机器学习的数学基础

应用-决策树与随机森林

张朝晖

zhangzhaohui_hbsd@163.com

河北师范大学软件学院

2017.11.08

主要内容

1.决策树

面向非数值特征数据的分类决策—决策树

包括：决策树构建方法；决策树本量的控制；

决策树过学习的克服及推广能力的提高

1.1 非数值特征(*nonmetric features*)

1.2 决策树

1.3 过学习与决策树的剪枝

2.随机森林

样本数据描述

(1)**数值特征** (*metric features*)

(2)**非数值特征**(*nonmetric features*)

如： 名义特征(*nominal features*)

序数特征(*ordinal features*)

区间特征(*interval features*)

非数值特征样本数据分类的处理方式

方式1 非数值特征 —编码→ 数值特征 —基于数值特征数据的分类→

编码可能{造成信息损失；引入人为信息}

方式2 基于非数值特征数据的直接分类-**本节**

主要内容

1.决策树

1.1 非数值特征(*nonmetric features*)

1.2 决策树

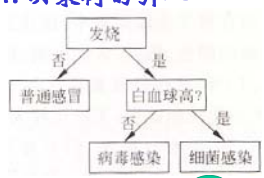
基于规则的决策、多级决策

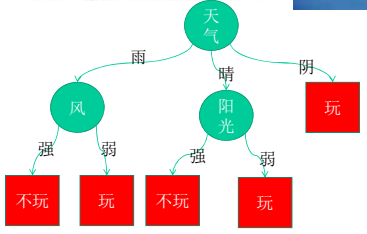
介绍三个著名的决策树构建方法 ID3 C4.5 CART

1.3 过学习与决策树的剪枝

2.随机森林

1. 决策树的引入





决策树是关于if-then规则的集合

规则互斥、完备

1. 决策树的引入

- 应用最广的归纳推理方法之一，模型直观
- 例立的树： 根结点；子结点；叶结点

描述“决策及其相应决策结果的对应关系”—决策规则

非叶结点----决策

根节点代表整个数据集

节点后有分支，每个分支代表一个决策划分

叶结点--------决策结果

分类树，决策结果为类别

回归树，决策结果为实数值

- 决策树的构造：决策规则的生成。

基于规则的推理，基于一定数量训练样本，从数据中学习决策规则，自动构造。

2. 决策树的学习

- **特征选择**
  选取关于训练样本具有更好鉴别能力的特征
- **决策树的生成(模型的局部选择)**
  拟合训练样本
- **决策树的剪枝(prunning)(模型的全局选择)**

  简化模型，使其泛化能力更好

  许多分枝反映的是训练数据中的噪声和孤立点，为避免过学习，应控制树的规模，检测和剪枝

3. 决策树的使用

对未知样本进行分类

决策树是关于样本分类规则的形象化描述

从决策树的根结点开始，将样本的特征值与决策树相比较，一直到叶结点的一条路径，即可完成对样本决策。

4. 有关概念

- **纯节点与不纯节点**

  若某结点的样本集只包含一类样本，则该结点为纯(pure)节点,或者为同质的(homogenous)；否则，为不纯(impure)、或异构(heterogeneous)结点。

- **不纯度(impurity，杂度)**

  关于决策树节点不纯程度的度量.

  如：熵不纯度、Gini不纯度、误差不纯度等。

- **熵不纯度(entropy impurity)**

  若某结点的N个样本来自k类，各类样本出现概率为p_i i=1,…,k，则该结点的熵不纯度

$$I(N) = -\sum_{i=1}^{k} p_i \log_2 p_i \qquad 0\log 0 = 0$$

4. 有关概念—续1

- Gini不纯度(Gini impurity)/基尼指数/方差不纯度

  若某节点N个样本中，来自第j类样本所占比例为P_j, j=1,…,k，则该节点的Gini不纯度

$$I(N) = \sum_{\substack{m,n\in\{1,\ldots,k\} \\ m\neq n}} P_m P_n = 1 - \sum_{j=1}^{k} P_j^2$$

- 误差不纯度

$$I(N) = 1 - \max_{j\in\{1,\ldots,k\}} P_j$$

4. 有关概念-续2

- **信息增益(*Information Gain*)** --绝对增益

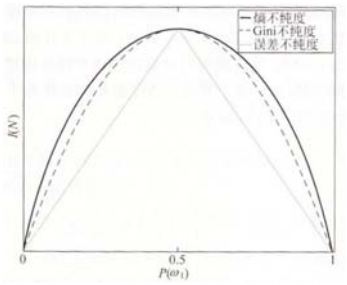
  用于描述决策树中节点分裂后，不纯度的减少量.

  若将某个父节点处N个样本划分成m组，各组样本分属于不同的子节点，相应样本数目N_i, i=1,…,m. 则该父节点做这样的划分，所带来的信息增益为：

$$\Delta I(N) = I(N) - \sum_{i=1}^{m} P_i I(N_i) = I(N) - \sum_{i=1}^{m} \frac{N_i}{N} I(N_i)$$

- **信息增益比率(*Information Gain Ratio*)—相对增益**

$$\Delta I_R(N) = \frac{\Delta I(N)}{I(N)}$$



两类问题情况，三种不纯度度量与某类概率关系

5. 决策树的构建

有许多决策树构建算法，如：

Hunt算法、ID3、C4.5、SLIQ、SPRINT、CART等

决策树构建的关键

- 如何对训练样本集划分(节点的分裂split条件)
  怎样为不同类型的特征指定测试条件?
  怎样评估每种测试条件?
- 如何停止分裂过程(节点分裂的终止条件)

ID3=>C4.5=>C5.0

- **John Ross Quinlan**
  - ID3 1986年
  - C4.5 1993年
  - C5.0 1998年
  - 2011年获得KDD创新奖

- KDD—Conference on Knowledge Discovery and Data mining
- http://www.rulequest.com/Personal/
- http://rulequest.com/download.html
- http://www.rulequest.com/

(1) ID3法(交互式对分法：Interactive Dichotomizer-3

ID3 算法基本思想：

基于奥克姆剃刀准则(Occam 's Razor-- We should always accept the simplest answer that correctly fits our data.)

➔ A good decision tree is the simplest decision tree.

The simplest decision tree that covers all examples should be the least likely to include unnecessary constraints

节点的评价----纯不纯度
新节点的生成，基于目前还没有使用的属性
"最大信息增益"

算法基本点：

- 若当前树节点的某后继节点只含同一类样本，则为**纯节点**，则停止分裂；
- 若当前属性表中**再无可用属性**，则根据**多数表决**确定该节点的类标号，停止分裂；
- 选择最佳分裂的**属性**(**最大信息增益足够大**)，根据所选属性取值(**特征取值数目决定了该节点分裂为后继子节点的数目**)，逐一进行分裂；递归构造决策树。

ID3算法

输入： 训练样本集D, 特征集A, 阈值ε

输出： 决策树T

步骤：

STEP1. 若D中所有样本属于同一类C_k,则T为单结点树，并将C_k作为该结点的类别标记，返回T;

STEP2. 若A为空集，则T为单结点树，并将D中具有最多训练样本的类别C_k作为该结点的类别标记，返回T;

STEP3. 若A不是空集，计算A中各特征 $A_k \in A$对样本集D的信息增益$\{g(D,A_k)\}$，并选择具有**最大信息增益**的特征A_g:
若特征A_g的**信息增益**$g(D,A_g)<\varepsilon$，则执行**3.1**；否则执行**3.2**.

ID3算法(续)

ID3算法只有决策树的生成部分，未涉及裁剪，易产生**过拟合**。

步骤：

STEP3. 若特征A_g的**信息增益**$g(D,A_g)<\varepsilon$，则执行**3.1**；否则执行**3.2**.

3.1 置T为单结点树，将D中具有最多训练样本的类别C_k作为该结点的类别标记，并且返回T;

3.2 对特征A_g的每一可能值a_i，按照$A_g=a_i$，并将D划分为若干非空子集D_i，将D_i中具有最多训练样本的类别作为标记，构建子结点，由结点及其子结点构成树T，返回T;

STEP4. 对第i个子节点，以D_i为训练集，以$A-\{A_g\}$为特征集，**递归调用STEP1-STEP3**得到子树T_i，返回T_i。

4. 决策树的构建-续5

(2)C4.5算法

决策树学习的实际问题:

决策树增长的深度的确定；
连续数值特征的处理；
用于筛选特征的度量指标的确定；
特征不完整的训练数据的处理；
处理不同代价的特征；
计算效率的提高；....

针对上述问题，ID3扩展为C4.5

C4.5 的特别之处： 可处理**连续数值**特征
对于某数值特征x，若包含n个取值，将其取值升序排列；以二分法从n-1中划分方案中选择信息增益率最大的一种划分，将数值特征离散化为二值特征。

Algorithm 1.1 C4.5(D)

Input: an attribute-valued dataset D

```
1:  Tree = {}
2:  if D is "pure" OR other stopping criteria met then
3:     terminate
4:  end if
5:  for all attribute a ∈ D do
6:     Compute information-theoretic criteria if we split on a
7:  end for
8:  a_best = Best attribute according to above computed criteria
9:  Tree = Create a decision node that tests a_best in the root
10: D_v = Induced sub-datasets from D based on a_best
11: for all D_v do
12:    Tree_v = C4.5(D_v)
13:    Attach Tree_v to the corresponding branch of Tree
14: end for
15: return Tree
```

引自：book-2009-The Top Ten Algorithms in Data Mining

C4.5算法

输入： 训练样本集D, 特征集A，阈值ε

输出： 决策树T

步骤：

STEP1. 若D中所有样本属于同一类C_k，则T为单结点树，并将C_k作为该结点的类别标记，返回T；

STEP2. 若A为空集，则T为单结点树，并将D中具有最多训练样本的类别C_k作为该结点的类别标记，返回T；

STEP3. 若A不是空集，计算A中各特征 $A_k \in A$对样本集D的信息增益比$\{g_R(D, A_k)\}$，并选择具有**最大信息增益比**的特征A_g:
若特征A_g的**信息增益比**$g_R(D, A_g) < \varepsilon$，则执行**3.1**；否则执行**3.2.**

C4.5算法(续)

步骤：

STEP3. 若特征A_g的**信息增益比**$g_R(D, A_g) < \varepsilon$，则执行**3.1**；否则执行**3.2.**

3.1 置T为单结点树，将D中具有最多训练样本的类别C_k作为该结点的类别标记，并且返回T；

3.2 对特征A_g的每一可能值a_i，按照$A_g = a_i$，并将D划分为若干非空子集D_i，将D_i中具有最多训练样本的类别作为标记，构建子结点，由结点及其子结点构成树T，返回T；

STEP4. 对第i个子节点，以D_i为训练集，以$A - \{A_g\}$为特征集，**递归调用STEP1-STEP3**得到子树T_i，返回T_i。

4. 决策树的构建-续6

(3)CART(Classification And Regression Tree)

核心思想相同

主要区别

- CART既可用于分类，也可用于对连续变量的回归
- 每个结点只能有两个子结点，决策树为二叉树，不易产生数据碎片，精确度往往也会高于多叉树，所以在CART算法中，采用了二元划分----**递归二叉树**
- 不纯性度量
  - 分类目标：Gini指标
  - 连续目标：**最小平方残差**、最小绝对残差

用**独立的验证集**对训练集生长的树进行**后剪枝**

CART树--最小二乘回归树的生成算法

基本思想：

一个回归树对应输入空间(或特征空间)的一个划分，以及在该划分单元上的输出值。

在训练样本集D所在的输入空间，递归地将每个区域划分为两个子区域，并根据落入每个子区域的训练样本输出值，决定该子区域的输出，构建二叉树。

CART树--最小二乘回归树生成算法

输入： 训练样本集 $D=\{(x_i,y_i),i=1,\ldots,N\}$，$x_i\in R^d$

输出： 回归树 $f(x)$

步骤：

STEP1. 从输入向量 x 中选择最优切分变量 j 以及切分点 s，求解：

$$\min_{j,s}\left[\min_{c_1}\sum_{x_i\in R_1(j,s)}(y_i-c_1)^2+\min_{c_2}\sum_{x_i\in R_2(j,s)}(y_i-c_2)^2\right]$$

遍历输入向量 x 的每个变量 j：对固定的切分变量 j，选择使上述目标函数值最小的 (j,s) 对。

CART树--最小二乘回归树生成算法(续)

步骤：

STEP2. 用上述 (j,s) 对，确定划分区域 $R_1(j,s)$，$R_2(j,s)$，并确定相应输出值。

$$R_1(j,s)=\{x|x^{(j)}\le s\},\ R_2(j,s)=\{x|x^{(j)}>s\}$$

$$\hat{c}_m=\frac{1}{N_m}\sum_{x_i\in R_m(j,s)}y_i,\ x\in R_m,\qquad m=1,2$$

STEP3. 继续对两个子区域调用**STEP1**、**STEP2**，直到满足停止条件。

STEP4. 将输入空间划分为 **M** 个区域：$R_1,\ldots,R_M$；生成决策树。

$$f(x)=\sum_{m=1}^{M}\hat{c}_m I(x\in R_m)$$

CART树--递归二叉分类树的生成算法

基本思想：

一个分类树对应输入空间(或特征空间)的一个划分，以及在该划分单元上的类别输出值。

根据训练样本集 D，从根节点开始，将输入空间进行划分，递归构建二叉分类树。

借助基尼指数进行特征选择，同时决定该特征的最优二值切分点

CART树--递归二叉分类树生成算法

输入： 训练样本集 $D=\{(x_i,y_i),i=1,\ldots,N\}$，

其中：$x_i\in R^d$，$y_i\in\{1,2,\ldots,K\}$

输出： CART决策树

步骤：

STEP1. 设到达当前节点的数据集为 D。遍历输入向量 x **现有变量**的每个特征 A，根据 D 中训练样本关于该变量的所有可能取值，确定所有可能的切分点 s；对于固定的切分变量 j，选择使**基尼指数**最小的切分点。最终得到基尼指数最小的 (j,s) 对。

$D=D_1\bigcup D_2$,

$D_1=\{(x_i,y_i)\in D\mid A(x_i)\le s\},D_2=\{(x_i,y_i)\in D\mid A(x_i)>s\}$

在特征 A 的切分点 s 处，集合 D 的基尼指数：

$$\mathrm{Gini}(D,A_s)=\frac{|D_1|}{|D|}\mathrm{Gini}(D_1)+\frac{|D_2|}{|D|}\mathrm{Gini}(D_2)$$

CART树--递归二叉分类树生成算法(续)

步骤：

STEP1.(续) $\mathrm{Gini}(D_m)=1-\sum_{k=1}^{K}\left(\frac{|C_{mk}|}{|D_m|}\right)^2\quad m=1,2$

C_{mk} 为 D_m 数据集中第 k 类训练样本子集

STEP2. 用上述最优特征及最优切分点 (j,s) 对，确定划分区域 $R_1(j,s)$，$R_2(j,s)$，将当前结点划分为两个子结点，并将训练集 D_1,D_2 按照特征分配到两个子结点中。

STEP3. 继续对两个子结点递归调用**STEP1**、**STEP2**，直到满足停止条件。

STEP4. 将输入空间划分为 **M** 个区域：$R_1,\ldots,R_M$；生成决策树。

主要内容

1.决策树

1.1 非数值特征(*nonmetric features*)

1.2 决策树

基于规则的决策、多级决策

介绍三个著名的决策树构建方法 ID3 C4.5 CART

1.3 过学习与决策树的剪枝

决策树规模的控制：先剪枝、后剪枝

2.随机森林

1.模型的过拟合(*overfitting*)和欠拟合(*underfitting*)

- **分类模型的误差：**
  训练误差，是在训练样本上误分类样本比例
  泛化误差，是模型关于未知样本分类的期望误差
  训练误差越低，模型的**学习能力**越好；
  泛化误差越低，模型的**推广能力**越强

- **好的分类模型**应具有**低训练误差**和**低泛化误差**。

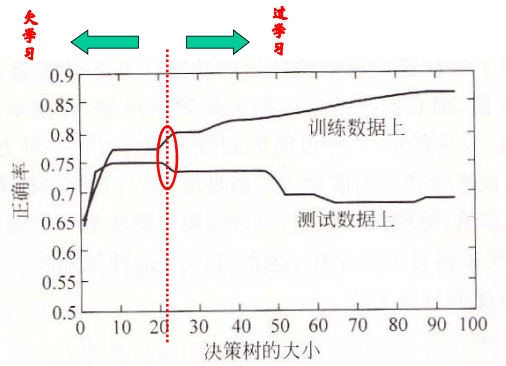
- **具有较低训练误差的模型**，其**泛化误差**可能**高于具有较高训练误差的模型**，这种情况称为**模型过拟合(过学习)**

1.模型的过拟合(*overfitting*)和欠拟合(*underfitting*)

- 决策树规模很小时，训练和检验误差都很大，这种情况为**模型的欠拟合(欠学习)**，原因是**模型尚未学习到数据的真实结构**。

- 随着决策树节点数的增加，模型的训练误差和检验误差都会随之下降。当树的规模变得太大时，即使训练误差还在继续降低，但是检验误差开始增大，导致**模型过拟合(过学习)**,**其原因在于过分关注采样偶然性或噪声等因素影响。**

- 若训练数据**缺乏具有代表性的样本**，并且**样本规模较小**，模型也会产生**过拟合**。



ID3决策树的过拟合现象

2.决策树的剪枝(*pruning*)

目的：控制决策树规模，防止过学习

策略1：先剪枝(*pre-pruning*)

实质—控制决策树的生长

在**完全拟合整个训练集**之前就停止决策树的生长。

生长过程中，某结点停止分枝、成为叶节点**条件：**

- 数据划分法
  不足--不能充分利用样本集信息
- 阈值法：节点记录数目足够小；信息增益足够低
  不足--阈值过高，欠拟合；阈值过低，过拟合
- 基于信息增益的统计显著性分析

策略2：后剪枝(*post-pruning*)

实质：决策树生长后处理，合并分枝

初始阶段--决策树按照最大规模生长

剪枝阶段--自底向上修剪完全增长的决策树

剪枝规则

(1)减少分类错误的修剪法

(2)**最小代价与复杂性的折中**:
平衡“错误率的增加”与“模型复杂程度的降低”

(3)最小描述长度(Minimum Description Length: MDL)
最简单的就是最好的

决策树的剪枝

设决策树T的叶结点数目为$|T|$，叶结点序号$t=1,...,|T|$;训练样本集到达叶结点t的样本数为N_t,其中第k类的样本数为$N_{tk}, k=1,...,K$;

叶结点t的经验熵$H_t(T)$，控制参数$\alpha \geq 0$

$$H_t(T)=-\sum_{k=1}^{K}\frac{N_{tk}}{N_t}\log\frac{N_{tk}}{N_t}$$

决策树T关于训练样本的拟合误差：

$$C(T)=\sum_{t=1}^{|T|}N_tH_t(T)=-\sum_{t=1}^{|T|}N_t\sum_{k=1}^{K}\frac{N_{tk}}{N_t}\log\frac{N_{tk}}{N_t}=-\sum_{t=1}^{|T|}\sum_{k=1}^{K}N_{tk}\log\frac{N_{tk}}{N_t}$$

“模型关于训练样本的拟合误差”+“模型的复杂度”=“模型的损失函数”

$$C_\alpha(T)=C(T)+\alpha|T|$$

决策树的后剪枝，就是在给定α的前提下，选择具有最小$C_\alpha(T)$的子树。

决策树T的后剪枝算法

输入：生成算法产生的整棵树T，参数α

输出：对树T修剪，得到的子树T_α

步骤：

STEP1.计算每个结点(不只是叶结点)的经验熵。

STEP2.递归地从树的叶结点向上回溯。

设**一组叶结点**回溯到其父结点**之前、之后**的整体树分别为T_B和T_A;对应的损失函数值分别为

$$C_\alpha(T_B)=C(T_B)+\alpha|T_B| \quad C_\alpha(T_A)=C(T_A)+\alpha|T_A|$$

若$C_\alpha(T_B)\le C_\alpha(T_A)$，则剪枝，将**叶结点**的父结点作为新的叶结点

STEP3.返回***STEP***2，直到不能继续为止，得到损失函数最小的子树T_α

关于“剪枝”的讨论：

- **后剪枝技术**倾向于产生更好的结果
  根据完全生长的决策树作出的剪枝决策
- **先剪枝**可能过早终止决策树的生长
- “先剪枝”与“后剪枝”结合

主要内容

1.决策树

1.1 非数值特征(*nonmetric features*)

1.2 决策树

1.3 过学习与决策树的剪枝

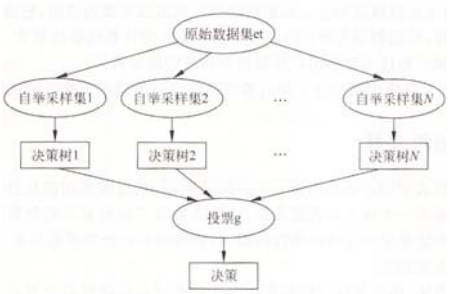
2.随机森林

(*RF: Random Forests, Random Forest Classifiers*)

分类；回归

1. 随机森林(合并分类器)

- 基于样本集构建**多棵决策树**，组成**决策树的“森林”**；
- 多棵决策树**投票决策**



2. 基本步骤

关键：样本采样、特征采样，各树彼此独立；投票无偏。

(1)模型学习--构造N棵决策树(不剪枝)。

每棵树的构建，需要：

A--对样本数据进行“自举法(bootstrapping,或自助法)”重采样，得到1个样本集

有放回地随机抽取

出发点：使用相同特征空间的不同数据点

B—为该样本集，生成备选特征。

从特征集内随机抽取m个特征，形成该决策树学习所需要的特征子集。

(2)模型的使用--决策：输入未知样本，得到多个决策树的输出： 分类--投票，胜者为王；回归—平均，得输出。