + w_{ij}^A$ 的形式, 其中 w_{ij}^S 和 w_{ij}^A 分别是反对称矩阵和反对称矩阵, 即对于所有的 i 和 j 都有 $w_{ij}^S = w_{ji}^S$ 和 $w_{ij}^A = -w_{ji}^A$ 。现在考虑 D 维空间高阶多项式中的二阶项, 由下式给出

$$\sum_{i=1}^D \sum_{j=1}^D w_{ij} x_i x_j \quad (1.131)$$

证明

$$\sum_{i=1}^D \sum_{j=1}^D w_{ij} x_i x_j = \sum_{i=1}^D \sum_{j=1}^D w_{ij}^S x_i x_j \quad (1.132)$$

从而来自反对称矩阵的贡献消失了。于是, 我们看到, 不失一般性, 系数 w_{ij} 的矩阵可以选择成对称的, 并且这个矩阵中并非所有 D^2 个元素都可以独立选取。证明, 在矩阵 w_{ij}^S 中, 独立参数的个数为 $\frac{D(D+1)}{2}$ 。

(1.15) (***) 在这个练习和下一个练习中, 我们研究多项式函数的独立参数的数量与多项式阶数 M 以及输入空间维度 D 之间的关系。首先, 我们写下 D 维空间多项式的 M 阶项, 形式为

$$\sum_{i_1=1}^D \sum_{i_2=1}^D \cdots \sum_{i_M=1}^D w_{i_1 i_2 \cdots i_M} x_{i_1} x_{i_2} \cdots x_{i_M} \quad (1.133)$$

系数 $w_{i_1 i_2 \cdots i_M}$ 由 D^M 个元素组成, 但是独立参数的数量远小于此, 因为因子 $x_{i_1} x_{i_2} \cdots x_{i_M}$ 有很多互换对称性。首先证明系数的冗余性可以通过把 M 阶项写成下面的形式的方法消除。

$$\sum_{i_1=1}^D \sum_{i_2=1}^{i_1} \cdots \sum_{i_M=1}^{i_{M-1}} \tilde{w}_{i_1 i_2 \cdots i_M} x_{i_1} x_{i_2} \cdots x_{i_M} \quad (1.134)$$

注意, \tilde{w} 系数和 w 系数之间的关系不需要显式表示。使用这个结果证明, M 阶项的独立参数的数量 $n(D, M)$ 满足下面的递归关系

$$n(D, M) = \sum_{i=1}^D n(i, M-1) \quad (1.135)$$

接下来, 使用归纳法证明下面的结果成立

$$\sum_{i=1}^D \frac{(i+M-2)!}{(i-1)!(M-1)!} = \frac{(D+M-1)!}{(D-1)!M!} \quad (1.136)$$

可以这样证明：首先证明 $D = 1$ 的情况下，对于任意的 M ，这个结果成立。证明的过程中会使用 $0! = 1$ 。然后假设这个结论对于 D 维成立，证明它对于 $D + 1$ 维也成立即可。最后，使用之前的两个结果，以及数学归纳法，证明

$$n(D, M) = \frac{(D + M - 1)!}{(D - 1)!M!} \quad (1.137)$$

可以这样证明：首先证明这个结果对于 $M = 2$ 且任意的 $D \geq 1$ 成立，这可以通过对比练习 1.14 的结果得出。然后使用公式 (1.135) 和公式 (1.136)，证明，如果结果对于 $M - 1$ 阶成立，那么它对于 M 阶也成立。

(1.16) (***) 在练习 1.15 中，我们证明了 D 维多项式 M 阶项的独立参数的个数满足公式 (1.135) 给出的关系。我们现在寻找阶数小于等于 M 阶的所有项的独立参数的总数 $N(D, M)$ 。首先，证明 $N(D, M)$ 满足

$$N(D, M) = \sum_{m=0}^M n(D, m) \quad (1.138)$$

其中 $n(D, m)$ 是 m 阶项的独立参数的数量。现在，使用公式 (1.137) 的结果，以及数学归纳法，证明

$$N(D, M) = \frac{(D + M)!}{D!M!} \quad (1.139)$$

可以这样证明：首先证明结果对于 $M = 0$ 以及任意的 $D \geq 1$ 成立，然后假设它对于 M 阶成立，证明它对于 $M + 1$ 阶也成立即可。最后，使用下面的 Stirling 近似

$$n! \simeq n^n e^{-n} \quad (1.140)$$

这个近似关系对于大的 n 成立。证明，对于 $D \gg M$ ， $N(D, M)$ 的增长方式类似于 D^M ，对于 $M \gg D$ ，它的增长方式类似于 M^D 。考虑 D 维的立方 ($M = 3$) 多项式，计算下面两种情形的独立参数的总数：(1) $D = 10$ 和 (2) $D = 100$ ，这对应于典型的小规模和中规模的机器学习应用问题。

(1.17) (**) Gamma 函数的定义为

$$\Gamma(x) \equiv \int_0^\infty u^{x-1} e^{-u} du \quad (1.141)$$

使用分部积分法，证明 $\Gamma(x + 1) = x\Gamma(x)$ 。并且证明， $\Gamma(1) = 1$ ，因此当 x 为整数时， $\Gamma(x + 1) = x!$ 。

(1.18) (**) 我们可以使用公式 (1.126) 的结果来推导 D 为空间中单位半径的球体的表面积 S_D 和体积 V_D 。为了完成这一点，考虑下面的结果。这个结果是通过从笛卡尔坐标系到极坐标系的坐标变换的方式得到的。

$$\prod_{i=1}^D \int_{-\infty}^{\infty} e^{-x_i^2} dx_i = S_D \int_0^\infty e^{-r^2} r^{D-1} dr \quad (1.142)$$

使用 Gamma 函数的定义 (1.141) 以及公式 (1.126)，计算方程的两侧，从而证明

$$S_D = \frac{2\pi^{\frac{D}{2}}}{\Gamma(\frac{D}{2})} \quad (1.143)$$

接下来，通过对半径从 0 到 1 进行积分，证明 D 维单位球体的体积为

$$V_D = \frac{S_D}{D} \quad (1.144)$$

最后，使用结果 $\Gamma(1) = 1$ 和 $\Gamma(\frac{3}{2}) = \frac{\sqrt{\pi}}{2}$ ，证明对于 $D = 2$ 和 $D = 3$ 的情形，公式 (1.143) 和公式 (1.144) 就是通常的结果。

(1.19) (***) 考虑 D 维空间的一个半径为 a 的球体和一个同心的边长为 $2a$ 的超立方体，球面与超立方体的每个面的中心接触。通过使用练习1.18的结果，证明球与超立方体的体积比为

$$\frac{\text{球的体积}}{\text{超立方体的体积}} = \frac{\pi^{\frac{D}{2}}}{D2^{D-1}\Gamma(\frac{D}{2})} \quad (1.145)$$

接下来使用下面形式的Stirling公式

$$\Gamma(x+1) \simeq (2\pi)^{\frac{1}{2}} e^{-x} x^{x+\frac{1}{2}} \quad (1.146)$$

对于 $x \gg 1$ 的情况成立。证明，对于 $D \rightarrow \infty$ ，比值(1.145)趋于零。并且证明，超立方体从中心到某个角的距离与从中心到某条边的垂直距离的比值为 \sqrt{D} ，从而对于 $D \rightarrow \infty$ ，这个比值也趋于 ∞ 。从这些结果中，我们可以看到，在高维空间中，立方体的大部分体积集中在数量众多的角上，这些角本身有着非常长的“尖刺”！

(1.20) (***) 在本练习中，我们研究高维高斯空间的高斯分布的行为。考虑 D 维空间的一个高斯分布，形式如下

$$p(\mathbf{x}) = \frac{1}{(2\pi\sigma^2)^{\frac{D}{2}}} \exp\left(-\frac{\|\mathbf{x}\|^2}{2\sigma^2}\right) \quad (1.147)$$

我们想要找到关于极坐标半径的概率密度，其中方向变量被已经被积分出去。为了完成这一点，证明，概率密度在一个半径为 r 且厚度为 ϵ 的球壳上的积分为 $p(r)\epsilon$ ，其中 $\epsilon \ll 1$ ，且

$$p(r) = \frac{S_D r^{D-1}}{(2\pi\sigma^2)^{\frac{D}{2}}} \exp\left(-\frac{r^2}{2\sigma^2}\right) \quad (1.148)$$

这里， S_D 是 D 维单位球体的表面积。证明，对于大的 D 值，函数 $p(r)$ 有一个驻点位于 $\hat{r} \simeq \sqrt{D}\sigma$ 处。通过考虑 $p(\hat{r} + \epsilon)$ ，其中 $\epsilon \ll \hat{r}$ ，证明对于大的 D 值

$$p(\hat{r} + \epsilon) = p(\hat{r}) \exp\left(-\frac{\epsilon^2}{\sigma^2}\right) \quad (1.149)$$

这表明， \hat{r} 是径向概率密度的最大值点，且远离最大值点 \hat{r} 时， $p(r)$ 会指数衰减，长度缩放因子为 σ 。我们已经看到，对于大的 D 值， $\sigma \ll \hat{r}$ ，因此我们看到大部分的概率质量都集中于大半径的薄球壳上。最后，证明概率密度 $p(\mathbf{x})$ 在原点处的值大于在半径 \hat{r} 处的值，二者的差别是一个值为 $\exp(\frac{D}{2})$ 的因子。于是我们看到，高维高斯分布的概率质量最大的位置不同于半径上概率密度最大的位置。当我们在后续章节中考虑模型参数的贝叶斯推断时，高维空间中的高斯分布的这个性质将会起重要的作用。

(1.21) (***) 考虑两个非负数 a 和 b ，证明，如果 $a \leq b$ ，那么 $a \leq (ab)^{\frac{1}{2}}$ 。使用这个结果证明，如果二分类问题的决策区域被选择为最小化误分类的概率，那么这个概率满足

$$p(\text{误分类}) \leq \int \{p(\mathbf{x}, \mathcal{C}_1)p(\mathbf{x}, \mathcal{C}_2)\}^{\frac{1}{2}} d\mathbf{x} \quad (1.150)$$

(1.22) (*) 给定一个损失矩阵，其元素为 L_{kj} ，如果对于每个 \mathbf{x} ，我们都选择使公式(1.81)取得最小值的类别，那么期望风险会最小。证明，如果损失矩阵为 $L_{kj} = 1 - I_{kj}$ ，其中 I_{kj} 是单位矩阵的元素，那么选择类别的方法就变成了选择具有最大后验概率的类别。这种形式的损失矩阵的意义是什么？

(1.23) (*) 对于一般的损失矩阵和一般的类先验概率，推导最小化期望损失的准则。

(1.24) (***) 考虑一个分类问题。这个问题中，把来自类别 \mathcal{C}_k 的输入向量分类为类别 \mathcal{C}_j 所造成的损失由损失矩阵 L_{kj} 给出。并且，选择拒绝选项所造成的损失为 λ 。找到最小化期望损失的决策准则。证明，当损失矩阵为 $L_{kj} = 1 - I_{kj}$ 时，这个结果就变成了1.5.3节讨论的拒绝准则。 λ 和拒绝阈值 θ 之间的关系是什么？

(1.25) (*) 考虑将一元目标变量 t 的平方和损失函数(1.87)推广到多元目标变量 \mathbf{t} 。推广后的形式为

$$\mathbb{E}[L(\mathbf{t}, \mathbf{y}(\mathbf{x}))] = \iint \|\mathbf{y}(\mathbf{x}) - \mathbf{t}\|^2 p(\mathbf{x}, \mathbf{t}) d\mathbf{x} dt \quad (1.151)$$

	0	1
0	1/3	1/3
1	0	1/3

表 1.3: 练习1.39使用的两个二值变量 x 和 y 的联合概率分布。行表示 x 的值，列表示 y 的值。

使用变分法，证明使得这个期望损失取得最小值的函数 $\mathbf{y}(\mathbf{x})$ 为 $\mathbf{y}(\mathbf{x}) = \mathbb{E}_t[t | \mathbf{x}]$ 。证明，对于一元目标变量 t ，这个结果就变成了公式 (1.89) 给出的结果。

(1.26) (*) 通过将公式 (1.151) 中的平方项展开，推导类似于公式 (1.90) 的结果，证明，对于目标变量组成向量 t 的情形，最小化期望平方损失的函数 $\mathbf{y}(\mathbf{x})$ 仍然是 t 的条件期望。

(1.27) (**) 考虑回归问题的期望损失，损失函数为公式 (1.91) 给出的 L_q 。写出为了最小化 $\mathbb{E}[L_q]$ ， $\mathbf{y}(\mathbf{x})$ 必须满足的条件。证明，对于 $q = 1$ ，这个解表示条件中位数，即函数 $\mathbf{y}(\mathbf{x})$ 使 $t < y(\mathbf{x})$ 的概率质量与 $t \geq y(\mathbf{x})$ 的概率质量相同。并且证明，对于 $q \rightarrow 0$ ，最小的期望 L_q 误差为条件众数，即函数 $\mathbf{y}(\mathbf{x})$ 等于最大化 $p(t | \mathbf{x})$ 的 t 值。

(1.28) (*) 在1.6节，我们介绍了熵 $h(x)$ 的思想，即观察到概率分布为 $p(x)$ 的随机变量 x 的值之后所获得的信息。我们看到，对于独立的变量 x 和 y ，有 $p(x, y) = p(x)p(y)$ ，且熵函数是可加的，即 $h(x, y) = h(x) + h(y)$ 。在这个练习中，我们推导 h 和 p 的函数关系 $h(p)$ 。首先证明 $h(p^2) = 2h(p)$ ，因此通过数学归纳法，有 $h(p^n) = nh(p)$ ，其中 n 是正整数。因此，证明 $h(p^{\frac{n}{m}}) = (\frac{n}{m})h(p)$ ，其中 m 也是一个正整数。这表明 $h(p^x) = xh(p)$ ，其中 x 是一个正有理数，从而根据连续性，这个结果对于 x 是正实数的情形也成立。最后，证明上述结果表明了 $h(p)$ 的形式一定为 $h(p) \propto \ln p$ 。

(1.29) (*) 考虑一个 M 状态的离散随机变量 x ，使用公式 (1.115) 给出的Jensen不等式，证明概率分布 $p(x)$ 的熵满足 $H[x] \leq \ln M$ 。

(1.30) (**) 计算两个高斯分布 $p(x) = \mathcal{N}(x | \mu, \sigma^2)$ 和 $q(x) = \mathcal{N}(x | m, s^2)$ 之间的由公式 (1.113) 给出的Kullback-Leibler散度。

(1.31) (**) 考虑两个变量 x 和 y ，联合概率分布为 $p(\mathbf{x}, \mathbf{y})$ 。证明这对变量的微分熵满足

$$H[\mathbf{x}, \mathbf{y}] \leq H[\mathbf{x}] + H[\mathbf{y}] \quad (1.152)$$

当且仅当 x 和 y 统计独立时等号成立。

(1.32) (*) 考虑一个连续向量 \mathbf{x} ，概率分布为 $p(\mathbf{x})$ ，对应的熵为 $H[\mathbf{x}]$ 。假设我们对 \mathbf{x} 进行了一个非奇异的线性变换，得到一个新的变量 $\mathbf{y} = \mathbf{Ax}$ 。证明对应的熵为 $H[\mathbf{y}] = H[\mathbf{x}] + \ln |\det(\mathbf{A})|$ ，其中 $\det(\mathbf{A})$ 表示 \mathbf{A} 的行列式的值。

(1.33) (**) 假设两个离散随机变量 x 和 y 的条件熵 $H[y | x]$ 为零。证明，对于所有的满足 $p(x) > 0$ 的 x ，变量 y 一定是 x 的函数。换句话说，对于每个 x ，只有一个 y 的值使得 $p(y | x) \neq 0$ 。

(1.34) (**) 使用变分法证明公式 (1.108) 之前的泛函的驻点由公式 (1.108) 给出。然后使用限制条件 (1.105)、(1.106) 和 (1.107)，消去拉格朗日乘数，从而证明最大熵的解由高斯分布 (1.109) 给出。

(1.35) (*) 使用公式 (1.106) 和公式 (1.107) 的结果，证明一元高斯分布 (1.109) 的熵为 (1.110)。

(1.36) (*) 一个严格凸函数的定义为：每条弦都位于函数图像上方的函数。证明，这等价于函数的二阶导数为正。

(1.37) (*) 使用定义 (1.111) 以及概率的乘积规则，证明公式 (1.112) 的结果。

(1.38) (**) 使用归纳法，证明从凸函数的不等式 (1.114) 可以推导出公式 (1.115)。

(1.39) (****) 考虑两个变量 x 和 y ，每个变量只有两个可能的取值。它们的联合概率分布在表1.3中给出。计算下面各式的值，画一个图说明这些量之间的关系。

$$\begin{array}{lll} H[\mathbf{x}] & H[y | \mathbf{x}] & H[\mathbf{x}, \mathbf{y}] \\ H[\mathbf{y}] & H[\mathbf{x} | \mathbf{y}] & I[\mathbf{x}, \mathbf{y}] \end{array}$$

(1.40) (*) 使用Jensen不等式 (1.115) , 其中 $f(x) = \ln x$, 证明一组实数的算术平均值永远不小于它们的几何平均值。

(1.41) (*) 使用概率的加和规则和乘积规则, 证明互信息 $I(\mathbf{x}, \mathbf{y})$ 满足关系 (1.121) 。

2 概率分布

在第一章中，我们强调了概率论在解决模式识别问题时的重要作用。我们现在探究一下某些特殊的概率分布的例子以及它们的性质。这些概率分布本身吸引了很多人的兴趣，也是构成更复杂模型的基石。我们将在整本书中频繁使用这些概率分布。本章中介绍的概率分布也有一个重要的目的，即让我们有机会在简单的模型中讨论一些关键的统计学概念，例如贝叶斯推断。我们在后续章节中会在更复杂的模型里遇到这些简单的模型。

本章中讨论的概率分布的一个作用是在给定有限次观测 x_1, \dots, x_N 的前提下，对随机变量 x 的概率分布 $p(x)$ 建模。这个问题被称为密度估计 (density estimation)。本章中，我们会假定数据点是独立同分布的。应该强调的是，密度估计问题本质上是病态的，因为产生有限的观测数据集的概率分布有无限多种。实际上，任何在数据点 x_1, \dots, x_N 处概率非零的概率分布 $p(x)$ 都是一个潜在的候选。选择一个合适的分布与模型选择的问题相关，这个我们已经在第一章中针对多项式曲线拟合问题讨论过了。这是模式识别领域的一个中心问题。

首先，我们考虑离散随机变量的二项分布和多项式分布，以及连续随机变量的高斯分布。这是参数分布 (parametric distribution) 的具体的例子。之所以被称为参数分布，是因为少量可调节的参数控制了整个概率分布。为了把这种模型应用到密度估计问题中，我们需要一个步骤，能够在给定观察数据集的条件下，确定参数的合适的值。在频率学家的观点中，我们通过最优化某些准则（例如似然函数）来确定参数的具体值。相反，在贝叶斯观点中，给定观察数据，我们引入参数的先验分布，然后使用贝叶斯定理来计算对应后验概率分布。

我们会看到，共轭先验 (conjugate prior) 有着很重要的作用。它使得后验概率分布的函数形式与先验概率相同，因此使得贝叶斯分析得到了极大的简化。例如，多项式分布的参数的共轭先验被叫做狄利克雷分布 (Dirichlet distribution)，而高斯分布的均值的共轭先验是另一个高斯分布。所有这些分布都是指数族 (exponential family) 分布的特例。指数族分布有很多重要的性质，将在本章中详细讨论。

参数方法的一个限制是它假定分布有一个具体的函数形式，这对于一个具体应用来说是不合适的。另一种替代的方法是非参数 (nonparametric) 密度估计方法。这种方法中分布的形式通常依赖于数据集的规模。这些模型仍然具有参数，但是这些参数控制的是模型的复杂度而不是分布的形式。本章最后，我们会考虑三种非参数化方法，分布依赖于直方图、最近邻以及核函数。

2.1 二元变量

首先，我们考虑一个二元随机变量 $x \in \{0, 1\}$ 。例如， x 可能描述了扔硬币的结果， $x = 1$ 表示“正面”， $x = 0$ 表示反面。我们可以假设由一个损坏的硬币，这枚硬币正面朝上的概率未必等于反面朝上的概率。 $x = 1$ 的概率被记作参数 μ ，因此

$$p(x = 1 | \mu) = \mu \quad (2.1)$$

其中 $0 \leq \mu \leq 1$ 。我们可以看到， $p(x = 0 | \mu) = 1 - \mu$ 。 x 的概率分布因此可以写成

$$\text{Bern}(x | \mu) = \mu^x (1 - \mu)^{1-x} \quad (2.2)$$

这被叫做伯努利分布 (Bernoulli distribution)。很容易证明，这个分布是归一化的，并且均值和方差为

$$\mathbb{E}[x] = \mu \quad (2.3)$$

$$\text{var}[x] = \mu(1 - \mu) \quad (2.4)$$

现在我们假设我们有一个 x 的观测值的数据集 $\mathcal{D} = \{x_1, \dots, x_N\}$ 。假设每次观测都是独立地从 $p(x | \mu)$ 中抽取的，因此我们可以构造关于 μ 的似然函数如下

$$p(\mathcal{D} | \mu) = \prod_{n=1}^N p(x_n | \mu) = \prod_{n=1}^N \mu^{x_n} (1 - \mu)^{1-x_n} \quad (2.5)$$

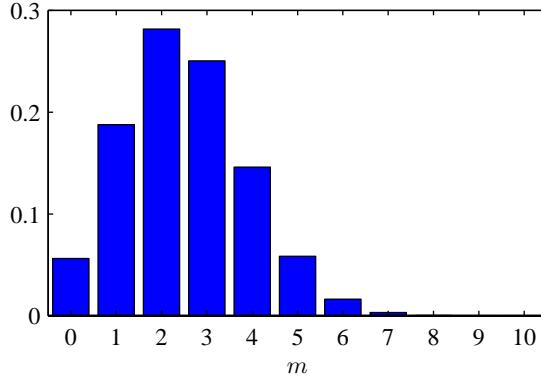


图 2.1: 二项分布 (2.9) 关于 m 的函数的直方图, 其中 $N = 10$ 且 $\mu = 0.25$ 。

在频率学家的观点看来, 我们可以通过最大化似然函数来估计 μ 的值, 或者等价地, 最大化对数似然函数。在伯努利分布的情形下, 对数似然函数为

$$\ln p(\mathcal{D} \mid \mu) = \sum_{n=1}^N \ln p(x_n \mid \mu) = \sum_{n=1}^N \{x_n \ln \mu + (1 - x_n) \ln(1 - \mu)\} \quad (2.6)$$

在这种观点中, 值得注意的一点是对数似然函数只通过和式 $\sum_n x_n$ 依赖于 x_n 的 N 次观察。这个和式是这个分布下数据的充分统计量 (sufficient statistic), 我们后面将详细研究充分统计量的重要作用。如果我们令 $\ln p(\mathcal{D} \mid \mu)$ 关于 μ 的导数等于零, 我们就得到了最大似然的估计值

$$\mu_{ML} = \frac{1}{N} \sum_{n=1}^N x_n \quad (2.7)$$

这也被称为样本均值 (sample mean)。如果我们把数据集里 $x = 1$ (正面朝上) 的观测的数量记作 m , 那么我们可以把公式 (2.7) 写成下面的形式

$$\mu_{ML} = \frac{m}{N} \quad (2.8)$$

因此在最大似然的框架中, 正面朝上的概率是数据集里正面向上的观测所占的比例。

现在假设我们扔一个硬币 3 次, 碰巧 3 次都是正面朝上。那么 $N = m = 3$, 且 $\mu_{ML} = 1$ 。这种情况下, 最大似然的结果会预测所有未来的观测值都是正面向上。常识告诉我们这个是不合理的。事实上, 这是最大似然中过拟合现象的一个极端例子。我们稍后会看到, 通过引入 μ 的先验分布, 我们会得到一个更合理的结论。

我们也可以求解给定数据集规模 N 的条件下, $x = 1$ 的观测出现的数量 m 的概率分布。这被称为二项分布 (binomial distribution)。根据公式 (2.5) 可以看到, 这个概率正比于 $\mu^m (1 - \mu)^{N-m}$ 。为了得到归一化系数, 我们注意到, 在 N 次抛掷中, 我们必须把所有获得 m 个正面朝上的方式都加起来, 因此二项分布可以写成

$$\text{Bin}(m \mid N, \mu) = \binom{N}{m} \mu^m (1 - \mu)^{N-m} \quad (2.9)$$

其中

$$\binom{N}{m} \equiv \frac{N!}{(N - m)! m!} \quad (2.10)$$

是从总数为 N 的完全相同的物体中选择 m 个物体的方式的总数。图 2.1 给出了 $N = 10$ 且 $\mu = 0.25$ 情况下的二项分布示意图。

二项分布的均值和方差可以使用练习1.10的结果得到。练习1.10的结果表明，对于独立的事件，加和的均值等于均值的加和，加和的方差等于方差的加和。由于 $m = x_1 + \dots + x_N$ ，并且对于每次观察，均值和方差都分别由公式 (2.3) 和公式 (2.4) 给出，因此我们有

$$\mathbb{E}[m] \equiv \sum_{m=0}^N m \text{Bin}(m | N, \mu) = N\mu \quad (2.11)$$

$$\text{var}[m] \equiv \sum_{m=0}^N (m - \mathbb{E}[m])^2 \text{Bin}(m | N, \mu) = N\mu(1 - \mu) \quad (2.12)$$

这些结果也可以直接使用微积分的方法得到。

2.1.1 Beta分布

根据公式 (2.8)，我们已经看到伯努利分布的参数 μ 的最大似然解，因此在二项分布中，这个最大似然解也是数据集里 $x = 1$ 的观测所占的比例。正如我们已经提到过的那样，这对于小规模的数据集会给出严重的过拟合结果。为了用贝叶斯的观点看待这个问题，我们需要引入一个关于 μ 的先验概率分布 $p(\mu)$ 。这里，我们考虑一种形式简单的先验分布。这种形式简单的先验分布有很多有用的性质。为了找到这个先验分布，我们注意到似然函数是某个因子与 $\mu^x(1 - \mu)^{1-x}$ 的乘积的形式。如果我们选择一个正比于 μ 和 $(1 - \mu)$ 的幂指数的先验概率分布，那么后验概率分布（正比于先验和似然函数的乘积）就会有着与先验分布相同的函数形式。这个性质被叫做共轭性（conjugacy），我们在本章的后续部分将看到几个这样的例子。因此，我们把先验分布选择为Beta分布，定义为

$$\text{Beta}(\mu | a, b) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \mu^{a-1} (1-\mu)^{b-1} \quad (2.13)$$

其中， $\Gamma(x)$ 是由公式 (1.141) 定义的Gamma函数，公式 (2.13) 保证了Beta分布式归一化的，即

$$\int_0^1 \text{Beta}(\mu | a, b) d\mu = 1 \quad (2.14)$$

Beta分布的均值和方差为

$$\mathbb{E}[\mu] = \frac{a}{a+b} \quad (2.15)$$

$$\text{var}[\mu] = \frac{ab}{(a+b)^2(a+b+1)} \quad (2.16)$$

参数 a 和 b 经常被称为超参数（hyperparameter），因为它们控制了参数 μ 的概率分布。图2.2给出了不同的超参数值对应的Beta分布的图像。

μ 的后验概率分布现在可以这样得到：把Beta先验 (2.13) 与二项似然函数 (2.9) 相乘，然后归一化。只保留依赖于 μ 的因子，我们看到后验概率分布的形式为

$$p(\mu | m, l, a, b) \propto \mu^{m+a-1} (1-\mu)^{l+b-1} \quad (2.17)$$

其中 $l = N - m$ ，即对应于硬币“反面朝上”的样本数量。我们看到公式 (2.17) 关于 μ 的函数形式与先验分布相同，这反映出先验关于似然函数的共轭性质。实际上，它仅仅是另一个Beta分布。通过与公式 (2.13) 对比，我们可以得到它的归一化系数。因此

$$p(\mu | m, l, a, b) = \frac{\Gamma(m+a+l+b)}{\Gamma(m+a)\Gamma(l+b)} \mu^{m+a-1} (1-\mu)^{l+b-1} \quad (2.18)$$

我们看到，如果一个数据集里有 m 次观测为 $x = 1$ ，有 l 次观测为 $x = 0$ ，那么从先验概率到后验概率， a 的值变大了 m ， b 的值变大了 l 。这让我们可以简单地把先验概率中的超参数 a 和 b 分别看成 $x = 1$ 和 $x = 0$ 的有效观测数（effective number of observation）。注意， a 和 b 不一定是整数。

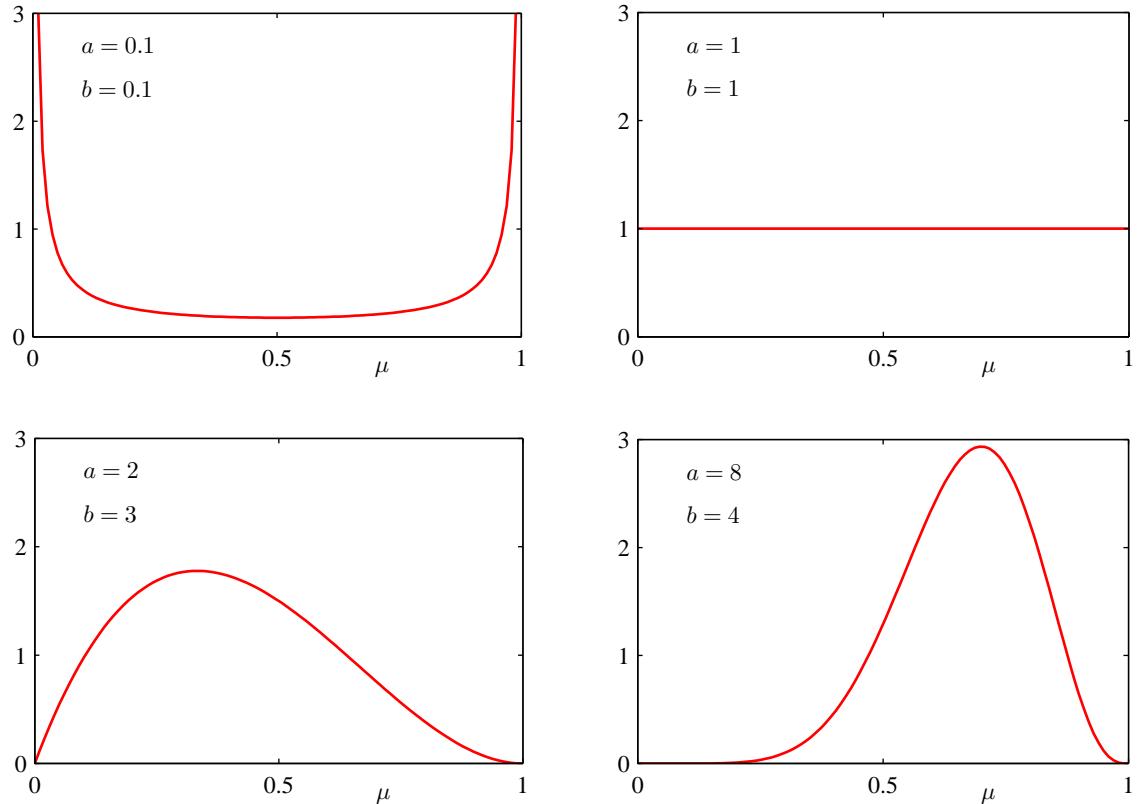


图 2.2: 对于不同的超参数 a 和 b , 公式 (2.13) 给出的Beta分布 $\text{Beta}(\mu | a, b)$ 关于 μ 的函数图像。

另外, 如果我们接下来观测到更多的数据, 那么后验概率分布可以扮演先验概率的角色。为了说明这一点, 我们可以假想每次值取一个观测值, 然后在每次观测之后更新当前的后验分布。更新方法是让当前的后验分布与新观测值的似然函数相乘, 然后归一化, 获得新的修正后的后验分布。在每个阶段, 后验概率是一个Beta分布, 对于 $x = 1$ 和 $x = 0$ 的观测总数(先验的和实际的)由参数 a 和 b 给出。观测到一个 $x = 1$ 仅仅对应于把 a 的值增加1, 而观测到 $x = 0$ 会使 b 增加1。图2.3说明了这个过程中的一个步骤。

我们看到, 如果我们接受了贝叶斯观点, 那么学习过程中的顺序 (sequential) 方法可以自然而然地得出。它与先验和似然函数的选择无关, 只取决于数据独立同分布的假设。顺序方法每次使用一个观测值, 或者每次使用一小批观测值, 然后在使用下一个观测值之前丢掉它们。例如, 顺序方法可以被用于实时学习的场景中。在实时学习的场景中, 输入为一个稳定持续的数据流, 模型必须在观测到所有数据之前就进行预测。由于顺序学习的方法不需要把所有的数据都存储到内存里, 因此顺序方法对于大的数据集也很有用。最大似然方法也可以转化成顺序的

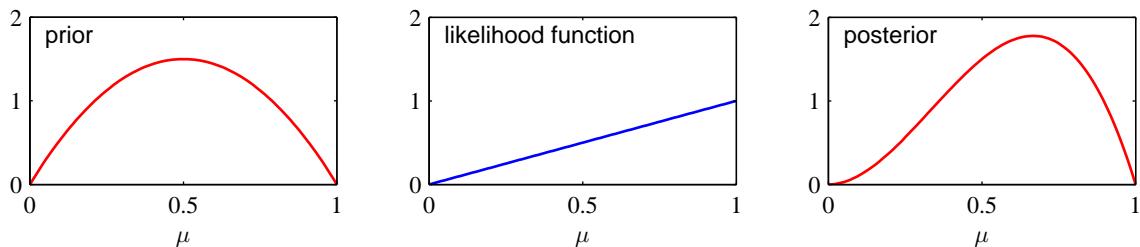


图 2.3: 贝叶斯顺序推断中的一个步骤的例子。先验概率为Beta分布, 参数为 $a = 2, b = 2$, 似然函数由公式 (2.9) 给出, 其中 $N = m = 1$, 对应于 $x = 1$ 的一次观测, 从而后验概率分布为Beta分布, 参数为 $a = 3, b = 2$ 。

框架。

如果我们的目标是尽可能好地预测下一次试验的输出，那么我们必须估计给定观测数据集 \mathcal{D} 的情况下， x 的预测分布。根据概率的加和规则和乘积规则，这个预测分布的形式为

$$p(x = 1 | \mathcal{D}) = \int_0^1 p(x = 1 | \mu)p(\mu | \mathcal{D}) d\mu = \int_0^1 \mu p(\mu | \mathcal{D}) d\mu = \mathbb{E}[\mu | \mathcal{D}] \quad (2.19)$$

使用公式 (2.18) 的结果，后验分布 $p(\mu | \mathcal{D})$ 以及Beta分布的均值的结果 (2.15)，我们可以得到

$$p(x = 1 | \mathcal{D}) = \frac{m + a}{m + a + l + b} \quad (2.20)$$

这个结果可以简单地表述为对应于 $x = 1$ 的观测结果（包括实际的观测值和假想的先验观测值）所占的比例。注意，在数据集无限大的极限情况下， $m, l \rightarrow \infty$ ，此时公式 (2.20) 的结果变成了最大似然的结果 (2.8)。正如我们将看到的那样，贝叶斯的结果和最大似然的结果在数据集的规模趋于无穷的情况下会统一到一起。这是一个很普遍的情况。对于有限规模的数据集， μ 的后验均值总是位于先验均值和公式 (2.7) 给出的 μ 的最大似然估计之间。

从图2.2中，我们可以看到，当观测的数量增加时，后验分布的图像变得更尖了。这通过公式 (2.16) 给出的Beta分布方差的结果也能够看出来。在公式 (2.16) 中，如果 $a \rightarrow \infty$ 或者 $b \rightarrow \infty$ ，那么方差就趋于零。实际上，我们可能想知道，下面这个性质是不是贝叶斯学习的一个共有的属性：随着我们观测到越来越多的数据，后验概率表示的不确定性将会持续下降。

为了说明这一点，我们可以用频率学家的观点考虑贝叶斯学习问题。我们可以证明，平均来看，这种性质确实成立。考虑一个一般的贝叶斯推断问题，参数为 θ ，并且我们观测到了一个数据集 \mathcal{D} ，由联合概率分布 $p(\theta, \mathcal{D})$ 描述。下面的结果

$$\mathbb{E}_{\theta}[\theta] = \mathbb{E}_{\mathcal{D}}[\mathbb{E}_{\theta}[\theta | \mathcal{D}]] \quad (2.21)$$

其中

$$\mathbb{E}_{\theta}[\theta] \equiv \int p(\theta)\theta d\theta \quad (2.22)$$

$$\mathbb{E}_{\mathcal{D}}[\mathbb{E}_{\theta}[\theta | \mathcal{D}]] \equiv \int \left\{ \int \theta p(\theta | \mathcal{D}) d\theta \right\} p(\mathcal{D}) d\mathcal{D} \quad (2.23)$$

表明， θ 的后验均值，在产生数据集的整个分布上面做平均，等于 θ 的先验均值。类似地，我们可以证明

$$\text{var}_{\theta}[\theta] = \mathbb{E}_{\mathcal{D}}[\text{var}_{\theta}[\theta | \mathcal{D}]] + \text{var}_{\mathcal{D}}[\mathbb{E}_{\theta}[\theta | \mathcal{D}]] \quad (2.24)$$

公式 (2.24) 左侧的项是 θ 的先验方差。在右侧，第一项是 θ 的平均后验方差，第二项是 θ 的后验均值的方差。由于这个方差是一个整数，因此这个结果表明，平均来看， θ 的后验方差小于先验方差。后验均值的方差越大，这个方差的减小就越大。但是需要注意的是，这个结果只在平均情况下成立，对于一个特定的观测数据集，有可能后验方差大于先验方差。

2.2 多项式变量

二元变量可以用来描述只能取两种可能值中的某一种这样的量。然而，我们经常会遇到可以取 K 个互斥状态中的某一种的离散变量。虽然有多种方式来表达这种变量，但是我们稍后会看到，一种比较方便的表示方法是“1-of- K ”表示法。这种表示方法中，变量被表示成一个 K 维向量 \mathbf{x} ，向量中的一个元素 x_k 等于1，剩余的元素等于0。例如，如果我们有一个能够取 $K = 6$ 种状态的变量，这个变量的某次特定的观测恰好对应于 $x_3 = 1$ 的状态，那么 \mathbf{x} 就可以表示为

$$\mathbf{x} = (0, 0, 1, 0, 0, 0)^T \quad (2.25)$$

注意，这样的向量满足 $\sum_{k=1}^K x_k = 1$ 。如果我们用参数 μ_k 表示 $x_k = 1$ 的概率，那么 \mathbf{x} 的分布就是

$$p(\mathbf{x} | \boldsymbol{\mu}) = \prod_{k=1}^K \mu_k^{x_k} \quad (2.26)$$

其中 $\mu = (\mu_1, \dots, \mu_K)^T$, 参数 μ_k 要满足 $\mu_k \geq 0$ 和 $\sum_k \mu_k = 1$, 因为它们表示概率。概率分布 (2.26) 可以被看成伯努利分布对于多个输出的一个推广。很容易看出, 这个分布是归一化的

$$\sum_{\mathbf{x}} p(\mathbf{x} | \boldsymbol{\mu}) = \sum_{k=1}^K \mu_k = 1 \quad (2.27)$$

并且

$$\mathbb{E}[\mathbf{x} | \boldsymbol{\mu}] = \sum_{\mathbf{x}} p(\mathbf{x} | \boldsymbol{\mu}) \mathbf{x} = (\mu_1, \dots, \mu_K)^T = \boldsymbol{\mu} \quad (2.28)$$

现在考虑一个有 N 个独立观测值 x_1, \dots, x_N 的数据集 \mathcal{D} 。对应的似然函数的形式为

$$p(\mathcal{D} | \boldsymbol{\mu}) = \prod_{n=1}^N \prod_{k=1}^K \mu_k^{x_{nk}} = \prod_{k=1}^K \mu_k^{(\sum_n x_{nk})} = \prod_{k=1}^K \mu_k^{m_k} \quad (2.29)$$

我们看到似然函数对于 N 个数据点的依赖只是通过 K 个下面形式的量

$$m_k = \sum_n x_{nk} \quad (2.30)$$

它表示观测到 $x_k = 1$ 的次数。这被称为这个分布的充分统计量 (sufficient statistics)。

为了找到 $\boldsymbol{\mu}$ 的最大似然解, 我们需要关于 μ_k 最大化 $\ln p(\mathcal{D} | \boldsymbol{\mu})$, 并且要限制 μ_k 的和必须等于 1。这可以通过拉格朗日乘数 λ 实现, 即最大化

$$\sum_{k=1}^K m_k \ln \mu_k + \lambda \left(\sum_{k=1}^K \mu_k - 1 \right) \quad (2.31)$$

令公式 (2.31) 关于 μ_k 的导数等于 0, 我们有

$$\mu_k = -\frac{m_k}{\lambda} \quad (2.32)$$

我们可以把公式 (2.32) 的结果代入到限制条件 $\sum_k \mu_k = 1$ 中, 解得 $\lambda = -N$ 。因此我们得到了最大似然解

$$\mu_k^{ML} = \frac{m_k}{N} \quad (2.33)$$

它是 N 次观测中, $x_k = 1$ 的观测所占的比例。

我们可以考虑 m_1, \dots, m_K 在参数 $\boldsymbol{\mu}$ 和 观测总数 N 条件下的联合分布。根据公式 (2.29), 这个分布的形式为

$$\text{Mult}(m_1, m_2, \dots, m_K | \boldsymbol{\mu}, N) = \binom{N}{m_1 m_2 \dots m_K} \prod_{k=1}^K \mu_k^{m_k} \quad (2.34)$$

这被称为多项式分布 (multinomial distribution)。归一化系数是把 N 个物体分成大小为 m_1, \dots, m_K 的 K 组的方案总数, 定义为

$$\binom{N}{m_1 m_2 \dots m_K} = \frac{N!}{m_1! m_2! \dots m_K!} \quad (2.35)$$

注意, m_k 满足下面的限制

$$\sum_{k=1}^K m_k = N \quad (2.36)$$



图 2.4: 三个变量 μ_1, μ_2, μ_3 上的狄利克雷分布被限制在一个单纯形中, 如图所示。这是由于限制条件 $0 \leq \mu_k \leq 1$ 和 $\sum_k \mu_k = 1$ 的存在所造成的。



图 2.5: 三个变量上的狄利克雷分布的图像, 其中两个水平轴是单纯形平面上的坐标轴, 垂直轴对应于概率密度的值。这里 $\{\alpha_k\} = 0.1$ 对应于左图, $\{\alpha_k\} = 1$ 对应于中图, $\{\alpha_k\} = 10$ 对应于右图。

2.2.1 狄利克雷分布

现在我们介绍多项式分布 (2.34) 的参数 $\{\mu_k\}$ 的一组先验分布。通过观察多项式分布的形式, 我们看到, 共轭先验为

$$p(\boldsymbol{\mu} | \boldsymbol{\alpha}) \propto \prod_{k=1}^K \mu_k^{\alpha_k - 1} \quad (2.37)$$

其中 $0 \leq \mu_k \leq 1$ 且 $\sum_k \mu_k = 1$ 。这里, $\alpha_1, \dots, \alpha_K$ 是分布的参数, $\boldsymbol{\alpha}$ 表示 $(\alpha_1, \dots, \alpha_K)^T$ 。注意, 由于加和的限制, $\{\mu_k\}$ 空间上的分布被限制在 $K - 1$ 维的单纯形 (simplex) 当中。图 2.4 给出了 $K = 3$ 的情形。

概率的归一化形式为

$$\text{Dir}(\boldsymbol{\mu} | \boldsymbol{\alpha}) = \frac{\Gamma(\alpha_0)}{\Gamma(\alpha_1) \cdots \Gamma(\alpha_K)} \prod_{k=1}^K \mu_k^{\alpha_k - 1} \quad (2.38)$$

这被称为狄利克雷分布 (Dirichlet distribution)。这里 $\Gamma(x)$ 是公式 (1.141) 定义的 Gamma 函数, 而

$$\alpha_0 = \sum_{k=1}^K \alpha_k \quad (2.39)$$

图 2.5 给出了在不同的参数 α_k 的情况下, 单纯形上的狄利克雷分布的图像。

用似然函数 (2.34) 乘以先验 (2.38), 我们得到了参数 $\{\mu_k\}$ 的后验分布, 形式为

$$p(\boldsymbol{\mu} | \mathcal{D}, \boldsymbol{\alpha}) \propto p(\mathcal{D} | \boldsymbol{\mu}) p(\boldsymbol{\mu} | \boldsymbol{\alpha}) \propto \prod_{k=1}^K \mu_k^{\alpha_k + m_k - 1} \quad (2.40)$$

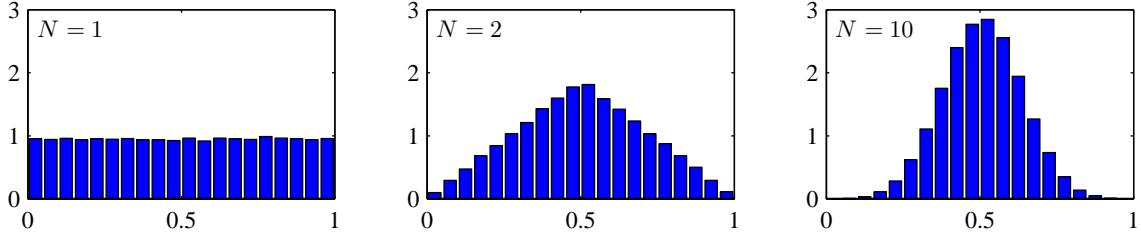


图 2.6: 对于不同的 N 值, N 个均匀分布的均值的直方图。我们观察到, 随着 N 的增加, 分布趋向于高斯分布。

我们看到后验分布的形式又变成了狄利克雷分布, 这说明, 狄利克雷分布确实是多项式分布的共轭先验。这让我们确定能够通过与公式 (2.38) 比较, 确定归一化系数。因此

$$\begin{aligned} p(\boldsymbol{\mu} \mid \mathcal{D}, \boldsymbol{\alpha}) &= \text{Dir}(\boldsymbol{\mu} \mid \boldsymbol{\alpha} + \mathbf{m}) \\ &= \frac{\Gamma(\alpha_0 + N)}{\Gamma(\alpha_1 + m_1) \cdots \Gamma(\alpha_K + m_K)} \prod_{k=1}^K \mu_k^{\alpha_k + m_k - 1} \end{aligned} \quad (2.41)$$

其中, $\mathbf{m} = (m_1, \dots, m_K)^T$ 。与二项分布的先验概率为Beta分布相同, 我们可以把狄利克雷分布的参数 α_k 看成 $x_k = 1$ 的有效观测数。

需要主要的是, 具有两个状态的量既可以表示为二元变量然后使用公式 (2.9) 的二项分布建模, 也可以表示为“1-of-2”的变量然后使用公式 (2.34) 的多项式分布建模。

2.3 高斯分布

高斯分布, 也被称为正态分布, 广泛应用于连续型随机变量分布的模型中。对于一元变量 x 的情形, 高斯分布可以写成下面的形式

$$\mathcal{N}(x \mid \mu, \sigma^2) = \frac{1}{(2\pi\sigma^2)^{\frac{1}{2}}} \exp \left\{ -\frac{1}{2\sigma^2}(x - \mu)^2 \right\} \quad (2.42)$$

其中 μ 是均值, σ^2 是方差。对于 D 维向量 \mathbf{x} , 多元高斯分布的形式为

$$\mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{\frac{D}{2}}} \frac{1}{|\boldsymbol{\Sigma}|^{\frac{1}{2}}} \exp \left\{ -\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}) \right\} \quad (2.43)$$

其中, $\boldsymbol{\mu}$ 是一个 D 维均值向量, $\boldsymbol{\Sigma}$ 是一个 $D \times D$ 的协方差矩阵, $|\boldsymbol{\Sigma}|$ 是 $\boldsymbol{\Sigma}$ 的行列式。

高斯分布会在许多不同的问题中产生, 可以从多个不同的角度来理解。例如, 我们已经看到, 对于一个一元实值向量, 使熵取得最大值的是高斯分布。这个性质对于多元高斯也成立。

当我们考虑多个随机变量之和的时候, 也会产生高斯分布。拉普拉斯提出的中心极限定理 (central limit theorem) 告诉我们, 对于某些温和的情况, 一组随机变量之和 (当然也是随机变量) 的概率分布随着和式中项的数量的增加而逐渐趋向高斯分布 (Walker, 1969)。考虑 N 个变量 x_1, \dots, x_N , 每一个都是区间 $[0, 1]$ 上的均匀分布, 然后考虑均值 $\frac{1}{N}(x_1 + \dots + x_N)$ 的分布。对于大的 N , 这个分布趋向于高斯分布, 如图 2.6 所示。在实际应用中, 随着 N 的增加, 分布会很迅速收敛为高斯分布。这个结论导致的一个结果是, 公式 (2.9) 定义的二项分布 (二元随机变量 x 在 N 次观测中出现次数 m 的分布) 将会在 $N \rightarrow \infty$ 时趋向于高斯分布 (图 2.1 给出了 $N = 10$ 的情形)。

高斯分布有许多重要的分析性质, 我们稍后将详细讨论这些性质。这就使得本节将会相当依赖于之前章节中的技术, 并且需要对各种矩阵性质比较熟悉。但是, 我们强烈鼓励读者能够使用这里介绍的技术熟练操作高斯分布, 因为这对于理解后续章节中出现的更加复杂的模型是非常有帮助的。

作为开始，我们考虑高斯分布的几何形式。高斯对于 x 的依赖是通过下面形式的二次型

$$\Delta^2 = (\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}) \quad (2.44)$$

这个二次型出现在指数位置上。 Δ 被叫做 $\boldsymbol{\mu}$ 和 x 之间的马氏距离（Mahalanobis distance）。当 $\boldsymbol{\Sigma}$ 是单位矩阵时，就变成了欧式距离。对于 x 空间中这个二次型是常数的曲面，高斯分布也是常数。

首先，我们注意到矩阵 $\boldsymbol{\Sigma}$ 可以取为对称矩阵，而不失一般性。这是因为任何非对称项都会从指数中消失。现在考虑协方差矩阵的特征向量方程

$$\boldsymbol{\Sigma} \mathbf{u}_i = \lambda_i \mathbf{u}_i \quad (2.45)$$

其中 $i = 1, \dots, D$ 。由于 $\boldsymbol{\Sigma}$ 是实对称矩阵，因此它的特征值也是实数，并且特征向量可以被选成单位正交的，即

$$\mathbf{u}_i^T \mathbf{u}_j = I_{ij} \quad (2.46)$$

其中 I_{ij} 是单位矩阵的第 i, j 个元素，满足

$$I_{ij} = \begin{cases} 1, & \text{如果 } i = j \\ 0, & \text{其他情况} \end{cases} \quad (2.47)$$

协方差矩阵 $\boldsymbol{\Sigma}$ 可以表示成特征向量的展开的形式

$$\boldsymbol{\Sigma} = \sum_{i=1}^D \lambda_i \mathbf{u}_i \mathbf{u}_i^T \quad (2.48)$$

类似地，协方差矩阵的逆矩阵 $\boldsymbol{\Sigma}^{-1}$ 可以表示为

$$\boldsymbol{\Sigma}^{-1} = \sum_{i=1}^D \frac{1}{\lambda_i} \mathbf{u}_i \mathbf{u}_i^T \quad (2.49)$$

把公式 (2.49) 代入公式 (2.44)，二次型就变成了

$$\Delta^2 = \sum_{i=1}^D \frac{y_i^2}{\lambda_i} \quad (2.50)$$

其中我们定义

$$y_i = \mathbf{u}_i^T (\mathbf{x} - \boldsymbol{\mu}) \quad (2.51)$$

我们可以把 $\{y_i\}$ 表示成单位正交向量 \mathbf{u}_i 关于原始的 x_i 坐标经过平移和旋转后形成的新的坐标系。定义向量 $\mathbf{y} = (y_1, \dots, y_D)^T$ ，我们有

$$\mathbf{y} = \mathbf{U}(\mathbf{x} - \boldsymbol{\mu}) \quad (2.52)$$

其中 \mathbf{U} 是一个矩阵，它的行是向量 \mathbf{u}_i^T 。从公式 (2.46) 可以看出 \mathbf{U} 是一个正交orthogonal矩阵，即它满足性质 $\mathbf{U}\mathbf{U}^T = \mathbf{I}$ ，因此也满足 $\mathbf{U}^T\mathbf{U} = \mathbf{I}$ ，其中 \mathbf{I} 是单位矩阵。

二次型在公式 (2.50) 为常数的曲面上为常数，因此高斯密度也是常数。如果所有的特征值 λ_i 都是正数，那么这些曲面表示椭球面，椭球中心位于 $\boldsymbol{\mu}$ ，椭球的轴的方向沿着 \mathbf{u}_i ，沿着轴向的缩放因子为 $\lambda_i^{1/2}$ ，如图2.7所示。

对于将要定义的高斯分布，有必要要求协方差矩阵的所有特征值 λ_i 严格大于零，否则分布将不能被正确地归一化。一个特征值严格大于零的矩阵被称为正定 (positive definite) 矩阵。在第12章，我们会遇到一个或者多个特征值为零的高斯分布，那种情况下分布是奇异的，被限制在了一个低维的子空间中。如果所有的特征值都是非负的，那么这个矩阵被称为半正定 (positive semidefinite) 矩阵。

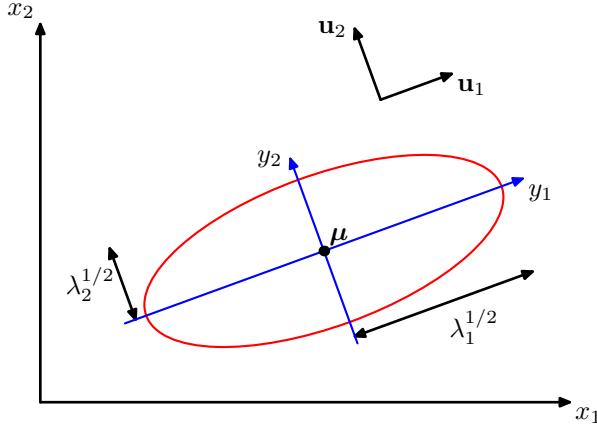


图 2.7: 红色曲线表示二维空间 $\mathbf{x} = (x_1, x_2)$ 的高斯分布的常数概率密度的椭圆面, 它表示的概率密度为 $\exp(-1/2)$, 值是在 $\mathbf{x} = \boldsymbol{\mu}$ 处计算的。椭圆的轴由协方差矩阵的特征向量 \mathbf{u}_i 定义, 对应的特征值为 λ_i 。

现在考虑在由 y_i 定义的新坐标系下高斯分布的形式。从 x 坐标系到 y 坐标系, 我们有一个 Jacobian 矩阵 \mathbf{J} , 它的元素为

$$J_{ij} = \frac{\partial x_i}{\partial y_j} = U_{ij} \quad (2.53)$$

其中 U_{ji} 是矩阵 \mathbf{U}^T 的元素。使用矩阵 \mathbf{U} 的单位正交性质, 我们看到 Jacobian 矩阵行列式的平方为

$$|\mathbf{J}^2| = |\mathbf{U}^T|^2 = |\mathbf{U}^T| |\mathbf{U}| = |\mathbf{U}^T \mathbf{U}| = |\mathbf{I}| = 1 \quad (2.54)$$

因此 $|\mathbf{J}| = 1$ 。并且, 行列式 $|\Sigma|$ 的协方差矩阵可以写成特征值的乘积, 因此

$$|\Sigma|^{\frac{1}{2}} = \prod_{j=1}^D \lambda_j^{\frac{1}{2}} \quad (2.55)$$

因此在 y_j 坐标系中, 高斯分布的形式为

$$p(\mathbf{y}) = p(\mathbf{x}) |\mathbf{J}| = \prod_{j=1}^D \frac{1}{(2\pi\lambda_j)^{\frac{1}{2}}} \exp\left\{-\frac{y_j^2}{2\lambda_j}\right\} \quad (2.56)$$

这是 D 个独立一元高斯分布的乘积。特征向量因此定义了一个新的旋转、平移的坐标系, 在这个坐标系中联合概率分布可以分解成独立分布的乘积。在 y 坐标系中, 概率分布的积分为

$$\int p(\mathbf{y}) d\mathbf{y} = \prod_{j=1}^D \int_{-\infty}^{\infty} \frac{1}{(2\pi\lambda_j)^{\frac{1}{2}}} \exp\left\{-\frac{y_j^2}{2\lambda_j}\right\} dy_j = 1 \quad (2.57)$$

我们现在考察高斯分布的矩, 这描述了参数 $\boldsymbol{\mu}$ 和 Σ 。高斯分布下 \mathbf{x} 的期望为

$$\begin{aligned} \mathbb{E}[\mathbf{x}] &= \frac{1}{(2\pi)^{\frac{D}{2}} |\Sigma|^{\frac{1}{2}}} \int \exp\left\{-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{x} - \boldsymbol{\mu})\right\} \mathbf{x} d\mathbf{x} \\ &= \frac{1}{(2\pi)^{\frac{D}{2}} |\Sigma|^{\frac{1}{2}}} \int \exp\left\{-\frac{1}{2}z^T \Sigma^{-1} z\right\} (z + \boldsymbol{\mu}) dz \end{aligned} \quad (2.58)$$

其中我们使用 $\mathbf{x} = \mathbf{x} - \boldsymbol{\mu}$ 进行了变量替换。我们现在注意到指数位置是 z 的偶函数, 并且由于积分区间为 $(-\infty, \infty)$, 因此在因子 $(z + \boldsymbol{\mu})$ 中的 z 中的项会由于对称性变为零。因此

$$\mathbb{E}[\mathbf{x}] = \boldsymbol{\mu} \quad (2.59)$$

因此我们把 μ 称为高斯分布的均值。

我们现在考虑高斯分布的二阶矩。在一元变量的情形下，二阶矩由 $\mathbb{E}[x^2]$ 给出。对于多元高斯分布，有 D^2 个由 $\mathbb{E}[x_i x_j]$ 给出的二阶矩，可以聚集在一起组成矩阵 $\mathbb{E}[xx^T]$ 。这个矩阵可以写成

$$\begin{aligned}\mathbb{E}[xx^T] &= \frac{1}{(2\pi)^{\frac{D}{2}}} \frac{1}{|\Sigma|^{\frac{1}{2}}} \int \exp \left\{ -\frac{1}{2}(x - \mu)^T \Sigma^{-1} (x - \mu) \right\} xx^T dx \\ &= \frac{1}{(2\pi)^{\frac{D}{2}}} \frac{1}{|\Sigma|^{\frac{1}{2}}} \int \exp \left\{ -\frac{1}{2}z^T \Sigma^{-1} z \right\} (z + \mu)(z + \mu)^T dz\end{aligned}$$

其中，我们再次应用了 $z = x - \mu$ 来进行变量替换。注意，涉及到 μz^T 和 $z \mu^T$ 的交叉项将再次由于对称性而变为零。项 $\mu \mu^T$ 是常数，可以从积分中拿出。它本身等于单位矩阵，因为高斯分布是归一化的。考虑涉及到 zz^T 的项。我们可以再次使用公式 (2.45) 给出的协方差矩阵的特征向量展开，以及特征向量集合的完备性，得到

$$z = \sum_{j=1}^D y_j u_j \quad (2.60)$$

其中 $y_j = u_j^T z$ ，因此

$$\begin{aligned}&\frac{1}{(2\pi)^{\frac{D}{2}}} \frac{1}{|\Sigma|^{\frac{1}{2}}} \int \exp \left\{ -\frac{1}{2}z^T \Sigma^{-1} z \right\} zz^T dz \\ &= \frac{1}{(2\pi)^{\frac{D}{2}}} \frac{1}{|\Sigma|^{\frac{1}{2}}} \sum_{i=1}^D \sum_{j=1}^D u_i u_j^T \int \exp \left\{ -\sum_{k=1}^D \frac{y_k^2}{2\lambda_k} \right\} y_i y_j dy \\ &= \sum_{i=1}^D u_i u_i^T \lambda_i = \Sigma\end{aligned} \quad (2.61)$$

推导过程中我们使用了特征向量方程 (2.45)，以及下面的事实：中间一行的等式右侧的积分由于对称性会等于零（除非 $i = j$ ）。最后一行，我们使用了公式 (1.50) 和公式 (2.55)，以及公式 (2.48)。因此我们有

$$\mathbb{E}[xx^T] = \mu \mu^T + \Sigma \quad (2.62)$$

对于一元随机变量的方差，为了定义方差，我们在取二阶矩之前会减掉均值。类似地，对于多元变量的情形，把均值减掉同样很方便。这给出了随机变量 x 的协方差 (covariance)，定义为

$$\text{var}[x] = \mathbb{E} [(x - \mathbb{E}[x])(x - \mathbb{E}[x])^T] \quad (2.63)$$

对于高斯分布这一特例，我们可以使用 $\mathbb{E}[x] = \mu$ 以及公式 (2.62) 的结果，得到

$$\text{var}[x] = \Sigma \quad (2.64)$$

由于参数 Σ 公式了高斯分布下 x 的协方差，因此它被称为协方差矩阵。

虽然高斯分布 (2.43) 被广泛用作概率密度模型，但是它有着一些巨大的局限性。考虑分布中自由参数的数量。一个通常的对称协方差矩阵 Σ 有 $\frac{D(D+1)}{2}$ 个独立参数， μ 中有另外 D 个独立参数，因此总计有 $\frac{D(D+3)}{2}$ 个参数。对于大的 D 值，参数的总数随着 D 以平方的方式增长，并且对大矩阵进行计算、求逆会变得无法计算。解决这个问题的一种方式是使用协方差矩阵的限制形式。如果我们考虑对角的 (diagonal) 协方差矩阵，即 $\Sigma = \text{diag}(\sigma_i^2)$ ，那么在概率密度模型中，我们就有总数 $2D$ 个独立参数。常数密度的对应的轮廓线是与轴对齐的椭球。我们可以进一步地把协方差矩阵限制成正比于单位矩阵， $\Sigma = \sigma^2 I$ ，被称为各向同性isotropic的协方差。这使得模型有 $D + 1$ 个独立的参数，并且常数概率密度是球面。图2.8给出了通常的协方差矩阵、对角的协方差矩阵以及各向同性协方差矩阵的概率。不幸的是，尽管这样的方法限制了概率分布的自由度的数量，并且使得求协方差矩阵的逆矩阵可以更快地完成，但是这样做也极大地限制了概率密度的形式，限制了它描述模型中有趣的相关性的能力。

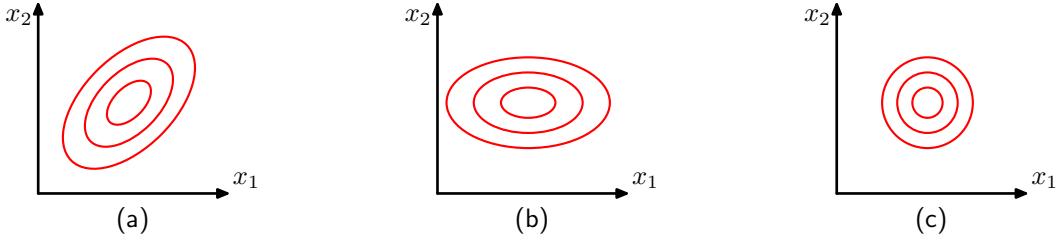


图 2.8: 二维高斯分布的常数概率密度轮廓线，其中，(a)图对应的协方差矩阵为一般形式，(b)图对应的协方差矩阵为对角矩阵，图中椭圆的轮廓线与坐标轴对齐，(c)图对应的协方差矩阵正比于单位矩阵，图中的轮廓线是同心圆。

高斯分布的另一个局限性是它本质上是单峰的（即只有一个最大值），因此不能够很好地近似多峰分布。因此高斯分布一方面相当灵活，因为它有很多参数。另一方面，它又有很大的局限性，因为它不能够近似很多概率分布。我们稍后会看到，引入潜在变量（latent variable），也被称为隐藏变量（hidden variable）或者未观察变量（unobserved variable），会让这两个问题都得到解决。特别地，通过引入离散型潜在变量，相当多的多峰分布可以使用混合高斯分布来描述（在2.3.9节讨论）。类似地，正如第12章所述，引入连续型潜在变量可以产生出一种模型，这种模型中自由参数可以被控制成与数据空间的维度 D 无关，同时仍然允许模型描述数据集里主要的相关性关系。实际上，这两种方法可以结合起来，进一步扩展，推导出一大类层次模型，这些模型可以适用于相当多的实际应用。例如，广泛用作图像的概率模型的高斯版本马尔科夫随机场（Markov random field）是像素灰度空间的高斯分布，但是通过引入能够反映空间中像素组织的结构，这种分布可以很方便地处理。类似地，线性动态系统（linear dynamical system），用来对涉及到时序数据的应用（例如视频跟踪）进行建模，也是一个联合高斯分布。这个分布涉及到相当多的观测变量和潜在变量。但是通过分布上的结构信息，我们可以很方便地进行处理。表达这种复杂分布的形式和性质的一个强大的框架是概率图模型，这是第8章的主题。

2.3.1 条件高斯分布

多元高斯分布的一个重要性质是，如果两组变量是联合高斯分布，那么以一组变量为条件，另一组变量同样是高斯分布。类似地，任何一个变量的边缘分布也是高斯分布。

首先考虑条件概率的情形。假设 \mathbf{x} 是一个服从高斯分布 $\mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma})$ 的 D 维向量。我们把 \mathbf{x} 划分成两个不相交的子集 \mathbf{x}_a 和 \mathbf{x}_b 。不失一般性，我们可以令 \mathbf{x}_a 为 \mathbf{x} 的前 M 个分量，令 \mathbf{x}_b 为剩余的 $D - M$ 个分量，因此

$$\mathbf{x} = \begin{pmatrix} \mathbf{x}_a \\ \mathbf{x}_b \end{pmatrix} \quad (2.65)$$

我们也定义对应的对均值向量 $\boldsymbol{\mu}$ 的划分，即

$$\boldsymbol{\mu} = \begin{pmatrix} \boldsymbol{\mu}_a \\ \boldsymbol{\mu}_b \end{pmatrix} \quad (2.66)$$

协方差矩阵 $\boldsymbol{\Sigma}$ 为

$$\boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma}_{aa} & \boldsymbol{\Sigma}_{ab} \\ \boldsymbol{\Sigma}_{ba} & \boldsymbol{\Sigma}_{bb} \end{pmatrix} \quad (2.67)$$

注意，协方差矩阵的对称性 $\boldsymbol{\Sigma}^T = \boldsymbol{\Sigma}$ 表明 $\boldsymbol{\Sigma}_{aa}$ 和 $\boldsymbol{\Sigma}_{bb}$ 也是对称的，而 $\boldsymbol{\Sigma}_{ba} = \boldsymbol{\Sigma}_{ab}^T$ 。

在许多情况下，使用协方差矩阵的逆矩阵比较方便。即

$$\boldsymbol{\Lambda} \equiv \boldsymbol{\Sigma}^{-1} \quad (2.68)$$

这被称为精度矩阵（precision matrix）。事实上，我们会看到，高斯分布的一些性质可以使用协方差来自然地表达出来，而其他的性质如果使用精度表示，形式会更简单。于是我们也可以引

入精度矩阵的划分形式

$$\boldsymbol{\Lambda} = \begin{pmatrix} \boldsymbol{\Lambda}_{aa} & \boldsymbol{\Lambda}_{ab} \\ \boldsymbol{\Lambda}_{ba} & \boldsymbol{\Lambda}_{bb} \end{pmatrix} \quad (2.69)$$

对应于向量 \mathbf{x} 的划分 (2.65)。由于一个对称矩阵的逆矩阵也是对称矩阵，我们可以看到 $\boldsymbol{\Lambda}_{aa}$ 和 $\boldsymbol{\Lambda}_{bb}$ 是对称的，而 $\boldsymbol{\Lambda}_{ab}^T = \boldsymbol{\Lambda}_{ba}$ 。这里应该强调的一点是， $\boldsymbol{\Lambda}_{aa}$ 不是简单地对 $\boldsymbol{\Sigma}_{aa}$ 求逆。事实上，我们稍后会简单考察划分矩阵的逆矩阵与各个分块的逆矩阵之间的关系。

首先，我们来寻找条件概率分布 $p(\mathbf{x}_a | \mathbf{x}_b)$ 的表达式。根据概率的乘积规则，我们看到，条件分布可以根据联合分布 $p(\mathbf{x}) = p(\mathbf{x}_a, \mathbf{x}_b)$ 很容易地计算出来。我们只需把 \mathbf{x}_b 固定为观测值，然后对得到的表达式进行归一化，得到 \mathbf{x}_a 的一个合法的概率分布。我们不显示地进行归一化，相反，我们可以用一种更有效率的方式求解。我们首先考虑由公式 (2.44) 给出的高斯分布指数项中出现的二次型，然后在计算的最后阶段重新考虑归一化系数。如果我们使用公式 (2.65)、公式 (2.66) 和公式 (2.69) 的划分方式，我们有

$$\begin{aligned} -\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu}) = \\ -\frac{1}{2}(\mathbf{x}_a - \boldsymbol{\mu}_a)^T \boldsymbol{\Lambda}_{aa}(\mathbf{x}_a - \boldsymbol{\mu}_a) - \frac{1}{2}(\mathbf{x}_a - \boldsymbol{\mu}_a)^T \boldsymbol{\Lambda}_{ab}(\mathbf{x}_b - \boldsymbol{\mu}_b) \\ -\frac{1}{2}(\mathbf{x}_b - \boldsymbol{\mu}_b)^T \boldsymbol{\Lambda}_{ba}(\mathbf{x}_a - \boldsymbol{\mu}_a) - \frac{1}{2}(\mathbf{x}_b - \boldsymbol{\mu}_b)^T \boldsymbol{\Lambda}_{bb}(\mathbf{x}_b - \boldsymbol{\mu}_b) \end{aligned} \quad (2.70)$$

我们把它看成 \mathbf{x}_a 的函数，这又是一个二次型，因此对应的条件分布 $p(\mathbf{x}_a | \mathbf{x}_b)$ 是高斯分布。由于分布由均值和协方差完全确定，因此我们的目标是通过观察公式 (2.70) 找到 $p(\mathbf{x}_a | \mathbf{x}_b)$ 的均值和协方差的表达式。

这是一个与高斯分布相关的相当常见的操作，有时被称为“完成平方项”。这种方法中，我们一直一个二次型，这个二次型定义了高斯分布的指数项，我们需要确定对应的均值和协方差。这种问题可以这样解决：我们注意到一个一般的高斯分布 $\mathcal{N}(\mathbf{x} | \boldsymbol{\mu}, \boldsymbol{\Sigma})$ 的指数项可以写成

$$-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu}) = -\frac{1}{2}\mathbf{x}^T \boldsymbol{\Sigma}^{-1}\mathbf{x} + \mathbf{x}^T \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} + \text{常数} \quad (2.71)$$

其中，“常数”表示与 \mathbf{x} 无关的项，并且我们用到了 $\boldsymbol{\Sigma}$ 的对称性。因此，如果我们把普通的二次型表示成公式 (2.71) 右侧的形式，那么我们可以立即令 \mathbf{x} 中的二阶项的系数矩阵等于协方差矩阵的逆矩阵 $\boldsymbol{\Sigma}^{-1}$ ，令 \mathbf{x} 中的线性项的系数等于 $\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}$ ，这样我们就可以得到 $\boldsymbol{\mu}$

现在让我们把这个方法应用到条件高斯分布 $p(\mathbf{x}_a | \mathbf{x}_b)$ 中。条件高斯分布的指数项的二次型由公式 (2.70) 给出。我们把这个分布的均值和协方差分别记作 $\boldsymbol{\mu}_{a|b}$ 和 $\boldsymbol{\Sigma}_{a|b}$ 。考虑公式 (2.70) 对 \mathbf{x}_a 的函数依赖关系，其中 \mathbf{x}_b 被当成常数。如果我们选出所有 \mathbf{x}_a 的二阶项，那么我们有

$$-\frac{1}{2}\mathbf{x}_a^T \boldsymbol{\Lambda}_{aa} \mathbf{x}_a \quad (2.72)$$

从这个公式中，我们可以立即看出， $p(\mathbf{x}_a | \mathbf{x}_b)$ 的协方差（精度矩阵的逆矩阵）为

$$\boldsymbol{\Sigma}_{a|b} = \boldsymbol{\Lambda}_{aa}^{-1} \quad (2.73)$$

现在考虑公式 (2.70) 中所有 \mathbf{x}_a 的常数项

$$\mathbf{x}_a^T \{\boldsymbol{\Lambda}_{aa} \boldsymbol{\mu}_a - \boldsymbol{\Lambda}_{ab}(\mathbf{x}_b - \boldsymbol{\mu}_b)\} \quad (2.74)$$

其中，我们使用了 $\boldsymbol{\Lambda}_{ba}^T = \boldsymbol{\Lambda}_{ab}$ 这个等式。根据我们对一般形式 (2.71) 的讨论，这个表达式中 \mathbf{x}_a 的系数一定等于 $\boldsymbol{\Sigma}_{a|b}^{-1} \boldsymbol{\mu}_{a|b}$ ，因此

$$\begin{aligned} \boldsymbol{\mu}_{a|b} &= \boldsymbol{\Sigma}_{a|b} \{\boldsymbol{\Lambda}_{aa} \boldsymbol{\mu}_a - \boldsymbol{\Lambda}_{ab}(\mathbf{x}_b - \boldsymbol{\mu}_b)\} \\ &= \boldsymbol{\mu}_a - \boldsymbol{\Lambda}_{aa}^{-1} \boldsymbol{\Lambda}_{ab}(\mathbf{x}_b - \boldsymbol{\mu}_b) \end{aligned} \quad (2.75)$$

推导过程中我们使用了公式 (2.73)。

结果 (2.73) 和 (2.75) 是根据原始联合分布 $p(\mathbf{x}_a, \mathbf{x}_b)$ 的分块精度矩阵进行表达的。我们也可以根据对应的分块协方差矩阵来表达这些结果。为了完成这一点，我们使用下面的关于分块矩阵的逆矩阵的恒等式

$$\begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{pmatrix}^{-1} = \begin{pmatrix} \mathbf{M} & -\mathbf{MBD}^{-1} \\ -\mathbf{D}^{-1}\mathbf{CM} & \mathbf{D}^{-1} + \mathbf{D}^{-1}\mathbf{CMBD}^{-1} \end{pmatrix} \quad (2.76)$$

其中我们已经定义了

$$\mathbf{M} = (\mathbf{A} - \mathbf{BD}^{-1}\mathbf{C})^{-1} \quad (2.77)$$

\mathbf{M}^{-1} 被称为公式 (2.76) 左侧矩阵关于子矩阵 \mathbf{D} 的舒尔补 (Schur complement)。使用定义

$$\begin{pmatrix} \boldsymbol{\Sigma}_{aa} & \boldsymbol{\Sigma}_{ab} \\ \boldsymbol{\Sigma}_{ba} & \boldsymbol{\Sigma}_{bb} \end{pmatrix}^{-1} = \begin{pmatrix} \boldsymbol{\Lambda}_{aa} & \boldsymbol{\Lambda}_{ab} \\ \boldsymbol{\Lambda}_{ba} & \boldsymbol{\Lambda}_{bb} \end{pmatrix} \quad (2.78)$$

使用公式 (2.76)，我们有

$$\boldsymbol{\Lambda}_{aa} = (\boldsymbol{\Sigma}_{aa} - \boldsymbol{\Sigma}_{ab}\boldsymbol{\Sigma}_{bb}^{-1}\boldsymbol{\Sigma}_{ba})^{-1} \quad (2.79)$$

$$\boldsymbol{\Lambda}_{ab} = -(\boldsymbol{\Sigma}_{aa} - \boldsymbol{\Sigma}_{ab}\boldsymbol{\Sigma}_{bb}^{-1}\boldsymbol{\Sigma}_{ba})^{-1}\boldsymbol{\Sigma}_{ab}\boldsymbol{\Sigma}_{bb}^{-1} \quad (2.80)$$

从这些结果中，我们可以得到条件概率分布 $p(\mathbf{x}_a | \mathbf{x}_b)$ 的均值和协方差的表达式

$$\boldsymbol{\mu}_{a|b} = \boldsymbol{\mu}_a + \boldsymbol{\Sigma}_{ab}\boldsymbol{\Sigma}_{bb}^{-1}(\mathbf{x}_b - \boldsymbol{\mu}_b) \quad (2.81)$$

$$\boldsymbol{\Sigma}_{a|b} = \boldsymbol{\Sigma}_{aa} - \boldsymbol{\Sigma}_{ab}\boldsymbol{\Sigma}_{bb}^{-1}\boldsymbol{\Sigma}_{ba} \quad (2.82)$$

对比公式 (2.73) 和 (2.82)，我们看到条件概率分布 $p(\mathbf{x}_a | \mathbf{x}_b)$ 如果使用分块精度矩阵而不是分块协方差矩阵表示，那么它的形式会更简单。注意，条件概率分布 $p(\mathbf{x}_a | \mathbf{x}_b)$ 的均值（由公式 (2.81) 给出）是 \mathbf{x}_b 的线性函数，协方差（由公式 (2.82) 给出）与 \mathbf{x}_b 无关。这是线性高斯 (linear-Gaussian) 模型的一个例子。

2.3.2 边缘高斯分布

我们已经看到，如果联合分布 $p(\mathbf{x}_a, \mathbf{x}_b)$ 是高斯分布，那么条件概率分布 $p(\mathbf{x}_a | \mathbf{x}_b)$ 也是高斯分布。现在我们要讨论边缘概率分布

$$p(\mathbf{x}_a) = \int p(\mathbf{x}_a, \mathbf{x}_b) d\mathbf{x}_b \quad (2.83)$$

正如我们即将看到的那样，这也是一个高斯分布。和之前一样，我们高效估计这个概率分布的策略是把注意力集中于联合分布的指数项的二次型，然后找出边缘分布 $p(\mathbf{x}_a)$ 的均值和协方差。

联合分布的二次型可以使用分块精度矩阵表示成公式 (2.70) 的形式。由于我们的目标是积分出 \mathbf{x}_b ，这可以按照下面的方式很容易地计算出来：首先考虑涉及到 \mathbf{x}_b 的项，然后配出平方项，使得积分能够更方便地计算。选出涉及到 \mathbf{x}_b 的项，我们有

$$-\frac{1}{2}\mathbf{x}_b^T \boldsymbol{\Lambda}_{bb} \mathbf{x}_b + \mathbf{x}_b^T \mathbf{m} = -\frac{1}{2}(\mathbf{x}_b - \boldsymbol{\Lambda}_{bb}^{-1}\mathbf{m})^T \boldsymbol{\Lambda}_{bb} (\mathbf{x}_b - \boldsymbol{\Lambda}_{bb}^{-1}\mathbf{m}) + \frac{1}{2}\mathbf{m}^T \boldsymbol{\Lambda}_{bb}^{-1}\mathbf{m} \quad (2.84)$$

其中，我们定义了

$$\mathbf{m} = \boldsymbol{\Lambda}_{bb}\boldsymbol{\mu}_b - \boldsymbol{\Lambda}_{ba}(\mathbf{x}_a - \boldsymbol{\mu}_a) \quad (2.85)$$

我们看到，与 \mathbf{x}_b 相关的项已经被转化为了一个高斯分布的标准二次型，这对应于公式 (2.84) 的右侧的第一项，加上一个与 \mathbf{x}_b 无关（但是与 \mathbf{x}_a 相关）的项。所以，当我们取这个二次型作为高斯分布的指数项时，我们看到公式 (2.83) 要求的关于 \mathbf{x}_b 的积分的形式为

$$\int \exp \left\{ -\frac{1}{2}(\mathbf{x}_b - \boldsymbol{\Lambda}_{bb}^{-1}\mathbf{m})^T \boldsymbol{\Lambda}_{bb} (\mathbf{x}_b - \boldsymbol{\Lambda}_{bb}^{-1}\mathbf{m}) \right\} d\mathbf{x}_b \quad (2.86)$$

这个积分很容易计算。我们注意到，它是一个在未归一化的高斯分布上做的积分，因此结果是归一化系数的倒数。从公式 (2.43) 给出的归一化的高斯分布的形式，我们可以看到，高斯分布的系数与均值无关，只依赖于协方差矩阵的行列式。因此，通过关于 \mathbf{x}_b 配出平方项的方法，我们能够积分出 \mathbf{x}_b ，这样由于公式 (2.84) 的左侧的贡献，唯一剩余的与 \mathbf{x}_a 相关的项就是公式 (2.84) 的右侧的最后一项，其中 \mathbf{m} 由公式 (2.85) 给出。把这一项与公式 (2.70) 中余下的与 \mathbf{x}_a 相关的项结合，我们有

$$\begin{aligned} & \frac{1}{2} [\Lambda_{bb}\boldsymbol{\mu}_b - \Lambda_{ba}(\mathbf{x}_a - \boldsymbol{\mu}_a)]^T \Lambda_{bb}^{-1} [\Lambda_{bb}\boldsymbol{\mu}_b - \Lambda_{ba}(\mathbf{x}_a - \boldsymbol{\mu}_a)] \\ & - \frac{1}{2} \mathbf{x}_a^T \Lambda_{aa} \mathbf{x}_a + \mathbf{x}_a^T (\Lambda_{aa}\boldsymbol{\mu}_a + \Lambda_{ab}\boldsymbol{\mu}_b) + \text{常数} \\ & = -\frac{1}{2} \mathbf{x}_a^T (\Lambda_{aa} - \Lambda_{ab}\Lambda_{bb}^{-1}\Lambda_{ba}) \mathbf{x}_a \\ & + \mathbf{x}_a^T (\Lambda_{aa} - \Lambda_{ab}\Lambda_{bb}^{-1}\Lambda_{ba}) \boldsymbol{\mu}_a + \text{常数} \end{aligned} \quad (2.87)$$

其中，“常数”表示与 \mathbf{x}_a 无关的量。再次与公式 (2.71) 比较，我们可以看到边缘概率分布 $p(\mathbf{x}_a)$ 的协方差矩阵为

$$\Sigma_a = (\Lambda_{aa} - \Lambda_{ab}\Lambda_{bb}^{-1}\Lambda_{ba})^{-1} \quad (2.88)$$

类似地，均值由下式给出

$$\Sigma_a(\Lambda_{aa} - \Lambda_{ab}\Lambda_{bb}^{-1}\Lambda_{ba})\boldsymbol{\mu}_a = \boldsymbol{\mu}_a \quad (2.89)$$

其中我们使用了公式 (2.88) 的结果。协方差 (2.88) 是用公式 (2.69) 给出分块精度矩阵表达的。我们可以用公式 (2.67) 给出的对应的分块协方差矩阵重写这个结果，就像我们在条件概率分布时做的那样。这两个分块矩阵的关系为

$$\begin{pmatrix} \Lambda_{aa} & \Lambda_{ab} \\ \Lambda_{ba} & \Lambda_{bb} \end{pmatrix}^{-1} = \begin{pmatrix} \Sigma_{aa} & \Sigma_{ab} \\ \Sigma_{ba} & \Sigma_{bb} \end{pmatrix} \quad (2.90)$$

使用公式 (2.76)，我们有

$$(\Lambda_{aa} - \Lambda_{ab}\Lambda_{bb}^{-1}\Lambda_{ba})^{-1} = \Sigma_{aa} \quad (2.91)$$

这样我们就得到了符合直觉的结果，即边缘概率 $p(\mathbf{x}_a)$ 的均值和协方差为

$$\mathbb{E}[\mathbf{x}_a] = \boldsymbol{\mu}_a \quad (2.92)$$

$$\text{cov}[\mathbf{x}_a] = \Sigma_{aa} \quad (2.93)$$

我们看到对于一个边缘概率分布，如果使用分块协方差矩阵表示，那么均值和方差的表示形式都会得到极大的简化，这与条件概率分布的情形恰好相反。在条件概率分布的情况下，使用分块精度矩阵会得到更加简单的表示形式。

我们关于分块高斯的边缘分布和条件分布的结果可以总结如下。

给定一个联合高斯分布 $\mathcal{N}(\mathbf{x} | \boldsymbol{\mu}, \Sigma)$ ，其中 $\Lambda \equiv \Sigma^{-1}$ ，且

$$\mathbf{x} = \begin{pmatrix} \mathbf{x}_a \\ \mathbf{x}_b \end{pmatrix}, \quad \boldsymbol{\mu} = \begin{pmatrix} \boldsymbol{\mu}_a \\ \boldsymbol{\mu}_b \end{pmatrix} \quad (2.94)$$

$$\Sigma = \begin{pmatrix} \Sigma_{aa} & \Sigma_{ab} \\ \Sigma_{ba} & \Sigma_{bb} \end{pmatrix}, \quad \Lambda = \begin{pmatrix} \Lambda_{aa} & \Lambda_{ab} \\ \Lambda_{ba} & \Lambda_{bb} \end{pmatrix} \quad (2.95)$$

条件概率分布：

$$p(\mathbf{x}_a | \mathbf{x}_b) = \mathcal{N}(\mathbf{x}_a | \boldsymbol{\mu}_{a|b}, \Lambda_{aa}^{-1}) \quad (2.96)$$

$$\boldsymbol{\mu}_{a|b} = \boldsymbol{\mu}_a - \Lambda_{aa}^{-1}\Lambda_{ab}(\mathbf{x}_b - \boldsymbol{\mu}_b) \quad (2.97)$$

边缘概率分布：

$$p(\mathbf{x}_a) = \mathcal{N}(\mathbf{x}_a | \boldsymbol{\mu}_a, \Sigma_{aa}) \quad (2.98)$$

在图2.9中，我们给出了一个涉及到两个变量的多元高斯分布，用来说明条件概率分布和边缘概率分布的思想。

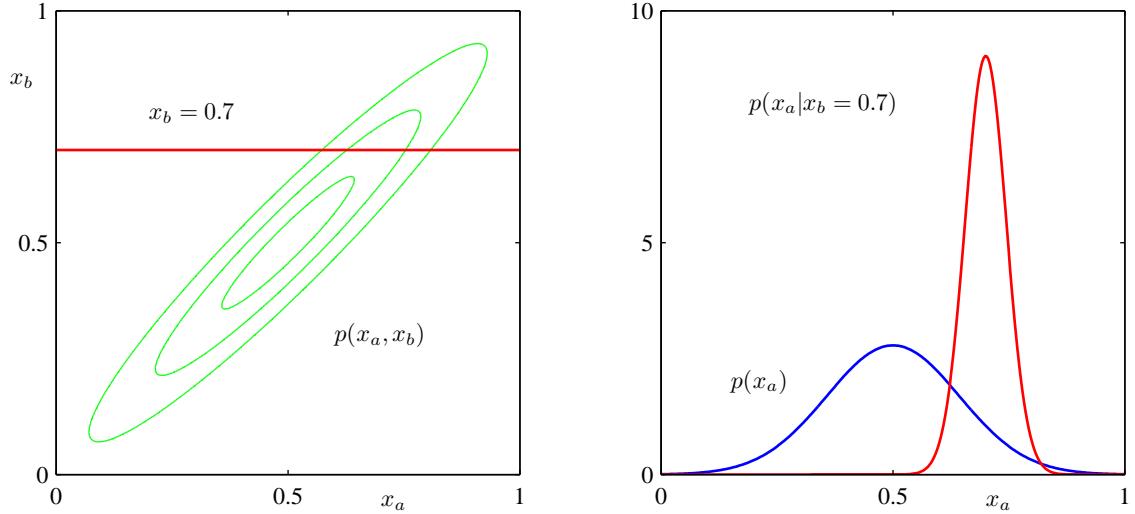


图 2.9: 左图给出了两个变量上的高斯概率分布 $p(x_a, x_b)$ 的轮廓线, 右图给出了边缘概率分布 $p(x_a)$ (蓝色曲线) 和 $x_b = 0.7$ 的条件概率分布 $p(x_a | x_b)$ (红色曲线)。

2.3.3 高斯变量的贝叶斯定理

在2.3.1节和2.3.2节, 我们考虑了高斯分布 $p(\mathbf{x})$ 。在分析的过程中, 我们把向量 \mathbf{x} 切分成了两个子向量 $\mathbf{x} = (\mathbf{x}_a, \mathbf{x}_b)$, 然后找到了条件概率分布 $p(\mathbf{x}_a | \mathbf{x}_b)$ 和边缘概率分布 $p(\mathbf{x}_a)$ 的表达式。我们注意到, 条件分布 $p(\mathbf{x}_a | \mathbf{x}_b)$ 是 \mathbf{x}_b 的线性函数。这里我们将会假定我们被给定一个高斯边缘分布 $p(\mathbf{x})$ 和一个高斯条件分布 $p(\mathbf{y} | \mathbf{x})$, 其中 $p(\mathbf{y} | \mathbf{x})$ 的均值是 \mathbf{x} 的线性函数, 协方差与 \mathbf{x} 无关。这是线性高斯模型 (linear Gaussian model) 的一个例子 (Roweis and Ghahramani, 1999)。我们将在8.1.4节在更一般的情况下研究它。我们想找到边缘概率分布 $p(\mathbf{y})$ 和条件概率分布 $p(\mathbf{x} | \mathbf{y})$ 。这是一个在后续章节中经常出现的问题, 在这里推导出一般的结果会很方便。

我们令边缘概率分布和条件概率分布的形式如下

$$p(\mathbf{x}) = \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}, \boldsymbol{\Lambda}^{-1}) \quad (2.99)$$

$$p(\mathbf{y} | \mathbf{x}) = \mathcal{N}(\mathbf{y} | \mathbf{A}\mathbf{x} + \mathbf{b}, \mathbf{L}^{-1}) \quad (2.100)$$

其中, $\boldsymbol{\mu}$, \mathbf{A} 和 \mathbf{b} 是控制均值的参数, $\boldsymbol{\Lambda}$ 和 \mathbf{L} 是精度矩阵。如果 \mathbf{x} 的维度为 M , \mathbf{y} 的维度为 D , 那么矩阵 \mathbf{A} 的大小为 $D \times M$ 。

首先, 我们寻找 \mathbf{x} 和 \mathbf{y} 的联合分布的表达式。为了做到这一点, 我们定义

$$\mathbf{z} = \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} \quad (2.101)$$

然后考虑联合概率分布的对数

$$\begin{aligned} \ln p(\mathbf{z}) &= \ln p(\mathbf{x}) + \ln p(\mathbf{y} | \mathbf{x}) \\ &= -\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Lambda}(\mathbf{x} - \boldsymbol{\mu}) \\ &\quad -\frac{1}{2}(\mathbf{y} - \mathbf{A}\mathbf{x} - \mathbf{b})^T \mathbf{L}(\mathbf{y} - \mathbf{A}\mathbf{x} - \mathbf{b}) + \text{常数} \end{aligned} \quad (2.102)$$

其中, “常数”表示与 \mathbf{x} 和 \mathbf{y} 无关的项。与之前相同, 我们看到这是 \mathbf{z} 的分量的一个二次函数, 因此 $p(\mathbf{z})$ 是一个高斯分布。为了找到这个高斯分布的精度, 我们考虑公式 (2.102) 的第二项, 它可以写成

$$\begin{aligned} &-\frac{1}{2}\mathbf{x}^T(\boldsymbol{\Lambda} + \mathbf{A}^T \mathbf{L} \mathbf{A})\mathbf{x} - \frac{1}{2}\mathbf{y}^T \mathbf{L} \mathbf{y} + \frac{1}{2}\mathbf{y}^T \mathbf{L} \mathbf{A} \mathbf{x} + \frac{1}{2}\mathbf{x}^T \mathbf{A}^T \mathbf{L} \mathbf{y} \\ &= -\frac{1}{2} \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix}^T \begin{pmatrix} \boldsymbol{\Lambda} + \mathbf{A}^T \mathbf{L} \mathbf{A} & -\mathbf{A}^T \mathbf{L} \\ -\mathbf{L} \mathbf{A} & \mathbf{L} \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} = -\frac{1}{2} \mathbf{z}^T \mathbf{R} \mathbf{z} \end{aligned} \quad (2.103)$$

因此 z 上的高斯分布的精度矩阵（协方差的逆矩阵）为

$$\mathbf{R} = \begin{pmatrix} \Lambda + \mathbf{A}^T \mathbf{L} \mathbf{A} & -\mathbf{A}^T \mathbf{L} \\ -\mathbf{L} \mathbf{A} & \mathbf{L} \end{pmatrix} \quad (2.104)$$

协方差矩阵可以通过取精度矩阵的逆矩阵的方式得到，求逆矩阵可以使用公式 (2.76)。因此

$$\text{cov}[z] = \mathbf{R}^{-1} = \begin{pmatrix} \Lambda^{-1} & \Lambda^{-1} \mathbf{A}^T \\ \mathbf{A} \Lambda^{-1} & \mathbf{L}^{-1} + \mathbf{A} \Lambda^{-1} \mathbf{A}^T \end{pmatrix} \quad (2.105)$$

类似地，我们可以找到 z 上的高斯分布的均值，方法是找到 (2.102) 中的线性项，即

$$\mathbf{x}^T \Lambda \boldsymbol{\mu} - \mathbf{x}^T \mathbf{A}^T \mathbf{L} \mathbf{b} + \mathbf{y}^T \mathbf{L} \mathbf{b} = \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix}^T \begin{pmatrix} \Lambda \boldsymbol{\mu} - \mathbf{A}^T \mathbf{L} \mathbf{b} \\ \mathbf{L} \mathbf{b} \end{pmatrix} \quad (2.106)$$

我们在更早的时候，在多元高斯的二次型中通过完成平方项的方法得到了结果 (2.71)。使用这个结果，我们可得 z 的均值为

$$\mathbb{E}[z] = \mathbf{R}^{-1} \begin{pmatrix} \Lambda \boldsymbol{\mu} - \mathbf{A}^T \mathbf{L} \mathbf{b} \\ \mathbf{L} \mathbf{b} \end{pmatrix} \quad (2.107)$$

使用公式 (2.105)，我们可以得到

$$\mathbb{E}[z] = \begin{pmatrix} \boldsymbol{\mu} \\ \mathbf{A} \boldsymbol{\mu} + \mathbf{b} \end{pmatrix} \quad (2.108)$$

接下来我们寻找边缘分布 $p(\mathbf{y})$ 的表达式，这个边缘分布是通过对 \mathbf{x} 积分得到的。回忆一下，对于高斯随机向量的分量的一个子集的边缘分布，当用分块协方差矩阵来表示时，形式会非常简单。具体地，它的均值和协方差分别由公式 (2.92) 和公式 (2.93) 给出。使用公式 (2.105) 和公式 (2.108)，我们看到边缘分布 $p(\mathbf{y})$ 的均值和协方差为

$$\mathbb{E}[\mathbf{y}] = \mathbf{A} \boldsymbol{\mu} + \mathbf{b} \quad (2.109)$$

$$\text{cov}[\mathbf{y}] = \mathbf{L}^{-1} + \mathbf{A} \Lambda^{-1} \mathbf{A}^T \quad (2.110)$$

一个特殊情况是 $\mathbf{A} = \mathbf{I}$ ，这时它变成了两个高斯的卷积。我们可以看到，卷积的均值是两个高斯的均值的和，卷积的协方差是它们的协方差的和。

最后，我们寻找条件分布 $p(\mathbf{x} | \mathbf{y})$ 的表达式。回忆一下，如果条件概率分布的结果用分块精度矩阵表示，那么结果的形式会更简洁，例如公式 (2.73) 和公式 (2.75)。把这些结果应用到 (2.105) 和 (2.108) 中，我们看到条件分布 $p(\mathbf{x} | \mathbf{y})$ 的均值和方差为

$$\mathbb{E}[\mathbf{x} | \mathbf{y}] = (\Lambda + \mathbf{A}^T \mathbf{L} \mathbf{A})^{-1} \{ \mathbf{A}^T \mathbf{L} (\mathbf{y} - \mathbf{b}) + \Lambda \boldsymbol{\mu} \} \quad (2.111)$$

$$\text{cov}[\mathbf{x} | \mathbf{y}] = (\Lambda + \mathbf{A}^T \mathbf{L} \mathbf{A})^{-1} \quad (2.112)$$

这个条件分布的估计可以看成贝叶斯定理的一个例子。我们可以把分布 $p(\mathbf{x})$ 看成 \mathbf{x} 的先验分布。如果变量 y 被观测到了，那么条件分布 $p(\mathbf{x} | \mathbf{y})$ 表示 \mathbf{x} 的对应的后验分布。找到边缘分布和条件分布，我们可以用 $p(\mathbf{x} | \mathbf{y})p(\mathbf{y})$ 的形式表示联合分布 $p(z) = p(\mathbf{x})p(\mathbf{y} | \mathbf{x})$ 。这些结果总结如下。

给定 \mathbf{x} 的一个边缘高斯分布，以及在给定 \mathbf{x} 的条件下 \mathbf{y} 的条件高斯分布，形式为

$$p(\mathbf{x}) = \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}, \Lambda^{-1}) \quad (2.113)$$

$$p(\mathbf{y} | \mathbf{x}) = \mathcal{N}(\mathbf{y} | \mathbf{A} \mathbf{x} + \mathbf{b}, \mathbf{L}^{-1}) \quad (2.114)$$

\mathbf{y} 的边缘分布以及给定 \mathbf{y} 的条件下 \mathbf{x} 的条件分布为

$$p(\mathbf{y}) = \mathcal{N}(\mathbf{y} | \mathbf{A} \boldsymbol{\mu} + \mathbf{b}, \mathbf{L}^{-1} + \mathbf{A} \Lambda^{-1} \mathbf{A}^T) \quad (2.115)$$

$$p(\mathbf{x} | \mathbf{y}) = \mathcal{N}(\mathbf{x} | \Sigma \{ \mathbf{A}^T \mathbf{L} (\mathbf{y} - \mathbf{b}) + \Lambda \boldsymbol{\mu} \}, \Sigma) \quad (2.116)$$

其中

$$\Sigma = (\Lambda + \mathbf{A}^T \mathbf{L} \mathbf{A})^{-1} \quad (2.117)$$

2.3.4 高斯分布的最大似然估计

给定一个数据集 $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_N)^T$, 其中观测 $\{\mathbf{x}_n\}$ 假定是独立地从多元高斯分布中抽取的。我们可以使用最大似然法估计分布的参数。对数似然函数为

$$\ln p(\mathbf{X} | \boldsymbol{\mu}, \boldsymbol{\Sigma}) = -\frac{ND}{2} \ln(2\pi) - \frac{N}{2} \ln |\boldsymbol{\Sigma}| - \frac{1}{2} \sum_{n=1}^N (\mathbf{x}_n - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x}_n - \boldsymbol{\mu}) \quad (2.118)$$

通过简单的重新排列, 我们看到似然函数对数据集的依赖只通过下面两个量体现

$$\sum_{n=1}^N \mathbf{x}_n, \quad \sum_{n=1}^N \mathbf{x}_n \mathbf{x}_n^T \quad (2.119)$$

这被称为高斯分布的充分统计量 (sufficient statistics)。使用公式 (C.19), 对数似然函数关于 $\boldsymbol{\mu}$ 的导数为

$$\frac{\partial}{\partial \boldsymbol{\mu}} \ln p(\mathbf{X} | \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^N \boldsymbol{\Sigma}^{-1} (\mathbf{x}_n - \boldsymbol{\mu}) \quad (2.120)$$

令这个导数等于零, 我们得到了均值的最大似然估计

$$\boldsymbol{\mu}_{ML} = \frac{1}{N} \sum_{n=1}^N \mathbf{x}_n \quad (2.121)$$

这是数据点的观测集合的均值。公式 (2.118) 关于 $\boldsymbol{\Sigma}$ 的最大化更加复杂。最简单的方法是忽略对称性限制, 然后证明结果是对称的, 正如要求的那样。这个结果的另一种推导方式显式地利用了对称性和正定性的限制, 可以在Magnus and Neudecker (1999) 中找到。结果是符合我们预想情况的, 形式为

$$\boldsymbol{\Sigma}_{ML} = \frac{1}{N} \sum_{n=1}^N (\mathbf{x}_n - \boldsymbol{\mu}_{ML})(\mathbf{x}_n - \boldsymbol{\mu}_{ML})^T \quad (2.122)$$

这个结果涉及到了 $\boldsymbol{\mu}_{ML}$, 因为这是关于 $\boldsymbol{\mu}$ 和 $\boldsymbol{\Sigma}$ 的联合最大值的结果。注意 $\boldsymbol{\mu}_{ML}$ 的解 (2.121) 与 $\boldsymbol{\Sigma}_{ML}$ 无关, 因此我们可以首先求出 $\boldsymbol{\mu}_{ML}$, 然后使用它来求 $\boldsymbol{\Sigma}_{ML}$ 。

如果我们估计真实概率分布下最大似然解的期望, 我们可以得到下面的结果

$$\mathbb{E}[\boldsymbol{\mu}_{ML}] = \boldsymbol{\mu} \quad (2.123)$$

$$\mathbb{E}[\boldsymbol{\Sigma}_{ML}] = \frac{N-1}{N} \boldsymbol{\Sigma} \quad (2.124)$$

我们看到对于均值的最大似然估计的期望等于实际的均值。然而, 对于协方差的最大似然估计的期望小于真正的值, 因此是有偏的。我们可以定义一个不同的估计值 $\tilde{\boldsymbol{\Sigma}}$ 来修正这个误差。新的估计的定义

$$\tilde{\boldsymbol{\Sigma}} = \frac{1}{N-1} \sum_{n=1}^N (\mathbf{x}_n - \boldsymbol{\mu}_{ML})(\mathbf{x}_n - \boldsymbol{\mu}_{ML})^T \quad (2.125)$$

很明显, 根据公式 (2.122) 和公式 (2.124), 期望 $\tilde{\boldsymbol{\Sigma}}$ 等于 $\boldsymbol{\Sigma}$ 。

2.3.5 顺序估计

我们关于高斯分布的参数的最大似然解的讨论提供了一个方便的机会来讨论一个更一般的话题: 最大似然的顺序估计。顺序的方法允许每次处理一个数据点, 然后丢弃这个点。这对于在线应用很重要。并且当数据集相当大以至于一次处理所有数据点不可行的情况下, 顺序方法也很重要。

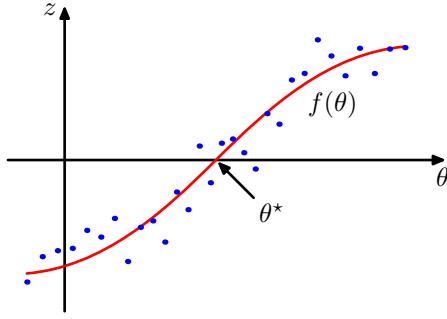


图 2.10: 两个相关的随机变量 z 和 θ 以及由条件期望 $E[z | \theta]$ 给出的回归函数 $f(\theta)$ 的图形表示。Robbins-Monro 算法提供了一个一般的顺序步骤来寻找这种函数的根 θ^* 。

考虑公式 (2.121) 给出的均值的最大似然估计结果 μ_{ML} 。当它依赖于第 N 次观察时，将被记作 $\mu_{ML}^{(N)}$ 。如果我们想分析最后一个数据点 x_N 的贡献，我们有

$$\begin{aligned}\mu_{ML}^{(N)} &= \frac{1}{N} \sum_{n=1}^N x_n \\ &= \frac{1}{N} x_N + \frac{1}{N} \sum_{n=1}^{N-1} x_n \\ &= \frac{1}{N} x_N + \frac{N-1}{N} \mu_{ML}^{(N-1)} \\ &= \mu_{ML}^{(N-1)} + \frac{1}{N} (x_N - \mu_{ML}^{(N-1)})\end{aligned}\tag{2.126}$$

这个结果有一个很好的意义，如后面所述。在观察到 $N-1$ 个数据点后，我们已经把 μ 估计为 $\mu_{ML}^{(N-1)}$ 。我们现在观察到了数据点 x_N ，这样我们就得到了一个修正的估计 $\mu_{ML}^{(N)}$ ，这个估计的获得方式为：把旧的估计沿着“错误信号” $(x_N - \mu_{ML}^{(N-1)})$ 方向移动一个微小的量，这个量正比于 $\frac{1}{N}$ 。注意，随着 N 的增加，后续数据点的贡献也会逐渐变小。

公式 (2.126) 的结果明显与公式 (2.121) 的结果相同，因为这两个公式相等。但是，我们不总是能够使用这种方法推导出一个顺序的算法，因此我们要寻找一个更加通用的顺序学习的方法，这就引出了 Robbins-Monro 算法。考虑一对随机变量 θ 和 z ，它们由一个联合概率分布 $p(z, \theta)$ 所控制。已知 θ 的条件下， z 的条件期望定义了一个确定的函数 $f(\theta)$ ，形式如下

$$f(\theta) \equiv E[z | \theta] = \int z p(z | \theta) dz\tag{2.127}$$

图 2.10 给出了图形化的说明。通过这种方式定义的函数被称为回归函数 (regression function)。

我们的目标是寻找根 θ^* 使得 $f(\theta^*) = 0$ 。如果我们有观测 z 和 θ 的一个大数据集，那么我们可以直接对回归函数建模，得到根的一个估计。但是假设我们每次观测到一个 z 的值，我们想找到一个对应的顺序估计方法来找到 θ^* 。下面的解决这种问题的通用步骤由 Robbins and Monro (1951) 给出。我们假定 z 的条件方差是有穷的，因此

$$E[(z - f)^2 | \theta] < \infty\tag{2.128}$$

并且不失一般性，我们也假设当 $\theta > \theta^*$ 时 $f(\theta) > 0$ ，当 $\theta < \theta^*$ 时 $f(\theta) < 0$ ，如图 2.10 所示。之后，Robbins-Monro 的方法定义了一个根 θ^* 的顺序估计的序列，由下式给出

$$\theta^{(N)} = \theta^{(N-1)} - \alpha_{N-1} z(\theta^{(N-1)})\tag{2.129}$$

其中 $z(\theta^{(N)})$ 是当 θ 的取值为 $\theta^{(N)}$ 时 z 的观测值。系数 $\{\alpha_N\}$ 表示一个满足下列条件的正数序列

$$\lim_{N \rightarrow \infty} \alpha_N = 0\tag{2.130}$$

$$\sum_{N=1}^{\infty} a_N = \infty \quad (2.131)$$

$$\sum_{N=1}^{\infty} a_N^2 < \infty \quad (2.132)$$

可以证明由公式 (2.129) 给出的顺序估计确实以概率1收敛于根 (Robbins and Monro, 1951; Fukunaga, 1990)。注意，第一个条件 (2.130) 确保了后续的修正的幅度会逐渐变小，从而这个过程可以收敛于一个极限值。第二个条件 (2.131) 用来确保算法不会收敛不到根的值。第三个条件 (2.132) 保证了累计的噪声具有一个有限的方差，因此不会导致收敛失败。

现在让我们考虑一个一般的最大似然问题如何使用Robbins-Monro算法顺序地解决。根据定义，最大似然解 θ_{ML} 是负对数似然函数的一个驻点，因此满足

$$\frac{\partial}{\partial \theta} \left\{ \frac{1}{N} \sum_{n=1}^N -\ln p(x_n | \theta) \right\} \Big|_{\theta_{ML}} = 0 \quad (2.133)$$

交换导数与求和，取极限 $N \rightarrow \infty$ ，我们有

$$-\lim_{N \rightarrow \infty} \frac{1}{N} \sum_{n=1}^N \frac{\partial}{\partial \theta} \ln p(x_n | \theta) = \mathbb{E}_x \left[-\frac{\partial}{\partial \theta} \ln p(x | \theta) \right] \quad (2.134)$$

因此我们看到寻找最大似然解对应于寻找回归函数的根。于是我们可以应用Robbins-Monro方法，此时它的形式为

$$\theta^{(N)} = \theta^{(N-1)} - \alpha_{N-1} \frac{\partial}{\partial \theta^{(N-1)}} \left[-\ln p(x_N | \theta^{(N-1)}) \right] \quad (2.135)$$

作为一个具体的例子，我们再次考虑高斯分布均值的顺序估计问题。在这种情况下，参数 $\theta^{(N)}$ 是高斯分布均值 $\mu_{ML}^{(N)}$ 的估计，随机变量 z 的形式为

$$z = -\frac{\partial}{\partial \mu_{ML}} \ln p(x | \mu_{ML}, \sigma^2) = -\frac{1}{\sigma^2} (x - \mu_{ML}) \quad (2.136)$$

因此 z 的分布是一个高斯分布，均值为 $-(\mu - \mu_{ML})/\sigma^2$ ，如图 2.11 所示。把公式 (2.136) 代入公式 (2.135)，我们得到了公式 (2.126) 的单变量形式，其中我们假定选择系数 a_N 的形式为 $a_N = \frac{\sigma^2}{N}$ 。注意，虽然我们刚在只讨论了一元变量的情形，同样的技术，以及公式 (2.130) 到公式 (2.132) 给出的关于系数 a_N 的限制，同样适用于多元变量的情形 (Blum, 1965)。

2.3.6 高斯分布的贝叶斯推断

最大似然框架给出了对于参数 μ 和 Σ 的点估计。现在我们通过引入这些参数的先验分布，介绍一种贝叶斯的方法。首先，让我们考虑一个简单的例子。考虑一个一元高斯随机变量 x ，我们假设方差 σ^2 是已知的。我们的任务是从一组 N 次观测 $\mathbf{x} = \{x_1, \dots, x_N\}$ 中推断均值 μ 。似然函数，即给定 μ 的情况下，观测数据集出现的概率。它可以看成 μ 的函数，由下式给出

$$p(\mathbf{x} | \mu) = \prod_{n=1}^N p(x_n | \mu) = \frac{1}{(2\pi\sigma^2)^{\frac{N}{2}}} \exp \left\{ -\frac{1}{2\sigma^2} \sum_{n=1}^N (x_n - \mu)^2 \right\} \quad (2.137)$$

我们再次强调似然函数 $p(\mathbf{x} | \mu)$ 不是 μ 的概率密度，没有被归一化。

我们看到，似然函数的形式为 μ 的二次型的指数形式。因此如果我们把先验分布 $p(\mu)$ 选成高斯分布，那么它就是似然函数的一个共轭分布，因为对应的后验概率是两个 μ 的二次函数的指数的成绩，因此也是一个高斯分布。于是我们令先验概率分布为

$$p(\mu) = \mathcal{N}(\mu | \mu_0, \sigma_0^2) \quad (2.138)$$

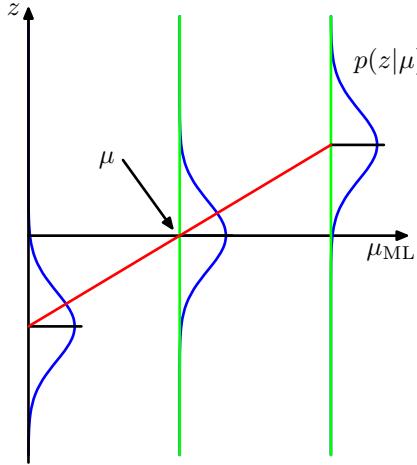


图 2.11: 在高斯分布的情形中, 图2.10所示的回归函数的形式是一条直线, 用红色标记出, 其中 θ 对应于 μ_{ML} 。在这种情况下, 随机变量 z 对应于对数似然函数的导数, 由 $-(x - \mu_{ML})/\sigma^2$ 给出, 定义了回归函数的期望是一条直线, 由 $-(\mu - \mu_{ML})/\sigma^2$ 给出。回归函数的根对应于真实的均值 μ 。

从而后验概率为

$$p(\mu | \mathbf{x}) \propto p(\mathbf{x} | \mu)p(\mu) \quad (2.139)$$

进行诸如对指数项进行完成平方项等简单的计算, 可以证明后验概率的形式为

$$p(\mu | \mathbf{x}) = \mathcal{N}(\mu | \mu_N, \sigma_N^2) \quad (2.140)$$

其中

$$\mu_N = \frac{\sigma^2}{N\sigma_0^2 + \sigma^2}\mu_0 + \frac{N\sigma_0^2}{N\sigma_0^2 + \sigma^2}\mu_{ML} \quad (2.141)$$

$$\frac{1}{\sigma_N^2} = \frac{1}{\sigma_0^2} + \frac{N}{\sigma^2} \quad (2.142)$$

其中 μ_{ML} 是 μ 的最大似然解, 由样本均值给出

$$\mu_{ML} = \frac{1}{N} \sum_{n=1}^N x_n \quad (2.143)$$

花一点时间来研究后验概率分布的均值和方差是很有意义的。首先, 我们注意到由公式(2.141)给出的后验分布的均值是先验均值 μ_0 和最大似然解 μ_{ML} 的折中。如果观测数据点的数量 $N = 0$, 那么与我们想的一样, 公式(2.141)就变成了先验均值。对于 $N \rightarrow \infty$, 后验均值由最大似然解给出。类似地, 考虑公式(2.142)给出的后验分布方差的结果。我们看到, 根据方差的倒数(被称为精度)来表达结果是很自然的事情。另外, 精度是可以相加的, 因此后验概率的精度等于先验的精度加上每一个观测数据点所贡献的一个精度。当我们增加观测数据点的数量时, 精度持续增加, 对应于后验分布的方差持续减少。没有观测数据点, 我们有先验的方差, 而如果数据点的数量 $N \rightarrow \infty$, 方差 σ_N^2 趋于零, 从而后验分布在在最大似然解附近变成了无限大的尖峰。于是我们看到公式(2.143)给出的 μ 的最大似然结果在观测数据点的数量趋于无穷时可以精确地由贝叶斯公式恢复。还要注意, 对于有限的 N 值, 如果我们取极限 $\sigma_0^2 \rightarrow \infty$, 先验的方差会变为无穷大, 那么后验均值(2.141)就变成了最大似然结果, 而后验方差(2.142)为 $\sigma_N^2 = \frac{\sigma^2}{N}$ 。

图2.12说明了高斯分布均值的贝叶斯推断。可以很直接地把这个结果推广到已知方差未知均值的 D 维高斯随机变量 \mathbf{x} 的情况。

我们已经看到高斯分布均值的最大似然表达是如何转化为顺序更新问题的。在顺序更新的框架下, 观测到 N 个数据点之后的均值会根据以下两个量进行表达: 观测到 $N - 1$ 个数据点之后的均值以及数据点 x_N 的贡献。实际上, 对于推断问题来说, 如果从一个顺序的观点来看, 那么贝

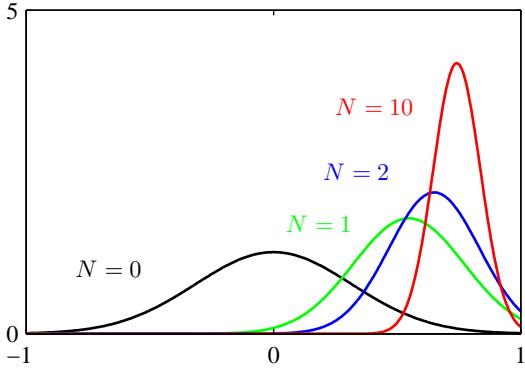


图 2.12: 高斯分布的均值 μ 的贝叶斯推断, 其中我们假设方差已知。曲线展示了 μ 上的先验概率分布 (标记为 $N = 0$ 的曲线), 在这种情况下, 它本身是一个高斯分布。同时给出的还有随着数据点数量 N 的增大, 公式 (2.140) 给出的后验概率分布。数据点由均值为0.8、方差为0.1的高斯分布生成, 先验分布的均值被选择为0。在先验概率分布和似然函数中, 方差都被设置为了真实值。

叶斯方法就变得非常自然了。为了在高斯分布均值推断的问题中说明这一点, 我们把后验分布中最后一个数据点 x_N 的贡献单独写出来, 即

$$p(\mu | \mathbf{x}) \propto \left[p(\mu) \prod_{n=1}^{N-1} p(x_n | \mu) \right] p(x_N | \mu) \quad (2.144)$$

方括号中的项是观测到 $N - 1$ 个数据点之后的后验概率分布 (忽略归一化系数)。我们看到它可以被看成一个先验分布, 然后使用贝叶斯定理与似然函数 (与 x_N 相关) 结合到了一起, 得到了观察到 N 个数据点之后的后验概率。这种贝叶斯推断的顺序观点是非常通用的, 可以应用于任何观测数据独立同分布的问题中。

目前为止, 我们已经假定数据集的高斯分布的方差是已知的, 我们的目标是推断均值。现在假设均值是已知的, 我们要推断方差。同之前一样, 如果我们选择先验分布的共轭形式, 那么计算将会得到极大的简化。可以证明使用精度 $\lambda \equiv \frac{1}{\sigma^2}$ 来进行计算是最方便的。 λ 的似然函数的形式为

$$p(\mathbf{x} | \lambda) = \prod_{n=1}^N \mathcal{N}(x_n | \mu, \lambda^{-1}) \propto \lambda^{\frac{N}{2}} \exp \left\{ -\frac{\lambda}{2} \sum_{n=1}^N (x_n - \mu)^2 \right\} \quad (2.145)$$

对应的共轭先验因此应该正比于 λ 的幂指数, 也正比于 λ 的线性函数的指数。这对应于Gamma分布, 定义为

$$\text{Gam}(\lambda | a, b) = \frac{1}{\Gamma(a)} b^a \lambda^{a-1} \exp(-b\lambda) \quad (2.146)$$

这里 $\Gamma(a)$ 是公式 (1.141) 定义的Gamma函数, 保证了公式 (2.146) 被正确地归一化。如果 $a > 0$, 那么Gamma分布有一个有穷的积分。如果 $a \geq 1$, 那么分布本身是有穷的。图2.13给出了不同的 a 和 b 的情况下分布的图像。Gamma分布的均值和方差为

$$\mathbb{E}[\lambda] = \frac{a}{b} \quad (2.147)$$

$$\text{var}[\lambda] = \frac{a}{b^2} \quad (2.148)$$

考虑一个先验分布 $\text{Gam}(\lambda | a_0, b_0)$ 。如果我们乘以公式 (2.145) 给出的似然函数, 那么我们得到后验分布

$$p(\lambda | \mathbf{x}) \propto \lambda^{a_0-1} \lambda^{\frac{N}{2}} \exp \left\{ -b_0 \lambda - \frac{\lambda}{2} \sum_{n=1}^N (x_n - \mu)^2 \right\} \quad (2.149)$$

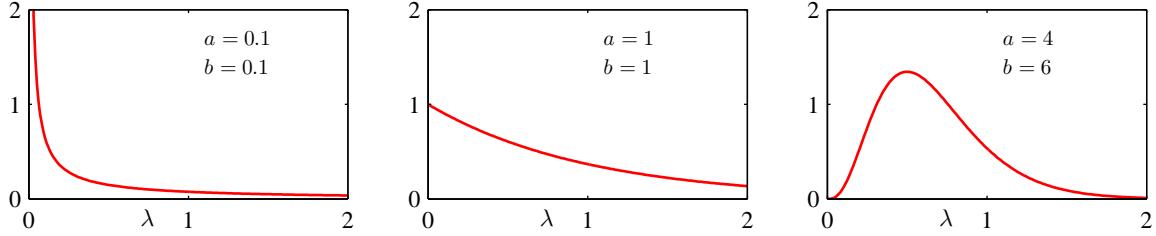


图 2.13: 对于不同的参数 a 和 b , 公式 (2.146) 定义的Gamma分布 $\text{Gam}(\lambda | a, b)$ 的图像。

我们可以把它看成形式为 $\text{Gam}(\lambda | a_N, b_N)$ 的Gamma分布, 其中

$$a_N = a_0 + \frac{N}{2} \quad (2.150)$$

$$b_N = b_0 + \frac{1}{2} \sum_{n=1}^N (x_n - \mu)^2 = b_0 + \frac{N}{2} \sigma_{ML}^2 \quad (2.151)$$

其中 σ_{ML}^2 是方差的最大似然估计。注意, 在公式 (2.149) 中, 不需要始终关注先验分布的归一化常数和似然函数, 因为如果有必要的话, 正确的系数可以在最后使用公式 (2.146) 给出的Gamma分布的表达式求出。

根据公式 (2.150), 我们看到观测 N 个数据点的效果是把系数 a 的值增加 $\frac{N}{2}$ 。因此我们可以把先验分布中的参数 a_0 看成 $2a_0$ 个“有效”先验观测。类似地, 根据公式 (2.151), 我们看到 N 个数据点对参数 b 贡献了 $\frac{N\sigma_{ML}^2}{2}$, 其中 σ_{ML}^2 是方差, 因此我们可以把先验分布中的参数 b_0 看成方差为 $\frac{2b_0}{2a_0} = \frac{b_0}{a_0}$ 的 $2a_0$ 个“有效”先验观测。回忆一下, 我们对于狄利克雷分布做过类似的表述。这些分布都是指数族分布的例子, 我们会看到, 对于指数族分布来说, 把共轭先验看成有效假想数据点是一个很通用的思想。

我们可以不使用精度进行计算, 而是考虑方差本身。这种情况下共轭先验被称为逆Gamma分布。但是我们不会详细地讨论这个分布, 因为我们发现使用精度来进行计算更加方便。

现在假设均值和精度都是未知的。为了找到共轭先验, 我们考虑似然函数对于 μ 和 λ 的依赖关系

$$\begin{aligned} p(\mathbf{x} | \mu, \lambda) &= \prod_{n=1}^N \left(\frac{\lambda}{2\pi} \right)^{\frac{1}{2}} \exp \left\{ -\frac{\lambda}{2} (x_n - \mu)^2 \right\} \\ &\propto \left[\lambda^{\frac{1}{2}} \exp \left(-\frac{\lambda\mu^2}{2} \right) \right]^N \exp \left\{ \lambda\mu \sum_{n=1}^N x_n - \frac{\lambda}{2} \sum_{n=1}^N x_n^2 \right\} \end{aligned} \quad (2.152)$$

我们现在想找到一个先验分布 $p(\mu, \lambda)$, 它对于 μ 和 λ 的依赖与似然函数有着相同的函数形式。于是我们假设先验分布的形式为

$$\begin{aligned} p(\mu, \lambda) &\propto \left[\lambda^{\frac{1}{2}} \exp \left(-\frac{\lambda\mu^2}{2} \right) \right]^\beta \exp \{ c\lambda\mu - d\lambda \} \\ &= \exp \left\{ -\frac{\beta\lambda}{2} (\mu - \frac{c}{\beta})^2 \right\} \lambda^{\frac{\beta}{2}} \exp \left\{ -\left(d - \frac{c^2}{2\beta} \right) \lambda \right\} \end{aligned} \quad (2.153)$$

其中 c, d 和 β 都是常数。由于我们总有 $p(\mu, \lambda) = p(\mu | \lambda)p(\lambda)$, 因此我们可以通过观察找到 $p(\mu | \lambda)$ 和 $p(\lambda)$ 。特别地, 我们看到 $p(\mu | \lambda)$ 是一个高斯分布, 这个高斯分布的精度是 λ 的一个线性函数。 $p(\lambda)$ 是一个Gamma分布, 因此归一化的先验概率的形式为

$$p(\mu, \lambda) = \mathcal{N}(\mu | \mu_0, (\beta\lambda)^{-1}) \text{Gam}(\lambda | a, b) \quad (2.154)$$

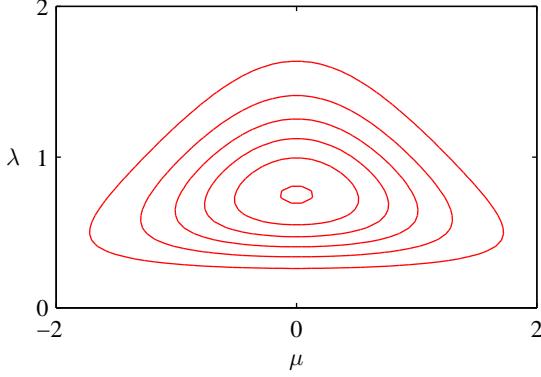


图 2.14: 公式 (2.154) 给出的正态-Gamma 分布在参数为 $\mu_0 = 0, \beta = 2, a = 5, b = 6$ 的条件下的轮廓线。

其中我们已经定义了新的常数如下: $\mu_0 = \frac{c}{\beta}, a = \frac{1+\beta}{2}, b = d - \frac{c^2}{2\beta}$ 。概率分布 (2.154) 被称为正态-Gamma 分布或者高斯-Gamma 分布, 图像如图 2.14 所示。注意这不是一个独立的 μ 的高斯分布与一个 λ 的 Gamma 分布的简单乘积, 因为 μ 的精度是 λ 的线性函数。即使我们选择一个 μ 和 λ 相互独立的先验, 后验概率中, μ 的精度和 λ 的值也会相互耦合。

对于 D 维向量 x 的多元高斯分布 $\mathcal{N}(x | \mu, \Lambda^{-1})$, 假设精度已知, 则均值 μ 的共轭先验分布仍然是高斯分布。对于已知均值未知精度矩阵 Λ 的情形, 共轭先验是 Wishart 分布, 定义为

$$\mathcal{W}(\Lambda | \mathbf{W}, \nu) = B |\Lambda|^{\frac{\nu-D-1}{2}} \exp\left(-\frac{1}{2} \text{Tr}(\mathbf{W}^{-1} \Lambda)\right) \quad (2.155)$$

其中 ν 被称为分布的自由度 degrees of freedom 数量, \mathbf{W} 是一个 $D \times D$ 的标量矩阵, $\text{Tr}(\cdot)$ 表示矩阵的迹。归一化系数 B 为

$$B(\mathbf{W}, \nu) = |\mathbf{W}|^{-\frac{\nu}{2}} \left(2^{\frac{\nu D}{2}} \pi^{\frac{D(D-1)}{4}} \prod_{i=1}^D \Gamma\left(\frac{\nu+1-i}{2}\right) \right)^{-1} \quad (2.156)$$

与之前一样, 定义协方差矩阵本身 (而不是精度) 的先验分布也可以, 这会推导出逆-Wishart 分布, 但是我们不会详细讨论这一点。如果均值和精度都是未知的, 那么类似于一元变量的推理方法, 共轭先验为

$$p(\mu, \Lambda | \mu_0, \beta, \mathbf{W}, \nu) = \mathcal{N}(\mu | \mu_0, (\beta \Lambda)^{-1}) \mathcal{W}(\Lambda | \mathbf{W}, \nu) \quad (2.157)$$

这被称为正态-Wishart 分布或者高斯-Wishart 分布。

2.3.7 学生 t 分布

我们已经看到高斯分布的精度的共轭先验是 Gamma 分布。如果我们有一个一元高斯分布 $\mathcal{N}(x | \mu, \tau^{-1})$ 和一个 Gamma 先验分布 $\text{Gam}(\tau | a, b)$, 我们把精度积分出来, 我们可以得到 x 的边缘分布, 形式为

$$\begin{aligned} p(x | \mu, a, b) &= \int_0^\infty \mathcal{N}(x | \mu, \tau^{-1}) \text{Gam}(\tau | a, b) d\tau \\ &= \int_0^\infty \frac{b^a e^{(-br)} \tau^{a-1}}{\Gamma(a)} \left(\frac{\tau}{2\pi}\right)^{\frac{1}{2}} \exp\left\{-\frac{\tau}{2}(x-\mu)^2\right\} d\tau \\ &= \frac{b^a}{\Gamma(a)} \left(\frac{1}{2\pi}\right)^{\frac{1}{2}} \left[b + \frac{(x-\mu)^2}{2}\right]^{-a-\frac{1}{2}} \Gamma(a + \frac{1}{2}) \end{aligned} \quad (2.158)$$

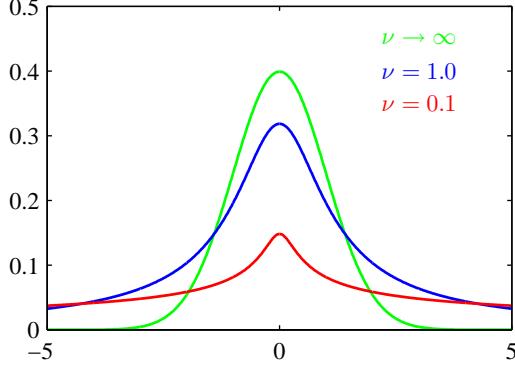


图 2.15: 对于不同的 ν 值, 公式 (2.159) 给出的学生t分布的图像, 其中 $\mu = 0$ 且 $\lambda = 1$ 。极限 $\nu \rightarrow \infty$ 对应于一个高斯分布, 均值为 μ , 精度为 λ 。

其中我们已经进行了变量替换 $z = \tau[b + \frac{(x-\mu)^2}{2}]$ 。遵循惯例, 我们定义新的参数 $\nu = 2a$ 和 $\lambda = \frac{a}{b}$ 。使用新的参数, 分布 $p(x | \mu a, b)$ 的形式为

$$St(x | \mu, \lambda, \nu) = \frac{\Gamma(\frac{\nu}{2} + \frac{1}{2})}{\Gamma(\frac{\nu}{2})} \left(\frac{\lambda}{\pi\nu} \right)^{\frac{1}{2}} \left[1 + \frac{\lambda(x-\mu)^2}{\nu} \right]^{-\frac{\nu}{2}-\frac{1}{2}} \quad (2.159)$$

这被称为学生t分布 (Student's t-distribution)。参数 λ 有时被称为t分布的精度 (precision), 即使它通常不等于方差的倒数。参数 ν 被称为自由度 (degrees of freedom), 它的作用如图2.15所示。对于 $\nu = 1$ 的情况, t分布变为了柯西分布 (Cauchy distribution), 而在极限 $\nu \rightarrow \infty$ 的情况下, t分布 $St(x | \mu, \lambda, \nu)$ 变成了高斯分布 $\mathcal{N}(x | \mu, \lambda^{-1})$, 均值为 μ , 精度为 λ 。

根据公式 (2.158), 我们看到学生t分布可以这样通过将无限多个同均值不同精度的高斯分布相加的方式得到。这可以表示为无限的高斯混合模型 (高斯混合模型将会在2.3.9节详细讨论)。结果是一个概率分布, 这个分布通常有着比高斯分布更长的“尾巴”, 正如我们在图2.15中看到的那样。这给出了t分布的一个重要性质: 鲁棒性 (robustness), 意思是对于数据集里的几个离群点outlier的出现, t分布不会像高斯分布那样敏感。t分布的鲁棒性在图2.16中说明。图中对比了高斯分布和t分布的最大似然解。注意, t分布的最大似然解可以使用期望最大化 (EM) 算法求出。这里我们看到少量的离群点对于t分布的影响要远远小于高斯分布。在实际应用中, 离群点可能产生于生成数据的过程, 这个过程对应于一个有着长尾的概率分布, 也可能产生于误标记的数据。鲁棒性也是回归问题的一个重要性质。毫不惊讶地说, 回归的最小平方的方法并不具有鲁棒性, 因为它对应于 (条件) 高斯分布下的最大似然解。通过让回归模型基于一个长尾的概率分布 (例如t分布), 我们可以得到一个更加鲁棒的模型。

如果我们回到公式 (2.158), 代入替换的参数 $\nu = 2a$, $\lambda = \frac{a}{b}$ 以及 $\eta = \frac{\tau b}{a}$, 我们看到t分布可以写成下面的形式

$$St(x | \mu, \lambda, \nu) = \int_0^\infty \mathcal{N}(x | \mu, (\eta\lambda)^{-1}) \text{Gam}(\eta | \frac{\nu}{2}, \frac{\nu}{2}) d\eta \quad (2.160)$$

之后, 我们可以把这个结果推广到多元高斯分布 $\mathcal{N}(\mathbf{x} | \boldsymbol{\mu}, \boldsymbol{\Lambda})$ 来得到对应的多元学生t分布, 形式为

$$St(\mathbf{x} | \boldsymbol{\mu}, \boldsymbol{\Lambda}, \nu) = \int_0^\infty \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}, (\eta\boldsymbol{\Lambda})^{-1}) \text{Gam}(\eta | \frac{\nu}{2}, \frac{\nu}{2}) d\nu \quad (2.161)$$

使用与一元变量相同的方法, 我们可以求出这个积分, 即

$$St(\mathbf{x} | \boldsymbol{\mu}, \boldsymbol{\Lambda}, \nu) = \frac{\Gamma(\frac{D}{2} + \frac{\nu}{2})}{\Gamma(\frac{\nu}{2})} \frac{|\boldsymbol{\Lambda}|^{\frac{1}{2}}}{(\pi\nu)^{\frac{D}{2}}} \left[1 + \frac{\Delta^2}{\nu} \right]^{-\frac{D}{2}-\frac{\nu}{2}} \quad (2.162)$$

其中 D 是 \mathbf{x} 的维度, Δ^2 是平方马氏距离, 定义为

$$\Delta^2 = (\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Lambda} (\mathbf{x} - \boldsymbol{\mu}) \quad (2.163)$$

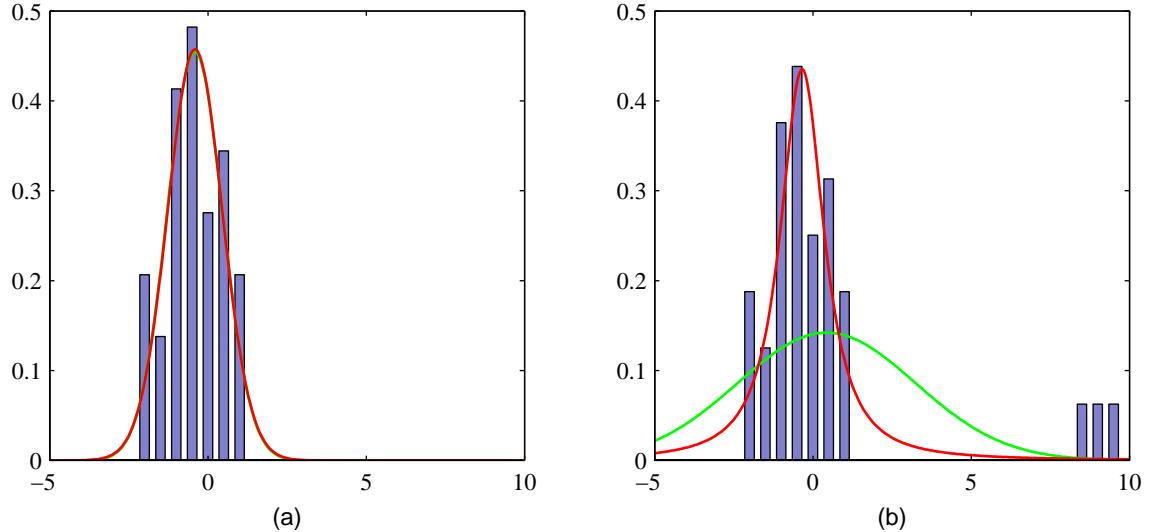


图 2.16: 与高斯分布相比, 学生t分布具有鲁棒性的例子。(a)从一个高斯分布中抽取的30个数据点的直方图, 以及得到的最大似然拟合。红色曲线表示使用t分布进行的拟合, 绿色曲线(大部分隐藏在了红色曲线后面)表示使用高斯分布进行的拟合。由于t分布将高斯分布作为一种特例, 因此它给出了与高斯分布几乎相同的解。(b)同样的数据集, 但是多了三个异常数据点。这幅图展示了高斯分布(绿色曲线)是如何被异常点强烈地干扰的, 而t分布(红色曲线)相对不受影响。

这是多元变量形式的学生t分布, 满足下面的性质

$$\mathbb{E}[\mathbf{x}] = \boldsymbol{\mu} \text{ 如果 } \nu > 1 \quad (2.164)$$

$$\text{cov}[\mathbf{x}] = \frac{\nu}{\nu - 2} \boldsymbol{\Lambda}^{-1} \text{ 如果 } \nu > 2 \quad (2.165)$$

$$\text{mode}[\mathbf{x}] = \boldsymbol{\mu} \quad (2.166)$$

对应地, 可以得到一元变量的结果。

2.3.8 周期变量

无论是高斯分布本身, 还是它作为更复杂的概率模型的基石, 高斯分布在实际应用中都非常重要。但是, 有些情况下, 对于连续变量, 使用高斯分布建模并不合适。一个重要的情况是周期变量, 这在实际应用中经常出现。

周期变量的一个例子是某个特定的地理位置的风向。例如, 我们可以测量许多天的风向值, 然后希望使用一个参数分布来总结风向的规律。另一个例子是日历时间, 其中我们可能感兴趣的是对周期为24小时或者周期为一年的变量进行建模。这种变量使用极坐标 $0 \leq \theta < 2\pi$ 表示更方便。

我们可能试图这样处理周期变量: 选择一个方向作为原点, 然后应用传统的概率分布(例如高斯分布)。但是, 这种方法的结果将会强烈依赖于原点的选择。例如, 假设我们有两个观测, 分别位于 $\theta_1 = 1^\circ$ 和 $\theta_2 = 359^\circ$, 然后我们使用一个标准的一元高斯分布建模。如果我们把原点选择为 0° , 那么这个数据集的样本均值为 180° , 标准差为 179° 。而如果我们把原点选择在 180° , 那么均值为 0° , 标准差为 1° 。很明显, 我们需要找到一种特别的方法来处理周期变量。

让我们考虑估计周期变量的观测数据集 $\mathcal{D} = \{\theta_1, \dots, \theta_N\}$ 的均值的问题。从现在开始, 我们假定 θ 的单位为弧度。我们已经看到, 简单的平均值 $\frac{\theta_1 + \dots + \theta_N}{N}$ 强烈依赖于坐标系的选择。为了找到均值的一个不变的度量, 我们注意到观测可以被看做单位圆上的点, 因此可以被描述为一个二维单位向量 $\mathbf{x}_1, \dots, \mathbf{x}_N$, 其中 $\|\mathbf{x}_n\| = 1$ 且 $n = 1, \dots, N$, 如图2.17所示。我们可以对向

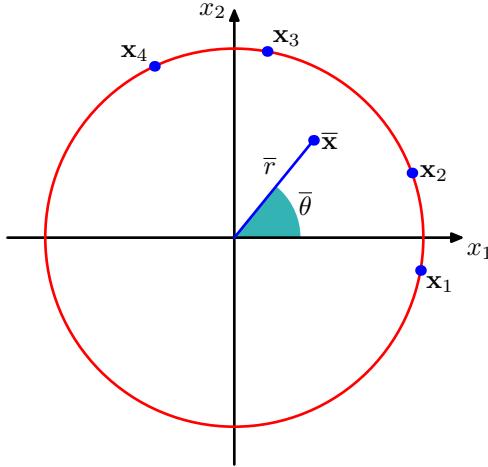


图 2.17: 将周期变量的值 θ_n 表示为单位圆上的二维向量 \mathbf{x}_n 。同时给出的还有这些向量的均值 $\bar{\mathbf{x}}$ 。

量 $\{\mathbf{x}_n\}$ 求平均，可得

$$\bar{\mathbf{x}} = \frac{1}{N} \sum_{n=1}^N \mathbf{x}_n \quad (2.167)$$

然后找到这个平均值对应的角度 $\bar{\theta}$ 。很明显，这个定义将会保证均值的位置与极坐标原点的选择无关。注意， $\bar{\mathbf{x}}$ 通常位于单位圆的内部。这个观测在笛卡尔坐标系下为 $\mathbf{x}_n = (\cos \theta_n, \sin \theta_n)$ ，我们可以把样本均值的笛卡尔坐标写成 $\bar{\mathbf{x}} = (\bar{r} \cos \bar{\theta}, \bar{r} \sin \bar{\theta})$ 。代入公式 (2.167)，然后令分量 x_1 和 x_2 相等，可得

$$\bar{x}_1 = \bar{r} \cos \bar{\theta} = \frac{1}{N} \sum_{n=1}^N \cos \theta_n, \quad \bar{x}_2 = \bar{r} \sin \bar{\theta} = \frac{1}{N} \sum_{n=1}^N \sin \theta_n \quad (2.168)$$

求两者的比值，使用恒等式 $\tan \theta = \frac{\sin \theta}{\cos \theta}$ ，我们可以求出 $\bar{\theta}$ ，即

$$\bar{\theta} = \tan^{-1} \left\{ \frac{\sum_n \sin \theta_n}{\sum_n \cos \theta_n} \right\} \quad (2.169)$$

我们稍后会看到，对于周期变量，如果恰当定义一个概率分布，最大似然方法可以很自然地得出这个结果。

我们现在考虑高斯分布对于周期变量的一个推广：von Mises分布。这里，我们应该把我们的注意力集中在一元分布，虽然周期分布也可以在任意维度的超球面中找到。对于一个关于周期分布的详细讨论，可以参考Mardia and Jupp (2000)。

感召惯例，我们考虑的周期概率分布 $p(\theta)$ 的周期为 2π 。 θ 上的任何概率密度 $p(\theta)$ 一定非负，积分等于1，并且一定是周期性的。因此， $p(\theta)$ 一定满足下面三个条件：

$$p(\theta) \geq 0 \quad (2.170)$$

$$\int_0^{2\pi} p(\theta) d\theta = 1 \quad (2.171)$$

$$p(\theta + 2\pi) = p(\theta) \quad (2.172)$$

根据公式 (2.172)，可以证明对于任意整数 M ，都有 $p(\theta + M2\pi) = p(\theta)$ 。

我们可以很容易地得到一个类似高斯的分布，满足这三个性质。考虑两个变量 $\mathbf{x} = (x_1, x_2)$ 的高斯分布，均值为 $\mu = (\mu_1, \mu_2)$ ，协方差矩阵为 $\Sigma = \sigma^2 \mathbf{I}$ ，其中 \mathbf{I} 是一个 2×2 的单位矩阵。因此

$$p(x_1, x_2) = \frac{1}{2\pi\sigma^2} \exp \left\{ -\frac{(x_1 - \mu_1)^2 + (x_2 - \mu_2)^2}{2\sigma^2} \right\} \quad (2.173)$$

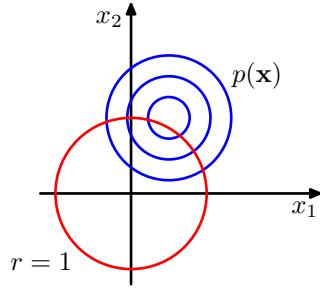


图 2.18: von Mises 分布可以通过公式 (2.173) 给出的二维高斯分布推导出来，它的密度轮廓线被画成了蓝色，概率的条件是红色的单位圆。

概率 $p(\mathbf{x})$ 为常数的轮廓线是圆形，如图 2.18 所示。现在假设我们考虑这个分布沿着一个固定半径的圆周的值。之后通过构造，这个分布将会具有周期性，虽然没有被归一化。我们可以确定这个分布的形式通过从笛卡尔坐标 (x_1, x_2) 转化为极坐标 (r, θ) 的方式得到，即

$$x_1 = r \cos \theta, \quad x_2 = r \sin \theta \quad (2.174)$$

我们也把均值 μ 映射到极坐标系中，即

$$\mu_1 = r_0 \cos \theta_0, \quad \mu_2 = r_0 \sin \theta_0 \quad (2.175)$$

接下来，我们把这些变换代入二维高斯分布 (2.173) 中，然后把分布限制在单位圆 $r = 1$ 上。注意，我们只对概率分布对于 θ 的相关性感兴趣。我们把注意力放在高斯分布的指数项上，可得

$$\begin{aligned} & -\frac{1}{2\sigma^2} \{(r \cos \theta - r_0 \cos \theta_0)^2 + (r \sin \theta - r_0 \sin \theta_0)^2\} \\ &= -\frac{1}{2\sigma^2} \{1 + r_0^2 - 2r_0 \cos \theta \cos \theta_0 - 2r_0 \sin \theta \sin \theta_0\} \\ &= \frac{r_0}{\sigma^2} \cos(\theta - \theta_0) + \text{常数} \end{aligned} \quad (2.176)$$

其中“常数”表示与 θ 无关的项，并且我们使用了下面的三角恒等式

$$\cos^2 A + \sin^2 A = 1 \quad (2.177)$$

$$\cos A \cos B + \sin A \sin B = \cos(A - B) \quad (2.178)$$

如果我们定义 $m = \frac{r_0}{\sigma^2}$ ，我们就得到了在单位圆 $r = 1$ 上的概率分布 $p(\theta)$ 的最终表达式

$$p(\theta | \theta_0, m) = \frac{1}{2\pi I_0(m)} \exp\{m \cos(\theta - \theta_0)\} \quad (2.179)$$

这被称为 von Mises 分布，或者环形正态分布 (circular normal)。这里，参数 θ_0 对应于分布的均值，而 m ，被称为 concentration 参数，类似于高斯分布的方差的倒数（精度）。公式 (2.179) 的归一化系数包含项 $I_0(m)$ ，它是零阶修正的第一类 Bessel 函数 (Abramowitz and Stegun, 1965)，定义为

$$I_0(m) = \frac{1}{2\pi} \int_0^{2\pi} \exp\{m \cos \theta\} d\theta \quad (2.180)$$

对于大的 m 值，分布逼近高斯分布。图 2.19 给出了 von Mises 分布的图像，图 2.20 给出了函数 $I_0(m)$ 的图像。

现在考虑 von Mises 分布的参数 θ_0 和参数 m 的最大似然估计。对数似然函数为

$$\ln p(\mathcal{D} | \theta_0, m) = -N \ln(2\pi) - N \ln I_0(m) + m \sum_{n=1}^N \cos(\theta_n - \theta_0) \quad (2.181)$$

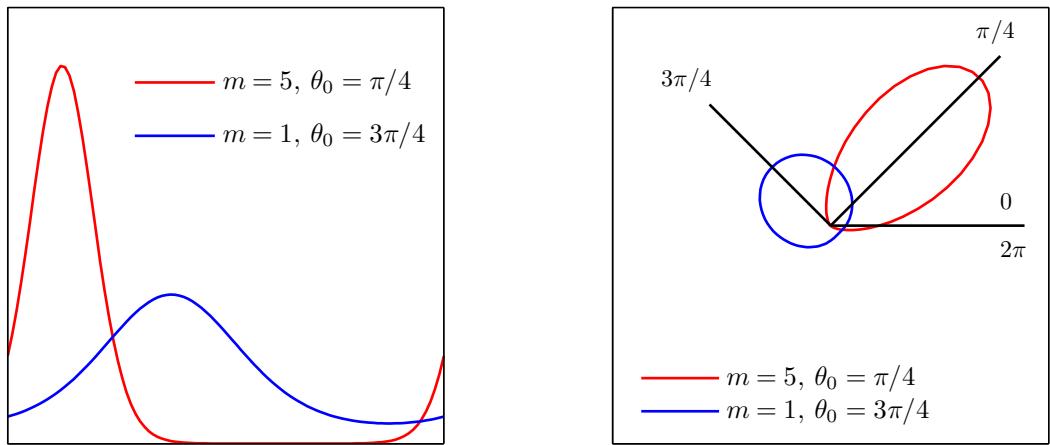


图 2.19: 对于两个不同的参数, von Mises 分布的图像。左图给出了笛卡尔坐标系中的图像, 右图给出了对应的极坐标系中的图像。

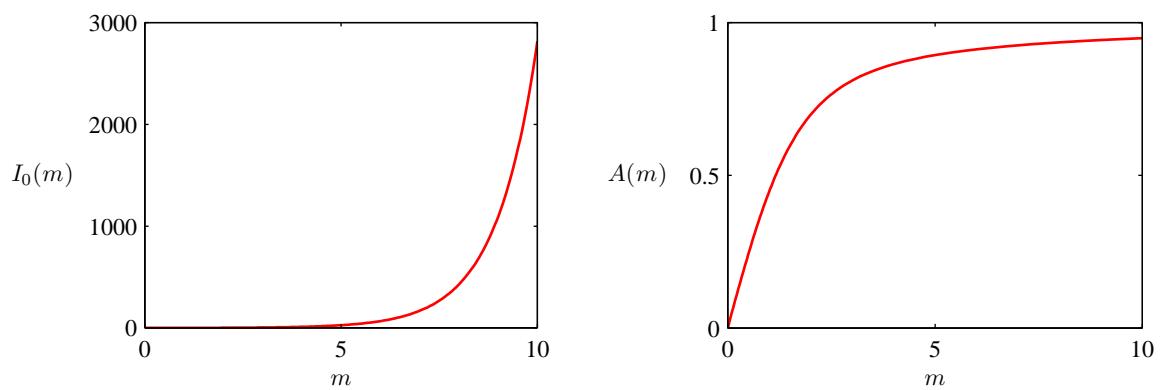


图 2.20: 公式 (2.180) 定义的Bessel函数 $I_0(m)$ 的图像, 以及公式 (2.186) 定义的函数 $A(m)$ 的图像。

令其关于 θ_0 的导数等于零，我们有

$$\sum_{n=1}^N \sin(\theta_n - \theta_0) = 0 \quad (2.182)$$

为了解出 θ_0 ，我们可以使用三角恒等式

$$\sin(A - B) = \cos B \sin A - \cos A \sin B \quad (2.183)$$

从而我们可以得到

$$\theta_0^{ML} = \tan^{-1} \left\{ \frac{\sum_n \sin \theta_n}{\sum_n \cos \theta_n} \right\} \quad (2.184)$$

这是我们之前在公式 (2.169) 中得到的结果，那里我们把它看成二维笛卡尔空间的观测的均值。

类似地，关于 m 最大化公式 (2.181)，使用 $I'_0(m) = I_1(m)$ (Abramowitz and Stegun, 1965)，我们有

$$A(m_{ML}) = \frac{1}{N} \sum_{n=1}^N \cos(\theta_n - \theta_0^{ML}) \quad (2.185)$$

其中我们已经用最大似然解 θ_0^{ML} 进行了变量替换（回忆一下我们正在关于 θ 和 m 进行联合最优化），并且我们定义

$$A(m) = \frac{I_1(m)}{I_0(m)} \quad (2.186)$$

图2.20给出了函数 $A(m)$ 的图像。使用公式 (2.178) 给出的三角恒等式，我们可以把公式 (2.185) 写成下面的形式

$$A(m_{ML}) = \left(\frac{1}{N} \sum_{n=1}^N \cos \theta_n \right) \cos \theta_0^{ML} + \left(\frac{1}{N} \sum_{n=1}^N \sin \theta_n \right) \sin \theta_0^{ML} \quad (2.187)$$

公式 (2.187) 的右侧很容易求出，并且函数 $A(m)$ 可以数值地求逆。

为了完整性，我们简要提一下其他的建立周期概率分布的方法。最简单的方法是使用观测的直方图。这种方法中，极坐标被划分成了固定大小的箱子。这种方法的优点是简洁并且灵活，但是这种方法也有着巨大的局限性。我们将在2.5节详细讨论直方图方法时看到这一点。另一种方法类似于von Mises分布，都是首先考察欧几里得空间的高斯分布。但是，这种方法在单位圆上做积分，而不是把单位圆的半径当成概率密度的条件 (Mardia and Jupp, 2000)。但是，这使得概率分布的形式更加复杂，因此我们不会详细讨论。最后一种方法的思想是，在实数轴上的任何合法的分布（例如高斯分布）都可以转化成周期分布。转化的方法是，持续地把宽度为 2π 的区间映射为周期变量 $(0, 2\pi)$ ，这相当于把实数轴沿着单位圆进行缠绕。与之前一样，与von Mises分布相比，这种方法最终求出的概率分布在计算上更加复杂。

von Mises分布的一个局限性是这个分布是单峰的。通过将多个von Mises分布混合，我们可以得到一个灵活的框架，来对能够处理多个峰值的周期变量进行建模。Lawrence et al. (2012) 给出了一个机器学习中使用了von Mises分布的例子。关于回归问题中条件概率密度的建模，可以参考Bishop and Nabney (1996)。

2.3.9 混合高斯模型

虽然高斯分布有一些重要的分析性质，但是当它遇到实际数据集时，也会有巨大的局限性。考虑图2.21给出的例子。这个数据集被称为“老忠实间歇喷泉”数据集，由美国黄石国家公园的老忠实间歇喷泉的272次喷发的测量数据组成。每条测量记录包括喷发持续了几分钟（横轴）和距离下次喷发间隔了几分钟（纵轴）。我们看到数据集主要聚集在两大堆中，一个简单的高斯分布不能描述这种结构，而两个高斯分布的线性叠加可以更好地描述这个数据集的特征。

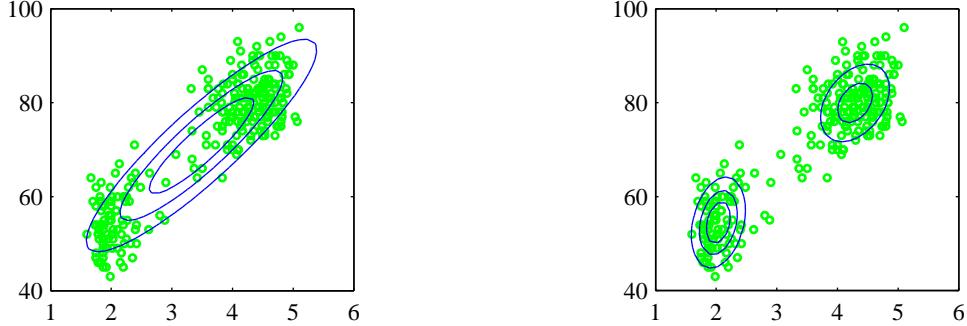


图 2.21: 老忠实间歇喷泉数据点，其中蓝色曲线给出了常数概率密度的轮廓线。左图是一个单一的高斯概率分布，已经使用最大似然法根据数据进行了调参。注意，这个概率分布未能描述数据中的两个聚集区域，并且把大部分的概率质量放在了中心区域，而这个区域的数据相对稀疏。右图是两个高斯概率分布进行线性组合得到的概率分布，已经使用第9章将要介绍的方法使用最大似然的方式根据数据进行了调参，它给出了关于数据的一个更好的表示。

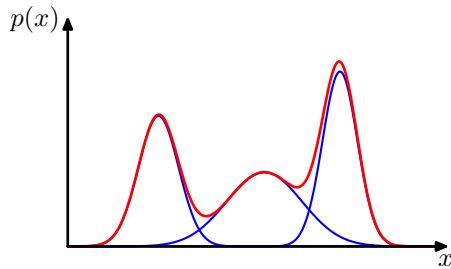


图 2.22: 一维高斯混合分布的例子。蓝色曲线给出了三个高斯分布（使用某个系数进行了缩放），红色曲线表示它们的和。

通过将更基本的概率分布（例如高斯分布）进行线性组合的这样的叠加方法，可以被形式化为概率模型，被称为混合模型（mixture distributions）（McLachlan and Basford, 1988; McLachlan and Peel, 2000）。在图2.22中，我们看到高斯分布的线性组合可以给出相当复杂的概率密度形式。通过使用足够多的高斯分布，并且调节它们的均值和方差以及线性组合的系数，几乎所有的连续概率密度都能够以任意的精度近似。

于是我们考虑 K 个高斯概率密度的叠加，形式为

$$p(\mathbf{x}) = \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \quad (2.188)$$

这被称为混合高斯（mixture of Gaussians）。每一个高斯概率密度 $\mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ 被称为混合分布的一个成分（component），并且有自己的均值 $\boldsymbol{\mu}_k$ 和协方差 $\boldsymbol{\Sigma}_k$ 。图2.23给出了具有3个成分的混合高斯分布的轮廓线和曲面。

在本节中，我们令混合模型的每个分量都是高斯分布，来说明混合模型的框架。更一般地，混合模型可以是其他类型的概率分布的线性组合。例如，在9.3.3节中，我们会考虑伯努利分布的混合，作为离散变量混合模型的一个例子。

公式 (2.188) 的参数 π_k 被称为混合系数（mixing coefficients）。如果我们对公式 (2.188) 的两侧关于 \mathbf{x} 进行积分，然后注意到 $p(\mathbf{x})$ 和各个高斯成分都是归一化的，我们可以得到

$$\sum_{k=1}^K \pi_k = 1 \quad (2.189)$$

并且，给定 $\mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_k)$ ，满足 $p(\mathbf{x}) \geq 0$ 这一要求的充分条件是对于所有的 k 都有 $\pi_k \geq 0$ 。把这个与条件 (2.189) 结合，我们有

$$0 \leq \pi_k \leq 1 \quad (2.190)$$

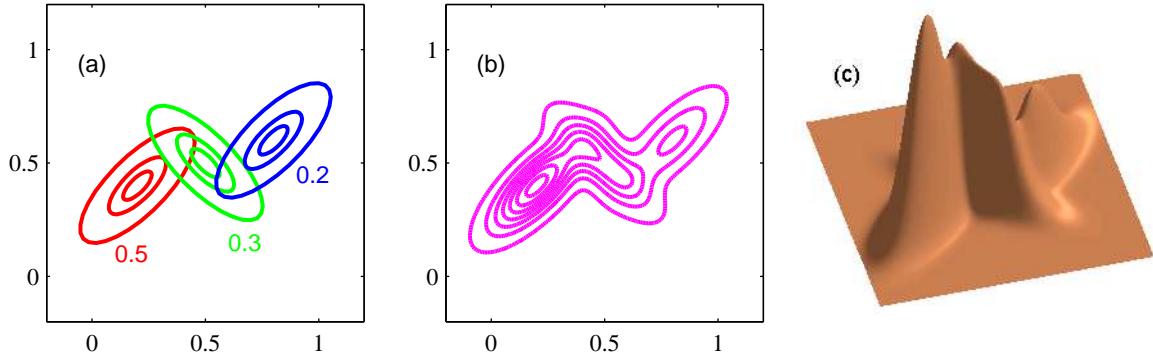


图 2.23: 二维空间中3个高斯分布混合的例子。(a)每个混合分量的常数概率密度轮廓线，其中三个分量分别被标记为红色、蓝色和绿色，且混合系数的值在每个分量的下方给出。(b)混合分布的边缘概率密度 $p(x)$ 的轮廓线。(c)概率分布 $p(x)$ 的一个曲面图。

于是我们看到混合系数满足概率的要求。

根据概率的加和规则和乘积规则，边缘概率密度为

$$p(\mathbf{x}) = \sum_{k=1}^K p(k)p(\mathbf{x} | k) \quad (2.191)$$

这等价于公式 (2.188)，其中我们把 $\pi_k = p(k)$ 看成选择第 k 个成分的先验概率，把密度 $\mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = p(\mathbf{x} | k)$ 看成以 k 为条件的 \mathbf{x} 的概率。正如我们在后面章节中将会看到的那样，后验概率 $p(k | \mathbf{x})$ 起着一个重要作用，它也被称为责任 (responsibilities)。根据贝叶斯定理，后验概率可以表示为

$$\begin{aligned} \gamma_k(\mathbf{x}) &\equiv p(k | \mathbf{x}) \\ &= \frac{p(k)p(\mathbf{x} | k)}{\sum_l p(l)p(\mathbf{x} | l)} \\ &= \frac{\pi_k \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_l \pi_l \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l)} \end{aligned} \quad (2.192)$$

我们将在第9章更加详细地讨论混合分布的概率表达。

高斯混合分布的形式由参数 $\boldsymbol{\pi}, \boldsymbol{\mu}$ 和 $\boldsymbol{\Sigma}$ 控制，其中我们令 $\boldsymbol{\pi} \equiv \{\pi_1, \dots, \pi_K\}$, $\boldsymbol{\mu} \equiv \{\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_K\}$ 且 $\boldsymbol{\Sigma} \equiv \{\boldsymbol{\Sigma}_1, \dots, \boldsymbol{\Sigma}_K\}$ 。一种确定这些参数值的方法是使用最大似然法。根据公式 (2.188)，对数似然函数为

$$\ln p(\mathbf{X} | \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^N \ln \left\{ \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\} \quad (2.193)$$

其中 $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ 。我们立刻看到现在的情形比一元高斯分布复杂得多，因为对数中存在一个求和式。这就导致参数的最大似然解不再有一个封闭形式的解析解。一种最大化这个似然函数的方法是使用迭代数值优化方法 (Fletcher, 1987; Nocedal and Wright, 1999; Bishop and Nabney, 2008)。另一种方法是使用一个被称为期望最大化 (expectation maximization) 的强大的框架，这将在第9章详细讨论。

2.4 指数族分布

我们目前为止在本章中研究的概率分布（高斯混合分布除外）都是一大类被称为指数族 (exponential family) 分布的概率分布的具体例子 (Duda and Hart, 1973; Bernardo and Smith, 1994)。指数族分布的成员有许多共同的重要性质，并且以某种程度的一般性下讨论这些性质是很有启发性的。

参数为 η 的变量 x 的指数族分布定义为具有下面形式的概率分布的集合

$$p(\mathbf{x} \mid \boldsymbol{\eta}) = h(\mathbf{x})g(\boldsymbol{\eta}) \exp\{\boldsymbol{\eta}^T \mathbf{u}(\mathbf{x})\} \quad (2.194)$$

其中 x 可能是标量或者向量，可能是离散的或者是连续的。这里 η 被称为概率分布的自然参数（natural parameters）， $\mathbf{u}(\mathbf{x})$ 是 x 的某个函数。函数 $g(\boldsymbol{\eta})$ 可以被看成系数，它确保了概率分布是归一化的，因此满足

$$g(\boldsymbol{\eta}) \int h(\mathbf{x}) \exp\{\boldsymbol{\eta}^T \mathbf{u}(\mathbf{x})\} d\mathbf{x} = 1 \quad (2.195)$$

如果 x 是离散变量，那么上式中的积分就要替换为求和。

首先，我们给出一些本章中讨论过的概率分布的例子，然后证明它们确实是指数族分布的成员。首先考虑伯努利分布

$$p(x \mid \mu) = \text{Bern}(x \mid \mu) = \mu^x(1 - \mu)^{1-x} \quad (2.196)$$

把右侧表示成指数的对数，我们有

$$\begin{aligned} p(x \mid \mu) &= \exp\{x \ln \mu + (1 - x) \ln(1 - \mu)\} \\ &= (1 - \mu) \exp\left\{\ln\left(\frac{\mu}{1 - \mu}\right)x\right\} \end{aligned} \quad (2.197)$$

与公式 (2.194) 比较，我们可以看出

$$\eta = \ln\left(\frac{\mu}{1 - \mu}\right) \quad (2.198)$$

从中我们可以解出 μ ，得到 $\mu = \sigma(\eta)$ ，其中

$$\sigma(\eta) = \frac{1}{1 + \exp(-\eta)} \quad (2.199)$$

被称为logistic sigmoid函数。因此我们可以使用公式 (2.194) 给出的标准形式把伯努利分布写成下面的形式

$$p(x \mid \mu) = \sigma(-\eta) \exp(\eta x) \quad (2.200)$$

其中我们使用了等式 $1 - \sigma(\eta) = \sigma(-\eta)$ ，这可以从公式 (2.199) 中很容易地证明出来。与公式 (2.194) 进行比较，我们有

$$\mathbf{u}(\mathbf{x}) = x \quad (2.201)$$

$$h(\mathbf{x}) = 1 \quad (2.202)$$

$$g(\eta) = \sigma(-\eta) \quad (2.203)$$

接下来考虑单一观测 x 的多项式分布，形式为

$$p(\mathbf{x} \mid \boldsymbol{\mu}) = \prod_{k=1}^M \mu_k^{x_k} = \exp\left\{\sum_{k=1}^M x_k \ln \mu_k\right\} \quad (2.204)$$

其中 $\mathbf{x} = (\mathbf{x}_1, \dots, \mathbf{x}_M)^T$ 。与之前一样，我们可以把它写成公式 (2.194) 的标准形式，即

$$p(\mathbf{x} \mid \boldsymbol{\mu}) = \exp(\boldsymbol{\eta}^T \mathbf{x}) \quad (2.205)$$

其中 $\mu_k = \ln \mu_k$ ，并且我们定义了 $\boldsymbol{\eta} = (\eta_1, \dots, \eta_M)^T$ 。再次与公式 (2.194) 比较，我们有

$$\mathbf{u}(\mathbf{x}) = \mathbf{x} \quad (2.206)$$

$$h(\mathbf{x}) = 1 \quad (2.207)$$

$$g(\boldsymbol{\eta}) = 1 \quad (2.208)$$

注意参数 η_k 不是相互独立的，因为参数 μ_k 要满足下面的限制

$$\sum_{k=1}^M \mu_k = 1 \quad (2.209)$$

因此给定任意 $M - 1$ 个参数 μ_k ，剩下的参数就固定了。在某些情况下，去掉这个限制比较方便。此时，我们只用 $M - 1$ 个参数来表示这个分布。我们可以这样做：使用公式 (2.209) 的关系，把 μ_M 用剩余的 $\{\mu_k\}$ 表示，其中 $k = 1, \dots, M - 1$ ，这样就只剩下了 $M - 1$ 个参数。注意，剩余的参数仍然满足下面的限制

$$0 \leq \mu_k \leq 1, \quad \sum_{k=1}^{M-1} \mu_k \leq 1 \quad (2.210)$$

使用公式 (2.209) 给出的限制，这种表达方式下多项式分布变成了

$$\begin{aligned} & \exp \left\{ \sum_{k=1}^M x_k \ln \mu_k \right\} \\ &= \exp \left\{ \sum_{k=1}^{M-1} x_k \ln \mu_k + \left(1 - \sum_{k=1}^{M-1} x_k \right) \ln \left(1 - \sum_{k=1}^{M-1} \mu_k \right) \right\} \\ &= \exp \left\{ \sum_{k=1}^{M-1} x_k \ln \left(\frac{\mu_k}{1 - \sum_{j=1}^{M-1} \mu_j} \right) + \ln \left(1 - \sum_{k=1}^{M-1} \mu_k \right) \right\} \end{aligned} \quad (2.211)$$

我们现在令

$$\ln \left(\frac{\mu_k}{1 - \sum_j \mu_j} \right) = \eta_k \quad (2.212)$$

从中我们可以解出 μ_k 。首先两侧对 k 求和，然后整理，回带，可得

$$\mu_k = \frac{\exp(\eta_k)}{1 + \sum_j \exp(\eta_j)} \quad (2.213)$$

这被称为 softmax 函数，或者归一化指数（normalized exponential）。在这个表达方式的形式下，多项式分布的形式为

$$p(\mathbf{x} | \boldsymbol{\eta}) = \left(1 + \sum_{k=1}^{M-1} \exp(\eta_k) \right)^{-1} \exp(\boldsymbol{\mu}^T \mathbf{x}) \quad (2.214)$$

这是指数族分布的标准形式，其中参数向量 $\boldsymbol{\eta} = (\eta_1, \dots, \eta_{M-1}, 0)^T$ 。在这个指数族分布中

$$\boldsymbol{u}(\mathbf{x}) = \mathbf{x} \quad (2.215)$$

$$h(\mathbf{x}) = 1 \quad (2.216)$$

$$g(\boldsymbol{\eta}) = \left(1 + \sum_{k=1}^{M-1} \exp(\eta_k) \right)^{-1} \quad (2.217)$$

最后，让我们考虑高斯分布。对于一元高斯分布，我们有

$$p(x | \mu, \sigma^2) = \frac{1}{(2\pi\sigma^2)^{\frac{1}{2}}} \exp \left\{ -\frac{1}{2\sigma^2}(x - \mu)^2 \right\} \quad (2.218)$$

$$= \frac{1}{(2\pi\sigma^2)^{\frac{1}{2}}} \exp \left\{ -\frac{1}{2\sigma^2}x^2 + \frac{\mu}{\sigma^2}x - \frac{1}{2\sigma^2}\mu^2 \right\} \quad (2.219)$$

在经过一些简单的推导后，它可以转化为公式 (2.194) 给出的标准指数族分布的形式，其中

$$\boldsymbol{\eta} = \begin{pmatrix} \frac{\mu}{\sigma^2} \\ \frac{-1}{2\sigma^2} \end{pmatrix} \quad (2.220)$$

$$\mathbf{u}(x) = \begin{pmatrix} x \\ x^2 \end{pmatrix} \quad (2.221)$$

$$h(x) = (2\pi)^{-\frac{1}{2}} \quad (2.222)$$

$$g(\boldsymbol{\eta}) = (-2\eta_2)^{\frac{1}{2}} \exp\left(\frac{\eta_1^2}{4\eta_2}\right) \quad (2.223)$$

2.4.1 最大似然与充分统计量

让我们考虑用最大似然法估计公式 (2.194) 给出的一般形式的指数族分布的参数向量 $\boldsymbol{\mu}$ 的问题。对公式 (2.195) 的两侧关于 $\boldsymbol{\mu}$ 取梯度，我们有

$$\begin{aligned} & \nabla g(\boldsymbol{\eta}) \int h(\mathbf{x}) \exp\{\boldsymbol{\eta}^T \boldsymbol{\mu}(\mathbf{x})\} d\mathbf{x} \\ & + g(\boldsymbol{\eta}) \int h(\mathbf{x}) \exp\{\boldsymbol{\eta}^T \boldsymbol{\mu}(\mathbf{x})\} \mathbf{u}(\mathbf{x}) d\mathbf{x} = 0 \end{aligned} \quad (2.224)$$

重新排列各项，然后再次使用公式 (2.195)，可得

$$-\frac{1}{g(\boldsymbol{\eta})} \nabla g(\boldsymbol{\eta}) = g(\boldsymbol{\eta}) \int h(\mathbf{x}) \exp\{\boldsymbol{\eta}^T \mathbf{u}(\mathbf{x})\} \mathbf{u}(\mathbf{x}) d\mathbf{x} = \mathbb{E}[\mathbf{u}(\mathbf{x})] \quad (2.225)$$

于是我们可得

$$-\nabla \ln g(\boldsymbol{\eta}) = \mathbb{E}[\mathbf{u}(\mathbf{x})] \quad (2.226)$$

注意， $\mathbf{u}(\mathbf{x})$ 的协方差可以根据 $g(\boldsymbol{\eta})$ 的二阶导数表达，对于高阶矩的情形也类似。因此，如果我们能够对一个来自指数族分布的概率分布进行归一化，那么我们总能够通过简单的求微分的方式找到它的矩。

现在考虑一组独立同分布的数据 $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ 。对于这个数据集，似然函数为

$$p(\mathbf{X} | \boldsymbol{\eta}) = \left(\prod_{n=1}^N h(\mathbf{x}_n) \right) g(\boldsymbol{\eta})^N \exp \left\{ \boldsymbol{\eta}^T \sum_{n=1}^N \mathbf{u}(\mathbf{x}_n) \right\} \quad (2.227)$$

令 $\ln p(\mathbf{X} | \boldsymbol{\eta})$ 关于 $\boldsymbol{\eta}$ 的导数等于零，我们可以得到最大似然估计 $\boldsymbol{\mu}_{ML}$ 满足的条件

$$-\nabla \ln g(\boldsymbol{\eta}_{ML}) = \frac{1}{N} \sum_{n=1}^N \mathbf{u}(\mathbf{x}_n) \quad (2.228)$$

原则上可以通过解这个方程来得到 $\boldsymbol{\mu}_{ML}$ 。我们看到最大似然估计的解只通过 $\sum_n \mathbf{u}(\mathbf{x}_n)$ 对数据产生依赖，因此这个量被称为分布 (2.194) 的充分统计量 (sufficient statistic)。我们不需要存储整个数据集本身，只需要存储充分统计量的值即可。例如，对于伯努利分布，函数 $\mathbf{u}(x)$ 就等于 x ，因此我们只需要存储数据点 $\{\mathbf{x}_n\}$ 的和即可。而对于高斯分布， $\mathbf{u}(x) = (x, x^2)^T$ ，因此我们应该同时存储 $\{\mathbf{x}_n\}$ 的和以及 $\{\mathbf{x}_n^2\}$ 的和。

如果我们考虑极限 $N \rightarrow \infty$ ，那么公式 (2.228) 的右侧变成了 $[\mathbf{u}(\mathbf{x})]$ ，因此通过与公式 (2.226) 比较，我们可以看到在这个极限的情况下， $\boldsymbol{\eta}_{ML}$ 与真实值 $\boldsymbol{\eta}$ 相等。

实际上，这种充分性对于贝叶斯推断也成立，但是我们要把关于这一点的讨论推迟到第8章。那时，我们已经有了图模型的知识，因此能够更深刻地理解这些重要的概念。

2.4.2 共轭先验

我们已经多次遇到共轭先验的概念。例如在伯努利分布中，共轭先验是Beta分布。在高斯分布中，均值的共轭先验是高斯分布，精度的共轭先验是Wishart分布。一般情况下，对于一个给定的概率分布 $p(\boldsymbol{x} | \boldsymbol{\mu})$ ，我们能够寻找一个先验 $p(\boldsymbol{\eta})$ 使其与似然函数共轭，从而后验分布的函数形式与先验分布相同。对于指数族分布（2.194）的任何成员，都存在一个共轭先验，可以写成下面的形式

$$p(\boldsymbol{\eta} | \boldsymbol{\chi}, \nu) = f(\boldsymbol{\chi}, \nu) g(\boldsymbol{\eta})^\nu \exp\{\nu \boldsymbol{\eta}^T \boldsymbol{\chi}\} \quad (2.229)$$

其中 $f(\boldsymbol{\chi}, \nu)$ 是归一化系数， $g(\boldsymbol{\eta})$ 与公式（2.194）中的含义相同。为了证明这个确实是共轭先验，让我们把先验分布（2.229）与似然函数（2.227）相乘，得到后验概率（忽略归一化系数），形式为

$$p(\boldsymbol{\eta} | \boldsymbol{X}, \boldsymbol{\chi}, \nu) \propto g(\boldsymbol{\eta})^{\nu+N} \exp\left\{\boldsymbol{\eta}^T \left(\sum_{n=1}^N \boldsymbol{u}(\boldsymbol{x}_n) + \nu \boldsymbol{\chi}\right)\right\} \quad (2.230)$$

这再次与先验分布（2.229）取得了相同的函数形式，从而证明了共轭性。此外，我们看到参数 ν 可以看成先验分布中假想观测的有效观测数。给定 $\boldsymbol{\chi}$ 的情况下，每个假想观测都对充分统计量 $\boldsymbol{u}(\boldsymbol{x})$ 的值有贡献。

2.4.3 无信息先验

在某些概率推断的应用中，我们可能有一些先验知识，可以方便地通过先验概率分布表达出来。例如，如果先验分布令变量的某些值的概率为零，那么后验分布也将会使那些值的概率为零，与后续的数据观测无关。但是，在许多情形下，我们可能对分布应该具有的形式几乎完全不知道。这时，我们可以寻找一种形式的先验分布，被称为无信息先验（noninformative prior）。这种先验分布的目的是尽量对后验分布产生尽可能小的影响（Jeffreys, 1946; Box and Tiao, 1973; Bernardo and Smith, 1994）。这有时被称为“让数据自己说话”。

如果我们有一个由参数 λ 控制的分布 $p(x | \lambda)$ ，那么我们可以尝试假设先验分布 $p(\lambda) = \text{常数}$ 作为一个合适的先验分布。如果 λ 是一个有 K 个状态的离散变量，这就相当于把每种状态的先验概率设置为 $\frac{1}{K}$ 。然而，在连续参数的情况下，这种方法有两个潜在的困难。第一个困难是，如果 λ 的取值范围是无界的，那么先验分布无法被正确地归一化，因为对 λ 的积分是发散的。这样的先验分布被称作反常的（improper）。实际应用中，如果对应的后验分布是正常的（proper），即它可以正确地被归一化，那么可以使用反常先验分布。例如，如果我们假设高斯分布的均值的先验分布为均匀分布，那么一旦我们观测到至少一个数据点，均值的后验分布就会是正常的。

第二个困难产生于概率非线性变量的概率密度的变换，由公式（1.27）给出。如果函数 $h(\lambda)$ 是常数，并且我们进行变量替换 $\lambda = \eta^2$ ，那么 $h(\eta) = h(\eta^2)$ 也会是常数。然而，如果我们令概率密度 $p_\lambda(\lambda)$ 为常数，那么根据公式（1.27）， η 的概率密度为

$$p_\eta(\eta) = p_\lambda(\lambda) \left| \frac{d\lambda}{d\eta} \right| = p_\lambda(\eta^2) 2\eta \propto \eta \quad (2.231)$$

从而 η 的概率密度就不再是常数了。如果我们使用最大似然估计，那么就不会有这种问题，因为似然函数 $p(\boldsymbol{x} | \lambda)$ 是 λ 的一个简单的函数，因此我们可以自由使用任意方便的对参数操作的方法。但是，如果我们要选择一个常数的先验概率分布，那么我们必须注意对于参数要使用一个合适的表达形式。

这里我们考虑无信息先验的两个简单的例子（Berger, 1985）。首先，如果概率密度的形式为

$$p(x | \mu) = f(x - \mu) \quad (2.232)$$

那么参数 μ 被称为位置参数（location parameter）。这一类概率分布具有平移不变性（translation invariance），因为如果我们把 x 平移一个常数，得到 $\hat{x} = x + c$ ，那么

$$p(\hat{x} | \hat{\mu}) = f(\hat{x} - \hat{\mu}) \quad (2.233)$$

其中我们已经定义 $\hat{\mu} = \mu + c$ 。因此新变量的概率密度的形式与原变量相同，因此概率密度与原点的选择无关。我们想要选择一个能够反映这种平移不变性的先验分布，因此我们选择的先验概率分布要对区间 $A \leq \mu \leq B$ 以及平移后的区间 $A - c \leq \mu \leq B - c$ 赋予相同的概率质量。这说明

$$\int_A^B p(\mu) d\mu = \int_{A-c}^{B-c} p(\mu) d\mu = \int_A^B p(\mu - c) d\mu \quad (2.234)$$

并且由于这必须对于任意的 A 和 B 的选择都成立，因此我们有

$$p(\mu - c) = p(\mu) \quad (2.235)$$

这表明 $p(\mu)$ 是常数。位置参数的一个例子是高斯分布的均值 μ 。正如我们已经看到的那样，这种情况下 μ 的共轭先验分布是一个高斯分布 $p(\mu | \mu_0, \sigma_0^2) = \mathcal{N}(\mu | \mu_0, \sigma_0^2)$ ，并且通过取极限 $\sigma_0^2 \rightarrow \infty$ ，我们得到了一个无信息先验。事实上，根据公式 (2.141) 和公式 (2.142)，我们可以看到这种极限情况下，在 μ 的后验分布中，先验的贡献消失了。

作为第二个例子，考虑概率分布的形式为

$$p(x | \sigma) = \frac{1}{\sigma} f\left(\frac{x}{\sigma}\right) \quad (2.236)$$

其中 $\sigma > 0$ 。注意，如果 $f(x)$ 被正确归一化，那么这是一个归一化的概率密度。参数 σ 被称为缩放参数 (scale parameter)，概率密度具有缩放不变性 (scale invariance) 因为如果我们把 x 缩放一个常数，得到 $\hat{x} = cx$ ，那么

$$p(\hat{x} | \hat{\sigma}) = \frac{1}{\hat{\sigma}} f\left(\frac{\hat{x}}{\hat{\sigma}}\right) \quad (2.237)$$

其中我们已经定义了 $\hat{\sigma} = c\sigma$ 。这个变换对应于单位的改变。例如如果 x 表示长度，那么这个变换可能从“米”变为“千米”。我们希望选择一个能够反映这种缩放不变性的先验分布。如果我们考虑一个区间 $A \leq \sigma \leq B$ ，以及一个缩放的区间 $\frac{A}{c} \leq \sigma \leq \frac{B}{c}$ ，那么先验分布应该给这两个区间赋予相同的概率质量。因此我们有

$$\int_A^B p(\sigma) d\sigma = \int_{\frac{A}{c}}^{\frac{B}{c}} p(\sigma) d\sigma = \int_A^B p\left(\frac{1}{c}\sigma\right) \frac{1}{c} d\sigma \quad (2.238)$$

由于这必须对于任意的 A 和 B 的选择都成立，因此我们有

$$p(\sigma) = p\left(\frac{1}{c}\sigma\right) \frac{1}{c} \quad (2.239)$$

因此 $p(\sigma) \propto \frac{1}{\sigma}$ 。注意，这是一个反常先验分布，因为对于 $0 \leq \sigma \leq \infty$ 上的概率分布的积分是发散的。有时把缩放参数的先验分布用参数的对数的概率密度表达更方便。使用公式 (1.27) 的概率密度变换规则，我们看到 $p(\ln \sigma) = \text{常数}$ 。因此，对于这个先验分布，在区间 $1 \leq \sigma \leq 10$ 和区间 $10 \leq \sigma \leq 100$ 以及区间 $100 \leq \sigma \leq 1000$ 上具有相同的概率质量。

缩放参数的一个例子是高斯分布的标准差 σ ，在我们考虑了位置参数 μ 之后。这是因为

$$\mathcal{N}(x | \mu, \sigma^2) \propto \sigma^{-1} \exp\left\{-\left(\frac{\tilde{x}}{\sigma}\right)^2\right\} \quad (2.240)$$

其中 $\tilde{x} = x - \mu$ 。正如之前讨论过的那样，通常更方便的做法是用精度 $\lambda = \frac{1}{\sigma^2}$ 计算，而不是用 σ 本身。使用概率密度的变换规则，我们看到一个概率密度 $p(\sigma) \propto \frac{1}{\sigma}$ 对应于 λ 上的形式为 $p(\lambda) \propto \frac{1}{\lambda}$ 的概率分布。我们已经看到， λ 的共轭先验是公式 (2.146) 给出的 Gamma 分布 $\text{Gam}(\lambda | a_0, b_0)$ 。无信息先验在 $a_0 = b_0 = 0$ 的特殊情况下得到。与之前一样，如果我们检查公式 (2.150) 和公式 (2.151) 给出的 λ 的后验概率分布的结果，我们看到对于 $a_0 = b_0 = 0$ ，后验分布只与数据相关，而与先验分布无关。

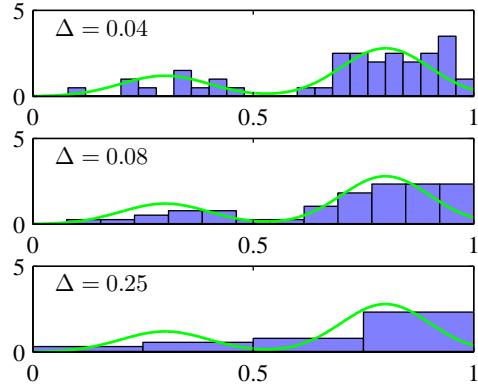


图 2.24: 直方图方法用于密度估计的一个例子，其中，50个数据点组成的数据集由绿色曲线代表的概率分布生成。方法基于的是公式 (2.241)，公用的箱宽度 Δ 在图上已经标出。图中给出了不同的 Δ 取值所对应的情形。

2.5 非参数化方法

本章中，我们已经关注过的概率分布都有具体的函数形式，并且由少量的参数控制。这些参数的值可以由数据集确定。这被称为概率密度建模的参数化 (parametric) 方法。这种方法的一个重要局限性是选择的概率密度可能对于生成数据来说，是一个很差的模型，从而会导致相当差的预测表现。流入，如果生成数据的过程是多峰的，那么这种分布不可能被高斯分布描述，因为它是单峰的。

在最后一节，我们考虑一些非参数化 (nonparametric) 方法进行概率密度估计。这种方法对概率分布的形式进行了很少的假设。这里，我们把注意力集中于简单的频率学家方法。但是，读者应该意识到，非参数化贝叶斯方法正在吸引越来越多的研究者的兴趣 (Walker et al., 1999; Neal, 2000; Müller and Quintana, 2004; Teh et al., 2006)。

首先让我们讨论密度估计的直方图方法。这种方法我们之前已经使用过。例如，图1.11的边缘分布和条件分布，以及图2.6的中心极限定理。这里，我们更加详细地探索直方图密度估计的性质。我们集中于一元连续变量 x 的情形。标准的直方图简单地把 x 划分成不同的宽度为 Δ_i 的箱子，然后对落在第 i 个箱子中的 x 的观测数量 n_i 进行计数。为了把这种计数转换成归一化的概率密度，我们简单地把观测数量除以观测的总数 N ，再除以箱子的宽度 Δ_i ，得到每个箱子的概率的值

$$p_i = \frac{n_i}{N\Delta_i} \quad (2.241)$$

从中很容易看出 $\int p(x) dx = 1$ 。这给出了概率密度 $p(x)$ 的一个模型，这个概率密度在每个箱子的宽度内是常数，并且通常箱子的宽度选成相同的，即 $\Delta_i = \Delta$ 。

在图2.24中，我们给出了一个直方图概率密度估计的例子。这里数据满足绿色曲线的概率分布，它由两个高斯分布混合而成。同时给出的还有三个直方图密度估计的例子，分布对应于箱子宽度 Δ 的三种不同的选择。我们看到，当 Δ 非常小的时候（最上方的图），最终的概率密度模型有很多尖刺，有很多结构没有出现在生成数据的概率分布中。相反，如果 Δ 过大（最下方的图），那么最终的概率模型会过于平滑，结果无法描述绿色曲线的双峰性质。当 Δ 取一个中等大小的值时（中间的图），可以得到最好的结果。原则上，一个直方图概率密度模型也依赖于箱子边缘位置的选择，但是这对于结果的影响通常会小于 Δ 的值的选择。

注意，与之前讨论过的方法不同，直方图方法具有下面的性质：一旦直方图被计算出来，数据本身就被丢弃了，这当数据量很大的时候会很有优势。并且，直方图方法也很容易应用到数据顺序到达的情形。

在实际应用中，直方图方法对于快速地将一维或者二维的数据可视化很有用，但是并不适用于大多数概率密度估计的应用。一个明显的问题是估计的概率密度具有不连续性，这种不连续性是因为箱子的边缘造成的，而不是因为生成数据的概率分布本身的性质造成。直方图方法的另一个主要的局限性是维数放大。如果我们把 D 维空间的每一维的变量都划分到 M 个箱子中，

那么箱子的总数为 M^D 。这种对于 D 的指数放大是维度灾难的一个例子。在高维空间中，如果想对于局部概率密度进行有意义的估计，那么需要的数据量是不可接受的。

但是，概率密度估计的直方图方法确实告诉了我们两个重要的事情。第一，为了估计在某个特定位置的概率密度，我们应该考虑位于那个点的某个邻域内的数据点。注意，局部性的概念要求我们假设某种形式的距离度量，这里我们假设的是欧几里得距离。对于直方图，这种邻域的性质由箱子定义，并且有一个自然的“平滑”参数描述局部区域的空间扩展，即这里的箱子宽度。第二，为了获得好的结果，平滑参数的值既不能太大也不能太小。这让我们回忆起了第1章讨论过的多项式曲线拟合问题中对于模型复杂度的选择，那里多项式的阶数 M 或者正则化参数 α ，被优化成了某些中等大小的值，既不太大也不太小。有了这些认识，现在让我们讨论两个广泛使用的密度估计的非参数化方法，核估计以及近邻估计。与简单的直方图方法相比，这两种方法对于维度的放大有着更好的适应性。

2.5.1 核密度估计

让我们假设观测服从 D 维空间的某个未知的概率密度分布 $p(\mathbf{x})$ 。我们把这个 D 维空间选择成欧几里得空间，并且我们想估计 $p(\mathbf{x})$ 的值。根据我们之前对于局部性的讨论，让我们考虑包含 \mathbf{x} 的某个小区域 \mathcal{R} 。这个区域的概率质量为

$$P = \int_{\mathcal{R}} p(\mathbf{x}) \, d\mathbf{x} \quad (2.242)$$

现在我们假设我们收集了服从 $p(\mathbf{x})$ 分布的 N 次观测。由于每个数据点都有一个落在区域 \mathcal{R} 中的概率 P ，因此位于区域 \mathcal{R} 内部的数据点的总数 K 将服从二项分布

$$\text{Bin}(K \mid N, P) = \frac{N!}{K!(N-K)!} P^K (1-P)^{N-K} \quad (2.243)$$

使用公式 (2.11)，我们看到落在区域内部的数据点的平均比例为 $\mathbb{E}[\frac{K}{N}] = P$ 。类似地，使用公式 (2.12)，我们看到，以此为均值的概率分布的方差为 $\text{var}[\frac{K}{N}] = \frac{P(1-P)}{N}$ 。对于大的 N 值，这个分布将会在均值附近产生尖峰，并且

$$K \simeq NP \quad (2.244)$$

但是，如果我们也假定区域 \mathcal{R} 足够小，使得在这个区域内的概率密度 $p(\mathbf{x})$ 大致为常数，那么我们有

$$P \simeq p(\mathbf{x})V \quad (2.245)$$

其中 V 是区域 \mathcal{R} 的体积。把公式 (2.244) 和公式 (2.245) 结合，我们得到概率密度的估计，形式为

$$p(\mathbf{x}) = \frac{K}{NV} \quad (2.246)$$

注意，公式 (2.246) 的成立依赖于两个相互矛盾的假设，即区域 \mathcal{R} 要足够小，使得这个区域内的概率密度近似为常数，但是也要足够大，使得落在这个区域内的数据点的数量 K 能够足够让二项分布达到尖峰。

我们有两种方式利用 (2.246) 的结果。我们可以固定 K 然后从数据中确定 V 的值，这就是 K 近邻方法。我们还可以固定 V 然后从数据中确定 K ，这就是核方法。在极限 $N \rightarrow \infty$ 的情况下，如果 V 随着 N 而合适地收缩，并且 K 随着 N 增大，那么可以证明 K 近邻概率密度估计和核方法概率密度估计都会收敛到真实的概率密度 (Duda and Hart, 1973)。

我们先详细讨论核方法。首先，我们把区域 \mathcal{R} 取成以 \mathbf{x} 为中心的小超立方体，我们想确定概率密度。为了统计落在这个区域内的数据点的数量 K ，定义下面的函数比较方便

$$k(\mathbf{u}) = \begin{cases} 1, & |u_i| \leq \frac{1}{2}, \quad i = 1, \dots, D, \\ 0, & \text{其他情况} \end{cases} \quad (2.247)$$

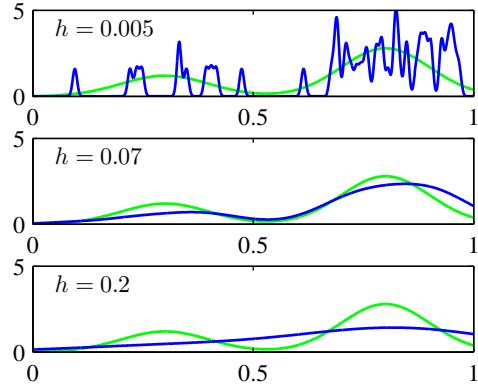


图 2.25: 公式 (2.250) 给出的核密度模型的例子。数据集与图2.24中用于说明直方图方法时使用的数据集相同。我们看到, h 的作用相当于平滑参数, 如果它被设置得过小 (最上方的图), 结果是一个噪声非常大的概率模型, 而如果它被设置得过大 (最下方的图), 那么用于生成数据的概率分布 (绿色曲线表示) 的双峰性质被抹去了。 h 取某个中等大小的值时, 可以得到最好的密度模型 (中间的图)。

这表示一个以原点为中心的单位立方体。函数 $k(\mathbf{u})$ 是核函数 (kernel function) 的一个例子, 在这个问题中也被称为 Parzen 窗 (Parzen window)。根据公式 (2.247), 如果数据点 \mathbf{x}_n 位于以 \mathbf{x} 为中心的边长为 h 的立方体中, 那么量 $k\left(\frac{\mathbf{x}-\mathbf{x}_n}{h}\right)$ 的值等于 1, 否则它的值为 0。于是, 位于这个立方体内的数据点的总数为

$$K = \sum_{n=1}^N k\left(\frac{\mathbf{x}-\mathbf{x}_n}{h}\right) \quad (2.248)$$

把这个表达式代入公式 (2.246), 可以得到点 \mathbf{x} 处的概率密度估计

$$p(\mathbf{x}) = \frac{1}{N} \sum_{n=1}^N \frac{1}{h^D} k\left(\frac{\mathbf{x}-\mathbf{x}_n}{h}\right) \quad (2.249)$$

推导过程中我们使用了 D 维边长为 h 的立方体的体积公式 $V = h^D$ 。使用函数 $k(\mathbf{u})$ 的对称性, 我们现在可以重新表述这个方程。之前我们把这个函数表述为以 \mathbf{x} 为中心的一个立方体, 但是现在我们把这个函数表述为以 N 个数据点 \mathbf{x}_n 为中心的 N 个立方体。

核密度估计 (2.249) 有一个问题, 这个问题也是直方图方法具有的问题中的一个。这个问题就是人为带来的非连续性。在之前所述的核密度估计方法中就是立方体的边界。如果我们选择一个平滑的核函数, 那么我们就可以得到一个更加光滑的模型。一个常见的选择是高斯核函数。使用高斯核函数, 可以得到下面的核概率密度模型

$$p(\mathbf{x}) = \frac{1}{N} \sum_{n=1}^N \frac{1}{(2\pi h^2)^{\frac{D}{2}}} \exp\left\{-\frac{\|\mathbf{x}-\mathbf{x}_n\|^2}{2h^2}\right\} \quad (2.250)$$

其中 h 表示高斯分布的标准差。因此我们的概率密度模型可以通过这种方式获得: 令每个数据点都服从高斯分布, 然后把数据集里的每个数据点的贡献相加, 之后除以 N , 使得概率密度正确地被归一化。在图2.25中, 我们把模型 (2.250) 应用于之前用来说明直方图方法的数据集上。我们看到, 正如我们期望的那样, 参数 h 对平滑参数起着重要的作用。小的 h 会造成模型对噪声过于敏感, 而大的 h 会造成过度平滑, 因此要进行一个折中。与之前一样, 对 h 的优化是一个模型复杂度的问题, 类似于直方图概率密度估计中对于箱子狂赌的选择, 也类似于曲线拟合问题中的多项式阶数。

我们可以任意选择公式 (2.249) 中的核函数, 只要满足下面的条件

$$k(\mathbf{u}) \geq 0 \quad (2.251)$$

$$\int k(\mathbf{u}) d\mathbf{u} = 1 \quad (2.252)$$

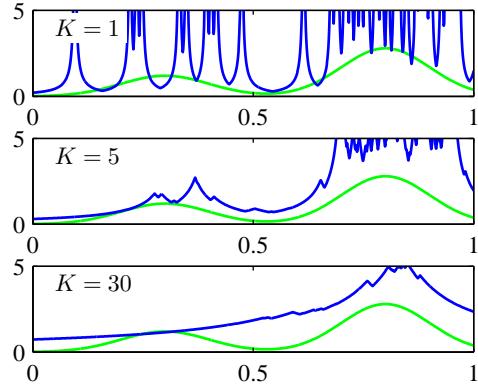


图 2.26: 使用与图2.24和图2.25相同的数据集, 进行 K 近邻密度估计的例子。我们看到参数 K 控制了平滑程度, 因此一个小的 K 值会产生一个噪声相当大的密度模型 (最上方的图), 而一个大的 K 值 (最下方的图) 平滑掉了用于生成数据的真实概率分布 (绿色曲线) 的双峰性质。

这确保了最终求得的概率分布在处处都是非负的, 并且积分等于1。公式 (2.249) 给出的概率密度模型被称为核密度估计, 或者Parzen估计。它由一个很大的优点, 即不需要进行“训练”阶段的计算, 因为“训练”阶段只需要存储训练集即可。然而, 这也是一个巨大的缺点, 因为估计概率密度的计算代价随着数据集的规模线性增长。

2.5.2 近邻方法

核方法进行概率密度估计的一个困难之处是控制核宽度的参数 h 对于所有的核都是固定的。在高数据密度的区域, 大的 h 值可能会造成过度平滑, 并且破坏了本应从数据中提取出的结构。但是, 减小 h 的值可能导致数据空间中低密度区域估计的噪声。因此, h 的最优选择可能依赖于数据空间的位置。这个问题可以通过概率密度的近邻方法解决。

因此我们回到局部概率密度估计的一般结果 (2.246)。与之前固定 V 然后从数据中确定 K 的值不同, 我们考虑固定 K 的值然后使用数据来确定合适的 V 值。为了完成这一点, 我们考虑一个以 x 为中心的小球体, 然后我们想估计概率密度 $p(x)$ 。并且, 我们允许球体的半径可以自由增长, 直到它精确地包含 K 个数据点。这样, 概率密度 $p(x)$ 的估计就由公式 (2.246) 给出, 其中 V 等于最终球体的体积。这种方法被称为 K 近邻方法。图2.26给出了对于不同参数 K , 使用与图2.24和图2.25相同的数据集, K 近邻方法的结果。我们看到 K 的值现在控制了光滑的程度, 并且与之前一样, K 的最有选择既不能过大也不能过小。注意, 由 K 近邻方法得到的模型不是真实的概率密度模型, 因为它在整个空间的积分是发散的。

在本章的最后, 我们要说明概率密度估计的 K 近邻方法如何推广到分类问题。为了完成这一点, 我们把 K 近邻概率密度估计方法分别应用到每个独立的类别中, 然后使用贝叶斯定理。假设我们有一个数据集, 其中 N_k 个数据点属于类别 C_k , 数据点的总数为 N , 因此 $\sum_k N_k = N$ 。如果我们想对一个新的数据点 x 进行分类, 那么我们可以画一个以 x 为中心的球体, 这个球体精确地包含 K 个数据点 (无论属于哪个类别)。假设球体的体积为 V , 并且包含来自类别 C_k 的 K_k 个数据点。这样公式 (2.246) 提供了与每个类别关联的一个概率密度的估计

$$p(\mathbf{x} \mid C_k) = \frac{K_k}{N_k V} \quad (2.253)$$

类似地, 无条件概率密度为

$$p(\mathbf{x}) = \frac{K}{NV} \quad (2.254)$$

而类先验为

$$p(C_k) = \frac{N_k}{N} \quad (2.255)$$

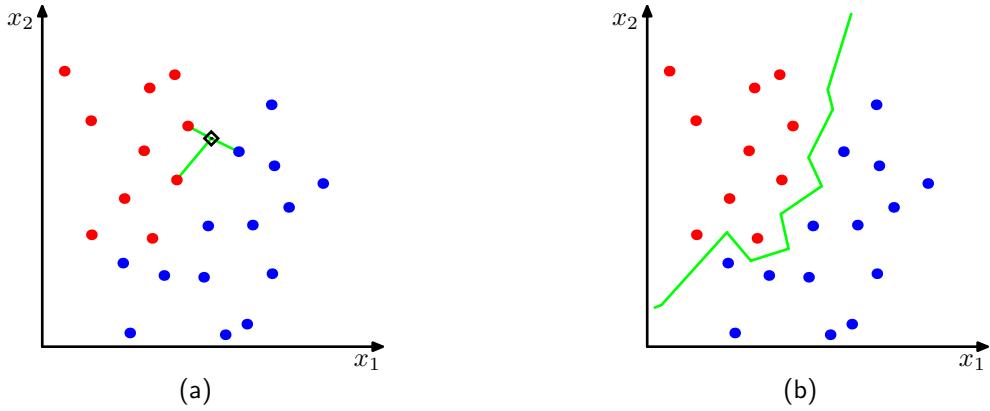


图 2.27: (a) 在 K 近邻分类器中, 一个新的数据点 (黑色菱形表示) 根据 K 个距离最近的训练数据点的主要类别确定, 其中 $K = 3$ 。(b) 在最近邻 ($K = 1$) 分类方法中, 生成的决策边界由不同类别的点对的垂直平分线组成的超平面确定。

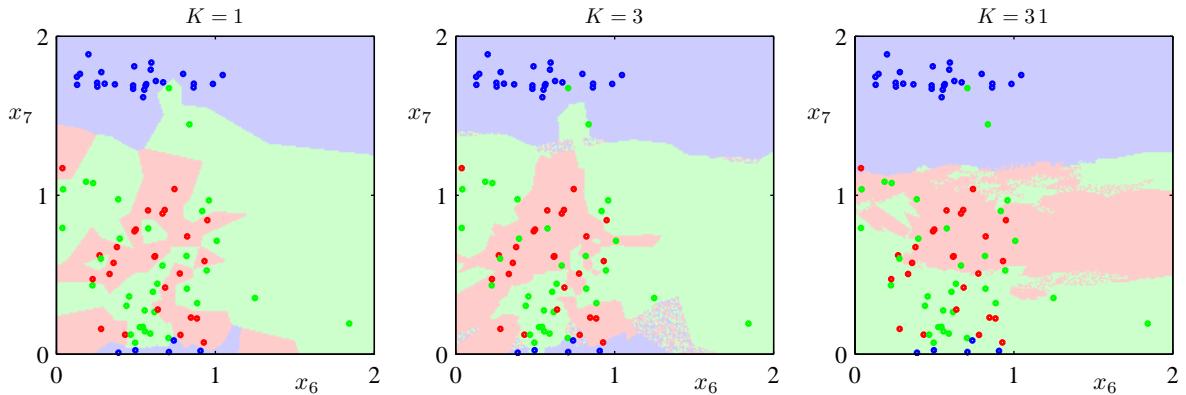


图 2.28: 石油数据集中的 200 个数据点的 x_6 与 x_7 的图像, 其中红色、绿色、蓝色的点分别对应于“薄片状”、“环状”、“同质状”的类别。同时给出的是对于不同 K 值, 由 K 近邻算法给出的输入空间的类别。

我们现在使用贝叶斯定理将公式 (2.253)、公式 (2.254) 和公式 (2.255) 结合起来, 可以得到类别的后验概率

$$p(C_k | \mathbf{x}) = \frac{p(\mathbf{x} | C_k)p(C_k)}{p(\mathbf{x})} = \frac{K_k}{K} \quad (2.256)$$

如果我们想最小化错误分类的概率, 那么我们可以把测试点 \mathbf{x} 分配给有着最大后验概率的类别, 这对应于最大的 $\frac{K_k}{K}$ 。因此为了分类一个新的数据点, 我们从训练数据中选择 K 个最近的数据点, 然后把新的数据点分配为这个集合中数量最多的点的类别。 $K = 1$ 的特例被称为最近邻规则 (nearest-neighbour rule), 因为测试点简单地被分类为训练数据集里距离最近的数据点的类别。图 2.27 给出了这些概念的说明。

在图 2.28 中, 我们给出了第一章介绍的石油流数据集在不同的 K 值下的 K 近邻算法的结果。正如我们期望的那样, 我们看到 K 控制了光滑的程度, 即小的 K 值会使得每个类别有许多小区域, 而大的 K 值会产生数量较少面积较大的区域。

最近邻 ($K = 1$) 分类器的一个有趣的性质是在极限 $N \rightarrow \infty$ 的情况下, 错误率不会超过最优分类器 (即使用真实概率分布的分类器) 可以达到的最小错误率的二倍 (Cover and Hart, 1967)。

正如到目前为止讨论的那样, K 近邻方法和核密度估计方法都需要存储整个训练数据。如果数据集很大的话, 这会造成很大的计算代价。通过建立一个基于树的搜索结构, 使得 (近似) 近邻可以高效地被找到, 而不必遍历整个数据集, 这种计算代价可以被抵消, 代价就是需要进行一次性的额外计算量。尽管这样, 这些非参数化方法仍然有很大的局限性。另一方面, 我们

已经看到，简单的参数化模型非常受限，因为它们只能表示某一种形式的概率分布。因此我们需要寻找一种概率密度模型，这种模型需要非常灵活，并且它的复杂度可以被控制为与训练数据的规模无关。我们在后续章节中将会看到如何找到这种概率密度模型。

2.6 练习

(2.1) (*) 证明伯努利分布 (2.2) 满足下面的性质。

$$\sum_{x=0}^1 p(x | \mu) = 1 \quad (2.257)$$

$$\mathbb{E}[x] = \mu \quad (2.258)$$

$$\text{var}[x] = \mu(1 - \mu) \quad (2.259)$$

证明，一个服从伯努利分布的随机二值变量 x 的熵 $H[x]$ 为

$$H[x] = -\mu \ln \mu - (1 - \mu) \ln(1 - \mu) \quad (2.260)$$

(2.2) (**) 公式 (2.2) 给出的伯努利分布的形式关于 x 的两个值不是对称的。在某些情况下，更方便的做法是使用一种等价的表示形式 $x \in \{-1, 1\}$ 。这种情况下，分布可以写成

$$p(x \in \mu) = \left(\frac{1-\mu}{2}\right)^{\frac{1-x}{2}} \left(\frac{1+\mu}{2}\right)^{\frac{1+x}{2}} \quad (2.261)$$

其中 $\mu \in [-1, 1]$ 。证明概率分布 (2.261) 是归一化的，并且计算它的均值、方差、熵。

(2.3) (**) 本练习中，我们证明二项分布 (2.9) 是归一化的。首先，使用从 N 个相同的物体中选择 m 个物体的组合数的定义 (2.10)，证明

$$\binom{N}{m} + \binom{N}{m-1} = \binom{N+1}{m} \quad (2.262)$$

使用这个结果，利用数学归纳法，证明

$$(1+x)^N = \sum_{m=0}^N \binom{N}{m} x^m \quad (2.263)$$

这被称为二项定理 (binomial theorem)，并且对于所有的实数 x 都成立。最后，证明二项分布是归一化的，即

$$\sum_{m=0}^N \binom{N}{m} \mu^m (1-\mu)^{N-m} = 1 \quad (2.264)$$

可以这样证明：首先从求和式中提出一个因子 $(1-\mu)^N$ ，然后使用二项定理即可。

(2.4) (**) 证明二项分布的均值由公式 (2.11) 给出。为了证明这一点，可以对公式 (2.264) 两侧关于 μ 求微分，然后整理即可得到 m 的均值。类似地，通过对公式 (2.264) 两侧关于 μ 求两次微分，使用公式 (2.11) 给出二项分布的均值，证明二项分布的方差由公式 (2.12) 给出。

(2.5) (**) 在本练习中，我们证明由公式 (2.13) 给出的 Beta 分布是归一化的，即公式 (2.14) 成立。这等价于证明

$$\int_0^1 \mu^{a-1} (1-\mu)^{b-1} d\mu = \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)} \quad (2.265)$$

根据 Gamma 函数的定义 (1.141)，我们有

$$\Gamma(a)\Gamma(b) = \int_0^\infty \exp(-x) x^{a-1} dx \int_0^\infty \exp(-y) y^{b-1} dy \quad (2.266)$$

使用这个表达式，按照下面的方法证明公式 (2.265)。首先把对于 y 的积分放到对于 x 的积分的被积函数中，然后进行变量替换 $t = y + x$ ，其中 x 固定。之后交换 x 和 t 的积分顺序，最后进行变量替换 $x = t\mu$ ，其中 μ 是固定的。

(2.6) (*) 使用公式 (2.265) 的结果证明公式 (2.13) 给出的Beta分布的均值、方差、众数分别为

$$\mathbb{E}[\mu] = \frac{a}{a+b} \quad (2.267)$$

$$\text{var}[\mu] = \frac{ab}{(a+b)^2(a+b+1)} \quad (2.268)$$

$$\text{mode}[\mu] = \frac{a-1}{a+b-2} \quad (2.269)$$

(2.7) (**) 考虑一个服从公式 (2.9) 给出的二项分布的随机变量 x ， μ 的先验分布为公式 (2.13) 给出的Beta分布。假设我们观察到了 m 次 $x = 1$ ，以及 l 次 $x = 0$ 。证明 μ 的后验均值位于先验均值和 μ 的最大似然估计值之间。为了完成这一点，证明后验均值可以写成 λ 乘以先验均值加上 $(1 - \lambda)$ 乘以最大似然估计，其中 $0 \leq \lambda \leq 1$ 。这表明后验概率分布的概念是先验概率分布和最大似然解的一种折中。

(2.8) (*) 考虑两个变量 x 和 y ，联合概率分布为 $p(x, y)$ 。证明下面两个结果。

$$\mathbb{E}[x] = \mathbb{E}_y[\mathbb{E}_x[x | y]] \quad (2.270)$$

$$\text{var}[x] = \mathbb{E}_y[\text{var}_x[x | y]] + \text{var}_y[\mathbb{E}_x[x | y]] \quad (2.271)$$

这里， $\mathbb{E}_x[x | y]$ 表示在条件分布 $p(x | y)$ 下， x 的期望。条件方差的记号与此类似。

(2.9) (***) 在本练习中，我们使用数学归纳法证明，公式 (2.38) 给出的狄利克雷分布是归一化的。我们已经在练习2.5中证明了狄利克雷分布的 $M = 2$ 的特殊情形 (Beta分布) 是归一化的。我们现在假设狄利克雷分布对于 $M - 1$ 个变量是归一化的，证明它对于 M 个变量也是归一化的。为了证明这一点，考虑 M 个变量上的狄利克雷分布，利用限制条件 $\sum_{k=1}^M \mu_k = 1$ 消除 μ_M ，从而狄利克雷分布可以写成

$$p_M(\mu_1, \dots, \mu_{M-1}) = C_M \prod_{k=1}^{M-1} \mu_k^{\alpha_{k-1}} \left(1 - \sum_{j=1}^{M-1} \mu_j\right)^{\alpha_{M-1}} \quad (2.272)$$

我们的目标是找到 C_M 的表达式。为了完成这一点，对 μ_{M-1} 积分，注意积分限。然后进行变量替换，使得积分限为0和1。假设 C_{M-1} 的结果正确，使用公式 (2.265)，推导出 C_M 的表达式。

(2.10) (**) 使用Gamma函数的性质 $\Gamma(x+1) = x\Gamma(x)$ ，证明由公式 (2.38) 给出的狄利克雷分布的均值、方差、协方差为下面的结果。

$$\mathbb{E}[\mu_j] = \frac{\alpha_j}{\alpha_0} \quad (2.273)$$

$$\text{var}[\mu_j] = \frac{\alpha_j(\alpha_0 - \alpha_j)}{\alpha_0^2(\alpha_0 + 1)} \quad (2.274)$$

$$\text{cov}[\mu_j \mu_l] = -\frac{\alpha_j \alpha_l}{\alpha_0^2(\alpha_0 + 1)}, \quad j \neq l \quad (2.275)$$

其中 α_0 由公式 (2.39) 定义。

(2.11) (*) 在狄利克雷分布 (2.38) 下，通过将 $\ln \mu_j$ 的期望表示为 α_j 的形式，证明

$$\mathbb{E}[\ln \mu_j] = \psi(\alpha_j) - \psi(\alpha_0) \quad (2.276)$$

其中 α_0 由公式 (2.39) 给出，且

$$\psi(a) \equiv \frac{d}{da} \ln \Gamma(a) \quad (2.277)$$

是一个Digamma函数。

(2.12) (*) 连续变量 x 的均匀分布被定义为

$$U(x | a, b) = \frac{1}{b-a}, \quad a \leq x \leq b \quad (2.278)$$

证明分布是归一化的，并求出分布的均值和方差。

(2.13) (**) 计算两个高斯分布 $p(\mathbf{x}) = \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}, \boldsymbol{\Sigma})$ 和 $q(\mathbf{x}) = \mathcal{N}(\mathbf{x} | \mathbf{m}, \mathbf{L})$ 的Kullback-Leibler散度 (1.113)。

(2.14) (**) 这个练习说明了，对于给定的协方差，具有最大熵的多元概率分布是高斯分布。概率分布 $p(\mathbf{x})$ 的熵为

$$H[\mathbf{x}] = - \int p(\mathbf{x}) \ln p(\mathbf{x}) d\mathbf{x} \quad (2.279)$$

我们想要对于所有的归一化的且具有均值和协方差的概率分布 $p(\mathbf{x})$ ，最大化 $H[\mathbf{x}]$ ，即

$$\int p(\mathbf{x}) d\mathbf{x} = 1 \quad (2.280)$$

$$\int p(\mathbf{x}) \mathbf{x} d\mathbf{x} = \boldsymbol{\mu} \quad (2.281)$$

$$\int p(\mathbf{x}) (\mathbf{x} - \boldsymbol{\mu})(\mathbf{x} - \boldsymbol{\mu})^T d\mathbf{x} = \boldsymbol{\Sigma} \quad (2.282)$$

通过使用变分法对 (2.279) 进行最大化，然后使用拉格朗日乘数法来引入限制条件 (2.280)、(2.281) 和 (2.282)，证明最大似然分布由高斯分布 (2.43) 给出。

(2.15) (**) 证明多元高斯分布 $\mathcal{N}(\mathbf{x} | \boldsymbol{\mu}, \boldsymbol{\Sigma})$ 的熵为

$$H[\mathbf{x}] = \frac{1}{2} \ln |\boldsymbol{\Sigma}| + \frac{D}{2} (1 + \ln(2\pi)) \quad (2.283)$$

其中 D 是 \mathbf{x} 的维度。

(2.16) (****) 考虑两个服从高斯分布的随机变量 x_1 和 x_2 ，均值分别为 μ_1 和 μ_2 ，精度分别为 τ_1 和 τ_2 。推导变量 $x = x_1 + x_2$ 的表达式。推导方法如下。首先使用下面的关系找到 x 的概率分布。

$$p(x) = \int_{-\infty}^{\infty} p(x | x_2) p(x_2) dx_2 \quad (2.284)$$

然后对指数项配平方。之后，我们观察到，它表示两个高斯分布的卷积，这个卷积本身是高斯分布。最后，使用公式 (1.110) 给出的一元高斯分布的熵的结果。

(2.17) (*) 考虑公式 (2.43) 给出的多元高斯分布。通过把精度矩阵（协方差矩阵的逆矩阵） $\boldsymbol{\Sigma}^{-1}$ 写成对称矩阵和反对称矩阵的和，证明反对称项不会出现在高斯分布的指数项中，因此我们可以令精度矩阵为对称矩阵而不失一般性。由于对称矩阵的逆矩阵还是对称矩阵（见练习2.22），因此我们也可以令协方差矩阵为对称矩阵而不失一般性。

(2.18) (****) 考虑一个实对称矩阵 $\boldsymbol{\Sigma}$ ，它的特征值方程由公式 (2.45) 给出。通过对这个方程取复共轭，然后与原方程相减，之后与特征向量 \mathbf{u}_i 做内积，证明特征值 λ_i 是实数。类似地，使用 $\boldsymbol{\Sigma}$ 的对称性，证明如果 $\lambda_j \neq \lambda_i$ ，那么两个特征值 \mathbf{u}_i 和 \mathbf{u}_j 正交。最后，证明不失一般性，特征向量的集合可以选择成单位正交的，即它们满足公式 (2.46)，即使某些特征值为零。

(2.19) (**) 证明，具有特征值方程 (2.45) 的实对称矩阵 $\boldsymbol{\Sigma}$ 可以表示成特征向量的展开式，系数由特征值给出，形式如公式 (2.48) 所示。类似地，证明，逆矩阵 $\boldsymbol{\Sigma}^{-1}$ 可以表示为公式 (2.49)。

(2.20) (**) 一个正定矩阵 $\boldsymbol{\Sigma}$ 的定义为：对于任意实值向量 \mathbf{a} ，下面的二次型都为正。

$$\mathbf{a}^T \boldsymbol{\Sigma} \mathbf{a} \quad (2.285)$$

证明， $\boldsymbol{\Sigma}$ 为正定矩阵的充分必要条件是 $\boldsymbol{\Sigma}$ 的所有特征值 λ_i 均为正。

(2.21) (*) 证明 $D \times D$ 的实对称矩阵有 $\frac{D(D+1)}{2}$ 个独立参数。

(2.22) (*) 证明对称矩阵的逆矩阵还是对称矩阵。

(2.23) (**) 通过使用特征向量展开式 (2.48) 对坐标系进行对角化, 证明, 对应于常数马氏距离 Δ 的超椭球体的体积为

$$V_D |\Sigma|^{\frac{1}{2}} \Delta^D \quad (2.286)$$

其中 V_D 是 D 维单位球体的体积, 马氏距离由公式 (2.44) 定义。

(2.24) (**) 证明恒等式 (2.76)。方法为: 将两边都乘以矩阵

$$\begin{pmatrix} A & B \\ C & D \end{pmatrix} \quad (2.287)$$

然后使用公式 (2.77)。

(2.25) (**) 在 2.3.1 节和 2.3.2 节, 我们考虑了多元高斯分布的条件分布和边缘分布。更一般地, 我们可以考虑将 x 的元素划分为三组 x_a, x_b 和 x_c , 对应的均值向量 μ 的划分和协方差矩阵 Σ 的划分如下

$$\mu = \begin{pmatrix} \mu_a \\ \mu_b \\ \mu_c \end{pmatrix}, \quad \Sigma = \begin{pmatrix} \Sigma_{aa} & \Sigma_{ab} & \Sigma_{ac} \\ \Sigma_{ba} & \Sigma_{bb} & \Sigma_{bc} \\ \Sigma_{ca} & \Sigma_{cb} & \Sigma_{cc} \end{pmatrix} \quad (2.288)$$

使用 2.3 节的结果, 找到条件概率 $p(x_a | x_b)$ 的表达式, 其中 x_c 已经被边缘化 (积分或求和)。

(2.26) (**) 线性代数的一个有用的结果是 Woodbury 矩阵求逆公式

$$(A + BCD)^{-1} = A^{-1} - A^{-1}B(C^{-1} + DA^{-1}B)^{-1}DA^{-1} \quad (2.289)$$

通过将两侧同时乘以 $(A + BCD)$, 证明这个结果的正确性。

(2.27) (*) 令 x 和 z 是两个独立的随机向量, 即 $p(x, z) = p(x)p(z)$ 。证明它们的和 $y = x + z$ 的均值等于各自分别的均值之和。类似地, 证明 y 的协方差矩阵等于 x 的协方差矩阵和 z 的协方差矩阵之和。证明这个结果与练习 1.10 的结果相符。

(2.28) (****) 考虑变量

$$z = \begin{pmatrix} x \\ y \end{pmatrix} \quad (2.290)$$

的联合概率分布, 它的均值和协方差分别由公式 (2.108) 和公式 (2.105) 给出。通过使用公式 (2.92) 和公式 (2.93) 的结果, 证明边缘概率分布 $p(x)$ 由公式 (2.99) 给出。类似地, 通过使用公式 (2.81) 和公式 (2.82) 的结果, 证明条件概率分布 $p(y | x)$ 由公式 (2.100) 给出。

(2.29) (**) 使用分块矩阵的求逆公式 (2.76), 证明精度矩阵 (2.104) 的逆矩阵由公式 (2.105) 的协方差矩阵给出。

(2.30) (*) 从公式 (2.107) 开始, 使用 (2.105) 的结果, 证明公式 (2.108)。

(2.31) (**) 考虑两个多维随机向量 x 和 z , 它们分别服从高斯分布 $p(x) = \mathcal{N}(x | \mu_x, \Sigma_x)$ 和 $p(z) = \mathcal{N}(z | \mu_z, \Sigma_z)$, 它们的和为 $y = x + z$ 。使用公式 (2.109) 和公式 (2.110) 的结果, 通过考虑由边缘概率分布 $p(x)$ 和条件概率分布 $p(y | x)$ 的乘积构成的线性高斯模型, 求出边缘概率分布 $p(y)$ 的表达式。

(2.32) (****) 本练习和下一个练习提供了计算线性高斯模型中的二次型的机会, 同时给出了一个独立的对于教材正文中推导结果的检查。考虑一个联合概率分布 $p(x, y)$, 它通过公式 (2.99) 和公式 (2.100) 给出的边缘概率分布和条件概率分布定义。通过考察联合分布指数项的二次型, 使用 2.3 节讨论的配平方的方法, 找到边缘概率分布 $p(y)$ 的均值和协方差的表达式, 其中变量 x 已经被积分出去了。为了做到这一点, 使用 Woodbury 矩阵求逆公式 (2.289)。证明这个结果和第 2 章中推导出的结果 (2.109) 和 (2.110) 是相符的。

(2.33) (****) 考虑与练习 2.32 相同的联合概率分布, 但是现在使用配平方技术寻找条件概率分布 $p(x | y)$ 的均值和协方差的表达式。与之前一样, 证明这个结果与对应的表达式 (2.111) 和 (2.112) 相符。

(2.34) (**) 为了找到多元高斯分布的协方差矩阵的最大似然解, 我们需要关于 Σ 最大化对数似然函数 (2.118)。注意, 协方差矩阵一定是对称的、正定的。这里, 我们忽略这些限

制，直接进行最大化。使用附录C中的结论 (C.21)、(C.26) 和 (C.28)，证明最大化对数似然函数 (2.118) 的协方差矩阵 Σ 由样本协方差 (2.122) 给出。我们注意到最终求得的结果确实是对称的、正定的（假设样本协方差矩阵非奇异）。

(2.35) (***) 使用公式 (2.59) 的结果证明公式 (2.62)。现在使用 (2.59) 和 (2.62)，证明

$$\mathbb{E}[\mathbf{x}_n \mathbf{x}_m^T] = \boldsymbol{\mu} \boldsymbol{\mu}^T + I_{nm} \boldsymbol{\Sigma} \quad (2.291)$$

其中 \mathbf{x}_n 表示从均值为 $\boldsymbol{\mu}$ 协方差为 $\boldsymbol{\Sigma}$ 的高斯分布中采样的数据点， I_{nm} 表示单位矩阵的第 (n, m) 个元素。从而证明了公式 (2.124) 给出的结论。

(2.36) (***) 使用与推导公式 (2.126) 类似的步骤，推导一元高斯分布的方差的顺序估计的表达式。推导的起点为最大似然表达式

$$\sigma_{ML}^2 = \frac{1}{N} \sum_{n=1}^N (x_n - \mu)^2 \quad (2.292)$$

证明，把这个高斯分布的表达式代入 Robbins-Monro 顺序估计公式 (2.135) 中，会得到同样形式的结果，因此就可以得到对应的系数 a_N 的表达式。

(2.37) (***) 使用与推导公式 (2.126) 类似的步骤，推导多元高斯分布的协方差的顺序轨迹的表达式。推导的起点为最大似然表达式 (2.122)。证明，把这个高斯分布的表达式代入 Robbins-Monro 顺序估计公式 (2.135) 中，会得到同样形式的结果，因此就可以得到对应的系数 a_N 的表达式。

(2.38) (*) 对指数项上的二次型进行配平方，推导出公式 (2.141) 和 (2.142) 给出的结果。

(2.39) (***) 从高斯随机变量的后验概率分布的结果 (2.141) 和 (2.142) 开始，分离出前 $N - 1$ 个数据点的贡献，因此就得到了 μ_N 和 σ_N^2 的顺序更新的表达式。现在从后验概率分布 $p(\boldsymbol{\mu} | x_1, \dots, x_{N-1}) = \mathcal{N}(\boldsymbol{\mu} | \boldsymbol{\mu}_{N-1}, \sigma_{N-1}^2)$ 开始，推导出同样的结果，然后与似然函数 $p(x_N | \boldsymbol{\mu}) = \mathcal{N}(x_N | \boldsymbol{\mu}, \sigma^2)$ 相乘，之后配平方、归一化，就得到了 N 次观察之后的后验概率分布。

(2.40) (**) 考虑 D 维高斯随机变量 \mathbf{x} ，分布为 $\mathcal{N}(\mathbf{x} | \boldsymbol{\mu}, \boldsymbol{\Sigma})$ ，其中协方差矩阵 $\boldsymbol{\Sigma}$ 已知，我们想从一组观测 $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ 中推断出均值 $\boldsymbol{\mu}$ 。给定一个先验概率分布 $p(\boldsymbol{\mu}) = \mathcal{N}(\boldsymbol{\mu} | \boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$ ，找到对应的后验概率分布 $p(\boldsymbol{\mu} | \mathbf{X})$ 。

(2.41) (*) 使用 Gamma 函数的定义 (1.141)，证明 Gamma 分布 (2.146) 是归一化的。

(2.42) (**) 计算 Gamma 分布 (2.146) 的均值、方差、众数。

(2.43) (*) 下面的分布

$$p(x | \sigma^2, q) = \frac{q}{2(2\sigma^2)^{\frac{1}{q}} \Gamma(\frac{1}{q})} \exp\left(-\frac{|x|^q}{2\sigma^2}\right) \quad (2.293)$$

是一元高斯分布的推广。证明这个分布是归一化的，即

$$\int_{-\infty}^{\infty} p(x | \sigma^2, q) dx = 1 \quad (2.294)$$

并且当 $q = 2$ 时，它会变为高斯分布。考虑一个回归模型，它的目标变量为 $t = y(\mathbf{x}, \mathbf{w}) + \epsilon$ ，其中 ϵ 是一个随机噪声，服从公式 (2.293) 给出的概率分布。对于输入向量的观测数据集 $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ 和对应的目标变量 $\mathbf{t} = (t_1, \dots, t_N)^T$ ，证明关于 \mathbf{w} 和 σ^2 的对数似然函数为

$$\ln p(\mathbf{t} | \mathbf{X}, \mathbf{w}, \sigma^2) = -\frac{1}{2\sigma^2} \sum_{n=1}^N |y(\mathbf{x}_n, \mathbf{w}) - t_n|^q - \frac{N}{q} \ln(2\sigma^2) + \text{常数} \quad (2.295)$$

其中，“常数”表示与 \mathbf{w} 和 σ^2 无关的项。注意，作为 \mathbf{w} 的函数，这就是 1.5.5 节讨论的 L_q 误差函数。

(2.44) (**) 考虑一个一元高斯分布 $\mathcal{N}(x | \mu, \tau^{-1})$ ，其共轭先验为公式 (2.154) 给出的高斯-Gamma 分布。我们还有一个数据集 $\mathbf{x} = \{x_1, \dots, x_N\}$ ，每个观测都是独立同分布的。证明后

验概率分布也是一个高斯-Gamma分布，与先验分布有着相同的函数形式。写出这个后验概率分布的参数的表达式。

(2.45) (*) 证明由公式 (2.155) 定义的Wishart分布确实是多元高斯分布的精度矩阵的共轭先验。

(2.46) (*) 证明，计算公式 (2.158) 的积分会得到公式 (2.159) 的结果。

(2.47) (*) 证明，在极限 $\nu \rightarrow \infty$ 的情况下，公式 (2.159) 给出的t分布会变成高斯分布。

提示：忽略归一化系数，只关注与 x 有关的项。

(2.48) (*) 使用与推导一元学生t分布 (2.159) 类似的过程，通过对公式 (2.161) 的变量 η 进行积分，证明多元学生t分布的形式为公式 (2.162)。使用定义 (2.161)，通过交换积分变量的方法，证明多元t分布被正确归一化了。

(2.49) (**) 通过将多元学生t分布的定义 (2.161) 看做Gamma分布和高斯分布的卷积，证明由公式 (2.162) 定义的多元t分布的性质 (2.164)、(2.165) 和 (2.166)。

(2.50) (*) 证明，在极限 $\nu \rightarrow \infty$ 的情况下，公式 (2.162) 给出的多元学生t分布会变为均值为 μ 精度为 Λ 的高斯分布。

(2.51) (*) 本章在讨论周期变量时使用的各种三角恒等式可以很容易地通过下面的关系证明。

$$\exp(iA) = \cos A + i \sin A \quad (2.296)$$

其中 i 是-1的平方根。通过考虑下面的恒等式

$$\exp(iA) \exp(-iA) = 1 \quad (2.297)$$

证明结果 (2.177)。类似地，使用恒等式

$$\cos(A - B) = \Re \exp\{i(A - B)\} \quad (2.298)$$

其中 \Re 表示实部，证明公式 (2.178)。最后，使用 $\sin(A - B) = \Im \exp\{i(A - B)\}$ ，其中 \Im 表示虚部，证明结果 (2.183)。

(2.52) (**) 对于大的 m ，von Mises分布 (2.179) 在众数 θ_0 附近会出现尖峰。通过定义 $\xi = m^{\frac{1}{2}}(\theta - \theta_0)$ ，使用余弦函数的泰勒展开式

$$\cos \alpha = 1 - \frac{\alpha^2}{2} + O(\alpha^4) \quad (2.299)$$

证明，随着 $m \rightarrow \infty$ ，von Mises分布趋近于高斯分布。

(2.53) (*) 使用三角恒等式 (2.183)，证明 (2.182) 关于 θ_0 的解为 (2.184)。

(2.54) (*) 通过计算von Mises分布 (2.179) 的一阶导数和二阶导数，并且使用 $m > 0$ 时 $I_0(m) > 0$ 的性质，证明当 $\theta = \theta_0$ 时，概率分布取得最大值；当 $\theta = \theta_0 + \pi \pmod{2\pi}$ 时，概率分布取得最小值。

(2.55) (*) 通过使用公式 (2.168) 给出的结果，以及公式 (2.184) 和三角恒等式

(2.178)，证明von Mises分布的concentration参数的最大似然解 m_{ML} 满足 $A(m_{ML}) = \bar{r}$ ，其中 \bar{r} 是当我们把观测看成二维欧几里得空间的单位向量时（如图2.17所示），观测的均值的半径。

(2.56) (**) 把Beta分布 (2.13)、Gamma分布 (2.146) 和von Mises分布 (2.179) 表达为指数族分布 (2.194) 的成员，从而就可以求出它们的自然参数。

(2.57) (*) 证明多元高斯分布可以转化为形如 (2.194) 的指数族分布，推导出类似于 (2.220) 到 (2.223) 的 η 、 $\mathbf{u}(x)$ 、 $h(x)$ 和 $g(\eta)$ 的表达式。

(2.58) (*) 公式 (2.226) 给出的结果表明，对于指数族分布， $\ln g(\eta)$ 的负梯度为 $\mathbf{u}(x)$ 的期望。通过对公式 (2.195) 的两侧取二阶导数，证明

$$-\nabla \nabla \ln g(\eta) = \mathbb{E}[\mathbf{u}(x)\mathbf{u}(x)^T] - \mathbb{E}[\mathbf{u}(x)]\mathbb{E}[\mathbf{u}(x)]^T = \text{cov}[\mathbf{u}(x)] \quad (2.300)$$

(2.59) (*) 通过使用 $y = \frac{x}{\sigma}$ 进行变量替换，证明，如果 $f(x)$ 被正确归一化了，那么概率密度 (2.236) 就会被正确归一化。

(2.60) (***) 考虑一个类似直方图的密度模型，其中空间 x 被分成固定的区域，且在第*i*个区域中，概率密度 $p(x)$ 取常数值 h_i ，且区域*i*的体积被记作 Δ_i 。假设我们有*N*次 x 的观测，这些观测中的 n_i 次落在区域*i*中。使用一个拉格朗日乘数给概率密度施加归一化的限制，推导出 $\{h_i\}$ 的最大似然估计的表达式。

(2.61) (*) 证明*K*近邻概率密度模型定义了一个反常的概率分布，这个分布在整个空间上的积分是发散的。

3 回归的线性模型

目前为止，本书的关注点是无监督学习，包括诸如概率密度估计和数据聚类等话题。我们现在开始讨论有监督学习，首先讨论的是回归问题。回归问题的目标是在给定 D 维输入（input）变量 \mathbf{x} 的情况下，预测一个或者多个连续目标（target）变量 t 的值。在第1章中，我们已经遇到了回归问题的一个例子：多项式曲线拟合问题。多项式是被称为线性回归模型的一大类函数的一个具体的例子。线性回归模型有着可调节的参数，具有线性函数的性质，将会成为本章的关注点。线性回归模型的最简单的形式也是输入变量的线性函数。但是，通过将一组输入变量的非线性函数进行线性组合，我们可以获得一类更加有用的函数，被称为基函数（basis function）。这样的模型是参数的线性函数，这使得其具有一些简单的分析性质，同时关于输入变量是非线性的。

给定一个由 N 个观测值 $\{x_n\}$ 组成的数据集，其中 $n = 1, \dots, N$ ，以及对应的目标值 $\{t_n\}$ ，我们的目标是预测对于给定新的 x 值的情况下， t 的值。最简单的方法是，直接建立一个适当的函数 $y(\mathbf{x})$ ，对于新的输入 \mathbf{x} ，这个函数能够直接给出对应的 t 的预测。更一般地，从一个概率的观点来看，我们的目标是对预测分布 $p(t | \mathbf{x})$ 建模，因为它表达了对于每个 \mathbf{x} 值，我们对于 t 的值的不确定性。从这个条件概率分布中，对于任意的 \mathbf{x} 的新值，我们可以对 t 进行预测，这种方法等同于最小化一个恰当选择的损失函数的期望值。正如在1.5.5节讨论的那样，对于实值变量来说，损失函数的一个通常的选择是平方误差损失，这种情况下最优解由 t 的条件期望给出。

虽然线性模型对于模式识别的实际应用来说有很大的局限性，特别是对于涉及到高维输入空间的问题来说更是如此，但是他们有很好的分析性质，并且组成了后续章节中将要讨论的更加复杂的模型的基础。

3.1 线性基函数模型

回归问题的最简单模型是输入变量的线性组合

$$y(\mathbf{x}, \mathbf{w}) = w_0 + w_1 x_1 + \dots + w_D x_D \quad (3.1)$$

其中 $\mathbf{x} = (x_1, \dots, x_D)^T$ 。这通常被简单地称为线性回归（linear regression）。这个模型的关键性质是它是参数 w_0, \dots, w_D 的一个线性函数。但是，它也是输入变量 x_i 的一个线性函数，这给模型带来的极大的局限性。因此我们这样扩展模型的类别：将输入变量的固定的非线性函数进行线性组合，形式为

$$y(\mathbf{x}, \mathbf{w}) = w_0 + \sum_{j=1}^{M-1} w_j \phi_j(\mathbf{x}) \quad (3.2)$$

其中 $\phi_j(\mathbf{x})$ 被称为基函数（basis function）。通过把下标 j 的最大值记作 $M - 1$ ，这个模型中的参数总数为 M 。

参数 w_0 使得数据中可以存在任意固定的偏置，这个值通常被称为偏置参数（bias parameter）。注意不要把这里的“偏置”与统计学中的“偏置”弄混淆。通常，定义一个额外的虚“基函数” $\phi_0(\mathbf{x}) = 1$ 是很方便的，这时

$$y(\mathbf{x}, \mathbf{w}) = \sum_{j=0}^{M-1} w_j \phi_j(\mathbf{x}) = \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}) \quad (3.3)$$

其中 $\mathbf{w} = (w_0, \dots, w_{M-1})^T$ 且 $\boldsymbol{\phi} = (\phi_0, \dots, \phi_{M-1})^T$ 。在许多模式识别的实际应用中，我们会对原始的数据变量进行某种固定形式的预处理或者特征抽取。如果原始变量由向量 \mathbf{x} 组成，那么特征可以用基函数 $\{\phi_j(\mathbf{x})\}$ 来表示。

通过使用非线性基函数，我们能够让函数 $y(\mathbf{x}, \mathbf{w})$ 成为输入向量 \mathbf{x} 的一个非线性函数。但是，形如(3.2)的函数被称为线性模型，因为这个函数是 \mathbf{w} 的线性函数。正是这种关于参数的线性极大地简化了对于这列模型的分析。然而，这也造成了一些巨大的局限性，正如我们在3.6节讨论的那样。

第1章中讨论的多项式拟合的例子是这个模型的一个特例，那里有一个输入变量 x ，基函数是 x 的幂指数的形式，即 $\phi_j(x) = x^j$ 。多项式基函数的一个局限性是它们是输入变量的全局函

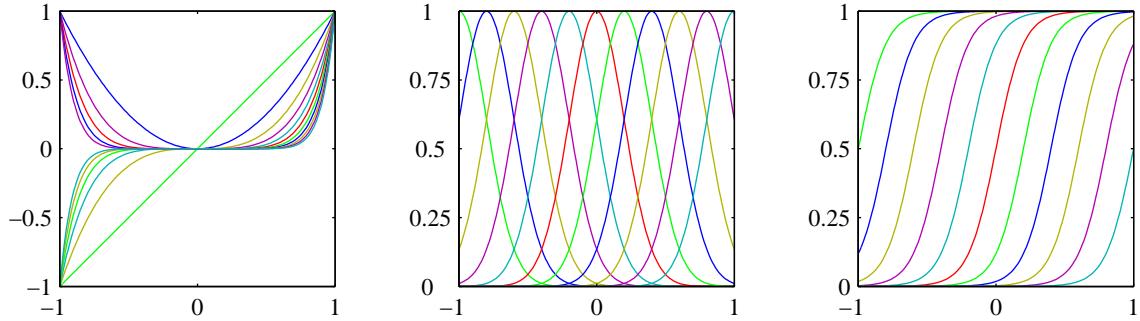


图 3.1: 基函数的例子, 左图是多项式基函数, 中图是形式为 (3.4) 的高斯基函数, 右图是形式为 (3.5) 的sigmoid基函数。

数, 因此对于输入空间一个区域的改变将会影响所有其他的区域。这个问题可以这样解决: 把输入空间切分成若干个区域, 然后对于每个区域用不同的多项式函数拟合。这样的函数叫做样条函数 (spline function) (Hastie et al., 2001)。

对于基函数, 有许多其他的选择, 例如

$$\phi_j(x) = \exp\left\{-\frac{(x - \mu_j)^2}{2s^2}\right\} \quad (3.4)$$

其中 μ_j 控制了基函数在输入空间中的位置, 参数 s 控制了基函数的空间大小。这种基函数通常被称为“高斯”基函数, 但是应该注意它们未必一定是一个概率表达式。特别地, 归一化系数不重要, 因为这些基函数会与一个调节参数 w_j 相乘。

另一种选择是sigmoid基函数, 形式为

$$\phi_j(x) = \sigma\left(\frac{x - \mu_j}{s}\right) \quad (3.5)$$

其中 $\sigma(a)$ 是 logistic sigmoid 函数, 定义为

$$\sigma_a = \frac{1}{1 + \exp(-a)} \quad (3.6)$$

等价地, 我们可以使用 \tanh 函数, 因为它和 logistic sigmoid 函数的关系为 $\tanh(a) = 2\sigma(2a) - 1$, 因此 logistic sigmoid 函数的一般的线性组合等价于 \tanh 函数的一般的线性组合。图 3.1 说明了基函数的不同选择情况。

基函数的另一种可能的选择是傅里叶基函数, 它可以用正弦函数展开。每个基函数表示一个具体的频率, 它在空间中有无限的延伸。相反, 限制在输入空间中的有限区域的基函数要由不同空间频率的一系列频谱组成。在许多信号处理的应用中, 一个吸引了研究者兴趣的问题是考虑同时在空间和频率受限的基函数。这种研究产生了一类被称为小波 (wavelet) 的函数。为了简化应用, 这些基函数被定义为相互正交的。当应用中的输入值位于正规的晶格中时, 应用小波最合适。这种应用包括时间序列中的连续的时间点, 以及图像中的像素。关于小波的有用的教科书包括 Ogden (1997), Mallat (1999) 和 Vidakovic (1999)。

但是, 本章中的大部分讨论都与基函数的选择无关。因此对于我们的大部分讨论, 我们不会具体化基函数的特定形式, 除非我们为了数值说明。事实上, 我们的大部分讨论将同等地适用于基函数向量 $\phi(x)$ 的形式为 $\phi(x) = x$ 的情形。此外, 为了保持记号的简洁, 我们把注意力集中于单一目标变量 t 的情形。但是在 3.1.5 节里, 我们将会简短地考虑必要的修改, 来处理多个目标变量的情形。

3.1.1 最大似然与最小平方

在第 1 章, 我们通过最小化平方和误差函数, 用多项式函数拟合数据集。我们也证明了, 这种误差函数可以看成高斯噪声模型的假设下的最大似然解。现在让我们回到这种讨论中, 更加详细地考虑最小平方的方法以及它与最大似然方法的关系。

与之前一样，我们假设目标变量 t 由确定的函数 $y(\mathbf{x}, \mathbf{w})$ 给出，这个函数被附加了高斯噪声，即

$$t = y(\mathbf{x}, \mathbf{w}) + \epsilon \quad (3.7)$$

其中 ϵ 是一个零均值的高斯随机变量，精度（方差的倒数）为 β 。因此我们有

$$p(t | \mathbf{x}, \mathbf{w}, \beta) = \mathcal{N}(t | y(\mathbf{x}, \mathbf{w}), \beta^{-1}) \quad (3.8)$$

回忆一下，如果我们假设一个平方损失函数，那么对于 \mathbf{x} 的一个新值，最优的预测由目标变量的条件均值给出。在公式 (3.8) 给出的高斯条件分布的情况下，条件均值可以简单地写成

$$\mathbb{E}[t | \mathbf{x}] = \int t p(t | \mathbf{x}) dt = y(\mathbf{x}, \mathbf{w}) \quad (3.9)$$

注意高斯噪声的假设表明，给定 \mathbf{x} 的条件下， t 的条件分布是单峰的，这对于一些实际应用来说是不合适的。第14.5.1节将扩展到条件高斯分布的混合，那种情况下可以描述多峰的条件分布。

现在考虑一个输入数据集 $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ ，对应的目标值为 t_1, \dots, t_N 。我们把目标向量 $\{t_n\}$ 组成一个列向量，记作 \mathbf{t} 。这个变量的字体与多元目标值的一次观测（记作 \mathbf{t} ）不同。假设这些数据点是独立地从分布 (3.8) 中抽取的，那么我们可以得到下面的似然函数的表达式，它是可调节参数 \mathbf{w} 和 β 的函数，形式为

$$p(\mathbf{t} | \mathbf{X}, \mathbf{w}, \beta) = \prod_{n=1}^N \mathcal{N}(t_n | \mathbf{w}^T \phi(\mathbf{x}_n), \beta^{-1}) \quad (3.10)$$

其中我们使用了公式 (3.3)。注意，在有监督学习问题中（例如回归问题和分类问题），我们不是在寻找模型来对输入变量的概率分布建模。因此 \mathbf{x} 总会出现条件变量的位置上。因此从现在开始，为了保持记号的简洁性，我们在诸如 $p(\mathbf{t} | \mathbf{x}, \mathbf{w}, \beta)$ 这类的表达式中不显式地写出 \mathbf{x} 。取对数似然函数的对数，使用一元高斯分布的标准形式 (2.146)，我们有

$$\begin{aligned} \ln p(\mathbf{t} | \mathbf{w}, \beta) &= \sum_{n=1}^N \ln \mathcal{N}(t_n | \mathbf{w}^T \phi(\mathbf{x}_n), \beta^{-1}) \\ &= \frac{N}{2} \ln \beta - \frac{N}{2} \ln(2\pi) - \beta E_D(\mathbf{w}) \end{aligned} \quad (3.11)$$

其中平方和误差函数的定义为

$$E_D(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^N \{t_n - \mathbf{w}^T \phi(\mathbf{x}_n)\}^2 \quad (3.12)$$

写出了似然函数，我们可以使用最大似然的方法确定 \mathbf{w} 和 β 。首先关于 \mathbf{w} 求最大值。正如我们已经在1.2.5节中已经看到的那样，我们看到在条件高斯噪声分布的情况下，线性模型的似然函数的最大化等价于平方和误差函数的最小化。平方和误差函数由 $E_D(\mathbf{w})$ 给出。公式 (3.11) 给出的对数似然函数的梯度为

$$\nabla \ln p(\mathbf{t} | \mathbf{w}, \beta) = \beta \sum_{n=1}^N \{t_n - \mathbf{w}^T \phi(\mathbf{x}_n)\} \phi(\mathbf{x}_n)^T \quad (3.13)$$

令这个梯度等于零，可得

$$0 = \sum_{n=1}^N t_n \phi(\mathbf{x}_n)^T - \mathbf{w}^T \left(\sum_{n=1}^N \phi(\mathbf{x}_n) \phi(\mathbf{x}_n)^T \right) \quad (3.14)$$

求解 \mathbf{w} ，我们有

$$\mathbf{w}_{ML} = (\Phi^T \Phi)^{-1} \Phi^T \mathbf{t} \quad (3.15)$$

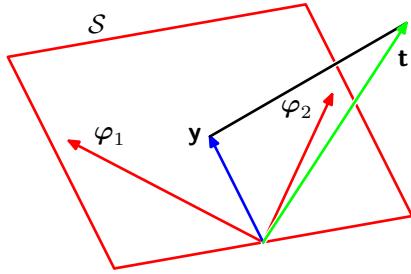


图 3.2: 最小平方解的几何表示, 在一个 N 维空间中, 坐标轴是 t_1, \dots, t_N 的值。最小平方回归函数可以通过下面的方式得到: 寻找数据向量 \mathbf{t} 在由基函数 $\phi_j(\mathbf{x})$ 张成的子空间上的正交投影, 其中每个基函数都可以看成一个长度为 N 的向量 φ_j , 它的元素为 $\phi_j(\mathbf{x}_n)$ 。

这被称为最小平方问题的规范方程 (normal equation)。这里 Φ 是一个 $N \times M$ 的矩阵, 被称为设计矩阵 (design matrix), 它的元素为 $\Phi_{nj} = \phi_j(\mathbf{x}_n)$, 即

$$\Phi = \begin{pmatrix} \phi_0(\mathbf{x}_1) & \phi_1(\mathbf{x}_1) & \cdots & \phi_{M-1}(\mathbf{x}_1) \\ \phi_0(\mathbf{x}_2) & \phi_1(\mathbf{x}_2) & \cdots & \phi_{M-1}(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_0(\mathbf{x}_N) & \phi_1(\mathbf{x}_N) & \cdots & \phi_{M-1}(\mathbf{x}_N) \end{pmatrix} \quad (3.16)$$

量

$$\Phi^\dagger \equiv (\Phi^T \Phi)^{-1} \Phi^T \quad (3.17)$$

被称为矩阵 Φ 的 Moore-Penrose 伪逆矩阵 (pseudo-inverse matrix) (Rao and Mitra, 1971; Golub and Van Loan, 1996)。它可以被看成逆矩阵的概念对于非方阵的矩阵的推广。实际上, 如果 Φ 是方阵且可逆, 那么使用性质 $(AB)^{-1} = B^{-1}A^{-1}$, 我们可以看到 $\Phi^\dagger \equiv \Phi^{-1}$ 。

现在, 我们可以更加深刻地认识偏置参数 w_0 。如果我们显式地写出偏置参数, 那么误差函数 (3.12) 变为

$$E_D(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^N \{t_n - w_0 - \sum_{j=1}^{M-1} w_j \phi_j(\mathbf{x}_n)\}^2 \quad (3.18)$$

令关于 w_0 的导数等于零, 解出 w_0 , 可得

$$w_0 = \bar{t} - \sum_{j=1}^{M-1} w_j \bar{\phi}_j \quad (3.19)$$

其中我们已经定义了

$$\bar{t} = \frac{1}{N} \sum_{n=1}^N t_n, \quad \bar{\phi}_j = \frac{1}{N} \sum_{n=1}^N \phi_j(\mathbf{x}_n) \quad (3.20)$$

因此偏置 w_0 补偿了目标值的平均值 (在训练集上的) 与基函数的值的平均值的加权求和之间的差。

我们也可以关于噪声精度参数 β 最大化似然函数 (3.11), 结果为

$$\frac{1}{\beta_{ML}} = \frac{1}{N} \sum_{n=1}^N \{t_n - \mathbf{w}_{ML}^T \phi(\mathbf{x}_n)\}^2 \quad (3.21)$$

因此我们看到噪声精度的倒数由目标值在回归函数周围的残留方差 (residual variance) 给出。

3.1.2 最小平方的几何描述

现在，考虑最小平方解的几何描述有助于理解这种方法。我们考虑一个 N 维空间，它的坐标轴由 t_n 给出，即 $\mathbf{t} = (t_1, \dots, t_N)^T$ 是这个空间中的一个向量。每个在 N 个数据点处估计的基函数 $\phi_j(\mathbf{x}_n)$ 也可以表示为这个空间中的一个向量，记作 φ_j ，如图3.2所示。注意， φ_j 对应于 Φ 的第 j 列，而 $\phi(\mathbf{x}_n)$ 对应于 Φ 的第 i 行。如果基函数的数量 M 小于数据点的数量 N ，那么 M 个向量 φ_j 将会张成一个 M 维的子空间 S 。我们定义 \mathbf{y} 是一个 N 维向量，它的第 n 个元素为 $y(\mathbf{x}_n, \mathbf{w})$ ，其中 $n = 1, \dots, N$ 。由于 \mathbf{y} 是向量 φ_j 的任意线性组合，因此它可以位于 M 维子空间的任何位置。这样，平方和误差函数 (3.12) 就等于 \mathbf{y} 和 \mathbf{t} 之间的平方欧氏距离（只相差一个因子 $\frac{1}{2}$ ）。因此， \mathbf{w} 的最小平方解对应于位于子空间 S 的与 \mathbf{t} 最近的 \mathbf{y} 的选择。直观来看，根据图3.2，我们猜想这个解对应于 \mathbf{t} 在子空间 S 上的正交投影。事实上确实是这样，并且很容易证明。注意到 \mathbf{y} 是由 $\Phi \mathbf{w}_{ML}$ 给出的，然后证明它的表达式为正交投影即可。

在实际应用中，当 $\Phi^T \Phi$ 接近奇异矩阵时，直接求解规范方程会导致数值计算上的困难。特别地，当两个或者更多的基向量 φ_j 共线或者接近共线时，最终的参数值会相当大。这样的退化在处理真实数据集的时候并不罕见。这种数值计算上的困难可以通过奇异值分解 (singular value decomposition) 或者简称SVD的方法解决 (Press et al., 1992; Bishop and Nabney, 2008)。注意，正则项的添加确保了矩阵是非奇异的，即使在退化的情况下也是如此。

3.1.3 顺序学习

最大似然解 (3.15) 的求解过程涉及到一次处理整个数据集。这种批处理技术对于大规模数据集来说计算量相当大。正如我们在第1章讨论的那样，如果数据集充分大，那么使用顺序算法（也被称为在线算法）可能更有价值。顺序算法中，每次只考虑一个数据点，模型的参数在每观测到一个数据点之后进行更新。顺序学习也适用于实时的应用。在实时应用中，数据观测以一个连续的流的方式持续到达，我们必须在观测到所有数据之前就做出预测。

我们可以获得一个顺序学习的算法通过考虑随机梯度下降 (stochastic gradient descent) 也被称为顺序梯度下降 (sequential gradient descent) 的方法。如果误差函数由数据点的和组成 $E = \sum_n E_n$ ，那么在观测到模式 n 之后，随机梯度下降算法使用下式更新参数向量 \mathbf{w}

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} - \eta \nabla E_n \quad (3.22)$$

其中 τ 表示迭代次数， η 是学习率参数。我们稍后会讨论 η 的选择问题。 \mathbf{w} 被初始化为某个起始向量 $\mathbf{w}^{(0)}$ 。对于平方和误差函数 (3.12) 的情形，我们有

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} + \eta(t_n - \mathbf{w}^{(\tau)T} \phi_n) \phi_n \quad (3.23)$$

其中 $\phi_n = \phi(\mathbf{x}_n)$ 。这被称为最小均方 (least-mean-squares) 或者LMS算法。 η 的值需要仔细选择，确保算法收敛 (Bishop and Nabney, 2008)。

3.1.4 正则化最小平方

在1.1节，我们介绍了为误差函数添加正则化项的思想来控制过拟合，因此需要最小化的总的误差函数的形式为

$$E_D(\mathbf{w}) + \lambda E_W(\mathbf{w}) \quad (3.24)$$

其中 λ 是正则化系数，控制数据相关的误差 $E_D(\mathbf{w})$ 和正则化项 $E_W(\mathbf{w})$ 的相对重要性。正则化项的一个最简单的形式为权向量的各个元素的平方和

$$E_W(\mathbf{w}) = \frac{1}{2} \mathbf{w}^T \mathbf{w} \quad (3.25)$$

如果我们考虑平方和误差函数

$$E_D(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^N \{t_n - \mathbf{w}^T \phi(\mathbf{x}_n)\}^2 \quad (3.26)$$

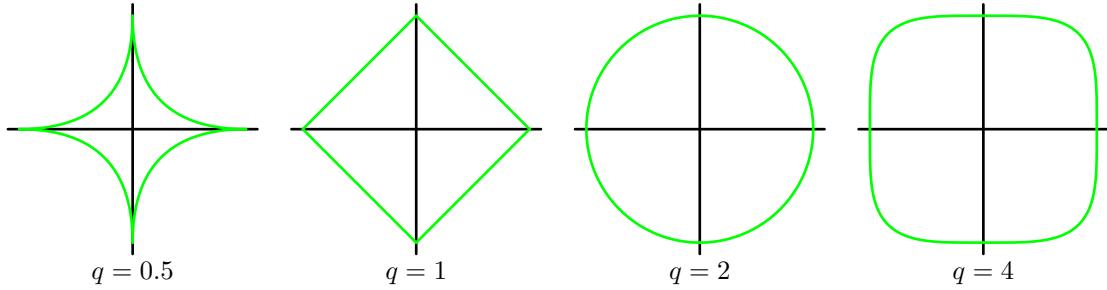


图 3.3: 对于不同的参数 q , 公式 (3.29) 中的正则化项的轮廓线。

那么总误差函数就变成了

$$\frac{1}{2} \sum_{n=1}^N \{t_n - \mathbf{w}^T \phi(\mathbf{x}_n)\}^2 + \frac{\lambda}{2} \mathbf{w}^T \mathbf{w} \quad (3.27)$$

这种对于正则化项的选择方法在机器学习的文献中被称为权值衰减 (weight decay)。这是因为在顺序学习算法中, 它倾向于让权值向零的方向衰减, 除非有数据支持。在统计学中, 它提供了一个参数收缩 (parameter shrinkage) 方法的例子, 因为这种方法把参数的值向零的方向收缩。这种方法的优点在于, 误差函数是 \mathbf{w} 的二次函数, 因此精确的最小值具有解析解。具体来说, 令公式 (3.27) 关于 \mathbf{w} 的梯度等于零, 解出 \mathbf{w} , 我们有

$$\mathbf{w} = (\lambda \mathbf{I} + \Phi^T \Phi)^{-1} \Phi^T \mathbf{t} \quad (3.28)$$

这是最小平方解 (3.15) 的一个简单的扩展。

有时使用一个更加一般的正则化项, 这时正则化的误差函数的形式为

$$\frac{1}{2} \sum_{n=1}^N \{t_n - \mathbf{w}^T \Phi(\mathbf{x}_n)\}^2 + \frac{\lambda}{2} \sum_{j=1}^M |w_j|^q \quad (3.29)$$

其中 $q = 2$ 对应于二次正则化项 (3.27)。图3.3给出了不同 q 值下的正则化函数的轮廓线。

在统计学的文献中, $q = 1$ 的情形被称为套索 (lasso) (Tibshirani, 1996)。它的性质为: 如果 λ 充分大, 那么某些系数 w_j 会变为零, 从而产生了一个稀疏 (sparse) 模型, 这个模型中对应的基函数不起作用。为了说明这一点, 我们首先注意到最小化公式 (3.19) 等价于在满足下面的限制的条件下最小化未正则化的平方和误差函数 (3.12)

$$\sum_{j=1}^M |w_j|^q \leq \eta \quad (3.30)$$

参数 η 要选择一个合适的值。这样, 这两种方法通过拉格朗日乘数法被联系到了一起。稀疏性的来源可以从图3.4中看出来。图3.4给出了在限制条件 (3.30) 下误差函数的最小值。随着 λ 的增大, 越来越多的参数会变为零。

正则化方法通过限制模型的复杂度, 使得复杂的模型能够在有限大小的数据集上进行训练, 而不会产生严重的过拟合。然而, 这样做就使确定最优的模型复杂度的问题从确定合适的基函数数量的问题转移到了确定正则化系数 λ 的合适值的问题上。我们稍后在本章中还会回到这个模型复杂度的问题上。

对于本章的其余部分, 我们将把注意力放在二次正则化项 (3.27) 上, 因为它在实际应用中很重要, 并且数学计算上比较容易。

3.1.5 多个输出

目前为止, 我们已经考虑了单一目标变量 t 的情形。在某些应用中, 我们可能想预测 $K > 1$ 个目标变量。我们把这些目标变量聚集起来, 记作目标向量 \mathbf{t} 。这个问题可以这样解决: 对于 \mathbf{t} 的

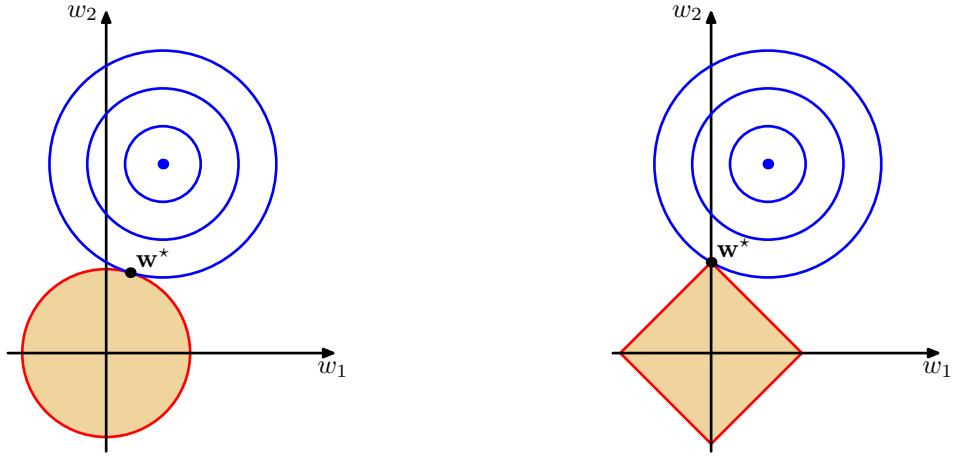


图 3.4: 未正则化的误差函数的轮廓线（蓝色）以及公式 (3.30) 给出的限制区域。左图是 $q = 2$ 的二次正则化项的限制区域，右图是 $q = 1$ 的套索正则化项的限制区域，其中参数向量 w 的值被记作 w^* 。套索正则化项给出了一个稀疏的解，其中 $w_1^* = 0$ 。

每个分量，引入一个不同的基函数集合，从而变成了多个独立的回归问题。但是，一个更有趣的并且更常用的方法是对目标向量的所有分量使用一组相同的基函数来建模，即

$$\mathbf{y}(\mathbf{x}, \mathbf{w}) = \mathbf{W}^T \phi(\mathbf{x}) \quad (3.31)$$

其中 \mathbf{y} 是一个 K 维列向量， \mathbf{W} 是一个 $M \times K$ 的参数矩阵， $\phi(\mathbf{x})$ 是一个 M 为列向量，每个元素为 $\phi_j(\mathbf{x})$ ，并且与之前一样， $\phi_0(\mathbf{x}) = 1$ 。假设我们令目标向量的条件概率分布是一个各向同性的高斯分布，形式为

$$p(\mathbf{t} | \mathbf{x}, \mathbf{W}, \beta) = \mathcal{N}(\mathbf{t} | \mathbf{W}^T \phi(\mathbf{x}), \beta^{-1} \mathbf{I}) \quad (3.32)$$

如果我们有一组观测 t_1, \dots, t_N ，我们可以把这些观测组合为一个 $N \times K$ 的矩阵 \mathbf{T} ，使得矩阵的第 n 行为 \mathbf{t}_n^T 。类似地，我们可以把输入向量 x_1, \dots, x_N 组合为矩阵 \mathbf{X} 。这样，对数似然函数为

$$\begin{aligned} \ln p(\mathbf{T} | \mathbf{X}, \mathbf{W}, \beta) &= \sum_{n=1}^N \ln \mathcal{N}(\mathbf{t}_n | \mathbf{W}^T \phi(\mathbf{x}_n), \beta^{-1} \mathbf{I}) \\ &= \frac{NK}{2} \ln \left(\frac{\beta}{2\pi} \right) - \frac{\beta}{2} \sum_{n=1}^N \|\mathbf{t}_n - \mathbf{W}^T \phi(\mathbf{x}_n)\|^2 \end{aligned} \quad (3.33)$$

与之前一样，我们可以关于 \mathbf{W} 最大化这个函数，可得

$$\mathbf{W}_{ML} = (\Phi^T \Phi)^{-1} \Phi^T \mathbf{T} \quad (3.34)$$

如果我们对于每个目标变量 t_k 考察这个结果，那么我们有

$$\mathbf{w}_k = (\Phi^T \Phi)^{-1} \Phi^T \mathbf{t}_k = \Phi^\dagger \mathbf{t}_k \quad (3.35)$$

这里， \mathbf{t}_k 是一个 N 维列向量，元素为 t_{nk} 其中 $n = 1, \dots, N$ 。因此不同目标变量的回归问题在这里被分解开，并且我们只需要计算一个伪逆矩阵 Φ^\dagger ，这个矩阵是被所有向量 \mathbf{w}_k 所共享的。

推广到具有任意协方差矩阵的一般的高斯噪声分布是很直接的。与之前一样，这个问题可以被分解为 K 个独立的回归问题。这种结果毫不令人惊讶，因为参数 \mathbf{W} 只定义了高斯噪声分布的均值，并且我们从2.3.4节中知道多元高斯分布均值的最大似然解与协方差无关。从现在开始，为了简单起见，我们只考虑单一目标变量 t 的情形。

3.2 偏置-方差分解

目前为止，我们对于回归的线性模型的讨论中，我们假定了基函数的形式和数量都是固定的。正如我们在第1章中看到的那样，如果使用有限规模的数据集来训练复杂的模型，那么使用最大似然方法，或者等价地，使用最小平方方法，会导致严重的过拟合问题。然而，通过限制基函数的数量来避免过拟合问题有一个副作用，即限制了模型描述数据中有趣且重要的规律的灵活性。虽然引入正则化项可以控制具有多个参数的模型的过拟合问题，但是这就产生了一个问题：如何确定正则化系数 λ 的合适的值。同时关于权值 w 和正则化系数 λ 来最小化正则化的误差函数显然不是一个正确的方法，因为这样做会使得 $\lambda = 0$ ，从而产生非正则化的解。

正如我们在之前的章节中看到的那样，过拟合现象确实是最大似然方法的一个不好的性质。但是当我们在使用贝叶斯方法对参数进行求和或者积分时，过拟合现象不会出现。本章中，我们会稍微深入地从贝叶斯观点讨论模型的复杂度。但是，在进行这样的讨论之前，从频率学家的观点考虑一下模型的复杂度问题是很有指导意义的。这种频率学家的观点被称为偏置-方差折中（bias-variance trade-off）。虽然我们将在线性基函数模型中介绍这个概念，因为这样介绍可以使用简单的例子来说明一些基本的思想，但是实际上这种讨论有着更加普遍的适用性。

在1.5.5节，当我们讨论回归问题的决策论时，我们考虑了不同的损失函数。一旦我们知道了条件概率分布 $p(t | \mathbf{x})$ ，每一种损失函数都能够给出对应的最优预测结果。使用最多的一个选择是平方损失函数，此时最优的预测由条件期望（记作 $h(\mathbf{x})$ ）给出，即

$$h(\mathbf{x}) = \mathbb{E}[t | \mathbf{x}] = \int tp(t | \mathbf{x}) dt \quad (3.36)$$

现在，有必要区分决策论中出现的平方损失函数以及模型参数的最大似然估计中出现的平方和误差函数。我们可以使用比最小平方更复杂的方法，例如正则化或者纯粹的贝叶斯方法，来确定条件概率分布 $p(t | \mathbf{x})$ 。为了进行预测，这些方法都可以与平方损失函数相结合。

我们在1.5.5节证明了平方损失函数的期望可以写成

$$\mathbb{E}[L] = \int \{y(\mathbf{x}) - h(\mathbf{x})\}^2 p(\mathbf{x}) d\mathbf{x} + \iint \{h(\mathbf{x}) - t\}^2 p(\mathbf{x}, t) d\mathbf{x} dt \quad (3.37)$$

回忆一下，与 $y(\mathbf{x})$ 无关的第二项，是由数据本身的噪声造成的，表示期望损失能够达到的最小值。第一项与我们对函数 $y(\mathbf{x})$ 的选择有关，我们要找一个 $y(\mathbf{x})$ 的解，使得这一项最小。由于它是非负的，因此我们希望能够让这一项的最小值等于零。如果我们有无限多的数据（以及无限多的计算资源），那么原则上我们能够以任意的精度寻找回归函数 $h(\mathbf{x})$ ，这会给出 $y(\mathbf{x})$ 的最优解。然而，在实际应用中，我们的数据集 \mathcal{D} 只有有限的 N 个数据点，从而我们不能够精确地知道回归函数 $h(\mathbf{x})$ 。

如果我们使用由参数向量 w 控制的函数 $y(\mathbf{x}, w)$ 对 $h(\mathbf{x})$ 建模，那么从贝叶斯的观点来看，我们模型的不确定性是通过 w 的后验概率分布来表示的。但是，频率学家的方法涉及到根据数据集 \mathcal{D} 对 w 进行点估计，然后试着通过下面的思想实验来表示估计的不确定性。假设我们有许多数据集，每个数据集的大小为 N ，并且每个数据集都独立地从分布 $p(t, \mathbf{x})$ 中抽取。对于任意给定的数据集 \mathcal{D} ，我们可以运行我们的学习算法，得到一个预测函数 $y(\mathbf{x}; \mathcal{D})$ 。不同的数据集会给出不同的函数，从而给出不同的平方损失的值。这样，特定的学习算法的表现就可以通过取各个数据集上的表现的平均值来进行评估。

考虑公式 (3.37) 的第一项的被积函数，对于一个特定的数据集 \mathcal{D} ，它的形式为

$$\{y(\mathbf{x}; \mathcal{D}) - h(\mathbf{x})\}^2 \quad (3.38)$$

由于这个量与特定的数据集 \mathcal{D} 相关，因此我们对所有的数据集取平均。如果我们在括号内加上然后减去 $\mathbb{E}_{\mathcal{D}}[y(\mathbf{x}; \mathcal{D})]$ ，然后展开，我们有

$$\begin{aligned} & \{y(\mathbf{x}; \mathcal{D}) - \mathbb{E}_{\mathcal{D}}[y(\mathbf{x}; \mathcal{D})] + \mathbb{E}_{\mathcal{D}}[y(\mathbf{x}; \mathcal{D})] - h(\mathbf{x})\}^2 \\ &= \{y(\mathbf{x}; \mathcal{D}) - \mathbb{E}_{\mathcal{D}}[y(\mathbf{x}; \mathcal{D})]\}^2 + \{\mathbb{E}_{\mathcal{D}}[y(\mathbf{x}; \mathcal{D})] - h(\mathbf{x})\}^2 \\ & \quad + 2\{y(\mathbf{x}; \mathcal{D}) - \mathbb{E}_{\mathcal{D}}[y(\mathbf{x}; \mathcal{D})]\}\{\mathbb{E}_{\mathcal{D}}[y(\mathbf{x}; \mathcal{D})] - h(\mathbf{x})\} \end{aligned} \quad (3.39)$$

我们现在关于 \mathcal{D} 求期望，然后注意到最后一项等于零，可得

$$\begin{aligned} \mathbb{E}_{\mathcal{D}}[\{y(\mathbf{x}; \mathcal{D}) - h(\mathbf{x})\}^2] \\ = \underbrace{\{\mathbb{E}_{\mathcal{D}}[y(\mathbf{x}; \mathcal{D})] - h(\mathbf{x})\}^2}_{\text{(偏置)}^2} + \underbrace{\mathbb{E}_{\mathcal{D}}[\{y(\mathbf{x}; \mathcal{D}) - \mathbb{E}_{\mathcal{D}}[y(\mathbf{x}; \mathcal{D})]\}^2]}_{\text{方差}} \end{aligned} \quad (3.40)$$

我们看到， $y(\mathbf{x}; \mathcal{D})$ 与回归函数 $h(\mathbf{x})$ 的差的平方的期望可以表示为两项的和。第一项，被称为平方偏置（bias），表示所有数据集的平均预测与预期的回归函数之间的差异。第二项，被称为方差（variance），度量了对于单独的数据集，模型所给出的解在平均值附近波动的情况，因此也就度量了函数 $y(\mathbf{x}; \mathcal{D})$ 对于特定的数据集的选择的敏感程度。稍后我们会考虑一个简单的例子，来直观地说明这些概念。

目前为止，我们已经考虑了单一输入变量 \mathbf{x} 的情形。如果我们把这个展开式带回到公式(3.37)中，那么我们就得到了下面的对于期望平方损失的分解

$$\text{期望损失} = \text{偏置}^2 + \text{方差} + \text{噪声} \quad (3.41)$$

其中

$$\text{偏置}^2 = \int \{\mathbb{E}_{\mathcal{D}}[y(\mathbf{x}; \mathcal{D})] - h(\mathbf{x})\}^2 p(\mathbf{x}) d\mathbf{x} \quad (3.42)$$

$$\text{方差} = \int \mathbb{E}_{\mathcal{D}}[\{y(\mathbf{x}; \mathcal{D}) - \mathbb{E}_{\mathcal{D}}[y(\mathbf{x}; \mathcal{D})]\}^2] p(\mathbf{x}) d\mathbf{x} \quad (3.43)$$

$$\text{噪声} = \iint \{h(\mathbf{x}) - t\}^2 p(\mathbf{x}, t) d\mathbf{x} dt \quad (3.44)$$

现在，偏置和方差指的是积分后的量。

我们的目标是最小化期望损失，它可以分解为（平方）偏置、方差和一个常数噪声项的和。正如我们将看到的那样，在偏置和方差之间有一个折中。对于非常灵活的模型来说，偏置较小，方差较大。对于相对固定的模型来说，偏置较大，方差较小。有着最优预测能力的模型时在偏置和方差之间取得最优的平衡的模型。这里通过第1章讨论过的正弦数据集来说明。我们产生了100个数据集合，每个集合都包含 $N = 25$ 个数据点，都是独立地从正弦曲线 $h(x) = \sin(2\pi x)$ 抽取的。数据集的编号为 $l = 1, \dots, L$ ，其中 $L = 100$ ，并且对于每个数据集 $\mathcal{D}^{(l)}$ ，我们通过最小化正则化的误差函数(3.27)拟合了一个带有24个高斯基函数的模型，然后给出了预测函数 $y^{(l)}(x)$ ，如图3.5所示。第一行对应着较大的正则化系数 λ ，这样的模型的方差很小（因为左侧图中的红色曲线看起来很相似），但是偏置很大（因为右侧图中的两条曲线看起来相当不同）。相反，在最后一行，正则化系数 λ 很小，这样模型的方差较大（因为左侧图中的红色曲线变化性相当大），但是偏置很小（因为平均拟合的结果与原始正弦曲线十分吻合）。注意，把 $M = 25$ 这种复杂模型的多个解进行平均，会产生对于回归函数非常好的拟合，这表明求平均是一个很好的步骤。事实上，将多个解加权平均是贝叶斯方法的核心，虽然这种求平均针对的是参数的后验分布，而不是针对多个数据集。

对于这个例子，我们也可以定量地考察偏置-方差折中。平均预测由下式求出

$$\bar{y}(x) = \frac{1}{L} \sum_{l=1}^L y^{(l)}(x) \quad (3.45)$$

并且积分后的平方偏置以及积分后的方差为

$$\text{偏置}^2 = \frac{1}{N} \sum_{n=1}^N \{\bar{y}(x_n) - h(x_n)\}^2 \quad (3.46)$$

$$\text{方差} = \frac{1}{N} \sum_{n=1}^N \frac{1}{L} \sum_{l=1}^L \{y^{(l)}(x_n) - \bar{y}(x_n)\}^2 \quad (3.47)$$

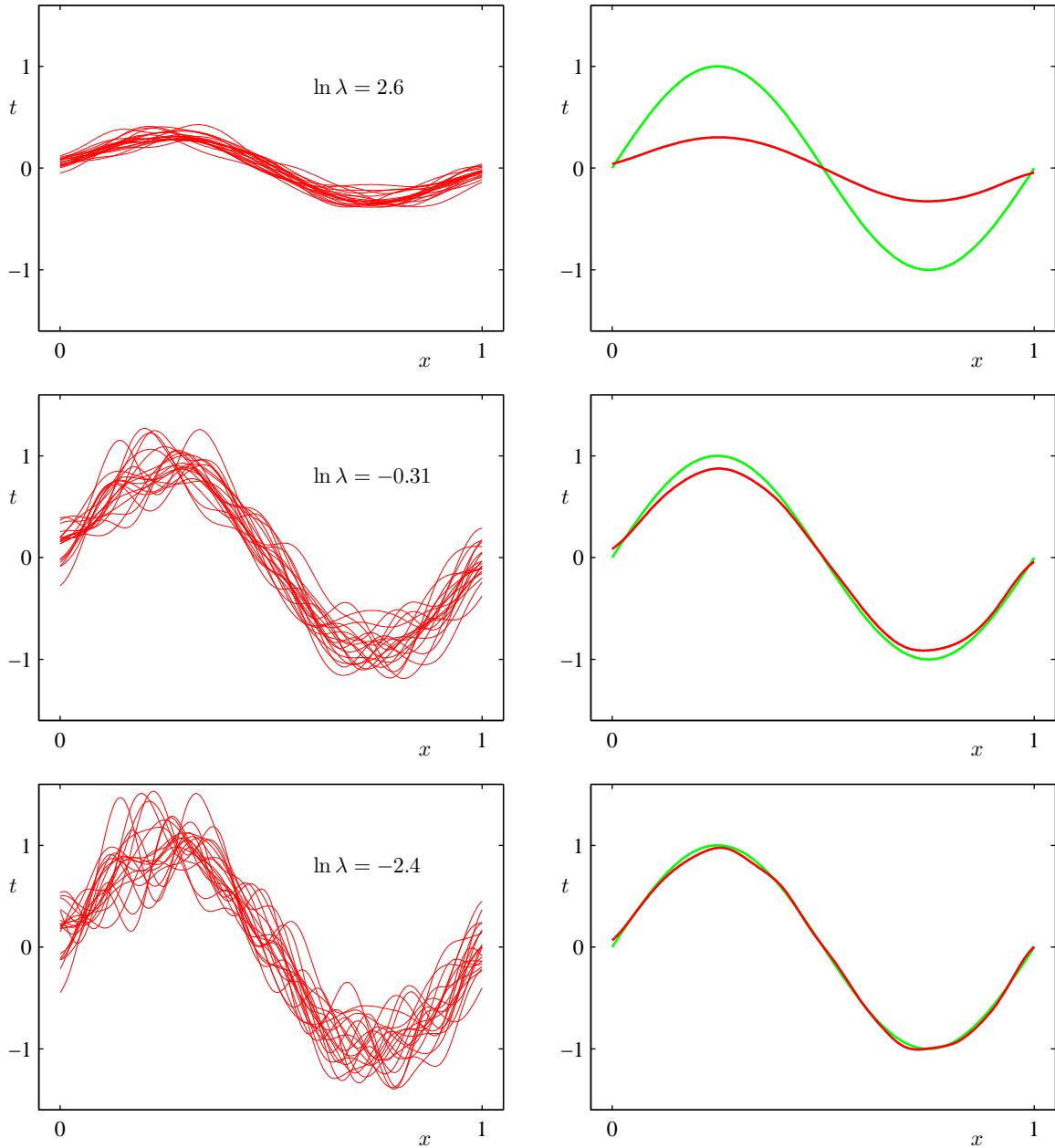


图 3.5: 模型复杂度对于偏置和方差的依赖的说明。模型的复杂度由正则化参数 λ 控制，数据集是第1章中的正弦数据。有 $L = 100$ 个数据集，每个数据集有 $N = 25$ 个数据点，每个模型有24个高斯基函数，从而参数的总数为 $M = 25$ （包括偏置参数）。左侧一列给出了对于不同的 $\ln \lambda$ 值，根据数据集拟合模型的结果。为了清晰起见，我们只给出了100个拟合模型中的20个。右侧一列给出了对应的100个拟合的均值（红色）以及用于生成数据集的正弦函数（绿色）。

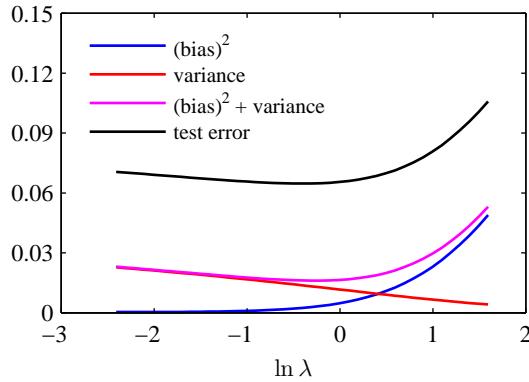


图 3.6: 平方偏置和方差的图像, 以及它们的加和, 对应于图3.5给出的结果。同样给出的还有大小为1000个数据点的测试数据的平均测试误差。 $(\text{偏置})^2 + \text{方差}$ 的最小值出现在 $\ln \lambda = -0.31$ 的位置, 它接近于在测试数据上取得最小误差的位置。

其中由概率分布 $p(x)$ 加权的 x 的积分由来自那个概率分布的有限数据点的加和来近似。图3.6给出了这些量以及它们的求和关于 $\ln \lambda$ 的函数图像。我们看到, 小的 λ 使得模型对于各个数据集里的噪声的拟合效果非常好, 导致了较大的方差。相反, 大的 λ 把权值参数拉向零, 导致了较大的偏置。

虽然偏置-方差分解能够从频率学家的角度对模型的复杂度提供一些有趣的认识, 但是它的实用价值很有限。这是因为偏置-方差分解依赖于对所有的数据集求平均, 而在实际应用中我们只有一个观测数据集。如果我们有大量的已知规模的独立的训练数据集, 那么我们最好的方法是把它们组合成一个大的训练集, 这显然会降低给定复杂度的模型的过拟合程度。

由于有这么多局限性, 因此我们在下一节里将讨论线性基函数模型的贝叶斯观点。它不仅提供了对于过拟合现象的深刻认识, 还提出了解决模型复杂度问题的实用的技术。

3.3 贝叶斯线性回归

在我们讨论使用最大似然方法设置线性回归模型的参数时, 我们已经看到由基函数的数量控制的模型的复杂度需要根据数据集的规模进行调整。为对数似然函数增加一个正则化项意味着模型的复杂度可以通过正则化系数的值进行控制, 虽然基函数的数量和形式的选择仍然对于确定模型的整体行为十分重要。

这就产生了对于特定的应用确定合适的模型复杂度的问题。这个问题不能简单地通过最大化似然函数来确定, 因为这总会产生过于复杂的模型和过拟合现象。独立的额外数据能够用来确定模型的复杂度, 正如1.3节所说的那样, 但是这需要较大的计算量, 并且浪费了有价值的数据。因此我们转而考虑线性回归的贝叶斯方法, 这会避免最大似然的过拟合问题, 也会引出使用训练数据本身确定模型复杂度的自动化方法。与之前一样, 为了简单起见, 我们只考虑单一目标变量 t 的情形。对于多个目标变量情形的推广是很直接的, 与3.1.5节的讨论很类似。

3.3.1 参数分布

关于线性拟合的贝叶斯方法的讨论, 我们首先引入模型参数 w 的先验概率分布。现在这个阶段, 我们把噪声精度参数 β 当做已知常数。首先, 我们注意到, 由公式 (3.10) 定义的似然函数 $p(\mathbf{t} | w)$ 是 w 的二次函数的指数形式。于是对应的共轭先验是高斯分布, 形式为

$$p(w) = \mathcal{N}(w | \mathbf{m}_0, \mathbf{S}_0) \quad (3.48)$$

均值为 \mathbf{m}_0 , 协方差为 \mathbf{S}_0 。

接下来我们计算后验分布, 它正比于似然函数与先验分布的乘积。由于共轭高斯先验分布的选择, 后验分布也将是高斯分布。我们可以对指数项进行配平方, 然后使用归一化的高斯分

布的标准结果找到归一化系数，这样就计算出了后验分布的形式。但是，我们在推导公式 (2.116) 已经进行了必要的工作，这让我们能够直接写出后验概率分布的形式

$$p(\mathbf{w} \mid \mathbf{t}) = \mathcal{N}(\mathbf{w} \mid \mathbf{m}_N, \mathbf{S}_N) \quad (3.49)$$

其中

$$\mathbf{m}_N = \mathbf{S}_N^{-1} (\mathbf{S}_0^{-1} \mathbf{m}_0 + \beta \Phi^T \mathbf{t}) \quad (3.50)$$

$$\mathbf{S}_N^{-1} = \mathbf{S}_0^{-1} + \beta \Phi^T \Phi \quad (3.51)$$

注意，由于后验分布是高斯分布，它的众数恰好与它的均值相同。因此最大后验权向量的结果就是 $\mathbf{w}_{MAP} = \mathbf{m}_N$ 。如果我们考虑一个无限宽的先验 $\mathbf{S}_0 = \alpha^{-1} \mathbf{I}$ ，其中 $\alpha \rightarrow 0$ ，那么后验概率分布的均值 \mathbf{m}_N 就变成了由公式 (3.15) 给出的最大似然值 \mathbf{w}_{ML} 。类似地，如果 $N = 0$ ，那么后验概率分布就变成了先验分布。此外，如果数据点是顺序到达的，那么任何一个阶段的后验概率分布都可以看成后续数据点的先验。此时新的后验分布再次由公式 (3.49) 给出。

对于本章的剩余部分，为了简化起见，我们将考虑高斯先验的一个特定的形式。具体来说，我们考虑零均值各向同性高斯分布。这个分布由一个精度参数 α 控制，即

$$p(\mathbf{w} \mid \alpha) = \mathcal{N}(\mathbf{w} \mid \mathbf{0}, \alpha^{-1} \mathbf{I}) \quad (3.52)$$

对应的 \mathbf{w} 的后验概率分布由公式 (3.49) 给出，其中

$$\mathbf{m}_N = \beta \mathbf{S}_N \Phi^T \mathbf{t} \quad (3.53)$$

$$\mathbf{S}_N^{-1} = \alpha \mathbf{I} + \beta \Phi^T \Phi \quad (3.54)$$

后验概率分布的对数由对数似然函数与先验的对数求和的方式得到。它是 \mathbf{w} 的函数，形式为

$$\ln p(\mathbf{w} \mid \mathbf{t}) = -\frac{\beta}{2} \sum_{n=1}^N \{t_n - \mathbf{w}^T \phi(\mathbf{x}_n)\}^2 - \frac{\alpha}{2} \mathbf{w}^T \mathbf{w} + \text{常数} \quad (3.55)$$

于是，后验分布关于 \mathbf{w} 的最大化等价于对平方和误差函数加上一个二次正则项进行最小化。正则项对应于公式 (3.27)，其中 $\lambda = \frac{\alpha}{\beta}$ 。

我们可以使用直线拟合的简单的例子来说明线性基函数的贝叶斯学习过程，以及后验概率分布的顺序更新过程。考虑一个单一输入变量 x ，一个单一目标变量 t ，以及一个形式为 $y(x, \mathbf{w}) = w_0 + w_1 x$ 的线性模型。由于这个模型只有两个可调节参数，因此我们可以直接在参数空间中画出先验分布和后验分布。我们从函数 $f(x, \mathbf{a}) = a_0 + a_1 x$ 中人工生成数据，其中 $a_0 = -0.3$ 且 $a_1 = 0.5$ 。生成数据的方法为：首先从均匀分布 $U(x \mid -1, 1)$ 中选择 x_n 的值，然后计算 $f(x_n, \mathbf{a})$ ，最后增加一个标准差为 0.2 的高斯噪声，得到目标变量 t_n 。我们的目标是从这样的数据中恢复 a_0 和 a_1 的值，并且我们想研究模型对于数据集规模的依赖关系。这里我们假设噪声方差是已知的，因此我们把精度参数设置为它的真实值 $\beta = (\frac{1}{0.2})^2 = 25$ 。类似地，我们把 α 固定为 2.0。我们稍后会简短地讨论从训练数据中确定 α 和 β 的值的策略。图 3.7 给出了当数据集的规模增加时贝叶斯学习的结果，还展示了贝叶斯学习的顺序本质，即当新数据点被观测到的时候，当前的后验分布变成了先验分布。花时间仔细研究一下这幅图是很值得的，因为它说明了贝叶斯推断的一些重要的概念。这张图的第一行对应于观测到任何数据点之前的情况，给出了 \mathbf{w} 空间的先验概率分布的图像，以及函数 $y(x, \mathbf{w})$ 的六个样本，这六个样本的 \mathbf{w} 都是从先验概率分布中抽取的。在第二行，我们看到了观测到一个数据点之后的情形。数据点的位置 (x, t) 由右侧一列中的蓝色圆圈表示。左侧一列是对于这个数据点的似然函数 $p(t, \mathbf{w})$ 关于 \mathbf{w} 的函数图像。注意，似然函数提供了一个温和的限制，即直线必须穿过数据点附近的位置，其中附近位置的范围由噪声精度 β 确定。为了进行对比，用来生成数据集的真实参数值 $a_0 = -0.3$ 以及 $a_1 = 0.5$ 在图 3.7 的左侧一列被标记为白色十字。如果我们把这个似然函数与第一行的先验概率相乘，然后归一化，我们就得到了第二行中间的图给出的后验概率分布。从这个后验概率分布中抽取 \mathbf{w} 的样本，对应的回归函数 $y(x, \mathbf{w})$ 被画在了右侧一列的途中。注意，这些样本直线全部穿过数据点的附近位置。这张图的第三行展示了观测到第二个数据点的效果。与之前一样，这个数据点由右侧一列的蓝色圆圈表示。第二个数据点自身对应的似然函数在左侧一列的图中给出。如果我们

把这个似然函数与第二行的后验概率分布相乘，我们就得到了第三行中间一列的图给出的后验概率分布。注意，这个后验概率分布与我们将原始的先验分布结合两个数据点的似然函数得到的后验概率分布完全相同。现在，后验概率分布被两个数据点影响。由于两个点足够定义一条直线，因此目前已经得到了相对较好的后验概率分布。从这个后验分布中抽取的样本产生了第三列中红色的函数，我们看到这些函数同时穿过了两个数据点的附近。第四行展示了观测到20个数据点的效果。左侧的图展示了第20个数据点自身的似然函数，中间的图展示了融合了20次观测信息的后验概率分布。注意与第三行相比，这个后验概率分布变得更加尖锐。在无穷多个数据点的极限情况下，后验概率分布会变成一个Delta函数。这个函数的中心是用白色十字标记出的真实参数值。

也可以考虑参数的其他形式的先验分布。例如，我们可以推广高斯先验分布，得到

$$p(\mathbf{w} | \alpha) = \left[\frac{q}{2} \left(\frac{\alpha}{2} \right)^{\frac{1}{q}} \frac{1}{\Gamma(\frac{1}{q})} \right]^M \exp \left(-\frac{\alpha}{2} \sum_{j=0}^{M-1} |w_j|^q \right) \quad (3.56)$$

其中 $q = 2$ 的情形对应于高斯分布，并且只有在这种情形下的先验分布才是公式 (3.10) 给出的似然函数的共轭先验。找到 \mathbf{w} 的后验概率分布的最大值对应于找到正则化误差函数 (3.29) 的最小值。在高斯先验的情况下，后验概率分布的众数等于均值，但是如果 $q \neq 2$ ，这个性质就不成立了。

3.3.2 预测分布

在实际应用中，我们通常感兴趣的不是 \mathbf{w} 本身的值，而是对于新的 x 值预测出 t 的值。这需要我们计算出预测分布 (predictive distribution)，定义为

$$p(t | \mathbf{t}, \alpha, \beta) = \int p(t | \mathbf{w}, \beta) p(\mathbf{w} | \mathbf{t}, \alpha, \beta) d\mathbf{w} \quad (3.57)$$

其中 \mathbf{t} 是训练数据的目标变量的值组成的向量。并且，为了简化记号，我们在右侧省略了条件概率中出现的输入向量。目标变量的条件概率分布 $p(t | \mathbf{w}, \mathbf{w}, \beta)$ 由公式 (3.8) 给出，后验分布由公式 (3.49) 给出。我们看到公式 (3.57) 涉及到两个高斯分布的卷积，因此使用2.3.3节的公式 (2.115) 的结果，我们看到预测分布的形式为

$$p(t | \mathbf{x}, \mathbf{t}, \alpha, \beta) = \mathcal{N}(t | \mathbf{m}_N^T \phi(\mathbf{x}), \sigma_N^2(\mathbf{x})) \quad (3.58)$$

其中预测分布的方差 $\sigma_N^2(\mathbf{x})$ 为

$$\sigma_N^2(\mathbf{x}) = \frac{1}{\beta} + \phi(\mathbf{x})^T \mathbf{S}_N \phi(\mathbf{x}) \quad (3.59)$$

公式 (3.59) 的第一项表示数据中的噪声，而第二项反映了与参数 \mathbf{w} 关联的不确定性。由于噪声和 \mathbf{w} 的分布是相互独立的高斯分布，因此它们的值是可以相加的。注意，当额外的数据点被观测到的时候，后验概率分布会变窄。从而可以证明出 $\sigma_{N+1}^2(\mathbf{x}) \leq \sigma_N^2(\mathbf{x})$ (Qazaz et al., 1997)。在极限 $N \rightarrow \infty$ 的情况下，公式 (3.59) 的第二项趋于零，从而预测分布的方差只与参数 β 控制的具有可加性的噪声有关。

为了说明贝叶斯线性回归模型的预测分布，让我们回到第1.1节人工生成的正弦数据集。在图3.8中，我们调整一个由高斯基函数线性组合的模型，使其适应于不同规模的数据集，然后观察对应的后验概率分布。这里，绿色曲线对应着产生数据点的函数 $\sin(2\pi x)$ (带有附加的高斯噪声)。大小为 $N = 1, N = 2, N = 4$ 和 $N = 25$ 的数据集在四幅图中用蓝色圆圈表示。对于每幅图，红色曲线是对应的高斯预测分布的均值，红色阴影区域是均值两侧的一个标准差范围的区域。注意，预测的不确定性依赖于 x ，并且在数据点的邻域内最小。还要注意，不确定性的程度随着观测到的数据点的增多而逐渐减小。

图3.8中的图像只给出了每个点处的预测方差与 x 的函数关系。为了更加深刻地认识对于不同的 x 值的预测之间的协方差，我们可以从 \mathbf{w} 的后验概率分布中抽取样本，然后画出对应的函数 $y(x, \mathbf{w})$ ，如图3.9所示。



图 3.7: 顺序贝叶斯学习的例子。模型是一个简单的线性模型，形式为 $y(x, \mathbf{w}) = w_0 + w_1 x$ 。本图的详细描述见正文。

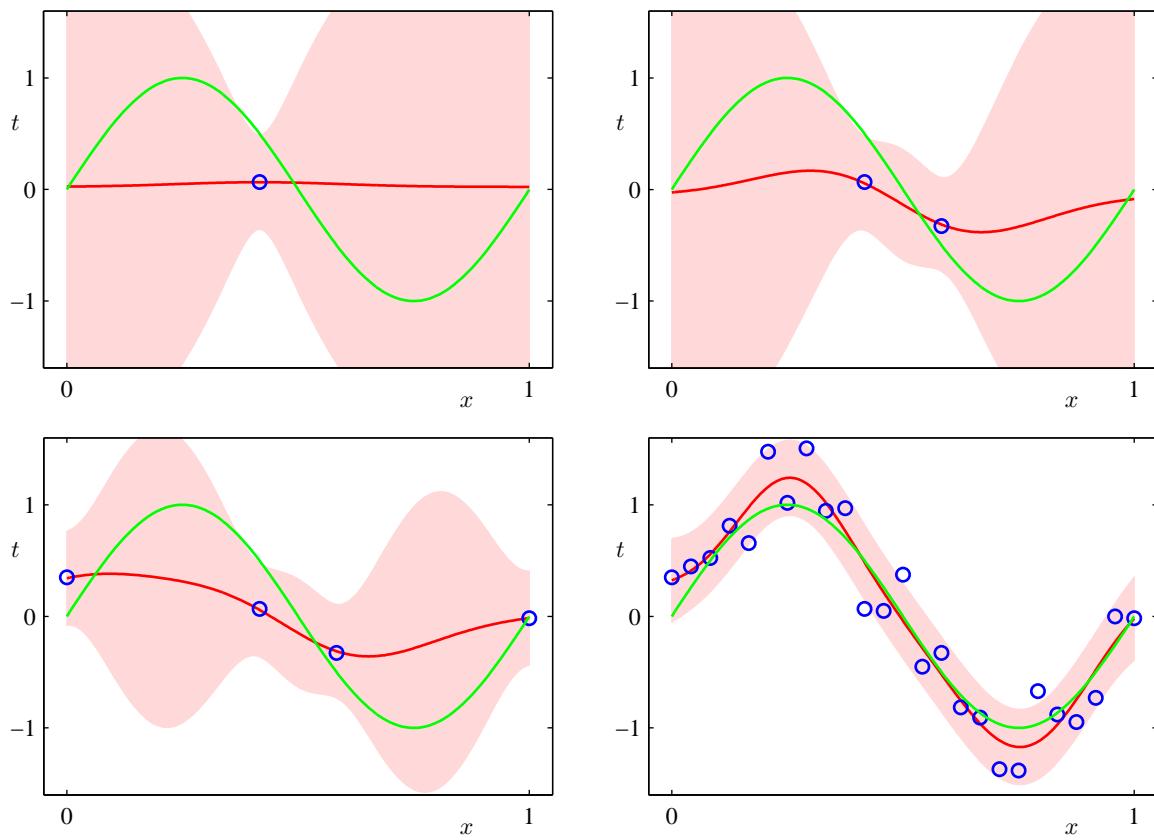


图 3.8: 包含9个高斯基函数 (3.4) 的模型的预测分布 (3.58) , 使用了1.1节的人工生成的正弦数据集。详细的讨论见正文。

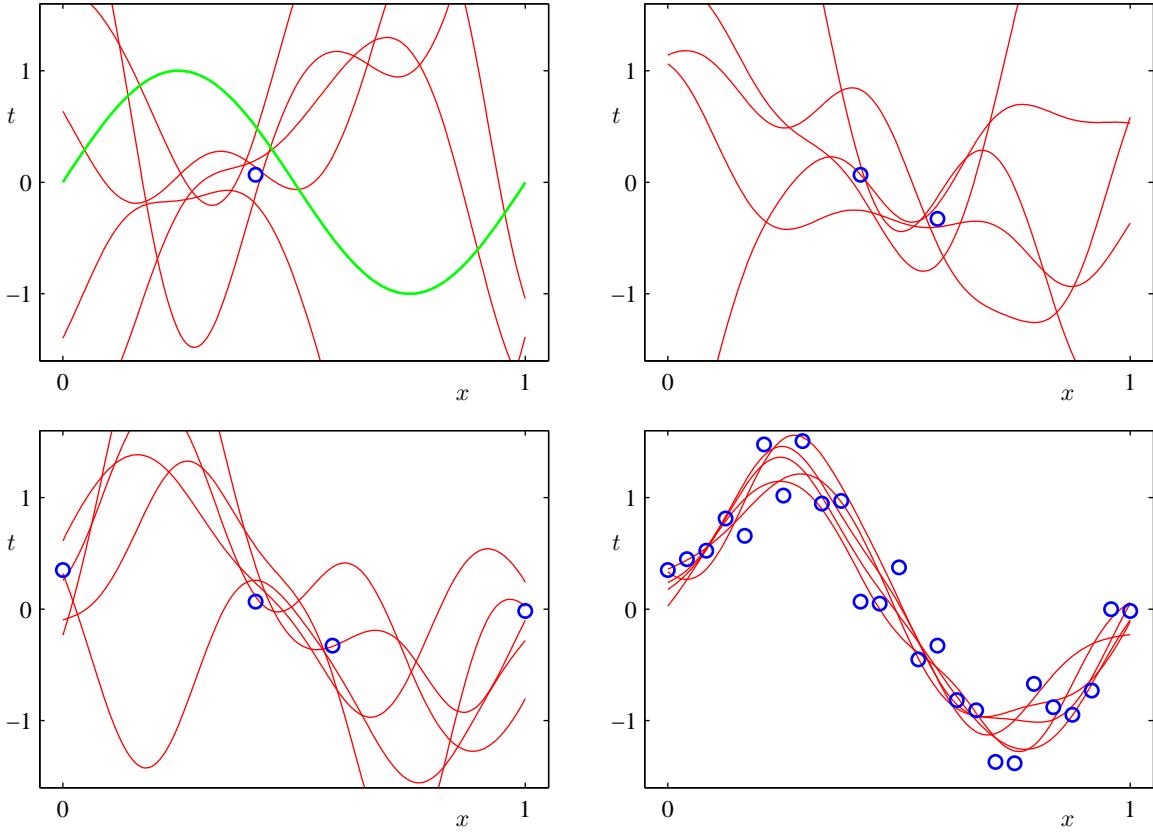


图 3.9: 函数 $y(x, \mathbf{w})$ 的图像, 使用了服从 \mathbf{w} 上的后验概率分布的样本, 对应于图3.8。

如果我们使用局部的基函数 (例如高斯基函数), 那么在距离基函数中心比较远的区域, 公式 (3.59) 给出的预测方差的第二项的贡献将会趋于零, 只剩下噪声的贡献 β^{-1} 。因此, 当对基函数所在的区域之外的区域进行外插的时候, 模型对于它做出的预测会变得相当确定, 这通常不是我们想要的结果。通过使用被称为高斯过程的另一种贝叶斯回归方法, 这个问题可以被避免。

注意, 如果 \mathbf{w} 和 β 都被当成未知的, 那么根据 2.3.6 节的讨论, 我们可以引入一个由高斯-Gamma 分布定义的共轭先验分布 $p(\mathbf{w}, \beta)$ (Denison et al., 2002)。在这种情况下, 预测分布是一个学生 t 分布。

3.3.3 等价核

公式 (3.53) 给出的线性基函数模型的后验均值解有一个有趣的解释, 这个解释为核方法 (包括高斯过程) 提供了舞台。如果我们把公式 (3.53) 代入表达式 (3.3), 我们看到预测均值可以写成下面的形式

$$y(\mathbf{x}, \mathbf{m}_N) = \mathbf{m}_N^T \phi(\mathbf{x}) = \beta \phi(\mathbf{x})^T \mathbf{S}_N \Phi^T \mathbf{t} = \sum_{n=1}^N \beta \phi(\mathbf{x})^T \mathbf{S}_N \phi(\mathbf{x}_n) t_n \quad (3.60)$$

其中 \mathbf{S}_N 由公式 (3.51) 定义。因此在点 \mathbf{x} 处的预测均值由训练集目标变量 t_n 的线性组合给出, 即

$$y(\mathbf{x}, \mathbf{m}_N) = \sum_{n=1}^N k(\mathbf{x}, \mathbf{x}_n) t_n \quad (3.61)$$

其中, 函数

$$k(\mathbf{x}, \mathbf{x}') = \beta \phi(\mathbf{x})^T \mathbf{S}_N \phi(\mathbf{x}') \quad (3.62)$$

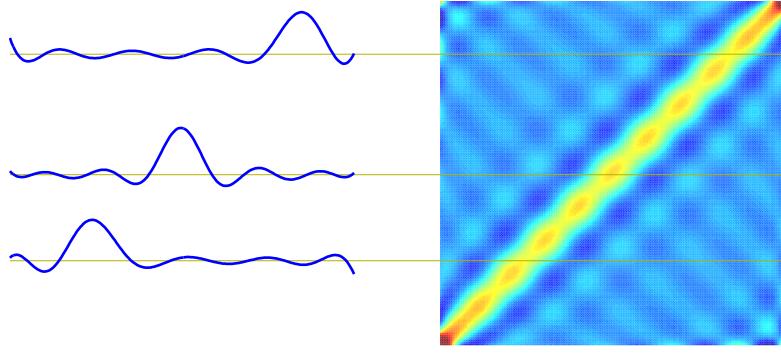


图 3.10: 图 3.1 中的高斯基函数的等价核 $k(x, x')$, 图中给出了 x 关于 x' 的图像, 以及通过这个矩阵的三个切片, 对应于三个不同的 x 值。用来生成这个核的数据集由 x 的 200 个值组成, x 均匀地分布在区间 $(-1, 1)$ 中。



图 3.11: $x = 0$ 时的等价核 $k(x, x')$ 的例子, 图中给出了关于 x' 的函数图像。左图对应于多项式基函数, 右图对应于 sigmoid 基函数, 如图 3.1 所示。注意, 这些是 x' 的局部函数, 即使对应的基函数不是局部的。

被称为平滑矩阵 (smoother matrix) 或者等价核 (equivalent kernel)。像这样的回归函数, 通过对训练集里目标值进行线性组合做预测, 被称为线性平滑 (linear smoother)。注意, 等价核依赖于来自数据集的输入值 \mathbf{x}_n , 因为这些输入值出现在了 \mathbf{S}_N 的定义中。图 3.10 给出了高斯基函数的情形下的等价核。图中给出了三个不同的 x 值的情况下, 核函数 $k(x, x')$ 与 x' 的函数关系。我们看到, 它们在局限在 x 的周围, 因此在 x 处的预测分布的均值 $y(\mathbf{x}, \mathbf{m}_N)$ 可以通过对目标值加权组合的方式获得。距离 x 较近的数据点可以赋一个较高的权值, 而距离 x 较远的数据点可以赋一个较低的权值。直观来看, 与远处的证据相比, 我们把局部的证据赋予更高的权值似乎是更合理的。注意, 这种局部性不仅对于局部的高斯基函数成立, 对于非局部的多项式基函数和 sigmoid 基函数也成立, 如图 3.11 所示。

我们还可以获得更多的关于等价核的认识。考虑 $y(\mathbf{x})$ 和 $y(\mathbf{x}')$ 的协方差

$$\begin{aligned}\text{cov}[y(\mathbf{x}), y(\mathbf{x}')] &= \text{cov}[\phi(\mathbf{x})^T \mathbf{w}, \mathbf{w}^T \phi(\mathbf{x}')] \\ &= \phi(\mathbf{x})^T \mathbf{S}_N \phi(\mathbf{x}') = \beta^{-1} k(\mathbf{x}, \mathbf{x}')\end{aligned}\quad (3.63)$$

其中我们使用了公式 (3.49) 和公式 (3.62)。根据等价核的形式, 我们可以看到在附近的点处的预测均值相关性较高, 而对于距离较远的点对, 相关性就较低。

图 3.8 给出的预测分布让我们能够可视化各个点处预测的不确定性 (由公式 (3.59) 控制)。然而, 通过从 \mathbf{w} 的后验分布中抽取样本并且在图 3.9 中画出对应的模型函数 $y(\mathbf{x}, \mathbf{w})$, 我们可视化了后验概率分布中位于两个 (或者更多) x 值处的 y 值之间的不确定性 (由等价核控制)。

用核函数表示线性回归给出了解决回归问题的另一种方法。我们不引入一组基函数 (它隐式地定义了一个等价的核), 而是直接定义一个局部的核函数, 然后在给定观测数据集的条件下, 使用这个核函数对新的输入变量 \mathbf{x} 做预测。这就引出了用于回归问题 (以及分类问题) 的一个很实用的框架, 被称为高斯过程 (Gaussian process)。这将在 6.4 节详细讨论。

我们已经看到, 一个等价核定义了模型的权值。通过这个权值, 训练数据集里的目标值被组

合，然后对新的 x 值做预测。可以证明这些权值的和等于1，即

$$\sum_{n=1}^N k(\mathbf{x}, \mathbf{x}_n) = 1 \quad (3.64)$$

对于所有的 x 值都成立。这个直观上令人兴奋的结果可以很容易地用非形式化的方式证明出来。我们注意到，这个加和等价于对于所有的 n 都有 $t_n = 1$ 的目标数据集的预测均值 $\hat{y}(\mathbf{x})$ 。假设基函数是线性独立的，且数据点的数量多于基函数的数量，并且其中一个基函数是常量（对应于偏置参数），那么很明显我们可以精确地拟合训练数据，因此预测均值就是简单的 $\hat{y}(\mathbf{x}) = 1$ ，这样我们就可以得到共识 (3.64)。注意，核函数可以为负也可以为正，因此它虽然满足加和限制，但是对应的预测未必是训练集的目标值的凸组合。

最后，我们注意到，公式 (3.62) 给出的等价核满足一般的核函数共有的一个重要性质，即它可以表示为非线性函数的向量 $\psi(\mathbf{x})$ 的内积的形式，即

$$k(\mathbf{x}, \mathbf{z}) = \psi(\mathbf{x})^T \psi(\mathbf{z}) \quad (3.65)$$

其中 $\psi(\mathbf{x}) = \beta^{\frac{1}{2}} \mathbf{S}_N^{\frac{1}{2}} \psi(\mathbf{x})$ 。

3.4 贝叶斯模型比较

在第1章中，我们强调了过拟合的问题，也介绍了通过使用交叉验证的方法，来设置正则化参数的值，或者从多个模型中选择合适的一个。这里，我们从贝叶斯的角度考虑模型选择的问题。在本节中，我们的讨论是非常一般的。之后在3.5节，我们将会看到这些想法是如何应用到线性回归的正则化参数确定的问题中的。

正如我们将看到的那样，与最大似然估计相关联的过拟合问题可以通过对模型的参数进行求和或者积分的方式（而不是进行点估计）来避免。这样，模型可以直接在训练数据上进行比较，而不需要验证集。这使得所有的数据都能够被用于训练，并且避免了交叉验证当中每个模型要运行多次训练过程的问题。它也让多个复杂度参数可以在训练过程中被确定。例如，在第7章，我们会介绍相关向量机 (relevance vector machine)，这是一个贝叶斯模型，它对于每个训练数据点都有一个复杂度参数。

模型比较的贝叶斯观点仅仅涉及到使用概率来表示模型选择的不确定性，以及恰当地使用概率的加和规则和乘积规则。假设我们想比较 L 个模型 $\{\mathcal{M}_i\}$ ，其中 $i = 1, \dots, L$ 。这里，一个模型指的是观测数据 \mathcal{D} 上的概率分布。在多项式曲线拟合的问题中，概率分布被定义在目标值 \mathbf{t} 上，而输入值 \mathbf{X} 被假定为已知的。其他类型的模型定义了 \mathbf{X} 和 \mathbf{t} 上的联合分布。我们会假设数据是由这些模型中的一个生成的，但是我们不知道究竟是哪一个。我们的不确定性通过先验概率分布 $p(\mathcal{M}_i)$ 表示。给定一个训练数据集 \mathcal{D} ，我们想估计后验分布

$$p(\mathcal{M}_i | \mathcal{D}) \propto p(\mathcal{M}_i) p(\mathcal{D} | \mathcal{M}_i) \quad (3.66)$$

先验分布让我们能够表达不同模型之间的优先级。让我们简单地假设所有的模型都有相同的先验概率。比较有意思的一项是模型证据 (model evidence) $p(\mathcal{D} | \mathcal{M}_i)$ ，它表达了数据展现出的不同模型的优先级，我们稍后会稍微详细地考察这一项。模型证据有时也被称为边缘似然 (marginal likelihood)，因为它可以被看做在模型空间中的似然函数，在这个空间中参数已经被求和或者积分。两个模型的模型证据的比值 $\frac{p(\mathcal{D} | \mathcal{M}_i)}{p(\mathcal{D} | \mathcal{M}_j)}$ 被称为贝叶斯因子 (Bayes factor) (Kass and Raftery, 1995)。

一旦我们知道了模型上的后验概率分布，那么根据概率的加和规则与乘积规则，预测分布为

$$p(t | \mathbf{x}, \mathcal{D}) = \sum_{i=1}^L p(t | \mathbf{x}, \mathcal{M}_i, \mathcal{D}) p(\mathcal{M}_i | \mathcal{D}) \quad (3.67)$$

这是混合分布 (mixture distribution) 的一个例子。这个公式中，整体的预测分布由下面的方式获得：对各个模型的预测分布 $p(t | \mathbf{x}, \mathcal{M}_i, \mathcal{D})$ 求加权平均，权值为这些模型的后验概

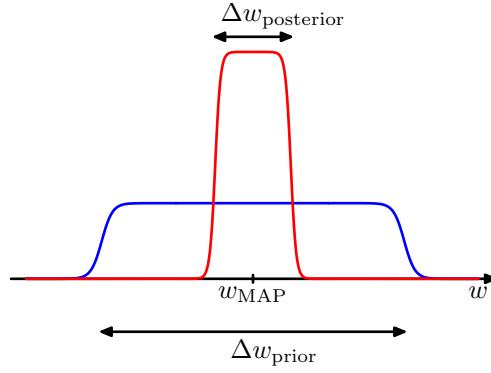


图 3.12: 我们可以粗略地近似模型证据, 如果我们假设参数上的后验概率分布在众数 w_{MAP} 附近有一个尖峰。

率 $p(\mathcal{M}_i | \mathcal{D})$ 。例如, 如果我们有两个模型, 这两个模型的后验概率相等。一个模型预测了 $t = a$ 附近的一个很窄的分布, 而另一个模型预测了 $t = b$ 附近的一个很窄的分布, 这样整体的预测分布是一个双峰的概率分布, 峰值位于 $t = a$ 和 $t = b$ 处, 而不是在 $t = \frac{a+b}{2}$ 处的一个单一的模型。

对于模型求平均的一个简单的近似是使用最可能的一个模型自己做预测。这被称为模型选择 (model selection)。

对于一个由参数 \mathbf{w} 控制的模型, 根据概率的加和规则和乘积规则, 模型证据为

$$p(\mathcal{D} | \mathcal{M}_i) = \int p(\mathcal{D} | \mathbf{w}, \mathcal{M}_i) p(\mathbf{w} | \mathcal{M}_i) d\mathbf{w} \quad (3.68)$$

从取样的角度来看, 边缘似然函数可以被看成从一个模型中生成数据集 \mathcal{D} 的概率, 这个模型的参数是从先验分布中随机取样的。还有一件有趣的事情是, 我们注意到模型证据恰好就是在估计参数的后验分布时出现在贝叶斯定理的分母中的归一化项, 因为

$$p(\mathbf{w} | \mathcal{D}, \mathcal{M}_i) = \frac{p(\mathcal{D} | \mathbf{w}, \mathcal{M}_i) p(\mathbf{w} | \mathcal{M}_i)}{p(\mathcal{D} | \mathcal{M}_i)} \quad (3.69)$$

通过对参数的积分进行一个简单的近似, 我们可以更加深刻地认识模型证据。首先考虑模型有一个参数 w 的情形。这个参数的后验概率正比于 $p(\mathcal{D} | w)p(w)$, 其中为了简化记号, 我们省略了它对于模型 \mathcal{M}_i 的依赖。如果我们假设后验分布在最大似然值 w_{MAP} 附近是一个尖峰, 宽度为 $\Delta w_{\text{后验}}$, 那么我们可以用被积函数的值乘以尖峰的宽度来近似这个积分。如果我们进一步假设先验分布是平的, 宽度为 $\Delta w_{\text{先验}}$, 即 $p(w) = \frac{1}{\Delta w_{\text{先验}}}$, 那么我们有

$$p(\mathcal{D}) = \int p(\mathcal{D} | w)p(w) dw \simeq p(\mathcal{D} | w_{MAP}) \frac{\Delta w_{\text{后验}}}{\Delta w_{\text{先验}}} \quad (3.70)$$

取对数可得

$$\ln p(\mathcal{D}) \simeq \ln p(\mathcal{D} | w_{MAP}) + \ln \left(\frac{\Delta w_{\text{后验}}}{\Delta w_{\text{先验}}} \right) \quad (3.71)$$

图 3.12 说明了这个近似。第一项表示拟合由最可能参数给出的数据。对于平的先验分布来说, 这对应于对数似然。第二项用于根据模型的复杂度来惩罚模型。由于 $\Delta w_{\text{后验}} < \Delta w_{\text{先验}}$, 因此这一项为负, 并且随着 $\frac{\Delta w_{\text{后验}}}{\Delta w_{\text{先验}}}$ 的减小, 它的绝对值会增加。因此, 如果参数精确地调整为后验分布的数据, 那么惩罚项会很大。

对于一个有 M 个参数的模型, 我们可以对每个参数进行类似的近似。假设所有的参数的 $\frac{\Delta w_{\text{后验}}}{\Delta w_{\text{先验}}}$ 都相同, 我们有

$$\ln p(\mathcal{D}) \simeq \ln p(\mathcal{D} | \mathbf{w}_{MAP}) + M \ln \left(\frac{\Delta w_{\text{后验}}}{\Delta w_{\text{先验}}} \right) \quad (3.72)$$

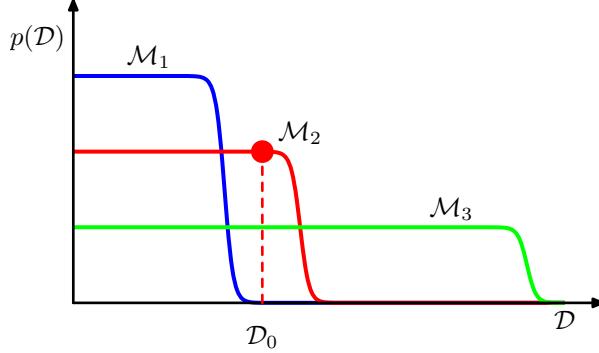


图 3.13: 对于三个具有不同复杂度的模型，数据集的概率分布的图形表示，其中 \mathcal{M}_1 是最简单的， \mathcal{M}_3 是最复杂的。注意，概率分布是归一化的。在这个例子中，对于特定的观测数据集 \mathcal{D}_0 ，具有中间复杂度的模型 \mathcal{M}_2 具有最大的模型证据。

因此，在这种非常简单的近似下，复杂度惩罚项的大小随着模型中可调节参数 M 的数量线性增加。随着我们增加模型的复杂度，第一项通常会增大，因为一个更加复杂的模型能够更好地拟合数据，而第二项会减小，因为它依赖于 M 。由最大模型证据确定的最优的模型复杂度需要在这两个相互竞争的项之间进行折中。我们后面会介绍这种近似的一个更加精炼的版本，那个版本依赖于后验概率分布的高斯近似。

通过图3.13，我们可以进一步深入认识贝叶斯模型比较，并且理解边缘似然是如何倾向于选择中等复杂度的模型的。这里，横轴是可能的数据集构成的空间的一个一维表示，因此轴上的每个点都对应着一个具体的数据集。我们现在考虑三个模型 \mathcal{M}_1 、 \mathcal{M}_2 和 \mathcal{M}_3 ，复杂度依次增加。假设我们让这三个模型自动产生样本数据集，然后观察生成的数据集的分布。任意给定的模型都能够生成一系列不同的数据集，这是因为模型的参数由先验概率分布控制，对于任意一种参数的选择，在目标变量上都可能有随机的噪声。为了从具体的模型中生成一个特定的数据集，我们首先从先验分布 $p(w)$ 中选择参数的值，然后对于这些参数的值，我们按照概率 $p(\mathcal{D} | w)$ 对数据进行采样。一个简单的模型（例如，基于一阶多项式的模型）几乎没有变化性，因此生成的数据集彼此之间都十分相似。于是它的分布 $p(\mathcal{D})$ 就被限制在横轴的一个相对小的区域。相反，一个复杂的模型（例如九阶多项式）可以生成变化性相当大的数据集，因此它的分布 $p(\mathcal{D})$ 遍布了数据集空间的一个相当大的区域。由于概率分布 $p(\mathcal{D} | \mathcal{M}_i)$ 是归一化的，因此我们看到特定的数据集 \mathcal{D}_0 对中等复杂度的模型有最高的模型证据。本质上说，简单的模型不能很好地拟合数据，而复杂的模型把它的预测概率散布于过多的可能的数据集当中，从而对它们当中的每一个赋予的概率都相对较小。

贝叶斯模型比较框架中隐含的一个假设是，生成数据的真实的概率分布包含在考虑的模型集合当中。如果这个假设确实成立，那么我们可以证明，平均来看，贝叶斯模型比较会倾向于选择出正确的模型。为了证明这一点，考虑两个模型 \mathcal{M}_1 和 \mathcal{M}_2 ，其中真实概率分布对应于模型 \mathcal{M}_1 。对于给定的有限数据集，确实有可能出现错误的模型反而使贝叶斯因子较大的事情。但是，如果我们把贝叶斯因子在数据集分布上进行平均，那么我们可以得到期望贝叶斯因子

$$\int p(\mathcal{D} | \mathcal{M}_1) \ln \frac{p(\mathcal{D} | \mathcal{M}_1)}{p(\mathcal{D} | \mathcal{M}_2)} d\mathcal{D} \quad (3.73)$$

上式是关于数据的真实分布求的平均值。这是Kullback-Leibler散度的一个例子，满足下面的性质：如果两个分布相等，则Kullback-Leibler散度等于零，否则恒为正。因此平均来讲，贝叶斯因子总会倾向于选择正确的模型。

我们已经看到，贝叶斯框架避免了过拟合的问题，并且使得模型能够基于训练数据自身进行对比。但是，与模式识别中任何其他的方法一样，贝叶斯方法需要对模型的形式作出假设，并且如果这些假设不合理，那么结果就会出错。特别地，我们从图3.12可以看出，模型证据对先验分布的很多方面都很敏感，例如在低概率处的行为等等。实际上，如果先验分布是反常的，那么模型证据无法定义，因为反常的先验分布有着任意的缩放因子（换句话说，归一化系数无法定义，因为分布根本无法被归一化）。如果我们考虑一个正常的先验分布，然后取一个适当的极限来获得一个反常的先验（例如高斯先验中，我们令方差为无穷大），那么模型证据就会趋

于零，这可以从公式 (3.70) 和图3.12中看出来。但是这种情况下也可能通过首先考虑两个模型的证据比值，然后取极限的方式来得到一个有意义的答案。

因此，在实际应用中，一种明智的做法是，保留一个独立的测试数据集，这个数据集用来评估最终系统的整体表现。

3.5 证据近似

在处理线性基函数模型的纯粹的贝叶斯方法中，我们会引入超参数 α 和 β 的先验分布，然后通过对超参数以及参数 w 求积分的方式做预测。但是，虽然我们可以解析地求出对 w 的积分或者求出对超参数的积分，但是对所有这些变量完整地求积分是没有解析解的。这里我们讨论一种近似方法。这种方法中，我们首先对参数 w 求积分，得到边缘似然函数 (marginal likelihood function)，然后通过最大化边缘似然函数，确定超参数的值。这个框架在统计学的文献中被称为经验贝叶斯 (empirical Bayes) (Bernardo and Smith, 1994; Gelman et al., 2004)，或者被称为第二类最大似然 (type 2 maximum likelihood) (Berger, 1985)，或者被称为推广的最大似然 (generalized maximum likelihood)。在机器学习的文献中，这种方法也被称为证据近似 (evidence approximation) (Gull, 1989; MacKay, 1992a)。

如果我们引入 α 和 β 上的超先验分布，那么预测分布可以通过对 w, α 和 β 求积分的方法得到，即

$$p(t | \mathbf{t}) = \iiint p(t | w, \beta) p(w | \mathbf{t}, \alpha, \beta) p(\alpha, \beta | \mathbf{t}) dw d\alpha d\beta \quad (3.74)$$

其中 $p(t | w, \beta)$ 由公式 (3.8) 给出， $p(w | \mathbf{t}, \alpha, \beta)$ 由公式 (3.49)，其中 m_N 和 S_N 分别由公式 (3.53) 和公式 (3.54) 定义。这里，为了让记号简洁，我们省略了对于输入变量 x 的依赖关系。如果后验分布 $p(\alpha, \beta | \mathbf{t})$ 在 $\hat{\alpha}$ 和 $\hat{\beta}$ 附近有尖峰，那么预测分布可以通过对 w 积分的方式简单地得到，其中 α 和 β 被固定为 $\hat{\alpha}$ 和 $\hat{\beta}$ ，即

$$p(t | \mathbf{t}) \simeq p(t | \mathbf{t}, \hat{\alpha}, \hat{\beta}) = \int p(t | w, \hat{\beta}) p(w | \mathbf{t}, \hat{\alpha}, \hat{\beta}) dw \quad (3.75)$$

根据贝叶斯定理， α 和 β 的后验分布为

$$p(\alpha, \beta | \mathbf{t}) \propto p(\mathbf{t} | \alpha, \beta) p(\alpha, \beta) \quad (3.76)$$

如果先验分布相对比较平，那么在证据框架中， $\hat{\alpha}$ 和 $\hat{\beta}$ 可以通过最大化边缘似然函数 $p(\mathbf{t} | \alpha, \beta)$ 来获得。我们接下来会计算线性基函数模型的边缘似然函数，然后找到它的最大值。这将使我们能够从训练数据本身确定这些超参数的值，而不需要交叉验证。回忆一下比值 $\frac{\alpha}{\beta}$ 类似于正则化参数。

此外，值得注意的一点是，如果我们定义 α 和 β 上的共轭 (Gamma) 先验分布，那么对公式 (3.74) 中的这些超参数求积分可以解析地计算出来，得到 w 上的学生t分布 (见第2.3.7节)。虽然得到的 w 上的积分不再有解析解，但是我们可以认为对这个积分求近似会给证据框架提供了另一种实用的方法 (Buntine and Weigend, 1991)。其中，可以使用拉普拉斯近似方法 (见第4.4节) 对这个积分求近似。拉普拉斯近似方法的基础是以后验概率分布的众数为中心的局部高斯近似方法。然而，作为 w 的函数的被积函数的众数通常很不准确，因此拉普拉斯近似方法不能描述概率质量中的大部分信息。这就导致最终的结果要比最大化证据的方法给出的结果更差 (MacKay, 1999)。

回到证据框架中，我们注意到有两种方法可以用来最大化对数证据。我们可以解析地计算证据函数，然后令它的导数等于零，得到了对于 α 和 β 的重新估计方程 (将在3.5.2节讨论)。另一种方法是，我们使用一种被称为期望最大化 (EM) 算法的方法，这个算法将在9.3.4节讨论，那里我们还会证明这两种方法会收敛到同一个解。

3.5.1 计算证据函数

边缘似然函数 $p(\mathbf{t} | \alpha, \beta)$ 是通过对权值参数 w 进行积分得到的，即

$$p(\mathbf{t} | \alpha, \beta) = \int p(\mathbf{t} | w, \beta) p(w | \alpha) dw \quad (3.77)$$

一种计算这个积分的方法是再次使用公式 (2.115) 给出的线性-高斯模型的条件概率分布的结果。这里，我们使用另一种方法计算这个积分，即通过对指数项配平方，然后使用高斯分布的归一化系数的基本形式。

根据公式 (3.11)、公式 (3.12) 和公式 (3.52)，我们可以把证据函数写成下面的形式

$$p(\mathbf{t} | \alpha, \beta) = \left(\frac{\beta}{2\pi} \right)^{\frac{N}{2}} \left(\frac{\alpha}{2\pi} \right)^{\frac{M}{2}} \int \exp\{-E(\mathbf{w})\} d\mathbf{w} \quad (3.78)$$

其中 M 是 \mathbf{w} 的维数，并且，我们定义了

$$\begin{aligned} E(\mathbf{w}) &= \beta E_D(\mathbf{w}) + \alpha E_W(\mathbf{w}) \\ &= \frac{\beta}{2} \|\mathbf{t} - \Phi \mathbf{w}\|^2 + \frac{\alpha}{2} \mathbf{w}^T \mathbf{w} \end{aligned} \quad (3.79)$$

我们看到，如果忽略一些比例常数，公式 (3.79) 等于正则化的平方和误差函数 (3.27)。我们现在对 \mathbf{w} 配平方，可得

$$E(\mathbf{w}) = E(\mathbf{m}_N) + \frac{1}{2} (\mathbf{w} - \mathbf{m}_N)^T \mathbf{A} (\mathbf{w} - \mathbf{m}_N) \quad (3.80)$$

其中我们令

$$\mathbf{A} = \alpha \mathbf{I} + \beta \Phi^T \Phi \quad (3.81)$$

以及

$$E(\mathbf{m}_N) = \frac{\beta}{2} \|\mathbf{t} - \Phi \mathbf{m}_N\|^2 + \frac{\beta}{2} \mathbf{m}_N^T \mathbf{m}_N \quad (3.82)$$

注意 \mathbf{A} 对应于误差函数的二阶导数

$$\mathbf{A} = \nabla \nabla E(\mathbf{w}) \quad (3.83)$$

被称为Hessian矩阵。这里我们也定义了 \mathbf{m}_N 为

$$\mathbf{m}_N = \beta \mathbf{A}^{-1} \Phi^T \mathbf{t} \quad (3.84)$$

使用公式 (3.54)，我们看到 $\mathbf{A} = \mathbf{S}_N^{-1}$ ，因此公式 (3.84) 等价于之前的定义 (3.53)，从而它表示后验概率分布的均值。

通过比较多元高斯分布的归一化系数，关于 \mathbf{w} 的积分现在可以很容易地计算出来了，即

$$\begin{aligned} &\int \exp\{-E(\mathbf{w})\} d\mathbf{w} \\ &= \exp\{-E(\mathbf{m}_N)\} \int \exp\left\{-\frac{1}{2} (\mathbf{w} - \mathbf{m}_N)^T \mathbf{A} (\mathbf{w} - \mathbf{m}_N)\right\} d\mathbf{w} \\ &= \exp\{-E(\mathbf{m}_N)\} (2\pi)^{\frac{M}{2}} |\mathbf{A}|^{-\frac{1}{2}} \end{aligned} \quad (3.85)$$

使用公式 (3.78)，我们可以把边缘似然函数的对数写成下面的形式

$$\ln p(\mathbf{t} | \alpha, \beta) = \frac{M}{2} \ln \alpha + \frac{N}{2} \ln \beta - E(\mathbf{m}_N) - \frac{1}{2} \ln |\mathbf{A}| - \frac{N}{2} \ln(2\pi) \quad (3.86)$$

这就是证据函数的表达式。

回到多项式回归问题，我们可以画出模型证据与多项式阶数之间的关系，如图3.14所示。这里，我们已经假定先验分布的形式为公式 (1.65)，参数 α 的值固定为 $\alpha = 5 \times 10^{-3}$ 。这个图像的形式非常有指导意义。我们回头看图1.4，我们看到 $M = 0$ 的多项式对数据的拟合效果非常差，结果模型证据的值也相对较小。 $M = 1$ 的多项式对于数据的拟合效果有了显著的提升，因此模型证据变大了。但是，对于 $M = 2$ 的多项式，拟合效果又变得很差，因为产生数据的正弦函数是奇函数，因此在多项式展开中没有偶次项。事实上，图1.5给出的数据残差从 $M = 1$ 到 $M = 2$ 只有微小的减小。由于复杂的模型有着更大的复杂度惩罚项，因此

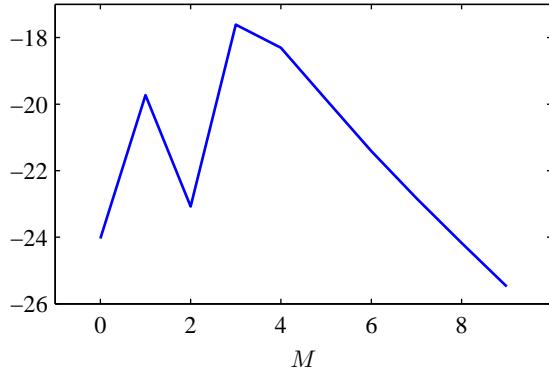


图 3.14: 多项式回归模型的模型对数证据与阶数 M 的关系图像, 表明证据倾向于选择 $M = 3$ 的模型。

从 $M = 1$ 到 $M = 2$, 模型证据实际上减小了。当 $M = 3$ 时, 我们对于数据的拟合效果有了很大的提升, 如图 1.4 所示, 因此模型证据再次增大, 给出了多项式拟合的最高的模型证据。进一步增加 M 的值, 只能少量地提升拟合的效果, 但是模型的复杂度却越来越复杂, 这导致整体的模型证据会下降。再次看图 1.5, 我们看到泛化错误在 $M = 3$ 到 $M = 8$ 之间几乎为常数, 因此单独基于这幅图很难对模型做出选择。然而, 模型证据的值明显地倾向于选择 $M = 3$ 的模型, 因为这是能很好地解释观测数据的最简单的模型。

3.5.2 最大化证据函数

让我们首先考虑 $p(\mathbf{t} | \alpha, \beta)$ 关于 α 的最大化。首先定义下面的特征向量方程

$$(\beta \Phi^T \Phi) \mathbf{u}_i = \lambda_i \mathbf{u}_i \quad (3.87)$$

根据公式 (3.81), 可知 \mathbf{A} 的特征值为 $\alpha + \lambda_i$ 。现在考虑公式 (3.86) 中涉及到 $\ln |\mathbf{A}|$ 的项关于 α 的导数

$$\frac{d}{d\alpha} \ln |\mathbf{A}| = \frac{d}{d\alpha} \ln \prod_i (\lambda_i + \alpha) = \frac{d}{d\alpha} \sum_i \ln(\lambda_i + \alpha) = \sum_i \frac{1}{\lambda_i + \alpha} \quad (3.88)$$

因此函数 (3.86) 关于 α 的驻点满足

$$0 = \frac{M}{2\alpha} - \frac{1}{2} \mathbf{m}_N^T \mathbf{m}_N - \frac{1}{2} \sum_i \frac{1}{\lambda_i + \alpha} \quad (3.89)$$

两侧乘以 2α , 整理, 可得

$$\alpha \mathbf{m}_N^T \mathbf{m}_N = M - \alpha \sum_i \frac{1}{\lambda_i + \alpha} = \gamma \quad (3.90)$$

由于 i 的求和式中一共有 M 项, 因此 γ 可以写成

$$\gamma = \sum_i \frac{\lambda_i}{\alpha + \lambda_i} \quad (3.91)$$

γ 的意义稍后会讨论。根据方程 (3.90), 我们看到最大化边缘似然函数的 α 满足

$$\alpha = \frac{\gamma}{\mathbf{m}_N^T \mathbf{m}_N} \quad (3.92)$$

注意, 这是 α 的一个隐式解, 不仅因为 γ 与 α 相关, 还因为后验概率本身的众数 \mathbf{m}_N 也与 α 的选择有关。因此我们使用迭代的方法求解。首先我们选择一个 α 的初始值, 使用这个初始值找到 \mathbf{m}_N (由公式 (3.53) 求得), 利用公式 (3.91) 计算 γ 。之后这些值被公式 (3.92) 用来重新

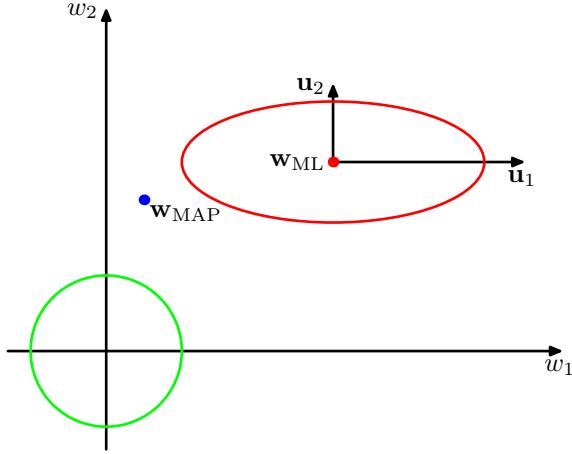


图 3.15: 似然函数的轮廓线（红色）和先验概率分布（绿色），其中参数空间中的坐标轴被旋转，与Hessian矩阵的特征向量 \mathbf{u}_i 对齐。对于 $\alpha = 0$ ，后验概率分布的众数由最大似然解 \mathbf{w}_{ML} 给出，而对于非零的 α ，众数位于 $\mathbf{w}_{MAP} = \mathbf{m}_N$ 的位置。在方向 w_1 上，由公式（3.87）定义的特征值 λ_1 与 α 相比较小，因此 $\lambda_1/(\lambda_1 + \alpha)$ 接近零，对应的 w_1 的MAP值也接近零。相反，在 w_2 的方向上，特征值 λ_2 与 α 相比较大，因此 $\lambda_2/(\lambda_2 + \alpha)$ 接近1， w_2 的MAP值接近于最大似然值。

估计 α 。这个过程不断进行，直到收敛。注意，由于矩阵 $\Phi^T \Phi$ 是固定的，因此我们可以在最开始的时候计算一次特征值，然后接下来只需乘以 β 就可以得到 λ_i 的值。

应该强调的是， α 的值是纯粹通过观察训练集确定的。与最大似然方法不同，最优化模型复杂度不需要独立的数据集。

我们可以类似地关于 β 最大化对数边缘似然函数（3.86）。为了完成这一点，我们注意到公式（3.87）定义的特征值 λ_i 正比于 β ，因此 $\frac{d}{d\beta} = \frac{\lambda_i}{\beta}$ 。于是

$$\frac{d}{d\beta} \ln |\mathbf{A}| = \frac{d}{d\beta} \sum_i \ln(\lambda_i + \alpha) = \frac{1}{\beta} \sum_i \frac{\lambda_i}{\lambda_i + \alpha} = \frac{\gamma}{\beta} \quad (3.93)$$

边缘似然函数的驻点因此满足

$$0 = \frac{N}{2\beta} - \frac{1}{2} \sum_{n=1}^N \{t_n - \mathbf{m}_N^T \phi(\mathbf{x}_n)\}^2 - \frac{\gamma}{2\beta} \quad (3.94)$$

整理，我们可以得到

$$\frac{1}{\beta} = \frac{1}{N - \gamma} \sum_{n=1}^N \{t_n - \mathbf{m}_N^T \phi(\mathbf{x}_n)\}^2 \quad (3.95)$$

与之前一样，这是 β 的一个隐式解，可以通过迭代的方法解出。首先选择 β 的一个初始值，然后使用这个初始值计算 \mathbf{m}_N 和 γ ，然后使用公式（3.95）重新估计 β 的值，重复直到收敛。如果 α 和 β 的值都要从数据中确定，那么他们的值可以在每次更新 γ 之后一起重新估计。

3.5.3 参数的有效数量

公式（3.92）给出的结果有一个十分优雅的意义（MacKay, 1992a），它提供给我们关于 α 的贝叶斯解的更深刻的认识。考虑似然函数的轮廓线以及先验概率分布，如图3.15所示。这里，我们隐式地把参数空间的坐标轴进行了旋转变换，使其与公式（3.87）定义的特征向量对齐。这样，似然函数的轮廓线就变成了轴对齐的椭圆。特征值 λ_i 度量了似然函数的曲率，因此在图3.15中，特征值 λ_1 小于 λ_2 （因为较小的曲率对应着似然函数轮廓线较大的拉伸）。由于 $\beta \Phi^T \Phi$ 是一个正定矩阵，因此它的特征值为正数，从而比值 $\frac{\lambda_i}{\lambda_i + \alpha}$ 位于0和1之间。结果，由公式（3.91）定义的 γ 的取值范围为 $0 \leq \gamma \leq M$ 。对于 $\lambda_i \gg \alpha$ 的方向，对应的参数 w_i 将会与最大似然值接近，且比值 $\frac{\lambda_i}{\lambda_i + \alpha}$ 接近1。这样的参数被称为良好确定的（well determined），因为它们的值被数据紧紧

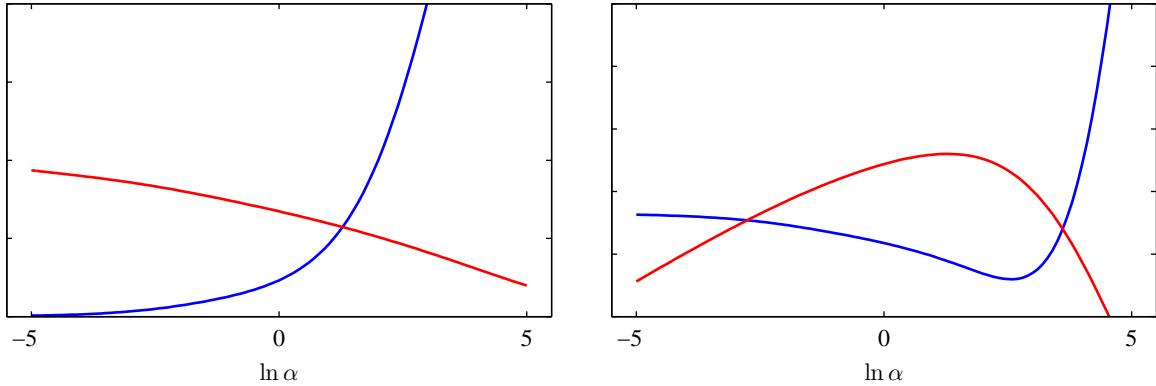


图 3.16: 左图给出了 γ 与 $\ln \alpha$ 的关系（红色曲线）以及 $2\alpha E_W(\mathbf{m}_N)$ 与 $\ln \alpha$ 的关系（蓝色曲线），数据集为正弦数据集。这两条曲线的交点定义了 α 的最优解，由模型证据的步骤给出。右图给出了对应的对数证据 $\ln p(\mathbf{t} | \alpha, \beta)$ 关于 $\ln \alpha$ 的图像（红色曲线），说明了峰值与左图中曲线的交点恰好重合。同样给出的时测试集误差（蓝色曲线），说明模型证据最大值的位置接近于具有最好泛化能力的点。

地限制着。相反，对于 $\lambda_i \ll \alpha$ 的方向，对应的参数 w_i 将会接近0，比值 $\frac{\lambda_i}{\lambda_i + \alpha}$ 也会接近0。这些方向上，似然函数对于参数的值相对不敏感，因此参数被先验概率设置为较小的值。公式 (3.91) 定义的 γ 因此度量了良好确定的参数的有效总数。

我们可以更深刻地研究一下用于重新估计 β 的公式 (3.95)。让我们把 β 和公式 (3.21) 给出的最大似然结果进行比较。这两个公式都把方差（精度的倒数）表示为目标值和模型预测值的差的平方的平均值。但是，它们的区别在于，最大似然结果的分母是数据点的数量 N ，而贝叶斯结果的分母是 $N - \gamma$ 。根据公式 (1.56)，我们看到单一变量 x 的高斯分布的方差的最大似然估计为

$$\sigma_{ML}^2 = \frac{1}{N} \sum_{n=1}^N (x_n - \mu_{ML})^2 \quad (3.96)$$

这个估计是有偏的，因为均值的最大似然解 μ_{ML} 拟合了数据中的一些噪声。从效果上来看，这占用了模型的一个自由度。对应的无偏的估计由公式 (1.59) 给出，形式为

$$\sigma_{MAP}^2 = \frac{1}{N-1} \sum_{n=1}^N (x_n - \mu_{ML})^2 \quad (3.97)$$

分母中的因子 $N - 1$ 反映了模型中的一个自由度被用于拟合均值的事实，它抵消了最大似然解的偏差。现在考虑线性回归模型的对应的结果。目标分布的均值现在由函数 $\mathbf{w}^T \phi(\mathbf{x})$ 给出，它包含了 M 个参数。但是，并不是所有的这些参数都按照数据进行了调整。由数据确定的有效参数的数量为 γ ，剩余的 $M - \gamma$ 个参数被先验概率分布设置为较小的值。这可以通过方差的贝叶斯结果中的因子 $N - \gamma$ 反映出来，因此修正了最大似然结果的偏差。

我们可以说明使用1.1节的正弦数据超参数的有效框架，以及由9个基函数组成的高斯基函数模型，因此模型中的参数的总数为 $M = 10$ ，这里包含了偏置。这里为了说明的简洁性，我们已经把 β 设置成了真实值11.1，然后使用证据框架来确定 α ，如图3.16所示。

我们也可以看到参数 α 是如何控制参数 $\{w_i\}$ 的大小的。图3.17给出了独立的参数关于有效参数数量 γ 的函数图像。

如果我们考虑极限情况 $N \gg M$ ，数据点的数量大于参数的数量，那么根据公式 (3.87)，所有的参数都可以根据数据良好确定。因为 $\Phi^T \Phi$ 涉及到数据点的隐式求和，因此特征值 λ_i 随着数据集规模的增加而增大。在这种情况下， $\gamma = M$ ，并且 α 和 β 的重新估计方程变为了

$$\alpha = \frac{M}{2E_W(\mathbf{m}_N)} \quad (3.98)$$

$$\beta = \frac{N}{2E_D(\mathbf{m}_N)} \quad (3.99)$$

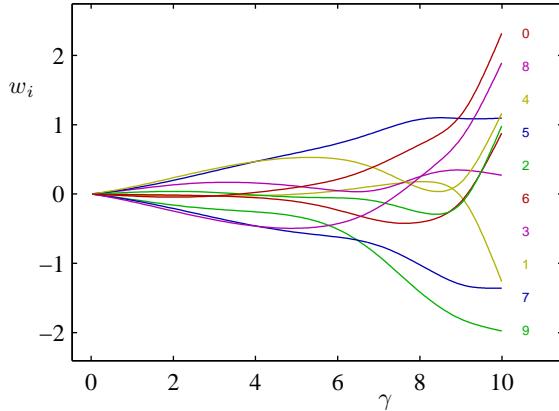


图 3.17: 高斯基函数模型中的 10 个参数 w_i 与参数有效数量 γ 的关系, 其中超参数的变化范围为 $0 \leq \alpha \leq \infty$, 使得 γ 的变化范围为 $0 \leq \gamma \leq M$ 。

其中 E_W 和 E_D 分别由公式 (3.25) 和公式 (3.26) 定义。这些结果可以用作完整的重新估计公式的简化计算的近似, 因为它们不需要计算 Hessian 矩阵的一系列特征值。

3.6 固定基函数的局限性

在本章中, 我们已经关注了由固定的非线性基函数的线性组合组成的模型。我们已经看到, 对于参数的线性性质的假设产生了一系列有用的性质, 包括最小平方问题的解析解, 以及容易计算的贝叶斯方法。此外, 对于一个合适的基函数的选择, 我们可以建立输入向量到目标值之间的任意非线性映射。在下一章中, 我们会研究类似的用于分类的模型。

因此, 似乎这样的模型建立的解决模式识别问题的通用框架。不幸的是, 线性模型有一些重要的局限性, 这使得我们在后续的章节中要转而关注更加复杂的模型, 例如支持向量机和神经网络。

困难的产生主要是因为我们假设了基函数在观测到任何数据之前就被固定了下来, 而这正是 1.4 节讨论的维度灾难问题的一个表现形式。结果, 基函数的数量随着输入空间的维度 D 迅速增长, 通常是指数方式的增长。

幸运的是, 真实数据集有两个性质, 可以帮助我们缓解这个问题。第一, 数据向量 $\{\mathbf{x}_n\}$ 通常位于一个非线性流形内部。由于输入变量之间的相关性, 这个流形本身的维度小于输入空间的维度。我们将在第 12 章中讨论手写数字识别时给出一个例子来说明这一点。如果我们使用局部基函数, 那么我们可以让基函数只分布在输入空间中包含数据的区域。这种方法被用在径向基函数网络中, 也被用在支持向量机和相关向量机当中。神经网络模型使用可调节的基函数, 这些基函数有着 sigmoid 非线性的性质。神经网络可以通过调节参数, 使得在输入空间的区域中基函数会按照数据流形发生变化。第二, 目标变量可能只依赖于数据流形中的少量可能的方向。利用这个性质, 神经网络可以通过选择输入空间中基函数产生响应的方向。

3.7 练习

(3.1) (*) 证明, 双曲正切函数与公式 (3.6) 定义的 logistic sigmoid 函数的关系为

$$\tanh(a) = 2\sigma(2a) - 1 \quad (3.100)$$

这也就能够证明, logistic sigmoid 函数的一个一般的线性组合

$$y(x, \mathbf{w}) = w_0 + \sum_{j=1}^M w_j \sigma\left(\frac{x - \mu_j}{s}\right) \quad (3.101)$$

等价于一个双曲正切函数的线性组合

$$y(x, \mathbf{u}) = u_0 + \sum_{j=1}^M u_j \tanh\left(\frac{x - \mu_j}{2s}\right) \quad (3.102)$$

寻找一个表达式，将新的参数 $\{u_0, \dots, u_M\}$ 与原始的参数 $\{w_0, \dots, w_M\}$ 关联起来。

(3.2) (***) 证明矩阵

$$\Phi(\Phi^T \Phi)^{-1} \Phi^T \quad (3.103)$$

会把任意的向量 \mathbf{v} 投影到由 Φ 的列张成的空间上。使用这个结果证明最小平方解 (3.15) 对应于向量 \mathbf{t} 在流形 S 上的一个正交投影，如图3.2所示。

(3.3) (*) 考虑一个数据集，其中每个数据点 t_n 都与一个权因子 $r_n > 0$ 相关联，从而平方和误差函数变为了

$$E_D(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^N r_n \{t_n - \mathbf{w}^T \phi(\mathbf{x}_n)\}^2 \quad (3.104)$$

找到最小化这个误差函数的解 \mathbf{w}^* 的表达式。说出这种加权的平方和误差函数的两个意义，分别根据 (1) 数据对噪声方差的依赖性 (2) 复制的数据点。

(3.4) (*) 考虑一个线性模型

$$y(\mathbf{x}, \mathbf{w}) = w_0 + \sum_{i=1}^D w_i x_i \quad (3.105)$$

以及平方和误差函数

$$E_D(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^N \{y(\mathbf{x}_n, \mathbf{w}) - t_n\}^2 \quad (3.106)$$

现在假设服从均值为零方差为 σ^2 的高斯分布的噪声 ϵ_i 被独立地加到每个输入变量 x_i 上。通过使用 $\mathbb{E}[\epsilon_i] = 0$ 和 $\mathbb{E}[\epsilon_i \epsilon_j] = \delta_{ij} \sigma^2$ ，证明，对在噪声分布上做平均的 E_D 进行最小化，等价于对附加权值衰减的正则化项的无噪声输入变量的平方和误差函数进行最小化，其中偏置参数 w_0 从正则化项中被省略掉。

(3.5) (*) 使用附录E中讨论的拉格朗日乘数法，证明最小化正则化的误差函数 (3.29) 等价于在限制条件 (3.30) 下最小化未正则化的平方和误差函数 (3.12)。讨论参数 η 和 λ 的关系。

(3.6) (*) 考虑多元目标变量 \mathbf{t} 的线性基函数回归模型，其中 \mathbf{t} 服从高斯分布，形式为

$$p(\mathbf{t} | \mathbf{W}, \Sigma) = \mathcal{N}(\mathbf{t} | \mathbf{y}(\mathbf{x}, \mathbf{W}), \Sigma) \quad (3.107)$$

其中

$$\mathbf{y}(\mathbf{x}, \mathbf{W}) = \mathbf{W}^T \phi(\mathbf{x}) \quad (3.108)$$

训练数据集由基向量输入 $\phi(\mathbf{x}_n)$ 和对应的目标向量 t_n 组成，其中 $n = 1, \dots, N$ 。证明参数矩阵 \mathbf{W} 的最大似然解 \mathbf{W}_{ML} 具有这样的性质：每一列由形如 (3.15) 的表达式给出，它是各向同性的噪声分布的解。注意，这个最大似然解与协方差矩阵 Σ 无关。证明， Σ 的最大似然解为

$$\Sigma = \frac{1}{N} \sum_{n=1}^N (t_n - \mathbf{W}_{ML}^T \phi(\mathbf{x}_n)) (t_n - \mathbf{W}_{ML}^T \phi(\mathbf{x}_n))^T \quad (3.109)$$

(3.7) (*) 通过使用配平方的方法，证明公式 (3.49) 给出的线性基函数模型中的参数 \mathbf{w} 的后验概率分布的结果，其中 \mathbf{m}_N 和 \mathbf{S}_N 分别由公式 (3.50) 和公式 (3.51) 定义。

(3.8) (***) 考虑3.1节的线性基函数模型。假设我们已经观测到了 N 个数据点，从而 \mathbf{w} 的后验概率分布由公式 (3.49) 给出。这个后验概率可以被当成下一次观测的先验概率。通过考虑一个额外的数据点 $(\mathbf{x}_{N+1}, t_{N+1})$ ，使用为指数项配平方的方法，证明最终的后验概率分布仍然由公式 (3.49) 给出，但是 \mathbf{S}_N 被替换为了 \mathbf{S}_{N+1} ， \mathbf{m}_N 被替换为了 \mathbf{S}_{N+1} 。

(3.9) (***) 重复上一个练习，但这次不是用手配平方，而是使用公式 (2.116) 给出的线性高斯模型的一般结果。

(3.10) (**) 使用公式 (2.115) 给出的结果，计算公式 (3.57) 的积分，证明贝叶斯线性回归模型的预测分布由公式 (3.58) 给出，其中与输入相关的变量由公式 (3.59) 给出。

(3.11) (**) 我们已经看到，随着数据集规模的增加，模型参数的后验概率分布的不确定性会降低。使用矩阵恒等式（附录C）

$$(\mathbf{M} + \mathbf{v}\mathbf{v}^T)^{-1} = \mathbf{M}^{-1} - \frac{(\mathbf{M}^{-1}\mathbf{v})(\mathbf{v}^T\mathbf{M}^{-1})}{1 + \mathbf{v}^T\mathbf{M}^{-1}\mathbf{v}} \quad (3.110)$$

证明公式 (3.59) 给出的线性回归函数的不确定性 $\sigma_N^2(\mathbf{x})$ 满足

$$\sigma_{N+1}^2(\mathbf{x}) \leq \sigma_N^2(\mathbf{x}) \quad (3.111)$$

(3.12) (**) 我们在2.3.6节看到，具有未知均值和未知精度（方差倒数）的高斯分布的共轭先验是正态-Gamma分布。这个性质对于线性回归模型的条件高斯分布 $p(t | \mathbf{x}, \mathbf{w}, \beta)$ 也成立。如果我们考虑似然函数 (3.10)，那么 \mathbf{w} 和 β 的共轭先验为

$$p(\mathbf{w}, \beta) = \mathcal{N}(\mathbf{w} | \mathbf{m}_0, \beta^{-1} \mathbf{S}_0) \text{Gam}(\beta | a_0, b_0) \quad (3.112)$$

证明对应的后验概率分布具有相同的函数形式，即

$$p(\mathbf{w}, \beta | \mathbf{t}) = \mathcal{N}(\mathbf{w} | \mathbf{m}_N, \beta^{-1} \mathbf{S}_N) \text{Gam}(\beta | a_N, b_N) \quad (3.113)$$

并且找出后验概率参数 $\mathbf{m}_N, \mathbf{S}_N, a_N$ 和 b_N 的表达式。

(3.13) (**) 证明练习3.12中讨论的模型的预测分布 $p(t | \mathbf{x}, \mathbf{t})$ 是学生t分布，形式为

$$p(t | \mathbf{x}, \mathbf{t}) = \text{St}(t | \mu, \lambda, \nu) \quad (3.114)$$

并求出 μ, λ 和 ν 的表达式。

(3.14) (**) 本练习中，我们仔细研究公式 (3.62) 定义的等价核的性质，其中 \mathbf{S}_N 由公式 (3.54) 定义。假设基函数 $\phi_j(\mathbf{x})$ 是线性独立的，且观测数据点的数量 N 大于基函数的数量 M 。此外，令某一个基函数为常数，例如 $\phi_0(\mathbf{x}) = 1$ 。通过对这些基函数进行恰当的线性变换，我们可以建立一个新的基的集合 $\psi_j(\mathbf{x})$ 。这个新的基的集合能够张成同样的空间，但是基是单位正交的，即

$$\sum_{n=1}^N \psi_j(\mathbf{x}_n) \psi_k(\mathbf{x}_n) = I_{jk} \quad (3.115)$$

其中，如果 $j = k$ ，则 I_{jk} 为 1，否则为 0。并且，我们取 $\psi_0(\mathbf{x}) = 1$ 。证明对于 $\alpha = 0$ ，等价核可以写成 $k(\mathbf{x}, \mathbf{x}') = \psi(\mathbf{x})^T \psi(\mathbf{x}')$ ，其中 $\psi = (\psi_0, \dots, \psi_{M-1})^T$ 。使用这个结果证明，核满足下面的加和限制

$$\sum_{n=1}^N k(\mathbf{x}, \mathbf{x}_n) = 1 \quad (3.116)$$

(3.15) (*) 考虑回归的线性基函数模型，其中参数 α 和 β 通过模型证据框架来设定。证明由公式 (3.82) 定义的函数 $E(\mathbf{m}_N)$ 满足关系 $2E(\mathbf{m}_N) = N$ 。

(3.16) (**) 使用公式 (2.115) 直接计算积分 (3.77)，推导线性回归模型的对数证据函数的结果 (3.86)。

(3.17) (*) 证明贝叶斯线性回归模型的证据函数可以写成公式 (3.78) 的形式，其中 $E(\mathbf{w})$ 由公式 (3.79) 定义。

(3.18) (**) 通过关于 \mathbf{w} 配平方，证明贝叶斯线性回归的误差函数 (3.79) 可以写成公式 (3.80) 的形式。

(3.19) (**) 证明贝叶斯线性回归模型中，对 \mathbf{w} 积分会得到结果 (3.85)。从而也就证明了对数边缘似然函数由公式 (3.86) 给出。

(3.20) (***) 证明，对于对数边缘似然函数 (3.86) 关于 α 进行最大化的步骤会产生出重估计方程 (3.92)。

(3.21) (***) 另一种推导模型证据框架中最优的 α 值的结果 (3.92) 的方法是使用恒等式

$$\frac{d}{d\alpha} \ln |\mathbf{A}| = \text{Tr} \left(\mathbf{A}^{-1} \frac{d}{d\alpha} \mathbf{A} \right) \quad (3.117)$$

通过考虑实对称矩阵 \mathbf{A} 的特征值展开式，然后使用由 \mathbf{A} 的特征值表示的行列式和迹的标准结果（附录C），证明这个恒等式。然后使用公式 (3.117)，从公式 (3.86) 开始，推导公式 (3.92)。

(3.22) (***) 证明，对于对数边缘似然函数 (3.86) 关于 β 进行最大化的步骤会产生出重估计方程 (3.95)。

(3.23) (***) 证明练习3.12描述的模型的数据的边缘概率分布（即模型证据）为

$$p(\mathbf{t}) = \frac{1}{(2\pi)^{\frac{N}{2}}} \frac{b_0^{a_0}}{b_N^{a_N}} \frac{\Gamma(a_N)}{\Gamma(a_0)} \frac{|\mathbf{S}_N|^{\frac{1}{2}}}{|\mathbf{S}_0|^{\frac{1}{2}}} \quad (3.118)$$

首先关于 \mathbf{w} 求积分，然后关于 β 求积分即可。

(3.24) (**) 重复上一个练习，但是这次使用贝叶斯定理

$$p(\mathbf{t}) = \frac{p(\mathbf{t} | \mathbf{w}, \beta)p(\mathbf{w}, \beta)}{p(\mathbf{w}, \beta | \mathbf{t})} \quad (3.119)$$

然后将先验概率分布、后验概率分布以及似然函数代入上面的表达式，推导出公式 (3.118) 的结果。

4 分类的线性模型

前一章中，我们研究了一类回归模型，这些模型有相当简单的数学性质和计算性质。我们现在讨论一类与此相似的模型，用于解决分类问题。分类的目标是将输入变量 \mathbf{x} 分到 K 个离散的类别 C_k 中的某一类。最常见的情况是，类别互相不想交，因此每个输入被分到唯一的一个类别中。因此输入空间被划分为不同的决策区域（decision region），它的边界被称为决策边界（decision boundary）或者决策面（decision surface）。在本章中，我们考虑分类的线性模型。所谓分类线性模型，是指决策面是输入向量 \mathbf{x} 的线性函数，因此被定义为 D 维输入空间中的 $(D - 1)$ 维超平面。如果数据集可以被线性决策面精确地分类，那么我们说这个数据集是线性可分的（linearly separable）。

对于回归问题来说，目标向量 \mathbf{t} 就是一个实数向量，它的值是我们想要预测的。在分类问题中，使用目标值的方式来表示类别标签有许多不同的方式。对于概率模型来说，在二分类问题的情况下，最方便的表达方式是二元表示方法。这种方法中，有一个目标变量 $t \in \{0, 1\}$ ，其中 $t = 1$ 表示类别 C_1 ，而 $t = 0$ 表示类别 C_2 。我们可以把 t 的值看成分类结果为 C_1 的概率，这个概率只取极端的值0和1。对于 $K > 2$ 类问题，比较方便的方法是使用“1-of- K ”编码规则。这种方法中， \mathbf{t} 是一个长度为 K 的向量。如果类别为 C_j ，那么 \mathbf{t} 的所有元素 t_k 中，只有 t_j 等于1，其余的都等于0。例如，如果我们有5个类别，那么来自第2个类别的模式给出的目标向量为

分类问题有三种方法：
1, 判别函数法。
2, 概率判别模型。
3, 概率生成模型。

$$\mathbf{t} = (0, 1, 0, 0, 0)^T \quad (4.1)$$

与之前一样，我们可以把 t_k 看成分类结果为 C_k 的概率。对于非概率模型，目标变量使用其他的表示方法有时候会更方便。

在第1章，我们提出了分类问题的三种不同方法。最简单的方法涉及到构造判别函数（discriminant function），它直接把向量 \mathbf{x} 分到具体的类别中。但是，一个更强大的方法是在推断阶段对条件概率分布 $p(C_k | \mathbf{x})$ 直接建模，然后使用这个概率分布进行最优决策。通过区分推断阶段和决策阶段，我们获得了很多有益的东西，正如1.5.4节讨论的那样。有两种不同的方法确定条件概率分布 $p(C_k | \mathbf{x})$ 。一种方法是直接对条件概率分布建模，例如把条件概率分布表示为参数模型，然后使用训练集来最优化参数。另一种方法是生成式的方法。这种方法中，我们对类条件概率密度 $p(\mathbf{x} | C_k)$ 以及类的先验概率分布 $p(C_k)$ 建模，然后我们使用贝叶斯定理计算后验概率分布

$$p(C_k | \mathbf{x}) = \frac{p(\mathbf{x} | C_k)p(C_k)}{p(\mathbf{x})} \quad \text{贝叶斯分类器} \quad (4.2)$$

我们将在本章中讨论这三种方法。

在第3章讨论的线性回归模型中，模型的预测 $y(\mathbf{x}, \mathbf{w})$ 由参数 \mathbf{w} 的线性函数给出。在最简单的情况下，模型对输入变量也是线性的，因此形式为 $y(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + w_0$ ，即 y 是一个实数。然而对于分类问题，我们想预测的是离散的类别标签，或者更一般地，预测位于区间(0, 1)的后验概率分布。为了完成这一点，我们考虑这个模型的一个推广，这个模型中我们使用非线性函数 $f(\cdot)$ 对 \mathbf{w} 的线性函数进行变换，即

$$y(\mathbf{x}) = f(\mathbf{w}^T \mathbf{x} + w_0) \quad (4.3)$$

在机器学习的文献中， $f(\cdot)$ 被称为激活函数（activation function），而它的反函数在统计学的文献中被称为链接函数（link function）。决策面对应于 $y(\mathbf{x}) = \text{常数}$ ，即 $\mathbf{w}^T \mathbf{x} + w_0 = \text{常数}$ ，因此决策面是 \mathbf{x} 的线性函数，即使函数 $f(\cdot)$ 是非线性函数也是如此。因此，由公式 (4.3) 描述的一类模型被称为推广的线性模型（generalized linear model）（McCullagh and Nelder, 1989）。但是，需要注意的是，与回归中使用的模型相反，它们不再是参数的线性模型，因为我们引入了非线性函数 $f(\cdot)$ 。这会导致计算比线性回归模型更加复杂。尽管这样，这些模型与后续章节中要讨论的更加一般的非线性模型相比，仍然相对简单。

本章中讨论的算法同样适用于下面的情形：我们对输入变量进行一个固定的非线性变换，这个变换使用一个基函数向量 $\phi(\mathbf{x})$ ，正如我们在第3章中对回归模型做的那样。本章的开始，我们考虑直接对原始输入空间 \mathbf{x} 分类的问题，而第4.3章中，我们会发现，为了与后续章节相容，我们引入基函数会比较方便。

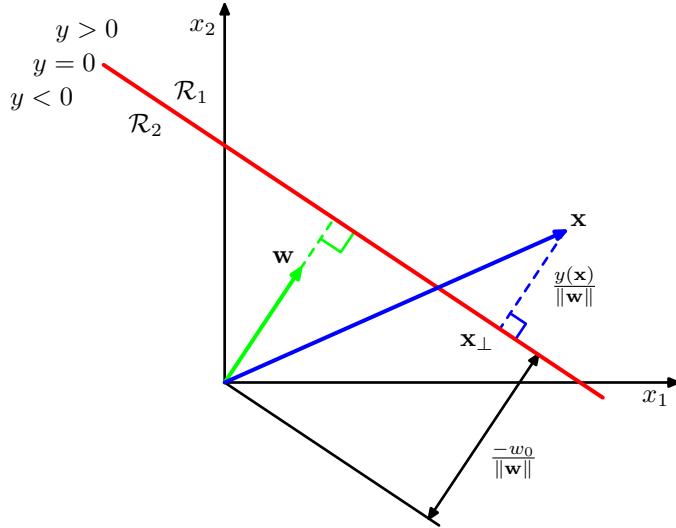


图 4.1: 二维线性判别函数的几何表示。决策面（红色）垂直于 \mathbf{w} ，它距离原点的偏移量由偏置参数 w_0 控制。此外，一个一般的点 \mathbf{x} 与决策面的有符号的正交距离为 $y(\mathbf{x})/\|\mathbf{w}\|$ 。

4.1 判别函数

判别函数是一个以向量 \mathbf{x} 为输入，把它分配到 K 个类别中的某一个类别（记作 C_k ）的函数。本章中，我们把我们的精力集中于线性判别函数（linear discriminant function），即那些决策面是超平面的判别函数。为了简化讨论，我们首先考虑两类的情形，然后把讨论扩展到 $K > 2$ 的情形。

4.1.1 二分类

线性判别函数的最简单的形式是输入向量的线性函数，即

$$y(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + w_0 \quad (4.4)$$

其中 \mathbf{w} 被称为权向量（weight vector）， w_0 被称为偏置（bias）。注意不要把这里的偏置与统计学中的偏置弄混淆。偏置的相反数有时被称为阈值（threshold）。对于一个输入向量 \mathbf{x} ，如果 $y(\mathbf{x}) \geq 0$ ，那么它被分到 C_1 中，否则被分到 C_2 中。对应的决策边界因此由 $y(\mathbf{x}) = 0$ 确定，它对应着 D 维空间的一个 $(D-1)$ 维的超平面。考虑两个点 \mathbf{x}_A 和 \mathbf{x}_B ，两个点都位于决策面上。由于 $y(\mathbf{x}_A) = y(\mathbf{x}_B) = 0$ ，我们有 $\mathbf{w}^T (\mathbf{x}_A - \mathbf{x}_B) = 0$ ，因此向量 \mathbf{w} 与决策面内的任何向量都正交，从而 \mathbf{w} 确定了决策面的方向。类似地，如果 \mathbf{x} 是决策面内的一个点，那么 $y(\mathbf{x}) = 0$ ，因此从原点到决策面的垂直距离为

$$\frac{\mathbf{w}^T \mathbf{x}}{\|\mathbf{w}\|} = -\frac{w_0}{\|\mathbf{x}\|} \quad \boxed{\text{表达式写错了}}$$
✓
表达式写错了
(4.5)

因此我们看到偏置参数 w_0 确定了决策面的位置。图 4.1 给出了 $D = 2$ 的情况下的这些性质。

此外，我们注意到 $y(\mathbf{x})$ 的值给出了点 \mathbf{x} 到决策面的垂直距离 r 的一个有符号的度量。为了说明这一点，考虑任意一点 \mathbf{x} 和它在决策面上的投影 \mathbf{x}_{\perp} ，我们有

$$\mathbf{x} = \mathbf{x}_{\perp} + r \frac{\mathbf{w}}{\|\mathbf{w}\|} \quad (4.6)$$

将这个等式的两侧同时乘以 \mathbf{w}^T ，然后加上 w_0 ，并且使用 $y(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + w_0$ 以及 $y(\mathbf{x}_{\perp}) = \mathbf{w}^T \mathbf{x}_{\perp} + w_0 = 0$ ，我们有

$$r = \frac{y(\mathbf{x})}{\|\mathbf{w}\|} \quad \boxed{\text{范数，表达式：初中学过，点到直线的距离。}}$$
✓
范数，表达式：初中学过，点到直线的距离。
(4.7)

图 4.1 说明了这个结果。

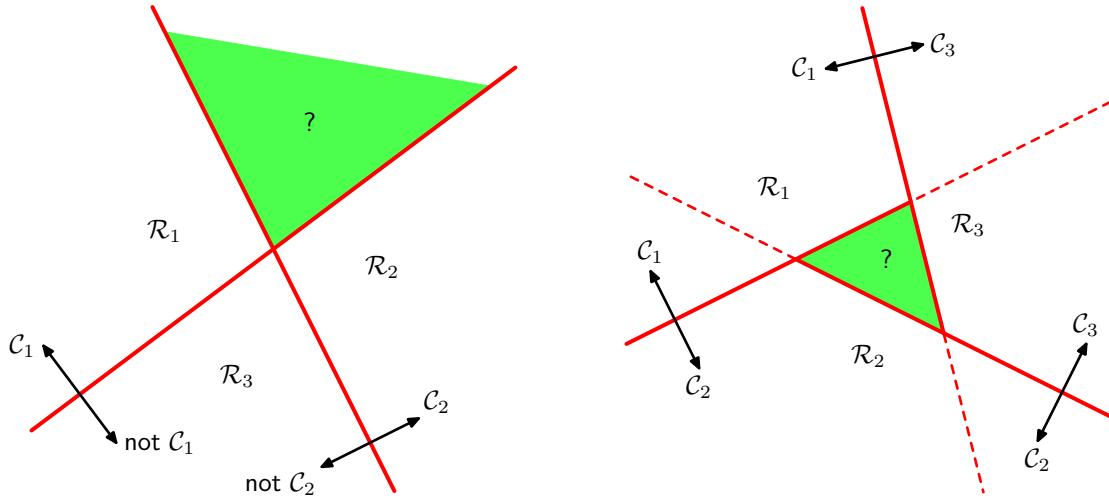


图 4.2: 尝试从一组两类的判别准则中构建出一个 K 类的判别准则会导致具有奇异性的区域，用绿色表示。左侧给出的例子涉及到使用两个判别准则，这两个判别准则将属于类别 C_k 的点与不属于类别 C_k 的点区分开。右侧给出的例子涉及到三个判别函数，每个函数用来区分一对类别 C_k 和 C_j 。

与第3章线性回归模型相同，我们可以引入一个额外的虚“输入” $x_0 = 1$ ，这会使得记号更简洁，比较方便。引入“虚”输入后，我们定义 $\tilde{\mathbf{w}} = (w_0, \mathbf{w})$ 以及 $\tilde{\mathbf{x}} = (x_0, \mathbf{x})$ ，从而

$$y(\mathbf{x}) = \tilde{\mathbf{w}}^T \tilde{\mathbf{x}} \quad (4.8)$$

在这种情况下，决策面是一个 D 维超平面，并且这个超平面会穿过 $D + 1$ 维扩展输入空间的原点。

4.1.2 多分类

提出了两种多分类组合的解决：

1, $k-1$ 个分类器，1 对其他，会出现同时属于 1 的情况。

2, 引入 $k(k-1)/2$ 的判别函数，出现不属于某一类的情况。

现在考虑把线性判别函数推广到 $K > 2$ 个类别。我们可能会尝试把多个二分类判别函数结合起来，构造一个 K 类判别函数。但是，这会产生一些严重的困难 (Duda and Hart, 1973)，正如我们马上要说明的那样。

考虑使用 $K - 1$ 个分类器，每个分类器用来解决一个二分类问题，把属于类别 C_k 和不属于那个类别的点分开。这被称为“1对其他”(one-versus-the-rest) 分类器。图4.2的左侧给出了一个涉及到三个类别的例子。这个例子中，这种方法产生了输入空间中无法分类的区域。

另一种方法是引入 $\frac{K(K-1)}{2}$ 个二元判别函数，对每一对类别都设置一个判别函数。这被称为“1对1”(one-versus-one) 分类器。这样，每个点的类别根据这些判别函数中的大多数输出类别确定。但是，这也会造成输入空间中的无法分类的区域，如图4.2右侧的图所示。

通过引入一个 K 类判别函数，我们可以避免这些问题。这个 K 类判别函数由 K 个线性函数组成，形式为

$$y_k(\mathbf{x}) = \mathbf{w}_k^T \mathbf{x} + w_{k0} \quad (4.9)$$

通过和其他类别的比较，将概率最大变形为对应的值最大，判别类别。

然后对于点 \mathbf{x} ，如果对于所有的 $j \neq k$ 都有 $y_k(\mathbf{x}) > y_j(\mathbf{x})$ ，那么就把它分到 C_k 。于是类别 C_k 和 C_j 之间的决策面为 $y_k(\mathbf{x}) = y_j(\mathbf{x})$ ，并且对应于一个 $(D - 1)$ 维超平面，形式为

$$(\mathbf{w}_k - \mathbf{w}_j)^T \mathbf{x} + (w_{k0} - w_{j0}) = 0 \quad (4.10)$$

这与4.1.1节讨论的二分类情形的决策边界的形式相同，因此也有类似的几何性质。

这样的判别函数的决策区域总是单连通的，并且是凸的。为了说明这一点，考虑两个点 \mathbf{x}_A 和 \mathbf{x}_B ，两个点都位于决策区域 R_k 中，如图4.3所示。任何位于连接 \mathbf{x}_A 和 \mathbf{x}_B 的线段上的点都可以表示成下面的形式

$$\hat{\mathbf{x}} = \lambda \mathbf{x}_A + (1 - \lambda) \mathbf{x}_B \quad (4.11)$$

备注：三次积分，单连通

其中 $0 \leq \lambda \leq 1$ 。根据判别函数的线性性质，有

$$y_k(\hat{\mathbf{x}}) = \lambda y_k(\mathbf{x}_A) + (1 - \lambda) y_k(\mathbf{x}_B) \quad (4.12)$$

线性性质：输入值进行线性组合，对应的函数值进行线性组合。

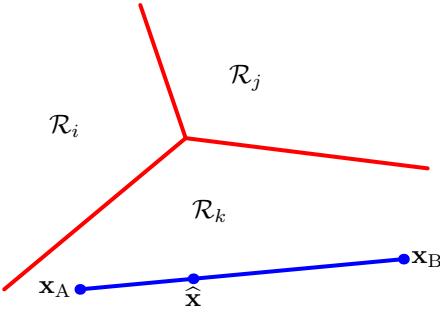


图 4.3: 多类判别函数的决策区域的说明, 决策边界用红色表示。如果两个点 x_A 和 x_B 位于同一个决策区域 \mathcal{R}_k , 那么任何位于连接这两个点的线段上的点 \hat{x} 一定位于区域 \mathcal{R}_k 内, 因此决策区域一定是单连通的、凸的。

任意两点能够直线连通。

由于 x_A 和 x_B 位于 \mathcal{R}_k 内部, 因此对于所有 $j \neq k$, 都有 $y_k(x_A) > y_j(x_A)$ 以及 $y_k(x_B) > y_j(x_B)$, 因此 $y_k(\hat{x}) > y_j(\hat{x})$, 从而 \hat{x} 也位于 \mathcal{R}_k 内部, 即 \mathcal{R}_k 是单连通的并且是凸的。

注意对于二分类的情形, 我们既可以使用这里讨论的方法, 基于两个判别函数 $y_1(x)$ 和 $y_2(x)$, 也可以使用 4.1.1 节给出的更简单的但是等价的方法, 基于单一的判别函数 $y(x)$ 。

我们现在介绍三种学习线性判别函数的参数的方法, 即基于最小平方的方法、Fisher 线性判别函数, 以及感知器算法。

4.1.3 用于分类的最小平方方法

- 确定判别函数参数：
1, 基于最小平方的方法
2, Fisher 线性判别函数
3, perceptron 模型

在第3章中, 我们考虑了由参数的线性函数组成的模型。我们看到, 最小平方误差函数的最小化产生了参数值的简单的解析解。因此, 我们很想考察一下能否把同样的方法用于分类问题。考虑一个一般的 K 分类问题, 其中目标向量 t 使用了“1-of- K ”二元表示方式。这种设置下, 使用最小平方方法的一个理由是它在给定输入向量的情况下, 近似了目标值的条件期望 $E[t | x]$ 。对于二元表示方法, 条件期望由后验类概率向量给出。但是不幸的是, 这些概率通常很难近似。事实上, 近似的过程有可能产生位于区间 $(0, 1)$ 之外的值, 这是因为线性模型的灵活性很受限, 正如我们稍后要讨论的那样。

每个类别 C_k 有自己的线性模型描述, 即

$$y_k(x) = \mathbf{w}_k^T \mathbf{x} + w_{k0} \quad (4.13)$$

其中 $k = 1, \dots, K$ 。使用向量记号, 我们可以很容易地把这些量聚集在一起表示, 即

$$\mathbf{y}(x) = \tilde{\mathbf{W}}^T \tilde{\mathbf{x}} \quad (4.14)$$

其中 $\tilde{\mathbf{W}}$ 是一个矩阵, 第 k 列由 $D + 1$ 维向量 $\tilde{\mathbf{w}}_k = (w_{k0}, \mathbf{w}_k^T)^T$ 组成, $\tilde{\mathbf{x}}$ 是对应的增广输入向量 $(1, \mathbf{x}^T)^T$, 它带有一个虚输入 $x_0 = 1$ 。这个表示方法在 3.1 节详细讨论过。这样, 一个新的输入 \mathbf{x} 被分配到输出 $y_k = \tilde{\mathbf{w}}_k^T \tilde{\mathbf{x}}$ 最大的类别中。

我们现在通过最小化平方和误差函数来确定参数矩阵 $\tilde{\mathbf{W}}$, 正如我们在第3章中对于回归问题做的那样。考虑一个训练数据集 $\{\mathbf{x}_n, t_n\}$, 其中 $n = 1, \dots, N$, 然后定义一个矩阵 \mathbf{T} , 它的第 n 行是向量 t_n^T 。我们还定义了一个矩阵 $\tilde{\mathbf{X}}$, 它的第 n 行是 $\tilde{\mathbf{x}}_n^T$ 。这样, 平方和误差函数可以写成

$$E_D(\tilde{\mathbf{W}}) = \frac{1}{2} \text{Tr} \left\{ (\tilde{\mathbf{X}} \tilde{\mathbf{W}} - \mathbf{T})^T (\tilde{\mathbf{X}} \tilde{\mathbf{W}} - \mathbf{T}) \right\} \quad (4.15)$$

令上式关于 $\tilde{\mathbf{W}}$ 的导数等于零, 整理, 可以得到 $\tilde{\mathbf{W}}$ 的解, 形式为

$$\tilde{\mathbf{W}} = (\tilde{\mathbf{X}}^T \tilde{\mathbf{X}})^{-1} \tilde{\mathbf{X}}^T \mathbf{T} = \tilde{\mathbf{X}}^\dagger \mathbf{T} \quad (4.16)$$

其中 $\tilde{\mathbf{X}}^\dagger$ 是矩阵 $\tilde{\mathbf{X}}$ 的伪逆矩阵, 正如 3.1.1 节讨论的那样。这样我们得到了判别函数, 形式为

$$y(x) = \tilde{\mathbf{W}}^T \tilde{\mathbf{x}} = \mathbf{T}^T (\tilde{\mathbf{X}}^\dagger)^T \tilde{\mathbf{x}} \quad (4.17)$$

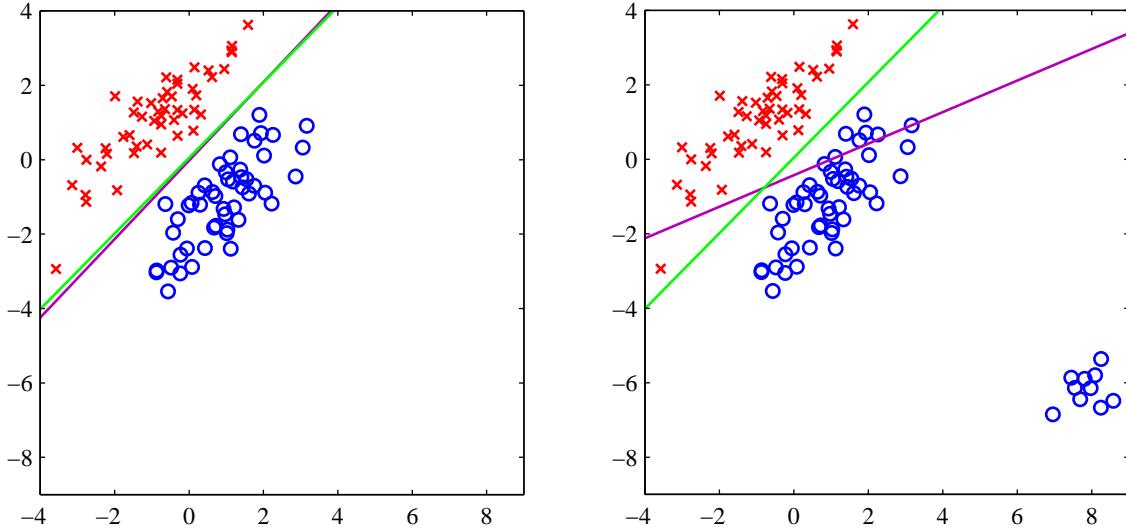


图 4.4: 左图给出了来自两个类别的数据，用红色叉形和蓝色圆圈表示。同时给出的还有通过最小平方方法找到的决策边界（洋红色曲线）以及logistic回归模型给出的决策边界（绿色曲线），这将在4.3.2节中讨论。右图给出了当额外的数据点被添加到左图的底部之后得到的结果，这表明最小平方方法对于异常点很敏感，这与logistic回归不同。

多目标变量的最小平方解的一个有趣的性质是，如果训练集里的每个目标向量都满足某个线性限制

$$\mathbf{a}^T \mathbf{t}_n + b = 0 \quad (4.18)$$

其中 a 和 b 为常数，那么对于任何 x 值，模型的预测也满足同样的限制，即

$$\mathbf{a}^T \mathbf{y}(\mathbf{x}) + b = 0 \quad (4.19)$$

因此如果我们使用 K 分类的“1-of- K ”表达方式，那么这个模型做出的预测会具有下面的性质：对于任意的 \mathbf{x} 的值， $\mathbf{y}(\mathbf{x})$ 的元素的和等于1。但是，这个对于加和的限制本身并不能够让模型的输出表示为概率的形式，因为它们没有被限制在区间(0, 1)中。

最小平方方法对于判别函数的参数给出了精确的解析解。但是，即使作为一个判别函数（我们使用它直接进行预测，抛弃掉所有的概率的表示），它仍然有很严重的问题。我们已经看到，最小平方解对于离群点缺少鲁棒性，这一点对于分类问题也是一样的，如图4.4所示。这里，我们看到，右图中的额外的数据点对决策边界的位置产生了极大的改变，即使这也点能够被左图中的原始的决策边界正确地分类。平方和误差函数惩罚了“过于正确”的预测，因为他们正确的一侧距离决策边界太远了。在第7.1.2节，我们会考虑几种其他的用于分类的误差函数，我们会看到这些误差函数不会有这种问题。

但是，最小平方方法的问题实际上比简单的缺乏鲁棒性更加严重，如图4.5所示。这幅图给出了二维空间 (x_1, x_2) 中，来自三个类别的人工生成的数据。线性决策边界能够将数据点完美地分开。实际上，在本章的后面将要介绍的逻辑回归方法可以给出一个令人满意的解，如右侧的图所示。然而，最小平方方法给出的结果相当差，输入空间中只有一个相当小的区域被分给了绿色的类别。

最小平方方法的失败并不让我们感觉惊讶。回忆一下，最小平方方法对应于高斯条件分布假设下的最大似然法，而二值目标向量的概率分布显然不是高斯分布。通过使用更恰当的概率模型，我们会得到性质比最小平方方法更好的分类方法。但是现在，我们继续研究另外的非概率方法来设置线性分类模型中的参数。

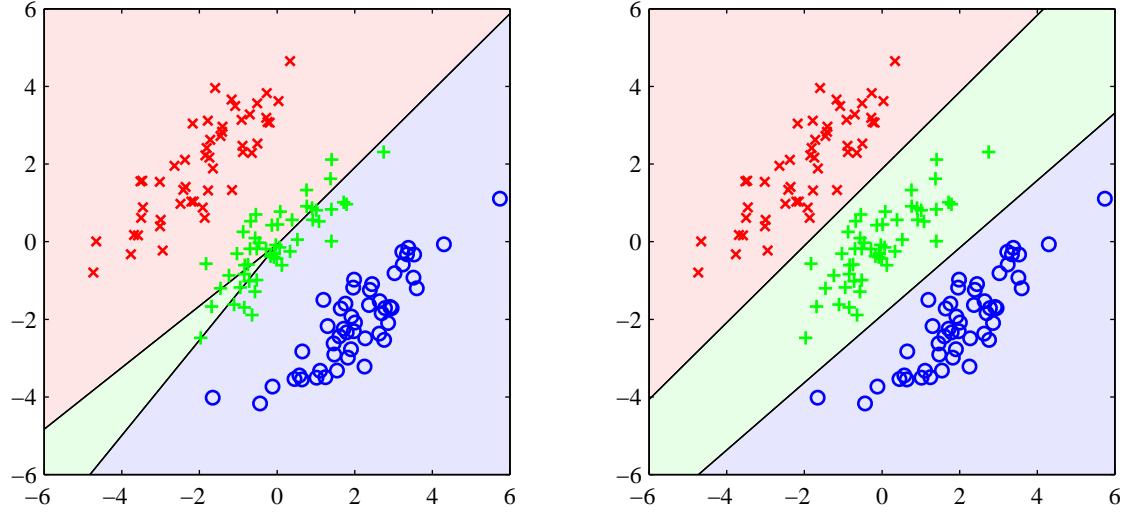


图 4.5: 由三个类别组成的人工数据集的例子，训练数据点分别用红色（ \times ）、绿色（ $+$ ）、蓝色（ \circ ）标出。直线表示决策边界，背景颜色表示决策区域代表的类别。左图是一个使用最小平方判别函数的结果。我们看到分配到绿色类别的输入空间的区域过小，大部分来自这个类别的点都被错误分类。右图是使用4.3.2节描述的使用logistic回归的结果，给出了训练数据的正确分类情况。

4.1.4 Fisher 线性判别函数

降维做投影的问题，通过Fisher线性判别函数选择投影界面。

我们可以从维度降低的角度考察线性分类模型。首先考虑二分类的情形。假设我们有一个 D 维输入向量 \mathbf{x} ，然后使用下式投影到一维

$$y = \mathbf{w}^T \mathbf{x} \quad (4.20)$$

如果我们在 y 上设置一个阈值，然后把 $y \geq -w_0$ 的样本分为 \mathcal{C}_1 类，把其余的样本分为 \mathcal{C}_2 类，那么我们就得到了之前讨论的标准的线性分类器。通常来说，向一维投影会造成相当多的信息丢失，因此在原始的 D 维空间能够完美地分离开的样本可能在一维空间中会相互重叠。但是，通过调整权向量 \mathbf{w} ，我们可以选择让类别之间分开最大的一个投影。首先，考虑一个二分类问题，这个问题中有 \mathcal{C}_1 类的 N_1 个点以及 \mathcal{C}_2 类的 N_2 个点。因此两类的均值向量为

$$\mathbf{m}_1 = \frac{1}{N_1} \sum_{n \in \mathcal{C}_1} \mathbf{x}_n, \quad \mathbf{m}_2 = \frac{1}{N_2} \sum_{n \in \mathcal{C}_2} \mathbf{x}_n \quad (4.21)$$

如果投影到 \mathbf{w} 上，那么最简单的度量类别之间分开程度的方式就是类别均值投影之后的距离。这说明，我们可以选择 \mathbf{w} 使得下式取得最大值

$$m_2 - m_1 = \mathbf{w}^T (\mathbf{m}_2 - \mathbf{m}_1) \quad (4.22)$$

其中

$$m_k = \mathbf{w}^T \mathbf{m}_k \quad (4.23)$$

是来自类别 \mathcal{C}_k 的投影数据的均值。但是，通过增大 \mathbf{w} ，这个表达式可以任意大。为了解决这个问题，我们可以将 \mathbf{w} 限制为单位长度，即 $\sum_i w_i^2 = 1$ 。使用拉格朗日乘数法来进行有限制条件的最大化问题的求解，我们可以发现 $\mathbf{w} \propto (\mathbf{m}_2 - \mathbf{m}_1)$ 。但是，这个方法还有一个问题，如图4.6所示。这幅图中的两个类别在原始二维空间 (x_1, x_2) 中可以完美地被分开，但是当投影到连接它们的均值的直线上时，就有了一定程度的重叠。如果类概率分布的协方差矩阵与对角化矩阵差距较大，那么这种问题就会出现。Fisher提出的思想是最大化一个函数，这个函数能够让类均值的投影分开得较大，同时让每个类别内部的方差较小，从而最小化了类别的重叠。

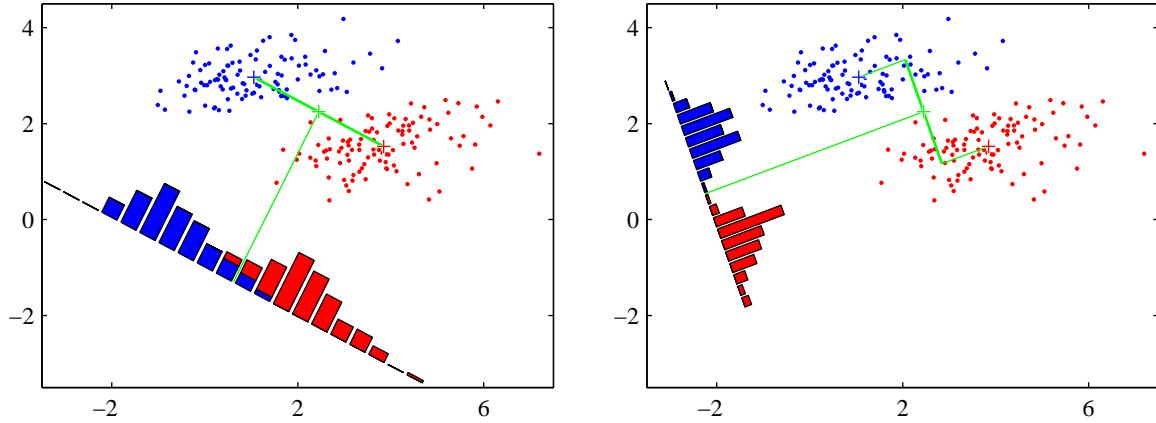


图 4.6: 左图给出了来自两个类别（表示为红色和蓝色）的样本，以及在连接两个类别的均值的直线上的投影的直方图。注意，在投影空间中，存在一个比较严重的类别重叠。右图给出的基于Fisher线性判别准则的对应投影，表明了类别切分的效果得到了极大的提升。

投影公式 (4.20) 将 x 的一组有标记的数据点变换为一位空间 y 的一组有标记数据点。来自类别 \mathcal{C}_k 的数据经过变换后的类内方差为

$$s_k^2 = \sum_{n \in \mathcal{C}_k} (y_n - m_k)^2 \quad \boxed{\text{变换后的方差}} \quad (4.24)$$

其中 $y_n = \mathbf{w}^T \mathbf{x}_n$ 。我们可以把整个数据集的总的类内方差定义为 $s_1^2 + s_2^2$ 。Fisher准则根据类间方差和类内方差的比值定义，即

$$J(\mathbf{w}) = \frac{(m_2 - m_1)^2}{s_1^2 + s_2^2} \quad \boxed{\text{类间方差大/类内方差小}} \quad (4.25)$$

我们可以使用公式 (4.20)、公式 (4.23) 和公式 (4.24) 对这个式子重写，显式地表达出 $J(\mathbf{w})$ 对 \mathbf{w} 的依赖。

$$J(\mathbf{w}) = \frac{\mathbf{w}^T \mathbf{S}_B \mathbf{w}}{\mathbf{w}^T \mathbf{S}_W \mathbf{w}} \quad (4.26)$$

其中 \mathbf{S}_B 是类间 (between-class) 协方差矩阵，形式为

$$\mathbf{S}_B = (\mathbf{m}_2 - \mathbf{m}_1)(\mathbf{m}_2 - \mathbf{m}_1)^T \quad (4.27)$$

\mathbf{S}_W 被称为类内 (within-class) 协方差矩阵，形式为

$$\mathbf{S}_W = \sum_{n \in \mathcal{C}_1} (\mathbf{x}_n - \mathbf{m}_1)(\mathbf{x}_n - \mathbf{m}_1)^T + \sum_{n \in \mathcal{C}_2} (\mathbf{x}_n - \mathbf{m}_2)(\mathbf{x}_n - \mathbf{m}_2)^T \quad (4.28)$$

对公式 (4.26) 关于 \mathbf{w} 求导，我们发现 $J(\mathbf{w})$ 取得最大值的条件为

$$(\mathbf{w}^T \mathbf{S}_B \mathbf{w}) \mathbf{S}_W \mathbf{w} = (\mathbf{w}^T \mathbf{S}_W \mathbf{w}) \mathbf{S}_B \mathbf{w} \quad (4.29)$$

根据公式 (4.27)，我们看到 $\mathbf{S}_B \mathbf{w}$ 总是在 $(\mathbf{m}_2 - \mathbf{m}_1)$ 的方向上。更重要的是，我们不关心 \mathbf{w} 的大小，只关心它的方向，因此我们可以忽略标量因子 $(\mathbf{w}^T \mathbf{S}_B \mathbf{w})$ 和 $(\mathbf{w}^T \mathbf{S}_W \mathbf{w})$ 。将公式 (4.29) 的两侧乘以 \mathbf{S}_W^{-1} ，我们有

$$\mathbf{w} \propto \mathbf{S}_W^{-1}(\mathbf{m}_2 - \mathbf{m}_1) \quad (4.30)$$

注意，如果类内协方差矩阵是各向同性的，从而 \mathbf{S}_W 正比于单位矩阵，那么我们看到 \mathbf{w} 正比于类均值的差。

公式 (4.30) 的结果被称为 Fisher 线性判别函数 (Fisher linear discriminant)，虽然严格来说它并不是一个判别函数，而是对于数据向一维投影的方向的一个具体选择。然而，投影的数据

可以接下来被用于构建判别函数，构建的方法为：选择一个阈值 y_0 ，使得当 $y(\mathbf{x}) \geq y_0$ 时，我们把数据点分到 \mathcal{C}_1 ，否则我们把数据点分到 \mathcal{C}_2 。例如，我们可以使用高斯概率分布对类条件概率密度 $p(y | \mathcal{C}_k)$ 建模，然后使用1.2.4节的方法通过最大似然方法找到高斯分布的参数值。找到投影类别的高斯近似之后，1.5.1节的方法给出了最优的阈值表达式。我们注意到 $y = \mathbf{w}^T \mathbf{x}$ 是一组随机变量的和，因此根据中心极限定理，我们可以做出高斯分布的假设。

4.1.5 与最小平方的关系

最小平方方法确定线性判别函数的目标是使模型的预测尽可能地与目标值接近。相反，Fisher判别准则的目标是使输出空间的类别有最大的区分度。考察一下这两种方法之间的关系是很有趣的。特别地，我们会证明，对于二分类问题，Fisher准则可以看成最小平方的一个特例。

目前为止，我们已经考虑了目标变量的“1-of- K ”表示方法。然而，如果我们使用一种稍微不同的表达方法，那么权值的最小平方解就会变得等价于Fisher解（Duda and Hart, 1973）。特别地，我们让属于 \mathcal{C}_1 的目标值等于 $\frac{N}{N_1}$ ，其中 N_1 是类别 \mathcal{C}_1 的模式的数量， N 是总的模式数量。这个目标值近似于类别 \mathcal{C}_1 的先验概率的导数。对于类别 \mathcal{C}_2 ，我们令目标值等于 $-\frac{N}{N_2}$ ，其中 N_2 是类别 \mathcal{C}_2 的模式的数量。

平方和误差函数可以写成

$$E = \frac{1}{2} \sum_{n=1}^N (\mathbf{w}^T \mathbf{x}_n + w_0 - t_n)^2 \quad (4.31)$$

令 E 关于 w_0 和 \mathbf{w} 的导数等于零，我们有

$$\sum_{n=1}^N (\mathbf{w}^T \mathbf{x}_n + w_0 - t_n) = 0 \quad (4.32)$$

$$\sum_{n=1}^N (\mathbf{w}^T \mathbf{x}_n + w_0 - t_n) \mathbf{x}_n = 0 \quad (4.33)$$

根据公式 (4.32)，使用我们对于目标值 t_n 的表示方法，我们可以得到偏置的表达式

$$w_0 = -\mathbf{w}^T \mathbf{m} \quad (4.34)$$

其中我们使用了下面的结果

$$\sum_{n=1}^N t_n = N_1 \frac{N}{N_1} - N_2 \frac{N}{N_2} = 0 \quad (4.35)$$

其中 \mathbf{m} 是所有数据的均值，定义为

$$\mathbf{m} = \frac{1}{N} \sum_{n=1}^N \mathbf{x}_n = \frac{1}{N} (N_1 \mathbf{m}_1 + N_2 \mathbf{m}_2) \quad (4.36)$$

通过一些简单的计算，并且再次使用我们对于 t_n 的新的表示方法，方程 (4.33) 变为

$$\left(\mathbf{S}_W + \frac{N_1 N_2}{N} \mathbf{S}_B \right) \mathbf{w} = N(\mathbf{m}_1 - \mathbf{m}_2) \quad (4.37)$$

其中 \mathbf{S}_W 由公式 (4.28) 定义， \mathbf{S}_B 由公式 (4.27) 定义，并且我们使用公式 (4.34) 的结果替换了偏置。使用公式 (4.27)，我们注意到 $\mathbf{S}_B \mathbf{w}$ 总是在 $(\mathbf{m}_2 - \mathbf{m}_1)$ 的方向上。因此我们有

$$\mathbf{w} \propto \mathbf{S}_W^{-1} (\mathbf{m}_2 - \mathbf{m}_1) \quad (4.38)$$

其中，我们已经忽略了不相关的标量因子。因此权向量恰好与根据Fisher判别准则得到的结果相同。此外，我们也发现，偏置 w_0 的值由公式 (4.34) 给出。这告诉我们，对于一个新的向量 \mathbf{x} ，如果 $y(\mathbf{x}) = \mathbf{w}^T (\mathbf{x} - \mathbf{m}) > 0$ ，那么 \mathbf{x} 应该被分到 \mathcal{C}_1 ，否则应该被分到 \mathcal{C}_2 。

4.1.6 多分类的Fisher判别函数

我们现在考虑Fisher判别函数对于 $K > 2$ 个类别的推广。我们假设输入空间的维度 D 大于类别数量 K 。接下来，我们引入 $D' > 1$ 个线性“特征” $y_k = \mathbf{w}_k^T \mathbf{x}$ ，其中 $k = 1, \dots, D'$ 。为了方便，这些特征值可以聚集起来组成向量 \mathbf{y} 。类似地，权向量 $\{\mathbf{w}_k\}$ 可以被看成矩阵 \mathbf{W} 的列。因此

$$\mathbf{y} = \mathbf{W}^T \mathbf{x} \quad (4.39)$$

注意，与之前一样，我们在 \mathbf{y} 的定义中没有包含任何的偏置参数。类内协方差矩阵可以使用公式(4.28)推广到 K 类，有

$$\mathbf{S}_W = \sum_{k=1}^K \mathbf{S}_k \quad (4.40)$$

其中

$$\mathbf{S}_k = \sum_{n \in \mathcal{C}_k} (\mathbf{x}_n - \mathbf{m}_k)(\mathbf{x}_n - \mathbf{m}_k)^T \quad (4.41)$$

$$\mathbf{m}_k = \frac{1}{N_k} \sum_{n \in \mathcal{C}_k} \mathbf{x}_n \quad (4.42)$$

其中 N_k 是类别 \mathcal{C}_k 中模式的数量。为了找到类间协方差矩阵的推广，我们使用Duda and Hart (1973) 的方法，首先考虑整体的协方差矩阵

$$\mathbf{S}_T = \sum_{n=1}^N (\mathbf{x}_n - \mathbf{m})(\mathbf{x}_n - \mathbf{m})^T \quad (4.43)$$

其中 \mathbf{m} 是全体数据的均值

$$\mathbf{m} = \frac{1}{N} \sum_{n=1}^N \mathbf{x}_n = \frac{1}{N} \sum_{k=1}^K N_k \mathbf{m}_k \quad (4.44)$$

其中 $N = \sum_k N_k$ 是数据点的总数。整体的协方差矩阵可以分解为公式(4.40)和公式(4.41)给出的类内协方差矩阵，加上另一个矩阵 \mathbf{S}_B ，它可以看做类间协方差矩阵。

$$\mathbf{S}_T = \mathbf{S}_W + \mathbf{S}_B \quad (4.45)$$

其中

$$\mathbf{S}_B = \sum_{k=1}^K N_k (\mathbf{m}_k - \mathbf{m})(\mathbf{m}_k - \mathbf{m})^T \quad (4.46)$$

协方差矩阵被定义在原始的 \mathbf{x} 空间中。我们现在在投影的 D' 维 \mathbf{y} 空间中定义类似的矩阵

$$\mathbf{S}_W = \sum_{k=1}^K \sum_{n \in \mathcal{C}_k} (\mathbf{y}_n - \boldsymbol{\mu}_k)(\mathbf{y}_n - \boldsymbol{\mu}_k)^T \quad (4.47)$$

以及

$$\mathbf{S}_B = \sum_{k=1}^K N_k (\boldsymbol{\mu}_k - \boldsymbol{\mu})(\boldsymbol{\mu}_k - \boldsymbol{\mu})^T \quad (4.48)$$

其中

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{n \in \mathcal{C}_k} \mathbf{y}_n, \quad \boldsymbol{\mu} = \frac{1}{N} \sum_{k=1}^K N_k \boldsymbol{\mu}_k \quad (4.49)$$

与之前一样，我们想构造一个标量，当类间协方差较大且类内协方差较小时，这个标量会较大。有许多可能的准则选择方式(Fukunaga, 1990)。其中一种选择是

$$J(\mathbf{W}) = \text{Tr}\{\mathbf{s}_W^{-1} \mathbf{s}_B\} \quad (4.50)$$

这个判别准则可以显式地写成投影矩阵 \mathbf{W} 的函数，形式为

$$J(\mathbf{W}) = \text{Tr}\{(\mathbf{W}^T \mathbf{S}_W \mathbf{W})^{-1} (\mathbf{W}^T \mathbf{S}_B \mathbf{W})\} \quad (4.51)$$

最大化这个判别准则是很直接的，虽然有些麻烦。详细的推导可以参考Fukunaga (1990)。权值由 $\mathbf{S}_W^{-1} \mathbf{S}_B$ 的特征向量确定，它对应了 D' 个最大的特征值。

值得强调的是，有一个重要的结果对于所有的这些判别准则都成立。首先，根据公式 (4.46)， \mathbf{S}_B 由 K 个矩阵的和组成，每一个矩阵都是两个向量的外积，因此秩等于 1。此外，由于公式 (4.44) 给出的限制条件，这些矩阵中只有 $(K - 1)$ 个是相互独立的。因此 \mathbf{S}_B 的秩最大等于 $(K - 1)$ ，因此最多有 $(K - 1)$ 个非零特征值。这表明，向由 \mathbf{S}_B 张成的 $(K - 1)$ 维空间上的投影不会改变 $J(\mathbf{W})$ 的值，因此通过这种方法我们不能够找到多于 $(K - 1)$ 个线性“特征” (Fukunaga, 1990)。

4.1.7 感知器算法

线性判别模型的另一个例子是 Rosenblatt (1962) 提出的感知器算法。它在模式识别算法的历史上占有重要的地位。它对应于一个二分类的模型，这个模型中，输入向量 x 首先使用一个固定的非线性变换得到一个特征向量 $\phi(x)$ ，这个特征向量然后被用于构造一个一般的线性模型，形式为

$$y(\mathbf{x}) = f(\mathbf{w}^T \phi(\mathbf{x})) \quad (4.52)$$

其中非线性激活函数 $f(\cdot)$ 是一个阶梯函数，形式为

输入向量 -> 特征向量 ->
线性模型 -> 激活函数

$$f(a) = \begin{cases} +1, & a \geq 0 \\ -1, & a < 0 \end{cases} \quad (4.53)$$

向量 $\phi(\mathbf{x})$ 通常包含一个偏置分量 $\phi_0(\mathbf{x}) = 1$ 。在之前对于二分类问题的讨论中，我们对于目标变量的表示方法为 $t \in \{0, 1\}$ ，这对于概率模型来说是很合适的。然而，对于感知器来说，更方便的做法是使用 $t = +1$ 表示 \mathcal{C}_1 ，使用 $t = -1$ 表示 \mathcal{C}_2 ，这与激活函数的选择相匹配。

用来确定感知器的参数 \mathbf{w} 的算法可以很容易地从误差函数最小化的思想中得到。误差函数的一个自然的选择是误分类的模式的总数。但是，这样做会使得学习算法不会很简单，因为这样做会使误差函数变为 \mathbf{w} 的分段常函数，从而当 \mathbf{w} 的变化使得决策边界移过某个数据点时，这个函数会不连续变化。这样做还使得使用误差函数改变 \mathbf{w} 的方法无法使用，因为在几乎所有的地方梯度都等于零。

因此我们考虑一个另外的误差函数，被称为感知器准则 (perceptron criterion)。为了推导这个函数，我们注意到我们正在做的是寻找一个权向量 \mathbf{w} 使得对于类别 \mathcal{C}_1 中的模式 \mathbf{x}_n 都有 $\mathbf{w}^T \phi(\mathbf{x}_n) > 0$ ，而对于类别 \mathcal{C}_2 中的模式 \mathbf{x}_n 都有 $\mathbf{w}^T \phi(\mathbf{x}_n) < 0$ 。使用 $t \in \{-1, +1\}$ 这种目标变量的表示方法，我们要做的就是使得所有的模式都满足 $\mathbf{w}^T \phi(\mathbf{x}_n) t_n > 0$ 。对于正确分类的模式，感知器准则赋予零误差，而对于误分类的模式 \mathbf{x}_n ，它试着最小化 $-\mathbf{w}^T \phi(\mathbf{x}_n) t_n$ 。因此，感知器准则为

$$E_P(\mathbf{w}) = - \sum_{n \in \mathcal{M}} \mathbf{w}^T \phi_n t_n \quad (4.54)$$

其中 $\phi_n = \phi(\mathbf{x}_n)$ 和 \mathcal{M} 表示所有误分类模式的集合。某个特定的误分类模式对于误差函数的贡献是 \mathbf{w} 空间中模式被误分类的区域中 \mathbf{w} 的线性函数，而在正确分类的区域，误差函数等于零。总的误差函数因此是分段线性的。

我们现在对这个误差函数使用随机梯度下降算法。这样，权向量 \mathbf{w} 的变化为

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} - \eta \nabla E_P(\mathbf{w}) = \mathbf{w}^{(\tau)} + \eta \phi_n t_n \quad (4.55)$$

其中 η 是学习率参数， τ 是一个整数，是算法运行次数的索引。如果我们将 \mathbf{w} 乘以一个常数，那么感知器函数 $y(\mathbf{x}, \mathbf{w})$ 不变，因此我们可以令学习率参数 η 等于 1 而不失一般性。注意，随着训练过程中权向量的不断改变，误分类的模式也会改变。

感知器学习算法可以简单地表示如下。我们反复对于训练模式进行循环处理，对于每个模式 \mathbf{x}_n 我们计算感知器函数 (4.52)。如果模式正确分类，那么权向量保持不变，而如果模式被

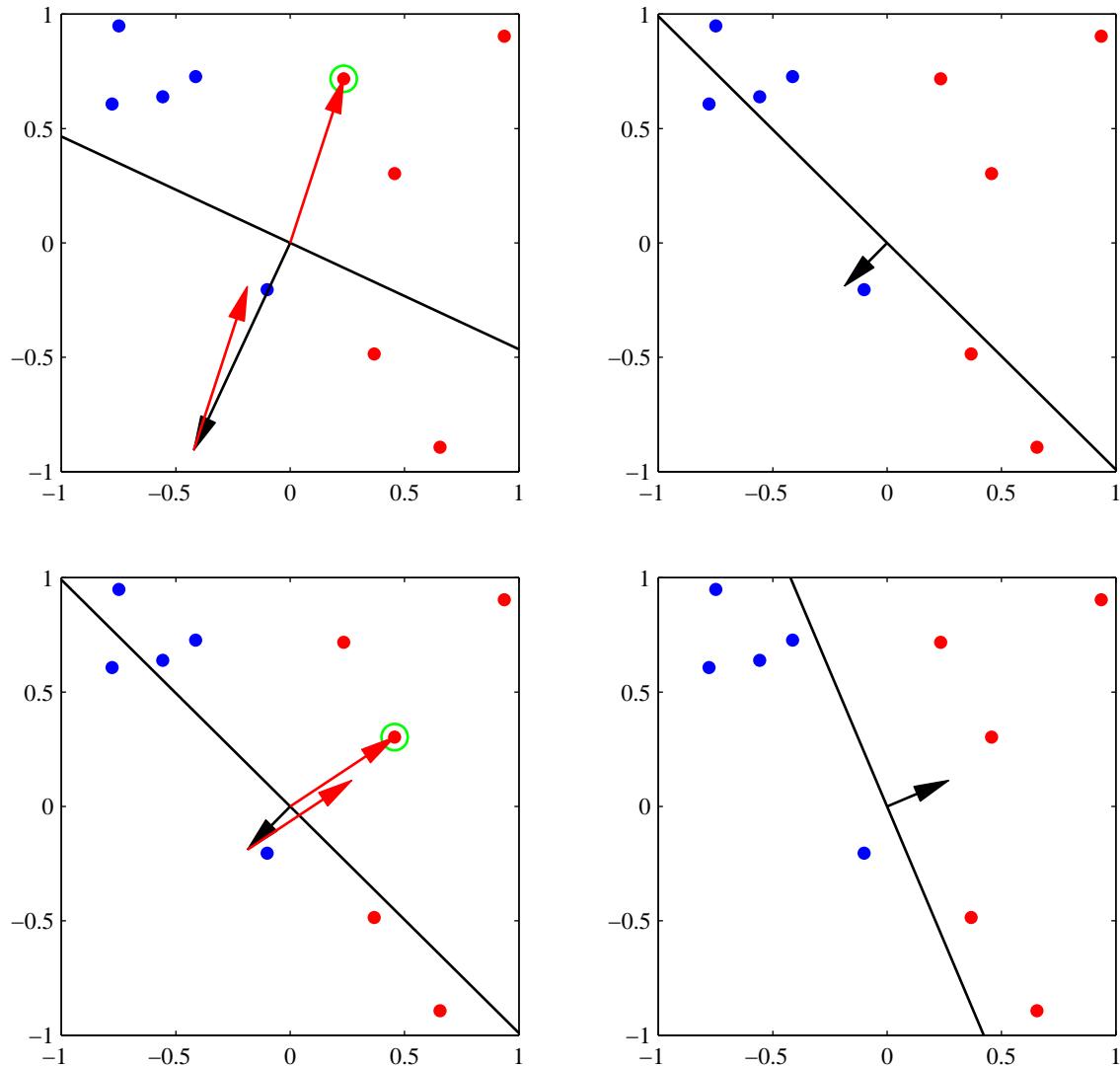


图 4.7: 感知器算法收敛性的说明,给出了二维特征空间(ϕ_1, ϕ_2)中的来自两个类别的数据点(红色和蓝色)。左上图给出了初始参数向量 w ,表示为黑色箭头,以及对应的决策边界(黑色直线),其中箭头指向被分类为红色类别的决策区域。用绿色圆圈标出的数据点被误分类,因此它的特征向量被加到当前的权向量中,给出了新的决策边界,如右上图所示。左下图给出了下一个误分类的点,用绿色圆圈标出,它的特征向量再次被加到权向量上,给出了右下图的决策边界。这个边界中所有的数据点都被正确分类。



图 4.8: Mark 1 感知器硬件。左图展示了输入是如何使用一个照相机得到的，其中输入空间，在这种情形下是一个印刷的汉字，被强光照射，从而一张图像被集中到了一个 20×20 的硫化镉光电管的阵列上，形成了一个原始的400像素图像。感知器也有一个接线板，如中图所示。它使得机器可以尝试输入特征的不同配置。通常这些线被随机连接，展示了感知器的学习能力不需要精确的接线，这与现代数字计算机不同。右图的照片展示了一个可调节权值的支架。每个权值使用一个滑动变阻器实现，这个滑动变阻器也被称为分压器。它被一个电动机驱动，因此使得权值可以通过学习算法自动被调节。

通过不断更新权向量
4.55, 不断更新决策边界

错误分类，那么对于类别 \mathcal{C}_1 ，我们把向量 $\phi(x_n)$ 加到当前对于权向量 w 的估计值上，而对于类别 \mathcal{C}_2 ，我们从 w 中减掉向量 $\phi(x_n)$ 。图4.7说明了感知器学习算法。

如果我们考虑感知器学习算法中一次权值更新的效果，我们可以看到，一个误分类模式对于误差函数的贡献会逐渐减小。因为根据公式 (4.55)，我们有

$$-\mathbf{w}^{(\tau+1)T} \phi_n t_n = -\mathbf{w}^{(\tau)T} \phi_n t_n - (\phi_n t_n)^T \phi_n t_n < -\mathbf{w}^{(\tau)T} \phi_n t_n \quad (4.56)$$

其中我们令 $\eta = 1$ ，并且使用了不等式 $\|\phi_n t_n\|^2 > 0$ 。当然，这并不表明其他的误分类模式对于误差函数的贡献会减小。此外，权向量的改变会使得某些之前正确分类的样本变为误分类。因此感知器学习规则并不保证在每个阶段都会减小整体的误差函数。

然而，感知器收敛定理 (perceptron convergence theorem) 表明，如果存在一个精确的解（即，如果训练数据线性可分），那么感知器算法可以保证在有限步骤内找到一个精确解。这个定理的证明可以参考Rosenblatt (1962)、Block (1962)、Nilsson (1965)、Minsky and Papert (1969)、Hertz et al. (1991) 以及Bishop (1995a)。但是，需要注意的是，达到收敛状态所需的步骤数量可能非常大，并且在实际应用中，在达到收敛状态之前，我们不能够区分不可分问题与缓慢收敛问题。

即使数据集是线性可分的，也可能有多个解，并且最终哪个解会被找到依赖于参数的初始化以及数据点出现的顺序。此外，对于线性不可分的数据集，感知器算法永远不会收敛。

除了学习算法的这些困难之处以外，感知器算法无法提供概率形式的输出，也无法直接推广到 $K > 2$ 个类别的情形。然而，最重要的局限性是它基于固定基函数的线性组合（本章中和前一章中讨论的所有模型都是这样）。关于感知器算法更多的局限性，可以参考Minsky and Papert (1969) 和Bishop (1995a)。

Rosenblatt建立了感知算法的一个模拟的硬件实现，使用发动机驱动的可变电阻来实现可调节参数 w_j 。图4.8说明了这一点。输入从一个摄像系统中得到，这个摄像系统基于光传感器阵列，而基函数 ϕ 可以选为不同的形式，例如基于输入图像的像素子集随机选择简单的固定基函数。典型的应用包括区分简单的图形和汉字。

同时，一个与感知器关系密切的系统adaline促进了感知器算法的发展。adaline是“adaptive linear element”的简称，由Widrow以及他的合作者开发。这个模型的函数形式与感知器相同，但是训练方法不同 (Widrow and Hoff, 1960; Widrow and Lehr, 1990)。

4.2 概率生成式模型

我们接下来用概率的观点考察分类问题，并且说明具有线性决策边界的模型如何通过对数据分布的简单假设得到。在1.5.4节，我们讨论了判别式模型和生成式模型的区别。这里我们会使用生成式的方法。这种方法中，我们对类条件概率密度 $p(x | \mathcal{C}_k)$ 和类先验概率分布 $p(\mathcal{C}_k)$ 建模，然后使用这两个概率密度通过贝叶斯定理计算后验概率密度 $p(\mathcal{C}_k | x)$ 。

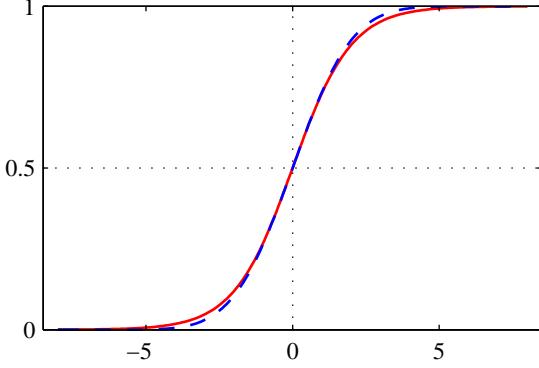


图 4.9: 由公式 (4.59) 定义的logistic sigmoid函数 $\sigma(a)$ 的图像, 用红色表示。同时给出的是放缩后的逆probit函数 $\Phi(\lambda a)$ 的图像, 其中 $\lambda^2 = \pi/8$, 用蓝色曲线表示, $\Phi(a)$ 由公式 (4.114) 定义。缩放因子 $\pi/8$ 使得两条曲线在 $a = 0$ 处的导数相同。

首先考虑二分类的情形。类别 \mathcal{C}_1 的后验概率可以写成

$$\begin{aligned} p(\mathcal{C}_1 | \mathbf{x}) &= \frac{p(\mathbf{x} | \mathcal{C}_1)p(\mathcal{C}_1)}{p(\mathbf{x} | \mathcal{C}_1)p(\mathcal{C}_1) + p(\mathbf{x} | \mathcal{C}_2)p(\mathcal{C}_2)} \\ &= \frac{1}{1 + \exp(-a)} = \sigma(a) \end{aligned} \quad (4.57)$$

其中我们定义了

$$a = \ln \frac{p(\mathbf{x} | \mathcal{C}_1)p(\mathcal{C}_1)}{p(\mathbf{x} | \mathcal{C}_2)p(\mathcal{C}_2)} \quad (4.58)$$

且 $\sigma(a)$ 是logistic sigmoid函数, 定义为

$$\sigma(a) = \frac{1}{1 + \exp(-a)} \quad (4.59)$$

图4.9给出了这个函数的图像。“sigmoid”的意思是“S形”。这种函数有时被称为“挤压函数”, 因为它把整个实数轴映射到了一个有限的区间中。我们在之前的章节中已经遇到了logistic sigmoid函数。这个函数在许多分类算法中都有着重要的作用。它满足下面的对称性

$$\sigma(-a) = 1 - \sigma(a) \quad (4.60)$$

这个性质很容易证明。logistic sigmoid的反函数为

$$a = \ln \left(\frac{\sigma}{1 - \sigma} \right) \quad (4.61)$$

被称为logit函数。它表示两类的概率比值的对数 $\ln \left[\frac{p(\mathcal{C}_1 | \mathbf{x})}{p(\mathcal{C}_2 | \mathbf{x})} \right]$, 也被称为log odds函数。

注意在公式 (4.57) 中, 我们只是把后验概率写成了一个等价的形式, 因此logistic sigmoid函数的出现似乎相当没有意义。然而, 假设 $a(\mathbf{x})$ 的函数形式相当简单, 那么这种表示方法就很有用了。我们稍后会考虑 $a(\mathbf{x})$ 是 \mathbf{x} 的线性函数的情形。这种情况下, 后验概率由一个通用的线性模型确定。

对于 $K > 2$ 个类别的情形, 我们有

$$\begin{aligned} p(\mathcal{C}_k | \mathbf{x}) &= \frac{p(\mathbf{x} | \mathcal{C}_k)p(\mathcal{C}_k)}{\sum_j p(\mathbf{x} | \mathcal{C}_j)p(\mathcal{C}_j)} \\ &= \frac{\exp(a_k)}{\sum_j \exp(a_j)} \end{aligned} \quad (4.62)$$

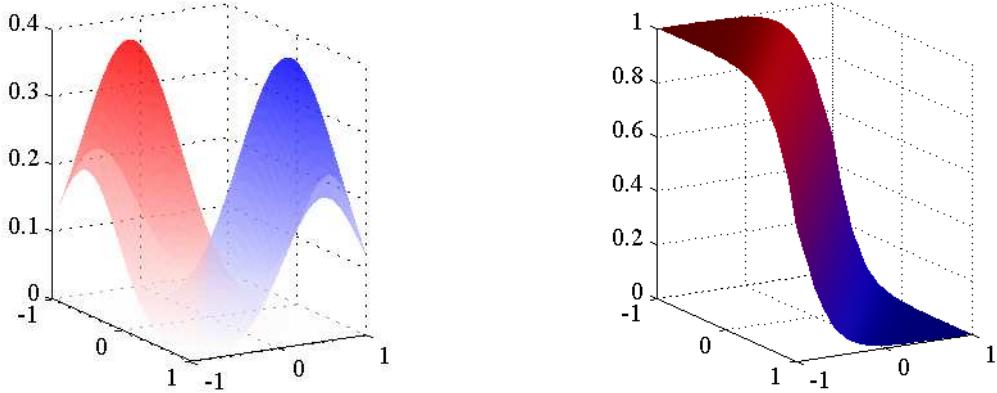


图 4.10: 左图给出了两个类别的类条件概率密度，分别用红色和蓝色表示。右图给出了对应的后验概率分布 $p(\mathcal{C}_1 | \mathbf{x})$ ，它由 \mathbf{x} 的线性函数的 logistic sigmoid 函数给出。右图的曲面的颜色中，红色所占的比例由 $p(\mathcal{C}_1 | \mathbf{x})$ 给出，蓝色所占的比例由 $p(\mathcal{C}_2 | \mathbf{x}) = 1 - p(\mathcal{C}_1 | \mathbf{x})$ 给出。

它被称为归一化指数 (normalized exponential)，可以被当做 logistic sigmoid 函数对于多类情况的推广。这里 a_k 被定义为

$$a_k = \ln p((\mathbf{x} | \mathcal{C}_k)p(\mathcal{C}_k)) \quad (4.63)$$

归一化指数也被称为 softmax 函数，因为它表示“max”函数的一个平滑版本。这是因为，如果对于所有的 $j \neq k$ 都有 $a_k \gg a_j$ ，那么 $p(\mathcal{C}_k | \mathbf{x}) \simeq 1$ 且 $p(\mathcal{C}_j | \mathbf{x}) \simeq 0$ 。

我们现在考虑选择具体的类条件概率密度形式的情况下的结果，首先讨论连续输入变量 \mathbf{x} 的情形，然后简短地讨论离散输入的情形。

4.2.1 连续输入

让我们假设类条件概率密度是高斯分布，然后求解后验概率的形式。首先，我们假定所有的类别的协方差矩阵相同。这样类别 \mathcal{C}_k 的类条件概率为

$$p(\mathbf{x} | \mathcal{C}_k) = \frac{1}{(2\pi)^{\frac{D}{2}} |\Sigma|^{\frac{1}{2}}} \exp \left\{ -\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu}_k)^T \Sigma^{-1} (\mathbf{x} - \boldsymbol{\mu}_k) \right\} \quad (4.64)$$

首先考虑两类的情形。根据公式 (4.57) 和公式 (4.58)，我们有

$$p(\mathcal{C}_1 | \mathbf{x}) = \sigma(\mathbf{w}^T \mathbf{x} + w_0) \quad (4.65)$$

其中我们定义了

$$\mathbf{w} = \Sigma^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) \quad (4.66)$$

$$w_0 = -\frac{1}{2} \boldsymbol{\mu}_1^T \Sigma^{-1} \boldsymbol{\mu}_1 + \frac{1}{2} \boldsymbol{\mu}_2^T \Sigma^{-1} \boldsymbol{\mu}_2 + \ln \frac{p(\mathcal{C}_1)}{p(\mathcal{C}_2)} \quad (4.67)$$

我们看到，高斯概率密度的指数项中 \mathbf{x} 的二次型消失了（这是因为我们假设类概率的协方差矩阵相同），从而得到了参数为 \mathbf{x} 的线性函数的 logistic sigmoid 函数。图 4.10 给出了二维输入空间 \mathbf{x} 的情况下的结果。最终求得的决策边界对应于后验概率 $p(\mathcal{C}_k | \mathbf{x})$ 为常数的决策面，因此由 \mathbf{x} 的线性函数给出，从而决策边界在输入空间是线性的。先验概率密度 $p(\mathcal{C}_k)$ 只出现在偏置参数 w_0 中，因此先验的改变的效果是平移决策边界，即平移后验概率中的常数轮廓线。

对于 K 个类别的一般情形，根据公式 (4.62) 和公式 (4.63)，我们有

$$a_k(\mathbf{x}) = \mathbf{w}_k^T \mathbf{x} + w_{k0} \quad (4.68)$$

其中我们定义了

$$\mathbf{w}_k = \Sigma^{-1} \boldsymbol{\mu}_k \quad (4.69)$$

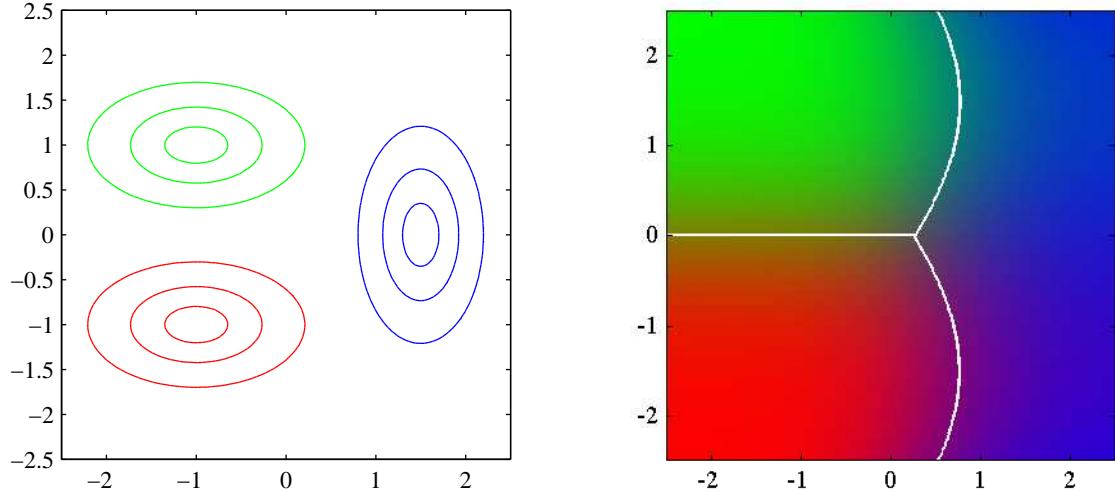


图 4.11: 左图给出了三个类别的类条件概率密度，每个都是高斯分布，分别用红色、绿色、蓝色表示，其中红色和绿色的类别有相同的协方差矩阵。右图给出了对应的后验概率分布，其中RGB的颜色向量表示三个类别各自的后验概率。决策边界也被画出。注意，具有相同协方差矩阵的红色类别和绿色类别的决策边界是线性的，而其他类别之间的类别的决策边界是二次的。

$$w_{k0} = -\frac{1}{2}\boldsymbol{\mu}_k^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_k + \ln p(\mathcal{C}_k) \quad (4.70)$$

我们看到 $a_k(\mathbf{x})$ 与之前一样是 \mathbf{x} 的线性函数，这是因为各个类别的协方差矩阵相同，使得二次项被消去。最终的决策边界，对应于最小错误分类率，会出现在后验概率最大的两个概率相等的位置，因此由 \mathbf{x} 的线性函数定义，从而我们再次得到了一个一般的线性模型。

如果我们不假设各个类别的协方差矩阵相同，允许每个类条件概率密度 $p(\mathbf{x} | \mathcal{C}_k)$ 有自己的协方差矩阵 $\boldsymbol{\Sigma}_k$ ，那么之前二次项消去的现象不会出现，从而我们会得到 \mathbf{x} 的二次函数，这就引出了二次判别函数（quadratic discriminant）。图 4.11 给出了线性决策边界和二次决策边界。

4.2.2 最大似然解

- 0 , 具体化条件概率。
- 1 , 生成后验的参数化形式（生成模型）。
- 2 , 通过最大似然求出参数的值。

一旦我们具体化了类条件概率密度 $p(\mathbf{x} | \mathcal{C}_k)$ 的参数化的函数形式，我们就能够使用最大似然法确定参数的值，以及先验类概率 $p(\mathcal{C}_k)$ 。这需要数据集由观测 \mathbf{x} 以及对应的类别标签组成。

首先考虑两类的情形，每个类别都有一个高斯类条件概率密度，且协方差矩阵相同。我们假设我们有一个数据集 $\{\mathbf{x}_n, t_n\}$ ，其中 $n = 1, \dots, N$ 。这里 $t_n = 1$ 表示类别 \mathcal{C}_1 ， $t_n = 0$ 表示类别 \mathcal{C}_2 。我们把先验概率记作 $p(\mathcal{C}_1) = \pi$ ，从而 $p(\mathcal{C}_2) = 1 - \pi$ 。对于一个来自类别 \mathcal{C}_1 的数据点 \mathbf{x}_n ，我们有 $t_n = 1$ ，因此

$$p(\mathbf{x}_n, \mathcal{C}_1) = p(\mathcal{C}_1)p(\mathbf{x}_n | \mathcal{C}_1) = \pi \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_1, \boldsymbol{\Sigma})$$

类似地，对于类别 \mathcal{C}_2 ，我们有 $t_n = 0$ ，因此

$$p(\mathbf{x}_n, \mathcal{C}_2) = p(\mathcal{C}_2)p(\mathbf{x}_n | \mathcal{C}_2) = (1 - \pi) \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_2, \boldsymbol{\Sigma})$$

于是似然函数为

$$p(\mathbf{t}, \mathbf{X} | \pi, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \boldsymbol{\Sigma}) = \prod_{n=1}^N [\pi \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_1, \boldsymbol{\Sigma})]^{t_n} [(1 - \pi) \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_2, \boldsymbol{\Sigma})]^{1-t_n} \quad (4.71)$$

其中 $\mathbf{t} = (t_1, \dots, t_N)^T$ 。与之前一样，最大化似然函数的对数比较方便。首先考虑关于 π 的最大化。对数似然函数中与 π 相关的项为

$$\sum_{n=1}^N \{t_n \ln \pi + (1 - t_n) \ln(1 - \pi)\} \quad (4.72)$$

令其关于 π 的导数等于零，整理，可得

$$\pi = \frac{1}{N} \sum_{n=1}^N t_n = \frac{N_1}{N} = \frac{N_1}{N_1 + N_2} \quad (4.73)$$

其中 N_1 表示类别 \mathcal{C}_1 的数据点的总数，而 N_2 表示类别 \mathcal{C}_2 的数据点总数。因此， π 的最大似然估计就是类别 \mathcal{C}_1 的点所占的比例，这与我们预期的相同。这个结果很容易推广到多类的情形。与两类的情况相同，在多类的情形中，类别 \mathcal{C}_k 的先验概率估计为这个类别的数据点数量占训练集总数据的比例。

现在考虑关于 μ_1 的最大化。与之前一样，我们把对数似然函数中与 μ_1 相关的量挑出来，即

$$\sum_{n=1}^N t_n \ln \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_1, \boldsymbol{\Sigma}) = -\frac{1}{2} \sum_{n=1}^N t_n (\mathbf{x}_n - \boldsymbol{\mu}_1)^T \boldsymbol{\Sigma}^{-1} (\mathbf{x}_n - \boldsymbol{\mu}_1) + \text{常数} \quad (4.74)$$

令它关于 $\boldsymbol{\mu}_1$ 的导数等于零，整理可得

$$\boldsymbol{\mu}_1 = \frac{1}{N_1} \sum_{n=1}^N t_n \mathbf{x}_n \quad (4.75)$$

这就是属于类别 \mathcal{C}_1 的输入向量 \mathbf{x}_n 的均值。通过类似的推导，对应的 $\boldsymbol{\mu}_2$ 的结果为

$$\boldsymbol{\mu}_2 = \frac{1}{N_2} \sum_{n=1}^N (1 - t_n) \mathbf{x}_n \quad (4.76)$$

与之前一样，它是属于类别 \mathcal{C}_2 的输入向量 \mathbf{x}_n 的均值。

最后，考虑协方差矩阵 $\boldsymbol{\Sigma}$ 的最大似然解。选出与 $\boldsymbol{\Sigma}$ 相关的项，我们有

$$\begin{aligned} & -\frac{1}{2} \sum_{n=1}^N t_n \ln |\boldsymbol{\Sigma}| - \frac{1}{2} \sum_{n=1}^N t_n (\mathbf{x}_n - \boldsymbol{\mu}_1)^T \boldsymbol{\Sigma}^{-1} (\mathbf{x}_n - \boldsymbol{\mu}_1) \\ & - \frac{1}{2} \sum_{n=1}^N (1 - t_n) \ln |\boldsymbol{\Sigma}| - \frac{1}{2} \sum_{n=1}^N (1 - t_n) (\mathbf{x}_n - \boldsymbol{\mu}_2)^T \boldsymbol{\Sigma}^{-1} (\mathbf{x}_n - \boldsymbol{\mu}_2) \\ & = -\frac{N}{2} \ln |\boldsymbol{\Sigma}| - \frac{N}{2} \text{Tr}\{\boldsymbol{\Sigma}^{-1} \mathbf{S}\} \end{aligned} \quad (4.77)$$

Tr 矩阵的迹；主对角线元素的和=特征值的和。

其中我们已经定义了

$$\mathbf{S} = \frac{N_1}{N} \mathbf{S}_1 + \frac{N_2}{N} \mathbf{S}_2 \quad (4.78)$$

$$\mathbf{S}_1 = \frac{1}{N_1} \sum_{n \in \mathcal{C}_1} (\mathbf{x}_n - \boldsymbol{\mu}_1)(\mathbf{x}_n - \boldsymbol{\mu}_1)^T \quad (4.79)$$

$$\mathbf{S}_2 = \frac{1}{N_2} \sum_{n \in \mathcal{C}_2} (\mathbf{x}_n - \boldsymbol{\mu}_2)(\mathbf{x}_n - \boldsymbol{\mu}_2)^T \quad (4.80)$$

使用高斯分布的最大似然解的标准结果，我们看到 $\boldsymbol{\Sigma} = \mathbf{S}$ ，它表示对一个与两类都有关系的协方差矩阵求加权平均。

这个结果很容易推广到 K 类问题，得到参数的对应的最大似然解。其中我们假定每个类条件概率密度都是高斯分布，协方差矩阵相同。注意，拟合类高斯分布的方法对于离群点并不鲁棒，因为高斯的最大似然估计是不鲁棒的。

4.2.3 离散特征

现在让我们考虑离散特征值 x_i 的情形。为了简化起见，我们首先考察二元特征值 $x_i \in \{0, 1\}$ ，稍后会讨论如何推广到更一般的离散特征。如果有 D 个输入，那么一般的概率分布会对应于一个大小为 2^D 的表格，包含 $2^D - 1$ 个独立变量（由于要满足加和限制）。由于这会随着特征的数量指数增长，因此我们想寻找一个更加严格的表示方法。这里，我们做出朴素贝叶斯（naive Bayes）的假设，这个假设中，特征值被看成相互独立的，以类别 \mathcal{C}_k 为条件。因此我们得到类条件分布，形式为

$$p(\mathbf{x} | \mathcal{C}_k) = \prod_{i=1}^D \mu_{ki}^{x_i} (1 - \mu_{ki})^{1-x_i} \quad (4.81)$$

其中对于每个类别，都有 D 个独立的参数。代入公式 (4.63)，我们有

$$a_k(\mathbf{x}) = \sum_{i=1}^D \{x_i \ln \mu_{ki} + (1 - x_i) \ln(1 - \mu_{ki})\} + \ln p(\mathcal{C}_k) \quad (4.82)$$

与之前一样，这是输入变量 x_i 的线性函数。对于 $K = 2$ 个类别的情形，我们可以考虑另一种方法，即公式 (4.57) 给出的logistic sigmoid函数。离散变量也有类似的结果，其中，每个离散变量有 $M > 2$ 种状态。

4.2.4 指数族分布

正如我们已经看到的，无论是服从高斯分布的输入，还是离散的输入，后验类概率密度都是由一般的线性模型和logistic sigmoid ($K = 2$ 个类别) 或者softmax ($K \geq 2$ 个类别) 激活函数给出。通过假定类条件概率密度 $p(\mathbf{x} | \mathcal{C}_k)$ 是指数族分布的成员，我们可以看到上述结果都是更一般的结果的特例。

使用公式 (2.194) 给出的指数族分布的形式，我们可以看到 \mathbf{x} 的分布可以写成下面的形式

$$p(\mathbf{x} | \boldsymbol{\lambda}_k) = h(\mathbf{x})g(\boldsymbol{\lambda}_k) \exp\{\boldsymbol{\lambda}_k^T \mathbf{u}(\mathbf{x})\} \quad (4.83)$$

我们现在把注意力集中在 $\mathbf{u}(\mathbf{x}) = \mathbf{x}$ 这种分布上。然后，我们使用公式 (2.236) 引入一个缩放参数 s ，这样我们就得到了指数族类条件概率分布的一个子集

$$p(\mathbf{x} | \boldsymbol{\lambda}_k, s) = \frac{1}{s} h\left(\frac{1}{s}\mathbf{x}\right) g(\boldsymbol{\lambda}_k) \exp\left\{\frac{1}{s} \boldsymbol{\lambda}_k^T \mathbf{x}\right\} \quad (4.84)$$

注意我们让每个类别有自己的参数向量 $\boldsymbol{\lambda}_k$ ，但是我们假定各个类别有同样的缩放参数 s 。

对于二分类问题，我们把这个类条件概率密度的表达式代入公式 (4.58)，我们看到后验概率与之前一样是一个作用在线性函数 $a(\mathbf{x})$ 上的logistic sigmoid函数。 $a(\mathbf{x})$ 的形式为

$$a(\mathbf{x}) = \frac{1}{s} (\boldsymbol{\lambda}_1 - \boldsymbol{\lambda}_2)^T \mathbf{x} + \ln g(\boldsymbol{\lambda}_1) - \ln g(\boldsymbol{\lambda}_2) + \ln p(\mathcal{C}_1) - \ln p(\mathcal{C}_2) \quad (4.85)$$

类似地，对于 K 类问题，我们把类条件概率密度的表达式代入公式 (4.63)，得

$$a_k(\mathbf{x}) = \frac{1}{s} \boldsymbol{\lambda}_k^T \mathbf{x} + \ln g(\boldsymbol{\lambda}_k) + \ln p(\mathcal{C}_k) \quad (4.86)$$

这又是一个 \mathbf{x} 的线性函数。

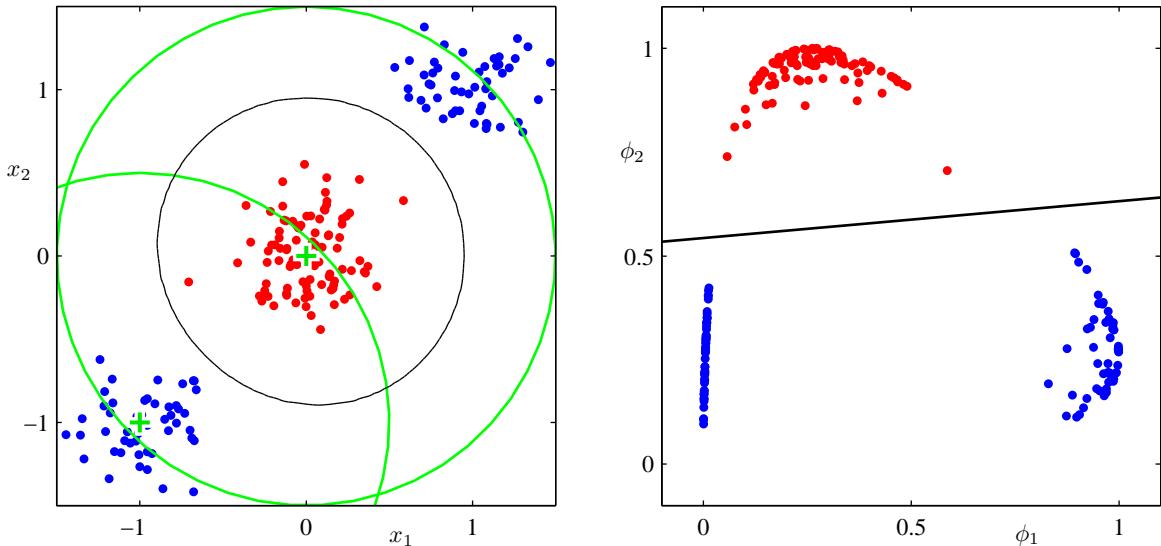


图 4.12: 线性分类模型的非线性基函数的作用的说明。做题给出了原始的输入空间 (x_1, x_2) 以及标记为红色和蓝色的数据点。这个空间中定义了两个“高斯”基函数 $\phi_1(x)$ 和 $\phi_2(x)$ ，中心用绿色十字表示，轮廓线用绿色圆形表示。右图给出了对应的特征空间 (ϕ_1, ϕ_2) 以及线性决策边界。决策边界由4.3.2节讨论的线性回归模型得到。对应的在原始空间中的非线性决策边界在左图中用黑色曲线标记出。

4.3 概率判别式模型

使用线性模型的函数形式，然后使用最大似然确定参数。

对于二分类问题，我们已经看到，对于一大类的类条件概率密度 $p(\mathbf{x} | C_k)$ 的选择，类别 C_1 后验概率分布可以写成作用于 \mathbf{x} 的线性函数上的logistic sigmoid函数的形式。类似地，对于多分类的情形，类别 C_k 的后验概率由 \mathbf{x} 的线性函数的softmax变换给出。对于类条件概率密度 $p(\mathbf{x} | C_k)$ 的具体的选择，我们已经使用了最大似然方法估计了概率密度的参数以及类别先验 $p(C_k)$ ，然后使用贝叶斯定理就可以求出后验类概率。

然而，另一种方法是显示地使用一般的线性模型的函数形式，然后使用最大似然法直接确定它的参数。我们会看到，寻找这样的解有一个高效的算法，被称为迭代重加权最小平方（iterative reweighted least squares），或者简称IRLS。

寻找一般的线性模型参数的间接方法是，分别寻找类条件概率密度和类别先验，然后使用贝叶斯定理。这是生成式建模的一个例子。这是因为，我们可以拿来这个模型，从边缘分布 $p(\mathbf{x})$ 中取出一个 \mathbf{x} 的值，然后人工生成数据。在直接方法中，我们最大化由条件概率分布 $p(C_k | \mathbf{x})$ 定义的似然函数。这种方法代表了判别式训练的一种形式。判别式方法的一个优点是通常有更少的可调节参数需要确定，正如我们稍后会看到的那样。并且预测表现也会提升，尤其是当类条件概率密度的假设没有很好地近似真实分布的时候更是如此。

概率判别式模型的优势

4.3.1 固定基函数

本章中目前为止，我们已经考虑了直接对输入向量 (\mathbf{x}) 进行分类的分类模型。然而，如果我们首先使用一个基函数向量 $\phi(\mathbf{x})$ 对输入变量进行一个固定的非线性变换，所有的这些算法仍然同样适用。最终的决策边界在特征空间 ϕ 中是线性的，因此对应于原始 \mathbf{x} 空间中的非线性决策边界，如图4.12所示。在特征空间 $\phi(\mathbf{x})$ 线性可分的类别未必在原始的观测空间 \mathbf{x} 中线性可分。与我们对于回归的线性模型的讨论一样，基函数中的某一个通常设置为常数，例如 $\phi_0(\mathbf{x}) = 1$ ，使得对应的参数 w_0 扮演偏置的作用。对于本章的剩余部分，我们会使用一个固定基函数变换 $\phi(\mathbf{x})$ ，因为这会引出一些与第3章中讨论的回归模型相似的地方。

对于许多实际问题来说，类条件概率密度 $p(\mathbf{x} | C_k)$ 之间有着相当大的重叠。这表明至少对于某些 \mathbf{x} 的值，后验概率 $p(C_k | \mathbf{x})$ 不等于0或1。在这种情况下，最优解可以通过下面的方式获得：对后验概率精确建模，然后使用第1章中讨论的标准的决策论。需要注意的是，非线性变换 $\phi(\mathbf{x})$ 不会消除这些重叠。实际上，这些变换会增加重叠的程度，或者在原始观测空间中不存在重叠的地方产生出新的重叠。然而，恰当地选择非线性变换能够让后验概率的建模过程更简

单。

这样的固定基函数模型有着重要的局限性，这些局限性在后续的章节中会被解决，解决方法为允许基函数自身根据数据进行调节。尽管有这些限制，固定基函数模型在实际应用中起着重要的作用。关于这个模型的讨论会引出许多重要的概念，这些概念对于理解更复杂的模型很必要。

4.3.2 logistic回归

我们首先通过二分类问题开始我们对于一般线性模型方法的讨论。在4.2节我们对于生成式方法的讨论中，我们看到在一些相当一般的假设条件下，类别 \mathcal{C}_1 的后验概率可以写成作用在特征向量 ϕ 的线性函数上的logistic sigmoid函数的形式，即

$$p(\mathcal{C}_1 | \phi) = y(\phi) = \sigma(\mathbf{w}^T \phi) \quad \boxed{\text{把概率表示成函数的形式，通过似然确定函数参数。}} \quad (4.87)$$

且 $p(\mathcal{C}_2 | \phi) = 1 - p(\mathcal{C}_1 | \phi)$ 。这里 $\sigma(\cdot)$ 是公式 (4.59) 定义的logistic sigmoid函数。使用统计学的术语，这个模型被称为logistic回归，虽然应该强调的一点是，这是一个分类模型而不是回归模型。

对于一个 M 维特征空间 ϕ ，这个模型有 M 个可调节参数。相反，如果我们使用最大似然方法调节了高斯类条件概率密度，那么我们有 $2M$ 个参数来描述均值，以及 $\frac{M(M+1)}{2}$ 个参数来描述（共享的）协方差矩阵。算上类先验 $p(\mathcal{C}_1)$ ，参数的总数为 $\frac{M(M+5)}{2} + 1$ ，这随着 M 的增长而以二次的方式增长。这和logistic回归方法中对于参数数量 M 的线性依赖不同。对于大的 M 值，直接使用logistic回归模型有着很明显的优势。

我们现在使用最大似然方法来确定logistic回归模型的参数。为了完成这一点，我们要使用logistic sigmoid函数的导数，它可以很方便地使用sigmoid函数本身表示如下

$$\frac{d\sigma}{da} = \sigma(1 - \sigma) \quad (4.88)$$

对于一个数据集 ϕ_n, t_n ，其中 $t_n \in \{0, 1\}$ 且 $\phi_n = \phi(\mathbf{x}_n)$ ，并且 $n = 1, \dots, N$ ，似然函数可以写成

$$p(\mathbf{t} | \mathbf{w}) = \prod_{n=1}^N y_n^{t_n} \{1 - y_n\}^{1-t_n} \quad (4.89)$$

其中 $\mathbf{t} = (t_1, \dots, t_N)^T$ 且 $y_n = p(\mathcal{C}_1 | \phi_n)$ 。与之前一样，我们可以通过取似然函数的负对数的方式，定义一个误差函数。这种方式产生了交叉熵(cross-entropy) 误差函数，形式为

$$E(\mathbf{w}) = -\ln p(\mathbf{t} | \mathbf{w}) = -\sum_{n=1}^N \{t_n \ln y_n + (1 - t_n) \ln(1 - y_n)\} \quad (4.90)$$

其中 $y_n = \sigma(a_n)$ 且 $a_n = \mathbf{w}^T \phi_n$ 。两侧关于 \mathbf{w} 取误差函数的梯度，我们有

$$\nabla E(\mathbf{w}) = \sum_{n=1}^N (y_n - t_n) \phi_n \quad \boxed{\text{(目标值-预测值) * 基函数向量}\Phi} \quad (4.91)$$

推导时我们使用了公式 (4.88)。我们看到，涉及到logistic sigmoid的导数的因子已经被消去，使得对数似然函数的梯度的形式十分简单。特别地，数据点 n 对梯度的贡献为目标值和模型预测值之间的“误差” $y_n - t_n$ 与基函数向量 ϕ_n 相乘。此外，与公式 (3.13) 的对比表明，它的函数形式与线性回归模型中的平方和误差函数的梯度的函数形式完全相同。

如果必要的话，我们可以使用公式 (4.91) 的结果提出一个顺序算法，这种算法中，每次只出现一个模式，权向量使用公式 (3.22) 更新，其中 ∇E_n 是公式 (4.91) 的第 n 项。

值得注意的一点是，最大似然方法对于线性可分的数据集会产生严重的过拟合现象。这是由于最大似然解出现在超平面对应于 $\sigma = 0.5$ 的情况，它等价于 $\mathbf{w}^T \phi = 0$ 。最大似然解把数据集分成了两类，并且 \mathbf{w} 的大小趋向于无穷大。这种情况下，logistic sigmoid函数在特征空间中变得非

常陡峭，对应于一个跳变的阶梯函数，使得每一个来自类别 k 的训练数据都被赋予一个后验概率 $p(\mathcal{C}_k | \mathbf{x}) = 1$ 。此外，通常这些解之间存在连续性，因为任何切分超平面都会造成训练数据点中同样的后验概率，正如后面在图10.13中将会看到的那样。最大似然方法无法区分某个解优于另一个解，并且在实际应用中哪个解被找到将会依赖于优化算法的选择和参数的初始化。注意，即使与模型的参数相比数据点的数量很多，只要数据是线性可分的，这个问题就会出现。通过引入先验概率，然后寻找 \mathbf{w} 的MAP解，或者等价地，通过给误差函数增加一个正则化项，这种奇异性就可以被避免。

4.3.3 迭代重加权最小平方

求最小误差

在第3章讨论线性回归模型的时候，在高斯噪声模型的假设的情况下，最大似然解有解析解。这是因为对数似然函数为参数向量 \mathbf{w} 的二次函数。对于logistic回归来说，不再有解析解了，因为logistic sigmoid函数是一个非线性函数。然而，函数形式不是二次函数并不是本质的原因。精确地说，正如我们将要看到的那样，误差函数是凸函数，因此有一个唯一的最小值。此外，误差函数可以通过一种高效的迭代方法求出最小值，这种迭代方法基于Newton-Raphson迭代最优化框架，使用了对数似然函数的局部二次近似。为了最小化函数 $E(\mathbf{w})$ ，Newton-Raphson对权值的更新的形式为 (Fletcher, 1987; Bishop and Nabney, 2008)

$$\mathbf{w}^{\text{新}} = \mathbf{w}^{\text{旧}} - \mathbf{H}^{-1} \nabla E(\mathbf{w}) \quad (4.92)$$

其中 \mathbf{H} 是一个Hessian矩阵，它的元素由 $E(\mathbf{w})$ 关于 \mathbf{w} 的二阶导数组成。

首先，让我们把Newton-Raphson方法应用到现行回归模型 (3.3) 上，误差函数为平方和误差函数 (3.12)。这个误差函数的梯度和Hessian矩阵为

$$\nabla E(\mathbf{w}) = \sum_{n=1}^N (\mathbf{w}^T \phi_n - t_n) \phi_n = \Phi^T \Phi \mathbf{w} - \Phi^T \mathbf{t} \quad (4.93)$$

$$\mathbf{H} = \nabla \nabla E(\mathbf{w}) = \sum_{n=1}^N \phi_n \phi_n^T = \Phi^T \Phi \quad (4.94)$$

其中 Φ 是 $N \times M$ 设计矩阵，第 n 行为 ϕ_n^T 。于是，Newton-Raphson更新的形式为

$$\begin{aligned} \mathbf{w}^{\text{新}} &= \mathbf{w}^{\text{旧}} - (\Phi^T \Phi)^{-1} \{ \Phi^T \Phi \mathbf{w}^{\text{旧}} - \Phi^T \mathbf{t} \} \\ &= (\Phi^T \Phi)^{-1} \Phi^T \mathbf{t} \end{aligned} \quad (4.95)$$

我们看到这是标准的最小平方解。注意，这种情况下误差函数是二次的，因此Newton-Raphson公式用1步就给出了精确解。

现在让我们把Newton-Raphson更新应用到logistic回归模型的交叉熵误差函数 (4.90) 上。根据公式 (4.91)，我们看到这个误差函数的梯度和Hessian矩阵为

$$\nabla E(\mathbf{w}) = \sum_{n=1}^N (y_n - t_n) \phi_n = \Phi^T (\mathbf{y} - \mathbf{t}) \quad (4.96)$$

$$\mathbf{H} = \nabla \nabla E(\mathbf{w}) = \sum_{n=1}^N y_n(1-y_n) \phi_n \phi_n^T = \Phi^T \mathbf{R} \Phi \quad (4.97)$$

推导过程中我们使用了公式 (4.88)。并且，我们引入了一个 $N \times N$ 的对角矩阵 \mathbf{R} ，元素为

$$R_{nn} = y_n(1-y_n) \quad (4.98)$$

我们看到Hessian矩阵不再是常量，而是通过权矩阵 \mathbf{R} 依赖于 \mathbf{w} 。这对应于误差函数不是二次函数的事实。使用性质 $0 < y_n < 1$ （这个性质来自于logistic sigmoid函数形式），我们看到对于任

意向量 \mathbf{u} 都有 $\mathbf{u}^T \mathbf{H} \mathbf{u} > 0$, 因此 Hessian 矩阵 \mathbf{H} 是正定的。因此误差函数是 \mathbf{w} 的一个凸函数, 从而有唯一的最小值。

这样, logistic 回归模型的 Newton-Raphson 更新公式就变成了

$$\begin{aligned}\mathbf{w}^{\text{新}} &= \mathbf{w}^{\text{旧}} - (\Phi^T \mathbf{R} \Phi)^{-1} \Phi^T (\mathbf{y} - \mathbf{t}) \\ &= (\Phi^T \mathbf{R} \Phi)^{-1} \{ \Phi^T \mathbf{R} \Phi \mathbf{w}^{\text{旧}} - \Phi^T (\mathbf{y} - \mathbf{t}) \} \\ &= (\Phi^T \mathbf{R} \Phi)^{-1} \Phi^T \mathbf{R} \mathbf{z}\end{aligned}\quad (4.99)$$

其中 \mathbf{z} 是一个 N 维向量, 元素为

$$\mathbf{z} = \Phi \mathbf{w}^{\text{旧}} - \mathbf{R}^{-1} (\mathbf{y} - \mathbf{t}) \quad (4.100)$$

我们看到更新公式 (4.99) 的形式为一组加权最小平方问题的规范方程。由于权矩阵 \mathbf{R} 不是常量, 而是依赖于参数向量 \mathbf{w} , 因此我们必须迭代地应用规范方程, 每次使用新的权向量 \mathbf{w} 计算一个修正的权矩阵 \mathbf{R} 。由于这个原因, 这个算法被称为迭代重加权最小平方 (iterative reweighted least squares), 或者简称为 IRLS (Rubin, 1983)。与加权的最小平方问题一样, 对角矩阵 \mathbf{R} 可以看成方差, 因为 logistic 回归模型的 t 的均值和方差为

$$\mathbb{E}[t] = \sigma(\mathbf{x}) = y \quad (4.101)$$

$$\text{var}[t] = \mathbb{E}[t^2] - \mathbb{E}[t]^2 = \sigma(\mathbf{x}) - \sigma(\mathbf{x})^2 = y(1-y) \quad (4.102)$$

其中我们使用了 $t \in \{0, 1\}$ 时 $t^2 = t$ 的性质。事实上, 我们可以把 IRLS 看成变量空间 $a = \mathbf{w}^T \phi$ 的线性问题的解。这样, \mathbf{z} 的第 n 个元素 z_n 就可以简单地看成这个空间中的有效的目标值。 z_n 可以通过对当前操作点 $w^{\text{旧}}$ 附近的 logistic sigmoid 函数的局部线性近似的方式得到。

$$\begin{aligned}a_n(\mathbf{w}) &\simeq a_n(\mathbf{w}^{\text{旧}}) + \left. \frac{da_n}{dy_n} \right|_{\mathbf{w}^{\text{旧}}} (t_n - y_n) \\ &= \phi_n^T \mathbf{w}^{\text{旧}} - \frac{y_n - t_n}{y_n(1-y_n)} = z_n\end{aligned}\quad (4.103)$$

4.3.4 多类 logistic 回归

在我们对于多分类的生成式模型的讨论中, 我们已经看到对于一大类概率分布来说, 后验概率由特征变量的线性函数的 softmax 变换给出, 即

$$p(C_k | \phi) = y_k(\phi) = \frac{\exp(a_k)}{\sum_j \exp(a_k)} \quad (4.104)$$

其中, “激活” a_k 为

$$a_k = \mathbf{w}_k^T \phi \quad (4.105)$$

那里, 我们使用了最大似然方法分别估计类条件概率密度和类先验概率, 然后使用贝叶斯定理找到对应的后验概率, 因此隐式地确定了参数 $\{\mathbf{w}_k\}$ 。这里, 我们考虑使用最大似然方法直接确定这个模型中的参数 $\{\mathbf{w}_k\}$ 。为了完成这一点, 我们需要求出 y_k 关于所有激活 a_j 的导数。这些导数为

$$\frac{\partial y_k}{\partial a_j} = y_k(I_{kj} - y_j) \quad \boxed{\text{与 sigmoid 函数比较: } \sigma(1-\sigma)} \quad (4.106)$$

其中 I_{kj} 为单位矩阵的元素。

接下来我们写出似然函数。最容易的方法是, 使用“1-of- K ”表达方式。这种表达方式中, 属于类别 C_k 的特征向量 ϕ_k 的目标向量 \mathbf{t}_n 是一个二元向量, 这个向量的第 k 个元素等于 1, 其余元素都等于 0。从而, 似然函数为

$$p(\mathbf{T} | \mathbf{w}_1, \dots, \mathbf{w}_K) = \prod_{n=1}^N \prod_{k=1}^K p(C_k | \phi_n)^{t_{nk}} = \prod_{n=1}^N \prod_{k=1}^K y_n^{t_{nk}} \quad (4.107)$$

输入第 N 组变量, 属于第 k 类

其中 $y_{nk} = y_k(\phi_n)$, \mathbf{T} 是目标变量的一个 $N \times K$ 的矩阵, 元素为 t_{nk} 。取负对数, 可得

$$E(\mathbf{w}_1, \dots, \mathbf{w}_K) = -\ln p(\mathbf{T} | \mathbf{w}_1, \dots, \mathbf{w}_K) = -\sum_{n=1}^N \sum_{k=1}^K t_{nk} \ln y_{nk} \quad (4.108)$$

它被称为多分类问题的交叉熵 (cross-entropy) 误差函数。

我们现在取误差函数关于参数向量 \mathbf{w}_j 的梯度。使用公式 (4.106) 给出的 softmax 函数的导数的结果, 我们有

$$\nabla_{\mathbf{w}_j} E(\mathbf{w}_1, \dots, \mathbf{w}_K) = \sum_{n=1}^N (y_{nj} - t_{nj}) \phi_n \quad (4.109)$$

其中我们使用了 $\sum_k t_{nk} = 1$ 。我们又一次看到了梯度的这种函数形式, 即误差 $(y_{nj} - t_{nj})$ 与基函数 ϕ_n 的乘积。这种梯度形式在线性模型的平方和误差函数以及 logistic 回归模型的误差函数中都出现过。和之前一样, 我们可以将这个公式用于顺序算法。这种顺序算法中每次只出现一个模式, 每个权向量都使用公式 (3.22) 更新。

我们已经看到, 对于数据点 n , 线性回归模型的对数似然函数关于参数向量 \mathbf{w} 的导数的形式为“误差” $y_n - t_n$ 乘以特征向量 ϕ_n 。类似地, 对于 logistic sigmoid 激活函数与交叉熵误差函数 (4.90) 的组合, 以及多类交叉熵误差函数 (4.108) 的 softmax 激活函数, 我们又一次得到了相同的函数形式。这是一个更一般的结果的特例, 正如我们将在 4.3.6 节中将看到的那样。

为了找到一个批处理算法, 我们再次使用 Newton-Raphson 更新来获得多类问题的对应的 IRLS 算法。这需要求出由大小为 $M \times M$ 的块组成的 Hessian 矩阵, 其中块 i, j 为

$$\nabla_{\mathbf{w}_i} \nabla_{\mathbf{w}_j} E(\mathbf{w}_1, \dots, \mathbf{w}_K) = \sum_{n=1}^N y_{nk} (I_{kj} - y_{nj}) \phi_n \phi_n^T \quad (4.110)$$

与二分类问题一样, 多类 logistic 回归模型的 Hessian 矩阵是正定的, 因此误差函数有唯一的最小值。多类问题的 IRLS 的细节可以参考 Bishop and Nabney (2008)。

4.3.5 probit 回归

我们已经看到, 对于由指数族分布描述的一大类的类条件概率分布, 最终求出的后验类概率为作用在特征变量的线性函数上的 logistic (或者 softmax) 变换。然而, 不是所有的类条件概率密度都有这样简单的后验概率函数形式 (例如, 如果类条件概率密度由高斯混合模型建模)。这表明研究其他类型的判别式概率模型可能会很有价值。但是本章中, 我们将会回到二分类的情形, 再次使用一般的线性模型的框架, 即

$$p(t = 1 | a) = f(a) \quad (4.111)$$

其中 $a = \mathbf{w}^T \phi$, 且 $f(\cdot)$ 为激活函数。

我们选择其他的链接函数的原因可以通过噪声阈值模型看出来, 如下所述。对于每个输入 ϕ_n , 我们计算 $a_n = \mathbf{w}^T \phi_n$, 然后按照下面的方式设置目标值

$$\begin{cases} t_n = 1, & \text{如果 } a_n \geq \theta \\ t_n = 0, & \text{其他情况} \end{cases} \quad (4.112)$$

如果 θ 的值从概率密度 $p(\theta)$ 中抽取, 那么对应的激活函数由累积分布函数给出

$$f(a) = \int_{-\infty}^a p(\theta) d\theta \quad (4.113)$$

如图 4.13 所示。

作为一个具体的例子, 假设概率密度 $p(\theta)$ 是零均值、单位方差的高斯概率密度。对应的累积分布函数为

$$\Phi(a) = \int_{-\infty}^a \mathcal{N}(\theta | 0, 1) d\theta \quad (4.114)$$

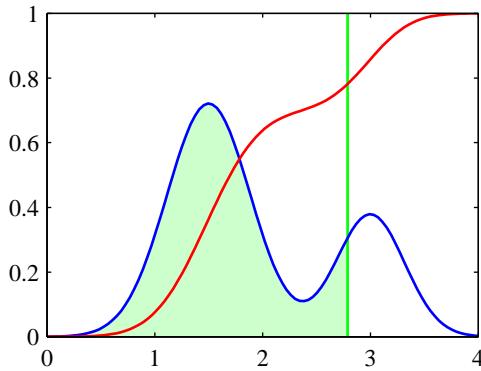


图 4.13: 概率分布 $p(\theta)$ 的图形表示, 这个概率分布用蓝色曲线标记出。这个例子中, 这个分布由两个高斯分布混合而成。同时给出的还有它的累积密度函数 $f(a)$, 用红色曲线表示。注意, 蓝色曲线上任意一点, 例如垂直绿色直线标记出的点, 对应于红色曲线在相同点处的斜率。相反, 红色曲线在这点上的值对应于蓝色曲线下方的绿色阴影的面积。在随机阈值模型中, 如果 $a = \mathbf{w}^T \phi$ 的值超过某个阈值, 则类别标签的取值为 $t = 1$, 否则它的取值为 $t = 0$ 。这等价于由累积密度函数 $f(a)$ 给出的激活函数。

这被称为逆probit (inverse probit) 函数。它的形状为sigmoid形, 并且在图 4.9 中与 logistic sigmoid 函数进行了对比。注意, 使用更一般的高斯分布不会改变模型, 因为这样做等价于对线性系数 \mathbf{w} 的重新缩放。许多用于计算这个函数的数值计算包都与下面的这个函数紧密相关

$$\text{erf}(a) = \frac{2}{\sqrt{\pi}} \int_0^a \exp(-\theta^2) d\theta \quad (4.115)$$

它被称为 erf 函数或者被称为 error 函数 (不要与机器学习模型中的误差函数相混淆)。它与逆probit函数的关系为

$$\Phi(a) = \frac{1}{2} \left\{ 1 + \text{erf}\left(\frac{a}{\sqrt{2}}\right) \right\} \quad (4.116)$$

基于 probit 激活函数的一般的线性模型被称为 probit 回归。

我们可以使用最大似然法来确定模型的参数, 这是之前讨论的思想的一个直接推广。在实际应用中, 使用 probit 回归得到的结果倾向于与 logistic 回归得到的结果类似。但是, 当我们在 4.5 节讨论 logistic 回归的贝叶斯观点时, 我们会找到 probit 模型的另一个应用。

在实际应用中经常出现的一个问题是离群点, 它可能由输入向量 \mathbf{x} 的测量误差产生, 或者由目标值 t 的错误标记产生。由于这些点可以位于错误的一侧中距离理想决策边界相当远的位置上, 因此他们会严重地干扰分类器。注意, 在这一点上, logistic 回归模型与 probit 回归模型的表现不同, 因为对于 $x \rightarrow \infty$, logistic sigmoid 函数像 $\exp(-x)$ 那样渐进地衰减, 而 probit 激活函数像 $\exp(-x^2)$ 那样衰减, 因此 probit 模型对于离群点会更加敏感。

然而, logistic 模型和 probit 模型都假设数据点被正确标记了。错误标记的影响可以很容易地合并到概率模型中。我们引入一个概率 ϵ , 它是目标值 t 被翻转到错误值的概率 (Opper and Winther, 2000a)。这时, 数据点 \mathbf{x} 的目标值的分布为

$$\begin{aligned} p(t | \mathbf{x}) &= (1 - \epsilon)\sigma(\mathbf{x}) + \epsilon(1 - \sigma(\mathbf{x})) \\ &= \epsilon + (1 - 2\epsilon)\sigma(\mathbf{x}) \end{aligned} \quad (4.117)$$

其中 $\sigma(\mathbf{x})$ 是输入向量 \mathbf{x} 的激活函数。这里, ϵ 可以事先设定, 也可以被当成超参数, 然后从数据中推断它的值。

4.3.6 标准链接函数

对于高斯噪声分布的线性回归模型, 误差函数, 对应于负对数似然函数, 由公式 (3.12) 给出。如果我们对数据点 n 对误差函数的贡献关于参数向量 \mathbf{w} 求导数, 那么导数的形式为“误

差” $y_n - t_n$ 与特征向量 ϕ_n 的乘积，其中 $y_n = \mathbf{w}^T \phi_n$ 。类似地，对于logistic sigmoid激活函数与交叉熵误差函数（4.90）的组合，以及多类交叉熵误差函数（4.108）的softmax激活函数，我们再次得到了同样的简单形式。现在我们证明，如果假设目标变量的条件分布来自于指数族分布，对应的激活函数选为标准链接函数（canonical link function），那么这个结果是一个一般的结果。

我们再次使用指数族分布的限制形式（4.84）。注意，这里我们把指数族分布的假设应用于目标变量 t ，而不是4.2.4节中应用于输入向量 \mathbf{x} 。于是，我们考虑目标变量的条件分布

$$p(t | \eta, s) = \frac{1}{s} h\left(\frac{t}{s}\right) g(\eta) \exp\left\{\frac{\eta t}{s}\right\} \quad (4.118)$$

使用与推导结果（2.226）时相同的过程，我们看到 t 的条件均值（记作 y ）为

$$y \equiv \mathbb{E}[t | \eta] = -s \frac{d}{d\eta} \ln g(\eta) \quad (4.119)$$

因此 y 和 η 一定相关，我们把这个关系记作 $\eta = \psi(y)$ 。

按照Nelder and Wedderburn (1972) 的方法，我们将一般线性模型（generalised linear model）定义为这样的模型： y 是输入变量（或者特征变量）的线性组合的非线性函数，即

$$y = f(\mathbf{w}^T \phi) \quad (4.120)$$

其中 $f(\cdot)$ 在机器学习的文献中被称为激活函数（activation function）， $f^{-1}(\cdot)$ 在统计学中被称为链接函数（link function）。

现在考虑这个模型的对数似然函数。它是 η 的一个函数，形式为

$$\ln p(\mathbf{t} | \eta, s) = \sum_{n=1}^N \ln p(t_n | \eta, s) = \sum_{n=1}^N \left\{ \ln g(\eta_n) + \frac{\eta_n t_n}{s} \right\} + \text{常数} \quad (4.121)$$

其中我们假定所有的观测有一个相同的缩放参数（它对应着例如服从高斯分布的噪声的方差），因此 s 与 n 无关。对数似然函数关于模型参数 \mathbf{w} 的导数为

$$\begin{aligned} \nabla_{\mathbf{w}} \ln p(\mathbf{t} | \eta, s) &= \sum_{n=1}^N \left\{ \frac{d}{d\eta_n} \ln g(\eta_n) + \frac{t_n}{s} \right\} \frac{d\eta_n}{dy_n} \frac{dy_n}{da_n} \nabla a_n \\ &= \sum_{n=1}^N \frac{1}{s} \{t_n - y_n\} \psi'(y_n) f'(a_n) \phi_n \end{aligned} \quad (4.122)$$

其中 $a_n = \mathbf{w}^T \phi_n$ ，并且我们使用了 $y_n = f(a_n)$ 以及公式（4.119）给出的 $\mathbb{E}[t | \eta]$ 的结果。我们现在看到，如果我们为链接函数 $f^{-1}(y)$ 选成下面的形式，那么表达式会得到极大的简化。

$$f^{-1}(y) = \psi(y) \quad (4.123)$$

上式表明 $f(\psi(y)) = y$ ，因此 $f'(\psi)\psi'(y) = 1$ 。并且，由于 $a = f^{-1}(y)$ ，我们有 $a = \psi$ ，因此 $f'(a)\psi'(y) = 1$ 。在这种情况下，误差函数的梯度可以化简为

$$\nabla E(\mathbf{w}) = \frac{1}{s} \sum_{n=1}^N \{y_n - t_n\} \phi_n \quad (4.124)$$

对于高斯分布， $s = \beta^{-1}$ ，而对于logistic模型， $s = 1$ 。

4.4 拉普拉斯近似

拉普拉斯近似用高斯分布去近似，中心位于众数的位置，考虑的是某个局部的性质，不是全局的性质。

在4.5节，我们会讨论logistic回归的贝叶斯观点。正如我们将看到的那样，这比3.3节和3.5节讨论的线性回归模型的贝叶斯观点更加复杂。特别地，我们不能够精确地关于参数向量 \boldsymbol{x} 求积分，因为后验概率分布不再是高斯分布。因此，有必要介绍某种形式的近似。稍后在本书中，我们会介绍一系列基于分析估计和数值采样的技术。variable和sampling

这里我们介绍一个简单的但是广泛使用的框架，被称为拉普拉斯近似。它的目标是找到定义在一组连续变量上的概率密度的高斯近似。首先考虑单一连续变量 z 的情形，假设分布 $p(z)$ 的定义为

$$p(z) = \frac{1}{Z} f(z) \quad (4.125)$$

其中 $Z = \int f(z) dz$ 是归一化系数。我们假定 Z 的值是未知的。在拉普拉斯方法中，目标是寻找一个高斯近似 $q(z)$ ，它的中心位于 $p(z)$ 的众数的位置。第一步是寻找 $p(z)$ 的众数，即寻找一个点 z_0 使得 $p'(z_0) = 0$ ，或者等价地

$$\left. \frac{df(z)}{dz} \right|_{z=z_0} = 0 \quad (4.126)$$

高斯分布有一个性质，即它的对数是变量的二次函数。于是我们考虑 $\ln f(z)$ 以众数 z_0 为中心的泰勒展开，即

$$\ln f(z) \simeq \ln f(z_0) - \frac{1}{2} A(z - z_0)^2 \quad (4.127)$$

其中

$$A = -\left. \frac{d^2}{dz^2} \ln f(z) \right|_{z=z_0} \quad (4.128)$$

注意，泰勒展开式中的一阶项没有出现，因为 z_0 是概率分布的局部最大值。两侧同时取指数，我们有

$$f(z) \simeq f(z_0) \exp \left\{ -\frac{A}{2}(z - z_0)^2 \right\} \quad (4.129)$$

这样，使用归一化的高斯分布的标准形式，我们就可以得到归一化的概率分布 $q(z)$ ，即

$$q(z) = \left(\frac{A}{2\pi} \right)^{\frac{1}{2}} \exp \left\{ -\frac{A}{2}(z - z_0)^2 \right\} \quad \text{找到一个高斯分布，中心在众数的位置。} \quad (4.130)$$

图4.14给出了拉普拉斯近似的说明。注意，高斯近似只在精度 $A > 0$ 时有良好的定义，换句话说，驻点 z_0 一定是一个局部最大值，使得 $f(z)$ 在驻点 z_0 处的二阶导数为负。

我们可以将拉普拉斯方法推广，去近似定义在 M 维空间 \boldsymbol{z} 上的概率分布 $p(\boldsymbol{z}) = \frac{f(\boldsymbol{z})}{Z}$ 。在驻点 \boldsymbol{z}_0 处，梯度 $\nabla f(\boldsymbol{z})$ 将会消失。在驻点处展开，我们有

$$\ln f(\boldsymbol{z}) \simeq \ln f(\boldsymbol{z}_0) - \frac{1}{2} (\boldsymbol{z} - \boldsymbol{z}_0)^T \boldsymbol{A} (\boldsymbol{z} - \boldsymbol{z}_0) \quad (4.131)$$

其中 $M \times M$ 的Hessian矩阵 \boldsymbol{A} 的定义为

$$\boldsymbol{A} = -\nabla \nabla \ln f(\boldsymbol{z})|_{\boldsymbol{z}=\boldsymbol{z}_0} \quad (4.132)$$

其中 ∇ 为梯度算子。两边同时取指数，我们有

$$f(\boldsymbol{z}) \simeq f(\boldsymbol{z}_0) \exp \left\{ -\frac{1}{2} (\boldsymbol{z} - \boldsymbol{z}_0)^T \boldsymbol{A} (\boldsymbol{z} - \boldsymbol{z}_0) \right\} \quad (4.133)$$

分布 $q(\boldsymbol{z})$ 正比于 $f(\boldsymbol{z})$ ，归一化系数可以通过观察归一化的多元高斯分布的标准形式（2.43）得到。因此

$$q(\boldsymbol{z}) = \frac{|\boldsymbol{A}|^{\frac{1}{2}}}{(2\pi)^{\frac{M}{2}}} \exp \left\{ -\frac{1}{2} (\boldsymbol{z} - \boldsymbol{z}_0)^T \boldsymbol{A} (\boldsymbol{z} - \boldsymbol{z}_0) \right\} = \mathcal{N}(\boldsymbol{z} | \boldsymbol{z}_0, \boldsymbol{A}^{-1}) \quad (4.134)$$

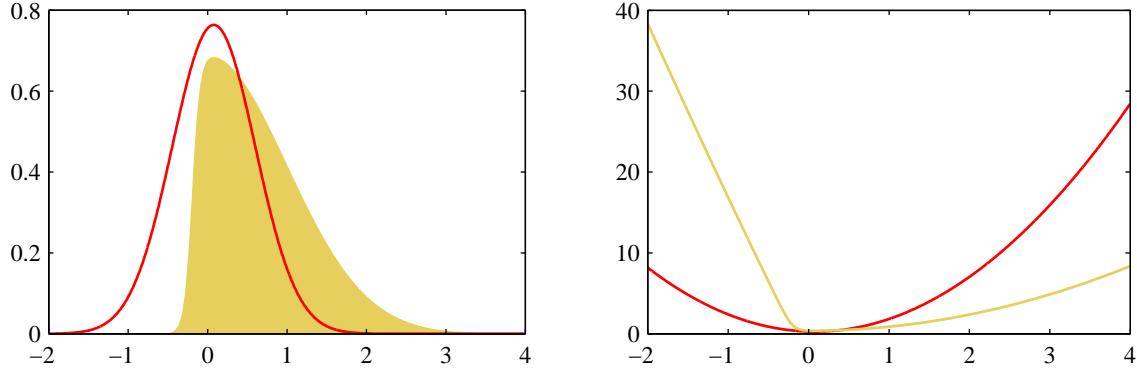


图 4.14: 应用于概率分布 $p(z) \propto \exp(-z^2/2)\sigma(20z + 4)$ 的拉普拉斯近似的例子, 其中 $\sigma(z)$ 是 logistic sigmoid 函数, 定义为 $\sigma(z) = (1 + e^{-z})^{-1}$ 。左图给出了归一化的概率分布 $p(z)$, 用黄色表示。同时给出了以 $p(z)$ 的众数 z_0 为中心的拉普拉斯近似, 用红色表示。右图给出了对应的曲线的负对数。

其中 $|\mathbf{A}|$ 是 \mathbf{A} 的行列式。这个高斯分布有良好定义的前提是, 精度矩阵 \mathbf{A} 是正定的, 这表明驻点 z_0 一定是一个局部最大值, 而不是一个最小值或者鞍点。

为了应用拉普拉斯近似, 我们首先需要寻找众数 z_0 , 然后计算在那个众数位置上的Hessian矩阵。在实际应用当中, 众数通常可以通过运行某种形式的数值最优化算法得到 (Bishop and Nabney, 2008)。许多在实际应用中遇到的概率分布都是多峰的, 因此根据考虑的峰值 (众数) 的不同, 会有不同的拉普拉斯近似。注意, 在应用拉普拉斯方法时, 真实概率分布的归一化常数 Z 不必事先知道。根据中心极限定理, 我们可以预见模型的后验概率会随着观测数据点的增多而越来越近似于高斯分布, 因此我们可以预见在数据点相对较多的情况下, 拉普拉斯近似会更有用。

拉普拉斯近似的一个主要缺点是, 由于它是以高斯分布为基础的, 因此它只能直接应用于实值变量。在其他情况下, 可以将拉普拉斯近似应用于变换之后的变量上。例如, 如果 $0 \leq \tau \infty$, 那么我们可以考虑 $\ln \tau$ 的拉普拉斯近似。但是, 拉普拉斯框架的最严重的局限性是, 它完全依赖于真实概率分布在变量的某个具体值位置上的性质, 因此会无法描述一些重要的全局属性。在第10章, 我们会考虑其他的方法, 这种方法从一个更加全局的角度考察了这个问题。

4.4.1 模型比较和BIC

除了近似概率分布 $p(z)$, 我们也可以获得对归一化常数 Z 的一个近似。使用公式 (4.133) 给出的近似, 我们有

$$\begin{aligned} Z &= \int f(z) dz \\ &\simeq f(z_0) \int \exp \left\{ -\frac{1}{2}(z - z_0)^T \mathbf{A}(z - z_0) \right\} dz \\ &= f(z_0) \frac{(2\pi)^{\frac{M}{2}}}{|\mathbf{A}|^{\frac{1}{2}}} \end{aligned} \tag{4.135}$$

推导过程中, 我们注意到了被积函数是高斯形式的, 并且使用了公式 (2.43) 给出的归一化高斯分布的标准结果。我们可以使用公式 (4.135) 的结果来获得对于模型证据的一个近似。正如3.4节讨论的那样, 模型证据在贝叶斯模型比较中起着相当重要的作用。

考虑一个数据集 \mathcal{D} 以及一组模型 $\{\mathcal{M}_i\}$, 模型参数为 $\{\boldsymbol{\theta}_i\}$ 。对于每个模型, 我们定义一个似然函数 $p(\mathcal{D} | \boldsymbol{\theta}_i, \mathcal{M}_i)$ 。如果我们引入一个参数的先验概率 $p(\boldsymbol{\theta}_i | \mathcal{M}_i)$, 那么我们感兴趣的是计算不同模型的模型证据 $p(\mathcal{D} | \mathcal{M}_i)$ 。从现在开始, 为了简化记号, 我们省略对于 \mathcal{M}_i 的条件依赖。根据贝叶斯定理, 模型证据为

$$p(\mathcal{D}) = \int p(\mathcal{D} | \boldsymbol{\theta}) p(\boldsymbol{\theta}) d\boldsymbol{\theta} \tag{4.136}$$

令 $f(\theta) = p(\mathcal{D} \mid \theta)p(\theta)$ 以及 $Z = p(\mathcal{D})$, 然后使用公式 (4.135) , 我们有

$$\ln p(\mathcal{D}) \simeq \ln p(\mathcal{D} \mid \boldsymbol{\theta}_{MAP}) + \underbrace{\ln p(\boldsymbol{\theta}_{MAP}) + \frac{M}{2} \ln(2\pi) - \frac{1}{2} \ln |\mathbf{A}|}_{\text{Occam因子}} \quad (4.137)$$

其中 $\boldsymbol{\theta}_{MAP}$ 是在后验概率分布众数位置的 θ 的值, \mathbf{A} 是负对数后验概率的二阶导数组成的Hessian矩阵。

$$\mathbf{A} = -\nabla\nabla \ln p(\mathcal{D} \mid \boldsymbol{\theta}_{MAP})p(\boldsymbol{\theta}_{MAP}) = -\nabla\nabla \ln p(\boldsymbol{\theta}_{MAP} \mid \mathcal{D}) \quad (4.138)$$

公式 (4.137) 表示使用最优参数计算的对数似然值, 而余下的三项由“Occam因子”组成, 它对模型的复杂度进行惩罚。

如果我们假设参数的高斯先验分布比较宽, 且Hessian矩阵是满秩的, 那么我们可以使用下式来非常粗略地近似公式 (4.137) 。

$$\ln p(\mathcal{D}) \simeq \ln p(\mathcal{D} \mid \boldsymbol{\theta}_{MAP}) - \frac{1}{2} M \ln N \quad (4.139)$$

其中 N 是数据点的总数, M 是 θ 中参数的数量, 并且我们省略了一些额外的常数。这被称为贝叶斯信息准则 (Bayesian Information Criterion) (BIC), 或者称为Schwarz准则 (Schwarz, 1978)。注意, 与公式 (1.73) 给出的AIC相比, 这个信息准则对模型复杂度的惩罚更严重。

像AIC和BIC这样的复杂度度量很容易计算, 但是也会产生有误导性的结果。特别地, 对于Hessian矩阵满秩的假设通常不成立, 因为许多参数都不是“良好确定”的。我们可以使用基于拉普拉斯近似的公式 (4.137) 来获得对于模型证据的一个更加准确的估计, 正如我们在5.7节在神经网络模型中做的那样。

4.5 贝叶斯logistic回归

我们现在考虑logistic回归的贝叶斯观点。对于logistic回归, 精确的贝叶斯推断是无法处理的。特别地, 计算后验概率分布需要对先验概率分布于似然函数的乘积进行归一化, 而似然函数本身由一系列logistic sigmoid函数的乘积组成, 每个数据点都有一个logistic sigmoid函数。对于预测分布的计算类似地也是无法处理的。这里我们考虑使用拉普拉斯近似来处理贝叶斯logistic回归的问题 (Spiegelhalter and Lauritzen, 1990; MacKay, 1992b) 。

4.5.1 拉普拉斯近似

回忆一下, 在4.4节中, 拉普拉斯近似由下面的方式获得: 首先寻找后验概率分布的众数, 然后调节一个以众数为中心的高斯分布。这需要计算对数后验概率的二阶导数, 这等价于寻找Hessian矩阵。

由于我们寻找后验概率分布的一个高斯表示, 因此我们在开始的时候选择高斯先验是很自然的。我们把高斯先验写成一般的形式

$$p(\mathbf{w}) = \mathcal{N}(\mathbf{w} \mid \mathbf{m}_0, \mathbf{S}_0) \quad (4.140)$$

其中 \mathbf{m}_0 和 \mathbf{S}_0 是固定的超参数。 \mathbf{w} 的后验概率分布为

$$p(\mathbf{w} \mid \mathbf{t}) \propto p(\mathbf{w})p(\mathbf{t} \mid \mathbf{w}) \quad (4.141)$$

其中 $\mathbf{t} = (t_1, \dots, t_N)^T$ 。两侧取对数, 然后代入先验分布 (4.140) , 对于使用公式 (4.89) 的似然函数, 我们有

$$\begin{aligned} \ln p(\mathbf{w} \mid \mathbf{t}) &= -\frac{1}{2}(\mathbf{w} - \mathbf{m}_0)^T \mathbf{S}_0^{-1}(\mathbf{w} - \mathbf{m}_0) \\ &\quad + \sum_{n=1}^N \{t_n \ln y_n + (1 - t_n) \ln(1 - y_n)\} + \text{常数} \end{aligned} \quad (4.142)$$

其中 $y_n = \sigma(\mathbf{w}^T \boldsymbol{\phi}_n)$ 。为了获得后验概率的高斯近似，我们首先最大化后验概率分布，得到MAP（最大后验）解 \mathbf{w}_{MAP} ，它定义了高斯分布的均值。这样协方差就是负对数似然函数的二阶导数矩阵的逆矩阵，形式为

$$\mathbf{S}_N^{-1} = -\nabla \nabla \ln p(\mathbf{w} | \mathbf{t}) = \mathbf{S}_0^{-1} + \sum_{n=1}^N y_n(1-y_n) \boldsymbol{\phi}_n \boldsymbol{\phi}_n^T \quad (4.143)$$

于是后验概率分布的高斯近似的形式为

$$q(\mathbf{w}) = \mathcal{N}(\mathbf{w} | \mathbf{w}_{MAP}, \mathbf{S}_N) \quad (4.144)$$

获得了后验概率分布的高斯近似之后，剩下的任务就是关于这个概率分布求积分来进行预测。

4.5.2 预测分布

给定一个新的特征向量 $\boldsymbol{\phi}(\mathbf{x})$ ，类别 \mathcal{C}_1 的预测分布可以通过对后验概率 $p(\mathbf{w} | \mathbf{t})$ 积分，后验概率本身由高斯分布 $q(\mathbf{w})$ 近似，即

$$p(\mathcal{C}_1 | \boldsymbol{\phi}, \mathbf{t}) = \int p(\mathcal{C}_1 | \boldsymbol{\phi}, \mathbf{w}) p(\mathbf{w} | \mathbf{t}) d\mathbf{w} \simeq \int \sigma(\mathbf{w}^T \boldsymbol{\phi}) q(\mathbf{w}) d\mathbf{w} \quad (4.145)$$

且类别 \mathcal{C}_2 的对应的概率为 $p(\mathcal{C}_2 | \boldsymbol{\phi}, \mathbf{t}) = 1 - p(\mathcal{C}_1 | \boldsymbol{\phi}, \mathbf{t})$ 。为了计算预测分布，我们首先注意到函数 $\sigma(\mathbf{w}^T \boldsymbol{\phi})$ 对于 \mathbf{w} 的依赖只通过它在 $\boldsymbol{\phi}$ 上的投影而实现。记 $a = \mathbf{w}^T \boldsymbol{\phi}$ ，我们有

$$\sigma(\mathbf{w}^T \boldsymbol{\phi}) = \int \delta(a - \mathbf{w}^T \boldsymbol{\phi}) \sigma(a) da \quad (4.146)$$

其中 $\delta(\cdot)$ 是狄拉克Delta函数。由此我们有

$$\int \sigma(\mathbf{w}^T \boldsymbol{\phi}) q(\mathbf{w}) d\mathbf{w} = \int \sigma(a) p(a) da \quad (4.147)$$

其中

$$p(a) = \int \delta(a - \mathbf{w}^T \boldsymbol{\phi}) q(\mathbf{w}) d\mathbf{w} \quad (4.148)$$

我们可以这样计算 $p(a)$ ：注意到Delta函数给 \mathbf{w} 施加了一个线性限制，因此在所有与 $\boldsymbol{\phi}$ 正交的方向上积分，就得到了联合概率分布 $q(\mathbf{w})$ 的边缘分布。由于 $q(\mathbf{w})$ 是高斯分布，因此根据2.3.2节，我们知道边缘概率分布也是高斯分布。我们可以通过计算各阶矩然后交换 a 和 \mathbf{w} 的积分顺序的方式计算均值和协方差，即

$$\mu_a = \mathbb{E}[a] = \int p(a) a da = \int q(\mathbf{w}) \mathbf{w}^T \boldsymbol{\phi} d\mathbf{w} = \mathbf{w}_{MAP}^T \boldsymbol{\phi} \quad (4.149)$$

推导过程中我们使用了公式 (4.144) 给出的后验概率分布 $q(\mathbf{w})$ 的结果。类似地

$$\begin{aligned} \sigma_a^2 &= \text{var}[a] = \int p(a) \{a^2 - \mathbb{E}[a]^2\} da \\ &= \int q(\mathbf{w}) \{(\mathbf{w}^T \boldsymbol{\phi})^2 - (\mathbf{w}_{MAP}^T \boldsymbol{\phi})^2\} d\mathbf{w} = \boldsymbol{\phi}^T \mathbf{S}_N \boldsymbol{\phi} \end{aligned} \quad (4.150)$$

注意， a 的分布的函数形式与线性回归模型的预测分布 (3.58) 相同，其中噪声方差被设置为零。因此我们对于预测分布的近似变成了

$$p(\mathcal{C}_1 | \mathbf{t}) = \int \sigma(a) p(a) da = \int \sigma(a) \mathcal{N}(a | \mu_a, \sigma_a^2) da \quad (4.151)$$

这个结果也可以直接使用2.3.2节给出的高斯分布的边缘概率的结果推导出来。

关于 a 的积分表示一个高斯分布和一个logistic sigmoid函数的卷积，不能够解析地求值。然而，我们可以利用公式(4.59)定义的logistic sigmoid函数 $\sigma(a)$ 和公式(4.114)定义的逆probit函数 $\Phi(a)$ 的高度相似性来获得一个较好的近似(Spiegelhalter and Lauritzen, 1990; MacKay, 1992b; Barber and Bishop, 1998a)。为了获得对于logistic函数的最好的近似，我们需要重新为横轴定义标度，使得我们可以用 $\Phi(\lambda a)$ 近似 $\sigma(a)$ 。通过令两个函数在原点处有同样的斜率，我们可以找到 λ 的一个恰当的值，这个值为 $\lambda^2 = \frac{\pi}{8}$ 。在这种 λ 的选择下，logistic sigmoid函数和逆probit函数的相似性如图4.9所示。

使用逆probit函数的一个优势是它与高斯的卷积可以用另一个逆probit函数解析地表示出来。特别地，我们可以证明

$$\int \Phi(\lambda a) \mathcal{N}(a | \mu, \sigma^2) da = \Phi\left(\frac{\mu}{(\lambda^{-2} + \sigma^2)^{\frac{1}{2}}}\right) \quad (4.152)$$

我们现在将逆probit函数的近似 $\sigma(a) \simeq \Phi(\lambda a)$ 应用于这个方程的两侧，得到下面的对于logistic sigmoid函数与高斯的卷积近似

$$\int \sigma(a) \mathcal{N}(a | \mu, \sigma^2) da \simeq \sigma(\kappa(\sigma^2)\mu) \quad (4.153)$$

其中我们已经定义了

$$\kappa(\sigma^2) = \left(1 + \frac{\pi\sigma^2}{8}\right)^{-\frac{1}{2}} \quad (4.154)$$

把这个结果应用于公式(4.151)，我们得到了近似的预测分布，形式为

$$p(C_1 | \phi, \mathbf{t}) = \sigma(\kappa(\sigma_a^2)\mu_a) \quad (4.155)$$

其中 μ_a 和 σ_a^2 分别由公式(4.149)和公式(4.150)定义， $\kappa(\sigma_a^2)$ 由公式(4.154)定义。

注意，对应于 $p(C_1 | \phi, \mathbf{t}) = 0.5$ 的决策边界由 $\mu_a = 0$ 给出，这与使用 w 的MAP值得到的结果相同。因此，如果决策准则是基于最小分类错误率的，且先验概率相同，那么对 w 的积分没有效果。然而，对于更复杂的决策准则，这个积分就起着重要的作用了。在后验概率分布的高斯近似下，对logistic sigmoid模型的积分会在图10.13中在变量推断的问题下进行说明。

4.6 练习

(4.1) (***) 给定一组数据点 $\{\mathbf{x}_n\}$ ，我们可以将凸壳(convex hull)定义为由下式给出的所有点 \mathbf{x} 组成的集合。

$$\mathbf{x} = \sum_n \alpha_n \mathbf{x}_n \quad (4.156)$$

其中 $\alpha_n \geq 0$ 且 $\sum_n \alpha_n = 1$ 。考虑另一个点集 $\{\mathbf{y}_n\}$ 以及对应的凸壳。根据定义，如果存在一个向量 \hat{w} 和一个标量 w_0 使得 $\hat{w}^T \mathbf{x}_n + w_0 > 0$ 对于所有 \mathbf{x}_n 都成立，且 $\hat{w}^T \mathbf{y}_n + w_0 < 0$ 对所有的 \mathbf{y}_n 都成立，那么这两个点集是线性可分的。证明，如果它们的凸壳有相交的部分，那么这两个点集是线性不可分的，相反如果它们是线性可分的，那么它们的凸壳就不会相交。

(4.2) (**) 考虑平方和误差函数(4.15)的最小化问题。假设训练集里的所有的目标向量满足线性限制

$$\mathbf{a}^T \mathbf{t}_n + b = 0 \quad (4.157)$$

其中 \mathbf{t}_n 对应于公式(4.15)中的矩阵 \mathbf{T} 的第 n 行。证明，由于这条限制的存在，最小平方解(4.17)给出的模型预测 $\mathbf{y}(\mathbf{x})$ 的元素也满足这条限制，即

$$\mathbf{a}^T \mathbf{y}(\mathbf{x}) + b = 0 \quad (4.158)$$

为了证明这一点，假设基函数中的一个 $\phi_0(\mathbf{x}) = 1$ ，从而对应的参数 w_0 扮演偏置的角色。

(4.3) (**). 扩展练习4.2的结果，证明目标向量同时满足多个线性性质，那么线性模型的最小平方预测也满足同样的限制。

(4.4) (*). 证明，对于公式(4.22)给出的关于 \mathbf{w} 的类别划分准则，使用拉格朗日乘数法强制其满足限制条件 $\mathbf{w}^T \mathbf{w} = 1$ ，可以推导出 $\mathbf{w} \propto (\mathbf{m}_2 - \mathbf{m}_1)$ 。

(4.5) (*). 使用公式(4.20)、(4.23)、(4.24)，证明Fisher准则(4.25)可以写成(4.26)的形式。

(4.6) (*). 使用公式(4.27)和(4.28)给出的类间协方差矩阵和类内协方差矩阵的定义，以及公式(4.34)、(4.36)，并且目标值按照4.1.5节描述的方式取得，证明最小化误差函数的表达式(4.33)可以写成(4.37)的形式。

(4.7) (*). 证明logistic sigmoid函数(4.59)满足性质 $\sigma(-a) = 1 - \sigma(a)$ ，它的反函数为 $\sigma^{-1}(y) = \ln\left\{\frac{y}{1-y}\right\}$ 。

(4.8) (*). 使用公式(4.57)和(4.58)，推导高斯概率密度的二分类生成模型的后验概率结果(4.65)，证明参数 \mathbf{w} 和 w_0 的结果(4.66)和(4.67)。

(4.9) (*). 考虑 K 个类别的生成式分类模型，先验概率为 $p(\mathcal{C}_k) = \pi_k$ ，一般的类条件概率密度为 $p(\phi | \mathcal{C}_k)$ ，其中 ϕ 是输入特征向量。假设我们有一个训练数据集 $\{\phi_n, t_n\}$ ，其中 $n = 1, \dots, N$ ， t_n 是长度为 K 的二值变量，并且使用了“1-of- K ”的表示方式，因此如果模式 n 来自类别 \mathcal{C}_k ，那么 $t_{nj} = I_{jk}$ 。假设数据点独立地从模型中抽取，证明先验概率的最大似然解为

$$\pi_k = \frac{N_k}{N} \quad (4.159)$$

其中 N_k 是被分配到类别 \mathcal{C}_k 的数据点的数量。

(4.10) (**). 考虑练习4.9中的分类模型。现在假设类条件概率密度为高斯分布，各个类别的协方差矩阵相同，即

$$p(\phi | \mathcal{C}_k) = \mathcal{N}(\phi | \boldsymbol{\mu}_k, \Sigma) \quad (4.160)$$

证明类别 \mathcal{C}_k 的高斯分布的均值的最大似然解为

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{n=1}^N t_{nk} \phi_n \quad (4.161)$$

这表示分配到类别 \mathcal{C}_k 的特征向量的均值。类似地，证明共享的协方差矩阵的最大似然解为

$$\Sigma = \sum_{k=1}^K \frac{N_k}{N} \mathbf{S}_k \quad (4.162)$$

其中

$$\mathbf{S}_k = \frac{1}{N_k} \sum_{n=1}^N t_{nk} (\phi_n - \boldsymbol{\mu}_k)(\phi_n - \boldsymbol{\mu}_k)^T \quad (4.163)$$

因此 Σ 等于与每个类别关联的数据的协方差的加权平均，其中权系数为类别的先验概率分布。

(4.11) (**). 考虑一个 K 个类别的分类问题，其中特征向量 ϕ 有 M 个元素，每个元素可以取 L 个离散的状态。令元素的值使用“1-of- L ”的表示形式进行表示。进一步假设，以类别 \mathcal{C}_k 为条件， ϕ 的 M 个元素是独立的，从而类条件概率密度可以关于特征向量的元素进行分解。证明，在描述后验类概率密度的softmax函数中出现的由公式(4.63)给出的 a_k 是 ϕ 的元素的线性函数。注意，这是8.2.2节讨论的朴素贝叶斯模型的一个例子。

(4.12) (*). 证明由(4.59)定义的logistic sigmoid函数的导数为(4.88)。

(4.13) (*). 通过使用logistic sigmoid函数的导数的结果(4.88)，证明logistic回归模型的误差函数(4.90)的导数为(4.91)。

(4.14) (*). 证明对于一个线性可分的数据集，logistic回归模型的最大似然解可以通过下面的方式得到：找到一个向量 \mathbf{w} ，它的决策边界 $\mathbf{w}^T \phi(x) = 0$ 将类别划分开，然后令 \mathbf{w} 的长度区域无穷。

(4.15) (***) 证明logistic由公式 (4.97) 定义的回归模型的Hessian矩阵 \mathbf{H} 是正定的。这里, \mathbf{R} 是一个对角矩阵, 元素为 $y_n(1 - y_n)$, y_n 是logistic回归模型对于输入向量 \mathbf{x}_n 的输出。从而就证明了误差函数是 w 的一个凸函数, 有唯一的最小值。

(4.16) (*) 考虑一个二分类问题, 其中每个观测 \mathbf{x}_n 属于两个类别之一, 对应于 $t = 0$ 和 $t = 1$ 。假设收集训练数据的步骤不完美, 使得训练数据有时会标记错误。对于每个数据点 \mathbf{x}_n , 我们没有类别标签 t_n , 而是有一个值 π_n , 表示 $t_n = 1$ 的概率。给定一个概率模型 $p(t = 1 | \phi)$, 写下适用于这个数据集的对数似然函数。

(4.17) (*) 证明softmax激活函数 (4.104) (其中 a_k 由公式 (4.105) 定义) 的导数为 (4.106)。

(4.18) (*) 使用公式 (4.106) 给出的softmax激活函数的导数的结果, 证明交叉熵误差函数 (4.108) 的梯度为 (4.109)。

(4.19) (*) 写出4.3.5节定义的probit回归模型的对数似然函数的梯度和对应的Hessian矩阵的表达式。这些是使用IRLS训练模型时需要的量。

(4.20) (**) 证明公式 (4.110) 定义的多类logistic回归问题的Hessian矩阵是半正定的。注意, 这个问题的完整的Hessian矩阵的大小为 $MK \times MK$, 其中 M 是参数的数量, K 是类别的数量。为了证明半正定性质, 考虑乘积 $\mathbf{u}^T \mathbf{H} \mathbf{u}$, 其中 \mathbf{u} 是任意一个长度为 MK 的向量, 然后应用Jensen不等式。

(4.21) (*) 证明逆probit函数 (4.114) 和erf函数 (4.115) 的关系为 (4.116)。

(4.22) (*) 使用结果 (4.135), 推导拉普拉斯近似下的对数模型证据的表达式 (4.137)。

(4.23) (**) 本练习中, 我们从公式 (4.137) 给出的模型证据的拉普拉斯近似的结果开始, 推导出BIC的结果 (4.139)。证明, 如果参数上的先验概率分布是高斯分布, 形式为 $p(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta} | \mathbf{m}, \mathbf{V}_0)$, 那么在拉普拉斯近似下, 模型证据的对数的形式为

$$\ln p(\mathcal{D}) \simeq \ln p(\mathcal{D} | \boldsymbol{\theta}_{MAP}) - \frac{1}{2}(\boldsymbol{\theta}_{MAP} - \mathbf{m})^T \mathbf{V}_0^{-1} (\boldsymbol{\theta}_{MAP} - \mathbf{m}) - \frac{1}{2} \ln |\mathbf{H}| + \text{常数}$$

其中 \mathbf{H} 是负对数似然函数 $\ln p(\mathcal{D} | \boldsymbol{\theta})$ 在 $\boldsymbol{\theta}_{MAP}$ 处计算的二阶导数组成的矩阵。现在假设先验概率分布很宽, 从而 \mathbf{V}_0^{-1} 很小, 公式右侧的二阶项可以忽略。此外, 考虑数据独立同分布的情形, 从而 \mathbf{H} 是一系列项的求和式, 每个数据点都有一项。证明对数模型证据可以近似写成BIC表达式 (4.139) 的形式。

(4.24) (**) 使用2.3.2节的结果, 推导logistic回归模型关于参数 w 的高斯后验概率分布进行求和或积分的结果 (4.151)。

(4.25) (**) 假设我们希望通过一个缩放的逆probit函数 $\Phi(\lambda a)$ 来近似公式 (4.59) 定义的logistic sigmoid函数 $\sigma(a)$, 其中 $\Phi(a)$ 由公式 (4.114) 定义。证明, 如果 λ 的选择使得两个函数在 $a = 0$ 处的导数相等, 那么 $\lambda^2 = \frac{\pi}{8}$ 。

(4.26) (**) 本练习中, 我们要证明逆probit函数与高斯分布的卷积的结果 (4.152)。为了完成这件事, 证明左侧关于 μ 的导数等于右侧的导数, 然后对两侧关于 μ 积分, 之后证明积分的常数等于零。注意, 在对左侧进行求导之前, 比较方便的做法是首先进行变量替换 $a = \mu + \sigma z$, 因此对 a 的积分被替换为对 z 的积分。当我们对公式 (4.152) 的左侧求导时, 我们就会得到对 z 的一个高斯积分, 这个积分可以解析地计算出来。

5 神经网络

基函数固定 -> 从基函数中选择子集 -> 基函数数量固定

在第3章和第4章中，我们考虑了由固定基函数的线性组合构成的回归模型和分类模型。我们看到，这些模型具有一些有用的分析性质和计算性质，但是它们的实际应用被维数灾难问题限制了。为了将这些模型应用于大规模的问题，有必要根据数据调节基函数。

支持向量机（将在第7章讨论）是这样解决这个问题的：首先定义以训练数据点为中心的基函数，然后在训练过程中选择一个子集。支持向量机的一个优点是，虽然训练阶段涉及到非线性优化，但是目标函数是凸函数，因此最优问题的解相对很直接。最终模型中基函数的数量通常远小于训练数据点的数量，虽然通常相对来说仍然很大，并且通常随着数据规模的增加而增多。相关向量机（将在7.2节讨论）也选择固定基函数集合的一个子集，通常会生成一个相当稀疏的模型。与支持向量机不同，相关向量机也产生概率形式的输出，虽然这种输出的产生会以训练阶段的非凸优化为代价。

另一种方法是事先固定基函数的数量，但是允许基函数可调节。换句话说，就是使用参数形式的基函数，这些参数可以在训练阶段调节。在模式识别中，这种类型的最成功的模型时前馈神经网络，也被称为多层感知器（multilayer perceptron），将在本章讨论。实际上，“多层感知器”是一个相当不正确的命名，因为模型是由多层logistic回归模型（带有连续的非线性性质）组成，而不是由多层感知器（带有非连续的非线性性质）组成。对于许多应用来说，与具有同样泛化能力的支持向量机相比，最终的模型会相当简洁，因此计算的速度更快。这种简洁性带来的代价就是，与相关向量机一样，构成了网络训练根基的似然函数不再是模型参数的凸函数。然而，在实际应用中，考察模型在训练阶段消耗的计算资源是很有价值的，这样做会得到一个简洁的模型，它可以快速地处理新数据。

术语“神经网络”来源于它尝试寻找生物系统信息处理的数学表示（McCulloch and Pitts, 1943; Widrow and Hoff, 1960; Rosenblatt, 1962; Rumelhart et al., 1986）。实际上，这个模型已经被广泛使用，它涵盖了相当多的不同种类的模型，许多模型过分夸张地宣称其具有生物的可信性。然而，从模式识别的实际应用角度来说，模仿生物的真实性会带来相当多的不必要的限制。因此，我们本章中的注意力集中于作为统计模式识别的高效模型的神经网络。特别地，我们要把我们的注意力集中于神经网络中的某个具体的类别上，这一类神经网络已经被证明有相当大的实用价值。这一类神经网络就是多层感知器。

首先，我们考虑神经网络的函数形式，包括基函数的具体参数，然后我们讨论使用最大似然框架确定神经网络参数的问题，这涉及到非线性最优化问题的解。这种方法需要计算对数似然函数关于神经网络参数的导数，我们会看到这些导数可以使用误差反向传播（error backpropagation）的方法高效地获得。我们还会说明误差反向传播的框架如何推广到计算其他的导数，例如Jacobian矩阵和Hessian矩阵。接下来，我们讨论神经网络训练的正则化的各种方法，以及方法之间的关系。我们还会考虑神经网络模型的一些扩展。特别地，我们会描述一个通用的框架，用来对条件概率密度建模。这个框架被称为混合密度网络（mixture density network）。最后，我们讨论神经网络的贝叶斯观点。额外的关于神经网络模型的背景可以参考Bishop (1995a)。

5.1 前馈神经网络

回归的线性模型和分类的线性模型分别在第3章和第4章中讨论过了。它们基于固定非线性基函数 $\phi_j(\mathbf{x})$ 的线性组合，形式为

$$y(\mathbf{x}, \mathbf{w}) = f \left(\sum_{j=1}^M w_j \phi_j(\mathbf{x}) \right) \quad (5.1)$$

其中 $f(\cdot)$ 在分类问题中是一个非线性激活函数，在回归问题中为恒等函数。我们的目标是推广这个模型，使得基函数 $\phi_j(\mathbf{x})$ 依赖于参数，从而能够让这些参数以及系数 $\{w_j\}$ 能够在训练阶段调节。当然，有许多种方法构造参数化的非线性基函数。神经网络使用与公式 (5.1) 形式相同的基函数，即每个基函数本身是输入的线性组合的非线性函数，其中线性组合的系数是可调节参数。

这就引出了基本的神经网络，它可以被描述为一系列的函数变换。首先，我们构造输入变量 x_1, \dots, x_D 的 M 个线性组合，形式为

$$a_j = \sum_{i=1}^D w_{ji}^{(1)} x_i + w_{j0}^{(1)} \quad (5.2)$$

其中 $j = 1, \dots, M$ ，且上标(1)表示对应的参数是神经网络的第一“层”。我们把参数 $w_{ji}^{(1)}$ 称为权 (weight)，把参数 $w_{j0}^{(1)}$ 称为偏置 (bias)，这遵循了第3章中的命名方式。 a_j 被称为激活 (activation)。每个激活都使用一个可微的非线性激活函数 (activation function) $h(\cdot)$ 进行变换，可得

$$z_j = h(a_j) \quad (5.3)$$

这些量对应于公式 (5.1) 中的基函数的输出，这些基函数在神经网络中被称为隐含单元 (hidden unit)。非线性函数 $h(\cdot)$ 通常被选为S形的函数，例如logistic sigmoid函数或者双曲正切函数。根据公式 (5.1)，这些值再次线性组合，得到输出单元激活 (output unit activation)

$$a_k = \sum_{j=1}^M w_{kj}^{(2)} z_j + w_{k0}^{(2)} \quad (5.4)$$

其中 $k = 1, \dots, K$ ，且 K 是输出的总数量。这个变换对应于神经网络的第二层，并且与之前一样 $w_{k0}^{(2)}$ 是偏置参数。最后，使用一个恰当的激活函数对输出单元激活进行变换，得到神经网络的一组输出 y_k 。激活函数的选择由数据本身以及目标变量的假定的分布确定，并且它的确定过程遵循第3章和第4章的线性模型确定激活函数的过程。因此对于标准的回归问题，激活函数是恒等函数，从而 $y_k = a_k$ 。类似地，对于多个二元分类问题，每个输出单元激活使用logistic sigmoid函数进行变换，即

$$y_k = \sigma(a_k) \quad (5.5)$$

其中

$$\sigma(a) = \frac{1}{1 + \exp(-a)} \quad (5.6)$$

最后，对于多类问题，我们使用了公式 (4.62) 给出的softmax激活函数。输出单元激活函数的选择在5.2节中会详细讨论。

我们可以将各个阶段结合，得到整体的网络函数。对于sigmoid输出单元激活函数，整体的网络函数为

$$y_k(\mathbf{x}, \mathbf{w}) = \sigma \left(\sum_{j=1}^M w_{kj}^{(2)} h \left(\sum_{i=1}^D w_{ji}^{(1)} x_i + w_{j0}^{(1)} \right) + w_{k0}^{(2)} \right) \quad (5.7)$$

其中所有权参数和偏置参数被聚集到一起，记作向量 \mathbf{w} 。因此神经网络模型可以简单地看成一个从输入变量 $\{x_i\}$ 到输出变量 $\{y_k\}$ 的非线性函数，并且由可调节参数向量 \mathbf{w} 控制。

这个函数可以被表示成图5.1所示的网络图的形式。这样，计算公式 (5.7) 的过程可以看做信息通过网络的前向传播 (forward propagation)。需要强调的是，这些图并不表示第8章将要讨论的概率图模型，因为内部结点表示的是确定的变量而不是随机变量。因此，我们对于这两类模型采用了稍微不同的图示方法。我们稍后会看到如何给神经网络一个概率的表示。

正如3.1节讨论的那样，可以通过定义额外的输入变量 x_0 的方式将公式 (5.2) 中的偏置参数整合到权参数集合中，其中额外的输入变量 x_0 的值被限制为 $x_0 = 1$ ，因此公式 (5.2) 的形式为

$$a_j = \sum_{i=0}^D w_{ji}^{(1)} x_i \quad (5.8)$$

我们可以类似地把第二层的偏置整合到第二层的权参数中，从而整体的网络函数为

$$y_k(\mathbf{x}, \mathbf{w}) = \sigma \left(\sum_{j=0}^M w_{kj}^{(2)} h \left(\sum_{i=0}^D w_{ji}^{(1)} x_i \right) \right) \quad (5.9)$$

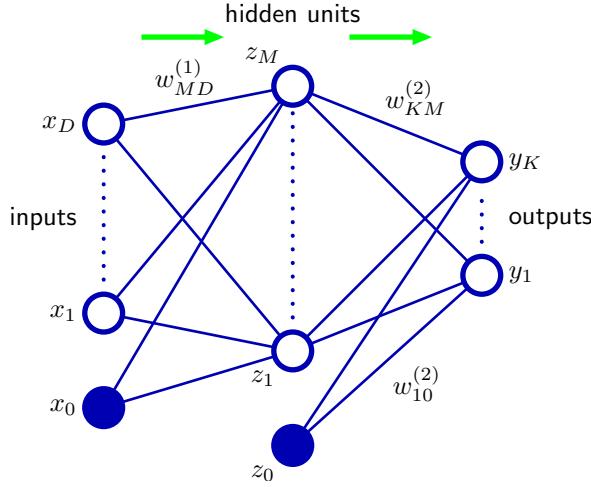


图 5.1: 对应于公式 (5.7) 的两层神经网络的网络图。输入变量、隐含变量、输出变量都表示为结点，权参数被表示为结点之间的链接，其中偏置参数被表示为来自额外的输入变量 x_0 和隐含变量 z_0 的链接。箭头表示信息流在网络中进行前向传播的方向。

正如从图5.1中可以看到的那样，神经网络模型由两个处理阶段组成，每个阶段都类似于4.1.7节讨论的感知器模型，因此神经网络也被称为多层感知器（multilayer perceptron），或者MLP。然而，与感知器模型相比，一个重要的区别是神经网络在隐含单元中使用连续的sigmoid非线性函数，而感知器使用阶梯函数这一非线性函数。这意味着神经网络函数关于神经网络参数是可微的，这个性质在神经网络的训练过程中起着重要的作用。

如果网络中的所有隐含单元的激活函数都取线性函数，那么对于任何这种网络，我们总可以找到一个等价的无隐含单元的网络。这是由于连续的线性变换的组合本身是一个线性变换。然而，如果隐含单元的数量小于输入单元的数量或者小于输出单元的数量，那么网络能够产生的变换不是最一般的从输入到输出的线性变换，因为在隐含单元出的维度降低造成了信息丢失。在12.4.2节，我们展示了线性单元的网络可以引出主成分分析。但是通常情况下，我们对线性单元的多层神经网络几乎不感兴趣。

图5.1给出的网络结构是在实际中最常用的一个。然而，它很容易扩展。例如，可以增加额外的处理层，每层包含一个由公式 (5.4) 形式的加权线性组合，以及一个使用非线性激活函数进行的元素级别的变换。注意，在文献中，关于计算这种网络的层数，有一些令人困惑的地方。因此图5.1中的网络可能被描述成一个3层网络（计算单元的层数，把输入当成单元），或者有时作为一个单一隐含层网络（计算隐含单元层的数量）。我们推荐的计算方法是把图5.1的网络称为两层网络，因为它是可调节权值的层数，这对于确定网络性质很重要。

神经网络结构的另一个扩展是引入跨层（skip-layer）链接，每个跨层链接都关联着一个对应的可调节参数。例如，在一个两层的神经网络中，跨层链接可能直接从输入链接到输出。原则上，有着sigmoid隐含单元的网络总能够模拟跨层链接（对于有界输入值），模拟的方法是使用足够小的第一层权值，从而使得隐含单元几乎是线性的，然后将隐含单元到输出的权值设置为足够大来进行补偿。然而在实际应用中，显示地包含跨层链接可能会更方便。

此外，网络可以是稀疏的。稀疏的网络中，并不是所有有可能的链接方式都被链接上。在5.5.6节讨论卷积神经网络时，我们会看到稀疏网络的一个例子。

由于在网络图和它的数学函数表达式之间有一个直接的对应关系，因此我们可以通过考虑更复杂的网络图来构造更一般的网络映射。然而，这些网络必须被限制为前馈（feed-forward）结构，换句话说，网络中不能存在有向圈，从而确保了输出是输入的确定函数。图5.2用一个简单的例子说明了这一点。这样的网络中每个（隐含或者输出）单元都计算了一个下面的函数

$$z_k = h \left(\sum_j w_{kj} z_j \right) \quad (5.10)$$

其中，求和的对象是所有向单元 k 发送链接的单元（偏置参数也包含在了求和式当中）。对于一

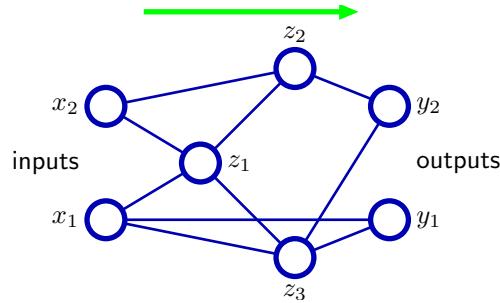


图 5.2: 具有一般的前馈拓扑结构的神经网络的例子。注意, 每个隐含电源和输出单元都与一个偏置参数关联 (为了清晰起见, 没有画出)。

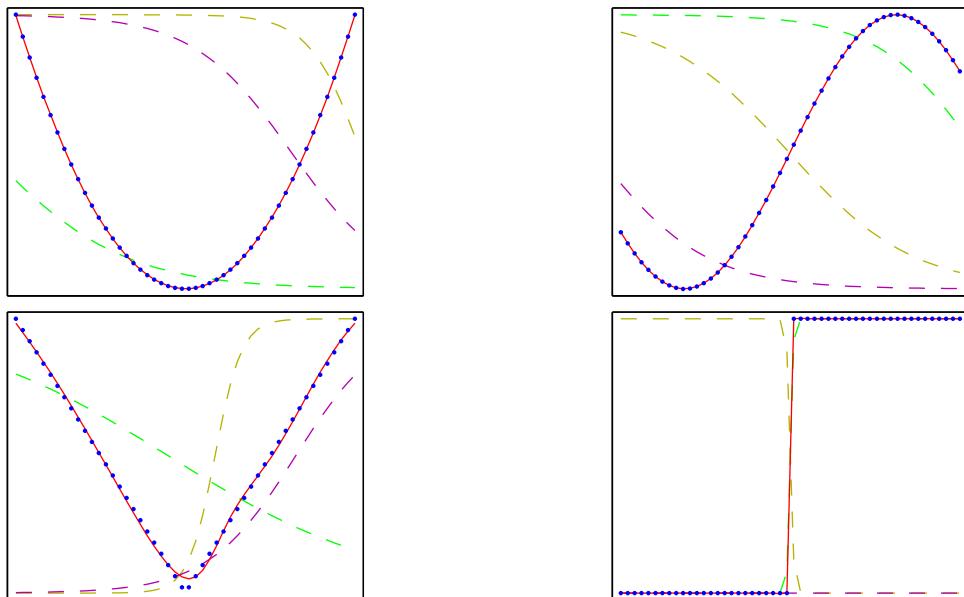


图 5.3: 多层感知器的能力说明, 它用来近似四个不同的函数。(a) $f(x) = x^2$, (b) $f(x) = \sin(x)$, (c) $f(x) = |x|$, (d) $f(x) = H(x)$, 其中 $H(x)$ 是一个硬阶梯函数。在每种情况下, $N = 50$ 个数据点 (用蓝点表示) 从区间 $(-1, 1)$ 中均匀分布的 x 中进行取样, 然后计算出对应的 $f(x)$ 值。这些数据点之后用来训练一个具有3个隐含单元的两层神经网络, 隐含单元的激活函数为tanh函数, 输出为线性输出单元。生成的网络函数使用红色曲线表示, 三个隐含单元的输出用三条虚线表示。

给定的值作用在神经网络的输入上, 不断应用公式 (5.10) 使得网络中所有单元 (包括输出单元) 的激活都能够被计算出来。

前馈网络的近似性质被广泛研究 (Funahashi, 1989; Cybenko, 1989; Hornik et al., 1989; Stinchcombe and White, 1989; Cotter, 1990; Ito, 1991; Hornik, 1991; Kreinovich, 1991; Ripley, 1996), 这些性质被发现相当通用。因此神经网络被称为通用近似 (universal approximator)。例如, 一个带有线性输出的两层网络可以在任意精度下近似任何输入变量较少的连续函数, 只要隐含单元的数量足够多。这个结果对于一大类隐含单元激活函数都成立, 但是不包括多项式函数。虽然这些定理是毋庸置疑的, 但是关键的问题是, 给定一组训练数据, 如何寻找合适的参数值。在本章的后续章节中, 我们会说明, 基于最大似然方法和贝叶斯方法, 这个问题存在高效的解法。

图5.3说明了两层网络建模一大类函数的能力。这个图也说明了独立的隐含单元是如何联合地近似最终的函数的。图5.4说明了在一个简单的分类问题中, 隐含单元的作用。使用的数据集是附录A中描述的人工生成的分类数据。

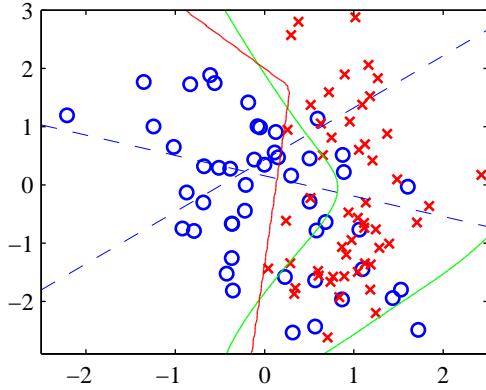


图 5.4: 简单的二分类问题的例子，数据集是人工生成的数据。模型为神经网络，网络具有两个输入结点，两个带有 \tanh 激活函数的隐含单元，以及带有 logistic sigmoid 激活函数的一个输出单元。蓝色虚线表示每个隐含单元的 $z = 0.5$ 的轮廓线，红线表示网络的 $y = 0.5$ 的决策面。为了对比，绿线表示根据生成数据的概率分布计算出的最有决策边界。

5.1.1 权空间对称性

前馈神经网络的一个性质是，对于多个不同的权向量 w 的选择，网络可能产生同样的从输入到输出的映射函数 (Chen et al., 1993)。这个性质在我们考虑贝叶斯模型比较的问题时会很有帮助。考虑图 5.1 中的两层网络，网络有 M 个隐含结点，激活函数是双曲正切函数，且两层之间完全链接。如果我们把作用于某个特定的隐含单元的所有权值以及偏置全部变号，那么对于给定的输入模式，隐含单元的激活的符号也会改变。这是因为双曲正切函数是一个奇函数，即 $\tanh(-a) = -\tanh(a)$ 。这种变换可以通过改变所有从这个隐含单元到输出单元的权值的符号的方式进行精确补偿。因此，通过改变特定一组权值（以及偏置）的符号，网络表示的输入-输出映射函数不会改变，因此我们已经找到了两个不同的权向量产生同样的映射函数。对于 M 个隐含单元，会有 M 个这样的“符号改变”对称性，因此任何给定的权向量都是 2^M 个等价的权向量中的一个。

类似地，假设我们将与某个特定的隐含结点相关联的所有输入和输出的权值（和偏置）都变为与不同的隐含结点相关联的对应的权值（和偏置）。与之前一样，这显然使得网络的输入-输出映射不变，但是对应了一个不同的权向量。对于 M 个隐含结点，任何给定的权向量都属于这种交换对称性产生的 $M!$ 个等价的权向量中的一个，它对应于 $M!$ 个不同的隐含单元的顺序。于是，网络有一个整体的权空间对称性因子 $M!2^M$ 。对于多于两层的网络，对称性的总数等于这些因子的乘积，每层隐含单元都有一个这样的因子。

可以证明，对于权空间中的各种类型的对称性，这些因子都存在（除了由于权值的具体选择导致的偶然的对称性）。此外，对称性的存在不仅是双曲正切函数的特有性质，而是对一大类的激活函数都存在的性质 (Kürková and Kainen, 1994)。在许多情况下，权空间的这种对称性几乎没有实际用处，虽然在 5.7 节我们会遇到需要考虑对称性的情形。

5.2 网络训练

目前为止，我们把神经网络看成从输入变量 x 到输出变量 y 的参数化非线性函数中的一大类。确定网络参数的一个简单的方法类似于我们在 1.1 节对多项式曲线拟合问题的讨论，因此我们需要最小化平方和误差函数。给定一个由输入向量 $\{x_n\} (n = 1, \dots, N)$ 组成的训练集，以及一个对应的目标向量 t_n 组成的集合，我们要最小化误差函数

$$E(w) = \frac{1}{2} \sum_{n=1}^N \|y(x_n, w) - t_n\|^2 \quad (5.11)$$

然而，通过给网络的输出提供一个概率形式的表示，我们可以获得对于神经网络训练的一个更加一般的观点。在 1.5.4 节，我们已经看到了使用概率进行预测的很多好处。这里，关于概率的讨论会让我们理解选择输出单元非线性函数以及选择误差函数的动机。

首先，我们讨论回归问题。现在我们只考虑一元目标变量 t 的情形，其中 t 可以取任何实数值。根据1.2.5节和3.1节的讨论，我们假定 t 服从高斯分布，均值与 \mathbf{x} 相关，由神经网络的输出确定，即

$$p(t | \mathbf{x}, \mathbf{w}) = \mathcal{N}(t | y(\mathbf{x}, \mathbf{w}), \beta^{-1}) \quad (5.12)$$

其中 β 是高斯噪声的精度（方差的倒数）。当然，这种假设有些严格。在5.6节，我们会看到如何把这种方法推广到能够接受更一般的条件概率分布。对于由公式(5.12)给出的条件分布，将输出单元激活函数取成恒等函数就足够了，因为这样的网络可以近似任何从 x 到 y 的连续函数。给定一个由 N 个独立同分布的观测组成的数据集 $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ ，以及对应的目标值 $\mathbf{t} = \{t_1, \dots, t_N\}$ ，我们可以构造对应的似然函数

$$p(\mathbf{t} | \mathbf{X}, \mathbf{w}, \beta) = \prod_{n=1}^N p(t_n | \mathbf{x}_n, \mathbf{w}, \beta)$$

取负对数，我们就得到了误差函数

$$\frac{\beta}{2} \sum_{n=1}^N \{y(\mathbf{x}_n, \mathbf{w}) - t_n\}^2 - \frac{N}{2} \ln \beta + \frac{N}{2} \ln(2\pi) \quad (5.13)$$

这可以用来学习参数 \mathbf{w} 和 β 。在5.7节，我们会讨论神经网络的贝叶斯方法，而这里我们考虑最大似然方法。注意，在神经网络的文献中，通常考虑最小化误差函数而不是最大化（对数）似然函数，因此这里我们遵循这个惯例。首先考虑 \mathbf{w} 的确定。最大化似然函数等价于最小化平方和误差函数

$$E(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^N \{y(\mathbf{x}_n, \mathbf{w}) - t_n\}^2 \quad (5.14)$$

其中我们已经去掉了相加的和相乘的常数。通过最小化 $E(\mathbf{w})$ 的方式得到的 \mathbf{w} 的值被记作 \mathbf{w}_{ML} ，因为它对应于最大化似然函数。在实际应用中，神经网络函数 $y(\mathbf{x}_n, \mathbf{w})$ 的非线性的性质导致误差函数 $E(\mathbf{w})$ 不是凸函数，因此在实际应用中可能寻找的是似然函数的局部最大值，对应于误差函数的局部最小值。这将在5.2.1节讨论。

已经找到了 \mathbf{w}_{ML} ， β 的值可以通过最小化似然函数的负对数的方式求得，结果为

$$\frac{1}{\beta_{ML}} = \frac{1}{N} \sum_{n=1}^N \{y(\mathbf{x}_n, \mathbf{w}_{ML}) - t_n\}^2 \quad (5.15)$$

注意，一旦我们寻找 \mathbf{w}_{ML} 的迭代最优化过程完成，我们就可以计算这个值。如果我们有多个目标变量，并且我们假设给定 \mathbf{x} 和 \mathbf{w} 的条件下，目标变量之间相互独立，且噪声精度均为 β ，那么目标变量的条件分布为

$$p(\mathbf{t} | \mathbf{x}, \mathbf{w}) = \mathcal{N}(\mathbf{t} | y(\mathbf{x}, \mathbf{w}), \beta^{-1} \mathbf{I}) \quad (5.16)$$

使用与一元目标变量的情形相同的推导过程，我们看到最大似然的权值由最小化平方和误差函数(5.11)确定。于是噪声的精度为

$$\frac{1}{\beta_{ML}} = \frac{1}{NK} \sum_{n=1}^N \|y(\mathbf{x}_n, \mathbf{w}_{ML}) - \mathbf{t}_n\|^2 \quad (5.17)$$

其中 K 是目标变量的数量。独立性的假设可以去掉，但是代价是使得最优化问题变得稍微复杂了一些。

回忆一下，根据4.3.6节的讨论，我们看到在误差函数（负对数似然函数）和输出单元激活函数之间有一个自然的对应关系。在回归问题中，我们可以把神经网络看成具有一个恒等输出激活函数的模型，即 $y_k = a_k$ 。对应的平方和误差函数有下面的性质

$$\frac{\partial E}{\partial a_k} = y_k - t_k \quad (5.18)$$

我们在5.3节讨论误差反向传播的时候将会用到这个结果。

现在考虑二分类的情形。二分类问题中，我们有一个单一目标变量 t ，且 $t = 1$ 表示类别 \mathcal{C}_1 ， $t = 0$ 表示类别 \mathcal{C}_2 。遵循4.3.6节中对于标准链接函数的讨论，我们考虑一个具有单一输出的网络，它的激活函数是logistic sigmoid函数

$$y = \sigma(a) \equiv \frac{1}{1 + \exp(-a)} \quad (5.19)$$

从而 $0 \leq y(\mathbf{x}, \mathbf{w}) \leq 1$ 。我们可以把 $y(\mathbf{x}, \mathbf{w})$ 表示为条件概率 $p(\mathcal{C}_1 | \mathbf{x})$ ，此时 $p(\mathcal{C}_2 | \mathbf{x})$ 为 $1 - y(\mathbf{x}, \mathbf{w})$ 。如果给定了输入，那么目标变量的条件概率分布是一个伯努利分布，形式为

$$p(t | \mathbf{x}, \mathbf{w}) = y(\mathbf{x}, \mathbf{w})^t \{1 - y(\mathbf{x}, \mathbf{w})\}^{1-t} \quad (5.20)$$

如果我们考虑一个由独立的观测组成的训练集，那么由负对数似然函数给出的误差函数就是一个交叉熵（cross-entropy）误差函数，形式为

$$E(\mathbf{w}) = - \sum_{n=1}^N \{t_n \ln y_n + (1 - t_n) \ln(1 - y_n)\} \quad (5.21)$$

其中 y_n 表示 $y(\mathbf{x}_n, \mathbf{w})$ 。注意，没有与噪声精度 β 相类似的东西，因为我们假定目标值的标记都正确。然而，模型很容易扩展到能够接受标记错误的情形。Simard et al. (2003) 发现，对于分类问题，使用交叉熵误差函数而不是平方和误差函数，会使得训练速度更快，同时提升了泛化能力。

如果我们有 K 个相互独立的二元分类问题，那么我们可以使用具有 K 个输出的神经网络，每个输出都有一个logistic sigmoid激活函数。与每个输出相关联的是一个二元类别标签 $t_k \in \{0, 1\}$ ，其中 $k = 1, \dots, K$ 。如果我们假定类别标签是独立的，那么给定输入向量，目标向量的条件概率分布为

$$p(\mathbf{t} | \mathbf{x}, \mathbf{w}) = \prod_{k=1}^K y_k(\mathbf{x}, \mathbf{w})^{t_k} [1 - y_k(\mathbf{x}, \mathbf{w})]^{1-t_k} \quad (5.22)$$

取似然函数的负对数，可以得到下面的误差函数

$$E(\mathbf{w}) = - \sum_{n=1}^N \sum_{k=1}^K \{t_{nk} \ln y_{nk} + (1 - t_{nk}) \ln(1 - y_{nk})\} \quad (5.23)$$

其中 y_{nk} 表示 $y_k(\mathbf{x}_n, \mathbf{w})$ 。与回归问题一样，对于指定的输出单元，误差函数关于激活的导数的形式为公式 (5.18)。

我们可以对比一下这个问题的神经网络解和第4章讨论过的线性分类模型给出的解，从而发现一些有趣的事情。假设我们使用图5.1所示的标准的两层神经网络。我们看到，网络第一层的权向量由各个输出所共享，而在线性模型中每个分类问题是独立地解决的。神经网络的第一层可以被看做进行了一个非线性的特征抽取，而不同的输出之间共享特征可以节省计算量，同时也提升了泛化能力。

最后，我们考虑标准的多分类问题，其中每个输入被分到 K 个互斥的类别中。二元目标变量 $t_k \in \{0, 1\}$ 使用“1-of- K ”表达方式来表示类别，从而网络的输出可以表示为 $y_k(\mathbf{x}, \mathbf{w}) = p(t_k = 1 | \mathbf{x})$ ，因此误差函数为

$$E(\mathbf{w}) = - \sum_{n=1}^N \sum_{k=1}^K t_{nk} \ln y_k(\mathbf{x}_n, \mathbf{w}) \quad (5.24)$$

根据4.3.4节的讨论，我们看到输出单元激活函数（对应于标准链接函数）是下面的softmax函数

$$y_k(\mathbf{x}, \mathbf{w}) = \frac{\exp(a_k(\mathbf{x}, \mathbf{w}))}{\sum_j \exp(a_j(\mathbf{x}, \mathbf{w}))} \quad (5.25)$$

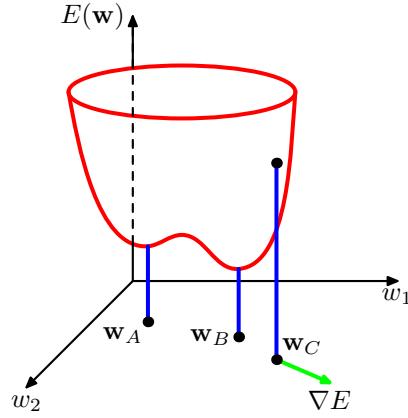


图 5.5: 误差函数 $E(\mathbf{w})$ 的几何表示。其中，误差函数被表示为权空间上的一个曲面。点 \mathbf{w}_A 是一个局部最小值，点 \mathbf{w}_B 是全局最小值。在任意点 \mathbf{w}_C 处，误差函数的局部梯度由向量 ∇E 给出。

它满足 $0 \leq y_k \leq 1$ ，且 $\sum_k y_k = 1$ 。注意，如果我们给所有的 $a_k(\mathbf{x}, \mathbf{w})$ 都加上一个常数，那么 $y_k(\mathbf{x}, \mathbf{w})$ 是不变的，这就使得误差函数在权空间的某些方向上是常数。如果我们给误差函数加上一个恰当的正则化项（第 5.5 节），那么这种问题就可以避免。

与之前一样，对于特定的输出单元，误差函数关于激活的导数的函数形式为公式 (5.18)。

总而言之，根据解决的问题的类型，关于输出单元激活函数和对应的误差函数，都存在一个自然的选择。对于回归问题，我们使用线性输出和平方和误差函数，对于（多类独立的）二元分类问题，我们使用 logistic sigmoid 输出以及交叉熵误差函数，对于多类分类问题，我们使用 softmax 输出以及对应的多分类交叉熵错误函数。对于涉及到两类的分类问题，我们可以使用单一的 logistic sigmoid 输出，也可以使用神经网络，这个神经网络有两个输出，且输出激活函数为 softmax 函数。

5.2.1 参数最优化

我们下面考虑寻找能够使得选定的误差函数 $E(\mathbf{w})$ 达到最小值的权向量 \mathbf{w} 。现在，考虑误差函数的几何表示是很有用的。我们可以把误差函数看成位于权空间的一个曲面，如图 5.5 所示。首先注意到，如果我们在权空间中走一小步，从 \mathbf{w} 走到 $\mathbf{w} + \delta\mathbf{w}$ ，那么误差函数的改变为 $\delta E \simeq \delta\mathbf{w}^T \nabla E(\mathbf{w})$ ，其中向量 $\nabla E(\mathbf{w})$ 在误差函数增加速度最大的方向上。由于误差 $E(\mathbf{w})$ 是 \mathbf{w} 的光滑连续函数，因此它的最小值出现在权空间中误差函数梯度等于零的位置上，即

$$\nabla E(\mathbf{w}) = 0 \quad (5.26)$$

这是因为，如果最小值不在这个位置上，我们就可以沿着方向 $-\nabla E(\mathbf{w})$ 走一小步，进一步减小误差。梯度为零的点被称为驻点，它可以进一步地被分为极小值点、极大值点和鞍点。

我们的目标是寻找一个向量 \mathbf{w} 使得 $E(\mathbf{w})$ 取得最小值。然而，误差函数通常与权值和偏置参数的关系是高度非线性的，因此权值空间中会有很多梯度为零（或者梯度非常小）的点。实际上，根据 5.1.1 节的讨论，我们看到，对于任意一个局部极小值点 \mathbf{w} ，在权空间中都存在等价的其他极小值点。例如，在图 5.1 所示的两层神经网络中，有 M 个隐含单元，权空间中的每个点都是 $M!2^M$ 个等价点中的一个。

此外，通常有多个不等价的驻点，通常会产生多个不等价的极小值。对于所有的权向量，误差函数的最小值被称为全局最小值（global minimum）。任何其他的使误差函数的值较大的极小值被称为局部极小值（local minima）。对于一个可以成功使用神经网络的应用来说，可能没有必要寻找全局最小值（并且通常无法知道是否找到了全局最小值），而是通过比较几个局部极小值就能够得到足够好的解。

由于显然无法找到方程 $\nabla E(\mathbf{w}) = 0$ 的解析解，因此我们使用迭代的数值方法。连续非线性函数的最优化问题是一个被广泛研究的问题，有相当多的文献讨论如何高效地解决。大多数方法涉及到为权向量选择某个初始值 \mathbf{w}_0 ，然后在权空间中进行一系列移动，形式为

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} + \Delta\mathbf{w}^{(\tau)} \quad (5.27)$$

其中 τ 表示迭代次数。不同的算法涉及到权向量更新 $\Delta w^{(\tau)}$ 的不同选择。许多算法使用梯度信息，因此就需要在每次更新之后计算在新的权向量 $w^{(\tau+1)}$ 处的 $\Delta E(w)$ 的值。为了理解梯度信息的重要性，有必要考虑误差函数基于泰勒展开的局部近似。

5.2.2 局部二次近似

通过讨论误差函数的局部二次近似，我们可以更深刻地认识最优化问题，以及各种解决最优化问题的方法。

考虑 $E(w)$ 在权空间某点 \hat{w} 处的泰勒展开

$$E(w) \simeq E(\hat{w}) + (w - \hat{w})^T b + \frac{1}{2}(w - \hat{w})^T H(w - \hat{w}) \quad (5.28)$$

其中立方项和更高阶的项已经被省略掉了。这里， b 被定义为 E 的梯度在 \hat{w} 处的值。

$$b \equiv \nabla E|_{w=\hat{w}} \quad (5.29)$$

Hessian矩阵 $H = \nabla \nabla E$ 的元素为

$$(H)_{ij} \equiv \left. \frac{\partial E}{\partial w_i \partial w_j} \right|_{w=\hat{w}} \quad (5.30)$$

根据公式 (5.18)，梯度的局部近似为

$$\nabla E \simeq b + H(w - \hat{w}) \quad (5.31)$$

对于距离点 \hat{w} 充分近的点 w ，这些表达式能够对误差函数和它的梯度给出合理的近似。

考虑一个特殊情况：在误差函数最小值点 w^* 附近的局部二次近似。在这种情况下，没有线性项，因为在 w^* 处 $\nabla E = 0$ ，公式 (5.28) 变成了

$$E(w) \simeq E(w^*) + \frac{1}{2}(w - w^*)^T H(w - w^*) \quad (5.32)$$

这里Hessian矩阵在点 w^* 处计算。为了用几何的形式表示这个结果，考虑Hessian矩阵的特征值方程

$$Hu_i = \lambda_i u_i \quad (5.33)$$

其中特征向量 u_i 构成了完备的单位正交集合（附录C），即

$$u_i^T u_j = \delta_{ij} \quad (5.34)$$

我们现在把 $(w - w^*)$ 展开成特征值的线性组合的形式

$$w - w^* = \sum_i \alpha_i u_i \quad (5.35)$$

这可以被看成坐标系的变换，坐标系的原点变为了 w^* ，坐标轴旋转，与特征向量对齐（通过列为 u_i 的正交矩阵）。附录C给出了更详细的讨论。将公式 (5.35) 代入公式 (5.32)，然后使用公式 (5.33) 和公式 (5.34)，误差函数可以写成下面的形式

$$E(w) = E(w^*) + \frac{1}{2} \sum_i \lambda_i \alpha_i^2 \quad (5.36)$$

矩阵 H 是正定的（positive definite）当且仅当

$$v^T H v > 0 \quad \text{对所有的 } v \neq 0 \text{ 都成立} \quad (5.37)$$

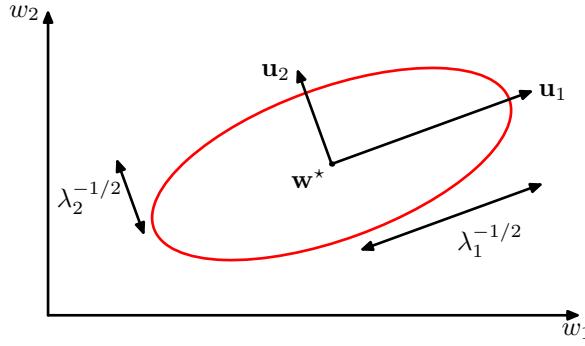


图 5.6: 在最小值 w^* 的邻域中, 误差函数可以用二次函数近似。这样, 常数误差函数的轮廓线为椭圆, 它的轴与Hessian矩阵的特征向量 u_i 给出, 长度与对应的特征值 λ_i 的平方根成反比。

由于特征向量 $\{u_i\}$ 组成了一个完备集, 因此任意的向量 v 都可以写成下面的形式

$$v = \sum_i c_i u_i \quad (5.38)$$

根据公式 (5.33) 和公式 (5.34), 我们可以得到

$$v^T H v = \sum_i c_i^2 \lambda_i \quad (5.39)$$

因此 H 是正定的, 当且仅当它的所有的特征值均严格为正。在新的坐标系中, 基向量是特征向量 $\{u_i\}$, E 为常数的轮廓线是以原点为中心的椭圆, 如图 5.6 所示。对于一维权空间, 驻点 w^* 满足下面条件时取得最小值

$$\left. \frac{\partial^2 E}{\partial w^2} \right|_{w^*} > 0 \quad (5.40)$$

对应的 D 维的结论是, 在 w^* 处的 Hessian 矩阵是正定矩阵。

5.2.3 使用梯度信息

正如我们将在 5.3 节看到的那样, 可以使用误差反向传播的方法高效地计算误差函数的梯度。这个梯度信息的使用可以大幅度加快找到极小值点的速度。原因如下所述。

在公式 (5.28) 给出的误差函数的二次近似中, 误差曲面由 b 和 H 确定, 它包含了总共 $\frac{W(W+3)}{2}$ 个独立的元素 (因为矩阵 H 是对称的), 其中 W 是 w 的维度 (即网络中可调节参数的总数)。这个二次近似的极小值点的位置因此依赖于 $O(W^2)$ 个参数, 并且我们不应该奢求能够在收集到 $O(W^2)$ 条独立的信息之前就能够找到最小值。如果我们不使用梯度信息, 我们不得不进行 $O(W^2)$ 次函数求值, 每次求值都需要 $O(W)$ 个步骤。因此, 使用这种方法求最小值需要的计算复杂度为 $O(W^3)$ 。

现在将这种方法与使用梯度信息的方法进行对比。由于每次计算 ∇E 都会带来 W 条信息, 因此我们可能预计找到函数的极小值需要计算 $O(W)$ 次梯度。正如我们将要看到的那样, 通过使用误差反向传播算法, 每个这样的计算只需要 $O(W)$ 步, 因此使用这种方法可以在 $O(W^2)$ 个步骤内找到极小值。因此, 使用梯度信息构成了训练神经网络的实际算法的基础。

5.2.4 梯度下降最优化

最简单的使用梯度信息的方法是, 将公式 (5.27) 中的权值更新方式选择为下面的形式: 每次权值更新都是在负梯度方向上的一次小的移动, 即

$$w^{(\tau+1)} = w^{(\tau)} - \eta \nabla E(w^{(\tau)}) \quad (5.41)$$

其中参数 $\eta > 0$ 被称为学习率 (learning rate)。在每次更新之后, 梯度会使用新的权值向量重新计算, 然后这个过程重复下去。注意, 误差函数是关于训练集定义的, 因此为了计算 ∇E , 每一

步都需要处理整个数据集。在每一步，权值向量都会沿着误差函数下降速度最快的方向移动，因此这种方法被称为梯度下降法（gradient descent）或者最陡峭下降法（steepest descent）。虽然这种方法在直觉上看比较合理，但是实际上可以证明它是一个很差的算法，原因可以参考Bishop and Nabney (2008)。

对于批量最优化方法，存在更高效的方法，例如共轭梯度法（conjugate gradient）或者拟牛顿法（quasi-Newton）。与简单的梯度下降方法相比，这些方法更鲁棒，更快（Gill et al., 1981; Fletcher, 1987; Nocedal and Wright, 1999）。与梯度下降方法不同，这些算法具有这样的性质：误差函数在每次迭代时总是减小的，除非权向量到达了局部的或者全局的最小值。

为了找到一个足够好的极小值，可能有必要多次运行基于梯度的算法，每次都使用一个不同的随机选择额起始点，然后在一个独立的验证集上对比最终的表现。

然而，梯度下降法有一个在线的版本，这个版本被证明在实际应用中对于使用大规模数据集来训练神经网络的情形很有用（LeCun et al., 1989）。基于一组独立观测的最大似然函数的误差函数由一个求和式构成，求和式的每一项都对应着一个数据点

$$E(\mathbf{w}) = \sum_{n=1}^N E_n(\mathbf{w}) \quad (5.42)$$

在线梯度下降，也被称为顺序梯度下降（sequential gradient descent）或者随机梯度下降（stochastic gradient descent），使得权向量的更新每次只依赖于一个数据点，即

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} - \eta \nabla E_n(\mathbf{w}^{(\tau)}) \quad (5.43)$$

这个更新在数据集上循环重复进行，并且既可以顺序地处理数据，也可以随机地有重复地选择数据点。当然，也有折中的方法，即每次更新依赖于数据点的一小部分。

与批处理相比，在线方法的一个优点是可以更加高效地处理数据中的冗余性。为了说明，让我们考虑这样一种极端的情形：给定一个数据集，我们将每个数据点都复制一次，从而将数据集的规模翻倍。注意这仅仅把误差函数乘以了一个因子2，因此等价于使用原始的误差函数。批处理方法必须付出两倍的计算量来计算误差函数的梯度，而在线方法不受影响。在线梯度下降方法的另一个性质是，可以逃离局部极小值点，因为整个数据集的关于误差函数的驻点通常不会是每个数据点各自的驻点。

非线性最优化算法，以及它们对于神经网络训练的实际应用，在Bishop and Nabney (2008) 中有详细的讨论。

5.3 误差反向传播

本节中，我们的目标是寻找一种计算前馈神经网络的误差函数 $E(\mathbf{w})$ 的梯度的一种高效的方法。我们会看到，可以使用局部信息传递的思想完成这一点。在局部信息传递的思想中，信息在神经网络中交替地向前、向后传播。这种方法被称为误差反向传播（error backpropagation），有时简称“反传”（backprop）。

应该注意的是，在神经网络计算的文献中，反向传播这个术语用于指代许多不同的事物。例如，多层感知器结构有时被称为反向传播网络。反向传播这个术语还用于描述将梯度下降法应用于平方和误差函数的多层感知器的训练过程。为了不让概念发生混淆，仔细研究一下训练过程的本质是很有用的。大部分训练算法涉及到一个迭代的步骤用于误差函数的最小化，以及通过一系列的步骤进行的权值调节。在每一个这样迭代过程中，我们可以区分这两个不同的阶段。在第一个阶段，误差函数关于权值的导数必须被计算出来。正如我们稍后会看到的那样，反向传播方法的一个重要的贡献是提供了计算这些导数的一个高效的方法。由于正是在这个阶段，误差通过网络进行反向传播，因此我们将专门使用反向传播这个术语来描述计算导数的过程。在第二个阶段，导数用于计算权值的调整量。最简单的方法，也是最开始由Rumelhart et al. (1986) 考虑的方法，涉及到梯度下降。认识到这两个阶段属于不同的阶段是很重要的。因此，第一阶段，即为了计算导数而进行的误差在网络中的反向传播阶段，可以应用于许多其他种类的网络，而不仅仅是多层感知器。它也可以应用于其他的误差函数，而不仅仅是简单的平方和误差函数。它也可以用于计算其他类型的导数，例如Jacobian矩阵和Hessian矩阵，正如我们将在本章后面看到的那样。类似地，第二阶段，即使使用计算过的导数调整权值的阶段，可以使用许多最优化方法处理，许多最优化方法本质上要比简单的梯度下降更强大。

5.3.1 误差函数导数的计算

我们现在推导适用于一般神经网络的反向传播算法。这种神经网络有着任意的前馈拓扑结构，任意可微的非线性激活函数，以及一大类的误差函数。推导的结果将会使用一个简单的层次网络结构说明，这个简单的层次网络结构有一个单层的sigmoid隐含单元以及平方和误差函数。

许多实际应用中使用的误差函数，例如针对一组独立同分布的数据的最大似然方法定义的误差函数，由若干项的求和式组成，每一项对应于训练集的一个数据点，即

$$E(\mathbf{w}) = \sum_{n=1}^N E_n(\mathbf{w}) \quad (5.44)$$

这里，我们要考虑的是计算 $\nabla E_n(\mathbf{w})$ 的问题。这可以直接使用顺序优化的方法计算，或者使用批处理方法在训练集上进行累加。

首先考虑一个简单的线性模型，其中输出 y_k 是输入变量 x_i 的线性组合，即

$$y_k = \sum_i w_{ki} x_i \quad (5.45)$$

对于一个特定的输入模式 n ，误差函数的形式为

$$E_n = \frac{1}{2} \sum_k (y_{nk} - t_{nk})^2 \quad (5.46)$$

其中 $y_{nk} = y_k(\mathbf{x}_n, \mathbf{w})$ 。这个误差函数关于一个权值 w_{ji} 的梯度为

$$\frac{\partial E_n}{\partial w_{ji}} = (y_{nj} - t_{nj}) x_{ni} \quad (5.47)$$

它可以表示为与链接 w_{ji} 的输出端相关联的“误差信号” $y_{nj} - t_{nj}$ 和与链接的输入端相关联的变量 x_{ni} 的乘积。在4.3.2节，我们看到，对于logistic sigmoid激活函数和交叉熵误差函数，以及softmax激活函数和与之匹配的交叉熵误差函数，也可以得到类似的结果。我们现在会看到这个简单的结果如何扩展到更复杂的多层前馈神经网络中。

在一个一般的前馈网络中，每个单元都会计算输入的一个加权和，形式为

$$a_j = \sum_i w_{ji} z_i \quad (5.48)$$

其中 z_i 是一个单元的激活，或者是输入，它向单元 j 发送一个链接， w_{ji} 是与这个链接关联的权值。在5.1节，我们看到偏置可以被整合到这个求和式中，整合的方法是引入一个额外的单元或输入，然后令激活恒为+1。于是我们不需要显示地处理偏置。公式 (5.48) 中的求和式通过一个非线性激活函数 $h(\cdot)$ 进行变换，得到单元 j 的激活 z_j ，形式为

$$z_j = h(a_j) \quad (5.49)$$

注意，公式 (5.48) 的求和式中的某个或某几个 z_i 可以是输入，类似地，公式 (5.49) 中的单元 j 可以是输出。

对于训练集里的每个模式，我们会假定我们给神经网络提供了对应的输入向量，然后通过反复应用公式 (5.48) 和公式 (5.49)，计算神经网络中所有隐含单元和输出单元的激活。这个过程通常被称为正向传播 (forward propagation)，因为它可以被看做网络中的一个向前流动的信息流。

现在考虑计算 E_n 关于权值 w_{ji} 的导数。各个单元的输出会依赖于某个特定的输入模式 n 。但是，为了保持记号的简介，我们将省略神经网络变量中的下标 n 。首先，我们注意到 E_n 只通过单元 j 的经过求和之后的输入 a_j 对权值 w_{ji} 产生依赖。因此，我们可以应用偏导数的链式法则，得到

$$\frac{\partial E_n}{\partial w_{ji}} = \frac{\partial E_n}{\partial a_j} \frac{\partial a_j}{\partial w_{ji}} \quad (5.50)$$



图 5.7: 对于隐含单元 j , 计算 δ_j 的说明。计算时使用了向单元 j 发送信息的那些单元 k 的 δ , 使用反向误差传播方法进行计算。蓝色箭头表示在正向传播阶段信息流的方向, 红色箭头表示误差信息的反向传播。

现在我们引入一个有用的记号

$$\delta_j \equiv \frac{\partial E_n}{\partial a_j} \quad (5.51)$$

其中 δ 通常被称为误差 (error), 原因我们稍后会看到。使用公式 (5.48), 我们有

$$\frac{\partial a_j}{\partial w_{ji}} = z_i \quad (5.52)$$

将公式 (5.51) 和公式 (5.52) 代入公式 (5.50), 我们有

$$\frac{\partial E_n}{\partial w_{ji}} = \delta_j z_i \quad (5.53)$$

公式 (5.53) 告诉我们, 要找的导数可以通过简单地将权值输出单元的 δ 值与权值输入端的 z 值相乘的方式得到 (对于偏置的情形, $z = 1$)。注意, 这与本节开始时讨论的简单线性模型的形式相同。因此, 为了计算导数, 我们只需要计算网络中每个隐含结点和输出结点的 δ_j 的值, 然后应用公式 (5.53) 即可。

正如我们已经看到的那样, 只要我们使用标准链接函数作为输出单元的激活函数, 那么对于输出单元, 我们就有

$$\delta_k = y_k - t_k \quad (5.54)$$

为了计算隐含单元的 δ 值, 我们再次使用偏导数的链式法则

$$\delta_j \equiv \frac{\partial E_n}{\partial a_j} = \sum_k \frac{\partial E_n}{\partial a_k} \frac{\partial a_k}{\partial a_j} \quad (5.55)$$

其中求和式的作用对象是所有向单元 j 发送链接的单元 k 。图 5.7 说明了单元和权值的设定。注意, 单元 k 可以包含其他的隐含单元和 (或) 输出单元。我们在给出公式 (5.55) 时, 我们使用了这个事实: a_j 的改变所造成的误差函数的改变的唯一来源是变量 a_k 的改变。如果我们把公式 (5.51) 给出的 δ 的定义代入公式 (5.55), 然后使用公式 (5.48) 和公式 (5.49), 我们就得到了下面的反向传播 (backpropagation) 公式

$$\delta_j = h'(a_j) \sum_k w_{kj} \delta_k \quad (5.56)$$

这表明, 一个特定的隐含单元的 δ 值可以通过将网络中更高层单元的 δ 进行反向传播来实现, 如图 5.7 所示。注意, 公式 (5.56) 中的求和式是对 w_{kj} 的第一个下标进行求和的 (对应于信息在网络中的反向传播), 而在正向传播方程 (5.10) 中, 求和过程针对的是第二个下标。由于我们已经知道了输出单元的 δ , 因此通过递归地应用公式 (5.56), 我们可以计算前馈网络中所有隐含单元的 δ 值, 无论它的拓扑结构是什么样的。

于是, 反向传播算法可以总结如下。

- 对于网络的一个输入向量 x_n , 使用公式 (5.48) 和公式 (5.49) 进行正向传播, 找到所有隐含单元和输出单元的激活。
- 使用公式 (5.54) 计算所有输出单元的 δ_k 。

- 使用公式 (5.56) 反向传播 δ , 获得网络中所有隐含单元的 δ_j 。
- 使用公式 (5.53) 计算导数。

对于批处理方法, 总误差函数 E 的导数可以通过下面的方式得到: 对于训练集里的每个模式, 重复上面的步骤, 然后对所有的模式求和, 即

$$\frac{\partial E}{\partial w_{ji}} = \sum_n \frac{\partial E_n}{\partial w_{ji}} \quad (5.57)$$

在上面的推导中, 我们隐式地假设网络中的每个隐含单元或输入单元都有相同的激活函数 $h(\cdot)$ 。然而, 这个推导很容易推广, 使得不同的单元可以有各自的激活函数, 只需记录那种形式的 $h(\cdot)$ 进入了那个单元即可。

5.3.2 一个简单的例子

上面对于反向传播算法的推导适用于一般形式的误差函数、激活函数、以及网络拓扑结构。为了说明这个算法的应用, 我们考虑一个具体的例子。这个例子很简单, 在实际应用中也很重要, 因为文献中出现的神经网络的许多应用都使用的这种类型的网络。具体地, 我们会考虑图 5.1 中的两层神经网络, 误差函数为平方和误差函数, 输出单元的激活函数为线性激活函数, 即 $y_k = a_k$, 而隐含单元的激活函数为 S 形函数, 形式为

$$h(a) \equiv \tanh(a) \quad (5.58)$$

其中

$$\tanh(a) = \frac{e^a - e^{-a}}{e^a + e^{-a}} \quad (5.59)$$

这个函数的一个有用的特征是, 它的导数可以表示成一个相当简单形式

$$h'(a) = 1 - h(a)^2 \quad (5.60)$$

我们也考虑一个标准的平方和误差函数, 即对于模式 n , 误差为

$$E_n = \frac{1}{2} \sum_{k=1}^K (y_k - t_k)^2 \quad (5.61)$$

其中, 对于一个特定的输入模式 x_n , y_k 是输出单元 k 的激活, t_k 是对应的目标值。

对于训练集里的每个模式, 我们首先使用下面的公式进行前向传播。

$$a_j = \sum_{i=0}^D w_{ji}^{(1)} x_i \quad (5.62)$$

$$z_j = \tanh(a_j) \quad (5.63)$$

$$y_k = \sum_{j=0}^M w_{kj}^{(2)} z_j \quad (5.64)$$

接下来我们使用下面的公式计算每个输出单元的 δ 值。

$$\delta_k = y_k - t_k \quad (5.65)$$

然后, 我们使用下面的公式将这些 δ 值反向传播, 得到隐含单元的 δ 值。

$$\delta_j = (1 - z_j^2) \sum_{k=1}^K w_{kj} \delta_k \quad (5.66)$$

最后, 关于第一层权值和第二层权值的导数为

$$\frac{\partial E_n}{\partial w_{ji}^{(1)}} = \delta_j x_i, \quad \frac{\partial E_n}{\partial w_{kj}^{(2)}} = \delta_k z_j \quad (5.67)$$

5.3.3 反向传播的效率

反向传播的一个重要的方面是它的计算效率。为了理解这一点，让我们考察误差函数导数的计算次数与网络中权值和偏置总数 W 的关系。计算一次误差函数（对于给定的输入模式）需要 $O(W)$ 次操作，其中 W 充分大。这是因为，除非网络的链接非常稀疏，否则权值的数量通常比单元的数量要大得多，因此正向传播的计算复杂度主要取决于公式（5.48）的求和式的计算，而激活函数的计算就相对耗时较少。公式（5.48）的求和式的每一项需要一次乘法和一次加法，从而整体的计算开销为 $O(W)$ 。

另一种计算误差函数导数的反向传播方法是使用有限差。首先让每个权值有一个扰动，然后使用下面的表达式来近似导数

$$\frac{\partial E_n}{\partial w_{ji}} = \frac{E_n(w_{ji} + \epsilon) - E_n(w_{ji})}{\epsilon} + O(\epsilon) \quad (5.68)$$

其中 $\epsilon \ll 1$ 。在软件仿真中，通过让 ϵ 变小，对于导数的近似的精度可以提升，直到 ϵ 过小，造成下溢问题。通过使用对称的中心差（central difference），有限差方法的精度可以极大地提高。中心差的形式为

$$\frac{\partial E_n}{\partial w_{ji}} = \frac{E_n(w_{ji} + \epsilon) - E_n(w_{ji} - \epsilon)}{2\epsilon} + O(\epsilon^2) \quad (5.69)$$

在这种情况下， $O(\epsilon)$ 修正项等于零，这可以通过公式（5.69）右侧的泰勒展开证明，从而剩下的修正项是 $O(\epsilon^2)$ 。然而，与公式（5.68）相比，计算步骤数大约变成了二倍。

计算数值导数的方法的主要问题是，计算复杂度为 $O(W)$ 这一性质不再成立。每次正向传播需要 $O(W)$ 步，而网络中有 W 个权值，每个权值必须被单独地施加扰动，因此整体的时间复杂度为 $O(W^2)$ 。

然而，数值导数的方法在实际应用中具有重要的作用，因为将反向传播算法计算的导数与使用中心差计算的导数进行对比，可以有效地检查反向传播算法的执行正确性。在实际应用中，当训练一个网络时，导数应该使用反向传播算法计算，因为这种方法有最高的精度和效率。然而，应该使用一些测试样例，将结果与公式（5.69）的数值导数的结果进行对比，检查执行的正确性。

5.3.4 Jacobian矩阵

我们已经看到了误差函数关于权值的导数是如何通过网络中的误差反向传播来获得的。误差反向传播技术也可以用来计算其他类型的导数。这里，我们考虑Jacobian矩阵的计算，它的元素的值是网络的输出关于输入的导数

$$J_{ki} \equiv \frac{\partial y_k}{\partial x_i} \quad (5.70)$$

其中，计算每个这样的导数时，其他的输入都固定。Jacobian矩阵在由许多不同模块构建的系统中很有用，如图5.8所示。每个模块可以由一个固定的或可调节的函数构成，可以是线性的或者非线性的，只要可微即可。假设我们想关于图5.8中的参数 w ，最小化误差函数 E 。误差函数的导数为

$$\frac{\partial E}{\partial w} = \sum_{k,j} \frac{\partial E}{\partial y_k} \frac{\partial y_k}{\partial z_j} \frac{\partial z_j}{\partial w} \quad (5.71)$$

其中，图5.8中的红色模块的Jacobian矩阵出现在中间项。

由于Jacobian矩阵度量了输出对于每个输入变量的改变的敏感性，因此它也允许与输入关联的任意已知的误差 Δx_i 在训练过的网络中传播，从而估计他们对于输出误差 Δy_k 的贡献。二者的关系为

$$\Delta y_k \simeq \sum_i \frac{\partial y_k}{\partial x_i} \Delta x_i \quad (5.72)$$

只要 $|\Delta x_i|$ 较小，这个关系就成立。通常，训练过的神经网络表示的网络映射是非线性的，因此Jacobian矩阵的元素不会是常数，而是依赖于具体使用的输入向量。因此公式（5.72）只在输入有较小的扰动时成立，并且对于每个新的输入变量，Jacobian矩阵必须重新计算。

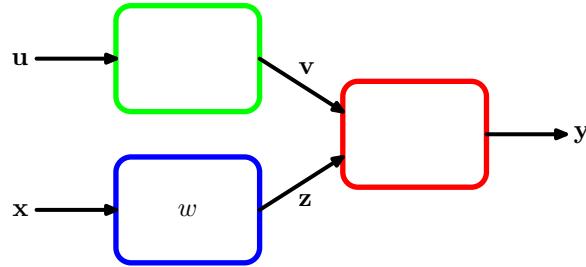


图 5.8: 模块化模式识别系统的例子，其中Jacobian矩阵可以用来将误差信号从输出模块在系统中反向传播到更早的模块。

Jacobian矩阵可以使用反向传播的方法计算，计算方法类似于之前推导误差函数关于权值的导数的方法。首先，我们把元素 J_{ki} 写成下面的形式

$$\begin{aligned} J_{ki} &= \frac{\partial y_k}{\partial x_i} = \sum_j \frac{\partial y_k}{\partial a_j} \frac{\partial a_j}{\partial x_i} \\ &= \sum_j w_{ji} \frac{\partial y_k}{\partial a_j} \end{aligned} \quad (5.73)$$

其中我们使用了公式 (5.48)。公式 (5.73) 中的求和式作用于所有单元*i*发送链接的单元*j*上（例如，之前讨论的层次拓扑结构中的第一个隐含层的所有单元）。我们现在一个递归的反向传播公式来确定导数 $\frac{\partial y_k}{\partial a_j}$ 。

$$\begin{aligned} \frac{\partial y_k}{\partial a_j} &= \sum_l \frac{\partial y_k}{\partial a_l} \frac{\partial a_l}{\partial a_j} \\ &= h'(a_j) \sum_l w_{lj} \frac{\partial y_k}{\partial a_l} \end{aligned} \quad (5.74)$$

其中求和的对象为所有单元*j*发送链接的单元*l*（对应于 w_{lj} 的第一个下标）。与之前一样，我们使用了公式 (5.48) 和公式 (5.49)。这个反向传播开始于输出单元。对于输出单元，导数可以直接从输出单元激活函数的函数形式中得到。例如，如果对于每个输出单元，我们都有各自的sigmoid函数，那么

$$\frac{\partial y_k}{\partial a_l} = \delta_{kl} \sigma'(a_l) \quad (5.75)$$

而对于softmax输出，我们有

$$\frac{\partial y_k}{\partial a_l} = \delta_{kl} y_k - y_k y_l \quad (5.76)$$

我们可以将计算Jacobian矩阵的方法总结如下。将输入空间中要寻找Jacobian矩阵的点映射成一个输入向量，将这个输入向量作为网络的输入，使用通常的正向传播方法，得到网络的所有隐含单元和输出单元的激活。接下来，对于Jacobian矩阵的每一行*k*（对应于输出单元*k*），使用递归关系 (5.74) 进行反向传播。对于网络中所有的隐含结点，反向传播开始于公式 (5.75) 和公式 (5.76)。最后，使用公式 (5.73) 进行对输入单元的反向传播。Jacobian矩阵的另一种计算方法是正向传播算法，它可以使用与这里给出的反向传播算法相类似的方式推导出来。

与之前一样，这个算法的执行可以通过下面的数值导数的方法检验正确性。

$$\frac{\partial y_k}{\partial x_i} = \frac{y_k(x_i + \epsilon) - y_k(x_i - \epsilon)}{2\epsilon} + O(\epsilon^2) \quad (5.77)$$

对于一个有着*D*个输入的网络来说，这种方法需要 $2D$ 次正向传播。

5.4 Hessian矩阵

我们已经说明了反向传播的方法如何用来得到误差函数关于网络的权值的一阶导数。反向传播也可以用来计算误差函数的二阶导数，形式为

$$\frac{\partial^2 E}{\partial w_{ji} \partial w_{lk}} \quad (5.78)$$

注意，有时将所有的权值和偏置参数看成一个向量（记作 \mathbf{w} ）的元素 w_i 更方便，此时二阶导数组成了Hessian矩阵 \mathbf{H} 的元素 H_{ij} ，其中 $i, j \in \{1, \dots, W\}$ ，且 W 是权值和偏置的总数。Hessian矩阵在神经网络计算的许多方面都有着重要的作用，包括：

- 一些用来训练神经网络的非线性最优化算法是基于误差曲面的二阶性质的，这些性质由Hessian矩阵控制 (Bishop and Nabney, 2008)。
- 对于训练数据的微小改变，Hessian矩阵构成了快速重新训练前馈网络的算法的基础 (Bishop, 1991)。
- Hessian矩阵的逆矩阵用来鉴别神经网络中最不重要的权值，这是网络“剪枝”算法的一部分 (LeCun et al., 1990)。
- Hessian矩阵是贝叶斯神经网络（见5.7节）的拉普拉斯近似的中心。它的逆矩阵用来确定训练过的神经网络的预测分布，它的特征值确定了超参数的值，它的行列式用来计算模型证据。

计算神经网络的Hessian矩阵有很多近似方法。然而，使用反向传播方法的一个扩展，Hessian矩阵可以精确地被计算出来。

对于Hessian矩阵的很多应用来说，一个重要的需要考虑的问题是计算效率。如果网络中有 W 个参数（权值和偏置），那么Hessian矩阵的维度为 $W \times W$ ，因此对于数据集里的每个模式来说，计算Hessian矩阵的计算量为 $O(W^2)$ 。正如我们将看到的那样，计算Hessian矩阵的高效方法的计算复杂度确实是 $O(W^2)$ 。

5.4.1 对角近似

上面讨论的Hessian矩阵的一些应用需要求出Hessian矩阵的逆矩阵，而不是Hessian矩阵本身。因此，我们对Hessian矩阵的对角化近似比较感兴趣。换句话说，就是把非对角线上的元素置为零，因为这样做之后，矩阵的逆矩阵很容易计算。与之前一样，我们考虑由一系列项的求和式组成的误差函数，每一项对应于数据集里的一个模式，即 $E = \sum_n E_n$ 。这样，Hessian矩阵可以通过每次考虑一个模式然后对所有模式求和的方法得到。根据公式 (5.48)，对于模式 n ，Hessian矩阵的对角线元素可以写成

$$\frac{\partial^2 E_n}{\partial w_{ji}^2} = \frac{\partial^2 E_n}{\partial a_j^2} z_i^2 \quad (5.79)$$

使用公式 (5.48) 和公式 (5.49)，公式 (5.79) 的右侧的二阶导数可以通过递归地使用微分的链式法则的方式求出。这样，可以得到反向传播方程的形式为

$$\frac{\partial^2 E_n}{\partial a_j^2} = h'(a_j)^2 \sum_k \sum_{k'} w_{kj} w_{k'j} \frac{\partial^2 E_n}{\partial a_k \partial a_{k'}} + h''(a_j) \sum_k w_{kj} \frac{\partial E_n}{\partial a_k} \quad (5.80)$$

如果我们忽略二阶导数中非对角线元素，那么我们有 (Becker and LeCun, 1989; LeCun et al., 1990)

$$\frac{\partial^2 E_n}{\partial a_j^2} = h'(a_j)^2 \sum_k w_{kj}^2 \frac{\partial^2 E_n}{\partial a_k^2} + h'' \sum_k w_{kj} \frac{\partial E_n}{\partial a_k} \quad (5.81)$$

注意，需要计算这个近似，所需的计算步骤数为 $O(W)$ ，其中 W 是网络中权值和偏置的总数。对于原始的Hessian矩阵，计算的步骤数为 $O(W^2)$ 。

Ricotti et al. (1998) 也使用了Hessian矩阵的对角近似，但是他们在计算 $\frac{\partial^2 E_n}{\partial a_j^2}$ 时保留了所有的项，从而得到了对角项的精确的表达式。注意，这样做的计算复杂度不再是 $O(W)$ 。然而，对角近似的主要问题是，在实际应用中Hessian矩阵通常是强烈非对角化的，因此为了计算方便而采取的这些近似手段必须谨慎使用。

5.4.2 外积近似

当神经网络应用于回归问题时，通常使用下面形式的平方和误差函数

$$E = \frac{1}{2} \sum_{n=1}^N (y_n - t_n)^2 \quad (5.82)$$

为了记号的简洁，我们考虑单一输出的情形（推广到多个输出是很直接的）。这样，我们可以把Hessian矩阵写成下面的形式

$$\mathbf{H} = \nabla \nabla E = \sum_{n=1}^N \nabla y_n (\nabla y_n)^T + \sum_{n=1}^N (y_n - t_n) \nabla \nabla y_n \quad (5.83)$$

如果网络已经在数据集上训练过，输出 y_n 恰好非常接近 t_n ，那么公式 (5.83) 的第二项会很小，可以被忽略。然而，更一般的情况下，忽略这一项可能更合适，理由如下。回忆一下，根据1.5.5节的讨论，最小化平方和误差函数的最优函数是目标数据的条件平均。这样， $(y_n - t_n)$ 是一个零均值的随机变量。如果我们假设它的值与公式 (5.83) 右侧的二阶导数项无关，那么在对于 n 的求和项中，整个项的平均值将会等于零。

通过忽略公式 (5.83) 的第二项，我们就得到了Levenberg-Marquardt近似，或者称为外积近似 (outer product approximation)（因为此时Hessian矩阵由向量外积的求和构造出来），形式为

$$\mathbf{H} \simeq \sum_{n=1}^N \mathbf{b}_n \mathbf{b}_n^T \quad (5.84)$$

其中 $\mathbf{b}_n \equiv \nabla a_n = \nabla y_n$ ，因为输出单元的激活函数就是恒等函数。Hessian矩阵外积近似的计算是很容易的，因为它只涉及到误差函数的一阶导数，这可以通过使用标准的反向传播算法在 $O(W)$ 个步骤内高效地求出。通过简单的乘法，矩阵的元素可以在 $O(W^2)$ 个步骤内计算出。需要特别强调的一点是，这种近似只在网络被恰当地训练时才成立，对于一个一般的网络映射，公式 (5.83) 的右侧的二阶导数项通常不能忽略。

在误差函数为交叉熵误差函数，输出单元激活函数为logistic sigmoid函数的神经网络中，对应的近似为

$$\mathbf{H} \simeq \sum_{n=1}^N y_n (1 - y_n) \mathbf{b}_n \mathbf{b}_n^T \quad (5.85)$$

对于输出函数为softmax函数的多类神经网络，可以得到类似的结果。

5.4.3 Hessian矩阵的逆矩阵

使用外积近似，我们可以提出一个计算Hessian矩阵的逆矩阵的高效方法 (Hassibi and Stork, 1993)。首先，我们用矩阵的记号写出外积近似，即

$$\mathbf{H}_N = \sum_{n=1}^N \mathbf{b}_n \mathbf{b}_n^T \quad (5.86)$$

其中， $\mathbf{b}_n \equiv \nabla_{\mathbf{w}} a_n$ 是数据点 n 产生的输出单元激活对梯度的贡献。我们现在推导一个建立Hessian矩阵的顺序步骤，每次处理一个数据点。假设我们已经使用前 L 个数据点得到了Hessian矩阵的逆矩阵。通过将第 $L+1$ 个数据点的贡献单独写出来，我们有

$$\mathbf{H}_{L+1} = \mathbf{H}_L + \mathbf{b}_{L+1} \mathbf{b}_{L+1}^T \quad (5.87)$$

为了计算Hessian矩阵的逆矩阵，我们考虑下面的矩阵恒等式

$$(\mathbf{M} + \mathbf{v}\mathbf{v}^T)^{-1} = \mathbf{M}^{-1} - \frac{(\mathbf{M}^{-1}\mathbf{v})(\mathbf{v}^T\mathbf{M}^{-1})}{1 + \mathbf{v}^T\mathbf{M}^{-1}\mathbf{v}} \quad (5.88)$$

这个恒等式是Woodbury恒等式 (C.7) 的一个特例。如果我们令 $\mathbf{H}_L = \mathbf{M}$ ，且 $\mathbf{b}_{L+1} = \mathbf{v}$ ，我们有

$$\mathbf{H}_{L+1}^{-1} = \mathbf{H}_L^{-1} - \frac{\mathbf{H}_L^{-1}\mathbf{b}_{L+1}\mathbf{b}_{L+1}^T\mathbf{H}_L^{-1}}{1 + \mathbf{b}_{L+1}^T\mathbf{H}_L^{-1}\mathbf{b}_{L+1}} \quad (5.89)$$

使用这种方式，数据点可以依次使用，直到 $L+1=N$ ，整个数据集被处理完毕。于是，这个结果表示一个计算Hessian矩阵的逆矩阵的算法，这个算法只需对数据集扫描一次。最开始的矩阵 \mathbf{H}_0 被选为 $\alpha\mathbf{I}$ ，其中 α 是一个较小的量，从而算法实际找的是 $\mathbf{H} + \alpha\mathbf{I}$ 的逆矩阵。结果对于 α 的精确值不是特别敏感。将这个算法推广到多于一个输出的情形是很直接的。

这里，我们注意到，Hessian矩阵有时可以作为神经网络训练算法的一部分被间接计算。特别地，拟牛顿非线性优化算法在训练过程中逐步建立起Hessian矩阵的逆矩阵的近似。关于这种算法的详细讨论，可以参考Bishop and Nabney (2008)。

5.4.4 有限差

与误差函数的一阶导数的形式相同，我们可以使用有限差的方法求二阶导数，精度受数值计算的精度限制。如果我们对每对可能的权值施加一个扰动，那么我们有

$$\begin{aligned} \frac{\partial^2 E}{\partial w_{ji} \partial w_{lk}} &= \frac{1}{4\epsilon^2} \{E(w_{ji} + \epsilon, w_{lk} + \epsilon) - E(w_{ji} + \epsilon, w_{lk} - \epsilon) \\ &\quad - E(w_{ji} - \epsilon, w_{lk} + \epsilon) + E(w_{ji} - \epsilon, w_{lk} - \epsilon)\} + O(\epsilon^2) \end{aligned} \quad (5.90)$$

与之前一样，通过使用对称的中心差，我们确保了残留的误差项是 $O(\epsilon^2)$ 而不是 $O(\epsilon)$ 。由于在Hessian矩阵中有 W^2 个元素，且每个元素的计算需要四次正向传播过程，每个传播过程需要 $O(W)$ 次操作（每个模式），因此我们看到这种方法计算完整的Hessian矩阵需要 $O(W^3)$ 次操作。所以，这个方法的计算性质很差，虽然在实际应用中它对于检查反向传播算法的执行的正确性很有用。

一个更加高效的数值导数的方法是将中心差应用于一阶导数，而一阶导数可以通过反向传播方法计算。即

$$\frac{\partial^2 E}{\partial w_{ji} \partial w_{lk}} = \frac{1}{2\epsilon} \left\{ \frac{\partial E}{\partial w_{ji}}(w_{lk} + \epsilon) - \frac{\partial E}{\partial w_{ji}}(w_{lk} - \epsilon) \right\} + O(\epsilon^2) \quad (5.91)$$

由于只需要对 W 个权值施加扰动，且梯度可以通过 $O(W)$ 次计算得到，因此我们看到这种方法可以在 $O(W^2)$ 次操作内得到Hessian矩阵。

5.4.5 Hessian矩阵的精确计算

目前为止，我们已经研究了各种计算Hessian矩阵或者逆矩阵的近似方法。对于一个任意的前馈拓扑结构的网络，Hessian矩阵也可以精确地计算。计算的方法是使用反向传播算法计算一阶导数的推广，同时也保留了计算一阶导数的方法的许多良好的性质，包括计算效率 (Bishop, 1991; Bishop, 1992)。这种方法可以应用于任何可微的可以表示成网络输出的函数形式的误差函数，以及任何具有可微的激活函数的神经网络。计算Hessian矩阵所需的计算步骤为 $O(W^2)$ 。类似的算法也可以参考Buntine and Weigend (1993)。

这里我们考虑一个具体的情况，即具有两层权值的网络。这种网络中待求的方程很容易推导。我们将使用下标 i 和 i' 表示输入，用下标 j 和 j' 表示隐含单元，用下标 k 和 k' 表示输出。首先我们定义

$$\delta_k = \frac{\partial E_n}{\partial a_k}, \quad M_{kk'} \equiv \frac{\partial^2 E_n}{\partial a_k \partial a_{k'}} \quad (5.92)$$

其中 E_n 是数据点 n 对误差函数的贡献。于是，这个网络的Hessian矩阵可以被看成三个独立的模块，即

- 两个权值都在第二层。

$$\frac{\partial^2 E_n}{\partial w_{kj}^{(2)} \partial w_{k'j'}^{(2)}} = z_j z_{j'} M_{kk'} \quad (5.93)$$

- 两个权值都在第一层。

$$\begin{aligned} \frac{\partial^2 E_n}{\partial w_{ji}^{(1)} \partial w_{j'i'}^{(1)}} &= x_i x_{i'} h''(a_{j'}) I_{jj'} \sum_k w_{kj'}^{(2)} \delta_k \\ &+ x_i x_{i'} h'(a_{j'}) h'(a_j) \sum_k \sum_{k'} w_{k'j'}^{(2)} w_{kj}^{(2)} M_{kk'} \end{aligned} \quad (5.94)$$

- 每一层有一个权值。

$$\frac{\partial^2 E_n}{\partial w_{ji}^{(1)} \partial w_{kj'}^{(2)}} = x_i h'(a_j) \left\{ \delta_k I_{j'j} + z_{j'} \sum_{k'} w_{k'j'}^{(2)} M_{kk'} \right\} \quad (5.95)$$

这里 $I_{jj'}$ 是单位矩阵的第 j, j' 个元素。如果权值中的一个或者两个是偏置项，那么只需将激活设为1即可得到对应的表达式。很容易将这个结果推广到允许网络包含跨层链接的情形。

5.4.6 Hessian矩阵的快速乘法

对于Hessian矩阵的许多应用来说，我们感兴趣的不是Hessian矩阵 \mathbf{H} 本身，而是 \mathbf{H} 与某些向量 \mathbf{v} 的乘积。我们已经看到Hessian矩阵的计算需要 $O(W^2)$ 次操作，所需的存储空间也是 $O(W^2)$ 。但是，我们想要计算的向量 $\mathbf{v}^T \mathbf{H}$ 只有 W 个元素。因此，我们可以不把计算Hessian矩阵当成一个中间的步骤，而是可以尝试寻找一种只需 $O(W)$ 次操作的直接计算 $\mathbf{v}^T \mathbf{H}$ 的高效方法。

为了完成这一点，我们首先注意到

$$\mathbf{v}^T \mathbf{H} = \mathbf{v}^T \nabla(\nabla E) \quad (5.96)$$

其中 ∇ 表示权空间的梯度算符。然后，我们可以写下计算 ∇E 的标准正向传播和反向传播的方程，然后将公式 (5.96) 应用于这些方程，得到一组计算 $\mathbf{v}^T \mathbf{H}$ 的正向传播和反向传播的方程 (Møller, 1993; Pearlmutter, 1994)。这对应于将微分算符 $\mathbf{v}^T \nabla$ 作用于原始的正向传播和反向传播的方程。Pearlmutter (1994) 使用记号 $\mathcal{R}\{\cdot\}$ 表示算符 $\mathbf{v}^T \nabla$ ，我们将遵从这个惯例。下面的分析过程很直接，我们会使用通常的微积分规则，以及下面的结果

$$\mathcal{R}\{\mathbf{w}\} = \mathbf{v} \quad (5.97)$$

我们会使用一个简单的例子来说明这个方法。与之前一样，我们使用图5.1所示的两层网络，以及线性的输出单元和平方和误差函数。我们考虑数据集里的一个模式对于误差函数的贡献。这样，我们所要求解的向量可以通过求出每个模式各自的贡献然后求和的方式得到。对于两层神经网络，正向传播方程为

$$a_j = \sum_i w_{ji} x_i \quad (5.98)$$

$$z_j = h(a_j) \quad (5.99)$$

$$y_k = \sum_j w_{kj} z_j \quad (5.100)$$

我们现在使用 $\mathcal{R}\{\cdot\}$ 算符作用于这些方程上，得到一组正向传播方程，形式为

$$\mathcal{R}\{a_j\} = \sum_i v_{ji} x_i \quad (5.101)$$

$$\mathcal{R}\{z_j\} = h'(a_j) \mathcal{R}\{a_j\} \quad (5.102)$$

$$\mathcal{R}\{y_k\} = \sum_j w_{kj} \mathcal{R}\{z_j\} + \sum_j v_{kj} z_j \quad (5.103)$$

其中， v_{ji} 是向量 \mathbf{v} 中对应于权值 w_{ji} 的元素。 $\mathcal{R}\{z_j\}$, $\mathcal{R}\{a_j\}$ 和 $\mathcal{R}\{y_k\}$ 可以被看做新的变量，它的值可以使用上面的方程得到。

由于我们考虑的时平方和误差函数，因此我们有下面的标准的反向传播表达式

$$\delta_k = y_k - t_k \quad (5.104)$$

$$\delta_j = h'(a_j) \sum_k w_{kj} \delta_k \quad (5.105)$$

与之前一样，我们将 $\mathcal{R}\{\cdot\}$ 算符作用于这些方程上，得到一组反向传播方程，形式为

$$\mathcal{R}\{\delta_k\} = \mathcal{R}\{y_k\} \quad (5.106)$$

$$\begin{aligned} \mathcal{R}\{\delta_j\} &= h''(a_j) \mathcal{R}\{a_j\} \sum_k w_{kj} \delta_k \\ &\quad + h'(a_j) \sum_k v_{kj} \delta_k + h'(a_j) \sum_k w_{kj} \mathcal{R}\{\delta_k\} \end{aligned} \quad (5.107)$$

最后，我们有误差函数的一阶导数的方程

$$\frac{\partial E}{\partial w_{kj}} = \delta_k z_j \quad (5.108)$$

$$\frac{\partial E}{\partial w_{ji}} = \delta_j x_i \quad (5.109)$$

使用 $\mathcal{R}\{\cdot\}$ 算符作用在这些方程上，我们得到了下面的关于 $\mathbf{v}^T \mathbf{H}$ 的表达式

$$\mathcal{R}\left\{\frac{\partial E}{\partial w_{kj}}\right\} = \mathcal{R}\{\delta_k\} z_j + \delta_k \mathcal{R}\{z_j\} \quad (5.110)$$

$$\mathcal{R}\left\{\frac{\partial E}{\partial w_{ji}}\right\} = x_i \mathcal{R}\{\delta_j\} \quad (5.111)$$

算法的执行涉及到将新的变量 $\mathcal{R}\{a_j\}$, $\mathcal{R}\{z_j\}$ 和 $\mathcal{R}\{\delta_j\}$ 引入到隐含单元，将 $\mathcal{R}\{\delta_k\}$ 和 $\mathcal{R}\{y_k\}$ 引入到输出单元。对于每个输入模式，这些量的值可以使用上面的结果求出， $\mathbf{v}^T \mathbf{H}$ 的元素的值由公式 (5.110) 和公式 (5.111) 给出。这种方法的一个好处是，计算 $\mathbf{v}^T \mathbf{H}$ 的方程与标准的正向传播和反向传播的方程相同，因此将现有的神经网络计算程序扩展到能够计算这个乘积通常很容易。

如果必要的话，这个方法可以用来计算完整的Hessian矩阵。计算的方法为：将向量 \mathbf{v} 选为一系列的形如 $(0, 0, \dots, 1, \dots, 0)$ 的单位向量，每个单位向量选出Hessian矩阵中的一列。这种方法的数学形式与Bishop (1992) 的反向传播算法等价，如5.4.5节所述。但是这种方法由于冗余的计算的存在，会损失一定的计算效率。

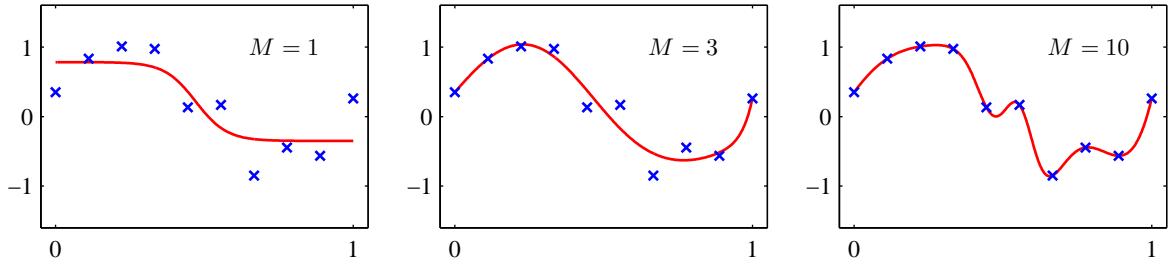


图 5.9: 使用从正弦数据集中抽取的 10 个数据点训练的两层神经网络的例子。各图分别给出了使用 $M = 1, 3, 10$ 个隐含单元调节网络的结果，调节的方法是使用放缩的共轭梯度算法来最小化平方和误差函数。

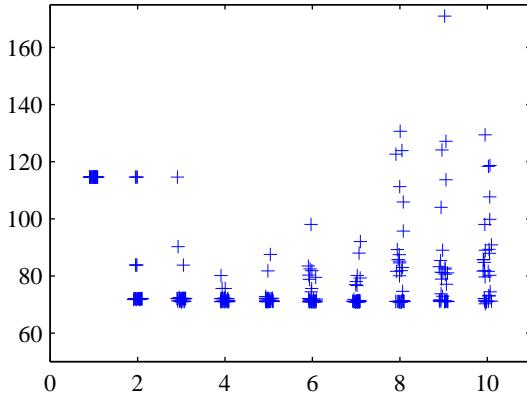


图 5.10: 对于多项式数据集，测试集的平方和误差与网络的隐含单元的数量的图像。对于每个网络规模，都随机选择了 30 个初始点，这展示了局部最小值的效果。对于每个新的初始点，权向量通过从一个各向同性的高斯分布中取样，这个高斯分布的均值为零，方差为 10。

5.5 神经网络的正则化

神经网络的输入单元和输出单元的数量通常由数据集的维度确定，而隐含单元的数量 M 是一个自由的参数，可以通过调节来给出最好的预测性能。注意， M 控制了网络中参数（权值和偏置）的数量，因此我们可以猜想，在最大似然的框架下，会存在一个泛化性能最好的最优的 M 值，这个值对应于拟合效果不好和过拟合之间的最优平衡。图 5.9 给出了不同的 M 值对于正弦曲线回归问题的效果。

然而，泛化误差与 M 的关系不是一个简单的函数关系，因为误差函数中存在局部极小值，如图 5.10 所示。这里，我们看到了对于不同的 M 值，权值的多次随机初始化的效果。在验证集上的整体最优表现出现于 $M = 8$ 的情况下的某个特定的解。在实际应用中，一种选择 M 的方法实际上是画一张类似图 5.10 的图，然后选择有最小验证集误差的具体的解。

然而，有其他的方式控制神经网络的模型复杂度来避免过拟合。根据我们第 1 章中对多项式曲线拟合问题的讨论，我们看到，一种方法是选择一个相对大的 M 值，然后通过给误差函数增加一个正则化项，来控制模型的复杂度。最简单的正则化项是二次的，给出了正则化的误差函数，形式为

$$\tilde{E}(\mathbf{w}) = E(\mathbf{w}) + \frac{\lambda}{2} \mathbf{w}^T \mathbf{w} \quad (5.112)$$

这个正则化项也被称为权值衰减（weight decay），已经在第 3 章中详细讨论过了。这样，模型复杂度可以通过选择正则化系数 λ 来确定。正如我们之前看到的那样，正则化项可以表示为权值 \mathbf{w} 上的零均值高斯先验分布的负对数。

5.5.1 相容的高斯先验

公式 (5.112) 给出的简单权值衰减的一个局限性是，它与网络映射的确定缩放性质不相容。为了说明这一点，考虑一个多层次感知器网络，这个网络有两层权值和线性输出单元，它给出了从输入变量集合 $\{x_i\}$ 到输出变量集合 $\{y_k\}$ 的映射。第一个隐含层的隐含单元的激活的形式为

$$z_j = h \left(\sum_i w_{ji} x_i + w_{j0} \right) \quad (5.113)$$

输出单元的激活为

$$y_k = \sum_j w_{kj} z_j + w_{k0} \quad (5.114)$$

假设我们对输入变量进行一个线性变换，形式为

$$x_i \rightarrow \tilde{x}_i = ax_i + b \quad (5.115)$$

然后我们可以根据这个映射对网络进行调整，使得网络给出的映射不变。调整的方法为，对从输入单元到隐含层单元的权值和偏置也进行一个对应的线性变换，形式为

$$w_{ji} \rightarrow \tilde{w}_{ji} = \frac{1}{a} w_{ji} \quad (5.116)$$

$$w_{j0} \rightarrow \tilde{w}_{j0} = w_{j0} - \frac{b}{a} \sum_i w_{ji} \quad (5.117)$$

类似地，网络的输出变量的线性变换

$$y_k \rightarrow \tilde{y}_k = cy_k + d \quad (5.118)$$

通过对第二层的权值和偏置进行线性变换的方式实现。变换的形式为

$$w_{kj} \rightarrow \tilde{w}_{kj} = cw_{kj} \quad (5.119)$$

$$w_{k0} \rightarrow \tilde{w}_{k0} = cw_{k0} + d \quad (5.120)$$

如果我们使用原始数据训练一个网络，还使用输入和（或）目标变量进行了上面的线性变换的数据训练一个网络，那么相容性要求这两个网络应该是等价的，差别仅在于上面给出的权值的线性变换。任何正则化项都应该与这个性质相容，否则模型就会倾向于选择某个解，而忽视某个等价的解。显然，简单的权值衰减 (5.112) 由于把所有的权值和偏置同等对待，因此不满足这个性质。

于是我们要寻找一个正则化项，它在线性变换 (5.116)、(5.117)、(5.119) 和 (5.120) 下具有不变性。这需要正则化项应该对于权值的重新缩放不变，对于偏置的平移不变。这样的正则化项为

$$\frac{\lambda_1}{2} \sum_{w \in \mathcal{W}_1} w^2 + \frac{\lambda_2}{2} \sum_{w \in \mathcal{W}_2} w^2 \quad (5.121)$$

其中 \mathcal{W}_1 表示第一层的权值集合， \mathcal{W}_2 表示第二层的权值集合，偏置未出现在求和式中。这个正则化项在权值的变换下不会发生变化，只要正则化参数进行下面的重新放缩即可： $\lambda_1 \rightarrow a^{\frac{1}{2}} \lambda_1$ 和 $\lambda_2 \rightarrow c^{-\frac{1}{2}} \lambda_2$ 。

正则化项 (5.121) 对应于下面形式的先验概率分布。

$$p(\mathbf{w} | \alpha_1, \alpha_2) \propto \exp \left(-\frac{\alpha_1}{2} \sum_{w \in \mathcal{W}_1} w^2 - \frac{\alpha_2}{2} \sum_{w \in \mathcal{W}_2} w^2 \right) \quad (5.122)$$

注意，这种形式的先验是反常的 (improper)（不能够被归一化），因为偏置参数没有限制。使用反常先验会给正则化系数的选择造成很大的困难，也会给贝叶斯框架下的模型选择造成很

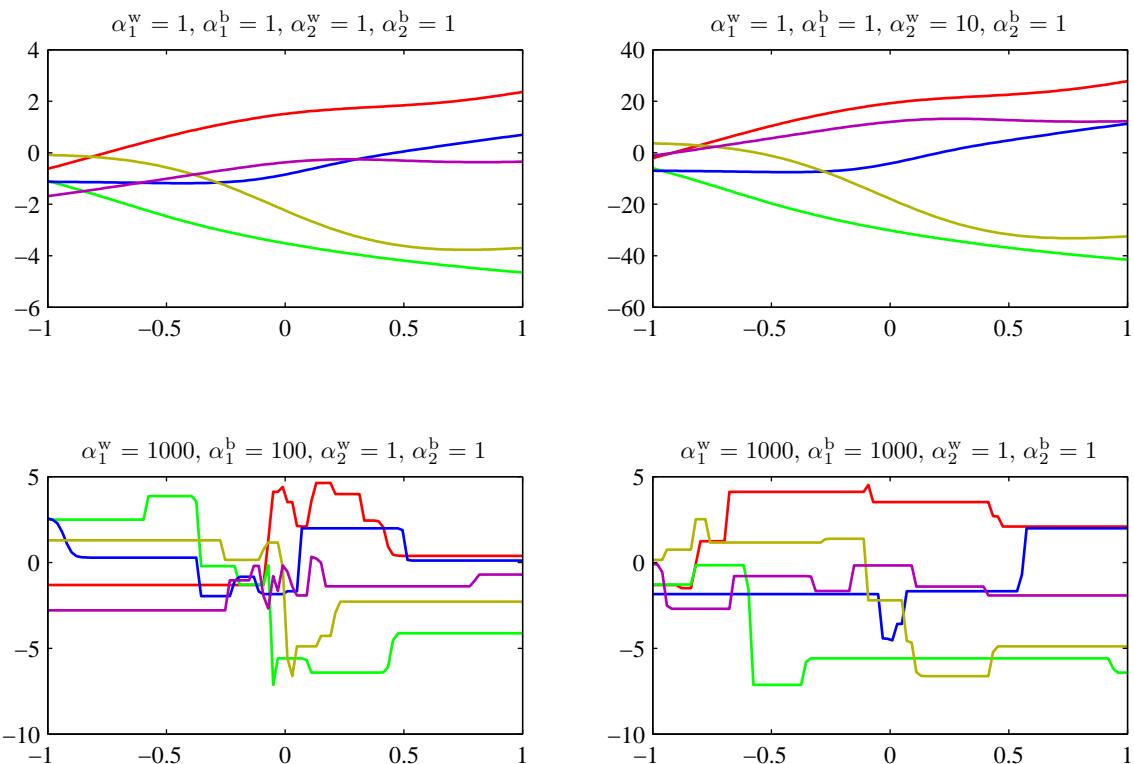


图 5.11: 控制两层神经网络的权值和偏置的先验概率分布的超参数的效果说明。其中，神经网络有一个输入，一个线性输出，以及12个隐含结点，隐含结点的激活函数为tanh。先验概率分布通过四个超参数 $\alpha_1^b, \alpha_1^w, \alpha_2^b, \alpha_2^w$ 控制，它们分别表示第一层的偏置、第一层的权值、第二层的偏置、第二层的权值。我们看到，参数 α_2^w 控制函数的垂直标度（注意上方两张图的垂直轴的标度不同）， α_1^w 控制函数值变化的水平标度， α_1^b 控制变化发生的水平范围。参数 α_2^b ，它的效果没有在这里说明，它控制了函数的垂直偏置的范围。

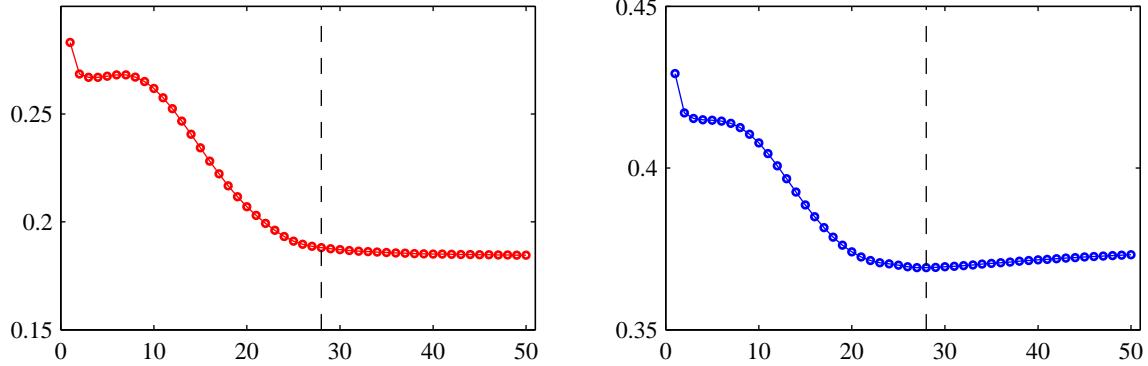


图 5.12: 训练集误差 (左图) 和验证集误差 (右图) 在典型的训练阶段的行为说明。图像给出了误差与迭代次数的函数，数据集为正弦数据集。得到最好的泛化表现的目标表明，训练应该在垂直虚线表示的点处停止，对应于验证集误差的最小值。

大的困难，因为对应的模型证据等于零。因此，通常的做法是单独包含一个有着自己单独的一套超参数的偏置的先验（这就破坏了平移不变性）。为了说明四个参数的效果，我们从先验中抽取样本，然后画出了对应的神经网络函数，如图5.11所示。

更一般地，我们可以考虑权值被分为任意数量的组 \mathcal{W}_k 情况下的先验，即

$$p(\mathbf{w}) \propto \exp\left(-\frac{1}{2} \sum_k \alpha_k \|\mathbf{w}\|_k^2\right) \quad (5.123)$$

其中

$$\|\mathbf{w}\|_k^2 = \sum_{j \in \mathcal{W}_k} w_j^2 \quad (5.124)$$

作为这种形式的先验的一个特殊情况，如果我们将每个输入单元关联的权值设为一个分组，并且关于对应的参数 α_k 最优化边缘似然函数，那么我们就得到了将在7.2.2节讨论的自动相关性确定 (automatic relevance determination) 的方法。

5.5.2 早停止

另一种控制网络的复杂度的正则化方法是早停止 (early stopping)。非线性网络模型的训练对应于误差函数的迭代减小，其中误差函数是关于训练数据集定义的。对于许多用于网络训练的最优化算法（例如共轭梯度法），误差函数是一个关于迭代次数的不增函数。然而，在独立数据（通常被称为验证集）上测量的误差，通常首先减小，接下来由于模型开始过拟合而逐渐增大。于是，训练过程可以在关于验证集误差最小的点停止，如图5.12所示。这样可以得到一个有着较好泛化性能的网络。

这种情况下，网络的行为有时可以通过网络的自由度有效数量来定量描述。自由度有效数量开始时很小，然后在训练过程中增长，对应于模型复杂度的持续增长。这样，在训练误差达到最小值之前停止训练就表示了一种限制模型复杂度的方式。

在二次误差函数的情况下，我们可以说明这种直观的描述，并且说明早停止的效果与使用简单的权值衰减的正则化项的效果类似。这可以通过图5.13来理解。图5.13中，权值空间的坐标轴已经进行了旋转，使得坐标轴平行于Hessian矩阵的特征向量。在没有权值衰减的情况下，如果权向量开始于原点，然后在训练过程中沿着局部负梯度向量确定的路径移动，那么权向量从最开始平行于 w_2 轴的位置，移动到大致对应于 $\tilde{\mathbf{w}}$ 的位置，然后移向最小化误差函数的位置 \mathbf{w}_{ML} 。这可以从误差曲面和Hessian矩阵的特征值得出。于是，在点 $\tilde{\mathbf{w}}$ 附近停止就类似于权值衰减。早停止和权值衰减之间的关系可以定量描述，因此说明了 $\tau\eta$ （其中 τ 是迭代次数， η 是学习率参数）扮演了正则化参数 λ 的倒数的角色。于是网络中有效参数的数量会在训练过程中增长。

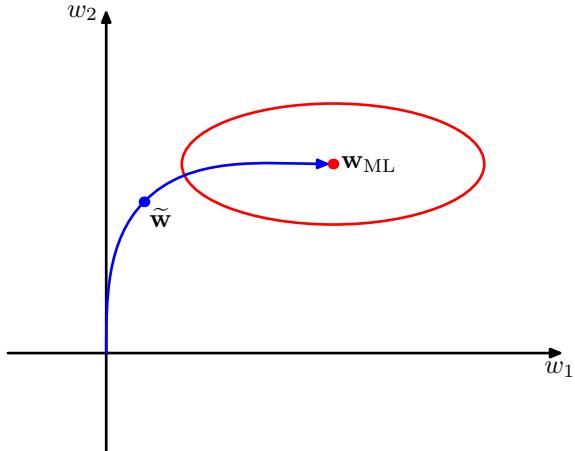


图 5.13: 在二次误差函数的情况下, 关于早停止可以给出与权值衰减类似的结果的原因说明。椭圆给出了常数误差函数的轮廓线, w_{ML} 表示误差函数的最小值。如果权向量的起始点为原点, 按照局部负梯度的方向移动, 那么它会沿着曲线给出的路径移动。通过对训练过程早停止, 我们找到了一个权值向量 \tilde{w} 。定性地说, 它类似于使用简单的权值衰减正则化项, 然后最小化正则化误差函数的方法得到的权值。通过与图 5.15 进行对比, 我们可以看到这一点。

5.5.3 不变性

在许多模式识别的应用中, 在对于输入变量进行了一个或者多个变换之后, 预测不应该发生变化, 或者说应该具有不变性 (invariant)。例如, 在二维图像 (例如手写数字) 的分类问题中, 一个特定的图像的类别应该与图像的位置无关 (平移不变性 (translation invariance)), 也应该与图像的大小无关 (缩放不变性 (scale invariance))。这样的变换对于原始数据 (用图像的每个像素的灰度值表示) 产生了巨大的改变, 但是分类系统还是应该给出同样的输出。类似地, 在语音识别中, 对于时间轴的微小的非线性变形 (保持了时间顺序) 不应该改变信号的意义。

如果可以得到足够多的训练模式, 那么可调节的模型 (例如神经网络) 可以学习到不变性, 至少可以近似地学习到。这涉及到在训练集里包含足够多的表示各种变换的效果的样本。因此, 对于一个图像的平移不变性, 训练集应该包含图像出现在多个不同位置的数据。

但是, 如果训练样本数受限, 或者有多个不变性 (变换的组合的数量随着变换的数量指数增长), 那么这种方法就很不实用。于是, 我们要寻找另外的方法来让可调节的模型能够表述所需的不变性。这些方法大致可以分为四类。

- 通过复制训练模式, 同时根据要求的不变性进行变换, 对训练集进行扩展。例如, 在手写数字识别的例子中, 我们可以将每个样本复制多次, 每个复制后的样本中, 图像被平移到了不同的位置。
- 为误差函数加上一个正则化项, 用来惩罚当输入进行变换时, 输出发生的改变。这引出了 5.5.4 节讨论的切线传播 (tangent propagation) 方法。
- 通过抽取在要求的变换下不发生改变的特征, 不变性被整合到预处理过程中。任何后续的使用这些特征作为输入的回归或者分类系统就会具有这些不变性。
- 最后一种方法是把不变性的性质整合到神经网络的构建过程中, 或者对于相关向量机的方法, 整合到核函数中。一种方法是通过使用局部接收场和共享权值, 正如 5.5.6 节在卷积神经网络中讨论的那样。

方法 1 通常实现起来相对简单, 并且可以用来处理复杂的不变性, 如图 5.14 所示。对于顺序训练算法, 可以这样做: 在模型观测到输入模式之前, 对每个输入模式进行变换, 从而使得如果模式被循环处理, 那么每次都会接收到一个不同的变换 (从一个适当的概率分布中抽取)。对于批处理方法, 可以将每个数据点复制多次, 然后独立地变换每个副本, 这样可以产生类似的效果。



图 5.14: 对手写数字进行人工形变的说明。原始图像见左图。在右图中，上面一行给出了三个经过了形变的数字，对应的位移场在下面一行给出。这些位移场按照下面的方法生成：在每个像素处，对唯一 $\Delta x, \Delta y \in (0, 1)$ 进行随机取样，然后分别与宽度为0.01, 30, 60的高斯分布做卷积，进行平滑。

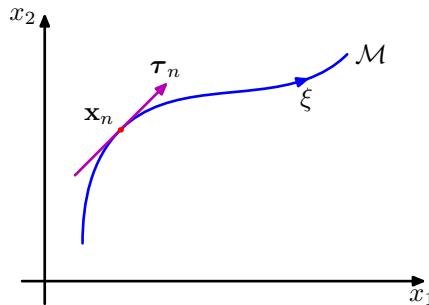


图 5.15: 二维输入空间的例子，展示了在一个特定的输入向量 x_n 上的连续变换的效果。一个参数为连续变量 ξ 的一维变换作用于 x_n 上会使它扫过一个一维流形 M 。局部来看，变换的效果可以用切向量 τ_n 来近似。

果。使用这些扩展后的数据可以大幅提升泛化能力 (Simard et al., 2003)，虽然计算开销比较大。

方法2保持了数据集的不变性，而是给误差函数增加了一个正则化项。在5.5.5节，我们会看到方法1与方法2关系密切。

方法3的一个优点是，对于训练集里没有包含的变换，它可以正确地进行外插。然而，找到符合要求的人工设计的特征很困难，因为这种特征要具有所需的不变性，还不能丢失对于判别很有帮助的信息。

5.5.4 切线传播

通过切线传播 (tangent propagation) 的方法，我们可以使用正则化来让模型对于输入的变换具有不变性 (Simard et al., 1992)。对于一个特定的输入向量 x_n ，考虑变换产生的效果。假设变换是连续的（例如平移或者旋转，而不是镜像翻转），那么变换的模式会扫过 D 维输入空间的一个流形 M 。图5.15说明了 $D = 2$ 的情形。假设变换由单一参数 ξ 控制（例如， ξ 可能是旋转的角度）。那么被 x_n 扫过的子空间 M 是一维的，并且以 ξ 为参数。令这个变换作用于 x_n 上产生的向量为 $s(x_n, \xi)$ ，且 $s(x, 0) = x$ 。这样曲线 M 的切线就由方向导数 $\tau = \frac{\partial s}{\partial \xi}$ 给出，且点 x_n 处的切线向量为

$$\tau_n = \left. \frac{\partial s(x_n, \xi)}{\partial \xi} \right|_{\xi=0} \quad (5.125)$$

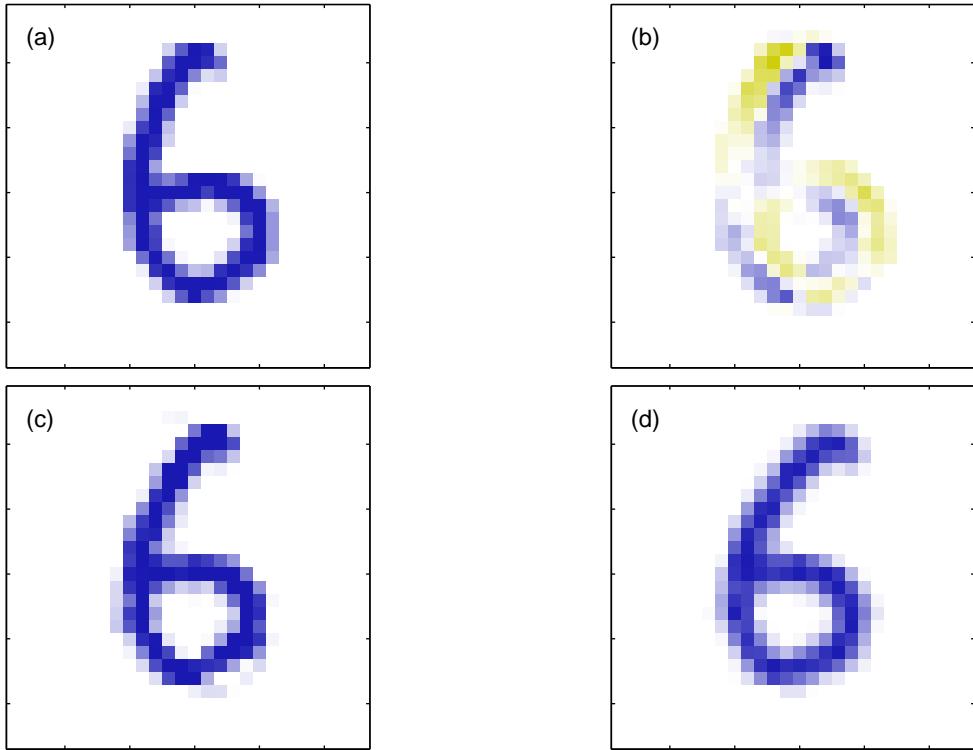


图 5.16: (a) 原始的手写数字 x , (b) 对应于无穷小顺时针旋转的切向量 τ , 其中蓝色和黄色分别对应于正值和负值, (c) 将来自这个切向量的微小贡献作用于原始图像的结果, 得到了 $x + \epsilon\tau$, 其中 $\epsilon = 15$ 度。 (d) 真实的图像旋转, 用作对比。

对于输入向量进行变换之后, 网络的输出通常会发生变化。输出 k 关于 ξ 的导数为

$$\frac{\partial y_k}{\partial \xi} \Big|_{\xi=0} = \sum_{i=1}^D \frac{\partial y_k}{\partial x_i} \frac{\partial x_i}{\partial \xi} \Big|_{\xi=0} = \sum_{i=1}^D J_{ki} \tau_i \quad (5.126)$$

其中 J_{ki} 为 Jacobian 矩阵 J 的第 (k, i) 个元素, 正如 5.3.4 节讨论的那样。公式 (5.126) 给出的结果可以用于修改标准的误差函数, 使得在数据点的邻域之内具有不变性。修改的方法为: 给原始的误差函数 E 增加一个正则化函数 Ω , 得到下面形式的误差函数

$$\tilde{E} = E + \lambda \Omega \quad (5.127)$$

其中 λ 是正则化系数, 且

$$\Omega = \frac{1}{2} \sum_n \sum_k \left(\frac{\partial y_{nk}}{\partial \xi} \Big|_{\xi=0} \right)^2 = \frac{1}{2} \sum_n \sum_k \left(\sum_{i=1}^D J_{nki} \tau_{ni} \right)^2 \quad (5.128)$$

当网络映射函数在每个模式向量的邻域内具有变换不变性时, 正则化函数等于零。 λ 的值确定了训练数据和学习不变性之间的平衡。

在实际执行过程中, 切线向量 τ_n 可以使用有限差近似, 即将原始向量 x_n 从使用了小的 ξ 进行变换后的对应的向量中减去, 再除以 ξ 。图 5.16 说明了这个过程。

正则化函数通过 Jacobian 矩阵 J 对网络的权值产生依赖。通过对 5.3 节中讨论的方法进行推广, 计算正则化项关于网络权值的导数的反向传播公式可以很容易地得到。

如果变换由 L 个参数控制 (例如, 对于二维图像的平移变换与面内旋转变换项结合), 那么流形 M 的维度为 L , 对应的正则化项由形如公式 (5.128) 的项求和得到, 每个变换都对应求和式中的一项。如果同时考虑若干个变换, 并且让网络映射对于每个变换分别具有不变性, 那么对于变换的组合来说就会具有 (局部) 不变性 (Simard et al., 1992)。

一个相关的技术, 被称为切线距离 (tangent distance), 可以用来构造基于距离的方法 (例如最近邻分类器) 的不变性 (Simard et al., 1993)。

5.5.5 用变换后的数据训练

我们已经看到，让模型对于一组变换具有不变性的一种方法是使用原始输入模式的变换后的模式来扩展训练集。这里，我们会说明，这种方法与切线传播的方法密切相关 (Bishop, 1995b; Leen, 1995)。

与5.5.4节一样，我们要考虑由单一参数 ξ 控制的变换，且这个变换由函数 $s(\mathbf{x}, \xi)$ 描述，其中 $s(\mathbf{x}, 0) = \mathbf{x}$ 。我们也会考虑平方和误差函数。对于未经过变换的输入，误差函数可以写成（在无限数据集的极限情况下）

$$E = \frac{1}{2} \iint \{y(\mathbf{x}) - t\}^2 p(t | \mathbf{x}) p(\mathbf{x}) d\mathbf{x} dt \quad (5.129)$$

正如1.5.5节讨论的那样。这里，为了保持记号的简洁，我们考虑有一个输出单元的网络。如果我们现在考虑每个数据点的无穷多个副本，每个副本都由一个变换施加了扰动，这个变换的参数为 ξ ，且 ξ 服从概率分布 $p(\xi)$ ，那么在这个扩展的误差函数上定义的误差函数可以写成

$$\tilde{E} = \frac{1}{2} \iiint \{y(s(\mathbf{x}, \xi)) - t\}^2 p(t | \mathbf{x}) p(\mathbf{x}) p(\xi) d\mathbf{x} dt d\xi \quad (5.130)$$

我们现在假设分布 $p(\xi)$ 的均值为零，方差很小，即我们只考虑对原始输入向量的小的变换。我们可以对变换函数进行关于 ξ 的展开，可得

$$\begin{aligned} s(\mathbf{x}, \xi) &= s(\mathbf{x}, 0) + \xi \frac{\partial}{\partial \xi} s(\mathbf{x}, \xi) \Big|_{\xi=0} + \frac{\xi^2}{2} \frac{\partial^2}{\partial \xi^2} s(\mathbf{x}, \xi) \Big|_{\xi=0} + O(\xi^3) \\ &= \mathbf{x} + \xi \boldsymbol{\tau} + \frac{1}{2} \xi^2 \boldsymbol{\tau}' + O(\xi^3) \end{aligned}$$

其中 $\boldsymbol{\tau}'$ 表示 $s(\mathbf{x}, \xi)$ 关于 ξ 的二阶导数在 $\xi = 0$ 处的值。这使得我们可以展开模型函数，可得

$$y(s(\mathbf{x}, \xi)) = y(\mathbf{x}) + \xi \boldsymbol{\tau}^T \nabla y(\mathbf{x}) + \frac{\xi^2}{2} [(\boldsymbol{\tau}')^T \nabla y(\mathbf{x}) + \boldsymbol{\tau}^T \nabla \nabla y(\mathbf{x}) \boldsymbol{\tau}] + O(\xi^3)$$

代入平均误差函数 (5.130)，我们有

$$\begin{aligned} \tilde{E} &= \frac{1}{2} \iint \{y(\mathbf{x}) - t\}^2 p(t | \mathbf{x}) p(\mathbf{x}) d\mathbf{x} dt \\ &\quad + \mathbb{E}[\xi] \iint \{y(\mathbf{x}) - t\} \boldsymbol{\tau}^T \nabla y(\mathbf{x}) p(t | \mathbf{x}) p(\mathbf{x}) d\mathbf{x} dt \\ &\quad + \mathbb{E}[\xi^2] \frac{1}{2} \iint \left[\{y(\mathbf{x}) - t\} \{(\boldsymbol{\tau}')^T \nabla y(\mathbf{x}) + \boldsymbol{\tau}^T \nabla \nabla y(\mathbf{x}) \boldsymbol{\tau}\} \right. \\ &\quad \left. + (\boldsymbol{\tau}^T \nabla y(\mathbf{x}))^2 \right] p(t | \mathbf{x}) p(\mathbf{x}) d\mathbf{x} dt + O(\xi^3) \end{aligned}$$

由于变换的分布的均值为零，因此我们有 $\mathbb{E}[\xi] = 0$ 。并且，我们把 $\mathbb{E}[\xi^2]$ 记作 λ 。省略 $O(\xi^3)$ 项，这样平均误差函数就变成了

$$\tilde{E} = E + \lambda \Omega \quad (5.131)$$

其中 E 是原始的平方和误差，正则化项 Ω 的形式为

$$\begin{aligned} \Omega &= \frac{1}{2} \int \left[\{y(\mathbf{x}) - \mathbb{E}[t | \mathbf{x}]\} \{(\boldsymbol{\tau}')^T \nabla y(\mathbf{x}) + \boldsymbol{\tau}^T \nabla \nabla y(\mathbf{x}) \boldsymbol{\tau}\} \right. \\ &\quad \left. + (\boldsymbol{\tau}^T \nabla y(\mathbf{x}))^2 \right] p(\mathbf{x}) d\mathbf{x} \end{aligned} \quad (5.132)$$

其中我们已经对 t 进行了积分。

我们可以进一步简化这个正则化项，如下所述。在1.5.5节，我们已经看到，使平方和误差函数达到最小值的函数为目标值 t 的条件均值 $\mathbb{E}[t | \mathbf{x}]$ 。根据公式（5.131），我们看到正则化的误差函数等于非正则化的误差函数加上一个 $O(\xi^2)$ 的项，因此最小化总误差函数的网络函数的形式为

$$y(\mathbf{x}) = \mathbb{E}[t | \mathbf{x}] + O(\xi^2) \quad (5.133)$$

从而，正则化项中的第一项消失，剩下的项为

$$\Omega = \frac{1}{2} \int (\boldsymbol{\tau}^T \nabla y(\mathbf{x}))^2 p(\mathbf{x}) d\mathbf{x} \quad (5.134)$$

这等价于切线传播的正则化项（5.128）。

如果我们考虑一个特殊情况，即输入变量的变换只是简单地添加随机噪声，从而 $\mathbf{x} \rightarrow \mathbf{x} + \boldsymbol{\xi}$ ，那么正则化项的形式为

$$\Omega = \frac{1}{2} \int \|\nabla y(\mathbf{x})\|^2 p(\mathbf{x}) d\mathbf{x} \quad (5.135)$$

这被称为Tikhonov正则化（Tikhonov and Arsenin, 1977; Bishop, 1995b）。这个正则化项关于网络权值的导数可以使用扩展的反向传播算法求出（Bishop, 1993）。我们看到，对于小的噪声，Tikhonov正则化与对输入添加随机噪声有关系。可以证明，在恰当的情况下，这种做法会提升模型的泛化能力。

5.5.6 卷积神经网络

另一种构造对输入变量的变换具有不变性的模型的方法是将不变性的性质融入到神经网络结构的构建中。这是卷积神经网络（convolutional neural network）（LeCun et al., 1989; LeCun et al., 1998）的基础，它被广泛地应用于图像处理领域。

考虑手写数字识别这个具体的任务。每个输入图像由一组像素的灰度值组成，输出为10个数字类别的后验概率分布。我们知道，数字的种类对于平移、缩放以及（微小的）旋转具有不变性。此外，网络还必须对一些更微妙的变换具有不变性，例如图5.14所示的弹性形变。一种简单的方法是把图像作为一个完全链接的神经网络的输入，例如图5.1所示的网络。假设数据集充分大，那么这样的网络原则上可以产生这个问题的一个较好的解，从而可以从样本中学习到恰当的不变性。

然而，这种方法忽略了图像的一个关键性质，即距离较近的像素的相关性要远大于距离较远的像素的相关性。计算机视觉领域中，许多现代的方法通过抽取只依赖于图像里小的子区域的局部特征的方式利用这个性质。之后，来自这些特征的信息就可以融合到后续处理阶段中，来检测更高级的特征，最后产生图像整体的信息。并且，对于图像的一个区域有用的局部特征可能对于图像的其他区域也有用，例如感兴趣的物体发生平移的情形。

这些想法被整合到了卷积神经网络中，通过下面三种方式：（1）局部接收场，（2）权值共享，（3）下采样。卷积网络的结构如图5.17所示。在卷积层，各个单元被组织在一系列平面中，每个平面被称为一个特征地图（feature map）。一个特征地图中的每个单元只从图像的一个小的子区域接收输入，且一个特征地图中的所有单元被限制为共享相同的权值。例如，一个特征地图可能由100个单元组成，这些单元被放在了 10×10 的网格中，每个单元从图像的一个 5×5 的像素块接收输入。于是，整个特征地图就有25个可调节的参数，加上一个可调节的偏置参数。来自一个像素块的输入值被权值和偏置进行线性组合，线性组合的结果通过公式（5.1）给出的S形非线性函数进行变换。如果我们把每个单元想象成特征检测器，那么特征地图中的所有单元都检测了输入图像中的相同的模式，但是位置不同。由于权值共享，这些单元的激活的计算等价于使用一个由权向量组成和“核”对图像像素的灰度值进行卷积。如果输入图像发生平移，那么特征地图的激活也会发生等量的平移，否则就不发生改变。这提供了神经网络输出对于输入图像的平移和变形的（近似）不变性的基础。由于我们通常需要检测多个特征来构造一个有效的模型，因此通常在卷积层会有多个特征地图，每个都有自己的权值和偏置参数。

卷积单元的输出构成了网络的下采样层的输入。对于卷积层的每个特征地图，有一个下采样层的单元组成的平面，并且下采样层的每个单元从对应的卷积层的特征地图中的一个小的接收

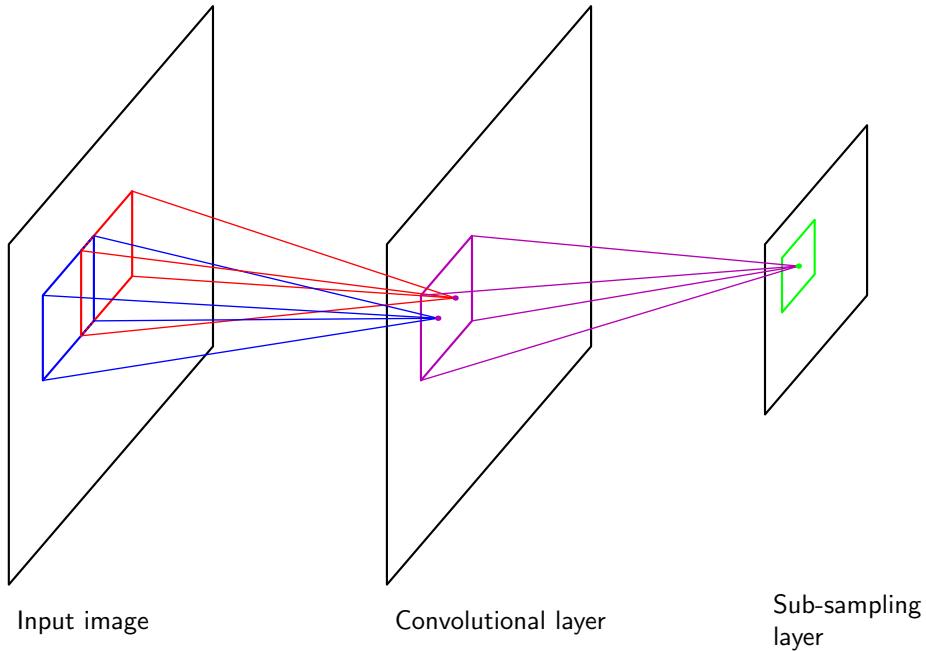


图 5.17: 卷积神经网络的一个例子，给出了一层卷积单元层跟着一个下采样单元层。可能连续使用这种层对。

场接收输入。这些单元完成了下采样。例如，每个下采样单元可能从对应的特征地图中的一个 2×2 单元的区域中接收输入，然后计算这些输入的平均值，乘以一个可调节的权值和可调节的偏置参数，然后使用S形非线性激活函数进行变换。选择的接收场是连续的、非重叠的，从而下采样层的行数和列数都是卷积层的一半。使用这种方式，下采样层的单元的响应对于对应的输入空间区域中的图片的微小平移相对不敏感。

在实际构造中，可能有若干对卷积层和下采样层。在每个阶段，与前一层相比，都会有一个更高层次的关于输入变换的不变性。在一个给定的卷积层中，对于每个由前一个下采样层的单元构成的平面，可能存在若干个特征地图，从而空间分辨率的逐层减小就可以通过增加特征的数量进行补偿。网络的最后一层通常是完全连接的，是一个完全可调节的层。在多分类问题中，输出层使用的是softmax非线性函数。

整个网络可以使用误差函数最小化的方法计算。误差函数梯度的计算可以使用反向传播算法。这需要对通常的反向传播算法进行微小的修改，确保共享权值的限制能够满足。由于使用局部接收场，网络中权值的数量要小于完全连接的网络的权值数量。此外，由于权值的本质数量的限制，需要从训练数据中学习到的独立参数的数量仍然相当小。

5.5.7 软权值共享

降低具有大量权值参数的网络复杂度的一种方法是将权值分组，然后令分组内的权值相等。这是图5.5.6中讨论的权值共享的方法，这种方法将网络对于图像的平移不变性整合到网络的构建过程中。然而，它只适用于限制的形式可以事先确定的问题中。这里，我们考虑软权值共享 (soft weight sharing) (Nowlan and Hinton, 1992)。这种方法中，权值相等的硬限制被替换为一种形式的正则化，其中权值的分组倾向于取近似的值。此外，权值的分组、每组权值的均值，以及分组内的取值范围全都作为学习过程的一部分被确定。

回忆一下，公式 (5.112) 给出的简单的权值衰减正则化项可以被看成权值上的高斯分布的负对数。我们可以将权值分为若干组，而不是将所有权值分为一个组。分组的方法是使用高斯混合概率分布。混合分布中，每个高斯分量的均值、方差，以及混合系数，都会作为可调节的参数在学习过程中被确定。于是，我们有下面形式的概率密度

$$p(\mathbf{w}) = \prod_i p(w_i) \quad (5.136)$$

其中

$$p(w_i) = \sum_{j=1}^M \pi_j \mathcal{N}(w_i | \mu_j, \sigma_j^2) \quad (5.137)$$

π_j 为混合系数。取负对数，即可得到正则化函数，形式为

$$\Omega(\mathbf{w}) = -\sum_i \ln \left(\sum_{j=1}^M \pi_j \mathcal{N}(w_i | \mu_j, \sigma_j^2) \right) \quad (5.138)$$

从而，总的误差函数为

$$\tilde{E}(\mathbf{w}) = E(\mathbf{w}) + \lambda \Omega(\mathbf{w}) \quad (5.139)$$

其中， λ 是正则化系数。这个误差函数同时关于权值 w_i 和混合模型参数 $\{\pi_j, \mu_j, \sigma_j\}$ 进行最小化。如果权值是常数，那么混合模型的参数可以由第9章讨论的EM算法确定。然而，权值分布本身在学习过程中是不断变化的，因此为了避免数值的不稳定性，我们同时关于权值和混合模型参数进行最优化。可以使用标准的最优化算法（例如共轭梯度法或拟牛顿法）来完成这件事。

为了最小化总的误差函数，能够计算出它关于各个可调节参数的导数是很有必要的。为了完成这一点，比较方便的做法是把 $\{\pi_j\}$ 当成先验概率，然后引入对应的后验概率。根据公式(2.192)，后验概率由贝叶斯定理给出，形式为

$$\gamma_j(w) = \frac{\pi_j \mathcal{N}(w | \mu_j, \sigma_j^2)}{\sum_k \pi_k \mathcal{N}(w | \mu_k, \sigma_k^2)} \quad (5.140)$$

这样，总的误差函数关于权值的导数为

$$\frac{\partial \tilde{E}}{\partial w_i} = \frac{\partial E}{\partial w_i} + \lambda \sum_j \gamma_j(w_i) \frac{(w_i - \mu_j)}{\sigma_j^2} \quad (5.141)$$

于是，正则化项的效果是把每个权值拉向第 j 个高斯分布的中心，拉力正比于对于给定权值的高斯分布的后验概率。这恰好就是我们要寻找的效果。

误差函数关于高斯分布的中心的导数也很容易计算，结果为

$$\frac{\partial \tilde{E}}{\partial \mu_j} = \lambda \sum_i \gamma_j(w_i) \frac{(\mu_j - w_i)}{\sigma_j^2} \quad (5.142)$$

它具有简单的直观含义，因为它把 μ_j 拉向了权值的平均值，拉力为第 j 个高斯分量产生的权值参数的后验概率。类似地，关于方差的导数为

$$\frac{\partial \tilde{E}}{\partial \sigma_j} = \lambda \sum_i \gamma_j(w_i) \left(\frac{1}{\sigma_j} - \frac{(w_i - \mu_j)^2}{\sigma_j^3} \right) \quad (5.143)$$

它将 σ_j 拉向权值在对应的中心 μ_j 附近的偏差的平方的加权平均，加权平均的权系数与之前一样，等于由第 j 个高斯分量产生的权值参数的后验概率。注意，在实际执行过程中，我们会引入一个新的变量 ξ_j ，它由下式定义。

$$\sigma_j^2 = \exp(\xi_j) \quad (5.144)$$

并且，最小化的过程是关于 ξ_j 进行的。这确保了参数 σ_j 是正数。此外，它还能够倾向于避免找到病态解，即一个或者多个 σ_j 趋于零，对应于一个高斯分量退化为一个权参数的值。9.2.1节会在高斯混合模型的问题中详细讨论这样的解。

对于关于混合系数 π_j 的导数，我们需要考虑下面的限制条件

$$\sum_j \pi_j = 1, \quad 0 \leq \pi_i \leq 1 \quad (5.145)$$



图 5.18: 左图给展示了一个具有两个连接的机械臂，其中，末端的笛卡尔坐标 (x_1, x_2) 由两个连接角 θ_1 和 θ_2 以及机械臂的（固定）长度 L_1 和 L_2 唯一确定。这被称为机械臂的正向运动学（forward kinematics）。在实际应用中，我们必须寻找给出所需的末端位置的连接角，如右图所示。这个逆向运动学（inverse kinematics）有两个对应的解，即“肘部向上”和“肘部向下”。

这个限制的产生，是因为我们把 π_j 看成了先验概率。可以这样做：将混合系数通过一组辅助变量 $\{\eta_j\}$ 用softmax函数表示，即

$$\pi_j = \frac{\exp(\eta_j)}{\sum_{k=1}^M \exp(\eta_k)} \quad (5.146)$$

这样，正则化的误差函数关于 $\{\eta_j\}$ 的形式为

$$\frac{\partial \tilde{E}}{\partial \eta_j} = \sum_i \{\pi_j - \gamma_j(w_i)\} \quad (5.147)$$

我们看到， π_j 被拉向第 j 个高斯分量的平均后验概率。

5.6 混合密度网络

有监督学习的目标是对条件概率分布 $p(t | x)$ 建模。对于许多简单的回归问题来说，这个分布都被选为高斯分布。然而，在实际的机器学习问题中，经常会遇到与高斯分布差别相当大的概率分布。例如，在逆问题（inverse problem）中，概率分布可以是多峰的，这种情况下，高斯分布的假设就会产生相当差的预测结果。

作为逆问题的一个简单的例子，考虑机械臂的运动学问题，如图 5.18 所示。正向问题（forward problem）是在给定连接角的情况下求解机械臂末端的位置，这个问题有唯一解。然而，在实际应用中，我们想把机械臂末端移动到一个具体的位置。为了完成移动，我们必须设定合适的连接角。于是，我们需要求解逆问题，它有两个解，如图 5.18 所示。

正向问题通常对应于物理系统的因果关系，通常有唯一解。例如，人体的某个具体的症状是由于特定的疾病造成的。然而在模式识别中，我们通常不得不求解逆问题，例如在给定症状的情况下，推断疾病的种类。如果正向问题涉及到多对一的映射，那么逆问题就会有多个解。例如，多种不同的疾病可能会导致相同的症状。

在机械臂的例子中，运动由几何方程定义，多峰的性质是显然的。然而，在许多机器学习问题中，尤其是涉及到高维空间的问题中，多峰性质的存在并不显然。但是，为了教学的目的，我们会考虑一个相当简单的问题，这个问题中我们可以很容易地看出多峰性质。这个问题的数据的生成方式为：对服从区间 $(0, 1)$ 的均匀分布的变量 x 进行取样，得到一组值 $\{x_n\}$ ，对应的目标值 t_n 通过下面的方式得到：计算函数 $x_n + 0.3 \sin(2\pi x_n)$ ，然后添加一个服从 $(-0.1, 0.1)$ 上的均匀分布的噪声。这样，逆问题就可以这样得到：使用相同的数据点，但是交换 x 和 t 的角色。图 5.19 给出了正向问题和逆问题的数据集，以及一个两层神经网络给出的结果。这个两层的神经网络有 6 个隐含单元，一个线性输出单元，误差函数为平方和误差函数。在高斯分布的假设下，最小平方方法对应于最大似然方法。我们看到，对于不服从高斯分布的逆问题，这种解法产生的模型非常差。

于是，我们寻找一个对条件概率密度建模的一般的框架。可以这样做：为 $p(t | x)$ 使用一个混合模型，模型的混合系数和每个分量的概率分布都是输入向量 x 的一个比较灵活的函数，这就构成了混合密度网络（mixture density network）。对于任意给定的 x 值，混合模型提供了一个

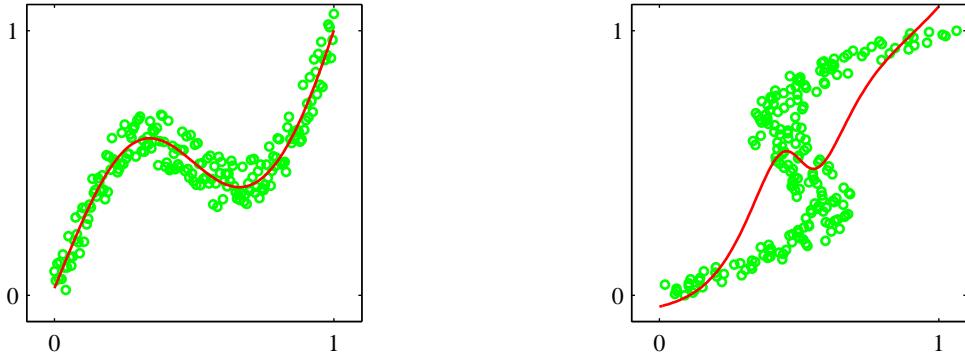


图 5.19: 左图是一个简单的“正向问题”的数据集，其中红色曲线给出了通过最小化平方和误差函数调节一个两层神经网络的结果。对应的逆问题，如右图所示，通过交换 x 和 t 的顺序的方式得到。这里，通过最小化平方和误差函数的方式训练的神经网络给出了对数据的非常差的拟合，因为数据集是多峰的。

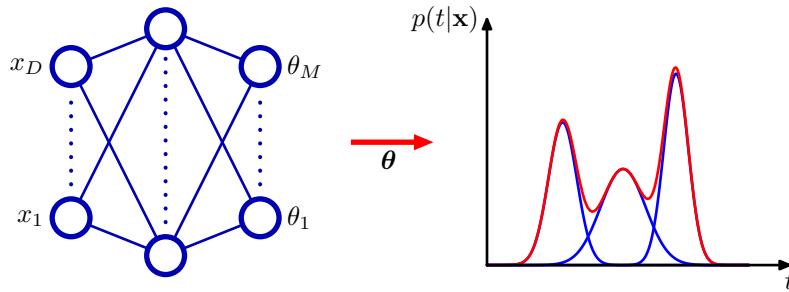


图 5.20: 混合密度网络（mixture density network）可以表示一般的条件概率密度 $p(t | x)$ ，方法为：考虑 t 的一个参数化的混合模型，它的参数由以 x 为输入的神经网络的输出确定。

通用的形式，用来对任意条件概率密度函数 $p(t | x)$ 进行建模。假设我们考虑一个足够灵活的网络，那么我们就有了一个近似任意条件概率分布的框架。

这里，我们显式地令模型的分量为高斯分布，即

$$p(t | x) = \sum_{k=1}^K \pi_k(x) \mathcal{N}(t | \mu_k(x), \sigma_k^2(x) I) \quad (5.148)$$

这是异方差模型（heteroscedastic model）的一个例子，因为数据中的噪声方差是输入向量 x 的一个函数。我们也可以使用高斯分布以外的其他分布，例如，如果目标变量是二值的而不是连续的，我们就可以使用伯努利分布。我们已经把情况具体到了各向同性的协方差的情形，虽然可以通过使用Cholesky分解（Williams, 1996）表示协方差的方式，将混合密度网络扩展到可以处理更一般的协方差的情形。即使每个分量的方差是各向同性的，但是我们仍然不能假设条件概率分布 $p(t | x)$ 能够关于 t 的分量进行分解（这与标准的平方和回归模型不同），这是由于概率分布是一个混合分布。

我们现在为混合模型取各种不同的参数，这些参数包括混合系数 $\pi_k(x)$ 、均值 $\mu_k(x)$ 以及方差 $\sigma_k^2(x)$ ，这些参数控制了以 x 作为输入的神经网络的输出。这个混合密度网络的结构如图5.20所示。混合密度网络与14.5.3节讨论的混合专家的关系十分紧密。主要的区别是，混合密度网络使用相同的函数来预测所有分量概率分布的参数以及混合参数，因此非线性隐含单元被依赖于输入的函数所共享。

图5.20所示的神经网络可以是一个两层的网络，网络具有S形（双曲正切）隐含单元。如果混合模型（5.148）中有 K 个分量，且 t 有 L 个分量，那么网络就会有 K 个输出单元激活（记作 a_k^π ）确定混合系数 $\pi_k(x)$ ，有 K 个输出（记作 a_k^σ ）确定核宽度 $\sigma_k(x)$ ，有 $K \times L$ 个输出（记作 a_{kj}^μ ）确定核中心 $\mu_k(x)$ 的分量 $\mu_{kj}(x)$ 。网络输出的总数为 $(L + 2)K$ ，这与通常的网络的 L 个输出不同。通常的网络只是简单地预测目标变量的条件均值。

混合系数必须满足下面的限制。

$$\sum_{k=1}^K \pi_k(\mathbf{x}) = 1, \quad 0 \leq \pi_k(\mathbf{x}) \leq 1 \quad (5.149)$$

可以通过使用一组softmax输出来实现。

$$\pi_k(\mathbf{x}) = \frac{\exp(a_k^\pi)}{\sum_{l=1}^K \exp(a_l^\pi)} \quad (5.150)$$

类似地，方差必须满足 $\sigma_k^2(\mathbf{x}) \geq 0$ ，因此可以使用对应的网络激活的指数形式表示，即

$$\sigma_k(\mathbf{x}) = \exp(a_k^\sigma) \quad (5.151)$$

最后，由于均值 $\mu_k(\mathbf{x})$ 有实数分量，因此它们可以直接用网络的输出激活表示

$$\mu_{kj}(\mathbf{x}) = a_{kj}^\mu \quad (5.152)$$

混合密度网络的可调节参数由权向量 \mathbf{w} 和偏置组成。这些参数可以通过最大似然法确定，或者等价地，使用最小化误差函数（负对数似然函数）的方法确定。对于独立的数据，误差函数的形式为

$$E(\mathbf{w}) = -\sum_{n=1}^N \ln \left\{ \sum_{k=1}^K \pi_k(\mathbf{x}_n, \mathbf{w}) \mathcal{N}(\mathbf{t}_n | \boldsymbol{\mu}_k(\mathbf{x}_n, \mathbf{w}), \sigma_k^2(\mathbf{x}_n, \mathbf{w}) \mathbf{I}) \right\} \quad (5.153)$$

其中，我们显式地写出了误差函数对于 \mathbf{w} 的依赖。

为了最小化误差函数，我们需要计算误差函数 $E(\mathbf{w})$ 关于 \mathbf{w} 的分量的导数。如果我们得到了误差函数关于输出单元激活的导数的表达式，那么我们就可以通过标准的反向传播方法来计算误差函数关于 \mathbf{w} 的分量的导数。误差函数关于输出单元激活的导数代表了每个模式和每个输出单元的误差信号 σ ，并且可以反向传播到隐含单元，从而误差函数的导数可以按照通常的方式进行计算。由于误差函数 (5.153) 由一组项的求和式构成，每一项都对应一个训练数据点，因此我们可以考虑对于特定的模式 n 的导数，然后通过求和的方式找到 E 的导数。

由于我们处理的是混合概率分布，因此比较方便的做法是把混合系数 $\pi_k(\mathbf{x})$ 看成与 \mathbf{x} 相关的先验概率分布，从而就引入了对应的后验概率，形式为

$$\gamma_{nk} = \gamma_k(\mathbf{t}_n | \mathbf{x}_n) = \frac{\pi_k \mathcal{N}_{nk}}{\sum_{l=1}^K \pi_l \mathcal{N}_{nl}} \quad (5.154)$$

其中 \mathcal{N}_{nk} 表示 $\mathcal{N}(\mathbf{t}_n | \boldsymbol{\mu}_k(\mathbf{x}_n), \sigma_k^2(\mathbf{x}_n))$ 。

关于控制混合系数的网络输出激活的导数为

$$\frac{\partial E_n}{\partial a_k^\pi} = \pi_k - \gamma_{nk} \quad (5.155)$$

类似地，关于控制分量均值的网络输出激活的导数为

$$\frac{\partial E_n}{\partial a_{kl}^\mu} = \gamma_k \left\{ \frac{\mu_{kl} - t_{nl}}{\sigma_k^2} \right\} \quad (5.156)$$

最后，关于控制分量方差的网络激活函数为

$$\frac{\partial E_n}{\partial a_k^\sigma} = \gamma_{nk} \left\{ L - \frac{\|\mathbf{t}_n - \boldsymbol{\mu}_k\|^2}{\sigma_k^2} \right\} \quad (5.157)$$

我们回到图5.19所示的逆问题的简单例子，来说明混合密度网络的应用。图5.21给出了混合系数 $\pi_k(x)$ 、均值 $\mu_k(x)$ 和对应于 $p(t | x)$ 的条件概率轮廓线。神经网络的输出，即混合模型

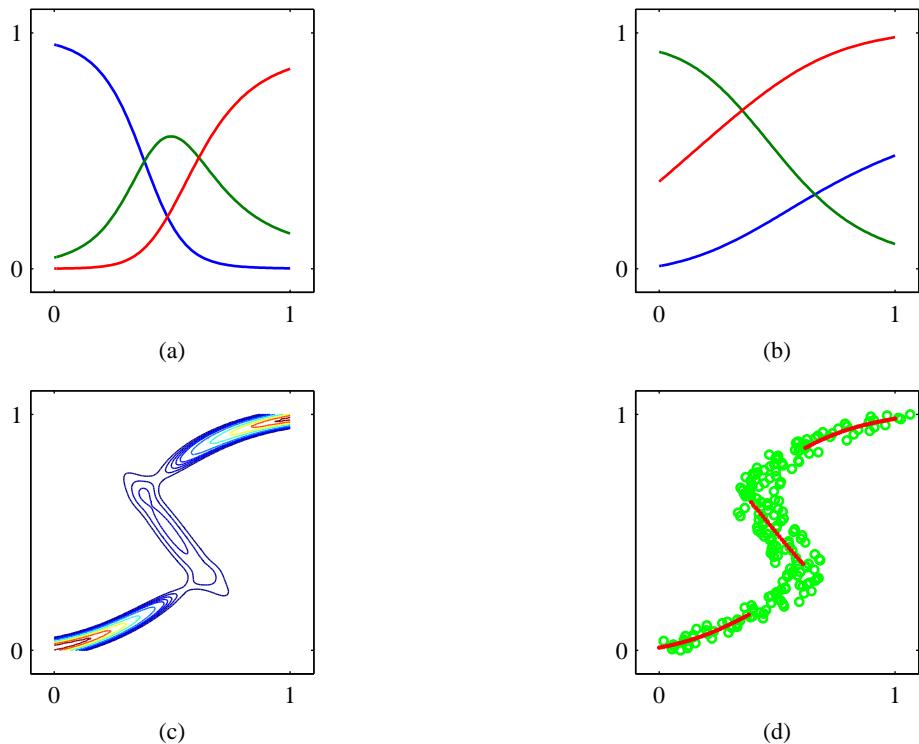


图 5.21: (a)对于使用图5.19给出的数据训练的混合密度网络的三个核函数，混合系数 $\pi_k(x)$ 与 x 的函数关系图像。模型有三个高斯分量，使用了一个多层感知器，在隐含层有五个“tanh”单元，同时有9个输出单元（对于高斯分量的3个均值、3个方差以及3个混合系数）。在较小的 x 值和较大的 x 值处，目标数据的条件概率密度是单峰的，对于它的先验概率分布，只有一个核具有最大的值。而在中间的 x 值处，条件概率分布具有3个峰，3个混合系数具有可比的值。(b)使用与混合系数相同的方法来表示均值 $\mu_k(x)$ 。(c)对于同样的混合密度网络，目标数据的条件概率密度的图像。(d)条件概率密度的近似条件峰值的图像，用红色点表示。

的参数，是输入变量的连续单值函数。然而，从图5.21(c)中我们可以看到，通过调整混合分量 $\pi_k(\mathbf{x})$ 的大小，模型能够产生一个对于某些 x 是单峰的，对于其他 x 值是多峰的概率分布。

一旦混合密度网络训练结束，他就可以预测对于任意给定的输入向量的目标数据的条件密度函数。只要我们关注的是预测输出向量的值的问题，那么这个条件概率密度就能完整地描述用于生成数据的概率分布。根据这个概率密度函数，我们可以计算不同应用中我们感兴趣的更加具体的量。一个最简单的量就是目标数据的条件均值，即

$$\mathbb{E}[\mathbf{t} | \mathbf{x}] = \int \mathbf{t} p(\mathbf{t} | \mathbf{x}) d\mathbf{t} = \sum_{k=1}^K \pi_k(\mathbf{x}) \boldsymbol{\mu}_k(\mathbf{x}) \quad (5.158)$$

其中我们使用了公式(5.148)。由于使用最小平方方法训练的标准的神经网络近似了条件均值，因此我们看到一个混合密度网络可以复制传统的最小平方的结果，作为一个特例。当然，正如我们已经注意到的那样，对于一个多峰分布，条件均值是一个受限的值。

类似地，我们可以利用条件均值的结果，计算密度函数的方差，结果为

$$s^2(\mathbf{x}) = \mathbb{E}[\|\mathbf{t} - \mathbb{E}[\mathbf{t} | \mathbf{x}]\|^2 | \mathbf{x}] \quad (5.159)$$

$$= \sum_{k=1}^K \pi_k(\mathbf{x}) \left\{ \sigma_k^2(\mathbf{x}) + \left\| \boldsymbol{\mu}_k(\mathbf{x}) - \sum_{l=1}^K \pi_l(\mathbf{x}) \boldsymbol{\mu}_l(\mathbf{x}) \right\|^2 \right\} \quad (5.160)$$

其中我们使用了公式(5.148)和(5.158)。这对应的最小平方结果相比，这个结果更一般，因为方差是 \mathbf{x} 的一个函数。

我们已经看到，对于多峰分布，用条件均值描述数据的效果很差。例如，在图5.18给出的控制机械臂的例子中，我们需要从两个可能的连接角中选出一个，来得到所需的末端位置。在这种情况下，条件众数可能更有价值。由于混合密度网络的条件众数没有一个简单的解析解，因此需要数值迭代。一个简单的替代方法是取每个 \mathbf{x} 对应的最可能分量（即具有最大混合系数的分量）的均值。对于之前的那个简单的例子，图5.21(d)给出了这个结果。

5.7 贝叶斯神经网络

目前为止，我们对于神经网络的讨论集中于使用最大似然方法来确定网络的参数（权值和偏置）。正则化的最大似然方法可以看成MAP（maximum posterior）方法，其中正则化项可以被看成先验参数分布的对数。然而，在贝叶斯方法中，为了进行预测，我们需要对参数的概率分布进行积分或求和。

在3.3节，我们研究了在高斯噪声假设下的简单线性回归模型的贝叶斯解。我们看到，后验概率分布（是一个高斯分布）可以精确计算，并且预测分布也具有解析解。在多层次神经网络的情况下，网络函数对于参数值的高度非线性的性质意味着精确的贝叶斯方法不再可行。事实上，后验概率分布的对数是非凸的，对应于误差函数中的多个局部极小值。

第10章将要讨论的变分推断方法已经被用在了贝叶斯神经网络中。这种方法使用了对后验概率的分解的高斯近似（Hinton and van Camp, 1993），也使用了一个具有完成协方差矩阵的高斯分布（Barber and Bishop, 1998a; Barber and Bishop, 1998b）。但是，最完整的贝叶斯方法是基于拉普拉斯近似的方法（MacKay, 1992c; MacKay, 1992b），这种方法构成了本节讨论的基础。我们会使用一个以真实后验概率的众数为中心的高斯分布来近似后验概率分布。此外，我们会假设这个高斯分布的协方差很小，从而网络函数关于参数空间的区域中的参数近似是线性关系。在参数空间中，后验概率距离概率为零的状态相当远。使用这两个近似，我们会得到与之前讨论的线性回归和线性分类的模型相类似的模型，从而我们就可以利用之前得到了结果了。这样，我们可以使用模型证据的框架来对参数进行点估计，并且比较不同的模型（例如，有着不同的隐含单元数量的网络）。首先，我们讨论回归问题的情形，然后，我们考虑进行必要的修改，用来解决分类问题。

5.7.1 后验参数分布

考虑从输入向量 \mathbf{x} 预测单一连续目标变量 t 的问题（扩展到多个目标变量的情形很容易）。我们假设条件概率分布 $p(t | \mathbf{x})$ 是一个高斯分布，均值与 \mathbf{x} 有关，由神经网络模型的输出 $y(\mathbf{x}, \mathbf{w})$ 确定，精度（方差的倒数） β 为

$$p(t | \mathbf{x}, \mathbf{w}, \beta) = \mathcal{N}(t | y(\mathbf{x}, \mathbf{w}), \beta^{-1}) \quad (5.161)$$

类似地，我们将权值 \mathbf{w} 的先验概率分布选为高斯分布，形式为

$$p(\mathbf{w} | \alpha) = \mathcal{N}(\mathbf{w} | \mathbf{0}, \alpha^{-1} \mathbf{I}) \quad (5.162)$$

对于 N 次独立同分布的观测 $\mathbf{x}_1, \dots, \mathbf{x}_N$ ，对应的目标值集合 $\mathcal{D} = \{t_1, \dots, t_N\}$ ，似然函数为

$$p(\mathcal{D} | \mathbf{w}, \beta) = \prod_{n=1}^N \mathcal{N}(t_n | y(\mathbf{x}_n, \mathbf{w}), \beta^{-1}) \quad (5.163)$$

因此最终的后验概率为

$$p(\mathbf{w} | \mathcal{D}, \alpha, \beta) \propto p(\mathbf{w} | \alpha) p(\mathcal{D} | \mathbf{w}, \beta) \quad (5.164)$$

由于 $y(\mathbf{x}, \mathbf{w})$ 与 \mathbf{w} 的关系是非线性的，因此后验概率不是高斯分布。

使用拉普拉斯近似，我们可以找到对于后验概率分布的一个高斯近似。为了完成这一点，我们必须首先找到后验概率分布的一个（局部）最大值，这必须使用迭代的数值最优化算法才能找到。与之前一样，比较方便的做法是最大化后验概率分布的对数，它可以写成下面的形式

$$\ln p(\mathbf{w} | \mathcal{D}) = -\frac{\alpha}{2} \mathbf{w}^T \mathbf{w} - \frac{\beta}{2} \sum_{n=1}^N \{y(\mathbf{x}_n, \mathbf{w}) - t_n\}^2 + \text{常数} \quad (5.165)$$

这对应于一个正则化的平方和误差函数。假设 α 和 β 都是定值，那么我们可以通过标准的非线性最优化算法（例如共轭梯度法），使用误差反向传播计算所需的导数，找到后验概率的最大值。我们将最大值的位置记作 \mathbf{w}_{MAP} 。

找到了 \mathbf{w}_{MAP} 的众数，我们就可以通过计算后验概率分布的负对数的二阶导数，建立一个局部的高斯近似。根据公式 (5.165)，负对数后验概率的二阶导数为

$$\mathbf{A} = -\nabla \nabla \ln p(\mathbf{w} | \mathcal{D}, \alpha, \beta) = \alpha \mathbf{I} + \beta \mathbf{H} \quad (5.166)$$

这里， \mathbf{H} 是一个Hessian矩阵，由平方和误差函数关于 \mathbf{w} 的分量组成。计算和近似Hessian矩阵的方法已经在5.4节讨论过。这样，后验概率对应的高斯近似由公式 (4.134) 给出，形式为

$$q(\mathbf{w} | \mathcal{D}) = \mathcal{N}(\mathbf{w} | \mathbf{w}_{MAP}, \mathbf{A}^{-1}) \quad (5.167)$$

类似地，预测分布可以通过将后验概率分布求积分的方式获得。

$$p(t | \mathbf{x}, \mathcal{D}) = \int p(t | \mathbf{x}, \mathbf{w}) q(\mathbf{w} | \mathcal{D}) d\mathbf{w} \quad (5.168)$$

然而，即使对于后验分布的高斯近似，这个积分仍然无法得到解析解，因为网络函数 $y(\mathbf{x}, \mathbf{w})$ 与 \mathbf{w} 的关系是非线性的。为了将计算过程进行下去，我们现在假设，与 $y(\mathbf{x}, \mathbf{w})$ 发生变化造成的变化幅度相比，后验概率分布的方差较小。这使得我们可以在 \mathbf{w}_{MAP} 附近对网络函数进行泰勒展开。只保留展开式的现行项，可得

$$y(\mathbf{x}, \mathbf{w}) \simeq y(\mathbf{x}, \mathbf{w}_{MAP}) + \mathbf{g}^T (\mathbf{w} - \mathbf{w}_{MAP}) \quad (5.169)$$

其中我们定义了

$$\mathbf{g} = \nabla_{\mathbf{w}} y(\mathbf{x}, \mathbf{w})|_{\mathbf{w}=\mathbf{w}_{MAP}} \quad (5.170)$$

使用这个近似，我们现在得到了一个线性高斯模型， $p(\mathbf{w})$ 为高斯分布。并且， $p(t | \mathbf{w})$ 也是高斯分布，它的均值是 \mathbf{w} 的线性函数，分布的形式为

$$p(t | \mathbf{x}, \mathbf{w}, \beta) \simeq \mathcal{N}(t | y(\mathbf{x}, \mathbf{w}_{MAP}) + \mathbf{g}^T(\mathbf{w} - \mathbf{w}_{MAP}), \beta^{-1}) \quad (5.171)$$

于是我们可以使用公式 (2.115) 给出的边缘分布 $p(t)$ 的一般结果，得到

$$p(t | \mathbf{x}, \mathcal{D}, \alpha, \beta) = \mathcal{N}(t | y(\mathbf{x}, \mathbf{w}_{MAP}), \sigma^2(\mathbf{x})) \quad (5.172)$$

其中，与输入相关的方差为

$$\sigma^2(\mathbf{x}) = \beta^{-1} + \mathbf{g}^T \mathbf{A}^{-1} \mathbf{g} \quad (5.173)$$

我们看到预测分布 $p(t | \mathbf{x}, \mathcal{D})$ 是一个高斯分布，它的均值由网络函数 $y(\mathbf{x}, \mathbf{w}_{MAP})$ 给出，参数设置为了MAP值。方差由两项组成。第一项来自目标变量的固有噪声，第二项是一个与 \mathbf{x} 相关的项，表示由于模型参数 \mathbf{w} 的不确定性造成的内插的不确定性。可以将这个结果与公式 (3.58) 和公式 (3.59) 给出的线性回归模型的对应的预测分布进行对比。

5.7.2 超参数最优化

目前为止，我们已经假定了超参数 α 和 β 是固定的、已知的。我们可以使用3.5节讨论的模型证据框架，结合使用拉普拉斯近似得到的后验概率的高斯近似，得到确定这些超参数的值的步骤。

超参数的边缘似然函数，或者模型证据，可以通过对网络权值进行积分的方法得到，即

$$p(\mathcal{D} | \alpha, \beta) = \int p(\mathcal{D} | \mathbf{w}, \beta) p(\mathbf{w} | \alpha) d\mathbf{w} \quad (5.174)$$

通过使用拉普拉斯近似的结果 (4.135)，这个积分很容易计算。取对数，可得

$$\ln p(\mathcal{D} | \alpha, \beta) \simeq -E(\mathbf{w}_{MAP}) - \frac{1}{2} \ln |\mathbf{A}| + \frac{W}{2} \ln \alpha + \frac{N}{2} \ln \beta - \frac{N}{2} \ln(2\pi) \quad (5.175)$$

其中 W 是 \mathbf{w} 中参数的总数。正则化误差函数的定义为

$$E(\mathbf{w}_{MAP}) = \frac{\beta}{2} \sum_{n=1}^N \{y(\mathbf{x}_n, \mathbf{w}_{MAP}) - t_n\}^2 + \frac{\alpha}{2} \mathbf{w}_{MAP}^T \mathbf{w}_{MAP} \quad (5.176)$$

我们看到这与线性回归模型的对应的结果 (3.86) 的函数形式相同。

在模型证据框架中，我们通过最大化 $\ln p(\mathcal{D} | \alpha, \beta)$ 对 α 和 β 进行点估计。首先考虑关于 α 进行最大化，这可以通过与3.5.2节讨论的线性回归的情形相类似的方法计算。首先，我们定义特征值方程

$$\beta \mathbf{H} \mathbf{u}_i = \lambda_i \mathbf{u}_i \quad (5.177)$$

其中 \mathbf{H} 是在 $\mathbf{w} = \mathbf{w}_{MAP}$ 处计算的Hessian矩阵，由平方和误差函数的二阶导数组成。通过类比公式 (3.92)，我们有

$$\alpha = \frac{\gamma}{\mathbf{w}_{MAP}^T \mathbf{w}_{MAP}} \quad (5.178)$$

其中 γ 表示参数的有效数量，定义为

$$\gamma = \sum_{i=1}^W \frac{\lambda_i}{\alpha + \lambda_i} \quad (5.179)$$

注意，这个结果与线性回归的情形完全相同。然而，对于非线性神经网络，它忽略了下面的事实： α 的改变会引起Hessian矩阵 \mathbf{H} 的改变，进而改变特征值。于是，我们隐式地忽略了涉及到 λ_i 关于 α 的导数的项。

类似地，根据公式 (3.95)，我们看到，关于 β 最大化模型证据，可以得到下面的重估计公式

$$\frac{1}{\beta} = \frac{1}{N - \gamma} \sum_{n=1}^N \{y(\mathbf{x}_n, \mathbf{w}_{MAP}) - t_n\}^2 \quad (5.180)$$

与线性模型一样，我们需要交替地进行超参数 α 和 β 的重新估计以及后验概率分布的更新。然而，对于神经网络来说，由于后验概率分布的多峰性质，情况更复杂。结果，使用最大化对数后验概率的方法找到的解 \mathbf{w}_{MAP} 将依赖于 \mathbf{w} 的初始化。只要我们考虑的是预测问题，那么仅仅由于隐含层的结点交换和符号改变所造成的不同结果将给出相同的预测，并且预测的结果与等价解中的哪一个解被找到没有关系。然而，也可能存在不等价的解，这些通常会产生不同的最优超参数。

为了比较不同的模型，例如具有不同隐含单元数量的神经网络，我们需要计算模型证据 $p(\mathcal{D})$ 。将使用迭代最优化过程得到的超参数值 α 和 β 代入公式 (5.175)，我们可以得到模型证据的近似。一个更加仔细的计算方法是关于 α 和 β 求积分，同时使用一个高斯近似 (MacKay, 1992; Bishop, 1995a)。在这两种方法中，都需要计算Hessian矩阵的行列式 $|\mathbf{A}|$ 。这在实际应用中会有很大的问题，因为与矩阵的迹不同，行列式对于小的特征值比较敏感，而这些特征值通常很难精确计算。

拉普拉斯近似基于的是权值的后验概率分布的众数附近的局部二次展开。在5.1.1节，我们已经看到，在两层神经网络中，任意给定的众数都是 $M!2^M$ 个等价的众数中的一个，这些等价的众数由网络的互换对称性和符号对称性造成，其中 M 是隐含结点的数量。当比较具有不同隐含结点数量的网络时，通过将模型证据乘以因子 $M!2^M$ ，就可以考虑到这一点。

5.7.3 用于分类的贝叶斯神经网络

目前，我们已经使用了拉普拉斯近似，推导出了神经网络回归模型的贝叶斯方法。我们现在要讨论的是，当应用于分类问题时，这个框架应该如何修改。这里，我们要考虑的网络有一个logistic sigmoid输出，对应于一个二分类问题。将网络扩展到多类softmax输出是很直接的。我们构建神经网络的过程与4.5节讨论线性分类模型的结果十分类似，因此我们建议读者在学习本节之前，应该对那一节的内容比较熟悉。

模型的对数似然函数为

$$\ln p(\mathcal{D} | \mathbf{w}) = \sum_{n=1}^N \{t_n \ln y_n + (1 - t_n) \ln(1 - y_n)\} \quad (5.181)$$

其中 $t_n \in \{0, 1\}$ 是目标值，且 $y_n \equiv y(\mathbf{x}_n, \mathbf{w})$ 。注意，这里没有超参数 β ，因为我们假定数据点被正确标记了。与之前一样，先验概率分布是公式 (5.162) 给出的各向同性高斯分布。

将拉普拉斯框架用在这个模型中的第一个阶段是初始化超参数 α ，然后通过最大化对数后验概率分布的方法确定参数向量 \mathbf{w} 。这等价于最小化正则化误差函数

$$E(\mathbf{w}) = -\ln p(\mathcal{D} | \mathbf{w}) + \frac{\alpha}{2} \mathbf{w}^T \mathbf{w} \quad (5.182)$$

最小化的过程可以通过使用误差反向传播方法结合标准的最优化算法得到，正如5.3节所说的那样。

找到权向量的解 \mathbf{w}_{MAP} 之后，下一步是计算由负对数似然函数的二阶导数组成的Hessian矩阵 \mathbf{H} 。这可以通过使用5.4.5节介绍的精确方法，或者使用公式 (5.85) 给出的外积近似方法求得。负对数后验概率的二阶导数可以写成公式 (5.166) 的形式，这样，后验概率的高斯近似就由公式 (5.167) 给出。

为了最优化超参数 α ，我们再次最大化边缘似然函数。很容易证明，边缘似然函数的形式为

$$\ln p(\mathcal{D} | \alpha) \simeq -E(\mathbf{w}_{MAP}) - \frac{1}{2} \ln |\mathbf{A}| + \frac{W}{2} \ln \alpha \quad (5.183)$$

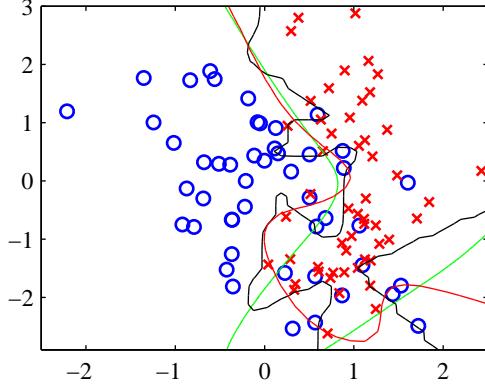


图 5.22: 模型证据框架应用于人工生成的二分类数据集的说明。绿色曲线表示最优的决策边界，黑色曲线表示通过最大化似然函数调节一个具有8个隐含结点的两层神经网络的结果，红色曲线表示包含一个正则化项的结果，其中 α 使用模型证据的步骤进行了最优化，初始值为 $\alpha = 0$ 。注意，模型证据步骤极大地缓解了模型的过拟合现象。

其中，正则化的误差函数为

$$E(\mathbf{w}_{MAP}) = -\sum_{n=1}^N \{t_n \ln y_n + (1-t_n) \ln(1-y_n)\} + \frac{\alpha}{2} \mathbf{w}_{MAP}^T \mathbf{w}_{MAP} \quad (5.184)$$

其中 $y_n \equiv y(\mathbf{x}_n, \mathbf{w}_{MAP})$ 。关于 α ，最大化这个模型证据函数，可以得到公式 (5.178) 给出的重估计方程。

使用模型证据的方法确定 α 的步骤如图5.22所示，所用的数据集在附录A中讨论。

最后，我们需要找到公式 (5.168) 定义的预测分布。与之前一样，由于网络函数的非线性的性质，积分是无法直接计算的。最简单的近似方法是假设后验概率非常窄，因此可以进行下面的近似

$$p(t | \mathbf{x}, \mathcal{D}) \simeq p(t | \mathbf{x}, \mathbf{w}_{MAP}) \quad (5.185)$$

然而，我们可以放宽这个假设，通过考虑后验概率分布的方差。在这种情况下，与回归问题的情形相同，对网络输出进行线性近似是不合适的，因为输出激活函数是logistic sigmoid函数，将输出限制在了区间(0, 1)。相反，我们对输出激活函数进行线性近似，形式为

$$a(\mathbf{x}, \mathbf{w}) \simeq a_{MAP}(\mathbf{x}) + \mathbf{b}^T(\mathbf{w} - \mathbf{w}_{MAP}) \quad (5.186)$$

其中， $a_{MAP}(\mathbf{x}) = a(\mathbf{x}, \mathbf{w}_{MAP})$ 以及向量 $\mathbf{b} \equiv \nabla a(\mathbf{x}, \mathbf{w}_{MAP})$ 都可以通过反向传播方法求出。

由于我们现在对 \mathbf{w} 的后验概率分布进行了高斯近似，并且 a 的模型是 \mathbf{w} 的线性函数，因此我们现在可以使用4.5.2节的结果。由神经网络的权值的分布引出的输出单元激活的值的分布为

$$p(a | \mathbf{x}, \mathcal{D}) = \int \delta(a - a_{MAP}(\mathbf{x}) - \mathbf{b}^T(\mathbf{x})(\mathbf{w} - \mathbf{w}_{MAP})) q(\mathbf{w} | \mathcal{D}) d\mathbf{w} \quad (5.187)$$

其中 $q(\mathbf{w} | \mathcal{D})$ 是公式 (5.187) 给出的对后验概率分布的高斯近似。根据4.5.2节，我们看到这个分布是一个高斯分布，均值为 $a_{MAP} \equiv a(\mathbf{x}, \mathbf{w}_{MAP})$ ，方差为

$$\sigma_a^2(\mathbf{x}) = \mathbf{b}^T(\mathbf{x}) \mathbf{A}^{-1} \mathbf{b}(\mathbf{x}) \quad (5.188)$$

最后，为了得到预测分布，我们必须对 a 进行积分

$$p(t = 1 | \mathbf{x}, \mathcal{D}) = \int \sigma(a) p(a | \mathbf{x}, \mathcal{D}) da \quad (5.189)$$

高斯分布与logistic sigmoid函数的卷积是无法计算的。于是我们将公式 (4.153) 给出的近似应用于公式 (5.189)，可得

$$p(t = 1 | \mathbf{x}, \mathcal{D}) = \sigma(\kappa(\sigma_a^2) a_{MAP}) \quad (5.190)$$

其中， $\kappa(\cdot)$ 由公式 (4.154) 定义。回忆一下， σ_a^2 和 \mathbf{b} 都是 \mathbf{x} 的函数。

图5.23给出了这种方法的一个例子。使用的数据集是附录A介绍的人工生成数据集。

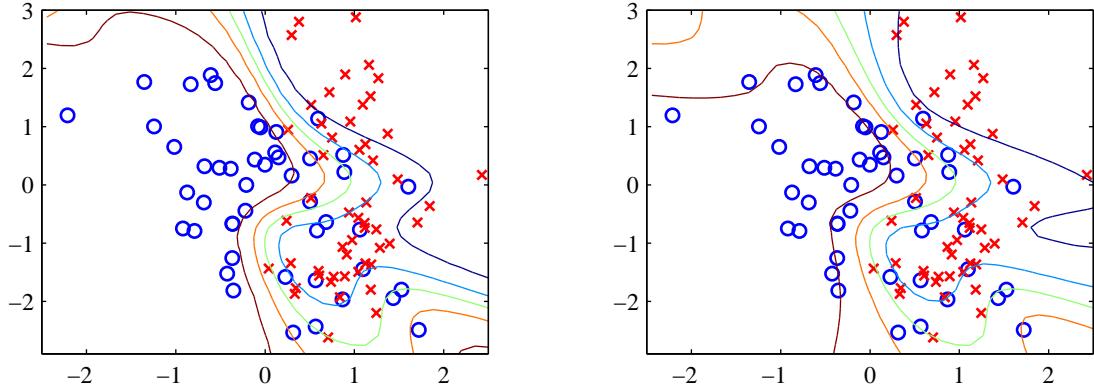


图 5.23: 对于一个具有8个隐含结点带有tanh激活函数和一个logistic sigmoid输出结点的贝叶斯网络应用拉普拉斯近似的说明。权参数使用缩放的共轭梯度方法得到，超参数 α 使用模型证据框架确定。左图是使用基于参数的 w_{MAP} 的点估计的简单近似 (5.185) 得到的结果，其中绿色曲线表示 $y = 0.5$ 的决策边界，其他的轮廓线对应于 $y = 0.1, 0.3, 0.7$ 和 0.9 的输出概率。右图是使用公式 (5.190) 得到的对应的结果。注意，求边缘概率分布的效果是扩散了轮廓线，使得预测的置信度变低，从而在每个输入点 x 处，后验概率分布向着 0.5 的方向偏移，而 $y = 0.5$ 的边界本身不受影响。

5.8 练习

(5.1) (***) 考虑形式为 (5.7) 的两层神经网络，其中隐含单元非线性激活函数 $h(\cdot)$ 为logistic sigmoid函数，形式为

$$\sigma(a) = \{1 + \exp(-a)\}^{-1} \quad (5.191)$$

证明，存在一个等价的网络，它计算了完全相同的函数，但是隐含单元激活函数为 $\tanh(a)$ ，其中 \tanh 函数由公式 (5.59) 定义。提示：首先找到 $\sigma(a)$ 与 $\tanh(a)$ 之间的关系，然后证明两个神经网络的参数的差别可以通过线性变换进行补偿。

(5.2) (*) 证明，在多输出神经网络的条件概率分布 (5.16) 下，最大化似然函数等价于最小化平方和误差函数 (5.11)。

(5.3) (***) 考虑一个涉及到多个目标变量的回归问题，其中我们假定，以输入向量 x 为条件，目标变量的概率分布是一个高斯分布，形式为

$$p(t | x, w) = \mathcal{N}(t | y(x, w), \Sigma) \quad (5.192)$$

其中， $y(x, w)$ 是神经网络的输出，输入向量为 x ，权向量为 w ， Σ 是目标变量上的假定高斯噪声的协方差。给定一组 x 和 t 的独立观测，写出为了找到 w 的最大似然解，我们必须最小化的误差函数的表达式，其中我们假定 Σ 固定且已知。现在假设 Σ 也需要从数据中确定，写下 Σ 的最大似然解的表达式。注意，现在关于 w 和 Σ 的优化过程偶合在了一起，这与5.2节讨论的独立目标变量的情形不同。

(5.4) (**) 考虑一个二分类问题，其中目标变量值为 $t \in \{0, 1\}$ ，网络输出 $y(x, w)$ 表示 $p(t = 1 | x)$ ，并且假设存在一个概率 ϵ 使得训练数据点的类别标签被标记错。假设数据集是独立同分布的，写出对于负对数似然函数的误差函数。验证误差函数 (5.21) 可以在 $\epsilon = 0$ 的时候得到。注意，这个误差函数使得模型对于错误标记的数据更加鲁棒，这与通常的误差函数不同。

(5.5) (*) 有一个多分类的神经网络模型，网络输出为 $y_k(x, w) = p(t_k = 1 | x)$ 。证明，对这个神经网络的似然函数进行最大化等价于对交叉熵误差函数 (5.24) 进行最小化。

(5.6) (*) 证明，误差函数 (5.21) 关于具有logistic sigmoid激活函数的输出单元的激活 a_k 的导数满足公式 (5.18)。

(5.7) (*) 证明，误差函数 (5.24) 关于具有softmax激活函数的输出单元的激活 a_k 的导数满足公式 (5.18)。

(5.8) (*) 从公式 (4.88) 中可以看出, logistic sigmoid激活函数的导数可以根据函数值本身表示。推导出公式 (5.59) 定义的tanh激活函数的对应结果。

(5.9) (*) 二分类问题的误差函数 (5.21) 是针对具有logistic sigmoid输出激活函数的神经网络推导的, 从而 $0 \leq y(\mathbf{x}, \mathbf{w}) \leq 1$, 并且数据具有目标值 $t \in \{0, 1\}$ 。如果我们考虑一个神经网络, 它的输出满足 $-1 \leq y(\mathbf{x}, \mathbf{w}) \leq 1$, 且对于类别 \mathcal{C}_1 , 有 $t = 1$, 对于类别 \mathcal{C}_2 , 有 $t = -1$, 推导对应的误差函数。输出单元激活函数的合适的选择是什么?

(5.10) (*) 考虑特征方程为 (5.33) 的Hessian矩阵 \mathbf{H} 。通过令公式 (5.39) 中的向量 \mathbf{v} 等于每个特征向量 \mathbf{u}_i , 证明 \mathbf{H} 是正定的当且仅当它的特征值全部为正。

(5.11) (**) 考虑公式 (5.32) 定义的二次误差函数, 其中Hessian矩阵 \mathbf{H} 的特征值方程为 (5.33)。证明, 对应的常数误差函数的轮廓线是椭圆, 椭圆的轴与特征向量 \mathbf{u}_i 对齐, 长度与对应的特征值 λ_i 的平方根成反比。

(5.12) (**) 通过考虑误差函数在驻点 \mathbf{w}^* 处的局部泰勒展开 (5.32), 证明驻点是误差函数的局部极小值的充要条件是, 公式 (5.30) 定义的Hessian矩阵 \mathbf{H} 是正定的, 其中 $\hat{\mathbf{w}} = \mathbf{w}^*$ 。

(5.13) (*) 证明, 由于Hessian矩阵 \mathbf{H} 具有对称性, 二次误差函数 (5.28) 的独立元素的数量为 $\frac{W(W+3)}{2}$ 。

(5.14) (*) 通过计算泰勒展开式, 验证公式 (5.69) 右侧中的 $O(\epsilon)$ 项被消去。

(5.15) (**) 在5.3.4节, 我们推导了使用反向传播方法计算神经网络的Jacobian矩阵的步骤。使用正向传播的方程, 推导出计算Jacobian矩阵的步骤。

(5.16) (*) 使用平方和误差函数的神经网络的Hessian矩阵的外积近似由公式 (5.84) 给出。将这个结果推广到多个输出的情形。

(5.17) (*) 考虑下面形式的平方损失函数

$$E = \frac{1}{2} \iint \{y(\mathbf{x}, \mathbf{w}) - t\}^2 p(\mathbf{x}, t) \, d\mathbf{x} \, dt \quad (5.193)$$

其中 $y(\mathbf{x}, \mathbf{w})$ 是参数化的函数, 例如神经网络。公式 (1.89) 给出的结果表明, 使误差达到最小值的函数 $y(\mathbf{x}, \mathbf{w})$ 等于给定 \mathbf{x} 的条件下 t 的条件期望。使用这个结果证明 E 关于向量 \mathbf{w} 的两个元素 w_r 和 w_s 的二阶导数为

$$\frac{\partial^2 E}{\partial w_r \partial w_s} = \int \frac{\partial y}{\partial w_r} \frac{\partial y}{\partial w_s} p(\mathbf{x}) \, d\mathbf{x} \quad (5.194)$$

注意, 对于来自 $p(\mathbf{x})$ 的有限样本, 我们可以得到公式 (5.84)。

(5.18) (*) 考虑图5.1所述的两层神经网络, 加上一些额外的参数, 对应于从输入直接到输出的跨层链接。通过扩展5.3.2节的讨论, 写出误差函数关于这些附加的参数的导数的方程。

(5.19) (*) 考虑这样一个神经网络, 它具有一个输出单元, 输出单元激活函数为logistic sigmoid函数, 误差函数为交叉熵误差函数。推导对于这个网络的Hessian矩阵的外积近似的表达式 (5.85), 这对应于平方和误差函数的结果 (5.84)。

(5.20) (*) 考虑这样一个神经网络, 它具有 K 个输出单元, 输出单元激活函数为softmax函数, 误差函数为交叉熵误差函数。推导对于这个网络的Hessian矩阵的外积近似的表达式, 它对应于平方和误差函数的结果 (5.84)。

(5.21) (***) 将Hessian矩阵的外积近似的表达式推广到 $K > 1$ 个输出单元的情形。从而, 推导出一个表达式使得公式 (5.87) 能够被用于将来自各个输出单元以及各个模式的贡献顺序地整合到一起。这个表达式与恒等式 (5.88) 一起, 使得我们可以用公式 (5.89), 通过顺序地整合来自各个输出和模式的贡献, 求出Hessian矩阵的逆矩阵。

(5.22) (**) 使用微积分的链式规则, 推导公式 (5.93)、(5.94) 和 (5.95) 给出的关于两层前馈网络的Hessian矩阵的结果。

(5.23) (**) 将5.4.5节给出的两层神经网络的精确的Hessian矩阵的结果进行推广, 使其包含从输入直接到输出的跨层链接。

(5.24) (*) 验证公式 (5.113) 和 (5.114) 定义的网络函数在将变换 (5.115) 作用于输入的情形下具有不变性, 只要权值和偏置同时使用公式 (5.116) 和 (5.117) 进行变换即可。类似地, 证明网络输出可以根据公式 (5.118) 进行变换, 方法是将公式 (5.119) 和 (5.120) 给出的变换作用于第二层的权值和偏置。

(5.25) (***) 考虑下面形式的二次误差函数

$$E = E_0 + \frac{1}{2}(\mathbf{w} - \mathbf{w}^*)^T \mathbf{H}(\mathbf{w} - \mathbf{w}^*) \quad (5.195)$$

其中 \mathbf{w}^* 表示最小值, Hessian 矩阵 \mathbf{H} 是正定的, 并且是常量。假设初始权向量 $\mathbf{w}^{(0)}$ 被选在原点处, 并且使用简单的梯度下降法进行更新

$$\mathbf{w}^{(\tau)} = \mathbf{w}^{(\tau-1)} - \rho \nabla E \quad (5.196)$$

其中 τ 表示迭代步骤数, ρ 是学习率 (假设很小)。证明, 在 τ 步之后, 与 \mathbf{H} 的特征向量平行的权向量的元素可以写成

$$w_j^{(\tau)} = \{1 - (1 - \rho\eta_j)^\tau\} w_j^* \quad (5.197)$$

其中 $w_j = \mathbf{w}^T \mathbf{u}_j$, 且 \mathbf{u}_j 和 η_j 分别是 \mathbf{H} 的特征向量和特征值, 从而

$$\mathbf{H}\mathbf{u}_j = \eta_j \mathbf{u}_j \quad (5.198)$$

证明, 当 $\tau \rightarrow \infty$ 时, 会得到 $\mathbf{w}^{(\tau)} \rightarrow \mathbf{w}^*$, 这与预期相符, 其中我们假设 $|1 - \rho\eta_j| < 1$ 。现在, 假设训练在有限的 τ 步骤之后停止。证明, 与 Hessian 矩阵的特征向量平行的权向量的元素满足

$$w_j^{(\tau)} \simeq w_j^* \quad \text{当 } \eta_j \gg (\rho\tau)^{-1} \text{ 时} \quad (5.199)$$

$$|w_j^{(\tau)}| \ll |w_j^*| \quad \text{当 } \eta_j \ll (\rho\tau)^{-1} \text{ 时} \quad (5.200)$$

将这个结果与 3.5.3 节关于简单的权值衰减的正则化的讨论进行对比, 从而证明 $(\rho\tau)^{-1}$ 类似于正则化参数 λ 。上述结果也表明, 公式 (3.91) 定义的网络中的参数的有效数量随着训练的进行而增大。

(5.26) (**) 考虑一个多层感知器网络, 具有任意的前馈拓扑结构, 使用最小化切向传播误差函数 (5.127) 的方式进行训练, 其中正则化函数由公式 (5.128) 给出。证明, 正则化项 Ω 可以写成模式上的求和式, 形式为

$$\Omega_n = \frac{1}{2} \sum_k (\mathcal{G}y_k)^2 \quad (5.201)$$

其中, \mathcal{G} 是微分算符, 定义为

$$\mathcal{G} \equiv \sum_i \tau_i \frac{\partial}{\partial x_i} \quad (5.202)$$

通过将算符 \mathcal{G} 作用于正向传播方程

$$z_j = h(a_j), \quad a_j = \sum_i w_{ji} z_i \quad (5.203)$$

证明, Ω_n 可以通过正向传播来计算, 计算时使用下面的方程

$$\alpha_j = h'(a_j) \beta_j, \quad \beta_j = \sum_i w_{ji} \alpha_i \quad (5.204)$$

其中我们已经定义了新的变量

$$\alpha_j \equiv \mathcal{G}z_j, \quad \beta_j \equiv \mathcal{G}a_j \quad (5.205)$$

现在, 证明 Ω_n 关于网络的权值 w_{rs} 的导数可以写成

$$\frac{\partial \Omega_n}{\partial w_{rs}} = \sum_k \alpha_k \{\phi_{kr} z_s + \delta_{kr} \alpha_s\} \quad (5.206)$$

其中我们已经定义

$$\delta_{kr} \equiv \frac{\partial y_k}{\partial a_r}, \quad \phi_{kr} \equiv \mathcal{G}\delta_{kr} \quad (5.207)$$

写出 δ_{kr} 的反向传播方程，从而推导出计算 ϕ_{kr} 的一组反向传播方程。

(5.27) (***) 考虑使用变换的数据进行训练的框架，其中变换的过程仅仅是增加一个随机噪声 $\mathbf{x} \rightarrow \mathbf{x} + \boldsymbol{\xi}$ ，其中 $\boldsymbol{\xi}$ 是一个高斯分布，均值为0，并且具有单位协方差。通过使用与5.5.5节类似的推导，证明得到的正则化项变成了Tikhonov形式 (5.135)。

(5.28) (*) 考虑一个神经网络，例如5.5.6节讨论的卷积网络，其中多个权值被限制为具有相同的值。为了确保在计算误差函数关于网络的可调节参数的导数时，这些限制条件能够满足，讨论应该对标准的反向传播算法进行怎样的修改。

(5.29) (*) 验证公式 (5.141) 给出的结果。

(5.30) (*) 验证公式 (5.142) 给出的结果。

(5.31) (*) 验证公式 (5.143) 给出的结果。

(5.32) (**) 证明，公式 (5.146) 定义的混合系数 $\{\pi_k\}$ 关于辅助参数 $\{\eta_j\}$ 的导数为

$$\frac{\partial \pi_k}{\partial \eta_j} = \delta_{jk}\pi_j - \pi_j\pi_k \quad (5.208)$$

因此，通过使用限制条件 $\sum_k \gamma_k(w_i) = 1$ （对于所有*i*都成立），推导公式 (5.147) 给出的结果。

(5.33) (*) 写出一对方程，用连接角 θ_1 和 θ_2 以及连接长度 L_1 和 L_2 表示图5.18所示的机械臂的笛卡尔坐标 (x_1, x_2) 。假设坐标系的原点由下方的机械臂的连接点给出，那么这些方程定义了机械臂的“正向运动学”。

(5.34) (*) 推导误差函数关于控制混合密度网络的混合系数的网络输出激活的导数 (5.155)。

(5.35) (*) 推导误差函数关于控制混合密度网络的分量均值的网络输出激活的导数 (5.156)。

(5.36) (*) 推导误差函数关于控制混合密度网络的分量方差的网络输出激活的导数 (5.157)。

(5.37) (*) 验证公式 (5.158) 和 (5.160) 给出的混合密度网络模型的条件均值和方差。

(5.38) (*) 使用一般的结果 (2.115)，推导贝叶斯神经网络模型的拉普拉斯近似的预测分布 (5.172)。

(5.39) (*) 使用拉普拉斯近似的结果 (4.135) 证明贝叶斯神经网络模型的超参数 α 和 β 的证据函数可以用 (5.175) 近似。

(5.40) (*) 为了将5.7.3节讨论的贝叶斯神经网络的框架推广到使用softmax输出单元激活函数的多类问题的网络中，说出需要进行的修改。

(5.41) (**) 遵照5.7.1节和5.7.2节给出的回归网络的类似的步骤，推导边缘似然函数的结果 (5.183)，其中网络具有交叉熵误差函数以及logistic sigmoid输出单元激活函数。

6 核方法

在第3章和第4章，我们考虑了回归问题和分类问题的线性参数模型，其中从输入 \mathbf{x} 到输出 y 的映射 $y(\mathbf{x}, \mathbf{w})$ 的形式由可调节参数构成的向量 \mathbf{w} 控制。在学习阶段，一组训练数据用来得到参数向量的点估计，或者用来确定这个向量的后验概率分布。然后，训练数据之后被丢弃，对于新输入的预测纯粹依靠学习到的参数向量 \mathbf{w} 。这个方法也被用于非线性参数模型，例如神经网络。

然而，有这样一类模式识别的技术：训练数据点或者它的一个子集在预测阶段仍然保留并且被使用。例如，由“核”函数的线性组合构成的Parzen概率密度模型，其中每一个核函数都以训练数据点为中心。类似地，在2.5.2节，我们介绍了一种简单的分类方法，即最近邻方法。这种方法把每个新的测试向量分配为训练数据集中距离最近的样本的标签。这些都是基于存储（memory-based）的方法的例子。基于存储的方法把整个训练数据存储起来，用来对未来的数据点进行预测。通常这种方法需要一个度量，来定义输入空间任意两个向量之间的相似度。这种方法通常“训练”速度很快，但是对测试数据点的预测速度很慢。

许多线性参数模型可以被转化为一个等价的“对偶表示”。对偶表示中，预测的基础也是在训练数据点处计算的核函数（kernel function）的线性组合。正如我们将看到的那样，对于基于固定非线性特征空间（feature space）映射 $\phi(\mathbf{x})$ 的模型来说，核函数由下面的关系给出。

$$k(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^T \phi(\mathbf{x}') \quad (6.1)$$

根据这个定义，我们看到核函数关于它的参数是对称的，即 $k(\mathbf{x}, \mathbf{x}') = k(\mathbf{x}', \mathbf{x})$ 。核的概念由Aizerman et al. (1964) 引入模式识别领域。那篇文章介绍了势函数的方法。之所以被称为势函数，是因为它类似于静电学中的概念。虽然被忽视了很多年，但是Boser et al. (1992) 在大边缘分类器的问题中把它重新引入到了极其学习领域。那篇文章提出了支持向量机（support vector machine）的方法。从那时起，这个话题在理论上和实用上都吸引了大家的兴趣。一个最重要的发展是把核方法进行了扩展，使其能处理符号化的物体，从而极大地扩展了这种方法能处理的问题的范围。

通过考虑公式(6.1)中特征空间的恒等映射 $\phi(\mathbf{x}) = \mathbf{x}$ ，我们就得到了核函数的一个最简单的例子，此时 $k(\mathbf{x}, \mathbf{x}') = \mathbf{x}^T \mathbf{x}'$ 。我们把这个称为线性核。

用特征空间的内积的方式表示核的概念使得我们能够对许多著名的算法进行有趣的扩展。扩展的方法是使用核技巧（kernel trick），也被称为核替换（kernel substitution）。一般的思想是，如果我们有一个算法，它的输入向量 \mathbf{x} 只以标量积的形式出现，那么我们可以用一些其他的核来替换这个标量积。例如，核替换方法可以用于主成分分析，从而产生了PCA的非线性变种（Schölkopf et al., 1998）。核替换的其他例子包括最近邻分类器和核Fisher判别函数（Mika et al., 1999; Roth and Steinhage, 2000; Baudat and Anouar, 2000）。

常用的核函数有各种不同的形式，我们会在本章中遇到若干个核函数的例子。许多核函数只是参数的差值的函数，即 $k(\mathbf{x}, \mathbf{x}') = k(\mathbf{x} - \mathbf{x}')$ ，这被称为静止核（stationary kernel），因为核函数对于输入空间的平移具有不变性。另一种核函数是同质核（homogeneous kernel），也被称为径向基函数（radial basis function），它只依赖于参数之间的距离（通常是欧几里得距离）的大小，即 $k(\mathbf{x}, \mathbf{x}') = k(\|\mathbf{x} - \mathbf{x}'\|)$ 。

最近的关于核方法的教材有Schölkopf and Smola (2002)，Herbrich (2002) 和Shawe-Taylor and Cristianini (2004)。

6.1 对偶表示

许多回归的线性模型和分类的线性模型的公式都可以使用对偶表示重写。使用对偶表示形式，核函数可以自然地产生。在我们下一章中讨论支持向量机的时候，这个概念十分重要。这里，我们考虑一个线性模型，它的参数通过最小化正则化的平方和误差函数来确定。正则化的平方和误差函数为

$$J(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^N \{\mathbf{w}^T \phi(\mathbf{x}_n) - t_n\}^2 + \frac{\lambda}{2} \mathbf{w}^T \mathbf{w} \quad (6.2)$$

其中 $\lambda \geq 0$ 。如果我们令 $J(\mathbf{w})$ 关于 \mathbf{w} 的梯度等于零，那么我们看到 \mathbf{w} 的解是向量 $\phi(\mathbf{x}_n)$ 的线性组合的形式，系数是 \mathbf{w} 的函数，形式为

$$\mathbf{w} = -\frac{1}{\lambda} \sum_{n=1}^N \{\mathbf{w}^T \phi(\mathbf{x}_n) - t_n\} \phi(\mathbf{x}_n) = \sum_{n=1}^N a_n \phi(\mathbf{x}_n) = \Phi^T \mathbf{a} \quad (6.3)$$

其中 Φ 是设计矩阵，第 n 行为 $\phi(\mathbf{x}_n)^T$ 。这里，向量 $\mathbf{a} = (a_1, \dots, a_N)^T$ ，并且我们定义了

$$a_n = -\frac{1}{\lambda} \{\mathbf{w}^T \phi(\mathbf{x}_n) - t_n\} \quad (6.4)$$

我们现在不直接对参数向量 \mathbf{w} 进行操作，而是使用参数向量 \mathbf{a} 重新整理最小平方算法，得到一个对偶表示（dual representation）。如果我们将 $\mathbf{w} = \Phi^T \mathbf{a}$ 代入 $J(\mathbf{w})$ ，那么可以得到

$$J(\mathbf{a}) = \frac{1}{2} \mathbf{a}^T \Phi \Phi^T \Phi \Phi^T \mathbf{a} - \mathbf{a}^T \Phi \Phi^T \mathbf{t} + \frac{1}{2} \mathbf{t}^T \mathbf{t} + \frac{\lambda}{2} \mathbf{a}^T \Phi \Phi^T \mathbf{a} \quad (6.5)$$

其中 $\mathbf{t} = (t_1, \dots, t_N)^T$ 。我们现在定义Gram矩阵 $\mathbf{K} = \Phi \Phi^T$ ，它是一个 $N \times N$ 的对称矩阵，元素为

$$K_{nm} = \phi(\mathbf{x}_n)^T \phi(\mathbf{x}_m) = k(\mathbf{x}_n, \mathbf{x}_m) \quad (6.6)$$

其中我们引入了公式(6.1)定义的核函数（kernel function） $k(\mathbf{x}, \mathbf{x}')$ 。使用Gram矩阵，平方和误差函数可以写成

$$J(\mathbf{a}) = \frac{1}{2} \mathbf{a}^T \mathbf{K} \mathbf{K} \mathbf{a} - \mathbf{a}^T \mathbf{K} \mathbf{t} + \frac{1}{2} \mathbf{t}^T \mathbf{t} + \frac{\lambda}{2} \mathbf{a}^T \mathbf{K} \mathbf{a} \quad (6.7)$$

使用公式(6.3)从公式(6.4)中消去 \mathbf{w} ，求解 \mathbf{a} ，我们有

$$\mathbf{a} = (\mathbf{K} + \lambda \mathbf{I}_N)^{-1} \mathbf{t} \quad (6.8)$$

如果我们将这个代入线性回归模型中，对于新的输入 \mathbf{x} ，我们得到了下面预测

$$y(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x}) = \mathbf{a}^T \Phi \phi(\mathbf{x}) = \mathbf{k}(\mathbf{x})^T (\mathbf{K} + \lambda \mathbf{I}_N)^{-1} \mathbf{t} \quad (6.9)$$

其中我们定义了向量 $\mathbf{k}(\mathbf{x})$ ，它的元素为 $k_n(\mathbf{x}) = k(\mathbf{x}_n, \mathbf{x})$ 。因此我们看到对偶公式使得最小平方问题的解完全通过核函数 $k(\mathbf{x}, \mathbf{x}')$ 表示。这被称为对偶公式，因为 \mathbf{a} 的解可以被表示为 $\phi(\mathbf{x})$ 的线性组合，从而我们可以使用参数向量 \mathbf{w} 恢复出原始的公式。注意，在 \mathbf{x} 处的预测由训练集数据的目标值的线性组合给出。实际上，我们已经在3.3节中得到过这个结果，只不过记号稍微不同。

在对偶公式中，我们通过对一个 $N \times N$ 的矩阵求逆来确定参数向量 \mathbf{a} ，而在原始参数空间公式中，我们要对一个 $M \times M$ 的矩阵求逆来确定 \mathbf{w} 。由于 N 通常远大于 M ，因此对偶公式似乎没有实际用处。然而，正如我们将要看到的那样，对偶公式的优点是，它可以完全通过核函数 $k(\mathbf{x}, \mathbf{x}')$ 来表示。于是，我们可以直接针对核函数进行计算，避免了显式地引入特征向量 $\phi(\mathbf{x})$ ，这使得我们可以隐式地使用高维特征空间，甚至无限维特征空间。

基于Gram矩阵的对偶表示的存在是许多线性模型的性质，包括感知器。在6.4节，我们会研究回归的概率线性模型和高斯过程方法的对偶性。当我们在第7章讨论支持向量机的时候，对偶性也起着重要的作用。

6.2 构造核

为了利用核替换，我们需要能够构造合法的核函数。一种方法是选择一个特征空间映射 $\phi(\mathbf{x})$ ，然后使用这个映射寻找对应的核，如图6.1所示。这里，一维空间的核函数被定义为

$$k(x, x') = \phi(x)^T \phi(x') = \sum_{i=1}^M \phi_i(x) \phi_i(x') \quad (6.10)$$

其中 $\phi_i(x)$ 是基函数。

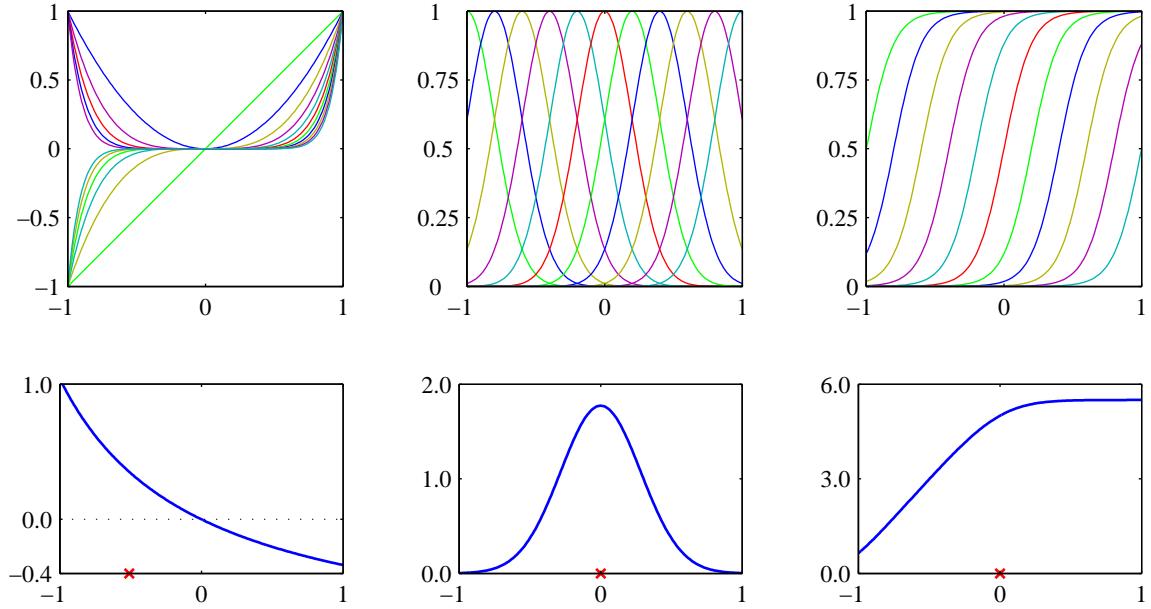


图 6.1: 从对应的基函数集合构建核函数的例子。在每一列中，下图给出了由公式 (6.10) 定义的核函数 $k(x, x')$ ，它是 x 的函数， x' 的值用红色叉号表示，而上图给出了对应的基函数，分别是多项式基函数（左列）、高斯基函数（中列）、logistic sigmoid 基函数（右列）。

另一种方法是直接构造核函数。在这种情况下，我们必须确保我们核函数是合法的，即它对应于某个（可能是无穷维）特征空间的标量积。作为一个简单的例子，考虑下面的核函数

$$k(\mathbf{x}, \mathbf{z}) = (\mathbf{x}^T \mathbf{z})^2 \quad (6.11)$$

如果我们取二维输入空间 $\mathbf{x} = (x_1, x_2)$ 的特殊情况，那么我们可以展开这一项，于是得到对应的非线性特征映射

$$\begin{aligned} k(\mathbf{x}, \mathbf{z}) &= (\mathbf{x}^T \mathbf{z})^2 = (x_1 z_1 + x_2 z_2)^2 \\ &= x_1^2 z_1^2 + 2x_1 z_1 x_2 z_2 + x_2^2 z_2^2 \\ &= (x_1^2, \sqrt{2}x_1 x_2, x_2^2)(z_1^2, \sqrt{2}z_1 z_2, z_2^2)^T \\ &= \phi(\mathbf{x})^T \phi(\mathbf{z}) \end{aligned} \quad (6.12)$$

我们看到特征映射的形式为 $\phi(\mathbf{x}) = (x_1^2, \sqrt{2}x_1 x_2, x_2^2)^T$ ，因此这个特征映射由所有的二阶项组成，每个二阶项有一个具体的系数。

但是，更一般地，我们需要找到一种更简单的方法检验一个函数是否是一个合法的核函数，而不需要显示地构造函数 $\phi(\mathbf{x})$ 。核函数 $k(\mathbf{x}, \mathbf{x}')$ 是一个合法的核函数的充分必要条件是 Gram 矩阵（元素由 $k(\mathbf{x}_n, \mathbf{x}_m)$ 给出）在所有的集合 $\{\mathbf{x}_n\}$ 的选择下都是半正定的 (Shawe-Taylor and Cristianini, 2004)。注意，一个半正定的矩阵与元素全部非负的矩阵不同。

构造新的核函数的一个强大的方法是使用简单的核函数作为基本的模块来构造。可以使用下面的性质来完成这件事。

给定合法的核 $k_1(\mathbf{x}, \mathbf{x}')$ 和 $k_2(\mathbf{x}, \mathbf{x}')$ ，下面的新核也是合法的

$$k(\mathbf{x}, \mathbf{x}') = c k_1(\mathbf{x}, \mathbf{x}') \quad (6.13)$$

$$k(\mathbf{x}, \mathbf{x}') = f(\mathbf{x}) k_1(\mathbf{x}, \mathbf{x}') f(\mathbf{x}') \quad (6.14)$$

$$k(\mathbf{x}, \mathbf{x}') = q(k_1(\mathbf{x}, \mathbf{x}')) \quad (6.15)$$

$$k(\mathbf{x}, \mathbf{x}') = \exp(k_1(\mathbf{x}, \mathbf{x}')) \quad (6.16)$$

$$k(\mathbf{x}, \mathbf{x}') = k_1(\mathbf{x}, \mathbf{x}') + k_2(\mathbf{x}, \mathbf{x}') \quad (6.17)$$

$$k(\mathbf{x}, \mathbf{x}') = k_1(\mathbf{x}, \mathbf{x}')k_2(\mathbf{x}, \mathbf{x}') \quad (6.18)$$

$$k(\mathbf{x}, \mathbf{x}') = k_3(\phi(\mathbf{x}), \phi(\mathbf{x}')) \quad (6.19)$$

$$k(\mathbf{x}, \mathbf{x}') = \mathbf{x}^T \mathbf{A} \mathbf{x}' \quad (6.20)$$

$$k(\mathbf{x}, \mathbf{x}') = k_a(\mathbf{x}_a, \mathbf{x}'_a) + k_b(\mathbf{x}_b, \mathbf{x}'_b) \quad (6.21)$$

$$k(\mathbf{x}, \mathbf{x}') = k_a(\mathbf{x}_a, \mathbf{x}'_a)k_b(\mathbf{x}_b, \mathbf{x}'_b) \quad (6.22)$$

其中 $c > 0$ 是一个常数， $f(\cdot)$ 是任意函数， $q(\cdot)$ 是一个系数非负的多项式， $\phi(\mathbf{x})$ 是一个从 \mathbf{x} 到 \mathbb{R}^M 的函数， $k_3(\cdot, \cdot)$ 是 \mathbb{R}^M 中的一个合法的核， \mathbf{A} 是一个对称半正定矩阵， \mathbf{x}_a 和 \mathbf{x}_b 是变量（未必互斥），且 $\mathbf{x} = (\mathbf{x}_a, \mathbf{x}_b)$ 。 k_a 和 k_b 是各自空间的合法的核函数。

知道了这些性质，我们现在可以开始构造适用于具体应用的更复杂的核函数了。我们要求核 $k(\mathbf{x}, \mathbf{x}')$ 是对称的半正定的，并且它表示面向具体应用中 \mathbf{x} 和 \mathbf{x}' 之间的适当形式的相似性。这里，我们考虑核函数的几个常见的例子。关于“核工程”的一个更加广泛的讨论，可以参考 Shawe-Taylor and Cristianini (2004)。

我们看到简单的多项式核 $k(\mathbf{x}, \mathbf{x}') = (\mathbf{x}^T \mathbf{x}')^2$ 值包含二次项。如果我们考虑稍微一般的核 $k(\mathbf{x}, \mathbf{x}') = (\mathbf{x}^T \mathbf{x}' + c)^2$ ，其中 $c > 0$ ，那么对应的特征映射 $\phi(\mathbf{x})$ 就会包含常数、线性项和二阶项。类似地， $k(\mathbf{x}, \mathbf{x}') = (\mathbf{x}^T \mathbf{x}')^M$ 包含所有 M 阶的单项式。例如，如果 \mathbf{x} 和 \mathbf{x}' 是两张图片，那么这个核表示第一张图片中 M 个像素和第二张图片中 M 个像素的所有可能的乘积的一个特定的加权和。这个可以类似地进行推广，使其包含所有次数最高为 M 的项。推广的方式为 $k(\mathbf{x}, \mathbf{x}') = (\mathbf{x}^T \mathbf{x}' + c)^M$ ，其中 $c > 0$ 。使用公式 (6.17) 和公式 (6.18) 给出的将核函数进行组合的方法，我们看到这些都是合法的核函数。

另一个经常使用的核函数的形式为

$$k(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\sigma^2}\right) \quad (6.23)$$

这个经常被称为高斯核。但是注意，在我们现在的讨论中，它不表示概率密度，因此归一化系数被省略了。这是一个合法的核，理由如下。我们把平方项展开

$$\|\mathbf{x} - \mathbf{x}'\|^2 = \mathbf{x}^T \mathbf{x} + (\mathbf{x}')^T \mathbf{x}' - 2\mathbf{x}^T \mathbf{x}' \quad (6.24)$$

从而

$$k(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\mathbf{x}^T \mathbf{x}}{2\sigma^2}\right) \exp\left(\frac{\mathbf{x}^T \mathbf{x}'}{\sigma^2}\right) \exp\left(-\frac{(\mathbf{x}')^T \mathbf{x}'}{2\sigma^2}\right) \quad (6.25)$$

然后使用公式 (6.14) 和公式 (6.16)，以及线性核 $k(\mathbf{x}, \mathbf{x}') = \mathbf{x}^T \mathbf{x}'$ 的合法性，即可看到高斯核是一个合法的核。注意，对应于高斯核的特征向量有无穷的维数。

高斯核并不局限于使用欧几里得距离。如果我们使用公式 (6.24) 中的核替换，将 $\mathbf{x}^T \mathbf{x}'$ 替换为一个非线性核 $\kappa(\mathbf{x}, \mathbf{x}')$ ，我们有

$$k(\mathbf{x}, \mathbf{x}') = \exp\left\{-\frac{1}{2\sigma^2}(\kappa(\mathbf{x}, \mathbf{x}) + \kappa(\mathbf{x}', \mathbf{x}') - 2\kappa(\mathbf{x}, \mathbf{x}'))\right\} \quad (6.26)$$

核观点的一个重要的贡献是可以扩展到符号化的输入，而不是简单的实数向量。核函数可以定义在多种对象上，例如图片、集合、字符串、文本文档。例如，考虑一个固定的集合，定义一个非向量空间，这个空间由这个集合的所有可能的子集构成。如果 A_1 和 A_2 是两个这样的子集，那么核的一个简单的选择可以是

$$k(A_1, A_2) = 2^{|A_1 \cap A_2|} \quad (6.27)$$

其中 $A_1 \cap A_2$ 表示集合 A_1 和 A_2 的交集， $|A|$ 表示 A 的元素的数量。这是一个合法的核，因为可以证明它对应于一个特征空间中的一个内积。

构造核的另一个强大的方法是从一个概率生成式模型开始构造 (Haussler, 1999) , 这使得我们可以在一个判别式的框架中使用生成式模型。生成式模型可以自然地处理缺失数据, 并且在隐马尔科夫模型的情况下, 可以处理长度变化的序列。相反, 判别式模型在判别式的任务中通常会比生成式模型的表现更好。于是, 将这两种方法结合吸引了一些人的兴趣 (Lasserre et al., 2006)。一种将二者结合的方法是使用一个生成式模型定义一个核, 然后在判别式方法中使用这个核。

给定一个生成式模型 $p(\mathbf{x})$, 我们可以定义一个核

$$k(\mathbf{x}, \mathbf{x}') = p(\mathbf{x})p(\mathbf{x}') \quad (6.28)$$

很明显, 这是一个合法的核, 因为我们可以把它看成由映射 $p(\mathbf{x})$ 定义的一维特征空间中的一个内积。它表明, 如果两个输入 \mathbf{x} 和 \mathbf{x}' 都具有较高的概率, 那么它们就是相似的。我们可以使用公式 (6.13) 和公式 (6.17) 扩展这类核。扩展的方法是考虑不同概率分布的乘积的加和, 带有正的权值系数 $p(i)$, 形式为

$$k(\mathbf{x}, \mathbf{x}') = \sum_i p(\mathbf{x} | i)p(\mathbf{x}' | i)p(i) \quad (6.29)$$

如果不考虑一个整体的乘法常数, 这个核就等价于一个混合概率密度, 它可以分解成各个分量概率密度, 下标 i 扮演着“潜在”变量的角色。如果两个输入 \mathbf{x} 和 \mathbf{x}' 在一大类的不同分量下都有较大的概率, 那么这两个输入将会使核函数输出较大的值, 因此就表现出相似性。在无限求和的极限情况下, 我们也可以考虑下面形式的核函数

$$k(\mathbf{x}, \mathbf{x}') = \int p(\mathbf{x} | \mathbf{z})p(\mathbf{x}' | \mathbf{z})p(\mathbf{z}) d\mathbf{z} \quad (6.30)$$

其中 \mathbf{z} 是一个连续潜在变量。

现在假设我们的数据由长度为 L 的有序序列组成, 即一个观测为 $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_L\}$ 。对于这种序列, 一个流行的生成式模型是隐马尔科夫模型, 它把概率 $p(\mathbf{X})$ 表示为对应的隐含状态序列 $\mathbf{Z} = \{z_1, \dots, z_L\}$ 上的积分或求和。我们可以使用这种方法定义一个核函数来度量两个序列 \mathbf{X} 和 \mathbf{X}' 的相似度。定义核函数的方法是扩展混合表示 (6.29), 得到

$$k(\mathbf{X}, \mathbf{X}') = \sum_{\mathbf{Z}} p(\mathbf{X} | \mathbf{Z})p(\mathbf{X}' | \mathbf{Z})p(\mathbf{Z}) \quad (6.31)$$

从而两个观测序列都通过相同的隐含序列 \mathbf{Z} 生成。这个模型很容易扩展, 使其能够比较不同长度的序列。

另一个使用生成式模型定义核函数的方法被称为Fisher核 (Jaakkola and Haussler, 1999)。考虑一个参数生成式模型 $p(\mathbf{x} | \boldsymbol{\theta})$, 其中 $\boldsymbol{\theta}$ 表示参数的向量。目标是找到一个核, 度量这个生成式模型的两个输入变量 \mathbf{x} 和 \mathbf{x}' 之间的相似性。Jaakkola and Haussler (1999) 考虑关于 $\boldsymbol{\theta}$ 的梯度, 它定义了“特征”空间的一个向量, 这个特征空间的维度与 $\boldsymbol{\theta}$ 的维度相同。特别地, 它们考虑 Fisher 得分

$$\mathbf{g}(\boldsymbol{\theta}, \mathbf{x}) = \nabla_{\boldsymbol{\theta}} \ln p(\mathbf{x} | \boldsymbol{\theta}) \quad (6.32)$$

根据 Fisher 得分, Fisher 核被定义为

$$k(\mathbf{x}, \mathbf{x}') = \mathbf{g}(\boldsymbol{\theta}, \mathbf{x})^T \mathbf{F}^{-1} \mathbf{g}(\boldsymbol{\theta}, \mathbf{x}') \quad (6.33)$$

这里, \mathbf{F} 是 Fisher 信息矩阵 (Fisher information matrix), 定义为

$$\mathbf{F} = \mathbb{E}_{\mathbf{x}}[\mathbf{g}(\boldsymbol{\theta}, \mathbf{x})\mathbf{g}(\boldsymbol{\theta}, \mathbf{x})^T] \quad (6.34)$$

其中, 期望是在概率分布 $p(\mathbf{x} | \boldsymbol{\theta})$ 下关于 \mathbf{x} 的期望。这样定义的动机可以从信息几何 (information geometry) 的角度看出来 (Amari, 1998), 它考虑了模型参数空间的微分几何。这里, 我们注意到, Fisher 信息矩阵的存在使得这个核在密度模型的非线性重参数化 $\boldsymbol{\theta} \rightarrow \psi(\boldsymbol{\theta})$ 下具有不变性。

在实际应用中，通常计算Fisher信息矩阵是不可行的。一种方法是把Fisher信息的定义中的期望替换为样本均值，可得

$$\mathbf{F} \simeq \frac{1}{N} \sum_{n=1}^N \mathbf{g}(\boldsymbol{\theta}, \mathbf{x}_n) \mathbf{g}(\boldsymbol{\theta}, \mathbf{x}_n)^T \quad (6.35)$$

这是Fisher得分的协方差矩阵，因此Fisher核对应于这些分数的一个漂白。更简单地，我们可以省略Fisher信息矩阵，使用非不变核

$$k(\mathbf{x}, \mathbf{x}') = \mathbf{g}(\boldsymbol{\theta}, \mathbf{x})^T \mathbf{g}(\boldsymbol{\theta}, \mathbf{x}') \quad (6.36)$$

Hofmann (2000) 给出了Fisher核在文档抽取上的一个应用。

核函数的最后的一个例子是sigmoid核，定义为

$$k(\mathbf{x}, \mathbf{x}') = \tanh(a \mathbf{x}^T \mathbf{x}' + b) \quad (6.37)$$

它的Gram矩阵通常不是半正定的。但是这种核在实际应用中也可以使用 (Vapnik, 1995)，可能是因为它赋予核展开（例如支持向量机）一个与神经网络模型的表面的相似性。正如我们将看到的那样，在基函数有无穷多的极限情况下，一个具有恰当先验的贝叶斯神经网络将会变为高斯过程，因此这就提供了神经网络与核方法之间的一个更深层的联系。

6.3 径向基函数网络

在第3章，我们讨论了基于固定基函数的线性组合的回归模型，但是我们没有详细讨论可以取哪种形式的基函数。一种广泛使用的基函数是径向基函数 (radial basis functions)。径向基函数中，每一个基函数只依赖于样本和中心 μ_j 之间的径向距离（通常是欧几里得距离），即 $\phi_j(\mathbf{x}) = h(\|\mathbf{x} - \mu_j\|)$ 。

历史上，径向基函数被用来进行精确的函数内插 (Powell, 1987)。给定一组输入向量 $\{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ 以及对应的目标值 $\{t_1, \dots, t_N\}$ ，目标是找到一个光滑的函数 $f(\mathbf{x})$ ，它能够精确地拟合每个目标值，即对于 $n = 1, \dots, N$ ，都有 $f(\mathbf{x}_n) = t_n$ 。可以这样做：将 $f(\mathbf{x})$ 表示为径向基函数的线性组合，每个径向基函数都以数据点为中心，即

$$f(\mathbf{x}) = \sum_{n=1}^N w_n h(\|\mathbf{x} - \mathbf{x}_n\|) \quad (6.38)$$

系数 $\{w_n\}$ 的值由最小平方方法求出。并且，由于具有与系数数量相同的限制条件，因此结果是一个能够精确拟合每个目标值的函数。但是，在模式识别应用中，目标值通常带有噪声，精确内插不是我们想要的，因为这对应于一个过拟合的解。

对径向基函数的展开来自正则化理论 (Poggio and Girosi, 1990; Bishop, 1995a)。对于一个使用微分算符定义的带有正则化项的平方和误差函数，最优解可以通过对算符的Green函数（类似于离散矩阵的特征向量）进行展开，每个数据点有一个基函数。如果微分算符是各向同性的，那么Green函数只依赖于与对应的数据点的径向距离。由于正则化项的存在，因此解不再精确地对训练数据进行内插。

径向基函数的另一个研究动机来源于输入变量（而不是目标变量）具有噪声时的内插问题 (Webb, 1994; Bishop, 1995a)。如果输入变量 \mathbf{x} 上的噪声由一个服从分布 $\nu(\xi)$ 的变量 ξ 描述，那么平方和误差函数就变成了

$$E = \frac{1}{2} \sum_{n=1}^N \int \{y(\mathbf{x}_n + \xi) - t_n\}^2 \nu(\xi) d\xi \quad (6.39)$$

使用变分法，我们可以关于函数 $y(\mathbf{x})$ 进行最优化，得到

$$y(\mathbf{x}) = \sum_{n=1}^N t_n h(\mathbf{x} - \mathbf{x}_n) \quad (6.40)$$

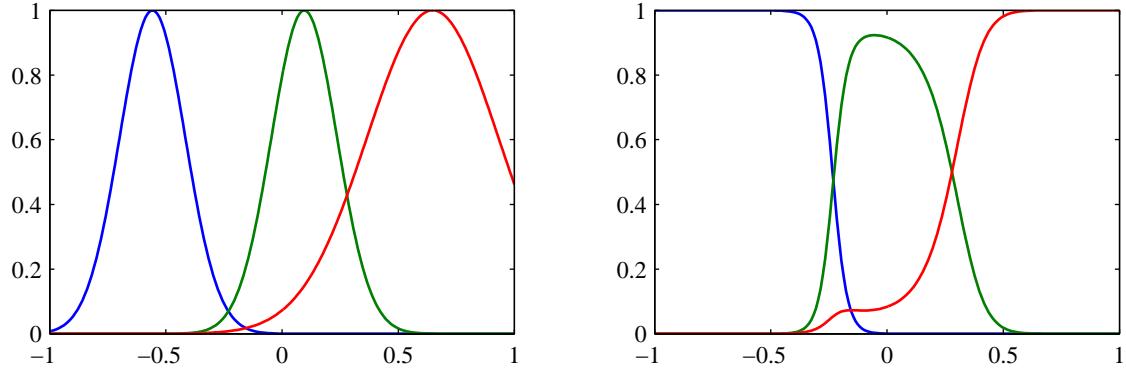


图 6.2: 左图给出了一组高斯基函数的图像，右图给出了对应的归一化的基函数的图像。

其中基函数为

$$h(\mathbf{x} - \mathbf{x}_n) = \frac{\nu(\mathbf{x} - \mathbf{x}_n)}{\sum_{n=1}^N \nu(\mathbf{x} - \mathbf{x}_n)} \quad (6.41)$$

我们看到这是一个以每个数据点为中心的基函数。这被称为Nadaraya-Watson模型。在6.3.1节，我们会从一个不同的角度再次推导出这个模型。如果噪声分布 $\nu(\xi)$ 是各向同性的，即它只是 $\|\xi\|$ 的一个函数，那么基函数就是径向的。

注意，基函数 (6.41) 是归一化的，即对于所有的 \mathbf{x} 值都有 $\sum_n h(\mathbf{x} - \mathbf{x}_n) = 1$ 。这种归一化的效果如图6.2所示。有时在实际应用中会用到归一化，因为它避免了输入空间中存在所有的基函数全部取较小值的区域，这种区域会导致在这些区域的预测值过小，或者完全由基参数控制。

另一个展开归一化径向基函数的情况是把核密度估计应用到回归问题中，正如我们将在6.3.1节讨论的那样。

由于每一个数据点都关联了一个基函数，因此当对于新的数据点进行预测时，对应的模型的计算开销会非常大。因此，一些新的模型被提出来 (Broomhead and Lowe, 1988; Moody and Darken, 1989; Poggio and Girosi, 1990)，这些模型仍然对径向基函数进行展开，但是基函数的数量 M 要小于数据点的数量 N 。通常，基函数的数量，以及它们的中心 μ_i ，都只是基于输入数据 $\{\mathbf{x}_n\}$ 自身来确定。然后基函数被固定下来，系数 $\{w_i\}$ 由最小平方方法通过解线性方程的方式确定，正如3.1.1节讨论的那样。

选择基函数中心的一种最简单的方法是使用数据点的一个随机选择的子集。一个更加系统化的方法被称为正交最小平方 (Chen et al., 1991)。这是一个顺序选择的过程，在每一个步骤中，被选择作为基函数的下一个数据点对应于能够最大程度减小平方和误差的数据点。展开系数的确定是算法的一部分。还可以使用聚类算法（例如K均值算法），这时得到的一组基函数中心不再与训练数据点重合。

6.3.1 Nadaraya-Watson模型

在3.3.3节，我们看到，对于新的输入 \mathbf{x} ，线性回归模型的预测的形式为训练数据集的目标值的线性组合，组合系数由“等价核” (3.62) 给出，其中等价核满足加和限制 (3.64)。

我们可以从核密度估计开始，以一个不同的角度研究核回归模型 (3.61)。假设我们有一个训练集 $\{\mathbf{x}_n, t_n\}$ ，我们使用Parzen密度估计来对联合分布 $p(\mathbf{x}, t)$ 进行建模，即

$$p(\mathbf{x}, t) = \frac{1}{N} \sum_{n=1}^N f(\mathbf{x} - \mathbf{x}_n, t - t_n) \quad (6.42)$$

其中 $f(\mathbf{x}, t)$ 是分量密度函数，每个数据点都有一个以数据点为中心的这种分量。我们现在要找

到回归函数 $y(\mathbf{x})$ 的表达式，对应于以输入变量为条件的目标变量的条件均值，它的表达式为

$$\begin{aligned} y(\mathbf{x}) &= \mathbb{E}[t | \mathbf{x}] = \int_{-\infty}^{\infty} tp(t | \mathbf{x}) dt \\ &= \frac{\int tp(\mathbf{x}, t) dt}{\int p(\mathbf{x}, t) dt} \\ &= \frac{\sum_n \int t f(\mathbf{x} - \mathbf{x}_n, t - t_n) dt}{\sum_m \int f(\mathbf{x} - \mathbf{x}_m, t - t_m) dt} \end{aligned} \quad (6.43)$$

简单起见，我们现在假设分量的密度函数的均值为零，即

$$\int_{-\infty}^{\infty} f(\mathbf{x}, t) t dt = 0 \quad (6.44)$$

对所有的 \mathbf{x} 都成立。使用一个简单的变量替换，我们有

$$\begin{aligned} y(\mathbf{x}) &= \frac{\sum_n g(\mathbf{x} - \mathbf{x}_n) t_n}{\sum_m g(\mathbf{x} - \mathbf{x}_m)} \\ &= \sum_n k(\mathbf{x}, \mathbf{x}_n) t_n \end{aligned} \quad (6.45)$$

其中 $n, m = 1, \dots, N$ ，且核函数 $k(\mathbf{x}, \mathbf{x}_n)$ 为

$$k(\mathbf{x}, \mathbf{x}_n) = \frac{g(\mathbf{x} - \mathbf{x}_n)}{\sum_m g(\mathbf{x} - \mathbf{x}_m)} \quad (6.46)$$

并且我们定义了

$$g(\mathbf{x}) = \int_{-\infty}^{\infty} f(\mathbf{x}, t) dt \quad (6.47)$$

公式 (6.45) 给出的结果被称为Nadaraya-Watson模型，或者称为核回归 (kernel regression) (Nadaraya, 1964; Watson, 1964)。对于一个局部核函数，它的性质为：给距离 \mathbf{x} 较近的数据点 \mathbf{x}_n 较高的权重。注意，核 (6.46) 满足加和限制

$$\sum_{n=1}^N k(\mathbf{x}, \mathbf{x}_n) = 1$$

事实上，这个模型不仅定义了条件期望，还定义了整个条件概率分布

$$p(t | \mathbf{x}) = \frac{p(t, \mathbf{x})}{\int p(t, \mathbf{x}) dt} = \frac{\sum_n f(\mathbf{x} - \mathbf{x}_n, t - t_n)}{\sum_m \int f(\mathbf{x} - \mathbf{x}_m, t - t_m) dt} \quad (6.48)$$

根据这个表达式，其他的期望也可以计算。

为了举例说明，我们考虑一元输入变量 x 的情形，其中 $f(x, t)$ 由变量 $z = (x, t)$ 上的一个零均值各向同性的高斯分布给出，方差为 σ^2 。对应的条件分布 (6.48) 由高斯混合模型给出。图6.3展示了对于正弦曲线人工生成数据集，这个条件分布的情况以及它的均值。

这个模型的一个明显的推广是允许形式更灵活的高斯分布作为其分量，例如让输入和目标值具有不同方差。更一般地，我们可以使用高斯混合模型对联合分布 $p(t, \mathbf{x})$ 建模，这个混合高斯模型使用第9章讨论的方法训练 (Ghahramani and Jordan, 1994)，然后找到对应的条件概率分布 $p(t | \mathbf{x})$ 。在后一种情况中，模型不再由训练数据点处的核函数表示，但是混合模型中分量的个数会小于训练数据点的个数，从而使得生成的模型对于测试数据点能够更快地计算。为了能够生成一个预测速度较快的模型，我们可以接受训练阶段的计算开销。

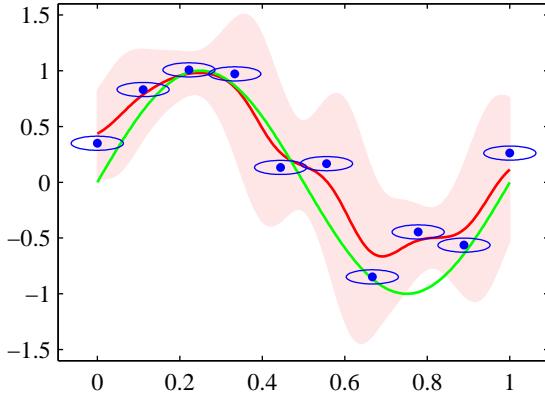


图 6.3: 使用各向同性的高斯核的Nadaraya-Watson核回归模型的说明。数据集为正弦数据集。原始的正弦函数由绿色曲线表示，数据点由蓝色的点表示，每个数据点是一个各向同性的高斯核的中心。得到的回归函数，由条件均值给出，用红线表示。同时给出的还有条件概率分布 $p(t | x)$ 的两个标准差的区域，用红色阴影表示。在每个数据点周围的蓝色椭圆给出了对应的核的一个标准差轮廓线。由于水平轴和垂直轴的标度不同，这些轮廓线似乎不是圆形的。

6.4 高斯过程

在6.1节，通过将对偶性的概念应用于回归的非概率模型，我们引出了核的概念。这里，我们把核的角色推广到概率判别式模型中，引出了高斯过程的框架。于是，我们会看到在贝叶斯方法中，核是如何自然地被引入的。

在第3章，我们考虑了线性回归模型，形式为 $y(\mathbf{x}, \mathbf{w}) = \mathbf{w}^T \phi(\mathbf{x})$ ，其中 \mathbf{w} 是一个参数向量， $\phi(\mathbf{x})$ 是一个与输入向量 \mathbf{x} 相关的固定非线性基函数向量。我们证明了， \mathbf{w} 上的先验分布会产生函数 $y(\mathbf{x}, \mathbf{w})$ 上的一个对应的先验分布。给定一个训练数据集，我们计算 \mathbf{w} 上的后验概率分布，从而就得到和回归函数的对应的后验概率分布。回归函数反过来（叠加上噪声）表示了对于新输入向量 \mathbf{x} 的一个预测分布 $p(t | \mathbf{x})$ 。

在高斯过程的观点中，我们抛弃参数模型，直接定义函数上的先验概率分布。乍一看来，在函数组成的不可数的无穷空间中对概率分布进行计算似乎很困难。但是，正如我们将看到的那样，对于一个有限的训练数据集，我们只需要考虑训练数据集和测试数据集的输入 \mathbf{x}_n 处的函数值即可，因此在实际应用中我们可以在有限的空间中进行计算。

等价于高斯过程的模型在许多不同领域被广泛研究。例如，在统计地质学中文献中，高斯过程回归被称为kriging (Cressie, 1993)。类似地，ARMA（自动回归移动平均）模型、Kalman滤波以及径向基函数网络都可以被看成高斯过程模型的形式。关于从机器学习的角度对高斯过程的回顾，可以参考MacKay (1998)、Williams (1999) 和MacKay (2003)。Rasmussen (1996) 给出了一个不同的方法来对各个高斯过程模型进行对比。有关高斯过程的最近的教科书，可以参考Rasmussen and Williams (2006)。

6.4.1 重新考虑线性回归问题

为了引出高斯过程的观点，让我们回到线性回归的例子中，通过对函数 $y(\mathbf{x}, \mathbf{w})$ 的计算，重新推导出预测分布。这会给出高斯过程的一个具体的例子。

考虑一个模型 M ，它被定义为由向量 $\phi(\mathbf{x})$ 的元素给出的 M 个固定基函数的线性组合，即

$$y(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x}) \quad (6.49)$$

其中 \mathbf{x} 是输入向量， \mathbf{w} 是 M 维权向量。现在，考虑 \mathbf{w} 上的一个先验概率分布，这个分布是一个各向同性的高斯分布，形式为

$$p(\mathbf{w}) = \mathcal{N}(\mathbf{w} | \mathbf{0}, \alpha^{-1} \mathbf{I}) \quad (6.50)$$

它由一个超参数 α 控制，这个超参数表示分布的精度（方差的倒数）。对于任意给定的 \mathbf{w} ，公式 (6.49) 定义了 \mathbf{x} 的一个特定的函数。于是，公式 (6.50) 定义的 \mathbf{w} 上的概率分布就产生了一个

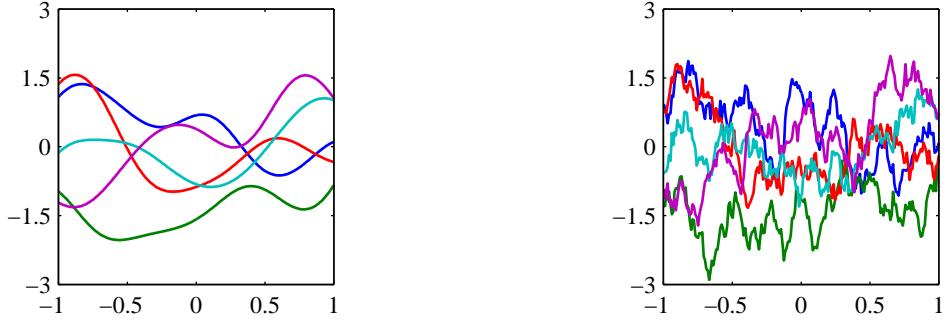


图 6.4: 左图为“高斯”核的高斯过程的样本，右图为指数核的高斯过程的样本。

函数 $y(\mathbf{x})$ 上的一个概率分布。在实际应用中，我们希望计算这个函数在某个具体的 \mathbf{x} 处的函数值，例如在训练数据点 $\mathbf{x}_1, \dots, \mathbf{x}_N$ 处的函数值。于是我们感兴趣的是函数值 $y(\mathbf{x}_1), \dots, y(\mathbf{x}_N)$ 的概率分布。我们把函数值的集合记作向量 \mathbf{y} ，它的元素为 $y_n = y(\mathbf{x}_n)$ ，其中 $n = 1, \dots, N$ 。根据公式 (6.49)，这个向量等于

$$\mathbf{y} = \Phi \mathbf{w} \quad (6.51)$$

其中 Φ 是设计矩阵，元素为 $\Phi_{nk} = \phi_k(\mathbf{x}_n)$ 。我们可以用下面的方式找到 \mathbf{y} 的概率分布。首先，我们注意到 \mathbf{y} 是由 \mathbf{w} 的元素给出的服从高斯分布的变量的线性组合，因此它本身是服从高斯分布。于是，我们只需要找到它的均值和方差。根据公式 (6.50)，均值和方差为

$$\mathbb{E}[\mathbf{y}] = \Phi \mathbb{E}[\mathbf{w}] = \mathbf{0} \quad (6.52)$$

$$\text{cov}[\mathbf{y}] = \mathbb{E}[\mathbf{y}\mathbf{y}^T] = \Phi \mathbb{E}[\mathbf{w}\mathbf{w}^T] \Phi^T = \frac{1}{\alpha} \Phi \Phi^T = \mathbf{K} \quad (6.53)$$

其中， \mathbf{K} 是Gram矩阵，元素为

$$K_{nm} = k(\mathbf{x}_n, \mathbf{x}_m) = \frac{1}{\alpha} \phi(\mathbf{x}_n)^T \phi(\mathbf{x}_m) \quad (6.54)$$

$k(\mathbf{x}, \mathbf{x}')$ 是核函数。

这个模型给我们提供了一个具体的例子。通常来说，高斯过程被定义为函数 $y(\mathbf{x})$ 上的一个概率分布，使得在任意点集 $\mathbf{x}_1, \dots, \mathbf{x}_N$ 处计算的 $y(\mathbf{x})$ 的值的集合联合起来服从高斯分布。在输入向量 \mathbf{x} 是二维的情况下，这也可以被称为高斯随机场 (Gaussian random field)。更一般地，可以用一种合理的方式为 $y(\mathbf{x}_1), \dots, y(\mathbf{x}_N)$ 赋予一个联合的概率分布，来确定一个随机过程 (stochastic process) $y(\mathbf{x})$ 。

高斯随机过程的一个关键点是 N 个变量 y_1, \dots, y_N 上的联合概率分布完全由二阶统计（即均值和协方差）确定。在大部分应用中，我们关于 $y(\mathbf{x})$ 的均值没有任何先验的知识，因此根据对称性，我们令其等于零。这等价于基函数的观点中，令权值 $p(\mathbf{w} | \alpha)$ 的先验概率分布的均值等于零。之后，高斯过程的确定通过给定两个 \mathbf{x} 处的函数值 $y(\mathbf{x})$ 的协方差来完成。这个协方差由核函数确定

$$\mathbb{E}[y(\mathbf{x}_n)y(\mathbf{x}_m)] = k(\mathbf{x}_n, \mathbf{x}_m) \quad (6.55)$$

对于由公式 (6.49) 定义的带有权值先验 (6.50) 的线性回归模型来说，核函数为 (6.54)。

我们也可以直接定义核函数，而不是间接地通过选择基函数。图6.4给出了对于两个不同的核函数，由高斯过程产生的函数的样本。第一个核函数是公式 (6.23) 定义的高斯核，第二个核函数是指数核，定义为

$$k(x, x') = \exp(-\theta|x - x'|) \quad (6.56)$$

它对应于Ornstein-Uhlenbeck过程。这个随机过程最开始由Uhlenbeck and Ornstein (1930) 提出，用来描述布朗运动。

6.4.2 用于回归的高斯过程

为了把高斯过程模型应用于回归问题，我们需要考虑观测目标值的噪声，形式为

$$t_n = y_n + \epsilon_n \quad (6.57)$$

其中 $y_n = y(\mathbf{x}_n)$, ϵ_n 是一个随机噪声变量，它的值对于每个观测 n 是独立的。这里，我们要考虑服从高斯分布的噪声过程，即

$$p(t_n | y_n) = \mathcal{N}(t_n | y_n, \beta^{-1}) \quad (6.58)$$

其中 β 是一个超参数，表示噪声的精度。由于噪声对于每个数据点是独立的，因此以 $\mathbf{y} = (y_1, \dots, y_N)^T$ 为条件，目标值 $\mathbf{t} = (t_1, \dots, t_N)^T$ 的联合概率分布是一个各向同性的高斯分布，形式为

$$p(\mathbf{t} | \mathbf{y}) = \mathcal{N}(\mathbf{t} | \mathbf{y}, \beta^{-1} \mathbf{I}_N) \quad (6.59)$$

其中 \mathbf{I}_N 表示一个 $N \times N$ 的单位矩阵。根据高斯过程的定义，边缘概率分布 $p(\mathbf{y})$ 是一个高斯分布，均值为零，协方差由 Gram 矩阵 \mathbf{K} 定义，即

$$p(\mathbf{y}) = \mathcal{N}(\mathbf{y} | \mathbf{0}, \mathbf{K}) \quad (6.60)$$

确定 \mathbf{K} 的核函数通常被选择成能够表示下面的性质：对于相似的点 \mathbf{x}_n 和 \mathbf{x}_m ，对应的值 $y(\mathbf{x}_n)$ 和 $y(\mathbf{x}_m)$ 的相关性要大于不相似的点。这里，相似性的概念取决于实际应用。

为了找到以输入值 $\mathbf{x}_1, \dots, \mathbf{x}_N$ 为条件的边缘概率分布 $p(\mathbf{t})$ ，我们需要对 \mathbf{y} 积分。可以通过使用 2.3.3 节的线性高斯模型的结果来完成。使用公式 (2.115)，我们看到 \mathbf{t} 的边缘概率分布为

$$p(\mathbf{t}) = \int p(\mathbf{t} | \mathbf{y}) p(\mathbf{y}) d\mathbf{y} = \mathcal{N}(\mathbf{t} | \mathbf{0}, \mathbf{C}) \quad (6.61)$$

其中协方差矩阵 \mathbf{C} 的元素为

$$C(\mathbf{x}_n, \mathbf{x}_m) = k(\mathbf{x}_n, \mathbf{x}_m) + \beta^{-1} \delta_{nm} \quad (6.62)$$

这个结果反映了下面的事实：两个随机的高斯分布（即与 $y(\mathbf{x})$ 相关的高斯分布和与 ϵ 相关的高斯分布）是独立的，因此它们的协方差可以简单地相加。

对于高斯过程回归，一个广泛使用的核函数的形式为指数项的二次型加上常数和线性项，即

$$k(\mathbf{x}_n, \mathbf{x}_m) = \theta_0 \exp \left\{ -\frac{\theta_1}{2} \|\mathbf{x}_n - \mathbf{x}_m\|^2 \right\} + \theta_2 + \theta_3 \mathbf{x}_n^T \mathbf{x}_m \quad (6.63)$$

注意，涉及到 θ_3 的项对应于一个参数模型，这个模型是输入变量的线性函数。图 6.5 给出了不同的参数 $\theta_0, \dots, \theta_3$ 的情况下，这个先验的图像。图 6.6 给出了一组从概率分布 (6.60) 中取样的样本点，以及由公式 (6.61) 定义的对应的值。

目前为止，我们已经使用高斯过程的观点来构建数据点的集合上的联合概率分布的模型。然而，我们在回归问题中的目标是在给定一组训练数据的情况下，对新的输入变量预测目标变量的值。让我们假设 $\mathbf{t}_N = (t_1, \dots, t_N)^T$ ，对应于输入值 $\mathbf{x}_1, \dots, \mathbf{x}_N$ ，组成观测训练集，并且我们的目标是对于新的输入向量 \mathbf{x}_{N+1} 预测目标变量 t_{N+1} 。这要求我们计算预测分布 $p(t_{N+1} | \mathbf{t}_N)$ 。注意，这个分布还要以变量 $\mathbf{x}_1, \dots, \mathbf{x}_N$ 和 \mathbf{x}_{N+1} 为条件。但是为了记号的简介，我们不会显式地写出这些条件变量。

为了找到条件分布 $p(t_{N+1} | \mathbf{t})$ ，我们首先写下联合概率分布 $p(\mathbf{t}_{N+1})$ ，其中 \mathbf{t}_{N+1} 表示向量 $(t_1, \dots, t_N, t_{N+1})^T$ 。然后，我们利用 2.3.1 节的结果来得到所求的条件概率分布，如图 6.7 所示。

根据公式 (6.61)， t_1, \dots, t_{N+1} 的联合概率分布为

$$p(\mathbf{t}_{N+1}) = \mathcal{N}(\mathbf{t}_{N+1} | \mathbf{0}, \mathbf{C}_{N+1}) \quad (6.64)$$

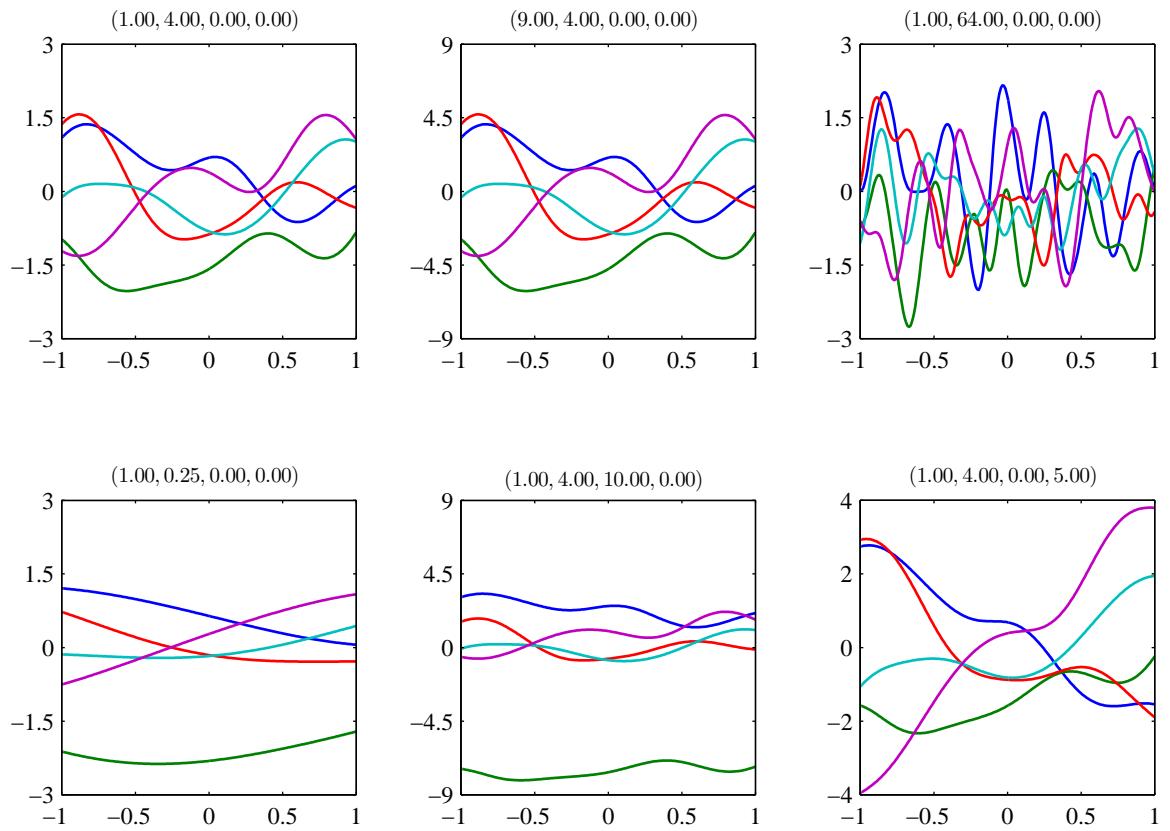


图 6.5: 由协方差函数 (6.63) 定义的高斯过程先验的样本。每张图上方的标题表示 $(\theta_0, \theta_1, \theta_2, \theta_3)$ 。

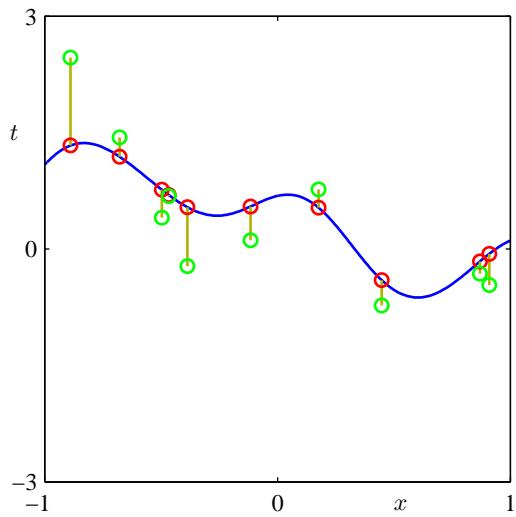


图 6.6: 高斯过程的数据点 $\{t_n\}$ 的取样的说明。蓝色曲线给出了函数上的高斯过程先验的一个样本函数，红点表示计算函数在一组输入值 $\{x_n\}$ 上计算得到的函数值 y_n 。对应的 $\{t_n\}$ 的值，用绿色表示，可以通过对每个 $\{y_n\}$ 添加独立噪声的方式得到。

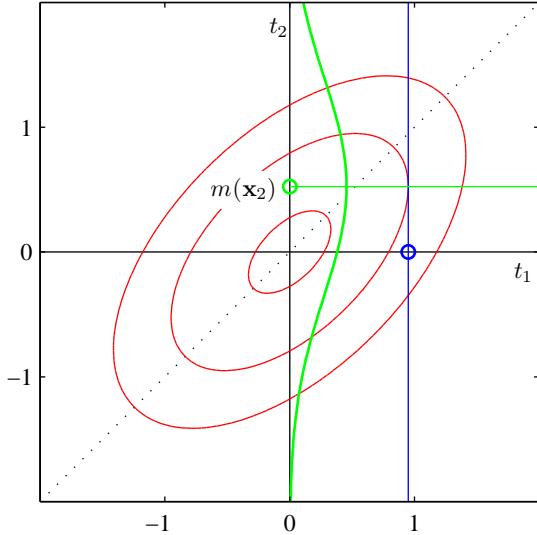


图 6.7: 高斯过程回归的原理说明, 其中只有一个训练点和一个测试点, 红色椭圆表示联合概率分布 $p(t_1, t_2)$ 的轮廓线。这里, t_1 是训练数据点。以 t_1 为条件 (蓝色直线), 我们得到了 $p(t_2 | t_1)$ 。绿色曲线表示它关于 t_2 的函数。

其中 \mathbf{C}_{N+1} 是一个 $(N+1) \times (N+1)$ 的协方差矩阵, 元素由公式 (6.62) 给出。由于这个联合分布是高斯分布, 因此我们可以使用 2.3.1 节的结果得到条件高斯分布。为了完成这一点, 我们将协方差矩阵分块如下

$$\mathbf{C}_{N+1} = \begin{pmatrix} \mathbf{C}_N & \mathbf{k} \\ \mathbf{k}^T & c \end{pmatrix} \quad (6.65)$$

其中 \mathbf{C}_N 是一个 $N \times N$ 的协方差矩阵, 元素由公式 (6.62) 给出, 其中 $n, m = 1, \dots, N$, 向量 \mathbf{k} 的元素为 $k(\mathbf{x}_n, \mathbf{x}_{N+1})$, 其中 $n = 1, \dots, N$, 标量 $c = k(\mathbf{x}_{N+1}, \mathbf{x}_{N+1}) + \beta^{-1}$ 。使用公式 (2.81) 和公式 (2.82), 我们看到条件概率分布 $p(t_{N+1} | \mathbf{t})$ 是一个高斯分布, 均值和协方差为

$$m(\mathbf{x}_{N+1}) = \mathbf{k}^T \mathbf{C}_N^{-1} \mathbf{t} \quad (6.66)$$

$$\sigma^2(\mathbf{x}_{N+1}) = c - \mathbf{k}^T \mathbf{C}_N^{-1} \mathbf{k} \quad (6.67)$$

这些是定义高斯过程回归的关键结果。由于向量 \mathbf{k} 是测试点输入值 \mathbf{x}_{N+1} 的函数, 因此我们看到预测分布是一个高斯分布, 它的均值和方差都依赖于 \mathbf{x}_{N+1} 。图 6.8 给出了高斯过程回归的一个例子。

核函数的唯一的限制是公式 (6.62) 给出的协方差矩阵一定是正定的。如果 λ_i 是 \mathbf{K} 的一个特征值, 那么 \mathbf{C} 的对应的特征值就是 $\lambda_i + \beta^{-1}$ 。因此可以证明对于任意点对 \mathbf{x}_n 和 \mathbf{x}_m , 核矩阵 $k(\mathbf{x}_n, \mathbf{x}_m)$ 一定是半正定的, 即 $\lambda_i \geq 0$, 因为任何等于零的特征值 λ_i 仍然会产生出 \mathbf{C} 的一个正的特征值, 因为 $\beta > 0$ 。这个限制条件与之前讨论的核函数的限制条件相同, 因此我们可以再次利用 6.2 节的所有方法构造恰当的核。

注意, 预测分布的均值 (6.66) 可以写成 \mathbf{x}_{N+1} 的函数, 形式为

$$(\mathbf{x}_{N+1}) = \sum_{n=1}^N a_n k(\mathbf{x}_n, \mathbf{x}_{N+1}) \quad (6.68)$$

其中 a_n 是 $\mathbf{C}_N^{-1} \mathbf{t}$ 的第 n 个元素。如果核函数 $k(\mathbf{x}_n, \mathbf{x}_m)$ 只依赖于距离 $\|\mathbf{x}_n - \mathbf{x}_m\|$, 那么我们就得到了径向基函数的一个展开。

公式 (6.66) 和公式 (6.67) 的结果定义了具有任意核函数 $k(\mathbf{x}, \mathbf{x}')$ 的高斯过程回归。在特殊情况下, 如果核函数 $k(\mathbf{x}, \mathbf{x}')$ 根据基函数的有限集定义, 那么我们就可以从高斯过程的观点开始, 推导出之前在 3.3.2 节得到的线性回归的结果。

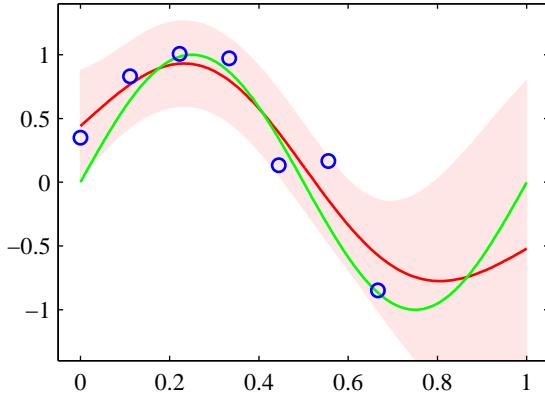


图 6.8: 高斯过程回归应用于图A.6的正弦数据集的说明，其中三个最右侧的点被略去。绿色曲线给出了正弦函数，其中数据点（蓝色点）通过对这个函数取样并且添加高斯噪声的方式得到。红线表示高斯过程预测分布的均值，阴影区域对应于两个标准差的位置。注意在数据点的右侧区域，不确定性是如何增加的。

因此对于这种模型，我们既可以通过参数空间的观点使用线性回归的结果得到预测分布，也可以通过函数空间的观点使用高斯过程的结果得到预测分布。

使用高斯过程的核心计算涉及到对 $N \times N$ 的矩阵求逆。标准的矩阵求逆方法需要 $O(N^3)$ 次计算。相反，在基函数模型中，我们要对一个 $M \times M$ 的矩阵 S_N 求逆，这需要 $O(M^3)$ 次计算。注意，对于两种观点来说，给定训练数据，矩阵求逆的计算必须进行一次。对于每个新的测试数据，两种方法都需要进行向量-矩阵的乘法，这在高斯过程中需要 $O(N^2)$ 次计算，在线性基函数模型中需要 $O(M^2)$ 次计算。如果基函数的数量 M 比数据点的数量 N 小，那么使用基函数框架计算会更高效。但是，高斯过程观点的一个优点是，我们可以处理那些只能通过无穷多的基函数表达的协方差函数。

但是，对于大的训练数据集，直接应用高斯过程方法就变得不可行了，因此一系列近似的方法被提出来。与精确的方法相比，这些近似的方法关于训练数据集的规模有着更好的时间复杂度 (Gibbs, 1997; Tresp, 2001; Smola and Bartlett, 2001; Williams and Seeger, 2001; Csató and Opper, 2002; Seeger et al., 2003)。

我们已经介绍了单一目标变量的高斯过程回归。扩展到多个目标变量的情形（被称为co-kriging）是很直接的 (Cressie, 1993)。也可以将高斯过程回归进行各种其他的扩展，用于对无监督学习的低维流形上的概率分布建模 (Bishop et al., 1998a) 以及解决随机微分方程 (Graepel, 2003)。

6.4.3 学习超参数

高斯过程模型的预测部分依赖于协方差函数的选择。在实际应用中，我们不固定协方差函数，而是更喜欢使用一组带有参数的函数，然后从数据中推断参数的值。这些参数控制了相关性的长度缩放以及噪声的精度等等，对应于标准参数模型的超参数。

学习超参数的方法基于计算似然函数 $p(\mathbf{t} | \boldsymbol{\theta})$ ，其中 $\boldsymbol{\theta}$ 表示高斯过程模型的超参数。最简单的方法是通过最大化似然函数的方法进行 $\boldsymbol{\theta}$ 的点估计。由于 $\boldsymbol{\theta}$ 表示回归问题的一组超参数，因此这可以看成类似于线性回归模型的第二类最大似然步骤。可以使用高效的基于梯度的最优化算法（例如共轭梯度法）来最大化对数似然函数 (Fletcher, 1987; Nocedal and Wright, 1999; Bishop and Nabney, 2008)。

使用多元高斯分布的标准形式，高斯过程模型的对数似然函数很容易计算。对数似然函数的形式为

$$\ln p(\mathbf{t} | \boldsymbol{\theta}) = -\frac{1}{2} \ln |\mathbf{C}_N| - \frac{1}{2} \mathbf{t}^T \mathbf{C}_N^{-1} \mathbf{t} - \frac{N}{2} \ln(2\pi) \quad (6.69)$$

对于非线性最优化，我们也需要对数似然函数关于参数向量 $\boldsymbol{\theta}$ 的梯度。我们假设计算 \mathbf{C}_N 的导数是比较简单的，它就是本章中讨论的协方差函数的情形。使用公式 (C.21) 给出的 \mathbf{C}_N^{-1} 的导数

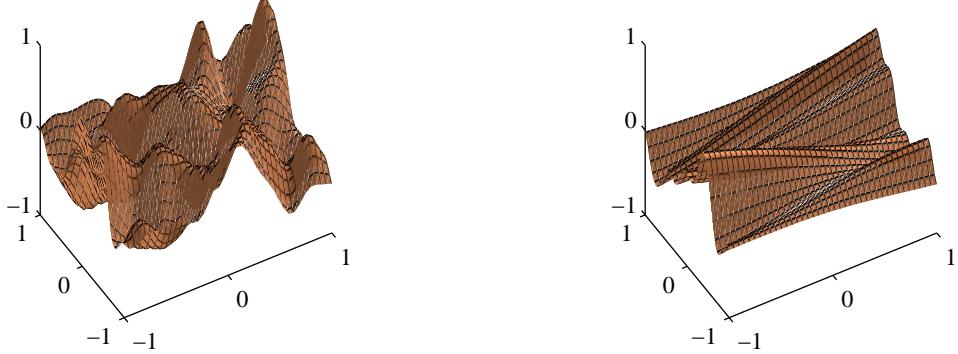


图 6.9: 来自高斯过程的ARD先验的样本，其中核函数由公式 (6.71) 给出。左图对应于 $\eta_1 = \eta_2 = 1$ ，右图对应于 $\eta_1 = 1, \eta_2 = 0.01$ 。

的结果，以及公式 (C.22) 给出的 $\ln |\mathbf{C}_N|$ 的结果，我们有

$$\frac{\partial}{\partial \theta_i} \ln p(\mathbf{t} | \boldsymbol{\theta}) = -\frac{1}{2} \text{Tr} \left(\mathbf{C}_N^{-1} \frac{\partial \mathbf{C}_N}{\partial \theta_i} \right) + \frac{1}{2} \mathbf{t}^T \mathbf{C}_N^{-1} \frac{\partial \mathbf{C}_N}{\partial \theta_i} \mathbf{C}_N^{-1} \mathbf{t} \quad (6.70)$$

由于 $\ln p(\mathbf{t} | \boldsymbol{\theta})$ 通常是一个非凸函数，因此它由多个极大值点。

引入一个 $\boldsymbol{\theta}$ 上的先验分布然后使用基于梯度的方法最大化对数后验是很容易的。在一个纯粹的贝叶斯方法中，我们需要计算 $\boldsymbol{\theta}$ 的边缘概率，乘以先验概率 $p(\boldsymbol{\theta})$ 和似然函数 $p(\mathbf{t} | \boldsymbol{\theta})$ 。然而，通常精确的积分或者求和是不可行的，我们必须进行近似。

高斯过程回归模型给出的预测分布的均值和方差是输入向量 \mathbf{x} 的函数。然而，我们已经假定由参数 β 控制的附加噪声对预测方差的贡献是常数。对于一些被称为异方差 (heteroscedastic) 的问题，噪声方差本身也依赖于 \mathbf{x} 。为了对这种问题进行建模，我们可以对高斯过程框架进行推广，引入第二个高斯过程来表示 β 对于输入 \mathbf{x} 的依赖性 (Goldberg et al., 1998)。由于 β 是一个方差，因此是非负的，所以我们使用高斯过程来对 $\ln \beta(\mathbf{x})$ 进行建模。

6.4.4 自动相关性确定

在前一节里，我们看到最大似然方法如何被用于确定高斯过程中的长度缩放参数的值。通过为每个输入变量整合一个单独的参数，这种方法可以很有用地推广 (Rasmussen and Williams, 2006)。正如我们将看到的那样，这样做的结果是，通过最大似然方法进行的参数最优化，能够将不同输入的相对重要性从数据中推断出来。这是高斯过程中的自动相关性确定 (automatic relevance determination) 或者ARD的一个例子。它起源于神经网络的框架 (MacKay, 1994; Neal, 1996)。这种方法倾向于选择合适的输入的机理在7.2.2节讨论。

考虑二维输入空间 $\mathbf{x} = (x_1, x_2)$ ，有一个下面形式的核函数

$$k(\mathbf{x}, \mathbf{x}') = \theta_0 \exp \left\{ -\frac{1}{2} \sum_{i=1}^2 \eta_i (x_i - x'_i)^2 \right\} \quad (6.71)$$

图6.9给出了两个不同的精度参数 η_i 的设定下， $y(\mathbf{x})$ 的先验概率分布。我们看到，随着特定的 η_i 的减小，函数逐渐对对应的输入变量 x_i 不敏感。通过使用最大似然法按照数据集调整这些参数，它可以检测到对于预测分布几乎没有影响的输入变量，因为对应的 η_i 会很小。这在实际应用中很有用，因为它使得这些输入可以被遗弃。图6.10使用一个具有三个输入 x_1, x_2 和 x_3 的简单人造数据集来说明ARD (Nabney, 2002)。目标变量 t 的生成方式为：从一个高斯分布中采样 100 个 x_1 ，计算函数 $\sin(2\pi x_1)$ ，然后加上添加上高斯噪声。 x_2 的值通过复制对应的 x_1 然后添加噪声的方式获得， x_3 的值从一个独立的高斯分布中采样。因此， x_1 很好地预测了 t ， x_2 对 t 的预测的噪声更大， x_3 与 t 之间只有偶然的相关性。一个带有ARD参数 η_1, η_2, η_3 的高斯过程的边缘似然

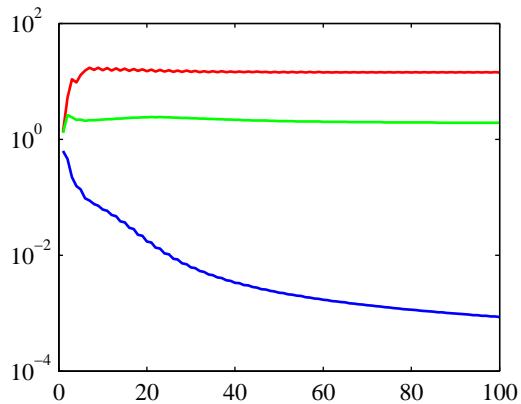


图 6.10: 高斯过程的自动相关性检测的例子。数据集是人工生成的数据集，由三个输入 x_1, x_2 和 x_3 。曲线表示对应的超参数的值与最优化边缘似然函数时的迭代次数的关系，红色表示 η_1 ，绿色表示 η_2 ，蓝色表示 η_3 。细节在正文中给出。注意垂直轴的对数标度。

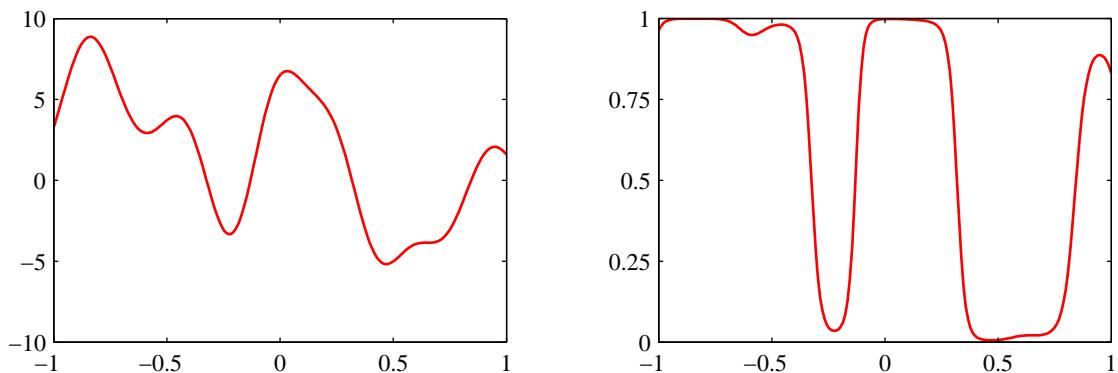


图 6.11: 左图给出了在函数 $a(\mathbf{x})$ 上定义了一个高斯过程先验的样本，右图给出了使用logistic sigmoid对这个样本进行变换得到的结果。

函数使用放缩的共轭梯度算法进行最优化。从图6.10中，我们看到 η_1 收敛到了一个相对较大的值， η_2 收敛到了一个小得多的值， η_3 变得非常小，表明 x_3 与预测 t 无关。

ARD框架很容易整合到指数-二次核 (6.63) 中，得到下面形式的核函数，它对于一大类将高斯过程应用于回归问题的实际应用都很有帮助。

$$k(\mathbf{x}_n, \mathbf{x}_m) = \theta_0 \exp \left\{ -\frac{1}{2} \sum_{i=1}^D \eta_i (x_{ni} - x_{mi})^2 \right\} + \theta_2 + \theta_3 \sum_{i=1}^D x_{ni} x_{mi} \quad (6.72)$$

其中 D 是输入空间的维度。

6.4.5 用于分类的高斯过程

在分类的概率方法中，我们的目标是在给定一组训练数据的情况下，对于一个新的输入向量，为目标变量的后验概率建模。这些概率一定位于区间 $(0, 1)$ 中，而一个高斯过程模型做出的预测位于整个实数轴上。然而，我们可以很容易地调整高斯过程，使其能够处理分类问题。方法为：使用一个恰当的非线性激活函数，将高斯过程的输出进行变换。

首先考虑一个二分类问题，它的目标变量为 $t \in \{0, 1\}$ 。如果我们定义函数 $a(\mathbf{x})$ 上的一个高斯过程，然后使用公式 (4.59) 给出的logistic sigmoid函数 $y = \sigma(a)$ 进行变换，那么我们就得到了函数 $y(\mathbf{x})$ 上的一个非高斯随机过程，其中 $y \in (0, 1)$ 。图6.11说明了一维输入空间的情况，其中目

标变量 t 上的概率分布是伯努利分布

$$p(t | a) = \sigma(a)^t (1 - \sigma(a))^{1-t} \quad (6.73)$$

与之前一样，我们把训练集的输入记作 $\mathbf{x}_1, \dots, \mathbf{x}_N$ ，对应的观测目标变量为 $\mathbf{t} = (t_1, \dots, t_N)^T$ 。我们还考虑一个单一的测试数据点 \mathbf{x}_{N+1} ，目标值为 t_{N+1} 。我们的目标是确定预测分布 $p(t_{N+1} | \mathbf{t})$ ，其中我们没有显式地写出它对于输入变量的条件依赖。为了完成这个目标，我们引入向量 a_{N+1} 上的高斯过程先验，它的分量为 $a(\mathbf{x}_1), \dots, a(\mathbf{x}_{N+1})$ 。这反过来定义了 \mathbf{t}_{N+1} 上的一个非高斯过程。通过以训练数据 \mathbf{t}_N 为条件，我们得到了求解的预测分布。 a_{N+1} 上的高斯过程先验的形式为

$$p(\mathbf{a}_{N+1}) = \mathcal{N}(\mathbf{a}_{N+1} | \mathbf{0}, \mathbf{C}_{N+1}) \quad (6.74)$$

与回归的情形不同，协方差矩阵不再包含噪声项，因为我们假设所有的训练数据点都被正确标记。然而，由于数值计算的原因，更方便的做法是引入一个由参数 ν 控制的类似噪声的项，它确保了协方差矩阵 \mathbf{C}_{N+1} 的元素为

$$C(\mathbf{x}_n, \mathbf{x}_m) = k(\mathbf{x}_n, \mathbf{x}_m) + \nu \delta_{nm} \quad (6.75)$$

其中 $k(\mathbf{x}_n, \mathbf{x}_m)$ 是6.2节讨论的一个任意的半正定核函数， ν 的值通常事先固定。我们会假定核函数 $k(\mathbf{x}, \mathbf{x}')$ 由参数向量 θ 控制，我们稍后会讨论如何从训练数据中学习到 θ 。

对于二分类问题，预测 $p(t_{N+1} = 1 | \mathbf{t}_N)$ 就足够了，因为 $p(t_{N+1} = 0 | \mathbf{t}_N)$ 的值等于 $1 - p(t_{N+1} = 1 | \mathbf{t}_N)$ 。求解的预测分布为

$$p(t_{N+1} = 1 | \mathbf{t}_N) = \int p(t_{N+1} = 1 | a_{N+1}) p(a_{N+1} | \mathbf{t}_N) da_{N+1} \quad (6.76)$$

其中 $p(t_{N+1} = 1 | a_{N+1}) = \sigma(a_{N+1})$ 。

这个积分无法求出解析解，因此可以使用采样的方法近似（Neal, 1997）。我们还可以使用另一种方法，这种方法基于一个解析的近似。在4.5.2节，我们推导了logistic sigmoid函数与高斯分布卷积的近似公式（4.153）。我们可以使用这个结果计算公式（6.76）中的积分，只要我们对后验概率分布 $p(a_{N+1} | \mathbf{t}_N)$ 进行高斯近似。通常对后验概率进行高斯近似的理由是，根据中心极限定理，随着数据点数量的增加，真实的后验概率将会趋向于一个高斯分布。在高斯过程的情形中，变量的数量随着数据点数量的增多而增多，因此这个结果不能直接应用。然而，如果我们考虑增加落在 \mathbf{x} 空间的固定区域中的数据点的数量，那么函数 $a(\mathbf{x})$ 中对应的不确定性就会减小，这就渐近地趋近于高斯分布（Williams and Barber, 1998）。

我们考虑三种不同的获得高斯近似的方法。一种方法基于变分推断（variational inference）（Gibbs and MacKay, 2000），并且使用了logistic sigmoid函数的局部变分界（10.144）。这使得sigmoid函数的乘积可以通过高斯的乘积近似，因此使得对 a_N 的积分可以解析地计算。这种方法也产生了似然函数 $p(\mathbf{t}_N | \theta)$ 的下界。通过使用softmax函数的高斯近似，高斯过程分类的变分法框架也可以扩展到多类（ $K > 2$ ）问题（Gibbs, 1997）。

第二种方法使用期望传播（expectation propagation）（Opper and Winther, 2000b; Minka, 2001b; Seeger, 2003）。正如我们将看到的那样，由于真实的后验概率是单峰的，期望传播方法可以给出很好的结果。

6.4.6 拉普拉斯近似

第三种高斯过程分类的方法基于拉普拉斯近似，我们现在将详细讨论。为了计算预测分布（6.76），我们寻找 a_{N+1} 的后验概率分布的高斯近似。使用贝叶斯定理，后验概率分布为

$$\begin{aligned} p(a_{N+1} | \mathbf{t}_N) &= \int p(a_{N+1}, \mathbf{a}_N | \mathbf{t}_N) d\mathbf{a}_N \\ &= \frac{1}{p(\mathbf{t}_N)} \int p(a_{N+1}, \mathbf{a}_N) p(\mathbf{t}_N | a_{N+1}, \mathbf{a}_N) d\mathbf{a}_N \\ &= \frac{1}{p(\mathbf{t}_N)} \int p(a_{N+1} | \mathbf{a}_N) p(\mathbf{a}_N) p(\mathbf{t}_N | \mathbf{a}_N) d\mathbf{a}_N \\ &= \int p(a_{N+1} | \mathbf{a}_N) p(\mathbf{a}_N | \mathbf{t}_N) d\mathbf{a}_N \end{aligned} \quad (6.77)$$

其中我们用到了 $p(\mathbf{t}_N \mid a_{N+1}, \mathbf{a}_N) = p(\mathbf{t}_N \mid \mathbf{a}_N)$ 。使用公式 (6.66) 和公式 (6.67) 给出的高斯过程回归的结果，我们可以得到条件概率分布 $p(a_{N+1} \mid \mathbf{a}_N)$ ，结果为

$$p(a_{N+1} \mid \mathbf{a}_N) = \mathcal{N}(a_{N+1} \mid \mathbf{k}^T \mathbf{C}_N^{-1} \mathbf{a}_N, c - \mathbf{k}^T \mathbf{C}_N^{-1} \mathbf{k}) \quad (6.78)$$

于是，通过找到后验概率分布 $p(\mathbf{a}_N \mid \mathbf{t}_N)$ 的拉普拉斯近似，然后使用两个高斯分布卷积的标准结果，我们就可以计算公式 (6.77) 中的积分。

先验概率 $p(\mathbf{a}_N)$ 由一个零均值高斯过程给出，协方差矩阵为 \mathbf{C}_N ，数据项（假设数据点之间具有独立性）为

$$p(\mathbf{t}_N \mid \mathbf{a}_N) = \prod_{n=1}^N \sigma(a_n)^{t_n} (1 - \sigma(a_n))^{1-t_n} = \prod_{n=1}^N e^{a_n t_n} \sigma(-a_n) \quad (6.79)$$

我们然后通过对 $p(\mathbf{a}_N \mid \mathbf{t}_N)$ 的对数进行泰勒展开，就可以得到拉普拉斯近似。忽略掉一些具有可加性的常数，这个概率的对数为

$$\begin{aligned} \Psi(\mathbf{a}_N) &= \ln p(\mathbf{a}_N) + \ln p(\mathbf{t}_N \mid \mathbf{a}_N) \\ &= -\frac{1}{2} \mathbf{a}_N^T \mathbf{C}_N^{-1} \mathbf{a}_N - \frac{N}{2} \ln(2\pi) - \frac{1}{2} \ln |\mathbf{C}_N| + \mathbf{t}_N^T \mathbf{a}_N \\ &\quad - \sum_{n=1}^N \ln(1 + e^{a_n}) \end{aligned} \quad (6.80)$$

首先我们需要找到后验概率分布的众数，这需要我们计算 $\Psi(\mathbf{a}_N)$ 的梯度。这个梯度为

$$\nabla \Psi(\mathbf{a}_N) = \mathbf{t}_N - \boldsymbol{\sigma}_N - \mathbf{C}_N^{-1} \mathbf{a}_N \quad (6.81)$$

其中 $\boldsymbol{\sigma}_N$ 是一个元素为 $\sigma(a_n)$ 的向量。寻找众数时，我们不能简单地令这个梯度等于零，因为 $\boldsymbol{\sigma}_N$ 与 \mathbf{a}_N 的关系是非线性的，因此我们需要使用基于Newton-Raphson方法的迭代的方法，它给出了一个迭代重加权最小平方 (IRLS) 算法。这需要求出 $\Psi(\mathbf{a}_N)$ 的二阶导数，而这个二阶导数也需要进行拉普拉斯近似，结果为

$$\nabla \nabla \Psi(\mathbf{a}_N) = -\mathbf{W}_N - \mathbf{C}_N^{-1} \quad (6.82)$$

其中 \mathbf{W}_N 是一个对角矩阵，元素为 $\sigma(a_n)(1 - \sigma(a_n))$ ，并且我们使用了公式 (4.88) 给出的logistic sigmoid函数的导数的结果。注意，这些对角矩阵元素位于区间 $(0, \frac{1}{4})$ ，因此 \mathbf{W}_N 是一个正定矩阵。由于 \mathbf{C}_N （以及它的逆矩阵）被构造为正定的，并且由于两个正定矩阵的和仍然是正定矩阵，因此我们看到Hessian矩阵 $\mathbf{A} = -\nabla \nabla \Psi(\mathbf{a}_N)$ 是正定的，因此后验概率分布 $p(\mathbf{a}_N \mid \mathbf{t}_N)$ 是对数凸函数，因此有一个唯一的众数，即全局最大值。然而，后验概率不是高斯分布，因为Hessian矩阵是 \mathbf{a}_N 的函数。

使用Newton-Raphson公式 (4.92)， \mathbf{a}_N 的迭代更新方程为

$$\mathbf{a}_N^{\text{新}} = \mathbf{C}_N (\mathbf{I} + \mathbf{W}_N \mathbf{C}_N)^{-1} \{ \mathbf{t}_N - \boldsymbol{\sigma}_N + \mathbf{W}_N \mathbf{a}_N \} \quad (6.83)$$

这个方程反复迭代，直到收敛于众数（记作 \mathbf{a}_N^* ）。在这个众数位置，梯度 $\nabla \Psi(\mathbf{a}_N)$ 为零，因此 \mathbf{a}_N^* 满足

$$\mathbf{a}_N^* = \mathbf{C}_N (\mathbf{t}_N - \boldsymbol{\sigma}_N) \quad (6.84)$$

一旦我们找到了后验概率的众数 \mathbf{a}_N^* ，我们就可以计算Hessian矩阵，结果为

$$\mathbf{H} = -\nabla \nabla \Psi(\mathbf{a}_N) = \mathbf{W}_N + \mathbf{C}_N^{-1} \quad (6.85)$$

其中 \mathbf{W}_N 的元素使用 \mathbf{a}_N^* 计算。这定义了我们对后验概率分布 $p(\mathbf{a}_N \mid \mathbf{t}_N)$ 的高斯近似，结果为

$$q(\mathbf{a}_N) = \mathcal{N}(\mathbf{a}_N \mid \mathbf{a}_N^*, \mathbf{H}^{-1}) \quad (6.86)$$

我们现在可以将这个结果与公式 (6.78) 结合，然后计算积分 (6.77)。因为这对应于线性高斯模型，我们可以使用一般的结果 (2.115) 得到

$$\mathbb{E}[a_{N+1} | \mathbf{t}_N] = \mathbf{k}^T(\mathbf{t}_N - \boldsymbol{\sigma}_N) \quad (6.87)$$

$$\text{var}[a_{N+1} | \mathbf{t}_N] = c - \mathbf{k}^T(\mathbf{W}_N^{-1} + \mathbf{C}_N)^{-1}\mathbf{k} \quad (6.88)$$

现在我们有一个 $p(a_{N+1} | \mathbf{t}_N)$ 的高斯分布，我们可以使用结果 (4.153) 近似积分 (6.76)。与 4.5 节的贝叶斯 logistic 回归模型相同，如果我们只对对应于 $p(t_{N+1} | \mathbf{t}_N) = 0.5$ 的决策边界感兴趣，那么我们只需考虑均值，可以忽略方差的效果。

我们还需要确定协方差函数的参数 θ 。一种方法是最大化似然函数 $p(\mathbf{t}_N | \theta)$ ，此时我们需要对数似然函数和它的梯度的表达式。如果必要的话，还可以加上正则化项，产生一个正则化的最大似然解。最大似然函数的定义为

$$p(\mathbf{t}_N | \theta) = \int p(\mathbf{t}_N | \mathbf{a}_N)p(\mathbf{a}_N | \theta) d\mathbf{a}_N \quad (6.89)$$

这个积分没有解析解，所以我们需要再次使用拉普拉斯近似。使用公式 (4.135) 的结果，我们得到了下面的对数似然函数的近似

$$\ln p(\mathbf{t}_N | \theta) = \Psi(\mathbf{a}_N^*) - \frac{1}{2} \ln |\mathbf{W}_N + \mathbf{C}_N^{-1}| + \frac{N}{2} \ln(2\pi) \quad (6.90)$$

其中 $\Psi(\mathbf{a}_N^*) = \ln p(\mathbf{a}_N^* | \theta) + \ln p(\mathbf{t}_N | \mathbf{a}_N^*)$ 。我们还需要计算 $\ln p(\mathbf{t}_N | \theta)$ 关于参数向量 θ 梯度。注意， θ 的改变会造成 \mathbf{a}_N^* 的改变，产生梯度中的附加项。因此，当我们对 (6.90) 关于 θ 求积分时，我们得到了两个项的集合，第一个集合产生于协方差矩阵 \mathbf{C}_N 对 θ 的依赖关系，第二个集合产生于 \mathbf{a}_N^* 对 θ 的依赖关系。

显式地依赖于 θ 的项可以使用公式 (6.80) 以及公式 (C.21) 和公式 (C.22) 给出的结果得到，结果为

$$\begin{aligned} \frac{\partial \ln p(\mathbf{t}_N | \theta)}{\partial \theta_j} &= \frac{1}{2} \mathbf{a}_N^{*T} \mathbf{C}_N^{-1} \frac{\partial \mathbf{C}_N}{\partial \theta_j} \mathbf{C}_N^{-1} \mathbf{a}_N^* \\ &\quad - \frac{1}{2} \text{Tr} \left[(\mathbf{I} + \mathbf{C}_N \mathbf{W}_N)^{-1} \mathbf{W}_N \frac{\partial \mathbf{C}_N}{\partial \theta_j} \right] \end{aligned} \quad (6.91)$$

为了计算由于 \mathbf{a}_N^* 对 θ 的依赖产生的项，我们注意到我们已经构造了拉普拉斯近似，从而在 $\mathbf{a}_N = \mathbf{a}_N^*$ 处， $\Psi(\mathbf{a}_N)$ 的均值为零，从而 $\Psi(\mathbf{a}_N^*)$ 对于梯度没有贡献。剩下的有贡献的项关于 θ 的分量 θ_j 的导数为

$$\begin{aligned} &- \frac{1}{2} \sum_{n=1}^N \frac{\partial \ln |\mathbf{W}_N + \mathbf{C}_N^{-1}|}{\partial a_n^*} \frac{\partial a_n^*}{\partial \theta_j} \\ &= - \frac{1}{2} \sum_{n=1}^N [(\mathbf{I} + \mathbf{C}_N \mathbf{W}_N)^{-1} \mathbf{C}_N]_{nn} \sigma_n^*(1 - \sigma_n^*)(1 - 2\sigma_n^*) \frac{\partial a_n^*}{\partial \theta_j} \end{aligned} \quad (6.92)$$

其中 $\sigma_n^* = \sigma(a_n^*)$ ，并且我们又一次使用了公式 (C.22) 给出的结果以及 \mathbf{W}_N 的定义。我们可以将公式 (6.84) 给出的关系关于 θ_j 求积分，得到 a_N^* 关于 θ_j 的导数，即

$$\frac{\partial a_n^*}{\partial \theta_j} = \frac{\partial \mathbf{C}_N}{\partial \theta_j}(\mathbf{t}_N - \boldsymbol{\sigma}_N) - \mathbf{C}_N \mathbf{W}_N \frac{\partial a_n^*}{\partial \theta_j} \quad (6.93)$$

整理，可得

$$\frac{\partial a_n^*}{\partial \theta_j} = (\mathbf{I} + \mathbf{W}_N \mathbf{C}_N)^{-1} \frac{\partial \mathbf{C}_N}{\partial \theta_j} (\mathbf{t}_N - \boldsymbol{\sigma}_N) \quad (6.94)$$

结合公式 (6.91)、(6.92) 和 (6.94)，我们可以计算对数似然函数的梯度，然后使用标准非线性优化算法来确定 θ 的值。

我们可以使用人工生成的两类数据来说明拉普拉斯近似对于高斯过程的应用，如图 6.12 所示。很容易将拉普拉斯近似推广到涉及 $K > 2$ 个类别的使用 softmax 激活函数的高斯过程 (Williams and Barber, 1998)。

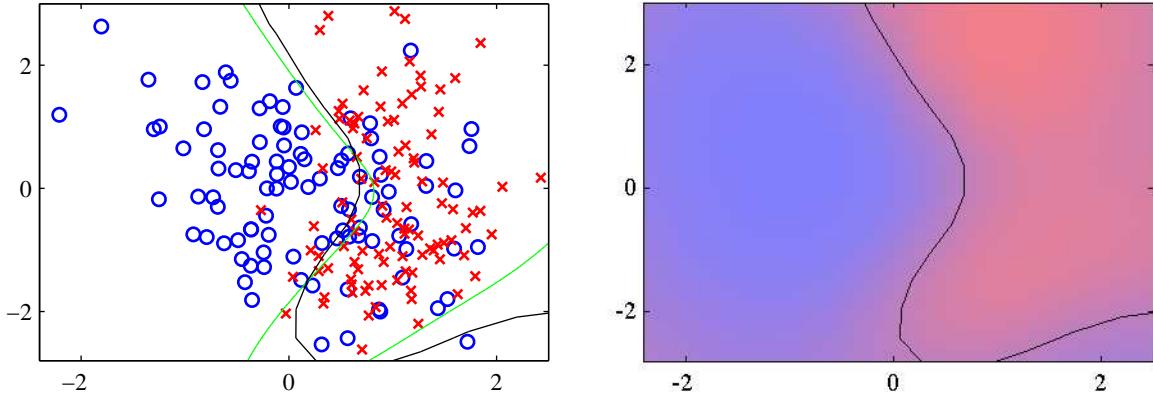


图 6.12: 使用高斯过程进行分类的说明。左图给出了数据点，以及来自真实概率分布的最优决策边界（绿色），还有来自高斯过程分类器的决策边界（黑色）。右图给出了蓝色类别和红色类别的预测后验概率分布，以及高斯过程决策边界。

6.4.7 与神经网络的联系

我们已经看到，神经网络可以表示的函数的范围由隐含单元的数量 M 控制，并且对于足够大的 M ，一个两层神经网络可以以任意精度近似任意给定的函数。在最大似然的框架中，隐含单元的数量需要有一定的限制（根据训练集的规模确定限制的程度），来避免过拟合现象。然而，从贝叶斯的角度看，根据训练集的规模限制参数的数量几乎毫无意义。

在贝叶斯神经网络中，参数向量 \mathbf{w} 上的先验分布以及网络函数 $f(\mathbf{x}, \mathbf{w})$ 产生了函数 $y(\mathbf{x})$ 上的先验概率分布，其中 y 是网络输出向量。Neal (1996) 已经证明，在极限 $M \rightarrow \infty$ 的情况下，对于 \mathbf{w} 的一大类先验分布，神经网络产生的函数的分布将会趋于高斯过程。然而，应该注意，在这种极限情况下，神经网络的输出变量会变为相互独立。神经网络的优势之一是输出之间共享隐含单元，因此它们可以互相“借统计优势”，即与每个隐含结点关联的权值被所有的输出变量影响，而不是只被它们中的某一个影响。这个性质在极限状态下的高斯过程中丢失了。

我们已经看到，高斯过程由它的协方差（核）函数确定。Williams (1998) 给出了在两种具体的隐含单元激活函数（probit 和 高斯）下，协方差的显式形式。这些核函数 $k(\mathbf{x}, \mathbf{x}')$ 是非静止的，即它们不能够表示为差 $\mathbf{x} - \mathbf{x}'$ 的函数，这是因为以零为中心的高斯权值先验破坏了权空间的平移不变性。

通过直接对协方差函数计算，我们隐式地在权值的分布上进行了积分。如果权值先验由超参数控制，那么它们的值会确定函数的分布的长度标度，这可以通过研究图5.11给出的有限数量单元情形的例子进行理解。注意我们不能解析地对超参数进行积分，而是必须借助6.4节讨论的技术。

6.5 练习

(6.1) (***) 考虑6.1节给出的最小平方线性回归问题的对偶形式。证明，向量 \mathbf{a} 的元素 a_n 的解可以表示为向量 $\phi(\mathbf{x}_n)$ 的元素的线性组合。将这些系数记作向量 \mathbf{w} ，证明对偶形式的对偶形式是用参数向量 \mathbf{w} 表示的原始表示。

(6.2) (**) 本练习中，我们研究感知器学习算法的对偶形式。使用感知器学习规则 (??)，证明学习的权向量 \mathbf{w} 可以表示为向量 $t_n \phi(\mathbf{x}_n)$ 的线性组合，其中 $t_n \in \{-1, 1\}$ 。将这个线性组合的系数记作 α_n ，推导感知器学习算法的公式，以及感知器的预测分布，用 α_n 表示。证明，特征向量 $\phi(\mathbf{x})$ 值出现在核函数 $k(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^T \phi(\mathbf{x}')$ 中。

(6.3) (*) 最近邻分类器 (2.5.2节) 将新的输入向量 \mathbf{x} 分配到训练集里距离最近的输入向量 \mathbf{x}_n 的类别，其中在最简单的情形中，距离被定义为欧几里得距离 $\|\mathbf{x} - \mathbf{x}_n\|^2$ 。通过将这个规则表示为标量积的形式，然后使用核替换，推导出对于一般的非线性核的最近邻分类器的公式。

(6.4) (*) 在附录C中，我们给出了一个具有正的元素但是负的特征值从而非正定的矩阵的例子。找到一个相反的例子，即一个 2×2 的矩阵，具有正的特征值，但是至少有一个元素为负。

(6.5) (*) 验证构造合法核的结果 (6.13) 和 (6.14)。

(6.6) (*) 验证构造合法核的结果 (6.15) 和 (6.16)。

(6.7) (*) 验证构造合法核的结果 (6.17) 和 (6.18)。

(6.8) (*) 验证构造合法核的结果 (6.19) 和 (6.20)。

(6.9) (*) 验证构造合法核的结果 (6.21) 和 (6.22)。

(6.10) (*) 证明, 学习函数 $f(\mathbf{x})$ 的一个比较好的核的选择是 $k(\mathbf{x}, \mathbf{x}') = f(\mathbf{x})f(\mathbf{x}')$, 证明方法为: 证明一个基于这个核的线性学习机器总会找到一个正比于 $f(\mathbf{x})$ 的解。

(6.11) (*) 通过使用展开式 (6.25), 然后将中间的因子展开为幂级数, 证明高斯核 (6.23) 可以表示为无限维特征空间中的内积。

(6.12) (**) 考虑给定一个固定集合 D 的情况下, 所有可能子集 A 组成的空间。证明, 核函数 (6.27) 对应于由映射 $\phi(A)$ 定义的维度为 $2^{|D|}$ 的特征空间中的内积, 其中, A 是 D 的一个子集, 元素 $\phi_U(A)$ 的下标为子集 U , 定义为

$$\phi_U(A) = \begin{cases} 1, & \text{如果 } U \subseteq A \\ 0, & \text{其他情况} \end{cases} \quad (6.95)$$

这里 $U \subseteq A$ 表示 U 是 A 的一个子集, 或者等于 A 。

(6.13) (*) 证明, 对于公式 (6.33) 给出的 Fisher 核, 如果我们对参数向量进行一个非线性变换 $\theta \rightarrow \psi(\theta)$, 那么这个核保持不变, 其中函数 $\psi(\cdot)$ 是可逆的、可微的。

(6.14) (*) 对于高斯分布 $p(\mathbf{x} | \boldsymbol{\mu}) = \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}, \mathbf{S})$, 其中均值为 $\boldsymbol{\mu}$, 协方差固定为 \mathbf{S} , 写出公式 (6.33) 给出的 Fisher 核的形式。

(6.15) (*) 通过考察一个 2×2 的 Gram 矩阵的行列式, 证明正定核函数 $k(x, x')$ 满足 Cauchy-Schwartz 不等式

$$k(x_1, x_2)^2 \leq k(x_1, x_1)k(x_2, x_2) \quad (6.96)$$

(6.16) (**) 考虑一个参数化模型, 它由参数向量 \mathbf{w} 、输入值 x_1, \dots, x_N 和一个非线性特征映射 $\phi(\mathbf{x})$ 控制。假设误差函数对 \mathbf{w} 的函数依赖关系的形式为

$$J(\mathbf{w}) = f(\mathbf{w}^T \phi(\mathbf{x}_1), \dots, \mathbf{w}^T \phi(\mathbf{x}_N)) + g(\mathbf{w}^T \mathbf{w}) \quad (6.97)$$

其中 $g(\cdot)$ 是一个单调递增函数。通过将 \mathbf{w} 写成

$$\mathbf{w} = \sum_{n=1}^N \alpha_n \phi(\mathbf{x}_n) + \mathbf{w}_\perp \quad (6.98)$$

其中 $\mathbf{w}_\perp^T \phi(\mathbf{x}_n) = 0$ 对于所有 n 都成立, 证明最小化 $J(\mathbf{w})$ 的 \mathbf{w} 的值的形式为基函数 $\phi(\mathbf{x}_n)$ 的线性组合, 其中 $n = 1, \dots, N$ 。

(6.17) (**) 考虑带有噪声的输入数据的平方和误差函数 (6.39), 其中 $\nu(\xi)$ 是噪声的分布。使用变分法, 关于函数 $y(\mathbf{x})$ 最小化这个误差函数, 从而证明最优解可以通过形如 (6.40) 的展开式给出, 其中基函数由公式 (6.41) 给出。

(6.18) (*) 考虑一个 Nadaraya-Watson 模型, 带有一个输入变量 x 和一个目标变量 t , 模型具有高斯分量, 分量的协方差是各向同性的, 从而协方差矩阵为 $\sigma^2 \mathbf{I}$, 其中 \mathbf{I} 是单位矩阵。使用核函数 $k(x, x_n)$, 写出条件概率密度 $p(t | x)$ 、条件均值 $\mathbb{E}[t | x]$ 和方差 $\text{var}[t | x]$ 的表达式。

(6.19) (**) 通过考察输入变量和目标变量被噪声污染过的回归问题, 我们可以得到核回归的另一个观点。假设每个目标值 t_n 与之前一样, 通过计算函数 $y(z_n)$ 在点 z_n 处的函数值然后添加噪声的方式得到。然而, z_n 的值不是直接观测到的, 而是一个被噪声污染的版本 $\mathbf{x}_n = z_n + \xi_n$, 其中, 随机变量 ξ 由某个概率分布 $g(\xi)$ 控制。考虑一组观测值 $\{\mathbf{x}_n, t_n\}$, 其中 $n = 1, \dots, N$, 以及对应的平方和误差函数, 通过对输入噪声取平均的方式定义, 即

$$E = \frac{1}{2} \sum_{n=1}^N \int \{y(\mathbf{x}_n - \xi_n) - t_n\}^2 g(\xi_n) d\xi_n \quad (6.99)$$

使用变分法 (附录 D), 关于函数 $y(z)$ 最小化 E , 证明 $y(\mathbf{x})$ 的最优解由 Nadaraya-Watson 核回归的解给出, 形式为 (6.45), 核函数为 (6.46)。

(6.20) (***) 验证结果 (6.66) 和 (6.67) 。

(6.21) (**) 考虑一个高斯过程回归模型，其中核函数根据一组固定的非线性基函数集合定义。证明，预测分布与3.3.2节讨论贝叶斯线性回归模型时得到的结果 (3.58) 完全相同。为了证明这一点，注意两个模型的预测分布都是高斯分布，因此只需证明条件均值和方差相同即可。对于均值，使用矩阵恒等式 (C.6)，对于方差，使用矩阵恒等式 (C.7) 。

(6.22) (**) 考虑一个回归问题，具有 N 个训练集输入向量 $\mathbf{x}_1, \dots, \mathbf{x}_N$ ，以及 L 个测试集输入向量 $\mathbf{x}_{N+1}, \dots, \mathbf{x}_{N+L}$ ，并且假设我们在函数 $t(\mathbf{x})$ 上定义了一个高斯过程先验。给定 $t(\mathbf{x}_1), \dots, t(\mathbf{x}_N)$ ，推导 $t(\mathbf{x}_{N+1}), \dots, t(\mathbf{x}_{N+L})$ 上的联合预测分布的表达式。对于一个测试观测 t_j ，其中 $N+1 \leq j \leq N+L$ ，证明这个概率分布的边缘概率分布由一般的高斯过程回归的结果 (6.66) 和 (6.67) 给出。

(6.23) (**) 考虑一个高斯过程回归模型，其中目标变量 t 的维度为 D 。给定输入变量 $\mathbf{x}_1, \dots, \mathbf{x}_N$ 以及对应的目标观测 $\mathbf{t}_1, \dots, \mathbf{t}_N$ 组成的训练集，对于一个测试输入向量 \mathbf{x}_{N+1} ，写出 \mathbf{t}_{N+1} 的条件概率分布。

(6.24) (*) 证明，元素满足 $0 < W_{ii} < 1$ 的对角矩阵 \mathbf{W} 是正定的。证明，两个正定矩阵的和本身为正定的。

(6.25) (*) 使用Newton-Raphson公式 (4.62) 公式，推导寻找高斯过程分类模型后验概率分布的峰值 a_N^* 的迭代更新公式 (6.83) 。

(6.26) (*) 使用公式 (2.115) 的结果，推导高斯过程分类模型后验概率分布的均值和方差的表达式 (6.87) 和 (6.88) 。

(6.27) (****) 推导高斯过程分类的拉普拉斯近似框架的对数似然函数的结果 (6.90) 。类似地，推导对数似然函数梯度中的项的结果 (6.91) 、 (6.92) 和 (6.94) 。

7 稀疏核机

在前一章中，我们研究了许多基于非线性核的学习算法。这种算法的一个最大的局限性是核函数 $k(\mathbf{x}_n, \mathbf{x}_m)$ 必须对所有可能的训练点对 \mathbf{x}_n 和 \mathbf{x}_m 进行求值，这在训练阶段的计算上是不可行的，并且会使得对新的数据点进行预测时也会花费过多的时间。本章中，我们会看到具有稀疏（sparse）解的基于核的算法，从而对新数据的预测只依赖于在训练数据点的一个子集上计算的核函数。

首先，我们详细讨论支持向量机（support vector machine）（SVM），它在一些年之前变得逐渐流行，可以用来解决分类问题、回归问题以及异常点检测问题。支持向量机的一个重要性质是模型参数的确定对应于一个凸最优化问题，因此许多局部解也是全局最优解。由于对支持向量机的讨论需要频繁用到拉格朗日乘数法，因此我们建议读者复习附录E中提到的关键的概念。额外的关于支持向量机的介绍，可以参考Vapnik (1995)、Burges (1998)、Cristianini and Shawe-Taylor (2000)、Müller et al. (2001)、Schölkopf and Smola (2002) 和Herbrich (2002)。

SVM是一个决策机器，因此不提供后验概率。我们已经在1.5.4节讨论过了确定概率的好处。另一种稀疏核方法，被称为相关向量机（relevance vector machine）（RVM），基于贝叶斯方法，提供了后验概率的输出，并且通常能产生比SVM更稀疏的解。

7.1 最大边缘分类器

为了开始我们关于支持向量机的讨论，我们回到使用线性模型的二分类问题。线性模型的形式为

$$y(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x}) + b \quad (7.1)$$

其中 $\phi(\mathbf{x})$ 表示一个固定的特征空间变换，并且我们显式地写出了偏置参数 b 。注意，我们会简要介绍使用核函数表达的对偶形式，这避免了显式地在特征空间中进行计算。训练数据集由 N 个输入向量 $\mathbf{x}_1, \dots, \mathbf{x}_N$ 组成，对应的目标值为 t_1, \dots, t_N ，其中 $t_n \in \{-1, 1\}$ ，新的数据点 \mathbf{x} 根据 $y(\mathbf{x})$ 的符号进行分类。

现阶段，我们假设训练数据集在特征空间中是线性可分的，即根据定义，存在至少一个参数 \mathbf{w} 和 b 的选择方式，使得对于 $t_n = +1$ 的点，函数 (7.1) 都满足 $y(\mathbf{x}_n) > 0$ ，对于 $t_n = -1$ 的点，都有 $y(\mathbf{x}_n) < 0$ ，从而对于所有训练数据点，都有 $t_n y(\mathbf{x}_n) > 0$ 。

当然，存在许多能够把类别精确分开的解。在4.1.7节，我们介绍了感知器算法，它能够保证在有限步骤之内找到一个解。然而，它找到的这个解依赖于 \mathbf{w} 和 b 的（任意的）初始值选择，还依赖于数据点出现的顺序。如果有多个能够精确分类训练数据点的解，那么我们应该尝试寻找泛化错误最小的那个解。支持向量机解决这个问题的方法是：引入边缘（margin）的概念，这个概念被定义为决策边界与任意样本之间的最小距离，如图7.1所示。

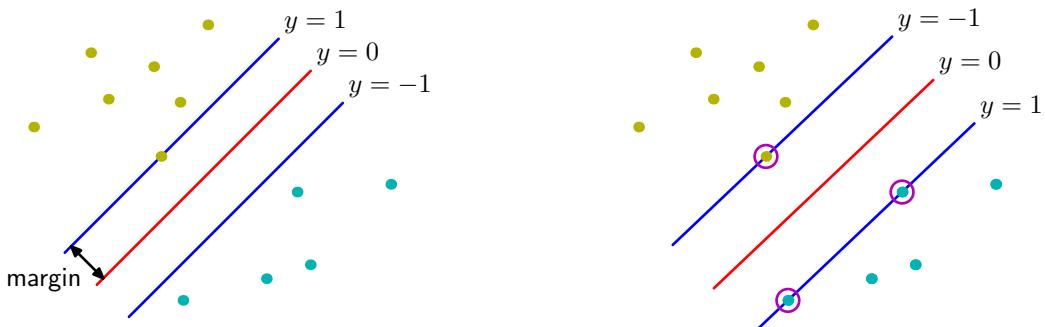


图 7.1: 边缘被定义为决策边界与最近的数据点之间的垂直距离，如左图所示。最大化边缘会生成对决策边界的一个特定的选择，如右图所示。这个决策边界的位置由数据点的一个子集确定，被称为支持向量，用圆圈表示。

在支持向量机中，决策边界被选为使边缘最大化的那个决策边界。采用最大边缘解的动机可以通过计算学习理论（computational learning theory）或者统计学习理论（statistical learning theory）进行理解。然而，Tong and Koller (2000) 给出了使用最大边缘解的一个简单的原因。他们考察了一个基于生成式方法和判别式方法组成的金字塔的分类框架，并且首先使用带有共同参数 σ^2 的高斯核的Parzen密度估计对每个类别的输入向量 \mathbf{x} 的分布进行建模。伴随着类别先验，这个分布定义了一个最优的分类错误率决策边界。然而，他们没有使用这个最优的决策边界，而是通过最小化学习到的模型的错误率来寻找最优的超平面。在极限 $\sigma^2 \rightarrow 0$ 的情况下，可以证明最优超平面是有着最大边缘的超平面。这个结果背后的直观含义是，随着 σ^2 的减小，距离超平面较近的点对超平面的控制能力逐渐大于距离较远的点。在极限情况下，超平面会变得与非支持向量的数据点无关。

我们会在图10.13中看到，对于一个简单的线性可分数据集，在贝叶斯方法中，关于参数的先验概率分布进行积分或求和，可以产生一个决策边界，这个决策边界位于分开数据点的区域中间。最大边缘解有着类似的行为。

回忆一下，根据图4.1，点 \mathbf{x} 距离由 $y(\mathbf{x}) = 0$ 定义的超平面的垂直距离为 $\frac{|y(\mathbf{x})|}{\|\mathbf{w}\|}$ ，其中 $y(\mathbf{x})$ 的函数形式由公式 (7.1) 给出。此外，我们感兴趣的是那些能够正确分类所有数据点的解，即对于所有的 n 都有 $t_n y(\mathbf{x}_n) > 0$ ，因此点 \mathbf{x}_n 距离决策面的距离为

$$\frac{t_n y(\mathbf{x}_n)}{\|\mathbf{w}\|} = \frac{t_n (\mathbf{w}^T \phi(\mathbf{x}_n) + b)}{\|\mathbf{w}\|} \quad (7.2)$$

边缘由数据集里垂直距离最近的点 \mathbf{x}_n 给出，我们希望最优化参数 \mathbf{w} 和 b ，使得这个距离能够最大化。因此，最大边缘解可以通过下式得到。

$$\arg \max_{\mathbf{w}, b} \left\{ \frac{1}{\|\mathbf{w}\|} \min_n [t_n (\mathbf{w}^T \phi(\mathbf{x}_n) + b)] \right\} \quad (7.3)$$

其中我们将因子 $\frac{1}{\|\mathbf{w}\|}$ 提到了对 n 的最优化之外，因为 \mathbf{w} 与 n 无关。直接求解这个最优化问题相当复杂，因此我们要把它转化为一个更容易求解的等价问题。为了完成这件事，我们注意到如果我们进行重新标度 $\mathbf{w} \rightarrow \kappa \mathbf{w}$ 以及 $b \rightarrow \kappa b$ ，那么任意点 \mathbf{x}_n 距离决策面的距离 $\frac{t_n y(\mathbf{x}_n)}{\|\mathbf{w}\|}$ 不会发生改变。我们可以使用这个性质，对于距离决策面最近的点，令

$$t_n (\mathbf{w}^T \phi(\mathbf{x}_n) + b) = 1 \quad (7.4)$$

在这种情况下，所有的数据点会满足限制

$$t (\mathbf{w}^T \phi(\mathbf{x}_n) + b) \geq 1, \quad n = 1, \dots, N \quad (7.5)$$

这被称为决策超平面的标准表示。对于使上式取得等号的数据点，我们说限制被激活（active），对于其他的数据点，我们说限制未激活（inactive）。根据定义，总会存在至少一个激活限制，因为总会有一个距离最近的点，并且一旦边缘被最大化，会有至少两个激活的限制。这样，最优化问题就简化为了最大化 $\|\mathbf{w}\|^{-1}$ ，这等价于最小化 $\|\mathbf{w}\|^2$ ，因此我们要在限制条件 (7.5) 下，求解最优化问题

$$\arg \min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2 \quad (7.6)$$

公式 (7.6) 的因子 $\frac{1}{2}$ 的引入是为了后续计算方便。这是二次规划（quadratic programming）问题的一个例子，其中我们试图在一组线性不等式的限制条件下最小化二次函数。似乎偏置 b 从最优化问题中消失了。然而，它可以通过限制条件隐式地确定，因为这些限制条件要求 $\|\mathbf{w}\|$ 的改变需要通过 b 的改变进行补偿。我们稍后会看到它是如何工作的。

为了解决这个限制的最优化问题，我们引入拉格朗日乘数 $a_n \geq 0$ 。公式 (7.5) 中的每个限制条件都对应着一个乘数 a_n 。从而可得下面的拉格朗日函数

$$L(\mathbf{w}, b, \mathbf{a}) = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{n=1}^N a_n \{t_n (\mathbf{w}^T \phi(\mathbf{x}_n) + b) - 1\} \quad (7.7)$$

其中 $\mathbf{a} = (a_1, \dots, a_N)^T$ 。注意拉格朗日乘数项前面的负号，因为我们要关于 \mathbf{w} 和 b 最小化，关于 \mathbf{a} 最大化。令 $L(\mathbf{w}, b, \mathbf{a})$ 关于 \mathbf{w} 和 b 的导数等于零，我们得到了下面两个条件

$$\mathbf{w} = \sum_{n=1}^N a_n t_n \phi(\mathbf{x}_n) \quad (7.8)$$

$$0 = \sum_{n=1}^N a_n t_n \quad (7.9)$$

使用这两个条件从 $L(\mathbf{w}, b, \mathbf{a})$ 中消去 \mathbf{w} 和 b ，就得到了最大化边缘问题的对偶表示（dual representation），其中我们要关于 \mathbf{a} 最大化

$$\tilde{L}(\mathbf{a}) = \sum_{n=1}^N a_n - \frac{1}{2} \sum_{n=1}^N \sum_{m=1}^N a_n a_m t_n t_m k(\mathbf{x}_n, \mathbf{x}_m) \quad (7.10)$$

限制条件为

$$a_n \geq 0, \quad n = 1, \dots, N \quad (7.11)$$

$$\sum_{n=1}^N a_n t_n = 0 \quad (7.12)$$

这里，核函数被定义为 $k(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^T \phi(\mathbf{x}')$ 。与之前一样，这是一个二次规划问题，其中我们要在不等式限制条件下最优化一个 \mathbf{a} 的二次函数。我们会在 7.1.1 节讨论求解这种二次规划问题的方法。

M 个变量的二次规划问题的求解，通常的时间复杂度为 $O(M^3)$ 。通过将原始问题转化为对偶问题，我们将涉及到 M 个变量的最小化公式 (7.6) 的问题转化为了涉及到 N 个变量的对偶问题 (7.10)。对于一组固定的基函数，其中基函数的数量 M 小于数据点的数量 N ，转化为对偶问题似乎没有什么好处。但是，对偶问题使得模型能够用核函数重新表示，因此最大边缘分类器可以被高效地应用于维数超过数据点个数的特征空间，包括无穷维特征空间。核公式也让核函数 $k(\mathbf{x}, \mathbf{x}')$ 正定这一限制条件存在的原因变得更显然，因为这确保了拉格朗日函数 $\tilde{L}(\mathbf{a})$ 有上界，从而使得最优化问题有良好的定义。

为了使用训练过的模型分类新的数据点，我们计算公式 (7.1) 定义的 $y(\mathbf{x})$ 的符号。通过使用公式 (7.8) 消去 \mathbf{w} ， $y(\mathbf{x})$ 可以根据参数 $\{a_n\}$ 和核函数表示，即

$$y(\mathbf{x}) = \sum_{n=1}^N a_n t_n k(\mathbf{x}, \mathbf{x}_n) + b \quad (7.13)$$

在附录 E 中，我们说明了这种形式的限制的最优化问题满足 Karush-Kuhn-Tucker (KKT) 条件。在这个问题中，下面三个性质要成立。

$$a_n \geq 0 \quad (7.14)$$

$$t_n y(\mathbf{x}_n) - 1 \geq 0 \quad (7.15)$$

$$a_n \{t_n y(\mathbf{x}_n) - 1\} = 0 \quad (7.16)$$

因此对于每个数据点，要么 $a_n = 0$ ，要么 $t_n y(\mathbf{x}_n) = 1$ 。任何使得 $a_n = 0$ 的数据点都不会出现在公式 (7.13) 的求和式中，因此对新数据点的预测没有作用。剩下的数据点被称为支持向量（support vector）。由于这些支持向量满足 $t_n y(\mathbf{x}_n) = 1$ ，因此它们对应于特征空间中位于最大边缘超平面内的点，如图 7.1 所示。这个性质是支持向量机在实际应用中的核心。一旦模型被训练完毕，相当多的数据点都可以被丢弃，只有支持向量被保留。

解决了二次规划问题，找到了 \mathbf{a} 的值之后，注意到支持向量 \mathbf{x}_n 满足 $t_n y(\mathbf{x}_n) = 1$ ，我们就可以确定阈值参数 b 的值。使用公式 (7.13)，可得

$$t_n \left(\sum_{m \in \mathcal{S}} a_m t_m k(\mathbf{x}_n, \mathbf{x}_m) + b \right) = 1 \quad (7.17)$$



图 7.2: 二维空间中来自两个类别的人工生成数据的例子。图中画出了具有高斯核函数的支持向量机的得到的常数 $y(\mathbf{x})$ 的轮廓线。同时给出的时决策边界、边缘边界以及支持向量。

其中 \mathcal{S} 表示支持向量的下标集合。虽然我们可以使用任意选择的支持向量 \mathbf{x}_n 解这个关于 b 的方程，但是我们可以通过下面的方式得到一个在数值计算上更加稳定的解。首先乘以 t_n ，使用 $t_n^2 = 1$ 的性质，然后对于所有的支持向量，整理方程，解出 b ，可得

$$b = \frac{1}{N_{\mathcal{S}}} \sum_{n \in \mathcal{S}} \left(t_n - \sum_{m \in \mathcal{S}} a_m t_m k(\mathbf{x}_n, \mathbf{x}_m) \right) \quad (7.18)$$

其中 $N_{\mathcal{S}}$ 是支持向量的总数。

对于接下来的模型比较，我们可以将最大边缘分类器用带有简单二次正则化项的最小化误差函数表示，形式为

$$\sum_{n=1}^N E_{\infty}(y(\mathbf{x}_n)t_n - 1) + \lambda \|\mathbf{w}\|^2 \quad (7.19)$$

其中 $E_{\infty}(z)$ 是一个函数，当 $z \geq 0$ 时，函数值为零，其他情况下函数值为 ∞ 。这就确保了限制条件 (7.5) 成立。注意，只要正则化参数满足 $\lambda > 0$ ，那么它的精确值就没有作用。

图7.2给出了一个分类问题的例子。分类用的模型使用支持向量机训练，训练数据是一个简单的人工生成的数据集，核函数是公式 (6.23) 给出的高斯核。虽然数据点在二维空间中显然不是线性可分的，但是它在隐式地由非线性核函数定义的非线性特征空间中是线性可分的。因此，训练数据点在原始数据空间中被完美地分开了。

这个例子也从几何角度说明了SVM中稀疏性的来源。最大边缘超平面由支持向量的位置定义，其他数据点可以自由移动（只要仍然在边缘区域之外）而不改变决策边界，因此解与这些数据点无关。

7.1.1 重叠类分布

目前为止，我们假设训练数据点在特征空间 $\phi(\mathbf{x})$ 中是线性可分的。解得的支持向量机在原始输入空间 \mathbf{x} 中会对训练数据进行精确地划分，虽然对应的决策边界是非线性的。然而，在实际中，类条件分布可能重叠，这种情况下对训练数据的精确划分会导致较差的泛化能力。

因此我们需要一种方式修改支持向量机，允许一些训练数据点被误分类。根据公式 (7.19)，我们看到在可以分开的类别的情况下，我们隐式地使用了一个误差函数。当数据点被错误分类时，这个误差函数等于无穷大，而当数据点被正确分类时，这个误差函数等于零，这样就将模型参数优化为了最大化边缘。我们现在修改这种方法，使得数据点允许在边缘边界的“错误侧”，但是增加一个惩罚项，这个惩罚项随着与决策边界的距离的增大而增大。对于接下来的最优化问题，令这个惩罚项是距离的线性函数比较方便。为了完成这一点，我们引入松弛变量 (slack variable) $\xi_n \geq 0$ ，其中 $n = 1, \dots, N$ ，每个训练数据点都有一个松弛变量 (Bennett, 1992; Cortes and Vapnik, 1995)。对于位于正确的边缘边界内部的点或者边

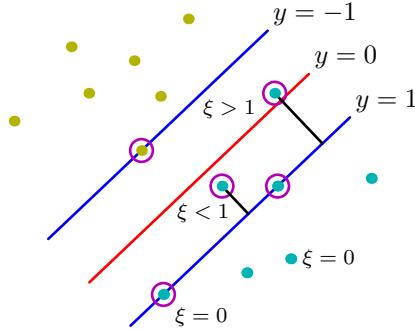


图 7.3: 松弛变量 $\xi_n \geq 0$ 的说明。圆圈标记的数据点是支持向量。

界上的点, $\xi_n = 0$, 对于其他点, $\xi_n = |t_n - y(\mathbf{x}_n)|$ 。因此, 对于位于决策边界 $y(\mathbf{x}_n) = 0$ 上的点, $\xi_n = 1$, 并且 $\xi_n > 1$ 的点就是被误分类的点。这样, 公式 (7.5) 给出的精确分类的限制条件就被替换为

$$t_n y(\mathbf{x}_n) \geq 1 - \xi_n, \quad n = 1, \dots, N \quad (7.20)$$

其中松弛变量被限制为满足 $\xi_n \geq 0$ 。 $\xi_n = 0$ 的数据点被正确分类, 要么位于边缘上, 要么在边缘的正确一侧。 $0 < \xi_n \leq 1$ 的点位于边缘内部, 但是在决策边界的正确一侧。 $\xi_n > 1$ 的点位于决策边界的错误一侧, 是被错误分类的点。如图7.3所示。这种方法有时被描述成放宽边缘的硬限制, 得到一个软边缘 (soft margin), 并且允许一些训练数据点被错分。注意, 虽然松弛变量允许类分布的重叠, 但是这个框架对于异常点很敏感, 因为误分类的惩罚随着 ξ 线性增加。

现在我们的目标是最大化边缘, 同时以一种比较柔和的方式惩罚位于边缘边界错误一侧的点。于是, 我们最小化

$$C \sum_{n=1}^N \xi_n + \frac{1}{2} \|\mathbf{w}\|^2 \quad (7.21)$$

其中参数 $C > 0$ 控制了松弛变量惩罚与边缘之间的折中。由于任何被误分类的数据点都有 $\xi_n > 1$, 因此 $\sum_n \xi_n$ 是误分类数据点数量的上界。于是, 参数 C 类似于 (作用相反的) 正则化系数, 因为它控制了最小化训练误差与模型复杂度之间的折中。在 $C \rightarrow \infty$ 的期限情况下, 我们就回到了之前讨论过的用于线性可分数据的支持向量机。

我们现在想要在公式 (7.20) 以及 $\xi_n \geq 0$ 的条件下最小化公式 (7.21)。对应的拉格朗日函数为

$$L(\mathbf{w}, b, \boldsymbol{\xi}, \mathbf{a}, \boldsymbol{\mu}) = \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{n=1}^N \xi_n - \sum_{n=1}^N a_n \{t_n y(\mathbf{x}_n) - 1 + \xi_n\} - \sum_{n=1}^N \mu_n \xi_n \quad (7.22)$$

其中 $\{a_n \geq 0\}$ 和 $\{\mu_n \geq 0\}$ 是拉格朗日乘数。对应的KKT条件为

$$a_n \geq 0 \quad (7.23)$$

$$t_n y(\mathbf{x}_n) - 1 + \xi_n \geq 0 \quad (7.24)$$

$$a_n (t_n y(\mathbf{x}_n) - 1 + \xi_n) = 0 \quad (7.25)$$

$$\mu_n \geq 0 \quad (7.26)$$

$$\xi_n \geq 0 \quad (7.27)$$

$$\mu_n \xi_n = 0 \quad (7.28)$$

其中 $n = 1, \dots, N$ 。

我们现在对 \mathbf{w}, b 和 $\{\xi_n\}$ 进行最优化。使用公式 (7.1) 给出的 $y(\mathbf{x})$ 的定义, 我们有

$$\frac{\partial L}{\partial \mathbf{w}} = 0 \Rightarrow \mathbf{w} = \sum_{n=1}^N a_n t_n \phi(\mathbf{x}_n) \quad (7.29)$$

$$\frac{\partial L}{\partial b} = 0 \Rightarrow \sum_{n=1}^N a_n t_n = 0 \quad (7.30)$$

$$\frac{\partial L}{\partial \xi_n} = 0 \Rightarrow a_n = C - \mu_n \quad (7.31)$$

使用这些结果，从拉格朗日函数中消去 w, b 和 $\{\xi_n\}$ ，我们得到了下面形式的拉格朗日函数

$$\tilde{L}(\mathbf{a}) = \sum_{n=1}^N a_n - \frac{1}{2} \sum_{n=1}^N \sum_{m=1}^N a_n a_m t_n t_m k(\mathbf{x}_n, \mathbf{x}_m) \quad (7.32)$$

这与线性可分的情况完全相同，唯一的区别就是限制条件多少有些差异。为了理解这些限制条件究竟是什么，我们注意到，由于 a_n 是拉格朗日乘数，因此必须有 $a_n \geq 0$ 。此外，公式 (7.31) 以及 $\mu_n \geq 0$ 表明 $a_n \leq C$ 。于是，我们关于对偶变量 $\{a_n\}$ 最大化公式 (7.32) 时必须要满足以下限制

$$0 \leq a_n \leq C \quad (7.33)$$

$$\sum_{n=1}^N a_n t_n = 0 \quad (7.34)$$

其中 $n = 1, \dots, N$ 。公式 (7.33) 被称为盒限制 (box constraint)。这又一次变成了一个二次规划的问题。如果我们将公式 (7.29) 代入公式 (7.1)，我们看到对于新数据点的预测又一次使用了公式 (7.13)。

我们现在可以表示最终的解。与之前一样，对于数据点的一个子集，有 $a_n = 0$ ，在这种情况下这些数据点对于预测模型 (7.13) 没有贡献。剩余的数据点组成了支持向量。这些数据点满足 $a_n > 0$ ，因此根据公式 (7.25)，它们必须满足

$$t_n y(\mathbf{x}_n) = 1 - \xi_n \quad (7.35)$$

如果 $a_n < C$ ，那么公式 (7.31) 表明 $\mu_n > 0$ ，根据公式 (7.28)，这要求 $\xi_n = 0$ ，从而这些点位于边缘上。 $a_n = C$ 的点位于边缘内部，并且如果 $\xi_n \leq 1$ 则被正确分类，如果 $\xi_n > 1$ 则分类错误。

为了确定公式 (7.1) 中的参数 b ，我们注意到 $0 < a_n < C$ 的支持向量满足 $\xi_n = 0$ 即 $t_n y(\mathbf{x}_n) = 1$ ，因此就满足

$$t_n \left(\sum_{m \in \mathcal{S}} a_m t_m k(\mathbf{x}_n, \mathbf{x}_m) + b \right) = 1 \quad (7.36)$$

与之前一样，一个对于数值计算比较稳定的解可以通过求平均的方式得到，结果为

$$b = \frac{1}{N_{\mathcal{M}}} \sum_{n \in \mathcal{M}} \left(t_n - \sum_{m \in \mathcal{S}} a_m t_m k(\mathbf{x}_n, \mathbf{x}_m) \right) \quad (7.37)$$

其中 \mathcal{M} 表示满足 $0 < a_n < C$ 的数据点的下标的集合。

支持向量机的另一种等价形式，被称为 ν -SVM，由Schölkopf et al. (2000) 提出。它涉及到最小化

$$\tilde{L}(\mathbf{a}) = -\frac{1}{2} \sum_{n=1}^N \sum_{m=1}^N a_n a_m t_n t_m k(\mathbf{x}_n, \mathbf{x}_m) \quad (7.38)$$

限制条件为

$$0 \leq a_n \leq \frac{1}{N} \quad (7.39)$$

$$\sum_{n=1}^N a_n t_n = 0 \quad (7.40)$$

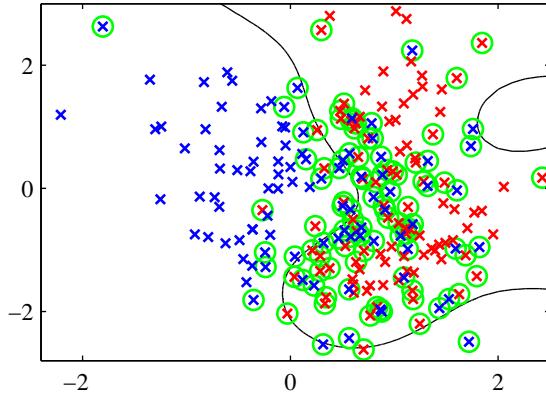


图 7.4: ν -SVM 应用于二维不可分数据集的例子。圆圈表示支持向量。

$$\sum_{n=1}^N a_n \geq \nu \quad (7.41)$$

这种方法的优点是，参数 ν 代替了参数 C ，它既可以被看做边缘错误（margin error）（ $\xi_n > 0$ 的点，因此就是位于边缘边界错误一侧的数据点，它可能被误分类也可能没被误分类）的上界，也可以被看做支持向量比例的下界。图 7.4 给出了 ν -SVM 用于人造数据集的一个例子。这里使用了形如 $\exp(-\gamma \|x - x'\|^2)$ 的高斯核，且 $\gamma = 0.45$ 。

虽然对新输入的预测只通过支持向量完成，但是训练阶段（即确定参数 a 和 b 的阶段）使用了整个数据集，因此找到一个解决二次规划问题的高效算法很重要。我们首先注意到由公式 (7.10) 或公式 (7.31) 给出目标函数 $\tilde{L}(a)$ 是二次的，因此如果限制条件定义了一个凸区域（由于限制条件的线性性质，实际情况确实是这样），那么任意局部最优解也是全局最优解。使用传统的方法直接求解二次规划问题通常是不可行的，因为需要的计算量和存储空间都相当大，因此我们需要寻找更实际的方法。分块 (chunking) 方法 (Vapnik, 1992) 利用了下面的事实：如果我们将核矩阵中对应于拉格朗日乘数等于零的行和列删除，那么拉格朗日函数不变。这使得完全的二次规划问题被分解为一系列小的二次规划问题，这些小的问题的目标是识别出所有的非零拉格朗日乘数，然后丢弃其他的。分块可以通过保护共轭梯度 (protected conjugate gradient) 方法实现 (Burges, 1998)。虽然分块可以将二次函数中矩阵的大小从数据点的个数的平方减小到近似等于非零拉格朗日乘数的个数的平方，但是对于大规模应用来说，这个数量仍然过大，从而内存无法满足要求。分解方法 (decomposition method) (Osuna et al., 1996) 也解决一系列较小的二次规划问题，但是这些问题被设计为具有同样的大小，因此这个方法可以应用于任意规模的数据集。然而，这种方法仍然涉及到二次规划子问题的数值解，求出这些数值解很困难，代价很高。一种最流行的训练支持向量机的方法被称为顺序最小化优化 (sequential minimal optimization)，或者称为 SMO (Platt, 1999)。这种方法考虑了分块方法的极限情况，每次只考虑两个拉格朗日乘数。这种情况下，子问题可以解析地求解，因此避免了数值二次规划。选择每一步骤中需要考虑的拉格朗日乘数对时，使用了启发式的方法。在实际应用中，SMO 与训练数据点数量的关系位于线性与二次之间，取决于具体的应用。

我们已经看到核函数对应于特征空间中的内积。特征空间可以是高维的，甚至是无穷维的。通过直接对核函数操作，而不显式地引入特征空间，支持向量机或许在一定程度上避免了维度灾难的问题。然而，事实并非如此，因为限制了特征空间维度的特征的值之间存在限制。为了说明这一点，考虑一个简单的二阶多项式核，我们可以用它的分量进行展开

$$\begin{aligned} k(\mathbf{x}, \mathbf{z}) &= (\mathbf{1} + \mathbf{x}^T \mathbf{z})^2 = (1 + x_1 z_1 + x_2 z_2)^2 \\ &= 1 + 2x_1 z_1 + 2x_2 z_2 + x_1^2 z_1^2 + 2x_1 z_1 x_2 z_2 + x_2^2 z_2^2 \\ &= (1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, \sqrt{2}x_1 x_2, x_2^2)(1, \sqrt{2}z_1, \sqrt{2}z_2, z_1^2, \sqrt{2}z_1 z_2, z_2^2)^T \\ &= \phi(\mathbf{x})^T \phi(\mathbf{z}) \end{aligned} \quad (7.42)$$

于是这个核函数表示六维特征空间中的一个内积，其中输入空间到特征空间的映射由向量函

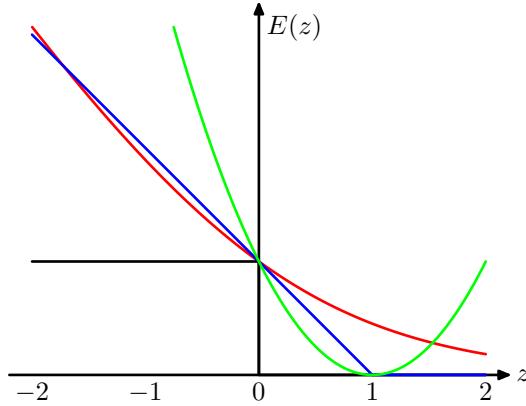


图 7.5: 支持向量机使用的“铰链”误差函数的图像，用蓝色表示。同时画出的还有logistic回归的误差函数，使用因子 $1/\ln(2)$ 重新放缩，从而通过点 $(0, 1)$ ，用红色表示。还画出了误分类误差函数（黑色）和平方误差函数（绿色）。

数 $\phi(x)$ 描述。然而，对这些特征加权的系数被限制为具体的形式。因此，原始二维空间 x 中的任意点集都会被限制到这个六维特征空间中的二维非线性流形中。

我们已经强调了这个事实：支持向量机不提供概率输出，而是对新的输入进行分类决策。Veropoulos et al. (1999) 讨论了对SVM的修改，使其能控制假阳性和假阴性之间的折中。然而，如果我们希望把SVM用作较大的概率系统中的一个模块，那么我们需要对于新的输入 x 的类别标签 t 的概率预测。

为了解决这个问题，Platt (2000) 提出了使用logistic sigmoid函数拟合训练过的支持向量机的输出的方法。具体来说，需要求解的条件概率被假设具有下面的形式

$$p(t = 1 | x) = \sigma(Ay(x) + B) \quad (7.43)$$

其中 $y(x)$ 由公式 (7.1) 定义。参数 A 和 B 的值通过最小化交叉熵误差函数的方式确定。交叉熵误差函数根据由 $y(x_n)$ 和 t_n 组成的训练数据集定义。用于拟合sigmoid函数的数据需要独立于训练原始SVM的数据，为了避免严重的过拟合现象。这种两个阶段的方法等价于假设支持向量机的输出 $y(x)$ 表示属于类别 $t = 1$ 的 x 的对数概率。由于SVM的训练过程并没有体现这种倾向，因此SVM给出的对后验概率的近似结果比较差 (Tipping, 2001)。

7.1.2 与logistic回归的关系

与线性可分的情形一样，对于线性不可分的概率分布，我们可以用最小化正则化的误差函数的方法重新表示SVM。这也使得我们能够强调与logistic回归模型之间的相似性和差别。

我们已经看到，对于位于边缘边界正确一侧的数据点，即满足 $y_n t_n \geq 1$ 的数据点，我们有 $\xi_n = 0$ ，对于剩余的数据点，我们有 $\xi_n = 1 - y_n t_n$ 。因此目标函数 (7.21) 可以写成（忽略整体的具有可乘性的常数）下面的形式

$$\sum_{n=1}^N E_{SV}(y_n t_n) + \lambda \|\mathbf{w}\|^2 \quad (7.44)$$

其中 $\lambda = (2C)^{-1}$ ， $E_{SV}(\cdot)$ 是铰链 (hinge) 误差函数，定义为

$$E_{SV}(y_n t_n) = [1 - y_n t_n]_+ \quad (7.45)$$

其中 $[\cdot]_+$ 表示正数部分。这个函数之所以被称为“铰链”误差函数，是因为它的形状，如图7.5所示。它可以被看做误分类误差函数的一个近似。误分类误差函数是我们在理想情况下希望最小化的函数，它也被画在了图7.5中。

当我们考虑4.3.2节的logistic回归模型的时候，我们发现比较方便的做法是对目标变量 $t \in \{0, 1\}$ 进行操作。为了与支持向量机进行对比，我们首先使用目标变量 $t \in \{-1, 1\}$ 重

写最大似然 logistic 回归函数。为了完成这一点，我们注意到 $p(t = 1 | y) = \sigma(y)$ ，其中 $y(\mathbf{x})$ 由公式 (7.1) 给出， $\sigma(y)$ 是公式 (4.59) 给出的 logistic sigmoid 函数。因此有 $p(t = -1 | y) = 1 - \sigma(y) = \sigma(-y)$ ，其中我们用到了 logistic sigmoid 函数的性质，因此我们有

$$p(t | y) = \sigma(yt) \quad (7.46)$$

从这个式子中我们通过对似然函数取负对数的方式构造一个误差函数。带有正则化项的误差函数的形式为

$$\sum_{n=1}^N E_{LR}(y_n t_n) + \lambda \|\mathbf{w}\|^2 \quad (7.47)$$

其中

$$E_{LR}(yt) = \ln(1 + \exp(-yt)) \quad (7.48)$$

为了与其他的误差函数进行比较，我们可以除以 $\ln(2)$ 使得误差函数通过点 $(0, 1)$ 。重新标度的误差函数也被画在了图 7.5 中。我们看到它的形式与支持向量机的误差函数类似。关键的区别在于 $E_{SV}(yt)$ 的平台区域产生了稀疏解。

logistic 误差函数与铰链损失都可以看成对误分类误差函数的连续近似。有时用于解决分类问题的另一个连续近似的误差函数时平方和误差函数，也被画在了图 7.5 中。但是，它具有下面的性质：它会着重强调那些被正确分类的在正确的一侧距离决策边界较远的点。如果这些点是误分类的点，那么这些点也会被赋予较高的权值。因此如果我们的目标是最小化分类错误率，那么一个单调递减的误差函数是一个更好的选择。

7.1.3 多类SVM

基本的支持向量机时一个两类分类器。然而在实际应用中，我们经常要处理涉及到 $K > 2$ 个类别的问题。于是，将多个两类 SVM 组合构造多类分类器的方法被提出来。

一种常用的方法 (Vapnik, 1998) 是构建 K 个独立的 SVM，其中第 k 个模型 $y_k(\mathbf{x})$ 在训练时，使用来自类别 C_k 的数据作为正例，使用来自剩余的 $K - 1$ 个类别的数据作为负例。这被称为“1 对剩余”(one-versus-the-rest) 方法。然而，在图 4.2 中，我们看到使用独立的分类器进行决策会产生不相容的结果，其中一个输入会同时被分配到多个类别中。这个问题有时可以这样解决：对于新的输入 \mathbf{x} ，使用下式做预测

$$y(\mathbf{x}) = \max_k y_k(\mathbf{x}) \quad (7.49)$$

不幸的是，这种启发式的方法会产生一个问题：不同的分类器是在不同的任务上进行训练的，无法保证不同分类器产生的实数值 $y_k(\mathbf{x})$ 具有恰当的标度。

“1 对剩余”方法的另一个问题是训练集合不平衡。例如，如果我们有 10 个类别，每个类别的训练数据点的数量相同，那么用于训练各个独立的分类器的训练数据由 90% 的负例和仅仅 10% 的正例组成，从而原始问题的对称性就消失了。Lee et al. (2001) 提出了“1 对剩余”方法的一种变体。这种变体修改了目标值，使得正例类别的目标值为 +1，负例类别的目标值为 $-\frac{1}{K-1}$ 。

Weston and Watkins (1999) 定义了一个单一目标函数用来同时训练所有的 K 个 SVM，基于的是最大化每个类别与其余剩余类别的边缘。然而，这会导致训练过程变慢，因为这种方法需要求解的不是 N 个数据点上的 K 个独立的最优化问题（整体代价为 $O(KN^2)$ ），而是要求解一个规模为 $(K - 1)N$ 的单一的最优化问题，整体代价为 $O(K^2 N^2)$ 。

另一种方法是在所有可能的类别对之间训练 $\frac{K(K-1)}{2}$ 个不同的二分类 SVM，然后将测试数据点分到具有最高“投票数”的类别中去。这种方法有时被称为“1 对 1”(one-versus-one)。与之前一样，我们从图 4.2 可以看到这会导致最终分类的歧义性。并且，对于较大的 K ，这种方法要比“1 对剩余”的方法花费更多的训练时间。类似地，为了计算数据点，这种方法需要更多的计算。

后一个问题可以通过将每对分类器组织成有向无环图（不要与概率图模型弄混淆）的方式解决，这就产生了 DAGSVM (Platt et al., 2000)。对于 K 个类别，DAGSVM 共有 $\frac{K(K-1)}{2}$ 个分类器。每次对新的测试点分类时，只需要 $K - 1$ 对分类器进行计算。选定的分类器是根据遍历图的路径确定的。

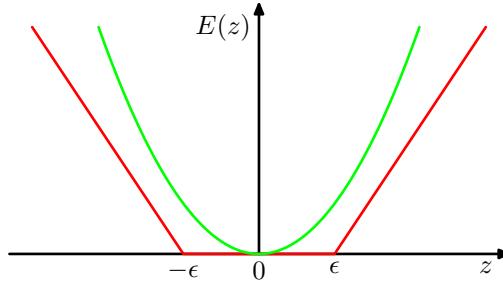


图 7.6: ϵ -不敏感误差函数（红色）的图像。在不敏感区域之外，误差函数值随着距离线性增大。作为对比，同时给出了二次误差函数（绿色）。

Dietterich and Bakiri (1995) 提出了一种不同的方法解决多分类问题。这种方法基于的是误差-修正输出编码，并且被Allwein et al. (2000) 用到支持向量机中。这种方法可以被看做“1对1”投票方法的一个推广。这种方法中，用来训练各个分类器的类别划分的方式更加一般。 K 个类别本身被表示为选定的两类分类器产生的响应的集合。结合一套合适的解码方法，这种方法对于错误以及各个分类器的输出的歧义性具有鲁棒性。虽然将SVM用于多分类问题仍然是一个没有标准答案的问题，但是在实际应用中，“1对剩余”是被最广泛使用的方法，尽管它有特定的形式，并且有着实际应用的局限性。

也存在单一类别 (single-class) 支持向量机，它解决与概率密度估计相关的无监督学习问题。但是，这种方法不是用来对数据的概率密度建模，而是想找到一个光滑的边界将高密度的区域包围起来。边界用来表示概率密度的等分点，即从概率密度分布中抽取的一个数据点落在某个区域的概率由一个0到1之间的固定的数给出，这个数事先指定好。与进行整体的密度估计相比，这个问题更加受限，但是对于某些具体的应用已经足够了。关于使用支持向量机解决这个问题，已经有两种方法被提出来。Schölkopf et al. (2001) 的算法尝试找到一个超平面，将训练数据中的固定比例 ν 的数据从原始数据集中分离，同时最大化超平面与原点之间的距离（边缘）。Tax and Duin (1999) 寻找特征空间中包含数据集的 ν 比例数据的最小球体。对于只是 $\mathbf{x} - \mathbf{x}'$ 的函数的核 $k(\mathbf{x}, \mathbf{x}')$ ，这两种算法等价。

7.1.4 回归问题的SVM

我们现在将支持向量机推广到回归问题，同时保持它的稀疏性。在简单的线性回归模型中，我们最小化一个正则化的误差函数

$$\frac{1}{2} \sum_{n=1}^N \{y_n - t_n\}^2 + \frac{\lambda}{2} \|\mathbf{w}\|^2 \quad (7.50)$$

为了得到稀疏解，二次误差函数被替换为一个 ϵ -不敏感误差函数 (ϵ -insensitive error function) (Vapnik, 1995)。如果预测 $y(\mathbf{x})$ 和目标 t 之间的差的绝对值小于 ϵ ，那么这个误差函数给出的误差等于零，其中 $\epsilon > 0$ 。 ϵ -不敏感误差函数的一个简单的例子是

$$E_\epsilon(y(\mathbf{x}) - t) = \begin{cases} 0, & \text{如果 } |y(\mathbf{x}) - t| < \epsilon \\ |y(\mathbf{x}) - t| - \epsilon & \text{其他情况} \end{cases} \quad (7.51)$$

它在不敏感区域之外，会有一个与误差相关联的线性代价。如图7.6所示。

于是我们最小化正则化的误差函数，形式为

$$C \sum_{n=1}^N E_\epsilon(y(\mathbf{x}) - t_n) + \frac{1}{2} \|\mathbf{w}\|^2 \quad (7.52)$$

其中 $y(\mathbf{x})$ 由公式 (7.1) 给出。按照惯例，(起着相反作用的) 正则化参数被记作 C ，出现在误差项之前。

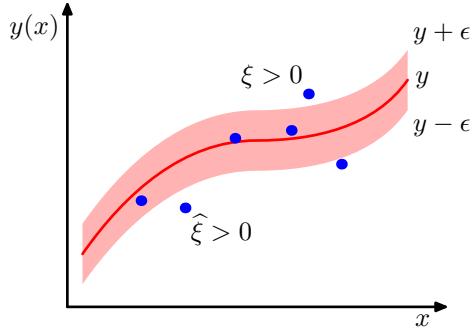


图 7.7: SVM 回归的说明。图中画出了回归曲线以及 ϵ -不敏感“管道”。同时给出的是松弛变量 ξ 和 $\hat{\xi}$ 的例子。对于 ϵ -管道上方的点， $\xi > 0$ 且 $\hat{\xi} = 0$ ，对于 ϵ -管道下方的点， $\xi = 0$ 且 $\hat{\xi} > 0$ ，对于 ϵ -管道内部的点， $\xi = \hat{\xi} = 0$ 。

与之前一样，通过引入松弛变量的方式，我们可以重新表达最优化问题。对于每个数据点 \mathbf{x}_n ，我们现在需要两个松弛变量 $\xi_n \geq 0$ 和 $\hat{\xi}_n \geq 0$ ，其中 $\xi_n > 0$ 对应于 $t_n > y(\mathbf{x}_n) + \epsilon$ 的数据点， $\hat{\xi}_n > 0$ 对应于 $t_n < y(\mathbf{x}_n) - \epsilon$ 的数据点，如图7.7所示。

目标点位于 ϵ -管道内的条件是 $y_n - \epsilon \leq t_n \leq y_n + \epsilon$ ，其中 $y_n = y(\mathbf{x}_n)$ 。引入松弛变量使得数据点能够位于管道之外，只要松弛变量不为零即可。对应的条件变为

$$t_n \leq y(\mathbf{x}_n) + \epsilon + \xi_n \quad (7.53)$$

$$t_n \geq y(\mathbf{x}_n) - \epsilon - \hat{\xi}_n \quad (7.54)$$

这样，支持向量回归的误差函数就可以写成

$$C \sum_{n=1}^N (\xi_n + \hat{\xi}_n) + \frac{1}{2} \|\mathbf{w}\|^2 \quad (7.55)$$

它必须在限制条件 $\xi_n \geq 0$ 和 $\hat{\xi}_n \geq 0$ 和公式 (7.53) 和公式 (7.54) 下进行最小化。可以这样做：引入拉格朗日乘数 $a_n \geq 0$, $\hat{a}_n \geq 0$, $\mu_n \geq 0$ 以及 $\hat{\mu}_n \geq 0$ ，然后最优化拉格朗日函数

$$\begin{aligned} L = & C \sum_{n=1}^N (\xi_n + \hat{\xi}_n) + \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{n=1}^N (\mu_n \xi_n + \hat{\mu}_n \hat{\xi}_n) \\ & - \sum_{n=1}^N a_n (\epsilon + \xi_n + y_n - t_n) - \sum_{n=1}^N \hat{a}_n (\epsilon + \hat{\xi}_n - y_n + t_n) \end{aligned} \quad (7.56)$$

我们现在使用公式 (7.1) 替换 $y(\mathbf{x})$ ，然后令拉格朗日函数关于 \mathbf{w}, b, ξ_n 和 $\hat{\xi}_n$ 的导数为零，有

$$\frac{\partial L}{\partial \mathbf{w}} = 0 \Rightarrow \mathbf{w} = \sum_{n=1}^N (a_n - \hat{a}_n) \phi(\mathbf{x}_n) \quad (7.57)$$

$$\frac{\partial L}{\partial b} = 0 \Rightarrow \sum_{n=1}^N (a_n - \hat{a}_n) = 0 \quad (7.58)$$

$$\frac{\partial L}{\partial \xi_n} = 0 \Rightarrow a_n + \mu_n = C \quad (7.59)$$

$$\frac{\partial L}{\partial \hat{\xi}_n} = 0 \Rightarrow \hat{a}_n + \hat{\mu}_n = C \quad (7.60)$$

使用这些结果消去拉格朗日函数中对应的变量，我们看到对偶问题涉及到关于 $\{a_n\}$ 和 $\{\hat{a}_n\}$ 最大化

$$\begin{aligned}\tilde{L}(\mathbf{a}, \hat{\mathbf{a}}) = & -\frac{1}{2} \sum_{n=1}^N \sum_{m=1}^N (a_n - \hat{a}_n)(a_m - \hat{a}_m) k(\mathbf{x}_n, \mathbf{x}_m) \\ & - \epsilon \sum_{n=1}^N (a_n + \hat{a}_n) + \sum_{n=1}^N (a_n - \hat{a}_n) t_n\end{aligned}\quad (7.61)$$

其中我们已经引入了核 $k(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^T \phi(\mathbf{x}')$ 。与之前一样，这是一个具有限制条件的最大化问题。为了找到限制条件，我们注意到 $a_n \geq 0$ 和 $\hat{a}_n \geq 0$ 必须成立，因为它们是拉格朗日乘数。并且 $\mu_n \geq 0$ 和 $\hat{\mu}_n \geq 0$ 以及公式 (7.59) 和公式 (7.60) 要求 $a_n \leq C$ 且 $\hat{a}_n \leq C$ ，因此我们又一次得到了盒限制

$$0 \leq a_n \leq C \quad (7.62)$$

$$0 \leq \hat{a}_n \leq C \quad (7.63)$$

以及条件 (7.58)。

将公式 (7.57) 代入公式 (7.1)，我们看到对于新的输入变量，可以使用下式进行预测

$$y(\mathbf{x}) = \sum_{n=1}^N (a_n - \hat{a}_n) k(\mathbf{x}, \mathbf{x}_n) + b \quad (7.64)$$

这又一次被表示为核函数的形式。

对应的Karush-Kuhn-Tucker (KKT) 条件说明了在解的位置，对偶变量与限制的乘积必须等于零，形式为

$$a_n(\epsilon + \xi_n + y_n - t_n) = 0 \quad (7.65)$$

$$\hat{a}_n(\epsilon + \hat{\xi}_n - y_n + t_n) = 0 \quad (7.66)$$

$$(C - a_n)\xi_n = 0 \quad (7.67)$$

$$(C - \hat{a}_n)\hat{\xi}_n = 0 \quad (7.68)$$

根据这些条件，我们能得到一些有用的结果。首先，我们注意到如果 $\epsilon + \xi_n + y_n - t_n = 0$ ，那么系数 a_n 只能非零，这表明数据点要么位于 ϵ -管道的上边界上 ($\xi_n = 0$)，要么位于上边界的上方 ($\xi_n > 0$)。类似地， \hat{a}_n 的非零值表示 $\epsilon + \hat{\xi}_n - y_n + t_n = 0$ ，这些点必须位于 ϵ -管道的下边界上或者下边界的下方。

此外，两个限制 $\epsilon + \xi_n + y_n - t_n = 0$ 和 $\epsilon + \hat{\xi}_n - y_n + t_n = 0$ 是不兼容的。可以这样证明：将两式相加，注意到 ξ_n 和 $\hat{\xi}_n$ 是非负的，而 ξ 是严格为正的，因此对于每个数据点 \mathbf{x}_n ， a_n 或者 \hat{a}_n 至少一个为零，或者都为零。

支持向量是对于由公式 (7.64) 给出的预测有贡献的数据点，换句话说，就是那些使得 $a_n \neq 0$ 或者 $\hat{a}_n \neq 0$ 成立的数据点。这些数据点位于 ϵ -管道边界上或者管道外部。管道内部的所有点都有 $a_n = \hat{a}_n = 0$ 。我们再次得到了一个稀疏解，在预测模型 (7.64) 中唯一必须计算的项就是涉及到支持向量的项。

参数 b 可以这样得到：考虑一个数据点，满足 $0 < a_n < C$ 。根据公式 (7.67)，一定有 $\xi_n = 0$ ，根据公式 (7.65)，一定有 $\epsilon + y_n - t_n = 0$ 。使用公式 (7.1)，然后求解 b ，我们有

$$\begin{aligned}b &= t_n - \epsilon - \mathbf{w}^T \phi(\mathbf{x}_n) \\ &= t_n - \epsilon - \sum_{m=1}^N (a_m - \hat{a}_m) k(\mathbf{x}_n, \mathbf{x}_m)\end{aligned}\quad (7.69)$$

其中我们使用了公式 (7.57)。通过考虑一个满足 $0 < \hat{a}_n < C$ 的数据点，我们可以得到一个类似的结果。在实际应用中，更好的做法是对所有的这些 b 的估计进行平均。

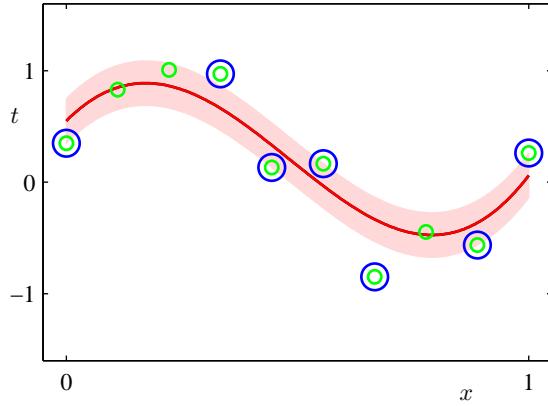


图 7.8: ν -SVM 回归应用到人工生成的正弦数据集上的说明, SVM 使用了高斯核。预测分布曲线为红色曲线, ϵ -不敏感管道对应于阴影区域。此外, 数据点用绿色表示, 支持向量用蓝色圆圈标记。

与分类问题的情形相同, 有另一种用于回归的SVM的形式。这种形式的SVM中, 控制复杂度的参数有一个更加直观的意义 (Schölkopf et al., 2000)。特别地, 我们不固定不敏感区域 ϵ 的宽度, 而是固定位于管道外部的数据点的比例 ν 。这涉及到最大化

$$\begin{aligned} \tilde{L}(\mathbf{a}, \hat{\mathbf{a}}) = & -\frac{1}{2} \sum_{n=1}^N \sum_{m=1}^N (a_n - \hat{a}_n)(a_m - \hat{a}_m) k(\mathbf{x}_n, \mathbf{x}_m) \\ & + \sum_{n=1}^N (a_n - \hat{a}_n)t_n \end{aligned} \quad (7.70)$$

限制条件为

$$0 \leq a_n \leq \frac{C}{N} \quad (7.71)$$

$$0 \leq \hat{a}_n \leq \frac{C}{N} \quad (7.72)$$

$$\sum_{n=1}^N (a_n - \hat{a}_n) = 0 \quad (7.73)$$

$$\sum_{n=1}^N (a_n + \hat{a}_n) \leq \nu C \quad (7.74)$$

可以证明至多有 νN 个数据点落在不敏感管道外部, 而至少有 νN 个数据点是支持向量, 因此位于管道上或者管道外部。

图7.8说明了使用支持向量机解决回归问题的一个例子, 数据集使用的是正弦曲线数据集。这里参数 ν 和 C 已经手动选择完毕。在实际应用中, 它们的值通常通过交叉验证的方法确定。

7.1.5 计算学习理论

历史上, 支持向量机大量地使用一个被称为计算学习理论 (computational learning theory) 的理论框架进行分析。这个框架有时候也被称为统计学习理论 (statistical learning theory) (Anthony and Biggs, 1992; Kearns and Vazirani, 1994; Vapnik, 1995; Vapnik, 1998)。这个框架起源于Valiant (1984), 他建立了概率近似正确 (probably approximately correct) 或者称为PAC的学习框架。PAC学习框架的目标是理解为两个给出较好的泛化能力, 需要多大的数据集。这个框架也给出了学习的计算代价的界限, 虽然我们不会在这里讨论。

假设我们从联合概率分布 $p(\mathbf{x}, t)$ 中抽取一个大小为 N 的数据集 \mathcal{D} , 其中 \mathbf{x} 是输入变量, t 表示类别标签。我们把注意力集中于“无噪声”的情况, 即类别标签由某个 (未知的) 判别函

数 $t = g(\mathbf{x})$ 确定。在PAC学习中，空间 \mathcal{F} 是一个以训练集 \mathcal{D} 为基础的函数组成的空间，我们从空间 \mathcal{F} 中抽取一个函数 $f(\mathbf{x}; \mathcal{D})$ ，如果它的期望错误率小于某个预先设定的阈值 ϵ ，即

$$\mathbb{E}_{\mathbf{x}, t} [I(f(\mathbf{x}; \mathcal{D}) \neq t)] < \epsilon \quad (7.75)$$

那么我们就说函数 $f(\mathbf{x}; \mathcal{D})$ 具有较好的泛化能力。其中 $I(\cdot)$ 是示性函数，期望是关于概率分布 $p(\mathbf{x}, t)$ 的期望。式子左侧的项是一个随机变量，因为它依赖于训练数据集 \mathcal{D} 。PAC框架要求，对于从概率分布 $p(\mathbf{x}, t)$ 中随机抽取的数据集 \mathcal{D} ，公式 (7.75) 成立的概率要大于 $1 - \delta$ 。这里 δ 是另一个预先设定的参数。术语“概率近似正确”来自于下面的要求：以一个较高的概率（大于 $1 - \delta$ ），使得错误率较小（小于 ϵ ）。对于一个给定的模型空间 \mathcal{F} ，以及给定的参数 ϵ 和 δ ，PAC学习的目标是提供满足这个准则所需的最小数据集规模 N 的界限。在PAC学习中，一个关键的量是Vapnik-Chervonenkis维度（Vapnik-Chervonenkis dimension），或者被称为VC维度。它提供了函数空间复杂度的一个度量，使得PAC框架能够扩展到包含无穷多个函数的空间。

在PAC框架中推导出的界限通常被看成是最坏的情况，因为它们适用于概率分布 $p(\mathbf{x}, t)$ 的任意选择，只要训练集和测试集是从相同的概率分布中（独立地）抽取即可，并且它们适用于函数 $f(\mathbf{x})$ 的任意选择，只要它属于 \mathcal{F} 即可。在真实世界的机器学习应用中，我们处理的分布通常有着很强的规律性，例如输入空间中的大片区域有着相同的类别标签。由于缺少关于分布形式的任何假设，因此PAC边界非常保守，换句话说，它们严重高估了得到给定的泛化性能所需的数据集的规模。因此，PAC界限几乎没有任何实际用处。

一种提升PAC界限的紧致程度的方法是PAC-贝叶斯框架（PAC-Bayesian framework）

(McAllester, 2003)，它考虑了空间 \mathcal{F} 上的函数的概率分布情况，有些类似于贝叶斯方法中的先验概率。这种方法仍然考虑任意可能的 $p(\mathbf{x}, t)$ 的选择，因此虽然这种方法得到的界限更加紧致，但是它们仍然是非常保守的。

7.2 相关向量机

支持向量机被用于一系列的分类和回归的应用中。尽管这样，支持向量机还是有许多局限性，某些局限性已经在本章中讨论过了。特别地，SVM的输出是一个决策结果而不是后验概率。并且，SVM最开始用于处理二分类问题，因此推广到 $K > 2$ 类有很多问题。有一个复杂度参数 C 或者 ν （以及回归问题中的参数 ϵ ）必须使用诸如交叉验证的方法确定。最后，预测是用核函数的线性组合表示的，核函数以训练数据点为中心，并且必须是正定的。

相关向量机（relevance vector machine）或者RVM（Tipping, 2001）是一个用于回归问题和分类问题的贝叶斯稀疏核方法，它具有许多SVM的特征，同时避免了SVM的主要的局限性。此外，它通常会产生更加稀疏的模型，从而使得在测试集上的速度更快，同时保留了可比的泛化误差。

与SVM不同，我们会发现比较方便的做法是首先介绍RVM的回归形式，然后将其扩展到分类任务中。

7.2.1 用于回归的RVM

用于回归的相关向量机的形式是第3章研究过的线性模型的形式，但是先验概率有所不同，从而产生了稀疏解。模型定义了给定一个输入向量 \mathbf{x} 的情况下，实值目标变量 t 的条件概率分布，形式为

$$p(t | \mathbf{x}, \mathbf{w}, \beta) = \mathcal{N}(t | y(\mathbf{x}), \beta^{-1}) \quad (7.76)$$

其中 $\beta = \sigma^{-2}$ 是噪声精度（噪声方差的倒数），均值是由一个线性模型给出，形式为

$$y(\mathbf{x}) = \sum_{i=1}^M w_i \phi_i(\mathbf{x}) = \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}) \quad (7.77)$$

模型带有固定非线性基函数 $\phi_i(\mathbf{x})$ ，通常包含一个常数项，使得对应的权参数表示一个“偏置”。

相关向量机是这个模型的一个具体实例，它试图重现支持向量机的结构。特别地，基函数由核给出，训练集的每个数据点关联着一个核。一般的表达式 (7.77) 于是就可以写成与SVM相类似的形式

$$y(\mathbf{x}) = \sum_{n=1}^N w_n k(\mathbf{x}, \mathbf{x}_n) + b \quad (7.78)$$

其中 b 是一个偏置参数。在目前的问题中，参数的数量为 $M = N + 1$ 。 $y(\mathbf{x})$ 与 SVM 的预测模型 (7.64) 具有相同的形式，唯一的差别是系数 a_n 在这里被记作 w_n 。应该强调的是，后面的分析对于任意的基函数的选择都成立。为了一般起见，我们将对公式 (7.77) 给出的形式进行操作。与 SVM 的情形相反，没有正定核的限制，基函数也没有被训练数据点的数量或位置所限制。

假设我们有输入向量 \mathbf{x} 的 N 次观测，我们将这些观测聚集在一起，记作数据矩阵 \mathbf{X} ，它的第 n 行是 \mathbf{x}_n^T ，其中 $n = 1, \dots, N$ 。对应的目标值为 $\mathbf{t} = (t_1, \dots, t_N)^T$ 。因此，似然函数为

$$p(\mathbf{t} | \mathbf{X}, \mathbf{w}, \beta) = \prod_{n=1}^N p(t_n | \mathbf{x}_n, \mathbf{w}, \beta) \quad (7.79)$$

接下来我们引入参数向量 \mathbf{w} 上的先验分布。与第3章一样，我们考虑零均值的高斯先验。然而，RVM 中的关键区别在于我们为每个权参数 w_i 都引入了一个单独的超参数 α_i ，而不是一个共享的超参数。因此权值先验的形式为

$$p(\mathbf{w} | \boldsymbol{\alpha}) = \prod_{i=1}^M \mathcal{N}(w_i | 0, \alpha_i^{-1}) \quad (7.80)$$

其中 α_i 表示对应参数 w_i 的精度， $\boldsymbol{\alpha}$ 表示 $(\alpha_1, \dots, \alpha_M)^T$ 。我们将会看到，当我们关于这些超参数最大化模型证据时，大部分都趋于无穷，对应的权参数的后验概率分布集中在零附近。与这些参数关联的基函数于是对于模型的预测没有作用，因此被高效地剪枝掉，从而生成了一个稀疏的模型。

使用公式 (3.49) 给出的线性模型的结果，我们看到权值的后验概率分布还是高斯分布，形式为

$$p(\mathbf{w} | \mathbf{t}, \mathbf{X}, \boldsymbol{\alpha}, \beta) = \mathcal{N}(\mathbf{w} | \mathbf{m}, \boldsymbol{\Sigma}) \quad (7.81)$$

其中，均值和方差为

$$\mathbf{m} = \beta \boldsymbol{\Sigma} \boldsymbol{\Phi}^T \mathbf{t} \quad (7.82)$$

$$\boldsymbol{\Sigma} = (\mathbf{A} + \beta \boldsymbol{\Phi}^T \boldsymbol{\Phi})^{-1} \quad (7.83)$$

其中， $\boldsymbol{\Phi}$ 是 $N \times M$ 的设计矩阵，元素为 $\Phi_{ni} = \phi_i(\mathbf{x}_n)$ ($i = 1, \dots, N$)，且 $\Phi_{nM} = 1$ ($n = 1, \dots, N$)， $\mathbf{A} = \text{diag}(\alpha_i)$ 。

α 和 β 的值可以使用第二类最大似然法（也被称为证据近似）来确定。这种方法中，我们最大化边缘似然函数。边缘似然函数通过对权向量积分的方式得到，即

$$\ln p(\mathbf{t} | \mathbf{X}, \boldsymbol{\alpha}, \beta) = \int p(\mathbf{t} | \mathbf{X}, \mathbf{w}, \beta) p(\mathbf{w} | \boldsymbol{\alpha}) d\mathbf{w} \quad (7.84)$$

由于这表示两个高斯分布的卷积，因此可以计算求得对数边缘似然函数，形式为

$$\begin{aligned} \ln p(\mathbf{t} | \mathbf{X}, \boldsymbol{\alpha}, \beta) &= \ln \mathcal{N}(\mathbf{t} | \mathbf{0}, \mathbf{C}) \\ &= -\frac{1}{2} \{ N \ln(2\pi) + \ln |\mathbf{C}| + \mathbf{t}^T \mathbf{C}^{-1} \mathbf{t} \} \end{aligned} \quad (7.85)$$

其中 $\mathbf{t} = (t_1, \dots, t_N)^T$ ，并且我们定义了 $N \times N$ 的矩阵 \mathbf{C} ，形式为

$$\mathbf{C} = \beta^{-1} \mathbf{I} + \boldsymbol{\Phi} \mathbf{A}^{-1} \boldsymbol{\Phi}^T \quad (7.86)$$

我们现在的目标是关于超参数 α 和 β 最大化公式 (7.85)。这只需要对3.5节给出的线性模型的证据近似进行微小的修改即可。我们可以区分出两种方法。第一种方法中，我们简单地令要解的边缘似然函数的导数等于零，然后得到了下面的重估计方程

$$\alpha_i^{\text{新}} = \frac{\gamma_i}{m_i^2} \quad (7.87)$$

$$(\beta^{\text{新}})^{-1} = \frac{\|\mathbf{t} - \Phi\mathbf{m}\|^2}{N - \sum_i \gamma_i} \quad (7.88)$$

其中 m_i 是公式 (7.82) 定义的后验均值 \mathbf{m} 的第*i*个分量。 γ_i 度量了对应的参数 w_i 由数据确定的效果，定义为

$$\gamma_i = 1 - \alpha_i \Sigma_{ii} \quad (7.89)$$

其中 Σ_{ii} 是公式 (7.83) 给出的后验协方差 Σ 的第*i*个对角元素。因此，学习过程按照下面的步骤进行：选择 α 和 β 的初始值，分别使用公式 (7.82) 和公式 (7.83) 计算后验概率的均值和协方差，然后交替地重新估计超参数（使用公式 (7.87) 和公式 (7.88) 进行）、重新估计后验均值和协方差（使用公式 (7.82) 和公式 (7.83) 进行），直到满足一个合适的收敛准则。

第二种方法是使用EM算法，将在9.3.4节讨论。这两种寻找最大化证据的超参数值的方法在形式上是等价的。然而，在数值计算上，使用对应于公式 (7.87) 和公式 (7.88) 的直接最优化方法可以更快地收敛 (Tipping, 2001)。

作为最优化的结果，我们发现超参数 $\{\alpha_i\}$ 的一部分趋于特别大的值（原则上是无穷大），因此对于这些超参数的权参数 w_i 的后验概率的均值和方差都是零。因此这些参数以及对应的基函数 $\phi_i(\mathbf{x})$ 被从模型中去掉，对于新输入的预测没有作用。在公式 (7.78) 给出的模型中，对于剩下的非零权值的输入 x_n 被称为相关向量 (relevance vector)，因为它们是通过自动相关性检测的方法得到的，类似于SVM中的支持向量。然而，值得强调的一点是，通过自动相关性检测得到概率模型的稀疏性的方法是一种相当通用的方法，可以应用于任何表示成基函数的可调节线性组合形式的模型。

找到了最大化边缘似然函数的超参数 α^* 和 β^* 的值之后，对于一个新的输入 \mathbf{x} ，我们可以计算 t 上的预测分布。使用公式 (7.76) 和公式 (7.81)，预测分布为

$$\begin{aligned} p(t | \mathbf{x}, \mathbf{X}, \mathbf{t}, \boldsymbol{\alpha}^*, \beta^*) &= \int p(t | \mathbf{x}, \mathbf{w}, \beta^*) p(\mathbf{w} | \mathbf{X}, \mathbf{t}, \boldsymbol{\alpha}^*, \beta^*) d\mathbf{w} \\ &= \mathcal{N}(t | \mathbf{m}^T \phi(\mathbf{x}), \sigma^2(\mathbf{x})) \end{aligned} \quad (7.90)$$

因此预测均值由公式 (7.76) 给出，其中 \mathbf{w} 被设置为后验均值 \mathbf{m} ，预测分布的方差为

$$\sigma^2(\mathbf{x}) = (\beta^*)^{-1} + \phi(\mathbf{x})^T \Sigma \phi(\mathbf{x}) \quad (7.91)$$

公式中的 Σ 由公式 (7.83) 给出，其中 α 和 β 被设置为了最优值 α^* 和 β^* 。这类似于公式 (3.59) 给出的线性回归模型的结果。回忆一下，对于局部的基函数，线性回归模型的预测方差在输入空间中没有基函数的区域会变小。于是，对于带有以数据点为中心的基函数的RVM的情形，当对数据以外的区域进行外插时，模型会对预测变得越来越确定 (Rasmussen and Quiñonero-Candela, 2005)，这当然不是我们想要的结果。高斯过程回归的预测分布没有这种问题。然而，高斯过程做预测的计算代价通常比RVM高得多。

图7.9给出了将RVM应用于正弦数据集回归问题的一个例子。这里，噪声精度 β 也通过证据最大化的方式确定。我们看到RVM中先关向量的数量比SVM中使用的支持向量的数量少得多。对于一大类回归任务和分类任务，RVM生成的模型通常比对应的支撑向量机生成的模型简洁了一个数量级，从而使得处理测试数据的速度有了极大的提升。值得注意的是，与SVM相比，这种稀疏性的增大并没有减小泛化误差。

与RVM相比，SVM的一个主要缺点是训练过程涉及到优化一个非凸的函数，并且与一个效果相似的SVM相比，训练时间要更长。对于有 M 个基函数的模型，RVM需要对一个 $M \times M$ 的矩阵求逆，这通常需要 $O(M^3)$ 次操作。在类似SVM的模型 (7.78) 这一具体情形下，我们有 $M = N + 1$ 。正如我们已经注意到的那样，存在训练SVM的高效方法，它的计算代价大致

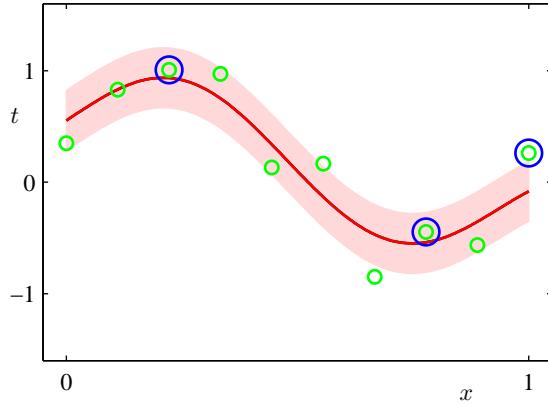


图 7.9: 使用与图7.8相同的数据集和相同的高斯核进行RVM回归的说明。RVM预测分布的均值用红色曲线表示，预测分布的一个标准差的位置用阴影区域表示。此外，数据点用绿色表示，相关向量用蓝色圆圈标记。注意，只有3个相关向量，而图7.8的 ν -SVM有7个支持向量。

是 N 的二次函数。当然，在RVM的情况下，我们总可以在开始时将基函数的数量设置为小于 $N + 1$ 。更重要的一点是，在相关向量机中，控制模型复杂度的参数以及噪声方差自动由一次训练过程确定，而在支持向量机中，参数 C 和 ϵ （或者 ν ）通常使用交叉验证的方法确定，这涉及到多次训练过程。此外，在下一节中，我们会推导另一种训练相关向量机的方法，它极大地提升了训练速度。

7.2.2 稀疏性分析

我们之前已经注意到自动相关性检测的过程使得参数的一个子集趋于零。我们现在更加详细地考察相关向量机的稀疏性的原理。在这个过程中，我们会得到一个与之前的直接方法相比更快的最优化超参数的方法。

关于贝叶斯线性模型的稀疏性的来源，在进行数学的分析之前，我们首先给出一些非形式化的观点。考虑一个数据集，这个数据集由 $N = 2$ 个观测 t_1 和 t_2 组成。我们有一个模型，它有一个基函数 $\phi(\mathbf{x})$ ，超参数为 α ，以及一个各向同性的噪声，精度为 β 。根据公式 (7.85)，边缘似然函数为 $p(\mathbf{t} | \alpha, \beta) = \mathcal{N}(\mathbf{t} | \mathbf{0}, \mathbf{C})$ ，其中协方差矩阵的形式为

$$\mathbf{C} = \frac{1}{\beta} \mathbf{I} + \frac{1}{\alpha} \boldsymbol{\varphi} \boldsymbol{\varphi}^T \quad (7.92)$$

其中 $\boldsymbol{\varphi}$ 表示 N 维向量 $(\phi(\mathbf{x}_1), \phi(\mathbf{x}_2))^T$ ，类似地 $\mathbf{t} = (t_1, t_2)^T$ 。注意，这是 \mathbf{t} 上的一个零均值的高斯过程模型，协方差为 \mathbf{C} 。给定 \mathbf{t} 的一个特定的观测，我们的目标是通过最大化边缘似然函数的方法找到 α^* 和 β^* 。从图7.10中，我们看到，如果 $\boldsymbol{\varphi}$ 的方向与训练数据向量 \mathbf{t} 之间没有很好地对齐的话，那么对应的超参数 α 会趋于 ∞ ，基向量会被从模型中剪枝掉。这种现象出现的原因是 α 的任意有限值总会给数据一个较低的概率，因此就减小了 \mathbf{t} 的值，假设 β 被设置为最优值。我们看到 α 的任意有限值会使得分布在远离数据的方向被拉长，从而增加了远离观测数据的区域的概率质量，因此就减小了目标数据向量本身概率密度的值。对于更一般的 M 个基向量 $\boldsymbol{\varphi}_1, \dots, \boldsymbol{\varphi}_M$ 的情形，也有类似的直观含义，即如果垂直的基向量与数据向量 \mathbf{t} 没有很好地对齐，那么它很可能被从模型中剪枝掉。

我们现在从一个更加数学的角度，对于涉及到 M 个基函数的一般情形，考察稀疏性的原理。为了进行这个分析，我们首先注意到，在公式 (7.87) 给出的参数 α_i 的重新估计的结果中，右侧的项本身也是 α_i 的函数。于是这些结果表示隐式解，需要用迭代的方式求出，即使对于所有的 $j \neq i$ 的 α_j 都固定时，确定单一的 α_i 也需要迭代。

这给出了解决RVM的最优化问题的一个不同的方法，其中我们显式地写出边缘似然函数 (7.85) 中所有对特定的 α_i 的依赖关系，然后显式地确定驻点 (Faul and Tipping, 2002; Tipping and Faul, 2003)。为了完成这一点，我们首先写出由公式 (7.86) 定义的矩阵 \mathbf{C} 中来自 α_i 的贡

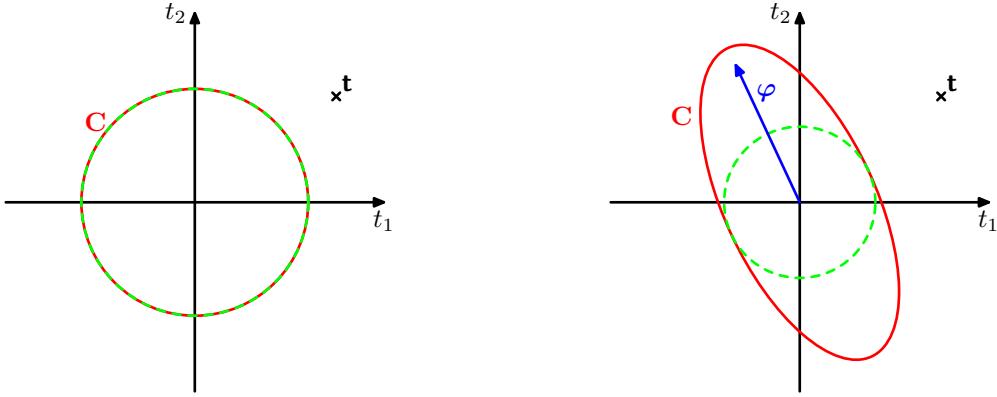


图 7.10: 贝叶斯线性回归模型的稀疏性的原理说明。图中给出了目标值的一组训练向量, 形式为 $\mathbf{t} = (t_1, t_2)^T$, 用叉号表示, 模型有一个基向量 $\varphi = (\phi(\mathbf{x}_1), \phi(\mathbf{x}_2))^T$, 它与目标数据向量 \mathbf{t} 的对齐效果很差。左图中, 我们看到一个只有各向同性的噪声的模型, 因此 $\mathbf{C} = \beta^{-1} \mathbf{I}$, 对应于 $\alpha = \infty$, β 被设置为概率最高的值。右图中, 我们看到了同样的模型, 但是 α 的值变成了有限值。在两种情况下, 红色椭圆都对应于单位马氏距离, $|\mathbf{C}|$ 对于两幅图的取值相同, 而绿色虚线圆表示由项 β^{-1} 产生的噪声的贡献。我们看到 α 的任意有限值减小了观测数据的概率, 因此对于概率最高的解, 基向量被移除。

献, 即

$$\begin{aligned}\mathbf{C} &= \beta^{-1} \mathbf{I} + \sum_{j \neq i} \alpha_j^{-1} \varphi_j \varphi_j^T + \alpha_i^{-1} \varphi_i \varphi_i^T \\ &= \mathbf{C}_{-i} + \alpha_i^{-1} \varphi_i \varphi_i^T\end{aligned}\quad (7.93)$$

其中 φ_i 表示矩阵 Φ 的第 i 列, 即 N 维向量, 元素为 $(\phi_i(\mathbf{x}_1), \dots, \phi_i(\mathbf{x}_N))$ 。这与 ϕ_n 不同, 它表示的是 Φ 的第 n 行。矩阵 \mathbf{C}_{-i} 表示将基函数 i 的贡献删除之后的矩阵 \mathbf{C} 。使用矩阵恒等式 (C.7) 和 (C.15), 矩阵 \mathbf{C} 的行列式和逆矩阵可以写成

$$|\mathbf{C}| = |\mathbf{C}_{-i}| (1 + \alpha_i^{-1} \varphi_i^T \mathbf{C}_{-i}^{-1} \varphi_i) \quad (7.94)$$

$$\mathbf{C}^{-1} = \mathbf{C}_{-i}^{-1} - \frac{\mathbf{C}_{-i}^{-1} \varphi_i \varphi_i^T \mathbf{C}_{-i}^{-1}}{\alpha_i + \varphi_i^T \mathbf{C}_{-i}^{-1} \varphi_i} \quad (7.95)$$

使用这些结果, 我们可以将对数边缘似然函数 (7.85) 写成下面的形式。

$$L(\boldsymbol{\alpha}) = L(\boldsymbol{\alpha}_{-i}) + \lambda(\alpha_i) \quad (7.96)$$

其中 $L(\boldsymbol{\alpha}_{-i})$ 是省略了基函数 φ_i 的对数边缘似然函数, $\lambda(\alpha_i)$ 被定义为

$$\lambda(\alpha_i) = \frac{1}{2} \left[\ln \alpha_i - \ln(\alpha_i + s_i) + \frac{q_i^2}{\alpha_i + s_i} \right] \quad (7.97)$$

包含了所有依赖于 α_i 的项。这里我们引入了两个量

$$s_i = \varphi_i^T \mathbf{C}_{-i}^{-1} \varphi_i \quad (7.98)$$

$$q_i = \varphi_i^T \mathbf{C}_{-i}^{-1} \mathbf{t} \quad (7.99)$$

这里 s_i 被称为稀疏度 (sparsity), q_i 被称为 φ_i 的质量 (quality), 并且正如我们将要看到的那样, s_i 的值相对于 q_i 的值较大意味着基函数 φ_i 更可能被模型剪枝掉。“稀疏度”度量了基函数 φ_i 与模型中其他基函数重叠的程度, “质量”度量了基向量 φ_i 与误差向量之间的对齐程度, 其中误差向量是训练值 $\mathbf{t} = (t_1, \dots, t_N)^T$ 与会导致 φ_i 从模型中被删除掉的预测向量 \mathbf{y}_{-i} 之间的差值 (Tipping and Faul, 2003)。



图 7.11: 对数边缘似然 $\lambda(\alpha_i)$ 与 $\ln \alpha_i$ 的图像。左图中，单一的最大值出现在有限的 α_i 处，此时 $q_i^2 = 4$ 且 $s_i = 1$ (从而 $q_i^2 > s_i$)。右图中，最大值位于 $\alpha_i = \infty$ 的位置，此时 $q_i^2 = 1$ 且 $s_i = 2$ (从而 $q_i^2 < s_i$)。

在边缘似然函数关于 α_i 的驻点处，导数

$$\frac{d\lambda(\alpha_i)}{d\alpha_i} = \frac{\alpha_i^{-1}s_i^2 - (q_i^2 - s_i)}{2(\alpha_i + s_i)^2} \quad (7.100)$$

等于零。有两种可能形式的解。回忆一下 $\alpha_i \geq 0$ ，我们看到如果 $q_i^2 < s_i$ ，那么 $\alpha_i \rightarrow \infty$ 提供了一个解。相反，如果 $q_i^2 > s_i$ ，我们可以解出 α_i ，得

$$\alpha_i = \frac{s_i^2}{q_i^2 - s_i} \quad (7.101)$$

图 7.11 给出了这两个解。我们看到质量项和稀疏度项的相对大小确定了一个特定的基向量是否会被模型剪枝掉。一个更加复杂的分析 (Faul and Tipping, 2002) 基于边缘似然函数的二阶导数，确保了这些解确实是 $\lambda(\alpha_i)$ 的唯一最大值。

注意，在给定其他超参数的值的情况下，这种方法产生了 α_i 的一个解析解。结合对于 RVM 中稀疏性来源的分析，上述分析也产生了一个高速最优化超参数的实用算法。这种算法使用固定候选基向量集合，然后在集合上循环，确定每个向量是否应该被包含在模型中。最终的顺序稀疏贝叶斯学习算法描述如下。

- 如果求解回归问题，初始化 β 。
- 使用一个基函数 φ_1 进行初始化，用公式 (7.101) 确定超参数 α_1 ，其余的 $j \neq 1$ 的超参数 α_j 被初始化为无穷大，从而只有 φ_1 被包含在模型中。
- 对于所有基函数，计算 Σ 和 m ，以及 q_i 和 s_i 。
- 选择一个候选的基函数 φ_i 。
- 如果 $q_i^2 > s_i$ 且 $\alpha_i < \infty$ ，从而基向量 φ_i 已经被包含在了模型中，那么使用公式 (7.101) 更新 α_i 。
- 如果 $q_i^2 > s_i$ 且 $\alpha_i = \infty$ ，那么将 φ_i 添加到模型中，使用公式 (7.101) 计算 α_i 。
- 如果 $q_i^2 \leq s_i$ 且 $\alpha_i < \infty$ ，那么从模型中删除基函数 φ_i ，令 $\alpha_i = \infty$ 。
- 如果求解回归问题，更新 β 。
- 如果收敛，则算法终止，否则回到第 3 步。

注意，如果 $q_i^2 \leq s_i$ 且 $\alpha_i = \infty$ ，那么基函数 φ_i 已经从模型中被去除掉了，不需要采取动作。

在实际应用中，比较方便的做法是计算下面的量

$$Q_i = \varphi_i^T C^{-1} t \quad (7.102)$$

$$S_i = \boldsymbol{\varphi}_i^T \mathbf{C}^{-1} \boldsymbol{\varphi}_i \quad (7.103)$$

这样，质量和稀疏性变量可以表示为

$$q_i = \frac{\alpha_i Q_i}{\alpha_i - S_i} \quad (7.104)$$

$$s_i = \frac{\alpha_i S_i}{\alpha_i - S_i} \quad (7.105)$$

注意，当 $\alpha_i = \infty$ 时，我们有 $q_i = Q_i$ 以及 $s_i = S_i$ 。使用公式 (C.7)，我们有

$$Q_i = \beta \boldsymbol{\varphi}_i^T \mathbf{t} - \beta^2 \boldsymbol{\varphi}_i^T \boldsymbol{\Phi} \boldsymbol{\Sigma} \boldsymbol{\Phi}^T \mathbf{t} \quad (7.106)$$

$$S_i = \beta \boldsymbol{\varphi}_i^T \boldsymbol{\varphi}_i - \beta^2 \boldsymbol{\varphi}_i^T \boldsymbol{\Phi} \boldsymbol{\Sigma} \boldsymbol{\Phi}^T \boldsymbol{\varphi}_i \quad (7.107)$$

其中 $\boldsymbol{\Phi}$ 和 $\boldsymbol{\Sigma}$ 只涉及到对应于有限的超参数 α_i 的基向量。在每个阶段，需要的计算量为 $O(M^3)$ ，其中 M 是模型中激活的基向量的数量，通常比训练模式的数量 N 要小得多。

7.2.3 RVM用于分类

我们可以将相关向量机框架推广到分类问题，推广的方法是将权值的ARD先验应用到第4章研究过的概率线性分类模型上。首先，我们考虑二分类问题，目标变量是二值变量 $t \in \{0, 1\}$ 。这个模型现在的形式为基函数的线性组合经过logistic sigmoid函数的变换，即

$$y(\mathbf{x}, \mathbf{w}) = \sigma(\mathbf{w}^T \boldsymbol{\phi}(\mathbf{x})) \quad (7.108)$$

其中 $\sigma(\cdot)$ 是公式 (4.59) 定义的logistic sigmoid函数。如果我们引入权值 \mathbf{w} 上的高斯先验，那么我们就得到了第4章讨论过的模型。这里的区别在于，在RVM中，模型使用的是ARD先验 (7.80)，其中每个权值参数有一个独立的精度超参数。

与回归模型不同，我们不在对参数向量 \mathbf{w} 解析地求积分。这里，我们按照Tipping (2001) 的方法，使用拉普拉斯近似，它曾经被应用于一个密切相关的问题，即4.5.1节介绍的贝叶斯logistic回归。

首先，我们初始化超参数向量 $\boldsymbol{\alpha}$ 。对于这个给定的 $\boldsymbol{\alpha}$ 值，我们接下来对后验概率建立一个高斯近似，从而得到了对边缘似然的一个近似。这个近似后的边缘似然函数的最大化就引出了对 $\boldsymbol{\alpha}$ 值的重新估计，并且这个过程不断重复，直到收敛。

让我们详细研究这个模型的拉普拉斯近似。对于固定的 $\boldsymbol{\alpha}$ 值， \mathbf{w} 的后验概率分布的众数可以通过最大化下式得到

$$\begin{aligned} \ln p(\mathbf{w} | \mathbf{t}, \boldsymbol{\alpha}) &= \ln \{p(\mathbf{t} | \mathbf{w})p(\mathbf{w} | \boldsymbol{\alpha})\} - \ln p(\mathbf{t} | \boldsymbol{\alpha}) \\ &= \sum_{n=1}^N \{t_n \ln y_n + (1 - t_n) \ln(1 - y_n)\} - \frac{1}{2} \mathbf{w}^T \mathbf{A} \mathbf{w} + \text{常数} \end{aligned} \quad (7.109)$$

其中 $\mathbf{A} = \text{diag}(\alpha_i)$ 。最大化可以使用4.3.3节讨论的迭代重加权最小平方 (IRLS) 方法完成。对于这个算法，我们需要求出对数后验概率分布的梯度向量和Hessian矩阵。根据公式 (7.109)，结果为

$$\nabla \ln p(\mathbf{w} | \mathbf{t}, \boldsymbol{\alpha}) = \boldsymbol{\Phi}^T (\mathbf{t} - \mathbf{y}) - \mathbf{A} \mathbf{w} \quad (7.110)$$

$$\nabla \nabla \ln p(\mathbf{w} | \mathbf{t}, \boldsymbol{\alpha}) = -(\boldsymbol{\Phi}^T \mathbf{B} \boldsymbol{\Phi} + \mathbf{A}) \quad (7.111)$$

其中 \mathbf{B} 是一个 $N \times N$ 的对角矩阵，元素为 $b_n = y_n(1 - y_n)$ 。向量 $\mathbf{y} = (y_1, \dots, y_N)^T$ ，矩阵 $\boldsymbol{\Phi}$ 是设计矩阵，元素为 $\Phi_{ni} = \phi_i(\mathbf{x}_n)$ 。这里，我们使用到了logistic sigmoid函数的导数的性质 (4.88)。在IRLS算法收敛的位置，负Hessian矩阵表示后验概率分布的高斯近似的协方差矩阵的逆矩阵。

后验概率的高斯近似的众数，对应于高斯近似的均值，可以通过令公式 (7.110) 等于零求得。得到的拉普拉斯近似的均值和方差的形式为

$$\mathbf{w}^* = \mathbf{A}^{-1} \boldsymbol{\Phi}^T (\mathbf{t} - \mathbf{y}) \quad (7.112)$$

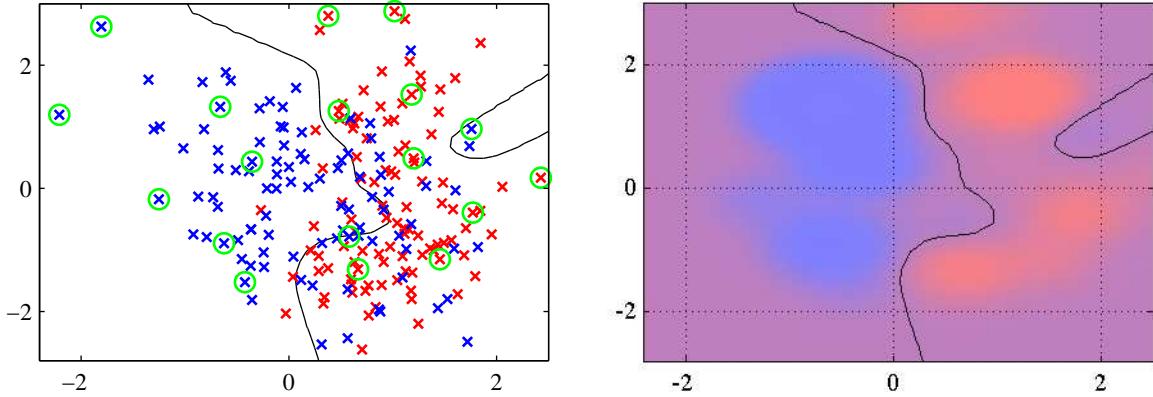


图 7.12: 相关向量机应用于人工数据集的说明。左图给出了决策边界和数据点，相关向量用圆圈标记出。将这个结果与图7.4给出的支持向量机的结果进行比较，表明RVM得到了更稀疏的模型。右图画出了由RVM给出的后验概率分布，其中红色（蓝色）所占的比重表示数据点属于红色（蓝色）类别的概率。

$$\Sigma = (\Phi^T B \Phi + A)^{-1} \quad (7.113)$$

我们现在使用这个拉普拉斯近似来计算边缘似然函数。使用公式 (4.135) 给出的使用拉普拉斯近似计算的积分的一般结果，我们有

$$\begin{aligned} p(\mathbf{t} | \boldsymbol{\alpha}) &= \int p(\mathbf{t} | \mathbf{w}) p(\mathbf{w} | \boldsymbol{\alpha}) d\mathbf{w} \\ &\simeq p(\mathbf{t} | \mathbf{w}^*) p(\mathbf{w}^* | \boldsymbol{\alpha}) (2\pi)^{\frac{M}{2}} |\Sigma|^{\frac{1}{2}} \end{aligned} \quad (7.114)$$

如果我们代入 $p(\mathbf{t} | \mathbf{w}^*)$ 和 $p(\mathbf{w}^* | \boldsymbol{\alpha})$ 的表达式，然后令边缘似然函数关于 α_i 的导数等于零，我们有

$$-\frac{1}{2}(w_i^*)^2 + \frac{1}{2\alpha_i} - \frac{1}{2}\Sigma_{ii} = 0 \quad (7.115)$$

定义 $\gamma_i = 1 - \alpha_i \Sigma_{ii}$ ，整理，可得

$$\alpha_i^{\text{新}} = \frac{\gamma_i}{(w_i^*)^2} \quad (7.116)$$

这与回归RVM的重估计公式 (7.87) 相同。

如果我们定义

$$\hat{\mathbf{t}} = \Phi \mathbf{w}^* + B^{-1}(\mathbf{t} - \mathbf{y}) \quad (7.117)$$

那么我们可以将近似对数边缘似然函数写成下面的形式

$$\ln p(\mathbf{t} | \boldsymbol{\alpha}) = -\frac{1}{2} \left\{ N \ln(2\pi) + \ln |\mathbf{C}| + (\hat{\mathbf{t}})^T \mathbf{C}^{-1} \hat{\mathbf{t}} \right\} \quad (7.118)$$

其中

$$\mathbf{C} = \mathbf{B} + \Phi \mathbf{A} \Phi^T \quad (7.119)$$

这与回归问题得到的公式 (7.85) 形式相同，因此我们可以应用同样的稀疏性分析的过程，得到同样的快速学习算法，这种算法中，我们在每一步最优化单独的一个超参数 α_i 。

图7.12给出了将相关向量机应用于人工生成的分类数据上的结果。我们看到相关向量倾向于不在决策边界区域内，这与支持向量机恰好相反。这与我们之前对于RVM的分析是相容的，因为以位于决策边界附近的数据点为中心的基函数 $\phi_i(\mathbf{x})$ 会产生一个向量 φ_i ，它与训练数据向量 \mathbf{t} 的对齐效果较差。

与SVM相比，相关向量机的一个潜在的优势是，它做出了概率形式的预测。例如，对于视频流人脸跟踪的线性动态系统的非线性扩展，可以用RVM来辅助构建它的发射概率密度 (Williams et al., 2005)。

目前为止，我们已经考虑了二分类问题的RVM。对于 $K > 2$ 个类别的情形，我们再次使用4.3.4节中的概率方法。这种方法中，有 K 个线性模型，形式为

$$a_k = \mathbf{w}_k^T \mathbf{x} \quad (7.120)$$

这些模型使用softmax函数进行组合，给出下面形式的输出。

$$y_k(\mathbf{x}) = \frac{\exp(a_k)}{\sum_j \exp(a_j)} \quad (7.121)$$

这样，对数似然函数为

$$\ln p(\mathbf{T} | \mathbf{w}_1, \dots, \mathbf{w}_K) = \prod_{n=1}^N \prod_{k=1}^K y_{nk}^{t_{nk}} \quad (7.122)$$

其中，对于每个数据点 n ， t_{nk} 的表示方式是“1-of- K ”的形式， \mathbf{T} 是一个矩阵，元素为 t_{nk} 。与之前一样，拉普拉斯近似可以用来最优化超参数（Tipping, 2001），其中模型和Hessian矩阵可以使用IRLS算法得到。与支持向量机使用的“类别对”形式的方法相比，RVM对多分类问题的处理的基础更加牢固，并且对于新的数据点，能够给出概率形式的预测。主要的缺点是，Hessian矩阵的维度为 $MK \times MK$ ，其中 M 是激活的基函数的数量，这使得与二分类的RVM相比，训练的计算代价多了一个额外的 K^3 因子。

相关向量机的主要缺点是，与SVM相比，训练时间相对较长。但是，RVM避免了通过交叉验证确定模型复杂度的过程，从而补偿了训练时间的劣势。此外，因为它产生的模型更稀疏，所以它对于测试点进行预测的计算时间通常更短，而对于测试点的计算时间通常在实际应用中更加重要。

7.3 练习

(7.1) (***) 假设我们有一个输入向量 $\{\mathbf{x}_n\}$ 以及对应的目标值 $t_n \in \{-1, 1\}$ 组成的数据集，并且假设我们使用Parzen核密度估计（见2.5.1节）对每个类别内部的输入向量的概率密度分别建模，核函数为 $k(\mathbf{x}, \mathbf{x}')$ 。假设两个类别具有相等的先验概率，写出最小错误分类的决策规则。证明，如果核函数为 $k(\mathbf{x}, \mathbf{x}') = \mathbf{x}^T \mathbf{x}'$ ，那么分类规则会简化为将新的输入向量分配到距离最近的均值的类别。最后，证明，如果核函数为 $k(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^T \phi(\mathbf{x}')$ ，那么分类基于的是特征空间 $\phi(\mathbf{x})$ 中的距离最近的均值。

(7.2) (*) 证明，如果限制条件 (7.5) 右侧的1被替换为某个任意的常数 $\gamma > 0$ ，那么最大边缘超平面的解保持不变。

(7.3) (***) 证明，与数据空间的维度无关，一个只有两个数据点的数据集（每个点属于一个类别）足以确定最大边缘超平面的位置。

(7.4) (***) 证明最大边缘超平面的边缘 ρ 为

$$\frac{1}{\rho^2} = \sum_{n=1}^N a_n \quad (7.123)$$

其中 $\{a_n\}$ 通过在限制条件 (7.11) 和 (7.12) 下最大化 (7.10) 的方式得到。

(7.5) (***) 证明上一个练习中的 ρ 和 $\{a_n\}$ 也满足

$$\frac{1}{\rho^2} = 2\tilde{L}(\mathbf{a}) \quad (7.124)$$

其中 $\tilde{L}(\mathbf{a})$ 由公式 (7.10) 定义。类似地，证明

$$\frac{1}{\rho^2} = \|\mathbf{w}\|^2 \quad (7.125)$$

(7.6) (*) 考虑一个logistic回归模型，目标变量为 $t \in \{-1, 1\}$ 。如果我们定义 $p(t=1 | \mathbf{y}) = \sigma(\mathbf{y})$ ，其中 $\mathbf{y}(\mathbf{x})$ 由公式 (7.1) 给出，证明负对数似然函数，加上二次正则化项，形式为 (7.47)。

(7.7) (*) 考虑回归支持向量机的拉格朗日函数 (7.56)。通过令拉格朗日函数关于 \mathbf{w}, b, ξ_n 和 $\hat{\xi}_n$ 的导数等于零，然后使用这些结果消去对应的变量，证明对偶拉格朗日函数为 (7.61)。

(7.8) (*) 对于 7.1.4 节讨论的回归支持向量机，证明，所有 $\xi_n > 0$ 的训练数据点都有 $a_n = C$ ，类似地，所有 $\hat{\xi}_n > 0$ 的数据点都有 $\hat{a}_n = C$ 。

(7.9) (*) 验证公式 (7.82) 和 (7.83) 给出的回归RVM的权值上的后验分布的均值和协方差的结果。

(7.10) (**) 推导公式 (7.85) 给出的回归RVM的边缘似然函数的结果，方法是使用对指数项配平方的方法，计算公式 (7.84) 中关于 \mathbf{w} 的高斯积分。

(7.11) (**) 重复上一个练习，但是这次使用一般的结果 (2.115)。

(7.12) (**) 证明，直接对回归相关向量机的对数似然函数 (7.85) 进行最大化会得到重估计方程 (7.87) 和 (7.88)，其中 γ_i 由 (7.89) 定义。

(7.13) (**) 在RVM回归的证据框架中，我们通过最大化边缘似然函数 (7.85)，得到了重估计方程 (7.87) 和 (7.88)。将这种方法进行扩展，将 (B.26) 给出的Gamma分布的超先验包含进去，通过关于 α 和 β 最大化对应的后验概率分布 $p(\mathbf{t}, \alpha, \beta | \mathbf{X})$ ，得到 α 和 β 的对应的重估计方程。

(7.14) (**) 推导回归的相关向量机的预测分布结果 (7.90)。证明预测方差为 (7.91)。

(7.15) (**) 使用公式 (7.94) 和 (7.95)，证明边缘似然函数 (7.85) 可以写成 (7.96) 的形式，其中 $\lambda(\alpha_n)$ 由公式 (7.97) 定义，稀疏度和质量因子分别由公式 (7.98) 和 (7.99) 定义。

(7.16) (*) 通过将回归RVM的对数边缘似然函数 (7.97) 关于超参数 α_i 取二阶导数，证明公式 (7.101) 给出的驻点是边缘似然函数的最大值。

(7.17) (**) 使用公式 (7.83) 和 (7.86)，以及矩阵恒等式 (C.7)，证明由公式 (7.102) 和 (7.103) 定义的 S_n 和 Q_n 可以写成 (7.106) 和 (7.107) 的形式。

(7.18) (*) 证明，分类相关向量机的对数后验分布 (7.109) 的梯度向量和Hessian矩阵由公式 (7.110) 和 (7.111) 给出。

(7.19) (**) 验证分类相关向量机的近似边缘似然函数 (7.114) 的最大化过程会产生公式 (7.116) 给出的超参数重估计方程的结果。

8 图模型

概率在现代模式识别中起着重要的作用。我们已经在第1章中看到了概率论可以使用两个简单的方程（加和规则和乘积规则）表示。本书中所有的概率推断以及学习操作，无论多么复杂，都是在重复使用这两个方程。因此，我们接下来将完全通过代数计算来对更加复杂的模型进行建模和求解。然而，我们会发现，使用概率分布的图形表示进行分析很有好处。这种概率分布的图形表示被称为概率图模型（probabilistic graphical models）。这些模型提供了几个有用的性质：

- 它们提供了一种简单的方式将概率模型的结构可视化，可以用于设计新的模型。
- 通过观察图形，我们可以更深刻地认识模型的性质，包括条件独立性质。
- 高级模型的推断和学习过程中的复杂计算可以根据图计算表达，图隐式地承载了背后的数学表达式。

一个图由结点（nodes）（也被称为端点（vertices））和它们之间的链接（links）（也被称为边（edges）或弧（arcs））组成。在概率图模型中，每个结点表示一个随机变量（或一组随机变量），链接表示这些变量之间的概率关系。这样，图描述了联合概率分布在所有随机变量上能够分解为一组因子的乘积的方式，每个因子只依赖于随机变量的一个子集。我们首先讨论贝叶斯网络（Bayesian network），也被称为有向图模型（directed graphical model）。这个模型中，图之间的链接有一个特定的方向，使用箭头表示。另一大类图模型是马尔科夫随机场（Markov random fields），也被称为无向图模型（undirected graphical models）。这个模型中，链接没有箭头，没有方向性质。有向图对于表达随机变量之间的因果关系很有用，而无向图对于表示随机变量之间的软限制比较有用。为了求解推断问题，通常比较方便的做法是把有向图和无向图都转化为一个不同的表示形式，被称为因子图（factor graph）。

本章中，我们会将注意力集中在那些能够用于模式识别和机器学习应用中的图模型的关键概念。关于图模型的更一般的讨论，可以参考Whittaker (1990)、Lauritzen (1996)、Jensen (1996)、Castillo et al. (1997)、Jordan (1999)、Cowell et al. (1999) 以及Jordan (2007)。

8.1 贝叶斯网络

为了理解有向图对于描述概率分布的作用，首先考虑三个变量 a, b, c 上的一个任意的联合分布 $p(a, b, c)$ 。注意，现阶段我们不需要对这些变量做出任何更多的假设，例如它们是离散的还是连续的。实际上，图模型的一个强大的方面是，一个具体的图可以描述一大类概率分布。通过使用概率的乘积规则（1.11），我们可以将联合概率分布写成下面的形式。

$$p(a, b, c) = p(c | a, b)p(a, b) \quad (8.1)$$

再次使用乘积规则，这次处理方程（8.1）右侧的第二项，我们有

$$p(a, b, c) = p(c | a, b)p(b | a)p(a) \quad (8.2)$$

注意，这个分解方法对于任意的联合概率分布的选择都成立。现在，我们使用一个简单的图模型表示方程（8.2）的右侧，如下所述。首先，我们为每个随机变量 a, b, c 引入一个结点，然后为每个结点关联上公式（8.2）右侧的对应的条件概率。然后，对于每个条件概率分布，我们在图中添加一个链接（箭头），链接的起点是条件概率的条件中的随机变量对应的结点。因此，对于因子 $p(c | a, b)$ ，会存在从结点 a, b 到结点 c 的链接，而对于因子 $p(a)$ ，没有输入的链接。结果就是图8.1中的图。如果存在一个从结点 a 到结点 b 的链接，那么我们说结点 a 是结点 b 的父结点，结点 b 是结点 a 的子结点。注意，我们不会形式化地区分结点和结点对应的变量，而是简单地使用同样的符号表示两者。

关于公式（8.2），很有趣的一点是，它的左侧关于三个变量 a, b, c 是对称的，而右侧不是。实际上，通过进行公式（8.2）的分解，我们隐式地选择了一个特定的顺序（即 a, b, c ）。如果选

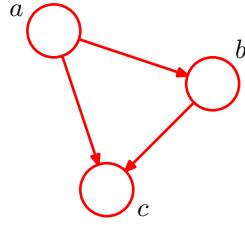


图 8.1: 一个有向图模型，表示三个变量 a, b, c 上的联合概率分布，对应于公式 (8.2) 右侧的分解。

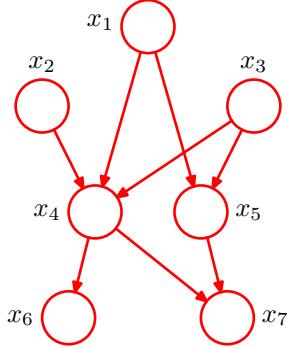


图 8.2: 有向无环图描述变量 x_1, \dots, x_7 。联合概率分布的对应的概率分解由公式 (8.4) 给出。

择一个不同的顺序，我们会得到一个不同的分解方式，因此就得到一个不同的图表示形式。我们稍后会回头讨论这个想法。

现在，让我们将图 8.1 给出的例子扩展到 K 个变量的联合概率分布 $p(x_1, \dots, x_K)$ 。通过重复使用概率的乘积规则，联合概率分布可以写成条件概率的乘积，每一项对应一个变量，形式如下

$$p(x_1, \dots, x_K) = p(x_K | x_1, \dots, x_{K-1}) \dots p(x_2 | x_1) p(x_1) \quad (8.3)$$

对应一个给定的 K ，我们可以将其表示为一个具有 K 个结点的有向图，每个结点对应于公式 (8.3) 右侧的一个条件概率分布，每个结点的输入链接包括所有以编号低于当前结点编号的结点为起点的链接。我们说，这个图是全连接的（fully connected），因为每对结点之间都存在一个链接。

目前为止，我们操作的对象是一个完全一般的联合概率分布，从而分解方式以及对应的全连接图表示，可以应用于概率分布的任意选择。正如我们将会看到的，真正传递出图表示的概率分布的性质的有趣信息的是图中链接的缺失（absence）。考虑图 8.2 的图。这不是一个全连接的图，因为从 x_1 到 x_2 或者从 x_3 到 x_7 之间不存在链接。

现在，我们将根据这幅图，写出对应的联合概率表达式。联合概率表达式由一系列条件概率的乘积组成，每一项对应于图中的一个结点。每个这样的条件概率分布只以图中对应结点的父结点为条件。例如， x_5 以 x_1 和 x_3 为条件。于是，7 个变量的联合概率分布为

$$p(x_1)p(x_2)p(x_3)p(x_4 | x_1, x_2, x_3)p(x_5 | x_1, x_3)p(x_6 | x_4)p(x_7 | x_4, x_5) \quad (8.4)$$

读者现阶段应该仔细研究公式 (8.4) 与图 8.2 之间的对应关系。

我们现在说明给定的有向图和变量上对应的概率分布之间的一般关系。在图的所有结点上定义的联合概率分布由每个结点上的条件概率分布的乘积表示，每个条件概率分布的条件都是图中结点的父结点所对应的变量。因此，对于一个有 K 个结点的图，联合概率为

$$p(\mathbf{x}) = \prod_{k=1}^K p(x_k | \text{pa}_k) \quad (8.5)$$

其中， pa_k 表示 x_k 的父结点的集合， $\mathbf{x} = \{x_1, \dots, x_K\}$ 。这个关键的方程表示有向图模型的联合概率分布的分解（factorization）属性。虽然我们之前考虑的情况是每个结点对应于一个变量的情形，但是我们可以很容易地推广到让图的每个结点关联一个变量的集合，或者关联向量值的

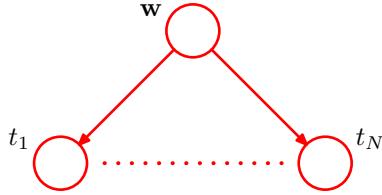


图 8.3: 有向图模型表示联合概率分布 (8.6) , 对应于1.2.6节介绍的贝叶斯多项式回归模型。

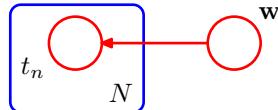


图 8.4: 一种更加简洁的方式表示图8.3中的图, 其中我们引入了一个板 (plate) (标记为 N 的方框) 来表示 N 个结点, 这些结点中, 只有一个例子 t_n 被显式地画出。

变量。很容易证明, 如果公式 (8.5) 右侧的每一个条件概率分布都是归一化的, 那么这个表示方法整体总是归一化的。

我们考虑的有向图要满足一个重要的限制, 即不能存在有向环 (directed cycle)。换句话说, 在图中不能存在这样的路径: 从某个结点开始, 沿着链接中箭头的方向运动, 结束点为起点。这种没有有向环的图被称为有向无环图 (directed acyclic graph), 或者DAG。这等价于存在一个将该点的排序, 使得不存在从某个结点到序号较小的结点的链接。

8.1.1 例子: 多项式回归

作为有向图描述概率分布的一个例子, 我们考虑1.2.6节介绍的贝叶斯多项式拟合模型。这个模型中的随机变量是多项式系数向量 w 和观测数据 $\mathbf{t} = (t_1, \dots, t_N)^T$ 。此外, 这个模型包含输入数据 $\mathbf{x} = (x_1, \dots, x_N)^T$ 、噪声方差 σ^2 以及表示 w 的高斯先验分布的精度的超参数 α 。所有这些都是模型的参数而不是随机变量。现阶段我们只关注随机变量, 我们看到联合概率分布等于先验概率分布 $p(w)$ 与 N 个条件概率分布 $p(t_n | w)$ 的乘积 ($n = 1, \dots, N$) , 即

$$p(\mathbf{t}, w) = p(w) \prod_{n=1}^N p(t_n | w) \quad (8.6)$$

图模型表示的联合概率分布如图8.3所示。

在本书的后面章节中, 当我们开始处理更加复杂的模型时, 我们会看到, 像图8.3那样显式地写出 t_1, \dots, t_N 的结点是很不方便的。于是, 我们引入一种图结构, 使得多个结点可以更简洁地表示出来。这种图结构中, 我们画出一个单一表示的结点 t_n , 然后用一个被称为板 (plate) 的方框圈起来, 标记为 N , 表示有 N 个同类型的点。用这种方式重新表示图8.3, 我们得到了图8.4所示的图。

我们有时会发现, 显式地写出模型的参数和随机变量是很有帮助的。此时, 公式 (8.6) 就变成了

$$p(\mathbf{t}, w | \mathbf{x}, \alpha, \sigma^2) = p(w | \alpha) \prod_{n=1}^N p(t_n | w, x_n, \sigma^2)$$

对应地, 我们可以在图表示中显式地写出 \mathbf{x} 和 α 。为了这样做, 我们会遵循下面的惯例: 随机变量由空心圆表示, 确定性参数由小的实心圆表示。如果我们让图8.4包含确定性参数, 我们就得到了图8.5。

当我们将图模型应用于机器学习或者模式识别的问题中时, 我们通常将某些随机变量设置为具体的值, 例如将变量 $\{t_n\}$ 根据多项式曲线拟合中的训练集进行设置。在图模型中, 我们通过给对应的结点加上阴影的方式来表示这种观测变量 (observed variables)。于是, 图8.5所示的图中, 如果 $\{t_n\}$ 是观测变量, 那么就变成了图8.6。注意, w 不是观测变量, 因此 w 是潜在变量 (latent variable) 的一个例子。潜在变量也被称为隐含变量 (hidden variable)。这样的变量在许多概率模型中有着重要的作用, 将在第9章和第12章详细讨论。

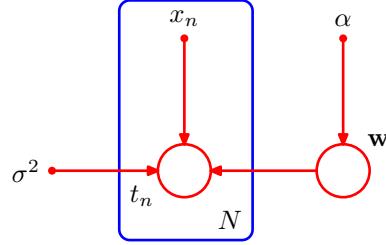


图 8.5: 本图给出了与图8.4相同的模型，但是显式地画出了确定性参数，用小的实心圆表示。

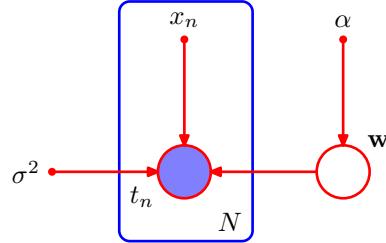


图 8.6: 与图8.5相同，但是结点 $\{t_n\}$ 被标记为阴影，表示对应的随机变量被设置成它们在训练集里的观测值。

观测到了 $\{t_n\}$ 的值，如果必要的话，我们可以计算系数 w 的后验概率，如1.2.5节讨论的那样。现阶段，我们注意到，这是贝叶斯定理的一个直接应用。

$$p(w | \mathbf{t}) \propto p(w) \prod_{n=1}^N p(t_n | w) \quad (8.7)$$

其中，我们再一次省略了确定性参数，使得记号简洁。

通常，我们对于 w 这样的参数本身不感兴趣，因为我们的最终目标是对输入变量进行预测。假设给定一个输入值 \hat{x} ，我们想找到以观测数据为条件的对应的 \hat{t} 的概率分布。描述这个问题的图模型如图8.7所示。以确定性参数为条件，这个模型的所有随机变量的联合分布为

$$p(\hat{t}, \mathbf{t}, w | \hat{x}, \mathbf{x}, \alpha, \sigma^2) = \left[\prod_{n=1}^N p(t_n | x_n, w, \sigma^2) \right] p(w | \alpha) p(\hat{t} | \hat{x}, w, \sigma^2) \quad (8.8)$$

然后，根据概率的加和规则，对模型参数 w 积分，即可得到 \hat{t} 的预测分布

$$p(\hat{t} | \hat{x}, \mathbf{x}, \mathbf{t}, \alpha, \sigma^2) \propto \int p(\hat{t}, \mathbf{t}, w | \hat{x}, \mathbf{x}, \alpha, \sigma^2) dw$$

其中我们隐式地将 \mathbf{t} 中的随机变量设置为数据集中观测到的具体值。计算的细节已经在第3章中讨论过。

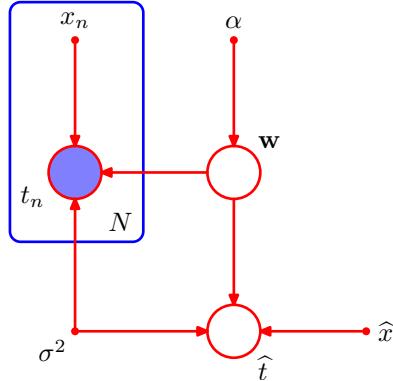


图 8.7: 多项式回归模型，对应于图8.6。同时画出了一个新的输入值 \hat{x} 以及对应的模型精度 \hat{t} 。

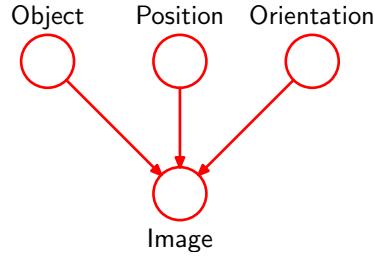


图 8.8: 一个图模型，表示物体的图像的创建过程。其中，物体的种类（一个离散变量）以及物体的位置和方向（连续变量）具有独立的先验概率。图像（一个像素灰度值的向量）的概率分布与物体的种类以及它的位置和方向无关。

8.1.2 生成式模型

许多情况下，我们希望从给定的概率分布中抽取样本。虽然我们将在第11章用整章的篇幅讨论取样方法，但是这里简要介绍一种方法是很有意义的。这种方法被称为祖先取样（ancestral sampling），与图模型特别相关。考虑 K 个变量的一个联合概率分布 $p(x_1, \dots, x_K)$ ，它根据公式(8.5)进行分解，对应于一个有向无环图。我们假设变量已经进行了排序，从而不存在从某个结点到序号较低的结点的链接。换句话说，每个结点的序号都大于它的父结点。我们的目标是从这样的联合概率分布中取样 $\hat{x}_1, \dots, \hat{x}_K$ 。

为了完成这一点，我们首先选出序号最小的结点，按照概率分布 $p(x_1)$ 取样，记作 \hat{x}_1 。然后，我们顺序计算每个结点，使得对于结点 n ，我们根据条件概率 $p(x_n | \text{pa}_n)$ 进行取样，其中父结点的变量被设置为它们的取样值。注意，在每个阶段，这些父结点的变量总是可以得到的，因为它们对应于已经采样过的序号较小的结点。按照具体的概率分布的取样方法将会在第11章详细讨论。一旦我们对最后的变量 x_K 取样结束，我们就达到了根据联合概率分布取样的目标。为了从对应于变量的子集的边缘概率分布中取样，我们简单地取要求结点的取样值，忽略剩余结点的取样值。例如，为了从概率分布 $p(x_2, x_4)$ 中取样，我们简单地对联合概率分布取样，然后保留 \hat{x}_2, \hat{x}_4 ，丢弃剩余的值 $\{\hat{x}_{j \neq 2, 4}\}$ 。

对于概率模型的实际应用，通常的情况是，数量众多的变量对应于图的终端结点（表示观测值），较少的变量对应于潜在变量。潜在变量的主要作用是使得观测变量上的复杂分布可以表示为由简单条件分布（通常是指数族分布）构建的模型。

我们可以将这样的模型表示为观测数据产生的过程。例如，考虑一个模式识别的任务，其中每个观测值对应于一幅图像（由像素灰度值的向量组成）。这种情况下，潜在变量可以看成物体的位置或者方向。给定一个特定的观测图像，我们的目标是找到物体上的后验概率分布，其中我们对于所有可能的位置和方向进行了积分。我们可以使用图8.8的图模型表示这个问题。

图模型描述了生成观测数据的一种因果关系（causal）过程（Pearl, 1988）。因此，这种模型通常被称为生成式模型（generative model）。相反，图8.5描述的多项式回归模型不是生成式模型，因为没有与输入变量 x 相关联的概率分布，因此无法从这个模型中人工生成数据点。通过引入合适的先验概率分布 $p(x)$ ，我们可以将模型变为生成式模型，代价是增加了模型的复杂度。

然而，概率模型中的隐含变量不必具有显式的物理含义。它的引入可以仅仅为了从更简单的成分中建立一个更复杂的联合概率分布。在任何一种情况下，应用于生成式模型的祖先取样方法都模拟了观测数据的创造过程，因此可以产生“幻想”的数据，它的概率分布（如果模型完美地表示现实）与观测数据的概率分布相同。在实际应用中，从一个生成式模型中产生人工生成的观测数据，对于理解模型所表示的概率分布形式很有帮助。

8.1.3 离散变量

我们已经讨论了指数族概率分布的重要性，我们看到这一类概率分布将许多著名的概率分布当成了指数族分布的特例。虽然指数族分布相对比较简单，但是它们组成了构建更复杂概率分布的基本元件。图模型的框架在表达这些基本元件之间的联系时非常有用。

如果我们将有向图中的每个父结点-子结点对的关系选为共轭的，那么这样的模型有一些特别好的性质，我们稍后会给出几个例子。两种情形很值得注意，即父结点和子结点都对应于离

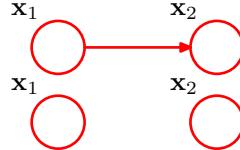


图 8.9: (a)全连接的图描述了两个 K 状态离散变量上的一般的分布，具有 $K^2 - 1$ 个参数。(b)通过丢弃结点之间的链接，参数的数量减小到了 $2(K - 1)$ 。

散变量的情形，以及它们都对应高斯变量的情形，因为在这两种情形中，关系可以层次化地推广，构建任意复杂的有向无环图。我们首先考察离散变量的情形。

对于有着 K 个可能状态（使用“1-of- K ”表达方式）的一元离散变量 x ，概率 $p(x | \mu)$ 为

$$p(x | \mu) = \prod_{k=1}^K \mu_k^{x_k} \quad (8.9)$$

并且由参数 $\mu = (\mu_1, \dots, \mu_K)^T$ 控制。由于限制条件 $\sum_k \mu_k = 1$ 的存在，因此为了定义概率分布，只需要指定 $K - 1$ 个 μ_k 的值即可。

现在假设我们有两个离散变量 x_1 和 x_2 ，每个都有 K 个状态，我们项对它们的联合概率分布建模。我们将 $x_{1k} = 1$ 和 $x_{2l} = 1$ 同时被观测到的概率记作参数 μ_{kl} ，其中 x_{1k} 表示 x_1 的第 k 个分量， x_{2l} 的意义与此相似。联合概率分布可以写成

$$p(x_1, x_2 | \mu) = \prod_{k=1}^K \prod_{l=1}^K \mu_{kl}^{x_{1k} x_{2l}}$$

由于参数 μ_{kl} 满足限制条件 $\sum_k \sum_l \mu_{kl} = 1$ ，因此这个分布由 $K^2 - 1$ 个参数控制。很容易看到，对于 M 个变量的任意一个联合概率分布，需要确定的参数的数量为 $K^M - 1$ ，因此随着变量 M 的数量指数增长。

使用概率的乘积规则，我们可以将联合概率分布 $p(x_1, x_2)$ 分解为 $p(x_2 | x_1)p(x_1)$ ，它对应于一个具有两个结点的图，链接从结点 x_1 指向结点 x_2 ，如图8.9(a)所示。边缘概率分布 $p(x_1)$ 与之前一样，由 $K - 1$ 个参数控制。类似地，条件概率分布 $p(x_2 | x_1)$ 需要指定 $K - 1$ 个参数，确定 x_1 的 K 个可能的取值。因此，与之前一样，在联合概率分布中，需要指定的参数的总数为 $(K - 1) + K(K - 1) = K^2 - 1$ 。

现在假设变量 x_1 和 x_2 是独立的，对应于图8.9(b)所示的图模型。这样，每个变量由一个独立的多项式概率分布描述，参数的总数是 $2(K - 1)$ 。对于 M 个独立离散变量上的概率分布，其中每个变量有 K 个可能的状态，参数的总数为 $M(K - 1)$ ，因此随着变量的数量线性增长。从图的角度看，我们通过删除结点之间链接的方式，减小了参数的数量，代价是类别的概率分布受到了限制。

更一般地，如果我们有 M 个离散变量 x_1, \dots, x_M ，那么我们可以使用有向图来对联合概率分布建模，每个变量对应于一个结点。每个结点的条件概率分布由一组非负参数给出，同时需要满足归一化限制条件。如果图是全连接的，那么我们有一个完全一般的概率分布，这个分布有 $K^M - 1$ 个参数。而如果图中不存在链接，那么联合概率分布可以分解为边缘概率分布的乘积，参数的总数为 $M(K - 1)$ 。连接度处于二者之间的图使得模型能够处理比完全分解的概率分布更加一般的概率分布，同时参数的数量比一般的联合概率分布的参数数量少。作为一个说明，考虑图8.10所示的结点链。边缘概率分布 $p(x_1)$ 需要 $K - 1$ 个参数，而对于 $M - 1$ 个条件概率分布 $p(x_i | x_{i-1})$ （其中 $i = 2, \dots, M$ ）需要 $K(K - 1)$ 个参数。从而，参数的总数为 $K - 1 + (M - 1)K(K - 1)$ ，这是 K 的二次函数，并且随着链的长度 M 线性增长（而不是指数增长）。

另一种减小模型中独立参数数量的方法是参数共享（sharing），也被称为参数捆扎（tying）。例如，在图8.10给出的结点链的例子，我们可以使所有的条件概率分布 $p(x_i | x_{i-1})$ （其中 $i = 2, \dots, M$ ）由同样的参数集合 $K(K - 1)$ 。加上控制 x_1 的 $K - 1$ 个参数，为了定义联合概率分布所需指定的参数的总数为 $K^2 - 1$ 。



图 8.10: M 个离散结点组成的链，每个结点有 K 个状态，要求指定 $K - 1 + (M - 1)K(K - 1)$ 个参数，它随着链的长度 M 线性增长。相反， M 个结点的一个完全连接的图具有 $K^M - 1$ 个参数，它随着 M 指数增长。

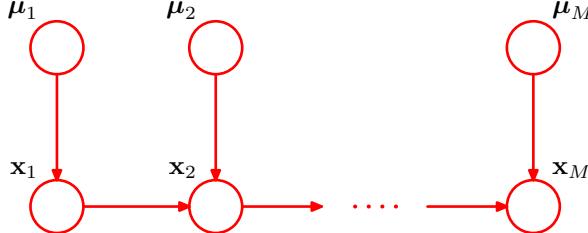


图 8.11: 图 8.10 的模型的扩展，包含了控制离散分布的参数的狄利克雷先验分布。

通过引入参数的狄利克雷先验，我们可以将离散变量上的图模型转化为贝叶斯模型。从图的观点来看，每个结点需要额外的父结点表示对应于每个离散结点的参数。这种情况在图 8.11 中进行了说明。如果我们将控制条件概率分布 $p(x_i | x_{i-1})$ （其中 $i = 2, \dots, M$ ）的参数进行参数共享，那么对应的模型如图 8.12 所示。

另一种控制离散变量模型参数数量的指数增长的方式是对条件概率分布使用参数化的模型，而不使用条件概率值的完整表格。为了说明这个想法，考虑图 8.13 所示的图，其中所有的结点表示二值变量。每个父结点变量 x_i 由单一参数 μ_i 控制，这个参数表示概率 $p(x_i = 1)$ ，从而对于 M 个父结点，参数总数为 M 。但是，条件概率分布 $p(x_1, \dots, x_M)$ 需要 2^M 个参数，每个参数表示 2^M 种父结点变量的可能配置下的概率 $p(y = 1)$ 。因此，通常来说，确定这个条件概率分布的参数的数量会随着 M 指数增长。将 logistic sigmoid 函数作用于父结点变量的线性组合上，我们可以得到一个更加简洁的条件概率分布，形式为

$$p(y = 1 | x_1, \dots, x_M) = \sigma\left(w_0 + \sum_{i=1}^M w_i x_i\right) = \sigma(\mathbf{w}^T \mathbf{x}) \quad (8.10)$$

其中 $\sigma(a) = (1 + \exp(-a))^{-1}$ 是一个 logistic sigmoid 函数， $\mathbf{x} = (x_0, x_1, \dots, x_M)^T$ 是一个 $(M + 1)$ 维向量，表示父结点的 M 个状态加上一个额外的变量 x_0 ，其值被固定为 1。 $\mathbf{w} = (w_0, w_1, \dots, w_M)^T$ 是一个 $M + 1$ 个参数的向量。与一般的情形相比，这是一个更加受限形式的条件概率分布，但是参数的数量随着 M 线性增长。在这种情况下，类似于选择多元高斯分布的协方差矩阵的限制形式（例如对角矩阵）。采用 logistic sigmoid 表示方法的原因在 4.2 节已经讨论过。

8.1.4 线性高斯模型

在前一节中，我们看到了如何在一组离散变量上构建联合概率分布，构建方法是将变量表示为有向无环图上的结点。这里，我们将说明多元高斯分布如何表示为一个对应于成分变量上的线性高斯模型的有向无环图。这使得我们在概率分布上施加有趣的结构，这些结构中的两个相

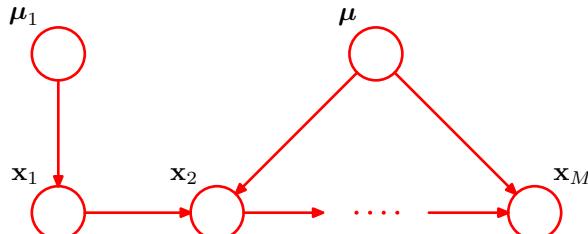


图 8.12: 与图 8.11 相同，但是所有的条件概率分布 $p(x_i | x_{i-1})$ 共享一个单一的参数 μ 的集合。

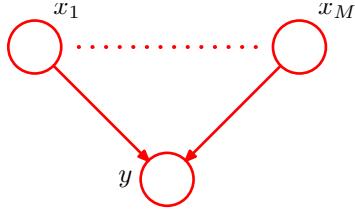


图 8.13: 一个由 M 个父结点 x_1, \dots, x_M 和一个单一子结点 y 组成的图，用来说说明离散变量的参数化条件概率分布的思想。

反的极端情况是一般的高斯分布和对角化协方差高斯分布。几种广泛使用的方法是线性高斯模型的例子，例如概率主成分分析，因子分析，以及线性动态系统（Roweis and Ghahramani, 1999）。在后续章节中，当我们详细讨论一些方法时，我们会频繁使用本节的结果。

考虑 D 个变量上的任意的有向无环图，其中结点 i 表示服从高斯分布的一元连续随机变量 x_i 。这个分布的均值是结点 i 的父结点 pa_i 的状态的线性组合，即

$$p(x_i | \text{pa}_i) = \mathcal{N} \left(x_i \middle| \sum_{j \in \text{pa}_i} w_{ij} x_j + b_i, v_i \right) \quad (8.11)$$

其中 w_{ij} 和 b_i 是控制均值的参数， v_i 是 x_i 的条件概率分布的方差。这样，联合概率分布的对数为图中所有结点上的这些条件分布的乘积的对数，因此形式为

$$\ln p(\mathbf{x}) = \sum_{i=1}^D \ln p(x_i | \text{pa}_i) \quad (8.12)$$

$$= -\sum_{i=1}^D \frac{1}{2v_i} \left(x_i - \sum_{j \in \text{pa}_i} w_{ij} x_j - b_i \right)^2 + \text{常数} \quad (8.13)$$

其中 $\mathbf{x} = (x_1, \dots, x_D)^T$ ，“常数”表示与 \mathbf{x} 无关的项。我们看到这是 \mathbf{x} 的元素的二次函数，因此联合概率分布 $p(\mathbf{x})$ 是一个多元高斯分布。

我们可以递归地确定联合概率分布的均值和方差，方法如下。每个变量 x_i 的概率分布都是（以父结点状态为条件的）高斯分布，形式为公式 (8.11) 所示。因此

$$x_i = \sum_{j \in \text{pa}_i} w_{ij} x_j + b_i + \sqrt{v_i} \epsilon_i \quad (8.14)$$

其中 ϵ_i 是一个零均值单位方差的高斯随机变量，满足 $\mathbb{E}[\epsilon_i] = 0$ 且 $\mathbb{E}[\epsilon_i \epsilon_j] = I_{ij}$ ，其中 I_{ij} 是单位矩阵的第 i, j 个元素。对公式 (8.14) 取期望，我们有

$$\mathbb{E}[x_i] = \sum_{j \in \text{pa}_i} w_{ij} \mathbb{E}[x_j] + b_i \quad (8.15)$$

这样，从一个序号最低的结点开始，沿着图递归地计算，我们就可以求出 $\mathbb{E}[\mathbf{x}] = (\mathbb{E}[x_1], \dots, \mathbb{E}[x_D])^T$ 的各个元素。这里，我们再一次假设所有结点的序号都大于它的父结点的序号。类似地，我们可以使用公式 (8.14) 和 (8.15)，以递归的方式得到 $p(\mathbf{x})$ 的协方差矩阵的第 i, j 个元素，即

$$\begin{aligned} \text{cov}[x_i, x_j] &= \mathbb{E}[(x_i - \mathbb{E}[x_i])(x_j - \mathbb{E}[x_j])] \\ &= \mathbb{E} \left[(x_i - \mathbb{E}[x_i]) \left\{ \sum_{k \in \text{pa}_j} w_{jk} (x_k - \mathbb{E}[x_k]) + \sqrt{v_j} \epsilon_j \right\} \right] \\ &= \sum_{k \in \text{pa}_j} w_{jk} \text{cov}[x_i, x_k] + I_{ij} v_j \end{aligned} \quad (8.16)$$

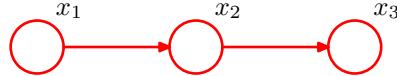


图 8.14: 三个高斯变量上的有向图, 有一个链接缺失。

因此协方差可以从序号最低的结点开始, 递归地计算。

让我们考虑两个极端的情形。首先, 假设图中不存在链接, 因此图由 D 个孤立的结点组成。在这种情况下, 不存在参数 w_{ij} , 因此只有 D 个参数 b_i 和 D 个参数 v_i 。根据递归关系 (8.15) 和 (8.16), 我们看到 $p(\mathbf{x})$ 的均值为 $(b_1, \dots, b_D)^T$, 协方差矩阵是一个对角矩阵, 形式为 $\text{diag}(v_1, \dots, v_D)$ 。联合概率分布总计有 $2D$ 个参数, 表示 D 个独立的一元高斯分布组成的集合。

现在考虑一个全连接的图, 其中每个结点的序号都低于其父结点的序号。这样矩阵 w_{ij} 的第 i 行有 $i - 1$ 项, 因此矩阵是一个下三角矩阵 (主对角线上没有元素)。参数 w_{ij} 的数量从而可以通过下面的方式得到: 取 $D \times D$ 的元素个数 D^2 , 减去 D , 表示主对角线上没有元素, 再除以2, 因为矩阵只在对角线下方存在元素, 从而参数的总数为 $\frac{D(D-1)}{2}$ 。独立参数 $\{w_{ij}\}$ 加上协方差矩阵中的 $\{v_i\}$, 因此独立参数的总数为 $\frac{D(D+1)}{2}$, 对应于一个一般的对称协方差矩阵。

复杂度处于两种极端情况之间的图对应于协方差矩阵取特定形式的联合高斯分布。考虑图 8.14 中的图, 它在变量 x_1 和 x_3 之间不存在链接。使用递归关系 (8.15) 和 (8.16), 我们看到联合高斯分布的均值和协方差为

$$\boldsymbol{\mu} = (b_1, b_2 + w_{21}b_1, b_3 + w_{32}b_2 + w_{32}w_{21}b_1)^T \quad (8.17)$$

$$\boldsymbol{\Sigma} = \begin{pmatrix} v_1 & w_{21}v_1 & w_{32}w_{21}v_1 \\ w_{21}v_1 & v_2 + w_{21}^2v_1 & w_{32}(v_2 + w_{21}^2v_1) \\ w_{32}w_{21}v_1 & w_{32}(v_2 + w_{21}^2v_1) & v_3 + w_{32}^2(v_2 + w_{21}^2v_1) \end{pmatrix} \quad (8.18)$$

我们已经可以将线性高斯图模型扩展到结点表示多元高斯变量的情形。在这种情况下, 我们可以将结点 i 的条件概率分布写成下面的形式

$$p(\mathbf{x}_i | \text{pa}_i) = \mathcal{N} \left(\mathbf{x}_i \mid \sum_{j \in \text{pa}_i} \mathbf{W}_{ij} \mathbf{x}_j + \mathbf{b}_i, \boldsymbol{\Sigma}_i \right) \quad (8.19)$$

现在 \mathbf{W}_{ij} 是一个矩阵。如果 \mathbf{x}_i 和 \mathbf{x}_j 的维度不同, 那么 \mathbf{W}_{ij} 不是方阵。与之前一样, 很容易证明所有变量上的联合概率分布是高斯分布。

注意, 我们已经看到高斯变量 \mathbf{x} 的均值 $\boldsymbol{\mu}$ 的共轭先验本身是 $\boldsymbol{\mu}$ 上的一个高斯分布。此时我们已经遇到了线性高斯关系的一个具体的例子。因此 \mathbf{x} 和 $\boldsymbol{\mu}$ 的联合分布就是高斯分布。这对应于一个简单的具有两个结点的图, 其中表示 $\boldsymbol{\mu}$ 和结点 i 是表示 \mathbf{x} 的结点的父结点。 $\boldsymbol{\mu}$ 上的概率分布的均值是控制先验分布的参数, 因此它可以被看做超参数。由于超参数的值本身是未知的, 因此我们可以再一次使用贝叶斯的观点, 引入一个超参数上的先验概率分布。这个先验概率分布有时被称为超先验 (hyperprior), 它还是一个高斯分布。这种构造过程原则上可以延伸到任意层次。这个模型是层次贝叶斯模型 (hierarchical Bayesian model) 的一个例子, 我们会在后续章节中遇到这个模型的更多例子。

8.2 条件独立

多变量概率分布的一个重要概念是条件独立 (conditional independence) (Dawid, 1980)。考虑三个变量 a, b, c , 并且假设给定 b, c 的条件下 a 的条件概率分布不依赖于 b 的值, 即

$$p(a | b, c) = p(a | c) \quad (8.20)$$

我们说, 给定 c 的条件下, a 条件独立于 b 。如果我们考虑以 c 为条件下的 a, b 的联合分布, 我们可以用一种稍微不同的方式表示, 即

$$\begin{aligned} p(a, b | c) &= p(a | b, c)p(b | c) \\ &= p(a | c)p(b | c) \end{aligned} \quad (8.21)$$

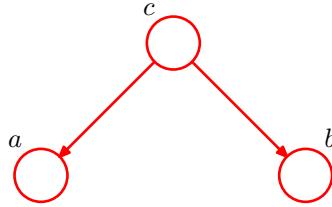


图 8.15: 三个变量 a, b, c 上的图模型的三个例子中的第一个, 这些例子用来讨论有向图模型的条件独立性质。

其中我们使用了概率的乘积规则以及公式 (8.20)。因此, 我们看到了, 以 c 为条件, a 和 b 的联合概率分布分解为了 a 的边缘概率分布和 b 的边缘概率分布的乘积 (全部以 c 为条件)。注意, 我们对于独立性的定义需要公式 (8.20) 对于 c 的所有可能值成立, 或者等价地需要公式 (8.21) 对于 c 的所有可能值成立, 而不是对于某些特定的 c 值。我们有时会使用条件独立的一种简洁记号 (Dawid, 1979), 即

$$a \perp\!\!\!\perp b \mid c \quad (8.22)$$

表示给定 c 的条件下 a 与 b 条件独立, 等价于公式 (8.20)。

模式识别中, 使用概率模型时, 条件独立性起着重要的作用。它简化了模型的结构, 降低了模型的训练和推断的计算量。我们稍后会看到这样的例子。

如果一组变量的联合概率分布的表达式是根据条件概率分布的乘积表示的 (即有向图的数学表达形式), 那么原则上我们可以通过重复使用概率的加和规则和乘积规则测试是否具有潜在的条件独立性。在实际应用中, 这种方法非常耗时。图模型的一个重要的优雅的特征是, 联合概率分布的条件独立性可以直接从图中读出来, 不用进行任何计算。完成这件事的一般框架被称为“d-划分” (d-separation), 其中“d”表示“有向 (directed)” (Pearl, 1988)。这里, 我们非形式化地介绍了 d-划分的概念, 给出了 d-划分准则的一个一般叙述。形式化的证明可以参考 Lauritzen (1996)。

8.2.1 图的三个例子

我们开始讨论有向图的条件独立性质。考虑三个简单的例子, 每个例子涉及到只有三个结点的图。这些例子会说明 d-划分中的关键概念。三个例子中的第一个如图 8.15 所示。使用公式 (8.5) 给出的一般结果, 对于这个图的联合概率分布很容易写出来, 即

$$p(a, b, c) = p(a \mid c)p(b \mid c)p(c) \quad (8.23)$$

如果没有变量是观测变量, 那么我们通过对公式 (8.23) 两边进行积分或求和的方式, 考察 a 和 b 是否是相互独立的, 即

$$p(a, b) = \sum_c p(a \mid c)p(b \mid c)p(c) \quad (8.24)$$

一般地, 这不能分解为乘积 $p(a)p(b)$, 因此

$$a \not\perp\!\!\!\perp b \mid \emptyset \quad (8.25)$$

其中, \emptyset 表示空集, 符号 $\not\perp\!\!\!\perp$ 表示条件独立性质不总是成立。当然, 通过给各个概率分布关联具体的数值, 可能存在一个特定的分布使得条件独立的性质成立, 但是一般情形下, 不能构建图结构。

现在假设我们以变量 c 为条件, 如图 8.16 所示。根据公式 (8.23), 我们可以很容易地写出给定 c 的条件下, a 和 b 的条件概率分布, 形式为

$$\begin{aligned} p(a, b \mid c) &= \frac{p(a, b, c)}{p(c)} \\ &= p(a \mid c)p(b \mid c) \end{aligned}$$

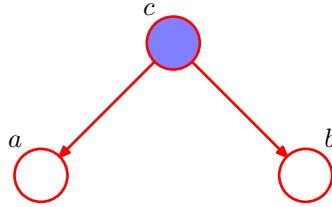


图 8.16: 与图8.15相同，但是我们以变量 c 为条件。

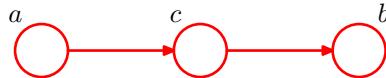


图 8.17: 3结点图的三个例子中的第二个，这些例子用来说明有向图模型的条件独立框架。

因此我们可以得到条件独立性质

$$a \perp\!\!\!\perp b \mid c$$

通过考虑从结点 a 经过结点 c 到结点 b ，我们可以给这个结果一个简单的图表示。结点 c 被称为关于这个路径“尾到尾”(tail-to-tail)，因为结点与两个箭头的尾部相连。这样的一个连接结点 a 和结点 b 的路径的存在使得结点相互依赖。然而，当我们以结点 c 为条件时（如图8.16所示），被用作条件的结点“阻隔”了从 a 到 b 的路径，使得 a 和 b 变得（条件）独立了。

我们可以类似地考虑图8.17给出的图。对应于这幅图的联合概率分布可以通过一般形式的公式 (8.5) 得到，形式为

$$p(a, b, c) = p(a)p(c \mid a)p(b \mid c) \quad (8.26)$$

首先，假设所有的变量都不是观测变量。与之前一样，我们可以考察 a 和 b 是否是相互独立的，方法是对 c 积分或求和，结果为

$$p(a, b) = p(a) \sum_c p(c \mid a)p(b \mid c) = p(a)p(b \mid a)$$

这通常不能够分解为 $p(a)p(b)$ ，因此

$$a \not\perp\!\!\!\perp b \mid \emptyset \quad (8.27)$$

这个结果与之前的结果相同。

现在假设我们以结点 c 为条件，如图8.18所示。使用贝叶斯定理，以及公式 (8.26)，我们有

$$\begin{aligned} p(a, b \mid c) &= \frac{p(a, b, c)}{p(c)} \\ &= \frac{p(a)p(c \mid a)p(b \mid c)}{p(c)} \\ &= p(a \mid c)p(b \mid c) \end{aligned}$$

从而我们又一次得到了条件独立性质

$$a \perp\!\!\!\perp b \mid c$$

与之前一样，我们可以用图表示这个结果。结点 c 被称为关于从结点 a 到结点 b 的路径“头到尾”(head-to-tail)。这样的一个路径连接了结点 a 和结点 b ，并且使它们互相之间存在依赖关系。如果我们现在观测结点 c ，如图8.18所示，那么这个观测“阻隔”了从 a 到 b 的路径，因此我们得到了条件独立性质 $a \perp\!\!\!\perp b \mid c$ 。

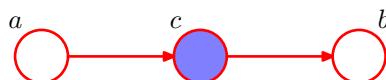


图 8.18: 与图8.17相同，但是现在以 c 为条件。

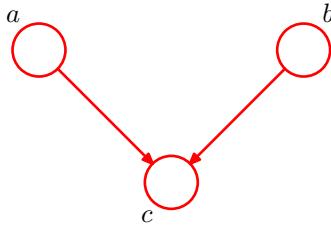


图 8.19: 3结点图的三个例子中的最后一个例子，这些例子用来研究图模型中的条件独立性质。这张图与前两个例子的性质相当不同。

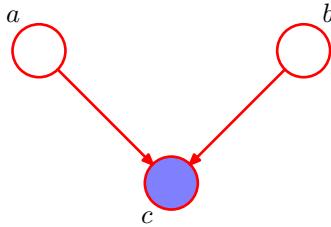


图 8.20: 与图 8.19 相同，但是以结点c的值为条件。这张图中，引入条件结点使得a和b之间产生了依赖关系。

最后，我们考虑第三个3结点图的例子，如图8.19所示。正如我们将看到的那样，这幅图的行为比之前两幅图更微妙。

与之前一样，联合概率分布可以使用我们的一般结果（8.5）得到。

$$p(a, b, c) = p(a)p(b)p(c | a, b) \quad (8.28)$$

首先考虑当没有变量是观测变量时的情形。对公式（8.28）两侧关于c积分或求和，我们有

$$p(a, b) = p(a)p(b)$$

因此当没有变量被观测时，a和b是独立的，这与前两个例子相反。我们可以把这个结果写成

$$a \perp\!\!\!\perp b | \emptyset \quad (8.29)$$

现在假设我们以c为条件，如图8.20所示。a和b的条件概率分布为

$$\begin{aligned} p(a, b | c) &= \frac{p(a, b, c)}{p(c)} \\ &= \frac{p(a | c)p(b | c)p(c | a, b)}{p(c)} \end{aligned}$$

这通常无法被分解为乘积 $p(a)p(b)$ ，因此

$$a \not\perp\!\!\!\perp b | c$$

因此，我们第三个例子与前两个例子的行为相反。图形上，我们说结点c关于从a到b的路径是“头到头”（head-to-head），因为它连接了两个箭头的头。当结点c没有被观测到的时候，它“阻隔”了路径，从而变量a和b是独立的。然而，以c为条件时，路径被“解除阻隔”，使得a和b相互依赖了。

第三个例子还有一个更加微妙的地方需要考虑。首先，我们引入一些新的概念。如果存在从结点x到结点y的一条路径，其中路径的每一步都沿着箭头的方向，那么我们说结点y是结点x的后继（descendant）。这样，可以证明，在一个头到头的路径中，如果任意结点或者它的任意一个后继被观测到，那么路径会被“解除阻隔”。

总之，一个尾到尾结点或者头到尾结点使得一条路径没有阻隔，除非它被观测到，之后它就阻隔了路径。相反，一个头到头结点如果没有被观测到，那么它阻隔了路径，但是一旦这个结点或者至少一个后继被观测到，那么路径就被“解除阻隔”了。

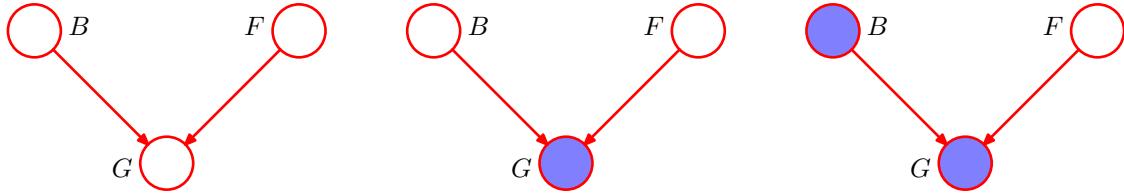


图 8.21: 一个3结点图的例子，用来说明“辩解”的现象。三个结点表示电池的状态 (B)、油箱的状态 (F) 以及油量计读数的状态 (G)。详细说明见正文。

花一些时间进一步理解图8.20给出的图的不寻常的行为是很有意义的。考虑一个特定的实例，即这个图对应于下面的问题：问题中有三个二值随机变量，这些变量与汽车的燃料系统相关，如图8.21所示。变量 B 表示电池的状态是充电过 ($B = 1$) 还是没充电 ($B = 0$)，变量 F 表示油箱是满的 ($F = 1$) 还是空的 ($F = 0$)，变量 G 表示电子油量测量计给出的读数是满的 ($G = 1$) 还是空的 ($G = 0$)。电池要么充电过，要么没充电。与此独立，油箱要么是满的，要么是空的。二者的先验概率为

$$p(B = 1) = 0.9$$

$$p(F = 1) = 0.9$$

给定油箱和电池的状态，油量计给出“满的”读数的概率为

$$p(G = 1 | B = 1, F = 1) = 0.8$$

$$p(G = 1 | B = 1, F = 0) = 0.2$$

$$p(G = 1 | B = 0, F = 1) = 0.2$$

$$p(G = 1 | B = 0, F = 0) = 0.1$$

所以这是一个相当不可靠的油量计！所有剩下的概率根据加和等于1的要求来确定，因此我们得到了一个完整的概率模型。

在我们观测到任何数据之前，油箱为空的先验概率为 $p(F = 0) = 0.1$ 。现在假设我们观察油量计，发现读数为“空的”，即 $G = 0$ ，对应于图8.21的中间的图。我们可以使用贝叶斯定理计算油箱为空的后验概率。首先，我们计算贝叶斯定理的分母，结果为

$$p(G = 0) = \sum_{B \in \{0,1\}} \sum_{F \in \{0,1\}} p(G = 0 | B, F) p(B) p(F) = 0.315 \quad (8.30)$$

类似地，我们计算

$$p(G = 0 | F = 0) = \sum_{B \in \{0,1\}} p(G = 0 | B, F = 0) p(B) = 0.81 \quad (8.31)$$

使用这些结果，我们有

$$p(F = 0 | G = 0) = \frac{p(G = 0 | F = 0) p(F = 0)}{p(G = 0)} \simeq 0.257 \quad (8.32)$$

从而 $p(F = 0 | G = 0) > p(F = 0)$ 。因此观测到油量计的读数为空使得油箱确实为空的概率增加，这与我们的直觉相符。接下来假设我们也检查了电池的状态，发现它没充电，即 $B = 0$ 。我们现在观测到了油量计的状态和电池的状态，如图8.21的右侧图所示。给定油量计的观测以及电池状态的观测，油箱为空的后验概率为

$$p(F = 0 | G = 0, B = 0) = \frac{p(G = 0 | B = 0, F = 0) p(F = 0)}{\sum_{F \in \{0,1\}} p(G = 0 | B = 0, F) p(F)} \simeq 0.111 \quad (8.33)$$

其中先验概率 $p(B = 0)$ 在分子和分母之间消去了。因此，由于电池状态的观测结果，油箱为空的概率减小了（从0.257到0.111）。这与我们的直觉相符，即发现电池没充电“辩解”（explain away）了油量计的读数为“空的”。我们看到，由于观测到了油量计的读数，油箱的状态和电池的状态确实变得不独立了。事实上，如果我们没有直接观测到油量计的读数，而是观测到了 G 的后继，那么情况仍然相同。注意，概率 $p(F = 0 | G = 0, B = 0) \simeq 0.111$ 大于先验概率 $p(F = 0) = 0.1$ ，因为观测到油量计读数为零仍然给油箱为空提供了一定的证据。



图 8.22: d-划分概念的说明。详细解释见正文。

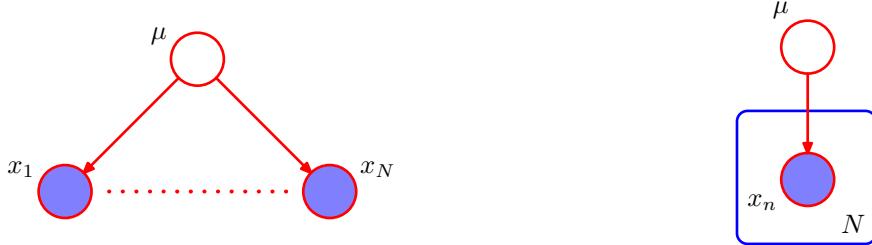


图 8.23: (a) 对应于推断观测 x_1, \dots, x_N 上的一元高斯分布的均值 μ 的问题的有向图。(b) 同样的图, 使用板的记号表示。

8.2.2 d-划分

我们现在给出有向图d-划分性质的一个一般的叙述 (Pearl, 1988)。考虑一个一般的有向图, 其中 A, B, C 是任意无交集的结点集合 (它们的并集可能比图中结点的完整集合要小)。我们希望弄清楚, 一个有向无环图是否暗示了一个特定的条件依赖表述 $A \perp\!\!\!\perp B | C$ 。为了解决这个问题, 我们考虑从 A 中任意结点到 B 中任意结点的所有可能的路径。我们说这样的路径被“阻隔”, 如果它包含一个结点满足下面两个性质中的任何一个。

- 路径上的箭头以头到尾或者尾到尾的方式交汇于这个结点, 且这个结点在集合 C 中。
- 箭头以头到头的方式交汇于这个结点, 且这个结点和它的所有后继都不在集合 C 中。

如果所有的路径都被“阻隔”, 那么我们说 C 把 A 从 B 中 d-划分开, 且图中所有变量上的联合概率分布将会满足 $A \perp\!\!\!\perp B | C$ 。

图 8.22 说明了 d-划分的概念。在图(a)中, 从 a 到 b 的路径没有被结点 f 阻隔, 因为对于这个路径来说, 它是一个尾到尾结点, 并且没有被观测到。这条路径也没有被结点 e 阻隔, 因为虽然后者是一个头到头的结点, 但是它有一个后继 c 在条件集合中。因此条件独立关系 $a \perp\!\!\!\perp b | c$ 在这个图中不成立。在图(b)中, 从 a 到 b 的路径被结点 f 阻隔, 因为它是一个尾到尾的结点, 并且被观测到, 因此使用这幅图进行分解的任何概率分布都满足条件独立性质 $a \perp\!\!\!\perp b | f$ 。注意, 这个路径也被结点 e 阻隔, 因为 e 是一个头到头的结点, 并且它和它的后继都没在条件集合中。

对于 d-划分的目的来说, 图 8.5 中用小实心圆表示的参数 (例如 α 和 σ^2) 与观测结点的行为相同。然而, 这些结点没有边缘概率分布。结果, 参数结点本身没有父结点, 因此所有通过这些结点的路径总是尾到尾的, 因此是阻隔的。从而它们在 d-划分中没有作用。

1.2.4 节介绍的独立同分布数据的概念提供了条件独立和 d-划分的另一个例子。考虑寻找一元高斯分布的均值的后验概率分布的问题。这可以表示为图 8.23 的有向图的形式, 其中联合概率分布由先验概率分布 $p(\mu)$ 和一组条件概率分布 $p(x_n | \mu)$ 表示, 其中 $n = 1, \dots, N$ 。在实际应用中, 我们观测到 $\mathcal{D} = \{x_1, \dots, x_N\}$, 我们的目标是推断 μ 。我们现在假设我们以 μ 为条件, 考虑观测的联合概率分布。使用 d-划分, 我们注意到从任意结点 x_i 到其他的结点 $x_{j \neq i}$ 有一条唯一的路径, 这个路径关于观测结点 μ 是尾到尾的。每条这样的路径都是阻隔的, 因此给定 μ , 观测 $\mathcal{D} = \{x_1, \dots, x_N\}$ 是独立的, 即

$$p(\mathcal{D} | \mu) = \prod_{n=1}^N p(x_n | \mu) \quad (8.34)$$

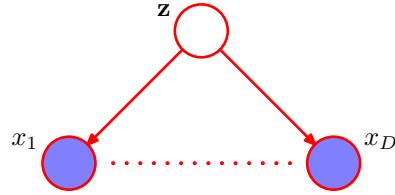


图 8.24: 用于分类的“朴素贝叶斯”模型的图表示。以类别标签 z 为条件，观测向量 $\mathbf{x} = (x_1, \dots, x_D)^T$ 的元素假设是独立的。

然而，如果我们对 μ 积分，通常观测不再独立，即

$$p(\mathcal{D}) = \int_{-\infty}^{\infty} p(\mathcal{D} | \mu)p(\mu) d\mu \neq \prod_{n=1}^N p(x_n) \quad (8.35)$$

这里 μ 是一个潜在变量，因为它的值未被观测。

另一个表示独立同分布数据模型的例子如图8.7所示，它对应于贝叶斯多项式回归。这里，随机结点对应于 $\{t_n\}$, w 和 \hat{t} 。我们看到， w 的结点关于从 \hat{t} 到任意结点 t_n 的路径是尾对尾的，因此我们有下面的条件独立性质

$$\hat{t} \perp\!\!\!\perp t_n \mid w \quad (8.36)$$

因此，以多项式系数 w 为条件， \hat{t} 的预测分布独立于训练数据 $\{t_1, \dots, t_N\}$ 。于是我们可以首先使用训练数据确定系数 w 的后验概率分布，然后我们就可以丢弃训练数据，使用 w 的后验概率分布对新输入观测 \hat{x} 做出 \hat{t} 的预测。

一种被称为朴素贝叶斯 (naive Bayes) 模型的分类方法可以产生一种相关的图结构，其中我们使用条件独立性假设来简化模型的结构。假设观测变量由 D 维向量 $\mathbf{x} = (x_1, \dots, x_D)^T$ 组成，我们希望将 \mathbf{x} 的观测值分配到 K 个类别中的一个。使用“1-of- K ”表示方式，我们可以使用一个 K 维二值向量 z 表示这些类别。然后我们可以这样定义一个生成式模型：引入类别标签上的多项式先验概率分布 $p(z | \mu)$ ，其中 μ 的第 k 个元素 μ_k 表示类别 C_k 的先验概率，再引入观测向量 \mathbf{x} 的条件概率分布 $p(\mathbf{x} | z)$ 。朴素贝叶斯模型的关键假设是，以类别 z 为条件，输入变量 x_1, \dots, x_D 的分布是独立的。这个模型的图表示如图8.24所示。我们看到 z 的观测阻隔了从 x_i 到 x_j 的路径，其中 $j \neq i$ ，因为这样的路径在结点 z 处是尾到尾的。因此给定 z 的条件下， x_i 和 x_j 是条件独立的。然而，如果我们对 z 求和或积分（即 z 不是观测变量），那么从 x_i 到 x_j 的尾到尾路径就不再是阻塞的了。这告诉我们，通常边缘概率密度 $p(\mathbf{x})$ 不可以关于 \mathbf{x} 的元素进行分解。在1.5节中讨论将不同来源的医疗诊断数据整合到一起的问题时，我们遇到了朴素贝叶斯模型的一个简单的例子。

如果给定一个有标记的训练集，由输入 $\{x_1, \dots, x_N\}$ 以及它们的类别标签组成，那么我们可以使用最大似然法，根据训练数据调整朴素贝叶斯模型，其中我们假设数据是独立地从模型中抽取的。使用每个类别对应的标记数据，我们可以为每个类别分别调整一个模型，得到最终解。举例来说，假设每个类别的概率密度分布被选为高斯分布。在这种情况下，朴素贝叶斯的假设表明每个高斯分布的协方差矩阵是对角矩阵，且每个类别中常数密度的轮廓线是与坐标轴对齐的椭球。然而，边缘概率密度由对角高斯的叠加组成（权系数由类别先验给出），因此不再能够关于各个分量进行分解。

当输入空间的维度 D 很高时，在完整的 D 维空间进行概率密度估计比较困难，此时朴素贝叶斯的假设很有帮助。如果输入向量既包含离散变量又包含连续变量，那么朴素贝叶斯的假设也很有意义，因为每个变量都可以分别使用合适的模型进行表示，例如用伯努利分布表示二值观测，或者用高斯分布表示实值变量。这个模型中的条件独立性假设显然过于强烈，可能会导致对类条件概率密度的表示相当差。尽管这样，即使这个假设无法精确满足，但是模型仍然可能给出较好的分类效果，因为决策边界对于类条件概率的细节不敏感，如图1.27所示。

我们已经看到一个特定的有向图表示将联合概率分布分解为条件概率分布乘积形式的一个具体的分解方式。图也表示一组条件独立的性质，这些性质通过d-划分的方式得到，并且d-划分定理实际上是一个等价于这两个性质的表示。为了让这一点更明显，将有向图想象成滤波器是很有帮助的。假设我们考虑 \mathbf{x} 上的一个特定的联合概率分布 $p(\mathbf{x})$ ，其中 \mathbf{x} 对应于图中的（未观测）

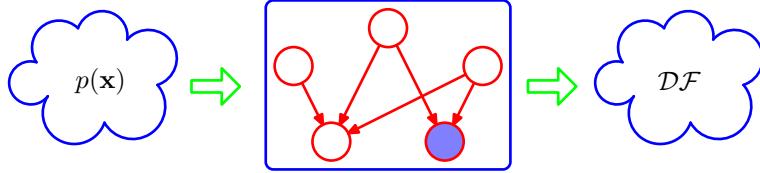


图 8.25: 我们可以将图模型（在这幅图中是有向图）看成滤波器。当且仅当概率分布 $p(\mathbf{x})$ 满足有向分解性质 (8.5) 时， $p(\mathbf{x})$ 才可以通过滤波器。通过滤波器的所有可能的概率分布 $p(\mathbf{x})$ 被记作 \mathcal{DF} 。我们也可以根据概率分布是否满足由图的d-划分性质表示的所有条件独立性质来使用图对概率分布进行过滤。d-划分定理表明，这两种滤波方式得到的概率分布集合 \mathcal{DF} 是相同的。

结点。一个概率分布能够通过滤波器当且仅当它能够用与图对应的公式 (8.5) 给出的分解方式进行分解。如果我们将变量 \mathbf{x} 的集合上的所有可能的概率分布 $p(\mathbf{x})$ 输入到滤波器中，那么通过滤波器的概率分布的子集被记作 \mathcal{DF} ，表示有向分解 (directed factorization)，如图8.25所示。我们还可以将图用作另一种滤波器，首先将d-划分准则应用到图中，列出所有得到的条件独立性质，然后只有当一个概率分布满足所有这些性质时才允许通过。如果我们将所有可能的概率分布输入到这一类滤波器中，那么d-划分定理告诉我们，允许通过的概率分布的集合就是 \mathcal{DF} 。

应该强调的是，从d-划分中得到的条件独立性质适用于任何由那个特定的有向图描述的概率模型。例如，无论变量是离散的还是连续的还是二者的组合，这个性质都成立。与之前一样，我们看到特定的图描述了一大类概率分布。

在一种极限的情况下，我们有一个全连接的图，它不表示任何的条件独立性质，可以表示给定变量上的任何可能的联合概率分布。集合 \mathcal{DF} 将包含所有可能的概率分布 $p(\mathbf{x})$ 。在另一种情况下，我们有一个完全非连接的图，即一张不存在任何链接的图。这对应的联合概率分布可以分解为图结点组成的变量上的边缘概率分布的乘积。

注意，对于任意给定的图，分布的集合 \mathcal{DF} 中的概率分布还会具有图中未描述的独立性质。例如，一个完全分解的概率分布总会通过由对应变量组成的任意图结构表示的滤波器。

最后，我们通过研究马尔科夫毯 (Markov blanket) 或者马尔科夫边界 (Markov boundary) 的概念来结束我们关于条件独立性的讨论。考虑一个联合概率分布 $p(\mathbf{x}_1, \dots, \mathbf{x}_D)$ ，它由一个具有 D 个结点的有向图表示。考虑变量 \mathbf{x}_i 对应的结点上的条件概率分布，其中条件为所有剩余的变量 $\mathbf{x}_{j \neq i}$ 。使用分解性质 (8.5)，我们可以将条件概率分布表示为下面的形式

$$\begin{aligned} p(\mathbf{x}_i | \mathbf{x}_{\{j \neq i\}}) &= \frac{p(\mathbf{x}_1, \dots, \mathbf{x}_D)}{\int p(\mathbf{x}_1, \dots, \mathbf{x}_D) d\mathbf{x}_i} \\ &= \frac{\prod_k p(\mathbf{x}_k | \text{pa}_k)}{\int \prod_k p(\mathbf{x}_k | \text{pa}_k) d\mathbf{x}_i} \end{aligned}$$

对于离散变量，上式中的积分需要替换为求和式。我们现在观察到任何与 \mathbf{x}_i 没有函数依赖关系的因子都可以提到 \mathbf{x}_i 的积分外面，从而在分子和分母之间消去。唯一剩余的因子是结点 \mathbf{x}_i 本身的条件概率分布 $p(\mathbf{x}_i | \text{pa}_i)$ ，以及满足下面性质的结点 \mathbf{x}_k 的条件概率分布：结点 \mathbf{x}_i 在 $p(\mathbf{x}_k | \text{pa}_k)$ 的条件集合中，即 \mathbf{x}_i 是 \mathbf{x}_k 的父结点。条件概率分布 $p(\mathbf{x}_i | \text{pa}_i)$ 依赖于结点 \mathbf{x}_i 的父结点，而条件概率分布 $p(\mathbf{x}_k | \text{pa}_k)$ 依赖于 \mathbf{x}_i 的子结点以及同父结点 (co-parents)，即那些对应于 \mathbf{x}_k (而不是 \mathbf{x}_i) 的父结点的变量。由父结点、子结点、同父结点组成的结点集合被称为马尔科夫毯，如图8.26所示。我们可以将结点 \mathbf{x}_i 的马尔科夫毯想象成将 \mathbf{x}_i 与图的剩余部分隔离开的最小结点集合。注意，只包含 \mathbf{x}_i 的父结点和子结点是不够的，因为之前的例子表明，子结点的观测不会阻隔某个结点到同父结点的路径。因此我们必须也观测同父结点。

8.3 马尔科夫随机场

我们已经看到有向图模型表示将一组变量上的联合概率分布分解为局部条件概率分布的乘积的一种分解方式。有向图模型也定义了一组条件独立性质，根据图进行分解的任何概率分布都必须满足这些条件独立性质。我们现在考虑图模型的第二大类，使用无向图描述的图模型。与之前一样，它表示一个分解方式，也表示一组条件独立关系。

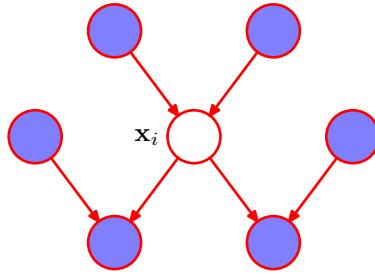


图 8.26: 结点 x_i 的马尔科夫毯由父结点、子结点、同父结点组成的集合构成。它的性质为：以图中所有剩余结点为条件， x_i 的条件概率分布值依赖于马尔科夫毯中的变量。

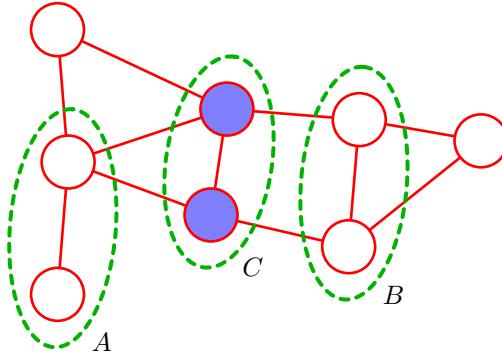


图 8.27: 无向图的一个例子，其中从集合 A 中的任意结点到集合 B 中的任意结点的每条路径都通过集合 C 中的至少一个结点。结果，对于所有由这个图描述的任意概率分布，以 C 为条件， A 与 B 都条件独立。

一个马尔科夫随机场 (Markov random field)，也被称为马尔科夫网络 (Markov network) 或者无向图模型 (undirected graphical model) (Kindermann and Snell, 1980)，包含一组结点，每个结点都对应着一个变量或一组变量。链接是无向的，即不含有箭头。在无向图的情形中，首先讨论条件独立性质是比较方便的。

8.3.1 条件独立性质

在有向图的情形下，我们看到可以通过使用被称为d-划分的图检测方法判断一个特定的条件独立性质是否成立。这涉及到判断链接两个结点集合的路径是否被“阻隔”。然而，由于头到头结点的存在，阻隔的定义多少有些微妙。我们可能会问，是否可以定义另一种概率分布的图语义表示，使得条件独立性由单一的图划分确定。这种情形确实存在，对应于无向图模型。通过移除图中链接的方向性，父结点和子结点的非对称性也被移除了，因此头到头结点的微妙性也就不再存在了。

假设在一个无向图中，我们有三个结点集合，记作 A, B, C 。我们考虑条件独立性质

$$A \perp\!\!\!\perp B \mid C \quad (8.37)$$

为了判定由图定义的概率分布是否满足这个性质，我们考虑连接集合 A 的结点和集合 B 的结点的所有可能路径。如果所有这些路径都通过了集合 C 中的一个或多个结点，那么所有这样的路径都被“阻隔”，因此条件独立性质成立。然而，如果存在至少一条未被阻隔的路径，那么性质条件独立的性质未必成立，或者更精确地说，存在至少某些对应于图的概率分布不满足条件独立性质。图 8.27 给出了一个例子。注意，这与 d 划分的准则完全相同，唯一的差别在于没有头到头的现象。因此，无向图的条件独立性的检测比有向图简单。

另一种条件独立性的检测的方法是假设从图中把集合 C 中的结点以及与这些结点相连的链接全部删除。然后，我们考察是否存在一条从 A 中任意结点到 B 中任意结点的路径。如果没有这样的路径，那么条件独立的性质一定成立。

无向图的马尔科夫毯的形式相当简单，因为结点只条件依赖于相邻结点，而条件独立于任何其他的结点，如图 8.28 所示。

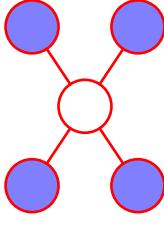


图 8.28: 对于一个无向图, 结点 x_i 的马尔科夫毯由相邻结点的集合组成。它的性质为: 以图中所有剩余变量为条件, x_i 的条件概率分布只依赖于马尔科夫毯中的变量。

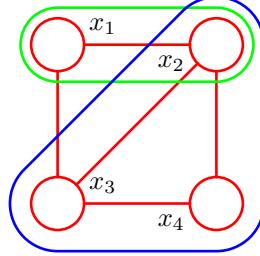


图 8.29: 4结点无向图。图中画出了一个团块（用绿色圈出）和一个最大团块（用蓝色圈出）。

8.3.2 分解性质

我们现在寻找无向图的一个分解规则, 对应于上述条件独立性检测。与之前一样, 这涉及到将联合概率分布 $p(\mathbf{x})$ 表示为在图的局部范围内的变量集合上定义的函数的乘积。于是, 我们需要给出这种情形下, 局部性的一个合适定义。

如果我们考虑两个结点 x_i 和 x_j , 它们不存在链接, 那么给定图中的所有其他结点, 这两个结点一定是条件独立的。这是因为两个结点之间没有直接的路径, 并且所有其他的路径都通过了观测的结点, 因此这些路径都是被阻隔的。这个条件独立性可以表示为

$$p(x_i, x_j | \mathbf{x}_{\setminus\{i,j\}}) = p(x_i | \mathbf{x}_{\setminus\{i,j\}})p(x_j | \mathbf{x}_{\setminus\{i,j\}}) \quad (8.38)$$

其中 $\mathbf{x}_{\setminus\{i,j\}}$ 表示所有变量 \mathbf{x} 去掉 x_i 和 x_j 的集合。于是, 联合概率分布的分解一定要让 x_i 和 x_j 不出现在同一个因子中, 从而让属于这个图的所有可能的概率分布都满足条件独立性质。

这将我们引向了一个图形的概念, 团块 (clique)。它被定义为图中结点的一个子集, 使得在这个子集中的每对结点之间都存在链接。换句话说, 团块中的结点集合是全连接的。此外, 一个最大团块 (maximal clique) 是具有下面性质的团块: 不可能将图中的任何一个其他的结点包含到这个团块中而不破坏团块的性质。图8.29说明了四个变量的无向图中的这些概念。这个图中有五个具有两个结点的团块, 即 $\{x_1, x_2\}, \{x_2, x_3\}, \{x_3, x_4\}, \{x_4, x_2\}$ 和 $\{x_1, x_3\}$, 还有两个最大团块 $\{x_1, x_2, x_3\}$ 和 $\{x_2, x_3, x_4\}$ 。集合 $\{x_1, x_2, x_3, x_4\}$ 不是一个团块, 因为在 x_1 和 x_4 没有链接。

于是, 我们可以将联合概率分布分解的因子定义为团块中变量的函数。事实上, 我们可以考虑最大团块的函数而不失一般性, 因为其他团块一定是最团块的子集。因此, 如果 $\{x_1, x_2, x_3\}$ 是一个最大团块, 并我们在这个团块上定义了任意一个函数, 那么定义在这些变量的一个子集上的其他因子都是冗余的。

让我们将团块记作 C , 将团块中的变量的集合记作 \mathbf{x}_C 。这样, 联合概率分布可以写成图的最大团块的势函数 (potential function) $\psi_C(\mathbf{x}_C)$ 的乘积的形式

$$p(\mathbf{x}) = \frac{1}{Z} \prod_C \psi_C(\mathbf{x}_C) \quad (8.39)$$

这里, Z 有时被称为划分函数 (partition function), 是一个归一化常数, 等于

$$Z = \sum_{\mathbf{x}} \prod_C \psi_C(\mathbf{x}_C) \quad (8.40)$$

它确保了公式 (8.39) 给出的概率分布 $p(\mathbf{x})$ 被正确地归一化。通过只考虑满足 $\psi_C(\mathbf{x}_C) \geq 0$ 的势函数，我们确保了 $p(\mathbf{x}) \geq 0$ 。在公式 (8.40) 中，我们假设 \mathbf{x} 由离散变量组成，但是这个框架也同样适用于连续变量，或者两者结合的情形。此时，求和式被替换成恰当的求和与积分的组合。

注意，我们不把势函数的选择限制为具有具体的概率含义（例如边缘概率分布或者条件概率分布）的函数。这与有向图的情形相反。在有向图的情形中，每个因子表示对应变量以它的父结点为条件的条件概率分布。然而，在特殊情况下，例如无向图是通过有向图构建的情况，势函数可能确实有这样的意义，正如我们将要看到的那样。

势函数 $\psi_C(\mathbf{x}_C)$ 的这一通用性产生的一个结果是它们的乘积通常没有被正确地归一化。于是，我们必须引入一个显式的归一化因子，由公式 (8.40) 给出。回忆一下，对于有向图的情形，由于分解后的每个作为因子的条件概率分布都被归一化了，因此联合概率分布会自动地被归一化。

归一化常数的存在是无向图的一个主要的缺点。如果我们的模型中有 M 个离散结点，每个结点有 K 个状态，那么归一化项的计算涉及到对 K^M 个状态求和，因此（在最坏的情况下），计算量是模型大小的指数形式。对于参数学习来说，划分函数是必要的，因为划分函数是控制势函数 $\psi_C(\mathbf{x}_C)$ 的任意参数的函数。但是，对于局部条件概率分布的计算，划分函数是不需要的，因为条件概率是两个边缘概率的比值，当计算这个比值时，划分函数在分子和分母之间被消去了。类似地，对于计算局部边缘概率，我们可以计算未归一化的联合概率分布，然后在计算的最后阶段显式地归一化边缘概率。假设边缘概率只涉及到少量的变量，那么归一化系数的计算是可行的。

目前为止，我们基于简单的图划分，讨论了条件独立性的概念，并且我们提出了对联合概率分布的分解，来尝试对应条件独立的图结构。然而，我们并没有将条件独立性和无向图的分解形式化地联系起来。为了形式化地描述，我们需要把注意力限制于那些严格为正的势函数 $\psi_C(\mathbf{x}_C)$ ，即对于任意的 \mathbf{x}_C 的选择都永远不等于零也不取负值的势函数。给定这个限制，我们可以给出分解和条件独立之间的精确关系。

为了给出精确的关系，我们再次回到作为滤波器的图模型的概念中，对应于图 8.25。考虑定义在固定变量集合上的所有可能的概率分布，其中这些变量对应于一个具体的无向图的节点。我们可以将 $\mathcal{U}\mathcal{I}$ 定义为满足下面条件的概率分布的集合：从使用图划分的方法得到的图中可以读出条件独立性质，这个概率分布应该与这些条件独立性质相容。类似地，我们可以将 $\mathcal{U}\mathcal{F}$ 定义为满足下面条件的概率分布的集合：可以表示为关于图中最大团块的分解的形式的概率分布，其中分解方式由公式 (8.39) 给出。Hammersley-Clifford 定理 (Clifford, 1990) 表明，集合 $\mathcal{U}\mathcal{I}$ 和 $\mathcal{U}\mathcal{F}$ 是完全相同的。

由于我们的势函数被限制为严格大于零，因此将势函数表示为指数的形式更方便，即

$$\psi_C(\mathbf{x}_C) = \exp\{-E(\mathbf{x}_C)\} \quad (8.41)$$

其中 $E(\mathbf{x}_C)$ 被称为能量函数 (energy function)，指数表示被称为玻尔兹曼分布 (Boltzmann distribution)。联合概率分布被定义为势函数的乘积，因此总的能量可以通过将每个最大团块的能量相加的方法得到。

与有向图的联合分布的因子不同，无向图中的势函数没有一个具体的概率意义。虽然这使得选择势函数具有更大的灵活性，因为没有归一化的限制，但是这确实产生了一个问题，即对于一个具体的应用来说，如何选择势函数。可以这样做：将势函数看成一种度量，它表示了局部变量的哪种配置优于其他的配置。具有相对高概率的全局配置对于各个团块的势函数的影响进行了很好的平衡。我们现在通过一个具体的例子来说明无向图的用处。

8.3.3 例子：图像去噪

我们可以使用二值图像中图像去噪的例子来说明无向图的应用 (Besag, 1974; Geman and Geman, 1984; Besag, 1986)。虽然这是一个非常简单的例子，但是它可以代表许多更复杂的应用。我们令观测的噪声图像通过一个二值像素值 $y_i \in \{-1, +1\}$ 组成的数组来描述，其中下标 $i = 1, \dots, D$ 覆盖了所有的像素。我们假设图像通过下面的方式获得：取一张未知的无噪声图像，这幅图像由二值像素值 $x_i \in \{-1, +1\}$ 描述，然后以一个较小的概率随机翻转像素值的符

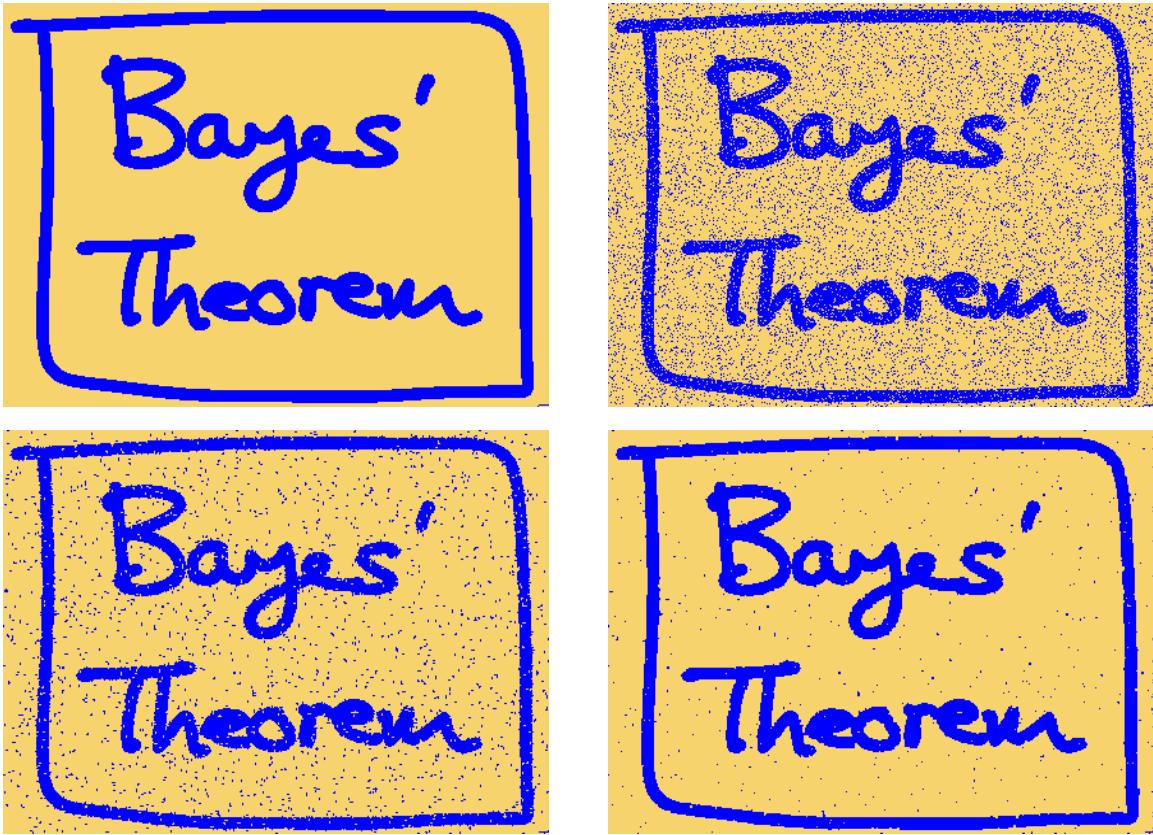


图 8.30: 使用马尔科夫随机场进行图像去噪的例子。上面一行中，左侧是原始的二值图像，右侧是随机改变10%的像素后得到的带有噪声的图像。下面一行中，左图是使用迭代条件模型 (ICM) 恢复的图像，右图是使用最大割算法得到的图像。ICM产生的图像中，96%的像素与原始图像相符，而最大割算法产生的图像中，这个比例为99%。

号。图8.30给出了一个二值图像以及一副噪声图像，其中噪声图像中像素值发生符号翻转的概率为10%。给定带有噪声的图像，我们的目标是恢复原始的无噪声的图像。

由于噪声等级比较小，因此我们知道 x_i 和 y_i 之间有着强烈的相关性。我们还知道图像中相邻像素 x_i 和 x_j 的相关性很强。这种先验知识可以使用马尔科夫随机场模型进行描述，它的无向图如图8.31所示。这个图中有两种类型的团块，每一种团块包含两个变量。形如 $\{x_i, y_i\}$ 的团块有一个关联的能量函数，表达了这些变量之间的相关性。对于这些团块，我们选择一个非常简单的能量函数 $-\eta x_i y_i$ ，其中 η 是一个正的常数。这个能量函数的效果是：当 x_i 和 y_i 符号相同时，能量函数会给出一个较低的能量（即，较高的概率），而当 x_i 和 y_i 符号相反时，能量函数会给出一个较高的能量。

剩余的团块由变量 $\{x_i, x_j\}$ 组成，其中 i 和 j 是相邻像素的下标。与之前一样，我们希望当两个像素符号相同时能量较低，当两个像素符号相反对时能量较高，因此我们选择能量函数 $-\beta x_i x_j$ ，其中 β 是一个正的常数。

由于势函数是最大团块上的一个任意的非负的函数，因此我们可以将势函数与团块的子集上的任意非负函数相乘，或者等价地，我们可以加上对应的能量。在这个例子中，这使得我们可以为无噪声图像的每个像素 i 加上一个额外的项 $h x_i$ 。这样的项具有下面的效果：将模型进行偏置，使得模型倾向于选择一个特定的符号，而不选择另一个符号。

于是，模型的完整的能量函数的形式为

$$E(\mathbf{x}, \mathbf{y}) = h \sum_i x_i - \beta \sum_{\{i,j\}} x_i x_j - \eta \sum_i x_i y_i \quad (8.42)$$

它定义了 \mathbf{x} 和 \mathbf{y} 上的一个联合概率分布，形式为

$$p(\mathbf{x}, \mathbf{y}) = \frac{1}{Z} \exp\{-E(\mathbf{x}, \mathbf{y})\} \quad (8.43)$$



图 8.31: 一个无向图模型，表示图像去噪的马尔科夫随机场，其中 x_i 是一个二值变量，表示像素*i*在一个未知的无噪声的图像中的状态， y_i 表示在观测到的噪声图像中，像素*i*的对应值。

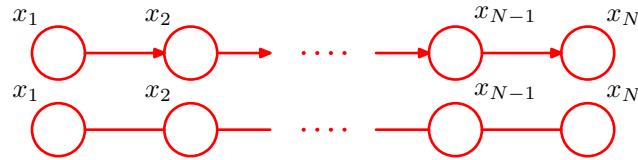


图 8.32: (a)有向图的例子。(b)等价的无向图。

我们现在固定噪声图像的像素给出的观测值 \mathbf{y} 的元素，这个噪声图像隐式地定义了一个无噪声图像上的条件概率分布 $p(\mathbf{x} | \mathbf{y})$ 。这是Ising模型的一个例子，这个模型在统计物理学中被广泛研究。为了恢复图像，我们希望找到一个具有较高概率（理想情况下具有最高概率）的图像 \mathbf{x} 。为了完成这件事，我们要使用一个简单的迭代方法，叫做迭代条件峰值（iterated conditional modes），或者称为ICM (Kittler and Föglein, 1984)。这种方法仅仅是坐标间的梯度上升方法的一个应用。这种方法的思想是，首先初始化变量 $\{x_i\}$ ，这个过程中我们只是简单地令 $x_i = y_i$ 对于所有*i*都成立。然后，我们每次取一个 x_j 结点，计算两个可能状态 $x_j = +1$ 和 $x_j = -1$ 的总能量，保持其他所有结点变量固定，将 x_j 设置为能量较低的状态。如果 x_j 不变，则概率不变，否则概率就会增大。由于只有一个变量发生改变，因此这是一个可以高效进行的简单局部计算。然后，我们对其他的结点重复更新过程，以此类推，知道满足某个合适的停止条件。结点可以用一种系统的方式更新，例如重复地依次扫描图像，或者随机地选择结点。

如果我们有一个更新的顺序，使得每个像素都至少被访问一次，且没有变量发生改变，那么根据定义，算法会收敛于概率的一个局部最大值。然而，这未必对应于全局最大值。

对于这个简单的例子来说，我们将参数固定为 $\beta = 1.0$, $\eta = 2.1$ 以及 $h = 0$ 。注意，令 $h = 0$ 意味着两个状态 x_i 的先验概率是相等的。首先，我们使用噪声图像进行初始化，然后运行ICM直到收敛，得到了图8.30左下角的去噪图像。注意，如果我们令 $\beta = 0$ ，从而去除了相邻像素点之间的联系，那么整体概率最大的解为 $x_i = y_i$ （对于所有的*i*），这对应于观测到的噪声图像。

稍后，我们会讨论一种更加高效的算法寻找高概率的解，这种算法被称为最大加和算法，它通常会产生更好的解，虽然这种算法仍然不保证找到后验概率的全局最大值。然而，对于某类模型，包括由公式 (8.42) 给出的模型，存在基于图割 (graph cut) 的高效的算法，保证找到全局的最大值 (Greig et al., 1989; Boykov et al., 2001; Kolmogorov and Zabih, 2004)。图8.30的右下角给出了将图割算法应用于去噪问题的结果。

8.3.4 与有向图的关系

我们已经介绍了表示概率分布的两个图模型的框架，对应于有向图和无向图。讨论二者之间的关系是很有意义的。首先考虑下面的问题：取一个使用有向图描述的模型，尝试将其转化为无向图。在某些情况下，转化方法很直接，例如图8.32给出的简单例子。这里，有向图的联合概率分布由一组条件概率分布的乘积给出，形式为

$$p(\mathbf{x}) = p(x_1)p(x_2 | x_1)p(x_3 | x_2) \cdots p(x_N | x_{N-1}) \quad (8.44)$$



图 8.33: (a)一个简单的有向图的例子。(b)对应的道德图。

现在假设我们将其转化为无向图的表示方法，如图8.32所示。在无向图中，最大团块为相邻结点对，因此根据公式 (8.39)，我们希望将联合概率分布写成下面的形式

$$p(\mathbf{x}) = \frac{1}{Z} \psi_{1,2}(x_1, x_2) \psi_{2,3}(x_2, x_3) \cdots \psi_{N-1,N}(x_{N-1}, x_N) \quad (8.45)$$

这很容易做。我们只需令

$$\psi_{1,2}(x_1, x_2) = p(x_1)p(x_2 | x_1)$$

$$\psi_{2,3}(x_2, x_3) = p(x_3 | x_2)$$

⋮

$$\psi_{N-1,N}(x_{N-1}, x_N) = p(x_N | x_{N-1})$$

其中我们将第一个结点的边缘概率分布 $p(x_1)$ 放到了第一个势函数中。注意，在这种情况下，划分函数为 $Z = 1$ 。

让我们考虑如何推广这个结构，使得我们可以将任意由有向图的分解给出的概率分布转化为用无向图的分解表示的概率分布。如果无向图的团块势函数由有向图的条件概率分布给出，那么这个任务就可以完成。为了保持这个过程的合法性，我们必须确保出现在每个条件概率分布中的变量的集合是无向图中至少一个团块的成员。对于有向图中只有一个父结点的结点，可以通过简单地将有向链接替换为无向链接的方式完成。然而，对于有向图中具有多个父结点的结点来说，这样做是不够的。这些结点是我们在讨论条件独立性时遇到的“头到头”路径的结点。考虑图8.33所示的具有4个结点的简单有向图。有向图的联合概率分布为

$$p(\mathbf{x}) = p(x_1)p(x_2)p(x_3)p(x_4 | x_1x_2x_3) \quad (8.46)$$

我们看到因子 $p(x_4 | x_1, x_2, x_3)$ 涉及到四个变量 x_1, x_2, x_3 和 x_4 ，所以如果这个条件概率分布被整合到团块势函数中时，这些变量一定属于一个团块。为了确保这一点，我们在 x_4 的所有父结点之间添加额外的链接。使用一种过时的说法，这种“与父结点结婚”的过程被称为伦理 (moralization)，去掉箭头后生成的无向图被称为道德图 (moral graph)。很重要的一点是，这个例子中的道德图是完全链接的，因此没有表现出条件独立性质，这与原始的有向图相反。

因此，通常为了将有向图转化为无向图，我们首先在图中每个结点的所有父结点之间添加额外的无向链接，然后去掉原始链接的箭头，得到道德图。之后，我们将道德图的所有的团块势函数初始化为1。接下来，我们拿出原始有向图中所有的条件概率分布因子，将它乘到一个团块势函数中去。由于“伦理”步骤的存在，总会存在至少一个最大的团块，包含因子中的所有变量。注意，在所有情形下，划分函数都为 $Z = 1$ 。

将有向图转化为无向图的过程在精确推断方法中起着重要的作用，例如联合树算法 (junction tree algorithm)。从一个无向图转化到有向图表示不太常用，通常表示归一化限制中出现的问题。

我们看到从一个有向图表示转化为无向图表示的过程中，我们必须从图中丢弃掉一些条件独立性质。当然，通过简单地使用全连接的无向图，我们可以很容易地将有向图上的任意概率分布转化为无向图上的概率分布。但是，这会丢弃掉所有的条件独立性质，因此没有意义。“伦理”过程增加了最少的额外链接，因此保持了最大的条件独立性质。

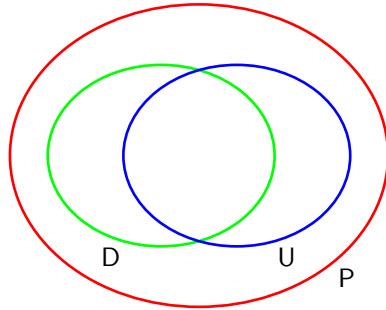


图 8.34: Venn 图, 给出了一个给定的变量集合上的所有分布的集合 P 以及可以用有效图表示为完美图的分布集合 D , 还有可以使用无向图表示的完美图的分布集合 U 。

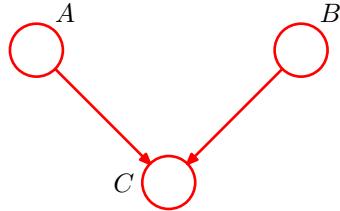


图 8.35: 条件独立性质不能够使用同样的三个变量上的无向图表示的有向图。

我们已经看到判断条件独立性质的步骤对于有向图和无向图是不同的。可以证明, 这两种类型的图可以表示不同的条件独立性质。这一点很值得详细研究。为了研究这个问题, 我们回到一个具体的用作滤波器的有向图或者无向图, 从而给定变量上的所有可能的概率分布的集合都可以被化简为一个子集, 这个子集保持了图给出的条件独立性质。如果一个概率分布中的所有条件独立性质都通过一个图反映出来, 那么这个图被称为这个概率分布的D图 (D map, 表示“依赖图” (dependency map))。因此一个完全非连接的图 (不存在链接) 是任意概率分布的平凡D图。

我们还可以考虑一个具体的概率分布, 判断哪些图具有适当的条件独立性质。如果一个图的每个条件独立性质都可以由一个具体的概率分布满足, 那么这个图被称为这个概率分布的I图 (I map, 表示“独立图” (independence map))。显然, 一个完全连接的图是任意概率分布的平凡I图。

如果概率分布的每个条件独立性质都可以由图反映出来, 反之也成立, 那么这个图被称为是概率分布的完美图 (perfect map)。于是, 一个完美图既是I图又是D图。

考虑概率分布的集合, 对于每个概率分布, 都存在一个有向图, 且这个有向图是完美图。这个集合与概率分布组成的下面的集合不同: 对于每个概率分布, 存在一个无向图, 这个无向图是完美图。此外, 存在这样的概率分布: 有向图和无向图都无法成为它的完美图。图8.34给出了这个关系的Venn图表示。

图8.35给出了一个有向图, 它是满足条件独立性质 $A \perp\!\!\!\perp B | \emptyset$ 和 $A \perp\!\!\!\perp B | C$ 的概率分布的一个完美图。这三个变量上的对应的无向图中, 不存在完美图。

相反, 考虑四个变量上的无向图, 如图 8.36 所示。这个图表示条件独立性

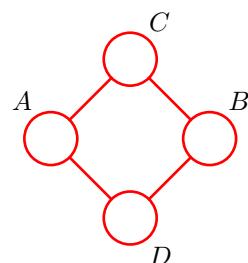


图 8.36: 条件独立性质不能够使用同样的变量集合上的有向图表示的无向图。

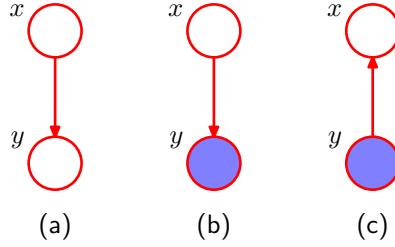


图 8.37: 贝叶斯定理的图表示。详细讨论见正文。

质 $A \perp\!\!\!\perp B | \emptyset, C \perp\!\!\!\perp D | A \cup B$ 以及 $A \perp\!\!\!\perp B | C \cup D$ 。这四个变量上的有向图中，不存在表示同样的条件独立性质集合的有向图。

图框架可以用一种相容的方式，扩展为同时包含有向链接和无向链接的图。这种图被称为链图（chain graphs）（Lauritzen and Wermuth, 1989; Frydenberg, 1990），将有向图和无向图都当成了具体的实例。虽然与有向图或者无向图自身相比，这种图可以表示更多的概率分布，但是仍然存在概率分布，使得链图也无法给出一个完美图。本书不会详细讨论链图。

8.4 图模型中的推断

我们现在考虑图模型中的推断问题，图中的一些结点被限制为观测值，我们想要计算其他结点中的一个或多个子集的后验概率分布。正如我们将看到的那样，我们可以利用图结构找到高效的推断算法，也可以让这些算法的结构变得透明。具体来说，我们会看到许多算法可以用图中局部信息传播的方式表示。本节中，我们会把注意力主要集中于精确推断的方法。在第10章中，我们会考虑许多近似推断的算法。

首先，让我们考虑贝叶斯定理的图表示。假设我们将两个变量 x 和 y 上的联合概率分布 $p(x, y)$ 分解为因子的形式 $p(x, y) = p(x)p(y | x)$ 。这可以用图 8.37(a) 中的有向图表示。现在假设我们观测到了 y 的值，如图 8.37(b) 中的阴影结点所示。我们可以将边缘概率分布 $p(x)$ 看成潜在变量 x 上的先验概率分布，我们的目标是推断 x 上对应的后验概率分布。使用概率的加和规则和乘积规则，我们可以计算

$$p(y) = \sum_{x'} p(y | x') p(x') \quad (8.47)$$

这个式子然后被用于贝叶斯定理中，计算

$$p(x | y) = \frac{p(y | x)p(x)}{p(y)} \quad (8.48)$$

因此现在联合概率分布可以通过 $p(y)$ 和 $p(x | y)$ 。从图的角度看，联合概率分布 $p(x, y)$ 现在可以表示为图 8.37(c) 所示的图，其中箭头的方向翻转了。这是图模型中推断问题的最简单的例子。

8.4.1 链推断

现在考虑一个更加复杂的问题，涉及到图 8.32 所示的结点链。这个例子是本节中对更一般的图的精确推断的讨论的基础。

具体地，我们会考虑图 8.32(b) 所示的无向图。我们已经看到，有向链可以被转化为一个等价的无向链。由于有向图中任何结点的父结点数量都不超过一个，因此不需要添加任何额外的链接，并且图的有向版本和无向版本表示完全相同的条件依赖性质集合。

这个图的联合概率分布形式为

$$p(\mathbf{x}) = \frac{1}{Z} \psi_{1,2}(x_1, x_2) \psi_{2,3}(x_2, x_3) \cdots \psi_{N-1,N}(x_{N-1}, x_N) \quad (8.49)$$

我们会考虑一个具体的情形，即 N 个结点表示 N 个离散变量，每个变量都有 K 个状态。这种情况下的势函数 $\psi_{n-1,n}(x_{n-1}, x_n)$ 由一个 $K \times K$ 的表组成，因此联合概率分布有 $(N - 1)K^2$ 个参数。

让我们考虑寻找边缘概率分布 $p(x_n)$ 这一推断问题，其中 x_n 是链上的一个具体的结点。注意，现阶段，没有观测结点。根据定义，这个边缘概率分布可以通过对联合概率分布在除 x_n 以外的所有变量上进行求和的方式得到，即

$$p(x_n) = \sum_{x_1} \cdots \sum_{x_{n-1}} \sum_{x_{n+1}} \cdots \sum_{x_N} p(\mathbf{x}) \quad (8.50)$$

在一个朴素的实现中，我们首先计算联合概率分布，然后显式地进行求和。联合概率分布可以表示为一组数，对应于 \mathbf{x} 的每个可能的值。因为有 N 个变量，每个变量有 K 个可能的状态，因此 \mathbf{x} 有 K^N 个可能的值，从而联合概率的计算和存储以及得到 $p(x_n)$ 所需的求和过程，涉及到的存储量和计算量都会随着链的长度 N 而指数增长。

然而，通过利用图模型的条件独立性质，我们可以得到一个更加高效的算法。如果我们将联合概率分布的分解表达式 (8.49) 代入到公式 (8.50) 中，那么我们可以重新整理加和与乘积的顺序，使得需要求解的边缘概率分布可以更加高效地计算。例如，考虑对 x_N 的求和。势函数 $\psi_{N-1,N}(x_{N-1}, x_N)$ 是唯一与 x_N 有关系的势函数，因此我们可以进行下面的求和

$$\sum_{x_N} \psi_{N-1,N}(x_{N-1}, x_N) \quad (8.51)$$

得到一个关于 x_{N-1} 的函数。之后，我们可以使用它进行 x_{N-1} 上的求和，这涉及到这个新的函数以及势函数 $\psi_{N-2,N-1}(x_{N-2}, x_{N-1})$ ，因为这个势函数是唯一出现了 x_{N-1} 的地方。类似地， x_1 上的求和式只涉及到势函数 $\psi_{1,2}(x_1, x_2)$ ，因此可以单独进行，得到 x_2 的函数，以此类推。因为每个求和式都移除了概率分布中的一个变量，因此这可以被看成从图中移除一个结点。

如果我们使用这种方式对势函数和求和式进行分组，那么我们可以将需要求解的边缘概率密度写成下面的形式

$$p(x_n) = \frac{1}{Z} \underbrace{\left[\sum_{x_{n-1}} \psi_{n-1,n}(x_{n-1}, x_n) \cdots \left[\sum_{x_2} \psi_{2,3}(x_2, x_3) \left[\sum_{x_1} \psi_{1,2}(x_1, x_2) \right] \cdots \right] \cdots \right]}_{\mu_\alpha(x_n)} \quad (8.52)$$

$$\underbrace{\left[\sum_{x_{n+1}} \psi_{n,n+1}(x_n, x_{n+1}) \cdots \left[\sum_{x_N} \psi_{N-1,N}(x_{N-1}, x_N) \right] \cdots \right]}_{\mu_\beta(x_n)}$$

我们建议读者仔细研究这个重排序的方式，因为这背后的思想组成了后续对于一般的加和-乘积算法的讨论的基础。这里，我们利用的关键的概念是乘法对加法的分配率，即

$$ab + ac = a(b + c) \quad (8.53)$$

其中左侧涉及到三次算术计算而右侧将它简化成了两次计算。

让我们考察使用这种重排序的表达式之后，计算边缘概率分布所需的计算代价。我们必须进行 $N - 1$ 次求和，每次求和的对象是 K 个状态，并且每次求和涉及到两个变量组成的函数。例如，对 x_1 的求和只涉及到函数 $\psi_{1,2}(x_1, x_2)$ ，这是一个 $K \times K$ 的表格。对于每个 x_2 ，我们必须关于 x_1 对这个表进行求和，因此计算代价为 $O(K^2)$ 。得到的 K 个数字的向量与 $\psi_{2,3}(x_2, x_3)$ 的矩阵相乘，计算代价还是 $O(K^2)$ 。因为有 $N - 1$ 次这样的求和与乘积操作，因此计算边缘概率分布 $p(x_n)$ 的总代价是 $O(NK^2)$ 。这是链长度的一个线性函数，与朴素方法的指数代价不同。于是，我们已经能够利用这个简单图的许多条件独立性质来得到一个高效的计算方式。如果图是全连接的，那么将不存在条件独立性质，我们就必须直接计算完整的联合概率分布。

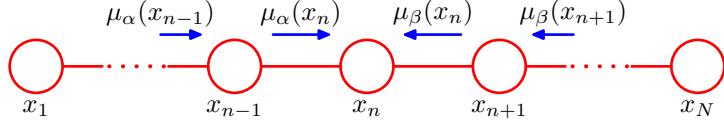


图 8.38: 对于结点链上的一个结点 x_n , 边缘概率分布可以通过下面的方式求得: 将两个信息 $\mu_\alpha(x_n)$ 和 $\mu_\beta(x_n)$ 相乘, 然后归一化。这些信息本身可以通过从结点链的两侧向结点 x_n 传递信息的方式递归地计算。

我们现在使用图中局部信息传递的思想, 给出这种计算的一个强大的直观意义。根据公式 (8.52), 我们看到边缘概率分布 $p(x_n)$ 的表达式分解成了两个因子的乘积乘以归一化常数

$$p(x_n) = \frac{1}{Z} \mu_\alpha(x_n) \mu_\beta(x_n) \quad (8.54)$$

我们把 $\mu_\alpha(x_n)$ 看成从结点 x_{n-1} 到结点 x_n 的沿着链向前传递的信息。类似地, $\mu_\beta(x_n)$ 可以看成从结点 x_{n+1} 到结点 x_n 的沿着链向后传递的信息。注意, 每条信息由 K 个值的集合构成, 每个值对应于 x_n 的一种选择, 因此两条信息的乘积可以被看做两条信息的元素之间的点积, 得到另外 K 个值的集合。

信息 $\mu_\alpha(x_n)$ 可以递归地计算, 因为

$$\begin{aligned} \mu_\alpha(x_n) &= \sum_{x_{n-1}} \psi_{n-1,n}(x_{n-1}, x_n) \left[\sum_{x_{n-2}} \dots \right] \\ &= \sum_{x_{n-1}} \psi_{n-1,n}(x_{n-1}, x_n) \mu_\alpha(x_{n-1}) \end{aligned} \quad (8.55)$$

因此我们首先计算

$$\mu_\alpha(x_2) = \sum_{x_1} \psi_{1,2}(x_1, x_2) \quad (8.56)$$

然后重复应用公式 (8.55) 直到我们到达需要求解的结点。注意一下信息传递方程的结构。公式 (8.55) 中的向外传播的信息 $\mu_\alpha(x_n)$ 通过下面的方式得到: 将输入信息 $\mu_\alpha(x_{n-1})$ 与涉及到结点变量与输出变量的势函数相乘, 然后对结点变量求和。

类似地, 信息 $\mu_\beta(x_n)$ 可以递归的计算。计算方法为: 从结点 x_N 开始, 使用

$$\begin{aligned} \mu_\beta(x_n) &= \sum_{x_{n+1}} \psi_{n,n+1}(x_n, x_{n+1}) \left[\sum_{x_{n+2}} \dots \right] \\ &= \sum_{x_{n+1}} \psi_{n,n+1}(x_n, x_{n+1}) \mu_\beta(x_{n+1}) \end{aligned} \quad (8.57)$$

这种递归的信息传递如图8.38所示。归一化常数 Z 很容易通过对公式 (8.54) 右侧关于 x_n 的所有状态求和的方式得到, 这只需要 $O(K)$ 次计算。

图8.38所示的图被称为马尔科夫链 (Markov chain), 对应的信息传递方程是马尔科夫过程的Chapman-Kolmogorov方程的一个例子 (Papoulis, 1984)。

现在假设我们将计算链中每个结点 $n \in \{1, \dots, N\}$ 的边缘概率分布 $p(x_n)$ 。简单地对每个结点单独地应用上面的步骤产生的计算代价为 $O(N^2 K^2)$ 。然而, 这种方法对于计算很浪费。例如, 为了得到 $p(x_1)$, 我们需要将信息 $\mu_\beta(\cdot)$ 从结点 x_N 传递到结点 x_2 。类似地, 为了计算 $p(x_2)$, 我们需要将信息 $\mu_\beta(\cdot)$ 从结点 x_N 传递到结点 x_3 。这涉及到许多重复的计算, 因为这两种情况下, 大部分信息都是相同的。

假设我们首先计算出结点 x_N 开始的信息 $\mu_\beta(x_{N-1})$, 然后将信息一路传递回结点 x_1 , 同时假设我们类似地计算出了从结点 x_1 开始的信息 $\mu_\alpha(x_2)$, 然后将信息一路向前传递到结点 x_N 。只要我们存储了所有的中间信息, 那么任何结点的边缘概率分布都可以通过使用公式 (8.54) 简单地计算出来。计算代价仅仅是找到一个结点的边缘概率分布的二倍, 而不是 N 倍。我们观察到,

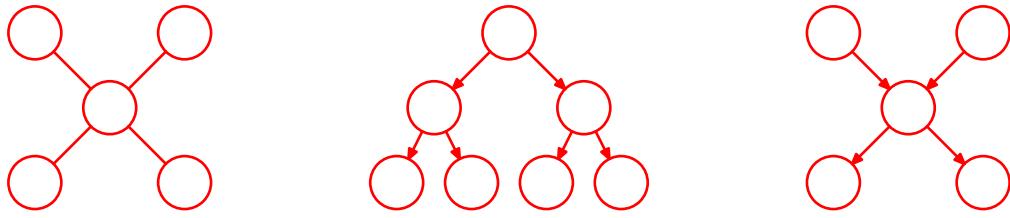


图 8.39: 三个树结构的例子。(a)一个无向树, (b)一个有向树, (c)一个有向多树。

一个信息沿着图中每个链接在每个方向上都传递了一次。还要注意，归一化常数 Z 只需计算一次，使用任何一个结点都可以计算。

如果图中的某些结点被观测到，那么对应的变量简单地被限制为观测值即可，不需要求和。为了说明这一点，我们注意到将变量 x_n 限制为一个观测值 \hat{x}_n 的效果可以表示为将联合概率分布乘以一个额外的函数 $I(x_n, \hat{x}_n)$ ，这个函数当 $x_n = \hat{x}_n$ 时取值为1，其他情况取值为0。这种函数可以被整合到包含 x_n 的势函数中。这样，对 x_n 的求和值包含 $x_n = \hat{x}_n$ 的一项。

现在假设我们项计算结点链中两个相邻结点的联合概率分布 $p(x_{n-1}, x_n)$ 。这类似于计算单一结点的边缘概率分布，区别在于现在有两个变量没有被求和出来。稍微思考一下，我们就会看到，需要求解的边缘概率分布可以写成下面的形式

$$p(x_{n-1}, x_n) = \frac{1}{Z} \mu_\alpha(x_{n-1}) \psi_{n-1,n}(x_{n-1}, x_n) \mu_\beta(x_n) \quad (8.58)$$

因此一旦我们完成了计算边缘概率分布所需的信息传递，我们就可以直接得到每个势函数中的所有变量上的联合概率分布。

这是一个很有用的结果，因为在实际应用中，我们可能希望使用团块势函数的参数形式，或者等价地，使用条件概率分布的参数形式（在有向图中）。为了在并非所有的变量都被观测到的情况下学习势函数的参数，我们可以使用EM算法。可以证明，以任意观测数据为条件，团块的局部联合概率分布恰好是E步骤中所需要的。我们会在第13章详细讨论一些例子。

8.4.2 树

我们已经看到，一个由结点链组成的图的精确推断可以在关于结点数量的线性时间内完成，方法是使用通过链中信息传递表示的算法。更一般地，通过局部信息在更大的一类图中的传递，我们可以高效地进行推断。这类图被称为树（tree）。特别地，我们会对之前在结点链的情形中得到的信息传递公式进行简单的推广，得到加和-乘积算法（sum-product algorithm），它为树结构图的精确推断提供了一个高效的框架。

在无向图的情形中，树被定义为满足下面性质的图：任意一对结点之间有且只有一条路径。于是这样的图没有环。在有向图的情形中，树的定义为：有一个没有父结点的结点，被称为根（root），其他所有的结点都有一个父结点。如果我们将有向树转化为无向图，我们会看到“伦理”步骤不会增加任何链接，因为所有的结点至多有一个父结点，从而对应的道德图是一个无向树。无向树和有向树的例子如图8.39(a)和8.39(b)所示。注意，一个表示为有向树的概率分布可以很容易地转化为一个由无向树表示的概率分布，反之亦然。

如果有向图中存在具有多个父结点的结点，但是在任意两个结点之间仍然只有一条路径（忽略箭头方向），那么这个图被称为多树（polytree），如图8.39(c)所示。这样的图中，存在多个没有父结点的结点，并且对应的道德无向图会存在环。

8.4.3 因子图

在下一节中我们将要推导的加和-乘积算法适用于无向树、有向树以及多树。如果我们首先引入一个新的图结构，被称为因子图（factor graph）（Frey, 1998; Kschischang et al., 2001），那么算法的形式会变得特别简单并且具有一般性。

有向图和无向图都使得若干个变量的一个全局函数能够表示为这些变量的子集上的因子的乘积。因子图显式地表示出了这个分解，方法是：在表示变量的结点的基础上，引入额外的结点表示因子本身。因子图也使我们能够更加清晰地了解分解的细节，正如我们将看到的那样。

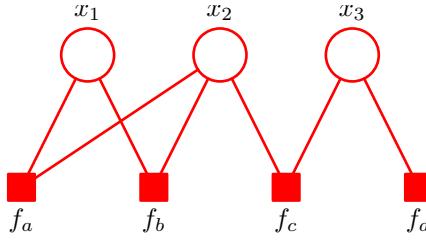


图 8.40: 因子图的例子, 对应于公式 (8.60) 的分解。

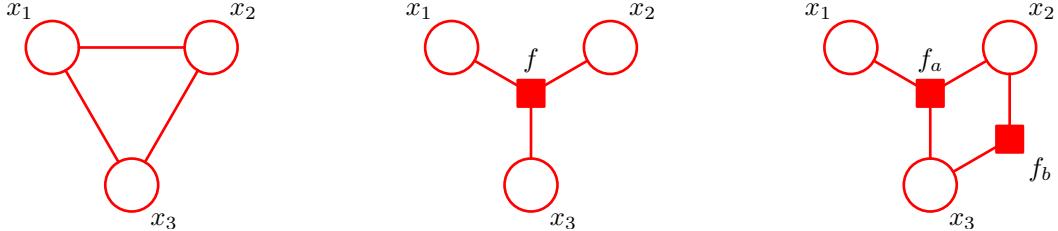


图 8.41: (a) 一个无向图, 有一个单一的团块势函数 $\psi(x_1, x_2, x_3)$ 。 (b) 一个因子图, 因子 $f(x_1, x_2, x_3) = \psi(x_1, x_2, x_3)$, 它表示与无向图相同的概率分布。 (c) 一个不同的因子图, 表示相同的概率分布, 它的因子满足 $f_a(x_1, x_2, x_3)f_b(x_2, x_3) = \psi(x_1, x_2, x_3)$ 。

让我们将一组变量上的联合概率分布写成因子的乘积形式

$$p(\mathbf{x}) = \prod_s f_s(\mathbf{x}_s) \quad (8.59)$$

其中 \mathbf{x}_s 表示变量的一个子集。为了方便, 我们把单独的变量记作 x_i , 但是在之前的讨论中, 这可以由一组变量组成 (例如向量或矩阵)。每个因子 f_s 是对应的变量集合 \mathbf{x}_s 的函数。

有向图的分解由公式 (8.5) 定义, 表示公式 (8.59) 的特殊情况, 即因子 $f_s(\mathbf{x}_s)$ 是局部条件概率分布。类似地, 公式 (8.39) 给出的无向图的分解, 也是一个特例, 即因子是最大团块上的势函数 (归一化系数 $\frac{1}{Z}$ 可以被看做定义在空变量集合上的因子)。

在因子图中, 概率分布中的每个变量都有一个结点 (与之前一样, 用圆圈表示), 这与有向图和无向图的情形相同。还存在其他的结点 (用小正方形表示), 表示联合概率分布中的每个因子 $f_s(\mathbf{x}_s)$ 。最后, 在每个因子结点和因子所依赖的变量结点之间, 存在无向链接。例如, 考虑一个表示为因子图形式的概率分布

$$p(\mathbf{x}) = f_a(x_1, x_2)f_b(x_1, x_2)f_c(x_2, x_3)f_d(x_3) \quad (8.60)$$

这可以表示为图 8.40 所示的因子图。注意有两个因子 $f_a(x_1, x_2)$ 和 $f_b(x_1, x_2)$ 定义在同一个变量集合上。在一个无向图中, 两个这样的因子的乘积被简单地合并到同一个团块势函数中。类似地, $f_c(x_2, x_3)$ 和 $f_d(x_3)$ 可以结合到 x_2 和 x_3 上的一个单一势函数中。然而, 因子图显式地写出这些因子, 因此能够表达出关于分解本身的信息。

由于因子图由两类不同的结点组成, 且所有的链接都位于两类不同的结点之间, 因此因子图被称为二分的 (bipartite)。于是, 因子图通常总可以被画成两排结点 (变量结点在上排, 因子结点在下排), 同时两排结点之间具有链接, 如图 8.40 所示。然而, 在某些情况下, 其他的表示因子图的方式可能更符合直觉, 例如因子图从有向图或者无向图中推导出的情形, 正如我们将要看到的那样。

如果我们有一个通过无向图表示的概率分布, 那么我们可以将其转化为因子图。为了完成这一点, 我们构造变量结点, 对应于原始无向图, 然后构造额外的因子结点, 对应于最大团块 \mathbf{x}_s 。因子 $f_s(\mathbf{x}_s)$ 被设置为与团块势函数相等。注意, 对于同一个无向图, 可能存在几个不同的因子图。图 8.41 说明了这些概念。

类似地, 为了将有向图转化为因子图, 我们构造变量结点对应于有向图中的结点, 然后构造因子结点, 对应于条件概率分布, 最后添加上合适的链接。与之前一样, 同一个有向图可能对应于多个因子图。有向图到因子图的转化如图 8.42 所示。

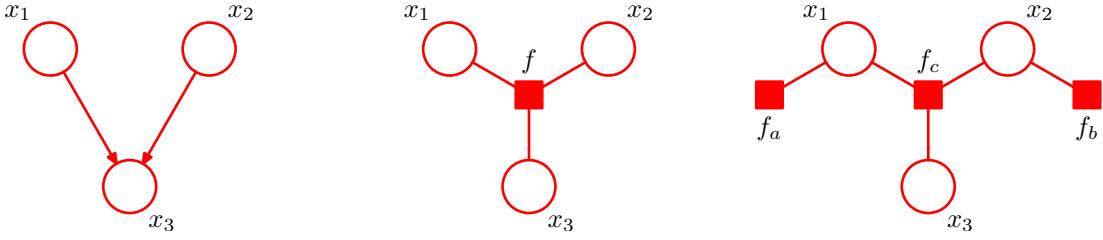


图 8.42: (a)一个有向图, 可以分解为 $p(x_1)p(x_2)p(x_3 | x_1, x_2)$ 。 (b)一个因子图, 表示与有向图相同的概率分布, 它的因子满足 $f(x_1, x_2, x_3) = p(x_1)p(x_2)p(x_3 | x_1, x_2)$ 。 (c)一个不同的因子图, 表示同样的概率分布, 因子为 $f_a(x_1) = p(x_1)$, $f_b(x_2) = p(x_2)$, $f_c(x_1, x_2, x_3) = p(x_3 | x_1, x_2)$ 。

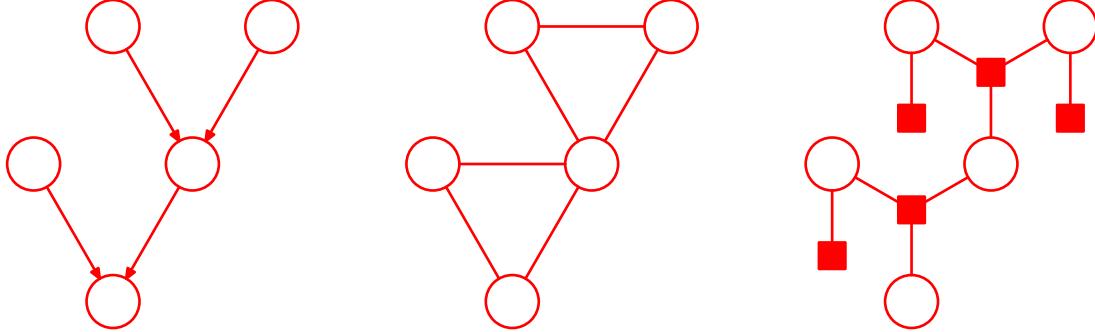


图 8.43: (a)一个有向多树。 (b)将多树转化为无向图的结果, 展示了环的形成。 (c)将多树转化为因子图的结果, 保留了树形结构。

我们已经看到了树结构图对于进行高效推断的重要性。如果我们将一个有向树或者无向树转化为因子图, 那么生成的因子图也是树 (即, 因子图没有环, 且任意两个结点之间有且只有一条路径)。在有向多树的情形中, 由于“伦理”步骤的存在, 转化为无向图会引入环, 而转化后的因子图仍然是树, 如图8.43所示。事实上, 有向图中由于链接父结点和子结点产生的局部环可以在转换到因子图时被移除, 只需定义合适的因子函数即可, 如图8.44所示。

我们已经看到多个不同的因子图可以表示同一个有向图或者无向图。这使得因子图对于分解的精确形式的表示更加具体。图8.45给出了一个全连接的无向图以及两个不同的因子图的例子。在图(b)中, 联合概率分布是一般形式 $p(\mathbf{x}) = f(x_1, x_2, x_3)$, 而在图(c)中, 它表示为一个更加具体的分解方式 $p(\mathbf{x}) = f_a(x_1, x_2)f_b(x_1, x_3)f_c(x_2, x_3)$ 。应该强调的是, (c)中的分解不对应于任何条件独立性质。

8.4.4 加和-乘积算法

我们会使用因子图框架推导一类强大的、高效的精确推断算法, 这些算法适用于树结构的图。这里, 我们把注意力集中于计算结点或者结点子集上的局部边缘概率分布, 这会引出加和-乘积算法 (sum-product algorithm)。稍后, 我们会修改这个方法, 使得概率最大的状态被找到, 这就引出了最大加和算法 (max-sum algorithm)。

此外, 我们假设模型中所有的变量都是离散的, 因此求边缘概率对应于求和的过程。然而,



图 8.44: (a)具有局部环的有向图的片段。 (b)转化得到的因子图的片段, 具有树形结构, 其中 $f(x_1, x_2, x_3) = p(x_1)p(x_2 | x_1)p(x_3 | x_1, x_2)$ 。

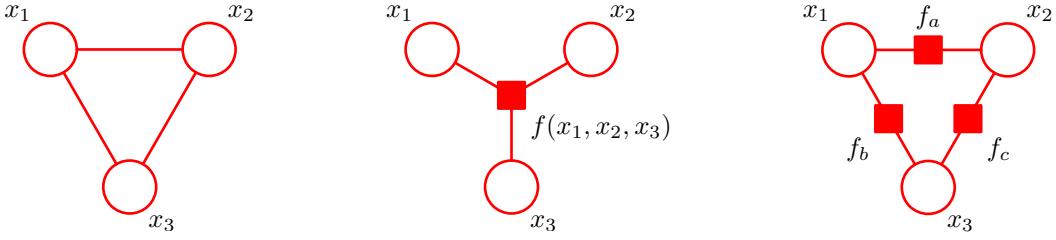


图 8.45: (a) 表示一个完全连接的无向图。(b) 和 (c) 表示两个因子图，每个因子图都对应于(a)中的无向图。

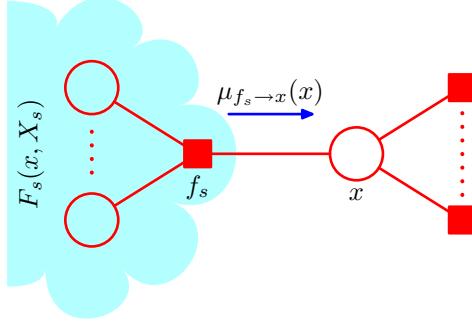


图 8.46: 因子图的片段，说明了边缘概率分布 $p(x)$ 的计算。

这个框架同样适用于线性高斯模型，这种情形下求边缘概率涉及到求积分。当我们讨论线性动态系统时，我们会详细讨论这种情形。

关于有向无环图的精确推断，有一个被称为置信传播（belief propagation）的算法（Pearl, 1988; Lauritzen and Spiegelhalter, 1988），它等价于加和-乘积算法的一个具体情形。这里，我们只考虑加和-乘积算法，因为它的推导和使用都更容易，并且更一般。

我们假设原始的图是一个无向树或者有向树或者多树，从而对应的因子图有一个树结构。首先，我们将原始的图转化为因子图，使得我们可以使用同样的框架处理有向模型和无向模型。我们的目标是利用图的结构完成两件事：(1) 得到一个高效的精确推断算法来寻找边缘概率，(2) 在需要求解多个边缘概率的情形，计算可以高效地共享。

首先，对于特定的变量结点 x ，我们寻找边缘概率 $p(x)$ 。现阶段，我们假设所有的变量都是隐含变量。稍后我们会看到如何修改这个算法，使得观测变量被整合到算法中。根据定义，边缘概率分布通过对所有 x 之外的变量上的联合概率分布进行求和的方式得到，即

$$p(x) = \sum_{\mathbf{x} \setminus x} p(\mathbf{x}) \quad (8.61)$$

其中 $\mathbf{x} \setminus x$ 表示变量 x 的集合去掉变量 x 。算法的思想是使用因子图的表达式 (8.59) 替换 $p(\mathbf{x})$ ，然后交换加和与乘积的顺序，得到一个高效的算法。考虑图 8.46 给出的图，我们看到图的树结构使得我们可以将联合概率分布中的因子划分为若干组，每组对应于变量结点 x 的相邻结点组成的因子结点集合。我们看到联合概率分布可以写成乘积的形式

$$p(\mathbf{x}) = \prod_{s \in \text{ne}(x)} F_s(x, X_s) \quad (8.62)$$

其中 $\text{ne}(x)$ 表示与 x 相邻的因子结点的集合， X_s 表示子树中通过因子结点 f_s 与变量结点 x 相连的所有变量的集合， $F_s(x, X_s)$ 表示分组中与因子 f_s 相关联的所有因子的乘积。

将公式 (8.62) 代入 (8.61)，交换加和与乘积的顺序，我们有

$$\begin{aligned} p(x) &= \prod_{s \in \text{ne}(x)} \left[\sum_{X_s} F_s(x, X_s) \right] \\ &= \prod_{s \in \text{ne}(x)} \mu_{f_s \rightarrow x}(x) \end{aligned} \quad (8.63)$$

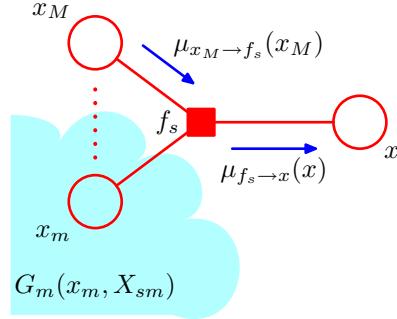


图 8.47: 与因子结点 f_s 关联的子图的分解。

这里我们引入了一组新的函数 $\mu_{f_s \rightarrow x}(x)$, 定义为

$$\mu_{f_s \rightarrow x}(x) \equiv \sum_{X_s} F_s(x, X_s) \quad (8.64)$$

这可以被看做从因子结点 f_s 到变量结点 x 的信息 (message)。我们看到, 需要求解的边缘概率分布 $p(x)$ 等于所有到达结点 x 的输入信息的乘积。

为了计算这些信息, 我们再次回到图8.46。我们注意到每个因子 $F_s(x, X_s)$ 由一个因子图 (因子子图), 因此本身可以被分解。特别地, 我们有

$$F_s(x, X_s) = f_s(x, x_1, \dots, x_M) G_1(x_1, X_{s1}) \dots G_M(x_M, X_{sM}) \quad (8.65)$$

其中, 为了方便, 我们将 x 之外的与因子 f_s 相关的变量记作 x_1, \dots, x_M 。图8.47说明了这个分解过程。注意变量集合 $\{x, x_1, \dots, x_M\}$ 是因子 f_s 依赖的变量的集合, 因此使用公式 (8.59) 的记号, 它也可以被记作 x_s 。

将公式 (8.65) 代入公式 (8.64) , 我们有

$$\begin{aligned} \mu_{f_s \rightarrow x}(x) &= \sum_{x_1} \dots \sum_{x_M} f_s(x, x_1, \dots, x_M) \prod_{m \in \text{ne}(f_s) \setminus x} \left[\sum_{X_{sm}} G_m(x_m, X_{sm}) \right] \\ &= \sum_{x_1} \dots \sum_{x_M} f_s(x, x_1, \dots, x_M) \prod_{m \in \text{ne}(f_s) \setminus x} \mu_{x_m \rightarrow f_s}(x_m) \end{aligned} \quad (8.66)$$

其中 $\text{ne}(f_s)$ 表示因子结点 f_s 的相邻变量结点的集合, $\text{ne}(f_s) \setminus x$ 表示同样的集合, 但是移除了结点 x 。这里, 我们定义了下面的从变量结点到因子结点的信息

$$\mu_{x_m \rightarrow f_s}(x_m) \equiv \sum_{X_{sm}} G_m(x_m, X_{sm}) \quad (8.67)$$

于是, 我们引入了两类不同的信息。一类信息是从因子结点到变量结点的信息, 记作 $\mu_{f \rightarrow x}(x)$, 另一类信息是从变量结点到因子结点的信息, 记作 $\mu_{x \rightarrow f}(x)$ 。在任何一种情况下, 我们看到沿着一条链接传递的信息总是一个函数, 这个函数是与那个链接相连的变量结点相关的变量的函数。

公式 (8.66) 给出的结果表明, 一个变量结点通过一个链接发送到一个因子结点的信息可以按照如下的方式计算: 计算沿着所有进入因子结点的其他链接的输入信息的乘积, 乘以与那个结点关联的因子, 然后对所有与输入信息关联的变量进行求和。如图8.47所示。值得注意的是, 一旦一个因子结点从所有其他的相邻变量结点的输入信息, 那么这个因子结点就可以向变量结点发送信息。

最后, 我们推导变量结点到因子结点的信息的表达式, 再次使用图分解 (子图分解)。根据图8.48, 我们看到与结点 x_m 关联的项 $G_m(x_m, X_{sm})$ 由项 $F_l(x_m, X_{lm})$ 的乘积组成, 每一个这样的项都与连接到结点 x_m 的一个因子结点 f_l 相关联 (不包括结点 f_s) , 即

$$G_m(x_m, X_{sm}) = \prod_{l \in \text{ne}(x_m) \setminus f_s} F_l(x_m, X_{lm}) \quad (8.68)$$

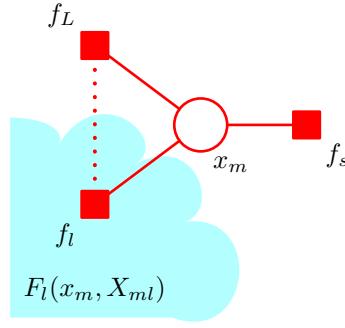


图 8.48: 由一个变量结点向一个相邻因子结点发送的信息的计算。



图 8.49: 加和-乘积算法的开始阶段是从叶结点发送信息，信息取决于叶结点是(a)变量结点，或者(b)因子结点。

其中求乘积的对象是结点 x_m 的所有相邻结点，排除结点 f_s 。注意，每个因子 $F_l(x_m, X_{lm})$ 表示原始图的一个子树，这个原始图与公式 (8.62) 表示的图的形式完全相同。将公式 (8.68) 代入 (8.67)，我们可以得到

$$\begin{aligned}\mu_{x_m \rightarrow f_s}(x_m) &= \prod_{l \in \text{ne}(x_m) \setminus f_s} \left[\sum_{X_{lm}} F_l(x_m, X_{lm}) \right] \\ &= \sum_{l \in \text{ne}(x_m) \setminus f_s} \mu_{f_l \rightarrow x_m}(x_m)\end{aligned}\quad (8.69)$$

其中我们使用了因子结点到变量结点的信息传递的表达式 (8.64)。因此，为了计算从一个变量结点到相邻因子结点沿着链接传递的信息，我们只需简单地在其他所有结点上对输入信息取乘积。注意，任何只有两个相邻结点的变量结点无需参与计算，只需将信息不变地传递过去即可。此外，我们注意到，一旦一个变量结点接收到了来自所有其他相邻因子结点的输入信息，那么这个变量结点就可以给因子结点发送信息。

回忆一下，我们的目标是计算变量结点 x 的边缘概率分布，这个边缘概率分布等于沿着所有到达这个结点的链接的输入信息的乘积。这些信息中的每一条信息都可以使用其他的信息递归地计算。为了开始这个递归计算的过程，我们可以将结点 x 看成树的根结点，然后从叶结点开始计算。根据公式 (8.69) 的定义，我们看到如果一个叶结点是一个变量结点，那么它沿着与它唯一相连的链接发送的信息为

$$\mu_{x \rightarrow f}(x) = 1 \quad (8.70)$$

如图8.49(a)所示。类似地，如果叶结点是一个因子结点，那么我们根据公式 (8.66) 可以看到，发送的信息的形式为

$$\mu_{f \rightarrow x}(x) = f(x) \quad (8.71)$$

如图8.49(b)所示。

现在，让我们停下来，总结一下计算边缘概率分布 $p(x)$ 时得到的加和-乘积算法。首先，我们将变量结点 x 看成因子图的根结点，使用公式 (8.70) 和公式 (8.71)，初始化图的叶结点的信息。之后，递归地应用信息传递步骤 (8.66) 和 (8.69)，直到信息被沿着每一个链接传递完毕，并且根结点收到了所有相邻结点的信息。每个结点都可以向根结点发送信息。一旦结点收到了所有其他相邻结点的信息，那么它就可以向根结点发送信息。一旦根结点收到了所有相邻结点的信息，需要求解的边缘概率分布就可以使用公式 (8.63) 进行计算。我们稍后会说明这个过程。

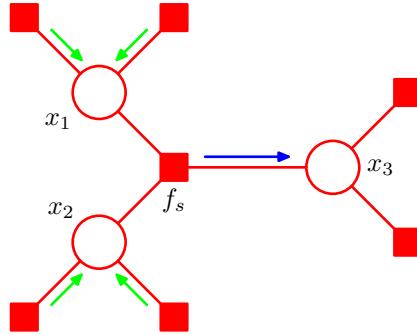


图 8.50: 加和-乘积算法可以被看做纯粹的因子结点与其他因子结点之间的信息传递。在这个例子中，蓝色箭头表示的输出信息可以这样计算：对所有绿色箭头表示的输入信息求乘积，然后乘以因子 f_s ，然后在变量 x_1 和 x_2 上求和或积分。

为了说明每个结点总会收到足够的信息来使得发送信息变得可能，我们可以使用归纳法简单地说明如下。很明显，对于一个由变量根结点直接与几个因子叶结点相连的图，算法仅仅涉及到直接从叶结点向根结点发送形如 (8.71) 的信息。现在，假设通过每次添加一个结点的方式构建一个一般的图，并且假设对于某个特定的图，我们有一个合法的算法。当添加了一个更多的结点（变量结点或因子结点）之后，这个结点只能通过一个单一的链接与图相连，因为整体的图必须仍然是树，因此新结点是一个叶结点。于是，这个结点向它连接的结点发送一个信息，反过来会收到为了将自己的信息送往根结点所需的所有信息，因此与之前一样，我们得到了一个合法的算法，从而完成了证明。

现在假设我们想寻找图中每个变量结点的边缘概率分布。这可以通过简单地对每个结点独立地运行上述算法的方式完成。然而，这会相当浪费计算结果，因为许多需要进行的计算被重复了多次。通过“叠加”多个信息传递算法，我们可以得到一个更加高效的步骤，从而得到一般的加和-乘积算法，如下所述。任意选择一个结点（变量结点或因子结点），然后将其指定为根结点。像之前一样，我们从叶结点向根结点传递信息。现在，根结点会接收到来自所有相邻结点的信息。因此，它可以向所有的相邻结点发送信息。反过来，这些结点之后会接收到来自所有相邻结点的信息，因此可以沿着远离根结点的链接发送出信息，以此类推。通过这种方式，信息可以从根结点向外传递到叶结点。现在，信息已经在两个方向上沿着图中所有的链接传递完毕，并且每个结点都已经接收到了来自所有相邻结点的信息。与之前一样，可以使用一个简单的归纳过程验证信息传递协议的合法性。因为每个变量结点会收到来自所有相邻结点的信息，所以我们可以计算图中每个变量的边缘概率分布。必须计算的信息的数量等于图中链接数量的二倍，因此所需的计算量仅仅是计算一个边缘概率分布的二倍。作为对比，如果我们对每个结点分别运行加和-乘积算法，那么计算量会随着图的规模以二次函数的形式增长。注意，这个算法实际上与哪个结点被选择为根结点无关。事实上，引入一个具有具体状态的结点仅仅是为了便于解释信息传递协议。

接下来，假设我们想找到边缘概率分布 $p(\mathbf{x}_s)$ ，它与属于每个因子的变量集合相关联。通过一个与之前类似的讨论，很容易看到与某个因子关联的边缘概率分布为到达这个因子结点的信息与这个结点的局部因子的乘积，即

$$p(\mathbf{x}_s) = f_s(\mathbf{x}_s) \prod_{i \in \text{ne}(f_s)} \mu_{x_i \rightarrow f_s}(x_i) \quad (8.72)$$

这与变量结点的边缘概率分布十分相似。如果因子是参数化的函数，我们希望使用EM算法学习到参数的值，那么这些边缘概率分布恰好就是我们在E步骤中需要计算的值，正如我们在第13章讨论隐马尔科夫模型时将要看到的那样。

正如我们已经看到的那样，一个变量结点发送到一个因子结点的信息仅仅其他链接上的输入信息的乘积。如果必要的话，我们可以用一个稍微不同的形式考查加和-乘积算法，即消去从变量结点到因子结点的信息，仅考虑由因子结点发送出的信息。考虑图8.50中的简单例子，我们可以很容易地看出这一点。

目前为止，我们始终忽略了归一化系数的问题。如果因子图是从有向图推导的，那么联合概率分布已经正确地被归一化了，因此通过加和-乘积算法得到的边缘概率分布会类似地被正确归

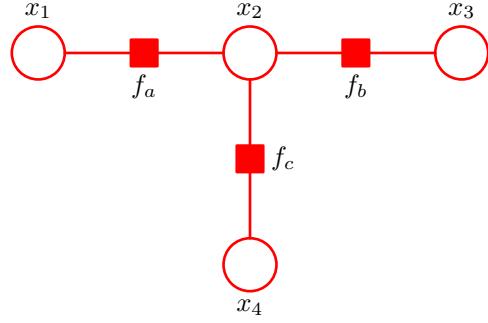


图 8.51: 一个简单的因子图, 用来说明加和-乘积算法。

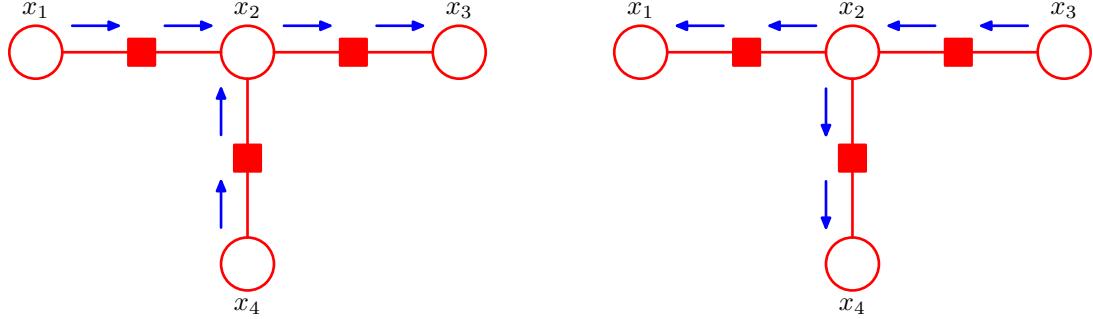


图 8.52: 应用于图8.51给出的图的加和-乘积算法的信息流。(a)从叶结点 x_1 和 x_4 向根结点 x_3 传递。(b)从根结点向叶结点传递。

一化。然而, 如果我们开始于一个无向图, 那么通常会存在一个未知的归一化系数 $\frac{1}{Z}$ 。与图8.38给出的简单例子相同, 通过对未归一化的联合概率 $\tilde{p}(\mathbf{x})$ 进行操作, 这个问题可以很容易处理, 其中 $p(\mathbf{x}) = \frac{\tilde{p}(\mathbf{x})}{Z}$ 。首先, 我们运行加和-乘积算法找到对应的未归一化的边缘概率分布 $\tilde{p}(x_i)$ 。之后, 系数 $\frac{1}{Z}$ 可以很容易地通过对任意一个边缘概率分布进行归一化的方式得到。这种计算很高效, 因为归一化是在单一变量上进行的, 而不是在整个变量集合上进行。如果在整个变量集合上进行, 那么我们就需要直接归一化 $\tilde{p}(\mathbf{x})$ 。

现在, 考虑一个简单的例子来说明加和-乘积算法是很有帮助的。图8.51给出了一个简单的4节点因子图, 它的未归一化联合概率分布为

$$\tilde{p}(\mathbf{x}) = f_a(x_1, x_2)f_b(x_2, x_3)f_c(x_2, x_4) \quad (8.73)$$

为了对这个图应用加和-乘积算法, 让我们令结点 x_3 为根结点, 此时有两个叶结点 x_1 和 x_4 。从叶结点开始, 我们有下面六个信息组成的序列。

$$\mu_{x_1 \rightarrow f_a}(x_1) = 1 \quad (8.74)$$

$$\mu_{f_a \rightarrow x_2}(x_2) = \sum_{x_1} f_a(x_1, x_2) \quad (8.75)$$

$$\mu_{x_4 \rightarrow f_c}(x_4) = 1 \quad (8.76)$$

$$\mu_{f_c \rightarrow x_2}(x_2) = \sum_{x_4} f_c(x_2, x_4) \quad (8.77)$$

$$\mu_{x_2 \rightarrow f_b}(x_2) = \mu_{f_a \rightarrow x_2}(x_2)\mu_{f_c \rightarrow x_2}(x_2) \quad (8.78)$$

$$\mu_{f_b \rightarrow x_3}(x_3) = \sum_{x_2} f_b(x_2, x_3)\mu_{x_2 \rightarrow f_b}(x_2) \quad (8.79)$$

信息流的方向如图8.52所示。一旦信息传播完成, 我们就可以将信息从根结点传递到叶结点, 这些信息为

$$\mu_{x_3 \rightarrow f_b}(x_3) = 1 \quad (8.80)$$

$$\mu_{f_b \rightarrow x_2}(x_2) = \sum_{x_3} f_b(x_2, x_3) \quad (8.81)$$

$$\mu_{x_2 \rightarrow f_a}(x_2) = \mu_{f_b \rightarrow x_2}(x_2) \mu_{f_c \rightarrow x_2}(x_2) \quad (8.82)$$

$$\mu_{f_a \rightarrow x_1}(x_1) = \sum_{x_2} f_a(x_1, x_2) \mu_{x_2 \rightarrow f_a}(x_2) \quad (8.83)$$

$$\mu_{x_2 \rightarrow f_c}(x_2) = \mu_{f_a \rightarrow x_2}(x_2) \mu_{f_b \rightarrow x_2}(x_2) \quad (8.84)$$

$$\mu_{f_c \rightarrow x_4}(x_4) = \sum_{x_2} f_c(x_2, x_4) \mu_{x_2 \rightarrow f_c}(x_2) \quad (8.85)$$

现在一个信息已经在两个方向上通过了每个链接，因此我们现在可以计算边缘概率分布。作为一个简单的检验，让我们验证边缘概率分布 $p(x_2)$ 由正确的表达式给出。使用公式 (8.63)，使用上面的结果将信息替换掉，我们有

$$\begin{aligned} \tilde{p}(x_2) &= \mu_{f_a \rightarrow x_2}(x_2) \mu_{f_b \rightarrow x_2}(x_2) \mu_{f_c \rightarrow x_2}(x_2) \\ &= \left[\sum_{x_1} f_a(x_1, x_2) \right] \left[\sum_{x_3} f_b(x_2, x_3) \right] \left[\sum_{x_4} f_c(x_2, x_4) \right] \\ &= \sum_{x_1} \sum_{x_3} \sum_{x_4} f_a(x_1, x_2) f_b(x_2, x_3) f_c(x_2, x_4) \\ &= \sum_{x_1} \sum_{x_3} \sum_{x_4} \tilde{p}(\mathbf{x}) \end{aligned} \quad (8.86)$$

这与我们预期的结果相同。

目前为止，我们已经假定图中所有的变量都是隐含变量。在大多数实际应用中，变量的一个子集会被观测到，我们希望计算以这些观测为条件的后验概率分布。观测结点在加和-乘积算法中很容易处理，如下所述。假设我们将 \mathbf{x} 划分为隐含变量 \mathbf{h} 和观测变量 \mathbf{v} ，且 \mathbf{v} 的观测值被记作 $\hat{\mathbf{v}}$ 。然后，我们简单地将联合概率分布 $p(\mathbf{x})$ 乘以 $\prod_i I(v_i, \hat{v}_i)$ ，其中如果 $v = \hat{v}$ ，则 $I(v, \hat{v}) = 1$ ，否则 $I(v, \hat{v}) = 0$ 。这个乘积对应于 $p(\mathbf{h}, \mathbf{v} = \hat{\mathbf{v}})$ ，因此是 $p(\mathbf{h} | \mathbf{v} = \hat{\mathbf{v}})$ 的一个未归一化版本。通过运行加和-乘积算法，我们可以高效地计算后验边缘概率 $p(h_i | \mathbf{v} = \hat{\mathbf{v}})$ ，忽略归一化系数。归一化系数的值可以使用一个局部的计算高效地计算出来。 \mathbf{v} 中变量上的任意求和式就退化成了单一的项。

我们在本节中一直假设我们处理的是离散变量。然而，无论是加和-乘积算法的图框架，还是算法的概率构建，方法都不局限于离散变量。对于连续变量，求和只需简单地替换为积分。当我们考虑线性动态系统时，我们会给出将加和-乘积算法应用于线性高斯变量的图结构中的例子。

8.4.5 最大加和算法

加和-乘积算法使得我们能够将联合概率分布 $p(\mathbf{x})$ 表示为一个因子图，并且高效地求出成分变量上的边缘概率分布。有两个其他的比较常见的任务，即找到变量的具有最大概率的一个设置，以及找到这个概率的值。这两个任务可以通过一个密切相关的算法完成，这个算法被称为最大加和 (max-sum)，可以被看成动态规划 (dynamic programming) 在图模型中的一个应用 (Cormen et al., 2001)。

一个简单的寻找具有最大概率的潜在变量值的方法是，运行加和-乘积算法，得到每个变量的边缘概率分布 $p(x_i)$ ，然后，反过来对于每个边缘概率分布，找到使边缘概率最大的 x_i^* 。然而，这回给出一组值，每个值都单独取得最大的概率。在实际应用中，我们通常希望找到联合起来具有最大概率的值的集合，换句话说，找到向量 $\mathbf{x}^{\text{最大}}$ ，使得联合概率分布达到最大值，即

$$\mathbf{x}^{\text{最大}} = \arg \max_{\mathbf{x}} p(\mathbf{x}) \quad (8.87)$$

这样，联合概率分布的对应值为

$$p(\mathbf{x}^{\text{最大}}) = \max_{\mathbf{x}} p(\mathbf{x}) \quad (8.88)$$

	$x = 0$	$x = 1$
$y = 0$	0.3	0.4
$y = 1$	0.3	0.0

表 8.1: 两个二值变量上的联合概率分布，其中联合概率分布的最大值出现的位置与两个边缘概率分布的最大值出现的位置不同。

通常， \mathbf{x} 最大与 x_i^* 的集合不同，我们会用一个简单的例子说明这一点。考虑两个二值变量 $x, y \in \{0, 1\}$ 上的联合概率分布 $p(x, y)$ ，由表 8.1 给出。通过令 $x = 1$ 以及 $y = 0$ ，联合概率分布被最大化，值为 0.4。然而，通过对 y 的值进行求和得到的 $p(x)$ 的边缘概率分布为 $p(x = 0) = 0.6$ 和 $p(x = 1) = 0.4$ ，类似地， y 的边缘概率分布为 $p(y = 0) = 0.7$ 和 $p(y = 1) = 0.3$ ，因此当 $x = 0$ 且 $y = 0$ 时，边缘概率分布取最大值，此时联合概率分布的值为 0.3。事实上，不难构造出这样的例子：各自的概率最大的值在联合概率分布下的概率为零。

于是，我们寻找一个高效的算法，来求出最大化联合概率分布 $p(\mathbf{x})$ 的 \mathbf{x} 的值，这会使得我们得到在最大值处的联合概率分布的值。为了解决第二个问题，我们只需简单地写出分量的最大值算符，即

$$\max_{\mathbf{x}} p(\mathbf{x}) = \max_{x_1} \dots \max_{x_M} p(\mathbf{x}) \quad (8.89)$$

其中 M 是变量的总数。之后，使用 $p(\mathbf{x})$ 的因子乘积形式表示的展开式替换 $p(\mathbf{x})$ 即可。在推导加和-乘积算法时，我们使用了乘法的分配律（8.53）。这里，我们使用最大化算符的类似定律

$$\max(ab, ac) = a \max(b, c) \quad (8.90)$$

这对于 $a \geq 0$ 的情形成立（这对于图模型的因子总成立）。这使得我们交换乘积与最大化的顺序。

首先考虑公式（8.49）描述的结点链这一简单的例子。概率最大值的计算可以写成

$$\begin{aligned} \max_{\mathbf{x}} p(\mathbf{x}) &= \frac{1}{Z} \max_{x_1} \dots \max_{x_N} [\psi_{1,2}(x_1, x_2) \dots \psi_{N-1,N}(x_{N-1}, x_N)] \\ &= \frac{1}{Z} \max_{x_1} \left[\max_{x_2} \left[\psi_{1,2}(x_1, x_2) \left[\dots \max_{x_N} \psi_{N-1,N}(x_{N-1}, x_N) \right] \dots \right] \right] \end{aligned}$$

正如边缘概率的计算一样，我们看到交换最大值算符和乘积算法会产生一个更高效的计算，并且更容易表示为从结点 x_N 沿着结点链传递回结点 x_1 的信息。

我们可以将这个结果推广到任意树结构的因子图上，推广的方法为：将因子图表达式（8.59）代入公式（8.89）中，然后交换乘积与最大化的计算顺序。这种计算的结构与加和-乘积算法完全相同，因此我们能够简单地将那些结果转化为当前的问题中。特别地，假设我们令图中的一个特定的变量结点为根结点。之后，我们计算起始的一组信息，然后从树的叶结点向内部传递到根结点。对于每个结点，一旦它接收到来自其他相邻结点的输入信息，那么它就向根结点发送信息。最后对所有到达根结点的信息的乘积进行最大化，得出 $p(\mathbf{x})$ 的最大值。这可以被称为最大化乘积算法（max-produce algorithm），与加和-乘积算法完全相同，唯一的区别是求和被替换为了求最大值。注意，现阶段，信息被从叶结点发送到根结点，而没有相反的方向。

在实际应用中，许多小概率的乘积可以产生数值下溢的问题，因此更方便的做法是对联合概率分布的对数进行操作。对数函数是一个单调函数，因此如果 $a > b$ ，那么 $\ln a > \ln b$ ，因此求最大值的运算符可以与取对数的运算交换顺序，即

$$\ln \left(\max_{\mathbf{x}} p(\mathbf{x}) \right) = \max_{\mathbf{x}} \ln p(\mathbf{x}) \quad (8.91)$$

分配性质仍然成立，因为

$$\max(a + b, a + c) = a + \max(b, c) \quad (8.92)$$

所以取对数的唯一效果是把最大化乘积算法中的乘积替换成了加和，因此我们得到了最大化加和算法（max-sum algorithm）。根据之前在加和-乘积算法中得到的公式（8.66）和公式（8.69）给出的结果，我们可以基于信息传递写出最大化加和算法，只需把“加和”替换为“最大化”，把“乘积”替换为对数求和即可。结果为

$$\mu_{f \rightarrow x}(x) = \max_{x_1, \dots, x_M} \left[\ln f(x, x_1, \dots, x_M) + \sum_{m \in \text{ne}(f) \setminus x} \mu_{x_m \rightarrow f}(x_m) \right] \quad (8.93)$$

$$\mu_{x \rightarrow f}(x) = \sum_{l \in \text{ne}(x) \setminus f} \mu_{f_l \rightarrow x}(x) \quad (8.94)$$

最开始的由叶结点发送的信息可以通过类比公式（8.70）和公式（8.71）得到，结果为

$$\mu_{x \rightarrow f}(x) = 0 \quad (8.95)$$

$$\mu_{f \rightarrow x}(x) = \ln f(x) \quad (8.96)$$

而在根结点处的最大概率可以通过类比公式（8.63）得到，结果为

$$p^{\text{最大}} = \max_x \left[\sum_{s \in \text{ne}(x)} \mu_{f_s \rightarrow x}(x) \right] \quad (8.97)$$

目前为止，我们已经看到了如何通过从叶结点到任意选择的根结点传递信息的方式找到联合概率分布的最大值。这个结果与根结点的选择无关。现在，我们转向第二个问题，即寻找联合概率达到最大值的变量的配置。目前，我们已经将信息从叶结点发送到了根结点。计算公式（8.97）的过程也会得到根结点变量的概率最高的值 $x^{\text{最大}}$ ，定义为

$$x^{\text{最大}} = \arg \max_x \left[\sum_{s \in \text{ne}(x)} \mu_{f_s \rightarrow x}(x) \right] \quad (8.98)$$

现在，我们可能试图简单地继续使用信息传递方法，使用公式（8.93）和公式（8.94），将信息从根结点传回叶结点，然后将公式（8.98）应用于所有剩余的变量结点。然而，由于我们现在进行的是最大化过程而不是求和过程，因此有可能存在多个 x 的配置，它们都会给出 $p(x)$ 的最大值。在这种情况下，这个策略就失效了，因为通过对属于不同的最大化配置的每个结点处的信息的乘积进行最大化得到的各个变量值可能给出一个并不对应于最大值的整体配置。

通过使用一个从根结点到叶结点的一个相当不同的信息传递方式，这个问题可以得到解决。为了说明工作原理，让我们再次回到简单的结点链的例子中，其中有 N 个变量 x_1, \dots, x_N ，每个变量有 K 个状态，对应于图8.38所示的图。假设我们令结点 x_N 是根结点。那么在第一阶段，我们从叶结点 x_1 开始，将信息传递到根结点，使用下面的公式

$$\begin{aligned} \mu_{x_n \rightarrow f_{n,n+1}}(x_n) &= \mu_{f_{n-1,n} \rightarrow x_n}(x_n) \\ \mu_{f_{n-1,n} \rightarrow x_n}(x_n) &= \max_{x_{n-1}} [\ln f_{n-1,n}(x_{n-1}, x_n) + \mu_{x_{n-1} \rightarrow f_{n-1,n}}(x_{n-1})] \end{aligned}$$

将公式（8.94）和公式（8.93）应用到这个特定的图上即可得到上面的结果。叶结点发送的初始信息为

$$\mu_{x_1 \rightarrow f_{1,2}}(x_1) = 0 \quad (8.99)$$

这样， x_N 的概率最高的值为

$$x_N^{\text{最大}} = \arg \max_{x_N} [\mu_{f_{N-1,N} \rightarrow x_N}(x_N)] \quad (8.100)$$

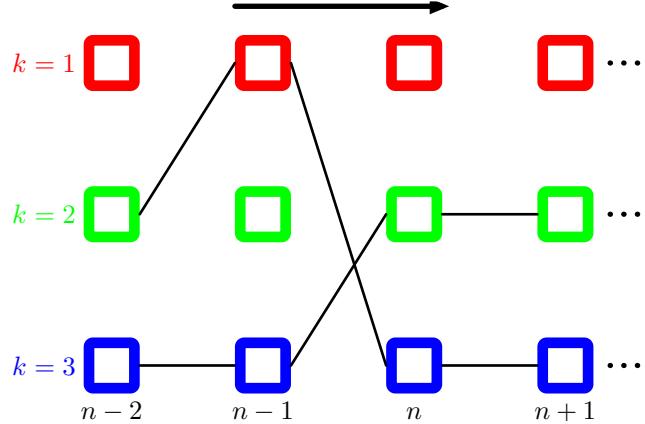


图 8.53: 一个晶格图, 或者格子图, 显式地画出了结点链模型中每个变量 x_n 的 K 个可能的状态 (图中的每一行表示一个状态)。在这个例子中, $K = 3$ 。箭头表示在最大乘积算法中信息传播的方向。对于每个变量 x_n (对应于图中第 n 列) 的每个状态 k , 函数 $\phi(x_n)$ 定义了之前变量的一个唯一的状态, 用黑线表示。穿过晶格的两条路径对应于能够得到联合概率分布最大值的配置, 每一条路径都可以沿着黑线, 按照箭头的相反方向反向跟踪的方式找到。

现在我们需要确定对应于同样的最大化配置的前一个变量的状态。可以这样做: 跟踪变量的哪个值产生了每个变量的最大值状态, 即存储下面的量

$$\phi(x_n) = \arg \max_{x_{n-1}} [\ln f_{n-1,n}(x_{n-1}, x_n) + \mu_{x_{n-1} \rightarrow f_{n-1,n}}(x_{n-1})] \quad (8.101)$$

为了更好地理解工作过程, 比较有帮助的做法是将变量链表示为晶格图 (lattice diagram) 或者格子图 (trellis diagram), 如图8.53所示。注意, 这不是一个概率图模型, 因为结点表示变量的独立状态, 而每个变量对应于图中这个状态的一列。对于给定变量的每个状态, 存在前一个变量的一个唯一的状态使得概率取最大值, 对应于公式 (8.101) 给出的函数 $\phi(x_n)$, 这通过连接结点的线表示。一旦我们知道了最终结点 x_N 的最可能的值, 我们就可以沿着链接回退, 找到结点 x_{N-1} 的最可能状态, 并且以此类推, 回到最初的结点 x_1 。这对应于将信息沿着链进行反方向的传递, 使用下面的公式

$$x_{n-1}^{\text{最大}} = \phi(x_n^{\text{最大}}) \quad (8.102)$$

被称为反向跟踪 (back-tracking)。注意, 可能存在多个 x_{n-1} 的值, 每个都能给出公式 (8.101) 的最大值。在进行反向跟踪时, 只要我们选择了这些变量中的一个, 那么我们就能够保证得到一个全局相容的最大化配置。

在图8.53中, 我们画出了两条路径。对于每条路径, 我们都假设对应于联合概率分布的一个全局最大值。如果 $k = 2$ 和 $k = 3$ 都表示 $x_N^{\text{最大}}$ 的可能值, 那么从任意一个状态开始沿着黑线回退 (对应于公式 (8.102) 的迭代), 我们都可以得到一个合法的全局最大值配置。注意, 如果我们运行一个正向的最大加和信息传递, 然后运行一个反向的传递, 之后对每个节点分别应用公式 (8.98), 那么我们最后会从一条路径中选出某些状态, 从另一条路径中选出另外一些状态, 得到一个并非为全局最大值的整体配置。我们看到, 有必要在正向信息传递时, 使用函数 $\phi(x_n)$ 对最大化状态进行跟踪, 然后使用反向跟踪找到一个相容的解。

现在, 推广到一般的树形结构因子图的方法就比较明显了。如果一条信息从因子结点 f 发送到变量结点 x , 那么最大化针对的是因子结点的全部其他变量结点 x_1, \dots, x_M , 使用公式 (8.93)。当我们进行这个最大化时, 我们记录了给出最大值的变量 x_1, \dots, x_M 的值。这样, 找到了 $x_N^{\text{最大}}$ 之后, 我们在反向跟踪步骤中可以使用这些存储的值, 为相容的最大状态 $x_1^{\text{最大}}, \dots, x_M^{\text{最大}}$ 的值。只要因子图是树, 最大加和算法以及反向跟踪方法就可以给出变量的精确最大化配置。这种方法的一个重要应用是寻找隐马尔科夫模型中隐含状态的最可能序列, 这种情况下被称为Viterbi算法。

与加和-乘积算法一样, 引入观测变量是很直接的。观测变量被限制为它们的观测值, 最大化过程针对剩余的隐含变量进行。形式化地, 可以通过引入恒等函数的方式, 将观测变量引入到因子函数中, 正如我们之前在加和-乘积算法中做的那样。

将最大加和算法与8.3.3节描述的迭代条件峰值算法 (ICM) 进行对比是很有趣的。ICM中的每一步计算都比较简单，因为从一个结点传递到下一个结点的“信息”由一个包含结点新状态的单一值组成。对于这个结点，条件概率分布被最大化。最大化加和算法更加复杂，因为信息是结点变量 x 的函数，从而由 x 的可能状态的 K 个值组成。然而，与最大化加和算法不同，即使对于树结构的图，ICM也无法保证找到一个全局的最大值。

8.4.6 一般图的精确推断

加和-乘积算法和最大化加和算法提供了树结构图中的推断问题的高效精确解法。然而，对于许多实际应用，我们必须处理带有环的图。

信息传递框架可以被推广到任意的图拓扑结构，从而得到一个精确的推断步骤，被称为联合树算法 (junction tree algorithm) (Lauritzen and Spiegelhalter, 1988; Jordan, 2007)。这里，我们简短地给出算法的关键步骤。这里不打算给出算法的细节，而是给出各个阶段的大致思想。如果我们的起始点是一个有向图，那么我们首先通过“伦理”步骤，将其转化为无向图。而如果起始点是无向图，那么这个步骤就不需要了。接下来，图被三角化 (triangulated)，这涉及到寻找包含四个或者更多结点的无弦环，然后增加额外的链接来消除无弦环。例如，在图8.36所示的图中，环 $A - C - B - D - A$ 是一个无弦环，从而一个连接应该添加到在 A 和 B 之间或者 C 和 D 之间。注意，三角化后的图的联合概率分布仍然由同样的势函数乘积定义，但是这些势函数现在被看做是扩展的变量集合上的势函数。接下来，三角化的图被用于构建新的树结构无向图，被称为联合树 (junction tree)，它的结点对应于三角化的图的最大团块，它的链接将具有相同变量的团块对连接在了一起。这种方法中连接哪对团块是很重要的问题。正确的做法是选择能得到最大生成树 (maximal spanning tree) 的连接方式，如下所述。对于连接了某个团块的所有可能的树，被选择的树是树的权值最大的一个，其中链接的权值是由它所连接的两个团块所共享的结点的数量，树的权值是链接的权值之和。由于三角化步骤的存在，得到的树满足运行相交性质 (running intersection property)，意思是如果一个变量被两个团块所包含，那么它必须也被连接这两个团块的路径上的任意团块所包含。这确保了变量推断在图之间是相容的。最后，一个二阶段的信息传递算法，或者等价的加和-乘积算法，现在可以被应用于这个联合树，得到边缘概率分布和条件概率分布。虽然联合树算法听起来比较复杂，但是它的核心是一个简单的想法。我们已经利用这个想法研究了概率的分解性质，使得加和与乘积能够相互交换，从而可以进行部分求和，避免了直接对联合概率分布的操作。联合树的作用是提供一种组织这些计算的精确高效的方法。值得注意的是，这些完全是通过图操作实现的！

联合树对于任意的图都是精确的、高效的。对于一个给定的图，通常不存在计算代价更低的算法。不幸的是，算法必须对每个结点的联合概率分布进行操作（每个结点对应于三角化的图的一个团块），因此算法的计算代价由最大团块中的变量数量确定。在离散变量的情形中，计算代价会随着这个数量指数增长。一个重要的概念是图的树宽度 (treewidth) (Bodlaender, 1993)，它根据最大团块中变量的数量进行定义。事实上，它被定义为最大团块的规模减一，来确保一个树的树宽度等于1。由于通常情况下，从一个给定的起始图开始，可以构建出多种不同的联合树，因此树宽度由最大团块具有最少变量的联合树来定义。如果原始图的树宽度比较大，那么联合树算法就变得不可行了。

8.4.7 循环置信传播

对于许多实际应用问题来说，使用精确推断是不可行的，因此我们需要研究有效的近似方法。这种近似方法中，一个重要的类别被称为变分 (variational) 方法，将在第10章详细讨论。作为这些确定性方法的补充，有一大类取样 (sampling) 方法，也被称为蒙特卡罗 (Monte Carlo) 方法。这些方法基于从概率分布中的随机数值取样，将在第11章中详细讨论。

这里，我们考虑带有环的图中的近似推断的一个简单方法，它直接依赖于之前对树的精确推断的讨论。主要思想就是简单地应用加和-乘积算法，即使不保证能够产生好的结果。这种方法被称为循环置信传播 (loopy belief propagation) (Frey and MacKay, 1998)。这种方法是可行的，因为加和-乘积算法的信息传递规则 (8.66) 和 (8.69) 完全是局部的。然而，由于现在图中存在环，因此信息会绕着图流动多次。对于某些模型，算法会收敛，而对于其他模型则不会。

为了应用这种方法，我们需要定义一个信息传递时间表（message passing schedule）。让我们假设在任意给定的链接以及任意给定的方向上，每次传递一条信息。从一个结点发送的每条信息替换了之前发送的任何沿着同一链接的同一方向的信息，并且本身是一个函数，这个函数只与算法的前一步的结点接收到的最近的信息有关。

我们已经看到，只有当结点从所有其他的链接接收到所有其他的信息之后，它才会沿着一条链接发送信息。由于图中存在环，因此这就产生了如何初始化信息传递算法的问题。为了解决这个问题，我们假设由单位函数给出的初始信息已经在所有方向上通过了每个链接。这样，每个结点都处在了发送信息的位置上。

现在有许多可能的方法来组织信息传递时间表。例如，洪水时间表（flooding schedule）在每一步同时向两个方向沿着每条链接同时传递信息，而每次只传递一个信息的时间表被称为连续时间表（serial schedule）。

根据Kschischang et al. (2001)，对于结点（变量结点或因子结点） a 和结点 b ，如果 a 自从上次向 b 发送信息后，从任何其他的链接接收到了任何信息，那么我们说结点 a 在到结点 b 的链接上有一个信息挂起（pending）。因此，当一个结点接收到了它的一个链接发送的信息，就在所有其他的链接上产生了挂起的信息。只有挂起的信息需要被传送，因为其他的信息仅仅复制了同样链接上的前一条信息。对于具有树结构的图来说，任何只发送挂起信息的时间表最后会终止于一条在任意方向上沿着任意链接发送过的信息。此时，没有挂起信息，并且每个变量接收到的信息给出了精确的边缘概率分布。然而，在具有环的图中，算法永远不会终结，因为总可能存在挂起信息，虽然在实际应用中发现，对于大部分应用，它都会在一个合理的时间内收敛。一旦算法收敛，或者如果未观测到收敛时算法停止，那么（近似）局部边缘概率分布可以使用每条链接上的每个变量结点或因子结点最近接收到的输入信息的乘积进行计算。

在一些应用中，循环置信传播算法会给出很差的结果，而在其他应用中，它被证明非常有效。特别地，对特定类型的误差修正编码的最好的解码算法等价于循环置信传播算法（Gallager, 1963; Berrou et al., 1993; McEliece et al., 1998; MacKay and Neal, 1999; Frey, 1998）。

8.4.8 学习图结构

在我们关于图模型的推断的讨论中，我们假设图的结构已知且固定。然而，也有一些研究超出了推断问题的范围，关注于从数据推断图结构本身（Friedman and Koller, 2003）。这需要我们定义一个可能结构的空间，以及用于对每个结构评分的度量。

从贝叶斯的观点来看，理想情况下，我们需要计算图结构上的后验概率分布，然后关于概率分布求平均，进行预测。如果我们有一个关于第 m 个图的先验概率分布 $p(m)$ ，那么后验概率分布为

$$p(m | \mathcal{D}) \propto p(m)p(\mathcal{D} | m) \quad (8.103)$$

其中 \mathcal{D} 是一个观测数据集。模型证据 $p(\mathcal{D} | m)$ 提供了每个模型的分数。然而，计算模型证据涉及到对潜在变量的积分或求和，这对于许多模型来说是一个计算量相当大的问题。

探索图结构的空间也是一个问题。由于不同图结构的数量随着结点数量的增加而指数增长，因此通常需要借助启发式方法找到好的候选。

8.5 练习

(8.1) (*) 通过按顺序将变量积分出去的方式，证明有向图的联合概率分布的表达式
(8.5) 被正确地归一化，假设每个条件概率分布都被归一化。

(8.2) (*) 证明，有向图中没有有向环的性质可以由下面的叙述得出：存在一个排序后的结点序号序列，使得对于每个结点，不存在通向序号较低的结点的链接。

(8.3) (**) 考虑三个二值变量 $a, b, c \in \{0, 1\}$ ，联合概率分布如表8.2所示。通过直接计算，证明这个概率分布中， a 和 b 的边缘概率分布不是独立的，即 $p(a, b) \neq p(a)p(b)$ ，但是当以 c 为条件时，它们变为独立，即对于 $c = 0$ 和 $c = 1$ ，都有 $p(a, b | c) = p(a | c)p(b | c)$ 。

(8.4) (**) 计算与表8.2给出的联合概率分布相对应的 $p(a), p(b | c)$ 和 $p(c | a)$ 。从而，通过直接计算证明 $p(a, b, c) = p(a)p(c | a)p(b | c)$ 。画出对应的有向图。

(8.5) (*) 画一个与公式 (7.79) 和 (7.80) 描述的相关向量机相对应的有向概率图模型。

a	b	c	$p(a, b, c)$
0	0	0	0.192
0	0	1	0.144
0	1	0	0.048
0	1	1	0.216
1	0	0	0.192
1	0	1	0.064
1	1	0	0.048
1	1	1	0.096

表 8.2: 不同阶数的多项式的系数 w^* 的值。观察随着多项式阶数的增加，系数的大小是如何剧烈增大的。

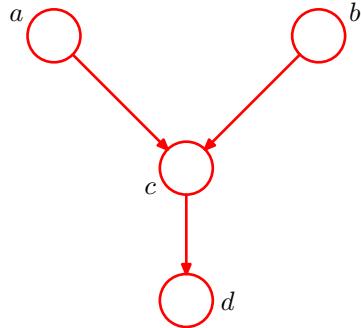


图 8.54: 用来研究头到头路径 $a - c - b$ 的条件独立性质的图模型，其中 c 的一个后代即结点 d 被观测到。

(8.6) (*) 对于图8.13所示的模型，我们看到，使用logistic sigmoid的表示方法，确定条件概率分布 $p(y | x_1, \dots, x_M)$ 的参数的数量可以从 2^M 减小到 $M + 1$ ，其中 $x_i \in \{0, 1\}$ 。另一种表示方法 (Pearl, 1988) 为

$$p(y = 1 | x_1, \dots, x_M) = 1 - (1 - \mu_0) \prod_{i=1}^M (1 - \mu_i)^{x_i} \quad (8.104)$$

其中 $0 \leq \mu_i \leq 1 (i = 0, \dots, M)$ 。条件概率分布 (8.104) 被称为“噪声或”(noisy-OR)。证明，它可以被看成逻辑或函数（即至少有一个 $x_i = 1$ 时会得到 $y = 1$ 的函数）的一个“软”（概率）形式。讨论 μ_i 的意义。

(8.7) (**) 使用递归关系 (8.15) 和 (8.16)，证明图8.14给出的图模型的联合概率分布的均值和协方差分别为 (8.17) 和 (8.18)。

(8.8) (*) 证明 $a \perp\!\!\!\perp b, c | d$ 可以推导出 $a \perp\!\!\!\perp b | d$ 。

(8.9) (*) 使用d-划分准则，证明对于有向图中的一个结点 x ，以马尔科夫毯中的所有结点为条件，它的条件概率分布与图中剩余的变量独立。

(8.10) (*) 考虑图8.54中的有向图，其中没有观测变量。证明 $a \perp\!\!\!\perp b | \emptyset$ 。假设我们现在观测到了变量 d 。证明一般情况下 $a \perp\!\!\!\perp b | d$ 。

(8.11) (**) 考虑图8.21给出的汽车燃料系统的例子。假设我们不直接观测到油量计 G 的状态，而是由司机 D 观测，然后向我们报告读数。报告要么是油量计的读数为“满” $D = 1$ ，要么是油量计的读数为“空” $D = 0$ 。我们的司机有些不可靠，正如下面的概率所表示的那样

$$p(D = 1 | G = 1) = 0.9 \quad (8.105)$$

$$p(D = 0 | G = 0) = 0.9 \quad (8.106)$$

假设司机告诉我们油量计的读数是空的，换句话说我们观测到 $D = 0$ 。只给定这个观测，计算油箱为空的概率。类似地，假设我们还观测到电池没有电，计算对应的给定。注意，第二个概率更低。讨论这个结果背后的直观思想，将这个结果与图8.54相关联。

(8.12) (*) 证明在 M 个不同的随机向量组成的集合上，存在 $2^{M(M-1)/2}$ 个不同的随机变量。画出 $M = 3$ 情形下的8个概率。

(8.13) (*) 考虑使用迭代条件峰值 (ICM) 来最小化公式 (8.42) 给出的能量函数。写出与一个特定变量 x_j 相关联的两个状态的能量值之差的表达式，保持所有其他的变量固定。证明，这个值仅仅依赖于图中局部在 x_j 的量。

(8.14) (*) 考虑公式 (8.42) 给出的能量函数的一个特定的形式，其中系数 $\beta = h = 0$ 。证明潜在变量的概率最高的配置为 $x_i = y_i$ (对于所有的*i*)。

(8.15) (**) 证明图 8.38 所示的图中两个相邻结点的联合概率分布 $p(x_{n-1}, x_n)$ 由形如 (8.58) 的表达式给出。

(8.16) (**) 对于图 8.38 中的图，对于所有结点 $n \in \{1, \dots, N-1\}$ ，考虑计算 $p(x_n | x_N)$ 的推断问题。证明 8.4.1 节讨论的信息传递算法可以高效地解决这个问题。讨论哪些信息被修改，如何修改。

(8.17) (**) 考虑图 8.38 给出的图，结点数为 $N = 5$ ，结点 x_3 和 x_5 被观测。使用 d-划分证明 $x_2 \perp\!\!\!\perp x_5 | x_3$ 。证明，如果 8.4.1 节的信息传递算法被应用于 $p(x_2 | x_3, x_5)$ 的计算，那么结果独立于 x_5 的值。

(8.18) (**) 证明有向树表示的概率分布可以简单地写成对应的无向树上的一个等价的概率分布。并且证明，通过对团块势函数进行适当的归一化，表示为无向树的概率分布可以写成有向树。计算可以从给定的无向树构建的不同的有向树的数量。

(8.19) (**) 将 8.4.4 节讨论的加和-乘积算法应用到 8.4.1 节讨论的结点链模型，证明 (8.54)、(8.55) 和 (8.57) 给出的结果可以作为一种具体的情形被求出。

(8.20) (*) 考虑树结构的因子图上的加和-乘积算法的信息传递协议，其中信息首先从叶结点传递到一个任意选择的根结点，然后从根结点向外传递到叶结点。使用归纳法证明信息可以用下面的方式传递：在每一个步骤中，每个必须发送信息的结点已经接收到用来构建输出信息的所有必要的输入信息。

(8.21) (**) 证明，在一个因子图中，与每个因子 $f_s(\mathbf{x}_s)$ 相关联的变量 \mathbf{x}_s 的集合上的边缘概率分布 $p(\mathbf{x}_s)$ 可以用下面的方式求出：首先运行加和-乘积信息传递算法，然后使用公式 (8.72) 计算所需的边缘概率分布。

(8.22) (*) 考虑一个树结构的因子图，其中变量结点的一个给定的子集组成了一个连接子图（即子集的任意变量结点都通过一个单一的因子结点与至少一个其他的变量结点相连接）。说明如何使用加和乘积算法来计算在这个子集上的边缘概率分布。

(8.23) (**) 在 8.4.4 节，我们证明了因子图中的一个变量结点 x_i 的边缘概率分布 $p(x_i)$ 等于从相邻因子结点到这个结点的信息的乘积，形式为 (8.63)。证明边缘概率分布 $p(x_i)$ 也可以写成输入信息的乘积，输入信息所在的链接与输出信息所在的链接相同。

(8.24) (**) 证明，在运行了加和-乘积信息传递算法之后，一个树结构因子图中的因子 $f_s(\mathbf{x}_s)$ 的变量 \mathbf{x}_s 的边缘概率分布可以写成沿着所有链接到达这个因子结点的信息的乘积，乘以形式为 (8.72) 的局部因子 $f(\mathbf{x}_s)$ 。

(8.25) (**) 在公式 (8.86) 中，我们验证了在图 8.51 所示的图中运行加和-乘积算法，并且令结点 x_3 被设置为根结点，可以给出 x_2 的正确的边缘概率。证明我们也可以得到 x_1 和 x_3 的正确的边缘概率。类似地，证明，在这个图上运行加和-乘积算法之后，使用结果 (8.72) 可以得到 x_1, x_2 的正确的联合概率分布。

(8.26) (*) 考虑离散变量上的一个树结构的因子图，假设我们希望计算与两个变量 x_a 和 x_b 关联的联合概率分布 $p(x_a, x_b)$ ，这两个变量不属于同一个因子。定义一个使用加和-乘积算法计算这个联合概率分布的步骤，其中一个变量被连续地限制等于它的每个合法的值。

(8.27) (**) 考虑两个离散变量 x 和 y ，每个变量有三个可能的状态，例如 $x, y \in \{0, 1, 2\}$ 。构造这些变量上的一个联合概率分布 $p(x, y)$ ，它具有下面的性质：最大化边缘概率 $p(x)$ 的值 \hat{x} 以及最大化边缘概率 $p(y)$ 的值 \hat{y} 在联合概率分布下的整体概率为零，即 $p(\hat{x}, \hat{y}) = 0$ 。

(8.28) (**) 因子图的加和-乘积算法的挂起 (pending) 信息的概念在 8.4.7 节定义。证明，如果图中存在一个或者多个环，那么总会存在至少一个挂起信息，它与算法运行的时间无关。

(8.29) (**) 证明，如果加和-乘积算法运行在一个树结构（没有环）的因子图上，那么信息被发送有限次之后，不会有挂起信息。

9 混合模型和EM

如果我们定义观测变量和潜在变量的一个联合概率分布，那么对应的观测变量本身的概率分布可以通过求边缘概率的方法得到。这使得观测变量上的复杂的边缘概率分布可以通过观测变量与潜在变量组成的扩展空间上的更加便于计算的联合概率分布来表示。因此，潜在变量的引入使得复杂的概率分布可以由简单的分量组成。本章中，我们会看到混合概率分布（例如2.3.9节讨论的高斯混合模型）可以用离散潜在变量来表示。连续潜在变量是第12章的主题。

除了提供了一个构建更复杂的概率分布的框架之外，混合模型也可以用于数据聚类。因此，在开始讨论混合概率分布时，我们会考虑寻找数据点集合中的聚类的问题。我们首先使用一个非概率的方法解决这个问题，这个方法被称为 K 均值算法（Lloyd, 1982）。之后，我们引入混合概率分布的潜在变量观点，其中离散潜在变量可以被看做将数据点分配到了混合概率分布的具体成分当中。潜在变量模型中寻找最大似然估计的一个一般的方法是期望最大化（EM）算法。我们首先使用高斯混合分布，以一种相当非形式化的方式介绍EM算法，然后我们会基于潜在变量的观点，给出一个更加仔细的处理方法。我们会看到， K 均值算法对应于用于高斯混合模型的EM算法的一个特定的非概率极限。最后，我们会以一种一般的方式讨论EM算法。

高斯混合模型广泛应用于数据挖掘、机器学习和统计分析中。在许多应用中，参数由最大似然方法确定，通常会使用EM算法。然而，正如我们将看到的那样，最大似然方法有一些巨大的局限性。在第10章中，我们会看到，使用变分推断的方法，可以得到一个优雅的贝叶斯处理方式。与EM相比，这种方法几乎不需要额外的计算量，并且它解决了最大似然方法中的主要困难，也使得混合模型的分量的数量可以自动从数据中推断。

9.1 K 均值聚类

首先，我们考虑寻找多维空间中数据点的分组或聚类的问题。假设我们有一个数据集 $\{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ ，它由 D 维欧几里得空间中的随机变量 \mathbf{x} 的 N 次观测组成。我们的目标是将数据集划分为 K 个类别。现阶段我们假定 K 的值是给定的。直观上讲，我们会认为由一组数据点构成的一个聚类中，聚类内部点之间的距离应该小于数据点与聚类外部的点之间的距离。我们可以形式化地说明这个概念。引入一组 D 维向量 $\boldsymbol{\mu}_k$ ，其中 $k = 1, \dots, K$ ，且 $\boldsymbol{\mu}_k$ 是与第 k 个聚类关联的一个代表。正如我们将看到的那样，我们可以认为 $\boldsymbol{\mu}_k$ 表示了聚类的中心。我们的目标是找到数据点分别属于的聚类，以及一组向量 $\{\boldsymbol{\mu}_k\}$ ，使得每个数据点和与它最近的向量 $\boldsymbol{\mu}_k$ 之间的距离的平方和最小。

现在，比较方便的做法是定义一些记号来描述数据点的聚类情况。对于每个数据点 \mathbf{x}_n ，我们引入一组对应的二值指示变量 $r_{nk} \in \{0, 1\}$ ，其中 $k = 1, \dots, K$ 表示数据点 \mathbf{x}_n 属于 K 个聚类中的哪一个，从而如果数据点 \mathbf{x}_n 被分配到类别 k ，那么 $r_{nk} = 1$ ，且对于 $j \neq k$ ，有 $r_{nj} = 0$ 。这被称为“1-of- K ”表示方式。之后我们可以定义一个目标函数，有时被称为失真度量（distortion measure），形式为

$$J = \sum_{n=1}^N \sum_{k=1}^K r_{nk} \|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2 \quad (9.1)$$

它表示每个数据点与它被分配的向量 $\boldsymbol{\mu}_k$ 之间的距离的平方和。我们的目标是找到 $\{r_{nk}\}$ 和 $\{\boldsymbol{\mu}_k\}$ 的值，使得 J 达到最小值。我们可以用一种迭代的方法完成这件事，其中每次迭代涉及到两个连续的步骤，分别对应 r_{nk} 的最优化和 $\boldsymbol{\mu}_k$ 的最优化。首先，我们为 $\boldsymbol{\mu}_k$ 选择一些初始值。然后，在第一阶段，我们关于 r_{nk} 最小化 J ，保持 $\boldsymbol{\mu}_k$ 固定。在第二阶段，我们关于 $\boldsymbol{\mu}_k$ 最小化 J ，保持 r_{nk} 固定。不断重复这个二阶段优化直到收敛。我们会看到，更新 r_{nk} 和更新 $\boldsymbol{\mu}_k$ 的两个阶段分别对应于EM算法中的E（期望）步骤和M（最大化）步骤。为了强调这一点，我们会在 K 均值算法中使用E步骤和M步骤的说法。

首先考虑确定 r_{nk} 。由于公式(9.1)给出的 J 是 r_{nk} 的一个线性函数，因此最优化过程可以很容易地进行，得到一个解析解。与不同的 n 相关的项是独立的，因此我们可以对每个 n 分别进行最优化，只要 k 的值使 $\|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2$ 最小，我们就令 r_{nk} 等于1。换句话说，我们可以简单地将数据点的聚类设置为最近的聚类中心。更形式化地，这可以表达为

$$r_{nk} = \begin{cases} 1 & \text{如果 } k = \arg \min_j \|\mathbf{x}_n - \boldsymbol{\mu}_j\|^2 \\ 0 & \text{其他情况} \end{cases} \quad (9.2)$$

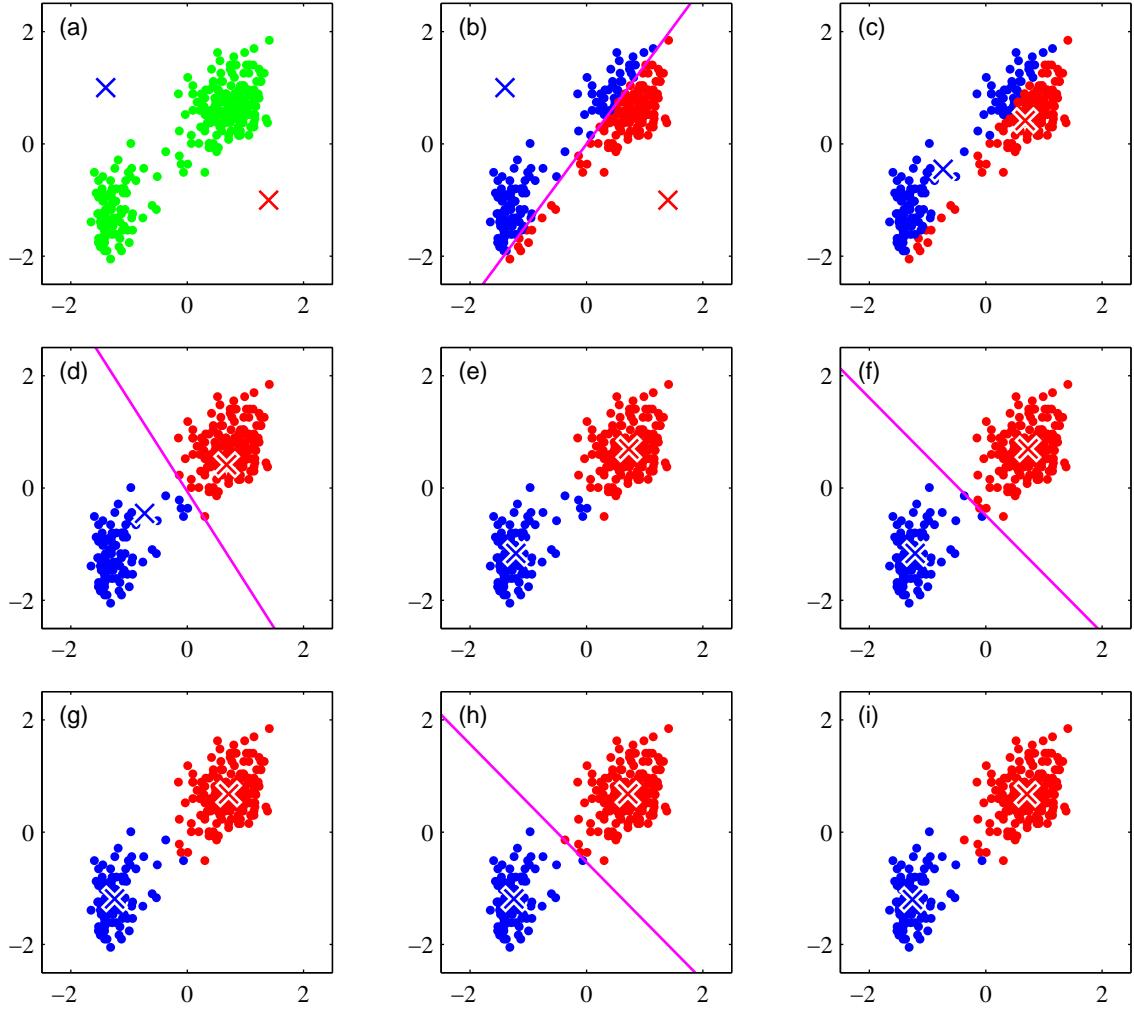


图 9.1: 使用重新缩放的老忠实间歇喷泉数据集对 K 均值算法进行说明。(a) 绿点表示二维欧几里得空间中的数据集, 中心 μ_1 和 μ_2 的初始选择分别用红色叉号和蓝色叉号表示。(b) 在初始的 E 步骤中, 每个数据点被分配为红色聚类或者蓝色聚类, 根据与哪个中心更近来确定类别。这等价于根据两个聚类中心的垂直平分线来对数据点进行分类, 中垂线用洋红色直线表示。(c) 在接下来的 M 步骤中, 每个聚类中心使用分配到对应类别的数据点重新计算。(d)-(i) 给出了接下来的 E 步骤和 M 步骤, 直到最终收敛。

现在考虑 r_{nk} 固定时, 关于 μ_k 的最优化。目标函数 J 是 μ_k 的一个二次函数, 令它关于 μ_k 的导数等于零, 即可达到最小值, 即

$$2 \sum_{n=1}^N r_{nk} (\mathbf{x}_n - \boldsymbol{\mu}_k) = 0 \quad (9.3)$$

可以很容易地解出 $\boldsymbol{\mu}_k$, 结果为

$$\boldsymbol{\mu}_k = \frac{\sum_n r_{nk} \mathbf{x}_n}{\sum_n r_{nk}} \quad (9.4)$$

这个表达式的分母等于聚类 k 中数据点的数量, 因此这个结果有一个简单的含义, 即令 $\boldsymbol{\mu}_k$ 等于类别 k 的所有数据点的均值。因此, 上述步骤被称为 K -均值 (K -means) 算法。

重新为数据点分配聚类的步骤以及重新计算聚类均值的步骤重复进行, 直到聚类的分配不改变 (或者直到迭代次数超过了某个最大值)。由于每个阶段都减小了目标函数 J 的值, 因此算法的收敛性得到了保证。然而, 算法可能收敛到 J 的一个局部最小值而不是全局最小值。 K -均值算法的收敛性质的讨论, 可以参考 MacQueen (1967)。

图 9.1 给出了将 K -均值算法应用于老忠实间歇喷泉数据集上的结果。对于这个例子, 我们对数据进行了一个线性的重新标度, 被称为标准化 (standardizing), 使得每个变量的均值为零,

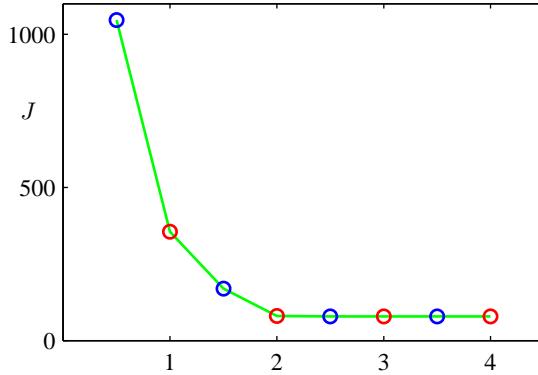


图 9.2: 对于图9.1给出的 K 均值算法，在每个E步骤（蓝点）和M步骤（红点）之后，代价函数 J 的图像。算法在第三个M步骤之后收敛，最后一个EM循环对于分类情况和代表向量都没造成改变。

标准差为单位标准差。这个例子中，我们选择了 $K = 2$ ，因此这种情况下，将每个数据点分配到最近的聚类中心等价于将数据点按照其位于两个数据中心的垂直平分线的哪一侧对数据点进行分类。对于老忠实间歇喷泉数据，公式 (9.1) 给出的代价函数如图9.2所示。

注意，我们故意将聚类中心选择了较差的初始值，从而算法在收敛之前执行了若干步。在实际应用中，一个更好的初始化步骤是将聚类中心选择为由 K 个随机数据点组成的子集。还有一点值得注意的地方， K 均值算法本身经常被用于在EM算法之前初始化高斯混合模型的参数。

直接实现这里讨论的 K 均值算法会相当慢，因为在每个E步骤中，必须计算每个代表向量与每个数据点之间的欧几里得距离。关于加速 K 均值算法，有很多方法被提出来，一些方法基于对数据结构的预先计算，例如将数据组织成树结构，使得相邻的数据点属于同一个子树 (Ramasubramanian and Paliwal, 1990; Moore, 2000)。另外一些方法使用距离的三角不等式，因此避免了不必要的距离计算 (Hodgson, 1998; Elkan, 2003)。

目前为止，我们已经研究了 K 均值算法的一个批处理版本，其中每次更新代表向量时都使用了整个数据集。我们也可以推导一个在线随机算法 (MacQueen, 1967)，方法是：将Robbins-Monro步骤应用到寻找回归函数的根的问题中，其中回归函数由公式 (9.1) 给出的 J 关于 μ_k 的导数给出。这产生了顺序更新算法，其中对于每个数据点 x_n ，我们使用下式更新最近的代表向量 μ_k 。

$$\mu_k^{\text{新}} = \mu_k^{\text{旧}} + \eta_n(x_n - \mu_k^{\text{旧}}) \quad (9.5)$$

其中 η_n 是学习率参数，通常令其关于数据点的数量单调递减。

K 均值算法的基础是将平方欧几里得距离作为数据点与代表向量之间不相似程度的度量。这不仅限制了能够处理的数据变量的类型（例如，它不能处理某些或全部变量表示类别标签的情形），而且使得聚类中心的确定对于异常点不具有鲁棒性。我们可以这样推广 K 均值算法：引入两个向量 x 和 x' 之间的一个更加一般的不相似程度的度量 $\mathcal{V}(x, x')$ ，然后最小化下面的失真度量

$$\tilde{J} = \sum_{n=1}^N \sum_{k=1}^K r_{nk} \mathcal{V}(x_n, \mu_k) \quad (9.6)$$

这就给出了 K 中心点算法 (K -medoids algorithm)。与之前一样，对于给定的聚类代表 μ_k ，E步骤涉及到为每个数据点分配聚类，使得与对应的聚类代表的不相似程度最小。这一步的计算代价为 $O(KN)$ ，与标准的 K 均值算法的情形相同。对于不相似程度度量的一般选择，M步骤通常比 K 均值的情形更加复杂，因此通常会将聚类原型限制为等于某个分配到那个聚类的数据向量，因为这使得算法可以适用于任何不相似程度的度量 $\mathcal{V}(\cdot, \cdot)$ ，只要它能够被计算。因此，对于每个聚类 k ，M步骤涉及到在分配到那个聚类的 N_k 个点上的离散搜索，这需要 $O(N_k^2)$ 次对 $\mathcal{V}(\cdot, \cdot)$ 的计算。

K 均值算法的一个值得注意的特征是，在每一次迭代中，每个数据点被分配到一个唯一的聚类中。虽然某些数据点与某个特定的中心 μ_k 的距离远远小于与其他中心的距离，但是也存在在其



图 9.3: 使用 K 均值聚类算法进行图像分割的两个例子。图中给出了原始图像以及使用不同的 K 值得到的 K 均值分割结果。这张图也说明了向量量子化用于数据压缩的效果，其中较小的 K 值会得到较高的压缩率，代价是图像的质量更差。

他的数据点，位于两个聚类中心的大概中间的位置。在后一种情形中，强行将数据点分配到最近的聚类不是最合适的选择。我们在下一节会看到，通过使用概率的方法，我们得到了对数据点聚类的“软”分配，它反映了在最合适聚类分配上的不确定性。这个概率形式带来了一些数值计算上的优势。

9.1.1 图像分割与压缩

作为 K 均值算法的一个应用，我们考虑两个相关的问题，即图像分割和图像压缩。图像分割的目标是将图像分割成若干的区域，每个区域有一个相对相似的视觉外观，或者对应于某个物体或物体的一部分 (Forsyth and Ponce, 2003)。图像中的每个像素是一个3维空间中的一个点，这个三维空间由红、绿、蓝通道的亮度值构成。我们的分割算法简单地将图像中的每个像素看做一个独立的数据点。注意，严格地说，这个空间不是欧几里得空间，因为通道亮度被限制在区间 $[0, 1]$ 。尽管这样，我们可以没有难度地应用 K 均值算法。我们给出了运行 K 均值算法直至收敛的结果。对于任意特定的 K 值，我们将每个像素的 $\{R, G, B\}$ 亮度三元组用聚类中心 μ_k 的亮度值替代。对于不同的 K 值，结果如图 9.3 所示。我们看到，对于给定的 K 值，算法使用一个只有 K 个颜色的调色板来表示图像。要强调的一点是， K 均值的使用对于图像分割来说不是一个特别复杂的方法，因为它没有考虑不同像素的空间上的近似性。图像分割问题通常特别困难，仍然是一个活跃的研究领域。在这里进行介绍只是为了说明 K 均值算法的行为。

我们也可以使用聚类算法的结果进行数据压缩。区分无损数据压缩 (lossless data compression) 与有损数据压缩 (lossy data compression) 是很有必要的。无损数据压缩中，我们的目标是能够从压缩的表示中精确地重建原始数据，而有损数据压缩中，我们接受重建过程中出现的一些错误。我们可以将 K 均值算法按照下面的方式应用到有损数据压缩中。对于 N 个数据点中的每一个，我们只存储它被分配的聚类种类 k 。我们还存储了 K 个聚类中心 μ_k 的值，这通常需要存储小得多的数据，其中我们假定 $K \ll N$ 。这样，每个数据点都根据它最近的



图 9.4: 混合模型的图形表示, 其中联合概率分布被表示为 $p(\mathbf{x}, \mathbf{z}) = p(\mathbf{z})p(\mathbf{x} | \mathbf{z})$ 的形式。

中心 μ_k 确定。新的数据点可以类似地压缩。首先找到最近的 μ_k , 然后存储标签 k 而不是原始的数据向量。这个框架被称为向量量子化 (vector quantization), 向量 μ_k 被称为编码书向量 (code-book vector)。

上面讨论的图像分割问题也说明了数据压缩中聚类的使用。假设原始图像有 N 个像素, 每个像素由 $\{R, G, B\}$ 三个值组成, 每个值由 8 比特的精度存储。这样, 直接传递整幅图像需要 $24N$ 比特。现在假设我们首先在图像数据上运行 K 均值算法, 然后, 我们不直接传递原始像素亮度向量, 而是传递最近的向量 μ_k 的亮度。由于有 K 个这样的向量, 因此每个像素需要 $\log_2 K$ 比特。我们还必须传送 K 个编码书向量 μ_k , 这需要 $24K$ 比特, 因此传递这个图像所需的比特总数为 $24K + N \log_2 K$ (四舍五入到最近的整数)。图 9.3 给出的原始图像有 $240 \times 180 = 43,200$ 个像素, 因此直接传递需要 $24 \times 43,200 = 1,036,800$ 个比特。作为对比, 传递压缩的图像分别需要 43,248 比特 ($K = 2$), 86,472 比特 ($K = 3$) 以及 173,040 比特 ($K = 10$)。这表示与原始图像相比, 压缩率分别为 4.2%, 8.3% 和 16.7%。我们看到存在一个压缩程度与图像质量之间的折中。注意, 在这个例子中, 我们的目的是说明 K 均值算法。如果我们的目标是生成一个好的图像压缩算法, 那么更好的方法是考虑相邻像素组成的小块, 例如 5×5 , 从而利用了自然图像中相邻像素之间存在的相关性。

9.2 混合高斯

在 2.3.9 节, 我们将高斯混合模型看成高斯分量的简单线性叠加, 目标是提供一类比单独的高斯分布更强大的概率模型。我们现在使用离散潜在变量来描述高斯混合模型。这会让我们更深刻地认识这个重要的分布, 也会让我们开始了解期望最大化算法。

回忆一下, 根据公式 (2.188), 高斯混合概率分布可以写成高斯分布的线性叠加的形式, 即

$$p(\mathbf{x}) = \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \quad (9.7)$$

让我们引入一个 K 为二值随机变量 \mathbf{z} , 这个变量采用了“1-of- K ”表示方法, 其中一个特定的元素 z_k 等于 1, 其余所有的元素等于 0。于是 z_k 的值满足 $z_k \in \{0, 1\}$ 且 $\sum_k z_k = 1$, 并且我们看到根据哪个元素非零, 向量 \mathbf{z} 有 K 个可能的状态。我们根据边缘概率分布 $p(\mathbf{z})$ 和条件概率分布 $p(\mathbf{x} | \mathbf{z})$ 定义联合概率分布 $p(\mathbf{x}, \mathbf{z})$, 对应于图 9.4 所示的图模型。 \mathbf{z} 的边缘概率分布根据混合系数 π_k 进行赋值, 即

$$p(z_k = 1) = \pi_k$$

其中参数 $\{\pi_k\}$ 必须满足

$$0 \leq \pi_k \leq 1 \quad (9.8)$$

以及

$$\sum_{k=1}^K \pi_k = 1 \quad (9.9)$$

使得概率是一个合法的值。由于 \mathbf{z} 使用了“1-of- K ”表示方法, 因此我们也可以将这个概率分布写成

$$p(\mathbf{z}) = \prod_{k=1}^K \pi_k^{z_k} \quad (9.10)$$

类似地，给定 z 的一个特定的值， \mathbf{x} 的条件概率分布是一个高斯分布

$$p(\mathbf{x} \mid z_k = 1) = \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

也可以写成

$$p(\mathbf{x} \mid \mathbf{z}) = \prod_{k=1}^K \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)^{z_k} \quad (9.11)$$

联合概率分布为 $p(z)p(\mathbf{x} \mid z)$ ，从而 \mathbf{x} 的边缘概率分布可以通过将联合概率分布对所有可能的 z 求和的方式得到，即

$$p(\mathbf{x}) = \sum_{\mathbf{z}} p(\mathbf{z})p(\mathbf{x} \mid \mathbf{z}) = \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \quad (9.12)$$

其中我们使用了公式 (9.10) 和公式 (9.11)。因此 \mathbf{x} 的边缘概率分布是公式 (9.7) 的高斯混合分布。如果我们有若干个观测 $\mathbf{x}_1, \dots, \mathbf{x}_N$ ，那么，由于我们已经用 $p(\mathbf{x}) = \sum_{\mathbf{z}} p(\mathbf{x}, \mathbf{z})$ 的方式表示了边缘概率分布，因此对于每个观测数据点 \mathbf{x}_n ，存在一个对应的潜在变量 z_n 。

于是，我们找到了高斯混合分布的一个等价的公式，将潜在变量显式地写出。似乎我们这么做没有什么意义。但是，我们现在能够对联合概率分布 $p(\mathbf{x}, z)$ 操作，而不是对边缘概率分布 $p(\mathbf{x})$ 操作，这会产生极大的计算上的简化。通过引入期望最大化 (EM) 算法，即可看到这一点。

另一个起着重要作用的量是给定 \mathbf{x} 的条件下， z 的条件概率。我们会用 $\gamma(z_k)$ 表示 $p(z_k = 1 \mid \mathbf{x})$ ，它的值可以使用贝叶斯定理求出

$$\begin{aligned} \gamma(z_k) &\equiv p(z_k = 1 \mid \mathbf{x}) = \frac{p(z_k = 1)p(\mathbf{x} \mid z_k = 1)}{\sum_{j=1}^K p(z_j = 1)p(\mathbf{x} \mid z_j = 1)} \\ &= \frac{\pi_k \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^K \pi_j \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_k)} \end{aligned} \quad (9.13)$$

我们将 π_k 看成 $z_k = 1$ 的先验概率，将 $\gamma(z_k)$ 看成观测到 \mathbf{x} 之后，对应的后验概率。正如我们将看到的那样， $\gamma(z_k)$ 也可以被看做分量 k 对于“解释”观测值 \mathbf{x} 的“责任” (responsibility)。

我们可以使用祖先取样的方法生成服从高斯混合模型的概率分布的随机样本。为了完成这件事，我们首先生成 z 的一个值，记作 \hat{z} ，它服从概率分布 $p(z)$ 。然后，根据条件概率分布 $p(\mathbf{x} \mid \hat{z})$ 生成 \mathbf{x} 的一个值。从标准的概率分布中取样的方法将在第11章讨论。我们可以用下面的方法描绘联合概率分布 $p(\mathbf{x}, z)$ ：首先画出 \mathbf{x} 的对应值的点，然后根据 z 的值对它进行着色，换句话说，根据哪个高斯分布负责生成这个数据进行着色，如图9.5(a)所示。类似地，服从边缘概率分布 $p(\mathbf{x})$ 的样本可以通过从联合概率分布中取样然后忽略 z 的值的方式得到。这些如图9.5(b)所示。图中画出了 \mathbf{x} 的值，没有任何颜色标记。

我们也可以使用这个人工生成的数据来说明“责任”的含义。对于每个数据点，我们计算生成了数据集的混合概率分布的每个分量的后验概率分布。特别地，我们可以表示出与数据点 \mathbf{x}_n 相关联的责任 $\gamma(z_{nk})$ 的值，方法是：对于 $k = 1, 2, 3$ ，我们分别用红色、蓝色、绿色来画出对应的点，点的颜色的红蓝绿分量的比例由 $\gamma(z_{nk})$ 给出，如图9.5(c)所示。因此， $\gamma(z_{n1}) = 1$ 的数据点会被标记为红色，而 $\gamma(z_{n2}) = \gamma(z_{n3}) = 0.5$ 的数据点的颜色中，蓝色和绿色的比例相同，因此是青色。应该将这幅图与图9.5(a)进行对比，那里数据点使用它们被生成的真实的分量类别进行了标记。

9.2.1 最大似然

假设我们有一个观测的数据集 $\{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ ，我们希望使用混合高斯模型来对数据进行建模。我们可以将这个数据集表示为一个 $N \times D$ 的矩阵 \mathbf{X} ，其中第 n 行为 \mathbf{x}_n^T 。类似地，对应的隐含变量会被表示为一个 $N \times K$ 的矩阵 \mathbf{Z} ，它的行为 z_n^T 。如果我们假定数据点独立地从概率分布中抽取，那么我们可以使用图9.6所示的图模型来表示这个独立同分布数据集的高斯混合模型。

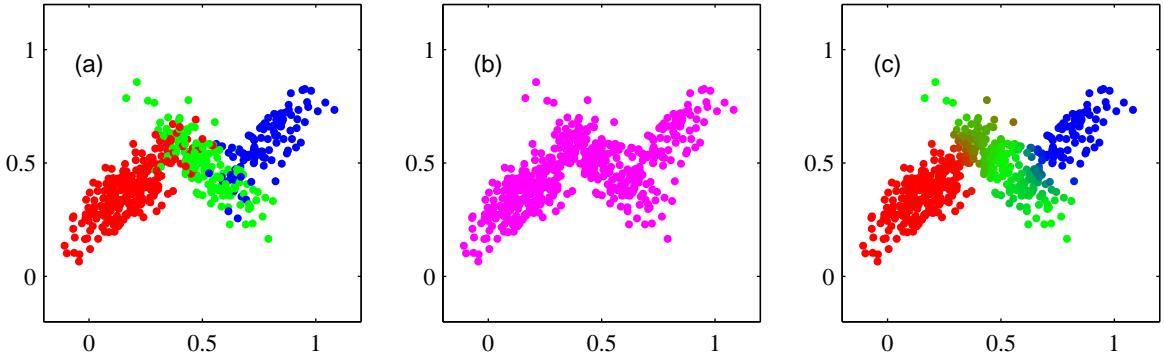


图 9.5: 从图 2.23 给出的 3 个高斯分布组成的混合分布中抽取的 500 个样本点。(a) 从联合概率分布 $p(z)p(x|z)$ 中抽取的样本, 其中 z 的三种状态对应于混合分布的三个分量, 用红色、绿色、蓝色表示。(b) 来自边缘概率分布 $p(x)$ 的对应的样本, 仅仅将 z 的值忽略, 画出 x 的值即可。(a) 中的数据集被称为完整的, (b) 中的数据集被称为不完整的。(c) 同样的样本, 颜色表示与数据点 x_n 关联的责任 $\gamma(z_{nk})$, 其中红色、蓝色、绿色所占的比重分别由 $\gamma(z_{nk})$, $k = 1, 2, 3$ 给出。

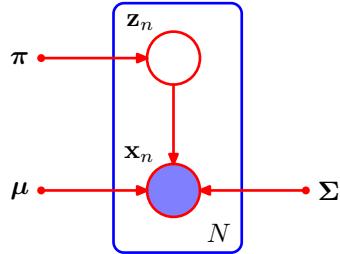


图 9.6: 一组 N 个独立同分布数据点 $\{x_n\}$ 的高斯混合模型的图表示, 对应的潜在变量为 $\{z_n\}$, 其中 $n = 1, \dots, N$ 。

根据公式 (9.7), 对数似然函数为

$$\ln p(\mathbf{X} | \pi, \mu, \Sigma) = \sum_{n=1}^N \ln \left\{ \sum_{k=1}^K \pi_k \mathcal{N}(x_n | \mu_k, \Sigma_k) \right\} \quad (9.14)$$

在我们讨论如何最大化这个函数之前, 有必要强调一下由于奇异性存在的造成应用于高斯混合模型的最大似然框架中的一个大问题。为了简化起见, 我们考虑一个高斯混合模型, 它的分量的协方差矩阵为 $\Sigma_k = \sigma_k^2 \mathbf{I}$, 其中 \mathbf{I} 是一个单位矩阵, 虽然结论对于一般的协方差矩阵仍然成立。假设混合模型的第 j 个分量的均值 μ_j 与某个数据点完全相同, 即对于某个 n 值, $\mu_j = x_n$ 。这样, 这个数据点会为似然函数贡献一项, 形式为

$$\mathcal{N}(x_n | x_n, \sigma_j^2 \mathbf{I}) = \frac{1}{(2\pi)^{\frac{1}{2}}} \frac{1}{\sigma_j^D} \quad (9.15)$$

如果我们考虑极限 $\sigma_j \rightarrow 0$, 那么我们看到这一项趋于无穷大, 因此对数似然函数也会趋于无穷大。因此, 对数似然函数的最大化不是一个具有良好定义的问题, 因为这种奇异性总会出现, 会发生在任何一个“退化”到一个具体的数据点上的高斯分量上。回忆一下, 这个问题在单一的高斯分布中没有出现。为了理解不同之处, 我们注意到, 如果单一的高斯分布退化到了一个数据点上, 那么它总会给由其他数据点产生的似然函数贡献可乘的因子, 这些因子会以指数的速度趋于零, 从而使得整体的似然函数趋于零而不是无穷大。然而, 一旦我们在混合概率分布中存在 (至少) 两个分量, 其中一个分量会具有有限的方差, 因此对所有的数据点都会赋予一个有限的概率值, 而另一个分量会收缩到一个具体的数据点, 因此会给对数似然函数贡献一个不断增加的值。如图 9.7 所示。这种奇异性提供了最大似然方法中出现的过拟合现象的另一个例子。我们后面会看到, 如果我们使用贝叶斯方法, 那么这种困难之处就不会出现。但是现阶段, 我们只需注意, 将最大似然方法应用到高斯混合模型中时必须避免这种病态解, 并且寻找

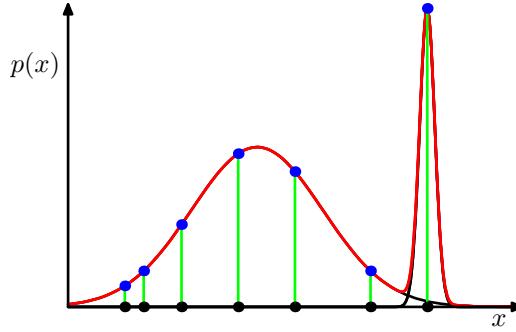


图 9.7: 似然函数的奇异性在混合高斯模型中如何出现的说明。应该将这幅图与图1.14展示的单变量高斯分布的情形进行对比，那里没有产生奇异性。

表现较好的似然函数的局部极大值。我们可以使用合适的启发式方法来避免这种奇异性，例如如果检测到高斯分量收缩到一个点，那么就将它的均值重新设定为一个随机选择的值，并且重新将它的方差设置为某个较大的值，然后继续最优化。

寻找最大似然解时的另一个问题产生于下面的事实：对于任意给定的最大似然解，一个由 K 个分量混合而成的概率分布总共会有 $K!$ 个等价的解，对应于 $K!$ 种将 K 个参数集合分配到 K 个分量上的方式。换句话说，对于参数值空间中任意给定的点，都会有 $K! - 1$ 个其他的点给出完全相同的概率分布。这个问题被称为可区分 (identifiability) 问题 (Casella and Berger, 2002)，当我们希望表示模型的参数时，这是一个重要的问题。当我们在第12章讨论具有连续潜在变量的模型时，可区分问题还会出现。但是，这个问题与找到一个好的概率模型无关，因为任意等价的解互相之间都一样好。

最大化高斯混合模型的对数似然函数 (9.14) 比单一的高斯分布的情形更加复杂。困难来源于在公式 (9.14) 中，对 k 的求和出现在对数计算内部，从而对数函数不再直接作用于高斯分布。如果我们令对数似然函数的导数等于零，那么我们不会得到一个解析解，正如我们将看到的那样。

一种方法是使用基于梯度的优化方法 (Fletcher, 1987; Nocedal and Wright, 1999; Bishop and Nabney, 2008)。虽然基于梯度的方法是可行的，并且当我们在第5章中讨论混合密度网络时起了重要的作用，但是我们现在考虑另一种方法，被称为EM算法。它具有广泛的适用性，是我们将在第10章讨论的变分推断的基础。

9.2.2 用于高斯混合模型的EM

一种优雅的并且强大的寻找带有潜在变量的模型的最大似然解的方法被称为期望最大化算法 (expectation-maximization algorithm)，或者EM算法 (Dempster et al., 1977; McLachlan and Krishnan, 1997)。稍后，我们会给出EM算法的一般形式，并且我们也会给出如何推广EM得到变分推断的框架。但是现在，我们会在高斯混合模型的问题中，给出EM算法的一种相对非形式化的描述。然而，我们要强调的是，EM算法具有广泛的适用性，实际上在本书中讨论的许多不同模型中都会遇到它。

首先，让我们写下似然函数的最大值必须满足的条件。令公式 (9.14) 中 $\ln p(\mathbf{X} \mid \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ 关于高斯分量的均值 $\boldsymbol{\mu}_k$ 的均值等于零，我们有

$$0 = \sum_{n=1}^K \underbrace{\frac{\pi_k \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_j \pi_j \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}}_{\gamma(z_{nk})} \boldsymbol{\Sigma}_k^{-1} (\mathbf{x}_n - \boldsymbol{\mu}_k) \quad (9.16)$$

其中我们使用了高斯概率分布的公式 (2.43)。注意，公式 (9.13) 给出的后验概率（或者成为“责任”）很自然地出现在了等式右侧。两侧同时乘以 $\boldsymbol{\Sigma}_k$ （假设矩阵是非奇异的），整理，可得

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{n=1}^N \gamma(z_{nk}) \mathbf{x}_n \quad (9.17)$$

其中我们定义了

$$N_k = \sum_{n=1}^N \gamma(z_{nk}) \quad (9.18)$$

我们可以将 N_k 看做分配到聚类 k 的数据点的有效数量。仔细研究这个解的形式。我们看到第 k 个高斯分量的均值 μ_k 通过对数据集里所有的数据点求加权平均的方式得到，其中数据点 x_n 的权因子由后验概率 $\gamma(z_{nk})$ 给出，而 $\gamma(z_{nk})$ 表示分量 k 对生成 x_n 的责任。

如果我们零 $\ln p(\mathbf{X} | \pi, \mu, \Sigma)$ 关于 Σ_k 的导数等于零，然后用一个类似的推理过程，使用单一高斯分布协方差矩阵的最大似然结果，我们有

$$\Sigma_k = \frac{1}{N_k} \sum_{n=1}^N \gamma(z_{nk})(x_n - \mu_k)(x_n - \mu_k)^T \quad (9.19)$$

这与一元高斯分布的对应的结果具有相同的函数形式，但是与之前一样，每个数据点都有一个权值，权值等于对应的后验概率，分母为与对应分量相关联的数据点的有效数量。

最后，我们关于混合系数 π_k 最大化 $\ln p(\mathbf{X} | \pi, \mu, \Sigma)$ 。这里我们必须考虑限制条件 (9.9)，它要求混合系数的加和等于1。使用拉格朗日乘数法，最大化下面的量

$$\ln p(\mathbf{X} | \pi, \mu, \Sigma) + \lambda \left(\sum_{k=1}^K \pi_k - 1 \right) \quad (9.20)$$

可得

$$0 = \sum_{n=1}^N \frac{\mathcal{N}(x_n | \mu_k, \Sigma_k)}{\sum_j \pi_j \mathcal{N}(x_n | \mu_j, \Sigma_j)} + \lambda \quad (9.21)$$

其中，我们再次看到了“责任”这一项。如果我们现在将两侧乘以 π_k ，然后使用公式 (9.9) 对 k 求和，我们会发现 $\lambda = -N$ 。使用这个结果消去 λ ，整理，可得

$$\pi_k = \frac{N_k}{N} \quad (9.22)$$

从而第 k 个分量的混合系数为那个分量对于解释数据点的“责任”的平均值。

值得强调的是，结果 (9.17)、(9.19) 和 (9.22) 并没有给出混合模型参数的一个解析解，因为“责任” $\gamma(z_{nk})$ 通过公式 (9.13) 以一种复杂的方式依赖于这些参数。然而，这些结果确实给出了一个简单的迭代方法来寻找问题的最大似然解。正如我们将看到的那样，这个迭代过程是 EM 算法应用于高斯混合模型的一个实例。我们首先为均值、协方差、混合系数选择一个初始值。然后，我们交替进行两个更新，被称为 E 步骤和 M 步骤，原因稍后会看到。在期望步骤 (expectation step) 或者 E 步骤中，我们使用参数的当前值计算公式 (9.13) 给出的后验概率 (也被称为“责任”)。然后，我们将计算出的概率用于最大化步骤 (maximization step) 或者 M 步骤中，使用公式 (9.17)、(9.19) 和 (9.22) 重新估计均值、方差和混合系数。注意，在进行这一步骤时，我们首先使用公式 (9.17) 计算新的均值，然后使用新的均值通过公式

(9.19) 找到协方差，这与单一高斯分布的对应结果保持一致。我们稍后会证明，每次通过 E 步骤和接下来的 M 步骤对参数的更新确保了对数似然函数的增大。在实际应用中，当对数似然函数的变化量或者参数的变化量低于某个阈值时，我们就认为算法收敛。图 9.8 给出了将两个高斯分布组成的混合概率分布的 EM 算法应用于老忠实间歇喷泉数据集的情形。这里，我们使用了两个高斯分布的混合，分布中心的初始值与图 9.1 中的 K 均值算法使用了相同的初始值，精度矩阵被初始化为正比于单位矩阵。图(a)用绿色标记出了数据点，以及初始的混合模型的配置，其中两个高斯分量的一个标准差位置的轮廓线分别用红色圆圈和蓝色圆圈标记。图(b)给出了初始 E 步骤的结果，其中每个数据点的颜色中，蓝色所占的比重等于由蓝色分量生成对应数据点的后验概率，红色所占的比重等于由红色分量生成对应数据点的后验概率。因此，对于属于两个聚类的后验概率都较大的数据点来说，颜色看起来是紫色的。图(c)给出了第一个 M 步骤之后的结果，其中蓝色高斯分布的均值被移至数据点的均值，同时根据属于蓝色类别的每个数据点的概率进行加权。换句话说，它被移到了蓝色标记数据点的质心。类似地，蓝色高斯分布的协方

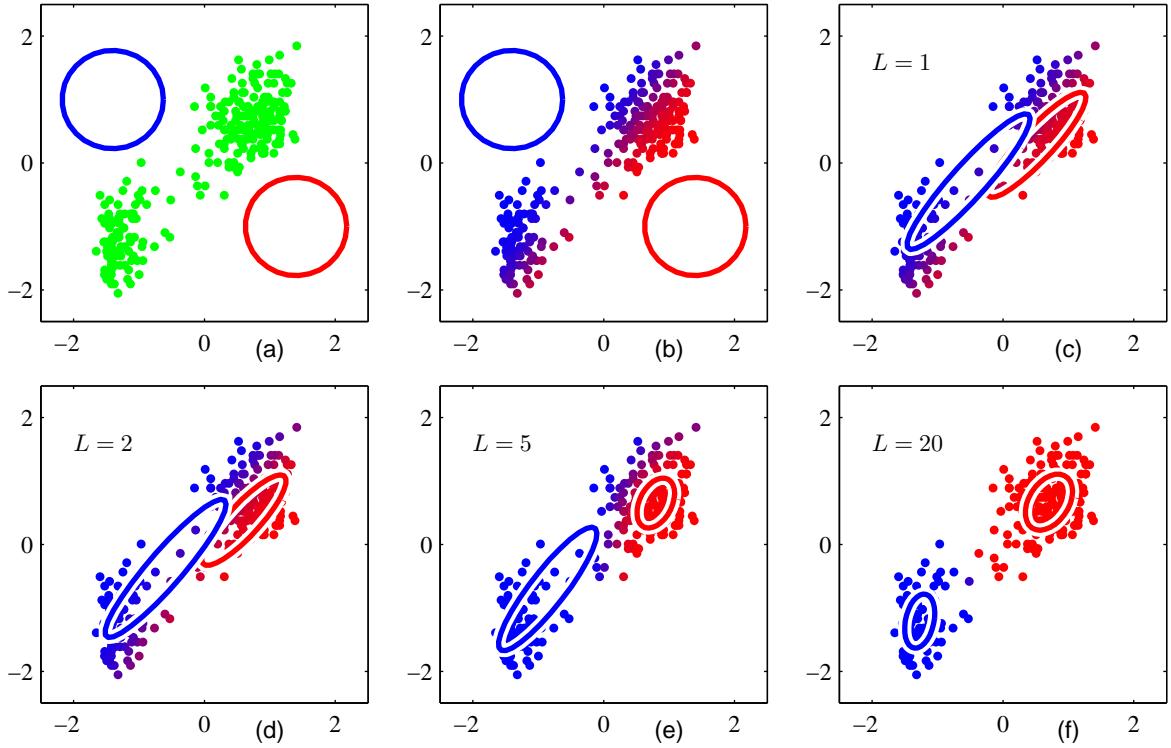


图 9.8: 对老忠实间歇喷泉数据集使用EM算法的说明, 这个数据集在图9.1中用来说 明 K 均值算法。详细说明见正文。

差被设置为蓝色标记数据点的协方差。红色分量的情形与此类似。图(d),(e)和(f)分别给出了2次、5次、20次完整的EM循环之后的结果。在图(f)中, 算法接近收敛。

注意, 与 K 均值算法相比, EM算法在达到(近似)收敛之前, 经历了更多次的迭代, 每次迭代需要更多的计算量。因此, 通常运行 K 均值算法找到高斯混合模型的一个合适的初始化值, 接下来使用EM算法进行调节。协方差矩阵可以很方便地初始化为通过 K 均值算法找到的聚类的样本协方差, 混合系数可以被设置为分配到对应类别中的数据点所占的比例。与最大化对数似然函数的基于梯度的方法相同, 算法必须避免似然函数带来的奇异性, 即高斯分量退化到一个具体的数据点。应该强调的是, 通常对数似然函数会有多个局部极大值, EM不保证找到这些极大值中最大的一个。由于高斯混合模型的EM算法非常重要, 因此我们总结如下。

给定一个高斯混合模型, 目标是关于参数(均值、协方差、混合系数)最大化似然函数。

- 初始化均值 μ_k 、协方差 Σ_k 和混合系数 π_k , 计算对数似然函数的初始值。
- **E步骤**。使用当前参数值计算“责任”。

$$\gamma(z_{nk}) = \frac{\pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^K \pi_j \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} \quad (9.23)$$

- **M步骤**。使用当前的“责任”重新估计参数。

$$\boldsymbol{\mu}_k^{\text{新}} = \frac{1}{N_k} \sum_{n=1}^N \gamma(z_{nk}) \mathbf{x}_n \quad (9.24)$$

$$\boldsymbol{\Sigma}_k^{\text{新}} = \frac{1}{N_k} \sum_{n=1}^N \gamma(z_{nk}) (\mathbf{x}_n - \boldsymbol{\mu}_k^{\text{新}})(\mathbf{x}_n - \boldsymbol{\mu}_k^{\text{新}})^T \quad (9.25)$$

$$\pi_k^{\text{新}} = \frac{N_k}{N} \quad (9.26)$$

其中

$$N_k = \sum_{n=1}^N \gamma(z_{nk}) \quad (9.27)$$

- 计算对数似然函数

$$\ln p(\mathbf{X} | \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}) = \sum_{n=1}^N \ln \left\{ \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\} \quad (9.28)$$

检查参数或者对数似然函数的收敛性。如果没有满足收敛的准则，则返回第2步。

9.3 EM的另一种观点

本节中，我们介绍EM算法的另一种观点，其中潜在变量起着重要的作用。我们首先使用一种抽象的方式讨论这种方法，然后我们再次考虑高斯混合模型的例子，来具体说明这个模型。

EM算法的目标是找到具有潜在变量的模型的最大似然解。我们将所有观测数据的集合记作 \mathbf{X} ，其中第 n 行表示 \mathbf{x}_n^T 。类似地，我们将所有潜在变量的集合记作 \mathbf{Z} ，对应的行为 \mathbf{z}_n^T 。所有模型参数的集合被记作 $\boldsymbol{\theta}$ ，因此对数似然函数为

$$\ln p(\mathbf{X} | \boldsymbol{\theta}) = \ln \left\{ \sum_{\mathbf{Z}} p(\mathbf{X}, \mathbf{Z} | \boldsymbol{\theta}) \right\} \quad (9.29)$$

注意，我们的讨论同样适用于连续潜在变量的情形，只需把对 \mathbf{Z} 的求和替换为积分即可。

一个关键的现象是，对于潜在变量的求和位于对数的内部。即使联合概率分布 $p(\mathbf{X}, \mathbf{Z} | \boldsymbol{\theta})$ 属于指数族分布，由于这个求和式的存在，边缘概率分布 $p(\mathbf{X} | \boldsymbol{\theta})$ 通常也不是指数族分布。求和式的出现阻止了对数运算直接作用于联合概率分布，使得最大似然解的形式更加复杂。

现在假定对于 \mathbf{X} 中的每个观测，我们都有潜在变量 \mathbf{Z} 的对应值。我们将 $\{\mathbf{X}, \mathbf{Z}\}$ 称为完整(complete)数据集，并且我们称实际的观测数据集 \mathbf{X} 是不完整的(incomplete)，如图9.5所示。完整数据集的对数似然函数的形式为 $\ln p(\mathbf{X}, \mathbf{Z} | \boldsymbol{\theta})$ ，并且我们假定对这个完整数据的对数似然函数进行最大化是很容易的。

然而，在实际应用中，我们没有完整数据集 $\{\mathbf{X}, \mathbf{Z}\}$ ，只有不完整的数据 \mathbf{X} 。我们关于潜在变量 \mathbf{Z} 的取值的知识仅仅来源于后验概率分布 $p(\mathbf{Z} | \mathbf{X}, \boldsymbol{\theta})$ 。由于我们不能使用完整数据的对数似然函数，因此我们反过来考虑在潜在变量的后验概率分布下，它的期望值，这对应于EM算法中的E步骤(稍后会看到)。在接下来的M步骤中，我们最大化这个期望。如果当前对于参数的估计为 $\boldsymbol{\theta}^{旧}$ ，那么一次连续的E步骤和M步骤会产生一个修正的估计 $\boldsymbol{\theta}^{新}$ 。算法在初始化时选择了参数 $\boldsymbol{\theta}_0$ 的某个起始值。对期望的使用看起来多少有些随意，但是当我们在9.4节更深入地讨论EM算法时，我们会看到这种选择的原因。

在E步骤中，我们使用当前的参数值 $\boldsymbol{\theta}^{旧}$ 寻找潜在变量的后验概率分布 $p(\mathbf{Z} | \mathbf{X}, \boldsymbol{\theta}^{旧})$ 。然后，我们使用这个后验概率分布计算完整数据对数似然函数对于一般的参数值 $\boldsymbol{\theta}$ 的期望。这个期望被记作 $\mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{旧})$ ，由下式给出。

$$\mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{旧}) = \sum_{\mathbf{Z}} p(\mathbf{Z} | \mathbf{X}, \boldsymbol{\theta}^{旧}) \ln p(\mathbf{X}, \mathbf{Z} | \boldsymbol{\theta}) \quad (9.30)$$

在M步骤中，我们通过最大化下式

$$\boldsymbol{\theta}^{新} = \arg \max_{\boldsymbol{\theta}} \mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{旧}) \quad (9.31)$$

来确定修正后的参数估计 $\boldsymbol{\theta}^{新}$ 。注意，在 $\mathcal{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^{旧})$ 的定义中，对数操作直接作用于联合概率分布 $p(\mathbf{X}, \mathbf{Z} | \boldsymbol{\theta})$ ，因此根据假设，对应的M步骤的最大化是可以计算的。

一般的EM算法总结如下。正如我们稍后会看到的那样，每个EM循环都会增大不完整数据的对数似然函数(除非已经达到局部极小值)。

给定观测变量 \mathbf{X} 和潜在变量 \mathbf{Z} 上的一个联合概率分布 $p(\mathbf{X}, \mathbf{Z} | \boldsymbol{\theta})$ ，由参数 $\boldsymbol{\theta}$ 控制，目标是关于 $\boldsymbol{\theta}$ 最大化似然函数 $p(\mathbf{X} | \boldsymbol{\theta})$ 。

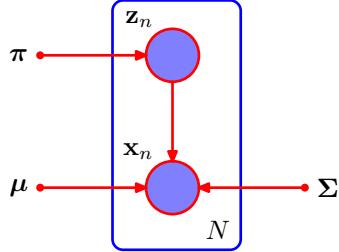


图 9.9: 本图与图9.6相同, 只是我们现在假定离散变量 z_n 以及观测变量 x_n 被观测到。

- 选择参数 $\theta^{\text{旧}}$ 的一个初始设置。
- E步骤。计算 $p(Z | X, \theta^{\text{旧}})$ 。
- M步骤。计算 $\theta^{\text{新}}$, 由下式给出。

$$\theta^{\text{新}} = \arg \max_{\theta} Q(\theta, \theta^{\text{旧}}) \quad (9.32)$$

其中

$$Q(\theta, \theta^{\text{旧}}) = \sum_Z p(Z | X, \theta^{\text{旧}}) \ln p(X, Z | \theta) \quad (9.33)$$

- 检查对数似然函数或者参数值的收敛性。如果不满足收敛准则, 那么令

$$\theta^{\text{旧}} \leftarrow \theta^{\text{新}} \quad (9.34)$$

然后回到第2步。

EM算法也可以用来寻找模型的MAP (最大后验概率) 解, 此时我们定义一个参数上的先验概率分布 $p(\theta)$ 。在这种情况下, E步骤与最大似然的情形相同, 而在M步骤中, 需要最大化的量为 $Q(\theta, \theta^{\text{旧}}) + \ln p(\theta)$ 。选择合适的先验概率分布会消除图9.7所示的奇异性。

这里, 我们考虑了使用EM算法最大化一个包含离散潜在变量的似然函数。然而, 它也适用于未观测的变量对应于数据集里的缺失值的情形。观测值的概率分布可以通过对所有变量的联合概率分布关于缺失变量求和或积分的方式得到。这样, EM算法可以用来最大化对应的似然函数。我们后面在图12.11中讨论主成分分析时, 会给出这种方法的一个应用。EM算法也适用于数据集随机缺失 (missing at random) 的情形, 即导致某个值缺失的原因不依赖于未观测的值。这种情况有很多, 例如当传感器的测量值超过某个阈值时, 传感器就不会成功地返回一个值。

9.3.1 重新考察高斯混合模型

我们现在考虑将EM算法的潜在变量观点应用与一个具体的例子, 即高斯混合模型。回忆一下, 我们的目标是最大化对数似然函数 (9.14), 它是使用观测数据集 X 进行计算的。我们看到这个计算比单一高斯分布的情形更困难, 因为对 k 的求和出现在对数运算内部。假设除了观测数据集 X , 我们还有对应的离散变量 Z 的值。回忆一下, 图9.5(a)给出了一个“完整”的数据集, 即给出了每个数据点由哪个分量生成, 而图9.5(b)给出了对应的“不完整”数据集。完整数据的图模型如图9.9所示。

现在考虑对完整数据 $\{X, Z\}$ 进行最大化。根据公式 (9.10) 和公式 (9.11), 似然函数的形式为

$$p(X, Z | \mu, \Sigma, \pi) = \prod_{n=1}^N \prod_{k=1}^K \pi_k^{z_{nk}} \mathcal{N}(x_n | \mu_k, \Sigma_k)^{z_{nk}} \quad (9.35)$$

其中 z_{nk} 表示 z_n 的第 k 个分量。取对数, 我们有

$$\ln p(X, Z | \mu, \Sigma, \pi) = \sum_{n=1}^N \sum_{k=1}^K z_{nk} \{ \ln \pi_k + \ln \mathcal{N}(x_n | \mu_k, \Sigma_k) \} \quad (9.36)$$

与不完整数据的对数似然函数 (9.14) 进行对比, 我们看到在 k 上的求和与对数运算的顺序交换了。对数运算现在直接作用于高斯分布上, 而高斯分布本身是指数族分布的一个成员。丝毫不令人惊讶, 这种方法产生了最大似然问题的一个简单得多的解, 说明如下。首先考虑关于均值和协方差的最大化。由于 \mathbf{z}_n 是一个 K 维向量, 并且只有一个元素等于 1, 其他所有元素均为 0, 因此完整数据的对数似然函数仅仅是 K 个独立的贡献的加和, 每个混合分量都有一个贡献。于是关于均值或协方差的最大化与单一高斯分布的情形完全相同, 唯一的区别是它只涉及到被“分配”到那个分量的数据点的子集。对于关于混合系数的最大化问题, 我们注意到由于加和限制 (9.9) 的存在, 不同 k 值的混合系数相互关联。与之前一样, 可以使用拉格朗日乘数法进行优化, 结果为

$$\pi_k = \frac{1}{N} \sum_{n=1}^N z_{nk} \quad (9.37)$$

从而混合系数等于分配到对应分量的数据点所占的比例。

因此我们看到, 完整数据的对数似然函数可以用一种简单的方法求出最大值的解析解。然而, 在实际应用中, 我们并没有潜在变量的值, 因此, 与之前的讨论一样, 我们考虑完整数据对数似然函数关于潜在变量后验概率分布的期望。使用公式 (9.10)、公式 (9.11) 以及贝叶斯定理, 我们看到这个后验概率分布的形式为

$$p(\mathbf{Z} | \mathbf{X}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}) \propto \prod_{n=1}^N \prod_{k=1}^K [\pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)]^{z_{nk}} \quad (9.38)$$

因此后验概率分布可以在 n 上进行分解, 从而 $\{z_n\}$ 是独立的。通过观察图 9.6 中的有向图然后使用 d-划分准则, 很容易证明这一点。这样, 在这个后验概率分布下, 指示值 z_{nk} 的期望为

$$\begin{aligned} \mathbb{E}[z_{nk}] &= \frac{\sum_{\mathbf{z}_n} z_{nk} \prod_{k'} [\pi_{k'} \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_{k'}, \boldsymbol{\Sigma}_{k'})]^{z_{nk'}}}{\sum_{\mathbf{z}_n} \prod_j [\pi_j \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)]^{z_{nj}}} \\ &= \frac{\pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^K \pi_j \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} = \gamma(z_{nk}) \end{aligned} \quad (9.39)$$

它就是 k 分量对于数据点 \mathbf{x}_n 的“责任”。于是, 完整数据的对数似然函数的期望值为

$$\mathbb{E}_{\mathbf{Z}}[\ln p(\mathbf{X}, \mathbf{Z} | \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi})] = \sum_{n=1}^N \sum_{k=1}^K \gamma(z_{nk}) \{\ln \pi_k + \ln \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\} \quad (9.40)$$

我们现在可以按照下面的方式进行处理。首先, 我们为参数 $\boldsymbol{\mu}^{\text{旧}}, \boldsymbol{\Sigma}^{\text{旧}}, \boldsymbol{\pi}^{\text{旧}}$ 选择某个初始值, 使用这些初始值计算“责任” (E 步骤)。然后我们保持“责任”固定, 关于 $\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k$ 和 π_k 最大化 (9.40) (M 步骤)。与之前一样, 这会得到由公式 (9.17)、(9.19) 和 (9.22) 给出的 $\boldsymbol{\mu}^{\text{新}}, \boldsymbol{\Sigma}^{\text{新}}$ 和 $\boldsymbol{\pi}^{\text{新}}$ 的解析解。这与之前推导的高斯混合模型的 EM 算法完全相同。当我们在 9.4 节证明 EM 算法的收敛性时, 我们会更加深刻地认识到完整数据的对数似然函数的作用。

9.3.2 与 K 均值的关系

对比高斯模型的 EM 算法与 K 均值算法, 可以看到二者有很强的相似性。 K 均值算法对数据点的聚类进行了“硬”分配, 即每个数据点只属于唯一的聚类, 而 EM 算法基于后验概率分布, 进行了一个“软”分配。实际上, 我们可以将 K 均值算法看成高斯混合模型的 EM 算法的一个特殊的极限情况, 如下所述。

考虑一个高斯混合模型, 其中混合分量的协方差矩阵为 $\epsilon \mathbf{I}$, ϵ 是一个被所有分量共享的方差参数, \mathbf{I} 是单位矩阵, 从而

$$p(\mathbf{x} | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \frac{1}{(2\pi\epsilon)^{\frac{D}{2}}} \exp \left\{ -\frac{1}{2\epsilon} \|\mathbf{x} - \boldsymbol{\mu}_k\|^2 \right\} \quad (9.41)$$

我们现在考虑 K 个这种形式的高斯分布组成的混合模型的EM算法，其中我们将 ϵ 看做一个固定的常数，而不是一个需要重新估计的参数。根据公式 (9.13)，对于一个特定的数据点 \mathbf{x}_n ，后验概率（或者“责任”）为

$$\gamma(z_{nk}) = \frac{\pi_k \exp\left\{-\frac{\|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2}{2\epsilon}\right\}}{\sum_j \pi_j \exp\left\{-\frac{\|\mathbf{x}_n - \boldsymbol{\mu}_j\|^2}{2\epsilon}\right\}} \quad (9.42)$$

如果我们考虑极限情况 $\epsilon \rightarrow 0$ ，那么我们看到，在分母中， $\|\mathbf{x}_n - \boldsymbol{\mu}_j\|^2$ 最小的项将会最慢地趋近于零，因此对于数据点 \mathbf{x}_n ，只有项 j 的“责任” $\gamma(z_{nj})$ 趋近于1，其他的项的“责任” $\gamma(z_{nk})$ 都趋近于0。因此，在这种极限情况下，我们得到了对数据点聚类的一个硬分配，与 K 均值算法相同，从而 $\gamma(z_{nk}) \rightarrow r_{nk}$ ，其中 r_{nk} 由公式 (9.2) 定义。因此，每个数据点都被分配为距离最近的均值的聚类。

这样，公式 (9.17) 给出的 $\boldsymbol{\mu}_k$ 的EM重估计就简化为了 K 均值的结果 (9.4)。注意，混合系数 (9.22) 的重估计公式仅仅将 π_k 的值重新设置为等于分配到聚类 k 中的数据点的比例，虽然这些参数在算法中不再起作用。

最后，在极限 $\epsilon \rightarrow 0$ 的情况下，公式 (9.40) 给出的完整数据的对数似然函数变成了

$$\mathbb{E}_{\mathbf{Z}}[\ln p(\mathbf{X}, \mathbf{Z} | \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi})] \rightarrow -\frac{1}{2} \sum_{n=1}^N \sum_{k=1}^K r_{nk} \|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2 + \text{常数} \quad (9.43)$$

因此在极限的情况下，最大化完整数据对数似然函数的期望等价于最小化公式 (9.1) 给出的 K 均值算法的失真度量 J 。

注意， K 均值算法没有估计聚类的协方差，而是只估计了聚类的均值。一个带有通常的协方差矩阵的硬分配版本的高斯混合模型被称为椭圆 K 均值算法 (elliptical K-means algorithm)，由Sung and Poggio (1994) 提出。

9.3.3 伯努利分布的混合

目前为止在本章中，我们的注意力集中于由混合高斯模型描述的连续变量上的概率分布。作为混合模型的另一个例子，同时为了在一个不同的问题中说明EM算法，我们现在讨论由伯努利分布描述的离散二值变量的混合。这个模型也被称为潜在类别分析 (latent class analysis)

(Lazarsfeld and Henry, 1968; McLachlan and Peel, 2000)。这个模型不仅具有实际应用的重要性，还是我们考虑离散变量上的马尔科夫模型的基础。

考虑 D 个二值变量 x_i 组成的集合，其中 $i = 1, \dots, D$ ，每个变量都由一个参数为 μ_i 的伯努利分布控制，即

$$p(\mathbf{x} | \boldsymbol{\mu}) = \prod_{i=1}^D \mu_i^{x_i} (1 - \mu_i)^{1-x_i} \quad (9.44)$$

其中 $\mathbf{x} = (x_1, \dots, x_D)^T$ 且 $\boldsymbol{\mu} = (\mu_1, \dots, \mu_D)^T$ 。我们看到，给定 $\boldsymbol{\mu}$ 的条件下，各个变量 x_i 是独立的。很容易看出，这个分布的均值和方差为

$$\mathbb{E}[\mathbf{x}] = \boldsymbol{\mu} \quad (9.45)$$

$$\text{cov}[\mathbf{x}] = \text{diag}\{\mu_i(1 - \mu_i)\} \quad (9.46)$$

现在让我们考虑这种分布的有限混合，即

$$p(\mathbf{x} | \boldsymbol{\mu}, \boldsymbol{\pi}) = \sum_{k=1}^K \pi_k p(\mathbf{x} | \boldsymbol{\mu}_k) \quad (9.47)$$

其中 $\boldsymbol{\mu} = \{\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_K\}$, $\boldsymbol{\pi} = \{\pi_1, \dots, \pi_K\}$ ，且

$$p(\mathbf{x} | \boldsymbol{\mu}_k) = \prod_{i=1}^D \mu_{ki}^{x_i} (1 - \mu_{ki})^{1-x_i} \quad (9.48)$$

这个混合分布的均值和方差为

$$\mathbb{E}[\mathbf{x}] = \sum_{k=1}^K \pi_k \boldsymbol{\mu}_k \quad (9.49)$$

$$\text{cov}[\mathbf{x}] = \sum_{k=1}^K \pi_k \{\boldsymbol{\Sigma}_k + \boldsymbol{\mu}_k \boldsymbol{\mu}_k^T\} - \mathbb{E}[\mathbf{x}] \mathbb{E}[\mathbf{x}]^T \quad (9.50)$$

其中 $\boldsymbol{\Sigma}_k = \text{diag}\{\mu_{ki}(1 - \mu_{ki})\}$ 。由于协方差矩阵 $\text{cov}[\mathbf{x}]$ 不再是对角矩阵，因此混合分布可以描述变量之间的相关性，这与单一的伯努利分布不同。

如果我们有一个数据集 $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ ，那么这个模型的对数似然函数为

$$\ln p(\mathbf{X} | \boldsymbol{\mu}, \boldsymbol{\pi}) = \sum_{n=1}^N \ln \left\{ \sum_{k=1}^K \pi_k p(\mathbf{x}_n | \boldsymbol{\mu}_k) \right\} \quad (9.51)$$

与之前一样，我们看到求和运算位于对数运算内部，从而最大似然解没有解析解。

我们现在推导混合伯努利分布的最大化似然函数的EM算法。为了完成这件事，我们首先显式地引入一个潜在变量 \mathbf{z} ，它与 \mathbf{x} 的每个实例相关联。与高斯混合模型的情形相同， $\mathbf{z} = (z_1, \dots, z_K)^T$ 是一个二值 K 维变量，其中只有一个元素等于 1，其余元素等于 0。这样，给定潜在变量，我们可以写出 \mathbf{x} 的条件概率分布，形式为

$$p(\mathbf{x} | \mathbf{z}, \boldsymbol{\mu}) = \prod_{k=1}^K p(\mathbf{x} | \boldsymbol{\mu}_k)^{z_k} \quad (9.52)$$

而潜在变量的先验概率分布与高斯混合模型的形式相同，即

$$p(\mathbf{z} | \boldsymbol{\pi}) = \prod_{k=1}^K \pi_k^{z_k} \quad (9.53)$$

如果我们将 $p(\mathbf{x} | \mathbf{z}, \boldsymbol{\mu})$ 和 $p(\mathbf{z} | \boldsymbol{\pi})$ 相乘，然后对 \mathbf{z} 求和，我们就恢复出了公式 (9.47)。

为了推导EM算法，我们首先写出完整数据的对数似然函数，形式为

$$\begin{aligned} \ln p(\mathbf{X}, \mathbf{Z} | \boldsymbol{\mu}, \boldsymbol{\pi}) &= \sum_{n=1}^N \sum_{k=1}^K z_{nk} \left\{ \ln \pi_k \right. \\ &\quad \left. + \sum_{i=1}^D [x_{ni} \ln \mu_{ki} + (1 - x_{ni}) \ln(1 - \mu_{ki})] \right\} \end{aligned} \quad (9.54)$$

其中 $\mathbf{X} = \{\mathbf{x}_n\}$ 且 $\mathbf{Z} = \{\mathbf{z}_n\}$ 。接下来我们取完整数据对数似然函数关于潜在变量后验概率分布的期望，得

$$\begin{aligned} \mathbb{E}_{\mathbf{Z}}[\ln p(\mathbf{X}, \mathbf{Z} | \boldsymbol{\mu}, \boldsymbol{\pi})] &= \sum_{n=1}^N \sum_{k=1}^K \gamma(z_{nk}) \left\{ \ln \pi_k \right. \\ &\quad \left. + \sum_{i=1}^D [x_{ni} \ln \mu_{ki} + (1 - x_{ni}) \ln(1 - \mu_{ki})] \right\} \end{aligned} \quad (9.55)$$

其中 $\gamma(z_{nk}) = \mathbb{E}[z_{nk}]$ 是给定数据点 \mathbf{x}_n 的条件下，分量 k 的后验概率分布，或者“责任”。在 E 步骤中，这些后验概率使用贝叶斯定理计算，形式为

$$\begin{aligned} \gamma(z_{nk}) &= \mathbb{E}[z_{nk}] = \frac{\sum_{\mathbf{z}_n} z_{nk} \prod_{k'} [\pi_{k'} p(\mathbf{x}_n | \boldsymbol{\mu}_{k'})]^{z_{nk'}}}{\sum_{\mathbf{z}_n} \prod_j [\pi_j p(\mathbf{x}_n | \boldsymbol{\mu}_j)]^{z_{nj}}} \\ &= \frac{\pi_k p(\mathbf{x}_n | \boldsymbol{\mu}_k)}{\sum_{j=1}^K \pi_j p(\mathbf{x}_n | \boldsymbol{\mu}_j)} \end{aligned} \quad (9.56)$$

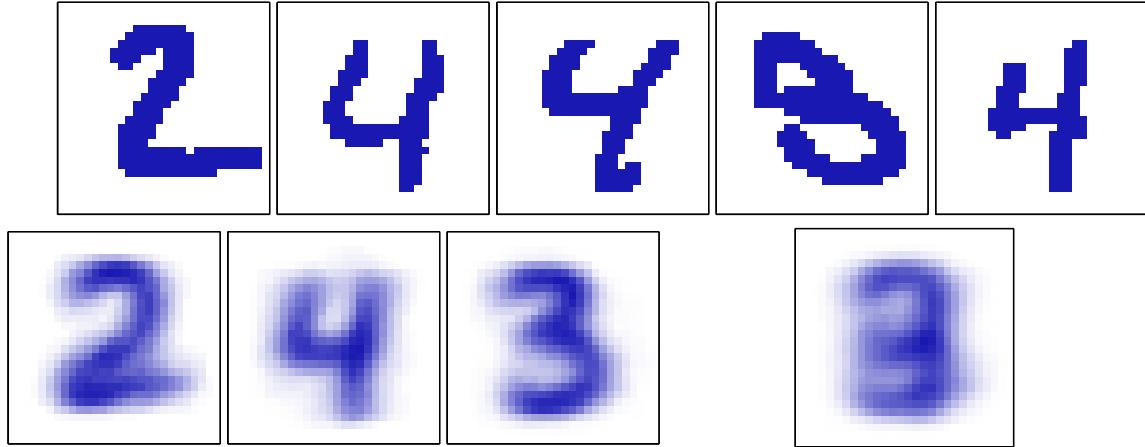


图 9.10: 伯努利混合模型的例子。上面一行给出了将手写数字数据集从灰度图转化为二值图之后的例子，转化时使用的阈值为0.5。下面一行中，最开始的三张图像展示了混合模型的三个分量的参数 μ_{ki} 。作为对比，我们也使用一个单一的多元伯努利分布对同样的数据进行了拟合，同样使用了最大似然方法。这对应于将每个像素点求平均，结果如下面一行最右侧的图所示。

如果我们考虑在公式 (9.55) 中对 n 求和，我们看到“责任”只出现在两项中，这两项可以写成

$$N_k = \sum_{n=1}^N \gamma(z_{nk}) \quad (9.57)$$

$$\bar{\mathbf{x}}_k = \frac{1}{N_k} \sum_{n=1}^N \gamma(z_{nk}) \mathbf{x}_n \quad (9.58)$$

其中 N_k 是与分量 k 关联的数据点的有效数量。在M步骤中，我们关于参数 μ_k 和 π 最大化完整数据对数似然函数的期望。如果我们令公式 (9.55) 关于 μ_k 的导数等于零，整理可得

$$\mu_k = \bar{x}_k \quad (9.59)$$

我们看到，分量 k 的均值组成的集合等于数据的加权平均值，权系数为分量 k 对于数据点的“责任”。对于关于 π_k 的最大化，我们需要引入一个拉格朗日乘数来满足限制条件 $\sum_k \pi_k = 1$ 。采用与高斯混合模型中类似的步骤，我们有

$$\pi_k = \frac{N_k}{N} \quad (9.60)$$

这与直觉相符，即分量 k 的混合系数等于数据集里那个分量的数据点所占的比例。

注意，与混合高斯模型不同，不存在似然函数趋于无穷大的奇异性。我们注意到似然函数是有界的，因为 $0 \leq p(\mathbf{x}_n | \boldsymbol{\mu}_k) \leq 1$ 。确实存在似然函数趋于零的奇异性，但是只要EM在初始化时没有选择一个病态的起始点，这些点就不会被找到，因为EM算法总是增大似然函数的值，直达到局部极大值。在图9.10中，我们说明了用伯努利混合模型对手写数字建模的结果。这里，数字的图像被转化为了二值向量，转化的方法是令所有超过0.5的值等于1，令其他的值等于0。我们现在有 $N = 600$ 张这样的图像，由数字“2”、“3”、“4”组成。我们将 $K = 3$ 个伯努利分布进行混合，运行EM算法进行10轮迭代。混合系数被初始化为 $\pi_k = \frac{1}{K}$ ，参数 μ_{kj} 被设置为随机值，这个随机值服从区间(0.25, 0.75)上的均匀分布，且满足限制 $\sum_j \mu_{kj} = 1$ 。我们看到，三个伯努利分布的混合能够找到数据里里对应于不同数字的三个聚类。

伯努利分布参数的共轭先验是Beta分布。我们已经看到一个Beta先验分布等价于引入 x 的额外的有效观测。类似地，我们可以引入伯努利混合模型的先验分布，然后使用EM算法最大化后验概率分布。

很容易将对伯努利混合模型的分析推广到具有 $M > 2$ 个状态的离散变量多项式分布的情形（由公式 (2.26) 定义）。与之前一样，在必要的条件下，我们可以引入模型参数的狄利克雷先验分布。

9.3.4 贝叶斯线性回归的EM算法

作为说明EM算法应用的第三个例子，我们回到贝叶斯线性回归的证据近似问题。在3.5.2节，我们通过计算模型证据然后令导数等于零的方式得到了超参数 α 和 β 的值。我们现在使用另一种寻找 α 和 β 的方法，这种方法基于EM算法。回忆一下，我们的目标是关于 α 和 β 最大化由公式(3.77)给出的证据函数 $p(\mathbf{t} | \alpha, \beta)$ 。由于参数 \mathbf{w} 已经被积分出去，因此我们可以将其当做一个潜在变量，因此我们可以使用EM算法来优化边缘似然函数。在E步骤中，我们计算在给定当前的 α 和 β 的条件下， \mathbf{w} 的后验概率分布，然后使用这个找到完整数据对数似然函数的期望。在M步骤中，我们关于 α 和 β 最大化这个量。我们已经推导出了 \mathbf{w} 的后验概率分布，即公式(3.49)。这样，完整数据的对数似然函数为

$$\ln p(\mathbf{t}, \mathbf{w} | \alpha, \beta) = \ln p(\mathbf{t} | \mathbf{w}, \beta) + \ln p(\mathbf{w} | \alpha) \quad (9.61)$$

其中似然函数 $p(\mathbf{t} | \mathbf{w}, \beta)$ 和先验概率分布 $p(\mathbf{w} | \alpha)$ 分别由公式(3.10)和公式(3.52)给出。关于 \mathbf{w} 的后验概率分布取期望，可得

$$\begin{aligned} \mathbb{E}[\ln p(\mathbf{t}, \mathbf{w} | \alpha, \beta)] &= \frac{M}{2} \ln\left(\frac{\alpha}{2\pi}\right) - \frac{\alpha}{2} \mathbb{E}[\mathbf{w}^T \mathbf{w}] + \frac{N}{2} \ln\left(\frac{\beta}{2\pi}\right) \\ &\quad - \frac{\beta}{2} \sum_{n=1}^N \mathbb{E}[(t_n - \mathbf{w}^T \boldsymbol{\phi}_n)^2] \end{aligned} \quad (9.62)$$

令它关于 α 的导数等于零，我们得到了M步骤的重新估计方程

$$\alpha = \frac{M}{\mathbb{E}[\mathbf{w}^T \mathbf{w}]} = \frac{M}{\mathbf{m}_N^T \mathbf{m}_N + \text{Tr}(\mathbf{S}_N)} \quad (9.63)$$

对于 β ，结果类似。

注意，这个重新估计方程与直接从证据函数推导出的对应的结果(3.92)的形式稍有不同。然而，两种形式都涉及到了对一个 $M \times M$ 的矩阵进行计算、求逆（或者特征分解），因此在每轮迭代时的计算代价是可比的。

这两种确定 α 的方法显然应该收敛到同样的结果（假设它们找到证据函数的同一个局部极大值）。可以用下面的方法验证。首先注意到 γ 的定义为

$$\gamma = M - \alpha \sum_{i=1}^M \frac{1}{\lambda_i + \alpha} = M - \alpha \text{Tr}(\mathbf{S}_N) \quad (9.64)$$

在证据函数的驻点处，重估计方程(3.92)一定成立，因此我们可以将 γ 替换掉，得到

$$\alpha \mathbf{m}_N^T \mathbf{m}_N = \gamma = M - \alpha \text{Tr}(\mathbf{S}_N) \quad (9.65)$$

解出 α ，我们得到了公式(9.63)的结果，这就是EM的重新估计方程。

作为最后一个例子，我们考虑一个密切相关的模型，即7.2.1节讨论的用于回归问题的相关向量机。那里，我们直接最大化边缘似然函数来推导超参数 α 和 β 的重估计方程。这里，我们考虑另一种方法，即把权向量 \mathbf{w} 看成一个潜在变量，然后使用EM算法。E步骤涉及到寻找权值的后验概率分布，这由公式(7.81)给出。在M步骤中，我们最大化完整数据对数似然函数的期望，定义为

$$\mathbb{E}_{\mathbf{w}}[\ln\{p(\mathbf{t} | \mathbf{X}, \mathbf{w}, \beta)p(\mathbf{w} | \alpha)\}] \quad (9.66)$$

其中期望值是关于使用旧的参数计算的后验概率分布进行计算的。为了计算新的参数值，我们关于 α 和 β 进行最大化，有

$$\alpha_i^{\text{新}} = \frac{1}{m_i^2 + \Sigma_{ii}} \quad (9.67)$$

$$(\beta^{\text{新}})^{-1} = \frac{\|\mathbf{t} - \Phi \mathbf{m}\|^2 + \beta^{-1} \sum_i \gamma_i}{N} \quad (9.68)$$

这些重估计方程在形式上等价于直接对边缘似然函数进行最大化得到的重估计方程。



图 9.11: 由公式 (9.70) 给出的分解的说明, 它对于分布 $q(\mathbf{Z})$ 的任意选择都成立。由于Kullback-Leibler散度满足 $\text{KL}(q \parallel p) \geq 0$, 因此我们看到 $L(q, \theta)$ 是对数似然函数 $\ln p(\mathbf{X} | \boldsymbol{\theta})$ 的下界。

9.4 一般形式的EM算法

期望最大化算法, 或者EM算法, 是寻找具有潜在变量的概率模型的最大似然解的一种通用的方法 (Dempster et al., 1977; MaLachlan and Krishnan, 1997)。这里, 我们给出一般形式的EM算法, 并且在这个过程中, 会证明9.2节和9.3节在讨论高斯混合模型时启发式地推导出的EM算法确实最大化了似然函数 (Csiszàr and Tusnàdy, 1984; Hathaway, 1986; Neal and Hinton, 1999)。我们的讨论也构成了变分推断框架推导的基础。

考虑一个概率模型, 其中我们将所有的观测变量联合起来记作 \mathbf{X} , 将所有的隐含变量记作 \mathbf{Z} 。联合概率分布 $p(\mathbf{X}, \mathbf{Z} | \boldsymbol{\theta})$ 由一组参数控制, 记作 $\boldsymbol{\theta}$ 。我们的目标是最大化似然函数

$$p(\mathbf{X} | \boldsymbol{\theta}) = \sum_{\mathbf{Z}} p(\mathbf{X}, \mathbf{Z} | \boldsymbol{\theta}) \quad (9.69)$$

这里, 我们假设 \mathbf{Z} 是离散的, 但是当 \mathbf{Z} 是连续变量或者离散变量与连续变量的组合时, 方法是完全相同的, 只需把求和换成适当的积分即可。

我们假设直接最优化 $p(\mathbf{X} | \boldsymbol{\theta})$ 比较困难, 但是最优化完整数据似然函数 $p(\mathbf{X}, \mathbf{Z} | \boldsymbol{\theta})$ 就容易得多。接下来, 我们引入一个定义在潜在变量上的分布 $q(\mathbf{Z})$ 。我们观察到, 对于任意的 $q(\mathbf{Z})$, 下面的分解成立

$$\ln p(\mathbf{X} | \boldsymbol{\theta}) = L(q, \boldsymbol{\theta}) + \text{KL}(q \parallel p) \quad (9.70)$$

其中, 我们定义了

$$L(q, \boldsymbol{\theta}) = \sum_{\mathbf{Z}} q(\mathbf{Z}) \ln \left\{ \frac{p(\mathbf{X}, \mathbf{Z} | \boldsymbol{\theta})}{q(\mathbf{Z})} \right\} \quad (9.71)$$

$$\text{KL}(q \parallel p) = - \sum_{\mathbf{Z}} q(\mathbf{Z}) \ln \left\{ \frac{p(\mathbf{Z} | \mathbf{X}, \boldsymbol{\theta})}{q(\mathbf{Z})} \right\} \quad (9.72)$$

注意, $L(q, \boldsymbol{\theta})$ 是概率分布 $q(\mathbf{Z})$ 的一个泛函 (关于泛函的讨论, 见附录D), 并且是参数 $\boldsymbol{\theta}$ 的一个函数。值得仔细研究的是表达式 (9.71) 和 (9.72) 的形式, 特别地, 需要注意, 二者的符号相反, 并且 $L(q, \boldsymbol{\theta})$ 包含了 \mathbf{X} 和 \mathbf{Z} 的联合概率分布, 而 $\text{KL}(q \parallel p)$ 包含了给定 \mathbf{X} 的条件下, \mathbf{Z} 的条件概率分布。为了验证公式 (9.70) 给出的分解方式, 我们首先使用概率的乘积规则, 可得

$$\ln p(\mathbf{X}, \mathbf{Z} | \boldsymbol{\theta}) = \ln p(\mathbf{Z} | \mathbf{X}, \boldsymbol{\theta}) + \ln p(\mathbf{X} | \boldsymbol{\theta}) \quad (9.73)$$

然后代入 $L(q, \boldsymbol{\theta})$ 的表达式。这得到了两项, 一项消去了 $\text{KL}(q \parallel p)$, 而另一项给出了所需的对数似然函数 $\ln p(\mathbf{X} | \boldsymbol{\theta})$, 其中我们用到了归一化的概率分布 $q(\mathbf{Z})$ 的积分等于1的事实。

根据公式 (9.72), 我们看到 $\text{KL}(q \parallel p)$ 是 $q(\mathbf{Z})$ 和后验概率分布 $p(\mathbf{Z} | \mathbf{X}, \boldsymbol{\theta})$ 之间的Kullback-Leibler散度。回忆一下, Kullback-Leibler散度满足 $\text{KL}(q \parallel p) \geq 0$, 当且仅当 $q(\mathbf{Z}) = p(\mathbf{Z} | \mathbf{X}, \boldsymbol{\theta})$ 时等号成立。因此, 根据公式 (9.70), $L(q, \boldsymbol{\theta}) \leq \ln p(\mathbf{X} | \boldsymbol{\theta})$, 换句话说, $L(q, \boldsymbol{\theta})$ 是 $\ln p(\mathbf{X} | \boldsymbol{\theta})$ 的一个下界。图9.11说明了公式 (9.70) 的分解。

EM算法是一个两阶段的迭代优化算法, 用于寻找最大似然解。我们可以使用公式 (9.70) 来定义EM算法, 证明它确实最大化了对数似然函数。假设参数向量的当前值为 $\boldsymbol{\theta}^{\text{旧}}$ 。在E步骤

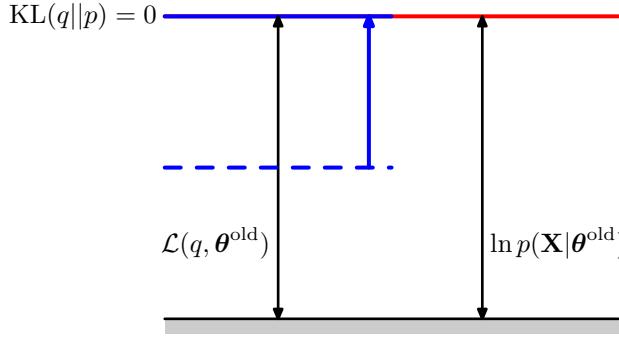


图 9.12: EM 算法的E步骤的说明。 q 分布被设置为当前参数值 $\theta^{\text{旧}}$ 下的后验概率分布，这使得下界上移到与对数似然函数值相同的位置，此时KL散度为零。

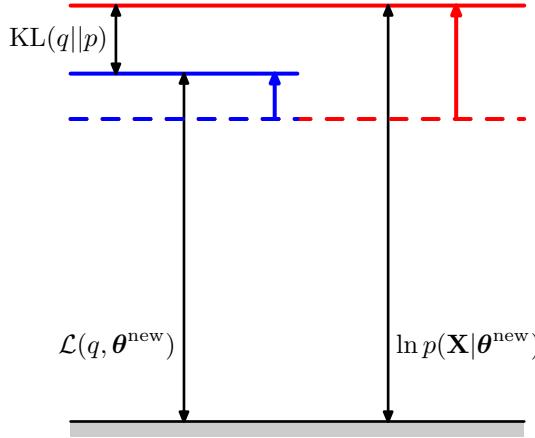


图 9.13: EM 算法的M步骤的说明。分布 $q(Z)$ 保持固定，下界 $L(q, \theta)$ 关于参数向量 θ 最大化，得到修正值 $\theta^{\text{新}}$ 。由于KL散度非负，因此这使得对数似然函数 $\ln p(\mathbf{X} | \theta)$ 的增量至少与下界的增量相等。

中，下界 $L(q, \theta^{\text{旧}})$ 关于 $q(Z)$ 被最大化，而 $\theta^{\text{旧}}$ 保持固定。最大化问题的解很容易看出来。我们注意到 $\ln p(\mathbf{X} | \theta^{\text{旧}})$ 不依赖于 $q(Z)$ ，因此 $L(q, \theta^{\text{旧}})$ 的最大值出现在Kullback-Leibler散度等于零的时候，换句话说，最大值出现在 $q(Z)$ 与后验概率分布 $p(Z | \mathbf{X}, \theta^{\text{旧}})$ 相等的时候。此时，下界等于对数似然函数，如图9.12所示。

在接下来的M步骤中，分布 $q(Z)$ 保持固定，下界 $L(q, \theta)$ 关于 θ 进行最大化，得到了某个新值 $\theta^{\text{新}}$ 。这会使得下界 L 增大（除非已经达到了极值），这会使得对应的对数似然函数增大。由于概率分布 q 由旧的参数值确定，并且在M步骤中保持固定，因此它不会等于新的后验概率分布 $p(Z | \mathbf{X}, \theta^{\text{新}})$ ，从而KL散度非零。于是，对数似然函数的增加量大于下界的增加量，如图9.13所示。如果我们将 $q(Z) = p(Z | \mathbf{X}, \theta^{\text{旧}})$ 代入公式 (9.71)，我们会看到，在E步骤之后，下界的形式为

$$\begin{aligned} L(q, \theta) &= \sum_Z p(Z | \mathbf{X}, \theta^{\text{旧}}) \ln p(\mathbf{X}, Z | \theta) - \sum_Z p(Z | \mathbf{X}, \theta^{\text{旧}}) \ln p(Z | \mathbf{X}, \theta^{\text{旧}}) \\ &= Q(\theta, \theta^{\text{旧}}) + \text{常数} \end{aligned} \quad (9.74)$$

其中，常数就是分布 q 的熵，因此与 θ 无关。从而在M步骤中，最大化的量是完整数据对数似然函数的期望，正如我们之前在混合高斯模型的情形中看到的那样。注意，我们进行优化的变量 θ 只出现在对数运算内部。如果联合概率分布 $p(Z, \mathbf{X} | \theta)$ 由指数族分布的成员组成，或者由指数族分布成员的乘积组成，那么我们看到对数运算会抵消指数运算，从而使得M步骤通常比最大化对应的不完整数据对数似然函数 $p(\mathbf{X} | \theta)$ 要容易得多。

EM算法的计算也可以被看做参数空间中的运算，如图9.14所示。这里，红色曲线表示（不完整数据）对数似然函数，它的最大值是我们想要得到的。我们首先选择某个初始的参数值 $\theta^{\text{旧}}$ ，然后在第一个E步骤中，我们计算潜在变量上的后验概率分布，得到了 $L(q, \theta^{\text{旧}})$ 的一个



图 9.14: EM 算法涉及到交替计算当前参数值下的对数似然函数的下界以及最大化下界的值得到新的参数值。完整的讨论见正文。

更小的下届，它的值等于在 $\theta^{\text{旧}}$ 处的对数似然函数值，用蓝色曲线表示。注意，下界与对数似然函数在 $\theta^{\text{旧}}$ 处以切线的方式连接，因此两条曲线的梯度相同。这个界是一个凹函数，对于指数族分布的混合分布来说，有唯一的一个最大值。在 M 步骤中，下界被最大化，得到了新的值 $\theta^{\text{新}}$ ，这个值给出了比 $\theta^{\text{旧}}$ 处更大的对数似然函数值。接下来的 E 步骤构建了一个新的下界，它在 $\theta^{\text{新}}$ 处与对数似然函数切线连接，用绿色曲线表示。

对于独立同分布数据集的特殊情形， \mathbf{X} 由 N 个数据点 $\{x_n\}$ 组成，而 \mathbf{Z} 由 N 个对应的潜在变量 $\{z_n\}$ 组成，其中 $n = 1, \dots, N$ 。根据独立性假设，我们有 $p(\mathbf{X}, \mathbf{Z}) = \prod_n p(x_n, z_n)$ ，并且通过关于 $\{z_n\}$ 求边缘概率分布，我们有 $p(\mathbf{X}) = \prod_n p(x_n)$ 。使用加和规则和乘积规则，我们看到在 E 步骤中计算的后验概率分布的形式为

$$p(\mathbf{Z} | \mathbf{X}, \boldsymbol{\theta}) = \frac{p(\mathbf{X}, \mathbf{Z} | \boldsymbol{\theta})}{\sum_{\mathbf{Z}} p(\mathbf{X}, \mathbf{Z} | \boldsymbol{\theta})} = \frac{\prod_{n=1}^N p(x_n, z_n | \boldsymbol{\theta})}{\sum_{\mathbf{Z}} \prod_{n=1}^N p(x_n, z_n | \boldsymbol{\theta})} = \prod_{n=1}^N p(z_n | x_n, \boldsymbol{\theta}) \quad (9.75)$$

因此后验概率分布也可以关于 n 进行分解。在高斯混合模型的情形中，这个结果意味着混合分布的每个分量对于一个特定的数据点 x_n 的“责任”只与 x_n 的值和混合分量的参数 $\boldsymbol{\theta}$ 有关，而与其他数据点无关。

我们已经看到，EM 算法的 E 步骤和 M 步骤都增大了对数似然函数的一个良好定义的下界的值，并且完整的 EM 循环会使得模型的参数向着使对数似然函数增大的方向进行改变（除非已经达到了一个极大值，此时参数保持不变）。

我们也可以使用 EM 算法来最大化模型的后验概率分布 $p(\boldsymbol{\theta} | \mathbf{X})$ ，其中我们已经引入了参数上的先验概率分布 $p(\boldsymbol{\theta})$ 。为了理解这一点，我们注意到作为一个 $\boldsymbol{\theta}$ 的函数，我们有 $p(\boldsymbol{\theta} | \mathbf{X}) = \frac{p(\boldsymbol{\theta}, \mathbf{X})}{p(\mathbf{X})}$ ，因此

$$\ln p(\boldsymbol{\theta} | \mathbf{X}) = \ln p(\boldsymbol{\theta}, \mathbf{X}) - \ln p(\mathbf{X}) \quad (9.76)$$

使用公式 (9.70) 的分解，我们有

$$\begin{aligned} \ln p(\boldsymbol{\theta} | \mathbf{X}) &= \mathcal{L}(q, \boldsymbol{\theta}) + \text{KL}(q \| p) + \ln p(\boldsymbol{\theta}) - \ln p(\mathbf{X}) \\ &\geq \mathcal{L}(q, \boldsymbol{\theta}) + \ln p(\boldsymbol{\theta}) - \ln p(\mathbf{X}) \end{aligned} \quad (9.77)$$

其中 $\ln p(\mathbf{X})$ 是一个常数。与之前一样，我们可以交替地关于 q 和 $\boldsymbol{\theta}$ 对右侧进行优化。关于 q 的优化产生了与标准 EM 算法相同的 E 步骤，因为 q 只出现在 $\mathcal{L}(q, \boldsymbol{\theta})$ 中。M 步骤的方程通过引入先验项 $\ln p(\boldsymbol{\theta})$ 进行修改，这通常只需要对标准的最大似然 M 步骤的方程进行很小的修改即可。

EM 算法将最大化似然函数这一困难的问题分解成了两个阶段，即 E 步骤和 M 步骤，每个步骤都很容易实现。尽管这样，对于复杂的模型来说，E 步骤或者 M 步骤仍然无法计算。这就引出了对 EM 算法的两个扩展，叙述如下。

推广EM算法 (generalized EM algorithm) , 或者简称GEM算法, 解决的是M步骤无法计算的问题。这个算法不去关于 θ 最大化 $\mathcal{L}(q, \theta)$, 而是改变参数的值去增大 $\mathcal{L}(q, \theta)$ 的值。与之前一样, 由于 $\mathcal{L}(q, \theta)$ 是对数似然函数的一个下界, 因此GEM算法的完整的EM循环保证了对数似然函数值的增大 (除非参数已经对应于一个局部极大值)。一种使用GEM的方法是在M步骤中使用某种非线性最优化策略, 例如共轭梯度算法。另一种形式的GEM算法, 被称为期望条件最大化算法 (expectation conditional maximization algorithm), 或者简称ECM算法, 涉及到在每个M步骤中进行若干具有限制条件的最优化 (Meng and Rubin, 1993)。例如, 参数可能被划分为若干组, 并且M步骤被划分成多个步骤, 每个步骤最优化一个子集, 同时保持其他的子集固定。

类似地, 我们可以用下面的方法推广EM算法中的E步骤: 对 $\mathcal{L}(q, \theta)$ 关于 $q(Z)$ 进行一个部分的最优化而不是完全的最优化 (Neal and Hinton, 1999)。正如我们已经看到的, 对于任意给定的 θ 值, $\mathcal{L}(q, \theta)$ 关于 $q(Z)$ 有一个唯一最大值, 它对应于后验概率分布 $q_\theta(Z) = p(Z | X, \theta)$, 并且对于这个 $q(Z)$ 的选择, 下界 $\mathcal{L}(q, \theta)$ 等于对数似然函数 $\mathcal{L}(q, \theta)$ 。因此任何收敛于 $\mathcal{L}(q, \theta)$ 的全局最大值的算法都会找到一个 θ 值, 这个值也是对数似然函数 $\ln p(X | \theta)$ 的全局最大值。只要 $p(X, Z | \theta)$ 是 θ 的一个连续函数, 那么根据连续性, $\mathcal{L}(q, \theta)$ 的任意一个局部极大值也会是 $\ln p(X | \theta)$ 的一个局部极大值。

考虑 N 个独立数据点 x_1, \dots, x_N 对应于潜在变量 z_1, \dots, z_N 的情形。联合概率分布 $p(X, Z | \theta)$ 可以在数据点上进行分解, 并且这个结构可以被增量形式的EM算法利用, 即在每个EM循环中, 只处理一个数据点。在E步骤中, 我们不重新计算所有数据点的“责任”, 而是只重新计算一个数据点的“责任”。似乎接下来的M步骤会需要涉及到所有数据点的“责任”的计算。但是, 如果混合的分量是指数族分布的成员, 那么“责任”只出现在简单的充分统计量之中, 这些量可以高效地更新。例如, 考虑高斯混合分布的情形, 假设我们对数据点 m 进行了一个更新, 其中对应的“责任”的旧值和新值分别为 $\gamma^{\text{旧}}(z_{mk})$ 和 $\gamma^{\text{新}}(z_{mk})$ 。在M步骤中, 所需的充分统计量可以增量地更新。例如, 对于均值来说, 充分统计量由公式 (9.17) 和公式 (9.18) 定义, 因此我们可以得到

$$\mu_k^{\text{新}} = \mu_k^{\text{旧}} + \left(\frac{\gamma^{\text{新}}(z_{mk}) - \gamma^{\text{旧}}(z_{mk})}{N_k^{\text{新}}} \right) (x_m - \mu_k^{\text{旧}}) \quad (9.78)$$

以及

$$N_k^{\text{新}} = N_k^{\text{旧}} + \gamma^{\text{新}}(z_{mk}) - \gamma^{\text{旧}}(z_{mk}) \quad (9.79)$$

对应的协方差和混合系数的结果与此类似。

因此E步骤和M步骤的计算时间都与数据点的总数无关。由于参数在每个数据点被使用之后进行修改, 而不是等到全部数据处理完毕之后才进行又该, 因此以批处理版本相比, 这个增量版本的收敛速度更快。这个增量算法中的每个E步骤或者M步骤都会增大 $\mathcal{L}(q, \theta)$ 的值, 并且正如我们之前说明的那样, 如果算法收敛于 $\mathcal{L}(q, \theta)$ 的一个局部的 (或者全局的) 最大值, 那么这会对应于对数似然函数 $\ln p(X | \theta)$ 的一个局部的 (或者全局的) 最大值。

9.5 练习

(9.1) (*) 考虑9.1节讨论的 K 均值算法。证明, 由于离散指示器变量 r_{nk} 的集合的可能赋值的数量是有限的, 且对于每种赋值, $\{\mu_k\}$ 有一个唯一的最优值, 因此 K 均值算法已经在有限次迭代之后收敛。

(9.2) (*) 将2.3.5节介绍的Robbins-Monro顺序估计方法应用到寻找回归函数的根的问题中, 回归函数为公式 (9.1) 中的 J 关于 μ_k 的导数。证明, 这会产生出一个随机的 K 均值算法, 其中对于每个数据点 x_n , 最近的代表向量 μ_k 使用公式 (9.5) 进行更新。

(9.3) (*) 考虑一个高斯混合模型, 其中潜在变量的边缘概率分布 $p(z)$ 由公式 (9.10) 给出, 观测变量的条件概率分布 $p(x | z)$ 由公式 (9.11) 给出。证明, 通过在 z 的所有可能值上对 $p(z)p(x | z)$ 进行求和的方式得到的边缘概率分布 $p(x)$ 是一个形式为 (9.7) 的高斯混合分布。

(9.4) (*) 假设我们希望使用EM算法来最大化一个包含潜在变量的模型的参数 $p(\theta | X)$ 上后验概率分布, 其中 X 是观测数据集。证明, E步骤与最大似然的情形相同, 而M步骤中需要最大化的量为 $\mathcal{Q}(\theta, \theta^{\text{旧}}) + \ln p(\theta)$, 其中 $\mathcal{Q}(\theta, \theta^{\text{旧}})$ 由公式 (9.30) 定义。

(9.5) (*) 考虑图9.6给出的高斯混合模型的有向图表示。通过使用8.2节讨论的d-划分准则，证明潜在变量的后验概率分布可以关于不同的数据点分解，即

$$p(\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}) = \prod_{n=1}^N p(z_n \mid x_n, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}) \quad (9.80)$$

(9.6) (**) 考虑高斯混合模型的一个具体的情形，其中各个分量的协方差矩阵 $\boldsymbol{\Sigma}_k$ 全部被限制为一个共同的值 $\boldsymbol{\Sigma}$ 。在这个模型下，推导最大化似然函数的EM方程。

(9.7) (*) 对于高斯混合模型，验证完整数据对数似然函数(9.36)的最大化过程会产生出下面的结果：每个分量的均值和协方差独立地通过对应分组的数据点进行调节，混合系数为每组的数据点的比例。

(9.8) (*) 证明，如果我们关于 $\boldsymbol{\mu}_k$ 最大化(9.40)，同时保持“责任” $\gamma(z_{nk})$ 固定，那么我们可以得到由公式(9.17)给出的解析解。

(9.9) (*) 证明，如果我们关于 $\boldsymbol{\Sigma}_k$ 和 π_k 最大化(9.40)，同时保持“责任” $\gamma(z_{nk})$ 固定，那么我们可以得到由公式(9.19)和(9.22)给出的解析解。

(9.10) (**) 考虑一个由下面的混合概率分布给出的概率密度模型

$$p(\mathbf{x}) = \sum_{k=1}^K \pi_k p(\mathbf{x} \mid k) \quad (9.81)$$

并且假设我们将 \mathbf{x} 划分为两部分，即 $\mathbf{x} = (\mathbf{x}_a, \mathbf{x}_b)$ 。证明条件概率分布 $p(\mathbf{x}_b \mid \mathbf{x}_a)$ 本身是一个混合概率分布。寻找混合系数以及分量概率密度的表达式。

(9.11) (*) 在9.3.2节，我们得到了高斯混合模型的 K 均值方法和EM方法的关系，方法是考虑一个混合模型，它的所有的分量的协方差为 $\epsilon\mathbf{I}$ 。证明，在极限 $\epsilon \rightarrow 0$ 的条件下，由公式(9.40)给出的最大化这个模型的完整数据对数似然函数的期望值等价于最小化公式(9.1)给出的 K 均值算法的失真度量 J 。

(9.12) (*) 考虑一个混合分布，形式为

$$p(\mathbf{x}) = \sum_{k=1}^K \pi_k p(\mathbf{x} \mid k) \quad (9.82)$$

其中 \mathbf{x} 的元素可以是离散的或者连续的或者二者的组合。将 $p(\mathbf{x} \mid k)$ 的均值和方差分别记作 $\boldsymbol{\mu}_k$ 和 $\boldsymbol{\Sigma}_k$ 。证明，混合分布的均值和方差为(9.49)和(9.50)。

(9.13) (**) 使用EM算法的重估计方程，证明，对于一个伯努利混合分布，且它的参数值被设置为了对应于似然函数最大值的那个值，这个伯努利混合分布具有下面的性质

$$\mathbb{E}[\mathbf{x}] = \frac{1}{N} \sum_{n=1}^N \mathbf{x}_n \equiv \bar{\mathbf{x}} \quad (9.83)$$

证明，如果这个模型的参数的初始化使得所有的分量具有同样的均值 $\boldsymbol{\mu}_k = \hat{\boldsymbol{\mu}}$ ，其中 $k = 1, \dots, K$ ，那么对于初始混合系数的任意选择，EM都会在一轮迭代之后收敛，并且这个解满足 $\boldsymbol{\mu}_k = \bar{\mathbf{x}}$ 。注意，这表示混合模型的一个退化形式，其中所有的分量都是相同的。在实际中，我们通过使用合适的初始化来试图避免这样的解。

(9.14) (*) 考虑伯努利分布的潜在变量和观测变量的联合概率分布，它通过计算公式(9.52)给出的 $p(\mathbf{x} \mid \mathbf{z}, \boldsymbol{\mu})$ 和公式(9.53)给出的 $p(\mathbf{z} \mid \boldsymbol{\pi})$ 的乘积的方式获得。证明，如果我们关于 \mathbf{z} 对这个联合概率分布积分或求和，那么我们就得到了公式(9.47)。

(9.15) (*) 证明，如果我们关于 $\boldsymbol{\mu}_k$ 对伯努利混合分布的完整数据对数似然函数的期望(9.55)进行最大化，那么我们会得到M步骤方程(9.59)。

(9.16) (*) 证明，如果我们关于混合系数 π_k 对伯努利混合分布的完整数据对数似然函数的期望(9.55)进行最大化，通过一个拉格朗日乘数来强制满足加和限制，那么我们会得到M步骤方程(9.60)。

(9.17) (*) 证明, 由于离散变量 \mathbf{x}_n 的限制条件 $0 \leq p(\mathbf{x}_n | \boldsymbol{\mu}_k) \leq 1$ 的结果, 伯努利混合分布的不完整数据的对数似然函数具有上界, 因此不存在似然函数趋于无穷大的奇异性。

(9.18) (**) 考虑9.3.3节讨论的伯努利混合模型, 以及由公式(2.13)给出的每个参数向量 $\boldsymbol{\mu}_k$ 上的Beta先验分布 $p(\boldsymbol{\mu}_k | a_k, b_k)$, 以及公式(2.38)给出的狄利克雷先验分布 $p(\boldsymbol{\pi} | \boldsymbol{\alpha})$ 。推导最大化后验概率 $p(\boldsymbol{\mu}, \boldsymbol{\pi} | \mathbf{X})$ 的EM算法。

(9.19) (**) 考虑一个 D 维向量 \mathbf{x} , 它的每个分量 i 本身是一个 M 阶多项式变量, 从而 \mathbf{x} 是一个二值向量, 分量为 x_{ij} , 其中 $i = 1, \dots, D$ 且 $j = 1, \dots, M$, 满足限制条件 $\sum_j x_{ij} = 1$, 对于所有的 i 都成立。假设这些变量的概率分布由2.2节讨论的离散多项式分布混合而成, 即

$$p(\mathbf{x}) = \sum_{k=1}^K \pi_k p(\mathbf{x} | \boldsymbol{\mu}_k) \quad (9.84)$$

其中

$$p(\mathbf{x} | \boldsymbol{\mu}_k) = \sum_{i=1}^D \sum_{j=1}^M \mu_{kij}^{x_{ij}} \quad (9.85)$$

参数 μ_{kij} 表示概率 $p(x_{ij} = 1 | \boldsymbol{\mu}_k)$, 必须满足 $0 \leq \mu_{kij} \leq 1$ 以及限制条件 $\sum_j \mu_{kij} = 1$ 对于所有的 k 和 i 。给定一个观测数据集 $\{\mathbf{x}_n\}$, 其中 $n = 1, \dots, N$, 推导使用最大似然方法对这个概率分布的混合系数 π_k 和分量参数 μ_{kij} 进行最优化的EM算法的E步骤和M步骤方程。

(9.20) (*) 证明贝叶斯线性回归模型的完整数据最大似然函数的期望(9.62)的最大化过程会产生公式(9.63)给出的 $\boldsymbol{\alpha}$ 的M步骤重新估计的结果。

(9.21) (**) 使用3.5节的模型证据框架, 推导贝叶斯线性回归模型的参数 β 的M步骤重新估计方程, 类似于公式(9.63)给出的 $\boldsymbol{\alpha}$ 的结果。

(9.22) (**) 通过最大化公式(9.66)给出的完整数据对数似然函数的期望, 推导用于重新估计回归的相关向量机的超参数的M步骤方程(9.67)和(9.68)。

(9.23) (**) 在7.2.1节, 我们使用对边缘似然函数的直接最大化来推导用于寻找回归RVM的超参数 $\boldsymbol{\alpha}$ 和 β 的值的重新估计方程(7.87)和(7.88)。类似地, 在9.3.4节, 我们使用EM算法来最大化相同的边缘似然函数, 得到了重新估计方程(9.67)和(9.68)。证明, 在任何驻点处, 这两组重估计方程在形式上是等价的。

(9.24) (*) 验证关系(9.70), 其中 $\mathcal{L}(q, \boldsymbol{\theta})$ 和 $\text{KL}(q \parallel p)$ 分别由公式(9.71)和公式(9.72)定义。

(9.25) (*) 证明, 在点 $\boldsymbol{\theta} = \boldsymbol{\theta}^{\text{旧}}$ 处, 公式(9.71)给出的下界 $\mathcal{L}(q, \boldsymbol{\theta})$ 关于 $\boldsymbol{\theta}$ 的梯度与似然 $\ln p(\mathbf{X} | \boldsymbol{\theta})$ 的梯度相同, 其中对于下界来说, $q(\mathbf{Z}) = p(\mathbf{Z} | \mathbf{X}, \boldsymbol{\theta}^{\text{旧}})$ 。

(9.26) (*) 考虑混合高斯分布的EM算法的增量形式, 其中“责任”只对于一个特定的数据点 \mathbf{x}_m 进行重新计算。从M步骤公式(9.17)和(9.18)开始, 推导更新分量均值的结果(9.78)和(9.79)。

(9.27) (**) 在高斯混合模型中, 当“责任”增量地被更新时, 推导更新协方差矩阵和混合系数的M步骤的公式, 类似于更新均值的结果(9.78)。

10 近似推断

在概率模型的应用中，一个中心任务是在给定观测（可见）数据变量 \mathbf{X} 的条件下，计算潜在变量 \mathbf{Z} 的后验概率分布 $p(\mathbf{Z} | \mathbf{X})$ ，以及计算关于这个概率分布的期望。模型可能也包含某些确定性参数，我们现在不考虑它。模型也可能是一个纯粹的贝叶斯模型，其中任何未知的参数都有一个先验概率分布，并且被整合到了潜在变量集合中，记作向量 \mathbf{Z} 。例如，在EM算法中，我们需要计算完整数据对数似然函数关于潜在变量后验概率分布的期望。对于实际应用中的许多模型来说，计算后验概率分布或者计算关于这个后验概率分布的期望是不可行的。这可能是由于潜在空间的维度太高，以至于无法直接计算，或者由于后验概率分布的形式特别复杂，从而期望无法解析地计算。在连续变量的情形中，需要求解的积分可能没有解析解，而空间的维度和被积函数的复杂度可能使得数值积分变得不可行。对于离散变量，求边缘概率的过程涉及到对隐含变量的所有可能的配置进行求和。这个过程虽然原则上总是可以计算的，但是我们在实际应用中经常发现，隐含状态的数量可能有指数多个，从而精确的计算所需的代价过高。

在这种情况下，我们需要借助近似方法。根据近似方法依赖于随机近似还是确定近似，方法大体分为两大类。随机方法，例如第11章介绍的马尔科夫链蒙特卡罗方法，使得贝叶斯方法能够在许多领域中广泛使用。这些方法通常具有这样的性质：给定无限多的计算资源，它们可以生成精确的结果，近似的来源是使用了有限的处理时间。在实际应用中，取样方法需要的计算量会相当大，经常将这些方法的应用限制在了小规模的问题中。并且，判断一种取样方法是否生成了服从所需概率分布的独立样本是很困难的。

本章中，我们介绍了一系列的确定性近似方法，有些方法对于大规模的数据很适用。这些方法基于对后验概率分布的解析近似，例如通过假设后验概率分布可以通过一种特定的方式分解，或者假设后验概率分布有一个具体的参数形式，例如高斯分布。对于这种情况，这些方法永远无法生成精确的解，因此这些方法的优点和缺点与取样方法是互补的。

在4.4节中，我们讨论了拉普拉斯近似，它基于对概率分布的峰值（即，最大值）的局部高斯近似。这里，我们考虑一类近似方法，被称为变分推断（variational inference）或者变分贝叶斯（variational Bayes），它使用了更加全局的准则，并且被广泛应用于实际问题中。我们最后简要介绍另一种变分的框架，被称为期望传播（expectation propagation）。

10.1 变分推断

变分的方法起源于18世纪的欧拉、拉格朗日，以及其他的研究变分法（calculus of variations）的研究。标准的微积分关注的是寻找函数的导数。我们可以将函数想象为一个映射，这个映射以一个变量的值作为输入，返回函数值作为输出。函数的导数描述了当输入变量有一个无限小的变化时，输出值如何变化。类似地，我们可以将泛函（functional）作为一个映射，它以一个函数作为输入，返回泛函的值作为输出。一个例子是熵 $H[p]$ ，它的输入是一个概率分布 $p(x)$ ，返回下面的量

$$H[p] = - \int p(x) \ln p(x) \, dx \quad (10.1)$$

作为输出。我们可以引入泛函的导数（functional derivative）的概念，它表达了输入函数产生无穷小的改变时，泛函的值的变化情况（Feynman et al., 1964）。变分法的规则与标准的微积分规则很相似，在附录D中讨论。许多问题可以表示为最优化问题，其中需要最优化的量是一个泛函。研究所有可能的输入函数，找到最大化或者最小化泛函的函数就是问题的解。变分方法有很广泛的适用性，包括有限元方法（Kapur, 1989）和最大熵方法（Schwarz, 1988）。

虽然变分方法本质上没有任何近似的东西，但是它们通常会被用于寻找近似解。寻找近似解的过程可以这样完成：限制需要最优化算法搜索的函数的范围，例如只考虑二次函数，或者考虑由固定的基函数线性组合而成的函数，其中只有线性组合的系数可以发生变化。在概率推断的应用中，限制条件的形式可以是可分解的假设（Jordan et al., 1999; Jaakkola, 2001）。

现在，让我们详细讨论变分最优化的概念如何应用于推断问题。假设我们有一个纯粹的贝叶斯模型，其中每个参数都有一个先验概率分布。这个模型也可以有潜在变量以及参数，我们会把所有潜在变量和参数组成的集合记作 \mathbf{Z} 。类似地，我们会把所有观测变量的集合记作 \mathbf{X} 。例如，我们可能有 N 个独立同分布的数据，其中 $\mathbf{X} = \{x_1, \dots, x_N\}$ 且 $\mathbf{Z} = \{z_1, \dots, z_N\}$ 。我们的

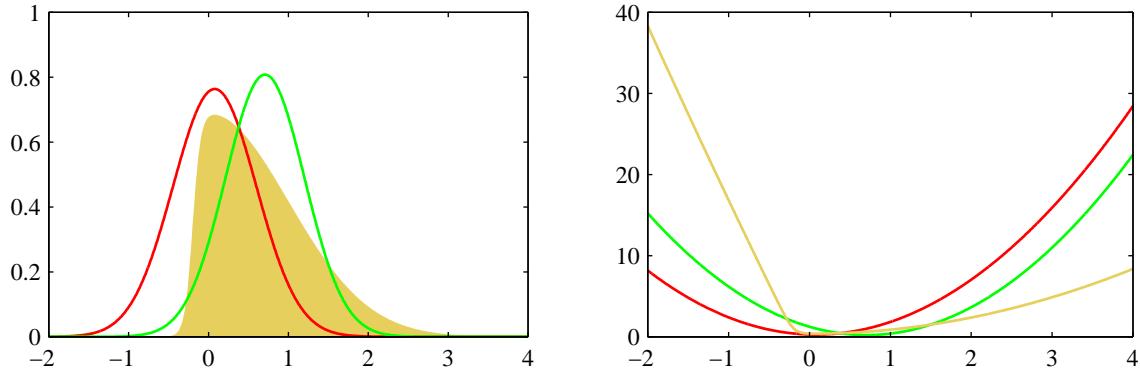


图 10.1: 对于之前在图4.14中考虑过的例子进行变分近似的结果。左图给出了原始的概率分布（黄色）以及拉普拉斯近似（红色）和变分近似（绿色），右图给出了对应曲线的负对数。

概率模型确定了联合概率分布 $p(\mathbf{X}, \mathbf{Z})$ ，我们的目标是找到对后验概率分布 $p(\mathbf{Z} | \mathbf{X})$ 以及模型证据 $p(\mathbf{X})$ 的近似。与我们关于EM的讨论相同，我们可以将对数边缘概率分解，即

$$\ln p(\mathbf{X}) = \mathcal{L}(q) + \text{KL}(q \| p) \quad (10.2)$$

其中我们定义了

$$\mathcal{L}(q) = \int q(\mathbf{Z}) \ln \left\{ \frac{p(\mathbf{X}, \mathbf{Z})}{q(\mathbf{Z})} \right\} d\mathbf{Z} \quad (10.3)$$

$$\text{KL}(q \| p) = - \int q(\mathbf{Z}) \ln \left\{ \frac{p(\mathbf{Z} | \mathbf{X})}{q(\mathbf{Z})} \right\} d\mathbf{Z} \quad (10.4)$$

这与我们关于EM的讨论的唯一的区别是参数向量 θ 不再出现，因为参数现在是随机变量，被整合到了 \mathbf{Z} 中。由于本章中我们主要感兴趣的是连续变量，因此我们在这个分解的公式中使用了积分而不是求和。但是，如果某些变量或者全部的变量都是离散变量，那么分析过程不变，只需根据需要把积分替换为求和即可。与之前一样，我们可以通过关于概率分布 $q(\mathbf{Z})$ 的最优化来使下界 $\mathcal{L}(q)$ 达到最大值，这等价于最小化 KL 散度。如果我们允许任意选择 $q(\mathbf{Z})$ ，那么下界的最大值出现在 KL 散度等于零的时刻，此时 $q(\mathbf{Z})$ 等于后验概率分布 $p(\mathbf{Z} | \mathbf{X})$ 。然而，我们假定在需要处理的模型中，对真实的概率分布进行操作是不可行的。

于是，我们转而考虑概率分布 $q(\mathbf{Z})$ 的一个受限制的类别，然后寻找这个类别中使得 KL 散度达到最小值的概率分布。我们的目标是充分限制 $q(\mathbf{Z})$ 可以取得的概率分布的类别范围，使得这个范围中的所有概率分布都是可以处理的概率分布。同时，我们还要使得这个范围充分大、充分灵活，从而它能够提供对真实后验概率分布的一个足够好的近似。需要强调的是，施加限制条件的唯一目的是为了计算方便，并且在这个限制条件下，我们应该使用尽可能丰富的近似概率分布。特别地，对于高度灵活的概率分布来说，没有“过拟合”现象。使用灵活的近似仅仅使得我们更好地近似真实的后验概率分布。

限制近似概率分布的范围的一种方法是使用参数概率分布 $q(\mathbf{Z} | \omega)$ ，它由参数集合 ω 控制。这样，下界 $\mathcal{L}(q)$ 变成了 ω 的函数，我们可以利用标准的非线性最优化方法确定参数的最优值。图 10.1 给出了这种方法的一个例子，其中变分分布是一个高斯分布，并且我们已经关于均值和协方差进行了最优化。

10.1.1 分解概率分布

这里，我们考虑另一种方法，这种方法里，我们限制概率分布 $q(\mathbf{Z})$ 的范围。假设我们将 \mathbf{Z} 的元素划分成若干个互不相交的组，记作 \mathbf{Z}_i ，其中 $i = 1, \dots, M$ 。然后，我们假定 q 分布关于这些分组可以进行分解，即

$$q(\mathbf{Z}) = \prod_{i=1}^M q_i(\mathbf{Z}_i) \quad (10.5)$$

需要强调的是，我们关于概率分布没有做更多的假设。特别地，我们没有限制各个因子 $q_i(\mathbf{Z}_i)$ 的函数形式。变分推断的这个分解的形式对应于物理学中的一个近似框架，叫做平均场理论（mean field theory）（Parisi, 1988）。

在所有具有公式 (10.5) 的形式的概率分布 $q(\mathbf{Z})$ 中，我们现在寻找下界 $\mathcal{L}(q)$ 最大的概率分布。于是，我们希望对 $\mathcal{L}(q)$ 关于所有的概率分布 $q_i(\mathbf{Z}_i)$ 进行一个自由形式的（变分）最优化。我们通过关于每个因子进行最优化来完成整体的最优化过程。为了完成这一点，我们首先将公式 (10.5) 代入公式 (10.3)，然后分离出依赖于一个因子 $q_j(\mathbf{Z}_j)$ 的项。为了记号的简洁，我们简单地将 $q_j(\mathbf{Z}_j)$ 记作 q_j ，这样我们有

$$\begin{aligned}\mathcal{L}(q) &= \int \prod_i q_i \left\{ \ln p(\mathbf{X}, \mathbf{Z}) - \sum_i \ln q_i \right\} d\mathbf{Z} \\ &= \int q_j \left\{ \int \ln p(\mathbf{X}, \mathbf{Z}) \prod_{i \neq j} q_i d\mathbf{Z}_i \right\} d\mathbf{Z}_j - \int q_j \ln q_j d\mathbf{Z}_j + \text{常数} \\ &= \int q_j \ln \tilde{p}(\mathbf{X}, \mathbf{Z}_j) d\mathbf{Z}_j - \int q_j \ln q_j d\mathbf{Z}_j + \text{常数}\end{aligned}\quad (10.6)$$

其中，我们定义了一个新的概率分布 $\tilde{p}(\mathbf{X}, \mathbf{Z}_j)$ ，形式为

$$\ln \tilde{p}(\mathbf{X}, \mathbf{Z}_j) = \mathbb{E}_{i \neq j} [\ln p(\mathbf{X}, \mathbf{Z})] + \text{常数} \quad (10.7)$$

这里，记号 $\mathbb{E}_{i \neq j}[\cdots]$ 表示关于定义在所有 $\mathbf{z}_i (i \neq j)$ 上的 q 概率分布的期望，即

$$\mathbb{E}_{i \neq j} [\ln p(\mathbf{X}, \mathbf{Z})] = \int \ln p(\mathbf{X}, \mathbf{Z}) \prod_{i \neq j} q_i d\mathbf{Z}_i \quad (10.8)$$

现在假设我们保持 $\{q_{i \neq j}\}$ 固定，关于概率分布 $q_j(\mathbf{Z}_j)$ 的所有可能的形式最大化公式 (10.6) 中的 $\mathcal{L}(q)$ 。这很容易做，因为我们看到公式 (10.6) 是 $q_j(\mathbf{Z}_j)$ 和 $\tilde{p}(\mathbf{X}, \mathbf{Z}_j)$ 之间的Kullback-Leibler散度的负值。因此，最大化公式 (10.6) 等价于最小化Kullback-Leibler散度，且最小值出现在 $q_j(\mathbf{Z}_j) = \tilde{p}(\mathbf{X}, \mathbf{Z}_j)$ 的位置。于是，我们得到了最优解 $q_j^*(\mathbf{Z}_j)$ 的一般的表达式，形式为

$$\ln q_j^*(\mathbf{Z}_j) = \mathbb{E}_{i \neq j} [\ln p(\mathbf{X}, \mathbf{Z})] + \text{常数} \quad (10.9)$$

很值得花一些时间研究一下解的形式，因为它是变分方法应用的基础。这个解表明，为了得到因子 q_j 的最优解的对数，我们只需考虑所有隐含变量和可见变量上的联合概率分布的对数，然后关于所有其他的因子 $\{q_i\}$ 取期望即可，其中 $i \neq j$ 。

公式 (10.9) 中的可加性常数通过对概率分布 $q_j^*(\mathbf{Z}_j)$ 进行归一化的方式来设定。因此，如果我们取两侧的指数，然后归一化，我们有

$$q_j^*(\mathbf{Z}_j) = \frac{\exp(\mathbb{E}_{i \neq j} [\ln p(\mathbf{X}, \mathbf{Z})])}{\int \exp(\mathbb{E}_{i \neq j} [\ln p(\mathbf{X}, \mathbf{Z})]) d\mathbf{Z}_j}$$

在实际应用中，我们会发现，更方便的做法是对公式 (10.9) 进行操作，然后在必要的时候，通过观察的方式恢复出归一化系数。这一点通过下面的例子就会变得逐渐清晰起来。

由公式 (10.9) 给定的方程的集合（其中 $j = 1, \dots, M$ ）表示在概率能够进行分解这一限制条件下，下界的最大值满足的一组相容的条件。然而，这些方程并没有给出一个显式的解，因为最优化 $q_j^*(\mathbf{Z}_j)$ 的公式 (10.9) 的右侧表达式依赖于关于其他的因子 $q_i(\mathbf{Z}_i) (i \neq j)$ 计算的期望。于是，我们会用下面的方式寻找出一个相容的解：首先，恰当地初始化所有的因子 $q_i(\mathbf{Z}_i)$ 然后在各个因子上进行循环，每一轮用一个修正后的估计来替换当前因子。这个修正后的估计由公式 (10.9) 的右侧给出，计算时使用了当前对于所有其他因子的估计。算法保证收敛，因为下界关于每个因子 $q_i(\mathbf{Z}_i)$ 是一个凸函数（Boyd and Vandenberghe, 2004）。

10.1.2 分解近似的性质

我们的变分推断的方法基于的是真实后验概率分布的分解近似。让我们现在考虑一下使用分解概率分布的方式近似一个一般的概率分布的问题。首先，我们讨论使用分解的高斯分布近似一个高斯分布的问题，这会让我们认识到在使用分解近似时会引入的不准确性有哪些类型。考虑两个相关的变量 $\mathbf{z} = (z_1, z_2)$ 上的高斯分布 $p(\mathbf{z}) = \mathcal{N}(\mathbf{z} | \boldsymbol{\mu}, \boldsymbol{\Lambda}^{-1})$ ，其中均值和精度的元素为

$$\boldsymbol{\mu} = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \quad \boldsymbol{\Lambda} = \begin{pmatrix} \Lambda_{11} & \Lambda_{12} \\ \Lambda_{21} & \Lambda_{22} \end{pmatrix} \quad (10.10)$$

并且由于精度矩阵的对称性， $\Lambda_{21} = \Lambda_{12}$ 。现在，假设我们希望使用一个分解的高斯分布 $q(\mathbf{z}) = q_1(z_1)q_2(z_2)$ 来近似这个分布。首先，我们使用一般的结果 (10.9) 来寻找最优因子 $q_1^*(z_1)$ 的表达式。在寻找表达式的过程中，我们注意到，在等式右侧，我们只需要保留哪些与 z_1 有函数依赖关系的项即可，因为所有其他的项都可以被整合到归一化常数中。因此我们有

$$\begin{aligned} \ln q_1^*(z_1) &= \mathbb{E}_{z_2} [\ln p(\mathbf{z})] + \text{常数} \\ &= \mathbb{E}_{z_2} \left[-\frac{1}{2}(z_1 - \mu_1)^2 \Lambda_{11} - (z_1 - \mu_1) \Lambda_{12} (z_2 - \mu_2) \right] + \text{常数} \\ &= -\frac{1}{2} z_1^2 \Lambda_{11} + z_1 \mu_1 \Lambda_{11} - z_1 \Lambda_{12} (\mathbb{E}[z_2] - \mu_2) + \text{常数} \end{aligned} \quad (10.11)$$

接下来，我们观察到这个表达式的右侧是 z_1 的一个二次函数，因此我们可以将 $q^*(z_1)$ 看成一个高斯分布。值得强调的是，我们不假设 $q(z_i)$ 是高斯分布，而是通过对所有可能的分布 $q(z_i)$ 上的KL散度的变分最优化推导出了这个结果。还要注意，我们不需要显式地考虑公式 (10.9) 中的可加性常数，因为它表示归一化常数。如果需要的话，这个常数可以在计算的最后阶段通过观察的方式得到。使用配平方的方法，我们可以得到这个高斯分布的均值和方差，有

$$q_1^*(z_1) = \mathcal{N}(z_1 | m_1, \Lambda_{11}^{-1}) \quad (10.12)$$

其中

$$m_1 = \mu_1 - \Lambda_{11}^{-1} \Lambda_{12} (\mathbb{E}[z_2] - \mu_2) \quad (10.13)$$

根据对称性， $q_2^*(z_2)$ 也是一个高斯分布，可以写成

$$q_2^*(z_2) = \mathcal{N}(z_2 | m_2, \Lambda_{22}^{-1}) \quad (10.14)$$

其中

$$m_2 = \mu_2 - \Lambda_{22}^{-1} \Lambda_{21} (\mathbb{E}[z_1] - \mu_1) \quad (10.15)$$

注意，这些解是相互偶合的，即 $q^*(z_1)$ 依赖于关于 $q^*(z_2)$ 计算的期望，反之亦然。通常，我们这样解决这个问题：将变分解看成重估计方程，然后在变量之间循环，更新这些解，直到满足某个收敛准则。我们稍后会给出一个例子。但是这里，我们注意到这个问题是相当简单的，因为可以找到一个解析解。特别地，由于 $\mathbb{E}[z_1] = m_1$ 且 $\mathbb{E}[z_2] = m_2$ ，因此我们看到，如果我们取 $\mathbb{E}[z_1] = \mu_1$ 且 $\mathbb{E}[z_2] = \mu_2$ ，那么这两个方程会得到满足。并且很容易证明，只要概率分布非奇异，那么这个解是唯一解。这个结果如图10.2(a)所示。我们看到，均值被正确地描述了，但是 $q(\mathbf{z})$ 的方差由 $p(\mathbf{z})$ 的最小方差的方向所确定，沿着垂直方向的方差被强烈地低估了。这是一个一般的结果，即分解变分近似对后验概率分布的近似倾向于过于紧凑。

作为比较，假设我们最小化相反的Kullback-Leibler散度 $\text{KL}(p \| q)$ 。正如我们将看到的那样，这种形式的KL散度被用于另一种近似推断的框架中，这种框架被称为期望传播 (expectation propagation)。于是，我们考虑一般的最小化 $\text{KL}(p \| q)$ 的问题，其中 $q(\mathbf{Z})$ 是形式为 (10.5) 的分解近似。这样，KL散度可以写成

$$\text{KL}(p \| q) = - \int p(\mathbf{Z}) \left[\sum_{i=1}^M \ln q_i(\mathbf{Z}_i) \right] d\mathbf{Z} + \text{常数} \quad (10.16)$$

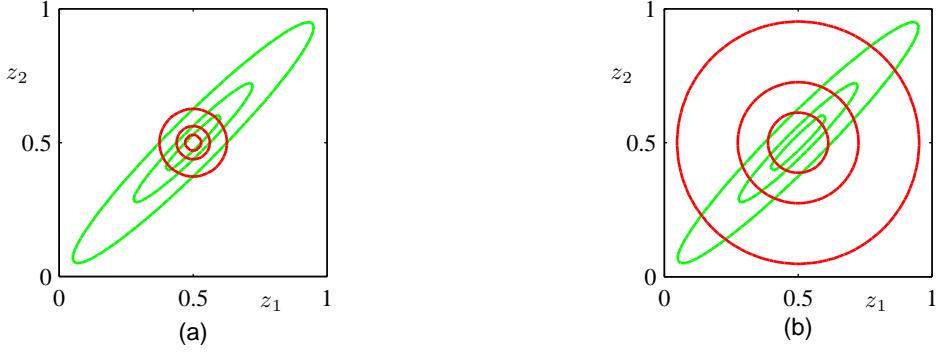


图 10.2: 两种形式的KL散度的对比。绿色轮廓线对应于两个变量 z_1 和 z_2 上的相关高斯分布 $p(z)$ 的1、2、3个标准差的位置，红色轮廓线表示相同变量上的近似分布 $q(z)$ 的同样位置。近似分布 $q(z)$ 由两个独立的一元高斯分布的乘积给出，(a)图中，参数通过最小化Kullback-Leibler散度 $\text{KL}(q \parallel p)$ 的方式获得，(b)图中，参数通过最小化相反的Kullback-Leibler散度 $\text{KL}(p \parallel q)$ 的方式获得。

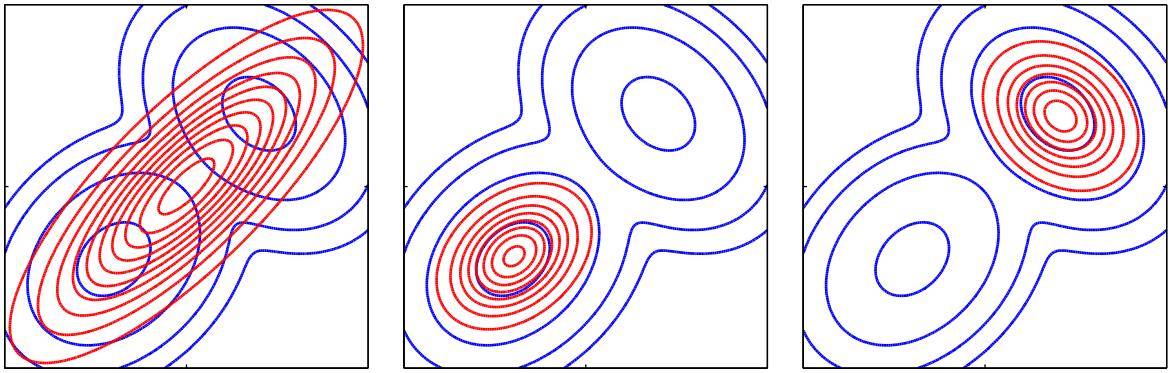


图 10.3: 两种形式的Kullback-Leibler散度的另一个对比。(a)蓝色轮廓线展示了由两个高斯分布混合而成的双峰概率分布 $p(\mathbf{Z})$ ，红色轮廓线对应于一个高斯分布 $q(\mathbf{Z})$ ，它最小化了Kullback-Leibler散度 $\text{KL}(p \parallel q)$ ，在这种意义上最好地近似了 $p(\mathbf{Z})$ 。(b)与(a)相同，但是此时红色轮廓线对应的高斯分布 $q(\mathbf{Z})$ 是通过使用数值方法最小化Kullback-Leibler散度 $\text{KL}(q \parallel p)$ 的方式得到的。(c)与(b)相同，但是给出了Kullback-Leibler散度的另一个局部最小值。

其中，常数项就是 $p(\mathbf{Z})$ 的熵，因此不依赖于 $q(\mathbf{Z})$ 。我们现在可以关于每个因子 $q_j(\mathbf{Z}_j)$ 进行最优化。使用拉格朗日乘数法，很容易得到结果

$$q_j^*(\mathbf{Z}_j) = \int p(\mathbf{Z}) \prod_{i \neq j} d\mathbf{Z}_i = p(\mathbf{Z}_j) \quad (10.17)$$

在这种情况下，我们看到 $q_j(\mathbf{Z}_j)$ 的最优解等于对应的边缘概率分布 $p(\mathbf{Z})$ 。注意，这是一个解析解，不需要迭代。

为了将这个结果应用到向量 \mathbf{z} 上的高斯分布 $p(\mathbf{z})$ 这个例子上，我们可以使用公式 (2.98)，它给出了图10.2(b)的结果。我们再一次看到，对均值的近似是正确的，但是它把相当多的概率质量放到了实际上具有很低的概率的变量空间区域中。

这两个结果的区别可以用下面的方式理解。我们注意到， \mathbf{Z} 空间中 $p(\mathbf{Z})$ 接近等于零的区域对于Kullback-Leibler散度

$$\text{KL}(q \parallel p) = - \int q(\mathbf{Z}) \ln \left\{ \frac{p(\mathbf{Z})}{q(\mathbf{Z})} \right\} d\mathbf{Z} \quad (10.18)$$

有一个大的正数的贡献，除非 $q(\mathbf{Z})$ 也接近等于零。因此最小化这种形式的KL散度会使得概率分布 $q(\mathbf{Z})$ 避开 $p(\mathbf{Z})$ 很小的区域。相反地，使得Kullback-Leibler散度 $\text{KL}(p \parallel q)$ 的散度取得最小值的概率分布 $q(\mathbf{Z})$ 在 $p(\mathbf{Z})$ 非零的区域中也是非零的。

如果我们考虑用一个单峰分布近似多峰分布的问题，我们会更深刻地认识两个KL散度的不同行为，如图10.3所示。在实际应用中，真实的后验概率分布经常是多峰的，大部分后验概率质

量集中在参数空间中的某几个相对较小的区域中。这些多个峰值可能是由于潜在空间的不可区分性所造成的，也可能是由于对参数的复杂的非线性依赖关系造成的。我们在第9章中讨论高斯混合模型的时候遇到过这两种类型的多峰性质，那里，这些峰值以似然函数的多个极大值的形式显现出来。基于最小化 $\text{KL}(q \parallel p)$ 的变分方法倾向于找到这些峰值中的一个。相反，如果我们最小化 $\text{KL}(p \parallel q)$ ，那么得到的近似会在所有的均值上取平均。在混合模型问题中，这种方法会给出较差的预测分布（因为两个较好的参数值的平均值通常不是一个较好的参数值）。可以使用 $\text{KL}(p \parallel q)$ 定义一个有用的推断步骤，但是这需要一种与这里讨论的内容相当不同的方法。当我们讨论期望传播的时候，我们会仔细讨论这一点。

两种形式的Kullback-Leibler散度都是散度的alpha家族（alpha family）的成员（Ali and Silvey, 1966; Amari, 1985; Minka, 2005），定义为

$$D_\alpha(p \parallel q) = \frac{4}{1 - \alpha^2} \left(1 - \int p(x)^{\frac{1+\alpha}{2}} q(x)^{\frac{1-\alpha}{2}} dx \right) \quad (10.19)$$

其中 $-\infty < \alpha < \infty$ 是一个连续参数。Kullback-Leibler散度 $\text{KL}(p \parallel q)$ 对应于极限 $\alpha \rightarrow 1$ ，而 $\text{KL}(q \parallel p)$ 对应于极限 $\alpha \rightarrow -1$ 。对于所有 α 的值，我们有 $D_\alpha(p \parallel q) \geq 0$ ，当且仅当 $p(x) = q(x)$ 时等号成立。假设 $p(x)$ 是一个固定的分布，我们关于某个概率分布 $q(x)$ 的集合最小化 $D_\alpha(p \parallel q)$ 。那么对于 $\alpha \leq -1$ 的情况，散度是零强制的（zero forcing），即，对于使得 $p(x) = 0$ 成立的任意 x 值，都有 $q(x) = 0$ ，通常 $q(x)$ 会低估 $p(x)$ 的支持，因此倾向于寻找具有最大质量的峰值。相反，对于 $\alpha \geq 1$ 的情况，散度是零避免的（zero avoiding），即，对于使得 $p(x) > 0$ 成立的任意 x 值，都有 $q(x) > 0$ ，通常 $q(x)$ 会进行拉伸来覆盖到所有的 $p(x)$ 值，从而高估了 $p(x)$ 的支持。当 $\alpha = 0$ 时，我们得到了一个对称的散度，它与 Hellinger 距离线性相关，定义为

$$D_H(p \parallel q) = \int \left(p(x)^{\frac{1}{2}} - q(x)^{\frac{1}{2}} \right)^2 dx \quad (10.20)$$

Hellinger 距离的平方根是一个合法的距离度量。

10.1.3 例子：一元高斯分布

我们现在使用一元变量 x 上的高斯分布来说明分解变分近似（MacKay, 2003）。我们的目标是在给定 x 的观测值的数据集 $\mathcal{D} = \{x_1, \dots, x_N\}$ 的情况下，推断均值 μ 和精度 τ 的后验概率分布。其中，我们假设数据是独立地从高斯分布中抽取的。似然函数为

$$p(\mathcal{D} \mid \mu, \tau) = \left(\frac{\tau}{2\pi} \right)^{\frac{N}{2}} \exp \left\{ -\frac{\tau}{2} \sum_{n=1}^N (x_n - \mu)^2 \right\} \quad (10.21)$$

我们现在引入 μ 和 τ 的共轭先验分布，形式为

$$p(\mu \mid \tau) = \mathcal{N}(\mu \mid \mu_0, (\lambda_0 \tau)^{-1}) \quad (10.22)$$

$$p(\tau) = \text{Gam}(\tau \mid a_0, b_0) \quad (10.23)$$

其中 $\text{Gam}(\tau \mid a_0, b_0)$ 是公式 (2.146) 定义的 Gamma 分布。这些分布共同给出了一个高斯-Gamma 共轭先验分布。

对于这个简单的问题，后验概率可以求出精确解，并且形式还是高斯-Gamma 分布。然而，为了讲解的目的，我们会考虑对后验概率分布的一个分解变分近似，形式为

$$q(\mu, \tau) = q_\mu(\mu)q_\tau(\tau) \quad (10.24)$$

注意，真实的后验概率分布不可以按照这种形式进行分解。最优的因子 $q_\mu(\mu)$ 和 $q_\tau(\tau)$ 可以从一般的结果 (10.9) 中得到，如下所述。对于 $q_\mu(\mu)$ ，我们有

$$\begin{aligned} \ln q_\mu^*(\mu) &= \mathbb{E}_\tau [\ln p(\mathcal{D} \mid \mu, \tau) + \ln p(\mu \mid \tau)] + \text{常数} \\ &= -\frac{\mathbb{E}[\tau]}{2} \left\{ \lambda_0(\mu - \mu_0)^2 + \sum_{n=1}^N (x_n - \mu)^2 \right\} + \text{常数} \end{aligned} \quad (10.25)$$

对于 μ 配平方，我们看到 $q_\mu(\mu)$ 是一个高斯分布 $\mathcal{N}(\mu | \mu_N, \lambda_N^{-1})$ ，其中，均值和方差为

$$\mu_N = \frac{\lambda_0 \mu_0 + N \bar{x}}{\lambda_0 + N} \quad (10.26)$$

$$\lambda_N = (\lambda_0 + N) \mathbb{E}[\tau] \quad (10.27)$$

注意，对于 $N \rightarrow \infty$ ，这给出了最大似然的结果，其中 $\mu_N = \bar{x}$ ，精度为无穷大。

类似地，因子 $q_\tau(\tau)$ 的最优解为

$$\begin{aligned} \ln q_\tau^*(\tau) &= \mathbb{E}_\mu [\ln p(\mathcal{D} | \mu, \tau) + \ln p(\mu | \tau)] + \ln p(\tau) + \text{常数} \\ &= (a_0 - 1) \ln \tau - b_0 \tau + \frac{N+1}{2} \ln \tau \\ &\quad - \frac{\tau}{2} \mathbb{E}_\mu \left[\sum_{n=1}^N (x_n - \mu)^2 + \lambda_0 (\mu - \mu_0)^2 \right] + \text{常数} \end{aligned} \quad (10.28)$$

因此 $q_\tau(\tau)$ 是一个Gamma分布 $\text{Gam}(\tau | a_N, b_N)$ ，参数为

$$a_N = a_0 + \frac{N+1}{2} \quad (10.29)$$

$$b_N = b_0 + \frac{1}{2} \mathbb{E}_\mu \left[\sum_{n=1}^N (x_n - \mu)^2 + \lambda_0 (\mu - \mu_0)^2 \right] \quad (10.30)$$

与之前一样，当 $N \rightarrow \infty$ 时，它的行为与预期相符。

应该强调的是，我们不假设最优概率分布 $q_\mu(\mu)$ 和 $q_\tau(\tau)$ 的具体的函数形式。它们的函数形式从似然函数和对应的共轭先验分布中自然地得到。

因此，我们得到了最优概率分布 $q_\mu(\mu)$ 和 $q_\tau(\tau)$ 的表达式，每个表达式依赖于关于其他概率分布计算得到的矩。因此，一种寻找解的方法是对例如 $\mathbb{E}[\tau]$ 进行一个初始的猜测，然后使用这个猜测来重新计算概率分布 $q_\mu(\mu)$ 。给定这个修正的概率分布之后，我们接下来可以计算所需的矩 $\mathbb{E}[\mu]$ 和 $\mathbb{E}[\mu^2]$ ，并且使用这些矩来重新计算概率分布 $q_\tau(\tau)$ ，以此类推。由于这个例子中，隐含变量空间是二维的，因此我们可以用图形来说明后验概率分布的变分近似过程。我们画出了真后验概率的轮廓线和分解近似的轮廓线，如图10.4所示。

通常，我们需要使用一种迭代的方法来得到最优分解后验概率分布的解。然而，对于我们这里讨论的非常简单的例子来说，我们可以通过求解最优因子 $q_\mu(\mu)$ 和 $q_\tau(\tau)$ 的方程，得到一个显式的解。在做这件事之前，我们可以通过考虑无信息先验来简化表达式。无信息先验分布中， $\mu_0 = a_0 = b_0 = \lambda_0 = 0$ 。虽然这些参数设置对应于一个反常先验，但是我们看到后验概率分布仍然具有良好的定义。使用Gamma分布的均值的标准结果 $\mathbb{E}[\tau] = \frac{a_N}{b_N}$ ，以及公式 (10.29) 和公式 (10.30)，我们有

$$\frac{1}{\mathbb{E}[\tau]} = \mathbb{E} \left[\frac{1}{N+1} \sum_{n=1}^N (x_n - \mu)^2 \right] = \frac{N}{N+1} (\bar{x}^2 - 2\bar{x}\mathbb{E}[\mu] + \mathbb{E}[\mu^2]) \quad (10.31)$$

之后，使用公式 (10.26) 和公式 (10.27)，我们得到了 $q_\mu(\mu)$ 的一阶矩和二阶矩，形式为

$$\mathbb{E}[\mu] = \bar{x}, \quad \mathbb{E}[\mu^2] = \bar{x}^2 + \frac{1}{N\mathbb{E}[\tau]} \quad (10.32)$$

现在，我们可以将这些矩代入公式 (10.31)，然后解出 $\mathbb{E}[\tau]$ ，可得

$$\frac{1}{\mathbb{E}[\tau]} = (\bar{x}^2 - \bar{x}^2) = \frac{1}{N} \sum_{n=1}^N (x_n - \bar{x})^2 \quad (10.33)$$

对于高斯分布的贝叶斯推断的可理解的介绍，包括与最大似然方法的相比的优势的讨论，可以参考Minka (1998)。

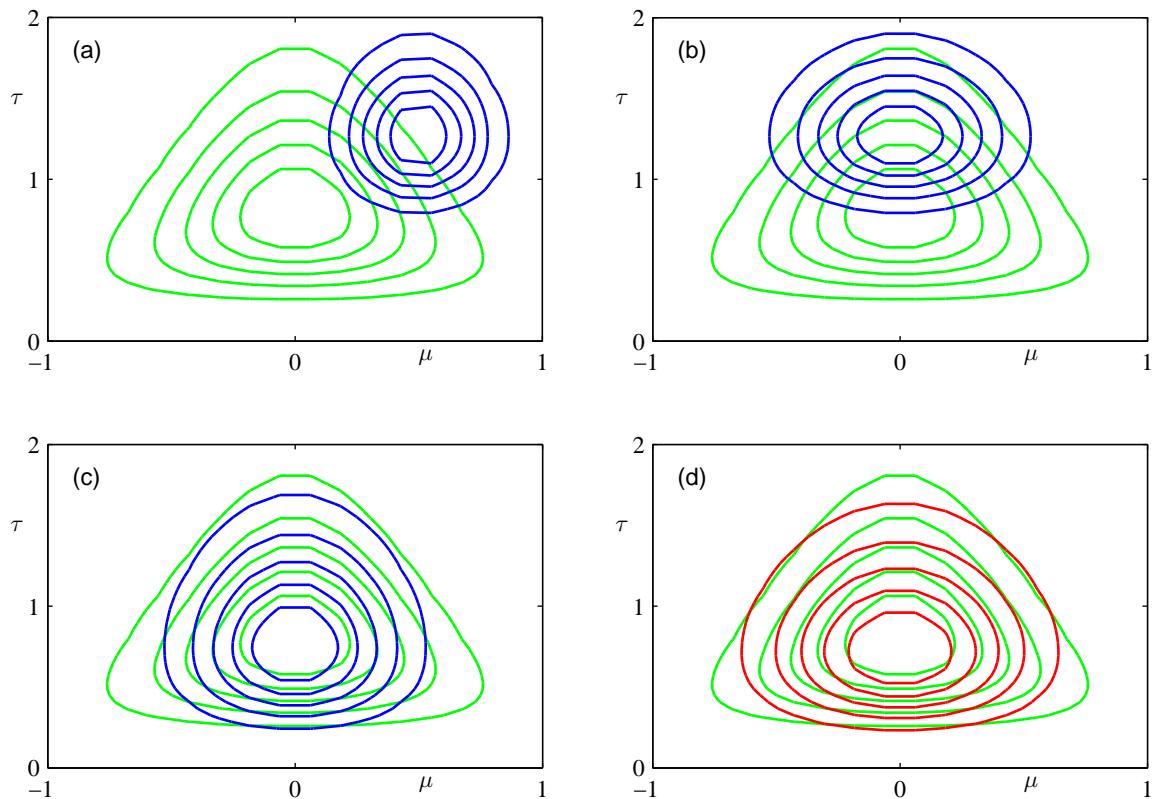


图 10.4: 一元高斯分布的均值 μ 和精度 τ 的变分推断的例子。真实后验概率分布 $p(\mu, \tau | \mathcal{D})$ 用绿色曲线表示。(a)初始的分解近似 $q_\mu(\mu)q_\tau(\tau)$, 用蓝色曲线表示。(b)重新估计了因子 $q_\mu(\mu)$ 之后的结果。(c)重新估计了因子 $q_\tau(\tau)$ 之后的结果。(d)最优分解近似的轮廓线, 其中迭代方法收敛, 用红色表示。

10.1.4 模型比较

除了在隐含变量 Z 上进行推断之外，我们可能还希望对比一组候选模型。索引为 m 的模型的先验概率分布为 $p(m)$ 。这样，我们的目标是近似后验概率分布 $p(m | \mathbf{X})$ ，其中 \mathbf{X} 是观测数据。这比我们目前为止考虑的情况稍微复杂一些，因为不同的模型可能具有不同的结构，并且隐含变量 Z 的维度实际上可能不同。因此我们不能简单地考虑分解近似 $q(Z)q(m)$ ，而是必须意识到 Z 的后验概率分布必须以 m 为条件，所以我们必须考虑 $q(Z, m) = q(Z | m)q(m)$ 。我们已经可以验证下面的基于变分概率分布的分解方式

$$\ln p(\mathbf{X}) = \mathcal{L} - \sum_m \sum_Z q(Z | m)q(m) \ln \left\{ \frac{p(Z, m | \mathbf{X})}{q(Z | m)q(m)} \right\} \quad (10.34)$$

其中 \mathcal{L} 是 $\ln p(\mathbf{X})$ 的下界，形式为

$$\mathcal{L} = \sum_m \sum_Z q(Z | m)q(m) \ln \left\{ \frac{p(Z, \mathbf{X}, m)}{q(Z | m)q(m)} \right\} \quad (10.35)$$

这里，我们假定 Z 是离散变量，但是同样的分析也适用于连续潜在变量，只要我们把求和替换为积分即可。我们可以使用拉格朗日乘数法关于概率分布 $q(m)$ 最大化 \mathcal{L} ，结果为

$$q(m) \propto p(m) \exp\{\mathcal{L}_m\} \quad (10.36)$$

其中

$$\mathcal{L}_m = \sum_Z q(Z | m) \ln \left\{ \frac{p(Z, \mathbf{X} | m)}{q(Z | m)} \right\}$$

然而，如果我们关于 $q(Z | m)$ 最大化 \mathcal{L} ，那么我们发现对于不同的 m 值，解是相互偶合的，这与我们预期相符，因为这些概率分布是以 m 为条件的。我们接下来首先通过最优化 (10.35)，或者等价地，最优化 \mathcal{L}_m ，来独立地最优化每个 $q(Z | m)$ ，然后使用公式 (10.36) 来确定 $q(m)$ 。在对求得的 $q(m)$ 值进行归一化之后，它的值可以用于模型选择或者模型平均。

10.2 例子：高斯的变分混合

我们现在回到我们对于高斯混合模型的讨论，并且使用前一节讨论的变分推断的方法。这会很好地说明变分方法的应用，也会展示出贝叶斯方法是如何优雅地解决最大似然方法中的许多困难之处的 (Attias, 1999b)。我们建议读者仔细研究这个例子，因为这个例子给出了变分方法在实际应用中的许多重要的思想。许多贝叶斯模型，对应于复杂得多的概率分布，可以通过对本节中的分析进行简单的扩展进行求解。

我们的起始点是高斯混合模型的似然函数。高斯混合模型如图9.6给出的图模型所示。对于每个观测 x_n ，我们有一个对应的潜在变量 z_n ，它是一个“1-of- K ”的二值向量，元素为 z_{nk} ，其中 $k = 1, \dots, K$ 。与之前一样，我们将观测数据集记作 $\mathbf{X} = \{x_1, \dots, x_N\}$ ，类似地，我们将潜在变量记作 $Z = \{z_1, \dots, z_N\}$ 。给定混合系数 π ，根据公式 (9.10)，我们可以写出 Z 的条件概率分布，形式为

$$p(Z | \pi) = \prod_{n=1}^N \prod_{k=1}^K \pi_k^{z_{nk}} \quad (10.37)$$

类似地，给定潜在变量和分量参数，根据公式 (9.11)，我们可以写出观测数据向量的条件概率分布，形式为

$$p(\mathbf{X} | Z, \boldsymbol{\mu}, \boldsymbol{\Lambda}) = \prod_{n=1}^N \prod_{k=1}^K \mathcal{N}(x_n | \boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k^{-1})^{z_{nk}} \quad (10.38)$$

其中 $\boldsymbol{\mu} = \{\boldsymbol{\mu}_k\}$ 且 $\boldsymbol{\Lambda} = \{\boldsymbol{\Lambda}_k\}$ 。注意，我们计算时使用的时精度矩阵而不是协方差矩阵，因为这在一定程度上简化了数学计算的复杂度。

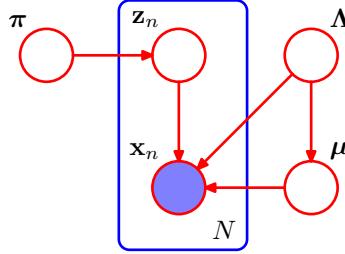


图 10.5: 表示高斯模型的贝叶斯混合的有向图，其中，方框表示一组 N 个独立同分布的观测。这里 μ 表示 $\{\mu_k\}$ ， Λ 表示 $\{\Lambda_k\}$ 。

接下来，我们引入参数 μ, Λ 和 π 上的先验概率分布。如果我们使用共轭先验分布，那么分析过程会得到极大的简化。于是，我们选择混合系数 π 上的狄利克雷分布。

$$p(\pi) = \text{Dir}(\pi | \alpha_0) = C(\alpha_0) \prod_{k=1}^K \pi_k^{\alpha_0-1} \quad (10.39)$$

其中，根据对称性，我们为每个分量选择了同样的参数 α_0 ， $C(\alpha_0)$ 是狄利克雷分布的归一化常数，由公式 (B.23) 定义。正如我们已经看到的那样，参数 α_0 可以看成与混合分布的每个分量关联的观测的有效先验数量。如果 α_0 的值很小，那么后验概率分布会主要被数据集影响，而受到先验概率的影响很小。

类似地，我们引入一个独立的高斯-Wishart 先验分布，控制每个高斯分布的均值和精度，形式为

$$\begin{aligned} p(\mu, \Lambda) &= p(\mu | \Lambda)p(\Lambda) \\ &= \prod_{k=1}^K \mathcal{N}(\mu_k | m_0, (\beta_0 \Lambda_k)^{-1}) \mathcal{W}(\Lambda_k | W_0, \nu_0) \end{aligned} \quad (10.40)$$

这是由于当均值和精度均未知的时候，它表示共轭先验分布。通常根据对称性，我们选择 $m_0 = \mathbf{0}$ 。

生成的模型可以表示为图 10.5 所示的有向图。注意，从 Λ 到 μ 之间存在一个链接，这是由于公式 (10.40) 中的 μ 上的概率分布的方差为 Λ 的函数。

这个例子很好地说明了潜在变量和参数之间的区别。像 z_n 这样出现在方框内部的变量被看做隐含变量，因为这种变量的数量随着数据集规模的增大而增大。相反，像 μ 这样出现在方框外的变量的数量与数据集的规模无关，因此被当做参数。然而，从图模型的观点来看，它们之间没有本质的区别。

10.2.1 变分分布

为了形式化地描述这个模型的变分方法，我们接下来写出所有随机变量的联合概率分布，形式为

$$p(\mathbf{X}, \mathbf{Z}, \pi, \mu, \Lambda) = p(\mathbf{X} | \mathbf{Z}, \mu, \Lambda)p(\mathbf{Z} | \pi)p(\pi)p(\mu | \Lambda)p(\Lambda) \quad (10.41)$$

其中，各种因子已经在之前定义过。读者现在应该验证一下这种分解方式确实对应于图 10.5 给出的概率图模型。注意，只有变量 $\mathbf{X} = \{x_1, \dots, x_N\}$ 是观测变量。

我们现在考虑一个变分分布，它可以在潜在变量与参数之间进行分解，即

$$q(\mathbf{Z}, \pi, \mu, \Lambda) = q(\mathbf{Z})q(\pi, \mu, \Lambda) \quad (10.42)$$

需要注意的是，为了让我们的贝叶斯混合模型能够有一个合理的可以计算的解，这是我们做出的唯一的假设。特别地，因子 $q(\mathbf{Z})$ 和 $q(\pi, \mu, \Lambda)$ 的函数形式会在变分分布的最优化过程中自动确定。注意，我们省略了 q 分布的下标，就像我们在公式 (10.41) 中做的那样。我们依赖参数来区分不同的分布。

通过使用一般的结果 (10.9)，这些因子的对应的顺序更新方程可以很容易地推导出来。让我们考虑因子 $q(\mathbf{Z})$ 的更新方程的推导。最优因子的对数为

$$\ln q^*(\mathbf{Z}) = \mathbb{E}_{\pi, \mu, \Lambda} [\ln p(\mathbf{X}, \mathbf{Z}, \pi, \mu, \Lambda)] + \text{常数} \quad (10.43)$$

我们现在使用公式 (10.41) 给出的分解方式。注意，我们只对等式右侧与变量 \mathbf{Z} 相关的函数关系感兴趣。因此，任何与变量 \mathbf{Z} 无关的项都可以被整合到可加的归一化系数中，从而有

$$\ln q^*(\mathbf{Z}) = \mathbb{E}_\pi [\ln p(\mathbf{Z} | \pi)] + \mathbb{E}_{\mu, \Lambda} [\ln p(\mathbf{X} | \mathbf{Z}, \mu, \Lambda)] + \text{常数} \quad (10.44)$$

替换右侧的两个条件分布，然后再次把与 \mathbf{Z} 无关的项整合到可加性常数中，我们有

$$\ln q^*(\mathbf{Z}) = \sum_{n=1}^N \sum_{k=1}^K z_{nk} \ln \rho_{nk} + \text{常数} \quad (10.45)$$

其中我们定义了

$$\begin{aligned} \ln \rho_{nk} &= \mathbb{E}[\ln \pi_k] + \frac{1}{2} \mathbb{E}[\ln |\Lambda_k|] - \frac{D}{2} \ln(2\pi) \\ &\quad - \frac{1}{2} \mathbb{E}_{\mu_k, \Lambda_k} [(\mathbf{x}_n - \boldsymbol{\mu}_k)^T \boldsymbol{\Lambda}_k (\mathbf{x}_n - \boldsymbol{\mu}_k)] \end{aligned} \quad (10.46)$$

其中 D 是数据变量 \mathbf{x} 的维度。公式 (10.45) 两侧取指数，我们有

$$q^*(\mathbf{Z}) \propto \prod_{n=1}^N \prod_{k=1}^K \rho_{nk}^{z_{nk}} \quad (10.47)$$

我们要求这个概率分布是归一化的，并且我们注意到对于每个 n 值， z_{nk} 都是二值的，在所有的 k 值上的加和等于1，因此我们有

$$q^*(\mathbf{Z}) = \prod_{n=1}^N \prod_{k=1}^K r_{nk}^{z_{nk}} \quad (10.48)$$

其中

$$r_{nk} = \frac{\rho_{nk}}{\sum_{j=1}^K \rho_{nj}} \quad (10.49)$$

我们看到，因子 $q(\mathbf{Z})$ 的最优解的函数形式与先验概率分布 $p(\mathbf{Z} | \pi)$ 的函数形式相同。注意，由于 ρ_{nk} 是一个实数值的指数，因此 r_{nk} 是非负的，且加和等于1，满足要求。

对于离散概率分布 $q^*(\mathbf{Z})$ ，我们有标准的结果

$$\mathbb{E}[z_{nk}] = r_{nk} \quad (10.50)$$

从中我们看到 r_{nk} 扮演着“责任”的角色。注意， $q^*(\mathbf{Z})$ 的最优解依赖于关于其他变量计算得到的矩，因此与之前一样，变分更新方程是偶合的，必须用迭代的方式求解。

现在，我们会发现定义观测数据关于“责任”的下面三个统计量会比较方便，即

$$N_k = \sum_{n=1}^N r_{nk} \quad (10.51)$$

$$\bar{\mathbf{x}}_k = \frac{1}{N_k} \sum_{n=1}^N r_{nk} \mathbf{x}_n \quad (10.52)$$

$$\mathbf{S}_k = \frac{1}{N_k} \sum_{n=1}^N r_{nk} (\mathbf{x}_n - \bar{\mathbf{x}}_k)(\mathbf{x}_n - \bar{\mathbf{x}}_k)^T \quad (10.53)$$

注意，这些类似于高斯混合模型的最大似然EM算法中计算的量。

现在让我们考虑变分后验概率分布中的因子 $q(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda})$ 。与之前一样，使用公式 (10.9) 给出的一般的结果，我们有

$$\begin{aligned} \ln q^*(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}) &= \ln p(\boldsymbol{\pi}) + \sum_{k=1}^K \ln p(\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k) + \mathbb{E}_{\mathbf{Z}}[\ln p(\mathbf{Z} | \boldsymbol{\pi})] \\ &\quad + \sum_{k=1}^K \sum_{n=1}^N \mathbb{E}[z_{nk}] \ln \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k^{-1}) + \text{常数} \end{aligned} \quad (10.54)$$

我们观察到，这个表达式的右侧分解成了若干项的和，一些项只与 $\boldsymbol{\pi}$ 相关，一些项只与 $\boldsymbol{\mu}$ 和 $\boldsymbol{\Lambda}$ 相关，这表明变分后验概率 $q(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda})$ 可以分解为 $q(\boldsymbol{\pi})q(\boldsymbol{\mu}, \boldsymbol{\Lambda})$ 。此外，与 $\boldsymbol{\mu}$ 和 $\boldsymbol{\Lambda}$ 相关的项本身由 k 个与 $\boldsymbol{\mu}_k$ 和 $\boldsymbol{\Lambda}_k$ 相关的项有关，因此可以进一步分解，即

$$q(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}) = q(\boldsymbol{\pi}) \prod_{k=1}^K q(\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k) \quad (10.55)$$

分离出公式 (10.54) 右侧的与 $\boldsymbol{\pi}$ 相关的项，我们有

$$\ln q^*(\boldsymbol{\pi}) = (\alpha_0 - 1) \sum_{k=1}^K \ln \pi_k + \sum_{k=1}^K \sum_{n=1}^N r_{nk} \ln \pi_k + \text{常数} \quad (10.56)$$

其中我们使用了公式 (10.50)。两侧取指数，我们将 $q^*(\boldsymbol{\pi})$ 看成狄利克雷分布

$$q^*(\boldsymbol{\pi}) = \text{Dir}(\boldsymbol{\pi} | \boldsymbol{\alpha}) \quad (10.57)$$

其中 $\boldsymbol{\alpha}$ 的元素为 α_k ，形式为

$$\alpha_k = \alpha_0 + N_k \quad (10.58)$$

最后，变分后验概率分布 $q^*(\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k)$ 无法分解成边缘概率分布的乘积，但是我们总可以使概率的乘积规则，将其写成 $q^*(\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k) = q^*(\boldsymbol{\mu}_k | \boldsymbol{\Lambda}_k)q^*(\boldsymbol{\Lambda}_k)$ 。两个因子可以通过观察公式 (10.54) 得到，并且可以读出 $\boldsymbol{\mu}_k$ 和 $\boldsymbol{\Lambda}_k$ 。与预期相符，结果是一个高斯-Wishart分布，形式为

$$q^*(\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k) = \mathcal{N}(\boldsymbol{\mu}_k | \mathbf{m}_k, (\beta_k \boldsymbol{\Lambda}_k)^{-1}) \mathcal{W}(\boldsymbol{\Lambda}_k | \mathbf{W}_k, \nu_k) \quad (10.59)$$

其中我们已经定义了

$$\beta_k = \beta_0 + N_k \quad (10.60)$$

$$\mathbf{m}_k = \frac{1}{\beta_k} (\beta_0 \mathbf{m}_0 + N_k \bar{\mathbf{x}}_k) \quad (10.61)$$

$$\mathbf{W}_k^{-1} = \mathbf{W}_0^{-1} + N_k \mathbf{S}_k + \frac{\beta_0 N_k}{\beta_0 + N_k} (\bar{\mathbf{x}}_k - \mathbf{m}_0)(\bar{\mathbf{x}}_k - \mathbf{m}_0)^T \quad (10.62)$$

$$\nu_k = \nu_0 + N_k \quad (10.63)$$

更新方程类似于混合高斯模型的最大似然解的EM算法的M步骤的方程。我们看到，为了更新模型参数上的变分后验概率分布，必须进行的计算涉及到在数据集上的求和操作与最大似然方法中的求和操作相同。

为了进行这个变分M步骤，我们需要得到表示“责任”的期望 $\mathbb{E}[z_{nk}] = r_{nk}$ 。这些可以通过对公式 (10.46) 给出的 ρ_{nk} 进行归一化的方式得到。我们看到，这个表达式涉及到关于变分分布的参数求期望，这些期望很容易求出，从而可得

$$\mathbb{E}_{\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k} [(\mathbf{x}_n - \boldsymbol{\mu}_k)^T \boldsymbol{\Lambda}_k (\mathbf{x}_n - \boldsymbol{\mu}_k)] = D \beta_k^{-1} + \nu_k (\mathbf{x}_n - \mathbf{m}_k)^T \mathbf{W}_k (\mathbf{x}_n - \mathbf{m}_k) \quad (10.64)$$

$$\ln \tilde{\Lambda}_k \equiv \mathbb{E}[\ln |\boldsymbol{\Lambda}_k|] = \sum_{i=1}^D \psi\left(\frac{\nu_k + 1 - i}{2}\right) + D \ln 2 + \ln |\mathbf{W}_k| \quad (10.65)$$

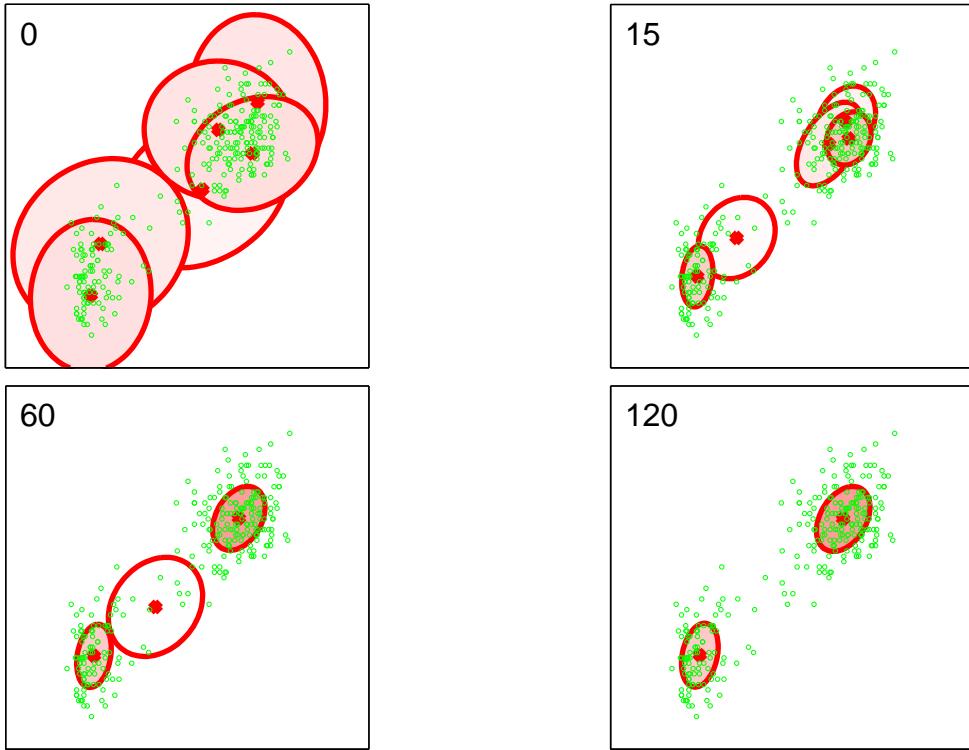


图 10.6: $K = 6$ 个高斯分布的变分贝叶斯混合，应用于老忠实间歇喷泉数据集，其中椭圆表示每个分量的概率密度的一个标准差位置的轮廓线，每个椭圆内部的红点对应于每个分量的混合系数的均值。每张图中左上角的数字表示变分推断迭代的次数。混合系数的期望在数值上与零无法区分的分量没有画出。

$$\ln \tilde{\pi}_k \equiv \mathbb{E}[\ln \pi_k] = \psi(\alpha_k) - \psi(\hat{\alpha}) \quad (10.66)$$

其中我们引入了 $\tilde{\Lambda}_k$ 和 $\tilde{\pi}_k$ 的定义， $\psi(\cdot)$ 是公式 (B.25) 定义的 Digamma 函数， $\hat{\alpha} = \sum_k \alpha_k$ 。公式 (10.65) 和公式 (10.66) 是从 Wishart 分布和狄利克雷分布的标准性质中得到的。

如果我们将公式 (10.64)、(10.65) 和 (10.66) 代入公式 (10.46)，然后使用公式 (10.49)，我们得到了下面的“责任”的结果

$$r_{nk} \propto \tilde{\pi}_k \tilde{\Lambda}_k^{\frac{1}{2}} \exp \left\{ -\frac{D}{2\beta_k} - \frac{\nu_k}{2} (\mathbf{x}_n - \mathbf{m}_k)^T \mathbf{W}_k (\mathbf{x}_n - \mathbf{m}_k) \right\} \quad (10.67)$$

注意这个结果与最大似然 EM 算法得到的“责任”的对应结果的相似性，后者根据公式 (9.13) 可以写成

$$r_{nk} \propto \pi_k |\Lambda_k|^{\frac{1}{2}} \exp \left\{ -\frac{1}{2} (\mathbf{x}_n - \boldsymbol{\mu}_k)^T \Lambda_k (\mathbf{x}_n - \boldsymbol{\mu}_k) \right\} \quad (10.68)$$

其中我们使用精度代替了协方差，来强调它与公式 (10.67) 之间的相似性。

因此变分后验概率分布的最优化涉及到在两个阶段之间进行循环，这两个阶段类似于最大似然 EM 算法的 E 步骤和 M 步骤。在变分推断的与 E 步骤等价的步骤中，我们使用当前状态下模型参数上的概率分布来计算公式 (10.64)、(10.65) 和 (10.66) 中的各阶矩，从而计算 $\mathbb{E}[z_{nk}] = r_{nk}$ 。然后，在接下来的与 M 步骤等价的步骤中，我们令这些“责任”保持不变，然后使用它们通过公式 (10.57) 和 (10.59) 重新计算参数上的变分分布。在任何一种情形下，我们看到变分后验概率的形式与联合概率分布 (10.41) 中对应因子的函数形式相同。这是一个一般的结果，是由于选择了共轭先验所造成的。

图 10.6 给出了将这种方法应用于老忠实间歇喷泉数据集上的结果。使用的模型是高斯混合模型，有 $K = 6$ 个分量。我们看到，在收敛之后，只有两个分量的混合系数的期望值可以与它们的先验值区分开。这种效果可以根据贝叶斯模型中数据拟合与模型复杂度之间的折中来定性地理解。这种模型中的复杂度惩罚的来源是参数被推离了它们的先验值。对于解释数据点没有作用的分量满足 $r_{nk} \simeq 0$ ，从而 $N_k \simeq 0$ 。根据公式 (10.58)，我们看到 $\alpha_k \simeq \alpha_0$ 。根据公式

(10.60) 至 (10.63)，我们看到其他的参数回到了它们的先验值。原则上，这些分量会微小地适应于数据点，但是对于一大类先验分布来说，这种微小的调整的效果太小了，以至于无法在数值上看出来。对于高斯混合模型，后验概率分布中的混合系数的期望值为

$$\mathbb{E}[\pi_k] = \frac{\alpha_0 + N_k}{K\alpha_0 + N} \quad (10.69)$$

考虑一个分量，其中 $N_k \simeq 0$ 且 $\alpha_k \simeq \alpha_0$ 。如果先验概率分布很宽，从而 $\alpha_0 \rightarrow 0$ ，那么 $\mathbb{E}[\pi_k] \rightarrow 0$ ，分量对模型不起作用。而如果先验概率与混合系数密切相关，即 $\alpha_0 \rightarrow \infty$ ，那么 $\mathbb{E}[\pi_k] \rightarrow \frac{1}{K}$ 。

在图10.6中，混合系数上的先验概率分布是一个狄利克雷分布，形式为 (10.39)。回忆一下，根据图2.5，对于 $\alpha_0 < 1$ ，先验概率分布倾向于选择某些混合系数趋近于零的解。图10.6是使用 $\alpha_0 = 10^{-3}$ 得到的结果，产生了两个混合系数非零的分量。如果我们选择 $\alpha_0 = 1$ ，那么我们得到三个混合系数非零的分量，对于 $\alpha = 10$ ，所有六个分量的混合系数都不等于零。

正如我们已经看到的那样，高斯分布的贝叶斯混合的变分解与最大似然的EM算法的解很相似。事实上，如果我们考虑 $N \rightarrow \infty$ 的极限情况，那么贝叶斯方法就收敛于最大似然方法的EM解。对于不是特别小的数据集来说，高斯混合模型的变分算法的主要的计算代价来自于“责任”的计算，以及加权数据协方差矩阵的计算与求逆。这些计算与最大似然EM算法中产生的计算相对应，因此使用这种贝叶斯方法几乎没有更多的计算代价。然而，这种方法有一些重要的优点。首先，在最大似然方法中，当一个高斯分量“退化”到一个具体的数据点时，会产生奇异性，而这种奇异性在贝叶斯方法中不存在。实际上，如果我们简单地引d入一个先验分布，然后使用MAP估计而不是最大似然估计，这种奇异性就会被消除。此外，当我们在混合分布中将混合分量的数量 K 选得较大时，不会出现过拟合问题，正如我们在图10.6中看到的那样。最后，变分方法使得我们可以在确定混合分布中分量的最优数量时不必借助于交叉验证的技术。

10.2.2 变分下界

我们也可以很容易地计算这个模型的下界 (10.3)。在实际应用中，能够在重新估计期间监视模型的下界是很有用的，这可以用来检测是否收敛。它也可以为解的数学表达式和它们的软件执行提供一个有价值的检查，因为在迭代重新估计的每个步骤中，这个下界的值应该不会减小。我们可以进一步地使用变分下界检查更新方程的数学推导和它们的软件执行的正确性，方法是使用有限差来检查每次更新确实给出了下界的一个（具有限制条件的）极大值 (Svensén and Bishop, 2004)。

对于高斯分布的变分混合，下界 (10.3) 为

$$\begin{aligned} \mathcal{L} &= \sum_{\mathbf{Z}} \iiint q(\mathbf{Z}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}) \ln \left\{ \frac{p(\mathbf{X}, \mathbf{Z}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda})}{q(\mathbf{Z}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda})} \right\} d\boldsymbol{\pi} d\boldsymbol{\mu} d\boldsymbol{\Lambda} \\ &= \mathbb{E}[\ln p(\mathbf{X}, \mathbf{Z}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda})] - \mathbb{E}[\ln q(\mathbf{Z}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda})] \\ &= \mathbb{E}[\ln p(\mathbf{X} | \mathbf{Z}, \boldsymbol{\mu}, \boldsymbol{\Lambda})] + \mathbb{E}[\ln p(\mathbf{Z} | \mathbf{X})] + \mathbb{E}[\ln p(\boldsymbol{\pi})] + \mathbb{E}[\ln p(\boldsymbol{\mu}, \boldsymbol{\Lambda})] \\ &\quad - \mathbb{E}[\ln q(\mathbf{Z})] - \mathbb{E}[\ln q(\boldsymbol{\pi})] - \mathbb{E}[\ln q(\boldsymbol{\mu}, \boldsymbol{\Lambda})] \end{aligned} \quad (10.70)$$

其中，为了保持记号简洁，我们省略了 q 分布上的*上标，以及期望算符的下标，因为每个期望是关于它的所有参数进行计算的。下界的各项很容易计算，结果为

$$\begin{aligned} \mathbb{E}[\ln p(\mathbf{X} | \mathbf{Z}, \boldsymbol{\mu}, \boldsymbol{\Lambda})] &= \frac{1}{2} \sum_{k=1}^K N_k \left\{ \ln \tilde{\Lambda}_k - D\beta_k^{-1} - \nu_k \text{Tr}(\mathbf{S}_k \mathbf{W}_k) \right. \\ &\quad \left. - \nu_k (\bar{\mathbf{x}}_k - \mathbf{m}_k)^T \mathbf{W}_k (\bar{\mathbf{x}}_k - \mathbf{m}_k) - D \ln(2\pi) \right\} \end{aligned} \quad (10.71)$$

$$\mathbb{E}[\ln p(\mathbf{Z} | \boldsymbol{\pi})] = \sum_{n=1}^N \sum_{k=1}^K r_{nk} \ln \tilde{\pi}_k \quad (10.72)$$

$$\mathbb{E}[\ln p(\boldsymbol{\pi})] = \ln C(\boldsymbol{\alpha}_0) + (\alpha_0 - 1) \sum_{k=1}^K \ln \tilde{\pi}_k \quad (10.73)$$

$$\begin{aligned} \mathbb{E}[\ln p(\boldsymbol{\mu}, \boldsymbol{\Lambda})] &= \frac{1}{2} \sum_{k=1}^K \left\{ D \ln \left(\frac{\beta_0}{2\pi} \right) + \ln \tilde{\Lambda}_k - \frac{D\beta_0}{\beta_k} \right. \\ &\quad \left. - \beta_0 \nu_0 (\mathbf{m}_k - \mathbf{m}_0)^T \mathbf{W}_k (\mathbf{m}_k - \mathbf{m}_0) \right\} + K \ln B(\mathbf{W}_0, \nu_0) \end{aligned} \quad (10.74)$$

$$+ \frac{\nu_0 - D - 1}{2} \sum_{k=1}^K \ln \tilde{\Lambda}_k - \frac{1}{2} \sum_{k=1}^K \nu_k \text{Tr}(\mathbf{W}_0^{-1} \mathbf{W}_k)$$

$$\mathbb{E}[\ln q(\mathbf{Z})] = \sum_{n=1}^N \sum_{k=1}^K r_{nk} \ln r_{nk} \quad (10.75)$$

$$\mathbb{E}[\ln q(\boldsymbol{\pi})] = \sum_{k=1}^K (\alpha_k - 1) \ln \tilde{\pi}_k + \ln C(\boldsymbol{\alpha}) \quad (10.76)$$

$$\mathbb{E}[\ln q(\boldsymbol{\mu}, \boldsymbol{\Lambda})] = \sum_{k=1}^K \left\{ \frac{1}{2} \ln \tilde{\Lambda}_k + \frac{D}{2} \ln \left(\frac{\beta_k}{2\pi} \right) - \frac{D}{2} - H[q(\boldsymbol{\Lambda}_k)] \right\} \quad (10.77)$$

其中 D 是 \mathbf{x} 的维度, $H[q(\boldsymbol{\Lambda}_k)]$ 是公式 (B.82) 给出的 Wishart 分布的熵, 系数 $C(\boldsymbol{\alpha})$ 和 $B(\mathbf{W}, \nu)$ 分别由公式 (B.23) 和公式 (B.79) 定义。注意, 涉及到 q 分布的对数的期望的项仅仅表示这些分布的熵的负值。当这些表达式进行加和给出下界的表达式时, 某些项可以组合到一起, 使表达式得到简化。然而, 我们将各个表达式分开写, 为了理解更容易。

最后, 值得注意的一点是, 下界提供了另一种推导变分重估计方程的方法 (变分重估计方程在 10.2.1 节已经得到)。为了说明这一点, 我们使用下面的事实: 由于模型有共轭先验, 因此变分后验分布 (即 \mathbf{Z} 的离散分布、 $\boldsymbol{\pi}$ 的狄利克雷分布以及 $(\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k)$ 的高斯-Wishart 分布) 的函数形式是已知的。通过使用这些分布的一般的参数形式, 我们可以推导出下界的形式, 将下界作为概率分布的参数的函数。关于这些参数最大化下界就会得到所需的重估计方程。

10.2.3 预测概率密度

在高斯模型的贝叶斯混合的应用中, 我们通常对观测变量的新值 $\hat{\mathbf{x}}$ 的预测概率密度感兴趣。与这个观测相关联的有一个潜在变量 $\hat{\mathbf{z}}$, 从而预测概率分布为

$$p(\hat{\mathbf{x}} | \mathbf{X}) = \sum_{\hat{\mathbf{z}}} \iiint p(\hat{\mathbf{x}} | \hat{\mathbf{z}}, \boldsymbol{\mu}, \boldsymbol{\Lambda}) p(\hat{\mathbf{z}} | \boldsymbol{\pi}) p(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda} | \mathbf{X}) d\boldsymbol{\pi} d\boldsymbol{\mu} d\boldsymbol{\Lambda} \quad (10.78)$$

其中 $p(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda} | \mathbf{X})$ 是参数的 (未知) 真实后验概率分布。使用公式 (10.37) 和公式 (10.38), 我们可以首先完成在 $\hat{\mathbf{z}}$ 上的求和, 得到

$$p(\hat{\mathbf{x}} | \mathbf{X}) = \sum_{k=1}^K \iiint \pi_k \mathcal{N}(\hat{\mathbf{x}} | \boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k^{-1}) p(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda} | \mathbf{X}) d\boldsymbol{\pi} d\boldsymbol{\mu} d\boldsymbol{\Lambda} \quad (10.79)$$

由于剩下的积分是无法计算的, 因此我们通过将真实后验概率分布 $p(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda} | \mathbf{X})$ 用它的变分近似 $q(\boldsymbol{\pi})q(\boldsymbol{\mu}, \boldsymbol{\Lambda})$ 替换的方式来近似预测概率分布, 结果为

$$p(\hat{\mathbf{x}} | \mathbf{X}) \simeq \sum_{k=1}^K \iiint \pi_k \mathcal{N}(\hat{\mathbf{x}} | \boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k^{-1}) q(\boldsymbol{\pi}) q(\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k) d\boldsymbol{\pi} d\boldsymbol{\mu}_k d\boldsymbol{\Lambda}_k \quad (10.80)$$

其中我们使用了公式 (10.55) 给出的分解方式, 并且在每一项中, 我们已经隐式地将 $j \neq k$ 的全部 $\{\boldsymbol{\mu}_j, \boldsymbol{\Lambda}_j\}$ 变量积分出去。剩余的积分现在可以解析地计算, 得到一个学生 t 分布的混合, 即

$$p(\hat{\mathbf{x}} | \mathbf{X}) \simeq \frac{1}{\hat{\alpha}} \sum_{k=1}^K \alpha_k \text{St}(\hat{\mathbf{x}} | \mathbf{m}_k, \mathbf{L}_k, \nu_k + 1 - D) \quad (10.81)$$

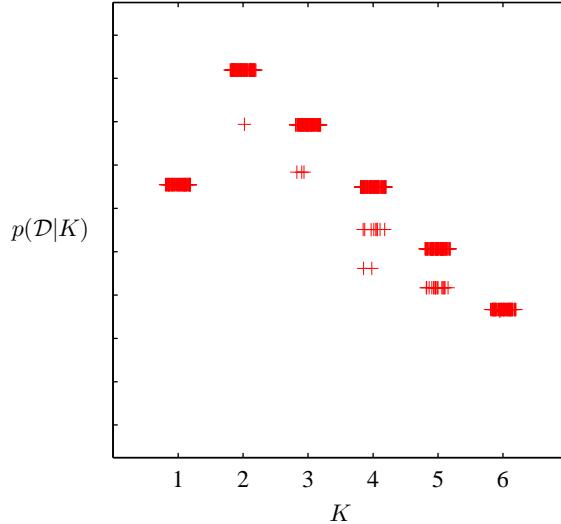


图 10.7: 变分下界 \mathcal{L} 与高斯混合模型的分量的数量 K 的关系图像，数据集是老忠实间歇喷泉的数据。图中展示了 $K = 2$ 个分量时的不同的峰值。对于每个 K 值，模型使用 100 个不同的起始点进行训练，结果用“+”符号表示。图像中，水平方向被施加了微小的扰动，从而它们可以被区分开。注意，某些解找到了次优的局部极大值，但是这个不经常发生。

其中第 k 个分量的均值为 m_k ，精度为

$$\mathbf{L}_k = \frac{(\nu_k + 1 - D)\beta_k}{1 + \beta_k} \mathbf{W}_k \quad (10.82)$$

其中 ν_k 由公式 (10.63) 给出。当数据集的大小 N 很大时，预测分布 (10.81) 就变成了高斯混合。

10.2.4 确定分量的数量

我们已经看到，变分下界可以用来确定具有 K 个分量的混合模型的后验概率分布。然而，这里有一个需要强调的比较微妙的地方。对于高斯混合模型的任意给定的参数设置（除了一些特殊的退化的设置之外），会存在一些其他的参数设置，对于这些参数设置，观测变量上的概率密度是完全相同的。这些参数值的差别仅仅是由于分量的重新标记产生的。例如，考虑两个高斯分布的混合以及一个单一的观测变量 x ，其中参数值为 $\pi_1 = a, \pi_2 = b, \mu_1 = c, \mu_2 = d, \sigma_1 = e, \sigma_2 = f$ 。那么对于参数值 $\pi_1 = b, \pi_2 = a, \mu_1 = d, \mu_2 = c, \sigma_1 = f, \sigma_2 = e$ ，即两个分量被交换，此时根据对称性，会给出同样的 $p(x)$ 值。如果我们有一个由 K 个分量组成的混合模型，那么每个参数设置都是 $K!$ 个等价设置中的一个。

在最大似然方法中，这种冗余性是不相关的，因为参数最优化算法（例如 EM 算法）会依赖于参数的初始值，找到一个具体的解，其他的等价的解不起作用。然而，在贝叶斯方法中，我们对所有可能的参数进行积分或求和。我们已经在图 10.3 中看到了，如果真实的后验概率分布是多峰的，那么基于最小化 $\text{KL}(q \parallel p)$ 的变分推断会倾向于在某一个峰值的邻域内近似这个分布，而忽视其他的峰值。由于等价的峰值具有等价的预测分布，因此只要我们考虑一个具有具体的数量 K 个分量组成的模型，那么这种等价性就无需担心。然而，如果我们项比较不同的 K 值，那么我们需要考虑这种多峰性。一个简单的近似解法是当我们进行模型比较和平均时，在下界中增加一项 $\ln K!$ 。

图 10.7 给出了包含多峰值因子的下界关于分量数量 K 的关系图像，数据集是老忠实间歇喷泉的数据。值得再次强调的是，最大似然方法会使得似然函数的值随着 K 的值单调递增（假设奇异解已经被避开，并且不考虑局部极大值的效果），因此不能够用于确定一个合适的模型复杂度。相反，贝叶斯推断自动地进行了模型复杂度和数据拟合之间的折中。

这种确定 K 的方法需要对一组具有不同 K 值的模型进行训练和比较。另一种确定一个合适的 K 值的方法是将混合系数 π 看成参数，通过关于 π 最大化下界的方式来对它们的值进行点估计（Corduneanu and Bishop, 2001），这种方法没有使用纯粹的贝叶斯方法为它们保留一个概率分布。这种方法会得到下面的重估计方程

$$\pi_k = \frac{1}{N} \sum_{n=1}^N r_{nk} \quad (10.83)$$

并且最大化过程与剩余参数上的分布 q 的变分更新过程相互交织在一起。对于解释数据集的贡献比较小的分量会让它们的混合系数在最优化的过程中趋于零，因此它们通过自动相关性确定（automatic relevance determination）的方式从模型中移除。这使得我们可以进行一轮训练，这一轮训练开始时，我们选择一个相对较大的 K 的初始值，然后让多于的分量从模型中被剪枝出去。关于超参数进行最优化时的稀疏性的来源已经在相关向量机中详细讨论过。

10.2.5 诱导分解

在推导高斯混合模型的这些变分更新方程时，我们假定了对变分后验概率分布的一种特定的分解方式，由公式（10.42）给定。然而，不同因子的最优解给出了额外的分解。特别地， $q^*(\mu, \Lambda)$ 的最优解由每个混合分量 k 上的独立分布 $q^*(\mu_k, \Lambda_k)$ 的乘积给定，而公式（10.48）给定的潜在变量上的变分后验概率分布 $q^*(Z)$ 可以分解为每个观测 n 的独立概率分布 $q^*(z_n)$ （注意它不能关于 k 进行分解，因为对于每个 n 值， z_{nk} 需要满足在 k 上的加和等于1的限制）。这些额外的分解的产生原因是假定的分解方式与真实分布的条件独立性质相互作用的结果，正如图10.5所示的有向图所描述的那样。

我们会把这些额外的分解方式成为诱导分解（induced factorizations），因为它们产生于在变分后验分布中假定的分解方式与真实联合概率分布的条件独立性质之间的相互作用。在变分方法的数值实现中，考虑这些附加的分解方式很重要。例如，对于一组变量上的高斯分布来说，如果分布的最优形式的精度矩阵总是对角矩阵（对应于关于由那个高斯分布独立描述的变量的分解方式），那么在计算过程中始终保留一个完整的精度矩阵是一种很低效的做法。

使用一种基于d-划分的简单的图检测方法，这种诱导的分解方式可以很容易地被检测到。我们将潜在变量划分为三个互斥的组 A, B, C ，然后让我们假定我们可以在变量 C 与剩余变量之间进行分解，即

$$q(A, B, C) = q(A, B)q(C) \quad (10.84)$$

使用一般的结果（10.9）以及概率的乘积规则，我们看到 $q(A, B)$ 的最优解为

$$\begin{aligned} \ln q^*(A, B) &= \mathbb{E}_C[\ln p(X, A, B, C)] + \text{常数} \\ &= \mathbb{E}_C[\ln p(A, B | X, C)] + \text{常数} \end{aligned} \quad (10.85)$$

我们现在考察这个解能否在 A 和 B 之间进行分解，即是否有 $q^*(A, B) = q^*(A)q^*(B)$ 。当且仅当 $\ln p(A, B | X, C) = \ln p(A | X, C) + \ln p(B | X, C)$ 时，这种情况成立，也就是说，下面的条件独立关系应该满足。

$$A \perp\!\!\!\perp B | X, C \quad (10.86)$$

我们也可以使用d-划分准则来检测对于任意的 A 和 B 的选择，这个关系是否确实成立。

为了说明这一点，再次考虑由图10.5中的有向图表示的高斯分布的贝叶斯混合，其中我们假定变分分解由公式（10.42）给出。我们立刻就可以看到，参数上的变分后验概率分布一定可以在 π 和剩余的参数 μ 和 Λ 之间进行分解，因为所有将 π 与 μ 或者 Λ 相连接的路径一定通过某个 z_n 结点，所有这些 z_n 结点都在我们的条件独立性检测的条件集合中，并且所有的 z_n 结点关于这种路径都是头到尾的。

10.3 变分线性回归

作为变分推断的第二个例子，我们回到3.3节的贝叶斯线性回归模型中。在模型证据框架中，我们通过使用最大化似然函数的方法进行点估计，从而近似了在 α 和 β 上的积分。一个纯粹

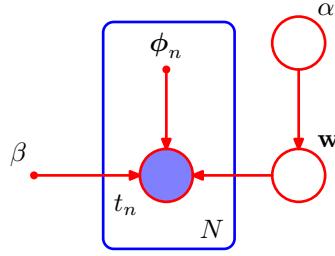


图 10.8: 表示贝叶斯线性回归模型的联合概率分布 (10.90) 的图模型。

的贝叶斯方法会对所有的超参数和参数进行积分。虽然精确的积分是无法计算的，但是我们可以使用变分方法来找到一个可以处理的近似。为了简化讨论，我们会假设噪声精度参数 β 已知，并且固定于它的真实值，虽然这个框架很容易扩展来包含 β 上的概率分布。对于线性回归模型来说，可以证明变分方法等价于模型证据的框架。尽管这样，这个例子给我们提供了使用变分方法的一个很好的练习，也是我们在10.6节讨论贝叶斯逻辑回归的变分方法的基础。

回忆一下， \mathbf{w} 的似然函数和 \mathbf{w} 上的先验概率分布为

$$p(\mathbf{t} \mid \mathbf{w}) = \prod_{n=1}^N \mathcal{N}(t_n \mid \mathbf{w}^T \phi_n, \beta^{-1}) \quad (10.87)$$

$$p(\mathbf{w} \mid \alpha) = \mathcal{N}(\mathbf{w} \mid \mathbf{0}, \alpha^{-1} \mathbf{I}) \quad (10.88)$$

其中 $\phi_n = \phi(\mathbf{x}_n)$ 。我们现在引入参数 α 上的先验概率分布。根据我们在2.3.6节的讨论，我们知道高斯分布的精度的共轭先验为Gamma分布，因此我们选择

$$p(\alpha) = \text{Gam}(\alpha \mid a_0, b_0) \quad (10.89)$$

其中 $\text{Gam}(\cdot \mid \cdot, \cdot)$ 由公式 (B.26) 定义。因此所有变量上的联合概率分布为

$$p(\mathbf{t}, \mathbf{w}, \alpha) = p(\mathbf{t} \mid \mathbf{w})p(\mathbf{w} \mid \alpha)p(\alpha) \quad (10.90)$$

这可以表示为图10.8中所示的有向图模型。

10.3.1 变分分布

我们的第一个目标是寻找对后验概率分布 $p(\mathbf{w}, \alpha \mid \mathbf{t})$ 的一个近似。为了完成这件事，我们使用10.1节的变分框架，变分后验概率分布的分解表达式为

$$q(\mathbf{w}, \alpha) = q(\mathbf{w})q(\alpha) \quad (10.91)$$

我们可以使用公式 (10.9) 给出的一般结果来找到这个分布中的因子的重估计方程。回忆一下，对于每个因子，我们取所有变量上的联合概率分布的对数，然后关于不在这个因子中的变量求平均。首先考虑 α 上的概率分布。只保留与 α 有函数依赖关系的项，我们有

$$\begin{aligned} \ln q^*(\alpha) &= \ln p(\alpha) + \mathbb{E}_{\mathbf{w}}[\ln p(\mathbf{w} \mid \alpha)] + \text{常数} \\ &= (a_0 - 1) \ln \alpha - b_0 \alpha + \frac{M}{2} \ln \alpha - \frac{\alpha}{2} \mathbb{E}[\mathbf{w}^T \mathbf{w}] + \text{常数} \end{aligned} \quad (10.92)$$

我们看到，这是Gamma分布的对数，因此通过观察 α 和 $\ln \alpha$ 的系数，我们有

$$q^*(\alpha) = \text{Gam}(\alpha \mid a_N, b_N) \quad (10.93)$$

其中

$$a_N = a_0 + \frac{M}{2} \quad (10.94)$$

$$b_N = b_0 + \frac{1}{2} \mathbb{E}[\mathbf{w}^T \mathbf{w}] \quad (10.95)$$

类似地，我们可以找到 \mathbf{w} 上的后验概率分布的变分重估计方程。与之前一样，使用一般的结果 (10.9)，只保留与 \mathbf{w} 有函数依赖关系的项，我们有

$$\ln q^*(\mathbf{w}) = \ln p(\mathbf{t} | \mathbf{w}) + \mathbb{E}_\alpha[\ln p(\mathbf{w} | \alpha)] + \text{常数} \quad (10.96)$$

$$= -\frac{\beta}{2} \sum_{n=1}^N \{\mathbf{w}^T \boldsymbol{\phi}_n - t_n\}^2 - \frac{1}{2} \mathbb{E}[\alpha] \mathbf{w}^T \mathbf{w} + \text{常数} \quad (10.97)$$

$$= -\frac{1}{2} \mathbf{w}^T (\mathbb{E}[\alpha] \mathbf{I} + \beta \boldsymbol{\Phi}^T \boldsymbol{\Phi}) \mathbf{w} + \beta \mathbf{w}^T \boldsymbol{\Phi}^T \mathbf{t} + \text{常数} \quad (10.98)$$

由于这是一个二次型，因此分布 $q^*(\mathbf{w})$ 是一个高斯分布，因此我们可以使用一般的配平方的方法，得到均值和协方差，结果为

$$q^*(\mathbf{w}) = \mathcal{N}(\mathbf{w} | \mathbf{m}_N, \mathbf{S}_N) \quad (10.99)$$

其中

$$\mathbf{m}_N = \beta \mathbf{S}_N \boldsymbol{\Phi}^T \mathbf{t} \quad (10.100)$$

$$\mathbf{S}_N = (\mathbb{E}[\alpha] \mathbf{I} + \beta \boldsymbol{\Phi}^T \boldsymbol{\Phi})^{-1} \quad (10.101)$$

注意这个结果与 α 被当成固定参数时得到的后验概率分布 (3.52) 的相似性。区别在于，这里 α 被替换为了它在变分分布下的期望 $\mathbb{E}[\alpha]$ 。实际上，在两种情形中，我们选择使用了同样的协方差矩阵 \mathbf{S}_N 的记号。

使用标准结果 (B.27)、(B.38) 和 (B.39)，我们可以得到所需的矩，形式为

$$\mathbb{E}[\alpha] = \frac{a_N}{b_N} \quad (10.102)$$

$$\mathbb{E}[\mathbf{w} \mathbf{w}^T] = \mathbf{m}_N \mathbf{m}_N^T + \mathbf{S}_N \quad (10.103)$$

变分后验概率分布的计算在开始时，对 $q(\mathbf{w})$ 或 $q(\alpha)$ 中的一个概率分布的参数进行初始化，然后交替地重新更新这些因子，直到满足一个合适的收敛准则（通常根据下界来确定，稍后讨论）。

将变分方法得到的解与 3.5 节使用模型证据得到的解练习起来是很有意义的。考虑 $a_0 = b_0 = 0$ 的情形，对应于 α 上的一个无限宽的鲜艳概率分布。变分后验概率 $q(\alpha)$ 的均值为

$$\mathbb{E}[\alpha] = \frac{a_N}{b_N} = \frac{\frac{M}{2}}{\frac{\mathbb{E}[\mathbf{w}^T \mathbf{w}]}{2}} = \frac{M}{\mathbf{m}_N^T \mathbf{m}_N + \text{Tr}(\mathbf{S}_N)} \quad (10.104)$$

与公式 (9.63) 进行对比，表明在这种特别简单的模型中，变分方法得到的解与使用EM算法最大化模型证据函数的方法得到的解完全相同，唯一的区别是 α 的点估计被替换为了它的期望值。由于分布 $q(\mathbf{w})$ 只通过期望 $\mathbb{E}[\alpha]$ 对 $q(\alpha)$ 产生依赖，因此我们看到这两种方法对于无限宽的先验概率分布会给出相同的结果。

10.3.2 预测分布

给定一个新的输入 \mathbf{x} ，使用参数的高斯变分后验概率很容易计算出 t 上的预测分布，即

$$\begin{aligned} p(t | \mathbf{x}, \mathbf{t}) &= \int p(t | \mathbf{x}, \mathbf{w}) p(\mathbf{w} | \mathbf{t}) d\mathbf{w} \\ &\simeq \int p(t | \mathbf{x}, \mathbf{w}) q(\mathbf{w}) d\mathbf{w} \\ &= \int \mathcal{N}(t | \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}), \beta^{-1}) \mathcal{N}(\mathbf{w} | \mathbf{m}_N, \mathbf{S}_N) d\mathbf{w} \\ &= \mathcal{N}(t | \mathbf{m}_N^T \boldsymbol{\phi}(\mathbf{x}), \sigma^2(\mathbf{x})) \end{aligned} \quad (10.105)$$

其中我们使用了公式 (2.115) 给出的线性高斯模型的结果计算积分。这里，与输入相关的方差为

$$\sigma^2(\mathbf{x}) = \frac{1}{\beta} + \boldsymbol{\phi}(\mathbf{x})^T \mathbf{S}_N \boldsymbol{\phi}(\mathbf{x}) \quad (10.106)$$

注意，这与我们固定 α 得到的结果 (3.59) 的形式相同，唯一的区别在于现在期望值 $\mathbb{E}[\alpha]$ 出现在 \mathbf{S}_N 的定义中。

10.3.3 下界

另一个很重要的量是下界 \mathcal{L} ，定义为

$$\begin{aligned} \mathcal{L}(q) &= \mathbb{E}[\ln p(\mathbf{w}, \alpha, \mathbf{t})] - \mathbb{E}[\ln q(\mathbf{w}, \alpha)] \\ &= \mathbb{E}_{\mathbf{w}}[\ln p(\mathbf{t} | \mathbf{w})] + \mathbb{E}_{\mathbf{w}, \alpha}[\ln p(\mathbf{w} | \alpha)] + \mathbb{E}_{\alpha}[\ln p(\alpha)] \\ &\quad - \mathbb{E}_{\alpha}[\ln q(\mathbf{w})]_{\mathbf{w}} - \mathbb{E}[\ln q(\alpha)] \end{aligned} \quad (10.107)$$

使用之前章节得到的结果，计算各项的值是很容易的，结果为

$$\begin{aligned} \mathbb{E}[\ln p(\mathbf{t} | \mathbf{w})]_{\mathbf{w}} &= \frac{N}{2} \ln \left(\frac{\beta}{2\pi} \right) - \frac{\beta}{2} \mathbf{t}^T \mathbf{t} + \beta \mathbf{m}_N^T \boldsymbol{\Phi}^T \mathbf{t} \\ &\quad - \frac{\beta}{2} \text{Tr}[\boldsymbol{\Phi}^T \boldsymbol{\Phi} (\mathbf{m}_N \mathbf{m}_N^T + \mathbf{S}_N)] \end{aligned} \quad (10.108)$$

$$\begin{aligned} \mathbb{E}[\ln p(\mathbf{w} | \alpha)]_{\mathbf{w}, \alpha} &= -\frac{M}{2} \ln(2\pi) + \frac{M}{2} (\psi(a_N) - \ln b_N) \\ &\quad - \frac{a_N}{2b_N} [\mathbf{m}_N^T \mathbf{m}_N + \text{Tr}(\mathbf{S}_N)] \end{aligned} \quad (10.109)$$

$$\begin{aligned} \mathbb{E}[\ln p(\alpha)]_{\alpha} &= a_0 \ln b_0 + (a_0 - 1)[\psi(a_N) - \ln b_N] \\ &\quad - b_0 \frac{a_N}{b_N} - \ln \Gamma(a_0) \end{aligned} \quad (10.110)$$

$$-\mathbb{E}[\ln q(\mathbf{w})]_{\mathbf{w}} = \frac{1}{2} \ln |\mathbf{S}_N| + \frac{M}{2} [1 + \ln(2\pi)] \quad (10.111)$$

$$-\mathbb{E}[\ln q(\alpha)]_{\alpha} = \ln \Gamma(a_N) - (a_N - 1)\psi(a_N) - \ln b_N + a_N \quad (10.112)$$

图10.9给出了下界 $\mathcal{L}(q)$ 与多项式模型的阶数的关系图像，数据集是从一个三阶多项式中人工生成的。这里，先验参数被设置为 $a_0 = b_0 = 0$ ，对应于无信息先验 $p(\alpha) \propto \frac{1}{\alpha}$ 。根据2.3.6节的讨论，它是 $\ln \alpha$ 上的均匀分布。正如我们在10.1节看到的那样， \mathcal{L} 表示模型的对数边缘似然函数 $\ln p(\mathbf{t} | M)$ 的下界。因此，变分框架将最高的概率赋予了 $M = 3$ 的模型。这与最大似然的结果相反。最大似然方法通过增加模型的复杂度尽可能地让误差变小，直到误差趋于零，这导致了最大似然方法倾向于选择具有严重过拟合现象的模型。

10.4 指数族分布

在第2章中，我们讨论了指数族概率分布和它们的共轭先验的重要作用。对于本书中讨论的许多模型来说，完整数据是服从指数族分布的。然而，通常这对于观测数据的边缘似然函数来说是不成立的。例如，在混合高斯模型中，观测数据 \mathbf{x}_n 和对应的隐含变量 z_n 的联合概率分布是指数族分布的成员，但是 \mathbf{x}_n 的边缘概率分布是高斯混合分布，因此不是指数族的成员。

目前为止，我们将模型中的变量分为了观测变量和隐含变量两组。我们现在进一步地将潜在变量和参数区分开。潜在变量（记作 Z ）是分散的（extensive），它的数量随着数据集规模的增大而增大。参数（记作 θ ）是聚集的（intensive），它的数量固定，与数据集的规模无关。例如，在高斯混合模型中，指示变量 z_{kn} （表示哪个分量 k 对生成数据点 \mathbf{x}_n 起作用）表示潜在变量，而均值 μ_k 、精度 Λ_k 以及混合系数 π_k 表示参数。

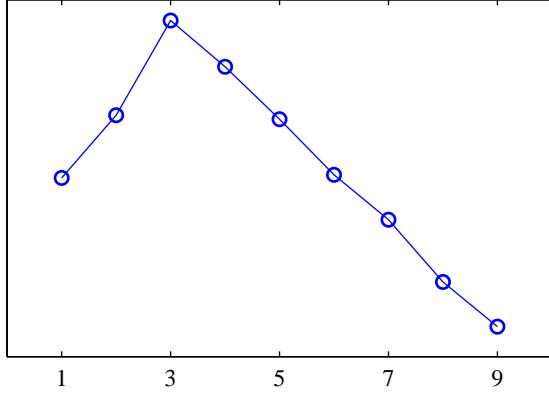


图 10.9: 对于一个多项式模型, 下界 \mathcal{L} 与多项式阶数 M 的关系曲线, 其中, 一个具有 10 个数据点的数据集由区间 $(-5, 5)$ 上的一个 $M = 3$ 的多项式生成, 同时附加了方差为 0.09 的高斯噪声。下界的值给出了模型的对数概率, 并且我们看到下界在 $M = 3$ 时达到峰值, 对应于生成数据集的真实模型。

考虑独立同分布数据的情形。我们将数据的值记作 $\mathbf{X} = \{\mathbf{x}_n\}$, 其中 $n = 1, \dots, N$, 对应的潜在变量为 $\mathbf{Z} = \{z_n\}$ 。现在假设观测变量和隐含变量的联合概率分布为指数族分布的成员, 参数为自然参数 $\boldsymbol{\eta}$, 即

$$p(\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\eta}) = \prod_{n=1}^N h(\mathbf{x}_n, z_n) g(\boldsymbol{\eta}) \exp\{\boldsymbol{\eta}^T \mathbf{u}(\mathbf{x}_n, z_n)\} \quad (10.113)$$

我们也会使用 $\boldsymbol{\eta}$ 的一个共轭先验, 它可以写成

$$p(\boldsymbol{\eta} \mid \nu_0, \chi_0) = f(\nu_0, \chi_0) g(\boldsymbol{\eta})^{\nu_0} \exp\{\nu_0 \boldsymbol{\eta}^T \chi_0\} \quad (10.114)$$

回忆一下, 共轭先验分布的意义为, 对于 \mathbf{u} 向量来说, 所有值为 χ_0 的观测的先验数量 ν_0 。现在考虑一个变分分布, 它可以在潜在变量和参数之间进行分解, 即 $q(\mathbf{Z}, \boldsymbol{\eta}) = q(\mathbf{Z})q(\boldsymbol{\eta})$ 。使用一般的结果 (10.9), 我们可以解出这两个因子, 如下所述。

$$\begin{aligned} \ln q^*(\mathbf{Z}) &= \mathbb{E}_{\boldsymbol{\eta}}[\ln p(\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\eta})] + \text{常数} \\ &= \sum_{n=1}^N \{\ln h(\mathbf{x}_n, z_n) + \mathbb{E}[\boldsymbol{\eta}^T] \mathbf{u}(\mathbf{x}_n, z_n)\} + \text{常数} \end{aligned} \quad (10.115)$$

因此我们看到它可以分解为一组相互独立的项的加和, 每个 n 都对应于一项, 因此 $q^*(\mathbf{Z})$ 的解可以在 n 上进行分解, 即 $q^*(\mathbf{Z}) = \prod_n q^*(z_n)$ 。这是诱导分解的一个例子。两侧取指数, 我们有

$$q^*(z_n) = h(\mathbf{x}_n, z_n) g(\mathbb{E}[\boldsymbol{\eta}]) \exp\{\mathbb{E}[\boldsymbol{\eta}^T] \mathbf{u}(\mathbf{x}_n, z_n)\} \quad (10.116)$$

其中归一化系数已经通过与指数族分布的标准形式进行比较的方式得到。

类似地, 对于参数上的变分分布, 我们有

$$\ln q^*(\boldsymbol{\eta}) = \ln p(\boldsymbol{\eta} \mid \nu_0, \chi_0) + \mathbb{E}_{\mathbf{Z}}[\ln p(\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\eta})] + \text{常数} \quad (10.117)$$

$$= \nu_0 \ln g(\boldsymbol{\eta}) + \nu_0 \boldsymbol{\eta}^T \chi_0 + \sum_{n=1}^N \{\ln g(\boldsymbol{\eta}) + \boldsymbol{\eta}^T \mathbb{E}_{z_n}[\mathbf{u}(\mathbf{x}_n, z_n)]\} + \text{常数} \quad (10.118)$$

与之前一样, 两侧取指数, 然后通过观察法确定归一化系数, 我们有

$$q^*(\boldsymbol{\eta}) = f(\nu_N, \chi_N) g(\boldsymbol{\eta})^{\nu_N} \exp\{\nu_N \boldsymbol{\eta}^T \chi_N\} \quad (10.119)$$

其中我们已经定义了

$$\nu_N = \nu_0 + N \quad (10.120)$$

$$\nu_N \chi_N = \nu_0 \chi_0 + \sum_{n=1}^N \mathbb{E}_{z_n} [\mathbf{u}(\mathbf{x}_n, z_n)] \quad (10.121)$$

注意, $q^*(z_n)$ 的解与 $q^*(\boldsymbol{\eta})$ 的解相互偶合, 因此我们可以使用一个两阶段的迭代方法进行求解。在变分E步骤中, 我们使用潜在变量上的当前后验概率分布 $q(z_n)$ 计算充分统计量的期望 $\mathbb{E}[\mathbf{u}(\mathbf{x}_n, z_n)]$, 并且使用这个结果计算参数上的修正的后验概率分布 $q(\boldsymbol{\eta})$ 。然后, 在接下来的变分M步骤中, 我们使用修正后的参数后验概率分布寻找自然参数的期望 $\mathbb{E}[\boldsymbol{\eta}^T]$, 它给出了潜在变量上的修正后的变分分布。

10.4.1 变分信息传递

我们通过详细讨论一个具体的模型来说明变分方法的应用, 这个模型是高斯模型的贝叶斯混合。这个模型可以被表示为图10.5中的有向图。这里我们从更一般的角度来讨论由有向图描述的模型中对变分方法的使用, 推导出一些具有广泛适用性的结果。

对应于有向图的联合概率分布可以写成下面的分解形式

$$p(\mathbf{x}) = \prod_i p(\mathbf{x}_i | \text{pa}_i) \quad (10.122)$$

其中 \mathbf{x}_i 表示与结点*i*关联的变量, pa_i 表示与结点*i*对应的父结点集合。注意, \mathbf{x}_i 可能是一个潜在变量, 也可能属于观测变量集合。现在, 考虑一个变分近似, 其中我们假定概率分布 $q(\mathbf{x})$ 可以关于 \mathbf{x}_i 进行分解, 即

$$q(\mathbf{x}) = \prod_i q_i(\mathbf{x}_i) \quad (10.123)$$

注意, 对于观测结点, 在变分分布中没有因子 $q(\mathbf{x}_i)$ 。我们现在将公示 (10.122) 代入我们的一般结果 (10.9) 中, 可得

$$\ln q_j^*(\mathbf{x}_j) = \mathbb{E}_{i \neq j} \left[\sum_i \ln p(\mathbf{x}_i | \text{pa}_i) \right] + \text{常数} \quad (10.124)$$

等式右侧的任何不依赖于 \mathbf{x}_j 的项都可以整合到可加性常数中。事实上, 唯一依赖于 \mathbf{x}_j 的项是由 $p(\mathbf{x}_j | \text{pa}_j)$ 给出的 \mathbf{x}_j 的条件概率分布以及任何在条件集合中具有 \mathbf{x}_j 的条件概率分布。根据定义, 这些条件概率分布对应于结点*j*的子结点, 因此他们也依赖于子结点的同父结点 (co-parents), 即子结点的除了结点 \mathbf{x}_j 本身之外的其他父结点。我们看到, $q_j^*(\mathbf{x}_j)$ 所依赖的所有结点组成的集合对应于结点 \mathbf{x}_j 的马尔科夫毯, 如图8.26所示。因此, 在变分后验概率分布中的更新因子表示图上的一个局部计算。这使得构建用于变分推断的具有一般性的软件成为可能, 在这种一般性的变分推断中, 模型的形式不必事先指定 (Bishop et al., 2003)。

如果我们现在确定模型的形式, 其中所有的条件概率分布都有一个共轭-指数族的结构, 那么变分推断的过程可以被转化为局部信息传递算法 (Winn and Bishop, 2005)。特别地, 对于一个特定的结点来说, 一旦它接收到来自所有的父结点和所有的子结点的信息, 那么与这个结点相关联的概率分布就可以被更新。这反过来需要子结点从它们的同父结点已经接收完毕信息。下界的计算也可以得到简化, 因为许多必要的值已经作为信息传递框架的一部分计算完毕。分布的信息传递形式有很好的缩放性质, 对于大的网络很合适。

10.5 局部变分方法

10.1节和10.2节讨论的变分框架可以被看做“全局”方法。之所以这样说, 是因为它直接寻找所有随机变量上的完整的后验概率分布的近似。另一种“局部”的方法涉及到寻找模型中的单独的变量或者变量组上定义的函数的界限。例如, 我们可能寻找条件概率分布 $p(y | x)$ 的界限, 这个条件概率本身仅仅是一个由有向图模型描述的更大的概率模型中的一个因子。引入界限的目的显然是简化最终得到的概率分布。这个局部近似可以应用于多个变量, 直到得到一个可以处理的近似。在10.6.1节, 我们会在logistic回归的问题中给出这种方法的一个实际例子。这里, 我们关注求解界限本身。

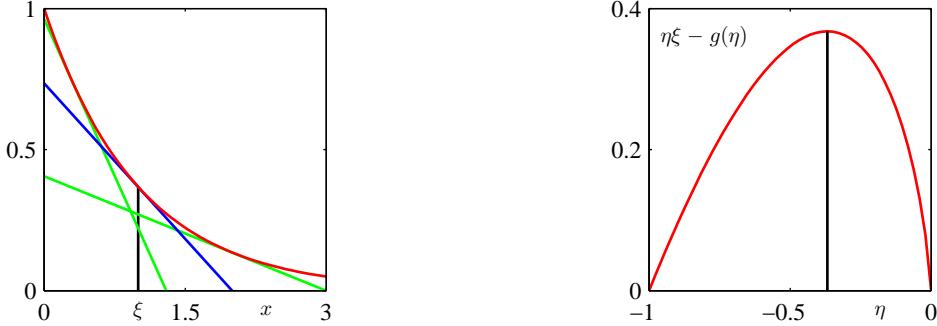


图 10.10: 在左图中, 红色曲线表示函数 $\exp(-x)$, 蓝色直线表示公式 (10.125) 定义的在 $x = \xi$ 处的切线, 其中 $\xi = 1$ 。这条直线的斜率为 $\eta = f'(\xi) = -\exp -\xi$ 。注意, 任何其他的切线, 例如绿色的切线, 在 $x = \xi$ 处都会有一个更小的 y 值。右图给出了函数 $\eta\xi - g(\eta)$ 关于 η 的图像, 其中 $g(\eta)$ 由公式 (10.131) 给出, $\xi = 1$, 此时最大值对应于 $\eta = -\exp(-\xi) = -1/e$ 。

我们已经看到, 在我们对 Kullback-Leibler 散度的讨论中, 对数函数的凸函数性质在求解全局变分方法的下界时起着关键的作用。我们将一个 (严格) 凸函数定义为每条弦都位于函数上方的函数。凸函数的性质对于局部变分的框架也起着核心的作用。注意, 我们的讨论同样适用于凹函数, 只需交换“最大值”运算与“最小值”运算, 将下界变为上界即可。

让我们首先考虑一个简单的例子, 即函数 $f(x) = \exp(-x)$, 它是 x 的一个凸函数, 如图 10.10 的左图所示。我们的目标是使用一个简单的函数来近似 $f(x)$, 特别地, 使用 x 的一个线性函数。根据图 10.10, 我们看到, 如果这个线性函数对应于一条切线, 那么它是函数 $f(x)$ 的下界。我们可以得到在一个具体的 x 处的 $y(x)$ 的切线, 例如 $x = \xi$ 处, 方法是使用一阶泰勒展开式

$$y(x) = f(\xi) + f'(\xi)(x - \xi) \quad (10.125)$$

从而 $y(x) \leq f(x)$, 且等号只在 $x = \xi$ 时成立。对于我们的例子, 函数 $f(x) = \exp(-x)$, 因此我们得到了切线的形式如下

$$y(x) = \exp(-\xi) - \exp(-\xi)(x - \xi) \quad (10.126)$$

它是一个以 ξ 为参数的线性函数。为了与后续的讨论相容, 让我们定义 $\eta = -\exp(-\xi)$, 即

$$y(x, \eta) = \eta x - \eta + \eta \ln(-\eta) \quad (10.127)$$

不同的 η 值对应于不同的切线, 并且由于所有的切线都是函数的下界, 因此我们有 $f(x) \geq y(x, \eta)$ 。因此我们可以将函数写成下面的形式

$$f(x) = \max_{\eta} \{ \eta x - \eta + \eta \ln(-\eta) \} \quad (10.128)$$

我们已经成功地用一个简单的线性函数 $y(x, \eta)$ 来近似凸函数 $f(x)$ 。代价是我们引入了一个变分参数 η , 并且为了得到最紧致的界限, 我们必须关于 η 进行最优化。

我们可以使用凸对偶 (convex duality) 的框架更加一般地形式化描述这种方法 (Rockafellar, 1972; Jordan et al., 1999)。考虑图 10.11 的左侧图给出的凸函数 $f(x)$ 。在这个例子中, 函数 ηx 是 $f(x)$ 的一个下界, 但不是斜率为 η 的线性函数能够达到的最好的下界, 因为最紧致的下界由切线给出。让我们将斜率为 η 的切线的方程写成 $\eta x - g(\eta)$, 其中截距 (的负值) $g(\eta)$ 显然依赖于切线的斜率 η 。为了确定截距, 我们注意到这条直线必须垂直移动一段距离, 这段距离等于直线和函数之间最小的垂直距离, 如图 10.11 所示。因此

$$\begin{aligned} g(\eta) &= -\min_x \{ f(x) - \eta x \} \\ &= \max_x \{ \eta x - f(x) \} \end{aligned} \quad (10.129)$$

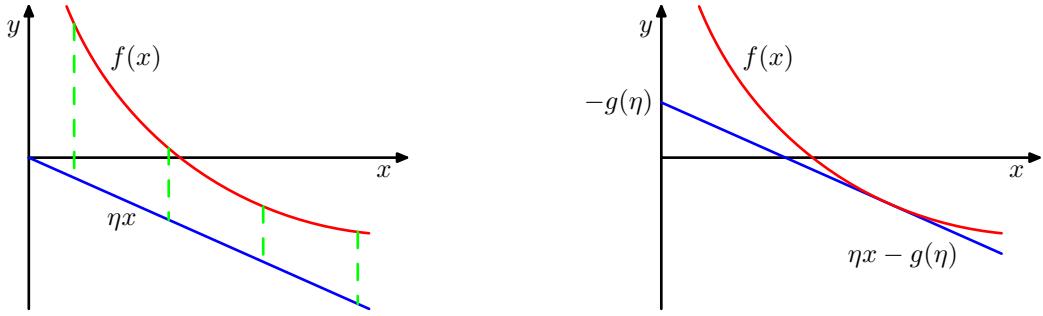


图 10.11: 在左图中, 红色曲线给出了一个凸函数 $f(x)$, 蓝色曲线表示线性函数 ηx , 它是 $f(x)$ 的一个下界, 因为对于所有的 x 都有 $f(x) > \eta x$ 。对于给定的斜率 η 的值, 具有相同斜率的切线的接触点可以通过关于 x 最小化差距 $f(x) - \eta x$ 的方式得到, 差距用绿色虚线表示。这定义了对偶函数 $g(\eta)$, 它对应于具有斜率 η 的切线的截距 (的负值)。

现在, 我们不去固定 η 改变 x , 而是可以考虑一个特定的 x 值, 然后调节 η , 直到切平面在这个特定的 x 处与函数 $f(x)$ 相切。由于在特定的 x 处, 当切线的 y 值与它的连接点的 y 值相等时, y 的值最大, 因此我们有

$$f(x) = \max_{\eta} \{\eta x - g(\eta)\} \quad (10.130)$$

我们看到函数 $f(x)$ 和 $g(\eta)$ 的角色是对偶的, 二者通过公式 (10.129) 和公式 (10.130) 相互关联。

让我们将这两个对偶关系应用到我们简单的例子 $f(x) = \exp(-x)$ 中。根据公式 (10.129), 我们看到 x 的最大值为 $\xi = -\ln(-\eta)$, 代回到公式中, 我们得到了共轭函数 $g(\eta)$, 形式为

$$g(\eta) = \eta - \eta \ln(-\eta) \quad (10.131)$$

与之前得到的结果相同。对于 $\xi = 1$ 的情况函数 $\eta \xi - g(\eta)$ 的图像如图 10.10 右侧所示。作为检查, 我们可以将公式 (10.131) 代入到公式 (10.130), 这给出了最大值 $\eta = -\exp(-x)$, 代回到公式中就恢复出了原始的函数 $f(x) = \exp(-x)$ 。

对于凹函数, 我们可以采用类似的推导方式, 得到上界, 其中“最大化”运算被替换为“最小化”运算, 即

$$f(x) = \min_{\eta} \{\eta x - g(\eta)\} \quad (10.132)$$

$$g(\eta) = \min_x \{\eta x - f(x)\} \quad (10.133)$$

如果感兴趣的函数不是凸函数 (或者凹函数), 那么我们不能直接应用这种方法得到上述界限。然而, 我们可以首先寻找函数或者参数的一个可逆变换, 这个变换将函数或者参数变换为一个凸函数的形式。然后, 我们计算共轭函数, 之后变换回原始的变量。

在模式识别中经常出现的一个重要的例子是 logistic sigmoid 函数, 它的定义为

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (10.134)$$

这个函数不是凹函数也不是凸函数。然而, 如果我们取对数, 那么我们就得到了一个凹函数, 这一点通过取二阶导数的方式很容易证明。根据公式 (10.133), 对应的共轭函数的形式为

$$g(\eta) = \min_x \{\eta x - f(x)\} = -\eta \ln \eta - (1 - \eta) \ln(1 - \eta) \quad (10.135)$$

我们看到, 它是一个二值变量的熵, 这个变量的取值为 1 的概率为 η 。使用公式 (10.132), 我们得到了对数 sigmoid 函数的一个上界

$$\ln \sigma(x) \leq \eta x - g(\eta) \quad (10.136)$$

然后取指数, 我们得到了 logistic sigmoid 函数的一个上界, 形式为

$$\sigma(x) \leq \exp(\eta x - g(\eta)) \quad (10.137)$$

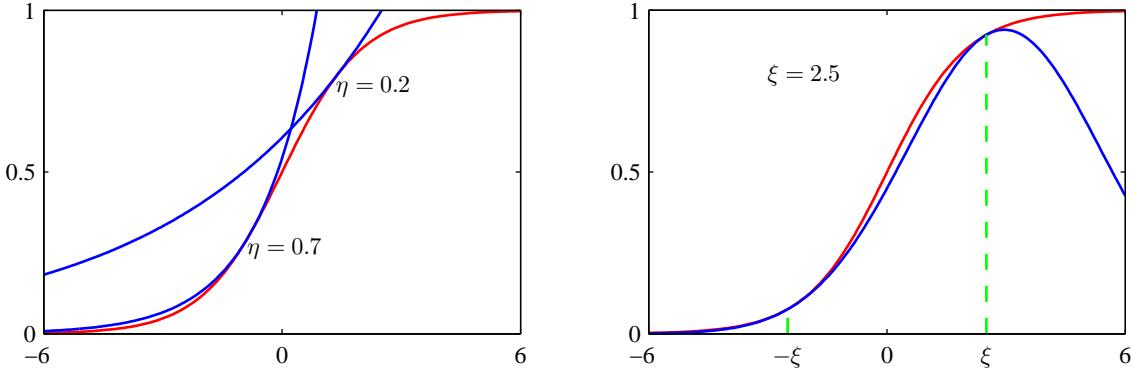


图 10.12: 左图中, 红色曲线给出了公式 (10.134) 定义的logistic sigmoid函数 $\sigma(x)$ 。同时给出的还有两个指数上界 (10.137) 的例子, 用蓝色曲线表示。右图再次用红色曲线给出了logistic sigmoid函数。同时给出的还有高斯下界 (10.144), 用蓝色曲线表示。这里, 参数 $\xi = 2.5$, 界限在 $x = \xi$ 和 $x = -\xi$ 出事精确的, 用绿色曲线标记。

对于两个不同的 η 值, 图像如图10.12的左图所示。

我们也可以得到sigmoid函数的下界, 下界的函数形式是高斯形式。为了完成这件事, 我们采用Jaakkola and Jordan (2000) 的方法, 对输入变量和函数本身都进行变换。首先, 我们取logistic函数的对数, 然后将其分解, 即

$$\begin{aligned}\ln \sigma(x) &= -\ln(1 + e^{-x}) = -\ln \left\{ e^{-\frac{x}{2}} \left(e^{\frac{x}{2}} + e^{-\frac{x}{2}} \right) \right\} \\ &= \frac{x}{2} - \ln \left(e^{\frac{x}{2}} + e^{-\frac{x}{2}} \right)\end{aligned}\quad (10.138)$$

我们现在注意到, 函数 $f(x) = -\ln \left(e^{\frac{x}{2}} + e^{-\frac{x}{2}} \right)$ 是变量 x^2 的一个凸函数, 这一点可以通过取二阶导数的方式证明。这产生了 $f(x)$ 的下界, 它是 x^2 的一个线性函数, 它的共轭函数为

$$g(\eta) = \max_{x^2} \left\{ \eta x^2 - f \left(\sqrt{x^2} \right) \right\} \quad (10.139)$$

根据驻点的条件可得

$$0 = \eta - \frac{dx}{dx^2} \frac{d}{dx} f(x) = \eta + \frac{1}{4x} \tanh \left(\frac{x}{2} \right) \quad (10.140)$$

如果我们将这个值记作 x , 对应于在这个特定的 η 值下, 函数与切线的接触点, 记作 η , 那么我们有

$$\eta = -\frac{1}{4\xi} \tanh \left(\frac{\xi}{2} \right) = -\frac{1}{2\xi} \left[\sigma(\xi) - \frac{1}{2} \right] = -\lambda(\xi) \quad (10.141)$$

其中, 我们定义了 $\lambda = -\eta$, 保持与Jaakkola and Jordan (2000) 的相容性。我们不把 λ 看成变分参数, 相反, 我们可以令 ξ 为变分参数, 因为这会产生共轭函数的更简单的表达式, 它的形式为

$$g(\lambda(\xi)) = -\lambda(\xi)\xi^2 - f(\xi) = -\lambda(\xi)\xi^2 + \ln \left(e^{\frac{\xi}{2}} + e^{-\frac{\xi}{2}} \right) \quad (10.142)$$

这里, $f(x)$ 的界限可以写成

$$f(x) \geq -\lambda(\xi)x^2 - g(\lambda(\xi)) = -\lambda(\xi)x^2 - \lambda(\xi)\xi^2 - \ln \left(e^{\frac{\xi}{2}} + e^{-\frac{\xi}{2}} \right) \quad (10.143)$$

sigmoid函数的界限就变成了

$$\sigma(x) \geq \sigma(\xi) \exp \left\{ \frac{x - \xi}{2} - \lambda(\xi)(x^2 - \xi^2) \right\} \quad (10.144)$$

其中 $\lambda(\xi)$ 的定义为 (10.141)。这个界限如图10.12的右图所示。我们看到，界限的函数形式是 x 的二次函数的指数形式。当我们寻找通过logistic sigmoid函数定义的后验概率分布的高斯表示时，这个界限的形式很有用。

logistic sigmoid函数在二值变量上的概率模型中经常出现，因为它是将log odds函数转换为后验概率分布的函数。对于多类分布，对应的变换由softmax函数给出。不幸的是，这里推导出logistic sigmoid函数的下界不能直接扩展到softmax函数。Gibbs (1997) 提出了一种构建高斯分布的方法，这个高斯分布被猜想为是一个界限（虽然没有给出严格的证明），这可以用于将局部变分方法应用到多分类问题。

我们会在10.6.1节看到局部变分界限的一个例子。然而，现阶段从一般的角度考虑这些界限如何被使用是很有意义的。假设我们想计算一个形式如下的积分

$$I = \int \sigma(a)p(a) da \quad (10.145)$$

其中 $\sigma(a)$ 是一个logistic sigmoid函数， $p(a)$ 是一个高斯概率密度。当我们项计算贝叶斯模型中的预测分布时，这种积分会经常出现，此时 $p(a)$ 表示一个后验参数分布。由于积分是无法直接计算的，因此我们使用变分界限 (10.144)，我们将它写成 $\sigma(a) \geq f(a, \xi)$ ，其中 ξ 是一个变分参数。积分现在变成了两个指数-二次函数的乘积，因此可以解析地求出积分，给出 I 的界限

$$I \geq \int f(a, \xi)p(a) da = F(\xi) \quad (10.146)$$

我们可以自由地选择变分参数 ξ ，这里我们选择最大化函数 $F(\xi)$ 的值 ξ^* 。得到的值 $F(\xi^*)$ 表示在所有的界限中最紧致的界限，可以用来近似 I 。然而，这个最优化的界通常不是精确的。虽然logistic sigmoid函数的界限 $\sigma(a) \geq f(a, \xi)$ 可以被精确地最优化，但是 ξ 的最优选择依赖于 a 的值，从而界限只对一个 a 的值是精确的。由于 $F(\xi)$ 可以通过对 a 的所有值上进行积分的方式得到，因此 ξ^* 的值表示一个折中，权值为概率分布 $p(a)$ 。

10.6 变分logistic回归

我们现在回到4.5节研究的贝叶斯logistic回归模型，说明局部变分方法的应用。在4.5节，我们将注意力集中于拉普拉斯近似的使用，而这里，我们考虑一种贝叶斯的方法，本方法基于Jaakkola and Jordan (2000) 的方法。与拉普拉斯方法相似，这也会生成后验概率分布的高斯近似。然而，变分方法的极大的灵活性使得模型的准确率与拉普拉斯相比有所提升。此外，与拉普拉斯方法不同，变分方法最优化一个具有良好定义的目标函数，这个目标函数由模型证据的一个严格界限给定。Dybowski and Roberts (2005) 也从贝叶斯的角度研究了logistic回归问题，使用了蒙特卡罗取样的技术。

10.6.1 变分后验概率分布

这里，我们会使用一种基于10.5节介绍的局部界限的变分方法。这使得logistic回归的似然函数（由logistic sigmoid函数控制）可以有指数的二次形式近似。因此，与之前一样，比较方便的做法是选择形式为 (4.140) 的共轭高斯先验。现阶段，我们会将超参数 m_0 和 S_0 看成固定的常数。在10.6.3节，我们会展示变分形式如何扩展到超参数未知的情形，这种情况下，超参数的值要从数据中进行推断。

在变分的框架上，我们寻找边缘似然函数的下界的最大值。对于贝叶斯logistic回归模型，边缘似然函数的形式为

$$p(\mathbf{t}) = \int p(\mathbf{t} | \mathbf{w})p(\mathbf{w}) d\mathbf{w} = \int \left[\prod_{n=1}^N p(t_n | \mathbf{w}) \right] p(\mathbf{w}) d\mathbf{w} \quad (10.147)$$

首先，我们注意到 t 的条件概率分布可以写成

$$\begin{aligned} p(t \mid \mathbf{w}) &= \sigma(a)^t \{1 - \sigma(a)\}^{1-t} \\ &= \left(\frac{1}{1 + e^{-a}} \right)^t \left(1 - \frac{1}{1 + e^{-a}} \right)^{1-t} \\ &= e^{at} \frac{e^{-a}}{1 + e^{-a}} = e^{at} \sigma(-a) \end{aligned} \quad (10.148)$$

其中 $a = \mathbf{w}^T \boldsymbol{\phi}$ 。为了得到 $p(\mathbf{t})$ 的下界，我们使用公式 (10.144) 给出的logistic sigmoid函数的变分下界。为了方便，我们在这里重新写一下。

$$\sigma(z) \geq \sigma(\xi) \exp \left\{ \frac{z - \xi}{2} - \lambda(\xi)(z^2 - \xi^2) \right\} \quad (10.149)$$

其中

$$\lambda(\xi) = \frac{1}{2\xi} \left[\sigma(\xi) - \frac{1}{2} \right] \quad (10.150)$$

于是，我们有

$$p(t \mid \mathbf{w}) = e^{at} \sigma(-a) \geq e^{at} \sigma(\xi) \exp \left\{ -\frac{a + \xi}{2} - \lambda(\xi)(a^2 - \xi^2) \right\} \quad (10.151)$$

注意，由于这个下界分别作用于似然函数的每一项，因此存在一个变分参数 ξ_n ，对应于训练集的每个观测 $(\boldsymbol{\phi}_n, t_n)$ 。使用 $a = \mathbf{w}^T \boldsymbol{\phi}$ ，乘以先验概率分布，我们可以得到下面的 \mathbf{t} 和 \mathbf{w} 的联合概率分布。

$$p(\mathbf{t}, \mathbf{w}) = p(\mathbf{t} \mid \mathbf{w})p(\mathbf{w}) \geq h(\mathbf{w}, \boldsymbol{\xi})p(\mathbf{w}) \quad (10.152)$$

其中， $\boldsymbol{\xi}$ 表示变分参数的集合 $\{\xi_n\}$ ，并且

$$\begin{aligned} h(\mathbf{w}, \boldsymbol{\xi}) &= \prod_{n=1}^N \sigma(\xi_n) \exp \{ \mathbf{w}^T \boldsymbol{\phi}_n t_n - (\mathbf{w}^T \boldsymbol{\phi}_n + \xi_n)/2 \\ &\quad - \lambda(\xi_n)([\mathbf{w}^T \boldsymbol{\phi}_n]^2 - \xi_n^2) \} \end{aligned} \quad (10.153)$$

精确计算这个后验概率分布需要对不等式的左侧进行归一化。由于这是无法计算的，因此我们反过来对右侧进行操作。注意，右侧的函数不能看成一个概率密度，因为它没有被归一化。但是，一旦它被归一化，表示一个后验概率分布 $q(\mathbf{w})$ ，它就不再表示下界了。

由于对数函数是单调递增的函数，因此不等式 $A \geq B$ 表示 $\ln A \geq \ln B$ 。这给出了 \mathbf{t} 和 \mathbf{w} 之间的联合概率分布的对数的下界，形式为

$$\begin{aligned} \ln \{p(\mathbf{t} \mid \mathbf{w})p(\mathbf{w})\} &\geq \ln p(\mathbf{w}) + \sum_{n=1}^N \{ \ln \sigma(\xi_n) + \mathbf{w}^T \boldsymbol{\phi}_n t_n \\ &\quad - (\mathbf{w}^T \boldsymbol{\phi}_n + \xi_n)/2 - \lambda(\xi_n)([\mathbf{w}^T \boldsymbol{\phi}_n]^2 - \xi_n^2) \} \end{aligned} \quad (10.154)$$

代入先验概率分布 $p(\mathbf{w})$ ，不等式的右侧变成了一个关于 \mathbf{w} 的函数，形式为

$$\begin{aligned} &- \frac{1}{2} (\mathbf{w} - \mathbf{m}_0)^T \mathbf{S}_0^{-1} (\mathbf{w} - \mathbf{m}_0) \\ &+ \sum_{n=1}^N \{ \mathbf{w}^T \boldsymbol{\phi}_n (t_n - 1/2) - \lambda(\xi_n) \mathbf{w}^T (\boldsymbol{\phi}_n \boldsymbol{\phi}_n^T) \mathbf{w} \} + \text{常数} \end{aligned} \quad (10.155)$$

这是 \mathbf{w} 的一个二次函数，因此我们可以通过分裂出 \mathbf{w} 的线性项和二次项，得到后验概率分布的对应的变分近似，这是一个高斯变分后验概率，形式为

$$q(\mathbf{w}) = \mathcal{N}(\mathbf{w} \mid \mathbf{m}_N, \mathbf{S}_N) \quad (10.156)$$

其中

$$\mathbf{m}_N = \mathbf{S}_N \left(\mathbf{S}_0^{-1} \mathbf{m}_0 + \sum_{n=1}^N \left(t_n - \frac{1}{2} \right) \boldsymbol{\phi}_n \right) \quad (10.157)$$

$$\mathbf{S}_N^{-1} = \mathbf{S}_0^{-1} + 2 \sum_{n=1}^N \lambda(\xi_n) \boldsymbol{\phi}_n \boldsymbol{\phi}_n^T \quad (10.158)$$

与拉格朗日框架一样，我们又一次得到了对后验概率分布的一个高斯近似。然后，变分参数 $\{\xi_n\}$ 提供的额外的灵活性使得这个近似的精度更高 (Jaakkola and Jordan, 2000)。

这里，我们考虑了一个批量学习的问题，其中所有的训练数据能够一次全部得到。然而，贝叶斯方法本质上相当适用于顺序学习的问题，其中数据点每次只处理一个，然后被丢弃。得到顺序学习情形下的变分方法的公式是很容易的。

注意，公式 (10.149) 给出的下界只适用于二分类问题，因此这个方法不能直接推广到 $K > 2$ 个类别的多类问题。Gibbs (1997) 研究了多分类问题的另一种下界的形式。

10.6.2 最优化变分参数

我们现在得到了后验概率分布的一个归一化的高斯近似。我们稍后会使用这个近似得到对于新数据的预测分布。然而，首先我们需要通过最大化边缘似然函数的下界，确定变分参数 $\{\xi_n\}$ 。

为了完成这一点，我们首先将不等式 (10.152) 代回到边缘似然函数，可得

$$\ln p(\mathbf{t}) = \ln \int p(\mathbf{t} | \mathbf{w}) p(\mathbf{w}) d\mathbf{w} \geq \ln \int h(\mathbf{w}, \boldsymbol{\xi}) p(\mathbf{w}) d\mathbf{w} = \mathcal{L}(\boldsymbol{\xi}) \quad (10.159)$$

与3.5节的线性回归模型的超参数 α 的最优化一样，有两种方法确定 ξ_n 。在第一种方法中，我们看到函数 $\mathcal{L}(\boldsymbol{\xi})$ 由 \mathbf{w} 上的积分定义，因此我们可以将 \mathbf{w} 看成一个潜在变量，然后使用EM算法。在第二种方法中，我们解析地对 \mathbf{w} 积分，然后直接关于 $\boldsymbol{\xi}$ 进行最大化。让我们首先考虑EM方法。

在EM算法中，首先选择参数 $\{\xi_n\}$ 的某个初始值，我们将这些初始值聚集在一起，记作 $\{\boldsymbol{\xi}\}^\text{旧}$ 。然后在EM算法的E步骤中，我们使用这些参数值找到 \mathbf{w} 上的后验概率分布，它由公式 (10.156) 给出。之后在M步骤中，我们最大化完整数据似然函数的期望，形式为

$$Q(\boldsymbol{\xi}, \boldsymbol{\xi}^\text{旧}) = \mathbb{E}[\ln\{h(\mathbf{w}, \boldsymbol{\xi}) p(\mathbf{w})\}] \quad (10.160)$$

其中期望是关于使用 $\boldsymbol{\xi}^\text{旧}$ 得到的后验概率分布 $q(\mathbf{w})$ 进行计算的。注意， $p(\mathbf{w})$ 不依赖于 $\boldsymbol{\xi}$ ，代入 $h(\mathbf{w}, \boldsymbol{\xi})$ ，我们有

$$Q(\boldsymbol{\xi}, \boldsymbol{\xi}^\text{旧}) = \sum_{n=1}^N \left\{ \ln \sigma(\xi_n) - \frac{\xi_n}{2} - \lambda(\xi_n) (\boldsymbol{\phi}_n^T \mathbb{E}[\mathbf{w} \mathbf{w}^T] \boldsymbol{\phi}_n - \xi_n^2) \right\} + \text{常数} \quad (10.161)$$

其中，“常数”表示与 $\boldsymbol{\xi}$ 无关的项。我们现在令关于 ξ_n 的导数等于零。经过简单的代数推导，使用 $\sigma(\xi)$ 和 $\lambda(\xi)$ ，有

$$0 = \lambda'(\xi_n) (\boldsymbol{\phi}_n^T \mathbb{E}[\mathbf{w} \mathbf{w}^T] \boldsymbol{\phi}_n - \xi_n^2) \quad (10.162)$$

现在，我们注意到，对于 $\xi \geq 0$ ， $\lambda'(\xi)$ 是 ξ 的一个单调函数，并且由于界限在 $\xi = 0$ 两侧的对称性，我们可以将我们的注意力限制在 ξ 的非负部分而不失一般性。因此， $\lambda'(\xi) \neq 0$ ，从而我们得到了下面的重估计方程

$$(\xi_n^\text{新})^2 = \boldsymbol{\phi}_n^T \mathbb{E}[\mathbf{w} \mathbf{w}^T] \boldsymbol{\phi}_n = \boldsymbol{\phi}_n^T (\mathbf{S}_N + \mathbf{m}_N \mathbf{m}_N^T) \boldsymbol{\phi}_n \quad (10.163)$$

推导过程中我们使用了公式 (10.156)。

让我们总结一下寻找变分后验概率分布的EM算法。首先，我们初始化变分参数 $\boldsymbol{\xi}^\text{旧}$ 。在E步骤中，我们计算由公式 (10.156) 给出的 \mathbf{w} 上的后验概率分布，其中均值和协方差分别由公式 (10.157) 和公式 (10.158) 定义。在M步骤中，我们使用这个变分后验概率，计算由公式

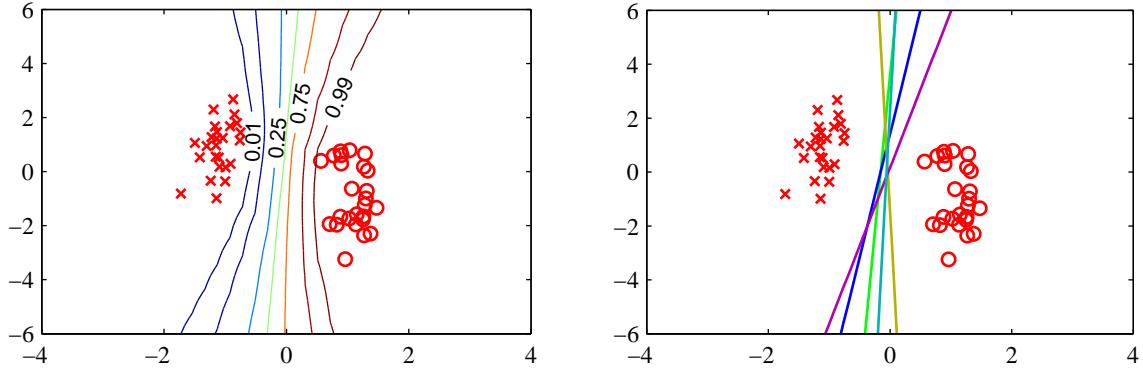


图 10.13: logistic 回归的贝叶斯方法的例子。数据集是一个简单的线性可分的数据集。左图给出了使用变分推断的方法得到的预测分布。我们看到决策边界大致位于数据点的聚类的中间位置，并且预测分布的轮廓线在远离数据点的位置发生分叉，这反映出了在这些区域进行分类的不确定性。右图给出了对应于从后验概率分布 $p(\mathbf{w} | \mathbf{t})$ 中抽取的参数 \mathbf{w} 的五个样本点的决策边界。

(10.163) 给出一个新的 ξ 值。不断重复 E 步骤和 M 步骤，直到满足一个适当的收敛准则，这在实际应用中通常只需要几步迭代。

我们介绍另一种得到 ξ 的重估计方程的方法。我们注意到，在下界 $\mathcal{L}(\xi)$ 的定义 (10.159) 中的关于 \mathbf{w} 的积分中，被积函数的形式类似于高斯分布，因此积分可以解析地计算。计算出这个积分之后，我们可以关于 ξ_n 进行求导。可以证明，这种方法得到的重估计方程与之前用 EM 方法得到的方程 (10.163) 完全相同。

正如我们已经强调过的那样，在变分方法的应用中，能够计算出由公式 (10.159) 给出的下界 $\mathcal{L}(\xi)$ 是很有用的。我们注意到 $p(\mathbf{w})$ 是一个高斯分布， $h(\mathbf{w}, \xi)$ 是 \mathbf{w} 的二次函数的指数形式，从而我们可以解析地计算 \mathbf{w} 上的积分。因此，通过配平方的方法，然后使用高斯分布的归一化系数的标准结果，我们可以得到解的精确形式如下

$$\begin{aligned}\mathcal{L}(\xi) = & \frac{1}{2} \ln \frac{|\mathbf{S}_N|}{|\mathbf{S}_0|} + \frac{1}{2} \mathbf{m}_N^T \mathbf{S}_N^{-1} \mathbf{m}_N - \frac{1}{2} \mathbf{m}_0^T \mathbf{S}_0^{-1} \mathbf{m}_0 \\ & + \sum_{n=1}^N \left\{ \ln \sigma(\xi_n) - \frac{1}{2} \xi_n + \lambda(\xi_n) \xi_n^2 \right\}\end{aligned}\quad (10.164)$$

变分框架也可以应用于数据顺序到达的情形 (Jaakkola and Jordan, 2000)。在这种情况下，我们保持 \mathbf{w} 上的一个高斯后验概率分布，它使用先验概率分布 $p(\mathbf{w})$ 进行初始化。随着每个数据点的到达，使用界限 (10.151)，然后归一化，我们就可以对后验概率进行更新，得到一个更新后的后验概率分布。

通过对后验概率分布进行积分，我们可以得到预测分布，它的形式与 4.5.2 节讨论的拉普拉斯近似的形式相同。图 10.13 给出了人工生成数据集的变分预测分布。这个例子为 7.1 节讨论的“大边缘”的概念提供了一些有趣的认识。“大边缘”的概念与贝叶斯的解有着定性的相似的行为。

10.6.3 超参数的推断

目前为止，我们将先验概率分布的超参数 α 看成一个已知参数。我们现在将贝叶斯 logistic 回归模型进行推广，使得这个参数的值可以从数据集中推断出来。这可以通过将全局变分近似和局部变分近似结合到一个框架中的方式完成，从而在每个阶段都保留边缘似然函数的下界。Bishop and Svensén (2003) 在研究专家模型的层次混合的贝叶斯方法中，采用了这样一种组合的方法。

特别地，我们再次考虑一个简单的各向同性的高斯先验概率分布，形式为

$$p(\mathbf{w} | \alpha) = \mathcal{N}(\mathbf{w} | \mathbf{0}, \alpha^{-1} \mathbf{I}) \quad (10.165)$$

我们的分析可以推广到更一般的高斯先验分布中，例如，如果我们希望为参数 w_j 的不同子集关联一个不同的超参数，那么我们就可以将我们的分析进行推广。与之前一样，我们考虑 α 上的共轭超先验，这是一个Gamma分布

$$p(\alpha) = \text{Gam}(\alpha | \alpha_0, b_0) \quad (10.166)$$

它由常数 a_0 和 b_0 控制。

这个模型的边缘似然函数现在的形式为

$$p(\mathbf{t}) = \iint p(\mathbf{w}, \alpha, \mathbf{t}) d\mathbf{w} d\alpha \quad (10.167)$$

其中，联合概率分布为

$$p(\mathbf{w}, \alpha, \mathbf{t}) = p(\mathbf{t} | \mathbf{w})p(\mathbf{w} | \alpha)p(\alpha) \quad (10.168)$$

我们现在无法直接计算关于 \mathbf{w} 和 α 的积分。我们会在同一个模型中使用全局的变分方法和局部的变分方法来解决这个问题。

首先，我们引入一个变分分布 $q(\mathbf{w}, \alpha)$ ，然后应用公式 (10.2) 给出的分解方式。在这种情况下，它的形式为

$$\ln p(\mathbf{t}) = \mathcal{L}(q) + \text{KL}(q \| p) \quad (10.169)$$

其中，下界 $\mathcal{L}(q)$ 和Kullback-Leibler散度 $\text{KL}(q \| p)$ 的定义为

$$\mathcal{L}(q) = \iint q(\mathbf{w}, \alpha) \ln \left\{ \frac{p(\mathbf{w}, \alpha, \mathbf{t})}{q(\mathbf{w}, \alpha)} \right\} d\mathbf{w} d\alpha \quad (10.170)$$

$$\text{KL}(q \| p) = - \iint q(\mathbf{w}, \alpha) \ln \left\{ \frac{p(\mathbf{w}, \alpha | \mathbf{t})}{q(\mathbf{w}, \alpha)} \right\} d\mathbf{w} d\alpha \quad (10.171)$$

现在，由于似然因子 $p(\mathbf{t} | \mathbf{w})$ 的形式，下界 $\mathcal{L}(q)$ 仍然无法求解。于是，与之前一样，我们对每个logistic sigmoid因子应用一个局部的变分界限。这使得我们可以使用不等式 (10.152)，得到 $\mathcal{L}(q)$ 的下界，这个下界也是对数似然函数的一个下界。

$$\begin{aligned} \ln p(\mathbf{t}) &\geq \mathcal{L}(q) \geq \tilde{\mathcal{L}}(q, \boldsymbol{\xi}) \\ &= \iint q(\mathbf{w}, \alpha) \ln \left\{ \frac{h(\mathbf{w}, \boldsymbol{\xi})p(\mathbf{w} | \alpha)p(\alpha)}{q(\mathbf{w}, \alpha)} \right\} d\mathbf{w} d\alpha \end{aligned} \quad (10.172)$$

接下来我们假设变分分布可以在参数和超参数之间进行分解，即

$$q(\mathbf{w}, \alpha) = q(\mathbf{w})q(\alpha) \quad (10.173)$$

有了这种分解，我们可以使用公式 (10.9) 给出的一般结果，得到最优因子的表达式。首先考虑概率分布 $q(\mathbf{w})$ 。丢弃与 \mathbf{w} 无关的项，我们有

$$\begin{aligned} \ln q(\mathbf{w}) &= \mathbb{E}_\alpha [\ln \{h(\mathbf{w}, \boldsymbol{\xi})p(\mathbf{w} | \alpha)p(\alpha)\}] + \text{常数} \\ &= \ln h(\mathbf{w}, \boldsymbol{\xi}) + \mathbb{E}_\alpha [\ln p(\mathbf{w} | \alpha)] + \text{常数} \end{aligned}$$

我们现在使用公式 (10.153) 消去 $\ln h(\mathbf{w}, \boldsymbol{\xi})$ ，使用公式 (10.165) 消去 $\ln p(\mathbf{w} | \alpha)$ ，有

$$\ln q(\mathbf{w}) = -\frac{\mathbb{E}[\alpha]}{2}\mathbf{w}^T \mathbf{w} + \sum_{n=1}^N \left\{ (t_n - \frac{1}{2})\mathbf{w}^T \boldsymbol{\phi}_n - \lambda(\xi_n)\mathbf{w}^T \boldsymbol{\phi}_n \boldsymbol{\phi}_n^T \mathbf{w} \right\} + \text{常数}$$

我们看到这是 \mathbf{w} 的一个二次函数，因此 $q(\mathbf{w})$ 的解是高斯分布。使用通常的配平方方法，我们有

$$q(\mathbf{w}) = \mathcal{N}(\mathbf{w} | \boldsymbol{\mu}_N, \boldsymbol{\Sigma}_N) \quad (10.174)$$

其中我们定义了

$$\boldsymbol{\Sigma}_N^{-1} \boldsymbol{\mu}_N = \sum_{n=1}^N \left(t_n - \frac{1}{2} \right) \boldsymbol{\phi}_n \quad (10.175)$$

$$\boldsymbol{\Sigma}_N^{-1} = \mathbb{E}[\alpha] \mathbf{I} + 2 \sum_{n=1}^N \lambda(\xi_n) \boldsymbol{\phi}_n \boldsymbol{\phi}_n^T \quad (10.176)$$

类似地，因子 $q(\alpha)$ 的最优解为

$$\ln q(\alpha) = \mathbb{E}_{\mathbf{w}} [\ln p(\mathbf{w} \mid \alpha)] + \ln p(\alpha) + \text{常数}$$

使用公式 (10.165) 消去 $\ln p(\mathbf{w} \mid \alpha)$ ，使用公式 (10.166) 消去 $\ln p(\alpha)$ ，我们有

$$\ln q(\alpha) = \frac{M}{2} \ln \alpha - \frac{\alpha}{2} \mathbb{E}[\mathbf{w}^T \mathbf{w}] + (\alpha_0 - 1) \ln \alpha - b_0 \alpha + \text{常数}$$

我们看到这是一个Gamma分布的对数，因此我们有

$$q(\alpha) = \text{Gam}(\alpha \mid a_N, b_N) = \frac{1}{\Gamma(a_N)} a_N^{b_N} \alpha^{a_N-1} e^{-b_N \alpha} \quad (10.177)$$

其中

$$a_N = a_0 + \frac{M}{2} \quad (10.178)$$

$$b_N = b_0 + \frac{1}{2} \mathbb{E}_{\mathbf{w}} [\mathbf{w}^T \mathbf{w}] \quad (10.179)$$

我们还需要最优化变分参数 ξ_n ，这也可以通过最大化下界 $\tilde{L}(q, \xi)$ 的方式得到。略去与 ξ 无关的项，对 α 积分，我们有

$$\tilde{L}(q, \xi) = \int q(\mathbf{w}) \ln h(\mathbf{w}, \xi) d\mathbf{w} + \text{常数} \quad (10.180)$$

注意，它的形式与公式 (10.160) 的形式完全相同，因此我们可以使用我们之前的结果 (10.163)，它可以通过直接对边缘似然函数的最优化得到，从而重估计方程的形式为

$$(\xi_n^{\text{新}})^2 = \boldsymbol{\phi}_n^T (\boldsymbol{\Sigma}_N + \boldsymbol{\mu}_N \boldsymbol{\mu}_N^T) \boldsymbol{\phi}_n \quad (10.181)$$

我们已经得到了三个量 $q(\mathbf{w})$, $q(\alpha)$ 和 ξ 的重估计方程，因此在进行合适的最优化之后，我们可以在这些量之间进行循环，每次都对各个量进行更新。所要求解的各阶矩为

$$\mathbb{E}[\alpha] = \frac{a_N}{b_N} \quad (10.182)$$

$$\mathbb{E}[\mathbf{w} \mathbf{w}^T] = \boldsymbol{\Sigma}_N + \boldsymbol{\mu}_N \boldsymbol{\mu}_N^T \quad (10.183)$$

10.7 期望传播

在本章的最后一节，我们讨论确定性近似推断的另一种形式，被称为期望传播 (expectation propagation)，或者EP (Minka, 2001a; Minka, 2001b)。与目前为止讨论的变分贝叶斯方法相同，这种方法也基于对Kullback-Leibler散度的最小化，但是现在形式相反，从而得到了性质相当不同的近似结果。

先考虑关于 $q(z)$ 最小化 $\text{KL}(p \parallel q)$ 的问题，其中 $p(z)$ 是一个固定概率分布， $q(z)$ 是指数族分布的一个成员，因此根据公式 (2.194)，可以写成

$$q(z) = h(z) g(\eta) \exp\{\eta^T u(z)\} \quad (10.184)$$

作为 η 的一个函数，Kullback-Leibler散度变成了

$$\text{KL}(p \parallel q) = -\ln g(\boldsymbol{\eta}) - \boldsymbol{\eta}^T \mathbb{E}_{p(z)}[\mathbf{u}(z)] + \text{常数} \quad (10.185)$$

其中常数项与自然参数 $\boldsymbol{\eta}$ 无关。我们可以通过令关于 $\boldsymbol{\eta}$ 的梯度等于零的方式，在这个概率分布族中最小化 $\text{KL}(p \parallel q)$ ，结果为

$$-\nabla \ln g(\boldsymbol{\eta}) = \mathbb{E}_{p(z)}[\mathbf{u}(z)] \quad (10.186)$$

然而，我们已经看到，在公式(2.226)中， $\ln g(\boldsymbol{\eta})$ 的负梯度有概率分布 $q(z)$ 下 $\mathbf{u}(z)$ 的期望给定。令这两个结果相等，我们有

$$\mathbb{E}_{q(z)}[\mathbf{u}(z)] = \mathbb{E}_{p(z)}[\mathbf{u}(z)] \quad (10.187)$$

我们看到，最优解仅仅对应于将充分统计量的期望进行匹配。因此，例如，如果 $q(z)$ 是一个高斯分布 $N(z \mid \boldsymbol{\mu}, \boldsymbol{\Sigma})$ ，那么我们通过令 $q(z)$ 的均值 $\boldsymbol{\mu}$ 等于分布 $p(z)$ 的均值并且令协方差 $\boldsymbol{\Sigma}$ 等于 $p(z)$ 的协方差，即可最小化Kullback-Leibler散度。这有时被称为矩匹配(moment matching)。图10.3(a)给出了这个的一个例子。

现在，让我们利用这个结果，得到近似推断的一个实用的算法。对于许多概率模型来说，数据 \mathcal{D} 和隐含变量(包括参数) $\boldsymbol{\theta}$ 的联合概率分布由一组因子的乘积组成，形式为

$$p(\mathcal{D}, \boldsymbol{\theta}) = \prod_i f_i(\boldsymbol{\theta}) \quad (10.188)$$

这个结果可能由独立同分布的数据的模型产生，其中对于每个数据点 x_n ，都有一个因子 $f_n(\boldsymbol{\theta}) = p(x_n \mid \boldsymbol{\theta})$ ，且因子 $f_0(\boldsymbol{\theta}) = p(\boldsymbol{\theta})$ 对应于先验概率分布。更一般地，它也适用于任何由有向图定义的模型，其中每个因子是一个条件概率分布，对应于一个结点。也适用于无向图，其中每个因子是一个团块势函数。我们感兴趣的是计算后验概率分布 $p(\boldsymbol{\theta} \mid \mathcal{D})$ 用于进行预测，以及计算模型证据 $p(\mathcal{D})$ 用于进行模型比较。根据公式(10.188)，后验概率分布为

$$p(\boldsymbol{\theta} \mid \mathcal{D}) = \frac{1}{p(\mathcal{D})} \prod_i f_i(\boldsymbol{\theta}) \quad (10.189)$$

模型证据为

$$p(\mathcal{D}) = \int \prod_i f_i(\boldsymbol{\theta}) \, d\boldsymbol{\theta} \quad (10.190)$$

这里，我们考虑连续变量，但是下面的讨论同样适用于离散变量，只需把积分替换为求和即可。我们假设 $\boldsymbol{\theta}$ 上的边缘概率分布以及关于用来进行预测的后验概率分布的边缘分布都是无法计算的，从而需要某种形式的近似。

期望传播基于后验概率分布的近似，这个近似也由一组因子的乘积给出，即

$$q(\boldsymbol{\theta}) = \frac{1}{Z} \prod_i \tilde{f}_i(\boldsymbol{\theta}) \quad (10.191)$$

其中，近似中的每个因子 $\tilde{f}_i(\boldsymbol{\theta})$ 对应于真实后验概率分布(10.189)中的一个因子 $f_i(\boldsymbol{\theta})$ ，因子 $\frac{1}{Z}$ 是归一化常数，用来确保公式(10.191)的左侧的积分等于1。为了得到一个实用的算法，我们需要对因子 $\tilde{f}_i(\boldsymbol{\theta})$ 进行一定的限制，特别地，我们会假定因子来自指数族分布。于是，因子的乘积也是指数族分布，因此可以用充分统计量的有限集合来描述。例如，如果每个 $\tilde{f}_i(\boldsymbol{\theta})$ 是一个高斯分布，那么整体的近似 $q(\boldsymbol{\theta})$ 也是高斯分布。

理想情况下，我们通过最小化真实后验概率分布与近似分布之间的Kullback-Leibler散度的方式来确定 $\tilde{f}_i(\boldsymbol{\theta})$ ，这个散度为

$$\text{KL}(p \parallel q) = \text{KL} \left(\frac{1}{p(\mathcal{D})} \prod_i f_i(\boldsymbol{\theta}) \middle\| \frac{1}{Z} \prod_i \tilde{f}_i(\boldsymbol{\theta}) \right) \quad (10.192)$$

注意与变分推断中使用的KL散度相比，这个KL的散度恰好相反。通常这个最小化是无法进行的，因为KL散度涉及到关于真实概率分布求平均。作为一个粗略的近似，我们反过来最小化对应的因子对 $f_i(\theta)$ 和 $\tilde{f}_i(\theta)$ 之间的KL散度。这个问题容易得多，并且具有算法无需迭代的优点。然而，由于每个因子被各自独立地进行近似，因此因子的乘积的近似效果可能很差。

期望传播通过在所有剩余因子的环境中对每个因子进行优化，从而取得了一个效果好得多的近似。首先，这种方法初始化因子 $\tilde{f}_i(\theta)$ ，然后在因子之间进行循环，每次优化一个因子。这种方法的思想类似于之前讨论的变分贝叶斯框架的因子更近过程。假设我们希望优化因子 $\tilde{f}_j(\theta)$ 。首先，我们将这个因子从乘积中移除，得到 $\prod_{i \neq j} \tilde{f}_i(\theta)$ 。从概念上讲，我们要确定因子 $f_j(\theta)$ 的一个修正形式，使得乘积

$$q^{\text{新}}(\theta) \propto \tilde{f}_j(\theta) \prod_{i \neq j} \tilde{f}_i(\theta) \quad (10.193)$$

尽可能地接近

$$f_j(\theta) \prod_{i \neq j} \tilde{f}_i(\theta) \quad (10.194)$$

其中我们保持所有 $i \neq j$ 的因子 $\tilde{f}_i(\theta)$ 固定。这保证了近似在由剩余的因子定义的后验概率较高的区域最精确。后面，当我们将EP应用于“聚类问题”的时候，我们会看到这种效果的一个例子。为了完成这一点，我们首先从当前的对后验概率的近似中移除因子 $\tilde{f}_j(\theta)$ ，方法是定义下面的未归一化的分布

$$q^{\setminus j}(\theta) = \frac{q(\theta)}{\tilde{f}_j(\theta)} \quad (10.195)$$

注意，我们反过来从 $i \neq j$ 的因子的乘积中求出 $q^{\setminus j}(\theta)$ ，虽然在实际应用中，除法通常更容易。它现在与因子 $f_j(\theta)$ 结合，得到概率分布

$$\frac{1}{Z_j} f_j(\theta) q^{\setminus j}(\theta) \quad (10.196)$$

其中 Z_j 是归一化常数，形式为

$$Z_j = \int f_j(\theta) q^{\setminus j}(\theta) d\theta \quad (10.197)$$

我们现在通过最小化Kullback-Leibler散度

$$\text{KL}\left(\frac{f_j(\theta) q^{\setminus j}(\theta)}{Z_j} \middle\| q^{\text{新}}(\theta)\right) \quad (10.198)$$

来确定一个修正的因子 $\tilde{f}_j(\theta)$ 。这很容易求解，因为近似分布 $q^{\text{新}}(\theta)$ 来自指数族分布，因此我们可以使用结果(10.187)，这个公式告诉我们，参数 $q^{\text{新}}(\theta)$ 可以通过匹配公式(10.196)的对应矩的充分统计量的期望的方式获得。我们会假设这是一个可以计算的操作。例如，如果我们选择 $q(\theta)$ 为高斯概率分布 $\mathcal{N}(\theta | \mu, \Sigma)$ ，那么 μ 被设置为(未归一化的)分布 $f_j(\theta) q^{\setminus j}(\theta)$ 的均值， Σ 被设置为它的方差。更一般地，得到指数族分布的任意成员的所需的分布是很容易的，只要它能够被归一化即可，因为充分统计量的期望可以与归一化系数的导数相关联，正如公式(2.226)所述。图10.14说明了EP近似的过程。

根据公式(10.193)，我们看到修正的因子 $\tilde{f}_j(\theta)$ 可以按照下面的方法得到：取 $q^{\text{新}}(\theta)$ ，然后除以剩余的因子，即

$$\tilde{f}_j(\theta) = K \frac{q^{\text{新}}(\theta)}{q^{\setminus j}(\theta)} \quad (10.199)$$

其中我们使用了公式(10.195)。系数 K 通过下面的方式确定：将等式(10.199)的两侧乘以 $q^{\setminus j}(\theta)$ ，然后积分，可得

$$K = \int \tilde{f}_j(\theta) q^{\setminus j}(\theta) d\theta \quad (10.200)$$

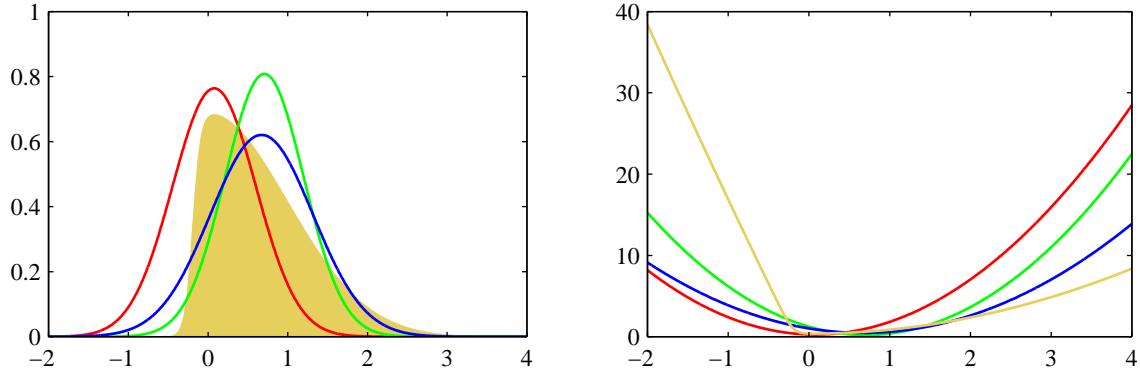


图 10.14: 用高斯分布进行期望传播近似的说明, 使用了之前在图4.14和图10.1中讨论的例子。左图给出了原始的概率分布 (黄色) 以及拉普拉斯近似 (红色)、全局变分近似 (绿色) 以及EP近似 (蓝色), 右图给出了对应的概率分布的负对数。注意, EP分布比变分推断得到的分布更宽, 这是由于不同形式的KL散度造成的结果。

其中我们已经使用了 $q^{\text{新}}(\boldsymbol{\theta})$ 已经归一化这一事实。于是, K 的值可以通过匹配零阶矩的方式得到

$$\int \tilde{f}_j(\boldsymbol{\theta}) q^{\backslash j}(\boldsymbol{\theta}) d\boldsymbol{\theta} = \int f_j(\boldsymbol{\theta}) q^{\backslash j}(\boldsymbol{\theta}) d\boldsymbol{\theta} \quad (10.201)$$

将这个式子与公式 (10.197) 结合, 我们看到 $K = Z_j$, 因此可以通过计算公式 (10.197) 中的积分的方式得到。

在实际应用中, 在因子集合中会进行多次迭代, 每次都修正所有的因子。之后, 使用公式 (10.191) 可以得到后验概率分布 $p(\boldsymbol{\theta} | \mathcal{D})$ 的近似, 模型证据 $p(\mathcal{D})$ 可以使用公式 (10.190) 来近似, 其中因子 $f_i(\boldsymbol{\theta})$ 被替换为它们的近似 $\tilde{f}_i(\boldsymbol{\theta})$ 。

我们给定观测数据集 \mathcal{D} 和随机变量 $\boldsymbol{\theta}$ 上的联合概率分布, 用因子的乘积的形式表示

$$p(\mathcal{D}, \boldsymbol{\theta}) = \prod_i f_i(\boldsymbol{\theta}) \quad (10.202)$$

我们希望使用下面形式的分布

$$q(\boldsymbol{\theta}) = \frac{1}{Z} \prod_i \tilde{f}_i(\boldsymbol{\theta}) \quad (10.203)$$

来近似后验概率分布 $p(\boldsymbol{\theta} | \mathcal{D})$ 。我们也希望近似模型证据 $p(\mathcal{D})$ 。

- 初始化所有的近似因子 $\tilde{f}_i(\boldsymbol{\theta})$ 。
- 通过设置

$$q(\boldsymbol{\theta}) \propto \prod_i \tilde{f}_i(\boldsymbol{\theta}) \quad (10.204)$$

初始化后验近似。

- 直到收敛:
 - 选择一个因子 $\tilde{f}_j(\boldsymbol{\theta})$ 进行优化。
 - 通过下面的除法

$$q^{\backslash j}(\boldsymbol{\theta}) = \frac{q(\boldsymbol{\theta})}{\tilde{f}_j(\boldsymbol{\theta})} \quad (10.205)$$

从后验概率分布中移除 $\tilde{f}_j(\boldsymbol{\theta})$ 。

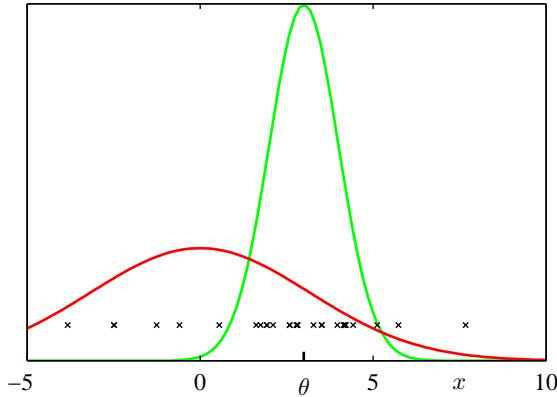


图 10.15: 维度为 $D = 1$ 的数据空间中的聚类问题的说明。训练数据点（用叉号表示），从两个高斯分布混合而成的分布中抽出，高斯分量用红色和蓝色表示。我们的目标是从观测数据中推断绿色高斯分布的均值。

- 计算新的后验概率分布，方法为：令 $q^{\text{新}}(\boldsymbol{\theta})$ 的充分统计量（矩）等于 $q^{\backslash j}(\boldsymbol{\theta})f_j(\boldsymbol{\theta})$ 的充分统计量（矩），包括计算归一化系数

$$Z_j = \int q^{\backslash j}(\boldsymbol{\theta})f_j(\boldsymbol{\theta}) d\boldsymbol{\theta} \quad (10.206)$$

- 计算和存储新的因子

$$\tilde{f}_j(\boldsymbol{\theta}) = Z_j \frac{q^{\text{新}}(\boldsymbol{\theta})}{q^{\backslash j}(\boldsymbol{\theta})} \quad (10.207)$$

- 计算模型证据的近似

$$p(\mathcal{D}) \simeq \int \prod_i \tilde{f}_i(\boldsymbol{\theta}) d\boldsymbol{\theta} \quad (10.208)$$

EP的一个特别的情况，被称为假定密度过滤（assumed density filtering, ADF）或者矩匹配（moment matching）（MayBeck, 1982; Lauritzen, 1992; Boyen and Koller, 1998; Opper and Winther, 1999），可以这样得到：对除了第一个因子以外的所有近似因子初始化为1，然后在所有因子之间进行一次迭代，每次更新因子中的每一个。假定密度过滤对于在线学习很适用，其中数据点顺序地到达，我们需要从每个数据点中进行学习，然后在考虑下一个数据点之间将其丢弃。然而，在批处理的设定中，我们有机会多次重新适用数据点来得到更高的精度，并且这正是期望传播所利用的思想。此外，如果我们将ADF应用于批量的数据，结果会依赖于数据点的处理顺序，这不是我们想要的，而EP可以克服这个缺点。

期望传播的一个缺点是，它不保证迭代会收敛。然而，对于指数族分布的近似 $q(\boldsymbol{\theta})$ ，如果迭代确实收敛，那么求得的解是特定的势函数的驻点（Minka, 2001a），虽然每轮EP迭代未必减小势函数的值。这与变分贝叶斯相反。变分贝叶斯中，每轮迭代保证不会减小界限。直接优化EP的代价函数是可能的，这种情况下，它保证收敛，虽然会导致算法更慢，实现起来更复杂。

变分贝叶斯和EP的另一个区别是来自于两个算法所最小化的KL散度的形式，因为前者最小化 $\text{KL}(q \parallel p)$ ，而后者最小化 $\text{KL}(p \parallel q)$ 。正如我们在图10.3中看到的那样，对于多峰的概率分布 $p(\boldsymbol{\theta})$ ，最小化 $\text{KL}(p \parallel q)$ 会产生较差的近似。特别地，如果将EP应用于混合概率分布，那么得到的结果没有意义，因为得到的近似试图覆盖后验概率分布的所有峰值。相反，在 logistic 类型的模型中，EP通常要比局部变分方法和拉普拉斯近似方法的表现更好（Kuss and Rasmussen, 2006）。

10.7.1 例子：聚类问题

遵从 Minka (2001b) 的做法，我们使用一个简单的例子来说明 EP 算法，其中我们的目标是在给定服从那个分布的一组观测的情况下，推断变量 x 上的多元高斯分布的均值 θ 。为了让问题

更加有趣，观测位于一个背景聚类中，它本身也是一个高斯分布，如图10.15所示。于是，观测值 \mathbf{x} 的概率分布是一个混合高斯分布，形式为

$$p(\mathbf{x} | \boldsymbol{\theta}) = (1 - w)\mathcal{N}(\mathbf{x} | \boldsymbol{\theta}, \mathbf{I}) + w\mathcal{N}(\mathbf{x} | \mathbf{0}, a\mathbf{I}) \quad (10.209)$$

其中， w 是背景聚类的比重，假设是已知的。 $\boldsymbol{\theta}$ 上的先验概率分布是高斯分布，形式为

$$p(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta} | \mathbf{0}, b\mathbf{I}) \quad (10.210)$$

Minka (2001a) 选择参数的值为 $a = 10, b = 100, w = 0.5$ 。 N 次观测 $\mathcal{D} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ 和 $\boldsymbol{\theta}$ 的联合概率分布为

$$p(\mathcal{D}, \boldsymbol{\theta}) = p(\boldsymbol{\theta}) \prod_{n=1}^N p(\mathbf{x}_n | \boldsymbol{\theta}) \quad (10.211)$$

因此后验概率分布由 2^N 个高斯分布混合而成。从而精确解决这个问题的计算代价会随着数据集的规模指数增长，因此对于大的 N 值，精确求解是不可行的。

为了将EP应用于聚类问题，我们首先看出，因子 $f_0(\boldsymbol{\theta}) = p(\boldsymbol{\theta})$ 且 $f_n(\boldsymbol{\theta}) = p(\mathbf{x}_n | \boldsymbol{\theta})$ 。接下来，我们从指数族分布中选择一个近似分布。对于这个例子，比较方便的做法是选择一个球形高斯分布

$$q(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta} | \mathbf{m}, v\mathbf{I}) \quad (10.212)$$

于是，因子近似会取指数-二次函数的形式，即

$$\tilde{f}_n(\boldsymbol{\theta}) = s_n \mathcal{N}(\boldsymbol{\theta} | \mathbf{m}_n, v_n \mathbf{I}) \quad (10.213)$$

其中 $n = 1, \dots, N$ ，并且我们令 $\tilde{f}_0(\boldsymbol{\theta})$ 等于先验概率分布 $p(\boldsymbol{\theta})$ 。注意，使用 $\mathcal{N}(\boldsymbol{\theta} | \cdot, \cdot)$ 不表示右侧是一个良好定义的高斯概率密度（事实上，正如我们将看到的那样，方差参数 v_n 可以为负），而是仅仅是一个方便的简化记号。近似 $\tilde{f}_n(\boldsymbol{\theta}), n = 1, \dots, N$ 可以被初始化为1，对应于 $s_n = (2\pi v_n)^{\frac{D}{2}}, v \rightarrow \infty$ 以及 $\mathbf{m}_n = \mathbf{0}$ ，其中 D 是 \mathbf{x} 的维度，因此也是 $\boldsymbol{\theta}$ 的维度。公式 (10.191) 定义的初始的 $q(\boldsymbol{\theta})$ 因此就等于先验概率分布。

我们接下来迭代地优化因子，方法是每次取一个因子 $f_n(\boldsymbol{\theta})$ ，然后使用公式 (10.205)、(10.206) 和 (10.207)。注意，我们不需要修改 $f_0(\boldsymbol{\theta})$ ，因为EP更新会让这一项保持不变。这里，我们给出结果，让读者自己来填充细节。

首先，我们从 $q(\boldsymbol{\theta})$ 中移除当前的估计 $\tilde{f}_n(\boldsymbol{\theta})$ ，方法是使用公式 (10.205) 做除法，得到 $q^{\setminus n}(\boldsymbol{\theta})$ ，它的均值和方差为

$$\mathbf{m}^{\setminus n} = \mathbf{m} + v^{\setminus n} v_n^{-1} (\mathbf{m} - \mathbf{m}_n) \quad (10.214)$$

$$(v^{\setminus n})^{-1} = v^{-1} - v_n^{-1} \quad (10.215)$$

接下来，我们使用公式 (10.206) 计算归一化常数，结果为

$$Z_n = (1 - w)\mathcal{N}(\mathbf{x}_n | \mathbf{m}^{\setminus n}, (v^{\setminus n} + 1)\mathbf{I}) + w\mathcal{N}(\mathbf{x}_n | \mathbf{0}, a\mathbf{I}) \quad (10.216)$$

类似地，我们通过寻找 $q^{\setminus n}(\boldsymbol{\theta}) f_n(\boldsymbol{\theta})$ 的均值，计算 $q^{\text{新}}(\boldsymbol{\theta})$ 的均值和方差，结果为

$$\mathbf{m}^{\text{新}} = \mathbf{m}^{\setminus n} + \rho_n \frac{v^{\setminus n}}{v^{\setminus n} + 1} (\mathbf{x}_n - \mathbf{m}^{\setminus n}) \quad (10.217)$$

$$v^{\text{新}} = v^{\setminus n} - \rho_n \frac{(v^{\setminus n})^2}{v^{\setminus n} + 1} + \rho_n (1 - \rho_n) \frac{(v^{\setminus n})^2 \|\mathbf{x}_n - \mathbf{m}^{\setminus n}\|^2}{D(v^{\setminus n} + 1)^2} \quad (10.218)$$

其中

$$\rho_n = 1 - \frac{w}{Z_n} \mathcal{N}(\mathbf{x}_n | \mathbf{0}, a\mathbf{I}) \quad (10.219)$$

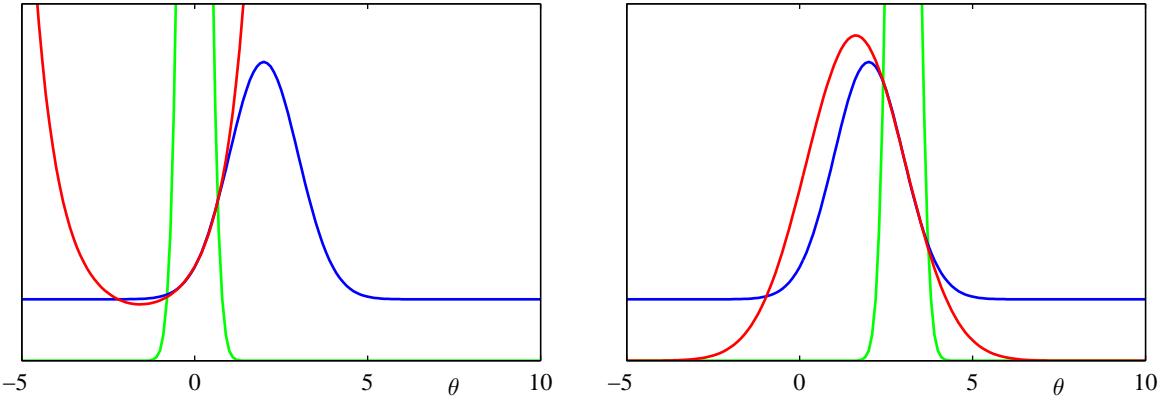


图 10.16: 对于聚类问题的一维版本, 具体因子的近似的例子。图中用蓝色表示 $f_n(\theta)$, 用红色表示 $\tilde{f}_n(\theta)$, 用绿色表示 $q^{(n)}(\theta)$ 。注意 $q^{(n)}(\theta)$ 现在的形式控制了 θ 的取值范围, 在这个范围上, $\tilde{f}(\theta)$ 是 $f_n(\theta)$ 的一个很好的近似。

它可以简单地表示为点 x_n 不在聚类中的概率。然后, 我们使用公式 (10.207) 计算优化因子 $\tilde{f}_n(\theta)$, 它的参数为

$$v_n^{-1} = (v^{\text{新}})^{-1} - (v^{(n)})^{-1} \quad (10.220)$$

$$\mathbf{m}_n = \mathbf{m}^{(n)} + (v_n + v^{(n)})(v^{(n)})^{-1}(\mathbf{m}^{\text{新}} - \mathbf{m}^{(n)}) \quad (10.221)$$

$$s_n = \frac{Z_n}{(2\pi v_n)^{\frac{D}{2}} \mathcal{N}(\mathbf{m}_n | \mathbf{m}^{(n)}, (v_n + v^{(n)})\mathbf{I})} \quad (10.222)$$

优化过程不断重复, 直到满足一个合适的终止准则, 例如在对所有因子进行的一次优化迭代中, 参数值的最大改变量小于一个阈值。最后, 我们使用公式 (10.208) 来计算模型证据的近似, 结果为

$$p(\mathcal{D}) \simeq (2\pi v^{\text{新}})^{\frac{D}{2}} \exp\left(\frac{D}{2}\right) \prod_{n=1}^N \left\{ s_n (2\pi v_n)^{-\frac{D}{2}} \right\} \quad (10.223)$$

其中

$$B = \frac{(\mathbf{m}^{\text{新}})^T \mathbf{m}^{\text{新}}}{v} - \sum_{n=1}^N \frac{\mathbf{m}_n^T \mathbf{m}_n}{v_n} \quad (10.224)$$

图10.16给出了对于一维参数空间 θ 的聚类问题的因子近似的例子。注意, 因子近似可以有无穷大的或者负数的“方差”参数 v_n 。这仅仅对应于曲线向上弯曲而不是向下弯曲的情形, 并且只要所有的近似后验概率 $q(\theta)$ 有正的方差, 这种情形就未必有问题。图10.17对比了在聚类问题中, EP的表现、变分贝叶斯 (平均场理论) 的表现以及拉普拉斯近似的表达。

10.7.2 图的期望传播

目前为止在我们对于EP的一般的讨论中, 我们让概率分布 $p(\theta)$ 中的所有因子 $f_i(\theta)$ 是 θ 的全部分量的函数, 类似地, 对于近似分布 $q(\theta)$ 的近似因子 $\tilde{f}(\theta)$ 的情形也相同。我们现在考虑下面的情形: 因子只依赖于变量的一个子集。这种限制可以很方便地使用第8章讨论的概率图模型的框架来表示。这里, 我们使用因子图表示, 因为它同时包含了有向图和无向图。

我们会把注意力集中于近似概率分布完全分解的情形, 我们会证明, 在这种情形下, 期望传播会简化为循环置信传播 (Minka, 2001a)。首先, 我们在一个简单的例子中证明这一点, 然后我们会研究一般的情形。

首先, 回忆一下, 根据公式 (10.17), 如果我们关于一个分解的概率分布 q 来最小化 Kullback-Leibler 散度 $\text{KL}(p \parallel q)$, 那么对于每个因子, 最优解为 p 的对应的边缘概率分布。

现在, 考虑图10.18左侧给出的因子图。我们之前在加和-乘积算法中介绍过这张图。联合概率分布为

$$p(\mathbf{x}) = f_a(x_1, x_2) f_b(x_2, x_3) f_c(x_2, x_4) \quad (10.225)$$

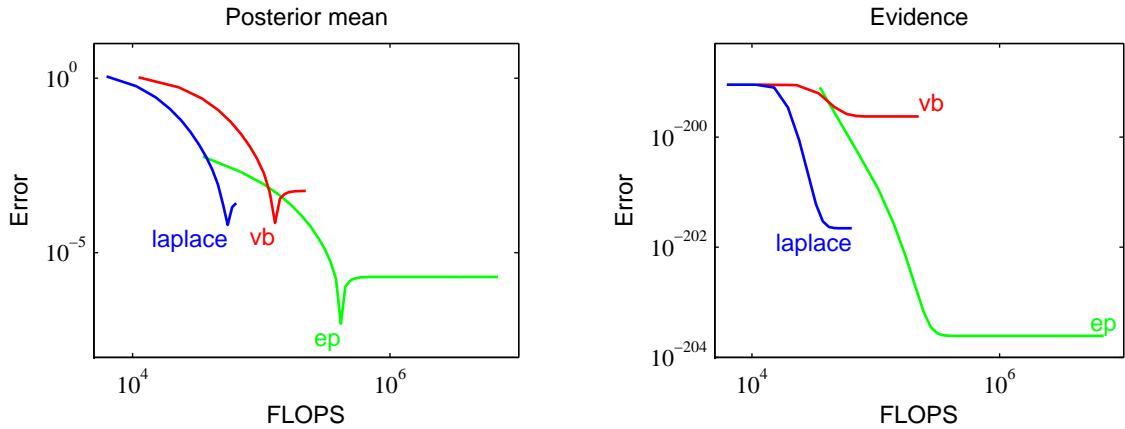


图 10.17: 期望传播、变分推断和拉普拉斯近似在聚类问题上的对比。左图给出了预测后验概率分布的均值与浮点运算的数量的关系，右图给出了对应的模型证据的结果。

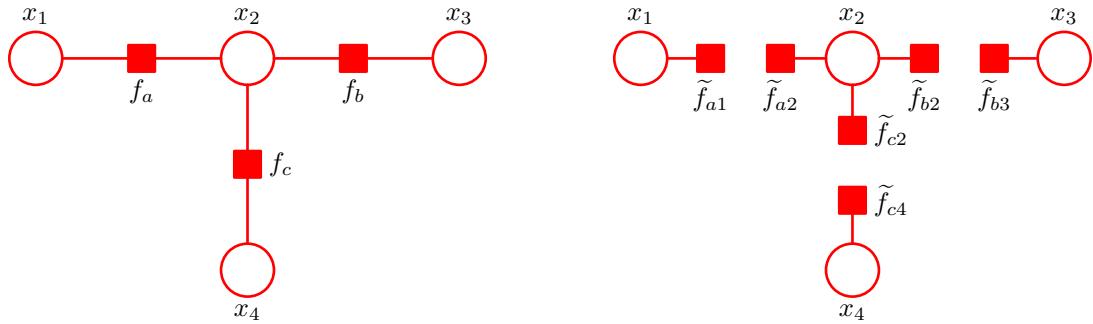


图 10.18: 左图是来自图8.51的一张简单的因子图，为了方便，这里重新画出。右图是对应的分解近似。

我们寻找具有相同分解方式的一个近似 $q(\mathbf{x})$ ，即

$$q(\mathbf{x}) \propto \tilde{f}_a(x_1, x_2) \tilde{f}_b(x_2, x_3) \tilde{f}_c(x_2, x_4) \quad (10.226)$$

注意，归一化常数被省略，这些可以在计算的最后使用局部归一化的方法计算出来，正如我们在置信传播中经常做的那样。现在，假设我们将注意力集中于近似分布上，其中因子本身可以关于各个变量进行分解，即

$$q(\mathbf{x}) \propto \tilde{f}_{a1}(x_1) \tilde{f}_{a2}(x_2) \tilde{f}_{b2}(x_2) \tilde{f}_{b3}(x_3) \tilde{f}_{c2}(x_2) \tilde{f}_{c4}(x_4) \quad (10.227)$$

它对应于图10.18右侧的因子图。由于各个独立的因子是分解的，因此整体概率分布 $q(\mathbf{x})$ 本身是完全分解的。

现在，我们使用这个完全分解的近似，应用EP算法。假设我们已经初始化了所有的因子，并且我们选择优化因子 $\tilde{f}_b(x_2, x_3) = \tilde{f}_{b2}(x_2) \tilde{f}_{b3}(x_3)$ 。首先，我们将这个因子从近似分布中移除，得到

$$q^b(\mathbf{x}) \propto \tilde{f}_{a1}(x_1) \tilde{f}_{a2}(x_2) \tilde{f}_{c2}(x_2) \tilde{f}_{c4}(x_4) \quad (10.228)$$

然后我们乘以精确因子 $f_b(x_2, x_3)$ ，可得

$$\hat{p}(\mathbf{x}) = q^b(\mathbf{x}) f_b(x_2, x_3) = \tilde{f}_{a1}(x_1) \tilde{f}_{a2}(x_2) \tilde{f}_{c2}(x_2) \tilde{f}_{c4}(x_4) f_b(x_2, x_3) \quad (10.229)$$

我们现在通过最小化Kullback-Leibler散度 $KL(\hat{p} \mid q^{\text{新}})$ 来寻找 $q^{\text{新}}(\mathbf{x})$ 。这个结果，正如之前注意到的那样，是 $q^{\text{新}}(z)$ 组成了因子的乘积，每个变量 x_i 对应一个因子，其中每个因子由 $\hat{p}(\mathbf{x})$ 的对应的边缘概率分布组成。这四个边缘概率分布为

$$\hat{p}(x_1) \propto \tilde{f}_{a1}(x_1) \quad (10.230)$$

$$\hat{p}(x_2) \propto \tilde{f}_{a1}(x_2) \tilde{f}_{c2}(x_2) \sum_{x_3} f_b(x_2, x_3) \quad (10.231)$$

$$\hat{p}(x_3) \propto \sum_{x_2} \left\{ f_b(x_2, x_3) \tilde{f}_{a2}(x_2) \tilde{f}_{c2}(x_2) \right\} \quad (10.232)$$

$$\hat{p}(x_4) \propto \tilde{f}_{c4}(x_4) \quad (10.233)$$

q^{new} 可以通过将这些边缘概率分布相乘的方式得到。我们看到，当我们更新 $\tilde{f}_b(x_2, x_3)$ 时， $q(\mathbf{x})$ 中唯一改变的因子是涉及到 f_b 中的变量的因子，即涉及到 x_2 和 x_3 的因子。为了得到优化的因子 $\tilde{f}_b(x_2, x_3) = \tilde{f}_{b2}(x_2) \tilde{f}_{b3}(x_3)$ ，我们将 $q^{\text{new}}(\mathbf{x})$ 除以 $q^{\setminus b}(\mathbf{x})$ ，结果为

$$\tilde{f}_{b2}(x_2) \propto \sum_{x_3} f_b(x_2, x_3) \quad (10.234)$$

$$\tilde{f}_{b3}(x_3) \propto \sum_{x_2} \left\{ f_b(x_2, x_3) \tilde{f}_{a2}(x_2) \tilde{f}_{c2}(x_2) \right\} \quad (10.235)$$

这些与使用置信传播得到的信息完全相同，其中从变量结点到因子结点的信息已经被整合到从因子结点到变量结点的信息当中。特别地， $\tilde{f}_{b2}(x_2)$ 对应于由因子结点 f_b 向变量结点 x_2 发送的信息 $\mu_{f_b \rightarrow x_2}(x_2)$ ，由公式 (8.81) 给出。类似地，如果我们将公式 (8.78) 代入公式 (8.79)，我们得到了公式 (10.235)，其中 $\tilde{f}_{a2}(x_2)$ 对应于 $\mu_{f_a \rightarrow x_2}(x_2)$ ，且 $\tilde{f}_{c2}(x_2)$ 对应于 $\mu_{f_c \rightarrow x_2}(x_2)$ ，给出了信息 $\tilde{f}_{b3}(x_3)$ ，它对应于 $\mu_{f_b \rightarrow x_3}(x_3)$ 。

这个结果与标准的置信传播稍微有些不同，因为信息同时向两个方向传递。我们可以很容易地修改EP步骤，给出加和-乘积算法的标准形式，修改方法为：每次只更新一个因子，例如如果我们只优化 $\tilde{f}_{b3}(x_3)$ ，那么根据定义， $\tilde{f}_{b2}(x_2)$ 不变，而 $\tilde{f}_{b3}(x_3)$ 的优化版本再次由公式 (10.235) 给出。如果我们每次只优化一项，那么我们可以选择我们所希望进行的优化的顺序。特别地，对于一个树结构的图，我们可以遵循两遍更新的框架，对应于标准的置信传播方法，它会产生对变量和因子的边缘概率分布的精确的推断。这种情况下，近似因子的初始化不再重要。

现在，让我们考虑一个一般的因子图，它对应于下面的概率分布

$$p(\boldsymbol{\theta}) = \prod_i f_i(\boldsymbol{\theta}_i) \quad (10.236)$$

其中 $\boldsymbol{\theta}_i$ 表示与因子 f_i 关联的变量的子集。我们使用一个完全分解的概率分布来近似它，形式为

$$q(\boldsymbol{\theta}) \propto \prod_i \prod_k \tilde{f}_{ik}(\boldsymbol{\theta}_k) \quad (10.237)$$

其中 $\boldsymbol{\theta}_k$ 对应于一个单独的变量结点。假设我们希望优化特定的项 $\tilde{f}_{jl}(\boldsymbol{\theta}_l)$ ，保持其他所有的项不变。首先，我们从 $q(\boldsymbol{\theta})$ 中移除项 $\tilde{f}_j(\boldsymbol{\theta}_j)$ ，可得

$$q^{\setminus j}(\boldsymbol{\theta}) \propto \prod_{i \neq j} \prod_k \tilde{f}_{ik}(\boldsymbol{\theta}_k) \quad (10.238)$$

然后乘以精确因子 $f_j(\boldsymbol{\theta}_j)$ 。为了确定优化项 $\tilde{f}_{jl}(\boldsymbol{\theta}_l)$ ，我们只需考虑对 $\boldsymbol{\theta}_l$ 的函数依赖，因此我们只需寻找

$$q^{\setminus j}(\boldsymbol{\theta}) f_j(\boldsymbol{\theta}_j) \quad (10.239)$$

对应的边缘概率分布。忽略一个可以做乘法的常数，这涉及到对 $f_j(\boldsymbol{\theta}_j)$ 与任意来自 $q^{\setminus j}(\boldsymbol{\theta})$ 中的属于 $\boldsymbol{\theta}_j$ 中任意变量的函数的项进行相乘得到的结果求边缘概率分布。当我们接下来除以 $q^{\setminus j}(\boldsymbol{\theta})$ 时，对应于 $i \neq j$ 的其他因子 $\tilde{f}_i(\boldsymbol{\theta}_i)$ 的项会在分子和分母之间消去。因此我们有

$$\tilde{f}_{jl}(\boldsymbol{\theta}_l) \propto \sum_{\boldsymbol{\theta}_{m \neq l} \in \boldsymbol{\theta}_j} f_j(\boldsymbol{\theta}_j) \prod_k \prod_{m \neq l} \tilde{f}_{km}(\boldsymbol{\theta}_m) \quad (10.240)$$

我们将这个式子看做是加和-乘积规则的形式，其中，从变量结点到因子结点的信息被消除，正如图 8.50 中给出的例子那样。 $\tilde{f}_{jm}(\boldsymbol{\theta}_m)$ 对应于信息 $\mu_{f_j \rightarrow \theta_m}(\boldsymbol{\theta}_m)$ ，其中因子结点 j 向变量结点 m 发

送信息，并且公式 (10.240) 中的在 k 上的乘积作用于所有依赖于与因子 $f_j(\theta_j)$ 有相同变量（除了变量 θ_l ）的变量 θ_m 。换句话说，为了计算来自一个因子结点的输出信息，我们对所有来自其他结点的输入信息求乘积，乘以局部因子，然后求和或积分。

因此，如果我们使用完全分解的近似概率分布，那么加和-乘积算法就可以作为期望传播的一个具体的例子。这表明，更加灵活的近似分布（对应于部分连接的图）可以得到更高的准确率。另一种推广是将因子 $f_i(\theta_i)$ 分成若干组，在一次迭代过程中优化组内的全部因子。这两种方法都可以产生精度的提升 (Minka, 2001b)。通常，选择最好的分组和断开连接的方式是一个开放的问题。

我们已经看到了变分信息传递和期望传播方法对 Kullback-Leibler 散度的两种不同的形式进行了最优化。Minka (2005) 证明，一大类信息传递方法可以从一个涉及到最小化散度的 alpha 家族的成员的通用框架中推导出来，其中，散度的 alpha 家族由公式 (10.19) 给出。这些信息传递方法包括变分信息传递、循环置信传播、期望传播，以及一大类其他的算法，例如树重加权信息传递 (tree-reweighted message passing) (Wainwright et al., 2005)、分数置信传播 (fractional belief propagation) (Wiegerinck and Heskes, 2003) 以及强 EP (power EP) (Minka, 2004)，篇幅所限，我们不会在这里介绍这些算法。

10.8 练习

(10.1) (*) 验证，观测数据的对数边缘分布 $\ln p(\mathbf{X})$ 可以被分解为公式 (10.2) 中的两项，其中 $\mathcal{L}(q)$ 由公式 (10.3) 给定， $\text{KL}(q \parallel p)$ 由公式 (10.4) 给定。

(10.2) (*) 使用性质 $\mathbb{E}[z_1] = m_1$ 和 $\mathbb{E}[z_2] = m_2$ 求解其次方程 (10.13) 和 (10.15)，证明，只要原始概率分布 $p(z)$ 非奇异，那么在近似分布中，因子的均值为 $\mathbb{E}[z_1] = \mu_1$ 和 $\mathbb{E}[z_2] = \mu_2$ 。

(10.3) (**) 考虑形如 (10.5) 的分解的变分分布 $q(\mathbf{Z})$ 。通过使用拉格朗日乘数法，验证 Kullback-Leibler 散度 $\text{KL}(q \parallel p)$ 关于一个因子 $q_i(Z_i)$ 的最小化（保持其他所有因子不变）会产生公式 (10.17) 给出的解。

(10.4) (**) 假设 $p(\mathbf{x})$ 是某个固定概率分布，我们希望使用一个高斯分布 $q(\mathbf{x}) = \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma})$ 来近似它。通过写出高斯 $q(\mathbf{x})$ 的情形下的 KL 散度 $\text{KL}(p \parallel q)$ ，然后求微分，证明， $\text{KL}(p \parallel q)$ 关于 $\boldsymbol{\mu}$ 和 $\boldsymbol{\Sigma}$ 的最小化会得到下面的结果： $\boldsymbol{\mu}$ 等于 \mathbf{x} 在 $p(\mathbf{x})$ 下的期望， $\boldsymbol{\Sigma}$ 等于协方差。

(10.5) (**) 考虑一个模型，其中所有隐含随机变量的集合（联合起来记作 \mathbf{Z} ）由某些潜在变量 \mathbf{z} 以及某些模型参数 $\boldsymbol{\theta}$ 组成。假设我们使用能够在潜在变量和参数之间分解的变分分布，即 $q(\mathbf{z}, \boldsymbol{\theta}) = q_{\mathbf{z}}(\mathbf{z})q_{\boldsymbol{\theta}}(\boldsymbol{\theta})$ ，其中概率分布 $q_{\boldsymbol{\theta}}(\boldsymbol{\theta})$ 使用形式为 $q_{\boldsymbol{\theta}}(\boldsymbol{\theta}) = \delta(\boldsymbol{\theta} - \boldsymbol{\theta}_0)$ 的点估计，其中 $\boldsymbol{\theta}_0$ 是自由参数的一个响亮。证明，这个分解的分布的变分最优化等价于 EM 算法，其中 E 步骤对 $q_{\mathbf{z}}(\mathbf{z})$ 进行最优化，M 步骤对 $\boldsymbol{\theta}$ 关于 $\boldsymbol{\theta}_0$ 的完整数据对数后验概率的期望进行最大化。

(10.6) (**) 散度的 alpha 家族由公式 (10.19) 定义。证明 Kullback-Leibler 散度 $\text{KL}(p \parallel q)$ 对应于 $\alpha \rightarrow 1$ 。证明方法为：写出 $p^{\epsilon} = \exp(\epsilon \ln p) = 1 + \epsilon \ln p + O(\epsilon^2)$ ，然后取 $\epsilon \rightarrow 0$ 。类似地，证明 $\text{KL}(q \parallel p)$ 对应于 $\alpha \rightarrow -1$ 。

(10.7) (**) 考虑使用 10.1.3 节讨论的分解的变分近似来推断一元高斯分布的均值和精度的问题。证明，因子 $q_{\mu}(\mu)$ 是一个高斯分布，形式为 $\mathcal{N}(\mu \mid \mu_N, \lambda_N^{-1})$ ，均值和方差分别为 (10.26) 和 (10.27)。类似地，证明因子 $q_{\gamma}(\gamma)$ 是一个 Gamma 分布，形式为 $\text{Gam}(\tau \mid a_N, b_N)$ ，参数由 (10.29) 和 (10.30) 给出。

(10.8) (*) 考虑一元高斯分布的精度的变分后验概率分布，它的参数由 (10.29) 和 (10.30) 给出。通过使用公式 (B.27) 和 (B.28) 给出的 Gamma 分布的均值和方差的标准结果，证明，如果我们令 $N \rightarrow \infty$ ，那么这个变分后验分布的均值为数据的方差的最大似然估计的倒数，方差趋于零。

(10.9) (**) 通过使用 Gamma 分布的均值的标准结果 $\mathbb{E}[\tau] = \frac{a_N}{b_N}$ ，以及 (10.26)、(10.27)、(10.29) 和 (10.30)，推导一元高斯分布的分解变分方法的期望精度的倒数的结果 (10.33)。

(10.10) (*) 推导变分推断方法中用于寻找模型上的近似后验概率分布的分解方式 (10.34)。

(10.11) (**) 通过使用拉格朗日乘数法来强制满足分布 $q(m)$ 上的归一化限制，证明下界 (10.35) 的下界的最大值为 (10.36)。

(10.12) (**) 从联合概率分布 (10.41) 开始, 应用一般的结果 (10.9), 证明, 潜在变量上的高斯分布的贝叶斯混合的最优变分分布 $q^*(\mathbf{Z})$ 是 (10.48)。证明方法是, 验证课本中给出的步骤。

(10.13) (**) 从公式 (10.54) 开始, 推导高斯分布的贝叶斯混合模型中, μ_k 和 Λ_k 上的最优变分后验概率分布, 从而验证由公式 (10.60) 和 (10.63) 给出的这个分布的参数的表达式。

(10.14) (**) 使用概率分布 (10.59), 验证 (10.64) 的结果。

(10.15) (*) 使用公式 (B.17) 的结果, 证明变分高斯混合中, 混合系数的期望值由 (10.69) 给出。

(10.16) (**) 验证公式 (10.70) 给出的变分高斯混合模型的下界的前两项的结果 (10.71) 和 (10.72)。

(10.17) (***) 验证公式 (10.70) 给出的变分高斯混合模型的下界的剩余各项的结果 (10.73) 到 (10.77)。

(10.18) (***) 本练习中, 我们通过直接对下界求微分, 推导高斯混合模型的变分重估计方程。为了完成这件事, 我们假设变分分布具有由 (10.42) 和 (10.55) 定义的分解方式, 因子由 (10.48)、(10.57) 和 (10.59) 定义。将这些代入 (10.70), 从而得到了下界关于变分分布的参数的函数。然后, 通过关于这些参数最大化下界, 推导变分分布中因子的重估计方程, 证明, 这些与 10.2.1 节得到的相同。

(10.19) (**) 推导高斯模型的贝叶斯混合的变分方法的预测分布的结果 (10.81)。

(10.20) (**) 本练习研究当数据集的大小 N 较大时, 高斯混合模型的变分贝叶斯解, 证明它简化为第 9 章基于 EM 算法推导的最大似然解 (与我们的预期相符)。注意, 附录 B 中的结果可以用来帮助解决本练习。首先, 证明精度的后验概率分布 $q^*(\Lambda_k)$ 在最大似然解周围具有尖峰。为均值的后验概率分布 $q^*(\mu_k | \Lambda_k)$ 做同样的事情。接下来, 考虑混合系数的后验概率分布 $q^*(\pi)$, 证明它在最大似然解周围也具有尖峰。类似地, 证明对于大的 N 值, “责任”等于对应的最大似然值。证明方法是使用下面的对于大的 x 的 Digamma 函数的渐近结果。

$$\psi(x) = \ln x + O\left(\frac{1}{x}\right) \quad (10.241)$$

最后, 通过使用公式 (10.80), 证明对于大的 N , 预测分布是一个高斯混合分布。

(10.21) (*) 证明, 在具有 K 个分量的混合模型中, 由于交换对称性而产生的等价的参数设置的数量为 $K!$ 。

(10.22) (**) 我们已经看到, 高斯混合模型的后验概率分布的每个峰值都是 $K!$ 个等价的峰值中的一个。假设运行变分推断算法的结果是一个近似的后验概率分布 q , 它位于这些峰值中的一个峰值的邻域内。然后, 我们可以将完整的后验概率分布近似为 $K!$ 个这样的 q 分布的混合, 每个分布以每个峰值为中心, 具有相等的混合系数。证明, 如果我们假设 q 混合分布的分量之间的重叠可以忽略, 那么得到的下界与通过添加额外的 $\ln K!$ 项得到的单一分量 q 的下界不同。

(10.23) (**) 考虑一个变分高斯模型, 其中混合系数 $\{\pi_k\}$ 上没有先验分布。相反, 混合系数被当成参数, 它的值要通过最大化对数边缘似然函数的变分下界的方式求出。证明, 关于混合系数最大化这个下界, 使用拉格朗日乘数法强制满足混合系数加和为 1 的限制, 会得到重估计结果 (10.83)。注意, 不需要考虑下界中的所有项, 只需考虑下界与 $\{\pi_k\}$ 的依赖关系即可。

(10.24) (**) 我们已经在 10.2 节看到, 高斯混合模型的最大似然方法产生的奇异性不会出现在贝叶斯方法中。讨论, 如果贝叶斯模型使用最大后验 (MAP) 估计求解, 是否会出现这种奇异性。

(10.25) (**) 10.2 节讨论的高斯分布的贝叶斯混合的变分方法使用了对后验概率分布的一个分解的近似 (10.5)。正如我们在 10.2 节看到的那样, 分解假设使得参数空间的某个特定的方向上的后验概率分布的方差被低估。定性讨论这一点对于模型证据的变分近似产生的效果, 以及这个效果随着混合分量的数量如何变化。解释变分高斯混合倾向于低估最优分量数量还是高估最优分量数量。

(10.26) (***) 将贝叶斯线性回归的变分方法推广, 使其包含 β 上的 Gamma 超先验 $\text{Gam}(\beta | c_0, d_0)$, 通过假设形式为 $q(\mathbf{w})q(\alpha)q(\beta)$ 的可分解的变分概率分布, 变分地求解。推导变分分布中三个因子的变分更新方程, 并且求出预测分布下界的一个表达式。

(10.27) (**) 通过使用附录 B 中给定的公式, 证明线性基函数回归模型的下界可以写成 (10.107) 的形式, 各个参数由 (10.108) 到 (10.112) 定义。

(10.28) (***) 将10.2节介绍的高斯分布的贝叶斯混合的模型重写为指数族分布的一个共轭模型，就像10.4节讨论的那样。从而，使用一般的结果(10.115)和(10.119)推导具体的结果(10.48)、(10.57)和(10.59)。

(10.29) (*) 通过计算二阶导数，证明函数 $f(x) = \ln(x)$ 对于 $0 < x < \infty$ 是凹函数。确定由公式(10.133)定义的对偶函数 $g(\eta)$ 的形式，验证根据(10.132)关于 η 对 $\eta x - g(\eta)$ 进行最小化确实恢复出了函数 $\ln(x)$ 。

(10.30) (*) 通过计算二阶导数，证明对数logistic函数 $f(x) = -\ln(1 + e^{-x})$ 是凹函数。直接使用对数logistic函数在点 $x = \xi$ 附近的一阶泰勒展开式推导变分上界(10.137)。

(10.31) (**) 通过寻找关于 x 的二阶导数，证明函数 $f(x) = -\ln(e^{x/2} + e^{-x/2})$ 是 x 的一个凹函数。现在考虑关于变量 x^2 的二阶导数，从而证明，它是 x^2 的凸函数。画出 $f(x)$ 关于 x 和 x^2 的图像。直接使用关于 x^2 的函数 $f(x)$ 在以 ξ^2 为中心的一阶泰勒展开式，推导logistic sigmoid函数的下界(10.144)。

(10.32) (**) 考虑顺序学习的logistic回归的变分方法，其中每次处理一个数据点，每个数据点必须在下一个数据点到达之前处理并且丢弃。证明，后验概率分布的高斯近似可以通过使用下界(10.151)来维护，其中概率分布使用先验分布来初始化，并且当每个数据点被整合到模型中之后，对应的变分参数 ξ_n 被最优化。

(10.33) (*) 通过将(10.161)定义的 $Q(\xi, \xi^{\text{旧}})$ 关于变分参数 ξ_n 求微分，证明贝叶斯logistic回归模型的 ξ_n 的更新方程由(10.163)给定。

(10.34) (**) 本练习中，我们通过直接对(10.164)给出的下界进行最大化，推导4.5节讨论的贝叶斯logistic回归模型的变分参数 ξ 的重估计方程。为了完成这一点，令 $\mathcal{L}(\xi)$ 关于 ξ_n 的导数等于零，使用行列式的对数的导数的结果(3.117)，以及定义了变分后验概率分布 $q(\mathbf{w})$ 的均值和方差的表达式(10.157)和(10.158)。

(10.35) (**) 推导变分logistic回归模型的下界 $\mathcal{L}(\xi)$ 的结果(10.164)。这很容易完成，方法是讲高斯先验 $q(\mathbf{w}) = \mathcal{N}(\mathbf{w} | \mathbf{m}_0, \mathbf{S}_0)$ 以及似然函数的下界 $h(\mathbf{w}, \xi)$ 的表达式代入定义了 $\mathcal{L}(\xi)$ 的公式(10.159)的积分中。接下来，将指数项中依赖于 \mathbf{w} 的项聚集在一起，配平方，得到高斯积分，然后可以通过使用多元高斯分布的归一化系数的标准结果来计算。最后，取对数，得到(10.164)。

(10.36) (**) 考虑10.7节讨论的ADF近似方法，证明，因子 $f_j(\boldsymbol{\theta})$ 的引入产生了下面形式的模型证据更新

$$p_j(\mathcal{D}) \simeq p_{j-1}(\mathcal{D}) Z_j \quad (10.242)$$

其中 Z_j 是公式(10.197)定义的归一化常数。通过递归地使用这个结果，用 $p_0(\mathcal{D}) = 1$ 进行初始化，推导下面的结果

$$p(\mathcal{D}) \simeq \prod_j Z_j \quad (10.243)$$

(10.37) (*) 考虑10.7节的期望传播算法，假设定义(10.188)中的一个因子 $f_0(\boldsymbol{\theta})$ 与近似分布 $q(\boldsymbol{\theta})$ 具有相同的指数族分布函数形式。证明，如果因子 $\tilde{f}_0(\boldsymbol{\theta})$ 被初始化为 $f_0(\boldsymbol{\theta})$ ，那么优化 $\tilde{f}_0(\boldsymbol{\theta})$ 的EP更新会保持 $\tilde{f}_0(\boldsymbol{\theta})$ 不变。这个情况通常出现在一个因子是先验概率 $p(\boldsymbol{\theta})$ 的时候。因此我们看到先验因子可以一次精确地被整合，无需优化。

(10.38) (***) 本练习和下个练习中，我们验证期望传播算法应用于聚类问题的结果(10.214)到(10.224)。首先，使用对指数项配平方的方法分离出均值和方差的方式，通过使用除法公式(10.205)，推导出表达式(10.214)和(10.215)。此外，证明对于聚类问题，由公式(10.206)定义的归一化常数 Z_n 由公式(10.216)给出。使用一般的结果(2.115)即可完成。

(10.39) (***) 证明，应用于聚类问题的EP的 $q^{\text{新}}(\boldsymbol{\theta})$ 的均值和方差为(10.217)和(10.218)。为了完成这件事，首先证明下面的在 $q^{\text{新}}(\boldsymbol{\theta})$ 下， $\boldsymbol{\theta}$ 和 $\boldsymbol{\theta}\boldsymbol{\theta}^T$ 的期望的结果。

$$\mathbb{E}[\boldsymbol{\theta}] = \mathbf{m}^{\backslash n} + v^{\backslash n} \nabla_{\mathbf{m}^{\backslash n}} \ln Z_n \quad (10.244)$$

$$\mathbb{E}[\boldsymbol{\theta}^T \boldsymbol{\theta}] = 2(v^{\backslash n})^2 \nabla_{v^{\backslash n}} \ln Z_n + 2\mathbb{E}[\boldsymbol{\theta}]^T \mathbf{m}^{\backslash n} - \|\mathbf{m}^{\backslash n}\|^2 + v^{\backslash n} D \quad (10.245)$$

然后使用公式(10.216)给出的 Z_n 的结果。接下来，通过使用(10.207)然后对指数项配平方的方法，证明结果(10.222)和(10.207)。最后，使用(10.208)推导结果(10.223)。

11 采样方法

对于大多数实际应用中的概率模型来说，精确推断是不可行的，因此我们不得不借助与某种形式的近似。在第10章中，我们讨论了基于确定性近似的推断方法，它包括诸如变分贝叶斯方法以及期望传播。这里，我们考虑基于数值采样的近似推断方法，也被称为蒙特卡罗（Monte Carlo）方法。

虽然对于一些应用来说，我们感兴趣的是非观测变量上的后验概率分布本身，但是在大部分情况下，后验概率分布的主要用途是计算期望，例如在做预测的情形下就是这样。因此，本章中，我们希望解决的基本的问题涉及到关于一个概率分布 $p(z)$ 寻找某个函数 $f(z)$ 的期望。这里， z 的元素可能是离散变量、连续变量或者二者的组合。因此，在连续变量的情形下，我们希望计算下面的期望

$$\mathbb{E}[f] = \int f(z)p(z) dz \quad (11.1)$$

在离散变量的情形下，积分被替换为求和。图11.1图形化地说明了单一连续变量的情形。我们假设，使用解析的方法精确地求出这种期望是十分复杂的。

采样方法背后的一般思想是得到从概率分布 $p(z)$ 中独立抽取的一组变量 $z^{(l)}$ ，其中 $l = 1, \dots, L$ 。这使得期望可以通过有限和的方式计算，即

$$\hat{f} = \frac{1}{L} \sum_{l=1}^L f(z^{(l)}) \quad (11.2)$$

只要样本 $z^{(l)}$ 是从概率分布 $p(z)$ 中抽取的，那么 $\mathbb{E}[\hat{f}] = \mathbb{E}[f]$ ，因此估计 \hat{f} 具有正确的均值。估计 f 的方差为

$$\text{var}[\hat{f}] = \frac{1}{L} \mathbb{E}[(f - \mathbb{E}[f])^2] \quad (11.3)$$

它是函数 $f(z)$ 在概率分布 $p(z)$ 下的方差。因此，值得强调的一点是，估计的精度不依赖于 z 的维度，并且原则上，对于数量相对较少的样本 $z^{(l)}$ ，可能会达到较高的精度。在实际应用中，10个或者20个独立的样本就能够以足够高的精度对期望做出估计。

然而，问题在于样本 $\{z^{(l)}\}$ 可能不是独立的，因此有效样本大小可能远远小于表面上的样本大小。并且，回到图11.1，我们注意到如果 $f(z)$ 在 $p(z)$ 较大的区域中的值较小，反之亦然，那么期望可能由小概率的区域控制，表明为了达到足够的精度，需要相对较大的样本大小。

对于许多模型来说，联合概率分布 $p(z)$ 可以使用图模型很容易地确定。在没有观测变量的有向图的情形，从联合概率分布中采样是很容易的（假设可以从每个节点处的条件概率分布中采样），方法是使用8.1.2节简短讨论过的祖先采样方法（ancestral sampling approach）。联合概率分布为

$$p(z) = \prod_{i=1}^M p(z_i | \text{pa}_i) \quad (11.4)$$

其中， z_i 是与结点*i*关联的一组变量， pa_i 表示与结点*i*的父结点关联的变量的集合。为了从联合概率分布中得到一个样本，我们按照 z_1, \dots, z_M 的顺序遍历一次变量集合，这些变量是从条件

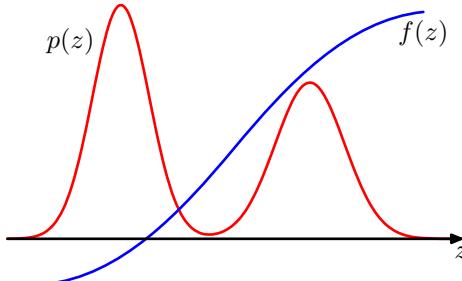


图 11.1: 函数 $f(z)$ 的期望的图形化表示， $f(z)$ 的期望是关于概率分布 $p(z)$ 计算得到的。

概率分布 $p(z_i | \text{pa}_i)$ 中抽取的。这总是可行的，因为在每一步，所有的父结点的值都已经被初始化。在对图遍历一次之后，我们会得到来自联合概率分布的一个样本。

现在，考虑某些结点被观测值进行初始化的有向图的情形。原则上，我们可以推广上述的步骤，至少在结点表示离散变量的情形下我们可以这样做。对上述步骤的推广给出了逻辑采样 (logic sampling) 的方法 (Henrion, 1988)，它可以被看做 11.1.4 节讨论的重要采样 (importance sampling) 的一种特殊情况。在每一个步骤中，当我们得到了变量 z_i 的一个采样值，它的值被观测，并且将采样值与观测值进行比较。如果它们相符，那么采样值被保留，算法继续运行，处理下一个变量。然而如果采样值与观测值不相符，那么目前为止得到的采样被丢弃，算法从图中的第一个结点重新开始。算法可以从后验概率分布中正确地采样，因为它对应于从隐含变量和数据变量的联合概率分布中采样然后丢弃那些与观测数据不相符的样本（稍微保留了一些从联合概率分布中采样的不连续性，只要观测到矛盾的值）。然而，接受一个来自后验概率分布的样本的整体概率会随着观测变量的数量的增加以及变量可以取得的状态数量的增加而迅速减小，因此这种方法在实际中很少被使用。

在由无向图定义的概率分布的情形中，如果我们希望从没有观测变量的先验概率分布中采样，那么不存在一遍采样的方法。相反，我们必须使用计算量更大的方法，例如 11.3 节讨论的吉布斯采样。

除了从条件概率分布中采样之外，我们可能也需要从边缘概率分布中采样。如果我们已经有了一种从联合概率分布 $p(x, v)$ 中采样的方法，那么得到从边缘概率分布 $p(u)$ 中的样本是很容易的，只需忽略每个样本中的 v 的值即可。

有许多讨论蒙特卡罗方法的文献。从统计推断的角度进行研究的文献包括 Chen et al. (2001)、Gamerman (1997)、Liu (2001)、Neal (1996) 和 Robert and Casella (1999)。并且有一些综述性的文章为统计推断的采样方法提供了额外的信息，例如 Besag et al. (2005)、Brooks (1998)、Diaconis and Saloff-Coste (1998)、Jerrum and Sinclair (1996)、Neal (1993)、Tierney (1994) 和 Andrieu et al. (2003)。

Robert and Casella (1999) 总结了马尔科夫链蒙特卡罗算法的收敛性检测。

11.1 基本采样算法

本节中，我们研究从一个给定的概率分布中生成随机样本的一些简单的方法。由于样本是通过计算机算法生成的，因此这些样本实际上是伪随机数 (pseudo-random numbers)，也就是说，它们通过计算的方法确定，但是仍然会通过随机性的检测。生成这种数字会产生一些微妙的性质 (Press et al., 1992)，不在本书的讨论范围内。这里，我们假定算法生成的是 $(0, 1)$ 之间均匀分布的伪随机数，事实上大部分软件开发环境都有这种功能。

11.1.1 标准概率分布

授信，我们考虑如何从简单的非均匀分布中生成随机数，假定我们已经有了一个均匀分布的随机数的来源。假设 z 在区间 $(0, 1)$ 上均匀分布，我们使用某个函数 $f(\cdot)$ 对 z 的值进行变换，即 $y = f(z)$ 。 y 上的概率分布为

$$p(y) = p(z) \left| \frac{dz}{dy} \right| \quad (11.5)$$

其中，在这种情况下， $p(z) = 1$ 。我们的目标是选择一个函数 $f(z)$ 使得产生出的 y 值具有某种所需的具体的分布形式 $p(y)$ ，对公式 (11.5) 进行积分，我们有

$$z = h(y) \equiv \int_{-\infty}^y p(\hat{y}) d\hat{y} \quad (11.6)$$

它是 $p(y)$ 的不定积分。因此， $y = h^{-1}(z)$ ，因此我们必须使用一个函数来对这个均匀分布的随机数进行变换，这个函数是所求的概率分布的不定积分的反函数，如图 11.2 所示。

考虑指数分布 (exponential distribution)

$$p(y) = \lambda \exp(-\lambda y) \quad (11.7)$$

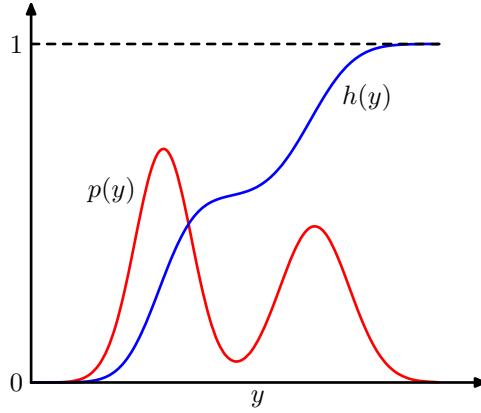


图 11.2: 生成非均匀分布的随机数的变换方法的几何表示。 $h(y)$ 是所求概率分布 $p(y)$ 的不定积分。如果一个均匀分布的随机变量 z 使用 $y = h^{-1}(z)$ 进行变换，那么 y 会服从概率分布 $p(y)$ 。

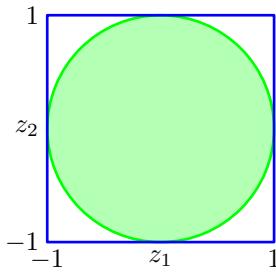


图 11.3: Box-Muller方法用于生成高斯分布的随机数，方法在开始时使用的是单位圆内部均匀分布的样本。

其中 $0 \leq y < \infty$ 。在这种情况下，公式 (11.6) 的积分下界为0，因此 $h(y) = 1 - \exp(-\lambda y)$ 。从而，如果我们将均匀分布的变量 z 使用 $y = -\lambda^{-1} \ln(1 - z)$ 进行变换，那么 y 就会服从指数分布。

另一种可以应用变换方法的概率分布是柯西分布

$$p(y) = \frac{1}{\pi} \frac{1}{1 + y^2} \quad (11.8)$$

这种情况下，不定积分的反函数可以用 \tan 函数表示。

对于多个变量情形的推广是很容易的，涉及到变量变化的Jacobian行列式，即

$$p(y_1, \dots, y_M) = p(z_1, \dots, z_M) \left| \frac{\partial(z_1, \dots, z_M)}{\partial(y_1, \dots, y_M)} \right| \quad (11.9)$$

作为变换方法的最后一个例子，我们考虑Box-Muller方法，用于生成高斯概率分布的样本。首先，假设我们生成一对均匀分布的随机变量 $z_1, z_2 \in (-1, 1)$ ，我们可以这样生成：对 $(0, 1)$ 上的均匀分布的变量使用 $z \rightarrow 2z - 1$ 的方式进行变换。接下来，我们丢弃那些不满足 $z_1^2 + z_2^2 \leq 1$ 的点对。这产生出单位圆内部的一个均匀分布，且 $p(z_1, z_2) = \frac{1}{\pi}$ ，如图11.3所示。然后，对于每对 z_1, z_2 ，我们计算

$$y_1 = z_1 \left(\frac{-2 \ln r^2}{r^2} \right)^{\frac{1}{2}} \quad (11.10)$$

$$y_2 = z_2 \left(\frac{-2 \ln r^2}{r^2} \right)^{\frac{1}{2}} \quad (11.11)$$

其中 $r^2 = z_1^2 + z_2^2$ 。这样， y_1 和 y_2 的联合概率分布为

$$\begin{aligned} p(y_1, y_2) &= p(z_1, z_2) \left| \frac{\partial(z_1, z_2)}{\partial(y_1, y_2)} \right| \\ &= \left[\frac{1}{\sqrt{2\pi}} \exp\left(\frac{-y_1^2}{2}\right) \right] \left[\frac{1}{\sqrt{2\pi}} \exp\left(\frac{-y_2^2}{2}\right) \right] \end{aligned} \quad (11.12)$$

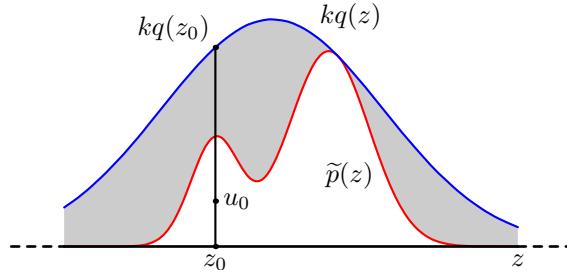


图 11.4: 在拒绝采样的方法中, 样本从一个简单的概率分布 $q(z)$ 中抽取。如果样本落到了未归一化的概率分布 $\tilde{p}(z)$ 与放缩的概率分布 $kq(z)$ 之间的灰色区域, 那么样本会被拒绝。得到的样本服从 $p(z)$ 的分布, 它是 $\tilde{p}(z)$ 的归一化版本。

因此 y_1 和 y_2 是独立的, 且每个都服从高斯分布, 均值为零, 方差为1。

如果 y 服从高斯分布, 且均值为零, 方差为1, 那么 $\sigma y + \mu$ 也服从高斯分布, 均值为 μ , 方差为 σ^2 。为了生成向量值的变量, 且这个变量服从多元高斯分布, 均值为 μ , 协方差为 Σ , 我们可以使用 Cholesky 分解, 它的形式为 $\Sigma = LL^T$ (Press et al., 1992)。这样, 如果 z 是一个向量值的随机变量, 且它的元素是独立的, 并且服从均值为零、方差为1的高斯分布, 那么 $y = \mu + Lz$ 的均值为 μ , 协方差为 Σ 。

显然, 变换方法依赖于它能够进行计算所需的概率分布, 并且能够求所需概率分布的不定积分的反函数。这样的计算只对于一些非常有限的简单的概率分布可行, 因此我们必须寻找一些更加一般的方法。这里, 我们考虑两种方法, 即拒绝采样 (rejection sampling) 和重要采样 (importance sampling)。虽然这些方法主要限制在单变量概率分布, 因此无法直接应用于多维的复杂问题, 但是这些方法确实是更一般的方法的重要成分。

11.1.2 拒绝采样

拒绝采样框架使得我们能够在满足某些限制条件的情况下, 从相对复杂的概率分布中采样。首先, 我们考虑单变量分布, 然后接下来讨论对于多维情形的推广。

假设我们希望从概率分布 $p(z)$ 中采样, 这个概率分布不是我们目前为止讨论过的简单的标准的概率分布中的一个, 从而直接从 $p(z)$ 中采样是很困难的。此外, 正如经常出现的情形那样, 我们假设我们能够很容易地计算对于任意给定的 z 值的 $p(z)$ (不考虑归一化常数 Z), 即

$$p(z) = \frac{1}{Z_p} \tilde{p}(z) \quad (11.13)$$

其中 $\tilde{p}(z)$ 可以很容易地计算, 但是 Z_p 未知。

为了应用拒绝采样方法, 我们需要一些简单的概率分布 $q(z)$, 有时被称为提议分布 (proposal distribution), 并且我们已经可以从提议分布中进行采样。接下来, 我们引入一个常数 k , 它的选择满足下面的性质: 对所有的 z 值, 都有 $kq(z) \geq \tilde{p}(z)$ 。函数 $kq(z)$ 被称为比较函数, 并且图 11.4 给出了单变量概率分布的说明。拒绝采样器的每个步骤涉及到生成两个随机数。首先, 我们从概率分布 $q(z)$ 中生成一个数 z_0 。接下来, 我们在区间 $[0, kq(z_0)]$ 上的均匀分布中生成一个数 u_0 。这对随机数在函数 $kq(z)$ 的曲线下方是均匀分布。最后, 如果 $u_0 > \tilde{p}(z_0)$, 那么样本被拒绝, 否则 u_0 被保留。因此, 如果它位于图 11.4 的灰色阴影部分, 它就会被拒绝。这样, 剩余的点对在曲线 $\tilde{p}(z)$ 下方是均匀分布的, 因此对应的 z 值服从概率分布 $p(z)$, 正如我们所需的那样。

z 的原始值从概率分布 $q(z)$ 中生成, 这些样本之后被接受的概率为 $\frac{\tilde{p}(z)}{kq(z)}$, 因此一个样本会被接受的概率为

$$\begin{aligned} p(\text{接受}) &= \int \left\{ \frac{\tilde{p}(z)}{kq(z)} \right\} q(z) dz \\ &= \frac{1}{k} \int \tilde{p}(z) dz \end{aligned} \quad (11.14)$$

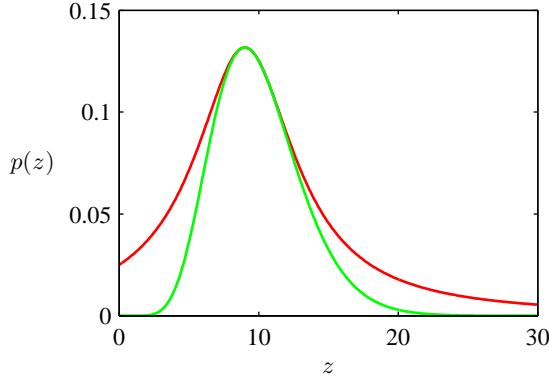


图 11.5: 绿色曲线表示公式 (11.15) 给出的Gamma分布的图像，红色曲线表示放缩后的柯西提议分布。从Gamma分布中抽取的样本可以通过从柯西分布中采样然后使用拒绝采样准则的方法得到。

因此，被这种方法拒绝的点的比例依赖于曲线 $kq(z)$ 下方的未归一化概率分布 $\tilde{p}(z)$ 的面积的比例。于是，我们看到，常数 k 应该尽量小，同时满足下面的限制条件： $kq(z)$ 一定处处不小于 $\tilde{p}(z)$ 。

作为拒绝采样的一个例子，我们考虑从Gamma分布中采样的任务，Gamma分布的形式为

$$\text{Gam}(z | a, b) = \frac{b^a z^{a-1} \exp(-bz)}{\Gamma(a)} \quad (11.15)$$

对于 $a > 1$ 的情形，它的形状是钟形曲线，如图11.5所示。于是，一个合适的提议分布为柯西分布 (11.8)，因为这个分布也是一个钟形曲线，并且因为我们可以使用之前讨论的变换方法从这个分布中进行采样。我们需要对柯西分布稍稍进行推广，来确保它处处的值都不小于Gamma分布。可以这样做：对一个均匀分布的变量 y ，使用 $z = b \tan y + c$ 进行变换，它给出了服从下面概率分布的随机数

$$q(z) = \frac{k}{1 + \frac{(z-c)^2}{b^2}} \quad (11.16)$$

最小的拒绝率在下面的条件下得到：令 $c = a - 1$, $b^2 = 2a - 1$ ，并且将常数 k 选得尽可能小，同时满足 $kq(z) \geq \tilde{p}(z)$ 的要求。函数的对比也在图11.5中给出。

11.1.3 可调节的拒绝采样

在许多我们希望应用拒绝采样的情形中，确定概率分布 $q(z)$ 的一个合适的解析形式是很困难的。另一种确定其函数形式的方法是基于概率分布 $p(z)$ 的值直接构建函数形式 (Gilks and Wild, 1992)。对于 $p(z)$ 是对数凹函数的情形，即 $\ln p(z)$ 的导数是 z 的单调非增函数时，界限函数的构建是相当简单的。图11.6给出了一个合适的界限函数的构建的例子。

函数 $\ln p(z)$ 和它的切线在某些初始的格点处进行计算，生成的切线的交点被用于构建界限函数。接下来，我们从界限分布中抽取一个样本值。这很容易，因为界限函数的对数是一系列的线性函数，因此界限函数本身由一个分段指数分布组成，形式为

$$q(z) = k_i \lambda_i \exp\{-\lambda_i(z - z_i)\} \quad \hat{z}_{i-1,i} < z \leq \hat{z}_{i,i+1} \quad (11.17)$$

其中 $\hat{z}_{i-1,i}$ 是在点 z_{i-1} 和 z_i 处的切线的交点， λ_i 是切线在 z_i 处的斜率， k_i 表示对应的偏移量。一旦一个样本点被抽取完毕，我们就可以应用通常的拒绝准则了。如果样本被接受，那么它就是所求的概率分布中的一个样本。然而，如果样本被拒绝，那么它被并入格点的集合中，计算出一条新的切线，从而界限函数被优化。随着格点数量的增加，界限函数对所求的概率分布的近似效果逐渐变好，拒绝的概率就会减小。

这个算法存在一种变体，这种变体中不用计算导数 (Gilks, 1992)。可调节的拒绝采样的框架也可以扩展到不是对数凹函数的概率分布中，只需将每个拒绝采样的步骤中使

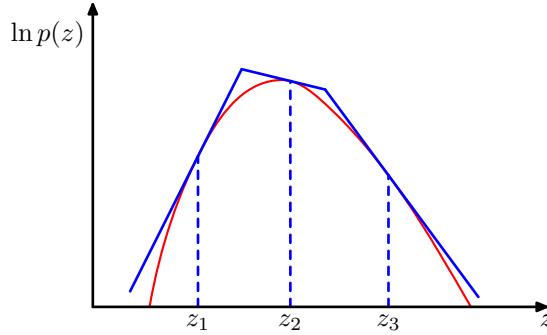


图 11.6: 在对数凹函数的情形下, 拒绝采样中用到的界限函数可以使用在一组格点处计算的切线来构造。如果一个样本点被拒绝, 那么它被添加到格点集合中, 被用于优化界限函数。

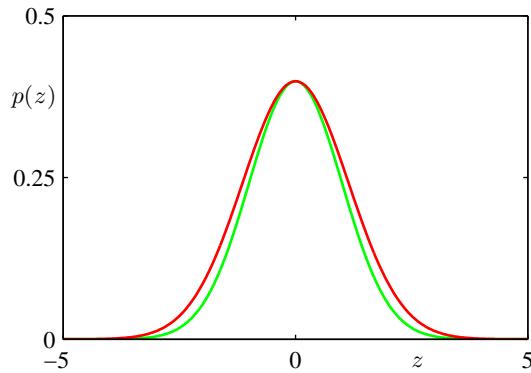


图 11.7: 从高斯分布 $p(z)$ (绿色曲线) 中进行拒绝采样的例子, 提议分布 $q(z)$ 也是一个高斯分布, 它的缩放版本 $kq(z)$ 用红色曲线表示。

用Metropolis-Hastings阶梯函数 (将在11.2.2节讨论) 即可, 这就产生了可调节拒绝Metropolis采样 (adaptive rejection Metropolis sampling) 方法 (Gilks et al., 1995)。

显然, 对于具有实际价值的拒绝采样来说, 我们要求对比函数要接近所求的概率分布, 从而拒绝率要保持一个最小值。现在, 让我们考察当我们试图在高维空间中使用拒绝采样的方法会出现什么现象。为了说明的方便, 考虑一个多少有些人造感觉的问题, 其中我们想要从一个零均值多元高斯分布中采样, 这个高斯分布的协方差为 $\sigma_p^2 \mathbf{I}$, 其中 \mathbf{I} 是单位矩阵。根据拒绝采样方法, 提议分布本身就是一个零均值的高斯分布, 协方差为 $\sigma_q^2 \mathbf{I}$ 。很明显, 为了使得 $kq(z) \geq p(z)$ 的 k 值存在, 我们必须有 $\sigma_q^2 \geq \sigma_p^2$ 。在 D 维的情形中, k 的值为 $k = \left(\frac{\sigma_q}{\sigma_p}\right)^D$, 图11.7给出了 $D = 1$ 的情形。接受率是 $p(z)$ 和 $kq(z)$ 下方的体积的比值。由于分布是归一化的, 这个比值就是 $\frac{1}{k}$ 。因此, 接受率随着维度的增大而指数地减小。即使 σ_q 只比 σ_p 高一个百分点, 对于 $D = 1000$, 接受率大约为 $\frac{1}{20,000}$ 。在这个说明的例子中, 对比函数接近于所求的概率分布。对于更实际的例子来说, 所求的概率分布可能是多峰的, 并且具有尖峰, 从而找到一个较好的提议分布和比较函数是一件相当困难的事情。此外, 接受率随着维度的指数下降是拒绝采样的一个一般特征。虽然拒绝采样在一维或二维空间中是一个有用的方法, 但是它不适用于高维空间。然而, 对于高维空间中的更加复杂的算法来说, 它起着子过程的作用。

11.1.4 重要采样

想从复杂概率分布中采样的一个主要原因是能够使用公式 (11.1) 计算期望。重要采样 (importance sampling) 的方法提供了直接近似期望的框架, 但是它本身并没有提供从概率分布 $p(z)$ 中采样的方法。

公式 (11.2) 给出的期望的有限和近似依赖于能够从概率分布 $p(z)$ 中采样。然而, 假设直接从 $p(z)$ 中采样无法完成, 但是对于任意给定的 z 值, 我们可以很容易地计算 $p(z)$ 。一种简单的计

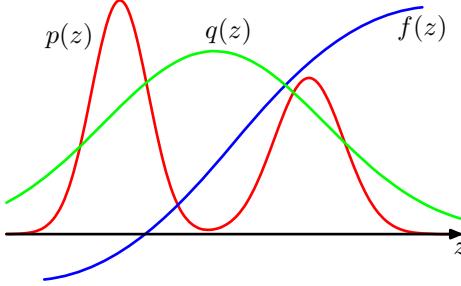


图 11.8: 重要采样解决了计算函数 $f(z)$ 关于分布 $p(z)$ 的期望的问题，其中，从 $p(z)$ 中直接采样比较困难。相反，样本 $\{z^{(l)}\}$ 从一个简单的概率分布 $q(z)$ 中抽取，求和式中的对应项的权值为 $p(z^{(l)})/q(z^{(l)})$ 。

算期望的方法是将 z 空间离散化为均匀的格点，将被积函数使用求和的方式计算，形式为

$$\mathbb{E}[f] \simeq \sum_{l=1}^L p(z^{(l)})f(z^{(l)}) \quad (11.18)$$

这种方法的一个明显的问题是求和式中的项的数量随着 z 的维度指数增长。此外，正如我们已经注意到的那样，我们感兴趣的概率分布通常将它们的大部分质量限制在 z 空间的一个很小的区域，因此均匀地采样非常低效，因为在高维的问题中，只有非常小的一部分样本会对求和式产生巨大的贡献。我们希望从 $p(z)$ 的值较大的区域中采样，或者理想情况下，从 $p(z)f(z)$ 的值较大的区域中采样。

与拒绝采样的情形相同，重要采样基于的是对提议分布 $q(z)$ 的使用，我们很容易从提议分布中采样，如图11.8所示。之后，我们可以通过 $q(z)$ 中的样本 $\{z^{(l)}\}$ 的有限和的形式来表示期望

$$\begin{aligned} \mathbb{E}[f] &= \int f(z)p(z) dz \\ &= \int f(z) \frac{p(z)}{q(z)} q(z) dz \\ &\simeq \frac{1}{L} \sum_{l=1}^L \frac{p(z^{(l)})}{q(z^{(l)})} f(z^{(l)}) \end{aligned} \quad (11.19)$$

$r_l = \frac{p(z^{(l)})}{q(z^{(l)})}$ 被称为重要性权重 (importance weights)，修正了由于从错误的概率分布中采样引入的偏差。注意，与拒绝采样不同，所有生成的样本都被保留。

常见的情形是，概率分布 $p(z)$ 的计算结果没有归一化，即 $p(z) = \tilde{p}(z)/Z_p$ ，其中 $\tilde{p}(z)$ 可以很容易地计算出来，而 Z_p 未知。类似地，我们可能希望使用重要采样分布 $q(z) = \tilde{q}(z)/Z_q$ ，它具有相同的性质。于是我们有

$$\begin{aligned} \mathbb{E}[f] &= \int f(z)p(z) dz \\ &= \frac{Z_q}{Z_p} \int f(z) \frac{\tilde{p}(z)}{\tilde{q}(z)} q(z) dz \\ &\simeq \frac{Z_q}{Z_p} \frac{1}{L} \sum_{l=1}^L \tilde{r}_l f(z^{(l)}) \end{aligned} \quad (11.20)$$

其中 $\tilde{r}_l = \frac{\tilde{p}(z^{(l)})}{\tilde{q}(z^{(l)})}$ 。我们可以使用同样的样本集合来计算比值 $\frac{Z_p}{Z_q}$ ，结果为

$$\begin{aligned} \frac{Z_p}{Z_q} &= \frac{1}{Z_q} \int \tilde{p}(z) dz = \int \frac{\tilde{p}(z)}{\tilde{q}(z)} q(z) dz \\ &\simeq \frac{1}{L} \sum_{l=1}^L \tilde{r}_l \end{aligned} \quad (11.21)$$

因此

$$\mathbb{E}[f] \simeq \sum_{l=1}^L w_l f(\mathbf{z}^{(l)}) \quad (11.22)$$

其中我们已经定义

$$w_l = \frac{\tilde{r}_l}{\sum_m \tilde{r}_m} = \frac{\frac{\tilde{p}(\mathbf{z}^{(l)})}{q(\mathbf{z}^{(l)})}}{\sum_m \frac{\tilde{p}(\mathbf{z}^{(m)})}{q(\mathbf{z}^{(m)})}} \quad (11.23)$$

与拒绝采样的情形相同，重要采样方法的成功严重依赖于采样分布 $q(\mathbf{z})$ 与所求的概率分布 $p(\mathbf{z})$ 的匹配程度。经常出现的情形是 $p(\mathbf{z})$ 变化剧烈，并且大部分的质量集中于 \mathbf{z} 空间的一个相对较小的区域中，此时重要性权重 $\{r_l\}$ 由几个具有较大值的权值控制，剩余的权值相对较小。因此，有效的样本集大小会比表面上的样本集大小 L 小得多。如果没有样本落在 $p(\mathbf{z})f(\mathbf{z})$ 较大的区域中，那么问题会更加严重。此时， r_l 和 $r_l f(\mathbf{z}^{(l)})$ 表面上的方差可能很小，即使期望的估计可能错得离谱。因此，重要性采样方法的一个主要的缺点是它具有产生任意错误的结果的可能性，并且这种错误无法检测。这也强调了采样分布 $q(\mathbf{z})$ 的一个关键的要求，即它不应该在 $p(\mathbf{z})$ 可能较大的区域中取得较小的值或者为零的值。

对于根据图模型定义的概率分布，我们可以用多种方式使用重要采样。对于离散变量，一个简单的方法被称为均匀采样（uniform sampling）。有向图的联合概率分布由公式（11.4）定义。联合概率分布中的每个样本都按照下面的方式获得：首先令证据集合中的变量 \mathbf{z}_i 等于它们的观测值。之后，每个剩余的变量从可能的实例空间中的均匀分布中独立地抽取。为了确定与一个样本 $\mathbf{z}^{(l)}$ 相关联的对应的权值，我们注意到采样分布 $\tilde{q}(\mathbf{z})$ 是 \mathbf{z} 的可能选择上的均匀分布，并且 $\tilde{p}(\mathbf{z} | \mathbf{x}) = \tilde{p}(\mathbf{z})$ ，其中 \mathbf{x} 表示观测变量的子集，等式来源于下面的事实：每个产生的样本 \mathbf{z} 都与证据相容。因此，权值 r_l 简单地正比于 $p(\mathbf{z})$ 。注意，变量可以以任意顺序采样。如果后验概率分布与均匀分布的差距较大，那么这种方法会产生较差的结果，而这正是实际应用中经常出现的情形。

这种方法的一个重要的提升被称为似然加权采样（likelihood weighted sampling）（Fung and Chang, 1990; Shachter and Peot, 1990），基于对变量的祖先采样。反过来对于每个变量，如果变量在证据集合中，那么它被简单地设置为它的实例值。如果它没在证据集合中，那么它从条件概率分布 $p(\mathbf{z}_i | \text{pa}_i)$ 中采样，其中条件变量被设置为它们当前的采样值。于是，与最终的样本 \mathbf{z} 关联的权值为

$$r(\mathbf{z}) = \prod_{\mathbf{z}_i \notin e} \frac{p(\mathbf{z}_i | \text{pa}_i)}{p(\mathbf{z}_i | \text{pa}_i)} \prod_{\mathbf{z}_i \in e} \frac{p(\mathbf{z}_i | \text{pa}_i)}{1} = \prod_{\mathbf{z}_i \in e} p(\mathbf{z}_i | \text{pa}_i) \quad (11.24)$$

这种方法可以进一步扩展，使用自重要采样（self-importance sampling）（Shachter and Peot, 1990），其中重要采样分布连续地更新，反映当前估计的后验概率分布。

11.1.5 采样-重要性-重采样

11.1.2节讨论的拒绝采样方法部分依赖于它能够成功确定常数 k 的一个合适的值。对于许多对概率分布 $p(\mathbf{z})$ 和 $q(\mathbf{z})$ 来说，确定一个合适的 k 值是不现实的，因为任意的足够大的 k 值都能够保证产生所求的分布的上界，但是这会产生相当小的接受率。

与拒绝采样的情形相同，采样-重要性-重采样（sampling-importance-resampling, SIR）方法也使用采样分布 $q(\mathbf{z})$ ，但是避免了必须确定常数 k 。这个方法有两个阶段。在第一个阶段， L 个样本 $\mathbf{z}^{(1)}, \dots, \mathbf{z}^{(L)}$ 从 $q(\mathbf{z})$ 中抽取。然后在第二个阶段，权值 w_1, \dots, w_L 通过公式（11.23）被构建出来。最后， L 个样本的第二个集合从离散概率分布 $(\mathbf{z}^{(1)}, \dots, \mathbf{z}^{(L)})$ 中抽取，概率由权值 (w_1, \dots, w_L) 给定。

生成的 L 个样本只是近似地服从 $p(\mathbf{z})$ ，但是在极限 $L \rightarrow \infty$ 的情况下，分布变为了正确的分

布。为了说明这一点，考虑一元变量的情形，并且注意重新采样的值的累积分布为

$$\begin{aligned} p(z \leq a) &= \sum_{l:z^{(l)} \leq a} w_l \\ &= \frac{\sum_l I(z^{(l)} \leq a) \tilde{p}(z^{(l)}) / q(z^{(l)})}{\sum_l \tilde{p}(z^{(l)}) / q(z^{(l)})} \end{aligned} \quad (11.25)$$

其中 $I(\cdot)$ 是示性函数（参数为真时函数值为1，否则为0）。取极限 $L \rightarrow \infty$ ，并且假设概率分布进行了适当的正则化，我们可以将求和替换为积分，权值为原始的采样分布 $q(z)$ ，即

$$\begin{aligned} p(z \leq a) &= \frac{\int I(z \leq a) \left\{ \frac{\tilde{p}(z)}{q(z)} \right\} q(z) dz}{\int \left\{ \frac{\tilde{p}(z)}{q(z)} \right\} q(z) dz} \\ &= \frac{\int I(z \leq a) \tilde{p}(z) dz}{\int \tilde{p}(z) dz} \\ &= \int I(z \leq a) p(z) dz \end{aligned} \quad (11.26)$$

它是 $p(z)$ 的累积分布函数。与之前一样，我们看到对 $p(z)$ 的归一化是不需要的。

对于 L 的一个有限值，以及一个给定的初始样本集合，重新采样的值只是近似地从所求的概率分布中抽取。与拒绝采样的情形相同，随着样本分布 $q(z)$ 接近所求的分布 $p(z)$ ，近似的效果也会提升。当 $q(z) = p(z)$ 时，初始样本 $(z^{(1)}, \dots, z^{(L)})$ 服从所求的概率分布，权值为 $w_n = \frac{1}{L}$ ，从而重新采样的值也服从所求的分布。

如果我们需要求出关于概率分布 $p(z)$ 的各阶矩，那么可以直接使用原始样本和权值进行计算，因为

$$\begin{aligned} \mathbb{E}[f(z)] &= \int f(z) p(z) dz \\ &= \frac{\int f(z) \left[\frac{\tilde{p}(z)}{q(z)} \right] q(z) dz}{\int \left[\frac{\tilde{p}(z)}{q(z)} \right] q(z) dz} \\ &\simeq \sum_{l=1}^L w_l f(z^{(l)}) \end{aligned} \quad (11.27)$$

11.1.6 采样与EM算法

蒙特卡罗方法除了为贝叶斯框架的直接实现提供了原理，还在频率学家的框架内起着重要的作用，例如寻找最大似然解。特别地，对于EM算法中的E步骤无法解析地计算的模型，采样方法也可以用来近似E步骤。考虑一个模型，它的隐含变量为 Z ，可见（观测）变量为 X ，参数为 θ 。在M步骤中关于 θ 最大化的步骤为完整数据对数似然的期望，形式为

$$Q(\theta, \theta^{\text{旧}}) = \int p(Z | X, \theta^{\text{旧}}) \ln p(Z, X | \theta) dZ \quad (11.28)$$

我们可以使用采样方法来近似这个积分，方法是计算样本 $\{Z^{(l)}\}$ 上的有限和，这些样本是从当前的对后验概率分布 $p(Z | X, \theta^{\text{旧}})$ 的估计中抽取的，即

$$Q(\theta, \theta^{\text{旧}}) \simeq \frac{1}{L} \sum_{l=1}^L \ln p(Z^{(l)}, X | \theta) \quad (11.29)$$

然后， Q 函数在M步骤中使用通常的步骤进行优化。这个步骤被称为蒙特卡罗EM算法（Monte Carlo EM algorithm）。

将这种方法推广到寻找 θ 上的后验概率的峰值（MAP估计）的问题是很容易的，其中先验概率分布 $p(\theta)$ 已经被定义。我们只需在进行M步骤之前，在函数 $Q(\theta, \theta^{(t)})$ 中加上 $\ln p(\theta)$ 即可。

蒙特卡罗EM算法的一个特定的情形，被称为随机EM（stochastic EM）。如果我们考虑有限数量的概率分布组成的混合模型，并且在每个E步骤中只抽取一个样本时，我们就会用到这种算法。这里，潜在变量 Z 描述了K个混合分量中的哪个分量被用于生成每个数据点。在E步骤中， Z 的样本从后验概率分布 $p(Z | X, \theta^{(t)})$ 中抽取，其中 X 是数据集。这高效地将每个数据点硬性地分配到混合分布中的一个分量中。在M步骤中，对于后验概率分布的这个采样的近似被用于按照平常的方式更新模型的参数。

现在假设我们从最大似然的方法转移到纯粹的贝叶斯方法，其中我们希望从参数向量 θ 上的后验概率分布中进行采样。原则上，我们希望从联合后验分布 $p(\theta, Z | X)$ 中抽取样本，但是我们假设这个计算十分困难。进一步地，我们假设从完整数据参数的后验概率分布 $p(\theta | Z, X)$ 中进行采样相对简单。这就产生了数据增广算法（data augmentation algorithm），它在两个步骤之间交替进行，这两个步骤被称为I步骤（归咎（imputation）步骤，类似于E步骤）和P步骤（后验（posterior）步骤，类似于M步骤）。

I步骤。我们希望从概率分布 $p(Z | X)$ 采样，但是我们不能直接进行。于是，我们注意到下面的关系

$$p(Z | X) = \int p(Z | \theta, X) p(\theta | X) d\theta \quad (11.30)$$

因此对于 $l = 1, \dots, L$ ，我们首先从当前对 $p(\theta | X)$ 的估计中抽取样本 $\theta^{(l)}$ ，然后使用这个样本从 $p(Z | \theta^{(l)}, X)$ 中抽取样本 $Z^{(l)}$ 。

P步骤。给定关系

$$p(\theta | X) = \int p(\theta | Z, X) p(Z | X) dZ \quad (11.31)$$

我们使用从I步骤中得到的样本 $\{Z^{(l)}\}$ ，计算 θ 上的后验概率分布的修正后的估计，结果为

$$p(\theta | X) \simeq \frac{1}{L} \sum_{l=1}^L p(\theta | Z^{(l)}, X) \quad (11.32)$$

根据假设，在I步骤中从这个近似分布中采样是可行的。

注意，我们对参数 θ 和隐含变量 Z 进行了（多少有些人为的）区分。从现在开始，我们不进行这种区分，仅仅集中于从给定的后验概率分布中抽取样本的问题。

11.2 马尔科夫链蒙特卡罗

前一节中，我们讨论了计算函数期望的拒绝采样方法和重要采样方法，我们看到在高维空间中，这两种方法具有很大的局限性。因此，我们在本节中讨论一个非常一般的并且强大的框架，被称为马尔科夫链蒙特卡罗（Markov chain Monte Carlo, MCMC），它使得我们可以从一大类概率分布中进行采样，并且可以很好地应对样本空间维度的增长。马尔科夫链蒙特卡罗方法起源于物理学（Metropolis and Ulam, 1949），直到20世纪80年代，这种方法才开始对统计学领域产生巨大的影响。

与拒绝采样和重要采样相同，我们再一次从提议分布中采样。但是这次我们记录下当前状态 $z^{(\tau)}$ ，以及依赖于这个当前状态的提议分布 $q(z | z^{(\tau)})$ ，从而样本序列 $z^{(1)}, z^{(2)}, \dots$ 组成了一个马尔科夫链。与之前一样，如果我们有 $p(z) = \frac{\tilde{p}(z)}{Z_p}$ ，那么我们会假定对于任意的 z 值都可以计算 $\tilde{p}(z)$ ，虽然 Z_p 的值可能未知。提议分布本身被选择为足够简单，从而直接采样很容易。在算法的每次迭代中，我们从提议分布中生成一个候选样本 z^* ，然后根据一个恰当的准则接受这个样本。

在基本的Metropolis算法中（Metropolis et al., 1953），我们假定提议分布是对称的，即 $q(z_A | z_B) = q(z_B | z_A)$ 对于所有的 z_A 和 z_B 成立。这样，候选的样本被接受的概率为

$$A(z^*, z^{(\tau)}) = \min \left(1, \frac{\tilde{p}(z^*)}{\tilde{p}(z^{(\tau)})} \right) \quad (11.33)$$

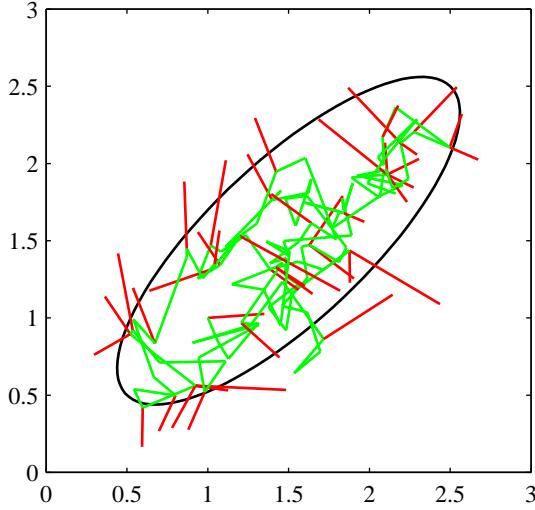


图 11.9: 使用Metropolis算法从一个高斯分布中采样的简单例子，这个高斯分布的一个标准差的位置用椭圆表示。提议分布是一个各向同性的高斯分布，标准差为0.2。被接受的步骤用绿线表示，被拒绝的步骤用红线表示。总共生成了150个候选样本，其中有43个被拒绝。

可以这样实现：在单位区间 $(0, 1)$ 上的均匀分布中随机选择一个数 u ，然后如果 $A(z^*, z^{(\tau)}) > u$ 就接受这个样本。注意，如果从 z^τ 到 z^* 引起了 $p(z)$ 的值的增大，那么这个候选样本当然会被保留。

如果候选样本被接受，那么 $z^{(\tau+1)} = z^*$ ，否则候选样本点 z^* 被丢弃， $z^{(\tau+1)}$ 被设置为 $z^{(\tau)}$ ，然后从概率分布 $q(z | z^{(\tau+1)})$ 中再次抽取一个候选样本。这与拒绝采样不同，那里拒绝的样本被简单地丢弃。在Metropolis算法中，当一个候选点被拒绝时，前一个样本点会被包含到最终的样本的列表中，从而产生了样本点的多个副本。当然，在实际实现中，每个保留的样本只会有一个副本，以及一个整数的权因子，记录状态出现了多少次。正如我们将看到的那样，只要对于任意的 z_A 和 z_B 都有 $q(z_A | z_B) \neq 0$ （这是一个充分条件但不是必要条件），那么当 $\tau \rightarrow \infty$ 时， $z^{(\tau)}$ 趋近于 $p(z)$ 。然而，应该强调的是，序列 $z^{(1)}, z^{(2)}, \dots$ 不是来自 $p(z)$ 的一组独立的样本，因为连续的样本是高度相关的。如果我们希望得到独立的样本，那么我们可以丢弃序列中的大部分样本，每 M 个样本中保留一个样本。对于充分大的 M ，保留的样本点对于所有实际用途来说都是独立的。图11.9给出了一个简单的例子，这个例子使用Metropolis算法从一个二维高斯分布中采样，其中提议分布是一个各向同性的高斯分布。

通过考察一个具体的例子，即简单的随机游走的例子，我们可以对马尔科夫链蒙特卡罗算法的本质得到更深刻的认识。考虑一个由整数组成的状态空间 z ，概率为

$$p(z^{(\tau+1)} = z^{(\tau)}) = 0.5 \quad (11.34)$$

$$p(z^{(\tau+1)} = z^{(\tau)} + 1) = 0.25 \quad (11.35)$$

$$p(z^{(\tau+1)} = z^{(\tau)} - 1) = 0.25 \quad (11.36)$$

其中 $z^{(\tau)}$ 表示在步骤 τ 的状态。如果初始状态是 $z^{(0)} = 0$ ，那么根据对称性，在时刻 τ 的期望状态也是零，即 $\mathbb{E}[z^{(\tau)}] = 0$ ，类似地很容易看到 $\mathbb{E}[(z^{(\tau)})^2] = \frac{\tau}{2}$ 。因此，在 τ 步骤之后，随机游走所经过的平均距离正比于 τ 的平方根。这个平方根依赖关系是随机游走行为的一个典型性质，表明了随机游走在探索状态空间时是很低效的。正如我们会看到的那样，设计马尔科夫链蒙特卡罗方法的一个中心目标就是避免随机游走行为。

11.2.1 马尔科夫链

在详细讨论马尔科夫链蒙特卡罗方法之前，仔细研究马尔科夫链的一些一般的性质是很有用的。特别地，我们考察在什么情况下马尔科夫链会收敛到所求的概率分布上。一阶马尔科夫链

被定义为一系列随机变量 $z^{(1)}, \dots, z^{(M)}$, 使得下面的条件独立性质对于 $m \in \{1, \dots, M-1\}$ 成立

$$p(z^{(m+1)} | z^{(1)}, \dots, z^{(m)}) = p(z^{(m+1)} | z^{(m)}) \quad (11.37)$$

这当然可以表示成链形的有向图, 如图 8.38 所示。之后, 我们可以按照下面的方式具体化一个马尔科夫链: 给定初始变量的概率分布 $p(z^{(0)})$, 以及后续变量的条件概率, 用转移概率 (transition probability) $T_m(z^{(m)}, z^{(m+1)}) \equiv p(z^{(m+1)} | z^{(m)})$ 的形式表示。如果对于所有的 m , 转移概率都相同, 那么这个马尔科夫链被称为同质的 (homogeneous)。

对于一个特定的变量, 边缘概率可以根据前一个变量的边缘概率用链式乘积的方式表示出来, 形式为

$$p(z^{(m+1)}) = \sum_{z^{(m)}} p(z^{(m+1)} | z^{(m)}) p(z^{(m)}) \quad (11.38)$$

对于一个概率分布来说, 如果马尔科夫链中的每一步都让这个概率分布保持不变, 那么我们说这个概率分布关于这个马尔科夫链是不变的, 或者静止的。因此, 对于一个转移概率为 $T(z', z)$ 的同质的马尔科夫链来说, 如果

$$p^*(z) = \sum_{z'} T(z', z) p^*(z') \quad (11.39)$$

那么概率分布 $p^*(z)$ 是不变的。注意, 一个给定的马尔科夫链可能有多个不变的概率分布。例如, 如果转移概率由恒等变换给出, 那么任意的概率分布都是不变的。

确保所求的概率分布 $p(z)$ 不变的一个充分 (非必要) 条件是令转移概率满足细节平衡 (detailed balance) 性质, 定义为

$$p^*(z) T(z, z') = p^*(z') T(z', z) \quad (11.40)$$

对特定的概率分布 $p^*(z)$ 成立。很容易看到, 满足关于特定概率分布的细节平衡性质的转移概率会使得那个概率分布具有不变性, 因为

$$\sum_{z'} p^*(z') T(z', z) = \sum_{z'} p^*(z) T(z, z') = p^*(z) \sum_{z'} p(z' | z) = p^*(z) \quad (11.41)$$

满足细节平衡性质的马尔科夫链被称为可翻转的 (reversible)。

我们的目标是使用马尔科夫链从一个给定的概率分布中采样。如果我们构造一个马尔科夫链使得所求的概率分布是不变的, 那么我们就可以达到这个目标。然而, 我们还要要求对于 $m \rightarrow \infty$, 概率分布 $p(z^{(m)})$ 收敛到所求的不变的概率分布 $p^*(z)$, 与初始概率分布 $p(z^{(0)})$ 无关。这种性质被称为各态历经性 (ergodicity), 这个不变的概率分布被称为均衡 (equilibrium) 分布。很明显, 一个具有各态历经性的马尔科夫链只能有唯一的一个均衡分布。可以证明, 同质的马尔科夫链具有各态历经性, 只需对不变的概率分布和转移概率做出较弱的限制即可 (Neal, 1993)。

在实际中, 我们经常可以从一组“基”转移 B_1, \dots, B_K 中构建转移概率, 方法为: 将各个“基”转移表示为混合概率分布, 形式为

$$T(z', z) = \sum_{k=1}^K \alpha_k B_k(z', z) \quad (11.42)$$

混合系数 $\alpha_1, \dots, \alpha_K$ 满足 $\alpha_k \geq 0$ 且 $\sum_k \alpha_k = 1$ 。此外, 基转移可以通过连续的应用组合到一起, 即

$$T(z', z) = \sum_{z_1} \dots \sum_{z_{K-1}} B_1(z', z_1) \dots B_{K-1}(z_{K-2}, z_{K-1}) B_K(z_{K-1}, z) \quad (11.43)$$

如果一个概率分布关于每个基转移都是不变的, 那么显然它关于公式 (11.42) 和 (11.43) 也是不变的。对于公式 (11.42) 的混合分布, 如果每个基转移满足细节平衡, 那么混合转移 T 也满足细节平衡。这对于使用公式 (11.43) 构造的转移概率不成立, 虽然通过将基转移的顺序对称化, 即采用 $B_1, B_2, \dots, B_K, B_K, \dots, B_2, B_1$ 的形式, 细节平衡的性质可以被恢复。使用组合转移概率的一个常见的例子是每个基转移只改变变量的一个子集的情形。

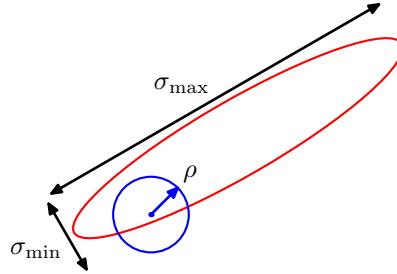


图 11.10: 使用Metropolis-Hastings算法, 用一个各项同性的高斯提议分布 (蓝色圆圈) 从一个具有相关性的多元高斯分布 (红色椭圆) 中采样, 这个多元高斯分布在不同的方向上的标准差的数值相当不同。为了让拒绝率较低, 提议分布的标度 ρ 应该与最小的标准差 σ_{\min} 处于同一个量级, 这会产生随机游走的行为, 达到独立的状态所需的步骤数的量级为 $(\sigma_{\max}/\sigma_{\min})^2$, 其中 σ_{\max} 是最大的标准差。

11.2.2 Metropolis-Hastings算法

之前我们介绍了基本的Metropolis算法, 没有实际演示它从所求的概率分布中采样的过程。在给出一个证明之前, 我们首先讨论一个推广, 被称为Metropolis-Hastings算法 (Hastings, 1970), 这种情形下, 提议分布不再是参数的一个对称函数。特别地, 在算法的步骤 τ 中, 当前状态为 $z^{(\tau)}$, 我们从概率分布 $q_k(z | z^{(\tau)})$ 中抽取一个样本 z^* , 然后以概率 $A_k(z^*, z^{(\tau)})$ 接受它, 其中

$$A_k(z^*, z^{(\tau)}) = \min \left(1, \frac{\tilde{p}(z^*) q_k(z^{(\tau)} | z^*)}{\tilde{p}(z^{(\tau)}) q_k(z^* | z^{(\tau)})} \right) \quad (11.44)$$

这里, k 标记出可能的转移集合中的成员。与之前一样, 接受准则的计算不需要知道概率分布 $p(z) = \frac{\tilde{p}(z)}{Z_p}$ 中的归一化常数 Z_p 。对于一个对称的提议分布, Metropolis-Hastings准则 (11.44) 会简化为标准的Metropolis准则 (11.33)。

我们现在可以证明 $p(z)$ 对于由Metropolis-Hastings算法定义的马尔科夫链是一个不变的概率分布, 方法是证明公式 (11.40) 定义的细节平衡是成立的。使用公式 (11.44), 我们有

$$\begin{aligned} p(z) q_k(z' | z) A_k(z', z) &= \min(p(z) q_k(z' | z), p(z') q_k(z | z')) \\ &= \min(p(z') q_k(z | z'), p(z) q_k(z' | z)) \\ &= p(z') q_k(z | z') A_k(z, z') \end{aligned} \quad (11.45)$$

证明完毕。

提议分布的具体的选择会对算法的表现产生重要的影响。对于连续状态空间来说, 一个常见的选择是一个以当前状态为中心的高斯分布, 这会在确定分布的方差参数时需要进行一个重要的折中。如果方差过小, 那么接受的转移的比例会很高, 但是遍历状态空间的形式是一个缓慢的随机游走过程, 导致较长的时间开销。然而, 如果方差过大, 那么拒绝率会很高, 因为在我们考虑的这种复杂问题中, 许多的步骤会到达 $p(z)$ 很低的状态。考虑一个多元概率分布 $p(z)$, 它在 z 的元素之间具有很强的相关性, 如图11.10所示。提议分布的标度 ρ 应该尽可能大, 同时要避免达到较高的拒绝率。这表明, ρ 应该与最小的长度标度 σ_{\min} 是同一个量级的。然后, 系统通过随机游走的方式探索伸长的方向, 因此到达一个与原始状态或多或少独立的状态所需的步骤数量是 $(\sigma_{\max}/\sigma_{\min})^2$ 量级的。事实上, 在二维的情形下, 随着 ρ 的增加, 拒绝率的增加会被接收的转移步骤数的增加所抵消。更一般地, 对于多元高斯分布, 得到独立样本所需的步骤的数量的增长量级是 $(\sigma_{\max}/\sigma_2)^2$ 的, 其中 σ_2 是第二小的标准差 (Neal, 1993)。抛开这些细节不谈, 如果概率分布在不同的方向上的差异非常大, 那么Metropolis-Hastings算法的收敛速度会非常慢。

11.3 吉布斯采样

吉布斯采样 (Geman and Geman, 1984) 是一个简单的并且广泛应用的马尔科夫链蒙特卡罗算法, 可以被看做Metropolis-Hastings算法的一个具体的情形。

考虑我们想采样的概率分布 $p(z) = p(z_1, \dots, z_M)$, 并且假设我们已经选择了马尔科夫链的某个初始的状态。吉布斯采样的每个步骤涉及到将一个变量的值替换为以剩余变量的值为条件,

剩余概率的值

从这个概率分布中抽取的那个变量的值。因此我们将 z_i 替换为从概率分布 $p(z_i | z_{\setminus i})$ 中抽取的值，其中 z_i 表示 z 的第*i*个元素， $z_{\setminus i}$ 表示 z_1, \dots, z_M 去掉 z_i 这一项。这个步骤要么按照某种特定的顺序在变量之间进行循环，要么每一步中按照某个概率分布随机地选择一个变量进行更新。

例如，假设我们有一个在三个变量上的概率分布 $p(z_1, z_2, z_3)$ ，在算法的第 τ 步，我们已经选择了 $z_1^{(\tau)}, z_2^{(\tau)}, z_3^{(\tau)}$ 的值。首先，我们将 $z_1^{(\tau)}$ 替换为新值 $z_1^{(\tau+1)}$ ，这个新值是从条件概率分布

$$p(z_1 | z_2^{(\tau)}, z_3^{(\tau)}) \quad (11.46)$$

中采样得到的。接下来，我们将 $z_2^{(\tau)}$ 替换为 $z_2^{(\tau+1)}$ ，这个新值是从条件概率分布

$$p(z_2 | z_1^{(\tau+1)}, z_3^{(\tau)}) \quad (11.47)$$

中采样得到的，即 z_1 的新值可以在接下来的采样步骤中直接使用。然后，我们使用样本 $z_3^{(\tau+1)}$ 更新 z_3 ，其中 $z_3^{(\tau+1)}$ 是从

$$p(z_3 | z_1^{(\tau+1)}, z_2^{(\tau+1)}) \quad (11.48)$$

中抽取的。以此类推，在这三个变量之间进行循环。

- 初始化 $\{z_i : i = 1, \dots, M\}$ 。
- 对于 $\tau = 1, \dots, T$ ：
 - 采样 $z_1^{(\tau+1)} \sim p(z_1 | z_2^{(\tau)}, z_3^{(\tau)}, \dots, z_M^{(\tau)})$ 。
 - 采样 $z_2^{(\tau+1)} \sim p(z_2 | z_1^{(\tau+1)}, z_3^{(\tau)}, \dots, z_M^{(\tau)})$ 。
 - \vdots
 - 采样 $z_j^{(\tau+1)} \sim p(z_j | z_1^{(\tau+1)}, \dots, z_{j-1}^{(\tau+1)}, z_{j+1}^{(\tau)}, \dots, z_M^{(\tau)})$ 。
 - \vdots
 - 采样 $z_M^{(\tau+1)} \sim p(z_M | z_1^{(\tau+1)}, z_2^{(\tau+1)}, \dots, z_{M-1}^{(\tau+1)})$ 。

为了证明这个步骤能够从所需的概率分布中采样，我们首先注意到对于吉布斯采样的每个步骤来说，概率分布 $p(z)$ 是不变的，因此对于整个马尔科夫链来说也是不变的。这是由于当我们从 $p(z_i | z_{\setminus i})$ 中采样时，边缘概率分布 $p(z_i)$ 显然是不变的，因为 $z_{\setminus i}$ 的值是不变的。并且，根据定义，对于每个步骤中来自正确条件概率分布 $p(z_i | z_{\setminus i})$ 的样本，条件概率分布都是不变的。由于条件概率分布和边缘概率分布共同确定的联合概率分布，因此我们看到联合概率分布本身是不变的。

为了让吉布斯采样能够从正确的概率分布中得到样本，第二个需要满足的要求为各态历经性。各态历经性的一个充分条件是没有条件概率分布处处为零。如果这个要求满足，那么 z 空间中的任意一点都可以从其他的任意一点经过有限步骤达到，这些步骤中每次对一个变量进行更新。如果这个要求没有满足，即某些条件概率分布为零，那么在这种情况下应用吉布斯采样时，必须显式地证明各态历经性。

为了完成算法，初始状态的概率分布也应该被指定，虽然在多轮迭代之后，样本与初始状态的分布无关。当然，马尔科夫链中的连续的样本是高度相关的，因此为了得到近似独立的样本，需要对序列进行下采样。

我们可以将吉布斯采样步骤看成Metropolis-Hastings算法的一个特定的情况，如下所述。考虑一个Metropolis-Hastings采样的步骤，它涉及到变量 z_k ，同时保持剩余的变量 $z_{\setminus k}$ 不变，并且对于这种情形来说，从 z 到 z^* 的转移概率为 $q_k(z^* | z) = p(z_k^* | z_{\setminus k})$ 。我们注意到 $z_{\setminus k}^* = z_{\setminus k}$ ，因为在采样的步骤中，向量的各个元素都不改变。并且， $p(z) = p(z_k | z_{\setminus k})p(z_{\setminus k})$ 。因此，确定Metropolis-Hastings算法中的接受概率的因子 (11.44) 为

$$A(z^*, z) = \frac{p(z^*)q_k(z | z^*)}{p(z)q_k(z^* | z)} = \frac{p(z_k^* | z_{\setminus k})p(z_{\setminus k}^*)p(z_k | z_{\setminus k}^*)}{p(z_k | z_{\setminus k})p(z_{\setminus k})p(z_k^* | z_{\setminus k})} = 1 \quad (11.49)$$

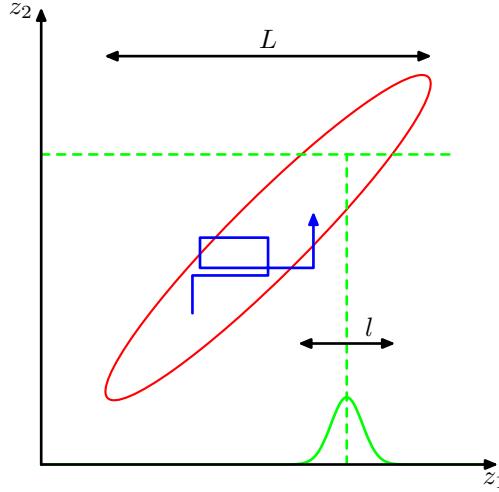


图 11.11: 通过交替更新两个变量的方式进行吉布斯采样。这两个变量服从一个相关的高斯分布。步长由条件概率分布（绿色曲线）的标准差控制，值为 $O(l)$ ，在联合概率分布较长的方向上的速度很慢。得到这个分布的独立样本所需的步骤数量为 $O((L/l)^2)$ 。

推导时我们用到了 $z_{\setminus k}^* = z_{\setminus k}$ 。因此 Metropolis-Hastings 步骤总是被接受的。

与 Metropolis 算法一样，我们可以通过研究吉布斯采样算法在高斯分布上的应用，更深刻地认识算法的原理。考虑两个相关变量上的一个高斯分布，如图 11.11 所示。这个高斯分布的条件概率分布的宽度为 l ，边缘概率分布的宽度为 L 。典型的步长由条件概率分布确定，从而量级为 l 。由于状态按照随机游走的方式进行转移，因此得到这个分布中的独立样本所需的步骤数量的量级为 $(L/l)^2$ 。当然，如果高斯分布不是相关的，那么吉布斯采样的效率是最高的。对于这个问题，我们可以将坐标系旋转，从而解除变量之间的相关关系。然而，在实际应用中，通常找到这种变换是不可行的。

一种减小吉布斯采样过程中的随机游走行为的方法被称为过松弛（over-relaxation）（Adler, 1981）。在这种方法的最初的形式中，它被用于处理条件概率分布是高斯分布的情形，这种情形要比多元高斯分布更一般，因为诸如非高斯分布 $p(z, y) \propto \exp(-z^2 y^2)$ 具有高斯条件分布的形式。在吉布斯采样算法的每个步骤中，对于一个特定的分量 z_i ，条件概率分布具有均值 μ_i 和方差 σ_i^2 。在过松弛框架中， z_i 被替换为

$$z'_i = \mu_i + \alpha(z_i - \mu_i) + \sigma_i(1 - \alpha^2)^{\frac{1}{2}}\nu \quad (11.50)$$

其中 ν 是一个高斯随机变量，均值为 0，方差为 1， α 是一个参数，满足 $-1 < \alpha < 1$ 。对于 $\alpha = 0$ 的情形，方法等价于标准的吉布斯采样，对于 $\alpha < 0$ ，步骤会偏向于与均值相反的一侧。这个步骤使得所求的概率分布具有不变性，因为如果 z_i 的均值为 μ_i ，方差为 σ_i^2 ，那么 z'_i 也是。过松弛的效果是当变量高度相关时，鼓励在状态空间中的直接移动。有序过松弛（ordered over-relaxation）框架（Neal, 1999）将这种方法推广到了非高斯分布的情形。

吉布斯采样的实际应用依赖于哪个样本可以从条件概率分布 $p(z_k | z_{\setminus k})$ 中抽取。在概率分布使用图模型表示的情况下，各个结点的条件概率分布只依赖于对应的马尔科夫链中的变量，如图 11.12 所示。对于有向图来说，以某个结点的父结点为条件，这个结点的一大类条件概率分布都会使得用于吉布斯采样的概率分布是对数凹函数。于是，11.1.3 节讨论的可调节拒绝采样方法提供了有向图的蒙特卡罗采样方法的一个框架，这种方法具有广泛的适用性。

如果图是使用指数族分布构建的，并且父结点-子结点关系保持共轭，那么吉布斯采样中的完整的条件概率分布会与定义在每个结点的原始的条件概率分布（以父结点为条件）具有相同的函数形式，因此可以使用标准的采样方法。通常，完整的条件概率分布的形式会很复杂，从而无法使用标准的采样方法。然而，如果这些条件概率分布是对数凹函数，那么使用可调整的拒绝采样方法，采样可以高效地完成（假设对应的变量是标量）。

如果在吉布斯采样算法的每个阶段，我们不从对应的条件概率分布中抽取样本，而是对变量进行一个点估计，这个点估计由条件概率分布的最大值给出，那么我们就得到了 8.3.3 节讨论的迭代条件峰值（ICM）算法。因此，ICM 可以看成是吉布斯采样的一种贪心近似。



图 11.12: 吉布斯采样方法要求样本从一个变量的条件概率分布中抽取，条件是其他的变量。对于图模型来说，条件概率分布只是马尔科夫毯中的结点状态的函数。对于无向图来说，马尔科夫毯由相邻结点的集合组成，如左图所示。而对于有向图来说，马尔科夫毯由父结点、子结点、同父结点组成，如右图所示。

由于基本的吉布斯采样方法每次只考虑一个变量，因此它在连续样本之间具有很强的依赖性。在另一个极端情况下，如果我们直接从联合概率分布中采样（我们一直假定这种操作无法完成），那么连续的样本点之间就是独立的。我们可以采用一种折中的方法来提升简单的吉布斯采样的效果，即我们连续地对一组变量进行采样，而不是对一个变量进行采样。这就是分块吉布斯（blocking Gibbs）采样算法。这种算法中，将变量集合分块（未必互斥），然后在每个块内部联合地采样，采样时以剩余的变量为条件（Jensen et al., 1995）。

11.4 切片采样

我们已经看到，Metropolis算法的一个困难之处是它对于步长的敏感性。如果步长过小，那么由于随机游走行为，算法会很慢。而如果步长过大，那么由于较高的拒绝率，算法会很无效。切片采样（slice sampling）方法（Neal, 2003）提供了一个可以自动调节步长来匹配分布特征的方法。与之前一样，它需要我们能够计算未归一化的概率分布 $\tilde{p}(z)$ 。

首先考虑一元变量的情形。切片采样涉及到使用额外的变量 u 对 z 进行增广，然后从联合的 (z, u) 空间中采样。当我们在11.5节讨论混合蒙特卡罗方法时，我们会看到这种方法的另一个例子。目标是从下面的概率分布

$$\hat{p}(z, u) = \begin{cases} \frac{1}{Z_p} & \text{如果 } 0 \leq u \leq \tilde{p}(z) \\ 0 & \text{其他情况} \end{cases} \quad (11.51)$$

中均匀地进行采样，其中 $Z_p = \int \tilde{p}(z) dz$ 。 z 上的边缘概率分布为

$$\int \hat{p}(z, u) du = \int_0^{\tilde{p}(z)} \frac{1}{Z_p} du = \frac{\tilde{p}(z)}{Z_p} = p(z) \quad (11.52)$$

因此，我们可以通过从 $\hat{p}(z, u)$ 中采样，然后忽略 u 值的方式得到 $p(z)$ 的样本。通过交替地对 z 和 u 进行采样即可完成这一点。给定 z 的值，我们可以计算 $\tilde{p}(z)$ 的值，然后在 $0 \leq u \leq \tilde{p}(z)$ 上均匀地对 u 进行采样，这很容易。然后，我们固定 u ，在由 $\{z : \tilde{p}(z) > u\}$ 定义的分布的“切片”上，对 z 进行均匀地采样。图11.13(a)给出了说明。

在实际应用中，直接从穿过概率分布的切片中采样很困难，因此我们定义了一个采样方法，它保持 $\hat{p}(z, u)$ 下的均匀分布具有不变性，这可以通过确保满足细节平衡的套件来实现。假设 z 的当前值记作 $z^{(\tau)}$ ，并且我们已经得到了一个对应的样本 u 。 z 的下一个值可以通过考察包含 $z^{(\tau)}$ 的区域 $z_{\min} \leq z \leq z_{\max}$ 来获得。根据概率分布的特征长度标度来对步长进行的调节就发生在这里。我们希望区域包含尽可能多的切片，从而使得 z 空间中能进行较大的移动，同时希望切片外的区域尽可能小，因为切片外的区域会使得采样变得低效。

一种选择区域的方法是，从一个包含 $z^{(\tau)}$ 的具有某个宽度 w 的区域开始，然后测试每个端点，看它们是否位于切片内部。如果有端点没在切片内部，那么区域在增加 w 值的方向上进行扩展，知道端点位于区域外。然后， z' 的一个样本被从这个区域中均匀抽取。如果它位于切片内，那么它就构成了 $z^{(\tau+1)}$ 。如果它位于切片外，那么区域收缩，使得 z' 组成一个端点，并且区

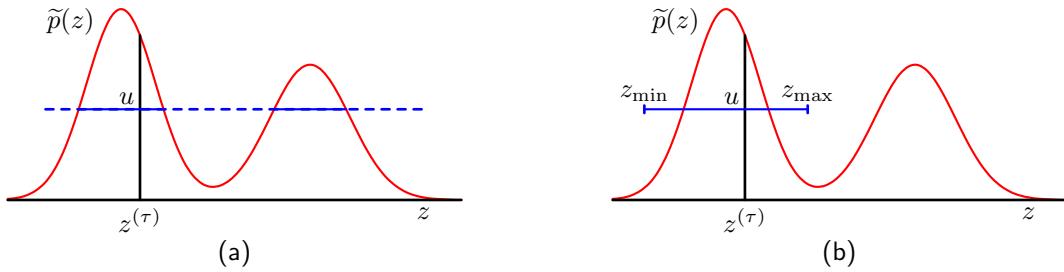


图 11.13: 切片采样的例子。(a)对于给定的 $z^{(\tau)}$, u 的值从 $0 \leq u \leq \tilde{p}(z^{(\tau)})$ 区域中均匀采样, 它之后定义了穿过这个概率分布的一个“切片”。(b)由于直接从切片中采样是不可行的, 因此 z 的一个新的样本被从区域 $z_{\min} \leq z \leq z_{\max}$ 中抽取, 它包含了前一个值 $z^{(\tau)}$ 。

域仍然包含 $z^{(\tau)}$ 。然后, 另一个样本点从这个缩小的区域中均匀抽取, 以此类推, 直到找到位于切片内部的一个 z 值。

切片采样可以应用于多元分布中, 方法是按照吉布斯采样的方式重复地对每个变量进行采样。这要求对于每个元素 z_i , 我们能够计算一个正比于 $p(z_i | z_{\setminus i})$ 的函数。

11.5 混合蒙特卡罗算法

正如我们已经注意到的那样, Metropolis算法的一个主要的局限是它具有随机游走的行为, 而在状态空间中遍历的距离与步骤数量只是平方根的关系。仅仅通过增大步长的方式是无法解决这个问题的, 因为这会使得拒绝率变高。

本节中, 我们介绍一类更加复杂的转移方法。这些方法基于对物理系统的一个类比, 能够让系统状态发生较大的改变, 同时让拒绝的概率较低。它适用于连续变量上的概率分布, 对于连续变量, 我们已经能够计算对数概率关于状态变量的梯度。我们会在11.5.1节讨论动态系统框架, 然后在11.5.2节, 我们会解释这个框架如何与Metropolis算法结合, 产生出一个强大的混合蒙特卡罗算法。这里不需要物理学的背景, 因为本节是自洽的, 并且关键的结果全部从基本的原理中推导出。

11.5.1 动态系统

随机采样的动态方法起源于模拟哈密顿力学下进行变化的物理系统的行为。在马尔科夫链蒙特卡罗模拟中, 目标是从一个给定的概率分布 $p(z)$ 中采样。通过将概率仿真转化为哈密顿系统的形式, 我们可以利用哈密顿力学 (Hamiltonian dynamics) 的框架。为了与这个领域的文献保持一致, 我们在必要的时候会使用相关动态系统的术语, 这些术语会随着我们内容的推进而给出定义。

我们考虑的动力学对应于在连续时刻 (记作 τ) 下的状态变量 $z = \{z_i\}$ 的演化。经典的动力学由牛顿第二定律描述, 即物体的加速度正比于施加的力, 对应于关于时间的二阶微分方程。我们可以将一个二阶微分方程分解为两个相互偶合的一阶方程, 方法是引入中间的动量 (momentum) 变量 r , 对应于状态变量 z 的变化率, 元素为

$$r_i = \frac{dz_i}{d\tau} \quad (11.53)$$

从动力学的角度, z_i 可以被看做位置 (position) 变量。因此对于每个位置变量, 都存在一个对应的动量变量, 位置和动量组成的联合空间被称为相空间 (phase space)。

不失一般性, 我们可以将概率分布 $p(z)$ 写成下面的形式

$$p(z) = \frac{1}{Z_p} \exp(-E(z)) \quad (11.54)$$

其中 $E(z)$ 可以看做状态 z 处的势能 (potential energy)。系统的加速度是动量的变化率, 通过施加力 (force) 的方式确定, 它本身是势能的负梯度, 即

$$\frac{dr_i}{d\tau} = -\frac{\partial E(z)}{\partial z_i} \quad (11.55)$$

使用哈密顿框架重新写出这个动态系统的公式是比较方便的。为了完成这一点，我们首先将动能 (kinetic energy) 定义为

$$K(\mathbf{r}) = \frac{1}{2} \|\mathbf{r}\|^2 = \frac{1}{2} \sum_i r_i^2 \quad (11.56)$$

系统的总能量是势能和动能之和，即

$$H(\mathbf{z}, \mathbf{r}) = E(\mathbf{z}) + K(\mathbf{r}) \quad (11.57)$$

其中 H 是哈密顿函数 (Hamiltonian function)。使用公式 (11.53)、(11.55)、(11.56) 和 (11.57)，我们现在可以将系统的动力学用哈密顿方程的形式表示出来，形式为

$$\frac{dz_i}{d\tau} = \frac{\partial H}{\partial r_i} \quad (11.58)$$

$$\frac{dr_i}{d\tau} = -\frac{\partial H}{\partial z_i} \quad (11.59)$$

在动态系统的变化过程中，哈密顿函数 H 的值是一个常数，这一点通过求微分的方式很容易看出来。

$$\begin{aligned} \frac{dH}{d\tau} &= \sum_i \left\{ \frac{\partial H}{\partial z_i} \frac{dz_i}{d\tau} + \frac{\partial H}{\partial r_i} \frac{dr_i}{d\tau} \right\} \\ &= \sum_i \left\{ \frac{\partial H}{\partial z_i} \frac{\partial H}{\partial r_i} - \frac{\partial H}{\partial r_i} \frac{\partial H}{\partial z_i} \right\} = 0 \end{aligned} \quad (11.60)$$

哈密顿动态系统的第二个重要性质是动态系统在相空间中体积不变，这被称为Liouville定理 (Liouville's Theorem)。换句话说，如果我们考虑变量 (\mathbf{z}, \mathbf{r}) 空间中的一个区域，那么当这个区域在哈密顿动态方程下的变化时，它的形状可能会改变，但是它的体积不会改变。可以这样证明：我们注意到流场 (位置在相空间的变化率) 为

$$\mathbf{V} = \left(\frac{d\mathbf{z}}{d\tau}, \frac{d\mathbf{r}}{d\tau} \right) \quad (11.61)$$

这个场的散度为零，即

$$\begin{aligned} \operatorname{div} \mathbf{V} &= \sum_i \left\{ \frac{\partial}{\partial z_i} \frac{dz_i}{d\tau} + \frac{\partial}{\partial r_i} \frac{dr_i}{d\tau} \right\} \\ &= \sum_i \left\{ +\frac{\partial}{\partial z_i} \frac{\partial H}{\partial r_i} - \frac{\partial}{\partial r_i} \frac{\partial H}{\partial z_i} \right\} = 0 \end{aligned} \quad (11.62)$$

现在考虑相空间上的联合概率分布，它的总能量是哈密顿函数，即概率分布的形式为

$$p(\mathbf{z}, \mathbf{r}) = \frac{1}{Z_H} \exp(-H(\mathbf{z}, \mathbf{r})) \quad (11.63)$$

使用体系的不变性和 H 的守恒性，可以看到哈密顿动态系统会使得 $p(\mathbf{z}, \mathbf{r})$ 保持不变。可以这样证明：考虑相空间的一个小区域，区域中 H 近似为常数。如果我们跟踪一段有限时间内的哈密顿方程的变化，那么这个区域的体积不会发生改变，从而这个区域的 H 的值不会发生改变，因此概率密度 (只是 H 的函数) 也不会改变。

虽然 H 是不变的，但是 \mathbf{z} 和 \mathbf{r} 会发生变换，因此通过在一个有限的时间间隔上对哈密顿动态系统积分，我们就可以让 \mathbf{z} 以一种系统化的方式发生较大的变化，避免了随机游走的行为。

然而，哈密顿动态系统的变化对 $p(\mathbf{z}, \mathbf{r})$ 的采样不具有各态历经性，因为 H 的值是一个常数。为了得到一个具有各态历经性的采样方法，我们可以在相空间中引入额外的移动，这些移动会改变 H 的值，同时也保持了概率分布 $p(\mathbf{z}, \mathbf{r})$ 的不变性。达到这个目标的最简单的方式是将 \mathbf{r} 的值

替换为一个从以 z 为条件的概率分布中抽取的样本。这可以被看成吉布斯采样的步骤，因此根据11.3节，我们看到这也使得所求的概率分布保持了不变性。注意， z 和 r 在概率分布 $p(z, r)$ 中是独立的，我们看到条件概率分布 $p(r | z)$ 是高斯分布，从中我们可以很容易地进行采样。

在这种方法的一个实际应用中，我们必须解决计算哈密顿方程的数值积分的问题。这会引入一些数值的误差，因此我们要设计一种方法来最小化这些误差产生的影响。事实上，可以证明，能够在Liouville定理仍然精确成立的条件下，对积分方法进行修改。这个性质在11.5.2节讨论混合蒙特卡罗算法时很重要。完成这件事的一种方法是蛙跳（leapfrog）离散化。这种方法使用下面的公式对位置变量和动量变量的离散时间近似 \hat{z} 和 \hat{r} 进行交替地更新。

$$\hat{r}_i \left(\tau + \frac{\epsilon}{2} \right) = \hat{r}_i(\tau) - \frac{\epsilon}{2} \frac{\partial E}{\partial z_i} (\hat{z}(r)) \quad (11.64)$$

$$\hat{z}_i(\tau + \epsilon) = \hat{z}_i(\tau) + \epsilon \hat{r}_i \left(\tau + \frac{\epsilon}{2} \right) \quad (11.65)$$

$$\hat{r}_i(\tau + \epsilon) = \hat{r}_i \left(\tau + \frac{\epsilon}{2} \right) - \frac{\epsilon}{2} \frac{\partial E}{\partial z_i} (\hat{z}(\tau + \epsilon)) \quad (11.66)$$

我们看到，这种方法对动量变量的更新形式是半步更新，步长为 $\frac{\epsilon}{2}$ ，接着是对位置变量的整步更新，步长为 ϵ ，然后是对动量变量的第二个半步更新。如果我们连续地使用几次蛙跳，那么可以看到，对动量变量的半步更新可以结合到步长为 ϵ 的整步更新中。于是，位置变量的更新和动量变量的更新互相之间以蛙跳的形式结合。为了将动态系统跨进一个时间间隔 τ ，我们需要进行 $\frac{\tau}{\epsilon}$ 个步骤。对连续时间动态系统的离散化近似引入的误差会在极限 $\epsilon \rightarrow 0$ 的情况下趋于零，假设函数 $E(Z)$ 是光滑的。然而，对于实际应用中使用的一个非零的 ϵ ，一些保留的误差仍然会存在。我们会在11.5.2节看到在混合蒙特卡罗算法中，这些误差的影响如何被消除。

总结一下，哈密顿动力学方法涉及到交替地进行一系列蛙跳更新以及根据动量变量的边缘分布进行重新采样。

注意，与基本的Metropolis方法不同，哈密顿动力学方法能够利用对数概率分布的梯度信息以及概率分布本身的信息。在函数最优化领域有一个类似的情形。大多数可以得到梯度信息的情况下，使用哈密顿动力学方法是很有优势的。非形式化地说，这种现象是由于下面的事实造成的：在 D 维空间中，与计算函数本身的代价相比，计算梯度所带来的额外的计算代价通常是一个与 D 无关的固定因子。而与函数本身只能传递一条信息相比， D 维梯度向量可以传递 D 条信息。

11.5.2 混合蒙特卡罗方法

正如我们在前一节讨论的那样，对于一个非零的步长 ϵ ，蛙跳算法的离散化会在哈密顿动力学方程的积分过程中引入误差。混合蒙特卡罗（hybrid Monte Carlo）（Duane et al., 1987; Neal, 1996）将哈密顿动态系统与Metropolis算法结合在一起，因此消除了与离散化过程关联的任何偏差。

具体来说，算法使用了一个马尔科夫链，它由对动量变量 r 的随机更新以及使用蛙跳算法对哈密顿动态系统的更新交替组成。在每次应用蛙跳算法之后，基于哈密顿函数 H 的值，确定Metropolis准则，确定生成的候选状态被接受或者拒绝。因此，如果 (z, r) 是初始状态， (z^*, r^*) 是蛙跳积分后的状态，那么候选状态被接受的概率为

$$\min(1, \exp\{H(z, r) - H(z^*, r^*)\}) \quad (11.67)$$

如果蛙跳积分完美地模拟了哈密顿动态系统，那么每个这种候选状态都会自动地被接受，因为 H 的值会保持不变。由于数值误差， H 的值有时可能会减小，因此我们希望Metropolis准则将这种效果引发的任何偏差都消除，并且确保得到的样本确实是所需的概率分布中抽取的。为了完成这件事，我们需要确保对应于蛙跳积分的更新方程满足细节平衡（11.40）。通过按照下面的方式修改蛙跳方法，这个目标很容易实现。

在开始蛙跳积分序列之前，我们等概率地随机选择是沿着时间向前的方向积分（步长为 ϵ ）还是沿着时间向后的方向积分（步长为 $-\epsilon$ ）。我们首先注意到，蛙跳积分方法（11.64）、（11.65）和（11.66）是时间可翻转的，即 L 步使用步长为 $-\epsilon$ 的积分会抵消 L 步使用步长为 ϵ 的积



图 11.14: 蛙跳算法 (11.64) 到 (11.66) 中的每一步修改位置变量 z_i 或者动量变量 r_i 中的一个。由于对一个变量的修改只是另一个变量的函数，因此相空间的任意区域在形变时不会改变体积。

分。接下来我们证明蛙跳积分精确地保持了相空间的体积不变性。这是因为，蛙跳方法中的每一步对 z_i 或者 r_i 的更新都只是另一个变量的函数。如图11.14所示，这个现象产生的效果是将相空间的一个区域进行形变而不改变它的体积。

最后，我们使用这些结果证明细节平衡是成立的。考虑相空间的一个小区域 \mathcal{R} ，它在 L 次步长为 ϵ 的蛙跳迭代序列之后被映射到了区域 \mathcal{R}' 。使用在蛙跳迭代下的体积的不变性，我们看到如果 \mathcal{R} 的体积为 δV ，那么 \mathcal{R}' 的体积也是。如果我们从概率分布 (11.63) 中选择一个初始点，然后使用 L 次蛙跳进行更新，那么从区域 \mathcal{R} 转移到 \mathcal{R}' 的概率为

$$\frac{1}{Z_H} \exp(-H(\mathcal{R})) \delta V \frac{1}{2} \min\{1, \exp(H(\mathcal{R}) - H(\mathcal{R}'))\} \quad (11.68)$$

其中，因子 $\frac{1}{2}$ 来自于选择用一个正的步长而不是负的步长进行积分的概率。类似地，从区域 \mathcal{R}' 开始，沿着时间的反方向回到区域 \mathcal{R} 的概率为

$$\frac{1}{Z_H} \exp(-H(\mathcal{R}')) \delta V \frac{1}{2} \min\{1, \exp(H(\mathcal{R}') - H(\mathcal{R}))\} \quad (11.69)$$

很容易看到，两个概率 (11.68) 和 (11.69) 是相等的，因此满足细节平衡。注意，这个证明忽略了区域 \mathcal{R} 和 \mathcal{R}' 之间有重叠的情况，但是很容易进行推广使其适用于这种存在重叠的情形。

不难构造蛙跳算法在有限次迭代之后返回起始点的例子。在这种情况下，每次蛙跳迭代之前对动量值的随机替换对确保各态历经性是不充分的，因为位置变量永远不会被更新。通过在蛙跳积分之前随机地从某个小区间中选择步长的大小，这种现象很容易避免。

通过考察混合蒙特卡罗算法在多元高斯分布上的应用，我们可以更深刻地理解算法的行为。为了方便，考虑具有独立分量的高斯分布 $p(\mathbf{z})$ ，它的哈密顿函数为

$$H(\mathbf{z}, \mathbf{r}) = \frac{1}{2} \sum_i \frac{1}{\sigma_i^2} z_i^2 + \frac{1}{2} \sum_i r_i^2 \quad (11.70)$$

我们的结论对于分量之间具有相关性的高斯分布同样适用，因为混合蒙特卡罗算法具有旋转不变性。在蛙跳积分阶段，每对相空间变量 z_i, r_i 独立变化。然而，对候选样本点接受或是拒绝基于的是 H 的值，它依赖于所有变量的值。因此，任何变量的一个较大的积分误差会产生一个较高的拒绝概率。为了让离散蛙跳积分对真实的连续时间动态系统产生一个较好的近似，有必要让蛙跳积分的标度 ϵ 小于势函数变化的最短的长度标度。这由 σ_i 的最小值控制，我们将其记作 σ_{\min} 。回忆一下，混合蒙特卡罗算法中的蛙跳积分的目标是在相空间中移动较大的距离到达新状态，这个新状态与初始状态相对独立，同时还能达到较高的接受率。为了实现这个目标，蛙跳积分必须连续进行多次迭代，迭代的次数是 $\sigma_{\max}/\sigma_{\min}$ 的量级。

相反，考虑之前讨论过的一个简单的Metropolis算法的行为，它具有各向同性的高斯提议分布，方差为 s^2 。为了避免高拒绝率， s 的值必须设置为 σ_{\min} 的量级。这样，对状态空间的探索通过随机游走的方式进行，达到近似独立的状态所需的步骤数是 $(\sigma_{\max}/\sigma_{\min})^2$ 量级的。

11.6 估计划分函数

正如我们已经看到的，本章中讨论的大部分采样算法只需要概率分布的函数形式，忽略一个可乘的常数。因此，如果我们有

$$p_E(z) = \frac{1}{Z_E} \exp(-E(z)) \quad (11.71)$$

那么为了从 $p(z)$ 中采样，归一化常数 Z_E 的值（也被称为划分函数）是不需要的。然而，关于 Z_E 的信息对于贝叶斯模型比较是有用的，因为它表示模型证据（即观测数据能够生成模型的概率），因此我们对它的值如何得到很感兴趣。我们假设在 z 的状态空间中，对函数 $\exp(-E(z))$ 求和或积分是不可行的。

对于模型比较来说，我们所需的实际是两个模型的划分函数的比值。将这个比值与先验概率的比值相乘可以得到后验概率的比值。之后可以用这个比值来进行模型选择或者模型平均。

一种估计划分函数比值的方法是使用概率分布的重要采样，这个概率分布的能量函数为 $G(z)$ ，即

$$\begin{aligned} \frac{Z_E}{Z_G} &= \frac{\sum_z \exp(-E(z))}{\sum_z \exp(-G(z))} \\ &= \frac{\sum_z \exp(-E(z) + G(z)) \exp(-G(z))}{\sum_z \exp(-G(z))} \\ &= \mathbb{E}_{G(z)}[\exp(-E + G)] \\ &\simeq \frac{1}{L} \sum_l \exp(-E(z^{(l)}) + G(z^{(l)})) \end{aligned} \quad (11.72)$$

其中 $\{z^{(l)}\}$ 是从 $p_G(z)$ 定义的概率分布中抽取的样本。如果概率分布 p_G 的划分函数可以解析地计算，例如它是一个高斯分布，那么 Z_E 的绝对值可以得到。

如果重要采样分布 p_G 很好地匹配概率分布 p_E ，即比值 p_E/p_G 变化不大，那么这种方法会产生准确的结果。在实际应用中，对于本书中考察的复杂的模型，我们无法找到一个可以很容易地解析计算的重要采样分布。

于是，另一种方法是使用从马尔科夫链中得到的样本来定义重要采样分布。如果马尔科夫链的转移概率为 $T(z, z')$ ，样本集合为 $z^{(1)}, \dots, z^{(L)}$ ，那么采样分布可以写成

$$\frac{1}{Z_G} \exp(-G(z)) = \frac{1}{L} \sum_{l=1}^L T(z^{(l)}, z) \quad (11.73)$$

这可以直接应用于公式 (11.72)。

计算两个划分函数的比值的方法需要对应的概率分布较好地匹配。如果我们希望找到一个复杂的概率分布的划分函数的绝对的值，那么这是一个很大的问题，因为只有对于相对简单的概率分布才能够直接计算划分函数，因此尝试直接估计划分函数的比值是无法完成的。使用链 (chaining) 方法，这个问题可以解决 (Neal, 1993; Barber and Bishop, 1997)。这种方法涉及到连续引入中间分布 p_2, \dots, p_{M-1} ，这些分布是在我们可以计算归一化系数 Z_1 的简单分布 $p_1(z)$ 和所求的复杂概率分布 $p_M(z)$ 之间进行的内插。于是我们有

$$\frac{Z_M}{Z_1} = \frac{Z_2}{Z_1} \frac{Z_3}{Z_2} \dots \frac{Z_M}{Z_{M-1}} \quad (11.74)$$

其中，中间的比值可以使用蒙特卡罗算法进行确定，与之前讨论的一样。一种建立中间系统序列的方法是使用一个包含连续参数 $0 \leq \alpha \leq 1$ 的势函数，在两个概率分布之间进行内插，即

$$E_\alpha(z) = (1 - \alpha)E_1(z) + \alpha E_M(z) \quad (11.75)$$

如果公式 (11.74) 中的中间比值使用蒙特卡罗算法得到，那使用一个单一的马尔科夫链可能要相对于每个比值都重新设置一个马尔科夫链的方式可能更高效。在这种情况下，马尔科夫链初始时设置为系统 p_1 ，然后在某个合适的迭代次数之后，移到序列中的下一个概率分布。然而需要注意的是，系统必须在每个阶段保持与均衡分布接近。



图 11.15: 两个变量 z_1 和 z_2 上的概率分布, 它在阴影区域上是均匀分布, 在其他地方概率为零。

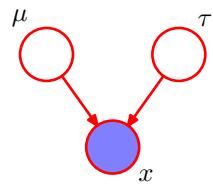


图 11.16: 涉及到一个高斯观测变量 x 的图模型, x 的先验概率分布的均值为 μ , 精度为 τ 。

11.7 练习

- (11.1) (*) 证明, 公式 (11.2) 定义的样本估计 \hat{f} 的均值为 $\mathbb{E}[f]$, 方差由 (11.3) 给定。
- (11.2) (*) 假设 z 是一个随机变量, 服从 $(0, 1)$ 上的均匀分布, 我们使用 $y = h^{-1}(z)$ 对 z 进行变换, 其中 $h(y)$ 由 (11.6) 定义。证明, y 的分布为 $p(y)$ 。
- (11.3) (*) 给定 $(0, 1)$ 上均匀分布的随机变量 z , 找到一个变换 $y = f(z)$, 使得 y 是由公式 (11.8) 给出的柯西分布。
- (11.4) (**) 假设 z_1 和 z_2 在单位圆上均匀分布, 如图11.3所示, 并且我们使用公式 (11.10) 和 (11.11) 进行变量替换。证明 (y_1, y_2) 服从公式 (11.12) 的概率分布。
- (11.5) (*) 令 z 是一个服从高斯分布的 D 维随机变量, 高斯分布的均值为零, 协方差矩阵是单位矩阵, 并且假设正定对称矩阵 Σ 具有Cholesky分解 $\Sigma = \mathbf{L}\mathbf{L}^T$, 其中 \mathbf{L} 是下三角矩阵 (即主对角线上方的元素全部为零)。证明, 变量 $y = \mu + \mathbf{L}z$ 服从高斯分布, 均值为 μ , 协方差为 Σ 。这提供了使用来自零均值单位方差的一元高斯分布的样本生成一般的多元高斯分布的方法。
- (11.6) (**) 本练习中, 我们更加详细地说明, 拒绝采样确实从所需的概率分布 $p(z)$ 中采样。假设提议分布是 $q(z)$, 证明样本值 z 被接受的概率为 $\frac{\tilde{p}(z)}{kq(z)}$, 其中 \tilde{p} 是任意的未归一化的分布, 正比于 $p(z)$, 常数 k 被设置为确保 $kq(z) \geq \tilde{p}(z)$ 对于所有 z 成立的最小值。注意, 抽取 z 值的概率等于从 $q(z)$ 中抽取那个值的概率乘以已知它被抽取的条件下接受这个值的概率。使用这一点, 以及概率的加和规则和乘积规则, 写出 z 上的概率分布的归一化形式, 证明它等于 $p(z)$ 。
- (11.7) (*) 假设 y 服从区间 $[0, 1]$ 上的均匀分布。证明变量 $z = b \tan y + c$ 服从 (11.16) 给出的柯西分布。
- (11.8) (**) 使用连续性和归一化的要求, 确定用于可调节拒绝采样的信封分布 (11.17) 的系数 k_i 。
- (11.9) (**) 通过使用11.1.1节讨论的从单一的指数分布中采样的方法, 设计一个从分段指数分布 (11.17) 中采样的算法。
- (11.10) (*) 证明, 由公式 (11.34)、(11.35) 和 (11.36) 定义的整数上的简单随机游走具有性质 $\mathbb{E}[(z^{(\tau)})^2] = \mathbb{E}[(z^{(\tau-1)})^2] + 1/2$, 从而根据归纳法, 具有性质 $\mathbb{E}[(z^{(\tau)})^2] = \tau/2$ 。
- (11.11) (**) 证明11.3节讨论的吉布斯采样算法满足 (11.40) 定义的细节平衡性质。
- (11.12) (*) 考虑图11.15所示的概率分布。讨论标准的吉布斯采样对于这个分布是否具有各态历经性, 是否可以正确地从这个分布中采样。

(11.13) (***) 考虑图 11.16 所示的简单的三结点图，其中观测结点 x 是一个高斯分布 $\mathcal{N}(x | \mu, \tau^{-1})$ ，均值为 μ ，精度为 τ 。假设均值和精度的边缘概率分布为 $\mathcal{N}(\mu | \mu_0, s_0)$ 和 $\text{Gam}(\tau | a, b)$ ，其中 $\text{Gam}(\cdot | \cdot, \cdot)$ 表示一个 Gamma 分布。写出为了将吉布斯采样方法应用到后验概率分布 $p(\mu, \tau | x)$ ，所需的条件概率分布 $p(\mu | x, \tau)$ 和 $p(\tau | x, \mu)$ 的表达式。

(11.14) (*) 验证过松弛更新 (11.50) 会得到均值为 μ_i 、方差为 σ_i^2 的值 z'_i 。公式 (11.50) 中， z_i 的均值为 μ_i ，方差为 σ_i^2 ， ν 的均值为零，方差是单位方差。

(11.15) (*) 使用公式 (11.56) 和 (11.57)，证明哈密顿方程 (11.58) 等价于 (11.53)。类似地，使用 (11.57) 证明 (11.59) 等价于 (11.55)。

(11.16) (*) 通过使用 (11.56)、(11.57) 和 (11.63)，证明条件概率分布 $p(r | z)$ 是一个高斯分布。

(11.17) (*) 验证两个概率 (11.68) 和 (11.69) 是相等的，从而细节平衡对于混合蒙特卡罗算法成立。

12 连续潜在变量

在第9章中，我们讨论了具有离散潜在变量的概率模型，例如高斯混合模型。我们现在研究某些潜在变量或者全部潜在变量为连续变量的模型。研究这种模型的一个重要的动机是许多数据集具有下面的性质：数据点几乎全部位于比原始数据空间的维度低得多的流形中。为了说明为什么会出现这种现象，考虑一个人造的数据集，这个数据集将一个 64×64 的灰度图像表示的手写数字嵌入到一个 100×100 的更大的图像中，用灰度值为零的像素（对应于白色像素）填充，并且数字的位置和方向被随机改变，如图12.1所示。每个生成的图像都可以表示为 $100 \times 100 = 10,000$ 维数据空间内的一个点。然而，对于这种图像的数据集，只有三个变化的自由度（degrees of freedom），对应于垂直平移、水平平移和旋转。于是，数据点会位于数据空间的一个子空间中，它的本质维度（intrinsic dimensionality）等于3。注意，这个流形是非线性的，因为例如如果我们把数字移过一个特定的像素，那么像素值会从0（白色）变为1（黑色），然后再回到0，这显然是数字位置的一个非线性函数。在这个例子中，平移和旋转变量是潜在变量，因为我们直观地看到图像向量，不知道创建它们所使用的平移或者旋转变量。

对于真实的手写数字图像数据，会存在另外一个自由度，这个自由度产生于图像的缩放。并且还会存在更多的自由度，这些自由度与更加复杂的形变有关，这些复杂的形变来自一个人的各次书写之间的变化，以及不同人之间的书写风格的差异。尽管这样，这种自由度与数据集的维度相比仍然很小。

另一个例子来源于石油流数据集，其中（对于给定的天然气、水、石油的几何配置）只有两个自由度，对应于管道中石油的比例和水的比例（之后就可以确定天然气的比例）。虽然数据空间由12个度量组成，但是一组数据点会近似位于这个空间内的一个二维流形当中。在这种情况下，流形由几个不同的片段组成，对应于不同的流的形式，每一个片段都是一个（带有噪声的）连续二维流形。如果我们的目标是数据压缩，或者对概率密度建模，那么利用这个流形结构是很有用的。

在实际应用中，数据点不会被精确限制在一个光滑的低维流形中，我们可以将数据点关于流形的偏移看做噪声。这就自然地引出了这种模型的生成式观点，其中我们首先根据某种潜在变量的概率分布在流形中选择一个点，然后通过添加噪声的方式生成观测数据点。噪声服从给定潜在变量下的数据变量的某个条件概率分布。

最简单的连续潜在变量模型对潜在变量和观测变量都作出了高斯分布的假设，并且使用了观测变量对于潜在变量状态的线性高斯依赖关系。这就引出了一个著名的技术——主成分分析（PCA）的概率表示形式，也引出了一个相关的模型，被称为因子分析。

本章中，我们首先介绍标准的、非概率的PCA方法，然后我们会说明，当求解线性高斯潜在变量模型的一种特别形式的最大似然解时，PCA如何自然地产生。这种概率形式的表示方法会带来很多好处，例如在参数估计时可以使用EM算法，对混合PCA模型的推广，以及主成分的数量可以从数据中自动确定的贝叶斯公式。最后，我们简短地讨论潜在变量概念的几个推广，使得潜在变量的概念不局限于线性高斯假设。这种推广包括非高斯潜在变量，它引出了独立成分分析（independent component analysis）的框架。这种推广还包括潜在变量与观测变量的关系是非线性关系的模型。

12.1 主成分分析

主成分分析，或者称为PCA，是一种被广泛使用的技术，应用的领域包括维度降低、有损数据压缩、特征抽取、数据可视化（Jolliffe, 2002）。它也被称为Karhunen-Loèv变换。

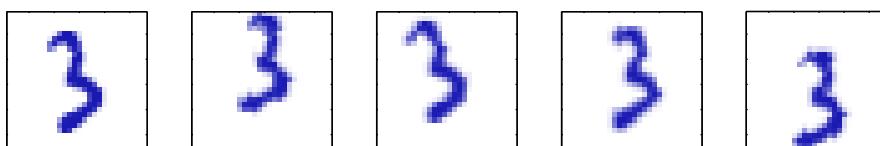


图 12.1：一个人工生成的数据集，以手写数字图像为输入，创建出多个副本，每个副本中，数字都在一个更大的图像中进行了一个随机的平移和旋转。每个生成的图像都有 $100 \times 100 = 10,000$ 个像素。

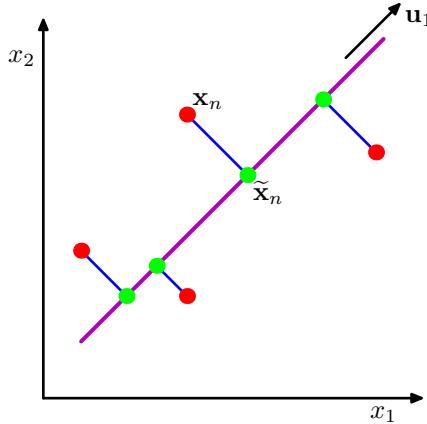


图 12.2: 主成分分析寻找一个低维空间, 被称为主子平面, 用紫色的线表示, 使得数据点(红点)在子空间上的正交投影能够最大化投影点(绿点)的方差。PCA 的另一个定义基于的是投影误差的平方和的最小值, 用蓝线表示。

有两种经常使用的PCA的定义, 它们会给出同样的算法。PCA可以被定义为数据在低维线性空间上的正交投影, 这个线性空间被称为主子空间(principal subspace), 使得投影数据的方差被最大化(Hotelling, 1933)。等价地, 它也可以被定义为使得平均投影代价最小的线性投影。平均投影代价是指数据点和它们的投影之间的平均平方距离(Pearson, 1901)。正交投影的过程如图12.2所示。我们依次讨论这些定义。

12.1.1 最大方差形式

考虑一组观测数据集 $\{x_n\}$, 其中 $n = 1, \dots, N$, 因此 x_n 是一个 D 维欧几里得空间中的变量。我们的目标是将数据投影到维度 $M < D$ 的空间中, 同时最大化投影数据的方差。现阶段, 我们假设 M 的值是给定的。稍后在本章中, 我们会研究从数据中确定合适的 M 值的方法。

首先, 考虑在一维空间($M = 1$)上的投影。我们可以使用 D 维向量 u_1 定义这个空间的方向。为了方便(并且不失一般性), 我们假定选择一个单位向量, 从而 $u_1^T u_1 = 1$ (注意, 我们只对 u_1 的方向感兴趣, 而对 u_1 本身的大小不感兴趣)。这样, 每个数据点 x_n 被投影到一个标量值 $u_1^T x_n$ 上。投影数据的均值是 $u_1^T \bar{x}$, 其中, \bar{x} 是样本集合的均值, 形式为

$$\bar{x} = \frac{1}{N} \sum_{n=1}^N x_n \quad (12.1)$$

投影数据的方差为

$$\frac{1}{N} \sum_{n=1}^N \{u_1^T x_n - u_1^T \bar{x}\}^2 = u_1^T S u_1 \quad (12.2)$$

其中 S 是数据的协方差矩阵, 定义为

$$S = \frac{1}{N} \sum_{n=1}^N (x_n - \bar{x})(x_n - \bar{x})^T \quad (12.3)$$

我们现在关于 u_1 最大化投影方差 $u_1^T S u_1$ 。很明显, 最化的过程必须满足一定的限制来防止 $\|u_1\| \rightarrow \infty$ 。恰当的限制来自归一化条件 $u_1^T u_1 = 1$ 。为了强制满足这个限制, 我们引入拉格朗日乘数, 记作 λ_1 , 然后对下式进行一个无限制的最大化

$$u_1^T S u_1 + \lambda_1 (1 - u_1^T u_1) \quad (12.4)$$

通过令它关于 u_1 的导数等于零, 我们看到驻点满足

$$S u_1 = \lambda_1 u_1 \quad (12.5)$$

这表明 \mathbf{u}_1 一定是 \mathbf{S} 的一个特征向量。如果我们左乘 \mathbf{u}_1^T , 使用 $\mathbf{u}_1^T \mathbf{u}_1 = 1$, 我们看到方差为

$$\mathbf{u}_1^T \mathbf{S} \mathbf{u}_1 = \lambda_1 \quad (12.6)$$

因此当我们把 \mathbf{u}_1 设置为与具有最大的特征值 λ_1 的特征向量相等时, 方差会达到最大值。这个特征向量被称为第一主成分。

我们可以用一种增量的方式定义额外的主成分, 方法为: 在所有与那些已经考虑过的方向正交的所有可能的方向中, 将新的方向选择为最大化投影方差的方向。如果我们考虑 M 维投影空间的一般情形, 那么最大化投影数据方差的最优线性投影由数据协方差矩阵 \mathbf{S} 的 M 个特征向量 $\mathbf{u}_1, \dots, \mathbf{u}_M$ 定义, 对应于 M 个最大的特征值 $\lambda_1, \dots, \lambda_M$ 。可以通过归纳法很容易地证明出来。

总结一下, 主成分分析涉及到计算数据集的均值 $\bar{\mathbf{x}}$ 和协方差矩阵 \mathbf{S} , 然后寻找 \mathbf{S} 的对应于 M 个最大特征值的 M 个特征向量。寻找特征值和特征向量的算法以及与特征向量分解相关的定理, 可以参考 Golub and Van Loan (1996)。注意, 计算一个 $D \times D$ 矩阵的完整的特征向量分解的代价为 $O(D^3)$ 。如果我们计划将我们的数据投影到前 M 个主成分中, 那么我们只需寻找前 M 个特征值和特征向量。这可以使用更高效的方法得到, 例如幂方法 (power method) (Golub and Van Loan, 1996), 它的时间复杂度为 $O(MD^2)$, 或者我们也可以使用 EM 算法。

12.1.2 最小误差形式

我们现在讨论 PCA 的另一种形式, 基于误差最小化的投影。为了完成这一点, 我们引入 D 维基向量的一个完整的单位正交集合 $\{\mathbf{u}_i\}$, 其中 $i = 1, \dots, D$, 且满足

$$\mathbf{u}_i^T \mathbf{u}_j = \delta_{ij} \quad (12.7)$$

由于基是完整的, 因此每个数据点可以精确地表示为基向量的一个线性组合, 即

$$\mathbf{x}_n = \sum_{i=1}^D \alpha_{ni} \mathbf{u}_i \quad (12.8)$$

其中, 系数 α_{ni} 对于不同的数据点来说是不同的。这对应于将坐标系旋转到了一个由 $\{\mathbf{u}_i\}$ 定义的新坐标系, 原始的 D 个分量 $\{x_{n1}, \dots, x_{nD}\}$ 被替换为一个等价的集合 $\{\alpha_{n1}, \dots, \alpha_{nD}\}$ 。与 \mathbf{u}_j 做内积, 然后使用单位正交性质, 我们有 $\alpha_{nj} = \mathbf{x}_n^T \mathbf{u}_j$, 因此不失一般性, 我们有

$$\mathbf{x}_n = \sum_{i=1}^D (\mathbf{x}_n^T \mathbf{u}_i) \mathbf{u}_i \quad (12.9)$$

然而, 我们的目标是使用限定数量 $M < D$ 个变量的一种表示方法来近似数据点, 这对应于在低维子空间上的一个投影。不失一般性, M 维线性子空间可以用前 M 个基向量表示, 因此我们可以用下式来近似每个数据点 \mathbf{x}_n

$$\tilde{\mathbf{x}}_n = \sum_{i=1}^M z_{ni} \mathbf{u}_i + \sum_{i=M+1}^D b_i \mathbf{u}_i \quad (12.10)$$

其中 $\{z_{ni}\}$ 依赖于特定的数据点, 而 $\{b_i\}$ 是常数, 对于所有数据点都相同。我们可以任意选择 $\{\mathbf{u}_i\}$, $\{z_{ni}\}$ 和 $\{b_i\}$, 从而最小化由维度降低所引入的失真。作为失真的度量, 我们使用原始数据点与它的近似点 $\tilde{\mathbf{x}}_n$ 之间的平方距离, 在数据集上取平均。因此我们的目标是最小化

$$J = \frac{1}{N} \sum_{n=1}^N \|\mathbf{x}_n - \tilde{\mathbf{x}}_n\|^2 \quad (12.11)$$

首先考虑关于 $\{z_{ni}\}$ 的最小化。消去 $\tilde{\mathbf{x}}_n$, 令它关于 z_{nj} 的导数为零, 然后使用单位正交的条件, 我们有

$$z_{nj} = \mathbf{x}_n^T \mathbf{u}_j \quad (12.12)$$

其中 $j = 1, \dots, M$ 。类似地，令 J 关于 b_i 的导数等于零，再次使用单位正交的关系，我们有

$$b_j = \bar{\mathbf{x}}^T \mathbf{u}_j \quad (12.13)$$

其中 $j = M + 1, \dots, D$ 。如果我们消去 (12.10) 中的 z_{ni} 和 b_i ，使用一般的展开式 (12.9)，我们有

$$\mathbf{x}_n - \tilde{\mathbf{x}}_n = \sum_{i=M+1}^D \{(\mathbf{x}_n - \bar{\mathbf{x}})^T \mathbf{u}_i\} \mathbf{u}_i \quad (12.14)$$

从中我们看到，从 \mathbf{x}_n 到 $\tilde{\mathbf{x}}_n$ 的位移向量位于与主子空间垂直的空间中，因为它是 $\{\mathbf{u}_i\}$ 的线性组合，其中 $i = M + 1, \dots, D$ ，如图 12.2 所示。这与预期相符，因为投影点 $\tilde{\mathbf{x}}_n$ 一定位于主子空间内，但是我们可以在那个子空间内自由移动投影点，因此最小的误差由正交投影给出。

于是，我们得到了失真度量 J 的表达式，它是一个纯粹的关于 $\{\mathbf{u}_i\}$ 的函数，形式为

$$J = \frac{1}{N} \sum_{n=1}^N \sum_{i=M+1}^D (\mathbf{x}_n^T \mathbf{u}_i - \bar{\mathbf{x}}^T \mathbf{u}_i)^2 = \sum_{i=M+1}^D \mathbf{u}_i^T \mathbf{S} \mathbf{u}_i \quad (12.15)$$

剩下的任务是关于 $\{\mathbf{u}_i\}$ 对 J 进行最小化，这必须是具有限制条件的最小化，因为如果不这样，我们会得到 $\mathbf{u}_i = 0$ 这一没有意义的结果。限制来自于单位正交条件，并且正如我们将看到的那样，解可以表示为协方差矩阵的特征向量展开式。在考虑一个形式化的解之前，让我们试着直观地考察一下这个结果。考虑二维数据空间 $D = 2$ 以及一维主子空间 $M = 1$ 的情形。我们必须选择一个方向 \mathbf{u}_2 来最小化 $J = \mathbf{u}_2^T \mathbf{S} \mathbf{u}_2$ ，同时满足限制条件 $\mathbf{u}_2^T \mathbf{u}_2 = 1$ 。使用拉格朗日乘数 λ_2 来强制满足这个限制，我们考虑最小化

$$\tilde{J} = \mathbf{u}_2^T \mathbf{S} \mathbf{u}_2 + \lambda_2 (1 - \mathbf{u}_2^T \mathbf{u}_2) \quad (12.16)$$

令关于 \mathbf{u}_2 的导数等于零，我们有 $\mathbf{S} \mathbf{u}_2 = \lambda_2 \mathbf{u}_2$ ，从而 \mathbf{u}_2 是 \mathbf{S} 的一个特征向量，且特征值为 λ_2 。因此任何特征向量都会定义失真度量的一个驻点。为了找到 J 在最小值点处的值，我们将 \mathbf{u}_2 的解代回到失真度量中，得到 $J = \lambda_2$ 。于是，我们通过将 \mathbf{u}_2 选择为对应于两个特征值中较小的那个特征值的特征向量，可以得到 J 的最小值。因此，我们应该将主子空间与具有较大的特征值的特征向量对齐。这个结果与我们的直觉相符，即为了最小化平均平方投影距离，我们应该将主成分子空间选为穿过数据点的均值并且与最大方差的方向对齐。对于特征值相等的情形，任何主方向的选择都会得到同样的 J 值。

对于任意的 D 和任意的 $M < D$ ，最小化 J 的一般解都可以通过将 $\{\mathbf{u}_i\}$ 选择为协方差矩阵的特征向量的方式得到，即

$$\mathbf{S} \mathbf{u}_i = \lambda_i \mathbf{u}_i \quad (12.17)$$

其中 $i = 1, \dots, D$ ，并且与平常一样，特征向量 $\{\mathbf{u}_i\}$ 被选为单位正交的。失真度量的对应的值为

$$J = \sum_{i=M+1}^D \lambda_i \quad (12.18)$$

这就是与主子空间正交的特征值的加和。于是，我们可以通过将这些特征向量选择成 $D - M$ 个最小的特征值对应的特征向量，来得到 J 的最小值，因此定义了主子空间的特征向量是对应于 M 个最大特征值的特征向量。

虽然我们已经考虑了 $M < D$ 的情形，但是 PCA 对于 $M = D$ 的情形仍然成立，这种情况下没有维度的降低，仅仅是将坐标轴旋转，与主成分对齐即可。

最后，值得注意的是，存在一个与此密切相关的线性维度降低的方法，被称为典型相关分析 (canonical correlation analysis)，或者 CCA (Hotelling, 1936; Bach and Jordan, 2002)。PCA 操作的对象是一个随机变量，而 CCA 考虑两个（或者更多）的变量，并且试图找到具有较高的交叉相关性的线性子空间对，从而在一个子空间中的每个分量都与另一个子空间的一个分量具有相关性。它的解可以表示为一般的特征向量问题。

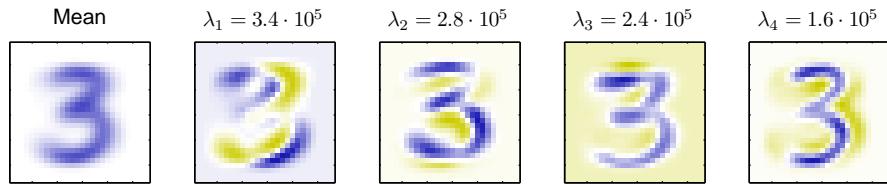


图 12.3: 对于离线手写数字数据集中的数字“3”，图中给出了均值向量 \bar{x} 以及前四个PCA特征向量 $\mathbf{u}_1, \dots, \mathbf{u}_4$ 还有对应的特征值。蓝色对应于正值，白色对应于零，黄色对应于负值。

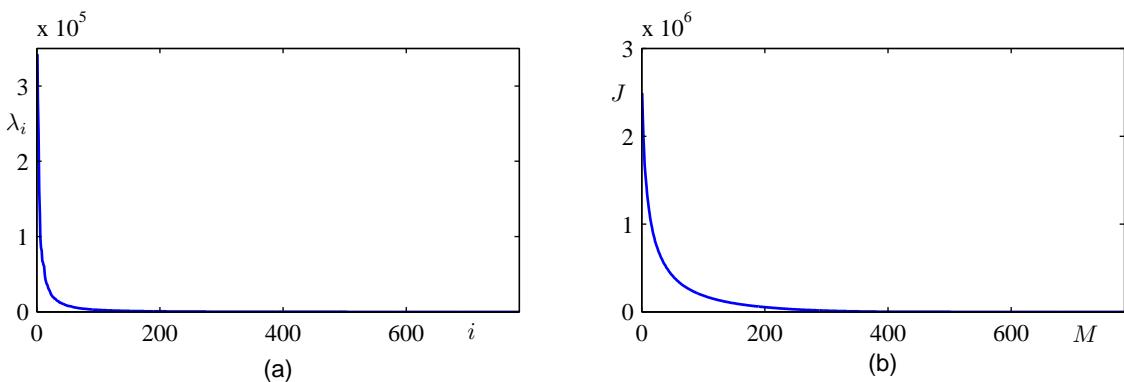


图 12.4: (a)对于离线手写数字数据集里的数字“3”的特征值谱线。(b)丢弃的特征值的加和的图像，它表示将数据投影到 M 维主成分空间中引入的平方和失真 J 。

12.1.3 PCA的应用

我们通过考虑离线手写数字数据集来说明PCA对于数据压缩的应用，其中我们关注与数字“3”的图像。由于协方差矩阵的每个特征向量是原始 D 维空间的一个向量，因此我们可以将特征向量表示为与数据点具有相同大小的图像。图12.3给出了前四个特征向量以及对应的特征值。完整的特征值的图像，按照降序排序，如图12.4(a)所示。选择 M 的一个特定的值造成的失真度量 J 由 $M+1$ 到 D 的特征值的求和给出。对于不同的 M 值，图像如图12.4(b)所示。

如果我们将公式 (12.12) 和 (12.13) 代入 (12.10)，我们可以写出对数据向量 x_n 的PCA近似，形式为

$$\tilde{x}_n = \sum_{i=1}^M (\mathbf{x}_n^T \mathbf{u}_i) \mathbf{u}_i + \sum_{i=M+1}^D (\bar{x}^T \mathbf{u}_i) \mathbf{u}_i \quad (12.19)$$

$$= \bar{x} + \sum_{i=1}^M (\mathbf{x}_n^T \mathbf{u}_i - \bar{x}^T \mathbf{u}_i) \mathbf{u}_i \quad (12.20)$$

其中我们使用了关系

$$\bar{x} = \sum_{i=1}^D (\bar{x}^T \mathbf{u}_i) \mathbf{u}_i \quad (12.21)$$

这个关系来自于 $\{\mathbf{u}_i\}$ 的完整性。这种方法表示了对数据集的一个压缩，因为对于每个数据点，我们将 D 维向量 \mathbf{x}_n 替换为 M 维向量，元素为 $(\mathbf{x}_n^T \mathbf{u}_i - \bar{x}^T \mathbf{u}_i)$ 。 M 的值越小，压缩的程度越大。对于手写数字数据集里的数字“3”的数据点，使用PCA重建的例子如图12.5所示。

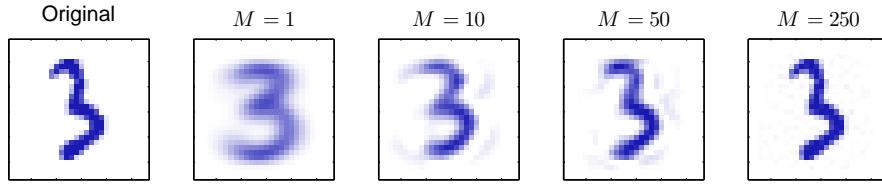


图 12.5: 来自离线手写数字数据集的原始样本，以及对于不同的 M 值，保留 M 个主成分得到的PCA重建。随着 M 的增加，重建变得越来越精确。当 $M = D = 28 \times 28 = 784$ 时，会得到一个完美的重建。

主成分分析的另一个应用是数据预处理。在这种情况下，目标不是维度降低，而是对数据集进行变换，使得数据集的某些属性得到标准化。这对于后续将模式识别算法成功应用于数据集来说很重要。通常，当原始变量使用不同的单位进行测量，或者变化情况相当不同的时候，我们会对数据集进行这样的变换。例如，在老忠实间歇喷泉数据集里，两次喷发的间隔时间通常要比喷发的持续时间大若干个数量级。当我们使用 K 均值算法应用于这个数据集时，我们首先对各个变量进行单独的重新标度，使得每个变量的均值为零，方差为单位方差。这被称为对数据的标准化（standardize），并且标准化的数据的协方差矩阵的元素为

$$\rho_{ij} = \frac{1}{N} \sum_{n=1}^N \frac{(x_{ni} - \bar{x}_i)}{\sigma_i} \frac{(x_{nj} - \bar{x}_j)}{\sigma_j} \quad (12.22)$$

其中 σ_i 是 x_i 的标准差。这被称为原始数据集的相关性矩阵（correlation matrix），具有下面的性质：如果数据的两个分量 x_i 和 x_j 完全相关，那么 $\rho_{ij} = 1$ ，如果它们不相关，那么 $\rho_{ij} = 0$ 。

然而，使用PCA，我们可以对数据进行更显著的归一化，得到零均值和单位方差的数据，从而不同的变量之间的相关性关系被消除。为了完成这一点，我们首先将特征向量方程 (12.17) 写成下面的形式

$$S\mathbf{U} = \mathbf{U}\mathbf{L} \quad (12.23)$$

其中， \mathbf{L} 是一个 $D \times D$ 的对角矩阵，元素为 λ_i ， \mathbf{U} 是一个 $D \times D$ 的正交矩阵，列为 \mathbf{u}_i 。然后对于每个数据点 \mathbf{x}_n ，我们定义一个变换，值为

$$\mathbf{y}_n = \mathbf{L}^{-\frac{1}{2}} \mathbf{U}^T (\mathbf{x}_n - \bar{\mathbf{x}}) \quad (12.24)$$

其中 $\bar{\mathbf{x}}$ 是公式 (12.1) 定义的样本均值。很明显，集合 $\{\mathbf{y}_n\}$ 的均值为零，协方差是单位矩阵，因为

$$\begin{aligned} \frac{1}{N} \sum_{n=1}^N \mathbf{y}_n \mathbf{y}_n^T &= \frac{1}{N} \sum_{n=1}^N \mathbf{L}^{-\frac{1}{2}} \mathbf{U}^T (\mathbf{x}_n - \bar{\mathbf{x}}) (\mathbf{x}_n - \bar{\mathbf{x}})^T \mathbf{U} \mathbf{L}^{-\frac{1}{2}} \\ &= \mathbf{L}^{-\frac{1}{2}} \mathbf{U}^T \mathbf{S} \mathbf{U} \mathbf{L}^{-\frac{1}{2}} = \mathbf{L}^{-\frac{1}{2}} \mathbf{L} \mathbf{L}^{-\frac{1}{2}} = \mathbf{I} \end{aligned} \quad (12.25)$$

这个操作被称为对数据的白化（whitening）或者球形化（spherizing）。图12.6使用老忠实间歇喷泉数据说明了这一点。

将PCA与4.1.4节讨论的Fisher线性判别分析进行对比是很有趣的。两种方法都可以看成线性维度降低的例子。然而，PCA是无监督的，值依赖于 \mathbf{x}_n 的值，而Fisher线性判别分析还使用了类别标签的信息。图12.7给出的例子强调了这个区别。

主成分分析的另一个常见应用是数据可视化。这里，每个数据点被投影到二维 ($M = 2$) 的主子空间中，从而数据点 \mathbf{x}_n 被画在了一个笛卡尔坐标系中，坐标系由 $\mathbf{x}_n^T \mathbf{u}_1$ 和 $\mathbf{x}_n^T \mathbf{u}_2$ 定义，其中 \mathbf{u}_1 和 \mathbf{u}_2 是特征向量，对应于最大的和第二大的特征值。对于石油流数据集，这种图的一个例子如图12.8所示。

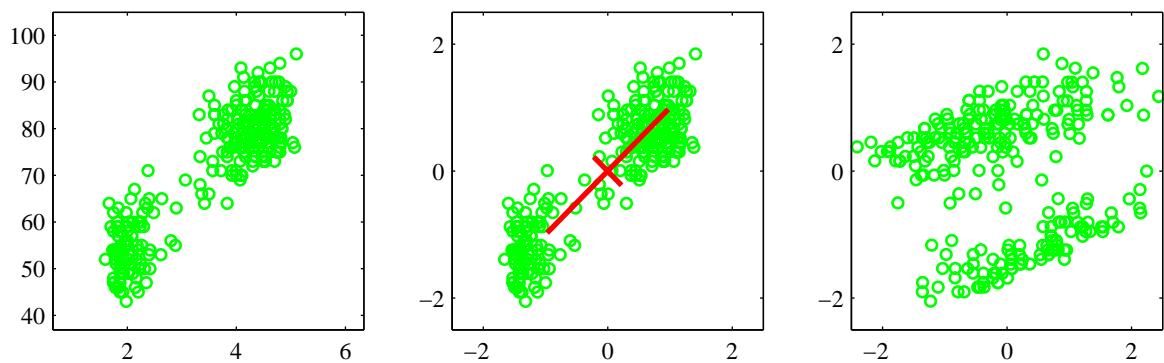


图 12.6: 对老忠实间歇喷泉数据集进行线性预处理的效果。左图给出的原始的数据。中图给出了将各个变量标准化为零均值单位方差的结果。同时画出的还有这个归一化数据集的主轴，画出了 $\pm \lambda_i^{1/2}$ 的范围。右图给出了对数据进行白化的结果，得到了零均值单位协方差的数据。

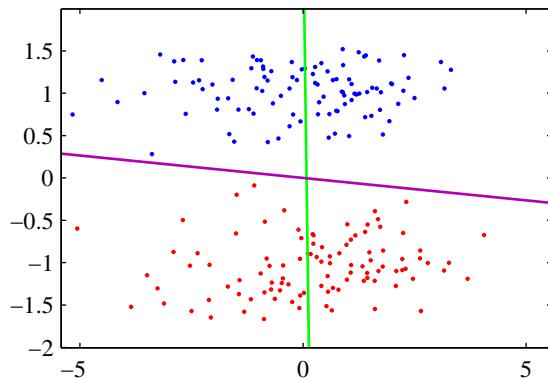


图 12.7: 用于线性维度降低的主成分分析与Fisher线性判别分析的对比。这里，数据位于二维空间中，属于两个类别，用红色和蓝色表示。数据要被投影到一维空间中。PCA选择了最大方差的方向，由紫色直线表示，它产生了严重的类别覆盖。而Fisher线性判别分析考虑类别标签，产生了在绿色直线上的投影。这种投影对类别的区分效果要好得多。

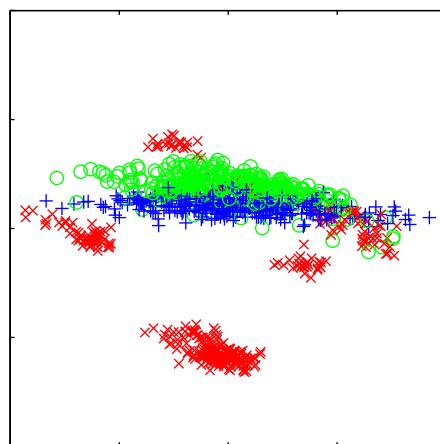


图 12.8: 石油流数据的可视化，通过将数据投影到前两个主成分上的方式实现。红色、蓝色和绿色点分别对应“薄片状”、“同质状”和“环状”的石油流配置。

12.1.4 高维数据的PCA

在主成分分析的一些应用中，数据点的数量小于数据空间的维度。例如，我们可能希望将PCA应用于由几百张图片组成的数据集，每个图片对应于几百万维（对应于图像中每个像素的三个颜色值）空间中的一个向量。注意，在一个 D 维空间中， N 个数据点 ($N < D$) 定义了一个线性子空间，它的维度最多为 $N - 1$ ，因此在使用PCA时，几乎没有 M 大于 $N - 1$ 的数据点。实际上，如果我们运行PCA，我们会发现至少 $D - N + 1$ 个特征值为零，对于沿着数据集的方差为零的方向的特征向量。此外，通常的寻找 $D \times D$ 矩阵的特征向量的算法的计算代价为 $O(D^3)$ ，因此对于诸如图像这种应用来说，直接应用PCA在计算上是不可行的。

我们可以这样解这个问题。首先，让我们将 \mathbf{X} 定义为 $(N \times D)$ 维中心数据矩阵，它的第 n 行为 $(\mathbf{x}_n - \bar{\mathbf{x}})^T$ 。这样，协方差矩阵 (12.3) 可以写成 $\mathbf{S} = N^{-1} \mathbf{X}^T \mathbf{X}$ ，对应的特征向量方程变成了

$$\frac{1}{N} \mathbf{X}^T \mathbf{X} \mathbf{u}_i = \lambda_i \mathbf{u}_i \quad (12.26)$$

现在，将两侧左乘 \mathbf{X} ，可得

$$\frac{1}{N} \mathbf{X} \mathbf{X}^T (\mathbf{X} \mathbf{u}_i) = \lambda_i (\mathbf{X} \mathbf{u}_i) \quad (12.27)$$

如果我们现在定义 $\mathbf{v}_i = \mathbf{X} \mathbf{u}_i$ ，那么我们有

$$\frac{1}{N} \mathbf{X} \mathbf{X}^T \mathbf{v}_i = \lambda_i \mathbf{v}_i \quad (12.28)$$

它是 $N \times N$ 矩阵 $N^{-1} \mathbf{X} \mathbf{X}^T$ 的一个特征向量方程。我们看到这个矩阵与原始的协方差矩阵具有相同的 $N - 1$ 个特征值，原始的协方差矩阵本身有额外的 $D - N + 1$ 个值为零的特征值。因此我们可以在低维空间中解决特征向量问题，计算代价为 $O(N^3)$ 而不是 $O(D^3)$ 。为了确定特征向量，我们将公式 (11.28) 两侧乘以 \mathbf{X}^T ，可得

$$\left(\frac{1}{N} \mathbf{X}^T \mathbf{X} \right) (\mathbf{X}^T \mathbf{v}_i) = \lambda_i (\mathbf{X}^T \mathbf{v}_i) \quad (12.29)$$

从中我们可以看到 $(\mathbf{X}^T \mathbf{v}_i)$ 是 \mathbf{S} 的一个特征向量，对应的特征值为 λ_i 。但是，需要注意，这些特征向量的长度未必等于1。为了确定合适的归一化，我们使用一个常数来对 $\mathbf{u}_i \propto \mathbf{X}^T \mathbf{v}_i$ 进行重新标度，使得 $\|\mathbf{u}_i\| = 1$ 。假设 \mathbf{v}_i 的长度已经被归一化，那么我们有

$$\mathbf{u}_i = \frac{1}{(N \lambda_i)^{\frac{1}{2}}} \mathbf{X}^T \mathbf{v}_i \quad (12.30)$$

总结一下，为了应用这种方法，我们首先计算 $\mathbf{X} \mathbf{X}^T$ ，然后找到它的特征向量和特征值，之后使用公式 (12.30) 计算原始数据空间的特征向量。

12.2 概率PCA

前一节讨论的PCA的形式所基于的是将数据线性投影到比原始数据空间维度更低的子空间内。我们现在说明，PCA也可以被视为概率潜在变量模型的最大似然解。PCA的这种形式，被称为概率PCA (probabilistic PCA)，与传统的PCA相比，会带来如下几个优势。

- 概率PCA表示高斯分布的一个限制形式，其中自由参数的数量可以受到限制，同时仍然使得模型能够描述数据集的主要的相关关系。
- 我们可以为PCA推导一个EM算法，这个算法在只有几个主要的特征向量需要求出的情况下，计算效率比较高，并且避免了计算数据协方差矩阵的中间步骤。
- 概率模型与EM的结合使得我们能够处理数据集里缺失值的问题。
- 概率PCA混合模型可以用一种有理有据的方式进行形式化，并且可以使用EM算法进行训练。

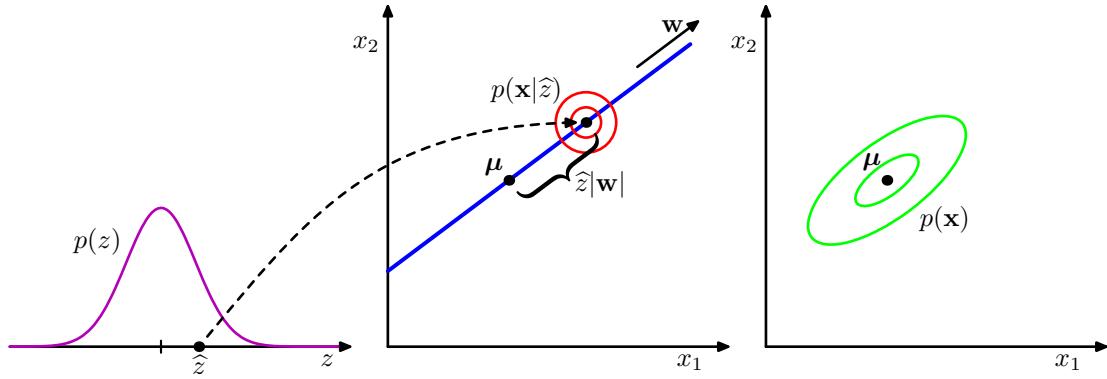


图 12.9: 概率PCA模型的生成式观点的说明, 数据空间为二维, 潜在空间为一维。一个观测数据点 x 的生成方式为: 首先从潜在变量的先验分布 $p(z)$ 中抽取一个潜在变量的值 \hat{z} , 然后从一个各向同性的高斯分布(用红色圆圈表示)中抽取一个 x 的值, 这个各向同性的高斯分布的均值为 $w\hat{z} + \mu$, 协方差为 $\sigma^2 I$ 。绿色椭圆画出了边缘概率分布 $p(x)$ 的密度轮廓线。

- 概率PCA构成了PCA的贝叶斯方法的基础, 其中主子空间的维度可以自动从数据中找到。
- 似然函数的存在使得直接与其他的概率密度模型进行对比成为可能。相反, 传统的PCA会将接近主子空间的数据点分配一个较低的重建代价, 即使这些数据点的位置距离训练数据任意远。
- 概率PCA可以被用来对类条件概率密度建模, 因此可以应用于分类问题。
- 概率PCA模型可以用一种生成式的方式运行, 从而可以按照某个概率分布生成样本。

这种概率模型形式的PCA由Tipping and Bishop (1997, 1999b) 和Roweis (1998) 独立提出。正如我们后面将会看到的那样, 它与因子分析(factor analysis)密切相关(Basilevsky, 1994)。

概率PCA是线性高斯框架的一个简单的例子, 其中所有的边缘概率分布和条件概率分布都是高斯分布。我们可以按照下面的方式建立概率PCA模型。首先显式地引入潜在变量 z , 对应于主成分子空间。接下来我们定义潜在变量上的一个高斯先验分布 $p(z)$ 以及以潜在变量的值为条件, 观测变量 x 的高斯条件概率分布 $p(x | z)$ 。具体来说, z 上的先验概率分布是一个零均值单位协方差的高斯分布

$$p(z) = \mathcal{N}(z | \mathbf{0}, I) \quad (12.31)$$

类似地, 以潜在变量 z 的值为条件, 观测变量 x 的条件概率分布还是高斯分布, 形式为

$$p(x | z) = \mathcal{N}(x | Wz + \mu, \sigma^2 I) \quad (12.32)$$

其中 x 的均值是 z 的一个一般的线性函数, 由 $D \times M$ 的矩阵 W 和 D 维向量 μ 控制。注意, 可以关于 x 的各个元素进行分解, 换句话说, 这是朴素贝叶斯模型的一个例子。正如我们稍后会看到的那样, W 的列张成了数据空间的一个线性子空间, 对应于主子空间。模型中的另一个参数 σ^2 控制了条件概率分布的方差。注意, 我们可以不失一般性地假设潜在变量分布 $p(z)$ 服从一个零均值单位协方差的高斯分布, 因为更一般的高斯分布会产生一个等价的概率模型。

我们可以从生成式的观点看待概率PCA模型, 其中观测值的一个采样值通过下面的方式获得: 首先为潜在变量选择一个值, 然后以这个潜在变量的值为条件, 对观测变量采样。具体来说, D 维观测变量 x 由 M 维潜在变量 z 的一个线性变换附加一个高斯“噪声”定义, 即

$$x = Wz + \mu + \epsilon \quad (12.33)$$

其中 z 是一个 M 维高斯潜在变量, ϵ 是一个 D 维零均值高斯分布的噪声变量, 协方差为 $\sigma^2 I$ 。这个生成式过程如图12.9所示。注意, 这个框架基于的是从潜在空间到数据空间的一个映射, 这与之前讨论的PCA的传统观点不同。从数据空间到潜在空间的逆映射可以通过使用贝叶斯定理的方式得到。

假设我们希望使用最大似然的方式确定参数 \mathbf{W} , $\boldsymbol{\mu}$ 和 σ^2 的值。为了写出似然函数的表达式，我们需要观测变量的边缘概率分布 $p(\mathbf{x})$ 的表达式。根据概率的加和规则和乘积规则，边缘概率分布的形式为

$$p(\mathbf{x}) = \int p(\mathbf{x} | \mathbf{z}) p(\mathbf{z}) d\mathbf{z} \quad (12.34)$$

由于这对应于一个线性高斯模型，因此边缘概率分布还是高斯分布，形式为

$$p(\mathbf{x}) = \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}, \mathbf{C}) \quad (12.35)$$

其中 $D \times D$ 协方差矩阵 \mathbf{C} 被定义为

$$\mathbf{C} = \mathbf{W}\mathbf{W}^T + \sigma^2 \mathbf{I} \quad (12.36)$$

这个结果也可以更直接地推导出来。我们注意到预测概率分布是高斯分布，然后使用公式 (12.33) 计算它的均值和协方差，结果为

$$\mathbb{E}[\mathbf{x}] = \mathbb{E}[\mathbf{W}\mathbf{z} + \boldsymbol{\mu} + \boldsymbol{\epsilon}] = \boldsymbol{\mu} \quad (12.37)$$

$$\begin{aligned} \text{cov}[\mathbf{x}] &= \mathbb{E}[(\mathbf{W}\mathbf{z} + \boldsymbol{\epsilon})(\mathbf{W}\mathbf{z} + \boldsymbol{\epsilon})^T] \\ &= \mathbb{E}[\mathbf{W}\mathbf{z}\mathbf{z}^T\mathbf{W}^T] + \mathbb{E}[\boldsymbol{\epsilon}\boldsymbol{\epsilon}^T] = \mathbf{W}\mathbf{W}^T + \sigma^2 \mathbf{I} \end{aligned} \quad (12.38)$$

其中我们使用了下面的事实： \mathbf{z} 和 $\boldsymbol{\epsilon}$ 是独立的随机变量，因此非相关。

直观地说，我们可以将概率分布 $p(\mathbf{x})$ 想象成由一个各向同性的高斯“喷雾罐”定义，然后将这个喷雾罐移过主子空间，喷射高斯分布的墨水，喷射的概率密度由 σ^2 定义，且权值为先验概率分布。累积的墨水密度产生了“薄煎饼”形状的概率分布，表示边缘概率密度 $p(\mathbf{x})$ 。

预测分布 $p(\mathbf{x})$ 由参数 $\boldsymbol{\mu}$, \mathbf{W} 和 σ^2 控制。然而，这些参数中存在冗余性，对应于潜在空间坐标的旋转。为了说明这一点，考虑一个矩阵 $\tilde{\mathbf{W}} = \mathbf{W}\mathbf{R}$ ，其中 \mathbf{R} 是一个正交矩阵。使用正交性质 $\mathbf{R}\mathbf{R}^T = \mathbf{I}$ ，我们看到协方差矩阵 \mathbf{C} 中的 $\tilde{\mathbf{W}}\tilde{\mathbf{W}}^T$ 的形式为

$$\tilde{\mathbf{W}}\tilde{\mathbf{W}}^T = \mathbf{W}\mathbf{R}\mathbf{R}^T\mathbf{W}^T = \mathbf{W}\mathbf{W}^T \quad (12.39)$$

因此与 \mathbf{R} 独立。从而有一大类的矩阵 $\tilde{\mathbf{W}}$ 会给出相同的预测分布。这种不变性可以理解为潜在空间中的旋转。我们稍后会回到对模型独立参数数量的讨论中。

当我们计算预测分布时，我们需要 \mathbf{C}^{-1} ，这涉及到对一个 $D \times D$ 的矩阵求逆。使用矩阵求逆的恒等式 (C.7)，所需的计算量可以被化简。使用这个矩阵恒等式得到的结果为

$$\mathbf{C}^{-1} = \sigma^{-2} \mathbf{I} - \sigma^{-2} \mathbf{W} \mathbf{M}^{-1} \mathbf{W}^T \quad (12.40)$$

其中 $M \times M$ 的矩阵 \mathbf{M} 的定义为

$$\mathbf{M} = \mathbf{W}^T \mathbf{W} + \sigma^2 \mathbf{I} \quad (12.41)$$

由于我们对 \mathbf{M} 进行求逆而不是直接对 \mathbf{C} 求逆，因此计算 \mathbf{C}^{-1} 从 $O(D^3)$ 减小到了 $O(M^3)$ 。

与预测分布 $p(\mathbf{x})$ 一样，我们也需要后验概率分布 $p(\mathbf{z} | \mathbf{x})$ ，这可以直接使用公式 (2.116) 给出的线性高斯模型的结果写出来，结果为

$$p(\mathbf{z} | \mathbf{x}) = \mathcal{N}(\mathbf{z} | \mathbf{M}^{-1} \mathbf{W}^T (\mathbf{x} - \boldsymbol{\mu}), \sigma^2 \mathbf{M}^{-1}) \quad (12.42)$$

注意，后验均值依赖于 \mathbf{x} ，而后验协方差与 \mathbf{x} 无关。

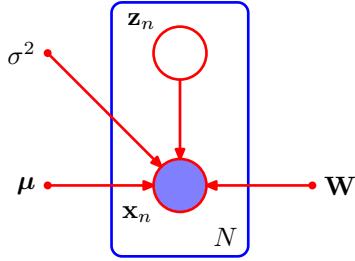


图 12.10: 对于观测变量 x 的 N 次观测组成的数据集, 概率PCA模型可以表示为一个有向图, 其中每个观测变量 x_n 与潜在变量的 z_n 的值相关联。

12.2.1 最大似然PCA

我们接下来考虑使用最大似然法确定模型的参数, 给定观测数据点的数据点 $\mathbf{X} = \{\mathbf{x}_n\}$, 概率PCA模型可以表示为一个有向图, 如图12.10所示。根据公式 (12.35) , 对应的对数似然函数为

$$\begin{aligned}\ln p(\mathbf{X} | \boldsymbol{\mu}, \mathbf{W}, \sigma^2) &= \sum_{n=1}^N \ln p(\mathbf{x}_n | \mathbf{W}, \boldsymbol{\mu}, \sigma^2) \\ &= -\frac{ND}{2} \ln(2\pi) - \frac{N}{2} \ln |\mathbf{C}| - \frac{1}{2} \sum_{n=1}^N (\mathbf{x}_n - \boldsymbol{\mu})^T \mathbf{C}^{-1} (\mathbf{x}_n - \boldsymbol{\mu})\end{aligned}\quad (12.43)$$

令似然函数关于 $\boldsymbol{\mu}$ 的导数等于零, 可以得到预期的结果 $\boldsymbol{\mu} = \bar{\mathbf{x}}$, 其中 $\bar{\mathbf{x}}$ 是公式 (12.1) 定义的数据均值。代回到似然函数中, 我们有

$$\ln p(\mathbf{X} | \mathbf{W}, \boldsymbol{\mu}, \sigma^2) = -\frac{N}{2} \{ D \ln(2\pi) + \ln |\mathbf{C}| + \text{Tr}(\mathbf{C}^{-1} \mathbf{S}) \} \quad (12.44)$$

其中 \mathbf{S} 是由公式 (12.3) 定义的协方差矩阵。由于对数似然函数是 $\boldsymbol{\mu}$ 的二次函数, 因此解具有唯一的大值, 可以通过计算二阶导数的方式验证这一点。

关于 \mathbf{W} 和 σ^2 的最大化更复杂, 但是尽管这样, 它们还是有一个近似的封闭解。Tipping and Bishop (1999b) 证明, 对数似然函数的所有驻点都可以写成

$$\mathbf{W}_{ML} = \mathbf{U}_M (\mathbf{L}_M - \sigma^2 \mathbf{I})^{\frac{1}{2}} \mathbf{R} \quad (12.45)$$

其中 \mathbf{U}_M 是一个 $D \times M$ 的矩阵, 它的列由数据协方差矩阵 \mathbf{S} 的特征向量的任意 (大小为 M 的) 子集给定。 $M \times M$ 的对角矩阵 \mathbf{L}_M 的元素是对应的特征值 λ_i , \mathbf{R} 是一个任意的 $M \times M$ 的正交矩阵。

此外, Tipping and Bishop (1999b) 证明, 当 M 个特征向量被选为前 M 个最大的特征值所对应的特征向量时, 对数似然函数可以达到最大值, 其他所有的解都是鞍点。类似的结果由Roweis (1998) 独立地提出猜想, 但是未给出证明。与之前一样, 我们假定特征向量按照对应的特征值的大小降序排列, 从而 M 个主特征向量是 $\mathbf{u}_1, \dots, \mathbf{u}_M$ 。在这种情况下, \mathbf{W} 的列定义了标准PCA的主子空间。这样, σ^2 的对应的最大似然解为

$$\sigma_{ML}^2 = \frac{1}{D-M} \sum_{i=M+1}^D \lambda_i \quad (12.46)$$

从而 σ_{ML}^2 是与丢弃的维度相关的平均方差。

由于 \mathbf{R} 是正交的, 因此它可以被看做是 M 维潜在空间中的一个旋转矩阵。如果我们将 \mathbf{W} 的解代入到 \mathbf{C} 的表达式中, 然后使用正交性质 $\mathbf{R}\mathbf{R}^T = \mathbf{I}$, 那么我们看到 \mathbf{C} 与 \mathbf{R} 无关。这表明, 与之前讨论的一样, 预测概率分布在潜在空间中具有旋转不变性。对于 $\mathbf{R} = \mathbf{I}$ 这一特定情形, 我们看到 \mathbf{W} 的列是主成分特征向量, 由方差参数的平方根 $\sqrt{\lambda_i - \sigma^2}$ 进行缩放。一旦我们认识到对

于独立高斯分布（本例中的潜在空间分布和噪声模型）的卷积来说，方差是可加的，那么这些放缩因子的意义就很明显了。因此，在特征向量 \mathbf{u}_i 方向上的方差 λ_i 由两部分相加得到，一部分来自于从单位方差潜在空间分布通过对对应的 \mathbf{W} 的列向数据空间投影的贡献 $\lambda_i - \sigma^2$ ，另一部分来自于在噪声模型的所有方向上相加的各项同性的方差的贡献 σ^2 。

值得花一些时间研究一下公式 (12.36) 给出的协方差矩阵的形式。考虑预测分布在由单位向量 \mathbf{v} 指定的方向上的方差，其中 $\mathbf{v}^T \mathbf{v} = 1$ ，这个方差为 $\mathbf{v}^T \mathbf{C} \mathbf{v}$ 。首先假设 \mathbf{v} 与主子空间正交，即它等于被丢弃的特征向量的某个线性组合。那么 $\mathbf{v}^T \mathbf{U} = \mathbf{0}$ ，因此 $\mathbf{v}^T \mathbf{C} \mathbf{v} = \sigma^2$ 。所以模型预测了一个噪声方差正交于主子空间。根据公式 (12.46)，这个方差就是丢弃的特征值的平均值。现在假设 $\mathbf{v} = \mathbf{u}_i$ ，其中 \mathbf{u}_i 是一个定义了主子空间的特征向量。那么 $\mathbf{v}^T \mathbf{C} \mathbf{v} = (\lambda_i - \sigma^2) + \sigma^2 = \lambda_i$ 。换句话说，这个模型正确地描述了数据沿着主轴方向的方差，并且用一个单一的均值 σ^2 近似了所有剩余方向上的方差。

一种建立最大似然密度模型的方式是寻找数据协方差矩阵的特征值和特征向量，然后使用上面的结果计算 \mathbf{W} 和 σ^2 。在这种情况下，为了方便，我们会选择 $\mathbf{R} = \mathbf{I}$ 。然而，如果最大似然解通过对似然函数的数值最优化的方式得到，例如使用诸如共轭梯度法 (Fletcher, 1987; Nocedal and Wright, 1999; Bishop and Nabney, 2008) 或者EM算法，那么得到的 \mathbf{R} 值就可能是任意的了。这表明 \mathbf{W} 的列不必是正交的。如果我们需要一组正交的基，那么矩阵 \mathbf{W} 可以进行恰当的后处理 (Golub and Van Loan, 1996)。此外，EM算法可以进行修改，直接产生单位正交的主方向，按照对应的特征值降序排序 (Ahn and Oh, 2003)。

潜在空间中的旋转不变性代表了一种形式的统计不可区分性，类似于我们在离散潜在变量的混合模型中遇到的情形。这里，有一组连续的参数会产生同样的预测密度，这不同于与混合模型中的分量重新标注相关联的离散不可区分性。

如果我们考虑 $M = D$ 的情形，从而不存在维度的降低，那么 $\mathbf{U}_M = \mathbf{U}$ 且 $\mathbf{L}_M = \mathbf{L}$ 。使用正交的性质 $\mathbf{U}\mathbf{U}^T = \mathbf{I}$ 以及 $\mathbf{R}\mathbf{R}^T = \mathbf{I}$ ，我们看到 \mathbf{x} 的边缘概率分布的协方差 \mathbf{C} 变成了

$$\mathbf{C} = \mathbf{U}(\mathbf{L} - \sigma^2 \mathbf{I})^{\frac{1}{2}} \mathbf{R} \mathbf{R}^T (\mathbf{L} - \sigma^2 \mathbf{I})^{\frac{1}{2}} \mathbf{U}^T + \sigma^2 \mathbf{I} = \mathbf{U} \mathbf{L} \mathbf{U}^T = \mathbf{S} \quad (12.47)$$

因此我们得到了无限制高斯分布的标准的最大似然解，其中协方差矩阵是样本的协方差。

传统的PCA通常的形式是 D 维空间的数据点在 M 维线性子空间上的投影。然而，概率PCA可以很自然地表示为从潜在空间到数据空间的映射，由公式 (12.33) 给出。对于数据可视化和数据压缩之类的应用，我们可以使用贝叶斯定理将这个映射取逆。这样，任何在数据空间中的点 \mathbf{x} 都可以使用潜在空间中的后验均值和方差进行概括。根据公式 (12.42)，均值为

$$\mathbb{E}[\mathbf{z} | \mathbf{x}] = \mathbf{M}^{-1} \mathbf{W}_{ML}^T (\mathbf{x} - \bar{\mathbf{x}}) \quad (12.48)$$

其中 \mathbf{M} 由公式 (12.41) 给出。它到数据空间的一个点的投影为

$$\mathbf{W} \mathbb{E}[\mathbf{z} | \mathbf{x}] + \boldsymbol{\mu} \quad (12.49)$$

注意，这与正则化的线性回归方程的形式相同，结果是最大化了线性高斯模型的对数似然函数。类似地，公式 (12.42) 的后验协方差为 $\sigma^2 \mathbf{M}^{-1}$ ，与 \mathbf{x} 无关。

如果我们取极限 $\sigma^2 \rightarrow 0$ ，那么后验均值为

$$(\mathbf{W}_{ML}^T \mathbf{W}_{ML})^{-1} \mathbf{W}_{ML}^T (\mathbf{x} - \bar{\mathbf{x}}) \quad (12.50)$$

这表示数据点在潜在空间上的正交投影，因此我们就恢复出了标准的PCA模型。然而在这种极限情况下，后验协方差是零，概率密度变得奇异。对于 $\sigma^2 > 0$ 的情形，潜在投影与正交投影相比，会向原点方向偏移。

最后，我们注意到，概率PCA模型在定义多元高斯分布时具有重要的作用，其中自由度的数量（即独立参数的数量）可以进行控制，同时仍然使得模型能够描述数据中的主要的相关关系。回忆一下，一个一般的高斯分布在协方差矩阵中有 $\frac{D(D+1)}{2}$ 个独立的参数（加上均值中的另外 D 个参数）。因此参数的数量随着 D 以二次函数的方式增多，从而在高位空间中变得无法处理。如果我们将协方差矩阵限制为对角化，那么它只有 D 个独立的参数，从而此时参数的数量随着维度线性增长。然而，现在它对变量的处理方式类似于将变量看成是独立的，从而无法表

达变量之间的相关性关系。概率PCA提供了一种优雅的折中方式，它能够描述 M 个最显著的相关性关系，同时使得参数的总数随着 D 线性增长。我们可以通过计算概率PCA模型的自由度的数量来理解这一点，如下所述。协方差矩阵 C 依赖于参数 \mathbf{W} （大小为 $D \times M$ ）和 σ^2 ，从而总的参数数量为 $DM + 1$ 。然而，我们已经看到参数中存在一些与潜在空间坐标系的旋转相关联的冗余性。表示这种旋转的正交矩阵 \mathbf{R} 的大小为 $M \times M$ 。这个矩阵的第一列有 $M - 1$ 个独立的参数，因为列向量必须归一化到单位长度，第二列有 $M - 2$ 个独立的参数，因为列向量必须被归一化，并且必须与前一列垂直，以此类推。对这个算术序列求和，我们看到 \mathbf{R} 总共有 $\frac{M(M-1)}{2}$ 个独立参数。因此协方差矩阵 C 的自由度的数量为

$$DM + 1 - \frac{M(M-1)}{2} \quad (12.51)$$

于是，对于固定的 M ，这个模型中的独立参数的数量随着 D 只是线性增长关系。如果我们令 $M = D - 1$ ，那么我们就恢复出了高斯分布的完整的协方差矩阵的标准结果。在这种情况下，沿着 $D - 1$ 个线性独立方向的方差由 \mathbf{W} 的列所控制，沿着剩余方向的方差由 σ^2 控制。如果 $M = 0$ ，那么模型等价于各向同性协方差的情形。

12.2.2 用于PCA的EM算法

正如我们已经看到的那样，概率PCA模型可以根据连续潜在空间 \mathbf{z} 上的积分或求和来表示，其中对于每个数据点 \mathbf{x}_n ，都存在一个对应的潜在变量 \mathbf{z}_n 。于是，我们可以使用EM算法来找到模型参数。这看起来似乎相当没有意义，因为我们已经得到了最大似然参数值的一个精确的解析解。然而，在高维空间中，使用迭代的EM算法而不是直接计算样本协方差矩阵可能会有一些计算上的优势。这个EM的求解步骤也可以推广到因子分析模型中，那里不存在解析解。最后，它使得我们可以用一种有理有据的方式处理缺失的数据。

我们可以使用一般的EM框架来推导用于概率PCA的EM算法。因此，我们写出完整数据对数似然函数，然后关于使用旧的参数值计算的潜在变量的后验概率分布求期望。最大化完整数据对数似然函数的期望就会产生新的参数值。因为我们假定数据点是独立的，因此完整数据对数似然函数的形式为

$$\ln p(\mathbf{X}, \mathbf{Z} | \boldsymbol{\mu}, \mathbf{W}, \sigma^2) = \sum_{n=1}^N \{\ln p(\mathbf{x}_n | \mathbf{z}_n) + \ln p(\mathbf{z}_n)\} \quad (12.52)$$

其中矩阵 \mathbf{Z} 的第 n 行由 \mathbf{z}_n 给出。我们已经知道 $\boldsymbol{\mu}$ 的精确的最大似然解是公式 (12.1) 定义的样本均值 $\bar{\mathbf{x}}$ 。在这个阶段将 $\boldsymbol{\mu}$ 替换掉是比较方便的。分别使用公式 (12.31) 和 (12.32) 给出的潜在概率分布和条件概率分布的表达式，然后关于潜在变量上的后验概率分布求期望，我们有

$$\begin{aligned} \mathbb{E}[\ln p(\mathbf{X}, \mathbf{Z} | \boldsymbol{\mu}, \mathbf{W}, \sigma^2)] &= - \sum_{n=1}^N \left\{ \frac{D}{2} \ln(2\pi\sigma^2) + \frac{1}{2} \text{Tr}(\mathbb{E}[\mathbf{z}_n \mathbf{z}_n^T]) \right. \\ &\quad + \frac{1}{2\sigma^2} \|\mathbf{x}_n - \boldsymbol{\mu}\|^2 - \frac{1}{\sigma^2} \mathbb{E}[\mathbf{z}_n]^T \mathbf{W}^T (\mathbf{x}_n - \boldsymbol{\mu}) \\ &\quad \left. + \frac{1}{2\sigma^2} \text{Tr}(\mathbb{E}[\mathbf{z}_n \mathbf{z}_n^T] \mathbf{W}^T \mathbf{W}) + \frac{M}{2} \ln(2\pi) \right\} \end{aligned} \quad (12.53)$$

注意，上式仅仅通过高斯分布的充分统计量对后验概率分布产生依赖。因此在E步骤中，我们使用旧的参数计算

$$\mathbb{E}[\mathbf{z}_n] = \mathbf{M}^{-1} \mathbf{W}^T (\mathbf{x}_n - \bar{\mathbf{x}}) \quad (12.54)$$

$$\mathbb{E}[\mathbf{z}_n \mathbf{z}_n^T] = \sigma^2 \mathbf{M}^{-1} + \mathbb{E}[\mathbf{z}_n] \mathbb{E}[\mathbf{z}_n]^T \quad (12.55)$$

这可以直接从后验概率分布 (12.42) 以及标准的结果 $\mathbb{E}[\mathbf{z}_n \mathbf{z}_n^T] = \text{cov}[\mathbf{z}_n] + \mathbb{E}[\mathbf{z}_n] \mathbb{E}[\mathbf{z}_n]^T$ 中得出。这里， \mathbf{M} 由公式 (12.41) 定义。

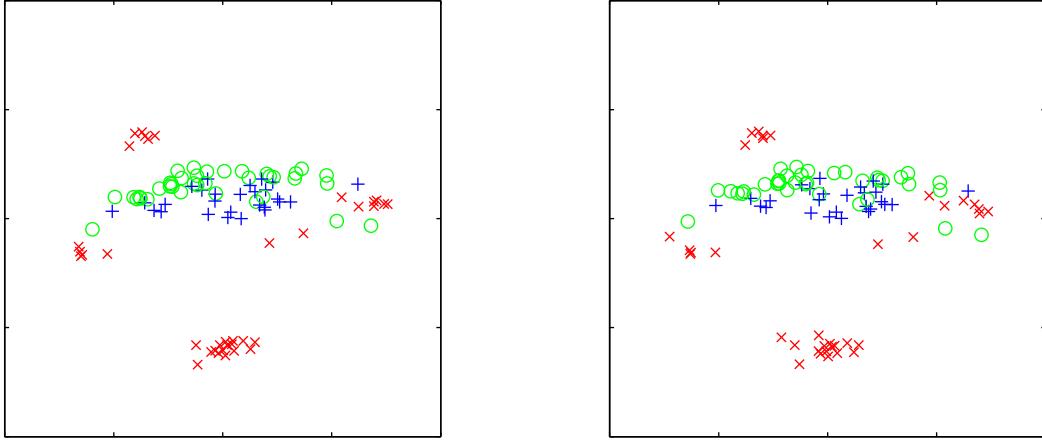


图 12.11: 概率PCA用于将石油流数据集的前100个数据点进行可视化。左图给出了数据点在主子空间上的后验均值投影。右图的获得方式是：首先随机略去30%的观测值，然后使用EM来处理缺失值。注意，每个数据点之后有至少一个缺失的度量，但是得到的图像与没有缺失值的图像相当相似。

在M步骤中，我们关于 \mathbf{W} 和 σ^2 进行最大化，保持后验统计量固定。关于 σ^2 的最大化很容易。对于关于 \mathbf{W} 的最大化，我们可以使用 (C.24)。求得的M步骤方程为

$$\mathbf{W}_{\text{新}} = \left[\sum_{n=1}^N (\mathbf{x}_n - \bar{\mathbf{x}}) \mathbb{E}[\mathbf{z}_n]^T \right] \left[\sum_{n=1}^N \mathbb{E}[\mathbf{z}_n \mathbf{z}_n^T] \right]^{-1} \quad (12.56)$$

$$\begin{aligned} \sigma_{\text{新}}^2 &= \frac{1}{ND} \sum_{n=1}^N \{ \| \mathbf{x}_n - \bar{\mathbf{x}} \|^2 - 2 \mathbb{E}[\mathbf{z}_n]^T \mathbf{W}_{\text{新}}^T (\mathbf{x}_n - \bar{\mathbf{x}}) \\ &\quad + \text{Tr}(\mathbb{E}[\mathbf{z}_n \mathbf{z}_n^T] \mathbf{W}_{\text{新}}^T \mathbf{W}_{\text{新}}) \} \end{aligned} \quad (12.57)$$

概率PCA的EM算法的执行过程为：对参数进行初始化，然后交替地在E步骤中使用公式 (12.54) 和 (12.55) 计算潜在空间的后验概率分布的充分统计量，以及在M步骤中使用公式 (12.56) 和 (12.57) 来修正参数的值。

用于PCA的EM算法的一个好处是对于大规模应用的计算效率 (Roweis, 1998)。与传统的基于样本协方差矩阵的特征向量分解的PCA不同，EM算法时迭代的，因此似乎没有什么吸引力。然而，在高维空间中，EM算法的每次迭代所需的计算量都要比传统的PCA小得多。为了说明这一点，我们注意到，对协方差矩阵的特征分解的计算复杂度为 $O(D^3)$ 。通常我们只对前 M 个特征向量和它们的特征值感兴趣，这种情况下我们可以使用 $O(MD^2)$ 的算法。然而，计算协方差矩阵本身需要 $O(ND^2)$ 的计算量，其中 N 是数据点的数量。有一些能够避免直接计算协方差矩阵的算法，例如快照方法 (snapshot method) (Sirovich, 1987) 假设特征向量是数据向量的线性组合，但是这种算法的计算复杂度为 $O(N^3)$ ，因此不适用于大规模数据。这里描述的EM算法也没有显式地建立协方差矩阵。相反，计算量最大的步骤是涉及到对数据集求和的操作，计算代价为 $O(NDM)$ 。对于较大的 D ， $M \ll D$ ，这与 $O(ND^2)$ 相比，计算量极大地降低，因此可以抵消EM算法的迭代本质。

注意，这个EM算法可以用一种在线的形式执行，其中每个 D 维数据点被读入、处理，然后在处理下一个数据点之前丢弃这个数据点。为了说明这一点，注意在E步骤中需要计算的量（一个 M 维向量和一个 $M \times M$ 的矩阵）可以分别对每个数据点单独计算，在M步骤中，我们需要在数据点上累积求和，这个可以增量地完成。如果 N 和 D 都很大，那么这种方法会很有优势。

由于我们现在对PCA建立了一个完全的概率模型，因此我们可以通过对未观测变量进行积分或求和的方式，处理缺失的数据，假设数据的缺失是随机的。与之前一样，这些缺失值可以使用EM算法进行处理。我们在图12.11中给出了使用这种方法进行数据可视化的一个例子。

EM算法的另一个特征是，我们可以取极限 $\sigma^2 \rightarrow 0$ ，对应于标准的PCA，仍然可以得到一个合法的类似EM的算法 (Roweis, 1998)。根据公式 (12.55)，我们看到我们在E步骤中需要计算的唯一的量是 $\mathbb{E}[\mathbf{z}_n]$ 。此外，M步骤可以得到简化，因为 $\mathbf{M} = \mathbf{W}^T \mathbf{W}$ 。为了强调算法的

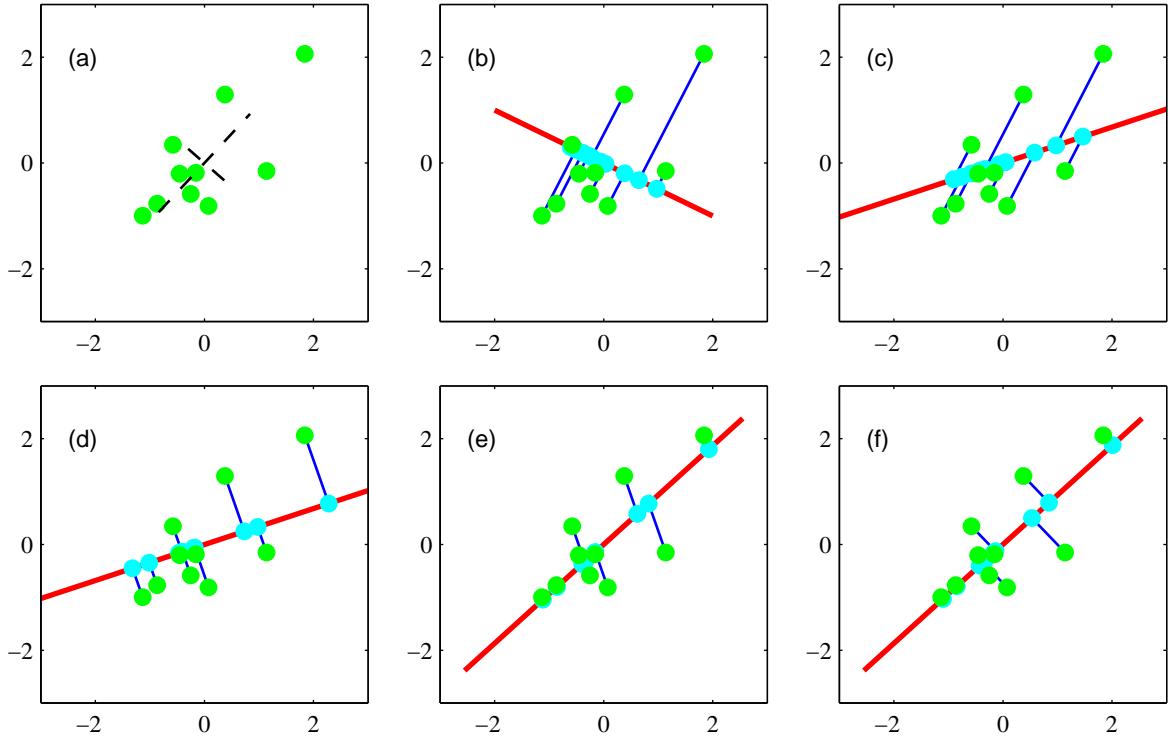


图 12.12: 人工生成的数据用来说明公式 (12.58) 和 (12.59) 定义的用于PCA的EM算法。(a)一个数据集合 \mathbf{X} , 数据点用绿色表示。同时画出了真实的主成分 (用特征向量表示, 使用特征值的平方根进行了缩放)。(b)由 \mathbf{W} 定义的主子空间的初始配置, 用红色表示。同时画出了 $\mathbf{Z}\mathbf{W}^T$ 给出的潜在点 \mathbf{Z} 在数据空间上的投影, 用青色表示。(c)在一次M步骤之后, 潜在空间被更新, 保持 \mathbf{Z} 固定。(d)在接下来的E步骤之后, \mathbf{Z} 的值被更新, 得到了正交投影, 保持 \mathbf{W} 固定。(e)在第二个M步骤之后的结果。(f)收敛的解。

简化, 让我们将 $\tilde{\mathbf{X}}$ 定义为一个 $N \times D$ 的矩阵, 它的第 n 行为向量 $\mathbf{x}_n - \bar{\mathbf{x}}$, 类似地, 定义 Ω 为一个 $M \times N$ 的矩阵, 它的第 n 行是向量 $\mathbb{E}[z_n]$ 。这样PCA的EM算法的E步骤 (12.54) 就变成了

$$\Omega = (\mathbf{W}_{\text{旧}}^T \mathbf{W}_{\text{旧}})^{-1} \mathbf{W}_{\text{旧}}^T \tilde{\mathbf{X}}^T \quad (12.58)$$

M步骤 (12.56) 的形式为

$$\mathbf{W}_{\text{新}} = \tilde{\mathbf{X}}^T \Omega^T (\Omega \Omega^T)^{-1} \quad (12.59)$$

与之前一样, 可以使用一种在线的方式执行。这些方程有一个很简单的意义, 如下所述。根据我们之前的讨论, 我们看到E步骤涉及到数据点在当前估计的主子空间上的正交投影。对应地, M步骤表示对主子空间的重新估计, 使得平方重建误差最小, 其中投影固定。

我们可以给出这个EM算法的一个简单的物理类比, 这对于 $D = 2$ 和 $M = 1$ 的情形很容易进行可视化。考虑二维空间中的一组数据点, 令一维主子空间用一个固体的杆表示。现在使用一个遵守胡克定律 (存储的能量正比于弹簧长度的平方) 的弹簧将每个数据点与杆相连。在E步骤中, 我们保持杆固定, 让附着的点沿着杆上下滑动, 使得能量最小。这使得每个数据点独立地到达对应的数据点在杆上的正交投影的位置。在M步骤中, 我们令附着点固定, 然后松开杆, 让杆达到能量最小的位置。然后E步骤和M步骤不断重复, 直到满足一个收敛准则, 如图12.12所示。

12.2.3 贝叶斯PCA

目前在我们关于PCA的讨论中, 我们假定主子空间的维度 M 是给定的。在实际应用中, 我们必须根据应用选择一个合适的值。为了数据可视化, 我们一般选择 $M = 2$, 而对于其他的应用, M 的合适的选择就没有这么明显了。一种方法是画出数据集的特征值谱线, 类似于离线手写数字数据集的图12.4的例子, 然后看特征值是否自然地分成了两组, 一组由很小的值组成, 另

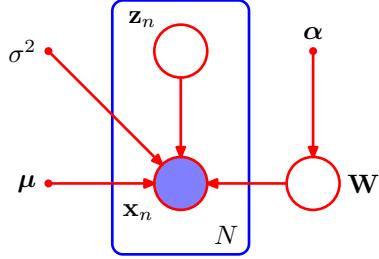


图 12.13: 贝叶斯PCA的概率图模型，其中参数矩阵 W 上的概率分布由超参数向量 α 控制。

一组由相对较大的值组成，两组之间有一个很明显的区分，表示 M 的选择存在一个很自然的值。在实际应用中，这种明显的区分通常无法看到。

由于概率PCA模型有一个具有良好定义的似然函数，因此我们可以使用交叉验证的方法，通过选择在验证数据集上的对数似然函数最大的模型来确定维度的值。然而，这种方法计算量很大，特别是如果我们考虑PCA混合概率模型时更是如此（Tipping and Bishop, 1999a）。在PCA混合概率模型中，我们要为每个混合分量单独确定合适的维度。

我们已经有了PCA模型的概率形式，似乎寻找贝叶斯模型选择的方法是很自然的。为了完成这件事，我们需要关于合适的先验概率分布，将模型参数 μ 、 W 和 σ^2 积分出去。可以使用变分框架来近似这个无法解析求解的积分（Bishop, 1999b）。这样，由变分下界给出的边缘似然函数的值就可以在不同的 M 值之间进行比较，然后选择具有最大边缘似然函数的 M 值。

这里，我们考虑一个更简单的方法，基于证据近似（evidence approximation），它适用于数据点的数量相对较大以及对应的后验概率分布有尖峰的情形（Bishop, 1999a）。它涉及到对 W 上的先验概率分布的一个具体的选择，使得主子空间中多余的维度可以从模型中剪枝掉。这对应于7.2.2节讨论的自动相关性确定（automatic relevance determination, ARD）的一个例子。具体来说，我们在 W 的每个列上定义一个独立的高斯先验，这些列表示定义了主子空间的响亮。每个这样的高斯分布有一个独立的方差，由精度超参数 α_i 控制，从而

$$p(W | \alpha) = \prod_{i=1}^M \left(\frac{\alpha_i}{2\pi} \right)^{\frac{D}{2}} \exp \left\{ -\frac{1}{2} \alpha_i w_i^T w_i \right\} \quad (12.60)$$

其中 w_i 是 W 的第 i 列。生成的模型可以使用图12.13的有向图表示。

α_i 的值可以通过最大化边缘似然函数的方式迭代地求解，其中 W 被积分出去。作为最优化的结果，某个 α_i 可能趋于无穷大，对应的参数向量 w_i 趋于零（后验概率分布变成了原点处的delta函数），得到了一个稀疏解。这样，主子空间的有效的维度由有限的 α_i 的值确定，对应的向量 w_i 可以被认为对于数据分布的建模是“有关系的”。通过这种方式，贝叶斯方法自动地在提升数据拟合程度（使用较多的向量 w_i 以及对应的根据数据调节的特征值 λ_i ）和减小模型复杂度（压制某些 w_i 向量的值）之间进行了折中。这种稀疏性的来源之前在相关向量机的问题中已经讨论过。

α_i 的值在训练阶段通过最大化似然函数的方式被重新估计，形式为

$$p(X | \alpha, \mu, \sigma^2) = \int p(X | W, \mu, \sigma^2) p(W | \alpha) dW \quad (12.61)$$

其中， $p(X | W, \mu, \sigma^2)$ 的对数由公式（12.43）给出。注意，为了简化起见，我们也将 μ 和 σ^2 看成待估计的参数，而没有在这些参数上定义先验概率分布。

由于积分无法直接计算，因此我们使用拉普拉斯近似。如果我们假设后验概率分布有尖峰，这种情况对于足够大的数据集确实会发生，那么重估计方程可以通过关于 α_i 最大化边缘似然函数的方式得到，形式为

$$\alpha_i^{\text{新}} = \frac{D}{w_i^T w_i} \quad (12.62)$$

这可以从公式（3.98）中推导出来，只需注意到 w_i 的维度是 D 即可。这些重新估计过程与确定 W 和 σ_2 的EM算法的更新过程交织在一起。与之前一样，E步骤方程由公式（12.54）和

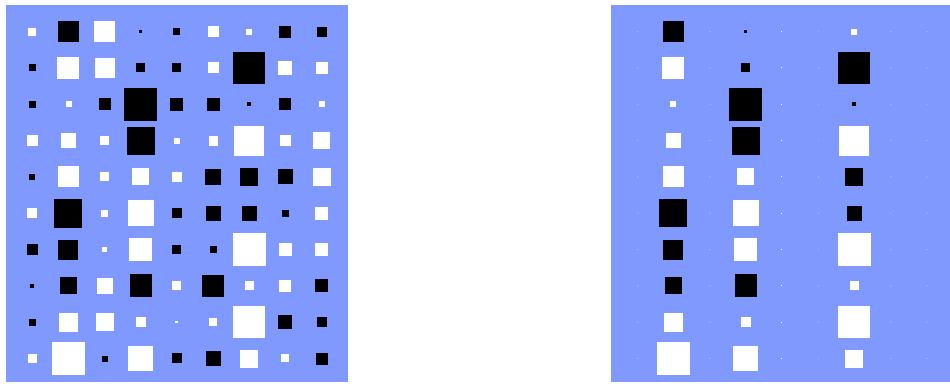


图 12.14: 矩阵 \mathbf{W} 的Hinton图, 其中矩阵的每个元素被表示为一个正方形, 白色表示正值, 黑色表示负值, 正方形的面积正比于那个元素的大小。人工生成的数据集由300个数据点构成, 数据点从一个 $D = 10$ 维的高斯分布中采样, 高斯分布在3个方向上的标准差为1.0, 在剩余的7个方向上的标准差为0.5。数据空间的维度为 $D = 10$, 在 $M = 3$ 个方向上的方差大于剩余的7个方向上的方差。左图给出了使用最大似然方法的概率PCA的结果, 右图给出了贝叶斯PCA的对应的结果。我们看到通过压制6个多余的自由度的方式来发现维度的合适的值。

(12.55) 给出。类似地, σ^2 的M步骤方程由公式 (12.57) 给出。在M步骤中的唯一的改变是 \mathbf{W} 的方程, 它修改后的形式为

$$\mathbf{W}_{\text{新}} = \left[\sum_{n=1}^N (\mathbf{x}_n - \bar{\mathbf{x}}) \mathbb{E}[\mathbf{z}_n]^T \right] \left[\sum_{n=1}^N \mathbb{E}[\mathbf{z}_n \mathbf{z}_n^T] + \sigma^2 \mathbf{A} \right]^{-1} \quad (12.63)$$

其中 $\mathbf{A} = \text{diag}(\alpha_i)$ 。与之前一样, μ 的值为样本均值。

如果我们令 $M = D - 1$, 那么如果所有的 α_i 是有限值, 那么模型表示一个具有完整协方差的高斯模型, 而如果所有的 α_i 区域无穷大, 那么模型等价于各向同性的高斯模型, 从而模型可以涵盖主子空间的有效维度的所有可能的值。也可以考察较小的 M 值, 这会降低计算量, 但是也限制了子空间的最大维度。这个算法与标准的概率PCA算法的对比如图12.14所示。

贝叶斯PCA使得我们有机会来说明11.3节讨论的吉布斯采样算法。图12.15给出了对超参数 $\ln \alpha_i$ 采样的例子, 数据集的维度为 $D = 4$, 潜在空间的维度为 $M = 3$, 但是数据集通过一个概率PCA模型生成, 这个模型在一个方向上的方差较大, 剩余方向由较低方差的噪声组成。结果很明显地展示了后验概率分布中三个不同峰值的存在。在每轮迭代中, 一个超参数具有较小的值, 剩下的两个具有较大的值, 因此三个潜在变量中的两个被压制。在吉布斯采样的过程中, 解在三个峰值之间会发生很明显的转移。

这里描述的模型仅仅涉及到矩阵 \mathbf{W} 上的先验概率分布。关于PCA的完整的贝叶斯方法, 包括 μ, σ^2, α 上的先验概率分布, 以及使用变分方法的解, 可以参考Bishop (1999b)。关于确定PCA模型的合适维度的不同的贝叶斯方法的讨论, 可以参考Minka (2001c)。

12.2.4 因子分析

因子分析是一个线性高斯潜在变量模型, 它与概率PCA密切相关。它的定义与概率PCA的唯一差别是给定潜在变量 \mathbf{z} 的条件下观测变量 \mathbf{x} 的条件概率分布的协方差矩阵是一个对角矩阵而不是各向同性的协方差矩阵, 即

$$p(\mathbf{x} | \mathbf{z}) = \mathcal{N}(\mathbf{x} | \mathbf{W}\mathbf{z} + \boldsymbol{\mu}, \boldsymbol{\Psi}) \quad (12.64)$$

其中 $\boldsymbol{\Psi}$ 是一个 $D \times D$ 的对角矩阵。注意, 与概率PCA模型相同, 因子分析模型假设在给定潜在变量 \mathbf{z} 的条件下, 观测变量 x_1, \dots, x_D 是独立的。本质上讲, 因子分析模型这样解释数据的观测协方差结构: 表示出矩阵 $\boldsymbol{\Psi}$ 中与每个坐标相关联的独立的变量, 然后描述矩阵 \mathbf{W} 中的变量之间的协方差。在因子分析的文献中, \mathbf{W} 的列描述了观测变量之间的相关性关系, 被称为因子载入 (factor loading)。 $\boldsymbol{\Psi}$ 的对角元素, 表示每个变量的独立噪声方差, 被称为唯一性 (uniqueness)。

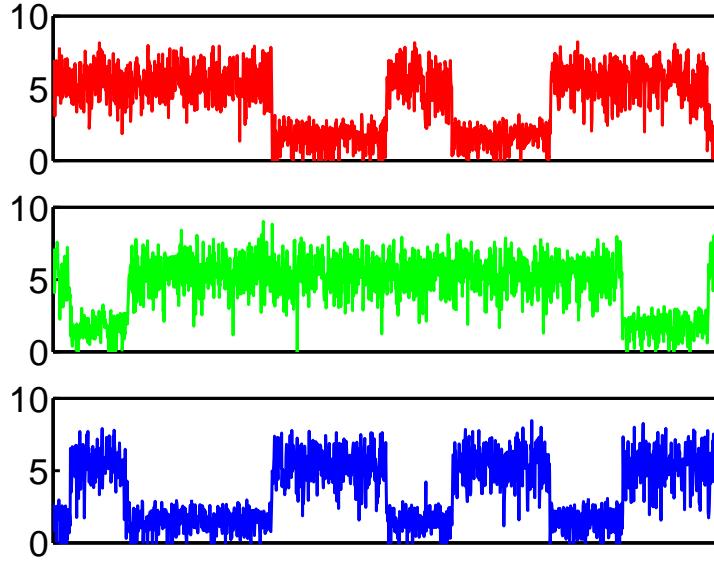


图 12.15: 用于贝叶斯PCA的吉布斯采样。图中给出了对于三个不同的 α 值， $\ln \alpha_i$ 关于迭代次数的图像。可以看出在后验概率分布的三个峰值之间的变化。

因子分析的起源于PCA一样早。关于因子分析的讨论可以参考Everitt (1984)、Bartholomew (1987) 和Basilevsky (1994)。Lawley (1953) 和Anderson (1963) 研究了因子分析与PCA之间的联系，证明了在似然函数的驻点处，对于一个 $\Psi = \sigma^2 \mathbf{I}$ 的因子分析模型， \mathbf{W} 的列是样本协方差的放缩后的特征向量， σ^2 是丢弃的特征值的平均值。后来，Tipping and Bishop (1999b) 证明，当组成 \mathbf{W} 的特征向量被选为主特征向量时，对数似然函数取得最大值。

使用公式 (2.115)，我们看到观测变量的边缘概率分布为 $p(\mathbf{x}) = \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}, \mathbf{C})$ ，其中

$$\mathbf{C} = \mathbf{WW}^T + \Psi \quad (12.65)$$

与概率PCA相同，模型对于潜在空间中的选择具有不变性。

历史上，在因子分析中，当我们试图给独立的因子（ z 空间的坐标）赋予一个直观的意义时，因子分析就变成了争论的焦点。由于潜在空间中的选择不变性，因子分析中存在不可区分的问题，这会造成很多麻烦。然而，从我们的角度来说，我们将因子分析看成一种形式的潜在变量密度模型，其中我们感兴趣的是潜在空间的形式，而不是描述它的具体的坐标系的选择。如果我们想要移除与潜在空间旋转相关联的模型的退化，那么我们必须考虑非高斯的潜在变量分布，这就产生了独立成分分析 (ICA) 模型。

我们可以使用最大似然方法确定因子分析模型中的参数 $\boldsymbol{\mu}, \mathbf{W}, \Psi$ 的值。与之前一样， $\boldsymbol{\mu}$ 的解是样本的均值。然而，与概率PCA不同， \mathbf{W} 的最大似然解不再具有解析解，因此必须迭代地求解。由于因子分析是一个潜在变量模型，因此可以使用与概率PCA模型中使用的EM算法相近似的EM算法来计算 (Rubin and Thayer, 1982)。具体来说，E步骤方程为

$$\mathbb{E}[z_n] = \mathbf{G}\mathbf{W}^T \Psi^{-1}(\mathbf{x}_n - \bar{\mathbf{x}}) \quad (12.66)$$

$$\mathbb{E}[z_n z_n^T] = \mathbf{G} + \mathbb{E}[z_n] \mathbb{E}[z_n]^T \quad (12.67)$$

其中我们已经定义了

$$\mathbf{G} = (\mathbf{I} + \mathbf{W}^T \Psi^{-1} \mathbf{W})^{-1} \quad (12.68)$$

注意，这里使用了一个 $M \times M$ 的矩阵求逆的表达方式，而不是 $D \times D$ 的表达方式（除非 Ψ 是 $D \times D$ 的对角矩阵，此时求逆很简单，只需 $O(D)$ 次计算），这通常很方便，因为通常 $M \ll D$ 。类似地，M步骤方程的形式为

$$\mathbf{W}_{\text{新}} = \left[\sum_{n=1}^N (\mathbf{x}_n - \bar{\mathbf{x}}) \mathbb{E}[z_n]^T \right] \left[\sum_{n=1}^N \mathbb{E}[z_n z_n^T] \right]^{-1} \quad (12.69)$$

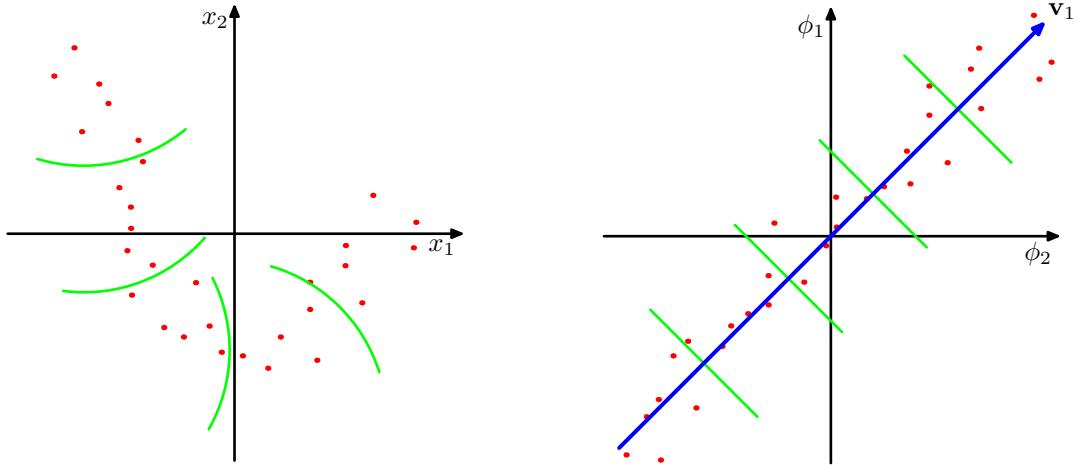


图 12.16: 核PCA的图形化说明。原始数据空间中的数据集（左图）被非线性变换 $\phi(\mathbf{x})$ 投影到特征空间中（右图）。通过在特征空间中执行PCA，我们得到了主成分，其中第一主成分用蓝色表示，记作向量 v_1 。特征空间中的绿色直线表示特征空间中在第一主成分上的线性投影，它对应于原始数据空间中的非线性投影。注意，通常不可能在 x 空间中表示非线性主成分。

$$\Psi_{\text{新}} = \text{diag} \left\{ S - W^{\text{新}} \frac{1}{N} \sum_{n=1}^N \mathbb{E}[z_n](x_n - \bar{x})^T \right\} \quad (12.70)$$

其中，“diag”算符将所有非对角线上的元素全部设置为零。使用本书中讨论过的方法，可以很容易地得到因子分析模型的贝叶斯方法。

概率PCA与因子分析的另一个不同点关注的是数据集在变换下的行为的差异。对于PCA和概率PCA来说，如果我们在数据空间中选择坐标系，那么我们对数据的拟合不会发生任何变化，但是 W 会使用对应的选择矩阵进行变换。然而，对于因子分析来说，类似的性质是，如果我们对于数据向量进行一个分量之间的重新缩放，那么这种缩放可以被整合到对 Ψ 的元素的重新缩放之中。

12.3 核PCA

在第6章中，我们看到了核替换的方法让我们能够使用形如 $\mathbf{x}^T \mathbf{x}'$ 的标量积表示的算法，并且通过使用一个非线性核替换标量积的方式来对算法进行推广。这里，我们将核替换的方法应用到主成分分析中，从而得到了一个非线性的推广，被称为核PCA（kernel PCA）（Schölkopf et al., 1998）。

考虑 D 维空间中的一个观测数据集 \mathbf{x}_n ，其中 $n = 1, \dots, N$ 。为了保持记号的简洁，我们假设我们已经从每个 \mathbf{x}_n 中减去了样本的均值，从而 $\sum_n \mathbf{x}_n = \mathbf{0}$ 。第一步是将传统的PCA表示为这样的形式：数据向量 $\{\mathbf{x}_n\}$ 只以标量积 $\mathbf{x}_n^T \mathbf{x}_m$ 的形式出现。回忆一下，主成分由协方差矩阵的特征向量 \mathbf{u}_i 定义，即

$$S\mathbf{u}_i = \lambda_i \mathbf{u}_i \quad (12.71)$$

其中 $i = 1, \dots, D$ 。这里 $D \times D$ 的样本协方差矩阵 S 的定义为

$$S = \frac{1}{N} \sum_{n=1}^N \mathbf{x}_n \mathbf{x}_n^T \quad (12.72)$$

特征向量被归一化，从而 $\mathbf{u}_i^T \mathbf{u}_i = 1$ 。

现在考虑到一个 M 维特征空间的一个非线性变换 $\phi(\mathbf{x})$ ，从而每个数据点 \mathbf{x}_n 被投影到一个数据点 $\phi(\mathbf{x}_n)$ 上。我们现在可以在特征空间上进行标准的PCA，它隐式地在原始数据空间中定义了一个非线性的主成分模型，如图12.16所示。

现阶段，让我们假设投影数据集的均值也为零，从而 $\sum_n \phi(\mathbf{x}_n) = \mathbf{0}$ 。我们稍后会回到这里。特征空间中的 $M \times M$ 样本协方差矩阵为

$$\mathbf{C} = \frac{1}{N} \sum_{n=1}^N \phi(\mathbf{x}_n) \phi(\mathbf{x}_n)^T \quad (12.73)$$

它特征向量展开式被定义为

$$\mathbf{C}\mathbf{v}_i = \lambda_i \mathbf{v}_i \quad (12.74)$$

其中 $i = 1, \dots, M$ 。我们的目标是求解这个特征值问题，而无需显式地在特征空间中计算。根据 \mathbf{C} 的定义，特征向量方程告诉我们 \mathbf{v}_i 满足

$$\frac{1}{N} \sum_{n=1}^N \phi(\mathbf{x}_n) \{\phi(\mathbf{x}_n)^T \mathbf{v}_i\} = \lambda_i \mathbf{v}_i \quad (12.75)$$

因此我们看到（假设 $\lambda_i > 0$ ）向量 \mathbf{v}_i 由 $\phi(\mathbf{x}_n)$ 的特征值给出，因此可以写成

$$\mathbf{v}_i = \sum_{n=1}^N a_{in} \phi(\mathbf{x}_n) \quad (12.76)$$

将这个表达式代回到特征向量方程中，我们有

$$\frac{1}{N} \sum_{n=1}^N \phi(\mathbf{x}_n) \phi(\mathbf{x}_n)^T \sum_{m=1}^N a_{im} \phi(\mathbf{x}_m) = \lambda_i \sum_{n=1}^N a_{in} \phi(\mathbf{x}_n) \quad (12.77)$$

现在关键的步骤是用核函数 $k(\mathbf{x}_n, \mathbf{x}_m) = \phi(\mathbf{x}_n)^T \phi(\mathbf{x}_m)$ 表示上式。我们可以将两侧乘以 $\phi(\mathbf{x}_l)^T$ ，得到

$$\frac{1}{N} \sum_{n=1}^N k(\mathbf{x}_l, \mathbf{x}_n) \sum_{m=1}^N a_{im} k(\mathbf{x}_n, \mathbf{x}_m) = \lambda_i \sum_{n=1}^N a_{in} k(\mathbf{x}_l, \mathbf{x}_n) \quad (12.78)$$

这可以用矩阵的记号表示为

$$\mathbf{K}^2 \mathbf{a}_i = \lambda_i N \mathbf{K} \mathbf{a}_i \quad (12.79)$$

其中 \mathbf{a}_i 是一个 N 维列向量，元素为 a_{in} ，其中 $n = 1, \dots, N$ 。我们可以通过求解下面的特征值方程

$$\mathbf{K} \mathbf{a}_i = \lambda_i N \mathbf{a}_i \quad (12.80)$$

来找到 \mathbf{a}_i 的解，其中我们已经从方程 (12.79) 两侧去掉了因子 \mathbf{K} 。注意，方程 (12.79) 和 (12.80) 的解的唯一差别在于 \mathbf{K} 的那些特征值为零的特征向量，这些特征向量不会影响主成分投影。

系数 \mathbf{a}_i 的归一化条件可以通过要求特征空间的特征向量被归一化的方式得到。使用公式 (12.76) 和 (12.80)，我们有

$$1 = \mathbf{v}_i^T \mathbf{v}_i = \sum_{n=1}^N \sum_{m=1}^N a_{in} a_{im} \phi(\mathbf{x}_n)^T \phi(\mathbf{x}_m) = \mathbf{a}_i^T \mathbf{K} \mathbf{a}_i = \lambda_i N \mathbf{a}_i^T \mathbf{a}_i \quad (12.81)$$

解出了特征向量方程之后，得到的主成分投影也可以根据核函数进行转化。使用公式 (12.76)，点 \mathbf{x} 在特征向量 i 上的投影为

$$y_i(\mathbf{x}) = \phi(\mathbf{x})^T \mathbf{v}_i = \sum_{n=1}^N a_{in} \phi(\mathbf{x})^T \phi(\mathbf{x}_n) = \sum_{n=1}^N a_{in} k(\mathbf{x}, \mathbf{x}_n) \quad (12.82)$$

从而我们又一次得到了根据核函数进行表示的形式。

在原始的 D 维 \mathbf{x} 空间中，有 D 个正交的特征向量，因此我们最多可以找到 D 个线性主成分。然而特征空间的维度 M 可以比 D 大得多，甚至可以是无穷大，因此我们可以找到多于 D 个非线性主成分。但是注意，非零特征值的数量不能超过数据点的数量 N ，因为（即使 $M > N$ ）特征空间中的协方差矩阵的秩最大等于 N 。这可以从下面的事实中反映出来：核PCA涉及到对 $N \times N$ 矩阵 \mathbf{K} 的特征向量展开。

目前为止，我们假设由 $\phi(\mathbf{x}_n)$ 给出的投影数据集的均值为零，通常的情况并非如此。我们不能简单地计算然后减去均值，因为我们希望避免直接在特征空间中进行计算，因此我们完全根据核函数来建立算法的公式。在中心化之后，投影的数据点（记作 $\tilde{\phi}(\mathbf{x}_n)$ ）为

$$\tilde{\phi}(\mathbf{x}_n) = \phi(\mathbf{x}_n) - \frac{1}{N} \sum_{l=1}^N \phi(\mathbf{x}_l) \quad (12.83)$$

从而Gram矩阵的对应元素为

$$\begin{aligned} \tilde{K}_{nm} &= \tilde{\phi}(\mathbf{x}_n)^T \tilde{\phi}(\mathbf{x}_m) \\ &= \phi(\mathbf{x}_n)^T \phi(\mathbf{x}_m) - \frac{1}{N} \sum_{l=1}^N \phi(\mathbf{x}_n)^T \phi(\mathbf{x}_l) \\ &\quad - \frac{1}{N} \sum_{l=1}^N \phi(\mathbf{x}_l)^T \phi(\mathbf{x}_m) + \frac{1}{N^2} \sum_{j=1}^N \sum_{l=1}^N \phi(\mathbf{x}_j)^T \phi(\mathbf{x}_l) \\ &= k(\mathbf{x}_n, \mathbf{x}_m) - \frac{1}{N} \sum_{l=1}^N k(\mathbf{x}_l, \mathbf{x}_m) \\ &\quad - \frac{1}{N} \sum_{l=1}^N k(\mathbf{x}_n, \mathbf{x}_l) + \frac{1}{N^2} \sum_{j=1}^N \sum_{l=1}^N k(\mathbf{x}_j, \mathbf{x}_l) \end{aligned} \quad (12.84)$$

使用矩阵的记号，这个结果可以表示为

$$\tilde{\mathbf{K}} = \mathbf{K} - \mathbf{1}_N \mathbf{K} - \mathbf{K} \mathbf{1}_N + \mathbf{1}_N \mathbf{K} \mathbf{1}_N \quad (12.85)$$

其中 $\mathbf{1}_N$ 表示 $N \times N$ 的矩阵，它的每个元素的值都是 $\frac{1}{N}$ 。因此，我们可以只使用核函数来计算 $\tilde{\mathbf{K}}$ ，然后使用 $\tilde{\mathbf{K}}$ 确定特征值和特征向量。注意，如果我们使用线性核 $k(\mathbf{x}, \mathbf{x}') = \mathbf{x}^T \mathbf{x}'$ ，那么我们就恢复出了标准的PCA算法。图12.17给出了核PCA应用于人工生成数据集的一个例子（Schölkopf et al., 1998）。这里，我们将一个“高斯”核

$$k(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{0.1}\right) \quad (12.86)$$

应用于人工生成数据集。图中的曲线对应于沿着曲线方向，在对应的主成分上投影为常数的轮廓线，投影的定义为

$$\phi(\mathbf{x})^T \mathbf{v}_i = \sum_{n=1}^N a_{in} k(\mathbf{x}, \mathbf{x}_n) \quad (12.87)$$

核PCA的一个明显的缺点是它涉及到寻找 $N \times N$ 矩阵 $\tilde{\mathbf{K}}$ 的特征向量，而不是传统的线性PCA中 $D \times D$ 的矩阵 \mathbf{S} 的特征向量，因此在实际应用中，对于较大的数据集，我们经常使用近似。

最后，我们注意到在标准的线性PCA中，我们通常保留 $L < D$ 个特征向量，然后使用数据向量 \mathbf{x}_n 在 L 为主子空间上的投影 $\hat{\mathbf{x}}_n$ 来近似数据向量 \mathbf{x}_n ，投影的定义为

$$\hat{\mathbf{x}}_n = \sum_{i=1}^L (\mathbf{x}_n^T \mathbf{u}_i) \mathbf{u}_i \quad (12.88)$$

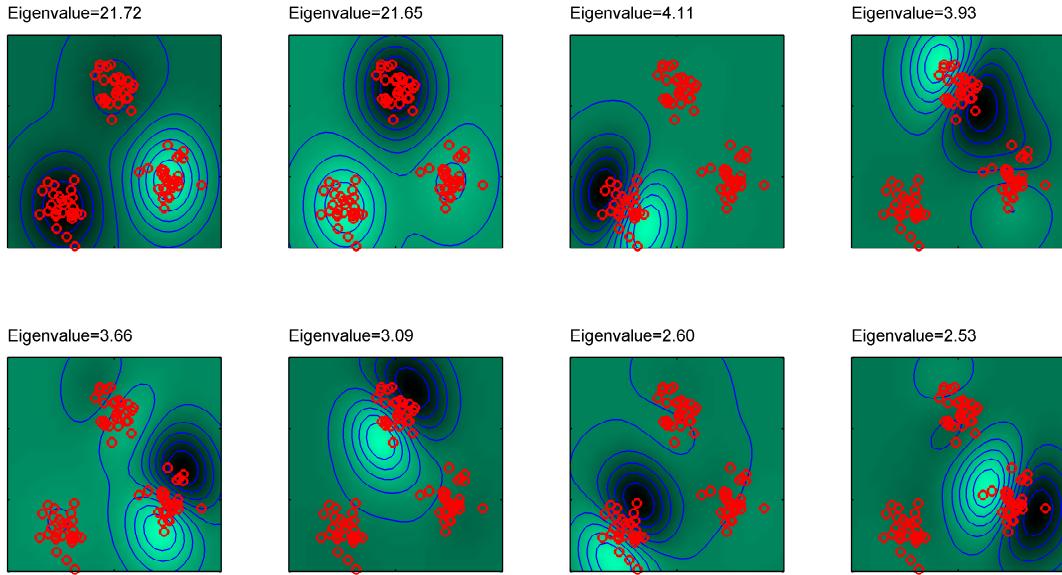


图 12.17: 使用高斯核的核PCA用于二维空间的人工生成数据集的例子，图中画出了前8个特征函数以及对应的特征值。轮廓线表示沿着曲线的方向，在对应的主成分上的投影是常数的轮廓线。注意前两个特征向量将三个聚类划分开，接下来的三个特征向量将每个聚类分成一两半，再接下来的三个特征向量再次将聚类划分为两半，方向是沿着与之前的划分正交的方向。

在核PCA中，这通常是不可能的。为了说明这一点，我们注意到映射 $\phi(x)$ 将 D 维 x 空间映射到了 M 维特征空间 ϕ 中的一个 D 维流形 (manifold) 中。向量 x 被称为对应点 $\phi(x)$ 的原像 (pre-image)。然而，特征空间中的点在特征空间的线性PCA子空间中的投影通常不会位于非线性 D 维流形中，因此在数据空间中不会存在一个对应的原像。于是，研究者们提出了一些寻找近似原像的方法 (Bakir et al., 2004)。

12.4 非线性隐含变量模型

本章中，我们将注意力集中与带有连续潜在变量的最简单的一类模型上，即基于线性高斯分布的模型。这些模型在实际应用中很重要，并且这些模型相对容易分析，容易拟合数据，也可以用作更复杂模型的基本成分。这里，我们简要讨论一下对这个框架的一些推广，推广到非线性的模型，或者非高斯的模型，或者二者兼具的模型。

实际上，非线性性质和非高斯性质是相互关联的，因为一般的概率密度可以从一个简单的固定的参考概率密度（例如高斯分布）中得到，只需对变量进行非线性变换即可。这个想法构成了几个实际应用中的潜在变量模型的基础，正如我们将看到的那样。

12.4.1 独立成分分析

首先，我们考虑观测变量与潜在变量线性相关的模型，但是潜在概率分布不是高斯分布。这种模型的一个重要的类别被称为独立成分分析 (independent component analysis)，或者ICA。如果我们考虑潜在变量上的概率分布的分解，即

$$p(\mathbf{z}) = \sum_{j=1}^M p(z_j) \quad (12.89)$$

那么我们就会应用到这个模型。为了理解这种模型的作用，考虑这样一个场景：两个人同时讲话，我们使用两个麦克风来记录他们的声音。如果我们忽略诸如时间延迟和回声之类的影响，那么在任意时间点，麦克风接收到的信号都是两个声音的振幅的线性组合。这个线性组合的系数是常数，并且如果我们可以从采样数据中推断它们的值，那么我们就可以将混合的过程（假

设非奇异) 进行求逆, 从而得到两个干净的信号, 每个信号只包含一个人的声音。这是盲源划分 (blind source separation) 问题的一个例子, 其中, “盲”表示我们只给定了混合数据, 而原始的数据源和混合系数都没有被观测到 (Cardoso, 1998)。

这类问题有时使用下面的方法解决 (MacKay, 2003), 其中我们忽略信号的时序本质, 将连续的样本看成是独立同分布的。我们考虑一个生成式模型, 其中有两个潜在变量, 对应于未观测的语音信号的幅值, 有两个观测变量, 由麦克风的信号值给定。潜在变量的联合概率分布可以按照上面的方式分解, 观测变量由潜在变量的线性组合给定。我们无需引入一个噪声分布, 因为潜在变量的数量等于观测变量的数量, 从而观测变量的边缘概率分布通常不会是奇异的, 因此观测变量仅仅由潜在变量的线性组合确定。给定一组观测数据, 模型的似然函数是线性组合的系数的一个函数。对数似然函数可以使用基于梯度的最优化方法进行最大化, 得到了独立成分分析的一个特定的版本。

这种方法的成功需要令潜在变量具有非高斯的概率分布。为了说明这一点, 回忆一下在概率PCA (以及因子分析) 中, 潜在空间分布是一个零均值的各向同性的高斯分布。于是, 模型无法区分那些区别仅仅在于潜在空间的旋转的潜在变量的不同选择。这一点可以用下面的方法直接验证: 我们注意到边缘概率密度 (12.35) 在变换 $\mathbf{W} \rightarrow \mathbf{WR}$ 下是不变的, 因此似然函数也是不变的, 其中 \mathbf{R} 是正交矩阵, 满足 $\mathbf{RR}^T = \mathbf{I}$, 这是因为公式 (12.36) 给出的矩阵 \mathbf{C} 本身是不变的。将这个模型进行扩展, 使得更多的概率潜在分布被包含到模型中, 结论不会改变, 因为正如我们已经看到的那样, 这种模型等价于零均值各向同性的高斯潜在变量模型。

我们用另一种方式说明为什么线性模型中的高斯潜在变量分布对于找到独立的成分是不够的。我们注意到, 主成分表示数据空间中的坐标系的一个旋转, 从而对协方差矩阵进行了对角化, 因此新的坐标系中的数据分布没有相关性。虽然不具有相关性是独立性的一个必要条件, 但是它不是充分条件。在实际应用中, 潜在变量分布的一个常见的选择是

$$p(z_j) = \frac{1}{\pi \cosh(z_j)} = \frac{2}{\pi(e^{z_j} + e^{-z_j})} \quad (12.90)$$

这与高斯分布相比, 具有长尾的性质, 这反映了许多现实世界中的概率分布同样具有这种性质。

最初的ICA模型 (Bell and Sejnowski, 1995) 基于的是由信息最大化定义的目标函数的最优化过程。概率潜在变量形式的一个优点是它有助于对基本ICA的推广进行形式化描述。例如, 独立因子分析 (independent factor analysis) 研究的是这样的模型: 潜在变量的数量和观测变量的数量可以不同, 观测变量带有噪声, 各个潜在变量的概率分布很灵活, 由混合高斯模型建模。这个模型的对数似然函数使用EM算法进行最大化, 潜在变量的重建使用变分方法进行近似。研究者们也在研究许多其他类型的模型, 现在有许多文献研究ICA及其应用 (Jutten and Herault, 1991; Comon et al., 1991; Amari et al., 1996; Pearlmutter and Parra, 1997; Hyvärinen and Oja, 1997; Hinton et al., 2001; Miskin and MacKay, 2001; Hojen-Sorensen et al., 2002; Choudrey and Roberts, 2003; Chan et al., 2003; Stone, 2004)。

12.4.2 自关联网络

在第5章中, 我们在有监督学习的环境中研究了神经网络, 其中网络的左右是在给定输入变量值的条件下预测输出。然而, 神经网络也被应用于无监督学习, 此时神经网络用于维度降低。使用输出结点与输入结点数量相同的神经网络, 通过最优化权值来最小化某种度量, 这种度量描述了在训练数据集上, 输入和输出之间的重建误差。

首先, 考虑图12.18所示的多层感知器网络, 它有 D 个输入, D 个输出, 以及 M 个隐含单元, 其中 $M < D$ 。用来训练网络所使用的目标棉量仅仅是输入向量本身, 因此网络试图将输入向量映射到它本身上。这样的网络构成了一个自相关映射 (autoassociative mapping)。由于隐含单元的数量小于输入的数量, 因此将所有的变量进行一个完美的重建通常是不可能的。于是, 我们通过最小化一个误差函数的方式来确定网络的参数 \mathbf{w} , 这个误差函数描述了输入向量和它们的重建之间的不匹配程度。特别地, 我们会现在一个平方和误差函数, 形式为

$$E(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^N \|\mathbf{y}(\mathbf{x}_n, \mathbf{w}) - \mathbf{x}_n\|^2 \quad (12.91)$$

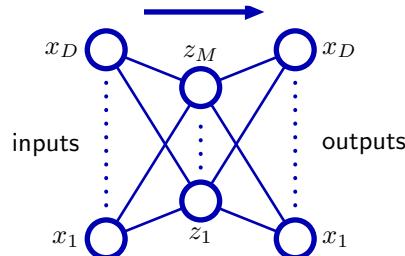


图 12.18: 一个自相关的多层感知器网络，具有两层权值。这样的一个网络通过最小化平方和误差的方式进行训练，得到从输入向量到其自身的一个映射。即使隐含层是非线性单元，这样一个网络也等价于线性主成分分析。为了清晰，表示偏置参数的链接已经被略去。

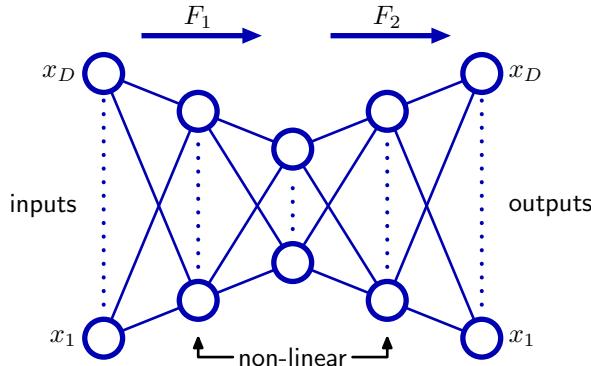


图 12.19: 增加额外的非线性单元组成的隐含层产生了一个能够进行非线性维度降低的自相关网络。

如果隐含单元具有线性激活函数，那么可以证明误差函数有唯一的全局最小值，在这个最小值处，网络实现了到一个 M 维子空间上的投影，这个子空间由数据的前 M 个主成分张成（Bourland and Kamp, 1988; Baldi and Hornik, 1989）。因此，图12.18的隐含单元的权向量构成了张成主子空间的基的集合。但是，注意，这些向量不需要正交或者归一化。这个结果毫不令人惊讶，因为主成分分析和神经网络都使用了线性维度降低、并且最小化相同的误差函数。

可能我们会认为，线性维度降低的局限性可以在网络的隐含单元中使用图12.18中的非线性（sigmoid）激活函数的方式来克服。但是，即使使用非线性隐含单元，误差函数的最小值同样通过在主子空间上投影的方式获得（Bourlard and Kamp, 1988）。于是使用两层神经网络在维度降低方面没有优势。主成分分析的标准方法（基于奇异值分解）保证在有限时间内给出正确的解，并且这种方法也产生了特征值的一个有序集合，对应于单位正交的特征向量。

然而，如果网络中有额外的隐含层，情况就会有所不同。考虑图12.19给出的四层自相关网络。与之前一样，输出单元是线性的，在第二个隐含层的 M 个单元也可以是线性的，但是第一个隐含层和第三个隐含层具有sigmoid非线性激活函数。网络同样用最小化误差函数（12.91）的方式确定。我们可以将这个网络看成两个连续的函数映射 F_1 和 F_2 ，如图12.19所示。第一个映射 F_1 将原始的 D 维数据映射到 M 维子空间 \mathcal{S} 上，这个子空间由第二个隐含层的单元的激活所定义。由于第一个非线性单元隐含层的存在，因此这个映射非常一般，并且特别地，这个映射不限于线性映射。类似地，网络的第二部分定义了从 M 维空间到原始 D 维输入空间中的一个任意的函数映射。这种映射有一个很简单的几何意义，图12.20给出了 $D = 3$ 和 $M = 2$ 的情形。

这样的网络能够有效地完成非线性主成分分析。它的优点在于，不局限于线性变换，虽然标准的主成分分析是它的一个具体的例子。然而，现在训练这个神经网络涉及到非线性最优化问题，因为误差函数（12.91）不再是网络参数的二次函数。我们必须使用需要大量计算的非线性最优化方法，并且有找到误差函数的局部极小值的风险。并且，子空间的维度必须在训练网络之前指定。

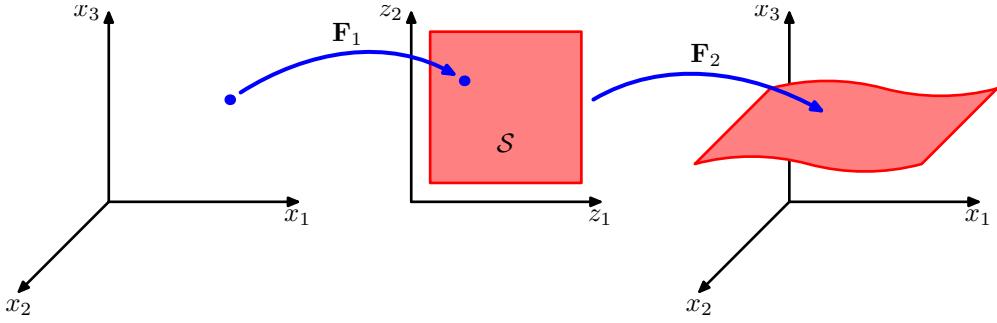


图 12.20: 图 12.19 的网络表示的映射的集合表示, 其中输入单元的数量为 $D = 3$, 中间隐含层单元的数量为 $M = 2$ 。函数 F_2 表示从 M 维空间 S 到 D 维空间的一个映射, 因此定义了空间 S 嵌入到原始 x 空间的方式。由于映射 F_2 可以是非线性的, 因此 S 嵌入的空间可以不是平面, 如图所示。这样, 映射 F_1 定义了原始 D 维空间中的一个点到 M 维子空间 S 中的投影。

12.4.3 对非线性流形建模

正如我们已经注意到的那样, 许多自然的数据源对应于低维的可能带有噪声的非线性流形, 这些流形镶嵌在更高维的观测数据空间中。显式地利用这个性质可以产生与一般的方法相比更好的概率密度模型。这里, 我们简要讨论尝试完成这一点的几种方法。

对这种非线性结构建模的一种方法是通过线性模型的组合, 从而我们对流形做了一个分段线性的近似。这个近似可以通过使用诸如 K 均值的聚类方法, 基于欧几里得距离, 将数据集划分为若干个局部的分组, 对每个分组分别使用标准的PCA。一种更好的方法是使用聚类分配的重建误差 (Kambhatla and Leen, 1997; Hinton et al., 1997), 然后在每个阶段优化一个共同的代价函数。然而, 这些方法会由于我们无法得到整体的概率密度模型而产生局限性。通过使用概率PCA, 很容易定义一个完整的概率模型, 只需考虑一个混合的概率分布, 每个分量都是概率PCA即可 (Tipping and Bishop, 1999a)。这样的模型既包含离散潜在变量, 对应于离散的混合, 也包含连续的潜在变量, 以及可以使用EM算法最大化的似然函数。基于变分推断的纯粹的贝叶斯方法 (Bishop and Winn, 2000) 使得混合分量的个数以及各个模型的有效维度可以从数据中进行推断。这个模型有很多变体, 例如将诸如 W 矩阵或噪声方差之类的参数与混合分量相关联, 或者将各向同性噪声概率分布替换为对角的噪声概率分布, 这就引出了因子分析的混合 (Ghahramani and Hinton, 1996a; Ghahramani and Beal, 2000)。概率PCA模型的混合也可以层次化地扩展, 产生了一个很有吸引力的数据可视化算法 (Bishop and Tipping, 1998)。

另一种处理方式是直接考虑一个单一的非线性模型, 而不是将线性模型混合。回忆一下, 传统的PCA寻找一个线性子空间, 这个子空间在最小平方的意义下能够以最近的距离通过数据集。这个概念可以推广到一维非线性曲面, 被称为主曲线 (principal curve) (Hastie and Stuetzle, 1989)。我们可以使用一个向量值的函数 $f(\lambda)$ 来描述 D 维数据空间中的一条曲线。这个函数的函数值是一个向量, 向量的元素是标量 λ 的一个函数。有许多种可能的方式来参数化这条曲线, 其中一种自然的选择是沿着曲线的弧的长度。对于数据空间中的任意给定的点 \hat{x} , 我们可以在曲线上寻找一个点, 它距离数据点的欧几里得距离最小。我们将这个点记作 $\lambda = g_f(x)$, 因为它依赖于一个特定的曲线 $f(\lambda)$ 。对于一个连续的数据概率密度 $p(x)$, 一个主曲线被定义为这样的曲线: 曲线上的每个点都是数据空间中那些投影到这个点的所有点的均值, 即

$$\mathbb{E}[x \mid g_f(x) = \lambda] = f(\lambda) \quad (12.92)$$

对于一个给定的连续概率密度, 可能存在多个主曲线。在实际应用中, 我们感兴趣的是有限的数据集, 并且我们还希望将注意力集中于光滑的曲线。Hastie and Stuetzle (1989) 提出了一个两阶段的迭代步骤来寻找这种主曲线, 与用于PCA的EM算法有些相似。曲线使用第一主成分进行初始化, 然后算法在数据投影步骤和曲线重估计步骤之间交替进行。在投影步骤中, 每个数据点被赋一个 λ 值, 对应于曲线上距离最近的点。然后, 在重估计步骤中, 曲线上的每个点都是那些投影到曲线上附近点的一个加权平均, 距离曲线最近的点的权重最大。在子空间是线性的情况下, 算法收敛于第一主成分, 等价于寻找协方差矩阵最大特征向量的幂方法。主曲线可以推广到多维流形中, 这个流形被称为主曲面 (principal surface), 但是主曲面的用途很有限, 因为高维空间的数据平滑很困难, 即使对于二维流形也是如此。

PCA经常被用于将数据集投影到低维空间中，例如二维空间中，用于数据的可视化。另一个目的很相似的线性方式是多维放缩（multidimensional scaling）或者被称为MDS（Cox and Cox, 2000）。这种方法寻找数据的一个低维投影，同时使得数据点之间的距离尽可能的近。这种方法需要寻找距离矩阵的特征向量。在距离的度量是欧几里得距离的情况下，它等价于PCA。MDS的概念可以推广到相当广泛的一大类数据类型中，这些数据类型根据相似度矩阵定义，得到了非度量MDS（nonmetric MDS）。

维度降低和数据可视化的另外两个非概率方法很值得一提。局部线性嵌入（locally linear embedding）或者LLE（Roweis and Saul, 2000）首先计算系数的集合，这些系数能够最好地从每个数据点中重建出它的相邻点。这些系数的设置使其对于数据点和相邻点的旋转、平移、缩放具有不变性，因此系数描述了相邻点的几何性质的特征。然后，LLE将高维数据点映射到低维空间中，同时保持这些邻域的系数。如果对于一个特定的数据点，局部的邻域可以被当做线性的，那么变换可以使用平移、旋转、缩放的组合来实现，从而保持数据点和它们的邻域之间的角度。由于权值对于这些变换具有不变性，因此我们预计重建低维空间的数据点和高维空间的数据点所需的权值相同。尽管具有非线性性质，对于LLE的优化不会有局部的极小值。

在等尺度特征映射（isometric feature mapping）或者isomap（Tenenbaum et al., 2000）中，目标是将数据点使用MDS投影到低维空间中，但是不相似度根据在流形上测量的曲面距离（geodesic distance）定义。例如，如果两个数据点位于一个圆上，那么曲面距离是沿着圆周测量的弧的长度，而不是沿着连接两点的弦的直线距离。首先，算法定义每个数据点的邻域，方法是寻找 K 个最近邻，或者寻找在一个半径为 ϵ 的球体内部的点。然后，通过将所有的邻域点进行连接，然后使用欧几里得距离标记这些距离，就可以构建出一个图。之后，任意点对之间的曲面距离通过对沿着连接它们的最短路径的弧的长度进行求和的方式得到。最后，有度量的MDS被应用于曲面距离矩阵上，用来寻找低维的投影。

我们在本章中关注的对象是观测变量为连续变量的模型。我们也可以考虑具有连续潜在变量以及离散官色变量的模型，这就产生了潜在特征模型（latent trait model）（Bartholomew, 1987）。在这种情况下，连续潜在变量上的积分无法解析地计算，即使潜在变量与观测变量之间具有线性关系的时候也是如此，因此我们需要更复杂的技术。Tipping (1999) 在一个具有二维潜在空间的模型中使用变分推断方法，使用一个二值的数据集可以进行可视化，这与使用PCA对连续数据可视化的情形类似。注意，这个模型是4.5节讨论的贝叶斯logistic回归问题的对偶问题。在logistic回归的情形中，我们有特征向量 ϕ_n 的 N 次观测，特征向量使用一个单一的权向量 w 进行参数描述，而在潜在空间可视化模型中，存在一个单一的潜在空间变量 x （类似于 ϕ ）以及潜在变量 w_n 的 N 个副本。Collins et al. (2002) 将概率潜在变量模型推广到了一般的指数族分布的情形。

我们已经注意到，通过使用一个恰当的非线性变换作用于高斯随机变量上，我们可以建立任意的概率分布。这个结论被用于更一般的潜在变量模型中，被称为密度网络（density network）（MacKay, 1995; MacKay and Gibbs, 1999），其中非线性函数由多层神经网络控制。如果网络有足够的隐含结点，那么它能够以任意的精度近似给定的非线性函数。如此灵活的模型的一个负面效果是，似然函数所需的潜在变量上的积分无法解析地计算。相反，似然函数可以通过从高斯先验概率分布中采样，使用蒙特卡罗方法近似。这样，在潜在变量上的积分变成了一个简单的求和，求和式中的每一项对应于一个样本。然而，由于为了得到边缘概率分布的一个准确的表示，我们需要相当多的数据点，因此这个方法的计算代价很高。

如果我们考虑非线性函数的一个更加受限的形式，并且恰当地选择离散变量概率分布，那么我们可以建立一个离散变量模型，这个模型是非线性的，并且训练上很高效。生成式地形映射（generative topographic mapping）或者GTM（Bishop et al., 1996; Bishop et al., 1997a; Bishop et al., 1998b）使用一个潜在的概率分布，这个概率分布由潜在空间（通常是二维的）上的delta函数的有限个正規格点定义。这样，在这个潜在空间中的积分只需对每个格点位置上的贡献进行求和即可。非线性映射由一个线性回归模型给出，这个线性模型允许一般的非线性性质，同时使得映射是可调节参数的一个线性函数。注意，由于维度灾难造成的线性回归模型的局限性在GTM中没有出现，因为流形通常具有两个维度，与数据空间的维度无关。这两种选择的一个结果是似然函数可以用封闭的形式解析地表示，可以使用EM算法高效地最优化。生成的GTM模型将一个二维的非线性流形按照数据集进行调节，并且通过计算数据点的潜在空间上的后验概率分布，数据点可以映射回潜在空间，用于数据的可视化。图12.21给出了使用线

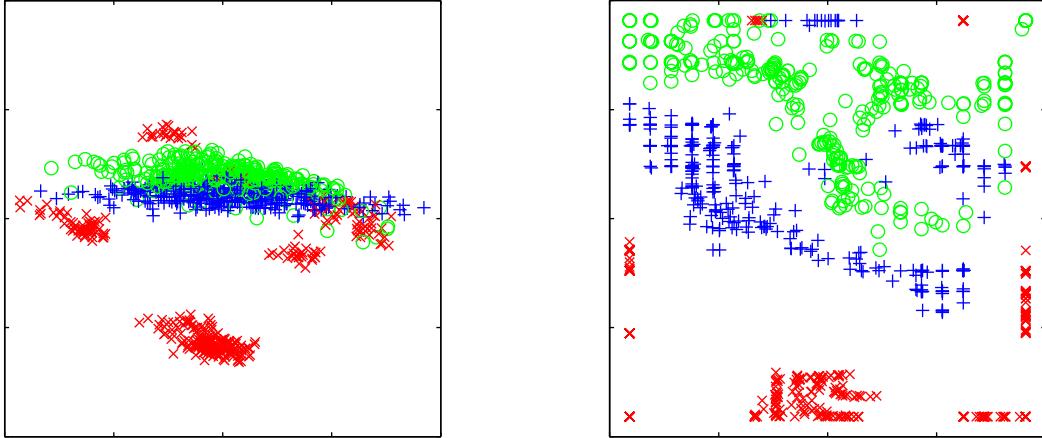


图 12.21: 使用PCA（左图）和GTM（右图）对石油流数据集进行可视化。对于GTM模型，每个数据点都画在了潜在空间的后验概率分布的均值位置。GTM模型的非线性性质使得数据点分组之间的划分可以更明显地看出。

性PCA和非线性GTM对石油流数据进行可视化的对比。

GTM可以被看成一个更早的模型的概率化版本，这个模型被称为自组织映射（self organizing map），或者SOM（Kohonen, 1982; Kohonen, 1995），它也将二维非线性流形表示为离散点的正规数组。SOM与K均值算法有些相似，因为数据点被分配到附近的代表向量中，然后被更新。初始阶段，代表向量被随机分布。在训练阶段，它们“自组织”，来近似一个光滑的流形。然而，与K均值不同，SOM没有优化任何具有良好定义的代价函数（Erwin et al., 1992），使得设置模型的参数以及评估收敛变得十分困难。并且不能保证“自组织”会发生，因为它依赖于对于特定数据集的恰当的参数选择。

相反，GTM最优化对数似然函数，得到的模型定义了数据空间的一个概率密度。事实上，它对应于一个受限的高斯混合，其中各个分量共享一个相同的方差，均值被限制在一个光滑的二维流形中。概率的基础也使得定义GTM的推广形式很容易（Bishop et al., 1998a），例如处理缺失值的贝叶斯方法，对离散变量的一个系统化的推广，使用高斯过程来定义流形，或者层次化GTM模型（Tino and Nabney, 2002）。

由于GTM中的流形被定义为连续曲面，而不像SOM那样仅仅定义一个代表向量，因此可以计算放大因子（magnification factor），对应于拟合数据集时所需的对流形的局部放大或压缩（Bishop et al., 1997b），也可以计算方向曲率（directional curvature）（Tino et al., 2001）。可以使用投影数据进行可视化，并且使我们能够更深刻地认识这个模型。

12.5 练习

(12.1) (***) 本练习中，我们使用归纳法证明在 M 维子空间上的线性投影中，最大化投影数据方差的投影由数据协方差矩阵 S （由公式(12.3)定义）的 M 个特征向量定义，对于 M 个最大的特征值。12.1节证明了 $M = 1$ 的情况下的这个结果。现在假设结果对于某个一般的 M 值成立，证明它对于 $M + 1$ 维也成立。为了证明这一点，首先令投影数据的方差关于定义了数据空间的新方向的向量 u_{M+1} 的导数等于零。这可以通过考虑下面的限制条件完成：向量 u_{M+1} 正交于存在的向量 u_1, \dots, u_M ，且已经被归一化为单位长度。使用拉格朗日乘数法来强制满足这些限制。然后使用向量 u_1, \dots, u_M 的单位正交性质证明新的向量 u_{M+1} 是 S 的一个特征向量。最后，证明如果特征向量被选为对应于 λ_{M+1} 的特征向量，那么方差被最大化，其中特征值按照降序排序。

(12.2) (**) 证明，公式(12.15)给出的PCA失真度量 J 关于 u_i 的最小值，在满足单位正交性的限制条件(12.7)的情况下，出现在 u_i 是数据协方差矩阵 S 的特征向量的情形中。为了证明这一点，引入拉格朗日乘数的矩阵 H ，每个拉格朗日乘数有对应于一个限制条件，从而修改后的失真度量用矩阵的记号表示为

$$\tilde{J} = \text{Tr} \left\{ \hat{U}^T S \hat{U} \right\} + \text{Tr} \left\{ H(I - \hat{U}^T \hat{U}) \right\} \quad (12.93)$$

其中 $\widehat{\mathbf{U}}$ 是一个 $D \times (D - M)$ 的矩阵，列为 \mathbf{u}_i 。现在关于 $\widehat{\mathbf{U}}$ 最小化 \tilde{J} ，证明解满足 $\mathbf{S}\widehat{\mathbf{U}} = \widehat{\mathbf{U}}\mathbf{H}$ 。很明显，一个可能的解是 $\widehat{\mathbf{U}}$ 的列是 \mathbf{S} 的特征向量，此时 \mathbf{H} 是一个对角矩阵，包含对应的特征值。为了得到一般的解，证明 \mathbf{H} 可以被假设为对称矩阵，然后使用特征向量展开证明 $\mathbf{S}\widehat{\mathbf{U}} = \widehat{\mathbf{U}}\mathbf{H}$ 得到的 \tilde{J} 与 $\widehat{\mathbf{U}}$ 的列是 \mathbf{S} 的特征向量这一具体的解对应的 \tilde{J} 相同。由于这些解都是独立的，因此比较方便的做法是选择特征向量解。

(12.3) (*) 验证公式 (12.30) 定义的特征向量被归一化为单位长度，假设特征向量 \mathbf{v}_i 具有单位长度。

(12.4) (*) 假设我们使用一个一般的高斯分布 $\mathcal{N}(\mathbf{z} | \mathbf{m}, \Sigma)$ 来代替概率PCA模型中的零均值单位协方差潜在空间概率分布 (12.31)。通过重新定义模型的参数，证明，对于 \mathbf{m} 和 Σ 的任意合法的选择，这个模型都会得到完全相同的观测变量上的边缘概率 $p(\mathbf{x})$ 。

(12.5) (**) 令 \mathbf{x} 为 D 维随机变量，服从高斯分布 $\mathcal{N}(\mathbf{x} | \boldsymbol{\mu}, \boldsymbol{\Sigma})$ ，考虑 $\mathbf{y} = \mathbf{Ax} + \mathbf{b}$ 定义的 M 维随机变量，其中 \mathbf{A} 是一个 $M \times D$ 的矩阵。证明， \mathbf{y} 也是一个高斯分布，并且寻找它的均值和协方差的表达式。讨论 $M < D$, $M = D$ 以及 $M > D$ 时的这个高斯分布的形式。

(12.6) (*) 画出 12.2 节介绍的概率PCA模型的有向概率图，其中观测变量 \mathbf{x} 的分量显式地表示为单独的结点。从而证明，概率PCA模型与 8.2.2 节讨论的朴素贝叶斯模型具有相同的独立性结构。

(12.7) (**) 通过使用一般的分布的均值和协方差的结果 (2.270) 和 (2.271)，推导概率PCA模型中的边缘概率分布 $p(\mathbf{x})$ 的结果 (12.35)。

(12.8) (**) 通过使用公式 (2.116) 给出的结果，证明概率PCA模型的后验概率分布 $p(\mathbf{z} | \mathbf{x})$ 为 (12.42)。

(12.9) (*) 验证，对概率PCA模型的对数似然函数 (12.43) 关于参数 $\boldsymbol{\mu}$ 进行最大化会得到结果 $\boldsymbol{\mu}_{ML} = \bar{\mathbf{x}}$ ，其中 $\bar{\mathbf{x}}$ 是数据向量的均值。

(12.10) (**) 通过计算概率PCA模型的对数似然函数 (12.43) 关于参数 $\boldsymbol{\mu}$ 的二阶导数，证明驻点 $\boldsymbol{\mu}_{ML} = \bar{\mathbf{x}}$ 表示唯一的最大值。

(12.11) (**) 证明，在极限 $\sigma^2 \rightarrow 0$ 的情况下，概率PCA模型的后验均值会变为主子空间的正交投影，与传统的PCA相同。

(12.12) (**) 对于 $\sigma^2 > 0$ ，证明，与正交投影相比，概率PCA模型的后验均值会向着原点偏移。

(12.13) (**) 证明，根据传统PCA的最小平方投影代价，在概率PCA模型下，一个数据点的最优重建为

$$\tilde{\mathbf{x}} = \mathbf{W}_{ML}(\mathbf{W}_{ML}^T \mathbf{W}_{ML})^{-1} \mathbf{M} \mathbb{E}[\mathbf{z} | \mathbf{x}] \quad (12.94)$$

(12.14) (*) M 维潜在空间和 D 维数据空间的概率PCA模型的协方差矩阵中的独立参数的数量有公式 (12.51) 给出。验证在 $M = D - 1$ 的情况下，独立参数的数量与一般的高斯分布的协方差相同，而对于 $M = 0$ 的情形，它与各向同性的高斯分布的协方差相同。

(12.15) (**) 通过对完整数据对数似然函数的期望 (12.53) 进行最大化，推导概率PCA模型的 M 步骤方程 (12.56) 和 (12.57)。

(12.16) (***) 在图 12.11 中，我们给出了概率PCA模型的一个应用，数据集里的某些数据值随机缺失。推导在这种情况下最大化概率PCA模型的似然函数的EM算法。注意， $\{\mathbf{z}_n\}$ 以及属于向量 $\{\mathbf{x}_n\}$ 的分量的缺失数据现在都是潜在变量。证明在所有数据值都被观测到的具体情况下，这就简化为了 12.2.2 节推导的概率PCA模型的EM算法。

(12.17) (**) 令 \mathbf{W} 是一个 $D \times M$ 的矩阵，它的列定义了镶嵌在 D 维数据空间中的一个 M 维线性子空间，令 $\boldsymbol{\mu}$ 是一个 D 维向量。给定一个数据集 $\{\mathbf{x}_n\}$ ，其中 $n = 1, \dots, N$ ，我们可以使用 M 维向量的集合 $\{\mathbf{z}_n\}$ 上的一个线性映射近似数据点，从而 \mathbf{x}_n 由 $\mathbf{W}\mathbf{z}_n + \boldsymbol{\mu}$ 近似。关联的平方和重建代价为

$$J = \sum_{n=1}^N \|\mathbf{x}_n - \boldsymbol{\mu} - \mathbf{W}\mathbf{z}_n\|^2 \quad (12.95)$$

首先证明 J 关于 $\boldsymbol{\mu}$ 的最小化会产生一个类似的表达式，其中 \mathbf{x}_n 和 \mathbf{z}_n 分别被替换为零均值变量 $\mathbf{x}_n - \bar{\mathbf{x}}$ 和 $\mathbf{z}_n - \bar{\mathbf{z}}$ ， $\bar{\mathbf{x}}$ 和 $\bar{\mathbf{z}}$ 表示样本均值。然后，证明 J 关于 \mathbf{z}_n 的最小化（保持 \mathbf{W} 固定）会得到PCA的 E 步骤 (12.58)， J 关于 \mathbf{W} 最小化（保持 $\{\mathbf{z}_n\}$ 固定）会得到PCA的 M 步骤 (12.59)。

(12.18) (*) 推导 12.2.4 节描述的因子分析模型的独立参数数量的表达式。

(12.19) (***) 证明12.2.4节描述的因子分析模型对于潜在空间的坐标系旋转具有不变性。

(12.20) (**) 通过考虑二阶导数，证明12.2.4节讨论的因子分析模型的对数似然函数的关于参数 μ 的唯一驻点是公式(12.1)定义的样本均值。并且证明这个驻点是一个最大值点。

(12.21) (**) 推导因子分析模型的EM算法的E步骤公式(12.66)和(12.67)。注意，根据练习12.20的结果，参数 μ 可以被替换为样本均值 \bar{x} 。

(12.22) (**) 写出因子分析模型的完整数据对数似然函数的期望的表达式，从而推导出对应的M步骤方程(12.69)和(12.70)。

(12.23) (*) 画出表示概率PCA模型的离散混合的有向概率图模型，其中每个PCA模型有自己的 W , μ 和 σ^2 的值。现在，画一个修改的图，其中这些参数值被混合分布中的各个分量所共享。

(12.24) (****) 在2.3.7节，我们看到学生t分布可以被看做高斯分布的无穷混合，其中我们关于连续潜在变量进行积分。通过使用这种表示方法，形式化地表示给定一组观测数据点的条件下的多元学生t分布的对数似然函数进行最大化的EM算法，并推导E步骤方程和M步骤方程的形式。

(12.25) (**) 考虑一个线性高斯潜在变量模型，它具有潜在空间分布 $p(z) = \mathcal{N}(z | \mathbf{0}, \mathbf{I})$ ，以及观测变量上的条件概率分布 $p(x | z) = \mathcal{N}(x | Wz + \mu, \Phi)$ ，其中 Φ 是一个任意的对称正定噪声协方差矩阵。现在假设我们对数据变量进行一个非奇异的线性变换 $x \rightarrow Ax$ ，其中 A 是一个 $D \times D$ 的矩阵。如果 μ_{ML} , W_{ML} 和 Φ_{ML} 表示对应于原始的未变换数据的最大似然解，证明 $A\mu_{ML}$, AW_{ML} 和 $A\Phi_{ML}A^T$ 表示变换后的数据集的对应的最大似然解。最后，证明模型的形式在下面两种情况下具有不变性：(1) A 是一个对角矩阵， Φ 是一个对角矩阵。这对应于因子分析的情形。变换后的 Φ 仍然是对角的，因此因子分析在数据变量的分量之间的重新缩放是共同变化的(covariant)。(2) A 是正交矩阵， Φ 正比于单位矩阵，即 $\Phi = \sigma^2 \mathbf{I}$ 。这对应于概率PCA。变换后的矩阵 Φ 仍然正比于单位矩阵，因此概率PCA在数据空间的坐标轴的旋转下是共同变化的，这与传统的PCA的情形相同。

(12.26) (**) 证明，满足(12.80)的任意向量 a_i 也满足(12.79)。并且证明对于(12.80)的任意具有 λ 特征值的解，我们可以加上具有零特征值的 K 的特征向量的任意倍数，得到(12.79)的一个解，它也具有特征值 λ 。最后，证明这样的修改不会影响公式(12.82)给出的主成分投影。

(12.27) (**) 证明，在核PCA中，如果我们选择 $k(x, x') = x^T x'$ 的核，那么传统的PCA会被作为一个具体的实例恢复出来。

(12.28) (**) 使用概率密度在变量替换下的变换性质(1.27)，证明，任意的概率密度 $p(y)$ 都可以从一个固定的处处非零的概率密度 $q(x)$ 中得到，方法是进行一个非线性的变量替换 $y = f(x)$ ，其中 $f(x)$ 是一个单调递增的函数，从而 $0 \leq f'(x) < \infty$ 。写出 $f(x)$ 满足的微分方程，画图说明概率密度的变换。

(12.29) (**) 假设两个变量 z_1 和 z_2 是独立的，从而 $p(z_1, z_2) = p(z_1)p(z_2)$ 。证明变量之间的协方差矩阵是对角矩阵。这表明独立性对于两个变量不相关是一个充分条件。现在考虑两个变量 y_1 和 y_2 ，其中 y_1 在0附近对称分布，且 $y_2 = y_1^2$ 。写出条件概率分布 $p(y_2 | y_1)$ 。我们观察到它是依赖于 y_1 的，证明两个变量不是独立的。现在证明两个变量之间的协方差矩阵同样是对角的。为了证明这一点，使用关系 $p(y_1, y_2) = p(y_1)p(y_2 | y_1)$ 证明非对角线项是零。这个反例证明了零相关性不是条件独立的充分条件。

13 顺序数据

本书目前为止，我们主要的注意力集中在数据集里的数据点是独立同分布的情形。这个假设使得我们将似然函数表示为在每个数据点处计算的概率分布在所有数据点上的乘积。然而，对于许多应用来说，独立同分布的假设不成立。这里，我们考虑这样的数据集中一个重要的类型，即描述了顺序数据的数据集。这些数据集通常产生于沿着时间序列进行的测量，例如某个特定位置的连续若干天的降水量测量，或者每天汇率的值，或者对于语音识别任务，在连续的时间框架下的声学特征。图13.1给出了一个涉及到语音数据的例子。顺序数据也可以在时间序列以外的问题中出现，例如一段DNA上的碱基对序列，或者一个英语句子中的字符序列。方便起见，我们有时会用“过去”观测或者“未来”观测来称呼某个观测。然而，本章中研究的模型同样适用于所有形式的顺序数据，而不仅仅是时间序列数据。

区分静止顺序分布和非静止顺序分布是很有用的。在静止分布中，数据会随着时间发生变化，但是生成数据的概率分布保持不变。对于更复杂的非静止分布的情形，生成概率本身会随着时间变化。这里，我们关注的是静止分布的情形。

对于许多应用来说，例如金融预测，我们希望能够在给定时间序列中的前一个观测值的条件下，预测下一个观测值。直觉上讲，我们会猜想，与历史的观测相比，当前的观测值会为预测未来值提供更多的信息。图13.1的例子表明，语音谱的连续观测确实具有高度的相关性。此外，考虑未来的观测对所有之前的观测的一个一般的依赖关系是不现实的，因为这样一个模型的复杂度会随着观测数量的增加而无限制地增长。这使得我们要考虑马尔科夫模型（Markov model），其中我们假定未来的预测仅与最近的观测有关，而独立于其他所有的观测。

虽然这样的模型可以计算，但是仍然具有很严重的局限性。通过引入潜在变量，我们可以得到一个更加一般的框架，同时仍然保持计算上的可处理性，这就引出了状态空间模型（state space model）。与第9章和第12章一样，我们会看到复杂的模型可以从简单的成分中构建，特别地，从指数族分布中构建，并且可以使用概率图模型的框架进行描述。这里，我们关注状态空间模型的两个最重要的例子，即隐马尔可夫模型（hidden Markov model），其中潜在变量是离散的，以及线性动态系统（linear dynamical system），其中潜在变量服从高斯分布。这两个模型都使用具有树结构（没有环）的有向图描述，这样就可以使用加和-乘积算法来高效地进行推断。

13.1 马尔科夫模型

处理顺序数据的最简单的方式是忽略顺序的性质，将观测看做独立同分布，对应于图13.2所示的图。然而，这种方法无法利用数据中的顺序模式，例如序列中距离较近的观测之间的相关性。例如，假设我们观测一个二值变量，这个二值变量表示某一天是否下雨。给定这个变量的一系列观测，我们希望预测下一天是否会下雨。如果我们将所有的数据都看成独立同分布的，那么我们能够从数据中得到的唯一的信息就是雨天的相对频率。然而，在实际生活中，我们知道天气经常会呈现出持续若干天的趋势。因此，观测到今天是否下雨对于预测明天是否下雨会有极大的帮助。

为了在概率模型中表示这种效果，我们需要放松独立同分布的假设。完成这件事的一种最简单的方式是考虑马尔科夫模型（Markov model）。首先我们注意到，不失一般性，我们可以使用概率的乘积规则来表示观测序列的联合概率分布，形式为

$$p(\mathbf{x}_1, \dots, \mathbf{x}_N) = p(\mathbf{x}_1) \prod_{n=2}^N p(\mathbf{x}_n | \mathbf{x}_1, \dots, \mathbf{x}_{n-1}) \quad (13.1)$$

如果我们现在假设右侧的每个条件概率分布只与最近的一次观测有关，而独立于其他所有之前的观测，那么我们就得到了一阶马尔科夫链（first-order Markov chain），如图13.3所示。这个模型中， N 次观测的序列的联合概率分布为

$$p(\mathbf{x}_1, \dots, \mathbf{x}_N) = p(\mathbf{x}_1) \sum_{n=2}^N p(\mathbf{x}_n | \mathbf{x}_{n-1}) \quad (13.2)$$



图 13.1: 单词“Bayes' theorem”的声音分析图，画出了谱系数的强度与时间的关系。

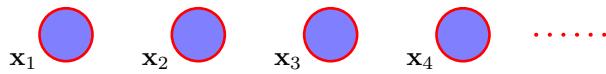


图 13.2: 对顺序观测建模的最简单的方法是将它们看做独立的，对应于没有链接的图。

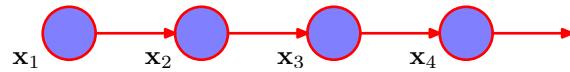


图 13.3: 观测 $\{x_n\}$ 的一阶马尔科夫链，其中，特定的观测 x_n 的条件概率分布 $p(x_n | x_{n-1})$ 只以前一次观测 x_{n-1} 为条件。

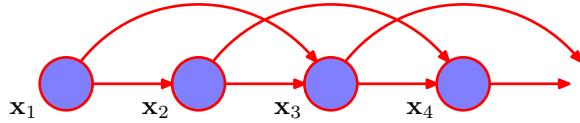


图 13.4: 一个二阶马尔科夫链，其中特定的观测 x_n 依赖于前两次观测 x_{n-1} 和 x_{n-2} 的值。

根据d-划分的性质，给定时刻 n 之前的所有观测，我们看到观测 x_n 的条件概率分布为

$$p(x_n | x_1, \dots, x_{n-1}) = p(x_n | x_{n-1}) \quad (13.3)$$

从公式 (13.2) 开始，使用概率的乘积规则，这个等式很容易验证。因此如果我们使用这样的模型预测序列中的下一次观测，那么预测分布只依赖于最近的一次观测的值，而与所有更早的观测都无关。

在这种模型的大部分应用中，条件概率分布 $p(x_n | x_{n-1})$ 被限制为相等的，对应于静止时间序列的假设。这样，这个模型被称为同质马尔科夫链 (homogeneous Markov chain)。例如，如果条件概率分布依赖于可调节的参数（参数的值可以从训练数据中确定），那么链中所有的条件概率分布会共享相同的参数值。

虽然这比独立的模型要一般一些，但是仍然非常受限。对于许多顺序的观测来说，我们预计若干个连续观测的数据的趋势会为下一次预测提供重要的信息。一种让更早的观测产生影响的方法是使用高阶的马尔科夫链。如果我们允许预测除了与当前观测有关以外，还与当前观测的前一次观测有关，那么我们就得到了二阶马尔科夫链，如图13.4所示。现在，联合概率分布为

$$p(x_1, \dots, x_N) = p(x_1)p(x_2 | x_1) \prod_{n=3}^N p(x_n | x_{n-1}, x_{n-2}) \quad (13.4)$$

与之前一样，使用d-划分或者直接计算，我们看到给定 x_{n-1} 和 x_{n-2} 的条件下 x_n 的条件概率分布与所有的 x_1, \dots, x_{n-3} 的观测无关。现在，每次观测由之前的两次观测所影响。我们可以类似地考虑扩展到 M 阶马尔科夫链，其中一个特定的变量依赖于前 M 个变量。然而，这种增长的灵活性是有代价的，因为现在模型中参数的数量要多得多。假设观测是具有 K 个状态的离散变量，那么一阶马尔科夫链中的条件概率分布 $p(x_n | x_{n-1})$ 由 $K - 1$ 个参数指定，每个参数都对应于 x_{n-1} 的 K 个状态，因此参数的总数为 $K(K - 1)$ 。现在假设我们将模型推广到 M 阶马尔科夫链，从而联合概率分布由条件概率分布 $p(x_n | x_{n-M}, \dots, x_{n-1})$ 构建。如果变量是离散变量，且条件概率分布使用一般的条件概率表的形式表示，那么这种模型中参数的数量为 $K^M(K - 1)$ 。由于这个量随着 M 指数增长，因此通常对于大的 M 来说，使用这种方法是不实际的。

对于连续变量来说，我们可以使用线性高斯条件概率分布，其中每个结点都是一个高斯概率分布，均值是父结点的一个线性函数。这被称为自回归 (autoregressive) 模型或者AR模型 (Box et al., 1994; Thiesson et al., 2004)。另一种方法是为 $p(x_n | x_{n-M}, \dots, x_{n-1})$ 使用参数化的模型，例如神经网络。这种方法有时被称为抽头延迟线 (tapped delay line)，因为它对应于存储 (延迟) 观测变量的前面 M 个值来预测下一个值。这样，参数的数量远远小于一个一般的模型（例如此时参数的数量可能随着 M 线性增长），虽然这样做会使得条件概率分布被限制在一个特定的类别中。

假设我们希望构造任意阶数的不受马尔科夫假设限制的序列模型，同时能够使用较少数量的自由参数确定。我们可以引入额外的潜在变量来使得更丰富的一类模型能够从简单的成分中构建，正如我们在第9章讨论混合概率分布和第12章讨论连续潜在变量模型时所做的那样。对于每个观测 x_n ，我们引入一个对应的潜在变量 z_n （类型或维度可能与观测变量不同）。我们现在假设潜在变量构成了马尔科夫链，得到的图结构被称为状态空间模型 (state space model)，如图 13.5 所示。它满足下面的关键的条件独立性质，即给定 z_n 的条件下， z_{n-1} 和 z_{n+1} 是独立的，从而

$$z_{n+1} \perp\!\!\!\perp z_{n-1} \mid z_n \quad (13.5)$$

这个模型的联合概率分布为

$$p(x_1, \dots, x_N, z_1, \dots, z_N) = p(z_1) \left[\prod_{n=2}^N p(z_n | z_{n-1}) \right] \prod_{n=1}^N p(x_n | z_n) \quad (13.6)$$

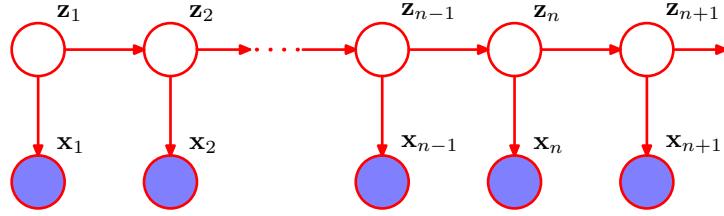


图 13.5: 我们可以使用潜在变量的马尔科夫链来表示顺序数据，每个观测都以对应的潜在变量的状态为条件。这个重要的图结构组成了隐马尔科夫模型和线性动态系统的基础。

使用d-划分准则，我们看到总存在一个路径通过潜在变量连接了任意两个观测变量 x_n 和 x_m ，并且这个路径永远不会被阻隔。因此对于观测变量 x_{n+1} 来说，给定所有之前的观测，条件概率分布 $p(x_{n+1} | x_1, \dots, x_n)$ 不会表现出任何的条件独立性，因此我们对 x_{n+1} 的预测依赖于所有之前的观测。然而，观测变量不满足任何阶数的马尔科夫性质。我们在本章的后面几节会讨论如何计算预测分布。

对于顺序数据来说，这个图描述了两个重要的模型。如果潜在变量是离散的，那么我们得到了隐马尔科夫模型 (hidden Markov model) 或者HMM (Elliott et al., 1995)。注意，HMM中的观测变量可以是离散的或者是连续的，并且可以使用许多不同的条件概率分布进行建模。如果潜在变量和观测变量都是高斯变量（结点的条件概率分布对于父结点的依赖是线性高斯的形式），那么我们就得到了线性动态系统 (linear dynamical system)。

13.2 隐马尔科夫模型

隐马尔科夫模型可以被看成图13.5所示的状态空间模型的一个具体实例，其中潜在变量是离散的。然而，如果我们考察模型的一个单一的时间切片，那么我们看到它对应于一个混合概率分布，对应的分量密度为 $p(x | z)$ 。于是，它也可以表述为混合概率模型的一个推广，其中每个观测的混合系数不是独立地选择的，而是依赖于对于前一次观测的分量的选择。HMM广泛用于语音识别 (Jelinek, 1997; Rabiner and Juang, 1993)、自然语言建模 (Manning and Schütze, 1999)、在线手写识别 (Nag et al., 1986) 以及生物序列（例如蛋白质和DNA）的分析 (Krogh et al., 1994; Durbin et al., 1998; Baldi and Brunak, 2001)。

与标准的混合模型的情形相同，潜在变量是服从多项式分布的变量 z_n ，描述了那个混合分量用于生成对应的观测 x_n 。与之前一样，比较方便的做法是使用1-of- K 表示方法，就像第9章那样。我们现在让 z_n 的概率分布通过条件概率分布 $p(z_n | z_{n-1})$ 对前一个潜在变量 z_{n-1} 产生依赖。由于潜在变量是 K 维二值变量，因此条件概率分布对应于数字组成的表格，记作 \mathbf{A} ，它的元素被称为转移概率 (transition probabilities)。元素为 $A_{jk} \equiv p(z_{nk} = 1 | z_{n-1,j} = 1)$ 。由于它们是概率值，因此满足 $0 \leq A_{jk} \leq 1$ 且 $\sum_k A_{jk} = 1$ ，从而矩阵 \mathbf{A} 有 $K(K - 1)$ 个独立的参数。这样，我们可以显式地将条件概率分布写成

$$p(z_n | z_{n-1}, \mathbf{A}) = \prod_{k=1}^K \prod_{j=1}^K A_{jk}^{z_{n-1,j} z_{nk}} \quad (13.7)$$

初始潜在结点 z_1 很特别，因为它没有父结点，因此它的边缘概率分布 $p(z_1)$ 由一个概率向量 $\boldsymbol{\pi}$ 表示，元素为 $\pi_k \equiv p(z_{1k} = 1)$ ，即

$$p(z_1 | \boldsymbol{\pi}) = \prod_{k=1}^K \pi_k^{z_{1k}} \quad (13.8)$$

其中 $\sum_k \pi_k = 1$ 。

有时可以将状态画成状态转移图中的一个结点，这样就可以图形化地表示出转移矩阵。图13.6给出了 $K = 3$ 的情形。注意，这不是一个概率图模型，因为结点不是单独的变量而是一个变量的各个状态，因此我们用方框而不是圆圈来表示状态。

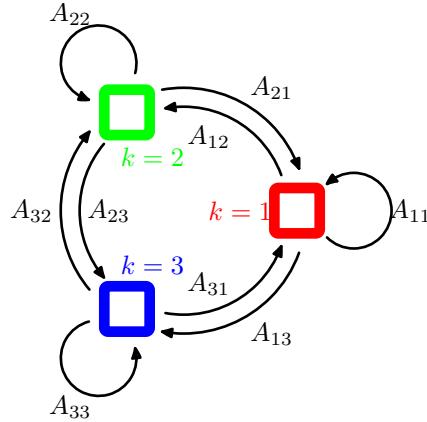


图 13.6: 转移图表示一个模型，它的潜在变量有三种可能的状态，对应于三个方框。黑线表示转移矩阵的元素 A_{jk} 。

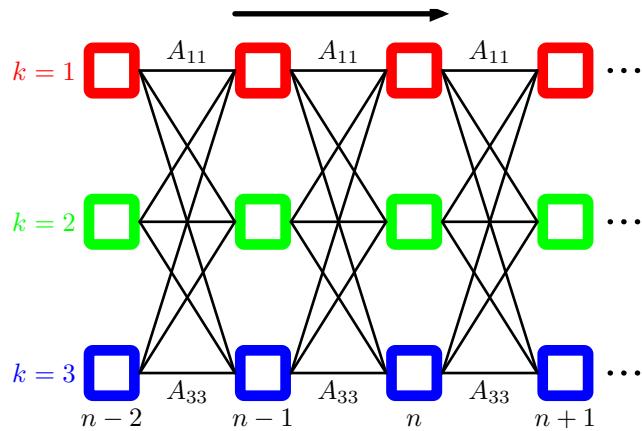


图 13.7: 如果我们将图13.6所示的状态转移图在时间上展开，那么我们旧得到了潜在状态的晶格图表示或者格子图表示。图的每一列对应于一个潜在变量 z_n 。

有时比较有用的做法是将图13.6所示的状态转移图在时间上展开。这给出了潜在变量之间转移的另一种表示方法，被称为晶格图（lattice diagram）或者格子图（trellis diagram）。图13.7给出了隐马尔科夫模型的晶格图。

可以通过定义观测变量的条件概率分布 $p(\mathbf{x}_n | z_n, \phi)$ 来确定一个概率模型，其中 ϕ 是控制概率分布的参数集合。这些条件概率被称为发射概率（emission probabilities），可以是例如 (9.11) 这样的高斯分布（ x 是连续变量），也可以是条件概率表格（ x 是离散变量）。由于 \mathbf{x}_n 是观测值，因此对于一个给定的 ϕ 值，概率分布 $p(\mathbf{x}_n | z_n, \phi)$ 由一个 K 维的向量组成，对应于二值向量 z_n 的 K 个可能的状态。我们可以将发射概率表示为

$$p(\mathbf{x}_n | z_n, \phi) = \prod_{k=1}^K p(\mathbf{x}_n | \phi_k)^{z_{nk}} \quad (13.9)$$

我们将注意力集中在同质的（homogeneous）模型上，其中所有控制潜在变量的条件概率分布都共享相同的参数 \mathbf{A} ，类似地所有发射概率分布都共享相同的参数 ϕ （推广到更一般的情形很容易）。注意，对于一个独立同分布的数据集，一个混合模型对应于参数 A_{jk} 对于所有的 j 值都相同的情况，从而条件概率分布 $p(z_n | z_{n-1})$ 与 z_{n-1} 无关。这对应于将图13.5所示的图模型中的水平链接都删除。

从而观测变量和潜在变量上的联合概率分布为

$$p(\mathbf{X}, \mathbf{Z} | \boldsymbol{\theta}) = p(z_1 | \pi) \left[\prod_{n=2}^N p(z_n | z_{n-1}, \mathbf{A}) \right] \prod_{m=1}^M p(\mathbf{x}_m | z_m, \phi) \quad (13.10)$$

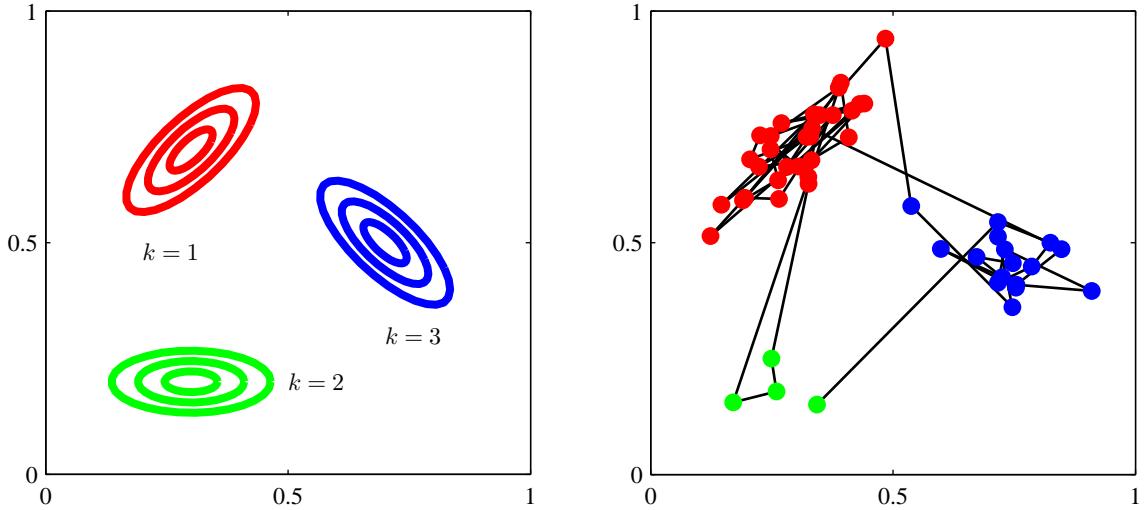


图 13.8: 从一个隐马尔科夫模型中进行采样的例子，这个模型的潜在变量 z 有三个状态，发射概率 $p(x | z)$ 是高斯概率，其中 x 是二维的。(a)发射概率密度为常数的轮廓线，对应于潜在变量的三个状态。(b)从隐马尔科夫模型中抽取的50个样本点，数据点的颜色对应于生成它们的分量的颜色，数据点之间的连线表示连续的观测。这里，转移矩阵是固定的。在任何状态，都有5%的概率转移到每个其他的状态，有90%的概率保持相同的状态。

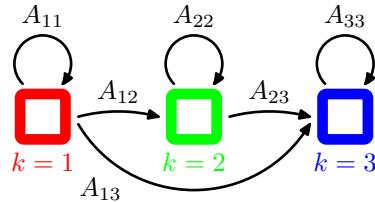


图 13.9: 三状态隐马尔科夫模型的状态转移图的例子。注意，一旦离开了某个状态，就无法再次回到这个状态。

其中 $\mathbf{X} = \{x_1, \dots, x_N\}$, $\mathbf{Z} = \{z_1, \dots, z_N\}$ 和 $\theta = \{\pi, \mathbf{A}, \phi\}$ 表示控制模型参数的集合。我们关于隐马尔科夫模型的大部分讨论与发射概率的特定选择无关。事实上，模型对于一大类发射概率的选择都是可以计算的，包括离散表格、高斯以及混合高斯。也可以利用判别式模型例如神经网络。这些可以用来直接对发射概率密度 $p(x | z)$ 建模，也可以用来给出 $p(z | x)$ 的一个表达式，这个表达式可以使用贝叶斯定理转化为所需的发射概率密度 $p(x | z)$ (Bishop et al., 2004)。

从生成式的观点考虑隐马尔科夫模型，我们可以更好地理解隐马尔科夫模型。回忆一下，为了从一个混合高斯分布中生成样本，我们首先随机算侧一个分量，选择的概率为混合系数 π_k ，然后从对应的高斯分量中生成一个样本向量 x 。这个过程重复 N 次，产生 N 个独立样本组成的数据集。在隐马尔科夫模型的情形，这个步骤修改如下。首先我们选择初始的潜在变量 z_1 ，概率由参数 π_k 控制，然后采样对应的观测 x_1 。现在我们使用已经初始化的 z_1 的值，根据转移概率 $p(z_2 | z_1)$ 来选择变量 z_2 的状态。从而我们以概率 A_{jk} 选择 z_2 的状态 k ，其中 $k = 1, \dots, K$ 。一旦我们知道了 z_2 ，我们就可以对 x_2 采样，从而也可以对下一个潜在变量 z_3 采样，以此类推。这是有向图模型的祖先采样的一个例子。例如，如果我们有一个模型，其中对角转移元素 A_{kk} 比非对角的元素大得多，那么一个典型的数据序列中，会有连续很长的一系列点由同一个概率分布生成，而从一个分量转移到另一个分量不会经常发生。图13.8说明了从隐马尔科夫模型生成样本的过程。

这个标准的HMM模型有很多变体，例如通过对转移矩阵 \mathbf{A} 的形式进行限制的方式进行限制 (Rabiner, 1989)。这里我们介绍一种在实际应用中很重要的变体，被称为从左到右HMM (left-to-right HMM)，它将 \mathbf{A} 中 $k < j$ 的元素 A_{jk} 设置为零。图13.9给出了具有三个

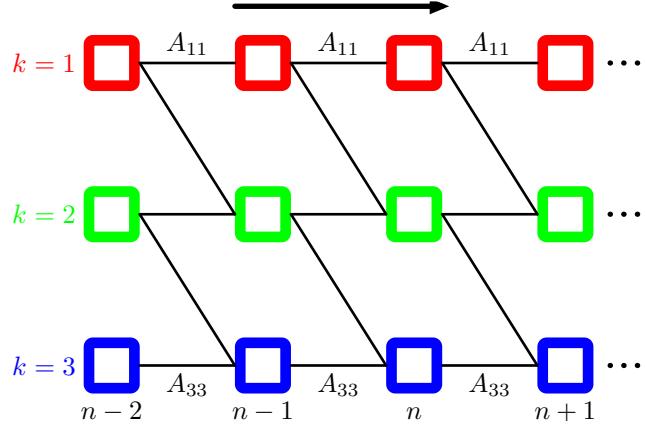


图 13.10: 三状态从左到右HMM的晶格图，其中状态下标 k 在每轮迭代时最多允许加1。

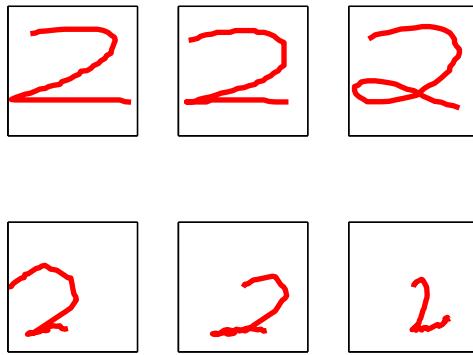


图 13.11: 第一行：在线手写数字的例子。第二行：生成式地采样得到的数字，模型时一个从左到右的隐马尔科夫模型，在45个手写数字组成的数据集上进行训练。

状态的这种HMM变体的状态转移图。通常对于这种模型，初始状态概率 $p(z_1)$ 被修改，使得 $p(z_{1j}) = 1$ 且 $p(z_{1j}) = 0, j \neq 1$ ，换句话说，每个序列被限制为从状态 $j = 1$ 开始。转移矩阵可以进一步被限制，来确保状态的下标不会发生过大的变化，即如果 $k > j + \Delta$ ，那么 $A_{jk} = 0$ 。图13.10给出了这种模型的晶格图。

隐马尔科夫模型的许多应用，例如语音识别或在线字符识别都使用了这种从左到右的结构。作为从左到右隐马尔科夫模型的一个例子，我们考虑手写数字识别的一个例子。这个例子使用在线的数据，即每个手写数字由钢笔的轨迹与时间的函数表示，函数的形式是钢笔坐标的一个序列，这与附录A介绍的离线手写数字的例子不同，那个数据集由二维像素化的图像组成。图13.11给出了在线手写数字的例子。这里，我们在由45个数字“2”的例子组成的数据子集上训练一个马尔科夫模型。有 $K = 16$ 种状态，每个状态可以生成可以生成固定长度的线段，它具有16种可能的角度中的一个，因此发射概率是一个 16×16 的概率表，与每个状态下标的值所允许的角度值相关联。除了那些使得状态下标 k 不变或者加1的转移概率之外，其他的转移概率全部被设置为零。模型使用了25轮的EM迭代进行最优化。通过生成式地运行这个算法，我们可以获得对模型的一些更深刻的认识，如图13.11所示。

隐马尔科夫模型的一个强大的性质是它对于时间轴上局部的变形（压缩和拉伸）具有某种程度的不变性。为了理解这一点，考虑在线手写数字例子中，数字“2”的书写方式。一个通常的手写数字由两个不同的部分组成，两个部分连接处有一个转折点。数字的第一部分从左上方开始，有一个光滑的圆弧，然后向下到转折点，或者在左下角转一个圈，接下来是第二个近似于直线的部分，扫到右下方。书写风格的自然的变化会使得这两个部分的相对大小发生变化。从生成式的观点来看，这种变化可以整合到隐马尔科夫模型中，方法是改变状态模型中保持在同一个状态的转移的数量和在连续的状态之间转移的数量。但是注意，如果数字“2”用相反的顺序

书写，即从右下角开始，结束于左上角，那么即使笔迹的坐标与训练集里的一个例子完全相同，在这个模型下的观测的概率会非常小。在语音识别的问题中，对时间轴的变形与语速的自然变化相关，隐马尔科夫模型可以适应这种变形，不会对这种变形赋予过多的惩罚。

13.2.1 用于HMM的最大似然法

如果我们观测到一个数据集 $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ ，那么我们可以使用最大似然法确定HMM的参数。似然函数通过对联合概率分布 (13.10) 中的潜在变量进行求和的方式得到，即

$$p(\mathbf{X} | \boldsymbol{\theta}) = \sum_{\mathbf{Z}} p(\mathbf{X}, \mathbf{Z} | \boldsymbol{\theta}) \quad (13.11)$$

由于联合概率分布 $p(\mathbf{X}, \mathbf{Z} | \boldsymbol{\theta})$ 无法在 n 上进行分解（与第9章关于混合概率分布的讨论不同），因此我们不能独立地在每个 z_n 上进行求和。我们也不能显示地完成这个求和，因为有 N 个变量需要求和，每个都有 K 个状态，从而总计有 K^N 个求和项。因此求和式中的项的数量随着链的长度指数增长。事实上，公式 (13.11) 中的求和对应于在图13.7的晶格图中通过指数多条路径进行的求和。

我们之前在讨论图8.32所示的简单变量链的推断问题时，已经遇到了一个类似的困难。那里，我们能够使用图的条件独立性质对求和式重新排序，得到一个计算代价与链的长度呈线性关系而不是指数关系的算法。我们将类似的方法应用到隐马尔可夫模型中。

似然函数表达式 (13.11) 的另一个问题是，由于它对应于混合概率分布的一个推广，因此它表示潜在变量的不同配置下，对发射概率进行求和。因此直接对这个似然函数进行最大化会导致复杂的表达式，没有解析解。这一点与简单的混合模型一样（回忆一下，独立同分布数据的混合模型是HMM的一个具体实例）。

于是我们使用期望最大化算法来寻找对隐马尔可夫模型中似然函数进行最大化的有效框架。EM算法的开始阶段是对模型参数的某些初始的选择，我们记作 $\boldsymbol{\theta}^{\text{旧}}$ 。在E步骤中，我们使用这些参数找到潜在变量的后验概率分布 $p(\mathbf{Z} | \mathbf{X}, \boldsymbol{\theta}^{\text{旧}})$ 。然后，我们使用这个后验概率分布计算完整数据似然函数的对数的期望，得到了一个关于参数 $\boldsymbol{\theta}$ 的函数 $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{旧}})$ ，定义为

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{旧}}) = \sum_{\mathbf{Z}} p(\mathbf{Z} | \mathbf{X}, \boldsymbol{\theta}^{\text{旧}}) \ln p(\mathbf{X}, \mathbf{Z} | \boldsymbol{\theta}) \quad (13.12)$$

现在，引入一些记号会比较方便。我们使用 $\gamma(z_n)$ 来表示潜在变量 z_n 的边缘概率分布，用 $\xi(z_{n-1}, z_n)$ 表示两个连续的潜在变量的联合后验概率分布，即

$$\gamma(z_n) = p(z_n | \mathbf{X}, \boldsymbol{\theta}^{\text{旧}}) \quad (13.13)$$

$$\xi(z_{n-1}, z_n) = p(z_{n-1}, z_n | \mathbf{X}, \boldsymbol{\theta}^{\text{旧}}) \quad (13.14)$$

对于每个 n 值，我们可以使用 K 个非负数来存储 $\gamma(z_n)$ ，这些数的和等于 1。类似地，我们可以使用一个由非负数组成的 $K \times K$ 的矩阵来存储 $\xi(z_{n-1}, z_n)$ ，同样加和等于 1。我们也会使用 $\gamma(z_{nk})$ 来表示 $z_{nk} = 1$ 的条件概率，类似地使用 $\xi(z_{n-1,j}, z_{nk})$ 来表示后面介绍的另一个概率。由于二值随机变量的期望就是取值为 1 的概率，因此我们有

$$\gamma(z_{nk}) = \mathbb{E}[z_{nk}] = \sum_{z_n} \gamma(z_n) z_{nk} \quad (13.15)$$

$$\xi(z_{n-1,j}, z_{nk}) = \mathbb{E}[z_{n-1,j} z_{nk}] = \sum_{z_{n-1}, z_n} \xi(z_{n-1}, z_n) z_{n-1,j} z_{nk} \quad (13.16)$$

如果我们将公式 (13.10) 的联合概率分布 $p(\mathbf{X}, \mathbf{Z} | \boldsymbol{\theta})$ 代入公式 (13.12)，使用 γ 和 ξ 的定义，我们有

$$\begin{aligned} Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{旧}}) &= \sum_{k=1}^K \gamma(z_{1k}) \ln \pi_k + \sum_{n=2}^N \sum_{j=1}^K \sum_{k=1}^K \xi(z_{n-1,j}, z_{nk}) \ln A_{jk} \\ &\quad + \sum_{n=1}^N \sum_{k=1}^K \gamma(z_{nk}) \ln p(\mathbf{x}_n | \phi_k) \end{aligned} \quad (13.17)$$

E步骤的目标是高效地计算 $\gamma(z_n)$ 和 $\xi(z_{n-1}, z_n)$ ，我们后面会详细讨论。

在M步骤中，我们关于参数 $\theta = \{\pi, A, \phi\}$ 最大化 $Q(\theta, \theta^{\text{旧}})$ ，其中我们将 $\gamma(z_n)$ 和 $\xi(z_{n-1}, z_n)$ 看做常数。关于 π 和 A 的最大化可以使用拉格朗日乘数法很容易求出，结果为

$$\pi_k = \frac{\gamma(z_{1k})}{\sum_{j=1}^K \gamma(z_{1j})} \quad (13.18)$$

$$A_{jk} = \frac{\sum_{n=2}^K \xi(z_{n-1,j}, z_{nk})}{\sum_{l=1}^K \sum_{n=2}^N \xi(z_{n-1,j}, z_{nl})} \quad (13.19)$$

EM算法在初始化时必须选择 π 和 A 的初始值，这当然应该遵守概率的加和性质。注意，如果将 π 或 A 的任何元素都设为零，那么在接下来的EM更新中也会保持为零。一个典型的初始化步骤是在满足加和限制和非负限制的条件下，为这些参数随机选择初始值。注意，对于从左到右的模型的情形，我们无需对EM的结果进行特别的修改，只需在 A_{jk} 的适当的元素设置为零即可，因为这些元素始终为零。

为了关于 ϕ_k 最大化 $Q(\theta, \theta^{\text{旧}})$ ，我们注意到公式 (13.17) 中，只有最后一项依赖于 ϕ_k ，并且这一项的形式与独立同分布数据的标准混合分布的对应的函数中的数据依赖项完全相同，这一点可以通过与高斯混合模型的公式 (9.40) 进行对比的方式看出来。这里， $\gamma(z_{nk})$ 起着“责任”的作用。如果对于不同的分量，参数 ϕ_k 独立，那么这一项可以分解为一组项的加和形式，每一项对应于一个 k 值，每一项都可以独立地最大化。这样，我们可以简单地最大化发射概率密度 $p(x | \phi_k)$ 的加权的对数似然函数，权值为 $\gamma(z_{nk})$ 。这里，我们假设这个最大化过程可以高效地完成。例如，在高斯发射密度的情形下，我们有 $p(x | \phi_k) = \mathcal{N}(x | \mu_k, \Sigma_k)$ ，最大化函数 $Q(\theta, \theta^{\text{旧}})$ 可得

$$\boldsymbol{\mu}_k = \frac{\sum_{n=1}^N \gamma(z_{nk}) \mathbf{x}_n}{\sum_{n=1}^N \gamma(z_{nk})} \quad (13.20)$$

$$\boldsymbol{\Sigma}_k = \frac{\sum_{n=1}^N \gamma(z_{nk}) (\mathbf{x}_n - \boldsymbol{\mu}_k)(\mathbf{x}_n - \boldsymbol{\mu}_k)^T}{\sum_{n=1}^N \gamma(z_{nk})} \quad (13.21)$$

对于观测变量服从离散多项式分布的情形，观测变量的条件概率分布为

$$p(\mathbf{x} | \mathbf{z}) = \prod_{i=1}^D \prod_{k=1}^K \mu_{ik}^{x_i z_k} \quad (13.22)$$

对应的M步骤方程为

$$\mu_{ik} = \frac{\sum_{n=1}^N \gamma(z_{nk}) x_{ni}}{\sum_{n=1}^N \gamma(z_{nk})} \quad (13.23)$$

对于服从伯努利分布的观测变量，可以得到类似的结果。

EM算法要求有发射概率分布的参数的初始值。一种设置的方式是首先将数据集看成独立同分布的，然后通过最大似然方法调节发射概率密度，之后使用得到的值来初始化EM的参数。

13.2.2 前向后向算法

接下来我们寻找计算 $\gamma(z_{nk})$ 和 $\xi(z_{n-1,j}, z_{nk})$ 的高效的方法，对应于EM算法中的E步骤。图 13.5给出的隐马尔科夫模型的图表示是一棵树，因此我们知道潜在变量的后验概率分布可以使用两阶段的信息传递算法高效地求出。在隐马尔可夫模型这一特定的问题中，这个被称为前向后向算法 (forward-backward) 算法 (Rabiner, 1989)，或者Baum-Welch算法 (Baum, 1972)。事实上，基本算法有几种变体，每个变体都可以根据沿着链传播的信息的精确形式，得到精确的边缘概率 (Jordan, 2007)。我们会关注这些变体中使用最广泛的一个，被称为alpha-beta算法。

前向后向算法除了本身具有重要的实际应用价值以外，还很好地说明了之前章节中介绍的许多概念。因此我们在本节中会给出前向后向算法的一个“传统的”推导，使用概率的加和规则和乘积规则，并且利用由d-划分从对应的图模型中得到的条件独立性质。之后在13.2.3节，我们会看到前向后向算法如何作为8.4.4节讨论的加和-乘积算法的一个具体事例的方式简单地得到。

值得强调的是，潜在变量的后验概率分布的计算与发射概率密度 $p(\mathbf{x} | z)$ 的形式无关，事实上与观测变量是连续的或者离散的也无关。我们所需要的全部东西是对于所有 n 的每个 z_n 值的概率 $p(x_n | z_n)$ 。并且，在本节和下一节中，我们会省略对于模型参数 θ^{HMM} 的显式依赖关系，因为模型参数始终是固定的。

首先，我们写出下面的条件独立性质（Jordan, 2007）。

$$p(\mathbf{X} | z_n) = p(x_1, \dots, x_n | z_n)p(x_{n+1}, \dots, x_N | z_n) \quad (13.24)$$

$$p(x_1, \dots, x_{n-1} | x_n, z_n) = p(x_1, \dots, x_{n-1} | z_n) \quad (13.25)$$

$$p(x_1, \dots, x_{n-1} | z_{n-1}, z_n) = p(x_1, \dots, x_{n-1} | z_{n-1}) \quad (13.26)$$

$$p(x_{n+1}, \dots, x_N | z_n, z_{n+1}) = p(x_{n+1}, \dots, x_N | z_{n+1}) \quad (13.27)$$

$$p(x_{n+2}, \dots, x_N | z_{n+1}, x_{n+1}) = p(x_{n+2}, \dots, x_N | z_{n+1}) \quad (13.28)$$

$$p(\mathbf{X} | z_{n-1}, z_n) = p(x_1, \dots, x_{n-1} | z_{n-1})p(x_n | z_n)p(x_{n+1}, \dots, x_N | z_n) \quad (13.29)$$

$$p(x_{N+1} | \mathbf{X}, z_{N+1}) = p(x_{N+1} | z_{N+1}) \quad (13.30)$$

$$p(z_{N+1} | z_N, \mathbf{X}) = p(z_{N+1}, z_N) \quad (13.31)$$

其中 $\mathbf{X} = \{x_1, \dots, x_N\}$ 。这些关系很容易使用d-划分证明。例如在第二个结果中，我们注意到结点 x_1, \dots, x_{n-1} 中的任何一个结点到结点 x_n 的路径都要通过结点 z_n ，它被观测到。由于所有这种路径都是头到尾的，因此这个条件独立性质一定成立。作为d-划分的一个练习，读者应该花一些时间验证每一条性质。这些关系也可以使用概率的加和规则和乘积规则，从隐马尔科夫模型的联合概率分布中直接证明，但是麻烦得多。

首先让我们计算 $\gamma(z_{nk})$ 。回忆一下，对于离散的服从多项式分布的随机变量，分量的期望值就是这个分量的值为1的概率。因此我们感兴趣的是在给定观测数据 x_1, \dots, x_N 的条件下，计算 z_n 的后验概率分布 $p(z_n | x_1, \dots, x_N)$ 。这表示一个长度为 K 的向量，它的项对应于 z_{nk} 的期望值。使用贝叶斯定理，我们有

$$\gamma(z_n) = p(z_n | \mathbf{X}) = \frac{p(\mathbf{X} | z_n)p(z_n)}{p(\mathbf{X})} \quad (13.32)$$

注意，分母 $p(\mathbf{X})$ 隐式地以HMM的参数 θ^{HMM} 为条件，因此表示似然函数。使用条件独立性质（13.24），以及概率的加和乘积规则，我们有

$$\gamma(z_n) = \frac{p(x_1, \dots, x_n, z_n)p(x_{n+1}, \dots, x_N | z_n)}{p(\mathbf{X})} = \frac{\alpha(z_n)\beta(z_n)}{p(\mathbf{X})} \quad (13.33)$$

其中我们定义了

$$\alpha(z_n) \equiv p(x_1, \dots, x_n, z_n) \quad (13.34)$$

$$\beta(z_n) \equiv p(x_{n+1}, \dots, x_N | z_n) \quad (13.35)$$

$\alpha(z_n)$ 表示观测到时刻 n 及其之前时刻的所有数据和 z_n 的值的联合概率，而 $\beta(z_n)$ 表示在给定 z_n 的条件下，从时刻 $n+1$ 到 N 的所有未来数据的条件概率。与之前一样， $\alpha(z_n)$ 和 $\beta(z_n)$ 都表示 K 个数字组成的集合，每个数字都对应于二值向量 z_n 的“1-of- K ”表示方法的一个可能的配置。我们使用 $\alpha(z_{nk})$ 表示 $z_{nk} = 1$ 时 $\alpha(z_n)$ 的值，对于 $\beta(z_{nk})$ 也有类似的含义。

我们现在推导能够高效计算 $\alpha(z_n)$ 和 $\beta(z_n)$ 的递归关系。与之前一样，我们使用条件独立性质，尤其是（13.25）和（13.26），以及加和规则和乘积规则，得到用 $\alpha(z_{n-1})$ 表示的 $\alpha(z_n)$ ，如

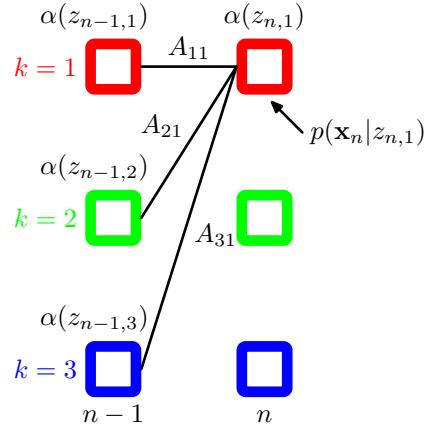


图 13.12: 计算 α 变量的前向递归方程 (13.36) 的说明。在这个晶格图片段中，我们看到 $\alpha(z_{n,1})$ 的计算方式是将 $n-1$ 步的 $\alpha(z_{n-1})$ 的元素 $\alpha(z_{n-1,j})$ 加权求和，权值为 A_{j1} ，对应于 $p(z_n | z_{n-1})$ 的值，然后乘以概率分布 $p(\mathbf{x}_n | z_{n,1})$ 。

下所述。

$$\begin{aligned}
 \alpha(z_n) &= p(\mathbf{x}_1, \dots, \mathbf{x}_n, z_n) \\
 &= p(\mathbf{x}_1, \dots, \mathbf{x}_n | z_n)p(z_n) \\
 &= p(\mathbf{x}_n | z_n)p(\mathbf{x}_1, \dots, \mathbf{x}_{n-1} | z_n)p(z_n) \\
 &= p(\mathbf{x}_n | z_n)p(\mathbf{x}_1, \dots, \mathbf{x}_{n-1}, z_n) \\
 &= p(\mathbf{x}_n | z_n) \sum_{z_{n-1}} p(\mathbf{x}_1, \dots, \mathbf{x}_{n-1}, z_{n-1}, z_n) \\
 &= p(\mathbf{x}_n | z_n) \sum_{z_{n-1}} p(\mathbf{x}_1, \dots, \mathbf{x}_{n-1}, z_n | z_{n-1})p(z_{n-1}) \\
 &= p(\mathbf{x}_n | z_n) \sum_{z_{n-1}} p(\mathbf{x}_1, \dots, \mathbf{x}_{n-1} | z_{n-1})p(z_n | z_{n-1})p(z_{n-1}) \\
 &= p(\mathbf{x}_n | z_n) \sum_{z_{n-1}} p(\mathbf{x}_1, \dots, \mathbf{x}_{n-1}, z_{n-1})p(z_n | z_{n-1})
 \end{aligned}$$

使用公式 (13.34) 给出的 $\alpha(z_n)$ 的定义，我们有

$$\alpha(z_n) = p(\mathbf{x}_n | z_n) \sum_{z_{n-1}} \alpha(z_{n-1})p(z_n | z_{n-1}) \quad (13.36)$$

值得花时间仔细研究一下这个递归的关系。注意，求和式中有 K 项，右侧必须对 z_n 的 K 个值中的每一个进行计算，因此 α 递归的每一步的计算代价为 $O(K^2)$ 。图13.12用晶格图说明了 $\alpha(z_n)$ 的递归方程。

为了开始这个递归过程，我们需要一个初始条件，形式为

$$\alpha(z_1) = p(\mathbf{x}_1, z_1) = p(z_1)p(\mathbf{x}_1 | z_1) = \prod_{k=1}^K \{\pi_k p(\mathbf{x}_1 | \phi_k)\}^{z_{1k}} \quad (13.37)$$

这表明对于 $k = 1, \dots, K$ ， $\alpha(z_{1k})$ 的值为 $\pi_k p(\mathbf{x}_1 | \phi_k)$ 。从链的第一个结点开始，我们可以沿着链计算每个潜在结点的 $\alpha(z_n)$ 。由于递归的每一步涉及到与一个 $K \times K$ 的矩阵相乘，因此计算整个链的这些量的整体代价是 $O(K^2 N)$ 。

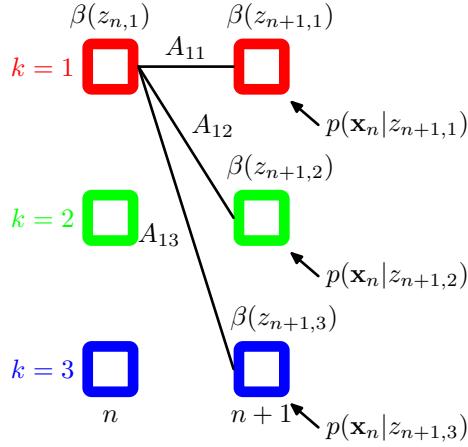


图 13.13: 计算 β 变量的后向递归方程 (13.38) 的说明。在这个晶格图片段中, 我们看到 $\beta(z_{n,1})$ 的计算方式是将 $n+1$ 步的 $\beta(z_{n+1})$ 的元素 $\beta(z_{n+1,k})$ 加权求和, 权值为 A_{ik} (对应于 $p(z_{n+1} | z_n)$) 与发射概率密度 $p(x_n | z_{n+1,k})$ 的对应值的乘积。

类似地我们可以使用条件独立性质 (13.27) 和 (13.28) 得到 $\beta(z_n)$ 的递归关系, 即

$$\begin{aligned}\beta(z_n) &= p(x_{n+1}, \dots, x_N | z_n) \\ &= \sum_{z_{n+1}} p(x_{n+1}, \dots, x_N, z_{n+1} | z_n) \\ &= \sum_{z_{n+1}} p(x_{n+1}, \dots, x_N | z_n, z_{n+1}) p(z_{n+1} | z_n) \\ &= \sum_{z_{n+1}} p(x_{n+1}, \dots, x_N | z_{n+1}) p(z_{n+1} | z_n) \\ &= \sum_{z_{n+1}} p(x_{n+2}, \dots, x_N | z_{n+1}) p(x_{n+1} | z_{n+1}) p(z_{n+1} | z_n)\end{aligned}$$

使用公式 (13.35) 给出的 $\beta(z_n)$ 的定义, 我们有

$$\beta(z_n) = \sum_{z_{n+1}} \beta(z_{n+1}) p(x_{n+1} | z_{n+1}) p(z_{n+1} | z_n) \quad (13.38)$$

注意, 在这种情况下, 我们得到了一个后向信息传递算法, 它根据 $\beta(z_{n+1})$ 计算 $\beta(z_n)$ 。在每一步中, 我们通过发射概率 $p(x_{n+1} | z_{n+1})$ 将观测 x_{n+1} 的效果吸收进来, 然后对 z_{n+1} 求和。图 13.13 说明了这个过程。

与之前一样, 我们需要一个递归的起始条件, 即 $\beta(z_N)$ 的一个值。可以这样获得: 令公式 (13.33) 中的 $n = N$, 然后使用定义 (13.34) 代替 $\alpha(z_N)$, 可得

$$p(z_N | \mathbf{X}) = \frac{p(\mathbf{X}, z_N) \beta(z_N)}{p(\mathbf{X})} \quad (13.39)$$

只要我们对于所有的 z_N 都有 $\beta(z_N) = 1$, 这个结果就是正确的。

在M步方程中, $p(\mathbf{X})$ 可以消去。例如, (13.20) 给出 μ_k 的M步骤方程的形式为

$$\boldsymbol{\mu}_k = \frac{\sum_{n=1}^N \gamma(z_{nk}) \mathbf{x}_n}{\sum_{n=1}^N \gamma(z_{nk})} = \frac{\sum_{n=1}^N \alpha(z_{nk}) \beta(z_{nk}) \mathbf{x}_n}{\sum_{n=1}^N \alpha(z_{nk}) \beta(z_{nk})} \quad (13.40)$$

然而, $p(\mathbf{X})$ 表示似然函数, 我们通常在EM优化过程中能够监视它的值, 因此能够计算出这个值是很有用的。如果我们将 (13.33) 的两侧对 z_n 求和, 使用左侧是一个归一化分布的事实, 我们有

$$p(\mathbf{X}) = \sum_{z_n} \alpha(z_n) \beta(z_n) \quad (13.41)$$

因此我们可以任意选择方便的 n , 通过计算这个求和式计算似然函数。例如, 如果我们只希望计算似然函数, 那么我们可以从链的起点到终点运行 α 的递归过程, 然后使用 $n = N$ 的结果, 利用 $\beta(z_N)$ 是一个元素全部为1的向量这一事实。在这种情况下, β 递归不再需要, 从而我们有

$$p(\mathbf{X}) = \sum_{z_N} \alpha(z_N) \quad (13.42)$$

让我们花一些时间考察 $p(\mathbf{X})$ 的这个结果的意义。回忆一下, 为了计算似然函数, 我们应该在 Z 的所有可能值上对联合概率分布 $p(\mathbf{X}, Z)$ 求和。每个这样的值表示每个时间步骤下对隐含状态的一个特定的选择, 换句话说, 求和式中的每一项都是晶格图中的一个路径, 并且回忆一下, 这种路径有指数多条。通过将似然函数表示为 (13.42) 的形式, 我们将计算代价从关于链长度的指数量级减小到了线性量级, 方法是交换了加和与乘积的顺序, 从而在每个时间步骤 n 中, 我们对通过每个状态 z_{nk} 的所有路径的贡献进行求和, 得到了中间的量 $\alpha(z_n)$ 。

接下来我们考虑 $\xi(z_{n-1}, z_n)$ 的计算, 它对应于 (z_{n-1}, z_n) 的 $K \times K$ 个配置下的每一个配置的条件概率 $p(z_{n-1}, z_n | \mathbf{X})$ 的值。使用 $\xi(z_{n-1}, z_n)$ 的定义, 应用贝叶斯定理, 我们有

$$\begin{aligned} \xi(z_{n-1}, z_n) &= p(z_{n-1}, z_n | \mathbf{X}) \\ &= \frac{p(\mathbf{X} | z_{n-1}, z_n)p(z_{n-1}, z_n)}{p(\mathbf{X})} \\ &= \frac{p(x_1, \dots, x_{n-1} | z_{n-1})p(x_n | z_n)p(x_{n+1}, \dots, x_N | z_n)p(z_n | z_{n-1})p(z_{n-1})}{p(\mathbf{X})} \\ &= \frac{\alpha(z_{n-1})p(x_n | z_n)p(z_n | z_{n-1})\beta(z_n)}{p(\mathbf{X})} \end{aligned} \quad (13.43)$$

其中我们已经使用了条件独立性质 (13.29) 以及公式 (13.34) 和 (13.35) 给出的 $\alpha(z_n)$ 和 $\beta(z_n)$ 的定义。因此我们可以使用 α 和 β 的递归结果直接计算 $\xi(z_{n-1}, z_n)$ 。

让我们总结一下使用EM算法训练隐马尔科夫模型所需的步骤。首先, 我们需要进行对参数 θ 进行一个初始的选择, 其中 $\theta \equiv (\pi, \mathbf{A}, \phi)$ 。参数 \mathbf{A} 和 π 要么均匀地初始化, 要么从一个均匀分布中随机初始化 (满足非负限制与加和限制)。参数 ϕ 的初始化依赖于概率分布的形式。对于高斯分布的情形, 参数 μ_k 可以通过对数据使用 K 均值算法进行初始化, Σ_k 可以使用对应的 K 均值聚类的协方差矩阵初始化。然后我们运行前向 α 递归过程和后向 β 递归过程, 使用这些结果计算 $\gamma(z_n)$ 和 $\xi(z_{n-1}, z_n)$ 。在这个阶段, 我们也可以计算似然函数。这完成了E步骤, 然后我们使用这个结果, 使用13.2.1节的M步骤方程找到一个修正参数 $\theta^{\text{新}}$ 。然后我们继续交替进行E步骤和M步骤, 直到满足某些收敛准则, 例如似然函数的变化低于某个阈值。

注意, 在这些递归关系中, 观测只出现在条件概率分布 $p(x_n | z_n)$ 中。因此, 递归过程与观测变量的种类和维度无关, 也于这个条件概率的形式无关, 只要对于 z_n 的 K 种可能状态的每一个, 这个概率的值可以计算即可。

在之前的章节中, 我们已经看到, 当数据点的数量相对于参数的数量来说较大的时候, 最大似然方法最有效。这里, 我们注意到, 使用最大似然方法, 隐马尔可夫模型可以高效地训练, 只要训练的序列足够长。我们还可以使用多个较短的序列, 这需要对隐马尔可夫模型EM算法进行一些简单的修改。在从左到右模型的情况下, 这特别重要, 因为在一个给定的观测序列中, 对应于 A 的非对角元素的给定的状态转移最多出现一次。

我们感兴趣的另一个量是预测分布, 其中观测数据是 $\mathbf{X} = \{x_1, \dots, x_N\}$, 我们希望预测 x_{N+1} , 这对于诸如金融预测这种实时的应用来说很重要。与之前一样, 我们使用加和规则和

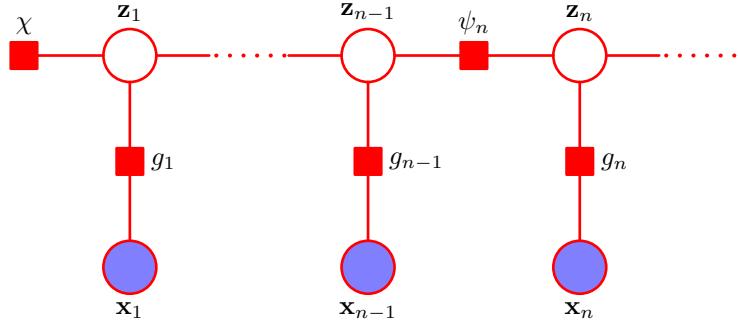


图 13.14: 隐马尔可夫模型的因子图表示的一个片段。

乘积桂策以及条件独立性质 (13.30) 和 (13.31)，可得

$$\begin{aligned}
 p(\mathbf{x}_{N+1} | \mathbf{X}) &= \sum_{\mathbf{z}_{N+1}} p(\mathbf{x}_{N+1}, \mathbf{z}_{N+1} | \mathbf{X}) \\
 &= \sum_{\mathbf{z}_{N+1}} p(\mathbf{x}_{N+1} | \mathbf{z}_{N+1}) p(\mathbf{z}_{N+1} | \mathbf{X}) \\
 &= \sum_{\mathbf{z}_{N+1}} p(\mathbf{x}_{N+1} | \mathbf{z}_{N+1}) \sum_{\mathbf{z}_N} p(\mathbf{z}_{N+1}, \mathbf{z}_N | \mathbf{X}) \\
 &= \sum_{\mathbf{z}_{N+1}} p(\mathbf{x}_{N+1} | \mathbf{z}_{N+1}) \sum_{\mathbf{z}_N} p(\mathbf{z}_{N+1} | \mathbf{z}_N) p(\mathbf{z}_N | \mathbf{X}) \\
 &= \sum_{\mathbf{z}_{N+1}} p(\mathbf{x}_{N+1} | \mathbf{z}_{N+1}) \sum_{\mathbf{z}_N} p(\mathbf{z}_{N+1} | \mathbf{z}_N) \frac{p(\mathbf{z}_N, \mathbf{X})}{p(\mathbf{X})} \\
 &= \frac{1}{p(\mathbf{X})} \sum_{\mathbf{z}_{N+1}} p(\mathbf{x}_{N+1} | \mathbf{z}_{N+1}) \sum_{\mathbf{z}_N} p(\mathbf{z}_{N+1} | \mathbf{z}_N) \alpha(\mathbf{z}_N)
 \end{aligned} \tag{13.44}$$

这可以通过首先运行前向 α 递归然后计算最后一个式子中关于 \mathbf{z}_N 和 \mathbf{z}_{N+1} 的求和式的方式得到。第一项关于 \mathbf{z}_N 的求和式可以被存储起来，一旦 \mathbf{x}_{N+1} 被观测到，就可以用来运行 α 递归的前向步骤，进行到下一步，来预测接下来的值 \mathbf{x}_{N+2} 。注意，在 (13.44) 中，从 \mathbf{x}_1 和 \mathbf{x}_N 的所有数据的影响被聚集到了 $\alpha(\mathbf{z}_N)$ 的 K 个值当中。因此预测分布可以使用固定量的存储空间向前推进无穷多次，这正是实时应用所要求的。

这里，我们已经讨论了使用最大似然方法估计HMM的参数。这个框架很容易推广到正则化的最大似然函数，方法是引入模型参数 π, A 和 ϕ 上的先验概率分布，然后通过最大化后验概率的方式估计参数的值。这个也可以使用EM算法计算，其中E步骤与上面的讨论相同，M步骤在最大化之前给似然函数 $Q(\theta, \theta^{\text{旧}})$ 加上先验概率分布 $p(\theta)$ 的对数，可以直接应用本书讨论的多种方法进行求解。此外，我们可以使用变分方法，得到HMM的一个纯粹的贝叶斯方法，其中我们对参数概率分布进行积分或求和 (MacKay, 1997)。与最大似然方法相同，这产生了一个两遍的前向后向递归的过程来计算后验概率分布。

13.2.3 用于HMM的加和-乘积算法

图13.5给出的表示隐马尔可夫模型的有向图是一棵树，因此我们可以使用加和-乘积算法来求解寻找局部边缘概率的问题。毫不令人惊讶的事实是，这等价于前一节讨论的前向-后向算法，因此加和-乘积算法给我们提供了一种简单的方式推导alpha-beta递归公式。

首先，我们将图13.5所示的有向图变换为因子图，图13.14给出了一个代表性的片段。这种形式的因子图显式地画出了潜在结点和观测结点。然而，对于解决推断问题来说，我们总是以变量 $\mathbf{x}_1, \dots, \mathbf{x}_N$ 为条件，因此我们可以通过将发射概率整合到转移概率因子中的方式来简化因子图。这就产生了图13.15给出的简化的因子图表示，其中因子为

$$h(\mathbf{z}_1) = p(\mathbf{z}_1)p(\mathbf{x}_1 | \mathbf{z}_1) \tag{13.45}$$

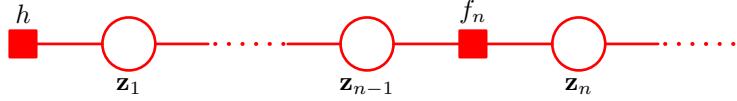


图 13.15: 一个简化形式的因子图用来描述隐马尔可夫模型。

$$f_n(z_{n-1}, z_n) = p(z_n | z_{n-1})p(x_n | z_n) \quad (13.46)$$

为了推导alpha-beta算法，我们将最后的隐含变量 z_N 看成根结点，首先从叶结点 h 向根结点传递信息。根据公式 (8.66) 和 (8.69) 给出的信息传播的一般结果，我们看到在隐马尔可夫模型中传递的信息的形式为

$$\mu_{z_{n-1} \rightarrow f_n}(z_{n-1}) = \mu_{f_{n-1} \rightarrow z_{n-1}}(z_{n-1}) \quad (13.47)$$

$$\mu_{f_n \rightarrow z_n}(z_n) = \sum_{z_{n-1}} f_n(z_{n-1}, z_n) \mu_{z_{n-1} \rightarrow f_n}(z_{n-1}) \quad (13.48)$$

这些方程表示沿着链的信息前向传递，等价于前一节推导出的alpha递归，说明如下。注意，由于变量结点 z_n 只有两个相邻结点，因此它们不进行计算。

我们可以使用公式 (13.47) 从公式 (13.48) 中消去 $\mu_{z_{n-1} \rightarrow f_n}(z_{n-1})$ ，得到 $f \rightarrow z$ 的信息的递归方程，形式为

$$\mu_{f_n \rightarrow z_n}(z_n) = \sum_{z_{n-1}} f_n(z_{n-1}, z_n) \mu_{f_{n-1} \rightarrow z_{n-1}}(z_{n-1}) \quad (13.49)$$

如果我们回忆一下定义 (13.46)，并且如果我们定义

$$\alpha(z_n) = \mu_{f_n \rightarrow z_n}(z_n) \quad (13.50)$$

那么我们就得到了公式 (13.36) 给出的alpha递归方程。我们还需要验证 $\alpha(z_n)$ 本身等价于之前的定义。可以这样做：使用初始条件 (8.71)，然后注意到 $\alpha(z_1)$ 为 $h(z_1) = p(z_1)p(x_1 | z_1)$ ，这与公式 (13.37) 完全相同。由于初始的 α 是相同的，并且它们使用同样的方程进行迭代地计算，因此所有后续的 α 一定相同。

接下来我们研究从根结点传递回到叶结点的信息，形式为

$$\mu_{f_{n+1} \rightarrow z_n}(z_n) = \sum_{z_{n+1}} f_{n+1}(z_n, z_{n+1}) \mu_{f_{n+2} \rightarrow z_{n+1}}(z_{n+1}) \quad (13.51)$$

其中，与之前一样，我们消去了形如 $z \rightarrow f$ 的信息，因为变量结点不参与计算。使用定义 (13.46) 消去 $f_{n+1}(z_n, z_{n+1})$ ，然后定义

$$\beta(z_n) = \mu_{f_{n+1} \rightarrow z_n}(z_n) \quad (13.52)$$

我们就得到了公式 (13.38) 定义的beta递归方程。我们同样可以验证beta变量本身是等价的。我们注意到公式 (8.70) 表明根变量结点发送的初始结点为 $\mu_{z_N \rightarrow f_N}(z_N) = 1$ ，这与13.2.2节给出了对 $\beta(z_N)$ 的初始化完全相同。

加和-乘积算法也指定了如何计算边缘概率，一旦所有的信息都已经被计算出来。特别地，公式 (8.63) 给出的结果表明结点 z_n 处的局部边缘概率是输入信息的乘积。由于我们以变量 $\mathbf{X} = \{x_1, \dots, x_N\}$ 为条件，因此我们计算的是联合概率分布

$$p(z_n, \mathbf{X}) = \mu_{f_n \rightarrow z_n}(z_n) \mu_{f_{n+1} \rightarrow z_n}(z_n) = \alpha(z_n) \beta(z_n) \quad (13.53)$$

将两侧同时除以 $p(\mathbf{X})$ ，我们有

$$\gamma(z_n) = \frac{p(z_n, \mathbf{X})}{p(\mathbf{X})} = \frac{\alpha(z_n) \beta(z_n)}{p(\mathbf{X})} \quad (13.54)$$

这与公式 (13.33) 相符。公式 (13.43) 给出的结果可以类似地从公式 (8.72) 中推导出。

13.2.4 缩放因子

在我们能够在实际应用中使用前向后向算法之前，有一件事情必须强调。根据递归关系 (13.36)，我们注意到在每一步中，新的值 $\alpha(z_n)$ 为前一个值 $\alpha(z_{n-1})$ 乘以 $p(z_n | z_{n-1})$ 和 $p(x_n | z_n)$ 。由于这些概率通常远远小于1，因此随着我们沿着链向前推进， $\alpha(z_n)$ 很快就会指数地趋近于零。对于中等的链长度（例如100左右）， $\alpha(z_n)$ 的计算很快就会超出计算机的计算范围，即使使用双精度浮点数也是如此。

在独立同分布数据的情形，我们使用取对数的方式，隐式地避开了计算似然函数的这个问题。不幸的是，这种方法在这里没有作用，因为我们对很小的数字的乘积进行求和（事实上我们隐式地对图13.7的晶格图中的所有可能的路径求和）。因此我们使用重新缩放的 $\alpha(z_n)$ 和 $\beta(z_n)$ 来计算，它们的值保持与单位长度在同一个量级上。正如我们将看到的那样，当我们在EM算法中使用这些缩放的量时，对应的缩放因子会消去。

在公式 (13.34) 中，我们定义了 $\alpha(z_n) = p(x_1, \dots, x_n, z_n)$ ，表示所有截止到 x_n 的观测以及潜在变量 z_n 的联合概率分布。现在我们定义 α 的一个归一化的版本，形式为

$$\hat{\alpha}(z_n) = p(z_n | x_1, \dots, x_n) = \frac{\alpha(z_n)}{p(x_1, \dots, x_n)} \quad (13.55)$$

我们预计这个量在数值计算上可以表现良好，因为对任意 n 值，它都是 K 个变量上的一个概率分布。为了将缩放的alpha变量与原始的alpha变量关联起来，我们引入缩放因子，它由观测变量上的条件概率分布定义，即

$$c_n = p(x_n | x_1, \dots, x_{n-1}) \quad (13.56)$$

根据乘积规则，我们有

$$p(x_1, \dots, x_n) = \prod_{m=1}^n c_m \quad (13.57)$$

因此

$$\alpha(z_n) = p(z_n | x_1, \dots, x_n) p(x_1, \dots, x_n) = \left(\prod_{m=1}^n c_m \right) \hat{\alpha}(z_n) \quad (13.58)$$

然后我们可以将 α 的递归方程 (13.36) 转化为 $\hat{\alpha}$ 的递归方程，形式为

$$c_n \hat{\alpha}(z_n) = p(x_n | z_n) \sum_{z_{n-1}} \hat{\alpha}(z_{n-1}) p(z_n | z_{n-1}) \quad (13.59)$$

注意，在用于计算 $\hat{\alpha}(z_n)$ 的前向信息传播阶段的每一步，我们必须计算和存储 c_n ，这很容易做到，因为它是将公式 (13.59) 的右侧归一化得到 $\hat{\alpha}(z_n)$ 的归一化系数。

类似地，我们可以使用下式

$$\beta(z_n) = \left(\prod_{m=n+1}^N c_m \right) \hat{\beta}(z_n) \quad (13.60)$$

定义重新缩放的变量 $\hat{\beta}(z_n)$ 。它的值再次保持在机器的精度范围内，因为根据公式 (13.35)， $\hat{\beta}(z_n)$ 仅仅是两个条件概率分布的比值

$$\hat{\beta}(z_n) = \frac{p(x_{n+1}, \dots, x_N | z_n)}{p(x_{n+1}, \dots, x_N | x_1, \dots, x_n)} \quad (13.61)$$

这样，根据 β 的递归结果 (13.38) 可以得到下面的对重新标准的变量的递归方程

$$c_{n+1} \hat{\beta}(z_n) = \sum_{z_{n+1}} \hat{\beta}(z_{n+1}) p(x_{n+1} | z_{n+1}) p(z_{n+1} | z_n) \quad (13.62)$$

在应用这个递归关系时，我们使用之前在 α 阶段计算的缩放因子 c_n 。

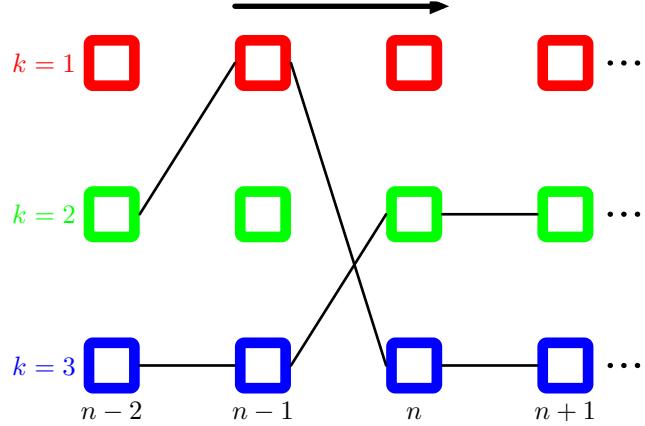


图 13.16: HMM晶格图片段，画出了两条可能的路径。维特比算法从指数多种可能的路径中高效地确定概率最高的路径。对于任意给定的路径，对应的概率为转移矩阵的元素 A_{jk} （对应于每个路径片段的概率 $p(z_{n+1} | z_n)$ ）和与路径上的每个结点相关联的发射概率密度 $p(x_n | k)$ 的乘积。

根据公式 (13.57) , 我们看到似然函数可以使用下式求出

$$p(\mathbf{X}) = \prod_{n=1}^N c_n \quad (13.63)$$

类似地，使用 (13.33) 和 (13.43) 以及 (13.63) , 我们看到所求的边缘概率为

$$\gamma(z_n) = \hat{\alpha}(z_n)\hat{\beta}(z_n) \quad (13.64)$$

$$\xi(z_{n-1}, z_n) = c_n^{-1}\hat{\alpha}(z_{n-1})p(x_n | z_n)p(z_n | z_{n-1})\hat{\beta}(z_n) \quad (13.65)$$

最后，我们注意到前向后向算法有另一种公式 (Jordan, 2007) , 其中后向传递由基于 $\gamma(z_n) = \hat{\alpha}(z_n)\hat{\beta}(z_n)$ 的递归定义，而不是使用 $\hat{\beta}(z_n)$ 。这个 $\alpha - \gamma$ 递归要求前向传递过程首先完成，从而在后向传递过程中能得到所有的 $\hat{\alpha}(z_n)$ ，而 $\alpha - \beta$ 算法的前向和波uxiangguocheng可以独立地进行。虽然这两个算法的计算代价是可比的，但是在隐马尔可夫模型的情形下， $\alpha - \beta$ 版本是最经常遇到的，而对于线性动态系统，与 $\alpha - \gamma$ 形式类似的递归规程更常见。

13.2.5 维特比算法

在隐马尔可夫模型的许多应用中，潜在变量有许多有意义的直观意义，因此对于给定的观测序列，我们常常感兴趣的是寻找概率最高的隐含状态序列。例如，在语音识别中，对于一个给定的声音观测序列，我们可能希望找到概率最大的音素序列。由于隐马尔可夫模型的图是一棵有向树，因此这个问题可以使用最大加和算法精确地求解。回忆一下，根据8.4.5节，寻找潜在变量的概率最高的序列与寻找分别概率最高的状态的集合是不相同的。后一个问题可以这样解决：首先运行前向后向算法（加和-乘积算法）找到潜在变量边缘概率 $\gamma(z_n)$ ，然后单独最大化每个概率 (Duda et al., 2001) 。然而，通常这样的状态集合不会对应于最可能的状态序列。事实上，如果对于两个连续的状态，它们单独的概率都是最高的，但是连接它们的转移矩阵的元素为零，那么这个状态集合表示一个具有零概率的序列。

在实际应用中，我们通常感兴趣的是寻找最可能的状态序列 (sequence) , 这可以使用最大加和算法高效地求出，这个算法在隐马尔科夫模型中被称为维特比算法 (Viterbi algorithm)

(Viterbi, 1967)。注意，最大加和算法作用于对数概率，因此无需使用前向后向算法中的重新缩放的变量。图13.16给出了隐马尔科夫模型的晶格图的一个片段。正如我们已经注意到的，通过经过的可能的路径的数量随着链的长度指数增长。维特比算法高效地搜索这个路径空间，找到概率最高的路径，计算代价仅仅随着链的长度线性增长。

与加和-乘积算法相同，我们首先将隐马尔可夫模型表示为因子图，如图13.15所示。与之前一样，我们将变量结点 z_N 当成根结点，从根结点开始向叶结点传递信息。使用公式 (8.93) 和

(8.94) 的结果，我们看到在最大加和算法中传递的信息为

$$\mu_{z_n \rightarrow f_{n+1}}(z_n) = \mu_{f_n \rightarrow z_n}(z_n) \quad (13.66)$$

$$\mu_{f_{n+1} \rightarrow z_{n+1}}(z_{n+1}) = \max_{z_n} \{\ln f_{n+1}(z_n, z_{n+1}) + \mu_{z_n \rightarrow f_{n+1}}(z_n)\} \quad (13.67)$$

如果消去两个方程间的 $\mu_{z_n \rightarrow f_{n+1}}(z_n)$ ，然后使用公式 (13.46)，我们得到了 $f \rightarrow z$ 的信息的递归方程，形式为

$$\omega(z_{n+1}) = \ln p(x_{n+1} | z_{n+1}) + \max_{z_n} \{\ln p(z_{n+1} | z_n) + \omega(z_n)\} \quad (13.68)$$

其中我们引入了记号 $\omega(z_n) \equiv \mu_{f_n \rightarrow z_n}(z_n)$ 。

根据公式 (8.95) 和 (8.96)，这些信息使用下面的公式初始化

$$\omega(z_1) = \ln p(z_1) + \ln p(x_1 | z_1) \quad (13.69)$$

其中我们已经使用了公式 (13.45)。注意，为了保持记号简洁，我们略去了对模型参数 θ 的依赖关系，它在我们寻找概率最高的序列的过程中保持固定。

维特比算法也可以直接从联合概率分布的定义 (13.6) 中直接推导，方法是取对数，然后交换求最大值和求和的顺序。很容易看到 $\omega(z_n)$ 具有下面的概率意义

$$\omega(z_n) = \max_{z_1, \dots, z_{n-1}} \ln p(x_1, \dots, x_n, z_1, \dots, z_n) \quad (13.70)$$

一旦我们完成了在 z_N 上的最大化过程，那么我们就得到了对应于概率最大的路径的联合概率分布 $p(X, Z)$ 。我们还希望找到对应于这条路径的潜在变量值的序列。为了完成这一点，我们简单地使用 8.4.5 节讨论的反向跟踪方法。具体来说，我们注意到在 z_n 上的最大化过程必须在 z_{n+1} 的 K 个可能值的每一个值上进行。假设对于 z_{n+1} 的 K 个值中的每一个值，我们都记录下与最大值相对应的 z_n 的值。让我们将这个函数记作 $\psi(k_n)$ ，其中 $k \in \{1, \dots, K\}$ 。一旦我们将信息传递到了链的末端，找到了概率最大的状态 z_N ，那么我们可以使用这个函数来沿着链进行反向跟踪，方法是递归地应用下式

$$k_{n-1}^{\text{最大}} = \psi(k_n^{\text{最大}}) \quad (13.71)$$

直观上讲，我们可以按照下面的方式理解维特比算法。朴素地说，我们可以显式地考虑通过晶格的指数多条路径，计算每条路径的概率，然后选择具有最高概率的路径。然而，我们注意到，我们可以对计算量进行极大的简化。假设对于每条路径，我们在沿着通过晶格的每条路径前向计算时，通过将转移概率与发射概率的乘积进行求和的方式求出这个概率。考虑一个特定的时刻 n 以及在那个时刻的一个特定的状态 k 。会存在许多条路径收敛到晶格图中的对应的结点。然而，我们只需要保留当前具有最高概率的特定的路径即可。由于在时刻 n 有 K 个状态，因此我们需要跟踪 K 个这样的路径。在时刻 $n+1$ ，会存在 K 个可能的路径要考虑，由在 K 个当前状态中的每个状态引出的 K 个可能的路径组成，但是在 $n+1$ 时刻，我们还是只需保留对应于最优路径的 K 个状态。当我们到达最后的时刻 N 时，我们会发现哪个状态对应于整体上概率最高的路径。由于存在一个唯一的一条路径进入这个状态，因此我们可以反向跟踪这条路径到 $N-1$ 步，看到那一时刻出现了哪个状态，以此类推，沿着晶格跟踪到状态 $n=1$ 。

13.2.6 隐马尔科夫模型的扩展

基本的隐马尔科夫模型以及基于最大似然方法的标准训练算法已经通过很多种方式进行了扩展，来满足特定应用的需求。这里，我们讨论几个更重要的例子。

我们从图 13.11 的手写数字的例子中可以看到，隐马尔可夫模型对于数据来说，是一个相当差的生成式模型，因为许多人工生成的数字对于训练集来说看起来相当不具有代表性。如果目标是序列分类，那么在确定隐马尔科夫模型的参数时，使用判别式方法而不是最大似然方法会产生很多好处。假设我们有一个训练集，由 R 个观测序列 X_r 组成，其中 $r = 1, \dots, R$ ，每个序列根

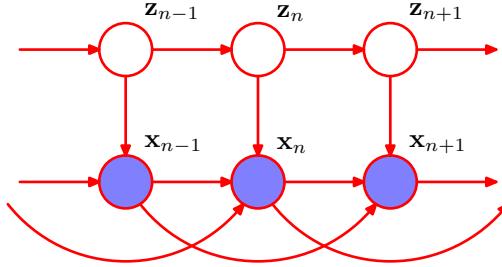


图 13.17: 自回归隐马尔可夫模型的一部分，其中，观测 x_n 的概率分布依赖于之前的观测的子集以及隐状态 z_n 。在这个例子中， x_n 的分布依赖于两个之前的观测 x_{n-1} 和 x_{n-2} 。

据它的类别 m 进行标记，其中 $m = 1, \dots, M$ 。对于每个类别，我们有一个独立的隐马尔可夫模型，它的参数为 θ_m ，我们将确定参数值的问题看成标准的分类问题，其中我们想最优化交叉熵

$$\sum_{r=1}^R \ln p(m_r | \mathbf{X}_r) \quad (13.72)$$

使用贝叶斯定理，这个可以使用与隐马尔可夫模型相关联的序列概率表示，即

$$\sum_{r=1}^R \ln \left\{ \frac{p(\mathbf{X}_r | \theta_r) p(m_r)}{\sum_{l=1}^M p(\mathbf{X}_r | \theta_l) p(l_r)} \right\} \quad (13.73)$$

其中 $p(m)$ 是类别 m 的先验概率。对这个代价函数的最优化比最大化似然函数更复杂 (Kapadia, 1998)，特别地，为了计算公式 (13.73) 的坟墓，这种方法需要每个训练序列在每个模型下进行计算。隐马尔科夫模型加上判别式的训练方法在语音识别中广泛应用 (Kapadia, 1998)。

隐马尔科夫模型的一个很大的缺点是，系统保持在一个给定的状态下，模型对于时间分布的表示方法。为了说明这个问题，我们注意到，从一个给定的隐马尔科夫模型中采样到一个序列，这个序列在状态 k 恰好花费了 T 个步骤，然后转移到了一个不同的状态，这种情形出现的概率为

$$p(T) = (A_{kk})^T (1 - A_{kk}) \propto \exp(T \ln A_{kk}) \quad (13.74)$$

因此它是 T 的一个指数衰减的函数。对于许多应用，这对于状态持续时间来说是一个相当不现实的模型。问题可以这样解决：直接对状态持续时间建模，其中对角系数 A_{kk} 全部被设置为零，每个状态 k 显式地与可能的持续时间的概率分布 $p(T | k)$ 相关联。从生成式的观点来看，当系统进入状态 k 时，表示系统保持在状态 k 的时间数 T 会从 $p(T | k)$ 中抽取。模型之后发射出观测变量 x_t 的 T 个值，这通常被假定为独立的，从而对应的发射概率分布为 $\prod_{t=1}^T p(x_t | k)$ 。这种方法需要对EM最优化步骤进行简单的修改 (Rabiner, 1989)。

标准HMM的另一个局限性是它在描述观测变量的长距离相关性（被许多时间步骤分开的变量的相关性）时，效果很差，因为这些相关性必须被隐含状态的一阶马尔科夫链所调解。长距离的效果原则上可以通过在图13.5所示的图模型中添加额外链接的方式被包含到模型中。一种解决的办法是将HMM进行推广，得到了自回归隐马尔科夫模型 (autoregressive hidden Markov model) (Ephraim et al., 1989)。图13.17给出了这个模型的一个例子。对于离散的观测来说，这对应于将发射概率分布的条件概率表进行扩展。在高斯发射概率密度的情形下，我们可以使用线性高斯的框架，其中，给定前一个观测的值以及 z_n 的值的条件下， x_n 的条件概率分布是一个高斯分布，均值为条件变量值的一个线性组合。很明显，图中附加的链接必须被限制，避免自由参数的数量过多。在图13.17给出的例子中，每个观测依赖于前两个观测变量以及隐含状态。虽然这个图看起来很短，但是我们再次采用d-划分，可以看到，事实上，它有一个简单的概率结构。特别地，如果我们假设以 z_n 为条件，那么我们看到，与标准的HMM相同， z_{n-1} 和 z_{n+1} 的值是独立的，对应于条件独立性质 (13.5)。这很容易验证。我们注意到，每个从结点 z_{n-1} 到结点 z_{n+1} 的路径都要穿过至少一个关于那条路径头到尾连接的观测结点。从而，在EM算法的E步骤中，我们可以再次使用前向后向递归，确定潜在变量的后验概率分布，计算时间与链的长度是线性关系。类似地，M步骤值涉及到对标准的M步骤方程的一个微小的修改。在高斯发射密度的情形下，这涉及到使用第3章讨论的标准线性回归方程估计参数。

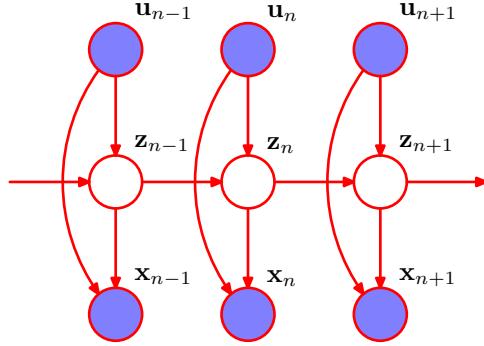


图 13.18: 输入输出隐马尔可夫模型的一个例子。在这种情况下, 发射概率和转移概率都依赖于观测序列 u_1, \dots, u_N 的值。

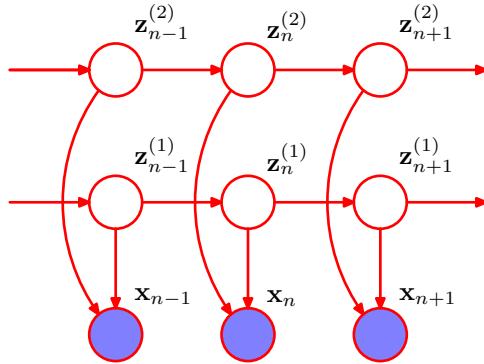


图 13.19: 由两个潜在变量马尔科夫链组成的因子隐马尔可夫模型。对于连续的观测变量 x , 发射模型的一种可能的选择是线性高斯概率密度, 其中高斯分布的均值是对应的潜在变量状态的线性组合。

我们已经看到, 当我们使用图模型时, 自回归HMM可以看成标准HMM的一个很自然的扩展。事实上, 概率图模型的观点会产生基于HMM的相当多种不同的图结构。另一个例子是输入输出隐马尔科夫模型 (input-output hidden Markov model) (Bengio and Frasconi, 1995), 其中我们有一个观测变量的序列 u_1, \dots, u_N , 以及输出变量的序列 x_1, \dots, x_N , 观测变量的值要么影响潜在变量的分布, 要么影响输出变量的分布, 或者对两者都产生影响。图13.18给出了一个例子。这将HMM的框架推广到了顺序数据的有监督学习领域。与之前一样, 通过使用d-划分, 很容易证明潜在变量链的马尔科夫性质 (13.5) 仍然成立。为了证明这一点, 我们注意到从结点 z_{n-1} 到结点 z_{n+1} 只有一条路径, 这条路径关于观测结点 z_n 是头到尾的。这个条件独立性质又一次使得高效的学习算法的公式能够成立。特别地, 我们可以通过最大化似然函数 $L(\theta) = p(\mathbf{X} | \mathbf{U}, \theta)$ 的方式确定模型参数 θ , 其中 \mathbf{U} 是一个矩阵, 它的行等于 u_n^T 。由于条件独立性质 (13.5), 可以使用EM算法对似然函数进行最大化, 其中, E步骤涉及到前向和后向的递归。

HMM的另一个值得一提的变体是因子隐马尔可夫模型 (factorial hidden Markov model) (Ghahramani and Jordan, 1997), 其中存在多个独立的潜在变量马尔科夫链, 并且在一个给定的时刻, 观测变量的概率分布以相同时间的所有对应的潜在变量的状态为条件。图13.19展示了对应的图模型。为了说明研究因子HMM的动机, 我们注意到, 在一个给定的时刻, 为了表示例如10比特的信息, 标准的HMM需要 $K = 2^{10} = 1024$ 个潜在状态, 而因子HMM可以使用10个二值潜在链。然而, 因子HMM的主要缺点是训练时的额外的复杂度。因子HMM的M步骤很容易。然而, x 变量的观测引入了潜在链之间的依赖关系, 从而给E步骤带来了困难, 说明如下。我们注意到在图13.19中, 变量 $z_n^{(1)}$ 和 $z_n^{(2)}$ 由一个在结点 x_n 处的头到头的路径链接, 因此不是d-划分的。这个模型的精确的E步骤无法对应于在 M 个马尔科夫链上独立地运行前向和后向递归。我们注意到关键的条件独立性质 (13.5) 对于因子HMM模型中的各个马尔科夫链不成立, 图13.20给出了使用d-划分的结果, 从而证实了确实无法独立地运行前向和后向递归。现在假设有 M 个隐含结点链, 并且为了简化起见, 我们假设所有的潜在变量的状态数量都为 K 。这样, 在一个给定的时刻, 一种方法会关注潜在变量的 K^M 种组合, 因此我们可以将模型转化为一个

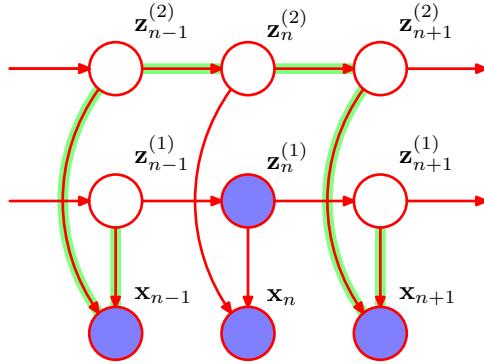


图 13.20: 绿色标记的路径在观测结点 x_{n-1} 和 x_{n+1} 处是头到头的，在非观测结点 $z_{n-1}^{(2)}$, $z_n^{(2)}$ 和 $z_{n+1}^{(2)}$ 处是头到尾的。因此路径没有被阻隔，从而条件独立性质 (13.5) 对于因子HMM模型的各个潜在链不成立。结果，这个模型没有高效的精确E步骤。

等价的标准HMM，它由一个单独的潜在变量链，每个潜在变量有 K^M 个潜在状态。然后我们可以在E步骤中运行标准的前向后向递归方法。计算复杂度为 $O(NK^{2M})$ ，它与潜在链的数量 M 是指数的关系，因此除了对于很小的 M 值以外均无法计算。一个解决方法是使用采样方法（第11章讨论）。作为另一个优雅的确定性的解决方法，Ghahramani and Jordan (1997) 研究了使用变分推断方法来得到近似推断的一个可以计算的算法。可以这样做：使用一个简单的变分后验概率分布，它关于潜在变量可以完全分解，或者使用一个更强大的方法，其中变分分布由独立的马尔科夫链描述，马尔科夫链对应于原始模型中的潜在变量链。在后一种情形中，变分推断算法涉及到沿着每条链独立地运行前向和后向递归，这在计算上很有效率，同时也能够描述同一个链上的变量之间的相关性。

很明显，根据特定的应用需要，可以构建许多可能的概率模型。图模型提供了一个一般的方法来提出、描述、分析这些结果，变分方法提供了一个强大的框架，对无法得到精确解的模型进行推断。

13.3 线性动态系统

为了说明线性动态系统的概念，让我们考虑下面这个简单的例子，它经常在实际问题中出现。假设我们希望使用一个有噪声的传感器测量一个未知量 z 的值，传感器返回一个观测值 x ，表示 z 的值加上一个零均值的高斯噪声。给定一个单次的测量，我们关于 z 的最好的猜测是假设 $z = x$ 。然而，我们可以通过取多次测量然后求平均的方法提高我们对 z 的估计效果，因为随机噪声项倾向于彼此抵消。现在，让我们将情况变得更复杂。假设我们希望测量一个随着时间变化的量 z 。我们可以对进行常规的测量 x ，从而我们得到了 x_1, \dots, x_N ，我们希望找到对应的 z_1, \dots, z_N 。如果我们简单地对测量求平均，那么由于随机噪声产生的误差会被消去，但是不幸的是我们会仅仅得到一个单一的平均估计，对 z 的变化进行了平均，从而引入了一种新的误差。

直观上讲，我们可以用下面的方式稍微好一些地完成这个任务。为了估计 z_N 的值，我们只取最近的几次测量，例如 x_{N-L}, \dots, x_N ，然后求平均。如果 z 的变化很慢，并且传感器的随机噪声的水平很高，那么选择一个相对长的窗口求平均是有意义的。相反，如果信号变化很快，并且噪声水平相对较小，那么我们直接使用 x_N 来估计 z_N 会更合适。如果我们求加权平均，即最近的测量比之前的测量的贡献更大，那么或许效果会更好。

虽然这种主观的讨论似乎是可行的，但是它并没有告诉我们如何求加权平均，并且任何一种人工设计的权值都很难成为最优的。幸运的是，我们可以更加系统化地解决这种问题，方法是定义一个概率模型，它描述了时间的演化和测量过程，然后应用了之前章节中讨论的推断和学习方法。这里，我们关注一类广泛使用的模型，被称为线性动态系统 (linear dynamical system)。

正如我们已经看到的，HMM对应于图13.5给出的状态空间模型，其中潜在变量是离散的，但是发射概率分布是任意的。这个图显然描述了相当大的一类概率分布，所有的都可以根据公式 (13.6) 进行分解。我们现在考虑对潜在变量的其他类型的概率分布的推广。特别地，我们考

虑连续潜在变量，其中加和-乘积算法的求和变成了积分。然而，推断算法的一般形式与隐马尔可夫模型相同。值得注意的很有趣的一点是，历史上，隐马尔可夫模型和线性动态系统是独立研究的。然而，一旦它们都用图模型进行表示，它们之间的深层关系就立刻变得明显了。

一个重要的要求是，我们保留了推断的高效算法，它与链的长度是线性关系。例如，这要求，在给定观测 x_1, \dots, x_{n-1} 的条件下，表示 z_{n-1} 的后验概率分布的量 $\hat{\alpha}(z_{n-1})$ 在与转移概率 $p(z_n | z_{n-1})$ 和发射概率 $p(x_n | z_n)$ 相乘然后在 z_{n-1} 上求和或积分之后，我们得到的 z_n 上的概率分布与 $\hat{\alpha}(z_{n-1})$ 上的概率分布具有相同的函数形式。这就是说，在每个阶段，概率分布不可以变得更复杂，而是仅仅在参数值上发生改变。毫不令人惊讶地说，在多次相乘之后具有这个性质的唯一的分布就是指数族分布的成员。

这里，我们从实际应用的角度考虑一个最重要的例子，即高斯分布。特别地，我们考虑一个线性高斯状态空间模型，从而潜在变量 $\{z_n\}$ 以及观测变量 $\{x_n\}$ 是多元高斯分布，均值是图表示中的状态的线性函数。我们已经看到，线性高斯单元的有向图等价于所有变量上的联合高斯分布。此外，诸如 $\hat{\alpha}(z_n)$ 的边缘概率分布也是高斯分布，从而信息的函数形式被保留了下来，我们可以得到一个高效的推断算法。相反，假设发射概率密度 $p(x_n | z_n)$ 由 K 个高斯分布混合而成，每个高斯分布的均值都是 z_n 的线性函数，那么即使 $\hat{\alpha}(z_1)$ 是一个高斯分布， $\hat{\alpha}(z_2)$ 会是 K 个高斯分布的混合， $\hat{\alpha}(z_3)$ 会是 K^2 个高斯分布的混合，以此类推，因此精确的推断没有实际价值。

我们已经看到隐马尔科夫模型可以看成第9章的混合模型的一个推广，它允许数据之间具有顺序相关性。类似地，我们可以将线性动态系统看成第12章的连续潜在变量模型（例如概率PCA和因子分析）的推广，每对结点 $\{z_n, x_n\}$ 表示那个特定的观测下的一个线性高斯潜在变量模型。然而，潜在变量 $\{z_n\}$ 不再被看成独立的，而是构成了一个马尔科夫链。

由于模型由树结构的有向图表示，因此推断问题可以使用加和-乘积算法高效地求解。前向递归方程，类似于隐马尔可夫模型的 α 信息，被称为Kalman滤波（Kalman filter）方程（Kalman, 1960; Zarchan and Musoff, 2005），后向递归方程，类似于 β 信息，被称为Kalman平滑（Kalman smoother）方程，或者Rauch-Tung-Striebel (RTS)方程（Rauch et al., 1965）。Kalman滤波被广泛应用于许多实时跟踪应用中。

由于线性动态系统是一个线性高斯模型，因此在所有变量上的联合概率分布以及边缘分布和条件分布都是高斯分布。它遵循下面的事实：单独地概率最大的潜在变量值组成的序列与概率最大的潜在变量序列相同。因此对于线性动态系统，无需考虑与维特比算法类似的算法。

由于模型的条件概率分布是高斯分布，因此我们可以将转移分布和发射分布写成一般的形式

$$p(z_n | z_{n-1}) = \mathcal{N}(z_n | Az_{n-1}, \Gamma) \quad (13.75)$$

$$p(x_n | z_n) = \mathcal{N}(x_n | Cz_n, \Sigma) \quad (13.76)$$

初始潜在变量也服从高斯分布，我们写成

$$p(z_1) = \mathcal{N}(z_1 | \mu_0, P_0) \quad (13.77)$$

注意，为了简化记号，我们省略了高斯分布的均值中额外的可加性常数。事实上，如果必要的话，加上这些常数是很容易的。传统上，这些概率分布通常使用噪声线性方程表示为一个等价的形式，噪声线性方程为

$$z_n = Az_{n-1} + w_n \quad (13.78)$$

$$x_n = Cz_n + v_n \quad (13.79)$$

$$z_1 = \mu_0 + u \quad (13.80)$$

其中噪声项的概率分布为

$$w \sim \mathcal{N}(w | \mathbf{0}, \Gamma) \quad (13.81)$$

$$v \sim \mathcal{N}(v | \mathbf{0}, \Sigma) \quad (13.82)$$

$$u \sim \mathcal{N}(u | \mathbf{0}, P_0) \quad (13.83)$$

模型的参数被记作 $\theta = \{A, \Gamma, C, \Sigma, \mu_0, P_0\}$ ，可以通过EM算法使用最大似然的方法确定。在E步骤中，我们需要求解确定潜在变量的局部后验边缘概率的推断问题，这可以使用加和-乘积算法高效地求出，我们将在下一节讨论。

13.3.1 LDS中的推断

我们现在考虑寻找以观测序列为条件的潜在变量的边缘概率分布的问题。在实时应用中，对于给定的参数设置，我们也希望以观测数据 x_1, \dots, x_{n-1} 为条件，对于下一个潜在状态 z_n 以及下一个观测 x_n 做出预测。这些推断问题可以使用加和-乘积算法高效地解决，这个算法在线性动态系统的问题中会给出Kalman滤波方程和Kalman平滑方程。

值得强调的是，因为线性动态系统是线性高斯模型，因此所有潜在变量和观测变量上的联合概率分布是高斯分布，因此原则上我们可以使用之前章节推导的多元变量高斯分布的边缘概率和条件概率的标准结果来解决这个推断问题。加和-乘积算法的作用是为这些计算提供了一个更加高效的方式。

线性动态系统与隐马尔可夫模型具有相同的分解方式，由公式 (13.6) 给出，并且由图13.14 和图13.15的因子图描述。于是，推断问题的形式完全相同，唯一的差别在于潜在变量上的求和被替换为积分。首先，我们考虑前向方程，其中我们将 z_N 看做根结点，然后从叶结点 $h(z_1)$ 将信息传递到根结点。根据公式 (13.77)，初始信息服从高斯分布，并且由于每个因子都服从高斯分布，因此所有后续的信息也都服从高斯分布。按照传统，我们传递的信息是归一化的边缘概率分布，对应于 $p(z_n | x_1, \dots, x_n)$ ，我们将其记作

$$\hat{\alpha}(z_n) = \mathcal{N}(z_n | \mu_n, V_n) \quad (13.84)$$

这与公式 (13.59) 给出的隐马尔科夫模型的离散变量情形的缩放变量 $\hat{\alpha}(z_n)$ 的传播完全相同，因此递归方程的形式为

$$c_n \hat{\alpha}(z_n) = p(x_n | z_n) \int \hat{\alpha}(z_{n-1}) p(z_n | z_{n-1}) dz_{n-1} \quad (13.85)$$

使用公式 (13.75) 和 (13.76) 替换 $p(z_n | z_{n-1})$ 和 $p(x_n | z_n)$ ，然后使用公式 (13.84)，我们看到 (13.85) 变成了

$$\begin{aligned} c_n \mathcal{N}(z_n | \mu_n, V_n) &= \mathcal{N}(x_n | Cz_n, \Sigma) \\ &\quad \int \mathcal{N}(z_n | Az_{n-1}, \Gamma) \mathcal{N}(z_{n-1} | \mu_{n-1}, V_{n-1}) dz_{n-1} \end{aligned} \quad (13.86)$$

这里我们假设 μ_{n-1} 和 V_{n-1} 是已知的，并且通过计算公式 (13.86) 中的积分，我们希望确定 μ_n 和 V_n 的值。使用公式 (2.115) 给出的结果，这个积分很容易计算。我们有

$$\begin{aligned} &\int \mathcal{N}(z_n | Az_{n-1}, \Gamma) \mathcal{N}(z_{n-1} | \mu_{n-1}, V_{n-1}) dz_{n-1} \\ &= \mathcal{N}(z_n | A\mu_{n-1}, P_{n-1}) \end{aligned} \quad (13.87)$$

其中我们定义了

$$P_{n-1} = AV_{n-1}A^T + \Gamma \quad (13.88)$$

我们现在可以将这个结果与公式 (13.86) 右侧的第一个因子结合，使用公式 (2.115) 和 (2.116)，有

$$\mu_n = A\mu_{n-1} + K_n(x_n - CA\mu_{n-1}) \quad (13.89)$$

$$V_n = (I - K_n C) P_{n-1} \quad (13.90)$$

$$c_n = \mathcal{N}(x_n | CA\mu_{n-1}, CP_{n-1}C^T + \Sigma) \quad (13.91)$$

这里，我们使用了矩阵求逆的恒等式 (C.5) 和 (C.7)，并且定义了Kalman增益矩阵 (Kalman gain matrix)

$$K_n = P_{n-1}C^T(CP_{n-1}C^T + \Sigma)^{-1} \quad (13.92)$$

因此，给定 μ_{n-1} 和 V_{n-1} ，以及新的观测 x_n ，我们可以计算 z_n 的高斯边缘分布，均值为 μ_n ，协方差为 V_n ，归一化系数为 c_n 。

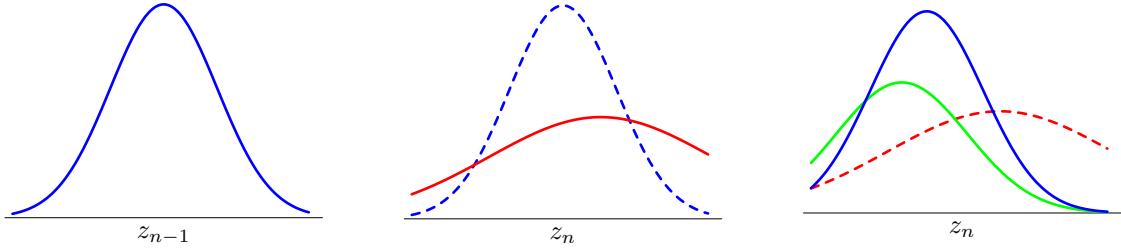


图 13.21: 线性动态系统可以被看成一个步骤序列, 其中由于传播造成的状态变量的逐渐增大的不确定性由新数据的到达所补偿。在左图中, 蓝色曲线表示概率分布 $p(z_{n-1} | x_1, \dots, x_{n-1})$, 它整合了截止到第 $n-1$ 步的所有数据。由方差非零的转移概率 $p(z_n | z_{n-1})$ 产生的传播过程给出了概率分布 $p(z_n | x_1, \dots, x_{n-1})$, 在中间图中表示为红色曲线。注意, 与蓝色曲线相比, 红色曲线更宽, 并且有所偏移。为了对比, 蓝色曲线用虚线画出。下一个数据观测 x_n 通过发射概率密度 $p(x_n | z_n)$ 产生贡献。右图中的绿色曲线表示发射概率与 z_n 的函数关系。注意, 它不是关于 z_n 的概率密度, 因此没有被归一化。使用这个新的数据点会产生状态概率密度的一个修正的概率分布 $p(z_n | x_1, \dots, x_n)$, 用蓝色表示。我们看到, 与 $p(z_n | x_1, \dots, x_{n-1})$ 相比, 数据的观测使得概率分布产生偏移, 并且变得更窄了。为了对比, $p(z_n | x_1, \dots, x_{n-1})$ 在右图中用红色虚线表示。

这些递归方程的初始条件为

$$c_1 \hat{\alpha}(z_1) = p(z_1)p(x_1 | z_1) \quad (13.93)$$

由于 $p(z_1)$ 由公式 (13.77) 给出, $p(x_1 | z_1)$ 由公式 (13.76) 给出, 因此我们可以再次使用 (2.115) 计算 c_1 , 使用 (2.116) 计算 μ_1 和 V_1 , 结果为

$$\mu_1 = \mu_0 + K_1(x_1 - C\mu_0) \quad (13.94)$$

$$V_1 = (I - K_1 C)P_0 \quad (13.95)$$

$$c_1 = \mathcal{N}(x_1 | C\mu_0, CP_0C^T + \Sigma) \quad (13.96)$$

其中

$$K_1 = P_0 C^T (CP_0 C^T + \Sigma)^{-1} \quad (13.97)$$

类似地, 线性动态系统的似然函数由公式 (13.63) 给出, 其中因子 c_n 使用 Kalman 滤波方程求解。

我们可以直观地给出从 z_{n-1} 上的后验边缘分布到 z_n 上的后验边缘分布的步骤, 如下所述。在公式 (13.89) 中, 我们可以将 $A\mu_{n-1}$ 看成 z_n 上的均值的预测, 得到这个预测的方法是在 z_{n-1} 上取均值, 然后使用一个前向的步骤, 使用转移概率矩阵 A 进行投影。预测均值会给出 x_n 的一个预测观测 $CA\mu_{n-1}$, 得到这个预测的方法是讲发射概率矩阵 C 作用在预测的隐含状态均值上。我们可以将隐含变量分布的均值的更新方程 (13.89) 看成将预测分布的均值 $A\mu_{n-1}$ 加上一个修正项, 这个修正项正比于预测观测与实际观测之间的误差 $x_n - CA\mu_{n-1}$ 。这个修正的系数由 Kalman 增益矩阵给出。因此我们可以将 Kalman 滤波的过程看成下面的过程: 首先做出后续的预测, 然后使用新的观测来修正这些预测。图 13.21 给出了图形说明。

如果我们考虑下面的情形: 与潜在变量的变化速率相比, 测量误差相对较小, 那么我们看到 z_n 的后验概率分布仅仅依赖于当前的测量 x_n , 这与我们在本节开始时的简单例子中获得的直观感受相符。类似地, 如果与观测的噪声水平相比, 潜在变量的变化速度较慢, 那么我们发现 z_n 的后验均值等于对截止到那个时间的所有测量求平均。

Kalman 滤波的一个重要应用是跟踪。图 13.22 使用一个物体在二维空间移动的简单例子说明了这一点。

目前位置, 我们已经解决了在给定 x_1 到 x_n 的观测的情况下寻找结点 z_n 的后验边缘概率的问题。接下来, 我们考虑在给定 x_1 到 x_N 的所有观测的条件下, 寻找结点 z_n 的边缘概率的问题。对于时序数据, 这对应于将未来的观测以及过去的观测全部包含在内。虽然这无法用于实时预测, 但是它在学习模型的参数中起着重要的作用。通过与隐马尔科夫模型的类比, 这个问题可以这样求解: 从结点 x_N 将信息反向传递到结点 x_1 , 然后将这个信息与计算 $\hat{\alpha}(z_n)$ 的前向信息传递阶段得到的信息相结合。

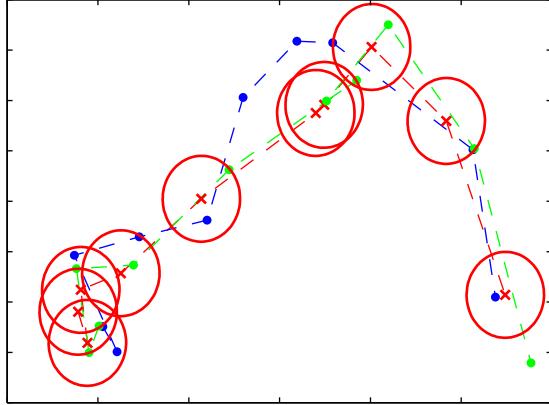


图 13.22: 线性动态系统用于移动物体跟踪的一个说明。蓝点表示在连续的时刻，二维空间中物体的真实位置，绿点表示带有噪声的对位置的测量，红色叉号表示使用Kalman滤波方程推断出的后验概率分布的均值。推断位置的协方差由红色椭圆表示，它对应于一个标准差的轮廓线。

在LDS的文献中，通常根据 $\gamma(z_n) = \hat{\alpha}(z_n)\hat{\beta}(z_n)$ 表示后向递归公式，而是不根据 $\hat{\beta}(z_n)$ 。由于 $\gamma(z_n)$ 一定也是高斯分布，因此我们有

$$\gamma(z_n) = \hat{\alpha}(z_n)\hat{\beta}(z_n) = \mathcal{N}(z_n | \hat{\mu}_n, \hat{V}_n) \quad (13.98)$$

为了推导所求的递归方程，我们从 $\hat{\beta}(z_n)$ 的反向递归方程 (13.62) 开始，它对于连续潜在变量，可以写成

$$c_{n+1}\hat{\beta}(z_n) = \int \hat{\beta}(z_{n+1})p(x_{n+1} | z_{n+1})p(z_{n+1} | z_n) dz_{n+1} \quad (13.99)$$

我们现在将 (13.99) 两侧乘以 $\hat{\alpha}(z_n)$ ，使用公式 (13.75) 和 (13.76) 消去 $p(x_{n+1} | z_{n+1})$ 和 $p(z_{n+1} | z_n)$ 。然后，我们使用 (13.89)、(13.90) 和 (13.91)，以及 (13.98)，经过一些计算，我们有

$$\hat{\mu}_n = \mu_n + J_n(\hat{\mu}_{n+1} + A\mu_n) \quad (13.100)$$

$$\hat{V}_n = V_n + J_n (\hat{V}_{n+1} - P_n) J_n^T \quad (13.101)$$

其中我们定义了

$$J_n = V_n A^T (P_n)^{-1} \quad (13.102)$$

并且我们使用了 $A V_n = P_n J_n^T$ 。注意，这些递归方程要求首先完成前向传递的过程，从而在后向过程中可以使用 μ_n 和 V_n 。

对于EM算法，我们也需要求出一对变量的后验边缘分布，它可以通过公式 (13.65) 求出，形式为

$$\begin{aligned} \xi(z_{n-1}, z_n) &= (c_n)^{-1} \hat{\alpha}(z_{n-1}) p(x_n | z_n) p(z_n | z_{n-1}) \hat{\beta}(z_n) \\ &= \frac{\mathcal{N}(z_{n-1} | \mu_{n-1}, V_{n-1}) \mathcal{N}(z_n | Az_{n-1}, \Gamma) \mathcal{N}(x_n | Cz_n, \Sigma) \mathcal{N}(z_n | \hat{\mu}_n, \hat{V}_n)}{c_n \hat{\alpha}(z_n)} \end{aligned} \quad (13.103)$$

使用公式 (13.84) 消去 $\hat{\alpha}(z_n)$ ，整理，我们看到 $\xi(z_{n-1}, z_n)$ 是一个高斯分布，均值为 $[\hat{\mu}_{n-1}, \hat{\mu}_n]^T$ ， z_n 和 z_{n-1} 之间的协方差为

$$\text{cov}[z_{n-1}, z_n] = J_{n-1} \hat{V}_n \quad (13.104)$$

13.3.2 LDS中的学习

目前为止，我们已经研究了线性动态系统中的推断问题，假设模型的参数 $\theta = \{A, \Gamma, C, \Sigma, \mu_0, P_0\}$ 已知。接下来，我们考虑使用最大似然方法确定这些参数

(Ghahramani and Hinton, 1996b)。由于模型具有潜在变量，因此可以使用第9章讨论的一般形式的EM算法来解决这个问题。

我们可以按照下面的方法推导线性动态系统的EM算法。让我们将算法在某个特定循环上的模型参数估计值记作 $\theta^{\text{旧}}$ 。对于这些参数，我们可以运行推断算法来确定潜在变量的后验概率分布 $p(\mathbf{Z} | \mathbf{X}, \theta^{\text{旧}})$ ，或者更精确地确定那些在M步骤中所需的局部后验边缘概率。特别地，我们需要下面的期望

$$\mathbb{E}[z_n] = \hat{\mu}_n \quad (13.105)$$

$$\mathbb{E}[z_n z_{n-1}^T] = \hat{\mathbf{V}}_n \mathbf{J}_{n-1}^T + \hat{\mu}_n \hat{\mu}_{n-1}^T \quad (13.106)$$

$$\mathbb{E}[z_n z_n^T] = \hat{\mathbf{V}}_n + \hat{\mu}_n \hat{\mu}_n^T \quad (13.107)$$

其中我们已经使用了公式 (13.104)。

现在我们考虑完整数据对数似然函数，它通过对公式 (13.6) 取对数的方式得到，因此结果为

$$\begin{aligned} \ln p(\mathbf{X}, \mathbf{Z} | \boldsymbol{\theta}) &= \ln p(z_1 | \boldsymbol{\mu}_0, \mathbf{P}_0) + \sum_{n=2}^N \ln p(z_n | z_{n-1}, \mathbf{A}, \boldsymbol{\Gamma}) \\ &\quad + \sum_{n=1}^N \ln p(x_n | z_n, \mathbf{C}, \boldsymbol{\Sigma}) \end{aligned} \quad (13.108)$$

其中我们显式地写出了对参数的依赖关系。我们现在对完整数据对数似然函数关于后验概率分布 $p(\mathbf{Z} | \mathbf{X}, \theta^{\text{旧}})$ 取期望，它定义了函数

$$Q(\boldsymbol{\theta}, \theta^{\text{旧}}) = \mathbb{E}_{\mathbf{Z} | \theta^{\text{旧}}} [\ln p(\mathbf{X}, \mathbf{Z} | \boldsymbol{\theta})] \quad (13.109)$$

在M步骤中，函数关于 $\boldsymbol{\theta}$ 的分量进行最大化。

首先考虑参数 $\boldsymbol{\mu}_0$ 和 \mathbf{P}_0 。如果我们使用 (13.77) 消去公式 (13.108) 中的 $p(z_1 | \boldsymbol{\mu}_0, \mathbf{P}_0)$ ，然后关于 \mathbf{Z} 取期望，那么我们有

$$Q(\boldsymbol{\theta}, \theta^{\text{旧}}) = -\frac{1}{2} \ln |\mathbf{P}_0| - \mathbb{E}_{\mathbf{Z} | \theta^{\text{旧}}} \left[\frac{1}{2} (z_1 - \boldsymbol{\mu}_0)^T \mathbf{P}_0^{-1} (z_1 - \boldsymbol{\mu}_0) \right] + \text{常数}$$

其中所有不依赖于 $\boldsymbol{\mu}_0$ 或者 \mathbf{P}_0 的项都被整合到了可加性常数中。使用2.3.4节讨论的高斯分布的最大似然解，关于 $\boldsymbol{\mu}_0$ 和 \mathbf{P}_0 进行最大化很容易进行，结果为

$$\boldsymbol{\mu}_0^{\text{新}} = \mathbb{E}[z_1] \quad (13.110)$$

$$\mathbf{V}_0^{\text{新}} = \mathbb{E}[z_1 z_1^T] - \mathbb{E}[z_1] \mathbb{E}[z_1^T] \quad (13.111)$$

类似地，为了最优化 \mathbf{A} 和 $\boldsymbol{\Gamma}$ ，我们使用公式 (13.75) 消去 (13.108) 中的 $p(z_n | z_{n-1}, \mathbf{A}, \boldsymbol{\Gamma})$ ，结果为

$$\begin{aligned} Q(\boldsymbol{\theta}, \theta^{\text{旧}}) &= -\frac{N-1}{2} \ln |\boldsymbol{\Gamma}| \\ &\quad - \mathbb{E}_{\mathbf{Z} | \theta^{\text{旧}}} \left[\frac{1}{2} \sum_{n=2}^N (z_n - \mathbf{A} z_{n-1})^T \boldsymbol{\Gamma}^{-1} (z_n - \mathbf{A} z_{n-1}) \right] + \text{常数} \end{aligned} \quad (13.112)$$

其中常数项由不依赖与 \mathbf{A} 和 $\boldsymbol{\Gamma}$ 的项组成。关于这些参数最大化可得

$$\mathbf{A}^{\text{新}} = \left(\sum_{n=2}^N \mathbb{E}[z_n z_{n-1}^T] \right) \left(\sum_{n=2}^N \mathbb{E}[z_{n-1} z_{n-1}^T] \right)^{-1} \quad (13.113)$$

$$\begin{aligned}\boldsymbol{\Gamma}^{\text{新}} = \frac{1}{N-1} \sum_{n=2}^N & \left\{ \mathbb{E} [\mathbf{z}_n \mathbf{z}_n^T] - \mathbf{A}^{\text{新}} \mathbb{E} [\mathbf{z}_{n-1} \mathbf{z}_n^T] \right. \\ & \left. - \mathbb{E} [\mathbf{z}_n \mathbf{z}_{n-1}^T] (\mathbf{A}^{\text{新}})^T + \mathbf{A}^{\text{新}} \mathbb{E} [\mathbf{z}_{n-1} \mathbf{z}_{n-1}^T] (\mathbf{A}^{\text{新}})^T \right\}\end{aligned}\quad (13.114)$$

注意， $\mathbf{A}^{\text{新}}$ 必须首先计算，然后它的结果用来确定 $\boldsymbol{\Gamma}^{\text{新}}$ 。

最后，为了确定 \mathbf{C} 和 $\boldsymbol{\Sigma}$ 的新值，我们使用公式(13.76)消去公式(13.108)中的 $p(\mathbf{x}_n | \mathbf{z}_n, \mathbf{C}, \boldsymbol{\Sigma})$ ，可得

$$\begin{aligned}Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{旧}}) = & -\frac{N}{2} \ln |\boldsymbol{\Sigma}| \\ & - \mathbb{E}_{\mathbf{Z}|\boldsymbol{\theta}^{\text{旧}}} \left[\frac{1}{2} \sum_{n=1}^N (\mathbf{x}_n - \mathbf{C} \mathbf{z}_n)^T \boldsymbol{\Sigma}^{-1} (\mathbf{x}_n - \mathbf{C} \mathbf{z}_n) \right] + \text{常数}\end{aligned}$$

关于 \mathbf{C} 和 $\boldsymbol{\Sigma}$ 最大化，可得

$$\mathbf{C}^{\text{新}} = \left(\sum_{n=1}^N \mathbf{x}_n \mathbb{E}[\mathbf{z}_n^T] \right) \left(\sum_{n=1}^N \mathbb{E}[\mathbf{z}_n \mathbf{z}_n^T] \right)^{-1} \quad (13.115)$$

$$\begin{aligned}\boldsymbol{\Sigma}^{\text{新}} = & \frac{1}{N} \sum_{n=1}^N \{ \mathbf{x}_n \mathbf{x}_n^T - \mathbf{C}^{\text{新}} \mathbb{E}[\mathbf{z}_n] \mathbf{x}_n^T \\ & - \mathbf{x}_n \mathbb{E}[\mathbf{z}_n^T] (\mathbf{C}^{\text{新}})^T + \mathbf{C}^{\text{新}} \mathbb{E}[\mathbf{z}_n \mathbf{z}_n^T] (\mathbf{C}^{\text{新}})^T \}\end{aligned}\quad (13.116)$$

我们得到了使用最大似然方法学习线性动态系统的参数的方法。引入先验概率分布得到MAP估计的方法很简单。使用第10章讨论的近似方法，可以得到一个完整的贝叶斯方法。篇幅所限，不在这里详细介绍这些内容。

13.3.3 LDS的推广

与隐马尔科夫模型相同，为了增强模型的能力，我们对推广基本的线性动态系统有着浓厚的兴趣。虽然线性高斯模型的假设会产生高效的推断和学习算法，但是它也暗示了观测变量的边缘概率分布是一个高斯分布，这会产生很大的局限性。线性动态系统的一个简单的推广是使用高斯混合分布作为初始分布 \mathbf{z}_1 。如果这个混合分布有 K 个分量，那么前向递归方程(13.85)会产生隐含变量 \mathbf{z}_n 上的 K 个高斯分布的混合分布，因此模型是可以计算的。

对于许多应用来说，高斯发射概率密度是一个很差的金丝。如果我们尝试使用 K 个高斯分布的混合分布作为发射概率密度，那么后验概率分布 $\hat{\alpha}(\mathbf{z}_1)$ 也会是 K 个高斯分布的混合。然而，根据公式(13.85)，后验概率分布 $\hat{\alpha}(\mathbf{z}_2)$ 由 K^2 个高斯分布混合而成，以此类推， $\hat{\alpha}(\mathbf{z}_n)$ 由 K^n 个高斯分布混合而成。因此，分量的数量随着链的长度指数增长，因此模型无法计算。

更一般地，引入与线性高斯（或者指数族）分布差距较大的转移模型或者发射模型会产生一个无法计算的推断问题。我们可以进行确定性的近似，例如假设的密度过滤或者期望传播，或者我们可以使用13.3.4节讨论的采样方法。一个广泛使用的方法是在预测分布的均值附近进行线性化从而进行了高斯近似，这就产生了推广的Kalman滤波（extended Kalman filter）（Zarchan and Musoff, 2005）。

与隐马尔可夫模型相同，我们可以通过扩展图表示的方法来对基本的线性动态系统进行有趣的推广。例如，切换状态空间模型（switching state space model）（Ghahramani and Hinton, 1998）可以被看成隐马尔科夫模型与一组线性动态系统的组合。模型有多个由连续线性高斯潜在变量组成的马尔科夫链，每一个都类似于之前讨论的线性动态系统的潜在链。模型中还包含了一个在隐马尔科夫模型中使用的离散变量形式的马尔科夫链。在每个时刻的输出按照下面的方式确定：随机选择一个连续潜在链，使用离散潜在变量作为一个开关，然后从对应的条件输出分布发射一个观测。这个模型中精确的推断是无法进行的，但是变分方法会产生出一个高效的推断方法，涉及到沿着每个连续的和离散的马尔科夫链独立进行的前向和后向算法。注意，如果我们考虑离散潜在变量的多个链，然后使用一个作为开关，从剩余的链中选择，那么我们得到了一个只有离散潜在变量的类似的模型，被称为切换隐马尔科夫模型（switching hidden Markov model）。

13.3.4 粒子滤波

对于没有线性高斯分布的动态系统，例如使用非高斯发射概率密度的动态系统，为了得到一个可以计算的推断算法，我们使用采样算法。特别地，我们可以使用11.1.5节讨论的采样-重要性-重采样方法，得到一个顺序的蒙特卡罗算法，被称为粒子滤波。

考虑图13.5中的图模型表示的概率分布，假设我们有观测变量 $\mathbf{X}_n = (\mathbf{x}_1, \dots, \mathbf{x}_n)$ ，我们希望从后验概率分布 $p(\mathbf{z}_n | \mathbf{X}_n)$ 中抽取 L 个样本。使用贝叶斯定理，我们有

$$\begin{aligned}\mathbb{E}[f(\mathbf{z}_n)] &= \int f(\mathbf{z}_n) p(\mathbf{z}_n | \mathbf{X}_n) d\mathbf{z}_n \\ &= \int f(\mathbf{z}_n) p(\mathbf{z}_n | \mathbf{x}_n, \mathbf{X}_{n-1}) d\mathbf{z}_n \\ &= \frac{\int f(\mathbf{z}_n) p(\mathbf{x}_n | \mathbf{z}_n) p(\mathbf{z}_n | \mathbf{X}_{n-1}) d\mathbf{z}_n}{\int p(\mathbf{x}_n | \mathbf{z}_n) p(\mathbf{z}_n | \mathbf{X}_{n-1}) d\mathbf{z}_n} \\ &\simeq \sum_{l=1}^L w_n^{(l)} f(\mathbf{z}_n^{(l)})\end{aligned}\tag{13.117}$$

其中 $\{\mathbf{z}_n^{(l)}\}$ 是从 $p(\mathbf{z}_n | \mathbf{X}_{n-1})$ 中抽取的一组样本，并且我们使用了条件独立性质 $p(\mathbf{x}_n | \mathbf{z}_n, \mathbf{X}_{n-1}) = p(\mathbf{x}_n | \mathbf{z}_n)$ ，这个性质来自于图13.5所示的图模型。采样权值 $\{w_n^{(l)}\}$ 的定义为

$$w_n^{(l)} = \frac{p(\mathbf{x}_n | \mathbf{z}_n^{(l)})}{\sum_{m=1}^L p(\mathbf{x}_n | \mathbf{z}_n^{(m)})}\tag{13.118}$$

其中我们在分子和分母中使用了同样的样本。因此后验概率分布 $p(\mathbf{z}_n | \mathbf{X}_n)$ 由样本集合 $\{\mathbf{z}_n^{(l)}\}$ 以及对应的权值 $\{w_n^{(l)}\}$ 表示。注意，权值一定满足 $0 \leq w_n^{(l)} \leq 1$ 以及 $\sum_l w_n^{(l)} = 1$ 。

由于我们希望找到一个顺序采样的方法，因此我们假设我们在时刻 n 已经得到了一组样本和权值，并且我们顺序地观测到了 \mathbf{x}_{n+1} 的值，我们希望找到时刻 $n+1$ 的权值和样本。我们首先从概率分布 $p(\mathbf{z}_{n+1} | \mathbf{X}_n)$ 中采样。这很容易做到，因为使用贝叶斯定义，我们有

$$\begin{aligned}p(\mathbf{z}_{n+1} | \mathbf{X}_n) &= \int p(\mathbf{z}_{n+1} | \mathbf{z}_n, \mathbf{X}_n) p(\mathbf{z}_n | \mathbf{X}_n) d\mathbf{z}_n \\ &= \int p(\mathbf{z}_{n+1} | \mathbf{z}_n) p(\mathbf{z}_n | \mathbf{X}_n) d\mathbf{z}_n \\ &= \int p(\mathbf{z}_{n+1} | \mathbf{z}_n) p(\mathbf{z}_n | \mathbf{x}_n, \mathbf{X}_{n-1}) d\mathbf{z}_n \\ &= \frac{\int p(\mathbf{z}_{n+1} | \mathbf{z}_n) p(\mathbf{x}_n | \mathbf{z}_n) p(\mathbf{z}_n | \mathbf{X}_{n-1}) d\mathbf{z}_n}{\int p(\mathbf{x}_n | \mathbf{z}_n) p(\mathbf{z}_n | \mathbf{X}_{n-1}) d\mathbf{z}_n} \\ &\simeq \sum_l w_n^{(l)} p(\mathbf{z}_{n+1} | \mathbf{z}_n^{(l)})\end{aligned}\tag{13.119}$$

其中我们使用了条件独立性质

$$p(\mathbf{z}_{n+1} | \mathbf{z}_n, \mathbf{X}_n) = p(\mathbf{z}_{n+1} | \mathbf{z}_n)\tag{13.120}$$

$$p(\mathbf{x}_n | \mathbf{z}_n, \mathbf{X}_{n-1}) = p(\mathbf{x}_n | \mathbf{z}_n)\tag{13.121}$$

这可以通过对图13.5所示的图模型应用d-划分准则的方式得到。公式 (13.119) 的概率分布是一个混合分布，样本可以通过下面的方式得到：根据混合系数 $w^{(l)}$ 指定的概率，选择一个分量 l ，然后从对应的分布中采样。

总结一下，我们可以将粒子滤波算法的每一步看成由两个阶段组成。在时刻 n ，我们有一个后验概率 $p(\mathbf{z}_n | \mathbf{X}_n)$ 的样本表示，它根据 $\{\mathbf{z}_n^{(l)}\}$ 以及对应的权值 $\{w_n^{(l)}\}$ 表示。这可以看成形如 (13.119) 的混合表示。为了得到下一个时刻的对应的表示，我们首先从混合概率

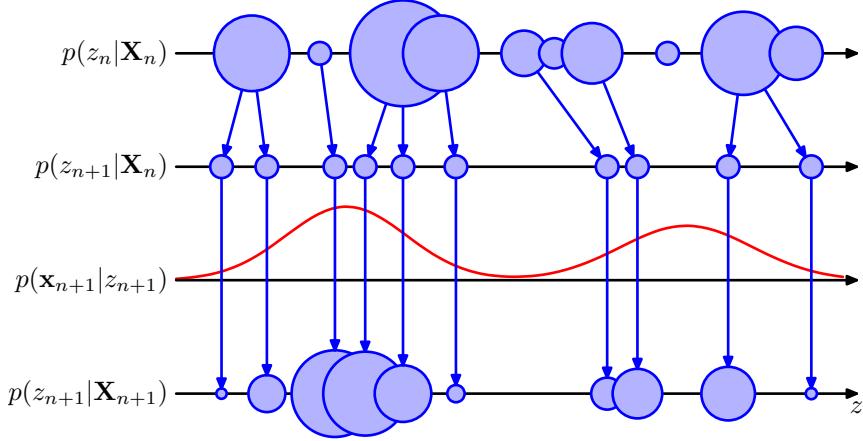


图 13.23: 对于一维潜在空间, 粒子滤波操作的图形表示。在时刻 n , 后验概率分布 $p(z_n | \mathbf{x}_n)$ 被表示为混合概率分布, 用圆圈表示, 它的大小正比于权值 $w_n^{(l)}$ 。之后, 一组 L 个样本从这个概率分布中抽取, 新的权值 $w_{n+1}^{(l)}$ 使用 $p(\mathbf{x}_{n+1} | z_{n+1}^{(l)})$ 计算。

分布 (13.119) 中抽取 L 个样本, 然后对于每个样本, 我们使用新的观测 \mathbf{x}_{n+1} 计算对应的权值 $w_{n+1}^{(l)} \propto p(\mathbf{x}_{n+1} | z_{n+1}^{(l)})$ 。图 13.23 说明了单一变量 z 的情形。

粒子滤波或者顺序蒙特卡罗方法在文献中有多个名字, 包括自助滤波 (bootstrap filter) (Gordon et al., 1993)、最适幸存 (survival of the fittest) (kanazawa et al., 1995) 以及凝结算法 (condensation algorithm) (Isard and Blake, 1998)。

13.4 练习

(13.1) (*) 使用 8.2 节讨论的 d 划分方法, 验证图 13.3 给出的共有 N 个结点的马尔科夫模型满足条件独立性质 (13.3), 其中 $n = 2, \dots, N$ 。类似地, 证明图 13.4 中的总计有 N 个结点的图描述的模型满足条件独立性质

$$p(\mathbf{x}_n | \mathbf{x}_1, \dots, \mathbf{x}_{n-1}) = p(\mathbf{x}_n | \mathbf{x}_{n-1}, \mathbf{x}_{n-2}) \quad (13.122)$$

其中 $n = 3, \dots, N$ 。

(13.2) (**) 考虑对应于图 13.3 的有向图的联合概率分布 (13.2)。使用概率的加和规则和乘积规则, 验证这个联合概率分布满足条件独立性质 (13.3), 其中 $n = 2, \dots, N$ 。类似地, 证明联合概率分布 (13.4) 描述的二阶马尔科夫链满足条件独立性质

$$p(\mathbf{x}_n | \mathbf{x}_1, \dots, \mathbf{x}_{n-1}) = p(\mathbf{x}_n | \mathbf{x}_{n-1}, \mathbf{x}_{n-2}) \quad (13.123)$$

其中 $n = 3, \dots, N$ 。

(13.3) (*) 通过使用 d-划分, 证明, 图 13.5 的有向图表示的状态空间模型的观测变量的分布 $p(\mathbf{x}_1, \dots, \mathbf{x}_N)$ 不满足任何条件独立性质, 因此无法利用任何阶数的马尔科夫性质。

(13.4) (**) 考虑一个马尔科夫模型, 其中发射概率由参数化模型 $p(\mathbf{x} | \mathbf{z}, \mathbf{w})$ 表示, 例如一个线性回归模型或者一个神经网络, 其中 \mathbf{w} 是可调节参数组成的向量。描述参数 \mathbf{w} 可以如何使用最大似然方法从数据中学习到。

(13.5) (**) 通过最大化完整数据对数似然函数的期望 (13.17), 使用拉格朗日乘数来强制满足 π 和 A 上的加和限制, 验证隐马尔可夫模型的初始状态概率和转移概率的参数的 M 步骤方程 (13.18) 和 (13.19)。

(13.6) (*) 证明, 如果隐马尔科夫模型的参数 π 或 A 的任意元素被初始化为零, 那么那些元素在 EM 算法的所有后续更新中会始终保持为零。

(13.7) (*) 考虑带有高斯发射密度的隐马尔可夫模型。证明, 函数 $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{\text{旧}})$ 关于高斯分布的均值和协方差的最大化会得到 M 步骤方程 (13.20) 和 (13.21)。

(13.8) (**) 一个隐马尔可夫模型, 它具有离散的观测, 服从一个多项式分布, 证明, 给定隐含变量的条件下, 观测的条件概率分布为 (13.22), 对应的 M 步骤方程为 (13.23)。对于

具有多个输出变量，且每个变量都由伯努利条件分布控制的隐马尔科夫模型，写出类似的条件概率分布方程和M步骤方程。提示：如果必要的话，参考2.1节和2.2节关于独立同分布数据的对应的最大似然解的讨论。

(13.9) (***) 使用d-划分准则，验证由公式 (13.6) 定义的隐马尔可夫模型的联合概率分布满足的条件独立性质 (13.24) 到 (13.31)。

(13.10) (****) 通过使用概率的加和规则和乘积规则，验证由公式 (13.6) 定义的隐马尔科夫模型的联合概率分布满足的条件独立性质 (13.24) 到 (13.31)。

(13.11) (**) 使用因子图的一个因子的变量上的边缘概率分布的表达式 (8.72) 以及13.2.3节得到的加和-乘积算法中的信息的结果，推导隐马尔可夫模型中两个相继的潜在变量上的联合后验概率分布的结果 (13.43)。

(13.12) (**) 假设我们希望通过最大似然方法，使用由 R 个独立的观测序列组成的数据(记作 $\mathbf{X}^{(r)}$ ，其中 $r = 1, \dots, R$)，训练一个马尔科夫模型。证明，在EM算法的E步骤中，我们通过对每个序列独立地运行 α 递归和 β 递归，简单地计算出了潜在变量上的后验概率分布。同时证明，在M步骤中，初始概率和转移概率的参数的重新估计过程使用的是 (13.18) 和 (13.19) 的一种修改的形式，形式为

$$\pi_k = \frac{\sum_{r=1}^R \gamma(z_{1k}^{(r)})}{\sum_{r=1}^R \sum_{j=1}^K \gamma(z_{1j}^{(r)})} \quad (13.124)$$

$$A_{jk} = \frac{\sum_{r=1}^R \sum_{n=2}^N \xi(z_{n-1,j}^{(r)}, z_{n,k}^{(r)})}{\sum_{r=1}^R \sum_{l=1}^K \sum_{n=2}^N \xi(z_{n-1,l}^{(r)}, z_{n,l}^{(r)})} \quad (13.125)$$

其中为了记号的方便，我们假设序列具有相同的长度（很容易推广到具有不同长度的序列的情形）。类似地，高斯发射模型的均值的重估计的M步骤方程为

$$\mu_k = \frac{\sum_{r=1}^R \sum_{n=1}^N \gamma(z_{nk}^{(r)}) \mathbf{x}_n^{(r)}}{\sum_{r=1}^R \sum_{n=1}^N \gamma(z_{nk}^{(r)})} \quad (13.126)$$

注意，其他发射模型的参数和概率分布的M步骤方程的形式与此类似。

(13.13) (**) 使用因子图中从因子结点到变量结点传递的信息的定义 (8.64)，以及隐马尔可夫模型的联合概率分布的表达式 (13.6)，证明alpha信息的定义 (13.50) 与 (13.34) 相同。

(13.14) (**) 使用因子图中从因子结点到变量结点传递的信息的定义 (8.67)，以及隐马尔可夫模型的联合概率分布的表达式 (13.6)，证明beta信息的定义 (13.52) 与 (13.35) 相同。

(13.15) (**) 使用隐马尔可夫模型中的边缘概率的表达式 (13.33) 和 (13.43)，推导用重新缩放的变量表达的对应的结果 (13.64) 和 (13.65)。

(13.16) (****) 本练习中，我们直接从联合概率分布的表达式 (13.6) 中推导维特比算法的前向信息传递方程。这涉及到在所有的隐含变量 z_1, \dots, z_N 上最大化。通过取对数，然后交换最大化运算与求和运算的顺序，推导递归方程 (13.68)，其中 $\omega(z_n)$ 由公式 (13.70) 定义。证明这个递归的初始条件由 (13.69) 给出。

(13.17) (*) 证明，图13.18表示的输入输出隐马尔可夫模型的有向图可以表示成图13.15所示的树结构的因子图的形式。写出初始因子 $h(z_1)$ 和一般的因子 $f_n(z_{n-1}, z_n)$ 的表达式，其中 $2 \leq n \leq N$ 。

(13.18) (****) 使用练习13.17的结果，推导图13.18所示的输入输出隐马尔可夫模型的前向后向算法的递归方程，包括初始条件。

(13.19) (*) Kalman滤波和平滑方程使得线性动态系统中各个潜在变量上的后验概率分布(以所有观测变量为条件)可以高效地求出。证明，通过分别最大化每个后验概率分布得到的潜在变量序列的值与潜在变量的最可能的序列相同。为了完成这一点，我们注意到线性动态系统中的潜在变量和观测变量的联合概率分布是高斯分布，从而所有的条件概率分布和边缘概率分布也是高斯分布，然后使用公式 (2.98) 给出的结果即可。

(13.20) (**) 使用公式 (2.115) 的结果证明 (13.87)。

(13.21) (**) 使用公式 (2.115) 和 (2.116) 的结果，以及矩阵恒等式 (C.5) 和 (C.7)，推导结果 (13.89)、(13.90) 和 (13.91)，其中Kalman增益矩阵 \mathbf{K}_n 由公式 (13.92) 定义。

(13.22) (***) 使用公式 (13.93) 以及定义 (13.76) 、 (13.77) 和公式 (2.115) 给出的结果, 推导 (13.96)。

(13.23) (***) 使用公式 (13.93) 以及定义 (13.76) 、 (13.77) 和公式 (2.116) 给出的结果, 推导 (13.94) 、 (13.95) 和 (13.97)。

(13.24) (***) 考虑对公式 (13.75) 和 (13.76) 的推广, 其中我们在高斯均值中引入常数项 \mathbf{a} 和 \mathbf{c} , 即

$$p(z_n | z_{n-1}) = \mathcal{N}(z_n | \mathbf{A}z_{n-1} + \mathbf{a}, \mathbf{\Gamma}) \quad (13.127)$$

$$p(\mathbf{x}_n | z_n) = \mathcal{N}(\mathbf{x}_n | \mathbf{C}z_n + \mathbf{c}, \mathbf{\Sigma}) \quad (13.128)$$

证明, 这个推广可以在本章讨论的框架中进行改写, 方法是定义一个状态向量 \mathbf{z} , 它具有一个额外的分量固定为 1, 然后对矩阵 \mathbf{A} 和 \mathbf{C} 进行增广, 使其包含额外的一列, 对应于参数 \mathbf{a} 和 \mathbf{c} 。

(13.25) (**) 本练习中, 我们证明, 当 Kalman 滤波方程被应用于独立观测时, 方程会简化为 2.3 节给出的单一高斯分布的最大似然解的结果。考虑寻找一个高斯随机变量 x 的均值 μ 的问题, 其中我们给定了一组独立的观测 $\{x_1, \dots, x_N\}$ 。为了对这个量进行建模, 我们可以使用由公式 (13.75) 和 (13.76) 控制的线性动态系统, 潜在变量为 $\{z_1, \dots, z_N\}$, 其中 $\mathbf{C} = 1, \mathbf{A} = 1, \mathbf{\Gamma} = 0$ 。令初始状态的参数 $\boldsymbol{\mu}_0$ 和 \mathbf{P}_0 分别记作 μ_0 和 σ_0^2 , 假设 $\mathbf{\Sigma}$ 变成了 σ^2 。从公式 (13.89) 和 (13.90) 给出的一般结果以及 (13.94) 和 (13.95) 开始, 写出对应的 Kalman 滤波方程。证明, 这些方程等价于直接考察独立数据得到的结果 (2.141) 和 (2.142)。

(13.26) (****) 考虑 13.3 节讨论的线性动态系统的一种等价于概率 PCA 的具体实例, 即转移矩阵 $\mathbf{A} = \mathbf{0}$, 协方差 $\mathbf{\Gamma} = \mathbf{I}$, 噪声协方差 $\mathbf{\Sigma} = \sigma^2 \mathbf{I}$ 。通过使用矩阵求逆的恒等式 (C.7), 证明, 如果发射概率密度矩阵 \mathbf{C} 被记作 \mathbf{W} , 那么隐含状态上的后验概率分布由公式 (13.89) 定义, 并且公式 (13.90) 会简化为概率 PCA 的结果 (12.42), 其中我们假设公式 (12.42) 中 $\boldsymbol{\mu} = \mathbf{0}$ 。

(13.27) (*) 考虑 13.3 节讨论的线性动态系统, 其中观测噪声的幅值趋于零, 即 $\mathbf{\Sigma} = \mathbf{0}$ 。证明, 在 $\mathbf{C} = \mathbf{I}$ 的情况下, z_n 的后验概率分布的均值为 \mathbf{x}_n , 方差为零。这与我们的直觉相符, 即如果没有噪声, 我们就可以简单地使用当前的观测 \mathbf{x}_n 来估计状态变量 z_n , 忽略所有之前的观测。

(13.28) (****) 考虑 13.3 节讨论的线性动态系统的一个具体的实例, 其中状态变量 z_n 被限制为与前一个状态变量相等, 这对应于 $\mathbf{A} = \mathbf{I}$ 和 $\mathbf{\Gamma} = \mathbf{0}$ 。为了简化, 我们还假设 $\mathbf{C} = \mathbf{I}$, 以及 $\mathbf{P}_0 \rightarrow \infty$, 从而 z 的初始条件不再重要, 预测完全由数据确定。使用归纳法, 证明状态 z_n 的后验均值由 $\mathbf{x}_1, \dots, \mathbf{x}_n$ 的均值确定。这对应于直观的结果, 即如果状态变量是常量, 那么我们最好的估计时对观测求平均。

(13.29) (****) 从反向递归方程 (13.99) 开始, 推导高斯线性动态系统的 RTS 平滑方程 (13.100) 和 (13.101)。

(13.30) (**) 从状态空间模型的对之间的后验边缘概率的结果 (13.65) 开始, 推导高斯线性动态系统情形下的具体形式 (13.103)。

(13.31) (**) 从公式 (13.103) 给出的结果开始, 通过使用公式 (13.84) 消去 $\hat{\alpha}(z_n)$, 验证 z_n 和 z_{n-1} 之间的协方差的结果 (13.104)。

(13.32) (**) 验证线性动态系统中 $\boldsymbol{\mu}_0$ 和 \mathbf{P}_0 的 M 步骤方程的结果 (13.110) 和 (13.111)。

(13.33) (**) 验证线性动态系统中 \mathbf{A} 和 $\mathbf{\Gamma}$ 的 M 步骤方程的结果 (13.113) 和 (13.114)。

(13.34) (**) 验证线性动态系统中 \mathbf{C} 和 $\mathbf{\Sigma}$ 的 M 步骤方程的结果 (13.115) 和 (13.116)。

14 组合模型

在之前的章节中，我们研究了一系列不同的模型用于解决分类问题和回归问题。经常发现的一件事情是，我们可以通过以某种方式将多个模型结合到一起的方法来提升性能，而不是独立地使用一个单独的模型。例如，我们可以训练 L 个不同的模型，然后使用每个模型给出的预测的平均值进行预测。这样的模型的组合有时被称为委员会（committee）。在14.2节，我们讨论在实际问题中使用委员会概念的方式，我们也会给出一些深刻的认识，来理解它为什么有时会是一个有效的方法。

委员会方法的一个重要的变体，被称为提升方法（boosting）。这种方法按顺序训练多个模型，其中用来训练一个特定模型的误差函数依赖于前一个模型的表现。与单一模型相比，这个模型可以对性能产生显著的提升，将在14.3节讨论。

与对一组模型的预测求平均的方法不同，另一种形式的模型组合是选择一个模型进行预测，其中模型的选择是输入变量的一个函数。因此不同的模型用于对输入空间的不同的区域进行预测。这种方法的一种广泛使用的框架被称为决策树（decision tree），其中选择的过程可以被描述为一个二值选择的序列，对应于对树结构的遍历，将在14.4节讨论。这种情况下，各个单独的模型通常被选得非常简单，整体的模型灵活性产生于与输入相关的选择过程。决策树既可以应用于分类问题也可以应用于回归问题。

决策树的一个局限性是对于输入空间的划分基于的是一种硬划分，对于输入变量的任意给定的值，只有一个模型用于做出预测。通过将一个概率框架用于模型组合，决策的过程可以被软化，将在14.5节讨论。例如，如果我们有一组 K 个模型用于描述条件概率分布 $p(t | \mathbf{x}, k)$ ，其中 \mathbf{x} 是输入变量， t 是目标变量， $k = 1, \dots, K$ 是模型的索引，那么我们可以进行一种概率形式的混合，形式为

$$p(t | \mathbf{x}) = \sum_{k=1}^K \pi_k(\mathbf{x}) p(t | \mathbf{x}, k) \quad (14.1)$$

其中 $\pi_k(\mathbf{x}) = p(k | \mathbf{x})$ 表示与输入相关的混合系数。这样的模型可以被看成混合概率分布，其中分量的概率密度以及混合系数都以输入变量为条件，被称为专家混合（mixture of experts）。这种模型与5.6节讨论的混合密度网络密切相关。

14.1 贝叶斯模型平均

将模型组合方法与贝叶斯模型平均方法区分开是很重要的，这两种方法经常被弄混淆。为了理解二者的差异，考虑使用高斯混合模型进行概率密度估计的例子，其中若干的高斯分量以概率的方式进行组合。模型包含一个二值潜在变量 z ，它表示混合分布中的哪个分量用于生成对应的数据点。因此，模型通过联合概率分布

$$p(\mathbf{x}, z) \quad (14.2)$$

进行具体化，观测变量 \mathbf{x} 上的对应的概率密度通过对潜在变量求和的方式得到，即

$$p(\mathbf{x}) = \sum_z p(\mathbf{x}, z) \quad (14.3)$$

在我们的高斯混合模型的例子中，这会得到一个概率分布，形式为

$$p(\mathbf{x}) = \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \quad (14.4)$$

各个符号的含义与之前相同。这是模型组合的一个例子。对于独立同分布的数据，我们可以使用公式(14.3)将数据集 $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ 的边缘概率写成下面的形式

$$p(\mathbf{X}) = \prod_{n=1}^N p(\mathbf{x}_n) = \prod_{n=1}^N \left[\sum_{z_n} p(\mathbf{x}_n, z_n) \right] \quad (14.5)$$

因此我们看到，每个观测数据点 \mathbf{x}_n 有一个对应的潜在变量 z_n 。

现在假设我们有若干个不同的模型，索引为 $h = 1, \dots, H$ ，先验概率分布为 $p(h)$ 。例如，一个模型可能是高斯混合模型，另一个模型可能是柯西分布的混合。数据集上的边缘概率分布为

$$p(\mathbf{X}) = \sum_{h=1}^H p(\mathbf{X} | h) p(h) \quad (14.6)$$

这是贝叶斯模型平均的一个例子。这个在 h 上的求和式的意义是，只有一个模型用于生成整个数据集， h 上的概率分布仅仅反映了我们对于究竟是哪个模型用于生成数据的不确定性。随着数据集规模的增加，这个不确定性会减小，后验概率分布 $p(h | \mathbf{X})$ 会逐渐集中于模型中的某一个。

这就强调了贝叶斯模型平均和模型组合的一个关键的不同，因为在贝叶斯模型平均中，整个数据集由单一的模型生成。相反，当我们像 (14.5) 那样组合多个模型时，我们看到数据集中的不同的数据点可以由潜在变量 z 的不同的值生成，即由不同的分量生成。

虽然我们研究的是边缘概率分布 $p(\mathbf{X})$ ，但是同样的讨论适用于预测分布 $p(\mathbf{x} | \mathbf{X})$ 以及诸如 $p(\mathbf{t} | \mathbf{x}, \mathbf{X}, \mathbf{T})$ 这样的条件概率分布。

14.2 委员会

构建一个委员会的最简单的方法是对一组独立的模型的预测取平均。这样的方法的动机可以从频率学家的观点看出来。这种观点考虑偏置和方差之间的折中，它将模型的误差分解为偏置分量和方差分量，其中偏置分量产生于模型和真实的需要预测的函数之间的差异，方差分量表示模型对于单独的数据点的敏感性。回忆一下，根据图 3.5，当我们使用正弦数据训练多个多项式函数，然后对得到的函数求平均时，来自方差项的贡献倾向于被抵消掉，从而产生了预测的提升。当我们对一组低偏置的模型（对应于高阶多项式）求平均时，我们得到的对用于生成数据的正弦函数的精确的预测。

当然，在实际应用中，我们只有一个单独的数据集，因此我们必须寻找一种方式来表示委员会中不同模型之间的变化性。一种方法是使用 1.2.3 节讨论的自助（bootstrap）数据集。考虑一个回归问题，其中我们试图预测一个连续变量的值，并且假设我们生成了 M 个自助数据集然后使用每个数据集训练处了预测模型的一个独立的副本 $y_m(\mathbf{x})$ ，其中 $m = 1, \dots, M$ 。委员会预测为

$$y_{COM}(\mathbf{x}) = \frac{1}{M} \sum_{m=1}^M y_m(\mathbf{x}) \quad (14.7)$$

这个方法被称为自助聚集（bootstrap aggregation）或者打包（bagging）（Breiman, 1996）。

假设我们试图预测的真实的回归函数为 $h(\mathbf{x})$ ，从而每个模型的输出可以写成真实值加上误差的形式，即

$$y_m(\mathbf{x}) = h(\mathbf{x}) + \epsilon_m(\mathbf{x}) \quad (14.8)$$

这样，平方和误差函数的形式为

$$\mathbb{E}_{\mathbf{x}} [\{y_m(\mathbf{x}) - h(\mathbf{x})\}^2] = \mathbb{E}_{\mathbf{x}} [\epsilon_m(\mathbf{x})^2] \quad (14.9)$$

其中 $\mathbb{E}_{\mathbf{x}}[\cdot]$ 表示关于输入向量 \mathbf{x} 的一个频率学家的期望。于是，各个模型独立预测的平均误差为

$$E_{AV} = \frac{1}{M} \sum_{m=1}^M \mathbb{E}_{\mathbf{x}} [\epsilon_m(\mathbf{x})^2] \quad (14.10)$$

类似地，委员会方法的预测 (14.7) 的期望误差为

$$\begin{aligned} E_{COM} &= \mathbb{E}_{\mathbf{x}} \left[\left\{ \frac{1}{M} \sum_{m=1}^M y_m(\mathbf{x}) - h(\mathbf{x}) \right\}^2 \right] \\ &= \mathbb{E}_{\mathbf{x}} \left[\left\{ \frac{1}{M} \sum_{m=1}^M \epsilon_m(\mathbf{x}) \right\}^2 \right] \end{aligned} \quad (14.11)$$

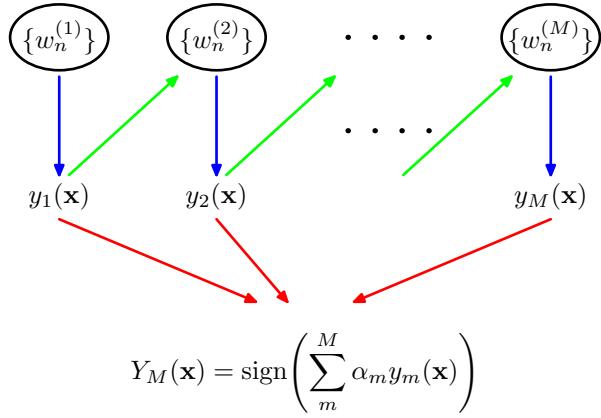


图 14.1: 提升方法框架的图形表示。每个基分类器 $y_m(\mathbf{x})$ 都在训练数据集的一个加权形式（蓝色箭头）上进行训练，权值 $w_n^{(m)}$ 依赖于前一个基分类器 $y_{m-1}(\mathbf{x})$ （绿色箭头）的表现。一旦所有的基分类器训练完毕，它们被组合得到最终的分类器 $Y_M(\mathbf{x})$ （红色箭头）。

如果我们假设误差的均值为零，且不具有相关性，即

$$\mathbb{E}_{\mathbf{x}}[\epsilon_m(\mathbf{x})] = 0 \quad (14.12)$$

$$\mathbb{E}_{\mathbf{x}}[\epsilon_m(\mathbf{x})\epsilon_l(\mathbf{x})] = 0, \quad m \neq l \quad (14.13)$$

那么我们有

$$E_{COM} = \frac{1}{M} E_{AV} \quad (14.14)$$

这个显然具有戏剧性的结果表明，一个模型的平均误差可以仅仅通过对模型的 M 个版本求平均的方式减小 M 倍。不幸的是，它依赖于我们的关键假设，即由各个单独的模型产生的误差是不相关的。在实际应用中，误差通常是高度相关的，因此整体的误差下降是通常是很小的。然而，可以证明，委员会误差的期望不会超过各个分量模型的期望误差，即 $E_{COM} \leq E_{AV}$ 。为了得到更显著的提升，我们转向一种更加复杂的构建委员会的方法，被称为提升方法。

14.3 提升方法

提升方法是一种很强大的方法，它将多个“基”分类器进行组合，产生一种形式的委员会，委员会的表现会比任何一个基分类器的表现好得多。这里，我们介绍提升方法的最广泛使用的一种形式，被称为AdaBoost，是“可调节提升方法 (adaptive boosting)”的简称，由Freund and Schapire (1996) 提出。即使基分类器的表现仅仅比随机猜测的表现稍好，提升方法仍可以产生比较好的结果。这种基分类器有时被称为弱学习器 (weak learner)。提升方法最初被用来解决分类问题，但是也可以推广到回归问题 (Friedman, 2001)。

提升方法和委员会方法（例如上面讨论的打包方法）的主要不同在于，基分类器是顺序训练的，每个基分类器使用数据集的一个加权形式进行训练，其中与每个数据点相关联的权系数依赖于前一个分类器的表现。特别地，被一个基分类器误分类的点在训练序列中的下一个分类器时会被赋予更高的权重。一旦所有的分类器都训练完毕，那么它们的预测就会通过加权投票的方法进行组合，如图14.1所示。

考虑一个二分类问题，其中训练数据由输入向量 $\mathbf{x}_1, \dots, \mathbf{x}_N$ 以及对应的二值目标变量 t_1, \dots, t_N 组成，其中 $t_n \in \{-1, 1\}$ 。每个数据点被赋予了一个关联的权值参数 w_n ，对于所有的数据点，它都被初始化为 $\frac{1}{N}$ 。我们假设我们有一个使用加权数据训练基分类器的方法，得到函数 $y(\mathbf{x}) \in \{-1, 1\}$ 。在算法的每个阶段，AdaBoost使用一个数据集训练一个新的分类器，其中权系数根据前一个训练的分类器的表现进行调节，从而为误分类的数据点赋予更高的权值。最后，当我们训练了所需数量的基分类器之后，它们进行组合，形成一个委员会，组合的系数会为不同的基分类器赋予不同的权值。AdaBoost算法的精确形式叙述如下。

- 初始化数据加权系数 $\{w_n\}$ ，方法是对 $n = 1, \dots, N$ ，令 $w_n^{(1)} = \frac{1}{N}$ 。

- 对于 $m = 1, \dots, M$:
 - 使用训练数据调节一个分类器 $y_m(\mathbf{x})$, 调节的目标是最小化加权的误差函数

$$J_m = \sum_{n=1}^N w_n^{(m)} I(y_m(\mathbf{x}_n) \neq t_n) \quad (14.15)$$

其中 $I(y_m(\mathbf{x}_n) \neq t_n)$ 是一个示性函数, 当 $y_m(\mathbf{x}_n) \neq t_n$ 时, 值为 1, 其他情况下值为 0。

- 计算

$$\epsilon_m = \frac{\sum_{n=1}^N w_n^{(m)} I(y_m(\mathbf{x}_n) \neq t_n)}{\sum_{n=1}^N w_n^{(m)}} \quad (14.16)$$

然后计算

$$\alpha_m = \ln \left\{ \frac{1 - \epsilon_m}{\epsilon_m} \right\} \quad (14.17)$$

- 更新数据权系数

$$w_n^{(m+1)} = w_n^{(m)} \exp\{\alpha_m I(y_m(\mathbf{x}_n) \neq t_n)\} \quad (14.18)$$

- 使用最终的模型进行预测, 形式为

$$Y_M(\mathbf{x}) = \text{sign} \left(\sum_{m=1}^M \alpha_m y_m(\mathbf{x}) \right) \quad (14.19)$$

我们看到第一个基分类器 $y_1(\mathbf{x})$ 使用全部相等的加权系数 $w_n^{(1)}$ 进行训练, 因此它对应于训练单一的分类器的通常的步骤。根据 (14.18), 我们看到在后续的迭代过程中, 权系数 $w_n^{(m)}$ 对于误分类的数据点会增大, 对于正确分类的数据点不改变。因此后续的分类器就会更关注那些被前一个分类器错误分类的数据点。 ϵ_m 表示每个基分类器在数据集上的错误率的加权度量。于是我们看到公式 (14.17) 定义的权系数 α_m 会在计算整体输出 (14.19) 时, 为更准确的分类器赋予更高的权值。

AdaBoost 算法如图 14.2 所示, 数据集是图 A.7 所示的分类数据集的由 30 个数据点组成的子集。这里, 每个基分类器由一个输入变量的阈值组成。这个简单的分类器对应于一种被称为“决策树桩”的决策树形式, 即一个具有单结点的决策树。因此, 每个基学习器根据一个输入特征是否超过某个阈值对输入进行分类, 因此仅仅使用一个与一个坐标轴垂直的线性决策面将空间划分为两个区域。

14.3.1 最小化指数误差

提升方法最早起源于统计学习理论, 得到了泛化误差的上界。然而, 这些上界过于宽松, 没有实际的价值。提升方法的实际表现要远优于上界给出的值。Friedman et al. (2000) 根据对一个指数误差函数的顺序最小化, 给出了提升方法的一个不同的且非常简单的表述。

考虑下面定义的指数误差函数

$$E = \sum_{n=1}^N \exp\{-t_n f_m(\mathbf{x}_n)\} \quad (14.20)$$

其中 $f_m(\mathbf{x})$ 是一个根据基分类器 $y_l(\mathbf{x})$ 的线性组合定义的分类器, 形式为

$$f_m(\mathbf{x}) = \frac{1}{2} \sum_{l=1}^m \alpha_l y_l(\mathbf{x}) \quad (14.21)$$

$t_n \in \{-1, 1\}$ 是训练集目标值。我们的目标是关于权系数 α_l 和基分类器 $y_l(\mathbf{x})$ 最小化 E 。

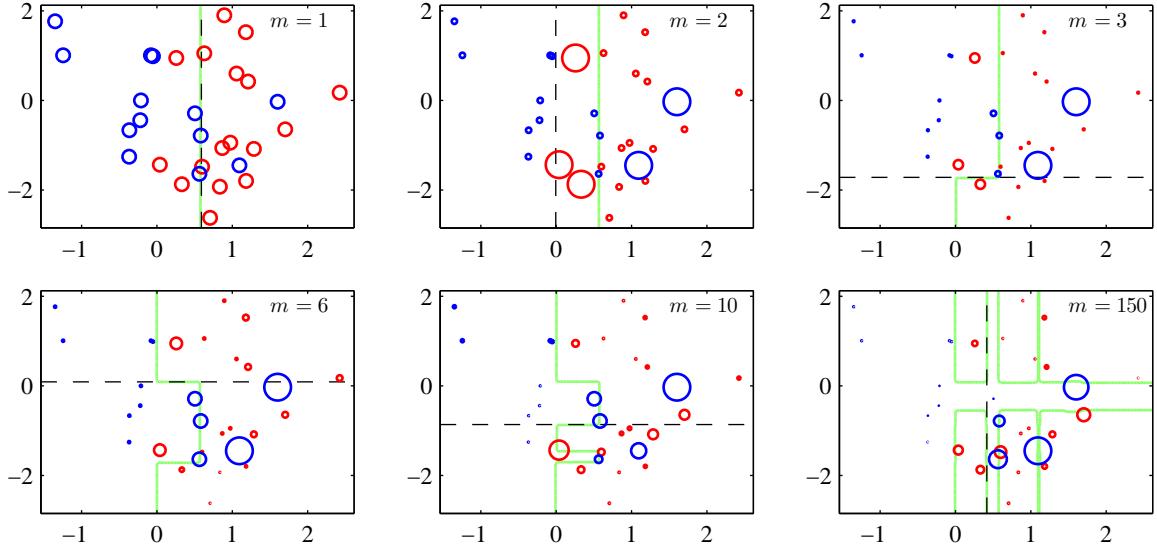


图 14.2: 提升方法的说明，其中基学习器由作用于某个轴的简单的阈值组成。每张图给出了目前训练的基学习器的数量 m ，以及最近的基学习器的决策边界（黑色虚线）和组合的决策边界（绿色实线）。每个数据点用圆圈表示，它的半径表示在训练最近添加的基学习器时数据点的权值。因此，例如，我们看到被 $m = 1$ 的学习器误分类的点在训练 $m = 2$ 的学习器时被赋予了更高的权值。

然而，我们不进行误差函数的全局最小化，而是假设基分类器 $y_1(\mathbf{x}), \dots, y_{m-1}(\mathbf{x})$ 以及它们的系数 $\alpha_1, \dots, \alpha_{m-1}$ 固定，因此我们只关于 α_m 和 $y_m(\mathbf{x})$ 进行最小化。分离出基分类器 $y_m(\mathbf{x})$ 的贡献，我们可以将误差函数写成

$$\begin{aligned} E &= \sum_{n=1}^N \exp \left\{ -t_n f_{m-1}(\mathbf{x}_n) - \frac{1}{2} t_n \alpha_m y_m(\mathbf{x}_n) \right\} \\ &= \sum_{n=1}^N w_n^{(m)} \exp \left\{ -\frac{1}{2} t_n \alpha_m y_m(\mathbf{x}_n) \right\} \end{aligned} \quad (14.22)$$

其中，系数 $w_n^{(m)} = \exp \{-t_n f_{m-1}(\mathbf{x}_n)\}$ 可以被看做常数，因为我们只针对 α_m 和 $y_m(\mathbf{x})$ 进行最优化。如果我们将被 $y_m(\mathbf{x})$ 正确分类的数据点的集合记作 \mathcal{T}_m ，并且将剩余的误分类的点记作 \mathcal{M}_m ，那么我们可以将误差函数写成下面的形式

$$\begin{aligned} E &= e^{-\frac{\alpha_m}{2}} \sum_{n \in \mathcal{T}_m} w_n^{(m)} + e^{\frac{\alpha_m}{2}} \sum_{n \in \mathcal{M}_m} w_n^{(m)} \\ &= (e^{\frac{\alpha_m}{2}} - e^{-\frac{\alpha_m}{2}}) \sum_{n=1}^N w_n^{(m)} I(y_m(\mathbf{x}_n) \neq t_n) + e^{-\frac{\alpha_m}{2}} \sum_{n=1}^N w_n^{(m)} \end{aligned} \quad (14.23)$$

当我们关于 $y_m(\mathbf{x})$ 进行最小化时，我们看到第二项是常数，因此这等价于对 (14.15) 进行最小化，因为在求和式前面的整个可乘性因子不影响最小值的位置。类似地，关于 α_m 最小化，我们得到了公式 (14.17)，其中 ϵ_m 由公式 (14.16) 定义。

根据公式 (14.22)，我们看到，找到 α_m 和 $y_m(\mathbf{x})$ 之后，数据点的权值使用下面的公式进行更新

$$w_n^{(m+1)} = w_n^{(m)} \exp \left\{ -\frac{1}{2} t_n \alpha_m y_m(\mathbf{x}_n) \right\} \quad (14.24)$$

使用下面的事实

$$t_n y_m(\mathbf{x}_n) = 1 - 2I(y_m(\mathbf{x}_n) \neq t_n) \quad (14.25)$$

我们看到在下一次迭代中，权值 $w_n^{(m)}$ 的更新为

$$w_n^{(m+1)} = w_n^{(m)} \exp \left(-\frac{\alpha_m}{2} \right) \exp \{ \alpha_m I(y_m(\mathbf{x}_n) \neq t_n) \} \quad (14.26)$$

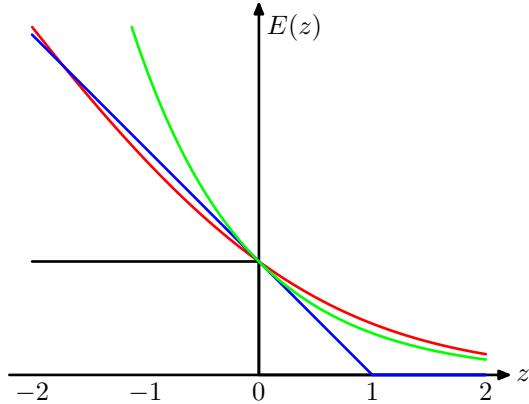


图 14.3: 指数误差函数 (绿色)、缩放的交叉熵误差函数 (红色) 以及支持向量机使用的铰链误差函数 (蓝色) 和误分类误差函数 (黑色) 的图像。注意, 对于 $z = ty(\mathbf{x})$ 的较大的负值, 交叉熵误差函数给出了一个线性增长的惩罚, 而指数误差函数给出了一个指数增长的惩罚。

由于 $\exp(-\frac{\alpha_m}{2})$ 与 n 无关, 因此我们看到它对于所有数据点的权值都贡献一个相同的因子, 从而可以丢弃。这样我们就得到了公式 (14.18)。

最后, 一旦所有的基分类器被训练完毕, 新数据点通过计算由 (14.21) 定义的组合函数的符号进行分类。由于因子 $\frac{1}{2}$ 不影响符号, 因此可以省略, 得到了公式 (14.19)。

14.3.2 提升方法的误差函数

AdaBoost 算法最小化的指数误差函数与之前章节讨论的误差函数不同。为了更深刻地理解指数误差函数的本质, 我们首先考虑期望误差, 形式为

$$\mathbb{E}_{\mathbf{x}, t}[\exp\{-ty(\mathbf{x})\}] = \sum_t \int \exp\{-ty(\mathbf{x})\} p(t | \mathbf{x}) p(\mathbf{x}) d\mathbf{x} \quad (14.27)$$

如果我们关于所有可能的函数 $y(\mathbf{x})$ 进行变分最小化, 那么我们有

$$y(\mathbf{x}) = \frac{1}{2} \ln \left\{ \frac{p(t=1 | \mathbf{x})}{p(t=-1 | \mathbf{x})} \right\} \quad (14.28)$$

它是 log odds 函数的一半。因此 AdaBoost 算法是在由基分类器的线性组合表示的函数空间中, 寻找对 log odds 的最好的近似, 对应于顺序最优化策略下的受限最小化。这个结果说明了在公式 (14.19) 中使用符号函数得到最终的分类决策的原因。

我们已经看到, 二分类问题的交叉熵误差函数 (4.90) 的最小函数 $y(\mathbf{x})$ 由后验类概率密度给出。在目标变量 $t \in \{-1, 1\}$ 的情形下, 我们已经看到误差函数为 $\ln(1 + \exp(-yt))$ 。图 14.3 给出了它与指数误差函数的对比, 其中我们将交叉熵误差函数除以了一个常数因子 $\ln(2)$, 从而它穿过点 $(0, 1)$, 使得更加容易进行对比。我们看到, 这两个函数都可以看成对理想误分类误差函数的连续近似。指数误差的一个优点是它的顺序最小化会得到简单的 AdaBoost 方法。然而, 一个缺点是, 与交叉熵误差函数相比, 它对负的 $ty(\mathbf{x})$ 的惩罚较大。特别地, 我们看到对于 ty 的很大的负值, 交叉熵随着 $|ty|$ 线性增长, 而指数误差随着 $|ty|$ 指数增长。因此指数误差函数对于异常点和误分类的数据点并不鲁棒。交叉熵误差函数和指数误差函数的另一个区别是后者无法表示为任何具有良好定义的概率模型的似然函数。此外, 指数误差无法推广到具有 $K > 2$ 个类别的分类问题, 这再次与概率模型的交叉熵相反, 它可以很容易地推广, 得到 (4.108)。

将提升方法表示为指数误差下的可加性模型的最优化 (Friedman et al., 2000) 引出了一大类与提升方法相似的算法, 包括对多类问题的推广, 方法是使用其他的误差函数。它也引出了对于回归问题的推广 (Friedman, 2001)。如果我们考虑回归问题的平方和误差函数, 那么形如 (14.21) 的可加性模型的顺序最小化仅仅涉及到将新的分类器根据前一个模型的残留误差 $t_n - f_{m-1}(\mathbf{x}_n)$ 进行调节。然而, 正如我们已经注意到的那样, 平方和误差函数对于异常点不鲁棒。这个问题可以通过将绝对偏差 $|y - t|$ 应用到提升方法中的方式得到解决。图 14.4 给出了这两个误差函数的对比。

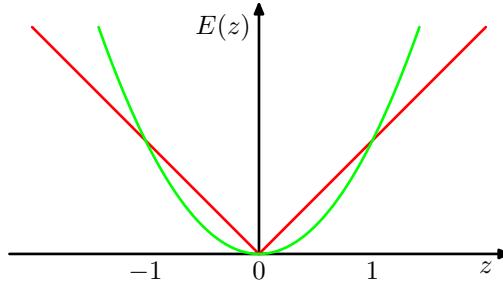


图 14.4: 平方误差 (绿色) 和绝对误差 (红色) 的对比。图中展示了后者为较大的误差赋予较低的重视程度, 从而后者对于异常点和误分类的点更加鲁棒。

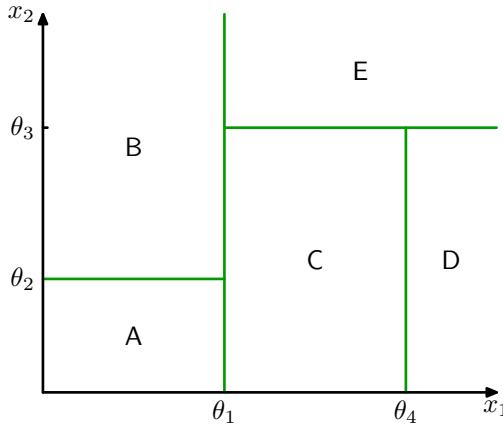


图 14.5: 二维输入空间使用与坐标轴平行的边界被划分为了五个区域。

14.4 基于树的模型

有许多简单但广泛使用的模型, 它们将输入空间划分为超立方体区域, 超立方体的边与坐标轴对齐, 然后为每个区域分配一个简单的模型 (例如, 一个常数)。这些模型可以被看成一种模型组合方法, 其中只有一个模型对于输入空间中任意给定点的预测起作用。给定一个新的输入 x , 选择一个具体的模型的过程可以由一个顺序决策的过程描述, 这个过程对应于对一个二叉树 (每个节点划分为两个分支的树) 的遍历。这里, 我们关注一个特定的基于树的框架, 被称为分类与回归树 (classification and regression tree), 或者CART (Breiman et al., 1994), 虽然还有很多其他的变体, 例如ID3和C4.5 (Quinlan, 1986; Quinlan, 1993)。

图14.5和图14.6给出了对输入空间进行递归二分的例子, 以及对应的树结构。在这个例子中, 第一步根据 $x_1 \leq \theta_1$ 或 $x_1 > \theta_1$, 将输入空间划分为两个区域, 其中 θ_1 是一个模型参数。这创建了两个子区域, 每个区域之后可以独立地进行划分。例如, 区域 $x_1 \leq \theta_1$ 进一步根据 $x_2 \leq \theta_2$ 或 $x_2 > \theta_2$ 进行进一步划分, 得到的区域被记作 A 和 B。递归的过程可以用图14.6给出的二叉树的遍历进行描述。对于任意新的输入 x , 我们确定它所属区域的方法是, 从树顶端的根结点开始, 根据每个结点的决策准则, 沿着路径向下走到具体的叶结点。注意, 这种决策树不是概率图模型。

在每个区域内, 有一个单独的模型预测目标变量的值。例如, 在回归问题中, 我们简单地在每个区域内预测一个常数, 或者在分类问题中, 我们将每个区域分配为一个具体的类别。基于树的模型的一个关键的性质是模型可以由人类表述, 因为模型对应于作用在输入变量上的一个二元决策序列。这使得模型在例如医疗诊断领域很流行。例如, 为了预测一个病人的疾病, 我们可以首先问“病人的体温是否大于某个阈值?”。如果回答是肯定的, 那么我们可以问“病人的血压是否低于某个阈值?”。然后树的每个叶结点都与一个具体的诊断相关联。

为了从一个训练数据集里学习到这样的一个模型, 我们必须确定树的结构, 包括在每个结点处选择哪个输入变量构成划分准则, 以及用于划分的阈值参数 θ_i 的值。我们也必须确定每个区域内的预测变量的值。

首先考虑一个回归问题, 其中我们的目标是从输入变量 D 维向量 $x = (x_1, \dots, x_D)^T$ 中预测单

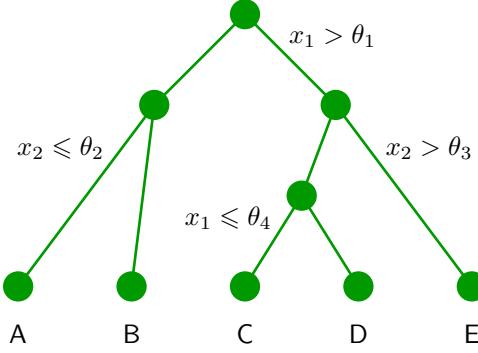


图 14.6: 对应于图14.5的输入空间的划分的二叉树。

一的目标变量 t 的值。训练数据由输入向量 $\{x_1, \dots, x_N\}$ 以及对应的连续标签 $\{t_1, \dots, t_N\}$ 组成。如果输入空间的划分给定，并且我们最小化平方和误差函数，那么在任意给定区域的预测变量的最优值就是落在哪个区域的数据点的 t_n 值的平均。

现在考虑如何确定决策树的结构。即使对于结点数量固定的树，确定最优结构（包括每次划分使用的输入变量以及对应的阈值）来最小化平方和误差函数的问题通常在计算上是不可行的，因为可能的组合数量相当大。相反，我们通常使用贪心的最优化。从对应于整个输入空间的一个单独的根结点开始，然后通过每次添加一个结点的方式构建树。在每一步，输入空间中会有若干个可以切分的候选的区域，对应于向当前的树中添加一对叶结点。对于每个这种候选区域，我们要选择使用 D 个输入变量中的哪一个进行划分，以及阈值的大小。划分区域的选择以及输入变量和阈值的选择可以通过彻底搜索的方式高效地进行联合最优化。我们注意到，对于给定的划分变量和阈值的选择，预测变量的最优选择是数据的局部平均值，如前所述。对划分变量的所有可能选择重复上述步骤，得到最小的平方和误差的一个划分变量被保留下。

得到构建树的贪心策略之后，剩下的问题是如何停止添加结点。一个简单的方法是当残留误差的减小量低于某个阈值时停止。然而，我们通过实验发现，经常出现这样的情形：没有划分方式会使误差函数产生显著的减小，但是再进行几次划分之后，就会找到一个使误差函数显著减小的划分方式。因此，在实际应用中通常构建一个较大的树，使用基于与叶结点关联的数据点数量的停止准则，然后进行剪枝，生成最终的树。剪枝的过程基于的准则会在残留误差与模型复杂度之间进行平衡。我们将剪枝开始时的树记作 T_0 ，然后我们对于 $T \subset T_0$ ，如果它能够通过从 T_0 剪枝（即通过合并对应区域来收缩内部结点）的方式被得到，那么它就被定义为 T_0 的一个子树。假设叶结点的索引为 $\tau = 1, \dots, |T|$ ，叶结点 τ 表示具有 N_τ 个数据点的区域 \mathcal{R}_τ ， $|T|$ 表示叶结点的总数。那么区域 \mathcal{R}_τ 给出的最优的预测为

$$y_\tau = \frac{1}{N_\tau} \sum_{x_n \in \mathcal{R}_\tau} t_n \quad (14.29)$$

它对于残留的平方和误差的贡献为

$$Q_\tau(T) = \sum_{x_n \in \mathcal{R}_\tau} \{t_n - y_\tau\}^2 \quad (14.30)$$

从而剪枝准则为

$$C(T) = \sum_{\tau=1}^{|T|} Q_\tau(T) + \lambda |T| \quad (14.31)$$

正则化参数 λ 确定了整体的残留平方和误差与模型复杂度之间的折中，模型复杂度用叶结点的数量 $|T|$ 表示，它的值通过交叉验证的方式确定。

对于分类问题，树的构建和剪枝的过程很类似，区别在于平方和误差函数被替换为一个更合适的性能的度量。如果我们将 $p_{\tau k}$ 定义为区域 \mathcal{R}_τ 中被分配到类别 k 的数据点的比例，其中 $k = 1, \dots, K$ ，那么经常使用的两个度量是交叉熵

$$Q_\tau(T) = - \sum_{k=1}^K p_{\tau k} \ln p_{\tau k} \quad (14.32)$$

以及基尼系数Gini index

$$Q_\tau(T) = \sum_{k=1}^K p_{\tau k}(1 - p_{\tau k}) \quad (14.33)$$

如果对于任意的 $k = 1, \dots, K$ 都有 $p_{\tau k} = 1$, 那么这两个量都等于零, 此时对于所有 $j \neq k$ 都有 $p_{\tau j} = 0$ 。如果对于所有的 $k = 1, \dots, K$ 都有 $p_{\tau k} = \frac{1}{K}$, 那么这两个量都会达到最大值。这两个量倾向于让区域中属于同一个类别的数据点的比例较高。在构建树的过程中, 与分类错误率相比, 交叉熵和基尼系数是一个更好的度量, 因为这两个量对于结点的概率更敏感。并且, 与分类错误率不同, 它们是可微的, 因此更适合基于梯度的最优化方法。对于接下来对树的剪枝过程, 通常使用分类错误率。

像CART这种树模型的可以由人类进行表述这一性质通常被视为它的一个重要的优点。然而, 在实际应用中, 学习到的特定的树结构对于数据集的细节非常敏感, 从而训练集的一个微小的改变就会产生一个相当不同的划分集合 (Hastie et al., 2001)。

本节讨论的这种基于树的方法有一些其他的问题。一个问题是, 划分边界是与特征空间的坐标轴对齐的, 这相当不好。例如, 为了将最优边界与坐标轴成45度角的两个类别划分开, 我们需要相当多的与坐标轴平行的划分, 这个数量要远大于一个单一的不与坐标轴平行的划分的数量。此外, 决策树中的划分是硬划分, 从而输入空间中的每个区域与一个叶结点模型关联, 并且只与一个叶结点模型关联。最后一个问题是解决回归问题时相当严重, 其中我们通常的目标是对光滑的函数建模, 但是树模型生成了分段常数的预测, 划分的边界是不连续的。

14.5 条件混合模型

我们已经看到, 标准的决策树被限制为对输入空间的硬的、与坐标轴对齐的划分。这些限制可以通过引入软的、概率形式的划分的方式得到缓解, 这些划分是所有输入变量的函数, 而不仅仅是某个输入变量的函数。这样做的代价是它的直观意义的消失。如果我们也给叶结点的模型赋予一个概率的形式, 那么我们就得到了一个纯粹的概率形式的基于树的模型, 被称为专家层次混合 (hierarchical mixture of experts), 将在14.5.3节讨论。

另一种得到专家层次混合模型的方法是从标准的非条件密度模型 (例如高斯分布) 的概率混合开始, 将分量概率密度替换为条件概率分布。这里, 我们考虑线性回归模型的混合 (14.5.1节) 以及logistic回归模型的混合 (14.5.2节)。在最简单的情况下, 混合系数与输入变量无关。如果我们进行进一步的泛化, 使得混合系数同样依赖于输入, 那么我们就得到了专家混合 (mixture of experts) 模型。最后, 如果我们使得混合模型的每个分量本身都是一个专家混合模型, 那么我们就得到了专家层次混合模型。

14.5.1 线性回归模型的混合

用概率形式表示线性回归模型的众多优点之一是它可以用作更复杂的概率模型的一个分量。例如, 将表示线性回归模型的条件概率分布看成有向概率图中的一个结点, 即可完成这件事。这里, 我们考虑一个简单的例子, 对应于线性回归模型的混合, 它是9.2节讨论的高斯混合模型的一个直接推广, 推广到了条件高斯分布的情形。

因此, 我们考虑 K 个线性回归模型, 每个模型都由自己的权参数 \mathbf{w}_k 控制。在许多应用中, 比较合适的做法是对所有 K 个分量使用一个共同的噪声方差, 由精度参数 β 控制, 这正是我们这里讨论的情形。我们再次将注意力集中于单一目标变量 t , 但是推广到多个输出是很容易的。如果我们将混合系数记作 π_k , 那么混合概率分布可以写成

$$p(t | \boldsymbol{\theta}) = \sum_{k=1}^K \pi_k \mathcal{N}(t | \mathbf{w}_k^T \boldsymbol{\phi}, \beta^{-1}) \quad (14.34)$$

其中 $\boldsymbol{\theta}$ 表示模型中所有可调节参数的集合, 即 $\mathbf{W} = \{\mathbf{w}_k\}$, $\boldsymbol{\pi} = \{\pi_k\}$ 以及 β 。给定一组观测数据集 $\{\mathbf{t}_n, t_n\}$, 这个模型的对数似然函数的形式为

$$\ln p(\mathbf{t} | \boldsymbol{\theta}) = \sum_{n=1}^N \ln \left(\sum_{k=1}^K \pi_k \mathcal{N}(t_n | \mathbf{w}_k^T \boldsymbol{\phi}_n, \beta^{-1}) \right) \quad (14.35)$$

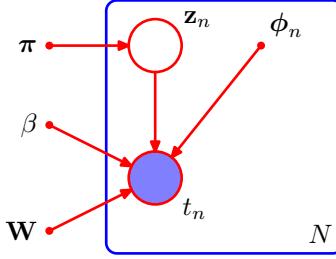


图 14.7: 表示由公式 (14.35) 定义的线性回归模型的混合模型的概率有向图。

其中 $\mathbf{t} = (t_1, \dots, t_N)^T$ 表示目标变量组成的向量。

为了最大化这个似然函数，我们可以再次使用EM算法。可以证明它是9.2节讨论的无条件高斯混合模型的EM算法的一个简单推广。于是我们可以基于我们对无条件混合分布的经验构造模型，引入一组二值潜在变量 $Z = \{z_n\}$ ，其中 $z_{nk} \in \{0, 1\}$ ，其中对于每个数据点 n ，所有的 $k = 1, \dots, K$ 中只有一个元素为1，其余元素都等于0。等于1的元素表示哪个混合分布用于生成数据点。潜在变量与观测变量的联合概率分布可以用图14.7的图模型表示。

这样，完整数据的对数似然函数的形式为

$$\ln p(\mathbf{t}, Z | \boldsymbol{\theta}) = \sum_{n=1}^N \sum_{k=1}^K z_{nk} \ln \{\pi_k \mathcal{N}(t_n | \mathbf{w}_k^T \phi_n, \beta^{-1})\} \quad (14.36)$$

EM算法在开始时，首先选择模型参数的初始值 $\boldsymbol{\theta}^{(0)}$ 。在E步骤中，这些参数用于计算每个数据点 n 的每个分量 k 的后验概率分布或者“责任”，结果为

$$\gamma_{nk} = \mathbb{E}[z_{nk}] = p(k | \phi_n, \boldsymbol{\theta}^{(0)}) = \frac{\pi_k \mathcal{N}(t_n | \mathbf{w}_k^T \phi_n, \beta^{-1})}{\sum_j \pi_j \mathcal{N}(t_n | \mathbf{w}_j^T \phi_n, \beta^{-1})} \quad (14.37)$$

然后，“责任”被用于确定完整数据对数似然函数关于后验概率分布 $p(Z | \mathbf{t}, \boldsymbol{\theta}^{(0)})$ 的期望，形式为

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(0)}) = \mathbb{E}_Z [\ln p(\mathbf{t}, Z | \boldsymbol{\theta})] = \sum_{n=1}^N \sum_{k=1}^K \gamma_{nk} \{\ln \pi_k + \ln \mathcal{N}(t_n | \mathbf{w}_k^T \phi_n, \beta^{-1})\}$$

在M步骤中，我们关于 $\boldsymbol{\theta}$ 最大化函数 $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(0)})$ ，保持 γ_{nk} 不变。对于关于混合系数 π_k 的最优化，我们需要考虑限制条件 $\sum_k \pi_k = 1$ ，这使用拉格朗日乘数法即可完成，得到了 π_k 的M步骤重估计方程，形式为

$$\pi_k = \frac{1}{N} \sum_{n=1}^N \gamma_{nk} \quad (14.38)$$

注意，这个函数形式与公式 (9.22) 给出的无条件的简单高斯混合的对应结果形式相同。

接下来，考虑关于第 k 个线性回归模型的参数向量 \mathbf{w}_k 的最大化。代入高斯分布的表达式，我们看到 $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(0)})$ 关于参数向量 \mathbf{w}_k 的函数形式为

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(0)}) = \sum_{n=1}^N \gamma_{nk} \left\{ -\frac{\beta}{2} (t_n - \mathbf{w}_k^T \phi_n)^2 \right\} + \text{常数} \quad (14.39)$$

其中常数项包含来自 $j \neq k$ 的其他权向量 \mathbf{w}_j 的贡献。注意，我们最大化的量类似于单一线性回归模型的标准平方和误差函数 (3.12) 的负对数，但是包含了责任项 γ_{nk} 。这代表了加权最小平方 (weighted least squares) 问题，其中对应于第 n 个数据点的项带有一个加权系数 $\beta \gamma_{nk}$ ，它可以被看成每个数据点的有效精度。我们看到，混合模型中的每个分量线性回归模型由自身的参数向量 \mathbf{w}_k 控制，在M步骤中使用整个数据集分别进行调节，但是每个数据点 n 由责任项 γ_{nk} 加权，它表示模型 k 对这个数据点的作用。令 (14.39) 关于 \mathbf{w}_k 的导数等于零，可得

$$0 = \sum_{n=1}^N \gamma_{nk} (t_n - \mathbf{w}_k^T \phi_n) \phi_n \quad (14.40)$$

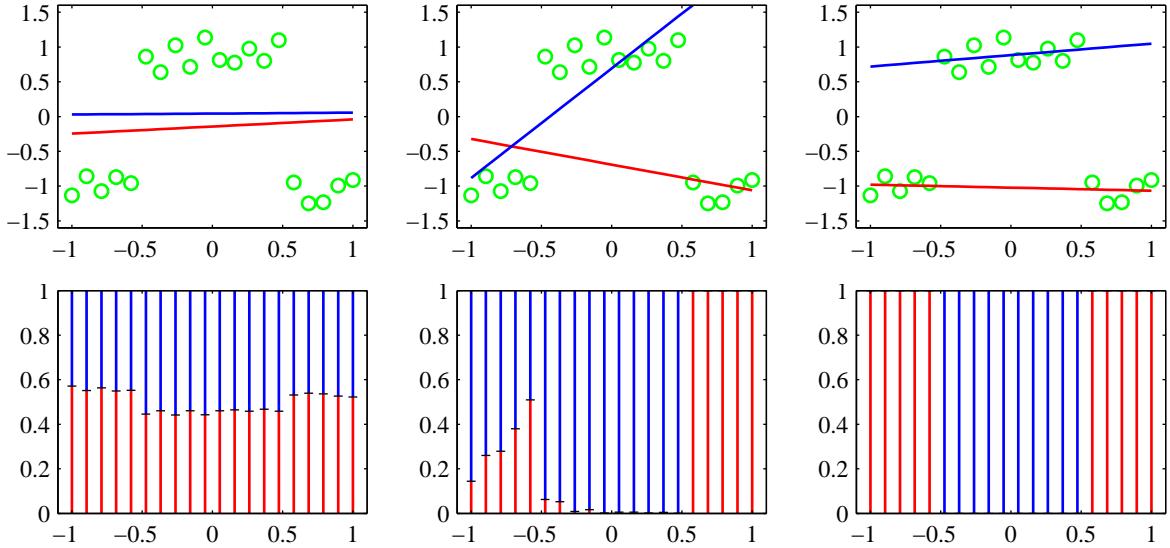


图 14.8: 人工生成的数据的例子，用绿色点表示，具有一个输入变量 x 和一个输出变量 t 。同时画出了两个线性回归模型的混合，它的均值函数 $y(x, \mathbf{w}_k)$ 用蓝线和红线表示，其中 $k \in \{1, 2\}$ 。上方三张图表示初始配置（左图）、运行了 30 轮 EM 迭代的结果（中图）以及运行了 50 轮 EM 迭代的结果（右图）。这里， β 被初始化为目标值集合的真实方差的倒数。下方三张图将每个数据点的对应的责任项用竖直线表示，其中蓝色线段的长度表示那个数据点的蓝色线的后验概率（红色线段的含义与之类似）。

它可以用矩阵的记号表示为

$$0 = \Phi^T \mathbf{R}_k (\mathbf{t} - \Phi \mathbf{w}_k) \quad (14.41)$$

其中 $\mathbf{R}_k = \text{diag}(\gamma_{nk})$ 是一个 $N \times N$ 的对角矩阵。解出 \mathbf{w}_k ，我们有

$$\mathbf{w}_k = (\Phi^T \mathbf{R}_k \Phi)^{-1} \Phi^T \mathbf{R}_k \mathbf{t} \quad (14.42)$$

它表示一组修改过的规范方程，对于加权的最小平方问题，与 logistic 回归问题中得到的结果 (4.99) 具有相同的形式。注意，在每个 E 步骤之后，矩阵 \mathbf{R}_k 会发生变化，因此我们在后续的 M 步骤中必须重新解规范方程。

最后，我们关于 β 最大化 $Q(\theta, \theta^{\text{旧}})$ 。只保留依赖于 β 的项，函数 $Q(\theta, \theta^{\text{旧}})$ 可以写成

$$Q(\theta, \theta^{\text{旧}}) = \sum_{n=1}^N \sum_{k=1}^K \gamma_{nk} \left\{ \frac{1}{2} \ln \beta - \frac{\beta}{2} (t_n - \mathbf{w}_k^T \phi_n)^2 \right\} \quad (14.43)$$

令它关于 β 的导数等于零，整理，我们得到了 β 的 M 步骤方程，形式为

$$\frac{1}{\beta} = \frac{1}{N} \sum_{n=1}^N \sum_{k=1}^K \gamma_{nk} (t_n - \mathbf{w}_k^T \phi_n)^2 \quad (14.44)$$

在图 14.8 中，我们使用了一个简单的例子来说明这个 EM 算法。这个例子中，我们根据数据集来调整由两条直线组成的混合模型，数据集有一个输入变量 x 和一个目标变量 t 。预测密度 (14.34) 如图 14.9 所示，使用了从 EM 算法中得到的收敛的参数值，对应于图 14.8 的右图。图中同时给出的是拟合单一的线性回归模型的结果，它给出了一个单峰的预测密度。我们看到，混合模型可以更好地表示数据分布，这一点通过更高的似然函数值反映出来。然而，混合模型也将相当大的概率质量分配到了没有数据的区域，因为它的预测分布对于 x 的所有值来说是双峰的。这个问题可以这样解决：将模型扩展，使得混合系数本身是 x 的一个函数，这就产生了 5.6 节讨论的混合密度网络模型，以及 14.5.3 节讨论的专家层次混合模型。

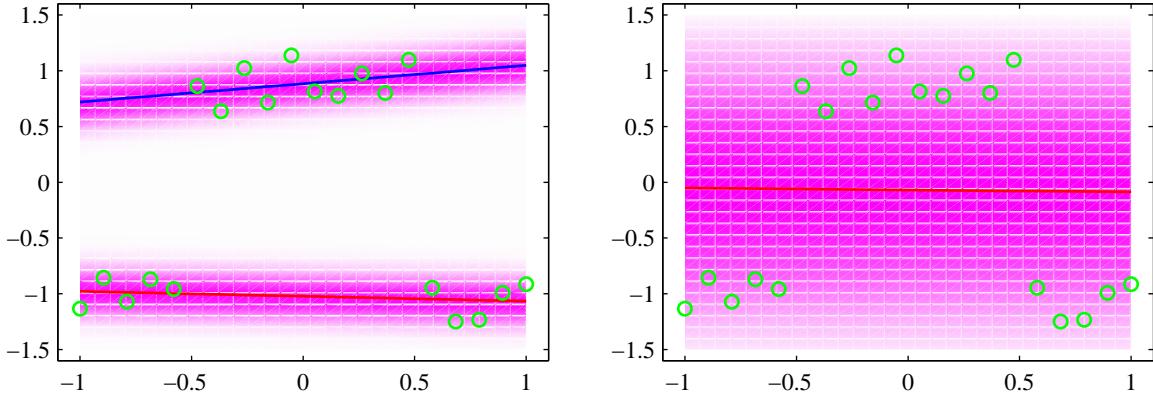


图 14.9: 左图表示对应于图14.8的收敛解的预测条件概率密度。对数似然函数的值为-3.0。在特定的 x 处，穿过图像的垂直切片表示条件概率分布 $p(t | x)$ ，可以看到它是双峰的。右图给出了使用最大似然方法用同样的数据集拟合的单一线性回归模型。模型的对数似然函数值较小，为-27.6。

14.6 logistic模型的混合

由于线性回归模型定义了给定输入变量的条件下目标变量的一个条件概率分布，因此很容易将其用作混合模型中的分量分布，从而与单一的logistic回归模型相比，可以表示更丰富的一类条件概率分布。这个例子涉及到对本书前面章节讨论的思想的一个直接组合，有助于帮助读者巩固这些知识。

对于 K 个logistic回归模型来说，目标变量的条件概率分布为

$$p(t | \phi, \theta) = \sum_{k=1}^K \pi_k y_k^{t_k} [1 - y_k]^{1-t_k} \quad (14.45)$$

其中 ϕ 是特征向量， $y_k = \sigma(\mathbf{w}_k^T \phi)$ 是分量 k 的输出， θ 表示可调节参数，即 $\{\pi_k\}$ 和 $\{\mathbf{w}_k\}$ 。

现在假设我们有一个数据集 $\{\phi_n, t_n\}$ 。从而对应的似然函数为

$$p(\mathbf{t} | \theta) = \prod_{n=1}^N \left(\sum_{k=1}^K \pi_k y_{nk}^{t_n} [1 - y_{nk}]^{1-t_n} \right) \quad (14.46)$$

其中 $y_{nk} = \sigma(\mathbf{w}_k^T \phi_n)$ ， $\mathbf{t} = (t_1, \dots, t_N)^T$ 。我们可以使用EM算法迭代地最大化这个似然函数。这涉及到引入潜在变量 z_{nk} ，它对应于每个数据点 n 的用1-of- K 方式编码的二值指示器变量。完整数据的似然函数为

$$p(\mathbf{t}, \mathbf{Z} | \theta) = \prod_{n=1}^N \prod_{k=1}^K \{ \pi_k y_{nk}^{t_n} [1 - y_{nk}]^{1-t_n} \}^{z_{nk}} \quad (14.47)$$

其中 \mathbf{Z} 是潜在变量矩阵，元素为 z_{nk} 。我们通过选择模型参数的一个初始值 $\theta^{(0)}$ 来初始化EM算法。之后在E步骤中，我们使用这些参数值来计算每个数据点 n 的分量 k 的后验概率，形式为

$$\gamma_{nk} = \mathbb{E}[z_{nk}] = p(k | \phi_n, \theta^{(0)}) = \frac{\pi_k y_{nk}^{t_n} [1 - y_{nk}]^{1-t_n}}{\sum_j \pi_j y_{nj}^{t_n} [1 - y_{nj}]^{1-t_n}} \quad (14.48)$$

这些责任项然后用于寻找完整数据对数似然函数的期望，它作为 θ 的一个函数，形式为

$$\begin{aligned} Q(\theta, \theta^{(0)}) &= \mathbb{E}_{\mathbf{Z}} [\ln p(\mathbf{t}, \mathbf{Z} | \theta)] \\ &\sum_{n=1}^N \sum_{k=1}^K \gamma_{nk} \{ \ln \pi_k + t_n \ln y_{nk} + (1 - t_n) \ln (1 - y_{nk}) \} \end{aligned} \quad (14.49)$$



图 14.10: logistic 回归模型的混合的例子。左图给出了从两个类别中抽取的数据点，两个类别分别用红色和蓝色表示，其中背景颜色（从纯红变化到纯蓝）表示模型标签的真实概率。中图表示使用最大似然方法拟合单一的 logistic 回归模型的结果，其中背景颜色表示类别标签的对应的概率。由于颜色几乎是均匀的紫色，因此我们看到模型在输入空间中的大部分区域都会分配一个近似为 0.5 的概率。右图给出了使用两个 logistic 回归模型的混合模型进行调节的结果，它对于蓝色类别中的许多点，都会给正确的标签赋予高得多的概率。

M 步骤涉及到关于 θ 最大化这个函数，保持 $\theta^{\text{旧}}$ 不变，从而 γ_{nk} 保持不变。关于 π_k 的最大化可以使用通常的方式进行，引入拉格朗日乘数来强制满足 $\sum_k \pi_k = 1$ 的限制，得到下面的熟悉的结果

$$\pi_k = \frac{1}{N} \sum_{n=1}^N \gamma_{nk} \quad (14.50)$$

为了确定 $\{w_k\}$ ，我们注意到 $Q(\theta, \theta^{\text{旧}})$ 由一组下标为 k 项的求和式组成，它只依赖于向量 w_k 中的一个，因此不同的向量在 EM 算法的 M 步骤中可以独立进行优化。换句话说，不同的分量只通过责任项产生相互作用，它在 M 步骤中是固定的。注意，M 步骤没有封闭解，必须使用例如迭代重加权最小平方 (IRLS) 算法迭代地求解。对于向量 w_k 的梯度和 Hessian 矩阵为

$$\nabla_k Q = \sum_{n=1}^N \gamma_{nk} (t_n - y_{nk}) \phi_n \quad (14.51)$$

$$H_k = -\nabla_k \nabla_k Q = \sum_{n=1}^N \gamma_{nk} y_{nk} (1 - y_{nk}) \phi_n \phi_n^T \quad (14.52)$$

其中 ∇_k 表示关于 w_k 的梯度。对于固定的 γ_{nk} ，梯度和 Hessian 矩阵独立于 $j \neq k$ 的 $\{w_j\}$ ，因此我们可以使用 IRLS 算法分别对每个 w_k 求解。因此分量 k 的 M 步骤方程仅仅对应于使用数据集调整一个单独的 logistic 回归模型，其中数据点 n 携带权值 γ_{nk} 。图 14.10 给出了 logistic 回归模型的混合模型应用于简单的分类问题中的例子。将这个模型推广为 softmax 模型的混合模型来处理多类问题是很容易的。

14.6.1 专家混合

在 14.5.1 节，我们考虑了线性回归模型的混合，在 14.5.2 节，我们讨论了线性分类器的类似的混合。虽然这些简单的混合扩展了线性模型的灵活程度，包含了更复杂的（例如多峰的）预测分布，但是它们仍然具有很大的局限性。我们可以进一步增强这些模型的能力，方法是使得混合系数本身是输入变量的函数，即

$$p(t | x) = \sum_{k=1}^K \pi_k(x) p_k(t | x) \quad (14.53)$$

这被称为专家混合 (mixture of experts) 模型 (Jacobs et al., 1991)，其中混合系数 $\pi_k(x)$ 被称为门函数 (gating function)，各个分量密度 $p_k(t | x)$ 被称为专家 (expert)。属于背后的思想是，

不同的分量可以对输入空间的不同区域的概率分布进行建模（它们是在它们自己的区域做预测的“专家”），门函数确定哪个分量控制哪个区域。

门函数 $\pi_k(\mathbf{x})$ 必须满足混合系数通常的限制，即 $0 \leq \pi_k(\mathbf{x}) \leq 1$ 以及 $\sum_k \pi_k(\mathbf{x}) = 1$ 。因此它们可以通过例如线性softmax函数 (4.104) 和 (4.105) 表示。如果专家也是线性（回归或分类）模型，那么整个模型可以使用EM算法高效地调节，在M步骤中要使用迭代重加权最小平方 (Jordan and Jacobs, 1994)。

由于门函数和专家函数使用了线性模型，因此这样的模型仍然有很大的局限性。一个更加灵活的模型时使用多层门函数，得到了专家层次混合 (hierarchical mixture of experts) 模型或者HME模型 (Jordan and Jacobs, 1994)。为了理解这个模型的结构，假设一个混合分布，它的每个分量本身都是一个混合分布。对于无条件的混合分布，层次混合简单地等价于一个普通的混合分布。然而，当混合系数与输入相关时，层次模型就变得不普通了。HME模型也可以被看成14.4节讨论的决策树的概率版本，并且与之前一样可以通过最大似然的方式使用EM算法以及M步骤中的IRLS算法高效计算。Bishop and Svensén (2003) 基于变分推断提出了HME的一个贝叶斯方法。

我们这里不会详细讨论HME。然而，值得指出的一点是，它与5.6节讨论的混合密度网络 (mixture density network) 有着密切的联系。专家混合的主要的优点在于它可以通过EM算法最优化，其中每个混合分量以及门函数的M步骤涉及到一个凸优化（虽然整体的最优化不是凸优化）。相反，混合密度网络方法的一个优势是分量密度和混合系数共享神经网络的隐含单元。此外，与专家层次混合相比，在混合密度网络中，对输入空间的划分更加放松，因为划分不仅是软划分，并且不限于与坐标轴平行，而且还可以是非线性的。

14.7 练习

(14.1) 考虑一组形式为 $p(\mathbf{t} | \mathbf{x}, z_h, \theta_h, h)$ 的模型，其中 \mathbf{x} 是输入向量， \mathbf{t} 是目标向量， h 表示不同模型的索引， z_h 是模型 h 的潜在变量， θ_h 是模型 h 的参数向量。假设模型的先验概率分布为 $p(h)$ ，我们给定了一个训练数据集 $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ 和 $\mathbf{T} = \{\mathbf{t}_1, \dots, \mathbf{t}_N\}$ 。写出需要计算预测分布 $p(\mathbf{t} | \mathbf{x}, \mathbf{X}, \mathbf{T})$ 所需的公式，其中潜在变量和模型索引都被边缘化出去。使用这些公式，说明不同模型的贝叶斯平均和单一模型中使用潜在变量的不同。

(14.2) (*) 一个简单的委员会模型的平方和误差函数的期望 E_{AV} 可以由公式 (14.10) 定义，委员会本身的期望误差由公式 (14.11) 定义。假设各自的误差满足公式 (14.12) 和 (14.13)，推导公式 (14.14) 给出的结果。

(14.3) (*) 通过使用Jensen不等式 (1.115)，对于凸函数 $f(x) = x^2$ 这一具体情形，证明，公式 (14.10) 给出的一个简单的委员会模型的平方和误差函数的期望的平均值 E_{AV} ，以及公式 (14.11) 给出的委员会本身的期望误差 E_{COM} ，满足

$$E_{COM} \leq E_{AV} \quad (14.54)$$

(14.4) (**) 通过使用Jensen不等式 (1.115)，证明上一个练习中推导的结果对于任意的误差函数 $E(y)$ 都成立，而不仅仅是平方和误差函数，假设 y 是一个凸函数。

(14.5) (**) 考虑一个委员会模型，其中我们允许各个分量模型具有不同的权值，即

$$y_{COM}(\mathbf{x}) = \sum_{m=1}^M \alpha_m y_m(\mathbf{x}) \quad (14.55)$$

为了确保预测 $y_{COM}(\mathbf{x})$ 保持在合理的范围内，假设我们要求在每个 \mathbf{x} 值处，预测被限制在委员会的任意成员给出的最小值和最大值之间，即

$$y_{\min}(\mathbf{x}) \leq y_{COM}(\mathbf{x}) \leq y_{\max}(\mathbf{x}) \quad (14.56)$$

证明，这个限制的一个充分必要条件是系数 α_m 满足

$$\alpha_m \geq 0, \quad \sum_{m=1}^M \alpha_m = 1 \quad (14.57)$$

(14.6) (*) 通过对误差函数 (14.23) 关于 α_m 求微分, 证明AdaBoost算法中的参数 α_m 使用公式 (14.17) 进行更新, 其中 ϵ_m 由公式 (14.16) 定义。

(14.7) (*) 通过对 (14.27) 给出的期望指数误差函数关于所有可能的函数 $y(\mathbf{x})$ 进行变分最小化, 证明最小的函数由公式 (14.28) 给出。

(14.8) (*) 证明, 通过AdaBoost算法最小化的指数误差函数 (14.20) 不对应于任何具有良好定义的概率模型。可以通过证明对应的条件概率分布 $p(t | \mathbf{x})$ 无法正确归一化的方式来证明这件事。

(14.9) (*) 证明对于形如 (14.21) 的可加性模型的平方和误差函数用提升方法进行顺序仅仅涉及到从前一个模型中根据残留误差 $t_n - f_{m-1}(\mathbf{x}_n)$, 调节每个新的基分类器。

(14.10) (*) 验证, 如果我们最小化训练数据集 $\{t_n\}$ 与单一的预测值 t 之间的平方和误差, 那么 t 的最优解由 $\{t_n\}$ 的均值给出。

(14.11) (**) 考虑一个由类别 \mathcal{C}_1 的 400 个数据点和类别 \mathcal{C}_2 的 400 个数据点组成的数据集。假设一个树模型 A 在第一个叶结点 (预测 \mathcal{C}_1) 将数据集划分为 $(300, 100)$, 在第二个叶结点 (预测 \mathcal{C}_2) 将数据集划分为 $(100, 300)$, 其中 (n, m) 表示 n 个数据点被分类为 \mathcal{C}_1 , m 个数据点被分类为 \mathcal{C}_2 。类似地, 假设第二个树模型 B 将他们划分为 $(200, 400)$ 和 $(200, 0)$ 。计算两棵树的分类错误率, 从而证明它们是相等的。类似地, 计算两棵树在交叉熵 (14.32) 的情形和基尼系数 (14.33) 的情形下的剪枝准则 (14.31), 证明树 B 的这两个量都小于树 A 。

(14.12) (**) 将 14.5.1 节的线性回归模型混合的结果推广到多个目标变量值 (由向量 \mathbf{t} 表示) 的情形。为了完成这一点, 使用 3.1.5 节的结果。

(14.13) (*) 验证线性回归模型的混合模型的完整数据似然函数为 (14.36)。

(14.14) (*) 使用拉格朗日乘数法 (附录 E) 证明, 使用最大似然 EM 训练的线性回归模型的混合模型的混合系数的 M 步骤重估计方程为 (14.38)。

(14.15) (*) 我们已经注意到, 如果我们在回归问题中使用平方损失函数, 那么对于一个新的输入向量, 对应目标变量的最优预测是预测分布的条件均值。证明, 14.5.1 节讨论的线性回归模型的混合模型的条件均值为每个分量分布的条件均值的线性组合。注意, 如果目标数据的条件分布是多峰的, 那么条件均值给出的预测会很差。

(14.16) (***) 将 14.5.2 节讨论的 logistic 回归混合模型推广到 $C \geq 2$ 个类别的 softmax 分类器的混合。写出通过最大似然方法确定模型参数的 EM 算法。

(14.17) (**) 考虑条件概率分布 $p(t | \mathbf{x})$ 的一个混合模型, 形式为

$$p(t | \mathbf{x}) = \sum_{k=1}^K \pi_k \psi_k(t | \mathbf{x}) \quad (14.58)$$

其中每个混合分量 $\psi_k(t | \mathbf{x})$ 本身是一个混合模型。证明, 这个两层的层次混合模型等价于一个传统的单层混合模型。现在假设这样的层次模型中, 两层中的混合系数都是 \mathbf{x} 的任意函数。再次证明这个层次模型等价于一个单层的模型, 其中混合系数与 \mathbf{x} 相关。最后, 考虑下面的情形: 层次混合模型的两层的混合系数被限制为线性分类 (logistic 或 softmax) 模型。证明, 一般情况下, 层次混合模型无法表示为混合系数是线性分类模型的单层混合模型。提示: 为了完成这件事, 构造一个反例即可。因此考虑两个分量的混合, 其中一个分量本身是两个分量的混合, 混合系数是线性 logistic 模型。证明它无法表示为一个单层的混合模型, 这个模型具有 3 个分量, 混合系数由线性 softmax 模型确定。

A 附录A. 数据集

在本附录中，我们简要地介绍了本书中用于描述某些算法所使用的数据集。对于这些数据集的文件格式的详细信息，以及数据文件本身，可以从本书的网站中得到：<http://research.microsoft.com/~cmbishop/PRML>。

A.1 手写数字

本书使用的手写数字来自MNIST数据集 (LeCun et al., 1998)。这个数据集的构建方式是修改NIST (the National Institute of Standards and Technology) 产生的一个大数据集的子集。这个数据集由一个包含60000个样本的训练集和一个包含10000个样本的测试集组成。数据集里的某些数据采集自Census Bureau的员工，其余的采集自高中生。此外，数据集构建人员仔细确保了测试样本的书写者与训练样本的书写者不同。

原始的NIST数据为二元（黑白）像素。为了创建MNIST，这些图像的大小被统一成 20×20 像素，并且保留了长宽比。为了在改变图像分辨率之后减少失真，最终的MNIST是灰度图。这些图像然后被居中在一个 28×28 的盒子中。图A.1给出了MNIST数字的例子。

使用一个简单的线性分类器，数字分类的错误率为12%。使用一个仔细设计的支持向量机，错误率降至0.56%。使用卷积神经网络 (LeCun et al., 1998)，错误率为0.4%。

A.2 石油流

这是一个由某个项目产生的人工合成的数据。这个项目用来测量北海石油传输管道中，不混溶的石油、水、天然气的比例。它依赖于双能量伽马密度 (dual-energy gamma densitometry) 原则。这个原则的思想是，如果一窄束伽马射线穿过管道，射线强度的衰减提供了管道中材料密度的信息。例如，射线通过石油之后的衰减会强于通过天然气之后的衰减。

简单地测量射线的衰减提供的信息并不充分，因为有两个自由度，对应着石油的比例和水的比例（天然气的比例是冗余的，因为三个比例相加一定等于1）。为了体现这一点，两个有着不同能量（或者说不同频率或波长）的伽马射线沿着同样的路径穿过管道，两条射线的衰减分别测量。由于不同材料的吸收属性关于能量的变化函数不同，两种能量衰减的测量提供了两条独立的信息。给定两种能量下，石油、水、天然气的吸收属性，计算沿着伽马射线路径方向上的平均油水比例就很容易了。

但是还有一个复杂之处与沿着管道的材料的运动相关。如果流速很慢，那么石油会漂浮在水上面，天然气位于石油上面。这被叫做薄片状 (laminar) 或者层次化 (stratified) 流配置，如图A.2所示。随着流速增加，会产生更复杂的石油、水、天然气的几何配置。为了描述这种数据集，开发者考虑了两种特定的理想化情形。在环状 (annular) 配置中，石油、水、天然气构成了同心圆柱，水在最外层，天然气在中心。在同质状 (homogeneous) 配置中，开发者假定石油、水、天然气紧密混合。这种配置可能出现在高流速的情形中。这些配置也在图A.2中给出。

我们已经看到，简单的双能量伽马射线能够测量沿着射线传播方向上的油水比例。但是我们感兴趣的是石油和水的体积比。使用多条双能量伽马射线，每条射线通过管道的不同区域，我们就可以达到这个目的。对于这个特定的数据集，有六条射线，它们的空间分布如图A.3所示。因此，一个简单的观测由一个12维的向量表示，这个向量包含每条射线沿着路径方向的油水比例。但是，我们感兴趣的是管道中三种物质的整体体积比例。这很像经典的断层显像重建问题，用于诸如医学图像等领域。在断层显像重建技术中，通过一系列的一维均值，我们可以重建出一个二维的分布。在我们的问题中，线度量的数量要远远小于断层显像重建的应用。另一方面，我们的问题中，几何配置的种类也很有限，因此通过密度数据，我们可以在一个合理的精度下预测配置和各个物质所占的比例。

出于安全考虑，伽马射线的强度相对较弱，因此为了准确测量强度的衰减，测量的射线强度在一个具体的时间区间内积分。对于有限的积分时间，测量的射线强度会有随机的涨落。这是因为伽马射线是由被称为量子的离散能量包组成的。在实际应用中，积分时间的选择要在降低噪声（需要较长的积分时间）和检测流的时序变化（需要较短的积分时间）之间进行折中。在生成石油流数据集时，两束伽马射线的能量已知，在这种能量下石油、水、天然气的能量也已知，积分时间选择被设定为一个特定的时间（10秒），这是实际应用中的典型设置。



图 A.1: MNIST 手写数据集的 100 个样本，从训练集中随机选择。

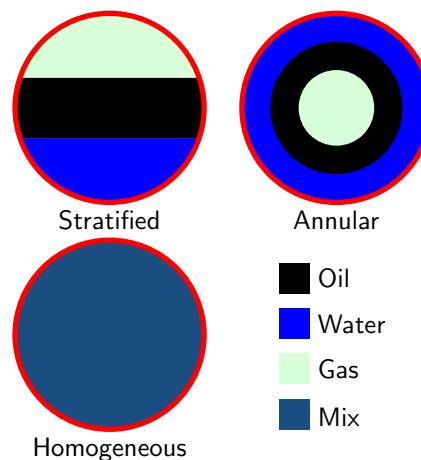


图 A.2: 石油、水、天然气的三种几何配置，用来生成石油流数据集。对于每种配置，三种成分的比例可以改变。

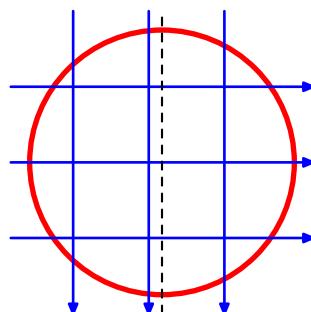


图 A.3: 管道的横切面，表示六个射线束的配置，每个射线束对应着一个双能量伽马射线密度计。注意，垂直射线束关于中心轴（虚线表示）不是对称的。



图 A.4: 黄石国家公园的老忠实间歇喷泉。www.brucegourley.com

数据集中的每个数据点独立地使用下面的步骤生成:

1. 等概率地随机选择三种配置中的一种。
2. 在 $(0, 1)$ 上的均匀分布上, 随机选择三个数 f_1, f_2, f_3 , 并且定义:

$$f_{oil} = \frac{f_1}{f_1 + f_2 + f_3}, \quad f_{water} = \frac{f_2}{f_1 + f_2 + f_3} \quad (\text{A.1})$$

这里平等地对待三种物质, 并且确保了体积分数的和等于1。

3. 对于六条射线中的每一个, 计算在给定的配置下通过石油和水的有效路径长度。
4. 根据已知的射线强度和积分时间, 使用泊松分布来扰乱路径长度, 从而模拟量子统计学的效应。

数据集里的每个点包括12个路径长度的测量、石油和水的比例, 以及一个描述配置的二元标签。数据集被切分成训练集、验证集和测试集, 每个都由1000个独立的数据点构成。数据格式的细节可以从本书的网站中得到。

在Bishop and James (1993)中, 根据12维测量的向量, 统计机器学习技术被用来预测体积分数, 以及图A.2所示的几何配置。12维观测向量也可以用在测试数据可视化算法当中。

这个数据集有着丰富的并且很有趣的结构。对于任意一个给定的配置, 有两个自由度, 分别对应于油和水的比例, 因此对于无限的积分时间, 数据将会位于一个局部的二维流形中。对于有限的积分时间, 各个数据点会被量子噪声干扰, 脱离流形。在同质状配置中, 石油和水中的路径长度与石油和水的比例线性相关, 因此数据点位于线性流形中。对于环状配置, 物质比例和路径长度的关系是非线性的, 因此流形就是非线性的。在薄片状配置中, 配置甚至更加复杂, 因为物质比例的微小的改变能够引起某个水平分界线移过某条伽马射线, 这会导致12维观测空间中的非连续跳变。这样, 薄片状配置的二维非线性流形就破裂为10个不同的碎片。还要注意, 对于不同的配置, 某些流形会在特定的点处交汇。例如, 如果管道中充满了石油, 那么它对应着薄片状、环状、同质状配置的特殊情况。

A.3 老忠实间歇喷泉

老忠实间歇喷泉, 如图A.4所示, 是美国怀俄明州黄石国家公园中的一个间歇喷泉, 也是一个著名的旅游景点。它的名字来源于它的喷发很有规律。

数据集由272次观测组成, 每次观测表示一次喷发, 包含两个变量, 分别对应喷发的持续时间(用分钟表示)和距离下次喷发的时间(也用分钟表示)。图A.5给出了距离下次喷发的时间关于喷发持续时间的图像。可以看到, 距离下次喷发的时间变化范围很大, 但是关于本次喷发持续时间的知识能够让我们进行更加准确的预测。需要注意的是, 关于老忠实间歇喷泉的喷发, 存在几个其他的数据集。

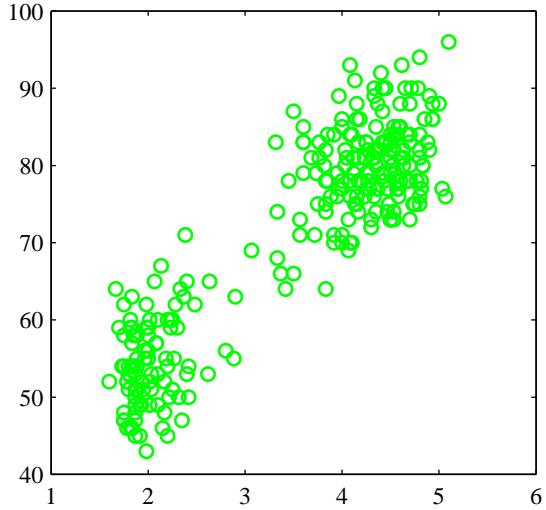


图 A.5: 对于老忠实间歇喷泉数据集,两次喷发的时间间隔 (竖直轴) 与喷发持续时间 (水平轴) 的关系。

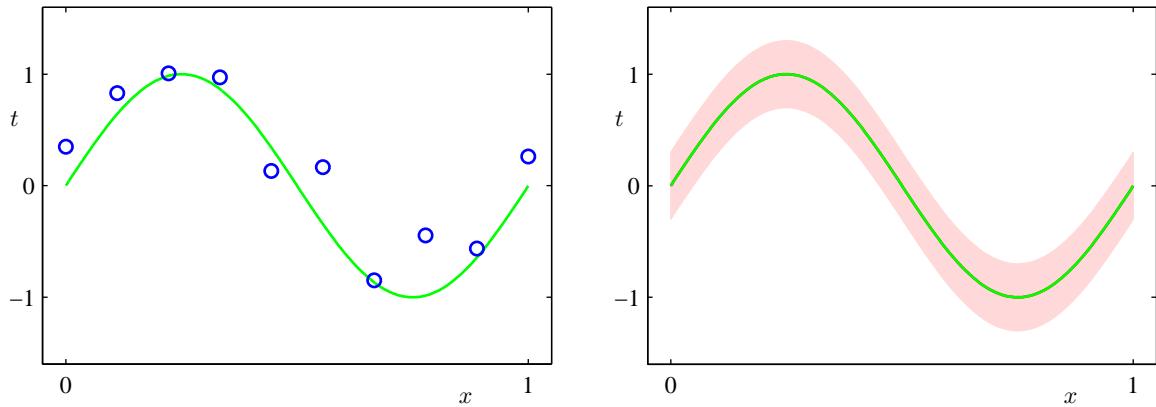


图 A.6: 左图给出了人工生成的回归数据,以及用于生成数据点的正弦函数。右图给出了生成标签的真实的条件概率分布 $p(t | x)$,其中绿色曲线表示均值,阴影区域表示均值两侧一个标准差的位置。

A.4 人工生成数据

在全书中,我们使用了两个简单的人工生成的数据来说明许多算法。第一个是回归问题,依据图A.6所示的正弦函数。输入变量 $\{x_n\}$ 在 $(0, 1)$ 内按照均匀分布生成,对应的目标值 $\{t_n\}$ 的获得方式为:首先计算函数 $\sin(2\pi x)$ 的对应值,然后加上一个满足标准差为0.3的高斯分布的噪声。本书使用了这个数据集的各种形式,每种形式的数据点数量都不同。

第二个数据集是一个分类问题,有两个类别,先验概率相同,如图A.7所示。蓝色的类别由一个高斯分布生成,而红色的类别由两个混合的高斯分布生成。由于我们知道类先验概率和类条件概率密度,因此很容易估计并画出真实的后验概率以及最小错误分类率决策边界,如图A.7所示。

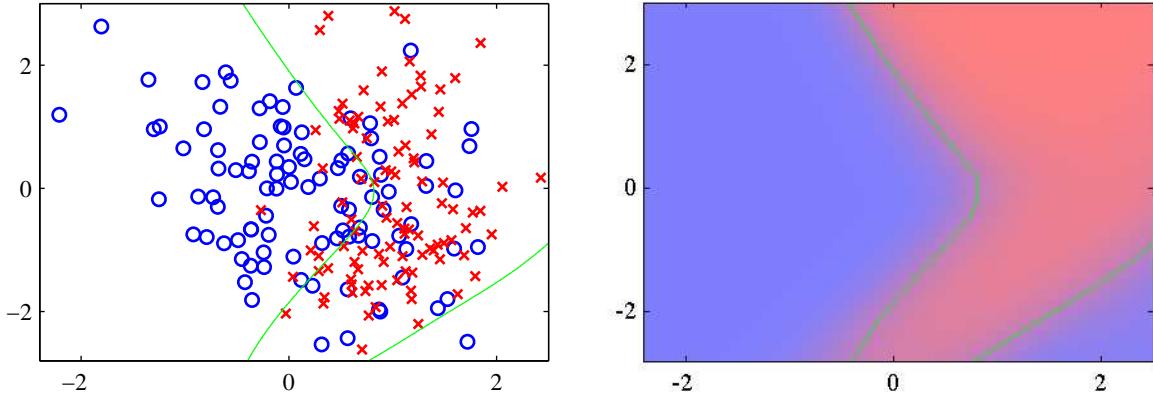


图 A.7: 左图给出了人工生成的分类数据集，两个类别用红色和蓝色表示。右图是对应的真实后验概率，颜色从纯的红色（表示属于红色类别的概率为1）变化到纯的蓝色（表示属于蓝色类别的概率为0）。由于这些概率是已知的，因此最小化误分类概率的最优决策边界（对于属于每个类别的概率等于0.5的轮廓线）可以计算，用绿色曲线表示。决策边界也在左图中给出。

B 附录B. 概率分布

在本附录中，我们总结了一些广泛使用的概率分布的性质。对于每个概率分布，我们列出了一些关键的统计性质，例如期望 $\mathbb{E}[x]$ 、方差（或者协方差），众数，熵 $H[x]$ 。所有这些分布都是指数族的成员，被广泛用作更高级的概率模型的基本模块。

B.1 伯努利分布

这是单一二元变量 $x \in \{0, 1\}$ 的分布，例如，抛硬币的结果。它由一个连续参数 $\mu \in [0, 1]$ 控制，这个参数表示 $x = 1$ 的概率。

$$\text{Bern}(x | \mu) = \mu^x(1 - \mu)^{1-x} \quad (\text{B.1})$$

$$\mathbb{E}[x] = \mu \quad (\text{B.2})$$

$$\text{var}[x] = \mu(1 - \mu) \quad (\text{B.3})$$

$$\text{mode}[x] = \begin{cases} 1 & \text{如果 } \mu \geq 0.5 \\ 0 & \text{否则} \end{cases} \quad (\text{B.4})$$

$$H[x] = -\mu \ln \mu - (1 - \mu) \ln(1 - \mu) \quad (\text{B.5})$$

伯努利分布是二项分布对于单一观测的特殊情况。它对于 μ 的共轭先验是Beta分布。

B.2 Beta分布

这是连续变量 $\mu \in [0, 1]$ 的分布，经常用于表示某些二元事件的概率。它有两个参数 a 和 b 。为了保证分布能够归一化，我们要求 $a > 0$ 并且 $b > 0$ 。

$$\text{Beta}(\mu | a, b) = \frac{\Gamma(a + b)}{\Gamma(a)\Gamma(b)} \mu^{a-1} (1 - \mu)^{b-1} \quad (\text{B.6})$$

$$\mathbb{E}[\mu] = \frac{a}{a + b} \quad (\text{B.7})$$

$$\text{var}[\mu] = \frac{ab}{(a + b)^2(a + b + 1)} \quad (\text{B.8})$$

$$\text{mode}[\mu] = \frac{a - 1}{a + b - 2} \quad (\text{B.9})$$

Beta分布是伯努利分布的共轭先验，其中 a 和 b 可以分别表示为 $x = 1$ 和 $x = 0$ 的观测的有效先验数量。如果 $a \geq 1$ 且 $b \geq 1$ ，那么它的概率密度是有限值，否则在 $\mu = 0$ 和（或） $\mu = 1$ 处会有奇异值。对于 $a = b = 1$ 的情形，它就简化成了均匀分布。Beta分布是 K 状态狄利克雷分布在 $K = 2$ 时的特殊情形。

B.3 二项分布

二项分布给出了来自伯努利分布的 N 个样本中观察到 m 次 $x = 1$ 的概率。伯努利分布中，观察到 $x = 1$ 的概率是 $\mu \in [0, 1]$ 。

$$\text{Bin}(m | N, \mu) = \binom{N}{m} \mu^m (1 - \mu)^{N-m} \quad (\text{B.10})$$

$$\mathbb{E}[m] = N\mu \quad (\text{B.11})$$

$$\text{var}[m] = N\mu(1 - \mu) \quad (\text{B.12})$$

$$\text{mode}[m] = \lfloor (N + 1)\mu \rfloor \quad (\text{B.13})$$

其中， $\lfloor (N + 1)\mu \rfloor$ 表示不超过 $(N + 1)\mu$ 的最大整数。此外

$$\binom{N}{m} = \frac{N!}{m!(N - m)!} \quad (\text{B.14})$$

表示从 N 个完全相同的物体中选择 m 个物体的总方案数量。这里 $m!$ 表示乘积 $m \times (m - 1) \times \dots \times 2 \times 1$ 。二项分布中 $N = 1$ 这一特殊情形被称为伯努利分布，对于大的 N 值，二项分布近似于高斯分布。 μ 的共轭先验是Beta分布。

B.4 狄利克雷分布

狄利克雷分布是 K 个随机变量 $0 \leq \mu_k \leq 1$ 的多变量分布，其中 $k = 1, \dots, K$ ，并且满足下面的限制

$$0 \leq \mu_k \leq 1, \quad \sum_{k=1}^K \mu_k = 1 \quad (\text{B.15})$$

记 $\boldsymbol{\mu} = (\mu_1, \dots, \mu_K)^T$, $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_K)^T$, 我们有

$$\text{Dir}(\boldsymbol{\mu} | \boldsymbol{\alpha}) = C(\boldsymbol{\alpha}) \prod_{k=1}^K \mu_k^{\alpha_k - 1} \quad (\text{B.16})$$

$$\mathbb{E}[\mu_k] = \frac{\alpha_k}{\hat{\alpha}} \quad (\text{B.17})$$

$$\text{var}[\mu_k] = \frac{\alpha_k(\hat{\alpha} - \alpha_k)}{\hat{\alpha}^2(\hat{\alpha} + 1)} \quad (\text{B.18})$$

$$\text{cov}[\mu_j \mu_k] = -\frac{\alpha_j \alpha_k}{\hat{\alpha}^2(\hat{\alpha} + 1)} \quad (\text{B.19})$$

$$\text{mode}[\mu_k] = \frac{\alpha_k - 1}{\hat{\alpha} - K} \quad (\text{B.20})$$

$$\mathbb{E}[\ln \mu_k] = \psi(\alpha_k) - \psi(\hat{\alpha}) \quad (\text{B.21})$$

$$H[\boldsymbol{\mu}] = - \sum_{k=1}^K (\alpha_k - 1) \{ \psi(\alpha_k) - \psi(\hat{\alpha}) \} - \ln C(\boldsymbol{\alpha}) \quad (\text{B.22})$$

其中

$$C(\boldsymbol{\alpha}) = \frac{\Gamma(\hat{\alpha})}{\Gamma(\alpha_1) \cdots \Gamma(\alpha_K)} \quad (\text{B.23})$$

并且

$$\hat{\alpha} = \sum_{k=1}^K \alpha_k \quad (\text{B.24})$$

这里

$$\psi(a) \equiv \frac{d}{da} \ln \Gamma(a) \quad (\text{B.25})$$

被称为digamma函数 (Abramowitz and Stegun, 1965)。为了保证概率归一化，参数 α_k 满足限制 $\alpha_k > 0$ 。

狄利克雷分布是多项式分布的共轭先验，是Beta分布的推广。这种情况下，参数 α_k 是 K 维二元观测向量 x 对应值的有效观测数量。和Beta分布相同，如果对于所有的 k 都有 $\alpha_k \geq 1$ ，那么狄利克雷分布在空间中所有位置的密度均为有限值。

B.5 Gamma分布

Gamma分布是正随机变量 $\tau > 0$ 的概率分布，参数为 a 和 b ，满足限制 $a > 0$ 和 $b > 0$ ，保证概率分布是归一化的。

$$\text{Gam}(\tau | a, b) = \frac{1}{\Gamma(a)} b^a \tau^{a-1} e^{-b\tau} \quad (\text{B.26})$$

$$\mathbb{E}[\tau] = \frac{a}{b} \quad (\text{B.27})$$

$$\text{var}[\tau] = \frac{a}{b^2} \quad (\text{B.28})$$

$$\text{mode}[\tau] = \frac{a-1}{b} \quad \text{当 } a \geq 1 \text{ 时成立} \quad (\text{B.29})$$

$$\mathbb{E}[\ln \tau] = \psi(a) - \ln b \quad (\text{B.30})$$

$$H[\tau] = \ln \Gamma(a) - (a-1)\psi(a) - \ln b + a \quad (\text{B.31})$$

其中， $\psi(\cdot)$ 是公式 (B.25) 定义的digamma函数。Gamma分布式单变量高斯分布的精度（方差的倒数）的共轭先验。当 $a \geq 1$ 时，概率密度处处为有限值， $a = 1$ 这一特殊情况被称为指数分布 (exponential distribution)。

B.6 高斯分布

高斯分布是连续变量中最广泛使用的概率分布。它也被称为正态分布 (normal distribution)。在一元变量 $x \in (-\infty, \infty)$ 的情况下，它由两个参数控制：均值 $\mu \in (-\infty, \infty)$ 和方差 $\sigma^2 > 0$ 。

$$\mathcal{N}(x | \mu, \sigma^2) = \frac{1}{(2\pi\sigma^2)^{\frac{1}{2}}} \exp\left(-\frac{1}{2\sigma^2}(x-\mu)^2\right) \quad (\text{B.32})$$

$$\mathbb{E}[x] = \mu \quad (\text{B.33})$$

$$\text{var}[x] = \sigma^2 \quad (\text{B.34})$$

$$\text{mode}[x] = \mu \quad (\text{B.35})$$

$$H[x] = \frac{1}{2} \ln \sigma^2 + \frac{1}{2}(1 + \ln(2\pi)) \quad (\text{B.36})$$

方差的倒数 $\tau = \frac{1}{\sigma^2}$ 被称为精度，方差的平方根 σ 被称为标准差。 μ 的共轭先验是高斯分布， τ 的共轭先验是Gamma分布。如果 μ 和 τ 都是未知的，那么它们的联合共轭先验是高斯-Gamma分布。

对于一个 D 维向量 \boldsymbol{x} , 高斯分布的参数是一个 D 维均值向量 $\boldsymbol{\mu}$ 和一个 $D \times D$ 的协方差矩阵 $\boldsymbol{\Sigma}$ 。协方差矩阵一定是对称的、正定的。

$$\mathcal{N}(\boldsymbol{x} | \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{\frac{D}{2}}} \frac{1}{|\boldsymbol{\Sigma}|^{\frac{1}{2}}} \exp \left\{ -\frac{1}{2} (\boldsymbol{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{x} - \boldsymbol{\mu}) \right\} \quad (\text{B.37})$$

$$\mathbb{E}[\boldsymbol{x}] = \boldsymbol{\mu} \quad (\text{B.38})$$

$$\text{cov}[\boldsymbol{x}] = \boldsymbol{\Sigma} \quad (\text{B.39})$$

$$\text{mode}[\boldsymbol{x}] = \boldsymbol{\mu} \quad (\text{B.40})$$

$$H[\boldsymbol{x}] = \frac{1}{2} \ln |\boldsymbol{\Sigma}| + \frac{D}{2} (1 + \ln(2\pi)) \quad (\text{B.41})$$

协方差矩阵的逆矩阵 $\boldsymbol{\Lambda} = \boldsymbol{\Sigma}^{-1}$ 叫做精度矩阵, 也是对称的、正定的。根据中心极限定理, 随机变量的平均值趋近于高斯分布, 并且两个高斯变量之和仍然是高斯。给定方差(或者协方差), 高斯分布是最大化熵值的分布。高斯随机变量的任意线性组合仍然是高斯分布。多元高斯的变量关于变量的一个子集的边缘分布仍然是高斯分布, 类似地, 条件分布也是高斯分布。 $\boldsymbol{\mu}$ 的共轭先验仍然是高斯分布, $\boldsymbol{\Lambda}$ 的共轭先验是一个Wishart分布, $(\boldsymbol{\mu}, \boldsymbol{\Lambda})$ 的共轭先验是高斯-Wishart分布。

如果我们有一个 \boldsymbol{x} 的边缘高斯分布, 以及在给定 \boldsymbol{x} 的条件下 \boldsymbol{y} 的条件高斯分布, 形式如下

$$p(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x} | \boldsymbol{\mu}, \boldsymbol{\Lambda}^{-1}) \quad (\text{B.42})$$

$$p(\boldsymbol{y} | \boldsymbol{x}) = \mathcal{N}(\boldsymbol{y} | \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}, \boldsymbol{L}^{-1}) \quad (\text{B.43})$$

那么 \boldsymbol{y} 的边缘分布, 以及给定 \boldsymbol{y} 的条件下 \boldsymbol{x} 的条件分布分别为

$$p(\boldsymbol{y}) = \mathcal{N}(\boldsymbol{y} | \boldsymbol{A}\boldsymbol{\mu} + \boldsymbol{b}, \boldsymbol{L}^{-1} + \boldsymbol{A}\boldsymbol{\Lambda}^{-1}\boldsymbol{A}^T) \quad (\text{B.44})$$

$$p(\boldsymbol{x} | \boldsymbol{y}) = \mathcal{N}(\boldsymbol{x} | \boldsymbol{\Sigma}\{\boldsymbol{A}^T\boldsymbol{L}(\boldsymbol{y} - \boldsymbol{b}) + \boldsymbol{\Lambda}\boldsymbol{\mu}\}, \boldsymbol{\Sigma}) \quad (\text{B.45})$$

其中

$$\boldsymbol{\Sigma} = (\boldsymbol{\Lambda} + \boldsymbol{A}^T\boldsymbol{L}\boldsymbol{A})^{-1} \quad (\text{B.46})$$

如果我们有一个联合高斯分布 $\mathcal{N}(\boldsymbol{x} | \boldsymbol{\mu}, \boldsymbol{\Sigma})$, 且 $\boldsymbol{\Lambda} \equiv \boldsymbol{\Sigma}^{-1}$, 并且我们定义下面的划分

$$\boldsymbol{x} = \begin{pmatrix} \boldsymbol{x}_a \\ \boldsymbol{x}_b \end{pmatrix}, \quad \boldsymbol{\mu} = \begin{pmatrix} \boldsymbol{\mu}_a \\ \boldsymbol{\mu}_b \end{pmatrix} \quad (\text{B.47})$$

$$\boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma}_{aa} & \boldsymbol{\Sigma}_{ab} \\ \boldsymbol{\Sigma}_{ba} & \boldsymbol{\Sigma}_{bb} \end{pmatrix}, \quad \boldsymbol{\Lambda} = \begin{pmatrix} \boldsymbol{\Lambda}_{aa} & \boldsymbol{\Lambda}_{ab} \\ \boldsymbol{\Lambda}_{ba} & \boldsymbol{\Lambda}_{bb} \end{pmatrix} \quad (\text{B.48})$$

那么条件概率分布 $p(\boldsymbol{x}_a | \boldsymbol{x}_b)$ 为

$$p(\boldsymbol{x}_a | \boldsymbol{x}_b) = \mathcal{N}(\boldsymbol{x}_a | \boldsymbol{\mu}_{a|b}, \boldsymbol{\Lambda}_{aa}^{-1}) \quad (\text{B.49})$$

$$\boldsymbol{\mu}_{a|b} = \boldsymbol{\mu}_a - \boldsymbol{\Lambda}_{aa}^{-1} \boldsymbol{\Lambda}_{ab} (\boldsymbol{x}_b - \boldsymbol{\mu}_b) \quad (\text{B.50})$$

边缘分布 $p(\boldsymbol{x}_a)$ 为

$$p(\boldsymbol{x}_a) = \mathcal{N}(\boldsymbol{x}_a | \boldsymbol{\mu}_a, \boldsymbol{\Sigma}_{aa}) \quad (\text{B.51})$$

B.7 高斯-Gamma分布

这是一元高斯分布 $\mathcal{N}(x | \mu, \lambda^{-1})$ 的共轭先验, 其中均值 μ 和精度 λ 均未知。这个分布也被称为正态-Gamma分布。它是精度正比于 λ 的 μ 的高斯分布与 λ 的Gamma分布的乘积。

$$p(\mu, \lambda | \mu_0, \beta, a, b) = \mathcal{N}(\mu | \mu_0, (\beta\lambda)^{-1}) \text{Gam}(\lambda | a, b) \quad (\text{B.52})$$

B.8 高斯-Wishart分布

这是多元高斯分布 $\mathcal{N}(\boldsymbol{x} \mid \boldsymbol{\mu}, \boldsymbol{\Lambda})$ 的共轭先验，其中均值 $\boldsymbol{\mu}$ 和精度 $\boldsymbol{\Lambda}$ 均未知。这个分布也被称为正态-Wishart分布。它是精度正比于 $\boldsymbol{\Lambda}$ 的 $\boldsymbol{\mu}$ 的高斯分布与 $\boldsymbol{\Lambda}$ 的Wishart分布的乘积。

$$p(\boldsymbol{\mu}, \boldsymbol{\Lambda} \mid \boldsymbol{\mu}_0, \beta, \mathbf{W}, \nu) = \mathcal{N}(\boldsymbol{\mu} \mid \boldsymbol{\mu}_0, (\beta \boldsymbol{\Lambda})^{-1}) \mathcal{W}(\boldsymbol{\Lambda} \mid \mathbf{W}, \nu) \quad (\text{B.53})$$

对于标量 x 的情况，它等价于高斯-Gamma分布。

B.9 多项式分布

如果我们把伯努利分布推广到 K 维二元变量 \boldsymbol{x} ，分量 $x_k \in \{0, 1\}$ 且 $\sum_k x_k = 1$ ，那么我们由下面的离散分布

$$p(\boldsymbol{x}) = \prod_{k=1}^K \mu_k^{x_k} \quad (\text{B.54})$$

$$\mathbb{E}[x_k] = \mu_k \quad (\text{B.55})$$

$$\text{var}[x_k] = \mu_k(1 - \mu_k) \quad (\text{B.56})$$

$$\text{cov}[x_j x_k] = -\mu_j \mu_k, \quad j \neq k \quad (\text{B.57})$$

$$H[\boldsymbol{x}] = -\sum_{k=1}^K \mu_k \ln \mu_k \quad (\text{B.58})$$

由于 $p(x_k = 1) = \mu_k$ ，因此参数必须满足 $0 \leq \mu_k \leq 1$ 以及 $\sum_k \mu_k = 1$ 。

多项式分布式二项分布对于多元变量的推广，给出了一个具有 K 个状态的离散变量在总计 N 次观测中处于状态 k 的次数 m_k 的分布。

$$\text{Mult}(m_1, m_2, \dots, m_K \mid \boldsymbol{\mu}, N) = \binom{N}{m_1 m_2 \dots m_K} \prod_{k=1}^K \mu_k^{m_k} \quad (\text{B.59})$$

$$\mathbb{E}[m_k] = N\mu_k \quad (\text{B.60})$$

$$\text{var}[m_k] = N\mu_k(1 - \mu_k) \quad (\text{B.61})$$

$$\text{cov}[m_j m_k] = -N\mu_j \mu_k, \quad j \neq k \quad (\text{B.62})$$

其中 $\boldsymbol{\mu} = (\mu_1, \dots, \mu_K)^T$ ，并且

$$\binom{N}{m_1 m_2 \dots m_M} = \frac{N!}{m_1! \dots m_K!} \quad (\text{B.63})$$

给出了把 N 个相同的物体中的 m_k 个放到箱子 k 中的方案总数，其中 $k = 1, \dots, K$ 。 μ_k 的值给出了随机变量处于 k 状态的概率，因此必须满足 $0 \leq \mu_k \leq 1$ 且 $\sum_k \mu_k = 1$ 。参数 $\{\mu_k\}$ 的共轭先验是狄利克雷分布。

B.10 正态分布

正态分布是高斯分布的另一个名字。本书中，我们始终使用高斯分布这个术语，虽然我们遵循惯例，用 \mathcal{N} 来表示这个分布。为了记号的统一，我们把正态-Gamma分布称为高斯-Gamma分布，把正态-Wishart分布称为高斯-Wishart分布。

B.11 学生t分布

这个分布由William Gosset在1908年提出，但是他的老板Guiness Breweries让他用笔名发表，因此它选择了“学生”这个笔名。在一元变量的形式下，学生t分布可以通过下列方式获得：拿出一元高斯分布的精度的共轭先验，然后把精度变量积分出来。因此这个分布可以看成无限多个有着相同均值不同方差的高斯分布的混合。

$$\text{St}(x | \mu, \lambda, \nu) = \frac{\Gamma(\frac{\nu}{2} + \frac{1}{2})}{\Gamma(\frac{\nu}{2})} \left(\frac{\lambda}{\pi\nu} \right)^{\frac{1}{2}} \left[1 + \frac{\lambda(x - \mu)^2}{\nu} \right]^{-\frac{\nu}{2} - \frac{1}{2}} \quad (\text{B.64})$$

$$\mathbb{E}[x] = \mu \text{ 当 } \nu > 1 \text{ 时成立} \quad (\text{B.65})$$

$$\text{var}[x] = \frac{1}{\lambda} \frac{\nu}{\nu - 2} \text{ 当 } \nu > 2 \text{ 时成立} \quad (\text{B.66})$$

$$\text{mode}[x] = \mu \quad (\text{B.67})$$

这里 $\nu > 0$ 被称为分布的自由度数量。 $\nu = 1$ 的特殊情况被叫做柯西分布 (Cauchy distribution)。

对于一个 D 维变量 \mathbf{x} ，学生t分布是将多元高斯的精度矩阵关于共轭Wishart先验积分的结果，形式为

$$\text{St}(\mathbf{x} | \boldsymbol{\mu}, \boldsymbol{\Lambda}, \nu) = \frac{\Gamma(\frac{\nu}{2} + \frac{D}{2})}{\Gamma(\frac{\nu}{2})} \frac{|\boldsymbol{\Lambda}|^{\frac{1}{2}}}{(\nu\pi)^{\frac{D}{2}}} \left[1 + \frac{\Delta^2}{\nu} \right]^{-\frac{\nu}{2} - \frac{D}{2}} \quad (\text{B.68})$$

$$\mathbb{E}[\mathbf{x}] = \boldsymbol{\mu} \text{ 当 } \nu > 1 \text{ 时成立} \quad (\text{B.69})$$

$$\text{cov}[\mathbf{x}] = \frac{\nu}{\nu - 2} \boldsymbol{\Lambda}^{-1} \text{ 当 } \nu > 2 \text{ 时成立} \quad (\text{B.70})$$

$$\text{mode}[\mathbf{x}] = \boldsymbol{\mu} \quad (\text{B.71})$$

其中， Δ^2 是平方马氏距离，定义为

$$\Delta^2 = (\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Lambda} (\mathbf{x} - \boldsymbol{\mu}) \quad (\text{B.72})$$

在极限 $\nu \rightarrow \infty$ 的情况下，t分布简化为均值 $\boldsymbol{\mu}$ ，精度 $\boldsymbol{\Lambda}$ 的高斯分布。学生t分布提供了对高斯分布泛化的一种形式，这种分布的最大似然参数值对离群点比较鲁棒。

B.12 均匀分布

这是连续变量 x 的一种简单分布。 x 定义在有限区间 $x \in [a, b]$ ，且 $b > a$ 。

$$U(x | a, b) = \frac{1}{b - a} \quad (\text{B.73})$$

$$\mathbb{E}[x] = \frac{b + a}{2} \quad (\text{B.74})$$

$$\text{var}[x] = \frac{(b - a)^2}{12} \quad (\text{B.75})$$

$$H[x] = \ln(b - a) \quad (\text{B.76})$$

如果 x 服从均匀分布 $U(x | 0, 1)$ ，那么 $a + (b - a)x$ 服从均匀分布 $U(x | a, b)$ 。

B.13 Von Mises分布

Von Mises分布，也被称为环形正态分布或者环形高斯分布，是一元变量 $\theta \in [0, 2\pi)$ 的类似高斯的周期分布。

$$p(\theta | \theta_0, m) = \frac{1}{2\pi I_0(m)} \exp\{m \cos(\theta - \theta_0)\} \quad (\text{B.77})$$

其中 $I_0(m)$ 是零阶第一类Bessel函数。分布的周期是 2π ，因此对于所有的 θ 都有 $p(\theta + 2\pi) = p(\theta)$ 。在表述这个分布时需要小心，因为简单的期望都要取决于变量 θ 的原点的（任意）选择。参数 θ_0 类似于一元高斯分布的均值。参数 $m > 0$ ，被称为concentration参数，类似于高斯分布的精度（方差的倒数）。对于大的 m 值，Von Mises分布近似于以 θ_0 为中心的高斯分布。

B.14 Wishart分布

Wishart分布是多元高斯的精度矩阵的共轭先验。

$$\mathcal{W}(\boldsymbol{\Lambda} \mid \mathbf{W}, \nu) = B(\mathbf{W}, \nu) |\boldsymbol{\Lambda}|^{\frac{\nu-D-1}{2}} \exp\left(-\frac{1}{2} \text{Tr}(\mathbf{W}^{-1} \boldsymbol{\Lambda})\right) \quad (\text{B.78})$$

其中

$$B(\mathbf{W}, \nu) \equiv |\mathbf{W}|^{-\frac{\nu}{2}} \left(2^{\frac{\nu D}{2}} \pi^{\frac{D(D-1)}{4}} \prod_{i=1}^D \Gamma\left(\frac{\nu+1-i}{2}\right) \right)^{-1} \quad (\text{B.79})$$

$$\mathbb{E}[\boldsymbol{\Lambda}] = \nu \mathbf{W} \quad (\text{B.80})$$

$$\mathbb{E}[\ln |\boldsymbol{\Lambda}|] = \sum_{i=1}^D \psi\left(\frac{\nu+1-i}{2}\right) + D \ln 2 + \ln |\mathbf{W}| \quad (\text{B.81})$$

$$H[\boldsymbol{\Lambda}] = -\ln B(\mathbf{W}, \nu) - \frac{\nu-D-1}{2} \mathbb{E}[\ln |\boldsymbol{\Lambda}|] + \frac{\nu D}{2} \quad (\text{B.82})$$

其中， \mathbf{W} 是一个 $D \times D$ 对称正定矩阵， $\psi(\cdot)$ 是公式 (B.25) 定义的digamma函数。参数 ν 被称为分布的自由度的数量 (number of degrees of freedom)，满足限制 $\nu > D - 1$ ，以保证归一化因子中的Gamma函数有着良好的定义。在一维情形下，Wishart分布就变成了公式 (B.26) 定义的Gamma分布 $\text{Gam}(\lambda \mid a, b)$ ，参数为 $a = \frac{\nu}{2}$ ， $b = \frac{1}{2W}$ 。

C 附录C. 矩阵的性质

在这个附录中，我们汇总了一些涉及到矩阵和行列式的有用的性质。这里不打算写成一个入门性教程，并且我们假定读者已经熟悉了基本的线性代数。对于某些结论，我们给出证明。对于更加复杂的结论，我们留给感兴趣的读者参考标准的教科书。在所有情况下，我们都假定逆矩阵存在，并且矩阵的维度能够让公式正确定义。线性代数的一个可理解的讨论可以参考Golub and Van Loan (1996)。Lütkepohl (1996) 汇编了矩阵的一些扩展性质。Magnus and Neudecker (1999) 讨论了矩阵的导数。

C.1 矩阵的基本性质

矩阵 \mathbf{A} 的第*i*行第*j*列的元素为 A_{ij} 。我们用 \mathbf{I}_N 表示 $N \times N$ 的单位矩阵。在没有歧义的情况下，我们简单地记作 \mathbf{I} 。转置矩阵 \mathbf{A}^T 的元素为 $(\mathbf{A}^T)_{ij} = A_{ji}$ 。根据转置的定义，我们有

$$(\mathbf{AB})^T = \mathbf{B}^T \mathbf{A}^T \quad (\text{C.1})$$

写出元素的下标，即可得出上面的结果。 \mathbf{A} 的逆矩阵，记作 \mathbf{A}^{-1} ，满足

$$\mathbf{AA}^{-1} = \mathbf{A}^{-1} \mathbf{A} = \mathbf{I} \quad (\text{C.2})$$

由于 $\mathbf{ABB}^{-1}\mathbf{A}^{-1} = \mathbf{I}$ ，我们有

$$(\mathbf{AB})^{-1} = \mathbf{B}^{-1} \mathbf{A}^{-1} \quad (\text{C.3})$$

我们还有

$$(\mathbf{A}^T)^{-1} = (\mathbf{A}^{-1})^T \quad (\text{C.4})$$

将公式 (C.2) 取转置，然后应用公式 (C.1)，这个公式可以很容易证明。

关于矩阵的逆矩阵，下面这个恒等式很有用

$$(\mathbf{P}^{-1} + \mathbf{B}^T \mathbf{R}^{-1} \mathbf{B})^{-1} \mathbf{B}^T \mathbf{R}^{-1} = \mathbf{P} \mathbf{B}^T (\mathbf{B} \mathbf{P} \mathbf{B}^T + \mathbf{R})^{-1} \quad (\text{C.5})$$

两侧同时右乘 $(\mathbf{B} \mathbf{P} \mathbf{B}^T + \mathbf{R})$ ，很容易证明上式的正确性。假设 \mathbf{P} 的维度为 $N \times N$ ，而 \mathbf{R} 的维度为 $M \times M$ ，从而 \mathbf{B} 的维度为 $M \times N$ 。这样，如果 $M \ll N$ ，那么估计公式 (C.5) 的右侧所花费的代价就远远小于估计左侧的代价。经常出现的一种情况是

$$(\mathbf{I} + \mathbf{AB})^{-1} \mathbf{A} = \mathbf{A} (\mathbf{I} + \mathbf{BA})^{-1} \quad (\text{C.6})$$

另一个与矩阵的逆矩阵相关的有用的恒等式为

$$(\mathbf{A} + \mathbf{BD}^{-1}\mathbf{C})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1} \mathbf{B} (\mathbf{D} + \mathbf{CA}^{-1}\mathbf{B})^{-1} \mathbf{C} \mathbf{A}^{-1} \quad (\text{C.7})$$

这被称为Woodbury恒等式。将两侧同时乘以 $(\mathbf{A} + \mathbf{BD}^{-1}\mathbf{C})$ 即可证明。例如，假设 \mathbf{A} 是一个很大的对角矩阵（因此很容易求逆矩阵）， \mathbf{B} 的行数很多列数很少（ \mathbf{C} 恰好相反），此时计算右侧的代价就远远小于计算左侧的代价。

一组向量 $\{\mathbf{a}_1, \dots, \mathbf{a}_N\}$ 被称为线性相关 (linearly independent) 如果关系 $\sum_n \alpha_n \mathbf{a}_n = 0$ 只在所有 $\alpha_n = 0$ 时成立。这表明，没有任何一个向量能够表示为其余向量的线性组合。矩阵的秩是线性无关的行的最大数量（或者等价地，线性无关的列的最大数量）。

C.2 迹和行列式

迹和行列式适用于方阵。矩阵 \mathbf{A} 的迹 $\text{Tr}(\mathbf{A})$ 被定义为主对角线上元素之和。通过把元素的下标写出来，我们可以看到

$$\text{Tr}(\mathbf{AB}) = \text{Tr}(\mathbf{BA}) \quad (\text{C.8})$$

通过多次把这个公式应用到三个矩阵的乘积上，我们看到

$$\text{Tr}(\mathbf{ABC}) = \text{Tr}(\mathbf{CAB}) = \text{Tr}(\mathbf{BCA}) \quad (\text{C.9})$$

这被称为迹操作符的循环 (cyclic) 性质。很明显这个性质可以扩展到任意数量矩阵的乘积。一个 $N \times N$ 矩阵的行列式 $|\mathbf{A}|$ 定义为

$$|\mathbf{A}| = \sum (\pm 1) A_{1i_1} A_{2i_2} \cdots A_{Ni_N} \quad (\text{C.10})$$

这个式子对所有满足下面性质的乘积进行求和：乘积包含每行的恰好一个元素和每列的恰好一个元素。系数+1或者-1取决于排列 $i_1 i_2 \dots i_N$ 是奇排列还是偶排列。注意 $|\mathbf{I}| = 1$ ，因此对于一个 2×2 矩阵，行列式的形式为

$$|\mathbf{A}| = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = a_{11}a_{22} - a_{12}a_{21} \quad (\text{C.11})$$

两个矩阵乘积的行列式为

$$|\mathbf{AB}| = |\mathbf{A}||\mathbf{B}| \quad (\text{C.12})$$

这个可以从公式 (C.10) 得到。此外，矩阵的逆矩阵的行列式为

$$|\mathbf{A}^{-1}| = \frac{1}{|\mathbf{A}|} \quad (\text{C.13})$$

取公式 (C.2) 的行列式然后应用公式 (C.12) 即可证明。

如果 \mathbf{A} 和 \mathbf{B} 是 $N \times M$ 的矩阵，那么

$$|\mathbf{I}_N + \mathbf{AB}^T| = |\mathbf{I}_M + \mathbf{A}^T \mathbf{B}| \quad (\text{C.14})$$

一种特殊情况是

$$|\mathbf{I}_N + \mathbf{ab}^T| = 1 + \mathbf{a}^T \mathbf{b} \quad (\text{C.15})$$

其中 \mathbf{a} 和 \mathbf{b} 是 N 维列向量。

C.3 矩阵的导数

有时，我们需要考虑向量和矩阵关于标量的导数。向量 \mathbf{a} 关于标量 x 的导数本身是一个向量，它的分量为

$$\left(\frac{\partial \mathbf{a}}{\partial x} \right)_i = \frac{\partial a_i}{\partial x} \quad (\text{C.16})$$

矩阵的导数的定义与此类似。关于向量和矩阵的导数也可以被定义。例如

$$\left(\frac{\partial x}{\partial \mathbf{a}} \right)_i = \frac{\partial x}{\partial a_i} \quad (\text{C.17})$$

类似地

$$\left(\frac{\partial a}{\partial b} \right)_{ij} = \frac{\partial a_i}{\partial b_j} \quad (\text{C.18})$$

写出矩阵的各个元素，下面的性质很容易证明

$$\frac{\partial}{\partial x}(\mathbf{x}^T \mathbf{a}) = \frac{\partial}{\partial x}(\mathbf{a}^T \mathbf{x}) = \mathbf{a} \quad (\text{C.19})$$

类似地

$$\frac{\partial}{\partial x}(\mathbf{AB}) = \frac{\partial \mathbf{A}}{\partial x} \mathbf{B} + \mathbf{A} \frac{\partial \mathbf{B}}{\partial x} \quad (\text{C.20})$$

矩阵的逆矩阵的导数可以表示为

$$\frac{\partial}{\partial x}(\mathbf{A}^{-1}) = -\mathbf{A}^{-1} \frac{\partial \mathbf{A}}{\partial x} \mathbf{A}^{-1} \quad (\text{C.21})$$

使用公式 (C.20) 对方程 $\mathbf{A}^{-1}\mathbf{A} = \mathbf{I}$ 求微分，然后右乘 \mathbf{A}^{-1} 即可证明。并且

$$\frac{\partial}{\partial x} \ln |\mathbf{A}| = \text{Tr} \left(\mathbf{A}^{-1} \frac{\partial \mathbf{A}}{\partial x} \right) \quad (\text{C.22})$$

这个我们稍后会证明。如果我们把 x 选成 \mathbf{A} 中的元素，那么我们有

$$\frac{\partial}{\partial A_{ij}} \text{Tr}(\mathbf{AB}) = B_{ji} \quad (\text{C.23})$$

写出矩阵的下标即可证明这个等式。我们可以把这个结论写成更加简洁的形式

$$\frac{\partial}{\partial \mathbf{A}} \text{Tr}(\mathbf{AB}) = \mathbf{B}^T \quad (\text{C.24})$$

使用这种记号，我们有下列性质

$$\frac{\partial}{\partial \mathbf{A}} \text{Tr}(\mathbf{A}^T \mathbf{B}) = \mathbf{B} \quad (\text{C.25})$$

$$\frac{\partial}{\partial \mathbf{A}} \text{Tr}(\mathbf{A}) = \mathbf{I} \quad (\text{C.26})$$

$$\frac{\partial}{\partial \mathbf{A}} \text{Tr}(\mathbf{ABA}^T) = \mathbf{A}(\mathbf{B} + \mathbf{B}^T) \quad (\text{C.27})$$

这些也可以通过写出矩阵下标的方式证明出。我们也有

$$\frac{\partial}{\partial \mathbf{A}} \ln |\mathbf{A}| = (\mathbf{A}^{-1})^T \quad (\text{C.28})$$

根据公式 (C.22) 和公式 (C.24) 即可证得。

C.4 特征向量方程

对于一个 $M \times M$ 的方阵 \mathbf{A} ，特征向量方程的定义为

$$\mathbf{Au}_i = \lambda_i \mathbf{u}_i \quad (\text{C.29})$$

其中 $i = 1, \dots, M$ ， \mathbf{u}_i 被称为特征向量 (eigenvector)， λ_i 被称为对应的特征值 (eigenvalue)。这可以看成 M 个齐次线性方程组，解存在的条件为

$$|\mathbf{A} - \lambda_i \mathbf{I}| = 0 \quad (\text{C.30})$$

这被称为特征方程 (characteristic equation)。由于这是 λ_i 的 M 阶多项式，因此它一定有 M 个解（虽然这些解未必不同）。 \mathbf{A} 的秩等于非零特征值的个数。

我们特别感兴趣的是对称矩阵。协方差矩阵、核矩阵、Hessian 矩阵都是对称矩阵。对阵矩阵的性质为 $A_{ij} = A_{ji}$ 或者等价地， $\mathbf{A} = \mathbf{A}^T$ 。对称矩阵的逆矩阵也是对称的。将 $\mathbf{A}^T \mathbf{A} = \mathbf{I}$ 取转置，然后使用 $\mathbf{AA}^{-1} = \mathbf{I}$ 以及 \mathbf{I} 的对称性即可证明这一点。

通常情况下，矩阵的特征值是复数。但是对于对称矩阵，特征值 λ_i 为实数。这点可以用下面的方式证明。首先将公式 (C.29) 左乘 $(\mathbf{u}_i^*)^T$ ，其中 * 表示复共轭，我们可以得到

$$(\mathbf{u}_i)^T \mathbf{A} \mathbf{u}_i = \lambda_i (\mathbf{u}_i^*)^T \mathbf{u}_i \quad (\text{C.31})$$

之后，我们对公式 (C.29) 取复共轭，然后左乘 \mathbf{u}_i^T ，可得

$$\mathbf{u}_i^T \mathbf{A} \mathbf{u}_i^* = \lambda_i^* \mathbf{u}_i^T \mathbf{u}_i^* \quad (\text{C.32})$$

推导过程中，我们使用了 $\mathbf{A}^* = \mathbf{A}$ ，因为我们只考虑实对称矩阵 \mathbf{A} 。将第二个方程取转置，使用 $\mathbf{A}^T = \mathbf{A}$ ，我们看到两个方程的左侧相同，从而 $\lambda_i^* = \lambda_i$ ，因此 λ_i 一定是实数。

实对称矩阵的特征向量 \mathbf{u}_i 可以被选成单位正交的（即正交的并且长度为单位长度），使得

$$\mathbf{u}_i^T \mathbf{u}_j = I_{ij} \quad (\text{C.33})$$

其中 I_{ij} 是单位矩阵 \mathbf{I} 的元素。为了证明这一点，我们首先将公式 (C.29) 左乘 \mathbf{u}_j^T ，得到

$$\mathbf{u}_j^T \mathbf{A} \mathbf{u}_i = \lambda_i \mathbf{u}_i^T \mathbf{u}_i \quad (\text{C.34})$$

因此，通过交换下标，我们有

$$\mathbf{u}_i^T \mathbf{A} \mathbf{u}_j = \lambda_j \mathbf{u}_i^T \mathbf{u}_j \quad (\text{C.35})$$

我们现在对第二个方程取转置，使用对称性质 $\mathbf{A}^T = \mathbf{A}$ ，然后将两个方程相减，可得

$$(\lambda_i - \lambda_j) \mathbf{u}_i^T \mathbf{u}_j = 0 \quad (\text{C.36})$$

因此，对于 $\lambda_i \neq \lambda_j$ ，我们有 $\mathbf{u}_i^T \mathbf{u}_j = 0$ ，因此 \mathbf{u}_i 和 \mathbf{u}_j 是正交的。如果两个特征值是相等的，那么任意线性组合 $\alpha \mathbf{u}_i + \beta \mathbf{u}_j$ 也是一个有着相同特征值的特征向量，因此我们可以任意选择一个线性组合，然后选择第二个特征向量正交于第一个（可以证明这种退化的特征向量永远不会线性相关）。因此特征向量可以选择为正交的，然后归一化为单位长度。由于有 M 个特征值，对应的 M 个特征向量组成了一个完备集，因此任意一个 M 维的向量都可以表示为特征向量的线性组合。

我们可以令特征向量 \mathbf{u}_i 是 $M \times M$ 的矩阵 \mathbf{U} ，根据单位正交性，我们有

$$\mathbf{U}^T \mathbf{U} = \mathbf{I} \quad (\text{C.37})$$

这样的矩阵被称为正交的（orthogonal）。有趣的是，矩阵的行也是正交的，即 $\mathbf{U} \mathbf{U}^T = \mathbf{I}$ 。为了证明这一点，我们注意到，公式 (C.37) 表明 $\mathbf{U}^T \mathbf{U} \mathbf{U}^{-1} = \mathbf{U}^{-1} = \mathbf{U}^T$ ，因此 $\mathbf{U} \mathbf{U}^{-1} = \mathbf{U} \mathbf{U}^T = \mathbf{I}$ 。使用公式 (C.12)，也可以看出 $|\mathbf{U}| = 1$ 。

特征向量方程 (C.29) 可以使用 \mathbf{U} 表示成下面的形式

$$\mathbf{A} \mathbf{U} = \mathbf{U} \Lambda \quad (\text{C.38})$$

其中 Λ 是一个 $M \times M$ 的对角矩阵，对角线上的元素为特征值 λ_i 。

如果我们考虑一个列向量 \mathbf{x} ，它经过正交矩阵 \mathbf{U} 进行变换，得到新向量

$$\tilde{\mathbf{x}} = \mathbf{U} \mathbf{x} \quad (\text{C.39})$$

变换前后向量的长度不变，因为

$$\tilde{\mathbf{x}}^T \tilde{\mathbf{x}} = \mathbf{x}^T \mathbf{U}^T \mathbf{U} \mathbf{x} = \mathbf{x}^T \mathbf{x} \quad (\text{C.40})$$

类似地，任意两个向量的角度在变换前后也不变，因为

$$\tilde{\mathbf{x}}^T \tilde{\mathbf{y}} = \mathbf{x}^T \mathbf{U}^T \mathbf{U} \mathbf{y} = \mathbf{x}^T \mathbf{y} \quad (\text{C.41})$$

因此，乘以 \mathbf{U} 可以表示为坐标系的刚性旋转。

根据公式 (C.38) 可得

$$\mathbf{U}^T \mathbf{A} \mathbf{U} = \Lambda \quad (\text{C.42})$$

因为 Λ 是对角矩阵，我们说矩阵 \mathbf{A} 被矩阵 \mathbf{U} 对角化（diagonalised）。如果我们左乘 \mathbf{U} 然后右乘 \mathbf{U}^T ，我们有

$$\mathbf{A} = \mathbf{U} \Lambda \mathbf{U}^T \quad (\text{C.43})$$

取这个方程的逆，然后使用公式 (C.3) 以及 $\mathbf{U}^{-1} = \mathbf{U}^T$ ，我们有

$$\mathbf{A}^{-1} = \mathbf{U} \Lambda^{-1} \mathbf{U}^T \quad (\text{C.44})$$

最后两个方程也可以写成

$$\mathbf{A} = \sum_{i=1}^M \lambda_i \mathbf{u}_i \mathbf{u}_i^T \quad (\text{C.45})$$

$$\mathbf{A}^{-1} = \sum_{i=1}^M \frac{1}{\lambda_i} \mathbf{u}_i \mathbf{u}_i^T \quad (\text{C.46})$$

如果我们取公式 (C.43) 的行列式，然后使用公式 (C.12)，我们有

$$|\mathbf{A}| = \prod_{i=1}^M \lambda_i \quad (\text{C.47})$$

类似地，取公式 (C.43) 的迹，使用迹运算的循环性 (C.8) 以及 $\mathbf{U}^T \mathbf{U} = \mathbf{I}$ ，我们有

$$\text{Tr}(\mathbf{A}) = \sum_{i=1}^M \lambda_i \quad (\text{C.48})$$

使用结论 (C.33)、(C.45)、(C.46) 和 (C.47)，可以证明公式 (C.22)，我们把证明留给读者作为练习。

一个矩阵 \mathbf{A} 被称为正定的 (positive definite)，记作 $\mathbf{A} \succ 0$ ，如果对于向量 \mathbf{w} 的所有非零值都有 $\mathbf{w}^T \mathbf{A} \mathbf{w} > 0$ 。等价地，一个正定矩阵的所有特征值都有 $\lambda_i > 0$ 。令 \mathbf{w} 为每一个特征向量，然后注意到任意的向量都可以展开为特征向量的组合，我们即可以证明这一点。注意，正定不同于所有元素都为正。例如，矩阵

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \quad (\text{C.49})$$

的特征值为 $\lambda_1 \simeq 5.37$ 且 $\lambda_2 \simeq -0.37$ 。一个矩阵被称为半正定的 (positive semidefinite)，如果对于 \mathbf{w} 的所有值都有 $\mathbf{w}^T \mathbf{A} \mathbf{w} \geq 0$ ，记作 $\mathbf{A} \succeq 0$ 。它等价于 $\lambda_i \geq 0$ 。

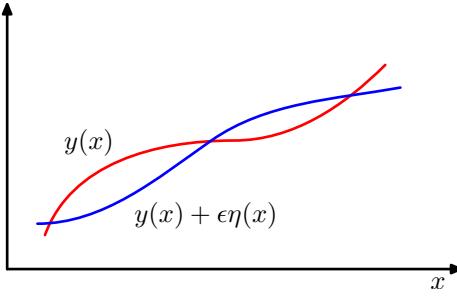


图 D.1: 泛函的导数可以按照如下的方式定义: 考虑函数从 $y(x)$ 变化到 $y(x) + \epsilon\eta(x)$ 时, 泛函 $F[y]$ 的值如何变化, 其中 $\eta(x)$ 是 x 的一个任意的函数。

D 附录D. 变分法

我们可以把函数 $y(x)$ 看成一个运算符。对于任意输入 x , 这个运算符都能返回一个输出 y 。使用同样的方式, 我们可以定义泛函 (functional) $F[y]$ 是一个运算符, 这个运算符以函数 $y(x)$ 作为输入, 返回输出 F 。泛函的一个例子是二维平面中的一条曲线的长度, 这条曲线的轨迹要根据函数来定义。在机器学习领域, 广泛使用的泛函是连续变量 x 的熵 $H[x]$, 因为对于任意概率密度函数 $p(x)$ 的选择, 它都返回一个标量值表示这个概率密度下 x 的熵。因此, $p(x)$ 的熵写成 $H[p]$ 也一样没错。

传统的微积分中的一个常见的问题是找到一个 x 值使得 $y(x)$ 取得最大值或者最小值。类似地, 变分法中, 我们寻找一个函数 $y(x)$ 来最大化或者最小化泛函 $F[y]$ 。即, 对于所有可能的函数 $y(x)$, 我们想找到一个特定的函数, 使得 $F[y]$ 达到最大值或者最小值。变分法可以用来说明两点之间的最短路径是一条直线, 或者最大熵分布是高斯分布。

如果我们不熟悉普通微积分的规则, 那么我们在求传统的导数 $\frac{dy}{dx}$ 时, 我们可以首先让变量 x 产生一个微小的改变 ϵ , 然后对 ϵ 进行幂级数展开, 即

$$y(x + \epsilon) = y(x) + \frac{dy}{dx} \epsilon + O(\epsilon^2) \quad (\text{D.1})$$

最后取极限 $\epsilon \rightarrow 0$ 。类似地, 对于一个多变量函数 $y(x_1, \dots, x_D)$, 对应的偏导数通过下式定义

$$y(x_1 + \epsilon_1, \dots, x_D + \epsilon_D) = y(x_1, \dots, x_D) + \sum_{i=1}^D \frac{\partial y}{\partial x_i} \epsilon_i + O(\epsilon^2) \quad (\text{D.2})$$

类似地, 我们可以得到泛函的导数的定义。当我们对函数 $y(x)$ 做一个微小的改变 $\epsilon\eta(x)$ (其中 $\eta(x)$ 是 x 的一个任意的函数) 时, 我们考虑泛函 $F[y]$ 的变化, 如图D.1所示。我们把泛函 $F[y]$ 关于 $y(x)$ 的导数记作 $\frac{\delta F}{\delta y(x)}$, 通过下面的关系定义

$$F[y(x) + \epsilon\eta(x)] = F[y(x)] + \epsilon \int \frac{\delta F}{\delta y(x)} \eta(x) dx + O(\epsilon^2) \quad (\text{D.3})$$

这可以被看成公式 (D.2) 的一个自然的推广, 其中 $F[y]$ 现在依赖于变量的一个连续集合, 即在所有 x 处的 y 值。令泛函的值在函数 $y(x)$ 发生微小改变时几乎不变, 可得

$$\int \frac{\delta F}{\delta y(x)} \eta(x) dx = 0 \quad (\text{D.4})$$

由于这必须对任意的 $\eta(x)$ 都成立, 因此我们必须令泛函的导数等于零。为了证明这一点, 让我们假设选择一个扰动 $\eta(x)$, 这个扰动只在点 \hat{x} 的邻域内等于零, 在其他各处均不等于零。这种情况下, 泛函的导数必须在 $x = \hat{x}$ 处等于零。但是, 由于这个结论必须对于任意的 \hat{x} 都成立, 因此泛函的导数必须对所有的 x 值都等于零。

考虑一个泛函, 这个泛函由函数 $G(y, y', x)$ 的积分定义。函数 $G(y, y', x)$ 既依赖于 $y(x)$ 又依赖于它的导数 $y'(x)$, 还直接依赖于 x 。因此, 这个泛函的形式为

$$F[y] = \int G(y(x), y'(x), x) dx \quad (\text{D.5})$$

其中，我们假设 $y(x)$ 的值在积分边界（可能是无穷）处是定值。如果我们考虑函数 $y(x)$ 的改变，那么我们有

$$F[y(x) + \epsilon\eta(x)] = F[y(x)] + \epsilon \int \left\{ \frac{\partial G}{\partial y} \eta(x) + \frac{\partial G}{\partial y'} \eta'(x) \right\} dx + O(\epsilon^2) \quad (\text{D.6})$$

我们现在必须把它转化为公式 (D.3) 的形式。为了完成这一点，我们将第二项进行分部积分，然后使用 $\eta(x)$ 必须在积分边界处等于零的事实（因为 $y(x)$ 在边界处为定值）。因此

$$F[y(x) + \epsilon\eta(x)] = F[y(x)] + \epsilon \int \left\{ \frac{\partial G}{\partial y} - \frac{d}{dx} \left(\frac{\partial G}{\partial y'} \right) \right\} \eta(x) dx + O(\epsilon^2) \quad (\text{D.7})$$

与公式 (D.3) 对比，我们可以直接读出泛函的导数。令泛函的导数等于零，我们有

$$\frac{\partial G}{\partial y} - \frac{d}{dx} \left(\frac{\partial G}{\partial y'} \right) = 0 \quad (\text{D.8})$$

这被称为欧拉-拉格朗日方程 (Euler-Lagrange equation)。例如，如果

$$G = y(x)^2 + (y'(x))^2 \quad (\text{D.9})$$

那么，欧拉-拉格朗日方程的形式为

$$y(x) - \frac{d^2 y}{dx^2} = 0 \quad (\text{D.10})$$

使用 $y(x)$ 的边界条件，我们可以解出这个关于 $y(x)$ 的二阶微分方程。

通常情况下，我们考虑定义在积分上的泛函时，被积函数的形式为 $G(y, x)$ ，不依赖于 $y(x)$ 的导数。这种情况下，驻点只需要令 $\frac{\partial G}{\partial y(x)} = 0$ 对于所有的 x 都成立即可。

如果我们关于概率分布对泛函进行优化，那么我们需要保持概率的归一化限制。使用拉格朗日乘数法来进行优化是最方便的。使用拉格朗日乘数法之后，我们就可以进行无限制条件的最优化。

上述结果在多维变量 x 上的扩展是很直接的。对于变分法的一个可理解的讨论，可以参考Sagan (1969)。

E 附录E. 拉格朗日乘数法

拉格朗日乘数法 (Lagrange multiplier) , 有时也被称为不确定乘数法 (undetermined multiplier) , 被用于寻找多元变量在一个或者多个限制条件下的驻点。

考虑寻找函数 $f(x_1, x_2)$ 的最大值, 其中 x_1 和 x_2 要满足一定的限制, 限制的形式为

$$g(x_1, x_2) = 0 \quad (\text{E.1})$$

一种方法是求解限制方程 (E.1) , 把 x_2 表示为 x_1 的函数, 形式为 $x_2 = h(x_1)$ 。这之后就可以代入 $f(x_1, x_2)$, 变为关于 x_1 单一变量的函数, 形式为 $f(x_1, h(x_1))$ 。关于 x_1 的最大值能够使用通常的方法用微分的方式求出, 给出驻点值 x_1^* , 对应的 x_2 的值为 $x_2^* = h(x_1^*)$ 。

这种方法的一个问题是, 把 x_2 显式地表示为 x_1 的函数, 即找到限制方程的解析解很困难。并且, 这种方法把 x_1 和 x_2 区别对待, 这破坏了这些变量之间自然存在的对称性。

一个更加优雅且通常很简单的方法依赖于引入一个被称为拉格朗日乘数的参数 λ 。我们从几何角度来说一下这个方法。考虑一个 D 维变量 \mathbf{x} , 分量为 x_1, \dots, x_D 。限制方程 $g(\mathbf{x}) = 0$ 表示 \mathbf{x} 空间中的一个 $(D - 1)$ 维曲面, 如图E.1所示。

我们首先注意到, 在限制曲面上的任何点处, 限制函数的梯度 $\nabla g(\mathbf{x})$ 都正交于限制曲面。为了证明这一点, 考虑一个位于限制曲面上的点 \mathbf{x} 以及这个点附近同样位于曲面上的点 $\mathbf{x} + \epsilon$ 。如果我们在点 \mathbf{x} 处进行泰勒展开, 那么我们有

$$g(\mathbf{x} + \epsilon) \simeq g(\mathbf{x}) + \epsilon^T \nabla g(\mathbf{x}) \quad (\text{E.2})$$

由于 \mathbf{x} 和 $\mathbf{x} + \epsilon$ 都位于限制曲面上, 我们有 $g(\mathbf{x}) = g(\mathbf{x} + \epsilon)$, 因此 $\epsilon^T \nabla g(\mathbf{x}) \simeq 0$ 。在极限 $\|\epsilon\| \rightarrow 0$ 的情况下, 我们有 $\epsilon^T \nabla g(\mathbf{x}) = 0$ 。由于 ϵ 平行于限制曲面, 因此我们看到向量 ∇g 正交于曲面。

接下来我们寻找限制曲面上的一个点 \mathbf{x}^* 使得 $f(\mathbf{x})$ 最大。这样的一个点一定满足这样的性质: 向量 $\nabla f(\mathbf{x})$ 也正交于限制曲面, 如图E.1所示, 因为如果这个性质不满足的话, 我们就可以沿着限制曲面移动一个较短的距离来使 $f(\mathbf{x})$ 增大。因此 ∇f 和 ∇g 是平行的 (或者反平行的) 向量, 因此一定存在一个参数 λ 使得

$$\nabla f + \lambda \nabla g = 0 \quad (\text{E.3})$$

其中 $\lambda \neq 0$ 被称为拉格朗日乘数 (Lagrange multiplier) 。注意, λ 的符号任意。

这里, 因数一个拉格朗日函数比较方便。拉格朗日函数定义如下

$$L(\mathbf{x}, \lambda) \equiv f(\mathbf{x}) + \lambda g(\mathbf{x}) \quad (\text{E.4})$$

公式 (E.3) 给出的函数驻点处的条件可以通过令 $\nabla_{\mathbf{x}} L = 0$ 来得到。更进一步, 条件 $\frac{\partial L}{\partial \lambda} = 0$ 会导出限制方程 $g(\mathbf{x}) = 0$ 。

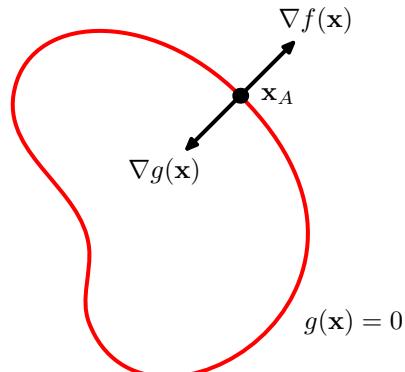


图 E.1: 拉格朗日乘数法的集合说明, 其中我们寻找函数 $f(\mathbf{x})$ 的最大值, 满足限制条件 $g(\mathbf{x}) = 0$ 。如果 \mathbf{x} 是 D 维的, 那么限制条件 $g(\mathbf{x}) = 0$ 对应于 $D - 1$ 维的子空间, 用红色曲线表示。问题可以通过最优化拉格朗日函数 $L(\mathbf{x}, \lambda) = f(\mathbf{x}) + \lambda g(\mathbf{x})$ 的方式求出。

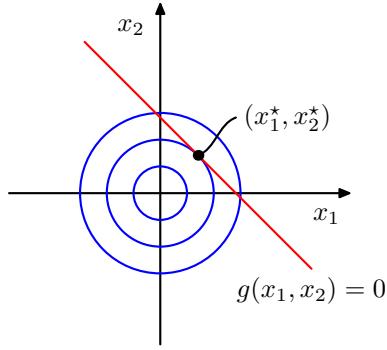


图 E.2: 使用拉格朗日乘数的一个简单的例子, 其中目标是最大化 $f(x_1, x_2) = 1 - x_1^2 - x_2^2$, 满足限制条件 $g(x_1, x_2) = 0$, 其中 $g(x_1, x_2) = x_1 + x_2 - 1$ 。圆形表示函数 $f(x_1, x_2)$ 的轮廓线, 对角线表示限制曲面 $g(x_1, x_2) = 0$ 。

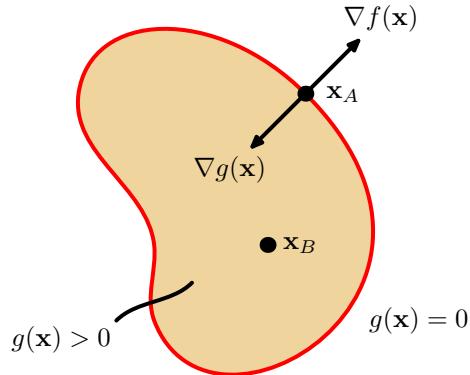


图 E.3: 满足不等式限制条件 $g(\mathbf{x}) \geq 0$ 下, 最大化 $f(\mathbf{x})$ 的问题的例子。

因此为了寻找函数 $f(\mathbf{x})$ 在限制条件 $g(\mathbf{x}) = 0$ 下的最大值, 我们定义了公式 (E.4) 给出的拉格朗日函数, 并且我们能够找到 $L(\mathbf{x}, \lambda)$ 关于 \mathbf{x} 和 λ 的驻点。对于一个 D 维向量 \mathbf{x} , 这种方法给出了 $D + 1$ 个方程确定驻点 \mathbf{x}^* 和 λ 的值。如果我们只对 \mathbf{x}^* 感兴趣, 那么我们可以从函数驻点处的方程 (E.3) 中消去 λ , 不需要找到它的值 (因此有了术语“不确定乘数法”)。

作为一个简单的例子, 假设我们想找到函数 $f(x_1, x_2) = 1 - x_1^2 - x_2^2$ 在限制条件 $g(x_1, x_2) = x_1 + x_2 - 1 = 0$ 下的驻点, 如图 E.2 所示。对应的拉格朗日函数为

$$L(\mathbf{x}, \lambda) = 1 - x_1^2 - x_2^2 + \lambda(x_1 + x_2 - 1) \quad (\text{E.5})$$

这个拉格朗日函数关于 x_1, x_2 和 λ 的驻点处的条件有下列方程给出

$$-2x_1 + \lambda = 0 \quad (\text{E.6})$$

$$-2x_2 + \lambda = 0 \quad (\text{E.7})$$

$$x_1 + x_2 - 1 = 0 \quad (\text{E.8})$$

这些方程的解给出了驻点 $(x_1^*, x_2^*) = (\frac{1}{2}, \frac{1}{2})$, 对应的拉格朗日乘数为 $\lambda = 1$ 。

目前为止, 我们已经考虑了在形式为 $g(\mathbf{x}) = 0$ 的等式限制 (equality constraint) 下最大化函数的问题。我们现在考虑形式为 $g(\mathbf{x}) \geq 0$ 的不等式限制 (inequality constraint) 下最大化函数 $f(\mathbf{x})$ 的问题, 如图 E.3 所示。

根据受限制条件下的驻点是否位于区域 $g(\mathbf{x}) > 0$ 中, 有两种可能的解。如果驻点位于 $g(\mathbf{x}) > 0$ 的区域中, 我们说限制条件未激活 (inactive)。如果驻点位于 $g(\mathbf{x}) = 0$ 的边界上, 我们说限制条件激活 (active)。在第一种情况下, 函数 $g(\mathbf{x})$ 不起作用, 函数在驻点处的条件只是 $\nabla f(\mathbf{x}) = 0$ 。这同样对应于拉格朗日方程 (E.4) 的驻点, 但是 $\lambda = 0$ 。在后一种情况下, 解位于边界上, 这类似于之前讨论过的等式限制的情形, 对应于拉格朗日方程 (E.4) 在 $\lambda \neq 0$ 的条件下的驻点。但是现在, 拉格朗日乘数的符号很重要, 因为只有当梯度向量指向远

离 $g(\mathbf{x}) > 0$ 的区域时，函数 $f(\mathbf{x})$ 才会取得最大值，如图 E.3 所示。于是对于某些 $\lambda > 0$ ，我们有 $\nabla f(\mathbf{x}) = -\lambda \nabla g(\mathbf{x})$ 。

对于两种情况中的任意一种，乘积 $\lambda g(\mathbf{x}) = 0$ 。因此在限制条件 $g(\mathbf{x}) \geq 0$ 下最大化 $f(\mathbf{x})$ 的问题的解可以通过下面的方式获得：关于 \mathbf{x} 和 λ 最优化拉格朗日函数 (E.4)，限制条件为

$$g(\mathbf{x}) \geq 0 \quad (\text{E.9})$$

$$\lambda \geq 0 \quad (\text{E.10})$$

$$\lambda g(\mathbf{x}) = 0 \quad (\text{E.11})$$

这些被称为 Karush-Kuhn-Tucker (KKT) 条件 (Karush, 1939; Kuhn and Tucker, 1951)。

注意，如果我们想在不等式限制 $g(\mathbf{x}) \geq 0$ 下最小化（而不是最大化）函数 $f(\mathbf{x})$ ，那么我们要关于 \mathbf{x} 最小化拉格朗日函数 $L(\mathbf{x}, \lambda) = f(\mathbf{x}) - \lambda g(\mathbf{x})$ ，限制条件为 $\lambda \geq 0$ 。

最后，将拉格朗日乘数法的技术推广到多个等式限制和不等式限制的情形是很直接的。假设我们希望在限制条件为 $g_j(\mathbf{x}) = 0, j = 1, \dots, J$ 和 $h_k(\mathbf{x}) \geq 0, k = 1, \dots, K$ 的情况下最大化 $f(\mathbf{x})$ ，我们就会引入拉格朗日乘数 $\{\lambda_j\}$ 和 $\{\mu_k\}$ ，然后最优化下面的拉格朗日函数

$$L(\mathbf{x}, \{\lambda_j\}, \{\mu_k\}) = f(\mathbf{x}) + \sum_{j=1}^J \lambda_j g_j(\mathbf{x}) + \sum_{k=1}^K \mu_k h_k(\mathbf{x}) \quad (\text{E.12})$$

限制条件为 $\mu_k \geq 0$ 且 $\mu_k h_k(\mathbf{x}) = 0, k = 1, \dots, K$ 。推广到有限制条件下的泛函的导数的情况也与此类似。关于拉格朗日乘数法的更加详细的讨论，请参考 Nocedal and Wright (1999)。