6 核方法

对于回归问题和分类问题的线性参数模型，从输入x到输出y的映射$y(x,w)$形式由可调节参数构成的向量w控制。学习阶段，训练数据用于得到参数向量的点估计，或者用来确定这个向量的后验概率分布，然后训练数据被丢弃，新输入的预测纯粹依靠学习得到的参数w，该方法也被用于如神经网络等非线性参数模型。
对于训练数据点或者其子集在预测阶段仍然保留并被使用的模式识别技术，如最近邻方法。这种方法将每一个新的测试向量分配为训练数据集里距离最近的样本的标签。基于存储的方法将整个训练集的数据存储起来，对未来的数据点进行预测，通常此类方法需要一个度量，来定义整个输入空间中任意两个向量之间的相似度，这种方法训练速度快，但是对于预测数据点的预测速度很慢。
许多线性参数模型可被转化为一个等价的“对偶表示”。对偶表示中，预测基础也是在训练数据点处计算核函数的线性组合。对于基于固定非线性的特征空间映射$\phi(x)$的模型来说，核函数关系如下：

$$k(x,x')=\phi(x)^T\phi(x')$$

所以可以看到，核函数对于其参数是对称的，即$k(x,x')=k(x',x)$，通过考虑特征空间的恒等映射$\phi(x)=x$，就得到$k(x,x')=x^Tx'$,我们将其称为线性核。
用特征空间的内积的方式表示核的概念使得我们能够对许多著名的算法进行有趣的扩展。扩展的方法是使用核技巧（kernel trick），也被称为核替换。一般的思想是如果我们有一个算法，其输入向量x只以标量积的形式出现，那么我们可以用一些其他的核来替换标量积。例如，可以使用核替换方法用于主成分分析，进而产生了PCA的非线性变种。核替换的其他例子包括最近邻分类器和核Fisher判别函数。
常用的核函数有各种不同的形式，许多核函数只是参数差值的函数，即$k(x,x')=k(x-x')$,这被称

为静止核，因为核函数对于输入空间的平移具有不变性。另一种是同质核，也被称为径向基函数，它只依赖于参数之间的距离大小，即$k(x,x')=k(||x-x'||)$

6.1 对偶表示

许多回归的线性模型和分类的线性模型的公式都可以使用对偶表示重写。使用对偶表示形式，核函数可以自然地产生。
我们考虑一个线性模型，其参数通过最小化正则化的平方和误差函数来确定。正则化平方和误差函数：

$$J(w)=\frac{1}{2}\sum_{n=1}^{N}\{w^T\phi(x_n)-t_n\}^2+\frac{\lambda}{2}w^Tw$$

其中$\lambda\geq 0$,若我们令$J(w)$关于w的导数为0，则可以看到w的解是向量$\phi(x_n)$的线性组合的形式，系数是w的函数，形式为：

$$w=-\frac{1}{\lambda}\sum_{n=1}^{N}\{w^T\phi(x_n)-t_n\}\phi(x_n)=\sum_{n=1}^{N}a_n\phi(x_n)=\Phi^Ta$$

其中Φ是设计矩阵，第n行为$\phi(x_n)^T$。这里向量$a=(a_1,\cdots,a_N)^T$,我们定义了：

$$a_n=-\frac{1}{\lambda}\{w^T\phi(x_n)-t_n\}$$

消去w，求解a，可以得到：

$$a=(K+\lambda I_N)^{-1}\mathbf{t}$$

我们现在不直接对参数向量w进行操作，而是将参数向量a重新计算整理最小平方算法，得到一个对偶表示。如果我们使用$w=\Phi^Ta$代入$J(w)$，可以得到：

$$J(a)=\frac{1}{2}a^T\Phi\Phi^T\Phi\Phi^Ta-a^T\Phi\Phi^T\mathbf{t}+\frac{1}{2}\mathbf{t}^T\mathbf{t}+\frac{\lambda}{2}a^T\Phi\Phi^Ta$$

其中$\mathbf{t}=(t_1,\cdots,t_N)^T$，定义Gram矩阵$K=\Phi\Phi^T$,一个$N\times N$的对称矩阵，元素为：

$$K_{nm}=\phi(x_n)^T\phi(x_m)=k(x_n,x_m)$$

平方和误差函数可写作：

$$J(a)=\frac{1}{2}a^TKKa-a^TK\mathbf{t}+\frac{1}{2}\mathbf{t}^T\mathbf{t}+\frac{\lambda}{2}a^TKa$$

将a带入到$y(x)$对于新的输入x，可以得到下面的预测：

$$y(x) = w^T\phi(x) = a^T\Phi\phi(x) = k(x)^T(K + \lambda I_N)^{-1}\mathbf{t}$$

其中，对于向量$k(x)$,它的元素为$k_n(x) = k(x_n, x)$.
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$k(x, x\prime)$表示。这被称为对偶公式,因为a的解可表示为$\phi(x)$的线性组合，从而可使用参数向量w恢复出原始的公式。注意，在x处的预测由训练集数据的目标值的线性组合得到。
在对偶公式中，我们通过对一个N * N 的矩阵求逆来确定参数向量a，而在原始参数空间公式中，我们要对一个M * M 的矩阵求逆来确定w。由于N 通常远大于M ，因此对偶公式似乎没有实际用处。然而，正如我们将要看到的那样，对偶公式的优点是，它可以完全通过核函数k(x, 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$\phi(x)$，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.

6.2 核方法

为了利用核替换，我们需要构造合法的核函数，一种方法是选择一个特征空间映射$\phi(x)$,然后利用该映射寻找对应的核，一维空间的核函数被定义为：

$$k(x, x') = \phi(x)^T\phi(x') = \sum_{i=1}^{M}\phi_i(x)\phi_i(x')$$

其中$\phi_i(x)$是基函数。
另一种方法是直接构造核函数，我们需要确保核函数是合法的，即它对应于某一个（可能是无限维的）特征空间的标量积，对于下面的核函数：

$$k(x, z) = (x^Tz)^2$$

如果我们取特殊情况，二维输入空间$x = (x_1, x_2)$，我们就可以展开这一项，得到对应的非线性特征映射：

$$\begin{aligned}
k(x, z) &= (x^Tz)^2 \\
&= (x_1z_1 + x_2z_2)^2 \\
&= x_1^2z_1^2 + 2x_1z_1x_2z_2 + x_2^2z_2^2 \\
&= (x_1^2, \sqrt{2}x_1x_2, x_2^2)(z_1^2, \sqrt{2}z_1z_2, z_2^2) \\
&= \phi(x)^T\phi(x)
\end{aligned}$$

所以看到特征映射的形式为：

$$\phi(x) = (x_1^2, \sqrt{2}x_1x_2, x_2^2)$$

因此这个特征映射由所有的二阶项组成，每一个二阶项有一个具体的系数。
更一般的我们找到一个合法核函数的充分必要条件是**Gram**矩阵(元素由$k(x_n, x_m)$给出)在所有的集合$\{x_n\}$的选择下都是半正定的。
所以我们给出下面的构造核函数的性质：

- 对于合法的核$k_1(x, x'), k_2(x, x'), c > 0$，核$k(x, x') = ck_1(x, x')$是合法的
- 对于合法的核$k_1(x, x'), k_2(x, x')$,任意函数f，核$k(x, x') = f(x)k_1(x, x')f(x')$是合法的
- 对于合法的核$k_1(x, x'), k_2(x, x')$,系数非负的多项式q，核$k(x, x') = q(k_1(x, x'))$是合法的
- 对于合法的核$k_1(x, x'), k_2(x, x')$，核$k(x, x') = \exp(k_1(x, x'))$是合法的
- 对于合法的核$k_1(x, x'), k_2(x, x')$，核$k(x, x') = k_1(x, x') + k_2(x, x')$是合法的
- 对于合法的核$k_1(x, x'), k_2(x, x')$，核$k(x, x') = k_1(x, x')k_2(x, x')$是合法的
- 对于一个由x到R^M中的函数，以及R^M中合法的核$k_3(.,.)$ 核$k(x, x') = k_3(\phi(x), \phi(x'))$是合法的
- 对于对称的半正定矩阵A，核$k(x, x') = x^T Ax'$是合法的
- 对于变量x_a, x_b,且$x = (x_a, x_b), k_a, k_b$是各自空间内的合法核函数，则：$k(x, x') = k_a(x, x') + k_b(x, x')$是合法的
- 对于变量x_a, x_b,且$x = (x_a, x_b), k_a, k_b$是各自空间内的合法核函数，则：$k(x, x') = k_a(x, x')k_b(x, x')$是合法的

我们可以根据以上性质构造更加复杂的核函数，要求$k(x, x')$是对称半正定的，并且它表示面向具体应用中x和x'的适当形式的相似性。
可以看到简单的多项式核$k(x, x') = (x^T x')^2$值包含二次项。我们考虑一般的核$k(x, x') = (x^T x' + c)^2$其中$c > 0$,则对应的映射$\phi(x)$就会包含常数、线性项和二次项，类似的，$k(x, x') = (x^T x')^M$包含所有M阶的单项式。可以进行类似推广$k(x, x') = (x^T x' + c)^M$,其中$c > 0$
另一种经常使用的核函数形式为：

$$k(x, x') = \exp(-\frac{||x - x'||^2}{2\sigma^2})$$

这被称为高斯核，我们可以将平方项展开：

$$||x - x'||^2 = x^T x + (x')^T x' - 2x^T x'$$

从而，

$$k(x,x')=\exp(-\frac{x^Tx}{2\sigma^2})\exp(\frac{x^Tx}{\sigma^2})\exp(-\frac{(x')^Tx'}{2\sigma^2})$$

可以看到这是一个合法的核，对应于高斯核的特征向量有无穷的维数。
高斯核不局限于使用欧几里得距离，如果将x^Tx'替换为非线性核$\kappa(x,x')$我们就得到：

$$k(x,x')=\exp(-\frac{1}{2\sigma^2}(\kappa(x,x)+\kappa(x'x')-2\kappa(x,x')))$$

核方法的重要贡献是可以扩展到符号化输入，而不是简单的实数向量。对于一个固定的集合，定义一个非向量空间，该空间由该集合的所有可能子集构成。如果A_1,A_2是两个这样的子集，则核的一个选择可以是：

$$k(A_1,A_2)=2^{|A_1\cap A_2|}$$

其中$A_1\cap A_2$表示两个集合的交集，这是一个合法的核函数，因为可以证明它对应于一个特征空间中的一个内积。
另一种构造核的方法是从一个概率生成式模型开始构造，使得我们在判别时框架中使用生成式模型.生成式模型可以很自然地处理缺失的数据，并且在隐马尔科夫模型的情况下，可以处理变长的序列，而判别式模型在判别式任务中的效果更好，于是，可以用生成式方法生成一个核，然后在判别式方法中使用这个核。
给定一个生成式模型，可以定义一个核：

$$k(x,x')=p(x)p(x')$$

它表明，如果两个输入x和x'都有较高的概率，则它们是相似的。我们可以进行拓展，对它们进行加和并附带正的权值系数$p(i)$,得到形式为：

$$k(x,x')=\sum_i p(i)p(x|i)p(x'|i)$$

在无限求和的极限情况下，也可以考虑：

$$k(x,x')=\int p(x|z)p(x'|z)p(z)\mathrm{d}z$$

z是一个连续潜在变量
现在假设我们的数据由长度为L的有序序列组成，即一个观测为

$$X=\{x_1,\cdots,x_L\}$$

对于这种序列，一个流行的生成式模型是隐马尔科夫模型，它把概率p(X)表示为对应的隐含状态序列$Z = \{z_1, \ldots, z_L\}$上的积分或求和。
我们可以使用这种方法定义一个核函数来度量两个序列X和X'的相似度,定义核函数的方式是拓展混合表示，得到：

$$k(X, X') = \sum_{Z} p(X|Z)p(X'|Z)p(Z)$$

从未两个观测预测通过相同的隐含序列Z产生
此外，还可以使用Fisher核，对于一个参数生成式模型$p(x|\theta)$,其中θ表示参数的向量。我们要找到一个核，度量生成式模型的两个输入变量x和x'之间的相似性，考虑关于θ的梯度，它定义了一个“特征”空间的向量，其维度与θ的维度相同，考虑Fisher得分：

$$g(\theta, x) = \nabla_\theta \ln p(x|\theta)$$

根据Fisher得分，Fisher核被定义为：

$$k(x, x') = g(\theta, x)^T F^{-1} g(\theta, x')$$

这里的F是信息矩阵，定义为：

$$F = \mathbb{E}_x[g(\theta, x)g(\theta, x)^T]$$

Fisher信息矩阵的存在使得这个核在密度模型的非线性重参数化$\theta \to \varphi(\theta)$下具有不变性。
在实际应用中计算Fisher信息矩阵是不可行的，所以可以将定义中的均值替换为样本均值，可以得到：

$$F \simeq \frac{1}{N} \sum_{n=1}^{N} g(\theta, x_n)g(\theta, x_n)^T$$

这是Fisher得分的协方差矩阵，我们也可以省略Fisher信息矩阵，使用非不变核：

$$k(x, x') = g(\theta, x)^T g(\theta, x')$$

sigmod核定义为：

$$k(x, x') = \tanh(ax^T x' + b)$$

它的Gram矩阵通常不是半正定的，但是它可以赋予和展开一个与神经网络模型表面的相似性。正如我们将看到的那样，在基函数有无穷多的极限情况下，一个具有恰当先验的贝叶斯神经网络将会变为高斯过程，因此这就提供了神经网络与核方法之间的一个更深层的联系。

6.3 径向基函数网络

一种广泛使用的基函数是径向基函数（radial basis functions）。径向基函数中，每一个基函数只依赖于样本和中心μ_j 之间的径向距离（通常是欧几里得距离），即:

$$\phi_j(x) = h(||x - \mu_j||)$$

径向基函数被用来进行精确的函数内插，给定一组输入$\{x_1, \cdots, x_N\}$以及对应目标值$\{t_1, \cdots, t_N\}$，目标是咋厚道一个光滑的函数$f(x)$,使之精确拟合每一个目标值，对于$n = 1, \ldots, N$，都有$f(x_n) = t_n$，可以将$f(x)$表示为径向基函数的线性组合，每一个径向基函数都以数据点为中心，即:

$$f(x) = \sum_{n=1}^{N} w_n h(||x - x_n||)$$

系数w_n的值可以通过最小平方方法求出。
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。
对于一个使
用微分算符定义的带有正则化项的平方和误差函数，最优解可以通过对算符的Green函数（类似于离散矩阵的特征向量）进行展开，每个数据点有一个基函数。如果微分算符是各向同性的，那么Green函数只依赖于与对应的数据点的径向距离。由于正则化项的存在，因此解不再精确地对训练数据进行内插。
对于输入变量（而不是目标变量）具有噪声时的内插问题，如果输入变量x的噪声由一个服从分布$v(\xi)$的变量ξ描述，平方和误差函数就变成了:

$$E = \frac{1}{2} \sum_{n=1}^{N} \int \{y(x_n + \xi) - t_n\}^2 v(\xi) \mathrm{d}\xi$$

使用变分法对y（x）进行优化:

$$y(x) = \sum_{n=1}^{N} t_n h(x - x_n)$$

其中基函数为:

$$h(x - x_n) = \frac{v(x - x_n)}{\sum_{n=1}^{N} v(x - x_n)}$$

可以看到，这是一个以每一个数据点为中心的基函数。如果噪声分布$v(\xi)$是各向同性的。基它是$||\xi||$的一个函数，则基函数就是径向的。

上面的基函数是归一化的，即对于所有的x都有$\sum_n h(x-x_n)=1$.有时在实际应用中会用到归一化，因为它避免了输入空间中存在所有的基函数全部取较小值的区域，这种区域会导致在这些区域的预测值过小，或者完全由基参数控制.
由于每一个数据点都关联了一个基函数，因此当对于新的数据点进行预测时，对应的模型的计算开销会非常大.因此，一些新的模型被提出来,这些模型仍然对径向基函数进行展开，但是基函数的数量M 要小于数据点的数量N .通常，基函数的数量，以及它们的中心μ_i，都只是基于输入数据$\{x_n\}$自身来确定。然后基函数被固定下来，系数$\{w_i\}$由最小平方方法通过解线性方程的方式确定.
选择基函数中心的一种最简单的方法是使用数据点的一个随机选择的子集。一个更加系统化的方法被称为正交最小平方。这是一个顺序选择的过程，在每一个步骤中，被选择作为基函数的下一个数据点对应于能够最大程度减小平方和误差的数据点。展开系数值的确定是算法的一部分。还可以使用聚类算法（例如K均值算法），这时得到的一组基函数中心不再与训练数据点重合。

6.3.1 Nadaraya-Watson模型

我们可以从核密度估计开始，以一个不同的角度研究核回归模型。假设我们有一个训练集$\{x_n,t_n\}$，我们使用Parzen密度估计来对联合分布p(x, t)进行建模，即:

$$p(x,t)=\frac{1}{N}\sum_{n=1}^{N}f(x-x_n,t-t_n)$$

其中$f(x,t)$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。
我们需要找到回归函数$y(x)$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:

$$\begin{aligned}y(x)&=\mathbb{E}[t|x]\\&=\int_{\infty}^{\infty}tp(t|x)\mathrm{d}t\\&=\frac{\int tp(x,t)\mathrm{d}t}{\int p(x,t)\mathrm{d}t}\\&=\frac{\sum_n\int tf(x-x_n,t-t_n)\mathrm{d}t}{\sum_m\int f(x-x_m,t-t_m)\mathrm{d}t}\end{aligned}$$

简单起见，我们假设分来难过的密度函数的均值为0:

$$\int_{-\infty}^{+\infty}f(x,t)t\mathrm{d}t=0$$

对于所有的x都成立，使用简单的变量替换，可以得到:

$$y(x) = \frac{\sum_n g(x - x_n)}{\sum_m g(x - x_m)} = \sum_n k(x, x_n) t_n$$

该结果叫做Nadaraya-Watson模型，或者称为核回归
核函数：

$$k(x, x_n) = \frac{g(x - x_n)}{\sum_m g(x - x_n)}$$

并且我们定义了：$g(x) = \int_{-\infty}^{\infty} f(x, t) \mathrm{d}t$
对于一个局部核函数，它的性质为：给距离x较近的数据点x_n较高的权重。注意，核满足加和限制:

$$\sum_{n=1}^{N} k(x, x_n) = 1$$

这个模型不仅定义了条件期望，还定义了整个条件概率分布:

$$p(t|x) = \frac{p(t, x)}{\int p(t, x) \mathrm{d}t} = \frac{\sum_n f(x - x_n, t - t_n)}{\sum_m \int f(x - x_m, t - t_m) \mathrm{d}t}$$

为了举例说明，我们考虑一元输入变量x的情形，其中$f(x, t)$由变量$z = (x, t)$上的一个零均值各向同性的高斯分布给出，方差为σ^2。对应的条件分布由高斯混合模型给出。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$p(t, x)$建模，这个混合高斯模型使用第9章讨论的方法训练，然后找到对应的条件概率分布$p(t|x)$。在后一种情况中，模型不再由训练数据点处的核函数表示，但是混合模型中分量的个数会小于训练数据点的个数，从而使得生成的模型对于测试数据点能够更快地计算。为了能够生成一个预测速度较快的模型，我们可以接受训练阶段的计算开销。

6.4 高斯过程

我们想要知道在贝叶斯方法中，核是如何引入的。
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。乍一看来，在函数组成的不可数的无穷空间中对概率分布进行计算似乎很困难。但是，正如我们将看到的那样，对于一个有限的训练数据集，我们只需要考虑训练数据集和测试数据集的输入x_n处的函数值即可，因此在实际应用中我们可以在有限的空间中进行计算。

6.4.1 重新考虑线性回归问题

考虑模型M，它被定义为$\phi(x)$的元素给出的M个固定基函数的线性组合，即：

$$y(x) = w^T \phi(x)$$

其中x是输入向量，w是M维的权向量。现考虑w上的先验分布，这是一个各向同性的高斯分布，形式为：

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

它是由超参数α控制的，表示分布的精度。对于任意给定的w，$y(x) = w^T \phi(x)$定义了x的一个特定的函数，于是，在上面$p(w) = \mathcal{N}(0, \alpha^{-1} I)$定义的w上的概率分布就产生了一个函数$y(x)$上的概率分布。在实际应用中，我们希望计算这个函数在某个具体的x处的函数值，于是我们感兴趣的是函数值$y(x_1, \cdots, y(x_N))$的概率分布，我们将函数值的集合记作向量$\mathbf{y}$,向量等价于：

$$\mathbf{y} = \mathbf{\Phi w}$$

其中$\mathbf{\Phi}$是设计矩阵，元素为$\Phi_{nk} = \phi_k(x_n)$。
首先，我们注意到y是由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：

$$\mathbb{E}[\mathbf{y}] = \mathbf{\Phi} \mathbb{E}[w] = 0$$
$$\mathbf{cov}[\mathbf{y}] = \mathbb{E}[\mathbf{y} \mathbf{y^T}] = \mathbf{\Phi} \mathbb{E}[w w^T] \mathbf{\Phi}^T = \frac{1}{\alpha} \mathbf{\Phi} \mathbf{\Phi}^T = K$$

其中K是Gram矩阵，元素为：

$$K_{nm} = k(x_n, x_m) = \frac{1}{\alpha} \phi(x_n)^T \phi(x_m)$$

$k(x, x')$是核函数。
通常来说，高斯过程被定义为函数$y(x)$上的一个概率分布，使得在任意点集$x_1, \cdots, x_N$ 处计算的$y(x)$的值的集合联合起来服从高斯分布。
在输入向量x是二维的情况下，这也可以被称为高斯随机场。更一般地，可以用一种合理的方式为$y(x_1), \cdots, y(x_N)$赋予一个联合的概率分布，来确定一个随机过程（stochastic process）$y(x)$。
高斯随机过程的关键点在于N个变量$y_1, \cdots, y_N$上的联合概率分布完全由二阶统计（均值和协方差）确定。大部分情况，我们对于$y(x)$的均值没有任何先验知识，因此根据对称性，我们将其设为0.这就等价于在基函数的观点中，令权值$p(w|\alpha)$先验概率分布的均值为0.之后，高斯过程的确定通过给定两个x处的函数值$y(x)$的协方差来完成。这个协方差由核函数确定:

$$\mathbb{E}[y(x_n) y(x_m)] = k(x_n, x_m)$$

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。

6.4.2 用于回归的高斯过程

为了将高斯过程模型应用于回归问题，我们需要考虑观测目标值的噪声，形式为：

$$t_n = y_n + \epsilon_n$$

其中$y_n = y(x_n)$,ε_n是一个随机噪声变量，它的值对于每个观测n是独立的。
我们考虑服从高斯分布的噪声过程：

$$p(t_n|y_n) = \mathcal{N}(t_n|y_n, \beta^{-1})$$

其中β是一个超参数，表示噪声精度，所以目标值$\mathbf{t}$的联合概率分布是一个各向同性的高斯分布，形式为：

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

根据高斯过程的定义，边缘概率分布$p(\mathbf{y})$是一个高斯分布，均值为0，协方差由Gram矩阵K定义，即：

$$p(\mathbf{y}) = \mathcal{N}(\mathbf{y}|0, K)$$

确定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x_n和x_m，对应的值$y(x_n)$和$y(x_m)$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$x_1, \cdots, x_N$为条件的边缘概率分布$p(\mathbf{t})$，需要对$\mathbf{y}$进行积分：

$$p(\mathbf{t}) = \int p(\mathbf{t}|\mathbf{y})p(\mathbf{y})\mathrm{d}\mathbf{y} = \mathcal{N}(\mathbf{t}|0, C)$$

其中协方差矩阵C的元素为：

$$C(x_n, x_m) = k(x_n, x_m) + \beta^{-1}\delta_{nm}$$

这个结果反映了下面的事实：两个随机的高斯分布（即与$y(x)$相关的高斯分布和与ε相关的高斯分布）是独立的，因此它们的协方差可以简单地相加。

对于高斯过程回归，一个广泛使用的核函数的形式为指数项的二次型加上常数和线性项，即:

$$k(x_n, x_m) = \theta_0 \exp\left\{-\frac{\theta_1}{2}||x_n - x_m||^2\right\} + \theta_2 + \theta_3 x_n^T x_m$$

注意，涉及到θ_3的项对应于一个参数模型，这个模型是输入变量的线性函数。
现在我们已经定义数据集上的联合概率分布的模型，现在要在给定一组训练数据的情况下，对新输入变量预测目标变量值。我们假设$\mathbf{t_N} = (t_1, \cdots, t_N)^T$，对应于输入值$x_1, \cdots, x_N$，组成观测训练集，

并且我们的目标是对于新的输入向量x_{N+1}预测目标变量t_{N+1}。这要求我们计算预测分布$p(t_{N+1}|\mathbf{t}_N)$。注意，这个分布还要以变量$x_1,\cdots,x_N$ 和x_{N+1}为条件。但是为了记号的简介，我们不会显式地写出这些条件变量。
为了找到$p(t_{N+1}|\mathbf{t}_N)$，先写下联合概率分布$p(\mathbf{t}_{N+1})$，然后求出条件概率分布。
由上面的公式可知：

$$p(\mathbf{t}_{N+1})=\mathcal{N}(\mathbf{t}_{N+1}|0,C_{N+1})$$

协方差矩阵C_{N+1}可以分块得到：

$$C_{N+1}=\begin{bmatrix}C_N & k\\ k^T & c\end{bmatrix}$$

其中C_N由上面的公式给出，k的元素为：$k(x_n,x_{N+1})$,标量$c=k(x_{N+1},x_{N+1})$。
我们就可以根据得到的联合概率分布进行积分求出条件概率分布$p(\mathbf{t}_{N+1}|\mathbf{t})$是一个高斯分布，均值和协方差：

$$m(x_{N+1})=k^TC_N^{-1}\mathbf{t}$$
$$\sigma^2(x_{N+1})=c-k^TC_N^{-1}k$$

核函数的唯一的限制是公式$C(x_n,x_m)=k(x_n,x_m)+\beta^{-1}\delta_{nm}$给出的协方差矩阵一定是正定的，如果$\lambda_i$是K的一个特征值，则C的对应的特征值就是$\lambda_i+\beta^{-1}$，因此对于任意点对$x_n,x_m$核矩阵$k(x_n,x_m)$一定是正定的，所以$\lambda_i>0$,而$\beta>0$，所以$\lambda_i+\beta^{-1}>0$。
而预测分布的均值可以写成x_{N+1}的函数，形式为：

$$m(x_{N+1})=\sum_{n=1}^{N}a_nk(x_n,x_{N+1})$$

其中a_n是$C_N^{-1}\mathbf{t}$的第n个元素如果核函数$k(x_n,x_m)$只依赖于距离$||x_n-x_m||$，就得到径向基函数的一个展开。

根据上面的结果可以定义具有任意核函数$k(x,x\prime)$的高斯过程回归。如果核函数$k(x,x\prime)$根据基函数的有限集定义，那么我们就可以从高斯过程的观点开始，推导出之前在3.3.2节得到的线性回归的结果
因此对于这种模型，我们既可以通过参数空间的观点使用线性回归的结果得到预测分布，也
可以通过函数空间的观点使用高斯过程的结果得到预测分布。
使用高斯过程的核心计算涉及到对$N\times N$ 的矩阵求逆。标准的矩阵求逆方法需要$O(N^3)$次计算。相反，在基函数模型中，我们要对一个$M\times M$ 的矩阵S_N 求逆，这需要$O(M^3)$次计算。注意，对于两种观点来说，给定训练数据，矩阵求逆的计算必须进行一次。对于每个新的测试数据，两种方法都需要进行向量-矩阵的乘法，这在高斯过程中需要$O(N^2)$次计算，在线性基函数模型中需要$O(M^2)$次计算。如果基函数的数量M 比数据点的数量N 小，那么使用基函数框架计算会更高效。但是，高斯过程观

点的一个优点是，我们可以处理那些只能通过无穷多的基函数表达的协方差函数。但是，对于大的训练数据集，直接应用高斯过程方法就变得不可行了，因此一系列近似的方法被提出来。
与精确的方法相比，这些近似的方法关于训练数据集的规模有着更好的时间复杂度.

6.4.3 学习超参数

高斯过程模型的预测部分依赖于协方差函数的选择。在实际应用中，我们不固定协方差函数，而是更喜欢使用一组带有参数的函数，然后从数据中推断参数的值。这些参数控制了相关性的长度缩放以及噪声的精度等等，对应于标准参数模型的超参数。
学习超参数的方法是基于计算似然函数$p(\mathbf{t}|\theta)$的，其中θ表示高斯过程模型的超参数，可以通过最大化似然函数的方法进行θ的点估计。
使用多元高斯分布标准形式，可以得到对数似然函数的形式：

$$\ln p(\mathbf{t}|\theta) = -\frac{1}{2}\ln|C_N| - \frac{1}{2}\mathbf{t}^T C_N^{-1}\mathbf{t} - \frac{N}{2}\ln 2\pi$$

我们可以得到：

$$\frac{\partial}{\partial\theta_i}\ln p(\mathbf{t}|\theta) = -\frac{1}{2}\mathbf{Tr}\left(C_N^{-1}\frac{\partial C_N}{\partial\theta_i}\right) + \frac{1}{2}\mathbf{t}^T C_N^{-1}\frac{\partial C_N}{\partial\theta_i}C_N^{-1}\mathbf{t}$$

由于$\ln p(\mathbf{t}|\theta)$是一个非凸函数，所以有多个极大值点。

引入θ上的先验之后最大化对数后验是很容易的，我们在纯粹的贝叶斯方法中，需要计算θ的边缘概率，但是通过精确的积分或者是求和是不可行的，需要近似。
高斯过程回归模型给出的预测分布的均值和方差是输入向量x的函数。然而，我们已经假定由参数 控制的附加噪声对预测方差的贡献是常数。对于一些被称为异方差的问题，噪声方差本身也依赖于x。为了对这种问题进行建模，我们可以对高斯过程框架进行推广，引入第二个高斯过程来表示 对于输入x的依赖性由于 是一个方差，因此是非负的，所以我们使用高斯过程来对ln (x)进行建模。

6.4.4 自动相关性确定

我们通过最大似然方法进行参数最优化，能将不同的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(ARD)的一个例子。
对于二维输入空间$x = (x_1, x_2)$，有核函数：

$$k(x, x') = \theta_0 \exp\left\{-\frac{1}{2}\sum_{i=1}^{2}\eta_i(x_i - x_i')^2\right\}$$

随着η_i的减小，函数逐渐对输入变量x_i变得不敏感。通过使用最大似然法按照数据集调整这些参数，它

可以检测到对于预测分布几乎没有影响的输入变量，因为对应的η_i会很小。这在实际应用中很有用，因为它使得这些输入可以被遗弃。
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，若D是输入空间的维数。

$$k(x_n, x_m) = \theta_0 \exp\left\{ -\frac{1}{2}\sum_{i=1}^{D}\eta_i(x_{xi} - x_{mi})^2 \right\} + \theta_2 + \theta_3 \sum_{i=1}^{D} x_{ni}x_{mi}$$

6.4.5 用于分类的高斯过程

我们的目标是在给定一组训练数据的情况下，对于一个新的输入向量，为目标变量的后验概率建模。这些概率一定位于区间(0, 1)中，而一个高斯过程模型做出的预测位于整个实数轴上。然而，我们可以很容易地调整高斯过程，使其能够处理分类问题。方法为：使用一个恰当的非线性激活函数，将高斯过程的输出进行变换。
对于一个二分类问题，目标变量为$t \in \{0, 1\}$,如果我们定义在$a(x)$上的高斯过程，然后用logistic sigmod 函数进行变换，就得到了一个非高斯随机过程，目标变量t上的概率分布为：

$$p(t|a) = \sigma(a)^t(1 - \sigma(a))^{(1-t)}$$

我们的目标是确定$p(t_{N+1}|\mathbf{t}_N)$,我们引入向量$\mathbf{a}_{N+1}$上的高斯过程先验，这反过来定义了$\mathbf{t}_{N+1}$上的非高斯过程。以训练数据$\mathbf{t}_N$为条件，得到求解的预测分布。a_{N+1}上的高斯过程先验形式为：

$$p(\mathbf{a}_{N+1}) = \mathcal{N}(\mathbf{a}_{N+1}|0, C_{N+1})$$

与回归的情形不同，协方差矩阵不再包含噪声项，因为我们假设所有的训练数据点都被正确标记。然而，由于数值计算的原因，更方便的做法是引入一个由参数ν控制的类似噪声的项，它确保了协方差矩阵是正定的。因此协方差矩阵C_{N+1}的元素为:

$$C(x_n, x_m) = k(x_n, x_m) + \nu\delta_{nm}$$

k是一个半正定核函数，而且由参数θ控制。
对于二分类问题，预测分布为：

$$p(t_{N+1}|\mathbf{t}_N) = \int p(t_{N+1}|a_{N+1})p(a_{N+1}|\mathbf{t}_N)\mathrm{d}a_{N+1}$$

该积分无法精确计算，但我们可以计算logistic sigmod 与高斯分布的卷积，所以我们需要对$p(a_{N+1}|\mathbf{t}_N)$进行高斯近似。通常对后验概率进行高斯近似的理由是，根据中心极限定理，随着数据点数量的增加，真实的后验概率将会趋向于一个高斯分布。在高斯过程的情形中，变量的数量随着数据点数量的增多而增多，因此这个结果不能直接应用。然而，如果我们考虑增加落在x空间的固定区域中的数据点的数量，那么函数a(x)中对应的不确定性就会减小，这就渐近地趋近于高斯分布。
我们有三种获得高斯近似的方法：

- 基于变分推断并且使用了logistic sigmoid函数的局部变分界。这使得sigmoid函数的乘积可以通过高斯的乘积近似，因此使得对a_N 的积分可以解析地计算。这种方法也产生了似然函数$p(t_N|\theta)$的下界。通过使用softmax函数的高斯近似，高斯过程分类的变分法框架也可以扩展到多类（K > 2）问题
- 第二种方法使用期望传播,由于真实的后验概率是单峰的，期望传播方法可以给出很好的结果
- 第三种高斯过程分类的方法基于拉普拉斯近似

6.4.6 拉普拉斯近似

为了计算预测分布，我们寻找a_{N+1}的后验概率分布的高斯近似。使用贝叶斯定理，后验概率分布为:

$$\begin{aligned} p(a_{N+1}|\mathbf{t}_N) &= \int p(a_{N+1}, \mathbf{a}_N|\mathbf{t}_N) \\ &= \frac{1}{p(\mathbf{t}_N)} \int p(a_{N+1}, \mathbf{a}_N) p(\mathbf{t}|a_{N+1}, \mathbf{a}_N) \mathrm{d}\mathbf{a}_N \\ &= \frac{1}{p(\mathbf{t}_N)} \int p(a_{N+1}, \mathbf{a}_N) p(\mathbf{a}_N) p(\mathbf{t}|\mathbf{a}_N) \mathrm{d}\mathbf{a}_N \\ &= \int p(a_{N+1}|\mathbf{a}_N) p(\mathbf{a}_N|\mathbf{t}_N) \mathrm{d}\mathbf{a}_N \end{aligned}$$

而根据前面高斯过程得到的结果，可以知道:

$$p(a_{N+1}|\mathbf{a}_N) = \mathcal{N}(a_{N+1}|k^T C_N^{-1}, c - k^T C_N^{-1} k)$$

所以只需要找到$p(\mathbf{a}_N|\mathbf{t}_n)$的拉普拉斯近似即可:
先验概率$p(\mathbf{a}_N)$由一个零均值高斯过程给出，协方差矩阵为C_N ，数据项（假设数据点之间具有独立性）为:

$$p(\mathbf{t}_N|\mathbf{a}_N) = \prod_{n=1}^{N} \sigma(a_n)_n^t (1 - \sigma(a_n))^{1-t_n} = \prod_{n=1}^{N} e^{a_n t_n} \sigma(-a_n)$$

我们然后通过对$p(\mathbf{a}_N|\mathbf{t}_N)$的对数进行泰勒展开，就可以得到拉普拉斯近似。忽略掉一些具有可加性的常数，这个概率的对数为:

$$\begin{aligned} \Phi(\mathbf{a}_N) &= \ln p(\mathbf{a}_N) + \ln p(\mathbf{t}_N|\mathbf{a}_N) \\ &= -\frac{1}{2}\mathbf{a}_N^T C_N^{-1} \mathbf{a}_N - \frac{N}{2}\ln 2\pi - \frac{1}{2}\ln|C_N| + \mathbf{t}_N^T \mathbf{a}_N - \sum_{n=1}^{N} \ln\left(1 + e^{a_n}\right) \end{aligned}$$

我们需要找到其众数,需要计算其梯度，该梯度为:

$$\nabla \Phi(\mathbf{a}_N) = \mathbf{t}_N - \sigma_N - C_N^{-1}\mathbf{a}_N$$

我们使用基于Newton-Raphson方法的迭代的方法，它给出了一个迭代重加权最小平方（IRLS）算法。这需要求出 $\Phi(\mathbf{a}_N)$的二阶导数，而这个二阶导数也需要进行拉普拉斯近似，结果为:

$$\nabla\nabla \Phi(\mathbf{a}_N) = -W_N - C_N^{-1}$$

使用迭代公式，可以得到:

$$\mathbf{a}_N^{新} = C_N(I + W_N C_N)^{-1}\{\mathbf{t}_N - \sigma_N + W_N\mathbf{a}_N\}$$

直到收敛于众数
一旦我们找到了后验概率的众数a_N^* ，我们就可以计算Hessian矩阵，结果为:

$$H = -\nabla\nabla \Phi(\mathbf{a}_N) = W_N + C_N^{-1}$$

所以得到高斯近似:

$$q(\mathbf{a}_N) = \mathcal{N}(\mathbf{a}_N|\mathbf{a}_N^*, H^{-1})$$

然后得到最后$p(a_{N+1}|\mathbf{t}_N)$的结果:

$$\mathbb{E}[a_{N+1}|\mathbf{t}_N] = k^T(\mathbf{t}_N - \sigma_N)$$
$$\mathbf{var}[a_{N+1}|\mathbf{t}_N] = c - k^T(W_N^{-1} + C_N)^{-1}k$$

我们还需确定协方差函数的参数θ,一种方式是最大化似然函数$p(\mathbf{t}_N|\theta)$,还可以加上正则化，产生正则化的最大似然解，最大似然函数定义为:

$$p(\mathbf{t}_N|\theta) = \int p(\mathbf{t}_N|\mathbf{a}_N)p(\mathbf{a}_N)\mathrm{d}\mathbf{a}_N$$

这个积分没有解析解，所以我们需要再次使用拉普拉斯近似，我们可以计算对数似然函数的梯度，然后使用标准非线性优化算法来确定θ的值,很容易将拉普拉斯近似推广到涉及K > 2个类别的使用**softmax**激活函数的高斯过程

6.4.7 与神经网络的联系.

我们已经看到，神经网络可以表示的函数的范围由隐含单元的数量M 控制，并且对于足够大的M ，一个两层神经网络可以以任意精度近似任意给定的函数。在最大似然的框架中，隐含单元的数量需要有一定的限制（根据训练集的规模确定限制的程度），来避免过拟合现象。然而，从贝叶斯的角度看，根据训练集的规模限制参数的数量几乎毫无意义。
在贝叶斯神经网络中，参数向量w上的先验分布以及网络函数$f(x, w)$产生了函数$y(x)$上的先验概率分

布，其中y是网络输出向量。
在极限$M \to \infty$的情况下，对于w的一大类先验分布，神经网络产生的函数的分布将会趋于高斯过程。然而，应该注意，在这种极限情况下，神经网络的输出变量会变为相互独立。神经网络的优势之一是输出之间共享隐含单元，因此它们可以互相“借统计优势”，即与每个隐含结点关联的权值被所有的输出变量影响，而不是只被它们中的某一个影响。这个性质在极限状态下的高斯过程中丢失了。
我们已经看到，高斯过程由它的协方差（核）函数确定。在两种具体的隐含单元激活函数（**probit**和高斯）下，协方差的显式形式。这些核函数$k(x, x\prime)$是非静止的，即它们不能够表示为差$x - x\prime$的函数，这是因为以零为中心的高斯权值先验破坏了权空间的平移不变性。
通过直接对协方差函数计算，我们隐式地在权值的分布上进行了积分。如果权值先验由超参数控制，那么它们的值会确定函数的分布的长度标度.