Handbook of Research on Methods and Techniques for Studying Virtual Communities:

Paradigms and Phenomena

Ben Kei Daniel
University of Saskatchewan and Saskatoon Health Region, Canada

Volume I

Information Science REFERENCE

INFORMATION SCIENCE REFERENCE
Hershey · New York

Director of Editorial Content: Kristin Klinger
Director of Book Publications: Julia Mosemann
Acquisitions Editor: Lindsay Johnston
Development Editor: Julia Mosemann
Publishing Assistant: Deanna Jo Zombro
Typesetter: Natalie Pronio and Deanna Jo Zombro
Production Editor: Jamie Snavely
Cover Design: Lisa Tosheff

Published in the United States of America by
Information Science Reference (an imprint of IGI Global)
701 E. Chocolate Avenue
Hershey PA 17033
Tel: 717-533-8845
Fax: 717-533-8661
E-mail: cust@igi-global.com
Web site: http://www.igi-global.com

Library of Congress Cataloging-in-Publication Data

Handbook of research on methods and techniques for studying virtual communities : paradigms and phenomena / Ben Kei Daniel, editor.
p. cm.
Includes bibliographical references and index.
ISBN 978-1-60960-040-2 (hbk.) -- ISBN 978-1-60960-041-9 (ebook) 1. Electronic villages (Computer networks)--Social aspects. 2. Online social networks. 3. Internet--Social aspects. I. Daniel, Ben Kei, 1971-
TK5105.83.H36 2011
303.48'34--dc22
2010042272

British Cataloguing in Publication Data
A Cataloguing in Publication record for this book is available from the British Library.

Chapter 5
Data Mining Techniques for Communities' Detection in Dynamic Social Networks

Céline Robardet
Université de Lyon, France

ABSTRACT

Social network analysis studies relationships between individuals and aims at identifying interesting substructures such as communities. This type of network structure is intuitively defined as a subset of nodes more densely linked, when compared with the rest of the network. Such dense subgraphs gather individuals sharing similar property depending on the type of relation encoded in the graph. In this chapter we tackle the problem of identifying communities in dynamic networks where relationships among entities evolve over time. Meaningful patterns in such structured data must capture the strong interactions between individuals but also their temporal relationships. We propose a pattern discovery method to identify evolving patterns defined by constraints. In this paradigm, constraints are parameterized by the user to drive the discovery process towards potentially interesting patterns, with the positive side effect of achieving a more efficient computation. In the proposed approach, dense and isolated subgraphs, defined by two user-parameterized constraints, are first computed in the dynamic network restricted at a given time stamp. Second, the temporal evolution of such patterns is captured by associating a temporal event types to each subgraph. We consider five basic temporal events: the formation, dissolution, growth, diminution and stability of subgraphs from one time stamp to the next one. We propose an algorithm that finds such subgraphs in a time series of graphs processed incrementally. The extraction is feasible thanks to efficient pruning patterns strategies. Experimental results on real-world data confirm the practical feasibility of our approach. We evaluate the added-value of the method, both in terms of the relevancy of the extracted evolving patterns and in terms of scalability, on two dynamic sensor networks and on a dynamic mobility network.

DOI: 10.4018/978-1-60960-040-2.ch005

INTRODUCTION

Social network analysis conceives social relationships in terms of graphs of interactions whose nodes represent individual actors within the networks and links social interactions such as ideas, friendship, collaboration, trade, etc. Virtual communities, and particularly online communities, are peculiar social networks whose analysis is facilitated by the fact that the network is in some sense monitored continuously. Social networks have attracted a large amount of attention from epidemiologists, sociologists, biologists and computer scientists that have shown the ubiquitous role played by social networks in determining the way problems are solved or organizations are run. The study of such networks has attracted much attention in the recent years and has proceeded along two main tracks: the analysis of graph properties, such as degree distribution, diameter or simple graph patterns such as cliques (Scherrer et al., 2008, Leskovec et al., 2005), and the identification of communities, which are loosely defined as collections of individuals who interact unusually frequently (Newmann, 2004, Palla et al., 2005). Communities reveal properties shared by related individuals. However, most of the interesting real-world social networks that have attracted the attention of researchers in the last few years are intrinsically time dependent and tend to change dynamically. As new nodes and edges appear while some others disappear over time, it seems decisive to analyze deeply the evolution of such dynamic graphs. Furthermore, there is a crucial need for incremental methods that enable to find groups of associated nodes and detect how these structures change over time.

Communities are loosely defined as highly connected subgraphs that are also isolated from the rest of the graph. Such properties can be captured by measures such as modularity (Newman, 2004) used to find disjoint communities forming a partition. The modularity of a given partition of nodes is the number of edges inside clusters (as opposed to crossing between clusters), minus the expected number of such edges if the graph was random conditioned on its degree distribution. Community structures often maximize the modularity measure. However, this measure has an intrinsic resolution scale, and can therefore fail to detect communities smaller than that scale and favor in general communities of similar size (Fortunato et al., 2007). Moreover, it has been shown (Brandes et al., 2008) that finding the community structure of maximum modularity for a given graph is NP-complete and thus heuristics have been proposed that approximate this optimization problem.

Instead of directly looking for a global structure of the graph, such as a partition of the vertices, it can be more efficient to proceed in two steps. One might first compute subgraphs that capture locally strong associations between vertices and then use these local patterns to construct a global model of the graph's dynamics. Such a framework provides more interesting patterns when the analyst can specify his inclination by means of constraints. Many pattern mining under local constraints techniques (e.g., looking for frequent patterns, data dependencies) have been studied extensively the last decade (Morik et al., 2005). One crucial characteristic of local pattern mining approaches is that the interestingness of a pattern can be computed independently of the other patterns. Such framework enables the analyst to specify a priori relevancy of pattern by means of constraints. The constraints have been identified as a key issue to achieve the tractability of many data mining tasks: useful constraints can be deeply pushed into the extraction process such that it is possible to get complete (every pattern which satisfies the user-defined constraint is computed) though efficient algorithms.

Specific subgraphs defined by constraints have already been examined. Fully connected subgraphs, also called cliques, are a local pattern type that has been considered as communities. Palla et al. (2005) consider that communities rely on several complete (fully connected) subgraphs of size

k that share *k-1* of their nodes. Such structures can be explored systematically with a deterministic algorithm. Although clique is a popular pattern type that captures dense subgraphs, it fails in properly handling experimental data that are intrinsically noisy. Indeed, in such data, some links may be missing even in dense substructures. To cope with this problem, a relaxed definition of cliques has been proposed. Pseudo cliques are natural extension of cliques which are subgraphs obtained by removing a small number of edges from cliques, expressed as a proportion compared to the number of links the subgraph would contained if it was a clique. Thus, pseudo cliques are subgraphs with a density higher than a given threshold and recent research results have shown that the constraints defining pseudo cliques can be efficiently used in a mining algorithm (Uno, 2007). We extend this result to derive a new algorithm that extracts isolated pseudo-cliques and their evolution in time. We consider five basic temporal event types that are associated to the computed subgraphs: the formation, dissolution, growth, diminution and stability of such patterns. Such evolving patterns make possible to describe the processes by which communities come together, attract new members, and develop over time. We propose an algorithm that mines such evolving patterns. The use of complete solvers allows us to answer constraint user queries without uncertainty. Algorithmic technical details can be found in (Robardet, 2009). In this chapter, we provide much more details and examples on how the proposed method identifies communities.

This chapter is organized as follows. The next section is dedicated to related work on the subject. It is followed by the presentation of the constraints that define the pattern types extracted in static graph. Then, the evolving pattern types are introduced. An algorithm that mines them is presented. Some experimental results are thus reported. Finally, some conclusions and future work close this chapter.

RELATED WORK

There is an increasing interest in mining dynamic graphs. Earlier work studied the properties of the time evolution of real graphs such as densification laws and shrinking diameters (Leskovec et al., 2005), and the evolution of known communities over time (Backstrom et al., 2006). Other papers have focused on community extraction thanks to constrained optimization (Chi et al., 2007), low-rank matrix approximation approaches (Tong et al., 2008), information theoretic principles (Sun et al., 2007) or combinatorial optimization problems (Tantipathananandh et al., 2007).

Another body of work considers constrained-based mining approaches to extract knowledge from static graphs. Efficient algorithms that compute maximal cliques have been proposed (Makino et al., 2004). Many papers propose to relax the clique property by allowing the absence of some links. Strongly self-referring subgraphs are defined in (Hamalainen et al., 2004) as a set of nodes S whose nodes are connected to at least a given proportion of nodes of S. Zhu et al. (2007) give a comprehensive study on the pruning properties of constraints on graphs. They study the pruning properties for involved structural constraints in graph mining which achieve pruning on the pattern search space and data space. A general mining framework is proposed that incorporates these pruning properties.

Pseudo clique mining, defined as the search for subgraphs having a density greater than a user-defined threshold, was first studied in (Pei et al., 2004), but the complete exploitation of the loose anti-monotonicity property of the pseudo clique constraint was only achieved in (Uno et al., 2007) where a polynomial delay algorithm that extracts all pseudo cliques is proposed.

Considering the extraction of patterns in dynamic graphs, Borgwardt et al. (2006) propose to apply frequent subgraph mining algorithms to time series of graphs to extract subgraphs that are frequent within the set of graphs. The extraction of

periodic or near periodic subgraphs is considered in (Lahiri et al., 2008) where the problem is shown to be polynomial. Finally, the so-called *change mining framework* is proposed in (Böttcher et al., 2008) as an abstract knowledge discovery process based on models and patterns learned from a non-stationary population. Its objective is to detect and analyze when and how changes occur, including the quantification, interpretation and prediction of changes.

IDENTIFYING DENSE AND ISOLATED SUBGRAPHS IN A STATIC GRAPH

Let us first present the static pattern type we are interested in. Let G=(V,E) be a simple undirected graph with a vertex set V and an edge set E. The subgraph induced by a subset of vertices S is the graph $G_S=(S,E_S)$ where Es={{u,v}∈ E and u,v ∈ S^2}. The degree, $deg_s(u)$, of a vertex u on the subgraph induced by S is the number of vertices of S adjacent to u, i.e., $deg_S(u)=|\{v \in S$ such that $\{u,v\} \in E\}|$.

Subgraphs of interest are usually those made of vertices that have a high density of edges. If any pair of vertices in a subgraph is connected by an edge, the subgraph is called a clique. Such subgraphs have a density of 1, where density is the number of edges in the subgraph divided by the maximal number of possible edges. To relax this strong property, we can consider subgraphs with density higher than a user-defined threshold. Such subgraphs are usually called pseudo cliques or quasi cliques. Given a user-defined threshold σ∈ [0,1] and a set of nodes S of size n, the subgraph $G_S=(S,E_S)$ induced by S is a pseudo clique if and only if it is connected and $2|E_S|/(n(n-1))>\sigma$.

Constraint-based mining algorithms require taking advantage of the constraints to prune huge parts of the search space which can not contain valid patterns. Pruning based on monotonic or anti-monotonic constraints has been proved efficient on hard problems since when a candidate does not satisfy the constraint then none of its generalizations or specializations can satisfy it as well.

Let us first remark that pseudo clique constraint is not anti-monotonic with respect to the enumeration of induced subgraphs based on the set inclusion of their vertices set: expanding a set of nodes S could make $2|E_S|/(n(n-1))$ increase or decrease. However, this constraint is loose anti-monotonic, that is to say, pseudo cliques can always be grown from a smaller pseudo cliques with one vertex less (Zhu et al., 2007, Bonchi et al., 2007). Zhu et al. have shown that if S is a valid pseudo clique, thus the set obtained by removing from S a vertex having the smallest degree on S is also a pseudo clique. Figure 1 illustrates this property: S={1,2,3,4,5} is a pseudo-clique with σ=2/3. If we remove node 2 that has the smallest degree

Figure 1. Illustration of the loose anti-monotonicity of pseudo-cliques

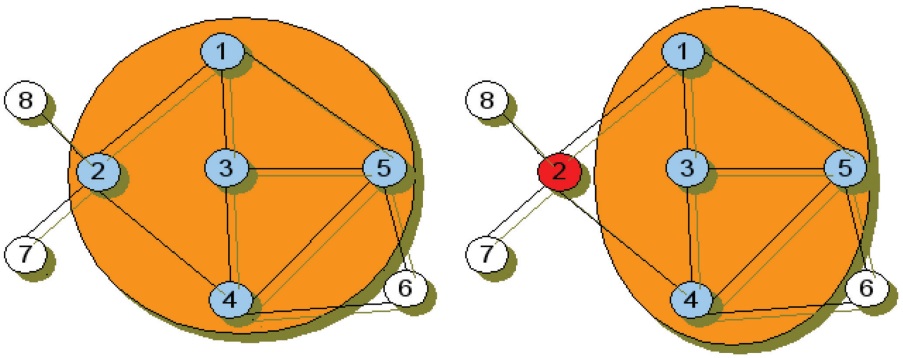
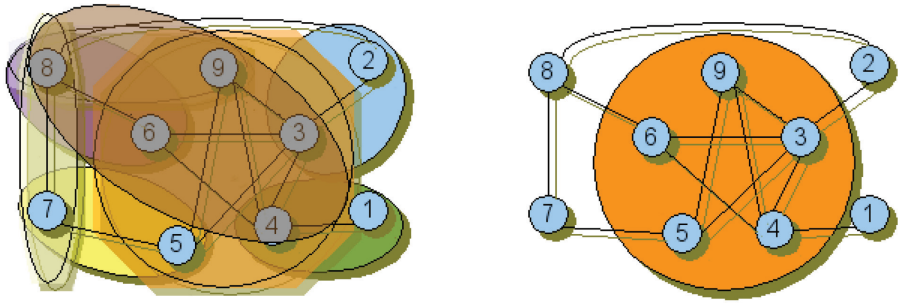


Figure 2. Pseudo cliques with σ=0.7 (left), and pseudo cliques (σ=0.7) that are isolated (a=1) and maximal (right)



on S, the resulting subgraph {1,3,4,5} is a also a valid pseudo-clique.

To be efficient, the pseudo cliques enumeration process must tap the pruning power from the loose anti-monotonicity of pseudo clique. It is clear that adding to a current pseudo clique S the node v that satisfies

$$\deg_{S\cup\{v\}}(v) = \min_{u\in S\cup\{v\}} \deg_{S\cup\{v\}}(u) \qquad (1)$$

leads to a pseudo clique, unless none of the supersets of S is a pseudo clique. Thus, an efficient algorithm would enumerate recursively nodes by finding at each iteration the node v that satisfies (Equation 1) and stop the enumeration if the obtained subgraph is not a valid pseudo clique. Note that if several nodes satisfy (Equation 1), the one of smallest index is taken. This leads to a polynomial delay time algorithm, that is to say the time needed to generate each single pseudo clique is bounded by a polynomial in the size of the input graph. Uno 2007 proposes an algorithm that checks if a subgraph is dense in constant time and finds the next vertex to be enumerated in $O(max_{v\in V} deg_S(v))$.

Pseudo cliques are local patterns that capture strong while not perfect associations in a graph. But, not all the pseudo cliques of a graph are of importance: some of them have many links to outside vertices, others are redundant. Figure 2 (left) illustrates this phenomenon. 9 pseudo cliques have been extracted (σ=0.7) in the graph. These pseudo cliques are highly redundant.

To select the most useful pseudo cliques, we consider two other constraints that coerce the patterns to be isolated and maximal. To pick out pseudo cliques S with few links to nodes outside S, we constrain the average number of outside links per vertex. This constraint is similar to the isolated constraints defined for formal concepts and their generalization in (Cerf et al., 2008). Given a user defined threshold α∈R, a subgraph S is isolated iff

$$\frac{\sum \deg_V(u) - \deg_S(u)}{|S|} \leq \pm$$

Figure 2 (right) shows that, with a=1, a single isolated and maximal pseudo clique is extracted in the graph example. Such a pseudo clique has an average number of outside links per vertex lower or equal to 1.

Even though this constraint is also loose anti-monotonic, its combination with the high density constraint is not loose anti-monotonic constraint. The two constraints cannot be ensured at the same time by an algorithm that exploits both loose anti-monotonic constraints. Hence, we propose

to ensure the new constraint in a post-processing of the previously computed pseudo cliques.

Extracting maximal patterns is even more difficult, since this constraint is global and requires enumerating supersets of a candidate to check whether it is maximal. A practical approach consists in extracting locally maximal isolated pseudo cliques. A subgraph S of size n is a local maximal isolated pseudo clique if it satisfies the two constraints and no supersets of S of size n+1 satisfies these two properties. With this constraint, the very large majority of non-maximal isolated pseudo cliques are removed, whereas the time complexity of the extraction remains the same.

MINING EVOLVING SUBGRAPHS

Local pattern mining algorithms provide a frequently large and unstructured set of patterns that cannot be readily interpreted or exploited by the users (De Raedt et al., 2007). We propose to complement the first phase where potentially interesting subgraphs are mined in static graphs, with a second phase, in which sets of pattern are post-processed to answer temporal queries on dynamic graphs.

We consider a dynamic graph $\hat{G} = (G^1, \ldots, G^T)$ which is a time-series of T graphs, where $G^t=(V^t,E^t)$ is the graph with edges E^t observed at time t, among the vertices of V^t.

The typical questions we want to consider are:

- Do the strong interactions observed at time t grow, diminish or remain the same over time?
- When do these subgraphs appear and disappear?

The objective here is to identify the temporal relationships that may occur between valid (i.e., locally maximal isolated) pseudo cliques. We denote by C^t the set of subgraphs of G^t that satisfy the constraints. We consider five basic temporal relationships between couples of subgraphs from consecutive time stamps:

Stability: S is said to stay the same at time t if it is a valid pseudo clique at time t and t-1: $S \in C^t \wedge S \in C^{t-1}$

- **Growth:** a subgraph S enlarges at time t if S is a valid pseudo clique at time t and a subpart of it forms a valid pseudo clique at time t-1: $S \hat{I} C^t \grave{U} \$R, R \grave{I} S$ such that $R \hat{I} C^{t-1}$
- **Diminution:** a subgraph S shrinks at time t if S is a valid pseudo clique at time t and is a subpart of a larger valid pseudo clique of time t-1: $S \hat{I} C^t \grave{U} \$R, S \grave{I} R$ such that $R \hat{I} C^{t-1}$
- **Extinction:** a subgraph S disappears at time t if it is a valid pseudo clique at time t-1 and if it is not involved in any previously defined pattern at time t: $S \hat{I} C^{t-1} \grave{U} "R$ such that $R \acute{I} S$, $R \ddot{I} C^t \grave{U} "R$ such that $S \acute{I} R, R \ddot{I} C^t$
- **Emergence:** a subgraph S emerges at time t if it is a valid pseudo clique in G^t and if none of its subsets or supersets are valid pseudo cliques in G^{t-1}: $S \hat{I} C^t \grave{U} "R$ such that $R \acute{I} S$, $R \ddot{I} C^{t-1} \grave{U} "R$ such that $S \acute{I} R, R \ddot{I} C^{t-1}$

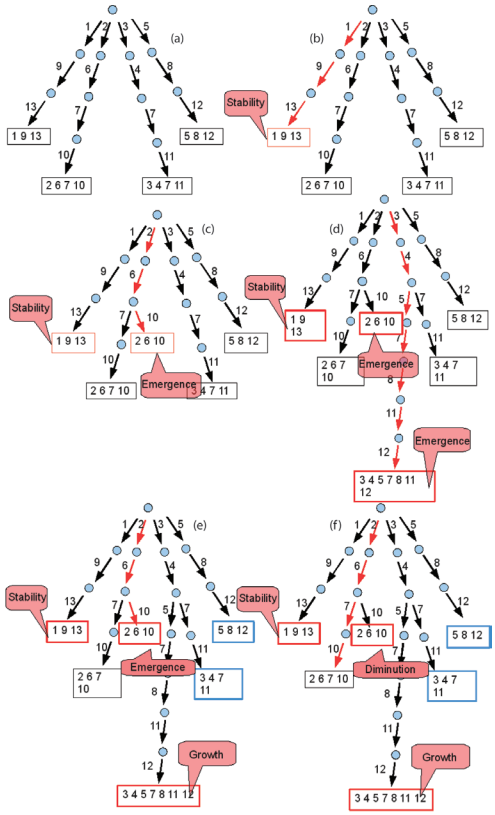
Those temporal relationships correspond to global constraints used to identify the dynamics of strong associations in graphs. We now present an incremental algorithm that processes each static graph sequentially. Inspired by the Trie-based Apriori implementation (Bodon, 2005), we propose to use a trie data structure (prefix tree) to store valid pseudo-cliques. Indeed, finding evolving patterns requires the evaluation of subset queries over valid pseudo-cliques. Such queries are computationally consuming and require special attention.

Suppose that pseudo-cliques of C^{t-1} are stored in a trie T. Each node of T consists of the set S of all

the vertices of the pseudo-clique, a list of temporal states, a list of pointers to other trie nodes and a list of time stamps. When a new valid pseudo-clique of G^t is computed, its vertex set S is inserted in T recursively. Figure 3 illustrates this process. Figure 3-A corresponds to the trie that contains the 4 pseudo cliques of time stamp t-1: {1,9,13}, {2,6,7,10}, {3,4,7,11} and {5,8,12}. Figure 3-B corresponds to the insertion of the valid pseudo-clique {1,9,13} of time stamp t: starting from the root node, we first go to the child corresponding to the first vertex of S ({1}) and process the remainder of S ({9,13}) recursively for that child. The recursion stops on a node whose vertex set is either S, or a prefix of S:

1. In the first case, the temporal label *"Stability"* is pushed back in the temporal label list of the node and its time stamp is set to t.

Figure 3. Evolving subgraphs construction



2. In the latter case, the node gets a new son with vertex set S, time stamp t and temporal label *"Emergent"* (see Figure 3-C where the pseudo-clique {2,6,10} is inserted and Figure 3-D where {3,4,5,7,8,11,12} is inserted).

Then we look whether S is involved in a growing evolving pattern. To do so, we have to retrieve all the subsets of S from T by means of the following doubly recursive procedure: We first go to the child corresponding to the first vertex of S and process the remainder of S recursively for that child and second discard the first vertex of S and process it recursively for the node itself. If there exists subsets of S that belongs to T with time stamp label t-1, then the temporal state associated to S is changed into *"Growth"* (see Figure 3-E) and pointers to the corresponding subsets are stored in the list associated to the node. Those nodes are also tagged to avoid their consideration in the following step.

Now, we need to check whether the pseudo-cliques of time stamp t-1 have shrunk (*"Diminution"*) or completely disappeared (*"Extinction"*). As tries are more effective to find subsets than to find supersets, a second traversal of the trie is performed when all pseudo cliques of C^t have been processed. For all the nodes with time stamp t-1 that are not involved in "Stability" or "Growth" pattern, the function that searches subsets is triggered. If there exists a subset that belongs to C^t, the state of the first node is set to *"Diminution"* and pointers to the corresponding subsets are stored in the node list, otherwise the state is set to *"Extinction"*, the pattern is output and the node is removed from the trie. For example, Figure 3-C illustrates the insertion of the pseudo-clique {2 6 10} whose temporal label is *"Emergent"*. When all the pseudo-cliques of time stamp t are inserted, the second traversal of the trie is performed and the label of this node is set to *"Diminution"* (see Figure 3-F).

EXPERIMENTATION

We evaluate the added-value of Evolving-Subgraphs and the general characteristics of evolving patterns subgraphs on three real-world dynamic networks: two dynamic sensor networks, imote and mit, and a dynamic mobility network velov, the shared bicycle system of Lyon. The main characteristics of these datasets are presented on Table 1. All experiments were done on a Pentium 3 with 2 Giga of memory running on Linux.

Dynamic Sensor Networks

The two studied mobility networks used are based on sensor measurements. The imote (Chaintreau et al., 2005) data set has been collected during the Infocom 2005 conference. Bluetooth sensors have been distributed to a set of participants who were asked to keep the sensors with them continuously. These sensors were able to detect and record the presence of other Bluetooth devices inside their radio-range neighborhood. The available data concern 41 sensors over a period of nearly 3 days which represent 254151 seconds. The mit or Reality Mining (Eagle et al., 2006) experimental data set constitutes of records from Bluetooth contacts for a group of cell-phones distributed to 100 mit students during

9 months. Each cellular phone conducts a Bluetooth device discovery scan and records the identities of all devices present in its neighborhood at a sampling period of 300 seconds. For both data sets, the Bluetooth devices may discover any kind of Bluetooth objects in its neighborhood. We have restricted our analysis to internal contacts only. Note also that the sensors had no localization capability. Therefore we do not have information on the actual movements of individuals carrying the sensors or on the proximity of two given sensors.

We study the imote dataset over a typical day and the mit data over a typical week. The number of edges and the average degree of those graphs are reported in Figure 4. We have carefully checked that the results obtained on these durations were similar for other periods. Both imote and mit graphs are sparse (the number of edges is low) and the number of edges and the average degree exhibit large variations during daytime.

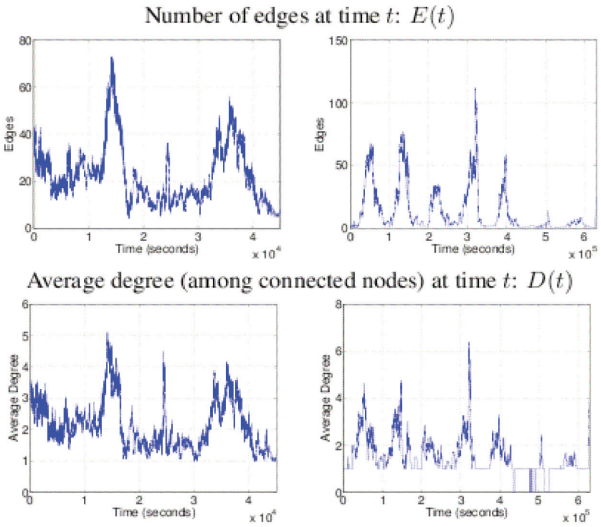
To densify the graphs and cope with the flickering edge problem that may append with experimental data, we aggregate the graphs over a period of 15 minutes for imote and 1 hour for mit: in both dynamic graphs, an edge exists if it appears at least once during the considered period. The resulting dynamic graphs have a maximum degree of 25 for imote and 22 for mit.

We extract evolving subgraphs with several density σ values, α being equal to 4.5 (average number of out-subgraph links per vertex) and the minimal size of the extracted locally maximal isolated pseudo cliques set to 4 for imote and to 3 for mit. The total runtimes and number of computed patterns are presented on Figure 5. These figures show that Evolving-Subgraphs is tractable in terms of execution time since it succeeds to extract the patterns in less than 20 minutes for different σ values varying between 1 and 0.6. The computational time is proportional to the number of output patterns what was expected according to the theoretical study of the time complexity of the pseudo clique mining algorithm. The time required

Table 1. Dataset characteristics

Dataset	Nb Edges	Nb timesteps	Avg. Density
Imote	11785	282	0.025
Mit	107770	11763	0.001
Velov	279208	930	0.003

Figure 4. Statistics of graph properties, displayed as a function of time (imote on the left and mit on the right)



to compute evolving patterns generally decreases with σ as well as the number of extracted patterns.

The numbers of evolving patterns of each type are shown on Figure 6. As the number of emergent patterns scales differently from other pattern types, their quantity is shown on the right ordinate axe, whereas the number of growth, stability and diminution patterns are plotted using the left ordinate axe. Even though the number of patterns decreases with the density threshold, we can observe that the number of each type of patterns varies irregularly.

Figure 7 shows the number of each pattern type at each time step. We can observe that the evolutions of these quantities are strongly correlated with the graph dynamic as depicted on Figure 4. The number of growth patterns is particularly correlated with the number of edges of the graph whereas the number of emergent patterns is more regular across the time.

Figure 5. Runtime and number of extracted patterns (logarithmic scales) for imote (left) and mit (right) dynamic graphs for different density threshold σ

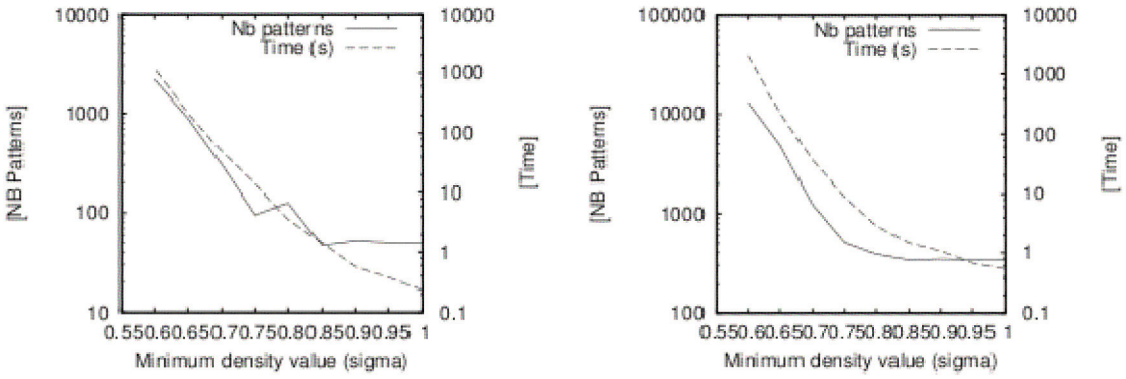


Figure 6. Number of patterns of each type for imote (left) and mit (right) dynamic graphs for different density threshold σ

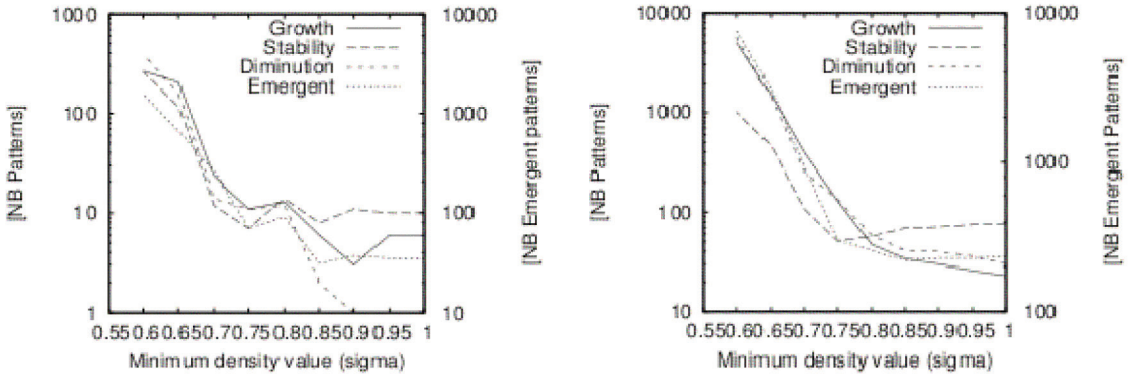


Figure 8 shows the output of our method: nodes represent valid pseudo-cliques and the numbers they contain are vertices identifiers, solid arrows show evolving patterns and dashed arrows are drawn between following subgraphs that intersect. We can identify three main groups of people. The first one is composed of individuals 9, 15, 31, 34 and 37. This group appears at time stamp 71, splits around time stamp 73 into two groups that then merge and integrate an additional vertex 5. The second group is made up of individuals 0, 4, 29 and 35. Individuals 1 and 33 are nearby. This group is stable since it remains unchanged during two consecutive time stamps. The third group contains individuals 2, 14, 19 and 25 and is also stable.

Figure 7. Number of patterns of each type at each time step for mit dynamic graph (σ=0.65)

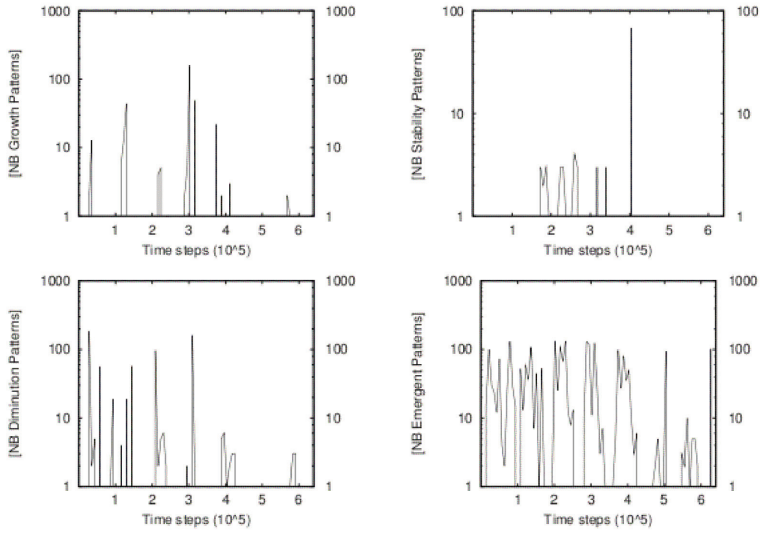
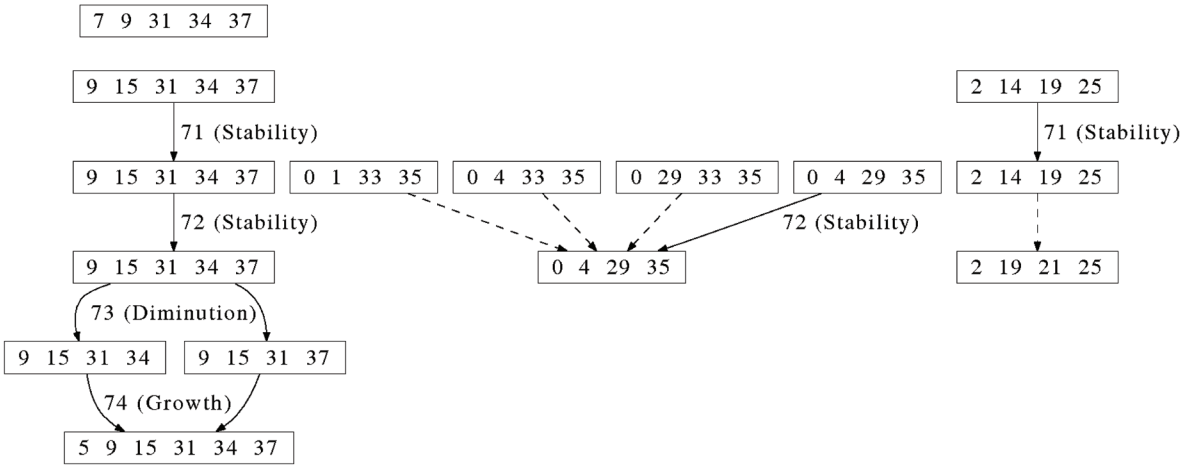


Figure 8. Display of the evolving patterns for imote with σ=0.8, α=3 and the minimum subgraph size equals 4 that occur in the morning



Shared Bicycle System VELOV

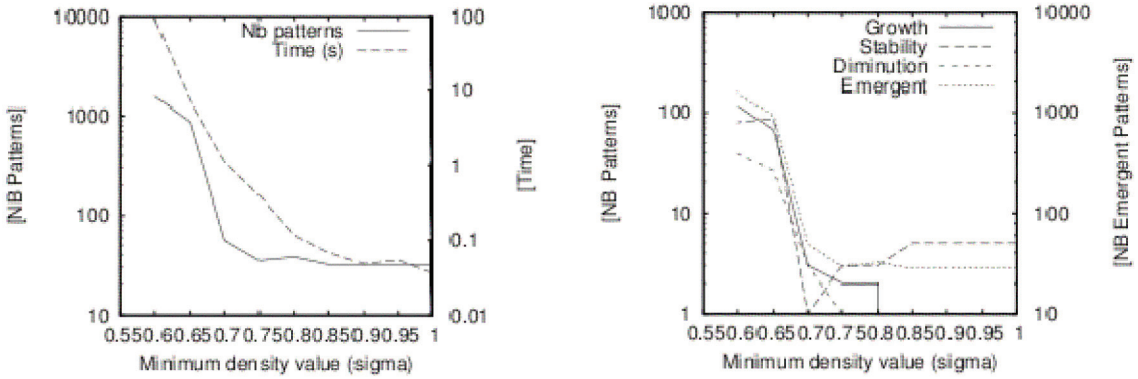
We analyze Lyon's shared bicycling system VELOV on the basis of the data provided by JCDecaux, promotor and operator of the program. The dataset contains all the bicycle trips that occurred between the 25th of May 2005 and the 12th of December 2007. Each record is anonymized and is made of the information about the date and time of the beginning of the trip, and of its end and the IDs of the departure and arrival stations (their geographical location being known). During this period, there were more than 13 million hired bicycle trips.

To analyze the velov dataset, we first aggregate the number of rentals for every days of the week and every hour over the two and a half years period of observation. We thus obtain 168 time stamps. Then to leverage the most important links, we remove the edges that had less than 50 rentals over this period.

Figure 9 (left) shows the total number of extracted evolving patterns and Evolving-Subgraphs runtime for several σ values. α being set to 5 and the minimum subgraph size is equal to 3. Here again, we can observe that the number of extracted patterns increases with σ. Figure 9 (right) shows the repartition of the patterns among the different types of evolving patterns. The majority of the extracted patterns are emergent. The number of identical patterns can increase or decrease with σ: when a stable pattern disappears, usually a growth or diminution pattern appears.

Figure 10 displays the main patterns output by Evolving-Subgraphs when applied on velov dataset for time stamp between Monday 6 PM and Tuesday 7 AM. The analysis of the output evolving patterns brings interesting pieces of information: for example, around Monday midnight, the identified patterns gather stations that are nearby to each other. Subgraph 58, 78, 115 is made of stations located on the largest campus of Lyon and shows that there are many tips between these stations. Such pattern grows at 1 AM, with the addition of a neighboring station. Stations 187, 71 and 90 are around the main Park of Lyon, also located in this area. Another important group of stations is the one made of stations 55, 84, 92 and 99 that are all located in the 7th district of the city where many student rooms are.

Figure 9. Runtime and number of extracted patterns for velov dynamic networks for different density threshold σ (left), number of patterns of each type (right)



CONCLUSION

This chapter bridges the gap between constraint-based mining techniques and dynamic graphs analysis. We have considered the evolving-pattern mining problem in dynamic graph. We introduced five new pattern types which rely on the extraction of dense subgraphs and the identification of their evolution. We formalized this task into a local-to-global framework: Local patterns are first mined in a static graph; then they are combined with the ones extracted in the previous graph to form evolving patterns. These patterns are defined by means of constraints that are used to efficiently mine the evolving patterns. Our experiments on real life datasets show that our approach produces

Figure 10. Example of interesting subgraphs for velov network

high quality patterns that are useful to understand the graph dynamics.

This technique can be of great interest for mining patterns of interactions in online communities, i.e. identifying groups of people that have strong social interactions and share some interest. Two main characteristics of our method make it a valuable tool for analysis online communities. First, whereas most of existing methods propose to identify group of interacting persons from a static point of view, here we propose to disclose how such groups emerge, attract new persons, or split, and disappear over time. This enables to analyze the temporal evolution of the online communities' structure and keep track of the changes in the interests of the communities' members. Second, the proposed method is incremental: For example, the graph of community member interactions can be updated everyday; Valid pseudo-cliques are thus extracted from it and then combined with the evolving patterns computed on the previous graphs. The global picture of the online communities is therefore maintained up to date without considering all the previous time steps (which would quickly becomes intractable) but just the previous time step graph. This is an important feature of the method which makes it usable on very long time periods.

REFERENCES

Backstrom, L., Huttenlocher, D., Kleinberg, J., & Lan, X. (2006). Group formation in large social networks: membership, growth, and evolution. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 44-54). Philadelphia, PA, USA, August 20-23. New York: ACM Press.

Bodon, F. (2005). A trie-based Apriori implementation for mining frequent item sequences. In *OSDM '05: Proceedings of the 1st International Workshop on Open Source Data Mining* (pp. 56-65). New York: ACM.

Bonchi, F., & Lucchese, C. (2007). Extending the state-of-the-art of constraint-based pattern discovery. *Data & Knowledge Engineering*, *60*(2), 377–399. doi:10.1016/j.datak.2006.02.006

Borgelt, C. (2003). Efficient implementations of Apriori and Eclat. In *1st Workshop of Frequent Item Set Mining Implementations*.

Borgwardt, K. M., Kriegel, H.-P., & Wackersreuther, P. (2006). Pattern mining in frequent dynamic subgraphs. In *Proceedings of the 6th IEEE International Conference on Data Mining (ICDM 2006), Hong Kong, China* (pp. 818-822), Washington, DC, USA. IEEE Computer Society.

Böttcher, M., Höppner, F., & Spiliopoulou, M. (2008). On exploiting the power of time in data mining. *SIGKDD Explor. Newsl.*, *10*(2), 3–11. doi:10.1145/1540276.1540278

Brandes, U., Delling, D., Gaertler, M., Görke, R., Hoefer, M., Nikoloski, Z., & Wagner, D. (2008). On modularity clustering. *IEEE Transactions on Knowledge and Data Engineering*, *20*(2), 172–188. doi:10.1109/TKDE.2007.190689

Cerf, L., Besson, J., Robardet, C., & Boulicaut, J.-F. (2008). Data-Peeler: Constraint-based Closed Pattern Mining in n-ary Relations. In *Proceedings SIAM International Conference on Data Mining (SIAM DM)* (pp. 37-48).

Chaintreau, A., Crowcroft, J., Diot, C., Gass, R., Hui, P., & Scott, J. (2005). Pocket switched networks and the consequences of human mobility in conference environments. In *WDTN '05: Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking* (pp. 244-251). New York: ACM.

Chi, Y., Zhu, S., Song, X., Tatemura, J., & Tseng, B. L. (2007). Structural and temporal analysis of the blogosphere through community factorization. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Jose, California, USA, August 12-15, 2007* (pp. 163-172). New York: ACM Press.

De Raedt, L., & Zimmermann, A. (2007). Constraint-based pattern set mining. In *Proceedings SIAM SDM'07, Minneapolis, USA*.

Eagle, N., & Pentland, A. (2006). Reality mining: Sensing complex social systems. *Journal of Personal and Ubiquitous Computing*, *10*(4), 255–268. doi:10.1007/s00779-005-0046-3

Fortunato, S., & Barthelemy, M. (2007). Resolution limit in community detection. *Proceedings of the National Academy of Sciences of the United States of America*, *104*(1), 36–41. doi:10.1073/pnas.0605965104

Hämäläinen, W., Toivonen, H., & Poroshin, V. (2004). Mining relaxed graph properties in internet. In P. T. Isaias, N. Karmakar, L. Rodrigues, & P. Barbosa (Eds.), *Proceedings of the IADIS International Conference WWW/Internet 2004, Madrid, Spain* (pp. 152-159).

Lahiri, M., & Berger-Wolf, T. Y. (2008). Mining periodic behavior in dynamic social networks. In *Proceedings of the 8th IEEE International Conference on Data Mining (ICDM 2008), December 15-19, 2008, Pisa, Italy* (pp. 373-382). IEEE Computer Society, 2008.

Leskovec, J., Kleinberg, J., & Faloutsos, C. (2005). Graphs over time: densification laws, shrinking diameters and possible explanations. In *Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Chicago, IL, USA, August 21-24, 2005* (pp. 177-187). New York: ACM Press.

Makino, K., & Uno, T. (2004). New algorithms for enumerating all maximal cliques. In *Algorithm Theory - SWAT 2004, 9th Scandinavian Workshop on Algorithm Theory, Humlebaek, Denmark, July 8-10, 2004, Proceedings* (LNCS 3111, pp. 260-272).

Morik, K., Boulicaut, J.-F., & Siebes, A. (Eds.). (2005). Local Pattern Detection. In *International Seminar, Dagstuhl Castle, Germany, April 12-16, 2004, Revised Selected Papers* (LNCS 3539).

Newman, M. E. J. (2004). Fast algorithm for detecting community structure in networks. *Physical Review E: Statistical, Nonlinear, and Soft Matter Physics*, *69*(6), 66–133. doi:10.1103/PhysRevE.69.066133

Palla, G., Derenyi, I., Farkas, I., & Vicsek, T. (2005). Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, *435*(7043), 814–818. doi:10.1038/nature03607

Pei, J., Jiang, D., & Zhang, A. (2005). On mining cross-graph quasi-cliques. In *Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Chicago, Illinois, USA, August 21-24, 2005* (pp. 228-238). New York: ACM.

Pensa, R., Robardet, C., & Boulicaut, J.-F. (2008). *Constraint-driven Co-Clustering of 0/1 Data* (pp. 123–144). CRC Press.

Robardet, C. (2009). Constraint-based Pattern Mining in Dynamic Graphs. In S. Ranka & P.S. Yu (Eds.), *Proceedings of the IEEE International Conference on Data Mining* (pp. 950-955).

Scherrer, A., Borgnat, P., Fleury, E., Guillaume, J.-L., & Robardet, C. (2008). Description and simulation of dynamic mobility networks. *Computer Networks*, *52*(15), 2842–2858. doi:10.1016/j.comnet.2008.06.007

Sun, J., Papadimitriou, S., Yu, P. S., & Faloutsos, C. (2007). Graphscope: Parameter-free mining of large time-evolving graphs. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Jose, California, USA, August 12-15, 2007* (pp. 687-696), San Jose, CA, USA.

Tantipathananandh, C., Berger-Wolf, T. Y., & Kempe, D. (2007). A framework for community identification in dynamic social networks. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Jose, California, USA, August 12-15, 2007* (pp. 717-726). ACM.

Tong, H., Papadimitriou, S., Sun, J., Yu, P. S., & Faloutsos, C. (2008). Colibri: fast mining of large static and dynamic graphs. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, Nevada, USA, August 24-27, 2008* (pp. 686-694). New York: ACM.

Uno, T. (2007). An efficient algorithm for enumerating pseudo cliques. In Algorithms and Computation. In *Proceedings of 18th International Symposium, ISAAC 2007, Sendai, Japan, December 17-19, 2007* (LNCS 4835, pp. 402-414).

Zhu, F., Yan, X., Han, J., & Yu, P. S. (2007). GPrune: A constraint pushing framework for graph pattern mining. In Z.-H. Zhou, H. Li, & Q. Yang (Eds.), *Advances in Knowledge Discovery and Data Mining, 11th Pacific-Asia Conference, PAKDD, Nanjing, China, May 22-25, Proceedings* (LNCS 4426, pp. 388-400).