Mr. Santosh D. Rokade, Mr. A. M. Bainwad / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com
Vol. 3, Issue 3, May-Jun 2013, pp.504-508

Survey of Combined Clustering Approaches

Mr. Santosh D. Rokade*, Mr. A. M. Bainwad**
*(M.Tech Student, CSE Department, SGGS IE&T, Nanded, India)
** (Assistant Professor, CSE Department, SGGS IE&T, Nanded, India)

ABSTRACT

Data clustering is one of the very important techniques and plays an important role in many areas such as data mining, pattern recognition, machine learning, bioinformatics and other fields. There are many clustering approaches proposed in the literature with different quality, complexity tradeoffs. Each clustering algorithm has its own characteristics and works on its domain space with no optimum solution for all datasets of different properties, sizes, structures, and distributions. Combining multiple clustering is considered as new progress in the area of data clustering. In this paper different combining clustering algorithms are discussed. Combining clustering is based on the level of cooperation between the clustering algorithms; either they cooperate on the intermediate level or end result level. Cooperation among multiple clustering techniques is for the goal of increasing the homogeneity of objects within the clusters

Keywords - Chameleon , Ensemble clustering, Generic and Kernel based ensemble clustering, Hybrid clustering,

1. Introduction

The goal of clustering is to group the data points or objects that are close or similar to each other and identify such grouping in an unsupervised manner, unsupervised is in the sense that no information is provided to the algorithm about which data point belongs to which cluster. In other words data clustering is a data analysis technique that enables the abstraction of large amounts of data by forming meaningful groups or categories of objects, these objects are formally known as clusters. This grouping is in such a way that objects in the same cluster are similar to each other, and those in different clusters are dissimilar according to some similarity measure or criteria. The increasing importance of data clustering in different areas has led to the development of a variety of algorithms. These algorithms are differing in many aspects, such as the similarity measure used, the types of attributes they use to characterize the dataset, and the representation of the clusters. Combination. of different clustering algorithm is new progress area in the document clustering to improve the result of different clustering algorithms. The cooperation is performed on end result level or intermediate level. Examples of end-result cooperation are the ensemble clustering and the hybrid clustering approaches [3-9, 11, 12]. Cooperative clustering model is an example of intermediate level cooperation [13]. Sometimes, k-means and agglomerative hierarchical approaches are combined so as to get the best results as compares to individual algorithm. For example, in the document domain Scatter/Gather [1], a document browsing system based on clustering, uses a hybrid approach involving both k-means and agglomerative hierarchical clustering. The k-means is used because of its run-time efficiency and the agglomerative hierarchical clustering is used because of its quality [2]. Survey of different ensemble clustering and hybrid clustering approaches is done in the upcoming section of this paper.

2. Ensemble clustering

The concept of cluster ensemble is introduced in [3] by A. Strehl and J. Ghosh. In this paper, they introduce the problem of combining multiple partitioning of a set of objects without accessing the original features. They call this problem as cluster ensemble problem. The cluster ensemble problem is then formalized as a combinatorial optimization problem in terms of shared mutual information. In addition to a direct maximization approach, they propose three effective and efficient techniques for obtaining high quality combiners or consensus functions. The first combiner induces a similarity measure from the partitioning and then reclusters the objects. The second combiner is based on hypergraph partitioning. The third one collapses groups of clusters into meta-clusters which then compete for each object to determine the combined clustering. Unlike classification or regression settings, there have been very few approaches proposed for combining multiple clusterings. Bradley and Fayyad [4] in 1998 proposed an approach for combining multiple clusterings; here they combine the results of several clusterings of a given dataset, where each solution resides in a common known feature space, for example combining multiple sets of cluster centers obtained by using k-means with different initializations.

According to D. Greene, P. Cunningham, recent techniques for ensemble clustering are effective in

improving the accuracy and stability of standard clustering algorithms though these techniques have drawback of computational cost of generating and combining multiple clusterings of the data. D. Greene, P. Cunningham proposed efficient ensemble methods for document clustering, in that they present an efficient kernel-based ensemble clustering method suitable for application to large, high-dimensional datasets [5].

2.1 Generic Ensemble Clustering

Ensemble clustering is based on the idea of combining multiple clusterings of a given dataset to produce a better combined or aggregated solution. The general process followed by these techniques is given in the fig.1 which has two different phases.

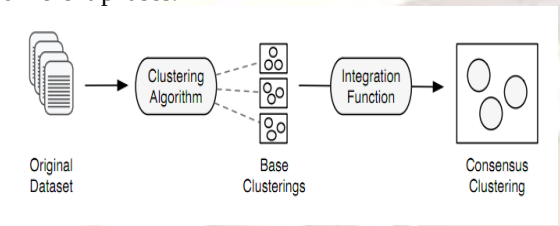

Fig.1 Generic ensemble clustering process.

Phase-1. Generation: Construct a collect-ion of τ base clustering solutions, denoted as $C = \{C_1, C_2, \ldots, C_t,\}$ which represents the members of the ensemble. This is typically done by repeatedly applying a given clustering algorithm in a manner that leads to diversity among the members.

Phase-2. Integration: Once a collection of ensemble members has been generated, a suitable integration function is applied to combine them to produce a final "consensus" clustering.

2.2 Kernel-based Ensemble Clustering

In order to avoid repeated recompution of similarity values in original feature space, D. Greene and P. Cunningham choose to represent the data in the form of an $n * n$ kernel matching k, where k indicates the close resemblance between object x_i and x_j. The main advantage of using kernel methods in the ensemble clustering is that after construction of single kernel matrix we may subsequently generate multiple partitions without using original data. In [5] Greene D.Tsymbal proposed a Kernel-based correspondence clustering with prototype reduction that produces more stable results than other schemes such as those based on pair-wise co-assignments, which are highly sensitive to the choice of final clustering algorithm. The Kernel-based correspondence clustering algorithm is described as follows:

1) Construct full kernel matrix *k* and set counter $t = 0$.
2) Increment t and generate base clustering C_t :
   - Produce a sub sampling without replacement.
   - Apply adjusted kernel k-means with random initialization to the samples.
   - Assign each out-of-sample object to the nearest centroid in C_t .
3) If $t = 1$, initialize V as the $n * k$ binary membership matrix for C_1. Otherwise, update V as follows:
   - Compute the current consensus clustering C from V such that: $x_i \in \bar{C}_j$ if $j = \arg\max V_{ij}$
   - Find the optimal correspondence $\pi(C_t)$ between the clusters in C_t and $\bar{C}$.
   - For each object x_i assigned to the j^{th} cluster in the (C_t), increment v_{ij}.
4) Repeat from Step 2 until $\bar{C}$ is stable or $t = T_{max}$.
5) Return the final consensus clustering $\bar{C}$.

2.3 Ensemble clustering with Kernel Reduction

The ensemble clustering approach introduced by D. Greene, P. Cunningham [5] allows each base clustering to be generated without referring back to the original feature space but, for larger datasets the computational cost of repeatedly applying an algorithm is very high $O((\beta_n)^2)$. To reduce this computational cost, the value of n should be reduced. After this the ensemble process becomes less computationally expensive. Greene and Cunningham [5] showed that the principles underlying the kernel-based prototype reduction technique may also be used to improve the efficiency of ensemble clustering. The proposed techniques mainly performed in three steps such as applying prototype reduction, performing correspondence clustering on the reduced representation and subsequently mapping the resulting aggregate solution back to the original data. The entire process is illustrated in Fig. 2.

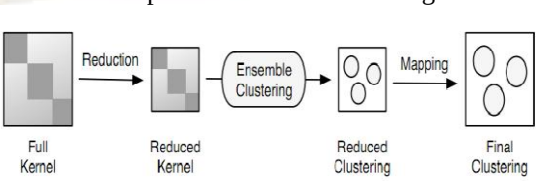

Fig. 2 Ensemble clustering process with prototype reduction

The entire ensemble process with prototype reduction is summarized in following algorithm.

1) Construct full n * n kernel matrix K from the original data X.

2) Apply prototype reduction to form $n' * n'$ the reduced kernel matrix k'.
3) Apply kernel-based correspondence clustering using K' as given in kernel-based correspondence clustering algorithm to produce a consensus clustering $\overline{C}$'.
4) Construct a full clustering $C^{\wedge}$ by assigning a cluster label to each x_i based on the nearest cluster in $\overline{C}$'.
5) Apply adjusted kernel k-means using $C^{\wedge}$ as an initial partition to produce a refined final clustering of X.

Recent ensemble clustering techniques have been shown to be effective in improving the accuracy and stability of standard clustering algorithms but computational complexity is the main drawback of ensemble clustering techniques.

3. Hybrid Clustering

To improve the performance and efficiency of algorithms several clustering methods have been proposed to combine the features of hierarchical and partitional clustering algorithms. In general, these algorithms first partition the input data set into m sub clusters and then a new hierarchical structure is constructed based on these m subclusters.

This idea of hybrid clustering is first proposed in [6] where a multilevel algorithm is developed. N. M. Murty and G. Krishna described a hybrid clustering algorithm based on the concepts of multilevel theory which is nonhierarchical at the first level and hierarchical from second level onwards to cluster data sets having chain-like clusters and concentric clusters. N. M. Murty and G. Krishna observed that this hybrid clustering algorithm gives the same results as the hierarchical clustering algorithm with less computation and storage requirements. At the first level, the multilevel algorithm partitions the data set into several partitions and then performs the k-means algorithm on each partition to obtain several subclusters. In subsequent levels, this algorithm uses the centroids of the subclusters identified in the previous level as the new input data points and performs the hierarchical clustering algorithm on those points. This process continues until exactly k clusters are determined. Finally, the algorithm performs a top-down process to reassign all points of each subcluster to the cluster of their centroids [7].

Balanced Iterative Reduced Clustering using Hierarchies (BIRCH) is another hybrid clustering algorithm designed to deal with large input data sets [8, 9]. BIRCH algorithm introduces two important concepts, first is cluster feature and another is cluster feature tree which are used to summarize cluster representations. These structures help the clustering method achieve good speed and scalability in large databases and also make it effective for incremental and dynamic clustering of incoming objects. A clustering feature (CF) is three dimensional vector summarizing information about clusters of objects. BIRCH uses CF to represent a subcluster. If CF of a subcluster is given, we can obtain the centroid, radius, and diameter of that subcluster easily. The CF vector of a new cluster is formed by merging two subclusters. This can be directly derived from the CF vectors of the two subclusters by algebra operations. A CF tree is a height balanced tree that stores the clustering features for a hierarchical clustering. The non leaf nodes store sums of the CFs of their children and thus summarize clustering information about their children.

BIRCH algorithm consists of four phases. In Phase 1, BIRCH scans the database to build an initial in-memory CF tree, which can be viewed as a multilevel compression of the data that tries to preserve the inherent clustering structure of the data. BIRCH partitions the input data set into many subclusters by a CF tree. In Phase 2, it reduces the size of the CF tree that is the number of subclusters in order to apply a global clustering algorithm in Phase 3 on those generated subclusters. In Phase 4, each point in the data set is redistributed to the closest centroids of the clusters produced in Phase 3. Among these phases, Phase 2 and Phase 4 are used to further improve the clustering quality. Therefore these two phases are optional. BIRCH tries to produce the best clusters with the available resources with a limited amount of main memory. An important consideration is to minimize the time required for input/output. BIRCH applies a multiphase clustering technique as: a single scan of the data set yields a basic good clustering and one or more additional scans can be used to further improve the quality [9].

G. Karypis, E.H. Han, and V. Kumar in [10] proposed another hybrid clustering algorithm named as CHAMELEON. Chameleon is a hierarchical clustering algorithm that uses dynamic modeling to determine the similarity between pairs of clusters. The Chameleon algorithm's key feature is that it gives importance for both interconnectivity and closeness in identifying the most similar pair of clusters. Interconnectivity is the number of links between two clusters and closeness is the length of those links. This algorithm is described as follows and summarized in fig. 3

1) Construct a k-nearest neighbour graph.
2) Partition the k-nearest neighbour graph into many small sub clusters.
3) Merge those sub clusters into final clustering results.

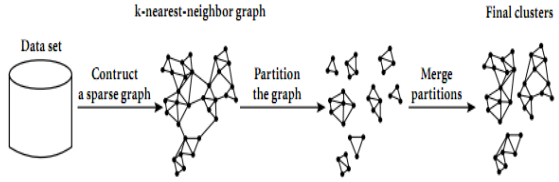

Fig.3 Chameleon Algorithm

Chameleon uses a k-nearest neighbour graph approach to construct a sparse graph. Each vertex of this graph represents a data object. There exists an edge between two vertices or objects if one object is among the k most similar objects of the other. The edges are weighted to reflect the similarity between objects. Chameleon uses a graph partitioning algorithm to partition the k-nearest neighbour graph into a large number of relatively small subclusters. It then uses an agglomerative hierarchical clustering algorithm that repeatedly merges subclusters based on their similarity. To determine the pairs of most similar subclusters, it takes into account both the interconnectivity as well as the closeness of the clusters [9].

Zhao and Karypis in [11] showed that the hybrid model of Bisecting k-means (BKM) and k-means (KM) clustering produces better results than individual BKM and KM. BKM [12] is a variant of KM clustering that produces either a partitional or a hierarchical clustering by recursively applying the basic KM method. It starts by considering the whole dataset to be one cluster. At each step, one cluster V is selected and bisected further into two partitions V_1 and V_2 using the basic KM algorithm. This process continues until the desired number of clusters or some other specified stopping condition is reached. There are a number of different ways to choose which cluster to split. For example, we can choose: the largest cluster at each step or the one with least overall similarity or a criterion that satisfies both size and overall similarity. This bisecting approach is very attractive in many applications such as document retrieval, document indexing problems and gene expression analysis as it is based on the homogeneity criterion. However, in some cases when a fraction of the dataset is left behind with no other way to re-cluster it again at each level of the binary hierarchical tree, a "refinement" is needed to re-cluster these resulting solutions. In [11], it has been conclude that the BKM with end-result refinement using the KM produces better results than KM and BKM. A drawback of this end- result enhancement is that KM has to wait until the former BKM finishes its clustering and then it takes the final set of centroids as initial centres for a better refinement [2].

Thus, in hybrid clustering, cascaded clustering algorithms cooperates together for the goal of refining the clustering solutions produced by a former clustering algorithm. Different hybrid clustering approaches discussed above are shown to be effective in improving the clustering quality but main drawback of this hybrid clustering approach is that it does not allow synchronous execution of the clustering algorithms that is one algorithm has to wait for another algorithm to finish its clustering.

4. Conclusions

Combining multiple clustering is considered as an example to further broaden a new progress in the area of data clustering. In this paper different combined clustering approaches have been discussed that are shown to be effective in improving the clustering quality. Thus, computational complexity is the main drawback of ensemble clustering techniques and idle time wastage is one of the drawback of hybrid clustering approaches

REFERENCES

[1] Cutting, D., Karger, D., Pedersen, J. and Turkey, J.W., "Scatter/Gather: A Cluster-based Approach to Browsing Large Document Collections," *SIGIR '92*, 1992, pp. 318-329.

[2] M. Steinbach, G. Karypis, V. Kumar, "A comparison of document clustering techniques", *In Proceeding of the KDD Workshop on Text Mining*, 2000, pp. 109-110.

[3] A. Strehl, J. Ghosh, "Cluster ensembles: knowledge reuse framework for combining partitioning", *Conference on Artificial Intelligence (AAAI 2002)*, AAAI/MIT Press, Cambridge, MA, 2002, pp. 93-98.

[4] U. M. Fayyad, C. Reina, and P. S. Bradley, "Initialization of iterative refinement clustering algorithms", *InProc. 14th Intl. Conf. on Machine learning (ICML)*, 1998, pp. 194-198.

[5] D. Greene, P. Cunningham, "Efficient ensemble methods for document clustering", Technical Report, Trinity College Dublin, Computer Science Department, 2006.

[6] N.M. Murty and G. Krishna, "A Hybrid Clustering Procedure for Concentric and Chain-Like Clusters", *Int'l J. Computer and Information Sciences, vol. 10, no. 6*, 1981, pp. 397-412.

[7] C. Lin, M. Chen, "Combining partitional and hierarchical algorithms for robust and efficient data clustering with cohesion self-merging", *IEEE Transactions on Knowledge and Data Engineering 17 (2)*, 2005, pp. 145-159.

[8] T. Zhang, R. Ramakrishna, and M. Livny, "BIRCH: An Efficient Data Clustering Method for Very Large Databases," *Proc. Conf. Management of Data (ACM SIGMOD '96)*, 1996, pp. 103-114.

[9] Jiawei Han and Micheline Kamber, *Data Mining: Concepts and Techniques* (Second Edition. Jim Gray, Series Editor, Morgan Kaufmann Publishers, March 2006).

[10] G. Karypis, E.H. Han, and V. Kumar, "Chameleon: Hierarchical Clustering Using Dynamic Modeling," *IEEE Computer Society, vol. 32, no. 8*, Aug. 1999, pp. 68-75.

[11] Y. Zhao, G. Karypis, "Criterion functions for document clustering: experiments and analysis", Technical Report, 2002.

[12] S.M. Savaresi, D. Boley, "On the performance of bisecting k-means and PDDP", *in: Proceedings of the 1st SIAM International Conference on Data Mining*, , 2001, pp. 114.

[13] Rasha Kashef, Mohamed S. Kamel, "Cooperative Clustering*", Pattern Recognition, 43*, 2010, pp. 2315-2329.