Dokumentacja wstępna

UMA
Andrii Gamalii,
Wiktor Topolski

Treść zadania

Algorytm analogiczny do lasu losowego z użyciem zbiorów reguł zamiast drzew decyzyjnych.

Interpretacja zadania

Las losowy z użyciem zbiorów reguł **uporządkowanych** zamiast drzew decyzyjnych.
Indukcja reguł za pomocą specjalizacji AQ.

Opis algorytmów

Zmodyfikowany las losowy

B zbiorów reguł (oryginalnie drzew decyzyjnych).

Uczenie

Hiperparametry

- B - maksymalna liczba zbiorów reguł
- M - wielkość podzbioru trenującego dla każdej reguły

Opis algorytmu

Nauczenie B zbiorów reguł na losowo wybranym podzbiorze zbioru trenującego o wielkości M (nie usuwając tych przykładów ze zbioru trenującego) (nie może mieć sprzecznych przykładów), z losowo (bez zwracania) wybranymi atrybutami o liczbie równej podłodze pierwiastka maksymalnej liczby atrybutów. Sposób nauczania (indukcja reguł) - algorytm AQ - opisany niżej.

Predykcja

Parametry

- B nauczonych zbiorów reguł
- V - wektor atrybutów

Opis algorytmu

Wybieramy tą klasę, którą wybrała większość zbiorów reguł.

Specjalizacja AQ

Algorytm tworzący zbiór reguł dla danego zbioru trenującego.
Algorytm zakłada dyskretne i skończone zbiory wartości atrybutów.

- **Kompleks** -⟨s1, s2, ..., sn⟩ - wektor n (koniunkcja) selektorów dla poszczególnych n atrybutów, spełniony gdy wszystkie selektory są spełnione, występują różne warunki np. ⟨2, 3 ∨ 2, ?⟩
  ? - selektor uniwersalny
- **Reguła** - kompleks i klasa którą mają przykłady spełniane przez kompleks.
- **Zbiór reguł** (uporządkowany) - każda reguła stosowana tylko dla przykładów niepokrytych przez wcześniejsze reguły, ostatnia reguła określa klasę domyślną, brak możliwości konfliktu reguł o różnych klasach.

Uczenie

Hiperparametry

- m - Maksymalna ilość kompleksów, które pozostają po specjalizacji (najlepiej ocenione kompleksy)
- T - Maksymalna ilość reguł w jednym zbiorze reguł

Modyfikowalne składowe algorytmu

- Sposób wyboru ziarna
- Sposób oceny kompleksu

Opis algorytmu

Algorytm powtarzamy aż wszystkie przykłady zostaną pokryte, lub gdy osiągniemy maksymalną liczbę reguł w zbiorze (t).
Ziarno x_s - przykład podstawowy, pierwszy wybierany, jest zawsze pokryty przez nową regułę.

$R^{(0)}_G$ - zbiór wszystkich pokrywanych przykładów niewłaściwej klasy przez kompleksy ze zbioru G w danej iteracji.

R - niepokryte przykłady
G = $< ? >$;
$x_s \in R$ - ziarno
$R^{(1)} = R_{c=c(x_s)}$ - klasa ziarna
$R^{(0)} = R_{c \neq c(x_s)}$
jak długo $R^{(0)}_G \neq \emptyset$ (kiedy są rzeczy w tym r0):

- wybierz $x_n \in R^{(0)}_G$ - ziarno negatywne
- dla wszystkich $k \in \{k' \in G \mid k' \triangleright x_n\}$
  - G = G - {k} $\cup$ specjalizacja(k, x_n, x_s)
- G = G - {k $\in$ G | $(\exists k' \in G)$ k' $\succ k$}
- G = $Arg^m max_{k \in G} \nu_{R^{(1)}, R^{(0)}}$ (k) (m najlepszych kompleksów)

zwróć $arg\ max_{k \in G} \nu_{R^{(1)}, R^{(0)}}$ (k) (najlepszy kompleks z G)
gdzie
$\nu_{R^{(1)}, R^{(0)}}(k)$ to ocena jakości kompleksu k

- zazwyczaj poprzez liczbę wcześniej niepokrywanych przykładów, pokrywanych przez dany kompleks

Uporządkowanie zbioru reguł:

- eliminacja pokrywania przykładów klas innych niż c(x_s) wyłącznie w zbiorze R (przykładów niepokrytych przez wcześniejsze reguły)

specjalizacja(k, x_n, x_s)
chcemy uzyskać zbiór maksymalnie ogólnych kompleksów k' spełniających warunki:

- $k' \prec k$
- $k' \not\triangleleft x_n$
- $k' \triangleright x_s$

Przykład działania

Dany zbiór trenujący R:

x	a_1	a_2	a_3	c
1	1	1	3	1
2	2	2	2	1
3	3	2	1	0
4	1	1	2	0

Gdzie a_1: X -> {1, 2, 3}, a_2: X -> {1, 2}, a_3: X -> {1, 2, 3}, c: X -> {0, 1}.
Maksymalny rozmiar **m** zbioru kompleksów **G** = 2.
Proces indukcji pierwszej reguły:

1. Jako ziarno $\mathbf{x_s}$ wybieramy pierwszy przykład, czyli $\mathbf{x_1}$.
   Ziarno jest klasy 1, więc $\mathbf{R^{(1)}} = \{x_1, x_2\}$, $\mathbf{R^{(0)}} = \{x_3, x_4\}$.
   Licznik reguł t = 0.

   Zbiór kompleksów **G** inicjalizujemy zbiorem zawierającym jeden kompleks uniwersalny:
   G = {< ? >}.

   Kroki 2, 3, 4, 5, 6 są dokonywane w pętli dopóki $\mathbf{R^{(0)}_G}$ nie jest pusty.

2. $\mathbf{R^{(0)}_G} = \{x_3, x_4\}$ - zbiór pokrywanych przykładów niewłaściwej klasy.
3. Jako ziarno negatywne $\mathbf{x_n}$ wybieramy x_3.
4. Kompleks **k** = < ? > jest poddawany operacji **specjalizacja(k, $\mathbf{x_n}$, $\mathbf{x_s}$)**, w wyniku której < ? > przechodzi na {< 1 ∨ 2, ?, ? >, <?, 1, ?>, <?, ?, 2 ∨ 3>}.
5. Ze zbioru **G** usuwany jest kompleks < ? > i wstawiane są kompleksy otrzymane po jego specjalizacji. G = {< 1 ∨ 2, ?, ? >, <?, 1, ?>, <?, ?, 2 ∨ 3>}.
6. Wielkość zbioru **G** wynosi 3, podczas gdy **m** = 2, więc musimy zostawić w nim tylko 2 najlepsze kompleksy, oceniane przez np. **pokrycie**. W tym przypadku według pokrycia najlpesze są kompleksy < 1 ∨ 2, ?, ? >, <?, ?, 2 ∨ 3>, więć kompleks <?, 1, ?> jest odrzucany.

2. **G** = {< 1 ∨ 2, ?, ? >, <?, ?, 3 ∨ 2>}, $\mathbf{R^{(0)}_G} = \{x_4\}$.
3. Jako ziarno negatywne $\mathbf{x_n}$ wybieramy x_4.
4. specjalizacja(< 1 ∨ 2, ?, ? >, x_4, x_1):
   < 1 ∨ 2, ?, ? > -> **< 1 ∨ 2, ?, 2 ∨ 3 >**
   specjalizacja(< ?, 1, ? >, x_4, x_1):
   <?, ?, 2 ∨ 3> -> **< ?, ?, 3>**
5. **G** = {< 1 ∨ 2, ?, 2 ∨ 3 >, < ?, ?, 3>}

6. Rozmiar zbioru **G** nie wykracza poza limit

7. $\mathbf{R^{(0)}_G}$ = {∅}, co oznacza, że nie zostało pokrywanych przykładów niewłaściwej klasy i możemy wyjść z pętli
8. Wybieramy najlepszy kompleks ze zbioru **G,** np. według miary pokrycia najlepszy jest kompleks < 1 ∨ 2, ?, 2 ∨ 3 >.
9. Zwracamy regułę **< 1 ∨ 2, ?, 2 ∨ 3 > -> 1.**
10. **R** = R \ {x_1}, t += 1.
11. Powyższe kroki są powtarzane dopóki R != {∅} oraz t <= T.

Predykcja

Pierwsza reguła spełniająca dany przykład wyznacza klasę, ostatnia reguła wyznacza klasę domyślną.

Zbiór danych

http://archive.ics.uci.edu/dataset/73/mushroom
Klasyfikacja czy grzyb jest trujący na podstawie 20 atrybutów dyskretnych
8124 przykłady - 4208 jadalnych grzybów, 3916 niejadalnych
Zbiór trenujący wybieramy losowo

Eksperymenty

Zmienne algorytmów

- Sposoby oceny kompleksów, np. pokrycie, dokładność, dominacja klasy z uwzględnieniem pokrycia.
- Sposoby wyboru ziarna np. losowy, pierwszy przykład, losowy z klasy dominującej.

Hiperparametry

Użyjemy przeszukiwania **losowego** wartości hiperparametrów.

- Rozmiar zbioru trenującego (dla całego algorytmu), np. 80%, 90%, 95%
- Rozmiar zbiorów trenujących dla poszczególnych zbiorów reguł, np. 500 przykładów
- Maksymalna ilość zbiorów reguł, np. 100, 500, 1400
- Maksymalna ilość reguł w zbiorze reguł, np. 2, 10, 250

Miary jakości

- Confusion matrix
- Dokładność
- Precyzja
- F1-score