Implementacja drzewa decyzyjnego dla atrybutów ciągłych.

Kinga Świderek, Jakub Kowalczyk

1 Treść zadania

Implementacja drzewa decyzyjnego dla atrybutów ciągłych. Porównanie metod ustalenia liczby i miejsc podziałów przestrzeni ciągłej w testach.

W ramach projektu zbadamy jakości 3 metod dyskretyzacji dla zadanej liczby przedziałów (metoda równoczęstościowa, równoodległościowa i k-średnich) oraz porównamy je z metodami bazującymi na ocenie najlepszego miejsca podziału dla wartości ciągłych danego atrybutu podczas procesu uczenia się drzew decyzyjnych (metoda gini impurity i przyrostu informacji).

2 Opis algorytmu

Drzewo decyzyjne jest skierowanym acyklicznym grafem. Wierzchołki, z których wychodzi przynajmniej jedna krawędź - nazywana gałęzią - są węzłami. Te wierzchołki, z których nie wychodzą żadne są liśćmi. Węzły reprezentują testy wartości atrybutów, gałęzie - wyniki tych testów, natomiast liście są klasami. W przypadku atrybutów o wartościach ciągłych naszymi testami będą testy nierównościowe:

$$t(x) = \begin{cases} 1 \ kiedy \ a(x) \leqslant \theta \\ 0 \ kiedy \ a(x) > \theta \end{cases},$$

gdzie $a(x)$ jest wartością pewnego atrybutu dla przykładu x, a θ jest wartością progową z przeciwdziedziny atrybutu.

Innym podejściem jest dyskretyzacja zmiennych ciągłych - wówczas będziemy testować przynależność do zbioru:

$$t(x) = \begin{cases} 1 \ kiedy \ a(x) \in V \\ 0 \ kiedy \ a(x) \notin V \end{cases},$$

gdzie V jest pewnym podzbiorem przeciwdziedziny atrybutu a.

Zwyczajowo $t(x) = 1$ będzie oznaczało przejście do lewego poddrzewa, natomiast $t(x) = 0$ - do prawego.

Budowanie binarnego drzewa decyzyjnego zostało przedstawione w poniższym pseudokodzie:

function BUDUJ DRZEWO(P: zbiór etykietowanych przykładów, d: klasa domyślna, S: zbiór możliwych testów)
  if $kryterium\ stopu(P, S)$ **then**
    utwórz liść **l**
    $d_{\mathbf{l}} := klasa(P, d)$
    return l
  end if
  utwórz węzeł **n**
  $t_{\mathbf{n}} := wybierz\ test(P, S)$
  $d := klasa(P, d)$
  $\mathbf{n}[0] := buduj\ drzewo(P_{t_{\mathbf{n}}0}, d, S - t_{\mathbf{n}})$
  $\mathbf{n}[1] := buduj\ drzewo(P_{t_{\mathbf{n}}1}, d, S - t_{\mathbf{n}})$
  return n
end function

W powyższym algorytmie należy sprecyzować:

- kryterium stopu,
- wybór klasy,
- wybór testu.

2.1 Kryterium stopu

Podejmowana jest decyzja, czy następny węzeł powinien być liściem. Kryterium stopu jest spełnione, jeśli:

1. Próbujemy budować drzewo na zbiorze P, zawierającym przykłady należące tylko do jednej klasy. Wówczas będzie ona klasą liścia.
2. Próbujemy budować drzewo na pustym zbiorze P. W tym przypadku klasą liścia będzie domyślna klasa d.
3. Próbujemy budować drzewo na pustym zbiorze S. Oznacza to, że nasz zbiór treningowy nie był wystarczający do zbudowania poprawnego klasyfikatora. Algorytm zwróci wówczas liść o klasie większościowej ze zbioru P: $arg\ max_{d'}|P^{d'}|$

W innych przypadkach kryterium stopu nie jest spełnione.

2.2 Wybór klasy

Funkcja $klasa(P, d)$ powinna zwracać najbardziej liczną klasę w zadanym zbiorze P bądź klasę domyślną d, gdy zbiór P jest pusty:

$$\begin{cases} d & kiedy\ P = \emptyset \\ arg\ max_{d'}|P^{d'}| & w\ przeciwnym\ przypadku \end{cases}$$

2.3 Wybór testu

Do prześledzenia pierwszych czterech metod użyjemy posortowanych niemalejąco danych (w latach) wieku respondentów pewnej populacji: {25, 30, 32, 40, 42, 45, 50, 52, 55, 60, 65, 70}. Każda z metod umożliwia podział na dowolną skończoną liczbę przedziałów, lecz dla zobrazowania działania będziemy dokonywać podziału na 3 grupy.

Metody wyłaniania testów przynależnościowych:

1. **Metoda Equal-frequency** (Równoczęstościowa): dzieli zakres wartości atrybutu tak, aby każdy przedział zawierał zbliżoną liczbę obserwacji.

   Dla przykładu:

   Przydzielamy do każdej grupy $\frac{12}{3} = 4$ elementy.

   Grupa 1: {25, 30, 32, 40}

   Grupa 2: {42, 45, 50, 52}

   Grupa 3: {55, 60, 65, 70}

2. **Metoda Equal-width** (Równoodległościowa): dzieli zakres wartości atrybutu na przedziały o jednakowej szerokości.

   Dla przykładu:

   Obliczmy zakres wartości: $70 - 25 = 45$

   Dla trzech grup szerokość przedziału wynosi: $\frac{45}{3} = 15$

   Przedział 1: $< 25; 40)$

   Przedział 2: $< 40; 55)$

   Przedział 3: $< 55; 70 >$

3. **Algorytm k-means** (k-średnich): grupuje wartości atrybutu w k klastrach, gdzie k to liczba przedziałów, a klastry są wyznaczane w taki sposób, aby minimalizować sumę kwadratów odległości między punktami a ich przypisanymi środkami masowymi.

   Działanie wraz z obliczeniami dla przykładu:

   (a) Inicjalizujemy początkowe k środków masowych. Możemy to zrobić losowo lub poprzez wybór k początkowych punktów z zestawu danych. Wybierzmy trzy początkowe środki masowe: $m1 = 30$, $m2 = 45$, $m3 = 60$.

   (b) Przypisujemy każdy punkt danych do klastra, którego środek masowy jest mu najbliższy:

   $$25 \to m1, 30 \to m1, 32 \to m1, 42 \to m2, 45 \to m2,$$
   $$50 \to m2, 52 \to m3, 55 \to m3, 60 \to m3, 65 \to m3,$$
   $$70 \to m3$$

(c) Dla każdego klastra aktualizujemy wartość środka masowego będącego średnią arytmetyczną punktów przypisanych do tego klastra.

$m1 = \frac{25+30+32+40}{4} = 31,75$

$m2 = \frac{42+45+50}{3} = 45,67$

$m3 = \frac{52+55+60+65+70}{5} = 60,4$

(d) Powtarzamy kroki 2 i 3 do momentu, aż środki masowe będą stabilne (przestaną się zmieniać) lub osiągniemy zadaną liczbę iteracji.

Po dwóch iteracjach uzyskaliśmy stabilne środki masowe klastrów. Ostateczny podział wieku respondentów na trzy grupy to:

Grupa 1: {25, 30, 32, 40}

Grupa 2: {42, 45, 50}

Grupa 3: {52, 55, 60, 65, 70}

Metody wyłaniania testów nierównościowych:

1. **Gini impurity**: mierzy prawdopodobieństwo błędnej klasyfikacji losowo wybranej instancji. Im niższy wskaźnik Giniego, tym mniejsze prawdopodobieństwo błędnej klasyfikacji.

$$Gini = 1 - \sum_{i=1}^{N} P(i)^2,$$

gdzie N to liczba klas, $P(i)$ - częstotliwość występowania każdej klasy w zbiorze po podziale.

Rozważmy przykład, w którym decydujemy na podstawie informacji o wieku oraz preferencjach gry w koszykówkę i siatkówkę, czy dana osoba lubi grać w piłkę nożną.

Lubi Koszykówkę	**Lubi Siatkówkę**	**Wiek**	**Lubi piłkę nożną**
Tak	Tak	7	Nie
Tak	Nie	12	Nie
Nie	Tak	18	Tak
Nie	Tak	35	Tak
Tak	Tak	38	Nie
Tak	Nie	50	Nie
Nie	Nie	83	Nie

Skupmy się na obliczeniach dla atrybutu ciągłego - wieku.

(a) Sortujemy wartości w kolejności niemalejącej.

(b) Dla każdej z dwóch sąsiadujących wartości obliczamy ich średnią. Będą to potencjalne punkty podziału w drzewie ($wiek < wybrana\ srednia$).

Otrzymujemy kolejno $\frac{7+12}{2} = 9,5; 15; 26,5; 36,5; 44; 66,5$

(c) Obliczamy "gini impurity" dla każdego z wyznaczonych punktów.

Weźmy punkt 9,5.

Poniżej 9,5 roku życia znajduje się 1 osoba nie lubiącą grać w piłkę nożną oraz 0 osób lubiących ten sport.

$$Gini = 1 - P(tak)^2 - P(nie)^2 = 1 - \frac{0}{1} - \frac{1}{1} = 0$$

Wśród osób posiadających 9,5 i więcej lat możemy wyróżnić 3 lubiącę grę w piłkę nożną oraz 3 nielubiące.

$$Gini = 1 - (\frac{3}{6})^2 - (\frac{3}{6})^2 = 0,5$$

Jako średnią ważoną tych dwóch wskaźników liczymy całkowite $gini\ impurity = \frac{1}{7} * 0 + \frac{6}{7} * \frac{1}{2} = 0,429$

Punkt podziału	**Gini Impurity**
9,5	0,429
15	0,343
26,5	0,476
36,5	0,476
44	0,343
66,5	0,429

(d) Wybieramy dowolny z punktów o najniższym wskaźniku. Przyjmijmy, że będzie to 15. Wówczas uzyskaliśmy pierwszy atrybut wraz z wartością progu do podziału drzewa ($wiek < 15$).

(e) Kontynuujemy kroki 1-4 dla pozostałych atrybutów.

2. **Information gain** (przyrost informacji): używa się do określenia, jak bardzo dany atrybut pomaga w podziale zbioru danych, redukując niepewność co do klas obiektów. Przyrost informacji jest miarą tego, jak bardzo podział zbioru na podstawie danego atrybutu zmniejsza entropię (miarę nieuporządkowania) w porównaniu do entropii przed podziałem. Niższa entropia oznacza bardziej jednorodne podzbiory danych. Entropia wyraża się wzorem:

$$H(X) = -\sum_{i=1}^{n} p(x_i) \log_r p(x_i),$$

gdzie $p(x_i)$ oznacza prawdopodobieństwo wystąpienia danego atrybutu (lub klasy) x_i w danym zbiorze danych.

Przyrost informacji to różnica między entropią początkową a ważoną sumą entropii po podziale:

$$IG(T, a) = H(T) - H(T|a)$$

Znajdźmy punkt podziału przy użyciu przyrostu informacji bazując na poprzednim przykładzie.

(a) Sortujemy wartości atrybutu ciągłego - wieku w kolejności niemalejącej.

(b) Tym razem uwzględniamy tylko średnie wartości między punktami, które różnią się między sobą klasą. Będą to zatem punkty: $\frac{12+18}{2} = 15$; $\frac{35+38}{2} = 36,5$

(c) Liczymy entropię całego zbioru - w przypadku dwóch klas ze wzoru:

$$H(S) = -p_+ \cdot \log_2(p_+) - p_- \cdot \log_2(p_-),$$

gdzie p_+ to prawdopodobieństwo wystąpienia klasy "Tak" (Lubi piłkę nożną), p_- to prawdopodobieństwo wystąpienia klasy "Nie"

$$H(S) = -\left(\frac{2}{7}\log_2\left(\frac{2}{7}\right)\right) - \left(\frac{5}{7}\log_2\left(\frac{5}{7}\right)\right) \approx 0,863$$

(d) Liczymy entropię ważoną i przyrost informacji dla zbiorów S_1 i S_2 utworzonych po podziale całego zbioru S względem wybranego punktu ($wiek < 15$).

$$H(wazona) = \frac{|S_1|}{|S|} \cdot H(S_1) + \frac{|S_2|}{|S|} \cdot H(S_2)$$

$$H(S_1) = -\left(\frac{0}{2}\log_2\left(\frac{0}{2}\right)\right) - \left(\frac{2}{2}\log_2\left(\frac{2}{2}\right)\right) = 0$$

$$H(S_2) = -\left(\frac{2}{5}\log_2\left(\frac{2}{5}\right)\right) - \left(\frac{3}{5}\log_2\left(\frac{3}{5}\right)\right) \approx 0,971$$

$$H(wazona) = \frac{2}{7} \cdot 0 + \frac{5}{7} \cdot 0,971 \approx 0,693$$

(e) Przyrost informacji:

$$Gain(wiek < 15) = 0,971 - 0,693 = 0,276$$

(f) W analogiczny sposób liczmy przyrost informacji dla pozostałych punktów:

$$Gain(wiek < 36,5) = 0,971 - 0,286 = 0,685$$

(g) Jako punkt podziału wybieramy ten, dla którego przyrost informacji był największy, a więc 36,5.

(h) Kontynuujemy kroki 3-6 dla pozostałych atrybutów.

3 Plan eksperymentów

3.1 Zbiory danych

Eksperymenty będziemy przeprowadzać na dwóch zbiorach danych.

1. Klasyfikacja binarna. Zbiór do przewidywania cukrzycy u kobiet - na podstawie 8 atrybutów z dziedziny rzeczywistej klasyfikuje na osoby zdrowe (0) i chore (1). Ilość przykładów dla klasy 0: 500; ilość przykładów dla klasy 1: 268. Link do Kaggle.

2. Klasyfikacja wieloklasowa. Zbiór *iris*, klasyfikujący dane z 4 atrybutów ciągłych dotyczących rozmiaru kwiatu na 3 gatunki irysów. Ilość przykładów dla klasy *setosa*: 50; ilość przykładów dla klasy *versicolor*: 50; ilość przykładów dla klasy *virginica*: 50. Link do Kaggle.

Zbiory danych zostaną losowo podzielone na zbiory treningowe i testowe w proporcji 80:20.

3.2 Opis

Dla każdego zbioru danych algorytm zostanie uruchomiony 25 razy w celu uzyskania zagregowanych wyników. Wśród nich znajdzie się średnia, odchylenia standardowe, najlepszy i najgorszy wynik.

Dla każdej z 3 klasycznych metod dyskretyzacji zbadamy różne możliwości ustalenia liczby przedziałów i spróbujemy znaleźć tę optymalną. Metody gini impurity i przyrostu informacji wymagają podziału na 2 grupy wartości i poprzez ocenianie automatycznie znajdują najlepsze miejsce podziału; skupimy się zatem na porównaniu efektywności takiego rozwiązania względem ręcznego zadawania wymaganej liczby grup pozostałym trzem metodom.

Poza samą jakością modelu ocenianą poprzez opisanie w punkcie 4-tym miary, porównany zostanie także sam czas trwania obliczeń. Zwrócimy także szczególną uwagę na stabilność algorytmu dla różnych uruchomień tzn. w jakim stopniu zachowuje się on spójnie i przewidywalnie.

4 Ocena jakości modelu

Do oceny jakości modelu wykorzystamy następujące metody:

1. **błąd**: liczba niepoprawnie zaklasyfikowanych danych podzielona przez liczbę wszystkich danych testowych

2. **dokładność**: liczba poprawnie zaklasyfikowanych danych podzielona przez liczbę wszystkich danych testowych

3. **macierz pomyłek**:

(a) dla klasyfikacji binarnej: macierz, przechowująca ilości prawdziwych pozytywnych (TP), prawdziwych negatywnych (TN), fałszywych pozytywnych (FP) oraz fałszywych negatywnych (FN) po przetestowaniu na zbiorze testowym.

		h: 0	h: 1
c	0	TN	FP
	1	FN	TP

(b) dla klasyfikacji wieloklasowej: dla każdej klasy skonstruowana zostanie jedna macierz 2×2, zgodnie z powyższym wzorem. Przypadkiem "pozytywnym" będzie zaklasyfikowanie przykładu jako tej klasy; przypadkiem "negatywnym" będzie zaklasyfikowanie przykładu jako inną klasę.

Dla problemu klasyfikacji binarnej zostanie również policzone:

4. **odzysk** (recall): $\frac{TP}{TP+FN}$
5. **precyzja** (precision): $\frac{TP}{TP+FP}$
6. **miara F**: średnia harmoniczna precyzji i odzysku, $\frac{2*recall*precision}{recall+precision}$

Dodatkowo, dla problemu klasyfikacji wieloklasowej zostanie policzona:

7. **mikrośrednia**:
   (a) odzysku:
   $$\frac{\sum_{i=1}^{N} TP_i}{\sum_{i=1}^{N}(TP_i + FN_i)},$$
   gdzie TP_i - liczba prawdziwych pozytywnych przypadków dla klasy i; FN_i - liczba fałszywych negatywnych przypadków dla klasy i
   (b) precyzji:
   $$\frac{\sum_{i=1}^{N} TP_i}{\sum_{i=1}^{N}(TP_i + FP_i)}$$
   (c) miary F:
   $$\frac{2 * \sum_{i=1}^{N} recall_i * precision_i}{\sum_{i=1}^{N} recall_i + \sum_{i=1}^{N} precision_i},$$
   gdzie $recall_i = \frac{TP_i}{TP_i+FN_i}$; analogicznie $precision_i = \frac{TP_i}{TP_i+FN_i}$
8. **makrośrednia**:

(a) odzysku:

$$\frac{1}{N}\sum_{i=1}^{N}(\frac{TP_i}{TP_i + FN_i})$$

(b) precyzji:

$$\frac{1}{N}\sum_{i=1}^{N}(\frac{TP_i}{TP_i + FP_i})$$

(c) miary F:

$$\frac{1}{N}\sum_{i=1}^{N} F_i,$$

gdzie $F_i = \frac{2*recall_i*precision_i}{recall_i+precision_i}$