QUIZ-4 SOLUTIONS

[1.5] [1.5]

BUCKETS	HASH FUNC 1 (i+j mod 3) ①	HASH FUNC 2 (j mod 3) ②	Counts ①	Counts ②	Bitmap ①	Bitmap ②
0	(1,2) (1,5) (2,4) (4,5)	(1,3) (1,3) (2,3) (2,3)	4	4	1	1
1	(1,3) (3,4) (1,3) (3,4)	(2,4) (3,4) (3,4)	4	3	1	1
2	(2,3) (3,5) (2,3) (3,5)	(1,2) (3,5) (3,5) (4,5) (1,5)	4	5	1	1

Multihash

1ST PASS [0.75]

item counts:

1 => 2 , 2 => 2 , 3 => 4 , 4 => 2 , 5 => 2

Buckets	Hash ①	Hash ②
0	4	4
1	4	3
2	4	5

2ND PASS [0.75]

Freq items: 1, 2, 3, 4, 5

BITMAP ① :

B0	B1	B2
1	1	1

BITMAP ② :

B0	B1	B2
1	1	1

Candidate item pairs :

{1,2} {1,3} {1,5} {2,3} {2,4} {3,4} {3,5} {4,5}

Counts of pairs

{1,2} => 1 {1,3} => 2 {1,5} => 1 {2,3} => 2 {2,3} => 1 {3,4} => 2 {3,5} => 2 {4,5} => 1

∴ Frequent Pairs :- {1,3}, {2,3} {3,4}, {3,5}

MULTI STAGE

1st PASS [0.5 POINTS]

item counts :

1 => 2 , 2 => 2 , 3 => 4 , 4 => 2 , 5 => 2

- HASH ①

BUCKET	COUNT
0	4
1	4
2	4

2ND PASS [0.5 POINTS]

frequent items : 1, 2, 3, 4, 5

BITMAP ① :

B0	B1	B2
1	1	1

HASH ②

BUCKET	COUNT
0	4
1	3
2	5

3RD PASS [0.5 POINTS]

frequent items : 1, 2, 3, 4, 5

BITMAP ②

B0	B1	B2
1	1	1

Count of pairs :

{1,2} => 1 , {1,3} = 2 , {1,5} => 1 , {2,3} => 2 , {2,4} => 1 ,

{3,4} => 2 , {3,5} => 2 , {4,5} = 1

Frequent Pairs : {1,3}, {2,3} , {3,4}, {3,5}

2) General idea - [0.5]

For some applications, it is sufficient to discover most frequent itemsets and is not essential ~~every~~ to discover every single one.

Pros: less I/O cost, time, passes. [0.25]

Cons: False positive, false Negative are induced in the results. [0.25]

3) PHASE 1 MAP i/p :-

[0.25] a chunk/subset of all baskets (sample of i/p file)

PHASE 1 MAP o/p :-

set of pairs (f, 1) where f is a frequent itemset from sample.

PHASE 1 REDUCE i/p :-

[0.25] set of pairs (f, 1)

PHASE 1 REDUCE o/p :-

Candidate itemsets

PHASE 2 MAP i/p :-

[0.25] Result from phase1 and total output file

PHASE 2 MAP o/p :-

Set of pairs (C, V), C is candidate itemsets, V is the support for that itemset.

PHASE 2 Reduce i/p

[0.25] (C, V)

PHASE 2 REDUCE o/p :-

if $V \geq s$, emit (C, V)

4) [0.25] False positives :- Infrequent in entire data, frequent in sample.

[0.25] False Negative :- Frequent in entire data, Infrequent in sample.

[0.5] Increasing Support : will increase FN as it will be harder to be frequent in the sample, decrease FP.

Decreasing Support : will induce more FP in the data as it is easier to be frequent in the data/sample decrease FNs.

5)

[0.25] Singleton : $\{a\}$ is in -ve border,
iff $\{a\}$ is not frequent in the sample

[0.25] Pair : $\{a, b\}$ is in -ve border,
iff $\{a, b\}$ is not frequent in the sample
$\{a\}, \{b\}$ are frequent

0.5 marks

① Construct sample data set
② Find candidates frequent itemsets from sample
③ Construct Negative border.
④ process the whole file,
if no itemset from -ve border turns out to be frequent in whole dataset, correct ✓
if some ...., Repeat the algo with random sample.