Online ISSN 1848-3380 , Print ISSN 0005-1144
ATKAFF 52(4), 353–364(2011)

Sofija Pinjušić Ćurić, Mihaela Vranić, Damir Pintar

Improvement of Hierarchical Clustering Results by Refinement of Variable Types and Distance Measures

UDK 621.3.08:004.93
IFAC 3

Original scientific paper

Hierarchical clustering method is used to assign observations into clusters further connected to form a hierarchical structure. Observations in the same cluster are close together according to the predetermined distance measure, while observations belonging to different clusters are afar. This paper presents an implementation of specific distance measure used to calculate distances between observations which are described by a mixture of variable types. Data mining tool 'Orange' was used for implementation, testing, data processing and result visualization. Finally, a comparison was made between results obtained by using already available widget and the output of newly programmed widget which employs new variable types and new distance measure. The comparison was made on different well-known datasets.

Key words: Hierarchical clustering, Distance measure, Variable types, Dendrogram

Poboljšanje rezultata hijerarhijskog grupiranja podataka primjerenijim tretiranjem tipova podataka i prilagodbom mjere udaljenosti. Hijerarhijsko grupiranje se koristi za grupiranje objekata promatranja u grupe koje se dalje pripajaju te tako tvore hijerarhijsku strukturu. Prema odabranoj mjeri udaljenosti instance koje pripadaju istoj grupi su 'blizu' dok su instance koje pripadaju različitim grupama 'udaljenije'. U ovom članku prikazana je implementacija specifične mjere udaljenosti koja se koristi za izračun udaljenosti između instanci koje su opisane atributima različitih tipova podataka. Alat za dubinsku analizu podataka 'Orange' je odabran za implementaciju, testiranje, obradu podataka te vizualizaciju rezultata. Članak uz opis specifikacije novih varijabli te mjere udaljenosti također daje usporedbu rezultata dobivenih otprije poznatim modulom i novim modulom koji koristi nove tipove podataka i novu mjeru udaljenosti. Usporedba je napravljena nad različitim poznatim skupovima podataka.

Ključne riječi: hijerarhijsko grupiranje, mjera udaljenosti, tipovi podataka, dendrogram

1 INTRODUCTION

In the last few decades computers and databases have become indispensable and inseparable parts of business organizations. With increase of computers' performance abilities and decrease of data storage costs, the amount of data collected during each business year grows exponentially. However, the amount of valuable information extracted from this data as well as scientific research dealing with ways of extracting such information do not quite follow the same growth rate. Large databases are formed to support various research activities and it is often important to have the right insight into collected research data so that new discoveries and advances can be made.

Interesting and valuable information on business and research data is sometimes very difficult to notice due to it being hidden in the vast amount of collected data. Data mining is often used to obtain this information. It can discover unexpected and hidden data patterns which can truly enrich the knowledge database and lead to new discoveries regarding given business and scientific area.

The hierarchical cluster analysis is an unsupervised learning method which results in easily interpretable graphical structures. These structures enable the analyst to notice closeness between objects instances or attributes describing them. Gained relations can be very helpful in knowledge acquisition. However, depending on the chosen definition of how 'closeness' is calculated, very different results can be gained.

To address this issue, we decided to use open source data mining tool 'Orange' [1] and incorporate ability to recognize various attribute types more accurately. By using specific distance measure we generate more informative graphical structures. For that purpose we implemented new module called *Mixed variable type widget* and new variable types used within it. New module resolved shortcomings of the existing *Example distance widget* and built-in variable types.

This paper is structured as follows: the problem was discussed in detail in section 2. The precise description of newly defined variable types is given in section 3. Section 4 presents distance measures implemented within the *Example distance widget*, comparison to newly implemented distance measure and the brief review of cluster linkage criteria. In section 5, datasets chosen for testing along with selection reasoning are presented.

The implementation of the *Mixed variable type widget* is described in section 6. This section also presents results of widget testing along with comparison of hierarchical clustering results obtained by both existing and newly programmed widget. Both testing datasets were researched and examined. Section 7 presents possible improvements and future research directions. Finally, section 8 concludes the paper.

2 PROBLEM DISCUSSION

Hierarchical clustering is defined as a process of grouping similar observations into classes on multiple levels and results in a hierarchical decomposition of the given dataset. These classes are called clusters. The goal is to accomplish high similarity between observations of the same cluster and low similarity between observations of different clusters. Observations are physical or abstract entities represented by data objects or tuples consisting of attribute values. Grouped tuples form a dataset. This dataset is further used to calculate a dissimilarity matrix which stores measured differences for each pair of data objects. The measured differences (or distances) $d(i, j)$ represent dissimilarity between objects i and j. This distance matrix may then be used to generate a dendrogram - a graphical tree-like structure which effectively visualizes a hierarchy of clusters found in the explored dataset.

Cluster hierarchy is based on minimal distances between each pair of clusters. A linkage criterion is a distance measure used to calculate distances between pairs of clusters by telling the clustering algorithm how to compare all or certain cluster members. The shape and the depth of a dendrogram are influenced by the number of data objects and the used distance measure. Leaves of a dendrogram are clusters that contain only one observation, while the root represents a cluster containing all observations. Dendrogram is a powerful visualization tool because it can be easily interpreted even by users who are not data mining experts.

Aforementioned 'Orange' tool is an open source tool based on a Python script library for machine learning and data mining. This tool's strength lies in its modularity. It is easy to design new 'widgets' and augment its default functionalities. It also offers powerful visualization options.

'Orange' can be used to generate a distance matrix by using a specific 'widget' called *Example distance widget*. Another widget, called *Hierarchical clustering widget* can generate and visualize a dendrogram structure. Widgets are essentially basic elements of 'Orange Canvas' that represent GUI objects used to handle user input and visualize results.

The *Example distance widget* allows user to select one of several built-in distance measures which is then used to calculate the distance matrix for the input example table that contains data objects. This widget works very well if expected input is given, meaning that example variable types are either set by data analyst or automatically recognized by 'Orange' as continuous or discrete. This means that if variable types are not set by data analyst, 'Orange' will recognize their variables types automatically and assign them one of two predefined types. However, 'Orange' will only consider variable values but not the meaning behind them. In this scenario, data analyst cannot influence type assignment in any way. By default, the *Example distance widget* ignores variables whose type was not automatically recognized.

Problems arise when the target dataset cannot be described using only these two built-in variable types and/or when user wants to refine the type description for each variable. Being restricted to the two built-in variable types can be very limiting in variable description since discrete variables are interpreted only as strings and continuous only as numerical data. Using this approach, ordinal variables for example can only be described as discrete, even though by doing this their basic characteristic - order of values - is being completely ignored. Binary variables can also be described only as discrete and are compared on the basis of string similarity which is not always a good approach for this particular type of variables. Discrete variable types always being interpreted as strings will also affect the results of distance measuring in an unfavorable way when actual data type of contained values is numerical. All these issues lead us to believe that there is space for improvement, and a decision to implement a new software solution which will address those issues was made. Some approaches regarding this problem matter and theoretical foundations on similarity measures of objects described by attributes belonging to different variable types can be found in [2].

As mentioned previously, two basic attribute types defined in 'Orange' are *discrete* and *continuous*. Only these two types can be used by built-in learning methods. Attributes of discrete type are denoted with 'd' or '$discrete$'. Alternative is to list all possible values of the attribute and separate them by spaces. The listing of attributes is useful since it prescribes the order of the values that differs from default, which is the same as encountered when reading the examples. The data type of discrete attributes is enumerated so it can contain both numerical and categorical data. Continuous attributes are denoted with 'c' or '$continuous$'. The associated data type is $float$, which means it can contain only numerical data. These attribute types are most commonly assigned by data analyst. No mistakes are tolerated (in assigning, formatting or spelling). Although these attribute types are powerful and can describe a large number of attributes, in some cases it is necessary to refine data description so that some specific data characteristics could be considered.

For resolving mentioned problems a new module: *Mixed variable type widget* was implemented. The main and most important difference between this module and the previously mentioned *Example distance widget* is that it recognizes new variable types, efficiently resolving shortcomings of the default widget and built-in variable types.

To compare outputs of hierarchical clustering produced by *Example distance widget* and *Mixed variable type widget* (in further text abbreviated to *MVT widget*) two representative well known datasets of different characteristics were used.

3 INTRODUCTION OF NEW VARIABLE TYPES

The *MVT widget* identifies several variable types which are presented in this section. These variable types are listed and assigned to corresponding variable name in a specific description file used by the *MVT widget*. Newly defined variable types are presented below.

Interval-scaled variable type. Interval-scaled variables are continuous measurements of a roughly linear scale. Typical examples are 'weight' and 'height'. This type can be compared to the *continuous* 'Orange' variable type. The matching key word listed in the Description file for this variable type is *'intervalScaled'*. In most applications this measure should be standardized so that measurement units and outliers don't affect the clustering analysis. Afterwards dissimilarity between data objects is computed. *Numerical* is the only data type which is supported.

Symmetric binary variable type. By definition, binary variable has only two states: '0' or '1', where '0' suggests the absence of observed feature and '1' suggests its presence. A binary variable is symmetric if both states are equally valuable and carry the same weight. A typical example is gender which can be either 'male' or 'female'. In this example there is no preference over which certain value should be coded as '0' or '1'.

Treating these variables as interval-scaled can result in misleading clustering results so usage of specific approach is required. Only symmetric binary variables are considered because they are most commonly present in datasets. *'SymmetricBinary'* is the matching key word assigned to this variable type. This variable type supports both numerical and categorical values in an input dataset.

Nominal variable type. Nominal variable type can be considered as a generalization of the binary variable type because it can have more than two states. The order of these states isn't important and is not considered while calculating data object dissimilarity. This type can be compared to the discrete 'Orange' variable type. The matching key word listed in the description file is *'nominal'*. Both numerical and categorical data types are supported.

Ordinal variable type. Discrete ordinal variable type is similar to nominal type. Unlike nominal states, ordinal states are ordered in a meaningful sequence which often suggests subjective assessments of qualities that cannot be measured objectively. Continuous ordinal variable can be interpreted as a set of continuous data where relative ordering of the values is crucial, not their actual magnitude. Values of ordinal variables need to be mapped to ranks - r_f. M ordered states of ordinal variable f define the ranking - $r_f = \{1, \ldots, M_f\}$. Any f variable value is then replaced by corresponding rank r_{if}. This approach stems from the assumption that different adjacent variables are equally distant. Different ordinal variables can vary in number of possible ordinal states so range of each variable has to be mapped to $[0, 1]$ interval. With this standardization mapped ranks z_{if} are computed. In this way all variables are of equal weight. While computing data object distances, mapped ranks z_{if} are handled as interval-scaled variables. In the description file no corresponding keyword is defined for ordinal variable type. Ordered states are listed as a comma delimited string instead. Also, this variable type supports both numerical and categorical data types.

These four new variable types are defined because of their common usage in various datasets and are important for discovering natural groupings of observations. Other variable types can also be implemented. This is the subject of our further work.

The goal of research presented in this paper is to calculate distance matrix using most elaborate information available for each data object. Using as much information as possible will result in more informative dendrograms. A prerequisite for this approach is having the exact knowledge of meaning behind each variable in explored dataset but not necessarily of variable interrelationships.

4 DISTANCE MEASURES AND CLUSTER LINKAGE

Distance measures define distances or dissimilarities between observations. More similar observations have shorter distance and more diverse ones have greater distance [3]. Distance measures, i.e. metrics are essential for the distance matrix computation. A distance matrix is an object-to-object structure which stores a collection of proximities for all pairs of n observations [4]. It is often represented by an n-by-n table where each element $d(i, j)$ is the measured distance between objects i and j. This is a symmetrical matrix since $d(i, j) = d(j, i)$. Also, distance between each object and itself is zero: $d(i, i) = 0$.

The built-in *Example distance widget* offers the choice between three distance measures. Those are Euclidean, Manhattan and Hamming distance. Also, Pearson and Spearman rank correlations are supported but will not be considered because their purpose is to measure the quality of clustering. These distance metrics give proper results only when objects described by one variable type are presented.

By formal definition, Euclidean and Manhattan distances are derivations of Minkowski distance metrics [5]. For two n-dimensional data objects the Minkowski distance is defined as

$$d_p(a, b) = \sqrt[p]{\sum_{i=1}^{n} |a_i - b_i|^p} \qquad (1)$$

where p is a positive integer. Euclidean and Manhattan (or 'city block') distances are special cases of Minkowski distance defined by setting $p = 2$ for the former or $p = 1$ for the latter distance measure. These two distance measures are well-known and commonly used metrics. They are mostly used to calculate distances between observations described by interval-scaled, i.e. continuous variables.

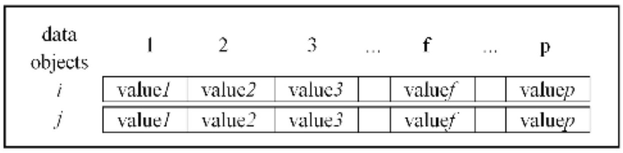
The Hamming distance is suitable only if all variables that describe data object are of discrete type. It is used for calculating string equality in a way that if strings are completely equal distance is set to zero, otherwise it is set to one. In this way it counts different attribute values of two data objects. It does not calculate standard Hamming distance between string representations of each variable value pair which is defined as the number of positions at which corresponding symbols of two equally long strings are different.

Besides different formal definitions, these measures also differ in how correctly they treat unknown values. The Manhattan and the Hamming distance do not excel in this respect: when computing data object distances, if any of the two values is missing, the corresponding distance is set to 0.5 on a normalized scale where the largest difference in attribute values is 1.0. The most correct treatment of unknown values is done by the implemented preprocessing of Euclidean metrics. Here, the probability distributions of discrete variables are used, while for continuous variables the tool uses the expected distance assuming the Gaussian distribution of attribute values (distribution's parameters are assessed from the dataset).

When calculating distances, the *Example distance widget* uses standardized values if variable type is continuous. When discrete variable types are present, the widget calculates equality of their string representations as mentioned when discussing Hamming distance.

The *MVT widget* considers all four new variable types and calculates data object distances taking all of them into account. Therefore, this widget is appropriate for calculation of dissimilarity between data objects described by mixed variable types. One method of computing dissimilarity between data objects is grouping corresponding kinds of variable types together and performing a separate cluster analysis for each type. Downside of this approach is that separate analyses often lead to disagreeable results. It is better to process all variable types together. In this way only one cluster analysis is performed and only one dendrogram is generated. One such technique described in [6] and implemented within the *MVT widget* considers different variable types while calculating distance matrix and brings all distance values onto a common scale of the interval $[0, 1]$.

Suppose the dataset is defined by p variables of different variable types and that two data object i and j are presented as shown in Fig. 1.



data objects	1	2	3	...	f	...	p
i	value*1*	value*2*	value*3*		value*f*		value*p*
j	value*1*	value*2*	value*3*		value*f*		value*p*

Fig. 1. Data objects i and j

Position of each variable describing data objects is denoted by f and ranges from 1 to p. The dissimilarity $d(i, j)$ between i and j across the entire variable set f is defined as

$$d(i, j) = \frac{\sum_{f=1}^{p} \delta_{ij}^{(f)} d_{ij}^{(f)}}{\sum_{f=1}^{p} \delta_{ij}^{(f)}} \qquad (2)$$

where the indicator $\delta_{ij}^{(f)} = 0$ if either one of variable values on position f is missing, otherwise $\delta_{ij}^{(f)} = 1$. It indicates that missing values are not being considered in calculation. The contribution of variable f to the distance between i and j, $d_{ij}^{(f)}$, is computed based on its type:

- If f is symmetric binary or nominal variable: $d_{ij}^{(f)} = 0$ if values x_{if} and x_{jf} are equal, otherwise

$d_{ij}^{(f)} = 1$. While computing dissimilarity, it is expected that variables with equal values will be ignored. Typical example of symmetric binary variable is 'gender' and of nominal is 'eye color', as shown in an example of four data objects presented in Fig. 2.

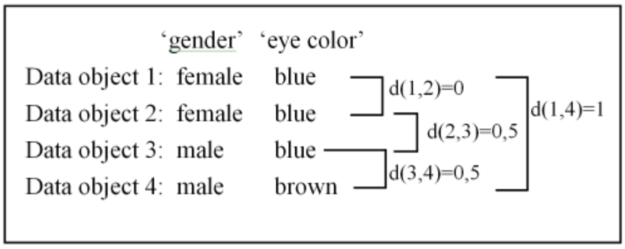


Fig. 2. Example: data object described with binary and nominal variables

Distance between each pair of data objects calculated using previously described metrics is shown to the right of brackets connecting those data objects. If two compared variable values are equal, distance between them is zero ($d_{2,3}^{(eye\ color)} = 0$) because their dissimilarity is the smallest. If the two compared variable values are different, the distance between them is one ($d_{2,3}^{(gender)} = 1$) because their dissimilarity is the greatest. If data objects are described only by symmetric binary and nominal variables, distance between those data objects is equal to the calculated average of distances between corresponding variables as in (2) (for example $d(2,3) = 0.5$). The greater the number of different variable value pairs, greater is the distance between data objects and vice versa.

- If f is an interval-scaled variable:

$$d_{ij}^{(f)} = \frac{|x_{if} - x_{jf}|}{max_h x_{hf} - min_h x_{hf}} \tag{3}$$

where h runs over all non-missing objects for variable f. This formula is used to standardize variable values by fitting them into $[0, 1]$ interval. In this way outliers can be easily detected. Typical example of interval-scaled variable is 'age'. Suppose three data objects are analyzed as presented in Fig. 3.

These examples show that closer the two compared variables are to opposite extremes (corresponding minimal and maximal values) distance between them will more closely approximate to a distance value of 1. Similarly, if the variable values are close together, distance between them will tend to a distance value of 0. Here it must be stated that the existence of outliers can significantly skew the values of distance measures if they are not dealt accordingly beforehand.

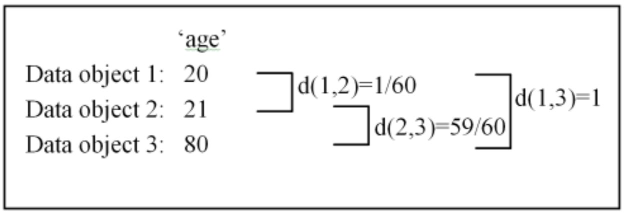


Fig. 3. Example: data object described with interval-scaled variable

- If f is ordinal: compute the ranks r_{if} and mapped ranks $z_{if} = \frac{r_{if}-1}{M_f-1}$. Mapped ranks represent standardized values and are treated as interval-scaled variables. Example of ordinal variable is shown in Fig. 4 in which variable 'class' has three states ordered as follows: c, s, q. Ranking range for this example is from one to three. Three data objects are taken into account and distances are calculated based on objects' mapped ranks. Maximal value of distance measure is one, and minimal is zero.

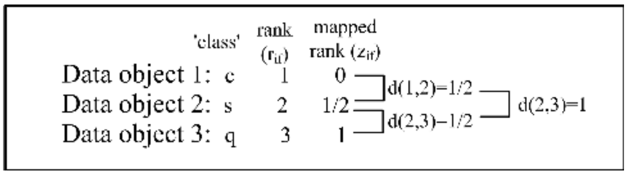


Fig. 4. Example: data object described with ordinal variable

Closer the rank and mapped rank values describing data objects are, smaller the distances between data objects get (i.e. closer to zero). Similarly, further apart compared values of mapped ranks are, distances between data objects grow larger (i.e. closer to one).

Usage of (2) enables computation of the distance matrix even when the variables describing researched dataset are of different types. For each variable type a specific metrics is used since each is treated in a specific way. Explained metrics were implemented because they correspond to those most commonly used for these variable types. Used metrics can also be altered so they reflect particular wishes and demands of data analyst. With their alternation, cluster analysis would provide different results more closely conforming to the analyst's preferences.

The calculated distance matrix is used to generate a cluster hierarchy that can be described as a cluster agglomeration tree which begins with single data object clusters merged together into larger ones, and ends with a cluster that holds all data objects. Clusters are merged together on the basis of linkage criterion which computes the distance between pair of clusters as a function of distances between members of compared clusters. There are three basic types of linkage criterion:

- single or minimum: $min\{d(a,b): a \in A, b \in B\}$
- complete or maximum:$max\{d(a,b): a \in A, b \in B\}$
- average or mean: $\frac{1}{|A||B|}\sum_{a\in A}\sum_{b\in B} d(a,b)$

where d is the chosen metrics and A, B represent a cluster pair.

In 'Orange' default linkage criterion is "average". User can set different linkage criteria through a *Hierarchical clustering widget*. The chosen linkage criterion is then applied to the distance matrix and a dendrogram is generated.

5 TESTING DATASETS

Testing in this paper was done using two datasets. Those are 'Titanic' [7] and 'Automobile' dataset [8]. These datasets are well-known and commonly used. Their characteristics are quite different, which - coupled with their availability - makes them representative for testing. They differ in number of observations, number of attributes, characteristics of objects that they describe, etc. Most importantly, they differ in variable types. This section brings detailed description of each dataset.

The first dataset used in this paper is the 'Titanic' dataset. It contains information regarding individual passengers that boarded the famous ship and describes their survival status. This dataset contains 1309 examples which are described using 14 attributes. Table 1 presents dataset variables taken into account for research described in this paper.

Table 2 presents variable types assigned to variables in 'Titanic' dataset. Column to the right of the 'Variable name' column is 'Orange type' column, which contains variable types that 'Orange' automatically assigned to corresponding variable. Next column is 'Refined type' where newly defined and more detail variable types are listed. Those were assigned by data analyst. The last column reveals that the class variable is variable 'survived'. This dataset is very convenient for our testing since it contains all newly defined variable types.

The second dataset explored in this paper is 'Automobile' or 'Car import' dataset. It contains 205 examples which are described using 26 attributes. Table 3 presents dataset variables and their characteristics.

This dataset consists of three categories of variables: (a) the specification of an automobile in terms of various characteristics, (b) its assigned insurance risk rating and, (c) its normalized losses in use as compared to the other cars [8].

Table 1. 'Titanic' data set variables

Variable name	Description	Units	Levels	Data type
pclass	Travelling Class	-	3	integer
name. surname	Name and surname of passenger	-	-	character
gender	Gender of passenger	-	2	integer
age	Age of passenger	year	-	double
sibsp	Sibl./spouses aboard	-	-	double
parch	Parents/ children aboard	-	-	double
ticket	Ticket no.	-	-	character
fare	Passenger fare	British pound	-	double
cabin	Cabin no.	-	187	integer
embarked	Port of embarkation	-	3	integer
boat	Lifeboat	-	28	integer
body	Body identification no.	-	-	double
home. dest	Home/ destination	-	-	character
survived	Survival flag	-	-	double

Table 2. 'Titanic' dataset variable types

Variable name	'Orange' type	Refined type	Class var
pclass	discrete	ordinal (1,2,3)	No
nam. sur.	discrete	ignore	No
gender	discrete	symmetricBinary	No
age	continuous	intervalScaled	No
sibsp	discrete	intervalScaled	No
parch	discrete	intervalScaled	No
ticket	discrete	nominal	No
fare	continuous	intervalScaled	No
cabin	discrete	nominal	No
embarked	discrete	ordinal (C,Q,S)	No
boat	discrete	nominal	No
body	discrete	nominal	No
home.dest	discrete	ignore	No
survived	discrete	classVar	Yes

The second category corresponds to the degree to which the auto is more risky than its price indicates.

All assigned variable types are listed in Table 4. This table is analog to Table 2 because it presents variable types

Table 3. 'Automobile' dataset variables

Variable name	**Description**	**Levels**	**Data type**
symboling	risk factor	7	integer
normalized-losses	loss of payment per insured vehicle year	-	integer
make	make of car	22	character
fuel-type	type of fuel used	2	character
aspiration	engine pressure source	2	character
num-of-doors	number of doors	2	character
body-style	car body styler	5	character
drive-wheels	driving force wheel	3	character
engine-location	location of engine	2	character
wheel-base	distance between centers of wheels	-	double
length	length of car	-	double
width	width of car	-	double
height	height of car	-	double
curb-weight	total weight of car	-	integer
engine-type	type of engine	7	character
num-of-cylinders	number of cylinders in engine	7	character
engine-size	size of engine	-	integer
fuel-system	type of fuel system	8	character
bore	diameter of a cylinder in a piston engine	-	double
stroke	reciprocating motion used in rec. engines	-	double
compression-ratio	ratio of the volume of engine combustion chamber	-	integer
horsepower	engine horsepower	-	integer
peak-rpm	maximal frequency of rotation	-	integer
city-mpg	miles per gallon during city drive	-	integer
highway-mpg	miles per gallon on highway	-	integer
price	price of car	-	integer

assigned to variables in this dataset by 'Orange' software and by data analyst. For each continuous 'Orange' variable type, refined data type is set to interval-scaled and for each discrete 'Orange' variable type, refined data type is set to nominal. Other variable type assignments could be made with the help of automobile domain expert.

Table 4. 'Automobile' dataset variable types

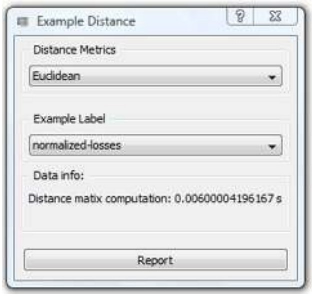
Variable name	**'Orange' type**	**Refined type**	**Class var**
symboling	continuous	classVar	Yes
normalized-losses	continuous	intervalScaled	No
make	discrete	nominal	No
fuel-type	discrete	nominal	No
aspiration	discrete	nominal	No
num-of-doors	discrete	nominal	No
body-style	discrete	nominal	No
drive-wheels	discrete	nominal	No
engine-location	discrete	nominal	No
wheel-base	continuous	intervalScaled	No
length	continuous	intervalScaled	No
width	continuous	intervalScaled	No
height	continuous	intervalScaled	No
curb-weight	continuous	intervalScaled	No
engine-type	discrete	nominal	No
num-of-cylinders	discrete	nominal	No
engine-size	continuous	intervalScaled	No
fuel-system	discrete	nominal	No
bore	continuous	intervalScaled	No
stroke	continuous	intervalScaled	No
compression-ratio	continuous	intervalScaled	No
horsepower	continuous	intervalScaled	No
peak-rpm	continuous	intervalScaled	No
city-mpg	continuous	intervalScaled	No
highway-mpg	continuous	intervalScaled	No
price	continuous	intervalScaled	No

6 IMPLEMENTATION AND RESULT DISCUSSION

The newly programmed *MVT widget* uses a tab-delimited file as input data format where values listed in each line are separated with tabulators and form a table. It is very simple, and is also Orange's native data format. The first line of .tab file lists names of variables and the name of class variable. The original 'Orange' *Example distance widget* which calculates a distance matrix expects for the second and the third line to contain some special attribute information. The second line gives types of attributes, one entry for each attribute. The third line contains optional flags that add an additional description to every column. The rest of the table presents the actual data.

The purpose of *MVT widget* is to calculate a distance matrix for a dataset when variables describing the dataset are of different types. The main goal is enabling data analyst to assign newly defined variable types to variables describing researched dataset. In this way, the dataset is described more closely, so that the resulting dendrogram reflects the view of dataset defined by data analyst.

Writing widgets is fairly easy using the provided environment which offers many built-in GUI related functionalities and leaves only the widget logic to the programmer. *Example distance widget* has very simple GUI which allows user to set distance metrics and define which label will be set (Fig. 5). The original *Example distance widget* doesn't display distance matrix computation time, but for testing purposes the original code was altered.



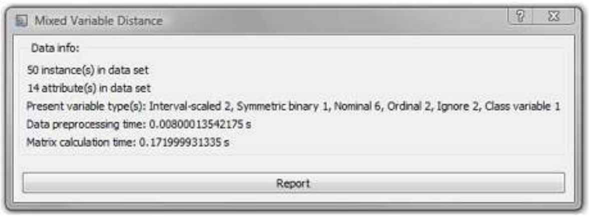
Distance Metrics
Eudidean

Example Label
normalized-losses

Data info:
Distance matix computation: 0.00600004196167 s

Fig. 5. Example distance widget GUI

GUI of the *MVT widget* is also very simple since its purpose is to calculate a distance matrix and not to handle user input. It consists of one widget box element which holds labels that display number of instances in a dataset, number of variables, number of present variable types and time elapsed during data preprocessing and distance matrix calculation (Fig. 6). The report button gives some additional information regarding input dataset.



Data info:
50 instance(s) in data set
14 attribute(s) in data set
Present variable type(s): Interval-scaled 2, Symmetric binary 1, Nominal 6, Ordinal 2, Ignore 2, Class variable 1
Data preprocessing time: 0.00800013542175 s
Matrix calculation time: 0.171999931335 s

Fig. 6. MVT widget GUI

Widgets communicate using input and output signals that are connected to input and output channels. The *Example distance widget* has only one input channel which inputs .txt file containing example table. This implies that no variable types were defined and that only list of variable names and data objects are given. In this approach it is left for 'Orange' to assign variable type for each variable in the example table. On the other hand, the *MVT widget* has two input channels. The first inputs an example table in .txt format. The second channel inputs variable description table contained in a description file, which is also in .txt format. It consists of only two lines. The first line lists variable names while the second lists variable types. In this way user can define different variable descriptions for the same example table, easily swap them by enabling and disabling signals between widgets and observe influence of various variable descriptions on the calculated distance matrix and the resulting dendrogram.

Class variables are set in the description file for both explored datasets using 'classVar' keyword as listed in Table 2 and Table 4. For 'Titanic' dataset class variable is variable 'survived' and for 'Automobile' dataset variable 'symboling'. Those variables are ignored while calculating distance matrix and therefore don't affect the shape of the generated dendrogram.

After widget inputs are set, widgets calculate a distance matrix using chosen or defined distance measures. *MVT widget* uses newly defined distance measure for mixed variable types described in section 3, while distance measure for *Example distance widget* is set to Euclidean distance. Then the input example table needs to be preprocessed. For *MVT widget* this includes mapping the range of each ordinal variable onto $[0, 1]$ interval by replacing variable values with standardized ranks and initializing vectors that hold minimal and maximal values of interval-scaled variable types and mapped ranks of ordinal variable types. Within both widgets for each pair of data objects distance is calculated considering assigned variable types and whether variable value is missing or not. The results are written into the corresponding place in the distance matrix. The final result for each widget is a symmetric distance matrix which is forwarded to the output channel.

The *Hierarchical clustering widget* inputs a distance matrix and visualizes a dendrogram. It can be used to set one of three linkage criteria and to modify dendrogram visualization. This widget outputs examples that correspond to those contained in the cluster selected interactively by user. It is also responsible for dendrogram visualization and interactive exploration by data analyst. Data processing and widget interactions are presented in Fig. 7. For convenience during result comparison only one 'Orange' scheme was used for testing purposes. The *MVT widget* was tested on Intel®Core^{TM}2 Duo CPU processor with operating frequency of 2.00 GHz and 4.00 GB RAM memory.

For testing purposes the 'Titanic' dataset was divided into 14 subsets as presented in Table 5. For this purpose *Data Sampler widget* was used. Only number of examples

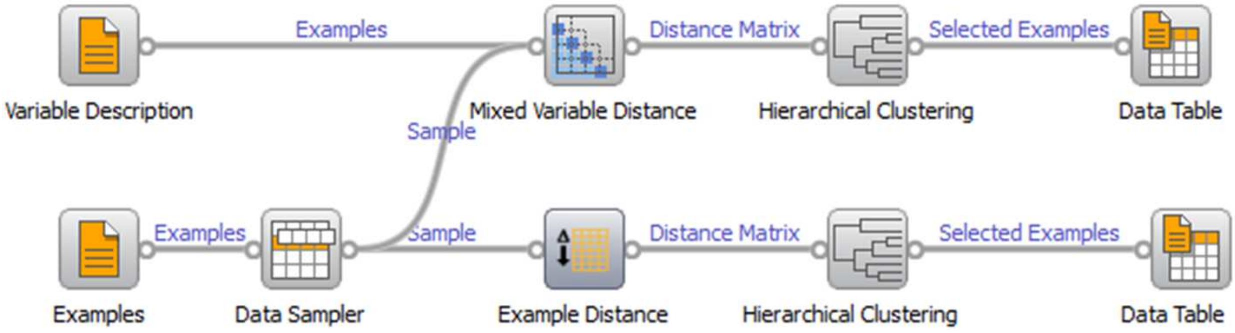


Fig. 7. Hierarchical clustering 'Orange' scheme

varied while number of attributes and variable type assignments remained the same. As expected, data preprocessing is much less time-consuming than calculation of distance matrix. Time requirements grew with the increase in number of analyzed observations. The calculation time for only 50 examples equals 0.172 seconds. The calculation time for the whole dataset, which holds 1309 examples, equals 111.568 seconds. Very useful functionality that 'Orange' offers is storing distance matrix as textual file and reloading it. In this way a distance matrix can be calculated only once and used repeatedly. Table 5 also presents distance matrix calculation time for the *Example distance widget*. Data preprocessing for this widget isn't presented because it doesn't preprocess the given dataset. While comparing distance matrix calculation time of *Example distance widget* and MVT one, it can be concluded that the latter one has higher processing demands. Substantial time required by MVT for distance matrix calculation is a direct consequence of widget logic implementation that relies on string comparison.

Table 5. 'Titanic' testing results

Number of examples	**MVT widget**		**Example distance widget**
	Data preprocessing [s]	Distance matrix calculation[s]	Distance matrix calculation[s]
50	0.008	0.172	0.011
100	0.017	0.698	0.046
200	0.035	2.707	0.176
300	0.051	6.234	0.396
400	0.065	11.076	0.699
500	0.082	17.186	1.099
600	0.099	24.956	1.582
700	0.114	33.113	2.672
800	0.131	43.511	2.812
900	0.147	53.790	3.573
1000	0.161	65.396	4.391
1100	0.177	79.469	5.609
1200	0.196	93.662	6.311
1309	0.212	111.568	7.516

Figure 8 presents a dendrogram based on the distance matrix produced by the Example distance widget. The dataset used is 'Titanic', the distance measure used is default Euclidean distance and the linkage criterion is average. Resulting dendrogram is very deep and branched. Also, clusters are merged by adding subsequent examples one by one, so final result is not very informative.

Figure 9 presents a dendrogram based on the distance matrix produced by the MVT widget. Again, the same 'Titanic' dataset is used and the linkage criterion is average. Presented dendrogram is considerably shorter and much less branched. This is a great advantage of the MVT widget since shorter and less branched hierarchies are preferred in cluster analysis. After the first level clusters are agglomerated by merging clusters that hold at least two data objects.

Dendrogram on Fig. 9 is very different from the one in Fig. 8. Here, first cluster holds objects 8 and 42, while in Fig. 8 they are agglomerated much later. Actual data objects contained in single object clusters 8 and 42 are presented in Fig. 10.

Considering metrics definition and assigned data types, it is very easy to see why these specific objects were joined together. Here, the first cluster consists of two female survivors of similar age belonging to first class with relatives on board and traveling in the same cabin. Further cluster analysis gives deeper insight. For example, most survivors are female passengers who traveled in the first and the second class. Male passengers who traveled in the third class have a very low survival rate. Also, most male passengers traveled alone while women mainly traveled with relatives.

'Automobile' dataset testing was done in a way very similar to 'Titanic' dataset testing. The dataset was divided into 15 subsets presented in Table 6. This table presents time requirements very similar to those presented

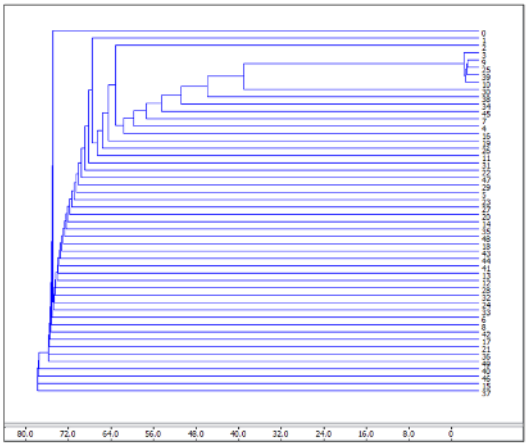


Fig. 8. 'Titanic' dendrogram based on a distance matrix produced by the Example distance widget

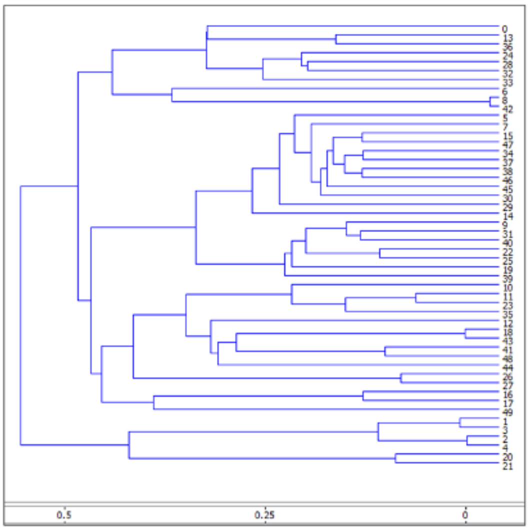
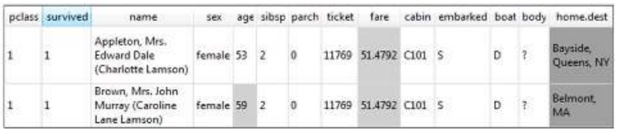


Fig. 9. 'Titanic' dendrogram based on a distance matrix produced by the MVT widget



pclass	survived	name	sex	age	sibsp	parch	ticket	fare	cabin	embarked	boat	body	home.dest
1	1	Appleton, Mrs. Edward Dale (Charlotte Lamson)	female	53	2	0	11769	51.4792	C101	S	D	?	Bayside, Queens, NY
1	1	Brown, Mrs. John Murray (Caroline Lane Lamson)	female	59	2	0	11769	51.4792	C101	S	D	?	Belmont, MA

Fig. 10. 'Titanic' data objects 8 and 42

in Table 5. Once again, data preprocessing is much less time-consuming than calculation of distance matrix. Also, time requirements grew with the increase in number of analyzed observations. The distance matrix calculation time required by *MVT widget* for only 10 data objects equals 0.049 seconds. The distance matrix calculation time for the whole dataset, which holds 205 data objects, equals 21.228 seconds. The distance matrix calculation time required by *Example distance widget* is considerably shorter than the one required by *MVT widget*. As expected, *MVT widget* has higher processing demands because of its inner logic.

Table 6. 'Car import' testing results

Number of examples	**MVT widget**		**Example distance widget**
	Data pre-processing [s]	Distance matrix calculation[s]	Distance matrix calculation[s]
10	0.003	0.049	0.001
20	0.006	0.199	0.002
30	0.009	0.448	0.005
40	0.012	0.804	0.008
50	0.015	1.25	0.012
60	0.018	1.813	0.019
70	0.021	2.438	0.026
80	0.024	3.247	0.032
90	0.027	4.161	0.040
100	0.031	5.083	0.050
125	0.031	7.942	0.077
150	0.046	11.331	0.112
175	0.053	15.435	0.152
200	0.059	19.923	0.195
205	0.061	21.228	0.243

Figure 11 presents a dendrogram based on the distance matrix produced by the *Example distance widget*. The dataset used is 'Car import', the distance measure used is default Euclidean distance and the linkage criterion is average. Resulting dendrogram is not informative and is quite incomprehensible. In this case used widget doesn't differentiate data objects well and this is why the majority of data objects are recognized as very similar. This similarity can't be accurate since the *Data Sampler widget* samples the input dataset in a random way.

Figure 12 presents a dendrogram based on the distance matrix produced by the *MVT widget*. Again, the same 'Car import' dataset is used and the linkage criterion is average. This dendrogram is much more informative since *MVT widget* differentiates samples data objects very well. Additionally, it is not very deep and is acceptably branched. This is another advantage of *MVT widget* since good data object differentiation leads to more informative dendrograms.

Considering the shape of the dendrogram the one produced by *MVT widget* is considerably different from the one produced by *Example distance widget*. Still, certain similarities in data object grouping can be noticed since on some levels both widgets recognize the same data objects

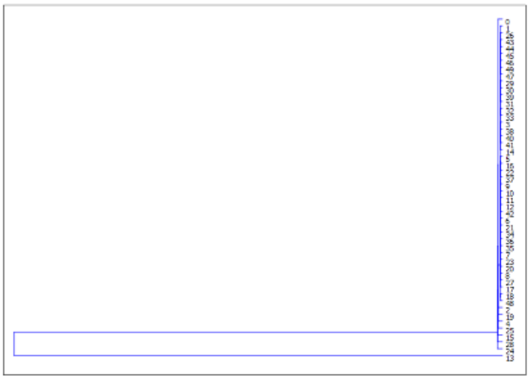


Fig. 11. ‘Car import’ dendrogram based on a distance matrix produced by the Example distance widget

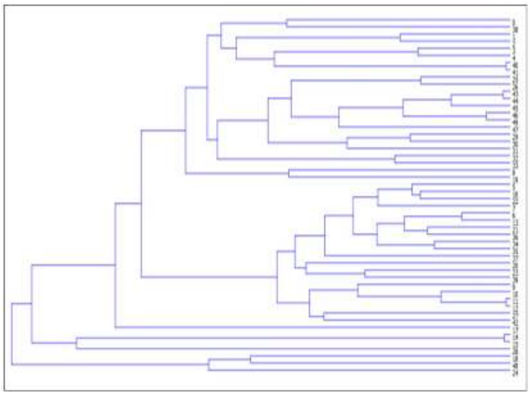
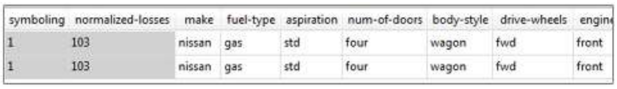


Fig. 12. ‘Car import’ dendrogram based on a distance matrix produced by the MVT widget

or clusters to be the most similar ones, but the calculated data object or cluster dissimilarity is quite different resulting in different dendrograms. For example, both widgets have recognized objects 20 and 22 as most similar ones and they form first level cluster. Actual data for few variables of data objects 20 and 22 is presented in Fig. 13.



symboling	normalized-losses	make	fuel-type	aspiration	num-of-doors	body-style	drive-wheels	engin
1	103	nissan	gas	std	four	wagon	fwd	front
1	103	nissan	gas	std	four	wagon	fwd	front

Fig. 13. ‘Car import’ data objects 20 and 22

Viewing data presented in Fig. 13 conclusion can be made that both widgets perform well while searching for the most similar data objects because single object clusters 20 and 22 hold almost identical data. With further analysis of the dendrogram presented in Fig. 11 other conclusions considering ‘Car import’ dataset were made. For example, on the third clustering level grouped data objects contain information regarding cars that are of the same make, fuel type, aspiration, engine location, etc. Also, for these cars initially assigned risk factor is the same. But those cars differentiate in price and curb weight, and in some cases price difference is considerable. This means that similar car model of the same make are not always in the same price category although they equal in some general car characteristics. Level of passenger safety that they offer is the same despite the price difference. Cars of different make that are in the same risk category can have quite different prices. Also, some Toyota, Mitsubishi and Volkswagen models are considered to be very risky while generally Volvo models are considered to be the safest. The safest cars are also the most expensive ones.

Class variables have been of great help during result interpretation while they show how cluster members are divided between classes and indicate the similarity between separate classes. In conclusion, it was shown that *MVT widget* is able to give more informative and more comprehensible dendrograms which offer data analyst specified view of the input dataset with the help of newly implemented metrics and defined data types.

7 FUTURE WORK

In the future, new variable types (such as ratio-scaled) will be considered. The most promising improvements address widgets GUI which handles user input so the user will be able to set variable types and adjust distance measures interactively. Also, programming code could be further optimized. Widget could compare actual variable types and not only their string representations. The duration of distance matrix computation could be considerably shorter if it was done in parallel on multiprocessor systems.

Missing values could be handled more correctly if they were calculated using probability distributions of variables, which requires further investigation.

Outlier recognition and their influence on results will be further researched. In data preprocessing outliers could be simply put into one special cluster which is only merged with others at the root. In this way the outliers do not have to be extracted from the dataset before dendrogram generation because this separate cluster doesn’t have a great influence on the final result.

8 CONCLUSION

In this paper we researched the problem of using hierarchical clustering method when analyzing objects described with variables of different data types. The focus in our research was on clustering method improvement and variable description refinement. Open source tool ‘Orange’ offers only two variable types for data description and uses few implemented metrics. In order to consider features of each variable as precisely as possible, introduction and definition of new variable types were made along with implementation of new metrics. The final distance measure uses all data available for each object and combines newly defined metrics in a specific way to gain a single distance measure for objects described by mixture of variable types.

Our results show that if the exact knowledge of meaning behind each variable in explored dataset can be obtained, much better and more informative dendrograms could be generated. Testing results also show that the metrics which considers many different variable types differentiates data objects in a more efficient way than the metrics which considers only two variable types. However, the human factor is of crucial importance. Analyst must be aware of the data and of exact metrics within the *MVT widget* in order to get results that fully satisfy his needs.

Implemented variable types and measure of dissimilarity used on the well known 'Titanic' and 'Car import' datasets enabled us to explore the available data in much better way than it was previously possible by the built-in *Example distance widget*.

REFERENCES

[1] J. Demšar, B. Zupan, G. Leban, and T. Curk, "Orange: from experimental machine learning to interactive data mining," in *Proceedings of the 8th European Conference on Principles and Practice of Knowledge Discovery in Databases*, PKDD '04, (New York, NY, USA), pp. 537–539, Springer-Verlag New York, Inc., 2004.

[2] J. C. Gower, "A general coefficient of similarity and some of its properties," *Biometrics*, vol. 27, no. 4, pp. 857–871, 1971.

[3] J. Han and M. Kamber, *Data mining: concepts and techniques*. The Morgan Kaufmann series in data management systems, Elsevier, 2006.

[4] E. Alpaydin, *Introduction to Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2004.

[5] P. Javor, *Mathematical Analysis 2*. Applied Mathematics, Element, 2 ed., 2002.

[6] L. Kaufman and P. J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley, 1990.

[7] F. Harrel, "Titanic data." `http://biostat.mc.vanderbilt.edu/wiki/pub/Main/DataSets/titanic.html`, 2009.

[8] J. Schlimmer, "Automobile data set." `http://archive.ics.uci.edu/ml/datasets/Automobile`, 2010.

Sofija Pinjušić Ćurić received her B.Sc and M.Sc degree in Computer Science from the Faculty of Electrical Engineering and Computing University of Zagreb in 2008 and 2010, respectively. She is currently working on her Ph.D. and is employed with Private School of Economics and Computing, Zagreb. Her research interests include databases, data warehouses, business intelligence, mobile learning and e-learning system design.

Mihaela Vranić received her B.Sc. and M.Sc. degree in Electrotechnics, and Ph.D. degree in Computer Science from the Faculty of Electrical Engineering and Computing, University of Zagreb in 2001, 2007 and 2011, respectively. She also received B.Sc. degree in Economics from the Faculty of Economics, University of Zagreb in 2002. Since 2001 she has been affiliated with Faculty of Electrical Engineering and Computing as a research assistant at the Department of Fundamentals of Electrical Engineering and Measurements. Her research interests include e-business, databases, data warehousing, business intelligence and data mining.

Damir Pintar received his B.Sc., M.Sc. and Ph.D. degree in Electrotechnics from the Faculty of Electrical Engineering and Computing, University of Zagreb in 2000, 2004 and 2009, respectively. Since 2000 he has been affiliated with Faculty of Electrical Engineering and Computing as a research assistant at the Department of Fundamentals of Electrical Engineering and Measurements. His research interests include e-business, web technologies, databases, data warehousing and business intelligence.

AUTHORS' ADDRESSES

Sofija Pinjušić Ćurić, M.Sc.
Private School of Economics and Computing
Budakova 1D, HR-10000, Zagreb, Croatia
email: sofija.pinjusic@skole.hr

Mihaela Vranić, Ph.D.
Damir Pintar, Ph.D.
Department of Fundamentals of Electrical Engineering and Measurements,
Faculty of Electrical Engineering and Computing, University of Zagreb,
Unska 3, HR-10000, Zagreb, Croatia
email: mihaela.vranic@fer.hr, damir.pintar@fer.hr

Received: 2010-12-07
Accepted: 2011-10-18