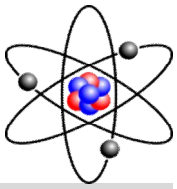
International Archive of Applied Sciences and Technology
Volume 3 [2] June 2012: 68 - 75
ISSN: 0976-4828

Website: www.soeagra.com/iaast/iaast.htm



Original Article

Survey of Recent Clustering Techniques in Data Mining

Anoop Kumar Jain[1] and Satyam Maheswari
Dept. of Computer Application
Samrat Ashok Technological Institute, Vidisha (M.P.), India
anoopjain0108@gmail.com
Dept. of Computer Application
Samrat Ashok Technological Institute, Vidisha (M.P.), India
satyam.vds@gmail.com

ABSTRACT
Cluster analysis or clustering is the task of assigning a set of objects into groups called clusters. Main task of clustering are explorative data mining, and a common technique for statistical data analysis used in many fields, including machine learning, pattern recognition, image analysis, information retrieval, and bioinformatics. Cluster analysis itself is not one specific algorithm, but the general task to be solved. It can be achieved by various algorithms that differ significantly in their notion of what constitutes a cluster and how to efficiently find them. Popular notions of clusters include groups with low distances among the cluster members, dense areas of the data space, intervals or particular statistical distributions. The appropriate clustering algorithm and parameter settings including values such as the distance function to use, a density threshold or the number of expected clusters depend on the individual data set and intended use of the results. Cluster analysis as such is not an automatic task, but an iterative process of knowledge discovery or interactive multi-objective optimization. It will often be necessary to modify preprocessing and parameters until the result achieves the desired properties. In this paper we represent a survey of clustering techniques in data mining. The clustering techniques are categorized based upon different approaches. This paper provides the major advancement in the clustering approach for data mining research using these approaches the features and categories in the surveyed work.
Keywords: Data Mining, Clustering Techniques, Performance Analysis.

INTRODUCTION

The notion of a "cluster" varies between algorithms and is one of the many decisions to take when choosing the appropriate algorithm for a particular problem. At first the terminology of a cluster seems obvious: a group of data objects. However, the clusters found by different algorithms vary significantly in their properties, and understanding
these "cluster models" is key to understanding the differences between the various algorithms.
Typical cluster models include: Connectivity models, Centroid models, Distribution models, Density models, Subspace models, Group models and Graph-based models [1-3] and [9].
A "clustering" is essentially a set of such clusters, usually containing all objects in the data set. Additionally, it may specify the relationship of the clusters to each other, for example a hierarchy of clusters embedded in each other. Data Clustering is one of the challenging mining techniques exploited in the knowledge discovery process. Clustering huge amounts of data is a difficult task since the goal is to find a suitable partition in a unsupervised way (i.e. without any prior knowledge) trying to maximize the similarity of objects belonging to the same cluster and minimizing the similarity among objects in different clusters. Many different clustering techniques have been defined in order to solve the problem from different perspective, i.e. partition based clustering, density based clustering, hierarchical methods and grid-based methods etc. In this paper we represent a survey of recent clustering approaches for data mining research.

BACKGROUND CLUSTERING TECHNIQUES

Connectivity based clustering (hierarchical clustering):

Connectivity based clustering, also known as hierarchical clustering, is based on the core idea of objects being more related to nearby objects than to objects farther away. As such, these algorithms connect "objects" to form "clusters" based on their distance. A cluster can be described largely by the maximum distance needed to connect parts of the cluster. At different distances, different clusters will form, which can be represented using a dendrogram, which explains where

the common name "hierarchical clustering" comes from: these algorithms do not provide a single partitioning of the data set, but instead provide an extensive hierarchy of clusters that merge with each other at certain distances. In a dendrogram, the y-axis marks the distance at which the clusters merge, while the objects are placed along the x-axis such that the clusters don't mix.

Connectivity based clustering is a whole family of methods that differ by the way distances are computed. Apart from the usual choice of distance functions, the user also needs to decide on the linkage criterion (since a cluster consists of multiple objects, there are multiple candidates to compute the distance to) to use. Popular choices are known as single-linkage clustering (the minimum of object distances), complete linkage clustering (the maximum of object distances) or UPGMA ("Un-weighted Pair Group Method with Arithmetic Mean", also known as average linkage clustering). Furthermore, hierarchical clustering can be agglomerative (starting with single elements and aggregating them into clusters) or divisive (starting with the complete data set and dividing it into partitions).

While these methods are fairly easy to understand, the results are not always easy to use, as they will not produce a unique partitioning of the data set, but a hierarchy the user still needs to choose appropriate clusters from. The methods are not very robust towards outliers, which will either show up as additional clusters or even cause other clusters to merge (known as "chaining phenomenon", in particular with single-linkage clustering). In the data mining community these methods are recognized as a theoretical foundation of cluster analysis, but often considered obsolete. They did however provide inspiration for many later methods such as density based clustering [2] and [6].

Centroid-based clustering:

In centroid-based clustering, clusters are represented by a central vector, which may not necessarily be a member of the data set. When the number of clusters is fixed to k, k-means clustering gives a formal definition as an optimization problem: find the k cluster centers and assign the objects to the nearest cluster center, such that the squared distances from the cluster are minimized.

The optimization problem itself is known to be NP-hard, and thus the common approach is to search only for approximate solutions. A particularly well known approximate method is Lloyd's algorithm, often actually referred to as "k-means algorithm". It does however only find a local optimum, and is commonly run multiple times with different random initializations. Variations of k-means often include such optimizations as choosing the best of multiple runs, but also restricting the centroids to members of the data set (k-medoids), choosing medians (k-medians clustering), choosing the initial centers less randomly (K-means++) or allowing a fuzzy cluster assignment.

Most k-means-type algorithms require the number of clusters k to be specified in advance, which is considered to be one of the biggest drawbacks of these algorithms. Furthermore, the algorithms prefer clusters of approximately similar size, as they will always assign an object to the nearest centroid. This often leads to incorrectly cut borders in between of clusters (which is not surprising, as the algorithm optimized cluster centers, not cluster borders).

K-means has a number of interesting theoretical properties. On one hand, it partitions the data space into a structure known as Voronoi diagram. On the other hand, it is conceptually close to nearest neighbor classification and as such popular in machine learning [3] and [8].

Distribution-based clustering:

The clustering model most closely related to statistics is based on distribution models. Clusters can then easily be defined as objects belonging most likely to the same distribution. A nice property of this approach is that this closely resembles the way artificial data sets are generated: by sampling random objects from a distribution.

While the theoretical foundation of these methods is excellent, they suffer from one key problem known as over fitting, unless constraints are put on the model complexity. A more complex model will usually always be able to explain the data better, which makes choosing the appropriate model complexity inherently difficult.

The most prominent method is known as expectation-maximization algorithm (or short: EM-clustering). Here, the data set is usually modeled with a fixed (to avoid over fitting) number of Gaussian distributions that are initialized randomly and whose parameters are iteratively optimized to fit better to the data set. This will converge to a local optimum, so multiple runs may

produce different results. In order to obtain a hard clustering, objects are often then assigned to the Gaussian distribution they most likely belong to, for soft clustering this is not necessary.
Distribution-based clustering is a semantically strong method, as it not only provides you with clusters, but also produces complex models for the clusters that can also capture correlation and dependence of attributes. However, using these algorithms puts an extra burden on the user: to choose appropriate data models to optimize, and for many real data sets, there may be no mathematical model available the algorithm is able to optimize (e.g. assuming Gaussian distributions is a rather strong assumption on the data) [5] and [9].

Density-based clustering:

In density-based clustering, clusters are defined as areas of higher density than the remainder of the data set. Objects in these sparse areas - that are required to separate clusters - are usually considered to be noise and border points.
The most popular density based clustering method is DBSCAN. In contrast to many newer methods, it features a well-defined cluster model called "density reach ability". Similar to linkage based clustering; it is based on connecting points within certain distance thresholds. However, it only connects points that satisfy a density criterion, in the original variant defined as a minimum number of other objects within this radius. A cluster consists of all density-connected objects (which can form a cluster of an arbitrary shape, in contrast to many other methods) plus all objects that are within these objects range. Another interesting property of DBSCAN is that its complexity is fairly low - it requires a linear number of range queries on the database - and that it will discover essentially the same results (it is deterministic for core and noise points, but not for border points) in each run, therefore there is no need to run it multiple times. OPTICS is a generalization of DBSCAN that removes the need to choose an appropriate value for the range parameter ε, and produces a hierarchical result related to that of linkage clustering. Density-Link-Clustering combines ideas from single-linkage clustering and OPTICS, eliminating the ε parameter entirely and offering performance improvements over OPTICS by using a tree index.
The key drawback of DBSCAN and OPTICS is that they expect some kind of density drop to detect cluster borders. Moreover they cannot detect an intrinsic cluster structure which is prevalent in majority of real life data. Variations of DBSCAN, End SCAN efficiently detect such kind of structures. On data sets with e.g. overlapping Gaussian distributions - a common use case in artificial data - the cluster borders produced by these algorithms will often look arbitrary, because the cluster density decreases continuously. On a mixture of Gaussians data set, they will almost every time be outperformed by methods such as EM clustering, which are able to precisely model this kind of data [4] and [7] and [10].

SURVEY OF CLUSTERING TECHNIQUES

Particle Swarm Optimization Based Hierarchical Agglomerative Clustering:

Shafiq Alam, Gillian Dobbie, Patricia Riddle, M. Asif Naeem et. al. proposed a novel clustering algorithm called Hierarchical Particle Swarm Optimization (HPSO) data clustering. The proposed algorithm exploits the swarm intelligence of cooperating agents in a decentralized environment. The experimental results were compared with benchmark clustering techniques, which include K-means, PSO clustering, Hierarchical Agglomerative clustering (HAC) and Density-Based Spatial Clustering of Applications with Noise (DBSCAN). The results are evidence of the effectiveness of Swarm based clustering and the capability to perform clustering in a hierarchical agglomerative manner.
Optimization in the process of knowledge discovery and data mining has emerged as an important technique for KDD. The involvement of optimization in the process has been proved to significantly improve efficiency, performance and quality of output. Data clustering is one of the comprehensive KDD techniques, which is gaining more and more importance in optimization research. In the recent past various optimization techniques were used to improve one or another aspect of clustering analysis to achieve optimal results. Swarm intelligence is one of such optimization techniques.
The approach to tackling the problem of hierarchical agglomerative clustering through hierarchical particle swarm optimization based clustering is inspired by the collective intelligent behavior of swarms. HPSO-clustering has the properties of both partitioned based data clustering and

hierarchical data clustering. Experimental results verify the performance of HPSO against PSO, traditional hierarchical agglomerative clustering and K-means clustering.

The issues with optimization based clustering that need to be addressed are optimization of the parameters and generalization of the techniques for future improvement [1].

Applying Clustering to Data Analysis of Physical Healthy Standard:

Lan Yu et. al. In this paper, we use one of the data mining algorithms (clustering) to analyze PHS test data with the help of SQL Server. In the experiments, the scores of vital capacity, grip strength, standing long jump and step test of a student are used for input attributes, and total score of the student is used for prediction attribute. The purposes of experiments are to discover how to construct a training set and how to set parameters of Microsoft clustering algorithm. Some valuable conclusions are achieved.

The ranking distribution of a good training set should be close to that of the prediction set. In the experiment, training set of case 1 has the same ranking distribution as that of the prediction set and training set of case 2 has a similar ranking distribution as that of the prediction set. Therefore, the accuracies of both cases are relatively high. A good training set should have sufficient records, but a huge number of records could degrade the performance of clustering. That is the reason why the accuracy of case 2 is

better than that of case 1. And using all data to compose a training set as case 1 of this experiment is not practical. When the number of the training set is determinate, the more records a ranking has, the higher accuracy of this ranking obtained. For example, the number of excellent is very large comparative to the other two cases, so the accuracy of this ranking is higher. Comparing to the other rankings, the excellent ranking is harder to predict.

The K-means algorithm is not sufficient suitable for the test data of PHS [2].

Fast Co-clustering Using Matrix Decomposition:

Yun Ling et. al. utilized correspondence analysis algorithm to process matrix decomposition and then make use of Bayesian approach for co-clustering. They find that utilizing the two methods synthetically is very significative to solve actual problems. Experiments on synthetic and real world data demonstrate the efficiency and effectiveness of the algorithm. They proposed a CA method for fast co-clustering on large data. The learned knowledge is useful when synthesizing these two methods. Because of the small size of the approximation matrices, it has runtime complexity equal to orders of magnitude faster than the runtime complexity of the previous co-clustering algorithms. Due to its low complexity and simple implementation the work presented will make a broad application.

This technique need more practical to solve sufficiently co-clustering problems. The Bayesian co-clustering is based on matrix decomposition that a given complex problems can be more easily [3].

Augmenting Rapid Clustering Method for Social Network Analysis:

J. Prabhu and M. Sudharshan et. al. propose an innovative clustering technique called the Rapid Clustering Method (RCM), which uses Subtractive Clustering combined with Fuzzy CMeans clustering along with a histogram sampling technique to provide quick and effective results for large sized datasets. Rapid Clustering Method can be used to cluster the dataset and analyze the characteristics in a social network. It can also be used to enhance the cross-selling practices using quantitative association rule mining.

The Rapid Clustering Method (RCM) consists of three different parts. First the dataset is sampled to obtain a smaller dataset using histogram sampling technique, then SCM-FCM is applied to the sampled dataset and the cluster centers are obtained. These cluster centers are used in SCM implementation and finally FCM is applied to the larger dataset.

The purpose of reducing the dataset is that the order of execution of SCM is n^2, thus larger the dataset more time it consumes to find the cluster centers. The histogram sampling technique is applied so as to maintain the density values similar to that of the original dataset so that they obtain the same cluster centers for the sample dataset.

The application of RCM in social network analysis comes in the form of clustering the modularity values of nodes in two different networks and identify whether the nodes are over lapping or not. When the probabilistic analysis is made for clustering, the nodes are said to overlap if the probability pair is like (0.6, 0.4) for the two communities. In other words, if the values are one sided say (0.9, 0.1) the nodes are not allowed to overlap. This kind of analysis is very important

when a subscriber is to be labeled based on the communities available in the network. RCM can also be used at regular intervals to identify the changes occurring in community structures with respect to time. Identifying change in community structure is a very important task in social network analysis. With the help of RCM they can obtain rapid results and identify structural changes much more quickly. They considered the problem of clustering data over time and proposed a rapid clustering technique so that it can be used to generate quicker results for social network analysis.

The Rapid Clustering is not used only to obtain two clusters, it need to applicable for two cluster environment in future work [4].

Performance Issues in Parallelizing Data-Intensive Applications on a Multi-core Cluster:

Vignesh T. Ravi and Gagan Agrawal et. al. described and evaluated various shared-memory parallelization techniques developed in our run-time system on a cluster of multi-cores, and reported on a detailed performance study to understand why certain parallelization techniques outperform other techniques for a particular application. The experimental study has been conducted using three popular data mining algorithms, which are K-Means clustering, E-M clustering, and Apriori association mining. They evaluated three shared-memory techniques developed in FREERIDE framework, along with use of MPI within shared memory as another possible technique. The shared-memory techniques evaluate are Full-replication, where the reduction object is replicated, Cache-sensitive locking, where one lock is used for all reduction elements within one cache-line, and Optimized-full locking, where each reduction element has a lock associated at the next memory address. The summary of the results are as follows. Both Full-replication and Cache sensitive locking can outperform each other based on the nature of the application. MPI competes well with best of these two schemes when processes are run on a small number of cores. However, when it used a larger number of threads on each node, it has communication overheads. Thus, specialized shared memory techniques are clearly needed to obtain high performance on multi-core machines.

They focused on the fraction of instructions for each application that update reduction object, the time spent on merging reduction object, and the cache performance of reduction object and the entire application as the number of threads is increased. Analysis of these factors and the performance obtained from the different techniques show that the most important trade-off is the one between the memory needs of the application, and the frequency of updating reduction elements. E-M is a memory-intensive application and therefore, it does not obtain good performance from full replication, and has the lowest scalability. Apriori, on the other hand, has the highest fraction of instructions that update reduction object, and as a result, locking techniques have a very high overhead. In addition, our experiments also show that FREERIDE's high-level API can be used to achieve good scalability on cluster of multi-core machines.

This analysis should be possible to design a module that can predict which parallelization technique would give the best performance for a given application in future improvement [5].

Scalability of Efficient Parallel K-Means:

David Pettinger and Giuseppe Di Fatta et. al. provided a parallel formulation for the KD-Tree based K-Means algorithm and address its load balancing issues. Methods for improving the efficiency of K-Means have been largely explored in two main directions. The amount of computation can be significantly reduced by adopting a more efficient data structure, notably a multi-dimensional binary search tree (KD-Tree) to store either centroids or data points. A second direction is parallel processing, where data and computation loads are distributed over many processing nodes. However, little work has been done to provide a parallel formulation of the efficient sequential techniques based on KD-Trees. Such approaches are expected to have an irregular distribution of computation load and can suffer from load imbalance. This issue has so far limited the adoption of these efficient K-Means techniques in parallel computational environments.

They generated an artificial data set with 500000 patterns in a 20 dimensional space with a mixed Gaussian distribution as described in the following.

First it generated 50 pattern prototypes in the multi-dimensional space. This corresponds to the number of clusters K. For each cluster generated 10000 patterns with a Gaussian distribution around the prototype and with a random standard deviation in the range [0.0, 0.1]. In order to create a more realistically skewed data distribution, we did not generate the prototypes uniformly

in the multi-dimensional space. They distributed 25 prototypes uniformly in the whole domain and 25 prototypes were restricted to a sub-domain.
This generated a higher density of prototypes in the sub-domain. The skewed distribution of data patterns in the domain emphasizes the load imbalance problem. The parameters were chosen in order to generate a dataset which contains some well separated clusters and some not well separated clusters.
They presented a parallel formulation of the K-Means algorithm based on an efficient data structure, namely multidimensional binary search trees. While the sequential algorithms benefits from the additional complexity, the same is not true in general for the parallel approaches. The experimental analysis shows that it is convenient to adopt the most efficient techniques for small and medium parallel environments (up to 64 computing elements). The cost of load imbalance still makes the adoption of these techniques unsuitable for large scale systems, where the simple parallel implementation of the K-Means algorithm will always provide a perfect load balance. However, this is valid only for dedicated homogeneous environments. It intend to test dynamic load balancing policies which could make the efficient techniques suitable also for large scale and heterogeneous environments.
This research is needed more optimization for the suitable communications where high network latency would make the communication cost dominate the computation [6].

A New Distributed Clustering Algorithm Based on K-means Algorithm:

Maryam hajiee et. al. envision a distributed clustering algorithm which is scalable and provides cooperation while preserving a high degree of independency for each site. In the proposed model, the sites cooperate by representing a set of statistical data while saving a great degree of data security and site independency. This model can be classified into four steps:

- Independent clustering of the data in the sites
- Relation and regulation of the sites with the central site representing their statistical data
- Global clustering in the central site and sending the results to the other sites.
- Updating the cluster centers in the other sites.

The proposed algorithm doesn't need to preprocess or distribute data as the data are intrinsically distributed in different sites. Then the central site is in charge of the regulation of the sites and global classification of the clusters. This model, On the other hand, doesn't transfer data between the sites rather a site gets wind of only the result of the others. The clustering with proposed model is done in two levels of local and global.
Distributed clustering is one of recent data mining models for clustering huge and distributed data sets. In this paper a new distributed clustering algorithm is proposed and its results on real data sets are analyzed. Clustering using the proposed model is performed in two levels of local and global.
Communication cost and needed bandwidth in comparison with other models is quite low moreover the privacy of local sites is highly preserved which makes it suitable for WAN networks [7].

K-Means Divide and Conquer Clustering:

Madjid Khalilian et. al. proposed a method which uses divide and conquer technique to improve the performance of the K-Means clustering method. Most clustering techniques ignore the fact about the different size or levels where in most cases, clustering is more concern with grouping similar objects or samples together ignoring the fact that even though they are similar, they might be of different levels.
This method can be more efficient and accurate than a single one pass clustering. Some hypothesis has been considered:
H1 - Proposed method would be able to group samples of similar size and find similarity among them.
H2 - Proposed method is faster than single step clustering due to use of divide and conquers technique.
H3 - Proposed method is more accurate than a single step clustering.
H4 – Proposed method would allow Euclidean distance to be used in high dimensional data.
In this study they assume that the space is orthogonal and dimensions for all objects are the same and finally it used ordinal data type because of the application. Using two steps clustering in high

dimensional data sets with considering size of objects helps us to improve accuracy and efficiency of original K-Means clustering. When objects are clustered base on their size, in fact, they used subspaces for clustering. It causes achieving more accurate and efficient results. For this purpose we should consider orthogonal space which means that there should be no correlation among attributes of objects and size of dimension should be equal in all objects.
For future improvement the investigate applying this method in different domain e.g. text mining. Also effects of different parameters in clustering for example K and number of dimensions should be studied [8].

CONCLUSION AND FUTURE WORKS

Data clustering is a common technique for statistical data analysis, which is used in many fields, including machine learning, data mining, pattern recognition, image analysis and bioinformatics. Clustering is the classification of similar objects into different groups, or more precisely, the partitioning of a data set into subsets (clusters), so that the data in each subset (ideally) share some common trait - often proximity according to some defined distance measure. Data clustering algorithms can be hierarchical or partitioned. Hierarchical algorithms find successive clusters using previously established clusters, whereas partition algorithms determine all clusters at once. Hierarchical algorithms can be agglomerative ("bottom-up") or divisive ("top-down"). Agglomerative algorithms begin with each element as a separate cluster and merge them into successively larger clusters. Divisive algorithms begin with the whole set and proceed to divide it into successively smaller clusters. During the survey, we also find some points that can be further improvement in the future using advanced clustering technique to achieve more efficient accuracy in result and reduce the time taken for data and/ or information retrieval from large data set.

REFERENCES

1. Shafiq Alam, Gillian Dobbie, Patricia Riddle, M. Asif Naeem,(2010). “Particle Swarm Optimization Based Hierarchical Agglomerative Clustering”. IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, pp. 64-68.
2. Lan Yu, “Applying Clustering to Data Analysis of Physical Healthy Standard”, 2010 Seventh International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2010), pp. 2766-2768.
3. Yun Ling and Hangzhou, “Fast Co-clustering Using Matrix Decomposition”, IEEE (2009). Asia-Pacific Conference on Information Processing, pp. 201-204.
4. J. Prabhu and M. Sudharshan and M. Saravanan and G.Prasad,(2010). Augmenting Rapid Clustering Method for Social Network Analysis”, International Conference on Advances in Social Networks Analysis and Mining, pp. 407-408.
5. Vignesh T. Ravi and Gagan Agrawal, “Performance Issues in Parallelizing Data-Intensive Applications on a Multi-core Cluster”, 9th IEEE/ACM International Symposium on Cluster Computing and the Grid, pp. 308-315.
6. David Pettinger and Giuseppe Di Fatta,(2009). “Scalability of Efficient Parallel K-Means”, IEEE e-Science Workshops, pp. 96-101.
7. Maryam hajiee,(2010). “A New Distributed Clustering Algorithm Based on K-means Algorithm”,3rd International Conference on Advanced Computer Theory and Engineering (1CACTE), pp. 408-411 (V2).
8. Madjid Khalilian, Farsad Zamani Boroujeni, Norwati Mustapha, Md. Nasir Sulaiman,(2009). “K-Means Divide and Conquer Clustering”, IEEE 2009, International Conference on Computer and Automation Engineering, pp. 306-309.
9. F. Yang, T. Sun, C. Zhang, (2009). An efficient hybrid data clustering method based on K-harmonic means, and Particle Swarm Optimization, Expert Systems with Applications, pp. 9847–9852.
10. Y.-T. Kao, E. Zahara, I.-W. Kao, (2008). A hybridized approach to data clustering, Expert Systems with Applications, pp. 1754-1762.