Name: Tak Yin Pang
Email: a1796036@student.adelaide.edu.au
ID:a1796036

Assignment - 3

Mining Big Data - COMP SCI 7306

Exercise-1:

1. Implement in Ex1.py as function random_sample.
2. Implement in Ex1.py as function son_algorithm.
3. And 4. Please see the complete result in google sheet link https://docs.google.com/spreadsheets/d/1g_Z64s1xdOUyagMGs76Xb_fS2Mamx34ZC4r3tBpXhs4/edit?usp=sharing

The experiment is run on my computer setting with 32GB Ram.
I will only show the result of the dataset T10I4D100K in the below.

Dataset = T10I4D100K threshold factor = 0.03, sample size factor = 0.01	Runtime(s)	Number of Basket	Supported Threshold	Number of Frequent Item	Frequent Item List
Apriori	225	100000	3000	60	[['12']
Ramdom Sample	2.3	1031	31	59	[['12']
SON Algorithm	36.6	1000	30	60	[['12']
Dataset = T10I4D100K threshold factor = 0.03, sample size factor = 0.02	Runtime(s)	Number of Basket	Supported Threshold	Number of Frequent Item	Frequent Item List
Apriori	225	100000	3000	60	[['12']
Ramdom Sample	4.63	1987	60	59	[['12']
SON Algorithm	61.57	2009	1507	60	[['12']
Dataset = T10I4D100K threshold factor = 0.03, sample size factor = 0.05	Runtime(s)	Number of Basket	Supported Threshold	Number of Frequent Item	Frequent Item List
Apriori	225	100000	3000	60	[['12']
Ramdom Sample	13.07	4965	149	67	[['12']
SON Algorithm	121.6	4981	149	60	[['12']
Dataset = T10I4D100K threshold factor = 0.03, sample size factor = 0.10	Runtime(s)	Number of Basket	Supported Threshold	Number of Frequent Item	Frequent Item List
Apriori	225	100000	3000	60	[['12']
Ramdom Sample	23.88	10025	301	60	[['12']
SON Algorithm	235.5	10022	302	60	[['12']

The left upper cell indicates the dataset used, threshold factor and sample size factor. Supported Threshold is calculated as Number of Basket * threshold factor. Random sample and SON algorithm are sampled using sample size factor. SON algorithm will do 10 RS in the first pass for all sample size factors for comparison convenience . Number of baskets and Supported Threshold of SON is the average value for the 10 runs of RS.

The table shows the runtime in second, Number of Basket, Supported Threshold and Number of frequent items. The last column is the frequent item list which is included in the google sheet link.

Observation:

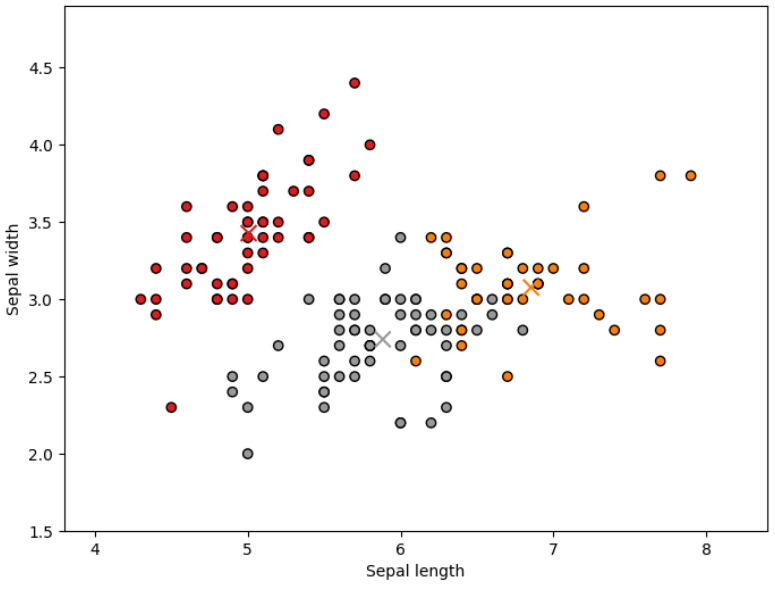
1. The SON algorithm can get the exact frequent item list as Apriori running on a full dataset.
2. The runtime of Random Sample is about the runtime of A-priori times the sample size factor.
3. The runtime of SON algorithm is about 10 times the runtime of Random Sample.
4. Random sample frequent item list tends to closer to true frequent item list as the sample size goes up.
5. The supported threshold is selected such that the number of frequent items is below 100.

Exercise-2:

1. [a1] denotes a1 is a cluster and a1 is the cluster mean.
   [a1,a2] - m denotes a1 and a2 is a cluster and m is the cluster mean.

step	Cluster
0	[1], [4], [9], [16], [25], [36], [49], [64], [81]
1	[1,4] - 2.5, [9], [16], [25], [36], [49], [64], [81]
2	[1,4,9] - 4.67, [16], [25], [36], [49], [64], [81]
3	[1,4,9] - 4.67, [16, 25] - 20.5, [36], [49], [64], [81]
4	[1,4,9] - 4.67, [16, 25] - 20.5, [36, 49] - 42.5, [64], [81]
5	[1,4,9, 16, 25] - 11, [36, 49] - 42.5, [64], [81]
6	[1,4,9, 16, 25] - 11, [36, 49] - 42.5, [64, 81] - 72.8
7	[1,4,9, 16, 25] - 11, [36, 49, 64, 81] - 57.5
8	[1,4,9, 16, 25, 36, 49, 64, 81] - 35.625

2. a) Please find the code in Ex2.py.



b) If we have prior knowledge about the query, we use the prior knowledge to determine k. In iris dataset, we already know there are 3 classes and therefore k = 3. Otherwise, we can use the elbow method to determine the values of k. We can plot the average squared distance between each data point and its centroid against the number of cluster k. The average distance decreases as k increases and there is a bend indicating a sharp decrease at one point and slowing down. The sharp turning point will be the number of cluster k that we pick.

Exercise 3

1. If the bidders sequence is as follows, CBCBzz.
   C bids on X, B bids on X, C bids on Y, B bids on Y.
   Then, zz will have no advertisers to bid on.
   It will assign at least 4 of the 6 queries xxyyzz for greedy algorithm.
2. Consider a sequence of queries xxzz, in worst case, the bidder C bids on the first two queries as CCzz, the last two queries zz will have no bidder. In the best case, the bidder sequence can be BBCC and all queries will be assigned.