INTERNATIONAL JOURNAL OF ENGINEERING AND MANAGEMENT SCIENCES

www.scienceandnature.org

ANALYSIS OF CLUSTERING TECHNIQUE FOR CRM

Chopra Manu,
Research Scholar, Singhania University, Pacheri Bari, Jhunjhunu, Rajasthan

ABSTRACT
The goal of this analysis is to provide a comprehensive review of different clustering techniques in data mining. Clustering is a division of data into groups of similar objects. Each group, called cluster, consists of objects that are similar between themselves and dissimilar to objects of other groups. Representing data by fewer clusters necessarily loses certain fine details, but achieves simplification. It represents many data objects by few clusters, and hence, it models data by its clusters. Data modeling puts clustering in a historical perspective rooted in mathematics, statistics, and numerical analysis. From a machine learning perspective clusters correspond to hidden patterns, the search for clusters is unsupervised learning, and the resulting system represents a data concept. Therefore, clustering is unsupervised learning of a hidden data.

KEYWORDS: Clustering, Customer Relationship Management, Similarity Measures.

INTRODUCTION

Clustering is one of the most important research areas in the field of data mining. Clustering means creating groups of objects based on their features in such a way that the objects belonging to the same groups are similar and those belonging to different groups are dissimilar. Clustering is an unsupervised learning technique. The main advantage of clustering is that interesting patterns and structures can be found directly from very large data sets with little or none of the background knowledge. Clustering algorithms can be applied in many domains.

Clustering is a division of data into groups of similar objects. Representing the data by fewer clusters necessarily loses certain fine details, but achieves simplification. It models data by its clusters. Data modeling puts clustering in a historical perspective rooted in mathematics, statistics, and numerical analysis. From a machine learning perspective clusters correspond to hidden patterns, the search for clusters is unsupervised learning, and the resulting system represents a data concept. From a practical perspective clustering plays an outstanding role in data mining applications such as scientific data exploration, information retrieval and text mining, spatial database applications, Web analysis, CRM, marketing, medical diagnostics, computational biology, and many others. Clustering is the subject of active research in several fields such as statistics, pattern recognition, and machine learning. This paper focuses on clustering in data mining. Data mining adds to clustering the complications of very large datasets with very many attributes of different types. This imposes unique computational requirements on relevant clustering algorithms. A variety of algorithms have recently emerged that meet these requirements and were successfully applied to real-life data mining problems. Therefore it is required to analyze theses algorithms to select best solution for CRM.

Components of a Clustering Task

Typical pattern clustering activity involves the following steps:

(1) Pattern representation (optionally including feature extraction and/or selection),
(2) Definition of a pattern proximity measure appropriate to the data domain,
(3) Clustering or grouping,
(4) Data abstraction (if needed), and
(5) Assessment of output (if needed).

Pattern representation refers to the number of classes, the number of available patterns, and the number, type, and scale of the features available to the clustering algorithm. Some of this information may not be controllable by the practitioner. Feature selection is the process of identifying the most effective subset of the original features to use in clustering. Feature extraction is the use of one or more transformations of the input features to produce new salient features. Either or both of these techniques can be used to obtain an appropriate set of features to use in clustering.

(a) Pattern proximity is usually measured by a distance function defined on pairs of patterns. A variety of distance measures are in use in the various communities. A simple distance measure like Euclidean distance can often be used to reflect dissimilarity between two patterns, whereas other similarity measures can be used to characterize the conceptual similarity between patterns.

The *grouping* step can be performed in a number of ways. The output clustering (or clustering) can be hard (a partition of the data into groups) or fuzzy (where each pattern has a variable degree of membership in each of the output clusters). Hierarchical clustering algorithms produce a nested series of partitions based on a criterion for merging or splitting clusters based on similarity. Partitional clustering

algorithms identify the partition that optimizes (usually locally) a clustering criterion. Additional techniques for the grouping operation include probabilistic and graph-theoretic clustering methods.

(b) ***Data abstraction*** is the process of extracting a simple and compact representation of a data set. Here, simplicity is either from the perspective of automatic analysis (so that a machine can perform further processing efficiently) or it is human-oriented (so that the representation obtained is easy to comprehend and intuitively appealing). In the clustering context, a typical data abstraction is a compact description of each cluster, usually in terms of cluster prototypes or representative patterns such as the centroid.

How is the output of a clustering algorithm evaluated? What characterizes a 'good' clustering result and a 'poor' one? All clustering algorithms will, when presented with data, produce clusters regardless of whether the data contain clusters or not. If the data does contain clusters, some clustering algorithms may obtain 'better' clusters than others. The assessment of a clustering procedure's output, then, has several facets. One is actually an assessment of the data domain rather than the clustering algorithm itself data which do not contain clusters should not be processed by a clustering algorithm. The study of *cluster tendency*, wherein the input data are examined to see if there is any merit to a cluster analysis prior to one being performed, is a relatively inactive research area, and will not be considered further in this paper.

(c)Cluster validity **analysis**, by contrast, is the assessment of a clustering procedure's output. Often this analysis uses a specific criterion of optimality; however, these criteria are usually arrived at subjectively. Hence, little in the way of 'gold standards' exist in clustering except in well-prescribed sub domains. Validity assessments are objective (Dubes 1993) and are performed to determine whether the output is meaningful. A clustering structure is valid if it cannot reasonably have occurred by chance or as an artifact of a clustering algorithm. When statistical approaches to clustering are used, validation is accomplished by carefully applying statistical methods and testing hypotheses. There are three types of validation studies. An *external* assessment of validity compares the recovered structure to an *a priori* structure. An *internal* examination of validity tries to determine if the structure is intrinsically appropriate for the data. A *relative* test compares two structures and measures their relative merit.

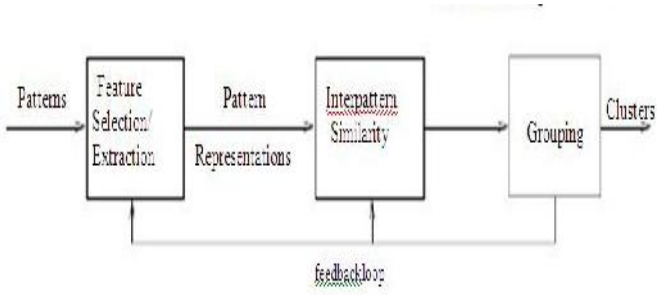

Figure 1 : Stages in clustering

LITERATURE REVIEW

CRM comprises a set of processes and enabling systems supporting a business strategy to build long term, profitable relationships with specific customers [Ling and Yen (2001)]. It is an important technology in every business because all the businesses are customer centric. It consists of identifying, attracting, retaining and developing customers. Customer identification includes target customer analysis and customer segmentation.

Target customer analysis analyzes the customer characteristics to seek segments of customers [Woo et.al. (2005)]. Customer segmentation is the process of dividing customers into homogeneous groups on the basis of common attributes [Zeling Wang and Xinghui Lei (2010)]. Customer segmentation is typically done by applying some form of cluster analysis to obtain a set of segments [Mirko Bottcher et.al. (2009)]. The customer identification is followed by customer attraction which motivates each segment of customers in different way.

Customer retention and customer development deals with retaining the existing customers and maximizing the customer purchase value respectively [Ngai et.al. (2009)].

Clustering is mainly classified into hierarchical and partitioning algorithms. The hierarchical algorithms are further sub divided into agglomerative and divisive. Agglomerative clustering treats each data point as a singleton cluster and then successively merges clusters until all points have been merged into a single cluster. Divisive clustering treats all data points in a single cluster and successively breaks the clusters till one data point remains in each cluster. Partitioning algorithms partition the data set into predefined k number of clusters [Han and Kamber (2001)].

DEFINITIONS AND NOTATION

The following terms and notation are used throughout this paper. A *pattern* (or *feature vector*, *observation*, or *datum*) **x** is a single data item used by the clustering algorithm. It typically consists of a vector of *d* measurements:

$$\mathbf{x} = (x_1, \ldots x_d)$$

The individual scalar components *xi* of a pattern **x** are called *features* (or *attributes*).

- *d* is the *dimensionality* of the pattern or of the pattern space.
- A *pattern set* is denoted $X = (x_1, \ldots x_n)$. The *i*th pattern in X is denoted $xi = (x_{i,1} \ldots x_{i,d})$. In many cases a pattern set to be clustered is viewed as an *n x d pattern matrix*.
- A *class*, in the abstract, refers to a state of nature that governs the pattern generation process in some cases. More concretely, a class can be viewed as a source of patterns whose distribution in feature space is governed by a probability density specific to the class. Clustering techniques attempt to group patterns so that the classes thereby obtained reflect the different pattern generation processes represented in the pattern set.
- *Hard* clustering techniques assign a *class label li* to each patterns **x***i*, identifying its class. The set of all

labels for a pattern set X is L = {l1,…..ln}with li ∈ *(1,…k)* where *k* is the number of clusters.

- *Fuzzy* clustering procedures assign to each input pattern **x***i* a fractional degree of membership *fij* in each output cluster *j*.
- *Fuzzy* clustering procedures assign to each input pattern **x***i* a fractional degree of membership *fij* in each output cluster *j*.
- A *distance measure* (a specialization of a proximity measure) is a metric (or quasi-metric) on the feature space used to quantify the similarity of patterns.

PATTERN REPRESENTATION, FEATURE SELECTION AND EXTRACTION

There are no theoretical guidelines that suggest the appropriate patterns and features to use in a specific situation. Indeed, the pattern generation process is often not directly controllable; the user's role in the pattern representation process is to gather facts and conjectures about the data, optionally perform feature selection and extraction, and design the subsequent elements of the clustering system. Because of the difficulties surrounding pattern representation, it is conveniently assumed that the pattern representation is available prior to clustering. Nonetheless, a careful investigation of the available features and any available transformations (even simple ones) can yield significantly improved clustering results. A good pattern representation can often yield a simple and easily understood clustering; a poor pattern representation may yield a complex clustering whose true structure is difficult or impossible to discern. Figure 2 shows a simple example. The points in this 2D feature space are arranged in a curvilinear cluster of approximately constant distance from the origin. If one chooses Cartesian coordinates to represent the patterns, many clustering algorithms would be likely to fragment the cluster into two or more clusters, since it is not compact. If, however, one uses a polar coordinate representation for the clusters, the radius coordinate exhibits tight clustering and a one-cluster solution is likely to be easily obtained. A pattern can measure either a physical object (e.g., a chair) or an abstract notion (e.g., a style of writing). As noted above, patterns are represented conventionally as multidimensional vectors, where each dimension is a single feature. These features can be either quantitative or qualitative. For example, if *weight* and *color* are the two features used, then ~20, *black*! is the representation of a black object with 20 units of weight. The features can be subdivided into the following types:

(1) Quantitative features: e.g.
   (a) Continuous values (e.g., weight);
   (b) Discrete values (e.g., the number of computers);
   (c) Interval values (e.g., the duration of an event).

(2) Qualitative features:
   (a) Nominal or unordered (e.g., color);
   (b) Ordinal (e.g., military rank or qualitative evaluations of temperature ("cool" or "hot") or sound intensity ("quiet" or "loud")).

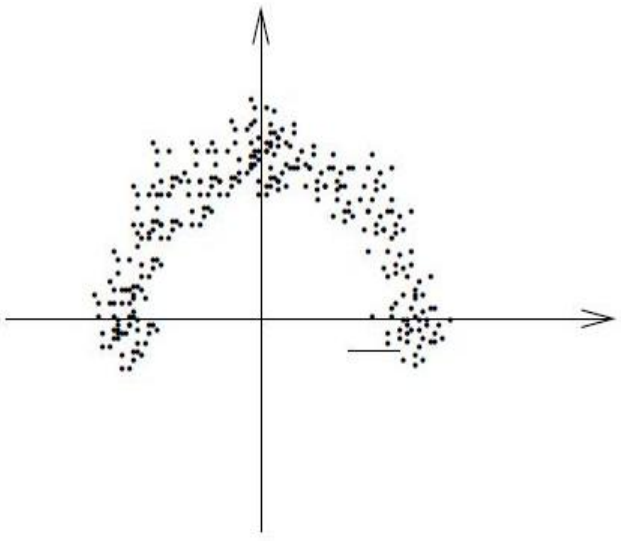

Figure 2: A curvilinear cluster whose points are approximately equidistant from the origin. Different pattern representations (coordinate systems) would cause clustering algorithms to yield different results for this data.

Quantitative features can be measured on a ratio scale (with a meaningful reference value, such as temperature), or on nominal or ordinal scales. One can also use structured features which are represented as trees, where the parent node represents a generalization of its child nodes. For example, a parent node "vehicle" may be a generalization of children labeled "cars," "buses," "trucks," and "motorcycles." Further, the node "cars" could be a generalization of cars of the type "Toyota," "Ford," "Benz," etc.

SIMILARITY MEASURES

Since similarity is fundamental to the definition of a cluster, a measure of the similarity between two patterns drawn from the same feature space is essential to most clustering procedures. Because of the variety of feature types and scales, the distance measure (or measures) must be chosen carefully. It is most common to calculate the *dissimilarity* between two patterns using a distance measure defined on the feature space. We will focus on the well-known distance measures used for patterns whose features are all continuous. The most popular metric for continuous features is the *Euclidean distance.*

$$D2 = (xi, xj) = (\sum_{k=1}^{d} (Xi,k - Xjk)^2)^{1/2}$$

$$= \| \mathbf{x}_i - \mathbf{x}_j \|_2$$

which is a special case (p = 2) of the Minkowski metric

$$d_p = (x_i - x_j) = \left(\sum_{k=1}^{d} |Xi,k - Xj,k|^p\right)^{1/p}$$
$$= \| x_i - x_j \|_p$$

The Euclidean distance has an intuitive appeal as it is commonly used to evaluate the proximity of objects in two or three-dimensional space. It works well when a data set has "compact" or "isolated" clusters.

The drawback to direct use of the Minkowski metrics is the tendency of the largest-scaled feature to dominate the others. Solutions to this problem include normalization of the continuous features (to a common range or variance) or other weighting schemes. Linear correlation among features can also distort distance measures; this distortion can be alleviated by applying a whitening transformation to the data or by using the squared Mahalanobis distance:

$$d_m(x_i, x_j) = (x_i - x_j) \sum^{-1} (x_i - x_j)^T$$

where the patterns **x***i* and **x***j* are assumed to be row vectors, and S is the sample covariance matrix of the patterns or the known covariance matrix of the pattern generation process; $d_m(.\ ,\ .)$ assigns different weights to different features based on their variances and pair wise linear correlations. Here, it is implicitly assumed that class conditional densities are unimodal and characterized by multidimensional spread, i.e., that the densities are multivariate Gaussian. Some clustering algorithms work on a matrix of proximity values instead of on the original pattern set. It is useful in such situations to pre-compute all the n(n - 1) / 2 pair wise distance values for the *n* patterns and store them in a (symmetric) matrix. Computation of distances between patterns with some or all features being non-continuous is problematic, since the different types of features are not comparable and (as an extreme example) the notion of proximity is effectively binary-valued for nominal-scaled features.

Nonetheless, practitioners (especially those in machine learning, where mixed-type patterns are common) have developed proximity measures for heterogeneous type patterns. A comparison of syntactic and statistical approaches for pattern recognition using several criteria was presented in Tanaka, 1995 and the conclusion was that syntactic methods are inferior in every aspect. Therefore, we do not consider syntactic methods further in this thesis. The similarity between two points **x***i* and **x***j*, given this context, is given by:

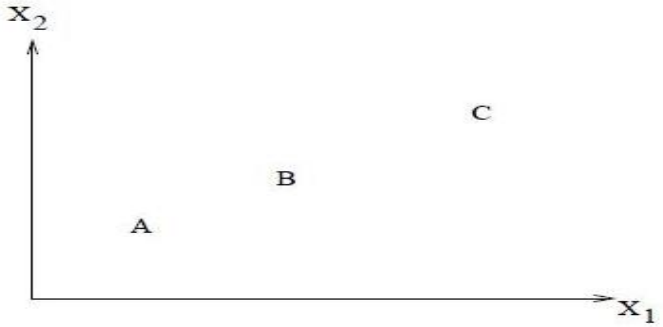

Figure 3:. A and B are more similar than A and C

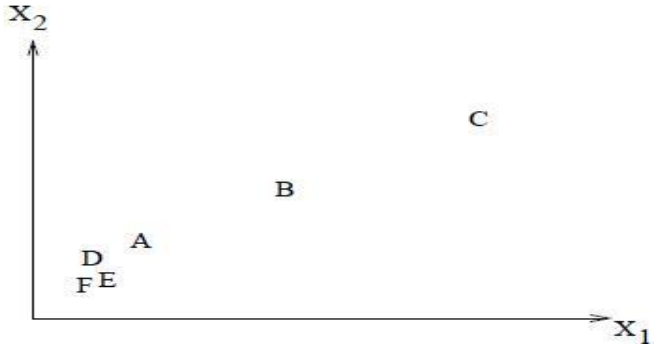

Figure 4: After a change in context, B and C are more similar than B and A.

$$s(xi, xj) = f(xi, xj, E),$$

where E is the context (the set of surrounding points). One metric defined using context is the *mutual neighbor distance* (MND), proposed in Gowda and Krishna, 1977, which is given by :

$$MND(xi, xj) = NN(xi, xj) + NN(xj, xi)$$

Where NN (xj, xi)is the neighbor number of **x***j* with respect to **x***i*. Figures 3 and 4 give an example. In Figure 3, the nearest neighbor of A is B, and B's nearest neighbor is A. So, *NN(A*, B) = *NN(B*, *A*) = 1 and the MND between A and B is 2. However, *NN (B*, *C*) = 1 but *NN(C*, *B*) = 2, and therefore *MND(B*, *C*) = 3. Figure 4 was obtained from Figure 4 by adding three new points D, E, and F. Now *MND~B*, *C*! 5 3 (as before), but *MND(A*, *B*) = 5. The MND between A and B has increased by introducing additional points, even though A and B have not moved. The MND is not a metric. In spite of this, MND has been successfully applied in several clustering. This observation supports the viewpoint that the dissimilarity does not need to be a metric.

Watanabe's theorem of the ugly duckling (Watanabe 1985) states: "Insofar as we use a finite set of predicates that are capable of distinguishing any two objects considered, the number of predicates shared by any two such objects is constant, independent of the choice of objects." This implies that it is possible to make any two arbitrary patterns equally similar by encoding them with a sufficiently large number of features. As a consequence, any two arbitrary patterns are equally similar, unless we use some additional domain information. For example, in the case of conceptual clustering (Michalski and Stepp 1983), the similarity between **x***i* and **x***j* is defined as:

$$s(xi, xj) = f(xi, xj, B, E),$$

where B is a set of pre-defined concepts. This notion is illustrated with the help of Figure 5. Here, the Euclidean distance between points A and B is less than that between B and C. However, B and C can be viewed as "more similar" than A and B because B and C belong to the same concept (ellipse) and A belongs to a different concept (rectangle).

The conceptual similarity measure is the most general similarity measure.

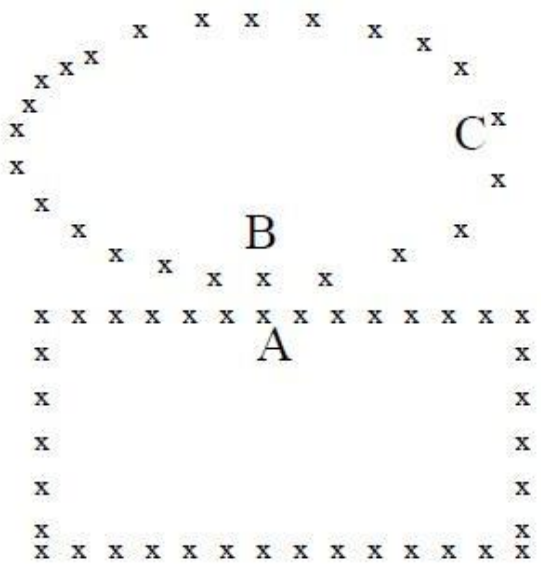

Figure:5: Conceptual similarity between points .

CLUSTERING TECHNIQUES

Traditionally clustering techniques are broadly divided in hierarchical and partitioning. Hierarchical clustering is further subdivided into agglomerative and divisive. The basics of hierarchical clustering include Lance-Williams formula, idea of conceptual clustering, now classic algorithms SLINK, COBWEB, as well as newer algorithms CURE and CHAMELEON. We study them in the section Hierarchical Clustering. While hierarchical algorithms build clusters gradually (as crystals are grown), partitioning algorithms learn clusters directly. In doing so, they either try to discover clusters by iteratively relocating points between subsets, or try to identify clusters as areas highly populated with data. They are further categorized into probabilistic clustering (EM framework, algorithms SNOB, AUTOCLASS, MCLUST), k-medoids methods (algorithms PAM, CLARA, CLARANS, and its extension), and k-means methods (different schemes, initialization, optimization, harmonic means, extensions). Such methods concentrate on how well points fit into their clusters and tend to build clusters of proper convex shapes.

Partitioning algorithms try to discover dense connected components of data, which are flexible in terms of their shape. Density-based connectivity is used in the algorithms DBSCAN, OPTICS, DBCLASD, while the algorithm DENCLUE exploits space density functions. These algorithms are less sensitive to outliers and can discover clusters of irregular shapes. They usually work with low-dimensional data of numerical attributes, known as spatial data. Spatial objects could include not only points, but also extended objects (algorithm GDBSCAN).

Some algorithms work with data indirectly by constructing summaries of data over the attribute space subsets. They perform space segmentation and then aggregate appropriate segments. We discuss them in the section Grid-Based Methods. They frequently use hierarchical agglomeration as one phase of processing. Algorithms BANG, STING, WaveCluster, and an idea of fractal dimension are discussed in this section. Grid-based methods are fast and handle outliers well. Grid-based methodology is also used as an intermadiate step in many other algorithms (for example, CLIQUE, MAFIA). Categorical data is intimately connected with transactional databases. The concept of a similarity alone is not sufficient for clustering such data. The idea of categorical data co-occurrence comes to rescue. The algorithms ROCK, SNN, and CACTUS are studied in the section Co-Occurrence of Categorical Data. The situation gets even more aggravated with the growth of the number of items involved. To help with this problem an effort is shifted from data clustering to pre-clustering of items or categorical attribute values. Development based on hyper-graph partitioning and the algorithm STIRR exemplifies this approach. Many other clustering techniques are developed, primarily in machine learning, that either have theoretical significance, are used traditionally outside the data mining community, or do not fit in previously outlined categories. The boundary is blurred. In the section Other Clustering Techniques we discuss relationship to supervised learning, gradient descent and ANN (LKMA, SOM), evolutionary methods (simulated annealing, genetic algorithms (GA)), and the algorithm AMOEBA. We start, however, with theemerging field of constraint-based clustering that is influenced by requirements of real world data mining applications.

Data Mining primarily works with large databases. Clustering large datasets presents scalability problems reviewed in the section Scalability and VLDB Extensions. Here we talk about algorithms like DIGNET, about BIRCH and other data squashing techniques, and about Hoffding or Chernoff bounds. Another trait of real-life data is its high dimensionality. The trouble comes from a decrease in metric separation when the dimension grows. One approach to dimensionality reduction uses attributes transformations (DFT, PCA, wavelets). Another way to address the problem is through subspace clustering (algorithms CLIQUE, MAFIA, ENCLUS, OPTIGRID, PROCLUS, ORCLUS). Still another approach clusters attributes in groups and uses their derived proxies to cluster objects. This double clustering is known as coclustering. Issues that are common to different clustering methods are overviewed in the section General Algorithmic Issues. We talk about assessment of results, determination of appropriate number of clusters to build, data preprocessing (attribute selection, data scaling, special data indices), proximity measures, and handling outliers.

Important Issues

The properties of clustering include:

- Type of attributes algorithm can handle
- Scalability to large datasets
- Ability to work with high dimensional data

- Ability to find clusters of irregular shape
- Handling outliers
- Time complexity (when there is no confusion, we use the term complexity)
- Data order dependency
- Labeling or assignment (hard or strict vs. soft of fuzzy)
- Reliance on a priori knowledge and user defined parameters
- Interpretability of results

Different approaches to clustering data can be described with the help of the hierarchy shown in Figure 6. At the top level, there is a distinction between hierarchical and partitional approaches (hierarchical methods produce a nested series of partitions, while partitional methods produce only one). The taxonomy shown in Figure 6. must be supplemented by a discussion of cross-cutting issues that may (in principle) affect all of the different approaches regardless of their placement in the taxonomy.

- Agglomerative vs. divisive: This aspect relates to algorithmic structure and operation. An agglomerative approach begins with each pattern in a distinct (singleton) cluster, and successively merges clusters together until a stopping criterion is satisfied. A divisive method begins with all patterns in a single cluster and performs splitting until a stopping criterion is met.
- Monothetic vs. polythetic: This aspect relates to the sequential or simultaneous use of features in the clustering process. Most algorithms are polythetic; that is, all features enter into the computation of distances between patterns, and decisions are based on those distances.
- Hard vs. fuzzy: A hard clustering algorithm allocates each pattern to a single cluster during its operation and in its output. A fuzzy clustering method assigns degrees of membership in several clusters to each input pattern. A fuzzy clustering can be converted to a hard clustering by assigning each pattern to the cluster with the largest measure of membership.
- Deterministic vs. stochastic: This issue is most relevant to partitional approaches designed to optimize a squared error function. This optimization can be accomplished using traditional techniques or through a random search of the state space consisting of all possible labeling.
- Incremental vs. non-incremental: This issue arises when the pattern set to be clustered is large, and constraints on execution time or memory space affect the architecture of the algorithm. The early history of clustering methodology does not contain many examples of clustering algorithms designed to work with large data sets, but the advent of data mining has fostered the development of clustering algorithms that minimize the number of scans through the pattern set, reduce the number of patterns examined during execution, or reduce the size of data structures used in the algorithm's operations.

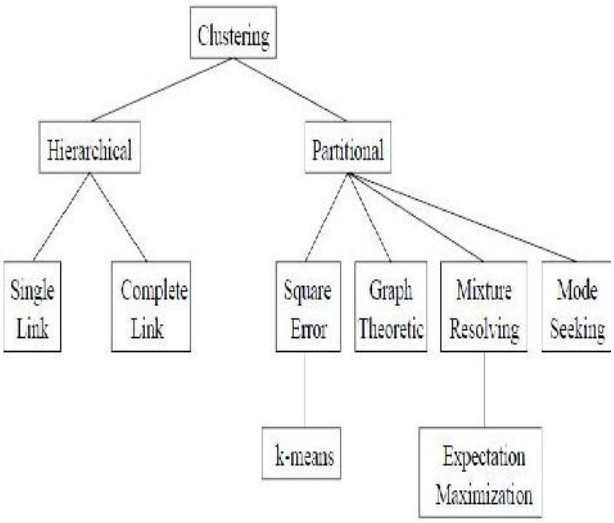

Figure 6: A taxonomy of clustering approaches

CONCLUSION

Customer Relationship Management is a technology that manages relationship with customers in order to improve the performance of business. In CRM, the customer segmentation plays an important role in identifying the customers by grouping similar customers.

This paper provides a comprehensive review of different clustering techniques in data mining. Data mining adds to clustering the complications of very large datasets with very many attributes of different types.

REFERENCES

[1]. Ngai E.W.T.; Li Xiu; D.C.K. Chau. (2009): Application of data mining techniques in customer relationship management: A literature review and classification, Expert Systems with Applications, 36(2), pp. 2592-2602.

[2]. Han, J.; Kamber, M. (2001): Data mining: Concepts and techniques, Morgan Kaufmann Publisher.

[3]. Ling.; and Yen.D.C. (2001): Customer relationship management: An analysis framework and implementation strategies, Journal of Computer Information Systems, 41, pp. 82–97.

[4]. Woo, J. Y.; Bae, S. M.; and Park, S. C. (2005): Visualization method for customer targeting using customer map, Expert Systems with Applications, 28, pp. 763–772.

[5]. Zeling Wang; Xinghui Lei. (2010): Study on Customer Retention under Dynamic Markets. In Proceedings of Second International Conference on Networks Security, Wireless Communications and Trusted Computing, 2, pp. 514-517.

[6]. Mirko Bottcher; Martin Spott; Detlef Nauck; Rudolf Kruse. (2009): Mining changing customer segments in dynamic markets, Expert Systems with Applications, 36(3), pp. 155-164.

[7]. Huaping Gong; Qiong Xia. (2009): Study on Application of Customer Segmentation Based on Data Mining Technology, In Proceedings of the 2009 ETP International Conference on Future Computer and Communication, IEEE Computer Society Washington, DC, pp. 167-170..

[8]. Seyed Mohammad Seyed Hosseini; Anahita Maleki; Mohammad Reza Gholamian. (2010): Cluster analysis using data mining approach to develop CRM methodology to assess the customer loyalty, Expert Systems with Applications, 37(7), pp.5259–5264.

[9]. Watanabe, S., "Theorem of the Ugly Duckling", Pattern Recognition: Human and Mechanical, Wiley, 1985.

[10]. TANAKA, E. 1995. Theoretical aspects of syntactic pattern recognition. Pattern Recogn. 28, 1053-1061.

[11]. MICHALSKI, R., STEPP, R. E., AND DIDAY, E. 1983. Automated construction of classifications: conceptual clustering versus numerical taxonomy. IEEE Trans. Pattern Anal. Mach. Intell. PAMI-5, 5 (Sept.), 396-409.