ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 6, Issue 5, May 2018

Document Clustering Using K-Means videHadoop

Manisha Agrawal[1], Nisha Pandey[2]

P.G. Student (MTech), Department of Computer Science & Engineering, Shree Ram College of Engineering and Management at Palwal, Haryana, India[1]

Assistant Professor, P.G. Student (MTech), Department of Computer Science & Engineering, Shree Ram College of Engineering and Management at Palwal, Haryana, India[2]

ABSTRACT: Clustering is a useful data mining technique which group's data points such that the points within a single group have similar characteristics, while the points in different groups are dissimilar. Partitioning algorithm methods such as k-means algorithm is one kind of widely used clustering algorithms. As there is an increasing trend of applications to deal with vast amounts of data, clustering such big data is a challenging problem. Recently, partitioning clustering algorithms on a large cluster of commodity machines using the MapReduce framework have received a lot of attention. Traditional way of clustering text documents is Vector space model, in which TF-IDF is used for k-means algorithm with supportive similarity measure. This scheme or paper exhibits an approach to cluster text documents in which results obtained by executing map reduce k-means algorithm on single node cluster on hadoop show that the performance of the algorithm increases as the text corpus increases thus forming the non-redundant results and appropriate information**.**

KEYWORDS: Big Data, Hadoop, Yet Another Resource Negotiator (YARN), K-Means.

I. INTRODUCTION

K-means is a partitioned clustering technique that helps to identify k clusters from a given set of n data points in d-dimensional space. It starts with k random centers and a single cluster, and refines it at each step arriving to k clusters. Currently, the time complexity for implementing k - means is O (I * k * d * n), where I is the number of iterations. If we could use the KD-Tree data structure in the implementation, it can further reduce the complexity to O (I * k * d * log (n)).

Clustering based on k-means is closely related to a number of other clustering and location problems. These include the Euclidean k-medians in which the objective is to minimize the sum of distances to the nearest center and the geometric k-center problem in which the objective is to minimize the maximum distance from every point to its closest center. There are no efficient solutions known to any of these problems and some formulations are NP-hard. The large constant factors suggest that it is not a good candidate for practical implementation. One of the most popular heuristics for solving the k-means problem is based on a simple iterative scheme for finding a locally minimal solution. This algorithm is often called the k-means algorithm

K-Means is one of the simplest unsupervised learning algorithms that solve the well known clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters).The number of clusters should match the data. An incorrect choice of the number of clusters will invalidate the whole process. An empirical way to find the best number of clusters is to try K-means clustering with different number of clusters and measure the resulting sum of squares.

The basic K-means Algorithm is as follows:
Step 1: Select K points as initial centroids
Step2: Repeat
Step 3: Form K clusters by assigning each point to its Closest centroid
Step 4: Recompute the centroid of each cluster

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 6, Issue 5, May 2018

Step 5: Until centroids do not change.

The key step of Basic K-means algorithm is selection of proper initial centroids. Initial clusters are formed by random initialization of centroids. Sequence File from Directory of Text Documents Map reduce programming is coined to process huge data sets in parallel and distributed environment. Suppose, we select the input data from a document set, where the text files in the directory are small in size. Since HDFS and Mapreduce are optimized for large files, convert the small text files into larger file i.e., SequenceFile format. SequenceFile is a hadoop class, which allows us to write document data in terms of binary <key, value> pairs, where key is a Text with unique document id and value is Text content within the document in UTF-8 format. SequenceFile packs the small files and process whole file as a record. Since the SequenceFile is in binary format, we could not able to read the content directly but faster for read /write operations.

Doug Cutting and his team developed an Open Source Project called HADOOP, using the solution provided by Google.Hadoop is an Apache open source framework written in java that allows distributed processing of large datasets across clusters of computers using simple programming models. The Hadoop framework application works in an environment that provides distributed storage and computation across clusters of computers. Hadoop is designed to scale up from single server to thousands of machines, each offering local computation and storage.Hadoop is widely used in industrial applications with Big Data, including spam filtering, network searching, clickstream analysis, and social recommendation. To distribute its products and services, such as spam filtering and searching, Yahoo has run Hadoop in 42,000 servers at four data centers as of June 2012. Currently, the largest Hadoop cluster contains 4,000 nodes, which is expected to increase to 10,000 with the release of Hadoop 2.0. Simultaneously, Facebook announced that their Hadoop cluster processed 100 PB of data, which increased at a rate of 0.5 PB per day as of November 2012. According to Wiki, 2013, some well-known organizations and agencies also use Hadoop to support distributed computations. In addition, various companies execute Hadoop commercially and/or provide support, including Cloudera, EMC, MapR, IBM, and Oracle. With Hadoop, 94% of users can analyze large amounts of data. Eighty-eight percent of users analyze data in detail, and 82% can retain more data (Sys.con Media, 2011). Facebook stores 100 PB of both structured and unstructured data using Hadoop. IBM, however, primarily aims to generate a Hadoop platform that is highly accessible, scalable, effective, and user-friendly. It also seeks to flatten the time-to-value curve associated with Big Data analytics by establishing development and runtime environments for advanced analytical application and to provide Big Data analytic tools for business users. Hadoop runs applications using the MapReduce algorithm, where the data is processed in parallel with others. In short, Hadoop is used to develop applications that could perform complete statistical analysis on huge amounts of data.

II. LITERATURE REVIEW

Apache Hadoop is an open source software framework for storage and large scale processing of data-sets on clusters of commodity hardware. Hadoop is an Apache top-level project being built and used by a global community of contributors and users. Hadoop was created by Doug Cutting and Mike Cafarella in 2005. It was originally developed to support distribution for the Nutch search engine project.

When data sets go beyond a single storage capacity, it is necessary to distribute them to multiple independent computers. Trans-computer network storage file management system is called distributed file system. A typical Hadoop distributed file system[9] contains thousands of servers, each server stores partial data of file system. HDFS cluster configuration is simple. It just needs more servers and some simple configuration to improve the Hadoop cluster computing power, storage capacity and IO bandwidth. In addition HDFS achieves reliable data replication, fast fault detection and automatic recovery, etc.

Apache Hadoop is an open source Java framework for processing and querying vast amounts of data on large clusters of commodity hardware. Hadoop is a top level Apache project, initiated and led by Yahoo!. It relies on an active community of contributors from all over the world for its success.

With a significant technology investment by Yahoo!, Apache Hadoop has become an enterprise-ready cloud computing technology. It is becoming the industry de facto framework for big data processing. Hadoop is designed to efficiently process large volumes of information by connecting many commodity computers together to work in

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 6, Issue 5, May 2018

parallel. The theoretical 1000-CPU machine described earlier would cost a very large amount of money, far more than 1,000 single-CPU or 250 quad-core machines. Hadoop will tie these smaller and more reasonably priced machines together into a single cost-effective compute cluster.

III. PROPOSED ALGORITHM

K-Means: It is a type of unsupervised algorithm which solves the clustering problem. Its procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters). Data points inside a cluster are homogeneous and heterogeneous to peer groups the proposed scheme is segregated in three parts respectively.

a) How to Map Reduce K-means

• Partition {x1 ,…,xn } into K clusters
K is predefined
• Initialization
– Specify the initial cluster centers (centroids)
• Iteration until no change
– For each object x1
• Calculate the distances between xi and the K centroids
• Re)assign xi to the cluster whose centroid is the closest to x1
– Update the cluster centroids based on current assignment.

b) K-MEANS Map/ Reduce Function

• KMeans() formation
• Assigncluster()
– For each point p
• Assign p the closest c
• Updatecluster ()
– For each cluster
• Update cluster center

c) MapReduce K-means Algorithm

– Runs multiple iteration jobs using mapper+combiner+reducer
• Mapper
– Configure: A single file containing cluster centers
– Input: Input data points
– Output: (data id, cluster id)
• Reducer
– Input: (data id, cluster id)
– Output: (cluster id, cluster centroid)
• Combiner
– Input: (data id, cluster id)
– Output: (cluster id, (partial sum, number of points))

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 6, Issue 5, May 2018

IV. SIMULATION AND RESULTS

The first step in designing the MapReduce routines for K-means is to define and handle the input and output of the implementation. The input is given as a <key,value> pair , where 'key' is the cluster center and 'value' is the serializable implementation of vector in the data set. The prerequisite to implement the Map and Reduce routines is to have two file one that houses the clusters with their centroids and the other that houses the vectors to be clustered. Once the set of initial set of clusters and chosen centroids is defined and the data vectors that are to be clustered properly organized in two files then the clustering of data using K-Means clustering technique can be accomplished by proposed scheme and algorithm to design the Map and Reduce routines for K-Means Clustering.

The initial set of centers is stored in the input directory of HDFS prior to Map routine call and they form the 'key' field in the pair. The instructions required to compute the distance between the given data set and cluster center fed as a pair is coded in the Mapper routine. The Mapper is structured in such a way that it computes the distance between the vector value and each of the cluster centers mentioned in the cluster set and simultaneously keeping track of the cluster to which the given vector is closest. Once the computation of distances is complete the vector should be assigned to the nearest cluster. Once Mapper is invoked the given vector is assigned to the cluster that it is closest related to. After the assignment is done the centroid of that particular cluster is recalculated. The recalculation is done by the Reduce routine and also it restructures the cluster to prevent creations of clusters with extreme sizes i.e. cluster having too less data vectors or a cluster having too many data vectors. Finally, once the centroid of the given cluster is updated, the new set of vectors and clusters is re-written to the disk and is ready for the next iteration the below output screens depicts the proposed scenarios and results.

```
[root@sandbox KMeans]# java -jar ProcessCorpus.jar
Enter the directory where the corpus is located: 20_newsgroups
Enter the name of the file to write the result to: vectors
Enter the max number of docs to use in each subdirectory: 100
20_newsgroups
Counting the number of occurs of each word in the corpus..1.000Found 46397 uniqu
e words in the corpus.
How many of those words do you want to use to process the doc 10000
Bad input!
How many of those words do you want to use to process the docs? 1000
Done creating the dictionary.
Converting the corpus to a list of vectors...Done vectorizing all of the docs!
[root@sandbox KMeans]#
```

Figure 1: Process the Corpus and Forming Vectors thus pertaining 100 Sub Directories to Evaluate the Corpus

```
[root@sandbox KMeans]# java -jar GetCentroids.jar
Enter the data file to select the clusters from: vectors
Enter the name of the file to write the result to: clusters
Enter the number of clusters to select: 20
..Done selecting centroids.
[root@sandbox KMeans]# java -jar KMeans.jar
Enter the file with the data vectors: vectors
Enter the name of the file where the clusers are loated: clusters
Enter the number of iterations to run: 10
..Done with pass thru data.
```

Figure 2: Forming the Centriods over vectors that are to be clustered using K-Means

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 6, Issue 5, May 2018

```
s.misc: 3; comp.windows.x: 2; talk.politics.guns: 2; comp.sys.mac.hardware: 2; t
alk.politics.mideast: 1; comp.graphics: 1; sci.med: 1; misc.forsale: 1;

******** cluster16 ******** comp.graphics: 2; sci.space: 2; comp.windows.x: 1; tal
k.religion.misc: 1; alt.atheism: 1; rec.sport.hockey: 1; comp.os.ms-windows.misc
: 1; comp.sys.ibm.pc.hardware: 1;

******** cluster17 ******** comp.windows.x: 7; rec.motorcycles: 5; comp.graphics:
4; sci.electronics: 3; talk.politics.misc: 1; alt.atheism: 1; rec.sport.hockey:
1; comp.sys.ibm.pc.hardware: 1;

******** cluster18 ******** misc.forsale: 19; comp.windows.x: 11; rec.autos: 10; r
ec.sport.baseball: 9; sci.electronics: 9; rec.sport.hockey: 8; talk.politics.mid
east: 6; comp.sys.mac.hardware: 6; comp.sys.ibm.pc.hardware: 6; rec.motorcycles:
 5; talk.politics.guns: 5; sci.crypt: 5; sci.med: 5; soc.religion.christian: 5;
comp.graphics: 4; sci.space: 3; talk.religion.misc: 2; talk.politics.misc: 2; al
t.atheism: 2; comp.os.ms-windows.misc: 1;

******** cluster19 ******** rec.sport.hockey: 18; comp.sys.ibm.pc.hardware: 15; ta
lk.politics.misc: 14; talk.politics.guns: 13; sci.crypt: 13; rec.autos: 12; sci.
electronics: 12; comp.sys.mac.hardware: 11; sci.space: 11; comp.windows.x: 10; t
alk.politics.mideast: 10; rec.motorcycles: 9; rec.sport.baseball: 8; comp.graphi
cs: 6; talk.religion.misc: 5; alt.atheism: 4; sci.med: 3; comp.os.ms-windows.mis
c: 2;
```

Figure 3: Clutered Results Omitting the Redudant Records and Files Thus Production and Appropriate Data/Information

Below is table opted for test under the scheme

S. No	Number of Doc in Each Sub-directory	Total Number of Documents	Unique Words Identified	Number of Cluster
1	250	5000	69,996	20
2	500	10,000	1,11,203	20
3	750	15,000	1,35,162	20
4	1000	20,000	1,53,832	20

V. CONCLUSION AND FUTURE WORK

This scheme presents enormous information Hadoop with MapReduce and give a brief on Clustering Techniques used to examine huge information. In this work near examination of Distributed K-Means method is done on remain solitary framework and different hub framework. So far numerical information been utilized for grouping yet we have finished with the absolute information. Some pre-handling methods have been connected on the absolute information which produces yield in numerical configuration. The Distributed K-Means approach is produced in java, sent in MapReduce system of Hadoop. The Experimental outcome have been assembled which demonstrates that the Distributed K-Means works all the more effectively on the Multiple Node then the Single Node when tried on the three datasets with Different measurements. However, for the future scope the same can be incorporated with data streaming service like apache spark and mahout on multimode clusters and balancers.

REFERENCES

1 D. Arthur and S. Vassilvitskii. k-means++ the advantages of careful seeding. In Symposium on Discrete Algorithms, 2007.
2 M. Craven, D. DiPasquo, A. McCallum, T. Mitchell, K. Nigam, and S. Slattery. Learning to extract symbolic knowledge from the world wide web. In AAAI-98, 1998.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 6, Issue 5, May 2018

3 D. R. Cutting, J. O. Pedersen, D. R. Karger, and J. W. Tukey. Scatter/gather: A cluster-based approach to browsing large document collections. In Proceedings of the ACM SIGIR, 1992.
4 M. Ester, H.-P.Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In Proceedings of 2nd International Conference on KDD, 1996.
5 N. Friburger and D. Maurel. Textual similarity based on proper names. In Proceedings of Workshop on Mathematical Formal Methods in Information Retrieval at th 25th ACM SIGIR Conference, 2002.
6 E. Han, D. Boley, M. Gini, R. Gross, K. Hastings, G. Karypis, V. Kumar, B. Mobasher, and J. Moore. Webace: A web agent for document categorization andexploartion. In Proceedings of the 2nd International Conference on Autonomous Agents., 1998.
7 A. Hotho, S. Staab, and G. Stumme.Wordnet improves text document clustering. In Proceedings of the SIGIR Semantic Web Workshop, Toronto,, 2003.
8 A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. ACM Computing Surveys (CSUR), 31(3):264–323, 1999.
9 B. Larsen and C. Aone. Fast and effective text mining using linear-time document clustering. In Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 1999.
10 D. D. Lewis. Reuters-21578 text categorization test collection distribution 1.0. http://www.research.att.com/ lewis, 1999.
11 J. Lin. Divergence measures based on the shannon entropy. IEEE Transaction on Information Theory, 37(1):145–151, 1991.
12 D. Milne, O. Medelyan, and I. H. Witten. Mining domain-specific thesauri from wikipedia: A case study. In Proc. of the International Conference on Web Intelligence (IEEE/WIC/ACM WI'2006), 2006.
13 J. M. Neuhaus and J. D. Kalbfleisch. Between- and within-cluster covariate effects in the analysis of clustered data. Biometrics, 54(2):638–645, Jun. 1998.
14 M. F. Porter. An algorithm for suffix stripping. Program, 14(3):130–137, 1980.
15 G. Salton. Automatic Text Processing. Addison-Wesley, New York, 1989.
16 M. Steinbach, G. Karypis, and V. Kumar.A comparison of document clustering techniques.In KDD Workshop on Text Mining, 2000.
17 A. Strehl, J. Ghosh, and R. Mooney.Impact of similarity measures on web-page clustering. In AAAI-2000: Workshop on Artificial Intelligence for Web Search, July 2000.
18 N. Z. Tishby, F. Pereira, and W. Bialek.The information bottleneck method.In Proceedings of the 37th Allerton Conference on Communication, Control and Computing, 1999.
19 E. Voorhees and D. Harman.Overview of the fifth text retrieval conference (trec-5).In Proc. of the Fifth Text REtrieval Conference (TREC-5), 1998.
20 P. Willett. Recent trends in hierarchic document clustering: a critical review. Information Processing and Management: an International Journal, 24(5):577–597, 1988.
21 R. B. Yates and B. R. Neto.Modern Information Retrieval. ADDISON-WESLEY, New York, 1999. 22 Y. Zhao and G. Karypis.Evaluation of hierarchical clustering algorithms for document datasets.