EE412 Foundation of Big Data Analytics, Fall 2018

HW1

Name: Park Jaeyoung

Student ID: 20170273

Discussion Group (People with whom you discussed ideas used in your answers):

On-line or hardcopy documents used as part of your answers:

Answer to Problem 1

(a) Solve the following problems which are based on the exercises in the MMDS 2nd edition textbook.

Exercise 2.2.1

(a) The skew might appear as number of words appeared in input data are different. Some words such as "the" would appear much more than others while some words appear once. This will causes time difference among various reducers.

(b) The skewness cannot be solved while 10 tasks are randomly selected. To reduce the skew, tasks are needed to select by length of value-list: to make sure none of reduce task has much larger value-list than others.

Combining the reducers into 10,000 reduce tasks will also cause problem. First, there is overhead associated with each map task creates. Also, this way is not solving length-problem which occurs in 10 tasks.

(c) Only using combiner at the map might speed up the whole speed, but doesn't reduce skewness of the process. As hashing method of the process doesn't changes, the amount of skewness remains.

Exercise 2.3.3

(a) <Bag Union>

The Map Function:

For u in R:

Construct (u, 1)

For v in S:

Construct (v, 1)

The Reduce Function:

For each keys produced by map function:

Construct (t, n+m)

(b) <Bag Intersection>

The Map Function:

For u in R:

If u in S:

Construct (u, 1)

For v in S:

If v in R:

Construct (v, 1)

First Reduce Function:

Turns (u, [1, 1, … 1]) into (u, n) – while n is number of 1 in the list

Turns (v, [1, 1, … 1]) into (v, m) – while m is number of 1 in the list

Second Reduce Function:

For each key t produced by previous process:

Construct (t, minimum(n, m))

(c) <Bag Difference> [(ex) R – S]

The Map Function:

For u in R:

  Construct (u, 1)

For v in S:

  If v in R:

    Construct (v, -1)

The Reduce Function:

For each keys produced by map function:

  Construct (t, n+m)

Example 2.4.1

answer) n*(t+9*p*t^2)/(1-p*t)

solution)

let n-tasks expected time is T(n).

then

T(n)

= (1-pt)(T(n-1) + t)   --- in case last task didn't fail

+ pt(T(n) + 10t) ------- in case last task failed

and T(0) = 0.

By solving this recurrence formula, T(n) can be calculated.

(b) Find potential friends in a social network using Spark.

18667 18672 84

18667 18675 83

18672 18677 83

| | | |
|---|---|---|
| 18672 | 18678 | 83 |
| 18667 | 18677 | 82 |
| 18675 | 18677 | 82 |
| 31490 | 31496 | 82 |
| 31491 | 31496 | 82 |
| 18667 | 18678 | 81 |
| 18675 | 18678 | 81 |

Code is in attached file.

---

Answer to Problem 2

---

(a) Solve the following problems which are based on the exercises in the MMDS textbook.

Exercise 6.1.1

(a) Number from 1 to 20.

(b) Any pair from bag 1 to 20.

Ex) from bag 4 = {1, 2, 4}, "(1, 2), (1, 4), (2, 4)" are produced

(c) Sum of number of divisor from 1 to 100. :482

Exercise 6.2.3

(a) I*(I-1)/2 * 4(bytes) = 2*I*(I-1)

About 2I^2 bytes

(b) Min(B*k*(k-1)/2, I*(I-1)/2)

First one choosing 2 elements from each basket, but if this might count repeating pair.

(c) Triple method spends 12 bytes per pair while triangular array spends 4 bytes per pair.

If 3 * Number of pair (or largest possible number of pairs, B*k*(k-1)/2) is smaller than possible pairs (2*I*(I-1) or approximately 2I^2), then triple method will use less space.

Exercise 6.2.7

Memory needed for first pass: 4 million bytes for 1 million items.

Memory needed for second pass:

1) 4*N bytes to store the ID of frequent items.
2) Memory needed for triangular or triple method(smaller one would be selected)
   A. Triangular table: 2N^2 (4 * n(n-1)/2)
   B. Triples: 12*10^6 + M (12 * 10^6 for frequent pairs and M for non-frequent but each item is frequent)

Total occupied memory is

4 million + 4N + min(2N^2, 12*10^6 + M) bytes

(b) Find frequent itemsets using the A-Priori algorithm

number of frequent items: 363

number of frequent pairs: 326

| | | |
|---|---|---|
| DAI62779 | ELE17451 | 1592 |
| FRO40251 | SNA80324 | 1412 |
| DAI75645 | FRO40251 | 1254 |
| FRO40251 | GRO85051 | 1213 |
| DAI62779 | GRO73461 | 1139 |
| DAI75645 | SNA80324 | 1130 |
| DAI62779 | FRO40251 | 1070 |
| DAI62779 | SNA80324 | 923 |
| DAI62779 | DAI85309 | 918 |

ELE32164 GRO59710 911

Code is in attached file.

Answer to Problem 3

(a) Solve the following problems which are based on the exercises in the MMDS textbook.

Exercise 3.3.2

| Row | 2x + 4 mod 5 | 3x - 1 mod 5 |
|---|---|---|
| 0 | 4 | 4 |
| 1 | 1 | 2 |
| 2 | 3 | 0 |
| 3 | 0 | 3 |
| 4 | 2 | 1 |

Exercise 3.4.2

| (r, b) | 1 – (1 – s^r)^b = 1/2 | (1/b)^(1/r) |
|---|---|---|
| (3, 10) | 0.406 | 0.464 |
| (6, 20) | 0.569 | 0.607 |
| (5, 50) | 0.424 | 0.457 |

Exercise 3.6.1

(a)

Probability p converts to 1 – (1-p^2)^3.

Function takes 6 times more than original.

Amplification is possible if low probability is below p and high probability is above p since p is solution for “p = 1 – (1-p^2)^3 = 0.389”.

Reduce both the false negative and false positive rates if amplification is possible.

(b)

Probability p converts to (1 – (1-p)^3)^2.

Function takes 6 times more than original.

Amplification is possible if low probability is below p and high probability is above p since p is solution for “p = (1 – (1-p)^3)^2= 0.152”.

Reduce both the false negative and false positive rates if amplification is possible.

(c)

Probability p converts to (1 – (1-p^2)^2)^2.

Function takes 8 times more than original.

Amplification is possible if low probability is below p and high probability is above p since p is solution for “p = (1 – (1-p^2)^2)^2= 0.847”.

Reduce both the false negative and false positive rates if amplification is possible.

(c)

Probability p converts to (1 – (1 - (1 – (1-p)^2)^2)^2)^2.

Function takes 16 times more than original.

Amplification is possible if low probability is below p and high probability is above p since p is solution for “p = (1 – (1 - (1 – (1-p)^2)^2)^2)^2= 0.382”.

Reduce both the false negative and false positive rates if amplification is possible.

(b) Find similar documents using minhash-based LSH

| | | |
|---|---|---|
| t1621 | t7958 | 1.0000 |
| t448 | t8535 | 1.0000 |
| t269 | t8413 | 1.0000 |
| t3268 | t7998 | 0.9917 |
| t2023 | t980 | 0.9917 |

Code is in attached file.