Volume 3, Issue 11, November 2013 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Improvement of Clustering Using Pairwise Distances Approach-Single Linkage Clustering Algorithm

Praveen Kumar Reddy.M
School of Computing Science
India

D.Adithya Chandra Varma
School of Computing Science
India

Prof. VarunKumar.M
School of Information Technology
India

Abstract— Hierarchical clustering is majorly used in data mining stream to manage categorical data. Partitioning methods meet the basic clustering requirement of organizing a set of objects into a number of exclusive groups, in some situations we may want to partition our data into groups at different levels such as in a hierarchy. Euclidean parameter is base for clustering using pair wise distances with clusters of different shapes, sizes and dimensions .Single linkage clustering, spectral clustering algorithms are working based on the pair wise distances approach. These algorithms have feasible properties in terms of clustering and robustness to outliers. Our work on the parameter single linkage clustering reflects the clustering in pair wise distance manner and then applying the algorithm on the data. Now we are going to parallelize single linkage-clustering algorithm using open-mp for giving good performance to cluster larger groups of data.

Keywords:

I. INTRODUCTION

There are mainly three clear approaches for the representation of Euclidean parameter or pair wise distance approach. There are three important algorithms using these approaches

1. Pair wise clustering approach. [2]
2. Spectral Clustering. [2]
3. Single linkage clustering [2].

Our main interest is single linkage clustering approach. It is the hierarchical clustering algorithm, in which the clusters are formed based on Euclidian distances approach. So many researchers use their own parameter for the clustering like Euclidian approach, Warden Approach etc. In our algorithm, we will cluster the parameters with the minimum Euclidian distance with respect to other points or data sets in the distance matrix and we recursively merge into single huge cluster in which the sub clusters are linked together obviously relates with the hierarchal clustering definition. We will stop at a point where the merging is not possible.

II. RELATED WORK

Cluster analysis [2] is a statically analysis which shows the correlation between different items or data sets. In the information retrieval cluster analysis gives a lot of ease to retrieve data .In the cluster analysis there are hierarchical and non- hierarchical methods just divides given item sets into finite clusters. In the hierarchical methods, it requires a huge computation and it gives a single cluster in which the sub clusters are linked so our main interest is to focus on the hierarchal methods. In hierarchal methods of clustering
There are two types of clustering they are
Agglomerative clustering.
Divisive clustering methods.

*A.*AGGLOMERATIVE CLUSTERING

In these clustering it is merging based approach here it will merge all the clusters based on the parameter like closest distance, farthest distance and average distance. Here the terminating condition is specified by the end user for the desirable clusters

B. DIVISIVE CLUSTERING

In the divisive clustering[2] here we are using splitting approach we can split the patterns based on the terminating condition specified by the user or each individual object forms its own cluster here agglomerative clustering approach is more preferable than the divisible clustering approach. Single linkage clustering is the main part of the Agglomerative clustering [2]

*C.*CLUSTERING USING PAIR WISE DISTANCES

In our clustering analysis the three approaches which describe the Euclidian approach [2] that is Pair wise clustering. If we take a graph like structure if we give epsilon it has the smallest distance then the connected components in the graph with respect to epsilon are clustered if we give epsilon as too large then we will get more pairs are connected components. Spectral clustering [3] will use the technique of dimensionality reduction technique where we have to use the and Eigen vectors to reduce the dimension. After getting the normalized matrix we to apply the K-means [3] on the clusters C1 to Ck.

*D.*SINGLE LINKAGE CLUSTERING ALGORITHM

ALGORITHM

```
1. Start.
2. Read the Data sets into distance matrix.
3. Repeat
        3.1 Compute the Euclidean distance between the patterns
        3.2 Repeat 1….n
                3.2.1 Find the minimal distance in the distance matrix.
                3.2.2Merge those patterns with minimal distance.
        3.3 Recomputed the distance matrix with respect to the clusters.
        3.4 Repeat until (n-1) or (itemsize-1).
4. Find the accuracy of the algorithm using RAND-INDEX method.
5. Stop.
```

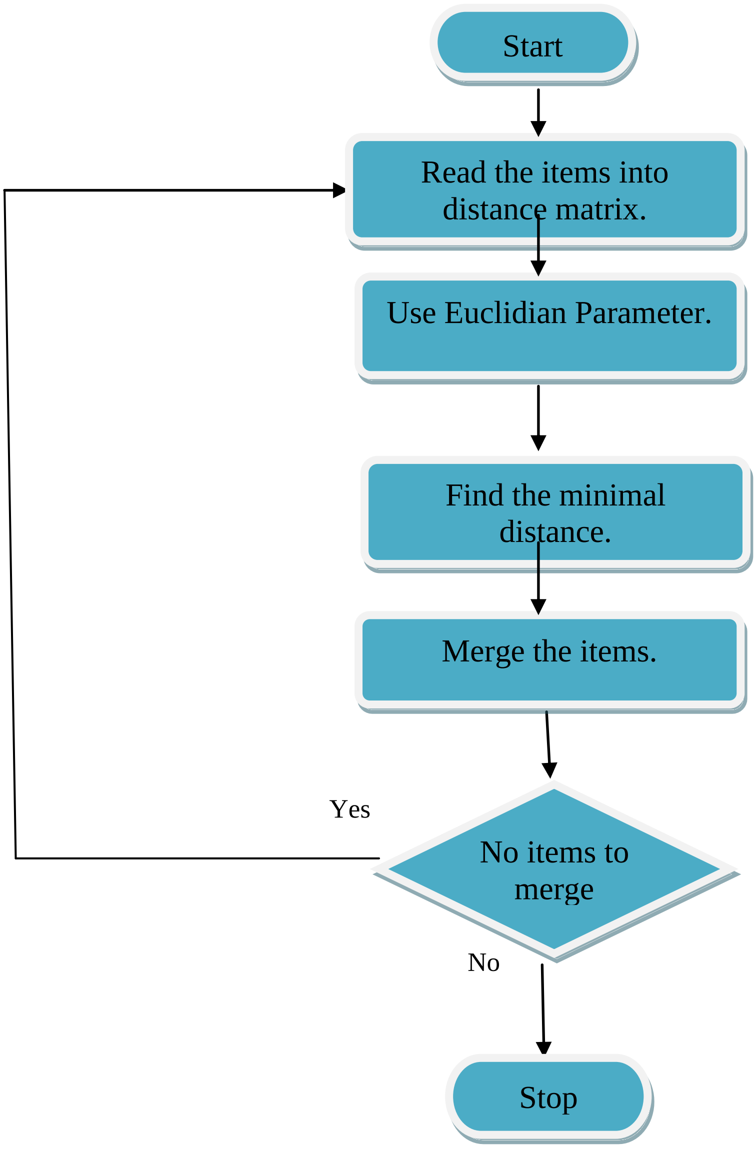

Fig-1: Flow chart for single linkage clustering.

© 2013, IJARCSSE All Rights Reserved

E. Descriptions for parallel algorithm

We parallelize the single linkage algorithm using open-mp constructs. Firstly, read the patterns from the file into the distance matrix .After getting the distance matrix we have to parallelize block by assigning the patterns to the threads (thread1, thread2, thread3, and thread4). Each thread in the block finds the Euclidean distance for the patterns relative to all the patterns.afterfinding that, find out the minimal distance clusters. Merge the minimum distance clusters if there are no clusters to merge stop else repeat the algorithm. While parallelizing critical sections, scheduling and sections are used to give good performance. Place each pattern in its own cluster. Construct a list of inter pattern distances for all distinct unordered pairs of patterns and sort this list in ascending order. Step through the sorted list of distances forming each distinct dissimilarity value dk of a graph patterns where pack of patterns closer than dk are connected by a graph edge if all the patterns are members of a connected graph, stop. Otherwise repeat this step. The output of the algorithm is a nested hierarchy a graphs which can be cut at a desired dissimilarity level formatting a partition (clustering) identified by simply connected components in the corresponding graphs.

F. Single linkage clustering -example

Consider the training data set with values {$i_{1,}$ $i_{2,}$ $i_{3,}$ $_{i4,}$ $i_{5,}$ i_6} with x and y coordinates as {{0.40,0.53},{0.22,0.38},{0.35,0.32},{0.26,0.19},{0.08,0.41},{0.45,0.30}}

1. At first we have to calculate the Euclidean distance using the parameter

D ($i_{1,}$ i2) = $\sqrt{| X_{I1} - X_{I2|} |^2 + | Y_{I1} - Y_{I2|} |^2}$ -------- equation (1)

2. We have to calculate the distance matrix using Euclidean formulae for the given training data sets

3. Find the minimum distance pattern in the distance matrix and merge them into a cluster, we have to repeat the step 2 for the newly formed clusters.

For our training data sets, the clusters are formed like this

Iteration-1 :{ i3, i6}
Iteration-2 :{ i2, i5}
Iteration-3 :{ i2, i5, i3, i6}
Iteration-4 :{ i2, i3, i5, i6, i4}
Iteration-5: finally the clusters are {i_2, i3, i5, i6, i4} and {i_1}

G. Dendrogram

Dendrogram [2] is the symbolic format to represent the hierarchical clustering methods. The Dendrogram for our training data-sets looks like

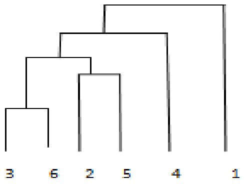

Fig-2: Dendrogram representation

III. ANALYSING PERFORMANCE

A. Verified datasets

Datasets on which single link clustering algorithm is implemented are:

1. Iris: This dataset is an iris plant data description with 3 classes, 150 instances; every instance has 4 attributes each.
2. Wine: This is a wine recognition data base containing 3 classes, class 1, 2, 3 have 59, 71 and 48 instances respectively. The number of attributes in this dataset is 13.
3. Iono: This is a dataset describing ionosphere database. It contains 351 numbers of instances and 2 classes.
4. Heart: This dataset consists of 270 patterns, each pattern containing 13 attributes. The data set consists of two classes.

B. Performance Analysis

A. Table values for two-core machine

S.NO	DATASET	DATA SIZE	SERIAL EXECUTION TIME	PARALLEL EXECUTION TIME	SPEED UP
1.	WINEDATA	278	0.265015	0.265500	0.9981
2.	HEARTDATA	860	2.120020	1.7894	1.1244
3.	IONODATA	1024	3.14300	3.0678	1.0245
4.	IRISDATA	1500	4.254517	3.38062	1.2585

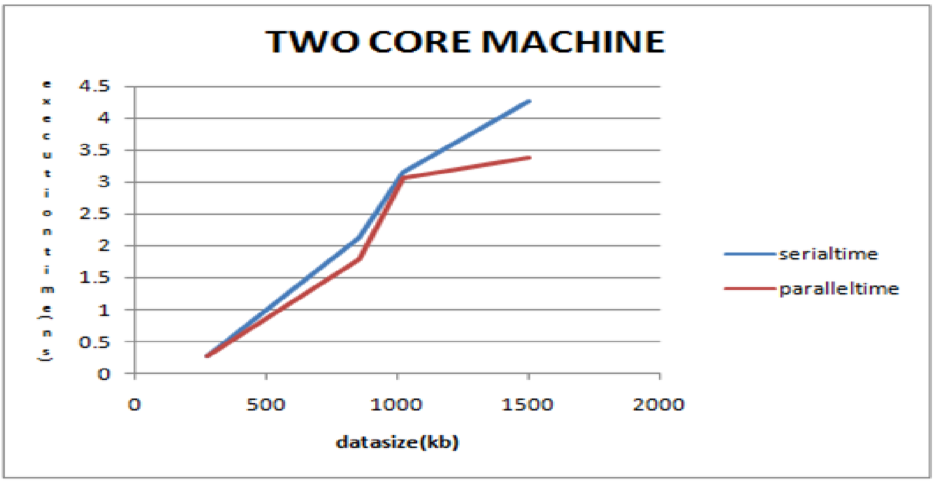

Fig-3: Graphical representation for two core machine

Observations

1. In the dual core system, we plotted the graph for data size and execution time.

2. In this scenario performance may not improves many times because of the thread latency in the parallel execution but it is not possible in serial execution (using one core)

B. **Table values for four-core machine**

S.NO	DATASET	DATASIZE	SERIAL EXECUTION TIME	PARALLEL EXECUTION TIME	SPEED UP
1.	WINEDATA	278	0.265015	0.2655	1.019288
2.	HEARTDATA	860	2.120020	1.671544	1.268300
3.	IONODATA	1024	3.14300	2.123000	1.480452
4.	IRISDATA	1500	4.254517	2.858277	1.488490

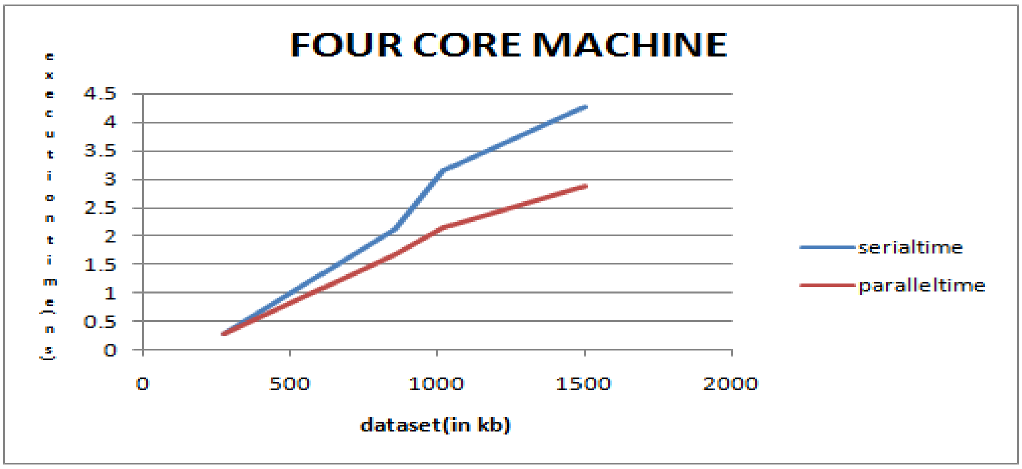

Fig-4: Graphical representation for four core machine

Observations

1. In the above graph the performance is clearly varied because as the cores increase execution time decreases and speed up increases.

2. In the above graph thread latency doesn't affected the performance.

C. **Table values for eight-core machine**

S.NO	DATASET	DATASIZE	SERIAL EXECUTION TIME	PARALLEL EXECUTION TIME	SPEED UP
1.	WINEDATA	278	0.265015	0.12499	2.1202
2.	HEARTDATA	860	2.120020	0.102470	2.4838
3.	IONODATA	1024	3.14300	0.70376	3.0124
4.	IRISDATA	1500	4.254517	1.27251	3.3434

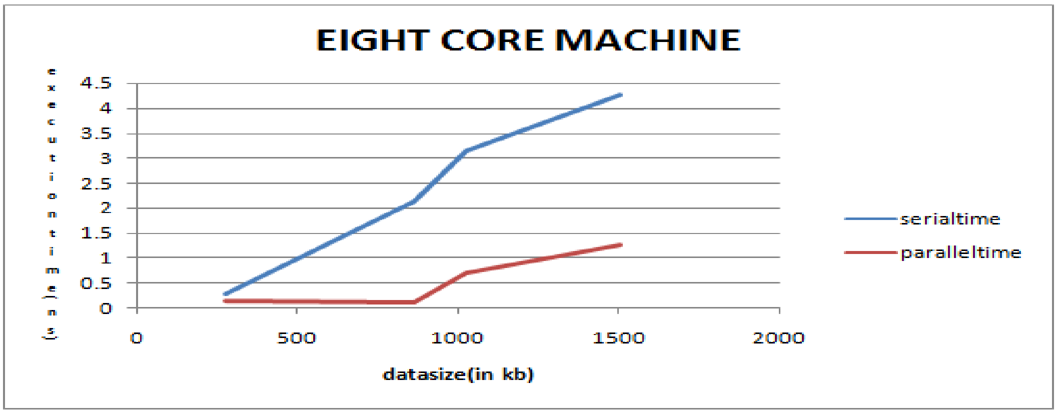

Fig-5: Graphical representation for eight core machine

Observations

1. In the above graph the performance is clearly varied because as the cores increase execution time decreases and speed up increases.
2. In the above graph thread latency doesn’t affected the performance.

D. **Table values for data size and speed up**

S.NO	DATASET	DATASIZE	2-CORE SPEED UP	4-CORE SPEED UP	8-CORE SPEEDUP
1.	WINEDATA	278	0.9981	1.019288	2.1202
2.	HEARTDATA	860	1.1244	1.268300	2.4838
3.	IONODATA	1024	1.0245	1.480452	3.0124
4.	IRISDATA	1500	1.2585	1.488490	3.3434

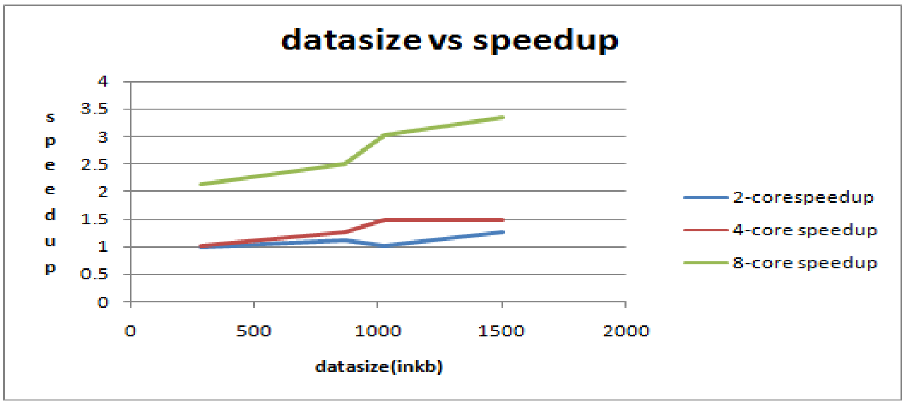

Fig-6: Graphical representation for speedup and data size

© 2013, IJARCSSE All Rights Reserved

Observations

In the above graph the performance is clearly varied because as the core increases execution time decreases and speed up increases.

IV. CONCLUSION

We improvised the single linkage algorithm which clusters the data based on the Euclidean metric in an efficient way. Algorithm grows more efficient proportional to the size of the datasets. This algorithm gives non-isotropic clusters including well-separated, chain like and concentric clusters... Using the parallel single linkage algorithm, the speed-up factor is increased than the corresponding serial single linkage algorithm, as the experimental results show. Hence the overall efficiency of the parallel single linkage algorithm is better than the corresponding serial single linkage algorithm.

ACKNOWLEDGMENT

The authors would like to thank the School of Computer Science and Engineering, VIT University, for giving them the opportunity to carry out this project and also for providing them with the requisite resources and infrastructure for carrying out the research.

REFERENCES

[1] Ery Arias Castro, “Clustering based on pair wise distances when data is of Mixed Dimensions,” IEEE Trans Inf.Theory, vol 57, no.3, pp.468-1123, March. 2011.

[2] A.Y.Ng, .M.I.Jordan, and Weiss “Clustering Analysis and an Algorithm, in Advances in Neural information,” IEEE Trans. Inf.Theory, vol.18, no.5, pp.645-390, May.2010.

[3] Dasgupta and P.M Long, “Performance guarantees for Hierarchal clustering,”IEEE Trans. Inf.Theory, vol.70, no.3, pp.653-790, March.2010.

[4] G. Biau, L. Devroye, and G. Lugosi, “On the performance of clustering in Hilbert spaces,” IEEE Trans.Inf.Theory, vol. 54, no.7, pp. 781–790, March.2011.

[5] L. Birge, “Estimating a density under order restrictions: Nonasymptoticminimax risk,” IEEETrans.Inf.Theory, vol. 15, no. 3, pp. 995–1012, Apr.2010.

[6] Y. Yang, “Minimax nonparametric classification. I. rates of convergence,” IEEE Trans. Inf. Theory, vol. 45 no.6, pp. 2271–2284, March.2011.

[7] C. Scott and R. D. Nowak, “Minimax-optimal classification with dyadic decision trees,” IEEE Trans. Inf. Theory, vol. 52, no. 4, pp. 1335–1353, Apr. 2010.

[8] Data mining concepts& Techniques by Jiaweitan&Kamber. , Elsevier publications, 3rd edition, 2011.