# Mohamed Shamir | 17110084

# CS 328 : Assignment 1

The entire solution and code is available at the **github repository** Link .

## Contents



### Problem 1

Is the following function d(x, y) = mini |xi – yi| a metric? Either prove it or give counter-examples.

**Solution**

Let's check whether the function follows the metric properties.

1. d(x,x) =0
   Here, min |xi - xi| =0. Therefore it satisfies that property.

2. d(x,y) = d(y,x)
   |yi - xi| = |xi - yi|
   min |yi - xi| = min |xi - yi|
   d(y,x) = d(x,y) It satisfies this property also.

3. d(x,y) + d(y,z) <= d(x,z)

   Let's take an example.
   x = [1,1,3,3,5]
   y = [2,9,13,1,2]
   z = [6,5,9,7,1]

   d(x,y) = min |xi - yi| = min (1,8,5,2,3) = 1
   d(y,z) = min |yi - zi | = min (3,4,4,6,1) = 1
   d(x,z) = min |xi - zi | = min (5,4,6,4,4) = 4

In this example,
d(x,y) + d (y,z) = 1+1 =2 >4
ie, d(x,y) + d(y,z) > d(x,z)

Hence, **it does not follows triangle inequality** which is proved by a counter example.

Therefore, **it is not a metric.**

---

## Problem 2

Suppose you define a clustering objecive in the following manner – give a partitioning C = {C1, . . . Ck},define

i.e. cost of a cluster is the sum of all pairwise squared distances. Give an algorithm for this.

**Solution**

```
Initialize K clusters c1,c2,c3...ck

iterate till convergence (means when the means does not change for all the
clusters):

   for point in P:
      for i in range(k):
         a= distance(point,mean{i})

      check which mean has the minimum distance.
      add the current point to that respective cluster. Say ith mean
curresponds to ith cluster.

   # Update the median
   for i in range(k):

      update(mean{i}) from the new points added in c{i} using the give cost
function.
```

---

## Problem 3

The k-median problem is defined in a similar way to the k-means problem, except that we do not take the squares of the distances when summing up. For the k-median problem, show that there is at most a factor of two ratio between the optimal value when we either require all cluster centers to be data points or allow arbitrary points to be centers. Based on this (or otherwise) propose a variant of the Lloyd's algorithm for Euclidean k-median. Can you say that the clustering cost is always decreasing?

**Solution:**

```
Initialize K clusters c1,c2,c3...ck

iterate till convergence (means when the medians does not change for all
the clusters):

   for point in P:
      for i in range(k):
         a= distance(point,median{i})

      check which median has the minimum distance.
      add the current point to that respective cluster. Say ith median
curresponds to ith cluster.

   # Update the median
   for i in range(k):

      update(median{i}) from the new points added in c{i}.
```

- The Clustering Cost would be always decrease because we are calculating the distance from the median always which represent the individuals. It can be in a way thought as centroids.

---

## Problem 4

Download the dataset in https://www.kaggle.com/arjunbhasin2013/ccdata./ As a good practice, normalize each feature such that the values are all in the range [0, 1]. Treat the CUST ID column as the identity of the point, not a feature. Use the L2 metric as distance. Implement the greedy k-center algorithm for this data and report the k-center objective value for k = 2, 4, 10. For small values of k, say k =2, 4 , find the optimal (when the centers are restricted to be input points) and report the approximation factor obtained by the greedy algorithm.

**Solution:**

**Greedy Method Algorithm**

Pseudo Code to find K centers:

Choose randomly a center from the given input point. Say the first point. and the input list be X.

```
center_list = []


distance_array =[INF]*len(X)

for i in range(K):
    current_center = select the point in X which has the maximum distance
```

```
from all the centers (basically the one with max value in the
distance_array)
    cluster_radius = {}
    center_list.add(current_center)

    for point in X:

        d = distance(point,c)
        distance_array[point] = min(distance_array[point],d) // Update the
distance_array.

        cluster_radius = max(distance_array[point],cluser_radius)

    cluster_radius[current_center] = cluster_radius
```

**To find the most optimal one**, the first k-center needs to be optimized. Earlier we had selected it randomly. Now, we would find the objective function for all the input points as first and the final objective value would be the minimum among all the k-centers chosen.

**Link to the Solution Notebook**

**Results**

- The Approximation Factor is coming about to be **1.3966834349636552** for **k=2** and **1.4667634183920994** for **k=4**.
- Theoretically the Approximation Factor should be less than or equal to 2.
- The Truthness of the Approximation Factor is validated using this experiment.

---

## Problem 5

For the following question you need to submit a link to a recorded video, YouTube link is preferable (can be unlisted). We intend to link to these videos from our public course webpage.

Go through the video at https://www.youtube.com/watch?v=hVimVzgtD6w. There are number of li- braries to create such visualization: one example is GapMinder animation, another is Plotly. Choose any dataset from any of the following websites:

- https://www.gapminder.org/data/
- http://www.healthdata.org/data-visualization/gbd-compare or http://ghdx.healthdata.org/ gbd-2017 (in Select Articles there are folder with data).
- https://niti.gov.in/state-statistics.

Take any two parameters, and either a number of Indian states, or a number of countries including India. Then create such a visualization. We rely on you to choose two parameters that make a somewhat interesting story as Hans Rosling does. If you want to use datasets about pandemic that is also fine — either come up with suggestions, or reach out to us. Note that you have to be sometimes careful about

missing data, data formatting etc these are all part of the problem. Document what problems you faced and what you did to handle these.

**Solution:**

- Used the CO2 emissions and the GDP per capita as the two parameters.
- Population of each country is used as cue for the scatter plot.
- Link to the Video
- Link to the Jupyter Notebook

**Challenges Faced:**

- Most of the countries did not have values of all the countries.
    - Selected the data from 1910 and neglected who had NaN values in 1910 (It would be better to analyze few than more. But make sured all the important countries are present within the dataset)
- Plotted the number of countries who had Null Values in their dataset.
- Filled it with Null value with the mean of that particular year as the variation was less at that time. And number of Null values where at most 2 per year.
- Plots were made using pyplots.

The datasets were taken from https://www.gapminder.org/data/