Piotr Kalinowski
Marcin Swend

ALGORYTMY HEURYSTYCZNE
PROJEKT C3

Treść zadania

Selekcja atrybutów do grupowania: przeszukiwanie przestrzeni podzbiorów atrybutów z funkcją oceny opartą na jakości modeli grupowania tworzonych przez algorytmy dostępne w R z wykorzystaniem ocenianego podzbioru.

Interpretacja treści zadania i reprezentacja danych

Danymi wejściowymi dla problemu są:

1. macierz danych numerycznych - dane na podstawie których odbywa się grupowanie
2. zadany podział na grupy

Zadaniem algorytmu heurystycznego jest znalezienie takich kolum, dla których algorytmy dostępne w R dają w wyniku największe podobieństwo do zadanego zbioru grup.

Przykładowe dane wejściowe

1. Macierz danych numerycznych (format csv)

   "Length","Left","Right","Bottom","Top","Diagonal"
   214.8,131,131.1,9,9.7,141

214.6,129.7,129.7,8.1,9.5,141.7
214.8,129.7,129.7,8.7,9.6,142.2
…

2. Zadany podział na grupy
   [1,1,2,…]

Dla reprezentacji elementów przestrzeni rozwiązań używany jest wektor binarny którego każdy element interpretowany jest jako uwzględnienie kolumny w algorytmie.
Motywacją przyjęcia tej reprezentacji jest prostota liczenia sąsiedztwa i intuicyjność interpretacji.

Reprezentacja przykładowego elementu przestrzeni

dla w/w danych (wybrane zostały kolumny “Length” i “Top”; jest to także reprezentacja przykładowego rozwiązania):

[1,0,0,0,1,0]

Rozwiązanie zadania

Do rozwiązania został użyty algorytm VNS.

Parametry wejściowe algorytmu

Algorytm rozwiązujący zadanie przyjmuje 4 parametry:

1. Macierz danych wejściowych
2. Zadany podział na grupy
3. Maksymalna odległość od bieżącego elementu dla generowania rozwiązań K (parametr algorytmu VNS)
4. funkcja używana do klastrowania (opisana niżej)

Algortymy grupowania

Program został napisany w sposób umożliwiający użytkownikowi dodawanie własnych algorytmów grupowania: parametrem funkcji szukającej najlepszego zbioru kolumn jest funkcja przyjmująca dane numeryczne i ilość klastrów i zwracająca podział na grupy (w formacie identycznym jak “zadany podział na grupy”)
Na potrzeby projektu zaimplementowane zostały dwa algorytmy grupowania:

- kmeans
- Mclust

Funkcja celu

Funkcja celu została zaprojektowana tak aby

1. wynik zależał od podobieństwa między wynikiem klastrowania a zadanymi grupami oraz
2. preferowanane były rozwiązania z mniejszą ilością kolumn

Dla (1): podobieństwo jest mierzone skorygowanym indeksem Randa (http://en.wikipedia.org/wiki/Rand_index#Adjusted_Rand_index).
Aby zapewnić (2) indeks Randa jest korygowany o czynnik będący iloczynem ilości wybranych kolumn i bardzo małej liczby rzeczywistej (np. 0.00001) (czynnik jest odejmowany od oceny).

W algortymie funkcja celu jest maksymalizowana.

Funkcje pomocnicze

generateNeighbours

Funkcja generująca sąsiadów zadanego elementu przestrzeni zwracająca jego sąsiadów w odległości równej drugiemu parametrowi. Odległość mierzona jest jako “ilość bitów jakie należy zanegować aby z jednego elementu otrzymać drugi”)

```
> generateNeighbours(c(0,0,1,0),1)
     [,1] [,2] [,3] [,4]
[1,]    1    0    0    0
[2,]    0    1    0    0
[3,]    1    1    0    1
[4,]    0    0    0    1
```

selBest

Funkcja która dla zadanego zbioru elementów (format wejściowy jest identyczny z wyjściowym funkcji generateNeighbours) wybiera element najlepszy korzystając z funkcji celu.
Parametry:

1. zbiór elementów
2. macierz danych
3. zadany przydział do grup
4. ilość grup (dla wywołania przez główną funkcję ilość grup jest liczona automatycznie jako ilość unikalnych elementów w zadanym przydziale do grup)
5. funkcja algorytmu grupowania

Przykładowe wyniki działania algorytmów dla różnych danych wejściowych

dane	algorytm	K	wartość funkcji celu	czas znajdowania rozwiązania
banknoty	kmeans	1,2	0.9799795	1 sek
banknoty	kmeans	6	0.9799795	4 sek

banknoty	mclust	1	0.9799795	1 sek
banknoty	mclust	6	0.9799795	6 sek
irysy	kmeans	1	0.885687	<1 sek
irysy	kmeans	4	0.885687	<1 sek
irysy	mclust	1	0.9409923	1 sek
irysy	mclust	4	0.9409923	1 sek
rak piersi	kmeans	1	0.8371486	55 sek
rak piersi	Mclust	1	0.8439179	1 min 57 sek
rak piersi	Mclust	2	0.8833855	18 min 6 sek
rak piersi	kmeans	2	0.850325	7 min 12 sek
szkło	kmeans	1	0.6588419	1 sek
szkło	kmeans	2	0.705367	4 sek
szkło	kmeans	3	0.6891754	21 sek
szkło	kmeans	4	0.7420731	20 sek

Zbiory danych:

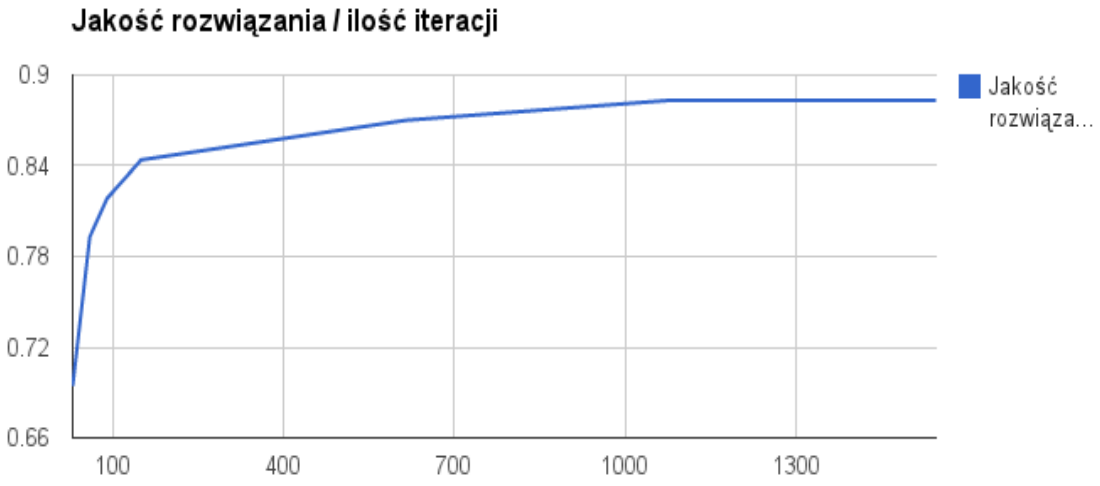
nazwa	wielkość	ilość atrybutów	ilość grup
banknoty	200	6	2
irysy	150	4	3
rak piersi	569	30	2
szkło	214	9	6

Ślad przebiegu dla przykładowych danych:

źródło danych :
http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Diagnostic%29
interpretacja danych:
diagnostyka raka piersi (Breast cancer diagnostic)

K = 2; Mclust; 30 parametrów, grupowanie binarne

Iteration	Solution quality
1	-1
30	0.6942844
60	0.7928478
90	0.8182232
120	0.8309029
150	0.8439179
615	0.8701096
1080	0.8833855
1545	0.8833855 (zakończenie algorytmu)



Instrukcje uruchamiania testów

1. Załadować plik “vns.R”
2. Dostępne testy:
    a. dane: banknoty
        i. kasaKmeans(): K = 2, algorytm kmeans
        ii. kasaMclust(): K = 6 (max), algorytm Mclust
        iii. kasatestbrute(): bruteforce
    b. dane: irysy
        i. irisKmeans(): K=2, algorytm kmeans
        ii. irisMclust(): K=2, algorytm Mclust
        iii. iristestbrute(): bruteforce, algorytm kmeans
        iv. iristestbruteM(): bruteforce, algorytm Mclust
    c. dane: rak piersi
        i. breastKmeans(): K=2, algorytm kmeans
        ii. breastMclust():K=2, algorytm Mclust

bruteforce - test generujący całą przestrzeń i wybierający najlepszy element przy pomocy selBest, przydatny do sprawdzania poprawności, niepraktyczny dla dużych przestrzeni.