Advances in Social Science, Education and Humanities Research, volume 223
International Conference on Education, Economics and Social Science (ICEESS 2018)

Analysis of Consumer Behavior on Campus Based on Spark

Xiao-zheng WANG*, Tong-qi XUE, Jia-xi XU
Nanjng Xiaozhuang University
Nanjing, China

***Abstract*—By analyzing the correlation of consumer behavior on campus, the students' personality could be understood. Specifically, the ratio of the number of simultaneous consumption of any two students in a year was calculated based on consumption time and location. In view of the huge amount of computation, the computing framework uses Spark distributed framework, and the analysis results are displayed to the users through the Zeppelin framework. Through analysis of the ratio of the number of simultaneous consumption of any two students, it shows most students have regular partners and a few of the students needs more attention from schools.**

Keywords—Spark; Consumer behavior; Mass data

I. INTRODUCTION

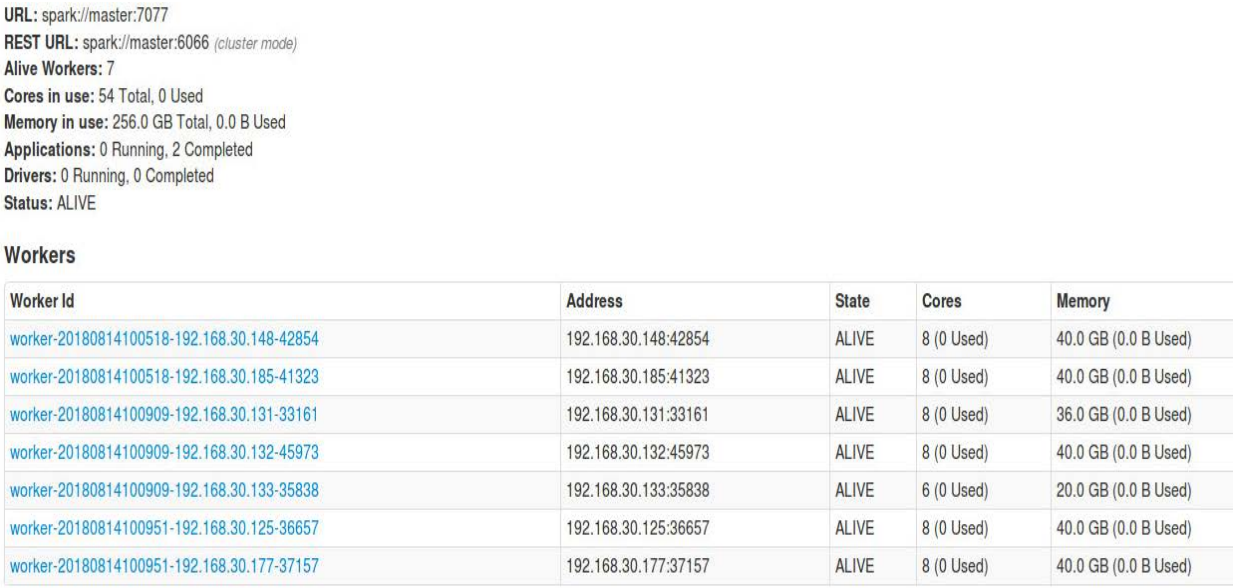
"Campus card" system as an important part of the construction of digital campus, has been popular in major universities. Large amount of consumption data generated by the system have attracted everyone's attention. The usual data analysis includes calculating the average, maximum and time point of students' consumption, so as to understand students' consumption outlook. This paper analyzes the daily consumption behavior of any student to determine whether the student has a fixed partner. The analysis results can provide reference for future analysis of students' character.

II. ANALYSIS SYSTEM DESIGN

The system is divided into three modular : data preprocessing modular, data analysis modular(core) and data visualization modular. The data preprocessing modular includes data cleaning, transformation, selection and other operation steps. The data analysis modular imported preprocessed data into the Spark cluster, and then the correlation analysis are performed. The results of the analysis are presented in a graphical way, so as to give the user an intuitive feel.

III. ANALYSIS SYSTEM IMPLEMENTATION

The system development framework includes JDK1.8, Hadoop2.6.0, Spark1.6.3, Scala2.10 and Zeppelin0.6.3 [1]. The spark cluster consists of 8 nodes, the 7 working nodes of them have 54 cores and 256G memory. The spark cluster configuration is shown in the following Fig. 1.



URL: spark://master:7077
REST URL: spark://master:6066 *(cluster mode)*
Alive Workers: 7
Cores in use: 54 Total, 0 Used
Memory in use: 256.0 GB Total, 0.0 B Used
Applications: 0 Running, 2 Completed
Drivers: 0 Running, 0 Completed
Status: ALIVE

Workers

Worker Id	Address	State	Cores	Memory
worker-20180814100518-192.168.30.148-42854	192.168.30.148:42854	ALIVE	8 (0 Used)	40.0 GB (0.0 B Used)
worker-20180814100518-192.168.30.185-41323	192.168.30.185:41323	ALIVE	8 (0 Used)	40.0 GB (0.0 B Used)
worker-20180814100909-192.168.30.131-33161	192.168.30.131:33161	ALIVE	8 (0 Used)	36.0 GB (0.0 B Used)
worker-20180814100909-192.168.30.132-45973	192.168.30.132:45973	ALIVE	8 (0 Used)	40.0 GB (0.0 B Used)
worker-20180814100909-192.168.30.133-35838	192.168.30.133:35838	ALIVE	6 (0 Used)	20.0 GB (0.0 B Used)
worker-20180814100951-192.168.30.125-36657	192.168.30.125:36657	ALIVE	8 (0 Used)	40.0 GB (0.0 B Used)
worker-20180814100951-192.168.30.177-37157	192.168.30.177:37157	ALIVE	8 (0 Used)	40.0 GB (0.0 B Used)

Fig. 1. The spark cluster configuration

This work was financially supported by Key Laboratory of Trusted Cloud Computing and Big Data Analysis (Nanjing XiaoZhuang University), Jiangsu Engineering Research Center for Networking of Elementary Education Resources (BM2013123).

Data analysis modular mainly implements data preprocessing, data analysis and integration of analysis results. This part uses spark cluster to process data. Because spark is based on memory processing, its biggest advantage is fast to deal with large-capacity data [2].The preprocessed data is stored in the distributed file system (HDFS), submit applications through spark-submit, spark will create and construct a Driver process, the Spark Context initialization, master and worker will respectively schedule node resource and task and execute defined the operators and functions until all operations executed so far. Finally, the processed data is stored on the HDFS [3].

Data visualization modular mainly implements the standardization, visualization, concrete and aesthetic appearance of the conclusion data obtained after the early excavation, and makes a visual secondary analysis for some of the intermediate data with large data volume. The system used the Zeppelin framework based on scala language as the data visualization modular. Zeppelin is the web-based notebook, which enables data-driven, interactive data analytics and collaborative documents with SQL, Scala and more [4].

IV. KEY ALGORITHM OF DISTRIBUTED DATA ANALYSIS

The main task of data analysis is to calculate the time correlation of students' consumption records. This task requires statistics of the number of times consumed of both of the students at a specific time interval, and finally sums up the number of times that both of the students consumption at the same time interval in a year. Because the amount of computation involved in this task is huge, all data are stored with open source HDFS and distributed computing framework spark is used for data analysis. [5]

The data set is a total of 18 million consumption records of all students in one year. The student consumption data attributes include card coding, consumption area code, consumption date and consumption time, consumption amount, etc. The process of data analysis is as follows:

The program read data from file and generated RDD map. The key code for data processing is as follows:

```
//defined class Consume

case class
    Consume(FLW_ID:String,FLW_ACCID:String,FLW_BL
    TID:String,
    FLW_CRDID:String,FLW_AREID:String,FLW_SERNO:
    String,FLW_TRDID:String,
    FLW_CLIENTNO:String,FLW_INSTIME:String ,
    FLW_CONTIME:String,
    FLW_AMOUNT:String,FLW_BALANCE:String,
    FLW_OPTID:String)

val dfConsume =
sc.textFile("hdfs://master:9000/input/ttnsmflwing-
max.txt").map(_.split(","))
    .map(p=>Consume(p(0),p(1),p(2),p(3),p(4),p(5),p(6),p(7),
p(8),p(9),p(10),p(12), p(13))).toDF()

//data result attributes(consumption card code, consumption
area code,consumption time) and consumption area code only
includes 01(south restaurant),03(north restaurant),05(students
service center)

Val
    df1=dfConsume.select("FLW_ACCID","FLW_AREID","
    FLW_CONTIME") where ("FLW_AREID='01' OR
    FLW_AREID='03' OR FLW_AREID='05'")

//generated key-value pairs, which key (consumption date -
consumption area code), value (consumption time, account
code), and convert consumption time units to minutes

val sdf = new SimpleDateFormat("yyyy-MM-dd
hh:mm:ss.SSS")

val myrdd = df1.rdd.map(p =>

(p(2).toString.split("")(0)+"-
"+p(1).toString,((sdf.parse(p(2).toString).getTime/60000).toIn
t,p(0))))

 //According to the consumption date, the consumption area
and the consumption time are sorted in order

val myrdd1=myrdd.sortBy(f=>(f._1,f._2._1))

//RDD map carries out inner join with itself,and tt will get the
pair of  two consumption cards on the same day in the same
place, and then selected the time interval of consumption less
than 10 minutes between two cases

val
rdd2=myrdd1.join(myrdd1).filter(f=>f._2._1._2 !=f._2._2._2)

        .filter(f=>f._2._1._2.toString.toInt <
f._2._2._2.toString.toInt)

        .filter(f=>(f._2._1._1-f._2._2._1<=10))

//generated new key-value pairs, which key (consumption card
code - consumption card code), value (1)

var rdd3=rdd2.map(x=>(x._2._1._2+"-"+x._2._2._2,1))

var rdd4=rdd3.sortByKey()

//collected values based on key

var rdd5=rdd4.reduceByKey(_+_)

//generated new result file
```

rdd5.saveAsTextFile("hdfs://master:9000/output/tmax-144-10.txt")

The above operation takes 25 minutes, and the data format of the result file attributes include consumption card code - consumption card code and numbers. The data format is shown in the following Fig.2.

```
(2000034012-2000040175,27)
(2000033653-2000037539,35)
(2000025545-2000025713,21)
(2000033822-2000038341,3)
(2000027113-2000036851,1)
(2000026912-2000034329,27)
(2000026770-2000038246,17)
(2000012334-2000027748,3)
(2000025505-2000035271,2)
(2000011955-2000033408,13)
(2000028649-2000030720,136)
```

Fig. 2. The data format of the result file

Then do some data processing for the above results.The key code for data processing is as follows:

```
val regex1="[0-9]+".r

val myrdd=sc.textFile("hdfs://master:9000/output/tmax-144-10.txt/",144).map(_.split(","))

    .map(p=>(p(0).toString.split("-")(0).split("\\(")(1)+"-"+p(1).split("\\)")(0).toInt,

      p(0).toString.split("-")(1)+"-"+p(1).split("\\)")(0).toInt))

    .flatMap(x=>x.toString().split(",")).map(_.split("-"))

    .map(x=>(regex1.findFirstMatchIn(x(0)).get.toString().toInt,regex1.findFirstMatchIn(x(1))

      .get.toString().toInt));

val dfmyrdd=myrdd.map(x=>Consume(x._1.toString,x._2.toInt)).toDF()

dfmyrdd.registerTempTable("alluserstable")

val groupmyrdd=sqlContext.sql("select ID_Pair,max(Counts1) maxcounts from alluserstable groupby ID_Pair").toDF()

//get UserId and intersecting times with others

val intersectings=dfsubmyrdd3.join(groupmyrdd,"ID_Pair")select("ID_Pair","maxcounts")

val dfConsume=sc.textFile("hdfs://master:9000/output/consumtimes.txt/",144).map(_.split(","))

    .map(x=>ConsumeTimes(x(0).toString.split("\\(")(1),x(1).split("\\)")(0).toInt)).toDF()
```

//get cardId , intersecting times with others, consumption times

```
val allresult=groupmyrdd.join(dfConsume,"ID_Pair")

    select("ID_Pair","maxcounts","ConsumeTimes")

allresult.rdd.saveAsTextFile("hdfs://master:9000/output/userId-consumTimes-10-all.txt")
```

The data format of the new file attributes include card Id, intersecting times with others, and consumption times. The data format is shown in the following Fig.3.

```
[2000033770,282,678]
[2000033888,76,232]
[2000033932,361,701]
[2000034166,323,438]
[2000034210,283,589]
[2000034328,293,587]
[2000034661,306,610]
[2000034779,475,421]
[2000034823,510,497]
[2000035057,182,889]
```

Fig. 3. The data format of the new file

V. ANALYSIS RESULTS OF THE DATA

Based on the above data, The distribution of students' consumption in school in one year can be obtained. The distribution of students' consumption times in school in one year is shown in Fig. 4.

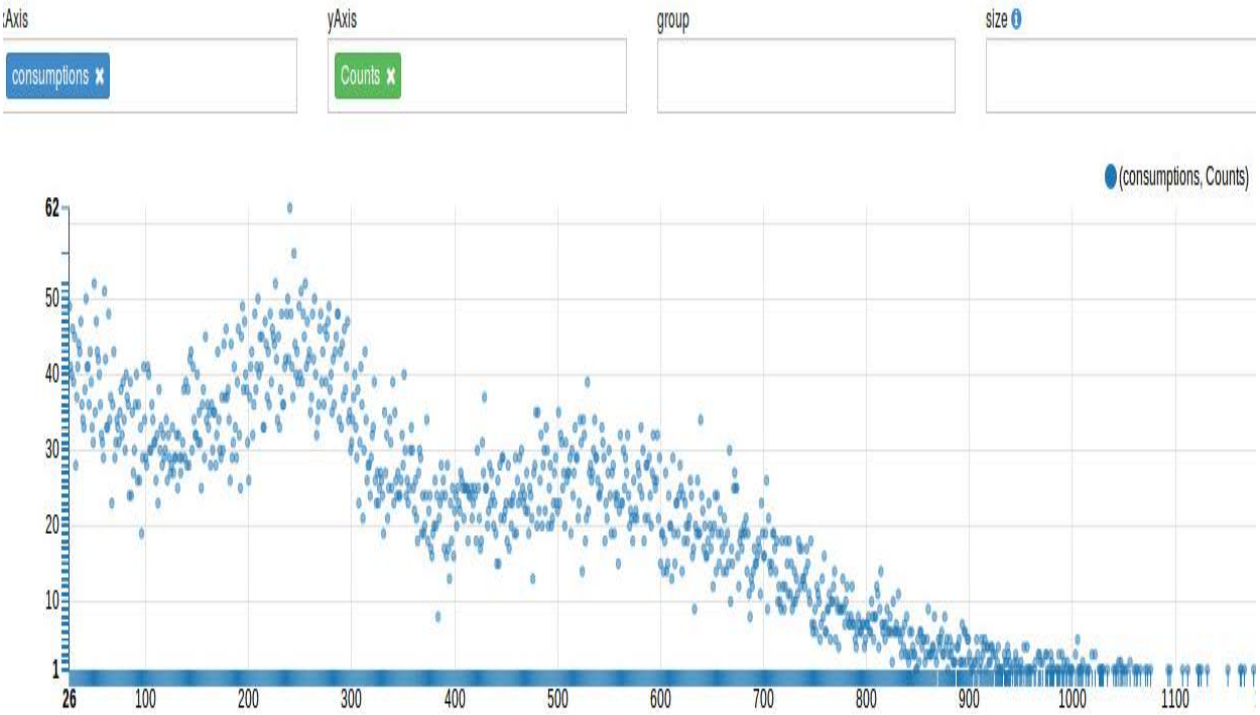


Fig. 4. The distribution of students' consumption times

By selecting 10 minutes for the calculation of the time interval, the distribution of students' consumption intersecting times with others in school in one year is shown in Fig.5.

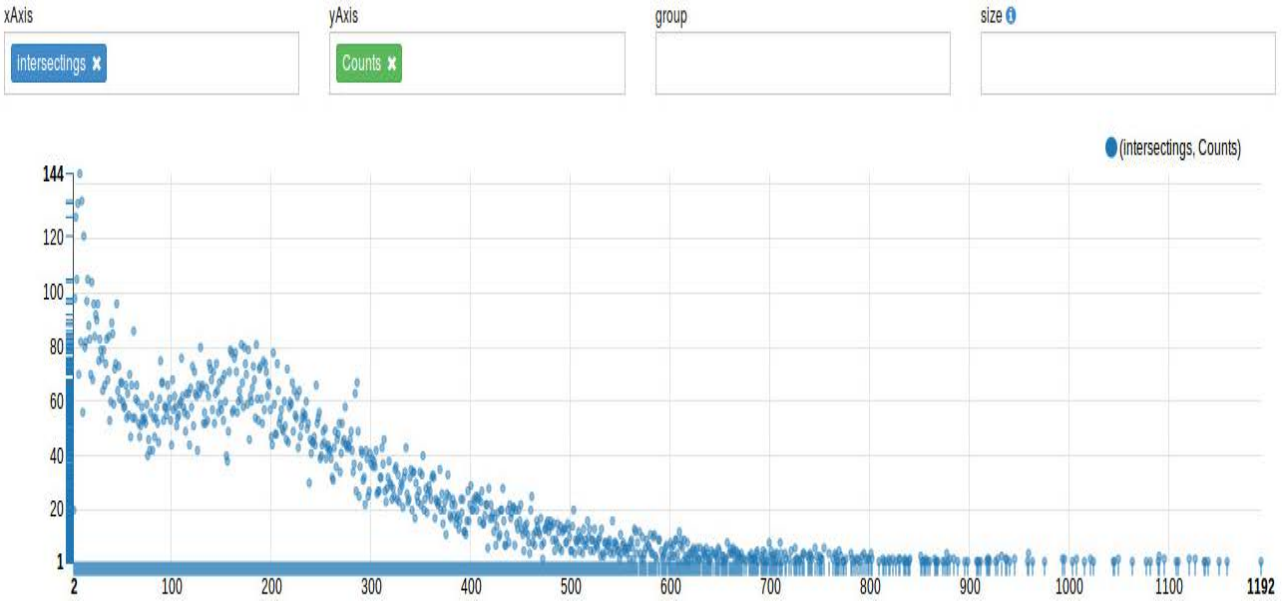


Fig. 5. The distribution of students' consumption intersecting times with others

The distribution of students' consumption intersecting times with others and consumption times is shown in Fig.6. Light color represents consumption times, and deep color represents consumption intersecting times with others.

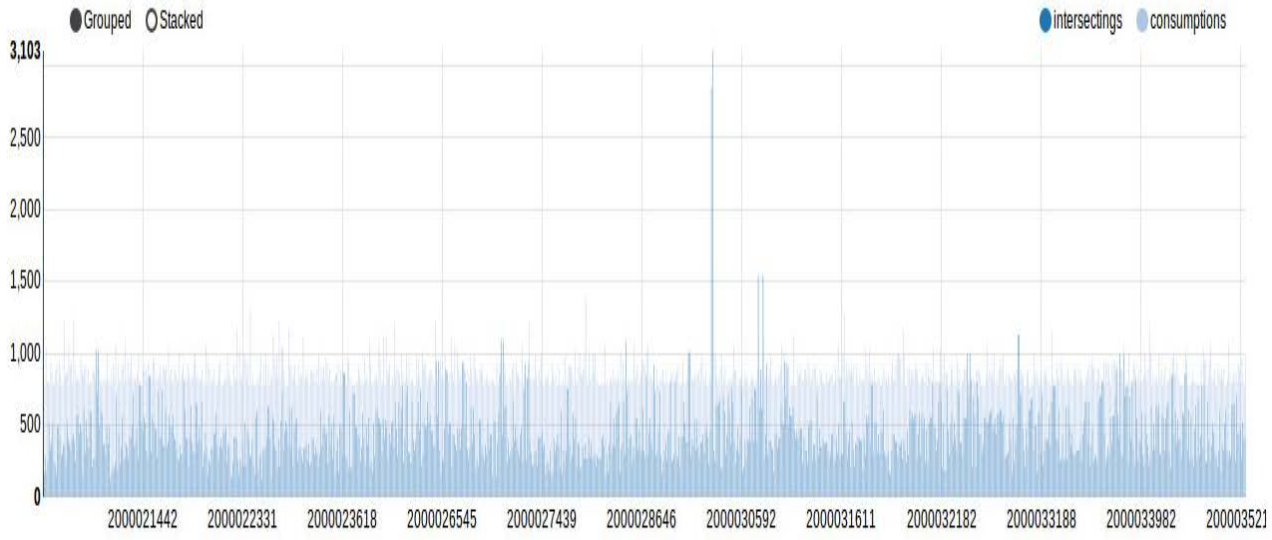


Fig. 6. The distribution of students' consumption intersecting times with others and consumption times

Based on the above data, the information of part of students whose consumption intersecting times with others and consumption times with large gender gap can be obtained. The information of part students is shown in Fig.7.

cardID	intersectings	consumptions	diff
2,000,027,690	67	569	502
2,000,029,104	66	453	387
2,000,026,680	74	434	360
2,000,021,798	48	434	386
2,000,004,792	74	415	341
2,000,022,029	71	410	339
2,000,022,499	72	408	336
2,000,025,681	78	404	326
2,000,027,851	78	403	325

Fig. 7. The information of part of students

VI. Conclusion

In the current era, the results of large-scale data analysis(such as large-scale research results) have played an important role in decision-making[6]. The purpose of this system is to mining the huge data stored on campus, so as to reuse the data which has been neglected for a long time.

Through analysis of the above result data, it can be seen that the number of college students spending on campus within a year is about 300 times. This shows that the main activities of students are still on campus. The ratio of the number of simultaneous consumption of any two students in a year to the number of consumption is about 30 percent, and the ratio of a few of the students is likely to be small. This part of the students needs more attention from schools, and most students have regular partners.

Acknowledgment

This work was financially supported by Key Laboratory of Trusted Cloud Computing and Big Data Analysis (Nanjing XiaoZhuang University), Jiangsu Engineering Research Center for Networking of Elementary Education Resources (BM2013123).

References

[1] Kui Zhang, “Make big data systems yourself,” in Publishing House of Electronics Industry, 2016, pp. 1-13. (In Chinese).

[2] Information on http://spark.apache.org/mllib/.

[3] Information on http://spark-summit.org/.

[4] Information on http://zeppelin.apache.org/.

[5] WEI Xiao-xiao, Design and Implementation of Campus Information Analysis System Based on Spark, Computer Engineering & Software, Beijing, vol. 38 no. 10, pp. , 2017. (In Chinese).

[6] Peng Liu, “Big Data,” in Publishing House of Electronics Industry, 2017, pp. 1-13. (In Chinese).