New Version of Davies-Bouldin Index for Clustering Validation Based on HyperRectangles

Juan Carlos Rojas Thomas
Facultad de Informática
Universidad Complutense de Madrid
Madrid, España
jc.rojas.thomas@gmail.com

Marco Mora Cofre
Departamento de Ciencias de la Computación
Universidad Católica del Maule
Talca, Chile
marcomoracofre@gmail.com

Matilde Santos
Facultad de Informática
Universidad Complutense de Madrid
Madrid, España
msantos@ucm.es

***Abstract*— this paper presents a new version of Davies-Bouldin index for clustering validation through the use of Hyper Rectangles for measuring the clusters dispersion. This new technique allows capturing the multidimensional reality of the data overcoming the limitations of the traditional measures. The new version of the index is evaluated with real datasets, showing the effectiveness of the proposal.**

Keywords— Clustering, Davies-Bouldin Index, HyperRectangles;

I. INTRODUCTION

The process of clustering consists on classifying in an unsupervised way a set of patterns (observations or data) into groups (clusters) [1]. In general, the clustering methods should search for clusters whose members are close to each other (in other words have a high degree of similarity) and well separated [2]. One of the most important issues in cluster analysis is the evaluation of clustering results to find the partition that best fits the underlying data. This is the main subject of cluster validity [2].

In general, there are three approaches to investigate cluster validity: external criteria, internal criteria and relative criteria [3]. Clustering validity approaches, which are based on relative criteria, aim at finding the best clustering scheme that a clustering algorithm can define under certain assumptions and parameter. Here the basic idea is the evaluation of a clustering structure by comparing it to other clustering schemes, resulting by the same algorithm but with different parameter values [4]. The Davies-Bouldin index [5] falls into the latter category. Such indexes are used when the partitions generated by the applied clustering algorithm are no overlapping, meaning by this that each data belongs strictly to an only one class [4].

The Davies-Bouldin index is based on the approximately estimation of the distances between clusters and their dispersion to obtain a final value that represents the quality of the partition. The traditional measure used by this index for measuring the dispersion of the clusters is the average distance of the data to the centroid of the cluster. This measure has the drawback that is not able to capture the multidimensionality of the data, causing, for example, that two clusters with different dispersions in one of the dimensions of the feature space have the same value for this statistics, as the Figure 1 shows.

Proposals to modify these estimates based on graphs [6], although they have improved its performance, continue to have essentially the same drawbacks.

In other hand, the concept of HyperRectangles has been widely used in a specific type of clustering algorithms called "Grid-based Clustering Algorithms". These algorithms generally partitioned the feature space into a finite number of cells, and then, according to the density of these cells the clustering is performed [7]. Examples of these algorithms are GDILC [8] and WaveCluster [9]. The GRIDCLUS algorithm [10] defines the volume of each HyperRectangular cell as the multiplication of the extent of the cell in each dimension of the space.

In order to improve the performance of the Davies Bouldin index, this paper proposes the use of HyperRectangles to estimate the dispersion of the clusters in each one of its dimensions, but instead of performing a partition of the whole space we find the HyperRectangles which best fit the data of each particular cluster. With this purpose the reference system is adapted to each cluster using Principal Components Analysis. Then the dispersion of the data is represented by the HyperVolume of the HyperRectangle associated to each cluster. This technique is validated by comparing its results with the original index and its variations that have been proposed in the literature.

This paper presents the following structure: first a description of the original index, then shows the variations of the index based on graphs, which have been suggested in the literature. Below is presented the proposed technique, its

comparative performance related to the original index and its variations based on graphs, and finally the conclusions and future works.

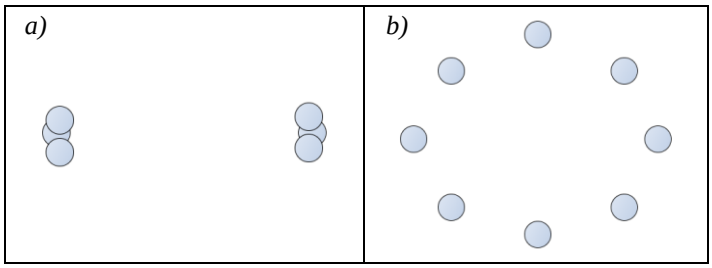


Fig. 1. The two symmetric clusters, *a* and *b*, show the same dispersion value using the average distance to their centroids.

II. DAVIES-BOULDIN INDEX

This index (DB) is based on the idea that for a good partition the inter cluster separation as well as intra cluster homogeneity and compactness should be high [6]. Then, to define the DB index, we need to define the dispersion measure and the cluster similarity measure [7]. In [1] the dispersion S_i of C_i cluster (1) and the separation D_{ij} between ith and jth clusters (2) are defined as:

$$S_i = \left(\frac{1}{|C_i|} \sum_{x \in C_i} D^p(x, c_i) \right)^{\frac{1}{p}}, p > 0 \quad (1)$$

Where $|C_i|$ is the number of data points in cluster C_i and c_i is the center of cluster C_i, and:

$$D_{ij} = \left(\sum_{l=1}^{d} |v_{il} - v_{jl}|^t \right)^{\frac{1}{t}}, t > 1 \quad (2)$$

Where v_i and v_j are the centroids of clusters C_i and C_j, respectively.

Then, the DB index is defined as:

$$V_{DB} = \frac{1}{k} \sum_{i=1}^{k} R_i \quad (3)$$

Where k is the number of clusters and R_i is defined as:

$$R_i = \max_{i \neq j} R_{ij} \quad (4)$$

Where R_{ij} is the similarity measure between clusters C_i and C_j, and is defined as:

$$R_{ij} = \frac{S_i + S_j}{D_{ij}} \quad (5)$$

Since the goal is to achieve minimum within-cluster dispersion and maximum between-cluster separation, the number of clusters c that minimizes V_{DB} is taken as the optimal value of c [6].

III. VERSIONS BASED ON GRAPHS

In [5] the Davies-Bouldin index is generalized through the use of graphs. Specifically, they use Minimal Spanning Tree (MST), Relative Neighborhood Graph (RNG) and Gabriel Graph (GG) to obtain the dispersion of each cluster. The measure of distance between clusters is still the distance between the means.

A. Minimal Spanning Tree (MST)

Corresponds to a tree that connects all nodes of a graph, and whose total sum of weights of its edges is the smallest of all possible configurations [6].

B. Relative Neighborhood Graph (RNG)

Let x_i, x_j be two data points. They are connected in the RGN if

$$d(x_i, x_j) \leq \max\{d(x_i, x_k), d(x_j, x_k) \forall k, k \neq i, k \neq j\} \quad (6)$$

Where $d(x_i, x_j)$ is the Euclidean distance between x_i and x_j. In other words, x_i, x_j are connected in RNG if no other point falls in LUNE(x_i, x_j), where LUNE(x_i, x_j) is the intersection of the two disks of radius $d(x_i, x_j)$ and having centers at x_i and x_j. This is the region of influence of RNG [6].

C. Gabriel Graph (GG)

The points x_i, x_j are connected in GG if

$$d^2(x_i, x_j) < d^2(x_i, x_k) + d^2(x_j, x_k) \forall k, k \neq i, k \neq j \quad (7)$$

In other words, x_i, x_j are connected in GG if no other point lie in DISK(x_i, x_j), where DISK(x_i, x_j) is the disk with diameter $d(x_i, x_j)$ centered at the mid-point of x_i and x_j [6].

D. Cluster Dispersion

A graph $G=(V,E)$ is a pair where $V=\{v_1,v_2,\ldots,v_m\}$ is a set of vertices and $E=\{e_1,e_2,\ldots,e_n\}$ is a set of distinct edges. Each edge $e_q=\{x_1, x_j\}$ connects a pair of vertices.

Given a partition with c clusters, the diameter of the class i, which corresponds with the measure of dispersion, is defined as:

$$d_i^* = \max\{e_{ij}^*, j = 0,1,\ldots, l_i, l_i = |E_i|\}, i = 0,1,\ldots, c \quad (8)$$

Where *=GG or RNG or MST.

IV. THE PROPOSAL

The proposal consists in to estimate the clusters dispersion as the HyperVolume of the HyperRectangle which best fits the data. The HyperRectangles can be calculated in three different ways. These three versions of the proposal are the "Average HyperRectangle", the "Percentile HyperRectangle" and the "Maximum HyperRectangle".

The details of the steps are now explained in the following subsections, and illustrated graphically in the Figure 2. Examples of the HyperRectangles created using the different versions of the proposal are illustrated by the Figures 4 and 5.

A. Local Reference System

The first step in this procedure is to obtain the local reference system of each cluster. This local reference system is obtained through the use of Principal Component Analysis over the data of the clusters. As a result the axes of this local reference system correspond to the normalized eigenvectors of the covariance matrix of the data cluster centered at its centroid.

B. Grouping of Data

The second step is the grouping of data in two sets, the positive set and the negative set, with the objective of adapting the HyperRectangles to asymmetric clusters. This classification process is performed over a specific axis of the local system of reference each time, and uses an hyperplane than passes through the centroid and is perpendicular to the local axis that is being analyzed. In order to determine the membership of the data to the positive or negative set, the algorithm obtains the distance vectors of each data from the centroid, and then calculates the dot product between the distances vectors and the eigenvector that represents the local axis, the result of which corresponds to the projections in the corresponding dimension. This is symbolized as follows:

$$P_j^i = \left| \vec{d}_j \bullet \vec{v}_i \right| \tag{9}$$

Where P_j^i is the projection over the *i-th* dimension of the *j-th* distance vector, d_j is the distance vector of the *j-th* data from the centroid of its cluster and v_i is the *i-th* eigenvector of the same cluster.

Depending on the sign of this product, data are grouped into one of the two sets. Let D^k be the set of the all distance vectors belonging to the cluster *k*. Then the two sets for the *i-th* local axis are determined as follows:

$$D^{k+}{}_i = \left\{ \vec{d}_j, \vec{d}_j \in D^k \wedge \left| \vec{d}_j \bullet \vec{v}^k{}_i \right| > 0 \right\} \tag{10}$$

$$D^{k-}{}_i = \left\{ \vec{d}_j, \vec{d}_j \in D^k \wedge \left| \vec{d}_j \bullet \vec{v}^k{}_i \right| < 0 \right\} \tag{11}$$

Where *i* corresponds to the *i-th* dimension of the local system of reference and $v^k{}_i$ is the *i-th* eigenvector of the cluster *k*.

C. Lengh of the Sides

The third step is to obtain the length of each side of the HyperRectangle. The lengths of the sides of the HyperRectangle are calculated as a function of the projections of each one of the distance vectors in the axes of the local reference system.

For each local axis, the length of the HyperRectangle is calculated for the positive and the negative set of data in an independent way, then both are added to obtain the total length. Let P_i^{k+} be the set of the projections of the distance vectors that belong to $D^{k+}{}_i$ over the *i-th* local axis, and P_i^{k-} the set of the projections of the distance vectors that belong to $D^{k-}{}_i$ over the same axis. This is symbolized as follows:

$$L_i^+ = f(P_i^{k+}) \tag{12}$$

$$L_i^- = f(P_i^{k-}) \tag{13}$$

$$L_i = L_i^+ + L_i^- \tag{14}$$

Where L_i corresponds to the total length of the HyperRectangle over the *i-th* local axis and *f* corresponds to the function of the projections used to calculate the length.

D. Calculation of the HyperVolume

Finally, the last step is to calculate the HiperVolume of the HyperRectangle as follows:

$$HV = \prod_{i=1}^{n} L_i \tag{15}$$

Where *n* is the number of dimensions.

E. Minimum Side Lenght

Because there is the possibility that some clusters have a cero projection over one or more axes of the local system of reference it is necessary to define a default value, L_{MIN}, to be assigned to the corresponding side of the HyperRectangle. This situation is illustrated by the Figure 3.

F. Average HyperRectangle Version

The lengths of the HyperRectagles sides are calculated as the average projections of the distance vectors, as follows:

$$c^+ = \left| P_i^{k+} \right| \tag{16}$$

$$c^- = \left| P_i^{k-} \right| \tag{17}$$

$$f(P_i^{k+}) = \frac{\sum_{j=1}^{c+} p_j}{c^+}, \quad p_j \in P_i^{k+} \tag{18}$$

$$f(P_i^{k-}) = \frac{\sum_{j=1}^{c-} \left| p_j \right|}{c^-}, \quad p_j \in P_i^{k-} \tag{19}$$

G. Percentile HyperRectangle Version

The length of the HyperRectagles sides is equal to the projection of which absolute value is greater or equal than a certain percentage of the projections (their absolute values). Let *Sort()* be a function that orders the projections sets as follows:

$$Sort(P_i^+) = (p_1, p_2, ..., p_{c+}), \ p_{t-1} \le p_t \le p_{t+1} \tag{20}$$

$$Sort(P_i^-) = (p_1, p_2, ..., p_{c-}), \ \left| p_{t-1} \right| \le \left| p_t \right| \le \left| p_{t+1} \right| \tag{21}$$

Let *Element()* be a function that recovers the *t-th* greater projection of a sorted set:

$$Element(Sort(P_i^+),t) = p_t \quad (22)$$

Let *perc* be the percentage of projections smaller than the one which we want to use as the representative length of the set. Then the positions of the projections inside the sorted sets are:

$$t^+ = round\left(\frac{perc * c^+}{100}\right) \quad (23)$$

$$t^- = round\left(\frac{perc * c^-}{100}\right) \quad (24)$$

Where *round* is an approximation function to an integer value. Then the functions to obtain the length in the *i-th* dimension are:

$$f(P_i^{k+}, perc) = Element(Sort(P_i^+),t^+) \quad (25)$$

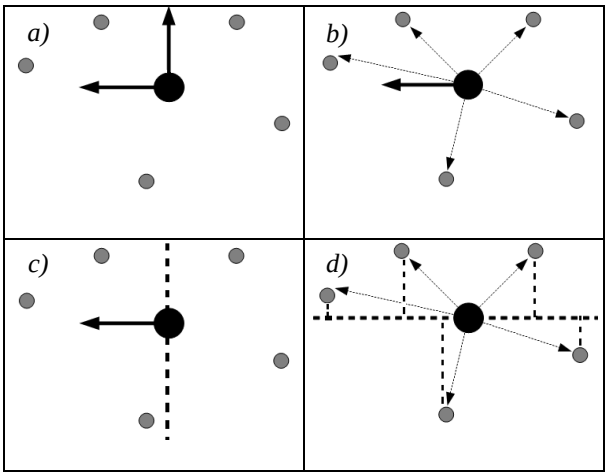
$$f(P_i^{k-}, perc) = Element(Sort(P_i^-),t^-) \quad (26)$$

H. Maximum HyperRectangle Version

The length of the HyperRectagles sides is equal to the projection of which absolute value is the greatest in each one of the projection sets. Then, using the prior definitions the lengths are obtained as follows:

$$f(P_i^{k+}) = Element(Sort(P_i^+),c^+) \quad (27)$$

$$f(P_i^{k-}) = Element(Sort(P_i^-),c^-) \quad (28)$$



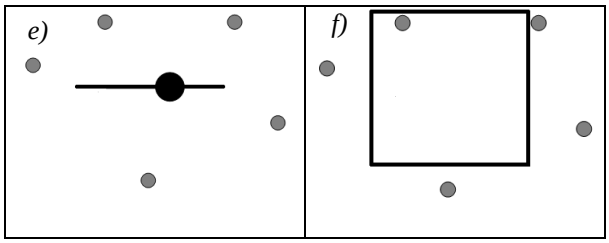


Fig. 2. The images show the process of calculating the HyperRectangle dimensions. First (a) the normalized eigenvectors of the cluster are obtained, creating the local system of reference and one is selected to start working with. Then (b) the distance vectors of the data (gray circles) from the centroid of the cluster (black circle) are generated. In (c) an hyperplane perpendicular to the selected eigenvector passing through the centroid of the cluster is created to split the data in two sets. In (d) the projections of the distance vector over the eigenvector are calculated. Based on these values (e) the length of the HyperRectangle is obtained over the local axis. The process is repeated to the remaining eigenvectors and finally the lengths of the all sides are obtained (f).

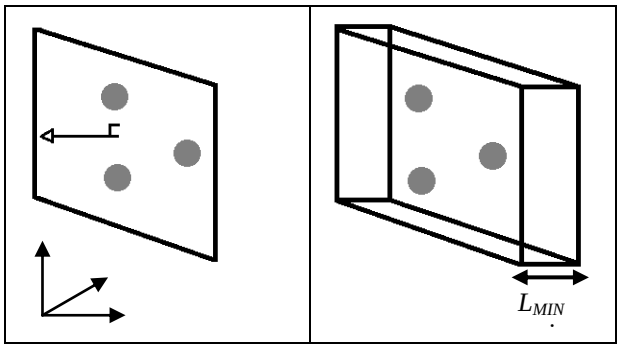
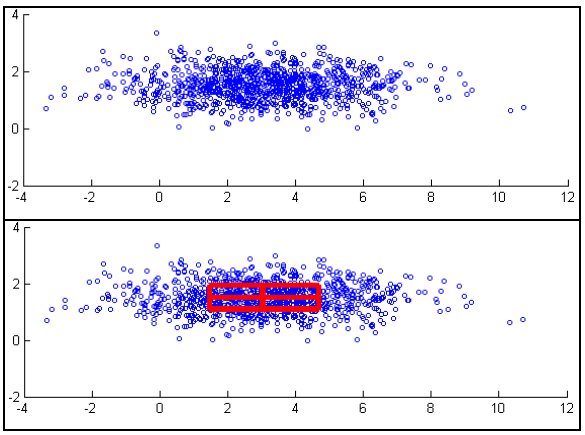


Fig. 3. Three data points in a three dimensional space determine an hyperplane. Then the projections of the distance vectors of the data from the cluster centroid over the eigenvector perpendicular to the hyperplane will be zero. Then a default value, L_{MIN}, is assigned to the corresponding side of the HyperRectangle.



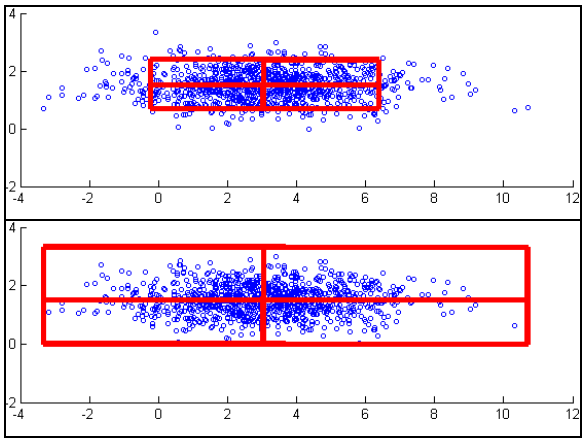


Fig. 4. The images show the HyperRectangles created over the same cluster with normal distribution using the three different versions. From top to down, the original cluster, using the Average version, using the Percentile version with a value of 90 and finally the Maximum version.

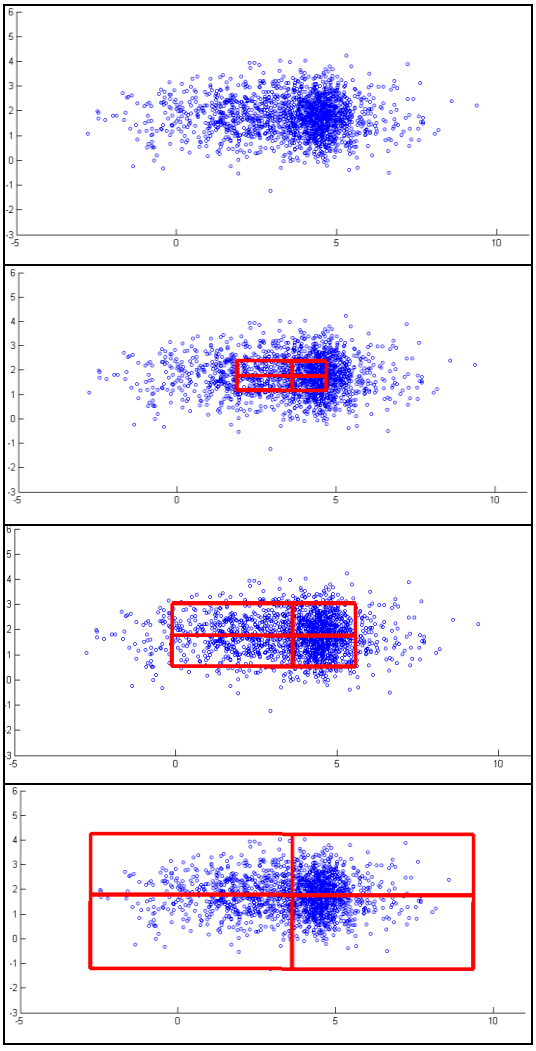


Fig. 5. The images show the HyperRectangles created over the same cluster with asymmetric distribution using the three different versions. From top to down, the original cluster, using the Average version, using the Percentile version with a value of 90 and finally the Maximum version.

V. RESULTS

A series of experiments on different sets of test were made to compare the performance of the proposed work and its different versions with the original Davies Bouldin index and the other implementations based on graphs. In the process of evaluation the Rand index was used, which allows to measure the level of similitude between two partitions, with values ranging from zero (minimal similitude) to one (maximal similitude) [7]. Another coefficient that was used is the Pearson correlation coefficient. This can take values from -1 to +1. A value of +1 show that the variables are perfectly linearly related by an increasing relationship, a value of -1 show that the variables are perfectly linearly related by an decreasing relationship, and a value of 0 show that the variables are not linearly related to each other. There is considered a strong correlation if the correlation coefficient is greater than 0.8 and a weak correlation if the correlation coefficient is less than 0.5 [11].

The methodology used was the following: applying the clustering algorithm *k-means* with variable parameters, was obtained 20 different partitions on each dataset used. Then, for each partition obtained, were calculated the next versions of the proposal (with a L_{MIN} value equal to 0.1% of the data range values, previously scaled to a 0-100 range): the average HyperRectangle (DB^{H}_{AV}), the 75 percentile HyperRectangle (DB^{H}_{75P}), the 90 percentile HyperRectangle (DB^{H}_{90P}) and the maximum HyperRectangle (DB^{H}_{MAX}), together with the original Davies-Bouldin index (*DB*) and the different implementation based on graphs, the Minimal Spanning Tree version (DB^{MT}), the Gabriel Graph version (DB^{GG}), and the Relative Neighborhood Graph version (DB^{NG}). Then to measure the accuracy of each Davies Bouldin version the Rand index was obtained with the objective of seeing the similitude between the partition and the original classes of the data set and finally the Pearson correlation between each version of the DB index and the Rand index was calculated. This process was applied over 7 real datasets. These are the IRIS data set (3 classes, 4 features and 150 instances), Breast Cancer Wisconsin (Diagnostic) data set (2 classes, 30 features and 569 instances), Wine data set (3 classes, 13 features and 178 instances), Vertebral Column data set (3 classes, 6 features and 310 instances), Ecoli dataset (8 classes, 7 features and 336 instances), Habersman Survival data set (2 classes, 3 features and 306 instances) and Breast Tissue data set (6 classes, 9 features and 106 instances) [12]. The Table 1 shows the values of Pearson coefficient obtained by each of the indexes evaluated over the different data sets, highlighting the best values obtained for each dataset. Because the objective of the Davies-Bouldin index and its derivatives is to be minimized, a high negative value in the Pearson coefficient indicates a good performance of the index.

VI. CONCLUSIONS AND FUTURE WORKS

The results show the effectiveness of the proposal. In five of the datasets one of the proposal versions showed the best performance, specially the average HyperRectangle version, and in the Iris data set where the performance of the proposal was overcome by the Gabriel Graph version, the values showed by the percentile versions (75 and 90) and the

maximum version were very similar or better than the original Davies Bouldin index and were not so below of the performance of the versions based on graph. Only in the Wine dataset the proposal performance was well under the rest of the indexes, the original and the based on graph versions.

Future works are focused on improving the proposal throw the creation of new measures of distances between the clusters based on the HyperRectangles created, and the extension of the comparisons of the proposal's performance against another clustering validation indexes.

References

[1] A.K. Jain, M.N. Murty, O.J. Flynn "Data Clustering: a review", ACM Computing Surveys, Vol. 31, No. 3, September 1999

[2] M. Halkidi, Y. Batistakis, M. Vazirgiannis, "On Clustering Validation Techniques", Intelligent Information Systems Journal, Kluwer Pulishers, 17(2-3): 107-145, 2001

[3] M. Halkidi, Y. Batistakis, M. Vazirgiannis. "Cluster Validity Methods: Part I", SIGMOD Record, June 2002

[4] M. Halkidi, Y. Batistakis, M. Vazirgiannis. "Cluster Validity Methods: Part II", SIGMOD Record, September 2002.

[5] D.L. Davies and D.W. Bouldin, "A Cluster Separation Measure," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 1, pp. 224-227, 1979.

[6] Nikhil R. Pal, J. Biswas: "Cluster validation using graph theoretic concepts". Pattern Recognition 30(6): 847-857 (1997)

[7] Guojun Gan, Chaoqun Ma, Jianhong Wu, "Data Clustering Theory, algorithms and applications" SIAM, Society for Industrial and Applied Mathematics (May 30, 2007)

[8] Zhao, S. and Song, J.(2001). "GDLIC: a grid-based density-isoline clustering algorithm". In Proceedings of the international conferences of info-tech and info-net,2001, volume 3, pages 140-145. Beijing: IEEE.

[9] Gholamhosein Sheikholeslami, Surojit Chatterjee, Aidong Zhang: "WaveCluster: A Wavelet Based Clustering Approach for Spatial Data in Very Large Databases". VLDB J. 8(3-4): 289-304 (2000)

[10] Schikuta, E. (1996). Grid-clustering: An efficient hierarchical clustering method for very large data sets. In Proceedings of the 13th International Conference on Pattern Recognition, volume 2, pages 101-105. Vienna, Austria: IEEE

[11] Sorana-Daniela BOLBOACĂ, Lorentz JÄNTSCHI, "Pearson versus Spearman, Kendall's Tau Correlation Analysis onStructure-Activity Relationships of Biologic Active Compounds", Leonardo Journal of Sciences, Issue 9, July-December 2006, p. 179-200

[12] Frank, A. & Asuncion, A. (2010). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

TABLE I.

Data set	DB	DB^{H}_{AV}	DB^{H}_{75P}	DB^{H}_{90P}	DB^{H}_{MAX}	DB^{GG}	DB^{MT}	DB^{NG}
B. Tissue	0.899057	**-0.291923**	0.904321	0.907228	0.914605	0.875026	0.882708	0.895302
B. Cancer	0.923769	0.791332	**-0.533505**	-0.330626	0.114576	0.758401	0.278733	0.289874
Column	0.614279	**-0.707301**	0.579498	0.482950	0.955490	0.102795	-0.013005	-0.014689
Ecoli	0.317982	**-0.023975**	-0.001866	-0.001866	-0.001866	0.507281	0.653676	0.638393
Habersman	0.976166	**-0.513818**	0.980458	0.979221	0.946005	0.951881	0.940705	0.947711
Iris	-0.596236	0.426395	-0.595097	-0.625890	-0.642793	**-0.687051**	-0.652730	-0.659146
Wine	**-0.913682**	0.358154	-0.321274	-0.163013	-0.462310	-0.889307	-0.774965	-0.807921