第三章 线性回归模型

之前第二章，讲的是无监督学习，就是一个预测没有对应的结果来对照好坏，没有一个输入向量到输出向量的函数，只针对一个变量x的密度估计或者数据聚集问题。现在开始讨论监督学习，从回归开始，回归问题指的是预测给定一个D维输入向量，预测一个或者多个连续的输出向量的值。比如多项式曲线拟合的问题，多项式是一种特殊的线性回归模型。

线性回归模型：指的是这个模型是参数的线性函数，输入变量不一定是线性。

basis function，应该翻译成基函数，是一组固定的输入变量的非线性函数的线性组合。

3.1 线性基函数模型

首先介绍了最简单的线性回归模型

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

然后扩展到一下形式，φj(x)是固定的（固定指不会随着参数变化，选择这个函数以后，观测数据不改变这些函数的意思）非线性函数，实际应用里预处理的特征提取过程就可以把φj(x)当成特征函数。w0是整体调整数值用的，叫偏差参数。

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

设定φ0(x) = 1，为了方便写成如下形式，其中，w = (w0, . . . , wM−1)T ，φ = (φ0, . . . , φM−1)T ，

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

φj(x)可以选什么呢？多项式曲线拟合里面选择了φj(x) = xj。还可以选择

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

μj决定φj(x)在输入空间的位置，s决定它们的空间规模。

还可以选择logistic sigmoid function，这个不知道怎么翻译

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

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

还有一种是傅里叶基函数，每个基函数代表一个频率，有无限的空间长度。局限于有限输入空间的基函数包含不同的空间频率。同时定位于指定的频率还有空间的叫小波函数。

后面讨论的φj(x)基本上都是φ(x) = x，还有单目标变量的情况

3.1.1 最大似然和最小二乘

考虑最小二乘的问题。

第一章讲了一个多项式函数曲线拟合的问题，现在回到这个问题，假设真实的目标变量是由一个决定式函数（还是不懂deterministic function是什么，这里的y(x,w)应该指的是**真实的模型**）和一个高斯噪音相加

生成。

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

噪音是后面那个（打不出来），是一个高斯随机变量，均值为0，精度为β。

用贝叶斯的思路推测t，t是某个数的概率，t的模型的分布可以写为，**这里的y(x,w)应该指自己预测的模型。**这个式子代表不同t的模型的分布的概率，把所有可能模型都涵盖进去，考虑了噪音的情况。

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

在第一章里面，有一个证明，最小化损失函数以后，最好的预测模型，即求解的回归方程的函数，是t的条件分布的均值。以下给出这个**证明。**

首先，给出损失函数，p(x,t)是联合分布，在对dx积分之后变成t是某个数的概率L(t, y(x))是损失函数

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

一般选择损失函数为差的平方，得到

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

目标是选择一个y(x)，使得损失函数E[L]最小。假定y是完全flexible function，可以直接利用变量的微积分得到

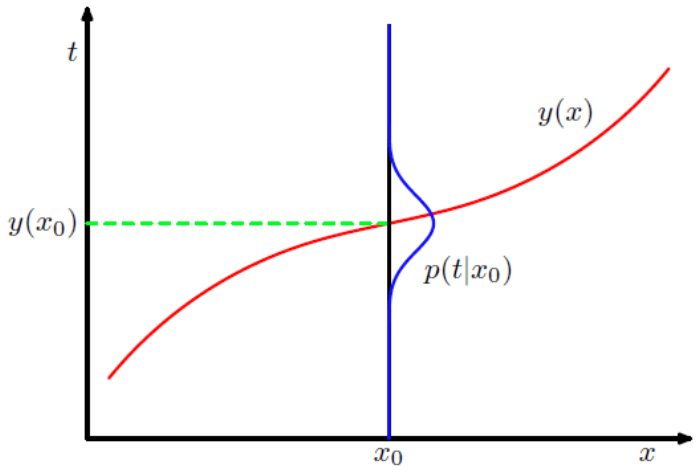
$$\frac{\delta \mathbb{E}[L]}{\delta y(\mathbf{x})} = 2\int \{y(\mathbf{x}) - t\}p(\mathbf{x}, t)\,\mathrm{d}t = 0.$$

解得

$$y(\mathbf{x}) = \frac{\int tp(\mathbf{x}, t)\,\mathrm{d}t}{p(\mathbf{x})} = \int tp(t|\mathbf{x})\,\mathrm{d}t = \mathbb{E}_t[t|\mathbf{x}]$$

证毕。这个证明的意义在于这个图，对于每一个特定的x，有一个t的推测的条件分布，然后最好的预测的t值就是t的条件分布的均值。所以猜测一个线性回归模型，就是根据数据，给出一个t的条件分布，这个条件分布包含很多个模型，这个条件分布的均值也是一个模型，就是当前最完美的模型了。

Figure 1.28 The regression function $y(x)$, which minimizes the expected squared loss, is given by the mean of the conditional distribution $p(t|x)$.



所以对于上述条件分布的均值，可以写成这样，这里隐含了t的条件推测是单峰分布，实际上t可能不是。

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

现在给一个数据集，每个数据集的每次数据有一个x对应一个t，给出最大似然方法的**步骤**。

写出这个数据集的输出变量的似然函数，黑体t代表把数据集所有t集合起来表示

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

对它取对数

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

ED(W)为误差函数

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

对取对数的似然函数求梯度，设梯度为0得到（相当于求导)

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

这个最大似然参数的作用应该是选定基函数后（第三章整章，没有说明默认选定基函数**φ(x) = x**）可以通过这个式子获得参数，获得参数以后代入y(x,w)=wTφ(x)，这就是想要的预测模型了。

其中，名为design matrix

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

$$\boldsymbol{\Phi}^{\dagger} \equiv \left(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}\right)^{-1}\boldsymbol{\Phi}^{\mathrm{T}}$$

在Φ中每个数据x对应M-1个基函数，Φ†是Moore-Penrose 伪逆矩阵

这里探讨了w0的性质，就是偏差参数，线性回归模型里面的第0个参数。它代表平均输出变量和基函数之间的加权平均的差（相当于每次除了把原结果减去基函数加权平均，还要再减去所有结果减去所有基函数加权平均的平均值，背后的数学道理不知道怎么表述出来）

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

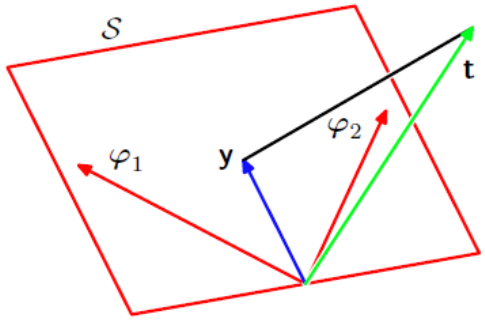
$$w_0 = \bar{t} - \sum_{j=1}^{M-1} w_j \overline{\phi_j}$$

where we have defined

$$\bar{t} = \frac{1}{N}\sum_{n=1}^{N} t_n, \qquad \overline{\phi_j} = \frac{1}{N}\sum_{n=1}^{N} \phi_j(\mathbf{x}_n).$$

以下是最小二乘法的图像表示。

Figure 3.2 Geometrical interpretation of the least-squares solution, in an N-dimensional space whose axes are the values of $t_1, \ldots, t_N$. The least-squares regression function is obtained by finding the orthogonal projection of the data vector $\mathbf{t}$ onto the subspace spanned by the basis functions $\phi_j(\mathbf{x})$ in which each basis function is viewed as a vector φ_j of length N with elements $\phi_j(\mathbf{x}_n)$.



3.1.4 正则化最小二乘

最小二乘指平方损失函数，正则化就是加入惩罚项，λ控制预测误差和参数值大小重要性的相对大小，EW(w)代表参数值大小的惩罚项，最简单的惩罚项就是参数的平方。选择惩罚项的方法在机器学习语言里面叫weight decay(权值衰减)。

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

套用之前的损失函数的式子，可以得到上式，对它的w的梯度最小化可以得到

$$\mathbf{w} = \left(\lambda\mathbf{I} + \boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}\right)^{-1}\boldsymbol{\Phi}^{\mathrm{T}}\mathbf{t}.$$

如果泛化一点，惩罚项w的平方变成q次方。1次方在统计学语言里面叫套索法。

正则化控制数据集大小，使之不会过拟合，本质上控制了有效模型复杂度。问题就转变为找λ的问题，这个问题后面会讨论，剩下的先讨论二次惩罚项的情况。因为有很多有用的性质。

3.1.5 多个输出结果的预测

之前的问题，都是一个数对应一个数，代表一次数据，如输入x=1，输出y=2，现在是一个向量对应一个向量，代表一次的数据，如输入x=[1,0,0,1]，y=[2,3,0,4]。他这里选择用相同一组基础函数预测数据集里面每次的数据。

这是预测的模型，y是K维列向量，W是M*K的参数矩阵，φ(x)是M维列向量。一个向量里的一个数对应M个参数和M个基函数，一个向量有K维，集合起来这样表示。

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

这是预测t是某一个向量的条件概率

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

每次数据集中起来，直接表示一个数据集，类似3.1.1，写出数据集的似然函数的对数，T是[t1, t2, t3, t4,....., tn]的转置，N行K列，每行代表一次数据，数据集有N维

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

最大化似然，可以得到一个最大似然估计的参数矩阵，有了基函数就可以代入得到具体的参数矩阵，

$$\mathbf{W}_{\mathrm{ML}}=\left(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}\right)^{-1}\boldsymbol{\Phi}^{\mathrm{T}}\mathbf{T}.$$

下式是分解到每个向量对应的参数求解结果

$$\mathbf{w}_k=\left(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}\right)^{-1}\boldsymbol{\Phi}^{\mathrm{T}}\mathbf{t}_k=\boldsymbol{\Phi}^{\dagger}\mathbf{t}_k$$

where tk is an N-dimensional column vector with components tnk for n = 1, . . . N.Thus the solution to the regression problem decouples between the different target variables

上面这句就有问题了，是不是n应该是K维的而不是N维的？tk应该指第k次的输入结果，它本身应该是K维向量的才对吧。。。这句话的意思应该是说每个向量会带来一个不同的解。

后面为了简洁还是讨论一次数据只是一个数的情况。

3.2 偏置-方差分解

频率学派视角的模型复杂度问题

分辨决策论的平方损失函数和误差函数平方和

首先给出最好的预测结果，即预测的t的分布的条件均值

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

给出决策论的损失函数代入h(x)的结果

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

第一项帮助我们选择模型y(x)，第二项代表t的噪音，是这个输入与输出内在的偏差。

对于一个特定数据集D，考虑第一项

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

在上式括号里面加减ED[y(x;D)]，然后取他的期望

$$\begin{aligned}&\mathbb{E}_{\mathcal{D}}\left[\{y(\mathbf{x};\mathcal{D})-h(\mathbf{x})\}^2\right]\\ &\quad=\underbrace{\{\mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})]-h(\mathbf{x})\}^2}_{(\text{bias})^2}+\underbrace{\mathbb{E}_{\mathcal{D}}\left[\{y(\mathbf{x};\mathcal{D})-\mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})]\}^2\right]}_{\text{variance}}.\end{aligned}$$

第一项是偏差的平方，代表所有数据集的平均预测和想要的回归方程之间的差，预测越准偏差越小。

第二项是方差，代表所有数据集的平均预测和每个数据集预测的偏离程度，衡量对特定数据集的敏感程度，越多种预测模型，越灵活，就越不敏感。

所以决策论的损失函数可以写为

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

where

$$\begin{aligned}(\text{bias})^2 &= \int \{\mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})] - h(\mathbf{x})\}^2 p(\mathbf{x})\,\mathrm{d}\mathbf{x} \\ \text{variance} &= \int \mathbb{E}_{\mathcal{D}}\left[\{y(\mathbf{x};\mathcal{D}) - \mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})]\}^2\right] p(\mathbf{x})\,\mathrm{d}\mathbf{x} \\ \text{noise} &= \int \{h(\mathbf{x}) - t\}^2 p(\mathbf{x},t)\,\mathrm{d}\mathbf{x}\,\mathrm{d}t\end{aligned}$$

目标是最小化损失函数。这可以看出是一个偏差与方差之间的权衡，如果模型灵活，他的形态可以有很多种，预测准，偏差小，方差大；模型不灵活，形态只有几种，预测不准，偏差大，方差小。

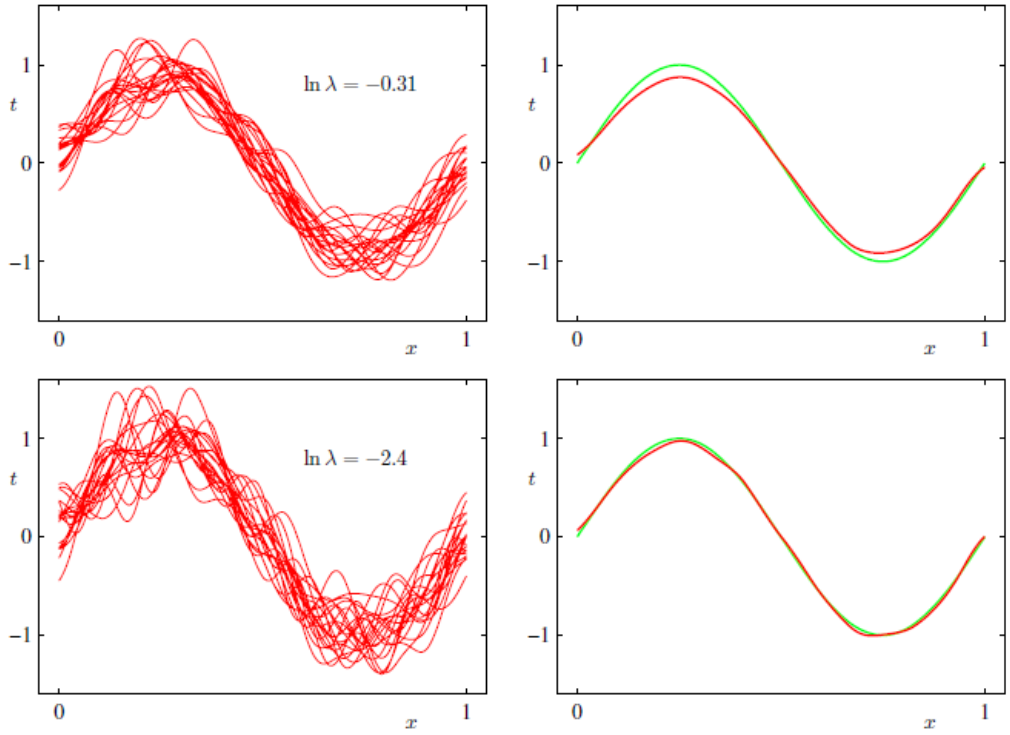


Figure 3.5 Illustration of the dependence of bias and variance on model complexity, governed by a regularization parameter λ, using the sinusoidal data set from Chapter 1. There are $L = 100$ data sets, each having $N = 25$ data points, and there are 24 Gaussian basis functions in the model so that the total number of parameters is $M = 25$ including the bias parameter. The left column shows the result of fitting the model to the data sets for various values of $\ln \lambda$ (for clarity, only 20 of the 100 fits are shown). The right column shows the corresponding average of the 100 fits (red) along with the sinusoidal function from which the data sets were generated (green).

λ越大，惩罚项惩罚的越多，偏差越大（预测越不准），方差越小（模型复杂度越小）。

以上的讨论，都是基于有很多个数据集的情况，现实当中只有一个数据集，如果有很多个数据集，更好地是把它变成一个数据集。接下来讨论贝叶斯的线性基函数模型，可以避免过拟合问题，自动根据数据决定模型的复杂度而不用费力去权衡。

3.3 贝叶斯线性回归

贝叶斯的方法首先还是要找参数的先验分布，因为模型是由参数决定的模型。最开始3.1.1数据集的似然函数是参数的指数二次方程，对应的共轭先验是高斯分布，

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

β是噪音精度。

后验分布是参数乘以似然函数，这个求法照旧，还是指数里面配方，然后归一化，第二章高斯分布有求过，就是已知p(x)和p(y|x)，求p(x|y)。直接代入得到

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

$$\begin{aligned} \mathbf{m}_N &= \mathbf{S}_N \left(\mathbf{S}_0^{-1}\mathbf{m}_0 + \beta\boldsymbol{\Phi}^{\mathrm{T}}\mathbf{t}\right) \\ \mathbf{S}_N^{-1} &= \mathbf{S}_0^{-1} + \beta\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}. \end{aligned}$$

接下来考虑简单的一种高斯先验，叫零均值等向高斯。

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

后验还是之前那个式子

$$\begin{aligned} \mathbf{m}_N &= \beta\mathbf{S}_N\boldsymbol{\Phi}^{\mathrm{T}}\mathbf{t} \\ \mathbf{S}_N^{-1} &= \alpha\mathbf{I} + \beta\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}. \end{aligned}$$

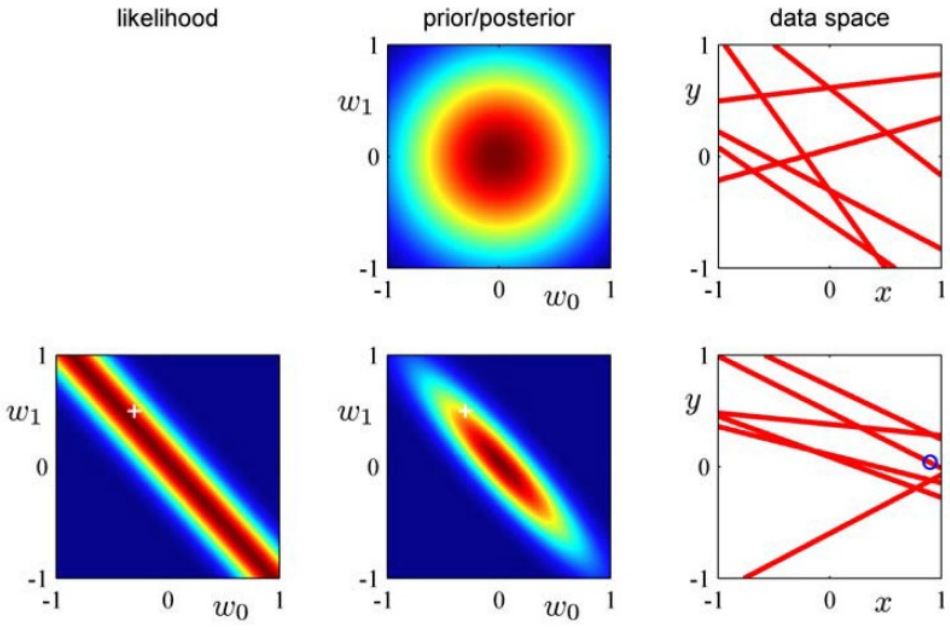
对数后验分布得到

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

最大化后验分布，相当于最小化平方误差函数和惩罚项。

举例，直线拟合的问题。单输入变量x和单输出变量t，线性模型y(x,w) = w0 + w1x，真实模型f(x, a) = a0+a1x，其中a0 = −0.3，a1 = 0.5，x从U(x|−1, 1)中挑选，经真实模型转化加入高斯噪音，得到输出。目的是从一组输入与输出当中找到真实模型，即使w0和w1尽量接近a0,a1。噪音精度β=25已知，分布精度α=2已知

根据上述条件画出了随着数据集大小增加，贝叶斯学习的性质。（光圈是怎么画出来的？似然在图像里又是代表什么？？？？如果我自己画的话，要怎么画出来呢？）



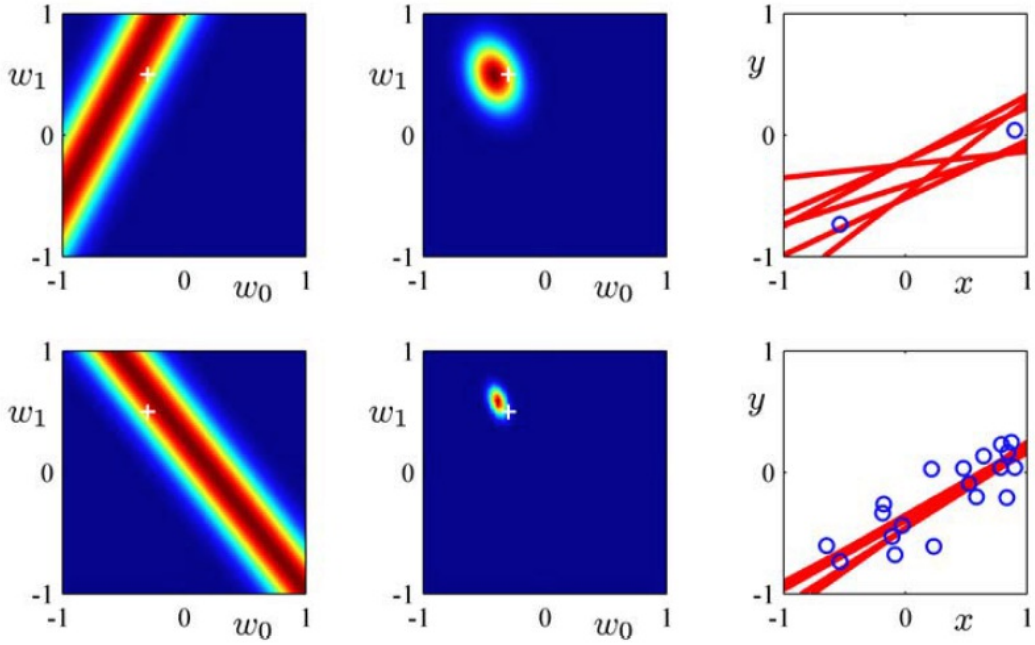


Figure 3.7 Illustration of sequential Bayesian learning for a simple linear model of the form $y(x, \mathbf{w}) = w_0 + w_1 x$. A detailed description of this figure is given in the text.

十字代表真实参数。第一行是没有任何数据，第三列是抽六个参数得到的模型。第二行是得到一个观测点的情况，第三行是观测到两个数据点，第四行是观测到二十个数据点，第四行的第一列是第二十个数据点的似然函数。随着数据点增加，w1w0的后验分布越来越接近十字。

3.3.2 预测的分布

3.3.1求的是参数的分布，实际要的是预测t，所以利用贝叶斯公式，定义一个预测t的分布。黑体t是从训练集得到的目标向量，所有的括号内都省去了x。

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

p(t|x,w, β)是3.1.1给出的预测模型，p(w|t, α, β)是3.3.1给出的参数的后验分布，和第二章高斯分布的问题，已知p(x)和p(y|x)，求p(y)一样，一番求解得到

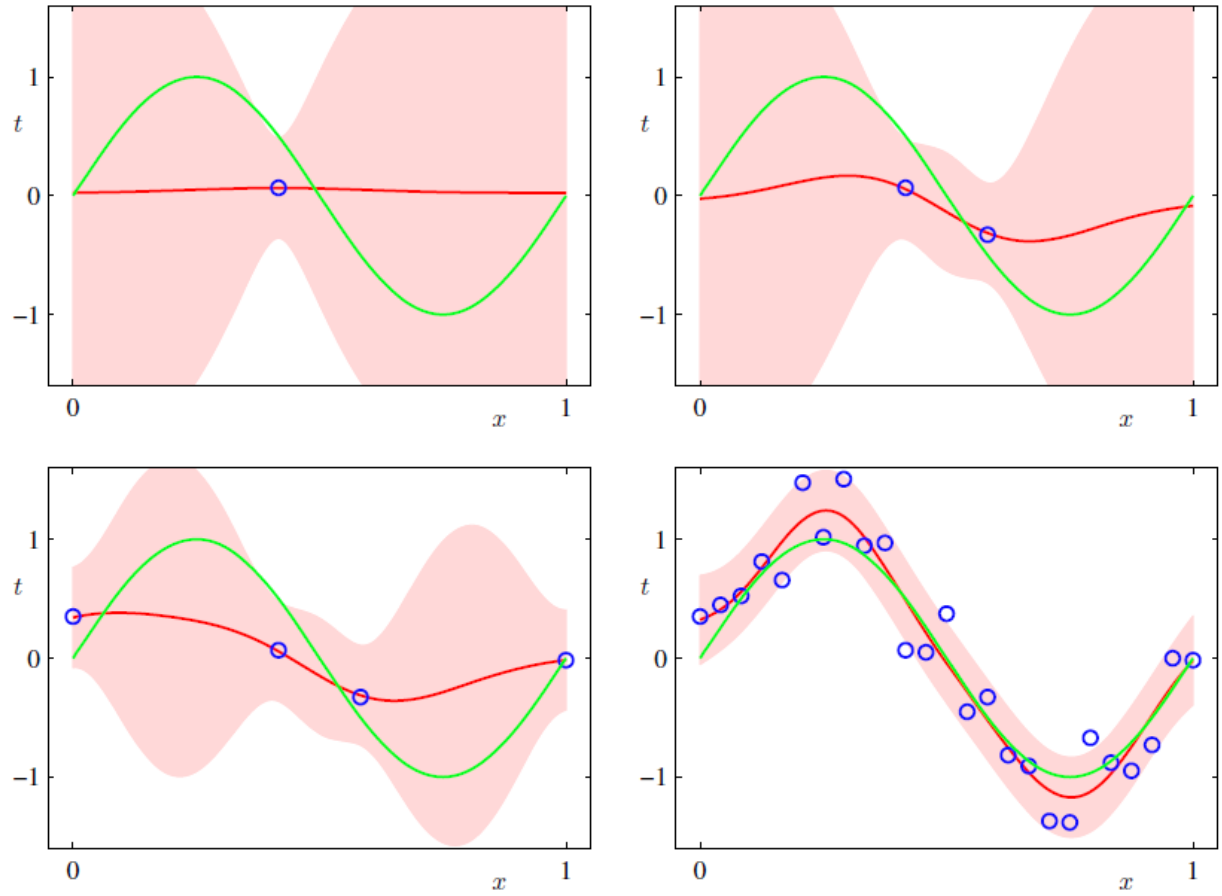
$$p(t|\mathbf{x}, \mathbf{t}, \alpha, \beta) = \mathcal{N}(t|\mathbf{m}_N^{\mathrm{T}} \boldsymbol{\phi}(\mathbf{x}), \sigma_N^2(\mathbf{x}))$$

其中

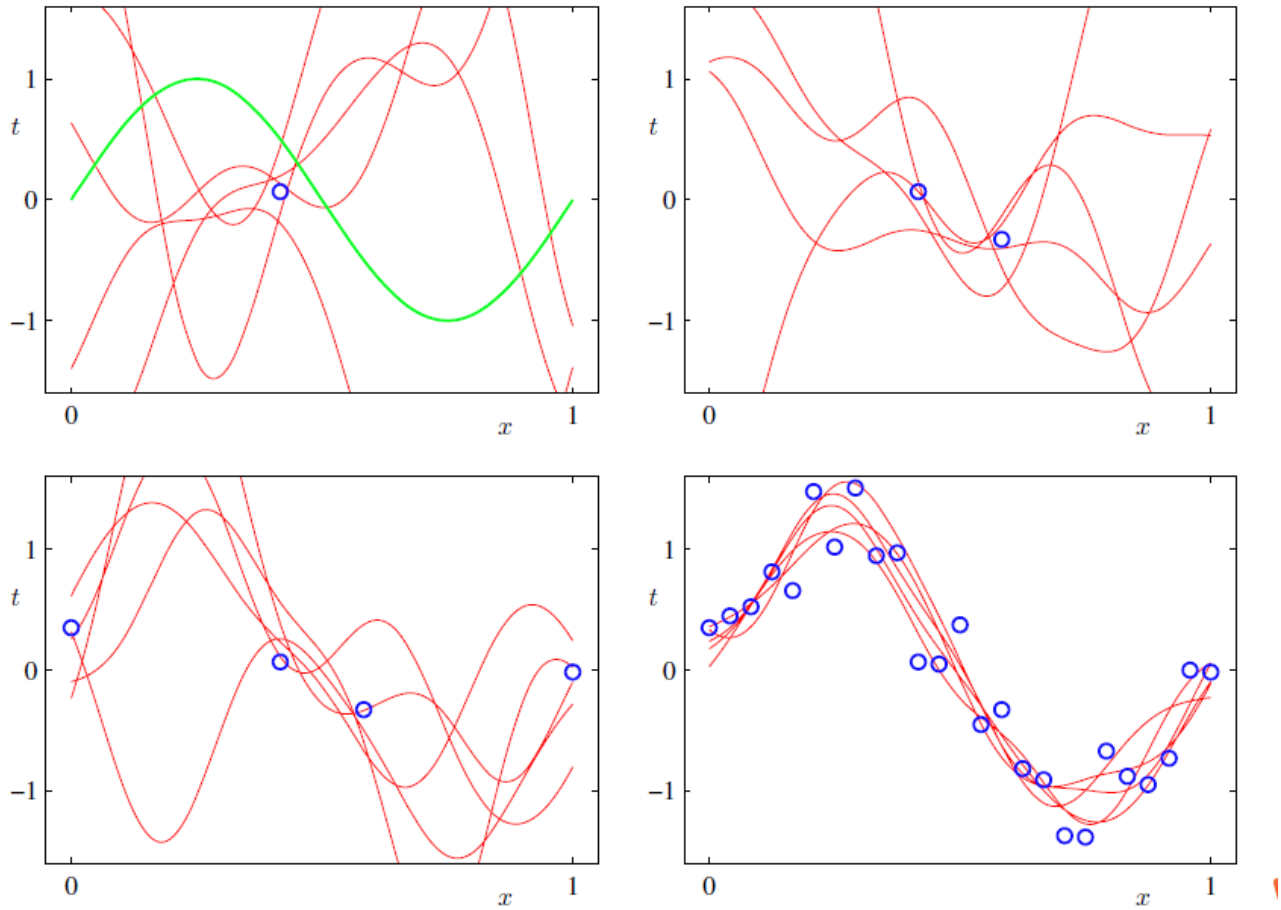
$$\sigma_N^2(\mathbf{x}) = \frac{1}{\beta} + \boldsymbol{\phi}(\mathbf{x})^{\mathrm{T}} \mathbf{S}_N \boldsymbol{\phi}(\mathbf{x}).$$

第一项表示数据噪音，第二项表示和w有关的不确定度。第二项从3.3.1可以知道随着点增多会不断减小，如下两图所示。

第一个图绿色是产生数据的真实模型正弦函数，数据点加了噪音。红色时预测出来的模型，红色区域是均值两侧各一个标准差长度，均值是指w后验指定了一个范围，这个范围每个w对应一个y(x,w)，把这个所有的y(x,w)平均起来可以得到均值，然后所有分布的均值画出来就是一条红线，结合第二个图可以理解标准差。



第二个图·是从w后验分布里面随机抽取几个w，得到y(x,w)模型，画出来的图像，上图的标准差应该是平均的y(x,w)和最远的y(x,w)的偏差。



3.3.3 等同核

3.3.1得到一个参数的后验均值mN = βSNΦTt，代入3.1最初定义的基函数预测模型y(x,w)=wTφ(x)，可以得到

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

可以视为训练集目标变量tn的线性组合

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

其中，下式叫平滑矩阵或者等同核

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

然后给了基函数为高斯和多项式函数和sigmoid函数的图，完全看不懂了，主要是等同核的这个函数的意义是什么？怎么体现在分布上面的？？？

Figure 3.10 The equivalent kernel $k(x, x')$ for the Gaussian basis functions in Figure 3.1, shown as a plot of x versus x', together with three slices through this matrix corresponding to three different values of x. The data set used to generate this kernel comprised 200 values of x equally spaced over the interval $(-1, 1)$.

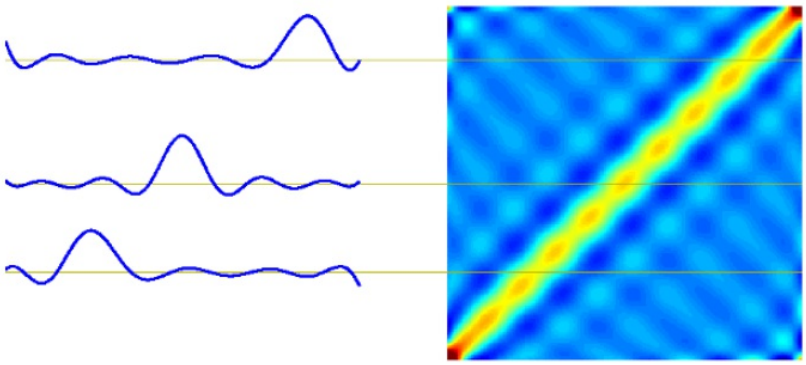
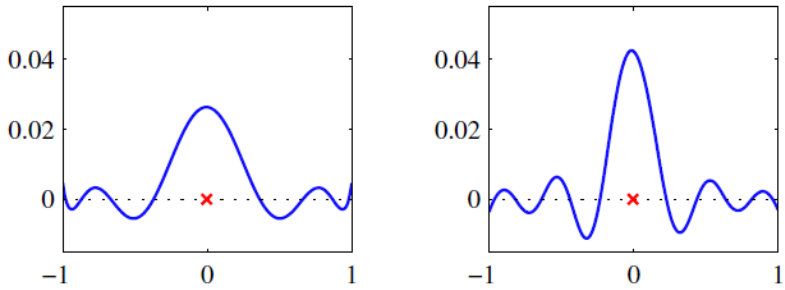


Figure 3.11 Examples of equivalent kernels $k(x, x')$ for $x = 0$ plotted as a function of x', corresponding (left) to the polynomial basis functions and (right) to the sigmoidal basis functions shown in Figure 3.1. Note that these are localized functions of x' even though the corresponding basis functions are nonlocal.



计算y(x) and y(x')的协方差

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

根据3.3.2的第二个图，可以看到在两个x值下的y值的后验分布的联合不确定度是被等同核决定的

利用核函数来构建线性回归方程的步骤如下。不再先找基函数，而是直接定义一个局部的核，根据现有的数据集预测新的输入向量x。这个框架叫高斯过程。

3.4 贝叶斯模型比较

贝叶斯的模型比较，概率主要表示选择模型时的不确定度

比较L个模型{Mi}，i = 1, . . . , L.这里的模型是概率分布，真实的模型就在这L个当中。不确定度为模型的不确定度p(Mi)

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

p(D|Mi)显示数据对不同模型的偏好，叫model evidence，也叫marginal likelihood

得到后验后作出预测，这个预测模型是混合分布

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

一个简单的近似就是直接用概率最大的模型预测，这就是模型选择。

对于被参数统治的模型，有

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

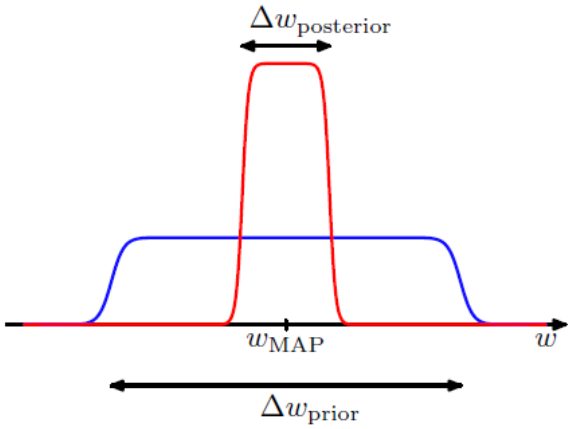
从样本的角度，边缘似然可以视为从参数的先验
随机抽取几个参数样本，得到相应的模型，从模型生成数据集D的概率。

接下来研究model evidence ，所有去掉下标M

假设后验分布在最可能参数处很尖，宽度为Δwposterior，参数先验完全平的，符合p(w) = 1/Δwprior，有

$$p(\mathcal{D}) = \int p(\mathcal{D}|w)p(w)\,\mathrm{d}w \simeq p(\mathcal{D}|w_{\mathrm{MAP}})\frac{\Delta w_{\mathrm{posterior}}}{\Delta w_{\mathrm{prior}}}$$

Figure 3.12 We can obtain a rough approximation to the model evidence if we assume that the posterior distribution over parameters is sharply peaked around its mode w_{MAP}.



取对数可以得到如下，第一项为似然函数，代表拟合模型的准确度，第二项代表惩罚项。

$$\ln p(\mathcal{D}) \simeq \ln p(\mathcal{D}|w_{\mathrm{MAP}}) + \ln\left(\frac{\Delta w_{\mathrm{posterior}}}{\Delta w_{\mathrm{prior}}}\right).$$

对于有M个参数的模型，设每个模型Δwposterior/Δwprior相同，有

$$\ln p(\mathcal{D}) \simeq \ln p(\mathcal{D}|\mathbf{w}_{\mathrm{MAP}}) + M\ln\left(\frac{\Delta w_{\mathrm{posterior}}}{\Delta w_{\mathrm{prior}}}\right).$$

增加模型的复杂度，第一项减小，第二项增加，所以这里需要权衡，最大化就自动权衡了。

下图为选择模型后，生成的数据集数量和概率关系图

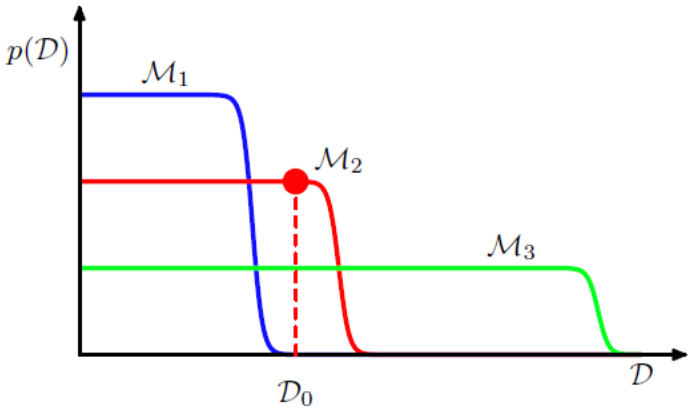


Figure 3.13 Schematic illustration of the distribution of data sets for three models of different complexity, in which $\mathcal{M}_1$ is the simplest and $\mathcal{M}_3$ is the most complex. Note that the distributions are normalized. In this example, for the particular observed data set $\mathcal{D}_0$, the model $\mathcal{M}_2$ with intermediate complexity has the largest evidence.

第一个模型分布简单，生成的数据集比较少，生成每个的概率大，第二个刚好合适，第三个复杂度最高，生成每个的概率小。

对于贝叶斯方法，最大化model evidence可以自动权衡模型复杂度，不会发生过拟合，因此不需要交叉验证和分出验证集。但是贝叶斯方法需要假设模型的形式，而且model evidence对于模型的先验的很多方面很敏感。实际应用里面还是要分出一个验证集出来。

3.5 证据近似

empirical Bayes，或者evidence approximation，指通过最大化对w积分的model evidence获得参数w的超参数α和β

本来式子是这样

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

其中

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

如果p(α, β|t)在∧α and ∧β,很尖，最简单的近似是直接变成

$$p(t|\mathbf{t}) \simeq p(t|\mathbf{t},\widehat{\alpha},\widehat{\beta}) = \int p(t|\mathbf{w},\widehat{\beta})p(\mathbf{w}|\mathbf{t},\widehat{\alpha},\widehat{\beta})\,\mathrm{d}\mathbf{w}.$$

从贝叶斯角度，

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

如果p(α, β).比较平，直接最大化p(t|α, β)，就得到∧α and ∧β

3.5.1 估计证据函数

估计

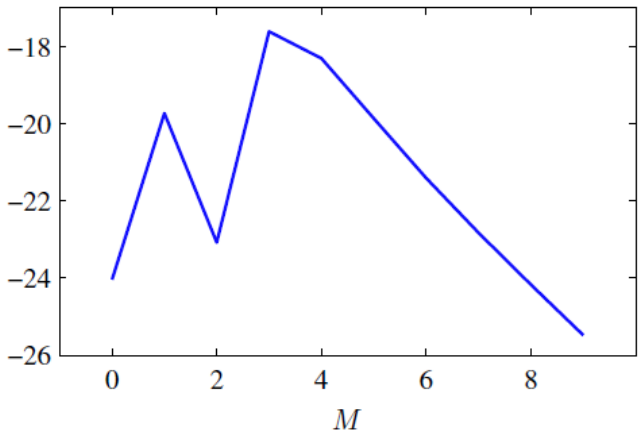
$$p(\mathbf{t}|\alpha,\beta) = \int p(\mathbf{t}|\mathbf{w},\beta)p(\mathbf{w}|\alpha)\,\mathrm{d}\mathbf{w}.$$

具体是一个数学过程，最后结果为

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

设定参数之后，根据这个式子可以画图，从这个图可以看出模型对数据的拟合程度还有复杂度两重影响。

Figure 3.14 Plot of the model evidence versus the order M, for the polynomial regression model, showing that the evidence favours the model with $M = 3$.



3.5.2 最大化证据函数

通过对上面估计的式子对α求导，还有一系列数学运算得到

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

其中

$$\gamma = \sum_i \frac{\lambda_i}{\alpha + \lambda_i}. \qquad \left(\beta\mathbf{\Phi}^{\mathrm{T}}\mathbf{\Phi}\right)\mathbf{u}_i = \lambda_i\mathbf{u}_i.$$

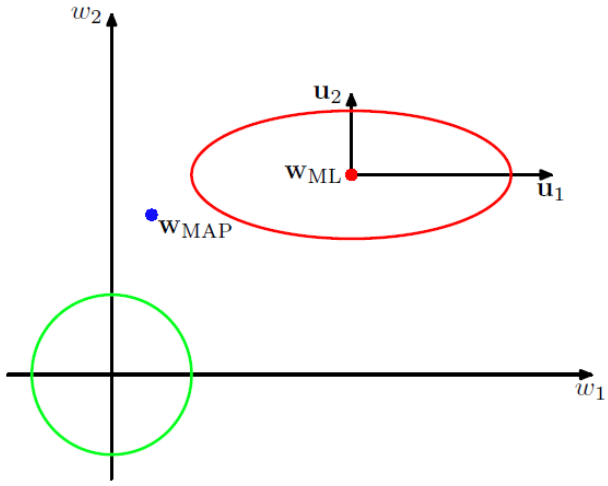
这是超参数α的隐式解，通过设定α的初始值，然后不断迭代，最后收敛到想要的解。

同理β的隐式解

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

3.5.3 参数的有效数

Figure 3.15 Contours of the likelihood function (red) and the prior (green) in which the axes in parameter space have been rotated to align with the eigenvectors $\mathbf{u}_i$ of the Hessian. For $\alpha = 0$, the mode of the posterior is given by the maximum likelihood solution $\mathbf{w}_{\mathrm{ML}}$, whereas for nonzero α the mode is at $\mathbf{w}_{\mathrm{MAP}} = \mathbf{m}_N$. In the direction w_1 the eigenvalue λ_1, defined by (3.87), is small compared with α and so the quantity $\lambda_1/(\lambda_1 + \alpha)$ is close to zero, and the corresponding MAP value of w_1 is also close to zero. By contrast, in the direction w_2 the eigenvalue λ_2 is large compared with α and so the quantity $\lambda_2/(\lambda_2+\alpha)$ is close to unity, and the MAP value of w_2 is close to its maximum likelihood value.



红线代表似然函数，绿线代表先验分布，特征值λ表述似然函数的弯曲程度

对于α=0，后验的最大似然解为wml，对于α≠0，解为wMAP = mN；

λ较α小的，γ接近于0，wMAP接近于0，这个方向似然函数对参数值不敏感，参数在先验里设定为比较小的值

λ较α大的，γ接近于M，wMAP接近于最大似然值，对应参数比较严格受数据的限制，这种参数叫well determined parameters

γ衡量了well determined parameters的总数

（还是有些看不懂，mode是求什么？这个方向似然函数对参数值不敏感是什么意思？？？λ1和w1怎么联系起来呢？？mode之前理解不够，mode是众数，体现在分布里面是它的顶点，就是最可能的w）

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

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

比较3.21和3.95，两式都表现了平均方差，差别在于平均的分母发生变化了。

这个分母的变化体现了什么呢？第一章的时候可以看到，对于单一变量高斯分布的方差的最大似然估计是

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

这个估计是有偏差的，因为μML把数据完全拟合了，数据本身是带有噪音的，拟合均值的过程用掉了模型的一个自由度，对应的无偏差的估计应该是

$$\sigma_{\mathrm{MAP}}^2 = \frac{1}{N-1}\sum_{n=1}^{N}(x_n - \mu_{\mathrm{ML}})^2.$$

回到线性回归方程，有γ个是有效参数，剩下的参数被先验设定为较小的值，所以方差的贝叶斯结果有N-γ个因子，纠正了最大似然结果的偏差。

3.6 固定基函数的限制

第三章一整个章节，主要讨论的是固定的非线性基函数的组合。在观测数据集之前就把基函数固定，会导致维度灾难，输入空间维度增加，基函数的数量也会呈指数级别增长。

真实数据的两个性质使得这个问题没有这么严重，第一是输入向量通常在一个维度相对输入空间低很多的子空间里，因为输入向量之间是有关联的。这时候就可以使用localized basis function，把它们设定为只分散在低维度的子空间里面，这个方法被用于 radial basis function networks 和支持向量机，关联向量机。第二个性质是目标变量会只依赖于这个低维子空间的一小部分方向。