实验二 决策树的实现

18308133 刘显彬

October 13, 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
  reurn Sign(W*X+b)

Algorithm 2 PLAtrain(X, l, W, b, iters, η)

Input: X: 输入 ;l: 标签; $iters$:迭代次数; η:学习率
Output: W, b : 更新后的W,b
  $N \leftarrow len(X)$
  $iter, i \leftarrow 0$
  while $iter \leq iters$ **do**
    $y \leftarrow label[i]$
    $x \leftarrow X[i]$
    //找到误分类点
    if predict(x,W,b) != y **then**
      $iter \leftarrow iter + 1$
      //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, iters, η)

Input: X: 输入 ;l: 标签; $iters$:迭代次数; η:学习率
Output: W, b : 更新后的W,b

$N \leftarrow len(X)$
$iter, i \leftarrow 0$
for $iter \leftarrow 0$ to iters **do**
  $p \leftarrow$ 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
$spData \leftarrow$ split d with different label
$N \leftarrow len(d)$
计算这些类的比重
$freqs \leftarrow \frac{len(dset)}{N}\ \forall dset \in spData$

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 ← split d with val ∈ attr
N ← len(d)
freqs ← len(dset)/N  ∀dset ∈ spData

gini ← 0
for i ← 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 ← −inf;
    bestattr ←'';
    //根据给定的算法找出最优的属性
    for all attr ∈ d.attr do
        best ← alg(d, attr);
        bestattr ← 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'] ← bestattr
    spData ← d splitted by bestattr;
    for sub ∈ spData do
        root['val'] ← 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 ← root
//当前不是叶子时，进行搜索
attr ← cur['attr']
while attr != 'leaf' do
    //进入data[attr]对应的一支分枝
    cur ← cur['val'][data[attr]]
    attr ← cur['attr']
end while
//返回叶子的预测label
return cur['val']
```