HOMEWORK 1

GROUP 36

**Analysis of MRPrintStatistics:**

We start by describing in the standard way the MRPrintStatistics function, which is a 3 round function.

**Input:**

* N pairs (point, group) where each point is represented as a Vector of coordinates and each group is a Boolean (True = A, False = B);
* K centroids {}, each one represented as a Vector of coordinates.

**Output:**

* K pairs where is the index of the centroid of the input and are the number of points of A and B, respectively, in the cluster centered in . To output them in the required way, these should be sorted by index .

1. **ROUND 1:**
   1. **Map phase:** for each (point, group) pair separately compute the index of the closest center to it and put a 1 on the group of the point and a 0 in the other one, achieving:

(point, A) -> or (point, B) ->

* 1. **Reduce phase:** for each partition separately compute for every index the triplet , where and are respectively the number of points of group A and B that are in the partition and with the closest centroid .

1. **ROUND 2:**
   1. **Map phase:** empty.
   2. **Reduce phase:** for each index separately compute by summing the and of each partition .
2. **ROUND 3 (only for sorting):**
   1. **Map phase:** map every triplet to the same key:
   2. **Reduce phase:** sort the triplets by key.

Note that the reduce phase of round 1 and the reduce phase of round 2 are handled implicitly by Spark, as we only call the function *reduceByKey* which exploits the partitions as indicated in the official guide [1]. Note also that Round 3 is entirely handled by *sortByKey* function. If we didn’t require the output of the function to be in the same order of the centroids in input, the third round could be omitted.

Let’s compute a bound to the amount of local space required by each phase:

1. **ROUND 1:**
   1. **Map phase:** we compare each point separately with all the centroids, so the local memory is ;
   2. **Reduce phase:** we group together all the triplets of the same random partition that have the same key , so the local memory is where is the maximum number of pairs of the same partition that have the same key. If is sufficiently large, with high probability it becomes
2. **ROUND 2:**
   1. **Map phase:** empty.
   2. **Reduce phase:** we have at most K triplets for each partition, so a total of .
3. **ROUND 3:**
   1. **Map phase:** is .
   2. **Reduce phase:** is .

In total, the required local space is . If N is sufficiently large, with high probability it becomes . If for example then with high probability the local space requirements become , which is if also .