

Cloud Computing for Data Analysis

Group Activity 05

k-means on MapReduce

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 X of n data points in the d -dimensional space \mathbf{R}^d . Given the number of clusters k and the set of k centroids 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 $A = [a_1, a_2, \dots, a_d]$ and $B = [b_1, b_2, \dots, b_d]$, the Euclidean distance between A and B is defined as:

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

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

$$\phi = \sum_{x \in X} \min_{c \in C} \|x - c\|^2 \quad (2)$$

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. Please use the dataset at <https://www.dropbox.com/sh/catm7lun0sijaf/AACMN-IsEADzqrjBNfAY9dhla?dl=0> for this problem. The zip has 4 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).

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

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 Mapper setup function

For more details see:

1. <http://hadoop.apache.org/docs/r1.0.4/api/org/apache/hadoop/conf/Configuration.html>
2. <http://hadoop.apache.org/docs/r1.0.4/api/org/apache/hadoop/mapreduce/Mapper.html>

Exploring initialization strategies with Euclidean distance

Task – 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 $i=1, \dots, 20$ 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.)

Task – 2:

What is the percentage change in cost after 10 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.

What to Submit?

1. Upload your source code, working .jar file
2. README file noting any assumptions you made in your program and steps to execute your code
3. A graph plot of cost vs. iteration for c1.txt and c2.txt for Task - 1
4. Percentage improvement values for Task - 2