CS328 : Introduction to Data Science
Homework-1

Nishikant Parmar, 18110108

Feb 02, 2021

1 Question 1

No, the function $d(x, y) = min_i|x_i - y_i|$ is not a metric for dimensions higher than 1.

1.1 For dimension $= 1$

- $d(x, x) = min_i|x_i - x_i| = |x - x| = 0$
- $d(x, y) = min_i|x_i - y_i| = |x - y| > 0$ say $x \neq y$
- $d(x, y) + d(y, z) = |x - y| + |y - z| \geq |x - z|$, by triangle inequality.

Hence, all three properties are satisfied, and given function is metric for dimension 1.

1.2 For dimensions ≥ 2

Proof by contradiction -
Suppose dimension is 2. Let,

- $x = (1, 3)$
- $y = (1, 5)$
- $z = (5, 5)$

Now,

- $d(x, y) = min(|1 - 1|, |3 - 5|) = min(0, 2) = 0$,
- $d(y, z) = min(|1 - 5|, |5 - 5|) = min(4, 0) = 0$,

- $d(x,z) = min(|1-5|, |3-5|) = min(4,2) = 2$

Hence, $d(x,y) + d(y,z) = 0 + 0 = 0 < 2 = d(x,z)$

This is a contradiction of metric property $d(x,y) + d(y,z) \geq d(x,z)$

The above example can be extended for higher dimensions by using $x = (1, 3, 100, 100, 100, 100, ...(\text{all } 100s))$, $y = (1, 5, -100, -100, -100, ...(\text{all } -100s))$, $z = (5, 5, -100, -100, -100, ...(\text{all } -100s))$

2 Question 2

2.1 Key idea

First we will prove that the given objective cost function is related to the k-means objective cost function. After this, we can apply k-means or k-means++ algorithm.

2.2 Proof

Suppose we have k-clusters denoted by set $C = \{C_1, C_2, ...C_k\}$, and each data point is a vector of $d \times 1$ dimension, then the given objective function is -

$$cost(C) = \sum_i \frac{1}{|C_i|} \sum_{x,y \in C_i} \|x - y\|_2^2 = \sum_i \frac{1}{|C_i|} \sum_{y \in C_i} \sum_{x \in C_i} \|x - y\|_2^2$$

Now, for a fixed C_i, let us fix a $y \in C_i$

Suppose

$$m_i = \frac{1}{|C_i|} \sum_{x \in C_i} x$$

i.e. m_i (a vector of $d \times 1$ dimension) is the mean of all points in the cluster C_i.

$\sum_{x \in C_i} \|x - y\|_2^2 = \sum_{x \in C_i} \|x - m_i + m_i - y\|_2^2$

$= \sum_{x \in C_i} \|x - m_i\|_2^2 + \sum_{x \in C_i} \|y - m_i\|_2^2 + \sum_{x \in C_i} 2(m_i - y)^T (x - m_i)$

$= \sum_{x \in C_i} \|x - m_i\|_2^2 + |C_i| \|y - m_i\|_2^2 + 2(m_i - y)^T \sum_{x \in C_i} (x - m_i)$

$= \sum_{x \in C_i} \|x - m_i\|_2^2 + |C_i| \|y - m_i\|_2^2 + 2(m_i - y)^T (\sum_{x \in C_i} x - \sum_{x \in C_i} m_i)$

$= \sum_{x \in C_i} \|x - m_i\|_2^2 + |C_i| \|y - m_i\|_2^2 + 2(m_i - y)^T (\sum_{x \in C_i} x - |C_i| m_i)$

(But, by definition of m_i, $\sum_{x \in C_i} x - |C_i| m_i = (0, 0, ...d - times)^T$ i.e. it is a zero vector of $d \times 1$ dimension)

$= \sum_{x \in C_i} \|x - m_i\|_2^2 + |C_i| \|y - m_i\|_2^2 + 0$

$= \sum_{x \in C_i} \|x - m_i\|_2^2 + |C_i| \|y - m_i\|_2^2$

Hence, the cost now becomes,

$cost(C) = \sum_i \frac{1}{|C_i|} \sum_{y \in C_i} (\sum_{x \in C_i} \|x - m_i\|_2^2 + |C_i| \|y - m_i\|_2^2)$

$= \sum_i \frac{1}{|C_i|} (\sum_{y \in C_i} \sum_{x \in C_i} \|x - m_i\|_2^2 + \sum_{y \in C_i} |C_i| \|y - m_i\|_2^2)$

$= \sum_i \frac{1}{|C_i|} (|C_i| \sum_{x \in C_i} \|x - m_i\|_2^2 + \sum_{y \in C_i} |C_i| \|y - m_i\|_2^2)$

$= \sum_i \frac{1}{|C_i|} (|C_i| \sum_{x \in C_i} \|x - m_i\|_2^2 + |C_i| \sum_{y \in C_i} \|y - m_i\|_2^2)$

But, these two terms are actually the same.

$cost(C) = \sum_i \frac{1}{|C_i|} (2|C_i| \sum_{x \in C_i} \|x - m_i\|_2^2)$

$= 2 \sum_i \sum_{x \in C_i} \|x - m_i\|_2^2$

Where, $m_i = \frac{1}{|C_i|} \sum_{x \in C_i} x$

This is actually two times of the k-means objective function if we assume that each point has been assigned to a cluster where it's distance from the cluster mean m_i is least among other cluster.

Hence, to find such clustering $C = \{C_1, C_2, ...C_k\}$ we can apply k-means++ algorithm (Llyod's algorithm in particular) on the given points because the given cost turns out into k-means cost with a constant multiplication factor (which does not matter).

3 Question 3

3.1 Proof for factor of two between optimal when centers are data points v/s when centers can be any point

Suppose, we have some data set say S and we want to find k centers for clustering it.

Now, let $a = \{a_1, a_2, a_3..., a_k\}$ be the optimal k-centers for the given data set S. Then, the cost is -

$$cost(a) = \sum_x min_i d(x, a_i)$$

where $a_i \in R^d$, where d = dimension of data set (here we do not have any restriction on the centers, i.e. a_i can be any point). Here, $d(x, a_i)$ is the L2 (without square) Euclidean distance between vectors x and a_i. This is the optimal cost for clustering for S using L2 Euclidean distance.

Now, let $b = \{b_1, b_2, b_3..., b_k\}$ be a set such that $b_i \in S \ \forall i$ and b_i is the closest data point to $a_i \ \forall i$

then, cost with these centers is -

$$cost(b) = \sum_x min_i d(x, b_i)$$

Trivially $cost(a) \leq cost(b)$ since $cost(a)$ is the optimal and in worst case it can always use $a_i = b_i \ \forall i$

Now, for any point $x \in S$ and for a fixed i, we have

$d(x, b_i) \leq d(x, a_i) + d(a_i, b_i)$ by triangle inequality

But,

$$d(a_i, b_i) \leq d(x, a_i)$$

since, b_i is the closest point to a_i that lies in S

$$d(x, b_i) \leq d(x, a_i) + d(x, a_i)$$

$$d(x, b_i) \leq 2 \times d(x, a_i)$$

$$min_i d(x, b_i) \leq min_i(2 \times d(x, a_i))$$

$$\sum_x min_i d(x, b_i) \leq \sum_x min_i(2 \times d(x, a_i))$$

$$\sum_x min_i d(x, b_i) \leq 2 \times \sum_x min_i(d(x, a_i))$$

$$cost(b) \leq 2 \times cost(a)$$

Hence, this cost (i.e. $cost(b)$, when centers are restricted to be data points only) is bounded by a factor of 2 with the optimal cost (i.e. $cost(a)$, when center can be any point)

But $cost(b)$ may not be the optimal cost when centers are are restricted to be among data points, let $cost(o)$ be that optimal cost.

Then,

$$cost(o) \leq cost(b)$$

Hence, from the previous equation we get,

$$cost(o) \leq cost(b) \leq 2 \times cost(a)$$

$$cost(o) \leq 2 \times cost(a)$$

Thus, the optimal cost when centers have to be among data points is bounded by a factor of 2 with the optimal cost when centers could be any point.

Hence, proved.

3.2 Variant of Lloyd's algorithm for Euclidean k-median

- Choose k centers (points) from the given set of data points (arbitrarily or by using the initialization of kmeans++).
- For each point in our data set assign it a center (from the set of k centers that we have currently) such that the center is closest to the data point based on Euclidean distance.
- Now, let us call a cluster as a set of those data points which have the same center. There will be k such clusters.
- Find medians for each of these k clusters. This can be done by finding median in each of the dimensions.
- Replace k center with these new medians.
- Go to step 2 and repeat until a criteria is fulfilled.
- At the end, for each of the k centers obtained we choose a point in the data set that is closest to that center (based on Euclidean distance). Let, these new centers (in the data points) be the final clustering points.

The criteria for stopping could be that we stop when k-median cost is not decreasing or is decreasing by a very small value or the k-medians are not changing very much.

At each step, we use dimension-wise median to find the new median which may not lie in the data points. But, in the end we choose closest points (in the data points) to the k centers obtained so that our final center points are part of the data points.

3.3 Is cost always decreasing

Yes, at each step the cost will decrease since we are updating centers using median (dimension-wise). However at last when we choose centers among data points, the cost may increase.

4 Question 4

The k-center cost obtained by using greedy k-center algorithm on all the data points ($n = 8950$) -

k	Cost
2	2.0106026585562478
4	1.789436782259935
10	1.4984577152624328

To find optimal cost, the algorithm I have used is brute force i.e. I try all possible centers, calculate the cost and then take minimum cost among all. The time complexity of this algorithm is $O(n^{k+1}kd)$, where d is the dimension of data set. Hence, with $n = 8950$ and $k = 2$ or 4, this algorithm may take up hours, days or even centuaries to finish.

Hence, to apply this algorithm, I choose random 20 (By putting $n = 20$ and $k = 4$ the algorithm gives results in few seconds) points from the 8950 points.

The k-center cost obtained by optimal algorithm with these 20 data points -

k	Cost
2	1.258411509001605
4	1.00772537014411

The k-center cost obtained by using greedy k-center algorithm with these 20 data points -

k	Cost
2	1.679758268511973
4	1.2612759905672448

Hence, approximation factor calculated as cost by greedy k-center on 20 data points / optimal cost on 20 data points -

k	Cost
2	1.3348243054806888
4	1.251606864265882

As, expected this value lies in range [1, 2] since greedy k-center algorithm is 2-approximation of the k-center problem.

Now, approximation factor calculated as cost by greedy k-center on 8950 data points / optimal cost on 20 data points -

k	Cost
2	1.597730666140692
4	1.775718698045715

In this case, since for calculation of optimal cost (denominator) we are considering only 20 random data points instead of all points whereas in greedy we have considered all the points (numerator), hence the approximation factor may not be that accurate and is not a good indicator (it is actually wrong by definition as we are applying the two algorithms on different data sets). This factor may come out to be greater than 2 sometimes.

Note - All the above values may vary on each execution of algorithm since it involves some randomness, however the observations remain the same.

4.1 Challenges Faced

- For some data points, some values are NULL / missing.
- To overcome this issue, I first took dimension-wise mean of the not NULL entries and then assigned this value to all the data points that had NULL in that dimension.

4.2 Code

Colab Notebook

Github - Dataset

5 Question 5

For this problem, I have created an animation using Plotly Express animation library about the percentange of internet users in a conutry (India included) v/s it's income per person (GDP/capita) yearwise from 1988 to 2017. To see the animation generated click here (Kindly run the animation till 2017, then click on autoscale on top right of the plot and then run animation again for better visualization), this file is generated by the code and is also present here , the video has been uploaded here

5.1 Problems faced and how did I handle them

5.1.1 Data set collection

For making animations I required these attributes for each country for each year - continent, population in that year, income per person (GDP/capita) in that year, percentage of internet users in that year with respect to the population in that year. All these data was not available in a plain csv file. To handle this I downloaded various files from https://www.gapminder.org/data/ that could make up the complete data -

- que5_country_name_with_continent.csv which contains for each country it's continent
- que5_population.csv which contains for each country it's population (in that year) yearwise.
- que5_income_per_person.csv which contains for each country it's income per person (in that year) yearwise.
- que5_internet_users.csv which contains for each country it's percentage of internet users (out of the total population of that country in that year) yearwise.

5.1.2 Making data consistent

It can be noticed that the number of unique countries in que5_income_per_person.csv is 193, in que5_internet_users.csv is 194 and in

que5_population.csv is 195 whereas in que5_country_name_with_continent.csv is 142, i.e. some countries are missing in continent data, some countries are missing income and internet users data for all the years. And not all the files contain data about all the countries.

To handle this first I chose an year range from 1988 to 2017 (because for these years most of the data is available). Now, for each of the files I extracted data value for these years for each country and saved in nested dictionaries namely population_data_country_and_year_wise, internet_users_data_country_and_year_wise, and income_per_person_data_country_and_year_wise.

Now, to handle the problem of missing countries I chose those countries whose data was available in all the files, i.e. I took intersection of all the countries, and found out that for only 133 countries data was available in all the files.

Note - This missing data is not like for a country for some of the years, the data is missing (which is actually handled in the 3rd part) but for a country for **all the years** there is no data i.e. the entire country is missing from the file.

5.1.3 Missing/NULL values

Now, even after choosing those countries which had data present in all the files, the values for some of the years was NULL / missing and we cannot replace value 0 at those places since it would mean as if the value is actually 0.

To handle this, for each year I calculated mean value by taking into consideration all the countries where value is not NULL, and assigned this value to the countries that had NULL value i.e. a missing data for a country is assigned equal to the average data about the world in that year.

5.2 Code

Colab Notebook

Github - dataset

6 Collaborators

Discussed with,

- Aditya Tripathi (18110010)
- Kushagra Sharma (18110091)