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

Evolutionary Clustering Algorithm in Data Mining

Ekta K. Chainani[1]
Computer Science and Engineering
H.V.P.M's C.O.E.T
Amravati, India

Rajeshri R. Shelke[2]
Computer Science and Engineering
H.V.P.M's C.O.E.T
Amravati, India

Abstract: With the huge amount of data being generated in the world every day, at a rate far higher than by which it can be analyzed by human comprehension alone, data mining becomes an extremely important task for extracting as much useful information from this data as possible. The standard data mining techniques are satisfactory to a certain extent but they are constrained by certain limitations, and it is for these cases that evolutionary approaches are both more capable and more efficient. In this paper we present the use of nature inspired evolutionary techniques for clustering in data mining augmented with various clustering algorithms for human interaction to handle situations for which concept definitions are abstract and hard to define, hence not quantifiable in an absolute sense. Finally, we propose some ideas for these techniques for future implementations.

Keywords: data mining, evolutionary algorithms, clustering, data mining tasks, clustering algorithms, partional, hierarchical

I. INTRODUCTION

In recent years, the massive growth in the amount of stored data has increased the demand for effective data mining methods to discover the hidden knowledge and patterns in these data sets. Data mining means to "mine" or extract relevant information from any available data of concern to the user. Data mining is not a new technique but has been around for centuries and has been used for problems like regression analysis, or knowledge discovery from records of various types. Clustering is a process which partitions a given data set into homogeneous groups based on given features such that similar objects are kept in a group whereas dissimilar objects are in different groups. It is the most important unsupervised learning problem. It deals with finding structure in a collection of unlabeled data. For better understanding please refer to Fig 1.

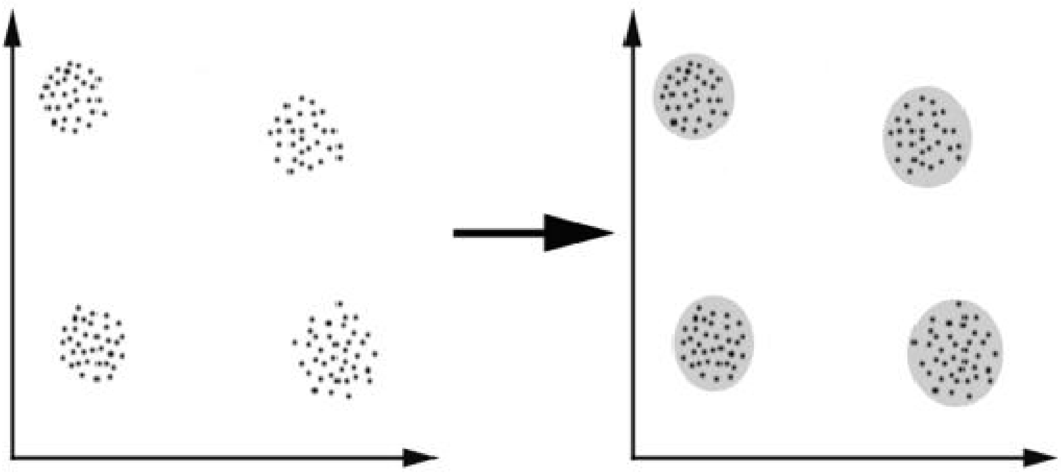

Fig 1: showing four clusters formed from the set of unlabeled data

Clustering is often confused with classification, but there is some difference between the two. In classification the objects are assigned to pre defined classes, whereas in clustering the classes are also to be defined. Precisely, Data Clustering is a technique in which, the information that is logically similar is physically stored together. In order to increase the efficiency in the database systems the numbers of disk accesses are to be minimized. In clustering the objects of similar properties are placed in one class of objects and a single access to the disk makes the entire class available.

Clustering algorithms are used to organize data, categorize data, for data compression and model construction, for detection of outliers etc. Common approach for all clustering techniques is to find clusters centre that will represent each cluster. Cluster centre will represent with input vector can tell which cluster this vector belong to by measuring a similarity metric between input vector and all cluster centre and determining which cluster is nearest or most similar one [1]. Cluster analysis can be used as a standalone data mining tool to gain insight into the data distribution, or as a preprocessing step for other data mining algorithms operating on the detected clusters. Many clustering algorithms have been developed and are categorized from several aspects such as partitioning methods, hierarchical methods, density-based methods, and grid-based methods. Further data set can be numeric or categorical. Inherent geometric properties of numeric data can be exploited to naturally define distance function between data points. Whereas categorical data can be derived from either quantitative or qualitative data where observations are directly observed from counts.

This paper begins by describing some concepts in data mining and general evolutionary algorithms by giving relevant concepts and descriptions. In the later sections we discuss some of clustering algorithm and the areas where these are implemented and lastly we give a few ideas of where these techniques may be implemented in the future.

II. An Overview of Evolutionary Algorithms

An Evolutionary Algorithm (EA) is essentially an algorithm inspired by the principle of natural selection and natural genetics. The basic idea is simple. In nature individuals are continuously evolving, getting more and more adapted to the environment. In EAs each \individual" corresponds to a candidate solution to the target problem, which could be considered a very simple\environment". Each individual is evaluated by a tenses function, which measures the quality of the candidate solution represented by the individual. At each generation (iteration), the best individuals (candidate solutions) have a higher probability of being selected for reproduction. The selected individuals undergo operations inspired by natural genetics, such as crossover (where part of the genetic material of two individuals are swapped) and mutation (where part of the generic material of an individual is replaced by randomly- generated genetic material), producing new upspring which will replace the parents, creating a new generation of individuals. This process is iteratively repeated until a stopping criterion is satisfied, such as until a number of generations has been performed or until a satisfactory solution has been found.

Evolutionary Algorithms (EAs) are stochastic search algorithms inspired by the process of neo-Darwinian evolution. The motivation for applying EAs to data mining is that they are robust, adaptive search techniques that perform a global search in the solution space. This chapter first presents a brief overview of EAs, focusing mainly on two kinds of EAs, viz. Genetic Algorithms (GAs) and Genetic Programming (GP). Then the chapter reviews the main concepts and principles used by EAs designed for solving several data mining tasks, namely: discovery of classification rules, clustering, attribute selection and attribute construction. Finally, it discusses Multi-Objective EAs, based on the concept of Pareto dominance, and their use in several data mining tasks.

III. Evolutionary Algorithms And Data Mining

Evolutionary algorithms have several features that make them attractive for the data mining process (Freitas, 2003; Vafaie & Jong, 1994). They are a domain independent technique, which makes them ideal for applications where domain knowledge is difficult to provide. They have the ability to explore large search spaces finding consistently good solutions. In addition, they are relatively insensitive to noise, and can manage attribute interaction better than the conventional data mining techniques. Therefore, several works have been done, in recent years, to develop new techniques for data mining using evolutionary algorithms. These attempts used evolutionary algorithms for different tasks of data mining such as feature extraction, feature selection, classification, and clustering (Cant´u-Paz & Kamath, 2001).

The main role of evolutionary algorithms in most of these approaches is optimization. They are used to improve the robustness and accuracy of some of the traditional data mining techniques. Different types of evolutionary algorithms have been developed over the years such as genetic algorithms, clustering algorithms, evolution strategies, evolutionary programming, evolution strategies, differential evolution, cultural evolution algorithms and co-evolutionary algorithms (Engelbrecht, 2007). Some of these types that are used in data mining are genetic algorithms, clustering algorithm and co-evolutionary algorithms.

IV. Categorization of Clustering Techniques And Previous Work

a) *BASIC CLUSTERING TECHNIQUES*

We distinguish two types of clustering techniques: *Partitional* and *Hierarchical*. Their definitions are as follows [HK01]:

1. Partitional:

Given a database of n objects, a partitional clustering algorithm constructs k partitions of the data, where each cluster optimizes a clustering criterion, such as the minimization of the *sum of squared distance from the mean* within each cluster. One of the issues with such algorithms is their high complexity, as some of them exhaustively enumerate all possible groupings and try to find the global optimum. Even for a small number of objects, the number of partitions is huge. That's why, common solutions start with an initial, usually random, partition and proceed with its refinement. A better practice would be to run the partitional algorithm for different sets of initial _ points (considered as representatives) and investigate whether all solutions lead to the same final partition. Partitional Clustering algorithms try to locally improve a certain criterion. First, they compute the values of the similarity or distance, they order the results, and pick the one that optimizes the criterion. Hence, the majority of them could be considered as greedy-like algorithms.

2. Hierarchical:

Hierarchical algorithms create a hierarchical decomposition of the objects. They are either *agglomerative* (*bottom-up*) or *divisive* (*top-down*):

A. *Agglomerative* algorithms start with each object being a separate cluster itself, and successively merge groups according to a distance measure. The clustering may stop when all objects are in a single group or at any other point the user wants. These methods generally follow a greedy-like bottom-up merging.

B. *Divisive* algorithms follow the opposite strategy. They start with one group of all objects and successively split groups into smaller ones, until each object falls in one cluster, or as desired. Divisive approaches divide the data objects in disjoint groups at every step, and follow the same pattern until all objects fall into a separate cluster. This is similar to the approach followed by divide-and-conquer algorithms. Most of the times, both approaches suffer from the fact that once a merge or a split is committed, it cannot be undone or refined. Figure 1(a) gives an example of two divisive algorithms performed in the same data set, with different initial parameters. A ''+'' sign denotes the centre of clusters, which in this case is defined as the mean of the values of a particular cluster. At the same time, Figure 1(b) depicts the *dendrogram* produced by either a divisive or agglomerative clustering algorithm.

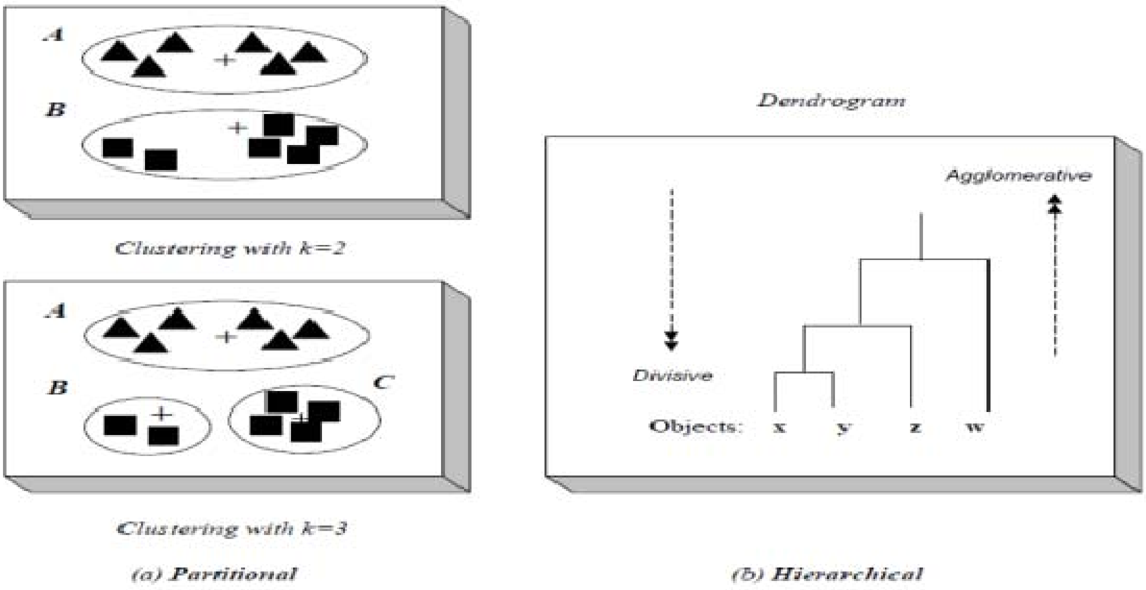

Fig 2: Examples of the classic clustering algorithms, where k is the number of clusters

Partitional and hierarchical methods can be integrated. This would mean that a result given by a hierarchical method can be improved via a partitional step, which refines the result via iterative relocation of points. Other classes of clustering algorithms are given in the next subsection.

b) Data Mining Clustering Techniques

Apart from the two main categories of partitional and hierarchical clustering algorithms, many other methods have emerged in cluster analysis, and are mainly focused on specific problems or specific data sets available. These methods include [HK01]:

1. Density-Based Clustering:

These algorithms group objects according to specific density objective functions. Density is usually defined as the number of objects in a particular neighborhood of a data objects. In these approaches a given cluster continues growing as long as the number of objects in the neighborhood exceeds some parameter. This is considered to be different from the idea in partitional algorithms that use iterative relocation of points given a certain number of clusters.

2. Grid-Based Clustering:

The main focus of these algorithms is spatial data, *i.e.*, data that model the geometric structure of objects in space, their relationships, properties and operations. The objective of these algorithms is to quantize the data set into a number of cells and then work with objects belonging to these cells. They do not relocate points but rather build several hierarchical levels of groups of objects. In this sense, they are closer to hierarchical algorithms but the merging of grids, and consequently clusters, does not depend on a distance measure but it is decided by a predefined parameter.

3. Model-Based Clustering:

These algorithms find good approximations of model parameters that best fit the data. They can be either partitional or hierarchical, depending on the structure or model they hypothesize about the data set and the way they refine this model to identify partitioning. They are closer to density-based algorithms, in that they grow particular clusters so that the preconceived model is improved. However, they sometimes start with a fixed number of clusters and they do not use the same concept of density.

4. Categorical Data Clustering:

These algorithms are specifically developed for data where Euclidean, or other numerical-oriented, distance measures cannot be applied. In the literature, we find approaches close to both partitional and hierarchical methods. For each category, there exists a plethora of sub-categories, *e.g.*, density-based clustering oriented towards geographical data [SEKX98], and algorithms for finding clusters. An exception to this is the class of categorical data approaches. Visualization of such data is not straightforward and there is no inherent geometrical structure in them, hence the approaches that have appeared in the literature mainly use concepts carried by the data, such as co-occurrences in tuples. On the other hand, categorical data sets are in abundance. Moreover, there are data sets with mixture of attribute types, such as the United States Census data set (see http://www.census.gov/) and data sets used in schema discovery [**?**]. As will be discussed, current clustering algorithms focus on situations in which all attributes of an object are of a single type. We believe that cluster analysis of categorical and mixed type data sets is an intriguing problem in data mining. But what makes a clustering algorithm efficient and effective? The answer is not clear. A specific method can perform well on one data set, but very poorly on another, depending on the size and dimensionality of the data as well as the objective function and structures used.

c) Partitional Algorithms

This family of clustering algorithms includes the first ones that appeared in the Data Mining Community. The most commonly used are *k-means*, [JD88, KR90], *PAM (Partitioning Around Medoids)*, [KR90], *CLARA (Clustering LARge Applications)*, [KR90] and *CLARANS (Clustering LARge ApplicatioNS)*, [NH94]. The goal in *k-means* is to produce k clusters from a set of m objects, so that the *squared-error* objective functions:

$$E = \sum_{i=1}^{k} \Sigma_{p \in C_i} |p - m_i|^2$$

is minimized. In the above expression, Ci are the clusters, p is a point in a cluster Ci , and mi the mean of Cluster Ci. The mean of a cluster is given by a vector, which contains, for each attribute, the mean values of the data objects in this cluster and. Input parameter is the number of clusters, k, and as an output the algorithm returns the centers, or means, of every cluster Ci , most of the times excluding the cluster identities of individual points. The distance measure usually employed is the Euclidean distance. Both for the optimization criterion and the proximity index, there are no restrictions, and they can be specified according to the application or the user's preference. The algorithm is as follows:

1. Select _ objects as initial centers;

2. Assign each data object to the closest center;

3. Recalculate the centers of each cluster;

4. Repeat steps _ and □ until centers do not change;

The algorithm is relatively scalable, since its complexity is, *O (I kn)* 1, where *I* denotes the number of iterations, and usually k << n.

d) Hierarchical Algorithms

As we already mentioned, standard hierarchical approaches suffer from high computational complexity, namely *O* (n^2). Some approaches have been proposed to improve this performance and one of the first ones is *BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies)* [ZRL96]. It is based on the idea that we do not need to keep whole tuples or whole clusters in main memory, but instead, their sufficient statistics. For each cluster, *BIRCH* stores only the triple (n,L.S,S.S) where n is the number of data objects in the cluster, L.S is the linear sum of the attribute values of the objects in the cluster and S.S is the sum of squares of the attribute values of the objects in the cluster. These triples are called *Cluster Features (CF)* and kept in a tree called *CF-tree*. In the paper by Zhang et al. [ZRL96] it is proved how standard statistical quantities, such as distance measures, can be derived from the CF's.

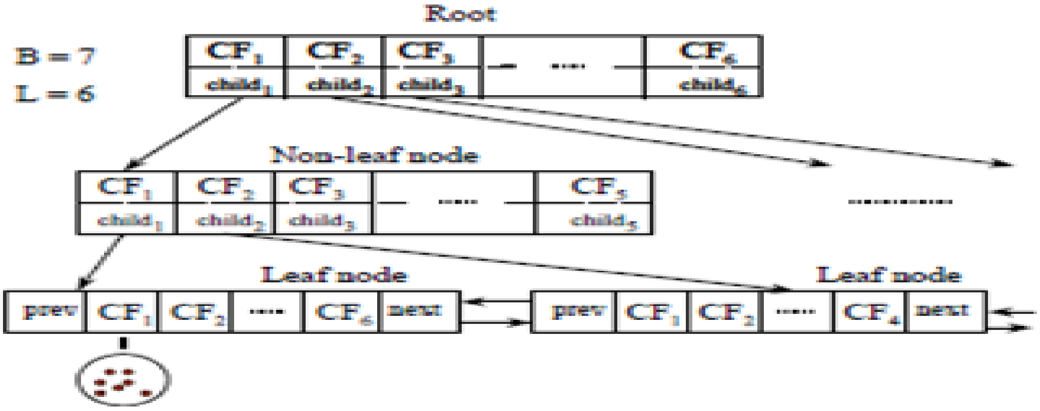

Fig 3. A CF-Tree used by the BIRCH

CF-trees are characterized by two parameters. These are the *Branching Factor, B* and the *Threshold, T*, the former being the maximum number of children for a non-leaf node and the latter the maximum distance between any pair of points, *i.e.* the *diameter*, in sub-clusters stored at leaf nodes. An example of a CF-tree is given in Figure 3. All nodes store CF's: non-leaf ones store the sums of the CF's of their children, while leaf nodes the CF's of the data objects themselves. *BIRCH* works as follows:

1. The data objects are loaded one by one and the initial CF-tree is constructed: an object is inserted into the closest leaf entry, *i.e.* sub-cluster. If the diameter of this sub-cluster becomes larger than T, the leaf node, and possible others, are split. When the object is properly inserted in a leaf node, all nodes towards the root of the tree are updated with necessary information.

2. If the CF-tree of stage 1 does not fit into memory, build a smaller CF-tree: the size of a CF-tree is controlled by parameter _ and thus choosing a larger value for it will merge some sub-clusters making the tree smaller. Zhang et al. show how this stage does not require to start reading the data from the beginning and guarantees the creation of a smaller tree.

3. Perform clustering: leaf nodes of the CF-tree hold sub-cluster statistics; in this stage *BIRCH* uses these statistics to apply some clustering technique, *e.g. k-means*, and produce an initial clustering.

4. Redistribute the data objects using centroids of clusters discovered in step 3 this is an optional stage which requires an additional scan of the data set and re-assigns the objects to their closest centroids. Optionally, this phase also includes the labeling of the initial data and discarding of outliers.

V. Conclusion Applications Of Data Mining Using IEA

In this section we examine some areas where data mining with interactive evolutionary algorithms IEA techniques has been successfully applied. The first approach detailed is very general in terms that it can be used to classify any text based data and hence is not limited to any specific discipline. The approach requires textual data in the form of reports, which can be just normal text files corresponding to the database for which the knowledge needs to be extracted.

1. Extracting Knowledge from a Text Database
2. Extracting Marketing Rules from User Data
3. Fraud Detection Using Data Mining and IEA Techniques

VI. Conclusion

Clustering lies at the heart of data analysis and data mining applications. The ability to discover highly correlated regions of objects when their number becomes very large is highly desirable, as data sets grow and their properties and data interrelationships change. At the same time, it is notable that any clustering "is a division of the objects into groups based on a set of rules – it is neither true nor false" .In this paper we described the process of clustering from the data mining point of view. We gave the properties of a "good" clustering technique and the methods used to find meaningful partitioning. At the same time, we concluded that research has emphasized numerical data sets, and the intricacies of working with large categorical databases is left to a small number of alternative techniques

ACKNOWLEDGMENT

My thanks to the Guide, Prof. R.R.Shelke and Principal Dr.A.B.Marathe, who provided me constructive and positive feedback during the preparation of this paper.

References

1. [ABKS96] Mihael Ankerst, Markus M. Breunig, Hans-Peter Kriegel, and J ¨org Sander. OPTICS: Ordering Points To Identify the Clustering Structure. In Proceedings of the International Conference on Managementof Data, (SIGMOD), volume 28(2) of SIGMOD Record, pages 49–60, Philadelphia, PA, USA, 1–3 June 1996. ACM Press.
2. [AGGR98] Rakesh Agrawal, Johannes Gehrke, Dimitrios Gunopulos, and Prabhakar Raghavan. Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications. In Proceedings of the International Conference on Management of Data, (SIGMOD), volume 27(2) of SIGMOD Record, pages 94–105, Seattle,WA, USA, 1–4 June 1998. ACM Press.
3. [And73] Michael R. Anderberg. Cluster analysis for applications. Academic Press, 1973.
4. [BFR98] Paul S. Bradley, Usama Fayyad, and Cory Reina. Scaling Clustering Algorithms to Large Databases. In Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining, (KDD), pages 9–15, New York, NY, USA, 27–31 August 1998. AAAI Press.
5. [BFR99] Paul S. Bradley, Usama Fayyad, and Cory Reina. ScalingEM(Expectation-Maximization) Clustering to Large Databases. Technical Report MSR-TR-98-35, Microsoft Research, Redmond, WA, USA, October 1999.
6. [EKSX96] Martin Ester, Hans-Peter Kriegel, J ¨oerg Sander, and Xiaowei Xu. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining, (KDD), pages 226–231, Portland, OR, USA, 2–4 August 1996. AAAI Press.
7. [Eve93] Brian S. Everitt. Cluster Analysis. Edward Arnold, 1993.
8. [GGR99] Venkatesh Ganti, Johannes Gehrke, and Raghu Ramakrishnan. CACTUS: Clustering Categorical Data Using Summaries. In Proceedings of the 5th International Conference on Knowledge Discovery and Data Mining, (KDD), pages 73–83, San Diego, CA, USA, 15–18 August 1999. ACM Press.
9. [Gil58] E.W. Gilbert. Pioneer Maps of Health and Disease in England. Geographical Journal, 124: 172– 183, 1958.
10. [GKR98] David Gibson, Jon M. Kleinberg, and Prabhakar Raghavan. Clustering Categorical Data: An Approach Based on Dynamical Systems. In Proceedings of the 24th International Conference on Very Large Data Bases, (VLDB), pages 311–322,NewYork, NY, USA, 24–27 August 1998. Morgan Kaufmann.

AUTHOR(S) PROFILE

Ekta K. Chainani , received the B.E.degree in Computer Science and Engineering from H.V.P.M's College Of Engineering And Technology, Amravati in 2014. She is currently pursuing Master's Degree in Computer Science and Engineering from H.V.P.M's College of Engineering And Technology, Amravati.

Prof. Rajeshri R. Shelke, received the B.E. and M.E degree in Computer Science. Her field of specialization is Data Mining. She is currently working as Associate Professor at H.V.P.M's college of Engineering and Technology, Amravati.