# Chapter 3: Algorithms

- 没有最好的分类器，只有最合适的分类器
- 逻辑回归LR：Logistic Regression
  - 名字有回归是因为历史性原因，但是是针对分类问题
  - 其因变量是离散的，线性回归中的因变量是连续的
  - vs 线性回归
    - 当自变量x与超参数θ确定的情况下，逻辑回归可以看做广义线性模型(Generalized Linear Models)在因变量y服从二元分布时的一个特殊情况，而使用最小二乘法求解线性回归时，认为因变量y服从正态分布。
    - 两者都使用了极大似然来对训练样本进行建模，都可以使用梯度下降法训练
  - 多标签的分类问题
    - 一个样本只对应于一个标签
      - 多项逻辑回归： Softmax Regression
        - 假设每个样本属于不同标签的概率服从几何分布
        - 实际上是二分类逻辑回归在多标签分类下的一种扩展
    - 样本可能属于多个标签
      - 可以训练K个二分类的逻辑回归分类器
      - 每个分类器i用来预测样本属于i与不属于i的概率
- 支持向量机SVM： Support Vector Machine
  - 对于任意线性可分的两组点，他们在SVM分类的超平面上的投影都是不可分的
  - SVM的分类结果与仅使用支持向量的分类结果一致
  - 存在一组参数使得SVM分类器训练误差为零，但是该参数不一定是满足SVM条件的一个解
  - 实际训练不加入松弛变量的SVM模型，优化目标只有训练误差，则其训练误差为0的分类器一定存在
  - 实际训练加入松弛变量的SVM模型，其训练误差不一定能达到0，因为优化目标变了，并不仅仅是训练误差
- 决策树：Decision Tree
  - 具有简单直观，解释性强的特点
  - 一种自上而下，对样本数据进行树形分类的过程，由结点与有向边组成，结点分为内部结点与叶节点，每个内部结点表示一个特征或属性，叶节点表示类别
  - 将决策时引用集成学习的思想可以得到随机森林、梯度提升决策树等模型
  - 决策树的生成
    - 特征选择
    - 树的构造
    - 树的剪枝
  - 启发函数
    - 最大信息增益：ID3
      - 数据集D的经验熵：H(D)
      - 某个特征A对于数据集D的经验条件熵：H(D|A)
      - 信息增益：g(D,A) = H(D) - H(D|A)
        - 倾向与特征取值较多的特征，处理离散型变量
      - 信息增益反映的是给定条件以后不确定性减少的程度，特征取值越多就意味着确定性更高，也就是条件熵越小(负数)，信息增益越大。
      - 存在缺陷，如DNA取值都不一样，数量多，但是泛化能力太差
      - 选择最大ID3对应的特征作为当前的结点
      - 对样本特征缺失值较敏感
    - 最大信息增益比：C4.5
      - 处理连续型变量
      - 数据集D关于A的取值熵H_A(D)
      - 特征A对于数据集D的信息增益比定义为：g_r(D,A) = g(D,A)/H_A(D)
      - 一定程度上对取值较多的特征进行处罚，避免ID3出现过拟合的特性，提升决策树的泛化能力
      - 选择最大C4.5值对应的特征作为当前的结点
    - 大基尼(Gini)指数: 分类回归树 CART Classification and Regression Tree
      - 处理连续型变量
      - 可以用于分类与回归
      - $$Gini(D) = 1 - \sum_{k=1}^{n}\left(\frac{|C_k|}{|D|}\right)^2$$
        Gini描述的是数据的纯度，与信息熵含义类似
      - $$Gini(D|A) = \sum_{k=1}^{n}\left(\frac{|D_k|}{|D|}\right)Gini(D_k)$$
        特征A的Gini指数
      - 利用全部数据发现所有可能的树结构进行对比，在每一次迭代中选择基尼指数最小的特征及其对应的切分点进行分类
      - 是一颗二叉树，采用二元切割法，每一步将数据按特征A的取值切成两份，分别进入左右子树
      - 每个结点只产生两个分支，且每个特征可以重复使用
  - 剪枝
    - 完全生长的决策树容易过拟合
    - 预剪枝：Pre-Pruning
      - 在生成决策树的过程中提前停止决策树的生长
      - 何时停止成长，通过经验判断，不同问题不同
        - 当树达到一定深度的时候
        - 当到达当前节点的样本数量小于某个阈值的时候
        - 计算每次分裂对测试集的准确的提升，当小于某个阈值的时候
      - 思想直接、算法简单、效率高等特点，适合解决大规模问题
      - 存在一定局限性，有欠拟合的风险。当前划分准确率下降，但是之后可能上升
    - 后剪枝：Post-Pruning
      - 在已生成的过拟合决策树上进行剪枝，得到简化版的剪枝决策树
      - 相比于预剪枝，通常可以得到泛化能力更强的决策树，但时间开销会更大
      - 常见方法
        - 错误率降低剪枝REP： Reduced Error Pruning
        - 悲观剪枝PEP：Pessimistic Error Pruning
        - 代价复杂度剪枝CCP：Cost Complexity Pruning
        - 最小误差剪枝MEP：Minimum Error Pruning
        - CVP: Critical Value Pruning
        - OOP: Optimal Pruning
    - 实际应用中需要根据不同的数据类型、规模、决定使用何种决策树以及对应的剪枝策略，灵活变通，找到最优选择
  - 奥卡姆剃刀定律： Occam's Razor, Ockham'azor
    - 如无必要，勿增实体，即简单有效原理
    - 决策树剪枝
    - ID3决策树算法
    - 神经网络Dropout

**只能用于分类任务，每个结点产生多个分支，且每个特征在层级之间不能复用，通过剪枝来权衡树的准确性与泛化能力**

*改进*

**可以对缺失值进行不同处理**

**结点中包含多个类别数据时，多数投票的原则得出最后的类别**

**2018.12.15 Saturday. By KuKuXia@github.com**

**ML/DL/RL深度交流讨论群: 622545311，加群时请备注: 姓名-方向-公司or高校**