Mining massive Datasets WS 2017/18

Problem Set 8

Rudolf Chrispens, Marvin Klaus, Daniela Schacherer

December 18, 2017

Exercise 01

a) The probability that C1 and C2 are matched?
QuastionnairesCount = b ($Q_1, \dots, Q_b$)
QuestionCount = r
QuestionCountSum = $b * r$
Propability for one Questionair that it matsches: p^r
Probability for one Questionair that it doesnt match: $(1-p)^r$
Probability that at least one Q. Matches: $1-(1-p)^r$

b) The probability that exactly two (no matter which) questionnaires match, i.e. have the same answers for both C1 and C2.
binomial coefficient
formula:
$\binom{b}{2} * (p^r)^2 * (1-(p^r))^{b-2}$

Exercise 02

siehe Ex8_2_3

Exercise 03

siehe Ex8_2_3

Exercise 04

- BALANCE Algorithm:
  For each query, assign it to an advertiser with the largest unspent budget (i.e. largest BALANCE) A(x,y) B(x,z)
  Worst Case Szenarios:

a) xyyy - AA__ (since there are 3 y no optimal solution possible, A has only 2 dollars.)

b) xyyx - AA_B (optimal solution is possible but not certain BAAB)

c) yyxx - AABB (Optimal)

d) xzyz - BBA_ (optimal solution is possible but not certain ABAB)

- Since we used the worst case possible query assignment with the balance Algorithm we can show that only c) would give an optimal solution. All the other queries have situations where an optimal solution is not certain. c's optimal solution is because A and B cant steal from each other, because their budget is empty before the other can take anything from their common x.

Exercise 05

- budget: 3 / ties in favor of lower index
- $A_1(Q_1,Q_2)$ $A_2(Q_2,Q_3)$ $A_3(Q_3,Q_4)$ $A_4(Q_1,Q_4)$
- Query: $Q_1, Q_2, Q_3, Q_3, Q_1, Q_2, Q_3, Q_1, Q_4, Q_1, Q_4$

a) What is the sequence of advertisers that the BALANCE algorithm will yield?
A_1,A_2,A_3,A_2,A_4,A_1,A_3,A_4,A_3,A_1,A_4
What is the competitive ratio for this instance?
$Competitive\ ratio = min_(all\ possible\ inputs)\frac{|M_(greedy)|}{(|M_(opt|))})$
$\frac{11}{11} = 1$

b) Rearrange the sequence of queries so that BALANCE results in a worse competitive ratio.
Query: $Q_1, Q_1, Q_1, Q_1, Q_4, Q_4, Q_2, Q_2, Q_3, Q_3, Q_3$
A_1,A_4,A_1,A_4,A_3,A_3,A_2,A_2,A_2,A_3, __
$\frac{10}{11}$ = 0,9090909091

Exercise 06

siehe Ex8_6.py

a) see *Ex8_6a.py*

b) see *Ex8_6b.py*

I had a problem while overwriting the old data. The error code was:

```
java.io.FileNotFoundException: File file:/home/immd-user/IdeaProjects/immd-
    ↪ project-example/timeseries/c4large_LinuxUNIX_ap-northeast-2c/part
    ↪ -00000-2590b8cb-3bc6-4592-a5c2-fc22a6742ef2-c000.csv does not exist
```

```
It is possible the underlying files have been updated. You can explicitly
    ↪ invalidate the cache in Spark by running 'REFRESH TABLE tableName'
    ↪ command in SQL or by recreating the Dataset/DataFrame involved.
```

I guess spark had problems with my filesystem (maybe access rights), but I don't know exactly. The solution was to write in a temp folder, delete the old one and rename the temp folder.

c) see *Ex8_6c.py*

Exercise 07

Since the beginning every bidder has the same budget and the same bids, value for $\Psi_i(q) = x_i(1-e^{-f_i})$ is equal across all bidders. Thats why the algorithm makes the decision randomly and the chosen bidders i budget will be reduced by one. It's $-f_i$ will also decrease. As x_i will never change this bidder will be unfavored for the next step. The conclusion is that since the bidding is the same the algorithm will perform as the basic balance algorithm.