05.12.2019r.

Jakub Skrzypiec
Marcel Zmeczerowski
Informatyka III rok UG

Algorytmy Numeryczne - zadanie 3

Metoda ALS w systemach rekomendacji

Zadanie polegało na zaimplementowaniu systemu rekomendacji w sklepie internetowym na podstawie profilu użytkownika, tj. wystawionych ocen kupionym produktom. Implementacja miała wykorzystywać metodę ALS oraz metodę Gaussa. Danymi miały być prawdziwe, archiwalne, zanonimizowane dane o produktach ze sklepu Amazon udostępnione w ramach projektu SNAP , a testy powinny zostać przeprowadzone na typie 'double' (podwójnej precyzji) dla 3 wybranych samodzielnie podzbiorów danych:

S: małym - od 10 produktów M: średnim - od 100 produktów *B: dużym - od 1000 produktów

Napisaliśmy program w języku Java, który implementuje metodę ALS. Poniżej znajdują się nasze wnioski i wyniki z testów.

1. Sposób doboru danych

1.1) Sposób doboru produktów

1.1.1) Na początku parsujemy plik (konstruktor klasy Parser) i tworzymy 548552 produktów w liście, po czym tworzymy inną, mniejszą listę produktów, która nie zawiera produktów wycofanych oraz tych z ogólną ilością ocen poniżej 70. Ta mniejsza lista zawiera 19207 produktów, które mają 70 lub więcej ocen;

1.1.2) Następnie metoda klasy Parser o nazwie 'dajPodliste(int ileProduktow)' tworzy jeszcze mniejszą listę produktów na podstawie listy produktów z 19207 elementami – nowo utworzona lista 'podlista' ma rozmiar 'ileProduktow' i produkty nie powtarzają się;

Sposób doboru produktów w metodzie 'dajPodliste(int ileProduktow)':

1.1.2.1) Metoda dodaje produkty do nowej listy 'podlista' o docelowym rozmiarze 'ileProduktow' tak długo, aż rozmiar listy nie osiągnie 'ileProduktów'

1.1.2.2) Dla każdego produktu w liście 19207 produktów, sprawdzamy każdy produkt w tej samej liście i sprawdzamy czy:

-produkt podobny do pierwszego jest również produktem podobnym do produktu drugiego;

-produkt podobny do pierwszego jest produktem drugim

i jeżeli którykolwiek z warunków został spełniony dodajemy produkt do nowej listy produktów, o ile już go tam nie ma. Na końcu procesu otrzymujemy listę produktów wielkości 'ileProduktów', w której produkty nie powtarzają się;

1.2) Sposób doboru użytkowników

Metoda parsera 'dajMacierz(List<produkt> podlista)' tworzy listę użytkowników w następujący sposób:

1.2.1) Dodajemy pierwszego oceniającego pierwszego produktu z listy produktów 'podlista';

1.2.2) Następnie dla każdego produktu z list 'podlista', dla każdego oceniającego go użytkownika dodajemy tego użytkownika do nowo utworzonej list użytkowników, o ile tego użytkownika jeszcze nie ma na liście. W ten sposób tworzona jest wielka lista wszystkich użytkowników, którzy ocenili produkty z listy 'podlista'. Chcielibyśmy wybrać tych użytkowników, którzy ocenili jak najwięcej produktów z listy 'podlista', metoda daj macierz robi to następująco:

1.2.2.1) Dla każdego użytkownika oceniającego produkty z listy 'podlista' zliczamy ilość ocen;

1.2.2.2) Sortujemy listę użytkowników po ilości ocen malejąco, tym sposobem użytkownicy na początku listy będą tymi, którzy ocenili jak najwięcej produktów z listy 'podlista';

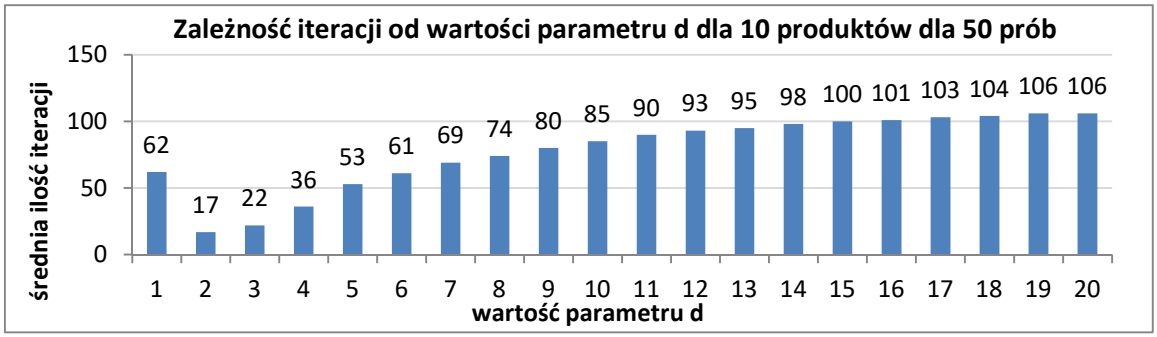
1.2.2.3) Na podstawie posiadanej listy użytkowników tworzymy nową, mniejszą listę użytkowników 'uzytkownicy_podlista', do której dodajemy K*podlista.size() pierwszych użytkowników z listy

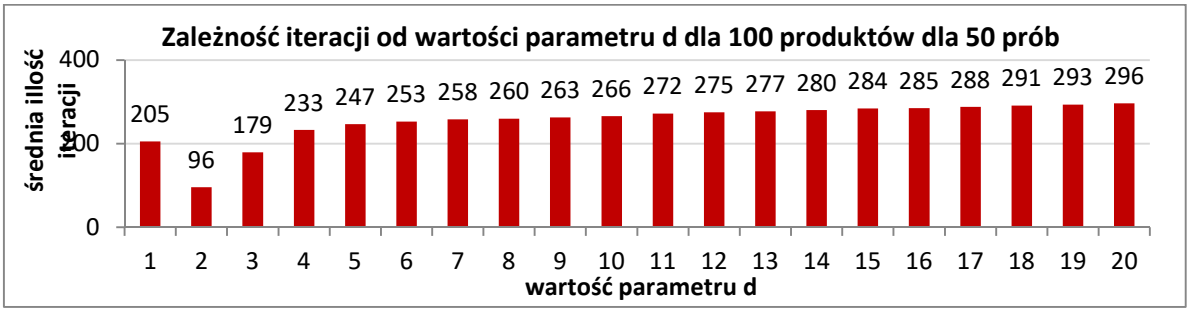
pierwotnej, gdzie podlista.size() jest rozmiarem listy produktów ‘podlista’, a K jest określone w ciele metody;

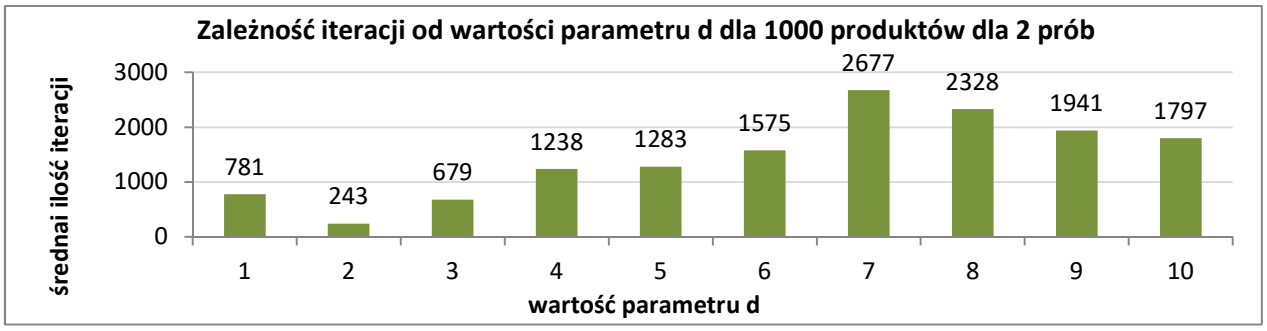
Na podstawie listy produktów ‘podlista’ oraz listy użytkowników ‘podlista_uzytkownicy’ tworzymy macierz o rozmiarze ‘*podlista.size()*’ i ‘*podlista_* użytkownicy.size()’, gdzie ‘*podlista.size()*’ jest rozmiarem listy produktów ‘podlista’, a ‘*podlista_* użytkownicy.size()’ jesy rozmiaru listy użytkowników ‘podlista_uzytkownicy’.

2. Tempo zbieżności metody ALS w zależności od parametru d

W kolejnych iteracjach pętli wyliczaliśmy kolejne funkcje celu oraz różnicę między obecnie wyliczaną funkcją celu a funkcją celu obliczoną w poprzedniej iteracji. Tempo zbieżności metody ALS określiliśmy na podstawie iteracji potrzebnych do tego, aby te dwie funkcje celu różniły się o mniej niż 0,01. Otrzymane wyniki:







Z czego wynika, że pod względem szybkości zbieżności dla wszystkich 3 rozmiarów testowanych podzbiorów najlepsza wartość parametru d to 2.

3. Wpływ parametru *d* na jakość stworzonych rekomendacji i czas obliczeń

Dla kilkudziesięciu prób obliczaliśmy średnio do jakiej liczby zbiega funkcja celu oraz jaki jest średni czas obliczenia rekomendacji. Otrzymane wyniki:

a) dla 10 produktów

d	jakość rekomendacji	Średni czas
1	38.044916547183725	0
2	11.99194658504823	0
3	10.739234399782644	0
4	10.57991345557119	0
5	10.574995999218066	0

b) dla 100 produktów

d	jakość rekomendacji	średni czas
1	603.7904724247262	1,46
2	273.0858847619213	1,08
3	127.34254871673654	2,16
4	96.74306221286366	2,04
5	**92.37730408124949**	**1,52**

c) dla 1000 produktów

d	jakość rekomendacji	Średni czas
1	20527.147105877295	39
2	13748.246119295405	39,5
3	9195.924525840397	47
4	6044.076084335995	70,5
5	4035.0476963158862	75

6	10.57312194888482	0
7	10.581886347694326	0
8	10.582203079713706	0
9	10.586517258487243	0
10	10.588629959861123	0
11	10.590152183002845	0,32
12	10.590032452363662	0
13	10.589798050561392	0
14	10.589890897513028	0,32
15	10.589769975735056	0,32
16	10.58995300813484	0
17	10.589718376818292	0,32
18	10.590529402407444	0
19	10.589658338941552	0,6
20	10.590033966066649	0,64

6	91.40038985611791	2,84
7	91.1560387736146	3,08
8	91.07491145525105	3,58
9	91.04045337661674	3,76
10	91.02681357241094	5,38
11	91.02474576925724	5,50
12	91.01918356103339	4,14
13	91.0172202141956	5,94
14	91.02268230446926	6,94
15	91.0189544993887	7,96
16	91.02011917215466	8,86
17	91.0205818084068	8,84
18	91.02218740085387	9,58
19	91.0267818246208	9,38
20	91.02504412348563	10,96

6	2805.2572670275295	70
7	2057.1328531043505	86
8	1605.1546962997536	94
9	1402.0783943835745	94
10	1256.673664854679	101

4. Przydatność implementowanej metody dla testowanych danych

Naszym zdaniem metoda ALS jest bardzo przydatna, uzyskana rekomendacje są zgodne z intuicją, tzn. produkty oceniane dobrze są rekomendowane, a te oceniane źle lub bardzo źle nie są rekomendowane.

Poniżej przedstawiamy przykład uzyskanych rekomendacji dla 10 produktów dla parametru d=2. Po lewej stronie jest początkowa macierz R, a po prawej wypełniona macierz rekomendacji (pogrubione nowo wyliczone wartości).

Każdy wiersz reprezentuje użytkownika, każda kolumna reprezentuje produkt.

	0	1	2	3	4	5	6	7	8	9
0	2.0	5.0	5.0	5.0	1.0	5.0	1.0	2.0	1.0	1.0
1	5.0	5.0	5.0	5.0	0.0	5.0	5.0	5.0	0.0	0.0
2	5.0	5.0	5.0	5.0	0.0	5.0	5.0	5.0	5.0	0.0
3	0.0	3.0	0.0	0.0	3.0	0.0	0.0	0.0	0.0	3.0
4	5.0	0.0	0.0	0.0	0.0	5.0	5.0	5.0	4.0	0.0
5	5.0	4.0	0.0	0.0	0.0	5.0	4.0	5.0	5.0	0.0
6	5.0	0.0	5.0	5.0	0.0	5.0	0.0	5.0	5.0	0.0
7	0.0	5.0	5.0	5.0	0.0	5.0	5.0	0.0	5.0	0.0
8	5.0	0.0	0.0	0.0	0.0	5.0	5.0	5.0	0.0	0.0
9	5.0	5.0	0.0	0.0	0.0	5.0	4.0	5.0	0.0	0.0

	0	1	2	3	4	5	6	7	8	9
0	2.0	5.0	5.0	5.0	1.0	5.0	1.0	2.0	1.0	1.0
1	5.0	5.0	5.0	5.0	**3.0**	5.0	5.0	5.0	**5.0**	**3.0**
2	5.0	5.0	5.0	5.0	**3.0**	5.0	5.0	5.0	5.0	**3.0**
3	**4.0**	3.0	**3.0**	**3.0**	3.0	**3.0**	**4.0**	**4.0**	**4.0**	3.0
4	5.0	**5.0**	**5.0**	**5.0**	**3.0**	5.0	5.0	5.0	5.0	**3.0**
5	5.0	4.0	**5.0**	**5.0**	**3.0**	5.0	5.0	5.0	5.0	**3.0**
6	5.0	**5.0**	5.0	5.0	**3.0**	5.0	**5.0**	5.0	5.0	**3.0**
7	**5.0**	5.0	5.0	5.0	**3.0**	5.0	5.0	**5.0**	5.0	**3.0**
8	5.0	**5.0**	**5.0**	**5.0**	**3.0**	5.0	5.0	5.0	**5.0**	**3.0**
9	5.0	5.0	**5.0**	**5.0**	**3.0**	5.0	4.0	5.0	**4.0**	**3.0**