Artificial Intelligence Research and Development
A. Cortés et al. (Eds.)

doi:10.3233/FAIA220335

Bootstrap-CURE Clustering: An Investigation of Impact of *Shrinking* on Clustering Performance

Ashutosh KARNA [a,1], Karina GIBERT [a]

[a] *Knowledge Engineering and Machine Learning Group at Intelligent Data Science and Artificial Intelligence Research Centre, Universitat Politècnica de Catalunya, Barcelona, Spain*

Abstract. Hierarchical clustering is one of the most popular techniques in unsupervised segmentation. However, since it has quadratic complexity as it is based on pairwise distance matrix construction, it tends to be less used with really large data cases. *CURE* clustering tackles this challenge by accelerating the process through a first hierarchical clustering over a smaller sample from which a set of representative points of resulting clusters is obtained and used to estimate the cluster shape. A *KNN* process with those representative points allows completing the cluster assignment to the remaining points. This clustering technique scales the hierarchical clustering to large datasets. This work is in continuation of the earlier research, *Bootstrap-CURE* which uses repeated samples in the first part of the process and gains both robustness and representativeness. Also, the proposed approach uses a criterion for automatic identification of the number of clusters from a dendrogram, so that the bootstrap samples can be automatically processed. In this paper, the concept of shrinkage is proposed as a hyperparameter to the *Bootstrap-CURE* clustering approach. The inclusion of shrinkage brings the proposed clustering technique closer to the original *CURE* clustering. The impact of shrinkage on the overall performance of *Bootstrap-CURE* is further explored. A real-life use case from 3D printers is presented to illustrate the performance of the proposed clustering.

Keywords. CURE clustering, Hierarchical clustering, Cluster validity indices, Bootstrapping, Dendrogram

1. Introduction

Clustering is one of the most important machine learning techniques to discover the hidden patterns in a dataset. The quality of clustering results depends on both the similarity metric and the implementation technique. Several works [2,16,1,15] can be found in the literature to help a researcher assess the pros and cons of various techniques and take a decision accordingly. Although, a major challenge that remains relevant is how the scale of data impacts the clustering performance. Techniques like *hierarchical clustering* [5]

[1]Corresponding Author: Ashutosh Karna, Knowledge Engineering and Machine Learning Group at Intelligent Data Science and Artificial Intelligence Research Centre, Universitat Politècnica de Catalunya, Catalonia, Spain; E-mail: ashutosh.karna@upc.edu

provide a tree-like graph, known as *dendrogram* that discloses the internal multivariate structure of the dataset and helps a researcher decide the appropriate number of partitions to make, but the quadratic time complexity of the algorithm prohibits its application on real-life datasets. Other techniques like *K-means* [7] do efficiently handle the large datasets but need a prior knowledge of K (number of clusters). Guha *et al.* proposed *CURE* (Clustering-Using-Representation) that scales hierarchical clustering to a large scale by incorporating a sampling strategy. In [13], Suman et al. proposed a new algorithm, *Bootstrap-CURE* that scales hierarchical clustering further by using several bootstrap samples in a *CURE* like strategy except the *shrinking* step. In this paper, the impact of *shrinking* and other related hyper-parameters are investigated on the overall clustering procedure in the context of the original dataset from 3D printing. The rest of the paper is structured as follows. Section 2 provides a summary of available research in this field, followed by a formal definition of the research problem in the section 3. Section 4 provides the proposed modification to original *Bootstrap-CURE* algorithm, followed by the methodology discussed in section 5. A summary of experiments conducted on a real-life dataset from 3D printing is discussed in section 6. The paper is finally concluded along with the discussion in section 7.

2. Literature Survey

As the size of the dataset increases, the increase in computational complexity makes it difficult to get clustering results in time, and thus traditional clustering methods are rendered impractical. Shirkhorshidi *et al.* [11] and Zerhari *et al.* [17] in their works, conducted a detailed investigation of clustering scenarios in context to big data and laid down the challenges in various methods. In both the works, the scale of data is rather suggested to be viewed in terms in *single vs multiple* machine-learning problem. For a single machine learning scenario, a small sample from the data is drawn from the larger dataset to use in clustering; while dimension-reduction techniques are used when the data is high-dimensional. Zhang *et al.* [18,19] introduced a novel clustering algorithm, called *BIRCH* (Balanced iterative reducing and clustering using hierarchies) to address the problem of processing large datasets with limited computing power. *BIRCH* introduces the concept of *Clustering-Factor* which is a triple (N, Ls, Ss) containing the number of items, linear sum, and squared sum of items in a subcluster respectively. *BIRCH* provides a two-step process, starting with one-pass scanning of the dataset and localized clusters are created. The localized clusters are expected to capture major patterns in the dataset and are further subject to another clustering. The method itself is based on top of the hierarchical clustering and scales it to big data.

McCallum *et al.* [8] proposed a canopy-based approach for clustering that is computationally cheap to estimate the distance matrix for large and high-dimensional datasets. The method efficiently divides the data into overlapping subsets, called *canopies* in the first stage, and then the distance measurements are computed in a common canopy.

CURE clustering [4] is another single machine learning technique that uses a set of representative points to estimate a cluster shape and shrink the representative points towards their respective cluster center.

In the earlier research, the authors [13,6] proposed a modification of *CURE* clustering using several bootstrap samples, however, the shrinking step from the original *CURE* algorithm is skipped.

Application of *shrinking* can be seen in several related works of clustering. In [14], Wang *et al.* proposed a new clustering algorithm based on local clustering that automates the discovery of clusters and right partitioning of data without any user input. In [3], Franti *et al.* proposed iterative-shrinking approach to clustering to obtain the suitable number of clusters. Shi *et al.* [10] uses shrinking as a data preprocessing step in high dimensional data clustering. In [9], Qian *et al.* proposed a modification of classical *CURE* clustering, called, *CURE-NS* which allows detecting non-spherical shaped clusters and ran experiments to compare the algorithm over the original *CURE* implementation.

3. Research Problem

Let us consider a multivariate numerical dataset, with the information about a set I of N, *k-dimensional* objects as $i_1, i_2, ...i_N$. The *Bootstrap-CURE* proposal [13] begins with drawing a small representative sample of ratio r from the original dataset, and divides it into S bootstrap samples of same size n_s without replacement. Each bootstrap sample is subject to hierarchical clustering individually and local clusters are obtained. A novel algorithm as described in [12] is used to deduce the number of clusters automatically. Further, let B_i represent i^{th} bootstrap sample and $c_{i,j}$ denote the local centroid of j^{th} cluster of i^{th} bootstrap sample. In the earlier research by the authors, the super-classification step involved applying a hierarchical clustering on all local centroids $(c_{i,j}, i \in 1,2,..S, j \in 1,2,..k_i)$, where k_i is the number of clusters in i^{th} bootstrap sample and super centroids are obtained. In the second-pass of the algorithm, the unsampled points $N*(1-r)$ are scanned and assigned to a super-cluster based on K-nearest-neighbor scheme.

The objective of this paper is to introduce the *shrinkage* step in the overall *Bootstrap-CURE* algorithm and assess its impact on the clustering results. In addition to *shrinkage*, following hyperparameters also play an important role in the quality of clustering and are evaluated as well.

1. **Sample Ratio** (r): Proportion of the original dataset drawn at random in step 1 of the Bootstrap CURE strategy.
2. **Number of Bootstrap samples** (S): Number of samples drawn without replacement from the initial sample obtained in step 1. Each bootstrap sample is of size $N*r/S$ and is individually clustered in step 2.
3. **Extreme or Reference points**(q): These refer to the points lying on the boundary of a cluster. They are identified as pairs of points with a bigger Euclidean distance between them, and capture the shape and extent of the cluster. The higher the value of q, better is the representation of the frontier of the cluster.
4. **Shrinkage** (α): It is a concept used in classical CURE definition. It implies moving extreme points towards the center of the cluster and using *shrunk* points as representatives of the cluster itself. Hence, the uncertainty associated with the frontier is reduced and robustness is gained. The higher the shrinkage rate, the closer are the reference points to the cluster centroid.

In this paper, the *Bootstrap-CURE* results with and without shrinking are compared.

4. Research Proposal

In the original *CURE* implementation, a set of extreme points in each cluster are selected as reference points which are then shrunk towards their respective cluster centers which help approximate the shape of each cluster in an unsupervised manner. In this research, the authors introduce and extend the concept of shrinking and the reference points to all the bootstrap samples.

The original *Bootstrap-CURE* algorithm is thus modified and summarised in the following steps:

1. **Initial Sampling phase**: Fix a sample ratio r and draw a random sample of size $N * r$ from the original dataset. Divide this random sample into S bootstrap samples, each of same size n_s, without replacement.
2. Subject each sample to hierarchical clustering (in this work with *Euclidean* distance and *Ward's* method), and obtain the S dendrograms.
3. Use method proposed in [12] to determine the number of clusters automatically.
4. Compute the centroid of all clusters found in the previous step and build a final dataset with all local centroids, represented as $c_{i,l}.i \in 1,2,..S, l \in 1,2,...k_i$, where k_i is the number of clusters detected in i^{th} sample.
5. **Super-classification phase**: Apply a hierarchical clustering on the local centroids dataset and compute super-centroids of each super-class. Let each super-centroid be denoted by $C_g, g \in 1,2,..k_g$ where k_g is the number of clusters in super-clustering step. Each C_g represent a centroid of a set of local centroids ($c_{i,l}$ and let n_{c_g} be the size of g^{th} super-cluster.
6. Fix a percentage q and compute $n_{c_g} * q$ extreme points in each super-cluster. Let $x_{g,t}, g \in 1,2,..k_g, t \in 1,2,..n_{c_g} * q$ represent the t^{th} extreme point in g^{th} super-cluster.
7. **Shrinking phase**: Fix a shrinkage percentage, α and shrink each extreme point $x_{g,t}$ towards its super-centroid C_g. This is done by computing a synthetic point by shrinking the euclidean distance by p percent between the extreme point and the super-centroid.
8. Trace all points from the sampled data belonging to each local centroid and in turn, to each super-centroid.
9. **Allocation phase**: For all $N * (1 - r)$ points which are not part of the original sample, assign the super-cluster based on the nearest shrunk extreme point.

In order to evaluate the impact of various hyperparameters (as mentioned in section 3), the original dataset is subject to *Bootstrap-CURE* with and without *shrinkage* respectively and the class assignments of each point in both the cases is tracked and assessed how the two approaches differ in the clustering results. The coincidences are computed for each cluster individually and the average coincidence rate (*ACR*) is obtained.

An illustrative example is discussed in section 6 using a real-life dataset from 3D printers.

5. Research Methodology

In this research, the authors are introducing the *Shrinkage* parameter during the super-clustering step of *Bootstrap-CURE* clustering. The objective of the study is to see how

various hyperparameters impact the clustering assignment in *Bootstrap-CURE* process. Hence, for each experiment, a pre-decided combination of hyperparameters is used and the data is subject to *Bootstrap-CURE* clustering with both *with* and *without* shrinkage, and the class assignment is studied.

Following the terminology defined in section 3, the general *Bootstrap-CURE* clustering can be formulated as the following:

$$\mathscr{B}_\alpha = \phi(r, S, \alpha, q); r \in [0,1], S >= 2, 0 <= \alpha < 1, 0 < q <= 1 \quad (1)$$

where α denotes the shrinkage.

The Eq 1 reduces to *Bootstrap-CURE clustering without shrinkage* when $\alpha = 0$, and let this be denoted by $\mathscr{B}_0$. Let k_α and k_0 denote the number of clusters obtained by using $\mathscr{B}_\alpha$ and $\mathscr{B}_0$ clustering algorithms respectively. Further, let A_k denote the coincidence for k^{th} cluster respectively and $\bar{A}$ denote the average coincidence rate (ACR) for overall data.

6. Application

The dataset used in the earlier research [13] has been continued for the current experiments. The data represent a collection of time-series-based sensor data from eight anonymous 3D printers with over 300 printing jobs. All experiments have been conducted on a GPU-enabled, four-core processor windows computer with 32 GB memory and *Python 3.6* has been used throughout for data analysis. The final working data after preprocessing contains 46821 records for 41 features. A list of features can be seen in the previous paper [13].

6.1. Bootstrap-CURE Clustering

In the earlier research [13], the authors proposed and conducted *Bootstrap-CURE* clustering on several samples of varying sizes drawn from the original dataset and subject them to hierarchical, *CURE* and *Bootstrap-CURE* clustering. While the number of clusters discovered remains the same (four clusters), the *Bootstrap-CURE* evidently showed a rapid decrease in the computation time as the dataset size increased. A summary of experimental results can be seen in [13].

6.2. Bootstrap-CURE with Shrinkage

In this research, the authors have modified the original *Bootstrap-CURE* algorithm with a new step, called *shrinking* added just before the final cluster-assignment stage. The cluster assignment before and after shrinkage is then tracked. Table 1 shows a schema of contingency matrix to compare the clustering assignments by the original (*without shrinkage*) and modified (*with shrinkage*) *Bootstrap-CURE* clustering algorithms, with O_{ij} representing the number of items classified into i^{th} cluster of original and j^{th} cluster of the modified *Bootstrap-CURE* algorithm. Further, coincidence rate for each individual cluster class (k= 1,2...k_0) is computed.

		Bootstrap-CURE with shrinkage			
		1	2	...	k_α
Bootstrap-CURE without shrinkage	1	O_{11}	O_{12}	...	O_{1k_α}
	2	O_{21}	O_{22}	...	O_{2k_α}
	...	...	...	...	...
	k_0	O_{k_01}	O_{k_02}	...	$O_{k_0k_\alpha}$

Table 1. Contingency matrix layout for *Bootstrap-CURE*

6.3. Impact of hyperparameters on Bootstrap-CURE

As discussed in section 5, the *Bootstrap-CURE* algorithm depends on 4 hyperparameters. For this research, the following range of values for these hyper-parameters have been tested.

- Initial sample ratio (r): $r \in [0.10, 0.20, 0.50, 0.75\ ]$
- Number of bootstrap samples (S): $S \in [5, 10, 15, 20\ ]$
- Percentage of extreme/reference points (q): $q \in [0.02, 0.05, 0.10, 0.20\ ]$
- Percentage of shrinkage (α) : $\alpha \in [0.05, 0.10, 0.15, 0.20\ ]$

A total of 256 experiments were conducted for every possible combination of the hyperparameters defined above. In the following, the main effect of each hyperparameter on the ACR is summarized.

6.3.1. Impact of Initial Sample Ratio (r)

The size of the initial sample drawn from the original data directly impacts the clustering quality. Fig 1 shows the impact of other hyperparameters on ACR conditioned to sample ratio. It is easily evident that a higher sample ratio ($r >= 0.5$) yields higher ACR for all hyper-parameters. Also, given a certain sample ratio, the ACR seems to be constant per shrinkage level, and increases with increase in percentage of extreme points. However, the trend is not clear regarding the number of bootstrap samples.

6.3.2. Impact of Number of Bootstrap samples (S)

In theory, the increase in the number of bootstrap samples improves the computational time of clustering as it allows parallel execution. However, as a hyperparameter, the number of bootstrap samples does not increase the ACR by itself. Conditioning to a given number of bootstrap samples, higher sample ratio has clear impact on better ACR but does not show any impact on other hyperparameters like shrinkage level or percent of extreme points as shown in Fig 2.

6.3.3. Impact of Shrinking Percentage (α)

Four levels of shrinkage have been considered for the experiments. In line with what was observed in previous graphs, the shrinkage level is not determining an average improvement on ACR. When conditioning to a given shrinkage level, one can see that the ACR increases with the sample ratio, and a very weak improvement is observed with the increase in percentage of extreme points. The figure has similar pattern as fig 2

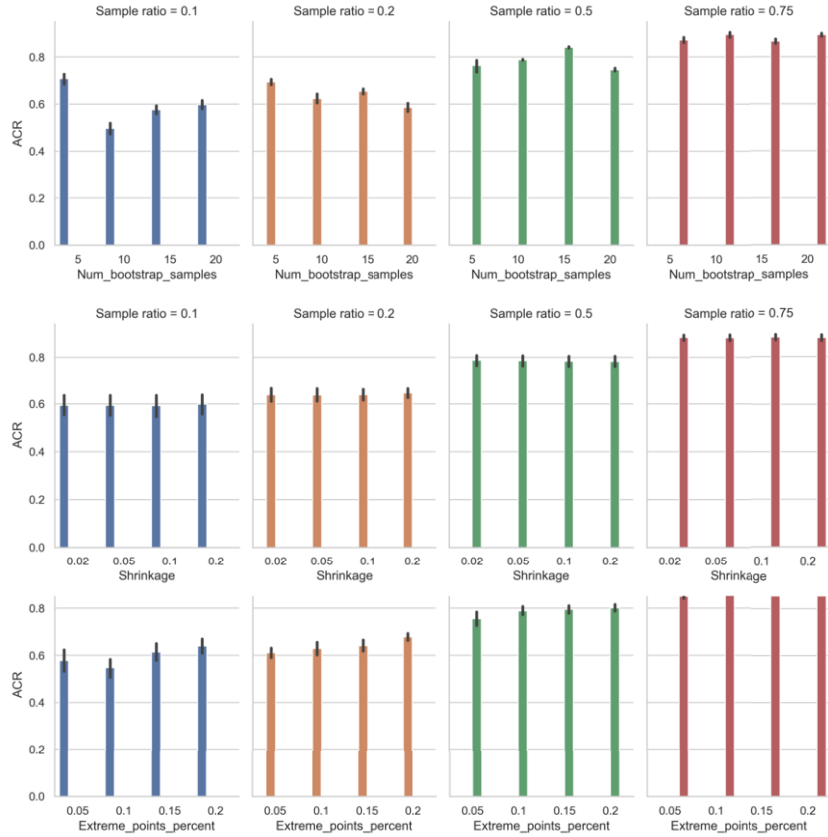

Figure 1. Impact of sample ratio

6.3.4. Impact of Percentage of extreme points (q)

The reference points are used during the super classification step of the *Bootstrap-CURE*. Fig 1 shows that the percent of reference points do not affect the average ACR by itself. Conditioning to a given percent of extreme points, the ACR increases with the sample ratio. The figure has similar pattern as 2.

7. Discussion and Conclusion

In this paper the original *Bootstrap-CURE* algorithm is modified by introducing shrinking after the super-classification step and consequently, the clustering assignments are evaluated both with and without shrinkage. The research is an attempt to discover the intrinsic nature of hyper-parameters and their interactions in a *Bootstrap-CURE* clustering procedure.

In general, the increase in *initial-sample-ratio* always increases the quality of the clustering. However, a higher sample ratio also implies a higher computational cost as the size of the data to be clustered increases. A lower sample ratio reduces the computation cost but the representation of the data adversely affects the clustering quality. A bal-

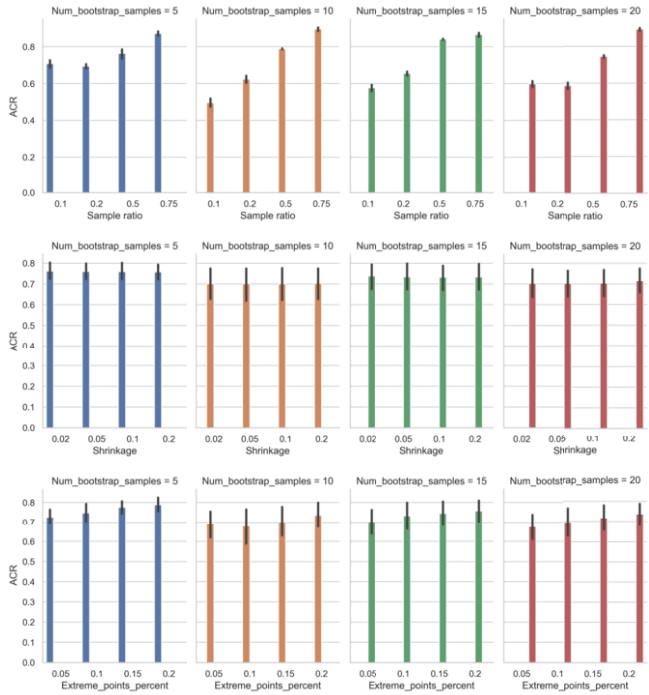

Figure 2. Impact of number of bootstrap samples

ance between sample ratio and quality (ACR) should be sought by the researchers. In the earlier research work, the authors used $r = 0.5$ as the initial sample ratio and this seems reasonable to the current experimental results. The number of bootstrap samples is another important factor that impacts the computational cost in *Bootstrap-CURE*. The experiments show when the number of bootstrap samples is fewer, the sample ratio can be increased to achieve better clustering quality. The application of shrinkage, however, is quite agnostic to the quality of clusters, and it seems that this is due to the topology of the analyzed data. This datasegt has a sufficiently strong structure with fewer observations in the areas of inter-clusters frontiers that might oscillate between one cluster to another depending on the shrinkage level. When the percentage of reference points increases, we get a better representation of the cluster shape and ACR shows improvement.

The experimental results indicate that the introduction of shrinkage does not impact the clustering quality for the tested domain. However, the number of bootstrap samples and the initial sample ratio play a rather more important role in deciding the overall quality, as well as the number of extreme points.

In the future lines, further experimentation is planned with a synthetic dataset with different topologies, so as to verify how the shrinkage percentage impacts ACR in different kinds of data structures. In theory, shrinkage improves the robustness of clusters as it manages the outliers better. Further analysis is being conducted to check the outliers' structure of the target dataset. From the computational point of view, introducing shrinkage does not show an observable impact as it can be implemented like a scaling operation on observation vectors with a fraction of milliseconds in computational costs.

Mathematical optimization of the hyperparameters has not been part of this research but would be very useful in the upcoming research in order to finetune the clustering results.

This research directly fits into a bigger project that aims at developing an intelligent decision support system for enterprise-scale 3D printers and the clustering procedure is aimed at the automatic discovery of patterns without human intervention. In the future line of research, the authors also aim to build the layout of decision support system and investigate how the clustering scheme fits into the overall strategy of detecting different operating modes in the machine during the runtime.

References

[1] Sabhia Firdaus and Md Uddin. A survey on clustering algorithms and complexity analysis. *International Journal of Computer Science Issues (IJCSI)*, 12(2):62, 2015.

[2] Chris Fraley and Adrian E Raftery. How many clusters? which clustering method? answers via model-based cluster analysis. *The computer journal*, 41(8):578–588, 1998.

[3] Pasi Fränti and Olli Virmajoki. Iterative shrinking method for clustering problems. *Pattern Recognition*, 39(5):761–775, 2006.

[4] Sudipto Guha, Rajeev Rastogi, and Kyuseok Shim. Cure: An efficient clustering algorithm for large databases. *Information Systems*, 26(1):35–58, 3 2001.

[5] Stephen C Johnson. Hierarchical clustering schemes. *Psychometrika*, 32(3):241–254, 1967.

[6] Ashutosh Karna and Karina Gibert. Automatic identification of the number of clusters in hierarchical clustering. *Neural Computing and Applications*, pages 1–16, 2021.

[7] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA, 1967.

[8] Andrew McCallum, Kamal Nigam, and Lyle H Ungar. Efficient clustering of high-dimensional data sets with application to reference matching. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 169–178, 2000.

[9] Yun-Tao Qian, Qing-Song Shi, and Qi Wang. Cure-ns: A hierarchical clustering algorithm with new shrinking scheme. In *Proceedings. International Conference on Machine Learning and Cybernetics*, volume 2, pages 895–899. IEEE, 2002.

[10] Yong Shi, Yuqing Song, and Aidong Zhang. A shrinking-based clustering approach for multidimensional data. *IEEE Transactions on Knowledge and Data Engineering*, 17(10):1389–1403, 2005.

[11] Ali Seyed Shirkhorshidi, Saeed Aghabozorgi, Teh Ying Wah, and Tutut Herawan. Big data clustering: a review. In *International conference on computational science and its applications*, pages 707–720. Springer, 2014.

[12] Shikha SUMAN, Ashutosh KARNA, and Karina GIBERT. Towards expert-nspired automatic criterion to cut a dendrogram for real-industrial applications. *ARTIFICIAL INTELLIGENCE RESEARCH AND DEVELOPMENT*, page 235, 2021.

[13] Shikha Suman, Ashutosh Karna, and Karina Gibert. Bootstrap–cure: A novel clustering approach for sensor data—an application to 3d printing industry. *Applied Sciences*, 12(4):2191, 2022.

[14] Xiaogang Wang, Weiliang Qiu, and Ruben H Zamar. Clues: A non-parametric clustering method based on local shrinking. *Computational Statistics & Data Analysis*, 52(1):286–298, 2007.

[15] Rui Xu and Donald Wunsch. Survey of clustering algorithms. *IEEE Transactions on neural networks*, 16(3):645–678, 2005.

[16] Mohamed Zait and Hammou Messatfa. A comparative study of clustering methods. *Future Generation Computer Systems*, 13(2-3):149–159, 1997.

[17] Btissam Zerhari, Ayoub Ait Lahcen, and Salma Mouline. Big data clustering: Algorithms and challenges. In *Proc. of Int. Conf. on Big Data, Cloud and Applications (BDCA'15)*, 2015.

[18] Tian Zhang, Raghu Ramakrishnan, and Miron Livny. Birch: an efficient data clustering method for very large databases. *ACM sigmod record*, 25(2):103–114, 1996.

[19] Tian Zhang, Raghu Ramakrishnan, and Miron Livny. Birch: A new data clustering algorithm and its applications. *Data mining and knowledge discovery*, 1(2):141–182, 1997.