Scrutiny of large datasets using clustering techniques

K. Pushpavathi and K. Gayathiri

Department of Faculty of Computer Applications, AVIT (An Engineering Institute) Chennai, India

Corresponding author: spushpavathi@rediffmail.com

Abstract

Clustering technique is critically important step in data mining process. 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). Data mining can do by passing through various phases. Mining can be done by using supervised and unsupervised learning. The clustering is unsupervised learning. A good clustering method will produce high superiority clusters with high intra-class similarity and low inter-class similarity. Its main distinctiveness is the fastest processing time. In this paper, an analysis of clustering and its different techniques in data mining is done. Results were quite encouraging and had shown high accuracy.

Keywords: data mining, clustering, Clustering techniques, clustered instances, weka tool.

Highlight Clustering is a statistical technique much similar to classification. It sorts raw data into meaningful clusters and groups of relatively homogeneous observations. The objects of a particular cluster have similar characteristics and properties but differ with those of other clusters. The grouping is accomplished by finding similarities among data according to characteristics found in raw data [1]. The main objective was to find optimum number of clusters. There are two basic types of clustering methods, hierarchical and non-hierarchical. Clustering process is not one time task but is continuous and an iterative process of knowledge discovery from huge quantities of raw and unorganized data [2]. For a particular classification problem, an appropriate clustering algorithm and parameters must be selected for obtaining optimum results. [3]. Clustering is a type of explorative data mining used in many application oriented areas such as machine learning, classification and pattern recognition [4].

For clustering method, the most important property is that a*tuple* of particular cluster is more likely to be similar to the other *tuples* within the same cluster than the *tuples* of other clusters. For classification, the similarity measure is defined as *sim* (t_i, t_j), between any two *tuples*, t_i , $t_j \in D$. For a given cluster, K_m of N points $\{t_{ml}, t_{m2} \ldots t_{mN}\}$, the centroid is defined as the *middle* of the cluster. The radius is the square root

of the average mean squared distance from any point in the cluster to the centroid. For given clusters K_i and K_j, there are several ways to determine the distance between the clusters. If clusters are represented by centroids, the distance between two clusters is the distance between their respective centroids. We thus have, $dis\ (K_i, K_j) = dis\ (C_i, C_j)$, where C_i and C_j are the centroid for K_i and K_j respectively.

Clustering Techniques

Clustering is a major task in data analysis and data mining applications. The superiority of a clustering technique is also calculated by its ability to find out some or all of the hidden patterns. In data mining, there are some requirements for clustering the data. These requirements are Scalability, Ability to deal with different types of attributes, Ability to handle dynamic data, Discovery of clusters with arbitrary shape, Minimal requirements for domain knowledge to determine input parameters, Able to deal with noise and outliers, Insensitive to order of input records, High dimensionality, Incorporation of user-specified constraints, Interpretability and usability. The types of data that are used for analysis of clustering are Interval-scaled variables, Binary variables, Nominal, ordinal, and ratio variables, Variables of mixed types [17]. The five types of clusters are used in clustering. The clusters are divided into these types according to their characteristics. The types of clusters are Well-separated clusters, Center-based clusters Contiguous clusters, Density-based clusters and Shared Property or Conceptual Clusters. Many applications of clustering are characterized by high dimensional data where each object is described by hundreds or thousands of attributes. The challenge in high dimensional is the curse of dimensionality faced by high dimensional data clustering algorithms, basically means the distance measures become gradually more worthless as the number of dimensions increases in the data set. Clustering has an extensive and prosperous record in a range of scientific fields in the vein of image segmentation, information retrieval and web data mining.

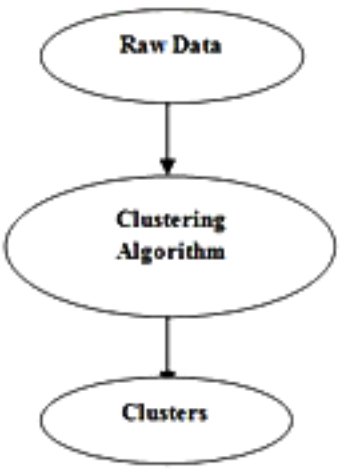


Fig. 1: Stages of Clustering

K-Means Clustering Technique

The algorithm is called *k*-means due to the fact that the letter *k* represents the number of clusters chosen. An observation is assigned to a particular cluster for which its distance to the cluster mean is the smallest. The principal function of algorithm involves finding the *k*-means. First, an initial set of means is defined and then subsequent classification is based on their distances to the centres [6]. Next, the clusters' mean is computed again and then reclassification is done based on the new set of means. This is repeated until cluster means don't change much between successive iterations [7]. Finally, the means of the clusters once again calculated and then all the cases are assigned to the permanent clusters.

Given a set of observations ($x_1, x_2, ..., x_n$), where each observation x_i is a *d*-dimensional real vector. The *k*-means clustering algorithm aims to partition the *n* observations into *k* groups of observations called clusters where $k \leq n$, so as to minimize the sum of squares of distances between observations within a particular cluster [8].

K-Means Algorithm

Simplified simulation flow of *k*-means algorithm
Begin
Inputs:
$X = (x_1, x_2, ..., x_n)$ Determine:
Clusters – k
Initial Centroids - $C_1, C_2, ..., C_k$
Assign each input to the cluster with the closest centroid Determine:
Update Centroids - $C_1, C_2, ..., C_k$
Repeat:
Until Centroids don't change significantly (specified threshold value)
Output:
Final Stable Centroids - $C_1, C_2, ..., C_k$
End

In most of the cases, *k*-means is quite slow to converge. For very accurate conditions, it takes quite a long time to converge exponentially. A reasonable threshold value may be specified for conversing in most of the cases to produce quick results without compromising much accuracy [9]. As shown in Table II, the Sum of Square of Errors (SSE) may be considerably reduced by defining more number of clusters. It is always desirable to improve SSE without increasing number of clusters which is possible due to the fact that *k*-means converges to a local minimum [10]. To decrease SSE, a cluster may be split or a new cluster centroid may be introduced.

Bisecting of *K*-Means Algorithm

Bisecting sequence of *k*-means algorithm
Begin
Initialize clusters
Do:
Remove a cluster from list
Select a cluster and bisect it using k-means algorithm
Compute SSE
Choose from bisected clusters one with least SSE
Add bisected clusters to the list of clusters
Repeat:
Until the number of cluster have been reached to k
End

To increase SSE, a cluster may be dispersed or two clusters may be merged. To obtain k-clusters from a set of all Observation points, the observation points are split into two clusters and again one of these clusters is split further into two clusters. Initially a cluster of largest size or a cluster with largest SSE may be chosen for splitting process. This is repeated until the *k* numbers of clusters have been produced. Thus it is easily observable that the SSE can be changed by splitting or merging the clusters [11]. This specific property of the *k*-means clustering is very much desirable for segmentation research. The new SSE is again computed after updating cluster centroid. This is repeated until SSE is reached to a minimum value or becomes constant without changing further, a condition similar to congruence. The SSE is represented mathematically by $SSE = \Sigma_{i=1}(\mu_i - x)^2$ where μ_i is the centroid of i^{th} cluster represented by c_i and x is any point in the same cluster. A condition for achieving minimum SSE can be easily computed by differentiating SSE, setting it equal to *0* and then solving the equation [12].

$$\frac{\partial}{\partial ..''} SSE = \frac{\partial}{\partial ..''} \sum_{i=1}^{k} \sum x\ c_i (..i - x)^2$$

$$= \sum_{i=1}^{k} \sum x\ c_i \frac{\partial}{\partial ..''} (..i - x)^2$$

$$= \sum x\ c_i\ 2 * (..i - xk) = 0$$

$$mk..k = \sum x\ c_k xk$$

Here m_k is total number of elements and μ_K is centroid in k^{th} cluster c_k. Further it can be simplified as –

$$\mu_k = 1\ \frac{\Sigma}{mk} x \in c\ x_k$$

This concludes that the minimum SSE can be achieved under the condition of the centroid of the cluster being equal to the mean of the points in the k^{th} cluster c_k.

Hierarchical Clustering Technique

It is a technique of clustering which divide the similar dataset by constructing a hierarchy of clusters. This method is based on the connectivity approach based clustering algorithms. It uses the distance matrix criteria for clustering the data. It constructs clusters step by step. Hierarchical clustering generally fall into two types: In hierarchical clustering, in single step, the data are not partitioned into a particular cluster. It takes a series of partitions, which may run from a single cluster containing all objects to 'n' clusters each containing a single object. Hierarchical Clustering is classified as

1. Agglomerative Nesting
2. Divisive Analysis

Agglomerative Nesting

It is also known as AGNES. It is bottom-up approach. This method construct the tree of clusters i.e. nodes. The criteria used in this method for clustering the data is min distance, max distance, avg distance, center distance. The steps of this method are:

1. Initially all the objects are clusters i.e. leaf.
2. It recursively merges the nodes (clusters) that have the maximum similarity between them.

At the end of the process all the nodes belong to the same cluster i.e. known as the root of the tree structure.

Divisive Analysis

It is also known as DIANA. It is top-down approach. It is introduced in Kaufmann and Rousseeuw (1990). It is the inverse of the agglomerative method. Starting from the root node (cluster) step by step each node forms the cluster (leaf) on its own. It is implemented in statistical analysis packages, e.g., plus.

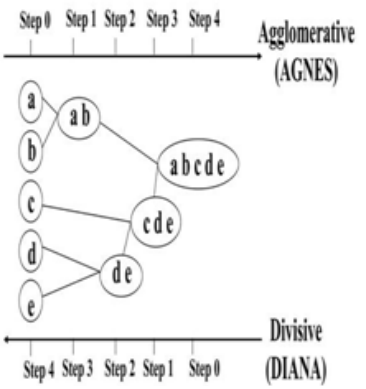


Fig. 2 : Representation of AGNES and DIANA

Advantages of hierarchical clustering [2]

1. Embedded flexibility with regard to the level of granularity.
2. Ease of handling any forms of similarity or distance.
3. Applicability to any attributes type.

Disadvantages of hierarchical clustering [2]

1. Vagueness of termination criteria.
2. Most hierarchical algorithm does not revisit once constructed clusters with the purpose of improvement.

Density Based Algorithms

Density based algorithms find the cluster according to the regions which grow with high density. It is the one-scan algorithms. It is able to find the arbitrary shaped clusters and handle noise. Representative algorithms include DBSCAN, GDBSCAN, OPTICS, and DBCLASD. The density based algorithm DBSCAN (Density Based Spatial Clustering of Applications with Noise) is commonly known. The Eps and the Minpts are the two parameters of the DBSCAN [6]. The basic idea of DBSCAN algorithm is that a neighborhood around a point of a given radius (ε) must contain at least minimum number of points (MinPts) [6]. The steps of this method are:

1. Randomly select a point t
2. Recover all density-reachable points from t wrt Eps and MinPts.
3. Cluster is created, if t is a core point
4. If t is a border point, no points are density-reachable from t and DBSCAN visits the next point of the database.

Continue the procedure until all of the points have been processed.

Farthest First Algorithm

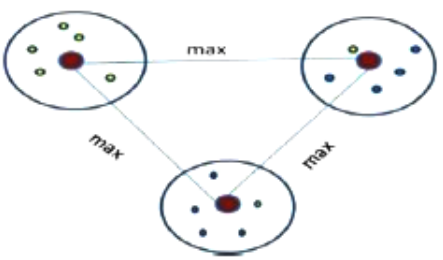
Farthest first algorithm proposed by Hochbaum and Shmoys 1985 has same procedure as k-means, this also chooses centroids and assign the objects in cluster but with max distance and initial seeds are value which is at largest distance to the mean of values, here cluster assignment is different, at initial cluster we get link with high Session Count, like at cluster-0 more than in cluster-1, and so on.

Farthest first algorithm need less adjustments and basic for this explained in [18].

Working as described here, it also defines initial seeds and then on basis of „k" number of cluster which we need to know prior. In farthest first it takes point P_i then chooses next point P_i which is at maximum distance. P_i is centroid and p_1, p_2, ..., p_n are points or objects of dataset belongs to cluster from equation 6

$$\min\{\max dist\,(p_1, p_1), \max dist\,(p_1, p_2)\}$$

Farthest first actually solves problem of k-centre and it is very efficient for large set of data.In farthest first algorithm we are not finding mean for calculating centroid ,it takes centrod arbitrary and distance of one centroid from other is maximum figure-2 shows cluster assignment using farthest –first. When we performed outlier detection for our dataset we get which objects is outlier.



Filtered Clustering Algorithm

Based on the definition of nearest neighbour pair C. S. Li et al. [19] proposed a new cluster center initialization method for K-Means algorithm. In iterative clustering algorithms, selection of initial cluster centers is extremely clustering algorithm called the filtering algorithm [2] shows that the algorithm runs faster as the separation between clusters increases.

Experiment and Results

Experimental setup

The performance of various clustering algorithms are measured based on the time to form the clusters. Here, two datasets are used namely, letter image dataset and abalone dataset.

Table 1: Datasets Used

Dataset	No. of Instances	No. of attributes
COCOMO 81	63	17
LABOUR	57	17

Weka, a data mining tool is used to execute the datasets. Various clustering algorithms are used to form clusters and their performance evaluation is being analyzed. The two datasets used here are described below.

COCOMO 81 dataset

This is a PROMISE Software Engineering Repository data set made publicly available in order to encourage repeatable, verifiable, refutable, and/or improvable predictive models of software engineering. Some of the attributes used here are programmers capability, turnaround time, process complexity, time constraint, etc…

LABOUR dataset

Data was used to test 2 tier approach with learning from positive and negative examples. Some of the attributes used here are duration, wages, number of working hours, pension, education allowance, etc…

Weka

Weka is a collection of open source ML algorithms used for pre-processing, classifiers, clustering, and association rule [14]. Weka is created by researchers at the University of Waikato in New Zealand. It is a Java based tool used in the field of data mining. It uses flat text files to describe the data. It can work with a wide variety of data files including its own ".arff" format and C4.5 file formats.

Results for datasets on different Clustering algorithms

The COCOMO 81 dataset and LABOUR dataset are processed on various clustering algorithms such as simple KMeans, Hierarchical, filtered clusterer, density based clusterer and farthest first clustering.

From table II, it is shown that the two datasets are subjected to generate the number of clustered instances based on Symmetric and Asymmetric attributes.

Table 2: Generation of Clustered Instances

Dataset	Cluster Algorithms	Clustered Instances			
		Symmetric Attributes	%	Asymmetric Attributes	%
Cocomo81	K-mean	23	37%	40	63%
	Hierarchical	62	98%	1	2%
	DensityBased	24	38%	39	62%
	FarthestFirst	50	79%	13	21%
	Filtered	23	37%	40	63%
Labour	K-mean	48	84%	9	16%
	Hierarchical	57	100%	0	0%
	DensityBased	44	77%	13	23%
	FarthestFirst	47	82%	10	18%
	Filtered	48	84%	9	16%

Graph representation for performance evaluation

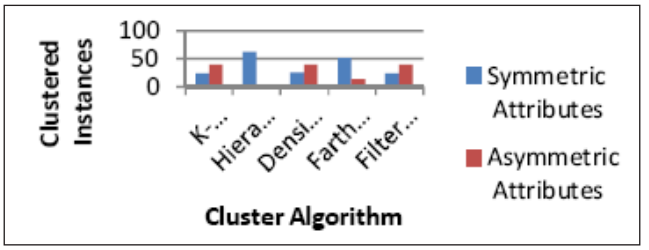


Fig. 1: Clusters Generation (COCOMO81 dataset)

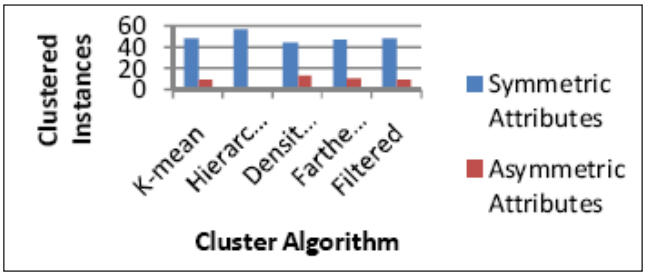


Fig. 2: Clusters Generations (LABOUR dataset)

Conclusion

A comparative study of clustering algorithms across two different data items is performed here. The performance of the various clustering algorithms is compared based on how the clustered instances are generated. The experimental results of various clustering algorithms to form clusters are depicted as a graph. The Hierarchical clustering algorithm generates a large number of clustered instances in both dataset whereas the simple K-Mean and DensityBased generates a less number of clustered instances in both dataset. This proposal can be used in future for similar type of research work.

References

I. S. Dhillon and D. M. Modha, "Concept decompositions for large sparse text data using clustering," *Machine Learning*, vol. 42, issue 1, 143-175, 2001.

T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "An efficient K-means clustering algorithm," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 24, pp. 881-892, 2002.

MacKay and David, "An Example Inference Task: Clustering," *Information Theory, Inference and Learning Algorithms*, Cambridge University Press, pp. 284-292, 2003.

M. Inaba, N. Katoh, and H. Imai, "Applications of weighted Voronoi diagrams and randomization to variance-based *k*-clustering," in *Proc.10th ACM Symposium on Computational Geometry*, 1994, pp. 332-339.

S. Dasgupta and Y. Freund, "Random Trees for Vector Quantization," *IEEE Trans. on Information Theory,* vol. 55, pp. 3229-3242, 2009.

M. Mahajan, P. Nimbhorkar, and K. Varadarajan, "The Planar K-Means Problem is NP-Hard," *LNCS*, Springer, vol. 5431, pp. 274-285, 2009.

A. Vattani, "K-means exponential iterations even in the plane," *Discrete and Computational Geometry,* vol. 45, no. 4, pp. 596-616, 2011.

C. Elkan, "Using the triangle inequality to accelerate K-means," in *Proc. the 12th International Conference on Machine Learning (ICML),* 2003.

H. Zha, C. Ding, M. Gu, X. He, and H. D. Simon, "Spectral Relaxation for K-means Clustering," *Neural Information Processing Systems,* Vancouver, Canada, vol.14, pp. 1057-1064, 2001.

C. Ding and X.-F. He, "K-means Clustering via Principal Component Analysis," in *Proc. Int'l Conf. Machine Learning (ICML*), 2004, pp. 225-232.

P.-N. Tan, V. Kumar, and M. Steinbach, *Introduction to Data Mining*, Pearson Educatio Inc. and Dorling Kindersley (India) Pvt. Ltd., New Delhi and Chennai Micro Print Pvt. Ltd., India, 2006.

H.-B. Wang, D. Huo, J. Huang, Y.-Q. Xu, L.-X. Yan, W. Sun, X.-L. Li, and Jr. A. R. Sanchez, "An approach for improving K-means algorithm on market segmentation," in *Proc. International Conference on Systme Science and Engineering (ICSSE)*, IEEE Xplore, 2010.

J.Daxin, C.Tang and A. hang (2004) Cluster Analysis for Gene Expression Data: A Survey, IEEE Transactions on Knowledge and Data Engineering, Vol. 16, Issue 11, pp. 1370-1386.

Zengyou He "Farthest-Point Heuristic based Initialization Methods for K-Modes Clustering.

C. S. Li, "Cluster Center Initialization Method for K-means Algorithm Over Data Sets with Two Clusters", "2011 International Conference on Advances in Engineering, Elsevier", pp. 324-328, vol.24, 2011.