Interactive Data Visualization of Fuzzy Membership Functions and Fuzzy Clustering

Olga Chovancova, Jana Laukova, Adam Dudas, and Jozef Kostolny
Faculty of Management Science and Informatics,
University of Zilina, Zilina, Slovakia
{olga.chovancova, adam.dudas, jozef.kostolny}@fri.uniza.sk

***Abstract*—Since the volume of currently available data is growing from day to day, it is necessary to design and implement tools for a fast and comprehensive gain of knowledge from datasets created with this data. We can use the visualization of data as one of these methods - it is human and easy to understand. The presented paper aims to design and implement web application, which can be used for interactive visualization of data gained from fuzzification algorithms and fuzzy clustering. In this application, the user can interact with the visualization system and change form of input data, parameters, and visualization method.**

***Index Terms*—Fuzzification, fuzzy clustering, fuzzy membership function, interactivity, data visualization.**

I. Introduction

The volume of currently available data is large and growing fast every day. Data can be described as information which is processed or stored in computer. This information can be stored in the form of text documents, images, sound clips or programs. Steep growth of volume of data leads to large datasets, containing great number of detailed information about data. That is, why it is important to gain information and use knowledge, which is stored in large datasets [1]. Visualization of data can be used as mean of gaining knowledge from data contained in sizable datasets.

There are several tools for data visualization. The problem is that there are no practical (user-friendly) and free tools for visualization of data gained for fuzzification and fuzzy clustering. Some of the tools for visualization are Weka, Orange, RapidMiner, TANAGRA, KNIME, and MATLAB.

Weka [2] fits for data analysis, clustering, and predictive modeling. It contains algorithms and visualization tools for machine learning. It contains a graphic user interface, which allows users to access all functions without deep programming skills. MATLAB Fuzzy Logic Toolbox [3], provides functions for analyzing, designing, and simulating systems based on fuzzy logic. Orange [4] is a software package with open source code, which can be used for machine learning and data analysis. It is suitable for data visualization, predictive modeling, text analysis, and it brings interactive methods for the analysis of data. RapidMiner [5] contains algorithms for classification, clustering, and data regression. It also contains accessories for text analysis and analysis of data from streams. TANAGRA [6] is used in data visualization, descriptive statistics, regression, clustering, and classification. Results are presented in HTML, which is suitable for easy export and visualization of output data.

In the section II is presented elementary information about fuzzy sets and clustering. Section III is an introduction to the visualization of fuzzy membership functions. Section IV contains a visualization of fuzzy clustering. Section V is focused on the description of the technologies used in the process of implementation of our application. Lastly, in Section VI, we present an application created for interactive data visualization.

II. Introduction to Fuzzy Clustering

Fuzzy logic is an approach, which generalizes conventional logic and production rules with so called linguistic variables and linguistic rules.

Linguistic variable is a special type of variable, which differs from numerical variables in which its values are worded expressions. Every linguistic variable has assigned meaning, i.e. specific fuzzy set defined on common universal set.

The theory of fuzzy sets is generalization of classic theory of sets. Fuzzy sets are used in processing of vague data. In classic theory of sets, each element does or does not fall into set. We can define characteristic function for classic sets, which is assignment defined as follows:

$$\mu_S : U \to \{0, 1\} \tag{1}$$

It is assignment of value 0 - not belonging or value 1 - belonging, to each element $x \in U$. The definition field of characteristic function is U called universe. Universe is set of all values, about which we decide whether they belong or do not belong to given set. We say: $S \subseteq U$

We can define characteristic function of classic set as follows [7]:

$$\mu_S(x) = \begin{cases} 1 & x \in S \\ 0 & x \notin S \end{cases} \tag{2}$$

Extension of this concept is established in theory of fuzzy sets, where element can fall to the set partially. Vagueness is in the problem of affiliation of element to a set.

Rate that defines elements belonging to a set is described by membership degree. Let A be fuzzy set. Membership degree of element x to set A is defined by $\mu_A(x)$. We can say, that $\mu_A(x)$ is membership function of fuzzy set A [7].

Membership function $\mu_A(x)$ is similar to characteristic function of classic set. They differ in field of values. Membership function is assignment:

$$\mu_A : U \rightarrow \quad < 0, 1 > \tag{3}$$

Field of values is in this case:

$$\mu_A(x) \in \quad < 0, 1 > \tag{4}$$

Clustering is method, which divides input set of objects to groups called clusters. Elements contained in same cluster are considered similar, while elements of different clusters are not similar. We can use two types of clustering: *hard* and *soft*. In *hard* clustering, there is no overlap between two or more clusters. Elements either belong to cluster or not. *Soft* clustering allows overlap between two or more clusters.

Fuzzy clustering is form of clustering, in which any element can can belong to more than one cluster. Hence, to every element is assigned membership degree from interval 0 to 1. Well known method of fuzzy clustering is Fuzzy C-Means(FCM). This method was presented by author of [8] and later extended by [9].

We have set $X = \{x_1, x_2, \ldots, x_n\}$ and number of clusters *k*. Fuzzy clusters can be expressed as membership matrix μ_X, which is in relation 5, where rows of matrix pose as elements and columns as clusters. Each element of matrix defines degree of membership of element to given cluster:

$$\mu_X = \begin{bmatrix} \mu_{11} & \cdots & \mu_{1k} \\ \vdots & \ddots & \vdots \\ \mu_{n1} & \cdots & \mu_{nk} \end{bmatrix} \tag{5}$$

Sum total of all membership degrees in given row is equal to 1. This can be illustrated by relation:

$$\sum_{i\ =\ 1}^{k} \mu_{ik} = 1, \forall k \tag{6}$$

A. FCM algorithm

Let $X = \{x_1, \ldots, x_n\} \subset R$ be an unlabeled dataset. The FCM clustering algorithm partitions the dataset *X* into *c* homogeneous subsets represented as fuzzy sets $(B_i, i = \{1, 2, \ldots, c\})$. For the number of clusters *c* given by user, the objective is to obtain the fuzzy *c*-partition $B = \{B_1, \ldots, B_c\}$ by minimizing the evaluation function J_m

$$\min J_m(U, V) = \sum_{j=1}^{n} \sum_{i=1}^{c} \mu_{ij}^m \parallel x_j - v_i \parallel^2, \tag{7}$$

where μ_{ij} are membership degrees of data point x_j with respect to the fuzzy cluster B_i, and are also elements of the $c \times n$ pattern matrix $U = [\mu_{ij}]$.

The *i*-th row of *U*, U_i, corresponds to a fuzzy cluster $B_i, V = \{v_1, \ldots, v_c\} \subset R$ is a vector of cluster centroids of the fuzzy cluster $\{B_1, B_2, \ldots, B_c\}$, and $\parallel x_j - v_i \parallel$ is the Euclidean norm between x_j and v_i. The parameter *m* controls the fuzziness of membership of each data. The cluster centroids and the respective membership functions, which are solutions of the constrained optimization problem in equation 7, are given by the following equations:

$$v_i = \frac{\sum_{j=1}^{n} (\mu_{ij})^m x_j}{\sum_{j=1}^{n} (\mu_{ij})^m}, 1 \leq i \leq c, \tag{8}$$

and

$$\mu_{ij} = \left[\sum_{k=1}^{c} \left(\frac{\parallel x_j - v_j \parallel^2}{\parallel x_j - v_k \parallel^2} \right)^{1/(m-1)} \right]^{-1}, \tag{9}$$

for $1 \leq i \leq c, 1 \leq j \leq n$

Equations 8 and 9 constitute an iterative optimization procedure. The goal is to iteratively improve a sequence of clusters $\{B_1, B_2, \ldots, B_c\}$ until no further improvement in $J_m(U, V)$ is possible. The FCM algorithm is executed in the following steps:

Step 1.Given a preselected number of clusters *c* and fuzzy factor *m*, $(m > 1)$, initialize the fuzzy partition matrix $U_i = [\mu_{ij}]$ for $i = 1, 2, \ldots, c$ such that

$$\sum_{i\ =\ 1}^{c} \mu_{ij} = 1.$$

Step 2.Calculate the fuzzy cluster centroid v_i for $i = 1, 2, \ldots, c$ using equation 8.

Step 3.Use eq. 9 to update the fuzzy membership $U_i = [\mu_{ij}]$ for $i = 1, 2, \ldots, c$.

Step 4.If the improvement in $J_m(U, V)$ is less than a certain threshold ε, then stop; otherwise go to Step 2.

Some of properties of fuzzy c-means algorithm are :

$$\mu_{ij} \in [0, 1], \qquad \forall i, j; \tag{10}$$

$$\sum_{j=1}^{C} \mu_{ik} = 1 \qquad \forall i; \tag{11}$$

$$0 < \sum_{i=1}^{N} \mu_{ij} < N \qquad \forall N. \tag{12}$$

Fuzzy clustering algorithms require, the user to predefine the number of clusters *c*, which is not always possible to know a priori. Selection of parameter *m* is in FCM often problematic. The problem of finding an optimal *c* is usually called the *cluster validity problem* [10]. In [11], Bezdek stated: *”no matter how good your index is; there is a dataset out there waiting to trick it (and you).”* Therefore, cluster validation is a challenging task.

III. Visualization of Fuzzification

Interactive visualization of data can be used as tool for interpretation and discovery of other dependencies. In this section we describe problem of interactive data visualization and analysis of tools and libraries for interactive data visualization.

Visualization is process of displaying data in graphical form. It allows user to interpret and analyze data effectively. It also

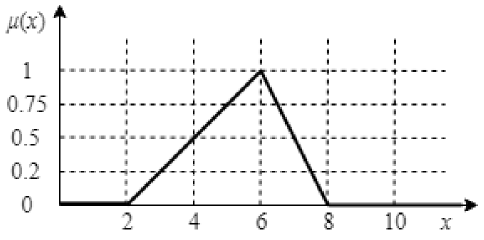


Figure 1: Example of triangular membership function

helps in decision making and saves time of data exploration [1].

Interactive visualization is process, which contains bidirectional communication between user and visualization system. It allows user to change parameters of visualization, such as shape, color, brightness, edges or texture. It can also help in discovery of information, which is otherwise hidden in large volume of data [1].

Fuzzification is transformation of numerical values to linguistic variables. In the fuzzification, we assign each element of numerical input data membership function and compute membership degree from interval 0 to 1.

This assignment is visualized by the membership function curves such as triangular, trapezoidal, linear, bell, Gauss or polynomial curve. Membership function curves can be composed of several elementary curves. Some of simplest fuzzy membership functions are triangular and trapezoidal functions.

A. Triangular membership function

Triangular membership function is the function formed with lines. This function, where *a, b, c* are parameters, is defined as follows:

$$\mu_X = \begin{cases} \frac{x-a}{b-a} & a \leq x \leq b \\ 1 & x = b \\ \frac{c-x}{c-b} & b \leq x \leq c \\ 0 & x \leq a \text{ or } c \leq x \end{cases} \tag{13}$$

For example, temperature is in between $2°C$ and $8°C$ and parameters $a = 2, b = 6, c = 8$. Triangular fuzzy membership function is:

$$\mu_A(x) = \begin{cases} \frac{x-2}{4} & 2 \leq x \leq 6 \\ 1 & x = 6 \\ \frac{8-x}{2} & 6 \leq x \leq 8 \\ 0 & x \leq 2 \text{ or } 8 \leq x \end{cases} \tag{14}$$

Graph of triangular membership function for temperature is presented in Figure 1.

B. Trapezoidal membership function

Trapezoidal membership function consists of trapezoid, where *a, b, c, d* are parameters and it is defined as:

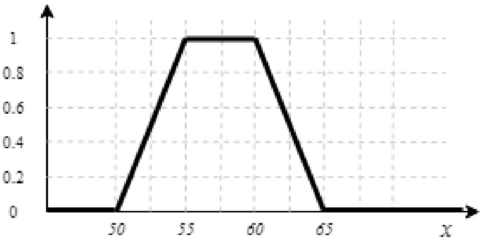


Figure 2: Example of trapezoidal membership function

$$\mu_X = \begin{cases} 0 & x \leq a \\ \frac{x-a}{b-a} & a \leq x \leq b \\ 1 & b \leq x \leq c \\ \frac{d-x}{d-c} & c \leq x \leq d \\ 0 & d \leq x \end{cases} \tag{15}$$

Parameters a and d represent the lower base and parameters b and c represent upper base of trapezoid.

Trapezoidal fuzzy membership function can be used for classification of age. If X is set of people in the age around 55 years, we expect, that people, who are over 65 years old can be classified as old. In this case the trapezoidal fuzzy membership is characterized with parameters $a = 50, b = 55, c = 60, d = 65$ and relation:

$$\mu_X = \begin{cases} 0 & x \leq 50 \\ \frac{x-50}{5} & 50 \leq x \leq 55 \\ 1 & 55 \leq x \leq 60 \\ \frac{65-x}{5} & 60 \leq x \leq 65 \\ 0 & 65 \leq x \end{cases} \tag{16}$$

Graph of trapezoidal fuzzy membership function for the example above is presented in Figure 2.

IV. Visualization of fuzzy clustering

Fuzzy clustering is one of most used for precise mirroring of real world. Result of fuzzy clustering is membership matrix, which can be used as complex multidimensional information. This type of information can make interpretation of fuzzy clustering results harder.

A. Radviz

Radviz is a method designed by Patrick Hoffman, for intuitive visualization of division of membership degrees for all data objects represented by points. This method is suitable for effective exchange of information between user and system, scalability while it is able to store important properties of original data. Radviz is used in biology, medical sciences, management and cybernetics [12].

B. *Visualization and interaction*

One of the results of fuzzy clustering is matrix of membership degrees μ, centers of individual clusters and values of validation indexes. Matrix μ_{nm} consist of *n* rows and *m* columns, representing *n* points divided to *m* clusters. Value μ_{nm} represents membership degree of point in relation to given cluster.

Creation of Radviz visualization consists of following steps:

1) **Visualization of points and arrangement of dimensional anchor** - Euclidean distances between centers of individual clusters create weighted non-directional graph *G*. Vertex of graph represents cluster and weight of edge represents Euclidean distance between centers of two clusters. With construction of spanning tree of graph *G* we create optimal layout of dimensional anchors.
2) **Visualization of points** - We can use two methods of visualization. First one is based on scatter chart. Each point in this chart is represented by circle. The circle is filled with color of cluster to which element belongs with greater membership degree. Second method of visualization is in form of pie chart. Each element is represented as pie chart and each section of pie chart represents the color of cluster and size of the section represents membership degree.
3) **Visualization of dimension anchors** - Color of dimensional anchors distinguishes individual clusters. Thickness of arc represents number of points, which belong to given cluster with high membership degree. The thicker the arc is, the more points belong to it. Detailed division of membership degrees can be visualized by line chart, which can be placed directly above each arc.
4) **Visualization of relations between clusters** - Some points in the cluster can belong to other clusters with some probability. This is called sharing of the relations between clusters. We can use *chord* diagram in combination with Radviz circle for visualization of these relations The thicker the *chord* diagram is and the higher its opacity is, the higher relationship between two clusters arises.
5) **Interactivity** - It allows user to point cursor on any point or cluster and display detail information about chosen data. Other functions of Radviz visualization are: filtering of points in clusters, changing mode of visualization of points in clusters and highlighting certain group of points.

C. *Parallel coordinate plot*

Another method for data visualization is parallel coordinate plot designed by Inselberg [13]. This method can be used for visualization of multidimensional numeric data, while comparing given attributes and analyzing relations between them. Parallel coordinates are used in algorithms for collision detection in aviation, data analysis, optimization and sorting of processes.

Problem of parallel coordinates plot is that while using high number of variables, plot becomes crowded and unreadable. This program can be solved by interactive visualization, where user can filter data of interest by pointing cursor on chosen axis or variable.

V. Implementation

There is great number of tools usable for design and implementation of web application. In this section of the paper, we present chosen frameworks and libraries used in the implementation of our application. We chose framework Vue.js which contains some functions of ReactJs and Angular and libraries for interactive visualization chart.js and D3.js. In the table I, we compare ReactJs, Angular and Vue.js.

Vue.js [14] is progressive JavaSript framework which contains detailed documentation. It can be used for creation of reactive web applications based on components or mobile applications using NativeScript and Ionic.

Since libraries chart.js and D3.js offer best functionality regarding our goal, we used them for the interactive visualization used in our application.

D3.js [15] is library used for effective manipulation with data documents. It is able to create dynamic and interactive data visualization in the web browsers. Data visualization in this library is done with the use of HTML, SVG and CSS standards. This library is fit for the creation of SVG graphs with smooth transitions and interactions. D3.js is flexible and fast.

Chart.js [16] is library usable for creation of animated interactive graphs via HTML element canvas. This library supports eight different types of responsive graphs. If there is no need for all eight types of graphs, user can choose only modules with the graphs needed [17].

A. *Architecture*

Our application is designed as *single-page application* (SPA), presented on the Figure 3. It is typical for SPA to load code of the application (HTML, CSS, JavaScript) only once. JavaScript caches actions of client (user) and instead of new query for server, application dynamically loads needed resources without reloading of the site. SPA is usable as desktop and mobile application and it is fast since user can obtain immediate feedback.

We use Vuex library for state management of our application. It serves as centralized storage space for all components of the application and contains rules for changing of the state

Table I: Comparison of ReactJs, Angular and Vue.js

Framework	Positives	Negatives
ReactJs	- Optional use of JSX - Components are reusable - Facebook support	- Syntax JSX - Usable only with JavaScript - Key functionality is added manually
Angular	- HTML, CSS - SPA are part of Angular - Google support	- TypeScript - Complex setup of project
Vue.js	- HTML, CSS, components are reusable - Built in basic functionality - Simple learning process	- No support of major companies - Key functionality is added manually

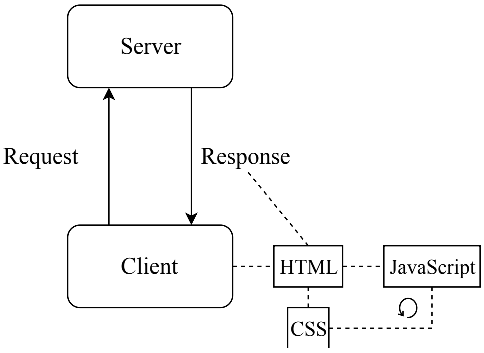


Figure 3: Single-page application (SPA)

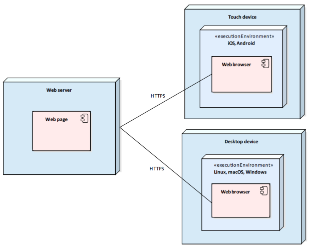


Figure 4: Diagram of applications deployment

of the site in predicted ways. The hearth of the Vuex library contains four objects:

- **State** - object containing shareable properties of the application.
- **Mutations** - functions, which are used for mutation of the state of storage space itself.
- **Actions** - functions responsible for calling all asynchronous actions and mutations.
- **Getters** - functions which contain computations and manipulations regarding storage space.

VI. Our Application

Our application is available at http://fuzzy.visualization.sk. Relations between components of the application, which are used for safe run of the application, are presented on the Figure 4. Our visualization tool is designed as responsive web application. On the Figure 5 is presented responsive behaviour of our application. The application was tested on four types of datasets presented below.

A. Dataset Iris

Iris is well known dataset used mainly in machine learning. This dataset contains 150 records each containing 5 attributes. First four attributes are measured in centimeters, fifth is text string encompassing three classes. Each of these classes contains 50 sub-records.

On the Figure 6 is presented an example of visualization of Iris dataset. First, we loaded dataset from examples in the application. Then, we chose the type of visualization (in this example, we chose point chart). We can see that attributes of data were clustered and color-coded based on the third argument.

Figure 7 presents possibilities of user - system interactivity:

1) **Axis renaming** - our application contains "*Change axis names*" button, which allows user to change names of x and y axis.
2) **Color palette change** - using button "*Change clusters color*", user can choose cluster and change its color on the graph.
3) **Parameter change** - lastly, the user can use "*Change parameters*" button to change parameters used in the visualization. User can change gradually pick indexes of columns for x axis, y axis and for clusters. Graph is reloaded with current parameters after last change in indexes.

B. Dataset Glass Identification

This dataset contains 214 types of data about glass and its properties. This data consists of 10 attributes - nine numeric values (chemical composition of glass) and one text string.

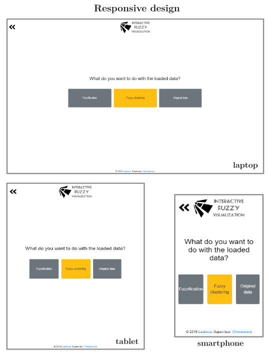
The Figure 8 shows fuzzification visualizations of the *Glass Identification* dataset. Since we chose fuzzification of the dataset, we need to create membership function of chosen attribute (user also chooses name of the function and coefficients). After using the "*Add*" button, tool visualized data properly - using membership function.

Figure 8 presents possibilities of user - system interactivity:

1) **Membership function change** - application offers button "*Change function*", which opens window for function change. This contains all properties of given function, coefficients, name and color of the function.
2) **Attribute change** - button "*Change attribute*" is used in case, that user needs to change properties of chosen attribute, analogically to function change.
3) **X axis change** - user can change x axis with the use of slider located under the axis itself.
4) **Adding of new function** - application contains button for adding function to the existing visualization.

C. Cars dataset

This dataset consists of 406 record related to older cars and their properties. Properties of these cars are presented as 9



What do you want to do with the loaded data?

© 2019 Laukova, Supervisor: Chovancova

Figure 5: Responsive behaviour of the application

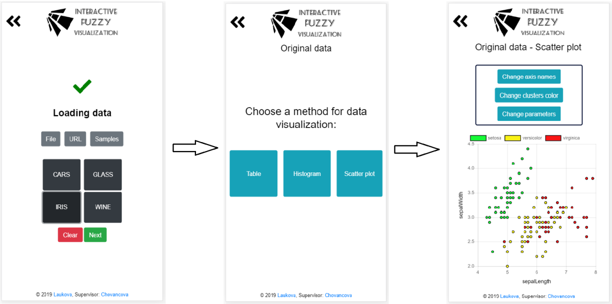


Figure 6: Visualization of Iris dataset

attributes - text string of cars model and eight numeric values related to technical components of each car.

On the Figure 9 we present visualization of *Cars* dataset. We used the table method for visualization of attributes chosen by user.

In the Figure 10 is presented parallel coordinates plot. We can see, that each variable has its own vertical axis and all variables are placed parallel to each other. Each axis has its own scale, since each variable is represented in different units.

Figure 10 presents possibilities of user - system interactivity:

1) **Selection of chosen attributes** - user can select attributes through simple table interface. Graph, which is the result of visualization, contains only chosen attributes.
2) **Order of the attributes** - user can drag y axis of each attribute to desired place in the graph.
3) **Color palette change** - with the use of *Change color* button, we can change colors used in parallel coordinates graph.

D. Wine Quality dataset

This dataset is result of chemical analysis of wines, which were grown in same region in Italy. Analysts measured 178 record, 14 attributes each. All of the mentioned attributes are numerical values such as type of wine (numeric value 1-3), alcohol content, acid content, etc.

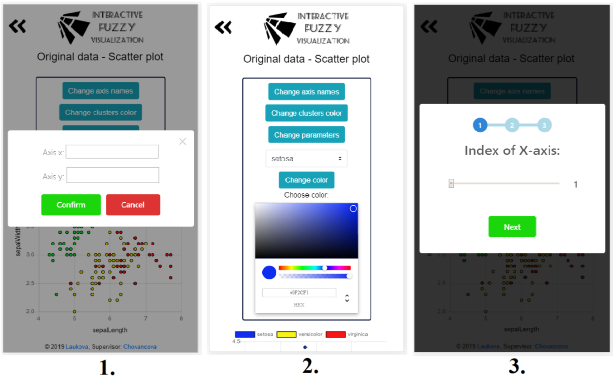
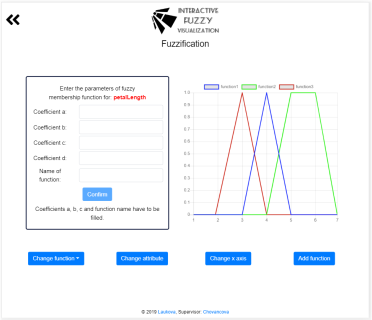


Figure 7: Interactivity on the Iris dataset



Enter the parameters of fuzzy membership function for: petalLength

Coefficient a:

Coefficient b:

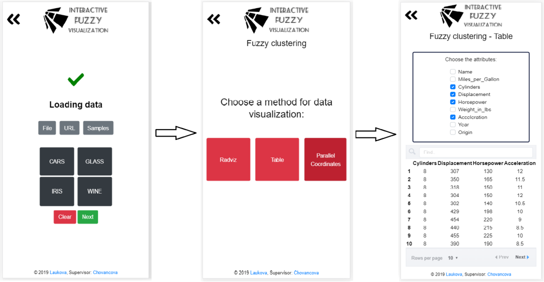
Coefficient c:

Coefficient d:

Name of function:

Coefficients a, b, c and function name have to be filled.

Figure 8: Visualisation of fuzzy membership function for *Glass* dataset



Choose the attributes:

☐ Name
☐ Miles_per_Gallon
☑ Cylinders
☑ Displacement
☑ Horsepower
☐ Weight_in_lbs
☑ Acceleration
☐ Year
☐ Origin

	Cylinders	Displacement	Horsepower	Acceleration
1	8	307	130	12
2	8	350	165	11.5
3	8	318	150	11
4	8	304	150	12
5	8	302	140	10.5
6	8	429	198	10
7	8	454	220	9
8	8	440	215	8.5
9	8	455	225	10
10	8	390	190	8.5

Figure 9: Visualization of *Cars* dataset

On the Figure 11 is presented visualization of Wine Quality dataset. This dataset is fit for fuzzy clustering and Radviz method of visualization.

1) **Selection of attributes** - Radviz method visualizes only attributes selected by user. These attributes can be changed through functions of application.
2) **View details for each points** - move the mouse over some point to see more information.

VII. Conclusion

The article presents an interactive web application for visualization of data obtained from fuzzy clustering and fuzzy algorithms.

Primary use of our application, presented in this paper, is in the teaching of fuzzy membership functions and fuzzy clustering. Students of courses focused on these sections of the curriculum can use this application in classes and at home.

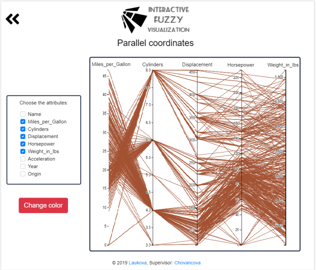
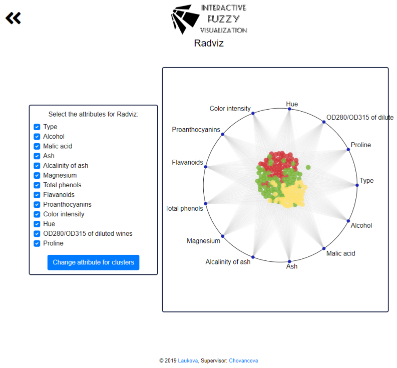


Figure 10: Interactivity on the *Cars* dataset



Select the attributes for Radviz:

- ☑ Type
- ☑ Alcohol
- ☑ Malic acid
- ☑ Ash
- ☑ Alcalinity of ash
- ☑ Magnesium
- ☑ Total phenols
- ☑ Flavanoids
- ☑ Proanthocyanins
- ☑ Color intensity
- ☑ Hue
- ☑ OD280/OD315 of diluted wines
- ☑ Proline

Figure 11: Visualization of *Wine* dataset

The main functionality of the application is to visualize data interactively. The application also allows visualization of the distribution of the original data. It can proceed with data, such as membership matrices, centers, validation index values, original data, and more. It also has interactive behavior and change of parameters.

In the future application can be extended for visualization of fuzzy decision trees, and also can be used in the education process.

References

[1] E. Zudilova-Seinstra, T. Adriaansen, and R. van Liere, "Overview of Interactive Visualization," *Adv. Inf. Knowl. Process.*, vol. 36, pp. 3–15, 2009.

[2] M. Hall, H. National, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, *The WEKA data mining software*, 2009, vol. 11, no. 1. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1656274.1656278

[3] "MATLAB." [Online]. Available: https://www.mathworks.com/products/matlab.html

[4] J. Demšar, T. Curk, A. Erjavec, Črt Gorup, T. Hočevar, M. Milutinovič, M. Možina, M. Polajnar, M. Toplak, A. Starič, M. Štajdohar, L. Umek, L. Žagar, J. Žbontar, M. Žitnik, and B. Zupan, "Orange: Data mining toolbox in python," *Journal of Machine Learning Research*, vol. 14, pp. 2349–2353, 2013. [Online]. Available: http://jmlr.org/papers/v14/demsar13a.html

[5] M. Hofmann and R. Klinkenberg, *RapidMiner: Data Mining Use Cases and Business Analytics Applications*. Chapman & Hall/CRC, 2013.

[6] G. Smith, J. Whitehead, and M. Mateas, "Tanagra: A mixed-initiative level design tool," in *Proceedings of the Fifth International Conference on the Foundations of Digital Games*, ser. FDG '10. New York, NY, USA: ACM, 2010, pp. 209–216. [Online]. Available: http://doi.acm.org/10.1145/1822348.1822376

[7] M. N. McAllister, "Fuzzy Logic with Engineering Applications (Timothy Ross)," *SIAM Review*, vol. 38, no. 1, pp. 174–175, mar 1996. [Online]. Available: http://epubs.siam.org/doi/10.1137/1038028

[8] J. C. Dunn, "A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters," *J. Cybern.*, 1973.

[9] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*. Boston, MA, USA: Springer US, 1981. [Online]. Available: http://link.springer.com/10.1007/978-1-4757-0450-1

[10] ——, "Cluster validity with fuzzy sets," *J. Cybern.*, vol. 3, no. 3, pp. 58–73, 1973.

[11] N. R. Pal and J. C. Bezdek, "On Cluster Validity for the Fuzzy c-Means Model," *IEEE Trans. Fuzzy Syst.*, vol. 3, no. 3, pp. 370–379, 1995.

[12] J. F. MCCARTHY, K. A. MARX, P. E. HOFFMAN, A. G. GEE, P. O'NEIL, M. L. UJWAL, and J. HOTCHKISS, "Applications of Machine Learning and High-Dimensional Visualization in Cancer Detection, Diagnosis, and Management," *Annals of the New York Academy of Sciences*, vol. 1020, no. 1, pp. 239–262, may 2004. [Online]. Available: http://doi.wiley.com/10.1196/annals.1310.020

[13] A. Inselberg and B. Dimsdale, "Parallel Coordinates for Visualizing Multi-Dimensional Geometry," in *Computer Graphics 1987*. Tokyo: Springer Japan, 1987, pp. 25–44. [Online]. Available: http://www.springerlink.com/index/10.1007/978-4-431-68057-4{_}3

[14] "Vue.js." [Online]. Available: https://vuejs.org/

[15] M. Bostock, V. Ogievetsky, and J. Heer, "D3 data-driven documents," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 2301–2309, Dec. 2011. [Online]. Available: http://dx.doi.org/10.1109/TVCG.2011.185

[16] "Chart.js." [Online]. Available: https://www.chartjs.org/

[17] A. De Rosa, "Creating Beautiful Charts with Chart.js," 2015. [Online]. Available: https://www.sitepoint.com/creating-beautiful-charts-chart-js/