ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 2, February 2015

State of Art of Different Clustering Approaches

Rohan Rawlani[1], Ritesh Natekar[2], Sahil Pathak[3], Dipali Salve[4], Bhushan S. Thakare[5]

Student, Computer Department Sinhgad Academy of Engineering, Pune, India[1,2,3,4]

Assistant Professor, Information Technology Department, Sinhgad Academy of Engineering, Pune, India[5]

Abstract: This Paper presents an overview of the clustering and its methods used in Data Mining. Firstly, different measures that are used for determining whether two clusters are similar or dissimilar are defined. Then different methods of clustering are presented and are divided into: hierarchical, partitional and evolutionary algorithms. Finally clustering is performed in large data sets and subsequently their challenges are discussed.

Keywords: Clustering, similarity measures, clustering analysis, k-means.

I. Introduction

A. Motivation

Data analysis underlies many computing applications, either in a design phase or as part of their on-line operations. Data analysis procedures can be dichotomized as either exploratory or confirmatory, based on the availability of appropriate models for the data source, but a key element in both types of procedures (whether for hypothesis formation or decision-making) is the grouping or classification of measurements based on either (i) goodness-of fit to a postulated model or (ii) natural groupings (clustering) revealed through analysis. Normally, Clustering is the process of dividing data into groups of similar objects. Each separate group is called cluster and consists of objects that are similar to each other and dissimilar to objects of other groups.

Formally, the clustering structure is represented as a set of subsets, C = C_1, C_2, ... C_k of S, such that:

$$S = \bigcup_{i=1}^{k} C_i \text{ and} \qquad (1)$$

$$C_i \cap C_j = \emptyset \; for \; i \neq j \qquad (2)$$

Consequently, any instance in S belongs to exactly one and only one subset.

Cluster analysis is the organization of a collection of patterns (usually represented as a vector of measurements, or a point in a multidimensional space) into clusters based on similarity. Intuitively, patterns within a valid cluster are more similar to each other than they are to a pattern belonging to a different cluster. An example of clustering is depicted in Fig. 1. The input patterns are shown in Fig. 1(a) and the desired clusters are shown in Fig. 1(b). Here, points belonging to the same cluster are given the same label.

The term ‘clustering’ is used in several research communities to describe methods for grouping of unlabeled data. These communities have different terminologies and assumptions for the components of the clustering process and the context in which clustering is used. Thus, we face a dilemma regarding the scope of this survey. The goal of this paper is to survey the core concepts and techniques in the large subset of cluster analysis with its roots in statistics and decision theory where appropriate, references will be made to key concepts and techniques arising from clustering methodology in the machine learning and other communities.

B. Components of Clustering Task

Typical pattern clustering activity involves the following steps [4]:

i) Pattern representation (optionally including feature extraction and/or selection),
ii) Definition of a pattern proximity measure appropriate to the data domain,
iii) Clustering or grouping,
iv) Data abstraction (if needed), and
v) Assessment of output (if needed).

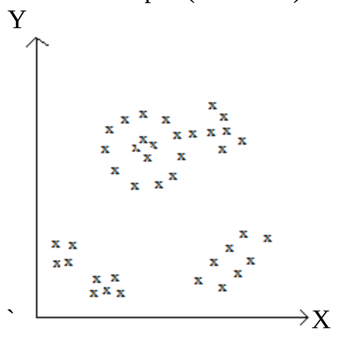

Fig. 1 (a) Data Clustering

Fig. 2 depicts a typical sequencing of the first three of these steps, including a feedback path where the grouping process output could affect subsequent feature extraction and similarity computations.

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.

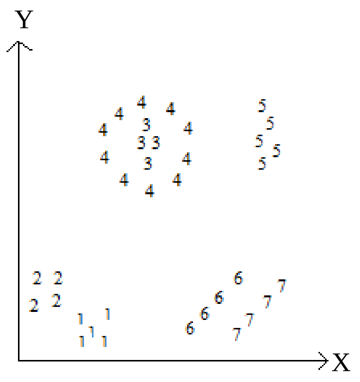

Fig.1 (b) Data Clustering

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 2, February 2015

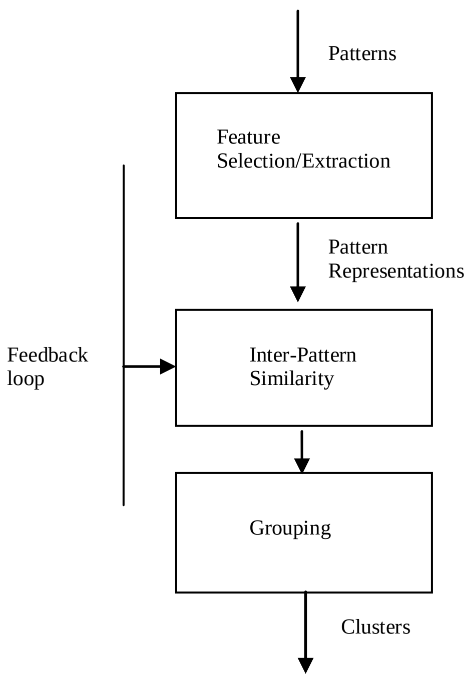

Fig. 2. Stages in clustering

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.

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 [3 4 5]. A simple distance measure like the 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 [6].

The grouping step can be performed in a number of ways. The output clustering (or clusterings) 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).

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 [5].

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.

II. 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. The paper will focus on the well-known distance measures used for patterns whose features are all continuous.

A. Euclidean Distance Measure

The most popular metric for continuous features is the Euclidean distance. It is given as:

$$d_2(x_i, x_j) = (\sum_{k=1}^{d} (x_{i,k} - x_{j,k})^2)^{\frac{1}{2}} = \|x_i - x_j\|2 \quad ...(3)$$

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 [7].

B. Minkowski Distance Measure

It is given as:

$$d_p(x_i, x_j) = (\sum_{k=1}^{d} |x_{i,k} - x_{j,k}|^2)^{\frac{1}{p}} = \|x_i - x_j\|p \quad ...(4)$$

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'.

C. Mahalanobis Distance Measure

It is given as:

$$d_M(x_i, x_j) = (x_i - x_j)\Sigma^{-1}(x_i - x_j)^T \quad ...(5)$$

where the patterns x_i and x_j are assumed to be row vectors, and 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 pairwise linear correlations. The regularized Mahalanobis distance was used in [7] to extract hyper ellipsoidal clusters. Recently, several researchers [29 9] have used the Hausdorff distance in a point set matching context.

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 2, February 2015

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) pairwise distance values for the n patterns and store them in a (symmetric) matrix. A variety of other metrics have been reported in [5 10] for computing the similarity between patterns represented using quantitative as well as qualitative features.

Patterns can also be represented using string or tree structures [11]. Strings are used in syntactic clustering [28]. Several measures of similarity between strings are described in [20]. A good summary of similarity measures between trees is given by [13]. A comparison of syntactic and statistical approaches for pattern recognition using several criteria was presented in [20] and the conclusion was that syntactic methods are inferior in every aspect. Therefore, we do not consider syntactic methods further in this paper.

There are some distance measures reported in the literature [15 24] that take into account the effect of surrounding or neighboring points. These surrounding points are called context in [6]. The similarity between two points' x_i and x_j, given this context, is given by:

$$s(x_i, x_j) = f(x_i, x_j, \varepsilon) \qquad \dots (6)$$

Where E is the context (the set of surrounding points). One metric defined using context is the mutual neighbor distance (MND), proposed in [15], which is given by:

$$\mathrm{MND}(x_i, x_j) = \mathrm{NN}(x_i, x_j) + NN(x_j, x_i) \qquad \dots (7)$$

Where NN (x_i, x_j) is the neighbor number of x_j with respect to x_i.

III. CLUSTERING TECHNIQUES

There are basically two clustering techniques:-

A) Hierarchical Clustering – This is further divided into sub clustering techniques:-
1) Single Link
2) Complete Link

B) Partitional Clustering – This also divided into following sub techniques:-
1) Squared Error – again divided into K-means
2) Graph Theoretic
3) Mixture Resolving – again divided into Expectation Maximization
4) Mode Seeking

This clustering hierarchy is as show in Fig. 3

A. Hierarchical Clustering

The operation of a hierarchical clustering algorithm is illustrated using the two-dimensional data set as shown in Fig. 4. This figure depicts seven patterns labelled A, B, C, D, E, F, and G in three clusters. A hierarchical algorithm yields a dendrogram representing the nested grouping of patterns and similarity levels at which groupings change. A dendrogram corresponding to the seven points in Fig. 3 (obtained from the single-link algorithm [4]) is shown in Fig. 4. The dendrogram can be broken at different levels to yield different clusterings of the data.

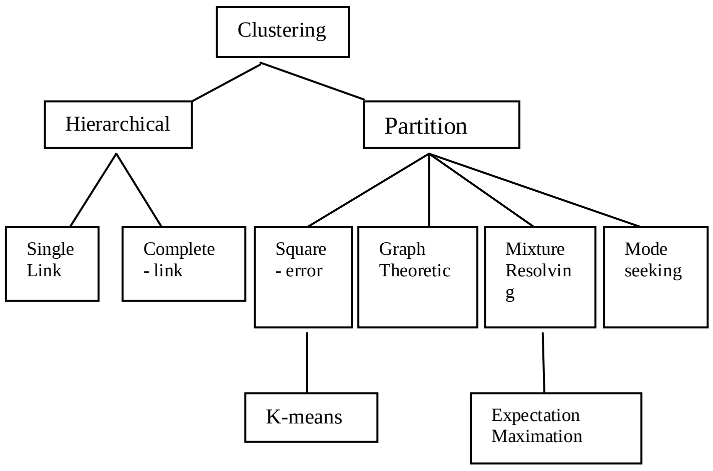

Fig. 2. Clustering Hierarchy

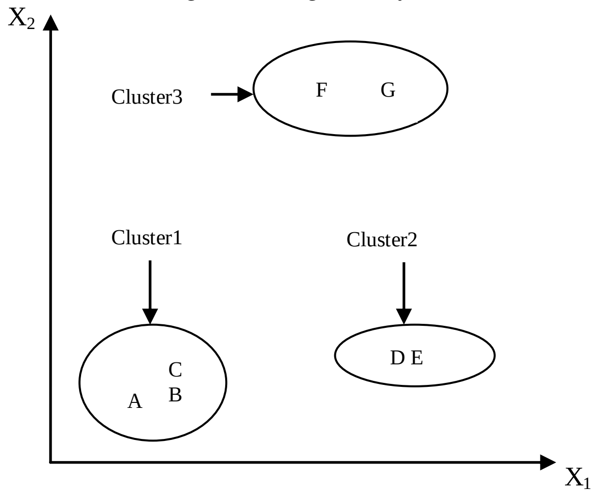

Fig. 3. Points falling three clusters

Most hierarchical clustering algorithms are variants of
1) Single-link [36], and
2) Complete-link [19], algorithms.

These two algorithms differ in the way they characterize the similarity between a pair of clusters. In the single-link method, the distance between two clusters is the minimum of the distances between all pairs of patterns drawn from the two clusters (one pattern from the first cluster, and the other from the second).

In the complete-link algorithm, the distance between two clusters is the maximum of all pairwise distances between patterns in the two clusters. In either case, two clusters are merged to form a larger cluster based on minimum distance criteria.

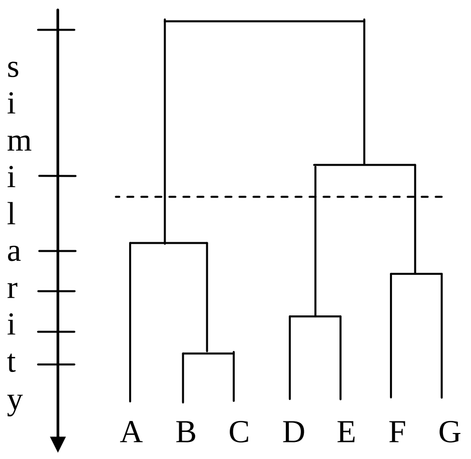

Fig. 4.The Dendrogram obtained using the single link algorithm

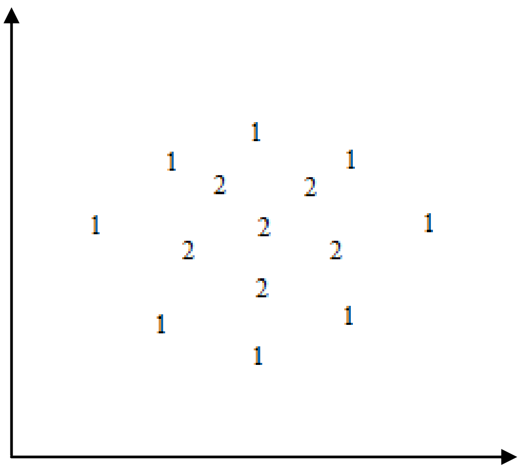

Fig. 5. Two concentric clusters

The complete-link algorithm produces tightly bound or compact clusters [20]. The single-link algorithm, by contrast, suffers from a chaining effect [21]. It has a tendency to produce clusters that are straggly or elongated. There are two clusters in Fig. 6 and Fig. 7 separated by a "bridge" of noisy patterns. The single-link algorithm produces the clusters shown in Fig. 6, whereas the complete-link algorithm obtains the clustering shown in Fig. 7. The clusters obtained by the complete-link algorithm are more compact than those obtained by the single-link algorithm. The cluster labelled 1 obtained using the single-link algorithm is elongated because of the noisy patterns labelled "*". The single-link algorithm is more versatile than the complete-link algorithm, otherwise.

For example, the single-link algorithm can extract the concentric clusters shown in Fig. 5, but the complete-link algorithm cannot. However, from a pragmatic viewpoint, it has been observed that the complete-link algorithm produces more useful hierarchies in many applications than the single-link algorithm [4].

Agglomerative Single-Link Clustering Algorithm is as given below:

1) Place each pattern in its own cluster. Construct a list of interpattern distances for all distinct unordered pairs of patterns, and sort this list in ascending order.

2) Step through the sorted list of distances, forming for each distinct dissimilarity value d_k; a graph on the patterns where pairs of patterns closer than d_k are connected by a graph edge. If all the patterns are members of a connected graph, stop. Otherwise, repeat this step.

3) The output of the algorithm is a nested hierarchy of graphs which can be cut at a desired dissimilarity level forming a partition (clustering) identified by simply connected components in the corresponding graph.

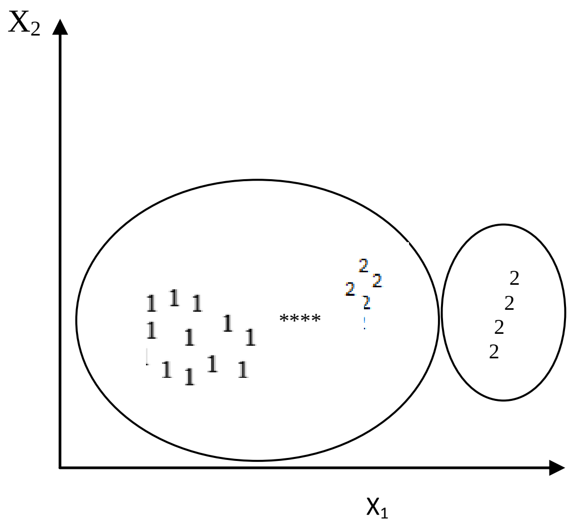

Fig. 6. A single link clustering of pattern set containing two classes (1 and 2) connected by a chain of noisy patterns (*)

Agglomerative Complete-Link Clustering Algorithm is as given below:

1) Place each pattern in its own cluster. Construct a list of interpattern distances for all distinct unordered pairs of patterns, and sort this list in ascending order.

2) Step through the sorted list of distances, forming for each distinct dissimilarity value dk; a graph on the patterns where pairs of patterns closer than dk are connected by a graph edge. If all the patterns are members of a completely connected graph, stop.

3) The output of the algorithm is a nested hierarchy of graphs which can be cut at a desired dissimilarity level forming a partition (clustering) identified by completely connected components in the corresponding graph.
Hierarchical Agglomerative Clustering Algorithm is as given below:
1) Compute the proximity matrix containing the distance between each pair of patterns. Treat each pattern as a cluster.

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 2, February 2015

2) Find the most similar pair of clusters using the proximity matrix. Merge these two clusters into one cluster. Update the proximity matrix to reflect this merge operation.

3) If all objects are in one cluster, stop. Else, go to step 2.

Based on the way the proximity matrix is updated in step 2, a variety of agglomerative algorithms can be designed. Hierarchical divisive algorithms start with a single cluster of all the given objects and keep splitting the clusters based on some criterion to obtain a partition of singleton clusters.

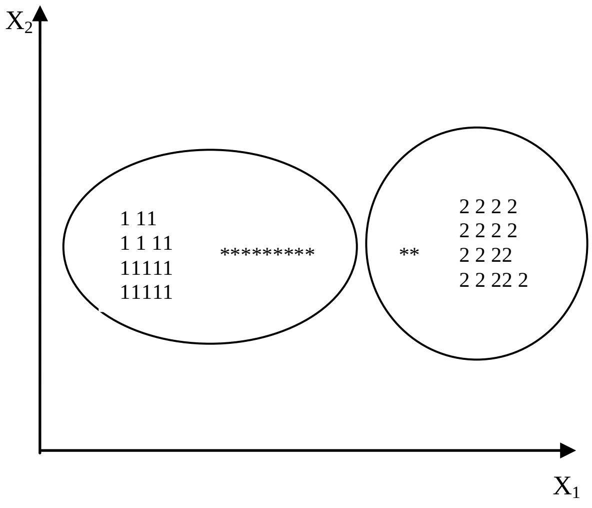

Fig. 7. A complete link clustering of pattern set containing two classes (1 and 2) connected by a chain of noisy patterns (*)

B. *Partitional Algorithms*

A partitional clustering algorithm obtains a single partition of the data instead of a clustering structure, such as the dendrogram produced by a hierarchical technique. Partitional methods have advantages in applications involving large data sets for which the construction of a dendrogram is computationally prohibitive. A problem accompanying the use of a partitional algorithm is the choice of the number of desired output clusters. A seminal paper [22] provides guidance on this key design decision. The partitional techniques usually produce clusters by optimizing a criterion function defined either locally (on a subset of the patterns) or globally (defined over all of the patterns). Combinatorial search of the set of possible labelling for an optimum value of a criterion is clearly computationally prohibitive. In practice, therefore, the algorithm is typically run multiple times with different starting states and the best configuration obtained from all of the runs is used as the output clustering.
The Partitional algorithms are subcategorized into:

1) *Squared Error Clustering:*

The most intuitive and frequently used criterion function in partitional clustering techniques is the squared error criterion, which tends to work well with isolated and compact clusters. The squared error for a clustering L of a pattern set X (containing K clusters) is
where x(ij) is the ith pattern belonging to the jth cluster and cj is the centroid of the jth cluster.

K-means Algorithm

The k-means algorithm is the simplest and most commonly used algorithm employing a squared error criterion [23]. It starts with a random initial partition and keeps on reassigning the patterns to clusters based on the similarity between the pattern and the cluster centres until a convergence criterion is met (e.g., there is no reassignment of any pattern from one cluster to another, or the squared error ceases to decrease significantly after some number of iterations). The k-means algorithm is popular because it is easy to implement and its time complexity is O (n), where n is the number of patterns. A major problem with this algorithm is that it is sensitive to the selection of the initial partition and may converge to a local minimum of the criterion function value if the initial partition is not properly chosen. Fig. 8 shows seven two-dimensional patterns. If we start with patterns A, B, and C as the initial means around which the 3 clusters are built, then we end up with the partition {{A}, {B, C}, {D, E, F, G}} shown by ellipses. The squared error criterion value is much larger for this partition than for the best partition {{A, B, C}, {D, E}, {F, G}} shown by rectangles, which yields the global minimum value of the squared error criterion function for a clustering containing three clusters. The correct three-cluster solution is obtained by choosing, for example, A, D, and F as the initial cluster means.

Agglomerative Squared Error Clustering Method

1) Select an initial partition of the patterns with a fixed number of clusters and cluster centres.

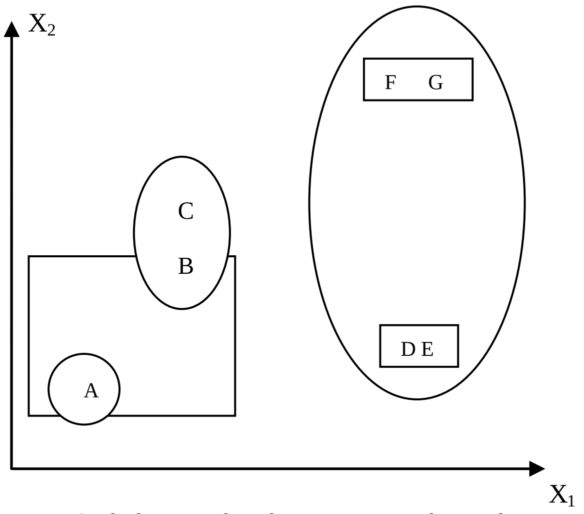

Fig. 8. The k-means algorithm is sensitive to the initial position

2) Assign each pattern to its closest cluster centre and compute the new cluster centres as the centroid of the clusters. Repeat this step until convergence is achieved, i.e., until the cluster membership is stable.

3) Merge and split clusters based on some heuristic information, optionally repeating step 2.

Agglomerative k-Means Clustering Algorithm

1) Choose k cluster centres to coincide with k randomly chosen patterns or k randomly defined points inside the hypervolume containing the pattern set.

2) Assign each pattern to the closest cluster centre.

3) Re-compute the cluster centres using the current cluster memberships.

4) If a convergence criterion is not met, go to step 2. Typical convergence criteria are: no (or minimal) reassignment of patterns to new cluster centres, or minimal decrease in squared error.

Several variants [3] of the k-means algorithm have been reported in the literature. Some of them attempt to select a good initial partition so that the algorithm is more likely to find the global minimum value. Another variation is to permit splitting and merging of the resulting clusters. Typically, a cluster is split when its variance is above a pre-specified threshold and two clusters are merged when the distance between their centroids is below another pre-specified threshold. Using this variant, it is possible to obtain the optimal partition starting from any arbitrary initial partition, provided proper threshold values are specified. The well-known ISODATA [24] algorithm employs this technique of merging and splitting clusters. If ISODATA is given the "ellipse" partitioning shown in Fig. 8 as an initial partitioning, it will produce the optimal three-cluster partitioning. ISODATA will first merge the clusters {A} and {B, C} into one cluster because the distance between their centroids is small and then split the cluster {D, E, F, G} (which has a large variance), into two clusters {D, E} and {F,G}.

2) Graph Theoretic Clustering:

The most well-known graph-theoretic divisive clustering algorithm is based on the construction of the minimal spanning tree (MST) of the data [25] and then deleting the MST edges with the largest lengths to generate more clusters.

The hierarchical approaches are also related to graph-theoretic clustering. Single-link clusters are sub-graphs of the minimum spanning tree of the data [26] which are also the connected components [27]. Complete-link clusters are maximal complete sub-graphs [27] and are related to the node colorability of graphs [28]. The maximal complete sub-graph was considered the strictest definition of a cluster in [29] and [30]. A graph-oriented approach for non-hierarchical structures and over-lapping clusters is presented in [31]. The Delaunay graph (DG) is obtained by connecting all the pairs of points that are Voronoi neighbors. The DG contains all the neighborhood information contained in the MST and the relative neighborhood graph (RNG) [2].

3) Mixture Resolving and Mode Seeking Algorithms:

The mixture resolving approach to cluster analysis has been addressed in a number of ways. The underlying assumption is that the patterns to be clustered are drawn from one of several distributions, and the goal is to identify the parameters of each and (perhaps) their number. Most of the work in this area has assumed that the individual components of the mixture density are Gaussian, and in this case the parameters of the individual Gaussians are to be estimated by the procedure. Traditional approaches to this problem involve obtaining (iteratively) a maximum likelihood estimate of the parameter vectors of the component densities [4].

More recently, the Expectation Maximization (EM) algorithm (a general-purpose maximum likelihood algorithm [33] for missing-data problems) has been applied to the problem of parameter estimation. A recent book [34] provides an accessible description of the technique. In the EM framework, the parameters of the component densities are unknown, as are the mixing parameters, and these are estimated from the patterns. The EM procedure begins with an initial estimate of the parameter vector and iteratively rescores the patterns against the mixture density produced by the parameter vector. The rescored patterns are then used to update the parameter estimates. In a clustering context, the scores of the patterns (which essentially measure their likelihood of being drawn from particular components of the mixture) can be viewed as hints at the class of the pattern. Those patterns, placed (by their scores) in a particular component, would therefore be viewed as belonging to the same cluster.

Nonparametric techniques for density-based clustering have also been developed [4]. Inspired by the Parzen window approach to non parametric density estimation, the corresponding clustering procedure searches for bins with large counts in a multidimensional histogram of the input pattern set.

4) Nearest Neighbor Clustering:

Since proximity plays a key role in our intuitive notion of a cluster, nearest-neighbor distances can serve as the basis of clustering procedures. An iterative procedure was proposed in [1]. It assigns each unlabeled pattern to the cluster of its nearest labelled neighbor pattern, provided the distance to that labelled neighbor is below a threshold. The process continues until all patterns are labelled or no additional labelling occurs. The mutual neighborhood value (described earlier in the context of distance computation) can also be used to grow clusters from near neighbors.

5) Fuzzy Clustering:

Traditional clustering approaches generate partitions; in a partition, each pattern belongs to one and only one cluster. Hence, the clusters in a hard clustering are disjoint. Fuzzy clustering extends this notion to associate each pattern

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 2, February 2015

with every cluster using a membership function [32]. The output of such algorithms is a clustering, but not a partition.

Agglomerative Fuzzy Clustering Algorithm

1) Select an initial fuzzy partition of the N objects into K clusters by selecting the N*K membership matrix U. An element u_{ij} of this matrix represents the grade of membership of object x_i in cluster c_j. Typically,

$$\mu_{i,j} \epsilon [0,1] \quad \ldots (8)$$

2) Using U, find the value of a fuzzy criterion function, e.g. a weighted squared error criterion function, associated with the corresponding partition. One possible fuzzy criterion function is

3) Reassign patterns to clusters to reduce this criterion function value and re-compute U.

4) Repeat step 2 until entries in U. Do not change significantly.

In fuzzy clustering, each cluster is a fuzzy set of all the patterns. Fig. 9 illustrates the idea. The rectangles enclose two "hard" clusters in the data: H1 = {1, 2, 3, 4, and 5} and H2 = {6, 7, 8, and 9}. A fuzzy clustering algorithm might produce the two fuzzy clusters F1 and F2 depicted by ellipses. The patterns will have membership values in [0, 1] for each cluster. For example, fuzzy cluster F1 could be compactly described as:
{(1, 0.9), (2, 0.8), (3, 0.7), (4, 0.6), (5, 0.55), (6, 0.2), (7, 0.2), (8, 0.0), (9, 0.0)} and
F2 could be described as:
{(1, 0.0), (2, 0.0), (3, 0.0), (4, 0.1), (5, 0.15), (6, 0.4), (7, 0.35), (8, 1.0), (9, 0.9)}

The ordered pairs (i, μ_i) in each cluster represent the ith pattern and its membership value to the cluster μ_i. Larger membership values indicate higher confidence in the assignment of the pattern to the cluster. A hard clustering can be obtained from a fuzzy partition by thresholding the membership value.

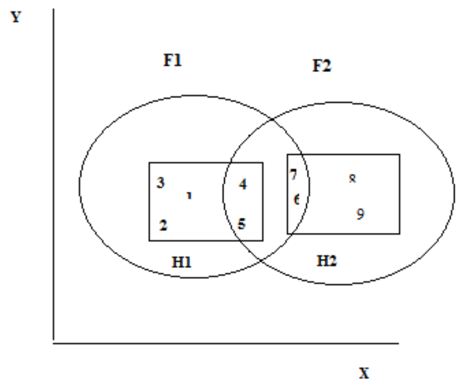

Fig. 9. Fuzzy clusters

Fuzzy set theory was initially applied to clustering in [35]. The most popular fuzzy clustering algorithm is the fuzzy c-means (FCM) algorithm [36]. Even though it is better than the hard k-means algorithm in avoiding local minima, FCM can still converge to local minima of the squared error criterion. The design of membership functions is the most important problem in fuzzy clustering; different choices include those based on similarity decomposition [36] and centroids of clusters. A generalization of the FCM algorithm was proposed by [36] through a family of objective functions. A fuzzy c-shell algorithm and an adaptive variant for detecting circular and elliptical boundaries were presented in [36].

6) *Evolutionary Approach for Clustering:*
Evolutionary approach, motivated by natural evolution, makes use of evolutionary operators and a population of solutions to obtain the globally optimal partition of the data. Candidate solutions to the clustering problem are encoded as chromosomes. The most commonly used evolutionary operators are: selection, recombination, and mutation. Each transforms one or more input chromosomes into one or more output chromosomes. A fitness function evaluated on a chromosome determines a chromosome's likelihood of surviving into the next generation. We give below a high-level description of an evolutionary algorithm applied to clustering.

An Evolutionary Algorithm for Clustering

1) Choose a random population of solutions. Each solution here corresponds to a valid k-partition of the data. Associate a fitness value with each solution. Typically, fitness is inversely proportional to the squared error value. A solution with a small squared error will have a larger fitness value.

2) Use the evolutionary operators' selection, recombination and mutation to generate the next population of solutions. Evaluate the fitness values of these solutions.

3) Repeat step 2 until some termination condition is satisfied.

IV. Clustering Large Data Sets

There are several applications where it is necessary to cluster a large collection of patterns. The definition of 'large' has varied (and will continue to do so) with changes in technology (e.g., memory and processing time). In the 1960s, 'large' meant several thousand patterns [38] now there are applications where millions of patterns of high dimensionality have to be clustered. For example, to segment an image of size 500 * 500 pixels, the number of pixels to be clustered is 250,000. In document retrieval and information filtering, millions of patterns with a dimensionality of more than 100 have to be clustered to achieve data abstraction. A majority of the approaches and algorithms proposed in the literature cannot handle such large data sets. Approaches based on genetic algorithms, tabu search and simulated annealing are optimization techniques and are restricted to reasonably small data sets. Implementations of conceptual clustering optimize some

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 2, February 2015

criterion functions and are typically computationally expensive.

The convergent k-means algorithm and its ANN equivalent, the Kohonen net, have been used to cluster large data sets [7]. The reasons behind the popularity of the k-means algorithm are:

- Its time complexity is O (nkl), where n is the number of patterns, k is the number of clusters, and l is the number of iterations taken by the algorithm to converge. Typically, k and l are fixed in advance and so the algorithm has linear time complexity in the size of the data set [38].

- Its space complexity is O (k + n). It requires additional space to store the data matrix. It is possible to store the data matrix in a secondary memory and access each pattern based on need. However, this scheme requires a huge access time because of the iterative nature of the algorithm and as a consequence processing time increases enormously.

- It is order-independent for a given initial seed set of cluster centres, it generates the same partition of the data irrespective of the order in which the patterns are presented to the algorithm.

However, the k-means algorithm is sensitive to initial seed selection and even in the best case, it can produce only hyper-spherical clusters.

V. CONCLUSION

Thus, Clustering is a process of grouping data items based on a measure of similarity. Clustering is a subjective process; the same set of data items often needs to be partitioned differently for different applications. This subjectivity makes the process of clustering hard. This is because a single algorithm or approach is not adequate to solve every clustering problem. A possible solution lies in reflecting this subjectivity in the form of knowledge. This knowledge is used either implicitly or explicitly in one or more phases of clustering. Knowledge-based clustering algorithms use domain knowledge explicitly.

The most challenging step in clustering is feature extraction or pattern representation.

In this paper, we have examined various steps in clustering and surveyed different clustering techniques such as hierarchical and partitional. Also we have discussed statistical, fuzzy, neural, and evolutionary, approaches to clustering.

The k-means algorithm and its neural implementation, the Kohonen net, are most successfully used on large data sets. This is because k-means algorithm is simple to implement and computationally attractive because of its linear time complexity. However, it is not feasible to use even this linear time algorithm on large data sets.

In summary, clustering is an interesting, useful and challenging problem. It has great potential in applications like object recognition, image segmentation, and information filtering and retrieval. However, it is possible to exploit this potential only after making several design choices carefully.

REFERENCE

[1] S.Y. Lu and K.S. Fu. A Sentence-to-sentence Clustering Procedure for Pattern Analysis. IEEE Trans. on Systems, Man and Cybernetics SMC-8:381-389, 1978.
[2] G. T. Toussaint. The Relative Neighborhood Graph of a Finite Planar Set. Pattern Recog-nition, 12:261-268, 1980.
[3] M. R. Anderberg. Cluster Analysis for Applications. Academic Press, Inc., New York, 1973.
[4] A. K. Jain and R. C. Dubes. Algorithms for Clustering Data, Prentice Hall, Englewood Cliffs, 1988.
[5] E. Diday and J. C. Simon. Clustering Analysis. In K. S. Fu, editor, Digital Pattern Recogni-tion, pp. 47-94, Springer Verlag, New York, 1976.
[6] R. S. Michalski and R. E. Stepp. Automated Construction of Classifications: Conceptual Clustering Versus Numerical Taxonomy. IEEE Trans. Pattern Analysis and Machine Intelligence, 5:396-409, 1983.
[7] J. Mao and A. K. Jain. A Self-Organizing Network for Hyper-ellipsoidal Clustering (HEC). IEEE Trans. Neural Networks, 7: 16-29, 1996.
[8] D.P. Huttenlocher, G.A. Klanderman and W.J. Rucklidge. Comparing Images Using the Hausdorff Distance. IEEE Trans. on Pattern Analysis and Machine Intelligence 15(9):850-863, 1993.
[9] M.P. Dubuisson and A.K. Jain. A Modified Hausdorff Distance for Object Matching. Proc. Int. Conf. on Pattern Recognition (ICPR '94) vol A, pp. 566-568, 1994.
[10] M. Ichino and H. Yaguchi. Generalized Minkowski Metrics for Mixed Feature-Type Data Analysis. IEEE Trans. Systems, Man, and Cybernetics, 24:698-708, 1994.
[11] D. E. Knuth. The Art of Computer Programming, Addision-Wesley, Reading, 1973.
[12] K. S. Fu and S. Y. Lu. A Clustering Procedure for Syntactic Patterns. IEEE Trans. Systems, Man and Cybernetics, 7:734-742, 1977.
[13] K. Zhang. Algorithms for the Constrained Editing Distance Between Ordered labeled Trees and Related Problems. Pattern Recognition, 28:463-474, 1995.
[14] E. Tanaka. Theoretical Aspects of Syntactic Pattern Recognition. Pattern Recognition, 28:1053-1061, 1995.
[15] K. C. Gowda and G. Krishna. Agglomerative Clustering Using the Concept of Mutual Nearest Neighborhood. Pattern Recognition, 10:105-112, 1977.
[16] R. A. Jarvis and E. A. Patrick. Clustering Using a Similarity Method Based on Shared Near Neighbors. IEEE Trans. Computers, 22:1025-1034, 1973.
[17] R. S. Michalski and R. E. Stepp. Automated Construction of Classifications: Conceptual Clustering Versus Numerical Taxonomy. IEEE Trans. Pattern Analysis and Machine Intelligence, 5:396-409, 1983.
[18] P. H. A. Sneath and R. R. Sokal. Numerical Taxonomy, Freeman, San Francisco, 1973.
[19] B. King. Step-Wise Clustering Procedures. Journal of the American Statistical Association, 69:86-101, 1967.
[20] R. A. Baeza-Yates. Introduction to Data Structures and Algorithms Related to Information Retrieval. In W. B. Frakes and R. A. Baeza-Yates, editors, Information Retrieval: Data Structures and Algorithms, pp. 13-27, Prentice Hall, New Jersey, 1992.
[21] G. Nagy. State of the Art in Pattern Recognition. Proc. IEEE, 56:836-862, 1968.
[22] R. C. Dubes. How Many Clusters Are Best? - An Experiment. Pattern Recognition, 20:645-663, 1987.
[23] J. McQueen. Some Methods for Classification and Analysis of Multivariate Observations. Fifth Berkeley Symposium on Mathematical Statistics and Probability, 1:281-297, 1967.
[24] G. H. Ball and D. J. Hall. ISODATA, A Novel Method of Data Analysis and Classification. technical Report, Stanford Research Institute, California, 1965.

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 2, February 2015

[25] C. T. Zahn. Graph-Theoretical Methods for Detecting and Describing Gestalt Clusters. IEEE Trans. Computers, 20:68-86, 1971.
[26] J. C. Gower and G. J. S. Ross. Minimum Spanning Trees and Single-Linkage Cluster Analysis. Applied Statistics, 18:54-64, 1969.
[27] G. C. Gotlieb and S. Kumar. Semantic Clustering of Index Terms. Journal of the ACM, 15:493-513, 1968.
[28] F. B. Backer and L. J. Hubert. A Graph-Theoretic Approach to Goodness-Of-Fit in Complete-Link Hierarchical Clustering. Journal of the American Statistical Association, 71:870-878, 1976.
[29] J. G. Augustson and J. Minker. An Analysis of Some Graph Theoretical Clustering Techniques. Journal of the ACM, 17:571-588, 1970.
[30] V. V. Raghavan and C. T. Yu. A Comparison of the Stability Characteristics of Some Graph Theoretic Clustering Methods. IEEE Trans. Pattern Analysis and Machine Intelligence, 3:393-402, 1981.
[31] K. Ozawa. A Stratificational Overlapping Cluster Scheme. Pattern Recognition, 18:279-286, 1985.
[32] L. A. Zadeh. Fuzzy Sets. Information and Control, 8:338-353, 1965.
[33] A.P. Dempster, N.M. Laird and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. J. Royal Statistical Society B39 (1):1-38, 1977.
[34] T. Mitchell. Machine Learning. WCB/McGraw-Hill, 1997.
[35] E. H. Ruspini. A New Approach to Clustering. Inform. and Control, 15:22-32, 1969.
[36] J. C. Bezdek. Pattern Recognition with Fuzzy Objective Function Algorithms, Plenum Press, New York, 1981.
[37] R. N. Dave. Generalized Fuzzy C-Shells Clustering and Detection of Circular and Elliptic Boundaries. Pattern Recognition, 25:713-722, 1992.
[38] G. J. S. Ross. Classification Techniques for Large Sets of Data. In A. J. Cole, editor, Numerical Taxonomy, Academic Press, N. Y., 1968.