Error Pattern-based similarity analysis of task performance in a virtual training environment: a meta graph clustering approach

Alexandre Pereira de Faria[1], Klaus de Geus[1], Sergio Scheer[1]

[1]*PPGMNE-UFPR*
Centro Politécnico – UFPR, Curitiba, Caixa Postal 19.011, 81.531-980, PR, Brasil
mscfaria@yahoo.com, klaus.de.geus@gmail.com, sergioscheer@gmail.com

Abstract. Intelligent Tutor Systems, Serious Games, and Simulations are user interaction-based instructional technologies capable of adapting to the needs of learners. Tracking and logging data from user interactions during the execution of a task allow for learning assessment and visualization of learner performance, which, in turn, allows for the identification of learner mistakes. Instructional tasks can be mapped as a graph, and paths represent the ordered sequences of task activities. Mining path patterns seek similar strategies and anomalous behaviors of learners in performing the task. In this paper, graph similarity methods are applied to clustering tasks performed in a virtual training system. Feasible paths on the graph represent the expected sequences for task execution, and errors are deviations from them. Sequences of activities performed by the learners correspond to free walks on the graph. Through task rules and reliability analysis, errors in learner walks were extracted and represented by vectors and then clustered in error patterns. A meta clustering analysis of error pattern-based clusterings and similarities clusterings reveals which of the former are closest to the error patterns. Based on the findings achieved, in future work, a new similarity method that is sensitive to error patterns will be proposed.

Keywords: Educational Data Mining, Learning Analytics, Graph Similarity, Meta Clustering, Error Patterns

1 Introduction

The features of interaction and adaptation to learners' needs found in many instructional technologies, such as intelligent tutoring systems, serious games, and simulations, are critical to the effectiveness of the learning process Sottilare2013. In this context, assessment and feedback on learner knowledge states depend on tracking and recording interaction data during the execution of an instructional task [1]. The learners' knowledge state is used to design instruction, for example, recommending reinforcement of specific activities and promoting correction of mistakes made [2]. The increasing use of these ubiquitous and pervasive instructional technologies, followed by the power of telemetry and interaction data storage, is driving research into the analysis of training and education data. Data mining techniques have been applied to both categorization and inference of learning models in virtual training systems [3]. Among these techniques, extracting similar patterns through clustering methods plays a key role in analyzing and visualizing training and education data [4]. Categorizing learners according to their task performance levels is the first step towards adapting the learning process to learner's needs. Taken an instructional task with N activities as a graph, called a task graph, whose vertices are the activities and directed edges map all feasible connections. Tracking a learner's performance corresponds to trace a walk in the task graph. Mining graph patterns can reveal both common strategies and anomalous behavior in the execution of the task [5]. An anomalous behavior, here identified to specific types of errors, is a deviation from the expected paths (Fig. 1). Examples of deviations are the errors in the task execution, such as omission, repetition, the inclusion of activities, an incomplete task, an inappropriate sequence of time or order [6].

This work aims is to evaluate different similarity-based clusterings of learners' walks on a task graph applied to the interaction data of professional electricians during the execution of the task named pedestal insulator replacement in a virtual training system for critical activities in electrical substations [7]. In particular, it seeks to identify which of those clusterings are closest to the patterns of errors committed by the learners. Approximating

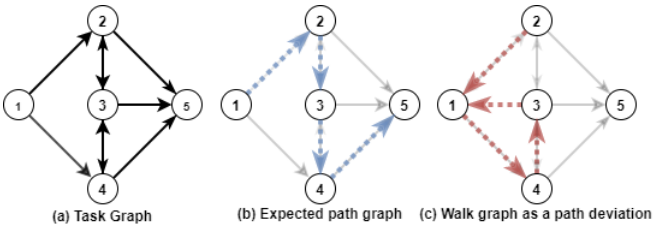

Figure 1. (a) A graph G. (b) A path graph P_G. (c) A walk W_G in G.

similarity to the error patterns can reveal potential graph similarity measures that are topologically sensitive to the path deviations. In the next section, graphs similarity measures used in this work are presented. In the Methods and Materials section, methodological aspects of modeling of error patterns, similarity calculations and clustering methods and validation are exposed. Results are then discussed.

2 Graphs and graph similarities

A graph is a mathematical abstraction of entities and their relationships. A graph $G(V, \mathscr{E})$ is defined by a set of $N = |V|$ vertices $V = \{V_1, ..., V_N\}$ and a set of edges $\mathscr{E} = \{\mathscr{E}_1, ..., \mathscr{E}_M\}$, $M = |\mathscr{E}|$. An edge $\mathscr{E}_k = \mathscr{E}(V_i, V_j)$ is a connection, or adjacency, $i \sim j$ between V_i e V_j. The A_G is the adjacency matrix of G, where $A_{i,j} = 1$, if $i \sim j$, and 0, otherwise. The degree dg_i of the vertex i is the number of edge connections to i. Therefore, the matrix D_G of the vertices degree is a diagonal matrix where $D_{i,i} = dg_i$. Furthermore, the Lapalacian Matrix $L_G = D_G - A_G$ is defined from A_G and D_G is defined . A walk W_G in G is a sequence of connected vertices V and a simple path P_G in G is a sequence of not repeated connected vertices of G [8].

Graph applications such as routing and sequencing problems, anomaly detection in networks, pattern behavior analysis, among others can be addressed as general graph-based pattern recognition problem [9] [10]. Thus, given G and G', two graphs, a similarity measure $Sim(G, G')$ may be defined as $Sim_d(G, G') = \frac{1}{1+d(G,G')}$, where $d(G, G')$ is the distance, or dissimilarity, between G and G' and $Sim(G, G') \in [0, 1]$. As with the G graph, the similarity of paths and walks are defined directly. For instance, given an instructional task, similar paths indicate a pattern in the execution of the task. Let $P = \{P_i\}$ be a set of paths over the graph G. The similarity between two paths i and j could be defined from a distance function $d(P_i, P_j)$. Paths in P are grouped in k categories according to the similarity index from a based-distance similarity matrix $S_{di,j} = Sim_d(P_i, P_j)$. The groups formed must show high similarity between paths in the same group and low similarity between different groups. Representing expected patterns are graph paths ones should ask how similar a user's execution and an optimal performance of activities . The distance functions summarized in Table 1 are available in *NetComp* [11] library designed to compute the similarity between graphs, which runs with another library for graph handling, called *Networkx* [12], both to *Python*.

Table 1. Graph Distances

Distance	Formulation		Inputs		Sensivity
λ_M	$d_{\lambda_M}(G, G') = \left(\sum_{i=1}^{k}(\lambda_{Mi} - \lambda_{M'i})^p\right)^{\frac{1}{p}}$	(1)	M_G e $M_{G'}$, $M = A$ or $M = L$		Global/Local
GED	$d_{GED}(G, G') = \|A - A'\| = \sum_{i,j}^{n} \|A_{i,j} - A'_{i,j}\|$	(2)	A_G e $A_{G'}$		Local
VEO	$d_{VEO}(G, G') = 2\frac{\|V_G \cap V_{G'}\| + \|E_G \cap V_{G'}\|}{\|V_G\| + \|V_{G'}\| + \|E_G\| + \|E_{G'}\|}$	(3)	G, G' e $G \cap G'$		Local
$\delta - con$	$d_{\delta-con}(G, G') = \left(\sum(\sqrt{S_{ij}} + \sqrt{S'_{ij}})\right)^{\frac{1}{2}}$	(4)	$S = [I + \epsilon D - \epsilon A]^{-1}$	(5)	Global/Local
$Resistance$	$d_{NRes}(G, G') = \left(\sum_{i=1}^{k}(R - R')^p\right)^{\frac{1}{p}}$	(6)	$R = diag(L^\dagger)[1]^T + [1]diag(L^\dagger)^T + 2L^\dagger$	(7)	Global
$Netsimile$	$d_{Netsim}(G, G') = \sum \frac{\|s - s'\|}{s + s'}$	(8)	$s =$ *signature* vector of G		Local

Different distance settings can be used depending on whether the goals of the analysis are sensitive (column Sensitivity in Table 1) to local features, such as detecting anomalies in connections, or global features, such as identifying communities between entities in the graph [13]. The *NetComp* library supports different similarity methods divided into spectral, matrix, or vector distances:

1. Spectral or λ distances [14] are based on evaluating the eigenvalues of the adjacency, laplacian, or normalized laplacian matrices of graphs . The spectra of a matrix M is the ordered sequence of its eigenvalues λ_i^M. Let λ_i^M and $\lambda_i^{M'}$, with $i = 1, 2, ..., n$ be the sequence of the eigenvectors of the matrix M and M' of the graphs G and G', respectively. Spectral distances in eq. 1 are sensitive to both global or local divergences between graphs.
2. The matrix-based graph edit distance [15] is calculated from the number of operations required to transform G into G' with minimal cost. Edit operations on graphs include deleting a vertex or edge or add a vertex or edge. The distance in eq. 2 is defined as the difference between the adjacency matrices A and A' of graphs G and G', respectively.
3. The matrix-based vertex-edge overlap distance [16] is calculated from the ratio of the quantity of shared vertices and edges to the sum of vertices and edges (eq. 3) .
4. The $DeltaCon_0$ [17] is an algorithm that compares affinities between vertices. The similarity between two graphs G and G' is calculated from the equation (eq. 4), where S and S' are the vertex affinity matrices between the G and G', D is the diagonal degree matrix of the vertices and A is the adjacency matrix. The columns of the S matrix contain the affinity vector s_i of vertex i which is the solution of eq. 5. The value of ϵ is a small constant associated with the neighborhood of the vertices and e_i is the vector whose component $i = 1$, and 0 otherwise.The equation 5 comes from the *Fast Belief Propagation* method that models information diffusion in a graph.
5. The normalized resistance distance in eq. 6 are inspired by the information diffusion across the graph. The resistance of a graph is calculated based on the analogy with an electric circuit where the edges between two vertices V_i and V_j represent resistors with resistance $\frac{1}{w_{ij}}$[18]. The normalized resistance distance between two graphs G and G' is given by eq. 7, where $diag(L^\dagger)$ is the diagonal matrix of $L^\dagger$, the generalized inverse of the Laplacian matrix and $[1]$ is the matrix with all components equal to 1.
6. Vector distances like the NetSimile (eq. 8) capture the differences between two graphs through their signatures vectors of the statistical measurements of the graph vertices and edges [19] based on characteristics of each vertex with respect to its neighborhood (*egonet features*) such as the quantity and average degree of neighboring vertices.

3 Methods and materials

The analysis shown in the Figure 2 have four phases: first, modeling data in a task graph, learner's walks, and feasible paths; after that, define similarities and error pattern for learners' walks; then, clustering learner's walks according to (a) the walks similarities and (b) the error patterns; and in the last, apply meta clustering analysis to identify similarities clusterings that are closest to the error pattern clustering. In this work, graph generation and manipulation use the *Networkx* package [12], and similarity calculations are carried out from the *Netcomp* library [11], both for *Python*. The application of the *k-means* method, silhouette coefficient calculation, and cluster visualization was achieved from its implementation in the *Scikit-learn* library [20] also for *Python*. The application context of this work is a virtual training system that reproduces an electrical substation in which users can interact with objects in the scene through a 3D immersion headset and joystick controls. Developed using the Unreal engine, the virtual environment supports a guided training of a twenty-activity task, named pedestal insulator replacement [7]. The interaction data used in this work refers to the 22 training sessions available in the system's database. The logs contain information about the type, time, and order of the performed activities in each session.

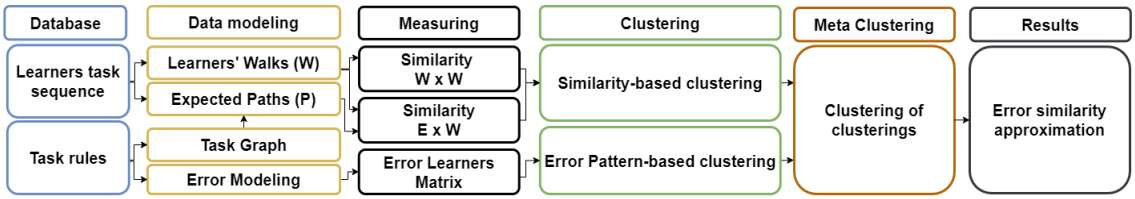

Figure 2. Workflow for meta clustering analysis

Data modeling Data modeling begins from a given task vector V with N activity entries V_j, $j = 1, \cdots, N$. On one hand, the rules for executing the activities are order constraints and define X feasible paths $P = \{P_x\} =$

$P_x(V_{P_x}, \mathscr{E}_{P_x}) = \{V_{P_x 1}, \cdots, V_{P_x N}\}$, with $V_{P_x i} \neq V_{P_x j}$, $\forall x \in X$ which are deduced from task rules. The paths union results in the directed task graph $G_T = (V_G, \mathscr{E}_G) = \bigcup_{i=1}^{X} P_x$, where $V_G = V_P$ and $\mathscr{E}_G = \bigcup_{i=1}^{X} \mathscr{E}_{P_x}$. Rigorously, the task graph is a unweighted multidigraph because their vertices support two directed edges. In the other hand, the sequences performed by the L learners are free walks $\{W_\ell\}$ in the task graph, $\ell = 1, \cdots, L$. Free walk means that an edge of some walk might not have the corresponding edge in the task graph. The order of activity performed by the ℓ-th learner corresponding to the order of the vertices in $W_\ell = \{V_{W_{\ell,1}}, \cdots, V_{W_{\ell,R}}\}$, where R is the total of concluded activities. Besides that, it is assumed that W_ℓ is embedded in G_T such that $|V_{P_x}| = |V_{W_\ell}|, \forall \ell, x$. In this way, a $W \times P$ comparison concerns only to difference among their directed edges.

From [6], the errors made by professional workers in live-line maintenance were identified and mapped for every W_ℓ. The error vectors entries are components signal of the occurrence of six types of errors in the execution of the task activities: improper timing (e_1), repetition (e_2), inclusion (e_3), inversion (e_4), incompleteness (e_5), omission (e_6). The error weight vector $\{e_i\} = \{i\}$, $i = 1, 2, \cdots, 6$ and a function $\phi(V_{\ell j}) = \vec{e}'$, where the j-th position of $\vec{e} = i$ whether error e_i affects activity V_j performed by learner ℓ, and 0, otherwise. For each learner ℓ the error assignment leads to a learner error matrix $E_{\ell|e|\times|V|} = [E_{\ell(e_i,V_j)}]$ (eq. 9), where $E_{\ell(\vec{e},V_j)} = \phi(V_{\ell j})$

$$E_\ell = \begin{bmatrix} E_{\ell(e_1,V_1)} & E_{\ell(e_1,V_2)} & \cdots & E_{\ell(e_1,V_N)} \\ E_{\ell(e_2,V_1)} & E_{\ell(e_2,V_2)} & \cdots & E_{\ell(e_2,V_N)} \\ \vdots & \vdots & \ddots & \vdots \\ E_{\ell(e_6,V_1)} & E_{\ell(e_6,V_2)} & \cdots & E_{\ell(e_6,V_N)} \end{bmatrix} \tag{9}$$

Matrix $E_{\ell|e|\times|V|}$ can be reshaped into a row vector with transpose columns $E_{\ell(e,V_j)}]$ (eq. 10):

$$E_{\ell 1\times|e|*|V|} = E_{\ell full} = [E_{\ell(e,V_1)}]^T, [E_{\ell(e,V_2)}]^T, \cdots [E_{\ell(e,V_N)}]^T] \tag{10}$$

The learners' error matrix is $E = \{E_{\ell full}\}$, $\ell = 1, \cdots, L$

Similarity measuring Two approaches set the matrix similarity computations for the walks: first, the similarities between the learners' walks and the all feasible paths ($W \times P$); second, the similarities between the learners' walks and each other ($W \times W$). Let a graph similarity measure Sim_d, based on a distance d, similarity matrices $[S_{d(W\times P)}] = [Sim_d(W_i, P_x)]$, $i = 1, \cdots, R$ and $j = 1, 2, \cdots, X$ and $[S_{d(W\times W)}] = [Sim_d(W_i, W_i)]$, $i = 1, \cdots, X$ are defined. Fig. 3 presents the similarity plots between the walks and paths for six distances. The similarity patterns of a walk are different for each method. However, dissimilarity patterns between different walks are invariant to methods. For example, the similarity patterns of walks **a** and **b** are closer than the patterns between walks **a** and **c** or **b** and **c**.

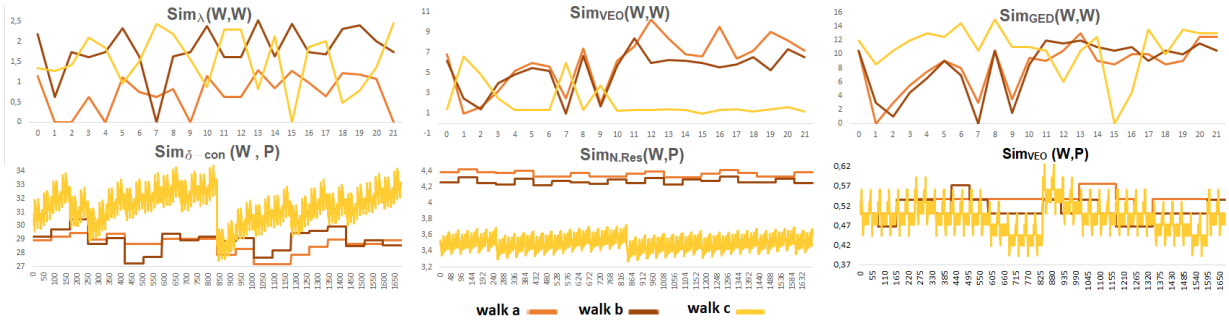

Figure 3. Invariant patterns throughout similarities

Error pattern measuring At this point, the feasible paths represent the standard sequences for performing the task, and a task error is a deviation from one of them. In turn, the sequences of activities performed by the learners

were defined as walks in the graph. The embedded learner errors result in a high-dimensional learning error vector (eq. 10), so for cluster analysis, two reduced forms were applied to the learner error matrices. In both cases, for each learner, a single error vector is assigned, the components of which are either a) the maximum row vector (eq. 11) or, b) the row sum error vector (eq. 12).

$$E_{\ell \max} = \left[\begin{array}{cccc} max(E_{(e_i,V_1)}) & maxE_{(e_i,V_2)}) & \cdots & max(E_{(e_i,V_N)}) \end{array} \right] \tag{11}$$

$$E_{\ell \sum} = \left[\begin{array}{cccc} \sum_{i=1}^{6} E_{(e_i,V_1)} & \sum_{i=1}^{6} E_{(e_i,V_2)} & \cdots & \sum_{i=1}^{6} E_{(e_i,V_N)} \end{array} \right] \tag{12}$$

Clustering Among several clustering methods, the k-means is the simplest and best-known partitional clustering method with a wide range of applications [21]. Implementation of *k-means* in *Scikit-learn* use the optimized *k-means++*. The number of clusters k = 3 was estimated by the Elbow method. Let $W = \{W_\ell\}$, $\ell = 1, \cdots, L$, the learners' walks and $P = \{P_x\}$, $x = 1, \cdots, X$ the expected paths and the learners' error matrix $E = \{E_\ell\}$, $\ell = 1, \cdots, L$, as already defined. The error path similarity matrices $[S_d]$, where $d \in \{d_{\lambda Adj}, d_{\lambda Lap}, d_{GED}, d_{VEO}, d_{ResN}, d_{\delta-con}, d_{Netsim}\}$, are calculated for $[S_{d(W \times P)}]$ and $[S_{d(W \times W)}]$. Given the learners's walks $W = \{W_\ell\}$, $\ell = 1, \cdots, L$, a cluster of walks is a partition $W^{(i)} = \{W_\ell\}$, $\ell \in [1, \cdots, L]$ of W, such that $W = \bigcup W^{(i)}$, $W^{(i)} \neq \emptyset$, $\forall i$ and $W^{(i)} \cap W^{(j)} = \emptyset$, $i, j = 1, 2, 3$ with $i \neq j$.

		Similarities Measures														Error Pattern		
Clusterings		δcon_WP	ResN_WP	VEO_WW	GED_WP	VEO_WP	Ntsim_WP	λ Adj_WP	λ Lap_WP	Ntsim_WW	λ Adj_WW	λ Lap_WW	ResN._WW	δcon_WW	GED_WW	E_max	E_sum	E_full
Clustered walks	Cluster 1	0	0	0	8	0	0	0	0	0	0	0	8	0	0	0	0	0
		1	4	4	12	4	4	1	4	6	5	5	11	4	5	9	5	9
		2	5	5	13	6	5	5	5	13	10	10	12	5	8	13	9	12
		3	6	6	15	10	8	7	6	16	12	13	13	6	13	15	11	13
		4	8	8	16	11	10	10	10	18	13	15	14	8	17	16	13	15
		6	10	10	18	12	13	15	14	19	15	18	16	10	19	17	15	16
		7	15	11	19	15	14	16	15	20	16	19	18	14	20	18	17	17
	Cluster 2	8	17	12	21	16	15	18	17	3	17	1	19	15	21	1	18	18
		9	18	13	1	17	16	19	18	4	18	2	20	17	1	2	19	1
		13	19	14	2	18	17	20	19	5	19	3	21	18	2	3	20	2
		15	20	15	3	1	18	2	1	8	20	4	1	21	3	4	21	3
		17	1	16	4	2	19	4	2	9	1	6	2	1	4	6	1	4
		18	2	17	5	3	20	6	3	10	3	8	3	2	6	7	2	6
		19	3	18	6	7	1	9	7	11	4	9	7	3	7	8	3	7
	Cluster 3	5	7	19	7	9	2	12	9	12	6	11	9	7	9	10	4	8
		10	9	20	9	5	3	13	8	14	8	12	0	9	10	5	6	10
		12	11	21	10	8	6	17	11	15	11	14	4	11	11	11	7	5
		14	12	1	11	13	7	21	12	17	14	21	5	12	12	12	8	11
		16	13	2	14	14	9	3	13	21	21	7	6	13	14	14	10	14
		20	14	7	20	19	11	8	16	1	2	16	10	16	15	19	12	19
		21	16	3	0	20	12	11	20	2	7	17	15	19	16	20	14	20
		11	21	9	17	21	21	14	21	7	9	20	17	20	18	21	16	21
Silhoutte		0,69567	0,666886	0,64832	0,63204	0,60737	0,56943	0,54841	0,51907	0,471128	0,46092	0,41833	0,373935	0,31584	0,24297	0,32466	0,2861	0,24039

Figure 4. Similarities and error pattern clusterings.

Thereby, a total of fourteen clustering derived from the similarity measures are evaluated. From error learners' matrix definition and its reduction forms, error patterns are extracted by clustering the three error learners' matrices $E_{\sum}$, E_{max} and E_{full}. Figure 4 shows clustering results of 22 learners' walks. The colors cells – yellow, gray, and blue –, identify the three clusters in each clustering column. The order of clusterings columns is in agreement with the average Silhouette coefficient in the last row. Silhouette values close to 1 indicate optimal clustering and, -1, the opposite case [22]. The highest silhouette values, around 6.0, wherein similarity was measured between the learners' walks and the expected paths. Another common feature of these clusterings is that they result from matrix-based similarities.

Meta Clustering In the Meta Clustering analysis, clusterings from similarity methods are compared with the clusterings defined from the error pattern [23]. It identifies which similarity methods can be an approximation of clustering learners' walks according to the errors committed during the execution of the task. Taking all 17 clusterings of learners' walks $\overline{W} = \overline{W}_{S_d} \cup \overline{W}_E$. Once more using k-means, meta clustering is applied to $\overline{W}^T$ Figures 5 (a) and (c) present the meta cluster labels distribution of learners' walks in three meta clusters and their plot using projections of principal components from Principal Components Analysis (PCA). In both, clusterings based on error patterns are framed. The average silhouette coefficient of meta clustering is the vertical dashed line

in Fig. 5 (b). The heights of strips is proportional to the number of clusterings in each meta cluster, and the width represents the silhouette coefficient for the cluster and, also, for their respective elements in a continuum range way. Results of meta clustering analysis reveal a low value for the average of silhouette coefficient below 0.3. It means that meta clusters are not dense, as shown in Fig. 5 (c). That can be a consequence of the variability of learners' walks clusters in similarity and error pattern clusterings, already discussed in the maximum occurrence discussion. Despite that, for the first and the second meta clusters, and also for the most of their elements, silhouette coefficients are above the average how is showed in Fig. 5 (b). Error pattern-based clusterings, framed in Fig. 5 (a) and Fig. 5 (c), are some of them.

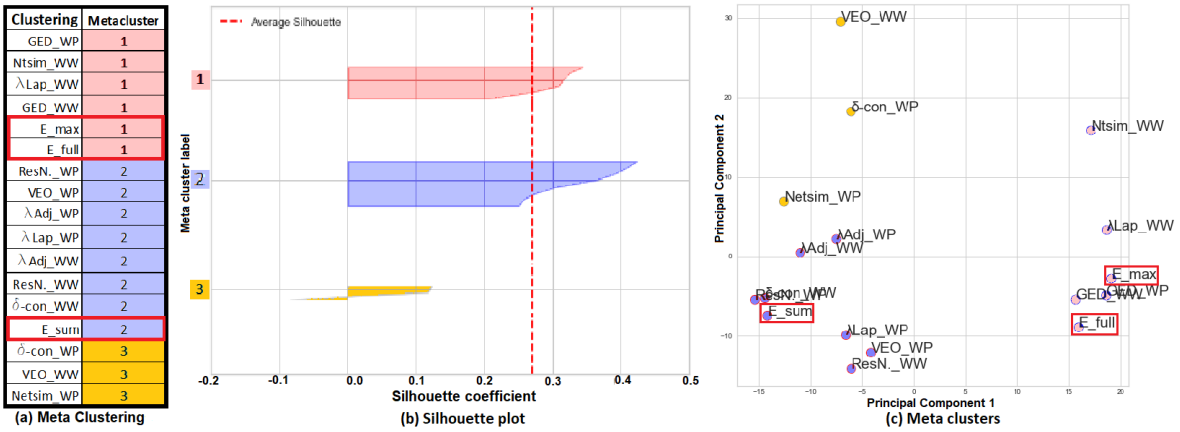

Figure 5. Metaclustering.

The error pattern-based clustering $\overline{W}_{E_{max}}$ and $\overline{W}_{E_{full}}$ are in the meta cluster 1 and $\overline{W}_{E_{sum}}$ in the meta cluster 2. The similarity-based clusterings $\overline{W}_S$ closest to $\overline{W}_{E_{max}}$ and $\overline{W}_{E_{max}}$ in the meta cluster 1 are the Graph Edit Distances $\overline{W}_{S_{GED_{(W\times W)}}}$ and $\overline{W}_{S_{GED_{(W\times P)}}}$. In the meta cluster 2, the normalized resistance similarity, $\overline{W}_{S_{ResN_{(W\times P)}}}$, and the $DeltaCon_0$ similarity, $\overline{W}_{S_{\delta-con_{(W\times W)}}}$ are the nearest clusterings to $\overline{W}_{E_{sum}}$, the row sum error vector of the error learners' walk matrices. Finally, among the 14 similarity schemes tested, only the matrix-based similarities achieved the targeted approximation to the students' error patterns. In addition, the silhouette coefficient of some of the clusters obtained with the matrix-based similarity approach achieved better results than the other similarity schemes. While the graph editing distance, GED, works on topological similarities through graph operations, such as adding and deleting vertices or edges, normalized resistance, and $DeltaCon_0$ comprise the flow information in the graph. As for sensitivity, while GED can detect local changes, Normalized Resistance is applied to global graph divergences. The $DeltaCon_0$ is sensitive to both. The results suggest that topological properties underlie the error pattern in a graph similarity sense.

4 Conclusions

This paper presents a clustering-based graph similarity evaluation applied to learners' walks extracted from a virtual training system database. In this context, data mining techniques and clustering methods provide analytical tools to categorizing learners' behavior during instructional tasks. The recognition of learners' patterns is fundamental to plan instruction, automate learning assessment and deliver real-time feedback to adapt the learning process to the learners' needs. Data modeling encompasses the graph based-modeling of an instructional task, its feasible paths and the errors mapping from the learnears' walks. The walks were clustered based on graph similarity measures and the error pattern. A meta clustering analysis of similarity based clusterings and clusterings of error learners' walks identify similarity measures that are an approximation to the learners' error pattern. Some limitations in this work are related to the small dataset (22 observations), the methods and libraries used in graph similarity and clustering tasks, and should be investigated in future work. The main results show that matrix-based graph similarities such as edit distance, $DeltaCon_0$, and normalized resistance capture similarities between two walks in a task graph that resemble clustering based on error patterns. By taking topological changes in an expected path as a deviation, a learner's walk might be generated from fundamental deviations. Therefore, a deeper understanding of deviation-related path changes must be taken into account to create a new error-based graph similarity measure.

Acknowledgements. This work was developed by the OneReal Research Group, R&D project PD-06491-0299/2013 proposed by Copel Geração e Transmissão S.A., under the auspices of the R& D Programme of Agência Nacional de Energia Elétrica (ANEEL), Brazil.

Authorship statement. The authors hereby confirm that they are the sole liable persons responsible for the authorship of this work, and that all material that has been herein included as part of the present paper is either the property (and authorship) of the authors, or has the permission of the owners to be included here.

References

[1] J. Gobert, M. Pedro, R. Baker, E. Toto, and O. Montalvo. Leveraging educational data mining for real time performance assessment of scientific inquiry skills within microworlds. *Journal of Educational Data Mining*, vol. 4, pp. 111–143, 2012.
[2] S. Ohlsson. Learning from error and the design of task environments. *International Journal of Educational Research*, vol. 25, n. 5, pp. 419–448, 1996.
[3] K. Chrysafiadi and M. Virvou. Student modeling approaches: A literature review for the last decade. *Expert Systems with Applications*, vol. 40, n. 11, pp. 4715–4729, 2013.
[4] M. T. Rodrigo, E. A. Anglo, J. O. Sugay, and R. S. J. D. Baker. Use of unsupervised clustering to characterize learner behaviors and affective states while using an intelligent tutoring system. In *International Conference on Computers in Education*, pp. 57–64. Citeseer, 2008.
[5] C. Lynch, T. Barnes, L. Xue, and N. Gitinabard. Graph-based educational data mining. In *Proceedings of G-EDM 2017*, 2017.
[6] D. Scherer, M. F. Q. Vieira, and J. A. N. Neto. Human error categorization: An extension to classical proposals applied to electrical systems operations. In *IFIP Advances in Information and Communication Technology*, pp. 234–245. Springer Berlin Heidelberg, 2010.
[7] K. Geus, R. Beê, V. Corrêa, R. Santos, A. Faria, E. Sato, V. Swinka-Filho, A. Miquelin, S. Scheer, P. Siqueira, W. Godoi, M. Rosendo, and Y. Gruber. Immersive serious game-style virtual environment for training in electrical live line maintenance activities. In *Proceedings of the 12th International Conference on Computer Supported Education*. SCITEPRESS - Science and Technology Publications, 2020.
[8] V. Vasiliauskaite. *Paths and Directed Acyclic Graphs*. PhD thesis, Imperial College London, 2020.
[9] L. Zager and G. C. Verghese. Graph similarity scoring and matching. *Applied Mathematics Letters*, vol. 21, n. 1, pp. 86–94, 2008.
[10] C. C. Aggarwal and H. Wang, eds. *Managing and Mining Graph Data*. Springer US, 2010.
[11] P. Wills. Netcomp v 0.2, 2017.
[12] A. Hagberg, D. Schult, and P. Swart. Networkx 2.5, 2020.
[13] P. Wills and F. G. Meyer. Metrics for graph comparison: A practitioner's guide. *PLOS ONE*, vol. 15, n. 2, 2020.
[14] Z. Stanić and I. Jovanović. Spectral distances of graphs based on their different matrix representations. *Filomat*, vol. 28, pp. 723–734, 2014.
[15] R. R. Martin. The edit distance in graphs: Methods, results, and generalizations. In *Recent Trends in Combinatorics*, pp. 31–62. Springer International Publishing, 2016.
[16] P. Papadimitriou, A. Dasdan, and H. Garcia-Molina. Web graph similarity for anomaly detection. *Journal of Internet Services and Applications*, vol. 1, n. 1, pp. 19–30, 2010.
[17] D. Koutra and C. Faloutsos. *Individual and collective graph mining : principles, algorithms, and applications*. Morgan & Claypool Publishers, San Rafael, California, 2018.
[18] D. J. Klein and M. Randic. Resistance distance. *Journal of Mathematical Chemistry*, vol. 12, n. 1, pp. 81–95, 1993.
[19] M. . Berlingerio, D. Koutra, T. Eliassi-Rad, and C. Faloutsos. Netsimile: A scalable approach to size-independent network similarity. *arXiv preprint arXiv:1209.2684*, vol. , 2012.
[20] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
[21] C. Aggarwal. *Data Clustering : Algorithms and Applications*. CRC Press, Hoboken, 2013.
[22] P. J. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53–65, 1987.
[23] R. Caruana, M. Elhawary, N. Nguyen, and C. Smith. Meta clustering. In *Sixth International Conference on Data Mining (ICDM'06)*, pp. 107–118. IEEE, 2006.