Graph construction with condition-based weights for spectral clustering of hierarchical datasets

Dávid Papp[1], Zsolt Knoll[2], and Gábor Szűcs[3]

Abstract—**Most of the unsupervised machine learning algorithms focus on clustering the data based on similarity metrics, while ignoring other attributes, or perhaps other type of connections between the data points. In case of hierarchical datasets, groups of points (point-sets) can be defined according to the hierarchy system. Our goal was to develop such spectral clustering approach that preserves the structure of the dataset throughout the clustering procedure. The main contribution of this paper is a set of conditions for weighted graph construction used in spectral clustering. Following the requirements – given by the set of conditions – ensures that the hierarchical formation of the dataset remains unchanged, and therefore the clustering of data points imply the clustering of point-sets as well. The proposed spectral clustering algorithm was tested on three datasets, the results were compared to baseline methods and it can be concluded the algorithm with the proposed conditions always preserves the hierarchy structure.**

Index Terms—**spectral clustering, hierarchical dataset, graph construction**

I. INTRODUCTION

Many clustering methods have been developed, each of which uses a different induction principle [22][29]. Farley and Raftery [8] suggest dividing the clustering methods into two main groups: hierarchical and partitioning methods [25]; and other authors [10] suggest categorizing the methods into additional three main categories: density-based methods [5], model-based clustering [19] and grid-based methods [11]. Partitioning methods are divided into two groups: center-based and graph-theoretic clustering (spectral clustering).

Clusterability for spectral clustering, i.e. the problem of defining what is a "good" clustering, has been studied in some papers [1][2]. HSC [16] algorithm was developed to cluster arbitrarily shaped data more efficiently and accurately by combining spectral and hierarchical clustering techniques. Francky Fouedjio suggested a novel spectral clustering algorithm, which integrates such similarity measure that takes into account the spatial dependency of data, and therefore it is able to discover spatially contiguous and meaningful clusters in multivariate geostatistical data [9]. Furthermore, Li and Huang proposed an effective hierarchical clustering algorithm called SHC [15] that is based on the techniques of spectral clustering method. Although, none of the above studies focus on the case when the input dataset itself is a hierarchical dataset. The spectral clustering method is computationally expensive compared to e.g. center-based clustering, as it needs to store and manipulate similarities (or distances) between all pairs of points instead of only distances to centers [20].

A regular dataset $X = \{x_1, \dots, x_n\}$ consists of n data points and usually there is no pre-defined connection between any two (x_i, x_j) data points. Then clustering X into k clusters can be performed without any restriction on the composition of clusters; this process yields clusters $C_1, \dots, C_k$. On the other hand, a hierarchical dataset designates parent-child relationships between the points (as can be seen in Fig. 1); e.g. x_i and x_j could be the children of x_l, so in this case (x_i, x_j) together form a so called *point-set*.

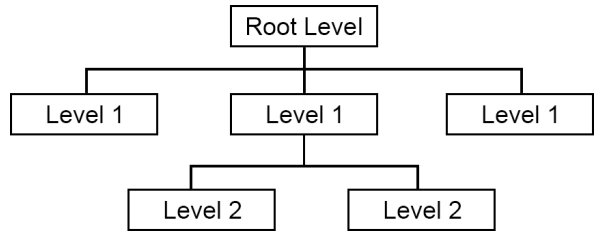

Figure 1. Structure of hierarchical dataset

Performing a traditional clustering algorithm also produces the $C_1, \dots, C_k$ clusters, however x_i could be part of C_g, while x_j could be assigned to C_h, and therefore the (x_i, x_j) point-set would be separated. This means that it is possible that clustering breaks the hierarchical structure of the dataset. In this paper we propose a set of conditions to control the weighted graph creation procedure in the course of spectral clustering [27] algorithm. Using the graph built accordingly will prevent the splitting of point-sets during clustering.

There are several different techniques to build the similarity graph in the spectral clustering, e.g. the ε-neighborhood, k-nearest neighbor and fully connected graphs [27]. The difference between them is how they determine whether two vertices (x_i and x_j) are connected by an edge or not. Let us

[1,3]Department of Telecommunications and Media Informatics, Budapest University of Technology and Economics, Budapest, Hungary; and Zs. Knoll ([2]) is student in BME Balatonfüred Student Research Group.

[1,3]E-mail: {pappd, szucs}@tmit.bme.hu

DOI: 10.36244/ICJ.2020.2.5

denote the similarity between x_i and x_j by s_{ij}; classic spectral clustering method creates a similarity graph G, and then proceed as follows:

1. First, a similarity matrix S is derived from G, where an s_{ij} element corresponds to the weight of the edge between x_i and x_j in G (in case of not connected points $s_{ij} = 0$).

2. Then diagonal matrix D is calculated by summing the columns of S, as can be seen in Eq. 1.

$$D = \{d_{ii}\};\ d_{ii} = \sum_j s_{ij} \tag{1}$$

3. After that the graph Laplacian matrix L is determined from S and D [12], which is a crucial part of spectral clustering, since different L lead to different approach. In this paper the symmetric normalized graph Laplacian is used, which can be computed as expressed in Eq. 2.

$$L_{sym} = D^{-1/2} * S * D^{-1/2} \tag{2}$$

4. Calculate the first k eigenvectors of L_{sym} and then construct a column matrix U from these vectors.

5. Perform K-means clustering on the rows of U to form $C_1, \dots, C_k$.

Majority of authors use graph Laplacian matrix [3][26] in the spectral clustering method, but there is possibility to use other type, so called adjacency matrix [4][14][21]. The eigen decomposition step can be computationally intensive. However, with an appropriate implementation, for example using sparse neighborhood graphs instead of all pairwise similarities, the memory and computational requirements can be solved. Several fast and approximate methods for spectral clustering have been proposed [6][17][28]. The traditional spectral clustering does not make any assumptions about the cluster shapes, but in our research, we dealt with point-sets instead of simple points, so points in a common set are expected to get a common cluster as well.

This concludes the spectral clustering and applying this procedure without any additional modification on a hierarchical dataset would result in a possible structure division. Two novel weight graphs were suggested, the Fully-Connected Weight Graph (FC-WG) and the Nearest Points of Point-sets Weight Graph (NPP-WG) [23]; that can influence the result of spectral clustering algorithms in such way that points belonging to the same point-set will stay together after the clustering is performed. To achieve this behavior the G similarity graph in the original algorithm should be replaced with either FC-WG or NPP-WG. The former is a fully connected graph, where the weight of an edge (w_{ij}) between two points (x_i, x_j) is calculated according to Eq. 3. Basically the weight is higher in case x_i and x_j are part of the same point-set ($\mathrm{x_i} \leftrightarrow \mathrm{x_j}$), and it is lower if they are not ($\mathrm{x_i} \nleftrightarrow \mathrm{x_j}$).

$$w_{ij} = \left\{ \begin{matrix} n \\ s_{ij} \end{matrix} \middle| \begin{matrix} \mathrm{x_i} \leftrightarrow \mathrm{x_j} \\ \mathrm{x_i} \nleftrightarrow \mathrm{x_j} \end{matrix} \right\} \tag{3}$$

where n denotes the number of points in the dataset. The NPP-WG is an incomplete graph, because connections between different point-sets are limited, however points that are part of the same point-set still form a fully connected subgraph; as can be seen in Eq. 4.

$$w_{ij} = \left\{ \begin{matrix} n \\ \mathrm{s_{ij}} \\ 0 \end{matrix} \middle| \begin{matrix} \mathrm{x_i} \leftrightarrow \mathrm{x_j} \\ \mathrm{x_i} \nleftrightarrow \mathrm{x_j}\ \&\ \mathrm{s_{ij}} \geq \mathrm{s_{it}}: \forall x_t \left(\mathrm{x_j} \leftrightarrow \mathrm{x_t}, x_j \neq x_t\right) \\ otherwise \end{matrix} \right\} \tag{4}$$

The fundamental idea behind these modifications is to connect any two points inside the same point-set with an increased edge weight that is higher than s_{ij}. Although this adjustment does not guarantee that the point-sets remain intact, it only reduces the chance to separate them. The focus of our research was to establish a set of conditions that the weighted graph creation process should satisfy in order to ensure the preservation of point-sets in the hierarchical dataset. In the next section we present the proposed condition system, then Section III contains the result of our experimental evaluation, and in the last section the conclusions of the research are summarized.

II. SET OF CONDITIONS FOR WEIGHTED GRAPH CONSTRUCTION

With appropriate conditions can be achieved that the points in the same point-set stay together, when using FC-WG and NPP-WG methods. For the formulas the following notations were used:

n: $number\ of\ points$
k: $number\ of\ clusters$
C_i: $i^{th}\ cluster$
$|C_i|$: $number\ of\ datapoints\ in\ the\ i^{th}\ cluster$
$\overline{C_i}$: $complement\ of\ C_i$
S_i: $i^{th}\ pointset$
A: $similarity\ matrix$
A_{ij}: $the\ j^{th} element\ of\ the\ i^{th}\ row\ in\ the\ A\ matrix$
Z: $edge\ weigths\ inside\ point\ sets$

The normalized spectral clustering is the relaxation of the normalized cut [26][27]:

$$Ncut(C_1, \dots, C_k) = \sum_{i=1}^{k} \frac{cut(C_i, \overline{C_i})}{vol(C_i)} = \\ = \frac{1}{2} \sum_{i=1}^{k} \frac{\sum_{j \in C_i} \sum_{l \in \overline{C_i}} A_{jl}}{\sum_{j \in C_i} \sum_{l \in \overline{C_i}} A_{jl} + \sum_{j \in C_i} \sum_{l \in C_i} A_{jl}} \tag{5}$$

We investigate two cases of cluster design, and express the formula presented by Eq. 5 in these situations. In the first case we assume that all points in the same point-set is assigned to the same cluster by the clustering algorithm. The second case is when a point (and only one point) was assigned into a different cluster than all other points of the point-set where this particular point belongs to. Note that in the second situation there is only one specific point that is separated from its point-set in the entire dataset.

Let InC^1 (inter cluster) be the sum of the edge weights between the clusters, and let WiC^1 (within cluster) be the sum of the edge weights inside the clusters; in the first investigated situation, which is denoted by "1" in the superscripts (as can be seen in Eq. 6 and Eq. 7).

$$InC^1(C_i) = \sum_{j \in C_i} \sum_{l \in \overline{C_i}} A_{jl} \tag{6}$$

$$WiC^1(C_i) = \sum_{j \mid S_j \in C_i} \left(\sum_{l \in S_j} \sum_{m \in S_j} Z + \sum_{l \in S_j} \sum_{m \in C_i \setminus S_j} A_{lm} \right) \tag{7}$$

According to Eq. 6 and Eq. 7, $Ncut$ of first case ($Ncut^1$) can be written as:

$$Ncut^1(C_1, \dots, C_k) = \frac{1}{2} \sum_{i=1}^{k} \frac{InC^1(C_i)}{InC^1(C_i) + WiC^1(C_i)} \tag{8}$$

Now let u be the separated point in the second case and C_k its assigned cluster, furthermore denote the cluster which contains all the other points from u's point-set by $C_{\overline{k^u}}$. In this second situation two different inter cluster and two different within cluster aggregates are examined, and the corresponding sub-cases are denoted in the superscripts; e.g. "2,1" refers for the first sub-case of the second situation. Define $InC^{2,1}$ as the sum of edge weights between cluster $C_{\overline{k^u}}$ and any other cluster, while $WiC^{2,1}$ represents the sum of the edge weights within $C_{\overline{k^u}}$; as expressed in Eq. 9 and Eq. 10.

$$InC^{2,1}(C_i, S_t, u) = \sum_{j \in C_{\overline{k^u}}} \sum_{l \in \overline{C_{\overline{k^u}} \cup S_t}} A_{jl} + \sum_{j \in S_t \setminus u} Z + \sum_{j \in \overline{C_{\overline{k^u}} \cup S_t \setminus u}} A_{uj} \tag{9}$$

$$WiC^{2,1}(C_i, S_t, u) = \sum_{j \neq t \mid S_j \in C_i} \left[\sum_{l \in S_j} \sum_{m \in S_j} Z + \sum_{l \in S_j} \sum_{m \in C_i \setminus S_j} A_{lm} \right] + \sum_{l \in C_i} A_{ul} + Z \tag{10}$$

For the summarized outer and inner edge weights of cluster C_k we introduce $InC^{2,2}$ and $WiC^{2,2}$, respectively; as can be seen in Eq. 11-12.

$$InC^{2,2}(C_i, S_t, u) = \sum_{j \in C_i} \sum_{l \in \overline{C_i \cup S_u}} A_{jl} + \sum_{j \in S_t \setminus u} Z + \sum_{j \in \overline{S_t \setminus u}} A_{uj} \tag{11}$$

$$WiC^{2,2}(C_i, S_t, u) = \sum_{j \neq t \mid S_j \in C_k} \left[\sum_{l \in S_j} \sum_{m \in S_j} Z + \sum_{l \in S_j} \sum_{m \in C_k \setminus S_j} A_{lm} \right] + \sum_{j \in C_k \setminus u} A_{uj} \tag{12}$$

Based on the above equations $Ncut$ of second case ($Ncut^2$) can be expressed as:

$$\begin{aligned} Ncut^2(C_1, \dots, C_k) = \frac{1}{2} \frac{InC^1(C_i)}{InC^1(C_i) + WiC^1(C_i)} + \\ + \frac{1}{2} \frac{InC^{2,1}(C_{\overline{k^u}}, S_t, u)}{InC^{2,1}(C_{\overline{k^u}}, S_t, u) + WiC^{2,1}(C_{\overline{k^u}}, S_t, u)} + \\ + \frac{1}{2} \frac{InC^{2,2}(C_i, S_t, u)}{InC^{2,2}(C_i, S_t, u) + WiC^{2,2}(C_i, S_t, u)} \end{aligned} \tag{13}$$

We will define the value of Z so that it satisfies the condition that $Ncut^1$ should be lower than $Ncut^2$. To achieve this, we estimated the value of $Ncut^1$ from above, and estimate the value of $Ncut^2$ from below.

In order to estimate $Ncut^1$ from above (see Eq. 16), we substituted InC^1 with a larger and replaced the value of WiC^1 with a smaller quantity. The substitution in case of InC^1 was accomplished by setting the elements of A to 1, and maximizing the number of point-sets, while during the calculation of WiC^1 the values of the elements of A were changed to 0, and the number of point-sets was minimized; as can be seen in Eq. 14 and Eq. 15, respectively.

$$InC^1(C_i) \leq n * n * 1 = n^2 \tag{14}$$

$$\begin{aligned} WiC^1(C_i) \geq \sum_{j \mid S_j \in C_i} \left(1^2 * Z + |S_j| \left(|C_i| - |S_j|\right) * 0\right) \geq \\ \geq n * Z \end{aligned} \tag{15}$$

$$\begin{aligned} Ncut^1(C_1, \dots, C_k) \leq \frac{1}{2} \sum_{i=1}^{k} \frac{n^2}{n^2 + n * Z} = \\ = \frac{k * n^2}{n^2 + n * Z} = \frac{k * n}{n + Z} \end{aligned} \tag{16}$$

To estimate the value of $Ncut^2$ from below, the previously defined substitutions were reversed, thus when computing the sum of inner edge weights ($InC^{2,1}$ and $InC^{2,2}$) the matrix A contained only 0 elements, and the number of point-sets was minimized. In accordance with this, the elements of A was set to 1, and the number of point-sets was maximized when $WiC^{2,1}$ and $WiC^{2,2}$ were calculated.

$$InC^1(C_i) \geq \sum_{j \in C_i} \sum_{l \in \overline{C_i}} 0 = 0 \tag{17}$$

$$\begin{aligned} InC^{2,2}(C_i, S_t, u) \geq \\ \geq \sum_{j \in C_{k-1}} \sum_{l \in \overline{C_{k-1} \cup S_t}} 0 + 1 * Z + \sum_{j \in \overline{C_{k-1} \cup S_t \setminus u}} 0 = Z \end{aligned} \tag{18}$$

$$\begin{aligned} InC^{2,2}(C_i, S_t, u) \geq \\ \geq \sum_{j \in C_i} \sum_{l \in \overline{C_i \cup S_u}} 0 + \sum_{j \in S_t \setminus u} Z + \sum_{j \in \overline{S_t \setminus u}} 0 = Z \end{aligned} \tag{19}$$

$$\begin{gathered} WiC^{2,1}(C_i, S_t, u) \leq \\ \leq \sum_{j \neq t | S_j \in C_i} [n^2 * Z + n * n * 1] + (n-1) * 1 + Z \leq \\ \leq n * [n^2 Z + n^2] + n - 1 + Z \leq \\ \leq n^3 Z + n^3 + n \end{gathered} \tag{20}$$

$$\begin{gathered} WiC^{2,2}(C_i, S_t, u) \leq \\ \leq \sum_{j \neq t | S_j \in C_k} [n^2 * Z + n * n * 1] + (n-1) * 1 \leq \\ \leq n * [n^2 Z + n^2] + n - 1 \leq n^3 Z + n^3 + n \end{gathered} \tag{21}$$

$$\begin{gathered} Ncut^2(C_1, \dots, C_k) \geq \\ \geq 0 + \frac{Z}{Z + n^3 * Z + n^3 + n + Z} + \frac{Z}{Z + n^3 * Z + n^3 + n} \geq \\ \geq \frac{2 * Z}{Z + n^3 * Z + n^3 + n + Z} = \frac{2 * Z}{(n^3 + 2) * Z + n^3 + n} \end{gathered} \tag{22}$$

The value of $Ncut^1$ should be lower than $Ncut^2$ in every case. Furthermore, both of them contain a multiplier of ½, and thus it could be eliminated in the equations.

$$Ncut^1(C_1, \dots, C_k) < Ncut^2(C_1, \dots, C_k) \tag{23}$$

$$\frac{k * n}{n + Z} < \frac{2 * Z}{(n^3 + 2) * Z + n^3 + n} \tag{24}$$

$$0 < 2Z^2 + (2n - kn^4 - 2kn)Z - (kn^4 + kn^2) \tag{25}$$

$$Y = \sqrt{k^2 n^8 + 4k^2 n^5 + 4k^2 n^2 + 8kn^4 - 4kn^5 + 4n^2} \tag{26}$$

$$Z < \frac{kn^4 + 2kn - 2n - Y}{4} \tag{27}$$

or

$$Z > \frac{kn^4 + 2kn - 2n + Y}{4} \tag{28}$$

Both (27) and (28) fulfills the conditions in (24) and (25), but the value of (27) is negative in all cases (see Eq. 29, 30 and 31), which means that (27) can not be interpreted as a similarity value.

$$kn^4 + 2kn - 2n - Y < 0 \tag{29}$$

$$\begin{gathered} k^2 n^8 + 4k^2 n^2 + 4n^2 + 4k^2 n^5 - 4kn^5 - 8kn^2 < \\ < k^2 n^8 + 4k^2 n^5 + 4k^2 n^2 + 8kn^4 - 4kn^5 + 4n^2 \end{gathered} \tag{30}$$

$$0 < 8kn^4 + 8kn^2 \tag{31}$$

Based on the above, the similarity value between points in the same point-set should be higher than the $Z_{threshold}$ (see Eq. 32) to avoid the separation of point-sets during spectral clustering. This is only true if the values of the similarity function are between 0 and 1.

$$Z_{threshold} = \frac{kn^4 + 2kn - 2n + Y}{4} \tag{32}$$

Note that $Z_{threshold}$ could be a very large number, even for a reasonably sized dataset, and therefore some sort of normalization of the edge weights is advised to prevent numerical limitations during the matrix manipulations.

III. EXPERIMENTAL RESULTS

We conducted experiments on three hierarchical datasets to demonstrate the efficiency of the proposed approach. The Free Music Analysis (FMA) audio dataset contains 106,574 tracks from 16,341 artists and 14,854 albums, arranged in a hierarchical taxonomy of 161 genres [7]. The first test dataset composed from the top 12 genres of the hierarchy. To form the second one, the artists were sorted in a decreasing order based on their number of corresponding tracks, and then the top 50 artists were selected. We call the former FMA1 dataset and it contains 9,355 tracks from 1,829 albums, while the latter is called FMA2 dataset, which involves 1,171 albums consist of 10,848 tracks (as can be seen in Table 1). Each track in the FMA collection is represented by a 518-long vector and we used them as input of the spectral clustering algorithm. In this case tracks are equivalent to the points on the lowest level of the hierarchy, while albums are analogous to point-sets.

The third test dataset is a subset of the image collection used in the competition of PlantCLEF 2015 [13]. A total of 91,759 images belongs in this dataset, each of them is a photo of a plant taken from one of the 7 pre-defined types of viewpoint (branch, entire, flower, fruit, leaf, stem and leaf-scan). Images about the same plant are organized into so-called observations, 27,907 plant-observations altogether. The original dataset was filtered in accordance with the provided contextual metadata, thus low quality pictures were discarded. The remaining 26,093 plant images from 9,989 observations form the third test dataset, which is called PCLEF dataset (see Table 1). Furthermore, observations were considered as point-sets and images as points. However, representations were unavailable for PlantCLEF images in the competition, and therefore we extracted visual features from the images to generate so called high-level descriptor vectors. 128 dimensional SIFT (Scale Invariant Feature Transform [18]) features were computed on an image and then they were encoded into 65,536 dimensional Fisher-Vectors [24] based on a codebook of 256 Gaussians.

Table 1. Number of points, number of point-sets and number of clusters in FMA1, FMA2 and PCLEF test datasets

	#points	#point-sets	#clusters
FMA1	9,355	1,829	12
FMA2	10,848	1,171	50
PCLEF	26,093	9,989	988

Four different graph construction approaches were tested, and their results were evaluated during our experiments. In each

case, other steps of the spectral clustering were identical and only the appropriate graphs were changed, which are the following:

- Fully-Connected Weight Graph using n as edge weights inside the point-sets (FC-WG) [23], where n is the number of the points,
- Nearest Points of Point-sets Weight Graph using n as edge weights inside the point-sets (NPP-WG) [23], where n is the number of the points,
- Fully-Connected Weight Graph using Z as edge weights inside the point-sets (FC-WG(Z)),
- Nearest Points of Point-sets Weight Graph using Z as edge weights inside the point-sets (NPP-WG(Z)).

Table 3 shows the result got on all three test datasets using each of the four different weighted graphs (note that "#ps" stands for "number of point-sets" in the second column). As can be seen, by satisfying the proposed condition, both FC-WG(Z) and NPP-WG(Z) were able to retain all of the point-sets throughout the spectral clustering. On the other hand, FC-WG and NPP-WG methods were unable to preserve the hierarchical structure in each case. Based on these results we conclude that the condition of setting the weights (inside point-sets) to at least the value of $Z_{threshold}$ guarantees that clustering the points on the lowest level of the hierarchy implies the clustering of the point-sets as well, without breaking them apart.

Table 2. The result of the number of separated point-sets during the spectral clustering of FMA1, FMA2 and PCLEF datasets

		#separated point-sets			
	#ps	FC-WG	NPP-WG	FC-WG(Z)	NPP-WG(Z)
FMA1	1,829	2	**0**	**0**	**0**
FMA2	1,171	43	34	**0**	**0**
PCLEF	9,989	11	**0**	**0**	**0**

IV. DISCUSSION

The known clustering methods can group the points in multidimensional space (where the dimensions of the space are the features of the original items, so a point in this space represent the corresponding item in the original reality), but majority of them is not able to group point-sets. In this paper we focused on point-sets (points that are related to each other) instead of only points, where the point-sets can be grouped into larger groups, so a hierarchical structure describes this grouping of data, resulting a hierarchical dataset. We investigated spectral clustering methods in the clustering literature. Our goal was to develop such spectral clustering approach that preserves the structure of the dataset throughout the clustering procedure. The main contribution of this paper was a set of conditions for weight graph construction used in spectral clustering. Following the requirements – given by the conditions – ensures that the hierarchical formation of the dataset remains unchanged, and therefore the clustering of data points imply the clustering of point-sets as well.

The proposed spectral clustering algorithm with graph construction was tested on three datasets and the results were compared to baseline methods. On the first and second datasets, albums with songs (tracks) were clustered, where tracks are equivalent to the points on the lowest level of the hierarchy, while albums are analogous to point-sets. The third dataset consists of pictures of plants. Here the images of plants represent the points, and the species are the point-sets in the hierarchical dataset. On the obtained clusters, we examined the relationships between the points from the point of view of how they reflect the expected structure, thus it was possible to compare different clustering algorithms with different graph construction approaches.

We demonstrated the clustering in hierarchical datasets with two levels, however our method is able to operate in more levels as well. In general, the point-sets should be constructed based on dendrogram (hierarchical tree) of the multi-level dataset. The user selects the required level (the user can choose any level) in this dendrogram, as can be seen in the Fig 2., and the crossing lines determine the point-sets (5 point-sets in the example) with the corresponding leaves of the tree as points.

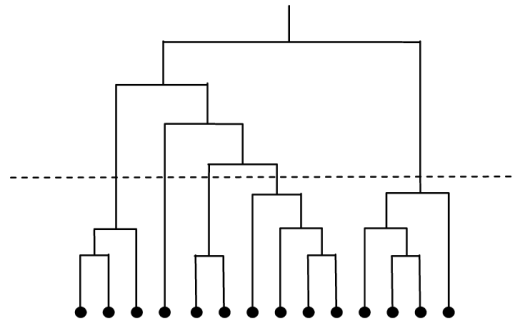

Figure 2. Determination the point-sets in hierarchical dataset

We investigated two clustering algorithms: FC-WG (Fully-Connected Weight Graph) and NPP-WG (Nearest Points of Point-sets Weight Graph), where these baseline methods used number of the points (n) as edge weights inside the point-sets, during the graph construction. From similarity matrix there are other possibilities to construct a graph, and we elaborated a condition for minimal weight among the points in a common point-set, while other weights come from directly the similarity matrix. So, two graph constructions (a baseline, and the elaborated one with $Z_{threshold}$ value) were investigated in both clustering algorithms, thus four different spectral clustering solutions were in the test: FC-WG, NPP-WG, FC-WG(Z), NPP-WG(Z).

The baseline algorithms using weighted graph approaches, where n values were in the edges, the points in a common point-set did not get into a common cluster; i.e. FC-WG and NPP-WG methods were unable to preserve the hierarchical structure. In the tests, by satisfying the proposed condition, both FC-WG(Z) and NPP-WG(Z) were able to retain all of the point-sets

throughout the spectral clustering. Based on these results we conclude that the condition of setting the weights (inside point-sets) to at least the value of $Z_{threshold}$ guarantees that clustering the points on the lowest level of the hierarchy implies the clustering of the point-sets as well, without breaking them apart.

The developed method is restricted to disjunct point-sets where the point-sets are not overlapping; in the future there is a plan to extend this method to hierarchical datasets with multiple class inheritance as well. The *Z* value influences the clustering result, as can be seen in the comparison with a previous work [23], where *Z* was equal to number of points; further thorough sensitivity analysis of *Z* value is a possible further development in the research.

ACKNOWLEDGMENT

The research has been supported by the European Union, co-financed by the European Social Fund (EFOP-3.6.2-16-2017-00013, Thematic Fundamental Research Collaborations Grounding Innovation in Informatics and Infocommunications).

REFERENCES

[1] Ackerman, M. and Ben-David, S. (2009). Clusterability: A theoretical study. In Dyk, D. A. V. and Welling, M., editors, Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics, AISTATS 2009, Clearwater Beach, Florida, USA, April 16-18, 2009, volume 5 of JMLR Proceedings, pages 1–8. JMLR.org.

[2] Balcan, M. and Braverman, M. (2009). Finding low error clusterings. In COLT 2009 - The 22nd Conference on Learning Theory, Montreal, Quebec, Canada, June 18-21, 2009

[3] Belkin M. and Niyogi. P. Laplacian eigenmaps for dimensionality reduction and data representation. Neural computation, 15(6):1373-1396, 2003. DOI: 10.1162/089976603321780317

[4] Brand M. and Huang. K. A unifying theorem for spectral embedding and clustering. In Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics, 2003.

[5] Bryant, A., & Cios, K. (2018). RNN-DBSCAN: A density-based clustering algorithm using reverse nearest neighbor density estimates. IEEE Transactions on Knowledge and Data Engineering, 30(6), 1109-1121. DOI: 10.1109/tkde.2017.2787640

[6] Chen, B., Gao, B., Liu, T.-Y., Chen, Y.-F., and Ma, W.-Y. (2006). Fast spectral clustering of data using sequential matrix compression. In Proceedings of the 17th European Conference on Machine Learning, ECML, pages 590–597. DOI: 10.1007/11871842_56

[7] Defferrard, M., Benzi, K., Vandergheynst, P., & Bresson, X. (2016). Fma: A dataset for music analysis. arXiv preprint arXiv:1612.01840.

[8] Farley C. and Raftery A.E., "How Many Clusters? Which Clustering Method? Answers Via Model-Based Cluster Analysis", Technical Report No. 329. Department of Statistics University of Washington, 1998. DOI: 10.1093/comjnl/41.8.578

[9] Fouedjio, F. (2017). A spectral clustering approach for multivariate geostatistical data. International Journal of Data Science and Analytics, 4(4), 301-312. DOI: 10.1007/s41060-017-0069-7

[10] Han, J. and Kamber, M. Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers, 2001.

[11] Hireche, C., Drias, H., & Moulai, H. (2020). Grid based clustering for satisfiability solving. Applied Soft Computing, Vol 88, 106069. DOI: 10.1016/j.asoc.2020.106069

[12] HU, P. (2012). Spectral Clustering Survey.

[13] Joly, A., Müller, H., Goeau, H., Glotin, H, Spampinato, C., Rauber, A., Bonnet, P., Vellinga, W. P., Fisher, B.: LifeCLEF 2015: multimedia life species identification challenges, Proceedings of CLEF 2015 (2015). DOI: 10.1007/978-3-319-24027-5_46

[14] Kannan, R., Vempala, S. and Vetta. A. On clusterings: Good, bad and spectral. Journal of the ACM (JACM), 51(3):497-515, 2004. DOI: 10.1109/sfcs.2000.892125

[15] Li, X., & Huang, J. (2009, November). SHC: a spectral algorithm for hierarchical clustering. In 2009 International Conference on Multimedia Information Networking and Security (Vol. 2, pp. 197-200). IEEE. DOI: 10.1109/mines.2009.107

[16] Liu, L., Chen, X., Luo, D., Lu, Y., Xu, G., & Liu, M. (2013). HSC: A spectral clustering algorithm combined with hierarchical method. Neural Network World, 23(6), 499-521. DOI: 10.14311/nnw.2013.23.031

[17] Liu, T.-Y., Yang, H.-Y., Zheng, X., Qin, T., and Ma, W.-Y. (2007). Fast large-scale spectral clustering by sequential shrinkage optimization. In Proceedings of the 29th European Conference on IR Research, pages 319–330. DOI: 10.1007/978-3-540-71496-5_30

[18] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. International journal of computer vision, 60(2), 91-110. DOI: 10.1023/b:visi.0000029664.99615.94

[19] McNicholas, P. D. (2016). Model-based clustering. Journal of Classification, 33(3), 331-373.

[20] Meila, M. (2016). Spectral Clustering: a Tutorial for the 2010's. In Handbook of cluster analysis (pp. 1-23). CRC Press.

[21] Ng, A.Y., Jordan, M.I. and Weiss. Y. On spectral clustering: Analysis and an algorithm. Advances in neural information processing systems, 2:849-856, 2002.

[22] Papp, D., & Szűcs, G. (2018). MMKK++ algorithm for clustering heterogeneous images into an unknown number of clusters. ELCVIA Electronic Letters on Computer Vision and Image Analysis, 16(3), 30-45. DOI: 10.5565/rev/elcvia.1054

[23] Papp, Dávid ; Szűcs, Gábor ; Knoll, Zsolt (2019). Machine preparation for human labelling of hierarchical train sets by spectral clustering, Proceedings of the 10th IEEE International Conference on Cognitive Infocommunications (CogInfoCom 2019), pp. 157-162. DOI: 10.1109/coginfocom47531.2019.9089906

[24] Perronnin, F., & Dance, C. (2007, June). Fisher kernels on visual vocabularies for image categorization. In 2007 IEEE conference on computer vision and pattern recognition (pp. 1-8). IEEE. DOI: 10.1109/cvpr.2007.383266

[25] Reddy, C. K., & Vinzamuri, B. (2018). A survey of partitional and hierarchical clustering algorithms. In Data Clustering (pp. 87-110). Chapman and Hall/CRC. DOI: 10.1201/9781315373515-4

[26] Shi J. and Malik. J. Normalized cuts and image segmentation. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 22(8):888-905, 2000.

[27] Von Luxburg, U. (2007). A tutorial on spectral clustering, Statistics and computing, 17(4), 395-416. DOI: 10.1007/s11222-007-9033-z

[28] Wauthier, F., Jojic, N., and Jordan, M. (2012). Active spectral clustering via iterative uncertainty reduction. In 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 1339–1347. DOI: 10.1145/2339530.2339737

[29] Xu, D., & Tian, Y. (2015). A comprehensive survey of clustering algorithms. Annals of Data Science, 2(2), 165-193. DOI: 10.1007/s40745-015-0040-1

Dávid Papp was born in 1990 in Hungary and he has received MSc in Computer Science (at specialization of media informatics) from Budapest University of Technology and Economics (BME) in 2016. He started his PhD work in 2016 in the field of Computer Science at the same university. His research topic includes artificial intelligence, machine learning, computer vision as well as development of algorithms on these fields (e.g. query strategies for classification of visual contents with active learning). He was awarded twice with the scholarship of New National Excellence Program of the Ministry of Human Capacities, in 2018 and 2019.

Zsolt Knoll was born in Szigetvár, Hungary in 1998. He is a BSc student at Budapest University of Technology and Economics, Faculty of Electrical Engineering and Informatics as computer engineering. He is a member of the Balatonfüred Student Research Group. In focus his research activities are data analytics and machine learning. He took second place in the Students' Scientific Conference at Budapest University of Technology and Economics.

Gábor Szűcs has received MSc in electrical engineering and PhD in computer science from Budapest University of Technology and Economics (BME) in 1994 and in 2002, respectively. He is an associate professor at Department of Telecommunications and Media Informatics of BME. His research areas are data science, artificial intelligence, deep learning, content-based image retrieval, multimedia mining. The number of his publications is more than 100. He is the president of the Artificial Intelligence Section of HTE (Scientific Association for Infocommunications), he is the leader of the research group DCLAB (Data Science and Content Technologies). He has earned János Bolyai Research Scholarship of the Hungarian Academy of Science some years ago.