Using sequence constraints for modelling network interactions

Using Sequence Constraints for Modelling Network Interactions

Johannes De Smedt[1], Junichiro Mori[2], and Masanao Ochi[2]

[1] The University of Edinburgh, Management Science and Business Economics Group
29 Buccleuch Place, EH8 9JS, Edinburgh, UK
johannes.desmedt@ed.ac.uk
[2] The University of Tokyo
7-3-1 Bunkyo-ku, Hongo, Tokyo, JAPAN
mori@mi.u-tokyo.ac.jp;ochi@ipr-ctr.t.u-tokyo.ac.jp

Abstract. The ubiquitous nature of networks has led to a vast number of works dedicated to the study of capturing their information. Various graph-based techniques exist that report on the characteristics of nodes and edges, e.g., author-citation networks, social interactions, and so on. A significant amount of information can be extracted by summarizing the surrounding network structure of nodes, e.g., by capturing motives, or walk patterns. In this work, we present a new way of capturing the interaction between nodes in a network by making use of the sequence in which they occur. (1) The objective of this paper is to make use of behavioural constraint patterns; a concise but detailed report of node's interactions can be constructed that can be used for various purposes. (2) It is shown how the constraint patterns can be mined form interaction data, and how they can be used for various applications. (3) A case study is presented which uses behavioural constraint patterns to profile user interactions in a communication network.

Keywords: Sequence mining, network analysis

1 Introduction

Networks are often formed by the interaction of various actors. For example, social networks grow based on friendship or interested-based relations, forum posts and emails link users according to their communication patterns, and citation networks are formed through authors referencing peers in their field. Typically, the construction of these networks is based on either undirected, or directed edges with weights. Furthermore, many network techniques focus on static relationships, I.e., the evolution over time is not investigated. However, a range of new techniques emerged recently that focus on the time-aspect of a network. Most notably, the use of motifs [11], and streams [8] allow to capture the evolution of a network over time. In this paper, we describe a new approach based on behavioral constraints, I.e., constraints based on sequence patterns that allow to describe the order of the interactions of nodes.

We investigate how they can be constructed from a network dataset, and use the various patterns to describe the evolution of the network over time. In particular, we apply the sequence mining method to the question-and-answer interaction-based network. Our preliminary results show that profiling network interactions patterns with sequence mining enables track the behaviour of nodes in a transactional network without relying on the typical partial-order based results.

This paper is structured as follows. In Section 2, the methodology is presented to mine constraints from network data. Next, Section 3 reports on the application on a real-life dataset. Finally, Section 4 concludes the paper and reports on the future directions.

2 Behavioural constraints for sequence interactions

In this section, an introduction to behavioural constraints is given, which includes a motivation which ones are suitable for network interactions. Next, the method for mining them is discussed. Finally, a number of potential applications are delineated.

2.1 Constraint set

Behavioural constraint templates have been long used in various areas of computer science. Most notably, a comprehensive set of Linear Temporal Logic (LTL) templates was proposed for the formal verification of program execution [6]. LTL provides an adequate formalism to search for various temporal properties, such as whether something happens eventually, next, and so on, and can be used in conjunction with typical logical operators to construct expressive relations. The initial set was extended to include various other relations, most notably unary ones. These constraints were later adapted towards the case of process modelling [13]. While initially proposed as LTL formulae which are convertible to Büchi automata, finite trace equivalent regular expressions were introduced in [2] and [15]. Models existing of multiple constraints at the same time can be obtained by conjoining the automata to obtain a global language or automaton, over which all constraints hold.

In Table 1, an overview of the most-commonly used constraints in literature. They are organized according to 7 different categories, including unary and binary constraints. Most notably, the binary constraints follow a hierarchy which is reported in [2] and which covers unordered relations up to chain ordered relations (using the next operator) at the top. Besides, the inclusion of negative constraints is unique, as typically only patterns of positive relations are reported. Negative constraints capture behaviour that has not occurred. Including negative behaviour can be used to find relations that are not apparent at first sight, e.g., in Figure 1, the fact that nodes A and E are both present in the sequence of C, but do not have interactions themselves, still allows the inference of not succession(A,E). While only unary and binary constraints are included, it is also

Type	**Template**	**LTL Formula [12]**	**Regular Expression [15]**
Unary	Existence(A,n)	$\Diamond(A \wedge \bigcirc(existence(n-1, A)))$	.*(A.*){n}
	Absence(A,n)	$\neg existence(n, A)$	[^A]*(A?[^A]*){n-1}
	Exactly(A,n)	$existence(n, A) \wedge absence(n+1, A)$	[^A]*(A[^A]*){n}
	Init(A)	A	(A.*)?
	Last(A)	$\Box(A \implies \neg X \neg A)$	.*A
Unordered	Responded existence(A,B)	$\Diamond A \implies \Diamond B$	[^A]*((A.*B.*) |(B.*A.*))?
	Co-existence(A,B)	$\Diamond A \iff \Diamond B$	[^AB]*((A.*B.*) |(B.*A.*))?
Simple ordered	Response(A,B)	$\Box(A \implies \Diamond B)$	[^A]*(A.*B)*[^A]*
	Precedence(A,B)	$(\neg B \, U \, A) \vee \Box(\neg B)$	[^B]*(A.*B)*[^B]*
	Succession(A,B)	$response(A, B) \wedge precedence(A, B)$	[^AB]*(A.*B)*[^AB]*
Alternating ordered	Alternate response(A,B)	$\Box(A \implies \bigcirc(\neg A \, U \, B))$	[^A]*(A[^A]*B[^A]*)*
	Alternate precedence(A,B)	$precedence(A, B) \wedge \Box(B \implies \bigcirc(precedence(A, B))$	[^B]*(A[^B]*B[^B]*)*
	Alternate succession(A,B)	$altresponse(A, B) \wedge precedence(A, B)$	[^AB]*(A[^AB]*B[^AB]*)*
Chain ordered	Chain response(A,B)	$\Box(A \implies \bigcirc B)$	[^A]*(AB[^A]*)*
	Chain precedence(A,B)	$\Box(\bigcirc B \implies A)$	[^B]*(AB[^B]*)*
	Chain succession(A,B)	$\Box(A \iff \bigcirc B)$	[^AB]*(AB[^AB]*)*
Negative	Not co-existence(A,B)	$\neg(\Diamond A \wedge \Diamond B)$	[^AB]*((A[^B]*) |(B[^A]*))?
	Not succession(A,B)	$\Box(A \implies \neg(\Diamond B))$	[^A]*(A[^B]*)*
	Not chain succession(A,B)	$\Box(A \implies \neg(\bigcirc B))$	[^A]*(A+[^AB][^A]*)*A*
Choice	Choice(A,B)	$\Diamond A \vee \Diamond B$	.*[AB].*
	Exclusive choice(A,B)	$(\Diamond A \vee \Diamond B) \wedge \neg(\Diamond A \wedge \Diamond B)$	([^B]*A[^B]*) |.*[AB].*([^A]*B[^A]*)

Table 1: An overview of Declare constraint templates with their corresponding LTL formula and regular expression.

Interactions:
A: A → B, A → B, A → C, A → B, C → A
B: A → B, B → D, A → B, B → D, D → B
C: A → C, C → A, C → E
D: B → D, B → D, D → B
E: C → E

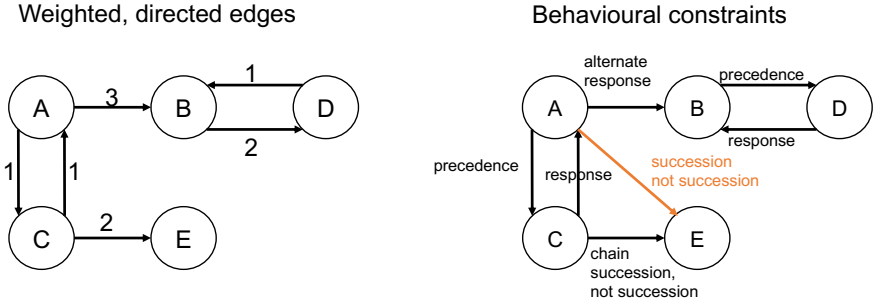

Fig. 1: Running example

possible to use target-branched constraints to model interactions between more than 2 nodes [4]. Motifs can be handy to capture various profiles of directed arcs between 2 or more nodes [10], but with the constraint sets it is possible to create even more intricate profiles of arc relations between nodes.

Despite not being useful for capturing interaction effects, the unary constraints can be used for adding information to a node's feature vector in case any exist. I.e., if a particular node is always occurring first in a sequence, this might signify a particular pattern, e.g., a person reporting recently-occurred disasters.

Not every constraint is suitable for binary interaction within a network context, i.e., not chain succession is, in general, not suitable for profiling behavior, as it holds in many situations. Besides, absence is hard to identify unless a particular node is scrutinized for this behaviour in the sequence of another node. Exclusive choice and not co-existence are similar in this respect, where the latter does not require the presence of either. Similar to not chain succession, this might lead to the discovery of many frequently non-occurring pairs.

2.2 Mining the patterns

We define transactional network data as an ordered set of interactions T between nodes from the set N, where each transaction is a tuple $(n_1, n_2, ts) \in T$ with $n_1 \in N$ the initiating node, $n_2 \in N$ the receiving node, and $ts \in \mathbb{N}^+$ a timestamp. T can be read sequentially, where each node $n \in N$ has a sequence $s_n \subset 2^{|N|}$

that is extended whenever a transaction $t \in T$ is for that node is witnessed. I.e., s_n gets extended with $\langle n, n_o \rangle$ whenever n is the initiating node, and with $\langle n_o, n \rangle$ when n is on the receiving end given another node $n_o \in N$.

By using the interesting Behavioural Constraint Miner [3], we can mine all patterns in a sequence s_n to obtain a set of constraints C_{s_n}. Note, however, that if a given binary constraint $c(n, n_2) \in C_{s_n}$ holds for n in its own sequence, this still has to be verified with the sequence of the other node. If $c(n, n_2)$ is not present in that sequence, the constraints do not hold. Consider for example the interaction in Figure 1. Despite the evidence in the sequence of A that there exists an alternate succession relationship between A and B due to the alternating ABABAAB pattern, the sequence of B rather indicates that other occurrences of B happen in between (e.g. B→D), breaking the pattern. Hence, a final step is required to recursively ensure that $C_n = \{c \mid c \in C_n \wedge c \in C_{n_i} \forall n_i \in \mathcal{N}(n) \vee c \notin C_n \wedge c \notin C_{n_i} \forall n_i \in \mathcal{N}(n)\}$ where $\mathcal{N}(n) \subseteq N$ denotes the neighbourhood of node n to check that all constraint pertaining to n are either both in its constraint set and the constraint set of its neighbours to avoid conflict, or that it is present in an unrelated node (e.g. the connection succession(A,E) in Figure 1). To conclude the discovery of sequence templates from the network interactions, the sets C_n are pruned according to the constraint hierarchy.

2.3 Applications

The mining of interactions in a network as sequences has several applications. Most notably, the sequence information can be used for analysing the patterns that exist between nodes, and their evolution over time. By tracking what patterns exist, and whether they return over time gives an overview of how certain relations change and what the underlying sequential behaviour is. In this case, unary constraints might not be useful, but especially the hierarchy of binary constraints can pin down how strong the relationship between two nodes is.

Next, the sequence patterns can be used as features of a node to obtain vector representations of nodes or relations between nodes. The presence of relations can be stored in a binary vector for relations, or the number of relations of each type can be stored for nodes. In this case, also unary constraints help define the node in terms of where in a sequence (init/last), how often (existence/exactly), besides with what other nodes the node is interacting (binary constraints). The features can be used towards node classification [1] or node relation prediction [9].

Given that all constraints have formal semantics in LTL or regular expressions, it is possible to exploit these in order to simulate the sequence behaviour later. These languages support state machine models which can generate strings of nodes by creating a global state machine that models all the constraints' behaviour. This can potentially feed into techniques such as node2vec [7], which typically employ random walks [14] for getting node representations.

Finally, by using the transitivity properties of the constraints, link inference/prediction [9] can also be made.

3 Application

In this section, the approach is applied to a real-life dataset in order to retrieve the evolution of constraints over time to profile the network interactions from a behavioural perspective.

3.1 Data

We apply the sequence method to the Math Overflow dataset, as used in [11]. On the Overflow web sites, users post questions and receive answers from other users, and users may comment on both questions and answers. We derive a transactional network by creating an edge (u, v, t) if, at time t, user u: (1) posts an answer to user v's question, (2) comments on user v's question, or (3) comments on user v's answer. The data contains 24,818 nodes with 506,550 interactions over 2,350 days and deals with question-and-answer data from users regarding mathematical problems.

We retrieve the constraints over the dataset by splitting the interactions into contingent blocks of a varying time length. In this case, we used blocks of 4 hours (14,102 blocks), 2 days (1,175 blocks), 100 days (23 blocks), and 1,000 days (2 blocks) in order to track the evolution of the constraints. The evolution is captured by tallying the shift in constraint type between nodes present in subsequent blocks. For this analysis, we limit the constraint set to the 7 most common sequence patterns. The following coding was used:

- 1: not succession
- 2: precedence
- 3: alternate precedence
- 4: chain precedence
- 5: response
- 6: alternate response
- 7: chain response

In the columns, 0 stands for the absence of a constraint in subsequent time blocks between nodes that both reoccur.

In order to understand the difference in behaviour of various user types, we analyse the evolution of various node type and look into nodes that have a degree less than or equal to 3 (low involvement), a degree of 4-10 (medium involvement), and a degree of more than 10 (high involvement - 5,821 nodes). The results can be found in Tables 2 and 3 for incoming and outgoing relations respectively. Besides, the node with the highest authority score [5] is included as well, in order to illustrate how the most important node in terms of the dispersion of information acts within the web site. The results can be found in Table 4. The colours denote the place in the distribution, where red is higher and green is lower. Scores with different colours and equal scores indicate a difference in value behind the significant digits.

		4 hours								2 days							
IN		**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
≤3	**1**	0.072	0.025	0.025	0.000	0.100	0.007	0.000	0.086	0.214	0.022	0.028	0.000	0.040	0.013	0.000	0.028
	2	0.063	0.013	0.011	0.000	0.011	0.001	0.000	0.012	0.144	0.018	0.022	0.000	0.004	0.008	0.000	0.004
	3	0.000	0.001	0.000	0.000	0.001	0.000	0.000	0.001	0.000	0.001	0.001	0.000	0.005	0.000	0.000	0.001
	4	0.076	0.064	0.032	0.001	0.038	0.008	0.001	0.034	0.079	0.038	0.030	0.001	0.010	0.014	0.001	0.015
	5	0.058	0.023	0.004	0.001	0.008	0.007	0.000	0.011	0.116	0.018	0.007	0.000	0.006	0.021	0.000	0.003
	6	0.000	0.001	0.000	0.000	0.001	0.000	0.000	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	7	0.050	0.042	0.011	0.000	0.012	0.015	0.000	0.075	0.027	0.021	0.006	0.000	0.004	0.008	0.000	0.019
4–10	**1**	0.086	0.018	0.037	0.000	0.047	0.011	0.000	0.059	0.229	0.014	0.038	0.001	0.011	0.021	0.000	0.016
	2	0.106	0.013	0.028	0.000	0.011	0.007	0.000	0.021	0.201	0.018	0.027	0.000	0.005	0.011	0.000	0.008
	3	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.003	0.000	0.003	0.000	0.001	0.000	0.000	0.001
	4	0.062	0.023	0.031	0.000	0.018	0.010	0.000	0.035	0.042	0.010	0.014	0.000	0.003	0.011	0.000	0.008
	5	0.091	0.010	0.010	0.000	0.007	0.014	0.000	0.013	0.164	0.014	0.007	0.000	0.003	0.019	0.000	0.006
	6	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.001	0.001	0.000	0.001	0.000	0.000	0.000	0.000	0.000
	7	0.062	0.019	0.012	0.000	0.008	0.021	0.000	0.103	0.038	0.007	0.006	0.000	0.001	0.011	0.000	0.019
>10	**1**	0.143	0.028	0.028	0.001	0.024	0.017	0.001	0.027	0.218	0.014	0.018	0.001	0.002	0.017	0.000	0.002
	2	0.146	0.024	0.024	0.001	0.008	0.011	0.000	0.013	0.274	0.017	0.023	0.001	0.001	0.007	0.001	0.002
	3	0.003	0.001	0.001	0.000	0.000	0.000	0.000	0.000	0.007	0.000	0.001	0.000	0.000	0.001	0.000	0.000
	4	0.046	0.028	0.015	0.000	0.010	0.010	0.000	0.015	0.016	0.002	0.002	0.000	0.001	0.002	0.000	0.001
	5	0.142	0.022	0.008	0.000	0.006	0.021	0.001	0.009	0.278	0.017	0.006	0.001	0.001	0.024	0.001	0.002
	6	0.003	0.001	0.001	0.000	0.000	0.000	0.000	0.000	0.006	0.000	0.001	0.000	0.000	0.001	0.000	0.000
	7	0.050	0.026	0.009	0.000	0.005	0.019	0.000	0.050	0.021	0.002	0.002	0.000	0.000	0.004	0.000	0.003

		100 days								1000 days							
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
≤3	**1**	0.330	0.022	0.024	0.001	0.001	0.017	0.000	0.000	0.277	0.022	0.027	0.000	0.000	0.018	0.000	0.000
	2	0.283	0.020	0.028	0.001	0.001	0.002	0.000	0.000	0.312	0.031	0.025	0.001	0.000	0.003	0.000	0.000
	3	0.002	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.001	0.000	0.001	0.000	0.000	0.001	0.000	0.000
	4	0.011	0.001	0.001	0.000	0.000	0.001	0.000	0.000	0.005	0.001	0.001	0.000	0.000	0.001	0.000	0.000
	5	0.212	0.013	0.008	0.001	0.000	0.013	0.000	0.000	0.230	0.022	0.008	0.001	0.000	0.011	0.000	0.000
	6	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	7	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000
4–10	**1**	0.309	0.023	0.032	0.001	0.000	0.029	0.000	0.000	0.256	0.024	0.040	0.002	0.000	0.032	0.001	0.000
	2	0.264	0.021	0.027	0.001	0.000	0.005	0.000	0.000	0.265	0.027	0.039	0.002	0.000	0.006	0.001	0.000
	3	0.005	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.008	0.001	0.001	0.000	0.000	0.001	0.000	0.000
	4	0.003	0.000	0.001	0.000	0.000	0.000	0.000	0.000	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	5	0.231	0.017	0.007	0.000	0.000	0.015	0.000	0.000	0.231	0.022	0.011	0.001	0.000	0.023	0.001	0.000
	6	0.003	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.004	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	7	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
>10	**1**	0.194	0.024	0.033	0.001	0.000	0.034	0.001	0.000	0.160	0.032	0.046	0.001	0.000	0.047	0.001	0.000
	2	0.252	0.032	0.047	0.001	0.000	0.012	0.001	0.000	0.216	0.043	0.062	0.001	0.000	0.019	0.001	0.000
	3	0.005	0.001	0.001	0.000	0.000	0.001	0.000	0.000	0.004	0.001	0.001	0.000	0.000	0.001	0.000	0.000
	4	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	5	0.254	0.033	0.012	0.001	0.000	0.048	0.001	0.000	0.222	0.043	0.017	0.001	0.000	0.068	0.002	0.000
	6	0.005	0.001	0.001	0.000	0.000	0.001	0.000	0.000	0.005	0.001	0.001	0.000	0.000	0.001	0.000	0.000
	7	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

Table 2: An overview of the proportion of incoming constraints that shift from one sequence pattern into another for nodes with different degrees between different time blocks of varying lengths.

		100 days								1000 days							
OUT		**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
≤3	**1**	0.136	0.058	0.017	0.000	0.059	0.014	0.002	0.068	0.133	0.021	0.030	0.000	0.017	0.016	0.001	0.030
	2	0.080	0.041	0.013	0.000	0.005	0.015	0.001	0.006	0.216	0.025	0.020	0.000	0.004	0.009	0.000	0.007
	3	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	4	0.013	0.086	0.009	0.000	0.013	0.008	0.000	0.010	0.018	0.019	0.004	0.000	0.004	0.004	0.000	0.006
	5	0.082	0.021	0.004	0.000	0.005	0.011	0.001	0.008	0.236	0.012	0.004	0.000	0.003	0.025	0.001	0.005
	6	0.002	0.002	0.000	0.000	0.000	0.000	0.000	0.000	0.002	0.001	0.000	0.000	0.000	0.000	0.000	0.000
	7	0.022	0.114	0.001	0.000	0.002	0.029	0.000	0.048	0.028	0.040	0.003	0.000	0.002	0.027	0.001	0.023
4–10	**1**	0.167	0.060	0.018	0.001	0.028	0.013	0.001	0.026	0.152	0.013	0.021	0.000	0.007	0.017	0.001	0.008
	2	0.116	0.038	0.017	0.000	0.007	0.007	0.001	0.007	0.253	0.022	0.018	0.001	0.003	0.009	0.002	0.004
	3	0.001	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.004	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	4	0.032	0.063	0.010	0.000	0.008	0.011	0.000	0.010	0.022	0.014	0.003	0.000	0.002	0.007	0.001	0.002
	5	0.119	0.036	0.003	0.000	0.007	0.017	0.001	0.007	0.268	0.022	0.004	0.001	0.003	0.023	0.002	0.005
	6	0.003	0.001	0.001	0.000	0.001	0.001	0.000	0.001	0.006	0.001	0.000	0.000	0.000	0.000	0.000	0.000
	7	0.033	0.061	0.005	0.000	0.004	0.023	0.001	0.034	0.033	0.015	0.004	0.000	0.001	0.018	0.002	0.009
>10	**1**	0.138	0.026	0.029	0.001	0.026	0.017	0.000	0.030	0.220	0.014	0.018	0.001	0.002	0.017	0.000	0.002
	2	0.144	0.023	0.025	0.001	0.008	0.010	0.000	0.014	0.272	0.017	0.023	0.001	0.001	0.008	0.000	0.002
	3	0.003	0.001	0.001	0.000	0.000	0.000	0.000	0.000	0.007	0.000	0.001	0.000	0.000	0.001	0.000	0.000
	4	0.048	0.026	0.017	0.000	0.011	0.010	0.000	0.017	0.017	0.002	0.003	0.000	0.001	0.002	0.000	0.001
	5	0.139	0.021	0.008	0.000	0.006	0.020	0.001	0.010	0.275	0.016	0.006	0.001	0.001	0.024	0.001	0.002
	6	0.002	0.000	0.001	0.000	0.000	0.000	0.000	0.000	0.006	0.000	0.001	0.000	0.000	0.001	0.000	0.000
	7	0.052	0.023	0.010	0.000	0.005	0.019	0.000	0.054	0.021	0.002	0.002	0.000	0.000	0.004	0.000	0.003

		100 days								1000 days							
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
≤3	**1**	0.180	0.010	0.014	0.000	0.000	0.017	0.002	0.001	0.211	0.007	0.023	0.000	0.000	0.028	0.002	0.000
	2	0.297	0.012	0.023	0.000	0.000	0.008	0.000	0.000	0.265	0.016	0.022	0.000	0.000	0.010	0.000	0.000
	3	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	4	0.002	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	5	0.359	0.016	0.003	0.000	0.000	0.037	0.001	0.001	0.338	0.020	0.002	0.000	0.000	0.043	0.001	0.000
	6	0.002	0.000	0.001	0.000	0.000	0.001	0.000	0.000	0.004	0.001	0.000	0.000	0.000	0.000	0.000	0.000
	7	0.007	0.001	0.000	0.000	0.000	0.001	0.000	0.001	0.002	0.000	0.000	0.000	0.000	0.000	0.000	0.000
4–10	**1**	0.167	0.009	0.015	0.000	0.000	0.018	0.001	0.000	0.179	0.014	0.022	0.000	0.000	0.027	0.001	0.000
	2	0.305	0.021	0.029	0.000	0.000	0.009	0.000	0.000	0.271	0.024	0.032	0.000	0.000	0.015	0.000	0.000
	3	0.004	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.005	0.000	0.000	0.000	0.000	0.001	0.000	0.000
	4	0.002	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	5	0.332	0.024	0.007	0.000	0.000	0.039	0.001	0.000	0.302	0.029	0.008	0.000	0.000	0.050	0.002	0.000
	6	0.007	0.000	0.001	0.000	0.000	0.001	0.000	0.000	0.011	0.001	0.001	0.000	0.000	0.002	0.000	0.000
	7	0.004	0.000	0.000	0.000	0.000	0.001	0.000	0.000	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000
>10	**1**	0.198	0.025	0.034	0.001	0.000	0.034	0.001	0.000	0.168	0.033	0.047	0.001	0.000	0.048	0.001	0.000
	2	0.251	0.032	0.047	0.001	0.000	0.012	0.001	0.000	0.215	0.044	0.063	0.001	0.000	0.018	0.001	0.000
	3	0.005	0.001	0.001	0.000	0.000	0.001	0.000	0.000	0.005	0.001	0.001	0.000	0.000	0.001	0.000	0.000
	4	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	5	0.251	0.032	0.012	0.001	0.000	0.047	0.001	0.000	0.214	0.043	0.017	0.001	0.000	0.066	0.001	0.000
	6	0.005	0.001	0.001	0.000	0.000	0.001	0.000	0.000	0.005	0.001	0.001	0.000	0.000	0.001	0.000	0.000
	7	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

Table 3: An overview of the proportion of outgoing constraints that shift from one sequence pattern into another for nodes with different degrees between different time blocks of varying lengths.

		4 hours								2 days							
IN		**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
	1	0.172	0.020	0.019	0.000	0.007	0.019	0.000	0.009	0.195	0.020	0.026	0.001	0.001	0.025	0.001	0.001
	2	0.219	0.021	0.026	0.001	0.003	0.008	0.001	0.005	0.267	0.028	0.035	0.001	0.001	0.011	0.001	0.002
	3	0.006	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.006	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	4	0.040	0.007	0.006	0.000	0.003	0.005	0.000	0.005	0.010	0.001	0.001	0.000	0.000	0.001	0.000	0.000
	5	0.224	0.020	0.007	0.000	0.003	0.026	0.001	0.004	0.264	0.028	0.009	0.001	0.001	0.036	0.001	0.002
	6	0.002	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.005	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	7	0.062	0.010	0.008	0.000	0.002	0.012	0.001	0.015	0.014	0.001	0.001	0.000	0.000	0.002	0.000	0.001

OUT		**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
	1	0.195	0.018	0.026	0.001	0.009	0.025	0.000	0.010	0.215	0.020	0.029	0.001	0.001	0.030	0.001	0.001
	2	0.201	0.016	0.021	0.001	0.005	0.010	0.001	0.008	0.249	0.025	0.033	0.001	0.001	0.011	0.001	0.001
	3	0.006	0.001	0.001	0.000	0.000	0.001	0.000	0.000	0.007	0.001	0.001	0.000	0.000	0.001	0.000	0.000
	4	0.046	0.006	0.011	0.000	0.002	0.011	0.000	0.004	0.009	0.001	0.002	0.000	0.000	0.001	0.000	0.000
	5	0.203	0.017	0.007	0.001	0.005	0.027	0.002	0.006	0.263	0.025	0.007	0.001	0.001	0.038	0.001	0.001
	6	0.005	0.001	0.001	0.000	0.000	0.001	0.000	0.000	0.005	0.001	0.001	0.000	0.000	0.001	0.000	0.000
	7	0.051	0.005	0.008	0.000	0.001	0.011	0.000	0.014	0.008	0.001	0.001	0.000	0.000	0.001	0.000	0.000

		100 days								1000 days							
IN		**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
	1	0.099	0.023	0.040	0.001	0.000	0.051	0.002	0.001	0.011	0.027	0.013	0.000	0.000	0.047	0.001	0.001
	2	0.185	0.043	0.087	0.001	0.000	0.032	0.001	0.001	0.077	0.059	0.049	0.000	0.003	0.121	0.003	0.001
	3	0.001	0.000	0.001	0.000	0.000	0.002	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	4	0.001	0.000	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.001	0.001	0.000	0.000	0.000	0.000	0.000
	5	0.209	0.053	0.023	0.001	0.000	0.121	0.003	0.001	0.138	0.101	0.039	0.000	0.000	0.285	0.005	0.000
	6	0.005	0.001	0.003	0.000	0.000	0.003	0.000	0.000	0.001	0.003	0.003	0.000	0.000	0.008	0.000	0.000
	7	0.002	0.000	0.001	0.000	0.000	0.001	0.000	0.000	0.001	0.000	0.000	0.000	0.000	0.001	0.000	0.000

OUT		**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
	1	0.175	0.059	0.069	0.001	0.000	0.060	0.000	0.000	0.070	0.026	0.219	0.006	0.000	0.100	0.000	0.001
	2	0.182	0.056	0.083	0.002	0.000	0.017	0.001	0.000	0.061	0.025	0.269	0.015	0.000	0.035	0.001	0.000
	3	0.003	0.001	0.001	0.000	0.000	0.001	0.000	0.000	0.003	0.000	0.001	0.000	0.000	0.000	0.000	0.000
	4	0.001	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.001	0.001	0.000	0.000	0.000	0.000	0.000
	5	0.152	0.049	0.026	0.001	0.000	0.053	0.000	0.000	0.019	0.015	0.060	0.001	0.000	0.061	0.000	0.000
	6	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.001	0.000	0.000	0.000	0.000	0.000
	7	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.001	0.003	0.000	0.003	0.000	0.000

Table 4: An overview of the proportion of constraints that shift from one sequence pattern into another, both for incoming and outgoing constraints of the node with the highest authority score in the network.

3.2 Interpretation

Influence of degree From Tables 2 and 3, we can learn that most relationships between nodes in subsequent blocks vanish, however, this seems to be more of an issue for blocks that are longer (> 4 hours) especially for nodes with a lower degree (≤ 10). It seems a majority of the activity is relatively one-off. In general, all alternate relations happen sparsely given their very low share in the overall constraint evolution tally, meaning most nodes only interact once during a particular time block.

More interesting patterns can be observed when looking at how different constraint types evolve over time. A vast majority of chain precedence relations, a result of uninterrupted interaction between nodes, are replaced by not succession (4→1) and vice versa (1→4), and a shift from not succession to chain response (1→7) mainly for a degree lower or equal to 3 and 4 hour blocks and incoming relationships. The same trend holds for 2-1000 days, but here there is a higher rate of precedence to precedence (2→2). This indicates that in the short term, mostly the absence of response of the node (1) is eventually replaced by a very close interaction (4) in the short run, and that for the longer run (100-1000 days), the 2→2 indicates that the other party initiates a message often without there being a strict conversion towards other constraints such as response (5).

For nodes with a degree between 4 and 10, the evolutions look different with a strong persistence of chain response (7→7) in the short run (4 hours), and a similar (1/2→1/2) in the longer run. The former indicates that nodes tend to maintain a close interaction over short time spans, which is substituted by more sporadic interchange for longer time spans.

For nodes with a high degree, 7→7 for 4 hours, and 5→5 relations are present for a large share of the constraint evolutions. Hence, these nodes tend to maintain close contact at first, and afterwards remain the consequent in a response relationship indicate being especially forthcoming in terms of responding to other users.

Overall, users with a high degree tend to be more involved in mutually positive relationships (not 1), and reciprocate by being mostly involved in response relationships (5-7).

For outgoing constraints, we see a different picture, where incoming chain responses (7) are often converted into not succession (1) by the receiving party, which persists for 4 hours and 2 days at a degree of 3. For longer periods, 5→5 or, response to response is prevalent.

For nodes with a degree between 4 and 10, most initial relations end in not succession, later converging to a similar profile as nodes with a lower degree, and a high proportion of 5→5. The same holds for nodes with a high degree, where initially 7→7 is prevalent.

These observations clarify which nodes tend to be the source of cutting of contact in short time blocks (mostly with degree $\leq$10), and explain the interpretation of the incoming nodes earlier. In this respect, the evolution of constraints can be tracked between different types of nodes in terms of degree

Authority The node with the highest authority score which has a degree of 2,680 does not follow the trends of the other nodes discussed above. Again, the high proportion of 5→5 evolution is present for incoming relationships, but instead of materialising later on, this happens even for shorter time blocks (4 hours). This indicates that the authority will respond regardless of any established relationship.

4 Conclusion and future work

In this paper, we have shown how mining network interaction patterns can be profiled using sequence mining techniques. We apply the sequence mining method to the question-and-answer interaction-based network. Our preliminary results show that employing sequence patterns enables us track the behaviour of nodes in a transactional network and summarize their interactions without relying on the typical partial-order based results that are offered in sequence mining, while still going beyond the typical general nature of motifs that focus on directed arcs between 2 or 3 actors [11] . In a small experimental evaluation, we demonstrate the usefulness of the approach in the context of message board analysis.

For future work, we envision to focus on testing the patterns in the context of feature engineering, and link inference.

References

1. Bhagat, S., Cormode, G., Muthukrishnan, S.: Node classification in social networks. In: Social Network Data Analytics, pp. 115–148. Springer (2011)
2. Ciccio, C.D., Mecella, M.: A two-step fast algorithm for the automated discovery of declarative workflows. In: IEEE Symposium on Computational Intelligence and Data Mining, CIDM 2013, Singapore, 16-19 April, 2013. pp. 135–142 (2013)
3. De Smedt, J., Deeva, G., De Weerdt, J.: Behavioral constraint template-based sequence classification. In: ECML/PKDD (2). Lecture Notes in Computer Science, vol. 10535, pp. 20–36. Springer (2017)
4. Di Ciccio, C.D., Maggi, F.M., Mendling, J.: Efficient discovery of Target-Branched Declare constraints. Inf. Syst. 56, 258–283 (2016)
5. Ding, C.H.Q., Zha, H., He, X., Husbands, P., Simon, H.D.: Link analysis: Hubs and authorities on the world wide web. SIAM Review 46(2), 256–268 (2004)
6. Dwyer, M.B., Avrunin, G.S., Corbett, J.C.: Patterns in Property Specifications for Finite-State Verification. In: ICSE. pp. 411–420. ACM (1999)
7. Grover, A., Leskovec, J.: node2vec: Scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 855–864. ACM (2016)
8. Latapy, M., Viard, T., Magnien, C.: Stream graphs and link streams for the modeling of interactions over time. Social Netw. Analys. Mining 8(1), 61:1–61:29 (2018)
9. Liben-Nowell, D., Kleinberg, J.: The link-prediction problem for social networks. Journal of the American society for information science and technology 58(7), 1019–1031 (2007)

10. Milo, R., Shen-Orr, S., Itzkovitz, S., Kashtan, N., Chklovskii, D., Alon, U.: Network motifs: simple building blocks of complex networks. Science 298(5594), 824–827 (2002)
11. Paranjape, A., Benson, A.R., Leskovec, J.: Motifs in temporal networks. In: WSDM. pp. 601–610. ACM (2017)
12. Pesić, M.: Constraint-Based Work on Management Systems: Shifting Control to Users. Ph.D. thesis, PhD thesis, Eindhoven University of Technology, 2008. 26
13. Pesic, M., van der Aalst, W.M.P.: A Declarative Approach for Flexible Business Processes Management. In: Business Process Management Workshops. Lecture Notes in Computer Science, vol. 4103, pp. 169–180. Springer (2006)
14. Pons, P., Latapy, M.: Computing communities in large networks using random walks. In: International symposium on computer and information sciences. pp. 284–293. Springer (2005)
15. Westergaard, M., Stahl, C., Reijers, H.A.: Unconstrainedminer: efficient discovery of generalized declarative process models. BPM Center Report BPM-13-28, BPMcenter. org p. 28 (2013)