COMP9313 2017s1 Assignment

Question 1. MapReduce (4 pts)

The following code of for computing the relative frequency is problematic. Describe how you can fix it.

```
class Mapper
        method Map(docid a, doc d)
                for all term w in doc d do
                        for all term u in Neighbors(w) do
                                Emit(pair (w, u), count 1)
                                Emit(pair (w, *), count 1)
class Reducer
        curMarginal <- 0
        method Reduce(pair p, counts [c1, c2, …])
                s <- 0
                for all count c in counts [c1, c2, …] do
                        s <- s + c
                if(p.contains(*))
                        Emit(p, s/curMarginal)
                Else
                        curMarginal <- s
```

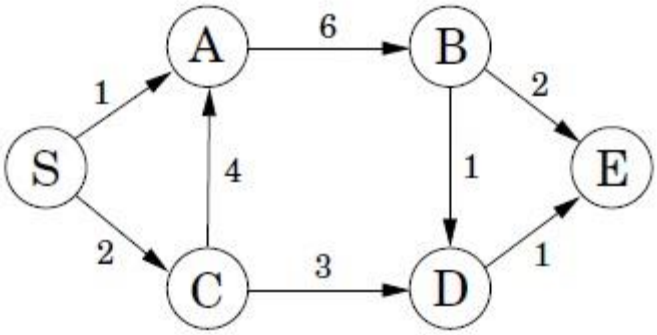
Answer:

1. A Partitioner is required to guarantee that the key-value pairs relevant to the same term w are sent to the same reducer.

2. “p.contains(*)” should be “!p.contains(*)”

Question 2. Graph Algorithms (6 pts)

Given the following graph, assume that you are using the single shortest path algorithm to compute the shortest path from node S to node E. Show the output of the mapper (sorted results of all mappers) and the reducer (only one reducer used) in each iteration (including both the distances and the paths).



Answer:

1.

Mapper:

(A, 1), (C, 2)

Reducer:

A: 1 | S->A | B:6

C: 2 | S->C | A:4, D:3

2.

Mapper:

(B, 7), (A, 6), (D, 5)

Reducer:

B: 7 | S->A->B | D:1, E:2

D: 5 | S->C->D | E:1

3.

Mapper:

(E, 9), (D, 8), (E, 6)

Reducer:

E: 6 |S->C->D->E | empty

Algorithm terminates

Question 3. Streaming Data Processing (5 pts)

Suppose we are maintaining a count of 1s using the DGIM method. We represent a bucket by (i, t), where i is the number of 1s in the bucket and t is the bucket timestamp (time of the most recent 1).

Consider that the current time is 200, window size is 60, and the current list of buckets is: (16, 148) (8, 162) (8, 177) (4, 183) (2, 192) (1, 197) (1, 200). At the next ten clocks, 201 through 210, the stream has 0101010101. What will the sequence of buckets be at the end of these ten inputs?

Answer:

There are 5 1s in the stream. Each one will update to windows to be: [each step 2 marks]

(1) (16, 148)(8, 162)(8, 177)(4, 183)(2, 192)(1, 197)(1, 200), (1, 202)

=> (16, 148)(8, 162)(8, 177)(4, 183)(2, 192)(2, 200), (1, 202)

(2) (16, 148)(8, 162)(8, 177)(4, 183)(2, 192)(2, 200), (1, 202), (1, 204)

(3) (16, 148)(8, 162)(8, 177)(4, 183)(2, 192)(2, 200), (1, 202), (1, 204), (1; 206)

=> (16, 148)(8, 162)(8, 177)(4, 183)(2, 192)(2, 200), (2, 204), (1, 206)

=> (16, 148)(8, 162)(8, 177)(4, 183)(4, 200), (2, 204), (1, 206)

(4) Windows Size is 60, so (16,148) should be dropped.

(16, 148)(8, 162)(8, 177)(4, 183)(4, 200), (2, 204), (1, 206), (1, 208) => (8, 162)(8, 177)(4, 183)(4, 200), (2, 204), (1, 206), (1, 208)

(5) (8, 162)(8, 177)(4, 183)(4, 200), (2, 204), (1, 206), (1, 208), (1, 210)

=> (8, 162)(8, 177)(4, 183)(4, 200), (2, 204), (2, 208), (1, 210)

Question 4. Recommender Systems (5 pts)

Consider three users u_1, u_2, and u_3, and four movies m_1, m_2, m_3, and m_4. The users rated the movies using a 4-point scale: -1: bad, 1: fair, 2: good, and 3: great. A rating of 0 means that the user did not rate the movie.

The three users' ratings for the four movies are: u_1 = (3, 0, 0, -1), u_2 = (2, -1, 0, 3), u_3 = (3, 0, 3, 1)

(i) (3 pts) Which user has more similar taste to u_1 based on cosine similarity, u_2 or u_3? Show detailed calculation process.

Answer: sim(u1, u2) = (3*2 -1*3)/(sqrt(10)*sqrt(14)) ≈ 0.2535, sim(u1, u3) = (3*3-1*1)/(sqrt(10)*sqrt(19)) ≈ 0.5804. Thus u3 is more similar to u1.

(ii) (2 pts) User u_1 has not yet watched movies m_2 and m_3. Which movie(s) are you going to recommend to user u_1, based on the user-based collaborative filtering approach? Justify your answer.

Answer: You can use either cosine similarity or Pearson correlation coefficient to compute the similarities between users. However, the conclusion should be that only m3 is recommended to u1.