机器学习

2022年4月23日 21:35

集成学习

2022年4月30日 9:37

1. Bagging和Boosting的区别？（机器学习-P90）
    - 数据采样方式不同：
        - Bagging：对数据集采用有放回采样，每个个体学习器都采用不同的独立数据集训练
        - Boosting：根据上一轮个体学习器的学习效果，动态的调整数据集中样本的权重，利用调整后的数据集训练下一轮的学习器
    - 个体学习器不同：
        - Bagging：个体学习器为强模型（低偏差高方差）；Bagging旨在降低方差，防止过拟合
        - Boosting：个体学习器为弱模型（高偏差低方差）；Boosting旨在降低偏差，提高泛化能力
    - 集成方式不同：
        - Bagging：个体学习器之间是弱依赖关系，采用并行式集成学习方法，集成时回归模型采用平均法，分类模型采用投票法
        - Boosting：个体学习器之间是强依赖关系，采用串行式的集成学习方法，集成时采用加权法
2. XGBoost和GBDT的区别？（机器学习-P95）
    - 基学习器的选择：
        - GBDT：采用CART回归树作为基学习器
        - XGB：既可以采用CART决策时，也可以采用其他线性分类器
    - 优化方法：
        - GDBT：优化时只利用了损失函数的一阶导数
        - XGB：对损失函数进行了泰勒展开，同时利用了一阶导和二阶导信息
    - 防止过拟合（XGB):
        - XGB在损失函数中添加了正则项（包含树的叶子结点的个数，以及每个叶子结点权重分数的平方和），从而防止过拟合
        - XGB定义了缩减系数，每完成一次迭代，会给树的叶子节点乘上缩减系数，削弱每棵树的影响，让后续学习空间更大
        - XGB采用随机森林的思想，在生成树时，对特征字段随机采样，而不是采用所有特征，防止过拟合
    - 缺失值处理：
        - 对于特征值有缺失的样本，XGB可以自动学习出它的分裂方向
    - 并行处理：
        - XGB支持特征粒度的并行计算：生成决策树时，需要计算每个特征的信息增益，XGB在数据预处理阶段对数据进行了排序，并保存为块结构，计算每个特征的信息增益时， 可以利用块结构实现并行计算
3. GBDT原理以及常用的调参参数？（机器学习-P93）
    - 原理：GBDT中所有弱分类器的结果相加等于预测值，因此下一个弱分类器是为了拟合上一个分类器的预测值与真实值的残差
    - 参数：基学习器的迭代次数，学习率，树的最大叶子结点个数，树的最大深度，树的叶子结点包含的最少样本数
4. Stacking和Blending的区别？
    - Blending：利用不同的没有交集的数据集训练不同的个体学习器，最终输出值是所有个体学习器的加权平均
    - Stacking：将数据集分为测试集和训练集，在同一个训练集上训练不同的个体学习器，再在测试数据集上进行测试，将个体学习器的预测值作为训练样本，将正确标签作为输出，再次训练一个高层的模型。
5. AdaBoost和GBDT的区别？（机器学习-P92）
    - AdaBoost
        - 模型的不足：通过在每一次迭代中调整错分样本的权重来定位
        - 最终的决策结果是所有基学习器的加权和
        - 每个基学习器的权重与他们的误差率有关
    - GBDT
        - 模型的不足：通过计算梯度来定位
        - GDBT的每一次迭代都是为了减小前一次模型的预测值与真实值之间的残差
        - 所有基学习器的结果相加等于预测值。
6. AdaBoost与随机森林的区别？（机器学习-P96）
    - 数据集不同：
        - AdaBoost：在每次迭代过程中都会改变数据集中样本的权值，使得错分类样本的权重较高，利用更新后的数据集训练下一个基学习器
        - RF：从整体数据集中有放回的随机抽取一部分数据用来训练每棵树
    - 决策方式不同：
        - AdaBoost：最终决策由所有基学习器加权求和决定，每个基学习器的权重与它的错误率有关
        - RF：随机森林使用简单投票法得到最终决策，即少数服从多数的原则

7. RF与GBDT的区别？（机器学习-P102）
    - 相同点：
        - 基学习器都是树
        - 最终结果由多棵树一起决定
    - 不同点：
        - 数据采样：
            - RF：训练样本采用有放回的随机采样，训练每棵树的数据集都不同，随机选取特征建立决策树（行采样，列采样）（不是每棵树随机选取一些特征，而是在构造树的时候，每个节点选取不同的子特征）
            - GBDT：每一次迭代都采用带权重的训练集
        - 基分类器：
            - RF：可以是分类树 or 回归树
            - GBDT：只能是CART回归树
        - 集成方式：
            - RF：Bagging-树粒度并行，降低方差
            - GBDT：Boosting-树粒度串行，降低偏差
        - 结合策略：
            - RF：分类问题采用简单投票法，回归问题采用简单平均法
            - GBDT：所有树的结果累加
        - 异常值：
            - RF：对异常值不敏感（取决于数据集和训练方式）
            - GBDT：对异常值很敏感（拟合残差）（GDBT当前模型的残差会延续给下一棵树）
8. RF，GBDT，XGBoost的区别

模型	RF	传统GBDT	XGBoost
基分类器	分类树、回归树	回归树，但也可以解决分类问题（设置阈值）	回归树、线性分类器（LR）、线性回归
节点分裂的方式	ID3用信息增益，C4.5用信息增益率，CART分类用基尼指数，CART回归用选择最优切分特征和值对(j,s)	选择最优切分特征和值对(j,s)	优化推导，详见附注
cost函数	-	只用到一阶导数信息（梯度）	代价函数加入正则项（树的叶子节点个数、每个叶子节点上输出的score的L2模的平方和），且同时用到了一阶g和二阶导数h
行采样与列采样	自采样（样本随机）和属性随机（每个节点都会随机选择一些特征），详见附注	-	支持行采样、列采样
集成方法	bagging	boosting	boosting
并行化处理	树并行生成	树串行生成	树串行生成，特征粒度上的并行，各个特征的增益计算可以开多线程进行

9. 树的剪枝
    - 前剪枝：通过提前停止树的构造实现对树的剪枝
        - 剪枝原则：树的深度达到要求，叶子结点达到要求的纯度，叶子节点包含样本数少于指定样本数
    - 后剪枝：首先构造完整的决策树，允许决策树过度拟合训练数据，之后根据最小化损失函数的原则进行剪枝
        - CART树剪枝：
            - 剪枝：从树的叶子结点开始不断剪枝直到树的根部，形成一个子树序列
                - 原则：衡量减掉当前节点后Loss的减少程度，剪掉使Loss减少最少的节点
            - 选择最优子树：通过交叉验证法对每棵子树进行测试，选择最优子树
10. XGBoost的重要性计算
    - 分裂节点：根据结构分数的Gini指数来选择特征作为分割点

- 特征重要性：某特征在所有子树中出现的次数之和

11. XGBoost的正则项表达式
    - 即树的复杂度：叶子结点的个数+叶子结点的权重向量的L2范数

$$\sum_{k}^{K}\Omega(f_k)$$ ，其中

- $$\Omega(f_k)=\gamma T+\frac{1}{2}\sum_{j=1}^{T}w_j^2$$

T是叶子节点的数量

12. XGBoost如何控制过拟合
    - 预剪枝：损失函数中添加了正则项，控制模型复杂度（叶子节点个数+叶子结点权重分数的L2范数）
    - 缩减因子：每完成一次迭代，给树的叶子节点的权重乘以缩减因子，削弱每棵树的影响
    - 随机森林思想：每个树生成时，采用列采样，随机选择特征子集，实现分裂节点处特征的选择
13. XGBoost并行化的实现
    - XGBoost并行是在特征粒度上的并行
        - 因为要确定最佳切分点，因此需要对特征的值进行排序
        - XGBoost预先根据每个特征的特征值对样本进行排序，保存为block结构，后续的迭代计算重复使用此block结构
        - 在进行节点分裂时，选择增益最大的特征做分裂。各个特征的增益计算可以开多个线程同时进行
14. XGBoost和LightBGM的区别
    - 树的切分策略：
        - XGBoost：采用level-wise的分裂策略，对每一层的所有节点都进行无差别分裂
        - LightBGM：采用leaf-wise的分裂策略，选择当前所有叶子结点中分裂后增益最大的节点进行分裂（容易使树过深，过拟合）
    - 实现并行化的方式：
        - XGBoost：通过预排序的方式
        - LightBGM：通过直方图算法对特征值进行离散化处理（分桶后只需保存特征离散化后的值）
15. GBDT应用与多分类任务如何实现?
    - 对于多分类任务，GDBT会针对每个类别训练M个基分类器，假设有K个类别，总共包含K*M个基分类器
    - 训练策略：K个类别都拟合完第一棵树，才开始拟合第二棵树，不允许将一个类别的M棵树都拟合完再学习其他类别
    - 做法

二、GBDT如何用于分类的

第一步：**训练的时候，是针对样本 X 每个可能的类都训练一个分类回归树。**如目前的训练集共有三类，即K = 3，样本x属于第二类，那么针对样本x的分类结果，我们可以用一个三维向量$[0,1,0]$来表示，0表示不属于该类，1表示属于该类，由于样本已经属于第二类了，所以第二类对应的向量维度为1，其他位置为0。

针对样本有三类的情况，我们实质上是在每轮的训练的时候是同时训练三颗树。第一颗树针对样本x的第一类，输入是$(x,0)$，第二颗树针对样本x的第二类，输入是$(x,1)$，第三颗树针对样本x的第三类，输入是$(x,0)$。

在对样本x训练后产生三颗树，对x 类别的预测值分别是$f_1(x),f_2(x),f_3(x)$，那么在此类训练中，样本x属于第一类，第二类，第三类的概率分别是：

- $$P_1(x)=exp(f_1(x))/\sum_{k=1}^{3}exp(f_k(x))$$
  $$P_2(x)=exp(f_2(x))/\sum_{k=1}^{3}exp(f_k(x))$$
  $$P_3(x)=exp(f_3(x))/\sum_{k=1}^{3}exp(f_k(x))$$

然后可以求出针对第一类，第二类，第三类的残差分别是：

$$y_{11}=0-P_1(x)$$
$$y_{22}=1-P_2(x)$$
$$y_{33}=0-P_3(x)$$

16. XGBoost为什么使用二阶梯度信息，不使用更高阶?
    - GDBT利用了一阶梯度的泰勒展开，丢失了较多的信息
    - 理论上，泰勒展开式展开的次数越多对原始函数的描述更加精准，但n阶展开就需要求函数的n阶导，很多函数不是n阶可导的
    - 二阶导是泰勒展开和损失函数的折中选择
17. Bootstrap方法是什么?
    - 在数据集中有放回的随机采样N次
18. 随机森林与Bagging的区别
    - RF会有放回的选取与样本集同样大小的数据来训练，而Bagging中训练集的大小一般会小于样本集
    - RF只选用部分特征来训练分类器，而Bagging一般使用全部特征来训练分类器
19. 提升树
    - 思想：提升树是一种迭代的训练多棵回归树然后利用他们进行共同决策的方法
    - 当损失函数是平方误差损失时，当前树学习的是之前所有树的结论的残差，拟合这个残差得到残差回归树，最后累加所有树的结果作

为最终结果
    - GBDT
        - 梯度提升树，当损失函数不是平方误差时，残差不再是简单的真实值减去预测值。因此GBDT利用损失函数的负梯度值作为残差的近似值来得到模型改进的方向
    - XGBoost
        - XGBoost是对GBDT的改进，在目标函数的基础上加了正则化项，正则化项包括树的叶子结点的个数，以及每个叶子结点输出分数的L2范数。其次，对目标函数做了二阶泰勒展开，从而提高了模型的训练速度和预测精度
    - XGBoost为什么要进行泰勒展开
        - 一阶导指引梯度方向，二阶导数指引梯度方向如何变化，有利于梯度下降更快更准
        - 便于自定义损失函数，只要二阶导存在即可
20. XGBoost是树模型，可以进行线性回归拟合吗?
    - XGBoost模型比线性回归模型复杂
    - XGB可以拟合出直线，但不是线性模型，只是当与线性回归模型输入相同时，输出也相同罢了
    - 如果用XGB拟合简单的直线的话，容易过拟合
21. XGB对缺失值的处理策略
    - XGB的基学习器可以是树模型，也可以是线性分类器
    - 训练过程中：
        - 当基学习器是线性模型时
            - 遇到特征f的缺失值，直接填充为0
        - 当基学习器是树模型
            - 对于特征f非缺失的样本数据，选择最优损失对应的（特征-特征值对）作为分裂节点的特征（即选取某个特征的某个阈值）
            - 对于特征f有缺失的样本数据，将该缺失值分别划分到左子树和右子树，分别计算左子树和右子树的损失，选择最优损失的方向为缺失值的分裂方向
    - 预测过程中：
        - 如果训练过程中，特征f出现过缺失值，按照训练过程中缺失值的划分方向进行划分
        - 如果训练过程中，特征f没有出现过缺失值，将缺失值划分到默认方向（左子树）
22. 为什么XGBoost模型的叶子结点值越大，越容易过拟合
    - 模型过拟合的一个原因
        - 模型在训练过程中太过关注数据集中每一个样本的特性，包括噪声，从而导致训练数据集loss低，测试数据集loss大
    - 叶子结点权重过大的影响
        - 叶子结点的权重过大，也就意味着模型容易依赖于训练数据集中某一部分的数据特征
        - 当多个树集成后进行共同决策时，每个模型预测结果的波动会比较大（方差较大），从而使得集成模型缺乏强的泛化能力，即过拟合
23. XGBoost的叶子结点的权重是如何计算的
    - 按照当前叶子结点上所有样本的一阶梯度和二阶梯度计算当前叶子节点的权重
24. GBDT残差为什么用负梯度表示
    - 负梯度方向是函数值下降最快的方向，因此用梯度来拟合当前模型
    - 用残差去拟合当前模型只是目标函数是平方误差的一种特殊情况，平方误差求导后。就是残差，因此看起来拟合残差合情合理
    - 对于一般的损失函数，如果只拟合残差，相当于只关注于最小化经验风险，但损失函数中一般还包括正则项，如果只关注经验风险损失，那么结构风险可能不会变小，甚至会由于过拟合而变大，此时整体的目标函数也不会变小。
    - 梯度综合了各个参数方向的变化，会选择一个总是最优的方向来优化目标函数

SVM

2022年4月26日 20:57

1. SVM什么时候用线性核，什么时候用高斯核？（机器学习-P69）
    - 线性核：当样本在输入空间线性可分时，当特征维数大于样本数时
    - 高斯核：当样本在输入空间线性不可分时，当特征维数小于样本数时
2. SVM与逻辑回归LR的区别？（机器学习-P75）
    - 相同点：LR和SVM都是有监督的二分类模型
    - 不同点：
        - LR的损失函数为交叉熵；SVM的损失函数为合页损失函数（自带正则项）
        - LR通过最大似然确定模型参数；SVM的学习策略为分类间隔最大化
        - SVM决策超平面由少数的支持向量决定；LR的决策面需要考虑全局
        - SVM对异常值不敏感（改变非支持向量不会改变决策面）；LR改变任意一个样本值都会导致决策面的变化
        - 对于非线性问题：
            - LR需要进行特征线性组合，特征离散化，在高维特征空间表现较差
            - SVM利用核函数将非线性数据映射到另一个空间，使得数据在新的空间尽量线性可分，此时只有支持向量参与运算，计算复杂度较低
3. SVM的作用及基本原理？
    - 作用：SVM是一个二分类模型，它的目的是找到一个分离超平面对所有样本进行分割
    - 原理：
        - 寻找最佳分离超平面的原则是间隔最大化（线性可分：硬间隔最大化；近似线性可分：软间隔最大化；线性不可分：核函数+软间隔最大化）
        - 最佳分离超平面需要满足：每个类别中离超平面最近的点的到超平面的距离最大
        - 整个过程：SVM通过间隔最大化原则寻找一个最优的分离超平面将数据集中的样本分隔开，最大化间隔可以构造一个约束最优化问题，然后利用拉格朗日函数把约束问题转换为无约束问题，得到对应的对偶问题，求解对偶问题，得到最优超平面的参数
4. SVM引入对偶问题的目的？
    - 原始最优化问题的求解较复杂，与特征维数有关，而转化为对偶问题后只与变量个数有关，变量个数即为支持向量的个数，比特征维数少。
    - 通过拉格朗日算法将带约束的优化问题转化为不带约束的优化问题，更易计算和理解
    - 方便引入核函数（拉格朗日表达式中有内积，而核函数也是通过内积映射的），方便推广到非线性分类问题中
    - 无论原始问题是否是凸函数，对偶问题都是凸优化问题
5. SVM的物理意义？
    - 构建一个最优的分离超平面将所有数据分离
6. 核函数的种类以及应用场景？
    - 核函数种类：线性核，高斯核，多项式核
    - 场景：
        - 特征维数很多 或者 样本数量非常多时 问题是线性可分的时 采用线性核（避免庞大的计算量）
        - 样本数据量适中，特征维数较少， 遇到的问题是线性不可分时 选择高斯核
7. 核函数的作用？
    - 将样本从原始低维空间映射到更高维的空间，并解决原始特征空间的线性不可分问题

    - 核函数将原始输入空间的不可分数据映射到高维空间使他们尽量线性可分，新的空间可能是很高维甚至是无限维，难以计算也难以表示。因此不显示的定义该映射函数，而是定义一个函数，将两个样本直接映射为他们在高维空间的内积，因为求解对偶问题时，只需要用到内积。
8. 为什么高斯核可以拟合无穷维度?
    - 因为高斯核中有exp()，而exp(x)的泰勒展开式是一个无限维的表达式
9. 支持向量机可以用于解决线性回归问题吗?
    - 支持向量机可以推广到解决回归问题，称为支持向量回归（SVR)
    - 与线性回归模型的区别
        - 线性回归：只有当样本预测值完全等于标签值时，才认为是预测正确，否则需要计算损失值
        - SVR：宽容的回归模型，只要样本预测值与标签值的偏差不太大，小于设置的阈值，即可认为预测正确，不需要计算损失
10. 随机森林与SVM的区别
    - 随机森林
        - 思想：多颗决策树共同决定输出结果（类似于if-else判断机制）
        - 能够处理很高维的数据（随机选择特征）
        - 训练速度快，基于树粒度的并行处理
        - 训练结束后，可以知道特征的重要性（特征在节点中出现的次数）
        - 对于有缺失值的数据，或者不平衡数据集都不太敏感
    - SVM
        - 思想：寻找最优分离超平面对数据集进行分割（间隔最大化）
        - 解决小样本下的分类问题
        - 可以解决高维问题，但样本数较多时，效率较低
        - SVM的决策函数由少量支持向量决定，计算复杂度取决于支持向量的个数，而不是样本空间的维数
        - 能够处理非线性样本数据，但核函数的高维映射解释性不强

线性回归，逻辑回归，决策树

2022年4月26日 20:55

1. 逻辑回归LR原理（机器学习-P1）
    - 逻辑回归是在线性回归的基础上增加了Sigmoid函数作为预测函数，将连续值映射为0 or 1
    - 线性回归的损失函数为平方误差损失，LR为交叉熵损失（将平方损失用于LR，那么损失函数时非凸的，难以利用梯度下降求解最优值）
    - 线性回归的参数计算方式是最小二乘法，LR的参数计算方式为最大似然估计（最小化交叉熵损失）
2. 逻辑回归怎么实现多分类?
    - 利用softmax代替sigmoid作为最终的激活函数，实现多分类输出，此时损失函数不再是用于二值分类的交叉熵损失函数，而是softmax loss
    - 构建多个逻辑回归分类器。假设有K个类别，对于每个类别都训练一个LR分类器。（本类别的标签为1，其余类别标签为0）
    - 如果多个类别之间明显互斥则采用softmax函数；如果多个类别之间不互斥有交叉情况则构造多个LR分类器
3. 如果给你一些数据，你会如何分类？（数据大小，特征维数，缺失值）（机器学习-P26)
    - 如果数据集比较大，可以选择LR逻辑回归模型
    - 如果数据的特征维数较多，可以使用SVM支持向量机（仅与少数基向量有关）
    - 如果存在缺失值，可以选择决策树模型
4. 如果数据有问题怎么处理?
    - 数据样本太少：数据增强，增加样本量
    - 数据质量不可靠：数据预处理，修改错误值，缺失值（均值插补，同类均值插补）， 异常值等
    - 数据不平衡：数据采样（欠采样，过采样）或者选择合适的评估标准
5. 线性回归与逻辑回归是区别？（机器学习-P32)
    - 线性回归用来做预测，LR用来做分类
    - 线性回归的输出值是连续值，LR的输出值是离散值
    - 线性回归的输出直接是AX,而AX在LR中只是一个决策边界
    - 线性回归用最小二乘法计算参数，逻辑回归用最大似然法计算参数
    - 线性回归对异常值较敏感，逻辑回归对异常值有较好的稳定性
6. 分类算法的种类以及应用场景？（机器学习-P26）
    - 贝叶斯分类器：
        - 优点：对缺失数据不敏感，结果可解释性强
        - 缺点：需要特征属性之间的独立性假设，需要先验概率
    - 逻辑回归：
        - 优点：易计算，易更新
        - 缺点：需要归一化处理
    - SVM:
        - 优点：解决超高维，非线性问题，对异常值不敏感，适用于小样本数据集
        - 缺点：对缺失数据敏感
    - 决策树
        - 优点：处理多种类型的数据，适合高维数据
        - 缺点：容易过拟合，不适合样本类别不平衡的数据集（信息增益偏向于样本多的特征），忽略了属性之间的关联性
    - KNN
        - 优点：思想简单易计算，可回归可分类
        - 缺点：计算量大，内存消耗量大，不适应与样本不平衡数据集
    - AdaBoost:
        - 优点：可以用各种方式生成个体学习器，不易过拟合
        - 缺点：异常值敏感
    - 神经网络

    - 优点：鲁棒性强，并行处理能力强，不易受噪声影响
    - 缺点：参数多，解释性弱，训练时间长
7. ID3，C4.5，CART三种决策树的区别？（机器学习-P85）
    - ID3：使用信息增益（大）为准则进行特征选择，递归的构建决策树
        - 信息增益偏向于选取取值数目较多的特征，易过拟合
        - ID3剪枝原则是最小化损失函数，通过比较某一分支被剪前后损失函数的变化来进行
    - C4.5：使用信息增益比（大）为准则进行特征选择，递归的构建决策树
        - 信息增益比在信息增益的条件下添加了惩罚项
    - CART：使用基尼指数（小）为准则进行特征选择，递归的构建决策二叉树（是 or 否）
        - CART树既可以用于回归，又可以用于分类
        - ID3和C4.5只能处理离散型数据，CART可以处理离散型和连续性数据
        - 回归树生成以最小化平方误差为准则；分类数生成以最小化基尼指数为准则
        - CART剪枝从决策树的叶子结点开始不断剪枝（通过衡量减掉某一个节点后整体Loss的减小程度）直到决策树的根节点，获得一个子树序列后，通过交叉验证（Gini指数 or 平方误差 最小）选择最优子树。
8. 传统机器学习模型?
    - 按照有监督和无监督可分为：
        - 有监督：逻辑回归，线性回归，决策树，随机森林，SVM，贝叶斯，KNN, 提升方法（AdaBosst, GBDT）
        - 无监督：SVD奇异值分解，K-means聚类， 主成分分析等
    - 按照集成和非集成可分为：
        - 非集成：以上方法都可以作为个体学习器的学习模型
        - 集成：Boosting（AdaBoost, GBDT, XGBoost), Stacking, Bagging（随机森林）
9. 哪些模型需要归一化?
    - 是否归一化主要取决于模型是否关心变量取值
        - 不需要归一化
            - 概率模型不需要归一化，因为这种模型不关心变量的取值，而是关心变量的分布和变量之间的条件概率
                - 决策树（概率模型）、随机森林（基学习器是决策树）、朴素贝叶斯（概率模型）不需要归一化
        - 需要归一化
            - SVM、线性回归之类的最优化问题
                - 归一化之后加快了梯度下降求最优解的速度，并有可能提高精度。
            - 神经网络
                - 一般变量的取值在-1到1之间，这样做是弱化某些变量的值较大而产生消极影响。一般神经网络中的隐藏层采用tanh激活函数比sigmoid要好，因为取值[-1, 1]，均值为0
            - 在k近邻算法
                - 如果不对解释变量进行标准化，那么具有小数量级的解释变量的影响就会微乎其微

需要	不需要
LR（线性回归、逻辑回归）	决策树
SVM（支持向量机）	随机森林
KNN	朴素贝叶斯
K-Means	XGBoost
高斯过程	lightGBM
AdaBoost	
神经网络	
LSTM	
GBDT	

10. LR解决非线性分类问题
    - 加核函数（类似于SVM），显式的将特征映射到高维空间使其在高维空间线性可分
11. 逻辑回归为什么对特征进行离散化

- 离散特征的增加和减少都比较容易，易于模型的快速迭代
- 离散化之后的数据对异常值具有更强的鲁棒性

12. 随机森林是否需要交叉验证?
    - 不需要
    - 随机森林在训练时，采用Bootstrap采样方法，每一次约有1/3的数据不会出现在训练数据集中，这部分没有出现的数据，被称作袋外数据（OOB）
    - 袋外数据OOB可以用于模型验证
13. KNN算法
    - 有监督分类方法
    - 原理
        - 给定样本数据集，包含每个样本的类别标签
        - 对于输入数据X：计算X与样本数据集中每个样本的距离，选择距离最近的K个样本
        - X的类别由K个带标签样本投票决定（X的类别为多数类别）
    - 三个基本要素
        - K值的选择：采用交叉验证法来选取最优的k值
        - 距离的度量方式：曼哈顿，欧氏距离
        - 决策方式：多数表决
    - 缺点
        - 对异常值敏感
        - 计算量大
14. KD树
    - 动机
        - KNN每次预测一个点的类别时。都需要重新计算当前点与每个样本点的距离
        - 当数据集比较大时，计算成本非常高
    - 思想
        - KD树将距离信息存储在一颗二叉树里，在计算之前先在树上进行检索，避免了重复计算
        - 利用KD树避免了对大部分数据点的搜索，提高计算效率
        - e.g. A和B距离很远，B和C距离很近，那么A和B距离很远
    - 方法
        - KD树是对K维空间的一个划分
        - 递归的用垂直于坐标轴的超平面对K维空间进行划分，构造一系列的K维超矩形区域，生成子节点
        - 直到子区域中不存在样本点为止
15. 线性分类器
    - 概念
        - 模型的决策函数是线性函数，分类平面是（超）平面
        - 线性分类器通过样本特征的线性组合来得到分类决策结果
        - 线性分类器可以看做在高维空间中找到一组超平面，将样本空间划分了两个区域
        - 线性分类器能找到一组权值向量，使得判别公式可以写成特征值的线性加权组合
    - LR回归模型也是线性分类器
        - 特征经过Sigmoid的转换，映射为离散值
        - 但LR的决策边界仍是线性方程
    - 朴素贝叶斯本质上是一种线性分类器
    - 常见的线性分类器有：LR，贝叶斯分类，单层感知机、线性回归。
    - 常见的非线性分类器：决策树、RF、GBDT、多层感知机MLP

优化方法，损失函数

2022年4月26日 11:13

1. 牛顿法，拟牛顿法的原理?
    - 牛顿法：
        - 思想：牛顿法是梯度下降法（“之”字形下降，相邻两次下降方向正交，收敛速度慢）的进一步发展，在梯度下降法只计算目标函数一阶偏导的基础上，还计算了二阶偏导，在每一次迭代过程中考虑到了梯度的变化趋势
        - 缺点：
            - 对目标函数要求较高，需要满足一阶导数和二阶导数都存在，对应的Hessian矩阵正定
            - 除了需要计算梯度外，还需要计算Hessian矩阵和其对应的逆矩阵，计算量和存储量都比较大
    - 拟牛顿法：
        - 思想：构造一个正定矩阵来近似表示Hessian矩阵的逆，简化运算的复杂度（根据拟牛顿条件，构造正定对称矩阵）
        - 优点：虽然不能像牛顿法一样保证最优化的方向，但逆矩阵是正定的，可以保证算法一直向最优的方向搜索
2. 互信息，熵， 条件熵，相对熵，交叉熵（融合分类-P6)
    - 熵：用于衡量随机变量的混乱程度（不确定性）
    - 条件熵：用于衡量随机变量X给定条件下，随机变量Y的不确定性
    - 互信息： 随机变量X中包含的随机变量Y的信息量，反应了两种变量之间的相互依赖程度
    - 相对熵：也称为KL散度。
        - 给定同一个随机变量的两个概率分布，相对熵用于衡量两个分布之间的差异
        - KL越小，两个分布越相似
    - 交叉熵：
        - 交叉熵用于衡量在给定真实分布的情况下，利用非真实分布所制定的策略来消除系统的不确定性所需要付出的努力的大小
        - 在机器学习中，交叉熵用来衡量数据的真实分布与模型学习到的分布之间的差异。通常作为逻辑回归问题的损失函数
3. L1和L2正则化的区别？（融合分类-P5)
    - L1正则化：
        - 形式：L1正则化是参数向量W中每个元素的绝对值之和
        - 作用：L1正则化使得模型的权值稀疏，可用于特征选择（过滤掉不重要的特征），也可用于防止过拟合
        - 图形：约束条件为菱形，非零特征值总是出现在坐标轴上（顶点处）
    - L2正则化：
        - 形式：L2正则化是参数向量中每个元素的平方和再开根号
        - 作用：L2使模型的权值更小，平滑权重，防止过拟合、
        - 图形：约束条件是圆，相比于L1更加的平滑，会使权值更小，但权值为0的几率很小
4. 损失函数的种类?
    - 线性回归：平方损失（最小二乘法）
    - 逻辑回归：交叉熵损失（极大似然估计）
    - SVM：合页损失
    - AdaBoost:指数损失函数
5. 如何防止过拟合?
    - 增强数据集
    - 特征筛选
    - 早停法：当模型在验证集上的泛化误差增大时，停止训练
    - 决策树剪枝
    - L1和L2正则化
    - 神经网络的Dropout，BN层
    - SVM的松弛变量
    - 集成学习（随机森林）
6. Focal Loss（深度学习-P67）
    - 作用：解决目标检测任务中难易样本不平衡问题。提高难样本对损失函数的贡献度，使模型更倾向于关注难样本
    - OHEM：在梯度反向传播过程中，将简单样本的梯度设置为0，只回传难样本的梯度
    - S-OHEM：采用分层抽样的方法选择难样本（e.g., 高分类误差，低回归误差），进行梯度回传
    - Focal Loss建立在交叉熵损失函数的基础之上

        - 交叉熵损失：当负样本较多时，负样本损失占主导地位，损失函数发生倾斜
        - 平衡交叉熵损失：给损失函数的添加权重因子，提高样本少的类别在损失函数中的权重（权重因子是超参）（更加注重正负样本不均衡问题）
        - Focal Loss：降低简单负样本在损失函数中的权重，增加难样本的贡献程度，权重动态调整
7. AUC的理解
    - AUC是ROC曲线下的面积
    - 反映了模型对样本的排序能力，即AUC反应了任意选取一个正样本和一个负样本，正样本得分大于负样本得分的概率
    - AUC（ROC）对正负样本比例不敏感
        - ROC曲线以假正率FPR为横坐标，真正率TPR为纵坐标
        - FPR = FP/N = FP/(TN+FP) TPR = TP/P = TP/(TP+FN)
        - 真正率只与正样本有关，假正率只与负样本有关，他们各自增大不会影响彼此
8. 特征选择的作用及方法
    - 作用：
        - 减少特征数量，降低计算复杂度
        - 降维
        - 提高模型泛化性能，避免过拟合
    - 方法：
        - 去除方差较小的特征
        - L1正则化（稀疏特征）
        - 随机森林
9. 归一化和标准化的区别
    - 相同点
        - 都避免了数据由于量纲不同引起的误差
    - 归一化
        - 将样本的特征值转换到同一量纲之下，将数据映射到【0,1】或者【-1,1】区间内，仅由变量的极值决定
        - 最大最小归一化
        - 易受到异常值的影响
    - 标准化
        - 标准化的输出范围不受限制，通常是按照特征矩阵的列处理数据，将其转换为标准正态分布，和数据的整体分布情况相关
10. 误差、偏差和方差的区别与联系
    - 偏差：衡量分类器的预测结果与训练数据的真实标签之间的差距，反应模型拟合训练数据的能力
    - 方差：方差描述分类器在不同迭代训练阶段中，预测值的变化波动情况，反应模型的稳定性
    - 噪音：反映问题本身的难度
    - 误差：反映的是整个模型的准确度
        - 误差 = 方差 + 偏差 + 噪音
    - 偏差和方差关系
        - 低偏差，低方差：理想模型，准确度高且稳定
        - 低偏差，高方差：过拟合，泛化能力弱
        - 高偏差，低方差：通常是模型训练初始阶段的状态
        - 高偏差，高方差：最糟糕的情况，准确度差，稳定性低

特征降维，聚类

2022年4月26日 10:51

1. 讲一下PCA？(机器学习-P49)
    - PCA是一种无监督的线性的降维方法
    - 目的：通过一定的线性变换，将高维数据映射到低维空间中，并期望在投影方向上数据的信息量最大（方差最大），使用较少的维度，尽可能多的保留原始数据的信息
    - 实现方法：
        - 样本去中心化$(x-\bar{x})$
        - 计算样本协方差矩阵$(x-\bar{x})(y-\bar{y})$
        - 对协方差矩阵做特征分解，求得特征值及特征向量
        - 特征值排序，选择最大的前N个特征值对应的特征向量
        - 将原始数据映射到特征向量（正交）构建的新空间中
    - 优缺点：
        - 优点：特征向量正交，因此各主成分之间正交，消除了原始数据成分之间的相互影响
        - 缺点：方差小的非主成分也可能包含重要信息，直接丢弃会损失信息
2. 聚类算法分类（机器学习-P81）
    - 基于分层的聚类算法：
        - 由上到下分裂的层次聚类：初始时所有样本属于一个类，之后将类中距离最远的两个样本分到两个新类，重复此操作直到满足停止条件
        - 由下到上聚合的层级聚类：初始时每个样本都是一个类，之后将距离最近的两个类合并形成一个新类，重复此操作直到满足停止条件（类数为1）
    - 基于划分的聚类算法：K-means算法
        - 步骤：
            - 选取K个样本点作为初始类中心
            - 计算每个样本点到K个类中心的距离，将其分到距离最近的类中
            - 计算新的类中心，再次重复上述步骤
            - 直到类中心不再发生变化或者损失函数（每个样本点到所属聚类中心的距离之和）小于给定阈值
        - 优点：可以处理大规模数据集
        - 缺点：K-means算法是局部最优的，依赖于初始值K，对异常值较敏感（异常值影响均值的计算），不适用于发现非球状的类
        - K值的选择：
            - 最有效的方法：根据先验知识，提前设置阈值K的大小
            - 轮廓系数法：尝试用不同的K值进行聚类，检测不同K的聚类效果，选择最优K（类内样本的距离越近，类间样本的距离越远，平均轮廓系数越大，聚类效果越好）
            - 类的平均直径衡量：K越小，平均直径越大；当K大于某个值时，平均直径不再发生变化，此值为最优K
            - Calinski-Harabasz准则：计算类间方差/类内方差（值越大，数据分类度越大）
        - 时间复杂度
            - O(TMNK)
                - T：迭代次数
                - M：样本个数
                - N：每个样本的特征维数
                - K：聚类个数
        - 空间复杂度
            - O((M+K)*N)
                - 需要存储样本点和类中心点，他们都有N维
    - 基于密度的聚类方法（DBSCAN)
        - 思想：
            - 将簇定义为密度相连（密度直达，密度可达，密度相连）的点的最大集合。将高密度的区域划分为一个簇，并在有噪声的区域发现任意形状的簇。
        - 步骤：
            - 定义领域半径和半径内元素个数阈值
            - 随机寻找一个核心对象（领域内点的个数大于阈值）作为初始簇，领域内的点作为边缘点
            - 任意选取一个边缘点，合并与核心对象密度可达的点
            - 重复上述步骤，直到没有新的点可以更新簇

        - 如果还有未处理的点，再次产生一个新的类别来重新启动此算法
        - 如果有样本点既不是边缘点也不是中心点，标记为噪声
    - 优点：不需要确定类的个数，可以发现任意形状的簇，对异常值不敏感
    - 缺点：不适用于高维数据的聚类，需要定义两个参数，参数对聚类结果影响较大
3. K-means与分类方法的区别
    - K-means也属于分类算法，是一种无监督分类算法
    - 无监督：因为样本不包含类别标签，模型自己学习样本数据的内部分布规律
    - 对K值敏感的原因
        - K值决定了聚类的个数，很难确定分为多少各类才合适
        - 不同的聚类中心会产生不同的聚类结果
4. 线性判别分析LDA原理（机器学习-P45）
    - LDA是一种有监督的降维方法，也可以用于分类
    - 思想：将数据集中的样本点映射到低一维空间，使得投影后同一类别的样本点距离尽可能近，不同类别样本点的距离尽可能远（类内方差最小，类间方差最大）
    - 步骤
        - 计算各个类别的均值以及协方差矩阵
        - 计算类内散度矩阵S（各个类别协方差矩阵之和）
        - 计算类间散度矩阵P
        - 计算矩阵（S-1P）的特征值及特征向量，求得最大的N个特征值以及对应的特征向量
        - 将N个特征向量记为投影矩阵
        - 将样本通过投影矩阵映射到新的特征空间
5. PCA与LDA的区别
    - 相同点
        - 两者都可用于数据降维
        - 两者都采用了矩阵分解的思想
    - 不同点
        - LDA属于有监督方法，PCA属于无监督
        - LDA即可用于降维也可用于分类，PCA只能降维
        - LDA只能降维到k-1维，PCA降维数量没有限制
6. PCA与SVD的联系与区别
    - 相同点
        - 两者都属于降维方法，可用于数据压缩和去噪
    - 不同点
        - PCA需要对输入数据X求协方差矩阵，再得到协方差矩阵S的特征值和特征向量（特征向量代表了投影方向），协方差矩阵S是对称矩阵
        - SVD直接对输入数据X进行奇异值分解，得到X的奇异值和左右奇异矩阵
        - X的奇异值和S的特征值之间也存在一一对应关系
            - $\lambda_i = \frac{\Sigma_i^2}{m}$。
        - PCA可以通过SVD计算得到，通常来说更加容易计算，避免了协方差矩阵的计算

再说奇异值分解。对于X进行奇异值分解可得

$$X = U\Sigma V^T \tag{4}$$

其中，$U_{m\times m}$、$V_{n\times n}$均为单位正交矩阵，Σ仅在主对角线上有非零值，就是奇异值。

经过奇异值分解可以得到X的另一种表达形式，此时再计算X的协方差矩阵可得(这里依旧假设X已经预先进行了零均值化处理)：

$$S = \frac{1}{m}X^TX = \frac{1}{m}V\Sigma^TU^TU\Sigma V^T = V\frac{\Sigma^2}{m}V^T \tag{5}$$

现在对比式（5）和式（2）就会发现，X的奇异值和S的特征值存在一一对应关系: $\lambda_i = \frac{\Sigma_i^2}{m}$。
此时再将X投影到前k个主成分的方向上，可得

$$\widetilde{X} = XV_k = U\Sigma V^TV_k = U_k\Sigma_k \tag{6}$$

其中，U_k为U的前k列，Σ_k为 Σ左上角的$k \times k$部分。

对于样本集$X_{m\times n}=\{x_1;x_2;\ldots;x_m\}$每一行表示一个$n$维样本。PCA将$n$维样本$x_i$投影到一个低维空间中去从而实现降维。具体来说，可以从两个角度来约束投影：**最大化方差**和**最小化投影误差**。有意思的是，从这两个方向推导最后会得到相同的结果。

从最大化方差的角度理解，假设投影的方向为v，数据的均值为$\bar{x}=\frac{1}{m}\sum_{i=1}^{m}x_i$那么使得$X$在方向$v$上的投影方差最大也就是

$$\max_{v}\frac{1}{m}\sum_{i=1}^{m}(x_iv-\bar{x})^2=\max_{v}v^TSv \tag{1}$$

S为协方差矩阵，若提前对数据进行零均值化处理，则$S=\frac{1}{m}\sum_{i=1}^{m}x_i^Tx_i=\frac{1}{m}X^TX$由于$S$是实对称矩阵，因此可以对其进行对角化处理

$$S=VLV^T \tag{2}$$

其中$V_{n\times n}$为正交矩阵，即$V^TV=I$，V的每一列都是S的一个特征向量。$L_{n\times n}$为对角矩阵，对角线上的值为特征值，和V的每一列一一对应并且按值递减排列。将S代入目标函数（1）中去，可得

$$v^TSv=v^TVLV^Tv=\lambda \tag{3}$$

所以方差最大值就是最大的特征值。也就是说当投影方向是最大特征值对应的特征向量时，数据在投影方向上的方差最大。因此，主成分分析选取较大的k个特征值对应的特征向量进行。投影后的数据为$\widetilde{X}=XV_k$，V_k为最大的k个特征值对应的特征矩阵。这样就将X从n维空间压缩到了k维。

概率，贝叶斯

2022年4月26日 10:05

1. 最大似然估计和最大后验概率的区别？（机器学习-P75)
    - 相同点：最大似然和最大后验都是提供了一种，给定一组观测值，再进行模型参数评估的方法
    - 不同点：
        - 最大似然估计假设所有采样数据都是独立同分布的（均匀分布）
        - 最大后验概率假设参数服从一定的分布。即引入一定的外部先验知识来辅助参数估计，减少参数估计对训练样本的依赖性
2. 什么是共轭先验分布?
    - 当参数的先验分布与后验分布属于同一类分布时，则两者为共轭分布
    - 此时称该先验分布为共轭先验分布
    - e.g. 高斯分布-高斯分布， 二项分布-Beta分布
3. 概率与似然的区别?
    - 概率：表示已知模型参数的情况下，随机变量取某一特定值的可能性
    - 似然：而似然恰好相反，表示在给定了观测值（结果）的情况下去推测产生这个结果的可能环境（参数）
4. 频率学派与贝叶斯学派的区别?
    - 频率学派：认为数据在无限多的情况下，决策的规则是精确地，存在一个不变的规律，不相信先验的存在
    - 贝叶斯学派：认为世界无时无刻都在变化，未知的变量和事件都有一定的概率，后验概率是先验概率的修正
5. 机器学习中的距离计算方法？（融合分类-P8)
    - 欧氏距离，曼哈顿距离，夹角余弦，明可夫斯基距离等
6. 朴素贝叶斯法的要求？（机器学习-P75)
    - 朴素贝叶斯进行了特征条件独立性假设，假设所有属性相互独立
    - 朴素贝叶斯属于生成模型，需要根据先验概率计算联合概率和条件概率，最后计算后验概率
7. 朴素贝叶斯与LR的区别?
    - 朴素贝叶斯属于生成模型，先根据样本计算先验概率P(Y),再计算条件概率P(X|Y)(特征独立性假设)，根据两者计算出联合概率P(XY)，最后根据贝叶斯定理计算后验概率P(Y|X)
    - LR属于判别模型，通过最小化交叉熵损失函数直接求得后验概率P(Y|X)
    - 朴素贝叶斯适用于小样本数据集，而LR可应用于大规模数据集
8. 生成模型和判别模型的区别？(机器学习-P30)
    - 生成模型：
        - 特点：先学习数据的联合概率分布，分析数据本身的分布特性，能够反应同类数据本身的相似性，再计算后验概率用于分类
        - 模型：贝叶斯，隐马尔科夫模型
    - 判别模型：
        - 特点：直接学习输入数据与最终类别之间的决策函数（相当于直接计算后验概率），反映的是数据之间的差异性。
        - 模型：LR, SVM，决策树，KNN，神经网络，Boosting等
9. HMM隐马尔科夫模型的参数估计方法是什么？（机器学习-P78）
    - EM算法：最大期望算法
        - EM算法是一种通过迭代实现极大似然估计的方法
        - EM算法通常用于对包含“隐变量”的概率分布模型，进行参数估计
10. EM算法
    - 定义
        - EM是用于迭代的寻找概率模型参数的极大似然估计方法，其中概率模型依赖于无法观测的隐性变量
    - 方法
        - EM算法通过两个步骤交替进行
            - 第一步计算期望（E）
                - 利用对隐变量的现有估计值，计算其极大似然估计值
            - 第二步最大化（M）
                - 最大化在E步上求得的极大似然估计值来计算参数值
            - M上的参数值被用于下一次E步的计算，这个过程不断交替
11. 先验概率与后验概率的区别
    - 先验概率

        - 事件发生之前，根据以往的经验与分析得到的概率
    - 后验概率
        - 已经发生的事件，导致该事件发生的原因有多种，判断该事件发生时是因某种原因发生的概率
    - 联系
        - 后验概率的计算以先验概率为基础
        - 后验概率可以根据通过贝叶斯公式，用先验概率和似然函数计算出来