

Mining massive Datasets WS 2017/18

Problem Set 9

Rudolf Chrispens, Marvin Klaus, Daniela Schacherer

January 8, 2018

Exercise 01

- a) If the support threshold is 5, all elements having at least 5 multiples ≤ 100 are considered frequent. Thus element 1 to 20 are frequent.
- b) Pairs of items can only be frequent if the single items are frequent as well. Thus we only have to consider the elements 1 to 20. If one element is a multiple of the other, then the pair is frequent. For two elements having no common divisor we consider the least common multiple of the two elements and check whether this number has at least five multiples ≤ 100 . Frequent pairs are according to this:

$\{1, 2\}, \{1, 3\}, \{1, 4\}, \dots, \{1, 20\}$
 $\{2, 3\}, \{2, 4\}, \{2, 5\}, \{2, 6\}, \{2, 7\}, \{2, 8\}, \{2, 9\}, \{2, 10\}, \{2, 12\}, \{2, 14\},$
 $\{2, 16\}, \{2, 18\}, \{2, 20\}$
 $\{3, 4\}, \{3, 5\}, \{3, 6\}, \{3, 7\}, \{3, 8\}, \{3, 9\}, \{3, 12\}, \{3, 15\}, \{3, 18\}$
 $\{4, 5\}, \{4, 6\}, \{4, 8\}, \{4, 10\}, \{4, 12\}, \{4, 16\}, \{4, 20\}$
 $\{5, 10\}, \{5, 15\}, \{5, 20\}$
 $\{6, 12\}$
 $\{7, 14\}$
 $\{8, 16\}$
 $\{9, 18\}$
 $\{10, 20\}$

- c) Let n be the total number of elements, in our case $n = 100$, and s be the size of all baskets. Then s can be estimated by the following formula

$$\begin{aligned} s &= n + \frac{1}{2}n + \frac{1}{3}n + \dots + \frac{1}{100}n \\ &= n\left(1 + \frac{1}{2} + \dots + \frac{1}{100}\right) \\ &= n \ln(n) \end{aligned}$$

d) The confidence of an association rule $I \rightarrow j$ is given by

$$conf(I \rightarrow j) = \frac{support(I \cup j)}{support(I)}$$

- Thus for $\{5, 7\} \rightarrow 2$ we have $\{5, 7\}$ appearing in 2 baskets (35 and 70) and $\{5, 7, 2\}$ appearing in 1 basket (70). Thus we yield: $conf = \frac{1}{2}$
- $\{2, 3, 4\}$ appear in baskets 12, 24, 36, 48, 60, 72, 84, 96, while $\{2, 3, 4, 5\}$ appear in basket 60. We yield: $conf = \frac{1}{8}$.

Exercise 02

Hypothetical steps of the A-priori algorithm on the setup of exercise 1

- $C1 = \{\{1\}, \{2\}, \dots, \{100\}\}$
- Count the support of itemsets in C1
- Prune non-frequent: $L1 = \{\{21\}, \{22\}, \dots, \{100\}\}$
- Generate $C2 =$ see Exercise 1 b)
- Count the support of itemsets in C2
- Prune non-frequent: $L2 =$ alle Kombinationen der Zahlen 1 bis 20, die nicht in Exercise 1 b) gelistet wurden
- Generate $C3 =$ Kombination aller Elemente in C2 mit den Zahlen 1 bis 20
- Count the support of itemsets in C3
- Prune non-frequent: L3

Exercise 03

$\{1, 2, 3\}$ $\{2, 3, 4\}$ $\{3, 4, 5\}$ $\{4, 5, 6\}$
 $\{1, 3, 5\}$ $\{2, 4, 6\}$ $\{1, 3, 4\}$ $\{2, 4, 5\}$
 $\{3, 5, 6\}$ $\{1, 2, 4\}$ $\{2, 3, 5\}$ $\{3, 4, 6\}$

Threshold: 4 PCY Algorithm we use a hash table with 11 buckets

I just did all calculations per hand and used the PCY Algorithm step by step without code.

- c) 4 and 8 are Frequent, because their Value exceeds our threshold of 4 and is most of all the others.
1 and 2 are Frequent as well because they exceed 4 as well.
- d) The count of each pair element in the Bucket.

Table 1 – 3a) Items						
Items	1	2	3	4	5	6
Support	4	6	8	8	6	4

Table 2 – 3a,b) Itempairs		
Bucket	Pairs	Support
2,3,6	(1,2)(1,3)(2,3)	2/3/3
6,8,1	(2,3)(2,4)(3,4)	3/4/4
1,4,9	(3,4)(3,5)(4,5)	4/4/3
9,2,8	(4,5)(4,6)(5,6)	3/3/2
3,5,4	(1,3)(1,5)(3,5)	3/1/4
8,1,2	(2,4)(2,6)(4,6)	4/1/3
3,4,1	(1,3)(1,4)(3,4)	3/2/4
8,10,9	(2,4)(2,5)(4,5)	4/2/3
4,7,8	(3,5)(3,6)(5,6)	4/2/2
2,4,8	(1,2)(1,4)(2,4)	2/2/4
6,10,4	(2,3)(2,5)(3,5)	3/2/4
1,7,2	(3,4)(3,6)(4,6)	4/2/3

Exercise 04



a) The total number of pairs is $n(n-1)/2$ where n is the total number of items (I). Since four bytes needed for every element the total space needed is 4 times the number of pairs. Approximately by $2n^2$.



b) Since just pairs with $i < j$ are stored and every pair can be nonzero, the maximal number of nonzero pairs is the number of pairs stored $n(n-1)/2$.

c) The triples method use tree times the memory for a pair, where the count is greater than zero. This means that the triple method will use less space if there are less then $1/3$ pairs with a count greater than 0.

Table 3 – 3b) Hashbucket											
Bucket	1	2	3	4	5	6	7	8	9	10	11
Val	5	5	3	6	1	3	2	6	3	2	0
Pairs	(3,4) (2,6)	(1,2) (4,6)	(1,3)	(1,4) (3,5)	(1,5)	(2,3)	(3,6)	(2,4) (5,6)	(4,5)	(2,5)	