The 33rd Annual Conference of the Japanese Society for Artificial Intelligence, 2019

Using Sequence Constraints for Modelling Network Interactions

Johannes De Smedt[*1] Junichiro Mori[*2] Masanao Ochi[*2]

[*1] The University of Edinburgh [*2] The University of Tokyo

The ubiquitous nature of networks has led 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.

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 [Paranjape 17], and streams [Latapy 18] 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 real-life datasets. Section 4 concludes the paper and reports on the future directions.

2. Behavioural constraint patterns in networks

In this section, a detailed overview of the constraints is given, and how they can be leveraged for various network analysis applications.

Contact: Johannes De Smedt, The University of Edinburgh, johannes.desmedt@ed.ac.uk

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 [Dwyer 99]. 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. While initially proposed as LTL formulae which are convertible to Büchi automata, finite trace equivalent regular expressions were introduced in [Di Ciccio 13]. Models allowing for 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 are exhibit a hierarchy which is reported in [Di Ciccio 13] and which covers unordered up to chain ordered (using the next operator). Besides, the inclusion of negative constraints is unique, as typically only existing patterns are reported. 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).

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-

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

Template	Regular Expression
Existence(A,n)	.*(A.*){n}
Absence(A,n)	[^A]*(A?[^A]*){n-1}
Exactly(A,n)	[^A]*(A[^A]*){n}
Init(A)	(A.*)?
Last(A)	.*A
Responded existence(A,B)	[^A]*((A.*B.*) |(B.*A.*))?
Co-existence(A,B)	[^AB]*((A.*B.*) |(B.*A.*))?
Response(A,B)	[^A]*(A.*B)*[^A]*
Precedence(A,B)	[^B]*(A.*B)*[^B]*
Succession(A,B)	[^AB]*(A.*B)*[^AB]*
Alternate response(A,B)	[^A]*(A[^A]*B[^A]*)*
Alternate precedence(A,B)	[^B]*(A[^B]*B[^B]*)*
Alternate succession(A,B)	[^AB]*(A[^AB]*B[^AB]*)*
Chain response(A,B)	[^A]*(AB[^A]*)*
Chain precedence(A,B)	[^B]*(AB[^B]*)*
Chain succession(A,B)	[^AB]*(AB[^AB]*)*
Not co-existence(A,B)	[^AB]*((A[^B]*) |(B[^A]*))?
Not succession(A,B)	[^A]*(A[^B]*)*
Not chain succession(A,B)	[^A]*(A+[^AB][^A]*)*A*
Choice(A,B)	.*[AB].*
Exclusive choice(A,B)	([^B]*A[^B]*) |.*[AB].*([^A]*B[^A]*)

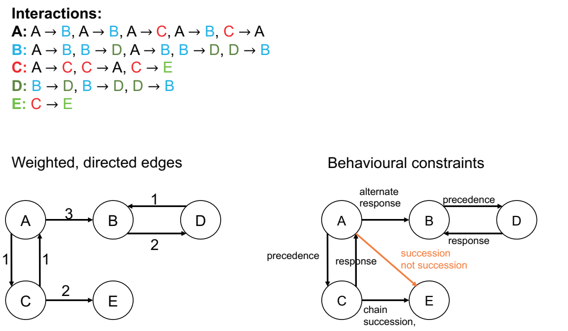

Figure 1: Running example

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 [De Smedt 17], we can mine all patterns in a 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 does 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 $C_n = \{c \mid c \in C_n \land c \in C_{n_i} \forall n_i \in \mathcal{N}(n) \lor c \notin C_n \land c \notin C_{n_i} \forall n_i \in \mathcal{N}(n)\}$ 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 analyzing the interactions' 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.

Next, the sequence patterns can be used as features of a node. In this case, also unary constraints help define the node in terms of where in a sequence, how often, and with what other nodes the node is interacting. The features can be used towards node classification [Bhagat 11]. Finally, by using the transitivity properties of the constraints, link inference/prediction [Liben 07] can also be made.

3. Results

We apply the sequence method to the Math Overflow dataset, as used in [Paranjape 17]. 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 vs question, (2) comments on user vs question, or (3) comments on user vs 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, 2 days, 100 days, and 1,000 days in order to track the evolution of the constraints. For this analysis, we limit the constraint set to the 7 most common sequence patterns. In order to illustrate the usefulness of the results, we focus on two active users with a different background. The first user (denoted B) is considered an authority as that node in the network has the highest authority score [Ding 04]. The high authority is pointed to by many high hubs and high hub points to many high authorities. Authority and hub scores are obtained by this iterative scoring.

Table 2: An overview of the proportion of constraints that shift from one sequence pattern into another, both for incoming and outgoing constraints of nodes A and B. The colours denote the place in the distribution, where red is higher and green lower. Scores with different colours and equal scores indicate a difference in value behind the significant digits.

	In								Out							
4 hours - A	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
NotSuc (1)	0.15	0.04	0.01	0.00	0.02	0.02	0.00	0.01	0.14	0.05	0.05	0.00	0.02	0.02	0.00	0.04
Prec (2)	0.14	0.03	0.02	0.00	0.01	0.01	0.00	0.01	0.13	0.01	0.03	0.00	0.01	0.02	0.00	0.02
AltPrec (3)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01
ChainPrec (4)	0.03	0.03	0.02	0.00	0.03	0.02	0.00	0.00	0.09	0.02	0.05	0.00	0.02	0.03	0.00	0.04
Resp (5)	0.15	0.02	0.01	0.00	0.01	0.03	0.00	0.01	0.11	0.01	0.00	0.00	0.00	0.03	0.00	0.00
AltRes (6)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
ChainRes (7)	0.04	0.05	0.01	0.00	0.01	0.02	0.00	0.04	0.13	0.00	0.02	0.00	0.01	0.02	0.00	0.15
4 hours - B																
NotSuc	0.17	0.02	0.02	0.00	0.01	0.01	0.00	0.01	0.21	0.02	0.03	0.00	0.01	0.02	0.00	0.01
Prec	0.22	0.02	0.02	0.00	0.01	0.01	0.00	0.01	0.19	0.01	0.02	0.00	0.00	0.01	0.00	0.01
AltPrec	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
ChainPrec	0.05	0.01	0.01	0.00	0.00	0.01	0.00	0.01	0.06	0.01	0.01	0.00	0.00	0.01	0.00	0.01
Resp	0.21	0.02	0.01	0.00	0.00	0.02	0.00	0.01	0.20	0.02	0.01	0.00	0.00	0.02	0.00	0.00
AltRes	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
ChainRes	0.07	0.01	0.01	0.00	0.00	0.01	0.00	0.03	0.06	0.01	0.01	0.00	0.00	0.02	0.00	0.02
2 days - A																
NotSuc	0.19	0.01	0.01	0.00	0.00	0.01	0.00	0.00	0.26	0.01	0.01	0.00	0.00	0.01	0.00	0.00
Prec	0.32	0.01	0.02	0.00	0.00	0.00	0.00	0.00	0.24	0.01	0.01	0.00	0.00	0.01	0.00	0.00
AltPrec	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
ChainPrec	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Resp	0.33	0.01	0.00	0.00	0.00	0.02	0.00	0.00	0.25	0.01	0.00	0.00	0.00	0.02	0.00	0.00
AltRes	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
ChainRes	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2 days - B																
NotSuc	0.21	0.02	0.02	0.00	0.00	0.02	0.00	0.00	0.23	0.01	0.02	0.00	0.00	0.02	0.00	0.00
Prec	0.29	0.02	0.02	0.00	0.00	0.01	0.00	0.00	0.27	0.02	0.02	0.00	0.00	0.01	0.00	0.00
AltPrec	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00
ChainPrec	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Resp	0.29	0.02	0.01	0.00	0.00	0.02	0.00	0.00	0.28	0.02	0.00	0.00	0.00	0.03	0.00	0.00
AltRes	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00
ChainRes	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Table 3: Similar overview as Table 3 containing the 100 and 1,000 days time frames.

	In								Out							
100 days - A	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
NotSuc	0.13	0.02	0.03	0.00	0.00	0.04	0.00	0.00	0.18	0.04	0.04	0.00	0.00	0.04	0.00	0.00
Prec	0.24	0.04	0.07	0.00	0.00	0.02	0.00	0.00	0.19	0.04	0.05	0.00	0.00	0.01	0.00	0.00
AltPrec	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00
ChainPrec	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Resp	0.25	0.04	0.02	0.00	0.00	0.07	0.00	0.00	0.18	0.04	0.01	0.00	0.00	0.04	0.00	0.00
AltRes	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
ChainRes	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
100 days - B																
NotSuc	0.10	0.02	0.04	0.00	0.00	0.05	0.00	0.00	0.15	0.05	0.06	0.00	0.00	0.05	0.00	0.00
Prec	0.19	0.04	0.08	0.00	0.00	0.03	0.00	0.00	0.16	0.05	0.07	0.00	0.00	0.02	0.00	0.00
AltPrec	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
ChainPrec	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Resp	0.22	0.05	0.02	0.00	0.00	0.11	0.00	0.00	0.13	0.04	0.02	0.00	0.00	0.05	0.00	0.00
AltRes	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
ChainRes	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1000 days - A																
NotSuc	0.07	0.03	0.06	0.00	0.00	0.07	0.00	0.00	0.11	0.04	0.07	0.00	0.00	0.06	0.00	0.00
Prec	0.14	0.04	0.09	0.00	0.00	0.06	0.00	0.00	0.12	0.06	0.09	0.00	0.00	0.03	0.00	0.00
AltPrec	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
ChainPrec	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Resp	0.17	0.06	0.04	0.00	0.00	0.14	0.00	0.00	0.10	0.05	0.04	0.00	0.00	0.07	0.00	0.00
AltRes	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
ChainRes	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1000 days - B																
NotSuc	0.01	0.03	0.01	0.00	0.00	0.05	0.00	0.00	0.06	0.02	0.19	0.01	0.00	0.09	0.00	0.00
Prec	0.08	0.06	0.05	0.00	0.00	0.12	0.00	0.00	0.05	0.02	0.23	0.01	0.00	0.03	0.00	0.00
AltPrec	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
ChainPrec	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Resp	0.14	0.10	0.04	0.00	0.00	0.29	0.01	0.00	0.02	0.01	0.05	0.00	0.00	0.05	0.00	0.00
AltRes	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
ChainRes	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

The other user (denoted A) has a similarly high degree (high number of connections in the network), but a lower authority score. Our hypothesis is that the interactions of the authority user result in several constraint patterns as he gains the authority through answering and commenting to questions within his expertise.

The results of the shifts in constraint patterns as expressed in their proportions, are included in Tables 2 and 3 for both incoming and outgoing constraints of both nodes. The cells indicate the proportion of connections between the same nodes that are both present again in two subsequent time frames that shifted from the template in the rows, to the template in the columns. '0' signifies that the constraint is no longer present between both rows.

Firstly, it can be seen that there is a high number of constraints that are not reoccurring over time, meaning they are not repeated in the subsequent time frame. This behaviour is expected, given that many question-answering threads stop after a few posts, and many users only tend to intervene in a limited number of threads. Considering different lengths of time frames, however, we note that especially for node B (the authority) the number of vanishing interactions is drastically lower for 1,000 days. In case of incoming constraints, we see many re-occurring response constraints, and with outgoing ones we see many precedence and not succession constraints appearing. This is in line with how we would expect question-answering is handled by an authority, who responds to all questions within his area of expertise.

Overall, the two nodes behave relatively similarly in terms of proportions of constraints up until the 1,000 days threshold. The change incurred by increasing the time frames does not yield drastically different results, but it can be noted that more connections are reoccurring (mostly response and precedence relationships) rather than vanishing (as captured by column '0'). Hence, nodes that are surviving longer, and hence are reoccurring themselves, seem to maintain their relations over time. Also, any 'stronger' constraints that model alternating or chain relations are very often not present. One final observation is interesting. The high number of chain response connections that are going out from node A indicates that many immediate answer-response messages were exchanged over a period of 4 hours, indicating that single conversations where picked up of which many reoccurred as well.

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 [Paranjape 17] . 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

[Paranjape 17] Paranjape, A., Benson, A. R., & Leskovec, J.: Motifs in temporal networks, Proceedings of the Tenth ACM International Conference on Web Search and Data Mining (2017)

[Namaki 17] Namaki, M. H., Wu, Y., Song, Q., Lin, P., & e, T.: Discovering graph temporal association rules, Proceedings of the 2017 ACM Conference on Information and Knowledge Management (2017)

[Dwyer 99] Dwyer, M. B., Avrunin, G. S., & Corbett, J. C.: Patterns in property specifications for finite-state verification, Proceedings of the 21st international conference on Software engineering (1999)

[Latapy 18] Latapy, M., Viard, T., & Magnien, C.: Stream graphs and link streams for the modeling of interactions over time. Social Network Analysis and Mining, 8(1) (2018)

[Di Ciccio 13] Di Ciccio, C., & Mecella, M.: A two-step fast algorithm for the automated discovery of declarative workflows, 2013 IEEE Symposium on Computational Intelligence and Data Mining (2013)

[De Smedt 17] De Smedt, J., Deeva, G., & De Weerdt, J.: Behavioral Constraint Template-Based Sequence Classification, European Conference on Machine Learning (2017)

[Ding 04] Ding, C. H., Zha, H., He, X., Husbands, P., & Simon, H. D.: Link analysis: hubs and authorities on the World Wide Web, SIAM review, 46(2) (2004)

[Bhagat 11] Bhagat, S., Cormode, G., & Muthukrishnan, S.: Node classification in social networks, Social network data analytics, Springer (2011)

[Liben 07] LibenNowell, D., & Kleinberg, J.: The linkprediction problem for social networks, Journal of the American society for information science and technology, 58(7) (2007)