Survey on Cluster Analysis

Arun Marx.M[1], Pavithra.T[2], Praveena.V[3]
Assistant professor, Dept of Information Technology, Prathyusha Engineering College,India.[1]
'Student,Dept of Information Technology, Prathyusha Engineering College,India.[2]
Student,Dept of Information Technology, Prathyusha Engineering College,India.[3]
arunmarx.it@prathyusha.edu.in[1];pavi.t33@gmail.com[2];pravi.it1996@gmail.com[3]

***Abstract*— Data analysis plays a vital role for understanding various phenomena. Cluster analysis is a challenging field of research in which its potential applications pose their own special necessities. Clustering is used to group the similar objects in the datasets and further precede these clustering data to perform other data analysis techniques. This paper deals with the types of clustering techniques and the type of data used in each clustering technique. And the efficiency of clustering technique over classification is given in this paper. Several clustering techniques are discussed here to find out the suitable technique for different datasets.**

***Keywords* — Hierarchical Clustering, Density-based Clustering, Model-based clustering, Constraint-based clustering, Partitioning methods.**

I.INTRODUCTION

Clustering is the process of grouping a set of physical or abstract objects into classes of similar objects is called clustering. A cluster is a collection of data objects that are similar to one another within the same cluster and are dissimilar to the objects in other clusters. A cluster of data objects can be treated collectively as one group and so may be considered as a form of data compression. The following are typical requirements of clustering in data mining:

- Scalability
- Ability to deal with different types of attributes
- Discovery of clusters with arbitrary
- Minimal requirements for domain knowledge to determine input parameters
- Ability to deal with noisy data
- High dimensionality
- Interpretability and usability.

II. TYPES OF DATA USED IN CLUSTERING ANALYSIS

We study the types of data that often occur in cluster analysis and how to preprocess them for such an analysis. Suppose that a data set to be clustered contains n objects, which may represent persons, houses, documents, countries, and so on. Main memory-based clustering algorithms typically operate on either of the following two data structures.

A. Data matrix (or object-by-variable structure): This represents n objects, such as persons, with p variables (also called measurements or attributes), such as age, height, weight, gender, and so on. The structure is in the form of a relational table, or n-by-p matrix (n objects _p variables):

B. Dissimilarity matrix (or object-by-object structure): This stores a collection of proximities that are available for all pairs of n objects. It is often represented by an n-by-n table: where d(i, j) is the measured difference or dissimilarity between objects i and j. In general, d(i, j) is a nonnegative number that is close to 0 when objects i and j are highly similar or "near" each other, and becomes larger the more they differ. Since d(i, j)=d(j, i), and d(i, i)=0, we have the matrix in (7.2). The rows and columns of the data matrix represent different entities, while those of the dissimilarity matrix represent the same entity. Thus, the data matrix is often called a two-mode matrix, whereas the dissimilarity matrix is called a one-mode matrix. Many clustering algorithms operate on a dissimilarity matrix. If the data are presented in the form of a data matrix, it can first be transformed into a dissimilarity matrix before applying such clustering algorithms.

a) Interval-Scaled Variables

"What are interval-scaled variables?" Interval-scaled variables are continuous measurements of a roughly linear scale. Typical examples include weight and height, latitude and longitude coordinates (e.g., when clustering houses), and weather temperature. The measurement unit used can affect the clustering analysis. For example, changing measurement units from meters to inches for height, or from kilograms to pounds for weight, may lead to a very different clustering structure. In general, expressing a variable in smaller units will lead to a larger range for that variable, and thus a larger effect on the resulting clustering structure.

How can the data for a variable be standardized?" To standardize measurements, one choice is to convert the original measurements to unit less variables. Given measurements for a variable f , this can be performed as follows. 1. Calculate the mean absolute deviation, 2. Calculate the standardized measurement, or z-score:

After standardization, or without standardization in certain applications, the dissimilarity (or similarity) between the objects described by interval-scaled variables is typically computed based on the distance between each pair of objects. The most popular distance measure is Euclidean distance, Another well-known metric is Manhattan (or city block) distance, Both the Euclidean distance and Manhattan distance satisfy the following mathematic requirements of a distance function:

1. d(i, j) _ 0: Distance is a nonnegative number.
2. d(i, i) = 0: The distance of an object to itself is 0.
3. d(i, j) = d(j, i): Distance is a symmetric function.

4. d(i, j) _ d(i, h)+d(h, j): Going directly from object i to object j in space is no more than making a detour over any other object h (triangular inequality).

b) Binary Variables

Let us see how to compute the dissimilarity between objects described by either symmetric or asymmetric binary variables.

A binary variable has only two states: 0 or 1, where 0 means that the variable is absent, and 1 means that it is present. Given the variable smoker describing a patient, for instance, 1 indicates that the patient smokes, while 0 indicates that the patient does not. Treating binary variables as if they are interval-scaled can lead to misleading clustering results. Therefore, methods specific to binary data are necessary for computing dissimilarities.

"What is the difference between symmetric and asymmetric binary variables?" A binary variable is

symmetric if both of its states are equally valuable and carry the same weight; that is, there is no preference on which outcome should be coded as 0 or 1. One such example could be the attribute gender having the states male and female. Dissimilarity that is based on symmetric binary variables is called symmetric binary dissimilarity. Its dissimilarity (or distance) measure, defined in Equation (7.9), can be used to assess the dissimilarity between objects i and j.

A binary variable is asymmetric if the outcomes of the states are not equally important, such as the positive and negative outcomes of a disease test. By convention, we shall code the most important outcome, which is usually the rarest one, by 1 (e.g., HIV positive) and the other by 0 (e.g., HIV negative). Given two asymmetric binary variables, the agreement of two 1s (a positive match) is then considered more significant than that of two 0s (a negative match). Therefore, such binary variables are often considered "monary" (as if having one state). The dissimilarity based on such variables is called asymmetric binary dissimilarity, where the number of negative matches, t, is considered unimportant and thus is ignored in the computation as shown below.

C) Categorical, Ordinal, and Ratio-Scaled Variables

1. Categorical Variables:

A categorical variable is a generalization of the binary variable in that it can take on more than two states.

For example, map color is a categorical variable that may have, say, five states: red, yellow, green, pink, and blue.

Let the number of states of a categorical variable be M. The states can be denoted by letters, symbols, or a set of integers, such as 1, 2, : : : , M.Notice that such integers are used just for data handling and do not represent any specific ordering.

"How is dissimilarity computed between objects described by categorical variables?" The dissimilarity between two objects i and j can be computed based on the ratio of mismatches:

where m is the number of matches (i.e., the number of variables for which i and j are in the same state), and p is the total number of variables. Weights can be assigned to increase the effect of m or to assign greater weight to the matches in variables having a larger number of states.

2. Ordinal Variables

A discrete ordinal variable resembles a categorical variable, except that the M states of the ordinal value are ordered in a meaningful sequence. Ordinal variables are very useful for registering subjective assessments of qualities that cannot be measured objectively. For example, professional ranks are often enumerated in a sequential order, such as assistant, associate, and full for professors. A continuous ordinal variable looks like a set of continuous data of an unknown scale; that is, the relative ordering of the values is essential but their actual magnitude is not. For example, the relative ranking in a particular sport (e.g., gold, silver, bronze) is often more essential than the actual values of a particular measure. Ordinal variables may also be obtained from the discretization of interval-scaled quantities by splitting the value range into a finite number of classes. The values ofan ordinal variable can be mapped to ranks. For example, suppose that an ordinal variable f has Mf states. These ordered states define the ranking 1, : : : , Mf

"How are ordinal variables handled?" The treatment of ordinal variables is quite similar to that of intervalscaled variables when computing the dissimilarity between objects. Suppose that f is a variable from a set of ordinal variables describing n objects. The dissimilarity computation with respect to f involves the following steps:

1. The value of f for the ith object is xi f , and f has Mf ordered states, representing theranking 1, : : : , Mf . Replace each xi f by its corresponding rank, ri f 2 f1, : : : , Mf g.
2. Since each ordinal variable can have a different number of states, it is often necessary to map the range of each variable onto [0.0,1.0] so that each variable has equal weight. This can be achieved by replacing the rank ri f of the ith object in the f th variable by
3. Dissimilarity can then be computed using any of the distance measures for interval-scaled variables, using zi f to represent the f value for the ith object.

3. Ratio-Scaled Variables

A ratio-scaled variable makes a positive measurement on a nonlinear scale, such as an exponential scale, approximately following the formula where A and B are positive constants, and t typically represents time. Common examples include the growth of a bacteria population or the decay of a radioactive element.

"How can I compute the dissimilarity between objects described by ratio-scaled variables?" There are three methods to handle ratio-scaled variables for computing the dissimilarity between objects.

• Treat ratio-scaled variables like interval-scaled variables. This, however, is not usually a good choice since it is likely that the scale may be distorted.

• Apply logarithmic transformation to a ratio-scaled variable f having value xi f for object i by using the formula yi f = log(xi f). The yi f values can be treated as interval valued, Notice that for some ratio scaled variables, log log or other transformations may be applied, depending on the variable's definition and the application.

- Treat xi f as continuous ordinal data and treat their ranks as interval-valued.

The latter two methods are the most effective, although the choice of method used may depend on the given application.

d) Variables of Mixed Types: To compute the dissimilarity between objects described by variables of the same type, where these types may be either interval-scaled, symmetric binary, asymmetric binary, categorical, ordinal, or ratio-scaled. However, in many real databases, objects are described by a mixture of variable types. In general, a database can contain all of the six variable types listed above. "So, how can we compute the dissimilarity between objects of mixed variable types?" One approach is to group each kind of variable together, performing a separate cluster analysis for each variable type. This is feasible if these analyses derive compatible results. However, in real applications, it is unlikely that a separate cluster analysis per variable type will generate compatible results. A more preferable approach is to process all variable types together, performing a single cluster analysis. One such technique combines the different variables into a single dissimilarity matrix, bringing all of the meaningful variables onto a common scale of the interval [0.0,1.0]. Suppose that the data set contains p variables of mixed type. The dissimilarity d(i, j) between objects i and j is defined as :

e) Vector Objects : In some applications, such as information retrieval, text document clustering, and biological taxonomy, we need to compare and cluster complex objects (such as documents) containing a large number of symbolic entities (such as keywords and phrases). To measure the distance between complex objects, it is often desirable to abandon traditional metric distance computation and introduce a nonmetric similarity function. There are several ways to define such a similarity function, s(x, y), to compare two vectors x and y. One popular way is to define the similarity function as a cosine measure as follows:

where xt is a transposition of vector x, jjxjj is the Euclidean normof vector x,1 jjyjj is the Euclidean norm of vector y, and s is essentially the cosine of the angle between vectors x and y. This value is invariant to rotation and dilation, but it is not invariant to translation and general linear transformation.

III.TYPES OF CLUSTERING METHODS

Many clustering algorithms exist in the literature. It is difficult to provide a crisp categorization of clustering methods because these categories may overlap, so that a method may have features from several categories.

Nevertheless, it is useful to present a relatively organized picture of the different clustering methods. In general, the major clustering methods can be classified into the following categories.

A. Hierarchical methods: A hierarchical method creates a hierarchical decomposition of the given set of data objects. A hierarchical method can be classified as being either agglomerative or divisive, based on how the hierarchical decomposition is formed. The agglomerative approach, also called the bottom-up approach, starts with each object forming a separate group. It successively merges the objects or groups that are close to one another, until all of the groups are merged into one (the topmost level of the hierarchy), or until a termination condition holds. The divisive approach, also called the top-down approach, starts with all of the objects in the same cluster. In each successive iteration, a cluster is split up into smaller clusters, until eventually each object is in one cluster, or until a termination condition holds.

B. Density-based methods: Most partitioning methods cluster objects based on the distance between objects. Such methods can find only spherical-shaped clusters and encounter difficulty at discovering clusters of arbitrary shapes. Other clustering methods have been developed based on the notion of density. Their general idea is to continue growing the given cluster as long as the density (number of objects or data points) in the "neighborhood" exceeds some threshold; that is, for each data point within a given cluster, the neighborhood of a given radius has to contain at least a minimum number of points. Such a method can be used to filter out noise (outliers) and discover clusters of arbitrary shape. DBSCAN and its extension, OPTICS, are typical density-based methods that grow clusters according to a density-based connectivity analysis. DENCLUE is a method that clusters objects based on the analysis of the value distributions of density functions.

C. Grid-based methods: Grid-based methods quantize the object space into a finite number of cells that form a grid structure. All of the clustering operations are performed on the grid structure (i.e., on the quantized space). The main advantage of this approach is its fast processing time, which is typically independent of the number of data objects and dependent only on the number of cells in each dimension in the quantized space. STING is a typical example of a grid-based method. Wave Cluster applies wavelet transformation for clustering analysis and is both grid-based and density-based. Grid based clustering methods

D. Model-based methods: Model-based methods hypothesize a model for each of the clusters and find the best fit of the data to the given model. A model-based algorithm may locate clusters by constructing a density function that reflects the spatial distribution of the data points. It also leads to a way of automatically determining the number of clusters based on standard statistics, taking "noise" or outliers into account and thus yielding robust clustering methods. Clustering high-dimensional data is a particularly important task in cluster analysis because many applications require the analysis of objects containing a large number of features or dimensions. For example, text documents may contain thousands of terms or keywords as features, and DNA microarray data may provide information on the expression levels of thousands of genes under hundreds of conditions. Clustering high-dimensional data is challenging due to the curse of dimensionality.

E. Constraint-based clustering is a clustering approach that performs clustering by incorporation of user-specified or application-oriented constraints. A constraint expresses a user's expectation or describes "properties" of the desired clustering results, and provides an effective means for communicating with the clustering process. Various kinds of constraints can be specified, either by a user or as per application requirements. Our focus of discussion will be on spatial clustering with the existence of obstacles and clustering under user-specified constraints. In addition, semisupervised clustering is described, which employs, for example, pairwise constraints (such as pairs of instances labeled as belonging to the same or different clusters) in order to improve the quality of the resulting clustering.

F. Partitioning Methods: Given a database of n objects or data tuples, a partitioning method constructs k partitions of the data, where each partition represents a cluster and k _ n. That is, it classifies the data into k groups, which together satisfy the following requirements: (1) each group must contain at least one object, and (2) each object must belong to exactly one group. Notice that the second requirement can be relaxed in some fuzzy partitioning techniques.Given k, the number of partitions to construct, a partitioning method creates an initial partitioning. It then uses an iterative relocation technique that attempts to improve the partitioning by moving objects from one group to another. The general criterion of a good partitioning is that objects in the same cluster are "close" or related to each other, whereas objects of different clusters are "far apart" or very different. There are various kinds of other criteria for judging the quality of partitions. Given D, a data set of n objects, and k, the number of clusters to form, a partitioning algorithm organizes the objects into k partitions (k _ n), where each partition represents a cluster. The clusters are formed to optimize an objective partitioning criterion, such as a dissimilarity function based on distance, so that the objects within a cluster are "similar," whereas the objects of different clusters are "dissimilar" in terms of the data set attributes.

Classical Partitioning Methods: k-Means and k-Medoids: The most well-known and commonly used partitioning methods are k-means, k-medoids, and their variations.

Example: The k-Means Method – A Centroid-Based Technique: The k-means algorithm takes the input parameter, k, and partitions a set of n objects into k clusters so that the resulting intracluster similarity is high but the intercluster similarity is low. Cluster similarity is measured in regard to the mean value of the objects in a cluster, which can be viewed as the cluster's centroid or center of gravity. In k-means clustering, initially it randomly selects k of the objects, each of which initially represents a cluster mean or center. For each of the remaining objects, an object is assigned to the cluster to which it is the most similar, based on the distance between the object and the cluster mean. It then computes the new mean for each cluster. This process iterates until the criterion function converges. Typically, the square-error criterion is used, defined as where E is the sum of the square error for all objects in the data set; p is the point in space representing a given object; and mi is the mean of cluster Ci (both p and mi are multidimensional). In other words, for each object in each cluster, the distance from the object to its cluster center is squared, and the distances are summed. This criterion tries to make the resulting k clusters as compact and as separate as possible.

IV. CONCLUSION

Although classification is an effective means for distinguishing groups or classes of objects, it requires the often costly collection and labeling of a large set of training tuples or patterns, which the classifier uses to model each group. It is often more desirable to proceed in the reverse direction: First partition the set of data into groups based on data similarity (e.g., using clustering), and then assign labels to the relatively small number of groups. Additional advantages of such a clustering-based process are that it is adaptable to changes and helps single out useful features that distinguish different groups.

REFERENCES

[1] Jain A, Dubes R (1988) Algorithms for clustering data. Prentice-Hall, Inc, Upper Saddle RiverMATH

[2] Xu R, Wunsch D (2005) Survey of clustering algorithms. IEEE Trans Neural Netw 16:645–678PubMedCrossRef

[3] Everitt B, Landau S, Leese M (2001) Clustering analysis, 4th edn. Arnold, LondonGoogle Scholar

[4] Gower J (1971) A general coefficient of similarity and some of its properties. Biometrics 27:857–871

[5] Estivill-Castro V (2002) Why so many clustering algorithms: a position paper. ACM SIGKDD Explor Newsl 4:65–75CrossRef

[6] Färber I, Günnemann S, Kriegel H, Kröger P, Müller E, Schubert E, Seidl T, Zimek A (2010) On using class-labels in evaluation of clusterings. In MultiClust: 1st international workshop on discovering, summarizing and using multiple clusterings held in conjunction with KDD, Washington, DC

[7] MacQueen J (1967) Some methods for classification and analysis of multivariate observations. Proc Fifth Berkeley Symp Math Stat Probab 1:281–297MathSciNet

[8] Park H, Jun C (2009) A simple and fast algorithm for K-medoids clustering. Expert Syst Appl 36:3336–3341CrossRef