Discovering Inter-Dimensional Rules in Dynamic Graphs

Kim-Ngan T. Nguyen[1], Loïc Cerf[1], Marc Plantevit[2], and Jean-François Boulicaut[1]

[1] Université de Lyon, CNRS, INRIA
INSA-Lyon, LIRIS Combining, UMR5205, F-69621, France
[2] Université de Lyon, CNRS, INRIA
Université Lyon 1, LIRIS Combining, UMR5205, F-69622, France

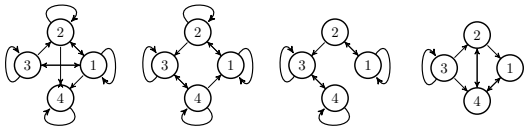
Abstract. Data mining methods that exploit graph/network have become quite popular and a timely challenge is to consider the discovery of dynamic properties in evolving graphs or networks. In this paper, we consider the dynamic oriented graphs that can be encoded as n-ary relations with $n \geq 3$ such that we have a least 3 dimensions: the dimensions of departure (tail) and arrival (head) vertices plus the time dimension. In other terms, it encodes the sequence of adjacency matrices of the graph. In such datasets, we propose a new semantics for inter-dimensional rules in dynamic graphs. We define rules that may involve subsets of any dimensions in their antecedents and their consequents and we propose the new objective interestingness measure called the exclusive confidence. We introduce a first algorithm for computing such inter-dimensional rules and we illustrate the added-value of exclusive confidence for supporting the discovery of relevant rules from a real-life dynamic graph.

1 Introduction

Graph mining is a popular topic. Many researchers have considered pattern discovery from large collections of graphs while others focus the analysis of one large graph or network. In the latter case, we observe two complementary directions of research. On one hand, global properties of graphs are studied (e. g., power-law distribution of node degrees or diameters). On the other hand, it is possible to use data mining algorithms to identify local patterns in the graphs (e. g., frequent subgraphs, clique patterns). Such local techniques can indeed benefit from the huge research effort on 0/1 data analysis, i. e., a graphs is seen as particular 0/1 table (the two involved domains being identical): its adjacency matrix.

In this paper, we investigate local pattern discovery from dynamic directed graphs, i. e., from of collection of static directed graphs that all share the same set of uniquely identified vertices. For instance, Fig. 1 depicts a dynamic directed graph involving four nodes. Four snapshots of this graph are available. The dynamic graph can be represented as the sequence of its adjacency matrices underneath. It describes the relationship between the tail vertices in $D^1 = \{d_1, d_2, d_3, d_4\}$ and the head vertices in $D^2 = \{a_1, a_2, a_3, a_4\}$ at the timestamps

in $D^3 = \{t_1, t_2, t_3, t_4\}$. Every '1', in the adjacency matrices is at the intersection of three elements $(d_i, a_j, t_k) \in D^1 \times D^2 \times D^3$, which indicate a directed edge from d_i to a_j at time t_k. Therefore at least three dimensions are necessary to encode a dynamic graph, which can be seen as a ternary relation (the one depicted in Fig. 1 is called $\mathcal{R}_E$). However, more dimensions may be used, for instance to encode information on edges and/or time aspects with different granularity.

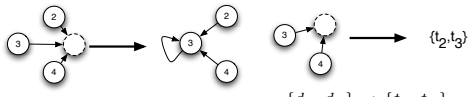


	a_1	a_2	a_3	a_4	a_1	a_2	a_3	a_4	a_1	a_2	a_3	a_4	a_1	a_2	a_3	a_4
d_1	1	1	1	1	1	1			1	1						1
d_2	1	1		1	1	1	1		1		1		1			1
d_3	1	1	1				1	1			1	1		1	1	1
d_4				1	1		1	1			1	1	1	1		
	t_1				t_2				t_3				t_4			

Fig. 1: The dynamic graph $\mathcal{R}_E \subseteq \{d_1, d_2, d_3, d_4\} \times \{a_1, a_2, a_3, a_4\} \times \{t_1, t_2, t_3, t_4\}$.

Studying descriptive rule mining from dynamic graphs is a rather new research topic and most of previous work impose severe restrictions on the form of the rules. The key contribution of this paper is the proposal of a quite general form of rules. These rules may involve any subset of dimensions in both the left-hand side and the right-hand side. In particular, the temporal dimensions can either explicitly appear in the rules or be used to measure the importance of the rules (i. e., the number of timestamps where the rule holds). Taking into account these different ways is complementary. It provides relevant patterns describing the evolution of a dynamic graph at a local level. Two examples of inter-dimensional rules that we want to extract are given in Fig. 2. Fig. 2a depicts a rule that is preserved at several timestamps. It intuitively means that if, at a time, the edges from vertices 2, 3 and 4 have the same heads then these heads are exclusively vertex 3. Rule in Fig. 2b means that if there are pairs of edges whose tails are nodes 3 and 4 and whose heads are the same vertex then it mainly occurs at times t_2 and t_3. To express the a priori relevancy of such rule, we use a straightforward extension of the classical frequency measure and an original extension of the confidence measure, the so-called *exclusive confidence*. The second contribution of this paper deals with the design of an algorithm that computes the a priori interesting rules. It exploits the principles (typically the enumeration strategy) of [7], i. e., the state-of-the-art algorithm for exploring the search space of multi-dimensional associations.

In Sect. 2, we provide the needed definitions to build the new pattern domain of inter-dimensional rules. Then, in Sect. 3, we define such rules and the exclusive



(a) (b)

Fig. 2: Example of rules.

confidence semantics. Sect. 4 introduces the first algorithm that computes a priori interesting rules from a dynamic graph. Sect. 5 deals with the empirical validation and various experiments on a real-life dynamic graph. Sect. 6 discusses related work and, finally, Sect. 7 briefly concludes.

2 Preliminary Definitions

Given n finite domains $\mathcal{D} = \{D^1, \ldots, D^n\}$ and an n-ary relation $\mathcal{R} \subseteq \times_{i=1..n} D^i$, the patterns of interest only involve some of the domains $\mathcal{D}' \subseteq \mathcal{D}$. E. g., the analyst may want to focus on subgraph patterns ($\mathcal{D}' = \{D_1, D_2\}$ in $\mathcal{R}_E$). She may, instead, want to discover pattern involving temporal dimensions. Without loss of generality, the dimensions are assumed ordered such that $\mathcal{D}' = \{D^1, \ldots, D^{|\mathcal{D}'|}\}$. We now formally define an *association* on $\mathcal{D}'$.

Definition 1 (Association). *$\forall \mathcal{D}' = \{D^1, \ldots, D^{|\mathcal{D}'|}\} \subseteq \mathcal{D}$, $\times_{i=1..|\mathcal{D}'|} X^i$ is an association on $\mathcal{D}'$ iff $\forall i = 1..|\mathcal{D}'|$, $X^i \neq \emptyset \wedge X^i \subseteq D^i$.*

$\times_{D^i \in \mathcal{D} \setminus \mathcal{D}'} D^i$ is called *support domain*. The support of an association generalizes that of an *itemset* in a binary relation ($n = 2$ and $|\mathcal{D}'| = 1$) [1]. Its formal definition uses the concatenation operator, denoted '$\cdot$'. E. g., $(d_2, a_3) \cdot (t_1) = (d_2, a_3, t_1)$.

Definition 2 (Support s). *$\forall \mathcal{D}' \subseteq \mathcal{D}$, let X an association on $\mathcal{D}'$. Its support, denoted $s(X)$, is $s(X) = \{u \in \times_{D^i \in \mathcal{D} \setminus \mathcal{D}'} D^i \mid \forall x \in X,\ x \cdot u \in \mathcal{R}\}$.*

The following definitions will ease the exposition of this paper.

Definition 3 (Projection π). *$\forall \mathcal{D}' = \{D^1, \ldots, D^{|\mathcal{D}'|}\} \subseteq \mathcal{D}$, let $X = X^1 \times \cdots \times X^{|\mathcal{D}'|}$ an association on $\mathcal{D}'$. $\forall D^i \in \mathcal{D}$, $\pi_{D^i}(X)$ is X^i if $D^i \in \mathcal{D}'$, $\emptyset$ otherwise.*

Definition 4 (Union $\sqcup$). *$\forall \mathcal{D}_X \subseteq \mathcal{D}$ and $\forall \mathcal{D}_Y \subseteq \mathcal{D}$, let X an association on $\mathcal{D}_X$ and Y an association on $\mathcal{D}_Y$. $X \sqcup Y$ is the association on $\mathcal{D}_X \cup \mathcal{D}_Y$ for which $\forall D^i \in \mathcal{D}$, $\pi_{D^i}(X \sqcup Y) = \pi_{D^i}(X) \cup \pi_{D^i}(Y)$.*

Definition 5 (Complement $\setminus$). *$\forall \mathcal{D}_X \subseteq \mathcal{D}$ and $\forall \mathcal{D}_Y \subseteq \mathcal{D}$, let X an association on $\mathcal{D}_X$ and Y an association on $\mathcal{D}_Y$. $Y \setminus X$ is the association on $\{D^i \in \mathcal{D}_Y \mid \pi_{D^i}(Y) \not\subseteq \pi_{D^i}(X)\}$ for which $\forall D^i \in \mathcal{D}$, $\pi_{D^i}(Y \setminus X) = \pi_{D^i}(Y) \setminus \pi_{D^i}(X)$.*

In $\mathcal{R}_E$, depicted in Fig. 1, $\{d_1, d_2\} \times \{a_1, a_2\}$ is an association on $\{D^1, D^2\}$, whereas $\{a_1, a_2\}$ is not because $\pi_{D^1}(\{a_1, a_2\}) = \emptyset$. It is an association on $\{D^2\}$. Their respective supports are $s(\{d_1, d_2\} \times \{a_1, a_2\}) = \{t_1, t_2\}$ and $s(\{a_1, a_2\}) = \{(d_1, t_1), (d_1, t_2), (d_1, t_3), (d_2, t_1), (d_2, t_2), (d_3, t_1), (d_4, t_4)\}$.

3 Inter-Dimensional Rules in Dynamic Graph

Given a dynamic graph, encoded as an n-ary relation $\mathcal{R}$ on $\mathcal{D}$, the analyst chooses the domains $\mathcal{D}_X \subseteq \mathcal{D}$ and $\mathcal{D}_Y \subseteq \mathcal{D}$ at, respectively, the left-hand side and the right-hand side of the rules to discover. E. g., to list rules involving tail vertices at their antecedents and timestamps at their consequents, $\mathcal{D}_X$ only contains one dimension of the relation (the tail vertices) and so does $\mathcal{D}_Y$ (the timestamps). Notice that $\mathcal{D}_X \cap \mathcal{D}_Y$ must be empty. An inter-dimensional rule on $(\mathcal{D}_X, \mathcal{D}_Y)$ is a couple of associations[3]. The first one on $\mathcal{D}_X$, the second one on $\mathcal{D}_Y$.

Definition 6 (Inter-dimensional rule). *$\forall \mathcal{D}_X \subseteq \mathcal{D}$, $\forall \mathcal{D}_Y \subseteq \mathcal{D}$, $X \to Y$ is an inter-dimensional rule on $(\mathcal{D}_X, \mathcal{D}_Y)$ iff X is an association on $\mathcal{D}_X$, Y is an association on $\mathcal{D}_Y$ and $\mathcal{D}_X \cap \mathcal{D}_Y = \emptyset$.*

In $\mathcal{R}_E$, $\{d_3\} \to \{a_3, a_4\}$ is an inter-dimensional rule on $(\{D^1\}, \{D^2\})$. The rule $\{d_3\} \times \{a_3, a_4\} \to \{d_4\}$ is not an inter-dimensional rule because elements in D^1 appear both at its left-hand side and at its right-hand side.

A rule is frequent if many "objects" verifies it. These objects are elements of a *support domain* for the rule, which is, in fact, $\times_{D^i \in \mathcal{D} \setminus (\mathcal{D}_X \cup \mathcal{D}_Y)} D^i$, i. e., that of the association (on $\mathcal{D}_X \cup \mathcal{D}_Y$) union of its antecedent and its consequent. The rule can be trusted, i. e., has a large enough confidence, if there is a high conditional probability to observe the consequent when the antecedent holds. In the context of inter-dimensional rules in dynamic graphs, a natural definition of the frequency exists. On the contrary, it is hard to define a confidence measure.

The (relative) frequency of an inter-dimensional rule in a dynamic graph is, in the support domain, the proportion of elements in the support of the union of its antecedent and its consequent.

Definition 7 (Frequency). *$\forall \mathcal{D}_X \subseteq \mathcal{D}$, $\forall \mathcal{D}_Y \subseteq \mathcal{D}$, the frequency of an inter-dimensional rule $X \to Y$ on $(\mathcal{D}_X, \mathcal{D}_Y)$ is $f(X \to Y) = \frac{|s(X \sqcup Y)|}{|\times_{D^i \in \mathcal{D} \setminus (\mathcal{D}_X \cup \mathcal{D}_Y)} D^i|}$.*

In $\mathcal{R}_E$, recursively applying Definitions 7, 4 and 2 gives $f(\{d_3\} \to \{a_3, a_4\}) = \frac{|s(\{d_3\} \times \{a_3, a_4\})|}{|D^3|} = \frac{|\{t_2, t_3, t_4\}|}{|\{t_1, t_2, t_3, t_4\}|} = \frac{3}{4}$.

Is it sensible to directly generalize the confidence measure of association rules in binary relations to n-ary relations? Doing so, the confidence of a rule $X \to Y$ would be $\frac{|s(X \sqcup Y)|}{|s(X)|}$. Unfortunately, this semantics is not satisfactory. Indeed, $s(X \sqcup Y)$ and $s(X)$ are disjoint sets and the ratio of their cardinalities does not make any sense. For instance, in $\mathcal{R}_E$, consider the rule $\{d_3\} \to \{a_3, a_4\}$. We have $s(\{d_3\} \times \{a_3, a_4\}) = \{t_2, t_3, t_4\}$ (i. e., a set of timestamps) while $s(\{d_3\}) = \{(a_1, t_1), (a_2, t_1), (a_3, t_1), (a_3, t_2), \dots\}$ (i. e., a set of couples (head vertices,timestamps)). However, it is possible to introduce a factor such that $|s(X)|$ and $|s(X \sqcup Y)|$ become comparable. The idea is to multiply $|s(X \sqcup Y)|$ by the cardinalities of its projections in the domains in $\mathcal{D}_Y$.

[3] The term "inter-dimensional association rule" often means, in the literature, a rule with *one* element per dimension. Our definition is more general.

Definition 8 (Confidence). *$\forall \mathcal{D}_X \subseteq \mathcal{D}$, $\forall \mathcal{D}_Y \subseteq \mathcal{D}$, the (exclusive) confidence of an inter-dimensional rule $X \to Y$ on $(\mathcal{D}_X, \mathcal{D}_Y)$ is $c(X \to Y) = \frac{|s(X \sqcup Y)| \times |\times_{D^i \in \mathcal{D}_Y} \pi_{D^i}(Y)|}{|s(X)|}$.*

Roughly speaking, the remedial factor, applied to $|s(X \sqcup Y)|$, allows to count the elements in $s(X \sqcup Y)$ "in the same way at the numerator and at the denominator of the fraction". For example, consider the rule $\{d_3\} \to \{a_3, a_4\}$ in $\mathcal{R}_E$, its exclusive confidence is $c(\{d_3\} \to \{a_3, a_4\}) = \frac{|s(\{d_3\} \times \{a_3, a_4\})| \times |\{a_3, a_4\}|}{|s(\{d_3\})|} = \frac{6}{10}$. Fig. 3 depicts, at every timestamp, the dynamic graph in Fig. 1 but it only keeps the ten edges with the vertex 3 as a tail. This number, "10", is found at the denominator of the fraction to compute the confidence. At its numerator, "6" actually is the count of those, among these 10 edges, that go to the vertices 3 and 4 at the same timestamp. They are thick in Fig. 3. At time t_1, there is an edge from d_3 to a_3 but there is no edge from d_3 to a_4 at this time. This "lowers" the confidence of the rule because a_4 is at its consequent too. At time t_4, there is an edge from d_3 to a_2. This also "lowers" the confidence in the fact that if d_3 is the tail of an edge then its head is either a_3 or a_4 (and not another vertex). That is why, this semantics of the confidence is said "exclusive". If $c(\{d_3\} \to \{a_3, a_4\})$ was 1, i. e., the maximal possible value, then, in every snapshot of the graph where the vertex 3 has a non-null output degree, it would *always* have two outgoing edges that would bind it with the vertex 3 and 4. *Any* other edge, with the vertex 3 as its tail, "lowers" the confidence.

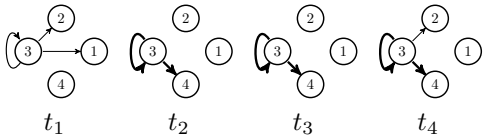


Fig. 3: Computing the confidence of $\{d_3\} \to \{a_3, a_4\}$.

Notice that the same speech applies to inter-dimensional rules involving the temporal dimension. E. g., Fig. 3 could illustrate, as is, the computation of $c(\{d_3\} \to \{t_2, t_3, t_4\})$, hence the same result $\frac{6}{10}$. This time however, the tick edges must be understood as those shared by the snapshots of the dynamic graph at t_2, t_3 and t_4 ("edgewise and" operation between the three graphs).

4 Computing Rules

Given an n-ary relation $\mathcal{R} \subseteq \times_{D^i \in \mathcal{D}} D^i$ and the parameters $(\mathcal{D}_X, \mathcal{D}_Y)$ (subsets of $\mathcal{D}$ and such that $\mathcal{D}_X \cap \mathcal{D}_Y = \emptyset$), $\mu \in [0, 1]$ and $\beta \in [0, 1]$, the *a priori* interesting inter-dimensional rules $X \to Y$ are such that (i) X is an association on $\mathcal{D}_X$, (ii) Y is an association on $\mathcal{D}_Y$, (iii) $f(X \to Y) \geq \mu$ and (iv) $c(X \to Y) \geq \beta$.

Our method, namely GEAR, first rewrites the relation by combining the components which are neither in $\mathcal{D}_X$ nor in $\mathcal{D}_Y$. In other terms, it builds the

support domain $D^{\text{supp}} = \times_{D^i \in \mathcal{D} \setminus (\mathcal{D}_X \cup \mathcal{D}_Y)} D^i$. The resulting relation, $\mathcal{R}_A$ is defined on the dimensions $\mathcal{D}_A = \mathcal{D}_X \cup \mathcal{D}_Y \cup \{D^{\text{support}}\}$. Then, GEAR extracts, in $\mathcal{R}_A$, every association A on $\mathcal{D}_X \cup \mathcal{D}_Y$ satisfying $\frac{|s(A)|}{|D^{\text{supp}}|} \geq \mu$. It entails that $\sqcup_{D^i \in \mathcal{D}_X} \pi_{D^i}(A) \rightarrow \sqcup_{D^i \in \mathcal{D}_Y} \pi_{D^i}(A)$ is a frequent inter-dimensional rule (and reciprocally, hence the completeness). Its exclusive confidence is finally computed. If it exceeds β, the rule is output.

The actual extraction of every frequent association A (associated with its support $A^{\text{supp}} \subseteq D^{\text{supp}}$), in $\mathcal{R}_A$, is now briefly detailed. A constraint-based approach is adopted, i. e., the problem is rewritten in terms of constraints and the patterns satisfying them all are the frequent associations. Here are the constraints:

- $\mathcal{C}_{\text{on-}(\mathcal{D}_X \cup \mathcal{D}_Y)}(A \sqcup A^{\text{supp}}) \equiv \forall D^i \in \mathcal{D}_X \cup \mathcal{D}_Y,\ \pi_{D^i}(A) \neq \emptyset$;
- $\mathcal{C}_{\text{connected}}(A \sqcup A^{\text{supp}}) \equiv A \sqcup A^{\text{supp}} \subseteq \mathcal{R}_A$;
- $\mathcal{C}_{\text{entire-supp}}(A \sqcup A^{\text{supp}}) \equiv A^{\text{supp}} = s(A)$;
- $\mathcal{C}_{\lceil \mu \times |D^{\text{supp}}| \rceil\text{-freq}}(A \sqcup A^{\text{supp}}) \equiv |A^{\text{supp}}| \geq \lceil \mu \times |D^{\text{supp}}| \rceil$.

Thanks to the last constraint, the frequency of the rule $\sqcup_{D^i \in \mathcal{D}_X} \pi_{D^i}(A) \rightarrow \sqcup_{D^i \in \mathcal{D}_Y} \pi_{D^i}(A)$ must reach or exceed μ. Indeed, $\frac{|s(A)|}{|D^{\text{supp}}|} \geq \mu$ is equivalent to $|s(A)| \geq \lceil \mu \times |D^{\text{supp}}| \rceil$ and, because the third constraint ($A^{\text{supp}} = s(A)$) must be satisfied as well, it is equivalent to $|A^{\text{supp}}| \geq \lceil \mu \times |D^{\text{supp}}| \rceil$. The third constraint, $\mathcal{C}_{\text{entire-supp}}$, forces a "closed" support. Indeed, by definition of the support (Definition 2), adding an element to A^{supp} ($= s(A)$) necessarily violates $\mathcal{C}_{\text{connected}}$. Thus, $\mathcal{C}_{\text{entire-supp}}(A \sqcup A^{\text{supp}})$ is equivalent to $\forall t \in D^{\text{supp}} \setminus A^{\text{supp}},\ A \sqcup \{t\} \not\subseteq \mathcal{R}_A$.

The algorithm traverses the search space by recursively partitioning it into two complementary parts ("divide and conquer"). In this way, a binary tree represents the performed enumeration. At every node of this tree, two associations, namely U and V, are updated. U is the smallest association that may be discovered in the enumeration sub-tree rooted by the node, whereas $U \sqcup V$ is the largest. That is why GEAR is initially called with $U = \emptyset$ and $V = \times_{D^i \in \mathcal{D}_A} D^i$. At every non-terminal node, an element e is chosen in $\cup_{D^i \in \mathcal{D}_A} \pi_{D^i}(V)$. In the enumeration sub-tree that derives from the first child, e is present in every U association (i. e., e is "moved" from V to U). In the enumeration sub-tree that derives from the second child, e is absent from every U association (i. e., e is "removed" from V). There are two reasons for an enumeration node to be a leaf of the enumeration tree. The first reason is that at least one of the four constraints is guaranteed to be violated by every U association in the sub-tree that would derive from the node. It happens when:

- $\exists D^i \in \mathcal{D}_X \cup \mathcal{D}_Y$ such that $\pi_{D^i}(U \sqcup V) = \emptyset$ ($\mathcal{C}_{\text{on-}(\mathcal{D}_X \cup \mathcal{D}_Y)}$ is violated);
- $\forall D^i \in \mathcal{D}_A, \pi_{D^i}(U) \neq \emptyset \wedge U \not\subseteq \mathcal{R}_A$ ($\mathcal{C}_{\text{connected}}$ is violated);
- $\exists t \in D^{\text{supp}} \setminus \pi_{D^{\text{supp}}}(U \sqcup V)$ such that $\Big((U \sqcup V) \setminus \pi_{D^{\text{supp}}}(U \sqcup V)\Big) \sqcup \{t\} \subseteq \mathcal{R}_A$ ($\mathcal{C}_{\text{entire-supp}}$ is violated);
- $|\pi_{D^{\text{supp}}}(U \sqcup V)| < \lceil \mu \times |D^{\text{supp}}| \rceil$ ($\mathcal{C}_{\lceil \mu \times |D^{\text{supp}}| \rceil\text{-freq}}$ is violated).

The proofs of these pruning properties are based on generalizations of monotone or anti-monotone properties that the four constraints have. The constraint

$\mathcal{C}_{\text{connected}}$ is monotone, i. e., if an association X violates the constraints then every larger association violates it as well. Since U is the smallest association in the sub-tree, $\neg\mathcal{C}_{\text{connected}}(U)$ is a safe pruning criterion. Dually, the three other constraints are anti-monotone, i. e., if an association X violates one of them then every smaller association violates it as well. That is why, to potentially prune the sub-tree rooted by the current enumeration node, their variables are replaced by the largest association in it: $U \sqcup V$. The second reason for an enumeration node to be a leaf is the actual discovery of a frequent association U. It happens when there is no more element to enumerate, i. e., when $V = \emptyset$.

An improved enumeration strategy avoids the generation of the nodes that violate $\mathcal{C}_{\text{connected}}$. To do so, in every first child (where an element e is "moved" to U), every element in $\cup_{D^i \in \mathcal{D}_A} \pi_{D^i}(V)$ that would violate $\mathcal{C}_{\text{connected}}$ if added to $U \sqcup \{e\}$ is "removed" from V. Algorithm 1 sums up the extraction of every frequent inter-dimensional association rules with high enough confidences. Other performance improvements (e. g., pertaining to the enforcement of $\mathcal{C}_{\text{entire-supp}}$) were implemented. They actually are analog to what is done in [7] for the extraction of closed patterns in n-ary relations. The **Choose** function is that of [7] too. Another useful feature, inherited from [7], is the ability to additionally and efficiently enforce any piecewise (anti)-monotone constraint the associations must satisfy. In some of the following experiments, the constraint $\mathcal{C}_{(\alpha^1,\ldots,\alpha^{|\mathcal{D}_X \cup \mathcal{D}_Y|})\text{-min-sizes}}$ (where $(\alpha^1, \ldots, \alpha^{|\mathcal{D}_X \cup \mathcal{D}_Y|}) \in \mathbb{N}^{|\mathcal{D}_X \cup \mathcal{D}_Y|}$) will be used:

$$\mathcal{C}_{(\alpha^1,\ldots,\alpha^{|\mathcal{D}_X \cup \mathcal{D}_Y|})\text{-min-sizes}}(A \sqcup A^{\text{supp}}) \equiv \forall D^i \in \mathcal{D}_X \cup \mathcal{D}_Y, |\pi_{D^i}(A)| \geq \alpha^i \ .$$

Algorithm 1: Algorithm GEAR.

Input: (U, V)
Output: Every *a priori* interesting association rule involving every element in $\cup_{D^i \in \mathcal{D}_X \cup \mathcal{D}_Y} \pi_{D^i}(U)$ and possibly some elements in $\cup_{D^i \in \mathcal{D}_X \cup \mathcal{D}_Y} \pi_{D^i}(V)$

if $\mathcal{C}_{on\text{-}(\mathcal{D}_X \cup \mathcal{D}_Y)}(U \sqcup V) \wedge \mathcal{C}_{entire\text{-}supp}(U \sqcup V) \wedge \mathcal{C}_{\lceil \mu \times |D^{supp}| \rceil\text{-}freq}(U \sqcup V)$ **then**
  if $V = \emptyset$ **then**
    if $c(\sqcup_{D^i \in \mathcal{D}_X} \pi_{D^i}(U) \rightarrow \sqcup_{D^i \in \mathcal{D}_Y} \pi_{D^i}(U)) \geq \beta$ **then**
      Output $\sqcup_{D^i \in \mathcal{D}_X} \pi_{D^i}(U) \rightarrow \sqcup_{D^i \in \mathcal{D}_Y} \pi_{D^i}(U)$;
    else
      Choose $e \in \cup_{D^i \in \mathcal{D}_A} \pi_{D^i}(V)$;
      GEAR($U \sqcup \{e\}$,$(V \setminus \{e\}) \setminus (f \in \pi_{D^i}(V) \mid \neg C_{\text{connected}}(U \sqcup \{e\} \sqcup \{f\}))_{D^i \in \mathcal{D}_A}$);
      GEAR(U, $V \setminus \{e\}$);

5 Experimental Study

GEAR was implemented in C++ and compiled with GCC 4.2.4. This section reports experiments, which were performed on a GNU/Linux™ system equipped with an Intel® Pentium® 4 processor cadenced at 3 GHz and 1 GB of RAM.

`Vélo'v` is a bicycle rental service run by the urban community of Lyon, France. 327 `Vélo'v` stations are spread over Lyon and its surrounding area. At any of these stations, the users can take a bicycle and bring it to any other station. Whenever a bicycle is rented or returned, this event is logged. Logs represent more than 13.1 million rides along 30 months. This dataset is seen as a dynamic directed graph evolving into two temporal dimension: the 7 days of the week and the 24 one-hour periods in a day. A significant amount of bicycles (local test inspired by the computation of a p-value), that are rented at the (departure) station ds on day d (e. g., Monday) at hour h (e. g., from 1pm to 2pm) and returned at the (arrival) station as, translates to an edge from ds to as in the graph timestamped with (d, h). In other terms, the (ds, as, d, h) in the related relation $\mathcal{R}_{\texttt{Vélov'v}} \subseteq \texttt{Departure} \times \texttt{Arrival} \times \texttt{Day} \times \texttt{Hour}$. In the end, this contains $117,411$ 4-tuples, hence a $\frac{117,411}{7\times24\times327\times327} = 0.7\%$ density.

We analyze the results of the experiments with regard to the following questions: (a) Do the discovered graph rules make sense? (b) How to handle time in these rules? (c) What does the exclusive confidence definition capture?, and (d) How does GEAR behave with respect to the parameter settings?

We first searched for rules with time periods and departure stations (tail vertices) at their antecedents; day information at their consequents. In this way, stations that, at some time, "emit" bicycles towards many other stations, but exclusively for some days, are discovered. With the minimum thresholds $\mu = 0.08$ and $\beta = 0.6$, 35 rules are extracted. Fig. 4 reports three of them. The rule in Fig. 4a means that most of the departures from Station 6002 and between 11am and 12am occur on Sundays ($c = 0.71$). This makes sense: this station is at the main entrance of the most popular park, where people like to ride on Sundays. The rule in Fig. 4b means that there rarely are departures from Station 1002 between 1am and 3am except on Sundays ($c = 0.62$). This makes sense: this station is located in a district with many pubs and the favored evenings to party are on Saturdays. Furthermore the public transportation services stop at midnight and the `Vélo'v` is a good alternative to come back home. The rule in 4c describes another known behavior. Many people living outside Lyon arrive by train between 8am and 9am and use `Vélo'v` to finish their trips towards their working place. Indeed, Station 3001 is at the train station inside the main working district. This behavior is specific to the working days ($c = 0.66$).

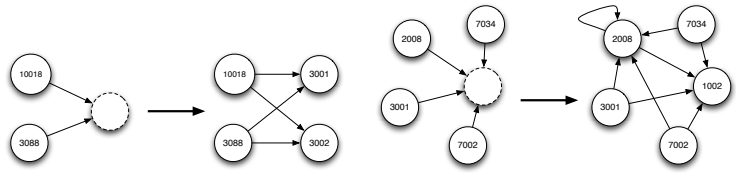
To answer the question "*which are the stations that often exchange bicycles?*", we searched for rules whose antecedents are departure stations (i. e., tail vertices) and consequents are arrival stations (i. e., head vertices). The support domain of these rules are the Cartesian product of the seven days and the 24 hours. The constraint $\mathcal{C}_{2,2\text{-min-sizes}}$ (see Sect. 4) is additionally enforced, i. e., every rule must involve at least two departure stations and two arrival stations. With $\mu = 0.03$ and $\beta = 0.8$, GEAR returns 27 rules. Fig. 5 reports some of them.

$\{6002\} \times \{11\text{-}12\text{am}\} \rightarrow \{\text{Sun.}\}$ $(s = 0.09,\ c = 0.71)$ (a)

$\{1002\} \times \{1\text{-}2\text{am}, 2\text{-}3\text{am}\} \rightarrow \{\text{Sun.}\}$ $(s = 0.09,\ c = 0.62)$ (b)

$\{3001\} \times \{8\text{-}9\text{am}\} \rightarrow \{\text{working days}\}$ $(s = 0.13,\ c = 0.66)$ (c)

Fig. 4: Example of rules of the form `Departures` × `Hours` → `Days`.



$\{10018, 3088\} \rightarrow \{3002, 3001\}$ $(s = 0.05,\ c = 0.86)$ (a)

$\{2008, 3001, 7002, 7034\} \rightarrow \{1002, 2008\}$ $(s = 0.04,\ c = 0.82)$ (b)

Fig. 5: Example of rules of the form `Departures` → `Arrivals`.

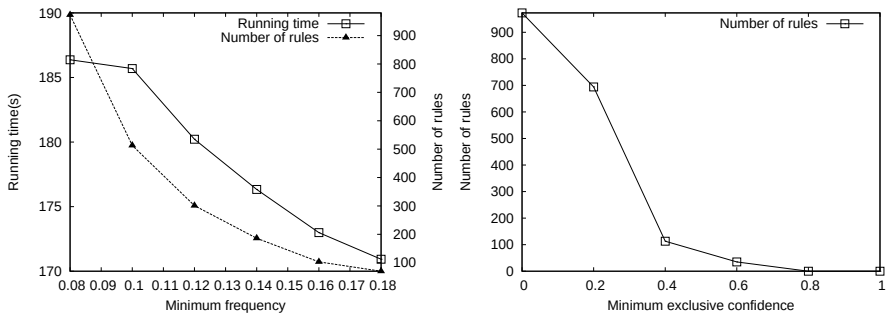
Do some stations exchange many bicycles at favored hours every day? To answer to this question, we search for rules whose antecedents consist of time periods and departure stations (i. e., tail vertices); their consequents are arrival stations (i. e., head vertices). To discover rules that hold every day, the minimal frequency threshold is set to 1. With $\beta = 0.8$, GEAR returns 40 rules which contain at least one time period, two departure stations and two arrival stations. These rules mean that there are some known time periods in which set of stations maintain some privileged bicycle exchanges. Some of them are given in Fig. 6. This kind of knowledge is valuable for the data owner. For instance, if there is no available bicycle at a `Vélo'v` station then other `Vélo'v` stations that maintain strong exchanges with it may be impacted as well.

$\{4\text{-}5\text{pm}, 5\text{-}6\text{pm}, 7\text{-}8\text{pm}\} \times \{1002, 2026\} \rightarrow \{1002, 2026\}$ $(s = 1,\ c = 0.82)$ (a)

$\{5\text{-}6\text{pm}\} \times \{3001, 10048\} \rightarrow \{3001, 10048\}$ $(s = 1,\ c = 1)$ (b)

Fig. 6: Example of rules of the form `Hours` × `Departures` → `Arrivals`.

We now report a performance study of GEAR discovering, in $\mathcal{R}_{\texttt{Vélo'v}}$, every frequent inter-dimensional rule of the form `Departures` $\times$ `Hours` $\rightarrow$ `Days`. When the minimal frequency threshold increases, the number of frequent associations and the running time decrease (Fig. 7a obtained with $\beta = 0$). Indeed, GEAR prunes large areas of the search space where every association violates the constraint $\mathcal{C}_{\lceil \mu \times |D^{\text{supp}}| \rceil\text{-freq}}$. When the minimum confidence threshold increases, the number of rules decreases too (Fig. 7b obtained with $\mu = 0.08$). GEAR's scalability was tested on the extraction of these rules (still with a frequency exceeding 0.08). To do so the nodes of the graphs were replicated, up to ten times, with their incoming edges only. It turns out that the algorithm scales linearly. More precisely a linear regression of $R \mapsto \frac{T_R}{T_1}$ (where R is how many times the arrival stations are replicated; T_R the running time on this replicated dataset) gives $y = 0.88x + 0.08$ with 0.05 as a standard error. Since $0.88 < 1$, it can be written that GEAR conforms to the proportions of the relation for faster extractions.



(a) Running time w.r.t. μ.

(b) Number of rules w.r.t. β.

Fig. 7: Effectiveness of GEAR.

6 Related Work

Mining graphs has recently received a lot of attention in the data mining community. Many different techniques (e. g., densification laws, shrinking diameter, factorization, clustering, evolution of communities, etc.) [2, 15, 9, 17, 13, 11, 3, 16, 5]. In this section, we focus on methods that mine local patterns. [6] extracts such patterns in labeled dynamic graphs. Frequent subgraph mining algorithms are adapted to time series of graphs. The approach aims at finding subgraphs that are topologically frequent and show an identical dynamic behavior over time, i. e., insertions and deletions of edges occur in the same order of time. Due to the complexity of the task, their algorithm is not complete. Computing the overlap-based support measure means solving a maximal independent set problem and this approach uses a greedy algorithm. [10] proposes a fast algorithm

to mine frequent transformation subsequences from a set of dynamic labeled graphs (the labels on vertices and edges can change over time). Starting with the hypothesis that the changes in a dynamic graph are gradual, they propose to succinctly represent the dynamics with a graph grammar: each change between two observed successive graph states is interpolated by axiomatic transformation rules. [18] studies how a graph is structurally transformed through time. The proposed method computes graph rewriting rules that describe the evolution of two consecutive graphs. These rules are then abstracted into patterns representing the dynamics of a sequence of graphs. [12] introduces the periodic subgraph mining problem, i. e., identifying every frequent closed periodic subgraph. They empirically demonstrate the efficiency and the interest of their proposal on several real-world dynamic social networks. By showing that dynamic graphs can be represented as ternary relations, [8] describes a constraint-based mining approach to discover maximal cliques that are preserved over almost-contiguous timestamps. The constraints are pushed into a closed n-ary pattern mining algorithm. [14] proposes a constraint-based approach too. It the evolution of dense and isolated subgraphs defined by two user-parameterized constraints. Associating a temporal event type with each pattern captures the temporal evolution of the identified subgraph, i. e., the formation, dissolution, growth, diminution and stability of subgraphs between two consecutive timestamps. The algorithm incrementally processes the time series of graphs. [4] introduces the problem of extracting graph evolution rules satisfying minimal support and confidence constraints. It finds isomorphic subgraphs that match the timestamps associated with each edge, and, if present, the properties of the vertices and edges of the dynamic graph. Graph evolution rules are then derived with two different confidence measures. This approach is the closest to ours: it aims at describing a time-evolving graph with rules. Nevertheless, this work focuses on the dynamic changes in the graph whereas we provide a generic framework to discover inter-dimensional rules where the time is either in the rule or in its support.

7 Conclusion

We tackled the problem of describing dynamic graphs via rules that can involve subsets of any dimension (including temporal dimensions) at its antecedent or consequent. We proposed a new semantics for inter-dimensional rules in dynamic graphs. It relies on a relevant objective interestingness measure called the exclusive confidence. We introduced and implemented Gear, an effective solution for computing such rules. Experiments on a real-world dynamic graph demonstrated the interest of our proposal. A timely challenge is to look for primitive constraints that can support more sophisticated knowledge discovery processes in dynamic graphs. Some of these constraints would deal with the temporal dimension(s) (e. g., time contiguity [8]). Other constraints would deal with the "form" of the patterns to discover (e. g., cliques, dense subgraphs, etc.). Another challenge is to revisit, in our setting, important techniques developed for classical association rules, for instance, non redundancy aspects (see, e. g., [19]).

Acknowledgements. This work was partly funded by the ANR project Bingo2 (MDCO 2007) and by a grant from the Vietnamese government.

References

1. Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., Verkamo, A.I.: Fast discovery of association rules. In: Advances in Knowledge Discovery and Data Mining, pp. 307–328. AAAI/MIT Press (1996)
2. Backstrom, L., Huttenlocher, D., Kleinberg, J., Lan, X.: Group formation in large social networks: membership, growth, and evolution. In: KDD. pp. 44–54. ACM Press (2006)
3. Berger-Wolf, T.Y., Saia, J.: A framework for analysis of dynamic social networks. In: KDD. pp. 523–528. ACM Press (2006)
4. Berlingerio, M., Bonchi, F., Bringmann, B., Gionis, A.: Mining graph evolution rules. In: ECML/PKDD. pp. 115–130. Springer (2009)
5. Berlingerio, M., Coscia, M., Giannotti, F., Monreale, A., Pedreschi, D.: As time goes by: Discovering eras in evolving social networks. In: PAKDD (1). pp. 81–90. Springer (2010)
6. Borgwardt, K.M., Kriegel, H.P., Wackersreuther, P.: Pattern mining in frequent dynamic subgraphs. In: ICDM. pp. 818–822. IEEE Computer Society (2006)
7. Cerf, L., Besson, J., Robardet, C., Boulicaut, J.F.: Closed patterns meet n-ary relations. ACM Trans. on Knowledge Discovery from Data 3(1), 1–36 (2009)
8. Cerf, L., Nguyen, T.B.N., Boulicaut, J.F.: Discovering relevant cross-graph cliques in dynamic networks. In: ISMIS. pp. 513–522. Springer (2009)
9. Chi, Y., Zhu, S., Song, X., Tatemura, J., Tseng, B.L.: Structural and temporal analysis of the blogosphere through community factorization. In: KDD. pp. 163–172. ACM Press (2007)
10. Inokuchi, A., Washio, T.: A fast method to mine frequent subsequences from graph sequence data. In: ICDM. pp. 303–312. IEEE Computer Society (2008)
11. Kumar, R., Novak, J., Tomkins, A.: Structure and evolution of online social networks. In: KDD. pp. 611–617. ACM Press (2006)
12. Lahiri, M., Berger-Wolf, T.Y.: Mining periodic behavior in dynamic social networks. In: ICDM. pp. 373–382. IEEE Computer Society (2008)
13. Leskovec, J., Kleinberg, J., Faloutsos, C.: Graphs over time: densification laws, shrinking diameters and possible explanations. In: KDD. pp. 177–187. ACM Press (2005)
14. Robardet, C.: Constraint-based pattern mining in dynamic graphs. In: ICDM. pp. 950–955. IEEE Computer Society (2009)
15. Sun, J., Papadimitriou, S., Yu, P.S., Faloutsos, C.: Graphscope: Parameter-free mining of large time-evolving graphs. In: KDD. pp. 687–696. ACM Press (2007)
16. Tantipathananandh, C., Berger-Wolf, T.Y., Kempe, D.: A framework for community identification in dynamic social networks. In: KDD. pp. 717–726. ACM Press (2007)
17. Tong, H., Papadimitriou, S., Sun, J., Yu, P.S., Faloutsos, C.: Colibri: fast mining of large static and dynamic graphs. In: KDD. pp. 686–694. ACM Press (2008)
18. You, C.H., Holder, L.B., Cook, D.J.: Learning patterns in the dynamics of biological networks. In: KDD. pp. 977–986. ACM Press (2009)
19. Zaki, M.J.: Mining non-redundant association rules. Data Min. Knowl. Discov. 9(3), 223–248 (2004)