International Journal of Recent Engineering Research and Development (IJRERD)
ISSN: 2455-8761
www.ijrerd.com || Volume 02 – Issue 11 || November 2017 || PP. 110-114

Index Algorithm for Clustering of Images Using Bigdata Platform Spark

M. Praveen Kumar[1], M. Pavithra[2]
[1]*Assistant Professor, Department of Information Technology, Rathinam Technical Campus, Coimbatore.*
[2]*Assistant Professor, Department of Information Technology, Rathinam Technical Campus, Coimbatore.*

Abstract: Clustering (or cluster analysis) aims to organize a collection of data items into clusters, such that items within a cluster are more "similar" to each other than they are to items in the other clusters. This notion of similarity can be expressed in very different ways, according to the purpose of the study, to domain-specific assumptions and to prior knowledge of the problem. Clustering is usually performed when no information is available concerning the membership of data items to predefined classes. For this reason, clustering is traditionally seen as part of unsupervised learning. We nevertheless speak here of unsupervised clustering to distinguish it from a more recent and less common approach that makes use of a small amount of supervision to "guide" or "adjust" clustering. To support the extensive use of clustering in computer vision, pattern recognition, information retrieval, data mining, etc., very many different methods were developed in several communities. In this paper, we are proposing the solution over big data platform Apache Spark which performs the clustering of images using different methods viz. Scalable K-means++, Bisecting Kmeans and Gaussian Mixture. Since the number of clusters is not known in advance in any of the methods, we also propose a Best of Breed approach of validating the number of clusters using Simple Silhouette Index algorithm and thus to provide the best clustering possible.

Keywords: Satellite images, Clustering, Scalable Kmeans++,Distributed Processing, Spark, Bisecting Kmeans, Gaussian Mixture

I. Introduction

In recent years there has been a growing interest in developing effective methods for content-based image retrieval (CBIR). Image clustering and categorization is a means for high-level description of image content. The goal is to find a mapping of the archive images into classes (clusters) such that the set of classes provide essentially the same information about the image archive as the entire image-set collection. The generated classes provide a concise summarization and visualization of the image content that can be used for different tasks related to image database management. Image clustering enables the implementation of efficient retrieval algorithms and the creation of a user-friendly interface to the database. A common approach to image clustering involves addressing the following issues:

1. Image features – how to represent the image.
2. Organization of feature data – how to organize the data.
3. Classifier – how to classify an image to a certain cluster.

In our work we propose a new method for unsupervised image clustering. We use probabilistic continuous Gaussian Mixture models (GMM), for the image representation, and information theoretic principles for image classification and database organization. Industry experts are facing challenges with respect to the processing of the large amount of data generated in satellite imagery. Hadoop is the distributed platform to store and process big data. It distributes the data and process it on distinct nodes in the cluster there by minimizing network transfers. The Hadoop Distributed File System (HDFS) helps in distributing the part of the file effectively [2]. Apache Spark is another distributed processing framework which can read from HDFS. Researchers have performed various studies to run Kmeans on Hadoop [3] [4]. Studies have been conducted to run the algorithm effectively on Hadoop to improve its performance and scalability [5] [6]. The paper explores the algorithms to run multiple parallel Scalable K-means++ clustering of satellite images for different values of k [7].As the number of clusters is usually not known in advance, experiments are conducted by selecting the initial value of k and then incrementing it for a certain number of times. Then the appropriate number of clusters is calculated by validating algorithms such as Elbow method and Silhouette Index [8] [9]. In this paper, we are proposing a solution which uses Apache Spark as the distributed computing framework and finds the best clustering among the different clustering algorithms. We have used three clustering algorithm sviz. Scalable K-Means++, Bisecting K-Means and Gaussian Mixture Model. Spark Mllib provides support for various clustering algorithms out of which K Means is one of them [10][11]. The Mllib library provides an implementation ofK means || [5]. The bisecting K-means is a divisive hierarchical clustering algorithm. It is also a variation of K-

means. Similar to K-means, the number of clusters must be predefined. The Gaussian mixture clustering algorithm is based on the so-called Gaussian Mixture Model for assembling the clusters. This algorithm is used to improve the performance of image segmentation [12]. Similar to K-means and bisecting K-means, the Gaussian mixture clustering algorithm implementation by Spark requires a predefined number of clusters. All the three mentioned algorithms have been implemented in the Spark MLLib library. In the next section, the paper provides the background & related work. In section 3, we discuss the Hadoop ecosystem with the basic information of Apache Spark. In section 4 we discuss the methodology and proposed solution. Section 5 provides all the details of the experiments and the last section of the paper concludes our findings.

II. Literature Survey

K-Means has been one of the most effective clustering algorithms. It has proven its effectiveness in almost all scientific applications including remote sensing. The groups are created taking similarity of data points into account. Along with simplicity and effectiveness, the algorithm has certain limitations and drawbacks. The first limitation is the choice of initial centroids. If they are not chosen appropriately, then there is a possibility that K-Means converge to just a local optimum. Also, the number of clusters needs to be known in advance.

Gaussian Mixture Model also has the same problem as far as image segmentation is concerned [12]. The survey paper [13] discusses all the different approaches that have been used for determination of number of clusters.

AlDaoud and Roberts proposed the initialization method based on the density of uniformly partitioned data [14]. The algorithm partitioned the data into N cell sand the centroids are then chosen from each cell on the basis of the density proportion of that particular cell.

Kauffman also came up with the greedy approach to initialize the K point [15]. In 2007, Arthur and Vassilvitskii proposed the K-means++ clustering algorithm [16]. The paper proposed the careful seeding methodology to select the initial points closer to the optimum result. The initial point is chosen at random. Next subsequent centers are calculated proportional to the closest squared distance from the already chosen center.

In [3], researchers proposed the K-Means clustering algorithm which ran in parallel based on Map Reduce. Zhenhua et al. in [1] ran the Map Reduce K-means clustering on satellite images. Taking the help of a Map Reduce execution framework, the algorithm scaled pretty well on commodity hardware. Later, [5] came up with scalable K-means++ to optimize it further by sampling more points in each iteration instead of single point. Even though the number of iterations decreases still many iterations are needed. In [4], the paper proposed an efficient k-means approximation with Map Reduce. It proposed that a single Map Reduce is sufficient for the initialization phase instead of multiple iterations. The Gaussian Mixture Model has proven its utility in image segmentation of MRI images [17]. Some studies have also evaluated K-means and its variants like Bisecting K-means, Fuzzy C-Means etc. [18].

III. Hadoop Ecosystem

Hadoop is a framework which deals with storage of large amounts of data and processing them in a distributed manner. A large file is split into multiple parts on different nodes of the cluster. The same task gets executed on each part of the file simultaneously. Hadoop is based on two core concepts. Hadoop Distributed File System (HDFS) and Distributed Processing Framework Map Reduce. After the creation of Hadoop by Doug Cutting, it has evolved a lot and has built a well-integrated ecosystem around its core concepts of Map Reduce and HDFS.

A. Hadoop Distributed File System (HDFS)

HDFS works with the help of two types of nodes. Name node and Data node. The Name node is the node in the cluster that stores the details of all the files on the cluster. The Hadoop client talks with this node to read and write from/to data nodes. The name node stores the locations of all the splits of the large file. The data node actually stores the part of the file. The block size is the maximum size of the file portion which can be stored on one node. In earlier versions of Hadoop, the default value of block size used to be 64 MB. This has been increased to 128MB now.

B. MapReduce

On the other hand, a Map Reduce layer consists of Job tracker and Task trackers. The job tracker is initialized every time a job starts executing. It is the responsibility of the job tracker to initialize the multiple task trackers on each and every node where the split of the file exists. The task trackers then execute the desired task on the respective nodes. In this way, Hadoop achieves parallel execution on commodity hardware. For reading binary data there are other file formats specific to Hadoop such as Sequence File Format which stores

data in binary key-value pair. Since our dataset deals with small and medium sized images, Sequence Files are appropriate for our experiment [2].

C. Apache Spark

Similar to Hadoop Map Reduce, there is another clustering computing framework, called Spark [11]. Spark can be deployed on HDFS as well as standalone. Spark has some in-memory processing capabilities, although it doesn't store all the data in the memory. In our experiments using Spark, we have deployed it over HDFS. The core concept of Apache Spark which has helped Spark in achieving high performance is Resilient Distributed Datasets (RDD). RDDs are distributed, lazyand can be persisted. Since the data is distributed on HDFS over multiple nodes the processing can happen on individual nodes minimizing the transfer of data over the network. This is similar to the Map Reduce approach. Another property is laziness. Spark reads all the operations to be executed for the particular output and doesn't execute them until the output is explicitly asked. The operations are not executed individually, which means the data need not be stored on disk in order to pass it to the next operation. RDD can be persisted in the memory as well on the disk.

IV. Methodology

In this paper, we are proposing a solution over Hadoop to find the best possible clustering of a satellite image using Spark library. We have implemented the solution which reads a sequence file containing multiple images and then apply different clustering algorithms for different values of *k*, where *k* is the number of desired clusters. The proposed framework has three phases viz. Reading, Clustering and Best of Breed. Figure 1 shows the schematic diagram of the methodology followed in this paper.

Spark comes with different libraries meant for special purposes. These include Spark SQL for structured data processing, Graph X for graph processing and Spark Streaming for streaming. Similarly, Mllib provides out-of-the-box capability for common learning algorithms such as classification, regression, clustering, and collaborative filtering. As the prerequisite for our Spark based experiment, satellite images were converted into a sequence file. In a sequence file, these satellite images were stored as values in key-value pairs with the file name as the key. Sequence Files were then read as byte arrays and converted into a Buffered Image. Following are the three phases of the experiment.

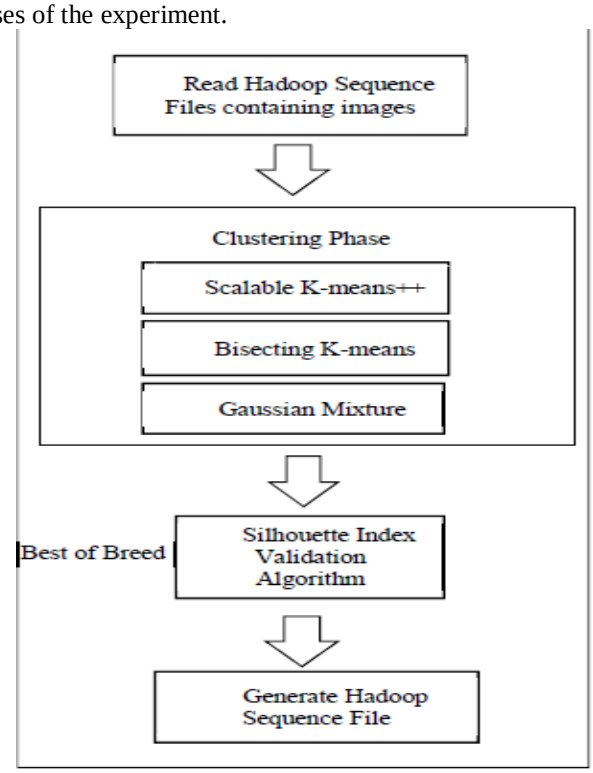

Figure 1. Methodology

A. Reading Sequence Files

In the first phase, the sequence file is read from HDFS using Spark context and created an RDD.

B. Clustering Phase

For each file that has been read from the sequence file, we performed different clustering algorithms available in the Spark MLLib library. The library includes parallelized version of K-means++ [10]. GMM and Bisecting K-means are also part of the library. Here, we have initialized the value of k to 2 and then incremented it till k is equal to 7. For each value of k all three clustering algorithms are performed and the centroids are calculated.

C. Best of Breed Approach - Validating clusters

As an output of the second phase we have different clusters for three clustering algorithms. We need to find the best clustering for the image which means the best possible number of clusters as well as the algorithm that has produced the clustered image. In this third phase, we have used Silhouette Index algorithm in order to validate the consistency of data clusters. This algorithm suggests the cohesion among the objects in the cluster. The purpose of this algorithm is to tell the cohesion among the objects in the cluster. For better clustering, cohesion needs to be more. For validating the clusters, [6] chose Simplified Silhouette Index (SSI) which is one of the variants of the original Silhouette Index. SSI is the simplified approach of the Silhouette Index as the distance is calculated between the centroid and the data point instead of calculating all the distances between all the data points. Elbow method sometimes doesn't work when the dataset is not clustered properly.

V. Conclusion & Future Scope

In our experiment we have created a solution which suggests the best clustering algorithm that could be applied to a particular satellite image and also the number of appropriate clusters required for image processing. We ran three clustering algorithms on the images and then finds out the best among them using the Simplified Silhouette Index. We also proved that the proposed Best of Breed solution is scalable for images with less than 200 MB in size. In our experiment, we assumed that the image will be processed on single node only. We are now working on the solution such that larger images which span over two nodes or more could be processed on multiple nodes. We further plan to add more clustering algorithms which are not present in Apache Spark. We are also going to evaluate these algorithms on the basis of their scalability and performance.

References

[1]. Tapan Sharma, Dr. Vinod Shokeen, Dr. Sunil Mathur, "*Best of Breed Solution for Clustering of Satellite Images Using Bigdata Platform Spark" in 2017,* KInternational Conference on Inventive Communication and Computational Technologies.

[2]. M. Praveen Kumar, S. P. Santhoshkumar, T.Gowdhaman, S. Syed Shajahaan, "A SURVEY ON IoT PERFORMANCES IN BIG DATA "International Journal of Computer Science and Mobile Computing, IJCSMC, Vol. 6, Issue. 10, pg.26 – 34, October 2017.

[3]. Z. Lv, Y. Hu, Z. Haidong, J. Wu, B. Li and H. Zhao, "Parallel K-Means Clustering of Remote Sensing Images Based on MapReduce," in *2010 International Conference on Web Information Systems and Mining, WISM 2010*, 2010.

[4]. T. White, "Hadoop I/O : File Based Data Structures," in *Hadoop - The Definitive Guide*, O'Reilly.

[5]. W. Zhao, H. Ma and Q. He, "Parallel K-Means Clustering Based on MapReduce," in *CloudCom*, Beijing, 2009.

[6]. Y. Xu, W. Qu, G. Min, K. Li and Z. Liu, "Efficient k-Means++ Approximation with MapReduce," *IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS*, vol. 25, no. 12, December, 2014.

[7]. B. Bahmani, B. Moseley, A. Vattani, R. Kumar and S. Vassilvitskii, "Scalable k-means++," in *International Conference on Very Large Databases*, 2012.

[8]. K. D. Garcia and M. C. Naldi, "Multiple Parallel MapReduce k-means Clustering with Validation and Selection," in *Brazilian Conference on Intelligent Systems, IEEE*, 2014.

[9]. T. Shama, V. Shokeen and S. Mathur, "Multiple K Means++ Clustering of Satellite Image Using Hadoop MapReduce and Spark," *International Journal Of Advanced Studies In Computer Science And Engineering*, vol. 5, no. 4, 2016.

[10]. Wikipedia, "Elbow Method (CLustering)," [Online]. Available: https://en.wikipedia.org/wiki/Elbow_method_(clustering).

[11]. Wiki, "Silhouette Index(Clustering)," [Online]. Available: https://en.wikipedia.org/wiki/Silhouette_(clustering).

[12]. A. Spark, "Clustering - RDD-based API," Apache Spark, [Online]. Available: http://spark.apache.org/docs/latest/mllib-clustering.html.

[13]. "Apache Spark," [Online]. Available: http://spark.apache.org/docs/1.3.0/mllib-clustering.html#kmean

[14]. K. A. Tran, N. V. Quang and G. Lee, "A novel clustering algorithm based Gaussian mixture model for image segmentation," in *Proceedings of the 8th International Conference on Ubiquitous Information Management and Communication*, Siem Reap, Cambodia, 2014.

[15]. E. Hance and D. Karaboga, "A comprehensive survey of traditional, merge-split and evolutionary approaches proposed for determination of cluster number," *Swarm and Evolutionary Computation*, 2016.

[16]. M. AlDaoud and S. A. Roberts, "New methods for the initialization of clusters," *Pattern Recognition Letters*, pp.451-455, 1996.

[17]. L. Kaufman and P. J. Rousseeuw, Finding Groups in Data: An Introduction to Cluster Analysis, Wiley, 1990.

[18]. D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," in *SODA, Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, 2007.

[19]. M. A. Balafar, "Gaussian mixture model based segmentation methods for brain MRI images," *Artificial Intelligence Review*, vol. 41, no. 3, pp. 429-439, 2014.

[20]. S. Banerjee, . A. Choudhary and S. Pal, "Empirical evaluation of K-Means, Bisecting K-Means, Fuzzy CMeans and Genetic K-Means clustering algorithms," in *IEEE International WIE Conference on Electrical and Computer Engineering (WIECON-ECE)*, 2015