Wyszukiwanie najbliższych sąsiadów

17 grudnia 2018

Opis pliku z zadaniami

Wszystkie zadania na zajęciach będą przekazywane w postaci plików `.pdf`, sformatowanych podobnie do tego dokumentu. Zadania będą różnego rodzaju. Za każdym razem będą one odpowiednio oznaczone:

- Zadania do wykonania na zajęciach oznaczone są symbolem $\triangle$ – nie są one punktowane, ale należy je wykonać w czasie zajęć.
- Punktowane zadania do wykonania na zajęciach oznaczone są symbolem $\diamond$ – należy je wykonać na zajęciach i zaprezentować prowadzącemu, w wypadku nie wykonania zadania w czasie zajęć lub nieobecności, zadania staje się zadaniem do wykonania w domu ($\star$).
- Zadania do wykonania w domu oznaczone są symbolem $\star$ – są one punktowane, należy je dostarczyć w sposób podany przez prowadzącego i w wyznaczonym terminie (zwykle przed kolejnymi zajęciami).
- Zadania programistyczne można wykonywać w dowolnym języku programowania, używając jedynie biblioteki standardowej dostępnej dla tego języka.

1 Wyszukiwanie najbliższych sąsiadów *10p.*◇

Treść

Wykorzystując dane dotyczące problemu MSDC rozwiąż problem wyszukiwania najbliższych użytkowników. Podobieństwo pomiędzy użytkownikami należy określić używając współczynnika Jaccarda na zbiorach odsłuchanych utworów muzycznych.

Podobieństwo Jaccarda pomiędzy dwoma zbiorami jest zdefiniowane jako iloraz mocy części wspólnej zbiorów i mocy sumy tych zbiorów:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|},$$

gdzie A i B są zbiorami.

Rozważmy następujący przykład. Niech zbiór $S_1 = \{s_1, s_3, s_4\}$ odpowiada pierwszemu użytkownikowi u_1, a zbiór $S_2 = \{s_2, s_3, s_6\}$ użytkownikowi drugiemu u_2. W postaci tabelarycznej możemy te zbiory zapisać następująco:

użytkownik	s_1	s_2	s_3	s_4	s_5	s_6	s_7
u_1	1	0	1	1	0	0	0
u_2	0	1	1	0	0	1	0

Dla powyższych danych współczynnik Jaccarda wynosi:

$$J(A,B) = \frac{1}{5}$$

Zadanie: Dla stu pierwszych użytkowników, pojawiających się w tabeli `facts`, znajdź stu najbliższych sąsiadów o podobieństwie Jaccarda większym od 0 (w całym zbiorze).

Punktacja:

- Realizacja zadania dowolnym podejściem: 5p.
- Optymalizacja kodu pod względem szybkości działania: 3p.
- Znalezienie stu najbliższych użytkowników dla wszystkich użytkowników: 2p.
- Algorytm liniowy najbliższych sąsiadów dla pojedynczego użytkownika: bonus 2p.