IX International Conference on Multidisciplinary Research
(IEI, Chandigarh) Institution of Engineers, India , Chandigarh
21st December 2019 www.conferenceworld.in
Conference World
ISBN : 978-81-943584-6-6

EFFECTIVE MANAGEMENT OF BIG DATA: USING DIFFERENT PARTITIONING TECHNIQUES AND ALGORITHMS

Dr. Kiranjit Kaur
Assistant Professor, Guru Nanak College, Moga

ABSTRACT

As we all ready know that technology is generating peta bytes of data on daily basis. It becoming bottleneck to storing and retrieve this data. The focus of this paper is to discuss detail information about different types of data partitioning which improves storage and access to this big data. Different algorithm of big data partitioning are also discussed which not only store but also helps to effectively managing this big data.

I. INTRODUCTION

Partitioning can be said as identification of similar classes of objects. Partitioning is a data mining technique of grouping set of data objects into multiple groups or clusters so that objects within the cluster have high similarity, but are very dissimilar to objects in the other clusters.

Partitioning is basically the process of grouping the entities in different classes called as clusters comprising similar objects [1]. A cluster is group of entities that are similar and belongs to the same class. In the process of partitioning, firstly the set of data is partitioned in groups on the basis of data similarity and afterward labels are assigned to those groups.

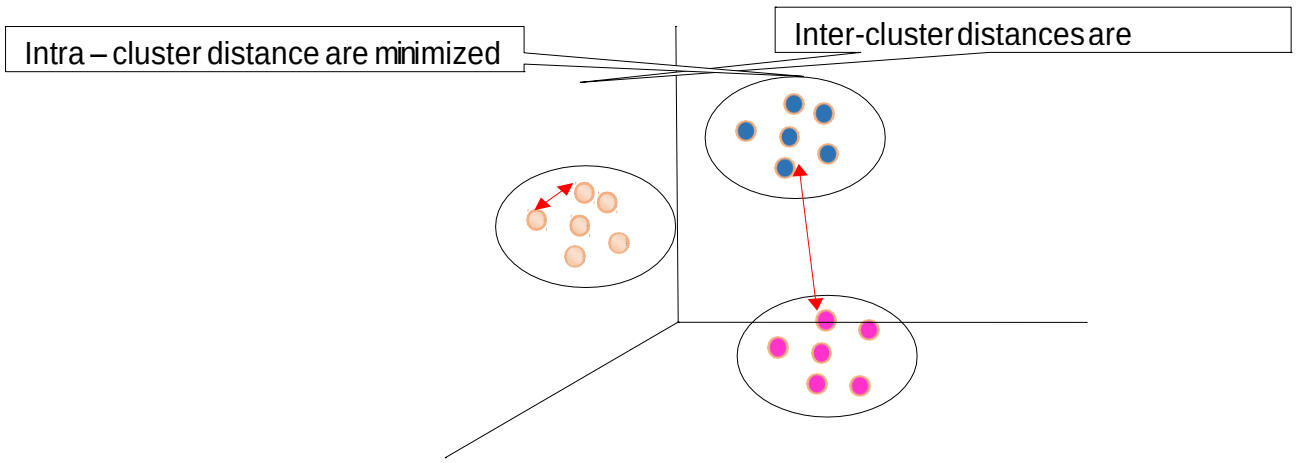

Fig 1: Partitioning Principle

ISBN : 978-81-943584-6-6

Partitioning can be considered the most important unsupervised learning technique so as every other problem of this kind. It deals with finding a structure in a collection of unlabelled data. In data mining, partitioning functions as a means to achieve awareness into the scattering of data to witness the features of each cluster. Cluster analysis has been widely used in many applications such as business intelligence image pattern recognition web search biology and Security. For example, in business intelligence partitioning can be used to organize a large number of customers into groups where customers within a group share similar characteristics. This facilitates the development of business strategies for enhanced customer relationship management. Partitioning plays an incomparable share in quite a few applications of the data mining i.e. information retrieval and scientific data exploration, spatial database applications, text mining [2]. The articles of investigation in the partitioning procedure could be people, wages, views, software units and many others. The features of such articles required to be presented carefully as these features are the primary variables of the problem and their selection considerably effects the products of partitioning algorithm.

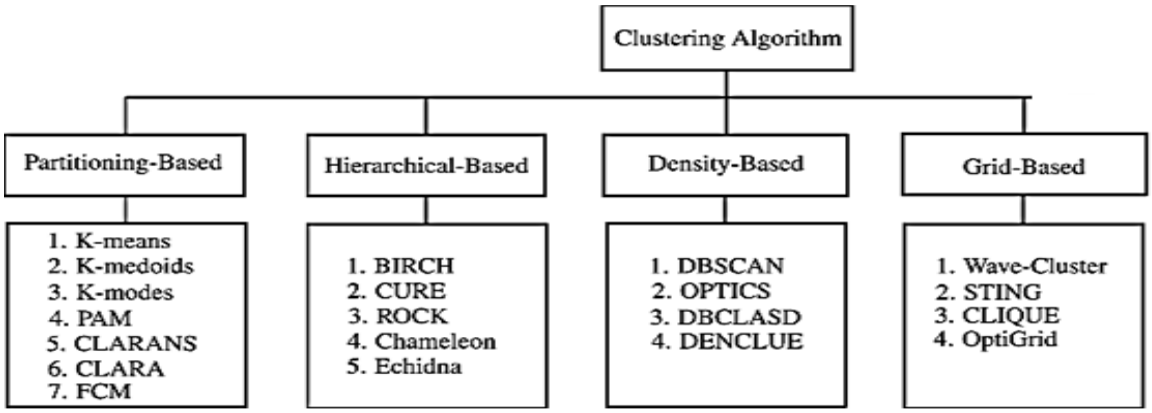

Fig 2: Classification of the Partitioning Algorithms

Database partitioning has been considered to be an activity that is conducted by passing through the following seven steps:

1. Define Object-View
2. Select Relevant Attributes
3. Generate Suitable Input Format for the Partitioning Tool
4. Define Similarity Measure
5. Select Parameter Settings for the Chosen Partitioning Algorithm
6. Run Partitioning Algorithm

7. Characterize the Computed Clusters

The first three steps of the suggested database partitioning methodology center on pre- processing the database and on generating a data set that can be processed by the employed partitioning algorithm(s). In these steps, a decision has to be made what objects in the database (usually databases contain multiple types of objects) and which of their properties will be used for the purpose of partitioning; moreover, the relevant information has to be converted to a format that can be processed by the selected partitioning tool(s).
In the fourth step similarity measures for the objects to be clustered have to be defined. Finally, in steps 5-7 the partitioning algorithm has to be run, and summaries of the obtained clusters are generated.

II. PARTITIONING IN BIG DATA

It is the main focus issue of data mining particularly for analysis of big data. In this partitioning large data volumes are grouped which are related to each other or we can say grouping of similar elements closely related to each other. The main issues that arise in data partitioning is the measure of similarity, how many clusters should be generated, validity of cluster etc. in partitioning n objects are collected and organized in a hierarchy. The partitions are based on how much number of distinct groups can be created of related elements [3].

Though issues also involve in partitioning as no advance knowledge of the quantity and type of clusters in data. Clusters also take time for generation. The advantages of partitioning is efficient browsing, data is organized properly and recommendation pop-up when some material is searched. As in facebook's grouping is based on title of status messages. During search for videos on YouTube by writing the related word and the output is group of related videos. Different partitioning algorithms are available like k-mean, Gaussian mixture models, spectral partitioning, nearest neighbor and kernel k-means etc. It is a type of exploration techniques used for collecting data in scientific field. The parameters and the algorithm to be chosen for partitioning is based on the data and is necessary for big data problem. Kernel k-means give good results for making of clusters as trade of is provided between scalability and partitioning accuracy. The challenged needed to be tackled are scalability, streaming data, enormous clusters [3].

III. TYPES OF PARTITIONING

In the cluster analysis, certain approaches developed for the purpose of partitioning the huge unstructured

dataset into meaningful subsets which can help users in better understanding. The process of partitioning is mainly classified into the following types [4]:

1) Partitional Partitioning

This method basically divides the data objects into number of partitions i.e. clusters. The data objects are divided into non-overlapping clusters, so that the data objects in the same clusters are closer to the center mean values. It rearranges the instances by transferring them from one to another cluster, beginning from an initial partitioning. It ensures that the convergence is confined and the optimum solution globally cannot be assured.

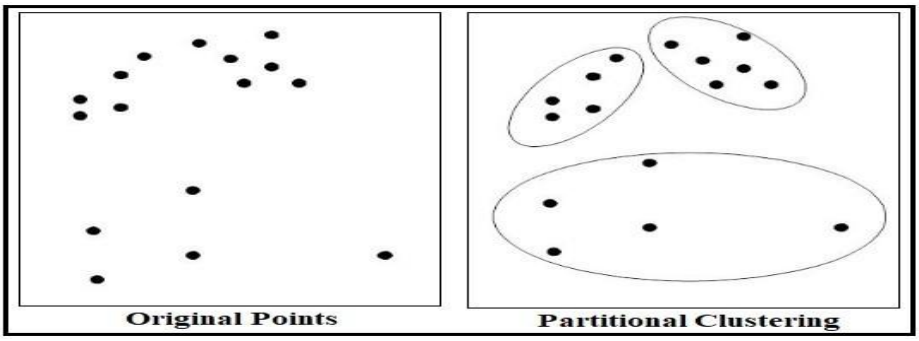

Fig 3 Example of Partitioned Partitioning

2) Hierarchical Partitioning

This method primarily constructs cluster in the hierarchical orders. It forms nested clusters that are organized in the hierarchical tree either in the top-to-bottom manner or bottom-to-up manner.

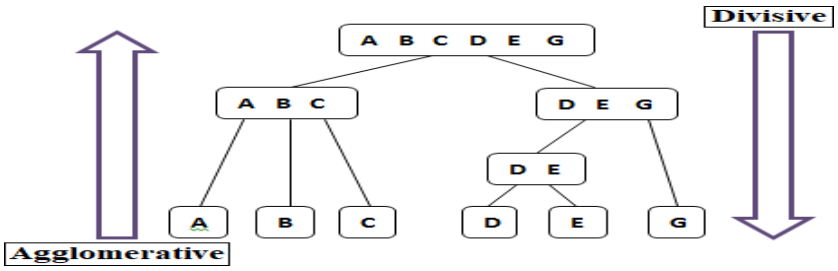

Fig 4: Agglomerative and Divisive approaches

The hierarchical partitioning is further divided into two types as follows:

- Agglomerative – This is a bottom-to-up approach where initially every data objects considered as a separate cluster, and after wards two or more suitable clusters are merged to form new clusters. The merging is performed recursively on the clusters until a stopping criteria is encountered.
- Divisive- This is a top-to-bottom approach, in which the entire datasets is firstly considered as one cluster and afterwards divided into sub-clusters and these sub- clusters are further divided into more sub-clusters. The division process continued until the stopping criteria came across.

3) Density-Based Partitioning

This method is primarily based on the concepts of density, connectivity and border. The clusters are formed on the basis of density in a region of data points and remain developing given cluster providing the density in the neighbourhood is beyond some threshold. The clusters in this method are formed with random shapes as they develop in any direction based on density. It also removes the outliers or noisy data points naturally.

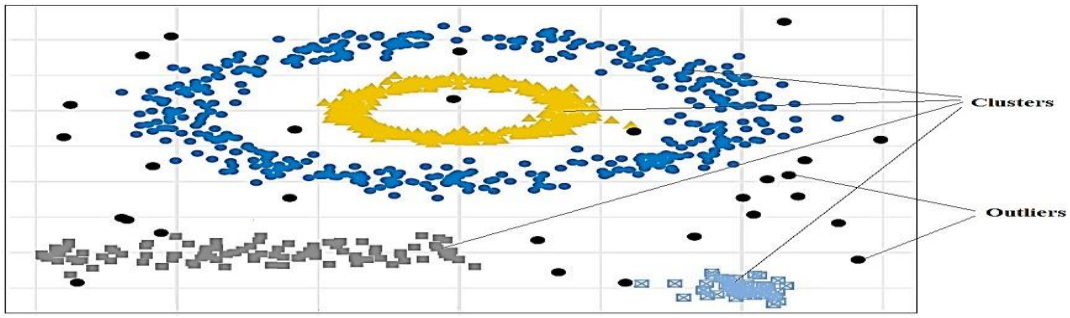

Fig 5: Example of Density Based Partitioning

4) Model Based Partitioning

This method generally optimizes the fit between the given data and predefined mathematical model. In this method, a model is anticipated for every cluster in order to determine the best fitting data for a particular cluster.

ISBN : 978-81-943584-6-6

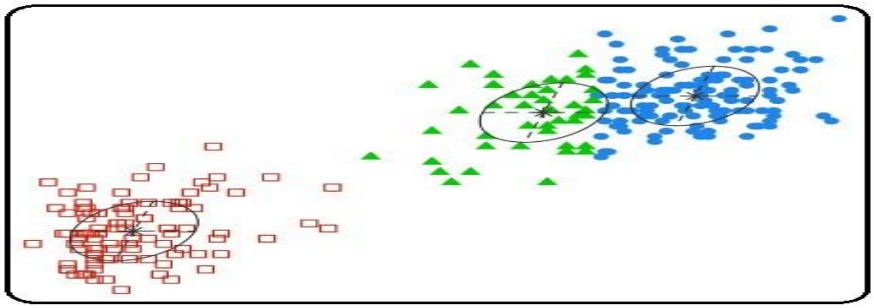

Fig 6: Example of Model Based Partitioning

5) Grid-Based Partitioning

This method primarily uses a multi-resolution grid data arrangement. This method is employed to form clusters in a large multidimensional space, in which the denser region is regarded as clusters. The space is partitioned into a finite number of cells which together forms a grid like structure. This method helps in expressing the data at different level of detail on the basis of all the features that are chosen as dimensional features.

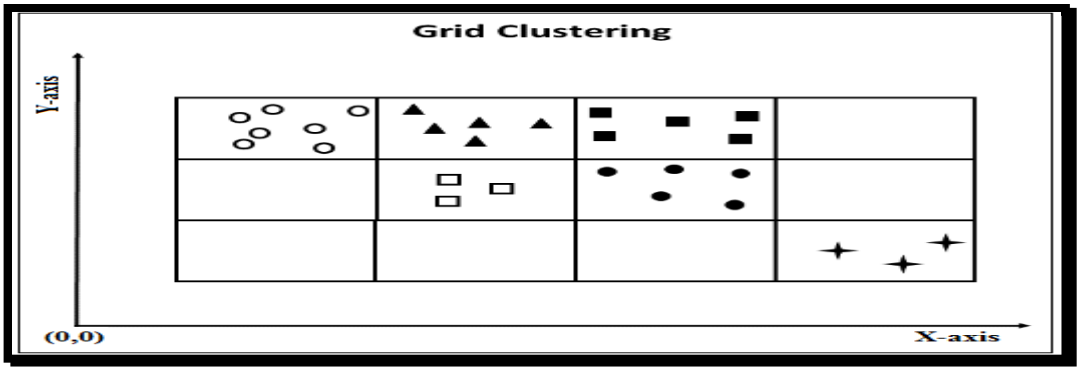

Fig 7: Represents Grid Based Partitioning

IV. PARTITIONING ALGORITHMS

The algorithms that are employed to perform partitioning on any type of dataset is primarily acknowledged as the partitioning algorithms [5]. The following are some of the most commonly used algorithms globally.

1) MapReduce Algorithm

Today the main problem of the organization data is the analysis of the big data which is very challenging. This problem is solved by map-reduce algorithm. It is a program design model which allows ease in development of scalable parallel applications to process big data. In this framework, initially a distributed file system divides the data in several machines and representation of data is in pair of key-value [6]. The map and reduce functions are used for computation. The key-value pair is taken as input and output is also produced in the key- value pair. Different partitions of input data is taken in parallel by the user defined map reduce function. The output key-value pair is merged by each different key. At last reduce function is invoked by the key which output the related values sharing the key.

i. Algorithm Preparation of Map ( ) input: System splits the input into *n* pieces and gets the map workers *n* on the cluster of machines.

ii. Running of user defined Map code: Each key-value pair is passed to a user-defined Map function and the key-value produced by the key-value is stored in the memory.

iii. Map output is put to reduce processors: From the map workers local disk the buffered data is read by the reduce worker. After reading the sorting of data is done by the key relationship as the data of same key are merged together.

iv. Run the Reduce Code: Over the sorted data reduce worker iterates and finds the unique intermediate key, if encountered it passes the key and its corresponding values to reduce function.

v. Final output is produced.

2) K-means Algorithm

The K-means algorithm, probably the first one of the partitioning algorithms proposed, is based on a very simple idea: Given a set of initial clusters, assign each point to one of them, and then each cluster center is replaced by the mean point on the respective cluster. These two simple steps are repeated until convergence. A point is assigned to the cluster which is close in Euclidean distance to the point.

K-means partitioning aims to partition n objects in to k clusters in which each observation belongs to the cluster with the nearest mean [7]. Every cluster is having a cluster head or centroid. The centroid of a cluster is equal to the mean of all points in that cluster. The number of clusters is randomly chosen by user. K-mean algorithm proceeds by iteratively allocating points to the cluster with the closest centroid. The “Closeness” is measured using Euclidean distance.

ISBN : 978-81-943584-6-6

Euclidean distance is an ideal metric for geometrical queries. Euclidean distance is basically described as the normal distance separating the two points and can be simply measured with a ruler either in two or three-dimensional space. Euclidean distance is widely used in partitioning problems, including partitioning text. It satisfies all the four conditions to be a metric and therefore is a true metric. It is also the default distance measure applied with the K-means algorithm [8].

Although, K-means has the great advantage of being easy to implement, it has two big drawbacks. First, it can be really slow since in each step the distance between each point to each cluster has to be calculated, which can be really expensive in the presence of a large dataset. Second, this method is really sensitive to the provided initial clusters, however, in recent years, this problem has been addressed with some degree of success.

(1) Choose k cluster centers to coincide with k randomly-chosen patterns or k randomly defined points inside the hyper volume containing the pattern set.

(2) Assign each pattern to the closest cluster center.

(3) Re-compute the cluster centers using the current cluster memberships.

(4) If a convergence criterion is not met, go to step 2. Typical convergence criteria are: no (or minimal) reassignment of patterns to new cluster centers, or minimal decrease in squared error.

The K-means partitioning algorithm uses iterative refinement to produce a final result. The algorithm inputs are the number of clusters K and the data set. The data set is a collection of features for each data point. The algorithm starts with initial estimates for the K centroids, which can either be randomly generated or randomly selected from the data set. The algorithm then iterates between the following two steps:

i. Data assignment step

Each centroid defines one of the clusters. In this step, each data point is assigned to its nearest centroid, based on the squared Euclidean distance.

ii. Centroid update step

This is the step where the centroids are recomputed, which is completed by taking the mean of total number of data points that are assigned to that particular centroid's cluster. The algorithm iterates in the middle of the both steps i.e. step one and step two, till an ending condition come across (i.e., none of the data points convert clusters, the sum of the distances is minimized, or nearly maximum number of iterations is arrived).

ISBN : 978-81-943584-6-6

3) Linear Discriminant Analysis (LDA)

Linear Discriminant Analysis (LDA) is a very significant tool in a wide-ranging diversity of problems. LDA technique was originally established in 1936 by R. A. Fisher. The term Discriminant Analysis is usually well-defined as a statistical approach which is employed to classify the entities into mutually exclusive and exhaustive groups on the basis of a set of measurable features of the entities [9]. LDA is established upon the conception of searching for a linear combination of variables (predictors) that proficiently separates two classes (targets). It is mainly a technique that can be used in statistics, pattern recognition and machine learning in order to find a linear combination of features that characterizes or separates two or more classes of objects or events.

LDA easily handles the case where the within-class frequencies are unequal and their performances have been examined on randomly generated test data. It maximizes the ratio of between-class variance to the within-class variance in any particular data set thereby guaranteeing maximal separability. It is mostly used in the machine learning problems like pattern recognition, face recognition, feature extraction and data dimensionality reduction. Data sets can be transformed and test vectors can be classified in the transformed space by two different approaches that are explained below:

1) Class-dependent transformation

This approach comprises expanding the ratio of between class differences to within class difference. The key objective is to expand this ratio so that suitable class separability is achieved. The class-specific approach includes utilization of two optimizing measures for transforming the data sets individually.

2) Class-independent transformation

This approach consists of expanding the ratio of total difference to within class difference. This approach utilizes only single optimizing measure to transform the data sets and therefore all data points regardless of their class individuality are transformed via this transformation approach. In this type of LDA methodology, each class is measured as a separate class against all other classes.

The selection of the type of LDA primarily relies on the data set and also on the objectives of the classification problem. The class independent transformation is preferably selected if the broad view is of importance. On the other hand, the class dependent type should be preferred if the aim is in a good discrimination.

REFERENCES

[1] Ionia Veritawati, Ito Wasito, Mujiono, "*Sparse Data for Document Clustering*", IEEE International Conference Information and Communication Technology (ICoICT), Pages 38-43, 2013.

[2] E. Alan Calvillo, Alejandro Padilla, Jaime Munoz, Julio Ponce, Jesualdo T. Fernandez, "*Searching Research Papers Using Clustering and Text Mining*", 23rd International Conference on Electronics Communication and Computing, Pages 78-81, 2013.

[3] Ranjana Agrawal, Madhura Phatak, "*A Novel Algorithm for Automatic Document Clustering*", 3rd IEEE International Advance Computing Conference (IACC), Pages 877-882, 2013.

[4] Hsi-Cheng Chang, Chiun-Chieh Hsu, "*Using Topic Keyword Clusters for Automatic Document Clustering*", IEEE Third International Conference on Information Technology and Applications (ICITA'05), Pages 419-424, Vol.1, 2005.

[5] Beil F., Ester M., Xu X., "*Frequent Term-based Text Clustering*. ", ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Pages 436-442, 2002.

[6] Fung B.C.M., Wang K. and Ester M, "*Hierarchical Document Clustering using Frequent Item sets*", Proceedings of SIAM International Conference on Data Mining, Pages 180-304,2003.

[7] Ponmuthuramalingam. P. Devi. D, "*Effective Dimension Reduction Techniques for Text Documents*", International Journal on Computer Science and Network Security (ICSNS), Vol. 10, No. 7, 2010.

[8] Yeming Hu, Evangelos E. Milios, James Blustein, "*Enhancing Semi-Supervised Document Clustering with Feature Supervision*", Pages 929-936 ACM, 2012.

[9] Yanjun Li, Soon M. Chung, John D. Holt, "*Text Document clustering based on frequent word meaning sequences* ", In Data and Knowledge Engineering, Pages 381-404, 2008.