第 1 章　绪　论

第 2 章　概率分布

第 3 章　线性回归模型

到目前为止，这本书主要讨论的内容还主要集中在非监督学习上，之前讨论的密度估计问题和数据聚类问题都属于这一类。现在我们将目光转向监督学习，从回归问题开始监督学习的相关讨论。回归问题的目的，是在给定了 D 维输入变量 $\mathbf{x}$ 的情况下，对一个或多个连续目标变量 t 进行预测。在第 1 章中我们已经见过这样的问题，也就是多项式曲线拟合问题。多项式是一个广泛的函数类型中的特例，这种函数类型称为线性回归模型。线性回归模型共同的性质是，它们都是可调参数的线性函数，我们将在本章节中重点研究它们。最简单的线性模型同样也是输入变量的线性函数。不过我们也可以通过将输入变量的非线性函数进行线性组合，从而得到更多的函数类型，这些非线性函数称为基底函数 (basis function)。这些模型都是参数的线性函数，同时还是关于输入变量的非线性函数，这样的特点使它们具有很多简单有效的性质。

我们要做的工作是，在给定了一个训练集 $\{\mathbf{x}_n\}$ 和对应的目标值 $\{t_n\}$(其中共有 N 组数据，$n = 1, ..., N$) 的情况下，对一个新的输入量 $\mathbf{x}$ 预测 t 的值。最简单的方法是直接构建一个适当的函数 $y(\mathbf{x})$，并认为在 $\mathbf{x}$ 处的函数值就是预测的 t 值。从概率的角度来说，一般地，我们要对预测分布 $p(t|\mathbf{x})$ 建模，因为预测分布可以表达每一个 $\mathbf{x}$ 所对应的 t 的不确定性。根据这个条件分布，我们可以通过将适当的损失函数进行最小化，从而对任何的 $\mathbf{x}$ 做出预测 t。在第 1.5.5 节中曾经讨论过，对于实值变量，通常会选择平方损失函数，这个损失函数的最优解是在 t 的条件期望处取得的。

尽管线性模型在模式识别的实际应用中具有明显的局限性，尤其是在应对高维的输入空间时显得力不从心，但它们具备的性质实在是太好了，而且是一些后文中更加实用的模型的基础，所以线性模型还是很值得讨论一下的。

3.1 线性基底函数模型

最简单的线性回归模型是输入变量的线性组合

$$y(\mathbf{x}, \mathbf{w}) = w_0 + w_1 x_1 + ... + w_D x_D \tag{3.1}$$

其中 $\mathbf{x} = (x_1, ..., x_D)^{\mathrm{T}}$。这个模型通常称为线性回归函数。这个模型最重要的性质是，它是关于参数 $w_0, ..., w_D$ 的线性函数。不过还有个特殊之处，它同时还是关于输入变量 x_i 的线性函数，这就是这个模型最明显的短板。所以我们将这个类型的模型进行扩展成输入变量非线性函数的线性组合，也就是

$$y(\mathbf{x}, \mathbf{w}) = w_0 + \sum_{j=1}^{M-1} w_j \phi_j(\mathbf{x}) \tag{3.2}$$

其中 $\phi_j(\mathbf{x})$ 称为基底函数 (basis function)。设参数 j 的最大值为 $M-1$ 可以使参数的总数量为 M，表达起来更加方便一些。

参数 w_0 称为偏差参数 (bias parameter)，是控制全部数据发生固定偏移的参数 (注意不要和统计学中的"bias" 搞混了)。通常来说把它表示为一个附加的 "基底函数" $\phi_0(\mathbf{x}) = 1$ 要更加方便一些，于是

$$y(\mathbf{x}, \mathbf{w}) = \sum_{j=0}^{M-1} w_j \phi_j(\mathbf{x}) = \mathbf{w}^{\mathrm{T}} \boldsymbol{\phi}(\mathbf{x}) \tag{3.3}$$

其中 $\mathbf{w} = (w_0, ..., w_{M-1})$，$\boldsymbol{\phi} = (\phi_0, \phi_{M-1})$。在模式识别的很多实际应用中，我们会对原始的数据进行某些形式的预处理，或者称为特征提取。如果将原始变量表示为向量 $\mathbf{x}$，那么特征就可以表示为基底函数 $\{\phi_j(\mathbf{x})\}$。

利用非线性基底函数，使得函数 $y(\mathbf{x}, \mathbf{w})$ 成为输入向量 $\mathbf{x}$ 的非线性函数。但 (3.2) 这样的函数仍然称为线性模型，因为它们是关于 $\mathbf{w}$ 的线性函数。正是这种关于参数的线性性质，使得线性模型的分析变得非常简单。但是这个性质所导致的问题也比较令人难受，具体内容将在第 3.6 节中进行讨论。

第 1 章中讨论的多项式回归问题是线性模型的典型案例，在这个问题中仅有一个输入变量 x，基底函数是 x 的幂函数，于是 $\phi_j(x) = x^j$。多项式基底函数的一大缺陷在于，它们都是输入变量的全局函数，所以在局部的调整会影响到全局。可以将输入控件进行划分，并在每个区域内分别进行多项式拟合，从而解决这个问题，这样得到的模型称为样条函数 (spline functions, Hastie et al., 2001)。

此外，基底函数还有很多的选择。比如，

$$\phi_j(x) = \exp\left\{-\frac{(x - \mu_j)^2}{2s^2}\right\} \tag{3.4}$$

其中 μ_j 控制了基底函数在输入控件中的位置，参数 s 控制了基底函数的空间尺度。这种基底函数通常称为高斯基底函数，尽管这些函数并没有明确的概率解释，而且归一化是完全不可能的，因为基底函数还要乘上一个参数 w_j。

另一种常见的基底函数称为 sigmoid 基底函数，

$$\phi_j(x) = \sigma\left(\frac{x-\mu_j}{s}\right) \tag{3.5}$$

其中 $\sigma(a)$ 为对数几率 sigmoid 函数 (logistic sigmoid function)，

$$\sigma(a) = \frac{1}{1+\exp(-a)} \tag{3.6}$$

等价地，我们也可以利用”tanh” 函数，因为它与对数几率函数是有关联的，$\tanh(a) = 2\sigma(2a) - 1$，所以一般地，对数几率函数的线性组合等价于 tanh 函数的线性组合。这些基底函数的图像如图 3.1 所示。

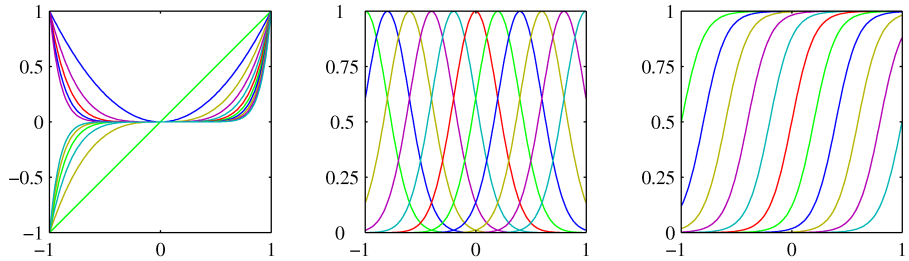

图 3.1: 基底函数示例，从左至右分别为多项式基底函数，高斯基底函数 (3.4) 和 sigmoid 基底函数 (3.5)。

还有另外一种基底函数称为 Fourier 基底函数，它可以展开为正弦函数。每个基底函数都代表一个特定的频率，并且在空间中可以无限延伸。与之不同的是，位于输入空间的有限区域内的基底函数必然包含了不同空间频率的频谱。在很多信号处理的实际应用中，需要研究位于空间和频率上的基底函数都是有必要的，于是产生了小波这一类型的函数。为了简化问题，它们往往被定义成正交的。当输入变量存在于规则的点阵中，比如在处理时间序列中的连续时间点，或者图像中的像素时，小波是最适用的。关于小波的具体内容可以参考 Ogden (1997), Mallat (1999) 和 Vidakovic (1999)。

不过，本章节主要研究的内容与基底函数的选择无关，所以在讨论的过程中并不会指明采用了哪一种基底函数，除非要进行数值计算。实际上，我们的讨论中基本都会使用 $\boldsymbol{\phi}(\mathbf{x}) = \mathbf{x}$ 作为基底函数。此外，为了让符号变得简洁一些，我们会主要针对单一目标变量 t 展开讨论，不过在第 3.1.5 节会简单研究一下如何处理多个目标变量的情况。

3.1.1 最大似然与最小二乘法

在第 1 章中，我们通过将平方和误差函数进行最小化来拟合多项式函数。我们同时也证明了这个误差函数就是在高斯噪声模型下的最大似然解。让我们再次把目光转向这个问题，并研究求解它的最小二乘法，并研究最小二乘法与最大似然之间的关系。

和以前一样，我们假设目标变量 t 是由函数 $y(\mathbf{x}, \mathbf{w})$ 加上高斯噪声得到的：

$$t = y(\mathbf{x}, \mathbf{w}) + \epsilon \tag{3.7}$$

其中 ϵ 是均值为 0，精度 (逆方差) 为 β 的高斯随机变量。于是

$$p(t|\mathbf{x}, \mathbf{w}, \beta) = \mathcal{N}(t|y(\mathbf{x}, \mathbf{w}), \beta^{-1}) \tag{3.8}$$

回想一下，如果我们构造出平方损失函数，那么对于任意的 $\mathbf{x}$，做出的预测将是目标变量的条件均值。——**第 1.5.5 节** 对于高斯条件分布 (3.8)，条件均值为

$$\mathbb{E}[t|\mathbf{x}] = \int t p(t|\mathbf{x}) \, \mathrm{d}t = y(\mathbf{x}, \mathbf{w}) \tag{3.9}$$

需要注意的是，将噪声视为高斯噪声的假设隐含着一个含义，那就是给定 $\mathbf{x}$ 的条件下 t 的条件分布是单峰值的，这可能会在一些应用中出现问题。有一种构造出多峰值条件分布的方法是混合条件高斯分布，这个问题将在第 14.5.1 节中详细讨论。

现在来研究一下数据集里的输入量 $\mathbf{X} = \{\mathbf{x}_1, ..., \mathbf{x}_N\}$ 和它们对应的目标值 $t_1, ..., t_N$。我们将目标值 $\{t_n\}$ 合并表示为列向量 $\mathbf{t}$，这里要与多元变量 t 区分开，$\mathbf{t}$ 表示的是多个一元变量合并在一起的列向量。假设这些数据是从分布 (3.8) 独立取出的，于是可以写出似然函数的表达式，而似然函数事实上是两个可调参数 $\mathbf{w}$ 和 β 的函数：

$$p(\mathbf{t}|\mathbf{X}, \mathbf{w}, \beta) = \prod_{n=1}^{N} \mathcal{N}(t_n|\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n), \beta^{-1}) \tag{3.10}$$

其中用到了 (3.3)。需要注意的是，在回归 (或分类) 这样的监督学习问题中，我们并不需要对输入变量的分布进行建模。由于 $\mathbf{x}$ 始终会存在于分布的条件变量中，所以从现在开始我们索性将 $p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \beta)$ 这样的表达中将 $\mathbf{x}$ 剔除掉，以保证符号的简洁。对似然函数求对数，并利用一元高斯分布的标准形式 (1.46)，可得

$$\begin{aligned} \ln p(\mathbf{t}|\mathbf{w}, \beta) &= \sum_{n=1}^{N} \ln \mathcal{N}(t_n|\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n), \beta^{-1}) \\ &= \frac{N}{2} \ln \beta - \frac{N}{2} \ln(2\pi) - \beta E_D(\mathbf{w}) \end{aligned} \tag{3.11}$$

其中的平方和误差函数为

$$E_D(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^{N} \{t_n - \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)\}^2 \tag{3.12}$$

在写出了似然函数之后，就可以利用最大似然确定 $\mathbf{w}$ 和 β 了。首先考虑关于 $\mathbf{w}$ 进行最大化。根据第 1.2.5 节中的内容，带有高斯噪声的线性模型条件分布的似然函数最大化等价于将平方和误差函数 $E_D(\mathbf{w})$ 进行最小化。对数似然函数 (3.11) 的梯度为

$$\nabla \ln p(\mathbf{t}|\mathbf{w},\beta) = \beta \sum_{n=1}^{N} \{t_n - \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)\}\boldsymbol{\phi}(\mathbf{x}_n)^{\mathrm{T}} \tag{3.13}$$

令梯度等于 0，

$$0 = \sum_{n=1}^{N} t_n \boldsymbol{\phi}(\mathbf{x}_n)^{\mathrm{T}} - \mathbf{w}^{\mathrm{T}} \left(\sum_{n=1}^{N} \boldsymbol{\phi}(\mathbf{x}_n)\boldsymbol{\phi}(\mathbf{x}_n)^{\mathrm{T}} \right) \tag{3.14}$$

求解这个方程，可得

$$\mathbf{w}_{\mathrm{ML}} = (\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^{\mathrm{T}}\mathbf{t} \tag{3.15}$$

这就是最小二乘问题中的正规方程 (normal equations)。其中的 $\boldsymbol{\Phi}$ 是 $N \times M$ 维矩阵，称为设计矩阵 (design matrix)，其元素为 $\Phi_n j = \phi_j(\mathbf{x}_n)$，于是

$$\boldsymbol{\Phi} = \begin{pmatrix} \phi_0(\mathbf{x}_1) & \phi_1(\mathbf{x}_1) & \cdots & \phi_{M-1}(\mathbf{x}_1) \\ \phi_0(\mathbf{x}_2) & \phi_1(\mathbf{x}_2) & \cdots & \phi_{M-1}(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_0(\mathbf{x}_N) & \phi_1(\mathbf{x}_N) & \cdots & \phi_{M-1}(\mathbf{x}_N) \end{pmatrix} \tag{3.16}$$

其中，

$$\boldsymbol{\Phi}^{\dagger} \equiv (\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^{\mathrm{T}} \tag{3.17}$$

为矩阵 $\boldsymbol{\Phi}$ 的 Moore-Penrose 伪逆 (Moore-Penrose pseudo-inverse, Rao and Mitra, 1971; Golub and Van Loan, 1996)。可以将它看成是矩阵的逆在矩阵并非方阵情况下的扩展。实际上，如果 $\boldsymbol{\Phi}$ 为方阵且可逆，利用 $(\mathbf{AB})^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1}$，可以看出 $\boldsymbol{\Phi}^{\dagger} \equiv \boldsymbol{\Phi}^{-1}$。

此时我们可以获得更多的关于偏差参数 w_0 的信息。将偏差参数单独拆分出来，于是误差函数 (3.12) 就变成了

$$E_D(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^{N} \{t_n - w_0 - \sum_{j=1}^{M-1} w_j \phi_j(\mathbf{x}_n)\}^2 \tag{3.18}$$

关于 w_0 求导并令导数等于 0，可以求解出 w_0

$$w_0 = \overline{t} - \sum_{j=1}^{M-1} w_j \overline{\phi_j} \tag{3.19}$$

其中，

$$\overline{t} = \frac{1}{N} \sum_{n=1}^{N} t_n, \quad \overline{\phi_j} = \frac{1}{N} \sum_{n=1}^{N} \phi_j(\mathbf{x}_n) \tag{3.20}$$

于是偏差 w_0 补偿了目标值 (在训练集上的) 的平均值与基底函数值的加权平均值之间的差异。

我们同样可以关于噪声的精度参数 β 将对数似然函数 (3.11) 进行最大化，可以得到

$$\frac{1}{\beta_{\mathrm{ML}}} = \frac{1}{N}\sum_{n=1}^{N}\{t_n - \mathbf{w}_{\mathrm{ML}}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)\}^2 \tag{3.21}$$

于是可以看出，噪声精度的倒数是由回归函数与目标值之间残差的方差给出的。

3.1.2 最小二乘法的几何意义

现在我们研究一下最小二乘法的几何意义。假想一个 N 维的空间，其坐标轴为 t_n，那么 $\mathbf{t} = (t_1, ..., t_N)^{\mathrm{T}}$ 为该空间中的向量。在 N 个数据点处所估计的基底函数 $\phi_j(\mathbf{x}_n)$ 也同样可以表示为该空间中的向量 $\boldsymbol{\varphi}_j$，如图 3.2 所示。需要注意的是，$\boldsymbol{\varphi}_j$ 对应的是 $\boldsymbol{\Phi}$ 的第 j 列，而 $\boldsymbol{\phi}(\mathbf{x}_n)$ 表示的是 $\boldsymbol{\Phi}$ 的第 n 行。如果基底函数的数量 M 小于数据点的数量 N，那么这 M 个向量 $\boldsymbol{\varphi}_j(\mathbf{x}_n)$ 将可以展开成 M 维的线性子空间 $\mathcal{S}$。我们定义 $\mathbf{y}$ 为 N 维向量，其元素 $y_n = y(\mathbf{x}_n, \mathbf{w})$，$n = 1, ..., N$。由于 $\mathbf{y}$ 是向量 $\boldsymbol{\varphi}_j$ 的任意线性组合，所以它可以存在于 M 维子空间中的任何位置。这样一来，平方和误差 (3.12) 就等于 $\mathbf{y}$ 与 $\mathbf{t}$ 之间欧氏距离的平方 (但相差一个因子 1/2)。因此，$\mathbf{w}$ 的最小二乘解与位于子空间 $\mathcal{S}$ 中的 $\mathbf{y}$ 的选择有关。根据图 3.2，似乎这个解就是 $\mathbf{t}$ 在空间 $\mathcal{S}$ 中的正交投影，而事实正是如此，只需注意到 $\mathbf{y}$ 的解是由 $\boldsymbol{\Phi}\mathbf{w}_{\mathrm{ML}}$ 给出的，然后利用正交投影就可以验证这个猜想。——**习题 3.2**

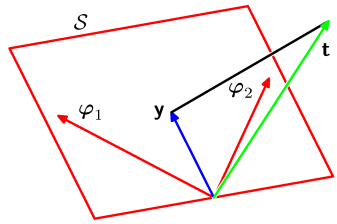

图 3.2: 分量为 $t_1, ..., t_N$ 的 N 维空间中最小二乘解的几何解释。通过寻找数据向量 $\mathbf{t}$ 在子空间中的正交投影确定最小二乘回归函数，这个子空间是基底函数 $\phi_j(\mathbf{x})$ 通过线性组合展开形成的，基底函数 $\boldsymbol{\varphi}_j$ 为长度为 N，元素为 $\phi_j(\mathbf{x}_n)$ 的向量。

在实际应用中，如果 $\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}$ 是接近奇异的，那么直接求解正规方程的话可能会比较困难。特别地，当两个或多个基底向量 $\boldsymbol{\varphi}_j$ 共线或接近共线时，最终得到的参数值可能会比较大。这样的近似简并在真实数据集的处理中并不罕见。计算困难的问题可以通过奇异值分解 (SVD, singular value decomposition, Press et al., 1992; Bishop and Nabney, 2008) 来解决。需要注意的是，附加的正则项可以确保矩阵的非奇异性，即使是简并的情况下也不例外。

3.1.3 顺序学习

形如最大似然解 (3.15) 这样的批处理方法要求一次性处理整个训练集，这样的做法导致的计算压力是非常大的。我们已经在第 1 章中讨论过，如果数据集是充分大的，那么使用顺序算法，也就是在线算法 (on-line algorithm) 就是比较划算的了，毕竟每次只需处理一组数据，并时刻更新模型参数。顺序学习也很适合实时应用，毕竟数据是以数据流的形式一点一点流过来的，而且不能等到全部的数据都到达后再做预测。

我们可以通过如下的随机梯度下降法 (stochastic gradient descent)，也就是顺序梯度下降法 (sequential gradient descent) 实现顺序学习算法。假设误差函数 $E = \sum_n E_n$ 是由数据的总和构成的，那么在观测到模式 n 之后，随机梯度下降法会利用如下方式更新参数 $\mathbf{w}$：

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} - \eta \nabla E_n \tag{3.22}$$

其中的 τ 为迭代的循环变量，η 为学习率参数。很快我们会提及 η 的选取方法。参数 $\mathbf{w}$ 的值可以初始化为 $\mathbf{w}^{(0)}$。对于平方和误差函数 (3.12)，

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} + \eta(t_n - \mathbf{w}^{(\tau)\mathrm{T}}\boldsymbol{\phi}_n)\boldsymbol{\phi}_n \tag{3.23}$$

其中 $\boldsymbol{\phi}_n = \boldsymbol{\phi}(\mathbf{x}_n)$。这个算法称为最小均方算法 (LMS algorithm, least-mean-squares algorithm)。η 的选取还需要考虑算法的收敛性 (Bishop and Nabney, 2008)。

3.1.4 正则化最小二乘法

在第 1.1 节中，我们在误差函数中加入了正则项，从而克服过拟合的问题，于是完整的误差函数为

$$E_D(\mathbf{w}) + \lambda E_W(\mathbf{w}) \tag{3.24}$$

其中 λ 为正则化系数，可以控制正则项 $E_W(\mathbf{w})$ 和基于数据的误差 $E_D(\mathbf{w})$ 之间的权重。最简单的正则项就是权重的平方和

$$E_W(\mathbf{w}) = \frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} \tag{3.25}$$

如果我们使用的是平方和误差函数，

$$E_D(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{t_n - \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)\}^2 \tag{3.26}$$

那么完整的误差函数就是

$$\frac{1}{2}\sum_{n=1}^{N}\{t_n - \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)\}^2 + \frac{\lambda}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} \tag{3.27}$$

这种正则项在机器学习中称为权重衰减 (weight decay)，因为在顺序学习算法中，这样的正则项“鼓励”权重衰减到 0，除非数据不允许。在统计学中，这是参数收缩 (parameter shrinkage) 的典型示例，因为参数值在不断向 0 衰减。这种正则项的有事在于，误差函数保留了一个 $\mathbf{w}$ 的二次项，所以可以求误差函数最小化的闭式解。讲得详细一些，令 (3.27) 关于 $\mathbf{w}$ 的梯度为 0 并求解 $\mathbf{w}$，可以得到

$$\mathbf{w} = (\lambda\mathbf{I} + \mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi})^{-1}\mathbf{\Phi}^{\mathrm{T}}\mathbf{t} \tag{3.28}$$

这是最小二乘解 (3.15) 的简单扩展。

一个更加一般的正则项为

$$\frac{1}{2}\sum_{n=1}^{N}\{t_n - \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)\}^2 + \frac{\lambda}{2}\sum_{j=1}^{M}|w_j|^q \tag{3.29}$$

其中，如果 $q = 2$，就与二次正则项 (3.27) 一样了。在 q 取不同的值时，正则项函数的图像如图 3.3 所示。

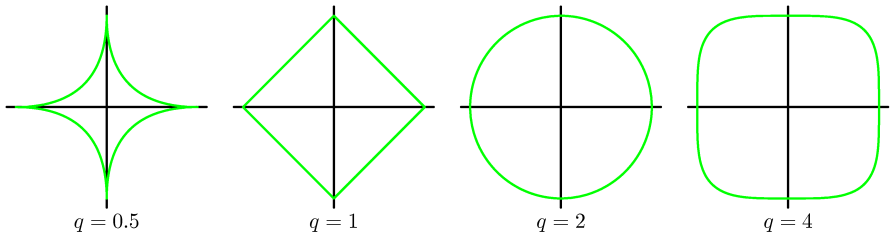

图 3.3: 正则项 (3.29) 在 q 取不同值时的图像

如果 $q = 1$ 那么就变成了统计学中的 lasso 方法 (Tibshirani, 1996)。如果 λ 充分大，那么某些系数 w_j 会变成 0，会形成基底函数不起作用的稀疏模型 (sparse model)。为了验证这一点，首先要留意到，对 (3.29) 进行最小化等价于将未正则化的平方和误差 (3.12) 在带有适当参数 η 的以下约束下进行最小化——**习题 3.5**

$$\sum_{j=1}^{M}|w_j|^q \leqslant \eta \tag{3.30}$$

这样就可以利用拉格朗日乘数法将两种方法结合起来——**附录 E**。误差函数在约束条件 (3.30) 下的最小值，也就是稀疏性的来源如图 3.4 所示。随着 λ 的增加，变成 0 的参数数量也在不断增加。

正则化可以通过限制有效模型的复杂度，在有限大小的数据集上训练出复杂的模型，同时避免严重过拟合的情况出现。然而，确定模型的最优复杂度这一问题就从寻找合适的基底函数数量变成了确定合适的正则化系数 λ。我们将在本章后面的内容中回到这个问题。

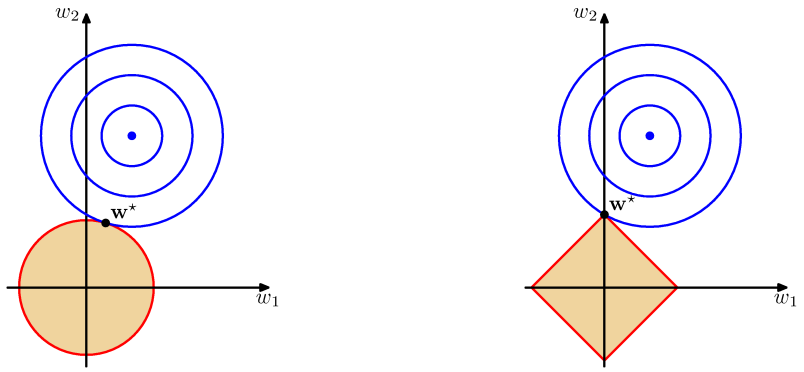

图 3.4: 未正则化的误差函数 (蓝色曲线) 与 $q = 2$(二次正则项，左) 和 1(lasso 正则项，右) 时的约束区域 (3.30)，参数向量 $\mathbf{w}$ 的最优解为 $\mathbf{w}^\star$。lasso 正则项可以给出 $w_1^\star = 0$ 这样的稀疏解。

考虑到实际应用和分析计算的简便，在本章的剩余内容中我们主要使用二次正则项 (3.27)。

3.1.5 多项输出

到目前为止我们一直在研究单一目标变量 t 的情况。但在一些实际应用中，我们还可能会希望得到 $K > 1$ 个目标变量，可以写成向量 $\mathbf{t}$。这可以利用对 $\mathbf{t}$ 的每个分量都使用不同的基底函数，于是问题就拆分成了所有分量相互无关的回归问题。然而更加常见的方法是对目标变量的分量采用相同的基底函数

$$\mathbf{y}(\mathbf{x}, \mathbf{w}) = \mathbf{W}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}) \tag{3.31}$$

其中的 $\mathbf{y}$ 为 K 维列向量，$\mathbf{W}$ 为 $M \times K$ 维的参数矩阵，$\boldsymbol{\phi}(\mathbf{x})$ 为 M 维的列向量，其元素为 $\phi_j(\mathbf{x})$，且 $\phi_0(\mathbf{x}) = 1$。如果我们假设目标向量的条件分布是各向同性的高斯分布，

$$p(\mathbf{t}|\mathbf{x}, \mathbf{W}, \beta) = \mathcal{N}(\mathbf{t}|\mathbf{W}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}), \beta^{-1}\mathbf{I}) \tag{3.32}$$

如果观测数据为 $\mathbf{t}_1, ..., \mathbf{t}_N$，那么就可以将它们集合在矩阵 $\mathbf{T}$ 中，$\mathbf{T}$ 为 $N \times K$ 维的矩阵，且它的第 n 行就是 $\mathbf{t}_n^{\mathrm{T}}$。类似地，我们也可以把输入向量 $\mathbf{x}_1, ..., \mathbf{x}_N$ 集合成矩阵 $\mathbf{X}$。那么对数似然函数就可以写成

$$\begin{aligned}\ln p(\mathbf{T}|\mathbf{X}, \mathbf{W}, \beta) &= \sum_{n=1}^{N} \ln \mathcal{N}(\mathbf{t}_n|\mathbf{W}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n), \beta^{-1}\mathbf{I}) \\ &= \frac{NK}{2}\ln\left(\frac{\beta}{2\pi}\right) - \frac{\beta}{2}\sum_{n=1}^{N}\|\mathbf{t}_n - \mathbf{W}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)\|^2\end{aligned} \tag{3.33}$$

和以前一样，可以对似然函数关于 $\mathbf{W}$ 进行最大化，得到的解为

$$\mathbf{W}_{\mathrm{ML}} = (\mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi})^{-1}\mathbf{\Phi}^{\mathrm{T}}\mathbf{T} \tag{3.34}$$

如果对每个目标变量 t_k 都做一下验证，那么

$$\mathbf{w}_k = (\mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi})^{-1}\mathbf{\Phi}^{\mathrm{T}}\mathbf{t}_k = \mathbf{\Phi}^{\dagger}\mathbf{t}_k \tag{3.35}$$

其中 $\mathbf{t}_k$ 为 $t_{nk}, n = 1, ..., N$ 组成的 N 维列向量。于是回归问题的解就拆分开了，而且只需要计算一个伪逆矩阵 $\mathbf{\Phi}^{\dagger}$，因为这个矩阵是所有向量 $\mathbf{w}_k$ 所共用的。

这些结论可以直接推广到带有一般的协方差矩阵的高斯噪声分布中。——**习题 3.6** 和上述情况一样，这样可以将回归问题拆分成 K 个相互独立的回归问题。这样的结果并不很出人意料，因为参数 $\mathbf{W}$ 仅与高斯噪声分布的均值有关，而且从第 2.3.4 节中我们可以得知，多元高斯分布均值的最大似然解是与协方差无关的。所以从现在开始，为了简化起见，我们会一直对单一目标变量 t 进行讨论。

3.2 偏差-方差分解

到目前为止所讨论的线性回归模型中，我们都做了这样的假设，即基底函数的形式和数量都是确定的。根据第 1 章中的相关内容，使用最大似然法 (或者最小二乘法) 可能会导致过拟合，尤其是在较小的数据集上训练复杂模型的时候更容易出这个问题。然而，为了防止过拟合的出现而限制基底函数的数量的话也会造成负面的影响，因为这样的做法严重限制了模型的灵活性，使得模型抓住数据中关键规律的能力下降。尽管加入正则项可以控制参数较多的模型出现过拟合的可能，但如何确定正则项系数 λ 又成了一个新的问题。通过将正则化误差函数关于权重向量 $\mathbf{w}$ 和正则化系数 λ 求取最小化来求正则项系数的做法明显不可靠，因为很明显最后的结果一定是 $\lambda = 0$。

在从前的讨论中我们可以看出，容易造成过拟合现象是最大似然方法的极大劣势，不过在贝叶斯方法的参数边缘化时完全不会出现这个问题。在本章中我们将使用贝叶斯观点研究模型的复杂度。不过在此之前，先来研究一个频率域观点的模型复杂度问题，即偏差-方差 (bias-variance) 之间的权衡。尽管我们将在线性基底函数模型的背景下介绍这个概念，毕竟解释起来比较简单，但这个内容并非局限于此，对于其他的模型也是适用的。

在第 1.5.5 节中研究回归问题的决策论时，我们探究了一些损失函数，在给定条件分布 $p(t|\mathbf{x})$ 时，每种损失函数都可以得到相应的最优预测。比较广泛的选择是平方损失函数，其对应的最优预测为条件期望，将其表示为 $h(\mathbf{x})$，那么

$$h(\mathbf{x}) = \mathbb{E}[t|\mathbf{x}] = \int tp(t|\mathbf{x})\ \mathrm{d}t \tag{3.36}$$

此时需要区分一下决策论中的平方损失函数与模型参数最大似然估计中的平方和误差函数。这里可能会用到类似正则化或完整贝叶斯方法这样比最小二乘法更加复杂的方法来确定条件分布 $p(t|\mathbf{x})$。在进行预测时，它们都可以与平方损失函数相结合。

在第 1.5.5 节中已经证明了期望平方损失可以写成如下形式：

$$\mathbb{E}[L] = \int \{y(\mathbf{x}) - h(\mathbf{x})\}^2 p(\mathbf{x})\,\mathrm{d}\mathbf{x} + \iint \{h(\mathbf{x}) - t\}^2 p(\mathbf{x}, t)\,\mathrm{d}\mathbf{x}\,\mathrm{d}t \tag{3.37}$$

回想一下，与 $y(\mathbf{x})$ 无关的第 2 项是由于数据中的噪声而出现的，表示的是期望误差所能达到的最小的值。第 1 项与函数 $y(\mathbf{x})$ 的选择有关，所以我们要做的就是选择一个使得该项取得最小值的 $y(\mathbf{x})$。由于这是一个非负项，所以它所能达到的最小值是 0。如果数据是无限的 (而且计算资源也是无限的)，原则上来说我们可以得到任意精确度的回归函数 $h(\mathbf{x})$，也就随之得到了最佳的 $y(\mathbf{x})$。然而实际应用中可不会这么理想，数据集 $\mathcal{D}$ 是有限的，于是回归函数 $h(\mathbf{x})$ 的精确形式也就无从知晓了。

如果使用带有参数 $\mathbf{w}$ 的函数 $y(\mathbf{x}, \mathbf{w})$ 对 $h(\mathbf{x})$ 进行建模，那么根据贝叶斯观点，模型的不确定性就可以通过 $\mathbf{w}$ 的后验分布来表示。然而频率域的方法却是要基于数据集 $\mathcal{D}$ 对参数 $\mathbf{w}$ 进行点估计，并根据如下思路来解释估计的不确定性。假设有很多个大小为 N 的数据集，每个数据集都是相互独立地从分布 $p(t, \mathbf{x}))$ 中获取的。对于任意的数据集 $\mathcal{D}$ 都可以进行学习并得到相应的预测函数 $y(\mathbf{x}; \mathcal{D})$。这堆不同的数据集会各自给出不同的预测函数，所以相应也会得到不同的平方损失，然后根据这些数据集上的平均值确定学习算法的性能。

对于 (3.37) 中的第 1 项，假设数据集已经钦定为 $\mathcal{D}$，那么积分项中的第 1 项就变成了

$$\{y(\mathbf{x}; \mathcal{D}) - h(\mathbf{x})\}^2 \tag{3.38}$$

由于这一项的值与数据集 $\mathcal{D}$ 的选择有关，所以要在不同的数据集上取平均值。在大括号中加上一个 $\mathbb{E}_{\mathcal{D}}[y(\mathbf{x}; \mathcal{D})]$ 再马上减掉，并做点处理，可以得到

$$\begin{aligned}
&\{y(\mathbf{x}; \mathcal{D}) - \mathbb{E}_{\mathcal{D}}[y(\mathbf{x}; \mathcal{D})] + \mathbb{E}_{\mathcal{D}}[y(\mathbf{x}; \mathcal{D})] - h(\mathbf{x})\}^2 \\
=&\{y(\mathbf{x}; \mathcal{D}) - \mathbb{E}_{\mathcal{D}}[y(\mathbf{x}; \mathcal{D})]\}^2 + \{\mathbb{E}_{\mathcal{D}}[y(\mathbf{x}; \mathcal{D})] - h(\mathbf{x})\} \\
&+ 2\{y(\mathbf{x}; \mathcal{D}) - \mathbb{E}_{\mathcal{D}}[y(\mathbf{x}; \mathcal{D})]\}\{\mathbb{E}_{\mathcal{D}}[y(\mathbf{x}; \mathcal{D})] - h(\mathbf{x})\}
\end{aligned} \tag{3.39}$$

关于 $\mathcal{D}$ 求期望，此时第 1 项就会消失了，于是

$$\mathbb{E}_{\mathcal{D}}[\{y(\mathbf{x}; \mathcal{D}) - h(\mathbf{x})\}^2] = \underbrace{\{\mathbb{E}_{\mathcal{D}}[y(\mathbf{x}; \mathcal{D})] - h(\mathbf{x})\}^2}_{(\text{bias})^2} + \underbrace{\mathbb{E}_{\mathcal{D}}[\{y(\mathbf{x}; \mathcal{D}) - \mathbb{E}_{\mathcal{D}}[y(\mathbf{x}; \mathcal{D})]\}^2]}_{\text{variance}} \tag{3.40}$$

可以看出 $y(\mathbf{x}; \mathcal{D})$ 与回归函数 $h(\mathbf{x})$ 之间的期望平方差可以拆分成两项的和。第 1 项是偏差 (bias) 的平方，表示所有数据集的预测结果的平均值与回归函数之间的差异。

第 2 项是方差 (variance)，表示对于每一个数据集，模型给出的解在其平均值附近变化的程度，也就是函数 $y(\mathbf{x})$ 对于数据集选择的敏感程度。我们会通过一个简单的案例来解释这些概念。

到目前为止，我们研究的都是单一的输入 $\mathbf{x}$。如果将这个展开形式代入 (3.37) 中，可以得到期望平方损失的拆分形式：

$$\text{expected loss} = (\text{bias})^2 + \text{variance} + \text{noise} \tag{3.41}$$

其中

$$(\text{bias})^2 = \int \{\mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})] - h(\mathbf{x})\}^2 p(\mathbf{x})\, \mathrm{d}\mathbf{x} \tag{3.42}$$

$$\text{variance} = \int \mathbb{E}_{\mathcal{D}}[\{y(\mathbf{x};\mathcal{D}) - \mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})]\}^2] p(\mathbf{x})\, \mathrm{d}\mathbf{x} \tag{3.43}$$

$$\text{noise} = \int \{h(\mathbf{x}) - t\}^2 p(\mathbf{x}, t)\, \mathrm{d}\mathbf{x}\, \mathrm{d}t \tag{3.44}$$

现在偏差项和方差项的定义变成了积分后的值。

我们的目标是将期望损失最小化，而且已经把它拆分成了偏差的平方、方差和噪声常数项的和。我们即将会进行偏差与方差之间的制衡，低偏差高方差的模型会具有很高的灵活性，而高偏差低方差的模型会具有很高的稳定性。将偏差与方差完美均衡的模型就是具有最好预测性能的模型。这里还是用第 1 章中的正弦函数数据集说事。——**附录 A** 假设现在生成成了 100 个数据集，每个数据集包含 $N = 25$ 个数据，所有数据都是各自独立地从正弦曲线 $h(x) = \sin(2\pi x)$ 中获取的。数据集的编号为 $l = 1, ..., L, L = 100$，且对于所有的数据集 $\mathcal{D}^{(l)}$，都使用包含 24 个高斯基底函数的模型进行拟合，拟合的过程是通过将正则误差函数 (3.27) 进行最小化来完成的，从而给出预测函数 $y^{(l)}(x)$，如图 3.5 所示。其中，第 1 行的结果是通过一个较大的正则化系数 λ 得到的，这样的模型方差很小 (第 1 行左图中各个曲线相差不大)，但偏差很大 (第 1 行右图中与真实曲线相比差别非常大)。与之不同的是，第 3 行的结果是通过一个较小的正则化系数 λ 得到的，这样的模型方差很大 (第 3 行左图中各个曲线相差很大)，但偏差很小 (取平均值之后就与真实曲线很接近了)。需要注意的是，将这个复杂模型的 $M = 25$ 个解求平均值，可以对回归函数产生很好的拟合结果，说明求平均值是一个很不错的想法。实际上，对多个解求取加权平均值是贝叶斯方法的核心，不过这里的求平均值针对的是参数的后验分布而非数据集。

我们可以顺便利用这个例子研究一下偏差-方差的制衡。预测的平均值为

$$\overline{y}(x) = \frac{1}{L} \sum_{l=1}^{L} y^{(l)}(x) \tag{3.45}$$

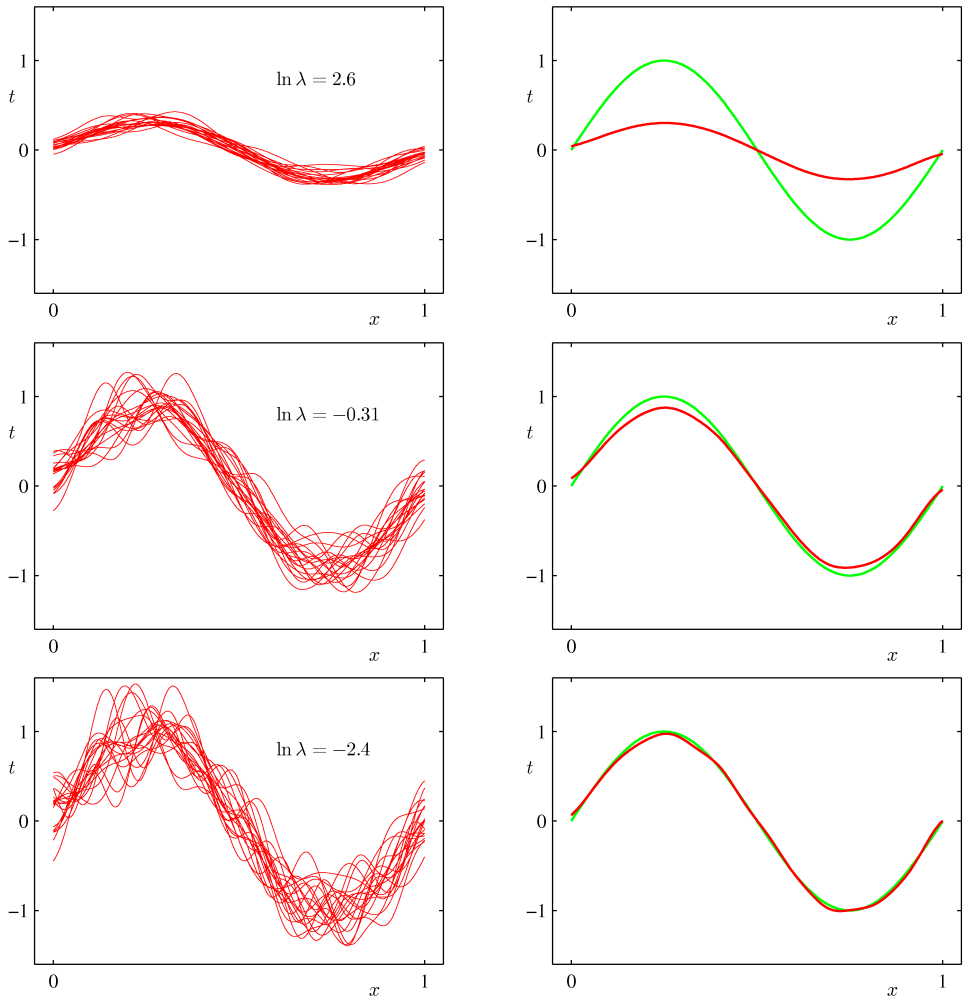

图 3.5: 偏差和方差与模型复杂度相关，模型的复杂度由参数 λ 控制，利用第 1 章中的正弦函数数据集得到的结果。数据集的数量 $L = 100$，每个数据集里包含 $N = 25$ 组数据，模型中包含 24 个高斯基底函数，算上偏差参数，参数的总数为 $M = 25$。左侧一列展示的是在不同 $\ln \lambda$ 下的模型拟合结果 (为了简洁起见并没有把 100 个拟合结果全画出来，而是只画了 20 个)。右侧一列展示的是相应产生的平均拟合结果 (红色曲线) 和真实的正弦函数曲线 (绿色曲线)。

经过积分的平方偏差和方差为

$$(\text{bias})^2 = \frac{1}{N}\sum_{n=1}^{N}\{\overline{y}(x_n) - h(x_n)\}^2 \tag{3.46}$$

$$\text{variance} = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{L}\sum_{l=1}^{L}\left\{y^{(l)}(x_n) - \overline{y}(x_n)\right\}^2 \tag{3.47}$$

其中关于 x 的积分都带有权重 $p(x)$，这个积分是通过对分布中取出的数据点集合进行求和来近似得到的。这些值和它们的和关于 $\ln\lambda$ 的函数图像如图 3.6 所示。可以看出，较小的 λ 会让模型对不同数据集中的噪声产生拟合，导致了较大的方差。与之相反，如果 λ 的值比较大，就会把权重参数向 0 拉近，形成较大的偏差。

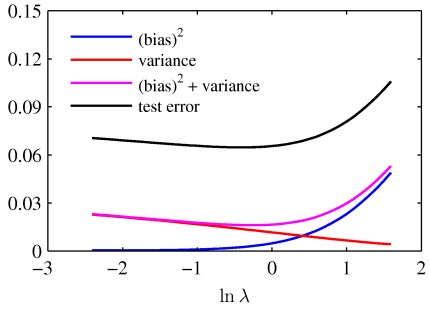

图 3.6: 平方偏差、方差及其总和的图像，这个结果是对应图 3.5 的。同时也展示了一个包含 1000 个数据的测试集的平均误差。平方偏差 + 方差会在 $\ln\lambda = -0.31$ 附近取得最小值，非常接近测试数据集上取得最小误差的位置。

尽管偏差-方差分解挺有意思，但实际应用中它的局限性相当明显，因为偏差-方差分解是要基于很多数据集求平均值的，然而在实际应用中，数据集可不一定有那么够用。如果训练集的数量非常大，我们也宁愿将它们全部汇总到一起，形成一个超大的数据集，因为这样做可以减少过拟合的出现。

在知晓这种方法的局限性后，我们接下来进行贝叶斯视角的线性基底函数模型分析，这种方法对过拟合问题进行了更加深刻的分析，并给出了解决模型复杂度问题的实用方法。

3.3 贝叶斯线性回归

我们从线性回归模型参数的最大似然方法中已经看出，模型的有效复杂度由基底函数的数量决定，而模型复杂度的选择应该取决于数据集的规模。在对数似然函数中加上一个正则项会影响到模型的有效复杂度，这样的做法使得模型复杂度可以由正则化系数来控制，不过基底函数的形式与数量仍然会对模型的性能产生重要的影响。

那么这就遗留了一个问题，如何对实际的问题确定适当的模型复杂度，而且不能简单地通过对似然函数进行最大化得到，因为这样的做法往往会得到一个复杂的模型，而且容易出现过拟合的问题。在第 1.3 节中我们提到过，利用单独预分出来的数据可以确定模型的复杂度，但这样的做法很浪费计算资源也很浪费训练数据。于是我们转向了另一个思路，即利用贝叶斯观点处理线性回归问题，从而避免最大似

然方法中可能出现的过拟合问题，而且可以仅通过训练数据就自动地确定模型的复杂度。为了方便起见，我们仍然针对一元目标变量 t 进行讨论。至于多元变量的情况，其实是可以由一元变量直接推广而来，可以参考一下第 3.1.5 节中的内容。

3.3.1 参数分布

我们从模型参数 $\mathbf{w}$ 的先验概率分布入手，开始讨论贝叶斯方法的线性回归。现在开始，姑且认为噪声精度参数 β 是一个已知的常数。首先需要注意到，(3.10) 中的似然函数 $p(\mathbf{t}|\mathbf{w})$ 是 $\mathbf{w}$ 的二次函数的指数。对应的共轭先验为如下的高斯分布，其均值为 $\mathbf{m}_0$，协方差为 $\mathbf{S}_0$：

$$p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\mathbf{m}_0, \mathbf{S}_0) \tag{3.48}$$

接下来计算后验分布，后验分布很明显是与先验和似然函数的乘积成正比的。由于选择的共轭先验就是高斯分布，所以后验分布一定同样是高斯分布。我们可以对指数进行常规的完成平方项来评估这个分布，并进行常规的归一化。——**习题 3.7** 不过我们以前已经做过这个工作，结果为 (2.116)，于是可以直接写出如下形式的后验分布

$$p(\mathbf{w}|\mathbf{t}) = \mathcal{N}(\mathbf{w}|\mathbf{m}_N, \mathbf{S}_N) \tag{3.49}$$

其中

$$\mathbf{m} = \mathbf{S}_N(\mathbf{S}_0^{-1}\mathbf{m}_0 + \beta\boldsymbol{\Phi}^{\mathrm{T}}\mathbf{t}) \tag{3.50}$$

$$\mathbf{S}_N^{-1} = \mathbf{S}_0^{-1} + \beta\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi} \tag{3.51}$$

需要注意的是，由于后验分布是高斯分布，所以其均值等于其模。所以最大后验权重向量为 $\mathbf{w}_{\mathrm{MAP}} = \mathbf{m}_N$。如果我们假设先验 $\mathbf{S}_0 = \alpha^{-1}\mathbf{I}$ 中的 $\alpha \to 0$，那么后验分布的均值 $\mathbf{m}_N$ 就退化成 (3.15) 中的最大似然值 $\mathbf{w}_{\mathrm{ML}}$。类似地，如果 $N = 0$，那么后验分布就与先验是完全相同的。此外，如果数据是以序列的形式到达的，那么任意一次概率更新中的后验分布都是下一个步骤的先验分布，然后再次得到 (3.49) 这样的后验分布。——**习题 3.8**

在本章剩余的内容中，我们将挑选一个特殊形式的高斯分布作为先验，从而简化推理的过程。0 均值且各向同性 (即协方差为 $\alpha^{-1}\mathbf{I}$) 的高斯分布是个不错的选择，因为这样的话，整个分布中将仅有一个参数 α，于是

$$p(\mathbf{w}|\alpha) = \mathcal{N}(\mathbf{w}|\mathbf{0}, \alpha^{-1}\mathbf{I}) \tag{3.52}$$

根据 (3.49) 可以写出对应的关于 $\mathbf{w}$ 的后验分布，具体的值为

$$\mathbf{m}_N = \beta\mathbf{S}_N\boldsymbol{\Phi}^{\mathrm{T}}\mathbf{t} \tag{3.53}$$

$$\mathbf{S}_N^{-1} = \alpha\mathbf{I} + \beta\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi} \tag{3.54}$$

后验分布的对数等于对数似然加上先验分布的对数，将其看成 $\mathbf{w}$ 的函数，于是有

$$\ln p(\mathbf{w}|\mathbf{t}) = -\frac{\beta}{2}\sum_{n=1}^{N}\left\{t_n - \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)\right\}^2 - \frac{\alpha}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} + \text{const} \tag{3.55}$$

于是，关于 $\mathbf{w}$ 求取后验分布的最大值这个问题，等价于对一个新函数，即平方和误差函数加上一个附加的二次正则项，进行最小化。与 (3.27) 比较一下，可以看出这里的 $\lambda = \alpha/\beta$。

我们可以利用一个简单的直线拟合问题来说明线性基底函数模型中的贝叶斯学习和后验分布的顺序更新。假设在线性模型 $y(x, \mathbf{w}) = w_0 + w_1 x$ 中，输入变量 x 和目标变量 t 均为一元变量。由于这个模型中只有两个可调节的参数，所以我们可以直接在参数空间中画出先验分布和后验分布。数据的来源是函数 $f(x, \mathbf{a}) = a_0 + a_1 x$，其中参数 $a_0 = -0.3$，$a_1 = 0.5$。首先从均匀分布 $U(x|-1, 1)$ 中选取 x_n，然后评估 $f(x_n, \mathbf{a})$，最后加上一个标准差为 0.2 的高斯噪声，从而得到目标变量 t_n。我们的目标是通过这样的数据来求取 a_0 和 a_1 的值，并研究结果对数据集规模的依赖性。这里已经假设噪声的方差是已知的，于是精度系数就是一个真实的值 $\beta = (1/0.2)^2 = 25$。类似地还要将参数 α 的值取为固定的 2.0。接下来简单讨论一下根据训练数据确定 α 和 β 的方法。如图 3.7 所示，随着数据集规模的增大时该模型的贝叶斯学习结果的变化情况，顺便展示了基于贝叶斯方法的顺序学习中，当前的后验分布就是下一步的先验分布。花点时间研究一下这些图是比较值得的，因为里面蕴含了很多贝叶斯推断的重要思想。第一行的图像表示在还没有拿到数据时 $\mathbf{w}$ 空间中的先验分布，其中展示了 6 个 $y(x, \mathbf{w})$ 的示例，$\mathbf{w}$ 的值是从先验中获取的。第二行的图像表示在拿到 1 组数据后的情况，数据位置 (x, t) 为右侧图像中的蓝色圆圈，左侧图像则是这个数据点对应的似然函数 $p(t|x, \mathbf{w})$，当然是表示为 $\mathbf{w}$ 的函数的形式了。需要注意的是，似然函数事实上给出了一个软约束，那就是它必须要很接近自身对应的数据点，活动的范围则由噪声精度 β 确定。作为对比，在图 3.7 的左侧一栏中用白色的"+"表示出了真实的参数 $a_0 = -0.3$ 和 $a_1 = 0.5$。如果将第一行的先验分布乘以这个似然函数并进行归一化，得到的结果就是第二行中间图像所示的后验分布。通过在后验分布中抽取 $\mathbf{w}$ 样本得到的回归函数 $y(x, \mathbf{w})$ 如第二行右侧的图所示。第三行展示的是第二组数据对分布形成的影响，同样在右侧图中表示为蓝色的圆圈。对应的似然函数如左侧图中所示。如果将第二行的后验分布乘以这个似然函数并进行归一化，就会得到第三行中间图像所示的后验分布。需要注意的是，如果将初始的先验分布乘以这两组数据各自的似然函数，将会得到相同的结果。现在的后验分布是经过两次更新得到的了，而两点可以确定一条直线，所以现在得到的后验分布已经有点意思了。根据这个后验分布得到的一些函数样本如图中第三行中右侧图像所示，可以看出这些函数都很靠近这两个数据点。第四行展示了经过 20 个数据点更新后的情况。

左侧图像展示了第 20 个数据点对应的似然函数，中间图像展示了逐一利用这 20 个观测进行后验分布更新之后得到的结果。与第三行比起来，现在的后验分布更加尖锐了。如果数据集是无限的，那么后验分布会成为一个中心位于真实参数值 (即白色的"+") 处的 delta 函数。

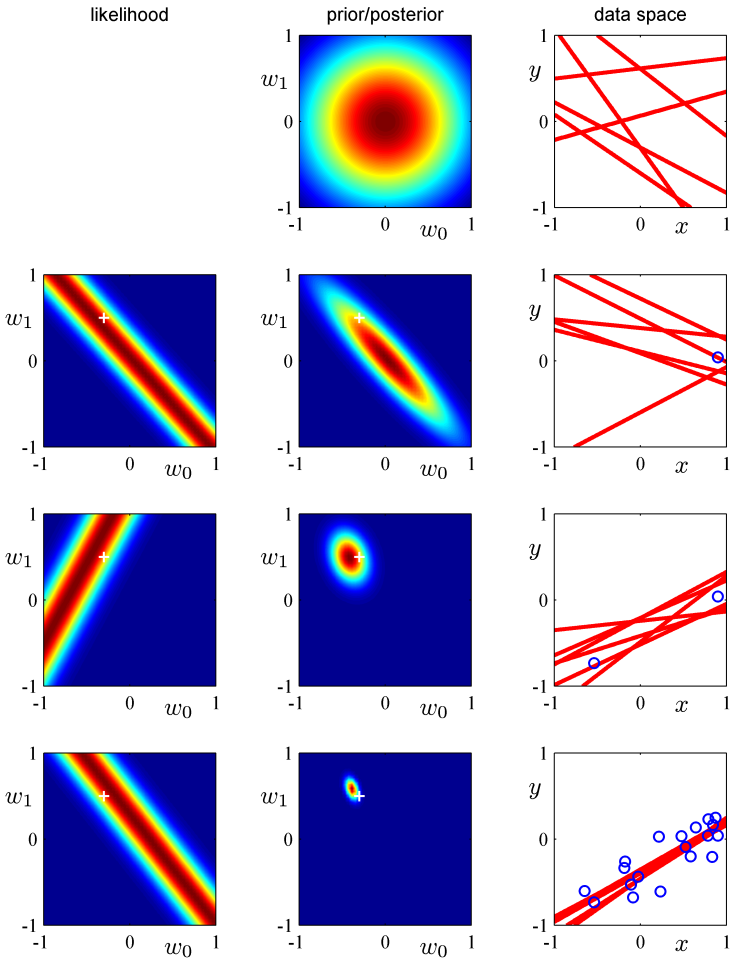

图 3.7: 对于线性模型 $y(x, \mathbf{w}) = w_0 + w_1 x$ 进行顺序贝叶斯学习的示意图。每个图像对应的具体含义请详见文字部分。

此外，参数的先验可以选择为另外一些形式。举例而言，可以是广义的高斯先验

$$p(\mathbf{w}|\alpha) = \left[\frac{q}{2}\left(\frac{\alpha}{2}\right)^{1/q}\frac{1}{\Gamma(1/q)}\right]^M \exp\left(-\frac{\alpha}{2}\sum_{j=0}^{M-1}|w_j|^q\right) \tag{3.56}$$

其中，如果 $q = 2$ 就是正常的高斯先验了，而且仅有在这种情况下，先验才与似然函数 (3.10) 为共轭关系。关于 $\mathbf{w}$ 求取后验分布的最大值等价于对正则化误差函数

(3.29) 进行最小化。对于高斯先验，后验分布的模等于均值，但如果 $q \neq 2$，这条性质就不成立了。

3.3.2 预测分布

在实际应用中，其实往往不太关心 $\mathbf{w}$ 的值，而是更关注对于 $\mathbf{x}$ 给出的预测 t 究竟如何。这就要求我们评估预测分布 (predictive distribution)

$$p(t|\mathbf{t}, \alpha, \beta) = \int p(t|\mathbf{w}, \beta)p(\mathbf{w}|\mathbf{t}, \alpha, \beta)\ \mathrm{d}\mathbf{w} \tag{3.57}$$

其中 $\mathbf{t}$ 为训练集中的目标值向量，并省略了等号右侧条件概率中的输入向量。目标变量的条件分布 $p(t|\mathbf{x}, \mathbf{w}, \beta)$ 由 (3.8) 给出，后验权重分布有 (3.49) 给出。可以看出 (3.57) 包含了两个高斯分布的卷积，根据第 2.3.3 节中的 (2.115)，可以看出预测分布的形式为——**习题 3.10**

$$p(t|\mathbf{x}, \mathbf{t}, \alpha, \beta) = \mathcal{N}(t|\mathbf{m}_N^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}), \sigma_N^2(\mathbf{x})) \tag{3.58}$$

其中，预测分布的方差 $\sigma_N^2(\mathbf{x})$ 为

$$\sigma_N^2(\mathbf{x}) = \frac{1}{\beta} + \boldsymbol{\phi}(\mathbf{x})^{\mathrm{T}}\mathbf{S}_N\boldsymbol{\phi}(\mathbf{x}) \tag{3.59}$$

(3.59) 中的第 1 项表示数据中的噪声，第 2 项则表示了与 $\mathbf{w}$ 有关的不确定性。由于噪声与 $\mathbf{w}$ 的分布是相互独立的高斯分布，所以它们是可加的。需要注意的是，随着我们获取到新的数据，后验分布会变得越来越狭窄。可以证明 $\sigma_{N+1}^2(\mathbf{x}) \leqslant \sigma_N^2(\mathbf{x})$(Qazaz et al., 1997)。——**习题 3.11** 取极限 $N \to \infty$，(3.59) 中的第 2 项将趋近于 0，于是预测分布的方差就只与参数 β 控制的噪声有关了。

现在回到第 1.1 节中正弦数据集拟合案例中，演示一下贝叶斯线性回归模型中的预测分布。如图 3.8 所示，我们可以利用高斯基底函数的线性组合来构建拟合模型，而且可以随着数据集规模的变化调整拟合的方式。其中，绿色的曲线表示函数 $sin(2\pi x)$，也就是真实的函数，当然是带有附加的高斯噪声的。数据点在图中用蓝色圆圈表示，当数据集的规模为 $N = 1, N = 2, N = 4, N = 25$ 时，红色曲线表示对应的高斯预测分布的均值，红色的阴影区域表示在均值两侧加上一个标准差的范围之后得到的区域。需要注意的是，预测的不确定性取决于 x，而且是数据点附近时取得最小值。此外，不确定性会随着数据点的增加而减小。

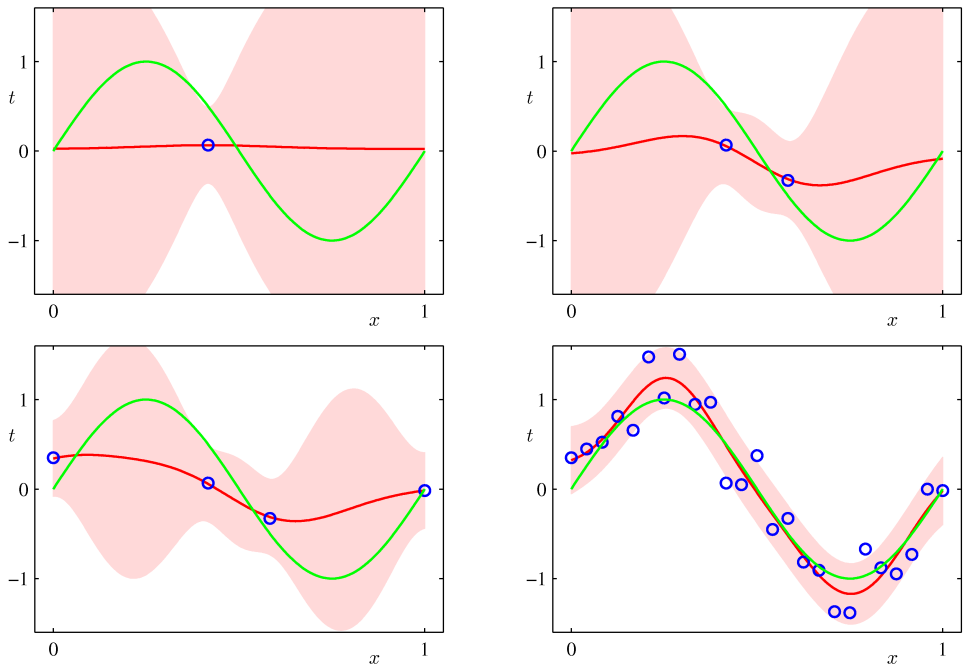

图 3.8: 根据第 1.1 节中的正弦函数数据集和一个包含 9 个高斯基底函数 (3.4) 的模型构建的预测分布 (3.58)。具体的讨论详见正文。

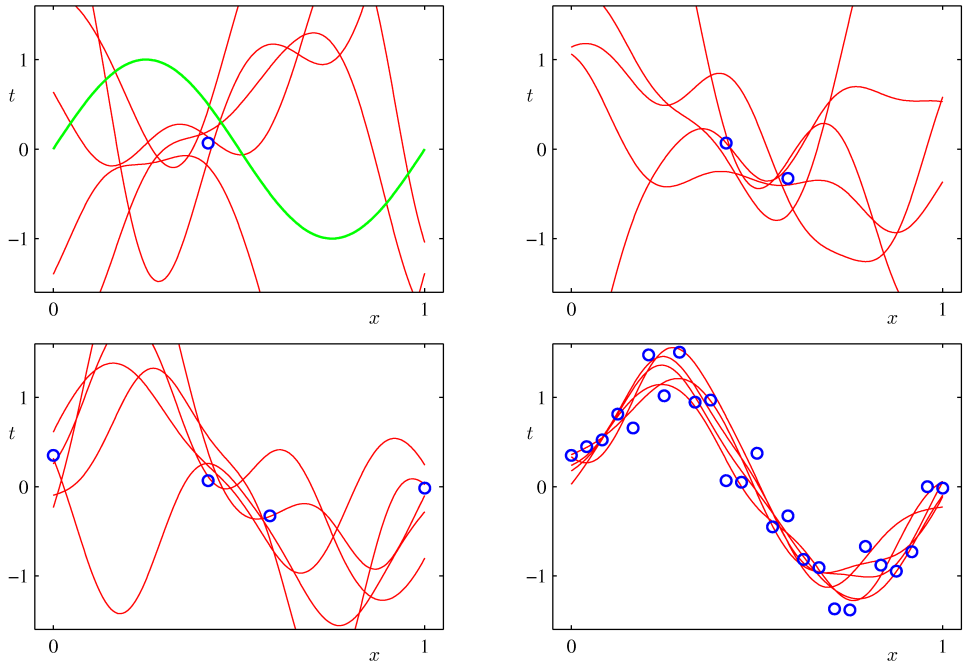

图 3.9: 从图 3.8 所示的分布中抽取 $\mathbf{w}$ 后画出的对应函数 $y(x, \mathbf{w})$ 的图像。

图 3.8 中的图像仅仅展示了每个数据点处预测方差与 x 的函数关系。为了更加深刻地了解对于不同 x 进行预测时产生的协方差，我们从 $\mathbf{w}$ 的后验分布中抽取一些样本，并画出对应的函数 $y(x, \mathbf{w})$，如图 3.9 所示。

如果我们利用局部基底函数，比如局部高斯基底函数，那么在远离基底函数中心的区域，预测方差 (3.59) 中的第 2 项将趋近于 0，于是只剩下了噪声造成的影响 β^{-1}。所以模型对自身预测的信心将大幅增加，不过这个结果并不是什么好事。一般情况下要使用高斯过程 (Gaussian process) 来回避这个问题，高斯过程是另一种贝叶斯方法。——**第 6.4 节**

需要注意的是，如果 $\mathbf{w}$ 和 β 都是未知的，那么可以先引入一个共轭先验分布 $p(\mathbf{w}, \beta)$，根据第 2.3.6 节，这是一个高斯-gamma 分布 (Denison et al., 2002)。——**习题 3.12** 在这种情况下，预测分布将是一个学生 t 分布。——**习题 3.13**

3.3.3 等价核

线性基底函数模型的后验均值解 (3.53) 有一个奇妙的解释，而且这种解释可以为接下来的核方法 (kernel method) 铺路，核方法中有一项就是高斯过程。——**第 6 章** 如果将 (3.53) 代入到 (3.3) 中，可以看出预测均值的形式为

$$y(\mathbf{x}, m_N) = \mathbf{m}_N^{\mathrm{T}} \boldsymbol{\phi}(\mathbf{x}) = \beta \boldsymbol{\phi}(\mathbf{x})^{\mathrm{T}} \mathbf{S}_N \boldsymbol{\Phi}^{\mathrm{T}} \mathbf{t} = \sum_{n=1}^{N} \beta \boldsymbol{\phi}(\mathbf{x})^{\mathrm{T}} \mathbf{S}_N \boldsymbol{\phi}(\mathbf{x}_n) t_n \tag{3.60}$$

其中 $\mathbf{S}_N$ 就是 (3.51) 定义的那个。所以在 $\mathbf{x}$ 处预测分布的均值就变成了训练集目标变量 t_n 的线性组合，于是

$$y(\mathbf{x}, \mathbf{m}_N) = \sum_{n=1}^{N} k(\mathbf{x}, \mathbf{x}_n) t_n \tag{3.61}$$

其中函数

$$k(\mathbf{x}, \mathbf{x}') = \beta \boldsymbol{\phi}(\mathbf{x})^{\mathrm{T}} \mathbf{S}_N \boldsymbol{\phi}(\mathbf{x}') \tag{3.62}$$

称为平滑矩阵 (smoother matrix) 或等价核 (equivalent kernel)。类似这样的可以通过训练集的目标变量值进行预测的回归函数称为线性平滑器 (linear smoother)。需要注意的是，等价核依赖于输入量 $\mathbf{x}_n$，因为它们在 $\mathbf{S}_N$ 中出现了。如图 3.10 所示的是以高斯基底函数为例的等价核，其中将核函数 $k(x, x')$ 画成了 3 种不同 x 取值下关于 x' 的函数。可以看出曲线一直在 x 附近波动，所以在 x 处预测分布的均值 $y(x, \mathbf{m}_N)$ 就可以通过构建目标变量的加权线性组合来确定，其中，距离 x 较近的数据点会被赋予较高的权重，反之亦然。从直觉上来说这似乎是挺合理的，毕竟远亲不如近邻【虽然这话放在这好像不很贴切，就那么回事吧】。需要注意的是，这个性质不仅适用于局部高斯基底函数，而是同样适用于非局部多项式和 sigmoid 基底函数，

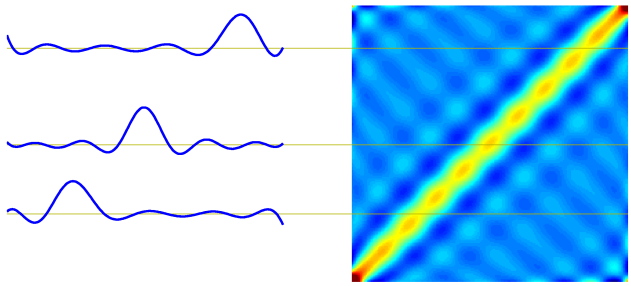

图 3.10: 图 3.1 中的高斯基底函数的等价核 $k(x, x')$，在这里画成了 $x - x'$ 的形式，同时给出了该矩阵的 3 个不同 x 取值处的具体情况。生成这个核所用的数据 x 是从 $(-1, 1)$ 之间均匀取出的 200 个值。

图 3.11: $x = 0$ 时的等价核，这里画成了 x 的函数的形式，左侧为多项式基底函数，右侧为 sigmoid 基底函数。需要注意的是，尽管它们对应的基底函数并非是局部的，它们也是关于 x 的局部函数。

如图 3.11 所示。

关于等价核这个问题，可以通过研究 $y(\mathbf{x})$ 与 $y(\mathbf{x}')$ 之间的协方差来获取更进一步的认知。这个协方差为

$$\begin{aligned}\operatorname{cov}[y(\mathbf{x}), y(\mathbf{x}')] &= \operatorname{cov}\left[\boldsymbol{\phi}(\mathbf{x})^{\mathrm{T}}\mathbf{w}, \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}')\right] \\ &= \boldsymbol{\phi}(\mathbf{x})^{\mathrm{T}}\mathbf{S}_N\boldsymbol{\phi}(\mathbf{x}') = \beta^{-1}k(\mathbf{x}, \mathbf{x}')\end{aligned} \tag{3.63}$$

其中用到了 (3.49) 和 (3.62)。从等价核的形式可以看出，距离较近的点也具有较强的相关性，而距离较远的点则相反。

图 3.8 中的预测分布使得我们可以看到预测中每个点的不确定性，这里的不确定性是由 (3.59) 确定的。通过从后验分布中抽取一些 $\mathbf{w}$ 的样本，并画出图 3.9 这样的对应模型函数 $y(\mathbf{x}, \mathbf{w})$，我们可以看到后验分布中 y 值与两个 (或多个)x 值之间的联合不确定性，这个不确定性是由等价核确定的。

核函数形式的线性回归提出了如下的求解回归问题的替代方法。这里并不需要扯进一大堆基底函数，因为基底函数实际上也构建了一个等价核，而是要直接定义一个局部核，并用它来对新的输入 $\mathbf{x}$ 做预测，当然是建立在一个训练集的基础上了。这个做法构建了一种求解回归问题 (或分类问题) 方法的重要框架，这个方法称为高斯过程，具体内容详见第 6.4 节。

我们已经看到等价核根据训练集中的目标变量值定义了权重，从而对新的 $\mathbf{x}$ 做出预测，而且这些权重的总和一定为 1，即——习题 3.14

$$\sum_{n=1}^{N} k(\mathbf{x}, \mathbf{x}_n) = 1 \tag{3.64}$$

这个结果引发强烈的舒适，如果不想太正式地证明，也可以用如下方法：这个求和等价于对一个 t_n 全部等于 1 的数据集求预测均值 $\hat{y}(\mathbf{x})$。由于基底函数都是线性无关的，所以数据的数量一定大于基底函数的数量，而且一定有一个基底函数是常数 (这样才有和偏差参数对应的基底函数)，那就一定可以对训练数据进行拟合，而且预测均值一定是 $\hat{y}(\mathbf{x}) = 1$，于是就得到了 (3.64)。需要注意的是，核函数可正可负，所以尽管它们满足上述的总和约束，但对应的预测并不一定是训练集中目标变量的凸组合。

最后还有点事情需要注意，等价核 (3.62) 满足一个一般的核函数都满足的重要性质，即它们可以写成非线性函数 $\boldsymbol{\psi}(\mathbf{x})$ 内积的形式：——第 6 章

$$k(\mathbf{x}, \mathbf{z}) = \boldsymbol{\psi}(\mathbf{x})^{\mathrm{T}} \boldsymbol{\psi}(\mathbf{z}) \tag{3.65}$$

其中 $\boldsymbol{\psi}(\mathbf{x}) = \beta^{1/2} \mathbf{S}_N^{1/2} \boldsymbol{\phi}(\mathbf{x})$。

3.4 贝叶斯模型的对比

在第 1 章中我们曾强调了过拟合这个问题，并提出利用交叉验证方法来设定正则项参数或进行模型的选择。选择我们研究一下贝叶斯视角下的模型选择问题。在这一节中，我们所进行的讨论会比较针对一般情况，在第 3.5 节中我们再集中精力去解决在线性回归中确定正则化参数的问题。

我们即将开拿到，最大似然方法中的过拟合问题可以通过对模型参数进行边缘化 (离散变量是求和，连续变量是积分) 来解决，相比之下，点估计的方法则很容易出问题。模型的比较可以直接在训练集上进行，而不需要独立的验证集。这就使得所有的数据都可以用于训练，而且避免了交叉验证的问题——毕竟交叉验证要多次训练，很是麻烦。而且这样的方法还可以在训练过程中顺便同时确定复杂度参数。举个例子，在第 7 章中我们会介绍相关向量机 (relevance vector machine)，这是一个典型的贝叶斯模型，对每组训练数据都有一个复杂度参数。

贝叶斯视角下的模型对比仍然是使用概率来表示模型选择的不确定性，当然，加法规则和乘法规则是必不可少的。假设希望比较的模型组成了一个规模为 L 的模型集合 $\{\mathcal{M}_i\}$，其中 $i = 1, ..., L$。这里的模型指的是观测数据 $\mathcal{D}$ 上的概率分布。在多项式曲线拟合问题中，这个分布就是建立在目标变量集合 $\mathbf{t}$ 上的，并认为输入变量集合 $\mathbf{X}$ 是已知的。另一种模型是 $\mathbf{X}$ 和 $\mathbf{t}$ 的联合分布。——第 1.5.4 节 我们假设数

据是从其中的某一个模型中获取的，但又不确定究竟是哪一个。不确定性是通过先验概率分布 $p(\mathcal{M}_i)$ 来表示的。在给定了训练集 $\mathcal{D}$ 的条件下，接下来我们希望评价后验分布

$$p(\mathcal{M}_i|\mathcal{D}) \propto p(\mathcal{M}_i)p(\mathcal{D}|\mathcal{M}_i) \tag{3.66}$$

先验表示我们对不同模型的偏好程度。现在先假设所有模型的先验概率都是一样的。其实最主要的部分是模型证据 (model evidence)$p(\mathcal{D}|\mathcal{M}_i)$，这一项表示的是数据对不同模型所表现出来的” 偏好程度”，接下来更加深入地聊聊这个问题。模型证据有时候也被称为边缘似然 (marginal likelihood)，因为它可以被视为是一个模型空间中的似然函数，在这个” 似然函数” 中，所有的参数都已经被边缘化掉了。两个模型的模型证据的比值 $p(\mathcal{D}|\mathcal{M}_i)/p(\mathcal{D}|\mathcal{M}_j)$ 称为贝叶斯因子 (Bayesian factor, Kass and Raftery, 1995)。

模型的后验分布一旦确定，那么根据加法规则和乘法规则，预测分布为

$$p(t|\mathbf{x}, \mathcal{D}) = \sum_{i=1}^{L} p(t|\mathbf{x}, \mathcal{M}_i, \mathcal{D})p(\mathcal{M}_i|\mathcal{D}) \tag{3.67}$$

这是一个混合分布 (mixture distribution)，其中，通过对预测分布 $p(t|\mathbf{x}, \mathcal{M}_i, \mathcal{D})$ 求加权平均值，且权重为每个模型各自的后验概率 $p(\mathcal{M}_i|\mathcal{D})$，从而确定最终的整体预测分布。举例而言，假设有两个模型，这两个模型的后验概率相同，一个是在 $t = a$ 附近，另一个在 $t = b$ 附近，那么最终的整体预测分布就是一个双峰分布，一个峰位于 $t = a$，另一个位于 $t = b$，而非一个 $t = (a + b)/2$ 的单峰分布。

对模型取平均值有一个简单的近似方法，那就是直接将可能性最大的模型作为最终结果，这个问题就是模型选择 (model selection) 问题。

对于一个由参数 $\mathbf{w}$ 控制的模型，根据加法规则和乘法规则，模型证据为

$$p(\mathcal{D}|\mathcal{M}_i) = \int p(\mathcal{D}|\mathbf{w}, \mathcal{M}_i)p(\mathbf{w}|\mathcal{M}_i)\ \mathrm{d}\mathbf{w} \tag{3.68}$$

从采样方法的角度来看，——**第 11 章** 边缘似然可以看成是能够从模型中得到数据集 $\mathcal{D}$ 的概率，而参数则是从先验中随机抽取出来的。另一方面，” 证据” 恰好又是估计参数的后验分布时贝叶斯定理分母中的归一化项。因为

$$p(\mathbf{w}|\mathcal{D}, \mathcal{M}_i) = \frac{p(\mathcal{D}|\mathbf{w}, \mathcal{M}_i)p(\mathbf{w}|\mathcal{M}_i)}{p(\mathcal{D}|\mathcal{M}_i)} \tag{3.69}$$

我们可以通过对参数的积分进行一个简单的近似，从而更加深刻地认识模型证据的含义。假设模型仅有一个参数 w。参数的后验分布是与 $p(\mathcal{D}|w)p(w)$ 成正比的，其中忽略了依赖项中的 $\mathcal{M}_i$ 从而简化符号。如果我们假设后验分布是一个在最大似然值 w_{MAP} 处的尖峰，其宽度为 $\Delta w_{\mathrm{posterior}}$，于是就可以将积分进行近似为被积函数

的函数值与尖峰宽度的乘积。如果进一步假设先验分布是平坦的，且宽度为 Δw_{prior}，那么 $p(w) = 1/\Delta w_{\text{prior}}$，于是

$$p(\mathcal{D}) = \int p(\mathcal{D}|w)p(w)\,\mathrm{d}w \approx p(\mathcal{D}|w_{\text{MAP}})\frac{\Delta w_{\text{posterior}}}{\mathrm{w}_{\text{prior}}} \tag{3.70}$$

对两侧同时取对数，则有

$$\ln p(\mathcal{D}) \approx \ln p(\mathcal{D}|w_{\text{MAP}}) + \ln\left(\frac{\Delta w_{\text{posterior}}}{\Delta w_{\text{prior}}}\right) \tag{3.71}$$

这个近似的过程如图 3.12 所示。第 1 项表示的是概率最大的参数值给出的数据拟合结果，对于一个平坦先验分布而言，与之对应的就是对数似然。第二项根据模型的复杂度进行模型惩罚。由于 $\Delta w_{\text{posterior}} < \Delta w_{\text{prior}}$，所以这一项是负值，而且随着这个比值的减小，其绝对值会增加。所以，如果取得某种参数下的后验分布对数据的拟合效果非常好，那么惩罚项也会相应变得非常大。

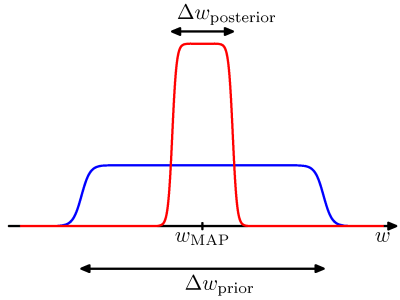

图 3.12: 如果我们假设参数的后验分布是在其模 w_{MAP} 附近的尖峰，可以获得如图所示的模型证据粗略近似。

对于带有 M 个参数的模型，可以对每个参数都进行类似的近似。假设所有的参数的 $\Delta w_{\text{posterior}}/\Delta w_{\text{prior}}$ 都相同，可以得到

$$\ln p(\mathcal{D}) \approx \ln p(\mathcal{D}|\mathbf{w}_{\text{MAP}}) + M\ln\left(\frac{\Delta w_{\text{posterior}}}{\Delta w_{\text{prior}}}\right) \tag{3.72}$$

所以在这个非常简单的近似中，复杂度惩罚项会随着模型中可调节参数的数量 M 线性增加。随着我们增加模型的复杂度，第 1 项通常会增大，因为复杂模型往往能够产生更好的拟合效果，而第 2 项则会相应减小。最优的模型复杂度是由最大模型证据给出的，而这个复杂度是两个竞争项之间的制衡。随后我们会给出一个更准确的近似，该方法是基于对后验分布进行高斯近似的。——第 4.4.1 节

通过图 3.13 我们可以进一步理解贝叶斯模型的对比，以及边缘似然是如何促成了中等复杂度的模型。其中，横轴表示的是数据集空间的一维表示，所以横轴上每个点都表示一个数据集。现在我们研究 3 个模型 $\mathcal{M}_1$，$\mathcal{M}_2$ 和 $\mathcal{M}_3$，3 个模型的复杂度

依次增加。假设现在把 3 个模型挨个跑了一遍，然后把生成的数据集分布结果拎出来。对于任意的模型都可以生成一系列不同的数据集，因为模型的参数是由先验概率分布控制的，而不管怎样选择参数，目标变量中都会带有随机噪声。为了对某个特定的模型生成某个特定的数据集，首先从参数的先验分布 $p(\mathbf{w})$ 中选择参数值，然后对于该参数值，从 $p(\mathcal{D}|\mathbf{w})$ 中抽取数据。类似于一阶多项式这样的简单模型可变性极小，所以生成的数据集区别也不会很大。因此其分布 $p(\mathcal{D})$ 也就被限制在了水平方向上较小的区域中。与之相反，类似于九阶多项式这样的复杂模型生成的数据集变化相当大，所以分布 $p(\mathcal{D})$ 覆盖了横轴相当大的一块区域。由于分布 $p(\mathcal{D}|\mathcal{M}_i)$ 是归一化的，所以某个特定的数据集 $\mathcal{D}_0$ 会在具有中等复杂度的模型上取得最高的模型证据值。从本质上来说，太简单的模型拟合效果肯定不好，太复杂的模型又会造成桃李满天下还必须得雨露均沾的尴尬局面，所以搞得每个数据集都分不到太多的概率。

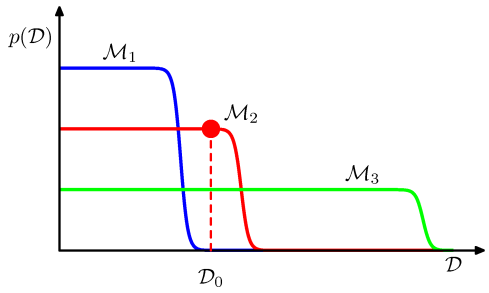

图 3.13: 三种不同复杂度的模型各自的数据集分布，$\mathcal{M}_1$ 是复杂度最低的，$\mathcal{M}_3$ 的复杂度最高的。这些分布都是经过归一化的。在这个例子中，对于某个数据集 $\mathcal{D}_0$，中等复杂的模型 $\mathcal{M}_2$ 给出的模型证据最大。

贝叶斯模型的比较中隐含了一个假设，那就是真实的分布 (数据的真实来源) 存在于模型集合中。如果事实如此，我们可以证明贝叶斯模型的对比是倾向于正确模型的。为了证明这个问题，假设有两个模型 $\mathcal{M}_1$ 和 $\mathcal{M}_2$，正确的模型是 $\mathcal{M}_1$。对于一个给定的有限数据集，错误的模型会产生较大的贝叶斯因子。然而，如果我们在整个数据集分布上对贝叶斯因子求平均数，就可以得到期望贝叶斯因子：

$$\int p(\mathcal{D}|\mathcal{M}_1) \ln \frac{p(\mathcal{D}|\mathcal{M}_1)}{p(\mathcal{D}|\mathcal{M}_2)} \, \mathrm{d}\mathcal{D} \tag{3.73}$$

其中已经关于数据的真实分布求取了平均值。这个值是 KL 距离的典型案例，——**第 1.6.1 节** 它满足一条性质，即只要两个分布不相等，这个距离就一定为正，当且仅当两个分布完全相同时，这个距离为 0。所以一般情况下，贝叶斯因子始终倾向于正确的模型。

我们已经可以看出，贝叶斯学习可以避免过拟合的问题，而且仅基于训练数据就可以进行模型之间的对比。然而，和其他的模式识别方法一样，贝叶斯方法也需要

对模型的形式做出假设。如果这些假设是不合理的，那么结果也会非常糟糕。特别地，从图 3.12 可以看出，模型证据对先验的诸多方面都很是神经过敏，尤其是拖尾的部分。实际上，如果先验是反常的，那么模型证据就是无意义的。一个反常的先验可以有任意的缩放因子，换句话说，这个分布压根就不可能被归一化，所以归一化参数是没意义的，所以模型证据也是无意义的。如果对一个适当的先验取适当的极限，这样得到的反常先验 (比如高斯先验带上一个无限大的方差这样的反常先验) 会使得模型证据趋向于 0，这点可以从 (3.70) 和图 3.12 看出来。不过即使是这种情况，也可以首先求取两个模型的模型证据比值，然后取极限，从而得到一个可用的结果。

所以在实际应用中，首先保留一个独立的测试数据集是比较明智的，因为这样可以评估最终得到的系统的整体表现。

3.5 证据近似

在完整的贝叶斯方法下的线性基底函数模型中，我们会引入关于超参数 α 和 β 的先验分布，并对这些超参数以及参数 $\mathbf{w}$ 进行边缘化，从而进行预测。然而，尽管在理论上我们可以对 $\mathbf{w}$ 和超参数们进行边缘化，但对这些参数进行彻底边缘化的计算代价是相当大的。现在我们要研究是一种近似方法，在这个方法中，我们通过对边缘似然函数 (marginal likelihood function) 进行最小化，将超参数设定为特定的值。这里的边缘似然函数是首先对参数 $\mathbf{w}$ 积分得到的。这个办法在统计学领域称为经验贝叶斯方法 (empirical Bayes, Bernardo and Smith, 1994; Gelman et al., 2004) 或者第二型最大似然方法 (type 2 maximum likelihood, Berger, 1985)，亦或称为广义最大似然方法 (generalized maximum likelihood, Wahba, 1975)，不过在机器学习领域，一般称为证据近似 (evidence approximation, Gull, 1989; MacKay, 1992a)。

假设现在引入 α 和 β 的超先验，那么对 $\mathbf{w}$，α 和 β 进行边缘化即可得到预测分布

$$p(t|\mathbf{t}) = \iiint p(t|\mathbf{w},\beta)p(\mathbf{w}|\mathbf{t},\alpha,\beta)p(\alpha,\beta|\mathbf{t})\,\mathrm{d}\mathbf{w}\,\mathrm{d}\alpha\,\mathrm{d}\beta \tag{3.74}$$

其中 $p(t|\mathbf{w},\beta)$ 为 (3.8) 所定义的，$p(\mathbf{w}|\mathbf{t},\alpha,\beta)$ 为 (3.49) 所定义的，其中的 $\mathbf{m}_N$ 和 $\mathbf{S}_N$ 为 (3.53) 和 (3.54) 所定义的。为了简化符号，现在开始省略依赖项中的输入变量 $\mathbf{x}$。如果后验分布 $p(\alpha,\beta|\mathbf{t})$ 是 $\hat{\alpha}$ 和 $\hat{\beta}$ 附近的尖峰，那么就可以将 α 和 β 设定为固定的 $\hat{\alpha}$ 和 $\hat{\beta}$，于是只对 $\mathbf{w}$ 进行边缘化就可以得到预测分布：

$$p(t|\mathbf{t}) \approx p(t|\mathbf{t},\hat{\alpha},\hat{\beta}) = \int p(t|\mathbf{w},\hat{\beta})p(\mathbf{w}|\mathbf{t},\hat{\alpha},\hat{\beta})\,\mathrm{d}\mathbf{w} \tag{3.75}$$

根据贝叶斯定理，α 和 β 的后验分布为

$$p(\alpha,\beta|\mathbf{t}) \propto p(\mathbf{t}|\alpha,\beta)p(\alpha,\beta) \tag{3.76}$$

如果先验比较平坦，那么在模型证据中，可以通过对边缘似然函数 $p(\mathbf{t}|\alpha,\beta)$ 进行最大化来获得 $\hat{\alpha}$ 和 $\hat{\beta}$。接下来我们会评估线性基底函数模型的边缘似然函数，并求其最大值。这样的做法使得我们仅通过训练数据就可以确定超参数的值，而不需要交叉验证这样的步骤。回想一下，比值 α/β 是类似于正则化参数的。

此外还有一点值得注意，如果我们定义了 α 和 β 的共轭先验分布 (Gamma 先验分布)，那么对于这些超参数根据 (3.74) 进行边缘化，得到的结果事实上就是一个关于 $\mathbf{w}$ 的学生 t 分布，相关的内容详见第 2.3.7 节。尽管关于 $\mathbf{w}$ 的积分不再有解析形式，但对这个积分求近似也会给出模型证据的替代结果 (Buntine and Weigend, 1991)，比如即将在第 4.4 节中讨论的 Laplace 近似。Laplace 近似是基于局部高斯近似的近似方法，这里的局部高斯近似是以后验概率分布的模为中心的近似。然而，被积函数是 $\mathbf{w}$ 的函数，而它的模往往不太准确，所以 Laplace 近似往往无法准确描述概率质量函数中的信息，这可能导致最终的结果比证据最大化方法得到的结果还差 (MacKay, 1999)。

回到模型证据的讨论中来，我们注意到有两种将对数模型证据进行最大化的方法。一方面，我们可以解析地评估证据函数并令其导数为 0，从而得到新的 α 和 β 的估计结果，这个方法我们将在第 3.5.2 节中进行讨论；另一方面，我们可以利用即将在第 9.3.4 节中讨论的 EM 算法，到时候我们还会证明这两种方法事实上会收敛到同一个解。

3.5.1 证据函数的评估

边缘似然函数 $p(\mathbf{t}|\alpha,\beta)$ 是通过对权重参数 $\mathbf{w}$ 进行边缘化得到的，于是

$$p(\mathbf{t}|\alpha,\beta)=\int p(\mathbf{t}|\mathbf{w},\beta)p(\mathbf{w}|\alpha)\,\mathrm{d}\mathbf{w} \tag{3.77}$$

评估这个积分，一种方法是再次利用线性高斯模型的条件分布的结论 (2.115)。——**习题 3.16** 现在我们通过对指数项进行完成平方项，并利用高斯函数的归一化系数的标准形式，评估这个积分。

根据 (3.11)，(3.12) 和 (3.52)，可以将证据函数写成如下形式：——**习题 3.17**

$$p(\mathbf{t}|\alpha,\beta)=\left(\frac{\beta}{2\pi}\right)^{N/2}\left(\frac{\alpha}{2\pi}\right)^{M/2}\int \exp\{-E(\mathbf{w})\}\,\mathrm{d}\mathbf{w} \tag{3.78}$$

其中 M 为向量 $\mathbf{w}$ 的维度，以及

$$\begin{aligned}E(\mathbf{w})&=\beta E_D(\mathbf{w})+\alpha E_W(\mathbf{w})\\&=\frac{\beta}{2}\|\mathbf{t}-\mathbf{\Phi}\mathbf{w}\|^2+\frac{\alpha}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w}\end{aligned} \tag{3.79}$$

可以看出 (3.79) 修改一下系数就与正则化平方和误差函数 (3.27) 是相等了。——习题 3.18 现在关于 $\mathbf{w}$ 进行完成平方项，

$$E(\mathbf{w}) = E(\mathbf{m}_N) + \frac{1}{2}(\mathbf{w} - \mathbf{m}_N)^{\mathrm{T}}\mathbf{A}(\mathbf{w} - \mathbf{m}_N) \tag{3.80}$$

其中

$$\mathbf{A} = \alpha\mathbf{I} + \beta\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi} \tag{3.81}$$

以及

$$E(\mathbf{m}_N) = \frac{\beta}{2}\|\mathbf{t} - \boldsymbol{\Phi}\mathbf{m}_N\|^2 + \frac{\alpha}{2}\mathbf{m}_N^{\mathrm{T}}\mathbf{m}_N \tag{3.82}$$

需要注意的是，$\mathbf{A}$ 是误差函数的二阶导数

$$\mathbf{A} = \nabla\nabla E(\mathbf{w}) \tag{3.83}$$

也就是海森矩阵 (Hessian matrix)。同时还可以定义 $\mathbf{m}_N$

$$\mathbf{m}_N = \beta\mathbf{A}^{-1}\boldsymbol{\Phi}^{\mathrm{T}}\mathbf{t} \tag{3.84}$$

根据 (3.54)，我们可以看出 $\mathbf{A} = \mathbf{S}_N^{-1}$，所以 (3.84) 是等价于 (3.53) 的，表示后验分布的均值。

现在可以通过与多元高斯分布归一化系数的标准结果做一下对比，从而评估这个关于 $\mathbf{w}$ 的积分，即——习题 3.19

$$\begin{aligned}
&\int \exp\{-E(\mathbf{w})\}\,\mathrm{d}\mathbf{w} \\
&= \exp\{-E(\mathbf{m}_N)\}\int \exp\left\{-\frac{1}{2}(\mathbf{w} - \mathbf{m}_N)^{\mathrm{T}}\mathbf{A}(\mathbf{w} - \mathbf{m}_N)\right\}\,\mathrm{d}\mathbf{w} \\
&= \exp\{-E(\mathbf{m}_N)\}(2\pi)^{M/2}|\mathbf{A}|^{-1/2}
\end{aligned} \tag{3.85}$$

根据 (3.78)，可以写出对数边缘似然函数

$$\ln p(\mathbf{t}|\alpha,\beta) = \frac{M}{2}\ln\alpha + \frac{N}{2}\ln\beta - E(\mathbf{m}_N) - \frac{1}{2}\ln|\mathbf{A}| - \frac{N}{2}\ln(2\pi) \tag{3.86}$$

这就是我们所需要的证据函数的表达式。

回到多项式回归问题中，我们可以关于多项式的阶数画出模型证据的图像，如图 3.14 所示。这里我们假设先验为 (1.65) 这样的形式，且参数 $\alpha = 5 \times 10^{-1}$。这个图像非常有指导意义。回看一下图 1.4，可以看出 $M = 0$ 的多项式对数据的拟合效果相当差，所以其模型证据也很低。当 $M = 1$ 时，拟合的效果好了很多，模型证据也高了很多。不过，当 $M = 2$ 时，拟合效果又变差了，因为真实的函数是正弦函数，而正弦函数的展开项里是没有偶数项的幂函数的。图 1.5 也展示了 $M = 1$ 变化成 $M = 2$ 时，数据残差仅有微小的减小，而且由于复杂模型的复杂度惩罚项很大，

所以在这个过程中，模型证据减小了。当 $M = 3$ 时，模型对数据的拟合效果得到了大幅的提升，所以模型证据也再次提升了，而且达到了所有多项式拟合中的最大值。在此之后，即使再次增加 M 的值，拟合的效果也只能得到稍许的提升，而模型复杂度却上升了，所以惩罚项也会随之变大，导致模型证据的下降。再看一下图 1.5，在 $M = 3$ 到 $M = 8$ 之间，泛化误差几乎就是一个常数，所以很难据此选择模型。不过，模型证据的结果很明显更加青睐 $M = 3$ 的模型，因为这是能够较准确描述数据的最简单的模型。

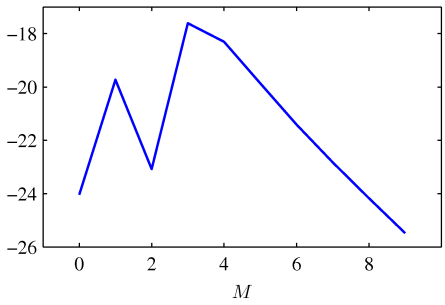

图 3.14: 关于阶数 M 绘制的模型对数证据的图像，对于多项式回归模型而言，模型证据最倾向于选择 $M = 3$ 的模型。

3.5.2 证据函数的最大化

首先研究 $p(\mathbf{t}|\alpha, \beta)$ 关于 α 的最大化。首先定义特征向量等式

$$(\beta\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi})\mathbf{u}_i = \lambda_i\mathbf{u}_i \tag{3.87}$$

根据 (3.81)，$\mathbf{A}$ 的特征值为 $\alpha + \lambda_i$。现在对 (3.86) 中含有 $\ln|\mathbf{A}|$ 的项关于 α 求导数，可得

$$\frac{d}{d\alpha}\ln|\mathbf{A}| = \frac{d}{d\alpha}\ln\prod_i(\lambda_i + \alpha) = \frac{d}{d\alpha}\sum_i\ln(\lambda_i + \alpha) = \sum_i\frac{1}{\lambda_i + \alpha} \tag{3.88}$$

于是 (3.86) 关于 α 的驻点满足

$$0 = \frac{M}{2\alpha} - \frac{1}{2}\mathbf{m}_N^{\mathrm{T}}\mathbf{m}_N - \frac{1}{2}\sum_i\frac{1}{\lambda_i + \alpha} \tag{3.89}$$

两侧同时乘以 2α 并重新组织一下，

$$\alpha\mathbf{m}_N^{\mathrm{T}}\mathbf{m}_N = M - \alpha\sum_i\frac{1}{\lambda_i\alpha} = \gamma \tag{3.90}$$

由于求和项中共有 M 项，于是 γ 可以写成

$$\gamma = \sum_i \frac{\lambda_i}{\alpha + \lambda_i} \tag{3.91}$$

简单解释一下 γ 的含义。根据 (3.90) 可以看出使得边缘似然函数取得最大值的 α 满足——习题 3.20

$$\alpha = \frac{\gamma}{\mathbf{m}_N^{\mathrm{T}}\mathbf{m}_N} \tag{3.92}$$

这是一个 α 的隐式解，一方面是因为 γ 与 α 是相关的，另一方面也是因为后验分布的模 $\mathbf{m}_N$ 本身也依赖于 α。所以我们采用迭代的方式来求解这个方程。首先选取一个 α 的初始值，然后用它根据 (3.53) 去求 $\mathbf{m}_N$，根据 (3.91) 计算 γ。接下来利用这些结果根据 (3.92) 重新估计 α，直到收敛为止。由于矩阵 $\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}$ 是固定的，我们可以在开始阶段计算它的特征值，然后在每一步更新中乘以 β 就可以得到 λ_i 了。

有个需要强调一下的地方，那就是 α 的值是完全通过对训练集的观察得到的。与最大似然方法需要独立的数据集来优化模型复杂度不同，这里不需要这样的措施。

类似地，我们也可以将对数边缘似然函数 (3.86) 关于 β 进行最大化。为了做到这一点，首先注意到 (3.87) 中的特征值 λ_i 与 β 成正比，所以 $d\lambda_i/d\beta = \lambda_i/\beta$，于是

$$\frac{d}{d\beta}\ln|\mathbf{A}| = \frac{d}{d\beta}\sum_i \ln(\lambda_i + \alpha) = \frac{1}{\beta}\sum_i \frac{\lambda_i}{\lambda_i + \alpha} = \frac{\gamma}{\beta} \tag{3.93}$$

于是边缘似然函数的驻点满足条件

$$0 = \frac{N}{2\beta} - \frac{1}{2}\sum_{n=1}^{N}\left\{t_n - \mathbf{m}_N^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)\right\}^2 - \frac{\gamma}{2\beta} \tag{3.94}$$

重新整理一下，——习题 3.22

$$\frac{1}{\beta} = \frac{1}{N - \gamma}\sum_{n=1}^{N}\left\{t_n - \mathbf{m}_N^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)\right\}^2 \tag{3.95}$$

和以前一样，这又是一个关于 β 的隐式解，可以首先选取 β 的初始值，然后计算 $\mathbf{m}_N$ 和 γ，再利用 (3.95) 更新 β，直到收敛为止。如果 α 和 β 的值要根据数据确定，那么可以在 γ 更新之后一起重新估计。

3.5.3 有效参数的数量

其实 (3.92) 还有另一种很不错的诠释 (MacKay, 1992a)，可以对 α 进行更加深刻的贝叶斯解释。为了看出这一点，我们研究一下如图 3.15 所示的似然函数与先验的轮廓线。这里我们比较隐晦地对参数空间的坐标轴进行了旋转变换，从而与 (3.87) 所定义的特征向量对齐。于是似然函数的轮廓线椭圆的两轴也与坐标轴对齐了。特

征值 λ_i 衡量的是似然函数的曲率，所以在图 3.15 中，特征值 λ_1 比 λ_2 要小 (因为曲率越小，似然函数的轮廓线就越扁)。由于 $\beta\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}$ 是正定矩阵，所以其特征值均为正数，所以比值 $\lambda_i/(\lambda_i+\alpha)$ 是处于 0 和 1 之间的数。再多推进一步，(3.91) 中的 γ 一定是位于范围 $0 \leqslant \gamma \leqslant M$ 中的。对于 $\lambda_i \gg \alpha$ 的方向，对应的参数 w_i 将非常接近其最大似然值，比值 $\lambda_i/(\lambda_i+\alpha)$ 也将非常接近 1. 这样的参数被称为 well determined 参数，因为它们的值严格受到数据的控制。相反，对于 $\lambda_i \ll \alpha$ 的方向，对应的参数 w_i 接近于 0，所以 $\lambda_i/(\lambda_i+\alpha)$ 也会随之非常接近 0。在这样的方向上，似然函数对参数值非常不敏感，所以先验会将参数设置为较小的值。所以 (3.91) 中的 γ 衡量的其实是 well determined 参数中有效参数的数量。

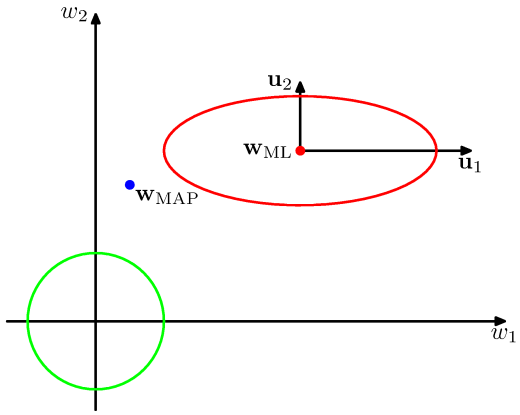

图 3.15: 似然函数的轮廓线 (红色曲线) 和先验概率分布 (绿色曲线)，参数空间的坐标轴经过旋转后与 Hessian 矩阵的特征向量 $\mathbf{u}_i$ 对齐。对于 $\alpha = 0$，后验的模为最大似然解 $\mathbf{w}_{\mathrm{ML}}$，而对于 $\alpha \neq 0$，模则位于 $\mathbf{w}_{\mathrm{MAP}} = \mathbf{m}_N$。在方向 w_1 上，由 (3.87) 定义的特征值 λ 比 α 小，因此 $\lambda_1/(\lambda_1+\alpha)$ 接近于 0，于是 w_1 的 MAP 值也接近于 0。与之相反，在 w_2 的方向上，特征值 λ_2 大于 α，因此 $\lambda_2/(\lambda_2+\alpha)$ 接近于 1，w_2 的 MAP 值接近于最大似然值。

我们可以通过将重新估计 β 的 (3.95) 与对应的最大似然解 (3.21) 做一下对比，从而得出更深刻的理解。这两个公式都将方差 (精度的逆) 表示为目标与模型预测之间误差求平方后再求平均得到的值。它们的区别在于，最大似然解的分母中数据点的数量是 N，贝叶斯方法中则是 $N-\gamma$。回想一下 (1.56)，一元高斯分布方差的最大似然估计为

$$\sigma^2_{\mathrm{ML}} = \frac{1}{N}\sum_{n=1}^{N}(x_n - \mu_{\mathrm{ML}})^2 \tag{3.96}$$

而且这个估计是有偏估计，因为期望的最大似然解 μ_{ML} 中包含了数据中的噪声。于是这会占用模型中的一个自由度。对应的无偏估计是 (1.59)，形式为

$$\sigma^2_{\mathrm{MAP}} = \frac{1}{N-1}\sum_{n=1}^{N}(x_n - \mu_{\mathrm{ML}})^2 \tag{3.97}$$

贝叶斯方法给出的解中，分母中的因子 $N-1$ 反映了一个自由度被均值占用了的这件事情，并相应修正了最大似然的偏差。现在研究一下线性回归模型中的相应结果，目标分布的均值为 $\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x})$，其中包含 M 个参数。不过并非所有的参数都是适合数据的。其中根据数据确定的有效参数数量为 γ，而且先验将其余的 $M-\gamma$ 个参数设置为较小的值。在贝叶斯方法给出的结果中也有所反映，即分母中的因子 $N-\gamma$，修正了最大似然解中的偏差。

我们可以在第 1.1 节中的正弦函数数据集中使用包含 9 个基底函数的高斯基底函数模型的模型证据来设定超参数，所以模型中的参数总数为 $M=10$，其中包含了偏差。为了简化起见，现在将 β 设定为真实值 11.1，然后使用模型证据确定 α，如图 3.16 所示。

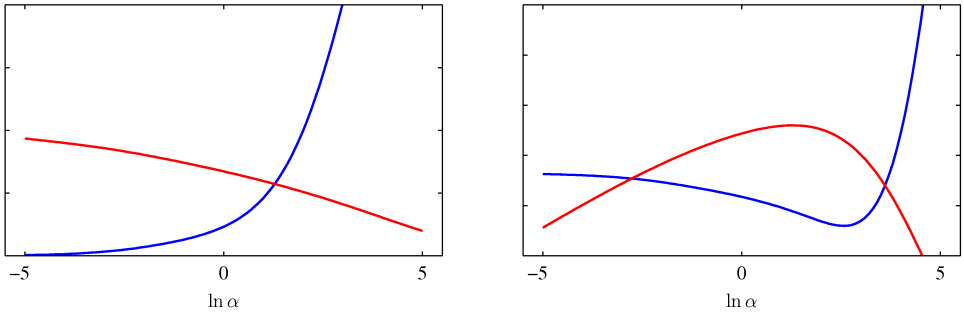

图 3.16: 左图为 γ 和 $2\alpha E_W(\mathbf{m}_N)$ 与 $\ln\alpha$ 的函数关系（分别为红色曲线和蓝色曲线），这里的数据集是正弦函数数据集。这两条曲线的交点正是最优解 α。右图中的红色曲线是对数模型证据 $\ln p(\mathbf{t}|\alpha,\beta)$ 关于 $\ln\alpha$ 的函数关系，恰好在左图中的交点处取得峰值。蓝色曲线为测试集误差，验证了在模型证据取得最大值时模型具有最优的泛化能力。

我们还可以通过绘制出独立参数与有效参数数量 γ 之间函数关系的的图像（如图 3.17 所示），看出参数 α 控制着参数 $\{w_i\}$。

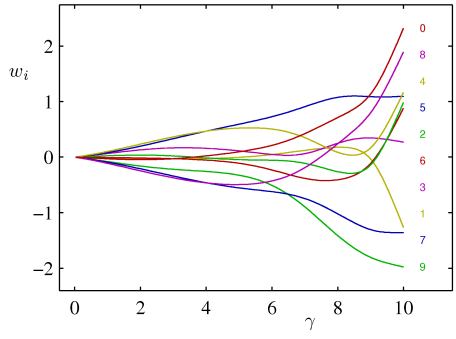

图 3.17: 高斯基底函数模型中 10 个参数 w_i 与有效参数数量 γ 的关系，其中超参数 $0\leqslant\alpha\leqslant\infty$，于是 γ 的范围是 $0\leqslant\gamma\leq M$。

假设一种极端情况 $N \gg M$，即数据的数量远大于参数的数量，那么根据 (3.87)，所有的参数都将是 well determined 参数，因为 $\mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi}$ 包含了所有数据的隐式求和，所以特征值 λ_i 会随着数据集规模的增大而增大。在这种情况下，$\gamma = M$，而且 α 和 β 的估计方程变成了

$$\alpha = \frac{M}{2E_W(\mathbf{m}_N)} \tag{3.98}$$

$$\beta = \frac{N}{2E_D(\mathbf{m}_N)} \tag{3.99}$$

其中 E_W 和 E_D 就是 (3.25) 和 (3.26) 中所定义的。这些近似结果大幅简化了概率模型重新估计的计算，因为不需要计算 Hessian 矩阵的特征值了。

3.6 固定基底函数的局限性

在本章中，我们主要关注的是固定非线性基底函数的线性组合的模型。我们已经看到线性参数具有很多有用的性质，包括最小二乘问题可以求出闭式解，贝叶斯方法也会比较容易使用。此外，为了选择适当的基底函数，我们可以对输入变量到目标变量之间的映射进行任意的非线性建模。在下一章中，我们将研究类似的分类模型。

由此可见，这样的线性模型似乎是构成了解决模式识别问题的通用框架。然而不幸的是，线性模型存在不少明显的缺点，所以我们需要在后面的章节中研究一些更复杂的模型，比如支持向量机和神经网络等。

问题究竟出在哪里？主要是因为在得到训练数据集之前就已经假设了基底函数 $\phi_j(x)$，而且基底函数一直是固定不变的，这是第 1.4 节中讨论的维数灾难的表现之一，因为接下来基底函数的数量会随着输入空间的维数 D 呈指数型飞速增长。

比较走运的是，我们可以利用实际数据集的两个属性来帮助缓解这个问题。首先，由于输入变量之间的强相关性，数据向量 $\{x_n\}$ 通常位于非线性流形内部，其内在维度小于输入空间的维数。在第 12 章中研究手写数字的图像时，我们会看到这样的例子。如果我们使用局部基底函数，就可以把它们仅分散到输入空间中包含数据的区域内。在使用径向基底函数网络和使用支持向量的相关向量机中会用到这个方法。基于 sigmoid 非线性自适应基底函数的神经网络模型可以自行调整参数，从而使基底函数在输入空间中的变化与数据流形相对应。第二个属性是目标变量可能仅在数据流形内的少量方向上具有显著的依赖性。神经网络可以通过选择基底函数响应的输入空间中的方向来利用此属性。