Peer-reviewed academic journal

Innovative Issues and Approaches in Social Sciences

IIASS – VOL. 8, NO. 1, JANUARY 2015

Innovative Issues and Approaches in Social Sciences

IIASS is a double blind peer review academic journal published 3 times yearly (January, May, September) covering different social sciences: political science, sociology, economy, public administration, law, management, communication science, psychology and education.

IIASS has started as a SIdip – Slovenian Association for Innovative Political Science journal and is now being published in the name of CEOs d.o.o. by Zalozba Vega (publishing house).

Typeset
This journal was typeset in 11 pt. Arial, Italic, Bold, and Bold Italic; the headlines were typeset in 14 pt. Arial, Bold

Abstracting and Indexing services
COBISS, International Political Science Abstracts, CSA Worldwide Political Science Abstracts, CSA Sociological Abstracts, PAIS International, DOAJ.

Publication Data:
CEOs d.o.o.

Innovative issues and approaches in social sciences, 2015, vol. 8, no. 1

ISSN 1855-0541

Additional information: www.iiass.com

PERFORMANCE OF SELECTED AGGLOMERATIVE HIERARCHICAL CLUSTERING METHODS

Nusa Erman[1], Ales Korosec[2], Jana Suklan[3]

Abstract

A broad variety of different methods of agglomerative hierarchical clustering brings along problems how to choose the most appropriate method for the given data. It is well known that some methods outperform others if the analysed data have a specific structure. In the presented study we have observed the behaviour of the centroid, the median (Gower median method), and the average method (unweighted pair-group method with arithmetic mean – UPGMA; average linkage between groups). We have compared them with mostly used methods of hierarchical clustering: the minimum (single linkage clustering), the maximum (complete linkage clustering), the Ward, and the McQuitty (groups method average, weighted pair-group method using arithmetic averages - WPGMA) methods. We have applied the comparison of these methods on spherical, ellipsoid, umbrella-like, "core-and-sphere", ring-like and intertwined three-dimensional data structures. To generate the data and execute the analysis, we have used R statistical software. Results show that all seven methods are successful in finding compact, ball-shaped or ellipsoid structures when they are enough separated. Conversely, all methods except the minimum perform poor on non-homogenous, irregular and elongated ones. Especially challenging is a circular double helix structure; it is being correctly revealed only by the minimum method. We can also confirm formerly published results of other simulation studies, which usually favour average method (besides Ward method) in cases when data is assumed to be fairly compact and well separated.

Keywords: hierarchical clustering, agglomerative methods, divisive methods, simulated data.

DOI: http://dx.doi.org/10.12959/issn.1855-0541.IIASS-2015-no1-art11

[1] Nuša Erman, Ph.D. is the associate of the School of Advanced Social Studies, Nova Gorica, Slovenia. Contact: nusa.erman (at) gmail.com

[2] Aleš Korošec is a data analyst at Health data center at the National Institute of Public Health, Ljubljana, Slovenia. Contact: ales.korosec (at) nijz.si

[3] Jana Suklan is a teaching assistant of the School of Advanced Social Studies, Nova Gorica, Slovenia. Contact: jana.suklan (at) fuds.si

Introduction

Clustering objects into groups is one of the human mental activities, which has been used for centuries. Thus it is not surprising that in scientific sphere, the application of clustering represents an oftenly used tool for separating scientific phenomena. The usage of clustering is reflected by the development of many statistical methods for clustering. Due to the fact, that there exists a broad variety of different clustering methods, we cannot overcome the problems, which emerge when choosing the most appropriate method for the given data. Namely, depending on the type and form of data structure, some methods outperform others.

Among clustering methods, we can distinct between hierarchical and non-hierarchical clustering methods, and the present paper focuses on the first ones. Among hierarchical clustering methods, we can further choose between divisive and agglomerative clustering methods, among which our study deals with the performance of the latter ones.

The primary aim of this paper is to examine the performance of three selected agglomerative hierarchical clustering methods, i.e. average, centroid and median method. To study the performance of these clustering methods, several data structures are simulated using statistical package R. Simulated data structures are applied to the selected hierarchical clustering methods, where the main purpose is to ascertain in which types of these data structures the selected clustering methods are successful in revealing and correctly classifying data points to the proper clusters. Because simulating data structures gives an advantage, since the cluster membership is known in advance, the comparison of the performance of selected clustering methods is actually possible.

The paper consists of two parts. In the first part, which consists of Sections two and three, we present the elementary theoretical background on cluster analysis, describe the selected agglomerative hierarchical clustering methods and outline a literature review on studies, analysing the performance of the hierarchical clustering methods. In the second part, in Section four, we delineate a process of simulating data structures with detailed description of each simulated data structure. In Section five, we present the results of the performance of the selected clustering methods. We also briefly outline the comparison between the selected and other agglomerative hierarchical clustering methods, in sense of their performance on simulated data structures. Section six concludes the study and proposes some directions for further research.

Cluster analysis

In the broadest sense, clustering signifies abstraction process in which a group of objects, for which we believe that they are in some way similar to each other, is appointed. The clustering algorithms have been developed using a wide range of conceptual models for studying all sorts of problems, where a common goal is the interest in grouping or segmenting a collection of objects into subsets or clusters. In this sense, several clustering algorithms and consequently different clustering techniques can be used. On the one hand, Everitt (1977) proposed the classification of cluster analysis techniques[4], and on the other hand, we can distinguish between several clustering methods[5]. The focus of the present paper is turned into studying the performance of agglomerative hierarchical clustering methods, for which we will observe the behaviour when several simulated data structures are applied (Ferligoj, 1989; Everitt, 1977).

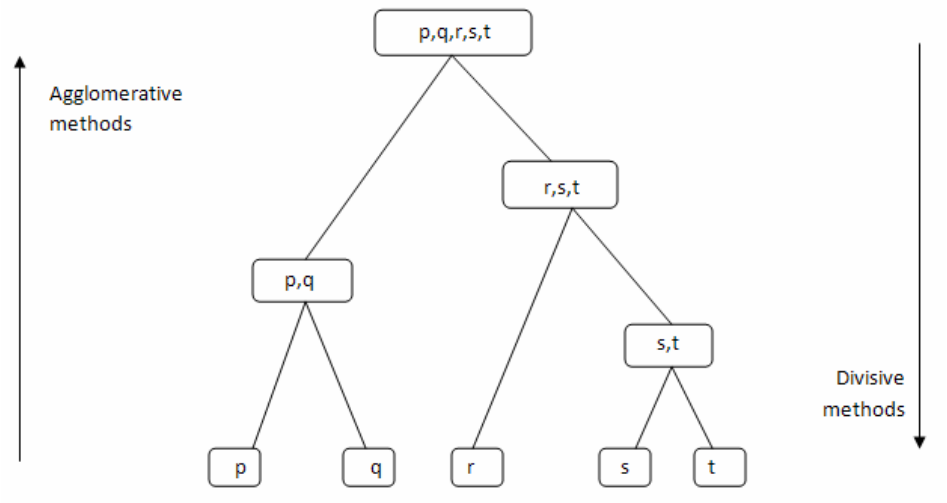
Agglomerative hierarchical clustering

The specificity of hierarchical methods is that they build a whole hierarchy of potential clusterings of the observations and can proceed in two ways. They can go bottom-up, i.e., start on the lowest hierarchy level and join clusters together towards the top; this approach is referred to as agglomerative clustering. The opposite approach is divisive clustering method that proceeds top-down, i.e., it starts at the top level with a single cluster and divides it towards the bottom level. The distinction between agglomerative and divisive hierarchical clustering methods is shown in Figure 1.

[4] Cluster analysis techniques can be classified roughly into five types: a) hierarchical techniques, b) optimization-partitioning techniques, c) density techniques, d) clumping techniques and e) others (Everitt, 1977).

[5] Clustering methods can be divided as a) sequential vs. simultaneous algorithms, b) agglomerative vs. divisive methods, c) monothetic vs. polythetic methods, d) hierarchical vs. non-hierarchical methods and e) probabilistic vs. non-probabilistic methods (Cluster Analysis, 2007).

Figure 1: Difference between agglomerative and divisive hierarchical clustering methods



Source: XLMiner (2008)

In our study, we focus on the analysis of different agglomerative hierarchical clustering methods' performance, which are in fact the most commonly used. The following algorithm presents the procedure of agglomerative hierarchical clustering:

Table 1: Basic algorithm for agglomerative hierarchical clustering

Step 1: Compute the proximity matrix, if necessary. **Step 2: repeat** **Step 3:** Merge the closest two clusters. **Step 4:** Update the proximity matrix to reflect the proximity between the new cluster and the original clusters. **Step 5: until** Only one cluster remains.

Source: Tan et al. (2006)

The aim of agglomerative clustering procedure is to merge objects with similar characteristics, where the purpose is to obtain internal homogeneity and external heterogeneity of the produced clusters. Here the main question is how to recognize similar units or how to decide on the criteria, according to which the objects are clustered. The answer is the application of the proper similarity or distance measure, which can be computed in different ways (Ferligoj, 1989).

Distance measures

The clustering criteria normally base on the proximity matrix whose elements represent the distances or similarities between the objects. Distance measures can be defined by various algorithms (e.g. Heeringa (2004) presenting Johnson's algorithm) and distances (e.g. Mahalonobis distance (Sharma, 1996) and others), but the most often used distance measures arise from the Minkowski measure, which is defined by

$$d_{ij} = \left(\sum_{k=1}^{p} |x_{ik} - y_{ik}|^r\right)^{\frac{1}{r}}, \qquad r > 0. \tag{2.1}$$

If $r = 1$, then we have Manhattan measure, and if $r = 2$, we get the Euclidean distance between two objects. In our study, to measure similarity between objects, the Euclidean distance is used. For the computation of the distance between cluster *k* and cluster (*ij*), which is formed by merging of clusters *i* and *j*, we can then use the following equation for Euclidean distance:

$$d_{ij} = \sqrt{\sum_{k=1}^{p} (x_{ik} - y_{jk})^2}. \tag{2.2}$$

Euclidean distance measures the spatial distance between two objects, so the two objects that are the closest are merged. Depending on the way of defining distance between the objects, differences between methods arise (Everitt, 1977).

According to Borcard (2007), agglomerative hierarchical clustering methods can be further divided to single-linkage (also known as Minimum distance method or Nearest neighbour method), complete-linkage (also known as Maximum distance method or Furthest neighbour), intermediate linkage[6] and average agglomerative clustering methods, among which the latter methods are the subject of the study presented in this paper. Generally, we can divide average agglomerative clustering methods as it is shown in *Table 2*.

[6] which includes all algorithms where group fusion occurs when a definite proportion of links is established between the members of the two clusters; this proportion is called connectedness and varies from 0 (single linkage) to 1 (complete linkage).

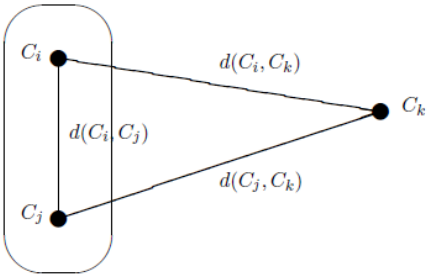
Table 2: The four methods of average agglomerative clustering (Borcard, 2007).

	Arithmetic average	**Centroid clustering**
Equal weights	*Unweighted arithmetic average clustering (UPGMA)*[7]	*Unweighted centroid clustering (UPGMC)*[8]
Unequal weights	Weighted arithmetic average clustering (WPGMA)[9]	*Weighted centroid clustering (WPGMC)*[10]

In this paper, we primarily concern the performance of average (UPGMA), centroid (UPGMC) and median (WPGMC) method. Furthermore, we compare the performance thereof to other, mostly used agglomerative hierarchical clustering methods.

The selected methods differ in the way of computing dissimilarity measure *d* between a new cluster and other remaining clusters in the clustering process. Let us consider, that in a specific step of the clustering procedure we have three clusters, as presented in *Figure 2.*

Figure 2: Three clusters C_i, C_j and C_k



Source: Ferligoj (1989)

We denote these clusters by C_i, C_j and C_k. In the clustering procedure the clusters which are the nearest (in this case C_i and C_j) are merged, and the dissimilarity measure between the new and the remaining cluster can be computed, depending on which method is chosen, in the following ways:

[7] also known as average method
[8] also known as centroid method
[9] also known as McQuitty method
[10] also known as median or Gower method

Average method (Sokal and Michener in Ferligoj, 1989):

$$d(C_i \cup C_j, C_k) = \frac{1}{(n_i+n_j)n_k} \sum_{U \in C_i \cup C_j} \sum_{V \in C_k} d(U,V), \quad (2.3)$$

where n_i denotes the number of data points in cluster C_i. This means, that in average method, the distance between two clusters is defined as the average of the distances between all pairs of objects in the two clusters.

Centroid method (Jesenko and Jesenko, 2007):

$$d(C_i \cup C_j, C_k) = d(T_{ij}, T_k), \quad (2.4)$$

where T_{ij} denotes the center of the merged cluster $C_i \cup C_j$ and T_k the center of cluster C_k. Here, the distance between clusters is defined as the distance between their centroids[11]. The procedure is to merge clusters according to the calculated distance between their centroids, where the clusters with the smallest distance are merged first.

Median method (Gower in Ferligoj, 1989)

$$d(C_i \cup C_j, C_k) = d^2(T_{ij}, T_k), \quad (2.5)$$

where T_{ij} denotes the centre of the merged cluster $C_i \cup C_j$ and T_k the centre of cluster C_k. The median method is actually very similar to the centroid method; the only difference is in the weighting, which is introduced into the computations, so the differences in cluster sizes are considered.

Most of the hierarchical clustering methods can also be presented as special cases of Lance and Williams' general agglomerative algorithm. The distance between cluster *k* and cluster (*ij*), which is formed by merging of clusters *i* and *j*, is defined by:

$$d_{k(ij)} = \alpha_i d_{ki} + \alpha_j d_{kj} + \beta d_{ij} + \gamma |d_{ki} - d_{kj}|, \quad (2.6)$$

where d_{ij} represents the distance between clusters *i* and *j*. The appropriate choice of coefficients α, β and γ defines the agglomerative

[11] Centroid can be considered also as the center of gravity of the particular cluster (StatSoft, 2008).

hierarchical method, which is going to be used. *Table 3* summarizes the clustering criteria in terms of their parameter values for average, centroid and median methods (For the clustering criteria for other agglomerative hierarchical clustering methods see Everitt (1977:17).

Table 3: Clustering criteria for average, centroid and median method for general agglomerative algorithm

Name	α_i	α_j	**β**	**γ**
Average method	$\frac{n_i}{(n_i + n_j)}$	$\frac{n_j}{(n_i + n_j)}$	0	0
Centroid method	$\frac{n_i}{(n_i + n_j)}$	$\frac{n_j}{(n_i + n_j)}$	$-\alpha_i\alpha_j$	0
Median method	$\frac{1}{2}$	$\frac{1}{2}$	$-\frac{1}{4}$	0

Source: Everitt (1977)

(Dis)advantages of the selected hierarchical clustering methods

Average, centroid and median methods all base on computation of average similarities among objects or on centroids of clusters, but still there is some differences in their performance.

In this manner, the main distinction among selected methods is, that centroid and median methods assume that the objects of clusters are represented by data points in some Euclidean space. In this sense, clusters are replaced on formation by the co-ordinates of their centroid. This means that the quantity $d(C_i, C_j)$ is defined as the distance between centroids of clusters C_i and C_j in the centroid method, and as the distance between weighted centroids of clusters C_i and C_j in the median method. In contrast, the average method does not involve this kind of assumption (Gordon, 1987; Everitt, 1977).

One of the desired characteristics of hierarchical clustering methods is their monotonicity. In the case of monotonicity, the similarity measure must decrease among the entire clustering process. The occurrence of non-monotonicity, or some might say reversals, indicates that the fusion of clusters in the latter step was carried out on the lower level of dissimilarity as in the previous steps. Among the selected methods, the centroid and median methods have the characteristic of being nonmonotonous, so they might produce reversals which are the most evident when drawing the dendrograms.

In relation to this problem, Batagelj (1981) proposed the theorem, in which he proves that the hierarchical clustering method, arising from Lance and Williams' algorithm presented in *Table 2*, can assure monotonous dendrograms when the following conditions are satisfied:

$$\gamma \geq -\min(\alpha_1 + \alpha_2)$$
$$\alpha_1 + \alpha_2 \geq 0 \qquad (2.7)$$
$$\alpha_1 + \alpha_2 + \beta \geq 1$$

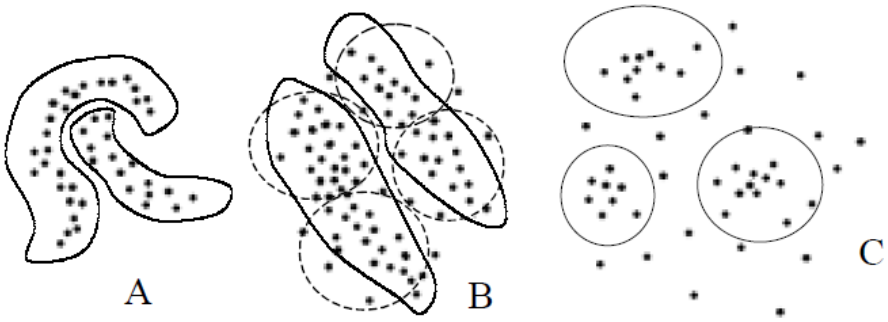
The first and the second conditions are satisfied by all selected hierarchical methods, whereas the third condition in case of centroid and median methods is not satisfied, which means that only average method assures the monotonicity.

Literature review

In the literature, it is often possible to gather facts about some hierarchical methods which are successful in finding clusters, consisting of relative dense data points, surrounded by an empty or relative rarely dotted space. For such clusters, Everitt (1974) proposed a term natural clusters, and Cormack (1971) and Gordon (1981) defined two desired characteristics of natural clusters, i.e. internal cohesion and external isolation. In literature single-linkage, complete-linkage and Ward's method are the most commonly mentioned and studied. For the remaining hierarchical clustering methods, we can notice a lack of studies, which would give applicable information about their performance.

To analyse the performance of the methods mentioned in the literature more precisely, we enclose the following figure, which helps understand some of the differences between various hierarchical methods.

Figure 3: Three possible forms of data structure



Source: CIAT (2008)

The diagram in *Figure 3* shows three possible forms of data structure. Let us first consider structure A in the diagram above. In this case, only single-linkage clustering method finds two separated clusters as they are denoted in the structure. In the structure B, two elongated elliptical clusters are presented, which are denoted by the hand-drawn solid curves. It is interesting that none of the methods can identify these two clusters as they are presented, however it is proved that Ward's and centroid methods will probably determine the four clusters denoted by dashed circles. Since the single-linkage and average clustering methods are more likely to find elongated clusters, the data should be rescaled to tighten up the clusters to assure that the methods would work well. Finally, in structure C, the diagram presents the data points, which are dispersed in such way, that they form dense spherical clusters in a background of random data points. Such clusters are best found by Ward's method (CIAT, 2008).

To sum up, it is proven that single-linkage method is very effective in finding elongated »sausage« shaped even non-elliptic structures. On the other hand, single-linkage method is inapplicable to non-explicitly separated clusters. Furthermore, the complete-linkage method presents a very effective in finding spherical clusters and Ward's method in finding ellipsoid-shaped clusters. Average clustering methods perform very well in finding compact clusters, and are in some circumstances very effective in tracing clusters with unusual shapes (StatSoft, 2008; Ferligoj, 1989).
The review of the performance of different agglomerative hierarchical clustering methods gave us some general information about which hierarchical method is useful for correctly classifying which data structures. What can we say about other data structures? For this

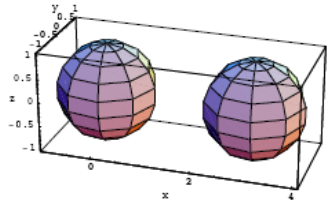
reason, we simulate some interesting data structures to supplement some facts to the already known ones.

Data simulation

To study the performance of average, centroid and median agglomerative hierarchical clustering methods, several data structures are simulated. The first step is to select the types of the three-dimensional data structures, to which authors have not yet paid much attention in the previous studies. The second step is the actual simulation of data using R package for statistical computing (http://www.R-project.org/). Hence, we simulate six types of three-dimensional data structure: a) spherical clusters, b) ellipsoid clusters, c) umbrella-like, d) core-and-sphere, e) ring-like and f) intertwined data structure, which we present in *Figure 4*. Each data structure consists of 200 data points, which are partitioned into two separate clusters. The first 100 data points define one cluster, and the following 100 data points define another cluster.

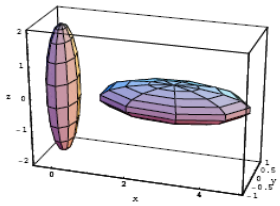
Figure 4: Six types of three-dimensional simulated data structures: a) spherical clusters, b) ellipsoid clusters, c) umbrella-like, d) core-and-crust, e) ring-like and f) intertwined data structures.

SPHERICAL CLUSTERS DATA STRUCTURE



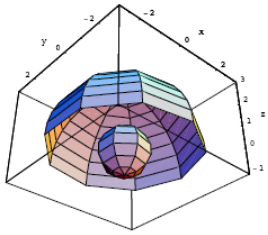
a)

ELLIPSOID CLUSTERS DATA STRUCTURE



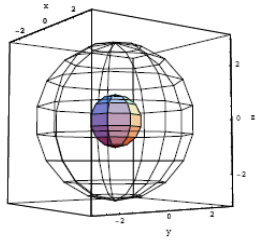
b)

UMBRELLA-LIKE DATA STRUCTURE



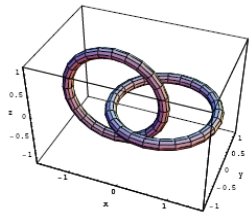
c)

CORE-AND-SPHERE DATA STRUCTURE



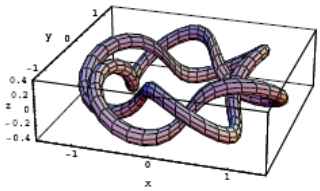
d)

RING-LIKE DATA STRUCTURE



e)

INTERTWINED DATA STRUCTURE



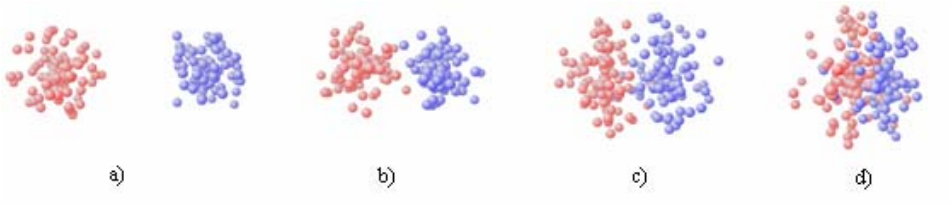
f)

For each data structure different positions of clusters and/or several degrees of dispersion of data points in clusters are applied, thus we get four different situations for each data structure.

Spherical clusters data structure

Spherical clusters data structure is presented by two separated spheres. The points in each sphere are uniformly distributed over the whole skeleton and the radius of each sphere is 1 unit. We observe four different situations by changing the position of the clusters from non-overlapping to overlapping clusters as the distance between the hubs of the clusters is reduced. In the first position, the distance between the hubs is 2 units, then it is reduced to 1.5 units, in the third position to 1 unit and finally, the distance between the hubs of the clusters is only 0.5 units. We present the described situation in *Figure 5*.

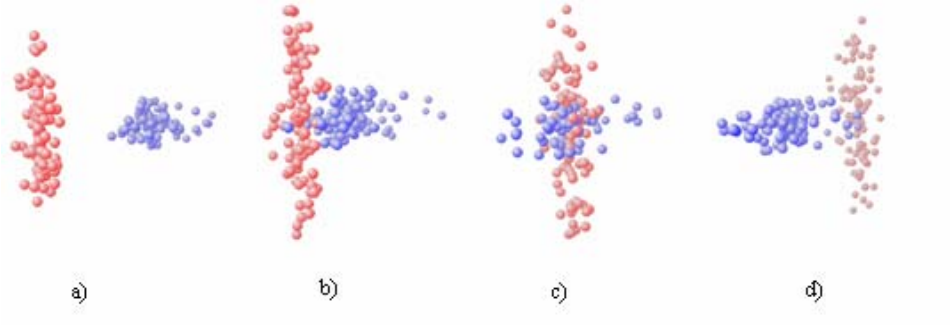
Figure 5: Four positions of the two clusters (represented by two different colours, red and blue) in the spherical clusters data structure. The positions differ in terms of the distance between the two clusters: a) distance = 2 units, b) distance = 1.5 units, c) distance = 1 unit, and d) distance = 0.5 units.



Ellipsoid clusters data structure

Ellipsoid clusters data structure consists of two separated ellipsoids, one situated horizontally and one vertically. The points in each ellipsoid are uniformly distributed over the whole skeleton. The ellipsoids are defined by the axes a=0.4, b=0.7, c=2.0, and a=2.0, b=1.0, c=0.5, respectively. Again, we observe different situations from non-overlapping to overlapping clusters as the second ellipsoid is moved from point (0,0) of the coordinate system in x-axis direction. At first, the position of the second ellipsoid is defined by 3 units in x-axis direction, then the position of the second ellipsoid is changed as the ellipsoid is placed to 1 unit, then to 0 units in and finally to -2 units, all in x-axis direction. The situation is presented in *Figure 6*, where the position of the vertical (blue) ellipsoid is being changed.

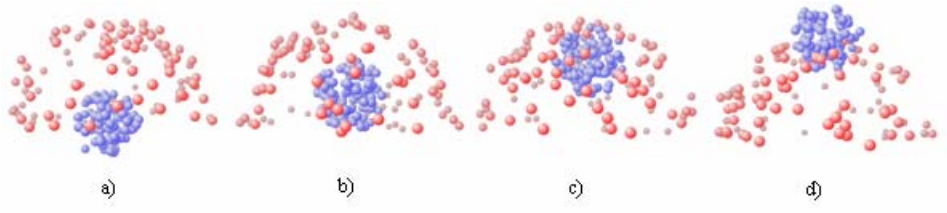
Figure 6: Four positions of the two clusters (represented by two different colours, red and blue) in the ellipsoid clusters data structure. The positions differ in terms of the movement of the blue cluster: a) movement = 3 units, b) movement = 1 unit, c) movement = 0 units, d) movement = -2 units.



Umbrella-like data structure

Umbrella-like data structure is combined from a semi-sphere with radius 3 and a core with radius 1. The points in the semi-sphere are uniformly distributed over the solid angle and normally distributed over the radius of the semi-sphere. The points in the core are uniformly distributed over the entire volume of the core. In the case of umbrella-like data structure, the position of the core cluster is changed. At first, the core cluster is in the centre of the virtual total sphere, then the core is moved for 1 unit, then for 2 units and finally for 3 units, all in z-axis direction. In the latter case the core is situated on the verge of the semi-sphere. The overall situation is presented in *Figure 7*.

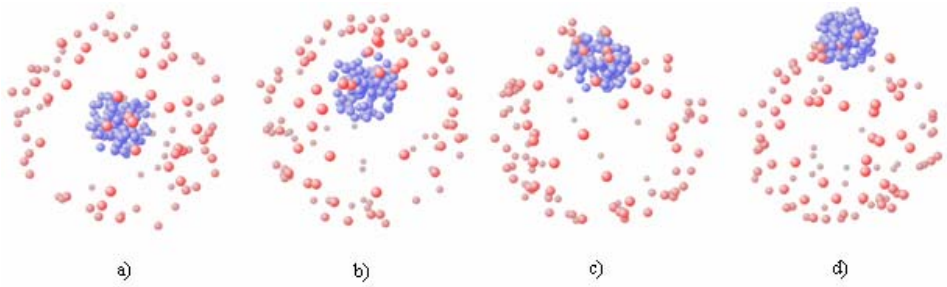
Figure 7: Four positions of the two clusters (represented by two different colours, red and blue) in the umbrella-like data structure. The positions differ in terms of the movement of the core cluster: a) movement = 0 units, b) movement = 1 unit, c) movement = 2 units, d) movement = 3 units.



Core-and-sphere data structure

Core-and-sphere data structure consists of a sphere with radius 3 and a core with radius 1. The points in the sphere are uniformly distributed over the solid angle and normally distributed over the radius of the sphere. The points in the core are uniformly distributed over the entire volume of the core. Like in the precedent case, the position of the core cluster is changed. At first, the core cluster is in the centre of the sphere, then the core is moved for 1 unit, for 2 units and finally for 3 units in z-axis direction, placing the core on the verge of the sphere. The situation is presented in *Figure 8*.

Figure 8: Four positions of the two clusters (represented by two different colours, red and blue) in the core-and-sphere data structure. The positions differ in terms of the movement of the core cluster: a) movement = 0 units, b) movement = 1 unit, c) movement = 2 units, d) movement = 3 units.

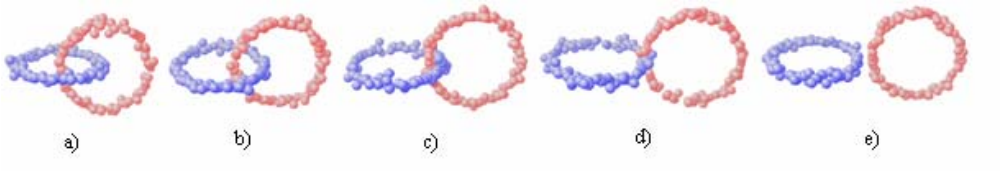


Ring-like data structure

Ring-like data structure consists of two rings (chain-shaped), where the rings are rectangular to each other. We observe the separation of the

two rings by moving the first ring for the certain distance in x-axis direction and the second ring for reverse value of certain distance in x-axis direction. The dispersion of the points is constant. The rings are moved apart step by step for 1/6 of the unit. Initially, the distance between the rings is 1/2 units, and is increased to 2/3 units, 5/6 units, 1 unit (the rings are touching), and finally to 7/6 units, when the rings are entirely separated. The situation is summed up in *Figure 9*.

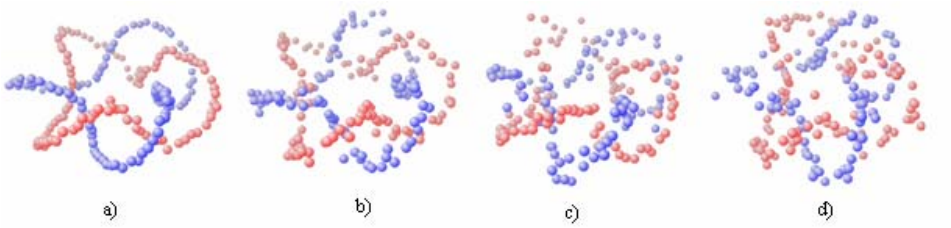
Figure 9: Five positions of the two clusters (represented by two different colours, red and blue) in the ring-like data structure. The positions differ in terms of the distance between the two clusters: a) distance = 1/2 units, b) distance = 2/3 units, c) distance = 5/6 units, d) distance = 1 unit, e) distance = 7/6 units.



Intertwined data structure

Intertwined data structure is made of two curves with three turns, where the second curve is phase shifted for exactly half of the turn in comparison to the first one. On these two curves, the points are situated and the deviation of the points from each initial curve is normally distributed. We get the intertwined, double-helix-like data structure, which reminds on the DNA double helix. In this case, we observe the change of dispersion of the points as the degree of dispersion of data points is changed. At first, the points are almost nondispersed, the standard deviation is 0.05. In the next three situations, the dispersion is increased, at first the dispersion is determined by sd=0.1, then by sd=0.15 and finally by sd=0.2. The situation is presented in *Figure 10*.

Figure 10: Four positions of the two clusters (represented by two different colours, red and blue) in the intertwined data structure. The positions differ in terms of the degree of variability of data points: a) sd=0.05 b) sd=0.1, c) sd=0.15, d) sd=0.2



Results

As we have shown, in each simulated data structure the data points are arranged in such a way, that a half of them compose one cluster and a half of them another cluster. The data points' memberships to the particular cluster are defined in the initial stage of simulating data. Taken as a whole, the first 100 data points define one cluster and the following 100 data points define another cluster. In this respect, we use the cluster analysis for assessment of correctly classified data points in each predefined cluster. In the first subsection, we have presented the analysis of the performance of the average, centroid and median methods, and in the second subsection, we have outlined the comparison of the selected methods to other agglomerative hierarchical clustering methods.

Performance of the average, centroid and median methods

To perform the experiments, we use the R (http://www.R-project.org/) implementation of the hierarchical agglomerative clustering algorithm (function *hclust* in R), which allows us to select among the three methods of average, centroid and median. As the distance measure between individual data points, the Euclidean distance is used. Due to the purpose of presented study, when performing the methods on simulated data structures, we have chosen to use a classification into two groups. The estimation of the performance of each selected clustering method for each data structure is presented numerically by the percentage of correctly classified data points and visually by dendrograms in the Appendix. On one hand, we have exposed the tables with the information about percentage of correctly classified data points, where the highest value for each data structure and each position is marked bold. Here, if the percentage is 100.0, then all data points are classified correctly. If the percentage is close to 50.0, then the method is unsuccessful in classifying data points as almost a half of data points are wrongly classified. Because of the symmetry, the percentage of correctly classified data points cannot attain the value lower than 50.0. We have witnessed the situation, in which the algorithm of the selected hierarchical method has indicated data points from the predefined first

cluster as data points constituting the second cluster and the other way around. In this way, it we should assume that the error in classifying data points is almost 100%. But we have been confronted with a perfect classification, where only order of precedence of clusters is inverted. On the other hand, dendrograms in the Appendix show to which cluster each individual branch belongs, but the membership is also visible from the coloured points (black and red), which are situated under the dendrograms. In the ideal case, the coloured points should not be mixed, which would mean that the method was successful in locating the data points to the proper predefined cluster. Let us now consider the results from the simplest to the more complicated data structures.

Table 4: Percentage of correctly classified data points for the four positions of the clusters in the spherical clusters data structure.

	METHOD								
	Average method			**Centroid method**			**Median method**		
	1st sphere	**2nd sphere**	**Total**	**1st sphere**	**2nd sphere**	**Total**	**1st sphere**	**2nd sphere**	**Total**
(a)	100,0	100,0	**100,**	100,0	100,0	**100,0**	100,0	100,0	**100,0**
(b)	99,0	100,0	**99,5**	99,0	100,0	**99,5**	64,0	100,0	82,0
(c)	96,0	92,0	**94,0**	100,0	1,0	50,5	100,0	1,0	50,5
(d)	100,0	4,0	52,0	100,0	2,0	51,0	85,0	24,0	**54,5**

The results for spherical clusters data structure show that in the first position, where the distance between the spheres is relatively great, each method classifies data points into two separate clusters, so the performance of each studied method is successful. Differences between the clustering methods become obvious when the position of the spheres is being changed. In the second position, where the overlapping of the data points is yet not very explicit, the error in classifying data points is still not very large. The average and centroid methods classify wrongly only one data point, whereas the median method correctly classifies only 82% of the data points. The more the data points are overlapped, the less successful the methods are in recognizing two separate clusters. Overall, we can conclude, that the average method is the most effective in finding two separate clusters in spherical clusters data structure, although in the last position all methods do not perform well. The conception and findings are obvious also from the presented in dendrograms in *Figure 11* in the Appendix.

Table 5: Percentage of correctly classified data points for the four positions of the clusters in the ellipsoid clusters data structure.

	METHOD								
	Average method			**Centroid method**			**Median method**		
	1st ellipsoid	**2nd ellipsoid**	**Total**	**1st ellipsoid**	**2nd ellipsoid**	**Total**	**1st ellipsoid**	**2nd ellipsoid**	**Total**
(a)	100,0	100,0	**100,0**	100,0	100,0	**100,0**	100,0	100,0	**100,0**

(b)	100,0	13,0	**56,5**	100,0	7,0	53,5	100,0	3,0	51,5
(c)	100,0	16,0	58,0	100,0	17,0	**58,5**	100,0	2,0	51,0
(d)	100,0	97,0	**98,5**	100,0	97,0	**98,5**	100,0	1,0	50,5

In the case of ellipsoid clusters data structure, the results show that in the first position, where the ellipsoids are positioned relatively far away from each other, all three methods are capable of finding two different clusters of data points. When the data points are beginning to overlap, the successfulness of the methods is reduced, but in the case of the fourth position, where the overlapping of the data points is not very explicit any more, the average and centroid methods perform well again. The median method is not very accurate method for revealing this type of clusters. The dendrograms are available in *Figure 12* in the Appendix, which represent the described conception and ascertainment for the ellipsoid clusters data structure.

Table 6: Percentage of correctly classified data points for the four positions of the clusters in the umbrella-like data structure.

	METHOD								
	Average method			**Centroid method**			**Median method**		
	Core	**Semi-sphere**	**Total**	**Core**	**Semi-sphere**	**Total**	**Core**	**Semi-sphere**	**Total**
(a)	100,0	26,0	**63,0**	100,0	2,0	51,0	100,0	3,0	51,5
(b)	100,0	26,0	**63,0**	100,0	2,0	51,0	100,0	19,0	59,5
(c)	100,0	22,0	**61,0**	100,0	20,0	60,0	100,0	3,0	51,5
(d)	100,0	39,0	**69,5**	100,0	3,0	51,5	100,0	17,0	58,5

The results for umbrella-like data structure show that all three methods are successful in finding the core cluster of the applied data structure, which confirms that these methods are able to reveal the clusters with internal cohesion, since this core cluster is an example of such a cluster. If we focus on the data points, composing the semi-sphere cluster, we are confronted with the entirely different situation. The data points of the semi-sphere cluster are dispersed around the core cluster where the distance between semi-sphere data points is often larger than the distance between the core cluster data points and the semi-sphere cluster data points. Due to this fact, all three methods are experiencing difficulties in finding the semi-sphere cluster in all four positions of the data structure. As it is shown in *Table 6* and in *Figure 13* in the Appendix, among the three selected clustering methods the average method is the most successful in finding two distinct clusters, although the percentage of correctly classified data points is only a good 60%.

Table 7: Percentage of correctly classified data points for the four positions of the clusters in the core-and-sphere data structure.

	METHOD								
	Average method			**Centroid method**			**Median method**		
	Core	**Sphere**	**Total**	**Core**	**Sphere**	**Total**	**Core**	**Sphere**	**Total**
(a)	100,0	32,0	**66,0**	100,0	2,0	51,0	100,0	16,0	58,0
(b)	100,0	21,0	**60,5**	100,0	11,0	55,5	100,0	8,0	54,0
(c)	100,0	20,0	60,0	100,0	28,0	64,0	100,0	55,0	**77,5**
(d)	100,0	76,0	**88,0**	100,0	2,0	51,0	100,0	13,0	56,5

In the core-and-sphere data structure we are confronted with very similar situation as in the previous case. All three methods disclose the core cluster successfully, but they have problems in identifying the data points of the sphere as a separated cluster. The data points of the sphere are more likely attached to the core cluster than to the sphere cluster. In spite of all that, the average method is proved to outperform others, which is also obvious from the dendrograms in *Figure 14* in the Appendix.

Table 8: Percentage of correctly classified data points for the five positions of the clusters in the ring-like data structure.

	METHOD								
	Average method			**Centroid method**			**Median method**		
	1. ring	**2. ring**	**Total**	**1. ring**	**2. ring**	**Total**	**1. ring**	**2. ring**	**Total**
(a)	54,0	100,0	77,0	100,0	55,0	**77,5**	100,0	50,0	75,0
(b)	53,0	100,0	76,5	100,0	69,0	84,5	100,0	77,0	**88,5**
(c)	54,0	100,0	77,0	100,0	57,0	78,5	100,0	79,0	**89,5**
(d)	66,0	100,0	83,0	71,0	100,0	85,5	100,0	74,0	**87,0**
(e)	81,0	100,0	90,5	100,0	60,0	80,0	100,0	100,0	**100,0**

The results for ring-like data structure show that the average method successfully identifies the second ring, whereas the data points from the first ring, especially the ones that are overlapping with the data points constituting the second ring, are to a great extent attached to the second ring. Centroid and median methods perform exactly opposite, as they successfully find the first ring. In the case of ring-like data structure, the median method turns out to outperform others. Again, dendrograms can be found in *Figure 15* in the Appendix.

Table 9: Percentage of correctly classified data points for the four positions of the clusters in the intertwined data structure.

	METHOD								
	Average method			**Centroid method**			**Median method**		
	1st curve	**2nd curve**	**Total**	**1st curve**	**2nd curve**	**Total**	**1st curve**	**2nd curve**	**Total**
(a)	52,0	52,0	52,0	8,0	100,0	54,0	100,0	13,0	**56,5**

(b)	62,0	40,0	51,0	38,0	63,0	50,5	5,0	100,0	**52,5**
(c)	55,0	49,0	**52,0**	100,0	2,0	51,0	89,0	13,0	51,0
(d)	43,0	58,0	50,5	69,0	36,0	52,5	42,0	73,0	**57,5**

The results based on the most complicated data structure, intertwined data structure, which reminds us on the DNA double-helix, show the very poor performance of all three selected hierarchical clustering methods. It is interesting, that although the dispersion of the data points is being increased from one position to another, the results of the average method are not changing considerably. Because of the shape of the applied data structure, it is a big surprise that the centroid and median method, both in two cases (first and third position for centroid method and first and second position for median method), identify one of the clusters correctly. Nevertheless, the total percentage of correctly classified data points is still low, because of the poor performance of the methods in the case of another cluster. Overall, although the percentage of the correctly classified data points is just a bit over 50%, the median method is performing the best again, but still not very successfully. Dendrograms for intertwined data structure are available in *Figure 16* in the Appendix.

Comparison to other agglomerative hierarchical clustering methods

To enable the comparison of the methods, we used the R (http://www.R-project.org/) implementation of the hierarchical agglomerative clustering algorithm (function *hclust* in R) also for all other agglomerative hierarchical clustering methods (single-linkage, complete-linkage, Ward and McQuitty methods). To ensure the proper comparison, as the distance measure between individual data points, the Euclidean distance was used. A brief overview of the results is presented in the following *Table 10*.

Table 10: Percentage of correctly classified data points for average, centroid, median, Mcquitty, Ward, single-linkage and complete-linkage hierarchical clustering methods for simulated data structures

		METHOD						
		Average	**Centroid**	**Median**	**Mcquitty**	**Ward**	**Single-linkage**	**Complete-linkage**
Spherical clusters data structure	(a)	**100,0**	**100,0**	**100,0**	**100,0**	**100,0**	**100,0**	**100,0**
	(b)	**99,5**	**99,5**	82,0	**99,5**	**99,5**	51,5	**99,5**
	(c)	94,0	50,5	50,5	87,0	**94,5**	50,5	86,0
	(d)	52,0	51,0	54,5	53,0	58,0	50,5	**58,5**

Ellipsoid data structure	(a)	**100,0**	**100,0**	**100,0**	**100,0**	**100,0**	**100,0**	**100,0**
	(b)	56,5	53,5	51,5	57,0	**97,5**	51,5	56,0
	(c)	58,0	58,5	51,0	**61,5**	55,5	52,0	60,5
	(d)	**98,5**	**98,5**	50,5	**98,5**	**98,5**	**98,5**	89,5
Umbrella-like data structure	(a)	63,0	51,0	51,5	62,0	77,0	**100,0**	72,0
	(b)	63,0	51,0	59,5	69,5	78,5	**100,0**	66,5
	(c)	61,0	60,0	51,5	**74,0**	70,0	51,0	59,5
	(d)	69,5	51,5	58,5	**74,0**	69,5	50,5	72,5
»Core-and-sphere« data structure	(a)	66,0	51,0	58,0	62,5	79,5	**100,0**	81,5
	(b)	60,5	55,5	54,0	64,5	73,5	50,5	**77,0**
	(c)	60,0	64,0	77,5	64,0	**90,0**	50,5	66,0
	(d)	**88,0**	51,0	56,5	66,5	85,0	51,0	72,0
Ring-like data structure	(a)	77,0	77,5	75,0	75,0	77,5	**100,0**	82,5
	(b)	76,5	84,5	88,5	87,0	80,0	**100,0**	76,5
	(c)	77,0	78,5	**89,5**	88,5	82,0	50,5	88,5
	(d)	83,0	85,5	**87,0**	80,5	84,0	50,5	86,5
	(e)	90,5	80,0	**100,0**	90,0	89,5	**100,0**	90,0
Intertwined data structure	(a)	52,0	54,0	56,5	52,5	51,5	**100,0**	57,0
	(b)	51,0	50,5	52,5	54,5	50,5	**100,0**	51,5
	(c)	52,0	51,0	51,0	**55,5**	52,5	51,0	**55,5**
	(d)	50,5	52,5	**57,5**	54,5	52,5	50,5	56,5

The results, presented in *Table 10* show that in the case of convex overlapping clusters, which are represented by spherical and ellipsoid clusters data structure, all methods correctly reveal clusters if they are well separated. Differences between the performance of the methods become obvious, as the position of clusters is changed. In comparison to the selected methods, for which the results of their performance were outlined in the previous section, other agglomerative clustering methods perform very similar. As long as the overlapping of the data points is yet not very explicit, the McQuitty, Ward and complete-linkage methods still perform well, the exception is only the single-linkage method. This

signifies the fact that the single-linkage method is very sensitive to chaining, which means that the single-linkage method can falsely link data points from one cluster to another. Overall, for convex (spherical and ellipsoid) structures it is obvious that the Ward's method outperforms the others and that confirms the fact that the Ward's method perform well in finding ellipsoid-shaped clusters.

For examples of spherical cluster with core cluster, represented by umbrella-like and core-and-sphere data structures, it is shown that all methods, except the single-linkage, have problems in finding a correct structure in a given data. In the previous section it was shown that the selected methods are able to find the core cluster, but incorrectly join data points from a surrounding spherical or semi-spherical cluster. The same situation is present in case of other agglomerative clustering methods, to which the selected methods are compared. The Mcquitty, Ward, single-linkage and complete-linkage methods all correctly classify the data points into the core cluster, whereas the points composing the semi-sphere or the sphere cluster are more likely to be attached to the core cluster. Exception is the single-linkage method, which is the only method that can reveal two clusters, of course if they are enough separated. With this fact we confirmed the proof that the single-linkage methods is indeed very effective in finding non-elliptic structures.

In the case of chaining structure, represented by ring-like and intertwined data structures, we can conclude, that a majority of the methods have problems in finding two separate clusters. The performance of the selected methods, presented in the previous section, and the performance of other agglomerative clustering methods are very similar. Only the single-linkage method clearly outperforms other methods, although in case of the ring-like data structure the median method also performs well. From the results it is thus obvious that the single-linkage method really performs well in finding elongated "sausage" shaped structures, although we can ascertain, as the dispersion and the overlapping of data points increases, even the single-linkage method cannot classify data points correctly.

Conclusion

Cluster analysis includes various methods and they all can be categorised in two main classes of hierarchical and non-hierarchical methods. This paper considered agglomerative hierarchical clustering methods, and among them the performance of average, centroid, and median, agglomerative methods was analysed. These methods were chosen as representatives of the wide range of average agglomerative hierarchical clustering methods. To execute the study, several data

structures were simulated to test the performance of selected clustering methods. We choose to simulate spherical clusters, ellipsoid clusters, umbrella-like, core-and-sphere, ring-like and intertwined data structures using statistical software R to test the performance of the selected clustering methods. For each data structure different positions of clusters and/or several degrees of dispersion of data points in clusters were applied. The performance of the methods was tested by R implementation of the hierarchical agglomerative clustering algorithm.

The results showed some interesting facts about the performance of each method for each simulated data structure. All methods performed well on data structures with explicitly separated clusters of data points. However, the change of the position of clusters and/or degree of dispersion of data points and the overlapping of data points caused some problems to the performance of the methods. We found out that in the case of compact structures, presented by spherical clusters and ellipsoid data structures, the best performance in finding two separate clusters was exposed by the average method, although also a centroid method was quite good in classifying data points to the two clusters. When we applied compact structure with core cluster, presented by umbrella-like and core-and-sphere data structures, again the average method was proved to be performing the best. In case of elongated structures, applied by ring-like and intertwined data structures, the average method did not perform the best any more, and the median method was proved to perform best instead. Nevertheless, in case of ring-like data structure the results of all three clustering methods were very comparable. We can conclude that average and centroid methods are successful when applying compact data structures, for compact and dispersed data structures the average method is the most appropriate and for elongated data structures the median method outperforms other methods, although for intertwined data structure the results also in case of median method are not very satisfactory. In all cases, we can ascertain that when the data points of the both clusters begin to overlap, the performance of the methods begin to weaken.

On the other hand, the comparison to other hierarchical clustering methods was made. We found out, that all seven methods were successful in finding compact, bell-shaped or ellipsoid structure. In case of convex overlapping clusters, Ward's method outperforms the others, although when the clusters were well separated, all methods correctly classified data points into two clusters. In case of spherical clusters with core cluster, the single-linkage method was the only one that revealed a correct structure, but only in case of very separated clusters. In case of

two chaining structures, again the single-linkage method clearly outperformed others.

This paper presented only a drop in the sea of all possible studies of hierarchical clustering methods and applied simulated data structures. Other hierarchical clustering methods could be used and different data structures could be simulated. Due to the fact, that for each data structure data points were fixed by one simulation, we could also use different varieties of the same data structures, defined by hundreds of simulations. As we can see, several of ways can be applied to do the study differently or even widely. Because of the curiosity and interest in hierarchical clustering, we hope to continue the study from the concluding point of this paper.

Acknowledgements

The authors would like to thank dr. Anuška Ferligoj for her help, guidance and support in preparing and accomplishing the presented study. We would also like to thank doc. dr. Ljupčo Todorovski for many useful comments and suggestions during the writing of this paper. We are also grateful to all of the co-students of statistics, who helped us to simulate and analyze the data: Tina Ostrež, Emil Polajnar, Kristijan Breznik, Rok Blagus, Sanja Filipič, Branka Golob and Mojca Čižek-Sajko.

Resources

Batagelj, Vladimir (1981): Note on ultrametric hierarchical clustering algorithms. Psychometrika, Vol.: 46, No.: 3, pp.: 351-352.

Borcard, Daniel (2007): Multivariate analysis. Available at: http://ubio.bioinfo.cnio.es/Cursos/CEU_MDA07_practicals/Further%20reading/Multivariate%20analysis%20Borcard%202006/Chap_3.pdf (16.6.2008)

CIAT: Floramap™: Theory. Available at: http://www.floramap-ciat.org/download/theory.pdf (16.6.2008)

Cluster Analysis: Lectures. Available at: http://www.plantbio.ohiou.edu/epb/instruct/multivariate/Week7Lectures.PDF (18.6.2008)

Everitt, Brian S. (1977): Cluster Analysis. London: Heinemann Educational Books Ltd.

Ferligoj, Anuška (1989): Razvrščanje v skupine. Teorija in uporaba v družboslovju. Metodološki zvezki, Vol.:4, Ljubljana.

Francetič, Matej, Mateja Nagode and Bojan Nastav (2005): Hierarchical Clustering with Concave Data Sets. Metodološki zvezki, Vol.: 2, No.: 2, pp.: 173-193.

Gordon, A.D. (1987): A Review of Hierarchical Classification. Journal of the Royal Statistical Society, Vol.: 150, No.: 2, pp.: 119-137.

Jesenko Jože and Manca Jesenko (2007): Multivariatne statistične metode. Kranj: Moderna organizacija.

Heeringa, Wilbert (2004): Measuring Dialect Pronunciation Differences using Levenshtein Distance. Available at: http://www.let.rug.nl/~heeringa/dialectology/thesis/thesis06.pdf (18.6.2008)

R Development Core Team, R: A language and environment for statistical computing, R Foundation for Statistical Computing, Vienna, Austria, ISBN3-900051-07-0. Available at: http://www.R-project.org/

Sharma, Subhash (1996): Applied Multivariate Techniques. New York: John Wiley&Sons, Inc.

StatSoft (2008): Cluster Analysis. Available at: http://www.statsoft.com/textbook/stcluan.html (21.6.2008)

Tan, Pang-Ning, Michael Steinbach and Vipin Kumar (2006): Introduction to Data Mining. Chapter 8: Cluster Analysis: Basic Concepts and Algorithms. Available at: http://www-users.cs.umn.edu/~kumar/dmbook/ch8.pdf (16.6.2008)

XLMiner (2008): Hierarchical Clustering. Available at: http://www.resample.com/xlminer/help/HClst/HClst_intro.htm (18.6.2008)