Clustering of Big Data Using Kekre's Fast Codebook Generation (KFCG)Algorithm

Dr. Vinayak Bharadi
HOD-IT Department,
FAMT,
University of Mumbai.

Romita Thally
Student, IT Department,
FAMT,
University of Mumbai.

Shruti Mane
Student, IT Department,
FAMT,
University of Mumbai.

ABSTRACT

In the information era, gigantic amount of data have become available on hand to decision makers. Big data refers to datasets that are not only big, but also excessive in variety and velocity, which makes them laborious to handle using conventional tools and techniques. Due to the expeditious growth of such data, solutions need to be studied and provided in order to handle and extract value and knowledge from these datasets. Moreover, decision makers need to be able to gain valuable judgements from such varied and rapidly changing data, ranging from daily transactions to customer interactions and social network data. Such value can be made accessible using big data analytics, which is the implementation of new analytic techniques on big data. This paper aims to give idea about KFCG algorithm for big data. KFCG was initially prepared for normal image, audio, video data but was never optimized for big data. We prepare modification in KFCG so that it becomes suitable for big data analytics.

Keywords

Big Data, Clustering, KFCG, K-means, BFR

INTRODUCTION

In the recent years, as the population continues to increase, the amount of data that is being created and stored is unbelievable and it just keeps growing. This introduces us with the concept of Big Data[1][2][13][14]. The term Big Data refers to all the data that is being produced across the world at an extraordinary rate. The data could be either structured or unstructured. Today's business enterprises owe a huge part of their success to an economy that is firmly knowledge-oriented. Information drives the modern organizations of the world and hence making sense of this data and unravelling the various designs and revealing unseen connections with the vast sea of information is critical and a hugely rewarding endeavor indeed. In Big Data, volume, variety and velocity are three main drivers that give a new dimension to the way of analytics[2][14]. A huge volume of information is generated through social media and the velocity in which the information gets uploaded is also high. This information comes from a wide variety of sources which can be in the form of pictures, videos and unstructured texts via social media.

Due to the rapid growth of such data, solutions need to be studied and provided in order to handle and extract value and knowledge from these datasets. Moreover, decision makers need to be able to gain valuable insights from such diverse and rapidly changing data, ranging from daily transactions to customer interactions and social network data. Such value can be provided using Big Data Analytics, which is the application of advanced analytics techniques on Big Data.

The common types of events involved in the processing of Big Data are[3]:

- Feeding data into the system.
- Persevering data in storage.
- Computing and analyzing the data.
- Visualizing the results.

Also, the increased size of datasets has boosted demand for efficient clustering techniques that satisfy memory use, document processing and execution time requirements. An issue related to Big Data concerns the grouping of objects such that data of the same group are more similar than those of the other groups or clusters. So traditionally, an extension of k-means algorithm[1][2] i.e. BFR algorithm is being used.

In the proposed work, we will be studying whether the Kekre's fast codebook generation (KFCG) algorithm can be used for this purpose. BFR algorithm is well known for it's capacity of handling large data whereas KFCG is well known for it's speed of codebook generation and minimum error. So, we will be trying to modify the existing KFCG algorithm using BFR algorithm so that we can use it for Big Data. In this paper we will give a brief idea about this algorithm.

2. EXISTING APPROACH FOR BIG DATA

2.1. BFR ALGORITHM

Clustering[4] is one of the important method by which large data sets are categorized into groups. The BFR algorithm[5] is a point assignment clustering algorithm which is an extension of K-means algorithm that is designed to deal with large data sets which cannot fit into main memory. Prior to BFR, clustering in large database was really difficult. This algorithm makes a very strong assumption about shape of clusters that they must be normally distributed about a centroid. Axes of cluster must align with the axes of space.

The BFR algorithm begins by selecting k points. Then the points of data files are read in chunks. These chunks might be from distributed file system or the conventional file might be partitioned into chunks of appropriate size. Each chunk must consist of few enough points that can be processed in main memory. To begin with the initial load we select the initial k centroids by some sensible approach[2][5][7]:

- Take k random points
- Take a small random sample and cluster optimally

- Take a sample, pick an arbitrary point and then k-1 more points, each as far from the previously selected points as possible.

The main memory data other than the chunk from the input consists of 3 types of objects which we keep track of:

- The Discard Set (DS): Points which are close enough to a centroid to be summarized.
- The Compressed Set (CS): Groups of points that are close enough to each other but not to any existing centroid. These points are to be summarized but not assigned to any cluster.
- The Retained Set (RS): Isolated points waiting to be assigned to a compression set.

For each cluster the discard set (DS) is summarized by:

- The number of points 'N'
- The vector 'SUM': ith component=sum of the coordinates of the points in the ith dimension.
- The vector 'SUMSQ': ith component=sum of squares of coordinates in the ith dimension.

Summarizing Points:

- 2d+1 values represent any size cluster where d= number of dimensions.
- Average in each dimension(the centroid) can be calculated as SUMi/N where SUMi= ith component of SUM.
- Variance of a cluster's discard set in dimension i is: (SUMSQi/N)-(SUMi/N)2 and standard deviation is the square root of that.

Processing The Memory Load of points:

- Find those points that are "sufficiently close" to a cluster centroid and add those points to that cluster and the DS. If these points are so close to the centroid then they can be summarized and then discarded.
- Use main memory clustering algorithms to cluster the remaining points and the old RS. Clusters go to the CS and outlying points to the RS.
- Discard Set: Adjust statistics for the clusters to account for the new points. Add Ns, SUMs, SUM SQs
- Consider merging compressed sets in the CS.
- If this is the last round, merge all compressed sets in the CS and all RS points into their nearest cluster.

3. KFCG ALGORITHM FOR CODEBOOK GENERATION

The Kekre's Fast Codebook Generation Algorithm (KFCG) is a newly suggested clustering algorithm[6][7][15[16][17]. It gives less MSE. The KFCG algorithm first calculates the average of the given cluster along the first dimension and then splits the cluster by keeping all the vectors which are less than or equal in one cluster and the remaining in another cluster. It then continues to divide the resulting cluster by computing their average with respect to the next dimension. The process is repeated till the desired number of code vectors are obtained. The algorithm is summarized as follows.

The algorithm reduces the codebook generation time since it avoids the Euclidean distance computations. Initially there is one cluster with the entire training vectors and the code vector C1 which is centroid. In the first iteration of the algorithm, the clusters are formed by comparing first element of training vector with first element of code vector C1. The vector Xi is grouped into the cluster 1 if xi1< c11 otherwise vector Xi is grouped into cluster 2. In second iteration, the cluster 1 is split into two by comparing second element xi2 of vector Xi belonging to cluster 1 with that of the element c12 of the code vector C1. Cluster 2 is divided in two by differentiating the element xi2 of vector Xi which is a member of cluster 2 with that of the element c22 of the code vector C2.

This procedure is repeated till the codebook size is reached to the size specified by user. It is detected that this algorithm gives minimum error and requires least time to generate codebook as compared to LBG and KPE[8].

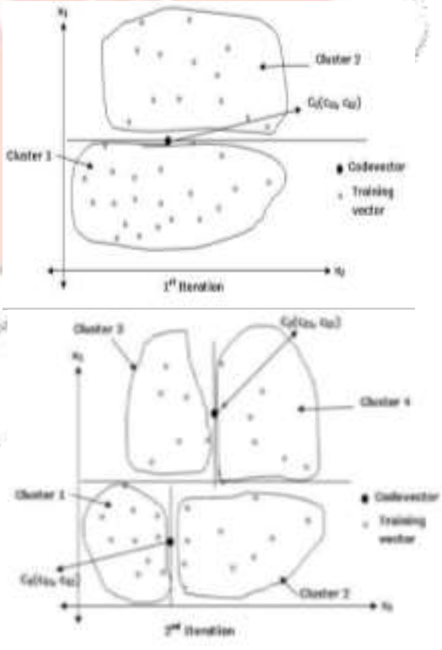

Fig1. Cluster Formation in KFCG

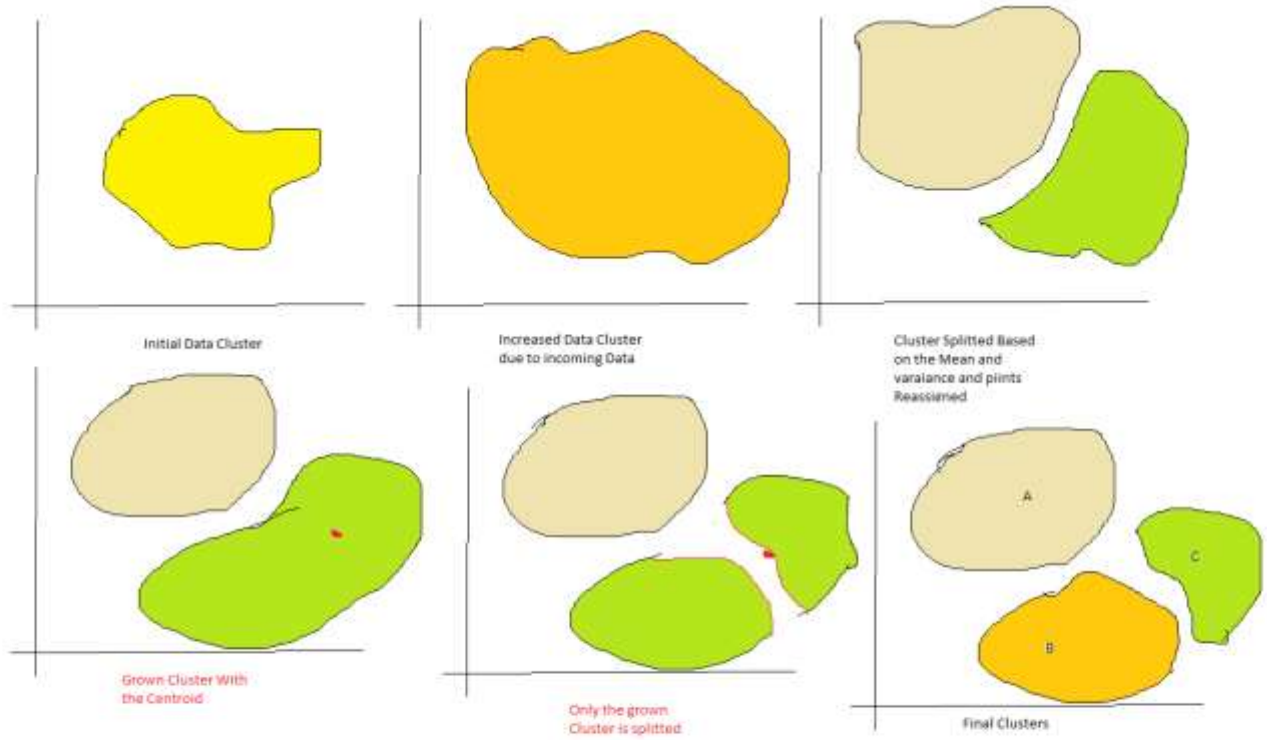
Initial Data Cluster
Increased Data Cluster due to incoming Data
Cluster Splitted Based on the Mean and varaiance and piints Reassigned
Grown Cluster With the Centroid
Only the grown Cluster is splitted
A
C
B
Final Clusters

Figure 2 Clustering of Streaming Data by Modified KFCG

4. PROPOSED WORK

4.1. Modified KFCG for Big Data

BFR algorithm is proposed for Big Data. But, KFCG algorithm is not proposed for Big Data yet. KFCG has several advantages over BFR such as:

- It needs no initiation.
- It needs no prior knowledge of datasets.
- It is faster as compared to other algorithms.
- It introduces minimum or no error.
- Less complexity.

So, for all these reasons KFCG is to be modified for Big Data approach.

4.1.1. For Streaming Data

1. Suppose we have large amount of streaming data.
2. Start with any 1 cluster initially.
3. Count the number of points ‘N’.
4. Average in each dimension(centroid) can be calculated as SUMi/N.
5. Variance of the cluster in dimension i is : (SUMSQi/N)-(SUMi/N)2 and standard deviation is square root of it.
6. The cluster has a threshold limit in each dimension which indicates the maximum allowed difference between values in that dimension.
7. When standard deviation goes beyond this threshold we will split the cluster into two on the basis of dimensions and find the average(centroid) for each cluster.
8. Similarly for these two clusters, we will repeat the procedure and split these two into four and so on.
9. There is a significant advantage of representing N,SUMi,SUMSQi because if we add new point to the cluster N is increased by 1, we can also add the vector representing location of the point to SUM to get the new SUM and we can add the squares of the i^{th} components to get the new SUMSQi.
   The Process is depicted in Fig. 2

4.1.2. For Stored data

1. Suppose we have 10 GB of data stored on the disk which cannot be fit in the main memory.
2. In this situation, we will perform data sampling which is a statistical analysis technique used to select, manipulate and analyze a representative subset of data points in order to identify patterns and trends in the larger data set being examined. The main three types of sampling techniques are[12]:
   a. **Simple Random Sampling**: A simple random sample is a sample selected in such a way that

every possible sample of the same size is equally likely to be chosen.

b. **Stratified Random Sampling**: It is obtained by separating the data into mutually exclusive sets, or strata, and then drawing simple random samples from each stratum.

c. **Cluster Sampling**: It is a simple random sample of groups or clusters of elements.

3. Then, as mentioned for streaming data we will use KFCG for clustering this sample too.

4. We will get various data segments after clustering.

5. Then we need not to perform clustering for the remaining data. We will just add the remaining points to the segmented data on the basis of similarity.

6. FUTURE WORK

Future work will be to implement this algorithm in C#.Net and test it on KDD Dataset made available by UCI [17].

7. CONCLUSION

In this paper a new algorithm of clustering of streaming as well as stored Big Data is Proposed. The proposed algorithm is modified from KFCG, so that it is possible to cluster streaming data as well as data having size bigger than available physical memory.

8. REFERENCE

1. Radha Shankarmani, M Vijayalakshmi,“Big Data Analytics”, Wiley Precise Textbook Series, ISBN: 978-81-265-5865-0.
2. Anand Rajaraman, Jeffrey David Ullman, “Mining of massive Datasets”, Cambridge University Press, ISBN: 978-1-107-44824-7.
3. http://mobile.developer.com/db/understanding-big-data-processing-and-analytics.html
4. Mythili S1, Madhiya E2, “An Analysis on Clustering Algorithms in Data Mining”, IJCSMC, Vol. 3, Issue. 1, January 2014.
5. Anand Rajaraman, Juri Leskovec, “https://web.stanford.edu/class/cs345a/slides/12-clustering.pdf”
6. Tanuja Sarode, Nabanita Mandal, “Performance Comparison of K-means Codebook Optimization using different Clustering Techniques” IOSR Journal of Computer Engineering (IOSR-JCE), e-ISSN: 2278-0661, p- ISSN: 2278-8727 Volume 14, Issue 3 (Sep. - Oct. 2013), PP 89-98
7. Y Pawan Kumar Reddy, G Kesavan, “Tumor identification using self organizing MAP and BFR algorithm”, Middle-East Journal of Scientific Research 24 (6): 2110-2115, 2016 ISSN 1990-9233 © IDOSI Publications, 2016 DOI: 10.5829/idosi.mejsr.2016.24.06.23604
8. Nutan Palshikar, P S Lokhande,“Analysis of Codebook Generation Techniques for Vector Quantization” ,International Conference & Workshop on Recent Trends in Technology, (TCET) 2012. Proceedings published in International Journal of Computer Applications® (IJCA)
9. H B Kekre, Vaishali Kulkarni,“Comparative Analysis of Automatic Speaker Recognition using Kekre’s Fast Codebook Generation Algorithm in Time and Transform Domain” ,International Journal of Computer Applications (0975 – 8887),Volume 7– No.1, September 2010
10. H B Kekre, Tanuja Sarode, Bhakti Raul,“Color Image Segmentation Using Kekre’s Algorithm for Vector Quantization”,World Academy of Science, Engineering and Technology, Vol:2, No:9, 2008
11. H. B. Kekre,Tanuja K. Sarode,Sudeep D. Thepade, Pallavi N. Halarnkar“Kekre’s fast codebook generation in VQ with various color spaces for colorization of grayscale images”,https://link.springer.com/chapter/10.1007/978-81-8489-989-4_18
12. Data collection and sampling,http://www.updallas.edu/~scniu/OPRE-6301/documents/Data_Collection_and _Sampling.pdf
13. G. M. D'silva, S. Thakare and V. A. Bharadi, "Real-time processing of IoT events using a Software as a Service (SaaS) architecture with graph database," 2016 International Conference on Computing Communication Control and automation (ICCUBEA), Pune, 2016, pp. 1-6.
doi: 10.1109/ICCUBEA.2016.7859984
14. M. Meena and V. A. Bharadi, "Performance analysis of cloud based software as a service (SaaS) model on public and hybrid cloud," 2016 Symposium on Colossal Data Analysis and Networking (CDAN), Indore, 2016, pp. 1-6.
doi: 10.1109/CDAN.2016.7570951
15. V. A. Bharadi, P. Mishra and B. Pandya, "Multimodal face recognition using multidimensional clustering on hyperspectral face images," 2014 5th International Conference - Confluence The Next Generation Information Technology Summit (Confluence), Noida, 2014, pp. 582-588.
doi: 10.1109/CONFLUENCE.2014.6949048
16. V. A. Bharadi, P. Mishra and B. Pandya, "Multimodal face recognition using multidimensional clustering on hyperspectral face images," 2014 5th International Conference - Confluence The Next Generation Information Technology Summit (Confluence), Noida, 2014, pp. 582-588.doi: 10.1109/CONFLUENCE.2014.6949048
17. KDD Datasets by UCI : http://kdd.ics.uci.edu/summary.data.application.html