Distributed and clustering techniques for Multiprocessor Systems

Elsayed A. Sallam

Associate Professor and Head of Computer and Control Engineering Department, Faculty of Engineering,

Tanta University, Egypt

Abstract

The Distributed Clustering area aims to solve several problems that currently limit the scalability of network resources. While, clustering methods determine relationships among the objects, and allow the determination of similar groups of objects. The goal of this paper is to partition the network into such a set of clusters, which have observed similar phenomena. This paper presents the results of an experimental study of some comparisons clustering techniques: K-means, and Hybrid Hierarchical-K-means.

Keywords: **K-means, cluster algorithms, and Distributed system.**

1. Introduction

Hierarchical clustering is often depicted as the better quality clustering approach, but is limited because of its second degree complexity. In contrast, K-means and its derivatives have a time complexity [1], but are consider evaluating less significant clusters. Sometimes K-means and hierarchical approaches are shared so as to "get the better of both clusters", which requires no parameter establishment to identify the similarities and dissimilarities between the objects [3]. This combination is facilitated by a new key concept, that of a 'mutual cluster'. A mutual cluster is defined as a group of objects collectively closer to each other than to any other object [2]. We compared between the proposed HHK-clustering algorithm and the existing K-means algorithm in different cases.

This paper is organized as follows. This section provides the introduction. While, section 2 are describes the clustering techniques. Section 3, IV, and V, includes three hybrid algorithms for clustering distributed objects. Simulation results and there analysis are covered in section VI. The conclusion of the paper summarizes the work presented. Finally, a list of references used in the research is given.

2. Clustering Techniques

There are many different algorithms that are available today, and the two of the algorithms that we investigate, fall into two general categories: hierarchical and nonhierarchical.

2.1 Hierarchical clustering

There are basically two types of hierarchical clustering procedures – agglomerative and divisive. In agglomerative hierarchical methods, each observation starts out as its own cluster. In subsequent steps, the two closest clusters are combined into a new aggregate cluster, thus reducing the number of clusters by one in each step. Two groups of individuals formed at an earlier stage may join together in a new cluster. Eventually, all individuals are fused into one large cluster. In divisive methods, an initial single group of objects is divided into two subgroups such that the objects in one subgroup are "far from" the objects in the other. These subgroups are then further divided into dissimilar subgroups; the process continues until there are as many subgroups as objects (each object forms a cluster). In both hierarchical methods, a hierarchy of a tree-like structure is constructed and usually represented as a dendrogram or tree graph [8]. The dendrogram illustrates the mergers or divisions that have been made at successive levels. In particular, Wishart [6] contends that the "top down" decision tree approach has inherently greater risk of misclassification by inefficiently splitting on a single variable than the "bottom up" approach. Each classification generated in a decision tree

is univariate by definition, and this limits the range of possible segments available for consideration. By comparison, the agglomerative approach is multivariate and exploratory, and allows for more feasible segments to be investigated in terms of the actual distribution of the scatter. Hence, this project concentrates on agglomerative hierarchical algorithms mainly (divisive methods act almost as agglomerative methods in reverse).The following are the steps in the agglomerative hierarchical clustering algorithm for grouping *N* objects [11]:

1. Start with *N* clusters, each containing a single entity and an *N* x *N* symmetric matrix of distances (or similarities) D= {d*jk*}
2. Search the distance matrix for the nearest (most similar) pair of clusters. Let the distance between "most similar" clusters *U* and *V* be *Duv*.
3. Merge clusters *U* and *V* . Label the newly formed cluster (*UV*). Update the entries in the distance matrix by
   a. deleting the rows and columns corresponding to clusters *U* and *V* and
   b. adding a row and column giving the distances between cluster (*UV)* and the remaining clusters.

Repeat Steps 2 and 3 a total of *N-1* times. (All objects will be in a single cluster after the algorithm terminates.) Record the identity of clusters that are merged and the levels (distances or similarities) at which the mergers take place [10].

2.1.1 VARIATIONS OF HIERARCHICAL ALGORITHM

This section describes the various variants of agglomerative hierarchical clustering algorithms – single linkage, complete linkage, average linkage and Ward's method (ESS).

2.1.1.a LINKAGE METHODS

The inputs to a linkage algorithm can be distances or similarities between pairs of objects. Single linkage, complete linkage and average linkage are the three linkage-based hierarchical clustering algorithms implemented.

Table 1: Between-clusters distances

Between-clusters distance	$d(Q_k,Q_l)$
Single linkage	$d_s = \min_{i,j}\left\{\left\|x_i - x_j\right\|\right\}$
Complete linkage	$d_c = \max_{i,j}\left\{\left\|x_i - x_j\right\|\right\}$
Average linkage	$d_a = \frac{\sum_{i,j}\left\|x_i - x_j\right\|}{N_k N_l}$
Between-clusters distance $d(Q_k, Q_l)$; $x_i \in Q_k, x_j \in Q_l, k \neq l$. N_k is the number of samples in cluster Q_k.	

Table 1 shows the between-clusters distance definition for each of the linkage methods. In this case, dissimilarity coefficient is employed. The selection of the distance criterion or similarity coefficient depends on application. Single Linkage: Groups are formed from the individual entities by merging nearest neighbours, where the term nearest neighbour connotes the smallest distance or largest similarity. Complete Linkage: The distance (similarity) between clusters is determined by the distance (similarity) between the two elements, one from each cluster, which are most distant (or least similar) [6,12]. Average Linkage: Average linkage treats the distance between two clusters as the average distance between all pairs of items where one member of a pair belongs to each cluster.

2.2 K- means algorithm

In this section, we briefly describe the direct k-means algorithm [9, 8, 3]. The number of clusters K is assumed to be fixed in k-means clustering. Let the K

prototypes $(w_1, \ldots, w_k)$ be initialized to one of the n input patterns $(i_1, \ldots, i_n)$. Therefore,

$$w_j = i_l,\ j \in \{1, \ldots, k\},\ l \in \{1, \ldots, n\}$$

Figure 1 shows a high level description of the direct k-means clustering algorithm. C_j is the j^{th} cluster whose value is a disjoint subset of input patterns. The quality of the clustering is determined by the following error function:

$$E = \sum_{j=1}^{k} \sum_{i_l \in C_j} |\, i_l - w_j \,|^2$$

The appropriate choice of K is problem and domain dependent and generally a user tries several values of. Assuming that there are ✓ patterns, each of dimensions, the computational cost of a direct k-means algorithm per iteration (of the repeat loop) can be decomposed into three parts:

1. The time required for the first *for* loop in Figure 1 is *O(nkd)*.
2. The time required for calculating the centroids (second *for* loop in Figure 1) is *O(nd)*.
3. The time required for calculating the error function is *O(nd)*.

The number of iterations required can vary in a wide range from a few to several thousand depending on the number of patterns, number of clusters, and the input data distribution. Thus, a direct implementation of the k-means method can be computationally very intensive. This is especially true for typical data mining applications with large number of pattern vectors [5, 7].

Function Direct-k-means()

Initialize K prototypes $(w_1, \ldots, w_k)$ such that $w_j = i_l,\ j \in \{1, \ldots, k\},\ l \in \{1, \ldots, n\}$

Each cluster C_j is associated with prototype w_j

Repeat

for each input vector i_l where $l \in \{1, \ldots, n\}$, *do*

Assign i_l *to the cluster* C_j *with nearest prototype* w_j (i.e., $|\, i_l - w_{j*} \,| \leq |\, i_l - w_j \,|,\ j \in \{1, \ldots, k\}$)

For each cluster C_j *where* $j \in \{1, \ldots, k\}$, *do*

Update the prototype w_j *to be centroid of all samples currently in* C_j, *so that* $w_j = \sum_{i_l \in C_j} i_l / C_j$

Compute the error function:

$$E = \sum_{j=1}^{k} \sum_{i_l \in C_j} |\, i_l - w_j \,|^2$$

Until E does not change significantly or cluster membership no longer changes.

---

3. Novel Hybrid Hierarchical K-means (HHK) clustering algorithm

Clustering is a technique to divide data sets into several subsets whose elements share similar attributes. Among clustering algorithms, Hierarchical and K-means clustering are the two most popular and classic methods. However, both have their innate disadvantages. Hierarchical clustering cannot represent distinct clusters with similar expression patterns. Also, as clusters grow in size, the actual expression patterns become less relevant. K-means clustering requires a specified number of clusters in advance and chooses initial centroids randomly; in addition, it is sensitive to outliers. We present a hybrid approach to combine the merits of the two classic approaches and discard disadvantages we mentioned. A brief description of HHK clustering algorithm (Chen et al., 2005) is as follows [8]. First, we carried out agglomerative hierarchical clustering and let the program stop at a certain terminal point (a user-defined percentage that is determined by the whole clustering process carried out by hierarchical clustering). From the clusters generated from hierarchical clustering, we computed the mean value of each cluster

as the initial point for K-means to obtain the initial centroid. Also, the number of clusters generated from hierarchical clustering is K-means number of clusters. After that, we worked on K-means clustering with which every cluster must at least contain the same objects generated from hierarchical clustering. This is because that hierarchical clustering had already put objects that were very close with one another into clusters, and the goal of K-means clustering is to put close objects together, which is in the same direction as what hierarchical clustering accomplished. Therefore, we can trust the results of hierarchical clustering. We apply HHK clustering algorithm for super-rules (He et al., 2005) generation in this paper. To avoid human intervention and let the super-rule present the original data nature, we modified our HHK clustering algorithm to become a fully parameter-free algorithm. The original HHK required the user to decide when to stop the hierarchical clustering and proceed to K-means clustering. Since the role of HHK clustering algorithm is to generate the super-rules, the results of the clustering should be as detailed as possible. Therefore, the approach we propose to avoid the parameter set-up is to let the agglomerative hierarchical clustering complete execution, and we record the number of clusters it generated. After that, we carry out the HHK clustering algorithm and let the hierarchical clustering stop when it generates the largest number of clusters. The reason for this process is that while the hierarchical clustering stops at the point we mentioned, the HHK clustering may generate the largest number of super-rules as well as the most detailed information. We may apply the HHK on the super-rules again to generate super-super-rules if necessary [8]. The advantage of this method was that people didn't have to choose an arbitrary number of k; instead, the user only had to choose the percentage for execution of hierarchical clustering (the stop point for the first step). The initial centroids were also generated in a much better way. Besides, points close to one another wouldn't be chosen as different centroids since they were already clustered together [9].

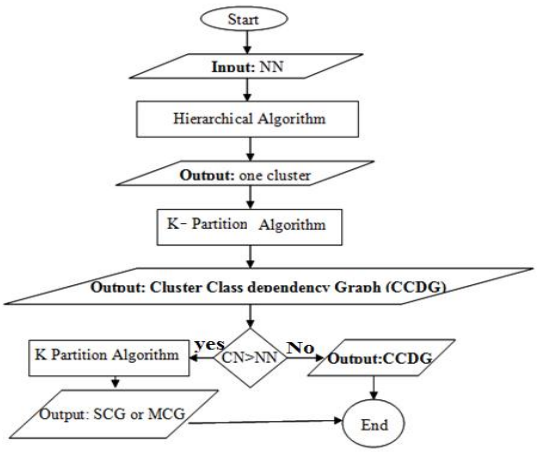

Figure 1: Steps of the HHK clustering algorithm

Figure 1 shows the steps of the new hybrid algorithm work as:

(1) Find two objects (clusters) with closest distance among all, and cluster them together.

(2) The value of attributes of new cluster are the average of attributes of two old objects (clusters);

(3) **Until** the percentage of hierarchical clusters requested by user is done;

(4) Calculate average attribute values of members of clusters that generate from step (1) to (4) as initial cluster centroids;

(5) **repeat**

(6) **for** all objects

(7) **if** the object already appeared in step(2)

then the object remain in original cluster;

(9) **else**
calculate distances between the object and existed clusters
(10) **if** the shortest distance lower than threshold
(11) **then** the object are assigned to the closest cluster
(12) **else**
the object belongs to minor group
(13) **end for loop**
(14) update the centroid attribute value;
(15) **until** no member changes belonging cluster [9].

4. Simulation results

The time cost for communications that occurs between different classes in the OO system. The computed values are then used as the weights assigned to the corresponding edges in the CDG. The communication cost between two clusters can be computed according to the following equation:

$$T_{v,k} = \sum_{i,j=1} (W_{ij})$$

Where:-
i, j: the two objects connected located at different clusters: v, k
W_{ij}: communication cost between objects i, j
The experiment uses the adjacency matrix of the CDG which is generated randomly having 200, 350, 400 objects. The performance of the different algorithms was reported considering architectures with different number of clusters (3 to 8 clusters).

Table 2: Inter-cluster communication cost measured over 200 classes partitioned into 3 to 8 clusters

No. of Clusters	HHK-clustering	K- means
3	673	1237
4	979	1957
5	1782	3476
6	2848	3778
7	3727	4917
8	4770	5779

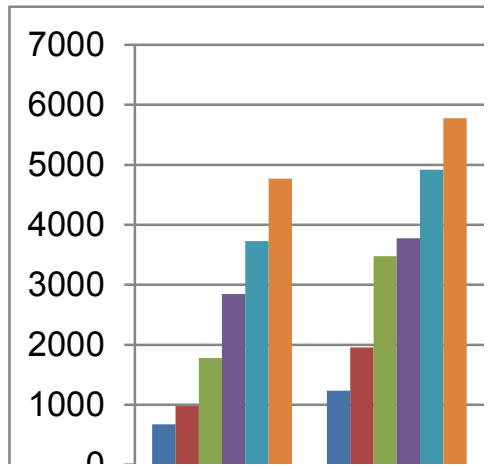

Figure 2: Inter-cluster communication cost measured over 200 classes after applying the different clustering algorithms

Table 3: Inter-cluster communication cost measured over 350 classes partitioned into 3 to 8 clusters

No. of Clusters	HHK-clustering	K- means
3	1533.88	834.52
4	2426.68	1213.96
5	4310.24	2209.68
6	4684.72	3531.52
7	6097.08	4621.48
8	7165.96	5914.8

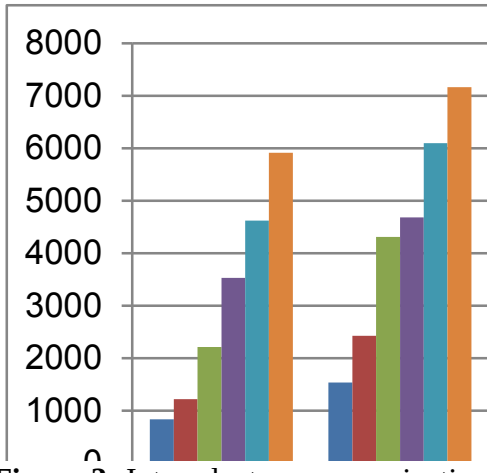

Figure 3: Inter-cluster communication cost measured over 350 classes after applying the different clustering algorithms

Table 4: Inter-cluster communication cost measured over 400 classes partitioned into 3 to 8 clusters

No. of Clusters	HHK-clustering	K- means
3	1346	2474
4	1958	3914
5	3564	6952
6	5696	7556
7	7454	9834
8	9540	11558

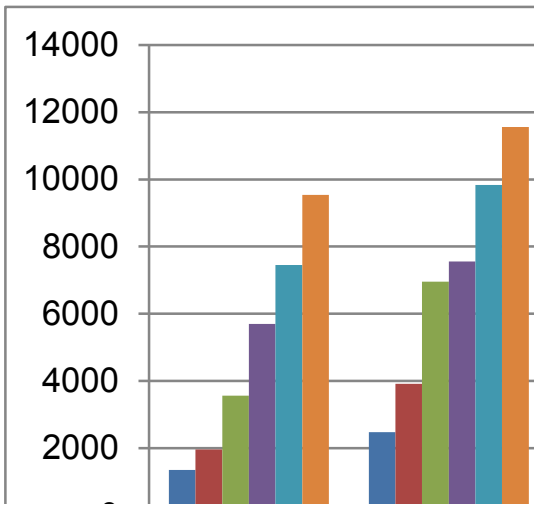

Figure 4: Inter-cluster communication cost measured over 400 classes after applying the different clustering algorithms

From these results we can conclude that the HHK- clustering algorithm authenticate the best performance over the other algorithm since it gives the minimum inter-cluster communication cost considering different numbers of clusters about 7%.

5. Conclusion

In this paper, we have proposed a novel clustering method. Our method of automatically finding good initial centroids for K-means clustering and dealing with outliers seems to provide better performance and more meaningful results. Comparing the proposed clustering algorithm with classical other clustering algorithm under different operating conditions, the simulation results show that the proposed algorithm achieved better performance. The main ingredient of our method is a novel geometric characterization of a class of communication cost that can be used to support hierarchical with k-Partition for N-body objects. We show that communication graphs of this hybrid algorithm H-K Partition algorithm method have a good partition that can be found in distributed technique particularly in the major clusters and objects number.

6. References

[1] Bradley, P.S. and Fayyad, U.M. (1998) 'Refining initial points for K-means clustering', *Proc. 15th International Conf. on Machine Learning*, Madison, Wisconsin, USA p.727.

[2] Brown, D.E. and Huntley, C.L. ,'A practical application of simulated annealing to clustering', *Pattern Recognition*, Vol. 25, No. 4, PP: 401–412, 1990.

[3]Chen, B., Tai, P.C. and Harrison, R. , 'Novel hybrid hierarchical-K-means clustering method (HK-means) for microarray analysis', *Computational Systems Bioinformatics Conference, 2005, Workshops and Poster Abstracts. IEEE*, Stanford University, California, USA, PP: 105–108, 2005.

[4] Chen, B., Tai, P.C., Harrison, R. and Pan, Y. ,'FGK model: an efficient granular computing model for protein sequence motifs information discovery', *IASTED Proc. International Conference on Computational and Systems Biology (CASB)*, Dallas, PP: 56–61, 2006.

[5] Chen, B., Pellicer, S., Tai, P.C., Harrison, R. and Pan, Y. (2007a) 'Super granular SVM feature elimination (Super GSVM-FE) model for protein sequence motif information extraction', *Computational Intelligence and Bioinformatics and Computational Biology*,

CIBCB'07. IEEE Symposium on, Honolulu, Hawaiii, USA, PP: 317–322, *2007*.

[6] Chen, B., Pellicer, S., Tai, P.C., Harrison, R. and Pan, Y., 'Super granular shrink-SVM feature elimination (Super GS-SVM-FE) model for protein sequence motif information extraction', *Bioinformatics and Bioengineering, BIBE 2007. Proceedings of the 7th IEEE International Conference on*, Boston, MS, USA, PP: 379–386, *2007*.

[7] Chen, B., Pellicer, S., Tai, P.C., Harrison, R. and Pan, Y., 'Efficient super granular SVM feature elimination (Super GSVM-FE) model for protein sequence motif information extraction', *Int. J. Functional Informatics and Personalised Medicine*, Vol. 1, PP: 8–25, 2008.

[8] Chen B, He J, Pellicer S, Pan Y, "Using hybrid hierarchical K-means (HHK) clustering algorithm for protein sequence motif super-rule-tree (SRT) structure construction", International Journal of Data Mining and Bioinformatics, Volume 4 Issue 3, PP: 316-330, June 2010.

[9] He, J., Chen, B., Hu, H.J., Harrison, R., Tai, P.C., Dong, Y. and Pan, Y., 'Rule clustering and super-rule generation for transmembrane segments prediction', *IEEE Computational Systems Bioinformatics Conference Workshops (CSBW'05)*, Stanford University, California, USA, PP: 224–227, 2005.

[10] Henikoff, S. and Henikoff, J.G., 'Amino acid substitution matrices from protein blocks', *Proceedings of the National Academy of Sciences of the United States of America*, Vol. 89, PP:10915–10919, 1992.

[11] Hu, J., Ray, B.K. and Singh, M., 'Statistical methods for automated generation of service engagement staffing plans-References', *IBM Journal of Research and Development*, Vol. 51, PP: 281–293,, 2007.

[12] Jain, A.K. and Dubes, R.C., "*Algorithms for Clustering Data"*, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1988.