ISSN: 2321-7782 (Online)
Volume 3, Issue 4, April 2015
International Journal of Advance Research in Computer Science and Management Studies
Research Article / Survey Paper / Case Study
Available online at: www.ijarcsms.com

Literature Survey on Clustering Algorithms

K. Aravinthan[1]
Research Scholar,
J. J. College of Arts and Science,
Pudukkottai, (TN) India.

Dr. M. Vanitha[2]
Assistant Professor,
J. J. College of Arts and Science,
Pudukkottai, (TN) India.

Abstract: ***In this paper, we examine currently proposed clustering algorithms. The goal of this survey is to provide a comprehensive review of different clustering algorithms in data mining. We also discuss improvements to be made for future proposed clustering schemes. This paper should provide the reader with a basis for research in clustering schemes.***

I. INTRODUCTION

Clustering (see, e.g., Alpaydin, 2004; Hand et al., 2001) is a traditional problem in data analysis. Given a set of objects, the task is to divide the objects to homogeneous groups based on some criteria, typically a distance function between the objects. Cluster analysis has applications in numerous fields, and a myriad of different algorithms for various clustering problems have been developed over the past decades. The reader is referred to the surveys by Xu and Wunsch (2005) and Berkhin (2006) for a more general discussion about clustering algorithms and their applications.

This work is about *clustering a set of orders*, a problem previously studied by Murphy and Martin (2003), Busse et al. (2007), and Kamishima and Akaho (2009). Rankings of items occur naturally in various applications, such as preference surveys, decision analysis, certain voting systems, and even bioinformatics. As an example, consider the Single transferable vote system (Tideman, 1995), where a vote is an ordered subset of the candidates. By clustering such votes, the set of voters can be divided to a number of groups based on their political views. Or, in gene expression analysis it is sometimes of interest to analyze the order of genes induced by the expression levels instead of the actual numeric values (Ben-Dor et al., 2002).

II. RELATED WORK

Previous research on cluster analysis in general is too numerous to be covered here in full. Instead, we refer the readers to recent surveys by Xu and Wunsch (2005) and Berkhin (2006). For the problem of clustering orders, surprisingly little work has been done. The problem discussed in this paper is also studied by Kamishima and Akaho (2009), and earlier by Kamishima and Fujiki (2003). Murphy and Martin (2003) propose a mixture model for clustering orders. However, they only consider inputs that consist of total orders, that is, every chain in the input must order all items in M. This restriction is not made by Busse et al. (2007) who study a setting similar to ours. An important aspect of their approach is to represent a chain using the set of total orders that are compatible with the chain. This idea can also be found in the work by Critchlow (1985), and is a crucial component of a part of our work in Section 3. Recently Clemencon and Jakubowicz (2010) propose a distance function for permutations based on earth mover's distance between doubly stochastic matrices. While this framework seems quite interesting, extending it for chains seems nontrivial. The use of randomization testing (Good, 2000) in the context of data mining was first proposed by et al. (2007). Theoretical aspects of the sampling approach are discussed by Besag and Clifford (1989) and Besag and Clifford (1991).

III. CLUSTERING ALGORITHMS

We present below a survey of different clustering algorithms.

Different starting points and criteria usually lead to different taxonomies of clustering algorithms (P. Berkhin. 2001), (B. Everitt et al., 2001), (P. Hansen and B. Jaumard 1997), (A. Jain and R. Dubes 1988), (A. Jain et al., 1999), (E. Kolatch 2001). A rough but widely agreed frame is to classify clustering techniques as hierarchical clustering and partitional clustering, based on the properties of clusters generated (B. Everitt et al., 2001), (A. Jain et al., 1999). Hierarchical clustering group's data objects with a sequence of partitions, either from singleton clusters to a cluster including all individuals or vice versa, while partitional clustering directly divides data objects into some prespecified number of clusters without the hierarchical structure. We follow this frame in surveying the clustering algorithms in the literature. Beginning with the discussion on proximity measure, which is the basis for most clustering algorithms, we focus on hierarchical clustering and classical partitional clustering algorithms in Section II-B–D. Starting from part E, we introduce and analyze clustering algorithms based on a wide variety of theories and techniques, including graph theory, combinatorial search techniques, fuzzy set theory, neural networks, and kernels techniques. Compared with graph theory and fuzzy set theory, which had already been widely used in cluster analysis before the 1980s, the other techniques have been finding their applications in clustering just in the recent decades. In spite of the short history, much progress has been achieved. Note that these techniques can be used for both hierarchical and partitional clustering. Considering the more frequent requirement of tackling sequential data sets, large-scale, and high-dimensional data sets in many current applications, we review clustering algorithms for them in the following three parts. We focus particular attention on clustering algorithms applied in bioinformatics. We offer more detailed discussion on how to identify appropriate number of clusters, which is particularly important in cluster validity, in the last part of the section.

A. Identifier-Based Clustering

A unique ID is assigned to each node. Nodes know the ID of its neighbors and clusterhead is chosen following some certain rules as given below.

1. Lowest ID Cluster Algorithm (LIC)

LIC (M. Gerla and J. T. Tsai 1995) is an algorithm in which a node with the minimum *id* is chosen as a clusterhead. Thus, the *ids* of the neighbors of the clusterhead will be higher than that of the clusterhead. A node is called a gateway if it lies within the transmission range of two or more clusterheads. Gateway nodes are generally used for routing between clusters. Each node is assigned a distinct *id*. Periodically, the node broadcasts the list of nodes that it can hear (including itself) .

- A node which only hears nodes with id higher than itself is a clusterhead.
- The lowest-id node that a node hears is its clusterhead, unless the lowest-id specifically gives up its role as clusterhead (deferring to a yet lower id node).
- A node which can hear two or more clusterheads is a gateway.
- Otherwise, a node is an ordinary node.

The Lowest-ID scheme concerns only with the lowest node *ids* which are arbitrarily assigned numbers without considering any other qualifications of a node for election as a clusterhead. Since the node *ids* do not change with time, those with smaller *ids* are more likely to become clusterheads than nodes with larger *ids*. Thus, drawback of lowest ID algorithm is that certain nodes are prone to power drainage due to serving as clusterheads for longer periods of time.

2. Max-Min D-Cluster Formation Algorithm

The A.D. Amis et al. (2000) Generalizes the cluster definition to a collection of nodes that are up to d-hops away from a clusterhead. Due to the large number of nodes involved, it is desirable to let the nodes operate asynchronously. The clock synchronization overhead is avoided, providing additional processing savings. Furthermore, the number of messages sent from each node is limited to a multiple of *d* the maximum number of hops away from the nearest clusterhead, rather than *n* the number of nodes in the network. This guarantees a good controlled message complexity for the algorithm. Additionally, because *d* is an input value to the heuristic, there is control over the number of clusterheads elected or the density of clusterheads in the network.

B. Connectivity-Based clustering

1. Highest Connectivity Clustering Algorithm (HCC)

The M. Gerla and J. T. Tsai (1995) The degree of a node is computed based on its distance from others. Each node broadcasts its id to the nodes that are within its transmission range. The node with maximum number of neighbors (i.e., maximum degree) is chosen as a clusterhead. The neighbors of a clusterhead become members of that cluster and can no longer participate in the election process. Since no clusterheads are directly linked, only one clusterhead is allowed per cluster. Any two nodes in a cluster are at most twohops away since the clusterhead is directly linked to each of its neighbors in the cluster. Basically, each node either becomes a clusterhead or remains an ordinary node.

2. K-Hop Connectivity ID Clustering Algorithm (KCONID)

The G. Chen et al., (2002) Combines two clustering algorithms: the Lowest-ID and the Highest-degree heuristics. In order to select clusterheads connectivity is considered as a first criterion and lower ID as a secondary criterion. Using only node connectivity as a criterion causes numerous ties between nodes On the other hand, using only a lower ID criterion generates more clusters than necessary. The purpose is to minimize the number of clusters formed in the network and in this way obtain dominating sets of smaller sizes. Clusters in the KCONID approach are formed by a clusterhead and all nodes that are at distance at most k-hops from the clusterhead.

C. Low Cost of Maintenance Clustering

1. Least Cluster Change Algorithm (LCC)

(C.-C. Chiang *et al* 1997) LCC has a significant improvement over LIC and HCC algorithms as for as the cost of cluster maintenance is consider. Most of protocols executes the clustering procedure periodically, and re-cluster the nodes from time to time in order to satisfy some specific characteristic of clusterheads. In HCC, the clustering scheme is performed periodically to check the "local highest node degree" aspect of a clusterhead. When a clusterhead finds a member node with a higher degree, it is forced to hand over its clusterhead role. This mechanism, involves frequent reclustering. In LCC the clustering algorithm is divided into two steps: cluster formation and cluster maintenance. The cluster formation simply follows LIC, i.e. initially mobile nodes with the lowest ID in their neighborhoods are chosen as clusterheads. Re-clustering is event-driven and invoked in only two cases:

» When two clusterheads move into the reach range of each other, one gives up the clusterhead role.

» When a mobile node cannot access any clusterhead, it rebuilds the cluster structure for the network according to LIC.

D. Combined-Weight Based Clustering

1. Weighted Clustering Algorithm (WCA)

WCA (M. Chatterjee et al., 2000) selects a clusterhead according to the number of nodes it can handle, mobility, transmission power and battery power. To avoid communications overhead, this algorithm is not periodic and the clusterhead election procedure is only invoked based on node mobility and when the current dominant set is incapable to cover all the nodes. To ensure that clusterheads will not be over-loaded a pre-defined threshold is used which indicates the number of nodes each clusterhead can ideally support. WCA selects the clusterheads according to the weight value of each node.

IV. Overview Of Different Clustering Algorithms

Clustering can be considered the most important *unsupervised learning* problem; so, as every other problem of this kind, it deals with finding a structure in a collection of unlabeled data (Pavel Berkhin). Clustering is a division of data into groups of similar objects. (S.Anitha Elavarasi et al., 2011) Clustering algorithm can be divided into the following categories:

A. Hierarchical Clustering

Hierarchical clustering algorithm groups data objects to form a tree shaped structure. It can be broadly classified into agglomerative hierarchical clustering and divisive hierarchical clustering. In agglomerative approach which is also called as bottom up approach, each data points are considered to be a separate cluster and on each iteration clusters are merged based on a criteria. The merging can be done by using single link, complete link, centroid or wards method. In divisive approach all data points are considered as a single cluster and they are splited into number of clusters based on certain criteria, and this is called as top down approach (S.Anitha Elavarasi et al., 2011) . Examples for this algorithms are LEGCLUST (Santos et al., 2008), BRICH (M. Livny et al., 1996) (Balance Iterative Reducing and Clustering using Hierarchies), CURE (Cluster Using Representatives) (S. Guha et al., 1998), and Chemeleon (Jiawei Han and Micheline Kamber). Under the hierarchical clustering we have different clustering algorithms as follows.

1. Classifying Large Data Sets Using SVM with Hierarchical Clusters:

In this clustering “Classifying Large Data Sets Using SVM with Hierarchical Clusters” present a new method, Clustering-Based SVM (CB-SVM), which is specifically designed for handling very large data sets. This paper proposes a new method called CB-SVM (Clustering- Based SVM) that integrates a scalable clustering method with an SVM method and effectively runs SVMs for very large data sets. The existing SVMs are not feasible to run such data sets due to their high complexity on the data size. CB-SVM tries to generate the best SVM boundary for very large data sets given limited amount of resource based on the philosophy of hierarchical clustering where progressive deepening can be conducted when needed to find high quality boundaries for SVM. This experiments on synthetic and real data sets show that CB-SVM is very scalable for very large data sets while generating high classification accuracy (Hwanjo Yu et al.,).

2. Efficient Hierarchical Clustering of Large Data Sets Using P-trees:

Hierarchical clustering methods have attracted much attention by giving the user a maximum amount of flexibility. Rather than requiring parameter choices to be predetermined, the result represents all possible levels of granularity. In this paper a hierarchical method is introduced that is fundamentally related to partitioning methods, such as k-medoids and k-means, as well as to a density based method, namely center-defined DENCLUE. It is superior to both k-means and k-medoids in its reduction of outlier influence. Nevertheless it avoids both the time complexity of some partition-based algorithms and the storage requirements of density-based ones. An Implementation is presented that is particularly suited to spatial-, stream-, and multimedia data, using P-trees1 for efficient data storage and access (Anne Denton et al.,).

B. K-Mean Clustering Algorithm

K-means clustering is a partitioning method. ***K-means* clustering** is a method of cluster analysis which aims to partition *n* observations into *k* clusters in which each observation belongs to the cluster with the nearest mean. The k –mean algorithm The *k*-means algorithm has the following important properties:

1. It is efficient in processing large data sets.

2. It often terminates at a local optimum

3. It works only on numeric values.

4. The clusters have convex shapes

1. Parallel k/h-Means Clustering for Large Data Sets:

In this paper presented a parallel version of the *k/h*-means clustering algorithm. The algorithm is designed to be used on large distributed data sets. Even on a very simple distributed computing environment, namely a PC cluster on a 10 MBits Ethernet, we are able to achieve about 90% efficiency for a configuration up to 32 processors. These results show that parallel *k/h*-means is scalable and thus enlarges its field of application to clustering tasks where it would be the preferred algorithm, but the task's computational complexity previously made it impossible (Kilian Stoffel and Abdelkader Belkoniene).

2. A Novel K-Means Based Clustering Algorithm for High Dimensional Data Sets:

The algorithm presents a method to use both advantages of HC and K-Means by introducing equivalency and compatible relation concepts. By these two concepts we defined similarity and our space and could divide our space by a specific criterion. Many directions exist to improve and extend the proposed method. Different applications can be used and examined the framework. Text mining is an interesting arena. Based on this method data stream processing can be improved. Data type is another direction to examine this method. In this study K-Means has been used for second phase whereas we can use other clustering algorithms e.g. genetic algorithm, HC algorithm, Ant clustering (U. Boryczka 2009), Self Organizing Maps (D. Isa et al., 2009), etc. Determining number of sub spaces can be studied as important direction for the proposed method.

B. Density Based Clustering Algorithm

Density based algorithm continue to grow the given cluster as long as the density in the neighbourhood exceeds certain threshold (Jiawei Han and Micheline Kamber). This algorithm is suitable for handling noise in the dataset. The following points are enumerated as the features of this algorithm.

1. Handles clusters of arbitrary shape

2. Handle noise

3. Needs only one scan of the input dataset.

4. Needs density parameters to be initialized.

DBSCAN, DENCLUE and OPTICS (Jiawei Han and Micheline Kamber) are 0examples for this algorithm.

1. DESCRY:

A Density Based Clustering Algorithm for Very Large Data Sets This paper described a new method, named DESCRY, to identify clusters in large high dimensional data set having different size and shape. The algorithms parametric the agglomerative method used in the pre-clustering step and the similarity metrics of interest. DESCRY has a very low computational complexity, indeed it requires O(Nmd) time, for high-dimensional data sets, and O(N logm) time, for low dimensional data sets, where m can be considered a constant characteristic of the data set. Thus DESCRY scales linearly both the size and the dimensionality of

the data set (Fabrizio Angiulli et al.,). Despite its low complexity, qualitative results are very good and comparable with those obtained by state of the art clustering algorithms. Future work includes, among other topics, the investigation of similarity metrics particularly meaningful in high-dimensional spaces, exploiting summaries extracted from the regions associated to midpoints.

2. Density-Based Clustering in Spatial Databases: The Algorithm GDBSCAN and its Applications

In this paper, presented the clustering algorithm GDBSCAN generalizing the density-based algorithm DBSCAN (Ester et al., 1996) in two important ways. GDBSCAN can cluster point objects as well as spatially extended objects according to both, their spatial and their non-spatial attributes. After a review of related work, the general concept of density-connected sets and an algorithm to discover them were introduced (Ester et al., 1996). A performance evaluation, analytical as well as experimental, showed the effectiveness and efficiency of GDBSCAN on large spatial databases.

V. CONCLUSION

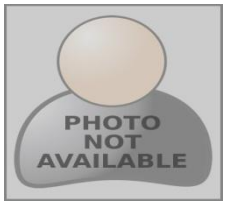
Various algorithms are used for clustering for example, k/h-means clustering algorithm is designed to be used on large distributed data sets and Hierarchical clustering algorithm groups data objects to form a tree shaped structure. This paper describes different methodologies and parameters associated with different clustering algorithms used in larger data sets. And it gives an overview of different clustering algorithms used in large data sets. Then describes about the general working behaviour, and the methodologies followed on these approaches and the parameters which used in these algorithms with large data sets. However there is still much work to be done.

References

1. A.D. Amis, R. Prakash, T.H.P Vuong, D.T. Huynh. "Max-Min DCluster Formation in Wireless Ad Hoc Networks". In proceedings of IEEE Conference on Computer Communications (INFOCOM) Vol. 1. pp. 32-41, 2000.
2. S.Anitha Elavarasi and Dr. J. Akilandeswari and Dr. B. Sathiyabhama, January 2011, "A Survey On Partition Clustering Algorithms".
3. Berkhin, "Grouping Multidimensional Data", chapter A Survey of Clustering Data Mining Techniques, pages 25–71. Springer, 2006.
4. J. Besag and P. Clifford. Sequential Monte Carlo p-values. Biometrika, 78(2):301–304, 1991.
5. P. Berkhi, "Survey of clustering data mining techniques", 2001. [Online]. Available: http://www.accrue.com/products/rp_cluster_review.pdf http://citeseer.nj.nec.com/berkhin02survey.html
6. U. Boryczka, "Finding groups in data: Cluster analysis with ants," Applied Soft Computing Journal, vol. 9, pp. 61-70,2009.
7. D. Critchlow, Metric Methods for Analyzing Partially Ranked Data, volume 34 of Lecture Notes in Statistics. Springer-Verlag, 1985.
8. M. Chatterjee, S. K. Das, and D. Turgut, “An On-Demand Weighted Clustering Algorithm (WCA) for Ad hoc Networks,” in proceedings of IEEE Globecom’00, pp. 1697–701, 2000.
9. P I Good, Permutation Tests: "A Practical Guide to Resampling Methods for Testing Hypotheses”, volume 2 of Springer series in statistics. Springer, 2000.
10. A. Gionis, H. Mannila, T. Mielik¨ainen, and P. Tsaparas. Assessing data mining results via swap randomization. ACM Transactions on Knowledge Discovery from Data, 1(3), 2007.
11. M. Gerla and J. T. Tsai, “Multiuser, Mobile, Multimedia Radio Network,” Wireless Networks, vol. 1, pp. 255–65, Oct. 1995.
12. P. Hansen and B. Jaumard, “Cluster analysis and mathematical programming", Math. Program., vol. 79, pp. 191–215, 1997.
13. D. Isa, V. P. Kallimani, and L. H. Lee, "Using the self organizing map for clustering of text documents," Expert Systems With Applications, vol. 36, pp. 9584-9591, 2009.
14. A. Jain, M. Murty, and P. Flynn, “Data clustering: A review,” ACM Computing Surveys, vol. 31, no. 3, pp. 264–323, 1999.
15. T. B. Murphy and D. Martin, Mixtures of distance-based models for ranking data. ComputationalStatistics & Data Analysis, 41:645–655, 2003.
16. N. Tideman, "The single transferable vote", Journal of Economic Perspectives, 9(1):27–38, 1995.
17. R. Xu and D. Wunsch, "Survey of Clustering Algorithms", IEEE Transactions on Neural Networks, 16(3):645–678, 2005.

AUTHOR(S) PROFILE

Aravinthan.K Pursuing Ph.D 2nd Year, Department of Computer Science in J. J. College of Arts and Science, Pudukkottai, Tamilnadu, India. I had Presented a paper on the topic of "Literature Survey On Clustering Algorithms" in International conference on Contemporary Trends in Computer Science held at J. J. College of Arts and Science. Participated in the National Workshop on "LATEX" at Annamalai University.



Dr. M. Vanitha is Assistant Professor, Department of Computer Science, J.J. College of Arts and Science, Pudukkottai. She Obtained her Bachelor's Degree in Mathematics during the year 1995 from SRC,Trichy. She did her Master's degree M.Sc(OR & CA) during the year 1997 at NIT, Trichy. Ph.D in Computer Science, from Mother Teresa University, Kodaikanal.