Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 35 (2014) 1082 – 1091

Procedia Computer Science

18th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems - KES2014

Exploratory clustering and visualization

Agnieszka Nowak-Brzezińska*, Tomasz Xięski

University of Silesia, Institute of Computer Science, ul. Będzińska 39, 41-200 Sosnowiec, Poland

Abstract

In this work the topic of applying clustering as a knowledge extraction method from real-world data is discussed. Authors propose a two-phase cluster creation and visualization technique, which combines hierarchical and density-based algorithms [1]. What is more, authors analyze the impact of data sampling on the result of searching through such a structure. Particular attention was also given to the problem of cluster visualization. Authors review selected, two-dimensional approaches, stating their advantages and drawbacks in the context of representing complex cluster structures.

Peer-review under responsibility of KES International.
Keywords: clustering; DBSCAN; AHC; cluster visualization; Squarified Treemaps

1. Introduction

Extraction and discovery of knowledge hidden in the data have become particularly important in recent years, especially when taking into consideration the constantly growing amount of information stored in databases and data warehouses. The data is collected because it can potentially be the source of previously unknown and useful correlations, anomalies and trends [2]. However, the discovered patterns denominated in the form of an analytical model, may possess a complicated structure, which hinder the further analysis process. But not only the excessive amount of available information affects the difficulty of research. A more important factor is their complicated structure, both in terms of high dimensionality, as well as used data types. Records in a database are often described by various attributes including binary, discrete, continuous, categorical and those representing date or time. Such data (gathered from existing monitoring systems) can be called complex and will be presented in this paper.

The need for efficient information analysis methods is clearly visible in the mobile telephony field. According to Cisco Virtual Networking Index [3], the global data traffic generated by mobile phones and other mobile devices in 2012 reached 885 petabytes per month. In order to meet the requirements of their clients, mobile telephony providers introduce more and more services and facilities including access to digital TV or high quality video-conferences. Unfortunately, maintenance and monitoring of telecommunication networks (to guarantee a high level of availability)

* Corresponding author. Tel.: +48-32-368-9757; fax: +48-32-368-9703.
E-mail address: agnieszka.nowak@us.edu.pl

Peer-review under responsibility of KES International.
doi:10.1016/j.procs.2014.08.196

is a difficult task. Uneven load distribution, incompatibility between network devices and power loss are just some factors which have a direct impact on the quality of offered services. Therefore, needed are methods which would allow to extract and discover new knowledge about the devices behavior or potential network failures. Also to be successful, it is believed, that such methods should use visualization to take advantage of human cognitive abilities[4] and present extracted patterns in a more accessible way.

The aim of this paper is to introduce a two-phase cluster creation and visualization technique, which will be tested on a real-world dataset about the operation of mobile transceivers. In the first phase, data is clustered using the density-based DBSCAN[5] algorithm. Because this often results in a large number of created groups, the second step involves the usage of the Agglomerative Hierarchical Clustering (AHC)[6], but restricted only to the cluster representatives (created in the first phase). Then, the obtained structure is presented graphically, based on a space-filling Squarified Treemap[7] visualization technique.

2. Structure of the cell_loss dataset

The dataset being analyzed aggregated information about mobile transceivers (cells) operation from April 2010 to January 2011. It consists of 143486 objects (records) described by 19 attributes. The availability of each cell was measured in hourly time intervals. The structure of each data record is as follows:

- **cellname** – the identifier of a specific cell,
- **regionId** – identifier of a geographical region where the cell is located,
- **sectorId** – identifier of the direction in which the cell is broadcasting,
- **controllerId** – identifier of the controller to which a cell is connected to,
- **vendorId** – identifier of the cell's manufacturer,
- **signalLoss** – degree of inaccessibility in a given hour for a specific cell expressed as a real number from zero to one,
- **signalLossD** – degree of inaccessibility in a given hour for a specific cell expressed as an integer number from one to five[II],
- **eventId** – identifier of a particular event[III],
- **eventStart**, **eventEnd** – start and end times of an event,
- **date** – the date of measurement,
- **eventInterval** – duration of the event,
- **ifPlanned** – determines if the event was planned,
- **eventCount** – the number of registered events throughout the day, associated with a particular cell,
- **ifProblem** – determines if a (monitoring system) user detected a problem with a cell,
- **problemType** – determines if the reason of a problem is known and what type it is,
- **ifWorkflow** – determines if there was an work order issued for this cell in a given time,
- **ifWoINN** – defines if the work order was commissioned to the network maintenance department,
- **ifWoOther** – defines if the work order was commissioned to some other department.

The goal of analysis was to detect the most problematic transceivers (characterized by a high average level of unavailability and high number of registered events) using clustering algorithms and visualization techniques. Based on the results, the mobile telephony provider can optimize the network structure, which should directly translate into improving the quality of offered services.

Another important issue addressed in the experiments is sampling of the dataset. Several clustering algorithms (including CLARA[2] and CURE[8]) to discover groups in large datasets, reduce the size of input by drawing a random sample from the entire dataset. Unfortunately the reduction in input data due to sampling can affect the efficiency of a cluster analysis algorithm (in terms of clustering quality). Even the creators of CURE state that "since we do

[II] Attribute *signalLoss* was divided by a domain expert to five classes.
[III] An event is registered when a cell is undergoing maintenance or is inaccessible regardless of the reason.

not consider the entire data set, information about certain clusters may be missing from the input. As a result, our clustering algorithms may miss out certain clusters or incorrectly identify certain clusters"[8]. That is why the authors of this paper have decided to test how sampling would affect the values of precision and recall[2] in the task of searching through a cluster structure (created by applying the proposed two-phased method).

3. Related work in terms of cluster visualization

Authors of this paper have selected the DBSCAN density-based algorithm as a basis for discovering trends and relations between objects (like network devices). This method has several advantages over traditional hierarchical or partitioning approaches like: the possibility to discover groups of irregular shapes and sizes, resistance to outliers or relatively low computational complexity[IV]. Also preliminary experiments on a dataset gathering information about mobile transceivers (described in detail in[9]) confirmed that it is possible to apply the mentioned technique with success in an information retrieval task[V]. Unfortunately, when dealing with large volumes of data, the DBSCAN algorithm can also create a large number of clusters, which makes their analysis difficult (in the context of knowledge discovery or extraction). That is why the research process should be supported by the usage of clustering visualization methods, which will be described in this section. Advantages and disadvantages of these techniques will be presented on the example of disk usage by files and folders in one of the authors' Windows directory.

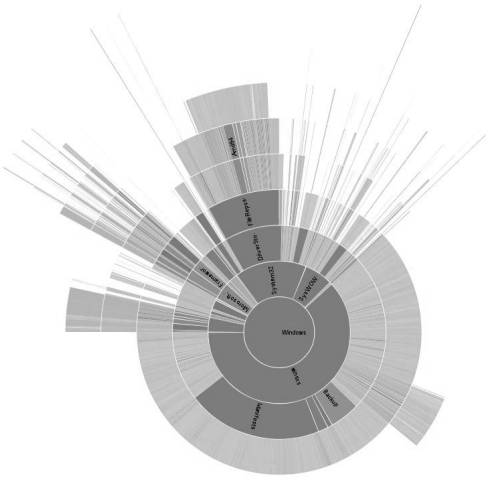

Fig. 1. Example of an sunburst tree.

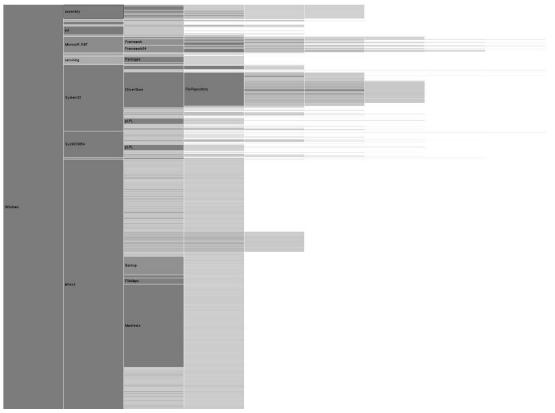

Fig. 2. Example of an icicle tree.

The first approach that will be described is called a *sunburst tree*. It is a visualization technique filling the display in a radial manner[11]. The initial part of the whole hierarchy (in this case the Windows directory) is located in the center of the workspace, as shown in Figure 1. The next levels of the hierarchy are drawn away from the center. Each level has an equal width, but based on the span of a particular circular segment, one can estimate the parameter value which a slice represents – in this case the storage occupancy of a folder or a file. Unfortunately small elements located on the outermost layer (corresponding to the lowest level in the hierarchy) are quite hard to see and could be omitted during the visual analysis[12]. There also exists an alternative version of this technique called the *sunray tree* in which the leaves are visualized as (sun) rays, but this factor does not affect the legibility of the whole diagram significantly.

Another visualization method of hierarchical cluster structures is named the *icicle tree*. Basically it is a *sunburst* transformed from polar to Cartesian coordinates. An example of this method was presented in Figure 2. The diagram is better suited for computer screens because it uses rectangles instead of circular segments, and therefore can be fitted

[IV] When using index structures, like R-trees, the average computational complexity is about $O(\log n)$, where n is the number of instances[5].
[V] Other approaches to the problem of clustering large volumes of complex data were discussed by authors of this paper in[10].

better. Furthermore, this diagram type is better when comparing two elements – radial slices are harder to examine than rectangles. Its main disadvantage is connected with representing further levels of the hierarchy – there may be still many regions of unused space. Like in the previous cases, this diagram has its alternative in the form of an *iceray tree*, which represents leaves as rays.

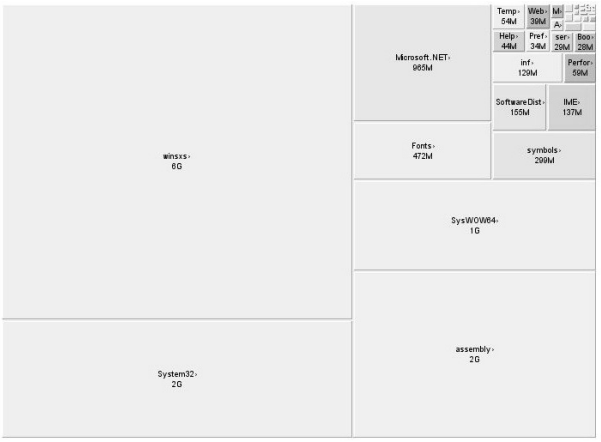

Fig. 3. Example of a classic treemap.

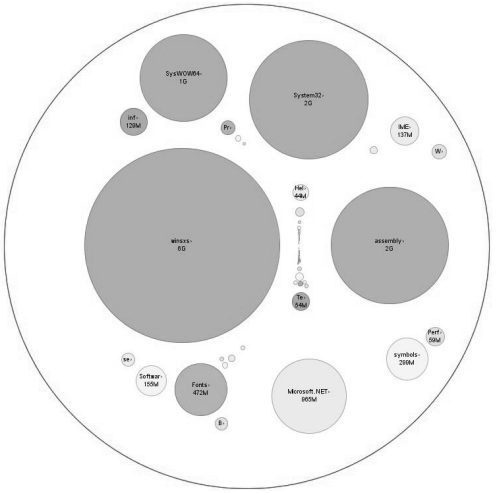

Fig. 4. Example of a circular treemap.

The *rectangular treemap*[13] is a visualization technique initially designed to present hierarchical structures by recursively dividing the available space into a number of rectangles which size is dependent on a chosen parameter (attribute), for example disk usage of a particular resource. The Figure 3 shows such a treemap, but restricted to only one hierarchy level, because for large datasets, it is usually a more readable approach. However, as mentioned previously, the technique can present multiple hierarchy levels at once. Additionally, because the described method guarantees to fill all of the available screen area, it allows to graphically lay out even a very complicated clustering structure, and thus has been selected as a technique used in the analysis of real-world, complex datasets.

The *circular treemap* shown in Figure 4 is an example of a diagram which uses the notion of nesting rather than element adhesion in visualizing hierarchies[12]. Similarly as before, the area of circles represents the disk usage of a folder or file. The main advantage of this technique is that nesting is clearly visible, but at the expense of wasted screen space. What is more, when the number of hierarchy elements and levels is large, circles of visually similar sizes can represent two very different (in terms of disk usage) resources, which may lead to erroneous conclusions.

4. Proposed knowledge discovery approach

Typically, the cluster analysis process uses just one algorithm, which divides data objects into meaningful groups. Unfortunately when the number of created clusters is large (thousands and more), it is nearly impossible to analyze such a structure in a reasonable time, even supported by a visualization technique[1]. That is why authors of this paper would like to propose a two-phase cluster creation technique, which combines hierarchical and density-based algorithms. It can be summarized in the following set of steps:

1. Feature selection based on the domain expert judgment.
2. Identification of missing values, duplicates or outliers and discretization of quantitative attributes (where applicable).
3. The usage of a density-based algorithm in order to create a set of clusters and their representatives.
4. Application of a hierarchical cluster generation algorithm, but restricted only to the group representatives formed in the previous step.

5. Visualization of the clustering structure using the treemap method[VI].

The data analysis of complex, real-world datasets begins with selecting features, which will be taken into account during clustering. This step should be done by an expert because it very often requires to have extensive domain knowledge. In case of the *cell_loss* dataset only 14 attributes were used during clustering. Those not included (according to a domain expert suggestion) were: *cellname*, *eventId*, *signalLoss*, *eventStart*, *eventEnd*.

Subsequently, in the next step, one should identify missing values, duplicates or outliers, by using descriptive statistics and plotting box plots or histograms[14]. This stage should be considered not only as data preprocessing, but also as a simple method of gaining insight and knowledge about the dataset[2]. For example the existence of missing values, in the case of a monitoring system may suggest its shut down (on purpose) or an error. Duplicate values may indicate, that the structure of the database could be changed to optimize memory usage. This step also could involve the discretization of quantitative attributes, possibly done by the domain expert. The proposed approach does not impose any restrictions on the chosen method of handling missing values, but some authors[15] suggest using linear (or logistic) regression (in case of quantitative attributes) and the k-nearest neighbor approach (in case of qualitative attributes) as the best methods for this task. Several outlier discovery techniques were reviewed in[16], and the one which is based on the interquartile range seems the most promising (and can be recommended as a default procedure).

The third step is to apply a density-based algorithm in order to generate clusters and their representatives. After the analysis of a number of clustering algorithms (which was described in[10]), the DBSCAN technique was chosen as optimal. Other clustering algorithms (like CLARA) use sampling (in case of which the clustering result may be far from optimal in terms of cohesion and separation) or are restricted to discovering spherical clusters (BIRCH)[6]. DBSCAN is based on the idea, that to define a new cluster (or to extend an existing one), a neighborhood around an object of a given radius (*Eps*) must contain at least a minimum number of objects (*MinPts*). The first step of the algorithm is to choose an arbitrary object p. Next a region query is performed, which finds the neighborhood of the object p. If this neighborhood contains fewer than *MinPts* objects, then p is considered as noise. Otherwise, a cluster is created and all objects in the neighborhood of p are placed in this cluster. In the next step, the neighborhood of each of p's neighbors is analyzed, to determine if it can be added to the cluster. If a cluster cannot be expanded further, the DBSCAN select another unclassified object. This procedure is repeated until all objects have been assigned to clusters or labeled as noise[5].

In[9] the authors introduced four concepts for creating group representatives and confirmed experimentally that the approach using the AND boolean logic operator is the most promising. The main advantage of defining the representative as the intersection of descriptors (attribute-value pairs) that describe objects belonging to one group is that it is known at first glance why objects were combined into one group and what values of particular attributes make them unique among other clusters. That is why this approach will also be used in this paper. But when studying large amounts of complex data, one has to expect that the number of generated groups may be too big for the analysis to be completed in a reasonable time. This statement is true even if the data analyst considers only the cluster representatives and tries to compare them. Therefore, the proposed by article authors concept involves also the usage of a second clustering technique.

The created cluster representatives should be considered as a generalization of knowledge (expressed as relationships and correlations between group members) hidden in the data itself. By using the representatives in the form of new input data, one can apply other data mining methods, which was not possible earlier because of the initial dataset size. The authors propose to apply AHC clustering, to create a two level hierarchy, which then will be presented to the user in the form of a rectangular treemap. Rectangles on the visualization symbolize specific groups, and their size is directly determined by number of objects belonging to the represented groups. Additionally, a number of statistics is associated with each rectangle like: the minimum, maximum, average values for attributes describing objects belonging to a cluster, number of objects in a group and its representative. It should be noted however, that the area of a rectangle can symbolize any quantitative parameter important from the analysis context. The data analyst should decide what particular feature is the most interesting at a given point – it may be the average value of registered events for all examined objects. This could potentially lead to the identification of the most defective network devices.

[VI] Because there are many ways in which a treemap layout can be created, the version called *Squarified Treemaps* was chosen. It guarantees the best aspect ratios[13], which is a very important factor when comparing the size of two rectangles.

Rectangles can also have assigned colors. In the case of the mobile telephony dataset an interesting parameter may be the degree of inaccessibility of a given transceiver. Then, a dark color of a particular rectangle would represent a high average value of this parameter, among all objects in the same cluster – it would be a group of often unavailable devices. A bright color of the rectangle would denote the opposite situation. Thanks to this approach, a data analyst has direct insight into the clustering structure, which in case of a large number of groups, can greatly facilitate the analysis' process.

5. Executed experiments

The experiments described in this section were to confirm the usefulness of applying the previously described, two-phase cluster analysis and visualization method in the process of knowledge extraction from a real-world dataset. It is also worth to note, that the conclusions from the presented experiments were confirmed by domain experts.

The aim of the first experiment was to determine the effect of sampling on the values of precision and recall in the task of searching through a cluster structure. In order to achieve this, four new datasets were created by reducing the original dataset (described in section 2) to respectively 1%, 10%, 25% and 50% of instances. To each one of the datasets a density based and then a hierarchical clustering algorithm was applied. Finally, a specific query was formed and a cluster which representative is most similar to the given query was returned in response. The efficiency of searching through cluster structures was examined based on the values of precision and recall. Precision was defined as the ratio of the number of relevant records retrieved to the total number of (relevant and irrelevant) records retrieved, whereas recall was the ratio of the number of relevant records retrieved to the number of relevant records in the dataset.

In order to determine the optimal input parameters for the density-based algorithm, several clusterings were created, each with different values of *Eps* and *MinPts* parameters. The quality of the generated clusterings was rated based on the following cluster evaluation measure:

$$clustering\ quality = \frac{\sum_{i=1}^{k} \frac{\sum_{x \in C_i} dist(x,u_i)}{|A| \cdot |C_i|}}{k} \tag{1}$$

where k – number of generated clusters, C_i – ith cluster, $dist(x, u_i)$ – Hamming distance[2] between an object x (belonging to cluster C_i) and the cluster's representative u_i, $|A|$ – number of attributes in the dataset, $|C_i|$ – number of objects belonging to cluster C_i.

The formula expressed in equation 1 measures cluster cohesion. Values closer to zero represent a better clustering (in terms of overall quality), whereas values closer to one designate the opposite. Results of the created clusterings are shown in Table 1. Because the density based algorithm can (for specific values of input parameters) generate clusters consisting of single objects (which can be regarded as outliers), in such case the distance of the object to its cluster representative is set as maximum. This way, the formula will not promote such clusters (consisting of only one object)[VII].

Cluster cohesion is only one of several other internal[VIII] cluster validity measures. One could also measure separation (to detect how distinct or well-separated a cluster is from others), but authors believe that cohesion is even more important, because it allows to check whether the data (objects) within groups are really well assigned to clusters – if there is a sufficient level of similarity between objects belonging to the same cluster. In the domain literature[2,17] there are defined several cluster validity measures based on cohesion and separation (like Sum of Squared Error or the Silhouette Coefficient) and could be used in addition to the measure presented in equation 1.

The results shown in Table 1 express the clustering quality for all 4 reduced datasets (created by arbitrary choosing a specified number of objects from the whole dataset) and the original one. The *MinPts* parameter is constant (set to 1) because for any other values, there were outliers present[IX]. A clustering is considered to be optimal when the number

[VII] Why the MinPts parameter is set to one and thus why such clusters can be generated is explained later in this section.

[VIII] Internal indexes (criteria) are used to measure the quality of a clustering structure without respect to external information like externally supplied class labels.

[IX] Setting *MinPts* to 1 can result in creating clusters with only one object (which could be also considered as outliers), but such clusters were left in the generated structure, because they will potentially be clustered later, after applying the AHC hierarchical algorithm.

Table 1. Selection of DBSCAN's optimal input parameters

Dataset name	Object count	Eps	MinPts	Group count	Clustering quality
cell_loss	143486	**1**	**1**	**7934**	**0.350634**
		2	1	1869	0.351028
		3	1	136	0.354063
		4	1	3	0.857143
		5	1	1	1.000000
cell_loss_1%	1438	1	1	1194	0.867672
		2	1	430	0.834385
		3	**1**	**186**	**0.825653**
		4	1	48	0.875000
		5	1	1	1.000000
cell_loss_10%	14384	**1**	**1**	**4746**	**0.687872**
		2	1	1355	0.739009
		3	1	372	0.732719
		4	1	30	0.864286
		5	1	1	1.000000
cell_loss_25%	35961	**1**	**1**	**6301**	**0.596005**
		2	1	1805	0.605144
		3	1	327	0.651813
		4	1	11	0.753247
		5	1	1	1.000000
cell_loss_50%	71743	**1**	**1**	**7200**	**0.481022**
		2	1	1967	0.487363
		3	1	216	0.494048
		4	1	7	0.48498
		5	1	1	1.000000

of outliers is minimal and the cluster validity measure is closest to zero. That is why, optimal input parameters for the DBSCAN algorithm are as follows: $Eps = 3$, $MinPts = 1$ for the dataset which consists of 1438 objects and $Eps = 1$, $MinPts = 1$ for all other datasets.

The next step of the proposed clustering approach involves using an agglomerative AHC algorithm, restricted only to cluster representatives (generated in the previous phase). This will allow to further reduce the number of obtained clusters, so that they can be visualized and analyzed in a reasonable amount of time (to gain new knowledge). This will also speed up the searching process. For each dataset, average linkage was used as the linkage criterion, and the generated clustering hierarchy was cut down to a fixed number of clusters. The number of clusters was defined as $\sqrt{N}$, where N is the number of objects in a dataset[18]. This procedure resulted in the creation of respectively: 397, 38, 120, 190, 268 clusters (for cell_loss, cell_loss_1%, cell_loss_10%, cell_loss_25%, cell_loss_50%).

Table 2. First experiment results

	cell_loss	**cell_loss_1%**	**cell_loss_10%**	**cell_loss_25%**	**cell_loss_50%**
Number of relevant objects	87	0	3	16	41
Number of objects returned in response	65	1	1	2	31
Recall	0.747126	0.000000	0.000000	0.125000	0.756098
Recall (with regard to the whole cell_loss set)	0.747126	0.000000	0.000000	0.022988	0.356321
Precision	1.000000	0.000000	0.000000	1.000000	1.000000

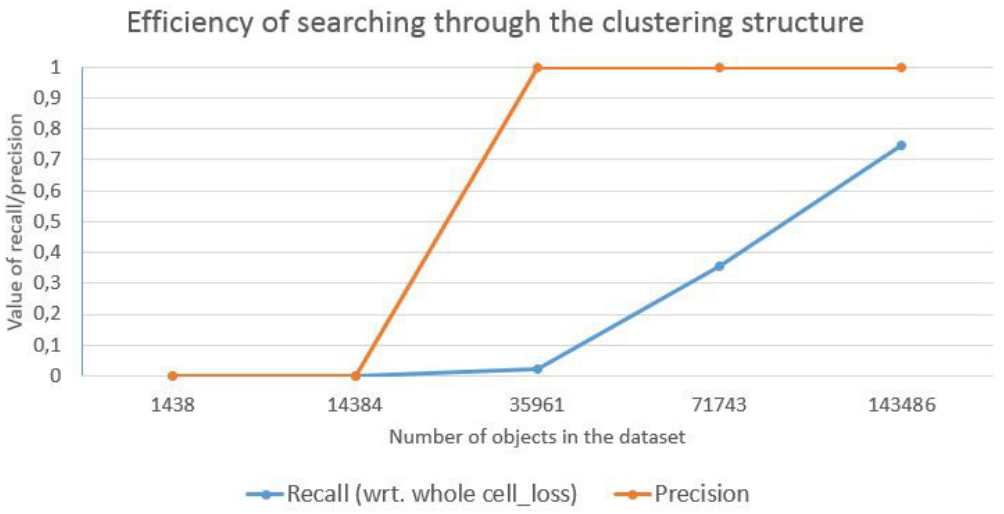

Fig. 5. Efficiency of searching through the clustering structure.

Having generated the final clustering structure for each dataset, a query was formed by a domain expert to retrieve all those transceivers from a specific vendor and region, which were unavailable for two hours because of a planned maintenance[X]. This query was compared to each cluster representative and a cluster which representative is most similar (to the query) was returned in response[XI]. Finally the values of precision and recall were calculated, which is shown in Table 2.

Results from the first experiment indicate that random sampling can have a very negative effect on the search efficiency (measured by precision and recall) as well as the clustering quality (because values for the clustering validity index in Table 1 are worse for reduced datasets compared to the original one).

Values presented in Table 2 clearly state that for this particular dataset, a sample of 25% (of objects) or less is not representative and causes a significant drop both in terms of relevant objects (to the given query) as well as recall. In two cases (cell_loss_1% and cell_loss_10%) the user did not get any relevant objects in response. The value of recall for the dataset containing 50% of instances is similar to the whole cell_loss dataset, but only when taking into consideration the fact, that there are only 41 relevant objects to the query, whereas in the whole dataset there are 87 such objects. That is why the authors included also a row in the analyzed Table, which includes the values of recall defined as the ratio of the number of relevant records retrieved to the number of relevant records in the whole cell_loss dataset (before sampling). In this case, one can state that recall is directly proportional to the sample size[XII]. That is why reviewing other sampling techniques on real-world datasets should be a key point of further research. To clearly show the relationship between the recall and precision parameters with regard to the dataset size, a line graph was created and shown in Figure 5.

After analyzing the graph presented in Figure 5 one can state that it is not only the sampling algorithm that has a direct impact on the efficiency of searching through the generated clustering structure. A more important factor is the size of the dataset. The more data is available for clustering, the higher is the clustering quality and in consequence the values of recall and precision. This leads to a conclusion that big datasets allow to discover and extract more knowledge (which potentially is of higher value and quality).

The second executed experiment concerned the usage of the proposed cluster creation and visualization technique. A DBSCAN clustering algorithm was applied to the cell_loss dataset (with previously determined parameters $Eps = 1$, $MinPts = 1$) resulting in the creation of nearly eight thousand groups containing from 1 to 6279 objects. As stated

[X] This is an example of a typical question, which is directed to a relation database which stores the described dataset.
[XI] Similarity between the cluster representative and the query was defined as the number of common features (descriptors).
[XII] Authors also executed similar experiments, by formulating other queries like the one presented, and conclusions were the same.

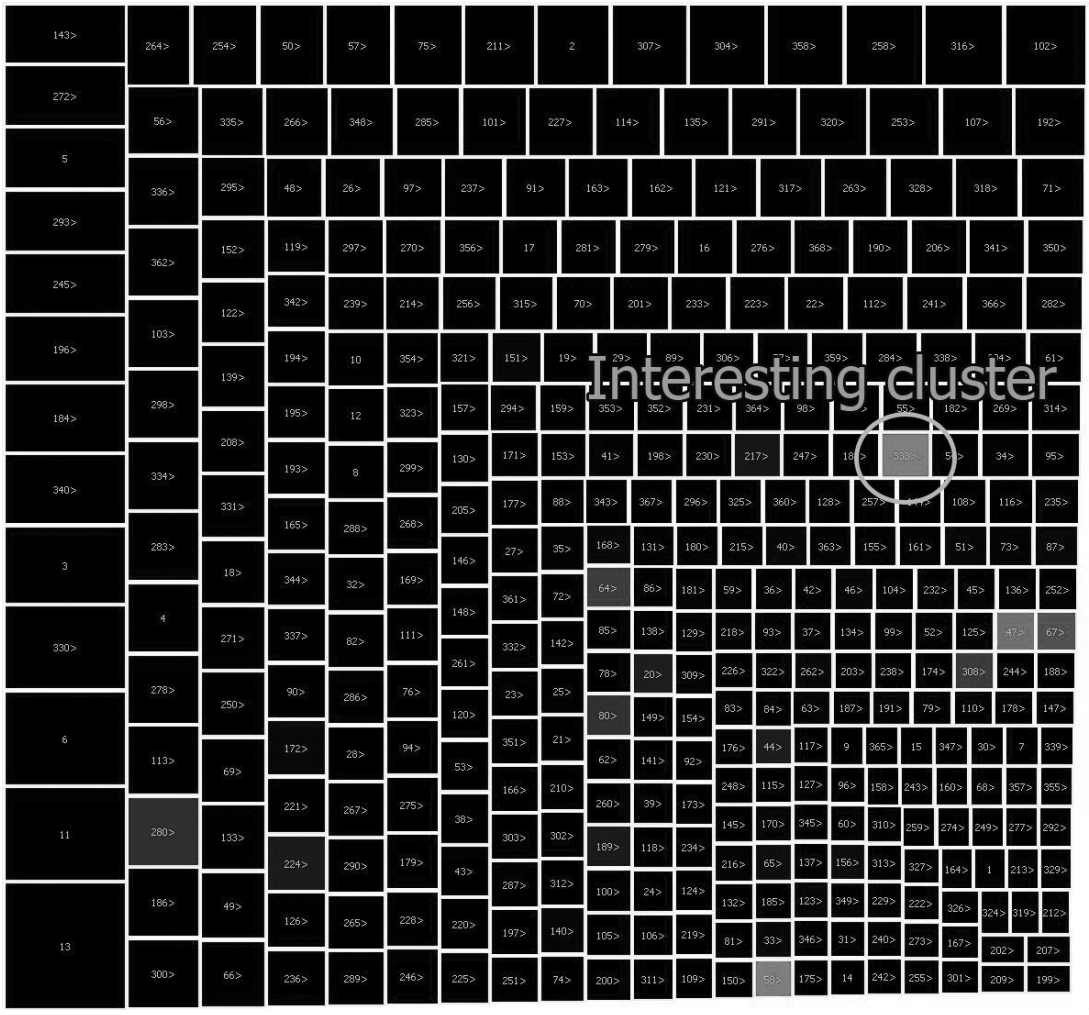

Fig. 6. A rectangular treemap generated using the proposed clustering approach.

earlier, this number of clusters is too large to be examined in a reasonable time. Therefore, the next step involved running an AHC agglomerative algorithm to obtain a two-level hierarchical structure, and thus significantly reduce the resulting number of clusters. As a result, the AHC algorithm generated nearly 400 clusters. Still, it is too much to be effectively analyzed by an expert. That is why the next step was to visualize the obtained structure on a rectangular treemap as shown in Figure 6.

In the Figure 6, the rectangle size represents the average number of registered events connected with the operation of cells belonging to one cluster, while the color determines the level of signal loss. The data analyst should look for big rectangles (which translates to a lot of events), and with a very bright color (which is more important for a network operator because it has a bigger effect on the quality of offered services). On this basis, one can locate and interesting cluster number *333* (marked on the Figure 6 with an ellipse).

The *333* cluster aggregates eleven records from the dataset, which describe three transceivers identified by *55171B2*, *58331A2*, *58331A1*. All of these cells originate from the same vendor and belong the *140* controller. Particularly interesting is the *55171B2*, because it was inactive for 119 hours. Thus, the presented technique allowed to identify three problematic devices, which should be further examined by a serviceman. It is also worth to mention, that the device *55171B2* has not been detected by other methods, although the authors tried different approaches to discover

it (for example by formulating specific SQL statements). Other interesting clusters (namely *280* and *58*) were also analyzed, but they refer to the same devices.

6. Summary

The aim of this paper was to discuss the topic of applying clustering as a knowledge extraction method from real-world, complex data. A two-phase cluster creation and visualization technique, which combines hierarchical and density-based algorithms was introduced. A single data mining method or algorithm, in the context of big data analysis, does not satisfy the requirements which modern knowledge discovery techniques have to fulfill – for example it is not possible to apply directly the AHC clustering to large datasets because of memory or computational complexity requirements. That is why a hybrid approach seems promising. Another important issue addressed in the experiments was to determine the effect of sampling on the values of precision and recall in the task of searching through a cluster structure (generated by the proposed approach). The third key objective of this study was to present solutions for visualization of clustering structures currently found in literature, with particular emphasis on their shortcomings and problems. The theoretical considerations were supported by two computational experiments.

Results from the first experiments confirmed, that random sampling can have a very negative effect on the search efficiency (measured by precision and recall) as well as the clustering quality. The generated hierarchical structure can be quickly searched without reducing the quality of clustering or recall of the response (to a given query). The second experiment verified that the proposed clustering technique allows to extract new knowledge (in the form of a network device that should be examined in actual working conditions and possibly repaired), which was not discovered by other means.

Acknowledgements

This work is a part of the project *"Exploration of rule knowledge bases"* founded by the Polish National Science Centre (NCN: 2011/03/D/S T6/03027).

References

1. Nowak-Brzezińska, A., Xięski, T.. Methods of complex data representation. *Studia Informatica* 2013;**34**(2A (111)):215–226. [in polish].
2. Han, J., Kamber, M., Pei, J.. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers Inc.; 2011.
3. Cisco, . Cisco visual networking index: Forecast and methodology, 2012-2017. Tech. Rep.; Cisco Systems; 2013.
4. Simoff, S.J., Böhlen, M.H., Mazeika, A., editors. *Visual Data Mining: Theory, Techniques and Tools for Visual Analytics*. Springer-Verlag; 2008.
5. Ester, M., Kriegel, H.P., Sander, J., Xu, X.. A density-based algorithm for discovering clusters in large spatial databases with noise. In: *In Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining*. AAAI Press; 1996, p. 226–231.
6. Berry, M.W., Browne, M., editors. *Lecture Notes in Data Mining*. World Scientific Publishing Co. Pte. Ltd; 2006.
7. Bruls, M., Huizing, K., van Wijk, J.. Squarified treemaps. In: *In Proceedings of the Joint Eurographics and IEEE TCVG Symposium on Visualization*. Press; 1999, p. 33–42.
8. Guha, S., Rastogi, R., Shim, K.. Cure: An efficient clustering algorithm for large databases. In: *Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data*. ACM; 1998, p. 73–84.
9. Wakulicz-Deja, A., Nowak-Brzezińska, A., Xięski, T.. Efficiency of complex data clustering. In: *Proceedings of the 6th International Conference on Rough Sets and Knowledge Technology*. Springer-Verlag; 2011, p. 636–641.
10. Wakulicz-Deja, A., Nowak-Brzezińska, A., Xięski, T.. Density-based method for clustering and visualization of complex data. In: *Proceedings of the 8th International Conference RSCTC 2012*. Springer-Verlag; 2012, p. 142–149.
11. Stasko, J.. An evaluation of space-filling information visualizations for depicting hierarchical structures. *Int J Hum-Comput Stud* 2000; **53**(5):663–694.
12. Bladh, T., Carr, D.A., Scholl, J.. Extending tree-maps to three dimensions: A comparative study. In: *Proceedings of the 6th Asia-Pacific Conference on Computer-Human Interaction (APCHI 2004)*. Springer-Verlag; 2004, p. 50–59.
13. Shneiderman, B., Wattenberg, M.. Ordered treemap layouts. In: *Proceedings of the IEEE Symposium on Information Visualization 2001 (INFOVIS'01)*. IEEE Computer Society; 2001, p. 73–78.
14. Dasu, T., Johnson, T.. *Exploratory Data Mining and Data Cleaning*. John Wiley & Sons, Inc.; 2003.
15. Gatnar, E.. *Symbolic methods for data classification*. Wydawnictwo Naukowe PWN; 1998. [in polish].
16. Nowak-Brzezińska, A.. Outlier mining in rule-based knowledge bases. *Studia Informatica* 2012;**33**(2A (105)):479–492. [in polish].
17. Tan, P.N., Steinbach, M., Kumar, V.. *Introduction to Data Mining*. Addison-Wesley Longman Publishing Co., Inc.; 2005.
18. Gelman, A., Hill, J.. *Data Analysis Using Regression and Multilevel/Hierarchical Models*. Cambridge University Press; 2007.