Dokumentacja wstepna projektu z UMA

Piotr Jabłoński (325163) i Paweł Wysocki (325248)

Grudzień 2024

Contents

1 Temat projektu **2**

2 Opis problemu **2**
2.1 Drzewo klasyfikacyjne . . . 2
2.1.1 Ogólna zasada działania . . . 2
2.1.2 Entropia . . . 3
2.1.3 Kryteria podziału . . . 3
2.2 Selekcja ruletkowa . . . 4
2.3 Algorytmy . . . 4
2.3.1 Algorytm budowania drzewa . . . 4
2.3.2 Algorytm wyboru testu z ruletką . . . 5
2.3.3 Algorytm obliczania zysku informacji (Information Gain) . . . 5

3 Plan eksperymentów **5**

4 Zbiory danych **5**
4.1 Red Wine Quality . . . 5
4.2 Loan Approval Classification . . . 6
4.3 Nursery . . . 7
4.4 Mobile Device Usage and User Behavior . . . 7

1 Temat projektu

Celem naszego projektu jest implementacja algorytmu konstruującego drzewo klasyfikujące, w którym test dla danego węzła jest wybierany za pomocą ruletki, tj. każdy test ma szansę na wybór proporcjonalną do swojej jakości.

2 Opis problemu

2.1 Drzewo klasyfikacyjne

2.1.1 Ogólna zasada działania

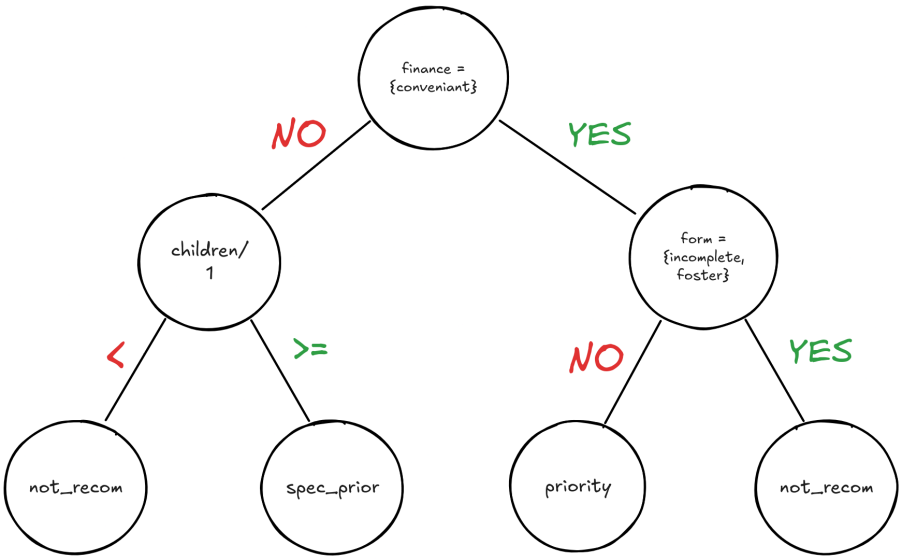
Drzewo klasyfikacyjne to relatywnie prosty model uczenia maszynowego. Polega on na konstrukcji drzewa binarnego, gdzie:

- **węzeł** - atrybut, na podstawie którego dzielimy klasy na podzbiory (tzw. "test")
- **liść** - klasa lub predykcja klasy

Drzewo można traktować jako wielką instrukcję "if-else" - żeby dojść do liścia, trzeba przejść przez wiele węzłów warunkowych. Kolejne testy w drzewie są wybierane automatycznie, wg wskaźnika **zysku informacji** (Information Gain) dla danego atrybutu (w każdym węźle wybieramy atrybut, dla którego jest on największy).

W podstawowej, nieograniczonej wersji drzewa klasyfikacyjnego, dążymy do momentu, gdy wszystkie dane w liściu należą do jednej klasy. Im lepszy test zostanie wybrany do podziału danych, tym precyzyjniejsze będzie nasze drzewo - oznacza to, że celem jest znalezienie takiego testu, dla którego entropia ("nieczystość") zbioru zmniejszy się najbardziej. Jest to ryzykowne podejście, ponieważ czyni ono drzewo bardzo wrażliwym na dane - niewielka zmiana danych wejściowych może znacznie zmienić konstrukcję całego drzewa. Dodatkowo, entropia jest wyliczana dla danych uczących, więc istnieje duże ryzyko przeuczenia drzewa, szczególnie dla niereprezentatywnych zbiorów danych.

Przykładowe drzewo dla zbioru danych "Nursery"



2.1.2 Entropia

Entropia to miara niejednorodności podziału klas. Im większą ma wartość, tym klasy są bardziej wymieszane. Jest obliczana wg wzoru:

$$E(X) = \sum_{d \in C} -P(c = d) \ \log P(c = d)$$

gdzie:

- X - zbiór przykładów w danym węźle
- $P(c = d)$ - p-stwo wystąpienia klasy d w zbiorze ($\frac{\text{liczba wystąpień klasy } d \text{ w zbiorze}}{\text{liczba wszystkich elementów w zbiorze}}$)

Entropia przyjmuje wartość z przedziału $[0, 1]$, dążymy do jej minimalizacji.

Entropia warunkowa określa entropię zbioru po jego podziale danym testem. Oblicza się ją wg wzoru:

$$E(X|T) = \sum_{t} \frac{|X_t|}{|X|} E(X_t)$$

gdzie:

- t - konkretna wartość dla danego testu T
- $|X_t|$ - liczba elementów zbioru, dla których testowany atrybut przyjmuje wartość t
- $|X|$ - liczba wszystkich elementów zbioru X
- $E(X_t)$ - entropia podzbioru X_t zbioru X, w którym wszystkie przykłady przyjmują wartość t dla testowanego atrybutu

Jest to efektywnie średnia ważona entropii podzbiorów podzielonych testem T.

2.1.3 Kryteria podziału

Przyjmujemy następujące kryteria podziału w węzłach:

- **atrybuty ciągłe**: binarny na podstawie nierówności $a_i \leq t$
- **atrybuty dyskretne**: binarny na podstawie przynależności do zbioru $a_i \in T$
  - wprowadzamy ograniczenie, że zbiór T może zawierać najwyżej połowę wartości atrybutu
  - w teście bierzemy pod uwagę tylko kombinację wartości zawartych w zbiorze T, która daje najwyższy przyrost informacji

2.2 Selekcja ruletkowa

Zwykłe drzewa decyzyjne są zachłanne, tzn. wybierają ten test, który ma największą jakość. Takie podejście jest proste w realizacji i bardzo efektywne, jednakże czyni drzewo bardzo podatnym na przeuczenie. W naszym przypadku selekcja będzie odbywała się dla każdego testu z zastosowaniem ruletki wg prawdopodobieństwa:

$$P(a_i) = \frac{IG(a_i)}{\sum_j^n IG(a_j)}$$

gdzie

- $P(a_i)$ - prawdopodobieństwo wyboru atrybutu a_i
- $IG(a_i)$ - zysk informacji dla atrybutu a_i
- $\sum_j^n IG(a_j)$ - suma zysków informacji dla wszystkich atrybutów możliwych do wyboru w danym węźle

Takie podejście sprawia, że prawdopodobieństwo wybrania testu dla atrybutu a_i jest wprost proporcjonalne do zysku informacji, dzięki czemu drzewa powinny mieć mniejszą podatność na przeuczenie. Problem wrażliwości na dane wejściowe również zostanie zredukowany - zmianie będą ulegały prawdopodobieństwa wyboru testu, lecz nie musi to oznaczać zmiany budowy drzewa.

2.3 Algorytmy

Do skonstruowania drzewa klasyfikacyjnego niezbędne są następujące algorytmy:

- Algorytm budowania drzewa
- Algorytm wyboru testu z ruletką
- Algorytm obliczania zysku informacji (**IG**)

Poniżej przedstawiamy pseudokody tych algorytmów w języku Pythono-podobnym w celu lepszego zwizualizowania ich działania:

2.3.1 Algorytm budowania drzewa

```
def build_tree(attrs, data, classes, max_depth) -> DecisionTree:
    if max_depth == 0 or len(attrs) == 0:
        # w przypadku danych niejednorodnych wybieramy najczęstszą klasę
        return most_common(data)
    tree = DecisionTree()

    # przeprowadzamy test z ruletką
    tree.attr, tree.threshold = test(attrs, data, classes)
    new_attr = attrs - tree.attr

    # dzielimy dane na postawie testu
    left_data, right_data = [...]
    tree.left = build_tree(new_attr, left_data, classes, max_depth - 1)
    tree.right = build_tree(new_attr, right_data, classes, max_depth - 1)

    return tree
```

2.3.2 Algorytm wyboru testu z ruletką

```
def test(attrs, data, classes):
    IGs = []
    for a in attrs:
        IGs.append(IG(a, data, classes))

    # ruletkowy wybór zysku informacji
    total = sum(IGs); running_total = 0; p = randint(0, total)
    probabilities = [ig.gain / total for ig in IGs]
    return numpy.random.choice(IGs, 1, p = probabilities)
```

2.3.3 Algorytm obliczania zysku informacji (Information Gain)

Information Gain stosowany jest do mierzenia zmiany entropii dla danego testu. Jest obliczany wg wzoru:

$$IG(X,T) = E(X) - E(X|T)$$

gdzie:

- $E(X)$ - entropia zbioru X
- $E(X|T)$ - entropia warunkowa zbioru X po podziale testem T

Wskaźnik ten pozwala nam w łatwy sposób określić, jak bardzo dany test wpływa na nasz zbiór danych, więc dążymy do jego maksymalizacji.

3 Plan eksperymentów

W celu przeprowadzenia odpowiednich testów statystycznych, eksperymenty będą prowadzone na różnych zbiorach danych, a uzyskane wyniki zostaną porównane z klasyfikatorem DecisionTreeClassifer z pakietu naukowego scikit-learn.

Macież błędów (tablica pomyłek) posłuży nam do zwizualizowania i zweryfikowania skuteczności klasyfikacji. Będziemy skupiać się na miarach: **PPV**, **Recall** i **F1**.

4 Zbiory danych

Przygotowaliśmy 4 zbiory danych, na których będziemy prowadzić eksperymenty.

4.1 Red Wine Quality

Zawiera 11 fizykochemicznych atrybutów win:

1. Kwasowość stała
2. Kwasowość lotna
3. Kwas cytrynowy
4. Cukier pozostały po fermentacji
5. Chlorki

6. Dwutlenek siarki wolny
7. Dwutlenek siarki całkowity
8. Gęstość
9. pH
10. Siarczany
11. Procent alkoholu

Wszystkie atrybuty są ciągłe, więc zastosowany będzie wyłącznie podział nierównościowy.

Zadanie klasyfikacji - **określenie jakości wina w skali całkowitoliczbowej (1-10)**.

4.2 Loan Approval Classification

Zawiera 9 atrybutów o osobie składającej wniosek o pożyczkę oraz 4 atrybuty o samej pożyczce - łącznie 13 atrybutów, na podstawie których należy sklasyfikować stan wniosku (zaakceptowany bądź odrzucony). Atrybuty:

1. Wiek
2. Płeć
3. Wykształcenie
4. Dochód roczny
5. Liczba lat doświadczenia zawodowego
6. Stan posiadania domu (wynajem, na własność, hipoteka)
7. Kwota pożyczki
8. Cel pożyczki
9. Oprocentowanie pożyczki
10. Wysokość wypożyczenia w relacji do dochodu rocznego (%)
11. Zdolność kredytowa
12. Długość historii kredytowej w latach
13. Indykator wcześniejszych niespłaconych wypożyczeń

W tym zbiorze danych występują atrybuty zarówno ciągłe, jak i dyskretne, w związku z czym zostaną wykorzystane oba typy podziałów.

Zadanie klasyfikacji - **akceptacja wniosku o pożyczkę (prawda/fałsz)**.

4.3 Nursery

Zawiera 8 atrybutów dotyczących rodziny:

1. Zawód rodziców
2. Jakość zapewnionej opieki nad dzieckiem
3. Struktura rodziny (kompletna, rozbita, itp.)
4. Liczba dzieci
5. Warunki zamieszkania
6. Sytuacja finansowa
7. Sytuacja społeczna
8. Sytuacja zdrowotna

W tym zbiorze występują atrybuty dyskretne, jednakże zostaną zastosowane oba typy podziałów, ponieważ liczbę dzieci da się podzielić nierównościowo.

Zadanie klasyfikacji - **ocena aplikacji do przedszkola (wieloklasowa)**.

4.4 Mobile Device Usage and User Behavior

Zawiera 10 atrybutów (korzystamy z 9), które opisują parametry urządzeń mobilnych oraz aktywności użytkowników:

1. Id użytkownika - nieużywane, bo nieistotne
2. Model urządzenia
3. System operacyjny
4. Czas używania aplikacji
5. SOT (Screen On Time)
6. Codzienne zużycie baterii (mAh)
7. Liczba zainstalowanych aplikacji
8. Codzienne zużycie danych (MB)
9. Wiek
10. Płeć (M/K)

W tym zbiorze występują atrybuty ciągłe i dyskretne, więc zostaną zastosowane oba typy podziałów.

Zadanie klasyfikacji - **ocena zachowania użytkownika (od lekkiego do ekstremalnego użycia w skali całkowitoliczbowej 1-5)**.