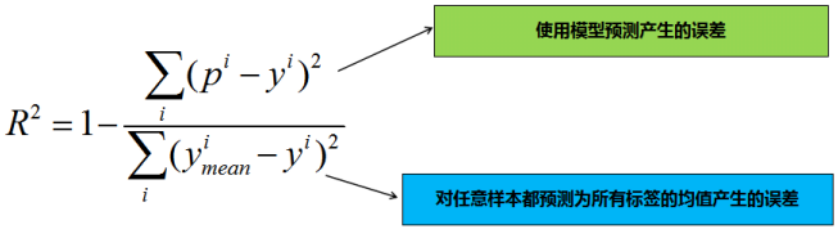
机器学习西瓜书题目总结

2025年5月21日　　14:15

1. 自助法和交叉验证均适合样本较少使用，样本更少选自助
2. 线性回归残差均值为0
3. R2计算：



4. 广义线性模型：
   a. 传统线性回归：存在解析解
   b. 逻辑回归
   c. 感知机回归
   d. 线性判别分析：类间距离尽可能大，类内方差尽可能小
   e. 多分类：OvO和OvR
5. 分类模型：
   a. KNN
   b. 决策树：
      i. ID3算法根据信息增益来构建。信息增益越大，代表当前特征越优

         1) 所以信息增益如果套上机器学习的话就是，如果把特征 A 对训练集 D 的信息增益记为 g(D, A) 的话，那么 g(D, A) 的计算公式就是：

$$g(D, A) = H(D) - H(D, A)$$

      ii. C4.5算法根据信息增益率来构建

         由于在使用信息增益这一指标进行划分时，更喜欢可取值数量较多的特征。为了减少这种**偏好**可能带来的不利影响，`Ross Quinlan`使用了**信息增益率**这一指标来选择最优划分属性。

         1) 信息增益率的数学定义为如下，其中D表示数据集，a表示数据集中的某一列，$Gain(D, a)$表示D中a的信息增益，V表示a这一列中取值的集合，v表示V中的某种取值，$|D|$表示D中样本的数量，$|D^v|$表示D中a这一列中值等于v的数量。

$$Gain_ratio(D, a) = \frac{Gain(D, a)}{-\sum_{v=1}^{V} \frac{|D^v|}{|D|} log_2 \frac{|D^v|}{|D|}}$$

      iii. CART算法根据基尼系数来构建，CART算法使用基尼系数来代替信息增益率，基尼系数代表了模型的不纯度，基尼系数越小，则不纯度越低，特征越好。

         1) 基尼系数的数学定义为如下，其中D表示数据集，p_k表示 D 中第 k 个类别在 D 中所占比例。

$$Gini(D) = 1 - sum_{k=1}^{|y|} p_k^2$$

2）

上面是基于数据集 D 的基尼系数的计算方法，那么基于数据集 D 与特征 a 的基尼系数怎样计算呢？其实和信息增益率的套路差不多。计算公式如下：

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

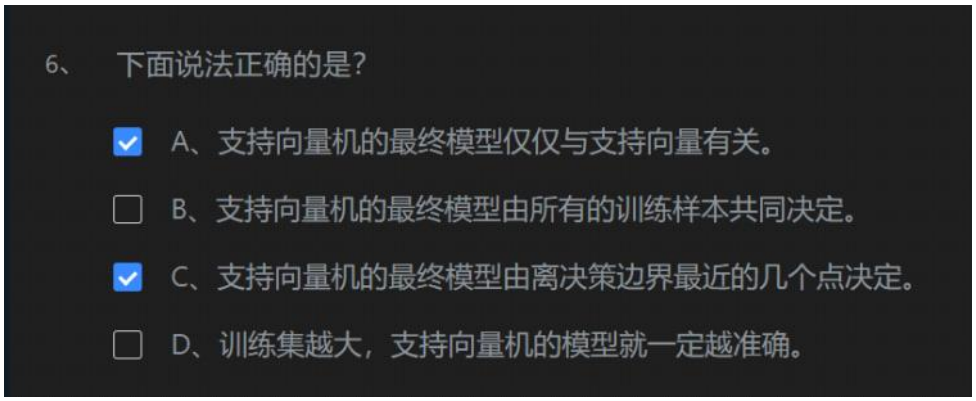
iv. 剪枝算法：预剪枝（如果验证集性能不增加则不分支）、后剪枝（先构建完整子树，再自底向上检查各节点，判断是否需要将其变为叶子结点）

c. 朴素贝叶斯

d. 神经网络

e. 支持向量机：SVM最终获得的直线叫做决策边界，其由支持向量计算得到。而支持向量则由离决策边界最近的点决定。

i.



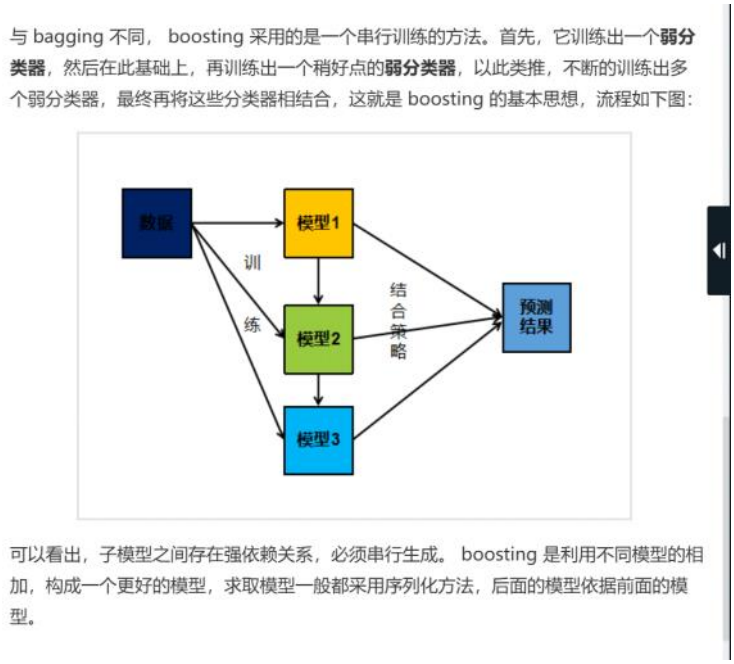
ii. 核函数是为了将非线性可分的数据，映射到高维空间使其线性可分

6. 集成学习模型

a. Bagging:投票策略

b. boosting：

i.



c. Adaboost:采用boosting策略，将多个弱分类器组合成强分类器。对于训练：每一轮训练会提高那些被前一轮弱分类器错误分类样本的权值，而降低那些被正确分类样本的权值。对于组合：采用加权多数表决，准确率高的分类器权重大，准确率低的分类器权重小。

d. Baggingboost:采样的时候回对所有样本采取有放回采样，并行训练n个模型，投票选举出最终结果。
e. RF:不仅对样本采样，对特征也随机采样k个，k为log2(n),n为样本特征数。

7. 聚类模型：
   a. Kmeans
   b. DBSCAN算法
      i. Minpts：构成核心点的邻域内最小样本数目，eps：定义邻域半径
      ii. 直接密度可达（有方向），密度可达（有方向），密度相连（无方向，对称）

1）

1. 定义回顾

- **密度直达（Directly Density-Reachable）**：
  点 q 是点 p 的**密度直达**，如果：
  - q 在 p 的 **ε-邻域**（即 $\text{dist}(p,q) \leq \epsilon$），
  - p 是**核心点**（即其 ϵ-邻域内至少有 **MinPts** 个点）。
- **密度可达（Density-Reachable）**：
  点 p 到点 q 是**密度可达**的，如果存在一系列点 $p_1, p_2, \ldots, p_n$（其中 $p_1 = p, p_n = q$），使得每个 p_{i+1} 是 p_i 的**密度直达**。
  - **特点**：密度可达是**单向的**（即 p 可达 q，但 q 不一定可达 p）。
- **密度相连（Density-Connected）**：
  点 p 和点 q 是**密度相连**的，如果存在某个点 o，使得 p 和 q 都**密度可达**于 o。
  - **特点**：密度相连是**对称的**（即 p 和 q 互相连接）。

2. 关键问题：密度可达是否一定密度相连？

答案：不一定。

- **密度可达**是**单向**的，而**密度相连**是**双向**的。
- **只有当两个点互相密度可达时，它们才是密度相连的。**

      iii. 算法流程：从随机一个点开始，先判断其是否是核心点，如果是则寻找其所有密度可达的点（慢慢扩散，扩散到点标记为簇x），直到没有密度可达点。之后再找到下一个没有被标记的点，重复上述过程。
   c. AGENS
      i. AGNES算法是一种自底向上聚合的层次聚类算法，它先会将数据集中的每个样本看作一个初始簇，然后在算法运行的每一步中找出**距离**最近的两个簇进行合并，直至达到预设的簇的数量。
      ii. 距离度量三种方式：簇之间最大距离、最小距离、平均距离
      iii. 使用该方法时，常需要使用minmax等标准化方式对特征进行标准化，以防止不同特征数值上的影响
   d. EM算法
      i. 极大似然估计：根据观测数据估计参数，叫做似然估计。极大似然估计，就是看参数去哪个值时，似然函数最大。
      ii. 算法相关概念：参数、观测数据、隐变量
      iii. 算法流程
         1）先给参数θ这是一个初始值
         2）根据给定观测数据和θ，计算隐变量z的概率分布的期望
         3）根据计算出的z的概率分布，修正参数θ
         4）重复（2）和（3）
   e. 混合高斯聚类
      i. 首先明白混合高斯，实际上就是n个高斯分布的加权，其中权重之和为1

ii. 混合高斯聚类，先是有k个高斯簇，然后判断每个样本i在哪个高斯簇当中。判断时需要使用概率计算，涉及贝叶斯公式

iii. 为了得到高斯簇和相应权重（$\alpha_i, \mu_i, \Sigma_i$），需要将其当做参数，将样本属于每个簇的概率当做隐变量，然后使用EM算法求解。

f. 聚类模型性能度量

i. 外部指标：需要与参考模型比较，通常包括JC系数、FM指数、Rand指数，都是值越大聚类效果越好

1）

JC系数

JC系数根据上面所提到的a，b，c来计算，并且值域为$[0,1]$，值越大说明聚类性能越好，公式如下：

$$JC = \frac{a}{a+b+c}$$

因此刚刚的例子中，$JC = \frac{2}{2+1+4} = \frac{2}{7}$

FM指数

FM指数根据上面所提到的a，b，c来计算，并且值域为$[0,1]$，值越大说明聚类性能越好，公式如下：

$$FMI = \sqrt{\frac{a}{a+b} * \frac{a}{a+c}}$$

因此刚刚的例子中，$FMI = \sqrt{\frac{2}{2+1} * \frac{2}{2+4}} = \sqrt{\frac{4}{18}}$

Rand指数

Rand指数根据上面所提到的a和d来计算，并且值域为$[0,1]$，值越大说明聚类性能越好，假设m为样本数量，公式如下：

$$RandI = \frac{2(a+d)}{m(m-1)}$$

因此刚刚的例子中，$RandI = \frac{2*(2+8)}{6*(6-1)} = \frac{2}{3}$。

ii. 内部指标：无需外部参考模型，常包括DB指数和Dunn指数，越小越好

1）

DB指数

DB 指数又称 DBI，计算公式如下：

$$DBI = \frac{1}{k}\sum_{i=1}^{k} max(\frac{avg(C_i) + avg(C_j)}{d_c(\mu_i, \mu_j)}), i \neq j$$

公式中的表达式其实很好理解，其中k代表聚类有多少个簇，μ_i代表第i个簇的中心点，$avg(C_i)$代表C_i第i个簇中所有数据与第i个簇的中心点的平均距离。$d_c(\mu_i, \mu_j)$代表第i个簇的中心点与第j个簇的中心点的距离。

2）

Dunn指数

Dunn 指数又称 DI，计算公式如下：

$$DI = min_{1 \leq i \leq k}\{min_{i \neq j}(\frac{d_min(C_i, C_j)}{max_{1 \leq l \leq k} diam(C_l)})\}$$

公式中的表达式其实很好理解，其中k代表聚类有多少个簇，$d_{min}(C_i, C_j)$代表第i个簇中的样本与第j个簇中的样本之间的最短距离，$diam(C_l)$代表第l个簇中相距最远的样本之间的距离。

8. 降维模型
   a. PCA：

降维的方法有很多，而最为常用的就是**PCA**(主成分分析)。 PCA 是将数据从原来的坐标系转换到新的坐标系，新的坐标系的选择是由数据本身决定的。第一个新坐标轴选择的是原始数据中**方差最大的方向**，第二个新坐标轴的选择和第一个坐标轴**正交**且**方差最大的方向。**然后该过程一直重复，重复次数为原始数据中的特征数量。**最后会发现大部分方差都包含在最前面几个新坐标轴中，因此可以忽略剩下的坐标轴，从而达到降维的目的。**

i.

PCA的算法流程

PCA 在降维时，需要指定将维度降至多少维，假设降至 k 维，则 PCA 的算法流程如下：

1. demean；
2. 计算数据的协方差矩阵；
3. 计算协方差矩阵的特征值与特征向量；
4. 按照特征值，将特征向量从大到小进行排序；
5. 选取前 k 个特征向量作为转换矩阵；
6. demean 后的数据与转换矩阵做矩阵乘法获得降维后的数据。

   b. MDS算法
      i. 核心思想：将一个数据集的距离矩阵还原成一个D维坐标来

表示数据

ii. 流程：

1) 通过三个式子求解出降维后的内积矩阵B
2) 对矩阵B进行特征分解
3) 取d个最大特征所构成的对角矩阵，以及对应的特征向量矩阵
4) 计算结果矩阵Z：其中V为向量矩阵，Λ为特征值构成的对角矩阵。Z及为降维之后每个样本的特征。

a) $$Z = V\Lambda^{\frac{1}{2}}$$

c. Isomap算法

i. 适用范围：数据分布是流行的时候，用他。

ii. 算法核心：Isomap 算法将样本与在其邻域内样本之间距离设为欧式距离，与其它样本距离设为无穷大(这里不同算法处理方式不同，有的会设置为最短路算法求出的最短距离），如此获得距离矩阵。后续推导同 MDS 算法。

d. LLE算法（局部线性嵌入）：

i. 核心思想：每个数据点，可以由其邻域数据线性表示，LLE算法希望降维之后这个线性表示的权重系数不发生太大变化。

ii. 算法流程

- 对每个样本确定其邻域。
- 求得矩阵C及其逆。
- 求得权重的值，对不属于邻域内的权重令其等于0。
- 求得M。
- 对M进行特征分解。
- 求得M第2小到第d+1小的特征值对应的特征向量。
- 求得Z。