Discovering Evolution Chains in Dynamic Networks

Corrado Loglisci, Michelangelo Ceci, Donato Malerba

Dipartimento di Informatica, Università degli Studi di Bari "Aldo Moro"
via Orabona, 4 - 70126 Bari - Italy
{corrado.loglisci, michelangelo.ceci, donato.malerba}@uniba.it

Abstract. Most of the works on learning from networked data assume that the network is static. In this paper we consider a different scenario, where the network is dynamic, i.e. nodes/relationships can be added or removed and relationships can change in their type over time. We assume that the "core" of the network is more stable than the "marginal" part of the network, nevertheless it can change with time. These changes are of interest for this work, since they reflect a crucial step in the network evolution. Indeed, we tackle the problem of discovering evolution chains, which express the temporal evolution of the "core" of the network. To describe the "core" of the network, we follow a frequent pattern-mining approach, with the critical difference that the frequency of a pattern is computed along a time-period and not on a static dataset. The proposed method proceeds in two steps: 1) identification of changes through the discovery of emerging patterns; 2) composition of evolution chains by joining emerging patterns. We test the effectiveness of the method on both real and synthetic data.

1 Introduction

In recent years, there has been a constantly growing interest in learning from networked data [5]. This is due to the fact that in many application domains, data naturally come in the form of a network, such as in protein interaction networks, social networks, linked web documents, and co-author networks, just to mention some of the most prominent examples. Any dataset represented as a set of relations and foreign key constraints in a relational database can be naturally represented as a network, which makes learning algorithms developed for networked data naturally applicable to any relational database. For the same reason, this class of algorithms is applicable to spatial data which are characterized by spatial relationships (e.g., topological, directional and distance-based relationships), although in this case the additional challenge comes from the fact that the (many) spatial relationships are *implicit* in the data [7].

Most of the algorithms developed to learn or analyze networked data assume that the network is static and unchangeable, i.e., the structure and the properties of a network do not vary over time. This assumption seems to be too restrictive in real scenarios where networks can be dynamic and exhibit changes especially

when modeling phenomena which evolve over time. In particular, nodes and edges of the networks may appear and disappear over time and relationships can change in their nature.

The importance of knowledge discovery from dynamic networks has been recognized only recently; hence the body of methods and techniques for the analysis of dynamic networks is much less developed than for static networks. Research on learning from dynamic networks follows three main lines: *i)* detection of communities over time, *ii)* characterization of the evolution of the networks, and *iii)* prediction of nodes/edges of the networks. Sun et al. [8] propose a technique to discover communities and detect changes in dynamic graphs represented in the form of contingency matrices with encoding schemes. A different approach principled on frequent graph-based patterns is reported in [2], where the representation of the time-evolving graphs as a sequence of cumulative graphs enables the discovery of rules which characterize the evolution of the network in terms of topological changes. Algorithms developed for predictive tasks are quite recent. Most of them try to make inferences at a specific time point, typically the time point next to the last observed. For instance, in [10] a hybrid framework combines the temporal information with topological patterns and a probabilistic relational model to infer the existence of links in social networks.

In this paper, we tackle a different task whose goal is to discover *evolution chains*, which express the temporal evolution of patterns, here intended as sets of labelled edges of the dynamic network. Indeed, in some applications where the network can be observed at different time-points, it is important to discover what is the "core" portion of the network which changes with time. We assume that the "core" of the network is more stable than the "marginal" part of the network, nevertheless it can change with time. These changes are of interest for this work, since they reflect a crucial step in the network evolution. To identify patterns concerning the "core" of the network, we follow a frequent pattern-mining approach, with the critical difference that the frequency of a pattern is computed along a time-period and not on a static dataset. In other terms, we assume that "frequent" patterns (along a time-period) do capture the more stable structure of the "core" of the network, while the many "infrequent" patterns do represent marginal aspects of the dynamic network, which are much more unstable and, thus, less interesting.

An example of evolution chain which can be extracted in the context of social network analysis is the following:

$\{(user\_a, user\_b, friendship), (user\_b, user\_c, participation\_to\_same\_event)\}^{\tau_1}$

$\{(user\_a, user\_b, friendship), (user\_b, user\_c, membership\_to\_a\_group)\}^{\tau_2}$

$\{(user\_b, user\_c, membership\_to\_a\_group), (user\_c, user\_d, publish\_on\_the\_wall\_of)\}^{\tau_3}$

It includes three frequent patterns discovered in as many consecutive time-periods (τ_1, τ_2 and τ_3). These patterns express topological regularities of the network. Nevertheless, not all frequent patterns are taken, but only those which meet two conditions:

i) their frequency significantly changes between the considered time-period and the previous one;

ii) the "similarity" between a pattern and the pattern associated with the previous time-period is maximized.

Thus, as a time-period is observed, we extract *emerging* patterns from that period and we incrementally join them with the sequence of patterns generated in the previous time-periods in order to generate complete evolution chains. In order to consider conservative periods, in the joining process we can join frequent patters extracted from non-consecutive time-periods if, in the intermediate periods, they do not show any changes in frequency.

The paper is organized as follows. In Section 2 we formally define the problem of discovering evolution chains. The proposed computational solution is reported in Section 3. In Section 4 we report and discuss experimental results on both real and synthetic data. Finally, conclusions are drawn.

2 Problem Formulation

Before formally defining the problem we intend to solve, some definitions are necessary. Let $D = \langle D_1, D_2, \ldots, D_n \rangle$ be a sequence of time-ordered observations of the network, obtained at regular time points. At each time-point t_i, the network is described by the set $D_i = (N_i \times N_i, E_i)$, $N_i \subseteq \mathcal{N}$ and $E_i \subseteq \mathcal{E}$, where $\mathcal{N}$ and $\mathcal{E}$ denote the sets of the nodes and types of edges observed in $\{t_1 \ldots t_n\}$, respectively. An edge is modelled as a triple (n_1, n_2, e_{12}), where n_1 and n_2 are the connected nodes while e_{12} is a label denoting the edge type.[1] We say that an edge (n_1, n_2, e_{12}) *occurs* at a time-point t_i if the observation $D_i = (N_i \times N_i, E_i)$ includes it, i.e., $(n_1, n_2, e_{12}) \in D_i$.

In this work, a *pattern* P is a set of edges. We say that P *occurs* at a time-point t_i if all edges in P occur at the same time-point t_i. To give the definition of frequent pattern in a temporal interval, we first introduce the concept of time-period.

A *time-period* (or simply *period*) τ in $\{t_1 \ldots t_n\}$ is a sequence of consecutive time-points $\{t_i, \ldots, t_j\}$ $(t_1 \leq t_i,\ t_j \leq t_n)$. The width w of τ is the number of time-points in τ, i.e. $w = |\{t_i, \ldots, t_j\}|$. Here we assume that all the periods have the same width w. Two periods $\tau = \{t_i, \ldots, t_{i+w}\}$ and $\tau' = \{t_{i+w+1} \ldots t_{i+2w}\}$ are said *consecutive*. As we consider consecutive time-periods, we can enumerate them and use the notation τ_{h+1} to indicate a period consecutive to τ_h.

Definition 1. *Given a time-period τ_h and a pattern P_{τ_h} we say that P_{τ_h} is* frequent *in τ_h if it occurs in at least minSupp time points of τ_h.*

We consider the relative frequency of P_{τ_h} computed as the number of time points in τ_h in which P_{τ_h} occurs, divided by the width of τ_h (i.e. $j - i + 1$). On the basis of the concept of frequent pattern, we can give the following definition:

[1] We assume that two nodes can be connected by multiple edges of different types and that edges are not symmetric.

Definition 2. *Given a node $X \in \mathcal{N}$, an* evolution chain L_X *is a sequence of frequent patterns $\langle P_{\tau_h}, P_{\tau_{h+1}}, \ldots, P_{\tau_{h+v}} \rangle$ where the node X belongs to some triple in P_{τ_h} and for each $i = 0, \ldots, v-1$, $P_{\tau_{h+i+1}}$ differs from $P_{\tau_{h+i}}$ in only one triple. The sequence $\tau_{h+1}, \ldots, \tau_{h+v}$ is called* supporting period *for the evolution chain.*

Intuitively, the patterns $P_{\tau_{h+i}}$ and $P_{\tau_{h+i+1}}$ represent a relevant state of the network in the two consecutive periods τ_{h+i} and τ_{h+i+1}. The fact that $P_{\tau_{h+i+1}}$ differs from $P_{\tau_{h+i}}$ in only one triple guarantees that an evolution chains catches only slight differences in the structure of the patterns. This is coherent with condition *ii)*, according to which the "similarity" between a pattern and the pattern associated with the previous time-period is maximized.

The problem we intend to solve can be formalized as follows:
Given: the set of time-stamped observations of the network $D = \langle D_1, D_2, \ldots, D_n \rangle$, a set of consecutive time-periods $\tau_1, \ldots, \tau_m$ $(n \gg m)$, and a node $X \in \mathcal{N}$,
Find: the set of evolution chains $\mathcal{L}_X = \{L_X\}$ whose supporting periods are included in $\tau_1, \ldots, \tau_m$.

A computational solution to this problem is described in the following.

3 The method

The proposed solution is structured in two-steps. The first step aims to discover emerging patterns, while the second step incrementally joins the extracted patterns in order to compose, through the periods, evolution chains. The two steps are detailed in the following.

3.1 Emerging Patterns to Represent Dynamic Networks

Emerging patterns (EPs) [6] are a particular kind of frequent patterns (FPs) used to characterize a partition of the data with respect to other partitions. The main property is that their support (relative frequency) significantly changes from one partition to another one. The greater the change of the support of a pattern, the more interesting the pattern. Changes in the support are quantitatively estimated in terms of growth rate (GR), which is a frequency ratio computed as the ratio $GR(P) = supp_{partition_i}(P)/supp_{partition_j}(P)$, where $supp_{partition_i}(P)$ is the support of the pattern P in the partition i and $supp_{partition_j}(P)$ is the support of P in the partition j. Examples of the application of emerging patterns in the spatio-temporal context can be found in [4],[3].

In our context, EPs are used to characterize the changes that the network may exhibit in a time-period with respect to the previous time-period both in the co-occurrences of the edges and in the presence of types of edges. In particular, EPs are discovered by evaluating the FPs generated in the period τ_i (FP_i) against those generated in the previous period τ_{i-1} (FP_{i-1}). Each pattern P of FP_i becomes emerging if it satisfies the following conditions:

- it differs, for only one triple, from at least one of the patterns of FP_{i-1};

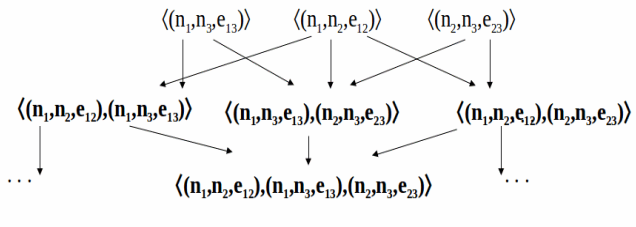


Fig. 1. The lattice generated during the process of frequent pattern mining.

– there exists a pattern $P' \in FP_{i-1}$, $P' \neq P$ such that

$$GR(P, P') = supp_{\tau_i}(P)/supp_{\tau_{i-1}}(P') \geq minGR,$$

where $minGR$ is a user-defined threshold.

Note that the above definition of EP differs from the classical definition which captures differences in the support on the same pattern, and not on "slightly" different patterns. However, this divergence is necessary to catch evolutions which, otherwise, would not be detected.

Algorithmically, FPs are discovered in each period by exploiting the well-known *Apriori* algorithm [1]. In addition to frequent patterns, the Apriori algorithm returns also a graph-based structure (lattice) whose nodes correspond to possibly generated patterns while the edges denote a subset relationship among the connected patterns. More precisely, each edge connects a pattern P of length k to k patterns $Q_1, \ldots, Q_k$ of length $k-1$ which are subsets of P. In Figure 1, the pattern $\langle (n_1, n_2, e_{12}), (n_1, n_3, e_{13}), (n_2, n_3, e_{23}) \rangle$ of length 3 is connected with the three patterns $\langle (n_1, n_2, e_{12}), (n_1, n_3, e_{13}) \rangle$, $\langle (n_1, n_3, e_{13}), (n_2, n_3, e_{23}) \rangle$, $\langle (n_1, n_2, e_{12}), (n_2, n_3, e_{23}) \rangle$. In the process of frequent pattern mining, it holds the *anti-monotonicity* of the support according to which if P is frequent then $Q_1, \ldots, Q_k$ are also frequent, while if one among $Q_1, \ldots, Q_k$ is infrequent, then P is infrequent too. This property is exploited in order to: *i)* generate k-length patterns from frequent $(k-1)$-length patterns, and *ii)* avoid to generate k-length patterns from infrequent $(k-1)$-length patterns.

In our approach, we exploit the anti-monotonicity property in the process of extracting emerging patterns. In particular, for each frequent k-length pattern P in FP_i, we consider its *(k-1)*-length patterns $Q_1, \ldots, Q_k$ and we retrieve them from the frequent *(k-1)*-length patterns in FP_{i-1}. From the retrieved set of frequent patterns in FP_{i-1}, we identify their corresponding *(k)*-length patterns in FP_{i-1}. The set of patterns obtained in this way (denoted as $\mathcal{P}_P, \tau_{i-1}$), contains patterns which have the same length of P, share $k-1$ triples with P, and are frequent in τ_{i-1}. Then, P is considered to be an emerging pattern if $\exists P' \in \mathcal{P}_P, \tau_{i-1},\ P' \neq P,\ s.t.\ GR(P, P') = supp_{\tau_i}(P)/supp_{\tau_{i-1}}(P') \geq minGR$.

A concrete example is reported in Figure 2. Let the pattern $P = \langle (n_1, n_2, e_{12}), (n_1, n_3, e_{13}), (n_2, n_3, e_{23}) \rangle$ (k=3) be frequent in the period τ_i. The patterns of

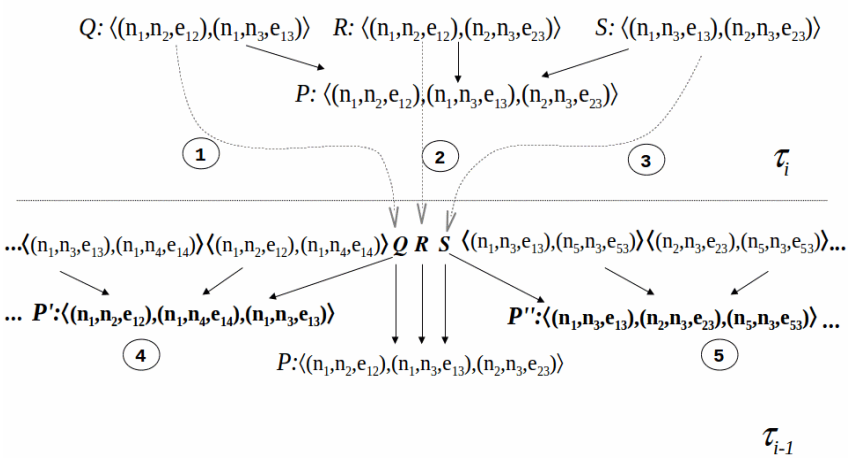


Fig. 2. Emerging patterns are selected among the frequent patterns which differ in one triple only.

length k=2 connected to P are $Q : \langle(n_1, n_2, e_{12}), (n_1, n_3, e_{13})\rangle$, $R : \langle(n_1, n_2, e_{12}), (n_2, n_3, e_{23})\rangle$ and $S : \langle(n_1, n_3, e_{13}), (n_2, n_3, e_{23})\rangle$.

Then, they are searched (circles 1,2,3) in the lattice of the period τ_{i-1} and, once found, are used to retrieve the set $\mathcal{P}_{P,\tau_{i-1}}$ of frequent k-length patterns which may have been derived by joining Q, R, S with other $(k-1)$-length patterns. In the example[2], k-length patterns are $P : \langle(n_1, n_2, e_{12}), (n_1, n_3, e_{13}), (n_2, n_3, e_{23})\rangle$, $P' : \langle(n_1, n_2, e_{12}), (n_1, n_4, e_{14}), (n_1, n_3, e_{13})\rangle$ and $P'' : \langle(n_1, n_3, e_{13}), (n_2, n_3, e_{23}), (n_5, n_3, e_{53})\rangle$. Finally, we determine the growth-rate $supp_{\tau_i}(P)/supp_{\tau_{i-1}}(P')$ and $supp_{\tau_i}(P)/supp_{\tau_{i-1}}(P'')$: if at least one of these values exceeds the threshold $minGR$, we consider P as emerging.

3.2 Discovering Evolution Chains as Incremental Join of EPs

In order to formally define the problem of discovering evolution chains, some preliminary definitions have to be introduced. Let $\mathcal{S}_\mathcal{N} : \mathcal{N} \times \mathcal{N} \to [0, 1]$ and $\mathcal{S}_\mathcal{E} : \mathcal{E} \times \mathcal{E} \to [0, 1]$ be two similarity functions between nodes and types of edges, respectively. These two functions return real values and are here considered as background knowledge for the investigated problem. They can naturally model similarity among types of edges or similarity between types of nodes in heterogeneous networks (in the social networks, the similarity between the edges corresponding to "friendship" and "membership to the same group" can be 0.9).

Their availability is a quite reasonable assumption since in real-world networks we can easily define notions of similarity on nodes and types of edges.

[2] For the sake of simplicity, in Figure 2 we do not report all the patterns which are derived by joining Q, R, S with other $(k-1)$-length patterns.

Considering the notions introduced so far, the problem of discovering evolution chains in a dynamic network can be so formulated:
Given: the set of sets of EPs $\mathcal{P}$ mined in the periods $\tau_1, \ldots, \tau_m$; $\mathcal{S}_\mathcal{N}$, $\mathcal{S}_\mathcal{E}$ two similarity measures on $\mathcal{N}$ and $\mathcal{E}$ respectively; $\sigma_\mathcal{N}$ and $\sigma_\mathcal{E}$ two minimum similarity thresholds for $\mathcal{S}_\mathcal{N}$ and $\mathcal{S}_\mathcal{E}$ respectively; a node $X \in \mathcal{N}$
Find: the set of evolution chains $\mathcal{L}_X$.

The intuition behind the solution here proposed is that dynamic networks actually may exhibit topological changes in some parts (nodes and relationships can be added/removed and relationships can change in their type over time) while keeping others unchanged, especially between adjacent periods. We use this intuition by considering as valid those chains which connect both unchanged and changed parts of the network. This is coherent with our assumption that the "core" of the network is more stable than the marginal part which makes our approach particularly adequate for networks that exhibit concept drift rather than concept shift [9].

The proposed solution joins EPs in adjacent periods (τ_{i-1}, τ_i) only if they have the same length (v edges) and differ for one edge: $v-1$ edges would represent the unchanged part of the network while the two different edges (one for each FP used in the construction of the EP) would denote the changed part. It is noteworthy that this does not inhibit our approach from considering multiple changes in the same network, since multiple EPs can extracted. When several patterns are candidates to be used for the join operation, we exploit the notion of similarity for nodes and types of edges by joining the candidate for which the new edge is "enough" similar to the removed one. Similarity is the average pairwise similarity between the nodes and the types of edges (computed according to $\mathcal{S}_\mathcal{N}$ and $\mathcal{S}_\mathcal{E}$). Indeed, in real-world dynamic networks, we do not expect drastic changes in adjacent periods but rather mild changes which could be originated from slight variations on the topological aspects and on the occurrences of the edges. The integration of the similarity measures $\mathcal{S}_\mathcal{N}$, $\mathcal{S}_\mathcal{E}$ allows us also to prevent the generation of meaningless and noise evolution chains.

In order to build chains and, at the same time, guarantee the completeness of the results, the approach adopts two mechanisms of space search:

- *backtracking*, which, starting from chains discovered until the previous time-period, explores backward the EPs of the previous periods in order to identify alternative chains;
- *skipping*, which, considering the possibility that EPs of adjacent periods could be not joined, analyzes forward the remaining periods in order to find EPs suitable for joining.

Indeed, this inability to join EPs in adjacent periods could be due to different factors: i) the nodes and the edges of the EPs might exhibit low similarity which does not exceed the minimum threshold, ii) EPs might present completely different edges or, conversely, identical edges. Indeed, when the FPs (in adjacent time-periods) are completely different, we cannot identify the unchanged parts of the network (as described above). On the contrary, when the FPs are the same and no EP can joined, we cannot identify the changed parts of the network.

Algorithm 1 Discovering Evolution Chains

1: **input:** $\mathcal{P}, \mathcal{S}_\mathcal{N}, \mathcal{S}_\mathcal{E}, \sigma_\mathcal{N}, \sigma_\mathcal{E}$, $minGR$, $X \in \mathcal{N}$
output: $\mathcal{L}_\mathcal{X}$
2: $found := false$; $h := 1$; $candidates := \oslash$;
3: **while** *not found* **do**
4: **for all** $EP \in getEPs(\mathcal{P}, h)$ **do**
5: **if** $contains(X, EP)$ **then**
6: $candidates := EP$
7: $found := true$
8: **end if**
9: **end for**
10: **if** $candidates = \oslash$ **then**
11: $h := h + 1$
12: **else**
13: $selected := \underset{EP \in candidates}{\arg\min} \; length(EP)$
14: $selected := \underset{EP \in selected}{\arg\max} \; Growth\_Rate(EP)$
15: **end if**
16: **end while**
17: $i := h + 1$
18: $push(\tau\_stack, i)$
19: $mark(selected)$
20: $push(EP\_stack, selected)$
21: **while** $EP\_stack <> \oslash$ **do**
22: $\mathcal{L}_X \leftarrow FWjoin(\tau\_stack, EP\_stack, \mathcal{P}, \mathcal{S}_\mathcal{N}, \mathcal{S}_\mathcal{E}, \sigma_\mathcal{N}, \sigma_\mathcal{E}, minGR, \mathcal{L}_X)$
23: $pop(EP\_stack)$
24: $pop(\tau\_stack)$
25: **end while**

The algorithmic description is reported in Algorithms 1 and 2. In order to clarify how they work, we report an explanatory example in Figure 3. Consider τ_1, τ_2, τ_3 as time-periods, the input node X as n_1 and the thresholds $\sigma_\mathcal{N}$=$\sigma_\mathcal{E}$=σ=0.25. The similarity measures $\mathcal{S}_\mathcal{N}, \mathcal{S}_\mathcal{E}$ return values reported in Figure 3a). The reflexive similarity of nodes and edges (e.g., (n_1, n_1)) is 1. As to the Algorithm 1, the first operation (lines 3-16) aims at finding the first period where X occurs. In the example, the search starts from τ_1 and finds n_1 in the EPs of τ_1. If n_1 had not been found there, the search would have proceeded through the next periods. The presence of n_1 in a set of EPs (*candidates*) leads to select only one EP with minimum length and maximum growth rate (lines 13-14), that is, FP_1 in Figure 3b. The use of a length-based selection criterion is justified by the fact that the anti-monotonicity property of the support guarantees that the shorter the pattern, the higher the frequency, and the better the pattern represents the network. Moreover, the selection by growth rate allows us to consider EPs which better represent the changes in the network between the previous period and the current one. However, only when X is found in the first period of the network τ_1, as in the current example, the selection is performed on the set of FPs and considers the length and support of the patterns (circles A,B in Figure 3b). This because we cannot discover EP in the first period τ_1 since EPs are determined by evaluating FPs in τ_i against the FPs in τ_{i-1}.

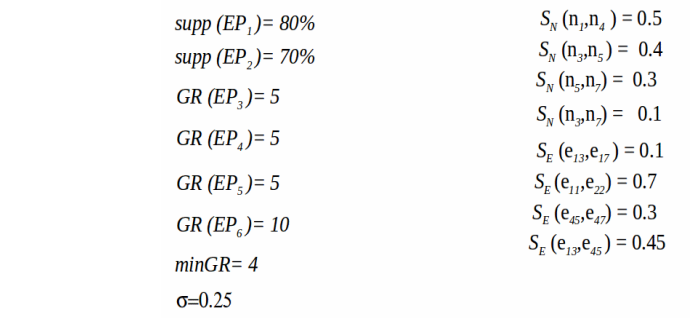
The EPs selected at the lines 13-14 and the associated period are stored in two stack structures (lines 18,20) which will be used to explore (in forward and backward mode) the next periods and the EPs there discovered.

Algorithm 2 Incremental Forward Join of EPs ($FWjoin$)

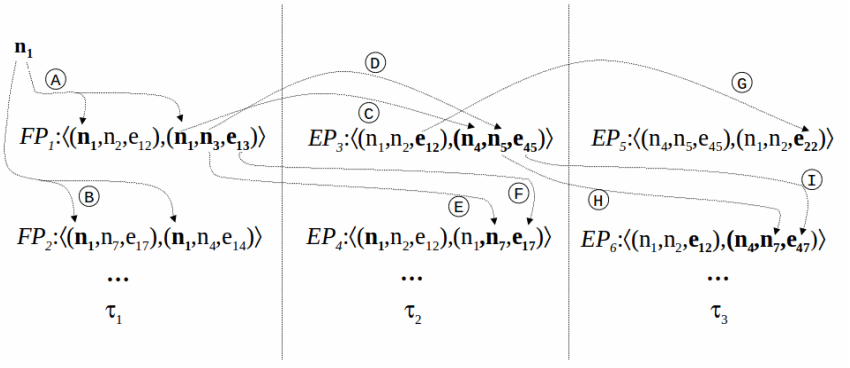
1: **input:** $\tau\_stack, EP\_stack, \mathcal{P}, \mathcal{S}_\mathcal{N}, \mathcal{S}_\mathcal{E}, \sigma_\mathcal{N}, \sigma_\mathcal{E}, minGR$
output: $\mathcal{L}_X$
2: $i := pop(\tau\_stack)$
3: $m := |\mathcal{P}|$ {Number of time-periods}
4: $selected\_EP := pop(EP\_stack)$
5: **while** $i \leq m$ **do**
6: $candidates \leftarrow getEPs(\mathcal{P}, i)$
7: $candidates \leftarrow select\_by\_length(candidates, selected\_EP)$
8: $candidates \leftarrow select\_by\_triples(candidates, selected\_EP)$
9: $candidates \leftarrow remove\_marked(candidates)$
10: $candidates \leftarrow select\_by\_similarity(candidates, selected\_EP, \mathcal{S}_\mathcal{N}, \mathcal{S}_\mathcal{E}, \sigma_\mathcal{N}, \sigma_\mathcal{E})$
11: **for all** $EP \in candidates$ **do**
12: **if** $i < m$ **then**
13: $candidate := \underset{EP \in candidates}{\arg\max} \; Growth\_Rate(EP)$
14: $push(\tau\_stack, i)$
15: $mark(candidate)$
16: $push(EP\_stack, candidate)$
17: $selected\_EP := candidate$
18: $\mathcal{L}_X \leftarrow \mathcal{L}_X \cup \; concatenate(\mathcal{L}_X, selected\_EP, \{candidate\})$
19: **else**
20: $candidates \leftarrow select\_by\_minGR(candidates, minGR)$
21: $\mathcal{L}_X \leftarrow \mathcal{L}_X \cup \; concatenate(\mathcal{L}_X, selected\_EP, candidates)$
22: **end if**
23: **end for**
24: $i = i + 1$
25: **end while**

Once the latest selected EP is stored in the stack, it is considered for the possible join with EPs of the next periods (forward mode - lines 2,4 of the algorithm $FWjoin$). The operation is performed by first selecting, among the FPs identified as emerging (see Section 3.1), those which have the same length of $selected\_EP$ and differ from it in only one edge (lines 6-8). In the case no EP is found, the process *skips* the current period τ_i and continues the search in the next period (line 24). In the example, we have EP_3 and EP_4 (discovered in τ_2) which are identified as candidates to be joined with FP_1 ($selected\_EP$), since they differ for only one triple from FP_1 (in Figure 3b, differences are represented by means of the circles C,D and E,F, respectively). Only one pattern among EP_3 and EP_4, identified with the similarity measures and the growth rate values, will be used for the join with FP_1 (lines 10,13 and 20).

The measures $\mathcal{S}_\mathcal{N}, \mathcal{S}_\mathcal{E}$ are used to determine the similarity of the pairs of patterns (FP_1, EP_3) and (FP_1, EP_4) by considering the similarity among nodes and among types of edges in the different edges of (FP_1, EP_3), and (FP_1, EP_4). In particular, the similarity value between two patterns is obtained as the mean of three similarities obtained from the two different edges, two values obtained from the pairs of nodes and one obtained from the pair of edges: in the example, we have 0.45 for (FP_1, EP_3) and 0.4 for (FP_1, EP_4) (Figure 3a). Among the patterns for which the similarity and GR thresholds are exceeded, the chosen pattern is that which shows the highest growth rate value (lines 13, 20). In Figure 3a, EP_3 and EP_4 exceed the thresholds, but EP_3 is preferred for its highest value of similarity with FP_1 (0.45). The pattern EP_3 is stored (lines 14-16) and considered for subsequent join operations. In the subsequent iteration,



(a)



(b)

Fig. 3. Incremental join of EPs: an example.

the algorithm processes the period τ_3. In this iteration, EP_5 and EP_6 differ from EP_3 for only one edge, more precisely, the edge at the circle G (EP_5) and the node and edge at the circles H, I (EP_6). The similarity values are 0.9 (EP_5) and 0.53 (EP_6) and both exceed the threshold σ.

The exploration of the last period ($\tau_3, m = 3$) completes the incremental joins with the chains created until to τ_2. Indeed, we consider all EPs returned by *select_by_similarity* which meet the condition of minimum growth rate (lines 10 and 20): each of these EPs will be evaluated to complete the chains created with the EPs previously selected, namely of FP_1 and EP_3. Once the last period (τ_m) is reached, the *backtracking* mechanism is performed (backward mode): the control returns to Algorithm 1 where the last stored EP (EP_3 in τ_2 in Figure 3b) is removed (lines 23-24, Algorithm 1) and the EPs of the period τ_1 are explored again. This means that we consider the possibility to join FP_1 with the EPs in τ_2 without evaluating the EPs already included in the chains previously created, namely those *marked* (line 15 in Algorithm 2, line 19 in Algorithm 1). So, the algorithm $FWjoin$ is executed again in order to evaluate the join between FP_1 and EP_4, and then, complete the chain with EP_5 and the chain with EP_6.

4 Experiments

In order to prove the viability of the proposed approach, we performed experiments on real world and synthetic datasets. The first one is a social political dynamic network derived from the news reports concerning the relationships among nations and world-wide organizations: social and political relationships correspond to the types of edges of the network while nations and world-wide organizations are the nodes. The second one has been specifically built in order to test the computational properties of the approach. Periods are determined according to an equal-width discretization technique which partitions the observations $D : \langle D_1, D_2 \dots D_i \dots D_n \rangle$ in a sequence $\langle \tau_1, \tau_2, \dots, \tau_m \rangle$ of consecutive time-periods with identical width.

4.1 Real-world Dynamic Network

The network is collected under the study KEDS (Kansas Event Data System)[3]. In this dataset, our approach aims at building an explanatory model able to identify particular connections established among nations over time as well as track the change of social and political relations.

Dataset Description. The dataset includes 123,821 edges collected between April 1979 and December 2009 (D) and the time-points are in the format year/month/day. The number of nodes $\mathcal{N}$ is 228 while the types of edges are 20, i.e. 20 names of social and political relations reported in natural language. In this domain, understanding the evolution of the network in terms of type of edges (social and political relations) can be more interesting than considering the evolution on the nodes (nations). Coherently, we fix $\mathcal{S}_{\mathcal{N}}$ to return the middle of the range of the similarity, namely 0.5, while the measure $\mathcal{S}_{\mathcal{E}}$ is defined as the semantic similarity on the types of edges. In particular, we exploited the "Measures of Semantic Relatedness tools"(http://cwl-projects.cogsci.rpi.edu/msr/).

Experimental Setup. Experiments are performed to test the influence of the input parameters on the final evolution chains. Moreover, we define a quantitative measure in order to conduct an objective evaluation of the discovered chains. Such a measure estimates the *rarity* of the information expressed in each chain. More formally, let L: $EP_1, EP_2, \dots, EP_q$ be a chain discovered in the periods $\tau_1, \tau_2, \dots, \tau_m$ and let $[i_1, s_1), [i_2, s_2), \dots, [i_k, s_k]$ be a pre-defined equal-width discretization[4] on the values of the growth-rate, the *rarity* is computed as:

$$rarity_{GR}(L) = 1 - \frac{1}{q}\left[\sum_{j=1,2,\dots q} rarity_{GR}(EP_j)\right] \tag{1}$$

$$rarity_{GR}(EP_j) = \frac{\#EPs^{(i,s)}}{\#EPs^{\tau_j}} \tag{2}$$

where $\#EPs^{(i,s)}$ is the number of EPs whose growth-rate is included in the same bin and $\#EPs^{\tau_j}$ is the number of EPs generated in the period when EP_j

[3] http://web.ku.edu/keds/data.html

[4] Note that GR cannot be equal to infinity, since constructed from frequent patterns.

is generated. Therefore, the *rarity* of a chain ranges in $[0, 1]$ where the higher values the rarer the chain is. The same measure can be also defined for the similarity. Intuitively, the *rarity* is high for evolution chains whose EPs have less concurrent EPs in the same GR bin. In this case, the considered chain represents evolutions possibly not caught by other chains. According to its definition, the higher the *rarity*, the better the chain.

Results. Results are collected by varying the minimum thresholds σ_N, σ_E and $minGR$ with two different width of time-periods δ_τ. The thresholds σ_N, σ_E are set to the same value of σ. The first node X is set as "usa" (United States of America) and $minSupp$=1.5%. Results are shown in Table 1 and Table 2 where we report the number of discovered chains, average length of the chains, average number of FPs and EPs generated in the periods involved in the chains and *rarity*. Each row in Table 1 presents the values averaged on $minGR$=64, 8, 4, 2, while the values of the rows in Table 2 are averaged on σ=0.4, 0.25, 0.15, 0.1.

A first consideration can be drawn from the number and length of the chains in Table 1: decreasing the minimum similarity leads to have a greater set of chains with higher length. Indeed, low values of σ lead to select EPs with low similarities in the join operations (besides those with high similarities) with the result of *i)* avoiding skipping and *ii)* continuing to apply the join operation for the chains currently processed. The same motivation applies also to the sets of FPs and EPs: the number of FPs and EPs tends to grow because we have to consider EPs with low similarities, due to the decrease of σ.

As expected, the threshold σ influences $rarity_{GR}$: the higher the value of similarity threshold the higher the average rarity. This allows us to point out a peculiarity of the approach: patterns of edges, which are dissimilar each other, can participate to a chain, but the resulting chains have relative low uniqueness in terms of frequency ($rarity_{GR}$), so chains which relate two nodes belonging to different time-periods with dissimilar intermediate edges can be very rare.

The different settings of δ_τ identify two different widths of the periods $\tau_1, \tau_{h+1}, \ldots, \tau_m$: when δ_τ is 240 we have a smaller set of periods which explains smaller *avg length* and higher number of chains. By considering results in Table 1, we notice that $\delta_\tau = 120$ leads to better values of $rarity_{GR}$. Indeed, with a larger duration of a time-period (δ_τ=240) we collect a greater set of edges, namely observations of the network, which can lead to the generation of new patterns, which, in their turn, motivate the lower values of $rarity_{GR}$ with respect to δ_τ=120.

In Table 2 we can observe the correlation between the threshold $minGR$ and the $rarity_{similarity}$ and the influence of minGR on the final chains. Indeed, low values of growth-rate (obtained by decreasing $minGR$) lead to consider a larger set of EPs (for the join operation) which can increase the probability that an higher number of EPs can fall into the bins of the discretization of $rarity_{similarity}$, with the final result of generating less rare chains. It is noteworthy that, in this case, δ_τ does not influence $rarity_{similarity}$. This means that chains are more uniformly distributed in terms of the similarity of involved EPs.

In the following we report the (unique) evolution chain obtained with σ=0.4, $minGR$=64, X = "usa".

Table 1. Results with different values of σ and δ_τ.

δ_τ	σ	# chains	avg length	avg FPs	avg EPs	rarity$_{GR}$
120	0,4	1,75	5,00	168,06	166,66	0,2875
	0,25	1,75	5,00	170,35	168,88	0,2875
	0,15	25,75	12,16	186,53	185,59	0,09028
	0,1	52,31	12,29	179,82	179,82	0,07775
240	0,4	23,75	2,96	380,73	351,70	0,09475
	0,25	24,25	3,45	977,71	945,55	0,08875
	0,15	30,5	4,10	937,74	905,88	0,06180
	0,1	32	4,39	931,98	900,31	0,06750

Table 2. Results with different values of $minGR$ and δ_τ.

δ_τ	minGR	# chains	avg length	avg FPs	avg EPs	rarity$_{similarity}$
120	64	4,25	6,325	172,87	172,2075	0,87387
	8	16,75	9,09	177,63	176,58	0,4875
	4	14,75	9,66	177,67575	176,63	0,6204
	2	45,8125	9,3725	176,5825	175,53	0,846
240	64	24,25	3,7725	813,75	780,825	0,92625
	8	23,75	3,995	704,9425	673,7375	0,91275
	4	23,5	3,965	712,09675	681,5525	0,87325
	2	39	3,165	997,365	967,325	0,74833

$\{(\mathbf{usa}, isr, consult), (igo, pse, consult)\}^{(1979-12-13/1980-04-11)}$,
$\{(usa, isr, consult), (syr, usa, consult)\}^{(1981-04-10/1981-08-08)}$,
$\{(usa, isr, consult), (isr, usa, appeal)\}^{(1981-08-09/1981-12-07)}$,
$\{(usa, isr, consult), (isr, usa, consult)\}^{(1984-08-02/1984-11-30)}$,
$\{(\mathbf{igo}, isr, appeal), (isr, usa, consult)\}^{(1998-07-02/1998-10-30)}$

It describes the chain developed from the period $1981-04-10/1981-08-08$ to the period $1998-07-02/1998-10-30$ and has "usa" as first node and "igo" (Intergovernmental organizations) as last node. It depicts the evolution on the edges of the network which involve also the nodes "pse", "syr", "isr" (Palestinian Occupied Territories, Syria, Israel). This chain has $rarity_{GR}$=0.22 and $rarity_{similarity}$=0.706.

With σ=0.1 and $minGR$=64 we obtain the following evolution chain:
$\{(usa, lbn, \mathbf{consult}), (lbn, usa, \mathbf{consult})\}^{(1979-12-13\ 1980-04-11)}$
$\{(usa, lbn, \mathbf{express\_intent\_to\_cooperate}), (lbn, usa, consult)\}^{(1983-04-06\ 1983-08-04)}$
$\{(usa, lbn, \mathbf{express\_intent\_to\_cooperate}), (lbn, usa, fight)\}^{(1983-08-05\ 1983-12-03)}$
$\{(usa, lbn, \mathbf{fight}), (lbn, usa, \mathbf{fight})\}^{(1983-12-04\ 1984-04-02)}$

In this chain ($rarity_{GR}$=0.14, $rarity_{similarity}$=0.4), the relation between the nodes "usa" and "lbn" (Lebanon) changes from "consult"to "fight" through "express_intent_to_cooperate". Note that in the time-period 1983-08-05 / 1983-12-03 there is an asymmetric relationship among the nodes "usa" and "lbn".

4.2 Synthetic Dynamic Network

Synthetic datasets are generated by varying δ_τ, cardinality of $\mathcal{E}$ and $\mathcal{N}$ as well as the number of edges per time-point. In Table 3, we report a summary of the

Table 3. Artificial dataset characteristics

δ_τ	$\lvert\mathcal{N}\rvert$	$\lvert\mathcal{E}\rvert$	#edge types
500	(5,10,15,20)	(5,10,15,20)	(5,10,15,20)
1000	(5,10,15,20)	(5,10,15,20)	(5,10,15,20)
2000	(5,10,15,20)	(5,10,15,20)	(5,10,15,20)
3000	(5,10,15,20)	(5,10,15,20)	(5,10,15,20)

Table 4. Experiments on synthetic datasets.

δ_τ		$(\#\ nodes, \#\ edges, \#edge\ types)$			
		(5,5,5)	**(10,10,10)**	**(15,15,15)**	**(20,20,20)**
500	**avg length**	2	2.14	2.02	2
	time(mins)	68.1	147.8	162.9	97.67
	# chains	2000	7891	6937	19
1000	**avg length**	2	2.47	2.11	–
	time(mins)	50,4	113.1	27.96	20.3
	# chains	1360	48894	165	0
2000	**avg length**	2	2.15	2.24	–
	time(mins)	66.4	117.6	60.7	38.4
	# chains	2000	3001	1074	0
3000	**avg length**	2	2.23	2.18	–
	time(mins)	80	145.2	71.7	41.8
	# chains	1960	3225	3082	0

artificial datasets: for instance, when δ_τ=500, we have four datasets where the cardinalities of $\mathcal{N}$, $\mathcal{E}$ and the number of types of edge per time-point are equal to (5,5,5), (10,10,10), (15,15,15), (20,20,20) for the first, second, third and fourth dataset, respectively. The edges of a period τ_i are generated independently from those of other periods in order to evaluate our approach in the (worst) case in which (possible) chains are randomly generated. This choice avoids possible biases introduced by the criterion used in the generation of chains, but, on the other hand, only allows a quantitative evaluation of generated chains and not a qualitative evaluation. Coherently with this choice, we set the threshold $minGR$ to 1.0. The values of similarities among the nodes and among the types of edges are identical and set to 0.1 (σ=0.1).

A first observation we can draw (see Table 4) is that the computational cost of the approach is related to the produced results, namely the number of discovered chains and their length: the time performances grows up when *# chains* and *avg length* increase, especially in the settings (10,10,10), (15,15,15). Indeed, when the network becomes more complex (e.g. (20,20,20)), the running times are shorter due to the small number of discovered chains. This behavior can be motivated by the fact that the increase of the size of $\mathcal{N}$ and $\mathcal{E}$ does not imply an increase in the frequency of the patterns and, subsequently, of the emerging patterns and chains, with shorter running time for their (incremental) evaluation.

5 Conclusions

In this paper we investigated the task of discovering evolution chains in dynamic networks. The proposed solution is based on the extraction of emerging pattens

which are subsequently joined in order to generate evolution chains expressed as time-period stamped patterns. Experiments prove the applicability of the approach in real-world challenges. In particular, obtained results qualitatively prove the soundness and the usefulness of extracted chains in capturing changes in the "core" of a social and political network. Moreover, experiments on artificially generated data show that the algorithm well scales on large datasets, depending on the data distribution. For future work, we plan two directions: *i)* automatically determination of time-period widths on the basis of the underlying distribution of the data, *ii)* discovering chains in streaming environments where the networks typically exhibit gradual and sudden concept drift.

Acknowledgments This work is in partial fulfillment of the PRIN 2009 Project "Learning Techniques in Relational Domains and Their Applications" funded by the Italian Ministry of University and Research (MIUR).

References

1. R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In J. B. Bocca, M. Jarke, and C. Zaniolo, editors, *VLDB*, pages 487–499. Morgan Kaufmann, 1994.
2. M. Berlingerio, F. Bonchi, B. Bringmann, and A. Gionis. Mining graph evolution rules. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases: Part I*, ECML PKDD '09, pages 115–130, Berlin, Heidelberg, 2009. Springer-Verlag.
3. M. Ceci, A. Appice, C. Loglisci, C. Caruso, F. Fumarola, and D. Malerba. Novelty detection from evolving complex data streams with time windows. In *ISMIS 09: Proceedings of the 18th International Symposium on Foundations of Intelligent Systems*, volume 5722 of *Lecture Notes in Computer Science*, pages 563–572, 2009.
4. M. Ceci, A. Appice, and D. Malerba. Discovering emerging patterns in spatial databases: A multi-relational approach. In *PKDD'07, 11th European Conference on Principles and Practice of Knowledge Discovery in Databases*, volume 4702 of *Lecture Notes in Computer Science*, pages 390–397, 2007.
5. N. Di Mauro and D. Malerba. Mining networked data. In N. Chawla, I. King, and A. Sperduti, editors, *Symposium on Computational Intelligence and Data Mining (IEEE-CIDM11)*, page xx. IEEE, 2011.
6. G. Dong and J. Li. Efficient mining of emerging patterns: Discovering trends and differences. In *KDD*, pages 43–52, 1999.
7. D. Malerba. A relational perspective on spatial data mining. *IJDMMM*, 1(1):103–118, 2008.
8. J. Sun, C. Faloutsos, S. Papadimitriou, and P. S. Yu. Graphscope: parameter-free mining of large time-evolving graphs. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '07, pages 687–696, New York, NY, USA, 2007. ACM.
9. G. Widmer and M. Kubat. Learning in the presence of concept drift and hidden contexts. *Machine Learning*, 23(1):69–101, 1996.
10. J. Zhu, Q. Xie, and E. J. Chin. A hybrid time-series link prediction framework for large social network. In S. W. Liddle, K.-D. Schewe, A. M. Tjoa, and X. Zhou, editors, *DEXA (2)*, volume 7447 of *Lecture Notes in Computer Science*, pages 345–359. Springer, 2012.