HOMEWORK 1

Jie Tang

SEPTEMBER 14, 2020

Q1. Suppose you have an input of very large file of integers, design a MapReduce algorithm to output the the following:

a) output how many distinct integers from the file [5 marks]
b) output how many times each distinct integer appear [5 marks]
c) output the smallest integer [5 marks]
d) output the average of all integers [5 marks]

Answer:

a) i: map the input file, let it be key-value pairs (k, v), k is the integer, v is 1
ii: group by key, collect all the pairs with the same key
iii: reduce by key, collect all values belonging to the key and output
iv: count the total number of the key

b) i: map the input file, let it be key-value pairs (k, v), k is the integer, v is 1
ii: group by key, collect all the pairs with the same key
iii: reduce by key, collect all values belonging to the key and output
iv: the values belong to each key are the times each distinct integer appear

c) i: map the input file, let it be key-value pairs (k, v), k is the integer, v is 1
ii: sort by key ascending, collect the first pair
iii: reduce by key, combine the step ii results
iv: sort by key ascending again, collect the first pair's key

d) method1: i: map the input file, let it be key-value pairs (k, v), k is the integer, v is 1
ii: group by key, collect all the pairs with the same key
iii: reduce by key, collect all values belonging to the key and output
iv: use sum(key * value)/sum(value)

d) method2: i: map the input file, let it be key-value pairs (k, v), k is the integer, v is 1
ii: group by key, collect all the pairs with the same key
iii: sum(key)/sum(values) of each map stage
iv: map the results to be pairs (k, v)
v: reduce by key, collect all values belonging to the key and output
iv: repeat iii

Q2. Write a hadoop/spark program of K-Means clustering. Test it on MNIST and calculate the accuracy. Here is the link for MNIST [30 marks].

First, let have a quick review of K-means. The most common type of k-means is Lloyd's algorithm. The following is the algorithm's logic:

(a) Select K initial centroids
(b) Repeat
    i. Assign each data point to its nearest centroid
    ii. Update and recompute centroids and clusters
(c) Until centroid don't change

So we are essentially solving a minimization problem, where each data point gets assigned to its closest centroid each iteration, until the centroid don't change.

However, when the dataset is so large, it's very computationally prohibited to run K-means in a single node. So we want to use the MapReduce methods to help us calculate the distance part. To be more specific, we can use MapReduce to help us compute the distance of each data point to all centroids, and then we can find the nearest centroid for each data point.

The algorithm's logic with MapReduce thinking can be written as follows:
(a) Select K initial centroids
(b) Repeat
    i. For the Map process, calculate the distance of each data point to all centroids, assign each data point to its nearest centroid,input is (c, p), c is a set of centroids, and p is the data point; output (c_id, p), c_id is cluster label, and p is the data point

    ii. Since there are a lot of data points, we calculates the sum of the data instances for each cluster in each Map process, output (c_id, intermediate_centroid), c_id is cluster label, intermediate_centroid is the average of data instances in each Map process

    iii. In reduce part, input is (c_id, X), c_id is the cluster label, while the X is all the data points in that cluster, calculate the average of data instances, update centroids, output would be (c_id, c_new), c_id is the cluster label, c_new is the updated centroids for this cluster label
(c) Until centroid don't change or reach the maximum iteration

The accuracy I got is 23.3%, which I thought was quite bad at first. However, since I didn't do feature engineering and the dataset has 10 classes; also the original picture dataset might not be suitable for K-means, since K-means is about calculating the distance.

Bonus [20 marks]
Try your K-Means program on Google Cloud Platform, and report the time costs with different settings (number of computers, and size of dataset).
You may need to re-size the dataset into four sizes: 1/8, 1/4, 1/2, original size for the experiment.

Answer:

Size	Nodes	Time
Original	1	205s
1 / 2	1	80s
1 / 4	1	32s
1 / 8	1	17s
Original	2	85s
1 / 2	2	30s
1 / 4	2	13s
1 / 8	2	5s
Original	4	10s
1 / 2	4	4s
1 / 4	4	2s
1 / 8	4	2s

Note: The nodes I used are c5.Xlarge (4 vCores, 8 GB), region is at us-east-2.