TECHNICKÁ UNIVERZITA V KOŠICIACH
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

ID3: Generovanie rozhodovacích stromov

Dokumentácia

2020 **Stanislav Mačák**

Obsah

Úvod

ID3 (Iteratívny Dichotomiser 3) je algoritmus, ktorý sa používa na generovanie rozhodovacieho stromu z dátovej sady. Vynašiel ho Ross Quinlan v roku 1986. ID3 je predchodcom algoritmu C4.5 a zvyčajne sa používa v doménach strojového učenia

Súhrn algoritmu

1. Vypočítajte entropiu každého atribútu dátovej sady
2. Rozdeľte dátovú sadu na podmnožiny pomocou atribútu, u ktorého je výsledná entropia po rozdelení minimalizovaná, alebo ekvivalentný zisk informácií je maximálny
3. Vytvorte uzol rozhodovacieho stromu obsahujúceho tento atribút.
4. Opakujte u podmnožín pomocou zostávajúcich atribútov

Prehľad vzorcov potrebných ku algoritmu

Entropia priestoru:

$$H(S) = \sum_{x \in X} -p(x) * log_2 p(x) \quad (1)$$

Entropia atribútu:

$$H(S, a) = \sum_{i} p(a_i) H(a_i) \quad (2)$$

Informačný zisk:

$$IG(S, a) = H(S) - H(S, a) \quad (3)$$

Vlastná implementácia algoritmu

Pri tvorbe algortimu boli použité knižnice:

- `import csv` – na načítanie dát
- `import os` – na načítanie dát
- `import math` – na matematické operácie
- `import numpy` – na matematické operácie

Entropia priestoru

```
def find_entropy(data)
```

Popis funkcie:

- Funkcia vypočíta entropiu priestoru pomocou vzorca 1.

Vstup:

- **data** – dátova množina načítaná pomocou funkcie **load_csv**(filename).

Výstup:

- Hodnota entropie priestoru.

Entropia atribútu

```
def find_entropy_attribute(df,attribute)
```

Popis funkcie:

- Funkcia vypočíta entropiu atribútu pomocou vzorca 2.

Vstup:

- **Df** – dátova množina načítaná pomocou funkcie **load_csv**(filename)
- **Attribute** – Atribút, ktorého entropiu chcem vypočítať, v pododobe Stringu

Výstup:

- Hodnota entropie atribútu.

Informačný zisk

```
def find_winner(df)
```

Popis funkcie:

- funkcia vypočíta Informačný zisk pre jednotlivé atribúty podľa vzorca a nájde z pomedzi nich jedného s najvyšší informačným ziskom.

Vstup:

- **Df** – dátova množina načítaná pomocou funkcie **load_csv**(filename).

Výstup:

- Atribút v podobe stringu.

Stavba stromu

```
def fit(df,tree=None)
```

Popis funkcie:

- Funkcia slúži na postavenie rozhodovacieho stromu.

Vstup:

- **Df** – dátova množina načítaná pomocou funkcie **load_csv**(filename)
- **Tree** – rozhodovancí strom, jeho hodnota je na začiatku algorihmu None. Funkcia volá samú seba rekurzívne a takto skladá strom.

Výstup:

- **Tree** – rozhodovací strom v dataformáte Dictonary

Predikcia

```
def predict(inst,tree)
```

Popis funkcie:

- Funkcia slúži na predikovanie pomocou vytvoreného rozhodovacieho stromu.

Vstup:

- **Inst** – príklad, ktorý chcem predikovať
- **Tree** – strom vygenerovaný pomocou funkcie **fit**(df,tree=None)

Výstup:

- Predikcia rozhodovacieho stromu v podobe stringu

Práca s dátami

```
def load_csv(filename)
```

Popis funkcie:

- Funkcia slúži na načítaní tabuľky zo súboru .csv. Dáta musia byť usporiadané tak, že cieľový atribút s triedamy sa nachádza v poslednom stĺpci tabuľky

Vstup:

- **Filename** – cesta ku súboru

Výstup:

- Dáta v dataformáte Dictonary

Pomocné funkcie

def getUnique(arr)

Popis funkcie:

- Funkcia vracia jedinečné hodnoty datátového formátu list

def getValueCounts(arr)

Popis funkcie:

- Funkcia vracia hodnotu akú časť z celku tvorí daná trieda

def log2(num)

Popis funkcie:

- Funkcia vracia logaritmus pri základe 2 z čisla num. Ak sa num rovná 0 tak vráti 0.

def getArgmax(arr)

Popis funkcie:

- Funkcia vracia index naväčšej hodnoty v poly arr.

def getSubTable(data, atrb,value):

Popis funkcie:

- Funkcia vracia ‘‘sub-tabuľku“ z dátovej množiny **data**

Vstup:

- **data** – dátova množina načítaná pomocou funkcie **load_csv**(filename)
- **atrb** – meno atribútu, z ktorého hodnôt chceme robiť sub-tabuľku
- **value** – hodnota atribútu

Výstup:

- Sub-tabuľka vo dátovom formáte Dictonary

Popis funkcie fit

```
def fit(df,tree=None):
    Class = list(data.keys())[-1] #Triedy
    node = find_winner(df) #hľadanie najlepšieho atribútu Podľa IG
    attValue = getUnique(df[node]) #jedinečné hodnoty daného atribútu

    #vytvorenie stromu na začiatku procesu
    if tree is None:
        tree={}
        tree[node] = {}

    # vytvorenie cyklu na konštrukciu stromu volaním funkcie rekurzívne
    for value in attValue: #Iterácie cez jedničné hodnoty atribútu
        #Načítanie sub-tabuľky danej jedinčnej hodnoty daného atribútu
        subtable = getSubTable(df,df[node],value)

        #jedinečné hodnoty cieľového atribútu subtabuľky
        clValue,counts = np.unique(subtable[Class],return_counts=True)

        if len(counts)==1: #ak subset obsahuje iba jednu triedu tak končíme

            tree[node][value] = clValue[0]
        else:
           # Ak neobsahuje jednu triedu tak volam funkciu rekurzívne
            tree[node][value] = fit(subtable)
    return tree
```

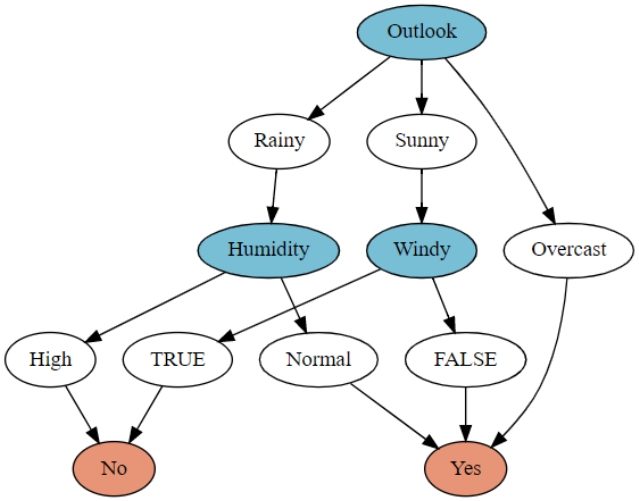
Ukážka využitia algoritmu

```
filename = "/content/drive/MyDrive/ML/golf-dataset.csv"
data = load_csv(filename)
```

	Humidity	Outlook	Play Golf	Temp	Windy
0	High	Rainy	No	Hot	FALSE
1	High	Rainy	No	Hot	TRUE
2	High	Overcast	Yes	Hot	FALSE
3	High	Sunny	Yes	Mild	FALSE
4	Normal	Sunny	Yes	Cool	FALSE
5	Normal	Sunny	No	Cool	TRUE
6	Normal	Overcast	Yes	Cool	TRUE
7	High	Rainy	No	Mild	FALSE
8	Normal	Rainy	Yes	Cool	FALSE
9	Normal	Sunny	Yes	Mild	FALSE
10	Normal	Rainy	Yes	Mild	TRUE
11	High	Overcast	Yes	Mild	TRUE
12	Normal	Overcast	Yes	Hot	FALSE
13	High	Sunny	No	Mild	TRUE

```
tree = fit(data)
printTreePretty(tree,data)
```

```
Target atribute classes: ['Yes', 'No']
Atributes : ['Outlook', 'Temp', 'Humidity', 'Windy']
```




```
test = \
{"Outlook"  : ["Sunny"],
 "Temp"     : ["Hot"],
 "Humidity" : ["Normal"],
 "Windy"    : ["TRUE"],
 "Play Golf": ["No"]
 }
print(test)
```

```
{'Outlook': ['Sunny'], 'Temp': ['Hot'], 'Humidity': ['Normal'], 'Windy': ['TRUE'], 'Play Golf': ['No']}
```

```
print(predict(test,tree))
```

```
No
```

Zdroje

[1]. Quinlan, J. R. 1986. Induction of Decision Trees. Mach. Learn. 1, 1 (Mar. 1986)

Prílohy

Implementácia je dostupná na odkaze:

https://colab.research.google.com/drive/1PSsnUOuEBj1kMexYFGWU41S6b8gb5h3r?usp=sharing