# 2021 Introduction to Massive Data Analysis

# HW3 - K-means

## Deadline: 2021.11.12 (Fri.) 23:59

This problem will help you understand the nitty gritty details of implementing clustering algorithms on Hadoop. In addition, this problem will also help you understand the impact of using various distance metrics and initialization strategies in practice. Let us say we have a set $\mathcal{X}$ of $n$ data points in the $d$-dimensional space $\mathbb{R}^d$. Given the number of clusters $k$ and the set of $k$ centroids $\mathcal{C}$, we now proceed to define various distance metrics and the corresponding cost functions that they minimize.

**Euclidean distance** Given two points $A$ and $B$ in $d$ dimensional space such that $A = [a_1, a_2 \cdots a_d]$ and $B = [b_1, b_2 \cdots b_d]$, the Euclidean distance between $A$ and $B$ is defined as:

$$||a - b|| = \sqrt{\sum_{i=1}^{d} (a_i - b_i)^2} \tag{1}$$

The corresponding cost function $\phi$ that is minimized when we assign points to clusters using the Euclidean distance metric is given by:

$$\phi = \sum_{x \in \mathcal{X}} \min_{c \in \mathcal{C}} ||x - c||^2 \tag{2}$$

**Manhattan distance** Given two random points $A$ and $B$ in $d$ dimensional space such that $A = [a_1, a_2 \cdots a_d]$ and $B = [b_1, b_2 \cdots b_d]$, the Manhattan distance between $A$ and $B$ is defined as:

$$|a - b| = \sum_{i=1}^{d} |a_i - b_i| \tag{3}$$

The corresponding cost function $\psi$ that is minimized when we assign points to clusters using the Manhattan distance metric is given by:

$$\psi = \sum_{x \in \mathcal{X}} \min_{c \in \mathcal{C}} |x - c| \tag{4}$$

**Iterative $k$-Means Algorithm:** We learned the basic $k$-Means algorithm in class which is as follows: $k$ centroids are initialized, each point is assigned to the nearest centroid and the centroids are recomputed based on the assignments of points to clusters. In practice, the above steps are run for several iterations. We present the resulting iterative version of $k$-Means in Algorithm 1.

**Algorithm 1** Iterative $k$-Means Algorithm

```
1: procedure ITERATIVE k-MEANS
2:     Select k points as initial centroids of the k clusters.
3:     for iterations := 1 to MAX_ITER do
4:         for each point p in the dataset do
5:             Assign point p to the cluster with the closest centroid
6:         end for
7:         for each cluster c do
8:             Recompute the centroid of c as the mean of all the data points assigned to c
9:         end for
10:    end for
11: end procedure
```

**Iterative $k$-Means clustering on Hadoop:** Implement iterative $k$-means using MapReduce where a single step of MapReduce completes one iteration of the $k$-means algorithm. So, to run $k$-means for $i$ iterations, you will have to run a sequence of $i$ MapReduce jobs.

*Hint about **job chaining**:*

*We need to run a sequence of Hadoop jobs where the output of one job will be the input for the next one. There are multiple ways to do this and you are free to use any method you are comfortable with. One simple way to handle such a multistage job is to configure the output path of the first job to be the input path of the second and so on.*

*The following pseudo code demonstrates job chaining.*

```
var inputDir
var outputDir
var centroidDir

for i in no-of-iterations (
     Configure job here with all params
     Set job input directory = inputDir
     Set job output directory = outputDir + i
     Run job
     centroidDir =  outputDir + i
)
```

*You will also need to share the location of the centroid file with the mapper. There are many ways to do this and you can use any method you find suitable. One way is to use the Hadoop Configuration object. You can set it as a property in the Configuration object and retrieve the property value in the Mapper setup function.*

*For more details see :*

1. http://hadoop.apache.org/docs/r1.0.4/api/org/apache/hadoop/conf/Configuration.html#set(java.lang.String,java.lang.String)
2. http://hadoop.apache.org/docs/r1.0.4/api/org/apache/hadoop/conf/Configuration.html#get(java.lang.String)
3. http://hadoop.apache.org/docs/r1.0.4/api/org/apache/hadoop/mapreduce/Mapper.html#setup(org.apache.hadoop.mapreduce.Mapper.Context)

**Problem:**

The hw3-kmeans-testcase.zip has 3 files:

1. **data .txt**
   contains the dataset which has 4601 rows and 58 columns. Each row is a document represented as a 58 dimensional vector of features. Each component in the vector represents the importance of a word in the document.
2. **c1.txt**
   contains k initial cluster centroids. These centroids were chosen by selecting k = 10 random points from the input data.
3. **c2 . txt**
   contains initial cluster centroids which are as far apart as possible. (You can do this by choosing 1st centroid c1 randomly, and then finding the point c2 that is farthest from c1. then selecting c3 which is farthest from c1 and c2. and so on).

Please set number of iterations (**MAX_ITER**) to **20** and number of clusters **k** to **10** for all the experiments carried out in the question.

**(a) Exploring initialization strategies with Euclidean distance**

1. Using the Euclidean distance (refer to Equation 1) as the distance measure, compute the cost function $\phi(i)$ (refer to Equation 2) for every iteration $i$. This means that, for your first MapReduce job iteration, you'll be computing the cost function using the initial centroids located in one of the two text files. Run the $k$-means on `data.txt` using `c1.txt` and `c2.txt`. Generate a graph where you plot the cost function $\phi(i)$ as a function of the number of iterations $i$=1..20 for `c1.txt` and also for `c2.txt`.
   *(Hint: Note that you do not need to write a separate MapReduce job to compute $\phi(i)$. You can just incorporate the computation of $\phi(i)$ into the Mapper/Reducer.)*
2. What is the percentage change in cost after 20 iterations of the K-Means algorithm when the cluster centroids are initialized using `c1.txt` vs. `c2.txt` and the distance metric being used is Euclidean distance? Is random initialization of $k$-means using `c1.txt` better than initialization using `c2.txt` in terms of cost $\phi(i)$? Explain your reasoning.

**(b) Exploring initialization strategies with Manhattan distance**

1. Using the Manhattan distance metric (refer to Equation 3) as the distance measure, compute the cost function $\psi(i)$ (refer to Equation 4) for every iteration $i$. This means that, for your first MapReduce job iteration, you'll be computing the cost function using the initial centroids located in one of the two text files. Run the $k$-means on `data.txt` using `c1.txt` and `c2.txt`. Generate a graph where you plot the cost function $\psi(i)$ as a function of the number of iterations $i$=1..20 for `c1.txt` and also for `c2.txt`.
   *(Hint: This problem can be solved in a similar manner to that of part (a))*
2. What is the percentage change in cost after 20 iterations of the K-Means algorithm when the cluster centroids are initialized using `c1.txt` vs. `c2.txt` and the distance metric being used is Manhattan distance? Is random initialization of $k$-means using `c1.txt` better than initialization using `c2.txt` in terms of cost $\psi(i)$? Explain your reasoning.

**Assignment Requirement:**

Part1 Code

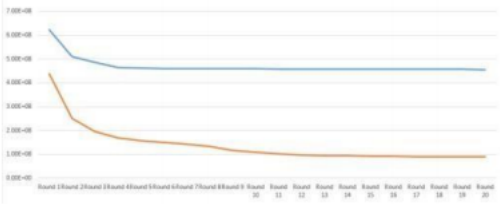
Upload the code for (a) and (b).
Should implement all mapper and reducer in **one file**.
Please make sure that your file has the same name as KMeans.
(**KMeans.java** or **KMeans.ipynb**)

Part2 Report (named as **Report.pdf**)

For question (a), you should show:

1. A plot of cost vs. iteration for 2 initialization strategies(c1 and c2) for (a)

| | C1 | C2 |
|---|---|---|
| Round 1 | 6.24E+08 | 4.39E+08 |
| Round 2 | 5.10E+08 | 2.50E+08 |
| Round 3 | 4.85E+08 | 1.94E+08 |
| Round 4 | 4.64E+08 | 1.70E+08 |
| Round 5 | 4.61E+08 | 1.56E+08 |



2. Percentage improvement values and your explanation for (a)

$$\frac{|Cost_{i=20} - Cost_{i=1}| * 100\%}{Cost_{i=1}}$$

3. The Euclidean and Manhattan Distances for all pairs of centroids, with 2 initialization strategies. (**4 tables** in total)

| Euclidean | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0.00 | value | value | value | value | value | value | value | value | value |
| 2 | | 0.00 | value | value | value | value | value | value | value | value |
| 3 | | | 0.00 | value | value | value | value | value | value | value |
| 4 | | | | 0.00 | value | value | value | value | value | value |
| 5 | | | | | 0.00 | value | value | value | value | value |
| 6 | | | | | | 0.00 | value | value | value | value |
| 7 | | | | | | | 0.00 | value | value | value |
| 8 | | | | | | | | 0.00 | value | value |
| 9 | | | | | | | | | 0.00 | value |
| 10 | | | | | | | | | | 0.00 |

For question (b), you should show:

1. A plot of cost vs. iteration for 2 initialization strategies(c1 and c2) for (b)
2. Percentage improvement values and your explanation for (b)
3. The Euclidean and Manhattan Distances for all pairs of centroids, with 2 initialization strategies. (**4 tables** in total)