机器学习中的信息论模型——决策树

信息论与编码理论 期末大作业

助教：曹志崴

15754301870

cao_zhiwei@pku.edu.cn

理科二号楼2331

主要内容

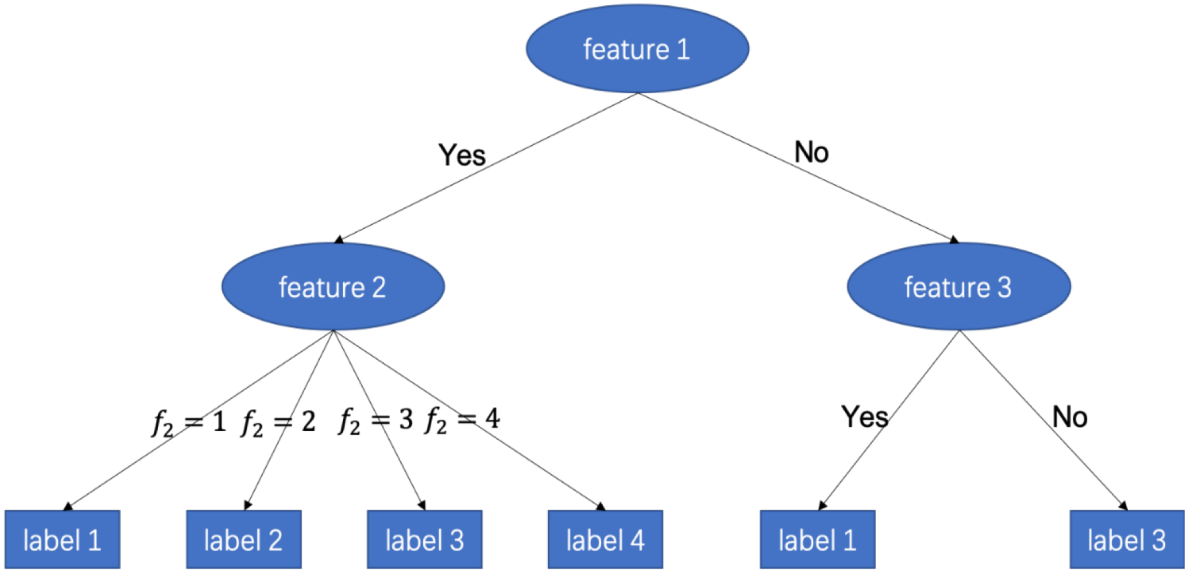
- 决策树简介（针对本次作业）
- 决策树构建的标准方法——ID3算法
- 决策树剪枝
- 决策树的模型集成方法——Bagging&Boosting
- 作业提交

主要内容

- **决策树简介（针对本次作业）**
- 决策树构建的标准方法——ID3算法
- 决策树剪枝
- 决策树的模型集成方法——Bagging&Boosting
- 作业提交

决策树简介

- 决策树是一类被广泛使用的分类与回归算法，它通过对问题相关的特征进行一系列测试，并且利用这些测试的结果确定最终查询样本的属类（对于分类任务）或是预测值（对于回归任务）。决策树由根结点，中间结点，叶子结点和分支结点构成，每一个非叶子结点都对应着针对一个特征的测试，被测试特征拥有的可能取值数目决定了其拥有的向下分支的数目。每一个叶子结点都对应了一种预测结果。



决策树示意图

主要内容

- 决策树简介（针对本次作业）
- **决策树构建方法——ID3算法**
- 决策树剪枝
- 决策树的模型集成方法——Bagging&Boosting
- 作业提交

决策树构建方法——ID3算法

2.1 信息增益 (Information Gain)

- 信息增益(Information Gain: IG)是一种对于某个特征所包含的信息量的度量。具体而言，某个特征的信息增益等于原数据集的熵减去对该特征进行测试并根据测试结果分割原数据集后各个子数据集的熵的加权和。信息增益的数学模型为：

$$IG(d,D) = H(t,D) - rem(d,D)$$

$$H(t,D) = -\sum_{l \in levels(t)} P(t=l) \times \log_2 P(t=l) \qquad rem(d,D) = \sum_{l=level(d)} \frac{|D_{d=l}|}{|D|} \times H(t, D_{d=l})$$

$H(t,D)$表示原数据集按照目标属类进行分类的熵，$rem(d,D)$表示将原集合D按照特征d进行划分之后各个子数据集的熵的加权和

决策树构建方法——ID3算法

2.1 信息增益 (Information Gain)

- 由IG定义，可以总结计算IG的3个步骤：

（1）计算原数据集按照目标属类进行划分的信息熵$H(t,D)$；

（2）对于一个特征，根据其不同的取值，将数据集划分为若干子数据集，计算每个子数据集内部按照目标属类进行划分的信息熵$H(t,D_{d=l})$，并对各个子数据集的结果进行加权求和；

（3）原数据集的信息熵减去（2）中得到的加权和，得到该特征的信息增益。

决策树构建方法——ID3算法

2.1 信息增益 (Information Gain)

IG算例

ID	SUSPICIOUS WORDS	UNKNOWN SENDER	CONTAINS IMAGES	CLASS
376	true	false	true	spam
489	true	true	false	spam
541	true	true	false	spam
693	false	true	true	ham
782	false	false	false	ham
976	false	false	false	ham

- 上面是一个简单的用于垃圾邮件分类的数据集，下面以这个数据集为例说明IG的计算方式。

决策树构建方法——ID3算法

2.1 信息增益 (Information Gain)

IG算例

ID	SUSPICIOUS WORDS	UNKNOWN SENDER	CONTAINS IMAGES	CLASS
376	true	false	true	spam
489	true	true	false	spam
541	true	true	false	spam
693	false	true	true	ham
782	false	false	false	ham
976	false	false	false	ham

- 第一步：计算整个数据集D根据目标变量(Class)划分后的熵：

$$H(t,D) = -\sum_{l\in\{soam,ham\}} P(t=l)\times\log_2 P(t=l)$$
$$= -[P(t=spam)$$
$$\times\log_2 P(t=spam) + P(t=ham)\times\log_2 P(t=ham)]$$
$$= -\left[\frac{3}{6}\times\log_2\frac{3}{6}+\frac{3}{6}\times\log_2\frac{3}{6}\right] = 1\ bit$$

决策树构建方法——ID3算法

2.1 信息增益 (Information Gain)

IG算例

ID	SUSPICIOUS WORDS	UNKNOWN SENDER	CONTAINS IMAGES	CLASS
376	true	false	true	spam
489	true	true	false	spam
541	true	true	false	spam
693	false	true	true	ham
782	false	false	false	ham
976	false	false	false	ham

- 第二步：计算以某个特征对D进行划分之后各个子数据集的熵的加权平均（以Suspicious Word特征为例）：

$$rem(\text{Words, D}) = \left(\frac{|D_{\text{Words=true}}|}{|D|}\times H(t, D_{\text{Words=true}}) + \frac{|D_{\text{Words=false}}|}{|D|}\times H(t, D_{\text{Words=false}})\right)$$

$$= \left\{\frac{3}{6}\times\left[-\sum_{l\in\{soam,ham\}} P(t=l)\times\log_2 P(t=l)\right] + \frac{3}{6}\times\left[-\sum_{l\in\{soam,ham\}} P(t=l)\times\log_2 P(t=l)\right]\right\}$$

$$= \left\{\frac{3}{6}\times\left[-\left(\frac{3}{3}\times\log_2\frac{3}{3}+\frac{0}{3}\times\log_2\frac{0}{3}\right)\right] + \frac{3}{6}\times\left[-\left(\frac{0}{3}\times\log_2\frac{0}{3}+\frac{3}{3}\times\log_2\frac{3}{3}\right)\right]\right\} = 0\ bit$$

决策树构建方法——ID3算法

2.1 信息增益 (Information Gain)

IG算例

ID	SUSPICIOUS WORDS	UNKNOWN SENDER	CONTAINS IMAGES	CLASS
376	true	false	true	spam
489	true	true	false	spam
541	true	true	false	spam
693	false	true	true	ham
782	false	false	false	ham
976	false	false	false	ham

- 第三步：得到给定特征的信息增益IG：

$$IG(Words, D) = H(t, D) - rem(Words, D) = 1 - 0 = 1bits$$

决策树构建方法——ID3算法

2.1 信息增益 (Information Gain)

- IG总结：

信息增益的物理意义类似于**互信息**，某个特征的信息增益越大表示将原数据集按照这个特征进行划分之后得到的子数据集的熵越小，或者等价地说，**针对该特征的测试能够尽可能多地减少原数据集的不确定性**。因此，IG天然地适合于进行**特征选择**——每当我们需要决定根据某个特征对数据集进行划分时，我们只需要计算各个特征的信息增益并选择具有最大信息增益的特征即可。

决策树构建方法——ID3算法

2.2 基于信息增益的决策树构建算法——ID3

ID3算法是一个**递归的**、**深度优先**的决策树构造算法，整个决策树的构造过程依赖于多叉树的**深度优先遍历**。

算法首先根据信息增益选择一个特征进行测试，并将该特征对应的测试作为决策树的根节点。训练数据集根据测试结果进行划分，每一个测试结果都产生一个子数据集，对于每一个子数据集，都将其作为根结点的一个孩子结点。上述过程对于根结点的每一个孩子结点不断重复（递归）：每个孩子结点都是一个子树（subtree）的根结点，其**拥有的数据集是父节点拥有的数据集的一个子集**，同时它在根据信息增益进行特征选择时**不用考虑在它的父节点中已经测试过的特征**。上述递归过程的停止条件是如果子数据集中所有样本都属于同一类，或是子数据集样本数等于0，或是子数据集能够使用的特征集合为空集。

决策树构建方法——ID3算法

2.2 基于信息增益的决策树构建算法——ID3

Algorithm 1 ID3 Algorithm

Input:

set of descriptive features $\boldsymbol{d}$
set of training instances $\mathcal{D}$

Output: decision tree T

if all the instance in $\mathcal{D}$ have the same target level C **then**
    return decision tree T cosisting of single leaf node with label C
end if
if $\boldsymbol{d} = \emptyset$ **then**
    return decision tree T cosisting of single leaf node with label of the the majority target level in $\mathcal{D}$
end if
if $\mathcal{D} = \emptyset$ **then**
    return decision tree T cosisting of single leaf node with the label of the the majority target level of the dataset of the its parent node
end if
$\boldsymbol{d}[best] \leftarrow \arg\max_{d \in \mathcal{D}} \ IG(d, \mathcal{D})$.
make a new node $Node_{d[best]}$ and label it with $\boldsymbol{d}[best]$.
partition dataset $\mathcal{D}$ using $\boldsymbol{d}[best]$.
$\boldsymbol{d} = \boldsymbol{d} - \boldsymbol{d}[best]$
for each partition $\mathcal{D}_i$ of $\mathcal{D}$ **do**
    ID3($\boldsymbol{d}$, $\mathcal{D}_i$)
end for

决策树构建方法——ID3算法

2.3 改进的特征选择指标

信息增益是经典的特征选择指标，但是它也存在缺点：信息增益倾向于选择具有更多可能取值的特征，因为如果一个特征具有很多取值，那么根据这个特征进行数据集划分之后各个子数据集的大小很可能都比较小，而比较小的数据集更可能是纯度高（低熵）的，于是信息增益较大。由于**信息增益偏爱具有更多取值的特征**，后续的研究提出使用信息增益比例（Information Gain Ratio, IGR）来进行特征选择。IGR的表达式为：

$$IGR(d,D)=\frac{IG(d,D)}{-\sum_{l\in level(D)}P(d=l)\times\log_2 P(d=l)}$$

分母是按照特征d对数据集进行分割的熵。根据IGR的表达式，某个特征d如果具有很多取值，那么$IG(d,D)$可能较大，与此同时分母也可能较大，因此IGR可能较小，因此IGR对于IG偏爱多取值特征这一缺陷有一定的修正作用。

主要内容

- 决策树简介（针对本次作业）
- 决策树构建的标准方法——ID3算法
- **决策树剪枝**
- 决策树的模型集成方法——Bagging&Boosting
- 作业提交

决策树剪枝

3.1 剪枝的原因

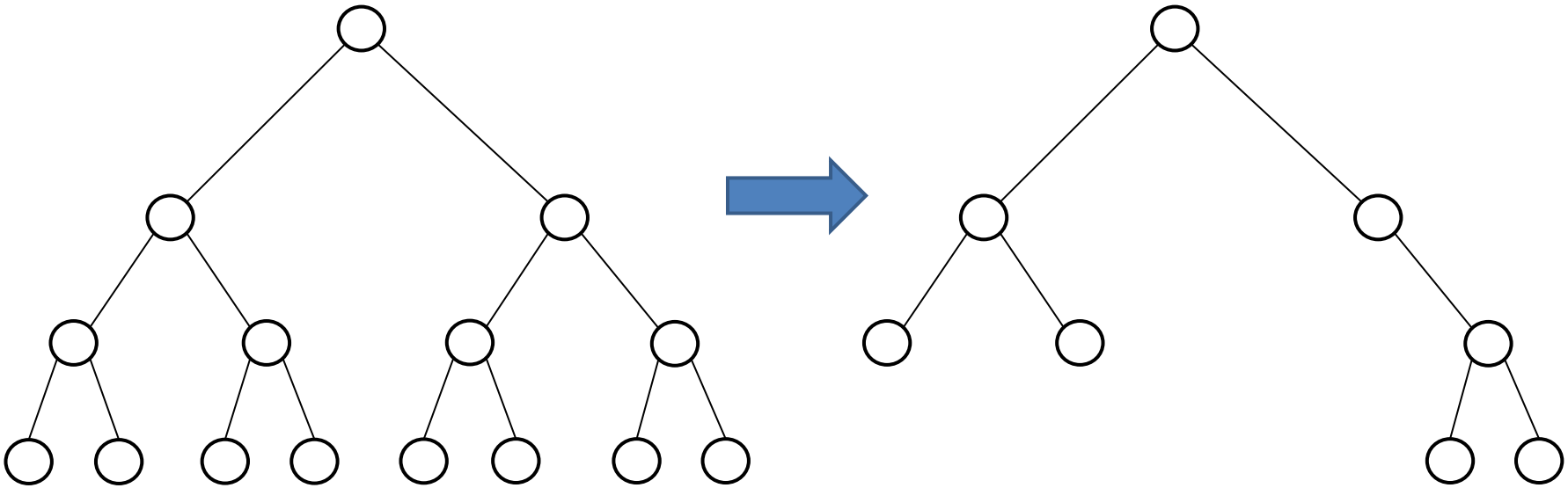
对抗过拟合！

决策树构建算法递归地划分数据集的特性，天然地会导致产生很多只包含一些噪声样本叶子结点。为了解决这个问题，决策树剪枝成为广泛使用地防止决策树模型过拟合的方法。显然，剪枝后的决策树不能完美地拟合训练数据集，但是却**滤除了一些噪声样本**，从而提高了模型的泛化能力。

决策树剪枝

3.2 剪枝的含义

将决策树的一个子树除了其根结点之外全部从决策树中移除，从而其根结点变为剪枝之后的决策树中的一个叶子结点



决策树剪枝

3.3 剪枝的两类基本策略

- 早停止（预剪枝）策略 (Early Stopping)

早停止策略是比较简单的剪枝方式，也被叫做预剪枝 (pre-pruning)。早停止的判断标准可以是最大树深度，每个叶子结点包含的最少样本数目等等。

- 后剪枝策略 (Post Pruning)

后剪枝策略中，首先根据标准决策树构建算法，如上节中的ID3等，构建一棵**未剪枝**的决策树。随后从完整的决策树的叶子结点开始，使用**自底向上** (bottom-up) 的方式进行剪枝，剪枝的过程即删除叶子结点的过程。

在本次大作业中，我们要求使用**基于验证集错误率的判断指标**来决定一个叶子结点是否需要从决策树中被删除。

决策树剪枝

3.3 剪枝的两类基本策略

基于验证集错误率的剪枝方法：

设当前决策树为DT，它的某个叶子结点为L_i，将**验证数据集**在当前决策树DT下进行测试，得到测试错误率R_1。将叶子结点L_i从当前决策树DT中剔除，得到剪枝后的决策树DT_{pruned}。将验证数据集在剪枝后的决策树DT_{pruned}下测试，得到测试错误率R_2，若$R_1 \geq R_2$，则将叶子结点L_i从当前决策树DT中删去。上述过程从决策树最底层的叶子结点开始，使用**层次优先**的遍历方式遍历决策树的各个结点，并在每个结点处都考虑是否将其**对应的子树**从决策树中删除，直到达到根结点为止。

主要内容

- 决策树简介（针对本次作业）
- 决策树构建方法——ID3算法
- 决策树剪枝
- **决策树的模型集成方法——Bagging & Boosting**
- 作业提交

决策树的模型集成方法——Bagging & Boosting

4.1 什么是模型集成 (Ensemble) ?

集成模型通过使用**同一数据集**产生一系列**不同的模型**实现**集体决策**来提升最终模型的泛化能力。集成模型背后的洞见在于一系列专家对于一个问题同时作出决策的质量高于单个专家单独决策的质量

集成模型的两个重要特征:

- 使用相同的数据集构建不同的模型，每个模型在构建时使用不同子数据集；
- 在决策时多个模型同时决策，最终决策结果是对多个模型决策结果的汇总。

决策树的模型集成方法——Bagging & Boosting

4. 2 Boosting Ensemble

4.2.1 加权数据集 (weighted dataset)

原数据集的样本被赋予一个权重，权重可以被视为一种分布，通过从样本权重确定的分布中采样获得一个**带有重复数据的数据集**，每个样本重复的次数正比于它的权重，即权重越大，重复次数越多。

决策树的模型集成方法——Bagging & Boosting

4. 2 Boosting Ensemble

4.2.2 Boosting基本步骤概括

Boosting通过**迭代地向集成模型中添加新模型**来获得最终的集成模型，具体在每一次迭代时，其都需要完成下面4个步骤：

- 利用加权数据集产生一个基模型（base model）——决策树；
- 更新每个训练样本的权重；
- 更新基模型权重；
- 将当前得到的基模型加入到集成模型集合中

决策树的模型集成方法——Bagging & Boosting

4. 2 Boosting Ensemble

4.2.3 Boosting各步详细解析

- 利用加权数据集产生一个基模型（base model）；

在本次大作业中每一个基模型都是决策树模型。在训练得到基模型后，计算基模型在训练数据集上的加权错误概率ϵ:

$$\epsilon = \sum_i w_i I\{f(x_i) \neq y_i\}$$

w_i是训练数据集样本x_i在当前迭代轮次的权重，$f(x_i)$是当前轮的基模型对样本x_i的预测结果，y_i是样本x_i的标签，$I(x)$是指示函数：如果x为真则$I(x) = 1$否则$I(x) = 0$。

决策树的模型集成方法——Bagging & Boosting

4. 2 Boosting Ensemble

4.2.3 Boosting各步详细解析

- 更新每个训练样本的权重；

对于被当前基模型正确分类的样本，权重设置为：

$$w_i \leftarrow w_i \times \frac{1}{2(1-\epsilon)}$$

对于被当前基模型错误分类的样本，权重设置为：

$$w_i \leftarrow w_i \times \frac{1}{2\epsilon}$$

决策树的模型集成方法——Bagging & Boosting

4. 2 Boosting Ensemble

4.2.3 Boosting各步详细解析

- 更新基模型权重：

每个基模型根据它的加权错误概率 ϵ 进行加权，权重计算方式为：

$$\alpha_i = \frac{1}{2}\ln\left(\frac{1-\epsilon}{\epsilon}\right)$$

加权错误概率ϵ越小，基模型权重α_i越大。

上述过程不断迭代直到集成模型中的模型数目达到预设的阈值时停止。最终进行决策时，测试样本被输入到不同的基模型中，**预测结果通过基模型的权重进行加权**，具有最高得分的预测结果作为最终返回的预测结果。

决策树的模型集成方法——Bagging & Boosting

4. 3 Bagging Ensemble

4.3.1 Bagging概念以及与Boosting的区别

- Bagging的方法与Boosting的不同之处在于Bagging使用原数据集的**随机放回抽样子数据集**进行训练，放回抽样决定了各个子数据集中可能会有重复的数据。放回抽样的目的是为了保证每个数据集中都有一些样本是重复的，同时抽样的随机性又保证了每个数据集不可能完全相同，那么在训练的时候每个模型都是在部分不同部分相同的数据集上学习得到的，一方面可以从数据集中相同的数据处学习共性，又可以从不同数据集的不同样本中学习每个数据集的个性。

决策树的模型集成方法——Bagging & Boosting

4. 3 Bagging Ensemble

4.3.2 Bagging的实现方式

训练时：

- 采用放回采样的方式每次从原数据集中采样一个子数据集 (Optional: 每次在采样子数据集时也可以对当前数据集包含的特征进行采样，即一个子数据集可以只包含部分原数据集中的特征)；
- 利用该子数据集，利用决策树算法得到一个基模型；
- 将该子模型放入集成模型集合中；

测试时：

将样本送入各个基模型中，得到不同模型的返回结果，使用**多数表决**的方式选择得票最高的结果作为最终的结果

主要内容

- 决策树简介（针对本次作业）
- 决策树构建方法——ID3算法
- 决策树剪枝
- 决策树的模型集成方法——Bagging & Boosting
- **作业提交**

作业提交

关于代码和实验：

- 本次作业需要提交**一份报告**以及**源代码文档**。使用的**编程语言不限（建议使用Python结合C++，提升程序运行效率的同时保证编程的灵活性）**，要求最终的大作业代码文档中**包含一个README文档**，文档中**详细描述使用方式以及预期的结果**。

- 代码要求能够在其他机器上正常运行，正常运行的含义是指：如果采用Python等脚本类语言，需要在提交的大作业代码文档的README中**详细注明入口脚本以及命令行参数的设置方式**；如果采用C++等静态语言，要求尽可能将所需要的第三方库函数以源代码形式放在最终的代码文档中，并通过CMakeList或MakeFile等方式明确编译方法，**尽量保证在其他机器上不修改CMakeList或MakeFile也能编译出可执行程序**。此外，如果有些同学编写的是C++的动态链接库配合脚本语言，则需要**详细写明动态链接库的编译与安装方法**。

- **报告所构建模型在大于等于3个数据集上的性能。**

作业提交

基础部分

- 实现基于信息增益与信息增益比例的决策树构造算法，比较它们在测试数据集上的准确率区别并试分析之；
- 对基于IG和IGR构造的决策树实现预剪枝与后剪枝策略，比较二者对于最终测试准确率的影响。

B. 附加部分（Optional）

- 实现集成的决策树模型（随机森林，AdaBoost），比较其与单模型的性能差异，分析产生这种差异的原因；
- 决策树算法亦可以拓展到回归问题以及连续值特征上，自己查找资料了解决策树算法如何处理连续值特征以及回归问题，实现这些方法，并且给出测试结果。

作业提交

数据集来源

我们提供了一些可能的公开数据集，这些数据集的特征都是类别（categorical）特征，且问题都是分类问题，可以使用前面介绍的决策树算法解决。

1. UCI乳腺癌数据集，该数据集是一个二分类数据集，包含201个样本，根据9个类别特征判断是否一个样本是no-recurrence-events还是recurrence-events。这9个特征为年龄、更年期、肿瘤尺寸等，具体情况请参考：
http://archive.ics.uci.edu/ml/datasets/Breast+Cancer
2. UCI汽车评估数据集，包含1728个样本，目标是根据汽车的6个类别特征对一辆汽车的价值进行评估，具体情况请参考：
http://archive.ics.uci.edu/ml/datasets/Car+Evaluation
3. UCI Monk问题数据集，Monk问题是最早提出的一个学习问题的benchmark，于1991年提出。该数据集来源于机器人领域，具有432个样本，目标是根据6个类别特征（头类型，身体类型等）完成一个二分类问题，即机器人是否属于某个类型。具体情况请参考
http://archive.ics.uci.edu/ml/datasets/MONK%27s+Problems
4. UCI大豆识别数据集，共有307个样本，19类大豆，35个特征，这些特征都是类别特征。注意该数据集包含缺失数据，如果使用的话需要注明缺失数据的处理方案。
http://archive.ics.uci.edu/ml/datasets/Soybean+%28Large%29

作业提交

注意事项

- 本次作业不允许调用现成的库或者API完成决策树的构造与测试过程，所有的**核心算法(决策树构建算法)必须自己独立完成**，能用的库只能是线性代数类的库、文件读写的库以及结果可视化库。如果发现直接调用现成库函数或是抄袭开源代码者一律按照0分处理。

- 完成附加部分可以获得一定的加分，加分额度视对附加题的完成度决定，最多不超过10分。

- 本次作业鼓励大家报告构造决策时所需要的时间，如果在构造决策树过程中使用了特殊的数据结构或使用了底层优化技术对程序进行了加速，欢迎大家在报告中着重强调，助教会根据报告中对内容核实其报告内容的可信度以及有效性，如果确有速度上的提升，亦可酌情加分。

- 建议同学们在完成决策树的基线算法后使用sklearn这个python库中的决策树模块在同一个数据集下构建一下决策树，看看是不是能够和开源库的性能对上，之后再向下走剩下的部分

作业提交

- **提交途径：**
  - 教学网——教学内容——2020决策树大作业
- **提交内容：**
  - 全部源码文件、文档以及报告
  - 报告请呈现预测准确率等结果以及相应分析
  - 将全部文件打包为zip文件上传
  - zip文件命名为“学号_姓名_决策树大作业”，
  - 如“1801213638_曹志崴_决策树大作业”
- **截止日期：2020年6月5日24点**

参考文献

- [1] 统计学习方法，李航
- [2] Fundamentals of Machine Learning for Predictive Data Analytics, John D. Kolleher, Brain MacNamee, Aoife D'Arcy

QA

Have Fun!