Data mining

Nguyen Hung Son

Motywacje

Algorytm SPRINT

Szukanie najlepszego podziału

Dokonanie podziału

SPRINT - wersja równoległa

Wykład 7: Drzewa decyzyjne dla dużych zbiorów danych

Nguyen Hung Son

Funkcja rekurencyjna $buduj\_drzewo(U, dec, \mathbf{T})$:

1: **if** $(kryterium\_stopu(U, dec) = \mathbf{true})$ **then**
2: $\quad \mathbf{T}.etykieta = kategoria(U, dec)$;
3: $\quad$ **return**;
4: **end if**
5: $t := wybierz\_test(U, \mathbf{TEST})$;
6: $\mathbf{T}.test := t$;
7: **for** $v \in R_t$ **do**
8: $\quad U_v := \{x \in U : t(x) = v\}$;
9: $\quad$ utwórz nowe poddrzewo $\mathbf{T}'$;
10: $\quad \mathbf{T}.$gałąź$(v) = \mathbf{T}'$;
11: $\quad buduj\_drzewo(U_v, dec, \mathbf{T}')$
12: **end for**

Funkcje pomocnicze

- Kryterium stopu: Zatrzymamy konstrukcji drzewa, gdy aktualny zbiór obiektów:
  - jest pusty lub
  - zawiera obiekty wyłącznie jednej klasy decyzyjnej lub
  - nie ulega podziale przez żaden test
- Wyznaczenie etykiety zasadą większościową:

$$kategoria(P, dec) = \arg \max_{c \in V_{dec}} |P_{[dec=c]}|$$

  tzn., etykietą dla danego zbioru obiektów jest klasa decyzyjna najliczniej reprezentowana w tym zbiorze.
- Kryterium wyboru testu: heurytyczna funkcja oceniająca testy.

Miary różnorodności zbioru

Każdy zbiór obiektów X ulega podziale na klasy decyzyjne:

$$X = C_1 \cup C_2 \cup ... \cup C_d$$

gdzie $C_i = \{u \in X : dec(u) = i\}$.
Wektor $(p_1, ..., p_r)$, gdzie $p_i = \frac{|C_i|}{|X|}$, nazywamy rozkładem klas decyzyjnych w X.

$$Conflict(X) = \sum_{i<j} |C_i| \times |C_j| = \frac{1}{2}\left(|X|^2 - \sum |C_i|^2\right)$$

$$Entropy(X) = -\sum \frac{|C_i|}{|X|} \cdot \log \frac{|C_i|}{|X|} = -\sum p_i \log p_i$$

$$Gini(X) = 1 - \sum p_i^2$$

Ocena funkcji testu

Data mining

Nguyen Hung Son

Motywacje

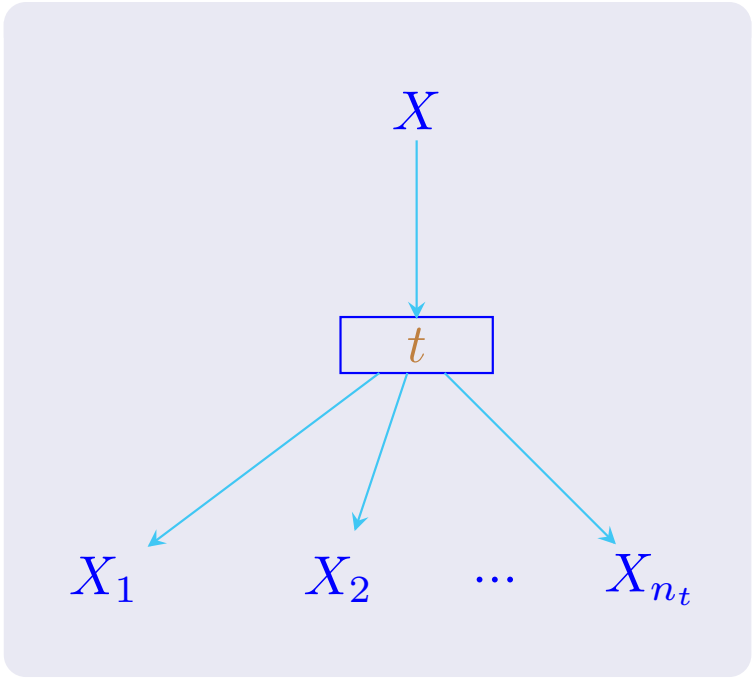
Algorytm SPRINT

Szukanie najlepszego podziału

Dokonanie podziału

SPRINT - wersja równoległa

Każdy test t jest oceniony na podstawie informacji zawartych w X, $X_1, ..., X_{n_t}$



Podział zbioru X dokonany przez test t;

Ocena funkcji testu

- Rozróżnialność:

$$disc(t, X) = conflict(X) - \sum conflict(X_i)$$

- Przyrostu informacji (Information gain).

$$Gain(t, X) = Entropy(X) - \sum \frac{|X_i|}{|X|} \cdot Entropy(X_i)$$

- Gini's index

$$G(t, X) = Gini(X) - \sum \frac{|X_i|}{|X|} \cdot Gini(X_i)$$

- Kara za zbyt drobny podział, np. gain ratio

$$Gain\_ratio = \frac{Gain(t, X)}{-\sum_{i=1}^{r} \frac{|X_i|}{|X|} \cdot \log \frac{|X_i|}{|X|}}$$

Przykład

Cheat	No	No	No	Yes	Yes	Yes	No	No	No	No
	Taxable Income									
Sorted Values →	60	70	75	85	90	95	100	120	125	220
Split Positions →	55	65	72	80	87	92	97	110	122	172	230
	<= >	<= >	<= >	<= >	<= >	<= >	<= >	<= >	<= >	<= >	<= >
Yes	0 3	0 3	0 3	0 3	1 2	2 1	3 0	3 0	3 0	3 0	3 0
No	0 7	1 6	2 5	3 4	3 4	3 4	3 4	4 3	5 2	6 1	7 0
Gini	0.420	0.400	0.375	0.343	0.417	0.400	*0.300*	0.343	0.375	0.400	0.420

Słabości standardowego algorytmu:

- Każdy węzeł jest skojarzony z podzbiorem danych: ograniczenie pamięciowe
- Wyznaczenie najlepszego podziału wymaga wielokrotnego sortowania danych: czasochłonne
  - Dany atrybut rzeczywisty a i zbiór możliwych cięć $(c_1, c_2, ...c_N)$, najlepszy test (a, c_i) można znaleźć w czasie $\Omega(N)$
  - Minimalna liczba prostych zapytań SQL potrzebna do szukania najlepszego testu jest $O(dN)$, gdzie d jest liczbą klas decyzyjnych
- Wniosek: szukanie najlepszego podziału jest kosztowne, jeśli atrybut zawiera dużo różnych wartości.

Charaterystyka algorytmu SPRINT

- Nadaje się dla danych częściowo umieszczonych na dysku
- Używa się techniki pre-sortowania w celu przyspieszenia procesu obliczenia na atrybutach rzeczywistych;
- Dane są sortowane tylko raz przed obliczeniem
- Łatwo można zrównoleglić

Struktura danych w SPRINT

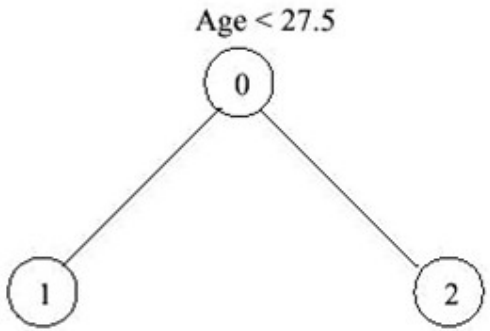
- Każdy atrybut ma swoją listę wartości
- Każdy element listy ma trzy pole:
  - *wartość atrybutu*,
  - *numer klasy* i
  - *rid* (numer obiektu w zbiorze danych)
- Rzeczywiste atrybuty są uporządkowane (tylko raz przy utworzeniu)
- Na początku listy są stowarzyszone z korzeniem drzewa
- Kiedy węzeł podlega podziale, listy są podzielone i są skojarzone z odpowiednimi następnikami
- Listy są zapisane na dysku w razie potrzeby

Przykład

Attribute lists for node 0

Age	Class	Tid
17	High	1
20	High	5
23	High	0
32	Low	4
43	High	2
68	Low	3

Car Type	Class	Tid
family	High	0
sports	High	1
sports	High	2
family	Low	3
truck	Low	4
family	High	5



Attribute lists for node 1

Age	Class	Tid
17	High	1
20	High	5
23	High	0

Car Type	Class	Tid
family	High	0
sports	High	1
family	High	5

Attribute lists for node 2

Age	Class	Tid
32	Low	4
43	High	2
68	Low	3

Car Type	Class	Tid
sports	High	2
family	Low	3
truck	Low	4

Struktura danych w SPRINT

Data mining

Nguyen Hung Son

Motywacje

Algorytm SPRINT

Szukanie najlepszego podziału

Dokonanie podziału

SPRINT - wersja równoległa

- SPRINT używa:
  - indeksu Gini do oceny jakości podziału
  - testu typu $(a \leq c)$ dla atrybutów rzeczywistych
  - testu typu $(a \in V)$ dla atrybutów symbolicznych
- Histogram: rozkład klas decyzyjnych zbadanego zbioru danych
- Dla atrybutu rzeczywistego dwa histogramy:
  C_{below} : histogram dla danych poniżej wartości progowej
  C_{above} : histogram dla danych powyżej wartości progowej
- Dla atrybutu symbolicznego jeden histogram zwany count matrix

Przykład

Car Type	*Class*	*rid*
family	High	0
sports	High	1
sports	High	2
family	Low	3
truck	Low	4
family	high	5

Punkt podziału

Age	*Class*	*rid*
17	High	1
20	High	5
23	High	0
32	Low	4
43	High	2
68	Low	3

Count matrix

	H	L
family	2	1
sports	2	0
truck	0	1

Histogram klas

	H	L
C_{below}	3	0
C_{above}	1	2

Outline

1 Motywacje

2 Algorytm SPRINT
- Szukanie najlepszego podziału
- Dokonanie podziału
- SPRINT - wersja równoległa

Wyznaczanie podziału atrybutu rzeczywistego

Attribute List

Age	Class	tid	Position of cursor in scan
17	High	1	← position 0
20	High	5	
23	High	0	← position 3
32	Low	4	
43	High	2	
68	Low	3	← position 6

State of Class Histograms

cursor position 0:

	H	L
C_{below}	0	0
C_{above}	4	2

cursor position 3:

	H	L
C_{below}	3	0
C_{above}	1	2

cursor position 6:

	H	L
C_{below}	4	2
C_{above}	0	0

Wyznaczanie podziału atrybutu symbolicznego

Lista wartości ***CarType***

Car Type	Class	rid
family	High	0
sports	High	1
sports	High	2
family	Low	3
truck	Low	4
family	high	5

Count Matrix

	H	L
family	2	1
sports	2	0
truck	0	1

1. Wyznacz macierz rozkładu klas obiektów w danym węźle
2. Używając algorytmu aproksymacyjnego (w SLIQ) *wyznacz* podzbiór wartości $V \subseteq D_a$ t. żeby test $(a \in V)$ był optymalny

Outline

1 Motywacje

2 Algorytm SPRINT
- Szukanie najlepszego podziału
- Dokonanie podziału
- SPRINT - wersja równoległa

Główna idea

- Każda lista jest podzielona na dwie listy
- **Atrybut zawierający test:** Podziel wartości listy zgodnie z testem
- **Atrybut niewierający test:**
  - Nie można korzystać z informacji w funkcji testu.
  - Skorzystaj z *rid*
  - Skorzystaj z tablicy haszującej
- **Przy podziale atrybutu zawierający test:** wstaw rid rekordów do tablicy haszującej.
- **Tablica haszująca:** informacje o tym do którego poddrzewa rekord został przeniesiony.
- **Algorytm:**
  - Przeglądaj kolejny rekord listy
  - Dla każdego rekordu wyznacz (na podstawie tablicy haszującej) poddrzewo, do którego rekord ma być przeniesiony

Problem: zbyt duża tablica haszująca

Data mining

Nguyen Hung Son

Motywacje

Algorytm SPRINT

Szukanie najlepszego podziału

Dokonanie podziału

SPRINT - wersja równoległa

Algorytm:

- Krok 1: Podziel zbiór wartości atrybutu testującego na małe porcje tak, żeby tablica haszująca mieściła się w pamięci
- Krok 2: Dla każdej porcji
  - Podziel rekordy atrybutu testującego do właściwego podrzewa
  - Buduj tablicę haszującej
  - Przeglądaj kolejny rekord atrybutu nietestującego i przynieś go do odpowiedniego poddrzewa jeśli rekord występuje w tablicy haszującej
- Krok 3: Jeśli wszystkie rekordy zostały przydzielone do poddrzew **stop**, wpp. **idź do** krok 2

Outline

1 Motywacje

2 Algorytm SPRINT
- Szukanie najlepszego podziału
- Dokonanie podziału
- SPRINT - wersja równoległa

Równoległy SPRINT

Data mining

Nguyen Hung Son

Motywacje

Algorytm SPRINT

Szukanie najlepszego podziału

Dokonanie podziału

SPRINT - wersja równoległa

- Listy wartości atrybutów są równo podzielone
- **Atrybut rzeczywisty:** sortuj zbiór wartości i podziel go na równe przedziały
- **Atrybut numeryczny:** podziel według rid
- Każdy procesor ma jedną część każdej listy

Szukanie najlepszego podziału

Data mining

Nguyen Hung Son

Motywacje

Algorytm SPRINT

Szukanie najlepszego podziału

Dokonanie podziału

SPRINT - wersja równoległa

- Dla atrybutu rzeczywistego:
  - Każdy procesor ma przedział wartości atrybutu
  - Każdy procesor inicjalizuje C_{below} i C_{above} uwzględniając rozkład klas w innych procesorach
  - Każdy procesor przegląda swoją listę i wyznacza najlepszą lokalną wartość progową
  - Procesory komunikują się w celu znalezienia globalnie najlepszego cięcia
- Dla atrybutu symbolicznego:
  - Każdy procesor buduje lokalne count matrix i wysyła wynik do centralnego procesora
  - Centralny procesor oblicza globalny count matrix
  - Procesory wyznaczają najlepszy podział na podstawie globalnego count matrix

Przykład

Procesor 0

Age	*Class*	*rid*
17	High	1
20	High	5
23	High	0

Car Type	*Class*	*rid*
family	High	0
sports	High	1
sports	High	2

Procesor 1

Age	*Class*	*rid*
32	Low	4
43	High	2
68	Low	3

Car Type	*Class*	*rid*
family	Low	3
truck	Low	4
family	high	5

Dokonanie podziału

- Podział atrybutu zawierający test: Każdy procesor wyznacza poddrzewa, do których rekordy w lokalnej liście będą przeniesione
- Procesory wymieniają ze sobą informacje $\langle rids, poddrzewo \rangle$
- Podział pozostałych atrybutów: Po otrzymaniu informacji ze wszystkich procesorów każdy procesor buduje tablicę haszującą i wykonuje podziały dla pozostałych atrybutów

Wady algorytmu SPRINT

- Dodatkowe struktury danych
- Nieefektywny jeśli atrybut ma dużo wartości
- Nie wykorzystuje mocnych narzędzi systemów baz danych