DISCOVER MAXIMIZED FEATURE SPACE USING CORRELATION SIMILARITY APPROACH

B.Anuradha*1, S.G.Nawaz*2

M.Tech, Dept of CSE, SKDEC, Gooty, D.t: Anantapuram, A.P, India

M.Tech, Associate Professor, Dept of CSE, SKDEC, Gooty, D.t: Anantapuram, A.P, India

ABSTRACT:

Clustering is the unsupervised classification of patterns (observations, data items, or feature vectors) into groups (clusters). The clustering problem has been addressed in many contexts and by researchers in many disciplines; this reflects its broad appeal and usefulness as one of the steps in exploratory data analysis. However, clustering is a difficult problem combinatorial, and differences in assumptions and contexts in different communities has made the transfer of useful generic concepts and methodologies slow to occur. This paper presents an overview of pattern clustering methods from a statistical pattern recognition perspective, with a goal of providing useful advice and references to fundamental concepts accessible to the broad community of clustering practitioners. We present taxonomy of clustering techniques, and identify cross-cutting themes and recent advances. We also describe some important applications of clustering algorithms such as image segmentation, object recognition, and information retrieval. This paper introduces the fundamental concepts of unsupervised learning while it surveys the recent clustering algorithms. Moreover, recent advances in unsupervised learning, such as ensembles of clustering algorithms and distributed clustering, are described.

KEYWORDS: Cross Cutting Techniques, Clustering algorithms, Correlation, similarity function.

I.INTRODUCTION

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. 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. The variety of techniques for representing data, measuring proximity (similarity) between data elements, and grouping data elements has produced a rich and often confusing assortment of clustering methods. It is important to understand the difference between clustering (unsupervised classification) and discriminant analysis (supervised classification). In supervised classification, we are provided with a collection of *labeled* (preclassified) patterns; the problem is to label a newly encountered, yet unlabeled, pattern. Typically, the given labeled (*training*) patterns are used to learn the descriptions of classes which in turn are used to label a new pattern. In the case of clustering, the problem is to group a given collection of unlabeled patterns into meaningful clusters. In a sense, labels are associated with clusters also, but these category labels are *data driven*; that is, they are obtained solely from the data. Clustering is useful in several exploratory pattern-analysis, grouping, decision- making, and machine-learning situations; including data mining, document retrieval, image segmentation, and pattern classification. However, in many such problems, there is little prior information (e.g., statistical models) available about the data, and the decision-maker must make as few assumptions about the data as possible. It is under these restrictions that clustering methodology is particularly appropriate for the exploration of interrelationships among the data points to make an assessment (perhaps preliminary) of their structure. 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 contexts in which clustering are used. Thus, we face a dilemma regarding the scope of this survey. The production of a truly comprehensive survey would be a monumental task given the sheer mass of literature in this area.

The accessibility of the survey might also be questionable given the need to reconcile very different vocabularies and assumptions regarding clustering in the various communities. Cluster analysis is an unsupervised learning method that constitutes a cornerstone of an intelligent data analysis process. It is used for the exploration of inter-relationships among a collection of patterns, by organizing them into homogeneous clusters. It is called unsupervised learning because unlike classification (known as supervised learning), no a priori labeling of some patterns is available to use in categorizing others and inferring the cluster structure of the whole data. Intra-connectivity is a measure of the density of connections between the instances of a single cluster. A high intra-connectivity indicates a good clustering arrangement because the instances grouped within the same cluster are highly dependent on each other. Inter-connectivity is a measure of the connectivity between distinct clusters. A low degree of interconnectivity is desirable because it indicates that individual clusters are largely independent of each other. Every instance in the data set is represented using the same set of attributes. The attributes are continuous, categorical or binary.

II.RELATED WORK

2.1 Partitioning Methods

Partitioning methods are divided into two major subcategories, the centroid and the medoids algorithms. The centroid algorithms represent each cluster by using the gravity centre of the instances. The medoid algorithms represent each cluster by means of the instances closest to the gravity centre. The most well-known centroid algorithm is the k-means. The k-means method partitions the data set into k subsets such that all points in a given subset are closest to the same centre. In detail, it randomly selects k of the instances to represent the clusters. Based on the selected attributes, all remaining instances are assigned to their closer centre. K-means then computes the new centers by taking the mean of all data points belonging to the same cluster. The operation is iterated until there is no change in the gravity centers. If k cannot be known ahead of time, various values of k can be evaluated until the most suitable one is found. The effectiveness of this method as well as of others relies heavily on the objective function used in measuring the distance between instances. The difficulty is in finding a distance measure that works well with all types of data. There are several approaches to define the distance between instances. Generally, the k-means algorithm has the following important properties: 1. It is efficient in processing large data sets, 2. It often terminates at a local optimum, 3. The clusters have spherical shapes, 4. It is sensitive to noise. The algorithm described above is classified as a batch method because it requires that all the data should be available in advance. However, there are variants of the k-means clustering process, which gets around this limitation

.Choosing the proper initial centroids is the key step of the basic K-means procedure. The k-modes algorithm is a recent partitioning algorithm and uses the simple matching coefficient measure to deal with categorical attributes. The k-prototypes algorithm, through the definition of a combined dissimilarity measure, further integrates the k-means and k-modes algorithms to allow for clustering instances described by mixed attributes. More recently, in [6] another generalization of conventional k-means clustering algorithm has been presented. This new one applicable to ellipse-shaped data clusters as well as ball-shaped ones without dead-unit problem, but also performs correct clustering without pre-determining the exact cluster number.

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 with

every cluster using a membership function. Larger membership values indicate higher confidence in the assignment of the pattern to the cluster. One widely used algorithm is the Fuzzy C-Means (FCM) algorithm, which is based on k-means. FCM attempts to find the most characteristic point in each cluster, which can be considered as the "center" of the cluster and, then, the grade of membership for each instance in the clusters. Other soft clustering algorithms have been developed and most of them are based on the Expectation-Maximization (EM) algorithm. They assume an underlying probability model with parameters that describe the probability that an instance belongs to a certain cluster. The strategy in this algorithm is to start with initial guesses for the mixture model parameters. These values are then used to calculate the cluster probabilities for each instance. These probabilities are in turn used to re-estimate the parameters, and the process is repeated. A drawback of such algorithms is that they tend to be computationally expensive. Another problem found in the previous approach is called over fitting. This problem might be caused by two reasons. On one hand, a large number of clusters may be specified. And on the other, the distributions of probabilities have too many parameters. In this context, one possible solution is to adopt a fully Bayesian approach, in which every parameter has a prior probability distribution. Hierarchical algorithms that create a hierarchical decomposition of the instances are covered in the following section.

III.PROPOSED SYSTEM EVOLUTION

3.1 Hierarchical Clustering

The hierarchical methods group data instances into a tree of clusters. There are two major methods under this category. One is the agglomerative method, which forms the clusters in a bottom-up fashion until all data instances belong to the same cluster. The other is the divisive method, which splits up the data set into smaller cluster in a top-down fashion until each cluster contains only one instance. Both divisive algorithms and agglomerative algorithms can be represented by dendrograms. Both agglomerative and divisive methods are known for their quick termination. However, both methods suffer from their inability to perform adjustments once the splitting or merging decision is made. Other advantages are: 1) does not require the number of clusters to be known in advance, 2) computes a complete hierarchy of clusters, 3) good result visualizations are integrated into the methods, 4) a "flat" partition can be derived afterwards (e.g. via a cut through the dendrogram). Hierarchical clustering techniques use various criteria to decide "locally" at each step which clusters should be joined (or split for divisive approaches). For agglomerative hierarchical techniques, the criterion is typically to merge the "closest" pair of clusters, where "close" is defined by a specified measure of cluster proximity. There are three definitions of the closeness between two clusters: single-link, complete-link and average-link. The single-link similarity between two clusters is the similarity between the two most similar instances, one of which appears in each cluster. Single link is good at handling non-elliptical shapes, but is sensitive to noise and outliers. The complete-link similarity is the similarity between the two most dissimilar instances, one from each cluster. Complete link is less susceptible to noise and outliers, but can break large clusters, and has trouble with convex shapes. The average-link similarity is a compromise between the two.

IV.Density-based Clustering

Density-based clustering algorithms try to find clusters based on density of data points in a region. The key idea of density-based clustering is that for each instance of a cluster the neighborhood of a given radius (*Eps*) has to contain at least a minimum number of instances (*MinPts)*. One of the most well known density-based clustering algorithms is the DBSCAN [9]. DBSCAN separate data points into three classes:

• Core points. These are points that are at the interior of a cluster. A point is an interior point if there are enough points in its neighborhood.

• Border points. A border point is a point that is not a core point, i.e., there are not enough points in its neighborhood, but it falls within the neighborhood of a core point.

• Noise points. A noise point is any point that is not a core point or a border point. To find a cluster, DBSCAN starts with an arbitrary instance (*p*) in data set (*D*) and retrieves all instances of *D* with respect to *Eps* and *MinPts*. The algorithm makes use of a spatial data structure - R*tree – to locate points within Eps distance from the core points of the clusters. An incremental version of DBSCAN (incremental DBSCAN) is presented in [10]. It was proven that this incremental algorithm yields the same result as DBSCAN. In addition, another clustering algorithm (GDBSCAN) generalizing the density-based algorithm DBSCAN is presented in. GDBSCAN can cluster point instances to both, their numerical and their categorical attributes. Moreover, in the PDBSCAN, a parallel version of DBSCAN is presented. Furthermore, DBCLASD (Distribution Based Clustering of Large Spatial Data sets) eliminates the need for MinPts and Eps parameters. DBCLASD incrementally augments an initial cluster by its neighboring points as long as the nearest neighbor distance set of the resulting cluster still fits the expected distance distribution. While the distance set of the whole cluster might fit the expected distance distribution, this does not necessarily hold for all subsets of this cluster. Thus, the order of testing the candidates is crucial. In [2] a new algorithm (OPTICS) is introduced, which creates an *ordering* of the data set representing its density-based clustering structure. It is a versatile basis for interactive cluster analysis. Another density-based algorithm is the DENCLUE . The basic idea of DENCLUE is to model the overall point density analytically as the sum of influence functions of the data points. The influence function can be seen as a function, which describes the impact of a data point within its neighborhood. Then, by determining the maximum of the overall density function can identify clusters. The algorithm allows a compact mathematical description of arbitrarily shaped clusters in high-dimensional data sets and is significantly faster than the other density based clustering algorithms. Moreover, DENCLUE produces good clustering results even when a large amount of noise is present. As in most other approaches, the quality of the resulting clustering depends on an adequate choice of the parameters. In this approach, there are two important parameters, namely σ and ξ. The parameter σ determines the influence of a point in its neighborhood and ξ describes whether a density-attractor is significant. Density-attractors are local maxima of the overall density function. FDC algorithm (Fast Density-Based Clustering) is presented in for density-based clustering defined by the density-linked relationship. The clustering in this algorithm is defined by an equivalence relationship on the objects in the database. The complexity of FDC is linear to the size of the database, which is much faster than that of the algorithm DBSCAN. More recently, the algorithm SNN (Shared Nearest Neighbors) [8] blends a density based approach with the idea of ROCK. SNN sparsifies similarity matrix by only keeping K-nearest neighbors, and thus derives the total strength of links for each x.

V.Grid-based Clustering

Grid-based clustering algorithms first quantize the clustering space into a finite number of cells (hyper-rectangles) and then perform the required operations on the quantized space. Cells that contain more than certain number of points are treated as dense and the dense cells are connected to form the clusters. Some of the grid-based clustering algorithms are: Statistical Information Grid-based method STING first divides the spatial area into several levels of rectangular cells in order to form a hierarchical structure. The cells in a high level are composed from the cells in the lower level. It generates a hierarchical structure of the grid cells so as to represent the clustering information at different levels. Although STING generates good clustering results in a short running time, there are two major problems with this algorithm. Firstly, the performance of STING relies on the granularity of the lowest level of the grid structure. Secondly, the resulting clusters are all bounded horizontally or

vertically, but never diagonally. This shortcoming might greatly affect the cluster quality.

VI.Model based Methods Auto Class uses the Bayesian approach, starting from a random initialization of the parameters, incrementally adjusts them in an attempt to find their maximum likelihood estimates. Moreover, in it is assumed that, in addition to the observed or predictive attributes, there is a hidden variable. This unobserved variable reflects the cluster membership for every case in the data set. Therefore, the data-clustering problem is also an example of supervised learning from incomplete data due to the existence of such a hidden variable. Their approach for learning has been called RBMNs (Recursive Bayesian Multi-nets).

VII.Ensembles of Clustering Algorithms

The theoretical foundation of combining multiple clustering algorithms is still in its early stages. In fact, combining multiple clustering algorithms is a more challenging problem than combining multiple classifiers. In the reason that impede the study of clustering combination has been identified as various clustering algorithms produce largely different results due to different clustering criteria, combining the clustering results directly with integration rules, such as sum, product, median and majority vote can not generate a good meaningful result. Cluster ensembles can be formed in a number of different ways, such as (1) the use of a number of different clustering techniques (either deliberately or arbitrarily selected). (2) The use of a single technique many times with different initial conditions. (3) The use of different partial subsets of features or patterns.In a split-and-merge strategy is followed. The first step is to decompose complex data into small, compact clusters. The K-means algorithm serves this purpose; an ensemble of clustering algorithms is produced by random initializations of cluster centroids. Data partitions present in these clustering's are mapped into a new similarity matrix between patterns, based on a voting mechanism. This matrix, which is independent of data sparseness, is then used to extract the natural clusters using the single link algorithm.

VIII.CONCLUSION

We should remark that the list of references is not a comprehensive list of papers discussing unsupervised methods: our aim was to produce a critical review of the key ideas, rather than a simple list of all publications which had discussed or made use of those ideas. Despite this, we hope that the references cited cover the major theoretical issues, and provide routes into the main branches of the literature dealing with such methods. Generally, we will say that partitioning algorithms typically represent clusters by a prototype. An iterative control strategy is used to optimize the whole clustering such that, e.g., the average or squared distances of instances to its prototypes are minimized. Consequently, these clustering algorithms are effective in determining a good clustering if the clusters are of convex shape, similar size and density, and if the number of clusters can be reasonably estimated

IX.REFERENCES

[1] Agrawal R., Gehrke J., Gunopulos D. and Raghavan P. (1998). Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications. In Proc. of the 1998

ACM-SIGMOD Conf. On the Management of Data, 94-105.

[2] Ankerst M., Breunig M., Kriegel H., Sander J., OPTICS: Ordering Points to Identify the Clustering Structure, Proc. ACM SIGMOD'99 Int. Conf. on Management of Data.

[3] H. Ayad and M. Kamel. Finding natural clusters using multi-clusterer combiner based on shared nearest neighbors. In Multiple Classifier Systems: Fourth International Workshop, MCS

2003, Guildford, Surrey, United Kingdom, June 11–13.

[4] M. Breunig, H.-P. Kriegel, R. Ng, and J. Sander. Lof: Identifying density-based local outliers. In Proc. of SIGMOD'2000, pages 93–104, 2000.

[5] Cheeseman P. & Stutz J., (1996), Bayesian Classification (AutoClass): Theory and Results,

In U. M. Fayyad, G. Piatetsky-Shapiro, P. mSmith, and R. Uthurusamy, editors, Advances in

Knowledge Discovery and Data Mining, pages 153-180, AAAI/MIT Press.

[6] Yiu-Ming Cheung, k*-Means: A new generalized k-means clustering algorithm,

Pattern Recognition Letters 24 (2003) 2883–2893.

[7] Chien-Yu Chen, Shien-Ching Hwang, and Yen-Jen Oyang, An Incremental Hierarchical Data Clustering Algorithm Based on Gravity Theory gravity theory, Advances in Knowledge Discovery and Data Mining: 6th Pacific-Asia Conference, PAKDD 2002, Taipel, Taiwan, May 6-8, 2002.Springer-Verlag LNCS 2336.

[8] L. Ertöz, M. Steinbach, and V. Kumar. Finding clusters of different sizes, shapes, and densities in noisy, high dimensional data. In Proceedings of Second SIAM International Conference on Data Mining, San Francisco, CA, USA, May 2003.

[9] Ester, M., Kriegel, H.-P., Sander, J., and Xu X. (1996), A density-based algorithm for discovering clusters in large spatial data sets with noise. Proc. 2nd Int. Conf. on Knowledge Discovery and Data Mining. Portland, OR, pp. 226–231.

[10] Ester, M., Kriegel, H.-P., Sander, J., Wimmer M. and Xu X. (1998), Incremental Clustering for Mining in a Data Warehousing Environment, Proceedings of the 24th VLDB Conference New York, USA, 1998.

B.Anuradha M.Tech,
SRI KRISHNA DEVARAYA ENGINEERING COLLEGE,
JNTU, Ananthapuram.

S.G.Nawaz M.Tech, Associate Professor, SRI KRISHNA DEVARAYA ENGINEERING COLLEGE,
JNTU, Ananthapuram.