Clustering of Paths in Complex Networks

Mareike Bockholt and Katharina A. Zweig

Abstract While network analysis is more than 70 years old, the analysis of paths in complex networks is yet almost negligible. Here, we introduce different measures of computing the pairwise similarity of paths, either simply based on the elements in the paths, their sequence, on the graph in which they are embedded, or incorporating all three features. Based on ground-truth in a data set concerning how people solve a one-player puzzle, we show that the classification of the paths using the similarity measures in a hierarchical clustering approach performs best for the similarity measures which integrate all three features. We thus give first evidence that path similarity measures provide another dimension to mine and analyze complex networks.

1 Introduction

The analysis of complex networks has become a large and active field in which a broad variety of results has been published. In many cases, entities use the network as environment and move from node to node. The most obvious example is human navigation in spatial networks, travels in a transportation network, users surfing the WWW, but also game players exploring the problem space of the game, or students using an e-learning environment by following different paths through interlinked documents and media. In all these examples, the entities move on paths (or trails or walks) through the network which are usually neither the shortest path nor totally random (we will use the term *path*, if not explicitly stated otherwise, it includes walks and trails). But while there has been research concerned with human mobility patterns in a broad sense [4, 6], there has been almost no work which considers the actual *paths* taken. Consider for example the network shown in Figure 1 which shows which paths humans have taken in it. All humans navigating in this network started in the leftmost node and aimed at reaching the nodes in the bottom-right corner

Mareike Bockholt (e-mail: mareike.bockholt@cs.uni-kl.de)✉ · Katharina A. Zweig (e-mail: zweig@cs.uni-kl.de)✉
Graph Theory and Complex Network Analysis Group, University of Kaiserslautern, Germany

H. Cherifi et al. (eds.), *Complex Networks & Their Applications V*,
Studies in Computational Intelligence 693,
DOI 10.1007/978-3-319-50901-3_15

(a) (b) Problem space and human navigation

Fig. 1: (a) An example for a *Rush Hour* board. The red car needs to be removed from the board. A legal move consists of horizontal (vertical) move of one horizontally (vertically) placed car. (b) Each node represents one state of a puzzle and two states are connected by an edge if there is a legal move between them; some states represent the solution of the puzzle. The width of an edge is proportional to the number of users that made this move. Paths from a distinct starting state of the puzzle are called *solving* when they reach one of the states representing the solution of the puzzle.

of the picture. The thickness of the edges corresponds to the number of humans who used this edge in their path. It is astonishing that there are some paths in the network which are used more often than others although they are not necessarily the shortest ones. A human eye can also recognize that there are some paths which are more similar to each other than others. Also in other cases, it makes sense not to treat every path as a single path, but to find groups of similar paths and use these groups for further analysis. This can help to find common or distinguishing patterns in the paths and reduces the large amount of taken paths into representative groups. If such a clustering procedure is able to partition given paths into groups such that the paths within one group share elementary structural commonalities, it can be used in different application scenarios. By clustering paths of students in an e-learning environment, one might be able to identify different learner types and structure the materials accordingly. Grouping paths of players solving a puzzle can be used to find different strategies to solve the game. Clustering paths in a road network can lead to a procedure for identifying different means of transportation.

However, such a clustering requires a similarity measure. A similarity measure needs to be able to incorporate the most essential information contained in a path and weight them in an appropriate way. Therefore, the question arises of how to quantify the similarity of paths. It is surprising that there has been no approach proposed to measure the similarity of paths in complex networks and to group paths by similarity. Thus, in this paper, we: (i) provide seven first similarity measures for paths in networks which are either based on the elements contained in the paths, or on their sequence, on their embeddedness in the network, or on all three features, (ii) compute the proposed similarity measures for all pairs of paths of a benchmark data set with more than 13000 paths from 20 different networks (of the same kind), and (iii) for each of the networks, we cluster all paths with a hierarchical clustering approach with each of the proposed measures, and (iv) evaluate the results with

respect to a property of the paths that we set as ground-truth. It is crucial to note that this work does not the aim at developing a classifier that partitions the paths according to the ground truth. This could be easily achieved by using other path-features or external features. The main goal is rather to evaluate the proposed similarity measures whether they are able to distinguish between structurally different paths.

The article is hence structured as follows: Section 2 gives an overview of research from other fields. Seven similarity measures for paths are introduced in Section 3. Section 4 gives the details of our approach for clustering paths, including the used data set (Sec. 4.1), the used ground truth and evaluation methods (Sec. 4.2), and the results (Sec. 4.3). Section 5 summarizes the findings of the article.

2 Related Work

While we know of no articles that proposed a similarity measure of paths in a complex network using their embeddedness in it, work that is related to the presented can be found in several different areas of research: In applications like video surveillance systems, it is desirable to track moving objects through consecutive video frames and to extract their trajectories. In order to automatically recognize anomalous movements of objects, a system needs to be able to distinguish between regular and anomalous trajectories. For this reason, there are several approaches how to compare and group trajectories of moving objects [1, 3, 15, 19]. The most often used similarity measures are the length of the longest common subsequence [3, 19] and the Hausdorff distance [12]. In the analysis of trajectories created from tracking moving individuals by (GPS) sensors, the Frchet distance has been extensively studied and applied [7], for example for detecting recurring patterns in trajectories [2]. In the context of web mining, it is beneficial to cluster similar user web sessions, for example for commercial or didactic interest, which is why there are several approaches to cluster sequential data. While Wang and Zaïane propose a clustering method for web sessions based on sequence alignment [20], Kumar proposes a new similarity metric for sequential data [13]. For comparing general sequential data, Moen, Mannila and Das presented several approaches [16, 17, 18] which use a measure similar to the longest common subsequence and eventually incorporates the similarity of the contained events themselves. Clustering of sequences has also been applied in order to make predictions, for example by Laasonen on routes of mobile phone users [14]. However, although some of these approaches can be adapted to paths, they do not consider the complex network in which the paths are embedded in. Taking into account the underlying complex networks is additional information which—as we will show in the following—will yield better results when finding groups of similar paths. Additionally, a systematic evaluation of possible similarity measures of paths has been not provided yet.

3 Similarity Measures for Paths

Definitions Let $G = (V,E)$ with $V = \{v_1,\ldots,v_n\}$ and $E \subseteq V \times V$ denote a simple, connected, undirected, and unweighted graph. We define a path P in G as finite sequence $P = (p_1, e_{p_1}, p_2, \ldots, p_{\ell-1}, e_{p_{\ell-1}}, p_\ell)$ with $p_i \in V$ for all $i \in \{1,\ldots,\ell\}$ and $e_{p_i} = (p_i, p_{i+1}) \in E$ for all $i \in \{1,\ldots,\ell-1\}$. Note that we do not require the edges or nodes of a paths to be distinct. Some authors would thus call P a walk. Since the considered graphs are simple, a path is uniquely determined by its node sequence and the notation can be simplified to $P = (p_1, p_2, \ldots, p_\ell)$ which is used in the following. Let $V(P) = \{p_1,\ldots,p_\ell\}$ and $E(P) = \{e_{p_1},\ldots,e_{p_{\ell-1}}\}$ denote the set of nodes and edges which are contained in a path P, respectively. The length $|P| = \ell - 1$ of a path P is defined as the number of (not necessarily distinct) edges. It holds that $|P| \geq |E(P)|$. Furthermore, let $I(P) = \{1,\ldots,\ell-1\}$ be the set of node indices of path P. For two nodes $v,w \in V$, we define the distance of v and w as the length of the shortest path between v and w. If there is no path from v to w, it is set $d(v,w) := \infty$. In the remainder of this article, we assume that G is a connected graph, hence $d(v,w) < \infty$ for all $v,w \in V$. For a path P and a node $v \in V$, we define the distance of v and P as $d(v,P) = \min\left\{d(v,w) \,\middle|\, w \in V(P)\right\}$.

In the following, we assume that we have a graph G and a set of paths $\mathcal{P}(G)$ of valid paths in that graph. The research question is how to cluster these paths into coherent groups, given a suitable similarity measure $\sigma : \mathcal{P}(G) \times \mathcal{P}(G) \to \mathbb{R}$. In order to derive meaningful similarity and distance measures for paths, the most essential information contained in them needs to be determined. There are three obvious pieces of information contained in any path: (i) the elements contained in the paths, i.e., its nodes and edges, (ii) the order of the contained elements, and (iii) the position of the contained elements in the graph, i.e., their distance to the elements of the other path. Thus, as a first approach to determine the similarity of two paths, they can either be modeled as sets and existing measures for comparing sets can be used, or they can be modeled as sequences and existing measures for comparing strings or sequences can be used. Finally, paths can be considered as objects in the network, which allows incorporating the distance of the path's nodes in the graph into the similarity measure.

Element-based measures If a path is represented as a set of nodes or as a set of edges, well-known similarity measures for sets can be used, such as the number of common nodes or edges, or—as its normalized version—the Jaccard index [9]. The measures *(normalized) node set similarity* σ_{nss} (σ^N_{nss}) and *(normalized) edge set similarity* σ_{ess} (σ^N_{ess}) for two given paths $P,Q \in \mathcal{P}(G)$, are then defined accordingly (cf. Table 1).

Order-based measures If a path is understood as a sequence of nodes, similarity measures for sequences can be used, for example the *longest common subsequence* of the two paths [8]. For a path $P = (p_1, p_2 \ldots p_{\ell-1} p_\ell)$, a subsequence of P is defined as any sequence of nodes which can be obtained by deleting nodes from P. Note that a subsequence of a path in a graph is not necessarily a valid path in that same graph anymore. For two paths P, Q, let $lcs(P,Q)$ denote the length of their longest common subsequence. The corresponding *LCS similarity* σ_{lcs} is as defined in Table 1, the

normalized similarity measure is obtained by dividing $lcs(P,Q)$ by the length of the longer path (see Table 1).

$\sigma^{(N)}_{nss}$	$\lvert V(P)\cap V(Q)\rvert$	$\frac{\lvert V(P)\cap V(Q)\rvert}{\lvert V(P)\cup V(Q)\rvert}$	
$\sigma^{(N)}_{ess}$	$\lvert E(P)\cap E(Q)\rvert$	$\frac{\lvert E(P)\cap E(Q)\rvert}{\lvert E(P)\cup E(Q)\rvert}$	
$\sigma^{(N)}_{lcs}$	$lcs(P,Q)$	$\frac{lcs(P,Q)}{\max\{\lvert P\rvert,\lvert Q\rvert\}+1}$	$lcs(P,Q)$ length of longest common subsequence of P,Q
$\delta^{(N)}_{sad}$	$\sum_{i\in I(P)} d(p_i,q_{G(i)})$	$\frac{\delta_{sad}(P,Q)}{\ell}$	G_{sad} identity function, $\lvert P\rvert=\lvert Q\rvert=\ell-1$
$\delta^{(N)}_{mad}$	$\begin{cases}\sum_{i=1}^{\ell} d(p_i,Q) \text{ if } \ell>k\\ \sum_{i=1}^{k} d(q_i,P) \text{ if } \ell<k\\ \min\{\sum_{i=1}^{\ell} d(p_i,Q),\sum_{i=1}^{k} d(q_i,P)\}\end{cases}$	$\frac{\delta_{mad}(P,Q)}{\max\{\lvert P\rvert,\lvert Q\rvert\}+1}$	$G_{mad}(i)=j$ s.t. $d(p_i,q_j)$ minimal, $\lvert P\rvert=\ell-1$, $\lvert Q\rvert=k-1$
$\delta^{(N)}_{comappa1}$	$\min_{G\in\mathcal{G}_{comappa1}}\left\{\sum_{i\in I(P)} d(p_i,q_{G(i)})\right\}$	$\frac{\delta_{comappa1}(P,Q)}{\max\{\lvert P\rvert,\lvert Q\rvert\}+1}$	$\lvert P\rvert\geq\lvert Q\rvert$, $\mathcal{G}_{comappa1}(P,Q)$ set of surjective and order-preserving functions $G: I(P)\to I(Q)$
$\delta^{(N)}_{comappa2}$	$\min_{G\in\mathcal{G}_{comappa2}(P,Q)}\left\{\sum_{(i,j)\in G} d(p_i,q_j)\right\}$	$\frac{\delta_{comappa2}(P,Q)}{\max\{\lvert P\rvert,\lvert Q\rvert\}+1}$	$\mathcal{G}_{comappa2}(P,Q)$ set of left-total, right-total, order-preserving relations $G\subseteq I(P)\times I(Q)$

Table 1: Definitions of the similarity and distance measures for paths P,Q. σ and σ^N denote unnormalized and normalized measure in the first and second columns, respectively, similarly for distance measures δ.

Position-based measures While the previously proposed similarity measures only take into account nodes or edges contained in the paths or their order, we also propose four measures which consider the position of the paths in the network. The motivation is that even two paths that do not share a single edge can be close or distant within the graph they are embedded in. For example, if two people drive from the same city to the same other city, but one on a highway and one on country roads next to the highway, the two paths should be rated as more similar than if one drives from north to south and the other from east to west. The idea of the following measures is, thus, to calculate the distance in the graph from each node in P to a corresponding node in Q and to calculate the average of these node distances. A position-based distance measure for two paths P and Q is defined as $\delta(P,Q)=\sum_{i\in I(P)} d(p_i,q_{G(i)})$ for a mapping function $G: I(P)\to I(Q)$ which determines the counterpart for each node. The main problem is to find the appropriate counterpart of each node. A first naive proposal for G constraints the distance measure to paths with equal length and matches the i-th nodes of the paths with each other. For two paths P,Q with

$|P| = |Q| = \ell - 1$, G is set to $G_{sad}(i) = i$ for all $i \in \{1, \ldots \ell - 1\}\}$. This yields the *(normalized) simple average distance* as defined in Table 1. The simple average distance is a distance metric, but has two main deficiencies: it is only applicable to paths of equal length, and the matching function G might not be a good choice in many cases. For these reasons, we also consider the *matched average distance* which matches each node of P onto the node of Q which is closest by its graph theoretic distance. Since it seems reasonable to map each node of the longer path onto a node of the shorter path, we get for two paths P and Q with $|P| = \ell - 1$ and $|Q| = k - 1$ the measure δ_{mad}, as defined in Table 1. The normalized matched average distance δ^N_{mad} is obtained by dividing by the length of the longer path. For this distance measure, the corresponding mapping function is thus $G_{mad}(i) = j$ such that $d(p_i, q_j)$ is minimal. Note that with this mapping, it might happen that there are nodes in the shorter path which are not matched at all, although it is the shorter path of the two. Furthermore, while the simple average distance takes into account the order of the nodes in the path by the restrictive mapping G_{sad}, this quality is lost by weakening the restrictions to the node mapping. By mapping each node of P onto its *closest* node in Q (or vice versa), the mapping allows for example that the last node of P is mapped onto the first node of Q. It follows directly that this measure does not satisfy coincidence since two paths with identical node sets, but where the nodes occur in different order will have a matched average distance of 0 although they are not identical.

In order to avoid this, we require G to be a surjective function which considers the order of the nodes: we say that $G : I(P) \to I(Q)$ is *order-preserving* if for all $i, i' \in I(P)$, it holds that $i \leq i' \Leftrightarrow G(i) \leq G(i')$. Let $\mathcal{G}_{comappa1}(P, Q)$ be the set of all functions $G : I(P) \to I(Q)$ with these properties. The corresponding distance measure called *(normalized) CoMapPa1 distance* $\delta_{comappa1}$ (for COnsecutive MAPping of PAths) is then obtained by taking the least expensive of these mappings (see Table 1). Note that $\mathcal{G}_{comappa1}(P, Q) = \emptyset$ if $|P| < |Q|$. A dynamic programming approach can be used to compute this measure in $\mathcal{O}((|P| - |Q| + 1) \cdot |Q|)$ assuming that the graph distances are precomputed.

The last distance measure to be introduced is a refinement of the CoMapPa1 distance leading to the CoMapPa2 distance measure. The CoMapPa1 distance measure exhibits an asymmetry because the longer path (P) is mapped onto the shorter path (Q): while each node of P is mapped onto exactly one node of Q, several nodes of P may be mapped onto one node of Q. In order to fix this issue, let $\mathcal{G}_{comappa2}$ be the set of all *relations* $G \subseteq I(P) \times I(Q)$ which are left-total, right-total, and order-preserving (where a relation G is *order-preserving*, if for all $(i, j), (i', j') \in G$, it holds that $i \leq i' \Leftrightarrow j \leq j'$). The corresponding distance measure, i.e., the *(normalized) CoMapPa2 distance* $\delta_{comappa2}$ ($\delta^N_{comappa2}$), is then defined as in Table 1. For two paths P and Q, this measure can be computed in $\mathcal{O}(|P| \cdot |Q|)$ using a dynamic programming approach, assuming the graph distances are precomputed.

Having these seven similarity and distance measures at hand, a data set of more than 13000 paths in 20 different networks is used to evaluate the proposed measures and give the proof of concept that clustering paths into groups is a viable way of mining complex networks.

4 Using the Measures for Clustering Paths

In Section 3, seven similarity (and distance) measures for paths are proposed (we will stick to the term *similarity measure*, if not explicitly stated otherwise, this term includes also the position-based measures although they are distance measures). The following approach clusters paths of a given data set by a hierarchical clustering approach, separately for each of the proposed similarity measures. We will give evidence that the similarity measure which incorporates information of the underlying complex network and the order of the nodes in the paths, i.e., the CoMapPa2 distance yield the most intuitive results for finding functional groups of paths. We start by providing information about the used data set before the method, the evaluation scheme, and the results are described.

4.1 Data

The networks of the data set are problem spaces of a board game such that the paths represent solutions of players. We consider the board game *Rush Hour* (invented by Nob Yoshigahara, distributed by ThinkFun Inc. and HCM Kinzel (Germany)) which is a one-player block sliding puzzle (see Figure 1a). It takes place on a board of 6×6 cells with one designated exit on which blocks are placed horizontally or vertically which represents a parking lot with parking cars. The blocks can have a length of 2 or 3 cells and a width of 1 cell. The goal of the game is to find a sequence of moves which allows a particular car to exit the board through the designated exit. A legal move is to move a car an arbitrary number of cells forwards or backwards, but not sideways. We call the exact positions of all cars a *configuration* of the game. We generate a graph $G^c = (V^c, E^c)$ from a *Rush Hour* start configuration c by taking all configurations reachable from the start configuration by legal moves as node set V^c, and the legal moves between them as edge set E^c. This graph is called the problem space associated to configuration c. We consider a *Rush Hour* game instance as solved when the cars on the board are in such positions that the particular car can be removed from the board with one additional move. We call such configurations *solution states*. With the concept of the problem space, solving a *Rush Hour* game instance can be understood as finding a path from c to a solution state. Such a path is called a solving path. In the optimal case, the found path is as short as possible.

Source The data set used for analysis was collected by Pelánek and Jarušek [11] who developed a *problem solving tutor* (available under tutor.fi.muni.cz) which is a web-based tool for learning by problem solving and is used in educational contexts. A detailed description is provided by Jarušek [10]. Among others, the system contains *Rush Hour* game instances of different degrees of difficulty. Twenty exemplary configurations with a sufficient amount of played paths were selected for analysis. Let $\mathcal{C}$ denote this set of start configurations of the game instances. The data set contains the log data of all users of the system how they solved (or attempted to solve) the instances. It is important to note that users can also skip to the next game, if they feel they cannot solve the puzzle (or lose interest).

Preprocessing For each configuration $c \in \mathscr{C}$, the associated problem space G^c is computed[1] The problem spaces of the selected games are of the order of several thousands of nodes each. Any user who attempts to solve a game instance creates a path in the problem space of the configuration. For each user, each configuration and each attempt, the generated path is extracted from the log data. Any move which is done after a solution state was reached is not considered anymore, but the path is considered as solving path. Let $\mathcal{P}_c$ denote the set of extracted paths for the configuration c. The table available under the given link also contains for each configuration how many paths were extracted (between 156 and 2934 paths) as well as the information of how many nodes of the problem spaces were actually visited by any of the players. Surprisingly, in average only 10 % of the nodes were visited by at least one player.

Clustering For each of the configurations, for all pairs of paths from $\mathcal{P}_c \times \mathcal{P}_c$, all of the seven similarity measures are computed. For computing the simple average distance, the paths were cut to equal length for each configuration. However, in preceding studies for evaluating all similarity measures on the paths cut to equal length, the simple average distance has less promising results than the other distance measures. Thus, and because the simple average distance will be too restrictive for any application, the results for the simple average distance are omitted, and we only discuss the analysis of the complete uncut paths. The values of all unnormalized measures were scaled to the interval $[0,1]$, the values of the similarity measures were then transformed by $1-\sigma^{(N)}(P,Q)$ to result in a distance measure. For each configuration, the matrices with the similarity values for all pairs of paths are the input for an hierarchical clustering algorithm with either complete, average linkage methods or by Ward's clustering criterion [21]. The results for all three clustering methods show the same qualitative results and differ very little quantitatively; we thus only discuss the results of the clustering with complete linkage.

4.2 Ground Truth and Evaluation of the Results

For interpreting the results of the clustering procedures and to evaluate the different similarity and distance measures for paths, an evaluation criterion is necessary. For this, we use a very simple ground truth: a clustering procedure with an appropriate similarity measure as input should be able to distinguish between solving and non-solving paths. It is important to note that the goal of this work is not the development of a classifier which is able to distinguish between solving and non-solving paths. This could be done easily by other methods. The primary aim is to evaluate the presented similarity measures whether they are able to distinguish between structurally similar and dissimilar paths. In order to evaluate this, the semantic feature of the paths of being solving or non-solving is used: a well-

[1] A detailed description of the data set and the problem spaces can be found online under http://gtna.cs.uni-kl.de/en/gruppe/bockholt/PDFs/CN2016SupplementaryMaterial.pdf.

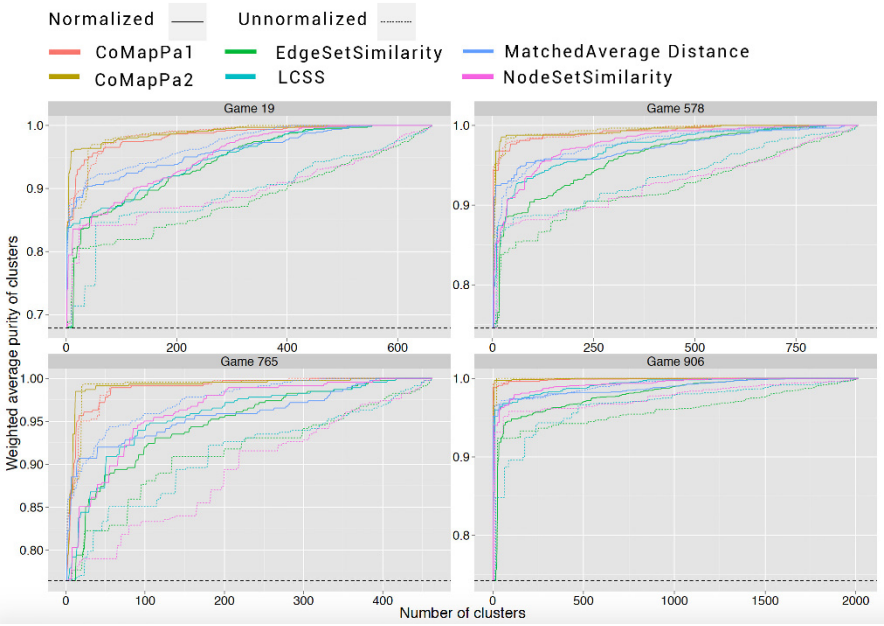

Fig. 2: Weighted average purity of the clustering results for some exemplary configurations, i.e., the Games 19, 578, 765, and 906.

designed similarity measure should at least distinguish between paths of these two classes. Hence, for each path of a configuration c, we define the binary attribute $q : \mathcal{P}_c \to \{0,1\}$ which yields a 1 for a solving path, and a 0 for a non-solving path. A given cluster $\gamma = \{p_1, \ldots, p_m\} \subseteq \mathcal{P}_c$ is then called *pure* if all paths in γ are either solving or non-solving. Since the requirement that a cluster should be pure, is a very strict one, we rather consider its *purity*. The purity of a cluster γ is defined as $purity(\gamma) = \frac{1}{|\gamma|} \max\{\sum_{p_i \in \gamma} q(p_i), |\gamma| - \sum_{p_i \in \gamma} q(p_i)\}$, i.e., the maximum of the two fractions of paths in γ which are solving or non-solving. Note that $purity(\gamma) \geq 0.5$ always holds. Let $q(\mathcal{P}_c) = \frac{1}{|\mathcal{P}_c|} \max\{\sum_{p \in \mathcal{P}_c} q(p), |\mathcal{P}_c| - \sum_{p \in \mathcal{P}_c} q(p)\}$ denote the fraction of paths for configuration c which are solving or non-solving.

For a given partition $\Gamma = \{\gamma_1, \ldots, \gamma_k\}$ of $\mathcal{P}_c$, the average purity of all groups can be used as an evaluation criterion for the given partition. However, an unweighted average of the purities has the effect that the average purity is higher if Γ contains many singletons because they contribute with a purity of 1.0 each. We therefore consider a weighted average purity for Γ where the purity of each cluster from Γ contributes proportionally to its size to the average. The weighted average purity for a set of clusters Γ is defined as $purity_w(\Gamma) = \frac{1}{\sum_{\gamma_i \in \Gamma} |\gamma_i|} \sum_{\gamma_i \in \Gamma} |\gamma_i| \cdot purity(\gamma_i)$. However, the optimal number of clusters is not known. We thus consider the weighted average purity of all possible number of clusters. For a configuration c, the number of possible clusters ranges from 1 to $|\mathcal{P}_c|$. The weighted average purity for any configuration c

and for any similarity measure is 1.0 for $|\mathcal{P}_c|$ many clusters, and $q(\mathcal{P}_c)$ for 1 cluster. The behaviour between these extremes can then be used as evaluation criterion and means of comparison between the proposed similarity measures, for example to find out which similarity measure reaches the highest average purity with the smallest numbers of clusters.

4.3 Results

For each start configuration c and each similarity measure, the weighted average purity is computed for each number of clusters between 1 and $|\mathcal{P}_c|$. Figure 2 shows the results for some exemplary configurations. The possible number of clusters (i.e., the number of paths) is drawn on the x-axes, the corresponding weighted average purity of the clusters on the y-axes. Note that the weighted average purity is always larger than $q(\mathcal{P}_c)$ which is indicated by the dashed line. The first observation is that clustering with any of the similarity measures yields partitions with a weighted average purity considerably higher than the corresponding q value. Furthermore, the CoMapPa1 and CoMapPa2 distance measures perform clearly better than the purely set- or order-based measures. With these two measures, it is possible to obtain a weighted average purity close to 1 with only a few clusters. This observation is supported by Table 2 which presents the weighted average purity for the clustering results for all similarity measures for some graphs, if the number of clusters is fixed to 5, 10, 20, or 30[2]. For each game and for each $x \in \{5, 10, 20, 30\}$, the highest p_x is highlighted. Table 2 reveals that for almost all games, the CoMapPa1 and CoMapPa2 distance obtain the highest weighted average purity, often close to 100 %. This is even achieved for game 723 where the number of solving and non-solving paths are almost equal. Nevertheless, clustering the 2704 paths with CoMapPa1 and CoMapPa2 yields almost pure clusters when only choosing 5 clusters. Figure 2 also indicates that the CoMapPa1 and CoMapPa2 measures perform almost equally well when using the normalized or unnormalized version of the measure. This is not the case for the set-based and order-based measures: here, the unnormalized measures consistently yield less good results.

In order to show that these observations are not only artifacts of single games, we adapt the idea of considering the area under the curve of the corresponding weighted average purity line. Informally, for a given sequence of weighted average purities (one entry per possible number of clusters) for one game and one similarity measure, we consider the area between the corresponding curve and the corresponding q line. Dividing this value by the size of the area of the "ideal" curve which reaches a weighted average purity of 100 % with 2 clusters, yields the *relative AUC*. The relative AUC is computed for every similarity measure and every game. The results are shown in Figure 3 (left). The observations made for single games can be confirmed here. The relative AUC is consistently higher for all games for the CoMapPa1 and

[2] The table with the results for all configurations is contained in the supplementary material available under http://gtna.cs.uni-kl.de/en/gruppe/bockholt/PDFs/CN2016SupplementaryMaterial.pdf

Table 2: The weighted average purity for each of the six similarity measures for a fixed number of clusters. For each game, results for the unnormalized measure are presented in the first line, results for the normalized measure are presented in the second line. p_x denotes the weighted average purity of the clustering when choosing x clusters. For each game and each $x \in \{5, 10, 20, 30\}$ the highest p_x is highlighted. $q(\mathcal{P}_c)$ is denoted by q and gives the fraction of solving or non-solving paths of all paths for the configuration. All values are percentages. Because of lack of space, the table only shows the results for a few games. The full table is available online under the given link.

	σ_{nss}				σ_{ess}				σ_{lcs}				δ_{mad}				$\delta_{comappa1}$				$\delta_{comappa2}$				
	p_5	p_{10}	p_{20}	p_{30}	p_5	p_{10}	p_{20}	p_{30}	p_5	p_{10}	p_{20}	p_{30}	p_5	p_{10}	p_{20}	p_{30}	p_5	p_{10}	p_{20}	p_{30}	p_5	p_{10}	p_{20}	p_{30}	q
Game 19	69	69	78	84	69	74	81	81	68	71	71	71	87	87	88	89	85	88	89	90	85	85	87	88	67.82
	79	79	84	84	68	68	81	84	84	84	84	85	84	86	89	89	85	85	92	94	92	96	96	96	
Game 357	72	82	82	87	75	75	81	81	74	81	82	85	90	91	95	95	99	99	100	100	93	98	99	99	71.71
	87	87	87	89	82	83	88	89	80	84	87	89	85	90	90	91	95	95	98	100	99	100	100	100	
Game 723	55	56	66	74	55	57	58	63	55	57	65	79	95	95	96	96	99	99	99	99	99	99	99	99	54.44
	74	90	94	94	55	56	58	61	81	84	93	94	95	95	96	96	96	99	99	99	99	99	99	99	
Game 765	76	78	79	79	76	78	78	82	76	77	77	80	86	86	89	91	86	88	95	95	86	86	99	99	76.41
	77	80	85	85	76	76	79	86	78	79	84	86	84	89	91	91	82	90	96	96	87	94	98	99	

CoMapPa2 measure, regardless whether the normalized or unnormalized version is used. The relative AUC for all other measures is smaller and there are high differences between the normalized and unnormalized versions. When considering the results shown in Figures 2 and 3 (left), it is striking that the unnormalized versions of the set- and order-based measures yield clusters with a considerably smaller weighted average purity than the normalized version. There is the possibility that the similarity measures only distinguish between shorter and longer paths (because clearly, a solving path needs to have a certain length while non-solving paths can be short) and reach high average purity by this effect. Therefore, Figure 3 (right) shows the relative AUC of the resulting clusters, if for each game, only paths at least as long as the shortest solving path are considered. The gap between the normalized and unnormalized versions of the measures clearly decreases, but the general trend of the previous results is confirmed. Thus, clustering the paths with the proposed similarity measures can distinguish quite well between solving and non-solving paths. This implies that solving and non-solving paths show structural differences that can be detected by such simple similarity measures.

5 Conclusion

In this paper we have shown on a first benchmark data set and a simple ground truth, that already very simple quantifications of the similarity of paths in complex networks yield interesting insights into this new dimension of analyzable data. We have shown that—using a simple clustering algorithm—the measures which incorporate the underlying graph and the traversal order of the paths, contain the most information to categorize the paths representing the solving attempts of games into those that finally

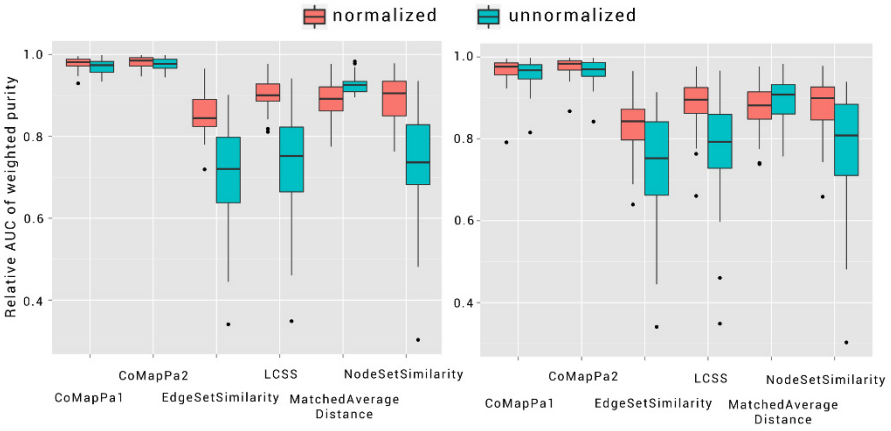

Fig. 3: Relative AUC of the weighted purity for all paths of all configurations (left) and when only sufficiently long paths are considered (right).

solve it and those that do not, to a quite high degree. The results imply that similarity measures which take into account the underlying network structure are best-suited to find groups of similar paths. However, the results are currently only valid for one specific data set which is why future work should aim at generalizing and validating the proposed measures on further data sets. In general, we believe that there is a wealth of data contained in the paths actually taken in a complex network rather than in the ones imposed by, e.g., centrality indices that always assume that either random walks or shortest paths are used. In another paper, Dorn, Lindenblatt and Zweig showed that centralities based on actual path data are also less prone to artifacts than classic centrality indices [5]. Thus, an important task for the community in network analysis should be to obtain such data and to publish it—preferably with ground truth regarding clusterings, centrality of nodes in the paths, external parameters like time taken or time stamps at the single nodes, etc.—to mine and analyze it together with the underlying network structures.

References

[1] Bashir, F., Khokhar, A., Schonfeld, D.: Segmented trajectory based indexing and retrieval of video data. In: Proceedings of the International Conference on Image Processing, vol. 2, pp. II–623. IEEE (2003)

[2] Buchin, K., Buchin, M., Gudmundsson, J., Löffler, M., Luo, J.: Detecting Commuting Patterns by Clustering Subtrajectories. In: Algorithms and Computation: 19th International Symposium, ISAAC 2008, Gold Coast, Australia, December 15-17, 2008. Proceedings, September, pp. 644–655 (2008)

[3] Buzan, D., Sclaroff, S., Kollios, G.: Extraction and clustering of motion trajectories in video. In: Proceedings of the 17th International Conference on Pattern Recognition, vol. 2, pp. 521–524. IEEE (2004)

[4] Cho, E., Myers, S.A., Leskovec, J.: Friendship and mobility: user movement in location-based social networks. In: Proceedings of the 17th ACM SIGKDD international conference on

Knowledge discovery and data mining, pp. 1082–1090. ACM (2011)
[5] Dorn, I., Lindenblatt, A., Zweig, K.A.: The trilemma of network analysis. In: Proceedings of the 2012 IEEE/ACM international conference on Advances in Social Network Analysis and Mining, Istanbul (2012)
[6] González, M.C., Hidalgo, C.A., Barabási, A.L.: Understanding individual human mobility patterns. Nature **453**(7196), 779–782 (2008)
[7] Gudmundsson, J., Thom, A., Vahrenhold, J.: Of Motifs and Goals: Mining Trajectory Data. In: Proceedings of the 20th International Conference on Advances in Geographic Information Systems - SIGSPATIAL '12, pp. 129–138. ACM (2012)
[8] Gusfield, D.: Algorithms on strings, trees and sequences: computer science and computational biology. Cambridge University Press, New York, NY, USA (1997)
[9] Jaccard, P.: Etude comparative de la distribution florale dans une portion des alpes et du jura. Bulletin del la Société Vaudoise des Sciences Naturelles **37**, 547–579 (1901)
[10] Jarušek, P.: Modeling problem solving times in tutoring systems. Ph.D. thesis, Masarykova univerzita, Fakulta informatiky (2013)
[11] Jarušek, P., Pelánek, R.: Analysis of a simple model of problem solving times. In: S. Cerri, W. Clancey, G. Papadourakis, K. Panourgia (eds.) Intelligent Tutoring Systems, *Lecture Notes in Computer Science*, vol. 7315, pp. 379–388. Springer, Berlin Heidelberg (2012)
[12] Junejo, I.N., Javed, O., Shah, M.: Multi feature path modeling for video surveillance. In: Proceedings of the 17th International Conference on Pattern Recognition, vol. 2, pp. 716–719. IEEE (2004)
[13] Kumar, P., Raju, B.S., Krishna, P.R.: A new similarity metric for sequential data. Exploring Advances in Interdisciplinary Data Mining and Analytics: New Trends: New Trends p. 233 (2011)
[14] Laasonen, K.: Clustering and prediction of mobile user routes from cellular data. In: Knowledge Discovery in Databases: PKDD 2005, *Lecture Notes in Computer Science*, vol. 3721, pp. 569–576. Springer, Berlin Heidelberg (2005)
[15] Makris, D., Ellis, T.: Path detection in video surveillance. Image and Vision Computing **20**(12), 895–903 (2002)
[16] Mannila, H., Moen, P.: Similarity between event types in sequences. In: Proceedings of the First International Conference on Data Warehousing and Knowledge Discovery, pp. 271–280. Springer, London (1999)
[17] Mannila, H., Ronkainen, P.: Similarity of event sequences. In: Proceedings of the 4th International Workshop on Temporal Representation and Reasoning (TIME), p. 136. IEEE Computer Society (1997)
[18] Moen, P.: Attribute, event sequence, and event type similarity notions for data mining. Ph.D. thesis, University of Helsinki, Department of Computer Science (2000)
[19] Vlachos, M., Kollios, G., Gunopulos, D.: Discovering similar multidimensional trajectories. In: Proceedings of the 18th International Conference on Data Engineering, pp. 673–684. IEEE (2002)
[20] Wang, W., Zaïane, O.R.: Clustering web sessions by sequence alignment. In: Database and Expert Systems Applications, 2002. Proceedings. 13th International Workshop on, pp. 394–398. IEEE (2002)
[21] Ward, J.H.: Hierarchical grouping to optimize an objective function. Journal of the American Statistical Association **58**(301), 236–244 (1963)