朴素贝叶斯算法

杨航锋

朴素贝叶斯算法是经典的机器学习算法之一，它是基于贝叶斯定理和条件独立性假设的分类算法，该算法在训练过程中学习生成数据的机制，所以属于生成模型 。贝叶斯学派的思想可以概括为**先验概率+数据=后验概率**，如果在实际问题中需要得到的后验概率，可以通过先验概率和数据一起综合得到。先验概率就是对于数据所在领域的历史经验，但是这个经验常常难以量化或者模型化，于是贝叶斯学派大胆的假设先验分布， 然后基于此分布对于给定的输入 X ，利用贝叶斯定理求出后验概率最大的输出 y 。 更多文章见GitHub地址

朴素贝叶斯模型的推导

假设输入空间 $\mathcal{X} \in \mathbb{R}^n$ 为 n 维向量的集合，输出空间为类别集合 $\mathcal{Y} = \{c_1, c_2, \cdots, c_t\}$，输入为特征向量 $x \in \mathcal{X}$，输出为所属类别 $y \in \mathcal{Y}$。 X 是定义在输入空间 $\mathcal{X}$ 上的随机向量， Y 是定义在输出空间 $\mathcal{Y}$ 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 $T = \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \cdots, (x^{(m)}, y^{(m)})\}$ 由 $P(X, Y)$ 独立同分布产生。

朴素贝叶斯分类时对给定的输入 $x^{(i)}$ ，通过学习到的模型计算后验概率分布 $P(Y = c_k | X = x^{(i)})$ ，将后验概率最大的类作为 x 的所属类别输出。后验概率根据贝叶斯定理计算方式如下：

$$P(Y = c_k | X = x^{(i)}) = \frac{P(X = x^{(i)} | Y = c_k) P(Y = c_k)}{\sum\limits_{k=1}^{t} P(X = x^{(i)} | Y = c_k) P(Y = c_k)} \qquad (*)$$

由于朴素贝叶斯算法对条件概率分布作了条件独立性假设，故

$$\begin{aligned} P(X = x^{(i)} | Y = c_k) &= P(X_1^{(i)} = x_1^{(i)}, \cdots, X_n^{(i)} = x_n^{(i)} | Y = c_k) \\ &= \prod_{j=1}^{n} P(X_j^{(i)} = x_j^{(i)} | Y = c_k) \end{aligned} \qquad (**)$$

将 $(**)$ 式带入 $(*)$ 式中得到朴素贝叶斯算法的基本形式：

$$P(Y = c_k | X = x^{(i)}) = \frac{\prod\limits_{j=1}^{n} P(X_j^{(i)} = x_j^{(i)} | Y = c_k)}{\sum\limits_{k=1}^{t} P(Y = c_k) \prod\limits_{j=1}^{n} P(X_j^{(i)} = x_j^{(i)} | Y = c_k)} P(Y = c_k)$$

因此朴素贝叶斯算法的优化模型为

$$\begin{aligned} \arg\max_{c_k} \ P(Y = c_k | X = x^{(i)}) &= \arg\max_{c_k} \ \frac{\prod\limits_{j=1}^{n} P(X_j^{(i)} = x_j^{(i)} | Y = c_k)}{\sum\limits_{k=1}^{t} P(Y = c_k) \prod\limits_{j=1}^{n} P(X_j^{(i)} = x_j^{(i)} | Y = c_k)} P(Y = c_k) \\ &= \arg\max_{c_k} \ P(Y = c_k) \prod_{j=1}^{n} P(X_j^{(i)} = x_j^{(i)} | Y = c_k) \end{aligned}$$

因为对于每一个类别 c_k ，分母 $\sum\limits_{k=1}^{t} P(Y = c_k) \prod\limits_{j=1}^{n} P(X_j^{(i)} = x_j^{(i)} | Y = c_k)$ 的值都是相同的。

朴素贝叶斯模型的求解

求解朴素贝叶斯模型相当于求解几个概率值，对于样本数据集可以求出先验概率 $p(Y=c_k)$

$$p(Y=c_k)=\frac{\sum\limits_{i=1}^{m}I(y^{(i)}=c_k)}{t}\quad k=1,2,\cdots,t$$

其中 $I(x)$ 为示性函数，当 x 为真时函数值为 1 否则为 0 。条件概率

$$P(X_j^{(i)}=x_j^{(i)}|Y=c_k)=\frac{\sum\limits_{i=1}^{m}I(x_j^{(i)}=a_{jl},y^{(i)}=c_k)}{\sum\limits_{i=1}^{m}I(y^{(i)}=c_k)}$$
$$j=1,2,\cdots,n$$
$$l=1,2,\cdots,S_j$$

其中$x_j^{(i)}$代表第 i 个样本的第 j 个特征 $x_j^{(i)}\in\{a_{j1},a_{j2},\cdots,a_{jS_j}\}$ ， a_{jl} 表示第 j 个特征取得第 l 个值。

拉普拉斯平滑

由于极大似然估计可能会出现需要估计的概率值为零的情况，而这会影响后验概率的计算，使分类产生偏差，于是可采用贝叶斯估计

$$P_\lambda(Y=c_k)=\frac{\sum\limits_{i=1}^{m}I(y^{(i)}=c_k)+\lambda}{t+t\lambda}$$
$$P_\lambda(X_j^{(i)}=x_j^{(i)}|Y=c_k)=\frac{\sum\limits_{i=1}^{m}I(x_j^{(i)}=a_{jl},y^{(i)}=c_k)+\lambda}{\sum\limits_{i=1}^{m}I(y^{(i)}=c_k)+S_j\lambda}$$

其中 $\lambda\geqslant 0$ ，当 $\lambda=1$ 时称为拉普拉斯平滑。

总结

朴素贝叶斯算法通过贝叶斯定理和条件独立性假设，从而把难求的概率问题转化为容易求解的概率问题，直观一点表示就是

$$P(\text{类别}|\text{特征})=\frac{P(\text{特征}|\text{类别})P(\text{类别})}{P(\text{特征})}$$

该算法的优点主要是模型简单实现上比较容易且有稳定的分类效果；缺点主要是现实世界中的数据一般不满足条件独立性假设，而且先验概率很多时候取决于假设的先验分布，假设的概率分布可以有很多种，因此在某些时候会由于假设的先验分布的原因导致预测效果不佳 。