

Information Cartography in Association Rule Mining

Iztok Fister Jr., *Member, IEEE*, Iztok Fister, *Member, IEEE*

Abstract—Association Rule Mining is a machine learning method for discovering the interesting relations between the attributes in a huge transaction database. Typically, algorithms for Association Rule Mining generate a huge number of association rules, from which it is hard to extract structured knowledge and present this automatically in a form that would be suitable for the user. Recently, an information cartography has been proposed for creating structured summaries of information and visualizing with methodology called “metro maps”. This was applied to several problem domains, where pattern mining was necessary. The aim of this study is to develop a method for automatic creation of metro maps of information obtained by Association Rule Mining and, thus, spread its applicability to the other machine learning methods. Although the proposed method consists of multiple steps, its core presents metro map construction that is defined in the study as an optimization problem, which is solved using an evolutionary algorithm. Finally, this was applied to four well-known UCI Machine Learning datasets and one sport dataset. Visualizing the resulted metro maps not only justifies that this is a suitable tool for presenting structured knowledge hidden in data, but also that they can tell stories to users.

Index Terms—information cartography, metro map, Evolutionary Algorithms, Machine Learning, eXplainable Artificial Intelligence

1 INTRODUCTION

ASSOCIATION Rule Mining (ARM) is a Machine Learning (ML) method for discovering the interesting relations between attributes in a huge transaction database. Usually, algorithms for ARM generate a huge number of association rules collected in large datasets. Hence, we are confronted with the problem of how to extract structured knowledge from the large datasets and then present this knowledge automatically to the user in a suitable form. Nowadays, this problem is being solved with eXplainable Artificial Intelligence (XAI) methods that explain to users how decisions, usually affecting human lives, are made by deployed AI models in practice. While the models created by ML methods, like linear/logistic regression, decision tree, k-nearest neighbors, are transparent and, thus, easy to understand, the models built by methods, like tree ensembles, random forest, deep learning, are too complex and therefore act as a black-box to users [1]. Consequently, danger has arisen on creating and using decisions that are not justifiable, legitimate, or do not allow detailed explanations of their behavior [2].

To overcome this problem, the XAI helps users to understand, trust, and effectively manage the emerging generation of AI partners [3]. Such explainable systems already help users to understand the information generated from the ML models in medicine, transportation, security and economy, among others [1]. These systems use several explanation techniques, like: explanation by simplification,

feature relevance explanation, explanation by example and visual explanation [1]. Recently, a metro map of the information concept has been developed [4] that is capable of creating structured summaries of information. The name was selected as a metaphor for a real cartographic map, i.e. in the same way that these maps help people understand their surroundings, metro maps help them understand the information landscape [5]. Moreover, visualization with metro maps (also information cartography) can tell stories to users, on the one hand, and provide them with good directions, on the other. Indeed, the metro map consists of a set of lines, where each line interprets the same story from a different aspect. Metro stops on these lines introduce salient pieces of information, while the interrelations among these pieces ensure the plot of the story.

The first algorithm for ARM was Apriori, which was proposed back in 1994 by Agrawal [6]. Apriori is still the most popular algorithm for mining association rules. This was also considered to be one of the top 10 algorithms in ML [7]. Some of the other well-established algorithms in this domain are Eclat [8], FP-Growth [9], Genetic Association Rules (GAR) [10], the Multi-Objective Differential Evolution algorithm for mining Numeric Association Rules (MODENAR) [11], and BatMiner [12].

Several algorithms were proposed for reducing the redundant or even meaningless association rules. For instance, Feng et al. [13] employed an algorithm for rule mining based on the theory of soft sets. Recently, the logical formulas over soft sets were proposed by Feng et al. [14] for solving the same problem. On the other hand, Luna et al. [15] used a series of algorithms based on the MapReduce framework and applied this for mining frequent patterns hidden in big data. To reduce long execution times by nature-inspired population-based algorithms for ARM, Djenouri et al. [16] developed a Genetic Algorithm running on Clusters of GPUs (CGPUGA) that improved performance of state-of-

- Iztok Fister Jr. and Iztok Fister are with the Faculty of Electrical Engineering and Computer Science, University of Maribor, Koroška cesta 46, 2000 Maribor, Slovenia. Email: iztok.fister1@um.si
- © 2021 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

the-art significant for ARM on big datasets. [17], [18]

Interestingly, the visualization of association rules has rarely been treated in literature. Indeed, the papers which referred to this topic can be summarized in the following review: The authors in [19] presented a design that is able to handle hundreds of multiple antecedent association rules in a three dimensional display with minimum human interaction, low occlusion percentage, and no screen swapping. The authors in [20] show that the use of Mosaic plots and their variant, called Double Decker plots, can be applied for visualizing association rules. Ong et al. [21] prototyped the two visualizations, called grid view and tree view, for visualizing the association rules in their application called CrystalClear. Appice and Buono [22] presented a graph-based visualization that supports data miners in the analysis of multi-level spatial association rules, while Herawan et al. [23] proposed an approach for visualizing soft maximal association rules. Hahsler [24] proposed the R-extension package *arulesViz* for visualizing the association rules using the most popular visualization techniques. A very interesting interactive visualization technique, which lets the user navigate through a hierarchy of association rule groups, is presented by Hahsler and Chelluboina [25]. The authors in paper [26] explored Hasse diagrams for the visualization of Boolean association rules. Fister et al. [27] have proposed a method for identifying dependencies among mined association rules based on population-based metaheuristics and complex networks. However, there are also generic tools, like the *CloseViz* [28] and the *SPMF* open-source data mining library Version 2 [29], specialized primarily in pattern mining, offering visual implementation of discovered data mined by ML algorithms that could also be used for visualization of ARM.

Information cartography is a process of building models (i.e., information maps) based on non-geographical data [30]. It draws on the metaphor of an information landscape populated by information landscapes forming an information map, and enables an analysis of data having a "geographic hook". In order to be visualized, information needs to be spatialized, i.e., transformed into a coordinate system and placed on a surface such as a map. The first definition of the term in the sense of visualization was made by Stephen Paling in [31], where he established that maps are among the best information systems and require little documentation to be used commonly and to be understandable.

The majority of people are linear thinkers and process information in a linear manner, i.e., analytically. Linear thinking is known as a step-by-step progression, where a response to one step must be elicited before another step is taken [32]. It is connected with the logical, analytical, and sequential processing of information in contrast to circular, non-linear, and parallel ones. Typically, this kind of thinker operates in a two-dimensional world where time is of the utmost importance [33]. The concept of information maps in ARM comply strongly with the perception of linear thinkers, where attributes in metro lines describe a linear sequence of attributes, while the mutual connections between metro lines reveal how the attributes in one metro line affect the attributes in others, and vice versa. These linear sequences can also be seen from a time point of view. This type of infor-

mation cannot be obtained by the other ARM visualization methods (e.g., scatter plot, mosaic plot, complex networks etc.).

A seminal work in automatic creation of the metro maps was performed by Shahaf et al. in 2012 [4] that treated their construction as an optimization problem. The proposed method was applied to analyzing the concise structured set of documents, from which structured summaries of information were created. In paper [34], the same authors solved the problem of information overload arisen in scientific literature by proposing the same methodology. The zoomable metro maps, proposed by Shahar et al. [5], help users that might be interested in information of different levels of granularity. In paper [35], Shahaf et al. showed that metro maps can even tell stories, as well as provide good directions. Interestingly, the authors of the majority of the reviewed papers proposed deterministic methods for constructing the metro maps.

The aim of the study is to develop a method for creating the metro maps of ARM information automatically. The proposed method acts as follows: The ARM information are hidden in ARM datasets produced by algorithms for ARM in the form of implication rules $X \Rightarrow Y$. In general, these rules are represented as conjunctions of more antecedent and more consequent attributes. At first, each complex rule is simplified to a set of simple ones consisting of one antecedent and one consequent. These rules serve as building blocks for the construction of an attribute graph. The attribute graph consists of nodes representing attributes (e.g. X, Y), and direct edges denoting an implication relation (e.g. $X \Rightarrow Y$). The construction of a metro map is defined as an optimization problem that searches for the best metro lines within the attribute graph. The problem is solved using an Evolutionary Algorithm (EA), because it can be applied to the problems where no specific knowledge is discovered and enables parameters to be set adaptively. Finally, the metro map is visualized.

The main contributions of this paper can be summarized as follows:

- the first application of information cartography for visualizing the association rules,
- widening the applicability of information cartography to ARM,
- supplementing the set of visual explanation techniques for post-hoc explainability with information cartography.

The software is planned to be included into the Universal ARM Solver package [36].

The paper is structured as follows: Section 2 introduces the basic information needed for understanding the subject that follows. In Section 3, the proposed method for information cartography in ARM is illustrated in detail. Section 4 presents the results of the method by constructing metro maps of information obtained by five well known datasets, while the paper concludes with a summary of the performed work and outlines directions for future work.

2 BASIC NOTATION

The present section consists of two subsections. The former introduces the problem of ARM, while the latter describes

the formal definition of the objectives necessary for identifying the information cartography in ARM as an optimization problem.

2.1 Association Rule Mining

ARM can be defined formally as follows: Let us assume, a set of objects $O = \{o_1, \dots, o_N\}$ and transaction dataset $T_D = \{T\}$ are given, where each transaction T is a subset of objects $T \subseteq O$. Then, an association rule is defined as an implication:

$$X \Rightarrow Y, \quad (1)$$

where $X \subset O, Y \subset O$, and $X \cap Y = \emptyset$. In order to estimate the quality of the mined association rule, two measures are defined: confidence and support. The confidence is defined as:

$$conf(X \Rightarrow Y) = \frac{n(X \cup Y)}{n(X)}, \quad (2)$$

whereas support is:

$$supp(X \Rightarrow Y) = \frac{n(X \cup Y)}{|T|}, \quad (3)$$

where the function $n(\cdot)$ calculates the number of repetitions of a particular rule within T_D , and $|T|$ is the total number of transactions in T_D . Let us emphasize that two additional variables are defined, i.e. minimum confidence C_{min} and minimum support S_{min} . These variables denote a threshold value limiting the particular association rule with lower confidence and support from being taken into consideration.

In our study, transaction databases are represented as a matrix of columns corresponding to features and rows corresponding to transactions with assigned attributes. Before using the databases, objects need to be created. Actually, the objects are represented as $\langle \text{feature}, \text{attribute} \rangle$ pairs obtained by enumerating the feature and corresponding attributes from a domain of values that the definite feature can capture. Typically, objects in the ARM databases are designated fully by concatenation of the feature and attribute using the underlined character.

2.2 Formal definition of objectives

The concept of a metro map is applied in order to visualize the archive of mined association rules [5]. In our study, the metro map is defined formally as $\mathcal{M} = (G, \Pi)$, where $G = (A, E)$ denotes an attribute graph of vertices $A = \{X_1, \dots, X_N\}$, representing objects (i.e., $\langle \text{feature}, \text{attribute} \rangle$ pairs), and edges $E = \{r_1, \dots, r_M\}$, representing simple rules, together with incident function ψ_G that associates an ordered pair $\psi_G(r_k) = (X, Y)$ with direct edge r_k , when a simple association rule exists in the form of $X \Rightarrow Y$, and Π represents a set of metro lines $\pi \in \Pi$, where each metro line is defined as a permutation of edges r_1, \dots, r_τ . In the definitions, variables N and M denote the maximum number of vertices and maximum number of edges, respectively. Thus, the simple association rule consists of only one antecedent and one consequent, where the former is mapped to the source node $X \in G$ and the latter to the sink node $Y \in G$ of the corresponding attribute graph, while the path $X \Rightarrow Y$ leads from the source to the sink node.

In general, the association rules in the archive consist of more antecedents and more consequences, in other words:

$$X_1 \wedge X_2 \wedge \dots \wedge X_p \Rightarrow Y_1 \wedge Y_2 \wedge \dots \wedge Y_q. \quad (4)$$

The simple association rules are obtained from the mined rules by pairing each antecedent with each consequent, in other words:

$$(X_1 \Rightarrow Y_1), (X_1 \Rightarrow Y_2), \dots, (X_p \Rightarrow Y_q). \quad (5)$$

In this process of simplifying rules, the $p \times q$ pairs of simple rules are obtained representing direct edges in the association graph.

In the ARM domain, there are three types of attributes, i.e. categorical, numerical, and mixed. The categorical attributes consist of a domain of discrete values, while the numerical ones are assigned to a domain of continuous real values that must be discretized before use. The last attribute type can employ elements from both of the aforementioned domains.

The objects can appear in mined association rules as: (1) antecedent only, (2) consequent only, or (3) antecedent in one and consequent in the other rule. In line with this, these are divided into three subsets, i.e. $Source(G)$, $Sink(G)$, and $Intern(G)$. In graph G , the objects in antecedent subset $X \in Source(G)$ represent source nodes with indegree zero, the objects in consequent subset $Y \in Sink(A)$ are sink nodes with outdegree zero, while the objects in the mixed subset $\langle X|Y \rangle \in Intern(G)$ denote the intern nodes with indegree and outdegree higher than zero. Indeed, the antecedent set consists of nodes suitable for starting metro stops on the metro lines, the consequent set for the final metro stops, while the intern set determines the intermediate metro stops and outlines a definite path towards achieving a certain end destination.

The algorithm for constructing the metro map for visualizing the association rules needs to fulfill the following four objectives:

- maximum line coherence,
- maximum map size,
- high coverage,
- high structure quality.

The maximum line coherence limits the number of intermediate metro stops in some metro line, and is expressed by the following relation:

$$coherence(\mathcal{M}) \leq \tau, \quad (6)$$

where the variable τ determines the maximum number of intermediate metro stops. The maximum map size is referring to the number of metro lines K , in other words:

$$|\mathcal{M}| \leq K. \quad (7)$$

Indeed, we are interested in covering our information domain by the number of the metro lines that are close to K , and all the metro lines must be as coherent in the number of metro stops as possible.

The coverage estimates how well the selected metro line exploits the attributes in the transaction database. In line with this, the lift measure of association rule $Lift(X \Rightarrow Y)$ is used that is expressed as:

$$Lift(X \Rightarrow Y) = \frac{supp(X \cup Y)}{supp(X) \times supp(Y)}. \quad (8)$$

Actually, we are interested for rules with lift value > 1 , that determines the degree to which the probability of occurrence of the antecedent and this of the consequent are dependent on one another, and makes those rules potentially useful for prediction. The coverage of the whole metro line $\pi \in \Pi$ is expressed as:

$$coverage(\pi) = \frac{1}{|\pi|} \sum_{r \in \pi} Lift(r), \quad (9)$$

where r represents the particular simple association rule $X \Rightarrow Y$. Finally, the coverage of the metro map is a simple average of all the proposed metro lines, in other words:

$$coverage(\Pi) = \frac{\sqrt{|\Pi|}}{|\mathcal{M}|} \sum_{\pi \in \Pi} coverage(\pi), \quad (10)$$

where $|\Pi|$ is the sum of edges (i.e., rules) in all the metro lines. Indeed, the ratio $\frac{\sqrt{|\Pi|}}{|\mathcal{M}|}$ expresses the connectivity of the observed metro line. The ratio prefers the metro line with many long metro lines, and violates those with many short ones.

The metro map structure quality demands that all the metro stops in each metro line cannot follow to the some metro stops that appear before the observed one in the other metro line. In the other words, all consequent attributes of an attribute $x_{i,k}$ in the i -th metro line cannot appear as an antecedent of the same attribute in all the other metro lines. For instance, let us assume two metro lines π_i and π_j :

$$\begin{aligned} \pi_i : & (x_{i,1} \Rightarrow \dots \Rightarrow x_{i,k} \Rightarrow \underbrace{x_{i,k+1} \Rightarrow \dots \Rightarrow x_{i,\tau_i}}_{Cons(x_{i,k}|\pi_i)}, \\ \pi_j : & (\underbrace{x_{j,1} \Rightarrow \dots \Rightarrow x_{j,l-1} \Rightarrow x_{j,l} \equiv x_{i,k}}_{Ante(x_{i,k}|\pi_j)} \Rightarrow \dots \Rightarrow x_{i,\tau_i}) \end{aligned} \quad (11)$$

Then, a violation between two metro lines $\chi(\pi_i|\pi_j)$ appears, when an intersection of the observed consequent $Cons(x_{i,k}|\pi_i)$ and antecedent $Ante(x_{i,k}|\pi_j)$ is not empty. Mathematically, the violation is expressed as:

$$\chi(\pi_i|\pi_j) = \begin{cases} 0, & \text{if } Cons(x_{i,k}|\pi_i) \cap Ante(x_{i,k}|\pi_j) = \emptyset, \\ -1, & \text{otherwise,} \end{cases} \quad (12)$$

for $i = 1, \dots, |\mathcal{M}| - 1$ and $j = i + 1, \dots, |\mathcal{M}|$.

In the sense of the defined objectives, the problem of metro map construction is defined formally as:

$$\max coverage(\Pi), \quad (13)$$

subject to

$$sQuality(\Pi) = \sum_{i=1}^{|\mathcal{M}|-1} \sum_{j=i+1}^{|\mathcal{M}|} \chi(\pi_i|\pi_j), \quad (14)$$

where $\pi_i, \pi_j \in \Pi$. Let us mention that the value of $sQuality(\Pi)$ in Eq. (14) refers to the number of violations which occurred in constructing the metro map. When the value is zero, the valid metro map is obtained.

3 PROPOSED METHOD

The proposed method for information cartography in ARM consists of the following four steps (Fig. 1):

- creating the ARM database,
- association rule simplification,
- attribute graph definition,
- metro map construction,
- metro map visualization.

The ARM database is a result of the ARM algorithm, where the modern stochastic population-based nature-inspired algorithms, like BatMiner, can also be used instead of the classical approaches, like Apriori or FPGrowth.

On the other hand, the results of the ARM algorithms are slightly confusing for information cartography in the sense that all objects in the mined association rules can emerge as antecedent in one and consequent in another rule. In a mathematical sense, we must ensure that any of the following three inequations: $Source(G) \neq \emptyset$, $Sink(G) \neq \emptyset$, and $Intern(G) \neq \emptyset$ is valid. The aforementioned conditions are satisfied in our study by proper filtering, where the set of the most similar rules is searched for by using the linear programming algorithm.

The main characteristics of the ARM databases are that the mined association rules are in a broad form with many antecedent, as well as consequent, attributes. This form is not appropriate for the creation of attribute graphs and, therefore, rules need to be transformed into a simplified form. The simplification rule procedure was discussed in detail in the previous section and therefore avoided here.

The third step is dedicated for creating the attribute graph $G = (A, E)$, where all the simplified association rules are incorporated in the adjacent matrix $\mathbf{A}_G = (a_{i,j})$ of dimension $N \times N$, where:

$$a_{i,j} = \begin{cases} 1, & \text{if } X \Rightarrow Y, \\ 0, & \text{otherwise.} \end{cases} \quad (15)$$

It should be mentioned that no loops are allowed in this graph because of $a(i,i) = 0$ for $i = 1, \dots, N$. In order to characterize the type of attribute graph, a metric Average Path Length (APL) is introduced that identifies the end-to-end hop distances among all possible node pairs over the graph [37]. Finally, the classification of attributes in the three distinguished sets (i.e. $Source(G)$, $Intern(G)$, and $Sink(G)$) is also performed in this step.

In the remainder of the section, the fourth step is described, where the metro map $\mathcal{M} = (G, \Pi)$ is constructed, while the visualization of the metro map is examined in the next section.

3.1 EA for metro map construction

Although several authors of the existing methods for creating metro maps proposed deterministic algorithms, the EA was introduced in our study. The motivation behind using this kind of stochastic nature-inspired population-based algorithms was due to their advantages over traditional optimization algorithms (such as gradient-based methods) that can be summarized as follows [38]: (1) Conceptual simplicity, (2) Broad applicability, (3) Their ability to outperform traditional methods on real problems, (4) The potential to use knowledge and hybridize with the other methods, (5)

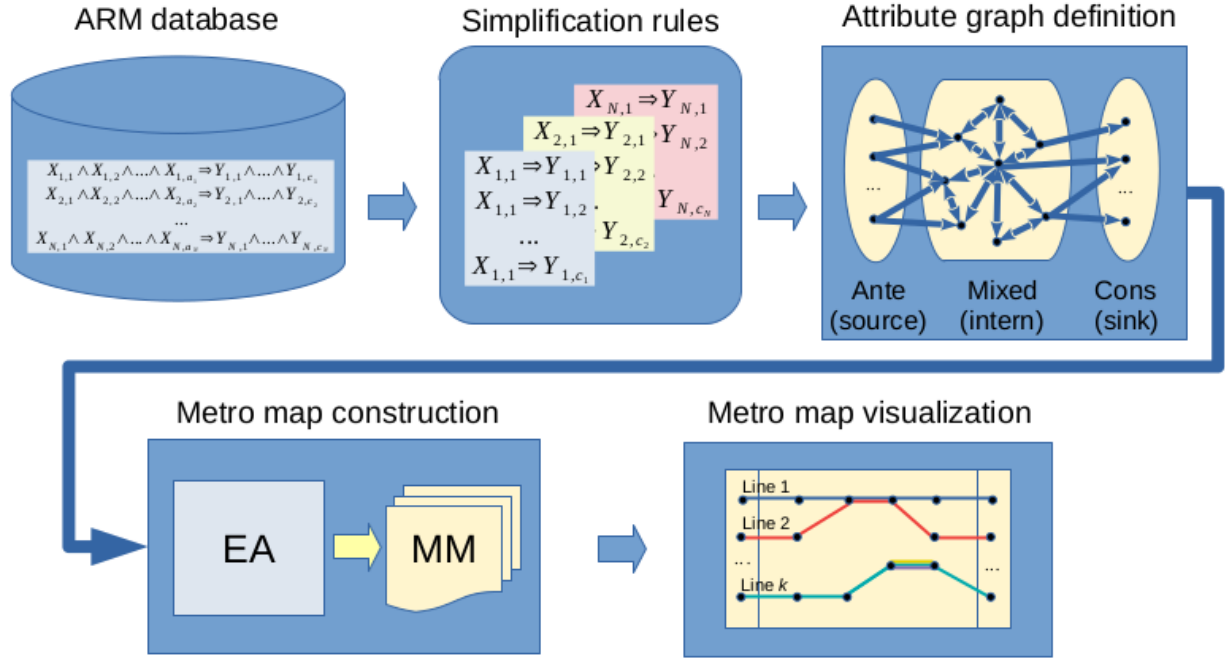


Fig. 1: Concept of metro map creation in association rule mining.

Parallelism, (6) Their robustness to dynamic changes, (7) Their capability of self-optimization, and (8) Their ability to solve problems that have no known solutions.

The EA for metro map construction demands modifications of the following algorithm components [39]:

- representation of a solution,
- variation operators (i.e. crossover and mutation),
- survivor and parent selection,
- fitness function evaluation,
- initialization,
- termination condition.

The problem of metro map construction is constrained, because every feasible solution must satisfy the relation $sQuality(\Pi) = 0$.

3.1.1 Representation of solutions

Each solution in the population of Np individuals represents a metro map y_i that is encoded as follows:

$$y_i = \left(\underbrace{n_i}_{\text{Control variable}}, \underbrace{\mathbf{x}_{i,1}, \mathbf{x}_{i,2}, \dots, \mathbf{x}_{i,n_i}}_{\text{Description of metro map } \Pi} \right), \quad (16)$$

where the first part of the representation is dedicated for control meta-parameter n_i that determines the number of metro lines and determines individuals of variable length. The second part consists of detailed descriptions for particular metro lines, expressed as:

$$\mathbf{x}_{i,j} = \{x_{i,j,1}, \dots, x_{i,j,|\mathbf{x}_{i,j}|}\}, \quad \text{for } i = 1, \dots, n_i, \quad (17)$$

where each element $x_{i,j,k}$ for $k = 1, \dots, |\mathbf{x}_{i,j}|$ encodes a specific simple association rule $X \Rightarrow Y$, and $|\mathbf{x}_{i,j}|$ determines the number of associative rules within the metro line $\mathbf{x}_{i,j} \in y_i$. The elements are ordered into a sequence of implication rules:

$$x_{i,j,1} \Rightarrow x_{i,j,2} \Rightarrow x_{i,j,|\mathbf{x}_{i,j}|-1} \Rightarrow x_{i,j,|\mathbf{x}_{i,j}|}. \quad (18)$$

in such a way that each consequence of the k -th rule appears as an antecedent in the $k + 1$ -th rule, in other words: If consecutive elements $x_{i,j,k}$ and $x_{i,j,k+1}$ encode the rules $X_l \Rightarrow Y_l$ and $X_{l+1} \Rightarrow Y_{l+1}$ in the metro line j , then the relation $Y_l = X_{l+1}$ must hold.

Indeed, the representation of solutions also includes control parameter n_i besides the problem variables. This means that the parameter controlling the maximum number of metro lines undergoes acting the variation operators and thus adapts to the fitness landscape online.

3.1.2 Variation operators

The proposed algorithms support two variation operators, i.e. crossover and mutation. While the crossover operates on metro lines as a whole, the mutation is also capable of modifying the structure inside a particular metro line. Obviously, the application of crossover is controlled using the probability of crossover p_c , and mutation using the probability of mutation p_m .

3.1.2.1 Crossover: is defined as follows: At first, for each target metro map $\mathbf{y}^{(org)}$, a trial metro map $\mathbf{y}^{(tri)}$ is created using the same control parameters. Then, a parent metro map $\mathbf{y}^{(par)}$ is selected randomly and the metro lines for the trial are taken either from the target or parent metro maps according to probability p_c . Mathematically, this crossover is expressed as:

$$\mathbf{x}_{i,j}^{(tri)} = \begin{cases} \mathbf{x}_{i,j}^{(par)}, & \text{if } |\mathbf{y}_i^{(par)}| < |\mathbf{y}_i^{(org)}|, \\ \emptyset, & \text{otherwise,} \end{cases} \quad \text{if } U(0,1) \leq p_c, \quad (19)$$

$$\mathbf{x}_{i,j}^{(tri)} = \begin{cases} \mathbf{x}_{i,j}^{(org)}, & \text{otherwise,} \end{cases}$$

for $i = 1, \dots, |\mathbf{y}^{(org)}|$. Consequently, the operator can produce infeasible solutions in two cases:

- the size of the parent metro map is smaller than the size of the trial, i.e. $|\mathbf{y}^{(par)}|_{i,j} < |\mathbf{x}^{(par)}|_{i,j}$,

- the first antecedent of the metro line representing the starting metro stop is replicated twice.

In the first case, the corresponding metro line is deleted from the trial solution, while in the second, the solution from the parent solution is preferred.

3.1.2.2 Mutation: modifies the structure of the metro line as follows: At first, the position of mutation k is selected randomly according to the probability of mutation p_m . Then, an antecedent X_k is extracted from a corresponding association rule $\langle X_k \Rightarrow Y_k \rangle$, and the new consequent Y'_k is attached to the rule among all the possible consequent as defined by the graph $G = (A, E)$. Finally, the remaining path from r_{k+1} towards the drain is generated randomly.

Mathematically, the mutation is presented as follows: Let us assume the metro line $\mathbf{x}_{i,j} = (x_{i,j,k})$ for $k = 1, \dots, |\mathbf{x}_{i,j}|$ is given, where each element represents an association rule $x_{i,j,k} = X_k \Rightarrow Y_k$, and a position of mutation $k \in [1, |\mathbf{x}_{i,j}|]$ is selected according to the probability of mutation p_m , in other words:

$$\mathbf{x}_{i,j} = (\underbrace{\langle X_1 \Rightarrow Y_1 \rangle}_{x_{i,j,1}}, \dots, \underbrace{\langle X_k \Rightarrow Y_k \rangle}_{x_{i,j,k}}, \underbrace{\langle X_{k+1} \Rightarrow Y_{k+1} \rangle}_{x_{i,j,k+1}}, \dots, \underbrace{\langle X_L \Rightarrow Y_L \rangle}_{x_{i,j,L}}).$$

The result of the mutation is expressed as follows:

$$\mathbf{x}'_{i,j} = (\underbrace{\langle X_1 \Rightarrow Y_1 \rangle}_{x_{i,j,1}}, \dots, \underbrace{\langle X_k \Rightarrow Y'_k \rangle}_{x'_{i,j,k}}, \underbrace{\langle X'_{k+1} \Rightarrow Y'_{k+1} \rangle}_{x'_{i,j,k+1}}, \dots, \underbrace{\langle X'_L \Rightarrow Y'_L \rangle}_{x'_{i,j,L}}).$$

where $L = |\mathbf{x}_{i,j}|$ and all the modified values of the metro line are denoted by apostrophes. Let us mention that the operation also has an impact on the metro line length L that can be increased or decreased within the allowed maximum metro line length (i.e. $L \leq \tau$).

3.1.3 Survivor and parent selection

One-to-one selection is applied as an operator of survivor selection that is borrowed from Differential Evolution (DE) [40]. This selection works on the whole metro map \mathbf{y}_i . Mathematically, it is expressed as follows:

$$\mathbf{x}_i^{(org)} = \begin{cases} \mathbf{y}_i^{(tri)}, & \text{if } f(\mathbf{y}_i^{(tri)}) \leq f(\mathbf{y}_i^{(org)}), \\ \mathbf{y}_i^{(org)}, & \text{otherwise,} \end{cases} \quad (20)$$

where the better between trial $\mathbf{y}_i^{(tri)}$ and target $\mathbf{y}_i^{(org)}$ vector is preserved as the candidate solution map that proceeds into the next generation.

Parent selection is applied by a crossover operator for generating the trial solution. Although there are many parent selection operators, the parent selection implemented in our study selects the parent randomly among the other population members.

3.1.4 Fitness function evaluation

The problem of constructing metro maps is constrained in its nature. However, constraints can be handled in several ways in EC [39]. In our case, penalizing the infeasible solutions is employed within the proposed EA, where the value of a penalty function is assigned to the fitness function, when its value is less than zero. When the penalty function becomes zero, the coverage function takes the initiative. Mathematically, the fitness function is expressed as follows:

$$f(\mathbf{y}_i) = \begin{cases} coverage(\mathbf{y}_i), & \text{if } sQuality(\mathbf{y}_i) = 0, \\ -sQuality(\mathbf{y}_i), & \text{otherwise.} \end{cases} \quad (21)$$

The task of the optimization algorithm is to maximize the value of the fitness function. The design of such the fitness function is capable of preferring the more promising simple association rules on the one hand, and ensures that the constructed metro lines are proper. Let us mention that the proposed fitness function also ensures the good behavior of the algorithm on highly non-separable datasets, where the metro-lines are highly interrelated.

3.1.5 Initialization

Initialization of individuals in the population is performed randomly. At first, the number of metro lines is generated randomly from the interval $[2, K]$. Then, the unique source node is selected from the set of source nodes. For each source node, the random path in the attribute graph $G = (A, E)$ is searched for so that the maximum metro line length τ is not exceeded.

3.1.6 Termination condition

In our study, the EA is terminated, when the fitness improvement is zero for more than the predefined *Threshold* number of generations. The termination condition parameter is adaptive, because it demands a feedback from the evolutionary search process in order to determine how big the loss of the variability is in improving an average of the fitness value within the population of solutions.

3.1.7 Pseudo-code of the algorithm

The pseudo-code of the proposed EA for constructing metro maps is illustrated in Algorithm 1, in which all components of a general EA as proposed by Eiben and Smith in [39] can be indicated. Actually, all these components are described in detail in the previous subsections. Therefore, here, we are focused on a description of the common principles used by solving the construction problem.

Algorithm 1 Pseudo-code of an EA.

```

1: procedure EVOLUTIONARY_ALGORITHM
2:   INITIALIZE_population_randomly;
3:   EVALUATE_each_candidate_solution;
4:   while TERMINATION_CONDITION_not_met do
5:     SELECT_PARENTS;
6:     RECOMBINE_pairs_of_parents;
7:     MUTATE_resulting_offspring;
8:     EVALUATE_each_candidate_solution;
9:     SELECT_SURVIVOR_solutions;
10:  end while
11:  return best_solution
12: end procedure

```

Indeed, the construction process must fulfill four objectives, where maximum coherence and maximum map size are obeyed implicitly by the variation operators (functions 'RECOMBINE_pairs_of_parents' and 'MUTATE_resulting_offspring'). This means that only the feasible solutions are taken into consideration, while the others are deleted from the population. Which parents enter into the variation process is determined by the parent selection operator (function 'SELECT_PARENTS'). The other two objectives (i.e., high coverage and high structure quality) are

captured in the fitness function of the evolutionary process (function 'EVALUATE_each_candidate_solution') that operates on the 'generate-and-test' principle. In this sense, only the best solutions according to the fitness function can survive and transfer their good characteristics into the next generations (function 'SELECT_SURVIVOR_solutions'). In order to run the EA, also initialization (function 'INITIALIZE_population_randomly'), termination condition (function 'TERMINATION_CONDITION_not_met'), and representation of solution (see Section 3.1.1) are incorporated within the proposed EA.

The remainder of the section is devoted to analysis of the time complexity of the proposed algorithm as well as to discussion of the parameters controlling the behavior of particular EA components.

3.1.7.1 Complexity and running time: Time complexity depends on the problem size (the population size) and the search space dimension (the length of individuals) [41]. However, EAs are stochastic. This means that there are no guarantees for the algorithm to reach an optimum. Moreover, this condition might never get satisfied and, consequently, such algorithm could never stop. As a result, performance measures, like the maximum number of generations or evaluations of the fitness function, or population diversity drop under some threshold, serve for approximating the optimum of a given fitness function [39]. Consequently, we are interested about the mathematical measures in the EA community (i.e., convergence velocity, convergence rate, convergence order), describing the convergence behavior of EAs more than the time complexity. In line with this, the better the convergence rate, the better the EA.

3.1.7.2 Parameters: Setting parameters has a crucial impact on the performance of the algorithms. Although an algorithm for metro map construction depends on even six parameters, we can propose some hints for their proper tuning. For instance, the maximum number of metro lines is a self-adaptive control parameter, where the proper value is drawn by the EA from some predefined interval of integer values. Thus, the reasonable minimum value of the interval should be set to 2, while the maximum value to the maximum number of the starting metro stops (i.e., $Source(G)$) found in the attribute graph. The last value can be delivered easily by the proposed method. The termination condition in our study is an adaptive control parameter that also could be set loosely depending on the precision of the calculation. The parameter population size needs to enable bias between exploration and exploitation firmly within the evolutionary search process, while setting the parameters probability of crossover and probability of mutation complies with the theory of Evolutionary Computation [39]. Finally, setting the optimal number of metro stops stays the only control parameter that needs explicit tuning, in which help of users is also welcome by estimating the quality of the produced metro maps. However, the reasonable interval of values for this parameter should be set as follows: the minimum value could be set to 3, while the maximum value to the average path length as found in the corresponding attribute graph.

4 RESULTS AND DISCUSSION

The purpose of our experimental work was to show that the huge amount of data obtained by algorithms for ARM can be extracted automatically in the form of structured knowledge called information cartography in ARM and visualized by metro maps of ARM information. These maps can help users understand information in many knowledge domains. In line with this, we applied our proposed method to five different datasets that accompany data from various domains, e.g. biology, chemistry, and sports.

In summary, using metro maps has several advantages for users. In line with this, the experimental study includes the following research questions:

- RQ-1: extract the most important association rules from a huge amount of data,
- RQ-2: find the more <feature,attribute> pairs by automated extracting of information and to direct the attention of users to them,
- RQ-3: apply the proposed visualization method to new application domain,
- RQ-4: derive the story that metro maps narrate.

The information cartography in ARM is a very complex task that comprises of five tasks as discussed in Section 3. Therefore, the results of each phase (except association rule simplification) are illustrated in detail in the remainder of the paper. Especially, we focus on exposing the characteristics of the algorithm for constructing metro maps.

4.1 ARM datasets creation

In our study, evaluating the efficiency of the proposed method for information cartography in ARM was performed on four public datasets from the UCI Machine Learning repository [42] and one Sport dataset, consisting of real data obtained from device trackers worn by sport athletes (i.e. cyclists) during their sport training sessions [27].

The characteristics of the aforementioned datasets are illustrated in Table 1. Let us emphasize that the results of ARM for the first four datasets were obtained using the Apriori algorithm [6], while for the Sport dataset they were obtained by using the BatMiner [12].

TABLE 1: The ARM datasets characteristics.

Dataset	Characteristics			Mined rules	
	Instan.	Featur.	Attrib.	All	Filtered
Mushroom	8,124	23	126	24,408	998
Iris	150	5	24	182	88
Abalone	4,177	9	28	36,388	2,779
Wine	178	13	55	5,483	2,355
Sport	80	14	87	4,191	10

Indeed, the table consists of two parts, in which the first indicates the characteristics of particular datasets, like the number of instances, the number of features, and the number of attributes, while the second contains the results of the algorithm for ARM. The results are presented in two columns: The column 'All' denotes the number of all mined association rules, while the column 'Filtered' includes the number of all rules obtained after filtering.

In a nutshell, the characteristics of the aforementioned datasets are as follows: The Mushroom dataset contains

logical rules for mushrooms, indicating if a specific one is poisonous or edible. The dataset includes descriptions of hypothetical samples corresponding to 23 species of gilled mushrooms, where each species is identified as definitely edible, definitely poisonous, or not recommended. The last class was combined with the poisonous one. As a result, there are two final classes, i.e. the edible (included into 865 rules as a consequent), and the poisonous mushrooms (included into 133 rules as a consequent). All samples are discrete values.

The Iris dataset collects transactions describing characteristics for the classification of three different iris plants (i.e. Setosa, Versicolour, and Virginica), where each transaction is constructed from five features with 21 numerical attributes. Thus, these transactions are divided equally among three classes (33.3 % for each of the three classes).

The Abalone dataset is devoted to predicting the age of abalone from physical measurements [42]. All measurements are numerical. The age of abalone is determined by cutting the shell through the cone, staining them, and then counting the number of rings through a microscope. Other measurements, which are easier to obtain, are used to predict the age. There are four classes originally included in the dataset. Thus, the first one is included into 69.95 % of rules as a consequent, the second one in 30.05 % of rules as a consequent, while the third one is not included in any association rule as a consequent and is, therefore, eliminated from the observation.

The data in the Wine dataset are obtained as the results of a chemical analysis of wines grown in the same region in Italy, but derived from three different cultivars. The analysis determined the quantities of 13 constituents (features) found in each of the three types of wines: the first is included into 59 rules as a consequent, the second included in 71 rules as a consequent, and the third included in 48 rules as a consequent. In summary, the number of attributes is equal to 19, where each of these is numerical.

A Sport dataset was produced from the TCX files of a professional, male cyclist with many years of experience, who donated his data voluntary for the purposes of this study. Every training session was tracked by a wearable sports watch and the data imported into a training dataset. In addition to these performance data, data for estimation of the athlete’s psycho-physical conditions were also merged with each training session by the sports trainer. As a result, the measurements are of mixed types, i.e., numerical and discrete. Let us emphasize that this dataset does not include any results of classification.

4.2 Attribute graph definition

In this step, the rules in the filtered ARM dataset are transformed into simple rules, at first. Because this step is trivial, it is omitted from a detailed discussion here. Obviously, the simple rules serve as building blocks for attribute graph definition. Actually, the results of the attribute graph definition are presented in Table 2 which, in addition to the attribute graph specification for the particular dataset (i.e., number of nodes, edges, and APL), also depicts the sizes of the antecedent, consequent, and mixed sets that actually represent the number of source, sink, and intern nodes in

TABLE 2: Characteristics of created attribute graphs.

Dataset	Graph			Story important nodes		
	Node	Edge	APL	Source	Intern	Sink
Mushroom	16	116	23	4	10	2
Iris	18	38	5	10	5	3
Abalone	22	124	8	8	12	2
Wine	37	308	11	11	23	3
Sport	21	52	6	5	7	9

the attribute graph, respectively. Together, these nodes are also called story important nodes.

As can be seen from the table, the Iris and Sport datasets are relatively simple from the problem solving point of view due to only five intern nodes (i.e. attributes). This restricts the algorithm for a metro map construction to search for solutions in a smaller search space, especially when we assume that the majority of the other nodes in the corresponding attribute graph are identified as source and sink ones. The remaining attribute graphs are more complex due to the higher number of nodes, as well as edges.

4.3 Metro map construction

The EA for the metro map construction used parameters set as illustrated in Table 3, from which it can be seen that the population size $Np = 100$ was applied in all experiments. Actually, this value ensures a good bias between exploration and exploitation, and avoids the evolutionary search process getting stuck in local optima. Also, both probabilities, p_c and p_m were set according to the guidelines valid in the EC community. The EA was terminated when the EA did not improve the fitness values during the last 100 generations (parameter *Threshold*). Obviously, the number of metro lines K is problem dependent, while its value refers to the number of source nodes in the attribute graph. Therefore, the optimal value of this parameter was self-adapted in our study. The most specific was setting of the length of metro lines τ that depends on the average path length of the constructed attribute graph. This parameter is also problem specific on the one hand and strongly affects the quality of the constructed metro maps on the other. Because the quality of metro maps is left to the judgment of the users, the proper setting of this parameter needs to be made by a user manually, by selecting the best solution from a collection of constructed metro maps. Each dataset was optimized 25 times, and the results of the best run obtained by constructing the metro maps based on five aforementioned datasets were selected for further analysis. This means that the run with the highest fitness of the 25 repeats was observed.

In the remainder of the section, the EA for constructing the metro maps is analyzed in detail. The analysis was conducted on the Wine dataset that is complex enough to reveal the characteristics of the algorithm. Obviously, a similar analysis could be performed on the other datasets as well.

¹Number of source nodes found in attribute graph.

²APL of the corresponding attribute length.

TABLE 3: Parameter setting of EA for metro map construction.

Parameter	Abbreviation	Value
Population size	Np	100
Probability of crossover	p_c	0.5
Probability of mutation	p_m	0.01
Period of fitness stagnation	<i>Threshold</i>	100
Number of metro lines	K	$[2, Source(G) ^1]$
Optimum metro line length	τ	$[3, APL_G^2]$

4.3.1 Analysis of the algorithm

The results of the EA for constructing the metro maps obtained on the Wine dataset by varying the metro line lengths $\tau \in [3, 11]$ are illustrated in Table 4, where the metro maps are presented with metro lines of path lengths 3, 5, 8, and 11. A legend, transforming the object numbers within the association rules to $\langle \text{feature}, \text{attribute} \rangle$ pairs, is attached to the table. Interestingly, the metro map with metro line length of $\tau = 5$ revealed the best result according to the fitness function.

As can be seen from the table, each metro lines obeys the maximum number of metro stops strictly. For instance, all metro lines in the corresponding metro map in column 'Path=3' consist of three metro stops (i.e., a start, one intern, and a destination metro stop). This is also valid for the all the other path lengths.

In summary, the number of metro lines decreases with increasing the metro stops. Moreover, some sequences of metro stops in metro lines of lower path lengths form sub-sequences that can be detected in metro lines of higher path lengths. For instance, the sub-sequence '34 \Rightarrow 8' is detected in ML-8 of $\tau = 3$, in the '34 \Rightarrow 8 \Rightarrow 4 \Rightarrow 30' of ML-3 by $\tau = 4$, and in the '34 \Rightarrow 8 \Rightarrow 4 \Rightarrow 39 \Rightarrow 0 \Rightarrow 21' of ML-4 by $\tau = 4$. Interestingly, this sub-sequence proliferates into two sub-sequences in ML-3 of $\tau = 11$, i.e., '34 \Rightarrow 8' and '4 \Rightarrow 39 \Rightarrow 0 \Rightarrow 21' connected by the metro stop 3. These sub-sequences present building blocks that connect themselves into more promising metro lines of higher path length. On the other hand, some more expressed destination metro stops may disappear from the metro maps. For instance, there exist metro maps with metro lines of higher path lengths (e.g., $\tau = 8$ and $\tau = 11$) with no destination metro stop of 'class_1'.

In order to identify the characteristics of the proposed EA for constructing the metro maps, two experiments were conducted further, in which we analyze:

- the influence of the adaptive population size on the results,
- the influence of the metro line length.

The purpose of the first test was to determine how big was the error caused by using the adaptive termination condition, while the second one how the metro line length τ affected the quality of the constructed metro maps. Let us also mention that the analysis of Wine dataset was employed in this test.

4.3.1.1 Influence of the adaptive population size:

In this test, the population size was varied in the interval $Np \in \{50, 100, 200, 500, 1000\}$. Thus, the five instances of the results were obtained, to which the results produced by

the EA using the adaptive termination condition were also added. In the last case, the number of generations needed is unknown in advance. However, the results represent fitness function values obtained by the corresponding instance. The values were collected according to different metro line length, where this parameter was varied in the interval $\tau \in [3, 11]$ in a step of one.

The results of the test are illustrated analytically in Table 5, in which the best fitness function values are presented in bold. It can be seen from the table that the highest fitness function values are obtained by the metro line lengths $\tau = 5$ and $\tau = 6$. Mainly, the values below and above these metro line lengths are lower.

The same results according to different number of metro stops $\tau \in [3, 11]$ are depicted graphically in Fig. 4a, where the dependence of the fitness values are presented according to the generations. Additionally, the observed generations at $G = \{50, 100, 200, 500, 1000\}$ are denoted by reference lines in the graph.

As can be seen from the graph, the EA converges to the optimal values very fast, i.e., only 200 generations was enough for the convergence for all path lengths. There is less improvement detected by increasing the number of generations from 500 to 1000.

The last finding is considered in detail by analysis of the error in fitness function caused by introducing the adaptive population size. In line with this, the results of the EA using adaptive population size were compared with the results obtained by the EA with the fixed population size of higher value (e.g., $G = 1000$). The results of this test are presented in Table 6, which consists of columns denoting fitness values obtained by both algorithms, the number of generation necessary for obtaining the results, and the difference in fitness values between both algorithms. As can be seen from the table, the EA with adaptive population size needed for even 54 % less generations for achieving the results that are 0.5 % worse than those obtained by the EA using the fixed population size $G = 1,000$.

In graphical form, the last results can be seen in Fig. 4b. This figure shows that the EA needs more than 500 generation by constructing the metro maps with metro lines of length $\tau = 4$, $\tau = 5$, and $\tau = 6$ only. All the other instances needed a lower number of generations. On the other hand, these instances also contributed the highest values to the total error, and demanded an increase in the threshold value to *Threshold* = 100.

4.3.1.2 Influence of the metro path length.: The aim of this test was to discover how the metro line length affected the results of the EA for constructing the metro maps. In line with this, the path length of the metro lines are varied in the interval $[3, 11]$ in steps of one.

Fig. 7a illustrates the influence of the metro line lengths on the fitness values obtained by the proposed EA. The results obtained by the EA using the generation numbers changing in interval $G \in \{50, 100, 200, 500, 1000\}$ are compared with the results obtained by its counterpart using the adaptive population size.

As can be seen from the figure, the results of the EA using the adaptive generation size is comparable with the results of the same using the fixed generation size of $G = 1000$. Interestingly, the best results according to the fitness

TABLE 4: The results of EA for metro map construction obtained by the Wine dataset.

ML	$\tau = 3$	$\tau = 5$ (Best)	$\tau = 8$	$\tau = 11$
1	31⇒30⇒7	26⇒1⇒4⇒30⇒7	31⇒30⇒0⇒15⇒23⇒24⇒21⇒14	32⇒9⇒0⇒6⇒23⇒22⇒24⇒19⇒10⇒12⇒14
2	32⇒9⇒7	17⇒2⇒4⇒3⇒7	32⇒9⇒27⇒23⇒24⇒19⇒21⇒14	31⇒30⇒0⇒21⇒6⇒23⇒24⇒1⇒27⇒11⇒25
3	17⇒3⇒7	34⇒8⇒4⇒30⇒7	35⇒22⇒23⇒1⇒24⇒19⇒21⇒14	34⇒8⇒3⇒4⇒30⇒0⇒21⇒23⇒24⇒19⇒25
4	33⇒22⇒14	35⇒22⇒24⇒21⇒14	34⇒8⇒4⇒30⇒0⇒21⇒6⇒25	
5	29⇒24⇒14	31⇒30⇒0⇒21⇒25	28⇒8⇒10⇒5⇒2⇒3⇒23⇒25	
6	26⇒22⇒14	28⇒4⇒30⇒0⇒25		
7	35⇒22⇒14	32⇒9⇒27⇒23⇒25		
8	34⇒8⇒25	33⇒22⇒24⇒21⇒25		
9	20⇒11⇒25			

Nr.	Attribute	Nr.	Attribute	Nr.	Attribute
0	alcalinity_(15.45-20.3]	12	color_intensity_(-inf-4.21]	26	color_intensity_(7.14-10.07]
1	ash_(2.295-2.7625]	14	class_2	27	nonflavanoid_phenols_(0.395-0.5275]
2	malic_acid_(-inf-2.005]	17	hue_(1.095-1.4025]	28	nonflavanoid_phenols_(-inf-0.2625]
3	alcohol_(12.93-13.88]	19	magnesium_(93-116]	29	malic_acid_(2.005-3.27]
4	flavonoids_(2.71-3.895]	20	total_phenols_(1.705-2.43]	30	proanthocyanins(1.995-2.7875]
6	proanthocyanins(1.2025-1.995]	21	total_phenols_(-inf-1.705]	31	alcohol_(13.88-inf)
7	class_1	22	flavonoids_(-inf-1.525]	32	alcohol_(-inf-11.98]
8	color_intensity_(4.21-7.14]	23	hue_(-inf-0.7875]	33	malic_acid_(3.27-4.535]
9	flavonoids_(1.525-2.71]	24	proanthocyanins(-inf-1.2025]	34	magnesium_(116-139]
10	hue_(0.7875-1.095]	25	class_3	35	nonflavanoid_phenols_(0.5275-inf)
11	magnesium_(-inf-93]				

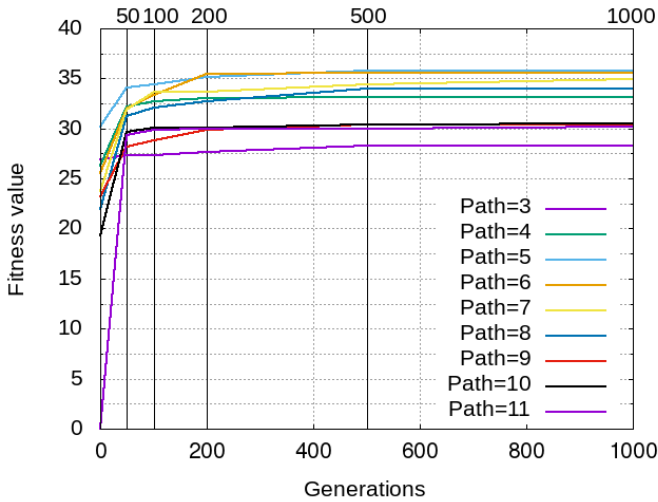


Fig. 2: subfigure

Influence of the generation number graphically.

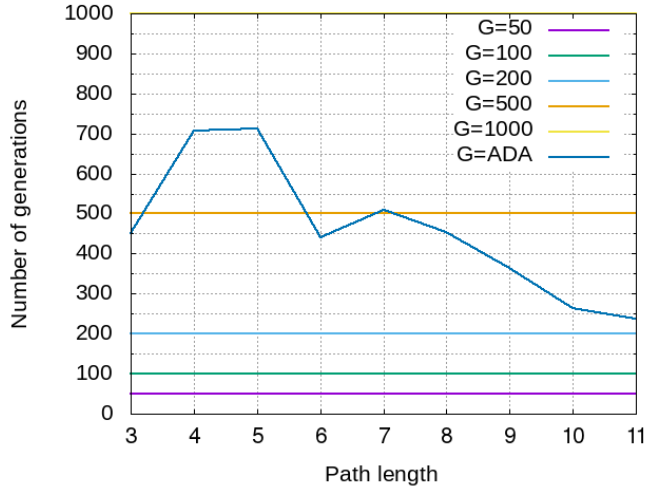


Fig. 3: subfigure

Influence of the adaptive termination condition.

Fig. 4: The results as affected by the number of generations.

TABLE 5: Influence of the generation number analytically.

τ	50	100	200	500	1,000	Adapt.
3	27.31	27.34	27.70	28.31	28.31	28.31
4	32.21	32.70	33.08	33.19	33.19	33.19
5	34.10	34.43	35.13	35.76	35.76	35.76
6	31.95	33.39	35.44	35.53	35.56	35.53
7	31.90	33.63	33.63	34.41	34.91	34.41
8	31.28	32.08	32.69	33.94	34.02	33.94
9	28.15	28.84	29.89	30.40	30.41	30.34
10	29.64	30.04	30.04	30.43	30.45	30.04
11	29.36	29.89	29.94	29.94	30.19	29.94

TABLE 6: Error by introducing the adaptive population size.

Path	f_{1000}	f_{ada}	G_{1000}	G_{ada}	Δ_f
3	28.31	28.31	1,000	451	0.00
4	33.19	33.19	1,000	707	0.00
5	35.76	35.76	1,000	713	0.00
6	35.56	35.53	1,000	440	0.02
7	34.91	34.41	1,000	509	0.50
8	34.02	33.94	1,000	455	0.08
9	30.41	30.34	1,000	363	0.07
10	30.45	30.04	1,000	265	0.41
11	30.19	29.94	1,000	237	0.26
Σ	292.82	291.46	9,000	4,140	1.35

value were obtained by the metro lines of length $\tau = 4$, while the instances below and above this values were worse. Obviously, this behavior was caused due to a connectivity

factor, expressed as a ratio of the total number of edges per metro line.

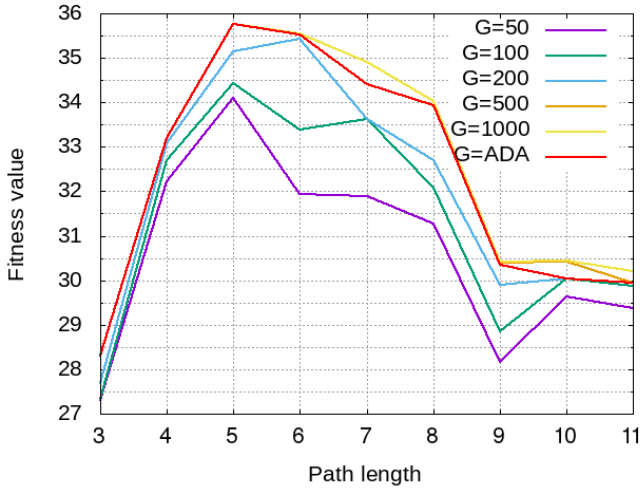


Fig. 5: subfigure
Influence of the metro line length.

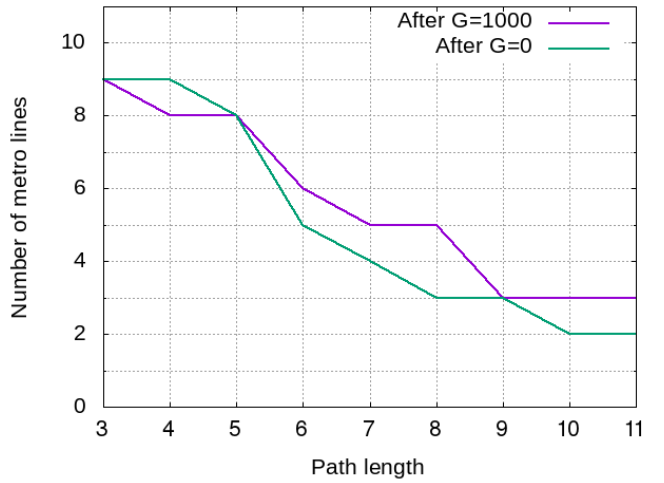


Fig. 6: subfigure
Influence of the number of metro lines.

Fig. 7: Searching for the characteristics of the optimal metro lines.

The last test was devoted to discovering how the number of metro lines evolves during the typical evolutionary cycle. In line with this, the number of metro lines in the best solution according to the fitness value was measured at the beginning (i.e., at $G = 0$) and at the end (i.e., $G = 1,000$) of the evolutionary run.

The results of the EA for constructing the metro maps are depicted in Fig. 7b, from which it can be seen that the number of metro lines decreases when the path length of the metro lines was lower, and increased in the opposite case.

4.4 Metro map visualization

In this subsection, the best results generated by the EA for ARM information are visualized and then the salient pieces of information (i.e. $\langle \text{feature}, \text{attribute} \rangle$ pairs) are integrated into metro lines and subsequently into the whole story. From the story point of view, the $\langle \text{feature}, \text{attribute} \rangle$ pairs of metro lines that are connected between each other with implication relations, represent the plot of the story. The sequence of relations also indicates the direction of the story plot. However, conflicts caused by the interrelation of metro stops of different metro lines are designated by vertical arcs.

This study is focused on the visualization of the best result obtained by constructing the metro map of ARM information based on the five previously mentioned datasets. The best results after optimizing more instances obtained by constructing the metro lines by varying the metro path length $\tau \in [3, \text{APL}_G]$ and the number of metro lines $K \in [2, |\text{Source}(G)|]$ (Table 7) are analyzed.

Thus, the best metro maps with the longer metro paths are compared several times with their counterparts with the best fitness, in order to highlight the interesting knowledge hidden in data. The results of the visualization of metro maps are illustrated in the remainder of the paper. Let us notice that all metro maps in the paper were visualized using the graphical tool embedded into the R programming environment.

TABLE 7: Experimental setup by constructing the metro maps.

Dataset	Path length	Number of ML	Instances
Mushroom	$\tau \in [3, 11]$	$K \in [2, 11]$	9
Iris	$\tau \in [3, 23]$	$K \in [2, 4]$	21
Abalone	$\tau \in [3, 5]$	$K \in [2, 10]$	3
Wine	$\tau \in [3, 8]$	$K \in [2, 8]$	6
Sport	$\tau \in [3, 6]$	$K \in [2, 5]$	4

4.4.1 Mushroom dataset

Visualization of the best run obtained by the construction of a metro map based on ARM information of the Mushroom dataset is presented in Fig. 10 that is divided into two diagrams: The upper depicts the metro map with metro lines (designated with ML-1 to ML-4) of path length $\tau = 6$ (Fig. 10a), and the lower the metro map with metro lines (designated with ML-1 and ML-2) of $\tau = 11$ (Fig. 10b). As a result, the first metro map consists of four, while the second one from two metro lines. Indeed, each metro map consists of metro stops designating the physical measurements identifying an edibility or poisonous of the observed mushrooms, where the searched characteristic is represented as a destination metro stop. Also, a legend is appended to the diagrams that transforms the tags denoting metro stops into $\langle \text{feature}, \text{attribute} \rangle$ pairs.

Interestingly, all the metro lines in Fig. 10a lead to the same destination (i.e., 'class_edible' $\langle \text{feature}/\text{attribute} \rangle$ pair), denoting the characteristics of the edible mushrooms. On the other hand, both the metro lines in Fig. 10b arose already in the first metro map, but drove to the different destination designating the poisonous mushrooms. From this evolution from the edible to poisonous mushrooms, we can infer which ingredients cause that the edible mushroom becomes poisonous. In line with this, each metro stop (i.e., measurement) in this metro map is designated with the mark 'Edi', when the metro stop is already present in the corresponding metro line

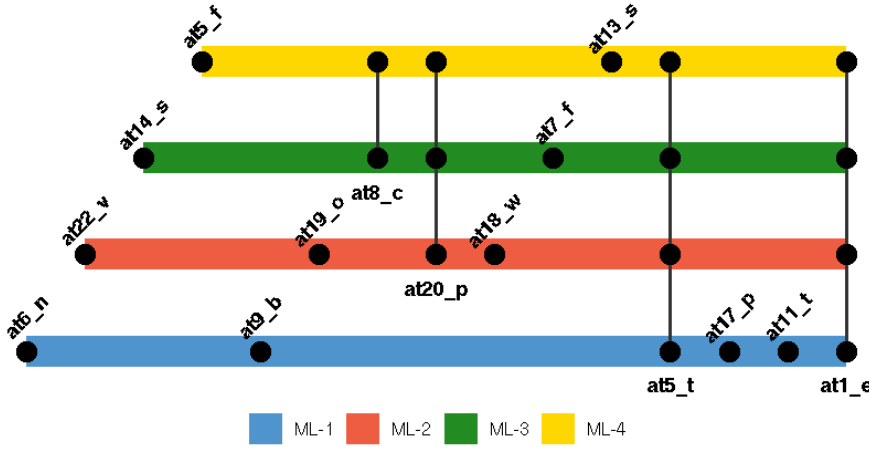


Fig. 8: subfigure
Path length $\tau = 6$.

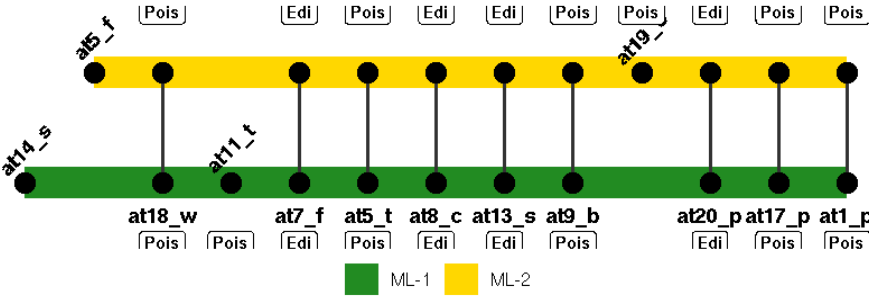


Fig. 9: subfigure
Path length $\tau = 11$.

Tag	Attribute
at1_e	class_edible
at1_p	class_poisonous
at5_f	bruises?_no
at5_t	bruises?_bruises
at6_n	odor_none
at7_f	gill-attachment_free
at8_c	gill-spacing_close
at9_b	gill-size_broad
at11_t	stalk-shape_tapering
at13_s	stalk-surface-above-ring_smooth
at14_s	stalk-surface-below-ring_smooth
at17_p	veil-type_partial
at18_w	veil-color_white
at19_o	ring-number_one
at20_p	ring-type_pendant
at22_v	population_several

Fig. 10: Visualization of the Mushroom dataset.

of the first metro map (e.g., 'gill-spacing_close', 'ring-type_pendant', 'gill-attachment_free', 'stalk-surface-above-ring_smooth', and 'bruises?_bruises') and with mark 'Pois', when these are inherited from the other metro lines of the first metro map (i.e., 'veil-color_white', 'stalk-shape_tapering', and 'veil-type_partial').

In summary, although all metro stops arisen in the first metro lines led to the edible mushrooms, their mutual combination in the second metro map can determine whether they are edible or not. This means, there is no simple rule for determining the edibility of the mushroom. The mutual combination of metro stops can be inherited either from the same metro lines of the first metro map (like 'gill-attachment_free' and 'stalk-surface-above-ring_smooth') or the other metro lines (like 'gill-size_broad', 'stalk-shape_tapering', and 'veil-color_white'). In the last case, the first two measures emerge as independent metro stops in the second metro map, while the third connects mutually to both the corresponding metro lines.

4.4.2 Iris dataset

A visualization of the best results obtained by the construction of a metro map based on ARM information within the Iris dataset is presented in Fig. 11, from which it can be seen that the corresponding metro map consists of seven metro lines of moderate path length ($\tau = 5$). In this case, metro stops identify physical measurements of the observed

iris plants, like sepal length/width and petal length/width, while the destination metro stop specifies their proper type.

Indeed, the first two metro lines (i.e., ML-1 and ML-2) started with sepal width less than 2.6 cm and petal length between 2.475 and 3.95 cm classifying the Iris versicolor class if the iris plant owns petal width between 0.7 and 1.3 cm. The next two metro lines (i.e., ML-3 and ML-4) describe how to recognize the setosa iris plant, starting with the floor of sepal width between 3.2 and 3.8 cm and the sepal length between 5.2 and 6.1 cm. These metro lines are inter-related, even with three metro stops (i.e., 'petalwidth_(-inf-0.7]', 'petalwidth_(-inf-2.475]', and 'sepalwidth_(-inf-5.2)'). The last three metro lines (i.e., ML-5 to ML-7) indicate that the iris virginica has arisen from a plant with sepal width between 2.6 and 3.2 cm, or petal width more than 1.9 cm, or sepal length more than 7 cm. However, the petal length of these plants must be more than 5.425 cm.

Actually, the Iris dataset contains two clusters: (1) one containing 'Iris setosa', and (2) the other containing 'Iris virginica' and 'Iris versicolor'. Based on this assumption, the first two metro lines classifying the second cluster should be interrelated, but, surprisingly, this fact is not reflected in the mined metro map. Moreover, it turns out that when the species information is used as proposed by Fisher [43], all three classes are linearly separable [44]. Indeed, the advantage of the proposed algorithm for generating metro

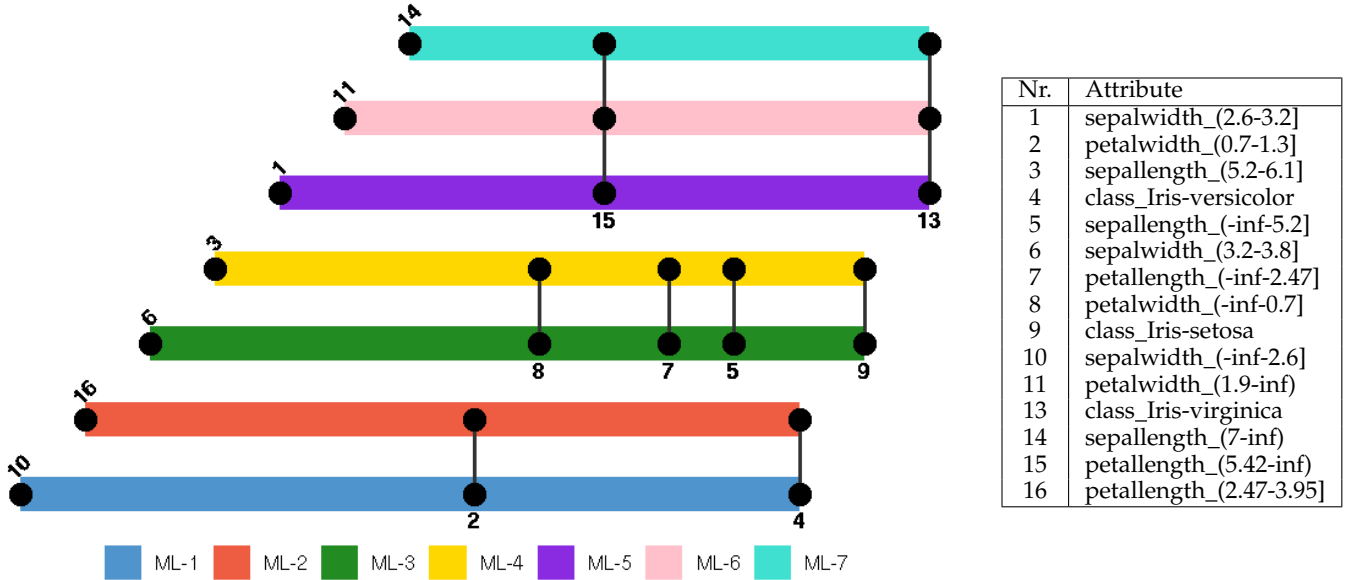


Fig. 11: Visualization of the Iris dataset (path length $\tau = 5$).

maps based on ARM information is that it already considers these clusters as linearly independent.

4.4.3 Abalone dataset

A visualization of the best metro map obtained with the proposed EA by analysis of ARM information within the Abalone dataset are illustrated in Fig. 14. The figure is divided into two diagrams, where the former depicts a metro map with four metro lines (denoted as ML-1 to ML-4) and the latter with two metro maps (denoted as ML-1 and ML-2). As a result, the first metro map is constructed with path length $\tau = 4$, and the second with path length $\tau = 9$. The metro stops denote the physical measurements of the observed abalone (like sex, length, diameter etc.), and connect them into a path that leads towards the destination metro stop determining the number of rings. However, the age of the corresponding mushroom is determined on basis of this number.

Actually, the metro lines ML-3 and ML-4 in the first metro map are depicted as ML-1 and ML-2 in the second one, respectively. These measurements are denoted using mark 'Path-4' in the second metro map. Indeed, the metro line ML-4 in the first metro map determining the metro stops leads to the number of rings between 8 and 15 (i.e., age of 9.5 to 16.5) change direction in the second metro line of the second metro map from the same starting metro stop to the number of rings less than 8 (i.e., age of less than 9.5 years).

From comparison of both the metro maps, it can be concluded that there are two main reasons for changing the age of the abalone from between 9.5 and 16.5 years in ML-4 of the first metro map to the age of less than 9.5 years in the ML-2 of the second metro map: (1) adding two new <feature,attribute> pairs into the ML-2 (i.e., the length of abalone higher than 0.63 mm, and the shucked weight between 0.37 and 0.74 grams), and (2) inheriting the <feature,attribute> pairs from the ML-1 (i.e., height less than 0.28 mm and diameter less than 0.20 mm).

4.4.4 Wine dataset

Visualizing the metro map that emerged on the basis of ARM information within the Wine dataset is depicted in Fig. 17, that is divided into two diagrams, representing the metro maps constructed by path lengths $\tau = 5$ and $\tau = 11$. Thereby, metro stops designate the quantities of chemical constituents found by analyzing, where each destination metro stop determines one of the three types of wines.

Indeed, the first metro map consists of seven metro lines, while the second of only three. The metro lines of the same colors emerging in both metro maps, in some way, reproduce the development of the three different wine types which originated from the same source during the time. The older the wine, the less complex is its chemical structure. This means that once simple chemical structure found in the analysis of the older wine types became more and more complex over time. Interestingly, this trend can be observed in the constructed metro maps.

For instance, the ML-7, ML-3, and ML-1 in the first metro map emerged as ML-1, ML-2, and ML-3 in the second metro map, respectively. Thus, the first four metro lines classify quantities of constituents needed for producing the 'class-3', ML-5 for the 'class-2' and the remaining three metro lines for 'class-1' type of wine. The second metro map incorporates only constituents found in the 'class-3' and 'class-2' types of wine.

Comparison between both metro maps revealed, in which direction the three different cultivars proceeded. Thus, the ML-1 in the second metro map originally describes the chemical analysis needed by producing the 'class-1' type of wine. Although all original chemical constituents, characteristic for 'class_1' can also be found in the new metro map (i.e., color_intensity_(4.21-7.14], alcohol_(12.93-13.88], flavanoids_(2.71-3.89], proanthocyanins(1.99-2.79]), the constituents determining the 'class_3' wine type were inherited from the first three metro lines of the first metro map (i.e., alkalinity_(15.45-20.3], malic_acid_(-inf-2.00], hue_(-inf-

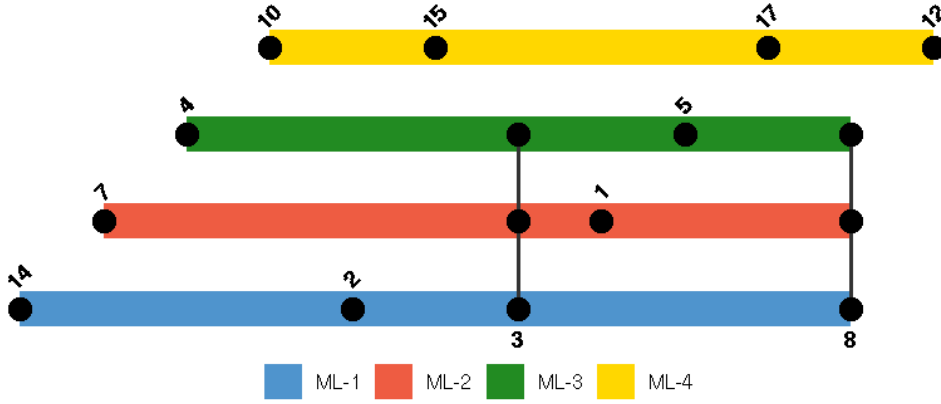


Fig. 12: subfigure
Path length $\tau = 4$.

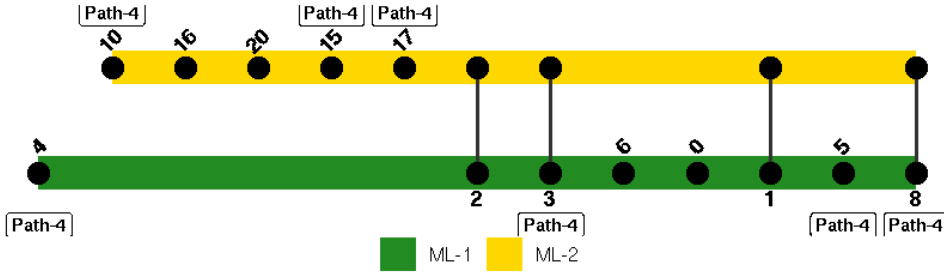


Fig. 13: subfigure
Path length $\tau = 9$.

Nr.	Attribute
0	shell_ $(-\text{inf}-0.25]$
1	whole_ $(-\text{inf}-0.71]$
2	height_ $(-\text{inf}-0.28]$
3	diameter_ $(-\text{inf}-0.20]$
4	length_ $(-\text{inf}-0.26]$
5	viscera_ $(-\text{inf}-0.19]$
6	shucked_ $(-\text{inf}-0.37]$
7	sex_I
8	rings_ $(-\text{inf}-8]$
10	shucked_ $(0.74-1.12]$
12	rings_ $(8-15]$
14	viscera_ $(0.38-0.57]$
15	whole_ $(1.41-2.12]$
16	length_ $(0.63-\text{inf})]$
17	diameter_ $(0.50-\text{inf})]$
20	shucked_ $(0.37-0.74]$

Fig. 14: Visualization of the Abalone dataset.

0.79], proanthocyanins(-inf-1.20]), as well as new ones have emerged (as 19). The ML-2 in the second metro map remains unchanged regarding the destination metro stop and also inherits all three constituents from the ML-3 of the original metro map. Besides the interrelation metro stops with the ML-1, there also emerged two new constituents determining the same class (i.e., proanthocyanins(-inf-1.20], magnesium_ $(-\text{inf}-93]$). The ML-3 suffers much of the modifications regarding ML-1 in the original metro map. At first, the destination of the metro map changes from the original 'class_2' to the 'class_2'. Then, two constituents are inherited from the original 'class_2' (i.e., flavanoids_ $(1.52-2.71]$, flavanoids_ $(-\text{inf}-1.52]$), while the next four shared with the new 'class_3' (i.e., alkalinity_ $(15.45-20.3]$, hue_ $(-\text{inf}-0.79]$, proanthocyanins(-inf-1.20], magnesium_ $(93-116]$). Finally, hue_ $(0.79-1.09]$ and color_intensity_ $(-\text{inf}-4.21]$ are emerged anew.

In summary, the 'class_1' disappears from the second metro map, but some constituents were inherited from it and emerged in the new 'class_3'. On the other hand, the 'class_2' type of wine stemmed from the 'class_3' appearing in this metro map. However, some new constituents were added to this.

4.4.5 Sport dataset

Visualization of the best metro map obtained on the basis of ARM information hidden within the Sport dataset is presented in Fig. 18, from which it can be seen that this map consists of four metro lines of length $\tau = 5$. Each metro line

describes paths from the source to one of five destination metro stops, where the particular metro stop denotes the particular psycho-physical measurement obtained by the observed athlete during a training session.

The metro map is strongly interrelated, due to several connected metro stops. For instance, the first three metro lines ML-1 to ML-3 describe paths from different starting metro stops toward the same destination 'FOOD_FRUITS', reached by even the same interrelated intermediate metro stops. The fourth metro line, on the other hand, leads to the destination metro stop 'BEV_WATER' from the 'WEAT_CLOUD' starting metro stop. Thus, all four metro lines are interrelated by the 'DUR_SHORT' metro stop.

The knowledge hidden in the presented metro map complies with the theory of sport, and is typical for an amateur athlete overcoming the easier stress during training. Actually, the metro map describes the food and beverage needs of an athlete during the realization of sport training sessions of low intensity. In line with this, it holds that the lower calorie food is necessary for easier training of short duration. Additionally, the fitter athlete with no injuries does not need any isotonic beverage by realizing the training sessions of short duration in cloudy weather.

4.5 Discussion

A metro map of information is a new visualization method that has taken inspiration from real metro maps. Metaphorically, like real metro maps help people understand their surroundings, the metro maps help them understand their

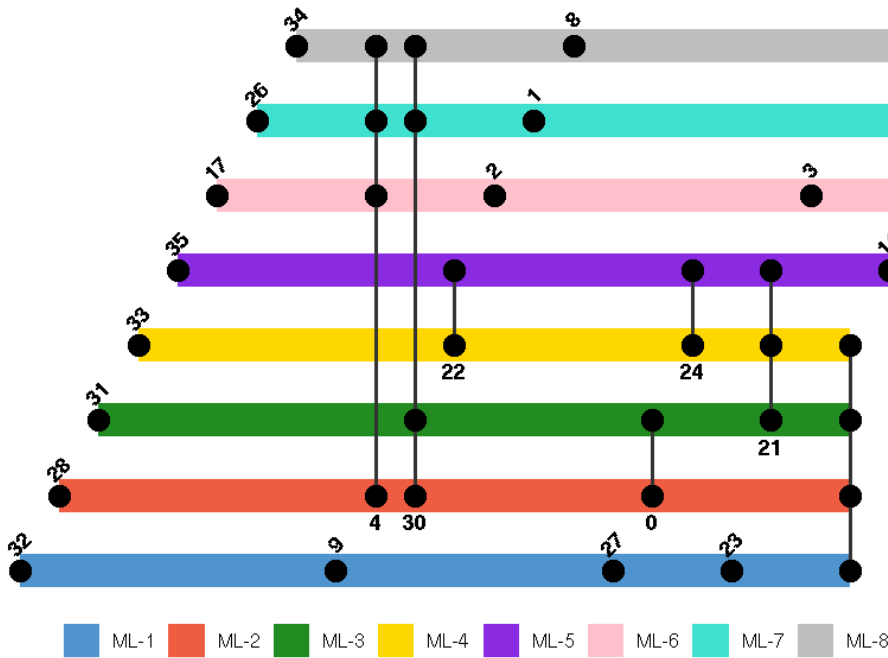


Fig. 15: subfigure
Path length $\tau = 5$.

Nr.	Attribute
0	alcalinity_(15.45-20.3]
1	ash_(2.29-2.76]
2	malic_acid_(-inf-2.00]
3	alcohol_(12.93-13.88]
4	flavanoids_(2.71-3.89]
6	C
7	class_1
8	color_intensity_(4.21-7.14]
9	flavanoids_(1.52-2.71]
10	hue_(0.79-1.09]
11	magnesium_(-inf-93]
12	color_intensity_(-inf-4.21]
14	class_2
17	hue_(1.09-1.40]
19	magnesium_(93-116]
20	total_phenols_(1.70-2.43]
21	total_phenols_(-inf-1.70]
22	flavanoids_(-inf-1.52]
23	hue_(-inf-0.79]
24	proanthocyanins(-inf-1.20]
25	class_3
26	color_intensity_(7.14-10.07]
27	nonflavanoid_phenols_(0.39-0.53]
28	nonflavanoid_phenols_(-inf-0.26]
29	malic_acid_(2.00-3.27]
30	proanthocyanins(1.99-2.79]
31	alcohol_(13.88-inf)
32	alcohol_(-inf-11.98]
33	malic_acid_(3.27-4.53]
34	magnesium_(116-139]
35	nonflavanoid_phenols_(0.53-inf)

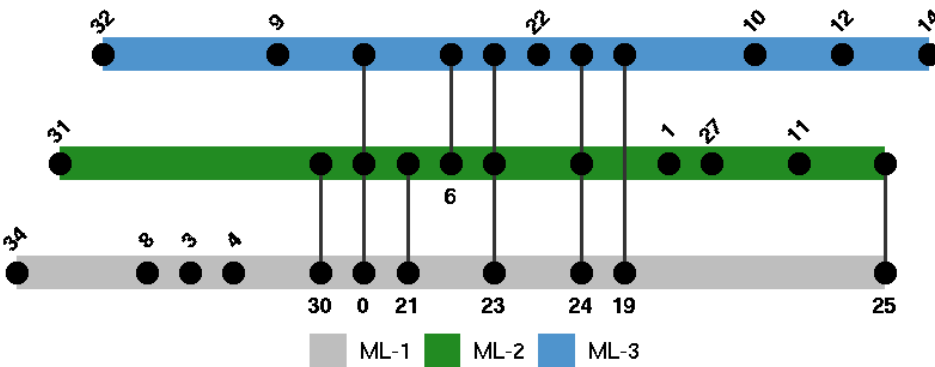


Fig. 16: subfigure
Path length $\tau = 11$.

Fig. 17: Visualization of the Wine dataset.

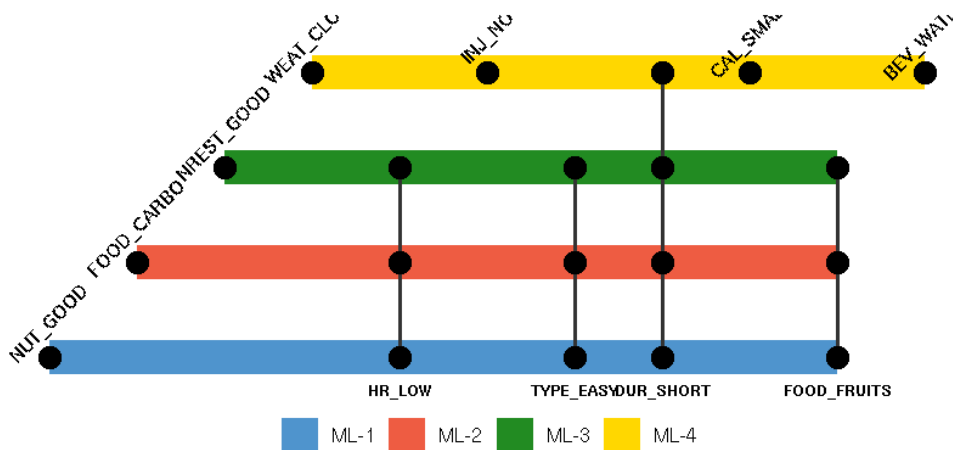


Fig. 18: Visualization of the Sport dataset (path length $\tau = 5$).

information landscape. Moreover, the structure of metro maps is designed so that these can even tell stories to users. Indeed, metro maps consist of metro lines that present different aspects of the same story. The metro lines contain metro stops that represent salient pieces of information. When these pieces interrelate between different metro lines, conflicts arise. These conflicts lead the plot of the story towards its resolution (the final metro stop). On the other hand, each story must have its own introduction (the destination metro stop).

Let us emphasize that the metro map of ARM information presents a generalization of all the association rules in the transaction database and, therefore, ignores the special cases. Consequently, the extracted association rule sequences in metro lines only highlight the most general truths. As a result, it can happen that some conclusion does not hold true for each cases using this method. Despite everything, the method offers a new aspect in the visualization of ARM rules, and can help users direct their attention to the more salient pieces of information.

The core of information cartography in ARM presents the construction of a metro map, which is defined as an optimization problem that has been solved by an EA. The task of the EA is to find the best paths (representing metro lines) within the attribute graph according to predefined objectives that define a structure of the designed metro map. The characteristics of the constructed metro maps are as follows: The shorter metro lines pass the basic truths representing building blocks, when the path lengths become longer. Actually, the building blocks connect together in longer metro lines and represent a complex knowledge hidden inside metro maps. On the other hand, the history of emerging some constituents, quantities of chemical substrates, or psycho-physical measurements can be predicted by evolution from shorter metro lines to the longer. Obviously, this evolution represents a potential idea to form the story with its introduction, plot of the play and conflicts that lead to the final resolution. Indeed, the story tells users even more than many other standard visualizations which can be revealed by our wide experimental work.

Let us see what some of observed metro maps narrate. For instance, the mushroom metro map could help the screenwriter to select the mushrooms with those constituents that are faithful for a character that must die according the screenplay due to poisoning. Abalone is known in cooking as a reach nutritional food resource that is full of high proteins, and low in fat and cholesterol. It represents a source of omega-3 fatty acids having a low risk of heart disease. The economic value of abalone is correlated positively with age [45]. Especially interesting for economy are so-called 'cocktail' abalones whose shell weights generally measure 7-11 cm. In the sense of a restaurant owner serving the healthy dishes, among which are abalones in the first place, it is very important to determine the age of the abalones being sold on the local fish market without expensive experiments by counting the number of rings under the microscope. In line with this, the abalone metro map informs us that physical measurements characteristic for older abalones (like a diameter of more than 0.5 mm and whole weight between 1.41 and 2.12 grams) does not mean that the observed abalone is not the 'cocktail' one.

However, as the stories describe specific situations, our metro maps of ARM information also stem from generalized extracted information that, in specific situations, sometimes does not hold up entirely. Despite this weakness, the technology of information cartography in ARM shows that there are new aspects in extracting the structured knowledge hidden in data and, especially, in transferring this knowledge to the user. The Wine story can undoubtedly be considered in this class.

The Wine story is one of the most fascinating, while the point of the story confirms the power of the selected visualization. Without any knowledge about the real situation that happened in the specific region in Italy, our story, inferred from the corresponding metro map, speculates that there was an original wine type, let's say 'class_1', which passes its good characteristics to its offspring plants in the same cultivar, let's say 'class_3'. This type of wine has been evolved, and by adding the new chemical constituents caused an emerging of the new type of wine, let's say 'class-2'. Finally, we got three types of plants steam from the same ancestor that have been evolved in three different directions, although they grow in the same area under the same conditions.

In summary, the goal of the visualization process using metro maps is not only to present information to the user in an understandable way, but to assemble the constituent salient information pieces of the metro map into the whole. Moreover, this technology is capable of directing the user's decision-making process and simulating exactly what consequences false decisions can have.

5 CONCLUSION

Nowadays, we are confronted with the large-scale creation of unstructured data that are hard to analyze manually. In line with this, a lot of ML methods have arisen, with the purpose of discovering new information hidden within data. One of these methods is also ARM, devoted to discovering the interesting relations between attributes in huge transaction databases. Normally, the algorithms for ARM generate a huge number of association rules collected in datasets in an unstructured form. From these datasets it is not so easy to extract structured knowledge and present this in a form that is automatically appropriate for ordinary users.

This paper proposes a new method for creating metro maps of ARM information automatically (also information cartography) that consist of five steps: creating an ARM dataset, association rule simplification, attribute graph definition, metro map construction, and metro map visualization. Actually, the contribution belongs to the XAI domain that has recently been achieving a notable momentum with the progress of deep learning. As a result, the study revealed that the ARM information cartography is suitable for explaining knowledge hidden in ARM databases on the one hand, and the concept spreads the applicability of the information cartography to the other ML methods on the other. Moreover, the EA for constructing the metro maps was applied primarily to the information cartography, where no, or a less domain specific knowledge, exists. The results on five ARM databases showed that the constructed

metro maps are robust enough for using for explanation purposes in practice.

In the future, information cartography could also be applied to the other UCI Machine Learning datasets. Particularly, its application to the sports domain could be very interesting, where this technology could be used for Interactive Machine Learning (iML) and, thus, help sport athletes optimize their learning behavior during a sport training session through interaction with the proposed information cartography.

ACKNOWLEDGMENTS

Iztok Fister Jr. is grateful for the financial support from the Slovenian Research Agency (Research Core Funding No. P2-0057). Iztok Fister is grateful the financial support from the Slovenian Research Agency (Research Core Funding No. P2-0042 - Digital twin).

REFERENCES

- [1] A. Barredo Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, and F. Herrera, "Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai," *Information Fusion*, vol. 58, pp. 82–115, 2020.
- [2] D. Gunning, "Explainable artificial intelligence (xai)," Defense Advanced Research Projects Agency (DARPA), Tech. Rep., 2017.
- [3] D. Gunning, M. Stefik, J. Choi, T. Miller, S. Stumpf, and G.-Z. Yang, "Xai—explainable artificial intelligence," *Science Robotics*, vol. 4, no. 37, 2019.
- [4] D. Shahaf, C. Guestrin, and E. Horvitz, "Trains of thought: Generating information maps," in *Proceedings of the 21st International Conference on World Wide Web*, ser. WWW '12. New York, NY, USA: Association for Computing Machinery, 2012, p. 899–908.
- [5] D. Shahaf, J. Yang, C. Suen, J. Jacobs, H. Wang, and J. Leskovec, "Information cartography: Creating zoomable, large-scale maps of information," in *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '13. New York, NY, USA: Association for Computing Machinery, 2013, p. 1097–1105.
- [6] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules in large databases," in *Proceedings of the 20th International Conference on Very Large Data Bases*, ser. VLDB '94. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1994, p. 487–499.
- [7] X. Wu, V. Kumar, J. R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, and S. Y. Philip, "Top 10 algorithms in data mining," *Knowledge and information systems*, vol. 14, no. 1, pp. 1–37, 2008.
- [8] M. J. Zaki, "Scalable algorithms for association mining," *IEEE Trans. on Knowl. and Data Eng.*, vol. 12, no. 3, p. 372–390, May 2000. [Online]. Available: <https://doi.org/10.1109/69.846291>
- [9] J. Han, J. Pei, and Y. Yin, "Mining frequent patterns without candidate generation," in *ACM Sigmod Record*, vol. 29, no. 2. ACM, 2000, pp. 1–12.
- [10] J. Mata, J.-L. Alvarez, and J.-C. Riquelme, "Discovering numeric association rules via evolutionary algorithm," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2002, pp. 40–51.
- [11] B. Alatas, E. Akin, and A. Karci, "Modenar: Multi-objective differential evolution algorithm for mining numeric association rules," *Applied Soft Computing*, vol. 8, no. 1, pp. 646–656, 2008.
- [12] I. Fister, I. Fister Jr., and D. Fister, *BatMiner for Identifying the Characteristics of Athletes in Training*. Cham: Springer International Publishing, 2019, pp. 201–221.
- [13] F. Feng, J. Cho, W. Pedrycz, H. Fujita, and T. Herawan, "Soft set based association rule mining," *Know.-Based Syst.*, vol. 111, no. C, p. 268–282, Nov. 2016. [Online]. Available: <https://doi.org/10.1016/j.knsys.2016.08.020>
- [14] F. Feng, Q. Wang, R. R. Yager, J. C. R. Alcantud, and L. Zhang, "Maximal association analysis using logical formulas over soft sets," *Expert Syst. Appl.*, vol. 159, p. 113557, 2020. [Online]. Available: <https://doi.org/10.1016/j.eswa.2020.113557>
- [15] J. Luna, F. Padillo, M. Pechenizkiy, and S. Ventura, "Apriori versions based on mapreduce for mining frequent patterns on big data," *IEEE Transactions on Cybernetics*, vol. 48, no. 10, pp. 2851–2865, Oct. 2018.
- [16] Y. Djenouri, A. Belhadi, P. Fournier-Viger, and H. Fujita, "Mining diversified association rules in big datasets: A cluster/gpu/genetic approach," *Information Sciences*, vol. 459, pp. 117–134, 2018.
- [17] Y. Djenouri, H. Drias, and A. Bendjoudi, "Pruning irrelevant association rules using knowledge mining," *IJBIDM*, vol. 9, no. 2, pp. 112–144, 2014. [Online]. Available: <https://doi.org/10.1504/IJBIDM.2014.065075>
- [18] Y. Djenouri and M. Comuzzi, "Combining apriori heuristic and bio-inspired algorithms for solving the frequent itemsets mining problem," *Information Sciences*, vol. 420, pp. 1 – 15, 2017.
- [19] P. C. Wong, P. Whitney, and J. Thomas, "Visualizing association rules for text mining," in *Proceedings 1999 IEEE Symposium on Information Visualization (InfoVis' 99)*. IEEE, 1999, pp. 120–123.
- [20] H. Hofmann, A. P. Siebes, and A. F. Wilhelm, "Visualizing association rules with interactive mosaic plots," in *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2000, pp. 227–235.
- [21] H.-H. Ong, K.-L. Ong, W.-K. Ng, and E. P. Lim, "Crystalclear: Active visualization of association rules," in *International Workshop on Active Mining AM 2002, in conjunction with the IEEE International Conference on Data Mining ICDM, 2002*.
- [22] A. Appice and P. Buono, "Analyzing multi-level spatial association rules through a graph-based visualization," in *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*. Springer, 2005, pp. 448–458.
- [23] T. Herawan, I. T. R. Yanto, and M. M. Deris, "Smarviz: Soft maximal association rules visualization," in *International Visual Informatics Conference*. Springer, 2009, pp. 664–674.
- [24] M. Hahsler, "arulesViz: Interactive visualization of association rules with R," *R Journal*, vol. 9, no. 2, pp. 163–175, December 2017. [Online]. Available: <https://journal.r-project.org/archive/2017/RJ-2017-047/RJ-2017-047.pdf>
- [25] M. Hahsler and S. Chelluboina, "Visualizing association rules in hierarchical groups," in *42nd Symposium on the Interface: Statistical, Machine Learning, and Visualization Algorithms (Interface 2011)*. The Interface Foundation of North America. Citeseer, 2011.
- [26] B. Jiang, C. Han, and X. Hu, "A finite ranked poset and its application in visualization of association rules," in *2008 IEEE International Conference on Granular Computing*. IEEE, 2008, pp. 322–325.
- [27] I. Fister Jr., A. Galvez, E. Osaba, J. D. Ser, A. Iglesias, and I. Fister, "Discovering dependencies among mined association rules with population-based metaheuristics," in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, 2019, pp. 1668–1674.
- [28] C. L. Carmichael and C. K.-S. Leung, "Closeviz: Visualizing useful patterns," in *Proceedings of the ACM SIGKDD Workshop on Useful Patterns*, ser. UP '10. New York, NY, USA: Association for Computing Machinery, 2010, p. 17–26.
- [29] P. Fournier-Viger, J. C.-W. Lin, A. Gomariz, T. Gueniche, A. Soltani, Z. Deng, and H. T. Lam, "The spmf open-source data mining library version 2," in *Machine Learning and Knowledge Discovery in Databases*, B. Berendt, B. Bringmann, É. Fromont, G. Garriga, P. Miettinen, N. Tatti, and V. Tresp, Eds. Cham: Springer International Publishing, 2016, pp. 36–40.
- [30] L. J. Old, "Information cartography: Using your gis for non-spatial data analysis," in *Proceedings of the GIS 2002 Conference*. Indianapolis, IN, USA: IU School of Informatics and Computing–IUPUI, 2002.
- [31] S. Paling, "Information cartography: A proposed model for access to heterogeneous end-user databases," *Advances in Classification Research Online*, vol. 11, no. 1, pp. 161–162, 2011. [Online]. Available: <https://journals.lib.washington.edu/index.php/acro/article/view/12782>
- [32] M. Castillo, "Thinking in different directions," *American Journal of Neuroradiology*, vol. 35, no. 4, pp. 615–616, 2014. [Online]. Available: <http://www.ajnr.org/content/35/4/615>

- [33] J. Horner, "The extraordinary characteristics of dyslexia," *Perspectives on Language and Literacy*, vol. 34, no. 3, 07 2008.
- [34] D. Shahaf, C. Guestrin, and E. Horvitz, "Metro maps of science." in *KDD*, Q. Yang, D. Agarwal, and J. Pei, Eds. ACM, 2012, pp. 1122–1130.
- [35] D. Shahaf, C. Guestrin, E. Horvitz, and J. Leskovec, "A metro map can tell a story, as well as provide good directions," *Communications of the ACM*, vol. 58, no. 11, pp. 62–73, November 2015.
- [36] I. Fister and I. Fister Jr, "uarmsolver: A framework for association rule mining," *arXiv preprint arXiv:2010.10884*, 2020.
- [37] B. S. Manoj, A. Chakraborty, and R. Singh, *Complex Networks: A Networking and Signal Processing Perspective*, 1st ed., ser. Communications Engineering & Emerging Technology. Pearson Education, Inc., 2018.
- [38] D. B. Fogel, "The advantages of evolutionary computation," in *Biocomputing and Emergent Computation: Proceedings of BCEC97*. World Scientific Press, 1997, p. 1–11.
- [39] A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing*, 2nd ed. Springer Publishing Company, Incorporated, 2015.
- [40] R. Storn and K. Price, "Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces," *J. of Global Optimization*, vol. 11, no. 4, pp. 341–359, dec 1997.
- [41] H.-G. Beyer, "Convergence analysis of evolutionary algorithms that are based on the paradigm of information geometry," *Evolutionary Computation*, vol. 22, no. 4, pp. 679–709, 2014, pMID: 24922548.
- [42] D. Dua and C. Graff, "UCI machine learning repository," 2017. [Online]. Available: <http://archive.ics.uci.edu/ml>
- [43] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of Eugenics*, vol. 7, no. 7, pp. 179–188, 1936.
- [44] A. Gorban, N. Sumner, and A. Zinovyev, "Topological grammars for data approximation," *Applied Mathematics Letters*, vol. 20, no. 4, p. 382–386, Apr 2007.
- [45] M. M. Hossain and M. N. M. Chowdhury, "Econometric Ways to Estimate the Age and Price of Abalone," University Library of Munich, Germany, MPRA Paper 91210, Jan. 2019. [Online]. Available: <https://ideas.repec.org/p/pramprapa/91210.html>