Discovering Evolution Chains for Link Mining 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

Abstract. Link discovery is a task of Data Mining aiming at discovering hidden and/or indirect relationships among objects. The task seems to be attractive when applied to network structures due to the characteristic of the networks to model complex and heterogeneous relationships among objects. Even more challenging is the time-varying nature exhibited by several real world networks which anyway seems to not be taken into account in the existing works, where nodes and relationships among nodes are considered unchangeable. In this paper, we consider dynamic networks in order to determine links which can relate nodes across time and which therefore can have been developed during the evolution of the network. The approach is grounded on the main topological changes exhibited by the network over time and proceeds in two steps: identification of the main changes with a technique to discover emerging patterns and determination of the links with a technique to join emerging patterns. Experiments on real data prove the viability of the proposal.

1 Introduction

Link mining (or discovery) is a Data Mining task which enables the discovery of knowledge on the existence of relationships among objects [4]. Link mining is motivated by the spread of data sources storing interrelated objects which are naturally represented in the form of networks. Networks have become ubiquitous in several social, economical and scientific fields ranging from the Internet to social sciences, biology, epidemiology, geography and many others.

In the link mining problem, relationships can be either described or predicted, for this reason, link mining comprises either descriptive or predictive data mining approaches. Solutions to this task have been designed mainly through the integration of graph-based techniques which boil down the problem of link mining into the problem of describing/determining edges or sequences of edges (paths) which link nodes of interest. For instance, in [5], the problem is explored in the specific fields of genomics and proteomics and the purpose is to generate significant hypotheses of biological relationships from biological static networks.

However, these approaches traditionally assume that the networks are static and unchangeable, namely the structures and properties of a network do not vary over time. This assumption seems to be too restrictive in real world scenarios where networks can be actually dynamic and exhibit changes especially when

modeling real-world phenomena which evolve over time. For example, in social networks user profiles may appear and disappear or may be involved over time in different relations with other users. So, nodes and edges of the networks may appear and disappear over time and also assume different values or labels.

Most of algorithms proposed in the literature for the analysis of dynamic networks converges basically on three research lines: detection of communities over time, characterization of the evolution of the networks, and prediction of nodes/edges of the networks. Sun et al. [6] propose a technique to discover communities and detect changes in dynamic graphs that is represented as matrix and encoding schemes. The evolution of time-varying graphs is investigated in [1] with a frequent graph-based pattern approach. The representation of the time-evolving graphs as a sequence of cumulative graphs enables the discovery of rules able to characterize the evolution of the network considering the topological changes. In the problem of prediction, techniques for the determination of links are quite recent. For instance, in [7] the authors present a probabilistic method able to rank nodes of the network on the basis of their probability to be involved in future links. However, the solutions of link prediction address the problem of discovering information on the links only at a specific time point of its evolution of the network (typically the time point next to the last observed).

In this paper, we tackle a different task whose goal is to identify the links which can be created during the evolution of the network over time. This task is purely descriptive and allows the user to discover *evolution chains*, that is, sequences of patterns which slightly change from one time period to another time period. Patterns represent frequent topological regularities of the network.

The task studied in this work is different from studies reported in the literature since it origins from three reasonable assumptions. The first assumption is that analyzing the network at the level of single time-points can turn out to be not very significant and not very interesting with respect to the overall dynamics of the network, while the most interesting aspects to be analyzed should be identified at the level of periods of time where significant changes are expected and statistically more significant conclusions can be drawn. The second assumption lies in the possibility that nodes and edges of the network, observed in a time-period, can be somehow related to nodes and edges observed in previous time-periods. The third assumption concerns the possibility that the network exhibits the alternation between conservative periods, where we can observe mild changes, and periods with strong variability, where changes can be sudden.

Methodologically, in our approach, links are discovered through a process which explores the network over consecutive time-periods. For each time-period, it extracts frequent patterns which express regularities topological of the network. Indeed, not all the frequent patterns are taken, but only those which are considered to be "emerging", that is, patterns whose frequency significantly changes between the considered time-period and the previous one. As a time-period is analyzed, the extracted patterns from that period are joined incrementally with the sequence of patterns generated in the previous time-periods in order to discover *evolution chains*. Each evolution chain is a sequence of pat-

terns, where each pattern is associated with a time-period and the "similarity" between a pattern and the pattern associated with the previous time period is maximized. This is addressed to characterize the evolution of the networks and mine links which connect nodes of the network during its evolution.

The paper is organized as follows. In the section 2 we define formally the problem of link mining. The proposed computational solution is reported in the section 3. Experiments on real data and evaluation of the discovered evolution chains are reported in 4 Finally, conclusions in the section 5 close the work.

2 Problem Formulation

Before formally defining the link mining problem we intend to solve, some definitions are necessary. Let $D : \{D_1, D_2 \dots D_i \dots D_n\}$ be a sequence of time-ordered observations of the network. At each time-point t_i, the network is described by means of the pair D_i=(N_i,E_i) which corresponds to the sets of the nodes and edges of the network, where $N_i \subseteq \mathcal{N}$, $E_i \subseteq \mathcal{E}$, and $\mathcal{N}$, $\mathcal{E}$ are the sets of the nodes and the edges observed in $\{t_1 \dots t_n\}$, respectively. A triple (n_1, n_2, e) denotes that the edge e connects two nodes n_1, n_2 ($n_1, n_2 \in \mathcal{N}$, $e \in \mathcal{E}$). We call *period* ρ in $\{t_1 \dots t_n\}$ a sequence of consecutive time-points $\{t_i \dots t_j\}$ ($t_1 \leq t_i$, $t_j \leq t_n$). Two periods ρ_h and ρ_{h+1} are consecutive if $\exists j$ such that ρ_h=$\{t_i \dots t_j\}$, ρ_{h+1}=$\{t_{j+1} \dots t_k\}$. Crucial to this work are the following definitions:

Definition 1. *Given a time period $\rho_h = \{t_i \dots t_j\}$, we define a* frequent pattern *P_{ρ_h} as a set of triples $P_{\rho_h} = \{(n_{p'}, n_{q'}, e_{p',q'}), \dots, (n_{r'}, n_{s'}, e_{r's'})\}$ such that all the triples in P_{ρ_h} occur together in $\{t_i, \dots, t_j\}$ with relative frequency greater than a user defined threshold minSupp.*

Definition 2. *Given a node $X \in \mathcal{N}$, an* evolution chain *L_X is a sequence of frequent patterns $\langle P_{\rho_h}, P_{\rho_{h+1}}, \dots, P_{\rho_{h+v}} \rangle$ where X belongs to some triple in P_{ρ_h} and $\rho_h, \rho_{h+1}, \dots, \rho_{h+v}$ are consecutive periods. Moreover, P_{ρ_i} differs from $P_{\rho_{i-1}}$ in at most one triple.*

Intuitively, the fact that P_{ρ_i} differs from $P_{\rho_{i-1}}$ in at most one triple guarantees that we are considering the evolution on the co-presence of triples of the network.

According to the notation introduced so far, the problem we intend to solve can be formalized as follows:

Given: the set of time-stamped observations of the network $D : \{D_1, D_2, \dots, D_n\}$; δ_ρ the width of the periods $\rho_1, \rho_{h+1}, \dots, \rho_m$; a node $X \in \mathcal{N}$, *Find:* the set of evolution chains $\mathcal{L}_X = \{L_X\}$

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

3 Link Mining

The proposed approach implements two algorithms which proceed as described in the following. The former extracts from the network statistically relevant

topological changes in the form of emerging patterns, while the latter incrementally joins the extracted patterns in order to form, through the periods, evolution chains. The identification of the periods is performed by means of an equal-width discretization technique which partitions the observations $D : \{D_1, D_2 \dots D_i \dots D_n\}$ in a set $\rho_1, \rho_{h+1}, \dots, \rho_m$ of consecutive periods of width equal to δ_ρ. The two algorithms are described in the following.

3.1 Emerging Patterns to Represent Dynamic Networks

Emerging patterns (EPs)[2] are a particular kind of frequent patterns (FPs) and are used to characterize a partition of the data with respect to other partitions. In our context, EPs are used to characterize the network in a time-period with the respect to the previous time-period. This allows us to detect topological changes in the network. More formally, the main property of EPs is that their support (relative frequency) significantly changes from one partition to another one. In our case, we consider the patterns whose support significantly changes from one time-period to the previous time period. The larger the difference on the support of a pattern, the more interesting the pattern. Changes in pattern frequency are estimated in terms of growth rate, which is a frequency ratio computed as the ratio $GR_{\rho_i}(P) = supp_{\rho_i}(P)/supp_{\rho_{i-1}}(P)$, where $supp_{\rho_i}(P)$ is the support of P in ρ_i and $supp_{\rho_{i-1}}(P)$ is the support of P in the previous time-period ρ_{i-1}.

Algorithmically, EPs are discovered by evaluating the FPs generated in the period ρ_i against those FPs generated in the previous period ρ_{i-1}. In our implementation, EPs are generated with an algorithmic solution which exploits the well-known FP-tree structure which compactly stores the data and guarantees computational performance during the generation and evaluation of patterns [3]. Here, the FP-tree is used to store the frequent patterns extracted in the period ρ_{i-1} and discover the EPs in ρ_i with respect to ρ_{i-1}. To discover EPs in ρ_i, each FP is searched among the FPs in ρ_{i-1} and, in the case it is found, the growth-rate is evaluated against a minimal threshold $minGR$: it is considered emerging if its support in ρ_i is greater than the support that it has in ρ_{i-1} of a factor equal to $minGR$. A concrete example is reported in the following.

Consider the set of FPs $P_1 : \langle (n_1, n_2, e_{12}), (n_3, n_4, e_{34}), (n_1, n_3, n_{13}), (n_2, n_5, e_{25}), (n_1, n_5, e_{15}) \rangle$, $P_2 : \langle (n_1, n_3, n_{13}), (n_3, n_4, e_{34}), (n_1, n_2, e_{12}), (n_3, n_5, e_{35}), (n_2, n_5, e_{25}) \rangle$, $P_3 : \langle (n_1, n_2, n_{12}), (n_3, n_5, e_{35}) \rangle$, from the period ρ_{i-1} . As indicated in [3], a table (Header table) containing the frequent triples (items) in descending order of support can be created from that set. The FP-tree is then built time-point by time-point in ρ_{i-1}, from the first time-point to the last one (Figure 1).

To identify the EPs, each FP from ρ_i is evaluated with respect to the FP-tree created with the FPs discovered in the previous period ρ_{i-1}. For instance, consider the pattern $P : \langle (n_1, n_3, e_{13}), (n_3, n_5, e_{35}), (n_2, n_5, e_{25}) \rangle$ discovered in ρ_i as pattern to be matched with FP-tree created with the patterns in ρ_{i-1}. We consider the triple of P which in the Header table has the lowest support, namely (n_2, n_5, e_{25}) (circle A in Figure 1b) and, with its pointer, we start the visit of the FP-tree (circle B). Each branch is explored in order to match the triples of $P : \langle (n_1, n_3, e_{13}), (n_3, n_5, e_{35}), (n_2, n_5, e_{25}) \rangle$ against the triples of the nodes of

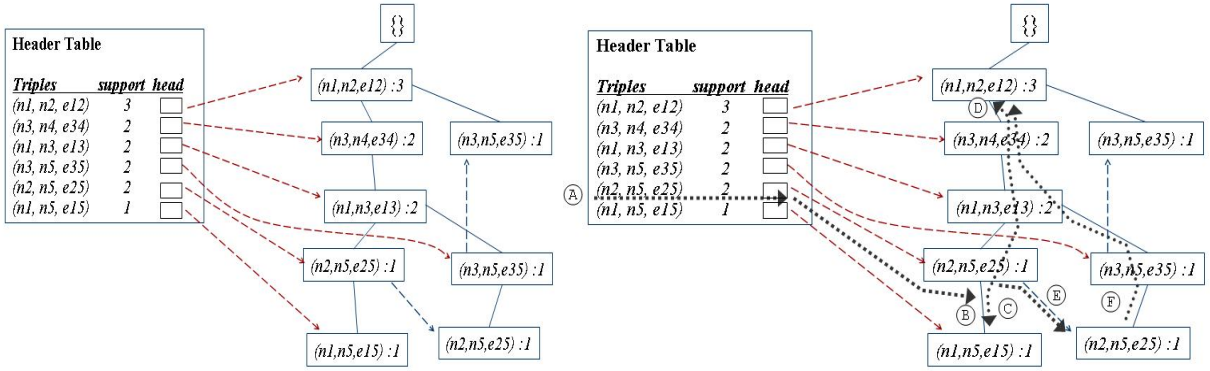

Fig. 1. (a) FPtrees construction. (b) Extraction of Emerging Patterns.

the branch. This matching operation is performed bottom-up (from circle B to circle D). In case the root cannot be reached, the matching operation fails and the pointer to the stored triple indicates the next branch to explore (circles E, F). Since the triples of the patterns are added in decreasing order, the support of P in ρ_{i-1} is obtained as the number of the occurrences associated to the last triple, namely the lower node of the FP-tree in the path that reaches the root.

Obviously, the process of extracting FP-tree in each time-period ρ_i is not repeated more than once and the FP-tree extracted at the time-period ρ_i is then used to evaluate EPs at the time-period ρ_{i+1}.

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 $\mathcal{S}_\mathcal{N} : \mathcal{N} \times \mathcal{N} \rightarrow \mathcal{R}$ be a function that returns a real value for each pair of nodes. Likewise for the edges, $\mathcal{S}_\mathcal{E} : \mathcal{E} \times \mathcal{E} \rightarrow \mathcal{R}$. These two functions can be considered as similarity measures between nodes and edges and are here considered as background knowledge for the investigated problem. Their availability is a quite reasonable assumption since in real-world networks we can easily define notions of similarity on nodes and edges. In particular, they can naturally model similarity among types of edges or similarity between types of nodes in heterogeneous networks (e.g., the similarity between the edges which, in the social networks, correspond to "friendship", "membership to the same group" can be 0.9).

Considering the notions introduced so far, the problem of mining links in a dynamic network can be so formulated:

Given: the set of sets of EPs $\mathcal{P}$ mined in $\rho_1, \ldots, \rho_m$; $\mathcal{S}_\mathcal{N}$, $\mathcal{S}_\mathcal{E}$ two similarity measures on $\mathcal{N}$ and $\mathcal{E}$ respectively; $\sigma_\mathcal{N}$ and $\sigma_\mathcal{E}$ two minimum threshold 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 approach proceeds by exploring the search spaces of EPs and incrementally joining the EPs in adjacent periods: since EPs denote particular occurrences of nodes and edges in a period, joining EPs has as result the generation of paths which express evolutions in the dynamic network.

The intuition behind the solution here proposed is that dynamic networks actually exhibit changes in some aspects while keep others unchanged, especially between adjacent periods. We use that intuition by considering as valid links those which connect both unchanged and changed aspects of the network. This means that, by keeping in mind that we consider EPs which represent changes in the frequencies, we are able to model evolutions in the data. The proposed solution joins EPs in adjacent periods (ρ_{i-1}, ρ_i) only if they have the same length (v triples) and differ for at most one triple: $v-1$ triples would represent the unchanged part of the network while the two different triples (one for each EP) would denote the changed part.

In case more than one candidates for joining are available, we exploit the notion of similarity for nodes and edges and, in particular, we join the candidate for which the new triple is "enough" similar to the removed one. Similarity is the average pairwise similarity between the nodes and the 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 topological aspects. 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, by starting from chains discovered at the previous time-period, explores backward the EPs of the previous periods in order to identify alternative chains, and *skipping*, which analyzes forward the EPs when a new time period is considered. In this case, the algorithm considers the possibility which EPs could be not joined with those extracted in adjacent periods.

This inability to join EPs in adjacent periods could be due to different factors: low similarity of nodes and edges, inadequateness of the EPs to the join operation and statistical parameters of EPs which not exceed the minimal thresholds.

The algorithmic description is illustrated in Algorithms 1, 2 of which we report an explanatory example in the following (Figure 2). Consider ρ_1, ρ_2, ρ_3 as periods, two similarity measures $\mathcal{S}_\mathcal{N}, \mathcal{S}_\mathcal{E}$ which produce the values of similarity reported in Figure 2a, the input node X as n_{11} and the thresholds $\sigma_\mathcal{N}$=$\sigma_\mathcal{E}$=σ=0.25. The first operation (lines 3-16, Algorithm 1) is the search of X in the first period as possible in which X occurs in the triples of EPs. The search starts from the EPs in ρ_1 and proceeds through the emerging patterns of the other periods until to found out n_{11}. In the example, X is found in the EPs of the period ρ_1 and among them (*candidates*) we select the EP with minimum length and maximum growth rate (lines 13-14): in Figure 2b, that is EP_1. This selection criteria is justified with fact that the monotonicity property of the support guarantees that the shorter length the more frequent pattern and better that pattern represents

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** $EP \in EP_h$ **do**
5: **if** $contains(X, EP)$ **then**
6: $candidates \leftarrow EP$
7: $found := true$
8: **end if**
9: **end for**
10: **if** $candidates == \oslash$ **then**
11: h++
12: **else**
13: $selected \leftarrow \underset{EP \in candidates}{\arg\min} \; length(EP)$
14: $selected \leftarrow \underset{EP \in selected}{\arg\max} \; Growth\_Rate(EP)$
15: **end if**
16: **end while**
17: $i := h + 1$
18: $push(\rho\_stack, i)$
19: $mark(selected)$
20: $push(EP\_stack, selected)$
21: **while** $EP\_stack <> \oslash$ **do**
22: $\mathcal{L}_X \leftarrow FWjoin(\rho\_stack, EP\_stack, \mathcal{P}, \mathcal{S}_\mathcal{N}, \mathcal{S}_\mathcal{E}, \sigma_\mathcal{N}, \sigma_\mathcal{E}, minGR, \mathcal{L}_X)$
23: $pop(EP\_stack)$
24: $pop(\rho\_stack)$
25: **end while**

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 (circles A,B in Figure 2b). When X is found out in the first period of the network ρ_1, as in the current example, we have no one EPs, so the EP is selected as FP by length and support. Information on the selected EPs are stored in two stack structures (lines 18,20) which will be used to explore the EPs over time in forward and backward direction and, finally, compose the final chains: in the example, EP_1 is stored in *EP_stack*. The search forward proceeds by joining the EP found out in the previous period with those of the next periods (line 2-3 Algorithm 2). The operation is performed first by selecting the EPs with the length equal to that of the *selected_EP* and then checking, for each pair (*selected*, *candidate*), which their triples are identical except only one (line 5-7) as previously described. In the case no one such EP is found out the process *skips* the current period ρ_i and continue the search to the next period (line 22). In the example, we have EP_3, EP_4 (discovered in ρ_2) which are identified as *selected_by_triples* (line 7) which can be joined with EP_1 (*selected_EP*) given that they differ for only one triple (respectively, circles C,D on the arrows toward EP_3 and circles E,F on the arrows toward EP_4). The emerging pattern, which will be joined with EP_1, it is selected from EP_3 and EP_4 by using the similarity measures and the growth rate values (lines 9,12). The measures $\mathcal{S}_\mathcal{N}, \mathcal{S}_\mathcal{E}$ are used to determine the similarity of the pairs of patterns EP_1, EP_3 and EP_1, EP_4 through the similarity of nodes and edges in the different triples between EP_1 and EP_3, and between EP_1, EP_4. The similarity value between two patterns is obtained as the mean of the similarity of the nodes and edges between the two

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

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

different triples: in the example, we have 0.45 for EP_1, EP_3 and 0.3 for EP_1, EP_4 (Figure 2a). Then, these values are considered and ordered if they exceed the minimal thresholds (line 9): in this example, both values exceed the value of the thresholds, which in this case is σ=0.25. The decision on the pattern to be selected is obtained by the growth rate value (line 12): in Figure 2a, EP_3 and EP_4 exceed the threshold min_{GR}=4 and have the same values whereas EP_3 is preferred for its higher value of similarity. The pattern EP_3 is stored (lines 13-15) and considered to perform next join operations given that $i = 2$, $m = 3$. The process continues at the line 11 where the examined period is ρ_3. Statements at the lines 5-15 are performed as previously illustrated. So, the EPs in ρ_3 are checked by length and similarity with EP_3 and possible joins are tested with EP_5 and EP_6 which differ from EP_3 for only one triple, namely the edge at the circle G (EP_5) and the node and edge at the circles H, I (EP_6). The similarity values of the triples are respectively 0.8(EP_5) and 0.3(EP_6) and both exceed the threshold σ. However, since ρ_3 is the last period to be examined ($i = m = 3$), all EPs in *select_by_similarity* which meet the condition of minimum growth rate are considered (lines 9,18): these EPs will used to complete the chain created with the join between the previously selected EPs, namely of EP_1 and EP_3. Once reached the last period (ρ_m) the *backtracking* mechanism is adopted: the process resumes from the Algorithm 1 where the last stored EP (EP_3 in ρ_2 in Figure 2b) is removed (line 23-24, Algorithm 1) and the EPs of the period ρ_1 are newly explored: there we consider the possibility to join EP_1 with the EPs in ρ_2 without evaluating the EPs already included in the previously created chains, namely those *marked* (line 14 in Algorithm 2, line 19 in Algorithm 1). In the

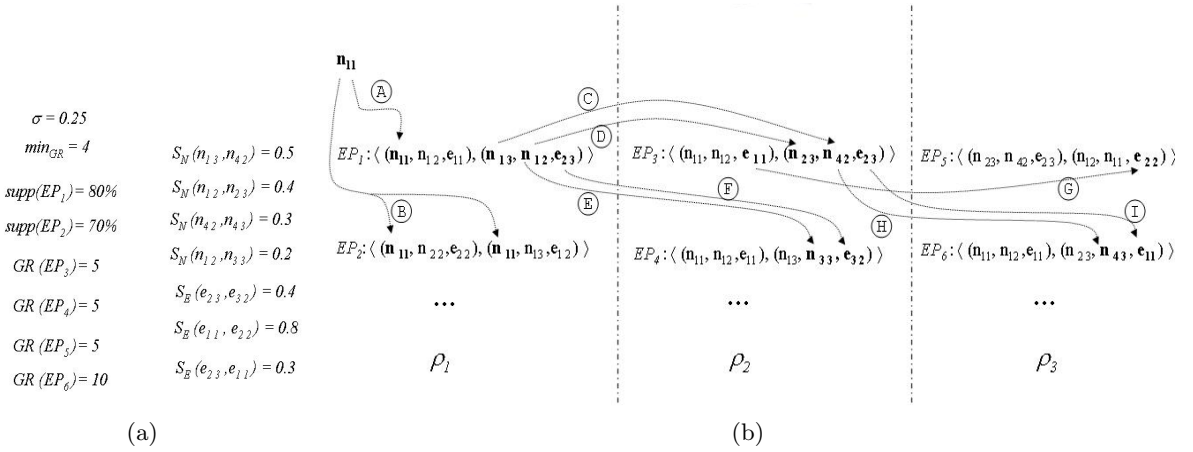

Fig. 2. Incremental join of EPs: an example.

example, possible joins are checked between EP_1 and EP_4, and then between EP_4 and EP_5, EP_6.

4 Experiments

In order to prove the viability of the proposed approach, we performed experiments on a real world dataset. It 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 edges of the network while nations and world-wide organizations are the nodes. The network is collected under the study KEDS (Kansas Event Data System)[1]. In this case the application of the approach aims at building an explanatory model able to identify particular connections which can have established among nations over time and track the change of the social and political relations among them.

Dataset Description. The dataset includes 123821 triples collected during April 1979 to December 2009 (D) and the time-points are in the format year/month/day. The set of nodes $\mathcal{N}$ amounts to 228 while the edges $\mathcal{E}$ are 20, more precisely 20 names of social and political relations reported in natural language. The similarity measure $\mathcal{S}_\mathcal{E}$ is obtained by pairs of relations as the semantic similarity applied to their names. The values of $\mathcal{S}_\mathcal{E}$ ranges in [0;1] and are computed by means of "Measures of Semantic Relatedness tools"[2]. The similarity measure $\mathcal{S}_\mathcal{N}$ is fixed to 0.5 for all pairs of nodes.

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

[1] http://web.ku.edu/keds/data.html
[2] http://cwl-projects.cogsci.rpi.edu/msr/

evolution chains. Such a measure estimates the *rarity* of the information expressed in each chain and it is intended as the uniqueness of the patterns used in the chain and the values of similarity and growth-rate associated to the patterns. Formally, *rarity* is defined as follows:

given the chain L: EP_1,EP_2,... EP_m discovered in the periods ρ_1,ρ_2,... ρ_m and a pre-defined discretization on the values of the growth-rate which produces the bins $[i_1;s_1)$,$[i_2;s_2)$,...,$[i_k;s_k)$, the *rarity* is determined as follows:

$$rarity_{GR}(L) = 1 - \sum_{EP_i \in EP_1, EP_2, \ldots EP_m} rarity_{GR}(EP_i) \quad (1)$$

$$rarity_{GR}(EP_i) = \frac{\#EPs^{[i;s)}}{\#EPs^{\rho_i}} \quad (2)$$

where $\#EPs^{[i;s)}$ is the number of EPs whose growth-rate is included in the same bin [i;s) where the growth-rate of EP_i of L is included, $\#EPs^{\rho_i}$ is the number of EPs generated in the period when EP_i of L is generated. Therefore, the *rarity* of a chain ranges in [0;1] where the higher values the rarer the chain is. Likewise for the similarity.

Results. Results are collected when varying the minimum thresholds $\sigma_N, \sigma_E, minGR$ with two different settings of δ_ρ. The thresholds σ_N, σ_E are set to the same value σ. The first node X is set as "usa" (United States of America) and $minSupp$=1.5. Results are illustrated in the tables 1, 2 where we have the number of discovered chains, average length of the chains as the average of the number of periods involved in the chains, average of the 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 done on the number and length of the chains in Table 1: decreasing the minimum similarity leads to have a greater set of chains and with higher length. Indeed, lower values of σ allow that EPs, with low similarities, are used for the join operation (besides to those with high similarities) with the result to i)avoid skipping of the remaining periods and ii) continue 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.

A correlation seems to exist between σ and $rarity_{GR}$: the higher the value of similarity threshold the more rare the chain. This allows us to point out a peculiarity of the approach to reveal a relevant aspect of the network: patterns of triples, which are dissimilar each other, can participate to create a chain with relative high uniqueness in terms of frequency ($rarity_{GR}$), so chains which connect two nodes with very dissimilar intermediate triples can be very rare.

The different settings of δ_ρ identify two different widths of the periods $\rho_1, \rho_{h+1}, \ldots, \rho_m$: when δ_ρ is 240 we have a smaller set of periods which explains the different orders of magnitude of *avg length*, whose tendency, anyway, remains unchanged when δ_ρ is 120. The wider duration of each period (240) explains also the different orders of magnitude of *#chains, avg FPs, avg EPs*: with a larger duration we collect a greater set of triples, namely observations of the network, which can

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

δ_ρ	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_ρ.

δ_ρ	minGR	# chains	avg length	avg FPs	avg EPs	rarity$_{similarity}$
120	64	45,8125	9,3725	176,5825	175,53	0,846
	8	14,75	9,66	177,67575	176,63	0,6204
	4	16,75	9,09	177,63	176,58	0,4875
	2	4,25	6,325	172,87	172,2075	0,87387
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

lead to the generation of new patterns, which, in their turn, motivate the lower orders of magnitude of $rarity_{GR}$ with respect to the case δ_ρ=120.

In Table 2 we can observe the correlation between $minGR$ and the rarity$_{similarity}$ and their influence on the final chains. Indeed, low values of growth-rate (obtained when decreasing $minGR$) lead to consider a greater 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 to generate less rare chains. As to the width of the period, we can confirm the observations done for Table 1: higher values of δ_ρ produce different orders of magnitude for *#chains, avg FPs, avg EPs*.

Examples of chains are reported below: when σ=0.4, $minGR$=64, X = "*usa*" only one chain is mined:

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

It reports the chain developed from the node "usa", observed at the period $1981-04-10/1981-08-08$, to the node *igo* (Intergovernmental organizations), observed at the period $1998-07-02/1998-10-30$. Contribution to this connection is provided by the triples of the network which involve the nodes *pse*, *isr* (Palestinian Occupied Territories,Israel). For instance the triple *(usa,isr,consult)* (relationship of "consult" between "usa" and "israel") contributes to the chains formed with "usa" and "igo" in the period $1981-08-09/1981-12-07$. This chains has rarity$_{GR}$=0.22, rarity$_{similarity}$=0.706.

With a lower σ (σ=0.1,$minGR$=64) we had a greater set of chains, among which

$\{(usa, lbn, \textbf{consult}), (lbn, usa, \textbf{consult})\}^{(1979-12-13\ 1980-04-11)}$ $\{(usa, lbn, \textbf{express\_intent\_to\_cooperate}),$
$(lbn, usa, consult)\}^{(1983-04-06\ 1983-08-04)}$ $\{(usa, lbn, \textbf{express\_intent\_to\_cooperate}), (lbn, usa, fight)\}^{(1983-08-05\ 1983-12-03)}$
$\{(usa, lbn, \textbf{fight}), (lbn, usa, \textbf{fight})\}^{(1983-12-04\ 1984-04-02)}$

It reports that the node "usa", observed in the triple $(usa, lbn, consult)$ and co-occurring to the triple *(lbn,usa,consult)* in (1979-12-13 / 1980-04-11)), establishes with "lbn" (Lebanon) a relationship which has been evolved across "consult", "express_intent_to_cooperate" and finally "fight" (rarity$_{GR}$=0.14, rarity$_{similarity}$=0.4).

5 Conclusions

We investigated the task of link mining in a perspective which seems to have not been considered yet, that of the dynamic networks. The approach allows to analyze the network as it changes over time and discover links which can have developed while the network changes. The proposed computational solution implements a descriptive data mining technique which enables the discovery of links on the basis of structural changes and frequent topological regularities. The experiments point out reasonably the applicability and usefulness of the approach to real-world challenges.

References

1. 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.
2. G. Dong and J. Li. Efficient mining of emerging patterns: Discovering trends and differences. In *KDD*, pages 43–52, 1999.
3. J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. In W. Chen, J. F. Naughton, and P. A. Bernstein, editors, *SIGMOD Conference*, pages 1–12. ACM, 2000.
4. J. H. Philip S. S. Yu and C. Faloutsos. *A Survey of Link Mining Tasks for Analyzing Noisy and Incomplete Networks*. Springer, 2010.
5. P. Sevon, L. Eronen, P. Hintsanen, K. Kulovesi, and H. Toivonen. Link discovery in graphs derived from biological databases. In U. Leser, F. Naumann, and B. A. Eckman, editors, *DILS*, volume 4075 of *Lecture Notes in Computer Science*, pages 35–49. Springer, 2006.
6. 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.
7. T. Tylenda, R. Angelova, and S. Bedathur. Towards time-aware link prediction in evolving social networks. In C. L. Giles, P. Mitra, I. Perisic, J. Yen, and H. Zhang, editors, *SNAKDD*, page 9. ACM, 2009.