1. GBDT简介

GBDT(Gradient Boosting Decision Tree), 梯度提升树

GBDT模型是一个集成模型，基分类器采用CART，集成方式为Gradient Boosting

2. 决策树简介

决策树，就是一种把决策节点画成树的辅助决策工具，一种寻找最优方案的画图法。

通过对不同特征的优先级区分判断后，优先选择优先级高的特征作为划分的特征。

特征优先级的评价标准：

ID3算法，信息增益(Information Gain)

C4.5算法，信息增益比(Information Gain Ration / Gain Ratio)

CART算法，基尼系数(Gini Index)

3. 信息熵(Information Entropy)和信息增益(Information Gain)

1948年，香农引入了信息熵，将其定义为离散随机事件出现的概率

信息熵是对不确定性的度量

一个系统越有序，信息熵就越低，反之一个系统越是混乱，它的信息熵就越高

信息熵：系统有序化程度的度量

信息增益：以某一个特征划分某个信息系统后，这个系统整体信息熵降低的数值

信息熵的计算公式：

对结果随机变量X={x\_1, x\_2, … , x\_n}的取值为{p\_1, p\_2, … , p\_n}

那么，系统的熵定义为：



子系统信息熵：



划分后的整体信息熵：



信息增益：



4. ID3简介

ID3(Iterative Dichotomiser 3)

核心思想: 利用信息熵原理选择信息增益最大的属性作为分类属性，递归地拓展决策树的分枝，完成决策树的构造

缺点：ID3 算法倾向于选择取值比较多的属性。

如果我们把“编号”作为一个属性，那么“编号”将会被选为最优属性

而实际上，编号属性是无关属性，它对分类没有太大作用

5. C4.5简介

核心思想：利用信息增益率最大的属性作为分类属性

信息增益率=信息增益/属性熵

当属性值很多，虽然信息增益变大，但对于c4.5来说，属性熵也会增大

所以整体的信息增益率并不大

6. 基尼系数(Gini Index)





7. CART简介

CART(Classification And Regression Tree), 分类回归二叉决策树

CART算法仅支持二叉树

CART算法既可以做分类树，也可以做回归树

8. CART作为分类树

CART作为分类树时，特征属性可以是连续类型也可以是离散类型

但观察属性(即标签属性或者分类属性)必须是离散类型

对于离散属性，可能会出现属性取值数N>=3的情况，因为CART是二叉树，此时需要考虑将N>=3个取值的离散特征的处理时也只能有两个分支，这就要通过组合人为的创建二取值序列并取GiniGain最小者作为树分叉决策点。如某特征值具有[‘young’,’middle’,’old’]三个取值,那么二分序列会有如下3种可能性:

[((‘young’,), (‘middle’, ‘old’)), ((‘middle’,), (‘young’, ‘old’)), ((‘old’,), (‘young’, ‘middle’))]。

先把连续属性转换为离散属性再进行处理。虽然本质上属性的取值是连续的，但对于有限的采样数据它是离散的，如果有N条样本，那么我们有N-1种离散化的方法：<=vj的分到左子树，>vj的分到右子树。计算这N-1种情况下最大的信息增益率。另外，对于连续属性先进行排序（升序），只有在决策属性（即分类发生了变化）发生改变的地方才需要切开，这可以显著减少运算量。

（1） 对特征的取值进行升序排序

（2） 两个特征取值之间的中点作为可能的分裂点，将数据集分成两部分，计算每个可能的分裂点的GiniGain。优化算法就是只计算分类属性发生改变的那些特征取值

（3）选择Gini Gain最小的分裂点作为该特征的最佳分裂点

注意，若修正则此处需对最佳分裂点的Gini Gain减去log2(N-1)/|D|

（N是连续特征的取值个数，D是训练数据数目）

根据离散特征分支划分数据集时，子数据集中不再包含该特征（因为每个分支下的子数据集该特征的取值就会是一样的，信息增益或者Gini Gain将不再变化，这也是C4.5等决策树离散型特征不会被重复选择为节点分裂的属性）；而根据连续特征分支时，各分支下的子数据集必须依旧包含该特征（当然，左右分支各包含的分别是取值小于、大于等于分裂值的子数据集），因为该连续特征再接下来的树分支过程中可能依旧起着决定性作用。

9. CART作为回归树

回归树要求观察属性是连续类型，由于节点分裂选择特征属性时通常使用最小绝对偏差（LAD）或者最小二乘偏差（LSD）法，因此通常特征属性也是连续类型。

以最小绝对偏差（LAD）为例

(1)先令最佳方差为无限大bestVar=inf。

(2)依次计算根据某特征（FeatureCount次迭代）划分数据后的总方差currentVar（，计算方法为：划分后左右子数据集的总方差之和），如果currentVar

(3)返回最佳分支特征、分支特征值（离散特征则为二分序列、连续特征则为分裂点的值），左右分支子数据集。

10. Boosting

Boosting是一种模型的组合方式，我们熟悉的AdaBoost就是一种Boosting的组合方式。和随机森林并行训练不同的决策树最后组合所有树的bagging方式不同，**Boosting是一种递进的组合方式，每一个新的分类器都在前一个分类器的预测结果上改进**，所以说boosting是减少bias而bagging是减少variance的模型组合方式。

基本思想：通过某种方式使得每一轮基学习器在训练过程中更加关注上一轮学习错误的样本

11. Gradient Boosting

Gradient Boosting中则将负梯度作为上一轮基学习器犯错的衡量指标，在下一轮学习中通过拟合负梯度来纠正上一轮犯的错误。这里的关键问题是：为什么通过拟合负梯度就能纠正上一轮的错误了？Gradient Boosting的发明者给出的答案是：函数空间的梯度下降。

Boosting，迭代，即通过迭代多棵树来共同决策。这怎么实现呢？难道是每棵树独立训练一遍，比如A这个人，第一棵树认为是10岁，第二棵树认为是0岁，第三棵树认为是20岁，我们就取平均值10岁做最终结论？--当然不是！且不说这是投票方法并不是GBDT，只要训练集不变，独立训练三次的三棵树必定完全相同，这样做完全没有意义。之前说过，GBDT是把所有树的结论累加起来做最终结论的，所以可以想到每棵树的结论并不是年龄本身，而是年龄的一个累加量。GBDT的核心就在于，每一棵树学的是之前所有树结论和的残差，这个残差就是一个加预测值后能得真实值的累加量。比如A的真实年龄是18岁，但第一棵树的预测年龄是12岁，差了6岁，即残差为6岁。那么在第二棵树里我们把A的年龄设为6岁去学习，如果第二棵树真的能把A分到6岁的叶子节点，那累加两棵树的结论就是A的真实年龄；如果第二棵树的结论是5岁，则A仍然存在1岁的残差，第三棵树里A的年龄就变成1岁，继续学。这就是Gradient Boosting在GBDT中的意义

12. 毕业设计

申报课题：《因果推断：基于观测数据缩小分布距离的对抗表征学习》

关键词：因果推断 分布距离 表征 对抗学习 观测数据

课题背景：人类和其他智能生命在与物质世界进行交互的过程，可以凭借因果关联达到期望的效果。比如说，医疗学者凭借因果推理预测一种新药是否对患者有益；政策制定者凭借因果推断预测某项新政策对公民失业率的影响；市场从业者拼接因果推断某一项产品的预期收入等等。因果推断的核心问题是反事实结果框架处理和平衡混杂因素的影响。目前较为普遍的方法是平衡干预组和对照组分布来预测干预效果，但这些直接平衡方法可能在平衡的过程中，过度采纳了无用信息或忽视了重要混杂因素的效果。为了解决该问题，基于深度学习方法，我们提出一种基于干预选择和结果因子的神经网络平衡分布特征学习，并采用对抗网络学习新一轮平衡权重，进一步平衡分布。这样既可以学习到干预信息，也可以保留有效的用于预测结果的信息，并剔除无关信息的干扰，相信能进一步提高因果推断的统计特性。

计划工作：

10.23~11.23 完成网络框架搭建，在人工数据集上进行性能测试

12.01~03.01 完成理论推导，解决模型可能出现的问题，书写成稿