Supercomputer Architecture

Hands-on 9

Autumn - 2015

Professor: Jordi Torres

Constantino Gómez Crespo

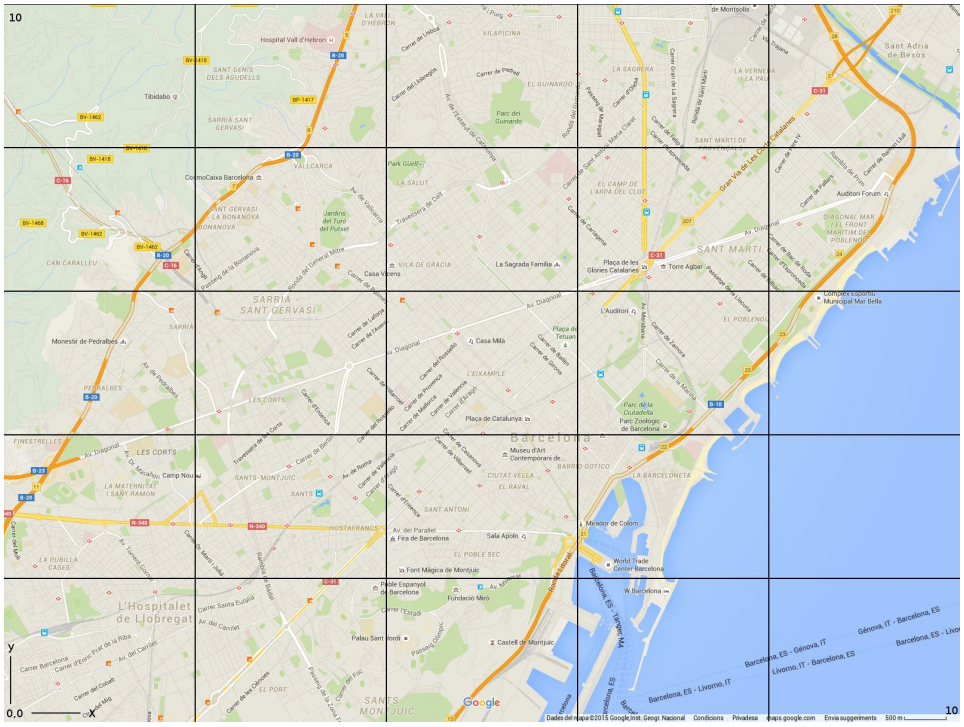
Cristóbal Ortega Carrasco

Albert Segura Salvador

1 Simple clustering with kmeans

Dataset:

The dataset used for this example are the locations of the FGC stations (displayed with their logo) on the area of Barcelona shown below in a space of 10x10. The information collected is shown in the right side of the image. With this information and the k-means clustering algorithm we will be able to group the nearest FGC stations.



mapa.txt

X	Y
1.8	0.1
1.3	0.8
2.1	0.3
2.6	0.9
3.2	1.8
1.8	5.8
2.1	5.8
4.1	5.8
5.6	4.1
1.3	7.1
3.0	7.3
3.3	6.7
3.7	6.3
0.2	8.3

Code:

The code used is the one provided in the example with the addition of a command to list the cluster centroids in order to display them later.

```
val data = sc.textFile("mapa.txt")
data.foreach(println)
import org.apache.spark.mllib.linalg.Vectors
val parsedData = data.map(s => Vectors.dense(s.split("
").map(_.toDouble))).cache()
parsedData.foreach(println)

import org.apache.spark.mllib.clustering.KMeans
```

```
val clusters = KMeans.train(parsedData, 3, 10)

val WSSSE = clusters.computeCost(parsedData)
print("Within Set Sum of Squared Error = " + WSSSE)

clusters.clusterCenters
```

Output:

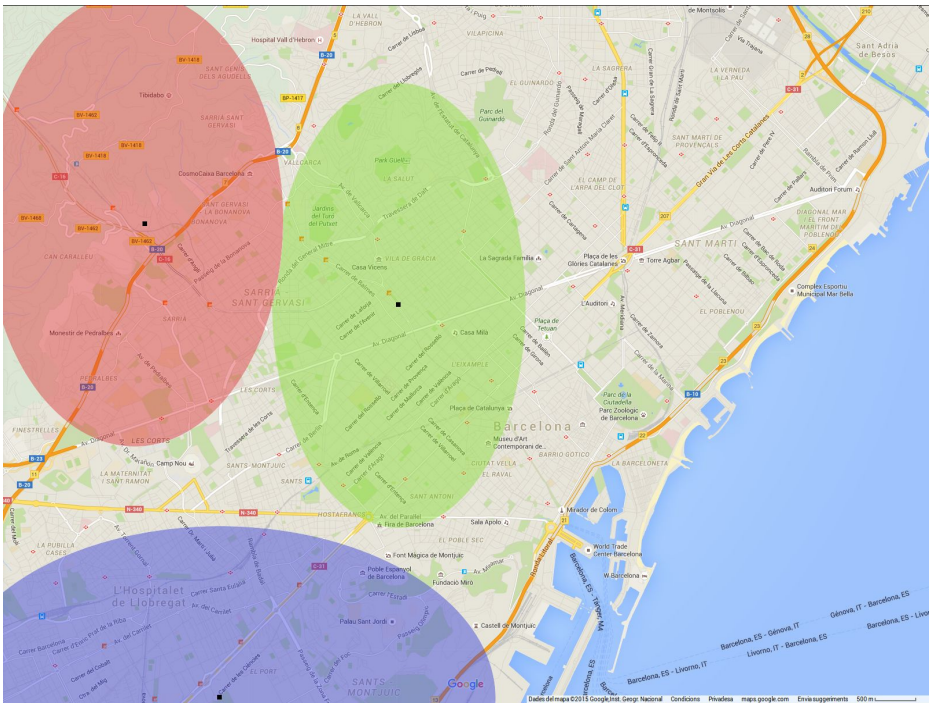
```
Within Set Sum of Squared Error = 19.68300000000002

Cluster Centroids:
Array([2.2,0.7800000000000001],
[1.6800000000000002,6.859999999999999],
[4.174999999999999,5.7250000000000005])
```

Clustering result:

The cluster centroids obtained are three, as specified in the command, and are the following:

X	Y
2.2	0.8
1.7	6.9
4.2	5.7

One obtained the centroids we can put these centers on the previous map and show the location of them and their nearest stations which are part of their cluster. The following image shows the result of the k-means clustering.



2 Testing other Apache-Spark functionalities

We want to find out which is the best place to rent a flat in barcelona, for that purpose we retrieved the data about the total female population per area in Barcelona from the City Hall Open Data webpage (http://opendata.bcn.cat/opendata/en/catalog/).

The dataset can be found here (http://opendata.bcn.cat/opendata/en/catalog/DEMOGRAFIA/ine-ine04/)

We are interested in the range from 20 to 30 years old. In apache spark we use the following code to quickly add up values from several columns.

```
val AllLabels =
List("DTE","BARRIS","TOTAL","ZEROANYS","UNANY","DOSANYS","TRESANYS","
QUATREANYS","CINCANYS" [ … ]
,"NOURANTATRESANYS","NOURANTAQUATREANYS","NOURANTACINCANYSIMES")

val lbla = List("VINTANYS","VINTIUNANYS","VINTIDOSANYS"
,"VINTITRESANYS" ,"VINTICUATREANYS" ,"VINTICINCANYS"
,"VINTISISANYS","VINTISETANYS","VINTIVUITANYS",
"VINTINOUANYS","TRENTAANYS")

val index_lbla = lbla.map(x => AllLabels.indexOf(x))

val dataRDD =
sc.textFile("opendata_2014_ine-ine04.csv").map(_.split(";"))

dataRDD.map(x => index_lbla.map(i => x(i).toInt).sum).collect
```

OUTPUT

This is the raw output

(3907, 1635, 1291, 2093, 2208, 3538, 3098, 3168, 4270, 2495, 2883, 72, 1838, 716, 1171, 1197, 1679, 2837, 2928, 1535, 617, 233, 1376, 1062, 1578, 2932, 1939, 902, 435, 859, 3927, 2235, 1645, 542, 2210, 459, 1881, 665, 363, 238, 296, 44, 1550, 1423, 1281, 980, 137, 711, 373, 1129, 802, 1469, 476, 180, 692, 66, 640, 140, 628, 3151, 1808, 775, 1317, 2371, 1663, 1099, 525, 1772, 611, 1441, 1293, 1364, 1573)

Therefore since we know the order of the neighborhoods we can conclude that we are interested in el Raval, Sagrada Familia, Nova esquerra eixample and la Vila de gràcia