子午线

学习笔记

作者：leekarry
组织：果壳
时间：July 18, 2019
版本：0.1

ElegantLaTeX Program

我很快，快到时间都会变慢。而我一慢，时间就会过得飞快。

目　录

第 I 部分 I

机器学习

第 1 章 机器学习概论

1.1 基本概念

统计学习的特点

统计学习 (statistical learning) 是关于计算机基于数据构建概率统计模型并运用模型对数据进行预测与分析的一门学科。

- 统计学习以计算机及网络为平台；
- 统计学习以数据为研究对象；
- 统计学习的目的是对数据进行预测与分析；
- 统计学习以方法为中心；
- 统计学习是概率论、统计学、信息论、计算理论、最优化理论及计算机科学等多个领域的交叉学科。

统计学习的对象

统计学习的对象是数据 (data)。它从数据出发，提取数据的特征，抽象出数据的模型，发现数据中的知识，又回到对数据的分析与预测中去。**统计学习关于数据的基本假设是同类数据具有一定的统计规律性，这是统计学习的前提**。

统计学习的目的

统计学习用于对数据进行预测与分析，特别是对未知新数据进行预测与分析。对数据的预测与分析是通过构建概率统计模型实现的。统计学习总的目标就是考虑学习什么样的模型和如何学习模型，以使模型能对数据进行准确的预测与分析，同时也要考虑尽可能地提高学习效率。

统计学习的方法

统计学习的方法是基于数据构建统计模型从而对数据进行预测与分析。

- 监督学习 (supervised learning)
- 非监督学习 (unsupervised learning)
- 半监督学习 (semi-supervised learning)
- 强化学习 (reinforcement learning)

实现统计学习方法的步骤如下：

(1) 得到一个有限的训练数据集合

(2) 确定包含所有可能的模型的假设空间，即学习模型的集合

(3) 确定模型选择的准则，即学习的策略

(4) 实现求解最优化模型的算法,即学习的算法
(5) 通过学习方法选择最优模型
(6) 利用学习的最优模型对新数据进行预测或分析

统计学习的研究

统计学习方法 (statistical learning method),旨在开发新的学习方法。

统计学习理论 (statistical learning theory),旨在探求统计学习方法的有效性与效率。

统计学习应用 (application of statistical learning),旨在将统计学习方法应用到实际问题中去,解决实际问题。

统计学习的重要性

统计学习是处理海量数据的有效方法;统计学习是计算机智能化的有效手段;统计学习是计算机科学习发展的一个重要组成部分。

1.2 统计学习三要素

(1) 模型:统计学习首先要考虑的问题是学习什么样的模型
(2) 策略:有了模型的假设空间,统计学习接着需要考虑的是按照什么样的准则学习或选择最优的模型
(3) 算法:算法是指学习模型的具体计算方法

1.3 模型评估与模型选择

统计学习的目的是使学到的模型不仅对已知数据而且对未知数据都能有很好的预测能力。不同的学习方法会给出不同的模型。当损失函数给定时,基于损失函数的模型的训练误差(training error)和模型的测试误差(test error)就自然成为学习方法评估的标准。测试误差反映了学习方法对未知的测试数据集的预测能力——泛化能力。

1.4 正则化与交叉验证

模型选择的典型方法是正则化 (regularization)。正则化是结构风险最小化策略的实现,是在经验风险上加一个正则化项 (regularizer) 和罚项 (penalty term)

$$\min_{f\in F} \quad \frac{1}{2}\sum_{i=1}^{N} L(y_i, f(x_i)) + \lambda J(f) \tag{1.1}$$

另一种常用的模型选择方法是交叉验证 (cross validation) 训练集用来训练模型验证集用来选择模型测试集用来对模型进行评估

1. 简单交叉验证

2. S 折交叉验证
3. 留一交叉验证

1.5 生成模型与判别模型

监督学习方法又可以分为生成方法 (generative approach) 和判别方法 (discriminative approach)。所学到的模型分别称为生成模型和判别模型。

生成方法由数据学习联合分布 $P(X,Y)$,然后求出条件概率分布 $P(Y|X)$ 作为预测的模型,即生成模型:

$$P(Y|X) = \frac{P(X,Y)}{P(X)} \tag{1.2}$$

这样的方法之所以称为生成方法,是因为模型表示了给定输入 X 产生输出 Y 的生成关系。

判别方法由数据直接学习决策函数 $f(X)$ 或者条件概率分布 $P(Y|X)$ 作为预测的模型,即判模型。判别方法关心的是对给定的输入 X,应该预测什么样的输出 Y。

生成方法的特点:生成方法可以还原出联合概率分布 P(X|Y),而判别方法则不能;生成方法的学习收敛速度更快,即当样本容量增加的时候,学到的模型可以更快地收敛于真实模型;当存在隐变量时,仍可以用生成方法学习,此时判别方法就不能用。

判别方法的特点:判别方法直接学习的是条件概率 P(Y|X) 或决策函数 $f(X)$,直接面对预测,往往学习的准确率更高;由于直接学习 P(Y|X) 或 $f(X)$,可以对数据进行各种程度上的抽象、定义特征并使用特征,因此可以简化学习问题。

1.6 频率学派和贝叶斯学派的参数估计

频率学派与贝叶斯学派的区别

简单地说,频率学派与贝叶斯学派探讨**不确定性**这件事的出发点与立足点同。频率学派从" 自然" 角度出发,试图直接为" 事件" 本身建模,即事件 A 在独立重复试验中发生的频率趋于极限 p,那么这个极限就是该事件发生的概率。贝叶斯学派并不从试图刻画" 事件" 本身,而从" 观察者" 角度出发。贝叶斯学派并不试图说" 事件本身是随机的",或者" 世界的本体带有某种随机性",而只是从" 观察者知识不完备" 这一出发点开始,构造一套在贝叶斯概率论的框架下可以对不确定知识做出推断的方法。体现在参数估计中,频率学派认为参数是客观存在,不会改变,虽然未知,但却是固定值;贝叶斯学派则认为参数是随机值,因此参数也可以有分布。

频率学派的参数估计

极大似然估计 (Maximum Likelihood Estimate,MLE),也叫最大似然估计。若总体 X 属离散型 (连续型与此类似),其分布律 $P\{X = x\} = p(x;\theta), \theta \in \Theta$ 的形式为已知,θ 为待估参数,Θ 是 θ 的取值范围,设 $X_1, X_2, \ldots, X_n$ 是来自 X 的样本,则 $X_1, X_2, \ldots, X_n$ 的联合概率分

布为

$$\prod_{i=1}^{n} p(x;\theta) \tag{1.3}$$

设 $x_1, x_2, \ldots, x_n$ 是相应的样本值，则

$$L(\theta) = L(x_1, x_2, \ldots, x_n; \hat{\theta}) = \underset{\theta \in \Theta}{argmax} \prod_{i=1}^{n} p(x;\theta) \tag{1.4}$$

贝叶斯学派的参数估计

最大后验估计 (Maximum a Posteriori estimation,MAP)，它与极大似然估计最大的区别就是，它考虑了参数本身的分布，也就是先验分布。最大后验估计是根据经验数据获得对难以观察的量的点估计。可以看作规则化的最大似然估计。假设 x 为独立同分布的采样，θ 为模型参数，p 为我们所使用的模型。那么最大似然估计可以表示为

$$\hat{\theta}_{MLE}(x) = \underset{\theta}{argmax}\, p(x|\theta) \tag{1.5}$$

现在，假设 θ 的先验分布为 g。通过贝叶斯理论，对于 θ 的后验分布如下式所示：

$$p(\theta|x) = \frac{p(x|\theta)g(\theta)}{\int_{\theta \in \Theta} p(x|\theta')g(\theta')d\theta'} \tag{1.6}$$

分母为 x 的边缘概率与 θ 无关，因此最大后验等价于使分子最大，故目标函数为

$$\hat{\theta}_{MAP}(x) = \underset{\theta}{argmax}\, p(x|\theta)g(\theta) \tag{1.7}$$

第 2 章 概率分布

概率论在解决模式识别问题时起着重要作用。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。概率分布的一个作用是在给定有限次观测 $x_1, \ldots, x_N$ 的前提下，对随机变量 $\vec{x}$ 的概率分布 $p(\vec{x})$ 建模。这个问题被称为密度估计 (density estimation)。本章中，我们假设数据点是独立同分布的。

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 (paramertric distribution) 的具体例子。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。指数族分布有很多重要的性质，将在本章详细讨论。

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 二元分布

考虑一个二元随机变量 $x \in \{0, 1\}$。$x = 1$ 的概率被记作参数 μ，因此

$$p(x = 1|\mu) = \mu \tag{2.1}$$

其中 $0 \leqslant \mu \leqslant 1$。$x$ 的概率分布因此可以写成

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

这被叫做伯努利分布 (Bernoulli distribution)。均值和方差为

$$\mathbb{E}[x] = \mu \tag{2.3}$$

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

用最大似然估计方法求得

$$\mu_{ML} = \frac{1}{N}\sum_{n=1}^{N} x_n = \frac{m}{N} \tag{2.5}$$

m 为数据集里 $x = 1$(正面朝上) 的观测数量。

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。这被称谓二项分布 (binomial distribution)。

$$\mathrm{Bin}(m|N,\mu) = \binom{N}{m}\mu^m(1-\mu)^{N-m} \tag{2.6}$$

其中

$$\binom{N}{m} \equiv \frac{N!}{(N-m)!m!} \tag{2.7}$$

均值和方差为

$$\mathbb{E}[m] \equiv \sum_{m=0}^{N} m\mathrm{Bin}(m|N,\mu) = N\mu \tag{2.8}$$

$$\mathrm{var}[m] \equiv \sum_{m=0}^{N}(m-\mathbb{E}[m])^2\mathrm{Bin}(m|N,\mu) = N\mu(1-\mu) \tag{2.9}$$

Beta 分布

我们已经看到伯努利分布的参数 μ 的最大似然解。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 的先验分布 $p(\mu)$。

在贝叶斯统计中，如果后验分布与先验分布属于同类，则先验分布与后验分布被称为**共轭分布**，而先验分布被称为似然函数的共轭先验。具体地说，就是给定贝叶斯公式，假定似然函数 $p(x|\theta)$ 是已知的，问题就是选取什么样的先验分布 $p(\theta)$ 会让后验分布与先验分布具有相同的数学形式。共轭先验的好处主要在于代数上的方便性，可以直接给出后验分布的封闭形式，否则的话只能数值计算。共轭先验也有助于获得关于似然函数如何更新先验分布的直观印象。所有指数家族的分布都有共轭先验。

因此，我们把先验分布选择为 Beta 分布，定义为

$$\mathrm{Beta}(\mu|a,b) = \frac{\Gamma(a+b)}{\Gamma(a)+\Gamma(b)}\mu^{a-1}(1-\mu)^{b-1} \tag{2.10}$$

其中，$\Gamma(x)$ 定义为

$$\Gamma(x) \equiv \int_0^\infty u^{x-1}e^{-u}du \tag{2.11}$$

有以下性质

$$\Gamma(x+1) = x\Gamma(x) \tag{2.12}$$

$$\Gamma(x+1) = x! \tag{2.13}$$

Beta 分布的均值和方差为

$$\mathbb{E}[\mu] = \frac{a}{a+b} \tag{2.14}$$

$$\text{var}[\mu] = \frac{ab}{(a+b)^2(a+b+1)} \tag{2.15}$$

参数 a 和 b 被称为超参数 (hyperparameter)，因为它们控制了参数 μ 的概率分布。

μ 的后验概率分布现在可以这样得到：把 Beta 先验与二项似然函数相乘，然后归一化。我们看到后验概率分布的形式为

$$p(\mu|m,l,a,b) = \frac{\Gamma(m+a+l+b)}{\Gamma(m+a)\Gamma(l+b)}\mu^{m+a+1}(1-\mu)^{l+b+1} \tag{2.16}$$

其中 $l = N - m$，即对应于硬币“反面朝上”的样本数量。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。

另外，如果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。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。

给定数据集 D 的情况下，x 的预测分布为

$$p(x=1|D) = \int_0^1 p(x=1|\mu)p(\mu|D)d\mu = \int_0^1 \mu p(\mu|D)d\mu = \mathbb{E}[\mu|D] \tag{2.17}$$

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，经常会遇到可以取 K 个互斥状态中的某一种的离散变量。一种比较方便的表示方法是“1-of-K”表示法。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 x 就可以表示为

$$\boldsymbol{x} = (0,0,1,0,0,0)^T \tag{2.18}$$

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 x 的分布就是

$$p(\boldsymbol{x}|\boldsymbol{\mu}) = p(x_1, x_2, \ldots, x_K|\mu_1, \ldots, \mu_K) = \prod_{k=1}^{K} \mu_k^{x_k} \tag{2.19}$$

参数 μ_k 要满足 $\mu_k \geqslant 0, \sum_k \mu_k = 1$。易知

$$\mathbb{E}[\boldsymbol{x}|\boldsymbol{\mu}] = \sum_{\boldsymbol{x}} p(\boldsymbol{x}|\mu)\boldsymbol{x} = (\mu_1, \mu_2, \ldots, \mu_K)^T = \boldsymbol{\mu} \tag{2.20}$$

现在考虑一个有 N 个独立观测值 $x_1, x_2, \ldots, x_N$ 的数据集 D。对应的似然函数的形式为

$$p(D|\mu) = \prod_{n=1}^{N}\prod_{k=1}^{K} \mu_k^{x_{nk}} = \prod_{k=1}^{K} \mu_k^{(\sum_n x_{nk})} = \prod_{k=1}^{K} \mu_k^{m_k} \tag{2.21}$$

其中

$$m_k = \sum_n x_{nk} \tag{2.22}$$

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。为了找到 $\boldsymbol{\mu}$ 的最大似然解，需要关于 μ_k 最大化 $\ln p(D|\boldsymbol{\mu})$，通过拉格朗日乘数 λ 实现。

$$\mu_k^{ML} = \frac{m_k}{N} \tag{2.23}$$

我们可以考虑 $m_1, m_2, \ldots, m_K$ 在参数 μ 和观测总数 N 条件下的联合分布。这个分布的形式为

$$Mult(m_1, m_2, \ldots, m_k|\boldsymbol{\mu}, N) = \binom{N}{m_1 m_2 \ldots m_K} \prod_{k=1}^{K} \mu_k^{m_k} \tag{2.24}$$

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 $m_1, m_2, \ldots, m_K$ 的 K 组的方案总数，定义为

$$\binom{N}{m1m2 \ldots m_K} = \frac{N!}{m_1!m_2! \ldots m_K!} \tag{2.25}$$

注意，m_k 满足下面的限制

$$\sum_{k=1}^{K} m_k = N \tag{2.26}$$

狄利克雷分布

现在介绍多项式分布的参数 $\{\mu_k\}$ 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

$$p(\boldsymbol{\mu}|\alpha) \propto \prod_{k=1}^{K} \mu_k^{\alpha_k - 1} \tag{2.27}$$

其中 $0 \leqslant \mu_k \leqslant 1$ 且 $\sum_k \mu_k = 1$。这里，$\alpha_1, \alpha_2, \ldots, \alpha_K$ 是分布的参数，$\boldsymbol{\alpha}$ 表示 $(\alpha_1, \alpha_2, \ldots, \alpha_K)^T$。概率的归一化形式为

$$Dir(\boldsymbol{\mu}|\boldsymbol{\alpha}) = \frac{\Gamma(\alpha_0)}{\Gamma(\alpha_1)\ldots\Gamma(\alpha_K)} \prod_{k=1}^{K} \mu_k^{\alpha_k - 1} \tag{2.28}$$

这被称为狄利克雷分布 (Dirichlet distribution)。

$$\alpha_0 = \sum_{k=1}^{K} \alpha_k \tag{2.29}$$

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

$$p(\boldsymbol{\mu}|D, \boldsymbol{\alpha}) \propto p(D|\boldsymbol{\mu})p(\boldsymbol{\mu}|\boldsymbol{\alpha}) \propto \prod_{k=1}^{K} \mu_k^{\alpha_k + m_k - 1} \tag{2.30}$$

我们看到后验分布的形式又变成了狄利克雷分布。确定归一化系数后变成

$$p(\boldsymbol{\mu}|D, \boldsymbol{\alpha}) = Dir(\boldsymbol{\mu}|\boldsymbol{\alpha} + \boldsymbol{m}) \tag{2.31}$$

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，

$$\mathcal{N}(x|\mu, \sigma^2) = \frac{1}{(2\pi\sigma^2)^{\frac{1}{2}}} \exp\left\{-\frac{1}{2\sigma^2}(x-\mu)^2\right\} \tag{2.32}$$

其中 μ 是均值，σ^2 是方差。

$$\mu_{ML} = \frac{1}{N}\sum_{i=1}^{N} x_i \qquad \text{无偏} \tag{2.33}$$

$$\sigma^2_{ML} = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu_{ML})^2 \qquad \text{有偏} \tag{2.34}$$

$$\mathbb{E}[\sigma^2_{ML}] = \frac{N-1}{N}\sigma^2 \tag{2.35}$$

$$\hat{\sigma}^2 = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \mu_{ML})^2 \qquad \text{无偏} \tag{2.36}$$

对于 D 维向量 $\boldsymbol{x}$，多元高斯分布的形式为

$$\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}, \Sigma) = \frac{1}{(2\pi)^{\frac{D}{2}}|\Sigma|^{\frac{1}{2}}} \exp\left\{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right\} \tag{2.37}$$

其中，μ 是一个 D 维均值向量，Σ 是一个 $D \times D$ 的协方差矩阵，$|\Sigma|$ 是 Σ 的行列式。对 Σ 进

行正交分解。

$$\Sigma = U\Lambda U^T \tag{2.38}$$

$$UU^T = U^TU = I \tag{2.39}$$

$$\Lambda = diag(\lambda_i) \tag{2.40}$$

$$U = (u_1, u_2, \ldots, u_p)_{p\times p} \tag{2.41}$$

考虑高斯分布的几何形式。高斯对于 $\boldsymbol{x}$ 的依赖是通过下面形式的二次型

$$\triangle^2 = (\boldsymbol{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu}) \tag{2.42}$$

$\triangle$ 被叫做 $\boldsymbol{\mu}$ 和 $\boldsymbol{x}$ 之间的马氏距离。协方差矩阵 Σ 可以表示成特征向量的展开的形式

$$\begin{aligned}\Sigma &= U\Lambda U^T \\ &= (u_1\ u_2\ \ldots\ u_p)\begin{pmatrix}\lambda_1 & \ldots & 0\\ \vdots & \lambda_i & \vdots\\ 0 & \ldots & \lambda_p\end{pmatrix}\begin{pmatrix}u_1^T\\ \vdots\\ u_P^T\end{pmatrix} \\ &= (u_1\lambda_1\ \ldots\ u_p\lambda_p)\begin{pmatrix}u_1^T\\ \vdots\\ u_P^T\end{pmatrix} \\ &= \sum_{i=1}^{P} u_i\lambda_i u_i^T\end{aligned} \tag{2.43}$$

于是

$$\begin{aligned}\Sigma^{-1} &= (U\Lambda U^T)^{-1} = U\Lambda^{-1}U^T \\ &= \sum_{i=1}^{P} u_i\frac{1}{\lambda_i}u_i^T\end{aligned} \tag{2.44}$$

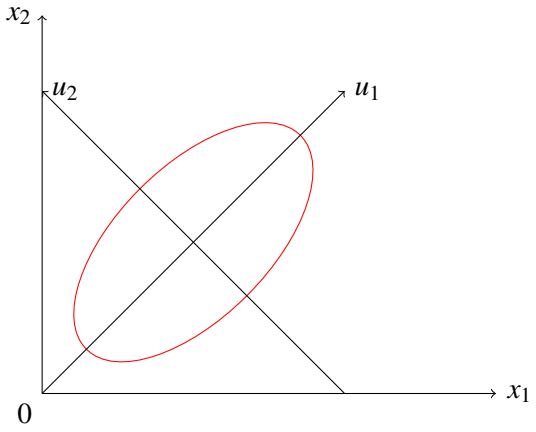
马氏距离就可以表示为

$$\begin{aligned}\triangle^2 &= (\boldsymbol{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu}) \\ &= \sum_{i=1}^{P}\underbrace{(\boldsymbol{x}-\boldsymbol{\mu})^Tu_i}_{y_i}\frac{1}{\lambda_i}\underbrace{u_i^T(\boldsymbol{x}-\boldsymbol{\mu})}_{y_i^T} \quad = \sum_{i=1}^{P}\frac{y_i^2}{\lambda_i}\end{aligned} \tag{2.45}$$

其中

$$y_i = (\boldsymbol{x}-\boldsymbol{\mu})^Tu_i \tag{2.46}$$

可以把 $\{y_i\}$ 表示成单位正交向量 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。



高斯分布的均值和方差为

$$\begin{aligned}\mathbb{E}[\boldsymbol{x}] &= \frac{1}{(2\pi)^{\frac{D}{2}}|\Sigma|^{\frac{1}{2}}}\int \exp\left\{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right\} x\mathrm{d}x \\ &= \frac{1}{(2\pi)^{\frac{D}{2}}|\Sigma|^{\frac{1}{2}}}\int \exp\left\{-\frac{1}{2}z^T\Sigma^{-1}z\right\}(z+\mu)\mathrm{d}z\end{aligned} \tag{2.47}$$

其中我们使用 $z = \boldsymbol{x} - \boldsymbol{\mu}$ 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 的中的项会由于对称性变为零。因此

$$\mathbb{E}[\boldsymbol{x}] = \boldsymbol{\mu} \tag{2.48}$$

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，可以聚焦在一起组成矩阵 $\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^T]$。这个矩阵可以写成

$$\begin{aligned}\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^T] &= \frac{1}{(2\pi)^{\frac{D}{2}}|\Sigma|^{\frac{1}{2}}}\int \exp\left\{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right\}\boldsymbol{x}\boldsymbol{x}^T\mathrm{d}\boldsymbol{x} \\ &= \frac{1}{(2\pi)^{\frac{D}{2}}|\Sigma|^{\frac{1}{2}}}\int \exp\left\{-\frac{1}{2}z^T\Sigma^{-1}z\right\}(z+\mu)(z+\boldsymbol{\mu})^T\mathrm{d}z\end{aligned} \tag{2.49}$$

其中，我们再次应用了 $z = \boldsymbol{x} - \boldsymbol{\mu}$ 来进行变量替换。注意，涉及到 $\boldsymbol{\mu}z^T$ 和 $z\boldsymbol{\mu}^T$ 的交叉项将再次由于对称性而变为零。项 $\boldsymbol{\mu}\boldsymbol{\mu}^T$ 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。我们再次使用协方差矩阵的特征向量展开，以及特征向量的完备性，得到

$$z = \sum_{j=1}^{D} y_j \boldsymbol{u}_j \tag{2.50}$$

其中 $y_j = \boldsymbol{\mu}_j^T \boldsymbol{z}$，因此

$$\begin{aligned}
&\frac{1}{(2\pi)^{\frac{D}{2}}|\Sigma|^{\frac{1}{2}}} \int \exp\left\{-\frac{1}{2}\boldsymbol{z}^T\Sigma^{-1}\boldsymbol{z}\right\}\boldsymbol{z}\boldsymbol{z}^T \mathrm{d}\boldsymbol{z} \\
&\quad = \frac{1}{(2\pi)^{\frac{D}{2}}|\Sigma|^{\frac{1}{2}}} \sum_{i=1}^{D}\sum_{j=1}^{D} \boldsymbol{\mu}_i\boldsymbol{\mu}_j^T \int \exp\left\{-\sum_{k=1}^{D}\frac{y_k^2}{2\lambda_k}\right\} y_i y_j d\boldsymbol{y} \\
&\quad = \sum_{i=1}^{D} \boldsymbol{\mu}_i\boldsymbol{\mu}_i^T \lambda_i = \Sigma
\end{aligned} \tag{2.51}$$

推导过程中我们使用了特征向量方程，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 $i = j$)。因此我们有

$$\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^T] = \boldsymbol{\mu}\boldsymbol{\mu}^T + \Sigma \tag{2.52}$$

对于高斯分布一元随机变量的方差这一特例，我们可以得到

$$\mathrm{var}[\boldsymbol{x}] = \Sigma \tag{2.53}$$

虽然高斯分布被广泛用作概率密度模型，但是它有着一些巨大的局限性.

1. 自由参数的数量。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。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。
2. 单峰。因此不能够很好地近似多峰分布。

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分布。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。

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 $\boldsymbol{x}$ 是一个服从高斯分布 $\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}, \Sigma)$ 的 D 维向量。我们把 $\boldsymbol{x}$ 划分成两个不相交的子集 x_a, x_b。不失一般性，我们可令 x_a 为 $\boldsymbol{x}$ 的前 M 个分量，令

x_b 为剩余的 $D-M$ 个分量，因此

$$\boldsymbol{x} = \begin{pmatrix} x_a \\ x_b \end{pmatrix} \tag{2.54}$$

$$\boldsymbol{\mu} = \begin{pmatrix} \mu_a \\ \mu_b \end{pmatrix} \tag{2.55}$$

$$\Sigma = \begin{pmatrix} \Sigma_{aa} & \Sigma_{ab} \\ \Sigma_{ba} & \Sigma_{bb} \end{pmatrix} \tag{2.56}$$

$$\tag{2.57}$$

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

定理 2.1

已知 $x \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma), Y = AX + B$ 有如下结论：

1. $\mathbb{E}[Y] = A\mu + B$
2. $\mathrm{var}[Y] = A \cdot \Sigma \cdot A^T$

♡

x_a 可以表示为

$$x_a = \underbrace{(I_m \ o)}_{A} \underbrace{\begin{pmatrix} x_a \\ x_b \end{pmatrix}}_{x} \tag{2.58}$$

$$\mathbb{E}[x_a] = (I_m \ o)(\mu_a \ \mu_b)^T = \mu_a \tag{2.59}$$

$$\mathrm{var}[x_a] = (I_m \ o) \begin{pmatrix} \Sigma_{aa} & \Sigma_{ab} \\ \Sigma_{ba} & \Sigma_{bb} \end{pmatrix} \begin{pmatrix} I_m \\ o \end{pmatrix} = \Sigma_{aa} \tag{2.60}$$

$$x_a \sim \mathcal{N}(\mu_a, \Sigma_{aa}) \tag{2.61}$$

同理

$$\mathbb{E}[x_b] = \mu_b \tag{2.62}$$

$$\mathrm{var}[x_b] = \Sigma_{bb} \tag{2.63}$$

$$x_b \sim \mathcal{N}(\mu_b, \Sigma_{bb}) \tag{2.64}$$

构造

$$x_{b|a} = x_b - \Sigma_{ba}\Sigma_{aa}^{-1}x_a \tag{2.65}$$

$$\mu_{b|a} = \mu_b - \Sigma_{ba}\Sigma_{aa}^{-1}\mu_a \tag{2.66}$$

$$\Sigma_{bb|a} = \Sigma_{bb} - \Sigma_{ba}\Sigma_{aa}^{-1}\Sigma_{ab} \tag{2.67}$$

$$x_{b|a} = \underbrace{(-\Sigma_{ba}\Sigma_{aa}^{-1}\ I_n)}_{A}\underbrace{\begin{pmatrix} x_a \\ x_b \end{pmatrix}}_{x} \tag{2.68}$$

$x_{b|a}$ 的分布为

$$\mathbb{E}[x_{b|a}] = (-\Sigma_{ba}\Sigma_{aa}^{-1}\ I_n)|\begin{pmatrix} \mu_a \\ \mu_b \end{pmatrix} = \mu_{b|a} \tag{2.69}$$

$$\text{var}[x_{b|a}] = (-\Sigma_{ba}\Sigma_{aa}^{-1}\ I_n)\begin{pmatrix} \Sigma_{aa} & \Sigma_{ab} \\ \Sigma_{ba} & \Sigma_{bb} \end{pmatrix}\begin{pmatrix} -\Sigma_{ba}\Sigma_{aa}^{-1} \\ I_n \end{pmatrix} = \Sigma_{bb|a} \tag{2.70}$$

$$p(x_{b|a}) \sim \mathcal{N}(\mu_{b|a}, \Sigma_{bb|a}) \tag{2.71}$$

因为

$$x_b = x_{b|a} + \Sigma_{ba}\Sigma_{aa}^{-1}x_a \tag{2.72}$$

所以

$$\mathbb{E}[x_b|x_a] = \mu_{b|a} + \Sigma_{ba}\Sigma_{aa}^{-1}x_a \tag{2.73}$$

$$\text{var}[x_b|x_a] = var[x_{b|a}] = \Sigma_{bb|a} \tag{2.74}$$

$$p(x_b|x_a) \sim \mathcal{N}(\mu_{b|a}, \Sigma_{bb|a}) \tag{2.75}$$

高斯变量的贝叶斯定理

令边缘概率分布和条件概率分布的形式如下

$$p(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}|\mu, \Lambda^{-1}) \tag{2.76}$$

$$p(\boldsymbol{y}|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{y}|\boldsymbol{A}\boldsymbol{x} + b, \boldsymbol{L}^{-1}) \tag{2.77}$$

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

1. 求 $p(y)$。

$$y = Ax + b + \epsilon, \quad \epsilon \sim \mathcal{N}(0, L^{-1}) \tag{2.78}$$

$$\mathbb{E}[y] = \mathbb{E}(Ax + b) + \mathbb{E}(\epsilon) = A\mu + b \tag{2.79}$$

$$\text{var}[y] = \text{var}[Ax + b + \epsilon] = A\Lambda^{-1}A^T + L^{-1} \tag{2.80}$$

$$y \sim \mathcal{N}(A\mu + b, A\Lambda^{-1}A^T + L^{-1}) \tag{2.81}$$

2. 求 $p(x|y)$。

定义

$$z = \begin{pmatrix} x \\ y \end{pmatrix} \tag{2.82}$$

$$z \sim \mathcal{N}\left(\begin{bmatrix} \mu \\ A\mu + b \end{bmatrix}, \begin{bmatrix} \Lambda^{-1} & \triangle \\ \triangle^T & A\Lambda^{-1}A^T + L^{-1} \end{bmatrix}\right) \tag{2.83}$$

其中

$$\begin{aligned} \triangle &= Cov(x, y) \\ &= E[(x - E(x)) \cdot (y - E(y))^T] \\ &= E[(x - \mu)(y - A\mu - b)^T] \\ &= E[(x - \mu)(Ax + b + \epsilon - A\mu - b)^T] \\ &= E[(x - \mu)(Ax - A\mu)^T + (x - \mu)\epsilon^T] \\ &= E[(x - \mu)(Ax - A\mu)^T] + E[(x - \mu)\epsilon^T] \\ &= E[(x - \mu)(Ax - A\mu)^T] \\ &= E[(x - \mu)(x - \mu)^T] \cdot A^T \\ &= \text{var}[x] \cdot A^T \\ &= \Lambda^{-1}A^T \end{aligned} \tag{2.84}$$

所以

$$\mathbb{E}[x|y] = (\Lambda + A^T LA)^{-1}\{A^T L(y - b) + \Lambda\mu\} \tag{2.85}$$

$$\text{cov}[x|y] = (\Lambda + A^T LA)^{-1} \tag{2.86}$$

高斯分布的最大似然估计

给定一个数据集 $X = (x_1, x_2, \ldots, x_N)^T$，其中观测 $\{x_n\}$ 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。

$$\mu_{ML} = \frac{1}{N}\sum_{n=1}^{N} x_n \tag{2.87}$$

$$\Sigma_{ML} = \frac{1}{N}\sum_{n=1}^{N}(x_n - \mu_{ML})(x_n - \mu_{ML})^T \tag{2.88}$$

$$\mathbb{E}[\mu_{ML}] = \mu \tag{2.89}$$

$$\mathbb{E}[\Sigma_{ML}] = \frac{N-1}{N}\Sigma \tag{2.90}$$

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。对于高斯分布

$$\begin{aligned}\mu_{ML}^{(N)} &= \frac{1}{N}\sum_{n=1}^{N} x_n \\ &= \frac{1}{N}x_N + \frac{1}{N}\sum_{n=1}^{N-1} x_n \\ &= \mu_{ML}^{(N-1)} + \frac{1}{N}(x_N - \mu_{ML}^{(N-1)})\end{aligned} \tag{2.91}$$

随着 N 的增加，后续数据点的贡献也会逐渐变小。我们不能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

$$f(\theta) \equiv \mathbb{E}[z|\theta] = \int z p(z|\theta) dz \tag{2.92}$$

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 给出。我们假设 z 的条件方差是有穷的，因此

$$\mathbb{E}[(z-f)^2|\theta] < \infty \tag{2.93}$$

不失一般性，假设当 $\theta > \theta^*$ 时 $f(\theta) > 0$，当 $\theta < \theta^*$ 时 $f(\theta^*) < 0$。下式定义了一个根 θ^* 的顺序估计的序列

$$\theta^{(N)} = \theta^{(N-1)} - \alpha_{N-1} z(\theta^{(N-1)}) \tag{2.94}$$

$z(\theta^{(N)})$ 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

$$\lim_{N\to\infty} \alpha_N = 0 \tag{2.95}$$

$$\sum_{N=1}^{\infty} \alpha_N = \infty \tag{2.96}$$

$$\sum_{N=1}^{\infty} \alpha_N^2 < \infty \tag{2.97}$$

可以证明由公式给出的顺序估计确实以概率 1 收敛于根。第一个条件确保了后续的修正幅度会逐渐变小，从而这个过程可以收敛于一个极限值。第二个条件用来确保算法不会收

敛不到根的值。第三个条件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

$$\frac{\partial}{\partial\theta}\left\{\frac{1}{N}\sum_{n=1}^{N}-\ln p(x_n|\theta)\right\}\Bigg|_{\theta_{ML}}=0 \tag{2.98}$$

交换导数与求和，取极限 $N\to\infty$，我们有

$$-\lim_{N\to\infty}\frac{1}{N}\sum_{n=1}^{N}\frac{\partial}{\partial\theta}\ln p(x_n|\theta)=\mathbb{E}_x\left[-\frac{\partial}{\partial\theta}\ln p(x|\theta)\right] \tag{2.99}$$

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

$$\theta^{(N)}=\theta^{(N-1)}-\alpha_{N-1}\frac{\partial}{\partial\theta^{(N-1)}}\left[-\ln p(x_N|\theta^{(N-1)})\right] \tag{2.100}$$

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

$$z=-\frac{\partial}{\partial\mu_{ML}}\ln p(x|\mu_{ML},\sigma^2)=-\frac{1}{\sigma^2}(x-\mu_{ML}) \tag{2.101}$$

因此 z 的分布是一个高斯分布，均值为 $-(\mu-\mu_{ML})/\sigma^2$。

高斯分布的贝叶斯推断

最大似然框架给出了对于参数 $\boldsymbol{\mu}$ 和 $\boldsymbol{\Sigma}$ 的点估计。现在我们通过引入高斯分布中的参数的先验分布，介绍一种贝叶斯的方法。

1. σ^2 已知，估计 μ。

   考虑一个一元高斯随机变量 x，我们的任务是从一组 N 次观测 $\boldsymbol{X}=\{x_1,\ldots,x_N\}$ 中推断均值 μ。似然函数，它可以看成 μ 的函数，由下式给出

$$p(\boldsymbol{X}|\mu)=\prod_{n=1}^{N}p(x_n|\mu)=\frac{1}{(2\pi\sigma^2)^{\frac{N}{2}}}\exp\left\{-\frac{1}{2\sigma^2}\sum_{n=1}^{N}(x_n-\mu)^2\right\} \tag{2.102}$$

   先验概率分布为

$$p(\mu)=\mathcal{N}(\mu|\mu_0,\sigma_0^2) \tag{2.103}$$

   从而后验概率为

$$p(\mu|\boldsymbol{X})\propto p(\boldsymbol{X}|\mu)p(\mu) \tag{2.104}$$

可以证明后验概率的形式为

$$p(\mu|\boldsymbol{X}) = \mathcal{N}(\mu|\mu_N, \sigma_N^2) \tag{2.105}$$

其中

$$\mu_N = \frac{\sigma^2}{N\sigma_)^2 + \sigma^2}\mu_0 + \frac{N\sigma_0^2}{N\sigma_0^2 + \sigma^2}\mu_{ML} \tag{2.106}$$

$$\frac{1}{\sigma_N^2} = \frac{1}{\sigma_0^2} + \frac{N}{\sigma^2} \tag{2.107}$$

其中 μ_{ML} 是 μ 的最大似然解，由样本均值给出

$$\mu_{ML} = \frac{1}{N}\sum_{n=1}^{N} x_n \tag{2.108}$$

首先，我们注意到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 $N = 0$，那么就变成了先验均值。对于 $N \to \infty$，后验均值由最大似然解给出。另外，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。如果数据点的数量 $N \to \infty$，方差 σ_N^2 趋于零，从而后验分布在最大似然解附近变成了无限大的尖峰。

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年，那么贝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

$$p(\mu|\boldsymbol{X}) \propto \left[p(\mu)\prod_{n=1}^{N-1} p(x_n|\mu)\right] p(x_N|\mu) \tag{2.109}$$

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。

2. μ 已知，估计 σ^2。

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

$$p(\boldsymbol{X}|\lambda) = \prod_{n=1}^{N} \mathcal{N}(x_n|\mu, \lambda^{-1}) \propto \lambda^{\frac{N}{2}} \exp\left\{-\frac{\lambda}{2}\sum_{n=1}^{N}(x_n - \mu)^2\right\} \tag{2.110}$$

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

$$\text{Gam}(\lambda|a,b) = \frac{1}{\Gamma(a)} b^a \lambda^{a-1} \exp(-b\lambda) \tag{2.111}$$

Gamma 分布的均值和方差为

$$\mathbb{E}[\lambda] = \frac{a}{b} \tag{2.112}$$

$$\text{var}[\lambda] = \frac{a}{b^2} \tag{2.113}$$

考虑一个先验分布 $\text{Gam}(\lambda|a_0, b_0)$。那么我们得到后验分布

$$p(\lambda|\boldsymbol{X}) =\propto \lambda^{a_0-1}\lambda^{\frac{N}{2}} \exp\left\{-b_0\lambda - \frac{\lambda}{2} - \frac{\lambda}{2}\sum_{n=1}^{N}(x_n-\mu)^2\right\} \tag{2.114}$$

我们可以把它看成形式为 $\text{Gam}(\lambda|a_N, b_N)$ 的 Gamma 分布，其中

$$a_N = a_0 + \frac{N}{2} \tag{2.115}$$

$$b_N = b_0 + \frac{1}{2}\sum_{n=1}^{N}(x_n-\mu)^2 = b_0 + \frac{N}{2}\sigma_{ML}^2 \tag{2.116}$$

其中 σ_{ML}^2 是方差的最大似然估计。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，我们看到 N 个数据点对参数 b 贡献了 $\frac{\sigma_{ML}^2}{2}$，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。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。

3. μ 和 σ_2 都未知。

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

$$\begin{aligned} p(\boldsymbol{X}|\mu,\lambda) &= \prod_{n=1}^{N}\left(\frac{\lambda}{2\pi}\right)^{\frac{1}{2}} \exp\left\{-\frac{\lambda}{2}(x_n-\mu)^2\right\} \\ &\propto \left[\lambda^{\frac{1}{2}}\exp\left(-\frac{\lambda\mu^2}{2}\right)\right]^N \exp\left\{\lambda\mu\sum_{n=1}^{N}x_n - \frac{\lambda}{2}\sum_{n=1}^{N}x_n^2\right\} \end{aligned} \tag{2.117}$$

我们现在想找到一个先验分布 $p(\mu,\lambda)$，它对于 μ 和 λ 的依赖与依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

$$\begin{aligned} p(\mu,\lambda) &\propto \left[\lambda^{\frac{1}{2}}\exp\left(-\frac{\lambda\mu^2}{2}\right)\right]^\beta \exp\{c\lambda\mu - d\lambda\} \\ &= \exp\left\{-\frac{\beta\lambda}{2}(\mu-\frac{c}{\beta})^2\right\}\lambda^{\frac{\beta}{2}}\exp\left\{-\left(d-\frac{c^2}{2\beta}\right)\lambda\right\} \end{aligned} \tag{2.118}$$

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

$$p(\mu, \lambda) = \mathcal{N}(\mu|\mu_0, (\beta\lambda)^{-1})\text{Gam}(\lambda|a, b) \tag{2.119}$$

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 2.119 被称为正态-Gamma 分布或者高斯-Gamma 分布。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 $\boldsymbol{x}$ 的多元高斯分布，假设精度已知，则均值 $\boldsymbol{\mu}$ 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 $\boldsymbol{\Lambda}$ 的情形，共轭先验是 Wishart 分布。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，共轭先验为正态-Wishart 分布。

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

$$\begin{aligned} p(x|\mu, a, b) &= \int_0^\infty \mathcal{N}(\mu, \tau^{-1})\text{Gam}(\tau|a, b)\text{d}\tau \\ &= \int_0^\infty \frac{b^a \exp(-br)\tau^{a-1}}{\Gamma(a)} \left(\frac{\tau}{2\pi}\right)^{\frac{1}{2}} \exp\left\{-\frac{\tau}{2}(x-\mu)^2\right\} \text{d}\tau \\ &= \frac{b^a}{\Gamma(a)} \left(\frac{1}{2}\right)^{\frac{1}{2}} \left[b + \frac{(x-\mu)^2}{2}\right]^{-a-\frac{1}{2}} \Gamma(a + \frac{1}{2}) \end{aligned} \tag{2.120}$$

我们定义新的参数 $v = 2a$ 和 $\lambda = \frac{a}{b}$。使用新的参数，分布 $p(x|\mu a, b)$ 的形式为

$$p(x|\mu\lambda, v) = \frac{\Gamma(\frac{v}{2} + \frac{1}{2})}{\Gamma(\frac{v}{2})} \left(\frac{\lambda}{\pi v}\right)^{\frac{1}{2}} \left[1 + \frac{\lambda(x-\mu)^2}{v}\right]^{-\frac{v}{2}-\frac{1}{2}} \tag{2.121}$$

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 v 被称为自由度 (degrees of freedom)。对于 $v = 1$ 的情况，t 分布变成了柯西分布 (Cauchy distribution)，而在极限 $v \to \infty$ 的情况下，t 分布变成了高斯分布，均值为 μ，精度为 λ。

根据公式 2.120，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。这个分布通常有着比高斯分布更长的“尾巴”，这给出了 t 分布的一个重要性质：鲁棒性 (robustness)。

周期变量

高斯分布在实际应用中非常重要，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。

这种变量使用极坐标 $0 \leqslant \theta \leqslant 2\pi$ 表示，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点，因此可以被描述为一个二维单位向量 $x_1, \ldots, x_N$，其中 $\|x_n\| = 1$ 且 $n = 1, \ldots, N$。我们可以对向量 $\{x_n\}$，可得

$$\bar{x} = \frac{1}{N} \sum_{n=1}^{N} x_n \tag{2.122}$$

然后找到这个平均值对应的角度 $\bar{\theta}$。$\bar{x}$ 通常位于单位圆的内部。这个观测在笛卡尔坐标系下为 $x_n = (\cos\theta_n, \sin\theta_n)$，把样本均值写成笛卡尔坐标系，并令两个分量相等，可得

$$\bar{\theta} = \tan^{-1}\left\{\frac{\sum_n \sin\theta_n}{\sum_n \cos\theta_n}\right\} \tag{2.123}$$

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。

现在考虑高斯分布对于周期变量的一个推广：von Mises 分布。假设考虑二维高斯分布沿着一个固定的半径的圆周的值。之后通过构造，这个分布将会具有周期性。

$$p(\theta|\theta_0, m) = \frac{1}{2\pi I_0(m)} \exp\{m\cos(\theta - \theta_0)\} \tag{2.124}$$

这被称为 von Mises 分布，或者环形正态分布 (circular normal)。

为了完整性，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。另一种方法类似于 von Mises 分布，都是首先考虑欧几里德空间的高斯分布。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。最后一种方法的思想是，在实数轴上的任何合法的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。

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。通过将更基本的概率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)。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。考虑 K 个高斯概率密度的叠加，形式为

$$p(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \Sigma_k) \tag{2.125}$$

这被称为混合高斯 (mixture of Gaussians)。其中

$$\sum_{k=1}^{K} \pi_k = 1,\ 0 \leqslant \pi_k \leqslant 1 \tag{2.126}$$

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 $\mathcal{N}(x|\mu_k, \Sigma_k) = p(x|k)$ 看成以 k 为条件的 x 的概率。后验概率 $p(k|x)$ 起着一个重要作用，· 经也被称为责任 (responsibilities)。

2.4 指数族分布

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

$$p(x|\eta) = h(x)g(\eta)\exp\{\eta^T u(x)\} \tag{2.127}$$

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)，$u(x)$ 是 x 的某个函数。函数 $g(\eta)$ 可以被看成系数，它确保了概率分布是归一化的，因此满足

$$\int p(x|\eta)dx = 1 \tag{2.128}$$

下面看一些例子

1. 伯努利分布

$$\begin{aligned} p(x|\mu) &= \text{Bern}(x|\mu) = \mu^x(1-\mu)^{1-x} \\ &= \exp\{x\ln\mu + (1-x)\ln(1-\mu)\} \\ &= (1-\mu)\exp\left\{\ln\left(\frac{\mu}{1-\mu}\right)x\right\} \end{aligned} \tag{2.129}$$

与公式 2.127 比较，可以看出

$$\eta = \ln\left(\frac{\mu}{1-\mu}\right) \tag{2.130}$$

从中可以解出 η，得到 $\mu = \sigma(\eta)$，其中

$$\sigma(\eta) = \frac{1}{1+\exp(-\eta)} \tag{2.131}$$

被称为 logistic sigmoid 函数。因此可以使用公式 2.127 给出的标准形式把伯努利分布写成下面的形式

$$p(x|\mu) = \sigma(-\eta)\exp(\eta x) \tag{2.132}$$

2. 单一观测 x 的多项式分布

$$
\begin{aligned}
p(x|\mu) &= \prod_{k=1}^{M} \mu_k^{x_k} = \exp\left\{\sum_{k=1}^{M} x_k \ln \mu_k\right\} \\
\mu(x) &= x \\
h(x) &= 1 \\
g(\eta) &= 1
\end{aligned}
\tag{2.133}
$$

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

$$
\sum_{k=1}^{M} \mu_k = 1 \tag{2.134}
$$

因此给定任意 M-1 个参数 μ_k，剩下的参数就固定了。使用这个限制，这种表达方式下多项式分布变成了

$$
\begin{aligned}
&\exp\left\{\sum_{k=1}^{M} x_k \ln \mu_k\right\} \\
&= \exp\left\{\sum_{k=1}^{M-1} x_k \ln \mu_k + \left(1 - \sum_{k=1}^{M-1} x_k\right) \ln\left(1 - \sum_{k=1}^{M-1} \mu_k\right)\right\} \\
&= \exp\left\{\sum_{k=1}^{M-1} x_k \ln\left(\frac{\mu_k}{1 - \sum_{j=1}^{M-1} \mu_k}\right) + \ln\left(1 - \sum_{k=1}^{M-1} \mu_k\right)\right\}
\end{aligned}
\tag{2.135}
$$

令

$$
\ln\left(\frac{\mu_k}{1 - \sum_j \mu_j}\right) = \eta_k \tag{2.136}
$$

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

$$
\mu_k = \frac{\exp(\eta_k)}{1 + \sum_j exp(\eta_j)} \tag{2.137}
$$

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

$$
p(x|\eta) = \left(1 + \sum_{k=1}^{M-1} \exp(\mu_k)\right)^{-1} \exp(\mu^T x) \tag{2.138}
$$

这是指数族分布的标准形式，其中参数向量 $\eta = (\eta_1, \ldots, \eta_{M-1}, 0)^T$。在这个指数族分

布中

$$\mu(x) = x \tag{2.139}$$

$$h(x) = 1 \tag{2.140}$$

$$g(\eta) = \left(1 + \sum_{k=1}^{M-1} \exp(\eta_k)\right)^{-1} \tag{2.141}$$

3. 高斯分布

$$\begin{aligned} p(x|\mu, \sigma^2) &= \frac{1}{(2\pi\sigma^2)^{\frac{1}{2}}} \exp\left\{-\frac{1}{2\sigma^2}(x-\mu)^2\right\} \\ &= \frac{1}{(2\pi\sigma^2)^{\frac{1}{2}}} \exp\left\{-\frac{1}{2\sigma^2}x^2 + \frac{\mu}{\sigma^2}x - \frac{1}{2\sigma^2}\mu^2\right\} \end{aligned} \tag{2.142}$$

经过简单的推导后，它可以转化为公式 2.127 给出的标准指数族分布的形式，其中

$$\eta = \begin{pmatrix} \frac{\mu}{\sigma^2} \\ \frac{-1}{2\sigma^2} \end{pmatrix} \tag{2.143}$$

$$\mu(x) = \begin{pmatrix} x \\ x^2 \end{pmatrix} \tag{2.144}$$

$$h(x) = (2\pi)^{\frac{1}{2}} \tag{2.145}$$

$$g(\eta) = (-2\eta_2)^{\frac{1}{2}} \exp\left(\frac{\eta_1^2}{4\eta_2^2}\right) \tag{2.146}$$

最大似然与充分统计量

用最大似然法估计公式 2.127 给出的一般形式的指数族分布的参数向量 μ 的问题。

$$\int h(x)g(\eta)\exp\{\eta^T u(x)\} = 1 \tag{2.147}$$

对上式的两侧关于 μ 取梯度，我们有

$$\begin{aligned} &\nabla g(\eta) \underbrace{\int h(x)\exp\{\eta^T u(x)\}dx}_{1/g(\eta)} + g(\eta)\int h(x)\exp\{\eta^T u(x)\}u(x)dx = 0 \\ &\Rightarrow -\frac{1}{g(\eta)}\nabla g(\eta) = \int \underbrace{g(\eta)h(x)\exp\{\eta^T u(x)\}}_{p(x|\eta)} u(x)dx = \mathbb{E}[\mu(x)] \\ &\Rightarrow -\nabla \ln g(\eta) = \mathbb{E}[u(x)] \end{aligned} \tag{2.148}$$

同理，$u(x)$ **的协方差，可以根据** $g(\eta)$ **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 $X = \{x_1, \ldots, x_N\}$。对于这个

数据集，似然函数为

$$p(X|\eta) = \left(\prod_{n=1}^{N} h(x_n)\right) g(\eta)^N \exp\left\{\eta^T \sum_{n=1}^{N} u(x_n)\right\} \tag{2.149}$$

令 $\ln p(X|\eta)$ 关于 η 的导数等于零，我们可以得到最大似然估计 μ_{ML} 满足的条件

$$-\nabla \ln g(\eta) = \frac{1}{N} \sum_{n=1}^{N} u(x_n) \tag{2.150}$$

原则上可以通过这个方程来得到 μ_{ML}。我们看到最大似然估计的解只通过 $\sum_n u(x_n)$ 对数据产生依赖，因此这个量被称为分布的充分统计量 (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。

共轭先验

对于指数族分布的任何成员，都存在一个共轭先验，可以写成下面的形式

$$p(\eta|\varkappa, v) = f(\varkappa, v) g(\eta)^v \exp\{v\eta^T \varkappa\} \tag{2.151}$$

其中 $f(\varkappa, v)$ 是归一化系数。为了证明这个确实是共轭先验，让我们反先验分布与似然函数相乘，得到后验概率，形式为

$$p(\eta|\boldsymbol{X}, \varkappa, v) \propto g(\eta)^{v+N} \exp\left\{\eta^T \left(\sum_{n=1}^{N} u(x_n) + v\varkappa\right)\right\} \tag{2.152}$$

这与先验分布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 v 可以看成先验分布中假想观测的有效观测数。给定 $\varkappa$ 的情况下，每个假想观测都对充分统计量 $u(x)$ 的值有贡献。

无信息先验

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。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 $p(\lambda) = C$ 常数作为一个合适的先验分布。在连续参数的情况下，这种方法有两个潜在的困难。第一个是，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)。第二个困难产生于概率非线性变量的概率密度的变换。

这里我们考虑无信息先验的两个简单的例子。

1. 如果概率密度的形式为

$$p(x|\mu) = f(x - \mu) \tag{2.153}$$

那么参数被称为位置参数 (location parameter)。这一类概率分布具有平移不变性 (translation invariance)。

2. 考虑概率分布的形式为

$$p(x|\sigma) = \frac{1}{\sigma} f\left(\frac{x}{\sigma}\right) \tag{2.154}$$

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)。

2.5 非参数化方法

使用一些非参数化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。

直方图法

标准的直方图简单地把 x 划分成不同的宽度为 $\triangle_i$ 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 $\triangle_i$,得到每个箱子的概率的值

$$p_i = \frac{n_i}{N\triangle_i} \tag{2.155}$$

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。直方图的另一个主要的局限性是维数放大。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。

1. 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领域内的数据点。
2. 为了获得好的结果,平滑参数的值既不能太大也不能太小。

核密度估计

假设观测服从 D 维空间的某个未知的概率密度分布 $p(x)$。把这个 D 维空间选择成欧几里德空间,并且我们想估计 $p(x)$ 的值。根据以前对于局部性的讨论,让我们考虑包含 x 的某个小区域 $\mathcal{R}$。这个区域的概率质量为

$$P = \int_{\mathcal{R}} p(x)dx \tag{2.156}$$

现在我们假设收集了服从 $p(x)$ 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

$$\text{Bin}(K|N,P) = \frac{N!}{K!(N-K)!}P^K(1-P)^{N-K} \tag{2.157}$$

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，以此为均值的概率分布的方差为 $var[\frac{K}{N}] = \frac{P(1-P)}{N}$。对于大的 N 值，这个分布将会在均值附近产生尖峰，并且

$$K \simeq NP \tag{2.158}$$

$$P \simeq p(x)V \tag{2.159}$$

$$p(x) = \frac{K}{NV} \tag{2.160}$$

上式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这个结果。

1. 固定 K 然后从数据中确定 V 的值，这就是 K 近邻方法。
2. 固定 V 然后从数据中确定 K 的值，这就是核方法。

在极限 $N \to \infty$ 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。

把区域 $\mathcal{R}$ 取成以 x 为中心的小超立方体，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

$$k(u) = \begin{cases} 1, & |u_i| \leqslant \frac{1}{2}, i = 1, \ldots, D \\ 0, & \text{其它情况} \end{cases} \tag{2.161}$$

这表示一个以原点为中心的单位立方体。函数 $k(u)$ 是核函数的一个例子，在这个问题中也被称为 Parzen 窗 (parzen window)。根据公式 2.161，如果数据点 x_n 位于以 x 为中心的边长为 h 的立方体中，那么量 $k(\frac{x-x_n}{h})$ 的值等于 1，否则它的值为 0. 于是

$$p(x) = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{h^D}k\left(\frac{x-x_n}{h}\right) \tag{2.162}$$

这个函数表述为以 N 个数据点 x_n 为中心的 N 个立方体。

核密度估计有人为带来的非连续性的问题。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。

$$p(x) = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{(2\pi h^2)^{\frac{D}{2}}}exp\left\{-\frac{\|x-x_n\|^2}{2h^2}\right\} \tag{2.163}$$

h 表示高斯分布的标准差。因此我们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。

这种估计方法有一个很大的优点，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。

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。与之不同，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 2.161 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。

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 2.160 提供了与每个类别关联的一个概率密度的估计

$$p(x|C_k) = \frac{K_k}{N_k V} \tag{2.164}$$

类似地，无条件概率密度为

$$p(x) = \frac{K}{NV} \tag{2.165}$$

而类先验为

$$p(C_k) = \frac{N_k}{N} \tag{2.166}$$

由贝叶斯定理，可以得到类别的后验概率

$$p(C_k|x) = \frac{p(x|C_k)p(C_k)}{p(x)} = \frac{K_k}{K} \tag{2.167}$$

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 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数量最多的点的类别。

最近邻 (K=1) 分类器的一个有趣的性质是在极限 $N \to \infty$ 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布器) 可以达到的最小错误率的二倍。

第 3 章 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。

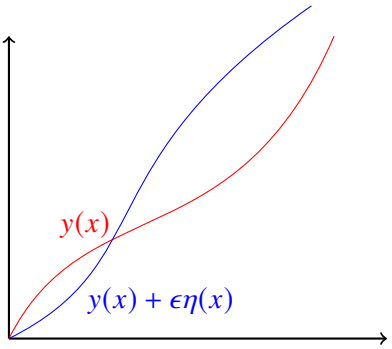
按照微积分规则，我们在求传统的导数 $\frac{\mathrm{d}y}{\mathrm{d}x}$ 时，我们可以首先让变量 x 产生一个小的改变 ϵ，然后对 ϵ 进行幂级数展开，即

$$y(x+\epsilon) = y(x) + \frac{\mathrm{d}y}{\mathrm{d}x}\epsilon + O(\epsilon^2) \tag{3.1}$$

最后取极限 $\epsilon \to 0$。类似地，对于一个多变量函数 $y(x_1, \ldots, x_D)$，对应的偏导数通过下式定义

$$y(x_1+\epsilon_1, \ldots, x_D+\epsilon_D) = y(x_1, \ldots, x_D) + \sum_{i=1}^{D} \frac{\partial y}{\partial x_i}\epsilon_i + O(\epsilon^2) \tag{3.2}$$

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 的一个信息任意的函数) 时，我们考虑泛函 $F[y]$ 的变化，如图所示



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

$$F[y(x)+\epsilon\eta(x)] = F[y(x)] + \epsilon \int \frac{\delta F}{\delta y(x)}\eta(x)dx + O(\epsilon^2) \tag{3.3}$$

这可以被看成公式 3.1 的一个自然推广，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

$$\int \frac{\delta F}{\delta y(x)} \eta(x) dx = 0 \tag{3.4}$$

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 $\hat{x}$f 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

$$F[y] = \int G(y(x), y'(x), x) dx \tag{3.5}$$

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

$$F[y(x) + \epsilon\eta(x)] = F[y(x)] + \epsilon \int \left\{ \frac{\partial G}{\partial y} \eta(x) + \frac{\partial G}{\partial y'} \eta'(x) \right\} dx + O(\epsilon^2) \tag{3.6}$$

我们现在必须把它转化为公式 3.3 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

$$F[y(x) + \epsilon\eta(x)] = F[y(x)] + \epsilon \int \left\{ \frac{\partial G}{\partial y} - \frac{d}{dx} \left(\frac{\partial G}{\partial y'} \right) \right\} \eta(x) dx + O(\epsilon^2) \tag{3.7}$$

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

$$\frac{\partial G}{\partial y} - \frac{d}{dx} \left(\frac{\partial G}{\partial y'} \right) = 0 \tag{3.8}$$

这被称为欧拉-拉格朗日方程 (Euler-Lagrange equation)。例如，如果

$$G = y(x)^2 + (y'(x))^2 \tag{3.9}$$

那么，欧拉-拉格朗日方程的形式为

$$y(x) - \frac{d^2 y}{dx^2} = 0 \tag{3.10}$$

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 $\boldsymbol{x}$ 上的扩展是很直接的。

第 4 章 矩阵的性质

4.1 矩阵的基本性质

矩阵 A 的第 i 行和第 j 列的元素为 A_{ij}。我们用 I_N 表示 $N \times N$ 的单位矩阵。在没有歧义的情形下，我们简单地记作 I。转置矩阵 A^T 的元素为 $(A^T)_{ij} = A_{ji}$。根据转置的定义，我们有

$$(AB)^T = B^T A^T \tag{4.1}$$

A 的逆矩阵，记作 A^{-1}，满足

$$AA^{-1} = A^{-1}A = I \tag{4.2}$$

由于 $ABB^{-1}A^{-1} = I$，我们有

$$(AB)^{-1} = B^{-1}A^{-1} \tag{4.3}$$

我们还有

$$(A^T)^{-1} = (A^{-1})^T \tag{4.4}$$

关于矩阵的逆矩阵，下面这个恒等式很有用

$$(P^{-1} + B^T R^{-1} B)^{-1} B^T R^{-1} = PB^T (BPB^T + R)^{-1} \tag{4.5}$$

两侧同时右乘 $(BPB^T + R)$，很容易证明上式的正确性。假设 P 的维度为 $N \times N$，而 R 的维度为 $M \times M$，从而 B 的维度为 $M \times N$。这样，如果 $M \ll N$，那么估计公式 4.5 的右侧所花费的低价就远远小于估计左侧的代价。经常出现的一种情况是

$$(I + AB)^{-1} A = A(I + BA)^{-1} \tag{4.6}$$

另一个与矩阵的逆矩阵相背的有用的恒等式为

$$(A + BD^{-1}C)^{-1} = A^{-1} - A^{-1}B(D + CA^{-1}B)^{-1}CA^{-1} \tag{4.7}$$

这被称为 Woodbury 恒等式。将两侧同时乘以 $(A + BD^{-1}C)$ 即可证明。例如，假设 A 是一个很大的对角矩阵 (因此很容易求逆矩阵)，B 的行数很多列数很少 (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 $\{a_1, \ldots, a_N\}$ 被称为线性相关 (linearly independent) 如果关系 $\sum_n a_n \boldsymbol{a}_n = 0$ 只在所有 $a_n = 0$ 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)。

4.2 迹和行列式

迹和行列式适用于方阵。矩阵 A 的迹 $\mathrm{Tr}(A)$ 被定义为主对角线上元素的和。我们可以看到

$$\mathrm{Tr}(AB) = \mathrm{Tr}(BA) \tag{4.8}$$

通过多次把这个公式应用到三个矩阵的乘积上，我们看到

$$\mathrm{Tr}(ABC) = \mathrm{Tr}(CAB) = \mathrm{Tr}(BCA) \tag{4.9}$$

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 $|A|$ 定义为

$$|A| = \sum (\pm 1) A_{1i_1} A_{2i_2} \dots A_{Ni_N} \tag{4.10}$$

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 $i_1 \dots i_N$ 是大奇排列还是偶排列。注意 $|I| = 1$，因此对于一个 2×2 矩阵，行列式的形式为

$$|A| = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = a_{11}a_{22} - a_{12}a_{21} \tag{4.11}$$

两个矩阵乘积的行列式为

$$|AB| = |A||B| \tag{4.12}$$

此外，矩阵的逆矩阵的行列式为

$$|A^{-1}| = \frac{1}{|A|} \tag{4.13}$$

如果 A 和 B 是 $N \times M$ 的矩阵，那么

$$|I_N + AB^T| = |I_M + A^T B| \tag{4.14}$$

一种特殊情况是

$$|I_N + ab^T| = 1 + a^T b \tag{4.15}$$

其中 a 和 b 是 N 维列向量。

4.3 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 $\boldsymbol{a}$ 关于标量 x 的导数本身是一个向量，它的分量为

$$\left(\frac{\partial \boldsymbol{a}}{\partial x}\right)_i = \frac{\partial a_i}{\partial x} \tag{4.16}$$

矩阵的导数的定义与些类似。关于向量和矩阵的导数也可以被定义。例如

$$\left(\frac{\partial x}{\partial \boldsymbol{a}}\right)_i = \frac{\partial x}{\partial a_i} \tag{4.17}$$

类似地

$$\left(\frac{\partial a}{\partial b}\right)_{ij} = \frac{\partial a_i}{\partial b_j} \tag{4.18}$$

写出矩阵的各个元素,下面的性质很容易证明

$$\frac{\partial}{\partial \boldsymbol{x}}(\boldsymbol{x}^T\boldsymbol{a}) = \frac{\partial}{\partial \boldsymbol{x}}(\boldsymbol{a}^T\boldsymbol{x}) = \boldsymbol{a} \tag{4.19}$$

类似地

$$\frac{\partial}{\partial \boldsymbol{x}}(AB) = \frac{\partial A}{\partial \boldsymbol{x}}B + A\frac{\partial B}{\partial \boldsymbol{x}} \tag{4.20}$$

矩阵的逆矩阵的导数可以表示为

$$\frac{\partial}{\partial x}(A^{-1}) = -A^{-1}\frac{\partial A}{\partial x}A^{-1} \tag{4.21}$$

使用公式 4.20 对方程 $A^{-1}A = I$ 求微分,然后右乘 A^{-1} 即可证明。并且

$$\frac{\partial}{\partial x}\ln|A| = \mathrm{Tr}\left(A^{-1}\frac{\partial A}{\partial x}\right) \tag{4.22}$$

如果我们把 x 选成 A 中的元素,那么我们有

$$\frac{\partial}{\partial A_{ij}}\mathrm{Tr}(AB) = B_{ji} \tag{4.23}$$

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

$$\frac{\partial}{\partial A}\mathrm{Tr}(AB) = B^T \tag{4.24}$$

使用这种记号,我们有下列性质

$$\frac{\partial}{\partial A}\mathrm{Tr}(A^TB) = B \tag{4.25}$$

$$\frac{\partial}{\partial A}\mathrm{Tr}(A) = I \tag{4.26}$$

$$\frac{\partial}{\partial A}\mathrm{Tr}(ABA^T) = A(B + B^T) \tag{4.27}$$

这些也可以通过写出矩阵下标的方式证明出。我们也有

$$\frac{\partial}{\partial A}\ln|A| = (A^{-1})^T \tag{4.28}$$

4.4 特征向量方程

对于一个 $M \times M$ 的方阵 A，特征向量方程的定义为

$$\boldsymbol{A}\boldsymbol{\mu}_i = \lambda_i \boldsymbol{\mu}_i \tag{4.29}$$

其中 $i = 1, \ldots, M$，$\boldsymbol{\mu}_i$ 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，角存在的条件为

$$|A - \lambda_i I| = 0 \tag{4.30}$$

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)。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。对称矩阵的性质为 $A_{ij} = A_{ji}$ 或者等价地，$A = A^T$。对称矩阵的逆矩阵也是对称的。将 $A^T A = I$ 取转置，然后使用 $AA^{-1} = I$ 以及 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 4.29 左乘 $(\boldsymbol{\mu}_i^*)^T$，其中 $*$ 表示复共轭，我们可以得到

$$(\boldsymbol{\mu}_i^*)^T \boldsymbol{A}\boldsymbol{\mu}_i = \lambda_i (\boldsymbol{\mu}_i^*)^T \boldsymbol{\mu}_i \tag{4.31}$$

之后，我们对公式 4.29 取复共轭，然后左乘 $\boldsymbol{\mu}_i^T$，可得

$$\boldsymbol{\mu}_i^T \boldsymbol{A}\boldsymbol{\mu}_i^* = \lambda_i^* \boldsymbol{\mu}_i^T \boldsymbol{\mu}_i^* \tag{4.32}$$

推导过程中，我们使用了 $A^* = A$，因为我们只考虑实对称矩阵 A。将第二个方程取转置，使用 $A^T = A$，我们看到两个方程在左侧相同，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 $\boldsymbol{\mu}_i$ 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

$$\boldsymbol{\mu}_i^T \boldsymbol{\mu}_j = I_{ij} \tag{4.33}$$

其中 I_{ij} 是单位矩阵 I 的元素。为了证明这一点，我们首先将公式 4.29 左乘 $\boldsymbol{\mu}_j^T$，得到

$$\boldsymbol{\mu}_j^T \boldsymbol{A}\boldsymbol{\mu}_i = \lambda_i \boldsymbol{\mu}_j^T \boldsymbol{\mu}_i \tag{4.34}$$

因此，通过交换下标，我们有

$$\boldsymbol{\mu}_i^T \boldsymbol{A}\boldsymbol{\mu}_j = \lambda_j \boldsymbol{\mu}_i^T \boldsymbol{\mu}_j \tag{4.35}$$

我们现在对第二个方程取转置，使用对称性质 $\boldsymbol{A}^T = \boldsymbol{A}$，然后将两个方程相减，可得

$$(\lambda_i - \lambda_j) \boldsymbol{\mu}_i^T \boldsymbol{\mu}_j = 0 \tag{4.36}$$

因此，对于 $\lambda_i \neq \lambda_j$，我们有 $\boldsymbol{\mu}_i^T \boldsymbol{\mu}_j = 0$，因此 $\boldsymbol{\mu}_i$ 和 $\boldsymbol{\mu}_j$ 是正交的。如果两个特征值是相等的，那么任意线性组合 $\alpha\boldsymbol{\mu}_i + \beta\boldsymbol{\mu}_j$ 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 $\boldsymbol{\mu}_i$ 是 $M \times M$ 的矩阵 $\boldsymbol{U}$，根据单位正交性，我们有

$$\boldsymbol{U}^T \boldsymbol{U} = \boldsymbol{I} \tag{4.37}$$

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 $\boldsymbol{U}\boldsymbol{U}^T = \boldsymbol{I}$。为了证明这一点，我们注意到，公式 4.37 表明 $\boldsymbol{U}^T\boldsymbol{U}\boldsymbol{U}^{-1} = \boldsymbol{U}^{-1} = \boldsymbol{U}^T$，因此 $\boldsymbol{U}\boldsymbol{U}^{-1} = \boldsymbol{U}\boldsymbol{U}^T = \boldsymbol{I}$。使用公式 4.12，也可以看出 $|U| = 1$。

特征向量方程 4.29 可以使用 $\boldsymbol{U}$ 表示成下面的形式

$$\boldsymbol{A}\boldsymbol{U} = \boldsymbol{U}\boldsymbol{\Lambda} \tag{4.38}$$

其中 $\boldsymbol{\Lambda}$ 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 $\boldsymbol{x}$，它经过正交矩阵 $\boldsymbol{U}$ 进行变换，得到新向量

$$\tilde{\boldsymbol{x}} = \boldsymbol{U}\boldsymbol{x} \tag{4.39}$$

变换前后向量的长度不变，因为

$$\tilde{\boldsymbol{x}}^T \tilde{\boldsymbol{x}} = \boldsymbol{x}^T \boldsymbol{U}^T \boldsymbol{U} \boldsymbol{x} = \boldsymbol{x}^T \boldsymbol{x} \tag{4.40}$$

类似地，任意两个向量的角度在变换前后也不变，因为

$$\tilde{\boldsymbol{x}}^T \tilde{\boldsymbol{y}} = \boldsymbol{x}^T \boldsymbol{U}^T \boldsymbol{U} \boldsymbol{y} = \boldsymbol{x}^T \boldsymbol{y} \tag{4.41}$$

因此，乘以 $\boldsymbol{U}$ 可以表示为坐标第的刚性旋转.

根据公式 4.38 可得

$$\boldsymbol{U}^T \boldsymbol{A} \boldsymbol{U} = \boldsymbol{\Lambda} \tag{4.42}$$

因为 $\boldsymbol{\Lambda}$ 是对角矩阵，我们说矩阵 $\boldsymbol{A}$ 被矩阵 $\boldsymbol{U}$ 对角化 (diagonalised)。如果我们左乘 $\boldsymbol{U}$ 然后右乘 $\boldsymbol{U}^T$，我们有

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T \tag{4.43}$$

取这个方程的逆，然后使用公式 4.3 以及 $\boldsymbol{U}^{-1} = \boldsymbol{U}^T$，我们有

$$\boldsymbol{A}^{-1} = \boldsymbol{U}\boldsymbol{\Lambda}^{-1}\boldsymbol{U}^T \tag{4.44}$$

最后两个方程也可以写成

$$\boldsymbol{A} = \sum_{i=1}^{M} \lambda_i \boldsymbol{\mu}_i \boldsymbol{\mu}_i^T \tag{4.45}$$

$$\boldsymbol{A}^{-1} = \sum_{i=1}^{M} \frac{1}{\lambda_i} \boldsymbol{\mu}_i \boldsymbol{\mu}_i^T \tag{4.46}$$

如果我们取公式 4.43 的行列式，然后使用公式 4.12，我们有

$$|\boldsymbol{A}| = \prod_{i=1}^{M} \lambda_i \tag{4.47}$$

类似地，取公式 4.43 的迹，使用迹运算的循环性以及 $\boldsymbol{U}^T\boldsymbol{U}^T = \boldsymbol{I}$，我们有

$$\mathrm{Tr}(\boldsymbol{A}) = \sum_{i=1}^{M} \lambda_i \tag{4.48}$$

一个矩阵 $\boldsymbol{A}$ 被称为正定的 (positive definite)，记作 $\boldsymbol{A} > 0$，如果对于向量 $\boldsymbol{w}$ 的所有非零值都有 $\boldsymbol{w}^T\boldsymbol{A}\boldsymbol{w} > 0$。等价地，一个正定矩阵的所有特征值都有 $\lambda_i > 0$。令 $\boldsymbol{w}$ 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

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \tag{4.49}$$

的特征值 $\lambda_1 \simeq 5.37$ 且 $\lambda_2 \simeq -0.37$。一个矩阵被称为半正定的 (positive semidefinite)，如果对于 $\boldsymbol{w}$ 的所有值都有 $\boldsymbol{w}^T\boldsymbol{A}\boldsymbol{w} \geqslant 0$，记作 $\boldsymbol{A} \succeq 0$。它等价于 $\lambda_i \geqslant 0$。

第 5 章 信息论

信息量可以被看成在学习 x 的值的时候的“惊讶程度”。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 $p(x)$ 对数有关。因此，我们有

$$h(x) = -\log_2 p(x) \tag{5.1}$$

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 5.1 关于概率分布 $p(x)$ 的期望得到。这个期望值为

$$H[x] = -\sum_x p(x) \log_2 p(x) \tag{5.2}$$

这个重要的量被叫做随机变量 x 的熵 (entropy)。无噪声编码定理 (noiseless coding theorem) 表明，熵是传输一个随机变量状态值所需的比特位的下界。

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。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。然而，我们不想区分每个箱子内部物体的重新排列。有第 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

$$W = \frac{N!}{\prod_i n_i!} \tag{5.3}$$

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

$$H = \frac{1}{N} \ln W = \frac{1}{N} \ln N! = \frac{1}{N} \sum_i \ln n_i! \tag{5.4}$$

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

$$\ln N! \simeq N \ln N - N \tag{5.5}$$

可以得到

$$H = -\lim_{N\to\infty} \sum_i \left(\frac{n_i}{N}\right) \ln\left(\frac{n_i}{N}\right) = -\sum_i p_i \ln p_i \tag{5.6}$$

推导时我们使用了 $\sum_i n_i = N$。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)。

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

$$\tilde{H} = -\sum_i p(x_i) \ln p(x_i) + \lambda \left(\sum_i p(x_i) - 1\right) \tag{5.7}$$

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。

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 $\triangle$ 的箱子。然后假设 $p(x)$ 是连续的。

$$H[x] = \lim_{\triangle\to 0} -\sum_i p(x_i) \ln p(x_i)\,\{\} = -\int p(x) \ln p(x) dx \tag{5.8}$$

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 $\ln \triangle$，这在极限 $\triangle \to 0$ 的情形下发散。这反映同一个事实：具体化一个连续变量需要大量的比特位。

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。为了让最大值有一个合理一定义，有必要限制 $p(x)$ 的一阶矩和二阶矩，同时要保留归一化的限制。因此我们要优化下面的关于 $p(x)$ 的函数

$$\begin{aligned} -\int_{-\infty}^{+\infty} p(x) \ln p(x) dx + \lambda_1 \left(-\int_{-\infty}^{+\infty} p(x) - 1\right) + \lambda_2 \left(-\int_{-\infty}^{+\infty} x p(x) dx - \mu\right) \\ +\lambda_3 \left(-\int_{-\infty}^{+\infty} (x-\mu)^2 p(x) dx - \sigma^2\right) \end{aligned} \tag{5.9}$$

使用变分法中，令这个函数的导数等于零，我们有

$$p(x) = \exp\{-1 + \lambda_1 + \lambda_2 x + \lambda_3 (x-\mu)^2\} \tag{5.10}$$

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

$$p(x) = \frac{1}{(2\pi\sigma^2)^{\frac{1}{2}}} \exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\} \tag{5.11}$$

因此最大化微分熵的分布是高斯分布。求解高斯分布的微分熵

$$
\begin{aligned}
H[x] &= -\int \frac{1}{(2\pi\sigma^2)^{\frac{1}{2}}} \exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\} \ln \frac{1}{(2\pi\sigma^2)^{\frac{1}{2}}} \exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\} dx \\
&= -\frac{1}{(2\pi\sigma^2)^{\frac{1}{2}}} \int \exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\} \left(-\ln(\sqrt{2\pi}\sigma) - \frac{(x-\mu)^2}{2\sigma^2}\right) dx \\
&= \frac{\ln(\sqrt{2\pi}\sigma}{(2\pi\sigma^2)^{\frac{1}{2}}} \int \exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\} dx + \frac{1}{(2\pi\sigma^2)^{\frac{1}{2}}} \int \exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\} \frac{(x-\mu)^2}{2\sigma^2} dx \\
&= \frac{\ln(\sqrt{2\pi}\sigma}{(2\pi\sigma^2)^{\frac{1}{2}}} \sqrt{2}\sigma \int \exp\left\{-\left(\frac{x-\mu}{\sqrt{2\sigma}}\right)^2\right\} d\left(\frac{x-\mu}{\sqrt{2\sigma}}\right) \\
&\quad + \frac{1}{(2\pi\sigma^2)^{\frac{1}{2}}} \sqrt{2}\sigma \int \exp\left\{-\left(\frac{x-\mu}{\sqrt{2\sigma}}\right)^2\right\} \frac{(x-\mu)^2}{2\sigma^2} d\left(\frac{x-\mu}{\sqrt{2\sigma}}\right) \\
&= \frac{\ln(\sqrt{2\pi}\sigma)}{\sqrt{\pi}} \int_{-\infty}^{+\infty} \exp(-y^2) dy + \frac{1}{\sqrt{\pi}} \int_{-\infty}^{+\infty} \exp(-y^2) y^2 dy \\
&= \ln(\sqrt{2\pi}\sigma) + \frac{1}{\sqrt{\pi}} \cdot -\frac{1}{2} \left(0 - \int_{-\infty}^{+\infty} \exp(-y^2) dy\right) \\
&= \ln(\sqrt{2\pi}\sigma) + \frac{1}{2} \\
&= \frac{1}{2} \left(\ln(2\pi\sigma^2) + 1\right)
\end{aligned}
\tag{5.12}
$$

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

$$
H[y|x] = -\iint p(y, x) \ln p(y|x) dy dx \tag{5.13}
$$

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

$$
H[x, y] = H[y|x] + H[x] \tag{5.14}
$$

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 自己抽需的信息，加上给定 x 的情况下具体化 y 所需的额外信息。

相对熵和互信息

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。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为

$$\begin{aligned}\mathrm{KL}(p\|q) &= -\int p(x)\ln q(x)dx - \left(-\int p(x)\ln p(x)dx\right) \\ &= -\int p(x)\ln\left\{\frac{q(x)}{p(x)}\right\}dx\end{aligned} \tag{5.15}$$

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。注意这不是一个对称量，即 $\mathrm{KL}(p\|q) \neq \mathrm{KL}(q\|p)$

$$\mathrm{KL}(p\|q) = -\int p(x)\ln\left\{\frac{q(x)}{p(x)}\right\}dx \geqslant -\ln\int q(x)dx = 0 \tag{5.16}$$

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 $\int q(x)dx = 1$ 和 Jensen 不等式。

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。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 $n = 1,\ldots,N$。那么，关于 $p(x)$ 的期望就可以通过这些点的有限加和，使用公式来近似

$$\mathrm{KL}(p\|q) \simeq \frac{1}{N}\sum_{n=1}^{N}\{-\ln q(x_n|\theta) + \ln p(x_n)\} \tag{5.17}$$

公式右侧的第二项与 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

$$\begin{aligned}I[x,y] &\equiv \mathrm{KL}(p(x,)\|p(x)p(y)) \\ &= -\iint p(x,y)\ln\left(\frac{p(x)p(y)}{p(x,y)}\right)dxdy\end{aligned} \tag{5.18}$$

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 $I[x,y] \geqslant 0$，当且仅当 x 和 y 相互独立时等号成立。互信息和条件熵之

间的关系为

$$I[x, y] = H[x] - H[x|y] = H[y] - H[y|x] \tag{5.19}$$

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 (反这亦然)。从贝叶斯的观点来年, 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。

第 6 章 回归的线性模型

目前为止，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 $\boldsymbol{x}$ 的情况下，预测一个或者多个连续目标 (target) 变量 t 的值。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，我们也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模型，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。

最简单的方法是，直接建立一个适当的函数 $y(\boldsymbol{x})$，对于新的输入 $\boldsymbol{x}$，这个函数能够直接给出对应的 t 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 $p(t|\boldsymbol{x})$ 建模，因为经表达了对于每个 $\boldsymbol{x}$ 的值，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 $\boldsymbol{x}$ 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的损失函数的期望值。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。

6.1 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

$$y(\boldsymbol{x}, \boldsymbol{w}) = w_0 + w_1 x_1 + \ldots w_D x_D$$

其中 $x = (x_1, \ldots, x_D)^T$。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 $w_0, \ldots, w_D$ 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，这给模型带来了极大的局限性。因此，我们扩展模型的类别：将输入变量的固定的非线性函数进行线性组合，形式为

$$y(\boldsymbol{x}, \boldsymbol{w}) = w_0 + \sum_{y=1}^{M-1} w_j \phi_j(\boldsymbol{x}) = \sum_{j=0}^{M-1} w_j \phi_j(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{\Phi}(\boldsymbol{x})$$

其中 $\phi_j(\boldsymbol{x})$ 被称为基函数 (basis function)。

1. 高斯基函数

$$\phi_j(x) = exp\left\{-\frac{(x-\mu_j)^2}{2s^2}\right\}$$

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。

2. sigmoid 基函数

$$\phi_j(x) = \frac{1}{1 + \exp\left(\frac{x-\mu_j}{s}\right)}$$

3. 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。在许多信息处理的应用中，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。

最大似然与最小平方

最小化平方和误差函数可以看成高斯噪声模型的假设下的最大似然解。假设目标变量 t 由确定的函数 $y(x, w)$ 给出，这个函数被附加了高斯噪声，即

$$t = y(\boldsymbol{x}, \boldsymbol{w}) + \epsilon \tag{6.1}$$

其中 ϵ 是一个零均值的高斯随机变量，精度为 β。因此我们有

$$p(t|\boldsymbol{x}, \boldsymbol{w}, \beta) = \mathcal{N}(t|y(\boldsymbol{x}, \boldsymbol{w}), \beta^{-1}) \tag{6.2}$$

最优的预测由目标变量的条件给出

$$\mathbb{E}[t|\boldsymbol{x}] = \int tp(t|\boldsymbol{x})dt = y(\boldsymbol{x}, \boldsymbol{w}) \tag{6.3}$$

现在考虑一个输入数据集 $X = \{x_1, \ldots, x_N\}$，对应的目标值为 $t_1, \ldots, t_N$。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 t) 不同。假设这些数据点是独立同分布的。那么可以得到下面的似然函数

$$p(\boldsymbol{t}|\boldsymbol{X}, \boldsymbol{w}, \beta) = \prod_{n=1}^{N} \mathcal{N}(t_n|\boldsymbol{w}^T \phi(x_n), \beta^{-1}) \tag{6.4}$$

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 x 总会出现在条件变量的位置上。为了保持记号的简洁性，不显示地写出 x。取似然函数的对数，我们有

$$\begin{aligned} \ln p(\boldsymbol{t}|\boldsymbol{w}, \beta) &= \sum_{n=1}^{N} \ln \mathcal{N}(t_n|\boldsymbol{w}^T \phi(x_n), \beta^{-1}) \\ &= \frac{N}{2} \ln \beta - \frac{N}{2} \ln(2\pi) - \beta E_D(\boldsymbol{w}) \end{aligned} \tag{6.5}$$

其中平方和误差函数的定义为

$$\begin{aligned} E_D(\boldsymbol{w}) &= \frac{1}{2}\sum_{n=1}^{N}\{t_n - \boldsymbol{w}^T\phi(x_n)\}^2 \\ &= \frac{1}{2}(\boldsymbol{w}^T\phi(x_1) - t_1 \ \ldots \ \boldsymbol{w}^T\phi(x_N) - t_N)\begin{pmatrix} \boldsymbol{w}^T\phi(x_1) - t_1 \\ \vdots \\ \boldsymbol{w}^T\phi(x_N) - t_N \end{pmatrix} \\ &= \frac{1}{2}(\boldsymbol{w}^T(\phi(x_1) \ \ldots \ \phi(x_N)) - (t_1 \ \ldots \ t_N))\left(\begin{pmatrix} \boldsymbol{w}^T\phi(x_1) \\ \vdots \\ \boldsymbol{w}^T\phi(x_N) \end{pmatrix} - \begin{pmatrix} t_1 \\ \vdots \\ t_N \end{pmatrix}\right) \\ &= \frac{1}{2}(\boldsymbol{w}^T\Phi^T - \boldsymbol{t}^T)(\Phi\boldsymbol{w} - \boldsymbol{t}) \\ &= \frac{1}{2}(\boldsymbol{w}^T\Phi^T\Phi\boldsymbol{w} - 2\boldsymbol{w}^T\Phi^T\boldsymbol{t} + \boldsymbol{t}^T\boldsymbol{t}) \end{aligned} \tag{6.6}$$

使用最大似然方法确定 $\boldsymbol{w}$ 和 β。首先关于 $\boldsymbol{w}$ 求最大值。

$$\begin{aligned} \nabla \ln p(\boldsymbol{t}|\boldsymbol{w},\beta) &= \beta\sum_{n=1}^{N}\{t_n - \boldsymbol{w}^T\phi(x_n)\}\phi(x_n)^T \\ &= \Phi^T\Phi\boldsymbol{w} - \Phi^T\boldsymbol{t} \end{aligned} \tag{6.7}$$

令梯度为零，可得

$$\boldsymbol{w} = (\Phi^T\Phi)^{-1}\Phi^T\boldsymbol{t} \tag{6.8}$$

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 $\Phi_{nj} = \phi_j(x_n)$，即

$$\begin{pmatrix} \Phi_0(x_1) & \Phi_0(x_1) & \ldots & \Phi_{M-1}(x_1) \\ \Phi_0(x_2) & \Phi_0(x_2) & \ldots & \Phi_{M-1}(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ \Phi_0(x_N) & \Phi_0(x_N) & \ldots & \Phi_{M-1}(x_N) \end{pmatrix} \tag{6.9}$$

量

$$\Phi^\dagger \equiv (\Phi^T\Phi)^{-1}\Phi^T \tag{6.10}$$

被称为矩阵 Φ 的 Moore-Penrose 伪逆矩阵 (pseudo-inverse matrix)。它可以被看成逆矩阵的概率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 $\Phi^\dagger \equiv \Phi^{-1}$

关于噪声精度参数 β 最大化似然函数，结果为

$$\frac{1}{\beta_{ML}} = \frac{1}{N}\sum_{n=1}^{N}\{t_n - \boldsymbol{w}_{ML}^T\phi(x_n)\}^2 \tag{6.11}$$

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。

最小平方的几何描述

平方和误差函数就等于 $\boldsymbol{y}$ 和 $\boldsymbol{t}$ 之间的平方欧氏距离 (只相差一个因子 $\frac{1}{2}$)。因此，$\boldsymbol{w}$ 的最小平方解对应于位于子空间 S 的与 $\boldsymbol{t}$ 最近的 $\boldsymbol{y}$ 的选择。这个解对应于 $\boldsymbol{t}$ 在子空间 S 上的正交投影。

补充一个图

顺序学习

最大似然解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。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 $\boldsymbol{w}$

$$\boldsymbol{w}^{(\tau+1)} = \boldsymbol{w}^{(\tau)} - \eta \nabla E_n \tag{6.12}$$

其中 τ 表示迭代次数，η 是学习率参数。对于平方和误差函数的情形，我们有

$$\boldsymbol{w}^{(\tau+1)} = \boldsymbol{w}^{(\tau)} + \eta(t_n - \boldsymbol{w}^{(\tau)T}\boldsymbol{\phi}_n)\boldsymbol{\phi}_n \tag{6.13}$$

其中 $\boldsymbol{\phi}_n = \boldsymbol{\phi}(\boldsymbol{x}_n)$。这被称为最小均方 (least-mean-squares) 或者 LMS 算法。

正则化最小平方

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

$$E_D(\boldsymbol{w}) + \lambda E_W(\boldsymbol{w}) \tag{6.14}$$

L1 正则化引起稀疏解的多种解释。

1. 用图解释：L2 正则相当于用圆去逼近目标，而 L1 正则相当于用菱形去逼近目标，所以更容易引起交点在坐标轴上即得到稀疏解。
2. 从导数角度解释：L2 正则无法将目标函数的极值点拉拢到稀疏解上，而 L1 正则因为 L1 导函数的特殊性从而可以在一定范围内将极值点直接拉拢到稀疏解上。
3. 从先验概率分布角度解释：L2 正则相当于假设参数是服从高斯分布的，而 L1 正则相当于假设了参数是服从拉普拉斯分布的，自然拉普拉斯分布比高斯分布更集中在

0 这个点上。

多个输出

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焦起来，记作目标向量 $\boldsymbol{t}$。这个问题可以这样解决：对于 $\boldsymbol{t}$ 的每个分量，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

$$\boldsymbol{y}(\boldsymbol{x}, \boldsymbol{w}) = \boldsymbol{W}^T \boldsymbol{\phi}(\boldsymbol{x}) \tag{6.15}$$

其中 $\boldsymbol{y}$ 是一个 K 维列向量，$\boldsymbol{W}$ 是一个 $M \times K$ 的参数矩阵，$\boldsymbol{\phi}(\boldsymbol{x})$ 是一个 M 维列向量，每个元素为 $\phi_j(\boldsymbol{x})$。

6.2 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。如果使用有限规模的数据集来训练复杂的模型，那么使用最大似然法，或者等价地使用最小平方法，会导致严重的过拟合问题。正如前面所说，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。从贝叶斯观点讨论模型的复杂度之前，从频率学家的观点考虑一下模型的复杂度的问题——偏置-方差折中 (bias-variance trade-off)。

最优的预测 (变分法可求出) 由条件期望 (记作 $h(x)$) 给出，即

$$h(\boldsymbol{x}) = \mathbb{E}[t|\boldsymbol{x}] = \int t p(t|\boldsymbol{x}) \mathrm{d}t \tag{6.16}$$

平方损失函数的期望可以写成

$$\begin{aligned} \mathbb{E}[L] &= \iint \{y(x) - h(x)\}^2 p(x,t) dx dt \\ &= \int \{y(x) - h(x)\}^2 p(x) dx + \iint \{h(x) - t\}^2 p(x,t) dx dt \end{aligned} \tag{6.17}$$

与 $y(\boldsymbol{x})$ 无关的第二项，是由数据本身的噪声造成的，表明期望损失能够达到的最小值。第一项与我们对函数 $y(\boldsymbol{x})$ 的选择有关，我们要找一个 $y(\boldsymbol{x})$ 的解，使得这一项最小。在实际应用中，我们的数据集 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 $h(\boldsymbol{x})$。

如果我们使用由参数向量 $\boldsymbol{w}$ 控制的函数 $y(\boldsymbol{x}, \boldsymbol{w})$ 对 $h(\boldsymbol{x})$ 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 $\boldsymbol{w}$ 的概率分布来表示的。但是，频率学家的方法涉及到根据数据集 D 对 $\boldsymbol{w}$ 进行点估计，然后度着通过下面的思想实验来表示估计的不确定性。

考虑第一项的被积函数，对于一个特定的数据集 D，它的形式为

$$\{y(x;D)-h(x)\}^2 \tag{6.18}$$

由于这个量与特定的数据集 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 $\mathbb{E}_D[y(x;D)]$，然后展开，我们有

$$\begin{aligned}&\{y(x;D)-\mathbb{E}_D[y(x;D)]+\mathbb{E}_D[y(x;D)]-h(x)\}^2\\&\quad=\{y(x;D)-\mathbb{E}_D[y(x;D)]\}^2\\&\quad+\{\mathbb{E}_D[y(x;D)]-h(x)\}^2\\&\quad+2\{y(x;D)-\mathbb{E}_D[y(x;D)]\}\{\mathbb{E}_D[y(x;D)]-h(x)\}\end{aligned} \tag{6.19}$$

现在关于 D 求期望，然后注意到**最后一项等于零**，可得

$$\begin{aligned}&\mathbb{E}_D[\{y(x;D)-h(x)\}^2]\\&\quad=\underbrace{\mathbb{E}_D[y(x;D)]-h(x)\}^2}_{(\text{偏置})^2}+\underbrace{\mathbb{E}_D[\{y(x;D)-\mathbb{E}_D[y(x;D)]\}^2]}_{\text{方差}}\end{aligned} \tag{6.20}$$

我们看到，$y(\boldsymbol{x};D)$ 与回归函数 $h(\boldsymbol{x})$ 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 $y(\boldsymbol{x};D)$ 对于特定的数据集的选择的敏感程度。

将式 6.20 代入式 6.17 中，就得到了对于期望平方损失的分解

$$\text{期望损失}=\text{偏置}^2+\text{方差}+\text{噪声} \tag{6.21}$$

其中

$$\text{偏置}^2=\int\{\mathbb{E}_D[y(x;D)]-h(x)\}^2p(x)dx \tag{6.22}$$

$$\text{方差}=\int\mathbb{E}_D[\{y(x;D)-\mathbb{E}_D[y(x;D)]\}^2]p(x)dx \tag{6.23}$$

$$\text{噪声}=\iint\{h(x)-t\}^2p(x,t)dxdt \tag{6.24}$$

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。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是在偏置和方差之前取得最优的平衡的模型。

6.3 贝叶斯线性回归

使用最大似然方法设置线性回归模型的参数时，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，但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。

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。把噪声精度参数 β 当做已知常数。对应的共轭先验是高斯分布。

$$p(w) \equiv \mathcal{N}(w|\boldsymbol{m}_0, \boldsymbol{S}_0) \tag{6.25}$$

均值为 $\boldsymbol{m}_0$ 协方差为 $\boldsymbol{S}_0$ 后验分布

$$p(w|\boldsymbol{t}) = \mathcal{N}(w|\boldsymbol{m}_N, \boldsymbol{S}_N) \tag{6.26}$$

其中，

$$\boldsymbol{m}_N = \boldsymbol{S}_N(\boldsymbol{S}_0^{-1} + \beta\Phi^T\boldsymbol{t}) \tag{6.27}$$

$$\boldsymbol{S}_N^{-1} = \boldsymbol{S}_0^{-1} + \beta\Phi^T\Phi \tag{6.28}$$

为了简化起见，考虑高斯先验的一个特定的形式——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

$$p(\boldsymbol{w}|\alpha) = \mathcal{N}(\boldsymbol{w}|0, \alpha^{-1}I) \tag{6.29}$$

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 w 的函数，形式为

$$\ln p(\boldsymbol{w}|\boldsymbol{t}) - \frac{\beta}{2}\sum_{n=1}^{N}\{t_n - \boldsymbol{w}^T\phi(x_n)\}^2 - \frac{\alpha}{2}\boldsymbol{w}^T\boldsymbol{w} + 常数 \tag{6.30}$$

于是，后验分布关于 $\boldsymbol{w}$ 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。

补充一个直线拟合的例子图示

预测分布

在实际应用中，我们通常感兴趣的不是 $\boldsymbol{w}$ 本身的值，而是对于新的 $\boldsymbol{x}$ 值预测出 t 的值。这需要计算出预测分布 (predictive distribution)，定义为

$$p(t|\boldsymbol{t},\alpha,\beta)=\int p(t|\boldsymbol{w},\beta)p(\boldsymbol{w}|\boldsymbol{t},\alpha,\beta)d\boldsymbol{w} \tag{6.31}$$

其中 $\boldsymbol{t}$ 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。预测分布的形式为

$$p(t|\boldsymbol{x},\boldsymbol{t},\alpha,\beta)=\mathcal{N}(t|\boldsymbol{m}_N^T\phi(\boldsymbol{x}),\sigma_N^2(\boldsymbol{x})) \tag{6.32}$$

其中预测分布的方差 $\sigma_N^2(\boldsymbol{x})$ 为

$$\sigma_N^2(\boldsymbol{x})=\frac{1}{\beta}+\phi(\boldsymbol{x})^T\boldsymbol{S}_N\phi(\boldsymbol{x}) \tag{6.33}$$

公式 6.33 的第一项表示数据中的噪声，而第二项反映了与参数 $\boldsymbol{w}$ 关联的不确定性。由于噪声和 $\boldsymbol{w}$ 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 $\sigma_{N+1}^2(\boldsymbol{x})\leqslant\sigma_N^2(\boldsymbol{x})$ 在极限 $N\to\infty$ 的情况下，公式 6.33 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。

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 6.33 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 $\boldsymbol{w}$ 和 β 都被当成未知的，我们可以引入一个由高斯-Gamma 分布定义的共轭先验分布 $p(\boldsymbol{w},\beta)$。在这种情况下，预测分布是一个学生 t 分布。

等价核

公式 6.27 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。如果把公式 6.27 代入线性基函数模型中，可以写成下面的形式

$$\begin{aligned}y(\boldsymbol{x},\boldsymbol{m}_N)&=\boldsymbol{m}_N^T\phi(\boldsymbol{x})\\&=\beta\phi(\boldsymbol{x})^T\boldsymbol{S}_N\Phi^T\boldsymbol{t}\\&=\sum_{n=1}^{N}\underline{\beta\phi(\boldsymbol{x})^T\boldsymbol{S}_N\phi(\boldsymbol{x}_n)}t_n\\&=\sum_{n=1}^{N}k(\boldsymbol{x},\boldsymbol{x}_n)t_n\end{aligned} \tag{6.34}$$

可以看成在点 $\boldsymbol{x}$ 处的预测均值由训练集目标变量 t_n 的线性组合给出。函数 $k(x, x_n)$ 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)。核函数 $k(x, x')$ 给出了 x 与 x' 的函数关系。可以看到，x 处的预测分布的均值 $y(x, \boldsymbol{m}_N)$ 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。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。

考虑 $y(\boldsymbol{x})$ 和 $y(\boldsymbol{x}')$ 的协方差

$$\begin{aligned} cov[y(\boldsymbol{x}), y(\boldsymbol{x}')] &= cov[\boldsymbol{\phi}(\boldsymbol{x})^T \boldsymbol{w}, \boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x}')] \\ &= \boldsymbol{\phi}(\boldsymbol{x})^T \boldsymbol{S}_N \boldsymbol{\phi}(\boldsymbol{x}') \\ &= \beta^{-1} k(\boldsymbol{x}, \boldsymbol{x}') \end{aligned} \tag{6.35}$$

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处，相关性就较低。

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 $\boldsymbol{x}$ 做预测。这就引入了用于回归问题 (以及分类问题) 的一个很实用的框架，被称为高斯过程 (Gaussian process)。

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 $\boldsymbol{x}$ 值做预测。可以证明这些权值的和等于 1，即

$$\sum_{n=1}^{N} k(\boldsymbol{x}, \boldsymbol{x}') = 1 \tag{6.36}$$

对于所有的 $\boldsymbol{x}$ 值都成立。同时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 $\psi(\boldsymbol{x})$ 和内积的形式，即

$$k(\boldsymbol{x}, \boldsymbol{z}) = \boldsymbol{\phi}(\boldsymbol{x})^T \psi(\boldsymbol{z}) \tag{6.37}$$

6.4 贝叶斯模型比较

前面介绍了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，从贝叶斯的角度考虑模型选择的问题。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 $\{M_i\}, i = 1, 2, \ldots, L$。这里，一个模型指的是观测数据 D 上的概率分布。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 $p(M_i)$ 表示。给定一个数据集 D，我们想估计后验分布

$$p(M_i|D) \propto p(M_i)p(D|M_i) \tag{6.38}$$

先验分布让我们能够表达不同模型之间的优先级。**我们简单的假设所有的模型都有相同的先验概率**。模型证据 (model evidence)$p(D|M_i)$，表达了数据展现出的不同模型的优先级。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 $\frac{p(D|M_i)}{p(D|M_j)}$ 被称为贝叶斯因子 (Bayes factor)。

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

$$p(t|\boldsymbol{x}, D) = \sum_{i=1}^{L} p(t|\boldsymbol{x}, M_i, D)p(M_i|D) \tag{6.39}$$

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 $p(t|\boldsymbol{x}, M_i, D)$ 求加权平均，权值为这些模型的后验概率 $p(M_i|D)$。

对模型求平均的一个简单的近似是使用最可能的一个模型做预测。这被称为模型选择 (model selection)。

对于一个由参数 $\boldsymbol{w}$ 控制的模型，根据概率的加和规则和乘积规则，模型证据为

$$p(D|M_i) = \int p(D|\boldsymbol{w}, M_i)p(\boldsymbol{w}|M_i)d\boldsymbol{w} \tag{6.40}$$

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 D 的概率，这个模型的参数是从先验分布中随机取样的。同时，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

$$p(\boldsymbol{w}|D, M_i) = \frac{p(D|\boldsymbol{w}, M_i)p(\boldsymbol{w}|M_i)}{p(D|M_i)} \tag{6.41}$$

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 $p(D|w)p(w)$，其中，为了简化记号，我们省略了它对于模型 M_i 的依赖。如果假设后验分布在最大似然值 w_{MAP} 附近是一个尖峰，宽度为 $\triangle w_{后验}$，那么可以用被积函数的值乘以尖峰的宽度来近似这个积分。进一步假设先验分布是平的，宽度为 $\triangle w_{先验}$，即 $p(w) = \frac{1}{\triangle w_{先验}}$，那么我们有

$$p(D) = \int p(D|w)p(w)dw \simeq p(D|w_{MAP})\frac{\triangle w_{后验}}{\triangle w_{先验}} \tag{6.42}$$

取对数可得

$$\ln p(D) \simeq \ln p(D|w_{MAP}) + \ln\left(\frac{\triangle w_{后验}}{\triangle w_{先验}}\right) \tag{6.43}$$

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 $\Delta w_{后验} < \Delta w_{先验}$，因此这一项为负，并且随着 $\frac{\Delta w_{后验}}{\Delta w_{先验}}$ 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 $\frac{\Delta w_{后验}}{\Delta w_{先验}}$ 都相同，我们有

$$\ln p(D) \simeq \ln p(D|w_{MAP}) + M \ln \left(\frac{\Delta w_{后验}}{\Delta w_{先验}} \right) \tag{6.44}$$

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。

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 M_1, M_2。其中真实的概率分布对应于模型 M_1。对于给定的有限的数据集，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

$$\int p(D|M_1) \ln \frac{p(D|M_1)}{p(D|M_2)} \mathrm{d}D \tag{6.45}$$

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，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。

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。

6.5 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。这里讨论一种近似方法。这种方法中，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，或者被称为第二类最大似然，或者被称为推广的最大似然。在机器学习的文献中，这种方法也被称为证据近似 (evidence approximation)。

如果引入 α 和 β 上的超先验分布，那么预测分布可以通过对 w, α, β 求积分的方法得到

$$p(t|\boldsymbol{t}) = \iiint p(t|\boldsymbol{w}, \beta)p(\boldsymbol{w}|\boldsymbol{t}, \alpha, \beta)p(\alpha, \beta|\boldsymbol{t})d\boldsymbol{w}\,d\alpha\,d\beta \tag{6.46}$$

其中 $p(t|\boldsymbol{w}, \beta)$ 由公式 6.2 给出，$p(\boldsymbol{w}|\boldsymbol{t}, \alpha, \beta)$ 由公式 6.26。这里，为了让记号简洁，我们省略了对于输入变量 $\boldsymbol{x}$ 的依赖关系。如果后验分布 $p(\alpha, \beta|\boldsymbol{t})$ 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 $\hat{\beta}$

$$p(t|\boldsymbol{t}) \simeq p(t|\boldsymbol{t}, \hat{\alpha}, \hat{\beta}) = \int p(t|\boldsymbol{w}, \hat{\beta})p(\boldsymbol{w}|\boldsymbol{t}, \hat{\alpha}, \hat{\beta})d\boldsymbol{w} \tag{6.47}$$

根据贝叶斯定理，α 和 β 的后验分布为

$$p(\alpha, \beta|\boldsymbol{t}) \propto p(\boldsymbol{t}|\alpha, \beta)p(\alpha, \beta) \tag{6.48}$$

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 $p(\boldsymbol{t}|\alpha, \beta)$ 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。

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。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。将会在后面章节中讨论，那里我们还会证明这两种方法会收敛到同一个解。

计算证据函数

边缘似然函数 $p(\boldsymbol{t}|\alpha, \beta)$ 是通过对权值参数 $\boldsymbol{w}$ 进行积分得到的，即

$$p(\boldsymbol{t}|\alpha, \beta) = \int p(\boldsymbol{t}|\boldsymbol{w}, \beta)p(\boldsymbol{w}|\alpha)d\boldsymbol{w} \tag{6.49}$$

一种计算这个积分的方法是使用线性-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 6.5 和公式 6.30，我们可以把证据函数写成下面的形式

$$p(\boldsymbol{t}|\alpha, \beta) = \left(\frac{\beta}{2\pi}\right)^{\frac{N}{2}} \left(\frac{\alpha}{2\pi}\right)^{\frac{N}{2}} \int \exp\{-E(\boldsymbol{w})\}\mathrm{d}\boldsymbol{w} \tag{6.50}$$

其中 M 是 $\boldsymbol{w}$ 的维数，并且，我们定义了

$$\begin{aligned} E(\boldsymbol{w}) &= \beta E_D(\boldsymbol{w}) + \alpha E_W(\boldsymbol{w}) \\ &= \frac{\beta}{2}\|\boldsymbol{t} - \Phi\boldsymbol{w}\|^2 + \frac{\alpha}{2}\boldsymbol{w}^T\boldsymbol{w} \end{aligned} \tag{6.51}$$

我们看到，如果忽略一些比例常数，公式 6.51 等于正则化的平方和误差函数。我们现在对

$\boldsymbol{w}$ 配平方，可得

$$E(\boldsymbol{w}) = E(\boldsymbol{m}_N) + \frac{1}{2}(\boldsymbol{w} - \boldsymbol{m}_N)^T \boldsymbol{A}(\boldsymbol{w} - \boldsymbol{m}_N) \tag{6.52}$$

其中我们令

$$\boldsymbol{A} = \alpha \boldsymbol{I} + \beta \Phi^T \Phi \tag{6.53}$$

注意 $\boldsymbol{A}$ 对应于误差函数的二阶导数

$$\boldsymbol{A} = \nabla\nabla E(\boldsymbol{w}) \tag{6.54}$$

被称为 Hessian 矩阵。这里我们也定义了 $\boldsymbol{m}_N$ 为

$$\boldsymbol{m}_N = \beta \boldsymbol{A}^{-1} \Phi^T \boldsymbol{t} \tag{6.55}$$

使用公式 6.28，我们看到 $\boldsymbol{A} = \boldsymbol{S}_N^{-1}$，因此公式 6.55 等价于之前的定义 6.27，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 $\boldsymbol{w}$ 的积分现在可以很容易地计算出来了，即

$$\begin{aligned} &\int \exp\{-E(\boldsymbol{w})\} \mathrm{d}\boldsymbol{w} \\ &= \exp\{-E(\boldsymbol{m}_N)\} \int \exp\left\{-\frac{1}{2}(\boldsymbol{w} - \boldsymbol{m}_N)^T \boldsymbol{A}(\boldsymbol{w} - \boldsymbol{m}_N)\right\} \mathrm{d}\boldsymbol{w} \\ &= \exp\{-E(\boldsymbol{m}_N)\}(2\pi)^{\frac{M}{2}} |A|^{-\frac{1}{2}} \end{aligned} \tag{6.56}$$

使用公式 6.50，我们可以把边缘似然函数的对数写成下面的形式

$$\ln p(\boldsymbol{t}|\alpha, \beta) = \frac{M}{2} \ln \alpha + \frac{N}{2} \ln \beta - E(\boldsymbol{m}_N) - \frac{1}{2} \ln |\boldsymbol{A}| - \frac{N}{2} \ln(2\pi) \tag{6.57}$$

这就是证据函数的表达式。

最大化证据函数

让我们首先考虑 $p(\boldsymbol{t}|\alpha, \beta)$ 关于 α 的最大化。首先定义下面的特征向量方程

$$(\beta \Phi^T \Phi)\boldsymbol{u}_i = \lambda_i \boldsymbol{u}_i \tag{6.58}$$

可知 $\boldsymbol{A}$ 的特征值为 $\alpha + \lambda_i$。现在考虑公式 6.57 中涉及到 $\ln |\boldsymbol{A}|$ 的项关于 α 的导数

$$\frac{\mathrm{d}}{\mathrm{d}\alpha} \ln |\boldsymbol{A}| = \frac{\mathrm{d}}{\mathrm{d}\alpha} \ln \prod_i (\lambda_i + \alpha) = \frac{\mathrm{d}}{\mathrm{d}\alpha} \sum_i \ln(\lambda_i + \alpha) = \sum_i \frac{1}{\lambda_i + \alpha} \tag{6.59}$$

因此函数 6.57 关于 α 的驻点满足

$$0 = \frac{M}{2\alpha} - \frac{1}{2} \boldsymbol{m}_N^T \boldsymbol{m}_N - \frac{1}{2} \sum_i \frac{1}{\lambda_i + \alpha} \tag{6.60}$$

两侧乘以 2α，整理，可得

$$\alpha \boldsymbol{m}_N^T \boldsymbol{m}_N = M - \alpha \sum_i \frac{1}{\lambda_i + \alpha} = \gamma \tag{6.61}$$

由于 i 的求和式中一共有 M 项，因此 γ 可以写成

$$\gamma = \sum_i \frac{\lambda_i}{\lambda_i + \alpha} \tag{6.62}$$

我们看到最大化边缘似然函数的 α 满足

$$\alpha = \frac{\gamma}{\boldsymbol{m}_N^T \boldsymbol{m}_N} \tag{6.63}$$

注意，这是 α 的一个隐式解，不仅因为 γ 与 α 有关，还因为后验概率本身的众数 $\boldsymbol{m}_N$ 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 $\boldsymbol{m}_N$，计算 γ。之后这些值被公式 6.63 用来重新估计 α。这个过程不断进行，直到收敛。

我们可以类似地关于 β 最大化对数边缘似然函数。我们注意到公式 6.58 定义的特征值 λ_i 正比于 β，因此 $\frac{\mathrm{d}}{\mathrm{d}\beta} = \frac{\lambda_i}{\beta}$。于是

$$\frac{\mathrm{d}}{\mathrm{d}\beta} \ln |\boldsymbol{A}| = \frac{\mathrm{d}}{\mathrm{d}\beta} \ln \sum_i \ln(\lambda_i + \alpha) = \frac{1}{\beta} \sum_i \frac{\lambda_i}{\lambda_i + \alpha} = \frac{\gamma}{\beta} \tag{6.64}$$

边缘似然函数的驻点因此满足

$$0 = \frac{N}{2\beta} - \frac{1}{2} \sum_{n=1}^{N} \{t_n - \boldsymbol{m}_N^T \phi(\boldsymbol{x}_n)\}^2 - \frac{\gamma}{\beta} \tag{6.65}$$

整理，我们可以得到

$$\frac{1}{\beta} = \frac{1}{N - \gamma} \sum_{n=1}^{N} \{t_n - \boldsymbol{m}_N^T \phi(\boldsymbol{x}_n)\}^2 \tag{6.66}$$

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。

参数的有效数量

公式 6.63 给出的结果有一个十分优雅的意义，它提供给我们关于 α 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，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 6.58 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 (较小的曲率对应着似然函数轮廓较大的延伸)。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 $\frac{\lambda_i}{\alpha + \lambda_i}$ 位于 0 和 1 之间。结果，由公式 **??** 定义的 γ 的聚会范围为 $0 \leqslant \gamma \leqslant M$。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 $\frac{\lambda_i}{\alpha + \lambda_i}$ 接近于 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 $\frac{\lambda_i}{\alpha+\lambda_i}$ 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 **??** 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 6.66。让我们把 β 和对应的最大似然结果进行比较。这两个公式都把方差表示为目标值和模型预测值的差的平方的平均值。区别在于，最大似然结果的分母是数据点数量 N，而贝叶斯结果的分母是 $N-\gamma$。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

$$\sigma^2_{ML} = \frac{1}{N}\sum_{n=1}^{N}(x_n - \mu_{ML})^2 \tag{6.67}$$

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。对应的无偏估计形式为

$$\sigma^2_{ML} = \frac{1}{N-1}\sum_{n=1}^{N}(x_n - \mu_{ML})^2 \tag{6.68}$$

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 $\boldsymbol{w}^T\boldsymbol{\phi}(\boldsymbol{x})$ 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解。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。

6.6 固定基函数的局限性

线性模型有一些重要的局限性，这使得我们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下来，而这正是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。

1. 数据向量 $\{x_n\}$ 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。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。
2. 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。

第 7 章 分类的线性模型

分类的目标是将输入变量 x 分到 K 个离散的类别 C_k 中的某一类。最常见的情况是，类别互相不相交，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 region)。所谓分类线性模型，是指决策面是输入向量 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。

对于分类问题，最简单的方法涉及到构造判别函数 (discriminant function)，它直接把向量 $\boldsymbol{x}$ 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 $p(C_k|\boldsymbol{x})$ 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。有两种不同的方法确定条件概率分布 $p(C_k|\boldsymbol{x})$。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 $p(\boldsymbol{x}|C_k)$ 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

$$p(C_k|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|C_k)p(C_k)}{p(\boldsymbol{x})} \tag{7.1}$$

我们将在本章中讨论这三种方法。

在上一章讨论的线性回归模型中，模型的预测 $y(\boldsymbol{x}, \boldsymbol{w})$ 由参数 $\boldsymbol{w}$ 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 $y(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + w_0$，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 $\boldsymbol{w}$ 的线性函数进行变换，即

$$y(\boldsymbol{x}) = f(\boldsymbol{w}^T\boldsymbol{x} + w_0) \tag{7.2}$$

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 $y(\boldsymbol{x}) =$ 常数，即 $\boldsymbol{w}^T\boldsymbol{x} + w_0 =$ 常数，因此决策面是 $\boldsymbol{x}$ 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 7.2 描述的一类模型被称为推广的线性模型 (generalized linear model)。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。

7.1 判别函数

判别函数是一个以向量 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。

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

$$y(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + w_0 \tag{7.3}$$

其中 $\boldsymbol{w}$ 被称为权向量 (weight vector)，w_0 被称为偏置 (bias)。对应的决策边界由 $y(\boldsymbol{x}) = 0$ 确定。对于一个输入向量 $\boldsymbol{x}$，如果 $y(\boldsymbol{x}) \geqslant 0$，那么它被分到 C_1 中，否则被分到 C_2 中。向量 $\boldsymbol{w}$ 与决策面内的任何向量都正交，从而 w 确定了决策面的方向。任意一点 x 到决策面的距离为

$$r = \frac{y(\boldsymbol{x})}{\|\boldsymbol{w}\|} \tag{7.4}$$

多分类

考虑把线性判别函数推广到 $K > 2$ 个类别。

1. 1 对其他。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。
2. 1 对 1。引入 $\frac{K(K-1)}{2}$ 个二元判别函数，对一对类别都设置一个判别函数。

上述两种方法都会产生输入空间无法分类的区域。通过引入 K 类判别函数可以避免这些问题。K 类判别函数由 K 个线性函数组成，形式为

$$y_k(\boldsymbol{x}) = \boldsymbol{w}_k^T\boldsymbol{x} + w_{k0} \tag{7.5}$$

对于点 $\boldsymbol{x}$，如果所有的 $j \neq k$ 都有 $y_k(\boldsymbol{x}) > y_j(\boldsymbol{x})$，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 $y_k(\boldsymbol{x}) = y_j(\boldsymbol{x})$，并且对应于一个 $(D-1)$ 维超平面，形式为

$$(\boldsymbol{w}_k - \boldsymbol{w}_j)^T\boldsymbol{x} + (w_{k0} - w_{j0}) = 0 \tag{7.6}$$

这样的判别函数的决策区域总是单连通的，并且是凸的。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。

用于分类的最小平方方法

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。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 $\mathbb{E}[t|\boldsymbol{x}]$。对于二元表示方法，条件期望由后验类概率向量给出。但不幸的是，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。

使用向量记号，我们可以很容易地把这量聚焦在一起表示，即

$$\boldsymbol{y}(\boldsymbol{x}) = \tilde{W}^T\tilde{\boldsymbol{x}} = \begin{pmatrix} w_{1,0} & w_{2,0} & \dots & w_{K,0} \\ w_{1,1} & w_{2,1} & \dots & w_{K,1} \\ \vdots & \vdots & \ddots & \vdots \\ w_{1,D} & w_{2,D} & \dots & w_{K,D} \end{pmatrix}^T \begin{pmatrix} 1 \\ x_1 \\ \vdots \\ x_D \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_K \end{pmatrix} \tag{7.7}$$

其中 $\tilde{W}$ 是一个矩阵，第 k 列由 $D+1$ 维向量 $\tilde{\boldsymbol{w}}_k = (w_{k0}, \boldsymbol{w}_k^T)^T$ 组成，$\tilde{\boldsymbol{x}}$ 是对应的增广输入向量 $(1, \boldsymbol{x}^T)^T$，它带有一个虚输入 $x_0 = 1$。这样，一个新的输入 $\boldsymbol{x}$ 被分配到输出 $y_k = \tilde{\boldsymbol{w}}_k^T\tilde{\boldsymbol{x}}$ 最大的类别中。

通过最小化平方和误差函数来确定参数矩阵 $\tilde{W}$，考虑一个训练数据集 $\boldsymbol{x}_n, \boldsymbol{t}_n$，其中 $n = 1, \dots, N$，然后定义一个矩阵 $\boldsymbol{T}$，它的第 n 行是向量 $\boldsymbol{t}_n^T$。还定义了一个矩阵 $\tilde{\boldsymbol{x}}_n^T$。这样，平方和误差函数可以写成

$$E_D(\tilde{W}) = \frac{1}{2}\mathrm{Tr}\{(\tilde{X}\tilde{W} - \boldsymbol{T})^T(\tilde{X}\tilde{W} - \boldsymbol{T})\} \tag{7.8}$$

令上式关于 $\tilde{W}$ 的导数等于零，整理，可以得到 $\tilde{W}$ 的解，形式为

$$\tilde{W} = (\tilde{X}^T\tilde{X})^{-1}\tilde{X}^T\boldsymbol{T} = \tilde{X}^\dagger\boldsymbol{T} \tag{7.9}$$

其中 $\tilde{X}^\dagger$ 是矩阵 $\tilde{X}$ 的伪逆矩阵。这样，我们得到了判别函数，形式为

$$y(\boldsymbol{x}) = \tilde{W}^T\tilde{\boldsymbol{x}} = \boldsymbol{T}^T(\tilde{X}^\dagger)^T\tilde{\boldsymbol{x}} \tag{7.10}$$

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，最小平方解对于离群点缺少鲁棒性。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。

Fisher 线性判别函数

我们可以从维度降低的角度考察线性分类模型。假设我们有一个 D 维输入向量 $\boldsymbol{x}$，然后使用下式投影到一维

$$y = \boldsymbol{w}^T\boldsymbol{x} \tag{7.11}$$

如果我们在 y 上设置一个阈值，然后把 $y \geqslant -w_0$ 的样本分为 C_1 类，把其余的样本分为 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 $\boldsymbol{w}$，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 C_1 类的 N_1 个点，以及 C_2 类的 N_2 个点。因

此两类的均值向量为

$$\boldsymbol{m}_1 = \frac{1}{N_1}\sum_{n\in C_1}\boldsymbol{x}_n, \qquad \boldsymbol{m}_2 = \frac{1}{N_2}\sum_{n\in C_2}\boldsymbol{x}_n \tag{7.12}$$

如果投影到 $\boldsymbol{w}$ 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 $\boldsymbol{w}$ 使得下式取得最大值

$$m_2 - m_1 = \boldsymbol{w}^T(\boldsymbol{m}_2 - \boldsymbol{m}_1) \tag{7.13}$$

其中

$$m_k = \boldsymbol{w}^T\boldsymbol{m}_k \tag{7.14}$$

是来自类别 C_k 的投影数据的均值。但是，通过增大 $\boldsymbol{w}$，这个表达式可以任意大。为了解决这个问题，我们可以将 $\boldsymbol{w}$ 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 $\boldsymbol{w} \propto (\boldsymbol{m}_2 - \boldsymbol{m}_1)$。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。

来自类别 C_k 的数据经过变换后的类内方差为

$$s_k^2 = \sum_{n\in C_k}(\boldsymbol{w}^T\boldsymbol{x}_n - m_k)^2 \tag{7.15}$$

可以把整个数据集的总的方差定义为 $s_1^2 + s_2^2$。Fisher 准则根据类间方差和类内方差的比值定义，即

$$J(\boldsymbol{w}) = \frac{(m_2 - m_1)^2}{s_1^2 + s_2^2} \tag{7.16}$$

利用公式 7.12, 7.14

$$
\begin{aligned}
(m_2 - m_1)^2 &= \left(\frac{1}{N_2} \sum_{n \in C_2} \boldsymbol{w}^T \boldsymbol{x}_n - \frac{1}{N_1} \sum_{n \in C_1} \boldsymbol{w}^T \boldsymbol{x}_n \right)^2 \\
&= \left(\boldsymbol{w}^T (\bar{x}_{C_1} - \bar{x}_{C_2}) \right)^2 \\
&= \boldsymbol{w}^T (\bar{x}_{C_1} - \bar{x}_{C_2})(\bar{x}_{C_1} - \bar{x}_{C_2})^T \boldsymbol{w} \\
&= \boldsymbol{w}^T S_B \boldsymbol{w} \\
s_1^2 &= \frac{1}{N} \sum_{n \in C_1} (\boldsymbol{w}^T \boldsymbol{x}_n - \frac{1}{N} \sum_{n \in C_1} \boldsymbol{w}^T \boldsymbol{x}_n)(\boldsymbol{w}^T \boldsymbol{x}_n - \frac{1}{N} \sum_{n \in C_1} \boldsymbol{w}^T \boldsymbol{x}_n)^T \\
&= \frac{1}{N} \sum_{n \in C_1} \boldsymbol{w}^T (\boldsymbol{x}_n - \bar{x}_{C_1})(\boldsymbol{x}_n - \bar{x}_{C_1})^T \boldsymbol{w} \\
&= \boldsymbol{w}^T S_1 \boldsymbol{w} \\
s_1^2 + s_2^2 &= \boldsymbol{w}^T (S_1 + S_2) \boldsymbol{w} \\
&= \boldsymbol{w}^T S_W \boldsymbol{w}
\end{aligned}
\tag{7.17}
$$

显式地表达出 $J(\boldsymbol{w})$ 对 $\boldsymbol{w}$ 的依赖

$$
J(\boldsymbol{w}) = \frac{\boldsymbol{w}^T S_B \boldsymbol{w}}{\boldsymbol{w}^T S_W \boldsymbol{w}} \tag{7.18}
$$

其中 S_B 是类间 (between-class) 协方差矩阵，S_W 被称为类内 (within-class) 协方差矩阵。对该式关于 $\boldsymbol{w}$ 求导，并令其为零，有

$$
\begin{aligned}
\frac{\partial J(\boldsymbol{w})}{\partial \boldsymbol{w}} &= \frac{\partial [\boldsymbol{w}^T S_B \boldsymbol{w} \cdot (\boldsymbol{w}^T S_W \boldsymbol{w})^{-1}]}{\partial \boldsymbol{w}} \\
&= 2 S_B \boldsymbol{w} (\boldsymbol{w}^T S_W \boldsymbol{w})^{-1} - 2 \boldsymbol{w}^T S_B \boldsymbol{w} \cdot (\boldsymbol{w}^T S_W \boldsymbol{w})^{-2} \cdot S_W \boldsymbol{w} \\
&= 0 \\
&\Rightarrow S_B \boldsymbol{w} \cdot \boldsymbol{w}^T S_W \boldsymbol{w} = \boldsymbol{w}^T S_B \boldsymbol{w} \cdot S_W \boldsymbol{w} \\
&\Rightarrow S_W \boldsymbol{w} = \frac{\boldsymbol{w}^T S_W \boldsymbol{w}}{\boldsymbol{w}^T S_B \boldsymbol{w}} \cdot S_B \boldsymbol{w} \\
&\Rightarrow \boldsymbol{w} \propto S_W^{-1} S_B \boldsymbol{w} = S_W^{-1} (\boldsymbol{m}_2 - \boldsymbol{m}_1)(\boldsymbol{m}_2 - \boldsymbol{m}_1)^T \boldsymbol{w}
\end{aligned}
\tag{7.19}
$$

我们看到 $S_B \boldsymbol{w}$ 总是在 $\boldsymbol{m}_2 - \boldsymbol{m}_1$ 的方向上。更重要的是，我们不关心 $\boldsymbol{w}$ 的大小，只关心它的方向，因此

$$
\boldsymbol{w} \propto S_W^{-1} (\boldsymbol{m}_2 - \boldsymbol{m}_1) \tag{7.20}
$$

公式 7.20 的结果被称为 Fisher 线性判别函数 (Fisher linear discriminant)。如果类内协方差是各向同性的，从而 S_W 正比于单位矩阵，那么我们看到 $\boldsymbol{w}$ 正比于类均值的差。

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 $y(\boldsymbol{x}) \geqslant y_0$ 时，我们把数据点分到 C_1，否则我们把数据分到 C_2。

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。对于二分类问题，Fisher 准则可以看成最小平方的一个特例。

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。特别地，我们让属于 C_1 的目标值等于 $\frac{N}{N_1}$，其中 N_1 是类别 C_1 的模型的数量，N 是总的模式数量。对于类别 C_2，我们令目标值等于 $-\frac{N}{N_2}$。

推导过程省略，我们得到权向量恰好与根据 Fisher 判别准则得到的结果相同。此外，我们也发现，偏置 w_0 的值为

$$w_0 = -\boldsymbol{w}^T \boldsymbol{m} \tag{7.21}$$

这告诉我们，对于一个新的向量 $\boldsymbol{x}$，如果 $y(\boldsymbol{x}) = \boldsymbol{w}^T(\boldsymbol{x} - \boldsymbol{m}) > 0$，那么 $\boldsymbol{x}$ 应该被分到 C_1，否则应该被分到 C_2。

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 大于类另数量 K。接下来，我们引入 $D' > 1$ 个线性“特征”$y_k = \boldsymbol{w}_k^T \boldsymbol{x}$，其中 $k = 1, \ldots, D'$。为了方便，这些特征值可以聚集起来组成向量 $\boldsymbol{y}$。类似地，权向量 $\{\boldsymbol{w}_k\}$ 可以被看成矩阵 $\boldsymbol{W}$ 的列。因此

$$\boldsymbol{y} = \boldsymbol{W}^T \boldsymbol{x} \tag{7.22}$$

类内协方差矩阵可以推广到 K 类，有

$$\boldsymbol{S}_W = \sum_{k=1}^{K} \boldsymbol{S}_k \tag{7.23}$$

其中

$$\boldsymbol{S}_k = \sum_{n \in C_k} (\boldsymbol{x}_n - \boldsymbol{m}_k)(\boldsymbol{x} - \boldsymbol{m}_k)^T \tag{7.24}$$

$$\boldsymbol{m}_k = \frac{1}{N_k} \sum_{n \in C_k} \boldsymbol{x}_n \tag{7.25}$$

其中 N_k 是类别 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 的方法，首先考虑整体的协方差矩阵

$$\boldsymbol{S}_T = \sum_{n=1}^{N} (\boldsymbol{x}_n - \boldsymbol{m})(\boldsymbol{x} - \boldsymbol{m})^T \tag{7.26}$$

其中 $\boldsymbol{m}$ 是全体数据的均值

$$\boldsymbol{m} = \frac{1}{N}\sum_{n=1}^{N}\boldsymbol{x}_n = \frac{1}{N}\sum_{k=1}^{K}N_k\boldsymbol{m}_k \tag{7.27}$$

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 7.23 给出的类内协方差矩阵，加上另一个矩阵 $\boldsymbol{S}_B$，它可以看做类间协方差矩阵。

$$\boldsymbol{S}_T = \boldsymbol{S}_W + \boldsymbol{S}_B \tag{7.28}$$

其中

$$\boldsymbol{S}_B = \sum_{k=1}^{K}N_k(\boldsymbol{m}_k - \boldsymbol{m})(\boldsymbol{m}_k - \boldsymbol{m})^T \tag{7.29}$$

协方差矩阵被定义在原始的 $\boldsymbol{x}$ 空间中。我们现在在投影的 D' 维 $\boldsymbol{y}$ 空间中定义类似的矩阵

$$\boldsymbol{S}_W = \sum_{k=1}^{K}\sum_{n\in C_k}(\boldsymbol{y}_n - \boldsymbol{\mu}_k)(\boldsymbol{y}_n - \boldsymbol{\mu}_k)^T \tag{7.30}$$

以及

$$\boldsymbol{S}_B = \sum_{k=1}^{K}N_k(\boldsymbol{\mu}_k - \boldsymbol{\mu})(\boldsymbol{\mu}_k - \boldsymbol{\mu})^T \tag{7.31}$$

其中

$$\boldsymbol{\mu}_k = \frac{1}{N_k}\sum_{n\in C_k}\boldsymbol{y}_n, \quad \boldsymbol{\mu} = \frac{1}{N}\sum_{k=1}^{K}N_k\boldsymbol{\mu}_k \tag{7.32}$$

与之前一样，我们想构造一个标题，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。其中一种选择是

$$J(\boldsymbol{W}) = \mathrm{Tr}\{\boldsymbol{s}_W^{-1}\boldsymbol{s}_B\} \tag{7.33}$$

这个判别准则可以显式地写成投影矩阵 $\boldsymbol{W}$ 的函数，形式为

$$J(\boldsymbol{W}) = \mathrm{Tr}\{(\boldsymbol{W}^T\boldsymbol{S}_W\boldsymbol{W})^{-1}(\boldsymbol{W}^T\boldsymbol{S}_B\boldsymbol{W})\} \tag{7.34}$$

最大化这个判别准则是很直接的，虽然有些麻烦。通过这种方法我们不能够找到多于 (K-1) 个线性“特征”。

感知器算法

线性判别模型的另一个例子是 Rosenblatt 提出的感知器算法。它对应于一个二分类的模型，这个模型中，输入向量 $\boldsymbol{x}$ 首先使用一个固定的非线性变换得到一个特征向量 $\boldsymbol{\phi}(\boldsymbol{x})$，

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

$$y(\boldsymbol{x}) = f(\boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x})) \tag{7.35}$$

其中非线性激活函数 $f(\cdot)$ 是一个阶梯函数，形式为

$$f(a) = \begin{cases} +1, & a \geqslant 0 \\ -1, & a < 0 \end{cases} \tag{7.36}$$

向量 $\boldsymbol{\phi}(\boldsymbol{x})$ 通常包含一个偏置分量 $\phi_0(\boldsymbol{x}) = 1$。用来确定感知器的参数 $\boldsymbol{w}$ 的算法可以很容易地从误差函数最小化的思想中得到。一个自然的选择是误分类的模型的总数。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 $\boldsymbol{w}$ 的分段常函数，从而当 $\boldsymbol{w}$ 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。这样会不连续变化。这样做还使得使用误差函数改变 $\boldsymbol{w}$ 的方法无法使用，因为在几乎所有的地方梯度都等于零。

因此考虑一个另外的误差函数，被称为感知器准则 (perceptron criterion)。

$$E_P(\boldsymbol{w}) = -\sum_{n \in \mathcal{M}} \boldsymbol{w}^T \boldsymbol{\phi}_n t_n \tag{7.37}$$

其中 $\boldsymbol{\phi}_n = \boldsymbol{\phi}(\boldsymbol{x}_n)$ 和 $\mathcal{M}$ 表示所有误分类模型的集合。某个特定的误分类模型对于误差函数的贡献是 $\boldsymbol{w}$ 空间中模式被误分类的区域中 $\boldsymbol{w}$ 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 $\boldsymbol{w}$ 的变化为

$$\boldsymbol{w}^{(\tau+1)} = \boldsymbol{w}^{(\tau)} - \eta \nabla E_P(\boldsymbol{w}) = \boldsymbol{w}^{(\tau)} + \eta \boldsymbol{\phi}_n t_n \tag{7.38}$$

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。

这里要画一个图

7.2 概率生成式模型

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。这里我们使用生成式的方法。

首先考虑二分类的情形。类别 C_1 的后验概率可以写成

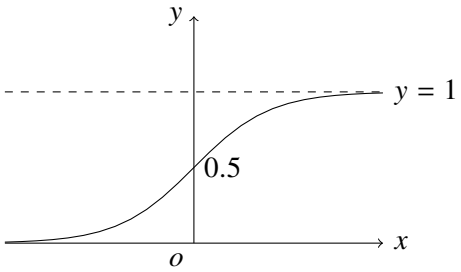
$$\begin{aligned} p(C_1|\boldsymbol{x}) &= \frac{p(\boldsymbol{x}|C_1)p(C_1)}{p(\boldsymbol{x}|C_1)p(C_1) + p(\boldsymbol{x}|C_2)p(C_2)} \\ &= \frac{1}{1+\exp(-a)} = \sigma(a) \end{aligned} \tag{7.39}$$

其中我们定义了

$$a = \ln \frac{p(\boldsymbol{x}|C_1)p(C_1)}{p(\boldsymbol{x}|C_2)p(C_2)} \tag{7.40}$$

$\sigma(a)$ 是 logistic sigmoid 函数。“sigmoid” 的意思是 “S 形”。这种函数有时被称为 “挤压函数”,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

$$\sigma(-a) = 1 - \sigma(a) \tag{7.41}$$



logistic sigmoid 的反函数为

$$a = \ln\left(\frac{\sigma}{1-\sigma}\right) \tag{7.42}$$

被称为 logit 函数。它表示两类的概率比值的对数 $\ln\left[\frac{p(C_1|\boldsymbol{x})}{p(C_2|\boldsymbol{x})}\right]$,也被称为 log odds 函数。

对于 K>2 个类别的情形,我们有

$$\begin{aligned} p(C_k|\boldsymbol{x}) &= \frac{p(\boldsymbol{x}|C_k)p(C_k)}{\sum_j p(\boldsymbol{x}|C_j)p(C_j)} \\ &= \frac{\exp(a_k)}{\sum_j \exp(a_j)} \end{aligned} \tag{7.43}$$

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

$$a_k = \ln p((\boldsymbol{x}|C_k)p(C_k)) \tag{7.44}$$

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 $p(C_k|\boldsymbol{x}) \simeq 1$ 且 $p(C_j|\boldsymbol{x}) \simeq 0$。

连续输入

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,假设所有的类别的协方差矩阵相同。这样类别 C_k 的类条件概率为

$$p(\boldsymbol{x}|C_k) = \frac{1}{(2\pi)^{\frac{D}{2}}} \frac{1}{|\Sigma|^{\frac{1}{2}}} \exp\left\{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_k)^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_k)\right\} \tag{7.45}$$

首先考虑两类的情形。

$$
\begin{aligned}
& p(C_1|\boldsymbol{x}) = \sigma(a) \\
\Rightarrow\ & a = \ln \frac{p(\boldsymbol{x}|C_1)p(C_1)}{p(\boldsymbol{x}|C_2)p(C_2)} \\
& = -\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_1)^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_1) + \frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_2)^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_2) + \ln\frac{p(C_1)}{p(C_2)} \\
& = -\frac{1}{2}\boldsymbol{x}^T\Sigma^{-1}\boldsymbol{x} + \frac{1}{2}\boldsymbol{\mu}_1^T\Sigma^{-1}\boldsymbol{x} + \frac{1}{2}\boldsymbol{x}^T\Sigma^{-1}\boldsymbol{\mu}_1 - \frac{1}{2}\boldsymbol{\mu}_1^T\Sigma^{-1}\boldsymbol{\mu}_1 \\
& \quad + \frac{1}{2}\boldsymbol{x}^T\Sigma^{-1}\boldsymbol{x} - \frac{1}{2}\boldsymbol{\mu}_2^T\Sigma^{-1}\boldsymbol{x} - \frac{1}{2}\boldsymbol{x}^T\Sigma^{-1}\boldsymbol{\mu}_2 + \frac{1}{2}\boldsymbol{\mu}_2^T\Sigma^{-1}\boldsymbol{\mu}_2 \\
& \quad + \ln\frac{p(C_1)}{p(C_2)} \\
& = \underbrace{\Sigma^{-1}(\boldsymbol{\mu}_1-\boldsymbol{\mu}_2)}_{\boldsymbol{w}^T}\boldsymbol{x} \underbrace{-\frac{1}{2}\boldsymbol{\mu}_1^T\Sigma^{-1}\boldsymbol{\mu}_1 + \frac{1}{2}\boldsymbol{\mu}_2^T\Sigma^{-1}\boldsymbol{\mu}_2 + \ln\frac{p(C_1)}{p(C_2)}}_{w_0} \\
& = \boldsymbol{w}^T\boldsymbol{x} + w_0 \\
\Rightarrow\ & \sigma(a) = \sigma(\boldsymbol{w}^T\boldsymbol{x} + w_0)
\end{aligned}
\tag{7.46}
$$

高斯概率密度的指数项中 $\boldsymbol{x}$ 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 $\boldsymbol{x}$ 的线性函数的 logistic sigmoid 函数。最终求得决策边界 $a = 0$ 为常数的决策面。先验概率密度 $p(C_k)$ 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 7.43,有

$$a_k(\boldsymbol{x}) = \boldsymbol{w}_k^T\boldsymbol{x} + w_{k0} \tag{7.47}$$

$$w_{k0} = -\frac{1}{2}\boldsymbol{\mu}_k^T\Sigma^{-1}\boldsymbol{\mu}_k + \ln p(C_k) \tag{7.48}$$

其中定义了

$$\boldsymbol{w}_k = \Sigma^{-1}\boldsymbol{\mu}_k \tag{7.49}$$

我们看到 $a_k(\boldsymbol{x})$ 与之前一样是 $\boldsymbol{x}$ 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 $\boldsymbol{x}$ 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 $p(\boldsymbol{x}|C_k)$ 有自己的协方差矩阵 Σ_k,那么之前二次项消去的现象不会出现,从而我们会得到 $\boldsymbol{x}$ 的二次函数,这就引出了二次判别函数 (quadratic discriminant)。

最大似然解

一旦具体化了类条件概率密度 $p(\boldsymbol{x}|C_k)$ 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 $p(C_k)$。

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。假设有一个数据集 $\{x_n, t_n\}$，其中 $n = 1, \ldots, N$。先验概率记作 $p(C_1) = \pi$，从而 $p(C_2) = 1-\pi$。对于一个来自类别 C_1 的数据点 $\boldsymbol{x}_n$，我们有 $t_n = 1$，因此

$$p(\boldsymbol{x}_n, C_1) = p(C_1)p(\boldsymbol{x}_n|C_1) = \pi\mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_1, \Sigma) \tag{7.50}$$

$$p(\boldsymbol{x}_n, C_2) = p(C_2)p(\boldsymbol{x}_n|C_2) = (1-\pi)\mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_2, \Sigma) \tag{7.51}$$

于是，似然函数为

$$p(\boldsymbol{t}, \boldsymbol{X}|\pi, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \Sigma) = \prod_{n=1}^{N}[\pi\mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_1, \Sigma)]^{t_n}[(1-\pi)\mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_2, \Sigma)]^{1-t_n} \tag{7.52}$$

其中 $\boldsymbol{t} = (t_1, \ldots, t_N)^T$。对数似然函数为

$$\begin{aligned}\log p(\boldsymbol{t}, \boldsymbol{X}|\pi, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \Sigma) &= \sum_{n=1}^{N}\log\left\{\pi\mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_1, \Sigma)]^{t_n}[(1-\pi)\mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_2, \Sigma)]^{1-t_n}\right\} \\ &= \sum_{n=1}^{N}\left\{\log\pi^{t_n}(1-\pi)^{1-t_n} + \log\mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_1, \Sigma)^{t_n} + \log\mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_2, \Sigma)^{1-t_n}\right\}\end{aligned} \tag{7.53}$$

最大化对数似然函数

1. 求 π

$$\begin{aligned}\frac{\partial}{\partial\pi}\left[\sum_{n=1}^{N}\log\pi^{t_n}(1-\pi)^{1-t_n}\right] &= \sum_{n=1}^{N}\frac{t_n}{\pi} - \frac{(1-t_n)}{1-\pi} = 0 \\ &= \sum_{n=1}^{N}t_n - \pi t_n - \pi + \pi t_n \\ &= \sum_{n=1}^{N}t_n - \pi \\ &= \sum_{n=1}^{N}t_n - N\pi = 0 \\ \Rightarrow \pi &= \frac{1}{N}\sum_{n=1}^{N}t_n = \frac{N_1}{N} = \frac{N_1}{N_1+N_2}\end{aligned} \tag{7.54}$$

2. 求 $\boldsymbol{\mu}$

$$\begin{aligned}
\frac{\partial}{\partial \boldsymbol{\mu}_1}\left[\log \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_1,\Sigma)^{t_n}\right] &\Rightarrow \frac{\partial}{\partial \boldsymbol{\mu}_1}\left[\sum_{n=1}^{N}-\frac{1}{2}t_n(\boldsymbol{x}_n-\boldsymbol{\mu}_1)^T\Sigma^{-1}(\boldsymbol{x}_n-\boldsymbol{\mu}_1)\right] \\
&= \frac{\partial}{\partial \boldsymbol{\mu}_1}\left[-\frac{1}{2}\sum_{n=1}^{N}t_n(\boldsymbol{x}_n^T\Sigma^{-1}-\boldsymbol{\mu}_1^T\Sigma^{-1})(\boldsymbol{x}_n-\boldsymbol{\mu}_1)\right] \\
&= \frac{\partial}{\partial \boldsymbol{\mu}_1}\left[-\frac{1}{2}\sum_{n=1}^{N}t_n(\underbrace{\boldsymbol{x}_n^T\Sigma^{-1}\boldsymbol{x}_n}_{\text{常数}}-2\boldsymbol{\mu}_1^T\Sigma^{-1}\boldsymbol{x}_n+\boldsymbol{\mu}_1^T\Sigma^{-1}\boldsymbol{\mu}_1)\right] \\
&= \sum_{n=1}^{N}t_n(\Sigma^{-1}\boldsymbol{x}_n+\Sigma^{-1}\boldsymbol{\mu}_1)=0 \\
&\Rightarrow \sum_{n=1}^{N}t_n(\boldsymbol{\mu}_1-\boldsymbol{x}_n)=0 \\
\Rightarrow \boldsymbol{\mu}_1 &= \frac{\sum_{n=1}^{N}t_n\boldsymbol{x}_n}{\sum_{n=1}^{N}t_n} \\
&= \frac{1}{N_1}\sum_{n=1}^{N}t_n\boldsymbol{x}_n
\end{aligned} \tag{7.55}$$

同理

$$\boldsymbol{\mu}_2 = \frac{1}{N_2}\sum_{n=1}^{N}(1-t_n)\boldsymbol{x}_n \tag{7.56}$$

3. 求 Σ

已知

$$Tr(AB) = \mathrm{Tr}(BA) \tag{7.57}$$

$$\frac{\partial \mathrm{Tr}(AB)}{\partial A} = B^T \tag{7.58}$$

$$\frac{\partial |A|}{\partial A} = |A|A^{-1} \tag{7.59}$$

$$\frac{\partial \ln|A|}{\partial A} = A^{-1} \tag{7.60}$$

考察部分分式

$$\begin{aligned}
\sum_{n=1}^{N}\log \mathcal{N}(\boldsymbol{\mu},\Sigma) &= \sum_{n=1}^{N}\log|\Sigma|^{-\frac{1}{2}}+(-\frac{1}{2}(\boldsymbol{x}_n-\boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{x}_n-\boldsymbol{\mu}))+C \\
&= -\frac{1}{2}N\log|\Sigma|-\frac{1}{2}\sum_{n=1}^{N}(\boldsymbol{x}_n-\boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{x}_n-\boldsymbol{\mu})+C \\
&= -\frac{1}{2}N\log|\Sigma|-\frac{1}{2}NTr(S\cdot\Sigma^{-1})+C
\end{aligned} \tag{7.61}$$

其中

$$S = \frac{1}{N}\sum_{n=1}^{N}(\boldsymbol{x}_n - \boldsymbol{\mu})(\boldsymbol{x}_n - \boldsymbol{\mu})^T \tag{7.62}$$

将与 Σ 相关的式子组合到一起，有，求关于 Σ 的偏导，并令其为零

$$\begin{aligned}
\triangle &= \sum_{n=1}^{N} \log \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_1, \Sigma)^{t_n} + \log \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_2, \Sigma)^{1-t_n} \\
&= -\frac{1}{2}\log|\Sigma| - \frac{1}{2}N_1 Tr(S_1 \cdot \Sigma^{-1}) - \frac{1}{2}N_2 Tr(S_2 \cdot \Sigma^{-1}) + C \\
\frac{\partial \triangle}{\partial \Sigma} &= -\frac{1}{2}(N\Sigma^{-1} - N_1 S_1 \Sigma^{-2} - N_2 S_2 \Sigma^{-2}) = 0 \\
&\Rightarrow \Sigma = \frac{N_1}{N}S_1 + \frac{N_2}{N}S_2
\end{aligned} \tag{7.63}$$

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。

离散特征

考虑离散特征值 x_i 的情形。为了简化起见，我们首先考虑二元特征值 $x_i \in \{0, 1\}$，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) 假设，这个假设中，特征值被看成相互独立的，以类别 C_k 为条件，因此。我们得到类条件分布，形式为

$$\begin{aligned}
p(\boldsymbol{X} = \boldsymbol{x}|C_k) &= p(X^{(1)} = x^{(1)}, \ldots, X^{(D)} = x^{(D)}|C_k) \\
&= \prod_{i=1}^{D} \mu_{k_i}^{x_i}(1 - \mu_{k_i})^{1-x_i}
\end{aligned} \tag{7.64}$$

其中对于每个类别，都有 D 个独立的参数。代入 softmax 函数中，有

$$\begin{aligned}
a_k(\boldsymbol{x}) &= \ln p(\boldsymbol{x}|C_k)p(C_k) \\
&= \ln \prod_{i=1}^{D} \mu_{k_i}^{x_i}(1 - \mu_{k_i})^{1-x_i} + \ln p(C_k) \\
&= \sum_{i=1}^{D}\{x_i \ln \mu_{k_i} + (1 - x_i)\ln(1 - \mu_{k_i})\} + \ln p(C_k)
\end{aligned} \tag{7.65}$$

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——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。

朴素贝叶斯法实际上学习到生成数据的机制，所以属于生成模型，条件独立性假设等于是说用于分类的特征在类确定的条件下都是条件独立的。这一假设使得朴素贝叶斯变

得简单,但有时会牺牲一定的分类准确率。

朴素贝叶斯法分类时,对给定的输入 x,通过学习到的模型计算后验概率分布 $p(C_k|X=x)$,将后验概率最大的类作为 x 类的输出。后验概率计算根据贝叶斯定理进行。

$$y=f(\boldsymbol{x})=arg\max_{c_k}\frac{p(C_k)\prod_j p(X^{(j)}|C_k)}{\sum_k p(C_k)\prod_j p(X^{(j)}|C_k)} \tag{7.66}$$

注意到分母对所有 C_k 都是相同的,所以

$$y=arg\max_{c_k} p(C_k)\prod_j p(X^{(j)}|C_k) \tag{7.67}$$

朴素贝叶斯法将实例分到后验概率最大的类中。这等价于期望风险最小化。假设选择 0-1 损失函数:

$$L(Y,f(X))=\begin{cases}1, & Y\neq f(X)\\ 0, & Y=f(X)\end{cases} \tag{7.68}$$

式中 $f(X)$ 是分类决策函数。这时,期望风险函数为

$$\begin{aligned}E_{exp}(f)&=E[L(Y,f(X))]\\&=\int_{x\mathcal{X}y}L(y,f(x))p(x,y)dxdy\\&=\int_x[L(y,f(x))p(y|x)dy]p(x)dx\\&=E_X\sum_{k=1}^{K}[L(C_k,f(X))]p(C_k|X)\end{aligned} \tag{7.69}$$

为了使期望风险最小化,只需对 $X=x$ 逐个极小化,由此得到:

$$\begin{aligned}f(x)&=arg\min_{y\in\dagger}\sum_{k=1}^{K}L(C_k,y)p(C_k|X=x)\\&=arg\min_{y\in\dagger}\sum_{k=1}^{K}p(y\neq C_k|X=x)\\&=arg\min_{y\in\dagger}(1-p(y=C_k|X=x))\\&=arg\max_{y\in\dagger}p(y=C_k|X=x)\end{aligned} \tag{7.70}$$

下面给出朴素贝叶斯法的学习与分类算法

输入:训练数据 $T=\{(x_1,y_1),(x_2,y_2),\ldots,(x_N,y_N)\}$,其中 $x_i=(x_i^{(1)},x_i^{(2)},\ldots,x_i^{(n)})^T$,$x_i^{(j)}$ 是第 i 个样本的第 j 个特征,$x_i^{(j)}\in\{a_{j1},a_{j2},\ldots,a_{jS_j}\}$,$a_{jl}$ 是第 j 个特征可能取的第 l 个值,$j=1,2,\ldots,n,l=1,2,\ldots,S_j,y_i\in\{c_1,c_2,\ldots,c_k\}$; 实例 x;

输出:实例 x 的分类。

1. 计算先验概率及条件概率

$$P(Y=C_k)=\frac{\sum_{i=1}^{N}I(y_i=C_k)}{N},k=1,2,\ldots,K \tag{7.71}$$

$$P(X^{(j)}=a_{jl}|Y=C_k)=\frac{\sum_{i=1}^{N}I(x_i^{(j)}=a_{jl},y_i=C_k)}{\sum_{i=1}^{N}I(y_i=C_k)} \tag{7.72}$$

$$j=1,2,\ldots,N,l=1,2,\ldots,S_j,k=1,2,\ldots,K \tag{7.73}$$

2. 对于给定的实例 $x=(x^{(1)},x^{(2)},\ldots,x^{(n)})^T$，计算

$$P(Y=C_k)\prod_{j=1}^{n}P(X^{(j)}=x^{(j)}|Y=C_k),k=1,2,\ldots,K \tag{7.74}$$

3. 确定实例 x 的类

$$y=arg\,\underset{C_k}{max}\,P(Y=C_k)\prod_{j=1}^{n}P(x^{(j)}=x^{(j)}|Y=Cc_k) \tag{7.75}$$

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($K\geqslant 2$ 个类别) 激活函数给出。通过假设类条件概率密度 $p(\boldsymbol{x}|C_k)$ 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。

7.3 概率判别式模型

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 $p(\boldsymbol{x}|C_k)$ 的选择，类别 C_1 后验概率分布可以写成作用于 $\boldsymbol{x}$ 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 $\boldsymbol{x}$ 的线性函数的 softmax 变换给出。对于类条件概率密度 $p(\boldsymbol{x}|C_k)$ 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。

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。

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。在直接方法中，我们最大化由条件概率分布 $p(C_k|\boldsymbol{x})$ 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的参数需要确定，并且预测表现会提升，尤其是当类条件概率密度的假设没有很好地近似真实的分布的时候更是如此。

固定基函数

如果首先使用一个基函数向量 $\boldsymbol{\phi}(\boldsymbol{x})$ 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 $\boldsymbol{\phi}$ 中是线性的，因此对应于原始 $\boldsymbol{x}$ 空间中的非线性决策边界。

对于许多实际问题来说，类条件概率密度 $p(\boldsymbol{x}|C_k)$ 之间有着相当大的重叠。这表明至少对于某些 $\boldsymbol{x}$ 的值，后验概率 $p(C_k|\boldsymbol{x})$ 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，然后使用前面章节讨论的标准的决策论。需要注意的是，非线性变换 $\boldsymbol{\phi}(\boldsymbol{x})$ 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，恰当的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。

logistic 回归

首先通过二分类问题开始对于一般线性模型的讨论。类别 C_1 的后验概率可以写成作用在特征向量 $\boldsymbol{\phi}$ 的线性函数上的 logistic sigmoid 函数的形式，即

$$p(C_1|\boldsymbol{\phi}) = y(\boldsymbol{\phi}) = \sigma(\boldsymbol{w}^T\boldsymbol{\phi}) \tag{7.76}$$

$$p(C_2|\boldsymbol{\phi}) = 1 - p(C_1|\boldsymbol{\phi}) \tag{7.77}$$

这个模型被称为 logistic 回归，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 $\boldsymbol{\phi}$，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 $\frac{M(M+1)}{2}$ 个参数来描述协方差矩阵。算上类先验 $p(C_1)$，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 logstic 回归方法中对于参数数据 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。

现在使用最大似然方法来确定 logistic 回归模型的参数。对于一个数据集 $\boldsymbol{\phi}_n, t_n$，其中 $t_n \in \{0,1\}, \boldsymbol{\phi}_n = \boldsymbol{\phi}(\boldsymbol{x}_n), n = 1,\ldots,N$，似然函数可以写成

$$p(\boldsymbol{t}|\boldsymbol{w}) = \prod_{n=1}^{N} y_n^{t_n}\{1-y_n\}^{1-t_n} \tag{7.78}$$

其中 $\boldsymbol{t} = (t_1,\ldots,t_N)^T$ 且 $y_n = p(C_1|\boldsymbol{\phi}_n)$。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

$$\begin{aligned} E(\boldsymbol{w}) &= -\ln p(\boldsymbol{t}|\boldsymbol{w}) \\ &= -\sum_{n=1}^{N}\{t_n \ln y_n + (1-t_n)\ln(1-y_n)\} \end{aligned} \tag{7.79}$$

其中 $y_n = \sigma(a_n)$ 且 $a_n = \boldsymbol{w}^T\phi_n$。两侧关于 $\boldsymbol{w}$ 取误差函数的梯度，我们有

$$\begin{aligned}\nabla E(\boldsymbol{w}) &= -\sum_{n=1}^{N}\{\frac{t_n}{y_n}y_n^{'} + \frac{(1-t_n)}{1-y_n}(1-y_n)^{'}\} \\ &= -\sum_{n=1}^{N}\frac{t_n}{\sigma}\sigma(1-\sigma)\phi_n - \frac{1-t_n}{1-\sigma}\sigma(1-\sigma)\phi_n \\ &= \sum_{n=1}^{N}(y_n - t_n)\phi_n\end{aligned} \tag{7.80}$$

推导时用到了

$$\frac{\mathrm{d}\sigma}{\mathrm{d}a} = \sigma(1-\sigma) \tag{7.81}$$

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”与基函数向量 ϕ_n 相乘。此外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。

问题变成了以对数似然函数为目标函数的最优化问题。logistic 回归学习中通常采用的方法是梯度下降法及拟牛顿法。

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。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。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 $\boldsymbol{w}$ 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。

迭代重加权最小平方

对于 logistic 回归来说，不再有解析解，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，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 $E(\boldsymbol{w})$，Newton-Raphson 对权值的更新的形式为

$$\boldsymbol{w}^{\text{新}} = \boldsymbol{w}^{\text{旧}} - H^{-1}\nabla E(\boldsymbol{w}) \tag{7.82}$$

其中 H 是一个 Hessian 矩阵。把 Newton-Raphson 方法应用到现行回归模型上，误差函数为平方和误差函数。这个误差函数的梯度和 Hessian 矩阵为

$$\nabla E[\boldsymbol{w}] = \sum_{n=1}^{N}(\boldsymbol{w}^T\phi_n - t_n)\phi_n = \Phi^T\Phi\boldsymbol{w} - \Phi^T\boldsymbol{t} \tag{7.83}$$

$$H = \nabla\nabla E(\boldsymbol{w}) = \sum_{n=1}^{N}\phi_n\phi_n^T = \Phi^T\Phi \tag{7.84}$$

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

$$\begin{aligned}\boldsymbol{w}^{新} &= \boldsymbol{w}^{旧} - (\Phi^T\Phi)^{-1}\{\Phi^T\Phi\boldsymbol{w}^{旧} - \Phi^T\boldsymbol{t}\} \\ &= (\Phi^T\Phi)^{-1}\Phi^T\boldsymbol{t}\end{aligned} \tag{7.85}$$

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。

现在把 Newton-Raphson 更新应用到 logistic 回归模型的交叉熵误差函数上。

$$\nabla E(\boldsymbol{w}) = \sum_{n=1}^{N}(y_n - t_n)\phi_n = \Phi^T(\boldsymbol{y} - \boldsymbol{t}) \tag{7.86}$$

$$H = \nabla\nabla E(\boldsymbol{w}) = \sum_{n=1}^{N}y_n(1 - y_n)\phi_n\phi_n^T = \Phi^T\mathcal{R}\Phi \tag{7.87}$$

推导过程中，引入了一个 $N \times N$ 的对角矩阵 $\mathcal{R}$，元素为

$$\mathcal{R}_{nn} = y_n(1 - y_n) \tag{7.88}$$

我们看到，Hessian 矩阵不再是常量，而是通过权矩阵 $\mathcal{R}$ 依赖于 $\boldsymbol{w}$。这对应于误差函数不是二次函数的事实。使用性质 $0 < y_n < 1$，我们看到对于任意向量 $\boldsymbol{u}$ 都有 $\boldsymbol{u}^T H\boldsymbol{u} > 0$，因此 Hessian 矩阵 H 是正定的。因此误差函数是 $\boldsymbol{w}$ 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

$$\begin{aligned}\boldsymbol{w}^{新} &= \boldsymbol{w}^{旧} - (\Phi^T\mathcal{R}\Phi)^{-1}\Phi^T(\boldsymbol{y} - \boldsymbol{t}) \\ &= (\Phi^T\mathcal{R}\Phi)^{-1}\{\Phi^T\mathcal{R}\Phi\boldsymbol{w}^{旧} - \Phi^T(\boldsymbol{y} - \boldsymbol{t})\} \\ &= (\Phi^T\mathcal{R}\Phi)^{-1}\Phi^T\mathcal{R}\boldsymbol{z}\end{aligned} \tag{7.89}$$

其中 $\boldsymbol{z}$ 是一个 N 维向量，元素为

$$\boldsymbol{z} = \Phi\boldsymbol{w}^{旧} - \mathcal{R}^{-1}(\boldsymbol{y} - \boldsymbol{t}) \tag{7.90}$$

更新公式的形式为一组加权最小平方问题的规范方程。由于权矩阵 $\mathcal{R}$ 不是常量，而是依赖于参数向量 $\boldsymbol{w}$，因此我们必须迭代地应用规范方程，每次使用新的权向量 $\boldsymbol{w}$ 计算一个修

正的权矩阵 $\mathcal{R}$，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。对角 $\mathcal{R}$ 可以看成方差。

事实上，我们可以把 IRLS 看成变量空间 $a = \boldsymbol{w}^T\phi$ 的线性问题的解。这样，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。

$$\begin{aligned} a_n(\boldsymbol{w}) &\simeq a_n(\boldsymbol{w}^{\text{旧}}) + \left.\frac{\mathrm{d}a_n}{\mathrm{d}y_n}\right|_{\boldsymbol{w}^{\text{旧}}} (t_n - y_n) \\ &= \phi_n^T \boldsymbol{w}^{\text{旧}} - \frac{y_n - t_n}{y_n(1-y_n)} = z_n \end{aligned} \tag{7.91}$$

多类 logistic 回归

对于多分类的生成式模型，后验概率由特征变量的线性函数的 softmax 变换给出，即

$$p(C_k|\phi) = y_k(\phi) = \frac{\exp(\boldsymbol{w}_k^T\phi)}{\sum_j \exp(\boldsymbol{w}_k^T\phi)} \tag{7.92}$$

probit 回归

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。回到二分类的情形，再次使用一般的线性模型的框架，即

$$p(t = 1|a) = f(a) \tag{7.93}$$

其中 $a = \boldsymbol{w}^T\phi$，且 $f(\cdot)$ 为激活函数。对于每个输入 ϕ_n，我们计算 $\boldsymbol{w}^T\phi_n$，然后按照下面的方式设置目标值

$$\begin{cases} t_n = 1, & \text{如果} a_n \geqslant \theta \\ t_n = 0, & \text{其他情况} \end{cases} \tag{7.94}$$

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

$$f(a) = \int_{-\infty}^{a} p(\theta)d\theta \tag{7.95}$$

在随机阈值模型中，如果 $a = \boldsymbol{w}^T\phi$ 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。

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

$$\Phi(a) = \int_{-\infty}^{a} \mathcal{N}(\theta|0,1)\mathrm{d}\theta \tag{7.96}$$

这被称为逆 probit(inverse probit) 函数。它的形状为 sigmoid 形。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

$$\text{erf}(a) = \frac{2}{\sqrt{\pi}} \int_0^a \exp(-\theta^2) d\theta \tag{7.97}$$

它被称为 erf 函数或者被称为 error 函数。它与逆 probit 函数的关系为

$$\Phi(a) = \frac{1}{2} \left\{ 1 + \text{erf}\left(\frac{a}{\sqrt{2}} \right) \right\} \tag{7.98}$$

基于 probit 激活函数的一般的线性模型被称为 probit 回归。

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 $\boldsymbol{x}$ 的测量误差产生，或者由目标值 t 的错误标记产生。他们会严重地干扰分类器，在这一点上 probit 模型对于离群点会更加敏感。对于错误标记的影响可以合并到概率模型中。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。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。

标准链接函数

对于高斯噪声分布的线性回归模型，如果我们对数据点 n 对误差函数的贡献关于参数向量 $\boldsymbol{w}$ 求导，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 $y_n = \boldsymbol{w}^T \phi_n$。类似地，对于 logistic sigmoid 激活函数与交叉熵误差函数的组合，以及多类交叉熵误差函数的 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。

我们使用指数族分布的限制形式。注意，我们把指数族分布的假设应用于目标变量 t，而不是应用于输入向量 $\boldsymbol{x}$。于是，我们考虑目标变量的条件分布

$$p(t|\eta, s) = \frac{1}{s} h\left(\frac{t}{s} \right) g(\eta) \exp\left\{ \frac{\eta t}{s} \right\} \tag{7.99}$$

使用与推导结果 2.148 时相同的过程，我们看到 t 的条件均值 (记作 y) 为

$$y \equiv [t|\eta] = -s \frac{\mathrm{d}}{\mathrm{d}\eta} \ln g(\eta) \tag{7.100}$$

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 的方法，我们将一般线性模型定义为这样的模型：y 是输入变量 (或者特征变量) 的线性组合的非线性函数，即

$$y = f(\boldsymbol{w}^T \phi) \tag{7.101}$$

其中 $f^{-1}(\cdot)$ 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

$$\ln p(\boldsymbol{t}|\eta, s) = \sum_{n=1}^{N} \ln p(t_n|\eta, s) = \sum_{n=1}^{N} \left\{ \ln g(\eta_n) + \frac{\eta_n t_n}{s} \right\} + \text{常数} \tag{7.102}$$

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 $\boldsymbol{w}$ 的导数为

$$\begin{aligned} \nabla_{\boldsymbol{w}} \ln p(\boldsymbol{t}|\eta, s) &= \sum_{n=1}^{N} \left\{ \frac{\mathrm{d}}{\mathrm{d}\eta_n} \ln g(\eta_n) + \frac{t_n}{s} \right\} \frac{\mathrm{d}\eta_n}{\mathrm{d}y_n} \frac{\mathrm{d}y_n}{\mathrm{d}a_n} \nabla a_n \\ &= \sum_{n=1}^{N} \frac{1}{s} \{t_n - y_n\} \psi'(y_n) f'(a_n) \phi_n \end{aligned} \tag{7.103}$$

其中 $a_n = \boldsymbol{w}^T \phi_n$，并且我们使用了 $y_n = f(a_n)$ 以及 $\mathbb{E}[t|\eta]$ 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。

$$f^{-1}(y) = \psi(y) \tag{7.104}$$

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

$$\nabla E(\boldsymbol{w}) = \frac{1}{s} \sum_{n=1}^{N} \{y_n - t_n\} \phi_n \tag{7.105}$$

对于高斯分布，$s = \beta^{-1}$，而对于 logistic 模型，$s = 1$。

7.4 拉普拉斯近似

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。后面章节中，我们会介绍一系列分析估计和数值采样的技术。

拉普拉斯近似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

$$p(z) = \frac{1}{Z} f(z) \tag{7.106}$$

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

$$\left. \frac{df(z)}{dz} \right|_{z=z_0} = 0 \tag{7.107}$$

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

$$\ln f(z) \simeq \ln f(z_0) - \frac{1}{2}A(z - z_0)^2 \tag{7.108}$$

其中

$$A = -\frac{d^2}{dz^2} \ln f(z)\Big|_{z=z_0} \tag{7.109}$$

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

$$f(z) \simeq f(z_0) \exp\left\{-\frac{A}{2}(z - z_0)^2\right\} \tag{7.110}$$

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

$$q(z) = \left(\frac{A}{2\pi}\right)^{\frac{1}{2}} \exp\left\{-\frac{A}{2}(z - z_0)^2\right\} \tag{7.111}$$

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，使其近似定义在 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

$$\ln f(\boldsymbol{z}) \simeq \ln f(\boldsymbol{z}_0) - \frac{1}{2}(\boldsymbol{z} - \boldsymbol{z}_0)^T \boldsymbol{A}(\boldsymbol{z} - \boldsymbol{z}_0) \tag{7.112}$$

其中 $M \times M$ 的 Hessian 矩阵 $\boldsymbol{A}$ 的定义为

$$\boldsymbol{A} = -\nabla\nabla \ln f(\boldsymbol{z})|_{\boldsymbol{z}=\boldsymbol{z}_0} \tag{7.113}$$

两边同时取指数，我们有

$$f(\boldsymbol{z}) \simeq f(\boldsymbol{z}_0) \exp\left\{-\frac{1}{2}(\boldsymbol{z} - \boldsymbol{z}_0)^T \boldsymbol{A}(\boldsymbol{z} - \boldsymbol{z}_0)\right\} \tag{7.114}$$

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

$$q(\boldsymbol{z}) = \frac{|\boldsymbol{A}|^{\frac{1}{2}}}{(2\pi)^{\frac{M}{2}}} \exp\left\{-\frac{1}{2}(\boldsymbol{z} - \boldsymbol{z}_0)^T \boldsymbol{A}(\boldsymbol{z} - \boldsymbol{z}_0)\right\} = \mathcal{N}(\boldsymbol{z}|\boldsymbol{z}_0, \boldsymbol{A}^{-1}) \tag{7.115}$$

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。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位置上的性质，因此会无

法描述一些重要的全局属性。

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。

$$\begin{aligned} Z &= \int f(z)dz \\ &\simeq f(z_0)\int \exp\left\{-\frac{1}{2}(z-z_0)^T A(z-z_0)\right\}dz \\ &= f(z_0)\frac{(2\pi)^{\frac{M}{2}}}{|A|^{\frac{1}{2}}} \end{aligned} \tag{7.116}$$

我们可以使用公式 7.116 的结果来获得对于模型证据的一个近似。考虑一个数据集 D 以及一组模型 $\{M_i\}$，模型参数为 $\{\boldsymbol{\theta}_i\}$。对于每个模型，我们定义一个似然函数 $p(D|\boldsymbol{\theta}_i, M_i)$。如果我们引入一个参数的先验概率 $p(\boldsymbol{\theta}_i|M_i)$，那么我们感兴趣的是计算不同模型的模型证据 $p(D|M_i)$。从现在开始，为了简化记号，我们省略对于 M_i 的条件依赖。根据贝叶斯定理，模型证据为

$$p(D) = \int p(D|\boldsymbol{\theta})p(\boldsymbol{\theta})\mathrm{d}\boldsymbol{\theta} \tag{7.117}$$

令 $f(\boldsymbol{\theta}) = p(D|\boldsymbol{\theta})p(\boldsymbol{\theta})$ 以及 $Z = p(D)$，然后使用公式 7.116，我们有

$$\ln p(D) \simeq \ln p(D|\boldsymbol{\theta}_{MAP}) + \underbrace{\ln p(\boldsymbol{\theta}_{MAP}) + \frac{M}{2}\ln(2\pi) - \frac{1}{2}\ln|\boldsymbol{A}|}_{Occam因子} \tag{7.118}$$

其中 $\boldsymbol{\theta}_{MAP}$ 是在后验概率分布众数位置的 $\boldsymbol{\theta}$ 的值，$\boldsymbol{A}$ 是负对数后验概率的二阶导数组成的 Hessian 矩阵。

$$\boldsymbol{A} = -\nabla\nabla \ln p(\boldsymbol{\theta}_{MAP}|D) \tag{7.119}$$

公式 7.118 表明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 7.118。

$$\ln p(D) \simeq \ln p(D|\boldsymbol{\theta}_{MAP}) - \frac{1}{2}M\ln N \tag{7.120}$$

其中 N 是数据点的总数，M 是 $\boldsymbol{\theta}$ 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。注意，这与 AIC 相比，这个信息准则对模型复杂度的惩罚更严重。

7.5 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。

拉普拉斯近似

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 $\boldsymbol{w}_{MAP}$，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

$$\boldsymbol{S}_N^{-1} = -\nabla\nabla \ln p(\boldsymbol{w}|\boldsymbol{t}) = \boldsymbol{S}_0^{-1} + \sum_{n=1}^{N} y_n(1-y_n)\boldsymbol{\phi}_n\boldsymbol{\phi}^T \tag{7.121}$$

于是后验概率分布的高斯近似的形式为

$$q(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}|\boldsymbol{w}_{MAP}, \boldsymbol{S}_N) \tag{7.122}$$

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。

预测分布

给定一个新的特征向量 $\boldsymbol{\phi}(\boldsymbol{x})$，类别 C_1 的预测分布可以通过对后验概率 $p(\boldsymbol{w}|\boldsymbol{t})$ 积分，后验概率本身由高斯分布 $q(\boldsymbol{w})$ 近似，即

$$p(C_1|\boldsymbol{\phi}, \boldsymbol{t}) = \int p(C_1|\boldsymbol{\phi}, \boldsymbol{w})p(\boldsymbol{w}|\boldsymbol{t})d\boldsymbol{w} \simeq \int \sigma(\boldsymbol{w}^T\boldsymbol{\phi})q(\boldsymbol{w})d\boldsymbol{w} \tag{7.123}$$

类别 C_2 的对应的概率为 $p(C_2|\boldsymbol{\phi}, \boldsymbol{t}) = 1 - p(C_1|\boldsymbol{\phi}, \boldsymbol{t})$。

为了计算预测分布，我们首先注意到函数 $\sigma(\boldsymbol{w}^T\boldsymbol{\phi})$ 对于 $\boldsymbol{w}$ 的依赖只通过它在 $\boldsymbol{\phi}$ 上的投影而实现。记 $a = \boldsymbol{w}^T\boldsymbol{\phi}$，我们有

$$\sigma(\boldsymbol{w}^T\boldsymbol{\phi}) = \int \delta(a - \boldsymbol{w}^T\boldsymbol{\phi})\sigma(a)da \tag{7.124}$$

其中 $\delta(\cdot)$ 是狄拉克 Delta 函数。由此我们有

$$\int \sigma(\boldsymbol{w}^T\boldsymbol{\phi})q(\boldsymbol{w})d\boldsymbol{w} = \int \sigma(a)p(a)da \tag{7.125}$$

其中

$$p(a) = \int \delta(a - \boldsymbol{w}^T\boldsymbol{\phi})q(\boldsymbol{w})d\boldsymbol{w} \tag{7.126}$$

我们可以这样计算 $p(a)$：注意到 Delta 函数给 $\boldsymbol{w}$ 施加了一个线性限制，因此在所有与 ϕ 正交的方向上积分，就得到了联合概率分布 $q(\boldsymbol{w})$ 的边缘分布。由于 $q(\boldsymbol{w})$ 是高斯分布，因此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 $\boldsymbol{w}$ 的积分顺序的方式计算均值和协方差，即

$$\begin{aligned}\mu_a = \mathbb{E}[a] &= \int p(a)a\mathrm{d}a \\ &= \int \left[\int \delta(a - \boldsymbol{w}^T\phi)q(\boldsymbol{w})\mathrm{d}\boldsymbol{w}\right] a\mathrm{d}a \\ &= \int \left[\int \delta(a - \boldsymbol{w}^T\phi)a\mathrm{d}a\right] q(\boldsymbol{w})d\boldsymbol{w} \\ &= \int q(\boldsymbol{w})\boldsymbol{w}^T\phi\mathrm{d}\boldsymbol{w} \\ &= \boldsymbol{w}_{MAP}^T\phi\end{aligned} \tag{7.127}$$

推导时使用了 7.122 给出的后验概率分布 $q(\boldsymbol{w})$ 的结果。类似地

$$\begin{aligned}\sigma_a^2 = var[a] &= \int p(a)\{a^2 - \mathbb{E}[a]^2\}da \\ &= \int q(\boldsymbol{w})\{(\boldsymbol{w}^T\phi)^2 - (\boldsymbol{m}_N^T\phi)^2\}d\boldsymbol{w} = \phi^T\boldsymbol{S}_N\phi\end{aligned} \tag{7.128}$$

因此我们对于预测分布的近似变成了

$$p(C_1|\boldsymbol{t}) = \int \sigma(a)p(a)da = \int \sigma(a)\mathcal{N}(a|\mu_a, \sigma_a^2)da \tag{7.129}$$

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 logistic sigmoid 函数 $\sigma(a)$ 和逆 probit 函数 $\Phi(a)$ 的高度相似性来获得一个较好的近似。

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 解析地表示出来。特别地，我们可以证明

$$\int \Phi(\lambda a)\mathcal{N}(a|\mu, \sigma^2)da = \Phi\left(\frac{\mu}{(\lambda^{-2} + \sigma^2)^{\frac{1}{2}}}\right) \tag{7.130}$$

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 loigstic sigmoid 函数与高斯的卷积的近似

$$\int \sigma(a)\mathcal{N}(a|\mu, \sigma^2)da \simeq \sigma(k(\sigma^2)\mu) \tag{7.131}$$

其中我们定义了

$$k(\sigma^2) = (1 + \frac{\pi\sigma^2}{8})^{-\frac{1}{2}} \tag{7.132}$$

把这个结果应用于公式中，我们得到了近似的预测分布，形式为

$$p(C_1|\phi, \boldsymbol{t}) = \sigma(k(\sigma_a^2)\mu_a) \tag{7.133}$$

对应于 $p(C_1|\phi, \boldsymbol{t}) = 0.5$ 的决策边界由 $\mu_a = 0$ 给出，这与使用 $\boldsymbol{w}$ 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 $\boldsymbol{w}$ 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。

第 8 章 神经网络

在上两章，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，因此最优化问的解相对很直接，并且通常随着数据规模的增加而增多。相关向量机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，嘎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。换名话，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是有前馈神经网络，也被称为多层感知器 (multilayer perceptron)。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。

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和各种方法，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。

8.1 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 5 章和第 6 章讨论过了。它们基于固定非线性基函数 $\phi_j(\boldsymbol{x})$ 的线性组合，形式为

$$y(\boldsymbol{x}, \boldsymbol{w}) = f\left(\sum_{j=1}^{M} w_j \phi_j(\boldsymbol{x})\right) \tag{8.1}$$

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 $\phi_j(\boldsymbol{x})$ 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。

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。深度前馈网络 (deep feedforward network) 也叫做前馈神经网络 (feedforward neural network) 或者多层感知机 (multilayer perceptron,MLP)，是典型的深度学习模型。与感知器相比，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。前馈神经网络的目标是近似某个函数 f^*。前馈网络定义了一个映射 $y = f(x;\theta)$，并且学习参数 θ 的值，使它能够得到最佳的函数近似。

前馈神经网络之所以被称作网络，是因为它们通常用许多不同函数复合在一起来表示。该模型与一个有向无环图相关联，而图描述了函数是如何复合在一起的。例如，我们有三个函数 $f^{(1)}, f^{(2)}, f^{(3)}$ 连接在一个链上以形成 $f^{(3)}(f^{(2)}(f^{(1)}(x)))$。

在神经网络训练过程中，我们让 $f(x)$ 去匹配 $f^*(x)$ 的值。训练数据为我们提供了在不同训练点上取值的、含有噪声的 $f^*(x)$ 近似实例。每个样本 x 都伴随着一个标签 $y \approx f^*(x)$。训练样本直接指明了输出层在每一点 x 上必须做什么；它必须产生一个接近 y 的值。但是训练数据并没有直接指明其他层应该怎么做。学习算法必须决定如何使用这些层来诞生想要的输出，但是训练数据并没有说每个单独的层应该做什么。相反，学习算法必须决定如何使用这些层来最好地实现 f^* 的近似。因为训练数据并没有给出这些层中的每一层所需的输出，所以这些层被称为隐藏层 (hidden layer)。

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

$$y_k(\boldsymbol{x}, \boldsymbol{w}) = \sigma\left(\sum_{j=1}^{M} w_{kj}^{(2)} h\left(\sum_{i=1}^{D} w_{ji}^{(1)} x_i + w_{j0}^{(1)}\right) + w_{k0}^{(2)}\right) \tag{8.2}$$

如果网络中的所有隐含单元的激活函数都限线性函数，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。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。

神经网络很容易扩展，例如，可以增加额外的处理层；引入跨层 (skip-layer) 链接，此外网络可以是稀疏的。

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。

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 $\boldsymbol{w}$ 的选择，网络可能产生同样的从输入到输出的映射函数。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。

8.2 网络训练

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。

1. 对于回归问题，我们使用线性输出和平方和误差函数
2. 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
3. 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。

参数最优化

下面考虑寻找能够使得选定的误差函数 $E(\boldsymbol{w})$ 达到最小值的权向量 $\boldsymbol{w}$。由于误差 $E(\boldsymbol{w})$ 是 $\boldsymbol{w}$ 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

$$\nabla E(\boldsymbol{w}) = 0 \tag{8.3}$$

我们的目标是寻找一个向量 $\boldsymbol{w}$ 使得 $E(\boldsymbol{w})$ 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。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 $\nabla E(\boldsymbol{w}) = 0$ 的解析解，因此我们使用迭代的数值方法。大多数方法涉及到为权向量选择某个初始值 $\boldsymbol{w}_0$，然后在权空间中进行一系列移动，形式为

$$\boldsymbol{w}^{(\tau+1)} = \boldsymbol{w}^{(\tau)} + \triangle w^{(\tau)} \tag{8.4}$$

其中 τ 表示迭代次数。不同的算法涉及到权向量更新 $\triangle w^{(\tau)}$ 的不同选择。许多算法使用梯度信息，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。

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 $E(\boldsymbol{w})$ 在权空间某点 $\hat{\boldsymbol{w}}$ 处的泰勒展开

$$E(\boldsymbol{w}) \simeq E(\hat{\boldsymbol{w}}) + (\boldsymbol{w} - \hat{\boldsymbol{w}})^T \boldsymbol{b} + \frac{1}{2}(\boldsymbol{w} - \hat{\boldsymbol{w}})^T \boldsymbol{H}(\boldsymbol{w} - \hat{\boldsymbol{w}}) \tag{8.5}$$

$\boldsymbol{b}$ 被定义为 E 的梯度在 $\hat{\boldsymbol{w}}$ 处的值。

$$\boldsymbol{b} \equiv \nabla E|_{\boldsymbol{w}=\hat{\boldsymbol{w}}} \tag{8.6}$$

Hessian 矩阵 $\boldsymbol{H} = \nabla\nabla E$。所以，梯度的局部近似为

$$\nabla E \simeq \boldsymbol{b} + \boldsymbol{H}(\boldsymbol{w} - \hat{\boldsymbol{w}}) \tag{8.7}$$

考虑一个特殊情况：在误差函数最小值点 $\boldsymbol{w}^*$ 附近的局部二次近似。因为 $\nabla E = 0$

$$E(\boldsymbol{w}) \simeq E(\boldsymbol{w}^*) + \frac{1}{2}(\boldsymbol{w} - \boldsymbol{w}^*)^T \boldsymbol{H}(\boldsymbol{w} - \boldsymbol{w}^*) \tag{8.8}$$

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

$$\boldsymbol{H}\boldsymbol{u}_i = \lambda_i \boldsymbol{u}_i \tag{8.9}$$

其中特征向量 $\boldsymbol{u}_i$ 构成了完备的单位正交集合，即

$$\boldsymbol{u}_i^T \boldsymbol{u}_j = \delta_{ij} \tag{8.10}$$

现在把 $(\boldsymbol{w} - \boldsymbol{w}^*)$ 展开成特征值的线性组合的形式

$$\boldsymbol{w} - \boldsymbol{w}^* = \sum_i \alpha_i \boldsymbol{u}_i \tag{8.11}$$

误差函数可以写成

$$E(\boldsymbol{w}) = E(\boldsymbol{w}^*) + \frac{1}{2}\sum_i \lambda_i \alpha_i^2 \tag{8.12}$$

在最小值 $\boldsymbol{w}^*$ 的领域中，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 $\boldsymbol{u}_i$ 给出，长度与对应的特征值 λ_i 的平方根成反比。在新的坐标第中，基向量是特征向量 $\{\boldsymbol{u}_i\}$，E 为常数的轮廓线是以原点为中心的椭圆。对应的 D 维的结论是，在 $\boldsymbol{w}^*$ 处的 Hessian 矩阵是正定矩阵。

使用梯度信息

使用误差反向传播的方法可以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。使用梯度信息构成了训练神经网络的实际算法的基础。

梯度下降最优化

最简单的使用梯度信息的方法是，每次权值更新都是在负梯度方向上的一次小的移动，即

$$\boldsymbol{w}^{(\tau+1)} = \boldsymbol{w}^{(\tau)} - \eta \nabla E(\boldsymbol{w}^{(\tau)}) \tag{8.13}$$

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。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)。这些方法具有这样的性质：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。

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。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

$$E(\boldsymbol{w}) = \sum_{n=1}^{N} E_n(\boldsymbol{w}) \tag{8.14}$$

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 (stochasitc gradient descent)，使得权向量的更新每次只依赖于一个数据点，即

$$\boldsymbol{w}^{(\tau+1)} = \boldsymbol{w}^{(\tau)} - \eta \nabla E_n(\boldsymbol{w}^{(\tau)}) \tag{8.15}$$

这个更新在数据集上循环重复进行，并且即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。在线梯度下隆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。

8.3 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 $E(\boldsymbol{w})$ 的梯度的一种高效的方法。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)。

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。在每一个迭代过程中，我们可以区分这两个不同的阶段。

1. 第一个阶段，误差函数关于权值的导数必须被计算出来。
2. 第二个阶段，导数用于计算权值的调整量。

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。

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。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，由若干的求和式组成，每一项对应于训练集的一个数据点，即

$$E(\boldsymbol{w}) = \sum_{n=1}^{N} E_n(\boldsymbol{w}) \tag{8.16}$$

这里，我们要考虑的是计算 $\nabla E_n(\boldsymbol{w})$ 的问题。这可以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，

$$y_k = \sum_i w_{ki} x_i \tag{8.17}$$

对于一个特定的输入模式 n，误差函数的形式为

$$E_n = \frac{1}{2} \sum_k (y_{nk} - t_{nk})^2 \tag{8.18}$$

其中 $y_{nk} = y_k(\boldsymbol{x}_n, \boldsymbol{w})$。这个误差函数关于一个权值 w_{ji} 的梯度为

$$\frac{\partial E_n}{\partial w_{ji}} = (y_{nj} - t_{nj}) x_{ni} \tag{8.19}$$

它可以表示为链接 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_{nj} 的乘积。反向传播算法可以总结如下

1. 对于网络的一个输入向量 $\boldsymbol{x}_n$，使用下列进行正向传播，找到所有隐含单元和输出单元的激活。

$$a_j = \sum_i w_{ji} z_i \tag{8.20}$$

$$z_j = h(a_j) \tag{8.21}$$

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。

2. 计算所有输出单元的 δ_k

$$\delta_k = y_k - t_k \tag{8.22}$$

3. 获得网络中所有隐含单元的 δ_j

$$\begin{aligned}\delta_j \equiv \frac{\partial E_n}{\partial a_j} &= \sum_k \frac{\partial E_n}{\partial a_k}\frac{\partial a_k}{\partial a_j} \\ &= h'(a_j)\sum_k w_{kj}\delta_k\end{aligned} \tag{8.23}$$

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。注意，单元 k 可以包含其他的隐含单元和输出单元。

4. 计算导数

$$\frac{\partial E_n}{\partial w_{ji}} = \delta_j z_i \tag{8.24}$$

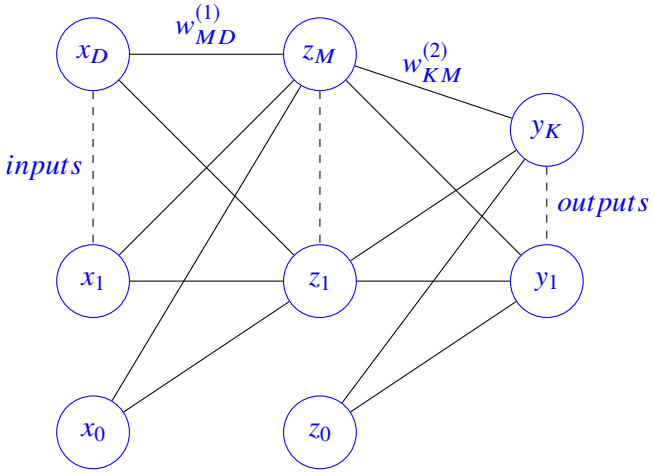
对于批处理方法，总误差函数 E 的导数可以通过下面的方式得到：对于训练集里的每个模式，重复上面的步骤，然后对所有的模式求和，即

$$\frac{\partial E}{\partial w_{ji}} = \sum_n \frac{\partial E_n}{\partial w_{ji}} \tag{8.25}$$

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相同的激活函数 $h(\cdot)$。

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。具体地，我们考虑两层神经网络



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 型函数，形式为

$$h(a) \equiv \tanh(a) = \frac{e^a - e^{-a}}{e^a + e^{-a}} \tag{8.26}$$

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的形式

$$h'(a) = 1 - h(a)^2 \tag{8.27}$$

也考虑一个标准的平方和误差函数，即对于模式 n，误差为

$$E_n = \frac{1}{2}\sum_{k=1}^{K}(y_k - t_k)^2 \tag{8.28}$$

其中，对于一个特定的输入模式 $\boldsymbol{x}_n$，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。

$$a_j = \sum_{i=0}^{D} w_{ji}^{(1)} x_i \tag{8.29}$$

$$z_j = \tanh(a_j) \tag{8.30}$$

$$y_k = \sum_{j=0}^{M} w_{kj}^{(2)} z_j \tag{8.31}$$

接下来，使用下面的公式计算每个输出单元的 δ 值

$$\delta_k = y_k - t_k \tag{8.32}$$

然后，将这些 δ 值反向传播，得到隐含单元的 δ 值

$$\delta_j = (1 - Z_j^2)\sum_{k=1}^{K} w_{kj}\delta_k \tag{8.33}$$

最后，关于第一层权值和第二层权值的导数为

$$\frac{\partial E_n}{\partial w_{ji}^{(1)}} = \delta_j x_i, \qquad \frac{\partial E_n}{\partial w_{ki}^{(2)}} = \delta_k z_i \tag{8.34}$$

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。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。

Jacobian 矩阵

误差反向传播技术也可以用来计算其他类型的导数。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

$$J_{ki} \equiv \frac{\partial y_k}{\partial x_i} \tag{8.35}$$

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。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 $\triangle x_i$ 在训练过的网络中传播，从而估计他们对于输出误差 $\triangle y_k$ 的贡献。二者的关系为

$$\triangle y_k \simeq \sum_i \frac{\partial y_k}{\partial x_i} \triangle x_i \tag{8.36}$$

只要 $|\triangle x_i|$ 较小，这个关系就成立。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

$$\begin{aligned} J_{ki} = \frac{\partial y_k}{\partial x_i} &= \sum_j \frac{\partial y_k}{\partial a_j} \frac{\partial a_j}{\partial x_i} \\ &= \sum_j w_{ji} \frac{\partial y_k}{\partial a_j} \end{aligned} \tag{8.37}$$

公式 8.37 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 $\frac{\partial y_k}{\partial a_j}$

$$\begin{aligned} \frac{\partial y_k}{\partial a_j} &= \sum_l \frac{\partial y_k}{\partial a_l} \frac{\partial a_l}{\partial a_j} \\ &= h'(a_j) \sum_l w_{lj} \frac{\partial y_k}{\partial a_l} \end{aligned} \tag{8.38}$$

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)。如果对于每个输出单元，我们都有各自的求导公式。

$$\frac{\partial y_k}{\partial a_l} = \delta_{kl} \sigma'(a_l) \qquad \text{sigmoid 函数} \tag{8.39}$$

$$\frac{\partial y_k}{\partial a_l} = \delta_{kl} y_k - y_k y_l \qquad \text{softmax 函数} \tag{8.40}$$

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 8.38 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 8.39 和公式 8.40。最后，使用公式 8.37 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。

8.4 Hessian 矩阵

反向传播也可以用来计算误差函数的二阶导数，形式为

$$\frac{\partial^2 E}{\partial w_{ji} \partial w_{lk}} \tag{8.41}$$

有时将所有的权值和偏置参数看成一个向量 (记作 $\boldsymbol{w}$) 的元素 w_i 更方便，此时二阶导数组成了 Hessian 矩阵 $\boldsymbol{H}$ 的元素 H_{ij}，其中 $i, j \in \{1, \ldots, W\}$，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

1. 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。
2. 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。
3. 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。
4. Hessian 矩阵是贝叶斯神经网络的拉普拉斯近似的核心。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。

对角近似

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，因此这样做之后，矩阵的逆矩阵很容易计算。Hessian 矩阵的对角线元素可以写成

$$\begin{aligned} \frac{\partial^2 E_n}{\partial w_{ji}^2} &= \frac{\partial^2 E_n}{\partial a_j^2} \frac{\partial a_j}{\partial w_{ji}} z_i + 0 \\ &= \frac{\partial^2 E_n}{\partial a_j^2} z_i^2 \end{aligned} \tag{8.42}$$

公式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

$$\begin{aligned} \frac{\partial^2 E_n}{\partial a_j^2} &= \frac{\partial}{\partial a_j} \left[\sum_k \frac{\partial E_n}{\partial a_k} \frac{\partial a_k}{\partial a_j} \right] \\ &= \frac{\partial}{\partial a_j} \left[h'(a_j) \sum_k w_{kj} \frac{\partial E_n}{\partial a_k} \right] \\ &= h'(a_j)^2 \sum_k \sum_{k'} w_{kj} w_{k'j} \frac{\partial^2 E_n}{\partial a_k \partial a_{k'}} + h''(a_j) \sum_k w_{kj} \frac{\partial E_n}{\partial a_k} \end{aligned} \tag{8.43}$$

如果忽略二阶导数中非对角线无素，那么我们有

$$\frac{\partial^2 E_n}{\partial a_j^2} = h'(a_j)^2 \sum_k w_{kj}^2 \frac{\partial^2 E_n}{\partial a_k^2} + h''(a_j) \sum_k w_{kj} \frac{\partial E_n}{\partial a_k} \tag{8.44}$$

注意，需要计算这个近似，所需的计算步骤为 $O(W)$，其中 W 是网络中权值和偏置的总数。对于原始的 Hessian 矩阵，计算的步骤为 $O(W^2)$。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。

外积近似

当神经网络应用于回归问题时，通常使用平方和误差函数。我们可以把 Hessian 矩阵写成下面的形式

$$\begin{aligned} \boldsymbol{H} = \nabla\nabla E &= \frac{\partial^2}{\partial y_n^2}\left[\frac{1}{2}\sum_{n=1}^{N}(y_n - t_n)^2\right] \\ &= \frac{\partial}{\partial y_n}\left[\nabla y_n \sum_{n=1}^{N}(y_n - t_n)\right] \\ &= \sum_{n=1}^{N} \nabla y_n (\nabla y_n)^T + \sum_{n=1}^{N}(y_n - t_n)\nabla\nabla y_n \end{aligned} \tag{8.45}$$

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，那么公式的第二项会很小，可以被忽略。我们就得到了 Levenberg-Marquardt 近似，或者称为外积近似 (outer product approximation)。形式为

$$\boldsymbol{H} \simeq \sum_{n=1}^{N} \boldsymbol{b}_n \boldsymbol{b}_n^T \tag{8.46}$$

其中 $\boldsymbol{b}_n \equiv \nabla a_n = \nabla y_n$，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，这可能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。需要强调的是，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

$$\boldsymbol{H} \simeq \sum_{n=1}^{N} y_n(1 - y_n)\boldsymbol{b}_n \boldsymbol{b}_n^T \tag{8.47}$$

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。

Hessian 矩阵的逆矩阵

使用外积近似可以提出一个计算 Hessian 矩阵的逆矩阵的高效方法。首先，我们用矩阵的记号写出外积近似，即

$$\boldsymbol{H} = \sum_{n=1}^{N} \boldsymbol{b}_n \boldsymbol{b}_n^T \tag{8.48}$$

其中，$\boldsymbol{b}_n \equiv \nabla_{\boldsymbol{w}} a_n$ 是数据点 n 产生的输出单元激活对梯度的贡献。

现上推导一个建立 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

$$\boldsymbol{H}_{L+1} = \boldsymbol{H}_L + \boldsymbol{b}_{L+1}\boldsymbol{b}_{L+1}^T \tag{8.49}$$

为了计算 Hessian 矩阵的逆矩阵，我们考虑 Sherman-Morrison-Woodbury 公式

$$(A + UV^T)^{-1} = A^{-1} - A^{-1}U(I_k + V^T A^{-1}U)^{-1}V^T A^{-1} \tag{8.50}$$

其中，$A \in \mathbb{R}^{n\times n}$ 非奇异，即 A^{-1} 存在，$U, V \in \mathbb{R}^{n\times k}$。当 $k = 1$ 时的特殊形式

$$\begin{aligned}(A + uv^T)^{-1} &= A^{-1} - A^{-1}u(1 + v^T A^{-1}u)^{-1}v^T A^{-1} \\ &= A^{-1} - \frac{A^{-1}uv^T A^{-1}}{1 + v^T A^{-1}u}\end{aligned} \tag{8.51}$$

SM 公式看似复杂，但可以通过求解以下线性方程组来推导出来：

$$(A + uv^T)x = b \tag{8.52}$$

式 8.52 两边同乘以 A^{-1}，令 $A^{-1}u = z, A^{-1}b = y$，则有

$$x + zv^T x = y \tag{8.53}$$

注意到 $V^T x$ 是标量，令 $a = v^T x$。式 8.53 两边同时乘以 v^T，得

$$a + v^T za = v^T y \tag{8.54}$$

由于式 8.54 中 $v^T z$ 和 $v^T y$ 都是标量，从而由式 8.54 可解得

$$a = \frac{v^T y}{1 + v^T z} \tag{8.55}$$

由式 8.53 和 y, z, a 的定义可得

$$\begin{aligned}x &= y - az \\ &= A^{-1}b - A^{-1}u(1 + v^T A^{-1}u)^{-1}v^T A^{-1}b \\ &= \left[A^{-1} - A^{-1}u(1 + v^T A^{-1}u)^{-1}v^T A^{-1}\right] b\end{aligned} \tag{8.56}$$

由 8.52 和 8.56 即可得 Sherman-Morrison 公式，即 8.51。

证明： 公式 8.50 两边同乘 $(A + UV^T)$

$$\begin{aligned} I_n &= (A + UV^T)\left[A^{-1} - A^{-1}U(I_k + V^T A^{-1}U)^{-1}V^T A^{-1}\right] \\ &= I_n + UV^T A^{-1} - U(I_k + V^T A^{-1}U)^{-1}V^T A^{-1} - UV^T A^{-1}U(I_k + V^T A^{-1}U)^{-1}V^T A^{-1} \\ &= I_n + UV^T A^{-1} - U(I_k + V^T A^{-1}U)(I_k + V^T A^{-1}U)^{-1}V^T A^{-1} \\ &= I_n \end{aligned} \tag{8.57}$$

Woodbury 恒等式的主要用途是当 $n > k$ 时把一个相对较大的 n 阶矩阵求逆的问题转化为求一个相对较小的 k 阶矩阵的逆矩阵。我们可以把公式看成在已知 A^{-1} 的情况下对 $A + \triangle A$ 求逆的工具。其中 $\triangle A$ 是一个低秩扰动。

如果我们令 $\boldsymbol{H}_L = A$，且 $\boldsymbol{b}_{L+1} = \boldsymbol{v}$，我们有

$$\boldsymbol{H}_{L+1}^{-1} = \boldsymbol{H}_L^{-1} - \frac{\boldsymbol{H}_L^{-1}\boldsymbol{b}_{L+1}\boldsymbol{b}_{L+1}^T\boldsymbol{H}_L^{-1}}{1 + \boldsymbol{b}_{L+1}^T\boldsymbol{H}_L^{-1}\boldsymbol{b}_{L+1}} \tag{8.58}$$

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 $\boldsymbol{H}_0$ 被选为 $\alpha\boldsymbol{I}$，其中 α 是一个较小的量，从而算法实际找的是 $\boldsymbol{H} + \boldsymbol{\aleph I}$ 的逆矩阵。如果对于 α 的精确值不是特别敏感。

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

$$\begin{aligned} \frac{\partial^2 E}{\partial w_{ji}\partial w_{lk}} = \frac{1}{4\epsilon^2}\{&E(w_{ji} + \epsilon, w_{lk} + \epsilon) - E(w_{ji} + \epsilon, w_{lk} - \epsilon) \\ &- E(w_{ji} - \epsilon, w_{lk} + \epsilon) + E(w_{ji} - \epsilon, w_{lk} - \epsilon)\} + O(\epsilon^2) \end{aligned} \tag{8.59}$$

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层数的方法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

$$\frac{\partial^2 E}{\partial w_{ji}\partial w_{lk}} = \frac{1}{2\epsilon}\{\frac{\partial E}{\partial w_{ji}}(w_{lk} + \epsilon) - \frac{\partial E}{\partial w_{ji}}(w_{lk} - \epsilon)\} + O(\epsilon^2) \tag{8.60}$$

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。

Hessian 矩阵的精确计算

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。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。

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

$$\delta_k = \frac{\partial E_n}{\partial a_k}, \quad M_{kk'} \equiv \frac{\partial^2 E_n}{\partial a_k \partial a_{k'}} \tag{8.61}$$

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

1. 两个权值都在第二层

$$\frac{\partial^2 E_n}{\partial w_{kj}^{(2)} \partial w_{k'j'}^{(2)}} = z_j z_{j'} M_{kk'} \tag{8.62}$$

2. 两个权值都在第一层

$$\begin{aligned} \frac{\partial^2 E_n}{\partial w_{ji}^{(1)} \partial w_{j'i'}^{(1)}} &= x_i x_{i'} h''(a_{j'}) I_{jj'} \sum_k w_{kj'}^{(2)} \delta_k \\ &\quad + x_i x_{i'} h'(a_{j'}) \sum_k \sum_{k'} w_{k'j'}^{(2)} w_{kj}^{(2)} M_{kk'} \end{aligned} \tag{8.63}$$

3. 每一层有一个权值

$$\frac{\partial^2 E_n}{\partial w_{ji}^{(1)} \partial w_{kj'}^{(2)}} = x_i h'(a_j) \left\{ \delta_k I_{j'j} + z^{j'} \sum_{k'} w_{k'j}^{(2)} M_{kk'} \right\} \tag{8.64}$$

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。

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 本身，而是 $\boldsymbol{H}$ 与某些向量 $\boldsymbol{v}$ 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 $\boldsymbol{v}^T\boldsymbol{H}$ 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 $\boldsymbol{v}^T\boldsymbol{H}$ 的高效方法。

为了完成这一点，我们首先注意到

$$\boldsymbol{v}^T\boldsymbol{H} = \boldsymbol{v}^T\nabla(\nabla E) \tag{8.65}$$

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，然后将公式应用于这些方程，得到一组计算 $\boldsymbol{v}^T\boldsymbol{H}$ 的正向传播和反向传播的方程。这对应于将微分算符 $\boldsymbol{v}^T\nabla$ 作用于原始的正向传播和反向传播的方程。使用记号 $\mathcal{R}\{\cdot\}$ 表示算符 $\boldsymbol{v}^T\nabla$，我们将遵从这个惯例。与之前一样，我们使用两层网络，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

$$a_j = \sum_i w_{ji}x_i \tag{8.66}$$

$$z_j = h(a_j) \tag{8.67}$$

$$y_k = \sum_j w_{kj}z_j \tag{8.68}$$

我们现在使用 $\mathcal{R}\{\cdot\}$ 作用于这些方程上，得到一组正向传播方程，形式为

$$\mathcal{R}\{a_j\} = \sum_i v_{ji}x_i \tag{8.69}$$

$$\mathcal{R}\{z_j\} = h'(a_j)\mathcal{R}\{a_j\} \tag{8.70}$$

$$\mathcal{R}\{y_k\} = \sum_j w_{kj}\mathcal{R}\{z_j\} + \sum_j v_{kj}z_j \tag{8.71}$$

其中，v_{ji} 是向量 $\boldsymbol{v}$ 中对应于权值 w_{ji} 的元素。

考虑的平方和误差函数，我们有下面的标准的反向传播表达式

$$\delta_k = y_k - t_k \tag{8.72}$$

$$\delta_j = h'(a_j)\sum_k w_{kj}\delta_k \tag{8.73}$$

与之前一样，我们将 $\mathcal{R}\{\cdot\}$ 作用于这些方程上，得到一组反向传播方程，形式为

$$\mathcal{R}\{\delta_k\} = \mathcal{R}\{y_k\} \tag{8.74}$$

$$\begin{aligned}\mathcal{R}\{\delta_j\} = {} & h''(a_j)\mathcal{R}\{a_j\}\sum_k w_{kj}\delta_k \\ & + h'(a_j)\sum_k v_{kj}\delta_k + h'(a_j)\sum_k w_{kj}\mathcal{R}\{\delta\}\end{aligned} \tag{8.75}$$

然后,我们有误差函数的一阶导数的方程

$$\frac{\partial E}{\partial w_{kj}} = \delta_k z_j \tag{8.76}$$

$$\frac{\partial E}{\partial w_{ji}} = \delta_j x_j \tag{8.77}$$

使用 $\mathcal{R}\{\cdot\}$ 作用在这些方程上,我们得到了下面的关于 $\boldsymbol{v}^T\boldsymbol{H}$ 的表达式

$$\mathcal{R}\left\{\frac{\partial E}{\partial w_{kj}}\right\} = \mathcal{R}\{\delta_k\}z_j + \delta_k\mathcal{R}\{z_j\} \tag{8.78}$$

$$\mathcal{R}\left\{\frac{\partial E}{\partial w_{ji}}\right\} = x_i\mathcal{R}\{\delta_j\} \tag{8.79}$$

算法的执行涉及到将新的变量 $\mathcal{R}\{a_j\},\mathcal{R}\{z_j\},\mathcal{R}\{\delta_j\}$ 引入到隐含单元, 将 $\mathcal{R}\{\delta_k\},\mathcal{R}\{y_k\}$ 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,$\boldsymbol{v}^T\boldsymbol{H}$ 的值由公式 8.78 和公式 8.79 给出。这种方法的一个好处是,计算 $\boldsymbol{v}^T\boldsymbol{H}$ 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。

8.5 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。M 控制了网络中参数 (权值和偏置) 的数量。然后, 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。有其他的方式控制神经网络的模型复杂度来避免过拟合。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

$$\tilde{E}(\boldsymbol{w}) = E(\boldsymbol{w}) + \frac{\lambda}{2}\boldsymbol{w}^T\boldsymbol{w} \tag{8.80}$$

这个正则化项也被称为权值衰减 (weight decay)。这样模型复杂度可以通过选择正则化系数 λ 来确定。正则化项可以表示为权值 $\boldsymbol{w}$ 上的零均值高斯先验分布的负对数。

相容的高斯先验

公式 8.80 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

$$z_j = h\left(\sum_i w_{ji}x_i + w_{j0}\right) \tag{8.81}$$

输出单元的激活为

$$y_k = \sum_j w_{kj} z_j + w_{k0} \tag{8.82}$$

假设我们对输入变量进行一个线性变换，形式为

$$x_i \to \tilde{x}_i = a x_i + b \tag{8.83}$$

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

$$w_{ji} \to \tilde{w}_{ji} = \frac{1}{a} w_{ji} \tag{8.84}$$

$$w_{j0} \to \tilde{w}_{j0} = w_{j0} - \frac{b}{a} \sum_i w_{ji} \tag{8.85}$$

类似地，网络的输出变量的线性变换

$$y_k \to \tilde{y}_k = c y_k + d \tag{8.86}$$

可以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

$$w_{kj} \to \tilde{w}_{kj} = c w_{kj} \tag{8.87}$$

$$w_{k0} \to \tilde{w}_{k0} = c w_{k0} + d \tag{8.88}$$

如果我们使用原始数据训练一个网络，还使用输入和 (或) 目标变换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

$$\frac{\lambda_1}{2} \sum_{w \in \mathcal{W}_1} w^2 + \frac{\lambda_2}{2} \sum_{w \in \mathcal{W}_2} w^2 \tag{8.89}$$

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：$\lambda_1 \to a^{\frac{1}{2}} \lambda_1, \lambda_2 \to c^{-\frac{1}{2}} \lambda_2$

正则化项 8.89 对应于下面形式的先验概率分布

$$p(\boldsymbol{w} | \alpha_1, \alpha_2) \propto \exp\left(-\frac{\alpha_1}{2} \sum_{w \in \mathcal{W}_1} w^2 - \frac{\alpha_2}{2} \sum_{w \in \mathcal{W}_2} w^2 \right) \tag{8.90}$$

注意，这种形式的先验是反常的 (improper)(不能被归一化)，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)。

更一般地，我们可以考虑权值被分为任意数量的组 W_k 的情况下的先验，即

$$p(\boldsymbol{w}) \propto \exp\left(-\frac{1}{2}\sum_k \alpha_k \|\boldsymbol{w}\|_k^2\right) \tag{8.91}$$

其中

$$\|\boldsymbol{w}\|_k^2 = \sum_{j\in W_k} w_j^2 \tag{8.92}$$

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，那么我们就得到了自动相关性确定 (automatic relevance determination) 的方法。

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，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，这样可以得到一个有着较好泛化性能的网络。

不变性

在许多模型识别的应用中，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，或者说应用具有不变性 (invariant)。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。这涉及到训练集里包含足够多的表示各种变换的效果的样本。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。这此方法大致可以分为四类。

1. 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。
2. 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。这引出了切线传播方法。
3. 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不变性。
4. 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。如卷积神经网络。

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。对于一个特定的输入向量 $\boldsymbol{x}_n$，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。考虑 $D = 2$ 的情形，假设变换由单一参数 ξ 控制 (例如，ξ 可能是旋转的一个角度)。那么被 $\boldsymbol{x}_n$ 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 $\boldsymbol{x}_n$ 上产生的向量为 $\boldsymbol{s}(\boldsymbol{x}_n, \xi)$，且 $\boldsymbol{s}(\boldsymbol{x}, 0) = \boldsymbol{x}$。这样曲线 M 的切线就由方向导数 $\boldsymbol{\tau} = \frac{\partial \boldsymbol{x}}{\partial \xi}$ 给出，且点 $\boldsymbol{x}_n$ 处的切线向量为

$$\boldsymbol{\tau}_n = \left.\frac{\partial \boldsymbol{s}(\boldsymbol{x}_n, \xi)}{\partial \xi}\right|_{\xi=0} \tag{8.93}$$

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

$$\left.\frac{\partial y_k}{\partial \xi}\right|_{\xi=0} = \sum_{i=1}^{D} \left.\frac{\partial y_k}{\partial x_i}\frac{\partial x_i}{\partial \xi}\right|_{\xi=0} = \sum_{i=1}^{D} J_{ki}\tau_i \tag{8.94}$$

其中 J_{ki} 为 Jacobian 矩阵 J 的第 (k, i) 个元素，公式 8.94 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

$$\tilde{E} = E + \lambda\Omega \tag{8.95}$$

其中 λ 是正则化系数，且

$$\Omega = \frac{1}{2}\sum_n\sum_k\left(\left.\frac{\partial y_{nk}}{\partial \xi}\right|_{\xi=0}\right)^2 = \frac{1}{2}\sum_n\sum_k\left(\sum_{i=1}^{D} J_{nk}\tau_{ni}\right)^2 \tag{8.96}$$

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。

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 8.96 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。

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。

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。

与上一节一样，我们要考虑由单一参数 ξ 控制的变换，且这个变换由函数 $\boldsymbol{s}(\boldsymbol{x}_n, \xi)$ 描

述，基中 $\boldsymbol{s}(\boldsymbol{x}_n, 0) = \boldsymbol{x}$。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)

$$E = \frac{1}{2}\iint \{y(\boldsymbol{x}) - t\}^2 p(t|\boldsymbol{x})p(\boldsymbol{x})\mathrm{d}\boldsymbol{x}\mathrm{d}t \tag{8.97}$$

这里，为了保持记号的简洁，我们考虑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

$$\tilde{E} = \frac{1}{2}\iiint \{y(\boldsymbol{s}(\boldsymbol{x}, \xi)) - t\}^2 p(t|\boldsymbol{x})p(\boldsymbol{x})p(\xi)\mathrm{d}\boldsymbol{x}\mathrm{d}t\mathrm{d}\xi \tag{8.98}$$

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

$$\begin{aligned}\boldsymbol{s}(\boldsymbol{x}, \xi) &= \boldsymbol{s}(\boldsymbol{x}, 0) + \xi\frac{\partial}{\partial\xi}\boldsymbol{s}(\boldsymbol{x}, \xi)\bigg|_{\xi=0} + \frac{\xi^2}{2}\frac{\partial^2}{\partial\xi^2}\bigg|_{\xi=0} + O(\xi^3)\\ &= \boldsymbol{x} + \xi\boldsymbol{\tau} + \frac{1}{2}\xi^2\boldsymbol{\tau}' + O(\xi^3)\end{aligned} \tag{8.99}$$

其中 $\boldsymbol{\tau}'$ 表示 $\boldsymbol{s}(\boldsymbol{x}, \xi)$ 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

$$y(\boldsymbol{s}(\boldsymbol{x}, \xi)) = y(\boldsymbol{x}) + \xi\boldsymbol{\tau}^T\nabla y(\boldsymbol{x}) + \frac{\xi^2}{2}\left[(\boldsymbol{\tau}')^T\nabla y(\boldsymbol{x}) + \boldsymbol{\tau}^T\nabla\nabla y(\boldsymbol{x})\boldsymbol{\tau}\right] + O(\xi^3) \tag{8.100}$$

代入平均误差函数 8.98，我们有

$$\begin{aligned}\tilde{E} = &\frac{1}{2}\iint \{y(\boldsymbol{x}) - t\}^2 p(t|\boldsymbol{x})p(\boldsymbol{x})\mathrm{d}\boldsymbol{x}\mathrm{d}t\\ &+ \mathbb{E}[\xi]\iint \{y(\boldsymbol{x}) - t\}\boldsymbol{\tau}^T\nabla y(\boldsymbol{x})p(t|\boldsymbol{x})p(\boldsymbol{x})\mathrm{d}\boldsymbol{x}\mathrm{d}t\\ &+ \mathbb{E}[\xi^2]\frac{1}{2}\iint \Big[\{y(\boldsymbol{x}) - t\}\{(\boldsymbol{\tau}')^T\nabla y(\boldsymbol{x}) + \boldsymbol{\tau}^T\nabla\nabla y(\boldsymbol{x})\boldsymbol{\tau}\}\\ &+(\boldsymbol{\tau}^T\nabla y(\boldsymbol{x}))^2\Big] p(t|\boldsymbol{x})p(\boldsymbol{x})\mathrm{d}\boldsymbol{x}\mathrm{d}t + O(\xi^3)\end{aligned} \tag{8.101}$$

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

$$\tilde{E} = E + \lambda\Omega \tag{8.102}$$

其中 E 是原始的平方和误差，正则化项 Ω 的形式为

$$\Omega = \frac{1}{2}\int \left[\{y(\boldsymbol{x}) - \mathbb{E}[t|\boldsymbol{x}]\}\{(\boldsymbol{\tau}')^T\nabla y(\boldsymbol{x}) + \boldsymbol{\tau}^T\nabla\nabla y(\boldsymbol{x})\boldsymbol{\tau}\} + (\boldsymbol{\tau}^T\nabla y(\boldsymbol{x}))^2\right] p(\boldsymbol{x})\mathrm{d}\boldsymbol{x} \tag{8.103}$$

我们可以进一步简化这个正则化项，如下所述，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 $\mathbb{E}[t|\boldsymbol{x}]$。根据公式 8.102，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

$$y(\boldsymbol{x}) = \mathbb{E}[t|\boldsymbol{x}] + O(\xi^2) \tag{8.104}$$

从而，正则化项中的第一项消失，剩下的项为

$$\Omega = \frac{1}{2}\int (\boldsymbol{\tau}^T \nabla y(\boldsymbol{x}))^2 p(\boldsymbol{x})\mathrm{d}\boldsymbol{x} \tag{8.105}$$

这等价于切线传播的正则化项。

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 $\boldsymbol{x} \to \boldsymbol{x} + \boldsymbol{\xi}$，那么正则化项的形式为

$$\Omega = \frac{1}{2}\int \|\nabla y(\boldsymbol{x})\|^2 p(\boldsymbol{x})\mathrm{d}\boldsymbol{x} \tag{8.106}$$

这被称为 Tikhonov 正则化。这个正则化项关于网络的权值的导数可以使用反向传播算法求出。我们看到，对于小的噪声，Tikhonov 正则化与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。

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。这就是卷积神经网络 (convolutional neural network) 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。我们知道，数字的各类对于平移、缩放以及旋转具有不变性。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。假如数据集充分大，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。这些想法被整合到了卷积神经网络中，通过下面的三种方式：

(1) 局部接收声场

(2) 权值共享

(3) 下采样

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。

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。

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。这里，我们考虑软权值共享 (soft weight sharing)。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。

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。我们可以将权值分成若干组，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

$$p(\boldsymbol{w}) = \prod_i p(w_i) \tag{8.107}$$

其中

$$p(w_i) = \sum_{j=1}^{M} \pi_j \mathcal{N}(w_i|\mu_j, \sigma_j^2) \tag{8.108}$$

π_j 为混合系数。取负对数，即可得到正则化函数，形式为

$$\Omega(\boldsymbol{w}) = -\sum_i \ln\left(\sum_{j=1}^{M} \pi_j \mathcal{N}(w_i|\mu_j, \sigma_j^2)\right) \tag{8.109}$$

从而，总的误差函数为

$$\tilde{E}(\boldsymbol{w}) = E(\boldsymbol{w}) + \lambda\Omega(\boldsymbol{w}) \tag{8.110}$$

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 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来完成这件事情。

为了最小化总的误差函数，难免计算出它关于各个可调节参数的参数是很有必要的。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。后验概率由贝叶斯定理给出，形式为

$$\gamma_j(w) = \frac{\pi_j \mathcal{N}(w_i|\mu_j, \sigma_j^2)}{\sum_k \pi_k \mathcal{N}(w_i|\mu_k, \sigma_k^2)} \tag{8.111}$$

这样，总的误差函数关于权值的导数为

$$\begin{aligned} \frac{\partial \tilde{E}}{\partial w_i} &= \frac{\partial E}{\partial w_i} + \lambda \frac{\partial \Omega}{\partial w_i} \\ &= \frac{\partial E}{\partial w_i} + \lambda \sum_j \gamma_j(w_i) \frac{(w_i - \mu_j)}{\sigma_j^2} \end{aligned} \tag{8.112}$$

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，拉力正比于给定权值的高斯分布的后验概率。

误差函数关于高斯分布的中心的导数

$$\frac{\partial \tilde{E}}{\partial \mu_j} = \lambda \sum_i \gamma_j(w_i) \frac{(\mu_j - w_i)}{\sigma_j^2} \tag{8.113}$$

效果是把 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。

关于方差的导数为

$$\frac{\partial \tilde{E}}{\partial \sigma_j} = \lambda \sum_i \gamma_j(w_i) \left(\frac{1}{\sigma_j} - \frac{(w_i - \mu_j)^2}{\sigma_j^3} \right) \tag{8.114}$$

效果是将 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。

关于混合系数 π_j 的导数，我们需要考虑下面的限制条件

$$\sum_j \pi_j = 1, \quad 0 \leqslant \pi_i \leqslant 1 \tag{8.115}$$

这个限制的产生，是因为我们把 π_i 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

$$\pi_j = \frac{exp(\eta_j)}{\sum_{k=1}^{M} exp(\eta_k)} \tag{8.116}$$

这样，正则化的误差函数关于 $\{\eta_j\}$ 的导数的形式为

$$\frac{\partial \tilde{E}}{\partial \eta_j} = \sum_i \{\pi_j - \gamma_j(w_i)\} \tag{8.117}$$

效果是 π_j 被拉向第 j 个高斯分量的平均后验概率。

8.6 混合密度网络

有监督学习的目标是对条件概率分布 $p(\boldsymbol{t}|\boldsymbol{x})$ 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，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。

正向问题通常对应于物理系统的因果关系，通常有唯一解。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。

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 $x_n + 0.3 \sim (2\pi x_n)$，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。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 $p(\boldsymbol{t}|\boldsymbol{x})$ 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 $\boldsymbol{x}$ 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 $\boldsymbol{x}$ 值。混合模型提供了一个通用的形式，用来对任意条件概率密度函数 $p(\boldsymbol{t}|\boldsymbol{x})$ 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

$$p(\boldsymbol{t}|\boldsymbol{x}) = \sum_{k=1}^{K} \pi_k(\boldsymbol{x}) \mathcal{N}(\boldsymbol{t}|\boldsymbol{\mu}_k(\boldsymbol{x}), \sigma_k^2(\boldsymbol{x})\boldsymbol{I}) \tag{8.118}$$

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 $\pi_k(\boldsymbol{x})$、均值 $\boldsymbol{\mu}_k(\boldsymbol{x})$ 以及方差 $\sigma_k^2(\boldsymbol{x})$，这些参数控制了以 $\boldsymbol{x}$ 作为输入的神经网络的输出。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。

如果混合模型中有 K 个分量，且 $\boldsymbol{t}$ 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 $\pi_k(\boldsymbol{x})$，有 K 个输出 (记作 a_k^σ) 确定核宽度 $\sigma_k(\boldsymbol{x})$，有 $K \times L$ 个输出 (记作 a_{kj}^μ) 确定核中心 $\boldsymbol{\mu}_k(\boldsymbol{x})$ 的分量 $\mu_{kj}(\boldsymbol{x})$。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。

$$\sum_{k=1}^{K} \pi_k(\boldsymbol{x}) = 1, \quad 0 \leqslant \pi_k(\boldsymbol{x}) \leqslant 1 \tag{8.119}$$

可以通过使用一组 softmax 输出来实现。

$$\pi_k(\boldsymbol{x}) = \frac{\exp(a_k^\pi)}{\sum_{l=1}^{K} \exp(a_l^\pi)} \tag{8.120}$$

类似地，方差必须满足 $\sigma_k^2(\boldsymbol{x}) \geqslant 0$，因此可以使用对应的网络激活的指数形式来表示，即

$$\sigma_k(\boldsymbol{x}) = \exp(a_k^\sigma) \tag{8.121}$$

最后，由于均值 $\boldsymbol{\mu}_k(\boldsymbol{x})$ 有实数分量，因此它们可以直接用网络的输出激活表示

$$\mu_{kj}(\boldsymbol{x}) = a_{kj}^\mu \tag{8.122}$$

混合密度网络的可调节参数由权向量 $\boldsymbol{w}$ 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

$$E(\boldsymbol{w}) = -\sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k(\boldsymbol{x}_n, \boldsymbol{w}) \mathcal{N}(\boldsymbol{t}_n | \boldsymbol{\mu}_k(\boldsymbol{x}_n, \boldsymbol{w}), \sigma_k^2(\boldsymbol{x}_n, \boldsymbol{w}) \boldsymbol{I} \right\} \tag{8.123}$$

8.7 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。

在多层神经网络的情况下，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。

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，也使用了一个具有完成协方差矩阵的高斯分布。但是，最完整的贝叶斯方法是基于拉普拉斯的方法，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布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。

后验参数分布

考虑从输入向量 $\boldsymbol{x}$ 预测单一连续目标变量 t 的问题。我们假设条件概率分布 $p(t|\boldsymbol{x})$ 是一个高斯分布，均值与 $\boldsymbol{x}$ 有关，由神经网络模型的输出 $y(\boldsymbol{x}, \boldsymbol{w})$ 确定，精度 β 为

$$p(t|\boldsymbol{x}, \boldsymbol{w}, \beta) = \mathcal{N}(t|y(\boldsymbol{x}, \boldsymbol{w}), \beta^{-1}) \tag{8.124}$$

类似地，我们将权值 $\boldsymbol{w}$ 的先验概率分布选为高斯分布，形式为

$$p(\boldsymbol{w}|\alpha) = \mathcal{N}(\boldsymbol{w}|0, \alpha^{-1}\boldsymbol{I}) \tag{8.125}$$

对于 N 次独立同分布的观测 $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$，对应的目标值集合 $\mathcal{D} = \{t_1, \ldots, t_N\}$，似然函数为

$$p(\mathcal{D}|\boldsymbol{w}, \beta) = \prod_{n=1}^{N} \mathcal{N}(t_n|y(\boldsymbol{x}_n, \boldsymbol{w}), \beta^{-1}) \tag{8.126}$$

因此最终的后验概率为

$$p(\boldsymbol{w}|\mathcal{D},\alpha,\beta) \propto p(\boldsymbol{w}|\alpha)p(\mathcal{D}|\boldsymbol{w},\beta) \tag{8.127}$$

由于 $y(\boldsymbol{w},\boldsymbol{x})$ 与 $\boldsymbol{w}$ 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。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

$$\ln p(\boldsymbol{w}|\mathcal{D}) = -\frac{\alpha}{2}\boldsymbol{w}^T\boldsymbol{w} - \frac{\beta}{2}\sum_{n=1}^{N}\{y(\boldsymbol{x}_n,\boldsymbol{w}) - t_n\}^2 + \text{常数} \tag{8.128}$$

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，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 $\boldsymbol{w}_{MAP}$

找到了众数，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。负对数后验概率的二阶导数为

$$\boldsymbol{A} = -\nabla\nabla \ln p(\boldsymbol{w}|\mathcal{D},\alpha,\beta) = \alpha\boldsymbol{I} + \beta\boldsymbol{H} \tag{8.129}$$

这里，$\boldsymbol{H}$ 是一个 Hessian 矩阵，由平方和误差函数关于 $\boldsymbol{w}$ 的分量组成。这样，后验概率对应的高斯近似形式为

$$q(\boldsymbol{w}|\mathcal{D}) = \mathcal{N}(\boldsymbol{w}_{MAP},\boldsymbol{A}^{-1}) \tag{8.130}$$

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

$$p(t|\boldsymbol{x},\mathcal{D}) = \int p(t|\boldsymbol{x},\boldsymbol{w})q(\boldsymbol{w}|\mathcal{D})d\boldsymbol{w} \tag{8.131}$$

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 $y(\boldsymbol{x},\boldsymbol{w})$ 与 $\boldsymbol{w}$ 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 $y(\boldsymbol{x},\boldsymbol{w})$ 发生变化造成的 $\boldsymbol{w}$ 幅度相比，后验概率分布的方差较小。这使得我们可以在 $\boldsymbol{w}_{MAP}$ 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

$$y(\boldsymbol{x},\boldsymbol{w}) \simeq y(\boldsymbol{x},\boldsymbol{w}_{MAP}) + \boldsymbol{g}^T(\boldsymbol{w} - \boldsymbol{w}_{MAP}) \tag{8.132}$$

其中，我们定义了

$$\boldsymbol{g} = \nabla_{\boldsymbol{w}} y(\boldsymbol{w},\boldsymbol{x})|_{\boldsymbol{w}=\boldsymbol{w}_{MAP}} \tag{8.133}$$

使用这个近似，我们现在得到了一个线性高斯模型，$p(\boldsymbol{w})$ 为高斯分布。并且，$p(t|\boldsymbol{w})$ 也是高斯分布，它的均值是 $\boldsymbol{w}$ 的线性函数，分布的形式为

$$p(t|\boldsymbol{x},\boldsymbol{w},\beta) \simeq \mathcal{N}(t|y(\boldsymbol{x},\boldsymbol{w}_{MAP}) + \boldsymbol{g}^T\underbrace{(\boldsymbol{w} - \boldsymbol{w}_{MAP})}_{\text{只有这里含有 } \boldsymbol{w}},\beta^{-1}) \tag{8.134}$$

于是,我们我们可以求出边缘分布 $p(t)$

$$p(t|\boldsymbol{x}, \mathcal{D}, \alpha, \beta) = \mathcal{N}(t|y(\boldsymbol{x}, \boldsymbol{w}_{MAP}), \sigma^2(\boldsymbol{x}) \tag{8.135}$$

其中,与输入相关的方差为

$$\sigma^2(\boldsymbol{x}) = \beta^{-1} + \boldsymbol{g}^T \boldsymbol{A}^{-1} \boldsymbol{g} \tag{8.136}$$

我们看到预测分布 $p(t|\boldsymbol{x}, \mathcal{D})$ 是一个高斯分布,它的均值由网络函数 $y(\boldsymbol{x}, \boldsymbol{w}_{MAP})$ 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 $\boldsymbol{x}$ 相关的项,表示由于模型参数 $\boldsymbol{w}$ 的不确定性造成的内插的不确定性。

超参数最优化

用于分类的贝叶斯神经网络

第 9 章 支持向量机

支持向量机(support vector machines,SVM)是一种二类分类模型。它的基本模型是定义在特征空间上的间隔最大的线性分类器,间隔最大使它有别于感知机;支持向量机还包括核技巧,这使它成为实质上的非线性分类器。支持向量机的学习策略就是间隔最大化,可形式化为一个求解凸二次规划(convex quadratic programming)的问题,也等价于正则化的合页损失函数的最小化问题。支持向量机的学习算法是求解凸二次规划的最优化算法。

支持向量机学习方法包含构建由简至繁的模型:线性可分支持向量机(linear support vector machine in linearly separable case)、线性支持向量机(linear support vector machine)及非线性支持向量机(non-linear support vector machine)。简单模型是复杂模型的基础,也是复杂模型的特殊情况。当训练数据线性可分时,通过软件间隔最大化(hard margin maximization),学习一个线性的分类器,即线性可分支持向量机,又称为硬间隔支持向量机;当训练数据近似线性可分时,通过软件间隔最大化(soft margin maximization),也学习一个线性的分类器,即线性支持向量机,又称谓软间隔支持向量机;当训练数据线性不可分时,通过使用核技巧(kernel trick)及软间隔最大化,学习非线性支持向量机。

当输入空间为欧氏空间或离散集合、特征空间为希尔伯特空间时,核函数(kernel function)表示将输入从输入空间映射到特征空间得到的特征向量之间的内积。通过使用核函数可以学习非线性支持微量机,等价于隐式地在高维的特征空间中学习线性支持向量机。这样的方法称为核技巧。核方法(kernel method)是比支持向量机更为一般的机器学习方法。

Cortes 与 Vapnik 提出线性支持向量机,Boser、Guyon 与 Vapnik 又引入核技巧,提出非线性支持向量机。

9.1 间隔与支持向量

给定训练样本集 $D = \{(x_1, y_1), \ldots, (x_m, y_m)\}, y_i \in \{-1, +1\}$,分类学习最基本的想法是基于训练集 D 在样本空间中找到一个划分超平面,将不同类别的样本分开。如图 9.1

样本空间中任意点 $\boldsymbol{x}$ 到超平面 $(\boldsymbol{w}, b)$ 的距离可写为

$$\gamma = \frac{y(w^T x + b)}{\|w\|} = \frac{y f(x)}{\|w\|}$$

注:$y f(x)$ 相当于 $|f(x)|$。

假设超平面 $(\boldsymbol{w}, b)$ 能将训练样本正确分类,即对于 $(x_i, y_i) \in D$,若 $y_i = +1$,则有 $\boldsymbol{w}^T x_i + b > 0$;若 $y_i = -1$,则有 $\boldsymbol{w}^T x_i + b < 0$。令

$$\begin{cases} \boldsymbol{w}^T x_i + b \geq +1, & y_i = +1; \\ \boldsymbol{w}^T x_i + b \leq -1, & y_i = -1. \end{cases} \tag{9.1}$$

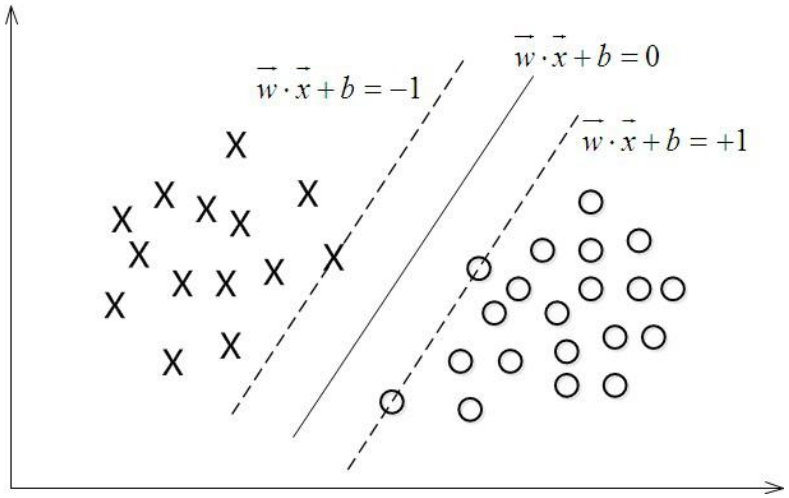


图 9.1

距离超平面最近的这几个训练样本点使式 9.1 的等号成立，它们被称为“**支持向量**(support vector)”，两个异类支持向量到超平面的距离之和为

$$\gamma = \frac{2}{\|w\|}, \tag{9.2}$$

它被称为“**间隔**”(margin)。

欲找到具有“最大间隔”的划分超平面，也就是要找到能满足式 9.1 中约束的参数 $\boldsymbol{w}$ 和 b，使得 γ 最大，即

$$\begin{aligned} min \quad & \frac{1}{2}\|w\|^2, \\ s.t. \quad & y_i(w^T x_i + b) \geq 1, \quad i = 1, \ldots, m. \end{aligned} \tag{9.3}$$

这就是支持向量机的基本型。

9.2 对偶问题

注意到式 9.3 本身是一个凸二次规划问题，能直接用现成的优化计算包求解，但我们可以有更高效的办法。由于这个问题的特殊结构，还可以通过拉格朗日对偶性变换到对偶变量的优化问题，即通过求解与原问题等价的对偶问题得到原始问题的最优解，这就是线性可分条件下支持向量机的对偶算法，这样做的优点在于：

1. 对偶问题往往更容易求解；
2. 可以自然的引入核函数，进而推广到非线性分类问题。

该问题的拉格朗日函数可写为

$$L(w, b, a) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{n} a_i(y_i(w^T x_i + b) - 1) \tag{9.4}$$

然后令

$$\theta(\boldsymbol{w}) = \max_{a_i \geq 0} L(\boldsymbol{w}, b, a) \tag{9.5}$$

具体写出来，目标函数变成了

$$\min_{\boldsymbol{w},b}\ \theta(\boldsymbol{w}) = \min_{\boldsymbol{w},b}\max_{a_i\geq 0}\ L(\boldsymbol{w},b,a) = p^* \tag{9.6}$$

这里用 p^* 表示这个问题的最优值，且和最初的问题是等价的。如果直接求解，那么一上来便得面对 w 和 b 两个参数，而 a_i 以是不等式约束，这个求解过程不好做。考虑对偶问题

$$\min_{\boldsymbol{w},b}\ \theta(\boldsymbol{w}) = \max_{a_i\geq 0}\ \min_{\boldsymbol{w},b}\ L(\boldsymbol{w},b,a) = d^* \tag{9.7}$$

原始问题通过满足 KKT 条件，已经转化成了对偶问题。而求解这个对偶问题，分为 3 个步骤

1. **让 $L(\boldsymbol{w},b,a)$ 关于 $\boldsymbol{w}$ 和 b 最小化**

   首先固定 a，要让 L 关于 w 和 b 最小化，分别对 w 和 b 求偏导，令其等于 0；

$$\begin{aligned}\frac{\partial L}{\partial \boldsymbol{w}} &= \|\boldsymbol{w}\| - \sum_{i=1}^{n} a_i y_i x_i \boldsymbol{w}^T = 0 \Rightarrow \boldsymbol{w} = \sum_{i=1}^{n} a_i y_i x_i \\ \frac{\partial L}{\partial b} &= \sum_{i=1}^{n} a_i y_i = 0 \Rightarrow \sum_{i=1}^{n} a_i y_i = 0\end{aligned} \tag{9.8}$$

   将以上结果代入之前的 L，得到

$$\begin{aligned}L(\boldsymbol{w},b,a) &= \frac{1}{2}\sum_{i,j=1}^{n} a_i a_j y_i y_j x_i^T x_j - \sum_{i,j=1}^{n} a_i a_j y_i y_j x_i^T x_j - b\sum_{i=1}^{n} a_i y_i + \sum_{i=1}^{n} a_i \\ &= \sum_{i=1}^{n} a_i - \frac{1}{2}\sum_{i,j=1}^{n} a_i a_j y_i y_j x_i^T x_j\end{aligned} \tag{9.9}$$

2. **求对 a 的极大**

   求对 a 的极大，即是关于对偶问题的最优化问题。经过上一个步骤的求解，得到的拉格朗日函数式子已经没有了变量 w 和 b，只有 a。从上面的式子得到

$$\begin{aligned}\max_{a}\quad & \sum_{i=1}^{n} a_i - \frac{1}{2}\sum_{i,j=1}^{n} a_i a_j y_i y_j x_i^T x_j \\ s.t.\quad & a_i \geq 0, i = 1,\ldots,n \\ & \sum_{i=1}^{n} a_i y_i = 0\end{aligned} \tag{9.10}$$

   这样，求出了 a_i，从而根据

$$\begin{aligned}\boldsymbol{w}^* &= \sum_{i=1}^{n} a_i y_i x_i \\ b^* &= -\frac{max\ \boldsymbol{w}^{*T} x_i + min\ \boldsymbol{w}^{*T} x_i}{2}\end{aligned} \tag{9.11}$$

即可求出 w, b，最终得出分离超平面和分类决策函数。

3. **利用 SMO 算法求解对偶问题中的拉格朗日乘子**

在求得 $L(w, b, a)$ 关于 w 和 b 最小化和对 a 的极大之后，最后一步便是利用 SMO 算法求解对偶问题中的拉格朗日乘子

9.3 序列最小最优算法

接上一小节，讨论支持向量机学习的实现问题。讲述其中的序列最小最优化 (sequential minimal optimization,SMO) 算法。

SMO 算法是一种启发式算法，其基本思路是：**如果所有变量的解都满足此最优化问题的 KKT 条件，那么这个最优化问题的解就得到了**。因为 KKT 条件是该最优化问题的充分必要条件。否则，选择两个变量，固定其他变量，针对这两个变量构建一个二次规划问题。这个二次规划问题关于这两个变量的解应该更接近原始二次规划问题的解，因为这会使得原始二次规划问题的目标函数值变得更小。重要的是，这时子问题可以通过解析方法求解，这样就可以大大提高整个算法的计算计算速度。子问题有两个变量，一个是违反 KKT 条件最严重的那一个，另一个由约束条件自动确定。如此，SMO 算法将原问题不断分解为子问题并对子问题求解，进而达到求解原问题的目的。

SMO 算法要解如下凸二次规划的对偶问题：

$$\min_{a} \quad \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} a_i a_j y_i y_j K(x_i, x_j) - \sum_{i=1}^{N} a_i \tag{9.12}$$

$$s.t. \quad C \geq a_i \geq 0, \quad i = 1, \ldots, n \tag{9.13}$$

$$\sum_{i=1}^{N} a_i y_i = 0 \tag{9.14}$$

注意，子问题的两个变量中只有一个是自由变量。假设 a_1, a_2 为两个变量，$a_3, a_4, \ldots, a_N$ 固定，那么由等式约束 9.14 可知

$$a_1 = -y_1 \sum_{i=2}^{N} a_i y_i \tag{9.15}$$

如果 a_2 确定，那么 a_1 也随之确定。所以子问题中同时更新两个变量。

整个 SMO 算法包括两个部分：**求解两个变量二次规划的解析方法和选择变量的启发式方法**。

于是 SMO 的最优化问题 9.12 的子问题可以写成

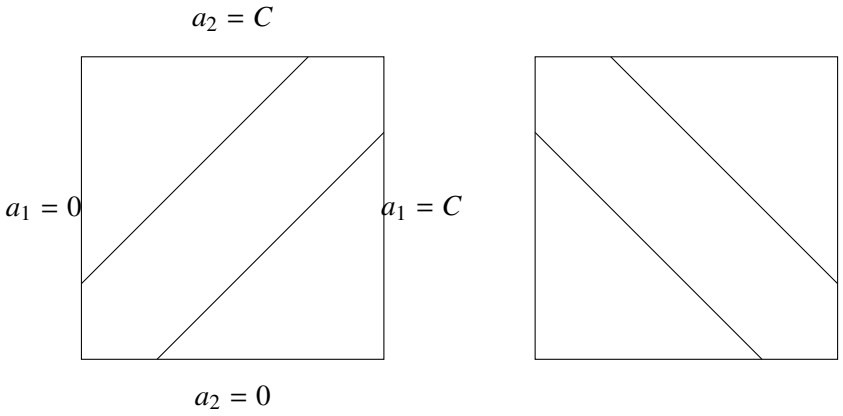
$$\min_{a_1,a_2} W(a_1,a_2)=\frac{1}{2}K_{11}a_1^2+\frac{1}{2}K_{22}a_2^2+y_1y_2K_{12}a_1a_2 -(a_1+a_2)+y_1a_1\sum_{i=3}^{N}y_ia_iK_{i1}+y_2a_2\sum_{i=3}^{N}y_ia_iK_{i2} \tag{9.16}$$

$$s.t\ a_1y_1+a_2y_2=-\sum_{i=3}^{N}y_ia_i=\varsigma \tag{9.17}$$

$$0\le a_i\le C,\ i=1,2 \tag{9.18}$$

其中，$K_{ij}=K(x_i,x_j),i,j=1,2,\ldots,N,\varsigma$ 是常数。

为了求解两个变量的二次规划问题 9.16，首先分析约束条件，然后在此约束条件下求极小。由于只有两个变量 (a_1,a_2)，约束可以用二维空间中的图形表示，如图所示



不等式约束使得 (a_1,a_2) 在盒子 $[0,C]\times[0,C]$ 内，等式约束使 (a_1,a_2) 在平行盒子的对角线的直线上。因此要求的是目标函数在一条平行于对角线线的线段上的最优值。这使得两个变量的最优化问题成为实质上的单变量的最优化问题，不妨考虑为变量 a_2 的最优化问题。

假设问题 9.16 的初始可行解为 a_1^{old},a_2^{old}，最优解为 a_1^{new},a_2^{new}，并且假设在沿着约束方向未经剪辑时 a_2 的最优解为 $a_2^{new,unc}$。

引进记号

$$g(x)=\sum_{i=1}^{N}a_iy_iK(x_i,x)+b \tag{9.19}$$

$$v_i=\sum_{j=3}^{N}y_ia_iK(x_i,x_j)=g(x_i)-\sum_{j=1}^{2}a_jy_jK(x_i,x_j)-b,\ i=1,2 \tag{9.20}$$

目标函数可写成

$$W(a_1,a_2)=\frac{1}{2}K_{11}a_1^2+\frac{1}{2}K_{22}a_2^2+y_1y_2K_{12}a_1a_2-(a_1+a_2)+y_1a_1v_1+y_2a_2v_2 \tag{9.21}$$

令

$$E_i=g(x_i)-y_i=\left(\sum_{j=1}^{N}a_jy_jK(x_j,x_i)+b\right)-y_i,\quad i=1,2 \tag{9.22}$$

当 $i=1,2$ 时，E_i 为函数 $g(x)$ 对输入 x_i 的预测值与真实输出 y_i 之差。由 $a_1y_1=\varsigma-a_2y_2$ 及 $y_i^2=1$，可将 a_1 表示为

$$a_1=(\varsigma-y_2a_2)y_1 \tag{9.23}$$

代入式 9.21，得到只是 a_2 的函数的目标函数，对 a_2 求导数

$$\frac{\partial W}{\partial a_2}=K_{11}a_2+K_{22}a_2-2K_{12}a_2-K_{11}\varsigma y_2+K_{12}\varsigma y_2+y_1y_2-1-v_1v_2+y_2v_2 \tag{9.24}$$

令其为 0，得到

$$\begin{aligned}(K_{11}+K_{22}-2K_{12})a_2=&y_2(y_2-y_1+\varsigma K_{11}-\varsigma K_{12}+v_1-v2)\\ =&y2\left[y_2-y_1+\varsigma K_{11}-\varsigma K_{12}+\left(g(x_1)-\sum_{j=1}^{2}y_ja_jK_{1j}-b\right)\right.\\ &\left.\left(g(x_2)-\sum_{j=1}^{2}y_ja_jK_{2j}-b\right)\right]\end{aligned} \tag{9.25}$$

将 $\varsigma=a_1^{old}y_1+a_2^{old}y_2$ 代入，得到

$$\begin{aligned}(K_{11}+K_{22}-2K_{12})a_2^{new,unc}&=y_2((K_{11}+K_{22}-2K_{12})a_2^{old}y_2+y_2-y_1+g(x_1)-g(x_2))\\ &=(K_{11}+K_{22}-2K_{12})a_2^{old}+y_2(E_1-E_2)\end{aligned} \tag{9.26}$$

将 $\eta=K_{11}+K_{22}-2K_{12}$ 代入，于是得到

$$a_2^{new,unc}=a_2^{old}+\frac{y_2(E_1-E_2)}{\eta} \tag{9.27}$$

要使其满足不等式的约束必须将其限制在区间 $[L,H]$ 内，从而得到 a_2^{new} 的表达式

$$a_2^{new}=\begin{cases}H, & a_2^{new,unc}>H\\ a_2^{new,unc}, & L\le a_2^{new,unc}\le H\\ L, & a_2^{new,unc}<L\end{cases} \tag{9.28}$$

如果 $y_1\neq y_2$

$$L=max(0,a_2^{old}-a_1^{old}),\quad H=min(C,C+a_2^{old}-a_1^{old})$$

如果 $y_1 = y_2$

$$L = max(0, a_2^{old} + a_1^{old} - C), \quad H = min(C, a_2^{old} + a_1^{old})$$

变量的选择方法

SMO 算法在每个子问题中选择两个变量优化，其中至少一个变量是违反 KKT 条件的。

1. 第 1 个变量的选择

   SMO 称选择第 1 个变量的过程为外层循环。外层循环在训练样本中选取违反 KKT 条件最严重的样本点，并将其对应的变量作为第 1 个变量。具体地，检验训练样本点 (x_i, y_i) 是否满足 KKT 条件，即

$$a_i = 0 \Leftrightarrow y_i g(x_i) \geq 1 \tag{9.29}$$

$$C > a_i > 0 \Leftrightarrow y_i g(x_i) = 1 \tag{9.30}$$

$$a_i = C \Leftrightarrow y_i g(x_i) \leq 1 \tag{9.31}$$

2. 第 2 个变量的选择

   SMO 称选择第 2 个变量的过程为内层循环，假设在外层循环中已经找到第 1 个变量 a_1，现在要在内层循环中找到第 2 个变量 a_2。第 2 个变量选择的标准是希望能使 a_2 有足够大的变化。由式 9.27 知，a_2^{new} 是依赖于 $|E_1 - E_2|$ 的，加了加快计算速度，一种简单的做法是选择 a_2，使其对应的 $|E_1 - E_2|$ 最大。

3. 计算阈值 b 和差值 E_i

9.4 核函数

SVM 处理线性可分的情况，而对于非线性的情况，SVM 的处理方法是选择一个核函数 $K < \cdot, \cdot >$，通过将数据映射到高维空间，来解决在原始空间中线性不可分的问题。

此外，用对偶形式表示学习器的优势在于该表示中可调参数的个数不依赖输入属性的个数，通过使用恰当的核函数来替代内积，可以隐式得将非线性的训练数据映射到高维空间，而不增加可调参数的个数。

在线性不可分的情况下，支持向量机首先在低维空间中完成计算，然后通过核函数将输入空间映射到高维牲空间，最终在高维特征空间中构造出最优分离超平面，从而把平面上本身不好分的非线性数据分开。

而在我们遇到核函数之前，如果用原始的方法，那么在用线性学习器学习一个非线性关系，需要选择一个非线性特征集，并且将数据写成新的表达形式，这等价于应用一个固定的非线性映射，将数据映射到特征空间，在特征空间中使用线性学习器，因此考虑的假设集是这种类型的函数：

$$f(x) = \sum_{i=1}^{N} w_i \phi_i(x) + b \tag{9.32}$$

这里 $\phi : X \rightarrow F$ 是从输入空间到某个特征空间的映射，这意味着建立非线性学习器分为

两步：

1. 首先使用一个非线性映射将数据变换到一个特征空间 F
2. 然后在特征空间使用线性学习器分类

而由于对偶形式就是线性学习器的一个重要性质，这意味着假设可以表达为训练点的线性组合，因此决策规则可以用测试点和训练点的内积来表示：

$$f(x) = \sum_{i=1}^{l} a_i y_i < \phi(x_i), \phi(x) > +b \tag{9.33}$$

如果有一种方式可以在特征空间中直接计算内积 $< \phi(x_i), \phi(x) >$，就像在原始输入点的函数中一样，就有可能将两个步骤融合到一起建立一个非线性的学习器，这样直接计算法的方法称为**核函数方法**

定义 9.1. Kernel

核是一个函数 K，对所有 $x, z \in X$，满足 $K(x, z) =< \phi(x), \phi(z) >$，这里 ϕ 是从 X 到内积特征空间 F 的映射。 ♣

下面举一个核函数把低维空间映射到高维空间的例子：

我们考虑核函数 $K(v_1, v_2) =< v_i, v_2 >^2$，即“内积平方”，这里 $v_1 = (x_1, y_1), v_2 = (x_2, y_2)$ 是二维空间中的两个点。这个核函数对应着一个二维空间到三维空间的映射，它的表达式是：

$$P(x, y) = (x^2, \sqrt{2}xy, y^2)$$

可以验证，

$$\begin{aligned} < P(v_1), p(v_2) > &=< (x_1^2, \sqrt{2}x_1y_1, y_1^2), (x_2^2, \sqrt{2}x_2y_2, y_2^2) > \\ &= x_1^2x_2^2 + 2x_1x_2y_1y_2 + y_1^2y_2^2 \\ &= (x_1x_2 + y_1y_2)^2 \\ &=< v_1, v_2 >^2 \\ &= K(v_1, v_2) \end{aligned}$$

上面的例子所说，核函数的作用就是隐含着一个从低维空间到高维空间的映射，而这个映射可以把低维空间中线性不可分的两类点变成线性可分的。

核函数的本质

1. 实际中，我们会经常遇到线性不可分的样例，此时，我们的常用做法是把特征映射到高维空间中去
2. 但进一步，如果凡是遇到线性不可分的样例，一律映射到高维空间，那么这个维度大小会高到可怕的。那咋办呢？
3. 此时，核函数就隆重登场了，核函数的价值在于它虽然也是讲特征进行从低维到高维的转换，但核函数绝就绝在它事先在低维上进行计算，而将实质上的分类效果表现在了高维上，也就避免了直接在高维空间中的复杂计算。

几个核函数：

- 多项式核 $K(x_1, x_2) = (< x_1, x_2 > +R)^d$
- 高斯核 $K(x_1, x_2) = exp(-\|x_1 - x_2\|^2/2\sigma^2)$
- 线性核 $K(x_1, x_2) =< x_1, x_2 >$

已知映射函数 ϕ，可以通过 $\phi(x)$ 和 $\phi(z)$ 的内积求得核函数 $K(x, z)$。不用构造映射 $\phi(x)$ 能否直接判断一个给定的函数 $K(x, z)$ 是不是核函数？或者说，函数 $K(x, z)$ 满足什么条件才能成为核函数？

已知，假设 $K(x, z)$ 是定义在 $\chi \times \chi$ 上的对称函数，并且对任意的 $x_1, \ldots, x_m \in \chi$，$K(x, z)$ 关于 $x_1, \ldots, x_m$ 的 Gram 矩阵是半正定的。可以依据函数 $K(x, z)$ 构成一个希尔伯特空间 (Hilbert space)，其步骤是：首先定义映射 ϕ 并构成向量空间 S；然后在 S 上定义内积构成内积空间；最后将 S 完备化构成希尔伯特空间。

定理 9.1. 正定核的充要条件

设 $K : \chi \times \chi \to \boldsymbol{R}$ 是对称函数，则 $K(x, z)$ 为正定核的充要条件是对任意 $x_i \in \chi, i = 1, 2, \ldots, m$，$K(x, z)$ 对应的 Gram 矩阵：

$$K = [K(x_i, x_j)]_{m \times m} \tag{9.34}$$

是半正定矩阵。 ♡

证明：

1. 必要性。由于 $K(x, z)$ 是 $\chi \times \chi$ 上的正定核，所以存在从 χ 到希尔伯特空间 $\mathcal{H}$ 的映射 ϕ，使得

$$K(x, z) = \phi(x) \cdot \phi(z) \tag{9.35}$$

于是，对任意 $x_1, \ldots, x_m$，构造 $K(x, z)$ 关于 $x_1, \ldots, x_m$ 的 Gram 矩阵

$$[K_{ij}]_{m \times m} = [K(x_i, x_j)]_{m \times m} \tag{9.36}$$

对任意 $c_1, \ldots, c_m \in \boldsymbol{R}$，有

$$\begin{aligned} \sum_{i,j=1}^{m} c_i c_j K(x_i, x_j) &= \sum_{i,j=1}^{m} c_i c_j (\phi(x_i) \cdot \phi(x_j)) \\ &= \left(\sum_{i,j=1}^{m} c_i \phi(x_i) \right) \cdot \left(\sum_{i,j=1}^{m} c_j \phi(x_j) \right) \\ &= \left\| \sum_{i} c_i \phi(x_i) \right\|^2 \geqslant 0 \end{aligned} \tag{9.37}$$

表明 $K(x, z)$ 关于 $x_1, \ldots, x_m$ 的 Gram 矩阵是半正定的。

2. 充分性根据已知，可以构造从 χ 到某个希尔伯特空间 $\mathcal{H}$ 的映射，从而表明 $K(x, z)$ 是 $\chi \times \chi$ 上的核函数。

9.5 软间隔与正则化

第 10 章 核方法

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。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。对于基于固定非线性特征空间 (feature space) 映射 $\phi(\boldsymbol{x})$ 的模型来说，核函数由下面的关系给出。

$$k(\boldsymbol{x}, \boldsymbol{x}') = \phi(\boldsymbol{x})^T \phi(\boldsymbol{x}') \tag{10.1}$$

核的概念由 Aizenman 引入模型识别领域。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 在边缘分类器的问题中把它重新引入到了机器学习领域。那篇文章提出了支持向量机的方法。从那里起，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。

10.1 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。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。

$$J(\boldsymbol{w}) = \frac{1}{2}\sum_{n=1}^{N}\{\boldsymbol{w}^T\phi(\boldsymbol{x}_n) - t_n\}^2 + \frac{\lambda}{2}\boldsymbol{w}^T\boldsymbol{w} \tag{10.2}$$

其中 $\lambda \geqslant 0$。如果我们令 $J(\boldsymbol{w})$ 关于 $\boldsymbol{w}$ 的梯度等于零，那么我们看到 $\boldsymbol{w}$ 的解是向量 $\phi(\boldsymbol{w})$ 的线性组合的形式，系数是 $\boldsymbol{w}$ 的函数，形式为

$$\begin{aligned}\frac{\partial J(\boldsymbol{w})}{\partial \boldsymbol{w}} &= \sum_{n=1}^{N}\{\boldsymbol{w}^T\phi(\boldsymbol{x}_n) - t_n\}\phi(\boldsymbol{x}_n) + \lambda\boldsymbol{w} \\ \Rightarrow \boldsymbol{w} &= -\frac{1}{\lambda}\sum_{n=1}^{N}\{\boldsymbol{w}^T\phi(\boldsymbol{x}_n) - t_n\}\phi(\boldsymbol{x}_n) \\ &= \sum_{n=1}^{N} a_n\phi(\boldsymbol{x}_n) = \Phi^T\boldsymbol{a}\end{aligned} \tag{10.3}$$

其中 Φ 是设计矩阵，第 n 行为 $\phi(\boldsymbol{x}_n)^T$。向量 $\boldsymbol{a} = (a_1, \ldots, a_N)^T$

$$a_n = -\frac{1}{\lambda}\{\boldsymbol{w}^T\phi(\boldsymbol{x}_n) - t_n\} \tag{10.4}$$

我们现在不直接对参数向量 $\boldsymbol{w}$ 进行操作，而是使用参数向量 $\boldsymbol{a}$ 重新整理最小平方算法，得到一个对偶表示。如果我们将 $\boldsymbol{w} = \Phi^T \boldsymbol{a}$ 代入 $J(\boldsymbol{w})$，那么可以得到

$$\begin{aligned} J(\boldsymbol{w}) &= \frac{1}{2}(\boldsymbol{w}^T\Phi^T - \boldsymbol{t}^T)(\Phi\boldsymbol{w} - \boldsymbol{t}) + \frac{\lambda}{2}\boldsymbol{w}^T\boldsymbol{w} \\ &= \frac{1}{2}\left[\boldsymbol{w}^T\Phi^T\Phi\boldsymbol{w} - \boldsymbol{w}^T\Phi^T\boldsymbol{t} - \boldsymbol{t}^T\Phi\boldsymbol{w} + \boldsymbol{t}^T\boldsymbol{t}\right] + \frac{\lambda}{2}\boldsymbol{w}^T\boldsymbol{w} \\ &\Rightarrow \text{把 } \boldsymbol{a} \text{ 看作参数，代入 } \boldsymbol{w}\Phi^T\boldsymbol{a} \\ J(\boldsymbol{a}) &= \frac{1}{2}\left[\boldsymbol{a}^T\Phi\Phi^T\Phi\Phi^T\boldsymbol{a} - \boldsymbol{a}\Phi\Phi^T\boldsymbol{t} - \boldsymbol{t}^T\Phi\Phi^T\boldsymbol{a} + \boldsymbol{t}^T\boldsymbol{t}\right] + \frac{\lambda}{2}\boldsymbol{a}^T\Phi\Phi^T\boldsymbol{a} \\ &= \frac{1}{2}\boldsymbol{a}^T\Phi\Phi^T\Phi\Phi^T\boldsymbol{a} - \boldsymbol{a}\Phi\Phi^T\boldsymbol{t} + \frac{1}{2}\boldsymbol{t}^T\boldsymbol{t} + \frac{\lambda}{2}\boldsymbol{a}^T\Phi\Phi^T\boldsymbol{a} \end{aligned} \tag{10.5}$$

其中 $\boldsymbol{t} = (t_1, \ldots, t_N)^T$。定义 Gram 矩阵 $K = \Phi\Phi^T$，它是一个 $N \times N$ 的对称矩阵，元素为

$$K_{nm} = \phi(\boldsymbol{x}_n)^T\Phi(\boldsymbol{x}_m) = k(\boldsymbol{x}_n, \boldsymbol{x}_m) \tag{10.6}$$

其中我们引入了公式 10.1 定义的核函数。使用 Gram 矩阵，平方和误差函数可以写成

$$J(\boldsymbol{a}) = \frac{1}{2}\boldsymbol{a}^T\boldsymbol{K}\boldsymbol{K}\boldsymbol{a} - \boldsymbol{a}^T\boldsymbol{K}\boldsymbol{t} + \frac{1}{2}\boldsymbol{t}^T\boldsymbol{t} + \frac{\lambda}{2}\boldsymbol{a}^T\boldsymbol{K}\boldsymbol{a} \tag{10.7}$$

由公式 10.4，求解 $\boldsymbol{a}$，我们有

$$\begin{aligned} \boldsymbol{a}^T &= -\frac{1}{\lambda}(\boldsymbol{w}^T\Phi^T - \boldsymbol{t}) \\ &\Rightarrow \lambda\boldsymbol{a}^T = \boldsymbol{t}^T - \boldsymbol{a}^T\Phi\Phi^T = \boldsymbol{t} - \boldsymbol{a}\boldsymbol{K} \\ &\Rightarrow \boldsymbol{a} = (K + \lambda\boldsymbol{I}_N)^{-1}\boldsymbol{t} \end{aligned} \tag{10.8}$$

如果我们将这个代入线性回归模型中，对于新的输入 $\boldsymbol{x}$，我们得到了下面预测

$$y(\boldsymbol{x}) = \boldsymbol{w}^T\phi(\boldsymbol{x}) = \boldsymbol{a}^T\Phi\phi(\boldsymbol{x}) = k(\boldsymbol{x})^T(\boldsymbol{K} + \lambda\boldsymbol{I}_N)^{-1}\boldsymbol{t} \tag{10.9}$$

其中我们定义了向量 $k(\boldsymbol{x})$，它的元素为 $k_n(\boldsymbol{x}) = k(\boldsymbol{x}_n, \boldsymbol{x})$。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 $k(\boldsymbol{x}, \boldsymbol{x}')$ 表示。这被称为对偶公式，因为 $\boldsymbol{a}$ 的解可以被表示为 $\phi(\boldsymbol{x})$ 的线性组合，从而我们可以使用参数向量 $\boldsymbol{w}$ 恢复出原始的公式。

对偶公式的优点是，它可以完全通过核函数 $k(\boldsymbol{x}, \boldsymbol{x}')$ 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 $\phi(\boldsymbol{x})$，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。后面，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。

10.2 构造核

为了利用核技巧,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 $\phi(\boldsymbol{x})$,然后利用这个映射寻找对应的核。一维空间的核函数被定义为

$$k(x, x^{'}) = \phi(x)^T \phi(x) = \sum_{i=1}^{M} \phi_i(x)\phi_i(x^{'}) \tag{10.10}$$

其中 $\phi_i(x)$ 是基函数。

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 (或能是无穷维) 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

$$k(\boldsymbol{x}, \boldsymbol{z}) = (\boldsymbol{x}^T \boldsymbol{z})^2 \tag{10.11}$$

如果我们取二维输入空间 $\boldsymbol{x} = (x_1, x_2)$ 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

$$\begin{aligned} k(\boldsymbol{x}, \boldsymbol{z}) &= (\boldsymbol{x}^T \boldsymbol{z})^2 = (x_1 z_1 + x_2 z_2)^2 \\ &= x_1^2 z_1^2 + 2x_1 z_1 x_2 z_2 + x_2^2 z_2^2 \\ &= (x_1^2, \sqrt{2}x_1 x_2, x_2^2)(z_1^2, \sqrt{2}z_1 z_2, z_2^2)^T \\ &= \phi(\boldsymbol{x})^T \phi(\boldsymbol{z}) \end{aligned} \tag{10.12}$$

我们看到特征映射的形式为 $\phi(\boldsymbol{x}) = (x_1^2, \sqrt{2}x_1 x_2, x_2^2)$,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 $\phi(\boldsymbol{x})$。核函数是一个合法的核函数的充分必要条件是 Gram 矩阵在所有的集合在所有的集合 $\{\boldsymbol{x}_n\}$ 的选择下都是半正定的。

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 $k_1(\boldsymbol{x}, \boldsymbol{x}^{'})$ 和 $k_2(\boldsymbol{x}, \boldsymbol{x}^{'})$,下面的新核也是合

法的

$$k(\boldsymbol{x}, \boldsymbol{x}') = ck_1(\boldsymbol{x}, \boldsymbol{x}') \tag{10.13}$$

$$k(\boldsymbol{x}, \boldsymbol{x}') = f(\boldsymbol{x})k_1(\boldsymbol{x}, \boldsymbol{x}')f(\boldsymbol{x}') \tag{10.14}$$

$$k(\boldsymbol{x}, \boldsymbol{x}') = q(k_1(\boldsymbol{x}, \boldsymbol{x}')) \tag{10.15}$$

$$k(\boldsymbol{x}, \boldsymbol{x}') = \exp(k_1(\boldsymbol{x}, \boldsymbol{x}')) \tag{10.16}$$

$$k(\boldsymbol{x}, \boldsymbol{x}') = k_1(\boldsymbol{x}, \boldsymbol{x}') + k_2(\boldsymbol{x}, \boldsymbol{x}') \tag{10.17}$$

$$k(\boldsymbol{x}, \boldsymbol{x}') = k_1(\boldsymbol{x}, \boldsymbol{x}')k_2(\boldsymbol{x}, \boldsymbol{x}') \tag{10.18}$$

$$k(\boldsymbol{x}, \boldsymbol{x}') = k_3(\phi(\boldsymbol{x}, \phi(\boldsymbol{x}'))) \tag{10.19}$$

$$k(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x}' \tag{10.20}$$

$$k(\boldsymbol{x}, \boldsymbol{x}') = k_a(\boldsymbol{x}_a, \boldsymbol{x}_a') + k_b(\boldsymbol{x}_b, \boldsymbol{x}_b') \tag{10.21}$$

$$k(\boldsymbol{x}, \boldsymbol{x}') = k_a(\boldsymbol{x}_a, \boldsymbol{x}_a')k_b(\boldsymbol{x}_b, \boldsymbol{x}_b') \tag{10.22}$$

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，$\phi(\boldsymbol{x})$ 是一个从 $\boldsymbol{x}$ 到 $\mathbb{R}^M$ 的函数，$k_3(\cdot, \cdot)$ 是 $\mathbb{R}^M$ 中的一个合法的核，$\boldsymbol{A}$ 是一个对称半正定矩阵，$\boldsymbol{x}_a$ 和 $\boldsymbol{x}_b$ 是变量，且 $\boldsymbol{x} = (\boldsymbol{x}_a, \boldsymbol{x}_b)$。$K_a, K_b$ 是各自空间的合法的核函数。

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。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，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，并且在隐马尔可夫模型的情况下，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收了一些人的兴趣。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 $p(\boldsymbol{x})$，我们可以定义一个核

$$k(\boldsymbol{x}, \boldsymbol{x}') = p(\boldsymbol{x})p(\boldsymbol{x}') \tag{10.23}$$

很明显，这是一个合法的核，因为我们可以把它看成由映射 $p(\boldsymbol{x})$ 定义的一维特征空间中的一个内积。它表明，如果两个输入 $\boldsymbol{x}$ 和 $\boldsymbol{x}'$ 都具有较高的概率，那么它们就是相似的。扩展这类核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

$$k(\boldsymbol{x}, \boldsymbol{x}') = \sum_i p(\boldsymbol{x}|i)p(\boldsymbol{x}'|i)p(i) \tag{10.24}$$

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 $\boldsymbol{x}$ 和 $\boldsymbol{x}'$ 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

$$k(\boldsymbol{x}, \boldsymbol{x}') = \int p(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{x}'|\boldsymbol{z})p(\boldsymbol{z})d\boldsymbol{z} \tag{10.25}$$

其中 $\boldsymbol{z}$ 是一个连续潜在变量。

另一个使用生成式模型定义核函数的方法被称为 Fisher 核。

10.3 径向基函数网络

讨论了基于固定基函数的线性组合的回归模型，但是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 $\boldsymbol{\mu}_j$ 之间的径向距离 (通常是欧几里德距离)，即 $\phi_j(\boldsymbol{x}) = h(\|\boldsymbol{x} - \boldsymbol{\mu}_j\|)$。历史上，径向基函数被用来进行精确的函数内插。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。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。如果输入变量 $\boldsymbol{x}$ 上的噪声由一个服从分布 $v(\boldsymbol{\xi})$ 的变量 $\boldsymbol{\xi}$，那么平方和误差函数就变成了

$$E = \frac{1}{2}\sum_{n=1}^{N}\int \{y(\boldsymbol{x}_n + \boldsymbol{\xi}) - t_n\}^2 v(\boldsymbol{\xi})d\boldsymbol{\xi} \tag{10.26}$$

使用变分法，我们可以关于函数 $y(\boldsymbol{x})$ 进行最优化，得到

$$y(\boldsymbol{x}) = \sum_{n=1}^{N} t_n h(\boldsymbol{x} - \boldsymbol{x}_n) \tag{10.27}$$

其中基函数为

$$h(\boldsymbol{x} - \boldsymbol{x}_n) = \frac{v(\boldsymbol{x} - \boldsymbol{x}_n)}{\sum_{n=1}^{N} v(\boldsymbol{x} - \boldsymbol{x}_n)} \tag{10.28}$$

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。

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。

Nadaraya-Watson 模型

对于新的输入 $\boldsymbol{x}$，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”给出，其中等价核满足加和限制。我们可以从核密度估计开始，以一个不同的角度研究该回归模型。假设我们有一个训练集 $\{\boldsymbol{x}_n, t_n\}$，我们使用 Parzen 密度估计来对联合分布 $p(\boldsymbol{x}, t)$ 进行建模，即

$$p(\boldsymbol{x}, t) = \frac{1}{N}\sum_{n=1}^{N} f(\boldsymbol{x} - \boldsymbol{x}_n, t - t_n) \tag{10.29}$$

其中 $f(\boldsymbol{x},t)$ 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 $y(\boldsymbol{x})$ 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

$$\begin{aligned} y(\boldsymbol{x}) = \mathbb{E}[t|\boldsymbol{x}] &= \int_{-\infty}^{\infty} tp(t|\boldsymbol{x})dt \\ &= \int_{-\infty}^{\infty} t\left(\frac{p(\boldsymbol{x},t)}{p(\boldsymbol{x})}\right)dt \\ &= \frac{\int tp(\boldsymbol{x},t)dt}{\int p(\boldsymbol{x},t)dt} \\ &= \frac{\sum_n \int tf(\boldsymbol{x}-\boldsymbol{x}_n, t-t_n)dt}{\sum_m \int f(\boldsymbol{x}-\boldsymbol{x}_m, t-t_m)dt} \end{aligned} \tag{10.30}$$

简单起见，我们现在假设分量的密度函数的均值为零，即

$$\int f(\boldsymbol{x},t)tdt = 0 \tag{10.31}$$

对所有 $\boldsymbol{x}$ 都成立。使用一个简单的变量替换，我们有

$$\begin{aligned} y(\boldsymbol{x}) &= \frac{\sum_n g(\boldsymbol{x}-\boldsymbol{x}_n)t_n}{\sum_m g(\boldsymbol{x}-\boldsymbol{x}_m)} \\ &= \sum_n k(\boldsymbol{x},\boldsymbol{x}_n)t_n \end{aligned} \tag{10.32}$$

其中 $n,m = 1,\ldots,N$，且核函数 $k(\boldsymbol{x},\boldsymbol{x}_n)$ 为

$$\frac{g(\boldsymbol{x}-\boldsymbol{x}_n)}{\sum_m g(\boldsymbol{x}-\boldsymbol{x}_m)} \tag{10.33}$$

并且我们定义了

$$g(\boldsymbol{x}) = \int f(\boldsymbol{x},t)dt \tag{10.34}$$

公式 10.32 给出的结果被称为 Nadaraya-Watson 模型，或者称为核回归 (kernel regression)。对于一个局部核函数，它的性质为：给距离 $\boldsymbol{x}$ 较近的数据点 $\boldsymbol{x}_n$ 较高的权重。

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 $p(t,\boldsymbol{x})$ 建模，然后找到对应的条件概率分布 $p(t|\boldsymbol{x})$。

10.4 高斯过程

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。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。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。

重新考虑线性回归问题

为了引出高斯过程的观点，我们回到线性回归的例子中，通过对函数 $y(\boldsymbol{x}\boldsymbol{w})$ 的计算，重新推导出预测分布。

考虑一个模型 M，它被定义为由向量 $\phi(\boldsymbol{x})$ 的元素给出的 M 个固定基函数的线性组合，即

$$y(\boldsymbol{x}) = \boldsymbol{w}^T \phi(\boldsymbol{x}) \tag{10.35}$$

其中 $\boldsymbol{x}$ 是输入向量，$\boldsymbol{w}$ 是 M 维权向量。现在，考虑 $\boldsymbol{w}$ 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

$$p(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}|0, \alpha^{-1}\boldsymbol{I}) \tag{10.36}$$

它由一个超参数 α 控制，这个超参数表示分布的精度。对于任意给定的 $\boldsymbol{w}$，公式 10.35 定义了 $\boldsymbol{x}$ 的一个特定的函数。于是 $\boldsymbol{w}$ 上的概率分布就产生了一个函数 $y(\boldsymbol{x})$ 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 $\boldsymbol{x}$ 处的函数值，例如在训练数据点 $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ 处的函数值。于是我们感兴趣的是函数值 $y(\boldsymbol{x}_1), \ldots, y(\boldsymbol{x}_N)$ 的概率分布。我们把函数值的集合记作向量 $\boldsymbol{y}$。根据公式 10.35，这个向量等于

$$\boldsymbol{y} = \Phi\boldsymbol{w} \tag{10.37}$$

其中 Φ 是设计矩阵，元素为 $\Phi_{nk} = \phi_k(\boldsymbol{x}_n)$。我们可以用下面的方式找到 $\boldsymbol{y}$ 的概率分布。首先，我们注意到 $\boldsymbol{y}$ 是由 $\boldsymbol{w}$ 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。

$$\mathbb{E}[y] = \Phi\mathbb{E}[\boldsymbol{w}] = 0 \tag{10.38}$$

$$cov[\boldsymbol{y}] = \mathbb{E}[\boldsymbol{y}\boldsymbol{y}^T] - 0 = \Phi\mathbb{E}[\boldsymbol{w}\boldsymbol{w}^T]\Phi^T = \frac{1}{\alpha}\Phi\Phi^T = \boldsymbol{K} \tag{10.39}$$

其中，$\boldsymbol{K}$ 是 Gram 矩阵，元素为

$$K_{nm} = k(\boldsymbol{x}_n, \boldsymbol{x}_m) = \frac{1}{\alpha}\phi(\boldsymbol{x}_n)^T\phi(\boldsymbol{x}_m) \tag{10.40}$$

$k(\boldsymbol{x}, \boldsymbol{x}')$ 是核函数。

这个模型给我们提供了高斯过程的一个具体的例子。通常来说，高斯过程被定义为函数 $y(\boldsymbol{x})$ 上的一个概率分布，使得在任意点集 $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ 处计算的 $y(\boldsymbol{x})$ 的值的集合联合起来服从高斯分布。

高斯随机过程的一个关键点是 N 个变量 $y_1, \ldots, y_N$ 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 $y(\boldsymbol{x})$ 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 $p(\boldsymbol{w}|\alpha)$ 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 $\boldsymbol{x}$ 处的函数值 $y(\boldsymbol{x})$ 的协方差来完成。这个协方差由核函数确定

$$\mathbb{E}[y(\boldsymbol{x}_n)y(\boldsymbol{x}_m)] = k(\boldsymbol{x}_n, \boldsymbol{x}_m) \tag{10.41}$$

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。

用于回归的高斯过程

为了把高斯过程模型应用一回归模型，我们需要考虑观测目标值的噪声，形式为

$$t_n = y_n + \epsilon_n \tag{10.42}$$

其中 $y_n = y(\boldsymbol{x}_n), \epsilon_n$ 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

$$p(t_n|y_n) = \mathcal{N}(t_n|y_n, \beta^{-1}) \tag{10.43}$$

其中 β^{-1} 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 $\boldsymbol{y} = (y_1, \ldots, y_N)^T$ 为条件，目标值 $\boldsymbol{t} = (t_1, \ldots, t_N)^T$ 的联合概率分布是一个各向同性的高斯分布，形式为

$$p(\boldsymbol{t}|\boldsymbol{y}) = \mathcal{N}(\boldsymbol{t}|\boldsymbol{y}, \beta^{-1}\boldsymbol{I}_N) \tag{10.44}$$

根据高斯过程的定义，边缘概率分布 $p(\boldsymbol{y})$ 是一个高斯分布，均值为零，协方差由 Gram 矩阵 $\boldsymbol{K}$ 定义，即

$$p(\boldsymbol{y}) = \mathcal{N}(\boldsymbol{y}|0, \boldsymbol{K}) \tag{10.45}$$

确定 $\boldsymbol{K}$ 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 $\boldsymbol{x}_n$ 和 $\boldsymbol{x}_m$，对应的值 $y(\boldsymbol{x}_n)$ 和 $y(\boldsymbol{x}_m)$ 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 $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ 为条件的边缘概率分布 $p(\boldsymbol{t})$，我们需要对 $\boldsymbol{y}$ 积分。可以通过使用公式 2.81，我们看到 $\boldsymbol{t}$ 的边缘概率分布为

$$p(\boldsymbol{t}) = \int p(\boldsymbol{t}|\boldsymbol{y})p(\boldsymbol{y})d\boldsymbol{y} = \mathcal{N}(\boldsymbol{t}|0, \boldsymbol{C}) \tag{10.46}$$

其中协方差矩阵 $\boldsymbol{C}$ 的元素为

$$C(\boldsymbol{x}_n, \boldsymbol{x}_m) = k(\boldsymbol{x}_n, \boldsymbol{x}_m) + \beta^{-1}\delta_{nm} \tag{10.47}$$

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

$$k(\boldsymbol{x}_n, \boldsymbol{x}_m) = \theta_0 \exp\left\{-\frac{\theta_1}{2}\|\boldsymbol{x}_n - \boldsymbol{x}_m\|^2\right\} + \theta_2 + \theta_3 \boldsymbol{x}_n^T \boldsymbol{x}_m \tag{10.48}$$

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。假设 $\boldsymbol{t}_N = (t_1, \ldots, t_N)^T$，对应于输入值 $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$，组成观测训练集，并且我们的目标是对于新的输入向量 $\boldsymbol{x}_{N+1}$ 预测目标变量 t_{N+1}。这要求我们计算预测分布 $p(t_{N+1}|\boldsymbol{t}_N)$。

为了找到条件条布 $p(t_{N+1}|\boldsymbol{t})$，我们首先写下联合概率分布 $p(\boldsymbol{t}_{N+1})$。

$$p(\boldsymbol{t}_{N+1}) = \mathcal{N}(\boldsymbol{t}_{N+1}|0, \boldsymbol{C}_{N+1}) \tag{10.49}$$

其中 $\boldsymbol{C}_{N+1}$ 是一个 $(N+1)\times(N+1)$ 的协方差矩阵，我们将协方差矩阵分块如下

$$\boldsymbol{C}_{N+1} = \begin{pmatrix} \boldsymbol{C}_N & \boldsymbol{k} \\ \boldsymbol{k} & c \end{pmatrix} \tag{10.50}$$

其中 $\boldsymbol{C}_N$ 是一个 $N \times N$ 的协方差矩阵，向量 $\boldsymbol{k}$ 的元素为 $k(\boldsymbol{x}_n, \boldsymbol{x}_{N+1})$，其中 $n = 1, \ldots, N$，标量 $c = k(\boldsymbol{x}_{N+1}, \boldsymbol{x}_{N+1})$。使用公式 2.66 和公式 2.67，我们看到条件概率分布 $p(t_{N+1}|\boldsymbol{t})$ 是一个高斯分布，均值和协方差为

$$m(\boldsymbol{x}_{N+1}) = \boldsymbol{k}^T \boldsymbol{C}_N^{-1} \boldsymbol{t} \tag{10.51}$$

$$\sigma^2(\boldsymbol{x}_{N+1}) = c - \boldsymbol{k}^T \boldsymbol{C}_N^{-1} \boldsymbol{k} \tag{10.52}$$

这些是定义高斯过程回归的关键结果。注意，预测分布的均值可以写成 $\boldsymbol{x}_{N+1}$ 的函数，形式为

$$m(\boldsymbol{x}_{N+1}) = \sum_{n=1}^{N} a_n k(\boldsymbol{x}_n, \boldsymbol{x}_{N+1}) \tag{10.53}$$

其中 a_n 是 $\boldsymbol{C}_N^{-1}\boldsymbol{t}$ 的第 n 个元素。

使用高斯过程的核心计算涉及到对 $N \times N$ 的矩阵求逆。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。

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 $p(\boldsymbol{t}|\boldsymbol{\theta})$，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来最大化对数似然函数。

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

$$\ln p(\boldsymbol{t}|\boldsymbol{\theta}) = -\frac{1}{2}\ln|\boldsymbol{C}_N| - \frac{1}{2}\boldsymbol{t}^T\boldsymbol{C}_N^{-1}\boldsymbol{t} - \frac{N}{2}\ln(2\pi) \tag{10.54}$$

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。

$$\begin{aligned}\frac{\partial}{\partial\theta_i}\ln p(\boldsymbol{t}|\boldsymbol{\theta}) &= -\frac{1}{2}\frac{\partial[\ln|\boldsymbol{C}_N|]}{\partial\theta_i} - \frac{1}{2}\boldsymbol{t}^T\frac{\partial[\boldsymbol{C}_N^{-1}]}{\partial\theta_i}\boldsymbol{t} \\ &= -\frac{1}{2}Tr\left(\boldsymbol{C}_N^{-1}\frac{\partial\boldsymbol{C}_N}{\partial\theta_i}\right) + \frac{1}{2}\boldsymbol{t}^T\boldsymbol{C}_N^{-1}\frac{\partial\boldsymbol{C}_N}{\partial\theta_i}\boldsymbol{C}_N^{-1}\boldsymbol{t}\end{aligned} \tag{10.55}$$

由于 $\ln p(\boldsymbol{t}|\boldsymbol{\theta})$ 通常是一个非凸函数,因此它有多个极大值点。

引入一个 θ 上的先验分布然后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 $p(\boldsymbol{t}|\boldsymbol{\theta})$。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。

自动相关性确定

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到一个单独的参数,这种方法可以很有用地推广。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 relecance detemination) 或者 ARD 的一个例子。

考虑二维输入空间 $\boldsymbol{x} = (x_1, x_2)$,有一个下面形式的核函数

$$k(\boldsymbol{x}, \boldsymbol{x}') = \theta_0\exp\left\{-\frac{1}{2}\sum_{i=1}^{2}\eta_i(x_i - x_i')^2\right\} \tag{10.56}$$

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。

ARD 框架很容易整合到指数-二次核中,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。

$$k(\boldsymbol{x}_n, \boldsymbol{x}_m) = \theta_0\exp\left\{-\frac{\theta_1}{2}\sum_{i=1}^{D}\eta_i(x_{ni} - x_{mi})^2\right\} + \theta_2 + \theta_3\sum_{i=1}^{D}\boldsymbol{x}_{ni}\boldsymbol{x}_{mi} \tag{10.57}$$

其中 D 是输入空间的维度。

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,使基能够处理分类问题。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。如果我们定义函数 $a(\boldsymbol{x})$ 上的一个高斯过程，然后使用 logistic sigmoid 函数 $y = \sigma(a)$ 进行变换。那么我们就得到了函数 $y(\boldsymbol{x})$ 上的一个非高斯随机过程。一维输入空间的情况下，目标变量 t 上的概率分布是伯努利分布

$$p(t|a) = \sigma(a)^t(1 - \sigma(a))^{1-t} \tag{10.58}$$

训练集的输入记作 $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$，对应的观测目标变量为 $\boldsymbol{t} = (t_1, \ldots, t_N)^T$。我们还考虑一个单一的观测数据点 $\boldsymbol{x}_{N+1}$，目标值为 t_{N+1}。我们的目标是确定预测分布 $p(t_{N+1}|\boldsymbol{t})$，其中我们没有显示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 $\boldsymbol{a}_{N+1}$ 上的高斯过程先验，它的分量为 $a(\boldsymbol{x}_1), \ldots, a(\boldsymbol{x}_{N+1})$ 这反过来定义了 $\boldsymbol{t}_{N+1}$ 上的一个非高斯过程。通过以训练数据 $\boldsymbol{t}_N$ 为条件，我们得到了求解的预测分布。$\boldsymbol{a}_{N+1}$ 上的高斯过程先验的形式为

$$p(\boldsymbol{a}_{N+1}) = \mathcal{N}(\boldsymbol{a}_{N+1}|0, \boldsymbol{C}_{N+1}) \tag{10.59}$$

协方差矩阵不包含噪声项，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 v 控制的类似噪声的项，它确保了协方差矩阵是正定的。因此协方差矩阵 $\boldsymbol{C}_{N+1}$ 的元素为

$$C(\boldsymbol{x}_n, \boldsymbol{x}_m) = k(\boldsymbol{x}_n, \boldsymbol{x}_m) + v\sigma_{nm} \tag{10.60}$$

其中 $k(\boldsymbol{x}_n, \boldsymbol{x}_m)$ 是任意的半正定核函数，v 的值通常事先固定。我们会假定核函数由参数向量 $\boldsymbol{\theta}$ 控制，稍后会讨论如何从训练数据中学习到 $\boldsymbol{\theta}$。

求解的预测分布为

$$p(t_{N+1}|\boldsymbol{t}_N) = \int p(t_{N+1} = 1|a_{N+1})p(a_{N+1}|\boldsymbol{t}_N)da_{N+1} \tag{10.61}$$

其中 $p(t_{N+1} = 1|a_{N+1}) = \sigma(\boldsymbol{a}_{N+1})$ 这个积分无法求出解析解，可以采用近似的方法。

1. 变分推断
2. 期望传播
3. 拉普拉斯近似

拉普拉斯近似

为了计算预测分布 10.61，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

$$\begin{aligned} p(a_{N+1}|\boldsymbol{t}_N) &= \int p(a_{N+1}, \boldsymbol{a}_N|\boldsymbol{t}_N)d\boldsymbol{a}_N \\ &= \frac{1}{p(\boldsymbol{t}_N)}\int p(a_{N+1}, \boldsymbol{a}_N, \boldsymbol{t}_N)d\boldsymbol{a}_N \\ &= \frac{1}{p(\boldsymbol{t}_N)}\int p(a_{N+1}, \boldsymbol{a}_N)p(\boldsymbol{t}_N|a_{N+1}, \boldsymbol{a}_N)d\boldsymbol{a}_N \\ &= \int p(a_{N+1}|\boldsymbol{a}_N)\frac{1}{p(\boldsymbol{t}_N)}p(\boldsymbol{a}_N)p(\boldsymbol{t}_N|\boldsymbol{a}_N)d\boldsymbol{a}_N \\ &= \int p(a_{N+1}|\boldsymbol{a}_N)p(\boldsymbol{a}_N|\boldsymbol{t}_N)d\boldsymbol{a}_N \end{aligned} \tag{10.62}$$

其中，我们用到了 $p(\boldsymbol{t}_N|a_{N+1}, \boldsymbol{a}_N) = p(\boldsymbol{t}_N|\boldsymbol{a}_N)$。使用公式 10.51 和公式 13.50 给出的高斯过程回归的结果，我们可以得到条件概率分布 $p(a_{N+1}|\boldsymbol{a}_N)$，结果为

$$p(a_{N+1}|\boldsymbol{a}_N) = \mathcal{N}(a_{N+1}|\boldsymbol{k}^T\boldsymbol{C}_N^{-1}\boldsymbol{a}_N, c - \boldsymbol{k}^T\boldsymbol{C}_N^{-1}\boldsymbol{k}) \tag{10.63}$$

于是，通过找到后验概率分布 $p(\boldsymbol{a}_N|\boldsymbol{t}_N)$ 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中的积分。

先验概率 $p(\boldsymbol{a}_N)$ 由一个零均值高斯过各给出，协方差矩阵为 $\boldsymbol{C}_N$，数据项 (假设数据点之间具有独立性) 为

$$p(\boldsymbol{t}_N|\boldsymbol{a}_N) = \prod_{n=1}^{N}\sigma(a_n)^{t_n}(1-\sigma(a_n))^{1-t_n} = \prod_{n=1}^{N}e^{a_n t_n}\sigma(-a_n) \tag{10.64}$$

然后通过对 $p(\boldsymbol{a}_N|\boldsymbol{t}_N)$ 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

$$\begin{aligned} \Psi(\boldsymbol{a}_N) &= \ln p(\boldsymbol{a}_N, \boldsymbol{t}_N) \quad \text{忽略了常数项} \\ &= \ln p(\boldsymbol{a}_N) + \ln p(\boldsymbol{t}_N|\boldsymbol{a}_N) \\ &= \underbrace{-\frac{1}{2}\boldsymbol{a}_N^T\boldsymbol{C}_N^{-1}\boldsymbol{a}_N - \frac{N}{2}\ln(2\pi) - \frac{1}{2}\ln|\boldsymbol{C}_N|}_{\boldsymbol{a}_N \text{ 服从零均值高斯分布}} \\ &\quad \underbrace{+\boldsymbol{t}_N^T\boldsymbol{a}_N - \sum_{n=1}^{N}\ln(1+e^{a_n})}_{\text{二项分布}} \end{aligned} \tag{10.65}$$

首先我们需要找到后验概率分布的众数，这需要我们计算 $\Psi(\boldsymbol{a}_N)$ 的梯度。这个梯度为

$$\nabla\Psi(\boldsymbol{a}_N) = \boldsymbol{t}_N - \boldsymbol{\sigma}_N - \boldsymbol{C}_N^{-1}\boldsymbol{a}_N \tag{10.66}$$

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 $\boldsymbol{a}_N$ 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 $\Psi(\boldsymbol{a}_N)$ 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

$$\nabla\nabla\Psi(\boldsymbol{a}_N) = -\boldsymbol{W}_N - \boldsymbol{C}_N^{-1} \tag{10.67}$$

其中 $\boldsymbol{W}_N$ 是一个对角矩阵，元素为 $\sigma(a_n)(1-\sigma(a_n))$，并且使用了 logistic sigmoid 函数的导数的结果

$$\frac{d\sigma}{da} = \sigma(1-\sigma) \tag{10.68}$$

注意，这些对角矩阵元素位于区间 $(0, \frac{1}{4})$，因此 $\boldsymbol{W}_N$ 是一个正定矩阵。由于 $\boldsymbol{C}_N$ 被构造成正定的，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 $\boldsymbol{A} = -\nabla\nabla\Psi(\boldsymbol{a}_N)$ 是正定的，因此后验概率分布 $p(\boldsymbol{a}_N|\boldsymbol{t}_N)$ 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 $\boldsymbol{a}_N$ 的函数。

使用 Newton-Raphson 公式，$\boldsymbol{a}_N$ 的迭代更新方程为

$$\boldsymbol{a}_N^{\text{新}} = \boldsymbol{C}_N(\boldsymbol{I} + \boldsymbol{W}_N\boldsymbol{C}_N)^{-1}\{\boldsymbol{t}_N - \boldsymbol{\sigma}_N + \boldsymbol{W}_N\boldsymbol{a}_N\} \tag{10.69}$$

这个方程反复迭代，直到收敛于众数 (记作 $\boldsymbol{a}_N^*$)。在这个众数位置，梯度 $\nabla\Psi(\boldsymbol{a}_N)$ 为零，因此 $\boldsymbol{a}_N^*$ 满足

$$\boldsymbol{a}_N^* = \boldsymbol{C}_N(\boldsymbol{t}_N - \boldsymbol{\sigma}_N) \tag{10.70}$$

一旦我们找到了后验概率的众数 $\boldsymbol{a}_N^*$，我们就可以计算 Hessian 矩阵，结果为

$$\boldsymbol{H} = -\nabla\nabla\Psi(\boldsymbol{a}_N) = \boldsymbol{W}_N + \boldsymbol{C}_N^{-1} \tag{10.71}$$

其中 $\boldsymbol{W}_N$ 的元素使用 $\boldsymbol{a}_N^*$ 计算。这定义了我们对后验概率分布 $p(\boldsymbol{a}_N|\boldsymbol{t}_N)$ 的高斯近似，结果为

$$q(\boldsymbol{a}_N) = \mathcal{N}(\boldsymbol{a}_N|\boldsymbol{a}_N^*, \boldsymbol{H}^{-1}) \tag{10.72}$$

我们现在可以将这个结果与公式 10.63 结合，然后计算积分 10.62。因为这对应于线性高斯模型，我们可以使用一般的结果得到

$$\mathbb{E}[a_{N+1}|\boldsymbol{t}_N] = \boldsymbol{k}^T(\boldsymbol{t}_N - \boldsymbol{\sigma}_N) \tag{10.73}$$

$$var[a_{N+1}|\boldsymbol{t}_N] = c - \boldsymbol{k}^T(\boldsymbol{W}_N^{-1} + \boldsymbol{C}_N)^{-1}\boldsymbol{k} \tag{10.74}$$

现在我们有一个 $p(a_{N+1}|\boldsymbol{t}_N)$ 的高斯分布，我们可以使用 7.133 的结果近似积分 10.62。

我们还需要确定协方差函数的参数 $\boldsymbol{\theta}$。一种方法是最大化似然函数 $p(\boldsymbol{t}_N|\boldsymbol{\theta})$，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。

与神经网络的联系

在贝叶斯神经网络中，参数向量 $\boldsymbol{w}$ 上的先验分布以及网络函数 $f(\boldsymbol{x}, \boldsymbol{w})$ 产生了函数 $y(\boldsymbol{x})$ 上的先验概率分布，其中 $\boldsymbol{y}$ 是网络输出向量。在极限 $M \Rightarrow \infty$ 的情况下，对于 $\boldsymbol{w}$ 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。

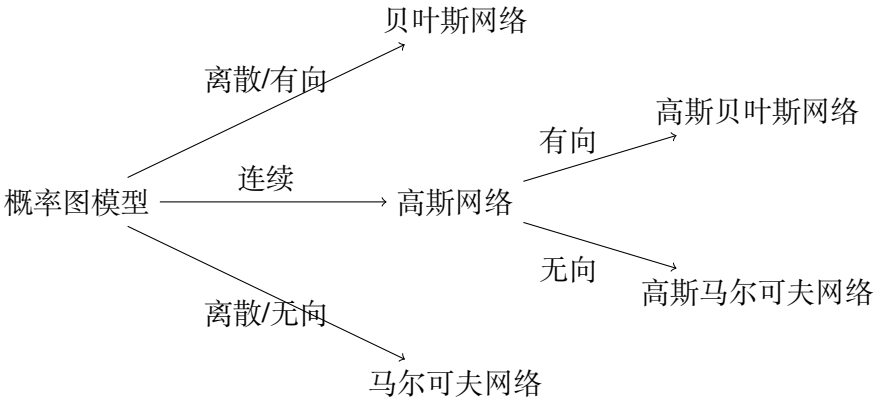
第 11 章 概率图模型

概率在现代模式识别中起着重要的作用。我们已经看到概率论可以使用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称为概率图模型 (probabilistic graphical models)。这些模型提供了几个有用的性质。

1. 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。
2. 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。
3. 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) 和它们之间的链接 (links) 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 (Beyesian 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。另一个大类图模型是马尔可夫随机场 (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)。



11.1 贝叶斯网络

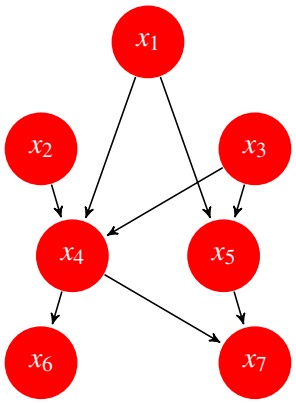
为了理解有向图对于描述概率分布的作用，考虑 K 个变量的联合概率分布 $p(x_1, \ldots, x_K)$。通过重复使用概率的乘积规则，联合概率分布可以写成条件概率的乘积，每一项对应一个

变量,形式如下

$$p(x_1,\ldots,x_K) = p(x_K|x_1,\ldots,x_{K-1})\ldots p(x_2|x_1)p(x_1) \tag{11.1}$$

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 11.1 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。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)。

现在,根据下面这幅图,写出对应的概率表达式。



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

$$p(x_1)p(x_2)p(x_3)p(x_4|x_1,x_2,x_3)p(x_5|x_1,x_3)p(x_6|x_4)p(x_7|x_4,x_5) \tag{11.2}$$

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

$$p(\boldsymbol{x}) = \prod_{k=1}^{K} p(x_k|\mathrm{pa}_k) \tag{11.3}$$

其中,pa_k 表示 x_k 的父结点的集合,$\boldsymbol{x} = \{x_1,\ldots,x_K\}$。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,或者关联向量值的变量。

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)。

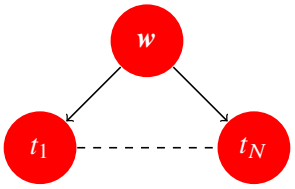
例子:多项式回归

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 $\boldsymbol{w}$ 和观测数据 $\boldsymbol{t} = (t_1,\ldots,t_N)^T$。此外,这个模型包含输入数据 $\boldsymbol{X} = (x_1,\ldots,x_N)^T$、噪声方差 σ^2 以及表示 $\boldsymbol{w}$ 的高斯先验分布的精度的超参数 α。所有

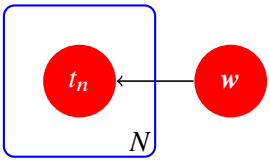
这些都是模型的参数而不是随机变量。现阶段我们只关注随机变量。

$$p(\boldsymbol{t}, \boldsymbol{w}) = p(\boldsymbol{w}) \prod_{n=1}^{N} p(t_n|\boldsymbol{w}) \tag{11.4}$$

图模型表示的联合概率分布如图所示



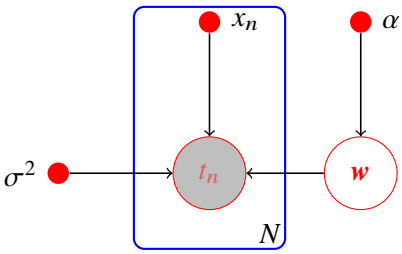
当我们开始处理更加复杂的模型时，我们会看到，像图中那样显式地写出 $t_1, \ldots, t_N$ 的结点是很不方便的。于是，我们引入一种图结构，使得多个结点可以更简洁地表示出来。这种图结构中，我们画出一个单一表示的结点 t_n，然后用一个被称为板 (plate) 的方框圈起来，标记为 N，表示有 N 个同类型的点。用这种表示重新表示上图，我们得到了下图



我们有时会发现，显式地定出模型的参数和随机变量是很有帮助的。此时，公式 11.4 就变成了

$$p(\boldsymbol{t}, \boldsymbol{w}|\boldsymbol{X}, \alpha, \sigma^2) = p(\boldsymbol{w}|\alpha) \prod_{n=1}^{N} p(t_n|\boldsymbol{w}, x_n, \sigma^2) \tag{11.5}$$

对应地，我们可以在图表示中显式地写出 $\boldsymbol{X}$ 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。



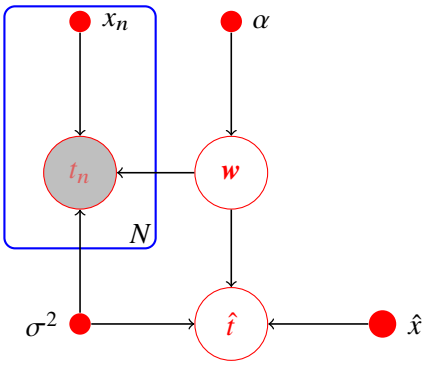
当我们将图模型应用于机器学习或者模型识别的问题中时，我们通常将某些随机变量设置为具体的值，例如将变量 $\{t_n\}$ 根据多项式曲线拟合中的训练集进行设置。在图模型中，我们通过给对应的结点加上阴影的方式来表示这种观测变量 (observed variables)。$\boldsymbol{w}$ 不是观测变量，因此 $\boldsymbol{w}$ 是潜在变量 (latent variable) 的一个例子。潜在变量也被称为隐含变量 (hidden variable)。这样的变量在许多概率模型中有着重要的作用，将在后面的章节详细讨论。

观测到了 $\{t_n\}$ 的值，如果必要的话，我们可以计算系数 $\boldsymbol{w}$ 的后验概率。

$$p(\boldsymbol{w}|\boldsymbol{t}) \propto p(\boldsymbol{w}) \prod_{n=1}^{N} p(t_n|\boldsymbol{w}) \tag{11.6}$$

其中，省略了确定性系数，使得记号简洁。

通常，我们对于 $\boldsymbol{w}$ 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下图所示



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

$$p(\hat{t}, \boldsymbol{t}, \boldsymbol{w}|\hat{x}, \boldsymbol{X}, \alpha, \sigma^2) = \left[\prod_{n=1}^{N} p(t_n|x_n, \boldsymbol{w}, \sigma^2)\right] p(\boldsymbol{w}|\alpha) p(\hat{t}|\hat{x}, \boldsymbol{w}, \sigma^2) \tag{11.7}$$

然后，根据概率的加和规则，对模型参数 $\boldsymbol{w}$ 积分，即可得到 $\hat{t}$ 的预测分布

$$p(\hat{t}|\hat{x}, \boldsymbol{X}, \boldsymbol{t}, \alpha, \sigma^2) \propto \int p(\hat{t}, \boldsymbol{t}, \boldsymbol{w}|\hat{x}, \boldsymbol{X}, \alpha, \sigma^2) d\boldsymbol{w} \tag{11.8}$$

其中我们隐式地将 $\boldsymbol{t}$ 中的随机变量设置为数据集中观测到的具体值。

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。后面用一章节讨论取样方法，这里简要介绍一种方法——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 $p(x_1, \ldots, x_K)$，它根据公式 11.3 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 $\hat{x}_1, \ldots, \hat{x}_K$。

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 $p(x_n|\mathrm{pa}_n)$ 进行取样，其中父结点的变量被设置为它们的取样值。按照具体的概率分布的取样方法将会在后面的章节中详细讨论。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。

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。潜在的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的条件分布 (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。图模型描述了生成观测数据的一种因果关系 (causal) 过程。因此，这种模型通常被称为生成式模型 (generatice model)。

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) 与观测数据的根据分布相同。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。

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。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。

现在假设我们有两个离散变量 $\boldsymbol{x}_1$ 和 $\boldsymbol{x}_2$，每个都有 K 个状态，我们想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 $\boldsymbol{x}_1$ 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

$$p(\boldsymbol{x}_1, \boldsymbol{x}_2|\boldsymbol{\mu}) = \prod_{k=1}^{K}\prod_{l=1}^{K} \mu_{kl}^{x_{1k}x_{2l}} \tag{11.9}$$

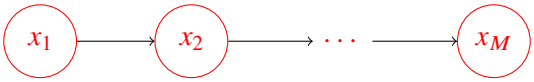
由于参数满足限制 $\sum_k \sum_l u_{kl} = 1$，因此这个分布由 $K^2 - 1$ 个参数控制。很容易看到，对于 M 个变量的任意一个联合概率分布，需要确定的参数的数量为 $K^M - 1$，因此随着变量 M 的数量指数增长。

使用概率的乘积规则，我们可以将联合概率分布 $p(\boldsymbol{x}_1, \boldsymbol{x}_2)$ 分解为 $p(\boldsymbol{x}_2|\boldsymbol{x}_1)p(\boldsymbol{x}_1)$，它对应于一个具有两个结点的图，链接从结点 $\boldsymbol{x}_1$ 指向结点 $\boldsymbol{x}_2$。边缘概率分布 $p(\boldsymbol{x}_1)$ 与之前一样，由 K-1 个参数控制。类似地，条件概率分布 $p(\boldsymbol{x}_2|\boldsymbol{x}_1)$ 需要指定 K-1 个参数，确定 $\boldsymbol{x}_1$ 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 $\boldsymbol{x}_1$ 和 $\boldsymbol{x}_2$ 是独立的。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

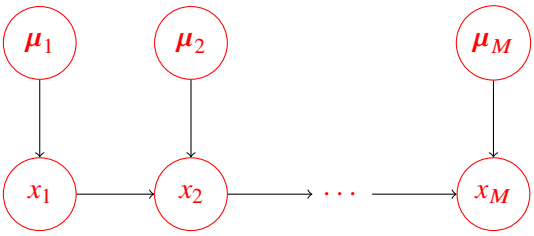
通过删除结点之间链接的方式，减小了参数的数量，代价是类别的概率分布受到了限制。

作为一个说明，考虑下图所示的结点链。

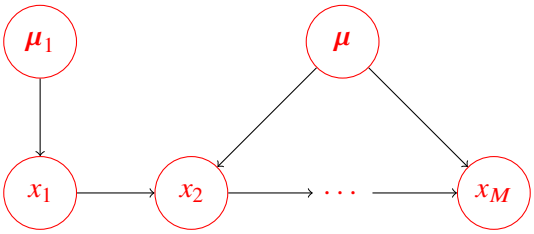


边缘概率分布 $p(\boldsymbol{x}_1)$ 需要 K-1 个参数，而对于 M-1 个条件概率分布 $p(\boldsymbol{x}_i|\boldsymbol{x}_{i-1})$ 需要 K(K-1) 个参数。从而，参数的总数为 $K-1+(M-1)K(K-1)$，这是 K 的二次函数，并且随着链的长度 M 线性增长 (而不是指数增长).

另一种减小模型中独立参数数量的方法是参数共享 (sharing)。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。如图所示



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。如果我们将控制条件概率分布进行参数共享，那么对应的模型如图所示



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。

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。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

$$p(x_i|\mathrm{pa}_i)=\mathcal{N}\left(x_i\middle|\sum_{j\in\mathrm{pa}_i}w_{ij}x_j+b_i,v_i\right) \tag{11.10}$$

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

$$\begin{aligned}\ln p(\boldsymbol{x}) &= \sum_{i=1}^{D} \ln p(x_i|\mathrm{pa}_i) \\ &= -\sum_{i=1}^{D} \frac{1}{2v_i}\left(x_i - \sum_{j\in \mathrm{pa}_i} w_{ij}x_j - b_i\right)^2 + \text{常数}\end{aligned} \tag{11.11}$$

其中 $\boldsymbol{x} = (x_1, \ldots, x_D)^T$，“常数”表示与 $\boldsymbol{x}$ 无关的项。我们看到这是 $\boldsymbol{x}$ 的元素的二次函数，因此联合概率分布 $p(\boldsymbol{x})$ 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。因此

$$x_i = \sum_{j\in \mathrm{pa}_i} w_{ij}x_j + b_i + \sqrt{v_i}\epsilon_i \tag{11.12}$$

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 $\mathbb{E}[\epsilon_i\epsilon_j] = \boldsymbol{I}_{ij}$，其中 $\boldsymbol{I}_{ij}$ 是单位矩阵的第 i, j 个元素。对公式 11.12 取期望，我们有

$$\mathbb{E}[x_i] = \sum_{j\in \mathrm{pa}_i} w_{ij}\mathbb{E}[x_j] + b_i \tag{11.13}$$

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 $\mathbb{E}[\boldsymbol{x}] = (\mathbb{E}[x_1], \ldots, \mathbb{E}[x_D])^T$ 的各个元素。这样，我们再一次假设所有结点的序号都大于它父结点的序号。类似地，我们可以使用公式 11.12 和 11.13 以递归的方式得到 $p(\boldsymbol{x})$ 的协方差矩阵的第 i,j 个元素，即

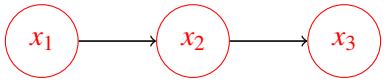
$$\begin{aligned}cov[x_i, x_j] &= \mathbb{E}[(x_i - \mathbb{E}[x_i])(x_j - \mathbb{E}[x_j])] \\ &= \mathbb{E}\left[(x_i - \mathbb{E}[x_i])\left\{\sum_{j\in \mathrm{pa}_i} w_{jk}x_k + b_i + \sqrt{v_j}\epsilon_j - \sum_{j\in \mathrm{pa}_i} w_{jk}\mathbb{E}[x_k] - b_i\right\}\right] \\ &= \mathbb{E}\left[(x_i - \mathbb{E}[x_i])\left\{\sum_{j\in \mathrm{pa}_j} w_{jk}(x_k - \mathbb{E}[x_k]) + \sqrt{v_j}\epsilon_j\right\}\right] \\ &= \mathbb{E}\left[\sum_{j\in \mathrm{pa}_j} w_{jk}(x_i - \mathbb{E}[x_i])(x_k - \mathbb{E}[x_k]) + \sqrt{v_j}\epsilon_j(x_i - \mathbb{E}[x_j])\right] \\ &= \sum_{j\in \mathrm{pa}_j} w_{jk}\mathbb{E}[(x_i - \mathbb{E}[x_i])(x_k - \mathbb{E}[x_k])] + \mathbb{E}[\sqrt{v_j}\epsilon_j(x_i - \mathbb{E}[x_i])] \\ &= \sum_{k\in \mathrm{pa}_i} w_{jk}cov[x_i, x_k] + \underbrace{\boldsymbol{I}_{ij}v_j}_{\text{看只保留了哪些项得出!}}\end{aligned} \tag{11.14}$$

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，我

们看到 $p(\boldsymbol{x})$ 的均值为 $(b_1, \ldots, b_D)^T$，协方差矩阵是一个对角矩阵，形式为 $\text{diag}(v_1, \ldots, v_D)$。

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 (对角线上没有元素)。

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。考虑下图



它在变量 x_1 和 x_3 之间不存在链接。使用递归关系，我们看到联合高斯分布的均值和协方差为

$$\boldsymbol{\mu} = (b_1, b_2 + w_{21}b_1, b_3 + w_{32}b_2 + w_{32}w_{21}b_1)^T \tag{11.15}$$

$$\boldsymbol{\Sigma} = \begin{pmatrix} v_1 & w_{21}v_1 & w_{32}w_{21}v_1 \\ w_{21}v_1 & v_2 + w_{21}^2 v_1 & w_{32}(v_2 + w_{21}^2 v_1) \\ w_{32}w_{21}v_1 & w_{32}(v_2 + w_{21}^2 v_1) & v_3 + w_{32}^2(v_2 + w_{21}^2 v_1) \end{pmatrix} \tag{11.16}$$

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

$$p(\boldsymbol{x}_i|\text{pa}+i) = \mathcal{N}\left(\boldsymbol{x}_i \middle| \sum_{j \in \text{pa}_i} \boldsymbol{W}_{ij}\boldsymbol{x}_j + \boldsymbol{b}_i, \boldsymbol{\Sigma}_i\right) \tag{11.17}$$

现在 $\boldsymbol{W}_{ij}$ 是一个矩阵。如果 $\boldsymbol{x}_i$ 和 $\boldsymbol{x}_j$ 的维度不同，那么 $\boldsymbol{W}_{ij}$ 不是方阵。很容易证明所有变量上的联合概率分布是高斯分布。

11.2 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)。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。完成这一件事的一般框架被称为“d-划分”(d-separation)，其中“d”表示有向 (directed)。

图的三个例子

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

$$\begin{aligned} p(a, b|c) &= p(a|b, c)p(b|c) \\ &= p(a|c)p(b|c) \end{aligned} \tag{11.18}$$

因此，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)。我们有时会使用条件独立的一种简洁记号，即

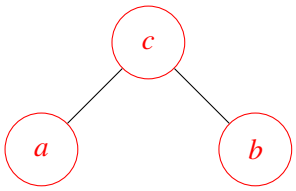
$$a \perp\!\!\!\perp b \mid c \tag{11.19}$$

表示给定 c 条件下 a 与 b 条件独立，等价于

$$p(a|b,c) = p(a|c) \tag{11.20}$$

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。

1. 三个例子中的第一个。如图所示。



使用公式给出的一般结果，对应于这个图的联合概率分布很容易写出来，即

$$p(a,b,c) = p(a|c)p(b|c)p(c) \tag{11.21}$$

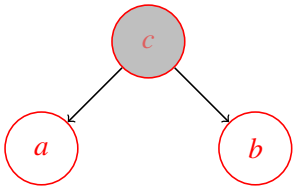
如果没有变量是观测变量，那么我们可以通过对公式 11.21 两边进行积分或求和的方式，考察 a 和 b 是否相互独立，即

$$p(a,b) = \sum_c p(a|c)p(b|c)p(c) \tag{11.22}$$

一般地，这不能分解为乘积 $p(a)p(b)$，因此

$$a \not\perp\!\!\!\perp b \mid \emptyset \tag{11.23}$$

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。现在假设我们以变量 c 为条件，如图



根据公式，给定 c 的条件下，a 和 b 的条件概率分布，形式为

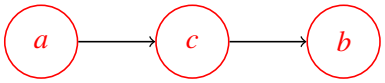
$$\begin{aligned} p(a,b|c) &= \frac{p(a,b,c)}{p(c)} \\ &= p(a|c)p(b|c) \end{aligned} \tag{11.24}$$

因此,我们可以得到条件独立性质

$$a \perp\!\!\!\perp b \mid c \tag{11.25}$$

通过考虑从结点 a 经过结点 c 到结点 b。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,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。

2. 三个例子中的第二个。如图所示。



对应于这幅图的联合概率分布可以通过一般形式的公式得到,形式为

$$p(a,b,c) = p(a)p(c|a)p(b|c) \tag{11.26}$$

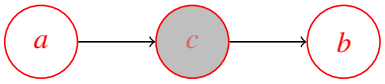
首先,假设所有的变量都不是观测变量。与之前一样,我们可以考察 a 和 b 是否是相互独立的,方法是对 c 积分或求和,结果为

$$p(a,b) = p(a)\sum_{c} p(c|a)p(b|c) = p(a)p(b|a) \tag{11.27}$$

这通常不能够分解为 $p(a)p(b)$,因此

$$a \not\perp\!\!\!\perp b \mid \emptyset \tag{11.28}$$

现在假设我们以 c 为条件,如图所示



使用贝叶斯定理,我们有

$$\begin{aligned} p(a,b|c) &= \frac{p(a,b,c)}{p(c)} \\ &= \frac{p(a)p(c|a)p(b|c)}{p(c)} \\ &= p(a|c)p(b|c) \end{aligned} \tag{11.29}$$

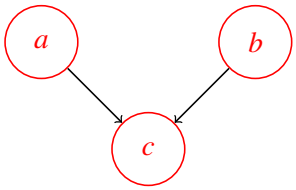
从而我们又一次得到了条件独立性质

$$a \perp\!\!\!\perp b \mid c \tag{11.30}$$

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，那么这样观测 “阻隔” 了从 a 到 b 的路径，因此我们得到了条件独立性质 $a \perp\!\!\!\perp b \mid c$。

3. 三个例子中的第三个。如图所示。



联合概率分布可以使用我们的一般结果得到

$$p(a,b,c) = p(a)p(b)p(c|a,b) \tag{11.31}$$

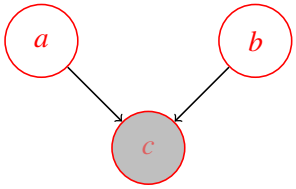
首先考虑没有变量是观测变量时的情形。对公式两侧关于 c 积分或求和，我们有

$$p(a,b) = p(a)p(b) \tag{11.32}$$

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

$$a \perp\!\!\!\perp b \mid \emptyset \tag{11.33}$$

现在假设我们以 c 为条件，如图所示



a 和 b 的条件概率分布为

$$\begin{aligned} p(a,b|c) &= \frac{p(a,b,c)}{p(c)} \\ &= \frac{p(a|c)p(b|c)p(c|a,b)}{p(c)} \end{aligned} \tag{11.34}$$

因此

$$a \not\perp\!\!\!\perp b \mid c \tag{11.35}$$

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。

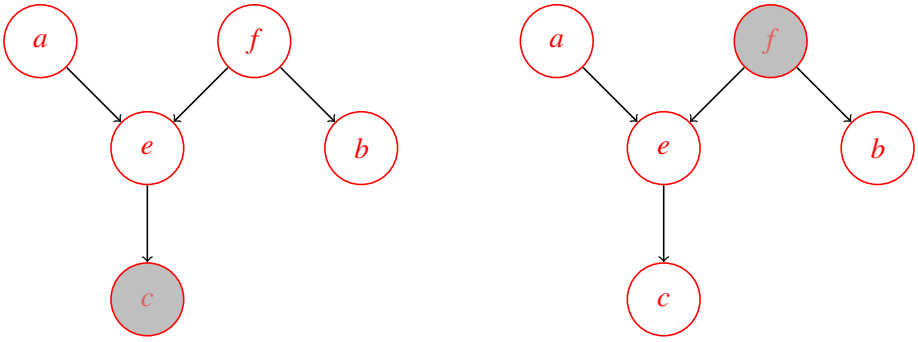
d-划分

现在给出有向图 d-划分性质的一个一般的叙述。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 $A \perp\!\!\!\perp B \mid C$。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

1. 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。
2. 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 $A \perp\!\!\!\perp B \mid C$。下图说明了 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。这个路径也没有被结点 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 在集合中。因此条件独立关系 $a \perp\!\!\!\perp b \mid c$ 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 $a \perp\!\!\!\perp b \mid f$。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，用小实心圆表示的参数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。

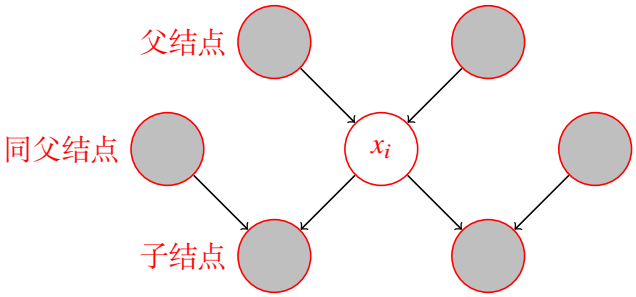
我们已经看到一个特定的有向图表示将联合概率分布分解为条件概率分布乘积形式的一个具体的分解方式。图也表示一组条件独立的性质，这些性质通常 d-划分的方式得到，并且 d-划分定理实际上是一个等价于这两个性质的表示。为了让这一点更明显，将有向图想象成滤波器。假设我们考虑 $\boldsymbol{x}$ 上的一个特定的联合概率分布 $p(\boldsymbol{x})$，其中 $\boldsymbol{x}$ 对应于图中的 (未观测) 结点。一个概率分布能够通过滤波器当且仅当它能够用与图对应的公式给出的分解方式进行分解。如果我们将变量 $\boldsymbol{x}$ 的集合上的所有可能的概率分布 $p(\boldsymbol{x})$ 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)。

考虑一个联合概率分布 $p(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_D)$，它由一个具有 D 个结点的有向图表示。考虑变量 $\boldsymbol{x}_i$ 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 $\boldsymbol{x}_{j \neq i}$。使用分解性质，我

们可以将条件概率分布表示为下面的形式

$$p(\boldsymbol{x}_i|\boldsymbol{x}_{\{j\neq i\}}) = \frac{p(\boldsymbol{x}_1,\ldots,\boldsymbol{x}_D)}{\int p(\boldsymbol{x}_1,\ldots,\boldsymbol{x}_D)d\boldsymbol{x}_i} = \frac{\prod_k p(\boldsymbol{x}_k|\mathrm{pa}_k)}{\int \prod_k p(\boldsymbol{x}_k|\mathrm{pa}_k)d\boldsymbol{x}_i} \tag{11.36}$$

观察到任何与 $\boldsymbol{x}_i$ 没有函数依赖关系的因子都可以提到 $\boldsymbol{x}_i$ 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 $\boldsymbol{x}_i$ 本身的条件概率分布 $p(\boldsymbol{x}_i|\mathrm{pa}_i)$，以及满足下面性质的结点 $\boldsymbol{x}_k$ 的条件概率分布：结点 $\boldsymbol{x}_i$ 在 $p(\boldsymbol{x}_k|\mathrm{pa}_k)$ 的条件集合中，即 $\boldsymbol{x}_i$ 是 $\boldsymbol{x}_k$ 的父结点。由父结点、子结点、同父结点组成的结点集合被称为马尔可夫毯 (Markov blanket) 或者马尔可夫边界 (Markov boundary)。如图所示



我们可以将结点 $\boldsymbol{x}_i$ 的马尔可夫毯想象成将 $\boldsymbol{x}_i$ 与图的剩余部分隔离开的最小结点集合。

11.3 马尔科夫随机场

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。一个马尔可夫随机场 (Markov random field)，也被称为马尔可夫网络 (Markov network) 或者无向图模型 (undirected graphical model)，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。

条件独立性

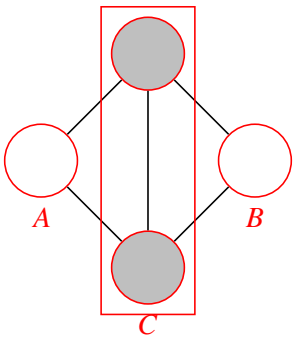
在有向图的情形下，我们看到可以通过使用被称为 d-划分的图检测方法判断一个特定的条件独立性质是否成立。这涉及到判断链接两个结点集合的路径是否被“阻隔”。然而，由于头到头结点的存在，阻隔的定义多少有些微妙。对应于无向图模型，通过移除图中链接的方向性，父结点和子结点的非对称性也被移除了，因此头到头的微妙性也就不再存在了。

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

$$A \perp\!\!\!\perp B \mid C \tag{11.37}$$

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

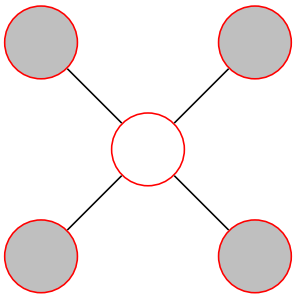
所有这样的路径都被“阻隔”，因此条件独立质成立。然而，如果存在至少一条未被阻隔的路径，那么条件独立的性质未必成立，或者更精确地说，存在至少某些对应于图的概率分布不满足条件独立性质。如下图中的例子



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 中的结点及与这些结点相连的链接全部删除。然后，我们考虑是否存在一条从 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。

无向图的马尔可夫毯的形式相当简单，因为结点只条件依赖于相信结点，而条件独立于任何其他的结点，如图所示



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 $p(\boldsymbol{x})$ 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

$$p(x_i, x_j|\boldsymbol{x}_{\setminus\{i,j\}}) = p(x_i|\boldsymbol{x}_{\setminus\{i,j\}})p(x_j|\boldsymbol{x}_{\setminus\{i,j\}}) \tag{11.38}$$

其中 $\boldsymbol{x}_{\setminus\{i,j\}}$ 表示所有变量 $\boldsymbol{x}$ 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。

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大团块的子集。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的一个函数，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 $\boldsymbol{x}_C$。这样，联合概率分布可以写成图的最大团块的势函数 (potential function)$\psi_C(\boldsymbol{x}_C)$ 的乘积的形式

$$p(\boldsymbol{x}) = \frac{1}{Z}\prod_C \psi_C(\boldsymbol{x}_C) \tag{11.39}$$

这里，Z 有时被称为划分函数 (partition function)，是一个归一化常数，等于

$$Z = \sum_{\boldsymbol{x}}\prod_C \psi_C(\boldsymbol{x}_C) \tag{11.40}$$

它确保了公式给出的概率分布被正确地归一化。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。势函数 $\psi_C(\boldsymbol{x}_C)$ 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。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缺点。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。

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。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 $\mathcal{UI}$ 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 $\mathcal{UF}$ 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 11.39 给出。Hammersley-Clifford 定理表明，集合 $\mathcal{UI}$ 和 $\mathcal{UF}$ 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

$$\psi_C(\boldsymbol{x}_C) = \exp\{-E(\boldsymbol{x}_C)\} \tag{11.41}$$

其中 $E(\boldsymbol{x}_C)$ 被称为能量函数 (energy function)，指数表示被称为玻尔兹曼分布 (Boltzmann

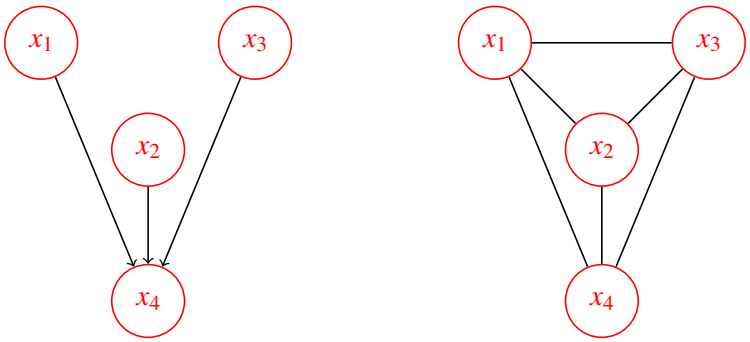
distribution)。联合概率分布被定义为势函数的乘积，因此总的能量可以通过将每个最大团块的能量相加的方法得到。

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：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。

与有向图的关系

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，划分函数都分 Z=1。

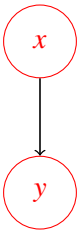
如图所示 (a) 一个简单的有向图的例。(b) 对应的道德图



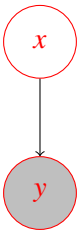
11.4 图模型中的推断

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。具体地说，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。后面的章节中，我们会考虑许多近似推断的算法。

首先，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 $p(x, y)$ 分解为因子的乘积的形式 $p(x, y) = p(x)p(y|x)$。这可以用图表示。



现在假设我们观测到了 y 的值，如图所示



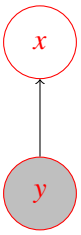
我们可以将边缘概率分布 $p(x)$ 看成潜在变量 x 上的先验概率分布，我们的目标是推断 x 上对应的后验概率分布。使用概率的加和规则和乘积规则，我们可以计算

$$p(y) = \sum_{x'} p(y|x')p(x') \tag{11.42}$$

这个式子然后被用于贝叶斯定理中，计算

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)} \tag{11.43}$$

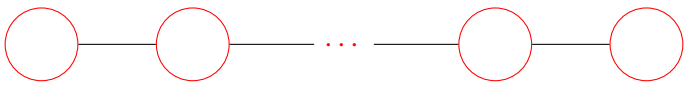
因此现在联合概率分布可以通过 $p(y)$ 和 $p(x|y)$。从图的角度看，联合概率分布 $p(x, y)$ 现在可以表示为下图，其中箭头的方向翻转了。



这是图模型中推断问题的最简单的例子。

链推断

现在考虑一个更加复杂的问题，涉及到图示中的结点链。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。



这个图的联合概率分布形式为

$$p(\boldsymbol{x}) = \frac{1}{Z}\psi_{1,2}(x_1, x_2)\psi_{2,3}(x_2, x_3)\ldots\psi_{N-1,N}(x_{N-1}, x_N) \tag{11.44}$$

我们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。

$$p(x_n) = \sum_{x_1} \cdots \sum_{x_{n-1}} \sum_{x_{n+1}} \cdots \sum_{x_N} p(\boldsymbol{x}) \tag{11.45}$$

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 $\boldsymbol{x}$ 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 $\boldsymbol{x}$ 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 11.44 代入到公式 11.45 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

$$\sum_{x_N} \psi_{N-1,N}(x_{N-1}, x_N) \tag{11.46}$$

得到一个关于 x_{N-1} 的函数。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

$$p(x_n) = \frac{1}{Z} \underbrace{\left[\sum_{x_{n-1}} \psi_{n-1,n}(x_{n-1}, x_n) \cdots \left[\sum_{x_2} \psi_{2,3}(x_2, x_3) \left[\sum_{x_1} \psi_{1,2}(x_1, x_2)\right]\right] \cdots\right]}_{\mu_\alpha(x_n)} \underbrace{\left[\sum_{x_{n+1}} \psi_{n,n+1}(x_n, x_{n+1}) \cdots \left[\sum_{x_N} \psi_{N-1,N}(x_{N-1}, x_N)\right] \cdots\right]}_{\mu_\beta(x_n)} \tag{11.47}$$

这个重排列的方式，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，我们利用的关键概率是乘法对加法的分配率，即

$$ab + ac = a(b + c) \tag{11.48}$$

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。

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11.47，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

$$p(x_n) = \frac{1}{Z}\mu_\alpha(x_n)\mu_\beta(x_n) \tag{11.49}$$

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。

信息 $\mu_\alpha(x_n)$ 可以递归地计算，因为

$$\begin{aligned}\mu_\alpha(x_n) &= \sum_{x_{n-1}} \psi_{n-1,n}(x_{n-1}, x_n)\left[\sum_{x_{n-2}} \cdots\right] \\ &= \sum_{x_{n-1}} \psi_{n-1,n}(x_{n-1}, x_n)\mu_\alpha(x_{n-1})\end{aligned} \tag{11.50}$$

因此我们首先计算

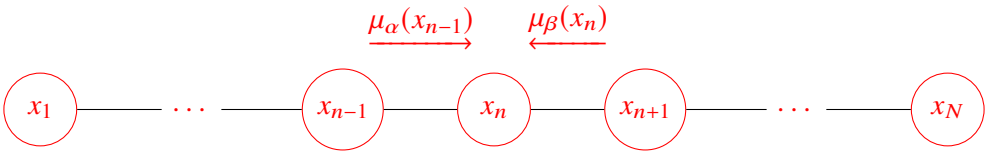
$$\mu_\alpha(x_2) = \sum_{x_1} \psi_{1,2}(x_1, x_2) \tag{11.51}$$

然后重复应用公式 11.50 直到我们到达需要求解的结点。

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

$$\begin{aligned}\mu_\beta(x_n) &= \sum_{x_{n+1}} \psi_{n,n+1}(x_n, x_{n+1})\left[\sum_{x_{n+2}} \cdots\right] \\ &= \sum_{x_{n+1}} \psi_{n,n+1}(x_n, x_{n+1})\mu_\beta(x_{n+1})\end{aligned} \tag{11.52}$$

这种递归的信息传递如图所示。



上图被称为马尔可夫链。对应的信息传递方程是马尔可夫过程的 Chapman-Kolmogorov 方程的一个例子。

现在我们想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。需要求解的边缘概率分布可以写成下面的形式

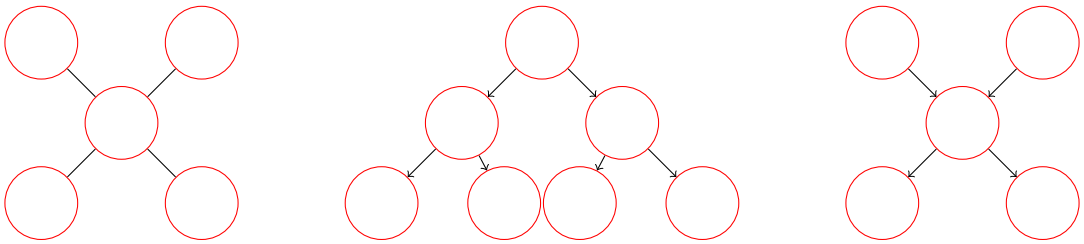
$$p(x_{n-1}, x_n) = \frac{1}{Z}\mu_\alpha(x_{n-1})\psi_{n-1,n}(x_{n-1}, x_n)\mu_\beta(x_n) \tag{11.53}$$

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。

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。

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



因子图

在推导加和-乘积算法之前，引入一个新的图结构，被称为因子图 (factor graph)，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图的乘积。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。

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

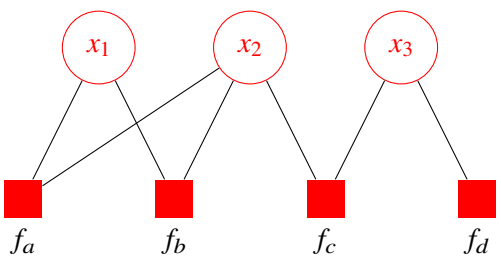
$$p(\boldsymbol{x}) = \prod_s f_s(\boldsymbol{x}_s) \tag{11.54}$$

其中 $\boldsymbol{x}_s$ 表示变量的一个子集。每个因子 f_s 是对应的变量集合 $\boldsymbol{x}_s$ 的函数。

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 $f_s(\boldsymbol{x}_s)$。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

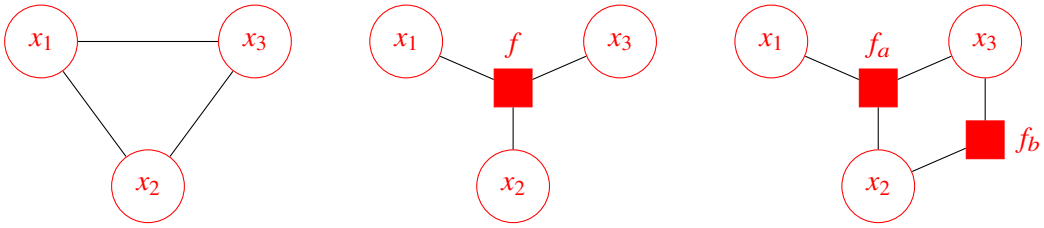
$$p(\boldsymbol{x}) = f_a(x_1, x_2) f_b(x1, x_2) f_c(x_2, x_3) f_d(x_3) \tag{11.55}$$

这可表示为下图所示的因子图。



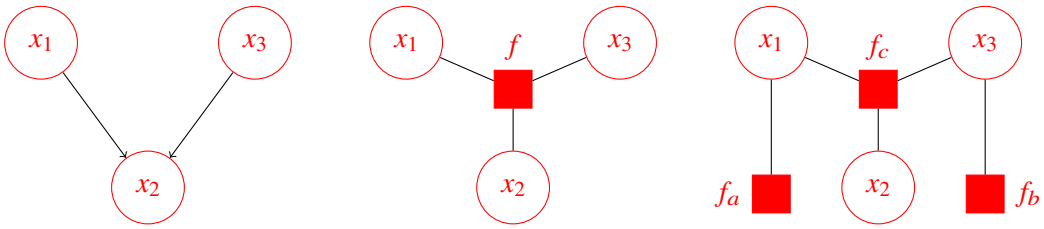
注意有两个因子 $f_a(x_1, x_2)$ 和 $f_b(x_1, x_2)$ 定义在同一个变量集合上。在一个无向图中，两个这样的因子的乘积被简单地合并到同一个团块势函数中。类似地，$f_c(x_2, x_3)$ 和 $f_d(x_3)$ 可以结合到 x_2 和 x_3 上的一个单一势函数中。然而，因子图显示地写出这些因子，因此能够表达出关于分解本身的更加细节的信息。

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 $\boldsymbol{x}_s$。因子 $f_s(\boldsymbol{x}_s)$ 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。下图说明了这些概念。



(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 $f(x_1, x_2, x_3) = \psi(x_1, x_2, x_3)$。(c) 一个不同的因子图，表示相同的概率分布，它的因子满足 $f_a(x_1, x_2, x_3)f_b(x_2, x_3) = \psi(x_1, x_2, x_3)$。

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所示。



(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 $f_a(x_1) = p(x_1)$, $f_b(x_2) = p(x_2)$, $f_c(x_1, x_2, x_3) = p(x_3|x_1, x_2)$。

我们已经看到了树结构图对于进行高维推断的重要性。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”的步骤的存在，转化为无向图会引入环，而转化后的因子图仍然是树。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转换到因子图时被移除，只需定义合适的因子函数即可。

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)。

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：

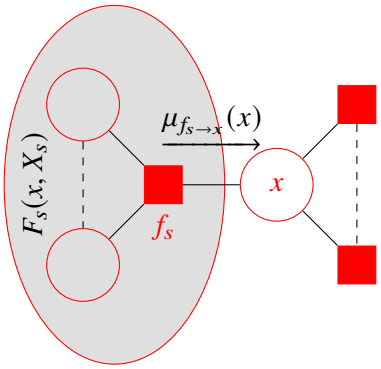
1. 得到一个高效的精确推断算法来寻找边缘概率；
2. 在需要求解多个边缘概率的情形，计算可以高效地共享。

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

$$p(x) = \sum_{\boldsymbol{x}\setminus x} p(\boldsymbol{x}) \tag{11.56}$$

其中 $\boldsymbol{x}\setminus x$ 表示变量 $\boldsymbol{x}$ 的集合去掉变量 x。算法的思想是使用因子图的表达式 11.54 替换 $p(\boldsymbol{x})$，然后交换加和与乘积的顺序，得到一个高效的算法。

考虑下图



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

$$p(\boldsymbol{x}) = \prod_{s\in \mathrm{ne}(x)} F_s(x, X_s) \tag{11.57}$$

其中 $\mathrm{ne}(x)$ 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 11.57 代入 11.56，交换加和与乘积的顺序，我们有

$$\begin{aligned} p(x) &= \prod_{s\in \mathrm{ne}(x)} \left[\sum_{\boldsymbol{X}_s} F_s(x, X_s)\right] \\ &= \prod_{s\in \mathrm{ne}(x)} \mu_{f_s\to x}(x) \end{aligned} \tag{11.58}$$

这里我们引入了一组新的函数 $\mu_{f_s \to x}(x)$，定义为

$$\mu_{f_s \to x}(x) \equiv \sum_{\boldsymbol{X}_s} F_s(x, X_s) \tag{11.59}$$

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，我们再次回到上图。我们注意到每个因子 $F_s(x, X_s)$ 有一个因子图 (因子子图)，因此本身可以被分解。特别地，我们有

$$F_s(x, X_s) = f_s(x, x_1, \ldots, x_M) G_1(x_1, X_{s1}) \ldots, G_M(x_M, X_{sM}) \tag{11.60}$$

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 $x_1, \ldots, x_M$。因此也可以记作 $\boldsymbol{x}_s$。

将公式 11.60 代入公式 11.59，我们有

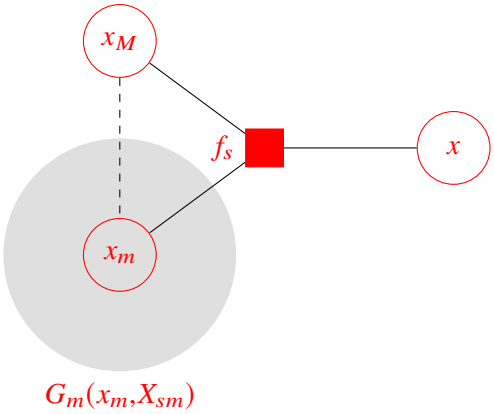
$$\begin{aligned} \mu_{f_s \to x}(x) &= \sum_{x_1} \cdots \sum_{x_M} f_s(x, x_1, \ldots, x_M) \prod_{m \in \mathrm{ne}(f_s) \backslash x} \left[\sum_{X_{sm}} G_m(x_m, X_{sm}) \right] \\ &= \sum_{x_1} \cdots \sum_{x_M} f_s(x, x_1, \ldots, x_M) \prod_{m \in \mathrm{ne}(f_s) \backslash x} \mu_{x_m \to f_s}(x_m) \end{aligned} \tag{11.61}$$

其中 $\mathrm{ne}(f_s)$ 表示因子结点 f_s 的相邻变量结点的集合，$\mathrm{ne}(f_s) \backslash x$ 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

$$\mu_{x_m \to f_s}(x_m) \equiv \sum_{X_{sm}} G_m(x_m, X_{sm}) \tag{11.62}$$

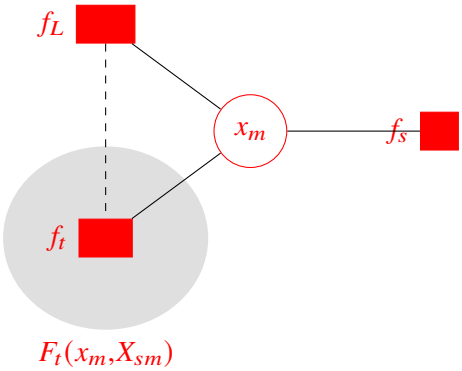
于是，我们引入了两类不同的信息。一类信息是从因子结点到变量结点的信息，记作 $\mu_{f \to x}(x)$，另一类信息是从变量结点到因子结点的信息，记作 $\mu_{x \to f}(x)$。在任何一种情况下，我们看到沿着一条链接传递的信息总是一个函数，这个函数是与那个链接相连的变量结点相关的变量的函数。

公式 11.61 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：计算沿着所有进行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)。根据下图



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

$$G_m(x_m, X_{sm}) = \prod_{l \in \mathrm{ne}(x_m) \setminus f_s} F_l(x_m, X_{lm}) \tag{11.63}$$

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 11.57 表示的图的形式完全相同。将公式 11.63 代入 11.62，我们可以得到

$$\begin{aligned} \mu_{x_m \to f_s}(x_m) &= \prod_{l \in \mathrm{ne}(x_m) \setminus f_s} \left[\sum_{X_{lm}} F_l(x_m, X_{lm}) \right] \\ &= \sum_{l \in \mathrm{ne}(x_m) \setminus f_s} \mu_{f_l \to x_m}(x_m) \end{aligned} \tag{11.64}$$

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 11.59。

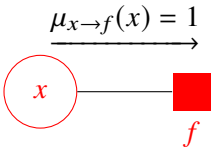
因此，为了计算从一个变量结点到相邻因子结点沿着链接传递的信息，我们只需简单地在其他所有结点上对输入信息取乘积。注意，任何只有两个相邻结点的变量结点无需参

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。

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 11.64 的定义，我们看到如果一个叶结点是一个变量结点，那么沿着与它唯一相连的链接发送的信息为

$$\mu_{x\to f}(x) = 1 \tag{11.65}$$

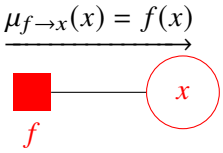
如图所示



类似地，如果叶结点是一个因子结点，那么我们根据公式 11.61 可以看到，发送的信息的形式为

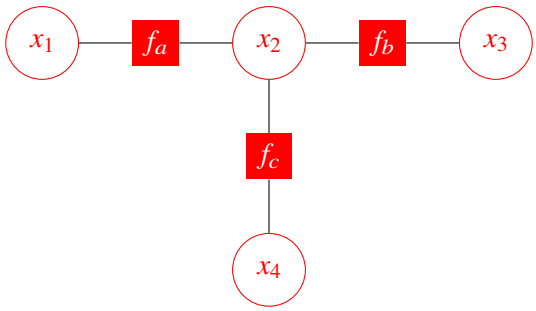
$$\mu_{f\to x}(x) = f(x) \tag{11.66}$$

如果所示



现在，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 11.65 和公式 11.66，初始化图的叶结点的信息。之后，递归地应用信息传递步骤 11.61 和 11.64，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 11.58 进行计算。

现在考虑一个简单的例子来说明加和-乘积算法。如图所示



图示给出了一个简单的 4 节点因子图，它的示归一化联合概率分布为

$$\tilde{p}(\boldsymbol{x}) = f_a(x_1, x_2) f_b(x_2, x_3) f_c(x_2, x_4) \tag{11.67}$$

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

$$\mu_{x_1 \to f_a}(x_1) = 1 \tag{11.68}$$

$$\mu_{f_a \to x_2}(x_2) = \sum_{x_1} f_a(x_1, x_2) \tag{11.69}$$

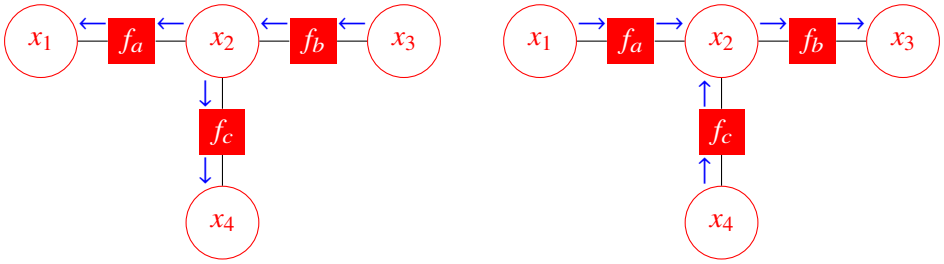
$$\mu_{x_4 \to f_c}(x_4) = 1 \tag{11.70}$$

$$\mu_{f_c \to x_2}(x_2) = \sum_{x_4} f_c(x_2, x_4) \tag{11.71}$$

$$\mu_{x_2 \to f_b}(x_2) = \mu_{f_a \to x_2}(x_2) \mu_{f_c \to x_2}(x_2) \tag{11.72}$$

$$\mu_{f_b \to x_3}(x_3) = \sum_{x_2} f_b(x_2, x_3) \mu_{x_2 \to f_b}(x_2) \tag{11.73}$$

信息流的方向如图所示



(a) 从根结点向叶结点传递。(b) 从叶结点向根结点传递。

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 11.58，使用上面的结果将信息替换掉，我们有

$$\begin{aligned} \tilde{p}(x_2) &= \mu_{f_a \to x_2}(x_2) \mu_{f_b \to x_2}(x_2) \mu_{f_c \to x_2}(x_2) \\ &= \left[\sum_{x_1} f_a(x_1, x_2)\right] \left[\sum_{x_3} f_b(x_2, x_3)\right] \left[\sum_{x_4} f_c(x_2, x_4)\right] \\ &= \sum_{x_1} \sum_{x_3} \sum_{x_4} \tilde{p}(\boldsymbol{x}) \end{aligned} \tag{11.74}$$

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 $\boldsymbol{x}$ 划分为隐含变量 $\boldsymbol{h}$ 和观测变量 $\boldsymbol{v}$，且 $\boldsymbol{v}$ 的观测值被记作 $\hat{\boldsymbol{v}}$。然后，我们简单地将联合概率分布 $p(\boldsymbol{x})$ 乘以 $\prod_i I(v_i, \hat{v}_i)$，其中如果

$v = \hat{v}$，则 $I(v_i, \hat{v}_i) = 1$，否则 $I(v, \hat{v}) = 0$。这个乘积对应于 $p(\boldsymbol{h}, \boldsymbol{v} = \hat{\boldsymbol{v}})$，因此 $p(\boldsymbol{h}|\boldsymbol{v} = \hat{\boldsymbol{v}})$ 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 $p(h_i|\boldsymbol{v} = \hat{\boldsymbol{v}})$，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。$\boldsymbol{v}$ 中变量上的任意求和式就退化成了单一的项。

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 $p(\boldsymbol{x})$ 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。

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，这会给出一组值，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 $\boldsymbol{x}^{\text{最大}}$，使得联合概率分布达到最大值，即

$$\boldsymbol{x}^{\text{最大}} = \arg\max_{\boldsymbol{x}} p(\boldsymbol{x}) \tag{11.75}$$

这样，联合概率分布的对应值为

$$p(\boldsymbol{x}^{\text{最大}}) = \max_{\boldsymbol{x}} p(\boldsymbol{x}) \tag{11.76}$$

通常，$\boldsymbol{x}^{\text{最大}}$ 与 x_i^* 的集合不同。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 $p(\boldsymbol{x})$ 的 $\boldsymbol{x}$ 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

$$\max_{\boldsymbol{x}} p(\boldsymbol{x}) = \max_{x_1} \dots \max_{x_M} p(\boldsymbol{x}) \tag{11.77}$$

其中 M 是变量的总数。之后，使用 $p(\boldsymbol{x})$ 的用因子乘积形式表示的展开式替换 $p(\boldsymbol{x})$ 即可。

在推导加和-乘积臭汗时，我们使用了乘法的分配律。这里，我们使用最大化算符的类似定律

$$\max(ab, ac) = a\max(b, c) \tag{11.78}$$

这对于 $a \geqslant 0$ 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 11.44 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

$$\begin{aligned} \max_{\boldsymbol{x}} p(\boldsymbol{x}) &= \frac{1}{Z} \max_{x_1} \dots \max_{x_N} [\psi_{1,2}(x_1, x_2) \dots \psi_{N-1,N}(x_{N-1}, x_N)] \\ &= \frac{1}{Z} \max_{x_1} \left[\max_{x_2} \left[\psi_{1,2}(x_1, x_2) \left[\dots \max_{x_N} \psi_{N-1,N}(x_{N-1}, x_N) \right] \dots \right] \right] \end{aligned} \tag{11.79}$$

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。

我们可以将这个结点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 11.54 代入公式 11.77 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到当前的问题中。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 $p(\boldsymbol{x})$ 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

$$\ln\left(\max_{\boldsymbol{x}} p(\boldsymbol{x})\right) = \max_{\boldsymbol{x}} \ln p(\boldsymbol{x}) \tag{11.80}$$

分配性质仍然成立，因为

$$\max(a+b, a+c) = a + \max(b, c) \tag{11.81}$$

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

$$\mu_{f_s \to x}(x) = \max_{x_1,\ldots,x_M} \left[\ln f(x, x_1, \ldots, x_M) + \sum_{m \in \mathrm{ne}(f)\setminus x} \mu_{x_m \to f}(x_m)\right] \tag{11.82}$$

$$\mu_{x \to f}(x) = \sum_{l \in \mathrm{ne}(x)\setminus f} \mu_{f_l \to x}(x) \tag{11.83}$$

最开始的由叶结点发送的信息可以通过类比公式得到，结果为

$$\mu_{f \to x}(x) = \ln f(x) \tag{11.84}$$

$$\mu_{x \to f}(x) = 0 \tag{11.85}$$

根结点处的最大概率为

$$p^{\text{最大}} = \max_{\boldsymbol{x}} \left[\sum_{s \in \mathrm{ne}(x)} \mu_{f_s \to x}(x)\right] \tag{11.86}$$

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 11.86 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

$$x^{\text{最大}} = \arg\max_{x} \left[\sum_{s \in \text{ne}(x)} \mu_{f_s \to x}(x) \right] \tag{11.87}$$

使用动态规划的方法应用在图模型上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可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。

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)。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。

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法 (variational)。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，这两种方法将在后续章节详细讨论。

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找结果。这种方法被称为循环置信传播 (loopy belief propagation)。

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。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。

11.5 隐马尔可夫模型

隐马尔可夫模型 (hidden Markov model,HMM) 是可用于标注问题的统计学习模型，描述由隐藏的马尔可夫链随机生成观测序列的过程，属于生成模型。本节首先介绍隐马尔可

夫模型的基本概念，然后分别叙述隐马尔可夫模型的概率计算算法、学习算法以及预测算法。隐马尔可夫模型在语音识别、自然语言处理、生物信息、模式识别等领域有着广泛的应用。

隐马尔可夫模型的基本概念

隐马尔可夫模型是关于时序的概率模型，描述由一个隐藏的马尔可夫链随机生成不可观测的状态随机序列，再由各个状态生成一个观测而产生观测随机序列的过程。隐藏的马尔可夫链随机生成的状态的序列，称为状态序列 (state sequence)；每个状态生成一个观测，而由此产生的观测的随机序列，称为观测序列 (observation sequence)。序列的每一个位置又可以看作是一个时刻。隐马尔可夫模型由初始概率分布、状态转移概率分布以及观测概率分布确定。隐马尔可夫模型定义如下：

设 Q 是所有可能的状态的集合，V 是所有可能的观测的集合。

$$Q = \{q_1, q_2, \ldots, q_N\}, \quad V = \{v_1, v_2, \ldots, v_M\} \tag{11.88}$$

设 N 是所有可能的状态数，M 是可能的观测数。

I 是长度为 T 的状态序列，O 是对应的观测序列。

$$I = \{i_1, i_2, \ldots, i_T\}, \quad O = \{o_1, o_2, \ldots, o_T\} \tag{11.89}$$

A 是状态转移概率矩阵：

$$A = \left[a_{ij}\right]_{N\times N} \tag{11.90}$$

其中，

$$a_{ij} = P(i_{t+1} = q_j | i_t = q_i), i = 1, 2, \ldots, N; j = 1, 2, \ldots, N \tag{11.91}$$

是在时刻 t 处于状态 q_i 的条件下在时刻 $t + 1$ 转移到状态 q_j 的概率。

B 是观测概率矩阵：

$$B = \left[b_j(k)\right]_{N\times N} \tag{11.92}$$

其中，

$$b_j(k) = P(o_t = v_k | i_t = q_j), k = 1, 2, \ldots, M; j = 1, 2, \ldots, N \tag{11.93}$$

是在时刻 t 处于状态 q_j 的条件下生成观测 v_k 的概率。π 是初始状态概率向量：

$$\pi = (\pi_i) \tag{11.94}$$

其中，

$$\pi_i = P(i_1 = q_i), i = 1, 2, \ldots, N \tag{11.95}$$

是时刻 $t = 1$ 处于状态 q_i 的概率。

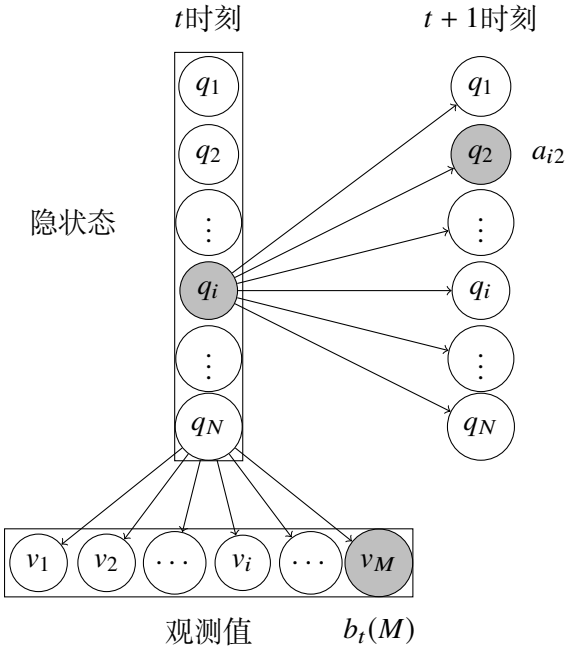
隐马尔可夫模型由初始状态概率向量 π、状态转移概率矩阵 A 和观测概率矩阵 B 决

定。π 和 A 决定状态序列，B 决定观测序列。因此，隐马尔可夫模型 λ 可以用三元符号表示，即

$$\lambda = (A, B, \pi) \tag{11.96}$$

A, B, π 称为隐马尔可夫模型的三要素。从定义可知，隐马尔可夫模型作了两个基本假设：

1. 齐次马尔可夫性假设，即假设隐藏的马尔可夫链在任意时刻 t 的状态只依赖于其前一时刻的状态，与其他时刻的状态及观测无关，也与时刻 t 无关。
2. 观测独立性假设，即假设任意时刻的观测只依赖于该时刻的马尔可夫链的状态，与其他观测及状态无关。



隐马尔可夫模型的 3 个基本问题

1. 概率计算问题。给定模型 $\lambda = (A, B, \pi)$ 和观测序列 $O = (o_1, o_2, \ldots, o_T)$，计算在模型 λ 下观测序列 O 出现的概率 $P(O|\lambda)$
2. 学习问题。已知观测序列 $O = (o_1, o_2, \ldots, o_T)$，估计模型 $\lambda = (A, B, \pi)$ 参数，使得在该模型下观测序列概率 $P(O|\lambda)$ 最大。即用极大似然估计的方法估计参数。
3. 预测问题。也称为解码（decoding）问题。已知模型 $\lambda = (A, B, \pi)$ 和观测序列 $O = (o_1, o_2, \ldots, o_T)$，求对给定观测序列条件概率 $P(I|\lambda)$ 最大的状态序列 $I = (i_1, i_2, \ldots, i_T)$。即给定观测序列，求最有可能的对应的状态序列。

问题一：概率计算算法

直接计算法

给定模型 $\lambda = (A, B, \pi)$ 和观测序列 $O = (o_1, o_2, \ldots, o_T)$，计算在模型 λ 下观测序列 O 出现的概率 $P(O|\lambda)$。最直接的方法是按概率公式直接计算。

$$\begin{aligned} P(O|\lambda) &= \sum_{I} P(O|I, \lambda)P(I|\lambda) \\ &= \sum_{i_1, i_2, \ldots, i_T} \pi_{i_1} b_{i_1}(o_1) a_{i_1 i_2} \ldots a_{i_{T-1} i_T} b_{i_T}(o_T) \end{aligned} \tag{11.97}$$

利用公式计算量大，这种算法不可行。

前向算法

首先定义前现概率

定义 11.1. 前向概率

给定隐马尔可夫模型 λ，定义到时刻 t 部分观测序列为 $O = (o_1, o_2, \ldots, o_t)$ 且状态为 q_i 的概率为前向概率，记作

$$\alpha_t(i) = P(o_1, o_2, \ldots, o_t, i_t = q_i|\lambda) \tag{11.98}$$

可以递推地求得前向概率 $\alpha_t(i)$ 及观测序列概率 $P(O|\lambda)$

输入：隐马尔可夫模型 λ，观测序列 O；

输出：观测序列概率 $P(O|\lambda)$

(1) 初值

$$\alpha_1(i) = \pi_i b_i(o_1), \qquad i = 1, 2, \ldots, N \tag{11.99}$$

(2) 递推，对 $t = 1, 2, \ldots, T-1$

$$\alpha_{t+1}(i) = \left[\sum_{j=1}^{N} \alpha_t(j) a_{ji}\right] b_i(o_{t+1}) \tag{11.100}$$

(3) 终止

$$P(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i) \tag{11.101}$$

前向算法实际是基于“状态序列的路径结构”递推计算 $P(O|\lambda)$ 的算法。前身算法的高效的关键是其局部计算前向概率，然后利用路径结构将前向概率“递推”到全局，得到 $P(O|\lambda)$.

后向概率

定义 11.2. 后向算法

给定隐马尔可夫模型 λ，定义在时刻 t 状态为 q_i 的条件下，从 $t+1$ 到 T 的部分观测序列为 $o_{t+1}, o_{t+2}, \ldots, o_T$ 的概率为后向概率，记作

$$\beta_t(i) = P(o_{t+1}, o_{t+2}, \ldots, o_T | i_t = q_i, \lambda) \tag{11.102}$$

可以用递推的方法求得后向概率 $\beta_t(i)$ 及观测序列概率 $P(O|\lambda)$

输入：隐马尔可夫模型 λ，观测序列 O；

输出：观测序列概率 $P(O|\lambda)$

(1) 初值

$$\beta_T(i) = 1, \qquad i = 1, 2, \ldots, N \tag{11.103}$$

(2) 递推，对 $t = T-1, T-2, \ldots, 1$

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(o_{t+1}) \beta_{t+1}(j), \quad i = 1, 2, \ldots, N \tag{11.104}$$

(3) 终止

$$P(O|\lambda) = \sum_{i=1}^{N} \pi_i b_i(o_1) \beta_1(i) \tag{11.105}$$

利用前向概率和后向概率的定义可以将观测序列概率 $P(O|\lambda)$ 统一写成

$$P(O|\lambda) = \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j), \quad t = 1, 2, \ldots, T-1 \tag{11.106}$$

一些概率与期望值的计算

1. 给定模型 λ 和观测 O，在时刻 t 处于状态 q_j 的概率。记 $\gamma_t(i)$
2. 给定模型 λ 和观测 O，在时刻 t 处于状态 q_j 且在时刻 $t+1$ 处于状态 q_j 的概率。记 $\xi_t(i, j)$
3. 一些有用的期望值

问题二：学习算法

隐马尔可夫模型的学习，根据训练数据是包括观测序列和对应的状态序列还是只有观测序列，可以分别由监督学习与非监督学习实现。

监督学习算法

假设训练数据包含 S 个长度相同的观测序列和对应的状态序列 $\{(O_1, I_1), \ldots, (O_S, I_S)\}$，那么可以利用**极大似然法**来估计隐马尔可夫模型的参数。具体方法如下。

1. 转移概率 a_{ij} 的估计

   设样本中时刻 t 处于状态 i 时刻 $t+1$ 转移到状态 j 的频数为 A_{ij}，那么状态转移概率 a_{ij} 的估计是

$$\hat{a}_{ij} = \frac{A_{ij}]}{\sum_{j=1}^{N} A_{ij}}, \qquad i = 1, 2, \ldots, N; j = 1, 2, \ldots, N \tag{11.107}$$

2. 观测概率 $b_j(k)$ 的估计

   设样本中状态为 j 并观测为 k 的频数是 B_{jk}，那么状态为 j 观测为 k 的概率 $b_j(k)$ 的估计是

$$\hat{b}_j(k) = \frac{B_{jk}}{\sum_{k=1}^{M} B_{jk}}, \qquad j = 1, 2, \ldots, k = 1, 2, \ldots, M \tag{11.108}$$

3. 初始状态概率 π_i 的估计 $\hat{\pi}_i$ 为 S 个样本中初始状态为 q_i 的频率

Baum-Welch 算法

假设训练数据包含 S 个长度相同的观测序列 $\{(O_1, I_1), \ldots, (O_S, I_S)\}$ 而没有对应的状态序列，目标是学习隐马尔可夫模型 $\lambda = (A, B, \pi)$ 的参数。我们将观测序列数据看作观测数据 O，状态序列数据看作不可观测的隐数据 I，那么隐马尔可夫模型事实上是一个含有隐变量的概率模型

$$P(O|\lambda) = \sum_{I} P(O|I, \lambda)P(I|\lambda) \tag{11.109}$$

它的参数学习可以由 EM 算法实现。

1. 确定完全数据的对数似然函数

   所有观测数据写成 $O = (o_1, o_2, \ldots, o_T)$，所有隐数据写成 $I = (i_1, i_2, \ldots, i_T)$ 完全数据是 $(O, I) = (o_1, o_2, \ldots, o_T, i_1, i_2, \ldots, i_T)$。完全数据的对数似然函数是 $logP(O, I|\lambda)$

2. EM 算法的 E 步：求 Q 函数 $Q(\lambda, \bar{\lambda})$

$$E_I[logP(O, I|\lambda)|O, \bar{\lambda}] = \sum_{I} logP(O, I|\lambda)P(I|O, \bar{\lambda}) \tag{11.110}$$

$$Q(\lambda, \bar{\lambda}) = \sum_{I} logP(O, I|\lambda)P(O, I|\bar{\lambda}) \tag{11.111}$$

$P(I|O, \bar{\lambda}) = P(I, O|\bar{\lambda})/P(O|\bar{\lambda})$ 省略了对 λ 而言的常数因子。

$$P(O, I|\lambda) = \pi_{i_1} b_{i_1}(o_1) a_{i_1 i_2} \ldots a_{i_{T-1} i_T} b_{i_T}(o_T) \tag{11.112}$$

于是函数 $Q(\lambda, \bar{\lambda})$ 可以写成：

$$\begin{aligned} Q(\lambda, \bar{\lambda}) = & \sum_{I} log\pi_{i_1} P(O, I|\bar{\lambda}) \\ & + \sum_{I} \left(\sum_{t=1}^{T-1} log a_{i_t i_{t+1}} \right) P(O, I|\bar{\lambda}) \\ & + \sum_{I} \left(\sum_{t=1}^{T} log b_{i_t}(o_t) \right) P(O, I|\bar{\lambda}) \end{aligned} \tag{11.113}$$

3. EM 算法的 M 步：极大化 $Q(\lambda, \bar{\lambda})$ 求模型参数 A, B, π

   由于要极大化的参数在式 11.113 中单独地出现在 3 个项中，所以只需对各项分别极大化。注意到

$$\sum_{i=1}^{N} \pi_i = 1, \sum_{j=1}^{N} a_{ij} = 1, \sum_{k=1}^{M} b_j(k) = 1,$$

   利用拉格朗日乘子法，求得

$$\pi_i = \frac{P(O, i_1 = i|\bar{\lambda})}{P(O|\bar{\lambda})} = \gamma_1(i) \tag{11.114}$$

$$a_{ij} = \frac{\sum_{t=1}^{T-1} P(O, i_t = i, i_{t+1} = j|\bar{\lambda})}{\sum_{t=1}^{T-1} P(O, i_t = i|\bar{\lambda})} = \frac{\sum_{t=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \tag{11.115}$$

$$b_j(k) = \frac{\sum_{t=1}^{T} P(O, i_t = j|\bar{\lambda}) I(o_t = v_k)}{\sum_{t=1}^{T} P(O, i_t = j|\bar{\lambda})} = \frac{\sum_{t=1, o_t = v_k}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)} \tag{11.116}$$

问题三：预测算法

近似算法

近似算法的想法是，在每个时刻 t 选择在该时刻最有可能出现的状态 i_t^*，从而得到一个状态序列 $I^* = (i_1^*, i_2^*, \ldots, i_T^*)$，将它作为预测的结果。近似算法的优点是计算简单，其缺点是不能保证预测的状态序列整体是最有可能的状态序列，因为预测的状态序列可能有实际不发生的部分。

维特比算法

维特比算法实际是用动态规划解隐马尔可夫模型预测问题，即用动态规划 (dynamic programming) 求概率最大路径 (最优路径)。这时一条路径对应着一个状态序列。

首先导入两个变量 δ 和 ψ。定义在时刻 t 状态为 i 的所有单个路径 $i_1, i_2, \ldots, i_t$ 中概率

最大值为

$$\begin{aligned}\delta_{t+1} &= \max_{i_1,i_2,\ldots,i_t} P(i_t = i, i_{t-1},\ldots,i_1, o_{t+1},\ldots,o_1|\lambda) \\ &= \max_{1\leqslant j\leqslant N}[\delta_t(j)a_{ji}]b_i(o_{t+1}), \quad i = 1,2,\ldots,N\end{aligned} \tag{11.117}$$

定义在时刻 t 状态为 i 的所有单个路径 $i_1, i_2, \ldots, i_t$ 中概率最大的路径的第 $t-1$ 个结点为

$$\psi_t(i) = arg \max_{1\leqslant j\leqslant N}[\delta_{t-1}(j)a_{ji}], \quad i = 1,2,\ldots,N \tag{11.118}$$

(维特比算法)

输入:模型 $\lambda = (A, B, \pi)$,观测序列 $O = (o_1, o_2, \ldots, o_T)$;

输出:最优路径 $I^* = (i_1^*, i_2^*, \ldots, i_T^*)$

1. 初始化

$$\begin{aligned}\delta_1 &= \pi_i b_i(o_1), \\ \psi_1(i) &= 0, \qquad i = 1,2,\ldots,N\end{aligned}$$

2. 递推。对 $t = 2, 3, \ldots, T$

$$\begin{aligned}\delta_{t+1} &= \max_{1\leqslant j\leqslant N}[\delta_t(j)a_{ji}]b_i(o_{t+1}) \\ \psi_t(i) &= arg \max_{1\leqslant j\leqslant N}[\delta_{t-1}(j)a_{ji}], \quad i = 1,2,\ldots,N\end{aligned}$$

3. 终止

$$\begin{aligned}P^* &= \max_{1\leqslant i\leqslant N}\delta_T(i) \\ i_T^* &= arg \max_{1\leqslant i\leqslant N}\delta_T(i)\end{aligned}$$

4. 最优路径回溯。对 $t = T-1, T-2, \ldots, 1$

$$i_t^* = \psi_{t+1}(i_{t+1}^*)$$

求得最佳路径 $I^* = (i_1^*, i_2^*, \ldots, i_T^*)$

11.6 条件随机场

第 12 章 混合模型和 EM 算法

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。连续潜在变量是后面章节的主题。

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。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成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法介绍 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法讨论 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。在近似推断章节中，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。

12.1 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的聚类的问题。假设我们有一个数据集 $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$，它由 D 维欧几里德空间中的随机变量 $\boldsymbol{x}$ 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 $k = 1, \ldots, K$，且 $\boldsymbol{\mu}_k$ 是与第 k 个聚类关联的一个代表。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 $\boldsymbol{x}_n$，我们引入一组对应的二值指示变量 $r_{nk} \in \{0, 1\}$，其中 $k = 1, \ldots, K$ 表示数据点 $\boldsymbol{x}_n$ 属于 K 个聚类中的哪一个，从而如果数据点 $\boldsymbol{x}_n$ 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

$$J = \sum_{n=1}^{N}\sum_{k=1}^{K} r_{nk}\|\boldsymbol{x}_n - \boldsymbol{\mu}_k\|^2 \tag{12.1}$$

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)t 步骤和 M(最大化) 步骤。

首先考虑确定 r_{nk}。由于公式 12.1 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 $|\boldsymbol{x}_n - \boldsymbol{\mu}_k\|^2$ 最小，我们就令 $r_{nk} = 1$。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

$$r_{nk} = \begin{cases} 1 & \text{如果} k = \arg\min_j |\boldsymbol{x}_n - \boldsymbol{\mu}_j\|^2 \\ 0 & \text{其他情况} \end{cases} \tag{12.2}$$

现在考虑 r_{nk} 固定时，关于 $\boldsymbol{\mu}_k$ 的最优化。目标函数 J 是 $\boldsymbol{\mu}_k$ 的一个二次函数，令它关于 $\boldsymbol{\mu}_k$ 的导数等于零，即可达到最小值，即

$$2\sum_{n=1}^{N} r_{nk}(\boldsymbol{x}_n - \boldsymbol{\mu}_k) = 0 \tag{12.3}$$

解出 $\boldsymbol{\mu}_k$，结果为

$$\boldsymbol{\mu}_k = \frac{\sum_n r_{nk}\boldsymbol{x}}{\sum_n r_{nk}} \tag{12.4}$$

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。

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德距离。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，例如数据组织成树结构，使得相信的数据点属于同一个子树。另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。

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，方法是：将 Robbins-Monro 步骤应用于寻找回归函数的根问题中，其中回归函数由公式 12.1 给出的 J 关于 $\boldsymbol{\mu}_k$ 的导数给出。这产生了顺序更新算法，其中对于每个数据点 $\boldsymbol{x}_n$，我们使用下式更新最近的代表向量 $\boldsymbol{\mu}_k$。

$$\boldsymbol{\mu}_k^{新} = \boldsymbol{\mu}_k^{旧} + \eta_n(\boldsymbol{x}_n - \boldsymbol{\mu}_k^{旧}) \tag{12.5}$$

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。

12.2 一般形式的 EM 算法

EM 算法是一种迭代算法，1977 年由 Dempster 等人总结提出的，用于含有隐变量 (hidden variable) 的概率模型参数的极大似然估计，或极大后验概率估计。EM 算法的每次迭代由两步组成：E 步，求期望 (expectation)；M 步，求极大 (maximization)。所以这一算法称为期望极大算法 (expection maximization)，简称 EM 算法。

概率模型有时既含有观测变量 (observable variable)，又含有隐变量或潜在变量 (latent variable)。如果概率模型的变量都是观测变量，那么给定数据，可以直接用极大似然估计法，或贝叶斯估计法估计模型参数。但是，当模型含有隐变量时，就不能简单地使用这些估计方法。EM 算法就是含有隐变量的概率模型参数的极大似然估计法，或极大后验概率估计法。仅讨论极大似然估计，极大后验概率估计与其类似。

EM 算法的导出

为什么 EM 算法能近似实验对观测数据的极大似然估计呢？下面通过近似求解观测数据的对数似然函数的极大化问题来导出 EM 算法。

我们面对一个含有隐变量的概率模型，目标是极大化观测数据 (不完全数据)Y 关于参数 θ 的对数似然函数，即极大化

$$\begin{aligned} L(\theta) &= logP(Y|\theta) = log\sum_Z P(Y,Z|\theta) \\ &= log\left(\sum_Z P(Y|Z,\theta)P(Z|\theta)\right) \end{aligned} \tag{12.6}$$

注意到这一极大化的主要困难是式 12.6 中有未观测数据并有包含和 (或积分) 的对数。

事实上，EM 算法是通过迭代逐步极大化 $L(\theta)$ 的。假设在第 i 次迭代后 θ 的估计值是 $\theta^{(i)}$。我们希望新估计值 θ 能使 $L(\theta)$ 增加，即 $L(\theta) > L(\theta^{(i)})$，并逐步达到极大值。为此，考虑两者的差：

$$L(\theta) - L(\theta^{(i)}) = log\left(\sum_Z P(Y|Z,\theta)P(Z|\theta)\right) - logP(Y|\theta^{(i)}) \tag{12.7}$$

利用 Jensen 不等式[1](Jensen inequality) 得到其下界：

$$\begin{aligned} L(\theta) - L(\theta^{(i)}) &= log\left(\sum_Z P(Y|Z,\theta^{(i)})\frac{P(Y|Z,\theta)P(Z|\theta)}{P(Y|Z,\theta^{(i)})}\right) - logP(Y|\theta^{(i)}) \\ &\geqslant \sum_Z P(Y|Z,\theta^{(i)})log\frac{P(Y|Z,\theta)P(Z|\theta)}{P(Y|Z,\theta^{(i)})} - logP(Y|\theta^{(i)}) \\ &= \sum_Z P(Y|Z,\theta^{(i)})log\frac{P(Y|Z,\theta)P(Z|\theta)}{P(Y|Z,\theta^{(i)})P(Y|\theta^{(i)})} \end{aligned} \tag{12.8}$$

令

$$B(\theta,\theta^{(i)}) = L(\theta^{(i)}) + \sum_Z P(Y|Z,\theta^{(i)})log\frac{P(Y|Z,\theta)P(Z|\theta)}{P(Y|Z,\theta^{(i)})P(Y|\theta^{(i)})} \tag{12.9}$$

则

$$L(\theta) \geqslant B(\theta,\theta^{(i)}) \tag{12.10}$$

即函数 $B(\theta,\theta^{(i)})$ 是 $L(\theta)$ 的一个下界，因此**任何可以使 $B(\theta,\theta^{(i)})$ 增大的 θ，也可以使 $L(\theta)$ 增大**。为了使 $L(\theta)$ 尽可能大的增大，选择 $\theta^{(i+1)}$ 使 $B(\theta,\theta^{(i)})$ 达到极大，即

$$\theta^{(i+1)} = arg\max_\theta B(\theta,\theta^{(i)}) \tag{12.11}$$

现在求 $\theta^{(i+1)}$ 的表达式。省去对 θ 的极大化而言是常数的项，有

$$\begin{aligned} \theta^{(i+1)} &= arg\max_\theta\left(L(\theta^{(i)}) + \sum_Z P(Y|Z,\theta^{(i)})log\frac{P(Y|Z,\theta)P(Z|\theta)}{P(Y|Z,\theta^{(i)})P(Y|\theta^{(i)})}\right) \\ &= arg\max_\theta\left(\sum_Z P(Y|Z,\theta^{(i)})log(P(Y|Z,\theta)P(Z|\theta))\right) \\ &= arg\max_\theta\left(\sum_Z P(Y|Z,\theta^{(i)})log(P(Y,Z|\theta))\right) \\ &= arg\max_\theta \underline{Q(\theta,\theta^{(i)})} \end{aligned} \tag{12.12}$$

式 12.12 等价于 EM 算法的一次迭代，即求 Q 函数及其极大化。**EM 算法是通过不断求解下界的极大化逼近求解对数似然函数极大化的算法**。

定义 12.1. Q 函数

完全数据的对数似然函数 $logP(Y,Z|\theta)$ 关于在给定观测数据 Y 和当前参数 $\theta^{(i)}$ 下对未观测数据 Z 的条件概率分布 $P(Z|Y,\theta^{(i)})$ 的期望称为 Q 函数。即

$$Q(\theta,\theta^{(i)}) = E_Z[logP(Y,Z|\theta)|Y,\theta^{(i)}] \tag{12.13}$$

[1]这里用到的是 $log\sum_j \lambda_j y_j \geqslant \sum_j \lambda_j log y_j$

EM 算法

输入：观测变量数据 Y，隐变量数据 Z，联合分布 $P(Y,Z|\theta)$，条件分布 $P(Z|Y,\theta)$；

输出：模型参数 θ

1. 选择参数的初值 $\theta^{(0)}$，开始迭代；
2. E 步：记 $\theta^{(i)}$ 为第 i 次迭代参数 θ 的估计值，在第 i+1 次迭代的 E 步，计算

$$\begin{aligned} Q(\theta,\theta^{(i)}) &= E_Z[logP(Y,Z|\theta)|Y,\theta^{(i)}] \\ &= \sum_Z logP(Y,Z|\theta)P(Z|Y,\theta^{(i)}) \end{aligned} \tag{12.14}$$

这里，$P(Z|Y,\theta^{(i)})$ 是在给定观测数据 Y 和当前的参数估计 $\theta^{(i)}$ 下隐变量数据 Z 的条件概率分布；

3. M 步：求使 $Q(\theta,\theta^{(i)})$ 极大化的 θ，确定第 i+1 次迭代的参数的估计值 $\theta^{(i+1)}$

$$\theta^{(i+1)} = arg\max_\theta Q(\theta,\theta^{(i)}) \tag{12.15}$$

4. 重复第 (2)、(3) 步，直到收敛。

EM 算法的收敛性

EM 算法提供一种近似计算含有隐变量概率模型的极大似然估计的方法。EM 算法的最大优点是简单性和普适性。我们自然地要问：EM 算法得到的估计序列是否收敛？如果收敛，是否收敛到全局最大值或局部极大值？

定理 12.1

设 $P(Y|\theta)$ 为观测数据的似然函数，$\theta^{(i)}(i=1,2,\dots)$ 为 EM 算法得到的参数估计序列，$P(Y|\theta^{(i)})(i=1,2,\dots)$ 为对应的似然函数序列，则 $P(Y|\theta^{(i)})$ 是单调递增的，即

$$P(Y|\theta^{(i+1)}) \geqslant P(Y|\theta^{(i)}) \tag{12.16}$$

设 $L(\theta) = logP(Y|\theta)$ 为观测数据的对数似然函数，$L(\theta^{(i)})(i=1,2,\dots)$ 为对应的对数似然函数序列。

1. 如果 $P(Y|\theta)$ 有上界，则 $L(\theta^{(i)}) = logP(Y|\theta^{(i)})$ 收敛到某一值 L^*
2. 在函数 $Q(\theta,\theta')$ 与 $L(\theta)$ 满足一定条件下，由 EM 算法得到的参数估计序列 $\theta^{(i)}$ 的收敛值 θ^* 是 $L(\theta)$ 的稳定点 ♡

12.3 混合高斯

EM 算法的一个重要应用是高斯混合模型的参数估计。高斯混合模型应用广泛，在许多情况下，EM 算法是学习高斯混合模型 (Gaussian misture model) 的有效方法。

定义 12.2. 高斯混合模型

高斯混合模型是指具有如下形式的概率分布模型：

$$P(y|\theta) = \sum_{k=1}^{K} \alpha_k \phi(y|\theta_k) \tag{12.17}$$

其中，α_k 是系数，$\alpha_k \geqslant 0, \sum_{k=1}^{K} \alpha_k = 1$；$\phi(Y|\theta_k)$ 是高斯分布密度，$\theta_k = (\mu_k, \sigma_k^2)$，

$$\phi(y|\theta_k) = \frac{1}{\sqrt{2\pi}\sigma_k} exp\left(-\frac{(y-\mu_k)^2}{2\sigma_k^2}\right) \tag{12.18}$$

称为第 k 个模型。 ♣

高斯混合模型参数估计的 EM 算法

假设观测数据 $y_1, y_2, \ldots, y_N$ 由高斯混合模型生成，

$$P(y|\theta) = \sum_{k=1}^{K} \alpha_k \phi(y|\theta_k) \tag{12.19}$$

其中，$\theta = (\alpha_1, \alpha_2, \ldots, \alpha_K; \theta_1, \theta_2, \ldots, \theta_K)$。我们用 EM 算法估计高斯混合模型的参数 θ

1. 明确隐变量，写出完全数据的对数似然函数

   隐变量由 γ_{jk} 表示，其定义如下：

$$\gamma_{jk} = \begin{cases} 1, & \text{第 j 个观测来自第 k 个模型} \\ 0, & \text{否则} \end{cases} \tag{12.20}$$

   γ_{jk} 是 0-1 随机变量。那么完全数据是

$$(y_j, \gamma_{j1}, \gamma_{j2}, \ldots, \gamma_{jK}), \qquad j = 1, 2, \ldots, K$$

于是可以写出完全数据的似然函数

$$\begin{aligned}P(y,\gamma|\theta)&=\prod_{j=1}^{N}P(y_j,\gamma_{j1},\gamma_{j2},\ldots,\gamma_{jK}|\theta)\\&=\prod_{k=1}^{K}\prod_{j=1}^{N}[\alpha_k\phi(y|\theta_k)]^{\gamma_{jk}}\\&=\prod_{k=1}^{K}\alpha_k^{n_k}\prod_{j=1}^{N}[\phi(y|\theta_k)]^{\gamma_{jk}}\\&=\prod_{k=1}^{K}\alpha_k^{n_k}\prod_{j=1}^{N}\left[\frac{1}{\sqrt{2\pi}\sigma_k}exp\left(-\frac{(y-\mu_k)^2}{2\sigma_k^2}\right)\right]^{\gamma_{jk}}\end{aligned}$$

式中，$n_k=\sum_{j=1}^{N}\gamma_{jk}$，$\sum_{k=1}^{K}n_k=N$。(意思是选择第 k 个高斯模型的次数)

那么，完全数据的对数似然函数为

$$logP(y,\gamma|\theta)=\sum_{k=1}^{K}\left\{n_klog\alpha_k+\sum_{j=1}^{N}\gamma_{jk}\left[log\left(\frac{1}{\sqrt{2\pi}}\right)-log\sigma_k-\frac{1}{2\sigma_k^2}(y_j-\mu_k)^2\right]\right\}$$

2. EM 算法的 E 步：确定 Q 函数

$$\begin{aligned}Q(\theta,\theta^{(i)})&=E[logP(y,\gamma|\theta)|y,\theta^{(i)}]\\&=E\left\{\sum_{k=1}^{K}\left\{n_klog\alpha_k+\sum_{j=1}^{N}\gamma_{jk}\left[log\left(\frac{1}{\sqrt{2\pi}}\right)-log\sigma_k-\frac{1}{2\sigma_k^2}(y_j-\mu_k)^2\right]\right\}\right\}\\&=\sum_{k=1}^{K}\left\{\overbrace{n_klog\alpha_k}^{\text{常数}}+\sum_{j=1}^{N}E(\gamma_{jk})\left[log\left(\frac{1}{\sqrt{2\pi}}\right)-log\sigma_k-\frac{1}{2\sigma_k^2}(y_j-\mu_k)^2\right]\right\}\end{aligned}$$

这里需要计算 $E(\gamma_{jk}|y,\theta)$，记为 $\hat{\gamma}_{jk}$

$$\begin{aligned}\hat{\gamma}_{jk}&=E(\gamma_{jk}|y,\theta)=P(\gamma_{jk}=1|y,\theta)(\text{二项分布})\\&=\frac{P(\gamma_{jk}=1|y,\theta)}{\sum_{k=1}^{K}P(\gamma_{jk}=1|y,\theta)}\\&=\frac{P(y_j|\gamma_{jk}=1,\theta)P(\gamma_{jk}=1|\theta)}{\sum_{k=1}^{K}P(y_j|\gamma_{jk}=1,\theta)P(\gamma_{jk}=1|\theta)}\\&=\frac{\alpha_k\phi(y_j|\theta_k)}{\sum_{k=1}^{K}\alpha_k\phi(y_j|\theta_k)},\quad j=1,2,\ldots,N;k=1,2,\ldots,K\end{aligned}$$

$\hat{\gamma}_{jk}$ 是在当前模型参数下第 j 个观测数据来自第 k 个分模型的概率，称为分模型 k 对观测数据 y_j 的响应度。代入式中得

$$Q(\theta, \theta^{(i)}) = \sum_{k=1}^{K} \left\{ n_k log\alpha_k + \sum_{j=1}^{N} \hat{\gamma}_{jk} \left[log\left(\frac{1}{\sqrt{2\pi}}\right) - log\sigma_k - \frac{1}{2\sigma_k^2}(y_j - \mu_k)^2 \right] \right\} \tag{12.21}$$

3. 确定 EM 算法的 M 步迭代的 M 步是求函数 $Q(\theta, \theta^{(i)})$ 对 θ 的极大值，即求新一轮迭代的模型参数：

$$\theta^{(i+1)} = arg\ \max_{\theta} Q(\theta, \theta^{(i)})$$

分别对 μ_k, σ_k^2 求偏导并令其为 0，即可得到 $\hat{\mu}_k, \hat{\sigma}_k$。求 $\hat{a}_k$ 是在 $\sum_{k=1}^{K} = 1$ 条件下求偏导并令其为 0 得到。

$$\hat{\mu}_k = \frac{\sum_{j=1}^{N} \hat{\gamma}_{jk} y_j}{\sum_{j=1}^{N} \hat{\gamma}_{jk}}, \quad k = 1, 2, \ldots, K \tag{12.22}$$

$$\hat{\sigma}_k^2 = \frac{\sum_{j=1}^{N} \hat{\gamma}_{jk}(y_j - \mu_k)^2}{\sum_{j=1}^{N} \hat{\gamma}_{jk}}, \quad k = 1, 2, \ldots, K \tag{12.23}$$

$$\hat{\alpha}_k = \frac{n_k}{N} = \frac{\sum_{j=1}^{N} \hat{\gamma}_{jk}}{N}, \quad k = 1, 2, \ldots, K \tag{12.24}$$

(12.25)

4. 重复以上计算，直到对数似然函数值不再有明显的变化为止。

现将估计高斯混合模型参数的 EM 算法总结如下

输入：观测数据 $y_1, y_2, \ldots, y_N$，高斯混合模型；

输出：高斯混合模型参数。

(1) 取参数的初始值开始迭代

(2) E 步：依据当前模型参数，计算分模型 k 对观测数据 y_j 的响应度

$$\hat{\gamma}_{jk} = \frac{\alpha_k \phi(y_j|\theta_k)}{\sum_{k=1}^{K} \alpha_k \phi(y_j|\theta_k)}, \quad j = 1, 2, \ldots, N; k = 1, 2, \ldots, K$$

(3) M 步:计算新一轮迭代的模型参数

$$\hat{\mu}_k = \frac{\sum_{j=1}^{N} \hat{\gamma}_{jk} y_j}{\sum_{j=1}^{N} \hat{\gamma}_{jk}}, \quad k = 1, 2, \ldots, K$$

$$\hat{\sigma}_k^2 = \frac{\sum_{j=1}^{N} \hat{\gamma}_{jk} (y_j - \mu_k)^2}{\sum_{j=1}^{N} \hat{\gamma}_{jk}}, \quad k = 1, 2, \ldots, K$$

$$\hat{\alpha}_k = \frac{n_k}{N} = \frac{\sum_{j=1}^{N} \hat{\gamma}_{jk}}{N}, \quad k = 1, 2, \ldots, K$$

(4) 重复以上计算,直到对数似然函数值不再有明显的变化为止。

12.4 EM 的另一种观点

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 $\boldsymbol{X}$,其中第 n 行表示 $\boldsymbol{x}_n^T$。类似地,我们将所有潜在变量的集合记作 $\boldsymbol{Z}$,对应的行为 $\boldsymbol{z}_n^T$。所有模型参数的集合被记作 $\boldsymbol{\theta}$,因此对数似然函数为

$$\ln p(\boldsymbol{X}|\boldsymbol{\theta}) = \ln \left\{ \sum_{\boldsymbol{Z}} p(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\theta}) \right\} \tag{12.26}$$

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 $p(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\theta})$ 属于指数族分布,由于这个求和式的存在,边缘概率分布 $p(\boldsymbol{X}|\boldsymbol{\theta})$ 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。

第 13 章 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 $\boldsymbol{X}$ 的条件下，计算潜在变量 $\boldsymbol{Z}$ 的后验概率分布 $p(\boldsymbol{Z}|\boldsymbol{X})$，以及计算关于这个概率分布的期望。模型可以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人先验概率分布，并且被整合到了潜在变量集合中，记作向量 $\boldsymbol{Z}$。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性近似，方法大体分为两大类。随机方法，例如后面章节介绍的马尔可夫链蒙特卡罗方法，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，判断的一种取样方法是否生成了服从所需的概论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有适用。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。

前面的章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)。

13.1 变分推断

变分的方法起源于 18 世纪的欧拉、拉格朗日，以及其他的关于变分法 (calculus of variations) 的研究。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。

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 $\boldsymbol{Z}$。类似地，我们会把所有观测变量的集合记作 $\boldsymbol{X}$。例如，我们可能有 N 个独立同分布的数据，其中 $\boldsymbol{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ 且 $\boldsymbol{Z} = \{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_N\}$。我们的概率模型确定了联合概率分布 $p(\boldsymbol{X}, \boldsymbol{Z})$，我们的目标是找到后验概率分布 $p(\boldsymbol{Z}|\boldsymbol{X})$ 以及模型证据 $p(\boldsymbol{X})$ 的近似。我们可以将对数边缘概率分解，即

$$\begin{aligned} \ln p(\boldsymbol{X}) &= \ln p(\boldsymbol{X}, \boldsymbol{Z}) - \ln p(\boldsymbol{Z}|\boldsymbol{X}) \\ &= \ln \frac{p(\boldsymbol{X}, \boldsymbol{Z})}{q(\boldsymbol{Z})} - \ln \frac{p(\boldsymbol{Z}|\boldsymbol{X})}{q(\boldsymbol{Z})} \end{aligned} \tag{13.1}$$

等式左右两边乘以 $q(\boldsymbol{Z})$ 并对 z 求积分有

$$\begin{aligned} \text{左边} &= \int_{\boldsymbol{Z}} \ln p(\boldsymbol{X}) q(\boldsymbol{Z}) d\boldsymbol{Z} = \ln p(\boldsymbol{X}) \\ \text{右边} &= \underbrace{\int_{\boldsymbol{Z}} q(\boldsymbol{Z}) \cdot \ln \frac{p(\boldsymbol{X}, \boldsymbol{Z})}{q(\boldsymbol{Z})} d\boldsymbol{Z}}_{\text{ELBO}(evidence\ lower\ bound)} + \underbrace{\int_{\boldsymbol{Z}} -q(\boldsymbol{Z}) \cdot \ln \frac{p(\boldsymbol{Z}|\boldsymbol{X})}{q(\boldsymbol{Z})} d\boldsymbol{Z}}_{KL(q\|p)} \end{aligned} \tag{13.2}$$

即

$$\frac{num}{den} \ln p(\boldsymbol{X}) = \zeta(q) + \text{KL}(q\|p) \tag{13.3}$$

与之前一样，我们可以通过关于概率分布 $q(\boldsymbol{Z})$ 的最优化来使下界 $\zeta(q)$ 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 $q(\boldsymbol{Z})$ 的最优化来使下界 $\zeta(q)$ 达到最大值出现在 KL 散度等于零的时刻，此时 $q(\boldsymbol{Z})$ 等于后验概率分布 $p(\boldsymbol{Z}|\boldsymbol{X})$。即

$$\begin{aligned} -\int_{\boldsymbol{Z}} q(\boldsymbol{Z}) \cdot \ln \frac{p(\boldsymbol{Z}|\boldsymbol{X})}{q(\boldsymbol{Z})} d\boldsymbol{Z} &= 0 \\ \Rightarrow \frac{p(\boldsymbol{Z}|\boldsymbol{X})}{q(\boldsymbol{Z})} &= 1 \\ \Rightarrow q(\boldsymbol{Z}) &= p(\boldsymbol{Z}|\boldsymbol{X}) \end{aligned} \tag{13.4}$$

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 $q(\boldsymbol{Z})$ 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 $q(\boldsymbol{Z})$ 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好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 $q(\boldsymbol{Z}|\omega)$，它由参数集合 ω 控制。这样，下界 $\zeta(q)$ 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。

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 $q(\boldsymbol{Z})$ 的范围。假设我们将 $\boldsymbol{Z}$ 的元素划分成若干个互不相交的组，记作 $\boldsymbol{Z}_i$，其中 $i = 1, \ldots, M$。然后，我们假定 q 分布关于这些分布可以进行分解，即

$$q(\boldsymbol{Z}) = \prod_{i=1}^{M} q_i(\boldsymbol{Z}_i) \tag{13.5}$$

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 $q_i(\boldsymbol{Z}_i)$ 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)。

在所有具有公式 13.5 的形式的概率分布 $q(\boldsymbol{Z})$ 中，我们现在寻找下界 $\zeta(q)$ 最大的概率分布。于是，我们希望对 $\zeta(q)$ 关于所有的概率分布 $q_i(\boldsymbol{Z}_i)$ 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 13.5 代入公式 13.2，然后分离出依赖于一个因子 $q_j(\boldsymbol{Z}_j)$ 的项。为了记号的简洁，我们简单地将 $q_j(\boldsymbol{Z}_j)$ 记作 q_j，这样我们有

$$\begin{aligned}
\zeta(q) &= \int_{\boldsymbol{Z}} q(\boldsymbol{Z}) \cdot \ln \frac{p(\boldsymbol{X}, \boldsymbol{Z})}{q(\boldsymbol{Z})} d\boldsymbol{Z} \\
&= \underbrace{\int_{\boldsymbol{Z}} q(\boldsymbol{Z}) \cdot \ln p(\boldsymbol{X}, \boldsymbol{Z})}_{①} - \underbrace{\int_{\boldsymbol{Z}} q(\boldsymbol{Z}) \cdot \ln q(\boldsymbol{Z}) d\boldsymbol{Z}}_{②} \\
① &= \int_{z_1 \ldots z_M} \prod_{i=1}^{M} q_i(z_i) \ln p(\boldsymbol{X}, \boldsymbol{Z}) dz_1 \ldots dz_M \\
&= \int_{z_j} q_j(z_j) \left(\int_{\boldsymbol{Z} \setminus j} \prod_{i \neq j}^{M} q_i(z_i) \ln p(\boldsymbol{X}, \boldsymbol{Z}) d\boldsymbol{Z} \setminus j \right) dz_j \\
&= \int_{z_j} q_j(z_j) \cdot \underbrace{\mathbb{E}_{\prod_{i \neq j}^{M} q_i(\boldsymbol{Z}_i)} [\ln p(\boldsymbol{X}, \boldsymbol{Z})]}_{\text{近似作 } \ln \tilde{p}(\boldsymbol{X}, z_j)} dz_j \\
② &= \int_{z_1 \ldots z_M} \prod_{i=1}^{M} q_i(z_i) \sum_{i=1}^{M} \ln q_i(z_i) dz_1 \ldots dz_M \\
&= \sum_{i=1}^{M} \int_{z_i} q_i(z_i) \ln q_i(z_i) dz_i \\
&= \int_{z_j} q_j(z_j) \ln q_j(z_j) dz_j + \text{常数} \ldots \text{(保持 } \{q_{i \neq j}\} \text{固定)} \\
\zeta(q) = ① - ② &= \int q_j \ln \tilde{p}(\boldsymbol{X}, \boldsymbol{Z}_j) d\boldsymbol{Z}_j - \int q_j \ln q_j d\boldsymbol{Z}_j + \text{常数}
\end{aligned} \tag{13.6}$$

其中我们定义了一个新的概率分布 $\tilde{p}(\boldsymbol{X}, \boldsymbol{Z}_j)$，形式为

$$\ln \tilde{p}(\boldsymbol{X}, \boldsymbol{Z}_j) = \mathbb{E}_{i \neq j}[\ln p(\boldsymbol{X}, \boldsymbol{Z})] + 常数 \tag{13.7}$$

这里，记号 $\mathbb{E}_{i \neq j}[\ldots]$ 表示关于定义在所有 $z_i (i \neq j)$ 上的 q 概率分布的期望，即

$$\mathbb{E}_{i \neq j}[\ln p(\boldsymbol{X}, \boldsymbol{Z})] = \int \ln p(\boldsymbol{X}, \boldsymbol{Z}) \prod_{i \neq j} q_i d\boldsymbol{Z}_i \tag{13.8}$$

现在假设我们保持 $\{q_{i \neq j}\}$ 固定，关于概率分布 $q_j(\boldsymbol{Z}_j)$ 的所有可能的形式最大化公式 13.6 中的 $\zeta(q)$。这很容易做，因为我们看到公式 13.6 是 $q_j(\boldsymbol{Z}_j)$ 和 $\tilde{p}(\boldsymbol{X}, \boldsymbol{Z}_j)$ 之间的 Kullback-Leibler 散度的负值。因此，最大化 13.6 等价于最小化 Kullback-Leibler 散度，且最小值出现在 $q_j(\boldsymbol{Z}_j) = \tilde{p}(\boldsymbol{X}, \boldsymbol{Z}_j)$ 的位置。于是，我们得到了最优解 $q_j^*(\boldsymbol{Z}_j)$ 的一般的表达式，形式为

$$\ln q_j^*(\boldsymbol{Z}_j) = \mathbb{E}_{i \neq j}[\ln p(\boldsymbol{X}, \boldsymbol{Z})] + 常数 \tag{13.9}$$

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 $i \neq j$。

公式中的可加性常数通过对概率分布 $q_j^*(\boldsymbol{Z}_j)$ 进行归一化的方式来设定。

分解近似的性质

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。我们的目标是在给定 x 的观测值的数据集 $D = \{x_1, \ldots, x_N\}$ 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

$$p(D|\mu, \tau) = \left(\frac{\tau}{2\pi}\right)^{\frac{N}{2}} \exp\left\{-\frac{\tau}{2}\sum_{n=1}^{N}(x_n - \mu)^2\right\} \tag{13.10}$$

我们现在引入 μ 和 τ 的共轭先验分布，形式为

$$p(\mu|\tau) = \mathcal{N}(\mu|\mu_0, (\lambda_0\tau)^{-1}) \tag{13.11}$$

$$p(\tau) = \mathrm{Gam}(\tau|a_0, b_0) \tag{13.12}$$

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

$$q(\mu, \tau) = q_\mu(\mu)q_\tau(\tau) \tag{13.13}$$

注意，真实的后验概率分布还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 13.9 中得到，如下所述。对于 $q_\mu(\mu)$，我们有

$$\begin{aligned}\ln q_\mu^*(\mu) &= \mathbb{E}_\tau[\ln p(D|\mu,\tau) + \ln p(\mu|\tau)] + \text{常数} \\ &= -\frac{\mathbb{E}[\tau]}{2}\left\{\lambda_0(\mu-\mu_0)^2 + \sum_{n=1}^{N}(x_n-\mu)^2\right\} + \text{常数}\end{aligned} \tag{13.14}$$

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

$$\mu_N = \frac{\lambda_0\mu_0 + N\bar{x}}{\lambda_0 + N} \tag{13.15}$$

$$\lambda_N = (\lambda_0 + N)\mathbb{E}[\tau] \tag{13.16}$$

类似地，因子 $q_\tau(\tau)$ 的最优解为

$$\begin{aligned}\ln q_\tau^*(\tau) &= \mathbb{E}_\mu[\ln p(D|\mu,\tau) + \ln p(\mu|\tau)] + \ln p(\tau) + \text{常数} \\ &= (a_0-1)\ln\tau - b_0\tau + \frac{N+1}{2}\ln\tau \\ &\quad - \frac{\tau}{2}\mathbb{E}_\mu\left[\sum_{n=1}^{N}(x_n-\mu)^2 + \lambda_0(\mu-\mu_0)^2\right] + \text{常数}\end{aligned} \tag{13.17}$$

因此 $q_\tau(\tau)$ 是一个 Gamma 分布 $\text{Gam}(\tau|a_N, b_N)$，参数为

$$a_N = a_0 + \frac{N+1}{2} \tag{13.18}$$

$$b_N = b_0 + \frac{1}{2}\mathbb{E}_\mu\left[\sum_{n=1}^{N}(x_n-\mu)^2 + \lambda_0(\mu-\mu_0)^2\right] \tag{13.19}$$

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 $\mathbb{E}[\tau]$ 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。

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 $\mathbb{E}[\tau] = \frac{a_N}{b_N}$。我们有

$$\frac{1}{\mathbb{E}[\tau]} = \frac{b_N}{a_N} = \mathbb{E}\left[\frac{1}{N+1}\sum_{n=1}^{N}(x_n-\mu)^2\right] = \frac{N}{N+1}(\bar{x^2} - 2\bar{x}\mathbb{E}[\mu] + \mathbb{E}[\mu^2]) \tag{13.20}$$

之后，使用公式 13.15 和公式 13.16，我们得到了 $q_\mu(\mu)$ 的一阶矩和二阶矩，形式为

$$\mathbb{E}[\mu] = \mathbb{E}\left[\frac{\lambda_0\mu_0 + N\bar{x}}{\lambda_0 + N}\right] = \bar{x} \tag{13.21}$$

$$\mathbb{E}[\mu^2] = \lambda_0 + \mathbb{E}[\mu]^2 = \bar{x}^2 + \frac{1}{N\mathbb{E}[\tau]} \tag{13.22}$$

现在，我们可以将这些矩代入公式中，然后解出 $\mathbb{E}[\tau]$，可得

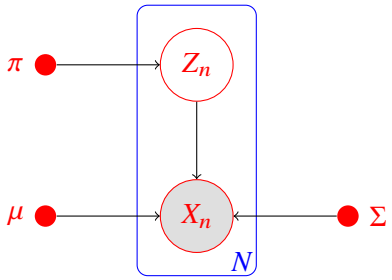
$$\frac{1}{\mathbb{E}[\tau]} = (\bar{x^2} - \bar{x}^2) = \frac{1}{N}\sum_{n=1}^{N}(x_n - \bar{x})^2 \tag{13.23}$$

模型比较

13.2 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。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。高斯混合模型如图所示



对于每个观测 $\boldsymbol{x}_n$，我们有一个对应的潜在变量 $\boldsymbol{z}_n$，它是一个“1-of-K”的二值向量，元素为 z_{nk}，其中 $k = 1, \ldots, K$。与之前一样，我们将观测数据集记作 $\boldsymbol{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$，类似地，我们瘦潜在变量记作 $\boldsymbol{Z} = \{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_N\}$。给定混合系数 $\boldsymbol{\pi}$，我们可以写出 $\boldsymbol{Z}$ 的条件概率分布，形式为

$$p(\boldsymbol{Z}|\boldsymbol{\pi}) = \prod_{n=1}^{N}\prod_{k=1}^{K}\pi_k^{z_{nk}} \tag{13.24}$$

类似地，给定潜在变量和分量参数，我们可以写出观测数据向量的条件概率分布，形式为

$$p(\boldsymbol{X}|\boldsymbol{Z}, \boldsymbol{\mu}, \boldsymbol{\Lambda}) = \prod_{n=1}^{N}\prod_{k=1}^{K}\mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k^{-1})^{z_{nk}} \tag{13.25}$$

其中 $\boldsymbol{\mu} = \{\boldsymbol{\mu}_k\}$ 且 $\boldsymbol{\Lambda} = \{\boldsymbol{\Lambda}_k\}$。注意，我们计算时使用的是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。

接下来，我们引入参数 $\boldsymbol{\mu}, \boldsymbol{\Lambda}$ 和 $\boldsymbol{\pi}$ 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 $\boldsymbol{\pi}$ 上的狄利克雷分布。

$$p(\boldsymbol{\pi}) = \mathrm{Dir}(\boldsymbol{\pi}|\boldsymbol{\alpha}_0) = C(\boldsymbol{\alpha}_0)\prod_{k=1}^{K}\pi_k^{\alpha_0-1} \tag{13.26}$$

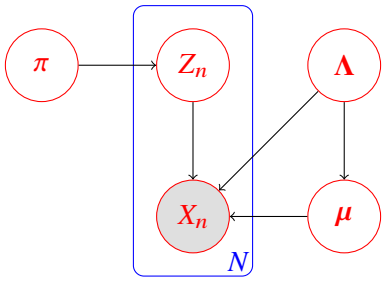
其中，根据对称性，我们为每个分量选择了同样的参数 α_0，$C(\boldsymbol{\alpha}_0)$ 是狄利克雷分布的归一化常数。参数 α_0 可以看成与混合分布的每个分量关联的观测的有效先验数量。如果 α_0 的值很小，那么后验概率分布会主要被数据集影响，而受到先验概率的影响很小。

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

$$\begin{aligned} p(\boldsymbol{\mu}, \boldsymbol{\Lambda}) &= p(\boldsymbol{\mu}|\boldsymbol{\Lambda})p(\boldsymbol{\Lambda}) \\ &= \prod_{k=1}^{K}\mathcal{N}(\boldsymbol{\mu}_k|\boldsymbol{m}_0, (\beta_0\boldsymbol{\Lambda}_k)^{-1})\mathcal{W}(\boldsymbol{\Lambda}_k|\boldsymbol{W}_0, v_0) \end{aligned} \tag{13.27}$$

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 $\boldsymbol{m}_0 = 0$。

生成的模型可以表示为下图



注意，从 $\boldsymbol{\Lambda}$ 到 $\boldsymbol{\mu}$ 之间存在一个链接，这是由于 $\boldsymbol{\mu}$ 上的概率分布的方差为 $\boldsymbol{\Lambda}$ 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 $\boldsymbol{z}_n$ 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 $\boldsymbol{\mu}$ 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。

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

$$p(\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}) = p(\boldsymbol{X}|\boldsymbol{Z}, \boldsymbol{\mu}, \boldsymbol{\Lambda})p(\boldsymbol{Z}|\boldsymbol{\pi})p(\boldsymbol{\pi})p(\boldsymbol{\mu}|\boldsymbol{\Lambda})p(\boldsymbol{\Lambda}) \tag{13.28}$$

不难验证这种分解方式对应于上图给出的概率图模型。注意，只有变量 $\boldsymbol{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ 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

$$q(\boldsymbol{Z}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}) = q(\boldsymbol{Z})q(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}) \tag{13.29}$$

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，这是我们需要做出的唯一假设。特别地，因子 $q(\boldsymbol{Z})$ 和 $q(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda})$ 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，我们依赖参数来区分不同的分布。

通过使用一般的结果，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 $q(\boldsymbol{Z})$ 的更新方程的推导。最优因子的对数为

$$\ln q^*(\boldsymbol{Z}) = \mathbb{E}_{\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\Lambda}}[\ln p(\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda})] + \text{常数} \tag{13.30}$$

我们现在使用公式 13.29 给出的分解方式。注意，我们只对等式右侧与变量 $\boldsymbol{Z}$ 相关的函数关系感兴趣。因此，任何与变量 $\boldsymbol{Z}$ 无关的项都可以被整合到可加的归一化系数中，从而有

$$\ln q^*(\boldsymbol{Z}) = \mathbb{E}_{\boldsymbol{\pi}}[\ln p(\boldsymbol{Z}|\boldsymbol{\pi})] + \mathbb{E}_{\boldsymbol{\mu},\boldsymbol{\Lambda}}[\ln p(\boldsymbol{X}|\boldsymbol{Z}, \boldsymbol{\mu}, \boldsymbol{\Lambda})] + \text{常数} \tag{13.31}$$

替换右侧的两个条件分布，然后再次把与 $\boldsymbol{Z}$ 无关的项整合到可加性常数中

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{\pi}}[\ln p(\boldsymbol{Z}|\boldsymbol{\pi})] + \mathbb{E}_{\boldsymbol{\mu},\boldsymbol{\Lambda}}[\ln p(\boldsymbol{X}|\boldsymbol{Z}, \boldsymbol{\mu}, \boldsymbol{\Lambda})] \\
=&\mathbb{E}_{\boldsymbol{\pi}}\left[\ln\left(\prod_{n=1}^{N}\prod_{k=1}^{K}\pi_k^{z_{nk}}\right)\right] + \mathbb{E}_{\boldsymbol{\mu},\boldsymbol{\Lambda}}\left[\ln\left(\prod_{n=1}^{N}\prod_{k=1}^{K}\mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k^{-1})^{z_{nk}}\right)\right] \\
=&\sum_{n=1}^{N}\sum_{k=1}^{K} z_{nk}\left\{\underbrace{\mathbb{E}[\ln \pi_k] + \frac{1}{2}\mathbb{E}[\ln|\boldsymbol{\Lambda}_k|] - \frac{D}{2}\ln(2\pi) - \frac{1}{2}\mathbb{E}_{\boldsymbol{\mu}_k,\boldsymbol{\Lambda}_k}[(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^T\boldsymbol{\Lambda}_k(\boldsymbol{x}_n - \boldsymbol{\mu}_k)]}_{\text{记作}\ln\rho_{nk}}\right\} \\
&+ \text{常数}
\end{aligned} \tag{13.32}$$

其中 D 是数据变量 $\boldsymbol{x}$ 的维度。所以，我们有

$$\ln q^*(\boldsymbol{Z}) = \sum_{n=1}^{N}\sum_{k=1}^{K} z_{nk}\ln\rho_{nk} + \text{常数} \tag{13.33}$$

公式两侧取指数，我们有

$$q^*(\boldsymbol{Z}) \propto \prod_{n=1}^{N}\prod_{k=1}^{K}\rho_{nk}^{z_{nk}} \tag{13.34}$$

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

$$q^*(\boldsymbol{Z}) = \prod_{n=1}^{N}\prod_{k=1}^{K} r_{nk}^{z_{nk}} \tag{13.35}$$

其中

$$r_{nk} = \frac{\rho_{nk}}{\sum_{j=1}^{K}\rho_{nk}} \tag{13.36}$$

从中我们看到 r_{nk} 扮演着“责任”的角色。注意，$q^*(\boldsymbol{Z})$ 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

$$N_k = \sum_{n=1}^{N} r_{nk} \tag{13.37}$$

$$\bar{\boldsymbol{x}}_k = \frac{1}{N_k} \sum_{n=1}^{N} r_{nk} \boldsymbol{x}_n \tag{13.38}$$

$$\boldsymbol{S}_k = \frac{1}{N_k} \sum_{n=1}^{N} r_{nk} (\boldsymbol{x}_n - \bar{\boldsymbol{x}}_k)(\boldsymbol{x}_n - \bar{\boldsymbol{x}}_k)^T \tag{13.39}$$

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，使用公式给出的一般结果，我们有

$$\begin{aligned}
\ln q^*(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}) &= \mathbb{E}_{\boldsymbol{Z}}[\ln p(\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda})] + \text{常数} \\
&= \mathbb{E}_{\boldsymbol{Z}}[\ln p(\boldsymbol{X}|\boldsymbol{Z}, \boldsymbol{\mu}, \boldsymbol{\Lambda}) + \ln p(\boldsymbol{Z}|\boldsymbol{\pi}) + \ln p(\boldsymbol{\pi}) + \ln p(\boldsymbol{\mu}, \boldsymbol{\Lambda})] + \text{常数} \\
&= \mathbb{E}_{\boldsymbol{Z}}[\ln p(\boldsymbol{X}|\boldsymbol{Z}, \boldsymbol{\mu}, \boldsymbol{\Lambda})] + \mathbb{E}_{\boldsymbol{Z}}[\ln p(\boldsymbol{Z}|\boldsymbol{\pi})] + \mathbb{E}_{\boldsymbol{Z}}[\ln p(\boldsymbol{\pi})] + \mathbb{E}_{\boldsymbol{Z}}[\ln p(\boldsymbol{\mu}, \boldsymbol{\Lambda})] + \text{常数} \\
&= \mathbb{E}_{\boldsymbol{Z}}\left[\ln \left(\prod_{n=1}^{N}\prod_{k=1}^{K} \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k^{-1})^{z_{nk}}\right)\right] \\
&\quad + \mathbb{E}_{\boldsymbol{Z}}\left[\ln \left(p(\boldsymbol{Z}|\boldsymbol{\pi})\right)\right] \\
&\quad + \mathbb{E}_{\boldsymbol{Z}}\left[\ln \left(p(\boldsymbol{\pi})\right)\right] \\
&\quad + \mathbb{E}_{\boldsymbol{Z}}\left[\ln \left(\prod_{k=1}^{K} p(\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k)\right)\right] + \text{常数} \\
&= \ln p(\boldsymbol{\pi}) + \sum_{k=1}^{K} \ln p(\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k) + \mathbb{E}_{\boldsymbol{Z}}\left[\ln \left(p(\boldsymbol{Z}|\boldsymbol{\pi})\right)\right] \\
&\quad + \sum_{k=1}^{K}\sum_{n=1}^{N} \mathbb{E}[z_{nk}] \ln \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k^{-1}) + \text{常数}
\end{aligned} \tag{13.40}$$

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 $\boldsymbol{\Lambda}$ 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

$$q(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}) = q(\boldsymbol{\pi}) \prod_{k=1}^{K} q(\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k) \tag{13.41}$$

分离出公式 13.40 右侧的与 $\boldsymbol{\pi}$ 相关的项，我们有

$$\begin{aligned}\ln q^*(\boldsymbol{\pi}) &= \mathbb{E}_{\boldsymbol{Z}}\left[\ln\left(p(\boldsymbol{Z}|\boldsymbol{\pi})\right)\right] + \ln p(\boldsymbol{\pi}) \\ &= \ln\left(C(\boldsymbol{\alpha}_0)\prod_{k=1}^{K}\pi_k^{\alpha_0-1}\right) + \mathbb{E}_{\boldsymbol{Z}}\left[\ln\left(\prod_{n=1}^{N}\prod_{k=1}^{K}\pi_k^{z_{nk}}\right)\right] \\ &= (\alpha_0-1)\sum_{k=1}^{K}\ln\pi_k + \sum_{k=1}^{K}\sum_{n=1}^{N} r_{nk}\ln\pi_k + \text{常数}\end{aligned} \tag{13.42}$$

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，但是我们总可以使用概率的乘积规则，将其写成 $q^*(\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k) = q^*(\boldsymbol{\mu}_k|\boldsymbol{\Lambda}_k)q^*(\boldsymbol{\Lambda}_k)$。结果是一个高斯-Wishart 分布，形式为

$$q^*(\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k) = \mathcal{N}(\boldsymbol{\mu}_k|\boldsymbol{m}_k, (\beta_k\boldsymbol{\Lambda}_k)^{-1})\mathcal{W}(\boldsymbol{\Lambda}_k|\boldsymbol{W}_k, \nu_k) \tag{13.43}$$

其中我们已经定义了

$$\beta_k = \beta_0 + N_k \tag{13.44}$$

$$\boldsymbol{m}_k = \frac{1}{\beta_k}(\beta_0\boldsymbol{m}_0 + N_k\bar{\boldsymbol{x}}_k) \tag{13.45}$$

$$\boldsymbol{W}_k^{-1} = \boldsymbol{W}_0^{-1} + N_k\boldsymbol{S}_k + \frac{\beta_0 N_k}{\beta_0 + N_k}(\bar{\boldsymbol{x}}_k - \boldsymbol{m}_0)(\bar{\boldsymbol{x}}_k - \boldsymbol{m}_0)^T \tag{13.46}$$

$$\nu_k = \nu_0 + N_k \tag{13.47}$$

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的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 13.32 中定义的 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

$$\rho_{nk} = \mathbb{E}[\ln\pi_k] + \frac{1}{2}\mathbb{E}[\ln|\boldsymbol{\Lambda}_k|] - \frac{D}{2}\ln(2\pi) - \frac{1}{2}\mathbb{E}_{\boldsymbol{\mu}_k,\boldsymbol{\Lambda}_k}[(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^T\boldsymbol{\Lambda}_k(\boldsymbol{x}_n - \boldsymbol{\mu}_k)] \tag{13.48}$$

其中

$$\mathbb{E}_{\boldsymbol{\mu}_k,\boldsymbol{\Lambda}_k}[(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^T\boldsymbol{\Lambda}_k(\boldsymbol{x}_n - \boldsymbol{\mu}_k)] = D\beta_k^{-1} + \nu_k(\boldsymbol{x}_n - \boldsymbol{m}_k)^T\boldsymbol{\Lambda}_k(\boldsymbol{x}_n - \boldsymbol{m}_k) \tag{13.49}$$

$$\ln\tilde{\Lambda}_k = \sum_{i=1}^{D}\psi\left(\frac{\nu_k + 1 - i}{2}\right) + D\ln 2 + \ln|\boldsymbol{W}_k| \tag{13.50}$$

$$\ln\tilde{\pi}_k = \mathbb{E}[\ln\pi_k] = \psi(\alpha_k) - \psi(\hat{\alpha}) \tag{13.51}$$

其中我们引入了 $\tilde{\Lambda}_k$ 和 $\tilde{\pi}_k$ 的定义，$\psi(\cdot)$ 是 Digamma 函数，$\hat{\alpha} = \sum_k \alpha_k$。公式 13.50 和公式 13.51 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 13.49、13.50 和 13.51 代入公式 13.32，然后使用公式 13.36，我们得

到了下面的“责任”的结果

$$r_{nk} \propto \tilde{\pi}_k \tilde{\Lambda}_k^{\frac{1}{2}} \exp\left\{-\frac{D}{2\beta_k} - \frac{\nu_k}{2}(\boldsymbol{x}_n - \boldsymbol{m}_k)^T \boldsymbol{W}_k(\boldsymbol{x}_n - \boldsymbol{m}_k)\right\} \tag{13.52}$$

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。

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 13.49、13.50 和 13.51 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 13.42 和 13.43 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 13.28 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。

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 $N \to \infty$ 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，如果我们简单地引入一个先验分布，煞有然后使用 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。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互验证的技术。

变分下界

下界提供了另一种推导变分重估计方程的方法。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 $\boldsymbol{Z}$ 的离散分布、π 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。

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 $\hat{\boldsymbol{x}}$ 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 $\hat{\boldsymbol{z}}$，从而预测概率分布为

$$p(\hat{\boldsymbol{x}}|\boldsymbol{X}) = \sum_{\hat{\boldsymbol{z}}} \iiint p(\hat{\boldsymbol{x}}|\hat{\boldsymbol{z}}, \boldsymbol{\mu}, \boldsymbol{\Lambda}) p(\hat{\boldsymbol{z}}|\boldsymbol{\pi}) p(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}|\boldsymbol{X}) d\boldsymbol{\pi} d\boldsymbol{\mu} d\boldsymbol{\Lambda} \tag{13.53}$$

其中 $p(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}|\boldsymbol{X})$ 是参数的 (未知) 真实后验概率分布。使用公式 13.24 和公式 13.25，我们可以首先完成在 $\hat{\boldsymbol{z}}$ 上的求和，得到

$$p(\hat{\boldsymbol{x}}|\boldsymbol{X}) = \sum_{k=1}^{K} \iiint \pi_k \mathcal{N}(\hat{\boldsymbol{x}}|\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k^{-1}) p(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}|\boldsymbol{X}) d\boldsymbol{\pi} d\boldsymbol{\mu} d\boldsymbol{\Lambda} \tag{13.54}$$

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 $p(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}|\boldsymbol{X})$ 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

$$p(\hat{\boldsymbol{x}}|\boldsymbol{X}) \simeq \sum_{k=1}^{K} \iiint \pi_k \mathcal{N}(\hat{\boldsymbol{x}}|\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k^{-1}) q(\boldsymbol{\pi}) q(\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k) d\boldsymbol{\pi} d\boldsymbol{\mu}_k d\boldsymbol{\Lambda}_k \tag{13.55}$$

其中我们使用了公式 13.41 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

$$p(\hat{\boldsymbol{x}}|\boldsymbol{X}) \simeq \frac{1}{\hat{\alpha}} \sum_{k=1}^{K} \alpha_k \mathrm{St}(\hat{\boldsymbol{x}}|\boldsymbol{m}_k, \boldsymbol{L}_k, \nu_k + 1 - D) \tag{13.56}$$

其中第 k 个分量的均值为 $\boldsymbol{m}_k$，精度为

$$\boldsymbol{L}_k = \frac{(\nu_k + 1 - D)\beta_k}{1 + \beta_k} \boldsymbol{W}_k \tag{13.57}$$

当数据集的大小 N 很大时，预测分布就变成了高斯混合。

确定分量的数量

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。

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。如果真实的后验概率分布是多峰的那么基于最小化 $\mathrm{KL}(q\|p)$ 的变分推断会倾向于在某一个峰值的领域内近似这个分布，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，如果我们想比较不同的 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 $\ln K!$。

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。

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 13.29 给定。然而，不同因子的最优解给出了额外的分解。特别地，$q^*(\boldsymbol{\mu}, \boldsymbol{\Lambda})$ 的最优解由每个混合分量 k 上的独立分布 $q^*(\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k)$ 的乘积给定，而公式 13.34 给定的潜在变量上的变分后验概率分布 $q^*(\boldsymbol{Z})$ 可以分解为每个观测 n 的独立概率分布 $q^*(\boldsymbol{z}_n)$(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。

我们会把这些额外的分解方式称为诱导分解 (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。

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。

13.3 变分线性回归

变分分布

预测分布

下界

13.4 指数族分布

变分信息传递

13.5 局部变分方法

13.6 变分 logistic 回归

变分后验概率分布

最优化变分参数

超参数的推断

13.7 期望传播

例子:聚类问题

图的期望传播

第 14 章 采样方法

对于大多数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。上一章，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结合。因此，在连续变量的情形下，我们希望计算下面的期望

$$\mathbb{E}[f] = \int f(z)p(zdz) \tag{14.1}$$

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 $l = 1, \ldots, L$。这使得期望可以通过有限和的方式计算，即

$$\hat{f} = \frac{1}{L}\sum_{l=1}^{L} f(z^{(l)}) \tag{14.2}$$

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

$$\mathrm{var}[\hat{f}] = \frac{1}{L}\mathbb{E}[(f - \mathbb{E}[f])^2] \tag{14.3}$$

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。

补充一个逻辑关系图

14.1 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。

标准概率分布

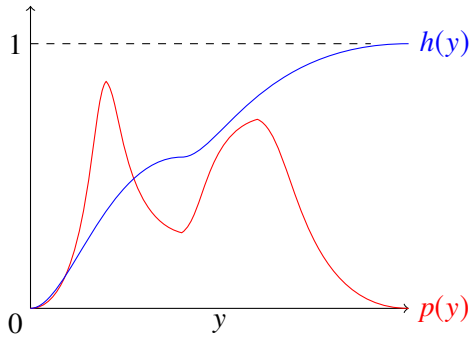
首先，我们考虑如何从简单的非均匀分布中生成随机数，假定我们已经有了一个均匀分布的随机数的来源。假设 z 在区间 $(0, 1)$ 上均匀分布，我们使用某个函数 $f(\cdot)$ 对 z 的值进行变换，即 $y = f(z)$。y 上的概率分布为

$$p(y) = p(z)\left|\frac{\mathrm{d}z}{\mathrm{d}y}\right| \tag{14.4}$$

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 14.4 进行积分，我们有

$$z = h(y) \equiv \int_{-\infty}^{y} p(\hat{y})d\hat{y} \tag{14.5}$$

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，如图所示



考虑指数分布 (exponential distribution)

$$p(y) = \lambda \exp(-\lambda y) \tag{14.6}$$

其中 $0 \leqslant y \leqslant \infty$。在这种情况下

$$h(y) = \int_{0}^{y} f(y)dy = 1 - \exp(-\lambda y) \tag{14.7}$$

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。

$$\begin{aligned} z &= h(y) = 1 - \exp(-\lambda y) \\ y &= h^{-1}(z) \\ &\Rightarrow \exp(-\lambda y) = 1 - z \\ &\Rightarrow -\lambda y = \ln(1-z) \\ &\Rightarrow y = -\frac{1}{\lambda}\ln(1-z) \end{aligned} \tag{14.8}$$

另一种可以应用变换方法的概率分布是柯西分布

$$p(y) = \frac{1}{\pi}\frac{1}{1+y^2} \tag{14.9}$$

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

$$p(y_1, \ldots, y_M) = p(z_1, \ldots, z_M)\left|\frac{\partial(z_1, \ldots, z_M)}{\partial(y_1, \ldots, y_M)}\right| \tag{14.10}$$

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 $z \to 2z - 1$ 的方式进行变换。接下来，我们丢弃那些不满足 $z_1^2 + z_2^2 \leqslant 1$ 的点对。这产生出单位圆内部的一个均匀分布，且 $p(z_1, z_2) = \frac{1}{\pi}$。然后，对于每对 z_1, z_2，我们计算

$$y_1 = z_1 \left(\frac{-2 \ln r^2}{r^2} \right)^{\frac{1}{2}} \tag{14.11}$$

$$y_2 = z_2 \left(\frac{-2 \ln r^2}{r^2} \right)^{\frac{1}{2}} \tag{14.12}$$

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

$$\begin{aligned} p(y_1, y_2) &= p(z_1, z_2) \left| \frac{\partial(z_1, z_2)}{\partial(y_1, y_2)} \right| \\ &= \left[\frac{1}{\sqrt{2\pi}} \exp\left(\frac{-y_1^2}{2} \right) \right] \left[\frac{1}{\sqrt{2\pi}} \exp\left(\frac{-y_1^2}{2} \right) \right] \end{aligned} \tag{14.13}$$

因此 y_1 和 y_2 是独立的，有每个都服从高斯分布，均值为零，方差为 1。

推导过程：(Box-Muller 变换原理)

1. 目标

$$\begin{aligned} p(y_1, y_2) &= \left[\frac{1}{\sqrt{2\pi}} \exp\left(\frac{-y_1^2}{2} \right) \right] \left[\frac{1}{\sqrt{2\pi}} \exp\left(\frac{-y_1^2}{2} \right) \right] \\ &= \frac{1}{2\pi} \exp\left(-\frac{y_1^2 + y_2^2}{2} \right) \end{aligned} \tag{14.14}$$

做极坐标变换，则 $y_1 = R\cos\theta, y_2 = R\sin\theta$，则有

$$\frac{1}{2\pi} \exp\left(-\frac{y_1^2 + y_2^2}{2} \right) = \frac{1}{2\pi} \exp\left(-\frac{R^2}{2} \right) \tag{14.15}$$

可以看到这个结果可以看成是两个概率分布的密度函数的乘积，其中一个可以看成是 $[0, 2\pi]$ 上均匀分布，将其转换为标准均匀分布则有 $\theta \sim U(0, 2\pi) = 2\pi z_1$，因为 $(z_1, z_2 \sim U(0, 1))$。

2. 求 $h(y_1, y_2)$

$$\begin{aligned} z &= P(R \leqslant r) \\ &= \int_0^{2\pi} d\theta \int_0^r \frac{1}{2\pi} \exp\left(-\frac{\rho^2}{2} \right) \rho d\rho \\ &= -\exp\left(\frac{-\rho^2}{2} \right) \Big|_0^r \\ &= -\exp\left(-\frac{r^2}{2} \right) + 1 \end{aligned} \tag{14.16}$$

其中 $r^2 = z_1^2 + z_2^2$。

3. 求 $y1, y2$

$$\begin{aligned} R &= h^{-1}(z) \\ &= \sqrt{-2\ln(1-z)} \end{aligned} \tag{14.17}$$

这里是二维联合分布，因此

$$\begin{aligned} y_1 &= R\cos\theta = \frac{z_1}{r}\sqrt{-2\ln(r^2)} \\ y_2 &= R\sin\theta = \frac{z_2}{r}\sqrt{-2\ln(r^2)} \end{aligned} \tag{14.18}$$

显然，变换方法依赖于它能够进行计算所需的概率分布，并且能够求所需的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，但是这些方法确定是更一般的方法的重要成分。

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。

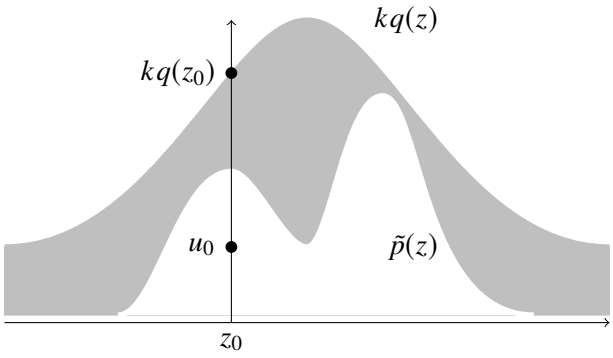
假设我们希望从 $p(z)$ 中采样，这个概率分布不是我们目前为止讨论过的简单的标准的概率分布中的一个，从而直接从 $p(z)$ 中采样是很困难的。此外，我们假设我们能够很容易地计算对于任意给定的 z 值的 $p(z)$(不考虑归一化常数 Z)，即

$$p(z) = \frac{1}{Z_p}\tilde{p}(z) \tag{14.19}$$

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，它的值的选择满足下面的性质：对所有的 z 值，都有 $kq(z) \geqslant \tilde{p}(z)$。函数 $kq(z)$ 被称为比较函数。

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 $[0, kq(z)]$ 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

$$\begin{aligned} p(\text{接受}) &= \int \left\{ \frac{\tilde{p}(z)}{kq(z)} \right\} q(z)dz \\ &= \frac{1}{k} \int \tilde{p}(z)dz \end{aligned} \tag{14.20}$$

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 $\tilde{p}(z)$。

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。

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。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。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

$$q(z) = k_i \lambda_i \exp\{-\lambda_i(z - z_i)\} \quad \hat{z}_{i-1,i} < z \leqslant \hat{z}_{i,i+1} \tag{14.21}$$

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。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用 Metropolis-Hasting 阶梯函数即可，这就产生了可调节拒绝 Metropolis 采样 (adaptive rejection Metropolis sampling) 方法。

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。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。

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 14.1 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 14.2 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

$$\mathbb{E}[f] \simeq \sum_{l=1}^{L} p(z^{(l)}) f(z^{(l)}) \tag{14.22}$$

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，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。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

$$\begin{aligned} \mathbb{E}[f] &= \int f(z)p(z)dz \\ &= \int f(z)\frac{p(z)}{q(z)}q(z)dz \\ &\simeq \frac{1}{L}\sum_{l=1}^{L}\frac{p(z^{(l)})}{q(z^{(l)})}f(z^{(l)}) \end{aligned} \tag{14.23}$$

$r_l = \frac{p(z^{(l)})}{q(z^{(l)})}$ 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 $p(z) = \frac{\tilde{p}(z)}{Z_p}$，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 $q(z) = \frac{\tilde{q}(z)}{Z_q}$，它

具有相同的性质。于是我们有

$$\begin{aligned}\mathbb{E}[f] &= \int f(z)p(z)dz \\ &= \frac{Z_q}{Z_p}\int f(z)\frac{\tilde{p}(z)}{\tilde{q}(z)}q(z)dz \\ &\simeq \frac{Z_q}{Z_p}\frac{1}{L}\sum_{l=1}^{L}\tilde{r}_l f(z^{(l)})\end{aligned} \tag{14.24}$$

其中 $\tilde{r}_l = \frac{\tilde{p}(z)}{\tilde{q}(z)}$。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

$$\begin{aligned}\frac{Z_p}{Z_q} &= \int \frac{\tilde{p}(z)}{\tilde{q}(z)}q(z)dz \\ &\simeq \frac{1}{L}\sum_{l=1}^{L}\tilde{r}_l\end{aligned} \tag{14.25}$$

因此

$$\mathbb{E}[f] \simeq \sum_{l=1}^{L} w_l f(z^{(l)}) \tag{14.26}$$

其中我们定义

$$w_l = \frac{\tilde{r}_l}{\sum_m \tilde{r}_m} \tag{14.27}$$

与拒绝采样的情形相同中，重要采样方法的成功严重依赖于采样分布 $q(z)$ 与所求的概率分布 $p(z)$ 的匹配程度。经常出现的情形是 $p(z)$ 变化剧烈，并且大部分的质量集中于 z 空间的一个相对较小的区域中，此时重要权重 $\{r_l\}$ 由几个具有圈套值的权值控制，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 $p(z)f(z)$ 圈套的区域中，那么问题会更加严重。此时，r_l 和 $r_l f(z^{(l)})$ 的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。这也强调采样分布 $q(z)$ 的一个关键的要求，即它不应该在 $p(z)$ 可能圈套的区域中取得较小的值或者为零的值。

采样-重要性-重采样

拒绝采样方法部分依赖于确定常数 k 的一个合适的值。对于许多对概率分布 $p(z)$ 和 $q(z)$ 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 值才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 $q(z)$，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 $z^{(1)},\ldots,z^{(L)}$ 从 $q(z)$ 中抽取。然后在第二个阶段，构建权值 $w_1,\ldots,w_L$。最后，L 个样本的第二个集合从离散概率分布 $(z^{(1)},\ldots,z^{(L)})$ 中抽取，概率由权值 $(w_1,\ldots,w_L)$ 给定。

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

$$Q(\theta, \theta^{\text{旧}}) = \int p(Z|X, \theta^{\text{旧}}) \ln p(Z, X|\theta) dZ \tag{14.28}$$

我们可以使用采样的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

$$Q(\theta, \theta^{\text{旧}}) \simeq \frac{1}{L} \sum_{l=1}^{L} \ln p(Z^{(l)}, X|\theta) \tag{14.29}$$

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)。

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 $p(\boldsymbol{\theta}, \boldsymbol{Z}|\boldsymbol{X})$ 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 $p(\boldsymbol{\theta}|\boldsymbol{Z}, \boldsymbol{X})$ 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)。

1. I 步骤。我们希望从概率分布 $p(\boldsymbol{Z}|\boldsymbol{X})$ 采样，但是我们不能直接进行。于是，我们注意到下面的关系

$$p(\boldsymbol{Z}|\boldsymbol{X}) = \int p(\boldsymbol{Z}|\boldsymbol{\theta}, \boldsymbol{X}) p(\boldsymbol{\theta}|\boldsymbol{X}) d\boldsymbol{X} \tag{14.30}$$

因此对于 $l = 1, \ldots, L$，我们首先从当前对 $p(\boldsymbol{\theta}|\boldsymbol{X})$ 的估计中抽取样本 $\boldsymbol{\theta}^{(l)}$，然后使用这个样本从 $p(\boldsymbol{Z}|\boldsymbol{\theta}^{(l)}, \boldsymbol{X})$ 中抽取样本 $\boldsymbol{Z}^{(l)}$。

2. P 步骤。给定关系

$$p(\boldsymbol{\theta}|\boldsymbol{X}) = \int p(\boldsymbol{\theta}|\boldsymbol{Z}, \boldsymbol{X}) p(\boldsymbol{Z}|\boldsymbol{X}) d\boldsymbol{Z} \tag{14.31}$$

我们使用从 I 步骤中得到的样本 $\{\boldsymbol{Z}^{(l)}\}$，计算 $\boldsymbol{\theta}$ 上的后验概率分布的修正后的估计，结果为

$$p(\boldsymbol{\theta}|\boldsymbol{X}) \simeq \frac{1}{L} \sum_{l=1}^{L} p(\boldsymbol{\theta}|\boldsymbol{Z}^{(l)}, \boldsymbol{X}) \tag{14.32}$$

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 $\boldsymbol{\theta}$ 和隐含变量 $\boldsymbol{Z}$ 进行了区分。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。

14.2 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，被称为马尔可夫链蒙特卡罗 (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空间维度的增长。

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 $z^{(1)}, z^{(2)}, \ldots$，组成了一个马尔可夫链。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，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

$$A(z^*, z^{(\tau)}) = \min\left(1, \frac{\tilde{p}(z^*)}{\tilde{p}(z^{(\tau)})}\right) \tag{14.33}$$

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。

通过考察一个具体的例子，即简单的随机游走的例子，我们可以对马尔可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

$$p(z^{(\tau+1)} = z^{(\tau)}) = 0.5 \tag{14.34}$$

$$p(z^{(\tau+1)} = z^{(\tau)} + 1) = 0.25 \tag{14.35}$$

$$p(z^{(\tau+1)} = z^{(\tau)} - 1) = 0.25 \tag{14.36}$$

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 $\mathbb{E}[(z^{(\tau)})] = \frac{\tau}{2}$。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。设计马尔可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。

MCMC 方法是使用马尔可夫链的蒙特卡罗积分，其基本思想是：构造一条 Markov 链使其平稳分布为待估参数的后验分布，通过这条马尔可夫链产生后验分布的样本，并基于马尔可夫链达到平稳分布时的样本 (有效样本) 进行蒙特卡罗积分。设为某一空间 n 为产生的总样本数 m 为链条达到平稳时的样本数则 MCMC 方法的基本思路可概括为:

1. 构造 Markov 链。构造一条 Markov 链，使其收敛到平稳分布；
2. 产生样本。由其中的某一点出发，用 (1) 中的 Markov 链进行抽样模拟，产生点序列；
3. 蒙特卡罗积分。任一函数的期望估计为

在采用 MCMC 方法时马尔可夫链转移核的构造至关重要，不同的转移核构造方法将产生不同的 MCMC 方法，当前常用的 MCMC 方法主要有两种 Gibbs 抽样和 Metropo-Lis-Hastings 算法。

马尔可夫链

在详细讨论 MCMC 方法之前，仔细研究马尔可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可夫链会收敛到所求的概率分布上。

如果对于所有的时刻 m，转移概率都相同，那么这个马尔可夫链被称为同质的 (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

$$p(z^{(m+1)}) = \sum_{z^{(m)}} p(z^{(m+1)}|z^{(m)})p(z^{(m)}) \tag{14.37}$$

对于一个概率分布来说，如果马尔可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可夫链是不变的，或者静止的。因此，对于一个转移概率为 $T(z', z)$ 的同质的马尔可夫链来说，如果

$$p^*(z) = \sum_{z'} T(z', z)p^*(z') \tag{14.38}$$

那么概率分布 $p^*(z)$ 是不变的。注意，一个给定的马尔可夫链可能有多个不变的概率分布。例如，如果转换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

$$p^*(z)T(z, z') = p^*(z')T(z', z) \tag{14.39}$$

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

$$\sum_{z'} p^*(z')T(z', z) = \sum_{z'} p^*(z)T(z, z') = p^*(z)\sum_{z'} p(z'|z) = p^*(z) \tag{14.40}$$

满足细节平衡性质的马尔可夫链被称为可翻转的 (reversible)。

我们的目标是使用马尔可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可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 $m \to \infty$，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可夫链只能有唯一的一个均衡分布。可以证明，同质的马尔可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。

Metropolis-Hastings 算法

在算法的步骤 τ 中，当前状态为 $z^{(\tau)}$，我们从概率分布 $q_k(z^{(\tau)})$ 中抽取一个样本 z^*，然后以概率 $A_k(z^*, z^{(\tau)})$ 接受它，其中

$$A_k(z^*, z^{(\tau)}) = \min\left(1, \frac{\tilde{p}(z^*)q_k(z^{(\tau)}|z^*)}{\tilde{p}(z^{(\tau)})q_k(z^{(\tau)}|z^*)}\right) \tag{14.41}$$

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。

我们现在证明 $p(z)$ 对于由 Metropolis-Hastings 算法定义的马尔可夫链是一个不变的概率分布，方法是证明满足细节平衡。我们有

$$\begin{aligned} p(z)q_k(z^{'}|z)A_k(z^{'}, z) &= \min(p(z)q_k(z^{'}|z), p(z^{'})q_k(z|z^{'}) \\ &= p(z^{'})q_k(z|z^{'})\min(\frac{p(z)q_k(z^{'}|z)}{p(z^{'})q_k(z|z^{'})}, 1) \\ &= p(z^{'})q_k(z|z^{'})A_k(z, z^{'}) \end{aligned} \tag{14.42}$$

证明完毕。

14.3 吉布斯采样

吉布斯采样是一个简单的并且广泛应用的马尔可夫链蒙特卡罗算法，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 $p(z) = p(z_1, \ldots, z_M)$，并且假设我们已经选择了马尔可夫链的某个初始的状态。吉布斯采样的每个步骤及到将一个变量的值替换为以剩余变量的值为条件，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 $z_1, \ldots, z_M$ 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

$$p(z_1|z_2^{(\tau)}, z_3^{(\tau)}) \tag{14.43}$$

中采样得到的。接下来，我们将 z_2^{τ} 替换为新值 $z_2^{\tau+1}$，这个新值是从条件概率分布

$$p(z_2|z_1^{(\tau+1)}, z_3^{(\tau)}) \tag{14.44}$$

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

$$p(z_3|z_1^{(\tau+1)}, z_2^{(\tau+1)}) \tag{14.45}$$

中抽取的。以此类推，在这三个变量之间进行循环。

1. 初始化 $\{z_i : i = 1, \ldots, M\}$。
2. 对于 $\tau = 1, \ldots, T$：
   - 采样 $z_1^{(\tau+1)} \sim p(z_1|z_2^{(\tau)}, z_3^{(\tau)}, \ldots, z_M^{(\tau)})$。
   - 采样 $z_2^{(\tau+1)} \sim p(z_2|z_1^{(\tau+1)}, z_3^{(\tau)}, \ldots, z_M^{(\tau)})$。
   - ⋮
   - 采样 $z_j^{(\tau+1)} \sim p(z_j|z_1^{(\tau+1)}, \ldots, z_{j-1}^{(\tau+1)}, z_{j+1}^{(\tau)}, \ldots, z_M^{(\tau)})$。
   - ⋮
   - 采样 $z_M^{(\tau+1)} \sim p(z_M|z_1^{(\tau+1)}, z_2^{(\tau+1)}, \ldots, z_{M-1}^{(\tau+1)})$。

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，因此对于整个马尔可夫链来说也是不变的。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处处为零。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，马尔可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 $z_{\backslash k}$ 不变，并且对于这种情形来说，从 z 到 z^* 的转移概率为 $q_k(z^*|z) = p(z_k^*|z_{\backslash})$。我们注意到 $z_{\backslash k}^* = z_{\backslash k}$，因为在采样的步骤中，向量的各个元素都不改变。并且，$p(z) = p(z_k|z_{\backslash k})p(z_{\backslash k})$。因此，确定 Metropolis-Hastings 算法中的接受概率的因子为

$$\begin{aligned} A(z^*, z) = \frac{p(z^*)q_k(z|z^*)}{p(z)q_k(z^*|z)} &= \frac{p(z_k^*|z_{\backslash k}^*)p(z_{\backslash k}^*)p(z_k|z_{\backslash k}^*)}{p(z_k|z_{\backslash k})p(z_{\backslash k})p(z_k^*|z_{\backslash k})} \\ &= \frac{p(z_k^*|z_{\backslash k}^*)p(z_{\backslash k}^*)p(z_k|z_{\backslash k}^*)}{p(z_k|z_{\backslash k}^*)p(z_{\backslash k}^*)p(z_k^*|z_{\backslash k}^*)} \\ &= 1 \end{aligned} \tag{14.46}$$

推导时我们用到了 $z_{\backslash k}^* = z_{\backslash k}$。因此 Metropolis-Mastings 步骤问题被接受的。

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。

14.4 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，算法会很低效。切片采样 (slice sampling) 方法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。目标是从下面的概率分布

$$\hat{p}(z,u) = \begin{cases} \frac{1}{Z_p} & \text{如果} 0 \leqslant u \leqslant \tilde{p}(z) \\ 0 & \text{其他情况} \end{cases} \tag{14.47}$$

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

$$\int \hat{p}(z,u)du = \int_0^{\tilde{p}(z)} \frac{1}{Z_p} du = p(z) \tag{14.48}$$

因此，我们可能通过从 $\hat{p}(z,u)$ 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 $0 \leqslant u \leqslant \tilde{p}(z)$ 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。

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，这可以通过确保萍踪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 $z_{\min} \leqslant z \leqslant z_{\max}$ 来获得。根据概率分布的特征长度标度来对步长进行调节就发生在这里。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。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，直到端点位于区域外。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。

14.5 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。我们会讨论动态系统框架，然后，我们会解释这个框架如何与 Metroplolis 算法结合，产生出一个强大的混合蒙特卡罗算法。

动态系统

随机采样的动态方法起源于模拟哈密顿动力学下进行变化的物理系统的行为。在马尔可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动力学 (Hamiltonian dynamics) 的框架。

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。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

$$r_i = \frac{\mathrm{d}z_i}{\mathrm{d}\tau} \tag{14.49}$$

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

$$p(z) = \frac{1}{Z_p}\exp(-E(z)) \tag{14.50}$$

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，通过施加力的方式确定，它本身是势能的负梯度，即

$$\frac{\mathrm{d}r_i}{\mathrm{d}\tau} = -\frac{\partial E(z)}{\partial z_i} \tag{14.51}$$

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

$$K(r) = \frac{1}{2}\|r\|^2 = \frac{1}{2}\sum_i r_i^2 \tag{14.52}$$

系统的总能量是势能和动能之和，即

$$H(z,r) = E(z) + K(r) \tag{14.53}$$

其中 H 是哈密顿函数 (Hamiltonian function)。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

$$\frac{\mathrm{d}z_i}{\mathrm{d}\tau} = \frac{\partial H}{\partial r_i} \tag{14.54}$$

$$\frac{\mathrm{d}r_i}{\mathrm{d}\tau} = -\frac{\partial H}{\partial z_i} \tag{14.55}$$

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。

$$\begin{aligned}\frac{\mathrm{d}H}{\mathrm{d}\tau} &= \sum_i \left\{ \frac{\partial H}{\partial z_i}\frac{\mathrm{d}z_i}{\mathrm{d}\tau} + \frac{\partial H}{\partial r_i}\frac{\mathrm{d}r_i}{\mathrm{d}\tau} \right\} \\ &= \sum_i \left\{ \frac{\partial H}{\partial z_i}\frac{\partial H}{\partial r_i} - \frac{\partial H}{\partial r_i}\frac{\partial H}{\partial z_i} \right\} \\ &= 0\end{aligned} \tag{14.56}$$

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。换句话说，如果我们考虑变量 (z,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

$$V = \left(\frac{\mathrm{d}z}{\mathrm{d}\tau}, \frac{\mathrm{d}r}{\mathrm{d}\tau} \right) \tag{14.57}$$

这个场的散度为零，即

$$\begin{aligned}\mathrm{div}V &= \sum_i \left\{ \frac{\partial}{\partial z_i}\frac{\mathrm{d}z_i}{\mathrm{d}\tau} + \frac{\partial}{\partial r_i}\frac{\mathrm{d}r_i}{\mathrm{d}\tau} \right\} \\ &= \sum_i \left\{ \frac{\partial}{\partial z_i}\frac{\partial H}{\partial r_i} - \frac{\partial}{\partial r_i}\frac{\partial H}{\partial z_i} \right\} \\ &= 0\end{aligned} \tag{14.58}$$

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

$$p(z,r) = \frac{1}{Z_H}\exp(-H(z,r)) \tag{14.59}$$

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 $p(z,r)$ 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 z 和 r 是会发生变换，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 z 以一种系统化的方式发生圈套的变化，避免了随机游走的行为。

然而，哈密顿动态系统的变化对 $p(z,r)$ 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 $p(z,r)$ 的不变性。达到这个目标的最简单的方式是将 r 的值为一个从以 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，我们看到这也使得所求的概率分布保持了不变性。

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。完成这件事的一种方法是蛙跳 (leapfrog) 离散化。

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。

混合蒙特卡罗方法

14.6 估计划分函数

第 15 章 连续潜在变量

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。

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 conponent 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。

15.1 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。它也被称为 Karhunen-Loeve 变换。

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空间 (principal subspace)，使得投影数据的方差被最大化。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。

考察一组观测数据集 $\{x_n\}$，其中 $n = 1, \ldots, N$，因此 x_n 是一个 D 维欧几里德空间中的变量。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。样本集合

$$X = (x_1\ x_2\ \ldots x_N)^T_{N\times p} = \begin{pmatrix} x_1^T \\ x_2^T \\ \vdots \\ x_N^T \end{pmatrix} = \begin{pmatrix} x_{11} & x_{12} & \ldots & x_{1p} \\ x_{21} & x_{22} & \ldots & x_{2p} \\ \ldots & \ldots & \ldots & \ldots \\ x_{11} & x_{12} & \ldots & x_{np} \end{pmatrix}_{N\times p} \tag{15.1}$$

样本均值

$$\bar{x} = \frac{1}{N}\sum_{n=1}^{N} x_n = \frac{1}{N}(x_1\ x_2\ \dots\ x_n)\begin{pmatrix}1\\1\\\vdots\\1\end{pmatrix}_{N\times 1} = \frac{1}{N}X^T I_N \tag{15.2}$$

样本方差

$$\begin{aligned} S &= \frac{1}{N}\sum_{n=1}^{N}(x_n-\bar{x})(x_n-\bar{x})^T \\ &= \frac{1}{N}(x_1-\bar{x}\ x_2-\bar{x}\ \dots\ x_N-\bar{x})\begin{pmatrix}(x_1-\bar{x})^T\\(x_2-\bar{x})^T\\\vdots\\(x_N-\bar{x})^T\end{pmatrix} \\ &= \frac{1}{N}X^T\overbrace{(I_N-\frac{1}{N}I_N I_N^T)}^{H_N-\text{中心矩阵}}\cdot(I_N-\frac{1}{N}I_N I_N^T)^T\cdot X \\ &= \frac{1}{N}X^T H X \end{aligned} \tag{15.3}$$

中心矩阵 H 有以下良好的性质

$$H = I_N - \frac{1}{N}I_N I_N^T \tag{15.4}$$

$$H^T = H \tag{15.5}$$

$$H^n = H \tag{15.6}$$

最大方差形式

首先，考虑在一维空间 ($M = 1$) 上的投影。我们可心使用 D 维向量 u_1 定义这个空间的方向。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 被投影到一个标题值 $x_1^T x_n$ 上。投影数据的均值是 $u_1^T\bar{x}$。投影数据的方差为

$$\begin{aligned} J &= \frac{1}{N}\sum_{n=1}^{N}\{u_1^T x_n - u_1^T\bar{x}\}^2 \\ &= u_1^T\underbrace{\sum_{n=1}^{N}\frac{1}{N}(x_n-\bar{x})\cdot(x_n-\bar{x})^T}_{S}u_1 \\ &= u_1^T S u_1 \end{aligned} \tag{15.7}$$

S 是数据的协方差矩阵，优化问题变成

$$\begin{aligned} \hat{u_1} &= arg\,max\, u_1^T S u_1 \\ s.t\ \ u_1^T u_1 &= 1 \end{aligned} \tag{15.8}$$

利用拉格朗日乘子法，求偏导并令其为零，我们看到驻点满足

$$Su_1 = \lambda_1 u_1 \tag{15.9}$$

表明 u_1 一定是 S 的一个特征向量，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。

总结一下，主成分分析涉及到计算数据集的均值 $\bar{x}$ 和协方差矩阵 S，然后寻找 S 的对应于 M 个最大特征值的 M 个特征向量。

最小误差形式

引入 D 维基向量的一个完整的单位正交集合 $\{u_i\}$，其中 $i = 1, \ldots, D$，且满足

$$u_i^T u_j = \delta_{ij} \tag{15.10}$$

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纯性组合，即

$$x_n = \sum_{i=1}^{D} a_{ni} u_i \tag{15.11}$$

其中，系数 a_{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 $\{u_i\}$ 定义的新坐标系，原始的 D 个分量 $\{x_{n1}, \ldots, x_{nD}\}$ 被替换为一个等价的集合 $\{a_{n1}, \ldots, a_{nD}\}$。我们有 $a_{nj} = x_n^T u_j$，因此不失一般性

$$x_n = \sum_{i=1}^{D} (x_n^T u_i) u_i \tag{15.12}$$

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 x_n

$$\widetilde{x}_n = \sum_{i=1}^{M} z_{ni} u_i + \sum_{i=M+1}^{D} b_i u_i \tag{15.13}$$

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 $\{u_i\}, \{z_{ni}\}, \{b_i\}$，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 $\widetilde{x}_n$ 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

$$J = \frac{1}{N}\sum_{n=1}^{N}\|x_n - \widetilde{x}_n\|^2 = \frac{1}{N}\sum_{n=1}^{N}\sum_{i=M+1}^{D}(x_n^T u_i - \bar{x}^T u_i)^2 = \sum_{i=M+1}^{D} u_i^T S u_i \tag{15.14}$$

剩下的任务是关于 $\{u_i\}$ 对 J 进行最小化。同样利用拉格朗日乘子法能够得到

$$Su_i = \lambda_i u_i \tag{15.15}$$

PCA 的应用

首先考虑 PCA 对于数据压缩的应用。我们可以写出对数据向量 $\boldsymbol{x}_n$ 的 PCA 近似，形式为

$$\begin{aligned}\tilde{\boldsymbol{x}}_n &= \sum_{i=1}^{M}(\boldsymbol{x}_n^T\boldsymbol{u}_i)\boldsymbol{u}_i + \sum_{i=M+1}^{D}(\bar{\boldsymbol{x}}^T\boldsymbol{u}_i)\boldsymbol{u}_i \\ &= \bar{\boldsymbol{x}} + \sum_{i=1}^{M}(\boldsymbol{x}_n^T\boldsymbol{u}_i - \bar{\boldsymbol{x}}^T\boldsymbol{u}_i)\boldsymbol{u}_i\end{aligned} \tag{15.16}$$

其中我们使用了关系

$$\bar{\boldsymbol{x}} = \sum_{i=1}^{D}(\bar{\boldsymbol{x}}^T\boldsymbol{u}_i)\boldsymbol{u}_i \tag{15.17}$$

这个关系来自于 $\{\boldsymbol{u}_i\}$ 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 $\boldsymbol{x}_n$ 替换为 M 维向量，元素为 $(\boldsymbol{x}_n^T\boldsymbol{u}_i - \bar{\boldsymbol{x}}^T\boldsymbol{u}_i)$。M 的值越小，压缩的程度越大。

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。

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写成下面的形式

$$\boldsymbol{SU} = \boldsymbol{UL} \tag{15.18}$$

其中，$\boldsymbol{L}$ 是一个 $D \times D$ 的对角矩阵，元素为 λ_i，$\boldsymbol{U}$ 是一个 $D \times D$ 的正交矩阵，列为 $\boldsymbol{u}_i$。然后对于每个数据点 $\boldsymbol{x}_n$，我们定义一个变换，值为

$$\boldsymbol{y}_n = \boldsymbol{L}^{-\frac{1}{2}}\boldsymbol{U}^T(\boldsymbol{x}_n - \bar{\boldsymbol{x}}) \tag{15.19}$$

其中 $\bar{\boldsymbol{x}}$ 是样本均值。很明显，集合 $\{\boldsymbol{y}_n\}$ 的均值为零，协方差是单位矩阵，因为

$$\begin{aligned}\frac{1}{N}\sum_{n=1}^{N}\boldsymbol{y}_n\boldsymbol{y}_n^T &= \frac{1}{N}\sum_{n=1}^{N}\boldsymbol{L}^{-\frac{1}{2}}\boldsymbol{U}^T(\boldsymbol{x}_n-\bar{\boldsymbol{x}})(\boldsymbol{x}_n-\bar{\boldsymbol{x}})^T\boldsymbol{U}\boldsymbol{L}^{-\frac{1}{2}} \\ &= \boldsymbol{L}^{-\frac{1}{2}}\boldsymbol{U}^T\boldsymbol{S}\boldsymbol{U}\boldsymbol{L}^{-\frac{1}{2}} \\ &= \boldsymbol{L}^{-\frac{1}{2}}\boldsymbol{L}\boldsymbol{L}^{-\frac{1}{2}} = \boldsymbol{I}\end{aligned} \tag{15.20}$$

这个操作被称为对数据的白化 (whitening) 或者球人形化 (sphereing)。

主成分分析的另一个常见应用是数据可视化。例如，每个数据点被投影到二维 (M=2) 的主子空间中，从而数据点 $\boldsymbol{x}_n$ 被画在了一个笛卡尔坐标第中，坐标系由 $\boldsymbol{x}_n^T\boldsymbol{u}_1$ 和 $\boldsymbol{x}_n^T\boldsymbol{u}_2$ 定义，其中 $\boldsymbol{u}_1$ 和 $\boldsymbol{u}_2$ 是特征向量，对应于最大的和第二大的特征值。

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。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，对应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，我们将 $\boldsymbol{X}$ 定义为 $(N\times D)$ 维中心数据矩阵，它的第 n 行为 $(\boldsymbol{x}_n-\bar{\boldsymbol{x}})^T$。这样，协方差矩阵可以写成 $\boldsymbol{S}=N^{-1}\boldsymbol{X}^T\boldsymbol{X}$，对应的特征向量方程变成了

$$\frac{1}{N}\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{u}_i=\lambda_i\boldsymbol{u}_i \tag{15.21}$$

现在，将两侧左乘 $\boldsymbol{X}$，可得

$$\frac{1}{N}\boldsymbol{X}\boldsymbol{X}^T(\boldsymbol{X}\boldsymbol{u}_i)=\lambda_i(\boldsymbol{X}\boldsymbol{u}_i) \tag{15.22}$$

如果我们现在定义 $\boldsymbol{v}_i=\boldsymbol{X}\boldsymbol{u}_i$，那么我们有

$$\frac{1}{N}\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{v}_i=\lambda_i\boldsymbol{v}_i \tag{15.23}$$

它是 $N\times N$ 矩阵 $N^{-1}\boldsymbol{X}\boldsymbol{X}^T$ 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 15.23 两侧乘以 $\boldsymbol{X}^T$，可得

$$\left(\frac{1}{N}\boldsymbol{X}^T\boldsymbol{X}\right)(\boldsymbol{X}^T\boldsymbol{v}_i)=\lambda_i(\boldsymbol{X}^T\boldsymbol{v}_i) \tag{15.24}$$

从中我们可以看到 $(\boldsymbol{X}^T\boldsymbol{v}_i)$ 是 $\boldsymbol{S}$ 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 $\boldsymbol{u}_i\propto\boldsymbol{X}^T\boldsymbol{v}_i$

进行重新标度，使得 $\|\boldsymbol{u}_i\| = 1$。假设 $\boldsymbol{v}_i$ 的长度已经被归一化，那么我们有

$$\boldsymbol{u}_i = \frac{1}{(N\lambda_i)^{\frac{1}{2}}}\boldsymbol{X}^T\boldsymbol{v}_i \tag{15.25}$$

总结一下，为了应用这种方法，我们首先计算 $\boldsymbol{X}\boldsymbol{X}^T$，然后找到它的特征向量和特征值，之后使用公式 15.25 计算原始数据空间的特征向量。

15.2 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。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。
- 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 会给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 和 Roweis 独立提出。它与因子分析 (factor analysis) 密切相关。

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 $\boldsymbol{z}$，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 $p(\boldsymbol{z})$ 以及以潜在变量的值为条件，观测变量 $\boldsymbol{x}$ 的高斯条件概率分布 $p(\boldsymbol{x}|\boldsymbol{z})$。具体地说，$\boldsymbol{z}$ 上的先验概率分布是一个零均值单位协方差的高斯分布

$$p(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}|0, \boldsymbol{I}) \tag{15.26}$$

类似地，以潜在变量 z 的值为条件，观测变量 $\boldsymbol{x}$ 的条件概率分布还是高斯分布，形式为

$$p(\boldsymbol{x}|z) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{W}z + \boldsymbol{\mu}, \sigma^2\boldsymbol{I}) \tag{15.27}$$

其中 $\boldsymbol{x}$ 的均值是 z 的一个一般的线性函数，由 $D \times M$ 的矩阵 $\boldsymbol{W}$ 和 D 维向量 $\boldsymbol{\mu}$ 控制。注意，可以关于 $\boldsymbol{x}$ 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。

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 $\boldsymbol{x}$ 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

$$\boldsymbol{x} = \boldsymbol{W}z + \boldsymbol{\mu} + \boldsymbol{\epsilon} \tag{15.28}$$

其中 z 是一个 M 维高斯潜在变量，$\boldsymbol{\epsilon}$ 是一个 D 维零均值高斯分布的噪声变量，协方差为 $\sigma^2\boldsymbol{I}$。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 $\boldsymbol{W}, \boldsymbol{\mu}$ 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 $p(\boldsymbol{x})$ 的表达式。

$$p(\boldsymbol{x}) = \int p(\boldsymbol{x}|z)p(z)dz \tag{15.29}$$

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

$$p(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}, \boldsymbol{C}) \tag{15.30}$$

其中 $D \times D$ 协方差矩阵 C 被定义为

$$\begin{aligned} C &= \text{var}[\boldsymbol{x}] = \text{var}[\boldsymbol{W}z + \boldsymbol{\mu} + \boldsymbol{\epsilon}] \\ &= \text{var}[\boldsymbol{W}z] + \text{var}[\boldsymbol{\epsilon}] \\ &= \boldsymbol{W}\boldsymbol{I}\boldsymbol{W}^T + \sigma^2\boldsymbol{I} \\ &= \boldsymbol{W}\boldsymbol{W}^T + \sigma^2\boldsymbol{I} \end{aligned} \tag{15.31}$$

我们注意到预测概率分布是高斯分布，然后使用公式 15.28 计算它的均值和协方差，结果为

$$\mathbb{E}[\boldsymbol{x}] = \mathbb{E}[\boldsymbol{W}z + \boldsymbol{\mu} + \boldsymbol{\epsilon}] = \boldsymbol{\mu} \tag{15.32}$$

$$\text{var}[\boldsymbol{x}] = \boldsymbol{W}\boldsymbol{W}^T + \sigma^2\boldsymbol{I} \tag{15.33}$$

其中我们使用了下面的事实：z 和 $\boldsymbol{\epsilon}$ 是独立的随机变量，因此非相关。

预测分布 $p(\boldsymbol{x})$ 由参数 $\boldsymbol{W}, \boldsymbol{\mu}$ 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。

与预测分布 $p(\boldsymbol{x})$ 一样，我们也需要后验概率分布 $p(\boldsymbol{z}|\boldsymbol{x})$，这可以直接使用公式给出的线性高斯模型的结果写出来，结果为

$$p(\boldsymbol{z}|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{z}|\boldsymbol{M}^{-1}\boldsymbol{W}^T(\boldsymbol{x}-\boldsymbol{\mu}), \sigma^2\boldsymbol{M}^{-1}) \tag{15.34}$$

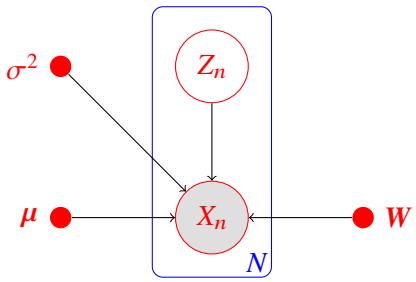
其中

$$\boldsymbol{M} = \sigma^2\boldsymbol{I}_M + \boldsymbol{W}^T\boldsymbol{W} \tag{15.35}$$

注意，后验均值依赖于 $\boldsymbol{x}$，而后验协方差与 $\boldsymbol{x}$ 无关。

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 $\boldsymbol{X} = \{\boldsymbol{x}_n\}$，概率 PCA 模型可以表示为一个有向图。



根据公式 15.30，对应的对数似然函数为

$$\begin{aligned}\ln p(\boldsymbol{X}|\boldsymbol{\mu}, \boldsymbol{W}, \sigma^2) &= \sum_{n=1}^{N} \ln p(\boldsymbol{x}_n|\boldsymbol{\mu}, \boldsymbol{W}, \sigma^2) \\ &= -\frac{ND}{2}\ln(2\pi) - \frac{N}{2}\ln|C| - \frac{1}{2}\sum_{n=1}^{N}(\boldsymbol{x}_n-\boldsymbol{\mu})^T C^{-1}(\boldsymbol{x}_n-\boldsymbol{\mu})\end{aligned} \tag{15.36}$$

令似然函数关于 $\boldsymbol{\mu}$ 的导数等于零，可以得到预期的结果 $\boldsymbol{\mu} = \bar{\boldsymbol{x}}$，代回到似然函数中，我们有

$$\begin{aligned}\ln p(\boldsymbol{X}|\boldsymbol{\mu}, \boldsymbol{W}, \sigma^2) &= -\frac{N}{2}\{D\ln(2\pi) + \ln|C|\} + \frac{N}{2}\mathrm{Tr}\left(C^{-1}\frac{1}{N}\sum_{n=1}^{N}(\boldsymbol{x}_n-\boldsymbol{\mu})^T(\boldsymbol{x}_n-\boldsymbol{\mu})\right) \\ &= -\frac{N}{2}\{D\ln(2\pi) + \ln|C| + \mathrm{Tr}(C^{-1}S)\}\end{aligned} \tag{15.37}$$

其中 S 是协方差矩阵。由于对数似然函数是 $\boldsymbol{\mu}$ 的二次函数，因此解具有唯一的最大值，可以通过计算二阶导数的方式验证这一点。

证明：迹运算描述 Frobenius 范数

设 A 是 m 行 n 列的矩阵，A 的行向量是 $\overrightarrow{b}_1^T, \ldots, \overrightarrow{b}_m^T$。那么

$$A = \begin{pmatrix} \overrightarrow{b}_1^T \\ \overrightarrow{b}_2^T \\ \vdots \\ \overrightarrow{b}_m^T \end{pmatrix}, \qquad A^T = \left(\overrightarrow{b}_1, \overrightarrow{b}_2, \ldots \overrightarrow{b}_m^T \right) \tag{15.38}$$

$$AA^T = \begin{pmatrix} \overrightarrow{b}_1^T \\ \overrightarrow{b}_2^T \\ \vdots \\ \overrightarrow{b}_m^T \end{pmatrix} \left(\overrightarrow{b}_1, \overrightarrow{b}_2, \ldots \overrightarrow{b}_m^T \right) = \begin{pmatrix} \overrightarrow{b}_1^T \overrightarrow{b}_1 & \overrightarrow{b}_1^T \overrightarrow{b}_2 & \ldots & \overrightarrow{b}_1^T \overrightarrow{b}_m \\ \overrightarrow{b}_2^T \overrightarrow{b}_1 & \overrightarrow{b}_2^T \overrightarrow{b}_2 & \ldots & \overrightarrow{b}_2^T \overrightarrow{b}_m \\ \vdots & \vdots & \ddots & \vdots \\ \overrightarrow{b}_m^T \overrightarrow{b}_1 & \overrightarrow{b}_m^T \overrightarrow{b}_2 & \ldots & \overrightarrow{b}_m^T \overrightarrow{b}_m \end{pmatrix} \tag{15.39}$$

因为迹运算返回的是矩阵对角线元素的和，所以：

$$\mathrm{Tr}(AA^T) = \sum_{i=1}^{m} \overrightarrow{b}_i^T \overrightarrow{b}_i \tag{15.40}$$

$\overrightarrow{b}_i^T$ 是矩阵 A 的第 i 行。$\overrightarrow{b}_i^T = (A_{i,1}, \ldots, A_{i,n})$，$\overrightarrow{b}_i^T \overrightarrow{b}_i$ 是矩阵 A 第 i 行行向量的内积。那么

$$\overrightarrow{b}_i^T \overrightarrow{b}_i = \sum_{j=1}^{n} A_{ij}^2 \tag{15.41}$$

推出

$$\mathrm{Tr}(AA^T) = \sum_{i=1}^{m} \sum_{j=1}^{n} A_{ij}^2 \tag{15.42}$$

即

$$\|A\|_F = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} A_{ij}^2} = \sqrt{\mathrm{Tr}(AA^T)} \tag{15.43}$$

令 $F = -\ln p(\boldsymbol{X}|\boldsymbol{\mu}, \boldsymbol{W}, \sigma^2)$(忽略掉常数项)。

$$F = \ln |C| - \mathrm{Tr}(C^{-1}S) \tag{15.44}$$

1. 令 $\frac{\mathrm{d}F}{\mathrm{d}\boldsymbol{W}} = 0$ 求 $\boldsymbol{W}$

$$\begin{aligned} \mathrm{d}F &= \mathrm{d}\ln|\boldsymbol{C}| + \mathrm{d}\mathrm{Tr}(\boldsymbol{C}^{-1}\boldsymbol{S}) \\ &= \mathrm{Tr}(\boldsymbol{C}^{-1}\mathrm{d}C) + \mathrm{Tr}(\mathrm{d}C^{-1}\boldsymbol{S}) \\ &= \mathrm{Tr}(\boldsymbol{C}^{-1}\mathrm{d}C) - \mathrm{Tr}(\boldsymbol{C}^{-1}\mathrm{d}\boldsymbol{C}\boldsymbol{C}^{-1}\boldsymbol{S}) \\ &= \mathrm{Tr}(\boldsymbol{C}^{-1}(\mathrm{d}\boldsymbol{W}\boldsymbol{W}^T + \boldsymbol{W}\mathrm{d}\boldsymbol{W}^T)) - \mathrm{Tr}(\boldsymbol{C}^{-1}(\mathrm{d}\boldsymbol{W}\boldsymbol{W}^T + \boldsymbol{W}\mathrm{d}\boldsymbol{W}^T)\boldsymbol{C}^{-1}\boldsymbol{S}) \end{aligned} \tag{15.45}$$

根据迹的性质，有

$$\mathrm{Tr}(\boldsymbol{C}^{-1}\mathrm{d}\boldsymbol{W}\boldsymbol{W}^T) = \mathrm{Tr}(\boldsymbol{W}\mathrm{d}\boldsymbol{W}^T\boldsymbol{C}^{-1}) = \mathrm{Tr}(\boldsymbol{C}^{-1}\boldsymbol{W}\mathrm{d}\boldsymbol{W}^T) \tag{15.46}$$

代入上式中,有

$$\begin{aligned} \mathrm{d}F &= 2\mathrm{Tr}(\boldsymbol{C}^{-1}\boldsymbol{W}\mathrm{d}\boldsymbol{W}^T) - 2\mathrm{Tr}(\boldsymbol{C}^{-1}\boldsymbol{S}\boldsymbol{C}^{-1}\boldsymbol{W}\mathrm{d}\boldsymbol{W}^T) \\ &= 2\mathrm{Tr}[(\boldsymbol{C}^{-1}\boldsymbol{W} - \boldsymbol{C}^{-1}\boldsymbol{S}\boldsymbol{C}^{-1}\boldsymbol{W})\mathrm{d}\boldsymbol{W}^T] \end{aligned} \tag{15.47}$$

因此,

$$\frac{1}{2}\frac{\mathrm{d}F}{\mathrm{d}\boldsymbol{W}} = \boldsymbol{C}^{-1}\boldsymbol{W} - \boldsymbol{C}^{-1}\boldsymbol{S}\boldsymbol{C}^{-1}\boldsymbol{W} \tag{15.48}$$

令 $\frac{\mathrm{d}F}{\mathrm{d}\boldsymbol{W}} = 0$,可求得

$$\begin{aligned} \boldsymbol{W} &= \boldsymbol{S}\boldsymbol{C}^{-1}\boldsymbol{W} \\ &= \boldsymbol{S}(\sigma^2\boldsymbol{I}_D + \boldsymbol{W}\boldsymbol{W}^T)^{-1}\boldsymbol{W} \\ &= \boldsymbol{S}\boldsymbol{W}(\sigma^2\boldsymbol{I}_M + \boldsymbol{W}^T\boldsymbol{W})^{-1} \end{aligned} \tag{15.49}$$

对 $\boldsymbol{W}^T\boldsymbol{W} = \boldsymbol{V}\Lambda\boldsymbol{V}^T$ 作分解,代入有

$$\begin{aligned} &\boldsymbol{S}\boldsymbol{W}(\boldsymbol{V}\sigma^2\boldsymbol{I}_M\boldsymbol{V}^T + \boldsymbol{V}\Lambda\boldsymbol{V}^T)^{-1} = \boldsymbol{W} \\ \Rightarrow\ &\boldsymbol{S}\boldsymbol{W}(\boldsymbol{V}^T)^{-1}(\sigma^2\boldsymbol{I}_M + \Lambda)^{-1}\boldsymbol{V}^{-1} = \boldsymbol{W} \\ \Rightarrow\ &\boldsymbol{S}\boldsymbol{W}\boldsymbol{V}(\sigma^2\boldsymbol{I}_M + \Lambda)^{-1} = \boldsymbol{W}\boldsymbol{V} \end{aligned} \tag{15.50}$$

对 $\boldsymbol{W}\boldsymbol{V}$ 正交化,有

$$\Lambda^{-\frac{1}{2}}\boldsymbol{V}^T\boldsymbol{W}^T\boldsymbol{W}\boldsymbol{V}\Lambda^{-\frac{1}{2}} = \Lambda^{-\frac{1}{2}}\boldsymbol{V}^T\boldsymbol{V}\Lambda\boldsymbol{V}^T\boldsymbol{V}\Lambda^{-\frac{1}{2}} = \boldsymbol{I} \tag{15.51}$$

因此,我们有

$$\boldsymbol{S}\boldsymbol{W}\boldsymbol{V}\Lambda^{-\frac{1}{2}} = \boldsymbol{W}\boldsymbol{V}\Lambda^{-\frac{1}{2}}(\sigma^2\boldsymbol{I} + \Lambda) \tag{15.52}$$

令 $U_M = \boldsymbol{W}\boldsymbol{V}\boldsymbol{\Lambda}^{-\frac{1}{2}}, L_M = \sigma^2\boldsymbol{I}_M + \Lambda$,我们有

$$\begin{aligned} &\boldsymbol{S}U_M = U_M L_M \\ \Rightarrow\ &\boldsymbol{W}_{ML} = U_M\Lambda^{\frac{1}{2}}\boldsymbol{V}^T = U_M(L_M - \sigma^2\boldsymbol{I}_M)^{\frac{1}{2}}\boldsymbol{V}^T \end{aligned} \tag{15.53}$$

2. 令 $\frac{\mathrm{d}F}{\mathrm{d}\sigma^2} = 0$ 求 σ^2

$$\begin{aligned} \mathrm{d}F &= \mathrm{Tr}(\boldsymbol{C}^{-1}\mathrm{d}\sigma^2\boldsymbol{I}) - \mathrm{Tr}(\boldsymbol{C}^{-1}\mathrm{d}\sigma^2\boldsymbol{C}^{-1}\boldsymbol{S}) \\ &= [\mathrm{Tr}(\boldsymbol{C}^{-1}) - \mathrm{Tr}(\boldsymbol{C}^{-1}\boldsymbol{S}\boldsymbol{C}^{-1})]\mathrm{d}\sigma^2 \end{aligned} \tag{15.54}$$

令

$$\frac{\mathrm{d}F}{\mathrm{d}\sigma^2} = \mathrm{Tr}(\boldsymbol{C}^{-1} - \boldsymbol{C}^{-1}\boldsymbol{S}\boldsymbol{C}^{-1}) = 0 \tag{15.55}$$

计算 $\boldsymbol{C}^{-1}$

$$\boldsymbol{C}^{-1} = (\sigma^2\boldsymbol{I}_D + \boldsymbol{W}\boldsymbol{W}^T)^{-1} = \sigma^{-2}\boldsymbol{I}_D - \sigma^{-2}\boldsymbol{W}(\sigma^2\boldsymbol{I}_M + \boldsymbol{W}^T\boldsymbol{W})^{-1}\boldsymbol{W}^T \tag{15.56}$$

等式两边同乘 $\boldsymbol{S}$ 有

$$\begin{aligned}\boldsymbol{S}\boldsymbol{C}^{-1} &= \sigma^{-2}\boldsymbol{S} - \sigma^{-2}\boldsymbol{S}\boldsymbol{W}(\sigma^2\boldsymbol{I}_M + \boldsymbol{W}^T\boldsymbol{W})^{-1}\boldsymbol{W}^T \\ &= \sigma^{-2}\boldsymbol{S} - \sigma^{-2}\boldsymbol{W}\boldsymbol{W}^T \quad (\text{因为}\boldsymbol{S}\boldsymbol{W}(\sigma^2\boldsymbol{I}_M + \boldsymbol{W}^T\boldsymbol{W})^{-1} = \boldsymbol{W})\end{aligned} \tag{15.57}$$

代入上式中，有

$$\begin{aligned}\boldsymbol{C}^{-1}\boldsymbol{S}\boldsymbol{C}^{-1} - \boldsymbol{C}^{-1} &= \boldsymbol{C}^{-1}\sigma^{-2}\boldsymbol{S} - \boldsymbol{C}^{-1}\sigma^{-2}\boldsymbol{W}\boldsymbol{W}^T - \boldsymbol{C}^{-1} \\ &= \sigma^{-2}\boldsymbol{C}^{-1}\boldsymbol{S} - \boldsymbol{C}^{-1}\sigma^{-2}\boldsymbol{W}\boldsymbol{W}^T - \boldsymbol{C}^{-1} \\ &= \sigma^{-2}(\sigma^{-2}\boldsymbol{S} - \sigma^{-2}\boldsymbol{W}\boldsymbol{W}^T) - \boldsymbol{C}^{-1}\sigma^{-2}\boldsymbol{W}\boldsymbol{W}^T - \boldsymbol{C}^{-1} \\ &= \sigma^{-2}(\sigma^{-2}\boldsymbol{S} - \sigma^{-2}\boldsymbol{W}\boldsymbol{W}^T) - \sigma^{-2}\boldsymbol{C}^{-1}(\boldsymbol{C} - \sigma^2\boldsymbol{I}_D) - \boldsymbol{C}^{-1} \\ &= \sigma^{-2}(\sigma^{-2}\boldsymbol{S} - \sigma^{-2}\boldsymbol{W}\boldsymbol{W}^T) - \sigma^{-2}\boldsymbol{I}_{\boldsymbol{D}} \\ &= \sigma^{-4}\boldsymbol{S} - \sigma^{-4}\boldsymbol{W}\boldsymbol{W}^T - \sigma^{-2}\boldsymbol{I}_D\end{aligned} \tag{15.58}$$

求得

$$\frac{\mathrm{d}F}{\mathrm{d}\sigma^2} = \mathrm{Tr}(\boldsymbol{C}^{-1} - \boldsymbol{C}^{-1}\boldsymbol{S}\boldsymbol{C}^{-1}) = \sigma^{-4}\mathrm{Tr}(\boldsymbol{S} - \boldsymbol{W}\boldsymbol{W}^T - \sigma^2\boldsymbol{I}_D) = 0 \tag{15.59}$$

我们有

$$\begin{aligned}\sigma^2\mathrm{Tr}(\boldsymbol{I}_D) &= \mathrm{Tr}(\boldsymbol{S}) - \mathrm{Tr}(\boldsymbol{W}^T\boldsymbol{W}) \\ D\sigma^2 &= \mathrm{Tr}(\boldsymbol{S}) - \mathrm{Tr}(L - \sigma^2\boldsymbol{I}_M) \\ &= \mathrm{Tr}(\boldsymbol{S}) - \mathrm{Tr}(L_M) + M\sigma^2 \\ (D-M)\sigma^2 &= \mathrm{Tr}(\boldsymbol{S}) - \mathrm{Tr}(L_M) \\ \sigma^2 &= \frac{\mathrm{Tr}(\boldsymbol{S}) - \mathrm{Tr}(L_M)}{D-M} \\ &= \frac{1}{D-M}\sum_{i=M+1}^{D}\lambda_i\end{aligned} \tag{15.60}$$

总结一下，

$$\boldsymbol{W}_{ML} = U_M(L_M - \sigma^2\boldsymbol{I}_M)^{\frac{1}{2}}\boldsymbol{V}^T \tag{15.61}$$

$$\sigma^2_{ML} = \frac{1}{D-M}\sum_{i=M+1}^{D}\lambda_i \tag{15.62}$$

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 15.28 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 $\boldsymbol{x}$ 都可以使用潜在空间中的后验均值和方差进行概括。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。

用于 PCA 的 EM 算法

概率 PCA 模型可以根据连续潜在空间 $\boldsymbol{z}$ 上的积分或求和来表示，其中对于每个数据点 $\boldsymbol{x}_n$，都存在一个对应的潜在变量 $\boldsymbol{z}_n$。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

$$L_c = \ln p(\boldsymbol{X}, \boldsymbol{Z} | \boldsymbol{\mu}, \boldsymbol{W}, \sigma^2) = \sum_{n=1}^{N} \{\ln p(\boldsymbol{x}_n | \boldsymbol{z}_n) + \ln p(\boldsymbol{z}_n)\} \tag{15.63}$$

这里，$\ln p(\boldsymbol{x}_n, \boldsymbol{z}_n)$ 是

$$\begin{aligned}
\ln p(\boldsymbol{x}_n, \boldsymbol{z}_n) &= \ln[p(\boldsymbol{x}_n | \boldsymbol{z}_n) p(\boldsymbol{z}_n)] \\
&= \ln \left[\frac{1}{(2\pi)^{\frac{D}{2}} \sigma^D} \exp(-\frac{1}{\sigma^2} \|\boldsymbol{x}_n - \boldsymbol{\mu} - \boldsymbol{W}\boldsymbol{z}_n\|^2) \exp(-\frac{1}{2} \|\boldsymbol{z}_n\|^2) \right] \\
&= -\frac{D}{2} \ln \sigma^2 - \frac{1}{\sigma^2} \|\boldsymbol{x}_n - \boldsymbol{\mu} - \boldsymbol{W}\boldsymbol{z}_n\|^2 - \frac{1}{2} \|\boldsymbol{z}_n\|^2 \quad \text{(忽略了常系数)} \\
&= -\frac{D}{2} \ln \sigma^2 - \frac{1}{\sigma^2} \|\boldsymbol{x}_n - \boldsymbol{\mu}\|^2 - \frac{1}{\sigma^2} \|\boldsymbol{W}\boldsymbol{z}_n\|^2 + \frac{1}{\sigma^2} (\boldsymbol{x}_n - \boldsymbol{\mu})^T (\boldsymbol{W}\boldsymbol{z}_n) - \frac{1}{2} \|\boldsymbol{z}_n\|^2
\end{aligned} \tag{15.64}$$

1. E-步：$\boldsymbol{\theta} = (\sigma^2, \boldsymbol{W})$，求 L_c 关于 $p(\boldsymbol{z}_n | \boldsymbol{\theta})$ 的期望

$$
\begin{aligned}
Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) &= \sum_{n=1}^{N} \int (\ln p(\boldsymbol{x}_n, \boldsymbol{z}_n)) p(\boldsymbol{z}_n|\boldsymbol{x}_n, \boldsymbol{\theta}^{(t)}) \mathrm{d}\boldsymbol{z}_n \\
&= \sum_{n=1}^{N} \left\{ - \int \frac{D}{2} \ln \sigma^2 p(\boldsymbol{z}_n|\theta^{(t)}) \mathrm{d}\boldsymbol{z}_n \right. \\
&\quad - \int \frac{1}{2\sigma^2} \|\boldsymbol{x}_n - \boldsymbol{\mu}\|^2 p(\boldsymbol{z}_n|\boldsymbol{x}_n, \boldsymbol{\theta}^{(t)}) \mathrm{d}\boldsymbol{z}_n \\
&\quad - \int \frac{1}{2\sigma^2} \|\boldsymbol{W}\boldsymbol{z}_n\|^2 p(\boldsymbol{z}_n|\boldsymbol{x}_n, \boldsymbol{\theta}^{(t)}) \mathrm{d}\boldsymbol{z}_n \\
&\quad - \int \frac{1}{\sigma^2} (\boldsymbol{x}_n - \boldsymbol{\mu})^T (\boldsymbol{W}\boldsymbol{z}_n) p(\boldsymbol{z}_n|\boldsymbol{x}_n, \boldsymbol{\theta}^{(t)}) \mathrm{d}\boldsymbol{z}_n \\
&\quad \left. - \int \frac{1}{2} \|\boldsymbol{z}_n\|^2 p(\boldsymbol{z}_n|\boldsymbol{x}_n, \boldsymbol{\theta}^{(t)}) \mathrm{d}\boldsymbol{z}_n \right\} \\
&= -\frac{ND}{2} \ln \sigma^2 - \frac{N}{2\sigma^2} \mathrm{Tr}(\boldsymbol{S}) \\
&\quad - \sum_{n=1}^{N} \int \left[\frac{1}{2\sigma^2} \|\boldsymbol{W}\boldsymbol{z}_n\|^2 - \frac{1}{\sigma^2} (\boldsymbol{x}_n - \boldsymbol{\mu})^T (\boldsymbol{W}\boldsymbol{z}_n) + \frac{1}{2} \|\boldsymbol{z}_n\|^2 \right] p(\boldsymbol{z}_n|\boldsymbol{x}_n, \boldsymbol{\theta}^{(t)}) \mathrm{d}\boldsymbol{z}_n
\end{aligned}
\tag{15.65}
$$

其中我们用到了

$$
\mathrm{Tr}(\boldsymbol{S}) = \frac{1}{N} \sum_{n=1}^{N} \left((\boldsymbol{x}_n - \boldsymbol{\mu})(\boldsymbol{x}_n - \boldsymbol{\mu})^T \right) \tag{15.66}
$$

我们定义

$$
\langle \boldsymbol{z}_n \rangle = \int \boldsymbol{z}_n p(\boldsymbol{z}_n|\boldsymbol{x}_n, \boldsymbol{\theta}^{(t)}) \mathrm{d}\boldsymbol{z}_n = \boldsymbol{M}_{(t)}^{-1} \boldsymbol{W}_{(t)}^T (\boldsymbol{x}_n - \boldsymbol{\mu}) \tag{15.67}
$$

因为 $(\boldsymbol{z}|\boldsymbol{x} - \boldsymbol{\mu}) \sim \mathcal{N}(\boldsymbol{M}^{-1}\boldsymbol{W}^T(\boldsymbol{x}_n - \boldsymbol{\mu}), \sigma^2 \boldsymbol{M}^{-1}$,这里 $\boldsymbol{M} = \sigma^2 \boldsymbol{I}_M + \boldsymbol{W}^T\boldsymbol{W})$

$$
\langle \boldsymbol{z}_n, \boldsymbol{z}_n^T \rangle = \int \boldsymbol{z}_n \boldsymbol{z}_n^T p(\boldsymbol{z}_n|\boldsymbol{x}_n, \boldsymbol{\theta}^{(t)}) \mathrm{d}\boldsymbol{z}_n = \sigma_{(t)}^2 \boldsymbol{M}_{(t)}^{-1} + \langle \boldsymbol{z}_n \rangle \langle \boldsymbol{z}_n \rangle^T \tag{15.68}
$$

因为

$$
\mathrm{Cov}(\boldsymbol{z}_n) = \mathbb{E}(\boldsymbol{z}_n \boldsymbol{z}_n^T) - \mathbb{E}(\boldsymbol{z}_n)\mathbb{E}(\boldsymbol{z}_n^T) \tag{15.69}
$$

因此,$Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)})$ 为

$$
\begin{aligned}
Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) &= -\frac{ND}{2} \ln \sigma^2 - \frac{N}{2\sigma^2} \mathrm{Tr}(\boldsymbol{S}) \\
&\quad - \sum_{n=1}^{N} \left\{ \frac{1}{2\sigma^2} \mathrm{Tr}(\boldsymbol{W}^T \boldsymbol{W} \langle \boldsymbol{z}_n, \boldsymbol{z}_n^T \rangle) - \frac{1}{\sigma^2} (\boldsymbol{x}_n - \boldsymbol{\mu})^T \boldsymbol{W} \langle \boldsymbol{z}_n \rangle + \frac{1}{2} \mathrm{Tr}(\langle \boldsymbol{z}_n, \boldsymbol{z}_n^T \rangle) \right\}
\end{aligned}
\tag{15.70}
$$

2. M-步:最大化 $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)})$

(a)

$$\begin{aligned}
&\frac{\mathrm{d}Q}{\mathrm{d}\boldsymbol{W}} = \frac{1}{\sigma^2}\sum_{n=1}^{N}(\boldsymbol{x}_n - \boldsymbol{\mu})\langle z_n^T\rangle - \frac{1}{\sigma^2}\sum_{n=1}^{N}\boldsymbol{W}\langle z_n, z_n^T\rangle = 0 \\
\Rightarrow &\boldsymbol{W}\sum_{n=1}^{N}\langle z_n, z_n^T\rangle = \sum_{n=1}^{N}(\boldsymbol{x}_n - \boldsymbol{\mu})\langle z_n^T\rangle \\
\Rightarrow &\boldsymbol{W}^{(t+1)} = \left(\sum_{n=1}^{N}(\boldsymbol{x}_n - \boldsymbol{\mu})\langle z_n^T\rangle\right)\left(\sum_{n=1}^{N}\langle z_n, z_n^T\rangle\right)^{-1}
\end{aligned} \tag{15.71}$$

(b)

$$\begin{aligned}
\frac{\partial Q}{\partial \sigma^2} = &-\frac{ND}{2}\frac{1}{\sigma^2} + \frac{N}{2\sigma^4}\mathrm{Tr}(\boldsymbol{S}) + \frac{1}{2\sigma^4}\sum_{n=1}^{N}\mathrm{Tr}(\boldsymbol{W}^T\boldsymbol{W}\langle z_n, z_n^T\rangle) \\
&- \frac{1}{\sigma^4}\sum_{n=1}^{N}(\boldsymbol{x}_n - \boldsymbol{\mu})^T\boldsymbol{W}\langle z_n^T\rangle = 0 \\
\Rightarrow \sigma^{2(t+1)} = &\frac{1}{D}\left[\mathrm{Tr}(\boldsymbol{S}) + \frac{1}{N}\sum_{n=1}^{N}\mathrm{Tr}(\boldsymbol{W}^T\boldsymbol{W}\langle z_n, z_n^T\rangle) - \frac{2}{N}(\boldsymbol{x}_n - \boldsymbol{\mu})^T\boldsymbol{W}\langle z_n^T\rangle\right]
\end{aligned} \tag{15.72}$$

因为

$$\boldsymbol{W}\sum_{n=1}^{N}\langle z_n, z_n^T\rangle = \sum_{n=1}^{N}(\boldsymbol{x}_n - \boldsymbol{\mu})\langle z_n^T\rangle \tag{15.73}$$

我们可以简化式子，得到

$$\sigma^{2(t+1)} = \frac{1}{D}\left[\mathrm{Tr}(\boldsymbol{S}) + \frac{1}{N}\sum_{n=1}^{N}(\boldsymbol{x}_n - \boldsymbol{\mu})^T\boldsymbol{W}^{(t+1)}\langle z_n\rangle\right] \tag{15.74}$$

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。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 算法是迭代的，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。注意，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。EM 算法的另一个特征是，我们可以取极限 $\sigma^2 \to 0$，对应于标准的 PCA，仍然可以得到一个合法的类似 EM 的算法。

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。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 $\boldsymbol{\mu}, \boldsymbol{W}$ 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。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。它涉及到对 $\boldsymbol{W}$ 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。

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 $\boldsymbol{z}$ 的条件下观测变量 $\boldsymbol{x}$ 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

$$p(\boldsymbol{x}|\boldsymbol{z}) = \mathcal{N}(\boldsymbol{x}|\boldsymbol{W}\boldsymbol{z} + \boldsymbol{\mu}, \Psi) \tag{15.75}$$

其中 Ψ 是一个 $D \times D$ 的对角矩阵。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 Φ 中与每个坐标相关联的独立的变量，然后描述矩阵 $\boldsymbol{W}$ 中的变量之间的协方差。

15.3 核 PCA

15.4 非线性隐变量模型

第 16 章 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。这样的模型组合有时被称为委员会 (committee)。

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。

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。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应用于回归问题。

16.1 贝叶斯模型平均

将模型组合方法与贝叶斯模型平均区分开是很重要的，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。

$$p(\boldsymbol{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \Sigma_k) \tag{16.1}$$

这是模型组合的一个例子。

现在假设我们有若干个不同的模型，索引为 $h = 1, \ldots, H$，先验概率分布为 $p(h)$。例如一个模型可能是高斯混合模型，另一个模型可能是西分布的混合。数据集上的边缘概率分布为

$$p(\boldsymbol{X}) = \sum_{h=1}^{H} p(\boldsymbol{X}|h)p(h) \tag{16.2}$$

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。随着数据规模的增加，这个不确定性会减小，后验概率分布 $p(h|\boldsymbol{X})$ 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 $\boldsymbol{z}$ 的不同的值生成，即由不同的分量生成。

16.2 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。

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。委员会预测为

$$y_{COM} = \frac{1}{M}\sum_{m=1}^{M} y_m(\boldsymbol{x}) \tag{16.3}$$

这个方法被称为自助聚集 (bootstrap aggregation) 或者打包 (bagging)。

假设我们试图预测的真实的回归函数为 $h(\boldsymbol{x})$，从而每个模型的输出可以写成真实值加上误差的形式，即

$$y_m(\boldsymbol{x}) = h(\boldsymbol{x}) + \epsilon_m(\boldsymbol{x}) \tag{16.4}$$

这样，平方和误差函数的形式为

$$\mathbb{E}_{\boldsymbol{x}}[\{y_m(\boldsymbol{x}) - h(\boldsymbol{x})\}^2] = \mathbb{E}_{\boldsymbol{x}}[\epsilon_m(\boldsymbol{x})^2] \tag{16.5}$$

其中 $\mathbb{E}[\cdot]$ 表示关于输入向量 $\boldsymbol{x}$ 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

$$E_{AV} = \frac{1}{M}\sum_{m=1}^{M} \mathbb{E}_{\boldsymbol{x}}[\epsilon_m(\boldsymbol{x})^2] \tag{16.6}$$

类似地，委员会方法的预测的期望误差为

$$\begin{aligned} E_{COM} &= \mathbb{E}_{\boldsymbol{x}}\left[\left\{\frac{1}{M}\sum_{m=1}^{M} y_m(\boldsymbol{x}) - h(\boldsymbol{x})\right\}^2\right] \\ &= \mathbb{E}_{\boldsymbol{x}}\left[\left\{\frac{1}{M}\sum_{m=1}^{M} \epsilon_m(\boldsymbol{x})\right\}^2\right] \end{aligned} \tag{16.7}$$

如果我们假设误差的均值为零，且不具有相关性，即

$$\mathbb{E}_{\boldsymbol{x}}[\epsilon_m(\boldsymbol{x})] = 0 \tag{16.8}$$

$$\mathbb{E}_{\boldsymbol{x}}[\epsilon_m(\boldsymbol{x})\epsilon_l(\boldsymbol{x})] = 0, \quad m \neq l \tag{16.9}$$

那么我们有

$$E_{COM} = \frac{1}{M} E_{AV} \tag{16.10}$$

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。

16.3 提升方法

提升 (boosting) 方法是一种常用的统计学习方法，应用广泛且有效。在分类问题中，它通过改变训练样本的权重，学习多个分类器，并将这些分类器进行线性组合，提高分类的性能。

提升方法基于这样一种思路：对于一个复杂任务来说，将多个专家的判断进行适当的综合所得出的判断，要比其中任何一个专家单独的判断好。提升方法就是从弱学习算法出发，反复学习，得到一系列弱分类器 (又称基本分类器)，然后组合这些弱分类器，构成一个强分类器。对提升方法来说，有两个问题需要回答：

1. 在每一轮如何改变训练数据的权值或概率分布；
2. 如果将弱分类器组合成一个强分类器。

AdaBoost 的做法是，提高那些被前一轮弱分类器错误分类样本的权值，而降低那些被正确分类样本的权值。这样一来，那些没有得到正确分类的数据，由于其权值的加大而受到后一轮的弱分类器的更大关注。于是，分类问题被一系列的弱分类器“分而治之”。至于第 2 个问题，即弱分类器的组合，AdaBoost 采取加权表决的方法。具体地，加大分类误差率小的弱分类器的权值，使其在表决中起较大作用，减小分类误差率大的弱分类器的权值，使其在表决中起较小的作用。AdaBoost 的巧妙之外就在于它将这些想法自然且有效地实现在一种算法里。

AdaBoost 算法

输入：训练数据集 $T = \{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\}$；弱学习算法；

输出：最终分类器 $G(x)$

(1) 初始化训练数据的权值分布

$$D_1 = (w_{11}, \ldots, w_{1i}, \ldots, w_{1N}), w_{1i} = \frac{1}{N}, i = 1, 2, \ldots, N \tag{16.11}$$

假设训练数据集具有均匀的权值分布，即每个训练样本在基本分类器的学习中作用相同，这一假设保证第 1 步能够在原始数据上学习基本分类器 $G_1(x)$。

(2) 对 $m = 1, 2, \ldots, M$

a. 使用具有权值分布 D_m 的训练数据集学习，得到基本分类器

$$G_m(x)X \to \{-1, +1\} \tag{16.12}$$

b. 计算 $G_m(x)$ 在训练数据集上的分类误差率

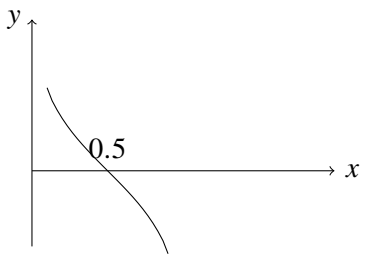
$$e_m = \sum_{i=1}^{N} P(G_m(x_i) \neq y_i) = \sum_{i=1}^{N} w_{mi} I(G_m(x_i) \neq y_i) \tag{16.13}$$

这表明，$G_m(x)$ 在加权的训练数据集上的分类误差率是被 $G_m(x)$ 误分类样本的权值之和。

c. 计算 $G_m(x)$ 的系数

$$a_m = \frac{1}{2} log \frac{1 - e_m}{e_m} \tag{16.14}$$

a_m 表示 $G_m(x)$ 在最终分类器中的重要性。



由上图可知，当 $e_m \leqslant \frac{1}{2}$ 时，$a_m \geqslant 0$，并且 a_m 随着 e_m 的减小而增大，所以**分类误差率越小的基本分类器在最终分类器中的作用越大**。

d. 更新训练数据集的权值分布

$$D_{m+1} = (w_{m+1,1}, \ldots, m_{m+1,i}, \ldots, w_{m+1,N}) \tag{16.15}$$

$$w_{m+1,i} = \frac{w_{mi}}{Z_m} exp(-a_m y_i G_m(x_i)), i = 1, 2, \ldots, N \tag{16.16}$$

这里，Z_m 是规范化因子

$$Z_m = \sum_{i=1}^{N} w_{mi} exp(-a_m y_i G_m(x_i)) \tag{16.17}$$

它使 D_{m+1} 成为一个概率分布。式 16.15 可以写成

$$w_{m+1,i} = \begin{cases} \frac{w_{mi}}{Z_m} e^{-a_m}, & G_m(x_i) = y_i \\ \frac{w_{mi}}{Z_m} e^{a_m}, & G_m(x_i) \neq y_i \end{cases} \tag{16.18}$$

由此可知，**被基本分类器 $G_m(x)$ 误分类样本的权值得以扩大，而被正确分类样本的权值却得以缩小**。因此，误分类样本在下一轮学习中起更大的作用。不改变所给的训练数据，而不断改变训练数据权值的分布，使得训练数据在基本分类器的学习中起不同的作用，这是 AdaBoost 的一个特点。

(3) 构建基本分类器的线性组合

$$f(x) = \sum_{m=1}^{M} a_m G_m(x) \tag{16.19}$$

得到最终分类器

$$G(x) = sign(f(x)) = sign\left(\sum_{m=1}^{M} a_m G_m(x)\right) \tag{16.20}$$

线性组合 $f(x)$ 实现 M 个基本分类器的加权表决。系数 a_m 表示了基本分类器 $G_m(x)$ 的重要性，这里，所有 a_m 之和并不为 1.$f(x)$ 的符号决定实例 x 的类，$f(x)$ 的绝对值表示分类的确信度。利用基本分类器的线性组合构建最终分类器是 AdaBoost 的另一个特点。

AdaBoost 算法的训练误差分析

AdaBoost 最基本的性质是它能在学习过程中不断减少训练误差，即在训练数据集上的分类误差率。关于这个问题有下面的定理：

定理 16.1. AdaBoost 的训练误差界

AdaBoost 算法最终分类器的训练误差界为

$$\frac{1}{N}\sum_{i=1}^{N} I(G(x_i) \neq y_i) \leqslant \frac{1}{N}\sum_{i} \exp(-y_i f(x_i)) = \prod_{m} Z_m \tag{16.21}$$

这里，$G(x)$，$f(x)$ 和 Z_m 分别由 16.20,16.19和 16.17 给出。 ♡

证明：

当 $G(x_i) \neq y_i$ 时，$y_i f(x_i) < 0$，因而 $\exp(-y_i f(x_i)) \geqslant 1$。由此直接推导出前半部分。

后半部分要用到 Z_m 的定义式 16.17 及 16.16 的变形：

$$w_{mi} \exp(-\alpha_m y_i G_m(x_i)) = Z_m w_{m+1,i} \tag{16.22}$$

现推导如下

$$\begin{aligned}
&\frac{1}{N}\sum_i \exp(-y_i f(x_i)) \\
&= \frac{1}{N}\sum_i \exp\left(-\sum_{m=1}^{M}\alpha_m y_i G_m(x_i)\right) \\
&= \sum_i w_{1i}\prod_{m=1}^{M}\exp(-\alpha_m y_i G_m(x_i)) \\
&= Z_1\sum_i w_{2i}\prod_{m=2}^{M}\exp(-\alpha_m y_i G_m(x_i)) \\
&\cdots \\
&= Z_1 Z_2 \ldots Z_{M-1}\sum_i w_{Mi}\exp(-\alpha_m y_i G_m(x_i)) \\
&= \prod_{m=1}^{M} Z_m
\end{aligned} \tag{16.23}$$

这一定理说明，可以在每一轮选取适当的 G_m 使得 Z_m 最小，从而使训练误差下降最快。对二类分类问题，有如下结果：

定理 16.2. 二类分类问题 AdaBoost 的训练误差界

$$\prod_{m=1}^{M} Z_m = \prod_{m=1}^{M}[2\sqrt{e_m(1-e_m)}] = \prod_{m=1}^{M}\sqrt{(1-4\gamma_m^2)} \leqslant \exp\left(-2\sum_{m=1}^{M}\gamma_m^2\right) \tag{16.24}$$

这里，$\gamma_m = \frac{1}{2} - e_m$。 ♡

证明： 由 Z_m 的定义式 16.17 及 16.13 得

$$\begin{aligned}
Z_m &= \sum_{i=1}^{N} w_{mi}\exp(-\alpha_m y_i G_m(x_i)) \\
&= \sum_{y_i = G_m(x_i)} w_{mi} e^{-\alpha_m} + \sum_{y \neq G_m(x_i)} w_{mi} e^{\alpha_m} \\
&= (1-e_m)e^{-\alpha_m} + e_m e^{\alpha_m} \\
&= 2\sqrt{e_m(1-e_m)} = \sqrt{1-4\gamma_m^2}
\end{aligned} \tag{16.25}$$

至于不等式

$$\prod_{m=1}^{M}\sqrt{(1-4\gamma_m^2)} \leqslant \exp\left(-2\sum_{m=1}^{M}\gamma_m^2\right) \tag{16.26}$$

则可由 e^x 和 $\sqrt{1-x}$ 在点 $x=0$ 的泰勒展开式推出不等式 $\sqrt{(1-4\gamma_m^2)} \leqslant \exp(-2\gamma_m^2)$，进而得到。

AdaBoost 算法的解释

AdaBoost 算法还有另一个解释，即可以认为 AdaBoost 算法是模型为加法模型、损失函数为指数函数、学习算法为前向分步算法时的二类分类学习方法

16.4 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 $\boldsymbol{x}$，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，这个过程对应于一个二叉树的遍历。决策树 (decision tree) 是一种基本的分类与回归方法。本节主要讨论用于分类的决策树。决策树模型呈树形结构，在分类问题中，表示基于特征对实例进行分类的过程。它可以认为是 if-then 规则的集合，也可以认为是定义在特征空间与类空间上的条件概率分布。其主要优点是模型具有可读性，分类速度快。学习时，利用训练数据，根据损失函数最小化的原则建立决策树模型。预测时，对新的数据，利用决策树模型进行分类。决策树学习通常包括 3 个步骤：特征选择、决策树的生成和决策树的修剪。这些决策树学习的思想主要源于由 Quinlan 在 1986 年提出的 ID3 算法和 1993 年提出的 C4.5 算法，以及由 Breiman 等人在 1984 年提出的 CART 算法。

本节首先介绍决策树的基本概念，然后通过 ID3 和 C4.5 介绍特征的选择、决策树的生成以及决策树的修剪，最后介绍 CART 算法。

决策树模型与学习

分类决策树模型是一种描述对实例进行分类的树形结构。决策树由结点 (node) 和有向边 (directed edge) 组成。结点有两种类型：内部结点和叶结点。内部结点表示一个特征或属性，叶结点表示一个类。

1. 可以将决策树看成一个 it-then 规则的集合。
2. 决策树还表示给定特征条件下类的条件概率分布。这一条件概率分布定义在特征空间的一个划分 (partition) 上。
3. 决策树学习本质上是从训练数据集中归纳出一组分类规则；从另一个角度看，决策树学习是由训练数据集估计条件概率模型。

特征选择

直观上，如果一个特征具有更好的分类能力，或者说，按照这一特征将训练数据集分割成子集，使得各个子集在当前条件下有最好的分类，那么就更应该选择这个特征。信息增益(information gain)就能够很好地表示这一直观的准则。

信息增益特征 A 对训练数据集 D 的信息增益 $g(D, A)$，定义为集合 D 的经验熵 $H(D)$ 与特征 A 给定条件下 D 的经验条件熵 $H(D|A)$ 之差，即

$$g(D, A) = H(D) - H(D|A)$$

一般地，熵 $H(Y)$ 与条件熵 $H(Y|X)$ 之差称为互信息 (mutual information)。

信息增益的算法

输入：训练数据集 D 和特征 A；

输出：特征 A 对训练数据集 D 的信息增益 $g(D, A)$

(1) 计算数据集 D 的经验熵 $H(D)$

$$H(D) = -\sum_{k=1}^{K} \frac{|C_k|}{|D|} log_2 \frac{|C_k|}{|D|} \tag{16.27}$$

(2) 计算特征 A 对数据集 D 的经验条件熵 $H(D|A)$

$$H(D|A) = \sum_{i=1}^{n} \frac{|D_i|}{|D|} H(|D_i|) = -\sum_{i=1}^{n} \frac{|D_i|}{|D|} \sum_{k=1}^{K} \frac{D_{ik}}{D_i} log_2 \frac{|D_{ik}|}{|D_i|} \tag{16.28}$$

(3) 计算信息增益

$$g(D, A) = H(D) - H(D|A) \tag{16.29}$$

以信息增益作为划分训练数据集的特征，存在偏向选择取值较多的特征的问题。使用信息增益比 (information gain ratio) 可以对这一问题进行校正。这是特征选择的另一准则。

信息增益比

$$g_R(D, A) = \frac{g(D, A)}{H_A(D)} \tag{16.30}$$

其中，$H_A(D) = -\sum_{i=1}^{n} \frac{|D_i|}{D} log_2 \frac{|D_i|}{|D|}$，n 是特征 A 取值的个数。

决策树的生成

1. ID3 算法

   输入：训练数据集 D，特征集 A，阈值 ϵ；

   输出：决策树 T

   (1) 若 D 中所有实例属于同一类 C_k，则 T 为单结点树，并将类 C_k 作为该结点的类标记，返回 T

   (2) 若 $A = \emptyset$，则 T 为单结点树，并将 D 中实例数最大的类 C_k 作为该结点的类标记，返回 T

   (3) 否则按信息增益算法计算 A 中各特征对 D 的信息增益，选择信息增益最大的特征 A_g

   (4) 如果 A_g 的信息增益小于阈值 ϵ，则置 T 为单结点树，并将 D 中实例数最大的类作为标记，返回 T

   (5) 否则，对 A_g 的每一可能值 a_i，依 $A_g = a_i$ 将 D 分割为若干非空子集 D_i，将 D_i 中实例数最大的类作为标记，构建子结点，由结点及其子结点构成树 T，返回 T

   (6) 对第 i 个子结点，以 D_i 为训练集，以 $A - \{A_g\}$ 为特征集，递归地调用 (1) (5)，得到子树 T_i，返回 T

2. C4.5 的生成算法

   C4.5 算法与 ID3 算法相似，C4.5 算法对 ID3 算法进行了改进。C4.5 在生成的过程

中，用信息增益比来选择特征。

决策树的剪枝

决策树生成算法递归地产生决策树，直到不能继续下去为止。这样产生的树往往对训练数据的分类很准确，但对未知的测试数据的分类却没有那么准确，即出现过拟合现象。

CART 算法

分类回归树 (classification and regression tree, CART)，是在给定输入随机变量 X 条件下输出随机变量 Y 的条件概率分布的学习方法。CART 假设决策树是二叉树，内部结点特征的取值为“是”和“否”，左分支是取值为“是”的分支，右分支是取值为“否”的分支。这样的决策树等价于递归地二分每个特征，将输入空间即特征空间划分为有限个单元，并在这些单元上确定预测的概率分布，也就是在输入给定的条件下输出的条件概率分布。

1. 决策树生成：基于训练数据集生成决策树，生成的决策树要尽量大；决策树的生成就是递归地构建二叉决策树的过程。对回归树用平方误差最小化准则，对分类树用基尼指数 (Gini index) 最小化准则，进行特征选择，生成二叉树。
2. 决策树剪枝：用验证数据集对已生成的树进行剪枝并选择最优子树，这时用损失函数最小作为剪枝的标准
   (1) 剪枝，形成一个子树序列
   (2) 在剪枝得到的子树序列 $T_0, T_1, \ldots, T_n$ 中通过交叉验证选取最优子树 T_a

16.5 条件混合模型

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)。

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以及 Logistic 回归模型的混合。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。

16.6 logistic 模型的混合