|  |  |  |
| --- | --- | --- |
| **算法** | **优点** | **缺点** |
| 朴素贝叶斯 | * 对小规模的数据表现很好，适合多分类任务，适合增量式训练。 | * 对输入数据的表达形式很敏感（离散、连续，值极大极小之类的） |
| Logistic Regression | * 实现简单 * 分类计算速度快，计算量小 | * 容易underfitting * 只能处理两类分类问题   Overfitting:怎么办：   * 减少feature个数 * 正则化 |
| KNN | * 思想简单，理论成熟，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)； * 准确度高，对数据没有假设，对outlier不敏感 | * 计算量大 * 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） * 需要大量的内存 |
| SVM | * 使用核函数可以向高维空间进行映射 * 使用核函数可以解决非线性的分类 * 分类思想很简单，就是将样本与决策面的间隔最大化 * 分类效果较好 | * 对大规模数据训练比较困难 * 无法直接支持多分类，但是可以使用间接的方法来做 |
| 决策树 | * 计算量简单，可解释性强，比较适合处理有缺失属性值的样本，能够处理不相关的特征 | * 单颗决策树分类能力弱，并且对连续值变量难以处理； * 容易过拟合（后续出现了随机森林，减小了过拟合现象） |
| 随机森林 | * 能够处理大量特征的分类，并且还不用做特征选择 * 在训练完成之后能给出哪些feature的比较重要 * 训练速度很快 * 很容易并行 * 实现相对来说较为简单 | 在噪声较大的分类或者回归问题上回过拟合。 |
| GBDT | * 预测精度高 * 适合低维数据 * 能处理非线性数据 * 可以灵活处理各种类型的数据，包括连续值和离散值。 * 在相对少的调参时间情况下，预测的准备率也可以比较高。这个是相对SVM来说的。 * 使用一些健壮的损失函数，对异常值的鲁棒性非常强。比如 Huber损失函数和Quantile损失函数。 | * 由于弱学习器之间存在依赖关系，难以并行训练数据。不过可以通过自采样的SGBT来达到部分并行。 * 如果数据维度较高时会加大算法的计算复杂度 |
| Shrinkage | * 精度高 * 能处理非线性数据 * 能处理多特征类型 * 适合低维稠密数据 | * 并行麻烦（因为上下两颗树有联系） * 多分类的时候 复杂度很大 |
| 神经网络 | * 网络实质上实现了一个从输入到输出的映射功能，而数学理论已证明它具有实现任何复杂非线性映射的功能。这使得它特别适合于求解内部机制复杂的问题； * 网络能通过学习带正确答案的实例集自动提取“合理的”求解规则，即具有自学习能力； * 网络具有一定的推广、概括能力。 | * BP算法的学习速度很慢，其原因主要有： * 由于BP算法本质上为梯度下降法，而它所要优化的目标函数又非常复杂，因此，必然会出现“锯齿形现象”，这使得BP算法低效； * 存在麻痹现象，由于优化的目标函数很复杂，它必然会在神经元输出接近0或1的情况下，出现一些平坦区，在这些区域内，权值误差改变很小，使训练过程几乎停顿； * 为了使网络执行BP算法，不能用传统的一维搜索法求每次迭代的步长，而必须把步长的更新规则预先赋予网络，这种方法将引起算法低效。 * 网络训练失败的可能性较大，其原因有： * 从数学角度看，BP算法为一种局部搜索的优化方法，但它要解决的问题为求解复杂非线性函数的全局极值，因此，算法很有可能陷入局部极值，使训练失败； * 网络的逼近、推广能力同学习样本的典型性密切相关，而从问题中选取典型样本实例组成训练集是一个很困难的问题。 * 难以解决应用问题的实例规模和网络规模间的矛盾。这涉及到网络容量的可能性与可行性的关系问题，即学习复杂性问题； * 网络结构的选择尚无一种统一而完整的理论指导，一般只能由经验选定。为此，有人称神经网络的结构选择为一种艺术。而网络的结构直接影响网络的逼近能力及推广性质。因此，应用中如何选择合适的网络结构是一个重要的问题； * 新加入的样本要影响已学习成功的网络，而且刻画每个输入样本的特征的数目也必须相同； * 网络的预测能力（也称泛化能力、推广能力）与训练能力（也称逼近能力、学习能力）的矛盾。一般情况下，训练能力差时，预测能力也差，并且一定程度上，随训练能力地提高，预测能力也提高。但这种趋势有一个极限，当达到此极限时，随训练能力的提高，预测能力反而下降，即出现所谓“过拟合”现象。此时，网络学习了过多的样本细节，而不能反映样本内含的规律 |
| Kmeans | * 原理比较简单，实现也是很容易，收敛速度快。 * 聚类效果较优。 * 算法的可解释度比较强。 * 主要需要调参的参数仅仅是簇数k。 | * K值不好选 * 不是凸数据集比较难选择 * Unbalanced数据会产生难收敛的情况 * 因为是迭代的方法，只能达到局部最优 * 对噪音和异常点比较敏感 * 初始值选择很重要 |
| EM | * 简单稳定 | * 迭代速度慢，次数多 * 容易陷入局部最优 |
| CNN | * 由于maxpooling层的存在，因此CNN在小范围内具有几何不变性，但是对于大范围就不具备这种几何不变性。 * 特征分层分类 * 收敛速度快 | * 卷及网络层次太深 * 使用gradient descent方法的收敛速度与选取的w和b的初始值有关。 * 使用gradient descent方法很容易使得训练的效果收敛到局部最小值而不是全局最小值。 |
| PCA | * 以方差衡量信息的无监督学习，不受样本标签限制。 * 各主成分之间正交，可消除原始数据成分间的相互影响 * 可减少指标选择的工作量 * 用少数指标代替多数指标，利用PCA降维是最常用的算法 * 计算方法简单，易于在计算机上实现。 | * 主成分解释其含义往往具有一定的模糊性，不如原始样本完整 * 贡献率小的主成分往往可能含有对样本差异的重要信息 * 特征值矩阵的正交向量空间是否唯一有待讨论 * 无监督学习 |

看一下如何应用

https://blog.csdn.net/iemyxie/article/details/40736773