Volume 8, No. 3, March – April 2017
ISSN No. 0976-5697
International Journal of Advanced Research in Computer Science
RESEARCH PAPER
Available Online at www.ijarcs.info

Enhancement in K-Mean Clustering in Big Data

Anju
M.Tech Student
Department of Computer Science and Applications
Maharishi Dayanand University, Rohtak, India

Preeti Gulia
Assistant Professor
Department of Computer Science and Applications
Maharishi Dayanand University, Rohtak, India

***Abstract*:** Big [1] data is a data set that is big in size. It is much complicated so traditional data processing application software is not capable to handle them. There are several challenges such as capture of data, storage of data, searching of data, & transfer of data. Some challenges are related to visualization & querying of data. Scientist has faced several challenges in e-Science such as meteorology, complicated physics simulation & environmental researches. Lot of challenges has been faced due to big data in case of biology & genomics. The problems with existing system[6] were search, sharing, storage, transfer, visualization, querying-updating. These problems can be reduced by using proposed algorithm. In this paper we have explain clustering and proposed algorithm is discussed.

***Keywords*:** Clustering, K-Mean, Data mining, Big data

1 INTRODUCTION

Data Mining [2] is defined as extracting information from huge sets of data. They can say that data mining is the process of mining knowledge from data. Analysis of data sets [1] can find new correlations to spot business trends, prevent diseases, combat crime & so on. Scientists, business executives, advertising & governments alike regularly meet difficulties with large data sets in areas including Internet search, finance & business informatics. Scientists encounter limitations in e-Science work, including meteorology, genomics, connectomics, complex physics simulations, biology & environmental research. Group of data are growing fatly in part because they are increasingly gathered by cheap & numerous information-sensing mobile devices, aerial (remote sensing), software logs, cameras, microphones, radio-frequency identification (RFID) readers & wireless sensor networks.

Clustering

By examining one or more attributes or classes, you may group individual pieces of data value together to form a structure opinion. At a simple stage, clustering is using one or more attributes as your basis for identifying a cluster of correlating results. Clustering is valuable to identify dissimilar info since this correlates with other examples so you may see where similarities & ranges agree. Clustering[7] may work both ways. You may assume that there is a cluster at a certain point & then use our credentials criteria to see if you are correct.

2 REQUIREMENTS OF CLUSTERING IN DATA MINING[1]

The following points throw light on why clustering is required in data mining

Scalability:- We need highly scalable clustering algorithms to deal with large databases.

Ability to deal with different kinds of attributes:- Algorithms should be capable to be applied on any kind of data such as interval-based (numerical) data, categorical, and binary data.

Discovery of clusters with attribute shape:- The clustering algorithm should be capable of detecting clusters of arbitrary shape. They should not be bounded to only distance measures that tend to find spherical cluster of small sizes.

High dimensionality:- The clustering algorithm[7] should not only be able to handle low-dimensional data but also the high dimensional space.

Ability to deal with noisy data:-Databases contain noisy, missing or erroneous data. Some algorithms are responsive to such data & poor quality clusters.

Interpretability:-The clustering results should be interpretable, comprehensible, and usable.

3 PROPOSED WORK

Before specifying proposed work, K-Mean algorithm[8] is discussed. The proposed work is based on it algorithm.

K-means clustering is know as partitioning method. In it objects are classified as belonging to one of K-groups. In each cluster there might be a centroid or a cluster presentative. In case where we think real valued data,

mathematics mean of attribute vectors for all objects[8] within a cluster given an appropriate representative; alternative types of centroid might be required within other cases.

Suppose we had following data set

2	5	6	8	12	15	18	28	30

Suppose K=3

C1=2

C2=12

C3=30

2	5	6	8	12	15	18	28	30
C1				C2				C3

The distance is calculated for each data point from centroid & data point having minimum distance from centriod of a cluster is assigned particular cluster.

So cluster according to distance are as follow

12-5>5-2

So cluster for data point 5 is C1

6-2>12-6

So cluster for data point 6 is C1

In same way cluster would be assigned

2	5	6	8	12	15	18	28	30
C1	C1	C1	C2	C2	C2	C2	C3	C3

Data member of C1 are 2,5,6

Data Member for C2 are 8,12,15,18

Data Member for C3 are 28,30

Clusters generated previously, centriod is again repeatly calculated means recalculation of centriod.

So mean of cluster C1 is (2+5+6)/3=4.3

So mean of cluster C2 is (8+12+15+18)/4=13.25

So mean of cluster C3 is (28+30)/2=29

Now distance would be recalculated within new mean & cluster[5] of data point would be changed according to new distance

2	5	6	**8**	12	15	18	28	30
C1	C1	C1	**C2**	C2	C2	C2	C3	C3

For example take 8 from C2 cluster

But Problems within existing system were analysis, capture, search, sharing, storage, transfer, visualization, querying updating. One more problems within K-means clustering [14] is that empty clusters are generated during execution, if within no data points are allocated of cluster under consideration during assignment phase. The proposed algorithm[5] overcome these problem. K-mean clustering proposed algorithm as follow.

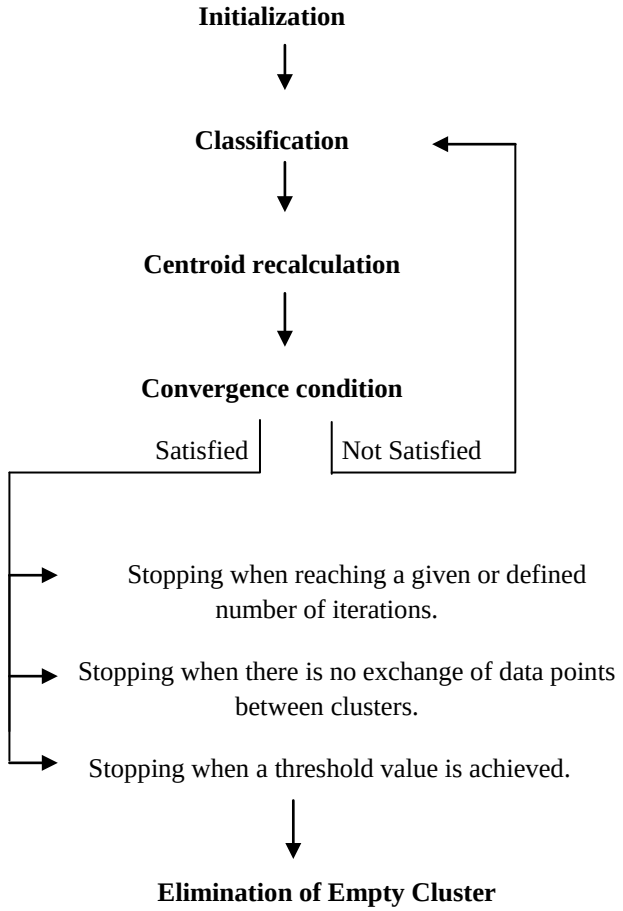

Fig 1 Proposed algorithm

Proposed algorithm

MSE=largenumber;
Select initial cluster centroids {*mj*}*j*
k=1;
Do
OldMSE=MSE;
*MSE*1=0;
For *j*=1 to *k*
mj=0; *nj*=0;
endfor
For *i*=1 to *n*
For *j*=1 to *k*
Compute squared Euclidean

distance *d*2(*xi*, *mj*);
endfor
Find closest centroid *mj* to *xi*;
mj=*mj*+*xi*; *nj*=*nj*+1;
*MSE*1=*MSE*1+*d*2(*xi*, *mj*);
endfor
For *j*=1 to *k*
nj=max(*nj*, 1); *mj*=*mj*/*nj*;
endfor
MSE=*MSE*1;
while (*MSE*<*OldMSE*)
For *j*=1 to *k*
If(sizeof(mj==0))
{
remove mj
}
Endfor

4 IMPLEMENTATION OF CLUSTER REMOVAL

getFileSize() function would check size of file that is representing a cluster & it function would be called in main to remove cluster if it is empty.

```
public static long getFileSize(String filename) {
    File file = new File(filename);
    if (!file.exists() || !file.isFile()) {
      System.out.println("File doesn\'t exist");
      return -1;
    }
    return file.length();
  }
  public static void main(String[] args) {
    long size = getFileSize("cluster1.txt");
    System.out.println("Filesize in bytes: " + size);
if(size==0)
{
try
{
      boolean success= (new File("cluster1.txt")).delete();
       if (success) {
         System.out.println("The file has been been successfully deleted");
       }
}
catch(Exception e)
{
System.out.println(e);
}}  }}
```

Comparative analysis of result between Existing & Proposed K-MEAN

Table 1 Comparative analysis of result between Existing & Proposed K-MEAN

Number of record	Existing (K-Mean)	Proposed Algorithm
1000	2	1
1500	2	2
2000	3	2
2500	3	3
3000	4	3
4000	6	4
5000	8	5
6000	9	6
7000	9	7
8000	12	8
9000	13	9
10000	14	10

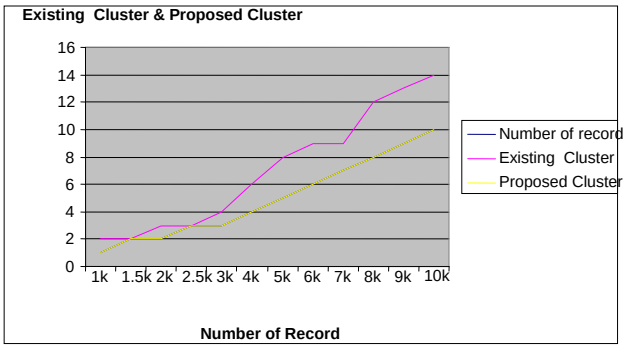

Fig 2 Analysis of Existing & Proposed cluster

Output in case of old K-Means:

Table 2: Number of record in cluster and total size of cluster in both cases

Number of record	No. of Cluster in case of old k-means	Total Size in old k-means	No. of clusters in case of new k-means	Total size in new k-means
1000	2	1220	1	1123
2000	3	1843	2	1750
3000	4	2490	3	2276
4000	6	4945	4	4760
5000	8	6734	5	6593
6000	9	7554	6	7345
7000	9	8454	7	8322
8000	12	12344	8	12222
9000	13	13454	9	12954
10000	14	15667	10	14322

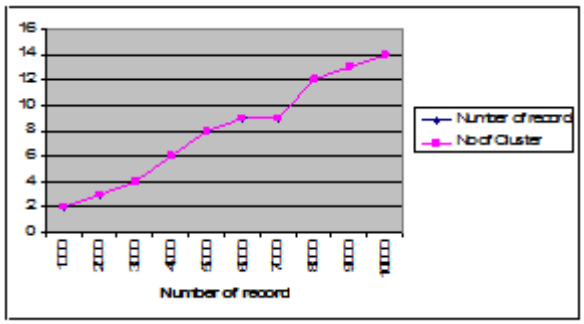

Fig 3 number of record & cluster in old k-means

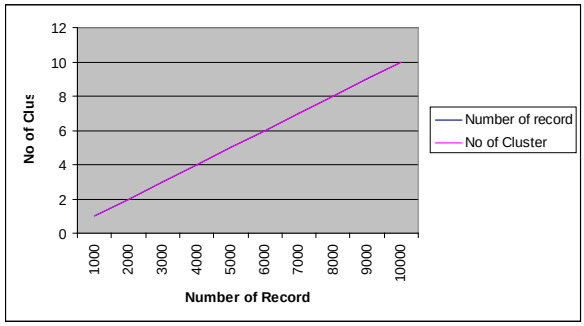

Fig 4 number of record & cluster in new k-means

Comparative analysis of result between old & enhanced K-MEAN

Table 3: Comparative analysis of result between old & enhanced K-MEAN

Number of record	Old K-Mean Algorithm	Enhanced Algorithm
1000	2	1
1500	2	2
2000	3	2
2500	3	3
3000	4	3
4000	6	4
5000	8	5
6000	9	6
7000	9	7
8000	12	8
9000	13	9
10000	14	10

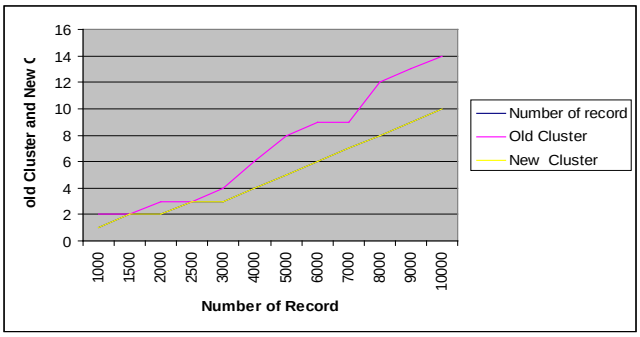

Fig 5 Analysis of old & new cluster

Above figure represent comparative analysis of number of clusters formed in case of old K mean clustering & enhaced K mean clustering . Number of vacant clusters has been removed in case of enhanced clustering algorithm so number of clusters get reduced in case of enhanced algorithm.

5 CONCLUSION

Clustering is process[6] of grouping objects that belongs to same class. Similar objects are grouped in one cluster & dissimilar objects are grouped in another cluster. We have explain comparative analysis of number of clusters formed in case of existing K mean clustering & proposed K mean clustering[14] . The number of vacant clusters had been removed in case of proposed clustering algorithm so number of clusters get reduced in case of proposed algorithm.

6 REFERENCES

1. Piatetsky-Shapiro, Gregory (1991), Discovery, analysis, & presentation of strong rules, in Piatetsky-Shapiro, Gregory; & Frawley, William J.; eds., Knowledge Discovery in Databases, AAAI/MIT Press, Cambridge, MA.
2. Agrawal, R.; Imieliński, T.; Swami, A. (1993). "Mining association rules between sets of items in large databases". Proceedings of 1993 ACM SIGMOD international conference on Management of data - SIGMOD '93. p. 207. doi:10.1145/170035.170072. ISBN 0897915925.
3. Hahsler, Michael (2005). "Introduction to arules – A computational environment for mining association rules & frequent item sets" (PDF). Journal of Statistical Software.
4. Michael Hahsler (2015). A Probabilistic Comparison of Commonly Used Interest Measures for Association Rules. http://michael.hahsler.net/research/association_rules/measures.html
5. Hipp, J.; Güntzer, U.; Nakhaeizadeh, G. (2000). "Algorithms for association rule mining --- a general survey & comparison". ACM SIGKDD Explorations Newsletter **2**: 58. doi:10.1145/360402.360421.
6. Tan, Pang-Ning; Michael, Steinbach; Kumar, Vipin (2005). "Chapter 6. Association Analysis: Basic Concepts & Algorithms" (PDF). Introduction to Data Mining. Addison-Wesley. ISBN 0-321-32136-7.
7. Pei, Jian; Han, Jiawei; & Lakshmanan, Laks V. S.; Mining frequent itemsets within convertible constraints, in Proceedings of 17th International Conference on Data Engineering, April 2–6, 2001, Heidelberg, Germany, 2001, pages 433-442
8. Agrawal, Rakesh; & Srikant, Ramakrishnan; Fast algorithms for mining association rules in large databases, in Bocca, Jorge B.; Jarke, Matthias; & Zaniolo, Carlo; editors, Proceedings of 20th International Conference on Very Large Data Bases (VLDB), Santiago, Chile, September 1994, pages 487-499
9. Zaki, M. J. (2000). "Scalable algorithms for association mining". IEEE Transactions on Knowledge & Data Engineering **12** (3): 372–390. doi:10.1109/69.846291.
10. Hájek, Petr; Havel, Ivan; Chytil, Metoděj; The GUHA method of automatic hypotheses determination, Computing 1 (1966) 293-308
11. Hájek, Petr; Feglar, Tomas; Rauch, Jan; & Coufal, David; The GUHA method, data preprocessing & mining, Database

Support for Data Mining Applications, Springer, 2004, ISBN 978-3-540-22479-2

12. Omiecinski, Edward R.; Alternative interest measures for mining associations in databases, IEEE Transactions on Knowledge & Data Engineering, 15(1):57-69, Jan/Feb 2003
13. Aggarwal, Charu C.; & Yu, Philip S.; A new framework for itemset generation, in PODS 98, Symposium on Principles of Database Systems, Seattle, WA, USA, 1998, pages 18-24
14. Brin, Sergey; Motwani, Rajeev; Ullman, Jeffrey D.; & Tsur, Shalom; Dynamic itemset counting & implication rules for market basket data, in SIGMOD 1997, Proceedings of ACM SIGMOD International Conference on Management of Data (SIGMOD 1997), Tucson, Arizona, USA, May 1997, pp. 255-264 .