

## 实验二 决策树的实现

18308133 刘显彬

October 20, 2021

### 1 伪代码

---

**Algorithm 1** PLApredict( $X, W, b$ )

---

**Input:**  $X$ : 输入数据,  $W$ : 权重,  $b$ : 常数偏置**Output:**  $Y$ : 预测值

---

return  $\text{Sign}(W * X + b)$ 

---

---

**Algorithm 2** PLAttrain( $X, l, W, b, \text{iters}, \eta$ )

---

**Input:**  $X$ : 输入;  $l$ : 标签;  $\text{iters}$ : 迭代次数;  $\eta$ : 学习率**Output:**  $W, b$ : 更新后的  $W, b$  $N \leftarrow \text{len}(X)$  $\text{iter}, i \leftarrow 0$ **while**  $\text{iter} \leq \text{iters}$  **do** $y \leftarrow \text{label}[i]$  $x \leftarrow X[i]$ 

// 找到误分类点

**if**  $\text{predict}(x, W, b) \neq y$  **then** $\text{iter} \leftarrow \text{iter} + 1$ // 对  $W$  进行梯度下降更新 $dW \leftarrow -y * x$  $W \leftarrow W - \eta * dW$ **end if** $i \leftarrow (i + 1) \% N$ **end while**return  $W, b$ 

---

---

**Algorithm 3** LRpredict( $X, W, b$ )

---

**Input:**  $X$ : 输入数据,  $W$ : 权重,  $b$ : 常数偏置

**Output:**  $Y$ : 预测值

$$Y1 \leftarrow W * X + b$$

$$Y \leftarrow \frac{1}{1+e^{-Y1}}$$

return  $Y$

---

---

**Algorithm 4** LRtrain( $X, l, W, b, \text{iters}, \eta$ )

---

**Input:**  $X$ : 输入;  $l$ : 标签;  $\text{iters}$ : 迭代次数;  $\eta$ : 学习率

**Output:**  $W, b$ : 更新后的  $W, b$

$$N \leftarrow \text{len}(X)$$

$$\text{iter}, i \leftarrow 0$$

**for**  $\text{iter} \leftarrow 0$  to  $\text{iters}$  **do**

$$p \leftarrow \text{predict}(X, W, b)$$

//对  $W$  进行梯度下降更新

$$dW \leftarrow \sum_{i=0}^N X_i * (-l_i + p_i)$$

$$dW \leftarrow \frac{dW}{N}$$

$$W \leftarrow W - \eta * dW$$

**end for**

return  $W, b$

---

---

**Algorithm 5** Gini( $d$ )

---

**Input:**  $d$ : dataset

**Output:**  $gini$

// 分成不同的类

$$spData \leftarrow \text{split } d \text{ with different label}$$

$$N \leftarrow \text{len}(d)$$

计算这些类的比重

$$freqs \leftarrow \frac{\text{len}(dset)}{N} \quad \forall dset \in spData$$

$$\text{return } 1 - \sum_{freq \in freqs} freq^2$$

---

---

**Algorithm 6** Gini(*d*, *attr*)

---

**Input:** *d*: dataset, *attr*: split attr**Output:** *gini*

// 像前一个算法一样分裂数据集

*spData*  $\leftarrow$  split *d* with val  $\in$  *attr**N*  $\leftarrow$  len(*d*)*freqs*  $\leftarrow \frac{\text{len}(dset)}{N} \quad \forall dset \in spData$ *gini*  $\leftarrow 0$ **for** *i*  $\leftarrow 0$  to len(*freqs*) **do**    *gini* += *freqs*[*i*] \* Gini(*spData*[*i*]);**end for**return *gini*

---

---

**Algorithm 7** buildTree(*root*, *d*, *alg*)

---

**Input:** *d*: dataset, *root*:决策树根, *alg*:计算信息熵的算法**if** *d.attr* == *null* or only one label in *d.labels* **then**

这是一片叶子

*root*['attr'] = 'leaf', *root*['val'] = vote\_max(*d.labels*)    return *root***else**    *best*  $\leftarrow -inf$ ;    *bestattr*  $\leftarrow ''$ ;

//根据给定的算法找出最优的属性

**for all** *attr*  $\in$  *d.attr* **do**        *best*  $\leftarrow$  *alg*(*d*, *attr*);        *bestattr*  $\leftarrow$  argmax(*best*, *attr*);    **end for**

//然后对最优属性进行分裂, 对子节点进行迭代

*root*['attr']  $\leftarrow$  *bestattr*    *spData*  $\leftarrow$  *d* splitted by *bestattr*;    **for** *sub*  $\in$  *spData* **do**        *root*['val']  $\leftarrow$  buildTree(*root*['val'], *sub*, *alg*);    **end for****end if****Output:** *root*

---

---

**Algorithm 8** predict(*root*, *data*)

---

**Input:** *root*: 决策树树根, *data*: 待预测的数据**Output:** *predVal*: 预测值*cur*  $\leftarrow$  *root*

//当前不是叶子时, 进行搜索

*attr*  $\leftarrow$  *cur*['*attr*']**while** *attr*  $\neq$  '*leaf*' **do**//进入data[*attr*]对应的一支分枝*cur*  $\leftarrow$  *cur*['*val*'] [*data* [*attr*]]*attr*  $\leftarrow$  *cur*['*attr*']**end while**

//返回叶子的预测label

**return** *cur*['*val*']

---