IF3270 Pembelajaran Mesin

Implementasi Decision Tree Learning

Oleh:

13517026 - Ahmad Mutawalli (K2)
13517033 - Harry Rahmadi Munly (K3)
13517062 - Ardysatrio Fakhri Haroen (K2)
13517077- Dandi Agus Maulana (K2)

PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG

2020

1. **Penjelasan Implementasi**
   a. **Proses pembentukan tree**

      Tree dibuat menggunakan metode rekursif dan bentuknya merupakan linked list dari objek - objek node, dengan node adalah atribut dari data yang diberikan.

      Langkah - langkah rekursif yang dilakukan yaitu :

      1. Apabila data sudah terbagi menjadi sempurna, maka buat *leaf* dengan nilai *leaf* adalah kelas dari data yang tersisa.
      2. Apabila tidak ada atribut yang tersisa untuk direkursif, maka buat *leaf* dengan nilai *leaf* adalah modus kelas dari data yang tersisa.
      3. Lakukan rekursif berikut ini hingga tercapai kondisi pertama atau kedua
         a. Cek apakah value dari atribut adalah diskrit atau continues.
            - Jika bertipe diskrit, maka langsung lakukan perhitungan info gain
            - Jika bertipe continues, lakukan sorting data berdasarkan kelas target, cari lokasi split, dan hitung info gainnya.
         b. Pilih atribut berdasarkan info gain yang terbesar.
         c. Cek apabila terdapat missing value
            - Jika terdapat missing value, lakukan impute missing value dengan menggunakan modus dari data atribut tersebut
         d. Tambahkan child node dari node saat ini dari split values yang ada.

   b. **Overfitting training data dengan post prunning**

      Selain training data dibutuhkan data validasi. Pruning dilakukan dari bawah ke atas dan kiri ke kanan. Pertama kali dilakukan pruning pada subtree terkiri dan terbawah menjadi daun dengan nilai daun ditentukan most common value target attribut training data. Bandingkan akurasi pohon awal dengan akurasi pohon post pruning terhadap data validasi. Jika akurasi pohon post pruning lebih tinggi, maka lakukan pruning dan lanjutkan ke subtree atas. jika tidak lanjutkan ke subtree sebelah kanan tanpa pruning.

   c. **Overfitting training data dengan rule post prunning**

      Pertama-tama dilakukan training pada pohon menggunakan training data, memperbolehkan pohon untuk terjadi overfitting. Setelah itu, dibentuk representasi rule berdasarkan pohon yang telah dibuat. Rule merepresentasikan setiap path dari akar menuju daun pada pohon. Untuk setiap rule, kita lakukan pruning pada rule tersebut. Pruning dilakukan dengan cara mencoba menghilangkan masing-masing prekondisi secara inkremental (Pertama dihilangkan satu prekondisi, kemudian dua, dst). Penghapusan prekondisi secara inkremental tersebut dihentikan jika pada suatu saat, penghapusan n buah prekondisi tidak memperbaiki akurasi dibandingkan penghapusan n-1 buah prekondisi, atau tidak ada prekondisi untuk dihapus lagi. Rule yang telah di-pruning tersebut disimpan beserta dengan akurasinya terhadap

validation set. Kemudian, rule diurutkan berdasarkan akurasi yang paling besar terhadap *validation set.*

d. Continues value attribute

Attribute yang valuenya bertipe continues dapat diatasi dengan cara mencari dimana lokasi - lokasi split value yang optimum. Caranya yaitu pertama-tama dilakukan sorting data kelas target agar dapat melihat berbagai lokasi split-nya. Kemudian dari berbagai lokasi split itu, ditentukan lokasi split paling optimum dengan menghitung information gain pada masing-masing split (dengan membagi data menjadi <= split dan > split). Pada akhirnya, atribut kontinu tersebut akan dibagi berdasarkan split paling optimum yang dihitung sebelumnya

e. Alternative measures for selecting attribute

Alternatif lain untuk pemilihan atribute dapat dilakukan dengan menggunakan gain ratio. Gain ratio dari attribute A diperoleh dari info gain attribute A dibagi dengan split information atribut A (split information adalah kalkulasi entropi pada suatu variabel tertentu).

f. Handling missing attribute value

Handling missing attribute value dilakukan dengan cara mengganti missing value dengan most common value dari atribut yang bersangkutan tersebut.

g. Predict new data

Untuk melakukan prediksi dari data yang baru masuk, pertama - tama dilakukan pembentukan tree terlebih dahulu dari training data. Setelah *tree* terbentuk, hasil prediksi didapatkan dengan menjelajahi *tree* dari akar hingga daun sesuai dengan kondisi yang dipenuhi oleh instance yang bersangkutan.

2. Hasil Eksekusi

a. myID3

Hasil pembentukan tree

```
-------tree-------
outlook
|--(Sunny)-->humidity
|            |--(High)-->{class : No}
|            |--(Normal)-->{class : Yes}
|--(Overcast)-->{class : Yes}
|--(Rain)-->wind
|            |--(Weak)-->{class : Yes}
|            |--(Strong)-->{class : No}
```

Training data

	outlook	temp	humidity	wind	play
0	Sunny	Hot	High	Weak	No
1	Sunny	Hot	High	Strong	No
2	Overcast	Hot	High	Weak	Yes
3	Rain	Mild	High	Weak	Yes
4	Rain	Cool	Normal	Weak	Yes
5	Rain	Cool	Normal	Strong	No
6	Overcast	Cool	Normal	Strong	Yes
7	Sunny	Mild	High	Weak	No
8	Sunny	Cool	Normal	Weak	Yes
9	Rain	Mild	Normal	Weak	Yes
10	Sunny	Mild	Normal	Strong	Yes
11	Overcast	Mild	High	Strong	Yes
12	Overcast	Hot	Normal	Weak	Yes
13	Rain	Mild	High	Strong	No

Data validasi untuk prediksi

	outlook	temp	humidity	wind	play
10	Sunny	Mild	Normal	Strong	Yes
11	Overcast	Mild	High	Strong	Yes
12	Overcast	Hot	Normal	Weak	Yes
13	Rain	Mild	High	Strong	No

Hasil prediksi dengan menggunakan modul myID3

```
print(tree.predict(data_X.tail(4)))
```

```
-------predict-------
['Yes', 'Yes', 'Yes', 'No']
```

b. myC45

Hasil pembentukan tree

```
-------tree-------
petal_length
|--(<=2.45)-->{class : 0}
|--(>2.45)-->petal_width
|            |--(<=1.7)-->sepal_length
|            |            |--(<=7.1)-->sepal_width
|            |            |            |--(<=2.8)-->{class : 1}
|            |            |            |--(>2.8)-->{class : 1}
|            |            |--(>7.1)-->{class : 2}
|            |--(>1.7)-->sepal_length
|            |            |--(<=5.9)-->sepal_width
|            |            |            |--(<=3.1)-->{class : 2}
|            |            |            |--(>3.1)-->{class : 1}
|            |            |--(>5.9)-->{class : 2}
```

Training data untuk prediksi

	sepal_length	sepal_width	petal_length	petal_width	label
0	5.1	3.5	1.4	0.2	0
1	4.9	3.0	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0
3	4.6	3.1	1.5	0.2	0
4	5.0	3.6	1.4	0.2	0
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	2
146	6.3	2.5	5.0	1.9	2
147	6.5	3.0	5.2	2.0	2
148	6.2	3.4	5.4	2.3	2
149	5.9	3.0	5.1	1.8	2

Data validasi untuk prediksi

	sepal_length	sepal_width	petal_length	petal_width	label
140	6.7	3.1	5.6	2.4	2
141	6.9	3.1	5.1	2.3	2
142	5.8	2.7	5.1	1.9	2
143	6.8	3.2	5.9	2.3	2
144	6.7	3.3	5.7	2.5	2
145	6.7	3.0	5.2	2.3	2
146	6.3	2.5	5.0	1.9	2
147	6.5	3.0	5.2	2.0	2
148	6.2	3.4	5.4	2.3	2
149	5.9	3.0	5.1	1.8	2

Hasil prediksi menggunakan modul myC45

```
test_data = iris_data.tail(10)
tree_iris.predict(test_data)
# tree_iris.accuracy_tree(test_data)
```

```
-------predict-------

[2, 2, 2, 2, 2, 2, 2, 2, 2, 2]
```

3. **Perbandingan dengan DTL sklearn dan id3Estimator**
   a. **myID3 dengan id3Estimator**

Hasil myID3

```
-------tree-------
outlook
|--(Sunny)-->humidity
|            |--(High)-->{class : No}
|            |--(Normal)-->{class : Yes}
|--(Overcast)-->{class : Yes}
|--(Rain)-->wind
|            |--(Weak)-->{class : Yes}
|            |--(Strong)-->{class : No}
```

Hasil id3Estimator

```
outlook <=0.50: 1 (4)
outlook >0.50
|   humidity <=0.50
|   |   temp <=1.50: 0 (2)
|   |   temp >1.50
|   |   |   wind <=0.50: 0 (1)
|   |   |   wind >0.50: 0 (1/1)
|   humidity >0.50
|   |   wind <=0.50
|   |   |   temp <=1.00: 0 (1)
|   |   |   temp >1.00: 1 (1)
|   |   wind >0.50: 1 (3)
```

Analisis :

Algoritma ID3 yang kami buat lebih sederhana daripada id3Estimator, lebih banyak hal yang dipertimbangkan di id3Estimator. id3Estimator menggunakan label encoding yang mengubah label menjadi ke binary encoding. Sedangkan, algoritma ID3 tidak menggunakan label encoding.

b. myC45 dengan DTL sklearn

Hasil myCH45

```
-------tree-------
petal_length
|--(<=2.45)-->{class : 0}
|--(>2.45)-->petal_width
|            |--(<=1.7)-->sepal_length
|            |            |--(<=7.1)-->sepal_width
|            |            |            |--(<=2.8)-->{class : 1}
|            |            |            |--(>2.8)-->{class : 1}
|            |            |--(>7.1)-->{class : 2}
|            |--(>1.7)-->sepal_length
|            |            |--(<=5.9)-->sepal_width
|            |            |            |--(<=3.1)-->{class : 2}
|            |            |            |--(>3.1)-->{class : 1}
|            |            |--(>5.9)-->{class : 2}
```

Hasil DTL sklearn

```
|--- petal_length <= 2.45
|   |--- class: 0
|--- petal_length >  2.45
|   |--- petal_width <= 1.75
|   |   |--- petal_length <= 4.95
|   |   |   |--- petal_width <= 1.65
|   |   |   |   |--- class: 1
|   |   |   |--- petal_width >  1.65
|   |   |   |   |--- class: 2
|   |   |--- petal_length >  4.95
|   |   |   |--- petal_width <= 1.55
|   |   |   |   |--- class: 2
|   |   |   |--- petal_width >  1.55
|   |   |   |   |--- sepal_length <= 6.95
|   |   |   |   |   |--- class: 1
|   |   |   |   |--- sepal_length >  6.95
|   |   |   |   |   |--- class: 2
|   |--- petal_width >  1.75
|   |   |--- petal_length <= 4.85
|   |   |   |--- sepal_length <= 5.95
|   |   |   |   |--- class: 1
|   |   |   |--- sepal_length >  5.95
|   |   |   |   |--- class: 2
|   |   |--- petal_length >  4.85
|   |   |   |--- class: 2
```

Analisis

Perhitungan statistik yang kami gunakan menggunakan information gain dan gain ratio. Sedangkan, pada algoritma sklearn menggunakan gain dan gini. Algoritma Sklearn tidak hanya digunakan untuk klasifikasi tetapi juga digunakan untuk regresi, klustering, preprocessing, dan lainnya.Algoritma klasifikasi yang digunakan sklearn adalah CART yang dioptimisasi.Sklearn tidak menyupport *categorical variable*.Sedangkan C4.5 bisa diterapkan pada *categorical variable*. CART mirip dengan C4.5 tetapi tidak mengkomputasi rule set.

4. Pembagian tugas

NIM	Tugas
13517026	rule_post_pruning,laporan,post_pruning
13517033	post_pruning,laporan,rule_post_pruning
13517062	Rule_post_pruning,tree,node,prediction, laporan
13517077	laporan,tree,node