International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 2 Issue 11, November - 2013

A Review : Study of Various Clustering Techniques

Miss. Priti K. Doad
Department of CSE,
G.H.Raisoni College, Amravati

Mr. Mahip M. Bartere
Department of CSE,
G.H.Raisoni College, Amravati

Abstract

Retrieval of information from the databases is now a day's significant issues. The thrust of information for decision making is challenging one. To overcome this problem, different techniques have been developed for this purpose. One of techniques is clustering. Clustering is a significant task in data analysis and data mining applications. It is the task of arrangement a set of objects so that objects in the identical group are more related to each other than to those in other groups (clusters).The clustering is unsupervised learning. In this paper we propose a methodology for comparing clustering methods based on the quality of the result and the performance of the execution. The quality of a clustering result depends on both the similarity measure used by the method and its implementation. Clustering has been widely used as a segmentation approach therefore, choosing an appropriate clustering method is very critical to achieve better results. A good clustering method will produce high superiority clusters with high intra-class similarity and low inter-class similarity. There are different types of Clustering algorithms partition-based algorithms such as K-Means, KNN, density-based algorithms and SSC-EA-based algorithms. Partitioning clustering algorithm splits the data points into k partition, where each partition represents a cluster. Density based algorithms find the cluster according to the regions which grow with high density. It is the one-scan algorithms.

1. Introduction

Clustering is a data mining technique of grouping set of data objects into multiple groups or clusters so that objects within the cluster have high similarity, but are very dissimilar to objects in the other clusters. Dissimilarities and similarities are assessed based on the attribute values describing the objects. Clustering algorithms are used to organize data, categorize data, for data compression and model construction, for detection of outliers etc. Common approach for all clustering techniques is to find clusters centre that will represent each cluster. Cluster centre will represent with input vector can tell which cluster this vector belong to by measuring a similarity metric between input vector and all cluster centre and determining which cluster is nearest or most similar one [1].

A good clustering method will produce high superiority clusters with high intra-class similarity and low inter-class similarity. The superiority of a clustering result depends on equally the similarity measure used by the method and its implementation. The superiority of a clustering technique is also calculated by its ability to find out some or all of the hidden patterns.

Clustering is a data mining technique of grouping set of data objects into multiple groups or clusters so that objects within the cluster have high similarity, but are very dissimilar to objects in the other clusters. Dissimilarities and similarities are assessed based on the attribute values describing the objects. Clustering algorithms are used to organize data, categorize data, for data compression and model construction, for detection of outliers etc. Common approach for all clustering techniques is to find clusters centre that will represent each cluster. Cluster centre will represent with input vector can tell which cluster this vector belong to by measuring a similarity metric between input vector and all cluster centre and determining which cluster is nearest or most similar one[1].

A good clustering method will produce high superiority clusters with high intra-class similarity and low inter-class similarity. The superiority of a clustering result depends on equally the similarity measure used by the method and its implementation. The superiority of a clustering technique is also calculated by its ability to find out some or all of the hidden patterns.

Cluster analysis can be used as a standalone data mining tool to gain insight into the data distribution, or as a pre-processing step for other data mining algorithms operating on the detected clusters. Many clustering algorithms have been developed and are categorized from several aspects such as partitioning methods such as K-means,KNN, density-based methods, and SSC-EA-based methods. Further data set can be numeric or categorical. Inherent geometric properties of numeric data can be exploited to naturally define distance function between data points. Whereas categorical data can be derived from either quantitative or qualitative data where observations are directly observed from counts.

2. Various Clustering Techniques

2.1. K-Means Clustering

K-means is one of the simplest unsupervised learning algorithms that solve the well-known clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters) fixed a priori. The main idea is to define k centroids, one for each cluster. These centroids should be placed in a cunning way because of different location causes different result[6]. So, the better choice is to place them as much as possible far away from each other. The next step is to take each point belonging to a given data set and associate it to the nearest centroid. When no point is pending, the first step is completed and an early group age is done. At this point we need to recalculate k new centroids as bary centers of the clusters resulting from the previous step. After we have these k new centroids, a new binding has to be done between the same data set points and the nearest new centroid. A loop has been generated. As a result of this loop we may notice that the k centroids change their location step by step until no more changes are done. In other words centroids do not move any more. Finally, this algorithm aims at minimizing an objective function.

The algorithm is composed of the following steps:

1. Place K points into the space represented by the objects that are being clustered. These points represent initial group centroids.
2. Assign each object to the group that has the closest centroid.
3. When all objects have been assigned, recalculate the positions of the K centroids.
4. Repeat Steps 2 and 3 until the centroids no longer move. This produces a separation of the objects into groups from which the metric to be minimized can be calculated.

Advantages

1. K-means is a simple algorithm that has been adapted to many problem domains.
2. More automated than manual thresholding of an image.
3. It is a good candidate for extension to work with fuzzy feature vectors.

Disadvantages

1. Although it can be proved that the procedure will always terminate, the k-means algorithm does not necessarily find the most optimal configuration, corresponding to the global objective function minimum.
2. The algorithm is also significantly sensitive to the initial randomly selected cluster centers. The k means algorithm can be run multiple times to reduce this effect.

2.2. *K*-nearest Neighbour's Algorithm (*k*-NN)

In pattern recognition, the ***k*-nearest neighbors algorithm** (*k*-NN) is a non-parametric method for classification and regression,[1] that predicts objects' "values" or class memberships based on the *k* closest training examples in the feature space. *k*-NN is a type of instance-based learning, or lazy learning where the function is only approximated locally and all computation is deferred until classification. The *k*-nearest neighbor algorithm is amongst the simplest of all machine learning algorithms: an object is classified by a majority vote of its neighbors, with the object being assigned to the class most common amongst its *k* nearest neighbors (*k* is a positive integer, typically small). If $k = 1$, then the object is simply assigned to the class of that single nearest neighbor.The same method can be used for regression, by simply assigning the property value for the object to be the average of the values of its *k* nearest neighbors. It can be useful to weight the contributions of the neighbors, so that the nearer neighbors contribute more to the average than the more distant ones. (A common weighting scheme is to give each neighbor a weight of $1/d$, where d is the

distance to the neighbor. This scheme is a generalization of linear interpolation.)

The neighbors are taken from a set of objects for which the correct classification (or, in the case of regression, the value of the property) is known. This can be thought of as the training set for the algorithm, though no explicit training step is required. The *k*-nearest neighbor algorithm is sensitive to the local structure of the data. Nearest neighbor rules in effect implicitly compute the decision boundary. It is also possible to compute the decision boundary explicitly, and to do so efficiently, so that the computational complexity is a function of the boundary complexity[2].

Advantages

KNN is a special case of a variable-bandwidth, kernel density "balloon" estimator with a uniform kernel. The naive version of the algorithm is easy to implement by computing the distances from the test example to all stored examples[7][8], but it is computationally intensive for large training sets. Using an appropriate nearest neighbor search algorithm makes *K*NN computationally tractable even for large data sets. Many nearest neighbor search algorithms have been proposed over the years; these generally seek to reduce the number of distance evaluations actually performed.

*K*NN has some strong consistency results. As the amount of data approaches infinity, the algorithm is guaranteed to yield an error rate no worse than twice the Bayes error rate (the minimum achievable error rate given the distribution of the data)[9] KNN is guaranteed to approach the Bayes error rate for some value of *k* (where *k* increases as a function of the number of data points). Various improvements to *k*NN are possible by using proximity graphs[10].

Disadvantages

A drawback of the basic "majority voting" classification occurs when the class distribution is skewed. That is, examples of a more frequent class tend to dominate the prediction of the new example, because they tend to be common among the *k* nearest neighbors due to their large number. One way to overcome this problem is to weight the classification, taking into account the distance from the test point to each of its *k* nearest neighbors[3]. The class (or value, in regression problems) of each of the k nearest points is multiplied by a weight proportional to the inverse of the distance from that point to the test point. Another way to overcome skew is by abstraction in data representation.

2.3. Density-Based Clustering

In density-based clustering, clusters are defined as areas of higher density than the remainder of the data set[7]. Objects in these sparse areas - that are required to separate clusters - are usually considered to be noise and border points. Density-based spatial clustering of applications with noise (DBSCAN) is a data clustering algorithm proposed by Martin Ester,Hans-Peter Kriegel, Jörg Sander and Xiaowei Xu in 1996. It is a density-based clustering algorithm because it finds a number of clusters starting from the estimated density distribution of corresponding nodes[1]. DBSCAN is one of the most common clustering algorithms and also most cited in scientific literature. OPTICS can be seen as a generalization of DBSCAN to multiple ranges, effectively replacing the ε parameter with a maximum search radius[2].

The most popular[8] density based clustering method is DBSCAN. In contrast to many newer methods, it features a well-defined cluster model called "density-reachability"[9]. Similar to linkage based clustering; it is based on connecting points within certain distance thresholds. However, it only connects points that satisfy a density criterion, in the original variant defined as a minimum number of other objects within this radius. A cluster consists of all density-connected objects (which can form a cluster of an arbitrary shape, in contrast to many other methods) plus all objects that are within these objects' range. Another interesting property of DBSCAN is that its complexity is fairly low - it requires a linear number of range queries on the database - and that it will discover essentially the same results (it is deterministic for core and noise points, but not for border points) in each run, therefore there is no need to run it multiple times. OPTICS[10] is a generalization of DBSCAN that removes the need to choose an appropriate value for the range parameter ε, and produces a hierarchical result related to that of linkage clustering. DeLi-Clu, Density-Link-Clustering[11] combines ideas from single-linkage clustering and OPTICS, eliminating the ε parameter entirely and offering performance improvements over OPTICS by using an R-tree index.

The key drawback of DBSCAN and OPTICS is that they expect some kind of density drop to detect cluster borders. Moreover, they cannot detect intrinsic cluster structures which are prevalent in the majority of real life data. A variation of DBSCAN, EnDBSCAN, efficiently detects such kinds of structures[12]. On data sets with, for example, overlapping Gaussian distributions - a common use case in artificial data - the cluster borders produced by these algorithms will often

look arbitrary, because the cluster density decreases continuously. On a data set consisting of mixtures of Gaussians, these algorithms are nearly always outperformed by methods such as EM clustering that are able to precisely model this kind of data.

Advantages

1. DBSCAN does not require one to specify the number of clusters in the data a priori, as opposed to k-means.
2. DBSCAN can find arbitrarily shaped clusters. It can even find a cluster completely surrounded by (but not connected to) a different cluster. Due to the MinPts parameter, the so-called single-link effect (different clusters being connected by a thin line of points) is reduced.
3. DBSCAN has a notion of noise.

Disadvantages

1. The quality of DBSCAN depends on the distance measure used in the function regionQuery(P,ε). The most common distance metric used is Euclidean distance. Especially for high-dimensional data, this metric can be rendered almost useless due to the so-called "Curse of dimensionality", making it difficult to find an appropriate value for ε. This effect, however, is also present in any other algorithm based on Euclidean distance.
2. DBSCAN cannot cluster data sets well with large differences in densities, since the minPts-ε combination cannot then be chosen appropriately for all clusters.

2.4. SSC-EA-Based Classifier

The SSC-EA-based algorithm performs clustering in N=m(m– 1)/2 low-dimensional subspaces Xi ∈R2.As we have shown, this provides a high discrimination power to detect and characterize different types of network attacks. However, the multiple clustering computation increases the total computational time (CT) of the algorithm, imposing scalability issues for online detection of network attacks in very-high speed networks. Scalability should be addressed regarding both the number of features used to describe traffic flows (m)and the number of flows to analyze (n). In the real traffic evaluations we have presented, the number of flows captured in a time slot of ΔT= 20 s rounds n= 2500 flows. For the m= 9 features we have used, the total number of clustering to compute is N= 36, which takes about 14.4 s in a standard single-processor machine. Two key features of the SSC-EA-based algorithm can be exploited to reduce scalability problems in m and n. First, clustering is performed in low-dimensional sub-spaces (R2), independently of the number of features that are used. Clustering in low-dimensional feature spaces is faster than in high dimensional spaces [12], which partially alleviate the overhead of multiple clustering computations. Second, the clustering of each subspace Xi can be performed independent of the analysis on the other subspaces, which is perfectly adapted for parallel computing architectures. Parallel computing has become the dominant paradigm for accelerating specific tasks and represents a booming domain, driven by the availability of strong-computational-power entities at low cost. Parallelization can be achieved by using various techniques: multiprocessor and multicore machines, GPU capabilities, network processor cards, distributed computing frameworks, or combining these techniques. We shall use the term slice as a reference to a single computational entity.

3. Conclusion

To conclude this paper; we have present a review of different type of Classification Technique. In this study, the basic concept of clustering & clustering technique are given. The processes of grouping a set of physical or abstract object into classes of similar objects are named as clustering. Clustering is a significant task in data analysis and data mining applications. There are different types of Clustering algorithms partition-based algorithms such as K-Means, KNN, density-based algorithms and SSC-EA-based algorithms. Partitioning clustering algorithm splits the data points into k partition, where each partition represents a cluster. Density based algorithms find the cluster according to the regions which grow with high density. It is the one-scan algorithms.

4. Referances

[1] Manish Verma, Mauly Srivastava, Neha Chack, Atul Kumar Diswar, Nidhi Gupta," A Comparative Study of Various Clustering Algorithms in Data Mining," International Journal of Engineering Reserch and Applications (IJERA), Vol. 2, Issue 3, pp.1379-1384, 2012.

[2] Miao Xie, Jiankun Hu, Member, IEEE, Song Han, Member, IEEE, and Hsiao-Hwa Chen, Fellow, IEEE "Scalable Hyper grid k-NN-Based Online Anomaly Detection in Wireless Sensor Networks"IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 24, NO. 8, AUGUST 2013 1661

[3] Alexander G. Tartakovsky, Senior Member, IEEE, Aleksey S. Polunchenko, and Grigory Sokolov"

Efficient Computer Network Anomaly Detection by Changepoint Detection Methods" IEEE JOURNAL OF SELECTED TOPICS IN SIGNAL PROCESSING, VOL. 7, NO. 1, FEBRUARY 2013.

[4] Pedro Casas, Johan Mazel, and Philippe Owezarski, CNRS and Université de Toulouse," *Knowledge-Independent Traffic Monitoring: Unsupervised Detection of Network Attacks*", IEEE Network January/February 2012

[5] Pedro Casas , Johan Mazel, and Philippe Owezarski *," Steps Towards Autonomous Network Security: Unsupervised Detection of Network Attack*s *",* IEEE 2011 .

[6] "A Review of Anomaly based Intrusion Detection Systems "International Journal of Computer Applications (0975 – 8887) Volume 28– No.7, August 2011

[7] A. K. Jain, "Data Clustering: 50 Years Beyond K-Means," Pattern Recognition Letters, vol. 31, no. 8, 2010, pp. 651–66.

[8] Anna Sperotto, Gregor Schaffrath, Ramin Sadre, Cristian Morariu, Aiko Pras and Burkhard Stiller," *An Overview of IP Flow-Based Intrusion Detection*", IEEE COMMUNICATIONS SURVEYS & TUTORIALS, VOL. 12, NO. 3, THIRD QUARTER 2010.

[9] Jiong Zhang and Mohammad Zulkernine, "Anomaly Based Network Intrusion Detection with Unsupervised Outlier Detection" IEEE International Conference on Communications, 2006.

[10] Rui Xu, Student Member, IEEE and Donald Wunsch II, Fellow, IEEE, *"Survey of Clustering Algorithms"*, IEEE TRANSACTIONS ON NEURAL NETWORKS, VOL. 16, NO. 3, MAY 2005

[11] S. Hansman, R. Hunt "*A Taxonomy of Network and Computer Attacks*", in Computers and Security, vol. 24 (1), pp. 31-43, 2005

[12] Fred and A. K. Jain, "*Combining Multiple Clustering Using Evidence Accumulation*", in IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 27 (6), pp. 835-850, 2005.

[13] Augustin Soule, Kav´ e SalamatianM. and Nina Taft, "*Combining Filtering and Statistical Methods for Anomaly Detection*", internet Measurement Conference 2005 .