Divisive Clustering Algorithm: DIANA (Divisive Analysis)

Bocheng Yin

by16@illinois.edu

Introduction

Divisive Analysis (DIANA) is a divisive algorithm to split the items as a whole cluster at the beginning, followed with iterating the bi-partitioning process until only one item is concluded in one cluster. This review will recapitulate its most classic method, Macnaughton-Smith (1964), walk through the dissimilarity criteria for bipartition evaluation, reallocation of items for clustering optimization, and initiation strategy for the 1st splitting. Also, this review will briefly present the workflow in which DIANA will be complemented with other algorithms for the processing of the unstructured data, messy data with lots of redundancy, and big dataset.

Body

Divisive Analysis (DIANA) was firstly systematically described in Chapter 6 of the book published in 1990(Kaufman & Rousseeuw, 1990). In this classic book, DIANA, is a reversal algorithm of AGNE. Both of them are hierarchical clustering algorithm which gradually splits or aggregates clusters by DIANA and AGNE, respectively(Kaufman & Rousseeuw, 1990). Unlike the prevalence of AGNE in the year of 1990, DIANA was not popular due to the division patterns could be computationally heavy. For a "n" size of data set, all possible divisions can count to 2^{n-1}-1, which is exponentially dependent on the growth of the data set size(Kaufman & Rousseeuw, 1990). Fortunately, there is no necessity to find an optimal division by examining all possible divisions. The exemplary algorithm introduced in the book was proposed by Macnaughton-Smith (MS) in 1964 to overcome the need for searching through all division patterns. DIANA-MS starts with one cluster consisted of all items and ends with singletons (each cluster only contains one item) with iterative steps. Each divisive step requires the calculation of dissimilarity matrix of the current cluster to be divided. The whole process initiates from forming a splinter group with a most dissimilar item in the parent cluster. The destination of remaining items will be carefully checked and relocated either to the splinter group or its sibling group one-by-one(Kaufman & Rousseeuw, 1990).

After decades, the divisive algorithm evolves, but along with agglomerative algorithm since they mirror each other. In this paper (Roux, 2018), M. Roux summarized the shared criteria between the divisive algorithm and agglomerative algorithm. As aforementioned (Kaufman & Rousseeuw, 1990), dissimilarity should be calculated. There are 9 criteria that can be used both for divisive and agglomerative algorithm. Out of the nine, five criteria based on distance are enlisted. The single linkage (shortest distance between clusters), average linkage (mean distance between clusters), and complete linkage (longest distance between clusters) have been taught in the CS410 course. One exception here is that complete linkage is adjusted for divisive algorithm. Instead of distance, the complete linkage is looking for the larger diameter between the clusters. Ward's original and its mutation (Szekely-Rizzo) are also distance based algorithm. Unlike the previous 5 mentioned distance type algorithm focusing only on the inter-cluster difference, ratio-type criteria consider both the intra- and inter-cluster difference. Here in M. Roux's paper (Roux, 2018), two criteria are explained, Dunn (Dunn, 1974) and Silhouette width (Kaufman & Rousseeuw, 1990). Dunn index can be simplified as the quotient of the inter-cluster dissimilarity divided by the intra-cluster dissimilarity. In contrast, Silhouette width represents the difference between inter-cluster dissimilarity and intra-cluster dissimilarity which is further normalized by the larger one of the two. To achieve a better division (aka, good bipartition), a larger dissimilarity criterion value should be met. Besides the dissimilarity criterion, node level is defined differently between divisive and agglomerative algorithms. Diameter of cluster and criterion value are used as the node level in divisive and agglomerative algorithms, respectively.

As discussed (Kaufman & Rousseeuw, 1990), a wise divisive method is not trying to exam all possibility. How to initiate a division and adjust the allocation of all the items is a priority. DIANA-MS is both discussed in the two papers (Kaufman & Rousseeuw, 1990; Roux, 2018) and the 1st paragraph of this manuscript. It starts with a most dissimilar item following with transfer function (the relocation of the remaining items). Other method to initiate the division is principal direction divisive partitioning algorithm (PDDP). PDDP (Roux, 2018) utilizes principal coordinates analysis (PCoA) to map all the items into a first principal coordinate axis. The items can be easily split into two subsets, one with negative coordinates while the other with positive coordinates or null. PDDP can be complemented with transfer function when it alone can not assign the correct division (Roux, 2018). M.Roux (Roux, 2018) used the Goodman-Kruskal‘s coefficient (G-K coef.) to cross-compare different algorithms practicing on the random data sets or real life datasets (e.g., Leukemia Dataset, Fisher’s Iris) and find the Silhouette width and DIANA-MS render the best results.

The quality of clustering results is not merely dependent on the algorithm. A proper workflow can help to improve the clustering. W. Wei (Wei et al., 2019) demonstrate a workflow working UCI benchmark datasets by splitting clusters based on selected parameters. E. Mangortey (Mangortey et al., 2020) meticulously performed a workflow with data pre-processing, clustering, and post-processing on aviation data for the analysis of prominent safety associated flight parameters. Following the Federal Aviation Administration (FAA) regulation, commercial flights are required to record thousands of parameters which could be irrelevant and redundant. In order to extract the most influential parameters apart from those confounding parameters, E. Mangortey (Mangortey et al., 2020)’s workflow utilized dimension reduction (e.g. t distributed stochastic neighbor embedding, t-SNE) to trim off the highly correlated parameters and refine the scope of parameters, clustering the dataset with conventional algorithms such as DIANA and K-mean clustering algorithm, and pinpointing the most influential parameters which show the biggest discrepancy of mean across all the clusters using the analysis of variance (ANOVA).The initial selection of a subset of parameters promotes the selection of the true influential parameters. Interestingly, in this paper (Mangortey et al., 2020), the clustering results are dramatically different given by a variety of clustering algorithms. DIANA divides the whole dataset (a training set with only 24 parameters into 2 clusters, while k-mean clustering algorithm yields eight optimal clusters. It seems like a larger number of clusters will facilitate the ANOVA.

There are many other challenges to apply a clustering algorithm like DIANA. For instance, the large dataset size imposes difficulties when applying conventional clustering algorithms. One plausible way to reduce the complexity of computation is parallelization of conventional clustering algorithm based on MapReduce (Praveen & Jayanth Babu, 2019). A decision diagram was shown to illustrate when parallel and distributed computing is required (Kurasova et al., 2014). Due to my limited understanding of the parallelization, this discussion will not be expanded here.

Due to the intrinsic nature of different clustering algorithm, each algorithm has its own application scope with respect to the data type. The report tabulates seven types of clustering algorithms and provides a clear vision of the suitability to seven various data types (Oyelade et al., 2019). DIANA, as a hierarchical method, suits the categorical and Time series dataset (Oyelade et al., 2019). Clearly, the partitional algorithm and model-based algorithm presents the broadest compatibility, working on 6 various data types including categorical, text, multimedia, uncertain, time series, and discrete data (Oyelade et al., 2019).

Conclusion

DIANA may not be the best performance clustering algorithm. But it provides hierarchy of the clusters and a unique perception of a dataset comparing to its counterparts. Like other clustering algorithms, DIANA is not a versatile solution for all clustering scenarios. But it suits relative dataset and provides an optimal analysis results in an empirical way. And it will be used more and more in a workflow with data cleaning, curation, and visualization instead of been applied alone.

Reference

Kaufman, L., & Rousseeuw, P. J. (1990). *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley.

Kurasova, O., Marcinkevicius, V., Medvedev, V., Rapecka, A., & Stefanovic, P. (2014). Strategies for Big Data Clustering. *Proceedings - International Conference on Tools with Artificial Intelligence, ICTAI*, *2014-December*, 740–747. https://doi.org/10.1109/ICTAI.2014.115

Mangortey, E., Monteiro, D., Ackley, J., Gao, Z., Puranik, T. G., Kirby, M., Pinon, O. J., & Mavris, D. N. (2020). Application of machine learning techniques to parameter selection for flight risk identification. *AIAA Scitech 2020 Forum*, *1 PartF*. https://doi.org/10.2514/6.2020-1850

Oyelade, J., Isewon, I., Oladipupo, O., Emebo, O., Omogbadegun, Z., Aromolaran, O., Uwoghiren, E., Olaniyan, D., & Olawole, O. (2019). Data Clustering: Algorithms and Its Applications. *Proceedings - 2019 19th International Conference on Computational Science and Its Applications, ICCSA 2019*, 71–81. https://doi.org/10.1109/ICCSA.2019.000-1

Praveen, P., & Jayanth Babu, C. (2019). Big Data Clustering: Applying Conventional Data Mining Techniques in Big Data Environment. *Lecture Notes in Networks and Systems*, *74*, 509–516. https://doi.org/10.1007/978-981-13-7082-3_58/FIGURES/2

Roux, M. (2018). A Comparative Study of Divisive and Agglomerative Hierarchical Clustering Algorithms. *Journal of Classification 2018 35:2*, *35*(2), 345–366. https://doi.org/10.1007/S00357-018-9259-9

Wei, W., Liang, J., Guo, X., Song, P., & Sun, Y. (2019). Hierarchical division clustering framework for categorical data. *Neurocomputing*, *341*, 118–134. https://doi.org/10.1016/J.NEUCOM.2019.02.043