Trend Mining in Dynamic Attributed Graphs

Elise Desmier[1], Marc Plantevit[2], Céline Robardet[1], and Jean-François Boulicaut[1]

[1]Université de Lyon, CNRS, INSA-Lyon, LIRIS UMR5205,
F-69621 Villeurbanne, France
elise.desmier@liris.cnrs.fr, celine.robardet@insa-lyon.fr,
Jean-Francois.Boulicaut@insa-lyon.fr
[2]Université de Lyon, CNRS, Univ. Lyon1, LIRIS UMR5205,
F-69622 Villeurbanne, France
marc.plantevit@liris.cnrs.fr

Abstract. Many applications see huge demands of discovering important patterns in dynamic attributed graph. In this paper, we introduce the problem of discovering trend sub-graphs in dynamic attributed graphs. This new kind of pattern relies on the graph structure and the temporal evolution of the attribute values. Several interestingness measures are introduced to focus on the most relevant patterns with regard to the graph structure, the vertex attributes, and the time. We design an efficient algorithm that benefits from various constraint properties and provide an extensive empirical study from several real-world dynamic attributed graphs.

1 Introduction

Data mining techniques are now sufficiently mature to investigate complex data such as graph, whose vertices stand for entities and edges represent their relationships or interactions. With the rapid development of social media, sensor technologies and bioinformatic assay tools, real-world graph data has become ubiquitous and new dedicated data mining techniques have been developed. Whereas dynamic graphs [2,4,13,15] and attributed graphs [12,14,16] have been separately considered so-far, we focus on the extraction of valuable information from dynamic attributed graphs. The simultaneous consideration of the graph structure, the vertex attributes and their evolution through time makes possible to tackle a wide variety of mining problems. A timely challenge is to provide tools and methods to describe the evolution of the whole graph but also the specific evolution of some particular sub-graphs.

The second problem was recently tackled in [6], where an algorithm that mines cohesive co-evolution patterns is proposed. These patterns identify sets of vertices that are similar from the point of view of their attribute values and of the vertices in their neighborhood. However, as this method under-utilizes the topological structure of the vertex sets (i.e., only similarity measure are computed from two vertex adjacency lists), it tends to fragment some reliable patterns.

In this paper, we propose to mine maximal dynamic attributed sub-graphs that satisfy some constraints on the graph topology and on the attribute values. To be more robust towards intrinsic inter-individual variability, we do not compare raw numerical values, but their trends, that is, their derivative at time stamp t. The connectivity of the dynamic sub-graphs is constrained by a maximum diameter value that limits the length of the longest shortest path between two vertices. Additional interestingness measures are used to assess the interest of the trend dynamic sub-graphs and guide their search by user-parameterized constraints. These constraints aim at answering the following questions:

- How similar are the vertices outside the trend dynamic sub-graph to the ones inside it?
- Are trends specific to the vertices of the pattern?
- What about the dynamic of the pattern? Does it appear suddenly or continuously?

The algorithm designed to compute these patterns traverses the lattice of dynamic attributed sub-graphs in a depth-first manner. It prunes and propagates constraints that are fully or partially monotonic or anti-monotonic [5], and thus takes advantage of a large variety of constraints that are usually not exploited by standard lattice-based approaches. To summarize, the main contributions of this paper are:

- The introduction of a novel problem: the discovery of trend dynamic sub-graphs in dynamic attributed graph. We define the trend dynamic sub-graph as a suitable mathematical notion for the study of dynamic attributed graphs and introduce the notions of vertex specificity, temporal dynamic, and trend relevancy characterizations.
- The design of an efficient algorithm that exploits the constraints, even those that are neither monotonic nor anti-monotonic.
- A quantitative and qualitative empirical study. We report on the evaluation of the efficiency and the effectiveness of the algorithm on several real-world dynamic attributed graphs.

The remainder of the paper is organized as follows. Section 2 defines the trend dynamic sub-graphs and their related interestingness measures. It also formalizes a new data mining task. Section 3 presents the algorithm that computes trend dynamic sub-graphs. An empirical evaluation on real-world attributed dynamic graphs is reported in Section 4. Section 5 discusses the related work. A conclusion ends the paper in Section 6.

2 Trend Dynamic Sub-graphs and Their Related Constraints

2.1 Trend Dynamic Sub-graphs

The input of our mining task is a dynamic graph $\mathcal{G} = \{G_t \mid t = 1 \dots t_{\max}\}$ over a discrete time span $T = [\![1, t_{\max}]\!]$. Each static graph is a non-directed

attributed graph $G_t = (V, E_t, A)$ where V is a set of n vertices $\{v_1, \ldots, v_n\}$ that is fixed throughout the time, $\{E_t \mid t \in T\}$ is a sequence of sets of edges that connect vertices of V at time t ($E_t \subseteq V \times V$), and A is a set of p ordinal attributes $\{a_1, \ldots, a_p\}$ whose values are defined for each vertex at each time step ($a_i : V \times T \to \mathbb{D}_i$, where $\mathbb{D}_i$ is the domain of a_i).

Intuitively, a *trend dynamic sub-graph* is an *induced* dynamic graph of $\mathcal{G}(V, T)$ whose vertices follow the same trend over a subset of attributes of A. Formally, given a subset of vertices $U \subseteq V$ and a subsequence $S = \langle t_1, \cdots, t_s \rangle$ of time stamps of T, the dynamic sub-graph of $\mathcal{G}$ induced by (U, S) is $\mathcal{G}(U, S) = \{G_t(U) \mid t \in S\}$ and $G_t(U)$ contains all the edges in E_t that have both ends in U. The induced dynamic graphs that are apt to convey a useful piece of information are those whose vertices follow a similar trend for a set of attributes, that is to say whose attribute value derivative at a time stamp t has the same sign over all the vertices and the time stamps of the dynamic sub-graph. We say that an attribute a shows an increasing trend over $\mathcal{G}(U, S)$, denoted a^+, if $\forall u \in U$ and $\forall t \in S$, $a(u, t) < a(u, t + 1)$. In a similar way, we also consider decreasing trend, a^-. Many trend dynamic sub-graph can be observed over a dynamic attributed graph, but those that are particularly important occur in nodes that are closely related through the induced sub-graph topology. To that end, we are looking for trend dynamic sub-graphs whose static induce sub-graphs have a small diameter. To summarize, a trend dynamic sub-graph is defined as follows:

Definition 1 (Trend dynamic sub-graph). *A trend dynamic sub-graph of an attributed dynamic graph* $\Big(\mathcal{G}(V, T), A \times \{+, -\}\Big)$ *is composed by (1) the induced dynamic sub-graph* $\mathcal{G}(U, S) = \{G_t(U) \mid t \in S\}$ *where* $U \subseteq V$ *and* $S = \langle t_1, \ldots, t_s \rangle$ *is a subsequence of* T*, and (2) a subset of signed attributes* Ω*, with* $\Omega \subseteq A \times \{+, -\}$*. It is denoted* $\big(\mathcal{G}(U, S), \Omega\big)$ *and satisfies the following properties :*

1. *At each time stamp* $t \in S$*, the sub-graph induced by* U *is* $G_t(U) = (U, F_t)$ *with* $F_t = E_t \cap (U \times U)$*.*
2. *At each time stamp* $t \in S$*, the diameter of the graph* $G_t(U)$ *is less than or equal to* k*, where* k *is a user-defined threshold. I.e., for any two vertices* $v, w \in U$*, there exists a path connecting them whose length is smaller than or equal to* k*. Formally, let* $d_{G_t(U)}(v, w)$ *be the shortest path length between the vertices* v *and* w *in* $G_t(U)$*. The diameter of* G *is thus defined by*

$$diam_{G_t(U)} \equiv \max_{v,w \in U} d_{G_t(U)}(v, w)$$

   and the diameter constraint, that is $diam_{G_t(U)} \leq k$, $\forall t \in S$*, is denoted* $diameter\big(\mathcal{G}(U, S), \Omega\big)$*.*
3. *Each signed attribute* $(a, m) \in \Omega$ *defined a trend that has to be satisfied by any vertex* $u \in U$ *at any timestamp* $t \in S$*:*

$$\begin{cases} a^+(u, t) \equiv a(u, t) < a(u, t + 1), \text{ if } m = + \\ a^-(u, t) \equiv a(u, t) > a(u, t + 1), \text{ if } m = - \end{cases}$$

   This constraint is denoted $trend\big(\mathcal{G}(U, S), \Omega\big)$*.*

4. *If* $\big(\mathcal{G}(U,S),\Omega\big)$ *is maximal, then the sets* U *and* Ω*, as well as the sequence* S *cannot be enlarged without invalidating one or more of the above properties. This constraint is denoted* $maximal\big(\mathcal{G}(U,S),\Omega\big)$.

2.2 Interestingness Measures

To further guide the extraction of trend dynamic sub-graphs toward most relevant ones, we propose several interstingness measures that offer the possibility to the end-users to express their needs. An interestingness measure is a function which assigns a value to a pattern according to its quality. Such a measure can easily be used as a constraint by specifying a user-defined threshold that makes possible the selection of patterns having a high or a low value on these measures.

Size measures: As most simple interestingness measures are often the most useful ones, we first consider size measures that characterize a pattern by the number of elements it contains: $sz\_vertices\big(\mathcal{G}(U,S),\Omega\big) = |U|$, $sz\_times\big(\mathcal{G}(U,S),\Omega\big) = |S|$ and $sz\_attributes\big(\mathcal{G}(U,S),\Omega\big) = |\Omega|$. These measures are generally used to constrain patterns to a minimal size.

Volume measure: In some contexts, it can also be useful to combine the three size measures in a single value: $volume\big(\mathcal{G}(U,S),\Omega\big) = \frac{|U|}{|V|} \times \frac{|S|}{|T|} \times \frac{|\Omega|}{|A|}$. This measure is also generally used to constrain patterns to a minimal volume.

Measure of vertex specificity: The question that aims to answer this measure is: How similar are the vertices outside the trend dynamic sub-graph to the ones inside it? We want to quantify the average proportion of trends that are satisfied by outside pattern vertices:

$$vertex\_specificity\big(\mathcal{G}(U,S),\Omega\big) = \frac{\sum_{w\in V\setminus U}\sum_{(a,m)\in\Omega}\sum_{t\in S}\delta_{a^m(w,t)}}{|V\setminus U|\times|\Omega|\times|S|}$$

where $\delta_{condition}$ is the Kronecker function that is equal to 1 if *condition* is satisfied, or 0 otherwise. The more the trend dynamic sub-graph is made of specific vertices with respect to attribute trends, the lower this measure.

Measure of trend relevancy: The question that aims to answer this measure is: Does the attributes that do not belong to Ω have an homogeneous trend on $\mathcal{G}(U,S)$? To that end, we evaluate the entropy of the attribute trends and consider the one that has the smallest entropy. Let

$$P_1(b^m,\mathcal{G}(U,S)) = \frac{\sum_{u\in U}\sum_{t\in S}\delta_{b^m(u,t)}}{\sum_{u\in U}\sum_{t\in S}\left(\delta_{b^-(u,t)}+\delta_{b^+(u,t)}\right)}$$

be the proportion of the trend m of attribute b on the vertices and time stamps of $\mathcal{G}(U,S)$. Then the trend relevancy interestingness measure is:

$$trend\_relevancy\big(\mathcal{G}(U,S),\Omega\big) = \min_{b\in A\setminus\Omega}\sum_{m\in\{-,+\}} -P_1(b^m,\mathcal{G}(U,S))\log P_1(b^m,\mathcal{G}(U,S))$$

The more a trend dynamic sub-graph is trend relevant, the higher this measure.

Measure of temporal dynamic: The question that aims to answer this measure is: How does a pattern appear in the time? Does it burst? To that end, we evaluate the dynamic of the proportion of vertices and attributes that satisfy the pattern before and after the time stamps of S: $P_2(t,(\mathcal{G}(U,S),\Omega)) = \frac{\sum_{u\in U}\sum_{(a,m)\in\Omega}\delta_{a^m(u,t)}}{|U|.|\Omega|}$. If a trend dynamic sub-graph bursts, then the proportion P_2 is below a threshold at every time stamps not in S:

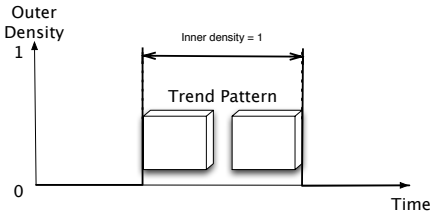

Fig. 1. Does the pattern burst ?

$$temporal\_dynamic(\mathcal{G}(U,S),\Omega) = \max_{t\in T\setminus S} P_2(t,(\mathcal{G}(U,S),\Omega))$$

3 Trend Sub-graph Enumeration

To compute all the trend attributed sub-graphs that satisfy the interestingness measures, we design `MINTAG` algorithm (for MINing Trend Attributed Graph) that enumerates induced dynamic sub-graphs based on the next partial order.

Definition 2 (Partial order on attributed induced dynamic sub-graphs). *Let $Q_1 = \big(\mathcal{G}(U_1,S_1),\Omega_1\big)$ and $Q_2 = \big(\mathcal{G}(U_2,S_2),\Omega_2\big)$ be two attributed induced dynamic sub-graphs. We say that Q_1 is more specific than Q_2, $Q_1 \preceq Q_2$, iff $U_1 \subseteq U_2$ and $S_1 \subseteq S_2$ and $\Omega_1 \subseteq \Omega_2$.*

This partial order forms a lattice: for any nonempty finite subset of attributed induced dynamic sub-graphs $\mathcal{F} = \{Q_i \mid i = 1\dots k\}$, $\mathcal{F}^\vee = \big(\mathcal{G}(\bigcup U_i, \bigcup S_i), \bigcup \Omega_i\big)$ and $\mathcal{F}^\wedge = \big(\mathcal{G}(\bigcap U_i, \bigcap S_i), \bigcap \Omega_i\big)$ are respectively the join and meet elements. The bounds of the lattice are $Q_\top = \big(\mathcal{G}(V,T), A\times\{+,-\}\big)$ and $Q_\bot = \big(\mathcal{G}(\emptyset,\emptyset),\emptyset\big)$. The enumeration strategy used by `MINTAG` is a binary partition [17]. In order to enumerate all the trend attributed sub-graphs $\mathcal{R}$ induced from $\big(\mathcal{G}(V,T), A\times\{+,-\}\big)$, a binary partition algorithm consists in choosing an element $e \in E = V \cup T \cup A \times \{+,-\}$ and divides $\mathcal{R}$ into two sets $\mathcal{R}_{+e}$ and $\mathcal{R}_{-e}$ so that $\mathcal{R}_{+e}$ consists of all the elements of $\mathcal{R}$ including e, and $\mathcal{R}_{-e}$ consists of those that do not include e. Therefore, e belongs to $\mathcal{R}^\wedge_{+e}$ and e does not belong to $\mathcal{R}^\vee_{-e}$. If $\mathcal{R}_{+e}$ (resp. $\mathcal{R}_{-e}$) is not empty and $\mathcal{R}^\vee_{+e} \neq \mathcal{R}^\wedge_{+e}$ (resp. $\mathcal{R}^\vee_{-e} \neq \mathcal{R}^\wedge_{-e}$), it is recursively divided by choosing another element in $\mathcal{R}^\vee_{+e}\backslash\mathcal{R}^\wedge_{+e}$ (resp. $\mathcal{R}^\vee_{-e}\backslash\mathcal{R}^\wedge_{-e}$). The number of iterations of a binary partition algorithm is linear in $|\mathcal{R}|$, which is the output size, if it is possible to check whether either $\mathcal{R}_{+e}$ or $\mathcal{R}_{-e}$ are empty. In the following, we explain how this test is performed.

3.1 Constraint Checking and Propagation Mechanisms

Let I and O be two subsets of E. We denote by $\mathcal{R}_{IO}$ a search space such that I is the set of elements that are included in all the patterns of $\mathcal{R}_{IO}$ and O is the

set of elements that cannot be included in any pattern of $\mathcal{R}_{IO}$. $\mathcal{R}_{IO}^{\vee}$ and $\mathcal{R}_{IO}^{\wedge}$ are respectively the join and meet elements of this search space and $I \subseteq \mathcal{R}_{IO}^{\wedge}$ and $O \cap \mathcal{R}_{IO}^{\vee} = \emptyset$. Checking whether the search space is empty can be done by evaluating the constraints on the join or the meet elements. Indeed, if a monotonic constraint is not satisfied by the join element, then $\mathcal{R}_{IO}$ is empty. Similarly, if an anti-monotonic constraint is not satisfied by the meet element, then $\mathcal{R}_{IO}$ is also empty. Constraints that are partially monotonic or anti-monotonic can also be pushed [5], as it is explained below.

Trend Sub-graph Constraints

Trend constraint: This constraint is anti-monotonic with respect to $\preceq$. That is, if Q_1 and Q_2 are two attributed induced dynamic sub-graphs such that $Q_1 \preceq Q_2$, then, $trend(Q_2) \Rightarrow trend(Q_1)$. The anti-monotonic property of the *trend* constraint implies that if $trend(\mathcal{R}_{IO}^{\wedge})$ is not satisfied, then $\mathcal{R}_{IO}$ is empty. In `MINTAG` algorithm, this constraint is propagated using the following procedure: if there exists e in $\mathcal{R}_{IO}^{\vee} \setminus \mathcal{R}_{IO}^{\wedge}$ such that $trend(\mathcal{R}_{IO}^{\wedge} \cup e)$ is not satisfied, then e is removed from $\mathcal{R}_{IO}^{\vee}$.
Diameter constraint: This constraint is neither monotonic nor anti-monotonic with respect to $\preceq$. However, noting that this constraint is monotonic or anti-monotonic in each of its parameters, we can derive a propagation mechanism of this constraint. That is, for all vertex v and all time stamp t in the trend sub-graph, we should have $\max_{w \in U_1} d_{G_t(U_2)}(v, w) \leq k$. This constraint is anti-monotonic on U_1 and monotonic on U_2, that is (a) if the constraint is satisfied on U_1, it is also satisfied for any of its subsets; (b) if the constraint is satisfied on a graph $G_t(U_2)$, then, adding some vertices and edges to $G_t(U_2)$ will not increase its value. Therefore, in `MINTAG` algorithm, this constraint is propagated using the following mechanisms: (1) if there exists $v \in \mathcal{R}_{IO}^{\vee} \setminus \mathcal{R}_{IO}^{\wedge}$, $w \in \mathcal{R}_{IO}^{\wedge}$ and $t \in \mathcal{R}_{IO}^{\wedge}$ such that $d_{G_t(\mathcal{R}_{IO}^{\vee} \cap V)}(v, w) > k$ then v is removed from $\mathcal{R}_{IO}^{\vee}$; (2) if there exists $t \in \mathcal{R}_{IO}^{\vee} \setminus \mathcal{R}_{IO}^{\wedge}$, $v \in \mathcal{R}_{IO}^{\wedge}$ and $w \in \mathcal{R}_{IO}^{\wedge}$ such that $d_{G_t(\mathcal{R}_{IO}^{\vee} \cap V)}(v, w) > k$ then t is removed from $\mathcal{R}_{IO}^{\vee}$.

Other Interestingness Constraints

Minimal size constraints: As these constraints are monotonic, if $sz\_vertices(\mathcal{R}_{IO}^{\vee} \cap V) <$ min_sz_vertices or $sz\_attributes(\mathcal{R}_{IO}^{\vee} \cap A \times \{+, -\}) <$ min_sz_attributes or $sz\_times(\mathcal{R}_{IO}^{\vee} \cap T) <$ min_sz_times, then $\mathcal{R}_{IO}$ is empty.

Minimal volume constraint: Similarly, this constraint is monotonic and if $volume(\mathcal{R}_{IO}^{\vee}) <$ min_volume, then $\mathcal{R}_{IO}$ is empty.

Maximal vertex_specificity constraint: As the diameter constraint, this constraint is monotonic or anti-monotonic on each of its parameters. Considering the equation $\frac{\sum_{w \in V \setminus U_1} \sum_{(a,m) \in \Omega_1} \sum_{t \in S_1} \delta_{a^m(w,t)}}{|V \setminus U_2| \times |\Omega_2| \times |S_2|} \leq$ max_vertex_spec, we can observe that it

is monotonic on U_1, S_2 and Ω_2 and anti-monotonic on U_2, S_1 and Ω_1. Thus, $\mathcal{R}_{IO}$ is empty if

$$\frac{\sum_{w\in(\mathcal{R}^{\vee}_{IO}\cap V)}\sum_{(a,m)\in(\mathcal{R}^{\wedge}_{IO}\cap(A\times\{+,-\})}\sum_{t\in(\mathcal{R}^{\wedge}_{IO}\cap T)}\delta_{a^m(w,t)}}{|\mathcal{R}^{\wedge}_{IO}\cap V|\times|\mathcal{R}^{\vee}_{IO}\cap(A\times\{+,-\})|\times|\mathcal{R}^{\vee}_{IO}\cap T|} > \text{max\_vertex\_spec}$$

Minimal trend_relevancy constraint: Handling this constraint is a little more tricky. Let us first consider the entropy function with two probability values: $f(x) = -x\log(x) - (1-x)\log(1-x)$. This function increases on $[0, \frac{1}{2}]$ and decreases on $[\frac{1}{2}, 1]$. Using this notation, the minimal trend_relevancy can be rewritten as $\min_{b\in A\setminus\Omega} f(P_1(b^+, \mathcal{G}(U,S)) \geq \text{min\_trend\_rel}$.[1] Second, we can derive the following upper bound on $P_1(b^m, \mathcal{G}(U,S))$:

$$P_1(b^m, \mathcal{G}(U,S)) \leq \frac{\sum_{u\in(\mathcal{R}^{\vee}_{IO}\cap U)}\sum_{t\in(\mathcal{R}^{\vee}_{IO}\cap S)}\delta_{b^m(u,t)}}{\sum_{u\in(\mathcal{R}^{\wedge}_{IO}\cap U}\sum_{t\in(\mathcal{R}^{\wedge}_{IO}\cap S)}\left(\delta_{b^-(u,t)}+\delta_{b^+(u,t)}\right)} = UB(b^m)$$

as P_1 is monotonic on its numerator parameters, and anti-monotonic on its denominator ones. Similarly, we can derive a lower bound[2] $LB(b^m) \leq P_1(b^m, \mathcal{G}(U,S))$. Thus, if $UB(b^m) \leq \frac{1}{2}$, then f is increasing and $f(P_1(b^m, \mathcal{G}(U,S))) \leq f(UB(b^m))$. Similarly, if $LB(b^m) \geq \frac{1}{2}$, then f is decreasing and $f(P_1(b^m, \mathcal{G}(U,S))) \leq f(LB(b^m))$.

Therefore, if there exists $b \in A \setminus \mathcal{R}^{\vee}_{IO}$ and $m \in \{+,-\}$ such that either (1) $UB(b^m) \leq \frac{1}{2}$ and $f(UB(b^m)) < \text{min\_trend\_rel}$, or (2) $LB(b^m) \geq \frac{1}{2}$ and $f(LB(b^m)) < \text{min\_trend\_rel}$ then $f(P_1(b^m, \mathcal{G}(U,S))) < \text{min\_trend\_rel}$ and we can conclude that $\mathcal{R}_{IO}$ is empty.

Maximal temporal_dynamic constraint: This constraint is anti-monotonic on its parameters on the numerator and monotonic on the ones on the denominator:

$$\max_{t\in T\setminus S}\frac{\sum_{u\in U}\sum_{(a,m)\in\Omega}\delta_{a^m(u,t)}}{|U|.|\Omega|} \leq \text{max\_temp\_dyn}$$

Therefore, if there exists $t \in T\setminus\mathcal{R}^{\vee}_{IO}$ such that $\frac{\sum_{u\in\mathcal{R}^{\wedge}_{IO}\cap U}\sum_{(a,m)\in\mathcal{R}^{\wedge}_{IO}\cap\Omega}\delta_{a^m(u,t)}}{|\mathcal{R}^{\vee}_{IO}\cap U|.|\mathcal{R}^{\vee}_{IO}\cap\Omega|} > \text{max\_temp\_dyn}$, then we can conclude that $\mathcal{R}_{IO}$ is empty.

3.2 `MINTAG` Algorithm

Algorithm 1 presents the main steps of `MINTAG`. Lines 1 and 2 initialize I and O to the emptyset. Line 3 and 4 initialize the sub-space join value to the lattice top and the meet value to the lattice bottom. Line 5 is the first call to `MINTAG_Enum` function which enumerates once and only once each trend dynamic sub-graph. The first line of the function tests if the search space contains a single trend dynamic sub-graph. If so, it is output. Line 4 reduces the search space join by

[1] This is equivalent to $\min_{b\in A\setminus\Omega} f(P_1(b^-, \mathcal{G}(U,S)) \geq \text{min\_trend\_rel}$ as $P_1(b^+, \mathcal{G}(U,S)) = 1 - P_1(b^-, \mathcal{G}(U,S))$.

[2] $LB(b^m) = \frac{\sum_{u\in(\mathcal{R}^{\wedge}_{IO}\cap U)}\sum_{t\in(\mathcal{R}^{\wedge}_{IO}\cap S)}\delta_{b^m(u,t)}}{\sum_{u\in(\mathcal{R}^{\vee}_{IO}\cap U}\sum_{t\in(\mathcal{R}^{\vee}_{IO}\cap S)}\left(\delta_{b^-(u,t)}+\delta_{b^+(u,t)}\right)} \leq P_1(b^m, \mathcal{G}(U,S))$

removing elements whose enumeration will emptied the search space due to the trend or the diameter constraints. Line 5 checks if the search space is empty by considering the maximality, minimal size, minimal volume, maximal vertex specificity, minimal trend relevancy and maximal temporal dynamic constraints. If one of these constraints is not relevant for the end-user, she can set the corresponding threshold to 0, for the minimal constraints, or to 1 for the other ones. In that case, these constraints do not coerce the result. If the search space is not empty, a new element, that belongs to the join but not to the meet, is enumerated. This element is first added to the search space meet before the recursive call (lines 7 and 8), and then it is removed from the search space join before the recursive call (lines 10 and 11).

Algorithm 1 `MINTAG`

Require: An attributed dynamic graph $\mathcal{G} = \{G_t = (V, E_t, A) \mid t \in T\}$ with $A\{a_1, \ldots, a_p\}$, $a_i : V \times T \rightarrow \mathbb{D}_i$ and the parameters.
Ensure: All trend dynamic subgraph that satisfy the constraints.

1: $I \leftarrow \emptyset$
2: $O \leftarrow \emptyset$
3: $\mathcal{R}^{\vee}_{IO} \leftarrow \Big(\mathcal{G}(V,T), A \times \{+,-\}\Big)$
4: $\mathcal{R}^{\wedge}_{IO} \leftarrow \Big(\mathcal{G}(\emptyset,\emptyset), \emptyset\Big)$
5: `MINTAG_Enum`$(\mathcal{R}^{\vee}_{IO}, \mathcal{R}^{\wedge}_{IO})$

Function `MINTAG_Enum`$(\mathcal{R}^{\vee}_{IO}, \mathcal{R}^{\wedge}_{IO})$

1: **if** $\mathcal{R}^{\vee}_{IO} = \mathcal{R}^{\wedge}_{IO}$ **then**
2: Ouput$(\mathcal{R}^{\vee}_{IO})$
3: **else**
4: $\mathcal{R}^{\vee}_{IO} \leftarrow$ `Constraint_Propagation`$(\mathcal{R}^{\vee}_{IO}, \mathcal{R}^{\wedge}_{IO})$
5: **if not** `Empty_Search_Space`$(\mathcal{R}^{\vee}_{IO}, \mathcal{R}^{\wedge}_{IO})$ **then**
6: **for all** $e \in \mathcal{R}^{\vee}_{IO} \setminus \mathcal{R}^{\wedge}_{IO}$ **do**
7: $I \leftarrow I \cup \{e\}$
8: `MINTAG_Enum`$(\mathcal{R}^{\vee}_{IO}, \mathcal{R}^{\wedge}_{IO} \cup \{e\})$
9: $I \leftarrow I \setminus \{e\}$
10: $O \leftarrow O \cup \{e\}$
11: `MINTAG_Enum`$(\mathcal{R}^{\vee}_{IO} \setminus \{e\}, \mathcal{R}^{\wedge}_{IO})$
12: $O \leftarrow O \setminus \{e\}$
13: **end for**
14: **end if**
15: **end if**

4 Experimental Study

In this section, we report on experimental results to illustrate the interest of the proposed approach. We start by describing the different real-world dynamic attributed graphs we use, as well as the questions we aim to answer. Then, we provide a performance study and give some qualitative results. All experiments were performed on a cluster. Nodes are equipped with 2 processors at 2.5GHz and 16GB of RAM under Linux operating systems. `MINTAG` algorithm is implemented in standard C++.

4.1 Real-world Dynamic Attributed Graphs Description

We considered 3 real-world dynamic attributed graphs whose characteristics are given in Figure 4.1.

Dynamic attributed graph		$\|V\|$	$\|T\|$	$\|A\|$	density
DBLP		2145	10	43	1.3×10^{-3}
US Flights	Last 20 years	361	20	8	3.2×10^{-2}
	September 2001	220	30	6	5.7×10^{-2}
	Two years around 9/11	234	25	8	5.7×10^{-2}
	Katrina	280	8	8	5×10^{-2}
Brazil landslides		394885	2	11	5.7×10^{-4}

Fig. 2. Main characteristics of the dynamic attributed graphs.

DBLP: This co-authorship graph is built from the DBLP digital library [3]. Each vertex represents an author who published at least ten papers in one of the major conferences and journals of the Data Mining and Database communities between January 1990 and December 2012. This time period is divided in 10 timestamps. Each timestamp describes the co-authorship relations and the publication records of the authors over 5 consecutive years. For sake of consistency in the data, two consecutive periods have a 3 year overlap[4]. Each edge at a time stamp t links two authors who co-authored at least one paper in this time interval. The vertex properties are the number of publications in each of the 43 journals or conferences.

US Flights: RITA "On-Time Performance" database[5] contains on-time arrival data for non-stop US domestic flights by major air carriers. From this database, we generated 4 dynamic attributed graphs that aggregate data over different period of time. Graph vertices stand for US airports and are connected by an edge if there is at least a flight connecting them during the time period. We consider 8 vertex attributes that are the number of departures/arrivals, the number of canceled flights, the number of flights whose destination airport has been diverted, the mean delay of departure/arrival and the ground waiting time departure/arrival. The four dynamic graphs are:

- Last 20 years: Data are aggregated over each year.
- September 2001: Data are aggregated over each day of September 2001.
- Two years around 9/11: Data are aggregated over each month between September 2000 and September 2002.
- Katrina: To study the consequences of hurricane Katrina on US airports, data are aggregated over each week between 01/08/2005 and 25/09/2005.

Brazil landslides: This dynamic attributed graph is derived from two satellite images taken before and after huge landslides in Brazil. It is composed of 394885 vertices that stand for image shapes (segmented areas), two time stamps and 11 attributes that are the spectral response in infra-red, red, blue green and indices computed from these values. There is an edge between two vertices if the corresponding shapes are contiguous.

The ensuing experimental study aims at answering the following questions: *What is the efficiency of* `MINTAG` *with regard to the graph characteristics that*

[3] http://dblp.uni-trier.de/
[4] [1990-1994][1992-1996][1994-1998]...[2008-2012]
[5] http://www.transtats.bts.gov

may affect its execution time? How effective are `MINTAG`*'s pruning properties? Does* `MINTAG` *scale? What about* `MINTAG`*'s trend dynamic sub-graph relevancy?*

4.2 Quantitative Results

We conduct intensive experiments to evaluate the performance of `MINTAG` in terms of computational cost and number of trend dynamic sub-graphs on different dynamic attributed graphs. Figure 3 shows the number of extracted patterns

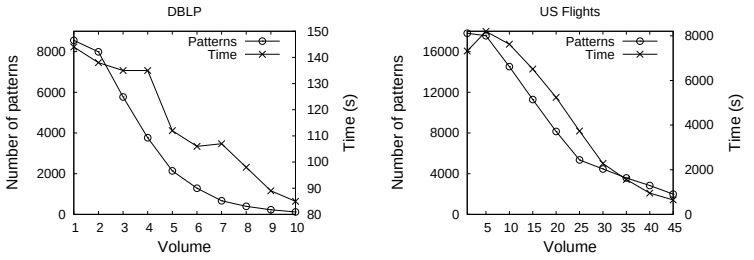

Fig. 3. Number of patterns and runtime for DBLP (left) and US flights (right) with respect to volume: max_vertex_spec = 0.5, min_trend_rel = 0.05 and max_temp_dyn = 0.8. The diameter is set to 2 on (left) and to 1 on (right).

and the execution times of `MINTAG` on DBLP and US Flights with respect to the volume threshold. When the minimum volume threshold decreases, more execution time is required since more trend dynamic sub-graph are obtained. Yet, `MINTAG` is able to extract trend dynamic sub-graphs when the minimum volume threshold is minimal, that is to say equals 1, since we report absolute volume values. `MINTAG` does not exhibit a similar monotonic behavior when varying the diameter constraint: the time computation is no more proportional to the number of extracted patterns. Actually, pushing this constraint needs to compute shortest paths in the graph, that is costly.

Figure 4 reports the execution times and the number of patterns with respect to the other interestingness measures: vertex specificity, trend relevancy and temporal dynamic. We can observe that for the graphs DBLP and US Flights, the less stringent the constraints, the higher the execution times and the number of patterns are. In most of the cases, the number of patterns increases dramatically. This behavior shows that our approach push efficiently these constraints that are neither monotonic nor anti-monotonic. It is noteworthy that in Figure 4, the execution time of `MINTAG` on DBLP for min_trend_rel = 0 is not available because the process was killed after several hours.

Figure 5 reports on the scability of `MINTAG`. We used DBLP and replicated alternatively the number of vertices, time stamps and attributes. As the number of extracted patterns is not preserved by these replications (i.e., the vertex replication adds connected components while the time replication introduces new

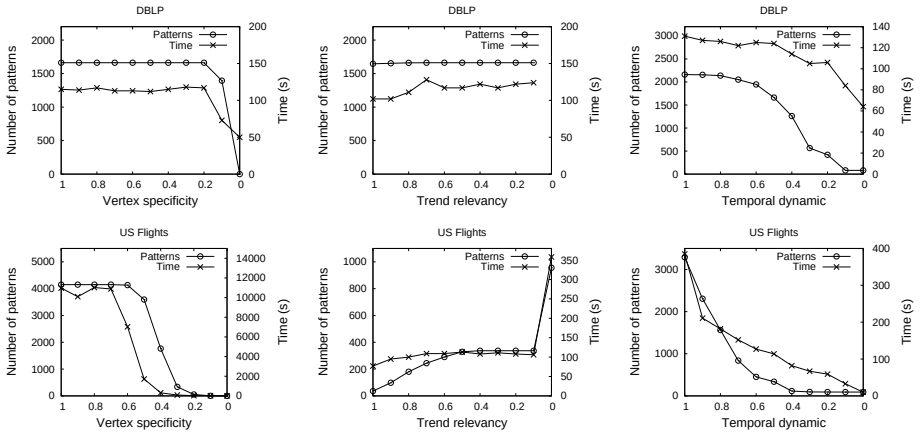

Fig. 4. Runtime and number of patterns with respect to the specificity measures (max_vertex_spec = 0.3, min_trend_rel = 0.1, max_temp_dyn = 0.5, min_volume = 5 and max_diameter = 2 for DBLP (top) or 1 for US flights (bottom)).

variations involving the last time stamp) we report the runtime per pattern. It appears that `MINTAG` is more robust to the increase of the number of attributes and to the number of vertices than to the number of time stamps. This is a good point since, in practice, the number of vertices is often large while the numbers of attributes and mainly the number of time stamps are rather small.

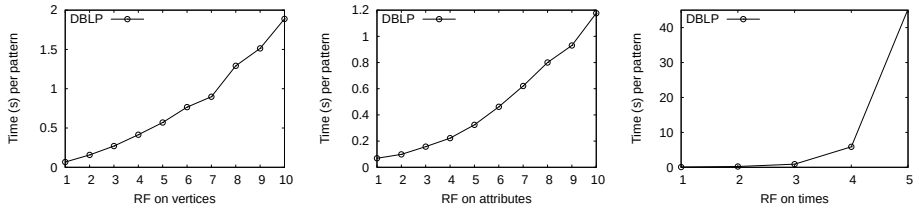

Fig. 5. Runtime per pattern with respect to replication factors on vertices, attributes and time stamps (max_vertex_spec = 0.3, min_trend_rel = 0.1, max_temp_dyn = 0.5, min_volume = 5 and max_diameter = 2.

We study the effectiveness of each constraint on both DBLP and US Flights, when varying the different thresholds (volume, vertex specificity, temporal dynamic and trend relevancy). To this end, we count the number of pruned unpromising candidates by each constraint. The results are shown in Figure 6 for DBLP (top) and US Flights (bottom). It is noteworthy that all the constraints enable to prune unpromising candidates and they have different impact on both graphs. We can observe that the trend relevancy constraint is effective on the

two graphs and prunes almost 50% of the unpromising candidates on DBLP in most of the cases. Even if this constraint has no anti-monotonic property, it is efficiently pushed in MINTAG. The volume constraint, more effective on DBLP than US Flights, makes possible to prune large part of the search space. This behavior is much more expected since this constraint is anti-monotonic. The pruning impact of the temporal_dynamic constraint is not negligible, since it prunes nearly 20% of the candidates on DBLP and up to 60% on US flights. This important difference is mainly due to the temporal regularity of US Flights. This can also explain the fact that the vertex specificity constraint plays a prominent role on the US Flights while having a limited impact of the DBLP dynamic graph.

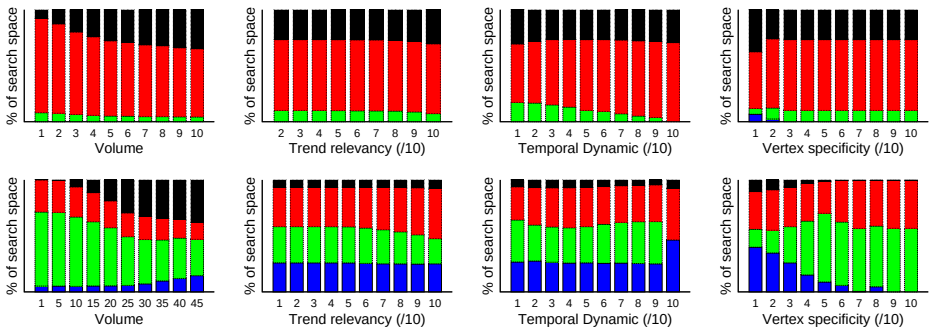

Fig. 6. Constraint efficiency on DBLP (top) and US Flights (bottom) w.r.t. specificity measures. From top to bottom: volume (black), trend_relevancy (red), temporal_dynamic (green) and vertex_specificity (blue). Same parameters as in Fig. 3 and 4.

4.3 Qualitative Results

Results on DBLP: We perform an extraction on DBLP dynamic attributed graph with max_diameter set to infinity (vertices belong to the same connected component) and min_volume = 5. Other constraints threshold are set so as not to constrain the result. We obtained 112 trend dynamic sub-graphs in less than 4 seconds. The top 2 largest patterns depict the same well-known phenomenon, explained below. The first pattern involves 171 authors having an increasing number of publications in PVLDB between 2004 and 2012. The second one involves 164 authors that have a decreasing number of publications in VLDB during the same period. These patterns reflect the new policy of the VLDB endowment. Indeed, PVLDB appeared in 2008 and, in 2010, the review process of the VLDB conference series was done in collaboration with, and entirely through PVLDB in 2011. Then, we carry out a new extraction taking into account all the constraints (max_diameter = 2, max_vertex_spec = 0.3, max_temp_dyn = 0.5) except min_trend_rel that was set to 0. We obtained 41 patterns in 8 seconds.

We first consider the pattern that has the longest duration and involves the most recent period, that is [2008-2012]. It implies the vertices related to Jimeng Sun and Christos Faloutsos, who have an increasing number of publications in KDD and SDM, while having a decreasing number of publications in VLDB. We consider another pattern which has the best *temporal_dynamic* value among the patterns having their *trend_relevancy* greater than 0.1. It involves two authors, Rong Zhou and Eric A. Hansen, and the time stamps between 1998 and 2008. On this period, the authors have an increasing number of publications in AAAI conference series. This pattern has good values on $vertex\_specificity$ (0.12), *temporal_dynamic* (0) and *trend_relevancy* (0.81). This publication trend is rare with regard to the whole graph.

Fig. 7. Airports (left) involved in the top temporal_dynamic trend dynamic sub-graph (in red) and in the top trend_relevancy (in yellow) and the Katrina's track (right).

Results on Katrina: Hurricane Katrina was the deadliest and most destructive Atlantic hurricane of the 2005 Atlantic hurricane season. It was the costliest natural disaster, as well as one of the five deadliest hurricanes, in the history of the United States. Among recorded Atlantic hurricanes, it was the sixth strongest overall. In this experiment, we aim to characterize the impact of this hurricane on the US domestic flights. To this end, we set constraints as follows: min_volume = 10, max_vertex_spec = 0.6, min_trend_rel = 0.1, max_temp_dyn = 0.2 and max_diameter = ∞. We extract 37 patterns in 14 seconds. We look for two patterns: (i) the trend dynamic sub-graph with largest *temporal_dynamic* value, and (ii) the pattern with the highest *trend_relevancy* value. These patterns and Katrina's track[6] are shown in Figure 7. Pattern (i) involves 71 airports (in red on Figure 7 (left)) whose arrival delays increase over 3 weeks. One week is not related to the hurricane but the two others are the two weeks after Katrina caused severe destruction along the Gulf coast. This pattern has a *temporal_dynamic* = 0, which means that arrival delays never increased in these airports during another week. The hurricane strongly influenced the domestic flight organization. Pattern (ii) has a *trend_relevancy* value equal to 0.81 and includes 5 airports (in yellow on Figure 7 (left)) whose number of departures and arrivals increased over the three weeks following Katrina hurricane. Three out of the 5 airports are in the Katrina's trajectory while the two other ones were

[6] Map from ©2013 Google, INEGI, Inav/Geosistemas SRL, MapLink
`http://commons.wikimedia.org/wiki/File:Katrina_2005_track`

Table 1. Trend dynamic sub-graphs extracted by MINTAG on September 2001 graph.

Pattern	$\|V\|$	Days	A	*vertex_spec.*	*temp._dyn.*	*trend_rel.*
P1	179	10, 11	#Cancel.$^+$	0.5	0.41	0.94
P2	111	13, 15	#Cancel.$^-$	0.52	0.83	0.9
P3	102	13, 14, 15	#Cancel.$^-$	0.6	0.84	0.81

impacted because of their connections to airports from damaged areas. Substitutions flights were provided from these airports during this period. The values on the other interestingness measures show that this behavior is rather rare in the rest of the graph ($vertex\_specificity = 0.29$, $temporal\_dynamic = 0.2$).

Results on September 2001: To characterize the impact of September 11 attacks, we look for patterns involving many airports (at least 100) whose trends are relevant ($trend\_relevancy = 0.8$). Given this setting, MINTAG returns 3 trend dynamic sub-graphs in 8 seconds. These patterns are reported in Table 4.3. They depict a large number of airports, whose number of canceled flights increased on September 11 and 12 compared to the previous days, and then decreased two days after the terrorist attacks (between the 13th and 16th September). These patterns identify the time required for a return to normal domestic traffic.

Results on Two years around 9/11: Considering longer periods before and after the September attacks, with more restrictive threshold values ($temporal\_dynamic = 1$, $vertex\_specificity = 0.5$ and $trend\_relevancy = 0.8$), we obtain 87 patterns in 67 seconds. The top $trend\_relevancy$ pattern involves 159 airports that have an increasing number of canceled flights in September 2001 and December 2000. Obviously, the number of canceled flights in September 2001 is related to terrorist attack. It is noteworthy that December 2000 snow storm had a similar impact on the cancellation of flights, because we do not quantify the strength of the trends. Actually, the number of canceled flights in September 2001 is four times bigger than the one in December 2000.

Results on Brazil landslides: In this series of 2 satellite images, the goal is to identify regions in which a landslide appears in the second image. Generally, the main consequence of a landslide if the disappearance of the vegetation. Therefore, we focus on the patterns that involve $NDVI^-$, since $NDVI$ is a computed index that quantifies the level of vegetation. MINTAG returns 4821 patterns in 2 hours that involve 34275 regions that are reported on Figure 8. These results were evaluated by an expert who testified that 69% of the true landslide regions appear in the computed patterns. These regions represent 46% of the extracted regions. The 54% remaining regions belong to one of the 4 following categories:(1) regions nearby true land-

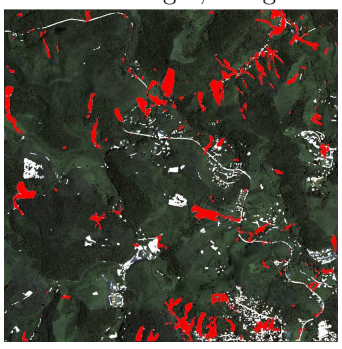

Fig. 8. Regions involved in the patterns: true landslides (red) and other phenomena (white).

slides which have not been interpreted as landslides by the expert (border effect), (2) deforested area not due to landslides (e.g., human activity), (3) regions found due to misalignment of the segmentation technique and (4) regions that represent cities and human activity footprints.

5 Related Work

Many proposals intend to characterize graph evolution by means of patterns or rules. Borgwardt et al. [4] introduce the problem of mining frequent sub-graphs in dynamic graphs. Lahiri and Berger-Wolf [10] extract frequent sub-graphs that appear periodically. Inokuchi and Washio [7] define frequent induced sub-graph subsequence whose isomorphic occurrences appear frequently in graph sequences. Ahmed et al. [1] propose to mine time-persistent edges and captures all maximal non-redundant evolution paths among them. You and Cook [18] compute graph rewriting rules that describe the evolution of consecutive graphs. Berlingerio et al. [2] extract patterns based on frequency and derive graph evolution rules. Descriptive n-ary association rules are defined in [13]. More recently, dynamic attributed graphs have received a particular interest. Boden et al. [3] propose to extract clusters in each static attributed graph and associate time consecutive clusters that are similar. Jin et al. [8] consider dynamic graph whose vertices are weighted. They extract groups of connected vertices whose vertex weights follow a similar evolution, increasing or decreasing, on consecutive time stamps. Desmier et al. [6] discover neighborhood similar set of vertices whose attributes follow the same trends. All the above works only assess the interest of the patterns by means of frequency-based constraints. They do not specify additional interestingness measures to guide the search toward relevant patterns. However, such constraints have been extensively studied in itemset mining, but not yet in dynamic attributed graph settings. To name a few, Morishita et al. [11] define a theoretical framework to compute significant association rules according to statistical measures and Kuznetsov [9] defines the stability of a formal concept.

6 Conclusion

In this paper, we propose to extract dynamic sub-graphs that have a small diameter. These dynamic sub-graphs are characterized by the attributes that have the same trend over the pattern vertices at each pattern time stamps. To only compute the most significant trend dynamic sub-graphs, we define three interestingness measures. We design an algorithm that actively uses all the constraints, even those that are neither monotonic nor anti-monotonic. It reduces the search space while preserving the completeness of the extraction. We provide experiments that prove that `MINTAG` computes the trend dynamic sub-graph in a feasible time. Moreover, experiments on real-world dynamic attributed graphs show that our method allows to extract truly relevant patterns.

Acknowledgements The authors thank ANR for supporting this work through the FOSTER project (ANR-2010-COSI-012-02). They also acknowledge support from the CNRS/IN2P3 Computing Center and the ICube laboratory for providing and preprocessing the Brazil landslide data.

References

1. Ahmed, R., Karypis, G.: Algorithms for Mining the Evolution of Conserved Relational States in Dynamic Networks. In: ICDM. pp. 1–10. IEEE (2011)
2. Berlingerio, M., Bonchi, F., Bringmann, B., Gionis, A.: Mining Graph Evolution Rules. In: ECML/PKDD. pp. 115–130. Springer (2009)
3. Boden, B., Günnemann, S., Seidl, T.: Tracing clusters in evolving graphs with node attributes. In: CIKM. pp. 2331–2334 (2012)
4. Borgwardt, K.M., Kriegel, H.P., Wackersreuther, P.: Pattern mining in frequent dynamic subgraphs. In: ICDM. pp. 818–822. IEEE (2006)
5. Cerf, L., Besson, J., Robardet, C., Boulicaut, J.F.: Closed patterns meet n-ary relations. TKDD 3(1), 3:1–3:36 (Mar 2009)
6. Desmier, E., Plantevit, M., Robardet, C., Boulicaut, J.F.: Cohesive co-evolution patterns in dynamic attributed graphs. In: Discovery Science. pp. 110–124 (2012)
7. Inokuchi, A., Washio, T.: Mining frequent graph sequence patterns induced by vertices. In: SDM. pp. 466–477. SIAM (2010)
8. Jin, R., McCallen, S., Almaas, E.: Trend Motif: A Graph Mining Approach for Analysis of Dynamic Complex Networks. In: ICDM. pp. 541–546. IEEE (2007)
9. Kuznetsov, S.O.: On stability of a formal concept. Ann. Math. Artif. Intell. 49(1-4), 101–115 (2007)
10. Lahiri, M., Berger-Wolf, T.Y.: Mining periodic behavior in dynamic social networks. In: ICDM. pp. 373–382. IEEE (2008)
11. Morishita, S., Sese, J.: Traversing itemset lattice with statistical metric pruning. In: PODS. pp. 226–236 (2000)
12. Moser, F., Colak, R., Rafiey, A., Ester, M.: Mining cohesive patterns from graphs with feature vectors. In: SDM. pp. 593–604. SIAM (2009)
13. Nguyen, K.N., Cerf, L., Plantevit, M., Boulicaut, J.F.: Discovering descriptive rules in relational dynamic graphs. Intell. Data Anal. 17(1), 49–69 (2013)
14. Prado, A., Plantevit, M., Robardet, C., Boulicaut, J.F.: Mining graph topological patterns. IEEE TKDE pp. 1–14 (2013)
15. Robardet, C.: Constraint-Based Pattern Mining in Dynamic Graphs. In: ICDM. pp. 950–955. IEEE (2009)
16. Silva, A., Meira Jr., W., Zaki, M.J.: Mining attribute-structure correlated patterns in large attributed graphs. PVLDB 5(5), 466–477 (2012)
17. Uno, T.: An efficient algorithm for solving pseudo clique enumeration problem. Algorithmica 56(1), 3–16 (2010)
18. You, C.H., Holder, L.B., Cook, D.J.: Learning Patterns in the Dynamics of Biological Networks. In: KDD. pp. 977–985. ACM (2009)