FAST ALGORITHMS FOR UNSUPERVISED LEARNING IN LARGE DATA SETS

Syed Quddus

Faculty of Science and Technology,
Federation University Australia, Victoria, Australia.

ABSTRACT

The ability to mine and extract useful information automatically, from large datasets, is a common concern for organizations (having large datasets), over the last few decades. Over the internet, data is vastly increasing gradually and consequently the capacity to collect and store very large data is significantly increasing.

Existing clustering algorithms are not always efficient and accurate in solving clustering problems for large datasets.

However, the development of accurate and fast data classification algorithms for very large scale datasets is still a challenge. In this paper, various algorithms and techniques especially, approach using non-smooth optimization formulation of the clustering problem, are proposed for solving the minimum sum-of-squares clustering problems in very large datasets. This research also develops accurate and real time L2-DC algorithm based with the incremental approach to solve the minimum sum-of-squared clustering problems in very large datasets, in a reasonable time.

***GENERAL TERMS* :** Data Mining.

***KEYWORDS* :**

Clustering analysis, k-means algorithm, Squared-error criterion, Large-data sets.

1. INTRODUCTION

Data classification by unsupervised techniques is a fundamental form of data analysis which is being used in all aspects of life, ranging from Astronomy to Zoology. There have been a rapid and massive increase in amount of data accumulated in recent years, due to this, the use of clustering has also expanded further, in its applications such as personalization and targeted advertising. Clustering is now a key component of interactive- systems which gather information on millions of users on everyday basis [1-10, 20]. A process of dividing, classifying or grouping a dataset into meaningful similar partitions or subclasses based on some criteria, normally a distance function between objects, called clusters.

Jae-Kwang Lee et al. (Eds) : CCSEA, AIFU, DKMP, CLOUD, EMSA, SEA, SIPRO - 2017
pp. 69– 76, 2017. © CS & IT-CSCP 2017 DOI : 10.5121/csit.2017.70207

Existing clustering algorithms are not always efficient & accurate in solving clustering problems for large datasets. The accurate and real time clustering is essential and important for making informed policy, planning and management decisions. Recent developments in computer hardware allows to store in RAM and repeatedly read data sets with hundreds of thousands and even millions of data points. However, existing clustering algorithms require much larger computational- time and fail to produce an accurate solution [16,17,18,19].

In this paper, we present an overview of various algorithms and approaches which are recently being used for Clustering of large data and E-document. In this paper we will discuss widely used evolutionary techniques and present results of DC-based clustering methodology in very large & big datasets.

1.1 Heuristics Approaches:

The k-means clustering technique and its modifications are representatives of such heuristics approaches. The global k-means and modified global k-means are representatives of incremental based heuristic algorithms. The within-cluster point scatter should be symmetric and it should attain its minimum value. The distance measure within cluster scatter is known as metric, we can measure this distance by the different methods such as Minkowski and Euclidean distance measure [6, 8, 9].

1.1.1 Minkowski Distance Measure

The distance between two data instances can be calculated using the Minkowski Metric as below[22]:

$D(x, y) = (|x_{i1} - x_{ji}|)g + |x_{i2} - x_{j1}|g + \ldots\ldots\ldots + |x_{in} - x_{jn}|g)1/g$

1.1.2 Euclidean Distance Measure

It is the most commonly used method to measure the distance between two objects when g = 2.when g = 1, the sum of absolute paraxial distance is obtained and when g = Infinity one gets the greatest of the paraxial distance. If the variable is assigned with a weight according to its importance then weighted distance should be measure [22].

1.1.3 Parallel k-means

The concept is to distribute processing of k-means on k machines which result in a satisfactory time complexity. Due to memory limitation it may not be efficient for massive data set.

1.1.4 Partial/Merge k-Means

Partial/merge k-means re-runs the k-means several times to get better result in each partition. However this algorithm is sensitive to the size of partitioning in massive data sets.

Different heuristics have been developed to tackle clustering problems. These heuristics include k-means algorithms and their variations such as h-means and j-means. However, these algorithms are very sensitive to the choice of initial solutions, they can find only local solutions and such solutions in large data sets may significantly differ from global ones [2-9]. However, the success of local methods depends on starting points.

1.2 Heuristics Based on the Incremental Approach

These algorithms start with the computation of the centre of the whole data set A and attempt to optimally add one new cluster centre at each stage. In order to solve Problem (2.5) for k > 1 these algorithms start from an initial state with the k-1 centres for the (k-1)-clustering problem and the remaining k-th centre is placed in an appropriate position. The global k-means and modified global k-means, a single pass incremental clustering algorithm, CURE, DC-based algorithm with the incremental approach are representatives of these algorithms [4, 5, 6, 7].

Usually the massive data set cannot fit into the available main memory, therefore the entire data matrix is stored in a secondary memory and data items are transferred to the main memory one at a time for clustering. Only the cluster representations are stored in the main memory to alleviate the space limitations. DC based algorithm with the incremental approach, is used to solve optimization problems in these massive and very large data sets in a reasonable time [7,8,9].

1.3 Population based evolutionary algorithms

Population based evolutionary algorithms are suitable to generate starting cluster centres as They can generate points from the whole search space. By using before mentioned five Evolutionary algorithms (Genetic algorithm, Particle swarm optimization, Ant colony Optimization, Artificial bee colony and Cuckoo search) in combination with the incremental Algorithm, a new algorithms will be designed to generate starting cluster centres.

Over the last several years different incremental algorithms have been proposed to solve clustering problems. These algorithms attempt to optimally add one new cluster centre at each stage. In order to compute k-partition of a set these algorithms start from an initial state with the k-1 centres for the (k-1)-clustering problem and the remaining k-th centre is placed in an appropriate position. In this paper, our aim is to discuss various clustering techniques for very large datasets and to present how smooth optimization algorithms to solve clustering problems. We propose the L2-DC based algorithm which is based on the combination of smoothing techniques and the incremental approach. In order to find starting points for cluster centres we introduce and solve the auxiliary cluster problem which is non-smooth and non-convex. The hyperbolic smoothing technique is applied to approximate both the cluster and auxiliary cluster functions. Then we apply the Quasi-Newton method with the BFGS update to minimize them. We present results of numerical experiments on five real-world data sets [8, 9].

2. EXPERIMENTAL RESULTS

Algorithms were implemented in Fortran95and compiled using the gfortran compiler. Computational results were obtained on a Laptop with the Intel(R) Core(TM) i3-3110M CPU @ 2.4GHz and RAM 4 GB (Toshiba). Five real-life data sets have been used in numerical experiments [21].The brief description of these data sets is given below in table.1. All data sets contain only numeric features and they do not have missing values. To get as more comprehensive picture about the performance of the algorithms as possible the datasets were chosen so that:(i) the number of attributes is ranging from very few (3) to large (128); (ii) the number of data points is ranging from tens of thousands(smallest13,910) to hundreds of thousands (largest434,874). We computed upto24clusters in all data sets. The CPU time used by algorithms is limited to 20h. Since the L2-DC based algorithm compute clusters incrementally we present results with the maximum number of clusters obtained by an algorithm during this time.

2.1 Tables

Table: 1 the brief description of datasets.

N	**Data Sets**	**Number of instances**	**Number of attributes**
1	Gas Sensor Array Drift Dataset	13910	128
2	Bank Marketing	45211	17
3	Shuttle Landing Control	58000	10
4	Educational Process Mining (EPM): A Learning Analytics Data Set	230318	9
5	3D Road Network (North Jutland, Denmark)	434874	3

We run experiments on these real-life data sets to compute the Cluster function values obtained by algorithms, CPU time and the total number of distance function evaluations for all these five datasets. For numerical results: k - is the number of clusters;

f - is the optimal value of the clustering function obtained by the algorithm; N - is the total number of distance function evaluations; t- is the CPU time.

Table: 2. Results for data set 1

k	**f**	**N**	**t**
2	7.91E+13	6.42E+07	88.2969
4	4.16E+13	3.53E+08	444.0938
6	2.74E+13	7.09E+08	878.5781
8	2.03E+13	1.34E+09	1651.594
10	1.66E+13	1.79E+09	2254.563
12	1.41E+13	2.46E+09	3068.984
14	1.21E+13	3.26E+09	4108.953
16	1.06E+13	3.87E+09	4906.828
18	9.65E+12	4.62E+09	5848.375
20	8.85E+12	5.55E+09	7027.031
22	8.14E+12	6.21E+09	7862.984
24	7.55E+12	6.97E+09	8842.969

Table: 3. Results for data set 2

k	f	N	t
2	2.02E+11	2.96E+08	11.1385
4	7.33E+10	2.85E+09	116.4079
6	3.62E+10	5.86E+09	239.8671
8	2.41E+10	9.34E+09	379.2696
10	1.64E+10	1.45E+10	605.1747
12	1.23E+10	1.77E+10	736.5119
14	1.05E+10	2.07E+10	852.3739
16	8.48E+09	2.65E+10	1116.811
18	7.16E+09	3.38E+10	1449.125
20	6.43E+09	3.99E+10	1741.174
22	5.66E+09	5.17E+10	2349.64
24	5.13E+09	6.85E+10	3165.276

Table: 4. Results for data set 3

k	F	N	t
2	2.13E+09	59566001	8.3773
4	8.88E+08	5.45E+08	67.6108
6	5.67E+08	2.30E+09	291.9091
8	3.73E+08	4.20E+09	538.6871
10	2.85E+08	6.29E+09	808.4908
12	2.21E+08	1.23E+10	1648.26
14	1.78E+08	2.04E+10	2249.597
16	1.46E+08	2.59E+10	2573.205
18	1.20E+08	3.37E+10	3048.447
20	1.06E+08	3.77E+10	3333.289
22	95703872	4.65E+10	3849.216
24	84889772	5.58E+10	4687.924

Table: 5. Results for data set 4.

K	F	N	t
2	2.19E+19	1.83E+09	93.117
4	4.10E+17	8.34E+09	396.6793
6	1.54E+17	2.08E+10	999.28
8	8.41E+16	3.62E+10	1765.261
10	5.78E+16	4.99E+10	2399.046
12	3.79E+16	7.85E+10	3889.526
14	2.79E+16	2.96E+11	15253.17
16	2.09E+16	3.36E+11	17159.49
18	1.49E+16	4.01E+11	20600.2
20	1.09E+16	5.38E+11	28124.38
22	7.80E+15	5.92E+11	30885.5
24	6.40E+15	6.70E+11	34813.82

Table: 6. Results for data set 5.

k	f	N	t
2	4.91E+07	8.18E+10	1443.181
4	1.35E+07	2.74E+11	4860.913
6	6.38E+06	4.68E+11	8231.611
8	3.78E+06	6.63E+11	11673.59
10	2.57E+06	8.63E+11	15169.38
12	1.85E+06	1.07E+12	18800.85
14	1424129	1.29E+12	22818.89
16	1139559	1.5E+12	26609.68
18	948040.9	1.72E+12	30518.32
20	808708.8	1.94E+12	34452.17
22	703308.7	2.17E+12	38592.34
24	638434.2	2.41E+12	42971.4

The results of implementation of the L2-DC based algorithm are shown, respectively, in Table 2 for vowel dataset, Table 3. Table 4, Table 5, and Table 6 for five real time datasets.

All five data sets can be divided into two groups. The first group contains data sets with small number of attributes (3or 9). 3D-Road Network data set and Educational Process Mining (EPM): A Learning Analytics Data Set belongs to this group. The number of points in these datasets ranges from 2072862 to 1304622. Results presented in Tables5 and 6 demonstrate that in these datasets the performance of algorithm is similar in the sense of accuracy. All algorithms can find at least near best known solutions in these datasets.

The second group contains data sets with relatively large number of attributes. Gas Sensor Array

Drift, Shuttle Control data and Bank Marketing data sets belong to this group. The number of attributes in these data sets ranges from 10 to128. Results show that the algorithm is very efficient to find (near) best known solutions. The dependence of the number of distance function evaluations on the number of clusters in group1 of datasets is similar and the dependence of the number of distance function evaluations on the number of clusters in group2 of datasets is also similar.

The dependence of the CPU-time on the number of clusters for all datasets in group1 is similar. As the number of clusters increase, the dependence of CPU time monotonically increases. It is obvious that as the size (the number of data points) of a data set increase this algorithm requires more CPU time. The dependence of the CPU-time on the number of clusters for all datasets in group1 is similar in a sense, as the number of clusters increase, the dependence of CPU time monotonically increases. But the algorithm takes almost similar time pattern in clustering datasets: Shuttle Control data and Bank Marketing data sets but in case of Gas Sensor Array Drift dataset, the algorithm requires much more CPU time.

3. CONCLUSION

In this paper the minimum sum-of-squares clustering problems are studied using L2-DC based approach. An incremental algorithm based on DC representation is designed to solve the minimum sum-of-squares clustering problems. A special algorithm is designed to solve non-smooth optimization problems at each iteration of the incremental algorithm. It is proved that this algorithm converges to inf-stationary points of the clustering problems.

4. FUTURE WORK

As we know very large data set clustering is an emerging field. In this research, different evolutionary methods, their features and their applications have been discussed. We have implemented one of the techniques and we may implement more in future. The expectation is to implement the best technique which can efficiently solve the minimum sum-of-squares clustering problems and find the best solution in real time.

REFERENCES

[1] Yasin, H., JilaniT. A., and Danish, M. 2011. Hepatitis-C Classification using Data Mining Techniques. International Journal of Computer Applications.Vol 24– No.3.

[2] K.S. Al-Sultan, A tabu search approach to the clustering problem, {\em Pattern Recognition}, 28(9)(1995) 1443-1451.

[3] A.M. Bagirov, Modified global k-means algorithm for sum-of-squares clustering problems, {\em Pattern Recognition,} 41(10), 2008, 3192--3199.

[4] A.M. Bagirov, A.M. Rubinov, J. Yearwood, A global optimisation approach to classification, {\em Optimization and Engineering,} 3(2) (2002) 129-155.

[5] A.M. Bagirov, A.M. Rubinov, N.V. Soukhoroukova, J. Yearwood, Supervised and unsupervised data classification via nonsmooth and global optimization, {\em TOP: Spanish Operations Research Journal,} 11(1)(2003) 1-93.

[6] A.M. Bagirov and J. Ugon, An algorithm for minimizing clustering functions, \emph{Optimization,} 54(4-5), 2005, 351-368.

[7] A.M. Bagirov, J. Ugon and D. Webb, Fast modified global k-means algorithm for sum-of-squares clustering problems, {\em Pattern Recognition,} 44, 2011, 866--876.

[8] A.M. Bagirov, J. Yearwood, A new nonsmooth optimization algorithm for minimum sum-of-squares clustering problems, {\em European Journal of Operational Research,} 170(2006) 578-596.

[9] A.M. Bagirov, A. Al Nuaimat and N. Sultanova, Hyperbolic smoothing method for minimax problems, \emph{Optimization,} accepted.

[10] H.H. Bock, Clustering and neural networks, in: A. Rizzi, M. Vichi, H.H. Bock (eds), {\em Advances in Data Science and Classification}, Springer-Verlag, Berlin, 1998, pp. 265-277.

[11] D.E. Brown, C.L. Entail, A practical application of simulated annealing to the clustering problem, {\em Pattern Recognition}, 25(1992) 401-412.

[12] G. Diehr, Evaluation of a branch and bound algorithm for clustering, {\em SIAM J. Scientific and Statistical Computing}, 6(1985) 268-284.

[13] R. Dubes, A.K. Jain, Clustering techniques: the user's dilemma, {\em Pattern Recognition}, 8(1976) 247-260.

[14] P. Hanjoul, D. Peeters, A comparison of two dual-based procedures for solving the p-median problem, {\em European Journal of Operational Research,} 20(1985) 387-396.

[15] P. Hansen, B. Jaumard, Cluster analysis and mathematical programming, {\em Mathematical Programming,} 79(1-3)(1997) 191-215.

[16] A. Likas, M. Vlassis, J. Verbeek, The global k-means clustering algorithm, {\em Pattern Recognition}, 36(2003) 451-461.

[17] O. du Merle, P. Hansen, B. Jaumard, N. Mladenovic, An interior point method for minimum sum-of-squares clustering, {\em SIAM J. on Scientific Computing,} 21(2001) 1485-1505.

[18] H. Spath, {\em Cluster Analysis Algorithms}, Ellis Horwood Limited, Chichester, 1980.

[19] L.X. Sun, Y.L. Xie, X.H. Song, J.H. Wang, R.Q. Yu, Cluster analysis by simulated annealing, {\em Computers and Chemistry,} 18(1994) 103-108.

[20] A.E. Xavier, The hyperbolic smoothing clustering method, \emph{Pattern Recognition}, 43(3), 2010, 731-737.

[21] http://www.ics.uci.edu/mlearn/MLRepository.html, UCI repository of machine learning databases.

[22] Neha Khan, Mohd Shahid, Mohd Rizwan, ‘Big data classification using evolutionary techniques:A survey’,(2015), IEEE International Conference (ICETECH), India.