University of Augsburg, Institute of Computer Science
Prof. Dr. P. Fischer
J. Kastner, L. Rudenko

SS 2019
15. July 2019
Solution 10

Analyzing Massive Data Sets

Exercise 1: Effectiveness Metrics (homework)

a) Fallout: $F_Q = \frac{|\bar{A} \cap B|}{\bar{A}} = \frac{\#non-relevant\ documents\ retrieved}{\#non-relevant\ documents} = \frac{150-120}{810+(150-120)} = \frac{30}{840} = \frac{1}{28} = 0,036$

b) F Measure $F_\beta = \frac{(\beta^2+1)RP}{R+\beta^2 P}$. We have $F_1 = \frac{(1^2+1)*R*0,5}{R+1^2*0,5} = 0,5$.

$$\frac{R}{R+0,5} = 0,5 \qquad 0,5R = 0,25 \qquad R = 0,5$$

Exercise 2: nDCG (homework)

$DCG@k = rel_1 + \sum_{i=2}^{p} \frac{rel_i}{log_2 i}$ $\qquad$ $nDCG@k = \frac{DCG@k}{iDCG@k}$

2, 0, 1, 3, 2, 2, 0, 1, 1, 3, 2, 0

- $DCG_1 =$ **2**
- $DCG_2 = 2 + 0/\log_2 2 = 2 + 0 =$ **2**
- $DCG_3 = 2 + 0/\log_2 2 + 1/\log_2 3 = 2 + 0 + 0,63 =$ **2,63**
- $DCG_4 = 2 + 0/\log_2 2 + 1/\log_2 3 + 3/\log_2 4 = 2 + 0 + 0,63 + 1,5 =$ **4,13**
- $DCG_5 = 2+0/\log_2 2+1/\log_2 3+3/\log_2 4+2/\log_2 5 = 2+0+0,63+1,5+0,86 =$ **4,99**
- $DCG_6 = 2+0/\log_2 2+1/\log_2 3+3/\log_2 4+2/\log_2 5+2/\log 26 = 2+0+0,63+1,5+0,86+0,77 =$ **5,76**
- $DCG_7 = 2 + 0/\log_2 2 + 1/\log_2 3 + 3/\log_2 4 + 2/\log_2 5 + 2/\log 26 + 0/\log_2 7 = 2 + 0 + 0,63 + 1,5 + 0,86 + 0,77 + 0 =$ **5,76**
- $DCG_8 = 2+0/\log_2 2+1/\log_2 3+3/\log_2 4+2/\log_2 5+2/\log 26+0/\log_2 7+1/\log_2 8 = 2 + 0 + 0,63 + 1,5 + 0,86 + 0,77 + 0 + 0,33 =$ **6,09**
- $DCG_9 = 2+0/\log_2 2+1/\log_2 3+3/\log_2 4+2/\log_2 5+2/\log 26+0/\log_2 7+1/\log_2 8+1/\log_2 9 = 2 + 0 + 0,63 + 1,5 + 0,86 + 0,77 + 0 + 0,33 + 0,32 =$ **6,41**
- $DCG_{10} = 2+0/\log_2 2+1/\log_2 3+3/\log_2 4+2/\log_2 5+2/\log 26+0/\log_2 7+1/\log_2 8+1/\log_2 9+3/\log_2 10 = 2 + 0 + 0,63 + 1,5 + 0,86 + 0,77 + 0 + 0,33 + 0,32 + 0,9 =$ **7,31**
- $DCG_{11} = 2+0/\log_2 2+1/\log_2 3+3/\log_2 4+2/\log_2 5+2/\log 26+0/\log_2 7+1/\log_2 8+1/\log_2 9+3/\log_2 10+2/\log_2 11 = 2+0+0,63+1,5+0,86+0,77+0+0,33+0,32+0,9+0,58 =$ **7,89**
- $DCG_{12} = 2+0/\log_2 2+1/\log_2 3+3/\log_2 4+2/\log_2 5+2/\log 26+0/\log_2 7+1/\log_2 8+1/\log_2 9+3/\log_2 10+2/\log_2 11+0/\log_2 12 = 2+0+0,63+1,5+0,86+0,77+0+0,33+0,32+0,9+0,58+0 =$ **7,89**

$iDCG@k$ refers to **perfect ranking list**:

3, 3, 2, 2, 2, 2, 1, 1, 1, 0, 0, 0

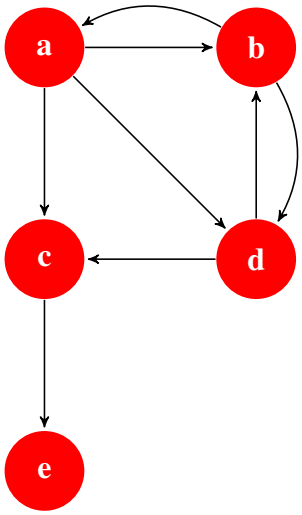
- $IDCG_1 =$ **3**
- $IDCG_2 = 3 + 3/\log_2 2 = 3 + 3 =$ **6**
- $IDCG_3 = 3 + 3/\log_2 2 + 2/\log_2 3 = 3 + 3 + 1,26 =$ **7,26**
- $IDCG_4 = 3 + 3/\log_2 2 + 2/\log_2 3 + 2/\log_2 4 = 3 + 3 + 1,26 + 1 =$ **8,26**
- $IDCG_5 = 3+3/\log_2 2+2/\log_2 3+2/\log_2 4+2/\log_2 5 = 3+3+1,26+1+0,86 =$ **9,12**
- $IDCG_6 = 3+3/\log_2 2+2/\log_2 3+2/\log_2 4+2/\log_2 5+2/\log_2 6 = 3+3+1,26+1+0,86+0,77 =$ **9,89**
- $IDCG_7 = 3+3/\log_2 2+2/\log_2 3+2/\log_2 4+2/\log_2 5+2/\log_2 6+1/\log_2 7 = 3+3+1,26+1+0,86+0,77+0,36 =$ **10,25**
- $IDCG_8 = 3+3/\log_2 2+2/\log_2 3+2/\log_2 4+2/\log_2 5+2/\log_2 6+1/\log_2 7+1/\log_2 8 = 3+3+1,26+1+0,86+0,77+0,36+0,33 =$ **10,58**
- $IDCG_9 = 3+3/\log_2 2+2/\log_2 3+2/\log_2 4+2/\log_2 5+2/\log_2 6+1/\log_2 7+1/\log_2 8+1/\log_2 9 = 3+3+1,26+1+0,86+0,77+0,36+0,33+0,32 =$ **10,9**
- $IDCG_{10} = 3+3/\log_2 2+2/\log_2 3+2/\log_2 4+2/\log_2 5+2/\log_2 6+1/\log_2 7+1/\log_2 8+1/\log_2 9+0/\log_2 10 = 3+3+1,26+1+0,86+0,77+0,36+0,33+0,32+0 =$ **10,9**
- $IDCG_{11} = 3+3/\log_2 2+2/\log_2 3+2/\log_2 4+2/\log_2 5+2/\log_2 6+1/\log_2 7+1/\log_2 8+1/\log_2 9+0/\log_2 10+0/\log_2 11 = 3+3+1,26+1+0,86+0,77+0,36+0,33+0,32+0+0 =$ **10,9**
- $IDCG_{12} = 3+3/\log_2 2+2/\log_2 3+2/\log_2 4+2/\log_2 5+2/\log_2 6+1/\log_2 7+1/\log_2 8+1/\log_2 9+0/\log_2 10+0/\log_2 11+0/\log_2 12 = 3+3+1,26+1+0,86+0,77+0,36+0,33+0,32+0+0+0 =$ **10,9**

- $NDCG_1 = 2/3 = 0,67$
- $NDCG_2 = 2/6 = 0,33$
- $NDCG_3 = 2,63/7,26 = 0,36$
- $NDCG_4 = 4,13/8,26 = 0,5$
- $NDCG_5 = 4,99/9,12 = 0,55$
- $NDCG_6 = 5,76/9,89 = 0,58$
- $NDCG_7 = 5,76/10,25 = 0,56$
- $NDCG_8 = 6,09/10,58 = 0,58$
- $NDCG_9 = 6,41/10,9 = 0,59$
- $NDCG_{10} = 7,31/10,9 = 0,67$
- $NDCG_{11} = 7,89/10,9 = 0,72$
- $NDCG_{12} = 7,89/10,9 = 0,72$

nDCD values: **0.67, 0.33, 0.36, 0.5, 0.55, 0.58, 0.56, 0.58, 0.59, 0.67, 0.72, 0.72**

Exercise 3: Transition Matrices (homework)

The following graph G is given:



a) Specify the compact representation **transition matrix** for the graph G.

source	degree	destination
a	3	b, c, d
b	2	a, d
c	1	e
d	2	b, c
e	0	–

b) Assume only the blocks of size $s = 2$ fit into main memory. Specify the compact representation **transition matrix** for the graph G with this restriction.

- The first block contains nodes a and b:

source	degree	destination
a	3	b
b	2	a
d	2	b

- The second block contains nodes c and d:

source	degree	destination
a	3	c, d
b	2	d
d	2	c

- The third block contains the node e:

source	degree	destination
c	1	e

Exercise 4: TrustRank (live)

The solution was discussed in the exercise.

Exercise 5: PageRank with MapReduce (homework)

```
from mrsim import mr_simulator

nodes = [(('a', 0.2), ['a','c']),
         (('b', 0.2), ['a','d']),
         (('c', 0.2), ['b','c','d']),
         (('d', 0.2), ['c','e']),
         (('e', 0.2), [])]

# this can be done in another map-reduce steps
# - collect all vertex lables
# - add all graph vertices to adjacency list of dead ends

def remove_deadends(nodes):
    vertex_ids = [v[0][0] for v in nodes]

    for v in nodes:
        adjacency_list = v[1]
        if not adjacency_list: # adjecency list is empty
            adjacency_list.extend(vertex_ids)

    return nodes


def map(key, val):
    res = []
    for link in val:
        res.append((link, ('val', key[1]/len(val))))
    res.append((key[0], ('link', val)))
    return res

def reduce(key, val):

    res = []
    pr_value = 0.0
    links = ()

    for v in val:
        if(v[0] == 'val'):
            pr_value = pr_value + v[1]
        else:
            links = v[1]
    pr_value = 0.8*pr_value + (0.2)/len(nodes)

    res.append(((key, pr_value), links))
    return res

intermediateResult = remove_deadends(nodes)

#first 10 iteration
for i in range(1, 11):
    print('Iteration ' +  str(i) + ': ')
    intermediateResult = mr_simulator(intermediateResult, map, reduce)
    print(intermediateResult)
```

Exercise 6: HITS: Hubs and Authorities (live)

The solution was discussed in the exercise.