Machine Learning

Indah Agustien Siradjuddin

Classification and Regression Trees - CART

Semester Gasal 2019-2020

CART dapat digunakan untuk permasalahan klasifikasi ataupun regresi berdasarkan beberapa level *decision tree*.
Decision Tree :

- Node : fitur / atribut
- link/branch/edge : *path*
- leaf : output

Kelebihan :

1. Proses pengambilan keputusan sama dengan cara berfikir logis
2. Decision Tree dapat divisualisasikan (bukan *a black box*)

Algoritma

CART adalah *binary tree* yang dibangun berdasarkan nilai **Gini Impurity** (Gini Index). Tahapan pembentukan binary tree :

1. Siapkan dataset (fitur dan targetnya)
2. Tentukan *best split* berdasarkan nilai Gini Impurity dan Gini Split
3. Bagi data menjadi dua bagian (left dan right)
4. Ulangi langkah 2-3 sampai *stopping criteria* terpenuhi (gini impurity = 0, atau kedalaman binary tree)

Gini Impurity

$$Gini_t = 1 - \sum_{k=1}^{n} (P_{t,k})^2$$

dimana:

- G_i : Gini Impurity dari sebuah node t,
- n : jumlah kelas,
- $P_{t,k}$: rasio kelas-k dari seluruh dataset pada suatu node -t

Gini Split

$$Gini(s,t) = Gini(t) - P_L Gini(t_L) - P_R Gini(t_R)$$

dimana:

- $Gini(s,t)$ = Gini Split dari split-s dan node-t
- $Gini(t)$ = Gini impurity atau Gini Index dari sebuah node-t
- $Gini(t_L)$ = Gini impurity atau Gini Index dari *left child node* setelah proses spliting pada split s
- P_L = rasio sample data (pada *left node*) atau rasio jumlah data pada left node berdasarkan seluruh data
- $Gini(t_R)$ = Gini impurity atau Gini Index dari *right child node* setelah proses spliting pada split s
- P_R=rasio sample data (pada *right node*) atau rasio jumlah data pada left node berdasarkan seluruh data

Cut Value

Data pelatihan dibagi menjadi dua grup (*left* dan *right*) berdasarkan *best split* dan *cut value*. Nilai *cut value* ini didapatkan dengan :

1. Uruthkan data berdasarkan data fitur yang berbeda
2. Cari median tiap dua data yang berurutan

Data

- Iris,
- feature : sepal length (1 feature)
- number of data : 6 (@2), yaitu :

Fitur	Kelas
5.1	0
4.9	0
7	1
6.4	1
6.3	2
5.8	2

Training - Pembentukan binary tree

1. Hitung cut value

Fitur	Kelas	cut value
4.9	0	
5.1	0	5
5.8	2	5.45
6.3	2	6.05
6.4	1	6.35
7	1	6.7

2. Gini Index/Impurity untuk masing-masing grup (*left* atau *right*)

Untuk pertama kali, hitung gini index/impurity dataset :

kelas	Jumlah Data	$P_{t,k}$
0	2	$2/6$
1	2	$2/6$
2	2	$2/6$
Total	**6**	

Gini Impurity

$$Gini_t = 1 - \sum_{k=1}^{n} (P_{t,k})^2$$

$$\begin{aligned} Gini_1 &= 1 - \sum_{k=0}^{2} (P_{1,k})^2 \\ &= 1 - (P_{1,0}^2 + P_{1,1}^2 + P_{1,2}^2) \\ &= 1 - (0.333^2 + 0.333^2 + 0.333^2) \\ &= 0.6667 \end{aligned}$$

3. Tentukan Best Split

Hitung Gini split masing-masing *cut value*, cut value akan digunakan sebagai node, jika mempunyai nilai gini split yang paling besar.

split = 5

Fitur	kelas	Grup
4.9	0	left
5.1	0	right
5.8	2	right
6.3	2	right
6.4	1	right
6.7	1	right

Grup - Left

kelas	Jumlah Data	$P_{1,k}$
0	1	$1/1$
1	0	$0/1$
2	0	$0/1$
Total	**1**	

Gini Index left

$$Gini_t = 1 - \sum_{k=1}^{n} (P_{t,k})^2$$

$$\begin{aligned} Gini_{1L} &= 1 - \sum_{k=0}^{2} (P_{1,k})^2 \\ &= 1 - (P_{1,0}^2 + P_{1,1}^2 + P_{1,2}^2) \\ &= 1 - (0.3331^2 + 0^2 + 0^2) \\ &= 0 \end{aligned}$$

Grup - Right

kelas	Jumlah Data	$P_{1,k}$
0	1	$1/5$
1	2	$2/5$
2	2	$2/5$
Total	**5**	

Gini Index right

$$Gini_t = 1 - \sum_{k=1}^{n} (P_{t,k})^2$$

$$\begin{aligned} Gini_{1R} &= 1 - \sum_{k=0}^{2} (P_{1,k})^2 \\ &= 1 - (P_{1,0}^2 + P_{1,1}^2 + P_{1,2}^2) \\ &= 1 - (0.2^2 + 0.4^2 + 0.2^2) \\ &= 0.64 \end{aligned}$$

Gini Split

$$\begin{aligned} Gini(s,t) &= Gini(t) - P_L Gini(t_L) - P_R Gini(t_R) \\ Gini(5,1) &= Gini_1 - P_L Gini_{1L} - P_R Gini_{1R} \\ &= 0.667 - \frac{1}{6} * 0 - \frac{5}{6} * 0.64 \\ &= 0.133 \end{aligned}$$

Lakukan tiga langkah diatas sampai menemukan nilai gini split terbesar.

CART with scikit

In [2]:

```
from sklearn import datasets
from sklearn. tree import DecisionTreeClassifier
import numpy as np
```

In [16]:

```
dataTraining=datasets.load_iris()
temp1=dataTraining.data[0:2,:]
temp2=dataTraining.data[50:52,:]
temp3=dataTraining.data[100:102,:]
feature=np.concatenate((temp1, temp2,temp3), axis=0)
#print('feature=',feature)
target=np.array([0,0,1,1,2,2])
#target=dataTraining.target
feat=feature[:,0]
print(feat)
print(target)
```

```
[5.1 4.9 7.  6.4 6.3 5.8]
[0 0 1 1 2 2]
```

In [20]:

```
a=np.reshape(feat, (-1, 1))
```

Training

In [22]:

```
treeClf=DecisionTreeClassifier(max_depth=2)
treeClf.fit(a,target)
```

Out[22]:

```
DecisionTreeClassifier(class_weight=None, criterion='gini', max_depth=2,
                       max_features=None, max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, presort=False,
                       random_state=None, splitter='best')
```

CART visualization

In [25]:

```
from sklearn. tree import export_graphviz
export_graphviz(
treeClf,
out_file="iris_tree. dot",
feature_names=dataTraining.feature_names[0:1] ,
class_names=dataTraining.target_names,
rounded=True,
filled=True
)
```

open http://webgraphviz.com/ (http://webgraphviz.com/) , and copy file tree

CART for the dataset

Testing

In [ ]:

```
data=np.array([[5.4,3.7, 1.5, 0.2]])
print(data)
output=treeClf.predict(dataTraining.data)
print(output)
```

CART with your own Code

In [7]:

```
import cart
import numpy as np

# Main
bestCut=cart.calculateCutVal(feat)
print('best cut=',bestCut)
#Gini awal, dengan anggapan split adalah 1000, oleh karena itu data berada di leftgrup
l,r=cart.splitData(feat,target,1000)
initGini=cart.calculateGiniImpurity(l)
giniSplit=np.zeros_like(bestCut)
for i in range(len(bestCut)):
    giniSplit[i]=cart.calculateGiniSplit(initGini,bestCut[i],feat,target)
print('gini split=',giniSplit)
bestSplit=bestCut[np.argmax(giniSplit)]
print(bestSplit)
```

```
best cut= [5.   5.45 6.05 6.35 6.7 ]
gini split= [0.13333333 0.33333333 0.22222222 0.33333333 0.13333333]
5.449999999999999
```

In [9]:

```
# depth =2
l,r=cart.splitData(feat,target,bestSplit)
print(l,r)
giniL=cart.calculateGiniImpurity(l)
giniR=cart.calculateGiniImpurity(r)
print(giniL,giniR)
```

```
[2. 0. 0.] [0. 2. 2.]
0.0 0.5
```

In [ ]: