RUTGERS

Implementation of Stream Clustering Algorithm

By Chaitanya Vallabhaneni,
Ishani Ghose &
Kaushik Vakadkar

Goal of our Project

- Implementation of stream clustering algorithms in Python.

- Analysis of its performance using the Sum of Squared Distances (SSQ) metric with weather sensor data to evaluate potential use of the algorithm in IOT devices.

Data

- Weather sensor data was used.
- The input is in the form of a log file with each data point having the following attributes: data, time, temperature(R), humidity(R), light(R), voltage(R), epoch(I) and moteid(I). R = Real number, I = Integer.
- The size of the data set is 150 MB. While the data is statically stored, a stream of data is generated from it.

Data

- The Stream_lsearch algorithm is run over sensor data by simulating a stream over 20,000 data points by taking chunks of 1000 data points running kmedians over it to search for k centres. The last 4 attributes of the raw data: Temperature, Humidity, Light and Voltage were used to form the clusters.
- As the different attributes lied in extremely different range, each value was normalized by the following formula: Norm_val = (val - min)/(max-min)
- The ranges of the 4 features are:
  - Temperature (-5,100)(degree celsius)
  - Humidity (0,100%)
  - Light: (0,100000)(flux)
  - Voltage (2,3)

Algorithms Implemented

- Stream LSEARCH: The implementation of the clustering subroutine (LSEARCH) used for a given set of points is available in C++. We have implement this in Python. This algorithm takes an upper and lower bound on the number of clusters that the algorithm can create and then performs a local search on it.

High Level Pseudocode of kmedains(kmax,kmin)

```
► Z = 0
► Z_max = summation (distance(x,x0)) where x0 is chosen as the first point in the chunk
► Z_min = 0
► Z = (zmax + zmin)/2
► I,a = Estimate an initial solution by calling function speedy2
► If number of centres identified < kmax: return solution
► Shuffle data
► (I,a) = Keep estimate a solution until number of centres is >= kmin
► Randomly pick theta((1/p)logk) points as median
► While k is outside the range(kmin, max) and zmax - zmin > 0.0001:
    ► Let (F, g) be the current solution
    ► Run F L(N, d, ,(F, g)) to obtain a new solution (F0 , g0 )
    ► If |F0 | is “about” k, then (F0 , g0 ) ← F L(N, d, 0 ,(F0 , g0 ))
    ►
        ► If |F0 | > k then {zmin ← z and z ← (zmax + zmin)/2};
        ►  else if |F0 | > k then {zmax ← z and z ← (zmax + zmin)/2}
► To simulate a continuous space call contcenters, move each cluster center to the center-of-mass for its cluster
► Return our solution (F0 , g0 )
```

Minimized Expression

- The Lsearch algorithm minimizes the sum of square distances of each point from the closes cluster and the total cost to keep the facility centres open.

$$FC(N, C) = z|C| + \sum_{i=1}^{|C|} \sum_{x \in N_i} d(x, c_i)$$

Results

SSQ on stream of 20000 data points broken into chunks of 1000 data points averaged over 10 runs:

Number of centres:k=62. **SSQ=1.97**

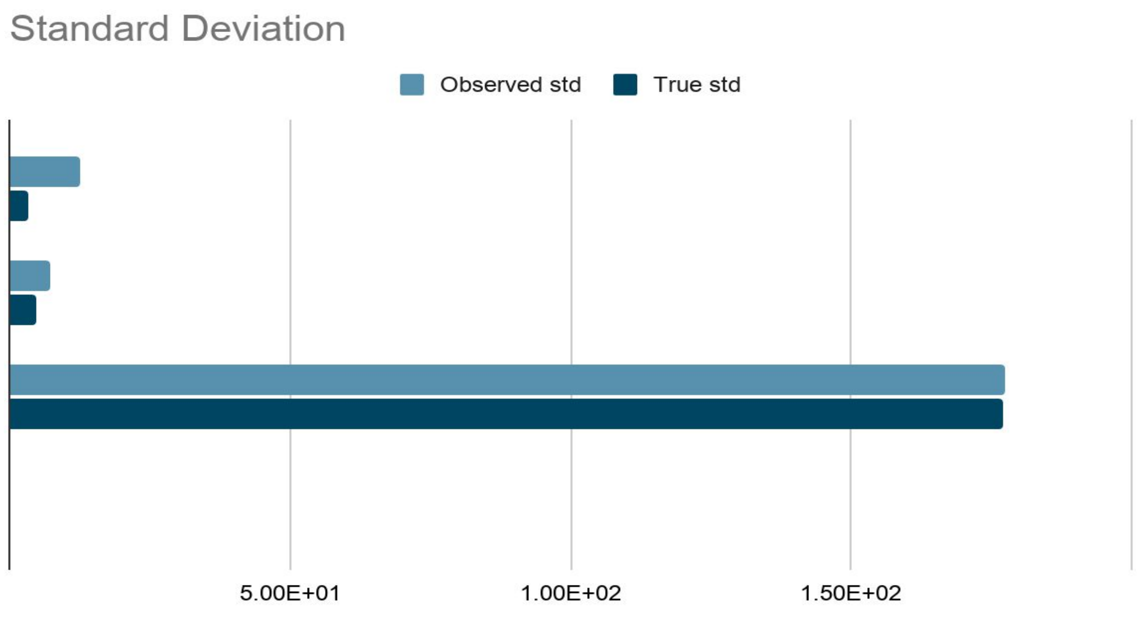
Kmeans clustering run over the static dataset of 20000 points for

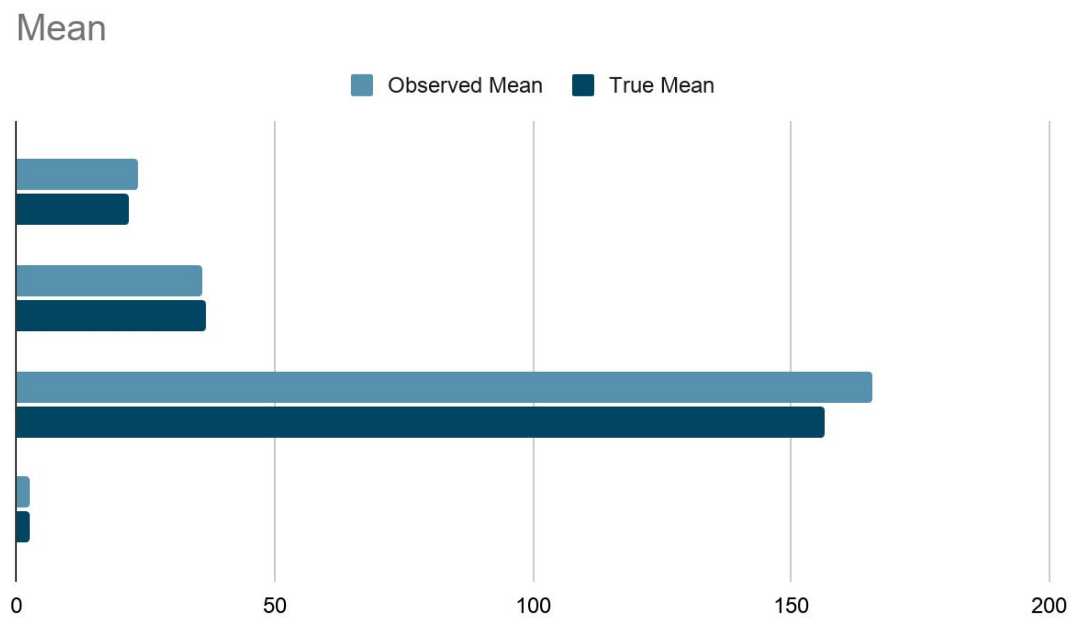
k=62. **SSQ: 1.48**

The statistics of the centres for each attribute in comparison to the statistics of the entire data

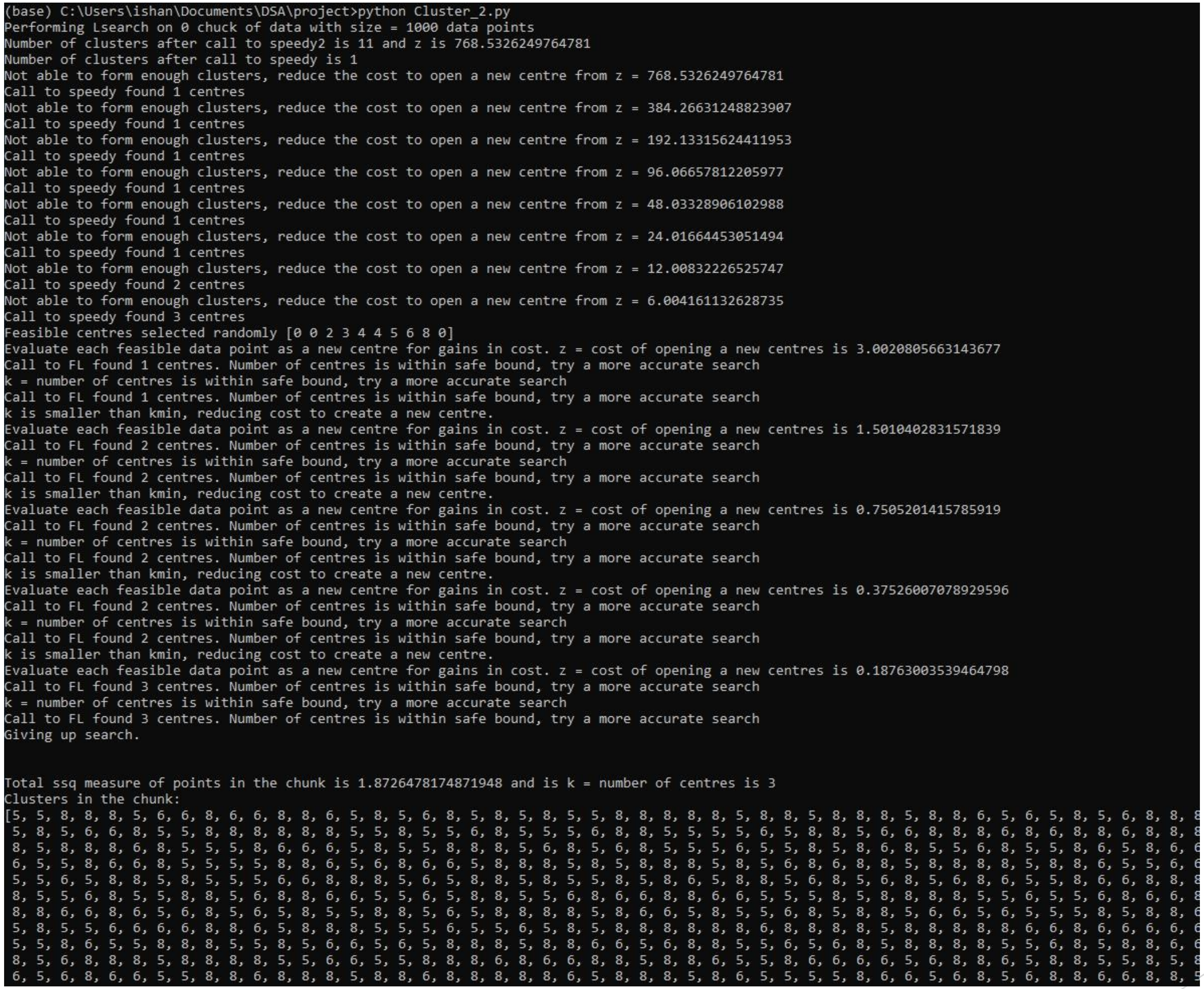
- Standard deviation of centres:[1.258e+01, 7.179e+00, 1.776e+02, 9.189e-02]
- True std_dev: [3.298e+00, 4.937e+00, 1.771e+02, 5.172e-02]

- Mean of centres: [23.443, 35.917, 165.642,   2.644]
- True mean: [21.925, 36.895, 156.644,   2.654]


Standard Deviation
Observed std
True std
5.00E+01
1.00E+02
1.50E+02

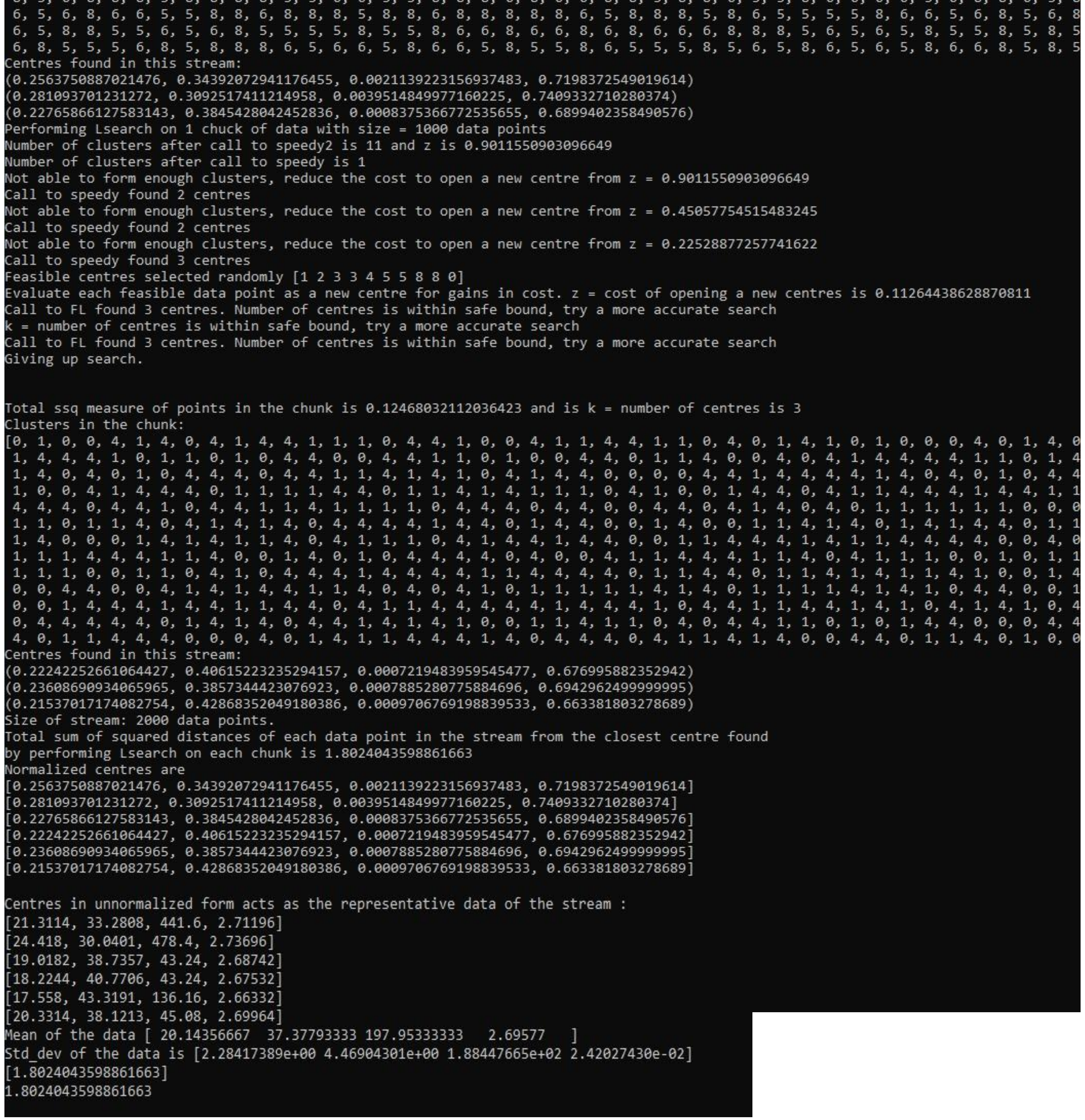

Mean
Observed Mean
True Mean
0
50
100
150
200

Performing stream lsearch over a stream of 2000 points broken into 2 chunks



```
(base) C:\Users\ishan\Documents\DSA\project>python Cluster_2.py
Performing Lsearch on 0 chuck of data with size = 1000 data points
Number of clusters after call to speedy2 is 11 and z is 768.5326249764781
Number of clusters after call to speedy is 1
Not able to form enough clusters, reduce the cost to open a new centre from z = 768.5326249764781
Call to speedy found 1 centres
Not able to form enough clusters, reduce the cost to open a new centre from z = 384.26631248823907
Call to speedy found 1 centres
Not able to form enough clusters, reduce the cost to open a new centre from z = 192.13315624411953
Call to speedy found 1 centres
Not able to form enough clusters, reduce the cost to open a new centre from z = 96.06657812205977
Call to speedy found 1 centres
Not able to form enough clusters, reduce the cost to open a new centre from z = 48.03328906102988
Call to speedy found 1 centres
Not able to form enough clusters, reduce the cost to open a new centre from z = 24.01664453051494
Call to speedy found 1 centres
Not able to form enough clusters, reduce the cost to open a new centre from z = 12.00832226525747
Call to speedy found 2 centres
Not able to form enough clusters, reduce the cost to open a new centre from z = 6.004161132628735
Call to speedy found 3 centres
Feasible centres selected randomly [0 0 2 3 4 4 5 6 8 0]
Evaluate each feasible data point as a new centre for gains in cost. z = cost of opening a new centres is 3.0020805663143677
Call to FL found 1 centres. Number of centres is within safe bound, try a more accurate search
k = number of centres is within safe bound, try a more accurate search
Call to FL found 1 centres. Number of centres is within safe bound, try a more accurate search
k is smaller than kmin, reducing cost to create a new centre.
Evaluate each feasible data point as a new centre for gains in cost. z = cost of opening a new centres is 1.5010402831571839
Call to FL found 2 centres. Number of centres is within safe bound, try a more accurate search
k = number of centres is within safe bound, try a more accurate search
Call to FL found 2 centres. Number of centres is within safe bound, try a more accurate search
k is smaller than kmin, reducing cost to create a new centre.
Evaluate each feasible data point as a new centre for gains in cost. z = cost of opening a new centres is 0.7505201415785919
Call to FL found 2 centres. Number of centres is within safe bound, try a more accurate search
k = number of centres is within safe bound, try a more accurate search
Call to FL found 2 centres. Number of centres is within safe bound, try a more accurate search
k is smaller than kmin, reducing cost to create a new centre.
Evaluate each feasible data point as a new centre for gains in cost. z = cost of opening a new centres is 0.37526007078929596
Call to FL found 2 centres. Number of centres is within safe bound, try a more accurate search
k = number of centres is within safe bound, try a more accurate search
Call to FL found 2 centres. Number of centres is within safe bound, try a more accurate search
k is smaller than kmin, reducing cost to create a new centre.
Evaluate each feasible data point as a new centre for gains in cost. z = cost of opening a new centres is 0.18763003539464798
Call to FL found 3 centres. Number of centres is within safe bound, try a more accurate search
k = number of centres is within safe bound, try a more accurate search
Call to FL found 3 centres. Number of centres is within safe bound, try a more accurate search
Giving up search.


Total ssq measure of points in the chunk is 1.8726478174871948 and is k = number of centres is 3
Clusters in the chunk:
[5, 5, 8, 8, 8, 5, 6, 6, 8, 6, 6, 8, 8, 6, 5, 8, 5, 6, 8, 5, 8, 5, 8, 5, 5, 8, 8, 8, 8, 8, 5, 8, 8, 5, 8, 8, 8, 5, 8, 8, 6, 5, 6, 5, 8, 5, 6, 8, 8, 8
 5, 8, 5, 6, 6, 8, 5, 5, 8, 8, 8, 8, 8, 8, 5, 5, 8, 5, 5, 6, 8, 5, 5, 5, 6, 8, 8, 5, 5, 5, 5, 6, 5, 8, 8, 5, 6, 6, 8, 8, 8, 6, 8, 6, 8, 8, 6, 8, 8, 8
 8, 5, 8, 8, 8, 6, 8, 5, 5, 5, 8, 6, 6, 6, 5, 8, 5, 5, 8, 8, 8, 5, 6, 8, 5, 6, 8, 5, 5, 5, 6, 5, 5, 8, 5, 8, 6, 8, 5, 5, 6, 8, 5, 5, 8, 6, 5, 8, 6, 6
 6, 5, 5, 8, 6, 6, 8, 5, 5, 5, 5, 8, 8, 6, 5, 6, 8, 6, 6, 5, 8, 8, 8, 5, 8, 5, 8, 8, 8, 5, 8, 5, 6, 8, 6, 8, 8, 5, 8, 8, 8, 8, 5, 8, 8, 6, 5, 5, 6, 6
 5, 5, 6, 5, 8, 8, 5, 8, 5, 5, 5, 6, 6, 8, 8, 8, 5, 6, 5, 8, 8, 5, 8, 5, 5, 8, 5, 8, 6, 5, 8, 8, 5, 6, 8, 5, 6, 8, 5, 6, 8, 6, 5, 5, 8, 6, 6, 8, 8, 8
 8, 5, 5, 6, 8, 5, 5, 8, 8, 5, 6, 8, 8, 6, 6, 5, 5, 6, 5, 8, 8, 5, 5, 6, 8, 6, 6, 8, 8, 6, 6, 5, 5, 5, 8, 5, 8, 8, 8, 8, 5, 5, 6, 5, 5, 6, 8, 6, 6, 8
 8, 8, 6, 6, 8, 6, 5, 6, 8, 5, 6, 5, 8, 5, 5, 8, 8, 5, 6, 5, 8, 8, 8, 8, 5, 8, 6, 6, 5, 8, 5, 5, 6, 8, 5, 8, 8, 5, 6, 6, 5, 6, 5, 5, 5, 8, 5, 8, 8, 6
 5, 8, 5, 5, 6, 6, 6, 6, 8, 8, 6, 5, 8, 8, 8, 5, 5, 5, 6, 5, 5, 6, 5, 8, 5, 8, 8, 8, 8, 8, 6, 8, 8, 8, 8, 5, 8, 8, 8, 8, 8, 6, 6, 8, 6, 6, 6, 6, 6
 5, 5, 8, 6, 5, 5, 8, 8, 8, 5, 5, 8, 5, 6, 6, 5, 6, 5, 8, 8, 8, 5, 8, 8, 6, 6, 5, 6, 8, 8, 5, 5, 6, 5, 6, 8, 5, 8, 8, 8, 8, 5, 5, 6, 8, 5, 8, 8, 6, 6
 8, 5, 6, 8, 8, 8, 5, 8, 8, 8, 8, 5, 5, 6, 6, 5, 5, 8, 8, 8, 6, 8, 6, 6, 8, 8, 5, 8, 8, 6, 5, 5, 8, 6, 6, 6, 6, 5, 6, 8, 8, 6, 5, 8, 8, 5, 5, 8, 5, 8
 6, 5, 6, 8, 6, 6, 5, 5, 8, 8, 6, 8, 8, 8, 5, 8, 8, 6, 8, 8, 8, 8, 8, 6, 5, 8, 8, 8, 5, 8, 6, 5, 5, 5, 5, 8, 6, 6, 5, 6, 8, 5, 6, 8, 8, 6, 6, 8, 8, 5
```



```
 6, 5, 6, 8, 6, 6, 5, 5, 8, 8, 6, 8, 8, 8, 5, 8, 8, 6, 8, 8, 8, 8, 8, 6, 5, 8, 8, 8, 5, 8, 6, 5, 5, 5, 5, 8, 6, 6, 5, 6, 8, 5, 6, 8
 6, 5, 8, 8, 5, 5, 6, 5, 6, 8, 5, 5, 5, 5, 8, 5, 5, 8, 6, 6, 8, 6, 6, 8, 6, 8, 6, 6, 6, 8, 8, 8, 5, 6, 5, 6, 5, 8, 5, 5, 8, 5, 8, 5
 6, 8, 5, 5, 5, 6, 8, 5, 8, 8, 8, 6, 5, 6, 6, 5, 8, 6, 6, 5, 8, 5, 5, 8, 6, 5, 5, 5, 8, 5, 6, 5, 8, 6, 5, 6, 5, 8, 6, 6, 8, 5, 8, 5
Centres found in this stream:
(0.2563750887021476, 0.34392072941176455, 0.0021139223156937483, 0.7198372549019614)
(0.281093701231272, 0.3092517411214958, 0.0039514849977160225, 0.7409332710280374)
(0.22765866127583143, 0.3845428042452836, 0.0008375366772535655, 0.6899402358490576)
Performing Lsearch on 1 chuck of data with size = 1000 data points
Number of clusters after call to speedy2 is 11 and z is 0.9011550903096649
Number of clusters after call to speedy is 1
Not able to form enough clusters, reduce the cost to open a new centre from z = 0.9011550903096649
Call to speedy found 2 centres
Not able to form enough clusters, reduce the cost to open a new centre from z = 0.45057754515483245
Call to speedy found 2 centres
Not able to form enough clusters, reduce the cost to open a new centre from z = 0.22528877257741622
Call to speedy found 3 centres
Feasible centres selected randomly [1 2 3 3 4 5 5 8 8 0]
Evaluate each feasible data point as a new centre for gains in cost. z = cost of opening a new centres is 0.11264438628870811
Call to FL found 3 centres. Number of centres is within safe bound, try a more accurate search
k = number of centres is within safe bound, try a more accurate search
Call to FL found 3 centres. Number of centres is within safe bound, try a more accurate search
Giving up search.


Total ssq measure of points in the chunk is 0.12468032112036423 and is k = number of centres is 3
Clusters in the chunk:
[0, 1, 0, 0, 4, 1, 4, 0, 4, 1, 4, 4, 1, 1, 1, 0, 4, 4, 1, 0, 0, 4, 1, 1, 4, 4, 1, 1, 0, 4, 0, 1, 4, 1, 0, 1, 0, 0, 0, 4, 0, 1, 4, 0
 1, 4, 4, 4, 1, 0, 1, 1, 0, 1, 0, 4, 4, 0, 0, 4, 4, 1, 1, 0, 1, 0, 0, 4, 4, 0, 1, 1, 4, 0, 0, 4, 0, 4, 1, 4, 4, 4, 4, 1, 1, 0, 1, 4
 1, 4, 0, 4, 0, 1, 0, 4, 4, 4, 0, 4, 4, 1, 1, 4, 1, 4, 1, 0, 4, 1, 4, 4, 0, 0, 0, 0, 4, 4, 1, 4, 4, 4, 4, 1, 4, 0, 4, 0, 1, 0, 4, 4
 1, 0, 0, 4, 1, 4, 4, 4, 0, 1, 1, 1, 1, 4, 4, 0, 1, 1, 4, 1, 4, 1, 1, 1, 0, 4, 1, 0, 0, 1, 4, 4, 0, 4, 1, 1, 4, 4, 4, 1, 4, 4, 1, 1
 4, 4, 4, 0, 4, 4, 1, 0, 4, 4, 1, 1, 4, 1, 1, 1, 1, 0, 4, 4, 4, 0, 4, 4, 0, 0, 4, 4, 0, 4, 1, 4, 0, 4, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0
 1, 1, 0, 1, 1, 4, 0, 4, 1, 4, 1, 4, 0, 4, 4, 4, 4, 1, 4, 4, 0, 1, 4, 4, 0, 0, 1, 4, 0, 0, 1, 1, 4, 1, 4, 0, 1, 4, 1, 4, 4, 0, 1, 1
 1, 4, 0, 0, 0, 1, 4, 1, 4, 1, 1, 4, 0, 4, 1, 1, 1, 0, 4, 1, 4, 4, 1, 4, 4, 0, 0, 1, 1, 4, 4, 4, 1, 4, 1, 1, 4, 4, 4, 4, 0, 0, 4, 0
 1, 1, 1, 4, 4, 4, 1, 1, 4, 0, 0, 1, 4, 0, 1, 0, 4, 4, 4, 4, 0, 4, 0, 0, 4, 1, 1, 4, 4, 4, 1, 1, 4, 0, 4, 1, 1, 1, 0, 0, 1, 0, 1, 1
 1, 1, 1, 0, 0, 1, 1, 0, 4, 1, 0, 4, 4, 4, 1, 4, 4, 4, 4, 1, 1, 4, 4, 4, 4, 0, 1, 1, 4, 4, 0, 1, 1, 4, 1, 4, 1, 1, 4, 1, 0, 0, 1, 4
 0, 0, 4, 4, 0, 0, 4, 1, 4, 1, 4, 4, 1, 1, 4, 0, 4, 0, 4, 1, 0, 1, 1, 1, 1, 1, 4, 1, 4, 0, 1, 1, 1, 1, 4, 1, 4, 1, 0, 4, 4, 0, 0, 1
 0, 0, 1, 4, 4, 4, 1, 4, 4, 1, 1, 4, 4, 0, 4, 1, 1, 4, 4, 4, 4, 4, 1, 4, 4, 4, 1, 0, 4, 4, 1, 1, 4, 4, 1, 4, 1, 0, 4, 1, 4, 1, 0, 4
 0, 4, 4, 4, 4, 4, 0, 1, 4, 1, 4, 0, 4, 4, 1, 4, 1, 4, 1, 0, 0, 1, 1, 4, 1, 1, 0, 4, 0, 4, 4, 1, 1, 0, 1, 0, 1, 4, 4, 0, 0, 0, 4, 4
 4, 0, 1, 1, 4, 4, 4, 0, 0, 0, 4, 0, 1, 4, 1, 1, 4, 4, 4, 1, 4, 0, 4, 4, 4, 0, 4, 1, 1, 4, 1, 4, 0, 0, 4, 4, 0, 1, 1, 4, 0, 1, 0, 0
Centres found in this stream:
(0.22242252661064427, 0.40615223235294157, 0.0007219483959545477, 0.676995882352942)
(0.23608690934065965, 0.3857344423076923, 0.0007885280775884696, 0.6942962499999995)
(0.21537017174082754, 0.42868352049180386, 0.0009706769198839533, 0.663381803278689)
Size of stream: 2000 data points.
Total sum of squared distances of each data point in the stream from the closest centre found
by performing Lsearch on each chunk is 1.8024043598861663
Normalized centres are
[0.2563750887021476, 0.34392072941176455, 0.0021139223156937483, 0.7198372549019614]
[0.281093701231272, 0.3092517411214958, 0.0039514849977160225, 0.7409332710280374]
[0.22765866127583143, 0.3845428042452836, 0.0008375366772535655, 0.6899402358490576]
[0.22242252661064427, 0.40615223235294157, 0.0007219483959545477, 0.676995882352942]
[0.23608690934065965, 0.3857344423076923, 0.0007885280775884696, 0.6942962499999995]
[0.21537017174082754, 0.42868352049180386, 0.0009706769198839533, 0.663381803278689]

Centres in unnormalized form acts as the representative data of the stream :
[21.3114, 33.2808, 441.6, 2.71196]
[24.418, 30.0401, 478.4, 2.73696]
[19.0182, 38.7357, 43.24, 2.68742]
[18.2244, 40.7706, 43.24, 2.67532]
[17.558, 43.3191, 136.16, 2.66332]
[20.3314, 38.1213, 45.08, 2.69964]
Mean of the data [ 20.14356667  37.37793333 197.95333333   2.69577    ]
Std_dev of the data is [2.28417389e+00 4.46904301e+00 1.88447665e+02 2.42027430e-02]
[1.8024043598861663]
1.8024043598861663
```

Running Time

- The running time of kmedians is $O(nm + nk \log k)$ where m is the number of facility centres opened during speedy2.
- Across chunks of 1000 data points, the number of centres opened by speedy2 was 12 on an average.
- N is the size of the chunk and k is the number of medians.

Space Complexity

- If the number of centres being maintained across all the chunks of data exceeds a pre-specified limit, max_centres, kmedian is called on the list of centres to reduce the amount of memory required to O(k_max) again. However, the amount of space required to store the centres in addition to the space required to run the algorithm is O(max_centres).
- The space required to run the algorithm is O(size of a chunk).

References

- https://www.cs.princeton.edu/courses/archive/spr05/cos598E/bib/stream_icde.pdf
- http://db.csail.mit.edu/labdata/labdata.html

Thank You