机器学习各算法对比

参考自：https://zhuanlan.zhihu.com/p/46831267

算法	决策树	KNN	朴素贝叶斯	线性回归	逻辑回归
训练数据	分类数据	分类数据	数值数据	数值数据	数值数据
目标数据	分类数据	分类数据	分类数据	数值数据	分类数据

1.决策树

决策树的一大优势就是易于解释。它可以毫无压力地处理特征间的交互关系并且是非参数化的，因此你不必担心异常值或者数据是否线性可分（举个例子，决策树能轻松处理好类别A在某个特征维度x的末端，类别B在中间，然后类别A又出现在特征维度x前端的情况）。它的缺点之一就是不支持在线学习，于是在新样本到来后，决策树需要全部重建。另一个缺点就是容易出现过拟合，但这也就是诸如随机森林RF（或提升树boosted tree）之类的集成方法的切入点。另外，随机森林经常是很多分类问题的赢家（通常比支持向量机好上那么一丁点），它训练快速并且可调，同时你无须担心要像支持向量机那样调一大堆参数，所以在以前都一直很受欢迎。

决策树中很重要的一点就是选择一个属性进行分枝，因此要注意一下信息增益的计算公式，并深入理解它。

信息熵的计算公式如下:

$$H = -\sum_{i=1}^{n} p(x_i) log_2 p(x_i)$$

其中的n代表有n个分类类别（比如假设是二类问题，那么n=2）。分别计算这2类样本在总样本中出现的概率 p_1 和 p_2 ，这样就可以计算出未选中属性分枝前的信息熵。

现在选中一个属性 x_i 用来进行分枝，此时分枝规则是：如果 $x_i = v$ 的话，将样本分到树的一个分支；如果不相等则进入另一个分支。很显然，分支中的样本很有可能包括2个类别，分别计算这2个分支的熵 H_1 和 H_2 ,计算出分枝后的总信息熵 $H' = p1H1 + p2H2$ ，则此时的信息增益 $\Delta H = H - H'$ 。以信息增益为原则，把所有的属性都测试一边，选择一个使增益最大的属性作为本次分枝属性。

优点

- 决策树易于理解和解释，可以可视化分析，容易提取出规则；
- 可以同时处理标称型和数值型数据；
- 比较适合处理有缺失属性的样本；
- 能够处理不相关的特征；
- 测试数据集时，运行速度比较快；
- 在相对短的时间内能够对大型数据源做出可行且效果良好的结果。

缺点

- 容易发生过拟合（随机森林可以很大程度上减少过拟合）；
- 容易忽略数据集中属性的相互关联；
- 对于那些各类别样本数量不一致的数据，在决策树中，进行属性划分时，不同的判定准则会带来不同的属性选择倾向；信息增益准则对可取数目较多的属性有所偏好（典型代表ID3算法），而增益率准则（CART）则对可取数目较少的属性有所偏好，但CART进行属性划分时候不再简单地直接利用增益率尽心划分，而是采用一种启发式规则）（只要是使用了信息增益，都有这个缺点，如RF）。
- ID3算法计算信息增益时结果偏向数值比较多的特征。

改进措施

- 对决策树进行剪枝。可以采用交叉验证法和加入正则化的方法。
- 使用基于决策树的combination算法，如bagging算法，randomforest算法，可以解决过拟合的问题；

应用领域

企业管理实践，企业投资决策，由于决策树很好的分析能力，在决策过程应用较多。

2.KNN算法

KNN即最近邻算法，其主要过程为：

```
1. 计算训练样本和测试样本中每个样本点的距离（常见的距离度量有欧式距离，马氏距离等）；
2. 对上面所有的距离值进行排序(升序)；
3. 选前k个最小距离的样本；
4. 根据这k个样本的标签进行投票，得到最后的分类类别；
```

如何选择一个最佳的K值，这取决于数据。一般情况下，在分类时较大的K值能够减小噪声的影响，但会使类别之间的界限变得模糊。一个较好的K值可通过各种启发式技术来获取，比如，交叉验证。另外噪声和非相关性特征向量的存在会使K近邻算法的准确性减小。近邻算法具有较强的一致性结果，随着数据趋于无限，算法保证错误率不会超过贝叶斯算法错误率的两倍。对于一些好的K值，K近邻保证错误率不会超过贝叶斯理论误差率。

优点

- 理论成熟，思想简单，既可以用来做分类也可以用来做回归；
- 可用于非线性分类；
- 训练时间复杂度为O(n)；
- 对数据没有假设，准确度高，对outlier不敏感；
- KNN是一种在线技术，新数据可以直接加入数据集而不必进行重新训练；
- KNN理论简单，容易实现；

缺点

- 样本不平衡问题（即有些类别的样本数量很多，而其它样本的数量很少）效果差；
- 需要大量内存；
- 对于样本容量大的数据集计算量比较大（体现在距离计算上）；
- 样本不平衡时，预测偏差比较大。如：某一类的样本比较少，而其它类样本比较多；
- KNN每一次分类都会重新进行一次全局运算；

- k值大小的选择没有理论选择最优，往往是结合K-折交叉验证得到最优k值选择；

应用领域

文本分类、模式识别、聚类分析，多分类领域

3.朴素贝叶斯

朴素贝叶斯属于生成式模型（关于生成模型和判别式模型，主要还是在于是否需要求联合分布），比较简单，你只需做一堆计数即可。如果注有条件独立性假设（一个比较严格的条件），朴素贝叶斯分类器的收敛速度将快于判别模型，比如逻辑回归，所以你只需要较少的训练数据即可。即使NB条件独立假设不成立，NB分类器在实践中仍然表现的很出色。它的主要缺点是它不能学习特征间的相互作用，用mRMR中R来讲，就是特征冗余。引用一个比较经典的例子，比如，虽然你喜欢Brad Pitt和Tom Cruise的电影，但是它不能学习出你不喜欢他们在一起演的电影。

优点：

- 朴素贝叶斯模型发源于古典数学理论，有着坚实的数学基础，以及稳定的分类效率。
- 对大数量训练和查询时具有较高的速度。即使使用超大规模的训练集，针对每个项目通常也只会有相对较少的特征数，并且对项目的训练和分类也仅仅是特征概率的数学运算而已；
- 对小规模的数据表现很好，能个处理多分类任务，适合增量式训练（即可以实时的对新增的样本进行训练）；
- 对缺失数据不太敏感，算法也比较简单，常用于文本分类；
- 朴素贝叶斯对结果解释容易理解；

缺点：

- 需要计算先验概率；
- 分类决策存在错误率；
- 对输入数据的表达形式很敏感；
- 由于**使用了样本属性独立性的假设，所以如果样本属性有关联时其效果不好**；

应用领域

- 欺诈检测中使用较多
- 一封电子邮件是否是垃圾邮件
- 一篇文章应该分到科技、政治，还是体育类
- 一段文字表达的是积极的情绪还是消极的情绪？
- 人脸识别

4.线性回归

线性回归是用于回归的，它不像Logistic回归那样用于分类，其基本思想是用**梯度下降法**对最小二乘法形式的误差函数进行优化，当然也可以用normal equation直接求得参数的解，结果为：

$$\hat{w} = (X^T X)^{-1} X^T y$$

而在LWLR（局部加权线性回归）中，参数的计算表达式为: $\hat{w} = (X^T W X)^{-1} X^T W y$

由此可见LWLR与LR不同，LWLR是一个非参数模型，因为每次进行回归计算都要遍历训练样本至少一次。

> **定义**：线性回归（Linear Regression）是利用称为线性回归方程的最小平方函数对一个或多个自变量和因变量之间关系进行建模的一种回归分析。
>
> 求解最优解的方法有**最小二乘法和梯度下降法。**

优点：

实现简单，计算简单；

缺点：

不能拟合非线性数据.

5.逻辑回归

逻辑回归属于判别式模型，同时伴有很多模型正则化的方法（L0， L1，L2，etc），而且你不必像在用朴素贝叶斯那样担心你的特征是否相关。与决策树、SVM相比，你还会得到一个不错的概率解释，你甚至可以轻松地利用新数据来更新模型（使用在线梯度下降算法-online gradient descent）。如果你需要一个概率架构（比如，简单地调节分类阈值，指明不确定性，或者是要获得置信区间），或者你希望以后将更多的训练数据快速整合到模型中去，那么使用它吧。

Sigmoid函数：表达式如下:

$$f(x) = \frac{1}{1 + e^{-x}}$$

优点：

- 实现简单，广泛的应用于工业问题上；
- 分类时计算量非常小，速度很快，存储资源低；
- 便利的观测样本概率分数；
- 对逻辑回归而言，多重共线性并不是问题，它可以结合L2正则化来解决该问题；
- 计算代价不高，易于理解和实现；

缺点：

- 当特征空间很大时，逻辑回归的性能不是很好；
- 容易**欠拟合**，一般准确度不太高
- 不能很好地处理大量多类特征或变量；
- 只能处理两分类问题（在此基础上衍生出来的softmax可以用于多分类），且必须**线性可分**；
- 对于非线性特征，需要进行转换；

应用领域：

- 用于二分类领域，可以得出概率值，适用于根据分类概率排名的领域，如搜索排名等。
- Logistic回归的扩展softmax可以应用于多分类领域，如手写字识别等。
- 信用评估
- 测量市场营销的成功度
- 预测某个产品的收益
- 特定的某天是否会发生地震