A New Clustering Algorithm Based on Regions of Influence with Self-Detection of the Best Number of Clusters

Fabrice Muhlenbach, Stéphane Lallich

A New Clustering Algorithm Based on Regions of Influence with Self-Detection of the Best Number of Clusters

Fabrice Muhlenbach
Université de Lyon, CNRS, UMR 5516
Laboratoire Hubert Curien, Saint-Étienne, France
fabrice.muhlenbach@univ-st-etienne.fr

Stéphane Lallich
Université de Lyon, Laboratoire ERIC (EA 3083)
Université Lumière Lyon 2, Bron, France
stephane.lallich@univ-lyon2.fr

***Abstract*—Clustering methods usually require to know the best number of clusters, or another parameter, e.g. a threshold, which is not ever easy to provide. This paper proposes a new graph-based clustering method called "GBC" which detects automatically the best number of clusters, without requiring any other parameter. In this method based on regions of influence, a graph is constructed and the edges of the graph having the higher values are cut according to a hierarchical divisive procedure. An index is calculated from the size average of the cut edges which self-detects the more appropriate number of clusters. The results of GBC for 3 quality indices (Dunn, Silhouette and Davies-Bouldin) are compared with those of K-Means, Ward's hierarchical clustering method and DBSCAN on 8 benchmarks. The experiments show the good performance of GBC in the case of well separated clusters, even if the data are unbalanced, non-convex or with presence of outliers, whatever the shape of the clusters.**

***Keywords*-clustering; neighborhood graph.**

I. Introduction

Clustering is an unsupervised machine learning task used successfully in many fields, including image analysis, bioinformatics and marketing. Given a set of n data objects described by p attributes, clustering procedures aim at grouping the objects into clusters such that similar objects are in the same cluster and dissimilar objects are in separate clusters. In this paper, we are interested with crisp clustering methods which consider that each object belongs to exactly one cluster at the contrary of fuzzy clustering.

Partitioning clustering aims at directly decompose the set of objects into $k^\star$ disjoint subsets, where $k^\star$ has to be predetermined. K-Means [1] is a popular iterative reallocation algorithm due to its simplicity and low complexity. Among the variants of K-Means, PAM [2] is more robust against the noise and CLARANS [3] is more efficient and scalable.

Hierarchical algorithms produce a hierarchy of partitions. According to the direction of the procedure, at each step these algorithms either merge the most similar clusters (agglomerative procedure, e.g., using Ward's criterion [4]) or split the most heterogeneous cluster (divisive procedure). Among others one can cite BIRCH [5] which concentrates on densely occupied portions of the space, CURE [6] which uses multiple representatives for each clusters to "capture" the shape of each one, and ROCK [7], or its improved version Q-ROCK [8], that does not employ distances but links when it merges the clusters.

Density-based clustering algorithms are designed to discover arbitrary-shaped clusters. Clusters are considered as dense regions of objects in the data space that are separated by regions of low density. DBSCAN [9] is a typical density-based algorithm. The core point in DBSCAN is that for a fixed radius, the neighborhood of each object in a cluster has to contain at least a minimum number of other objects.

Grid-based algorithms partition the data space into a finite number of cells to form a grid structure, and determine the cells whose density exceed a prefixed threshold. Then they perform all clustering operations on the obtained grid structure (e.g., STING [10] or CLIQUE [11]).

Graph-theoretic approaches are able to discover non usual data structures. The simplest algorithm is founded upon the minimum spanning tree (MST) [12]. First the minimum spanning tree is built, where the nodes are the objects and the edges sizes are the Euclidean distances in the objects space. The edges having the greatest size are removed to construct disjoint connected components.

The purpose of our work is to extend Zahn [12] and Urquhart [13] graph-theoretic approach by showing an easy way of selecting the "ideal" number of clusters. In section II, we discuss the different state-of-the-art clustering methods and investigate how they process to choose the best number of clusters. We will show that no one can give an appropriate answer because if they do not ask explicitly the user to give them this information, they asked him on other parameters that he can not give a response without having an intuition on the domain. This problem is fixed in our new method, presented in section III, which is a divisive data set processing that tries to find automatically the good number of clusters. In the section IV we perform an extensive experimental evaluation with different clustering quality indices obtained on different clustering methods.

II. How to Choose the Number of Clusters?

On Table I, we list the characteristics of main clustering methods found in the data mining literature.

Algorithm Category	Method	Parameters / Properties
Hierarchical clustering	Ward's algorithm [4]	agglomerative algorithm
	MST Divisive Hierarchical Clustering [12]	based on graph theory
	Clustering Using REpresentatives (CURE) [6]	each cluster is represented by a set of representatives
	RObust Clustering using linKs (ROCK) [7]	$k^\star$: number of clusters
	Q-ROCK [8]	θ: similarity threshold
Hard Clustering	k-Means [1], [14]	$k^\star$: number of clusters
Density-based clustering	DBSCAN [9]	ϵ: distance to consider that 2 points are neighbors or not
Sequential clustering	Basic Sequential Algorithm Scheme (BSAS) [15]	Θ: dissimilarity threshold $k^\star$: max. number of clusters

Table I
MAIN CLUSTERING METHODS

For the hierarchical clustering algorithms, the choice of the best number of clusters is not a simple task. Of course, a dendogram obtained with Ward's agglomerative method or the MST divisive algorithm can help the user, but in many cases it is only the knowledge of the domain that can be used to identify what is the best clustering. When no information is given to set $k^\star$, the number of clusters, the user have to provide a specific parameter, like a similarity (or dissimilarity) threshold Θ, or to rely on some heuristic arguments, like a $k^\star = f(n)$ function that will provide a number of clusters depending on n, the size of the data set.

The density-based algorithms, like DBSCAN [9], DBCLASD or DENCLUE, consider that clusters are dense regions of points. These methods can handle arbitrary shaped clusters, outliers, and their time complexity is lower than other clustering methods. Unfortunately, the choice of the parameters required for these methods (e.g., ϵ for DBSCAN) may lead to different results and it is not easy to discover which one is the best without doing several experiments.

The clustering algorithms based on regions of influence are an extension version of the MST divisive algorithm [12] aforesaid. These methods, based on the graph theory, are capable of detecting clusters for various shapes. These algorithms can provide good results when the clusters are well separated in the representation space. The key idea of these algorithms is to construct a graph by connecting with an edge each pair of points that are alone in a given region of influence, e.g. the relative neighborhood graph (RNG) [16] or the Gabriel graph (GG) [13], then removing the edges that are inconsistent compared with their neighboring edges. An edge is considered as inconsistent if it is greater of q standard deviations (typically $q = 2$) of the mean of a given number of its neighbors, which makes subjective this procedure. Nevertheless, an advantage of these algorithms is that the results do not depend on the order in which the data are considered and no initial conditions are required.

Finally an ideal method does not exist. In our opinion, a good clustering method has to be easy to implement and to use (no specific parameter values have to be asked to the user), and has to be suitable for various shapes of clusters.

III. GBC METHOD

A. Neighborhood Graphs

The neighborhood graphs, which are special tools of the computational geometry, can be used in the clustering algorithms based on regions of influence, but also in many other data mining tasks, and especially in the supervised machine learning [17], [18], [19], [20]. Such neighborhood structure can be for example the k-nearest neighbor, the Delaunay triangulation, the MST, the relative neighborhood graph (RNG), the Gabriel graph (GG).

For each graph, a specific condition is required, depending on a region of influence, to link two points with an edge. For the MST, the condition is to connect all vertices together with the minimal size of edges; for the RNG, the region of influence is a lune, the intersection of two hyperspheres centred on each pair of points; and for the GG, the region of influence is an hypersphere with each pair as a diameter.

B. GBC, a New Clustering Algorithm

GBC, the new clustering method proposed in this paper, is conducted in 2 phases. The first phase (in 10 steps) consists in doing a list of μ values which will be used in the formula 1 to detect the appropriate number of clusters and is conducted as written on Table II.

1	construction of a neighborhood graph NG
2	descent sorting (by size) of the edge set E of NG
3	initialization step: $k \leftarrow 2$, $\Sigma_k \leftarrow 0$, and $n_e \leftarrow 1$
4	cutting the edge $e_{\max} \in E$ of the (sub-)graph with the higher value
5	adding the size of $e_{\max}$ to Σ_k, the sum of the cut edges at the level k
6	testing if the (sub-)graph with the edges $E - \{e_{\max}\}$ is still connected
7	if the graph is still connected, increasing the number of edges: $n_e \leftarrow n_e + 1$
8	if the graph is not connected, modifying: $k \leftarrow k + 1$ (new level for having k clusters) $n_e \leftarrow 1$ (re-initialization of the number of cut edges) $\mu_k \leftarrow \Sigma_k / n_e$ (μ_k is the average of the sizes of the cut edges) $\Sigma_k \leftarrow 0$ (re-initialization of the sum of cut edge sizes)
9	$E \leftarrow E - \{e_{\max}\}$ (remove the bigger edge from the set E)
10	back to step 4 by using the next maximal edge $e_{\max} \in E$ while $k < n$

Table II
GBC ALGORITHM

After this first phase, we can calculate the δ values for each level k as follows:

$$\delta_k = \frac{\mu_k - \mu_{k+1}}{\mu_k + \mu_{k+1}}, \forall k = 2, \ldots, n-1 \ . \qquad (1)$$

The maximal value of δ_k is used to select $k_{\delta\_\max}$, the *ideal* number of clusters in the data set. The second phase is similar to the first one, but the loop runs until $k = k_{\delta\_\max}$ in the step 10 (instead of $k < n$).

Notice that we can equally use the MST, RNG or GG for the neighborhood graph on the step 1 of the algorithm. In the experiments, we did not find significant differences in the results, but following [13], we recommend the RNG of Toussaint [16] which is a structure that overcome some problems encountered with the MST.

C. Main Characteristics of δ

The first phase considers all the data as a unique set and tries to separate this set in subsets as it is done for the descendant hierarchical clustering method, but in a much lower time complexity thanks to the graph structure. At each level k, the size average of the cut edges μ_k is calculated.

Since the set of edges E is sorted by descending size, μ_k is a monotone descending function of k and can not be used as it is for finding the more relevant number of clusters.

The value $\mu_k - \mu_{k+1}$ represents the decreasing of μ while getting from a partition of k clusters to $k+1$ clusters. The parameter δ_k allows to normalize this difference by taking into account μ_k and μ_{k+1}. The splitting procedure continues while this normalized difference δ_k is increasing.

D. Properties of the Method

First, GBC is very sensitive to the outliers [21], because an outlier will be far from the other data in the representation space, and it will be detected as an independent cluster.

Secondly, GBC is limited to hard clustering: it will fail to detect different clusters if there is a recovery between them. If the clusters are not well-separated in the representation space, there will not be an edge with a big size between the clusters in the neighborhood graph.

IV. Experimentations

A. Methodology

For this study, we have compared GBC with 3 other clustering methods: K-Means [1], Ward's hierarchical clustering method [4] and DBSCAN [9]. For DBSCAN, we have used the value of ϵ given by the mean value of the 4-nearest neighbors (4-NN) of the edges of the data set [9], and we have increased this value (indicated $\epsilon^{\nearrow}$ or $\epsilon^{\nearrow\nearrow}$) for having better results (and less clusters).

Three quality indices have been calculated on Table III: Dunn's [22], Silhouette [23], and the Davies-Bouldin's [24] indices. In addition to these results, Table III indicates for each data set the numbers of individuals (Indiv.), of attributes (Attr.), of clusters discovered with GBC ($k_{\delta\_\max}$), of real classes or clusters for the domain when it exists (k_r). The number of clusters used for the test (k_u), and the respective sizes of the clusters obtained with these 3 different methods are also shown on this table.

B. Quality Indices in Clustering

1) Dunn's Index: This index, which tries to obtain well-separated sets with examples very closely located to each others, is calculated with the dissimilarity measure between two clusters and the diameter measure of a cluster. Large values of Dunn's index correspond to good cluster partitions. The Dunn's index has two major drawbacks: its computation requires a considerable amount of time, and this index is very sensitive to the noise and to the outliers.

2) Silhouette Index: The global silhouette partition, which is the average of all example silhouettes, takes its value between -1 and 1, and the partition with the maximum global silhouette is considered as the optimal partition.

3) Davies-Bouldin's Index (D.-B.): D.-B. reaches to minimize the average similarity between the different clusters, so D.-B. is a function of the ratio of the sum of within-cluster scatter to between-cluster separation. D.-B. exhibits no trends with respect to the number of clusters. The smaller D.-B. is, the better the partition is considered.

C. Data Sets

For these experimentations, we used 5 real benchmarks from the UCI Repository [25], 3 artificial data sets (*test-2c-2o*, *test-random*, *yin-yang*), and *ruspini* [26].

The data sets were chosen to represent a variety within a specific class of data sets characterized by: (1) small number of true classes, which may or may not correspond to coherent clusters; (2) moderate number of observations; (3) moderate number of features; (4) numerical attributes (continuous or multivalued values) for calculating the distances. The observations with missing values have been removed from the data sets, all values have been transformed with the Milligan and Cooper method [27] and the Euclidian distance has been used for all the clustering methods.

D. Global Results

On Table III, the best values of the clustering validity indices are emphasized in a bold font. The experimental setup allows to compare the performances of GBC, K-Means and Ward's hierarchical clustering 11 times, because we used the 8 data sets described in the section IV-C, and 3 of them were used with two different numbers of classes (*auto-mpg*, *e-coli* and *iris*). On Table III, the performance of each algorithm on each data set is evaluated according to the 3 quality indices described in the section IV-B.

In the case of the Dunn index, GBC outperforms K-Means and Ward's hierarchical clustering. Compared to K-Means, GBC improves significantly the value of the Dunn index 8 times out of 11 and there are 3 equalities (p-value of the sign test = 0.008). Compared to Ward's hierarchical clustering, GBC improves the value of the Dunn index 7 times out of 11, with 3 equalities and 1 defeat (p-value = 0.070).

According the Davies-Bouldin index, GBC has better results than K-Means (7 wins, 2 defeats, 2 equalities), but this superiority is not significant (p-value = 0.180). The results of GBC and WHC are quite the same (5 wins, 4 defeats, 2 equalities).

The performances of the three algorithms according to the Silhouette index are very close (GBC wins 3 times, K-Means 4 times, WHC 2 times, and 2 equalities). It is not easy to compare the performances of the three algorithms with those of DBSCAN because DBSCAN determines automatically the number of classes in function of the values chosen for

the two parameters of the algorithm. Only 4 comparisons can be achieved (*iris*, *ruspini*, *test-2c-2o*, *yin-yang*). For each of them, the results of DBSCAN are identical to those of GBC.

E. Detailed Results

Auto MPG is generally used for a regression task: the continuous class variable is to be predicted in terms of 3 multivalued discrete and 5 continuous attributes. Consequently, there is no "true real" number of classes to discover. GBC is not able to find an ideal number of clusters ($k_{\delta\_\max} = 388 \simeq 392 = n$) but a real number of cluster k_r does not exist for this data set. GBC obtains however the best result with the Dunn index for $k^\star = 5$ clusters.

Breast Cancer Wisconsin (Original) is a data set that contains 8 attributes and is used in a Boolean classification task. Here, GBC is not able to find an ideal number of clusters (336 seems to be irrelevant) but nevertheless, when it has to detect 2 clusters, it obtains good results with Dunn and D.-B. indices, just by isolating one outlier.

Ecoli Data Set contains 7 predictive attributes used to predict the cellular localization sites of proteins. They are 8 possible sites. With this data set, GBC does not detect the 8 classes but find 3 clusters; the 3 quality indices obtained with this method pass all the others.

Iris Data Set contains 4 predictive attributes with 3 classes of 50 instances each, where each class refers to a type of iris plant. One class is linearly separable from the other two; the latter are not linearly separable from each other. All the methods are able to discover the 2 clusters, but the GBC method gives an interesting result when it is forced to produce 3 clusters (one outlier detected, and best value with the D.-B. index).

Ruspini [26] is a data set consisting of 75 points in four groups with 2 variables giving the x and y coordinates. On this case, the results of the different methods are the same.

Test 2 Clusters with 2 Outliers (or *Test-2c-2o*) consists of 402 instances on 2 variables (the x and y coordinates) with 2 groups of 200 observations each and 2 outliers between the groups. Only GBC and DBSCAN are able to detect the 2 clusters and the 2 outliers.

Test with Random values (or *Test-Random*) is an artificial data set consisting of 1000 instances on 2 variables (the x and y coordinates) where the values have been randomly generated. The value of $k_{\delta\_\max} = 999$ obtained is close to $n = 1000$, indicating that the method is not able to find a structure in the data set (which is an appropriate behavior with random data).

Yin and Yang is a non-convex artificial data set consisting of 1000 instances randomly extracted from the black points of a *Yin and Yang* picture (the Chinese symbol). The 2 variables correspond to the x and y coordinates. GBC and DBSCAN are able to find the appropriate clusters. The Silhouette index is not the best in this case because it penalizes a partition with unbalanced clusters.

V. Conclusion and Future Work

The new clustering method that we proposed in this paper is in the line with the graph-theoretic approaches. GBC has been tested with 3 quality indices and compared to other clustering methods on 8 benchmarks. On artificial data sets where the clusters are well represented, GBC outperforms the other methods. On real data sets extracted from the UCI Repository, when clusters can intrinsically be found in the data, GBC detects automatically the correct cluster number and identify the relevant clusters. On the other hand, when the class clusters are not well separable, the method tends to "peel" the global clusters in isolated points.

Even if GBC is more time consuming than DBSCAN or K-Means, its computational complexity is equivalent to a hierarchical agglomerative clustering due to the neighborhood graph construction. We have also to remind that it is limited to the hard clustering and will fail if there is a recovery between the different clusters.

Nevertheless, this method has many advantages. First, we have shown that GBC obtains good results when the data are well-structured: it detects easily the well-formed clusters and the outliers, whether the cluster shapes are convex or not, whether the cluster sizes are homogeneous or not. Second, the main advantage of GBC is that the method does not need any parameter to perform on a data set. It is a considerable improvement compared with other clustering techniques developed in the data mining literature. Even for DBSCAN, which can discover the best number of clusters, the quality of the results is associated to one or more specific parameters (here, the size of the radius ϵ and the minimal number of neighborhood points for considering a core point).

Third, when they are some outliers in a data set, GBC can automatically find them, without any parameters, as well as DBSCAN when the ϵ parameter is well chosen. Fourth, GBC provides a dendogram –like the other hierarchical cluster algorithms– but can find automatically the ideal number of clusters. Fifth, it can be associated to a visualization method for neatly navigating into the data. This method is useful to prepare the data and to find the border points between clusters. Sixth, when there is no a-priori structure in the data of the set, GBC will indicate it by providing a value $k_{\delta\_\max}$ close to the number of examples in the data set. And seventh, while processing on a divisive way in a moderate computer complexity, GBC is better to represent the way of a human being processes (e.g., Zahn's approach has been driven by psychological considerations [12]).

For our future work, we plan to add another quality index adapted to graph-based clustering, in a similar way of the indices proposed by [28] which is an adaption of the Davies-Bouldin's index to neighborhood graphs. And finally we will make a more specific study of the outliers that GBC detects in a more selective way than DBSCAN.

Data set	Indiv.	Attr.	$k_{\delta\_\max}$	k_r	k_u	Method	Dunn	Silh.	D.-B.	Size of the clusters
auto-mpg	392	8	388	—	2	GBC	0.059	0.531	2.141	{245;147}
					2	K-Means	0.019	**0.634**	2.435	{176;216}
					2	WHC	0.061	0.559	**2.062**	{103;289}
auto-mpg	392	8	388	—	5	GBC	**0.074**	0.412	4.082	{175;69;1;68;79}
					5	K-Means	0.001	0.408	3.978	{20;178;73;42;79}
					5	WHC	0.007	0.502	3.828	{103;100;73;72;44}
auto-mpg	392	8	—	ϵ = 4-NN	152	DBSCAN	0.003	0.124	10674.887	{13;8;73;46;20;42;45;1×145}
				$\epsilon\nearrow$	35	DBSCAN	0.005	0.109	5773.051	{100;71;71;53;67;1×30}
breast	683	9	336	2	2	GBC	**0.127**	0.498	**0.982**	{1;682}
					2	K-Means	0.024	**0.754**	1.719	{231;452}
					2	WHC	0.012	0.708	1.471	{424;259}
breast	683	9	—	ϵ = 4-NN	282	DBSCAN	0.000	0.492	2395.722	{402;1 ×145}
				$\epsilon\nearrow$	218	DBSCAN	0.000	0.685	2174.749	{435;11;10;6;8;1×213}
e-coli	336	7	3	8	3	GBC	**0.494**	**0.724**	**1.002**	{1;9;326}
					3	K-Means	0.001	0.181	20.262	{118;109;109}
					3	WHC	0.005	0.569	19.479	{151;83;102}
e-coli	336	7	3	8	8	GBC	0.057	0.306	156.603	{1;3;1;1;4;1;1;324}
					8	K-Means	0.001	0.166	13.859	{11;31;16;83;19;51;107;18}
					8	WHC	0.004	0.367	12.008	{31;66;54;53;82;20;10;20}
e-coli	336	7	—	ϵ = 4-NN	85	DBSCAN	0.010	-0.221	4150.664	{159;84;11;1×82}
				$\epsilon\nearrow$	74	DBSCAN	0.007	-0.427	4380.764	{263;1×73}
				$\epsilon\nearrow\nearrow$	7	DBSCAN	0.017	0.681	2958.936	{1;1;326;5;1;1;1}
iris	150	4	2	3	2	GBC	**0.128**	**0.809**	1.301	{50;100}
					2	K-Means	**0.128**	**0.809**	1.301	{50;100}
					2	WHC	**0.128**	**0.809**	1.301	{50;100}
iris	150	4	2	3	3	GBC	0.039	0.700	**1.172**	{49;1;100}
					3	K-Means	0.002	0.542	1.543	{25;75;50}
					3	WHC	0.013	0.688	11.965	{50;67;33}
iris	150	4	—	ϵ = 4-NN	58	DBSCAN	0.000	0.265	3384.088	{43;21;17;8;8;1×53}
				$\epsilon\nearrow$	25	DBSCAN	0.003	0.121	2261.534	{49;78;1×23}
				$\epsilon\nearrow\nearrow$	2	DBSCAN	**0.128**	**0.809**	1.301	{50;100}
ruspini	75	2	4	4	4	GBC	**0.271**	**0.910**	**4.160**	{20;15;23;17}
					4	K-Means	**0.271**	**0.910**	**4.160**	{20;15;23;17}
					4	WHC	**0.271**	**0.910**	**4.160**	{20;15;23;17}
ruspini	75	2	—	ϵ = 4-NN	38	DBSCAN	0.005	0.736	1236.230	{9;6;20;6;1×34}
				$\epsilon\nearrow$	7	DBSCAN	0.014	0.849	367.630	{20;15;23;14;1;1;1}
				$\epsilon\nearrow\nearrow$	4	DBSCAN	**0.271**	**0.910**	**4.160**	{20;15;23;17}
test-2c-2o	402	2	4	4	4	GBC	**0.107**	**0.829**	**1.843**	{200;200;1;1}
					4	K-Means	0.000	0.634	10.837	{83;68;200;51}
					4	WHC	0.001	0.733	20.779	{201;96;53;52}
test-2c-2o	402	2	—	ϵ = 4-NN	200	DBSCAN	0.000	0.455	19763.053	{198;6;1×198}
				$\epsilon\nearrow$	4	DBSCAN	**0.107**	**0.829**	**1.843**	{1;1;200;200}
test-random	1000	2	999	—	4	GBC	**0.001**	-0.593	5.250	{1;1;995;3}
					4	K-Means	0.000	**0.584**	7.198	{233;245;236;286}
					4	WHC	**0.001**	0.465	**5.188**	{345;286;262;107}
test-random	1000	2	—	ϵ = 4-NN	611	DBSCAN	0.000	0.504	15192.390	{11;17;8;...;1×187}
				$\epsilon\nearrow$	123	DBSCAN	0.000	-0.428	8090.912	{370;14;171;32;81;...;1×103 }
yin-yang	1000	2	2	2	2	GBC	**0.023**	0.146	**1.063**	{31;969}
					2	K-Means	0.000	**0.586**	3.449	{527;473}
					2	WHC	0.000	0.529	1.583	{319;681}
yin-yang	1000	2	—	ϵ = 4-NN	388	DBSCAN	0.001	0.169	4733.136	{12;21;44;...;1×343}
				$\epsilon\nearrow$	2	DBSCAN	**0.023**	0.146	**1.063**	{969;31}

Table III
EXPERIMENTAL RESULTS OBTAINED ON THE 8 DATA SETS.

REFERENCES

[1] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*. Berkeley: University of California Press, 1967, vol. 1, pp. 281–297.

[2] L. Kaufman and P. J. Rousseeuw, *Finding Groups in Data: an Introduction to Cluster Analysis*. New-York: John Wiley and Sons, 1990.

[3] R. T. Ng and J. Han, "Efficient and effective clustering methods for spatial data mining," in *Proceedings of the 20th International Conference on Very Large Data Bases (VLDB'94)*, J. B. Bocca, M. Jarke, and C. Zaniolo, Eds. Morgan Kaufmann, 1994, pp. 144–155.

[4] J. H. Ward, "Hierachical grouping to optimize an objective function," *Journal of the American Statistical Association*, vol. 58, pp. 236–244, 1963.

[5] T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: an efficient data clustering method for very large databases," in *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data (SIGMOD'96)*. ACM Press, 1996, pp. 103–114.

[6] S. Guha, R. Rastogi, and K. Shim, "CURE: an efficient clustering algorithm for large databases," in *Proceedings of ACM SIGMOD International Conference on Management of Data (SIGMOD 1998)*, L. M. Haas and A. Tiwary, Eds. ACM Press, 1998, pp. 73–84.

[7] ——, "ROCK: A robust clustering algorithm for categorical attributes," in *Proceedings of the 15th International Conference on Data Engineering (ICDE'99)*. IEEE Computer Society, 1998, pp. 512–521.

[8] M. Dutta, A. Kakoti Mahanta, and A. K. Pujari, "QROCK: A quick version of the ROCK algorithm for clustering of categorical data," *Pattern Recognition Letters*, vol. 26, no. 15, pp. 2364–2373, 2005.

[9] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96)*, E. Simoudis, J. Han, and U. M. Fayyad, Eds. AAAI Press, 1996, pp. 226–231.

[10] W. Wang, J. Yang, and R. R. Muntz, "Sting: A statistical information grid approach to spatial data mining," in *Proceedings of 23rd International Conference on Very Large Data Bases (VLDB'97)*. Morgan Kaufmann, 1997, pp. 186–195.

[11] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan, "Automatic subspace clustering of high dimensional data for data mining applications," in *SIGMOD 1998, Proceedings ACM SIGMOD International Conference on Management of Data*, L. M. Haas and A. Tiwary, Eds. ACM Press, 1998, pp. 94–105.

[12] C. T. Zahn, "Graph-theoretical methods for detecting and describing Gestalt clusters," *IEEE Transactions on Computers*, vol. C, no. 20, pp. 68–86, 1971.

[13] R. Urquhart, "Graph theoretical clustering based on limited neighbourhood sets," *Pattern Recognition*, vol. 15, no. 3, pp. 173–187, 1982.

[14] S. P. Lloyd, "Least squares quantization in PCM," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–136, 1982.

[15] S. Theodoridis and K. Koutroumbas, *Pattern Recognition, Fourth Edition*. Academic Press, 2009.

[16] G. T. Toussaint, "The relative neighbourhood graph of a finite planar set," *Pattern Recognition*, vol. 12, no. 4, pp. 261–268, 1980.

[17] F. Muhlenbach and R. Rakotomalala, "Multivariate supervised discretization, a neighborhood graph approach," in *Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM 2002)*, V. Kumar, S. Tsumoto, N. Zhong, P. S. Yu, and X. Wu, Eds. IEEE Computer Society, 2002, pp. 314–321.

[18] F. Muhlenbach, S. Lallich, and D. A. Zighed, "Identifying and handling mislabelled instances," *Journal of Intelligent Information Systems (JIIS)*, vol. 22, no. 1, pp. 89–109, 2004.

[19] D. A. Zighed, S. Lallich, and F. Muhlenbach, "A statistical approach to class separability," *Applied Stochastic Models in Business and Industry*, vol. 22, no. 2, pp. 187–197, 2005.

[20] G. T. Toussaint, "Geometric proximity graphs for improving nearest neighbor methods in instance-based learning and data mining," *International Journal of Computational Geometry and Applications (IJCGA)*, vol. 15, no. 2, pp. 101–150, 2005.

[21] D. M. Hawkins, *The Identification of Outliers*. London: Chapman and Hall, 1980.

[22] J. Dunn, "Well separated clusters and optimal fuzzy partitions," *Journal of Cybernetics*, vol. 4, pp. 95–104, 1974.

[23] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53–65, 1987.

[24] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE Transactions on Pattern Recognition and Machine Intelligence*, vol. 1, no. 2, pp. 224–227, 1974.

[25] A. Asuncion and D. J. Newman, "UCI Machine Learning Repository [http://www.ics.uci.edu/∼mlearn/MLRepository.html]. Irvine, CA: University of California, School of Information and Computer Science," 2007.

[26] E. H. Ruspini, "Numerical methods for fuzzy clustering," *Information Sciences*, vol. 2, no. 3, pp. 319–350, 1970.

[27] G. W. Milligan and M. C. Cooper, "A study of standardization of variables in cluster analysis," *Journal of Classification*, vol. 5, pp. 181–204, 1988.

[28] N. R. Pal and J. Biswas, "Cluster validation using graph theoretic concepts," *Pattern Recognition*, vol. 30, no. 6, pp. 847–857, 1997.