

Report for lab assignment 5

Ting Xia ID 30

1. Question:

Spark and Smartphone/Watch Application

Implement a smart application with big data analytics related to your project showing the collaboration between Spark and Smart Apps. Note: For the big data analytics part, you can use either SparkR or Spark MLlib.

Description:

I use K-Means method in Spark MLlib to analyze heart rater. Here, I define five clusters: high danger, high, normal, low, low danger. After Spark analyzes the three clusters, the cluster centers will be sent to the Android phone using Android socket connection.

K-means code:

```
/**
 * Created by Ting on 2/24/16.
 */
object HeartRateKMeans {
  def main(args: Array[String]) {

    val sparkConf = new SparkConf().setAppName("SparkMLlib").setMaster("local[*]")

    val sc = new SparkContext(sparkConf)

    // Load and parse the data
    val data = sc.textFile("data/heartRate.txt")
    val parsedData = data.map(s => Vectors.dense(s.split(' ').map(_.toDouble))).cache()

    // Cluster the data into three classes using KMeans
    val numClusters = 5
    val numIterations = 50
    val clusters = KMeans.train(parsedData, numClusters, numIterations)

    // Evaluate clustering by computing Within Set Sum of Squared Errors
    val WSSSE = clusters.computeCost(parsedData)
    println("Within Set Sum of Squared Errors = " + WSSSE)

    // Save and load model
    clusters.save(sc, "myModelPath")
    val sameModel = KMeansModel.load(sc, "myModelPath")

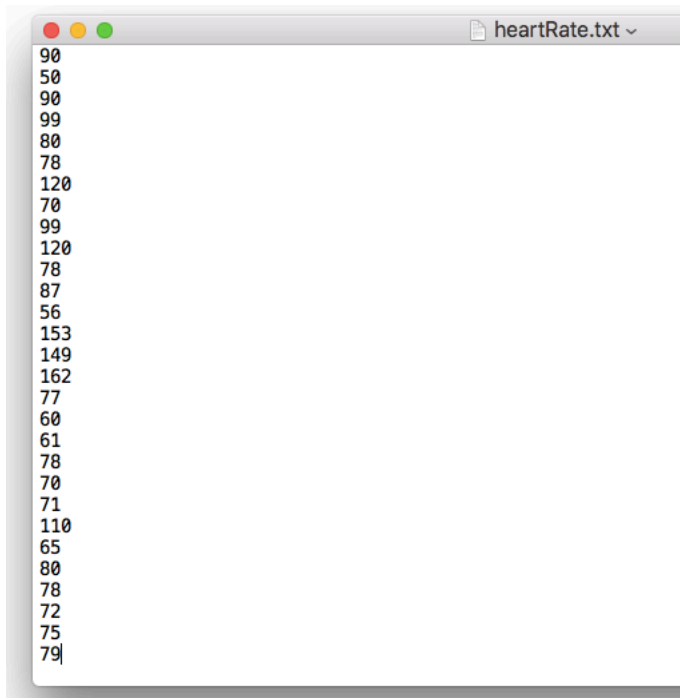
    // Shows the result
    println("Final Centers: ")
    sameModel.clusterCenters.foreach(println)
    // $example off$

    sc.stop()

    var s: String = "Final Centers\n"
    sameModel.clusterCenters.foreach { case (center) => {
      s += center + "\n"}
    }
  }
}
```

Screenshots:

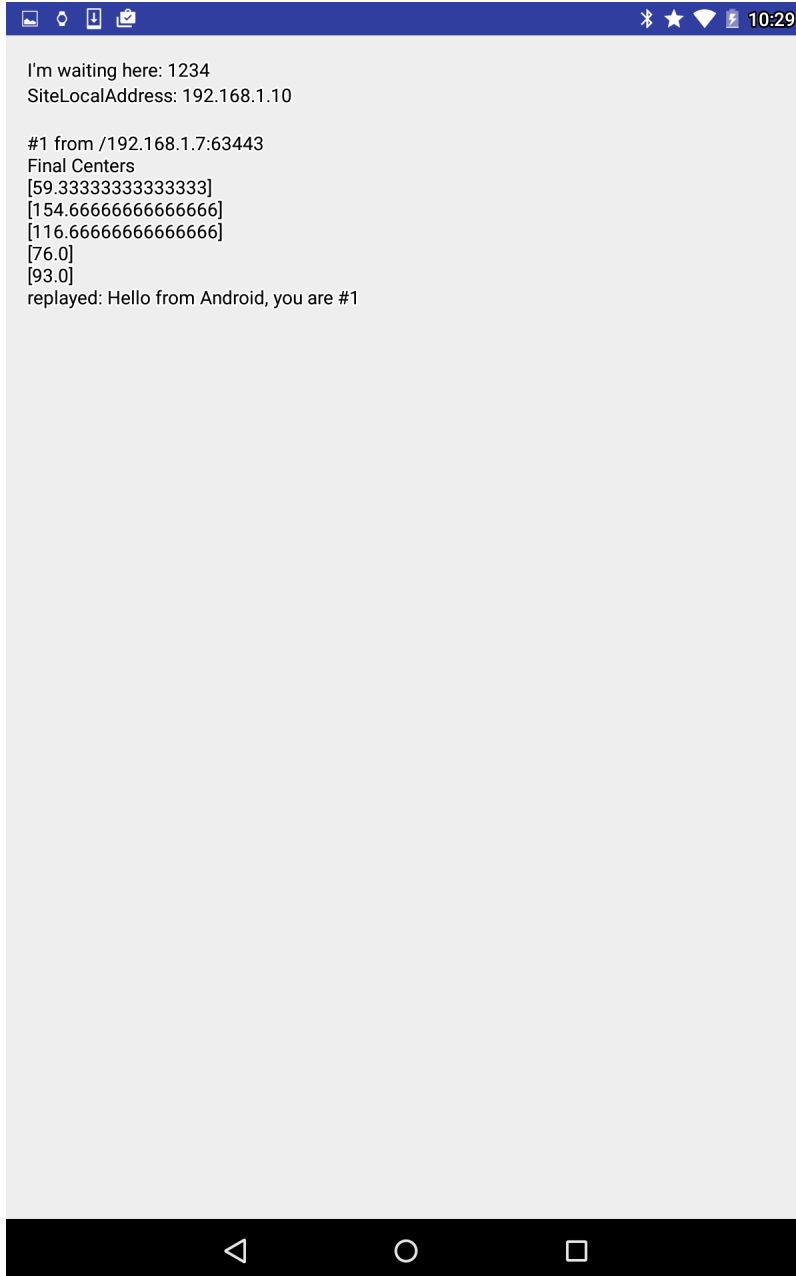
Data input (heart rate):



Run logs:

```
16/02/24 22:28:06 INFO TaskSchedulerImpl: Removed TaskSet 21.0, whose tasks have all comple
16/02/24 22:28:06 INFO DAGScheduler: ResultStage 21 (collect at KMeansModel.scala:147) fini
16/02/24 22:28:06 INFO DAGScheduler: Job 19 finished: collect at KMeansModel.scala:147, too
Final Centers:
[59.33333333333333]
[154.66666666666666]
[116.66666666666666]
[76.0]
[93.0]
16/02/24 22:28:06 INFO SparkUI: Stopped Spark web UI at http://192.168.1.7:4040
16/02/24 22:28:06 INFO MapOutputTrackerMasterEndpoint: MapOutputTrackerMasterEndpoint stopp
16/02/24 22:28:06 INFO MemoryStore: MemoryStore cleared
16/02/24 22:28:06 INFO BlockManager: BlockManager stopped
16/02/24 22:28:06 INFO BlockManagerMaster: BlockManagerMaster stopped
```

Android phone received:



2. Question:

SparkR – Extra Credit

Install Spark and Configure SparkR. Execute Word Count and K-Means Clustering algorithms using your own data (relevant to your project) with SparkR. Write a report including your algorithms and result screenshots. Modify the K-Means SparkR implementation for K-Medoids algorithm. Reference: <https://amplab-extras.github.io/SparkR-pkg/>

Description:

Word count: using existing script: wordcount.R

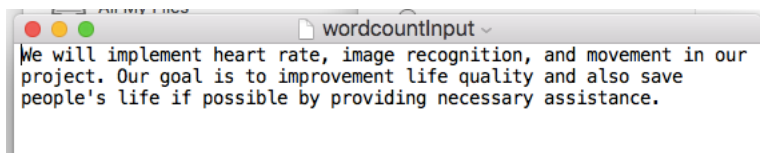
```
./bin/sparkR Users/Ting/Downloads/SparkR-SourceCodeData/wordcount.R local  
SparkR-SourceCodeData/SparkR-SourceCodeData/wordCount.txt
```

K-Means: using existing script: kmeans.R

```
./bin/sparkR SparkR-SourceCodeData/kmeans.R local SparkR-  
SourceCodeData/heartRate.txt 5 50
```

Screenshots:

Word count input:



Word count output:

```
16/02/22 17:16:01 INFO DAGScheduler: ResultStage 1 (collect at NativeMethodAccesso
hed in 0.133 s
16/02/22 17:16:01 INFO DAGScheduler: Job 0 finished: collect at NativeMethodAccess
1.709973 s
improvement : 1
assistance. : 1
providing : 1
rate, : 1
our : 1
We : 1
life : 2
is : 1
will : 1
Our : 1
heart : 1
movement : 1
implement : 1
necessary : 1
project. : 1
image : 1
also : 1
by : 1
goal : 1
if : 1
in : 1
save : 1
people's : 1
to : 1
quality : 1
possible : 1
recognition, : 1
and : 2
16/02/22 17:16:04 INFO SparkContext: Invoking stop() from shutdown hook
16/02/22 17:16:04 INFO SparkUI: Stopped Spark web UI at http://10.151.0.82:4040
```

K-means input:

```
heartRate.txt
90
50
90
99
80
78
120
70
99
120
78
87
56
153
149
162
77
60
61
78
70
71
110
65
80
78
72
75
79
```

K-means output:

```
16/02/24 22:36:01 INFO DAGScheduler: Result stage 101 (collect at null:
16/02/24 22:36:01 INFO DAGScheduler: Job 51 finished: collect at null:
Finished iteration (delta = 203090 )
[1] 50
There were 50 or more warnings (use warnings() to see the first 50)
Final centers:
59
61
87
77
79
153
70
66
54
80
120
110
56
75
57
60
65
99
72
162
90
78
71
149
59
61
```

99
72
162
90
78
71
149

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]	[,10]	[,11]	[,12]
[1,]	8000	7220	720	2420	2000	14580	4205	5445	10125	1805	2205	605
[2,]	13005	12005	2645	5445	4805	9245	8000	9680	15680	4500	500	0
[3,]	45	125	4805	2205	2645	47045	980	500	20	2880	20480	14580
[4,]	125	245	5445	2645	3125	49005	1280	720	0	3380	21780	15680
[5,]	1280	980	720	20	80	30420	125	405	2205	125	10125	6125
[6,]	53045	51005	28125	36125	34445	405	42320	46080	58320	33620	8820	13520
[7,]	245	125	2205	605	845	37845	80	0	720	980	14580	9680
[8,]	4805	4205	45	845	605	19845	2000	2880	6480	500	4500	2000
[9,]	1620	1280	500	0	20	28880	245	605	2645	45	9245	5445
[10,]	20	0	3380	1280	1620	42320	405	125	245	1805	17405	12005
[11,]	18605	17405	5445	9245	8405	5445	12500	14580	21780	8000	0	500
[12,]	40500	38720	19220	25920	24500	80	31205	34445	45125	23805	4205	7605
[13,]	2000	1620	320	20	0	27380	405	845	3125	5	8405	4805
[14,]	0	20	3920	1620	2000	44180	605	245	125	2205	18605	13005
[15,]	845	605	1125	125	245	32805	20	180	1620	320	11520	7220
[16,]	44180	42320	21780	28880	27380	0	34445	37845	49005	26645	5445	9245
[17,]	180	80	2420	720	980	38720	125	5	605	1125	15125	10125
[18,]	5	5	3645	1445	1805	43245	500	180	180	2000	18000	12500
[19,]	1805	1445	405	5	5	28125	320	720	2880	20	8820	5120
[20,]	20	80	4500	2000	2420	46080	845	405	45	2645	19845	14045
[21,]	720	500	1280	180	320	33620	5	125	1445	405	12005	7605
[22,]	3920	3380	0	500	320	21780	1445	2205	5445	245	5445	2645
[23,]	605	405	1445	245	405	34445	0	80	1280	500	12500	8000
[24,]	2205	1805	245	45	5	26645	500	980	3380	0	8000	4500

~/Downloads/spark-1.6.0-bin-hadoop2.6 --- -bash

[21,]	720	500	1280	180	320	33620	5	125	1445	405	12005	7605
[22,]	3920	3380	0	500	320	21780	1445	2205	5445	245	5445	2645
[23,]	605	405	1445	245	405	34445	0	80	1280	500	12500	8000
[24,]	2205	1805	245	45	5	26645	500	980	3380	0	8000	4500
[1,]	[,13]	[,14]	[,15]	[,16]	[,17]	[,18]	[,19]	[,20]	[,21]	[,22]	[,23]	[,24]
[2,]	14580	6125	14045	12500	10125	605	7220	13520	2000	5120	7605	7605
[3,]	0	1805	5	80	405	9245	1280	56180	5780	2420	1125	43245
[4,]	20	2205	45	180	605	10125	1620	58320	6480	2880	1445	45125
[5,]	1805	0	1620	1125	500	2880	45	37845	1125	45	80	27380
[6,]	56180	37845	55125	52020	47045	19845	40500	0	25920	35280	41405	845
[7,]	500	405	405	180	5	5445	180	46080	2880	720	125	34445
[8,]	5780	1125	5445	4500	3125	405	1620	25920	0	720	1805	17405
[9,]	2205	20	2000	1445	720	2420	125	36125	845	5	180	25920
[10,]	125	980	80	5	80	7220	605	51005	4205	1445	500	38720
[11,]	20480	10125	19845	18000	15125	2205	11520	8820	4500	8820	12005	4205
[12,]	43245	27380	42320	39605	35280	12500	29645	845	17405	25205	30420	0
[13,]	2645	80	2420	1805	980	2000	245	34445	605	5	320	24500
[14,]	45	1280	20	5	180	8000	845	53045	4805	1805	720	40500
[15,]	1280	45	1125	720	245	3645	0	40500	1620	180	5	29645
[16,]	47045	30420	46080	43245	38720	14580	32805	405	19845	28125	33620	80
[17,]	405	500	320	125	0	5780	245	47045	3125	845	180	35280
[18,]	80	1125	45	0	125	7605	720	52020	4500	1620	605	39605
[19,]	2420	45	2205	1620	845	2205	180	35280	720	0	245	25205
[20,]	5	1620	0	45	320	8820	1125	55125	5445	2205	980	42320
[21,]	1125	80	980	605	180	3920	5	41405	1805	245	0	30420
[22,]	4805	720	4500	3645	2420	720	1125	28125	45	405	1280	19220
[23,]	980	125	845	500	125	4205	20	42320	2000	320	5	31205
[24,]	2880	125	2645	2000	1125	1805	320	33620	500	20	405	23805

16/02/24 22:36:04 INFO SparkContext: Invoking stop() from shutdown hook

16/02/24 22:36:05 INFO SparkUI: Stopped Spark web UI at http://192.168.1.7:4040

16/02/24 22:36:05 INFO MapOutputTrackerMasterEndpoint: MapOutputTrackerMasterEndpoint stopped!

16/02/24 22:36:05 INFO MemoryStore: MemoryStore cleared