Scientific Journal Impact Factor (SJIF) : 1.711
International Journal of Modern Trends in Engineering and Research
ISSN (ONLINE) : 2349 - 9745
ISSN (PRINT) : 2393 - 8161

Hub Based and Subspace Clustering: A Review on High Dimensional Clustering

Shahany Habeeb[1], Syam Gopi[2]
[1,2]*CS Department, MG University*

Abstract— Clustering comes under unsupervised learning process. It is the process of grouping elements together in such a way that the elements in group posses same features. The intension of this paper is to provide a review on high dimensional clustering methods. Mainly subspace based clustering and hub based clustering methods are explained in this review. Hub based clustering is one of the new method in this area. Three algorithms under this category are reviewed in this paper.
Keywords— Clustering, Hubness, unsupervised learning, sub space clustering.

I. INTRODUCTION

Very often most of today's application area uses data mining as a knowledge discovery tool. With the help of data mining it is possible to summarize huge amount of data to useful information. This useful information can be in any form like rules, patterns, relationships, constraints etc. Data mining draws ideas from statistics, machine learning/AI, and database system [1]. The heterogeneous distributed nature, high dimensionality, enormity of data makes traditional techniques unsuitable.

Clustering is a powerful task in data mining and data analysis applications. It is an unsupervised process that finds natural combination of instances given unlabeled data. A cluster is considered as a subset of objects which are "comparable", or a subset of objects such that the distance between any two objects in the cluster is less than the distance between any object in the cluster and any object not located inside it.

Some of the good reasons while considering wide usage of clustering include,

- Unsupervised learning process
- Pattern detection
- Simplifications
- Useful in data concept construction

Clustering can be used in many areas like data mining, pattern-analysis, Information retrieval, image segmentation, text mining, web analysis, marketing, pattern classification, medical diagnostics etc.

II. HIGH DIMENSIONAL CLUSTERING

Clustering high-dimensional data is the cluster analysis of data with few dozen to many thousands of dimensions [2]. Low dimensional data clustering is very easier with the help of clustering algorithms. When dimensionality increases clustering became more difficult due to the increasing sparsity of data. Dealing with high dimensional data analysis, "Curse of dimensionality" [4], [5] is one among the most widely observed phenomenon, which is mainly due to empty space phenomenon and concentration of distances. When dimensionality increases the number of points

required to represent each distribution also increases exponentially. Thus data became sparse, this is referred as empty space phenomenon. As dimensionality increases it became difficult to distinguish distance between data points, and it is termed as concentration of distances. Thus the phenomenon "curse of dimensionality" negatively affects high dimensional clustering results. Some of the problems which we need to overcome during high dimensional clustering include,

- Curse of dimensionality.
- As the dimensionality increases distance discrimination becomes meaningless.
- Local feature relevance problem: different clusters might be found in different subspaces, so a global filtering of attributes is not sufficient.
- Given a large number of attributes, it is likely that some attributes are correlated. Hence, clusters might exist in arbitrarily oriented affine subspaces.

Attributes transformations and domain decomposition are two main techniques that are used to fight against high dimensionality. Attributes transformations are simple functions of existent attributes. In multivariate statistics principal components analysis (PCA) is popular, but leads to clusters with poor interpretability. Therefore this approach is problematic. Singular value decomposition (SVD) is used to reduce dimensionality in statistics and information retrieval. Low-frequency Fourier harmonics in conjunction with Parseval's theorem are successfully used in analysis of time series as well as wavelets and other transformations. As its name indicate domain decomposition divides the data into subsets using some similarity measure, so that the high dimensional computation happens over smaller datasets. Dimension stays the same, but the costs are reduced. This approach mainly targets the situation of high dimension, large data, and many clusters.

2.1 Subspace clustering

Subspace clustering [6], [7] can be considered as one of the extension of traditional clustering techniques. This method will search for clusters in different subspace with in a dataset. Subspaces can either be axis-parallel or affine. High dimensionality sometimes makes clusters noisy. By analyzing entire dataset it is possible to remove irrelevant attributes by feature selection. Feature selection extract most relevant feature required for data mining task under consideration.

A subspace clustering method searches various subspaces for clusters. Here, a cluster is a subset of objects that are most similar to each other in a subspace. The similarity is often captured by with the help of measures such as distance or density. Efficient subspaces search can be considered as one of the main challenge during subspace clustering. Based on the subspace search mainly two strategies are there

2.1.1. Bottom-up approaches: As its name indicate it start from low-dimensional subspaces and search higher-dimensional subspaces only when there may be clusters in those higher-dimensional subspaces. Different pruning techniques are used to reduce the number of higher-dimensional subspaces that need to be searched.

a) CLIQUE: CLustering In QUEst is a simple grid-based clustering method for finding density-based clusters in subspaces. This method partitions each dimension into non overlapping intervals, thereby partitioning the entire embedding space of the data objects into cells. Here a u density threshold is used to identify dense and sparse cells. If the number of objects mapped to it exceeds the density threshold then it is a dense cell. Then it uses the dense cells in each subspace to assemble clusters, which can be of arbitrary shape.

b) ENCLUS: ENCLUE is a subspace clustering method based on the CLIQUE clustering. It does not measure density or coverage directly, but instead measures entropy. Here the main observation is that a subspace with clusters typically has lower entropy than a subspace without clusters. It uses the same APRIORI style, bottom-up approach as CLIQUE to mine relevant subspaces. By using ENCLU clustering it is possible to identify overlapping clusters of arbitrary shapes.

c) Cell-based Clustering Method (CBF): One problem associated with other bottom-up algorithms is that as the number of dimensions increases the number of bins created also increases. CBF algorithms mainly try to dress scalability issues associated with many bottom-up algorithms. Here the cell creation algorithm repeatedly examine minimum and maximum values on a given dimension and produce fewer bins or cells. An efficient filtering-based index structure is used to store bins which results in improved retrieval performance.

d) MAFIA: Merging of Adaptive Finite Intervals is another extension of CLIQUE clustering. In this algorithm an adaptive grid based on the distribution of data is used to improve cluster quality. In order to improve scalability MAFIA also introduces parallelism. MAFIA creates histogram to determine the minimum number of bins for a dimension. Thousands of bins are used to compute histograms by reading blocks of data in core memory, then these bins are merged together to come up with a smaller number of variable-size bins than CLIQUE does.

e) CLTree: For a given dataset, CL Tree clustering uses a modified decision tree algorithm to select the best cutting planes. It uses a decision tree algorithm to partition each dimension into bins that is for separating areas of high density from of low density. CLTree follows the bottom-up strategy, evaluating each dimension separately and then using only those dimensions with areas of high density in further steps.

2.1.2 Top-down approaches: It starts from the full space and search smaller subspaces recursively. It is effective only if the locality assumption holds, which require that the subspace of a cluster can be determined by the local neighborhood.

a) PROCLUS (PROjected CLUstering): PROCLUS was the first top-down clustering algorithm. It uses the same approach as k-medoid clustering. Initially it take k-medoids and the iteratively improve clustering. Initial medoids are guessed, for each medoid determine the subspace spanned by attributes with low variance. Points are assigned to the closest medoid, considering only the subspace of that medoid in determining the distance. Then algorithm proceeds as the PAM (Partitioning Around Medoids) algorithm.

b) FINDIT: Considering subspace clustering, selecting correct dimension is very important. The main reason is that distance between points is easily changes according to the selected dimensions. This selection process is somewhat difficult, because data grouping and dimension selecting should be performed simultaneously. In order determines the correlated dimensions for each cluster FINDIT [8**]** uses two key ideas*:* dimension-oriented distance measure which completely utilizes dimensional difference information, and dimension voting policy which determines important dimensions in a probabilistic way based on V nearest neighbors' information.

c) ORCLUS (ORiented projected CLUSter generation): ORCLUS algorithm uses a same approach of projected clustering, but incorporate non-axes parallel subspaces of high dimensional space. This algorithm arose from the idea that many datasets contain inter-attribute correlations. The algorithm can be divided into three stages: assign clusters, subspace determination, and merge.

d) Clustering On Subsets of Attributes (COSA): It is an iterative process that assigns weights to each dimension for each instance, not each cluster. Starting with equally weighted dimensions, the process examines the k nearest neighbors of each instance. The respective dimension weights for each instance are calculated using these neighborhoods. Higher weights are assigned to those dimensions that have a smaller dispersion within the *knn* group. These weights are used to calculate dimension weights for pairs of instances which are in turn used to update the distances used in the knn calculation. Using the new distances repeat the process until the weights stabilize.

e) Co-clustering: Co-clustering is also termed as conjugate clustering, distributional clustering, information bottleneck method, bi-dimensional clustering, block clustering, and simultaneous clustering. Concept of co-clustering is from the idea of producing attribute groups in conjunction with clustering of points themselves. Co-clustering can be considered as clustering of both points and their attributes simultaneously. Co-clustering utilizes a canonical duality contained in the point-by-attribute representation. The co-clustering deals with distributional clustering of attributes based on the informational measures of attribute similarity. Two attributes (two columns in matrix X) with exactly the equal probability distributions are similar for data mining purpose, therefore one can be deleted. Attributes that have probability distributions that are close in terms of their Kullback-Leibler (KL) distance can still be grouped together without much of an impact. In addition, a natural derived attribute, the mixed distribution is now available to represent the group.

2.2 Hub based clustering

Subspace clustering and co-clustering techniques very common under high dimensional clustering. Both these include many efficient clustering algorithms. Hub based clustering is one of the new area under high dimensional clustering. The tendency of high-dimensional data to occur much more frequently in k-nearest neighbour lists of other points is termed as hubness. Formally we can represent hubness as $N_k(x)$, the number of k-occurrences of point $x \in \mathbf{R}^d$, is the number of times x occurs among k nearest neighbors of all other points in a data set. Under certain conditions, as dimensionality increases, the distribution of N_k becomes considerably skewed to the right, resulting in the emergence of hubs, i.e., points which appear in many more k-NN lists than other points. It is possible to use hubness for clustering in various ways, because hubness can be considered as a kind of local centrality measure. Based on hubness three clustering algorithm are there:

2.2.1 Deterministic Approach

One of the ways for using hubness for clustering is to use them as centroid. This approach is similar as that of k-means clustering method. This algorithm is referred as k-hubs algorithm. Here the point which having highest hubness score is taken as the cluster center. The neighboring points are assigned to the cluster center similar to k-means clustering. During each iteration, the center changes until there is no reassignment.

2.2.2 Probabilistic Approach

In order to increase the probability of obtaining global optimum a probabilistic approach is proposed. The probabilistic clustering algorithm comes under advance clustering methods. This approach incorporate squared hubness-proportional stochastic scheme along with simulated annealing approach to optimization. Here the algorithm is termed as hubness proportional clustering (HPC). In order to control various iterations of the algorithm a temperature value is taken. Based on the temperature value a probability is calculated, and then it is compared with a randomly generated float value. If the probability value is greater than the random value then the point which having highest hubness score is set as the cluster center. Otherwise the square of hubness score is set as the choosing probability of each point and chooses points probabilistically. During each iteration temperature is updated until there is no re assignments.

2.2.3 A Hybrid Approach

The algorithms k-hubs and HPC share a property that they do not require knowledge of data/object representation so all that is required is a distance/similarity. If the representation is also available such that it is possible to meaningfully calculate centroids, there also exists a third alternative: use point hubness scores to guide the search, but choose a centroid-based cluster configuration in the end. This approach is referred to as hubness-proportional K-means (HPKM). This algorithm is same as that of HPC algorithm, the only difference is instead of choosing those point which having highest hubness score as center point, centroid is taken. That is a combination of HPC and K-means algorithm. The rest of the algorithm is same as that of HPC.

III. CONCLUSION

Clustering is one of the important task in data mining. Many clustering algorithms are there. For high dimensional clustering subspace clustering algorithms are mainly applied. This paper gives detailed review of subspace clustering and one of the new high dimensional clustering methods, i.e hub based clustering. Three algorithms are reviewed in this paper. Deterministic approach, probabilistic approach and hybrid approach.

REFERENCES

[1] Rui Xu, Donald C. Wunsch II, "Survey of Clustering Algorithms", IEEE Transactions on neural Networks, vol. 16, pp. 645-678, May 2005.
[2] Prof. Neha Soni, Vasad, Prof. Amit Ganatra, "Categorization of Several Clustering Algorithms from Different Perspective: A Review", Volume 2, Issue 8, August 2012.
[3] J. Kelinberg, "An impossibility theorem for clustering", in NIPS 15, MIT Press,2002, pp. 446-453.
[4] Nenad Tomasev, Milos Radovanovic, Dunja Mladenic, and Mirjana Ivanovi "The Role of Hubness in Clustering High-Dimensional Data", IEEE Tran on knowledge and data engineering, vol. 26, NO. 3, March 2014.
[5] Michael Steinbach, Levent Ertoz, and Vipin Kumar, "The Challenges of Clustering High Dimensional Data."
[6] Lance Parsons, Ehtesham Haque, Huan LiuSubspace, "Clustering for High Dimensional Data: A Review".
[7] B.Hari Babu, N.Subash Chandra2 & T. Venu Gopal3, "Clustering Algorithms for High Dimensional Data – A Survey Of Issues And Existing Approaches".
[8] Kyoung-Gu Woo' , Jeong-Hoon Lee , Myoung-Ho Kim , Yoon-Joon Lee Department of Electrical Engineering and Computer Science, Korea Advanced Institute of Science and Technology, Kusong-Dong 373-1, Yusung-Gu Taejon 305-701, South Korea," FINDIT: a fast and intelligent subspace clustering algorithm using dimension voting".

