**2.朴素贝叶斯**

**2.1定义**

对于给定数据集，首先基于特征条件独立假设学习输入输出的**联合概率分布**，然后基于此模型对给定的输入x利用贝叶斯定理求出**后验概率最大**的输出y。

朴素贝叶斯算法是基于**贝叶斯定理**和**特征条件独立假设**的分类方法，通过学习联合概率分布求后验概率最大，实际上是学习到生成数据的机制，是典型的生成模型。

优点：在数据较少的情况下仍然有效，可以处理多类别问题。

缺点：对于输入数据的准备方式比较敏感。

**2.1.1基本方法**

假设输入空间χ⊆Rn为n维向量的集合，输出空间为类标记集合у={c1, c2, ......, cK}。输入为特征向量x∈χ，输出为类标记y∈у。X是定义在输入空间χ上的随机向量，Y是定义在输出空间у上的随机变量。P(X, Y)是X和Y的联合概率分布。训练数据集

T = {(x1, y1), (x2, y2), ......, (xN, yN)}

由P(X, Y)**独立同分布**产生。

（1）朴素贝叶斯方法通过训练数据学习联合概率分布P(X, Y)，具体是学习先验概率分布及条件概率分布。

1）先验概率分布

P(Y = ck)，k = 1，2，……，K

2）条件概率分布

P(X=x | Y=ck) = P (X(1) = x(1), ......, X(n) = x(n) | Y = ck)，k=1，2，……，K

条件概率分布的参数是指数级的，其估计实际是不可行的。假设x(j)可取值有Sj个(1≤j≤n)，Y可取值有K个，则参数个数为个。

所以朴素贝叶斯法对条件概率分布作了条件独立性假设（朴素贝叶斯方法也由此得名，也是“朴素”的原因）。条件独立性假设是

这一假设使朴素贝叶斯法变得简单，但有时也会牺牲一定的分类准确率。

（2）根据以上学习到的先验概率分布和条件概率分布对给定的输入x计算后验概率最大的类作为x的输出。后验概率根据贝叶斯定理计算：

根据条件独立性假设计算式带入上式可得到朴素贝叶斯分类的基本公式：

于是朴素贝叶斯分类器可表示为

因为对所有ck来说上式的分母都是相同的，所以

**2.1.2 后验概率最大化的含义**

后验概率最大化等价于期望风险最小化。假设损失函数为0-1损失函数：

其中f(X)是分类决策函数。这时期望风险函数为

Rexp(f) = E [L(Y, f(X))]

期望是对联合分布P(X, Y)取的，由此取条件期望

为了使期望风险最小化，只需对X=x逐个极小化，由此可得：

这样就根据期望风险最小化准则得到了后验概率最大化准则：

即朴素贝叶斯法采用的原理。

**2.2朴素贝叶斯法的参数估计**

**2.2.1极大似然估计**

可以应用极大似然估计法估计先验概率P(Y = ck)和条件概率P(X(j) = x(j) | Y = ck)。先验概率的极大似然估计是

，k = 1，2，……，K

设第j个特征x(j)可能取值的集合为{aj1, aj2, ......, ajSj}，条件概率P(X(j) = ajl | Y = ck)的极大似然估计是

j = 1，2，……，n；l = 1，2，……，Sj；k = 1，2，……，K

其中xi(j)是第i个样本第j个特征；ajl是第j个特征可能取的第l个值；I为指示函数。

算法流程：

输入：训练数据T = {(x1, y1), (x2, y2), ......, (xN, yN)}，其中xi = {xi(1), xi(2), ......, xi(n)}T，xi(j)是第i个样本的第j个特征，xi(j)∈{aj1, aj2, ......, ajSj}，ajl是第j个特征可能取的第l个值，j = 1，2，……，n，l = 1，2，……，Sj，yi∈{c1, c2, ......, cK}；实例x；

输出：实例x的分类。

（1）计算先验概率P(Y = ck)和条件概率P(X(j) = x(j) | Y = ck)

，k = 1，2，……，K

j = 1，2，……，n；l = 1，2，……，Sj；k = 1，2，……，K

（2）对于给定的实例x = (x(1), x(2), ......, x(n))T，计算

（3）确定实例x的类

**2.2.2贝叶斯估计**

用极大似然估计的缺点是可能会出现所要估计的概率值为0的情况，影响后验概率的计算结果，使分类产生偏差。可以采用贝叶斯估计解决这一问题。

条件概率的贝叶斯估计是

其中λ≥0。等价于在随机变量各个取值的频数上赋予一个正数λ>0。当λ=0时就是极大似然估计，常取λ=1，这时称为拉普拉斯平滑(Laplace smoothing)。

先验概率的贝叶斯估计是

朴素贝叶斯方法的另外一个缺点就是很多很小的数相乘导致数值下溢问题，一般采用对乘积取自然对数的方式避免这个问题。