分类模型与算法

张翼鹏

November 6, 2019

目录

0 什么是分类

分类方法是一种对离散型随机变量建模或预测的监督学习方法。其中分类学习的目的是从给定的人工标注的分类训练样本数据集中学习出一个分类函数或者分类模型，也常常称作分类器（classifier）。当新的数据到来时，可以根据这个函数进行预测，将新数据项映射到给定类别中的某一个类中。

1 kNN（k近邻）

思想：在一个含未知样本的空间，根据离这个未知样本最邻近的k个样本的数据类型来确定该样本的数据类型。

步骤：1、确定k值，确定距离的衡量方式。

2、对于未知样本，找到离它距离最小的k个样本。

3、找到这k个样本中数量最多（或权重最大，权重为距离平方的倒数）的类型，将未知样本归为这个类型。

优点：易于实现，无需估计参数，无需训练，支持增量学习，能对超多边形的复杂决策空间建模。

缺点：计算量较大，分析速度慢。

2 朴素贝叶斯分类器

思想：对于给出的待分类项，利用贝叶斯定理求解在此项出现的条件下各个类别出现的概率，哪个最大，就认为此待分类项属于哪个类别。

步骤：1、$x=\{x_1,x_2,\cdots,x_m\}$为一个待分类项，其中$x_k(1\le i\le m)$为它的特征。有类别集合$C=\{y_1,y_2,\cdots,y_n\}$。

2、求出$P(y_j|x)(1\le j\le n)$。

3、若$P(y_k|x)=\max\limits_{1\le j\le n}P(y_j|x)$，则$x\in y_k$。

对于步骤2中的条件概率，计算步骤如下：

（1）找到一个已知分类的训练集。

（2）求出每种类别出现的概率$P(y_j)(1\le j\le n)$以及在每种类别下各个特征的条件概率估计$P(x_i|y_j)(1\le i\le m,1\le j\le n)$。

（3）假设各个特征之间独立，根据贝叶斯公式：

$$P(y_j|x) = \frac{P(x|y_j)P(y_j)}{P(x)} = \frac{P(y_j)\prod_{i=1}^{m} P(x_i|y_j)}{P(x)}$$

求出条件概率。

优点：结合了先验概率和后验概率，既避免了只使用先验概率的主观偏见，又避免了单独使用样本信息的过拟合现象。算法逻辑简单，有着坚实的数学依据，误判率很低。算法较为稳定，对于不同类型的数据集不会呈现出太大的差异性，在数据集较大的情况下表现出较高的准确率。

缺点：该算法假设各特征之间相互独立，但该条件在实际应用中很难满足，因为特征之间往往存在着关联。当该假设不成立时，会导致分类效果大大降低。所以需要采用合适的方法进行特征选择，这样朴素贝叶斯分类器才能达到更高的分类效率。

3 决策树

思想：决策树是一种树形结构，其中每个内部节点表示一个特征上的测试，每个分支代表一个测试输出，每个叶节点代表一种类别。对于待分类项，从根节点开始判断，沿一定路径最终走到某个叶节点，我们把此项就归为这个类别。

决策树的构造：

[$ID3$算法]

首先介绍熵的概念：

$$Ent(D) = -\sum_{k=1}^{c} p_k \log_2 p_k$$

$Ent(D)$是信息熵，衡量信息的不确定度。其中c为样本集合D中的类型数量，p_k为D中第k类样本所占比例。$Ent(D)$的值越小，就代表该样本集D的纯度越高。

下面给出信息增益的概念：

$$Ent(D,a) = Ent(D) - \sum_{v=1}^{V} \frac{|D_v|}{|D|} Ent(D_v)$$

假设属性a有V个可能取值，那么用a来对样本集进行划分，就会产生V个分支节点，D_v是第v个分支所包含的样本。上式可计算出用属性a对样本集D进行划分所获得的信息增益。信息增益越大，用属性a对样本进行划分的纯度越高。

$ID3$算法就是以信息增益最大为原则创建决策树。

优点：方法简单。

缺点：对噪声敏感；算法倾向于选择取值比较多的属性，有些属性可能对分类任务没有太大作用，但是他们仍然可能会被选为最优属性。

[$C4.5$算法]

1、不以信息增益为依据，改用信息增益率。信息增益率=信息增益/属性熵。

2、使用悲观剪枝。悲观剪枝是后剪枝技术中的一种，通过递归估算每个内部节点的分类错误率，比较剪枝前后这个节点的分类错误率来决定是否对其进行剪枝。这种剪枝方法不再需要一个单独的测试数据集。

3、对连续的属性进行离散化的处理，$C4.5$选择具有最高信息增益的划分所对应的阈值。

4、针对数据集不完整的情况，$C4.5$也可以进行处理。

优点：对噪声不敏感；能处理连续型数据和不完整数据。

缺点：在构造树的过程中，需要对数据集进行多次的顺序扫描和排序，导致算法的低效。

决策树的剪枝：

- 预剪枝：在决策树构造时就进行剪枝。方法是，在构造的过程中对节点进行评估，如果对某个节点进行划分，在验证集中不能带来准确性的提升，那么对这个节点进行划分就没有意义，这时就会把当前节点作为叶节点，不对其进行划分。

- 后剪枝：在生成决策树之后再进行剪枝。通常会从决策树的叶节点开始，逐层向上对每个节点进行评估。如果剪掉这个节点子树，与保留该节点子树在分类准确性上差别不大，或者剪掉该节点子树，能在验证集中带来准确性的提升，那么就可以把该节点子树进行剪枝。方法是：用这个节点子树的叶子节点来替代该节点，类标记为这个节点子树中最频繁的那个类。

4 感知机

思想：感知机是二分类的线性模型，其输入是样本的特征向量，输出的是样本的类别，分别是-1和1。其学习目标是得到一个能够将训练数据集正实例点和负实例点完全正确分开的分离超平面。

步骤：1、假设超平面是$h = w \cdot x + b$，其中$w = (w_0, w_1, \cdots, w_m), x = (x_0, x_1, \cdots, x_m)$。对于样本点$x' = (x'_1, x'_2, \cdots, x'_n)$，其到超平面的距离为：

$$d = \frac{w \cdot x' + b}{||w||}$$

2、从输入空间到输出空间的模型如下：

$$y = \begin{cases} -1 & w \cdot x + b < 0 \\ 1 & w \cdot x + b \geq 0 \end{cases}$$

3、代价函数的优化目标，就是期望所有被误分类的样本到超平面的距离之和最小。代价函数定义如下（忽略$||w||$）：

$$L(w, b) = -\sum_{x_i \in M} y_i(w \cdot x_i + b)$$

4、由于只有误分类M集合里的样本才能参与损失函数的优化,所以我们不能用最普通的批量梯度下降,只能采用随机梯度下降。即首先给定w和b的初始值，依次选取单个数据点(x_i, y_i)，判断其是否被误分类（$y_i(w \cdot x_i + b) \leq 0$），若是，则按照梯度下降法更新$w$和$b$：

$$w' = w + \eta y_i x_i \qquad b' = b + \eta y_i$$

优点：作为支持向量机和神经网络的基础，算法简单且易于实现。

缺点：该算法分类的依据是符号的正负，没有一个量化标准，所以不适用于多元分类；对于非线性可分的数据（比如异或），最后结果会在一定范围内震荡，无法获得超平面。

5 logistic回归

思想：利用sigmoid函数，将一般的线性回归模型转换成一种概率函数，进而判断未知样本属于某一类的概率是多少。

步骤：1、目标函数为：

$$h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}$$

目标函数计算的是一个样本在第0类和第1类中属于第1类的概率。

2、使用交叉熵代价函数：

$$J(\theta) = -\frac{1}{n}\sum_{i=1}^{n}[(y^{(i)})\ln(h_\theta(x^{(i)})) + (1 - y^{(i)})\ln(1 - h_\theta(x^{(i)}))]$$

3、使用梯度下降法更新参数θ：

$$\theta'_j = \theta_j - \frac{1}{n}\sum_{i=1}^{n}(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)}$$

对于多元分类问题，我们有以下两种方式对上述二元分类模型进行改进。

（1）对于有k个类别的分类问题，首先将第一类看成正样本，其他所有类看成负样本，计算目标函数值计作p_1；然后将第二类看作正样本，其他所有类看作负样本，计算p_2；以此类推，最后若$p_j = \max\limits_{1\le i\le k} p_i$，则将未知样本归为第$j$类。

（2）利用softmax函数。目标函数为：

$$h_w(x) = \frac{1}{\sum_{i=1}^{k} e^{w_i x+b_i}} \begin{bmatrix} w_1 \cdot x + b_1 \\ w_2 \cdot x + b_2 \\ \vdots \\ w_k \cdot x + b_k \end{bmatrix}$$

其中k为类别的个数，w_i和b_i为第i个类别对应的权重向量和偏置标量。

我们可以将目标函数求出的第$j(1 \le j \le k)$个分量看成是样本x属于第j类的概率，即：

$$\frac{e^{w_j \cdot x+b_j}}{\sum_{i=1}^{k} e^{w_i \cdot x+b_i}} = P(y_j = 1|x)$$

其中y_j为样本x的期望输出的第j个分量，若该样本属于第j类，则该值为1，否则为0。

使用对数似然代价函数：

$$J(w,b) = -\frac{1}{n}\sum_{i=i}^{n}\sum_{j=1}^{k} y_j^{(i)} \ln(\frac{e^{w_j \cdot x^{(i)}+b_j}}{\sum_{l=1}^{k} e^{w_l \cdot x^{(i)}+b_l}})$$

其中$y_j^{(i)}$为第i个样本的期望输出的第j个分量，若该样本属于第j类，则该值为1，否则为0。

利用梯度下降法更新w：

$$w'_r = w_r + \frac{1}{n}\sum_{i=1}^{n} (x^{(i)} - P(y_r^{(i)} = 1|x^{(i)}) * x^{(i)})$$

优点：自变量可以是离散型也可以是连续型；借助概率的概念，易于理解；计算代价不高。

缺点：当特征空间很大时，性能不是很好；容易欠拟合，准确度不太高。

6 支持向量机

思想：支持向量机是一种二分类模型，目的是寻找一个超平面来对样本进行分割，分割的原则是间隔最大化，最终转化为一个凸二次规划问题来求解。

6.1 线性可分——硬间隔分类器

对于完全线性可分的样本：

步骤：1、超平面表达式为$w \cdot x + b = 0$，我们的目标就是求出w向量和b标量。

2、正样本$y = 1$，负样本$y = -1$。根据空间中点到平面的距离公式，我们一般令所有样本离超平面的距离至少为$\frac{1}{||w||}$。那么对于任意样本x，都满足$y(w \cdot x + b) \geq 1$。

3、距离超平面最近的样本点被称为支持向量，满足$y(w \cdot x + b) = 1$；支持向量中一个正样本和一个负样本到超平面的距离之和称为间隔，记为$\gamma = \frac{2}{||w||}$。

4、我们要最大化γ，相当于最小化γ的倒数，故优化问题为：

$$\begin{cases} \min \dfrac{1}{2}||w||^2 \\ s.t.\ y_i(w \cdot x_i + b) \geq 1 \end{cases}$$

可以采用拉格朗日乘子法对其对偶问题进行求解。该算法的一个重要特征是，当训练完成后，大部分样本都不需要保留，最终模型只与支持向量有关。

6.2 近似线性可分——软间隔分类器

对于近似线性可分（仅有少数点线性不可分）的样本，我们可以放宽标准，引入松弛变量的概念，同时给目标函数加入惩罚项。

$\xi_i \geq 0$称为第i个样本点的松弛变量，原优化问题变为：

$$\begin{cases} \min \dfrac{1}{2}||w||^2 + C\sum\limits_{i}^{n} \xi_i \\ s.t.\ y_i(w \cdot x_i + b) \geq 1 - \xi_i \end{cases}$$

其中C为惩罚因子。

6.3 非线性可分——核函数

对于非线性可分数据，我们可以通过升维，将原始数据转化为线性可分的数据。实际操作即在求解优化问题的对偶问题时，将其中的内积计算替换为核函数表达式。

表 1: 常用核函数表达式

名称	表达式	参数
线性核	$\kappa(x_i, x_j) = x_i^T x_j$	
多项式核	$\kappa(x_i, x_j) = (x_i^T x_j)^d$	$d \geq 1$为多项式的次数
高斯核	$\kappa(x_i, x_j) = \exp(-\frac{\|\|x_i - x_j\|\|^2}{2\sigma^2})$	$\sigma > 0$为高斯核的带宽
拉普拉斯核	$\kappa(x_i, x_j) = \exp(-\frac{\|\|x_i - x_j\|\|}{\sigma})$	$\sigma > 0$
sigmoid核	$\kappa(x_i, x_j) = \tanh(\beta x_i^T x_j + \theta)$	tanh为双曲正切函数$\beta > 0, \theta < 0$

优点：引入最大间隔，分类精确度高；样本量较小时，也能准确的分类，并且泛化能力好；引入核函数，能轻松解决非线性问题。

缺点：样本量非常大时，核函数中内积的计算量过大；核函数的选择通常没有明确的指导，有时难以选择一个合适的核函数，且需要调试的参数也较多。

7 神经网络

思想：人工神经网络是一种模仿动物神经网络行为特征，进行分布式并行信息处理的数学模型。这种网络依靠系统的复杂程度，通过调整内部大量节点之间相互连接的关系，从而达到处理信息的目的。

步骤：1、构造神经网络结构：输入层-隐藏层-输出层。

2、初始化权重和偏置，对于每个输入x，使用前向传播计算每个神经元的激活值a。

$$a^l = \sigma(w^l a^{l-1} + b^l)$$

3、通过反向传播计算代价函数对权重和偏置的偏导。

$$\delta^L = \nabla_a C \odot \sigma'(z^L)$$

$$\delta^l = ((w^{l+1})^T \delta^{l+1}) \odot \sigma'(z^l)$$

$$\frac{\partial C}{\partial b_j^l} = \delta_j^l$$

$$\frac{\partial C}{\partial w_{jk}^l} = a_k^{l-1} \delta_j^l$$

4、利用小批量梯度下降法更新权重和偏置，直到达到理想效果。

$$w'_k = w_k - \frac{\eta}{m} \sum_j \frac{\partial C_{X_j}}{\partial w_k}$$

$$b'_l = b_l - \frac{\eta}{m} \sum_j \frac{\partial C_{X_j}}{\partial b_l}$$

优点：具有很强的非线性拟合能力；上限很高，如果数据量充足且超参数选择得当，将达到非常惊艳的效果。

缺点：需要大量数据；算法较为复杂，需要确定的超参数较多；无法解释其推理过程和推理依据。

8 分类模型评价

8.1 错误率与精度

错误率与精度是分类任务中最常用的性能指标，适合于二分类及多分类任务。错误率E指错误分类的样本数占总样本数的比例，精度为$1-E$。

8.2 查准率与查全率

对于二分类问题，若正类和负类的样本数差距过大，则错误率与精度不足以衡量模型的性能，我们还关心查准率和查全率。

查准率P：在所有被标为“正例”的样本中，真正为“正例”的样本比例。

查全率R：在所有真正为“正例”的样本，被标记为“正例”的样本比例。

8.3 ROC曲线

对于二分类问题，我们较关注正例的情形，所以设置了两个相应的指标TPR和FPR。

TPR：将实际的1正确地预测为1的概率。

FPR：将实际的0错误地预测为1的概率。

TPR与FPR相互影响，我们希望TPR尽量大，而FPR尽量小。影响TPR与FPR的重要因素就是分类的阈值。

ROC曲线横轴为FPR，纵轴为TPR，范围均为$[0,1]$。ROC曲线越凸向左上角的顶点，分类器效果越好；或定义AUC为ROC曲线下的面积，AUC 越接近1，分类器效果越好。