|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model | 理论 | 步骤 | 代价函数 | 优劣 | 备注 |
| Logistic Regression | 假设数据服从伯努利分布,通过极大化似然函数的方法，运用梯度下降来求解参数 |  |  | 1.实现简单  2.分类时计算量非常小，速度很快，存储资源低 | [logistic regression](https://github.com/le773/PythonDebug/blob/master/supervised%20learning/06logistic/logistic.md) |
| 1.容易欠拟合，一般准确度不太高  2.只能处理二分类问题，且必须线性可分 | 使用softmax |
| SVM | 1.拉格朗日乘子法  2.对偶问题  3.二次规划  4.SMO | 1.优化目标函数  2.转换成拉格朗日形式  3.使用对偶理论转换目标函数  4.对w,b求导    = = |  | 1.可用于线性、非线性分类，也可回归  2.低泛化误差  3.容易解释  4.计算复杂度低 |  |
| 1.对参数和核函数的选择比较敏感  2.原始的SVM只擅长处理二分类问题 |  |
| KNN | 投票表决 | 1.假设有一个带有标签的样本数据集（训练样本集），其中包含每条数据与所属分类的对应关系。  2.输入没有标签的新数据后，将新数据的每个特征与样本集中数据对应的特征进行比较。  a.计算新数据与样本数据集中每条数据的距离。  b.对求得的所有距离进行从小到大排序  c.取前k（k一般小于等于 20）个样本数据对应的分类标签。  3.求 k 个数据中出现次数最多的分类标签作为新数据的分类。 |  | 1.理论简单，可分类可回归  2.可用于非线性分类  3.训练时间复杂度为O(n)  4.准确度高，对数据没有假设，对outlier不敏感 |  |
| 1.计算量大  2.样本不平衡问题  3.需要大量内存 | KD-Tree |
| KD-Tree | [KD-Tree.md](https://github.com/le773/PythonDebug/blob/master/supervised%20learning/02knn/KD-Tree.md) |  |  |  |  |
|  |
| Decision-Tree | 1.信息增益  2.信息增益率  3.Gini系数 |  |  | 1.计算简单，可解释性强，比较适合处理有缺失属性的样本，能够处理不相关的特征 | 随机森林 |
| 1.容易过拟合 |
| 朴素贝叶斯 |  |  |  | 1.对小规模的数据表现良好，适合多分类任务，适合增量式训练 |  |
| 1.对输入数据的表达形式很敏感 |
| Boosting |  | 先从初始训练集训练出一个基学习器，再根据基学习器的表现对训练样本分布进行调整，使得先前基学习器做错的训练样本在后续受到更多关注，然后基于调整后的样本分布训练下一个基学习器；如此重复进行，直到基学习器达到事先指定的值T，最终将这T个基学习器进行加权结合。 |  | 1.低泛化误差；  2.容易实现，分类准确率较高，没有太多的参数可调整 |  |
| 1.对outlier比较敏感 |
| Linear Regression | 用梯度下降法对最小二乘法形式的误差函数进行优化 |  | 普通线性回归 | 1.实现简单，计算简单 |  |
| 局部加权线性回归 | 1.不能拟合非线性数据 |
| K-means | 基于划分 | 1. 创建 k 个点作为起始质心（通常是随机选择）  2. 当任意一个点的簇分配结果发生改变时  2.1 对数据集中的每个数据点  2.1.1 对每个质心  2.1.2 计算质心与数据点之间的距离  2.1.3 将数据点分配到距其最近的簇   * 1. 对每一个簇, 计算簇中所有点的均值并将均值作为质心 |  | 1.算法简单、快速  2.对处理大数据集，该算法是相对可伸缩的和高效率的  3.当簇是密集、球状、团状且簇与簇之间区别明显时，聚类效果好 | [k-means.md](https://github.com/le773/PythonDebug/blob/master/unsupervised%20learning/01cluster/%E5%8E%9F%E5%9E%8B%E8%81%9A%E7%B1%BB%E7%AE%97%E6%B3%95k-means.md)  k-means++:  初始的聚类中心之间的相互距离要尽可能的远 |
| 1.对初值敏感  2.不适合发现非凸面形状的簇，或者大小差别很大的簇  3.对噪声、孤立点数据敏感，少量的该类数据能够对平均值产生极大影响。 |
| Agnes | 基于层次聚类  自底向上聚合策略 | 1.先对仅含一个样本的初始聚类簇和相应的距离矩阵进行初始化；  2.然后不断合并距离最近的聚类簇，并对合并得到的聚类簇的距离矩阵进行更新  3.上述过程1，2不断重复，直到达到预设的聚类簇数。 |  |  |  |
|  |  |
| Dbsacn | 基于密度聚类 |  |  | 1.将足够高密度的区域划分成簇，并能在具有噪声的空间数据库中发现任意形状的簇  2.在大规模数据库上更好的效率 |  |
|  |  |
| Wave  Cluster、  STING | 基于网格的方法 |  |  |  |  |
|  |  |
| EM、SOM、COBWEB | 基于模型的聚类 |  |  |  |  |
|  |  |
| GBDT | 一种迭代的决策树算法，该算法由多棵决策树组成，所有树的输出结果累加起来就是最终答案。 | 其核心就在于，每一棵树是从之前所有树的残差中来学习的。 |  |  |  |
|  |  |
| EM | 似然估计 | E步：选取一组参数，求出在该参数下隐含变量的条件概率值；  M步：结合E步求出的隐含变量条件概率，求出似然函数下界函数（本质上是某个期望函数）的最大值。  重复上面2步直至收敛。 |  |  |  |
|  |  |
| 异常检测 |  | 将特征的每一维看成是相互独立的高斯分布，根据异常样本拟合每个特征的()，然后在新的样本计算P(x)，如果小于某阈值，则认为Anomaly |  |  | [anomaly detection](https://github.com/le773/PythonDebug/blob/master/AndrewNg/ML/week9.md) |
|  |  |
| 关联 |  |  |  |  |  |
|  |  |
| Svd |  |  |  |  |  |
|  |  |
|  |  |  |  |  |  |
|  |  |

多元GBDT分类算法

样本k负梯度误差

=

=