Advanced Clustering Algorithm on Spark

1st Jingyu Ma
Computer Science Department
University of Nottingham (Ningbo China)
Ningbo, China
626164783@qq.com

2nd Hao Mao
Computer Science Department
University of Nottingham (Ningbo China)
Ningbo, China
scyhm1@nottingham.edu.cn

2nd Tianyu Tong
Computer Science Department
University of Nottingham (Ningbo China)
Ningbo, China
scytt2@nottingham.edu.cn

2nd Mingchen Li
Computer Science Department
University of Nottingham (Ningbo China)
Ningbo, China
scyml3@nottingham.edu.cn

Abstract*—In recent years, the research of big data machine learning and data mining parallelization algorithm has become a relatively important research hotspot in the field of big data. In the early years, researchers and industry in China and abroad paid more attention to the design of parallelization algorithm on Hadoop platform. However , with the emergence and advantage of Apache Spark , in recent years, domestic and international attention has been paid to how to implement various machine learning and data mining parallel algorithm design on Spark platform. Here mainly discussed K-means , DBSCAN and BIRCH . With analysis on such ML algorithm and big data technology . The final result is some code based on single node data processing *(standalone sparks).

***Keywords*—Spark, DBSCAN, parallel, BIRCH, clustering algorithm , python , pyspark**

I. Introduction

Coming straight to the point , ruling out the topic of big data , the design of these clustering algorithm is somehow limited , as it will only work perfectly on artificial dataset or dataset that has clear distinction(So does some other machine learning algorithms) . These machine learning algorithms will work on any kind of dataset it sees , so sometimes you will see the irony of doing clustering , the worker has to be understanding the data , and to compare the clustering result , we need the ground truth , which can only be collected and processed by human , not machine . Here are some examples [1] on DBSCAN , which you can play around with these man-made dataset that have no meaning at all . We can see that it is hard to find a fit-for-all clustering algorithm , and what we can do is that we need to adjust ε(epsilon) , which is the radius , size of the circle , and minPoints , which is the minimum points within the circle to be consider this point into the cluster . This kind of act as zombie infection . And you might always get some ridiculous result if not properly setting the parameter ,say there is only one cluster or mis-cluster so many data samples into different categories that have no meaning at all, and all we can do is to stick to the ground truth , and fit the dataset to the ground truth as close as possible (feels like buying a lottery , you never know which size would be the best).

While algorithm will take on whatever dataset it meets , and in most cases not accurate enough , the computers would say no if the data grow too big , current algorithm would stop outputting at around 1000 samples , the testing time is 10 minutes without too much stopping criteria . If we met a dataset that is larger than 1000 say 10000 , we could use spark to let several computer calculate the machine learning algorithm , and then combine the clustering result together (which will make the result more not accurate since the calculation of such data is separate) . And saying a dataset that has more than 10000 samples , that is way beyond what individual and small research group could do , and it makes the collection of such data very arduous .We would leave such questions to the government , more funded groups , companies and specially-designed computers that might have 256 bits registers , monster level hard drive , super CPU , tremendous memory , all-star hardware , have 256 bits formatting integers , which could process 2^8 times larger integers(0.21 x 2^8billion in decimal) and an algorithm that would stop at finite steps . And if all above scenario doesn't work , we could manually break the data into 100 – infinite parts and use USB driver to copy such files from one computer to another to do the work that spark should do.

While processing big data is a little bit hard , the collecting work for a small team is also extremely hard . But we can guarantee that our data is real and hand-collected .

II. Justification of Data

A. Iris Flower

The dataset is very famous , some people were mentioned in Wikipedia[2] , first appeared as biology dataset from by the British statistician, eugenicist, and biologist Ronald Fisher in his 1936 paper. And then it is sometimes called **Anderson's Iris data** set because Edgar Anderson collected the data to quantify the morphologic variation of Iris flowers of three related species.

The reason for choosing this dataset is that this is a ground truth dataset . And these flowers has distinctive features . It has already been separated apart , so you could also let machine do the work and see if the result works well. But probably all we can do is let this algorithm stick to the truth as accurate as possible . Such as set the K to be 3 , adjust the initial clustering centroids in K-means, adjust the circle size in DBSCAN , set the minPoints larger or smaller .This is called parameter tuning .Which again , violated our initial goal to not supervised on the machine , and give out the result .

B. Trying clustering on other dataset

These days the team leader have also being trying to collect data by himself which he found this process to be very inane and meaningless . And most probably , in some data centers , people are already gathering data , with written programs . Not in a such rudimentary way .

Our team is so excited that we have already collected some data on the selling price , publisher , developer economic condition , required graphics card , card price , release date , rating etc .

But unfortunately , we found out that in Metacritic , IGN , these **ratings are sometimes incomprehensible, cannot influenced by the public review** , but it is a number that is given by a person or a group of people , so far it is not sure if such score will impact on human decisions . Actually the time and energy of such organization is limited , of course they cannot do review on every product and give each one a rating , so **indie games and not-so-famous games would sometimes be ignored by the critics** .It is also noticeable that this dataset , seldom contains nameless games or not popular ones , since when you want to buy a game , you only want to buy something that is known by others or recommended by others . The result in the data is all of them are high-quality games .

Also some graphics cards are stopped to produced anymore , the only way to figure its price is use second-hand price , which is given by people . and sometimes ridiculous , say someone would give it for free , or someone might sell it higher than modern graphic cards .Leader has also make the ranking list[3] of Graphic card available(2020.11.19) , digitalized it , we could also put it into a database for easy searching .

However , trying other datasets would be meaningless if such dataset is not labelled . If it is already labeled then , we don’t necessarily have to use computers to do it again . Such implementation on Spark could have already be implemented by the government and larger company .

III. TEST RESULT

Some noticeable progression are made out by a guy from UCLA , about BIRCH [4] . But our team seriously question the truth of such test , but let’s just assume it is OK and it doesn’t bother the remaining contents. Meanwhile our group have implemented a standalone Spark for DBSCAN(which uses threads to simulate multiple machines working) , it is a shame that the maximum sample can be taken is around 2000-3000 samples .And the dataset given by MLlib is extraordinary small[5] , we haven’t try the limit of these code , but it seems not going to be too large , you could try it yourself , the guess is , it is not going to hold too much . The code also doesn’t put label into the dataset , which is another point we need to innovate on . Try K-means in the reference link [6] , and if you want to input data , make sure it is big enough , to make this more suitable to the topic of “Big Data” . But I guess you will meet with “Stack Overflow” or some similar condition(Any student who learned programming , will met this in travelling salesman , it will allocate variable automatically to stack , unless you use dynamic allocation) .The dataset given is not convincing .

And besides the dataset , the algorithm itself is problematic regardless of if we are simulating the nodes . We have to break the dataset apart first , let each node (computers)do their own work , and then combined the colored(clustered or sorted) results together .**This is not as simple as word count . If we decided to let each node do computing , it means each result is separated , independent ,and if combine them together , more errors will occur . And the uneven distribution in different nodes would also result in errors , you know , one node is given 4 samples , and another is given 400 samples .**

Anyway , we will still give you a brief on the idea of such concepts and here I gathered all the results and codes to let you see.

A. K-means

Based on team leader’s humble opinion, this one is the most easiest one to comprehend among all other algorithms. You may want to refer to some of the machine learning slides from Nottingham [7], or you could take Professor Andrew Ng’s open course here[8] , or we think this YouTube channel called StatQuest with Josh Starmer very helpful [9] .

Basically you need to first identify how many kinds of things exists in the data(Which is used for the value K) . In Iris flower there are three species that are very different . This requires you to have some pre-knowledge on the data .Otherwise , you will have to use elbow method to make sure which K to use , which again you have to write a loss function , and you will have to know more about this dataset .

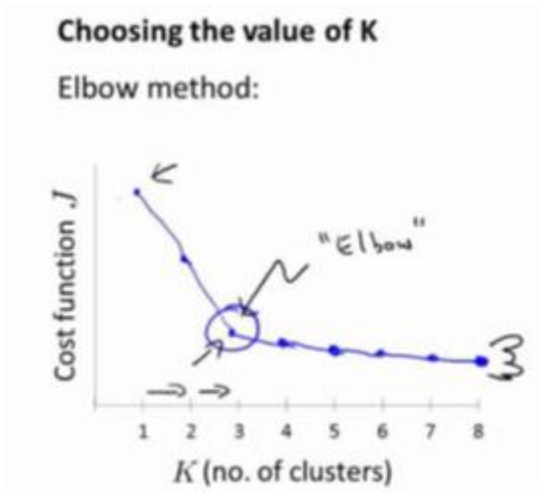


Fig. 1. Elbow method

Even if we could write a cost function for finding K , such cost function is independent and won’t help what we will talk about later .And the elbow point really depends on experience and pre-knowledge(Why not go for 2 ? It drop the most sharpest? Or 7 , you know it is much better than 3 .) .

You can random initialize 3 cluster centroid (or in some variation of this algorithm , you can alter the initial point a little bit) , the cluster centroid is used for gathering the points , which means measure the point to point distance (Euclidean Distance)to all the points , and see which centroid is near , and then assign(color) to the point to that centroid . Like in real world , you want to go to toilet , you looked at a map or nowadays (Baidu , Google Map) ,measure the distance between you and the toilet , and then go to the nearest one .

So normally speaking , after 1st iteration(round) , there has already been a result , but we were not satisfied , or someone first designed the algorithm thinking it is unreasonable , so he want to moved the toilet a little bit , he then calculated all the points , which you can assume a point is a person , add them together , take the average of X sum and Y sum , a new average point , a centroid is formed . And then normally the algorithm stopped when none of the centroid is moving . If you are not satisfied there are also other stopping methods you can choose [10].

Well if you really understand the above contents , then you probably already know the following situations (scenarios) , so this is one of the main contents of this paper . **So , it is almost a certain that machines will cluster the point into a wrong clusters (categories) , I wasn't certain of this , and I see a guy's video [11] and he did some experiments , I redid this experiment to make sure this guy is speaking the truth , also a guy contributed to our code tells me the same thing , and the output also tells me the same thing .**

For K-means , the wrong clustering is mainly on the border , and the cluster (group of people finding the toilet) , is in a sphere(circle) , since you know , it measure the distance between points (Euclidian distance) . And here are some typical wrong examples in Iris Flowers .

Please see the result below in Fig 1 . We believe other English speaking people should have done it also . If you can find someone speaking mandarin you can know its about PetalLengthCm and PetalWidthCm .Or you are so familiar with iris flower , you will remember the parameter once you saw the shape of the graph .

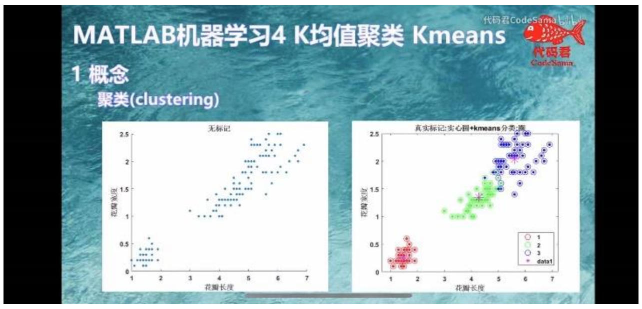


Fig. 2. Left is unsorted data and right is the real human collected result or human definition. (*Thanks for this guy on bilibili !*)

And a real classification in Fig 2

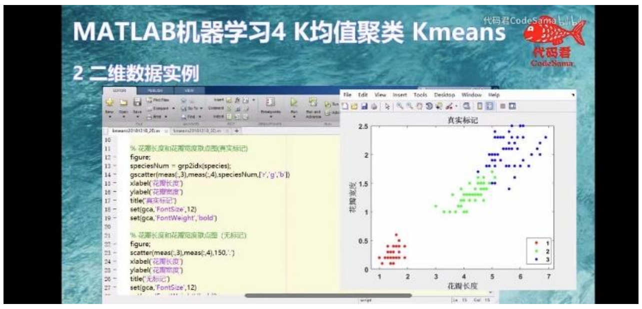


Fig. 3. Real marking. (*Or maybe someone who set the dataset to be like this*)

And a machine output here in Fig 3

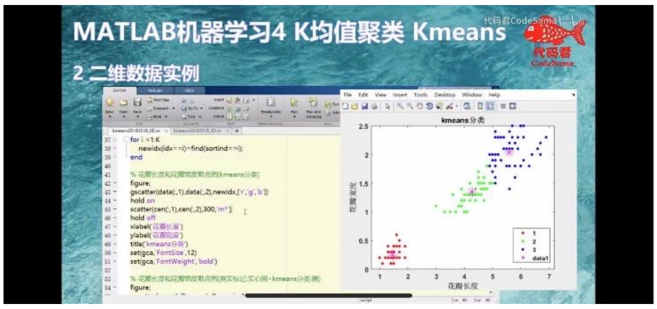


Fig. 4. K-means clustered result.

And here he compared the results on the border side in Fig4.

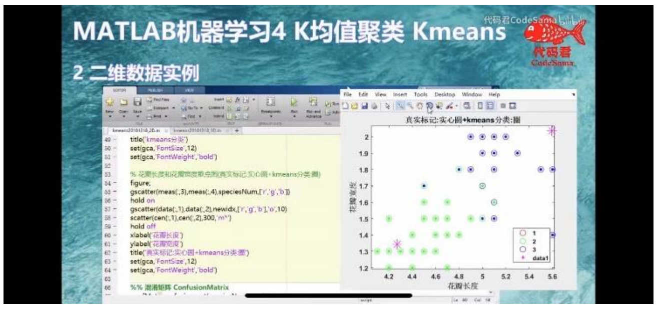
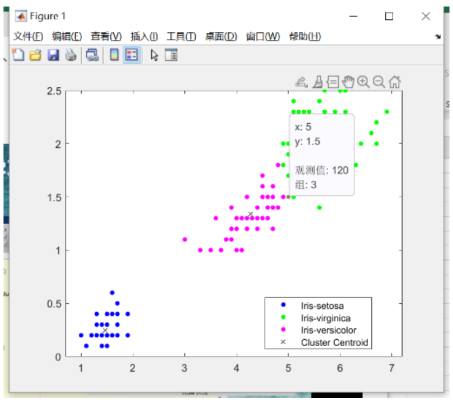
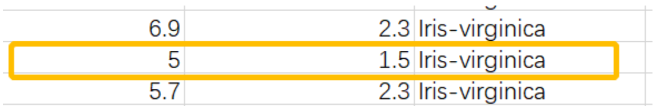

MATLAB机器学习4 K均值聚类 Kmeans
代码君CodeSama
2 二维数据实例
真实标记:实心圆+kmeans分类:圈
花瓣长度
花瓣宽度

Fig. 5. Comparing the result.

And the leader did it himself , here underneath is from his MATLAB. And his dataset . In Fig 5 .



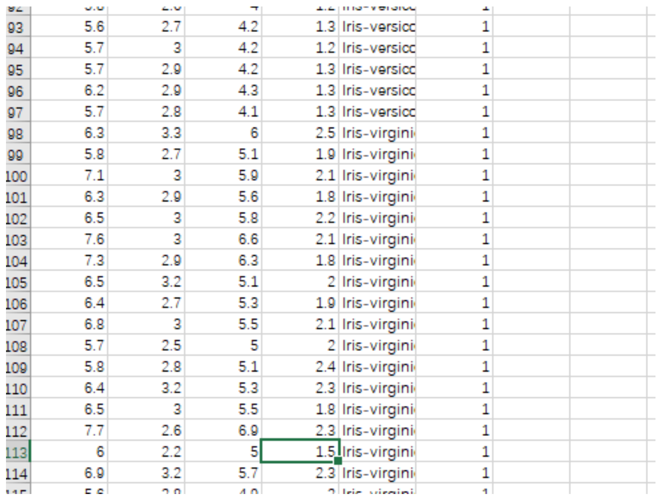


6.9	2.3	Iris-virginica
5	1.5	Iris-virginica
5.7	2.3	Iris-virginica

Fig. 6. For your convience I have name the clusters and see the original name and the clustered name.

I am excited to announce that , we have made some progress on combining Spark with K-means . And also someone started this in the documentation , however didn't give out the complete code .

However , it seems we need to adjust the parameter , since as you can see , a lot of outputs are wrong ! See Fig 6



93	5.6	2.7	4.2	1.3	Iris-versicc	1		
94	5.7	3	4.2	1.2	Iris-versicc	1		
95	5.7	2.9	4.2	1.3	Iris-versicc	1		
96	6.2	2.9	4.3	1.3	Iris-versicc	1		
97	5.7	2.8	4.1	1.3	Iris-versicc	1		
98	6.3	3.3	6	2.5	Iris-virgini	1		
99	5.8	2.7	5.1	1.9	Iris-virgini	1		
100	7.1	3	5.9	2.1	Iris-virgini	1		
101	6.3	2.9	5.6	1.8	Iris-virgini	1		
102	6.5	3	5.8	2.2	Iris-virgini	1		
103	7.6	3	6.6	2.1	Iris-virgini	1		
104	7.3	2.9	6.3	1.8	Iris-virgini	1		
105	6.5	3.2	5.1	2	Iris-virgini	1		
106	6.4	2.7	5.3	1.9	Iris-virgini	1		
107	6.8	3	5.5	2.1	Iris-virgini	1		
108	5.7	2.5	5	2	Iris-virgini	1		
109	5.8	2.8	5.1	2.4	Iris-virgini	1		
110	6.4	3.2	5.3	2.3	Iris-virgini	1		
111	6.5	3	5.5	1.8	Iris-virgini	1		
112	7.7	2.6	6.9	2.3	Iris-virgini	1		
113	6	2.2	5	1.5	Iris-virgini	1		
114	6.9	3.2	5.7	2.3	Iris-virgini	1		

Fig. 7. My Output with no visuliztion.

However we can see that such thing can be distributed implemented in Spark . And **the sample** just did this in Spark , give out centroids , but it was **highly abstracted(no detail)** . And it would be very hard to test the limit of the algorithm since the input was formatted , and if you want to add more , you will have to manually made this data . The sample only got 3 samples , which is a pretty "Big Data".

You can find the sample here[5][12].

It is also worth noticing that , it is also possible to do this in 3-d , but with more errors . [11] And the other parameters , SepalLengthCm , SepalWidthCm are not so distinguishable.

If you want to find out more , please check out this Naftali guy's visualization on K-means .[6]

But we got one self made Spark K-means method on Iris Flower[21] . And now the sample is running as well . It should be testament that it is possible that such technology could be combined and it could be working on a much larger scale than just standalone Spark (With more errors!) .

B. *DBSCAN*

Here comes our main dish . So this is not included in the Spark MLlib. DBSCAN stands for **Density-Based Spatial Clustering of Applications with Noise .** The leader felt very tired at explaining all these ! But he guessed by the time this thing is published , he will still explain it once again ! So he don't mind explain it here ! According to MATLAB's document which should be one of the most classical example of this algorithm [13]. Well remember the zombie infection example ? It is like this , so you first choose a point maybe randomly , draw a circle that is radius in epsilon , see if it reaches minPoints number of points within that circle ,if it is not , then it is isolated , and it is called Outlier , noise point , if there is , then this point is a core point , which act as a zombie mother that is capable of infecting others , then all the points inside the circle are all known as neighborhood points , you put these infected points and the core points into a cluster , and started to see if a unvisited point inside neighborhood points can be upgraded into a core point , zombie mother , and if it does , then it will infect other points as well . Do so until all neighborhood points are visited . Then start from points are not in the cluster , then we got another cluster . Do so until every point was given a label .

You could also play around with this program developed by Naftali Harris [14]. Or there is a video from UIUC [15] , to help you better understand this thing .

As required , I have implemented this on cslinux , UNNC , and the program can output as standalone spark . But put aside Spark , running DBSCAN requires parameter tuning as well .Such as epsilon and minPoints , and you will get very different results . Below is an experiment conducted on Iris Flower about DBSCAN , and please view the link and code here [16] .The result showed very distinctive output.

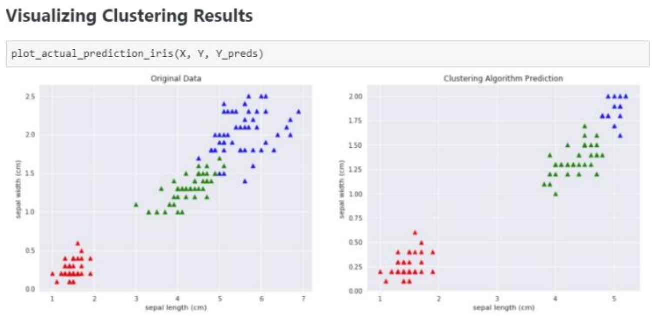
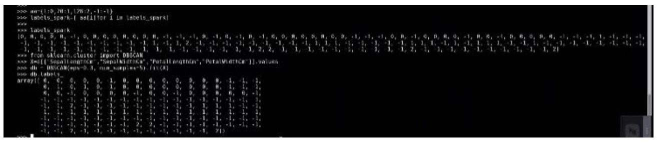


Fig. 8. A visulization on one possibe outcome of DBSCAN on Iris Flower



```
>>> aa={1:0,70:1,128:2,-1:-1}
>>> labels_spark=[ aa[i]for i in labels_spark]
>>>
>>> labels_spark
[illegible]
>>> from sklearn.cluster import DBSCAN
>>> X=df[['SepalLengthCm','SepalWidthCm','PetalLengthCm','PetalWidthCm']].values
>>> db = DBSCAN(eps=0.3, min_samples=5).fit(X)
>>> db.labels_
array([ [illegible]
```

Fig. 9. My output

From the result we can see that , a lot of points were move into the cluster of noise points . This indicates that clustering works not very well on natural dataset such as Iris Flower . But it still preserve the capability to distinguish the dataset apart , as we can see on artificial shapes .

Then we would move onto the topic of modifying DBSCAN , they is also an attempt that no one in the world have ever done before . And here we want to give you the detailed Boolean expression of square-based DBSCAN .

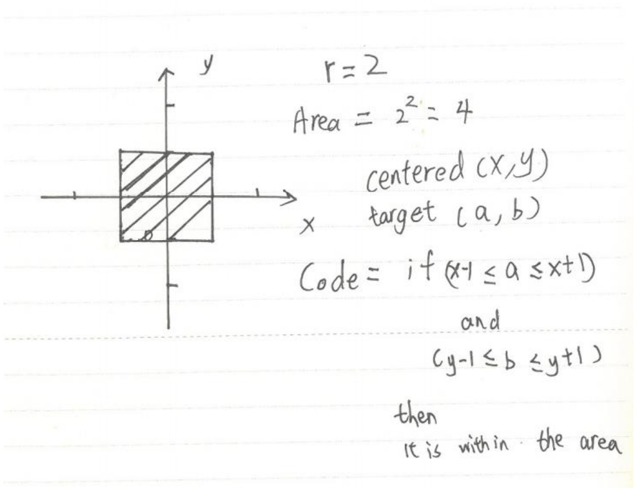


Fig. 10. Square-based DBSCAN

So for square-area the points within the region satisfy the following criteria , given square side length equal to r and random point (a, b) inside :

$$-\frac{r}{2} \le a \le \frac{r}{2} \qquad (1)$$

$$-\frac{r}{2} \le b \le \frac{r}{2} \qquad (2)$$

All the rest of the settings are equal to traditional DBSCAN.

Here is another alternative of DBSCAN , we want to make the algorithm triangle-based . Then we can use Linear programming to calculate the area within the triangle .Below is a equilateral triangle centered at half of its height.

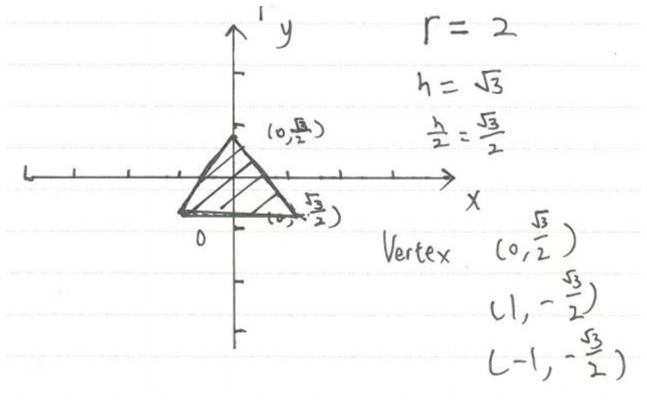


Fig. 11. Triangle-based DBSCAN

Given the side for this triangle pattern , the points around it , satisfy the following equations :

$$y \ge -\frac{\sqrt{3}}{4}r \qquad (1)$$

$$y \le -\sqrt{3}x + \frac{\sqrt{3}}{4}r \qquad (2)$$

$$y \le \sqrt{3}x + \frac{\sqrt{3}}{4}r \qquad (3)$$

Another way is to shift the central point of this area , it is still triangle . This is a triangle which is centered at its center of gravity , incenter , circumcenter , orthocenter .

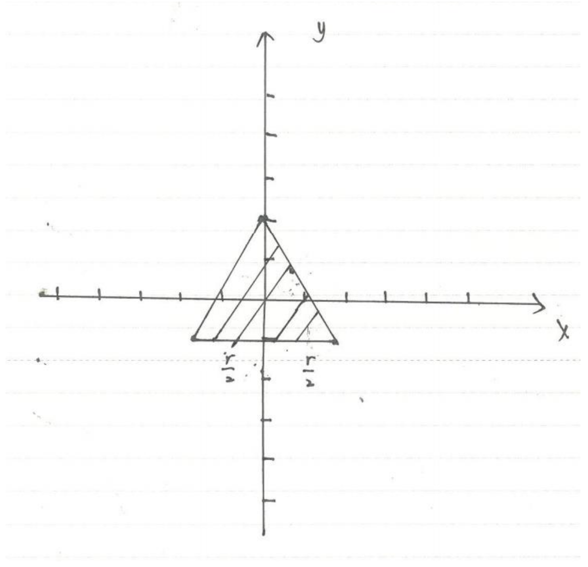


Fig. 12. Triangle-based DBSCAN

For this triangle the attribute of the points within the area satisfy the following :

$$y \le \sqrt{3}x + \frac{\sqrt{3}}{3}r \qquad (1)$$

$$y \le -\sqrt{3}x + \frac{\sqrt{3}}{3}r \qquad (2)$$

$$y \ge -\frac{\sqrt{3}}{6}r \qquad (3)$$

Lastly we came up with a solution that could altering the shape to a Cardioid which is first published by de Castillon in 《Philosophical Transactions of the Royal Society》 , 1741.

The formula our team choose is the classic Cardioid , which looks the line below first one in the picture .

$$\boldsymbol{r} = \mathbf{a}(\mathbf{1} - \mathbf{sin\theta})$$

Further formula and computer code require our team working with mathematicians .

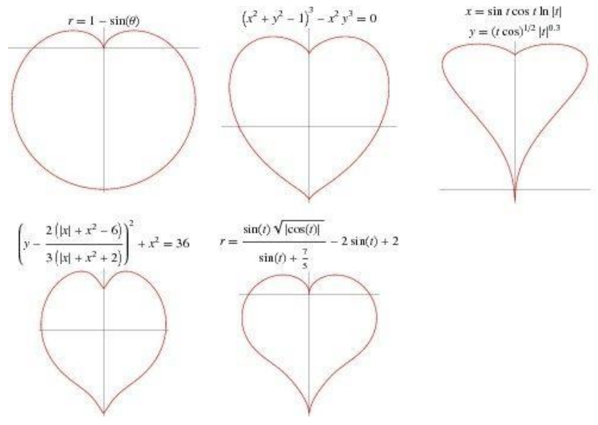


Fig. 13. Cardioid-based DBSCAN

Some other thinking about clustering algorithm will come out in the future , there is still a lot of place that can be modified , and some new algorithms can be created .

Here should also mention some other people who have tried Spark and DBSCAN[17].

C. Birch

Lastly I want to mention this hierarchical clustering algorithm . It has already be implemented and designed in multiple papers , the code is also available , here we just want to state that this method is proved to be successful by Xingyu Zhou from UCLA [3] , and several other papers [18][19][20] , academically . Thus cover most of the clustering algorithms in the paper .

IV. Future Work and Conclusion

A. Future Work

The future work which is believed to be , and implemented to be successful , which will be able to take real machines and much more big dataset .

There is still space in the current Spark Algorithm , that could support much larger dataset , even if it a man-made dataset , but that is way beyond what an undergraduate student team could do .

And of course , hope someone sees this could support us economically thus will make these classical algorithm go further !

B. Conclusion

This paper examines the frontier of machine learning , clustering algorithm , and big data . And seeks to innovate through the current barren reality of machine learning and data science . **Created two classic machine learning algorithm in Spark context which is not included in the MLlib of Apache Spark** [21].

It seems the machine learning in this area has reached its ceiling , and everyone has crashed into this wall , in the past and in the future . But this is maybe what the true natural of science do , young generations of people will carry on what the old generation do , and achieve what no one has done before , some would be meaningless , some , however , will push our race , all mankind forward .

While doing the project , it is also worth mentioning that the team leader is gathering a dataset on game sale , which in the future will be completed . And this might help with whoever is working on marketing and economic .

Our result is combined with Spark , and we use standalone spark , the real time spark is also implemented , if you see this and agreed to conduct such experiment we will , and we will move this thing forward to real computers instead of simulations .

A triangle , square , Cardioid based DBSCAN is also proposed . Which is believed to be state-of-art and none precedented. Equations are also given . Further variation could be achieved by collaborating with mathematicians .

Acknowledgment

I would like to thank the Supervisor **Prof. Saeid Pourroostaei Ardakani** , as a representative of FOSE CS 's excellent academic , technical support , super patient teaching , extraordinary computer background . **Dr. Richard Hoban** who left Nottingham a few years ago **,** as a representative of CELE’s prominent teaching force , and first introduced Microsoft office set , data representation technology to me . This is just to name a few , among all the faculties helped me along my graduation and college life .

Thanks to my teammate who supported me all along the module.

Additional acknowledgment and thanks to a programmer called **WenJie Cheng** who is the CTO of a company and another **anonymous programmer** who helped us fix bugs and provide additional technical support . And so many code sources online which is unable to reference .

Last but not the least , everyone helped me , teach me , talked to me , meet me on my life journey which is impossible to trace where they are now .

References

[1] Naftaliharris.com. 2021. Visualizing DBSCAN Clustering. [online] Available at: <https://www.naftaliharris.com/blog/visualizing-dbscan-clustering/> [Accessed 6 May 2021].

[2] En.wikipedia.org. 2021. Iris flower data set - Wikipedia. [online] Available at: <https://en.wikipedia.org/wiki/Iris_flower_data_set> [Accessed 6 May 2021].

[3] Baidu forum, 2021. 2021 显卡天梯图（含游戏性能帧数整理）. Available at: <https://tieba.baidu.com/p/6133450546> [Accessed 6 May 2021].

[4] 2021. Parallel BIRCH Clustering on Spark. 1st ed. [ebook] Los Angeles: Computer Science Department， UCLA, p.10. Available at: <https://github.com/XingyuGit/spark-birch/blob/master/parallel-birch-clustering-on-spark.pdf> [Accessed 6 May 2021].

[5] GitHub. 2021. apache/spark. [online] Available at: <https://github.com/apache/spark/blob/master/data/mllib/kmeans_data.txt> [Accessed 6 May 2021].

[6] Naftaliharris.com. 2021. Visualizing K-Means Clustering. [online] Available at: <https://www.naftaliharris.com/blog/visualizing-k-means-clustering/> [Accessed 6 May 2021].

[7] Nottingham University, 2021. Machine Learning. [video] Available at: <https://video.nottingham.edu.cn/Panopto/Pages/Viewer.aspx?id=67cc4209-b411-4780-8ead-ac5d00a816ab> [Accessed 6 May 2021].

[8] Stanford University, 2021. K-Means Algorithm. [video] Available at: <https://www.coursera.org/learn/machine-learning/lecture/93VPG/k-means-algorithm> [Accessed 6 May 2021].

[9] StatQuest with Josh Starmer, 2021. StatQuest: K-means clustering. [image] Available at: <https://www.youtube.com/watch?v=4b5d3muPQmA> [Accessed 6 May 2021].

[10] StackExchange, 2021. Stopping condition of K-means. Available at: <https://stats.stackexchange.com/questions/260917/stopping-condition-of-k-means> [Accessed 6 May 2021].

[11] 代码君 CodeSama, 2021. K 均值聚类 Kmeans _bilibili. [video] Available at: <https://www.bilibili.com/video/BV15t411v7Uj?from=search&seid=18090983629889571233> [Accessed 6 May 2021].

[12] Apache.googlesource.com. 2021. data/mllib/kmeans_data.txt - spark - Git at Google. [online] Available at: <https://apache.googlesource.com/spark/+/branch-1.2/data/mllib/kmeans_data.txt> [Accessed 6 May 2021].

[13] Mathworks.com. 2021. Density-based spatial clustering of applications with noise (DBSCAN) - MATLAB dbscan. [online] Available at: <https://www.mathworks.com/help/stats/dbscan.html> [Accessed 6 May 2021].

[14] Naftaliharris.com. 2021. Visualizing DBSCAN Clustering. [online] Available at: <https://www.naftaliharris.com/blog/visualizing-dbscan-clustering/> [Accessed 6 May 2021].

[15] University of Illinois at Urbana-Champaign, 2021. DBSCAN: A Density-Based Clustering Algorithm. [video] Available at: <https://www.coursera.org/lecture/cluster-analysis/5-2-dbscan-a-density-based-clustering-algorithm-MWDS8> [Accessed 6 May 2021].

[16] CoderzColumn, 2021. Scikit-Learn - Clustering: Density-Based Clustering of Applications with Noise [DBSCAN]. Available at: <https://coderzcolumn.com/tutorials/machine-learning/scikit-learn-sklearn-clustering-dbscan> [Accessed 6 May 2021].

[17] Huang, Fang et al., 2017. Research on the Parallelization of the DBSCAN Clustering Algorithm for Spatial Data Mining Based on the Spark Platform. Remote sensing (Basel, Switzerland), 9(12), p.1301.

[18] 李帅 吴斌 杜修明 陈玉峰 2017. 基于 Spark 的 BIRCH 算法并行化的设计与实现. 计算机工程与科学, 39(1), pp.35‑41.

[19] Zhang, T., Ramakrishnan, R. and Livny, M., 1996. BIRCH: An Efficient Data Clustering Method for Very Large Databases. 1st ed. [ebook] Montreal, Canada: ACM, p.12. Available at: <https://www2.cs.sfu.ca/CourseCentral/459/han/papers/zhang96.pdf> [Accessed 6 May 2021].

[20] Dong, J., Wang, F. and Yuan, B., n.d. Accelerating BIRCH for Clustering Large Scale Streaming Data Using CUDA Dynamic Parallelism. 1st ed. [ebook] Shen Zhen , China: Intelligent Computing Lab, Division of Informatics Graduate School at Shenzhen, Tsinghua University, p.8. Available at: <http://boyuan.global-optimization.com/Mypaper/IDEAL2013-88.pdf> [Accessed 6 May 2021].