CS 328 : Homework 1

Sachin Yadav

Ques 1.

No, it is not a metric distance function. As it does not satisfies **Triangular Inequality**. Counter Example: Let x = $(1, 2)$, y = $(5, 6)$ and z = $(2, 6)$

$d(x, y)$ = $min(|1-5|, |2 - 6|)$ = 4
$d(x, z)$ = $min(|1-2|, |2 - 6|)$ = 1
$d(z, y)$ = $min(|2-5|, |6 - 6|)$ = 0

Here $d(x, z) + d(z, y)$ < $d(x, y)$ It violates Triangular Inequality, hence not metric.

Ques 2.

$$ Cost(C) = \frac{1}{C_{i}}\sum_{x, y \in C_{i}} \lVert x - y \rVert ^{2} $$

- Here we can manipulate the cost function and can reach a conclusion from it.
- We know that $$ \lVert a - b \rVert ^ {2} = \lVert a \rVert ^ {2} + \lVert b \rVert ^ {2} - 2 \times a ^{T}b $$

$\text{Cost}(C) =$

$$\frac{1}{C_i}\sum_{x,y\in C_i}\|x-y\|^2$$

As $\|a-b\|^2 = \|a\|^2 + \|b\|^2 - 2a^Tb$

$$= \frac{1}{C_i}\sum_{x,y\in C_i}\left(\|x\|^2 + \|y\|^2 - 2x^Ty\right)$$

$$= \frac{1}{C_i}\left(C_i\sum_{x\in C_i}\|x\|^2 + C_i\sum_{y\in C_i}\|y\|^2 - 2\sum_{x,y\in C_i}x^Ty\right)$$

$$= \sum_{x\in C_i}\|x\|^2 + \sum_{y\in C_i}\|y\|^2 - \frac{2}{C_i}\sum_{x,y\in C_i}x^Ty$$

$$= 2\sum_{x\in C_i}\|x\|^2 - \frac{2}{C_i}\sum_{x,y\in C_i}x^Ty$$

$$= 2\left[\sum_{x\in C_i}\|x\|^2 - \frac{1}{C_i}\sum_{x,y\in C_i}x^Ty\right]$$

$$= 2\left[\sum_{x\in C_i}\|x\|^2 - \frac{2}{C_i}\sum_{x,y\in C_i}x^Ty + \frac{1}{C_i}\sum_{x,y\in C_i}x^Ty\right]$$

$$= 2\left[\sum_{x\in C_i}\|x\|^2 - \frac{2}{C_i}\sum_{x,y\in C_i}x^Ty + \frac{1}{C_i}\sum_{y\in C_i}\|y\|^2\right]$$

$$= 2\left[\sum_{x\in C_i}\|x\|^2 - \frac{2}{C_i}\sum_{x,y\in C_i}x^Ty + C_i\sum_{y\in C_i}\left\|\frac{y}{C_i}\right\|^2\right]$$

$$= 2\left[\sum_{x\in C_i}\|x\|^2 - \frac{2}{C_i}\sum_{x,y\in C_i}x^Ty + C_i\sum_{y\in C_i}\left\|\frac{y}{C_i}\right\|^2\right]$$

$$= 2\left[\sum_{x\in C_i} \|x\|^2 - 2\sum_{x\in C_i} x^T\left(\sum_{y\in C_i} \frac{y}{C_i}\right) + C_i \sum_{y\in C_i} \left\|\frac{y}{C_i}\right\|^2\right]$$

$$= 2\left[\sum_{x\in C_i} \left\| x - \frac{1}{C_i}\sum_{y\in C_i} y \right\|\right]$$

$$= 2\,(\text{Sum of distance from mean}).$$

- From the above result, we get that the objective to reduce the square sum of distances from the cluster mean is the same as reducing the sum of average pairwise distance squares.
- Hence **Lloyd's alogrithm** would also work here.

Ques 3.

Ques 3.

Let's say, we perform continuos k-means clustering first.

Let C_i be an assigned median point, while clustering.

C_i need not be part of data set.

Cost of this cluster $= \sum_{x \in S} \left(\min \text{ of } dist(x, C_j) \text{ for } 1 \leq j \leq K \right)$

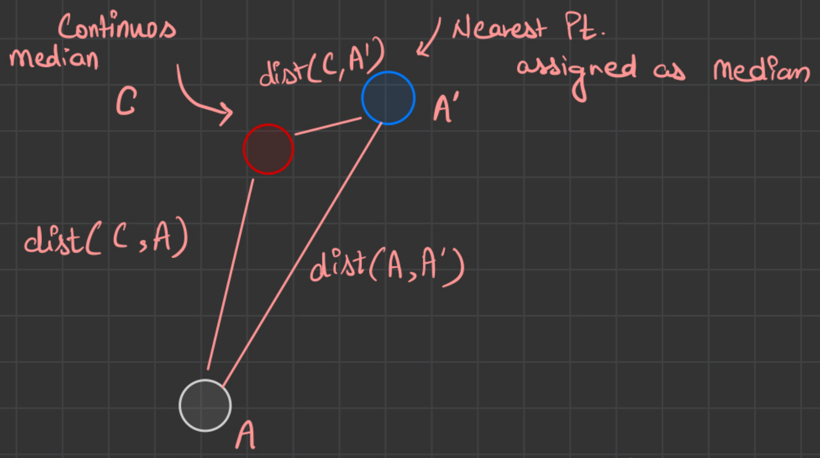
$S \to$ set of all points

Now if we want to assign these median points to corresponding points, which are part of dataset.

Then for each continuos median point, assign the closest point to it as the median point in discrete clustering.

Continuos k-median clustering $\longrightarrow$ Median point may not be part of dataset

Discrete k-median clustering $\longrightarrow$ Median point need to part of dataset



$dist(A', A) \leq dist(C, A) + dist(C, A')$

By Triangle Inequality

As A' is closer to C as compared to A.

$dist(C, A') \leq dist(C, A)$

Hence

$\underbrace{dist(A, A')}_{\text{distance of Pt.}} \leq 2 * \underbrace{dist(C, A)}_{\text{distance of Pt. from}}$

from new median pt. continuos median pt.

New Cost of Clustering:

$$\sum_{x \in S} \left(\min \{ dist(x, A_j') \text{ for } 1 \le j \le k \} \right)$$

$$\le 2 * \sum_{x \in S} \left(\min \{ dist(x, c_j) \text{ for } 1 \le j \le k \} \right)$$

Variation of Lloyd

Here we need to instead of taking mean in Lloyd, we will take median at every iteration.

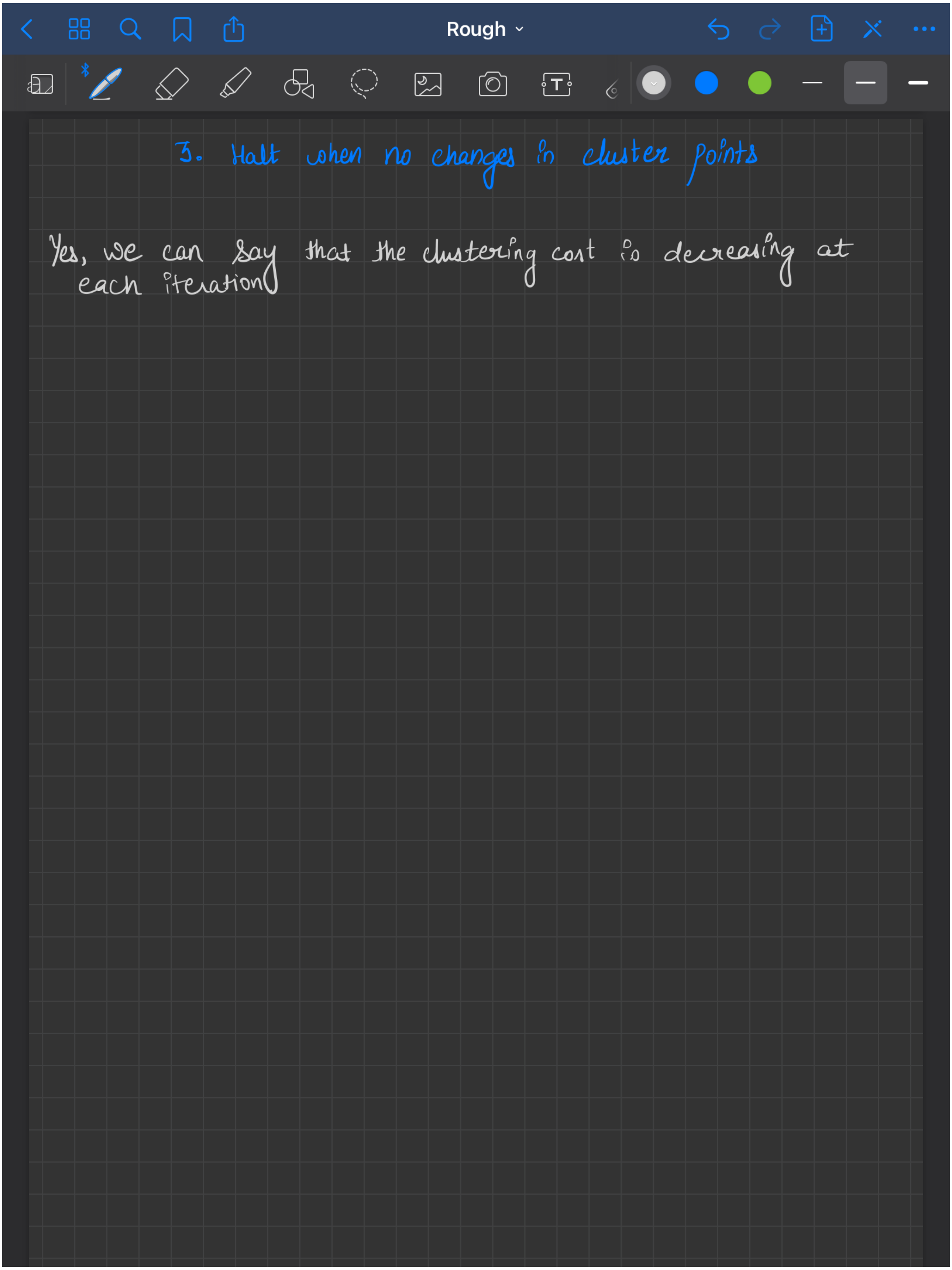
Median of a set, we define as

We take median of each dimension separately.

$$\underbrace{M_i}_{\text{i-th dimension vector element}} = \text{Median} \left[\underbrace{(X_j)_i}_{\text{median of ith dimension over entire cluster}} \text{ for } 1 \le j \le \underbrace{C_i}_{\text{No. of pts in cluster}} \right]$$

Algo:

1. Place k-point in domain at initial stage.
2. At each iteration:
   Compute median of each cell
   Change the cluster points to median



3. Halt when no changes in cluster points

Yes, we can say that the clustering cost is decreasing at each iteration

Ques 4.

Jupyter Notebook: **04.pynb**

Greedy K-Center Algo over entire Dataset

```
In [7]: def k_center_optimal(local_df, num_k):
            clustering_cost = math.inf
            num_rows = local_df.shape[0]
            allowed_bitmasks = generate_bitmasks_list(num_k, num_rows)
            for bitmask in tqdm(allowed_bitmasks, leave = False):
                centers_list = []
                for row_id in range(num_rows):
                    if (bitmask & (1 << row_id)):
                        centers_list.append(row_id)
                clustering_cost_candidate = max(min([find_distance(local_df, r_id, j) for j in centers_list]) for r_id in range(num_rows)
                clustering_cost = min(clustering_cost, clustering_cost_candidate)
            return clustering_cost
```

Greedy Algo over all DataSet

```
In [*]: for k in [2, 4, 10]:
            print("For Number of Clusters: ", k)
            greedy_objective_value = k_center_greedy(norm_df, k)
            print("Objective value of Greedy Algo :", greedy_objective_value, "\n")
```

```
For Number of Clusters:  2

Objective value of Greedy Algo : 2.1593968934215253

For Number of Clusters:  4

Objective value of Greedy Algo : 1.846269438999143

For Number of Clusters:  10

78% 7/9 [00:38<00:14, 7.27s/it]
```

Greedy Algo over all DataSet

```
In [20]: for k in [2, 4, 10]:
             print("For Number of Clusters: ", k)
             greedy_objective_value = k_center_greedy(norm_df, k)
             print("Objective value of Greedy Algo :", greedy_objective_value, "\n")
```

```
For Number of Clusters:  2

Objective value of Greedy Algo : 2.1593968934215253

For Number of Clusters:  4

Objective value of Greedy Algo : 1.846269438999143

For Number of Clusters:  10

Objective value of Greedy Algo : 1.4913641238206745
```

Comparison of Greedy vs Optimal over sample size of 20

jupyter 04 Last Checkpoint: 5 hours ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Greedy v/s Optimal on Sample size of 20

In [*]:

```
sample_df = norm_df.sample(n = 20)
for k in [2, 4, 10]:
    print("For Number of Clusters: ", k)
    greedy_objective_value = k_center_greedy(sample_df, k)
    optimal_objective_value = k_center_optimal(sample_df, k)
    approx_factor = greedy_objective_value / optimal_objective_value
    print("Objective value of Greedy Algo :", greedy_objective_value)
    print("Objective Value of Optimal Algo:", optimal_objective_value)
    print("Approximation Factor :", approx_factor, "\n\n")
```

```
For Number of Clusters:  2

Objective value of Greedy Algo : 1.464181001245336
Objective Value of Optimal Algo: 1.0535302518820355
Approximation Factor : 1.3897854367539142


For Number of Clusters:  4

Objective value of Greedy Algo : 1.1018027406242379
Objective Value of Optimal Algo: 0.9636419999491286
Approximation Factor : 1.1433735149385382


For Number of Clusters:  10

95% 176166/184756 [1:46:49<04:57, 28.86it/s]
```

In [19]:

```
sample_df = norm_df.sample(n = 20)
for k in [2, 4, 10]:
    print("For Number of Clusters: ", k)
    greedy_objective_value = k_center_greedy(sample_df, k)
    optimal_objective_value = k_center_optimal(sample_df, k)
    approx_factor = greedy_objective_value / optimal_objective_value
    print("Objective value of Greedy Algo :", greedy_objective_value)
    print("Objective Value of Optimal Algo:", optimal_objective_value)
    print("Approximation Factor :", approx_factor, "\n\n")
```

```
For Number of Clusters:  2

Objective value of Greedy Algo : 1.464181001245336
Objective Value of Optimal Algo: 1.0535302518820355
Approximation Factor : 1.3897854367539142


For Number of Clusters:  4

Objective value of Greedy Algo : 1.1018027406242379
Objective Value of Optimal Algo: 0.9636419999491286
Approximation Factor : 1.1433735149385382


For Number of Clusters:  10

Objective value of Greedy Algo : 0.511764351630979
Objective Value of Optimal Algo: 0.4616924620962943
Approximation Factor : 1.1084529067408542
```

Ques 5.

Youtube Video: https://youtu.be/a7FEbnSB6NM

Problems faced.

- Some of the rows (countries) were there which had most of the Nan Values for the coal consumption.
- SO I made a threshold of around 70%. All the rows which had more than or equal to 30 % of NaN values were removed from the data.

Getting rid of Missing Values:

- Because of the missing values, the animation was giving error sometimes.
- So as above, I chose a threshold and filtered the data accordingly.

- After that I checked every data table once of NaN values.

Combining Multiple tables for final table:

- Here for every country, I needed population, income and the coal consumption over the years.
- So I had to combine these multiple tables properly.