ChillyRain's Blog

Happy coding

PRML Notes - 3.4 Bayesian Model Comparison

贝叶斯方法避免过拟合是通过边缘化模型参数而实现的，而不是像极大似然那样来做点估计。这样，不同的模型可以使用全部数据来做训练，并进行比较，而不是像CV那样需要拆出一部分来做检验。

假设我们有L个模型，$M_i, i = 1...L$。训练数据是从其中一个模型中生成的，即其中有一个是真实模型，但是我们并不确定是哪一个。这种先验的不确定性可以通过先验的均匀分布来表达，而相应的后验分布则表达了我们在观测到数据后，对各个模型的偏好程度。

$$p(M_i|D) \propto p(M_i)p(D|M_i)$$

由于前面使用的均匀分布做为先验，区分度主要落在了model evidence上面，即$p(M_i|D)$，它表达了训练数据对各个模型的偏好程度。这个值也称为边缘似然性，因为它是对整个模型的似然性（不是针对某一个参数值）。两个模型的model evidence的比例称为贝叶斯因子。

$$BayesFactor = \frac{p(D|M_i)}{p(D|M_j)}$$

一旦我们得到的对各个模型的后验分布，我们就可以使用它来形成一个混合的预测分布，各个模型以后验概率做为权重对其输出的响应值进行加权。一种简化的方式是使用后验最高的模型来做预测，这个选择的过程就称为模型选择。

对于一个参数模型，model evidence的表达式为

$$p(D|M_i) = \int p(D|w, M_i)p(w|M_i)dw$$

做一些近似的考虑，假设参数的后验分布尖锐的分布在最大值w_{MAP}周围，宽度为$\delta w_{posterior}$，并假设参数的先验为$p(w) = 1/\delta w_{prior}$，那么model evidence可以写成

$$p(D) = \int p(D|w)p(w)dw = p(D|w_{MAP})\frac{\delta w_{posterior}}{\delta w_{prior}}$$

对应地，其对数似然性为

$$lnp(D) = lnp(D|w_{MAP}) + \frac{\delta w_{posterior}}{\delta w_{prior}}$$

由上式可知，当似然性越高时，evidence值越大，然而当参数值极好地拟合训练数据时，evidence越小，即模型复杂度越高，evidence越小。最优的模型应该是在两都之间取一个折中。

从模型生成数据集的角度来看，实际上，过于简单的模型灵活性不够，只能以较低的可能性生成给定的训练数据，而过于复杂的模型则灵活性过大，可能产生的数据集种类更多，产生给于训练数据的可能性也变低（概率被平分）。

对于一个特定的数据集，有可能存在非真实模型的model evidence高于真实模型，而期望意义上这是不可能存在的。假设有两个模型，其中M_1为真实模型，那么二者在所有数据集上的平均贝叶斯因子表达式为

$$\int p(D|M_1)ln\frac{p(D|M_1)}{p(D|M_2)}dD$$

这个统计量称为KL距离，当且仅当两个分布相同是取零，其它情况下均大于0，因此平均意义下贝叶斯因子是偏向于选择真实模型的。

与其它方法一样，贝叶斯模型选择是对模型的形式有所假设的。例如，当先验分布是improper的时候，model evidence不是良定义的。但是先得到两个模型的贝叶斯因子的表达式，再取极限仍然有可能得到一些有用的结论。

虽然贝叶斯模型比较方法可以省去CV的过程，但是保留一份独立的测试数据来做最终系统的效果评估仍然是明智之举。

Posted by *Chilly_Rain* 2012年6月12日 22:41

Category: Pattern Recognition and Maching Learning Tag: Comment: (1)

PRML Notes - 3.3 Bayesian Linear Regression

贝叶斯框架下的线性回归解决了频率框架下出现的过拟合问题，它是通过在推演阶段对参数引入先验分布，以及在决策阶段对参数值做积分来实现的。

参数分布

假设噪声精度β是已知的，那么似然性函数$p(t|\mathbf{w})$就是一个未知参数$\mathbf{w}$的二次函数的幂。根据第二章的内容，其对应的共轭先验分布为正态分布

$$p(\mathbf{w}) = N(\mathbf{w}|m_0, S_0)$$

而其后验分布同样也是正态分布

$$p(\mathbf{w}|\mathbf{t}) = N(\mathbf{w}|m_N, S_n)$$

$$m_N = S_N(S_0^{-1}m_0 + \beta\Phi^T\mathbf{t})$$

$$S_N^{-1} = S_0^{-1} + \beta\Phi^T\Phi$$

最 大后验参数值$\mathbf{w}_{MAP} = \mathbf{m}_N$，当先验的方差无穷大时，即$S_0 = \alpha^{-1}\mathbf{I}, \alpha -> 0$时，后验分布的均值就退化成了最大似然估计值。类似地，当N=0时，后验分布就退化成了先验分布。当数据点是一个接一个获得时，上次计算得到的后验分 布就可以作为下次计算时的先验分布，从而达到顺序计算的效果。

如果假设先验分布是一个零期望等方差的正态分布，那么其形式可以表示为

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

$$m_N = \beta S_N \Phi^T \mathbf{I}$$

$$S_N^{-1} = \alpha \mathbf{I} + \beta \Phi^T \Phi$$

根据贝叶斯定理，后验分布的对数可以表达为似然的对数，加上先验分布的对数，最终形成一个参数$\mathbf{w}$的函数，其形式是一个带规则化项的误差函数，其规则化项前的系数为$\lambda = \alpha/\beta$。

$$lnp(w|t) = -\frac{\beta}{2}\sum_{n=1}^{N}(t_n - w^T\phi(x_n))^2 - \frac{\alpha}{2}w^Tw + const$$

一般化的正态先验

$$p(w|\alpha) = [\frac{q}{2}(\frac{\alpha}{2})^{1/q}]^M exp(-\frac{\alpha}{2}\sum_{j=1}^{M}|w_j|^q)$$

预测分布

在得到了参数的后验分布之后，我们并不做实际做些什么，真正要得到的是预测分布，即结合参数的后验分布来为新的输入做出合适的预测值。

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

其中忽略了输入向量x以简化表达式。可以发现这个等式的右侧实际是两个正态分布的卷积，因而其结果也是一个正态分布，其期望和方差分别为

$$\mu = \mathbf{m}_N^T \mathbf{S}_N \phi(x)$$

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

期望值即为参数后验的期望与输入向量的内积；而方差值则可以分为两项，前者为数据噪声所引起的波动，而后者则可以看作参数值$\mathbf{w}$的波动，二者是独立可累加的。可以证明方差值是随着N值的变大而递减并趋向于0的，当样本足够大时，预测值的波动仅由训练数据中的噪声所带来。

预测分布的方差在训练数据点的附近最小，因而当数据点足够多时，预测分布的波动性也随之降低。

更一般地，如果参数w, β都被看作未知的，那么共轭先验就应该是一个Gaussian-gamma分布，而对应的预测分布就是一个学生t分布。

Equivalent Kernel

预测分布在一个点上的预测期望值可以表示为

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

其中定义函数

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

这个函数称为平滑矩阵或者equivalent kernel，而回归函数可以表达为训练数据响应值的线性加权，因而被称为线性平滑。特别地，当待预测点距离某训练数据点近时，该点的响应时对应的权重就高，反之则小（核函数的局部属性）。

可以观察两个待预测点的预测方差之间的关系

$$cov(y(x),y(x'))=cov(\phi(x)^Tw,w^T\phi(x'))=\phi(x)^T\mathbf{S}_N\phi(x')=\beta^{-1}k(x,x')$$

其中cov是相对于w来说的。上式说明，在输入空间中相邻的两个点的预测期望值相关性较大，相反地，相距较远的两个点之间则相关性较小。

核函数给了我们一种新的思路，即不通过定义基函数和回归来实现预测，而是定义一个核函数，并使用训练数据的输出做线性加权而得到。（高斯过程）

该核函数还满足归一化的性质，但是并不保证其中的每个子项都非负。

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

另外还有一个核函数都满足的特征（不限于Equivalent Kernel），它们都可以表达为非线性函数的乘积的形式

$$k(x,z)=f(x)f(z)$$

$$f(x)=\beta^{1/2}\mathbf{S}_N^{1/2}\phi(x)$$

Posted by *Chilly_Rain* 2012年5月20日 00:36

Category: Pattern Recognition and Maching Learning Tag: Comment: (0)

PRML Notes - 3.2 Bais-Variance Decomposition

在第一章中，我们把回归问题分解为两个部分：推演和决策。假设推演步骤已经完成，即后验分布$p(t|\mathbf{x})$已经得到，接下来就需要选择一种损失函数，并以最小化损失为目的进行决策的步骤。一种常用的损失函数为平方损失（注：请明确区分推演过程中使用的平方误差与决策步骤中使用的平方损失），在此条件下，最优预测值是通过以下条件期望得到的

$$h(\mathbf{x})=E[t|\mathbf{x}]=\int tp(t|\mathbf{x})dt$$

平方损失可以写为以下形式

$$E[L]=\int(y(x)-h(x))^2p(x)dx+\int(h(x)-t)^2p(x,t)dxdt$$

其中，第一项表达我们得到的预测与最优预测之间的差异，而第二项则表达了最优预测与真实响应值之间的差异，这项实际是以噪声引起的，是我们是不可控的。理想情况下，我们的预测与最优预测一致，这样就完全去掉了第一项，从而最小化平方损失。

实 际中，这个最优预测$y(\mathbf{x})=h(\mathbf{x})$是无法得到的，因为它的计算需要无限多的数据，而我们只有有限个数据点。因此，我们只能寄望于先找到一个模型$y(\mathbf{x},\mathbf{w})$，再通过训练得到最优参数以最大程度的接近于$h(\mathbf{x})$。

由 于数据是有限的，因此通过这些数据来训练而得到的模型是具有不确定性的。从贝叶斯的角度来看，这种不确定性是通过参数$\mathbf{w}$的分布来表达的。而在频率框架下，最终得到的并不是一个关于参数的分布，而是一个关于参数的点估计。不同的数据样本对应不同的参数点估计值，其不确定 性表达为不同数据训练造成的参数的波动以及损失值的波动，亦即

$$E_D[(y(\mathbf{x};D)-h(\mathbf{x}))^2]=(E_D[y(\mathbf{x};D)-h(\mathbf{x})])^2+E_D[(y(\mathbf{x};D)-E_D[y(\mathbf{x};D)])^2]$$

其中，第一项表达了该模型的平均预测与最优预测的差异（bias），而第二项则表达了该模型预测值自身的方差（variance）。结合前两个表达式，我们可以得到以下等式关系

$$expected\_loss=(bias)^2+variance+noise$$

我 们希望最小化损失值，但是实际上我们将不得在bias和variance之间做出权衡。当我们选用了一个非常复杂灵活的模型时，它能够很有效地降低 bias，却引入了较大的vairance；相反地，如果我们选用了一个严格的模型，它的variance较低，但是却引入了bias风险。最优的模型选 择就是找到bias和variance之间的一个最优点。

这个Bias-Variance Decomposition在实际中应用价值并不大，因为它需要计算E_D，而这是不可能的。理论意义上，这个分解式还是给我们带来了一些insightful ideas.

Posted by *Chilly_Rain* 2012年5月20日 00:34

Category: Pattern Recognition and Maching Learning Tag: Comment: (0)

PRML Notes - 3.1 Linear Basis Function Model

书中第三章的标题是Linear Model for Regression，即线性回归模型。线性模型的定义是相对于“未知参数”的线性函数。也就是说，虽然下面这个模型可以称为线性模型（实际上是最简单的线性模型，因为它不仅相对于未知参数是线性的，而且相对于输入变量也是线性的），但是其实际的定义要宽泛得多，因而才有了本节介绍的“线性基函数”模型。

$$y(x,w)=w_0+w_1x_1+\ldots+w_Dx_D$$

回归问题

给定训练数据集$(\mathbf{x}_i,t_i),\ i=1\ldots N$，其中$\mathbf{x}_i$为D维变量，$t_i$为响应值。

最简单的方法，我们希望通过数据得到一个响应值估计函数$y(\mathbf{x})$，从而可以对未来的输入变量进行响应值预测。从概率的观点来看，我们要得到的不仅是一个输入变量的单点估计值，而是一个条件分布$p(t|\mathbf{x})$，从而可以表达其响应值的不确定性。得到这个分布，实际我们就完成了“推演”的工作，“决策”步骤中，我们需要定义一个误差函数（比如第一章中提到的平方误差），再以最小化期望误差为目的来决策每个输入变量的响应值（在平方误差下，这个“决策”是响应值t的条件期望值）。

基函数（basis function）

相比于最开始提到的那个“最简单”的线性模型，更一般的线性模型可以表达为

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

其中所有的 $\phi_j(\mathbf{x})$ 都称为*基函数*。特别地，$\phi_0(\mathbf{x}) = 1$ 以容许 w_0 对函数的值产生一定的偏移(bias)。基函数的作用可以表现的*预处理（如特征抽取）*上，如PCA，MDS等经典的特征抽取算法都是以原坐标为基础进行线性组合，形成新坐标，再把数据表示的新的坐标下，以图可以更好地发现数据中的规律（例如PCA是把数据中方差最大的方向作为新的坐标体系）。而数据在新坐标体系下的坐标值就可以用这M个基函数来表示。

虽然是作用于线性模型中，基函数并不必须是线性函数，从而极大地扩充了最简单线性模型的表达能力。多项式模型就是非线性基函数的一个实例。

spline function：非输入变量的全局函数，因此可以把输入空间划分成独立的部分，一部分的变化不会影响其它部分（不懂，待研究）

除多项式基函数外，其它一些基函数的例子包括

- Gaussian基函数：$\phi_j(x) = exp(-\frac{(x-\mu_j)^2}{2s^2})$
- sigmoidal基函数：$\phi_j(x) = \sigma(\frac{x-\mu_j}{s})$，其中$\sigma(a) = \frac{1}{1+exp(-a)}$
- tanh基函数：$tanh(a) = 2\sigma(a) - 1$，因此sigmoidal基函数的线性组合也是tanh基函数的线性组合
- Fourier基函数：晕

极大似然法和最小二乘法

假设响应值t是由一个以x为输入的确定的函数加上一个高斯噪声而得到的，即

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

其中ϵ 是符合正态分布 $N(0,\beta^{-1})$ 的噪声变量。那么我们希望得到的条件分布可以写作

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

如果假设我们在决策时所选的误差函数就是平方误差，那么决策函数就是以上分布的条件期望，即 $y(\mathbf{x},\mathbf{w})$。

这里需要注意的一点是，高期噪声假设在多峰值的场景下是不适合的。

假设给定的N个数据是独立同分布的（iid），那么对数似数函数可写作

$$lnP(\mathbf{t}|\mathbf{x},\mathbf{w},\beta) = \sum_{i=1}^{N} lnN(t_i|\mathbf{w}^T\phi(\mathbf{x_i}),\beta^{-1}) = \frac{N}{2}ln\beta - \frac{N}{2}ln(2\pi) - \beta E_D(\mathbf{w})$$

其中 $E_D(\mathbf{w})$ 为Sum-of-square误差函数（请参见第一章）。当我们使用求导的方法来计算参数 $\mathbf{w}$ 时会发现，正如第一章所述，由于表达的前两项没包含 $\mathbf{w}$，因此对ML和LS求导实际是等价的。

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

这个等式称为最小二乘问题的正规式(normal equations)。而Φ是一个N*M阶的矩阵，其中$\Phi_{ij} = \phi_j(x_i)$。 等号右侧除t之外的部分称为矩阵Φ的伪逆矩阵，可以看作是非方阵的逆阵。特别地，当Φ是方阵时，伪逆阵等于逆阵。

下面对w_0的意义做深入探索。对$E_D(\mathbf{w})$稍作变形，可得

$$E_D(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{N}(t_i - w_0 - \sum_{j=1}^{M-1} w_j\phi_j(x_i))^2$$

再对w_0对导数，可解得

$w_0 = \bar{t} - \sum_{j=1}^{M-1} w_j\bar{\phi_j}$，其中$\bar{t} = \frac{1}{N}\sum_{i=1}^{N} t_i$，$\bar{\phi_j} = \frac{1}{N}\sum_{i=1}^{N}\phi_j(x_i)$。

因此w_0可以解释为训练集中平均响应值，与训练集平均基函数值的加权平均之间的差。

同样地，对极大似数函数相对于β进行求导，可得参数值为

$$\frac{1}{\beta} = \frac{1}{N}\sum_{i=1}^{N}(t_i - \mathbf{w}^T\phi(x_i))^2$$

β从上式中可解释为响应值围绕着回归函数值的波动程度，这也与其开始的定义“噪声方差”相一致。

最小二乘法的几何解释

如果把响应值

$$t = (t_1, t_2, \ldots, t_n)$$

看作n维空间中的一个向量，同时将每个基函数应用于所有$x_i, i = 1...n$形成M个基函数向量

$$\bar{\phi_j}, j = 1...M$$

的话，最小二乘法在几何意义上就可以解释为M个基函数向量形成的线性子空间中找到一个最优向量，使其与响应向量的欧氏距离最小。

进一步地，这个最优向量实际就是响应向量在这个基函数向量所形成的子空间中的投影，最优解就是基于M个基函数向量线权加权得到这个最优向量时的参数w, β。

在线学习

如果训练数据集非常大，那么使用其对模型进行一次训练可能是极其耗时的，此时使用在线学习（或者流式学习）算法将是一种更好的选择。在这种场景下，我们让数据一个接着一个地使用，当每获取一个数据点时，就基于这个新的数据点来更新模型的参数值。

我们可以采用一种称为随机梯度下降的方式来实现在线学习。具体地，如果n个数据点的误差函数可以表示为

$$E=\sum_n E_n$$

那么当第n个点到来时，模型参数的更新方式为

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

其中η为学习率，τ为迭代次数。参数的初始值可以随机选。特别地，对平方误差函数，上式可表示为

$$\mathbf{w}^{(\tau+1)}=\mathbf{w}^{(\tau)}-\eta(t_n-\mathbf{w}^{(\tau)^T}\phi_n)\phi_n$$

这个算法被称为LMS(least-mean-square)算法。

带有规则化项的最小二乘法

为了控制过拟合的出现，第一章中提到一种利用规则化项的方法，即最小化带有惩罚项的误差函数

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

其中λ为规则化项系数，用以控制两者之间的权衡关系。一种简单的规则化项称 二次规则化项，称为权重衰减，即

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

如果将原始误差替换为平方误差，那么整个误差函数就可以表示为

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

基于以上误差函数，通过最小二乘法得到的最优参数值为

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

实际上，其就是对最小二乘结果的一个简单扩展。

更加一般的规则化项可以表达为

$$\frac{1}{2}\sum_{j=1}^M|w_j|^q$$

其中当q=2的时候，即为权平方的规则化项。当q=1时，这个带规则化项的最小二乘法也称为套索(lasso)，当λ足够大时，它可以产生一个稀疏的模型，即让多数基函数对应的系数接近于0.

本质上，规则化项方法并没有从根本上解决过拟合问题，或者模型复杂度问题，它仅仅是将问题做了转移，从寻找合适数量的基函数，变为了如何确定规则化项系数λ。

本章下面内容均假设规则化项为二次规则化项。

多维响应值

一些情况下，每个x_n所对的输出t_n并不是一维的，而是多维的。最简单的方式就是为每一个响应维度确定一组基函数，从而把问题分解为多个线性回归问题。当然，也可以选择一组相同的基函数，来应付所有的响应维度。

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

同样通过最小二乘法来求解，可得

$$\mathbf{W_{ML}} = (\Phi^T\Phi)^{-1}\Phi^T\mathbf{T}$$

而每响应维度上，其对应的参数值为

$$\mathbf{w_k} = (\Phi^T\Phi)^{-1}\Phi^T\mathbf{t_k}$$

因此，我们可以将响应维度进行解耦，我们只需要计算一次Φ，然后就可以计算任何一个维度上的参数值。>

更一般地，正态分布的协方差阵可以不是严格对角阵，而可以是任意矩阵，我们同样可以对每个响应维度进行解耦。这是因为$\mathbf{W}$是只与分布的均值相同，而其估计值是可以独立于协方差来估计出来的，即它是独立于协方差的。（详见第一章）

Posted by *Chilly_Rain* 2012年1月31日 22:43

Category: Pattern Recognition and Maching Learning Tag: 基函数 线性回归模型 Comment: (0)

PRML Notes - 1.6 Information Theory

信息量和香农熵

一个变量取值的信息量可以看作是它带来的“使人惊讶的程度”，一个必然事件没有任何信息量，而一个极其偶然的事件的发生则会使人非常“惊讶”，因而包括大量信息。

自然地，信息量的概率就与变量的概率分布联系在了一起。香农熵（Shannon Entropy）成功表达了一个离散型变量所带来的平均信息量：

$$H(x) = -\sum_x p(x) log_2 p(x)$$

注意到$limit_{p->0} plnp = 0$，因此计算某个变量的香农熵时只考虑非零取值即可。另外，香农熵是非负的。

无噪声编码定理：香农熵是传递一个变量状态所需要的比特数的下界。也就是说，在期望意义下，对一个变量的取值进行编码所需要的最小的比特数即为香农熵。一般情况下，香农熵对数的底取2。

对于一个概率分布，当概率集中于较少的某几个取值时（绝大多数情况下变量会取少数的几个值之一），香农熵的值会较低，相反地，如果概率在各种取值上比较平均（几乎无法判断变量会取哪个值），那么香农熵会较高。使用拉格朗日乘子法（约束概率分布的归一化）计算香农熵的最大值，可知当概率分布是均匀分布时，香农熵可取到最大值$H = lnM$，其中M为变量的状态总数（所有可能取值的个数）。因此，香农熵也可以看作是一个变量*不确定度*的度量。

物理上关于香农熵的解释：mulitplicity, microstate, macrostate, weight

连续型变量的微分熵

对于一个连续型变量，无法直接使用上面香农熵的定义。可以近似地对连续型变量的取值进行离散化，将整个取值范围划分成宽度为Δ的小区域。均值定值告诉我们，在每个小区域内总存在一个值x_i，使得以下等式成立

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

因此，我们可以把每个落入第i个小区域的的点赋予x_i。这样，我们就可以套用离散型变量的香农熵公式

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

而当Δ趋近于0时，上式最右侧第二项趋近于0，而第一个项则趋近的表达式称为微分熵(differential entropy)：

$$H(x) = -\int p(x) lnp(x) dx$$

仍然使用拉格朗日乘子法，约束均值和方差，以及概率分布的归一化，可知在均值和方差一定的情况下，使微分熵最大的概率分布为正态分布。而正态分布的微分熵表达式为

$$H(x) = \frac{1}{2}(1 + ln(2\pi\sigma^2))$$

由以上的表达式可知，香农熵随着方差而增大。同时，我们也可以看出，与离散型变量的香农熵不同，微分熵可以是负的。

条件熵(conditional entropy)

$$H[y|x] = -\int\int p(x,y)lnp(y|x)dydx$$

$$H[x,y] = H[y|x] + H[x]$$

相对熵(relative entropy)

依然从编码角度来考虑，若一个变量的真实分布为$p(x)$，而我们实际上使用了$q(x)$来对这个变量进行编码，那么由此而使用了的多余的比特数定义为*相对熵*或者*KL距离（Kullback-Leibler divergence）*。

$$KL(p||q) = -\int p(x)ln(q(x))dx - (-\int p(x)ln(p(x))dx) = -\int p(x)ln(\frac{q(x)}{p(x)})dx$$

注意到，虽然名为距离，但是KL距离（相对熵）没有对称性。另外，相对熵是非负的，当且仅当$p(x) = q(x)$时相对熵取零。其证明用到了以下内容：

*凸函数*定义为$f(\lambda a + (1-\lambda)b) <= \lambda f(a) + (1-\lambda)f(b)$。等价地，函数的二阶导数各处均非负。如果仅当$\lambda = 0, 1$时等号成立，那个这个函数称为严格凸函数。凸函数的相反数为凹函数。香农熵为凹函数。

简森不等式（Jensen's inequality）

$f(\sum_i \lambda_i x_i) <= \sum_i \lambda_i f(x_i)$，其中$\lambda_i >= 0, \sum_i \lambda_i = 1$，$f(x)$为凸函数

如果$\lambda_i = p(x_i)$，那么有

- 连续型：$f(\int xp(x)dx) <= \int f(x)p(x)dx$
- 离散型：$f(E[x]) <= E[f(x)]$

于是，可证相对熵的非负性

$$KL(p||q) = -\int p(x)ln(\frac{q(x)}{p(x)})dx >= -ln(\int q(x)dx) = 0$$

其中$-lnx$严格凸函数，因而当且仅当$p = q$时取等号。

相对熵与似然函数的关系

假设未知真实分布为$p(x)$，我们希望使用一个参数模型$q(x|\theta)$结合N个观测数据来确定一个最优的θ来模拟真实分布。一种自然的方法是使用KL距离做为误差函数，以最小化$p(x)$和$q(x|\theta)$的KL距离为标准来确定最优的参数值。

$$KL(p||q) = \sum_i(-lnq(x_i|\theta) + lnp(x_i))$$

将上面的误差函数相对于参数θ求导，可知：*最小化KL距离等价于最大化似然函数*。

互信息(mutual information)

互信息描述了两个变量之间互相包含关于对方的信息量。定义为两个分布 $p(x,y)$ 和 $p(x)p(y)$ 之间的KL距离

$$I(x,y) = KL(p(x,y)||p(x)p(y)) = -\int\int p(x,y)ln(\frac{p(x)p(y)}{p(x,y)})dxdy$$

根据相对熵的非负性可知，互信息是非负的，当仅且当两个变量相互独立时互信息为零。

$$I(x,y) = H(x) - H(x|y) = H(y) - H(y|x)$$

由此可知，互信息可以看作，当已知一个变量的情况下，另一个变量不确定性降低的程度。

[终于搞定了第一章，春节假期任务完成^_^]

Posted by *Chilly_Rain* 2012年1月27日 17:19

Category: Pattern Recognition and Maching Learning Tag: 信息论 Comment: (0)

PRML Notes - 1.5 Decision Theory

概率论为我们提供了一个用于表达不确性的框架，而最终我们需要使用这个框架所提供的内容来做出决策，比如，预测某个文本的真正的类别是什么。这就是决策论所扮演的角色。

推演和决策

推演（inference）指通过给定的有限的数据集，学习得到输入变量x与响应变量t的联合概率分布 $p(x,t)$;

决策（decision）指根据我们对 $p(x,t)$ 的理解，如何在面对新的问题时（x）去做出最优的决定（t）。

在分类问题中，我们需要决策的就是面对新的样本点，应该它哪一个类标签。而这个问题在贝叶斯框架下，很自然地我们对给定一个样本后，其属于各个类的后验概率更感兴趣。

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

如果我们的目标是最小化将样本分错类的可能性，那么直觉上这就变成了计算MAP的问题。

最小化错分比率

假设我们将输入空间X分成不同的*决策区域（decision regions）*，每个决策区域记为 R_k。每个区域记录了算法每个样本的*决策*类别，而区域之间的边界即为*决策边界*或者*决策面*。以二分类问题为例，一个样本被错分的可能性可表示为

$$p(mistake) = p(x \in R_1, C_2) + p(x \in R_2, C_1) = \int_{R_1} p(x, C_2)dx + \int_{R_2} p(x, C_1)dx$$

其中R_k表示决策类别，C_i表示真实类别。很自然地，如果对于一个输入x，$p(x, C_1) > p(x, C_2)$，那么就应该将该样本分到第一个类的决策区域R_1之中，以最小化上面这个误差函数。而通过乘法法则可知

$$p(x, C_k) = p(C_k|x)p(x)$$

因此，对于每个样本，我们都应该将其赋予其MAP所对应的类。对于多分类问题，通过最大化$p(correct)$会更方便一些，但是结论是相同的：计算每个样本的MAP。

最小化期望损失

更实际一些，不同类别判错的代价也是不相同的。例如，将某种疾病阴性的人判为阳性的代价（一些不安，以及进一步检查的费用），就明显小于，将阳性的人判为阴性（可能会错过治疗的最佳时机而死亡）。为了反映这种情况，损失函数（loss function）的概念被提出来。更进一步地，我们引入了损失矩阵（loss matrix），其中每个元素L_{kj}表示当某样本真实类别为k，而被决策为j时带来的损失。

理想情况下，我们希望获得使L_{kj}最小的j，而实际上我们却无从了解真实类别k的值。因此，实际使用的方法涉及了推演步骤中得到的联合概率分布$p(x, C_k)$，使这个信息来计算平均损失，并以最小化平均损失为目的来做出决策:

$$E[L] = \sum_k \sum_j \int_{R_j} L_{kj} p(x, C_k)dx$$

而对于上式中的每个样本x，我们需要做的是最小化

$$\sum_k L_{kj} p(x, C_k) = p(x) \sum_k L_{kj} p(C_k|x)$$

因此，对每个样本决策步骤中，我们需要做的就是最小化上式右侧的第二项（除$p(x)$之外的部分）。

拒绝决策选项

当对于一个样本x，如果其$p(x, C_k)$对于所有的k都有着相似的概率，那么，算法对这个样本的决策将不那么肯定。对于这种情况，最好是让机器放弃决策，而是提示让人工来确定。为此，需要设定一个阈值θ，如果最大的后验概率$p(C_k|x) <= \theta$，那么机器就拒绝做出决策。注意到，当阈值取1时，所有样本将被拒绝，而如果取1/K时，没有样本将被拒绝。

学习过程的种类

至今我们把学习过程划分成了推演和决策两个步骤，实际上，解决问题的方式可以有以下三种

1. 生成模型（generative model）：显式或稳式地（可能是分别计算出似然性和先验）计算出联合概率分布$p(x, C_k)$，然后计算出后验概率分布，再进行决策。

2. 区分模型（discrimitive model）：直接计算后验概率分布，不必完全了解联合分布，然后进行决策。
3. 区分函数（discrimitive function）：直接学习到一个函数，将输入直接通过这个函数映射到一个类标签，它将推演和决策合为一体。

生成模型计算消耗较大，但是通过联合分布可以了解得到一些额外的信息，比如可以通过归一化得到$p(x)$，从而了解一个待测样本点是噪声点的可能性有多大（噪声检测）。区分函数的方法最直接，绕过了后验分布的计算，但是在一些情况下，后验分布还是很有意义的，可以简化一些计算过程。

- 最小化期望损失：如果损失矩阵是经常变动的，那么如果已有后验概率分布，那么可以直接使用它来计算新的损失矩阵下各样本的分类（根据$\sum_k L_{kj}p(C_k|x)$），否则可能就需要将算法重新跑一遍。
- 拒绝决策选项：必须需要后验分布才成奏效。
- 补偿先验：为了模型学习的有效性，我们需要人工构造出各类内训练数据量相等的数据集，在计算出后验分布后，再通过乘除运算将模型中的人工先验替换成实际中的真实先验。
- 合并模型：如果输入变量的两个特征子集之间是独立的，那么可以分别计算每个特征子集的后验概率分布，再使用一些方法合并两个分布。例如，朴素贝叶斯模型。

回归损失函数

面向连续型变量的学习过程称为回归。此时推演的过程仍然是计算得到x与t的联合概率分布，而决策的过程就是得到输入变量x的响应估计值y(x)。与分类问题的决策过程相似，回归问题也需要一个损失函数，而一个常用的损失函数称为平方误差(square loss)。相应的，期望平方误差为

$$E[L] = \int \int (y(x) - t)^2 p(x,t) dx dt$$

我们的目标就是选择一个$y(x)$以最小化$E[L]$。通过计算损失函数$E[L]$相对于$y(x)$的导数，可得

$$y(x) = E_t[t|x] = \int t p(t|x) dt$$

即给定x后的响应t的条件平均值，这个$y(x)$也称为回归函数（regression function）。从另一个角度来看，

$$(y(x) - t)^2 = (y(x) - E[t|x] + E[t|x] - t)^2$$

代入$E[L]$可得如下形式

$$E[L] = \int (y(x) - E[t|x])^2 p(x) dx + \int (E[t|x] - t)^2 p(x,t) dx dt = bias + variance$$

为最小化上面这个损失函数，等式右侧第一项当$y(x) = E_t[t|x]$时取得最小值为零，而第二项则表示了期望意义下输入变量x的响应值t的波动情况（方差）。因为它仅与联合概率分布有关与$y(x)$无关，所以它表示了损失函数中无法约减的部分。

与分类问题相似，回归问题学习的过程可以分为

1. 学习得到联合分布，再归一化得到条件分布，最后得出条件均值$y(x)$；
2. 直接得到条件分布，再计算条件均值$y(x)$；

3. 直接从训练数据中获得$y(x)$。

当然，平方误差并不是唯一可能的误差函数。平方误差的一般化称为明科夫斯基误差（Minkowski loss），其相应的期望误差为

$$E[L_q] = \int \int |y(x) - t|^q p(x,t) dx dt$$

Posted by *Chilly_Rain* 2012年1月25日 22:46

Category: Pattern Recognition and Maching Learning Tag: 决策论 Comment: (1)

PRML Notes - 1.4 The Curse of Dimensionality

前面的内容基本上都是基于单元变量的，而实际的绝大部分情况下，我们需要面对的是多元变量，这时候数据的稀疏性就会展现出来，导致一些看似科学而优雅的算法根本无可适从。这种现象就被称为*维数灾难*。

Examples：多数投票和多项式模型

多数投票是一种简化的KNN分类方法。假设我们有一些已分类的数据点，同时数据是平滑的，我们可以先挑出与待分类点最近的K个点，并根据这K个点的类标签来统计出哪个类在这个*局部*内是最可能出现的，并将此类标签赋予待测分类点。在低维数据集中，这个方法是简单而有效的，然而，当维数增大，这种方法很快就失效了，因为所需要的数据量会随着维数而呈指数级别的增长。

多项式模型的复杂度也会随着维数的增长而变大，导致模型的不可用。例如，3阶多项式模型，对于D维变量的多项式形式为

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

独立参数的个数变为了D^3，更一般地，对于M阶模型，这个数字将变为D^M。虽然这个数字只是随着D而呈幂律增长，但是也足以让这个模型的参数估计由于数据量的不足而变得极度困难，最终致使模型无法使用。

几何直觉

我们考虑D维空间中的球体。首先，半径为r的D维球体的体积为

$$V_D(r) = K_D r^D$$

接下来，考虑半径为r=1的D维单位球体的体积，与半径为$r = 1 - \epsilon$的球体体积之间的一层球壳，与单位球体体积的比例为

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

当维度D趋近于无穷时，这个比例趋近于1，也就是说当维数足够大时，几乎所有的体积都聚集中一个球体表面的薄壳之中。相似地，分析极坐标下的多元正态分布，当维数趋近于无穷时，绝大多数概率都集中在以r为半径的很薄的壳中。

这意味着低维空间下的机器学习算法，有可能在高维空间中变得完全不可用。然而，应用于高维空间的方法也是存在的。首先，数据很可能只存在于高维空间上的一个低维流形(manifold)上，通过流形学习(manifold learning, e.g. Isomap, LLE)算法可以有效地降维。其次，实际数据中往往存在局部平滑性（smothness），因此可以采用局部类插值技术去对未知数据进行预测。

Posted by *Chilly_Rain* 2012年1月25日 18:02

Category: Pattern Recognition and Maching Learning Tag: 维数灾难 Comment: (0)

PRML Notes - 1.3 Model Selection

对于多项式模型中阶数的选择可以看作是一种模型选择的过程，然而，更一般地，我们可能需要从更多*种类*的模型（即不限于多项式模型，例如人工神经网络模型等）中来进行选择，挑出最合适当前数据的模型。

如果数据量是足够大的，那么我们可以取出一部分数据（训练集），对所有的候选模型进行训练，再使用一组验证集(validation set)来评估哪个模型才是最优的。如果模型本身的设计是迭代式的，即需要验证集的反馈来不断地调整模型参数，那么上面的方法就有过拟合的危险。这种情况下，就需要再从数据中独立地分离出一部分，称为测试集(test set)，使用这部分数据来进行最终的模型评估和选择。

交叉验证

然后在实际的多数情况下，数据量是有限的。一种解决方案是使用交叉验证（cross-validation），将数据进行切分成相等的S份，每次迭代均使用其中的S-1份来进行训练，余下的一份来做测试。因此，一共需要进行S次的“训练-测试”，并取这S次的测试结果的均值来作为模型评估的标准。特别地，当S=N时，这种方法被称为leave-one-out技术。

交叉验证方法也是有缺陷的。当S较大时，交叉验证方法的计算消耗是很大的。另外，如果模型中有多个模型的复杂度参数（例如，多个规则化参数），那么交叉验证的计算量将随参数个数而指数级增长。

信息标准

信息标准的提出目的是使用每个模型对数据的拟合程度作为标准，来做模型的比较。然而，使用最大似然性来作为这个拟合程度是很自然的，但是使用其来评估模型已经被证明是有偏的（由于过拟合的原因）。各信息标准都在最大似然性的基础上增加了一个惩罚项（penalty term）克服过拟合问题。已存在方法包括AIC, BIC等，然而这些方法的缺陷在于它们都更偏向于选择*简单的模型*。

Posted by *Chilly_Rain* 2012年1月25日 17:16

Category: Pattern Recognition and Maching Learning Tag: 模型选择简介 Comment: (0)

PRML Notes - 1.2 Probability Theory

模式识别过程中使用的数据集的特点确定了整个学习过程中具有不确定性。一方面，数据集中的样本容量是有限的，无法完整刻画数据中的规律；另一方面，现实世界中的数据集几乎总是有噪声的。

基本概念

加法规则，乘法规则，边缘分布，联合分布，条件概率，贝叶期定理（先验概率，后验概率），条件独立。

对于一个连续型变量，当δx趋近于0时，$p(x)\delta x$的取值称为该变量在此处的概率密度（probability density），$p(x)$称为该变量的概率密度函数。概率密度满足非负性和归一化性质。对于一个离散型变量，$p(x)$称为该变量的概率质量函数（probability mass function）。

当两个连续型变量存在非线性关系$y=g(x)$，那么需要保证$p_x(x)\delta x=p_y(y)\delta y$，从而得到两个概率密度函数的关系

$$p_x(x)\frac{dx}{dy}=p_y(y)$$

基本统计量

期望，条件期望，方差，方差与期望的等式关系，协方差

概率的定义

1. 频率角度：N（N趋近于无穷）次实验中，某件事情发生的次数M与N的比值。
2. 贝叶斯角度：不确定性的度量（或者置信程度的度量）

例如，多项式拟合的示例中，带有噪声的响应变量，其取值可以使用频率角度来解释，而对于学习得到的模型参数的不确定性，则需要使用贝叶斯角度来解释。

贝叶斯概率

假设D为数据集，w为模型参数。通过贝叶斯定理，

$$P(w|D)=\frac{P(w)P(D|w)}{P(D)}$$

可以发现对比先验分布$P(w)$和后验分布$P(w|D)$，我们对模型参数取值的置信度发生了变化，而触发这种变化的正是我们所观察到的数据，更准确地说是基于先验分布而得到的数据的*似然性*$P(D|w)$。在给定参数值的情况下，似然性越大，说明这个取值与观察数据更加符合，我们也会在后验中更加确信（注意到似然性与后验是正比的关系），否则确信度就会降低。

*后验概率 正比于 先验概率*数据似然性*

似然性函数(likelihood function)在频率观点和贝叶斯观点中都极其重要，但是在两种不同框架下的用法是截然不同的。

- 在频率观点下，参数值被认为是一个定点，是根据观测数据结合某个方法得到的。参数估计值的偏差或不确定性是通过取所有可能的数据集计算得到的一系列参数值的方差来评估的。
- 在贝叶斯观点下，参数的取值被认为是不确定的，而数据集是唯一的，因此输出的是一个关于这个参数的分布，并以此来表示参数取值的不确定性。

频率框架下的一种常用的估计方法为极大似然估计法(ML estimator)。目标的误差函数被定义为“似然性的负对数”，并此为目标来估计出能使此误差函数最小的参数值。正如前面所说，此方法针对一个数据集，只有能得一个定点的参数估计值，要评估其波动性，需要使用多个数据集分别进行参数估计，再统计这些估计值的方差等指标。一种可行的方法是使用重采样技术，如bootstrap，来从原始数据集中衍生出多个数据集，再对参数的波动性。

在贝叶斯框架下，算法的输出自然地是一个关于参数的概率分布，从而表达了这个参数的不确定性。贝叶斯框架的一个重要优势在于其可以融入一些先验知识，从而使结果不容易出现过拟合。然而，先验也同时是贝叶斯框架遭人诟病之处，因为它具有主观色彩。有些时候人们对先验的选择仅仅是根据其是否便于数学推导而进行，但是先验选择是否合适对最终的效果影响很大。非信息性先验（Noninformative Prior）可以缓解对先验主观性的依赖，但是却无法保证其是否合适，是否能达到良好的效果。同时，在做模型选择时，不同的先验也使比较无法平等进行，这时就不得不使用频率框架下的交叉验证等模型选择方法。

正态分布

正态分布也称为高斯分布，是一种常见的连续型概率分布。其单元变量的概率密度函数形式为

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

其中 μ 为期望，σ^2为方差。方差的倒数称为*精度*，记作 $\beta=\frac{1}{\sigma^2}$。一个概率分布的最大值称为它的mode，正态分布的mode位于其期望处。

对于N个独立同分布(iid)的样本，其在正态分布下的对数似然性为

$$ln(p(x|\mu,\sigma^2))=\sum_{n=1...N}ln(N(x_n|\mu,\sigma^2))$$

使以上对数似然函数最大的 μ 和 σ^2，称为正态分布期望和方差的*极大似然估计值*。特别地，二者的求解是相互解耦的，即求解 μ_{ML} 的过程是可以独立于σ^2_{ML} 的求解过程的。

$$\mu_{ML}=\frac{1}{N}\sum_{n=1...N}x_n$$

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

一般地，极大似然估计方法是容易过拟合的。特别地，以上对方差的估计值是有偏的(bias):

$$E[\sigma^2_{ML}]=(\frac{N-1}{N})\sigma^2$$

因此，可以相应地对方差的极大似然估计值乘以N/(N-1)的系数，来使其变成无偏估计。同时我们也注意到，若使样本容量N趋近于无穷，那么极大似然估计出的方差值也是会趋近于真实的方差值的。然而样本容量终究是有限的，特别地，当模型复杂度很高时，过拟合所引起的bias问题就会显得犹为严重。

正态分布可应用于多项式拟合问题，一般情况下，可以假设在某个x处的响应值t是符合以下分布的

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

或者说，x处的响应值会受到分布为$N(0,\beta^{-1})$的噪声的影响。

在给定了训练数据集$x_i,t_i,i=1...N$之后，我们可以使用前面的极大似然估计法来计算出分布的w,β值，从而得到每个x处其响应值的概率分布。

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

对以上对数似然性函数求w的导数，可以发现，其等价于最小化Sum-of-squares误差函数所得到的结果。由此可知，*正态噪声分布的假设下，最大化似然函数值等价于最小化Sum-of-squares误差函数值。*

同样地，可以在计算求得w_{ML}后独立地计算β_{ML}（同样是求导的方法）。在二者都确定之后，就得到了一个*预测分布（predicitive distribution）*。

更进一步地，我们不止最大化似然函数，可以为参数w引入一个先验分布，来反映在未观察到任何数据之前，对这个参数取值的偏好。简单起见，可以假设这个也是如下形式的正态分布，

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

其中的α控制了模型参数w的取值，因此被称为超参数(hyperparameter)。根据*后验分布正比于似然性函数乘以先验分布*的关系，可得后验分布的形式为$p(w|x,t,\alpha,\beta)$。接下来可以通过最大化这个后验分布函数来取得最可能w值(maximum posterior, MAP)。而在此求解过程中也会发现，*最大化后验分布函数等价于规则化参数$\lambda=\alpha/\beta$的岭回归*。

以上的两种方式终究仍然是对*参数的点估计*。采用更贝叶斯一点的方法，我们需要得到的是*参数的概率分布*，并通过积分来结合参数所有的可能值来做出决策。

假设参数α,β是已知的，那么预测分布可以表示为

$$p(t|x,\mathbf{x},\mathbf{t}) = \int p(t|x,w)p(w|\mathbf{w},\mathbf{t})dw$$

其中被积函数的第二项即为w的后验分布，这个等式表示为了得到x的预测响应值，应该根据每个w的后验概率来计算每个对应的响应估计值的加权平均。后面的章节会提到，实际上这个后验分布也是正态分布，因此整个预测分布也是正态分布，而其方差为

$$\beta^{-1} + \phi(x)^T S\phi(x)$$

其中第一项反映了由于噪声而引起的响应值t的波动，而后一项则反映了参数w的波动（后验概率分布，非点估计）。

Posted by *Chilly_Rain* 2012年1月20日 18:50

Category: Pattern Recognition and Maching Learning Tag: 概率 模式识别 机器学习 Comment: (0)

PRML Notes - 1.1 Introduction

模式识别的目标

自动从数据中发现潜在规律，以利用这些规律做后续操作，如数据分类等。

模型选择和参数调节

类似的一族规律通常可以以一种模型的形式为表达，选择合适模型的过程称为模型选择（Model Selection）。模型选择的目的只是选择模型的形式，而模型的参数是未定的。

从数据中获得具体规律的过程称为训练或学习，训练的过程就是根据数据来对选定的模型进行参数调节（Parameter Estimation）的过程，此过程中使用的数据为训练数据集（Training Set）。

对于相同数据源的数据来讲，规律应该是一般的（泛化Generalization），因此评估一个学习结果的有效性可以通过使用测试数据集（Testing Set）来进行的。

预处理

对于大多数现实中的数据集来讲，使用其进行学习之前，通常需要进行预处理，以提高学习精度及降低学习的开销。

以图像识别为例，若以像素做为一个特征，往往一幅图像的特征就能达到几万的数量级，而很多特征（如背景色）都是对于图像辨识起不到太大作用的，因此对于图像数据集，预处理过程通常包括维数约减（特征变换，特征选择），仅保留具有区分度的特征。

文本数据分类任务中，对训练文本也有类似的处理方式，只不过此时扮演特征的是单词，而不是像素值。

监督学习和非监督学习

输入向量（input vector）：$\mathrm{x}_1, ..., x_n$，响应向量（target vector）：$\mathrm{t}_1, ..., t_n$

监督学习采用的数据集是包括输入向量和目标向量的，其目标就是发现二者之间的关系，学习的结果表示为函数$\mathrm{y}(\mathrm{x})$，使用函数的输出来近似响应值。如果t_i为离散值，则此类学习任务称为分类（classification），若为连续值则称为回归（regresssion）。

非监督学习使用的数据集只包括输入向量，目的是直接探索数据的内在结构。发现数据中相似的簇的任务称为聚类（clustering），计算数据分布情况的任务称为密度估计（density estimation），将数据映射到三维及以下的任务称为可视化（visulization）。

还有一种学习形式，称为加强学习（reinforcement learning），指在一定的环境下，发现最合适的决策来最大化收益。通常这类学习任务需要在使用（exploit）和探索（explore）之间做出权衡。

Example：多项式回归

给定了数据集（输入向量和响应向量），首先进行模型选择，选定多项式的阶数。高阶的多项式模型是包含低阶多项式模型的（可将高阶项的系数设为零，从而退化成低阶模型），因此高阶模型拥有比低阶模型更强的拟合数据的能力。使用相对于数据来讲过强的模型，会使模型不但捕获数据中的规律，而会拟合进噪声，造成泛化能力不佳，这种情况称为过拟合（overfitting）。相反地，如果使用了相对于数据来讲太弱的模型，就无法捕获数据中的规律，这种情况称为欠拟合（underfitting）。选择合适的模型阶数是学习成功的前提条件，模型选择的方法包括基于经验的bootstrap，cross-validation以及基于信息论的AIC，BIC，MDL等。

在选好模型阶数，确定模型之后，下一步的工作就是参数调节（或称参数估计）。成熟的模型一般都有其相应的参数估计方法，如GMM-EM，RBF-BP，AR-YW等。对于多项式模型可采取较为一般的方法，即定义一个误差函数，通过求导数来计算得到误差最小值时的参数值。

Sum-of-square误差函数：$E = \frac{1}{2} \sum_{i=1...n} (y(x_i) - t_i)^2$

Root-mean-squre误差函数：$E_{RMS} = \sqrt{2E/N}$（单样本误差）

当过拟合发生时，训练得到多项式曲线表现得波动极大，相应地模型参数的模也很大。当数据量足够大时，模型发现过拟合的概率降低，因为模型总是在参数估计中尽可能地去迎合数据，从这个角度讲，数据越多，对模型的约束就越大。

如果希望使用一个相对复杂的模型，而不产生过似合现象，一种可行的方法是在误差函数中加入规则化项（regularization term），以约束模型系数的模。使用2-范数来做规则化项，得到修改后的Sum-of-square误差函数表示为

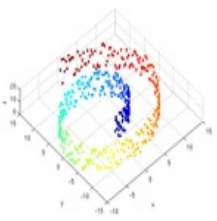
$$E = \frac{1}{2} \sum_{i=1...n} (y(x_i) - t_i)^2 + \frac{\lambda}{2} ||w||^2$$

使用此误差函数进行学习的过程也称为岭回归（ridge-regression）或者权重退化（weight decay）。本质上，规则化项的引入是把一个参数转化为了另一个参数，即将模型阶数M转化成了系数λ，并没有使学习过程变成更有效，

Posted by *Chilly_Rain* 2012年1月20日 18:52

Category: Pattern Recognition and Maching Learning Tag: 简介 模式识别 机器学习 Comment: (0)

yantx



分类

- Bla, Bla, ...
- Coursera_PGM
- Hadoop
- Python源码剖析
- Coursera_NLP
- Pattern Recognition and Maching Learning

最新评论

- cleaning services...
- deep cleaning ser...
- deep cleaning abu...
- cleaning services...
- cleaning companie...

最新留言

- timlp60 : Nude Se...
- evelynkq16 : Hot ...
- estherjd2 : My ne...
- elsiexh1 : New su...
- georginaau60 : En...

链接

- code6
- murongxixi

RSS

功能

- 注册
- 登录
- 忘记密码？
- 文章 RSS
- 评论 RSS
- 留言 RSS

- Styled with scribbish