Contributions to Pattern Mining in Augmented Graphs

Marc Plantevit

UNIVERSITÉ DE LYON

École Doctorale InfoMaths de Lyon

Contributions to Pattern Mining in Augmented Graphs

THÈSE

soutenue le 14 Décembre 2018

pour l'obtention d'une

Habilitation de l'Université Claude Bernard Lyon 1

(mention informatique)

par

Marc Plantevit

Composition du jury

Rapporteurs :	Pr. Toon Calders	University of Antwerp
	Dr. Florent Masseglia	Inria Sophia Antipolis - Méditerranée
	Pr. Dino Pedreschi	University of Pisa
Examinateurs :	Pr. Angela Bonifati	Université Claude Bernard Lyon 1
	Pr. Bruno Crémilleux	Université de Caen Normandie
	Pr. Eric Gaussier	Université Grenoble Alpes
	Pr. Céline Robardet	INSA Lyon
	Pr. Alexandre Termier	Université de Rennes 1

Laboratoire d'InfoRmatique en Image et Systèmes d'information (LIRIS) – UMR 5205

Mis en page avec la classe thesul.

Remerciements

Je tiens tout d'abord à remercier les membres de mon jury d'habilitation : Toon Calders, Dino Pedreschi, Florent Masseglia, Angela Bonifati, Bruno Crémilleux, Eric Gaussier, Céline Robardet et Alexandre Termier. Je suis très honoré de votre présence dans ce jury. Une mention particulière aux trois premiers qui, malgré leur emploi du temps très chargé, ont accepté de se plonger dans ce manuscrit et de rédiger un rapport.

Les travaux réunis dans ce mémoire couvre dix années et sont le fruit d'un travail d'équipe. Je tiens à exprimer ma gratitude à toutes les personnes qui m'ont donné la chance et le plaisir de travailler avec elles: les étudiants chercheurs – Elise, Ngan, Gunce, Maëlle, Adnene, Aimene, Anes, Corentin, Mohamed – qui m'ont beaucoup apporté sur de nombreux plans et n'ont cessé de m'impressionner; les post-doctorants – Adriana, Albrecht et Yoann – dont les passages furent trop brefs; les plus expérimentés – Mehdi, Rémy, Céline et Jean-François – qui font que malgré tous les aléas de la vie d'un enseignant chercheur, je garde au quotidien une passion intacte pour cette fonction grâce à des échanges passionnés; les collègues qui ont participé aux différents projets que j'ai pu conduire, chacun sait à quel point sa contribution aura été importante.

Je n'oublie pas les personnes qui m'ont fait découvrir la recherche. Anne, Maguelonne, Pascal, c'est avec vous que tout a commencé lorsqu'avec Chedy nous cherchions des encadrants pour un sujet de TER. Puis, il y a Bruno et Jean-François qui ont pris le relais en me prodiguant de nombreux conseils éclairés et m'évitant de nombreuses erreurs. Bruno tu es aussi le seul grâce à ton organisation sans faille à pouvoir exhumer certaines de nos discussions datant de plusieurs années et me demander d'approfondir certains de ces points. Bref, vous êtes des exemples et j'espère pouvoir marcher dans vos pas.

Je remercie également les personnes qui ont su donner à ces dix dernières années un parfum de convivialité et sérénité à savoir les habitués des pauses café et les membres de l'équipe administrative et technique qui ont débloqué de nombreuses situations.

Enfin, je ne peux pas terminer sans citer Laure, Quentin, Maxime et Thibaud dont la vie est un peu bousculée par mon travail. Merci d'être si patients et conciliants.

Contents

Preface

Writing a *Habilitation à Diriger des Recherches* thesis is probably for most of us an exercise that is much more delicate and disturbing than writting conference or journal papers for which the standards are known. It mainly consists in highlighting a logical sequence on the work of these first years of research post PhD (July 2008 in my case). Many options are allowed including an extended curriculum vitæ, an as exhaustive as possible document depicting the obtained results, or a synthesis on an emerging field of research. After many hesitations on the form, I finally decided to write a document that will - I hope - be pleasant for the reader. I wanted this document not too technical to not lose the essence of the pattern mining research that is the discovery of patterns of interest. Of course, the algorithms are the core business of pattern miners since the explored search space are exponential by nature which requires the definition of efficient and effective algorithms. However, I will briefly describe the algorithms in this manuscript except those that are under review. Similarly, I will not report quantitative studies in the experiments preferring focusing on the pattern definitions and intuitions. Hence, each type of pattern considered in this manuscript is illustrated on a real-world dataset.

Another dilemma that I had to deal with is the use of 'I' or 'we'. Obviously, a *Habilitation à Diriger des Recherches* thesis is an individual diploma that assesses the research activities after the PhD. However, I cannot conceive doing Research alone. This is not my vision of the scientific Research which rhymes – for me– with sharing, listening, fruitful exchanges, long discussions in front of a blackboard. This therefore cannot be achieved alone. This is especially true when one tries to apply one's work in other disciplines which requires thorough discussion with the data owner (i.e., neuroscientists, biologists). This is why I will use, in this manuscript, "we" instead of "I". Most work presented in this thesis was done in collaboration with other enthusiastic colleagues.

Chapter 1

Introduction

I obtained my PhD degree in 2008 from Université de Montpellier, on the topic of multidimensional sequential pattern mining under the supervision of Maguelonne Teisseire and Anne Laurent. After that, I did a one year post-doc in the team of Bruno Crémilleux at Université de Caen Basse-Normandie during which I devised sequence mining methods and applied them for named entity recognition and relations discovery in bio-medical texts.

Since I joined the LIRIS laboratory in 2009 as an Associate Professor, I have been working on constraint-based pattern mining. My interest has particularly focused on the definition of new pattern languages and their related primitives as well as the consideration of domain knowledge, user's priors and preferences into the definition of new algorithms to discover patterns that could produce knowledge for the users of the algorithms. For a decade, I have been fortunate to collaborate with scientists from various disciplines (e.g., Biology, Neurosciences, Material Engineering, Social Sciences, Geology, Chemistry) who bring me new applications and data mining challenges.

Most of my research – and more generally, research in pattern mining – can be easily summarized from an Inductive Database perspective [Imielinski and Mannila, 1996] as the computation of a theory Th defined as:

$$Th(\mathscr{L},\mathscr{D},\mathscr{C}) = \{\psi \in \mathscr{L} \,|\, \mathscr{C}(\psi,\mathscr{D}) \text{ is true}\}.$$

Given a pattern language $\mathscr{L}$, some constraints $\mathscr{C}$ and a database $\mathscr{D}$, a pattern mining algorithm aims at enumerating the elements of the language that fulfill the constraints within the data. In practice, the user defines her interest in a declarative way and does not specify how to compute the solution. This is an elegant way to define pattern mining. Notice that this is not just a theoretical abstraction and that there have been Inductive Database prototypes proposed in the literature, e.g., the mining view system [Blockeel *et al.*, 2012]. Of course, many variants are possible. For instance, the pattern mining algorithms can be required to be complete (as in the above formalization) or they can look for a subset of all the patterns (top k, representative samples) for which the constraints hold. Another example deals with the constraint part. The user is supposed to express her interest in term of constraints (maximal patterns, threshold-based constraints) so that the pattern mining task is a satisfaction problem. Expressing her interest with constraint is difficult for the user, alternatively she can emit explicit preferences or have implicit preferences that have to be acquired within the mining process. In this case, the pattern mining task turns out to be an optimization problem.

From this formalization, we can highlight what a pattern mining researcher can do. That is, devising algorithms that mine **faster** and **better**:

Faster: Pattern miners aim to define algorithms that compute the solutions as quickly as possible or simply make the extraction feasible. The size of the dataset is a factor that pattern miners have to

take into account but this is not the main factor that may jeopardize the scalability of the algorithms. Indeed, they have to handle search spaces of exponential size. Therefore, pattern miners keep defining new pruning properties as well as data structures to make the extractions faster or simply feasible in some cases.

Better: Keep in mind that there may be many users behind each dataset and they may have different interests as well as different background knowledge. Pattern miners aim at providing algorithms to automatically discover novel insights about the domain in which the data was measured and therefore boost the human expertise. Expressed like this, the task seems simple but it is incredibly complicated with many research challenges. A good pattern mining algorithm must simultaneously handle the complexity of the data (e.g., graphs, sequences, numerical attributes), the complexity of the domain and the complexity of the user to provide high quality outputs.

Pattern mining researchers may contribute on one or several elements of the triptych data, pattern language and constraints (or more generally user's interest). They may focus on the definition of efficient algorithms for a well know pattern mining task (e.g., frequent itemsets) to mine faster or design new pattern domains (i.e., pattern language and some related primitive constraints) while taking into account some domain knowledge to mine better.

From a pattern miner point of view, mine better and mine faster are now naturally indissociable, especially in approaches that aim to learn the user interest through interactive loops for which instant mining is required (i.e., instantaneously provide patterns to user without waiting time to maintain the interaction loop) [Rueping, 2009, Bhuiyan *et al.*, 2012, van Leeuwen, 2014, Dzyuba *et al.*, 2014]. However, this has not always been the case. For instance, much effort was devoted – with dozens of algorithms proposed – to earn milliseconds in frequent itemset mining without asking if the collection of patterns produced was really interesting for the end-user. The very first bursts toward a better quality of the pattern mining results were the studies to limit the redundancy of the frequent patterns with the condensed representations of frequent patterns [Calders *et al.*, 2004, Bastide *et al.*, 2000, Calders and Goethals, 2002, Boulicaut *et al.*, 2003]. Such condensed representations allow to obtain smaller collections (but still too large to be Human manageable) while introducing additional pruning properties and thus to mine faster. Since the introduction of condensed representations, the pattern mining research has slowly but surely evolved towards an increasing consideration of the end-user. In the first years of pattern mining, researchers focused on only frequent patterns, then constraint-based pattern mining appeared to offer the possibility to the user to express her interest by means of constraints. A large attention has been given to the study of the constraint properties (anti-monotonicity [Ng *et al.*, 1998], monotonicity [Bonchi and Lucchese, 2007, Bonchi *et al.*, 2005], convertible [Pei *et al.*, 2004], primitive-based or piecewise constraints [Soulet and Crémilleux, 2009, Cerf *et al.*, 2009, Buzmakov *et al.*, 2015]), and how to efficiently push them. It gives some strong foundations to pattern mining that allow the consideration of more sophisticated tasks.

As a result of more than two decades of research in pattern mining, several pattern domains to address various types of data (e.g., itemsets, sequences, graphs, images, n-ary relations) have been investigated and primitives more sophisticated than the simple frequency have been proposed (e.g., density in graph, time-constraints in sequences, information content). On the shelf, many algorithms are now available ranging from *ad-hoc* algorithms (only one constraint is handled, e.g. frequency) for frequent itemsets [Bayardo *et al.*, 2005, Goethals and Zaki, 2004], frequent sub-sequences [Masseglia *et al.*, 1998, Zaki, 2001, Yan *et al.*, 2003, Wang *et al.*, 2007], frequent subgraphs [Yan and Han, 2002], [Nijssen and Kok, 2004] and dense subgraphs [Liu and Wong, 2008, Uno, 2007, Jiang and Pei, 2009] to more generic algorithms (i.e., available to handle any – or several – constraints) as [Négrevergne *et al.*, 2014, Khiari *et al.*, 2010, Guns *et al.*, 2017] in itemsets, [Négrevergne and Guns, 2015] in sequences and [Cerf *et al.*, 2013] in n-ary relations. Researchers have striven to enhance the quality of the output of pattern mining algorithm. This involves the definition of new patterns as mentioned just before but also by new methods to assess the

patterns [Gionis *et al.*, 2007, Webb and Petitjean, 2016, Petitjean *et al.*, 2016, Low-Kam *et al.*, 2013], to mine pattern sets [Raedt and Zimmermann, 2007] with global constraints or by compression [Vreeken *et al.*, 2011], by taking into account domain or user knowledge [Anand *et al.*, 1995, Bie, 2011b] or by learning the user interest [Rueping, 2009, Bhuiyan *et al.*, 2012, van Leeuwen, 2014, Dzyuba *et al.*, 2014]. Some pattern mining tasks were also defined in supervised settings to characterize or co-characterize set of objects like emerging patterns [Dong and Li, 1999], subgroup discovery [Klosgen, 1996, Wrobel, 1997], exceptional model mining [Leman *et al.*, 2008] and redescription mining [Galbrun and Miettinen, 2017].

Following this line of research, I have made contributions in sequence mining striving to handle the inherent multidimensionality of the data [Plantevit *et al.*, 2010], extending some results from itemset mining to sequence mining [Plantevit and Crémilleux, 2009, Holat *et al.*, 2014] or applying sequence mining algorithms in biomedical texts [Cellier *et al.*, 2015] or in data streams [Raïssi and Plantevit, 2008]. I have been interested in the definition of pattern domain in rating or vote data [Belfodil *et al.*, 2017]. I have also worked to make constraint-based pattern mining algorithm easier to use for non data miner, for instance by using (explicit) user preferences to overcome the thresholding problem with the so-called skypatterns [Soulet *et al.*, 2011, Ugarte *et al.*, 2017]. I do not details these contributions in this manuscript.

In this *Habilitation à Diriger des Recherches* thesis, I present some contributions on pattern mining in augmented graphs I obtained with colleagues. **Why augmented graphs?** During the first years as a young researcher, I was interested in multidimensional sequence mining. Even if I defined efficient algorithms, the interpretation of the multidimensional sequential patterns was very difficult for the users. That is why I decided to investigate pattern domains that are easier to assimilate for the end-users. In this sense, graphs are a formidable mathematical tool that enables to describe some complex phenomena and, surprisingly they are very intuitive for any users. Accordingly, I have investigated pattern mining in augmented graphs since 2009. Augmented graphs are plain graphs that are enriched with additional pieces of information. For instance, one can add a temporal dimension to depict the appearance or disappearance of some edges or vertices, such time augmented graphs are known under the vocable *dynamic graphs* or *time evolving graphs*. Graphs whose vertices (respectively edges) are augmented with attributes describing them are called vertex (resp. edge) attributed graphs. Notice that the graphs considered in this manuscript are *relational*, i.e., the vertices are uniquely identified which allows to cope with the subgraph isomorphism problem when searching the occurrences of a pattern within the graph.

Contributions

This manuscript only contains my work related to augmented graph mining. In such graphs, I have contributed to define new pattern languages, primitives and the associated mining algorithms. I have striven to continually improve the quality of the patterns reported to the user, this requires the consideration of domain knowledge, user's prior knowledge and user feedback. The contributions of this thesis are the following:

- The definition of (hierarchical) co-evolution patterns to analyze dynamic attributed graphs as well as interestingness measures to assess these patterns according to each dimension of the dynamic attributed graphs (i.e., the dynamics, the attributes – organized within a hierarchy or not – and the vertices). Examples of patterns from spatio-temporal data are reported (landslide detection from satellite images, US domestic air flight analysis).
- The definition of pattern domains to elucidate links between the graph structure and the vertex attributes in static or dynamic attributed graphs. Such patterns are illustrated on co-authorship network built from DBLP[1].

[1] `https://dblp.org/`

- The discovery of exceptional subgraphs in edge-attributed or vertex-attributed graphs rooted in Subgroup Discovery / Exceptional Model Mining. The patterns proposed here allow to depict subgraphs whose vertices or edges have attributes values that are very different from the rest of the graph. Qualitative experiments are reported and demonstrate the interest of these pattern to depict urban areas and to analyze bicycle sharing system network.
- The consideration of domain knowledge into the discovery of exceptional attributed subgraphs. We borrow some mobility models from statistical physics to assess the unexpectedness of some trajectories in mobility networks and thus to better take into account the spatial information (e.g., the distance between vertices, the importance of a vertex in term of population).
- A method to take into account the user feedback into biased quality measures. This method is defined in the context of social media analysis, especially geo-located event detection on Twitter. Geo-located events that involve tags or areas of interest for the user (considering her feedback) are promoted in this approach.
- The subjective interestingness in attributed graphs or how to take into account user's prior knowledge in the discovery of exceptional subgraphs. The subjective interestingness framework [Bie, 2011b] is applied on the discovery of exceptional attributed graphs. In this work, a particular interest is given to the pattern assimilation for the user. To this end, alternative descriptions of exceptional (w.r.t. user's priors) subgraphs easier to assimilate are provided.

Roadmap

The manuscript is structured in two parts. The first part of this manuscript is related to the definition of different types of patterns for analyzing augmented graphs. It contains three chapters: the discovery of trends (co-evolution patterns) in dynamic attributed graphs (Chapter 2), Looking for links between structure and vertex attributes (Chapter 3) and Exceptional Model Mining in attributed graphs (Chapter 4). The second part focuses on finding patterns of higher interest by taking into account domain knowledge (Chapter 5), user feedback (Chapter 6) and user's prior knowledge (Chapter 7). Finally, I conclude and give some research perspectives I would like to address in future years in Chapter 8.

Note that the chapters correspond to independent research articles with little modifications, there might thus be some redundancy in definitions.

Part I

New Pattern Domains and Primitives for Augmented Graphs

Chapter 2

Mining Trends in Dynamic Attributed Graphs

Contents

2.1 Introduction

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 required since the beginning of the 21th century. Such graphs are naturally augmented with their dynamics (i.e., they evolve through time) and with additional pieces of information (i.e., vertices are depicted by attributes). The investigation of much more complex data than the classical itemsets was made possible by the fact that the data mining techniques has become sufficiently mature since the mid-2000s (i.e., efficient and effective pruning strategies). For theses reasons, the analysis of augmented graphs has become a timely challenge for pattern mining researchers.

Our first contribution in this domain was the redefinition of rules and related confidence measures in dynamic graphs [Nguyen *et al.*, 2011, Nguyen *et al.*, 2013]. This is the PhD work of Kim-Ngan Nguyen. Five years later, the semantics conveyed by such rules is still difficult to apprehend and needs to be deeply investigated to be fully usable in practice. With Pierre-Nicolas Mougel and Christophe Rigotti, we also defined an approach to discover maximal homogeneous clique sets, a new kind of pattern for graphs with feature vectors [Mougel *et al.*, 2014]. These two contributions are quite representative and symptomatic of the pattern mining research in attributed graphs at that time: dynamic graphs [Berlingerio *et al.*, 2009, Borgwardt *et al.*, 2006, Robardet, 2009] and attributed graphs [Moser *et al.*, 2009, Silva *et al.*, 2012] were separately considered. In this regard, our major contribution to the field of augmented graph mining is the extraction of valuable information in dynamic attributed graphs, i.e., the simultaneous consideration of the graph structure, the vertex attributes and their evolution through time. This makes it possible to describe the evolution of some particular subgraphs.

In this chapter, we present the main results from Elise Desmier's PhD thesis [Desmier *et al.*, 2012, Desmier *et al.*, 2013, Desmier *et al.*, 2014].

- We first present the co-evolution patterns, the pattern domain we defined to support the analysis of dynamic attributed graphs.
- We present the interestingness measures we introduced to assess the interest of such patterns with regard to their different dimensions (i.e., the dynamics, the attributes and the vertices).
- Next, we discuss the consideration of hierarchical relations on the vertex attributes. Taking into account these hierarchies leads to the definition of both a new type of co-evolution patterns and new dedicated interestingness measures.

2.2 Co-Evolution Patterns

Dynamic attributed graphs allow to easily depict real phenomena where entities have particular characteristics and relations between each other. Entities are represented by vertices, relations between entities are described by edges and entities characteristics by attribute values.

Definition 2.1 (Dynamic Attributed Graph)**.** *A dynamic attributed graph $\mathcal{G} = (\mathcal{V}, \mathcal{T}, \mathcal{A})$ is a sequence over a time period $\mathcal{T}$ of attributed graphs $\langle G_1, \cdots, G_{|\mathcal{T}|} \rangle$ where $\mathcal{A}$ is the set of attributes and each attributed graph G_t is a triplet $(\mathcal{V}, E_t, A_t)$, where:*

- *$\mathcal{V}$ is a set of vertices that is fixed throughout the time.*
- *$E_t \in \mathcal{V} \times \mathcal{V}$ is a set of edges at time t.*
- *A_t is a vector of numerical values for the attributes of $\mathcal{A}$ that depends on t.*

An example of dynamic attributed graph is provided in Figure 2.1 with $\mathcal{V} = \{v_1, v_2, v_3, v_4, v_5\}$, $\mathcal{T} = \{t_1, t_2, t_3\}$ and $\mathcal{A} = \{a_1, a_2, a_3\}$. For instance, $G_{t_1}(v_1, a_1) = 2$.

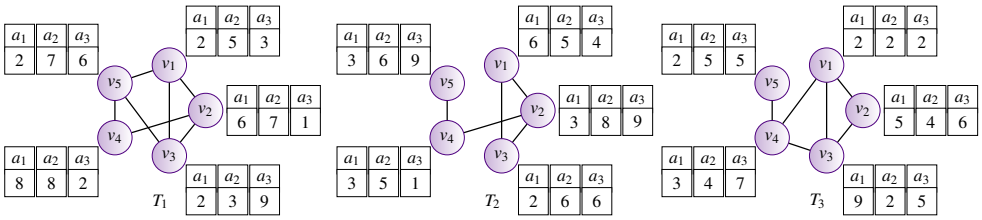


Figure 2.1: Toy example of a dynamic attributed graph.

The patterns introduced in this chapter aim to highlight trends on the attributes, i.e., evolutions of the attribute values between two consecutive timestamps. Therefore, vertex attributes must be numerical or ordinal, i.e., values of the attribute domains must be ordered. Given an attribute and a vertex, the trend (i.e., increase, decrease, constancy) is defined by the order between the values observed at two consecutive timestamps for the attribute of the vertex. Hence, a preprocessing step may produce an alternative representation of a dynamic attributed graph focusing on the attribute trends instead of the values. For instance, Figure 2.2 is the alternative representation of Figure 2.1. Notice that timestamp t_i in this new representation depicts the interval $[t_i, t_{i+1}]$ in the original dynamic graph.

Intuitively, patterns of interest in such a graph are set of vertices which are closely related through the graph structure and which follow the same trends over a set of attributes over time. Co-evolution patterns were introduced to uncover such behavior. Given a dynamic attributed graph $\mathcal{G}$, a co-evolution pattern $P = (V, T, \Omega)$ is a subset of $\mathcal{V}$, a subset of $\mathcal{T}$ and a subset of signed attributes (to express trends) from

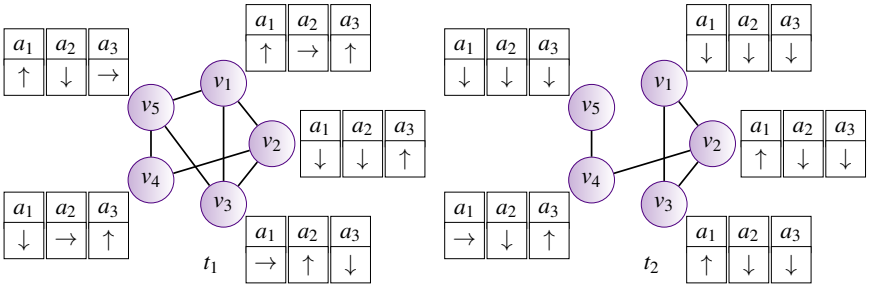


Figure 2.2: Toy example: from values (Fig. 2.1 to trends.

$\mathscr{A}$ which respect two constraints, one on the trend of the attribute values and one on the structure of the graph.

As an example, on the DBLP dataset, a pattern is a set of authors that have similar co-authors and that have increased or decreased their number of publications in a subset of conferences or journals during a (possibly discontinuous) time period. We define two predicates:

- $related(P) = true$ denotes the fact that the structural relation is respected for all timestamps, i.e., $\forall t \in T$ then $related(V, E_t) = true$
- $evol(P) = true$ denotes the fact that the trend is respected for all triplets, i.e., $\forall v \in V$, $t \in T$ and $a^s \in \Omega$ the $evol(v,t,a^s) = true$ with S a trend associated to each attribute. In this manuscript $S \in \{+,-\}$ meaning respectfully a $\{increase, decrease\}$ trend.

The constraint $evol(v,t,a^s) = true$ can be any constraint on the evolution of the attribute values[2] and the constraint $related(V,E_t) = true$ can be any constraint to depict if some vertices are closely related or not. Set of vertices can be considered as related is they share a similar neighborhood or if they play a similar role within the graphs [Desmier *et al.*, 2012], or simply if the diameter of the induced subgraph is lower than a threshold [Desmier *et al.*, 2013]. Formally, a co-evolution pattern is:

Definition 2.2 (Co-evolution Pattern)**.** *Given a dynamic attributed graph $\mathscr{G} = (\mathscr{V}, \mathscr{T}, \mathscr{A})$, a co-evolution pattern, is a triplet $P = (V, T, \Omega)$:*

- *$V \subseteq \mathscr{V}$ is a subset of the vertices of the graph.*
- *$T \subset \mathscr{T}$ is a subset of not necessarily consecutive timestamps.*
- *Ω is a set of signed attributes, i.e., $\Omega \subseteq A \times S$ with $A \subseteq \mathscr{A}$.*

The two following conditions must hold:

1. *Each signed attribute a^s with $a \in A$ and $s \in \{+,-\}$ defines a trend that has to be satisfied by any vertex $v \in V$ at any timestamp $t \in T$: $evol(P) = true$*
2. *At each time $t \in T$, the vertices of the pattern have to be closely related through the structure E_t of the graph: $related(P) = true$.*

Additional constraints (e.g., maximality, minimum set size, volume) can be considered to make the extraction feasible. The study of the properties of each constraint make it possible to devise a complete algorithm called `MINTAG` that efficiently push these constraints.

[2]Given the application, we can consider significant increases or decreases.

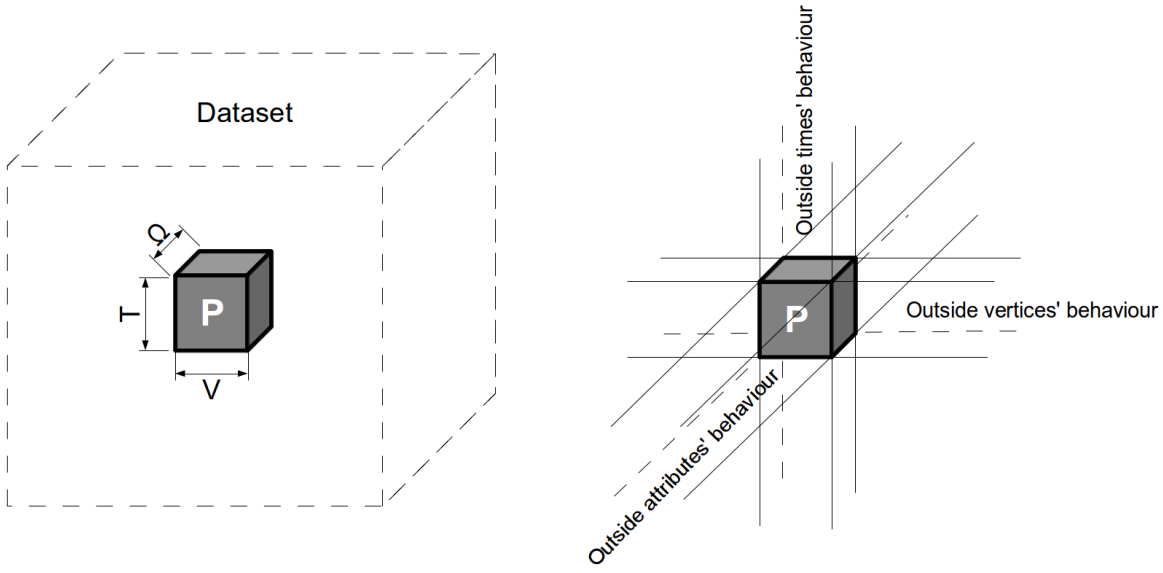


Figure 2.3: Outside densities overview: each measure assesses the specificity of the pattern with regard to one dimension (i.e., the two others are fixed).

2.3 Outside Densities Based Interestingness Measures for Co-Evolution Patterns

The constraints previously mentioned are not enough to obtain interesting results. Especially, they do not allow to discard redundancy among the collection of patterns as well as patterns that are obvious. For instance, on a co-author network whose authors are depicted by their venues (the conferences and the journals they have published in), `MINTAG` provides some patterns that identify set of authors for which we observed a decrease of publication at VLDB and an increase of publication at PVLDB. Such results are obvious since it corresponds to the change of publication policy of the VDLB endowment. Other obvious results could be the identification of individuals who become older through time.

To discard such obvious patterns, we introduced measures that aim to assess the specificity of the a pattern according to its 3 dimensions (i.e., the vertices, the attributes trends, and the time). Given a dynamic attributed graph $\mathscr{G} = (\mathscr{V}, \mathscr{T}, \mathscr{A})$, the idea is to discard patterns that depict a behavior which is not specific and that is supported by many other elements of the graph. Figure 2.3 illustrates a pattern as a cube in the dataset and highlights the outside elements that have to be considered for each measure. Given a co-evolution pattern $P = (V, T, \Omega)$, 3 measures of "outside densities" are proposed to evaluate the behavior of outside vertices, outside times and outside attributes on the elements of the pattern. The proposed measures aim at assessing how the pattern is isolated within the data and at providing insights to the following questions:

- **Vertex specificity:** How similar are the vertices outside the trend dynamic sub-graph to the ones inside it?
- **Temporal dynamic:** What about the dynamics of the pattern? Does it appear suddenly?
- **Trend relevancy:** Do the vertices of the pattern co-evolve only on the attributes of the pattern?

More precisely, the measure of vertex specificity evaluate how the trends of Ω at times of T are followed by outside vertices $\mathscr{V} \setminus V$. The measure of temporal dynamic checks if the behavior of vertices of V for attributes of Ω is really specific to the timestamps of T. And the measure of trend relevancy

computes how similar is the behavior of vertices of V at timestamps of T for outside attributes $\mathscr{A} \setminus A$.

Vertex specificity: With this measure, the aim is to quantify the average proportion of trends that are satisfied by the vertices that do not belong to the pattern. Let us consider Figure 2.3 as a representation of the pattern within the dynamic attributed graph by a a three-dimension matrix where there is a 1 if $trend(v,t,a^s)$ is respected and 0 otherwise.

Definition 2.3 (Vertex Specificity (κ))**.** *Given $\delta_{condition}$ the Kronecker function that is equal to 1 if condition is satisfied, or 0 otherwise, $P = (V,T,\Omega)$ a pattern and $\kappa \in [0,1]$ a user defined threshold:*

$$\begin{aligned} vertexSpecificity(P) &= \frac{\sum_{v \in \mathscr{V} \setminus V} \sum_{t \in T} \sum_{a^s \in \Omega} \delta_{a^s(v,t)}}{|\mathscr{V} \setminus V| \times |T| \times |\Omega|} \\ specificityV(P,\kappa) &= true \Leftrightarrow vertexSpecificity(P) \leq \kappa \end{aligned} \tag{2.1}$$

The more the behavior depicted by the timestamps and the attribute trends of the pattern is specific to the vertices of the pattern, the lower this measure. For instance, in a co-author network, lots of patterns are extracted whose authors have a decreasing number of publications in the VLDB conference and an increasing number of publications in PVLDB between 2002 and 2010. This is due to the new policy of the "VLDB endowment". Such patterns will have a relatively high value of vertex specificity and thus not considered as interesting according to this measure. Notice that a pattern that describes author with an increasing number of publications in a conference that increased its selection criterion will have a small vertex specificity value.

Given $\mathscr{G}$ the graph in Figure 2.2 (page 9), and $\kappa = 0.25$, Figure 2.4 represents the elements considered to compute vertex specificity. The colored vertices are the outside vertices, the dark colored attributes are those that respect the trends and the light colored attributes are those that do not.

- Given $P = \{(v_1,v_2,v_3),(t_2),(a_2^-,a_3^-)\}$, $vertexSpecificity(P) = \frac{3}{4} = 0.75$, then $specificityV(P,\kappa)$ is false, P is not a valid pattern (not enough specific according the outside vertices).
- Given $Q = \{(v_2,v_4),(t_1),(a_1^-,a_3^+)\}$, $vertexSpecificity(Q) = \frac{1}{6} = 0.17$, then $specificityV(Q,\kappa)$ is true, Q is a valid pattern.

Temporal dynamics: The aim of this measure is to evaluate the dynamics of the proportion of vertices and attributes that satisfy the pattern before, between and after the timestamps of T. Similarly to vertex specificity, this measure studies the outside times' behavior. However, contrary to the vertex specificity, this measure does not evaluate the global proportion but how the behavior appear in time, i.e., it considers the outside time where the number of "1" compared to the number of possible "outside vertices triset" is the highest. Considering individuals depicted by their age, it is obvious that their age do not increase only at the timestamps of the pattern. On DBLP, this measure will check if there is at least one other timestamp outside the patten for which the authors follow the trends on the attributes.

Definition 2.4 (Temporal Dynamics (τ))**.** *Given $\delta_{condition}$ the Kronecker function, $P = (V,T,\Omega)$ a pattern and $\tau \in [0,1]$ a user defined threshold:*

$$\begin{aligned} temporalDynamic(P) &= \max_{t \in \mathscr{T} \setminus T} \frac{\sum_{v \in V} \sum_{a^s \in \Omega} \delta_{a^s(v,t)}}{|V|.|\Omega|} \\ dynamicT(P,\tau) &= true \Leftrightarrow temporalDynamic(P) \leq \tau \end{aligned} \tag{2.2}$$

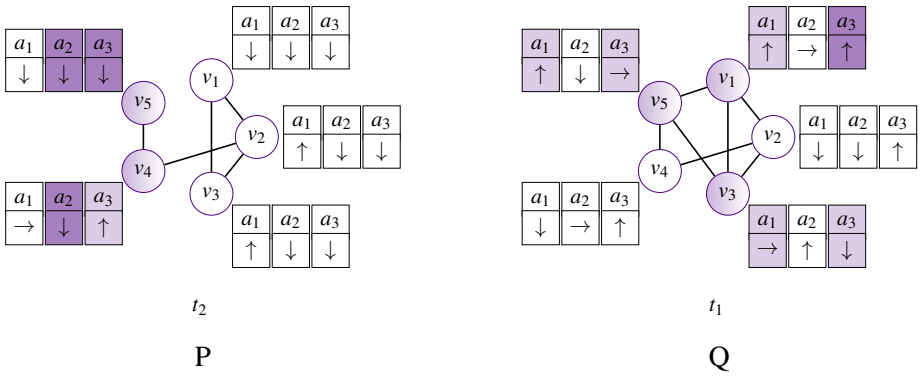


Figure 2.4: Vertex specificity for patterns P and Q from the toy example of Figure 2.3. (left) $P = \{(v_1, v_2, v_3), (t_2), (a_2^-, a_3^-)\}$, $vertexSpecificity(P) = \frac{3}{4} = 0.75$, then $specificityV(P, \kappa)$ is false, P is not a valid pattern (not enough specific according the outside vertices). (right) $Q = \{(v_2, v_4), (t_1), (a_1^-, a_3^+)\}$, $vertexSpecificity(Q) = \frac{1}{6} = 0.17$, then $specificityV(Q, \kappa)$ is true, Q is a valid pattern.

The more the co-evolution pattern bursts, the lower this measure. For instance, on "DBLP", if the trends of the pattern have ever been followed by 80% of the authors at the previous timestamp, the value of temporal dynamics will be high, indeed the pattern does not burst and this behaviour is not specific to these timestamps. On the contrary, a pattern that describes a complete change of publication policy of a group of authors will have a small temporal dynamic value.

Given $\mathcal{G}$ the graph in Figure 2.2 (page 9), and $\tau = 0.25$, Figure 2.5 represents the elements considered to compute the temporal dynamics. The colored vertices are the vertices of the pattern at timestamps outside the pattern, the dark colored attributes are the one that respect the trends and the light colored attributes are the ones that do not.

- Given $P = \{(v_1, v_2, v_3), (t_2), (a_2^-, a_3^-)\}$, $temporalDynamic(P, \tau) = \frac{2}{6} = 0.33$, then, $dynamicT(P)$ is false, P is not a valid pattern.
- Given $Q = \{(v_2, v_4), (t_1), (a_1^-, a_3^+)\}$, $temporalDynamic(Q, \tau) = \frac{1}{4} = 0.25$, then, $dynamicT(Q)$ is true, Q is a valid pattern.

Trend relevancy: The aim of this measure is to evaluate the entropy of the outside attribute trends and consider the one that has the smallest entropy. It is interesting to evaluate if the vertices of the pattern follow coherent trends on outside attributes, i.e., if they follow the same trends on the same attributes at the same times. To this aim, a measure of entropy is used, commonly understood as a measure of disorder or a measure of the uncertainty in a random variable. The Shannon entropy of a finite sample is written $H(X) = -\sum_i P(x_i) \times log_b(P(x_i))$ [Borda, 2011]. Illustrating it on DBLP, the aim of this measure is to evaluate how similar is the publication policy of the authors at these timestamps in other conferences.

Definition 2.5 (Trend Relevancy (ρ))**.** *Given $\delta_{condition}$ the Kronecker function, $P = (V, T, \Omega)$ a pattern and $\rho \in [0, 1]$ a user defined threshold:*

$$
\begin{aligned}
E(a^s, V, T) &= \frac{\sum_{v \in V} \sum_{t \in T} \delta_{a^s(v,t)}}{\sum_{v \in V} \sum_{t \in T} \left(\delta_{a^-(v,t)} + \delta_{a^+(v,t)}\right)} \\
trendRelevancy(P) &= \min_{a \in \mathscr{A} \setminus A} \sum_{s \in \{-,+\}} -E(a^s, V, T) \log E(a^s, V, T) \\
relevancyT(P, \rho) &= true \Leftrightarrow trendRelevancy(P) \geq \rho
\end{aligned}
\tag{2.3}
$$

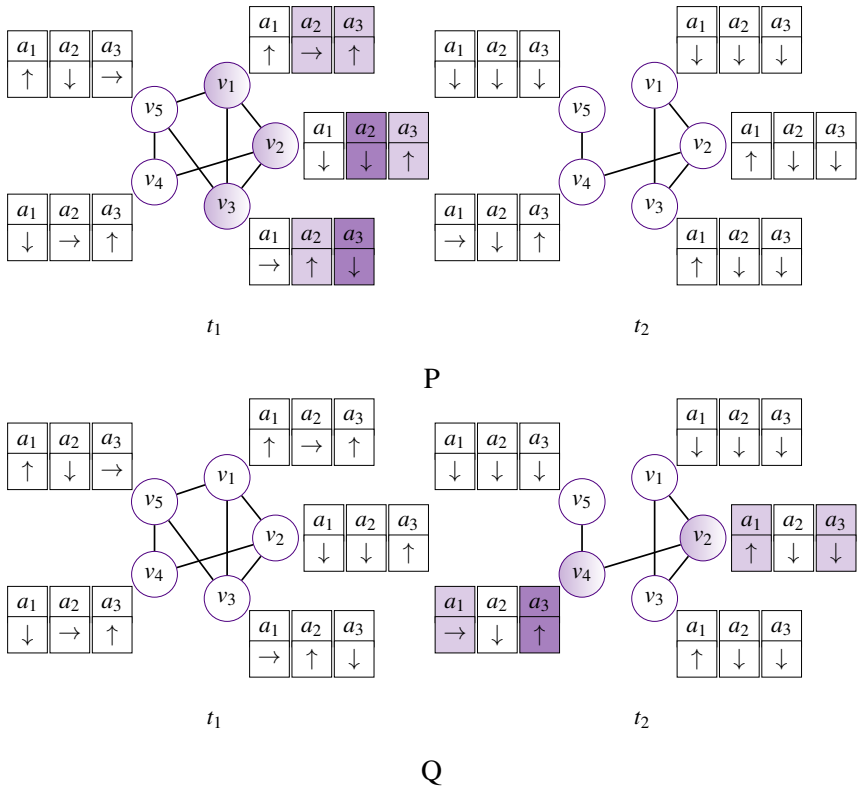


Figure 2.5: Illustration of the temporal dynamics on the toy example (Figure **??**) for (top) pattern $P = \{(v_1, v_2, v_3), (t_2), (a_2^-, a_3^-)\}$ ($temporalDynamic(P, \tau) = \frac{2}{6} = 0.33$) and (bottom) pattern $Q = \{(v_2, v_4), (t_1), (a_1^-, a_3^+)\}$ ($temporalDynamic(Q, \tau) = \frac{1}{4} = 0.25$)

The more a co-evolution pattern is trend relevant, the higher this measure. For instance, on DBLP, if the pattern involves two authors who publish nearly always together, the value of trend relevancy will be low as they have a common publication policy in almost all conferences. Interestingly, a pattern that describes authors who collaborate on a work but write papers in parallel in different conferences, may have a high trend relevancy value.

Given $\mathscr{G}$ the graph in Figure 2.2 (page 9) and $\rho = 0.25$, Figure 2.6 represents the elements considered to compute trend relevancy. The colored vertices are the vertices of the pattern at the timestamps of the pattern. Contrary to previous examples, the dark colored attributes are the increasing trends and the light colored attributes are the decreasing trends, while constant trends are not colored.

- Given $P = \{V, T, \Omega\} = \{(v_1, v_2, v_3), (t_2), (a_2^-, a_3^-)\}$,

$$\begin{aligned} trendRelevancy(P) &= \min_{a \in \{a_1\}} \sum_{s \in \{-,+\}} -E(a^s, (V), (T)) \times log((E(a^s, (V), (T)) \\ &= -\frac{2}{3} \times log(\frac{2}{3}) - \frac{1}{3} \times log(\frac{1}{3}) \\ &= 0.28 \end{aligned}$$

then $relevancyT(P, \rho)$ is true, P is a valid pattern.

- Given $Q = \{V', T', \Omega'\} = \{(v_1, v_2, v_4), (t_1), (a_3^+)\}$,

$$\begin{aligned} trendRelevancy(Q) &= \min_{a \in \{a_1, a_2\}} \sum_{s \in \{-,+\}} -E(a^s, (V'), (T')) \times log(E(a^s, (V'), (T')) \\ &= min\big(-\frac{2}{3} \times log(\frac{2}{3}) - \frac{1}{3} \times log(\frac{1}{3}), -\frac{1}{3} \times log(\frac{1}{3}) - \frac{0}{3} \times log(\frac{0}{3})\big) \\ &= min(0.28, 0.16) = 0.16 \end{aligned}$$

  then $relevancyT(Q, \rho)$ is false, Q is not a valid pattern.

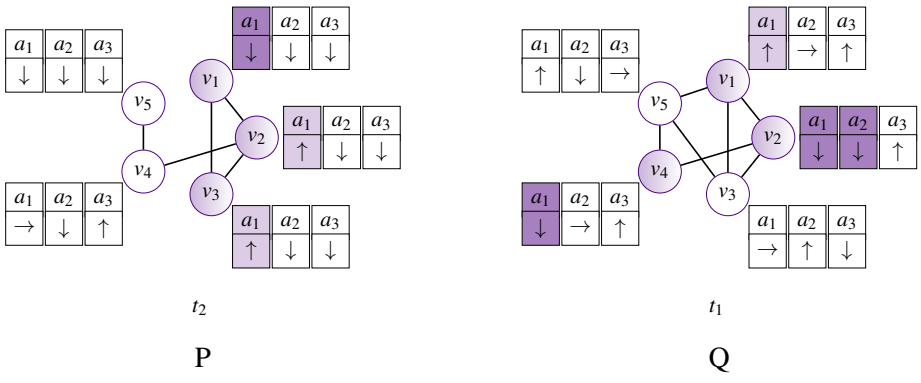


Figure 2.6: Trend relevancy (toy example from Figure 2.3) for (left) $P = \{V, T, \Omega\} = \{(v_1, v_2, v_3), (t_2), (a_2^-, a_3^-)\}$ and (right) $Q = \{V', T', \Omega'\} = \{(v_1, v_2, v_4), (t_1), (a_3^+)\}$. Relevancy of P is higher (0.28) than relevancy of Q (0.16).

These three measures makes it possible to discard non interesting – because trivial – patterns. However, these measures are really useful only if the related constraints are efficiently pushed into an algorithm. Fortunately, it was made possible thanks to the definition of lower bounds and upper bounds on these measures. More precisely, during the enumeration of a pattern, this allows to bound the measure the descendants of the pattern may reach and therefore to early prune unpromising parts of the search space.

2.4 Hierarchical Co-Evolution Patterns

Taking into account prior knowledge in the pattern discovery leads to much more relevant patterns. A direct and easy way to take some prior into account is the consideration of hierarchies among the items as in the seminal work of Srikant and Aggrawal [Srikant and Agrawal, 1996] or later in [Han and Fu, 1999]. Similarly, some attempts have been done in plain graphs [Inokuchi, 2004, Cakmak and Özsoyoglu, 2008]. Considering the co-evolution patterns, taking into account hierarchical relations over the vertex attributes allows to obtain smaller patterns, taking benefit from the hierarchies to find patterns with the good granularity level of description.

A hierarchy $\mathscr{H}$ on $\mathscr{A}$ is a tree where the edges are a relation is_a, i.e., specialization (resp. generalization) corresponds to a path from the root to the leaves (resp. from the leaves to the root) in the tree. The node $\mathscr{A}ll$ is the root of the tree and attributes of $\mathscr{A}$ are the leaves. Different functions are defined to use such hierarchy:

- $parent(x)$ returns the direct parent of the node x
- $children(x)$ returns the direct children of the node x
- $up(x)$ returns all the ancestors of x, x included, $\mathscr{A}ll$ excluded

- $down(x)$ returns all the descendants of x, x included, $\mathscr{A}ll$ excluded
- $leaf(x)$ returns all the leaves descendants of x, i.e., $down(x) \cap \mathscr{A}$

Moreover the domain of the hierarchy $dom(\mathscr{H})$ is defined as all the nodes except the root, i.e., $dom(\mathscr{H}) = down(\mathscr{A}ll)$. This hierarchy is an a priori knowledge of the expert.

Figure 2.7 presents an example of hierarchy on a toy example.

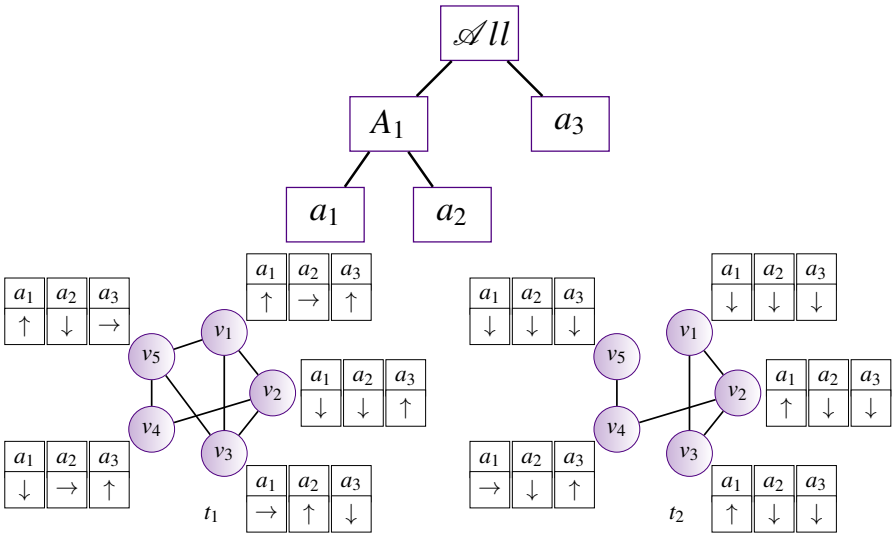


Figure 2.7: A toy example whose vertex attributed are provided with a hierarchy.

The root of the hierarchy is not in the domain of the hierarchy. It makes possible to consider unrelated attributes as, for instance, the number of publication of a scientist and its hair color. Unrelated categories of attributes will appear as root children.

The hierarchy is a tree where each node has at most one parent. The use of a forest or a directed acyclic graph could also be studied. However the constraints defined thereafter are not valid on these types of hierarchies.

A hierarchical co-evolution pattern P is very similar to a co-evolution pattern defined previously. The intuition behind this kind of pattern remains the same. It is a dynamic attributed subgraph whose vertices follow the same trend over a subset of attributes. The difference here is that the subset of attributes does not belong to $\mathscr{A}$ but $dom(\mathscr{H})$.

Definition 2.6 (Hierarchical co-evolution Pattern)**.** *Given a graph $\mathscr{G} = (\mathscr{V}, \mathscr{T}, \mathscr{A})$ and a hierarchy $\mathscr{H}$, a hierarchical co-evolution pattern is a triplet $P = (V, T, \Omega)$ s.t.:*

- *$V \subseteq \mathscr{V}$ is a subset of the vertices of the graph.*
- *$T \subset \mathscr{T}$ is a subset of not necessarily consecutive timestamps.*
- *Ω is a set of signed attributes, i.e., $\Omega \subseteq A \times S$ with $A \subseteq dom(\mathscr{H})$ and $S = \{+, -\}$ a set of trends.*

The two following conditions must hold:

1. *Each signed attribute a^s with $a \in A$ and $s \in \{+, -\}$ defines a trend that has to be satisfied by any vertex $v \in V$ at any timestamp $t \in T$: $evolHierarchical(P) = true$,*
2. *At each timestamp $t \in T$, the vertices of the pattern have to be closely related through the structure E_t of the graph: $related(P) = true$.*

Taking into account a hierarchy imposes to modify the computation of the trends. Indeed, an attribute which is not a leaf in the hierarchy does not appear explicitly in the dynamic attributed graphs. I.e., the

value of such an attribute is computed, for each vertex, from the leaves it covers. Therefore, the value of an attribute at a time t is defined as the sum of the values of its "children" attributes.

The attribute value of a parent node within the hierarchy is evaluated by adding the corresponding values of its children. Therefore, even if the trend conveyed by an attribute is true, it is important to assess how this information is valid, i.e., if the trends associated to its children are similar. Indeed, if a children attribute has a large increase while the other attributes have a small decrease, the sum associated to the parent attribute may result in an increase that is not followed by most of its leaves, leading to misleading patterns. To overcome this issue, the purity measure of a pattern P evaluates the proportion of leaves of an attribute that effectively share the trends.

Definition 2.7 (Purity (P) and associated constraint)**.** *Given the Kronecker function $\delta_{condition}$ and given a user-defined threshold $\psi \in [0,1]$, the purity of the pattern is defined as:*

$$purity(P) = \frac{\sum_{v \in V} \sum_{t \in T} \sum_{a^s \in leaf(\Omega)} \delta_{a^s(v,t)}}{|V| \times |T| \times |leaf(\Omega)|} \tag{2.4}$$
$$purityMin(P, \psi) \Leftrightarrow purity(P) \geq \psi$$

A pattern that respects the predicate *purityMin* is then considered as pure enough to provide a valid and relevant information.

For instance, in a DBLP graph, a pattern that describes the increasing number of publications in data-mining because the author has changed its publication policy and passed from 0 to several publications in data mining conferences is interesting. However a pattern that describes an increasing number of publications in data mining conferences whereas it increases in only one of them and stayed constant in the other is less interesting than the same pattern with the given conference.

Given $\mathscr{G}$ and $\mathscr{H}$ from Figure 2.7, $\Delta = 5$ and $\psi = 0.6$, Figure 2.8 represents the elements considered to compute purity. The colored vertices are the vertices of the pattern, the dark colored attributes are the "children" attributes that respect the trends and the light colored attributes are those that do not.

- Given $P = \{(v_1, v_2, v_3, v_4, v_5), (t_2), (A_1^-)\}$, $purity(P) = \frac{7}{10} = 0.7$ then $purityMin(P, \psi)$ is true, P is a valid pattern.
- Given $Q = \{(v_1, v_2, v_3, v_4, v_5), (t_1, t_2), (A_1^-)\}$, $purity(Q) = \frac{11}{20} = 0.55$ then $purityMin(Q, \psi)$ is false, Q is not a valid pattern.

One inconvenient when dealing with hierarchy is that it may introduce a lot a redundancy among the patterns. An important issue is thus to avoid this redundancy by identifying the good level of granularity of a pattern, i.e., deciding if the parent attribute well depict a phenomenon within the data or if it needs to be specialized. Figure 2.9 illustrates this. Two patterns are represented in two dimensions, i.e., a line is a vertex, a column is an attribute. A cell is colored in black if the trend of the attribute is respected by the vertex, in grey otherwise. Considering the pattern on the left, the trends are mainly respected and its purity is equal to 0.81. Specializing the attributes would lead to four patterns: $\{(v_2, v_3, v_4)(a_1)\}$, $\{(v_1, v_3, v_4)(a_2)\}$, $\{(v_1, v_2, v_3, v_4)(a_3)\}$ and $\{(v_1, v_2, v_3)(a_4)\}$. Then it seems much more interesting to keep the "parent" attribute A instead of fragmenting it. Considering the pattern on the right, the trends are much less respected and its purity is equal to 0.44. Trends of attributes a_1, a_2 and a_3 are respected by only one vertex, then specializing the attributes would lead to only one interesting pattern: $\{(v_1, v_2, v_3, v_4)(a_3)\}$. This pattern provides an information much more precise than its "parent", then it is here more interesting to specialize the attribute.

The question to answer is then: *Does the gain of purity obtained by specializing the attribute compensate the loss of generality ?* To this end, the *gain* of purity aims to evaluate whether the purity of the pattern would increase if it gets specialized or not.

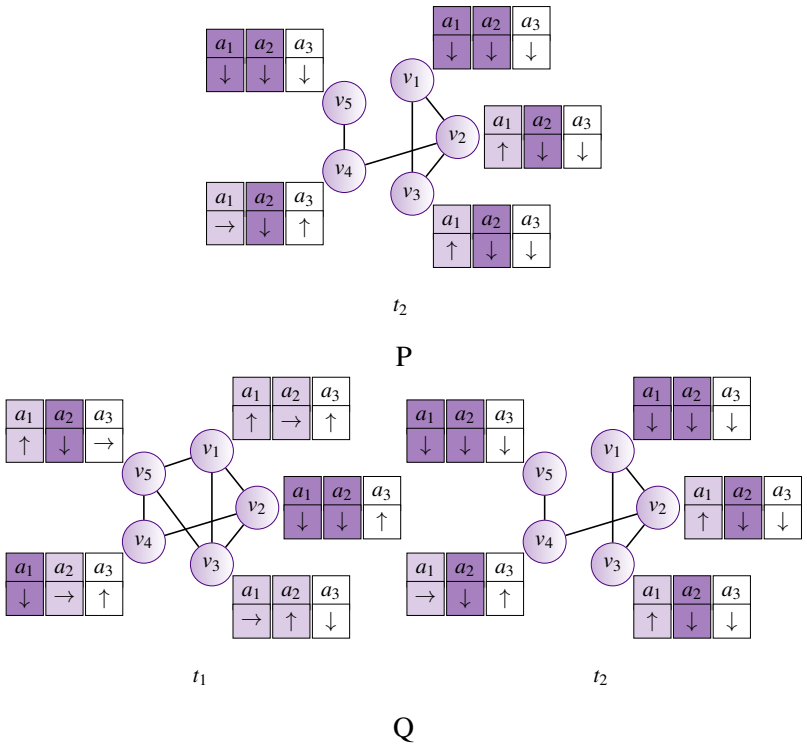


Figure 2.8: Purity of patterns for toy example of Figure 2.4. (Top) $P = \{(v_1, v_2, v_3, v_4, v_5), (t_2), (A_1^-)\}$, $purity(P) = \frac{7}{10} = 0.7$ (Bottom) $Q = \{(v_1, v_2, v_3, v_4, v_5), (t_1, t_2), (A_1^-)\}$, $purity(Q) = \frac{11}{20} = 0.55$. Given a purity threshold ψ set to 0.6, P is pure enough while Q is not.

Definition 2.8 (Gain of purity and associated constraint)**.** *Given a user-threshold $\gamma \geq 1$, the gain of purity is defined as the purity of the "children" pattern compared to the purity of its "parent" patterns:*

$$gain(P) \Leftrightarrow \frac{purity(P)}{max_{P_i \in parent(P)}(purity(P_i))} \geq \gamma$$
$$gainMin(P, \gamma) \Leftrightarrow gain(P) > \gamma \tag{2.5}$$

where $P_i \in parent(P)$ if $\exists a_i \in P_i.A$ and $\exists a \in P.A$ s.t. $a \in children(a_i)$ and $(P_i.A \setminus a_i) = (P.A \setminus a)$.

A pattern that respects $gainMin(P)$ is a pattern that provides an information more pure that its "parent" patterns. Let us consider the "children" pattern $Ch = \{(v_1, v_2, v_3, v_4), (t_1), (a_3)\}$ and its only "parent" pattern $Pa = \{(v_1, v_2, v_3, v_4), (t_1), (A)\}$ in Figure 2.9. Considering the figure on the left, the gain of purity of Ch is $gain(Ch) = \frac{1}{\frac{13}{16}} = 1.23$, considering the figure on the right, the gain of purity is $gain(Ch) = \frac{1}{\frac{7}{16}} = 2.29$. In the figure on the left the "parent" attribute abstracts well the behavior of the vertices, then the purity is low compared to the one computed for the figure on the right where the behavior is mainly followed by attribute a_3.

If no "children" pattern P_i of a pattern P respects $gainMin(P_i)$, then P provides more information than its "children", then the "children" pattern is discarded. It is noticeable that other "descendant" patterns could have a higher purity (purity of a leaf is necessary 1). Yet, the specialization would lead to a large number of possible patterns without necessary more information. Figure 2.10 illustrates the purity gain on the toy example.

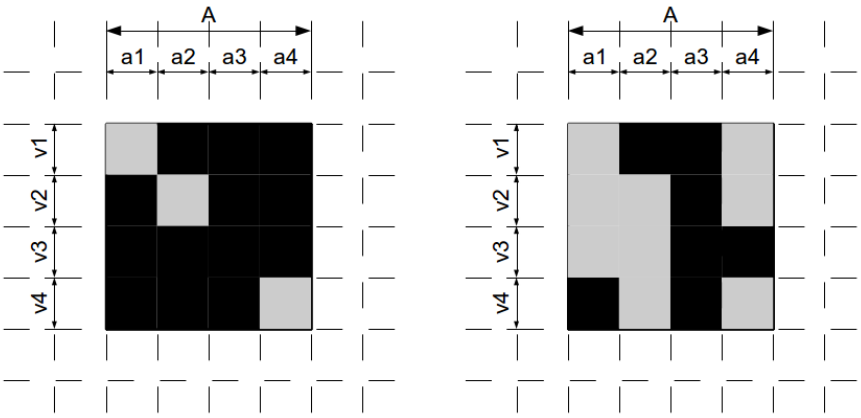


Figure 2.9: Considering a black square is a respected trend and a grey square is not: Do we keep A? Or do we develop a_1 and/or a_2 and/or a_3 and/or a_4?

Hierarchies on vertex attributes are taken into account in the discovery of hierarchical co-evolution patterns thanks to introduction of the purity measure, the purity gain and their associated constraints. This allows to depict phenomena within the data with the good level of granularity, not too general to avoid misleading interpretation, not too specific to uselessly increase the number of discovered patterns. Once again, these constraints are of interest only if they are efficiently pushed into an algorithm. The purity gain is easily handled since it just requires the comparison of the purity of a child pattern with the purity of its parent. The purity constraint is itself more challenging to handle because it has non monotonic behavior. Pushing this constraint requires the definition and the exploitation of a upper bound on the purity measures of the specializations of a pattern.

2.5 Examples of Co-Evolution Patterns in Real-World Datasets

Co-evolution patterns were part of the PhD thesis of Elise Desmier funded by the FOSTER project which aims at providing geologists tools for monitoring and better understanding the soil erosion. This process is based on multi-temporal very high resolution satellite images, sensor data and expert knowledge. Satellite images are transformed into spatio-temporal graphs in which some co-evolution patterns can describe soil erosion.

In this context, a slight modification of the predicated *evol* and focusing on patterns that contain a decreasing trend of NDVI (the vegetation index) is enough to discover co-evolution patterns that describe landslides. Figure 2.11 returns patterns that involve $NDVI^-$ and whose vertices are connected. These results were evaluated by an expert who certified that 69% of the shapes that are considered as a true landslide appear in the computed patterns, i.e., we found a large majority of the landslides in this extraction. Considering the extracting segments, 46% correspond to “landslides” while the rest does not. The 54% remaining regions can be classed in the 4 following categories:

1. *Regions nearby true landslides which have not been interpreted as landslides by the expert.* Looking more precisely at the picture, the red regions are often surrounded by small white ones which we consider mainly as “border effect”. Indeed, first the classification as “landslide” or “not landslide” was made by an expert looking at the picture, however two experts can make different choices on a same segment. Second, the vegetation around a landslide can also be damaged and if the respective segments can not be considered as landslides, yet, the NDVI decreased.

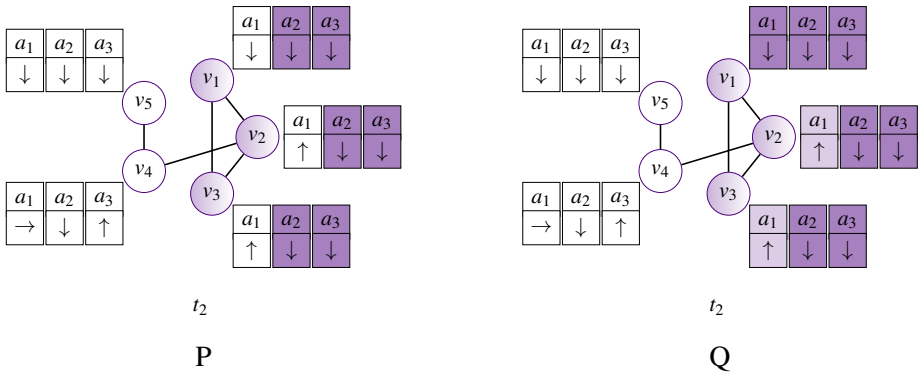


Figure 2.10: Given $\mathscr{H}$ the hierarchy in Figure 2.7 and $\gamma = 1.2$, this Figure represents the elements considered to compute purity of the "children" pattern P and the "parent" pattern Q. The colored vertices are the vertices of the pattern, the dark colored attributes are the attributes that respect the trends and the light colored attributes are those that do not. Given $P = \{(v_1, v_2, v_3), (t_2), (a_2^-, a_3^-)\}$, it has only one "parent" pattern $Q = \{(v_1, v_2, v_3), (t_2), (A_1^-, a_3^-)\}$. $purity(P) = 1$ and $purity(Q) = \frac{7}{9} = 0.78$. Then $gain(P) = \frac{1}{0.78} = 1.28$ and $gainMin(P, \gamma)$ is valid and P is reported to the user instead of Q.

2. *Deforested area not due to landslides.* When comparing the two images, some patterns correspond clearly to human activity. For instance, the large white region on the middle left of the picture. Before the landslide there were a forest, after the landslide there is no tree anymore.
3. *Regions found due to misalignment of the segmentation technique.* We explained in last section that the two images are not perfectly aligned. This causes some patterns to be extracted, as for instance the main road that cross the image from top left to bottom down.
4. *Regions that represent cities and human activity footprints.* Cities and other human constructions are indeed almost all found, some of them are easily recognizable in the picture. There is no vegetation in this regions, however the spectral response in red and infra-red seems to imply a decreasing of the NDVI due to the change of luminosity of the two images.

When the expert knowledge is not given to guide the discovery, identifying the good patterns in much more challenging and the use of the different interestingness measures as well as the use of hierarchical relations become necessary. The following dataset illustrates how these measures and the hierarchies are beneficial to discover co-evolution patterns.

RITA "On-Time Performance" database [3] contains on-time arrival data for non-stop US domestic flights by major air carriers. From this database, many dynamic attributed graphs that aggregate data over different period of time can be built. For instance, let us aggregate the data over each week between 2005/08/01 and 2005/09/25 to study the consequences of hurricane Katrina on US airports.

To characterize the impact of the hurricane Katrina on the US domestic flights, we set constraints as follows: $\vartheta = 10$ (minimum volume), $\kappa = 0.6$ (maximum vertex specificity, p.11), $\tau = 0.2$ (maximum temporal dynamic, p.11), $\rho = 0.1$ (minimum trend relevancy, p.12) and the vertices are required to be connected at each timestamp. 37 patterns were extracted in 14 seconds. Let us study two of these patterns: (i) the pattern with the smallest *temporalDynamic* value, and (ii) the pattern with the highest *trendRelevancy* value. These patterns and Katrina's track[4] are shown in Figure 2.12.

Pattern (i) involves 71 airports (in red on Figure 2.12 (left)) whose arrival delays increase over 3

[3] http://www.transtats.bts.gov

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

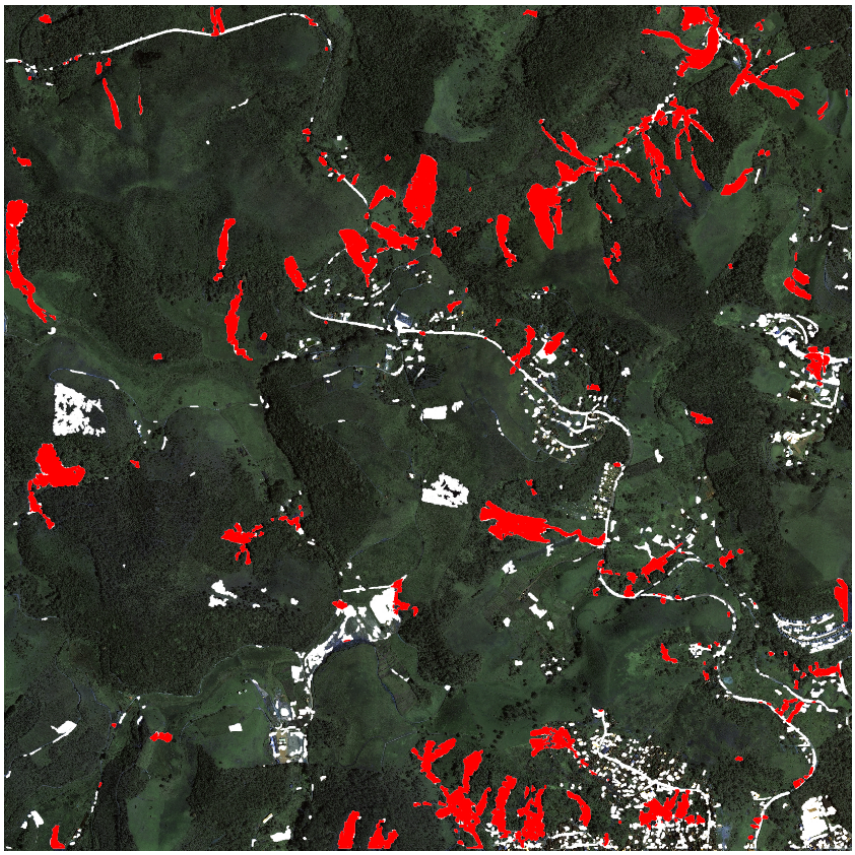


Figure 2.11: Shapes involved in the patterns: true landslides (red) and other phenomena (white).

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 $temporalDynamic = 0$, 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 $trendRelevancy$ value equal to 0.81 and includes 5 airports (in yellow on Figure 2.12 (left)) whose number of departures and arrivals increased over the three weeks following Katrina hurricane. Three out of the five airports are in the Katrina's trajectory while the two other ones were 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 ($vertexSpecificity = 0.29$, $temporalDynamic = 0.2$).

Looking for hierarchical co-evolution patterns enables to better summarize the output. Let us illustrate this through the discovery of frequent trends. The parameters are set to $min_V = 50$, $min_T = 3$ (minimum number of timestamps), $\psi = 0.9$ and $\gamma = 1$ or $\gamma = 1.2$. In both extractions, results are obtained in 1 second ; 12 patterns were extracted in the first experiment and 6 in the second one. Figure 2.13 reports a comparison of the results of both extractions. Light colored circles represent the first extraction (i.e., with $\gamma = 1$) and dark colored circles and bold attributes represent the second extraction (i.e., with $\gamma = 1.2$).

Two patterns are also found larger in the second extraction, i.e., with the "parent" attribute and more airports. These two examples are presented in the bottom right of Figure 2.13. The first example is a pattern that contains 50 airports that have a decreasing number of departure in the first extraction (i.e., with $\gamma = 1.2$) and 51 airports that have a decreasing number of flights in the second extraction (i.e., with $\gamma = 1.2$) including the 50 vertices of the first extraction. The second example is similar with respectively

Figure 2.12: Airports (left) involved in the top temporalDynamic pattern (in red) and in the top trendRelevancy (in yellow) and the Katrina's track (right).

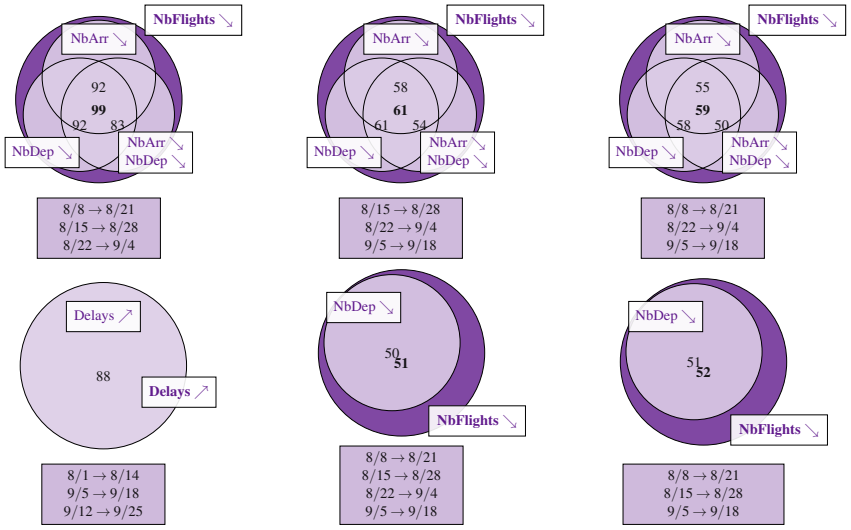


Figure 2.13: Comparison of the resulting patterns with "Katrina" while varying γ. Light colored circles represent the first extraction (i.e., with $\gamma = 1$) and dark colored circles and bold attributes represent the second extraction (i.e., with $\gamma = 1.2$).

51 and 52 airports. It is noticeable that even if the pattern in the first extraction concerns only the number of departures and not the number of arrivals, the patterns extracted in the second experiment with the number of flights have a purity greater than 90%. Then the patterns in the second extraction provide more information.

2.6 Conclusion

We designed some pattern domains for dynamic attributed graphs in which vertex attributes are numeric and can be organized through a hierarchy. We defined some primitives to enforce a structure on the induced subgraphs (i.e., connected components, cliques, dense subgraphs with the diameter constraint) or similarities between vertices. We also introduced some interestingness measures to assess co-evolution patterns regarding their specificity according to each dimension (i.e., attributes, timestamps, and vertices). The associated constraints were studied in order to be efficiently pushed into the algorithm we defined. We applied the so-called `MINTAG` on different types of dynamic attributed graphs. This algorithm is able to uncover interesting phenomena in such data.

This work can be extended in several ways. First the co-evolution patterns – in their current definition – are tri-set of vertices, timestamps, and trends on attributes. It would be interesting and challenging to define a pattern domain that involve sequences of attributed subgraphs, i.e., it could describe phenomena where (i) some vertices appear partially and not on all the timestamps, (ii) a vertex may follow different trends through time within the pattern. Obviously, the consideration of such a pattern domain requires the definition of adequate primitives and also an efficient associated discovery algorithm. Notice that the search space would be huger and would require the use of heuristic based algorithms returning a subset of all solutions. Another interesting line of research is the identification of the simplest patterns in term of syntax that uncover a phenomenon. Indeed, co-evolution patterns are tri-set and each set must be non empty. Therefore, the discovered patterns may be unnecessarily over-specified, thus difficult to assimilate for the end-user and potentially misleading.

Chapter 3

Looking for Links Between Structure and Vertex Attributes

Contents

3.1 Introduction

The previous chapter aims to describe our different contributions on the analysis of dynamic attributed graphs, i.e., the definition of pattern domains, interestingness measures and the related constraints. In the co-evolution patterns, the graph structure is taken into account via the definition of the *related* constraint: the patterns depict vertices that fulfill either a constraint on the diameter of the vertex induced subgraph or a constraint on the similarity of the vertices. In other words, the graph structure is simply considered as a possible constrained dimension and not really as an analysis dimension itself. Therefore, it is not possible to establish links between the graph topology and the vertex attributes to answer questions such as *do nodal attributes have an impact on the role (i.e., the importance) of the vertices within the graph?*

In this chapter, we propose to mine the graph topology simultaneously with the vertex attributes. We strive to elucidate relationship between graph topology and vertex attributes in both static and dynamic graphs.

We present two contributions:

- We first present the topological patterns that aim to reveal the links that exist between the relation encoded by the graph and the vertex attributes in static graphs. This work is a collaboration with Adriana Prado, Céline Robardet and Jean-François Boulicaut [Prado *et al.*, 2013, Boulicaut *et al.*, 2016].
- We then present the triggering pattern discovery problem to follow similar motivations for dynamic attributed graphs. Especially, this mining task aims to find sequences of vertex attribute variations that are generally followed by a toppological variations. This work was done with Mehdi Kaytoue, Yoann Pitarch and Céline Robardet [Kaytoue *et al.*, 2015].

3.2 Mining Graph Topological Patterns to Find Co-Variations Among Vertex Descriptors

Existing methods that support the discovery of local patterns in graphs mainly focus on the topological structure of the patterns, by extracting specific subgraphs while ignoring the vertex attributes (cliques [Makino and Uno, 2004], quasi-cliques [Liu and Wong, 2008, Uno, 2010]), or compute frequent relationships between vertex attribute values (frequent subgraphs in a collection of graphs [Jiang and Pei, 2009] or in a single graph [Bringmann and Nijssen, 2008]), while ignoring the topological status of the vertices within the whole graph, e.g., the vertex connectivity or centrality. The same limitation holds for the methods proposed in [Khan *et al.*, 2010, Mougel *et al.*, 2012] and [Silva *et al.*, 2012], which identify sets of vertices that have similar attribute values and that are close neighbors. Such approaches only focus on a local neighborhood of the vertices and do not consider the connectivity of the vertex in the whole graph.

To investigate the relations that may exist between the position of the vertices within the graph and their attribute values, we proposed to extract topological patterns that are sets made of vertex attributes and topological measures. Such measures quantify the topological status of each vertex within the graph. Some of these measures are based on the close neighborhood of the vertices (e.g., the vertex degree), while others describe the connectivity of a vertex by considering its relationship with all other vertices (e.g., the centrality measures). Combining such microscopic and macroscopic properties characterizes the connectivity of the vertices and it may be a sound basis to explain why some vertices have similar attribute values.

Topological patterns of interest are composed of vertex properties that behave similarly over the vertices of the graph. The similarity among vertex properties can be captured by quantifying their correlation, which may be positive or negative. To that end, we extend the Kendall rank correlation coefficient to any number of variables –initially proposed in [Calders *et al.*, 2006] – as well as to negative correlation. Whereas this measure is rather theoretically sounded, its evaluation is computationally demanding as it requires to consider all vertex pairs to estimate the proportion of which that supports the pattern. The well known optimization techniques that are used for evaluating the correlation between two variables (and that leads to a theoretical complexity in $O(n \log n)$) do not extend directly when a higher number of variables is considered. We tackled this issue and proposed several optimization and pruning strategies that makes it possible to use this approach on large graphs.

We also introduced several interestingness measures of topological patterns that differ by the pairs of vertices that are considered while evaluating the correlation between descriptors: (1) While all the vertex pairs are considered, patterns that are true all over the graph are extracted; (2) When including only the vertex pairs that are in a specific order regarding to a selected numerical or ordinal attribute reveals the topological patterns that emerge with respect to this attribute; (3) Examining the vertex pairs that are connected in the graph makes it possible to identify patterns that are *structurally correlated* to the relationship encoded by the graph. Besides, we designed an operator that identifies the top k representative vertices of a topological pattern.

Topological patterns were used to analysis different types of networks (e.g., biological networks, patent network, co-authorship network). Especially, we studied the network of authors who cooperate at some time with Pr. Katharina Morik according the data available in DBLP database. This special use-case was done on the occasion of her 60th birthday [Michaelis *et al.*, 2016].

3.2.1 Topological Vertex Properties

Let us consider a non-directed attributed graph $G = (V, E, L)$, where V is a set of n vertices, E a set of m edges, and $L = \{l_1, \cdots, l_p\}$ a set of p numerical or ordinal attributes associated with each vertex of V. Important properties of the vertices are encoded by the edges of the graph. From this relation, we can compute some topological properties that synthesize the role played by each vertex in the graph. The topological properties we are interested in range from a microscopic level – those that described a vertex based on its direct neighborhood – to a macroscopic level – those that characterize a vertex by considering its relationship to all other vertices in the graph. Statistical distributions of these properties are generally used to depict large graphs (see, e.g., [Albert and Barabási, 2000, Kang *et al.*, 2011]). We propose here to use them as vertex descriptors.

Microscopic Properties. Let us consider here only three topological properties to describe the direct neighborhood of a vertex v:

- The degree of v is the number of edges incident to v ($deg(v) = |\{u \in V, \{u,v\} \in E\}|$). When normalized by the maximum number of edges a vertex can have, it is called the degree centrality coefficient: $\text{DEGREE}(v) = \frac{deg(v)}{n-1}$.
- The clustering coefficient evaluates the connectivity of the neighbors of v and thus its local density:

$$\text{CLUST}(v) = \frac{2|\{\{u,w\} \in E, \{u,v\} \in E \wedge \{v,w\} \in E\}|}{deg(v)(deg(v)-1)}$$

Mesoscopic Property. We also consider the position of each vertex to the center of the graph, that is the distance – the number of edges of a shortest path – to a peculiar vertex. In the following, we call this property the MORIK_NUMBER(V) as we consider the relative position of the vertices to the vertex that corresponds to Katharina Morik.

Macroscopic Properties. We consider five macroscopic topological properties to characterize a vertex while taking into account its connectivity to all other vertices of the graph.

- The relative importance of vertices in a graph can be obtained through centrality measures [Freeman, 1977]. Closeness centrality $\text{CLOSE}(v)$ is defined as the inverse of the average distance between v and all other vertices that are reachable from it. The distance between two vertices is defined as the number of edges of the shortest path between them: $\text{CLOSE}(v) = \frac{n}{\sum_{u \in V} |shortest\_path(u,v)|}$.
- The betweenness centrality $\text{BETW}(v)$ of v is equal to the number of times a vertex appears on a shortest path in the graph. It is evaluated by first computing all the shortest paths between every pair of vertices, and then counting the number of times a vertex appears on these paths: $\text{BETW}(v) = \sum_{u,w} \mathbb{1}_{shortest\_path(u,w)}(v)$.
- The eigenvector centrality measure (EGVECT) favours vertices that are connected to vertices with high eigenvector centrality. This recursive definition can be expressed by the following eigenvector equation $Ax = \lambda x$ which is solved by the eigenvector x associated to the largest eigenvalue λ of the adjacency matrix A of the graph.
- The PAGERANK index [Brin and Page, 1998] is based on a random walk on the vertices of the graph, where the probability to go from one vertex to another is modelled as a Markov chain in which the

states are vertices and the transition probabilities are computed based on the edges of the graph. This index reflects the probability that the random walk ends at the vertex itself:

$$\text{PAGERANK}(v) = \alpha \sum_u \mathbb{1}_E(\{u,v\}) \frac{\text{PAGERANK}(u)}{deg(u)} + \frac{1-\alpha}{n}$$

where the parameter α is the probability that a random jump to vertex v occurs.

- Network constraint [Wang *et al.*, 2012] evaluates to what extent person's contacts are redundant

$$\text{NETWORK}(v) = \sum_{u|(u,v)\in E} \left[\frac{1}{deg(v)} + \sum_{w|(u,w)\ and\ (v,w)\in E} \left(\frac{1}{deg(v)}\frac{1}{deg(u)}\right)\right]^2$$

When its value is low, the contacts are rather disconnected, whereas when it is high, the contacts are close or strongly tied.

These 9 topological properties characterizes the graph relationship encoded by E. These properties, along with the set of vertex attributes L, constitutes the set of vertex descriptors $\mathscr{D}$ used in this paper.

3.2.2 Topological Patterns

Let us now consider topological patterns as a set of vertex attributes and topological properties that behave similarly over a large part of the vertices of the graph. We assume that all topological properties and vertex attributes are of numerical or ordinal type, and we propose to capture their similarity by quantifying their co-variation over the vertices of the graph. Topological patterns are defined as $P = \{{D_1}^{s_1}, \cdots, {D_\ell}^{s_\ell}\}$, where D_j, $j = 1 \ldots \ell$, is a vertex descriptor from $\mathscr{D}$ and $s_j \in \{+,-\}$ is its co-variation sign.

In the following, we propose three pattern interestingness measures that differ in the pairs of vertices considered for their evaluation: (1) considering all the pairs of vertices enable to find patterns that are true all over the graph; (2) examine the vertex pairs that are connected in the graph makes possible to identify patterns that are structurally correlated to the relationship encoded by the graph; (3) taking into account only the vertex pairs that are in a specific order with respect to a selected attribute, reveals the topological patterns that emerge with respect to this attribute.

Topological patterns over the whole graph. Several signed vertex descriptors co-vary if the orders induced by each of them on the set of vertices are consistent. This consistency is evaluated by the number of vertex pairs ordered the same way by all descriptors. The number of such pairs constitutes the so-called support of the pattern. This measure can be seen as a generalization of the Kendall's τ measure. When we consider all possible vertex pairs, this interestingness measure is defined as follows:

Definition 3.1 ($Supp_{all}$)**.** *The support of a topological pattern P over all possible pairs of vertices is:*

$$Supp_{all}(P) = \frac{|\{(u,v)\in V^2 \mid \forall D_j^{s_j}\in P : D_j(u) \rhd_{s_j} D_j(v)\}|}{\binom{n}{2}}$$

where $\rhd_{s_j}$ denotes $<$ when s_j is equal to $+$, and $\rhd_{s_j}$ denotes $>$ when s_j is equal to $-$.

This measure gives the number of vertex pairs (u,v) such that u is strictly lower than v on all descriptors with sign $+$, and u is strictly higher than v on descriptors with sign $-$.

As mentioned in [Calders *et al.*, 2006], $Supp_{all}$ is an anti-monotonic measure for positively signed descriptors. This is still true when considering negatively signed ones: adding D_{l+1}^- to a pattern P leads to a support lower than or equal to that of P since the pairs (u,v) that support P must also satisfy $D_{l+1}(u) > D_{l+1}(v)$. Besides, when adding descriptors with negative sign, the support of some patterns can be deduced from others, the latter referred to as symmetrical patterns.

Property 3.1 (Support of symmetrical patterns)**.** *Let P be a topological pattern and $\overline{P}$ be its symmetrical, that is, $\forall D_j^{s_j} \in P$, $D_j^{\overline{s_j}} \in \overline{P}$, with $\overline{s_j} = \{+,-\} \setminus \{s_j\}$. If a pair (u,v) of V^2 contributes to the support of P, then the pair (v,u) contributes to the support of $\overline{P}$. Thus, we have $Supp_{all}(P) = Supp_{all}(\overline{P})$.*

Topological patterns and their symmetrical patterns are semantically equivalent. To avoid the irrelevant computation of duplicate topological patterns, we exploit Property 3.1 and enforce the first descriptor of a pattern P to be signed by $+$.

Mining frequent topological patterns consists in computing all sets of signed descriptors P, but not their symmetrical ones, such that $Supp_{all}(P) \geq minsup$, where $minsup$ is a user-defined minimum support threshold.

Other interestingness measures To identify most interesting topological patterns, we propose to give to the end-user the possibility of guiding its data mining process by querying the patterns with respect to their correlation with the relationship encoded by the graph or with a selected descriptor. Therefore, we revisit the notion of emerging patterns [Dong and Li, 1999] by identifying the patterns whose support is significantly greater (i.e., according to a growth-rate threshold) in a specific subset of vertex pairs than in the remaining ones. This subset can be defined in different ways according to the end-user's motivations: either it is defined by the vertex pairs that are ordered with respect to a selected descriptor called the class descriptor, or it is equal to E, the set of edges. Whereas the former highlights the correlation of a pattern with the class descriptor, the latter enables to characterize the importance of the graph structure within the support of the topological pattern.

Emerging patterns w.r.t. a selected descriptor

Let us consider a selected descriptor $C \in \mathscr{D}$ and a sign $r \in \{+,-\}$. The set of pairs of vertices that are ordered by C^r is

$$\mathscr{C}_{C^r} \quad = \quad \{(u,v) \in V^2 \mid C(u) \rhd_r C(v)\}$$

The support measure based on the vertex pairs of $\mathscr{C}_{C^r}$ is defined below.

Definition 3.2 ($Supp_{C^r}$)**.** *The support of a topological pattern P over C^r is:*

$$Supp_{C^r}(P) = \frac{|\{(u,v) \in \mathscr{C}_{C^r} \mid \forall D_j^{s_j} \in P : D_j(u) \rhd_{s_j} D_j(v)\}|}{|\mathscr{C}_{C^r}|}$$

Analogously, the support of P over the pairs of vertices that do not belong to $\mathscr{C}_{C^r}$ is denoted $Supp_{C^{\bar{r}}}(P)$. To evaluate the impact of C^r on the support of P, we consider the growth rate of the support of P over the partition of vertex pairs $\{\mathscr{C}_{C^r}, \mathscr{C}_{C^{\bar{r}}}\}$: $Gr(P,C^r) = \frac{Supp_{C^r}(P)}{Supp_{C^{\bar{r}}}(P)}$

If $Gr(P,C^r)$ is greater than a minimum growth-rate threshold, then P is referred to as emerging with respect to C^r. If $Gr(P,C^r) \approx 1$, P is as frequent in $\mathscr{C}_{C^r}$ as in $\mathscr{C}_{C^{\bar{r}}}$. If $gr(P,C^r) \gg 1$, P is much more frequent in $\mathscr{C}_{C^r}$ than in $\mathscr{C}_{C^{\bar{r}}}$. The intuition behind this definition is to identify the topological patterns that are mostly supported by pairs of vertices that are also ordered by the selected descriptor.

Emerging patterns w.r.t. the graph structure

It is interesting to measure if the graph structure plays an important role in the support of a topological pattern P. To this end, we define a similar support measure based on pairs that belongs to E, the set of edges of the graph:

$$\mathscr{C}_E = \{(u,v) \in V^2 \mid \{u,v\} \in E\}$$

Based on this set of pairs, we define the support of P as:

Definition 3.3 ($Supp_E$)**.** *The support of a topological pattern P over the pairs of vertices that are linked in G is:*

$$Supp_E(P) = \frac{2|\{(u,v)\in\mathscr{C}_E \mid \forall D_j^{s_j}\in P: D_j(u)\rhd_{s_j} D_j(v)\}|}{|\mathscr{C}_E|}$$

The maximum value of the numerator is $\frac{|\mathscr{C}_E|}{2}$ since: (1) if $(u,v)\in\mathscr{C}_E$ then $(v,u)\in\mathscr{C}_E$, and (2) it is not possible that $\forall D_j^{s_j}\in P$, $D_j(u)\rhd_{s_j} D_j(v)$ and $D_j(v)\rhd_{s_j} D_j(u)$ at the same time. For instance, the pattern $\{h^+,i^-\}$ is supported by all the twenty possible pairs that are edges, its support is thus equal to 1. The support of P over the pairs of vertices that do not belong to $\mathscr{C}_E$ is denoted $Supp_{\overline{E}}(P)$.

As before, to evaluate the impact of E on the support of P, we consider the growth rate of the support of P over the partition of vertex pairs $\{\mathscr{C}_E,\mathscr{C}_{\overline{E}}\}$: $Gr(P,E)=\frac{Supp_E(P)}{Supp_{\overline{E}}(P)}$.

$Gr(P,E)$ enables to assess the impact of the graph structure on the pattern. Therefore, if $Gr(P,E)\gg 1$, P is said to be *structurally* correlated. If $Gr(P,E)\ll 1$, the graph structure tends to inhibit the support of P.

3.2.3 Top k Representative Vertices of Topological Patterns

The user may be interested in identifying the vertices that are the most representative of a given topological pattern, thus enabling the projection of the patterns back into the graph. As an example, in a DBLP like network, the representative vertices of the pattern $\{\text{TKDE}^+, \text{BETW}^-\}$ would be researchers with a relatively large number of IEEE TKDE papers and a low betweenness centrality measure.

We denote by $S(P)$ the set of vertex pairs (u,v) that constitutes the support of a topological pattern P:

$$S(P)=\{(u,v)\in V^2 \mid \forall D_j^{s_j}\in P : D_j(u)\rhd_{s_j} D_j(v)\}$$

which forms, with V, a directed graph $G_P=(V,S(P))$. This graph satisfies the following property.

Property 3.2. *The graph $G_P=(V,S(P))$ is transitive and acyclic.*

Proof. Let us consider $(u,v)\in V^2$ and $(v,w)\in V^2$ such that, $\forall D_j^{s_j}\in P : D_j(u)\rhd_{s_j} D_j(v)$ and $D_j(v)\rhd_{s_j} D_j(w)$. Thus, $D_j(u)\rhd_{s_j} D_j(w)$ and $(u,w)\in S(P)$. Therefore, G_P is transitive.

As $\rhd_s\in\{<,>\}$, it stands for a strict inequality. Thus, if $(u,v)\in S(P)$, $(v,u)\notin S(P)$. Furthermore, as G_P is transitive, if there exists a path between u and v, there is also an arc $(u,v)\in S(P)$. Therefore, $(v,u)\notin S(P)$ and we can conclude that G_P is acyclic. □

As G_P is acyclic, it admits a topological ordering of its vertices, which is, in the general case, not unique. The top k representative vertices of a topological pattern P are identified on the basis of such a topological ordering of V and are the k last vertices with respect to this ordering. Considering that an arc $(u,v)\in S(P)$ is such that v dominates u on P, this vertex set contains the most dominant vertices on P. The top k representative vertices of P can be easily identified by ordering the vertices by their incoming degree.

3.2.4 TopGraphMiner Algorithm in a Nutshell

Although the support of topological patterns is an anti-monotonic measure, its computation is quadratic in the number of vertices of the graph which prevents the extraction of such patterns on large graph using classical pattern mining algorithms. To overcome this problem, we proposed in [Prado *et al.*, 2013] an upper bound on this measure that can be computed linearly in the number of vertices. This upper bound takes advantages of the presence of ties in the descriptor values. By pre-computing some indexes on the descriptors, almost all non frequent patterns are pruned without computing their support when the minimum support is high.

The computation of topological patterns is done in an ECLAT-based way [Zaki, 2000]. More precisely, all the subsets of a pattern P are always evaluated before P itself. In this way, by storing all frequent patterns in the hash-tree $\mathscr{M}$, the anti-monotonic frequency constraint is fully-checked on the fly.

We compute the upper bound on the support to prune non-promising topological patterns. When this upper bound is greater than the minimum threshold, the exact support is computed. Another optimization is based on the deduction of the support from already evaluated patterns: A pair of vertices that supports a pattern P can support pattern PA^+ or pattern PA^-, or none of them. Thus, another upper bound on $Supp_{all}(PA^-)$ is $Supp_{all}(P) - Supp_{all}(PA^+)$. Note that these patterns have already been considered before the evaluation of PA^-. So, to be stringent, we bound the support by taking the minimum between this value and the upper bound. When computing the support of the pattern, the top k representative vertices are also identified.

A full description of the algorithm and its optimization is given in [Prado *et al.*, 2013].

3.2.5 Studying Katharina Morik's Co-Authorship Network

TopGraphMiner is able to provide insightful patterns for analyzing a network. We illustrate this with the study of the scientific co-authorship network of Pr. Katharina Morik in the context of the book celebrating her [Michaelis *et al.*, 2016]. After presenting the attributed graph we generated from the DBLP digital library[5], we provide qualitative results that show the implication of Katharina Morik in the machine learning community.

Katharina Morik's co-authorship network

The co-authorship graph is built from the DBLP digital library. Regarding Katharina's bibliography, we select all the conference venues and journals in which Katharina has at least one DBLP entry[6]. We gather all the publications in these conference venues and journals since their foundation, and derived a graph where the vertices stand for the authors and edges link two authors who co-authored at least one paper in this corpus. To each vertex, we associate the number of publications in each of these 53 selected conferences or journals as vertex properties. We then removed isolated vertices, that is to say, authors who has no co-author in the selected publications. The resulting attributed graphs involves 81 222 vertices and 466 152 undirected edges. Notice that, even if this attributed graph is generated based on Katharina's publications, her co-authors only represent 0.1% of the vertices of the whole graph, while the vertices whose distance to Katharina is at most 2 represent less than 2% of the whole set of vertices. The average Morik number is 4.05 and 4033 authors have no path to Katharina (infinite Morik number). There are 1428 connected components.

Figure 3.1[7] presents this co-authorship graph restricted to the authors that are at most at a distance of 2 from Katharina and that have a degree value greater than 20. Applying the community detection Chinese Whisper algorithm [Biemann, 2006], we obtain 68 communities whose most salient are represented on the figure. The purple community, that gathers 177 authors including Katharina Morik herself, is very dense (1096 edges). It brings together well identified researchers in data mining, machine learning and data bases. The other main communities are labeled on the graph. Our goal is to analyse this graph with regard to several questions:

- Are there any interesting patterns among publications?
- Are there interesting trends between some authors' publications and topological properties?
- What about Katharina's role in this graph? Can we characterize the proximity to Katharina in terms of co-authorship?

[5] `http://dblp.uni-trier.de/`

[6] `http://www.dblp.org/search/index.php?query=author:katharina_morik`

[7] This visualization was generated by Gephi software: `https://gephi.org/`.

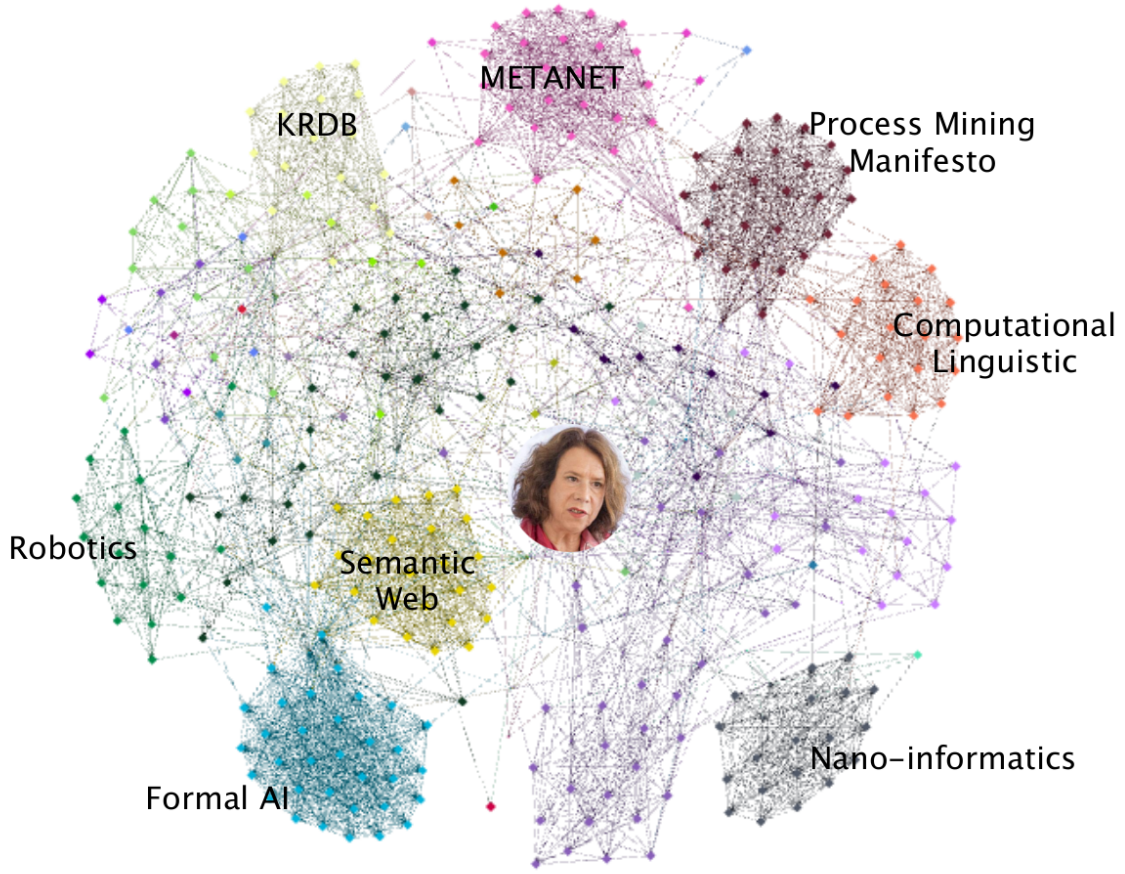


Figure 3.1: Research domains associated to Katharina's co-authors.

Most emerging pattern with respect to the Morik Number.

Table 3.1 presents two interesting patterns that strongly emerge with the Morik number. The first pattern gathers 4 conferences that are positively signed and the Morik number that is negatively signed: The more authors are close to Katharina Morik, the more they publish in IJCAI as well as in three other German conferences (KI - Künstliche Intelligenz, GWAI - German workshop on artificial intelligence and Informatik_Spektrum) Notice that GWAI changes its name to KI in 1993. The top 20 supporting authors gathers the German researchers in Artificial Intelligence. They are close to Katharina who is wellknown in the AI community research, and she also actively contributes to the animation of her national community.

The second pattern presented in Table 3.1 gathers the major conference venues and journals in Artificial Intelligence, Data Mining and Machine Learning. The top 20 supporting authors are all well established researchers in these research areas.

The first pattern with the Morik number positively signed is presented in Table 3.2. It gathers the conference ICASSP in signal processing that is positively signed and 3 conferences in Machine Learning that are negatively signed: The farther the authors from Katharina, the more they published at ICASSP and the less they contribute to AI conferences IJCAI, KI and KR (Principles of knowledge representation and reasoning). The support of this pattern is rather low (0.03% of pairs).

Pattern	Top 20
IJCAI$^+$, KI$^+$, GWAI$^+$, Informatik_Spektrum$^+$, Morik_number$^-$ **(Gr(P, Morik_number$^-$)= 11)**	Katharina Morik, Wolfgang Wahlster, Bernhard Nebel, Thomas Christaller, Wolfgang Hoeppner, Jörg H. Siekmann, Günther Görz, Frank Puppe, Udo Hahn, Hans-Hellmut Nagel, Franz Baader, Christopher Habel, Bernd Neumann, Ulrich Furbach, Joachim Hertzberg
IJCAI$^+$, ICML$^+$, Machine_Learning$^+$, Knowl._Inf._Syst.$^+$, Data_Min._Knowl._Discov.$^+$, Morik_number$^-$ **(Gr(P, Morik_number$^-$)= 10.5)**	Katharina Morik, Wray L. Buntine, Kristian Kersting, Floriana Esposito, Xindong Wu, Eamonn J. Keogh, Zhi-Hua Zhou, Siegfried Nijssen, Hiroshi Motoda, João Gama, Jie Tang, Salvatore J. Stolfo, Dacheng Tao, Michael J. Pazzani, Wei Liu, Chris H. Q. Ding, Tao Li, Bin Li

Table 3.1: Emerging patterns w.r.t. Morik_number$^-$.

Pattern	Top 20
ICASSP$^+$, IJCAI$^-$, KR$^-$, KI$^-$, Morik_number$^+$ **(Gr(P,Morik_number$^+$)= 16)**	Gyula Hermann, Victor Lazzarini, Joseph Timoney, Fred Kitson, Manuel Duarte Ortigueira, Abbas Mohammadi, Riwal Lefort, Jean-Marc Boucher, Artur Przelaskowski, Kenichi Miyamoto, Emiru Tsunoo, Olaf Schreiner, Murtaza Taj, Salim Chitroub, Saptarshi Das, Ales Procházka, Amrane Houacine, Yasuyuki Ichihashi, Pablo Javier Alsina, Valeri Mladenov

Table 3.2: Emerging pattern with respect to Morik_number$^+$.

The most emerging patterns w.r.t. the Morik number that mix vertex and topological attributes are presented in Table 3.3. The first pattern is similar to the pattern of Table 3.2 and the additional topological attributes corroborate the eccentricity of the pattern relative to the graph. The second pattern brings together confirmed researchers in artificial intelligence and machine learning, who have published at Künstliche Intelligenz. They are very central in the graph and their neighborhood is not so much connected.

Where are we in Katharina's network? An interactive exploration of the patterns

After considering the patterns that maximize the growth rate w.r.t. Morik number, we now look for patterns supported by the authors of this paper. Many of these patterns involve the French-speaking conference EGC (see Table 3.4) and journals in data mining. This is due to the fact that Katharina Morik gave a keynote at EGC in 2009. The top 20 supporting authors are either French or prestigious invited speakers at this conference.

The first pattern of Table 3.5 can be interpreted thanks to a Dagstuhl seminar organized by Katharina Morik called *Local Pattern Detection*. The goal of this seminar was to bring together prominent European researchers in the field of local pattern discovery. The Data Mining and Knowledge Discovery journal is the most important one that publishes results in that area. The second one is around the seminar *Parallel Universes and Local Patterns* that was also organized by Katharina Morik and colleagues.

Pattern	Top 20
ICASSP$^+$, IJCAI$^-$, Degree$^-$, Closeness$^-$, Betweennes$^-$, NetworkConstraint$^+$, morik_number$^+$ **(Gr(P, Morik_number$^+$)= 6.5)**	Jacob Ninan, Marc Beacken, Hinrich R. Martens, Jyun-Jie Wang, William H. Haas, J. G. Cook, Lawrence J. Ziomek, José R. Nombela, T. J. Edwards, Judith G. Claassen, Shigekatsu Irie, Alberto R. Calero, Takaaki Ueda, Hisham Hassanein, Peter Strobach, Liubomire G. Iordanov, N. A. M. Verhoeckx, Guy R. L. Sohie, Sultan Mahmood, Matt Townsend
KI$^+$, Degree$^+$, Closeness$^+$, NetworkConstraint$^-$, morik_number$^-$ **(Gr(P, Morik_number$^-$)= 4.8**)	Bernhard Nebel, Katharina Morik, Deborah L. McGuinness, Mark A. Musen, Rudi Studer, Steffen Staab, Hans W. Guesgen, Bamshad Mobasher, Simon Parsons, Thorsten Joachims, Alex Waibel, Kristian Kersting, Matthias Jarke, Manuela M. Veloso, Wolfgang Nejdl, Alfred Kobsa, Virginia Dignum, Alessandro Saffiotti, Hans Uszkoreit, Antonio Krüger

Table 3.3: Emerging patterns w.r.t. the Morik number that mix vertex and topological attributes.

Pattern	Top 20
EGC$^+$, Data_Min._Knowl._Discov.$^+$, Morik_number$^-$ **(Gr(P, Morik_number$^-$)= 9)**	Katharina Morik, Bart Goethals, Céline Robardet, Didier Dubois, Michèle Sebag, Luc De Raedt, Mohammed Javeed Zaki, Einoshin Suzuki, Heikki Mannila, Jian Pei, Élisa Fromont, Toon Calders, Adriana Prado, Gilles Venturini, Szymon Jaroszewicz, João Gama, Alice Marascu, Osmar R. Zaïane, Pascal Poncelet, Jean-François Boulicaut
EGC$^+$, Knowl._Inf._Syst.$^+$, Data_Min._Knowl._Discov.$^+$, Morik_number$^-$ **(Gr(P, Morik_number$^-$)= 9)**	Katharina Morik, Bart Goethals, João Gama, Mohammed Javeed Zaki, Jian Pei, Heikki Mannila, Osmar R. Zaïane, Toon Calders, Szymon Jaroszewicz, Einoshin Suzuki, Pascal Poncelet, Christophe Rigotti, Jean-François Boulicaut, Marie-Christine Rousset, Maguelonne Teisseire, Florent Masseglia, Gregory Piatetsky-Shapiro

Table 3.4: Emerging patterns involving the French-speaking data mining conference EGC.

3.3 Mining Triggering Patterns of Topology Changes

Analyzing the dynamics of graphs makes it possible to provide additional insights about the local structure (i.e., a vertex and its neighborhood) of graphs. We strive here to elucidate the temporal relationships between the evolution of vertex attribute values and the graph structure. Let us first illustrate this idea. Figure 3.2 depicts a social network whose users are linked if they mutually follow their blogs and evolves over 6 timestamps. Attributes *a*, *b*, and *c* denote the number of status updates, positive opinions sent to others and negative opinions received from other users. At each timestamp, each vertex is also naturally provided with topological properties giving his role in the current graph (centrality measures, clustering coefficient, etc.). To ease reading, only vertex degrees *deg* are represented. From this dynamic attributed graph, we consider that an attribute (or a topological property) strongly varies for a given vertex if the absolute difference between its current value and the one at the previous timestamp is at least of two. These variations are represented by dotted lines. It makes it possible to represent temporal relationships

Pattern	Top 20
LocalPatternDetection^{+}, Data_Min._Knowl._Discov.$^{+}$, Morik_number^{-} , **(Gr(P, Morik_number^{-})= 9.7)**	Katharina Morik, Stefan Rüping, Francesco Bonchi, Niall M. Adams, Marko Grobelnik, David J. Hand, Dunja Mladenic, Frank Höppner, Saso Dzeroski, Einoshin Suzuki, Nada Lavrac, Jean-François Boulicaut, Myra Spiliopoulou, Ruggero G. Pensa, Johannes Fürnkranz, Filip Zelezny
Parallel_Universes_and_Local_Patterns^{+}, Morik_number^{-} **(Gr(P, Morik_number^{-})= 6.4)**	Katharina Morik, Arno Siebes, Michael R. Berthold, Michael Wurst, David J. Hand, Bernd Wiswedel, Frank Höppner, Emmanuel Müller, Élisa Fromont, Claus Weihs, Niall M. Adams, Mirko Böttcher, Ralph Krieger, Bruno Crémilleux, Ira Assent, Marie-Odile Cordier, Thomas Seidl, Heike Trautmann, Rene Quiniou, Arnaud Soulet

Table 3.5: Patterns related to Dagstuhl seminars.

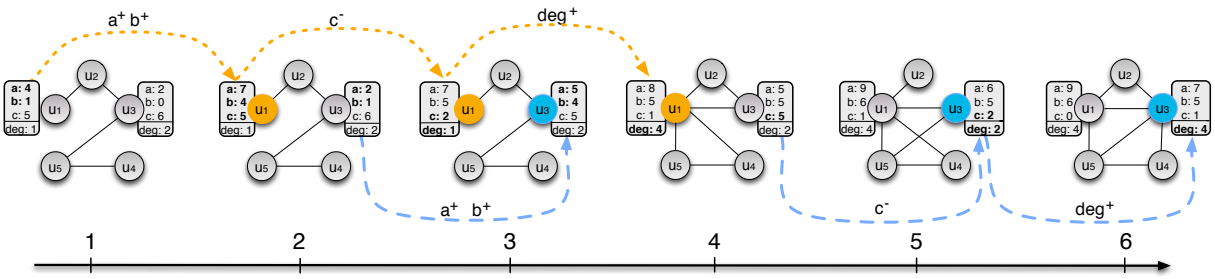


Figure 3.2: A dynamic attributed graph on 6 timestamps entailing the triggering pattern $\langle\{a^+,b^+\},\{c^-\},\{deg^+\}\rangle$. To ease reading, only attributes for vertices u_1 and u_3 are shown. Colors and bold attribute values indicate strong variations for supporting vertices.

between the vertex attributes and their topological properties as a *triggering pattern*: a sequence of variations that is supported by a given proportion of vertices. In the example, such a sequence is denoted as $\langle\{a^+,b^+\},\{c^-\},\{deg^+\}\rangle$ and is supported by the two vertices u_1 and u_3. It can be interpreted as *updating his status more often while giving positive opinions about others and then receiving less negative opinions from the others is often followed by an increase of user's popularity.*

We define the problem of finding temporal relationships between the vertex attributes and their topological properties as the *triggering pattern mining problem*. A given (possibly oriented) dynamic attributed graph is mapped into sequences of variations of both vertex attributes and topological properties. Each sequence is thus related to a vertex and describes its history. We consider this information to identify the sequences of attribute variations that are generally followed by a topological variation. To this end, we use the notion of emerging patterns in the framework of supervised descriptive rule discovery [Dong and Li, 1999, Novak *et al.*, 2009] where class labels are given by topological variations. To assist an in-depth analysis, we propose to examine the vertices that supports such an emerging subsequence, by considering several aspects:

How many vertices support the subsequence? Are those vertices highly connected? Do they convey

a high diffusion potential? Are the variations synchronized or spread through time as in Figure 3.2? Therefore, during the extraction, the mining algorithm has to consider not only the supporting vertices of the subsequence (e.g. [Yan *et al.*, 2003]), but also the structure of the subgraph induced by them. Additionally, we enhance patterns with semantics, through the definition of several measures and constraints. Furthermore, we introduce a particular representation of sequential patterns (*prefix-closed patterns*), that are the best non-redundant and most discriminant patterns highlighting temporal relationships.

3.3.1 Mining Topological Changes

Let us repeat some definitions introduced previously. A dynamic attributed graph is a sequence of attributed graphs where each vertex takes a value for all the different attributes. Values vary in time as well as edges that may appear or disappear.

Definition 3.4 (Dynamic attributed graph)**.** *Let $\mathcal{G} = \{G_1, \ldots, G_t\}$ be a sequence of t static attributed graphs $G_i = (V, E_i, F)$ with $T = \{1, \ldots, t\}$ the set of timestamps, V the set of vertices, E_i the set of edges that connect vertices of V at time $i \in T$ ($E_i \subseteq V \times V$) and F the set of numerical attributes that map each vertex-time pair to a real value: $\forall f \in F$, $f : V \times T \to \mathbb{R}$.*

The structure of each static graph can be characterized by some topological measures [Freeman, 1977, Leskovec and Sosič, 2014] that convey important information on the connectivity of the vertices within the graph at different granularity levels. We denote by M the set of topological measures m, with $m : V \times T \to \mathbb{R}$.

Example 3.1. *Figure 3.2 illustrates a dynamic attributed graph $\mathcal{G} = \{G_1, ..., G_6\}$, with $t = 6$ timestamps. Each graph $G_i = (V, E_i, F)$ shares the set of vertices $V = \{u_1, ..., u_5\}$ and the set of attributes $F = \{a, b, c\}$. The fact that the vertex u_3 takes the value 6 for the attribute c at timestamp 2 is written $c(u_3, 2) = 6$. The topological measure used in this example is the vertex degree: $M = \{deg\}$ and $deg(u_3, 6) = 4$.*

A change in the structure of the dynamic graph is observed through a strong variation on some topological measures. Our hypothesis is that those changes can be the consequences of strong variations on vertex attribute values. Let $D = F \cup M$ be the set of vertex descriptors that are either attributes or topological measures. Our goal is to highlight how variations of descriptor values of a vertex can later impact on its connectivity, that is to say variations on descriptions of D followed by variations on measures of M.

To characterize the vertex descriptor variations, an appropriate discretization has to be used. As such, we can represent, for each vertex, its behavior with a sequence of descriptor values variations between any two consecutive time stamps. This procedure has to be wisely chosen with respect to the goals and the properties of the dynamic graph. This choice has no impact on our problem definition: to ease reading of this section we consider the naive discretization of Example 3.2 that turns any attribute value into an element of $S = \{+, -, \emptyset\}$ to denote increase, decrease or no variation.

Definition 3.5 (Discretization function)**.** *Let $discr : V \times D \times T \to S$ be a discretization function with S a set of variation symbols. In the following, we call an element (d, s) from $D \times S$ a descriptor variation, denoted d^s.*

Example 3.2. *Let $S = \{+, -, \emptyset\}$ be a simple discretization function defined as, with $v \in V$, $d \in D$, i an index of T:*

$$discr(v, d, i) = \begin{cases} + & \text{if } d(v,i) - d(v,i-1) \geq 2 \text{ and } i > 1 \\ - & \text{if } d(v,i) - d(v,i-1) \leq -2 \text{ and } i > 1 \\ \emptyset & \text{otherwise} \end{cases}$$

The dynamic graph can thus be viewed as a set of vertex descriptive sequences defined as follows:

Definition 3.6 (Vertex descriptive sequence)**.** *The set of all variations for a vertex v at time i is a set* $vars(v,i) = \{d^{discr(v,d,i)}, \forall d \in D\}$. *A vertex v is described by a sequence* $\delta(v) = \langle vars(v,2), \ldots, vars(v,t)\rangle$. *We note* $\Delta = \{\delta(v) \mid v \in V\}$ *the set of all sequences.*

Example 3.3. *The graph in Figure 3.2 is transformed into*

$$\begin{aligned}
\Delta &= \{\\
\delta(u_1) &= \langle\{a^+,b^+,c^\emptyset,deg^\emptyset\},\{a^\emptyset,b^\emptyset,c^-,deg^\emptyset\},\{a^\emptyset,b^\emptyset,c^\emptyset,deg^+\},\{a^\emptyset,b^\emptyset,c^\emptyset,deg^\emptyset\},\{a^\emptyset,b^\emptyset,c^\emptyset,deg^\emptyset\}\rangle,\\
\delta(u_3) &= \langle\{a^\emptyset,b^\emptyset,c^\emptyset,deg^\emptyset\},\{a^+,b^+,c^\emptyset,deg^\emptyset\},\{a^\emptyset,b^\emptyset,c^\emptyset,deg^\emptyset\},\{a^\emptyset,b^\emptyset,c^-,deg^\emptyset\},\{a^\emptyset,b^\emptyset,c^\emptyset,deg^+\}\rangle\\
&\ \}
\end{aligned}$$

For the sake of simplicity, we drop any description variation with the symbol $(.)^\emptyset$*, i.e. we exclude the description of attribute that has no variation between two consecutive time stamps. We obtain for the vertex* u_1 *the descriptive sequence* $\delta(u_1) = \langle\{a^+,b^+\},\{c^-\},\{deg^+\}\rangle$. *It means that we observe, for this vertex, an increase of the values of attribute a and b simultaneously, then an decrease of the value of c and finally an increase of the degree of the vertex (deg). Note that in this sequential representation, we do not encode the time stamps of the variations, but simply keep their ordering of appearance. This allows to find later patterns that generalize both (almost) synchronous and asynchronous vertex behaviors. The notion of synchronicity is formalized in the following.*

Now that we properly encode the vertices temporal behavior with a sequence database, we are looking for possible explanations of a topological variation. Such an explanation is given by a sub-sequence (or sequential pattern) of descriptors variations L that happen before the considered topological variation R. It follows that a *triggering pattern* is formalized as a sequence $\langle L,R\rangle$ which is supported by a certain number of vertices. It makes it possible to represent temporal dependencies between D and M, and the confidence or strength of such dependency has to be high.

Definition 3.7 (Triggering pattern)**.** *A triggering pattern is a sequence* $P = \langle L,R\rangle$ *where L is a sequence of sets of descriptor variations* $L = \langle X_1,\ldots,X_k\rangle$ *with* $\forall j \leq k$, $X_j \subseteq (D \times S)$, *and R a single topological variation,* $R \in (M \times S)$.

Definition 3.8 (Triggering pattern support)**.** *We say that* $Q_1 = \langle X_1,\ldots,X_k\rangle$ *is a sub-sequence of* $Q_2 = \langle Y_1,\ldots,Y_\ell\rangle$, $Q_1 \preceq Q_2$, *if there exists* $\langle i_1,\ldots,i_k\rangle$ *a strictly increasing sequence in* $\mathbb{N}$ *such that* $X_j \subseteq Y_{i_j}, \forall j = 1,\ldots,k$. *The support of a pattern* $P = \langle L,R\rangle$ *is the set of vertices for which P is a sub-sequence of their descriptive sequence:* $\text{SUPP}(P,\Delta) = \{v \in V \mid P \preceq \delta(v)\}$.

Example 3.4. *With* $P_1 = \langle\{a^+\},\{deg^+\}\rangle$ *and* $P_2 = \langle\{a^+,b^+\},\{c^-\},\{deg^+\}\rangle$, *it follows that* $\text{SUPP}(P_1,\Delta) = \{u_1,u_3\}$ *and* $\text{SUPP}(P_2,\Delta) = \{u_1,u_3\}$.

We adapt a well known measure, the growth-rate [8] [Dong and Li, 1999, Novak *et al.*, 2009], to characterize patterns $\langle L,R\rangle$ whose left part L strongly discriminates the topological variation R.

Definition 3.9 (Triggering pattern growth rate)**.** *Let* $P = \langle L,R\rangle$, *we denote by* $\Delta^R \subseteq \Delta$ *the set of vertex descriptive sequences that contain R. The growth rate of P is given by:*

$$\text{GR}(P,\Delta^R) = \frac{|\text{SUPP}(L,\Delta^R)|}{|\Delta^R|} \times \frac{|\Delta \setminus \Delta^R|}{|\text{SUPP}(L,\Delta \setminus \Delta^R)|}$$

[8] The term growth-rate may be misleading: it is not related to time, but to the appearance of a target attribute in a sub group of objects w.r.t. the rest of the database.

The triggering pattern mining problem. Given a dynamic attributed graph $\mathcal{G}$, a minimum growth rate threshold *minGR* and a minimum support threshold *minSup*, the problem is to find all triggering patterns $P = \langle L, R \rangle$ such that $|\text{SUPP}(P, \Delta)| \geq minSup$ and $\text{GR}(P, \Delta^R) \geq minGR$. The support measure is defined by the vertices that satisfy the pattern, while the growth rate gives the discriminating power of the sequence variations L to explain a topological change R.

3.3.2 Non Redundant Triggering Patterns With Semantics

The triggering pattern problem is actually defined in the previous section as a well known frequent sequential pattern mining problem. After a data transformation into a database of (vertex descriptive) sequences, the goal is to find all frequent patterns with the particularity that they must finish with a particular item (a topological change) and being provided with a growth rate higher than a given threshold. However, this comes with several problems and limitations that we develop now and propose solutions to.

Mining covering triggering patterns

The first main problem is that the support and the growth rate are not enough for producing intelligible patterns: additional measures are required. We propose several examples of meaningful measures. The most important one is the coverage of a pattern (vertex cover [Cormen *et al.*, 2009]): given a graph, the coverage counts the proportion of nodes that can be directly reached from the nodes of the support. The coverage can be generalized to the nodes reachable at a distance n from nodes of the support. When $n = 0$, the coverage is exactly the support. The coverage provides better insights than the support, since it allows to express how the nodes of the support are rooted into a global behavior. However, triggering patterns appear at different time stamps in their supporting descriptive sequences and the set of connected vertices cannot be defined in the dynamic graph: the coverage of a pattern has to be defined up to an aggregated static graph $\mathcal{G}_{aggr}$ that sums-up the connectivity of each vertex along time. In the following we use $\mathcal{G}_{aggr} = (V, \bigcup_{i=1}^{t} E_i)$.

Definition 3.10 (Coverage of a triggering pattern)**.** *Let $\mathcal{G}_{aggr} = (V, E_{aggr})$ be an aggregated graph of the dynamic graph $\mathcal{G}$. The coverage of a pattern P is defined by:* $\text{COV}(P, \Delta, \mathcal{G}_{aggr}) = \text{SUPP}(P, \Delta) \cup \{v \in V \mid \exists w \in \text{SUPP}(P, \Delta)\, s.t.\, (w, v) \in E_{aggr}\}$.

Example 3.5. *For $P = \langle \{a^+, b^+\}, \{c^-\}, \{deg^+\} \rangle$, we have $cov(P, \Delta, \mathcal{G}_{aggr}) = V$.*

The coverage gathers the vertices that have a geodesic distance of 1 with at least one vertex of the pattern support. This measure can be generalized to any geodesic distance value n as follows:

Definition 3.11 (n-coverage of a triggering pattern)**.** *The n-coverage of a pattern P is recursively defined as* $\text{COV}^n(P, \Delta, \mathcal{G}_{aggr})$ *with* $\text{COV}^0(P, \Delta, \mathcal{G}_{aggr}) = \text{SUPP}(P, \Delta)$ *and* $\text{COV}^i(P, \Delta, \mathcal{G}_{aggr}) = \text{COV}^{i-1}(P, \Delta, \mathcal{G}_{aggr}) \cup \{v \in V \mid \exists w \in \text{COV}^{i-1}(P, \Delta, \mathcal{G}_{aggr})\ s.t.\, (w, v) \in E_{aggr}\}$.

The n-coverage measure is important for two reasons: (1) It conveys *per se* insights on the possibilities for dissemination of a pattern; (2) It makes it possible the definition of additional interestingness measures to focus on specific kinds of triggering patterns. Since the coverage is a generalization of the support,we propose to mine *covering triggering patterns*, i.e. patterns whose coverage is higher than a given threshold *minCov* and provided with a growth rate higher than *minGR*.

Additional measures that convey semantics and constraints

The definition of coverage enables to define several measures and constraints on the pattern domain. This is interesting for two reasons: firstly, triggering patterns are provided with more semantics, secondly, the

expert can specify some constraints on the patterns to drive her analysis and partly face the classical pattern flooding problem. Indeed, the n-coverage of a triggering pattern can be viewed as a set of nodes inducing a subgraph of $\mathscr{G}_{aggr}$. Constraining such subgraphs makes it possible to mine topological changes whose coverage has a specific structure. We can distinguish two types of constraints, those that rely on the vertices (see Definition 3.12) and the others that restrict the structure of the subgraph (see Definition 3.13).

Definition 3.12 (Constraint on vertices)**.** *Let μ be a centrality measure of a vertex v in the aggregated graph $\mathscr{G}_{aggr}$, such as:*

- *The vertex degree: $deg(v,\mathscr{G}_{aggr}) = |\{u \in V, \{u,v\} \in E_{aggr}\}|$*
- *Clustering coefficient: $\text{CLUST}(v,\mathscr{G}_{aggr}) = \frac{2|\{\{u,w\}\in E_{aggr}, \{u,v\}\in E_{aggr} \wedge \{v,w\}\in E_{aggr}\}|}{deg(v)(deg(v)-1)}$*
- *Closeness centrality: $\text{CLOSE}(v,\mathscr{G}_{aggr}) = \frac{n}{\sum_{u\in V} |shortest\_path_{\mathscr{G}_{aggr}}(u,v)|}$*
- *Betweeness centrality: $\text{BETW}(v,\mathscr{G}_{aggr}) = \sum_{u,w} \delta_{v\in shortest\_path_{\mathscr{G}_{aggr}}(u,w)}$*

The constraint on the vertices of the n-coverage of a triggering pattern is as follows:

$$\mathscr{C}(P,\Delta,\mathscr{G}_{aggr},\mu,m,n) \text{ iff } \forall v \in \text{COV}^n(P,\Delta) \mid \mu(v,\mathscr{G}_{aggr})\} \ \theta \ m$$

with $\theta \in \{\leq,<,\geq,>\}$.

Definition 3.13 (Constraint on the subgraph structure)**.** *Let ρ be a measure on a subgraph $G(V)$ induced by the set of vertices V in the aggregated graph $\mathscr{G}_{aggr}$. Examples of such measure are:*

- *Density: $density(V,\mathscr{G}_{aggr}) = \frac{|V\times V \cap E_{aggr}|}{|V|\times(|V|-1)}$*
- *Diameter: let $d_{G(V)}(u,v)$ be the shortest path length between the vertices u and v in $G(V)$. The diameter of $G(V)$ is thus defined by*

$$diam(V,G_{aggr}) \equiv \max_{u,v\in V} d_{G(V)}(u,v)$$

The constraint on the structure of the n-coverage of a triggering pattern is as follows:

$$\mathscr{C}(P,\Delta,\mathscr{G}_{aggr},\rho,m,n) \text{ iff } \rho(\text{COV}^n(P,\Delta),\mathscr{G}_{aggr}) \ \theta \ m$$

with $\theta \in \{\leq,<,\geq,>\}$.

Finally, remember that a pattern may appear at different time stamps in the vertex descriptive sequences that make its support. This is the case in the example of Figure 3.2: the pattern $\langle\{a^+,b^+\},\{c^-\},\{deg^+\}\rangle$ is supported by both vertices u_1 and u_3. However, these appearances are not synchronized. We propose a measure that allows to quantify the notion of synchronicity.

Definition 3.14 (Synchronicity)**.** *Let $P = \langle L,R\rangle$ be a triggering pattern supported by the set of vertices $SUPP(P) = \{u_1,...,u_n\}$. We introduce the set $A = \{a_1,a_2,...,a_n\}$ where $a_i \in T$ is the first time stamp where P occurs in the vertex descriptive sequence $\delta(u_i)$. Similarly, we introduce $B = \{b_1,b_2,...,b_n\} \subseteq T$ where b_i is the index of R in the first occurrence of P in $\delta(u_i)$. The synchronicity measure is given by:*

$$sync(P) = \frac{avg(\{|a_i - a_j|\}_{i,j\in 1...n, i\neq j}) + avg(\{|b_i - b_j|\}_{i,j\in 1...n, i\neq j})}{avg(\{|(b_i - a_i) - (b_j - a_j)|\}_{i,j\in 1...n, i\neq j})}$$

where $avg(.)$ is the mean function. The numerator is 0 if the time stamps of the supporting sequences are fully synchronous. The denominator normalizes w.r.t. the average lengths.

Limiting redundancy with closed triggering patterns

To limit more strongly the problem of pattern flooding, we propose here to define a condensed representation of triggering patterns, to avoid to output them all, while not loosing the information they convey. To this end, we adapt the notion of closed sequential patterns [Yan *et al.*, 2003, Wang *et al.*, 2007] to the case of triggering patterns. This solution is not straightforward and requires to develop a new algorithm called `TRIGAT`.

Classical closed sequential patterns are indeed chosen to limit redundancy: any sequential pattern is closed w.r.t. the support if it does not exist a super-sequence with the same support [Yan *et al.*, 2003]. Furthermore, the growth-rate is always maximized by those closed patterns [Plantevit and Crémilleux, 2009] and most of the constraints given above are necessary to be evaluated on closed patterns only (the case of the anti-monotone constraints, defined hereafter). We show how to define closed triggering patterns w.r.t. to the coverage measure that maximize the *GR* measure. As triggering patterns are particular sequences ending on a topological variation, one cannot trivially apply any closed sequential pattern mining algorithm. Instead, we prove several properties that allow to define closed triggering patterns as prefix-closed patterns, and that the latter can be defined from closed sequential patterns. Interestingly, the coverage measure still follows an anti-monotone behavior.

Definition 3.15 (Prefix-closed pattern)**.** *We say that a pattern $P = \langle L_P, R\rangle$ is prefix-closed w.r.t.* SUPP *iff $\nexists Q = \langle L_Q, R\rangle$ such that $L_P \preceq L_Q$ and* $\text{SUPP}(P) = \text{SUPP}(Q)$*: P cannot be extended on its left part L_P without changing its support.*

In Example 3.4, P_2 is a closed triggering pattern, i.e. a prefix-closed sequential pattern, while P_1 is not.

Property 3.3. *Let $P = \langle L_P, R\rangle$ and $Q = \langle L_Q, R\rangle$ be two triggering patterns. If $L_P \preceq L_Q$ and* $\text{SUPP}(L_P, \Delta^R) = \text{SUPP}(L_Q, \Delta^R)$*, then* $\text{GR}(L_P, \Delta^R) \leq \text{GR}(L_Q, \Delta^R)$.

Proof. We have $\text{SUPP}(L_P, \Delta^R) = \text{SUPP}(L_Q, \Delta^R)$. As $L_P \preceq L_Q$, we obtain that $\text{SUPP}(L_P, \Delta) \supseteq \text{SUPP}(L_Q, \Delta)$ thanks to the anti-monotonicity of the support. By definition 3.9, we conclude that $\text{GR}(L_P, \Delta^R) \leq \text{GR}(L_Q, \Delta^R)$. □ □

Property 3.3 makes it possible to only focus on prefix-closed patterns as they maximize the growth-rate. To extract prefix-closed patterns, we exploit the property asserting that prefix-closed patterns can be retrieved from closed patterns.

Property 3.4. *For any prefix-closed pattern P, there exists a closed pattern Q such that $P \preceq Q$ and* $\text{SUPP}(P) = \text{SUPP}(Q)$.

Proof. Prefix-closed patterns cannot be extended to the left without changing their support. However, they can be extended to the right while preserving their support. Thus, a closed pattern Q can be seen as a simultaneously prefix and suffix-closed: Q is included in a closed pattern. □ □

Property 3.5 states that COV is anti-monotone with respect to $\preceq$ (directly from Definition 3.10). Since by definition we have that for any pattern P, $\text{SUPP}(P) \subseteq \text{COV}(P)$, one can choose to enumerate the closed triggering patterns based either on support or coverage

Property 3.5. *The coverage is anti-monotone w.r.t. $\preceq$, i.e., $\forall P \preceq Q, \text{COV}(P, \Delta, \mathscr{G}_{aggr}) \supseteq \text{COV}(Q, \Delta, \mathscr{G}_{aggr})$.*

The same property applies for COV^n. Constraints μ and ρ presented in the previous subsection however, can be or not anti-monotone. When this is the case, we can use a conjunction of several constraints as a main constraint. Indeed, a conjunction of anti-monotone constraints is anti-monotone and it can be also used to prune the search space in the algorithm [Ng *et al.*, 1998]. When a constraint is not anti-monotone, we check it in a post-processing by removing afterwards the patterns that do not satisfies it.

3.3.3 Overview of TRIGAT Algorithm

TRIGAT mines the complete and correct collection of all coverage-based closed triggering patterns $P = \langle L, R \rangle$ such that $|\text{COV}^n(P, \Delta)| \geq \textit{minCov}$ and $\text{GR}(P, \Delta^R) \geq \textit{minGR}$. This algorithm takes benefits from the properties previously introduced to efficiently prune the pattern search space while pushing anti-monotone constraints among which the coverage constraint. It first builds each vertex descriptive sequence of the dynamic graph. Covering 1-item sequences, that are sequences of a single descriptor variation satisfying the coverage constraint, are computed in one scan on Δ and uncovering items are removed for any sequence of Δ. Then, a subroutine (TRIGAT_ENUM) is called. It achieves a depth-first search on a given prefix sequence using a pattern-growth approach that works on *projected databases* according to a prefix sequence s, denoted $\Delta_{|s}$ which returns all the suffixes of s in Δ [Yan *et al.*, 2003].

TRIGAT_ENUM exploits the pruning techniques based on closed patterns while pushing the coverage constraint. We use the *early termination by equivalence* pruning technique, first proposed in the CloSpan algorithm [Yan *et al.*, 2003][9]. The coverage is computed w.r.t. the union of the adjacency lists of vertices whose projected sequences along the prefix sequence s is not empty (i.e., $s \preceq \delta(v)$). Finally, as prefix sequences can grow either by adding a single descriptor variation in the last set of s, or by adding a new set made of this single descriptor variation at the end of the sequence, these two patterns are recursively considered.

Finally, the subroutine TRIGAT_ENUM returns all covering sequences of prefix $\langle s \rangle$. TRIGAT ends with a post-processing of C to retain only closed patterns. To avoid expensive tests that aim at comparing each sequence of C with other sequences in C, we adopt the fast subsumption checking algorithm [Zaki and Hsiao, 2002] using a hashmap with a sparse key distribution. Closed triggering patterns are then built from C. Prefix-closed sequences ending by a topological variation are built from closed sequences, and the growth-rate is computed in negligible time. A more detailed description of the algorithm is given in [Kaytoue *et al.*, 2015].

3.3.4 Triggering Patterns in Real-World Datasets

We report here qualitative results on a co-authorship network built from DBLP. The Digital Bibliography & Library Project[10] covers an important part of the computer science bibliography. All references published between 1990 and 2010 by 2,723 authors (recording more than 10 publications) are elected among 43 conferences/journals. Two additional attributes sum the publications resp. in *conferences* and *journals*. The dynamic attributed graph, with authors as vertices and edges as co-authoring, entails 9 time stamps of 5 years half-overlapping intervals ([1990-1994],[1992-1996],...,[2006-2010]).

Before extracting triggering patterns from this dynamic graph, we need to characterize the strength of the attribute value variations as stated in Definition 3.5. As we are interested in significant variations of vertex attribute values w.r.t. the proper history of each vertex between consecutive timestamps, we consider, for each attribute $d \in D$ and vertex $v \in V$, the set $\{d(v,i) - d(v,i-1),\ 1 < i \leq t\}$ of time derivatives of $d(v)$. Such a discrete set of values can be characterized by its mean $\overline{d(v)}$, that refers to the central value of the set, and its standard-deviation $std(d(v))$ that gives a hint to the homogeneity or heterogeneity of the set. Based on these values, we devised the following discretization function $discr_1$:

[9] This enables to avoid to consider any prefix sequence s' having an equivalent projected database than a sequence s discovered before, i.e., $\Delta_{|s} = \Delta_{|s'}$. Two cases are possible. Either $s' \prec s$ (backward sub-pattern) or $s \prec s'$ (backward super-pattern). In case of backward sub-pattern, the exploration of s' and its descendants is stopped. In case of backward super-pattern, the descendant of s are transplanted to s' instead of exploring an already scanned projected database.

[10] http://www.informatik.uni-trier.de/~ley/db/

$$discr_1(v,d,i) = \begin{cases} ++ & \text{if } d(v,i) - d(v,i-1) \geq \overline{d(v)} + 3 \cdot std(d(v)) \\ + & \text{if } d(v,i) - d(v,i-1) \geq \overline{d(v)} + std(d(v)) \\ - & \text{if } d(v,i) - d(v,i-1) \leq \overline{d(v)} - std(d(v)) \\ -- & \text{if } d(v,i) - d(v,i-1) \leq \overline{d(v)} - 3 \cdot std(d(v)) \\ \emptyset & \text{otherwise} \end{cases}$$

Table 3.6 reports the main characteristics of the resulting dynamic graph.

Number of vertices $\|V\|$	2773
Number of attributes $\|F\|$	45
Number of timestamps $\|T\|$	9
Number of descriptor variations	360
Average number of descriptor variations per vertex	34.4
Average number of timestamps per vertex having more than one variation	6.6
Average degree $\overline{\deg}_{sum}$	14.7
Density $density_{sum}$	0.005

Table 3.6: Main characteristics of the dynamic graph built from DBLP.

We now present some co-evolution patterns $\langle L,R \rangle$, written $L \rightarrow R$ to ease reading, and discuss their usefulness.

We set the minimal coverage threshold to 10%. One may argue that this threshold is too high when dealing with real-world data. However, we have shown that the support of a pattern can be much more smaller than its coverage: it may lead to the extraction of patterns with a support less than 1%. The mining task (including pre/post processing) takes 307 seconds and returned 3,261 patterns. The growth rate distribution for those patterns ranges from 1 to 87 ($minGr = 1$), with a mean value of 4.5 and standard deviation of 3.6. Table 3.7 gives the top-10 patterns with a growth rate greater than 50.

Consider now a measure given by the ratio $\alpha(P,\Delta) = \frac{|\text{COV}(P,\Delta,\mathcal{G}_{aggr})|}{|\text{SUPP}(P,\Delta)|} \in [1,|V|]$, which allows to distinguish the patterns supported by a group of isolated vertices (values close to 1) to the ones supported by very connected vertices (much higher values than 1). The support of each pattern given in Table 3.7 is in average around 1% and the coverage around 22%, thus a high α value.

These patterns explicit the conferences or journal venues that strongly impact and/or explain one's collaborations or authority in the DBLP network. For example, the first pattern tells us about the impact of publishing more in the IEEE Transactions on Knowledge and Data Engineering. Figure 3.3 (left) gives the subgraph induced by the support (labeled in red), and the coverage: the high density reveals a community aspect. In contrast, the graph of pattern 8 is much more sparser (Figure 3.3 right) which may reflect that publishing at VLDB triggers a positive variation of the author degree in the graph, for authors that cover well the graph (high coverage) but do not collaborate together (low density). As such, one can be interested in querying about a specific conference. Let us continue with the example of the VLDB conference: we wonder if publishing in VLDB, and probably presenting there, could be an interesting start for increasing one's degree, i.e. making collaborations. We query the top-k patterns w.r.t. growth rate that involve VLDB in the left hand-side and the degree in the right-hand side. All the patterns returned have a growth rate higher than 20. Consider now the IEEE ICDM conference: the top pattern that involves it has a growth rate of 5 and tells us that publishing at ICDM may help to be more central in the co-authoring

Rank	Pattern	Support	Coverage	Growth rate	α
1	$\{closeness_1^-\},\{IEEETransKnowlDtEn^+\},\{numCliques_1^+\} \rightarrow \{numCliques_1^-\}$	15	578	87.4	38.5
2	$\{clustering_1^{++},degree_1^{++}\},\{Journal^{++},eigenvector_2^{++}\} \rightarrow \{eigenvector_3^{++}\}$	31	546	71.6	17.6
3	$\{ICDE^+,numCliques_1^+\} \rightarrow \{numCliques_1^-\}$	22	606	64.1	27
4	$\{eigenvector_1^{++},degree_1^{++}\},\{VLDB^{++},degree_2^{++}\} \rightarrow \{degree_3^{++}\}$	29	580	63.8	20
5	$\{eigenvector_1^{++},clustering_1^{++}\},\{Journal^{++},eigenvector_2^{++}\} \rightarrow \{eigenvector_3^{++}\}$	36	619	59.3	17.19
6	$\{ACMTransDBSys^+\},\{numCliques_1^+\} \rightarrow \{numCliques_1^-\}$	20	547	58.3	27.35
7	$\{eigenvector_1^{++}\},\{Journal^{++},betweennes_3^{++}\} \rightarrow \{betweennes_4^{++}\}$	20	587	58.4	29.35
8	$\{eigenvector_1^{++}\},\{VLDB^{++},degree_2^{++}\} \rightarrow \{degree_3^{++}\}$	30	623	56.47	20.7
9	$\{SIGMOD^-\},\{numCliques_1^+\} \rightarrow \{numCliques_1^-\}$	32	754	53.3	23.56
10	$\{closeness_1^-\},\{SIGMOD^-\},\{numCliques_1^+\} \rightarrow \{numCliques_1^-\}$	18	552	52.4	30.6

Table 3.7: Top ten patterns of the DBLP co-authorship network according to the growth rate value.

graph. The lower growth rate (5) when comparing with VLDB (20) can be explained by the fact that ICDM is younger than VLDB.

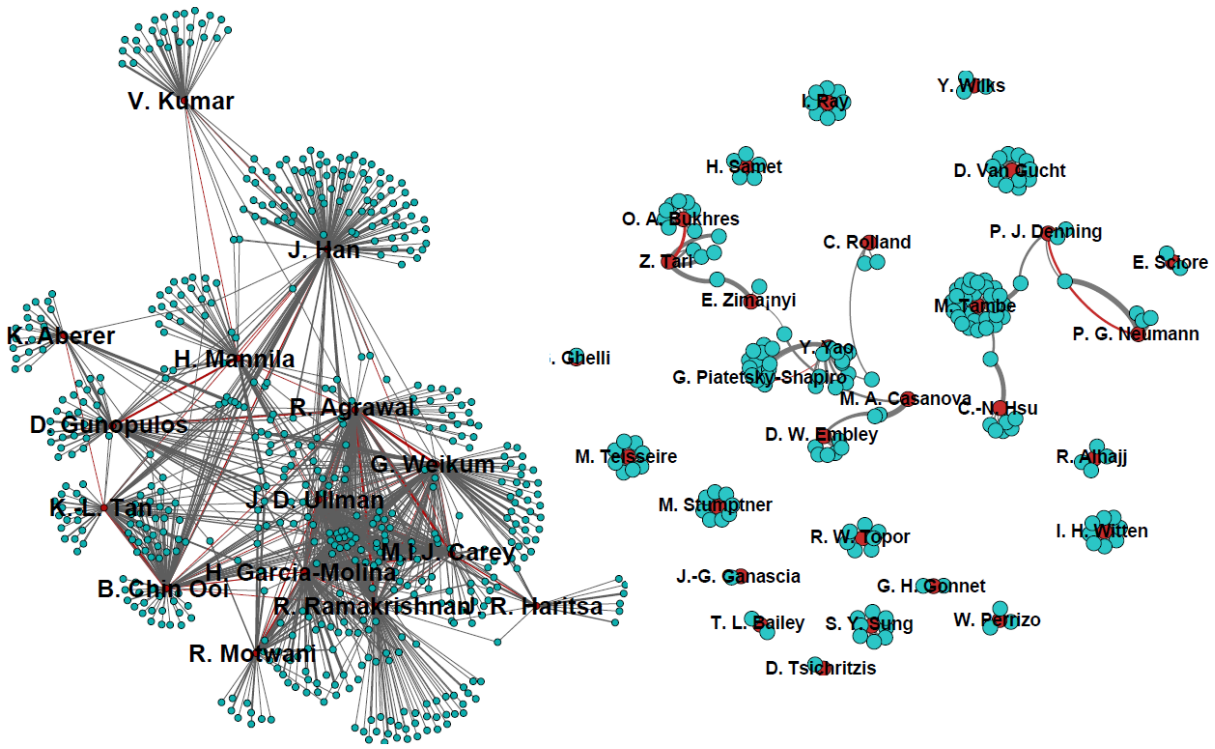
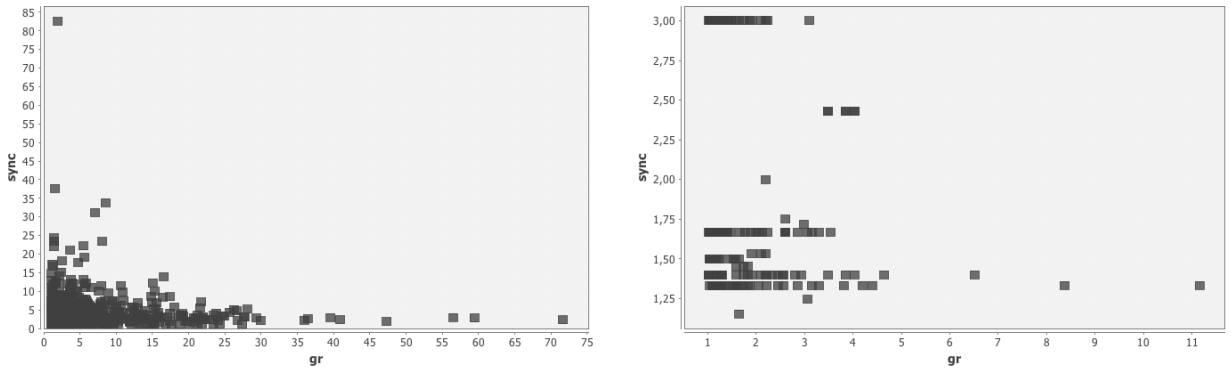


Figure 3.3: The patterns ranked 1^{st} (left) and 8^{th} (right) from the DBLP network w.r.t. growth rate. Labeled vertex (in red) represent the support, while all vertices form the coverage.

Synchronized and non synchronized patterns. TRIGAT is able to identify *synchronous* triggering patterns, whose vertex attributes change at the same timestamps in the supporting sequences of Δ. This phenomenon was observed in transportation networks (e.g., RITA network that depicts the US domestic flights). Such a network is more subject to external factors (e.g., hurricanes, terrorist attacks) than a co-citation network. As a result, *asynchronous* triggering patterns – whose vertex attributes change at different timestamps in the supporting sequences – are much more present in DBLP network. It makes senses since such a network depicts scientific careers of researchers with different experiences (PhD student, junior scientist, professor, etc.). This can be observed on Figure 3.4: the RITA patterns (with a growth rate higher than 1) have a synchronicity between 1 and 3 while it is more spread for DBLP (from 0 to 84).



(a) DBLP (b) RITA1

Figure 3.4: Synchonicity measure (sync Y-axis) vs. grow rate (gr X-axis) for patterns extracted from the DBLP dataset with $minCov = 20\%$ (left) and the RITA dataset with $minCov = 20\%$ (right).

3.4 Conclusion

The topological patterns and the triggering patterns allow new and original graph analyses. This analysis can be performed on (i) “static" graphs (i.e., aggregated on given period), looking for co-variations between vertex attributes and attributes that depict the role of the vertices within the graph, or (ii) on dynamic graphs in which triggering patterns aim to highlight sequences of events (i.e., attribute variations) that impact some topoligical properties of the vertices that support these sequences. These patterns with the definition of interestingness measures enable original attributed graph study.

This work can be followed up in many directions. In my opinion, the most promising one is the study of causality in such context to identify patterns of interest and avoid misleading interpretation. Indeed, users tend to see causality within the patterns whereas it is far from being proven. Of course, causality [Pearl, 2009] is itself a research question that has been gaining much attention in the pattern mining community [Budhathoki and Vreeken, 2018b, Budhathoki and Vreeken, 2018a] but we have to tackle it if we want to provide actionable – and non misleading – insights on link between vertex attributes and graph structure.

Chapter 4

Toward Exceptional Model Mining in Attributed Graphs

Contents

4.1 Introduction

In the previous chapters, we introduced some pattern domains to analyze (dynamic) attributed graphs and to provide new insights from such data. We defined some interestingness measures and their associated constraints to identify some patterns of interest according to a specific target/dimension (e.g., vertex specificity, emerging sequences). More generally, finding descriptions of subpopulations for which the distribution of a *single* pre-defined target value is significantly different from the distribution in the whole data is a problem that has been widely studied in *subgroup discovery* (SD) [Klosgen, 1996, Wrobel, 1997]. Many other data mining tasks have similar goals as SD [Novak *et al.*, 2009], e.g., emerging patterns [Dong and Li, 1999], significant rules [Terada *et al.*, 2013], contrast sets [Bay and Pazzani, 2001] or classification association rules [Liu *et al.*, 1998]. However, among these different tasks, SD is known as the most generic one, especially SD is agnostic of the data and the pattern domain. For instance, subgroups can be defined with conjunction of conditions on symbolic [Lavrac *et al.*, 2004] or numeric attributes [Grosskreutz and Rüping, 2009, Atzmueller and Puppe, 2006] as well as sequences [Grosskreutz *et al.*, 2013]. Furthermore, the single target can be discrete or numeric [Lemmerich *et al.*, 2016].

Finding subgroups of objects for which a more accurate and robust model of *multiple* target values can be learned/built, instead of considering the whole data, has then been introduced as *Exceptional Model Mining* (EMM) [Leman *et al.*, 2008, Duivesteijn *et al.*, 2016]. In this framework, depicted in Fig. 4.1, there are two types of attributes, those used to characterize the subgroups (i.e., the object description), and others employed to evaluate the subgroup quality (i.e., the targets). Subgroups of interest are selected based on the quality of a model evaluated on the targets (e.g., classifier [Leman *et al.*, 2008], Bayesian Networks [Duivesteijn *et al.*, 2010], encoding based on Minimum Description Length [van Leeuwen, 2010]), measures on preference matrix [de Sá *et al.*, 2018], correlations between two numeric targets [Downar and Duivesteijn, 2017]). The combination of large description and target spaces, as well as the use of non-monotonic measures require the adoption of heuristic search methods such as beam search.

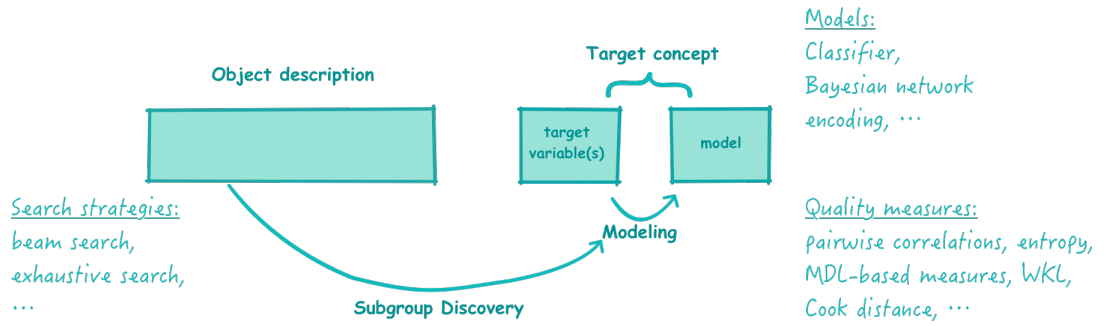


Figure 4.1: The Exceptional Model Mining framework (diagram adapted from [Duivesteijn, 2014]).

SD or EMM, that is the question. Indeed, SD and EMM are very similar: same search space (i.e., the object description space), same motivation. Therefore, some pattern miners do not see the need to use a new term instead of SD, the seminal one. This a respectable opinion that defends itself. Personally, I think that the use of EMM term is associated with a strong desire to focus on the model aspect, i.e., to define quality measures to highlight complex phenomena.

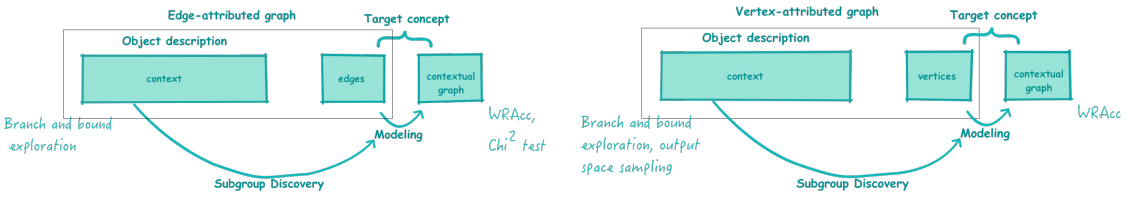


Figure 4.2: Exceptional subgraph mining for (a) edge attributed graphs and (b) vertex attributed graphs as instances of EMM.

In this chapter, we present two contributions to discover exceptional subgraphs. These contributions belong to this "exceptional subgroups" framework together with EMM and SD as illustrated in Figure 4.2. Importantly, when working in EMM/SD, we have to set a good model to characterize the whole data (i.e., an attributed graph in our case) as well a subset of the data (i.e., a subgraph). We investigated this in two – even if similar – different contexts: the edge-attributed graphs and the vertex-attributed graphs. Specificity of each kind of attributed graphs are considered in the definition of dedicated pattern domains and quality measures.

- Work on exceptional subgraph mining in edge-attributed graphs (Section 4.2) was done in collaboration with Mehdi Kaytoue, Céline Robardet and Albrecht Zimmermann and partially supported by

the projects GRAISearch (FP7-PEOPLE-2013-IAPP) and VEL'INNOV (ANR INOV 2012). The main result is [Kaytoue *et al.*, 2017] for which I would like to thank again the reviewers that made fruitful, constructive and insightful comments[11].

- Work on exceptional subgraph mining in vertex-attributed graph [Bendimerad *et al.*, 2016] and [Bendimerad *et al.*, 2018], detailed in Section 4.3, is part of Anes Bendimerad's PhD thesis.

4.2 Exceptional Subgraphs in Edge Attributed Graphs

We consider the challenge of mining graph data that result from the aggregation of individual behaviors. This type of data has become ubiquitous, for example with the advent of social networks that record connections between various entities made by users. As an illustration, Table 4.1(d) presents a graph of co-visitation sites built from the aggregation of bike trips of individual users (described in Table 4.1) that travel from one station to another one. Such a graph reflects the most general relationships among stations but the information about the users themselves is completely lost. Population specific behaviors are hidden inside this macroscopic view (i.e., the graph resulting from the aggregation of individual travels), whereas this information is highly interesting in many applications, such as recommender systems. It makes it possible to answer the following questions: *For a given population, what are the most strongly related subgraphs (i.e., behavior)?, For a given subgraph, what is the most strongly related population (i.e. representative users)?* Fig. 4.3(b) and (c) present two examples of contextual and exceptional contextual subgraphs whose description is given in the caption of the figure.

The data we consider consist of a collection of connections between nodes characterized by a set of attributes. This rich dataset is a multigraph, which can be envisaged as a transactional database anchored to a graph. In other words, each connection is recorded as a transaction containing attributes and associated to the edge along which the connection occurred. A context, i.e., a set of conditions on the transaction attributes, is used as a selection operator that identifies the subgroup of supporting transactions. A so-called contextual subgraph is derived from this subgroup of connections as the graph weighted by the number of transactions that for each edge that support the context. We propose to use a generalization mechanism on the contexts and to exploit it to identify exceptional contextual subgraphs, that is, contextual subgraphs whose weights are abnormally large in comparison to the most general contextual graph (the one containing all connections). Such exceptional subgraphs are of interest as most of the transactions that are associated to their edges in the whole graph support the context. For example, on the data of Table 4.1, the proposed method identifies connected stations that are visited in the same context. Fig. 4.3 (b) represents the contextual subgraph that corresponds to the stations that are visited by young people (age in $[20;23]$) at night. The number of trips that satisfy the context on each edge can be used as a support measure (see the weights on the edges) but this measure is not sufficient to evaluate how strongly the context is related to those edges, in contrast to all other movements occurring in this context.

To that end, we use the Weighted Relative Accuracy measure (WRAcc) to only retain contexts whose accuracy on the edge is markedly higher than the one obtained by the most general context on this edge. Fig. 4.3 (c) represents the subgraph of locations visited by young people at night whose edges have a positive value for $WRAcc$. The most specific context associated to this graph also includes the attribute `Type of area = {bars}`. The affinity of a context to an edge is also assessed by a χ^2 test that statistically assesses the dependency between the context and the edge.

[11] Especially, they really help us to have a better understanding on SD/EMM.

Station	Type of area
A	Bars
B	Bars
C	Bars
D	Bars
E	Residential

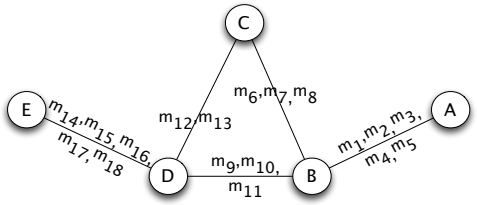
(a) Bike-share station characteristics.

UID	Gender	Age
u_1	F	20
u_2	M	23
u_3	F	45
u_4	M	50
u_5	F	30

(b) User characteristics

MID	Departure	Arrival	UID	Time	Weather
m_1	A	B	u_1	Day	Rainy
m_2	A	B	u_2	Night	Windy
m_3	A	B	u_3	Night	Cloudy
m_4	A	B	u_4	Day	Windy
m_5	A	B	u_5	Night	Rainy
m_6	B	C	u_1	Night	Cloudy
m_7	B	C	u_1	Night	Windy
m_8	B	C	u_1	Night	Rainy
m_9	B	D	u_2	Night	Cloudy
m_{10}	B	D	u_1	Night	Windy
m_{11}	B	D	u_1	Night	Cloudy
m_{12}	C	D	u_1	Night	Rainy
m_{13}	C	D	u_2	Night	Rainy
m_{14}	D	E	u_1	Night	Cloudy
m_{15}	D	E	u_2	Night	Windy
m_{16}	D	E	u_3	Day	Rainy
m_{17}	D	E	u_3	Night	Windy
m_{18}	D	E	u_4	Night	Rainy

(c) Bike trip characteristics.



(d) Augmented graph.

Table 4.1: Example of data: (a) Bike-share station attributes, (b) Users attributes, (c) Bike trip attributes and (d) Augmented graph corresponding to those data.

4.2.1 The Problem of Exceptional Contextual Subgraph Mining

In the following, we present the notion of augmented graph in which Exceptional Contextual Graphs are looked for. We describe the pattern domain incrementally: First *contexts* are introduced and different mappings/derivation operators allow to introduce contextual graphs. After, we introduced two evaluation measures used to filter uninteresting edges from such graphs. Finally, the problem of mining Exceptional Contextual Subgraphs is properly given.

Deriving Contextual Subgraphs

Augmented Graphs. The data we are interested in consist of a set of entities and a collection of connections between pairs of these entities, augmented with rich heterogeneous data about the entities and the circumstances of the connections. For instance, in Table 4.1, the entities can represent bike-share stations in a city with connections corresponding to bike trips made by users from one station to another. Additional details are available: The entities, i.e. stations, are geolocated and can be associated to additional information, characterizing their location (business vs. residential area, closeness to POI, elevation, urbanisation density, etc.) (see Table 4.1(a)). The connections, i.e. bike trips, are timestamped

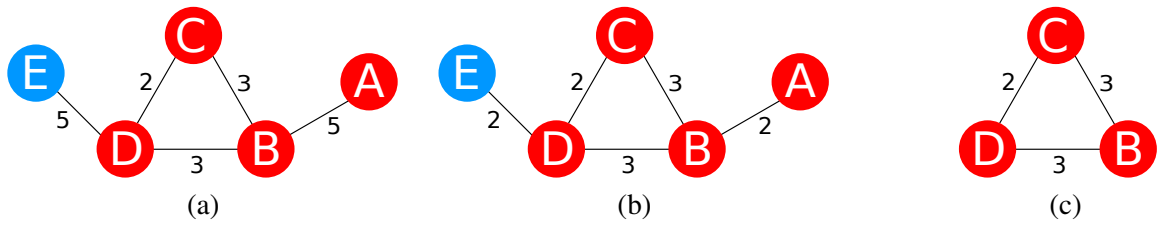


Figure 4.3: Example of contextual subgraph and exceptional contextual subgraph: (a) Contextual graph associated to the most general context $(\star,\star,\star,\star,\star) = (Age \in [20,50], Gender \in \{F,M\}, Time \in \{Day, Night\}, Weather \in \{Sunny, Cloudy, Windy, Rainy\}, type\ of\ area \in \{Bars, Residential\})$; (b) A contextual subgraph of bike trips made by young people at night $(Age \in [20,23], \star, Time \in \{Night\}, \star, \star)$; (c) An exceptional contextual subgraph with context $(Age \in [20,23], \star, Time \in \{Night\}, \star, type\ of\ area \in \{Bars\})$.

and can be augmented with the profile of the user, weather, events and other special conditions about the trip (see Tables 4.1(b) and (c)). This rich dataset is a multigraph, which can be viewed as a transactional database anchored to a graph (which is called augmented graph as explained afterwards). In other words, each connection is recorded as a transaction containing attributes (the join of tables 4.1(a), (b) and (c)) and associated to a source and a target entity that form the directed edge[12] along which the connection occurred. This type of data is called *augmented graph* and is formally defined below.

Definition 4.1 (Augmented graph)**.** *Let R be a relation whose schema is denoted $S_R = [R_1, \ldots, R_p]$. Each attribute R_i takes values in $\mathbf{dom}(R_i)$ that is either nominal, if there is no order relation among attribute modalities, or numerical. A transaction $t \in R$ of this relation is a tuple $(t_1, \ldots, t_p)$ with $t_i \in \mathbf{dom}(R_i)$. An augmented graph $G = (V, E, T, \text{EDGE})$ consists of a set V of vertices, a set $E \subseteq V \times V$ of edges, a set T of transactions, and a function that maps a transaction to its edge:* $\text{EDGE} : T \to E$.

Table 4.1(d) illustrates the augmented graph that corresponds to the data of Table 4.1. On such data, we aim to identify subgraphs that are typical for a context, as the one of Fig. 4.3 (c) whose bike trips mainly correspond to users of $\text{AGE} \in [20,23]$, and realized at *Night* between stations having many *Bars* in their neighborhood. This is what we define below as exceptional contextual subgraph mining, a problem rooted in the Exceptional Model Mining framework [Duivesteijn *et al.*, 2016, Leman *et al.*, 2008] (see Fig. 4.1).

EMM extends classical subgroup discovery – the discovery of subgroups described by a few conditions on their attributes and whose target attribute somehow deviates from the norm – to the case where several target attributes are considered and used to derive a model. A subgroup is thus deemed interesting when its associated model is substantially different from the model on the whole dataset. In such a framework, our problem, which is illustrated in Fig. 4.2(a), can be depicted as follows: The data (i.e., the augmented graph) consist of a collection of transactions (or records) composed of attributes and associated to an edge of the graph. A description, which is here called a *context*, is used to select transactions that support it. This set of transactions is then projected onto a so-called contextual graph, on which the interestingness or exceptionality of the context is evaluated. Hence, edges correspond to multiple targets and the contextual graph plays the same role as the model in EMM.

Contextual Graphs. Let us first define a context, that is, the description of a set of transactions.

[12]For sake of simplicity, we use the term *edge* to refer indifferently to directed or undirected edges without loss of generality.

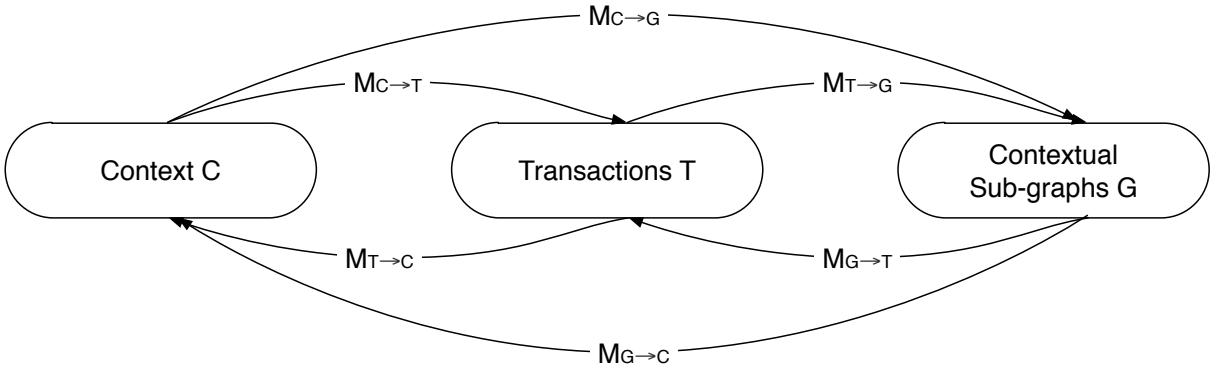


Figure 4.4: Mappings between the different data views of an augmented graph G

Definition 4.2 (Context)**.** *Given a set of transactions $S \subseteq T$, we define the function $M_{T\to C}(S)$ that maps S to the context $(C_1,\cdots,C_p)$ as*

- *$C_i = a$, with $a \in \textbf{dom}(R_i)$, iff R_i is nominal and $\forall(t_1,\cdots,t_i,\cdots,t_p) \in S$, $t_i = a$*
- *$C_i = \star_i$, with $\star_i$ a new symbol representing the whole set $\textbf{dom}(R_i)$, iff R_i is nominal and there exists two transactions $t, t' \in S$ such that $t_i \neq t'_i$.*
- *$C_i = [a,b]$, with $a = \min\{t_i \mid (t_1,\cdots,t_i,\cdots,t_p) \in S\}$ and $b = \max\{t_i \mid (t_1,\cdots,t_i,\cdots,t_p) \in S\}$ iff R_i is numerical.*

Analogously, a transaction $t = (t_1,\ldots,t_p)$ satisfies or supports a context $C = (C_1,\cdots,C_p)$, noted $t \preceq C$, if and only if $\forall i = 1\ldots p$

- *$t_i = C_i = a$, with $a \in \textbf{dom}(R_i)$ and R_i nominal*
- *t_i is any of $\textbf{dom}(R_i)$, with $C_i = \star_i$ and R_i nominal*
- *$a \leq t_i \leq b$, with $C_i = [a,b]$ and R_i numerical.*

It is important to note that a context is covered by a set of transactions. Each transaction is attached to an edge, so a set of transactions induces a subgraph. In a dual way, given an arbitrary subgraph, one can retrieve the set of transactions attached to its edges, and the most specific context that covers all these transactions. For convenience, we will use the following mappings between the different views of an augmented graph (illustrated on Fig. 4.4).

Definition 4.3 (Basic mappings of an augmented graph)**.** *We associate the six following mappings to an augmented graph:*

- *The mapping $M_{C\to T}$ takes a context C as argument and returns the set of transactions that are covered by C, $M_{C\to T}(C) = \{t \in T \mid t \preceq C\} \subseteq T$. With arguments C and S this mapping returns the subset of transactions of $S \subseteq T$ that are covered by C: $M_{C\to T}(C,S) = \{t \in S \mid t \preceq C\} \subseteq S$. For example, $M_{C\to T}(Age \in [23,45], \star, Time \in \{Night\}, \star, type\ of\ area \in \{Bars\}) = \{m_2, m_3, m_9, m_{13}\}$.*
- *The mapping $M_{T\to G}$ takes a set of transactions $S \subseteq T$ and returns the subgraph consisting of the edges to which these transactions are attached: $M_{T\to G}(S) = \bigcup_{t\in S} \textsc{Edge}(t)$.*
- *$M_{C\to G} = M_{T\to G} \circ M_{C\to T}$ is the composition of the two operators introduced above.*
- *The mapping $M_{T\to C}$ is the counterpart of $M_{C\to T}$. It takes a set of transactions $S \subseteq T$ and returns the most specific context that covers all transactions in S. For example, $M_{C\to T}(\{m_2, m_3\}) = (Age \in [23,45], \star, Time \in \{Night\}, \star, type\ of\ area \in \{Bars\})$*
- *The mapping $M_{G\to T}$ associates a set of transactions to an edge, that is the transactions that are attached to this(these) edge(s). It is the counterpart of $M_{T\to G}$.*
- *$M_{G\to C} = M_{T\to C} \circ M_{G\to T}$ is the composition of the two operators introduced above.*

By coupling these notions of augmented graph and context, we define a contextual subgraph as the projection of an augmented graph on a context, i.e. a graph whose edges are weighted by the number of their associated transactions that satisfy C:

Definition 4.4 (Contextual Subgraph)**.** *Given an augmented graph $G = (V, E, T, \text{EDGE})$ and a context C, the contextual subgraph generated by C is the weighted graph $G_C = (V, E_C, W_C)$ defined by:*

- $W_C : E_C \to \mathbb{R}$ *with* $W_C(e) = |M_{C\to T}(C, M_{G\to T}(e))|$*, the number of transactions associated to e that satisfy C,*
- $E_C = \{e \in E \mid W_C(e) > 0\}$.

For example, Fig. 4.3(b) shows the contextual subgraph of the context $(Age \in [20,23], \star, Time \in \{Night\}, \star, \star)$.

Closed Contexts. It may happen that some contexts map exactly to the same set of transactions: for C^1 and C^2 two different contexts, it is possible that $M_{C\to T}(C^1) = M_{C\to T}(C^2)$ which implies that $M_{C\to G}(C^1) = M_{C\to G}(C^2)$. By using an appropriate order relation, it is possible to avoid this redundancy by considering only *closed contexts*.

Definition 4.5 (Partial order on context set)**.** *We say that a context C^1 is more specific than a context C^2, denoted $C^1 \preceq C^2$, iff*

- $C^2_i = \star_i$ *or* $C^1_i = C^2_i = a \in \textbf{\textit{dom}}(R_i)$*, for R_i a nominal attribute,*
- $[a^1_i, b^1_i] \subseteq [a^2_i, b^2_i]$ *with* $C^1_i = [a^1_i, b^1_i]$ *and* $C^2_i = [a^2_i, b^2_i]$*, for all numerical attributes R_i.*

The set of all possible contexts embedded with the relation $\preceq$ forms a semi-lattice where the most general context C is such that $C_i = \star_i$ for all nominal attributes R_i, and $C_i = [\min(dom(R_i)), \max(dom(R_i))]$ for all numerical attributes R_i.

As such, instead of enumerating all contexts, it is enough to only enumerate the closed ones: The *closure operator* maps any context to the unique most specific one with the same image $M_{C\to T}$.

Definition 4.6 (Closed context)**.** *A context C is closed iff $\forall C'$ such that $M_{C\to T}(C) = M_{C\to T}(C')$, $C \preceq C'$. Thus, $M_{T\to C}(M_{C\to T}(C'))$ returns the closed pattern of C' and is called the closure operator.*

The proof that $M_{T\to C} \circ M_{C\to T}$ is a closure operator is omitted as it is a well-known notion in the pattern mining and formal concept analysis fields.

Deriving Exceptional Contextual Graphs

In pattern mining, it is usual to evaluate the interestingness of a pattern by well-chosen measures. To judge the strength of the dependency between a context and a derived graph (or each edge), we propose to use two evaluation measures: The Pearson's chi-squared test of independence [Pearson, 1900] and the Weighted Relative Accuracy measure.

χ^2 Test of Independence. To evaluate the dependency between a context C and an edge e, we consider the proportion of transactions associated to e that satisfy the context and propose to statistically assess this value by means of a Pearson's chi-squared test of independence [Pearson, 1900]. This test determines whether or not the context appears significantly more often in the transactions of e than in all the whole set of transactions of the augmented graph.

A transaction satisfies or not a context C, and is associated or not to an edge e. These four possible outcomes are denoted $\mathbf{C}$ and $\overline{\mathbf{C}}$, $\mathbf{e}$ and $\bar{\mathbf{e}}$. Table 4.2(a) is the contingency table $O(C, e)$ that collects the observed outcomes of $\mathbf{e}$ and $\mathbf{C}$. The null hypothesis states that e and C are statistically independent.

Under the hypothesis that $\mathbf{C}$ is uniformly satisfied by the edges of the augmented graph, there are $W_\star(e)\frac{\sum_{x\in E} W_C(x)}{\sum_{x\in E} W_\star(x)} = E(C,e)$ chances that a transaction that satisfies the context $\mathbf{C}$ is associated to the edge e. The three others outcomes under the null hypothesis are constructed on the same principle and are given in the contingency table E presented in Table 4.2(b). The value of the statistical test is thus

$$X^2(C,e) \quad = \quad \sum_{i\in\{C,\overline{C}\}} \sum_{j\in\{e,\overline{e}\}} \frac{(O(i,j)-E(i,j))^2}{E(i,j)}$$

The null distribution of the statistic is approximated by the χ^2 distribution with 1 degree of freedom, and for a significance level of 5%, the critical value is equal to $\chi^2_{0.05} = 3.84$. Consequently, $X^2(C,e)$ has to be greater than 3.84 to establish that the weight related to a context on a given edge deviates sufficiently to reject the null hypothesis and conclude that the edge weight is biased at 95% significance level.

	$\mathbf{e}$	$\overline{\mathbf{e}}$	
$\mathbf{C}$	$W_C(e)$	$\sum_{x\in E} W_C(x) - W_C(e)$	$\sum_{x\in E} W_C(x)$
$\overline{\mathbf{C}}$	$W_\star(e) - W_C(e)$	$\sum_{x\in E} W_\star(x) - W_\star(e) - \sum_{x\in E} W_C(x) + W_C(e)$	$\sum_{x\in E} W_\star(x) - \sum_{x\in E} W_C(x)$
	$W_\star(e)$	$\sum_{x\in E} W_\star(x) - W_\star(e)$	$\sum_{x\in E} W_\star(x) = \lvert T\rvert$

(a) Contingency table O of events $\mathbf{C}$ and $\mathbf{e}$.

	$\mathbf{e}$	$\overline{\mathbf{e}}$	
$\mathbf{C}$	$W_\star(e)\frac{\sum_{x\in E} W_C(x)}{\sum_{x\in E} W_\star(x)}$	$(\sum_{x\in E} W_\star(x) - W_\star(e)) \times \frac{\sum_{x\in E} W_C(x)}{\sum_{x\in E} W_\star(x)}$	$\sum_{x\in E} W_C(x)$
$\overline{\mathbf{C}}$	$W_\star(e) \times \left(1 - \frac{\sum_x W_C(x)}{\sum_{x\in E} W_\star(x)}\right)$	$(\sum_{x\in E} W_\star(x) - W_\star(e)) \times \left(1 - \frac{\sum_{x\in E} W_C(x)}{\sum_{x\in E} W_\star(x)}\right)$	$\sum_{x\in E} W_\star(x) - \sum_{x\in E} W_C(x)$
	$W_\star(e)$	$\sum_{x\in E} W_\star(x) - W_\star(e)$	$\sum_{x\in E} W_\star(x) = \lvert T\rvert$

(b) Contingency table E under the null hypothesis.

Table 4.2: Contingency tables O and E.

The Weighted Relative Accuracy Measure. In the χ^2 test of independence, the rejection of the null hypothesis can be due to either a very large or a very low value of $|M_{C\to T}(C, M_{G\to T}(e))|$. We distinguish these two cases thanks to an additional measure, based on the Weighted Relative Accuracy measure.

For a given context C, we aim to identify the edges for which the number of transactions satisfying the context C is greater than what is observed for all the edges of the augmented graph. The relative accuracy is based on the subtraction of the relative weight of the edge e in the whole augmented graph $G_\star$ from its relative weight in the contextual graph. We choose to normalize edge weights by the maximal weight of any edge matching the context:

$$\frac{W_C(e)}{\max_{x\in E} W_C(x)} - \frac{W_\star(e)}{\max_{x\in E} W_\star(x)}$$

If this term is larger than 0 then the edge weight is greater than expected from the marginal distribution over the whole graph. This means that this edge is of relatively greater importance for the context than it is for the full augmented graph. However, it is easy to obtain high relative accuracy with highly specific contexts. Such contexts have a low value on $\frac{\max_{x\in E} W_C(x)}{\max_{x\in E} W_\star(x)}$, the specificity weight. Therefore, to obtain interesting contexts, we use the WRACC measure that trades off the relative accuracy with the specificity weight:

$$\text{WRAcc}(C,e) = \frac{\max_{x\in E} W_C(x)}{\max_{x\in E} W_\star(x)} \times \left(\frac{W_C(e)}{\max_{x\in E} W_C(x)} - \frac{W_\star(e)}{\max_{x\in E} W_\star(x)} \right)$$

We consider that an edge e depends on a context C if $\text{WRAcc}(C,e) > 0$.

Exceptional Contextual Graphs. Until now, we presented how to derive contextual subgraphs of an augmented graphs and introduced two measures to asses the significance of its edges. It remains to filter out the insignificant edges to obtain so called *Exceptional contextual subgraphs*. We formalize this with the following definition.

Definition 4.7 (Exceptional edges with respect to a context)**.** *An edge e is considered to be exceptional with respect to a context C, denoted* Except*(C,e), iff*

$$\begin{aligned}
\text{Except}(C,e) \quad &\equiv \quad e \in M_{C\to G}(e) && (4.1)\\
&\textit{and} \quad |M_{C\to T}(C, M_{G\to T})| > \textit{min_weight} && (4.2)\\
&\textit{and} \quad X^2(C,e) > \chi^2_{0.05} && (4.3)\\
&\textit{and} \quad \text{WRAcc}(C,e) > 0 && (4.4)
\end{aligned}$$

Deriving Exceptional Contextual Connected Components

We have defined the notion of the Exceptional Contextual Graph and how to derive instances of it and evaluate the affinity of a (closed) context to an edge (with χ^2 test and WRAcc measure). Taking into account the topology of the subgraph associated to a context is also of interest. Its connectivity can be understood by examining its connected components. As numerical measures describing these connected components, we use the number of vertices and the number of edges. We also evaluate the global quality of the edges of each connected component by the sum of the individual WRAcc measures. We can now define precisely the kind of patterns that we are looking for, called *Exceptional Contextual Connected Components*, or simply *Exceptional Contextual (Sub-)Graphs* for sake of simplicity.

Problem 4.1 (The Exceptional Contextual Graph Mining Problem)**.** *Extracting meaningful patterns from an augmented graph $G = (V,E,T,\text{Edge})$ is achieved by computing the theory:*

$$\begin{aligned}
\{(C, CC_C) \quad | \quad & CC_C = (V_{CC}, E_{CC}) \textit{ is a maximal connected components of } G_C \\
& \textit{with } G_C = (V, \{e \in E \mid \text{Except}(C,e) \textit{ is true}\}) \\
& \textit{and } C \textit{ is closed} && (4.5)\\
& \textit{and } |V_{CC}| \geq \textit{min_vertex_size} && (4.6)\\
& \textit{and } |E_{CC}| \geq \textit{min_edge_size} && (4.7)\\
& \textit{and } \sum_{e\in E_{CC}} (\text{WRAcc}(C,e)) \geq \textit{min_sum_wracc} && (4.8)\\
\}&
\end{aligned}$$

As such, computing the whole collection of patterns requires one to enumerate the closed contexts and apply the different filtering and pruning operations as briefly explained in the next subsection.

4.2.2 COSMIc Algorithm Principle

The theoretical search space of *exceptional contextual subgraph* patterns contains all possible combinations of contexts and subgraphs. Considering that contexts are ordered by $\preceq$ and subgraphs by the inclusion of their set of edges, the pattern set is structured as a semi-lattice bounded by $\{\star, G_\star\}$. As contexts and subgraphs are linked by the mappings $M_{C\to T}$, $M_{T\to G}$, $M_{G\to T}$ and $M_{T\to C}$, we can enumerate one and derive the other one. In our proposed algorithm, named COSMIc[13], contexts are enumerated first and the associated subgraph is updated all along the enumeration process. Upper bounds and other pruning techniques are used to reduce the search space size.

COSMIc enumerates contexts in a depth-first search manner. Given the pattern (C, G_C) that is currently explored, the algorithm returns all the specializations of C that are *exceptional contextual subgraph*s. If all the attributes have been instantiated, the connected components of G_C are considered and if the constraints are checked, then pattern is output.

If the attribute R_i can still be specialized in the context, a new context C' is generated: If R_i is symbolic, a loop over the values of $\mathbf{dom}(R_i) \cup \star_i$ lists all the possible specializations C' of C on R_i . Then, the transactions of G_C that do not satisfy C' are removed. The closure F of C' is then computed. If C' is closed, it is then investigated for pruning aims. If $G_{C'}$ is not empty, $(C', G_{C'})$ is recursively enumerated to generate all valid *exceptional contextual subgraph* patterns.

When R_i is numerical, enumerating all possible contexts consists of listing all intervals, i.e. those whose end-points occurring in the relation R. Let $\mathbf{dom}_{R_i} = (v_i^1, \cdots, v_i^m)$ be the ordered set of values that appear for attribute i in relation R. The function `next` (analogously `previous`) provides access to the following (analogously preceding) value of the one given as parameter. To enumerate all intervals included in $\mathbf{dom}(R_i)$ once and only once, we generate, from each interval $[a,b]$, two intervals $[a, previous(b)]$ and $[next(a), b]$, the first one $[a, previous(b)]$ being generated only if its left end-point a has not been increased so far.

For each interval, a new context C' is generated and, as for nominal attributes, the transactions of G_C that do not satisfy C' are removed, the pruning techniques applied, $(C', G_{C'})$ is then recursively enumerated.

The algorithm is based on two pruning mechanisms. The first one consists of removing individual edges. The constraint $|M_{C\to T}(C, M_{G\to T}(e))| > min\_weight$ (constraint (2) in Definition 4.7) is anti-monotonic and can be used to safely remove edges as soon as they do not satisfy the constraint. Constraint (3) on $X^2(C,e)$ is not anti-monotonic, but we use an upper bound $X^2_{ub}(C,e)$ to remove the edge e from G_C as soon as we have guarantee that none of the specializations of C can lead to $X^2(C,e) > \chi^2_{0.05}$.

The second pruning mechanism focuses on the connected components CC of G_C. Constraints (6) and (7) of Problem 4.1 are anti-monotonic and are used to stop the enumeration as soon as they are not satisfied by CC. Constraint (8) is not anti-monotonic, but we propose a tight upper bound with pruning capabilities without loss of promising patterns.

4.2.3 Travel Patterns in the VÉLO'V System

VÉLO'V is the bike-sharing system run by the city of Lyon (France) and the company JCDecaux[14]. There are a total of 348 VÉLO'V stations across the city of Lyon. Our VÉLO'V dataset contains all the trips collected over a 2 year period (Jan. 2011 – Dec. 2012). Each trip includes the bicycle station and the time stamp for both departure and arrival, as well as some basic demographics about the users (gender, age, zip code, country of residence, type of pass). Hence, the VÉLO'V stations are the graph vertices ($|V| = 348$), and directed edges correspond to the fact that a VÉLO'V user checks out a bicycle at a station and returns

[13] COSMIc stands for *COntextual Subgraph MIning*.

[14] `http://www.velov.grandlyon.com/`

it at another. There are in total 164,390 users for which demographics are available and 6.7 million of transactions (i.e., movements).

The rapid development of bicycle sharing and renting systems has an impact on urban mobility practices. Studying this impact is crucial for the following reasons: (1) It is important to understand whether and how this new service contributes to the emergence of new mobility trends; (2) This study is multi-disciplinary and involves physicists, economists, geographers and sociologists as well as the practicians directly involved with the bicycle sharing system. Notice that our approach fits well in a multi-disciplinary context since the patterns we are interested in are interpretable without data mining expertise; (3) The conclusions of the analysis are of interest for several urban mobility actors (local authorities and private mobility operators). For instance, these conclusions can be transferred to new cities for the deployment of new services.

The problem setting for our experiments on the VÉLO'V data is essentially the one that we outlined in the introduction to motivate our work: Given the characteristics of different users, we aim to identify populations that use the rental bicycles in a particular manner. Hence each pattern is a hypothesis on a movement schema (connected subgraph) for a specific population (the context). We transformed the initial data set into several databases of transactions. This gives us the opportunity to experiment with the algorithm in various conditions with non-synthetic data while also exploring the data to elicit hypothesis. We generated $|\{2weeks, october, all\} \times \{basic, extended\}|$ datasets as defined by:

- *Number of transactions.* To vary this parameter, three subsets of data have been chosen: The first two weeks of October 2011, denoted as *2weeks*, with 312,185 transactions), the full month of October 2011, denoted as *october*, with 565,065 transactions, and the full dataset, denoted as *all*, with 6,713,937 transactions.
- *Number of attributes.* In its *basic* version, the dataset contains the following attributes: $daytime \in \{morning, midday, evening, lastmetro, night, other\}$ which denotes specific bike usage [Hamon, 2015]; the *zipcode*, gender, country, and age of the biker (where $age \in \{[14;25][25,60], \geq 60\}$ still according to [Hamon, 2015]) as well as the type of *pass* subscribed by the user.
  In its *extended* versions, the dataset contains properties of both departure and arrival stations (edge source and target attributes). We use census data provided by the National Institute of Statistics and Economic Studies (INSEE) that provides meaningful information about education, employment, industries, etc. Each station is labeled with some information of the INSEE division whose center is the closest to the bike station (using a Google API). The information used is *TrainStation, University, Companies, Hotel, Tourism*, which respectively are true if there is at least a train station or a university, at least 10 companies, at least one hotel and at least one tourism center. In total there are 9 attributes for the basic datasets and 19 for extended ones.

To evaluate the ability of COSMIc to deal with a real world dataset, we report the run times and number of extracted patterns (Fig. 4.5) for the dataset $(october, basic)$, with $min\_vertex\_size = 2$, $min\_edge\_size = 1$. The results indicate that COSMIc is able to mine patterns in a real life dataset, even with low minimal support on the edges (*min_weight*) and no other constraints that would otherwise reduce the size of the search space.

However, it should be noted that the extraction of the whole dataset with extended attributes $(all, extended)$ takes too long (more than two days). A way to solve this problem is to impose a syntactic constraint, that is to say to start the enumeration in a given context C_{root}. In this setting, COSMIc produces only more specific patterns than C_{root}, yet still taking the whole dataset for computing edges probabilities $W_\star(.)$. This allows an expert to partially materialize his hypothesis.

Fig. 4.6 presents a visualization on the map of Lyon of six patterns that we extracted from different datasets introduced above For each, we detail the experimental protocol and propose an interpretation.

To start, we mine the dataset $(october, extended)$ with $C_{root} = (\star_1, \ldots, \star_p)$. The extraction lasts 62 minutes and returns 1,703 patterns (with parameters $min\_vertex\_size = 2$, $min\_edge\_size = 1$ and

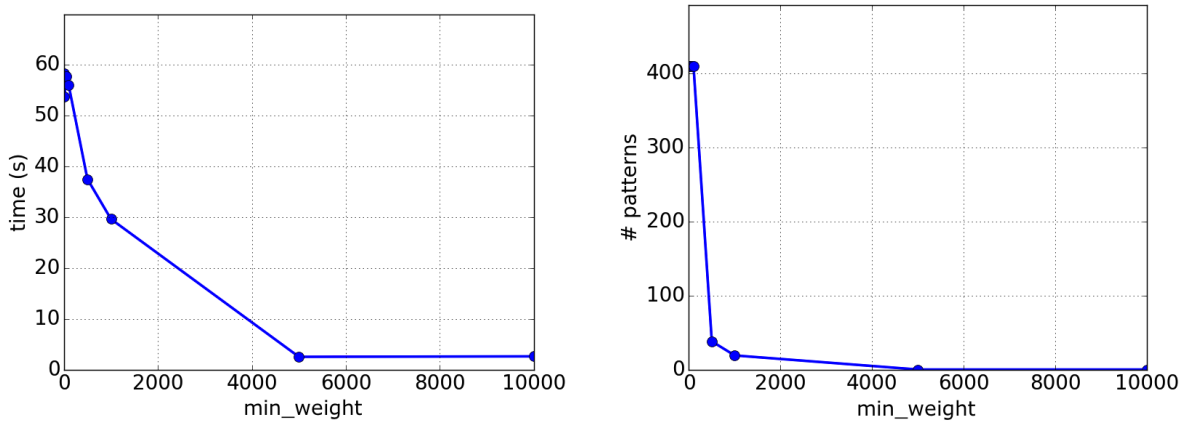
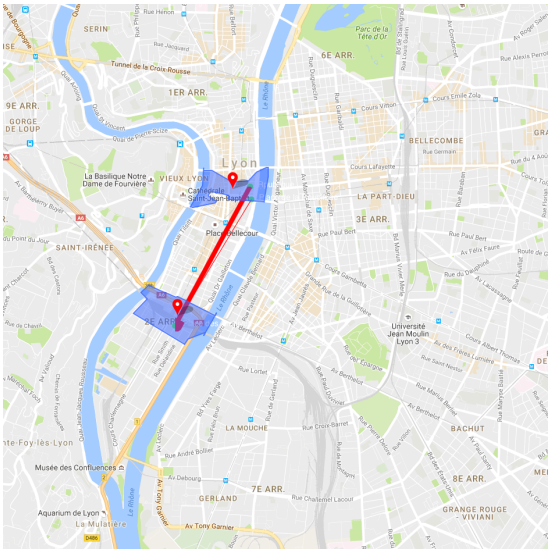


Figure 4.5: Run times (left) and number of patterns (right) on VÉLO'V when varying *min_weight*.

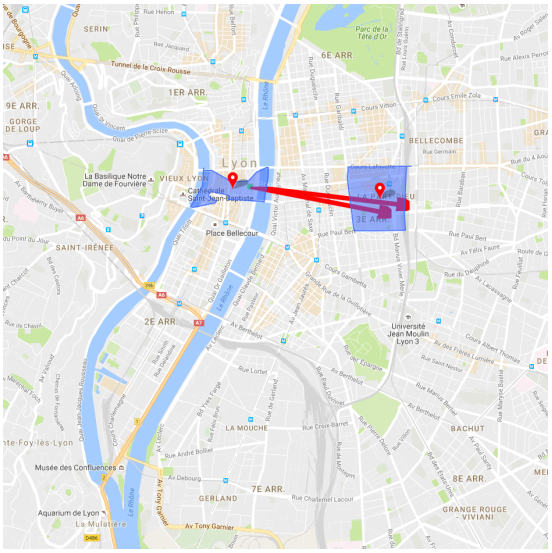
min_weight = 6). Instead of evaluating each pattern individually, we choose to filter out patterns whose context does involve the attribute-value pair *zip_code* = 38. This is the zip code of a neighboring area of Lyon reachable by car and train (at least 40 km away from Lyon). The idea is to understand the behavior of non-Lyon residents. Two patterns (out of a total of three) are presented in Fig. 4.6 (a) and (b). Both of their contexts C_1 and C_2 include *pass* = *OURA*, which means that the users have a VÉLO'V pass card linked to a rail pass (both patterns contain each 3 vertices and 4 edges and a WRACC sum of, resp. 0.04 and 0.12.). Their subgraphs involve the two main train stations of Lyon: Perrache (south-west) and Part-Dieu (center) that are connected to the main city center square of Lyon, named Bellecour. These patterns may thus identify stations that the VÉLO'V system operator should care of at the end of work hours: bikes must be available for workers that seek to reach the train stations.

While mining the dataset $(all, basic)$ with $\{daytime = night\} \in C_{root}$, we obtain 45 patterns in 80 seconds (with parameters *min_vertex_size* = 2, *min_edge_size* = 1 and *min_weight* = 6). Two of these patterns are shown in Fig. 4.6 (c) and (d): The graph associated to C_3 involves three areas known for their nightlife (left hand side of the figure) and two residential areas with many young inhabitants (on the right). Context C_4 contains the attribute-value *zipcode* = 69005 and its associated graph displays travels between this area (on the left along the river) and Lyon's opera as well as the Part-Dieu rail station. The pattern represented in Fig. 4.6 (c) (resp. (d)) contains 7 nodes and 10 edges with a WRACC sum of 0.03 (resp. 6, 10 and 0.05). These patterns may thus identify key stations and demographics that the VÉLO'V system operator could target for heightening awareness campaigns on, for example, dangers when biking at night or after parties.

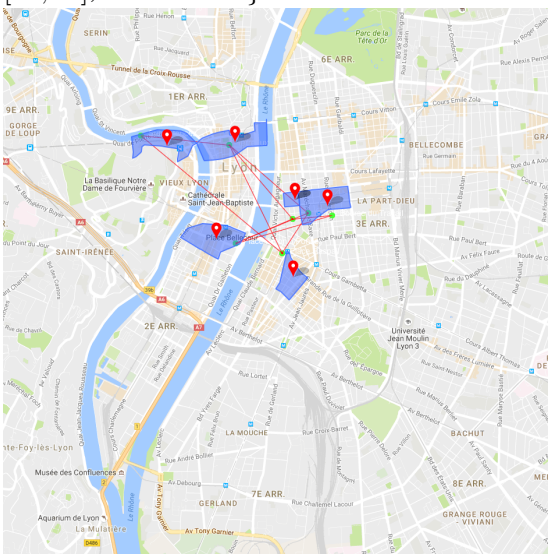
Finally, we run COSMIc on $(all, extended)$ starting the pattern enumeration with $\{age \in [14, 26]\} \in C_{root}$, thus aiming to get insights on young people's behaviour. The execution took 70 minutes with *min_vertex_size* = 15, *min_weight* = 100 and *min_sum_wracc* = 0.1. It returned 31 patterns. Two of the patterns obtained are shown in Fig. 4.6 (e) and (f), having, respectively, 18 vertices, 45 edges, and a WRACC sum of 0.28, and 16 vertices, 39 edges, and a WRACC sum of 0.3. In the graph associated to C_5, edges link city center areas with the city center campus. Pattern C_6 contains the attribute-value pair *zipcode* = 69004 which is the area where many edges depart (upper left part). The arrivals of these edges are the main components of the University of Lyon spread across the city. Most importantly, in both case the context hints the presence of universities in the IRIS attached to each node. One possible interpretation is that these two patterns depict students from the 4^{th} district of Lyon going to their universities. Here again, such hypotheses are valuable for the VÉLO'V system operator as it gives hints on the behavior for particular demographics (without specifying them explicitly: The root pattern has just a single attribute



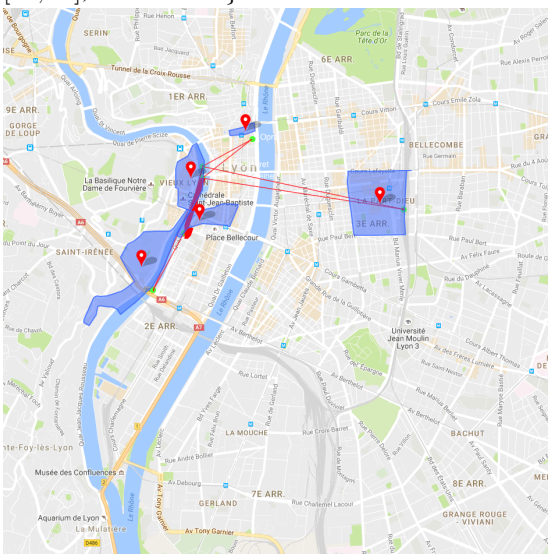
(a) $C_1 = \{Zip\_code = \underline{38}, Gender = men, Age = [26,60], Pass = oura\}$



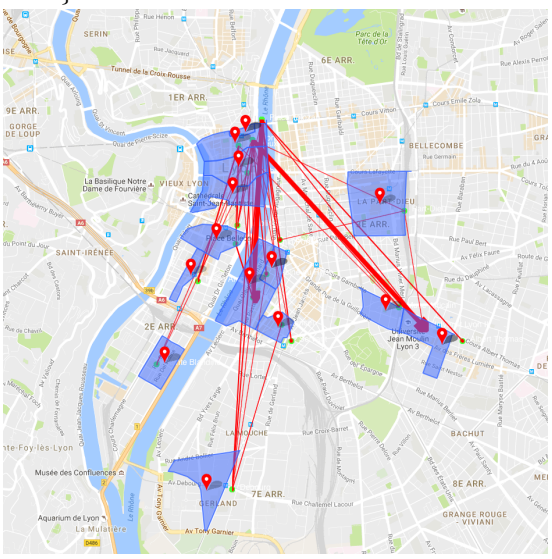
(b) $C_2 = \{Zip\_code = \underline{38}, Gender = men, Age = [26,60], Pass = oura\}$



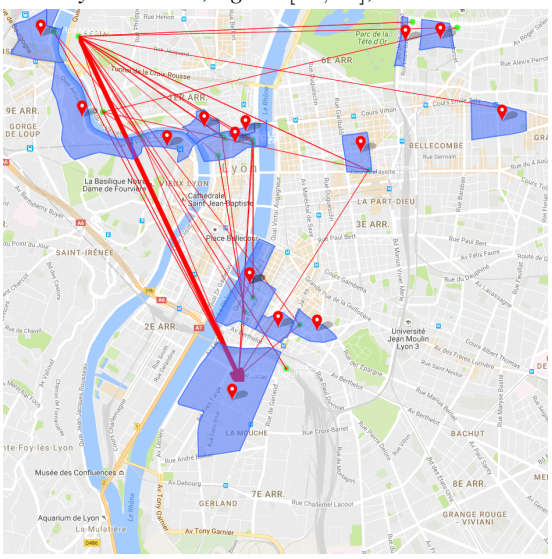
(c) $C_3 = \{Time = \underline{night}, Country = France, Pass = velov\}$



(d) $C_4 = \{Time = \underline{night}, Zip\_code = 69005, Country = France, Age = [26,60], Pass = standard\}$



(e) $C_5 = \{Gender = men, Age = \underline{[14,26]}, University\_In, University\_out\}$



(f) $C_6 = \{Zip\_code = 69004, Gender = men, Age = \underline{[14,26]}, University\_In, University\_out\}$

Figure 4.6: Several contextual subgraphs discovered in the VÉLO'V datasets.

instantiated).

4.3 Exceptional Subgraphs in Vertex Attributed Graphs

In Subsection 4.2, we have described the COSMIc algorithm to discover exceptional subgraphs in attributed graphs. Exceptionality is mainly assessed with regard to the edge attributes. Vertex attributes can involve within the patterns. However, in this framework, a subgraph with conditions on only vertex attributes cannot be considered as exceptional by COSMIc. Indeed, the quality measure evaluate how the weights of the edges of the graphs are unexpected compared to the global model. Therefore, conditions on vertex attributes does not allow to select a subset of adjacent edges and lead to expected subgraphs.

As a consequence, we have to define a new type of pattern to make possible the identification of subgraphs for which some attributes on vertices have exceptional values compared to the rest of the graph. This is the aim of this subsection.

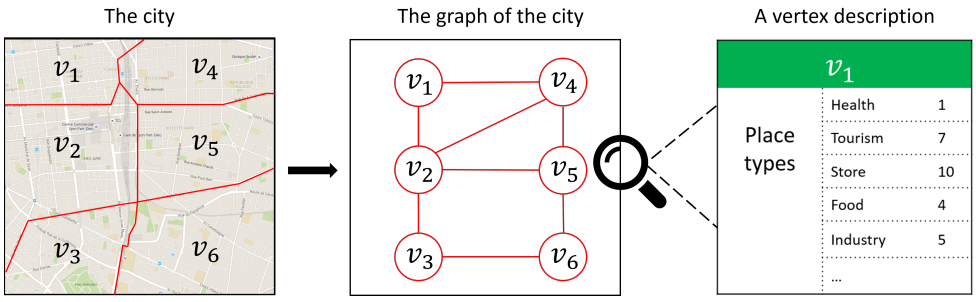


Figure 4.7: Example of a graph modeling a city.

4.3.1 The Problem of Mining Closed Exceptional Subgraphs in Vertex Attributed Graphs

Data describing geographic venues are numerous, ranging from census data to collaborative data produced through social-media platforms. To describe a city, nearby venues are grouped into small areas (geographers generally use tiles of 200 meters) over which venue characteristics are aggregated into count data. These areas are hereafter considered as the vertices V of a graph $G = (V, E, C, D)$ whose edges E connect adjacent areas (that share a part of their borders), $C = \{c_i, i \in [\![1, p]\!]\}$ is a set of p categories and the vertices of V are described by $D = \{c_i(v) \in \mathbb{N}, \text{ with } c_i \in C \text{ and } v \in V\}$, the counts of venues of each category in the area associated to each vertex. The values of D can be aggregated over a set of vertices $K \subseteq V$ and a set of categories $L \subseteq C$: $sum(L, K) = \sum_{v \in K} \sum_{c_i \in L} c_i(v)$. To simplify the notation, we use $sum(K)$ to denote $sum(C, K)$.

As an example, Fig. 4.7 presents a graph derived from the division of a city into 6 areas (from v_1 to v_6). The area represented by v_1 is adjacent to the ones represented by v_2 and v_4, and consequently an edge connects v_1 to v_2 and another one v_1 to v_4. The number of venues of each category in a given area composed a vector associated to the corresponding vertex. The distribution of venue categories $C = (Health, Tourism, Store, Food)$ is detailed in Fig. 4.8. $sum(health, \{v_1\}) = 1$ as there is one venue with the category *health* in the area associated to v_1. We can also observe that $sum(\{Health, Tourism, Store, Food\}, \{v_1\}) = 22$, and for the set $K = \{v_2, v_5\}$, $sum(K) = 49$.

Our objective is to identify neighborhoods whose characteristics distinguish them from the rest of the city. To that end, we propose to discover connected subgraphs associated to exceptional categories. A category is exceptional for a subgraph if it is more frequent in its vertices than in the remaining of the graph. The scarcity of a category can also be a relevant element to describe a neighborhood. For example,

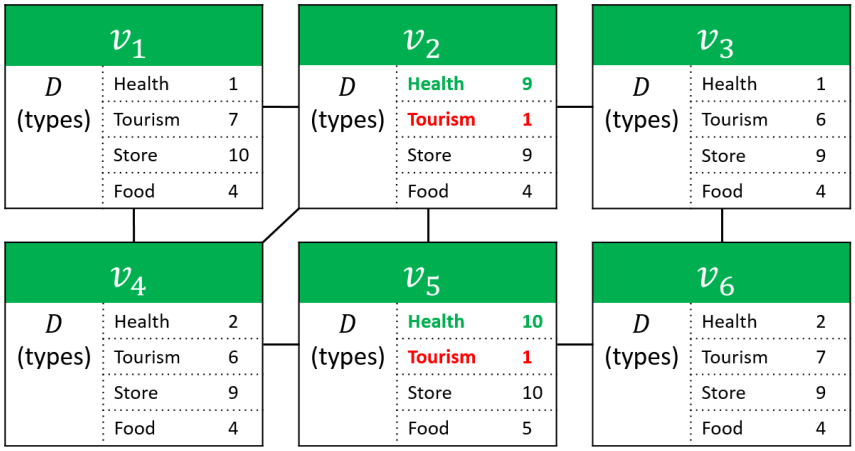


Figure 4.8: Example of the distribution of venues in areas.

in Fig. 4.8, vertices v_2 and v_5 have a surplus on the category *Health* compared to the rest of the graph, while having a loss on category *Tourism*. We formalize the excess and deficit in the amount of some categories by means of characteristics defined as

Definition 4.8 (Characteristic)**.** *A characteristic is defined as a pair $S = (S^+, S^-)$ with S^+ and S^- two disjoint subsets of C. The set of all characteristics is denoted $\mathscr{S}$. We also define operators between two characteristics $S_1 = (S_1^+, S_1^-)$ and $S_2 = (S_2^+, S_2^-)$:*

- $S_1 \cap S_2 = (S_1^+ \cap S_2^+, S_1^- \cap S_2^-)$
- $S_1 \cup S_2 = (S_1^+ \cup S_2^+, S_1^- \cup S_2^-)$
- $S_1 \subseteq S_2 \Leftrightarrow S_1^+ \subseteq S_2^+ \wedge S_1^- \subseteq S_2^-$
- $|S| = |S^+| + |S^-|$

In order to assess the relevancy of the characteristic S with respect to the subgraph induced by $K \subseteq V$, noted $G[K]$, we define the measure $WRAcc(S,K)$, an adaptation of the weighted relative accuracy measure widely used in Subgroup Discovery [Lavrac *et al.*, 2004].

A set of categories L is discriminant to $G[K]$ if it is more or, on the contrary, less frequent in $G[K]$ than in G. This is evaluated by the *gain* function:

$$gain(L,K) = \frac{sum(L,K)}{sum(K)} - \frac{sum(L,V)}{sum(V)}$$

The validity of a characteristic $S = (S^+, S^-)$ with respect to $G[K]$ is given by

$$valid(S,K) \equiv \bigwedge_{v \in K} \left(\left(\bigwedge_{c_i \in S^+} \delta_{gain(c_i,v)>0} \right) \bigwedge \left(\bigwedge_{c_i \in S^-} \delta_{gain(c_i,v)<0} \right) \right)$$

$valid(S,K)$ means that each vertex $v \in K$ has a positive gain for each category $c_i \in S^+$, and a negative gain for each category $c_i \in S^-$. The quality of a characteristic S can be globally measured by the numerical function A:

$$A(S,K) = gain(S^+,K) - gain(S^-,K)$$

However, a major drawback of the gain is that it is easy to obtain high value with highly specific characteristics [Lavrac *et al.*, 2004], more precisely characteristics associated to a small set of vertices. Once again, we use the Weighted relative accuracy which makes trade-off between generality and gain by considering the relative size of the subgraph.

$$WRAcc(S,K) = \begin{cases} A(S,K) \times \frac{sum(K)}{sum(V)} \text{ if } valid(S,K) \\ 0 \text{ otherwise} \end{cases}$$

The main differences with the $WRAcc$ used in Subgroup Discovery [Lavrac *et al.*, 2004] are (1) our adapted $WRAcc$ considers both the positive and the negative contrasts in an unsupervised setting (i.e., there is no class attribute in our setting, the "target" is settled by each pattern), (2) it takes into account the homogeneity of elements of K, using the predicate $valid(S,K)$.

In [Bendimerad *et al.*, 2016], we used a slightly different $Wracc$ measure that differs by its normalization factor (i.e., $\frac{|K|}{|V|}$ was used instead of $\frac{sum(K)}{sum(V)}$ in this chapter and in [Bendimerad *et al.*, 2018]). This new coefficient makes it possible to correct the defect of the previous measure consisting in fostering sparse areas.

We now define the pattern domain we consider:

Definition 4.9 (Exceptional subgraph)**.** *Given a graph $G = (V,E,C,D)$ and two thresholds σ and δ, an exceptional subgraph (S,K) is such that (1) $|K| \geq \sigma$, (2) $G[K]$ is connected, and (3) $WRAcc(S,K) \geq \delta$.*

Given an exceptional subgraph (S,K), a large number of less specific subgraphs can be derived, i.e. patterns (S',K') such that $S' \subseteq S$ and $K' \subseteq K$. As these patterns (S',K') are already described and covered by (S,K), they unnecessarily increase the size of the solution set. This redundancy can be avoided thanks to a closure operator [Kuznetsov, 1999] defined below.

Definition 4.10 (Formal concept)**.** *Let f and g be two closure operators forming a Galois connection:*

- *$f : 2^V \rightarrow \mathscr{S}$, that provides the most specific characteristic associated to the subgraph induced by $K \subseteq V$:*

$$f(K) = \left(\{c_i \in C \mid \bigwedge_{v \in K} \delta_{gain(c_i,v)>0} \}, \{c_i \in C \mid \bigwedge_{v \in K} \delta_{gain(c_i,v)<0} \} \right)$$

- *$g : \mathscr{S} \rightarrow 2^V$, that returns the set of vertices supporting the characteristic S:*

$$g(S) = \{v \in V \mid valid(S,\{v\})\}$$

A pair (S,K), with $S \in \mathscr{S}$ and $K \subseteq V$, is a formal concept iff $S = f(g(S))$ and $K = g(S)$, or equivalently, $S = f(K)$ and $K = g(f(K))$.

It may happen that a formal concept as defined above does not correspond to a connected subgraph. For example, in Fig. 4.9, (S,K) is a formal concept, with $S = (\{c_1\}^+, \{c_2\}^-)$ and $K = \{v_1, v_3, v_4, v_6\}$. However, (S,K) is not an exceptional subgraph because $G[K]$ is not connected. Maximal patterns address this limitation:

Definition 4.11 (Maximal pattern)**.** *A set of maximal patterns is derived from a formal concept (S,K) as:*

$$\{(f(CC),CC) \mid CC \text{ is a connected component of } G[K]\}$$

In other terms, a maximal pattern $(f(CC),CC)$ is made of the most specific characteristic for CC, but also, the connected subgraph $G[CC]$ cannot be extended to another connected subgraph while keeping the current characteristic $f(CC)$.

Following our example in Fig. 4.9, the formal concept (S,K) contains two connected components $CC_1 = \{v_1, v_4\}$ and $CC_2 = \{v_3, v_6\}$, with $f(CC_1) = (\{c_1, c_3\}^+, \{c_2\}^-)$ and $f(CC_2) = (\{c_1, c_4\}^+, \{c_2\}^-)$. From these two connected components, two maximal patterns $(f(CC_1),CC_1)$ and $(f(CC_2),CC_2)$ are derived.

Finally, all these definitions are used to establish the notion of closed exceptional subgraph:

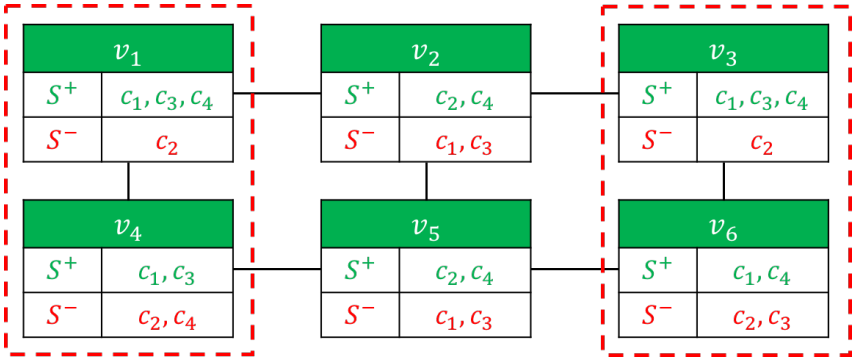


Figure 4.9: Example of a formal concept (S,K), with $S = (\{c_1\}^+, \{c_2\}^-)$, $K = \{v_1, v_3, v_4, v_6\}$ such that $G[K]$ is not connected.

Definition 4.12 (Closed exceptional subgraph)**.** *Let $S \in \mathscr{S}$ be a characteristic and $K \subseteq V$ a subset of vertices, (S,K) is a closed exceptional subgraph iff (1) (S,K) is a maximal pattern (2) (S,K) is an exceptional subgraph.*

4.3.2 Algorithms

To discover closed exceptional subgraphs, we devised two different algorithms. The first one is a sound and complete branch-and-bound algorithm. The second one is a heuristic algorithm that samples the space of closed exceptional subgraphs within a user-defined time-budget. This approach makes possible to obtain instant results and to successfully scale up on datasets with a large number of attributes for which an exact algorithm fails.

The complete approach

In order to enumerate the set of all closed exceptional subgraphs, we explore the space of characteristics $S = (S^+, S^-)$, and for each characteristic, we enumerate the maximal patterns that can be generated from S using the closure operators. We start from an empty characteristic $(S^+, S^-) = (\emptyset, \emptyset)$ and consider the candidate categories that can be used to expand S: $X = (X^+, X^-)$: X^+ contains the categories that can be added to S^+, and X^- the ones that can be added to S^-. $Y \subseteq V$ represents a set of vertices that verifies $valid(S,Y)$. Initially, Y contains all the vertices V, and $(X^+, X^-) = (C,C)$. In each recursive call of CENERGETICS, S is extended with an element x of X^+ or of X^-. Y is then reduced to the vertices v that satisfy $valid(S \cup \{x\}, \{v\})$.

The predicate *valid* is anti-monotone with respect to the inclusion of characteristics: Considering two characteristics S_1, S_2 such that $S_1 \subseteq S_2$ and $K \subseteq V$, we have $valid(S_2,K) \Rightarrow valid(S_1,K)$. By the contraposition, the invalid vertices for S_1 are also invalid for S_2, and therefore, the valid set of vertices associated to $S \cup \{x\}$ is a subset of Y. We also take benefit from this anti-monotony using the fail first principle: To extend the current characteristic S, we choose the characteristic x for which the set $\{v \in Y \mid valid(S \cup \{x\}, \{v\})\}$ is the smallest. After updating Y, we explore each connected component CC of $G[Y]$ independently and form $(f(CC), CC)$ that is, by definition, a maximal pattern. If $f(CC) \subseteq S \cup X$, then the maximal pattern $(f(CC), CC)$ has not yet been explored and `MINTAG` is recursively called with $S = f(CC)$ and $Y = CC$ (Line **??**). This allows to explore only characteristics S and vertices subsets Y that form maximal patterns (S,Y), and without redundancy.

Additionally, another pruning mechanism based on an upper bound on the *WRAcc* measure (i.e., relying on the aggregation property of the *WRAcc* measure) is used.

The exceptional subgraph space sampling approach

In practice, end-users want to obtain high-quality patterns in a short amount of time, especially in interactive data mining processes. However, MINTAG may require a considerable amount of time for graph with large number of attributes. To overcome this issue, we propose an approach that computes a sampling of the closed exceptional subgraphs within a user-given time-budget.

We adapt the randomized pattern mining technique of [Boley *et al.*, 2011] to exceptional subgraphs discovery. This so-called *Controlled Direct Pattern Sampling* enables the user to specify a time budget and computes a set of high-quality patterns whose size directly depends on the specified amount of time.

The idea consists of sampling the patterns based on a probability distribution that rewards high-quality patterns. In a first attempt, we proposed to first sample the characteristics and then derive the associated subgraphs. But this strategy failed in computing patterns with high WRAcc values because the graph structure was neglected. Thus, we adopted the reverse approach that consists in randomly generating maximal patterns (S,K).

We perform a random walk on a graph whose vertices are the maximal patterns and the edges connect couple of patterns (S_1,K_1) and (S_2,K_2) such that $K_1 \subseteq K_2$ and there does not exist a maximal pattern (S,K) such that $K_1 \subset K \subset K_2$ (strict inclusion).

To define how is constructed the graph on which the random walk is performed, we need to introduce two new functions

- $comp : 2^V \times 2^V \rightarrow 2^V$: Given two subsets of vertices H and K such that $K \subseteq H$ and $G[K]$ is connected, $comp(K,H)$ returns the connected component of H that contains K.
- $clo : 2^V \rightarrow 2^V$: Given a connected subgraph induced by K, $clo(K)$ returns the part of the closure of K that is connected and contains K:

$$clo(K) = comp(K, g(f(K)))$$

  $clo(K)$ can be computed by extending K recursively with all neighbors v that maintain $f(K \cup \{v\}) = f(K)$.

During the random walk, edges (transitions) are chosen following a probability measure that favors high-quality patterns:

1. The random walk starts by drawing a first vertex using the probability $\mathscr{P}(\{v\}) = \frac{WRAcc(f(\{v\}),clo(\{v\}))}{\sum_{u \in V} WRAcc(f(\{u\}),clo(\{u\}))}$ to form the first explored maximal pattern $(f(\{v\}), clo(\{v\}))$.
2. A new maximal pattern is generated from the pattern (S,K) by considering all maximal patterns that are direct super-sets of K. Such patterns are generated by alternatively adding a neighbor element $v \in N(K) \setminus K$ to K and considering the closure $clo(K \cup \{v\})$. $N(K)$ is the set of neighbors of K: $N(K) = \{v \in V \mid \exists u \in K : (u,v) \in E\}$. (S,K) is also considered among the patterns that can be generated in the next step. The set $Next(K)$ of all possible next subgraphs is then:

$$Next(K) = \{K\} \cup \{clo(K \cup \{v\}) \mid v \in N(K) \setminus K\}$$

   Thus, from $Next(K)$, all the direct successors to (S,K) can be enumerated by:

$$\{(S',K') \mid K' \in Next(K) \text{ and } S' = f(K')\}$$

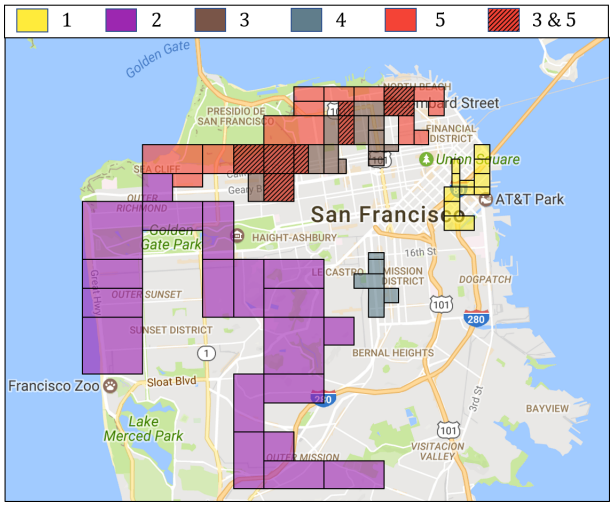
   The next random step is drawn based on the probability $\mathscr{P}(K' \mid K)$, that is the probability to reach $K' \in Next(K)$ from K: $\mathscr{P}(K' \mid K) = \frac{WRAcc(f(K'),K')}{\sum_{K_2 \in Next(K)} WRAcc(f(K_2),K_2)}$. This distribution of probabilities rewards transitions toward maximal patterns with large $WRAcc(f(K'),K')$ value.
3. The current random walk stops when $K' = K$ and a new one is started from step (1). Otherwise, the random walk continues by repeating Step (2) on the set of vertices K'. At each step of the random walk, if $WRAcc(f(K),K) \geq \delta$ and $|K| \geq \sigma$, the pattern is added to the output result set.

4.3.3 Exceptional Subgraphs For Urban Data Analysis

We considered 10 real-world datasets whose characteristics are given in Table 4.3. Eight of them come from [Falher *et al.*, 2015] and depict Foursquare venues over 4 US and 4 EU important cities. The venues are described by a hierarchy[15]. We consider the first level (10 attributes) in the first series of experiments and the second level (around 300 attributes) for the second ones. *SF. Crimes* data[16] are provided by a Kaggle challenge and describe the criminal activity in San Francisco. Finally, *San Francisco C&V* is the combination – after normalization – of *SF. Crimes* and Foursquare data over San Francisco. Each city is divided into rectangular zones in such a way that each rectangle contains a minimal number of venues.

dataset	$\|V\|$	$\|E\|$	$\|C\|$	**#objects**
New York	292	647	10 (356)	71954 venues
San Francisco	124	256	10 (328)	21654 venues
S.F. Crimes	898	2172	39	878049 crimes
S.F. C&V	342	767	49 (328)	878049 cr. + 21654 ven.

Table 4.3: Description of the real-world datasets



Id	S^+ Venues	S^+ Crimes	$WRAcc$	Size
P_1	Professional & other places	Embezzlement	0.007	11
P_2	Residence	Vandalism, Vehicle theft, Burglary	0.006	19
P_3	Shop & services	Burglary	0.06	21
P_4	Shop & services, Nightlife spot, Art & Entertainment, Food	Drunkenness	0.003	5
P_5	Outdoors & recreation	Burglary	0.002	20

Figure 4.10: Patterns discovered in San Francisco crimes and venues dataset (49 attributes)

We use CENERGETICS on San Francisco crimes and venues dataset to automatically identify typical areas of this city. Figure 4.10 displays 5 discovered patterns. Pattern P_1 depicts neighbourhoods with a high concentration of venues of type professional & other places, and crimes of type embezzlement. This

[15]https://developer.foursquare.com/categorytree
[16]https://www.kaggle.com/c/sf-crime

can be explained by its proximity to the Financial District. P_2 is a neighborhood located in the West and South-West of San Francisco. It contains a positive contrast of residences and crimes of type vandalism, vehicle theft, burglary. These crimes are known as the most common types of crimes in residential areas. P_3 and P_5 are overlapping patterns located in the North of the city. They characterize areas with a high concentration of venues of type shop & services, outdoors & recreation, and crimes of type burglary. Pattern P_4 describes a neighborhood with a positive contrast of crimes related to drunkenness, which can be explained by the high concentration of nightlife spots in this area.

We also report 9 discovered patterns on New York venues dataset. They are presented in Fig. 4.11. Four of them (on the left-hand side map) are discovered on the dataset with 10 attributes, whereas the 5 remaining ones (on the right-hand side map) are discovered on the dataset with 356 attributes. P_1 is located in the South of Central Park. This neighborhood is known to be a business and professional area with a low concentration of residences. A sub-area of P_1 is depicted by P_5 with a high concentration of offices, buildings, medical centers. P_2 describes areas with a high proportion of venues of type outdoors & recreation. It contains Central Park and some areas located near East River and Hudson River. P_3 covers a part of the South of Manhattan and the North of Brooklyn, with a high concentration of nightlife spots. P_4 covers John F. Kennedy and LaGuardia Airports and their surroundings. This explains the high presence of travel & transport venues. More precisely, P_9 contains neighborhoods of John F. Kennedy Airport, and it depicts them with venues of types: Taxi, parkings, donut shops, airport, and general travels. Both P_6 and P_8 represents areas with high proportion of residences. P_8 is also characterized with an important concentration of food & drink shops. P_7 is another pattern that describes a part of South Manhattan with a high concentration of offices.

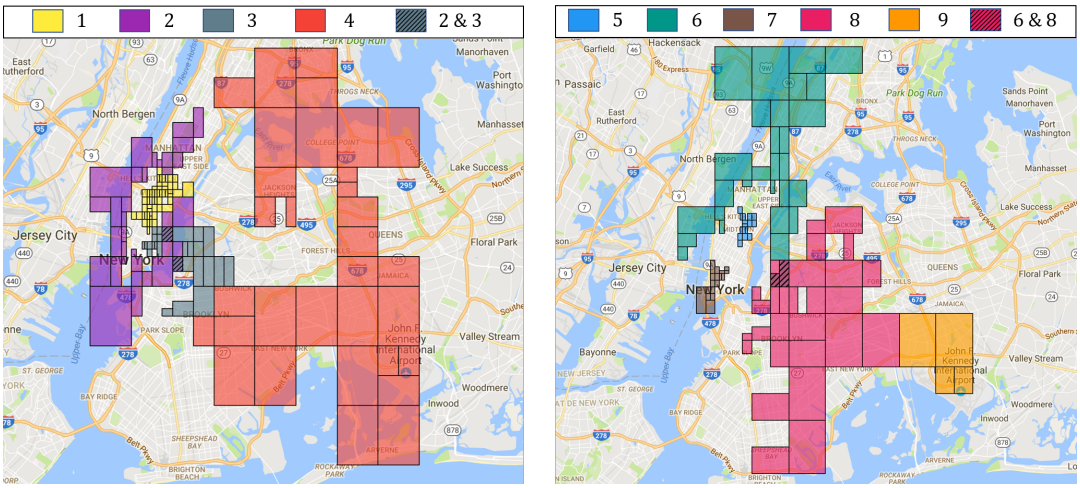
Besides, we mined exceptional subgraphs on the different cities. In most of them (e.g., Barcelona, Paris, Rome, Los Angeles, London), the nightlife spots are mainly located in the city center. The higher concentration of outdoor & recreation places is surrounding for London. For seaside towns, they are concentrated on the coasts.

4.4 Conclusion

Discovering the main trends (i.e., regularities) in graphs is often not enough for the end-user. In most of the cases, she is the data owner and she already has a good view of the “big pictures". Considering this, the discovery of exceptional subgraphs is more satisfactory since it can provide her new insights. We presented in this chapter two different approaches to discover exceptional subgraphs in attributed graphs. The first one (Section 4.2) focuses on the edge attributes to discover exceptional subgraphs. It is well adapted to analyze graphs that depict mobility data. Notice that this approach allows to take into account vertex attributes. However these attributes are handled as edge attribute and their full consideration may impact the quality measure defined to assess how exceptional are the subgraphs. The second approach we presented (Section 4.3) is devoted to vertex attributed graphs. It is particularly well adapted when the graph structure is fixed and does not evolve as for instance to characterize some areas of a city.

This chapter opens up several avenues for further research.

- From an algorithmic point of view, more efficient algorithms that forget the completeness can be developed. An exhaustive search could be also coupled to heuristic approach to explore the neighborhood a good solution returned by the heuristic approach. Also, output sampling based approach can help to set to tune the good parameters and constraints for a “local" exhaustive search.
- From an application point of view, this work can support new applications and problems. For instance, it could be interesting to analyze set of cities, finding intra-city and inter-city exceptional subgraphs. This requires the definition of the exceptional subgraph within a collection of graphs. We also applied CENERGETICS on fMRI data in order to better understand olfactory perception



Id	S^+	S^-	$WRAcc$	size
P_1	Professional & other places	Residence	0.039	50
P_2	Outdoors & recreation	Shop & services, Food	0.023	35
P_3	Nightlife spot	College & university, Travel & transport, Professional & other places	0.02	27
P_4	Travel & transport	Art & entertainment	0.016	32
P_5	Office, Building, Medical center		0.01	16
P_6	Residential building (apartment / condo), Home (private)		0.008	30
P_7	Office		0.008	17
P_8	Home (private), Food & drink shop		0.007	30
P_9	Taxi, Parking, Donut shop, Airport, General travel		0.006	4

Figure 4.11: Patterns discovered in New York datasets with 10 attributes (patterns P_1 to P_4 plotted on the left-hand side map) 356 attributes (patterns P_5 to P_9 plotted on the right-hand side map).

at the cerebral level [Moranges *et al.*, 2018]. To this end, we modeled fMRI data as an attributed graph whose vertices depict voxels and attributes encode the hemodynamic response related to a series of odorant molecules. We defined four different different ways to analyze the hemodynamic activity within the studied voxels. The strength of such an approach lies in its ability to be more robust to spatial imprecision and inter-individual variability than classical fMRI analysis approaches. This first attempt is promising and motivate the definition of new pattern domains to support the analysis of graphs whose vertices are described by sequences.

- From a user point of view, we have to make effort on how to present the results to the user. Indeed, the subgraphs are exceptional according to a global model. So we have to consider this global model when we present an exceptional subgraph to the user, especially to highlight the differences. To this end, we have to devise some visualization techniques adapted to our context.
- Some perspectives are dealt with in the following chapters, including how to better take into account the domain knowledge or the expert prior for the discovery of graphs of higher interest for her. This can be done in different ways: a finer definition of both the global model (i.e., the big picture) and the local model taking into account some well-known mobility models from statistical physics; the implementation of an interactive process to take into account the user's feedback and finally the consideration of prior knowledge to mine subjectively interesting exceptional subgraphs.

Part II

Integrating Priors and User Interest in Exceptional Attributed Subgraph Mining

Chapter 5

Taking Into Account Domain Knowledge Into Models

Contents

5.1 Introduction

The rapid progress of wireless sensor technologies in mobile environments (e.g., GPS, Wi-Fi, RFID) has lead to the development of new services for individuals, administrations and companies based on the monitoring of people mobility. For instance, mobility profiles play a central role in context-based search and advertising [Bayir *et al.*, 2009], location modeling [Ashbrook and Starner, 2003], traffic planning and route prediction [Harrington and Cahill, 2004], air pollution exposure estimation [Demirbas *et al.*, 2009]. Especially, the last decade has witnessed a huge growth in the analysis of mobility [Giannotti and Pedreschi, 2008]. These studies focus only on mining trajectories and their applications [Li *et al.*, 2010, Luo *et al.*, 2013] [Monreale *et al.*, 2009, Wang *et al.*, 2011, Zheng *et al.*, 2009], but do not take into account the contextual information of the individual trajectories. Related contexts of the trajectories are essential data to produce accurate and valuable models of mobility patterns. Recent work has opened this way [Atzmueller, 2016, Lemmerich *et al.*, 2016] and also [Kaytoue *et al.*, 2017] introduced in Chapter 4.

Considering a network whose vertices depict places or points of interest (POI) and edges stand for trips and are labeled by sets of transactions[17], that correspond to characteristics of the travels, the problem considered hereafter is to discover and characterize geographical areas that are attractive places and routes for specific contexts. Such areas are frequently visited together in certain contexts, e.g. by users of similar profiles or with trips of similar characteristics.

The problem is thus to identify contextual subgraphs (i.e., subgraphs related to a context) that are exceptional in the sense of EMM/SD [Leman *et al.*, 2008]. EMM aims to build a model on the whole

[17]To avoid any ambiguity, notice that the term transaction is different from the one used in Database Management System

data and a model on data related to a context (generally called pattern), and the exceptionality is assessed by a quality measure that compares the two models. In Chapter 4, we introduced this problem for network analysis. We used a simple model based on the number of trips associated with an edge. In other words, this model is agnostic and does not take into account some domain knowledge about the trajectories (e.g., the more distant (or the less populated) two areas are, the more surprising a trajectory between them is).

In this chapter, we show that well-known models from Statistical Physics are better for analyzing trajectories and they can be easily integrated into the quality measure and in our algorithm COSMIc to discovery exceptional subgraphs defined in Chapter 4. We study exceptional subgraphs with respect to mobility models that take into consideration the length or the surrounding demography of the trips. The two models we consider in this chapter are the gravity model [Zipf, 1946] and the radiation model [Simini *et al.*, 2011]. Through these models, we take into account domain knowledge.

This work was done in the context of Anes Bendimerad's PhD thesis in collaboration with Rémy Cazabet and Céline Robardet [Bendimerad *et al.*, 2017a].

5.2 Exceptional contextual subgraphs

In this section, we briefly recall the concepts introduced in Section 4.2 for exceptional subgraphs in edge-attributed graphs.

We define a graph $G = (V, E, T, \text{EDGE})$ to model urban mobility. V denotes city areas, E denotes the trips from one area to another, and T is a set of transactions of a relation R of schema $S_R = [R_1, \ldots, R_p]$. Each attribute R_i takes values in $\mathbf{dom}(R_i)$ that is either nominal or numerical. Thus a transaction $t \in R$ is a tuple $(t_1, \ldots, t_p)$ with $t_i \in \mathbf{dom}(R_i)$. EDGE is a mapping of a transaction to an edge: $\text{EDGE} : T \to E$. For example, if $t_k \in R$ is a transaction such that $\text{EDGE}(t_k) = e_{ij} = (v_i, v_j)$, t_k is a trip from v_i to v_j.

Besides the definition of G, we introduce the notion of *context* and its use to select subgraphs on G.

Let a context be a tuple $C = (C_1, \ldots, C_p)$ with C_i a restriction on $\mathbf{dom}(R_i)$. C_i can take two different forms depending on the type of R_i:

- If R_i is nominal, $C_i = a$ or $C_i = \star_i$, with $a \in \mathbf{dom}(R_i)$ and $\star_i = \mathbf{dom}(R_i)$
- If R_i is numerical, $C_i = [a, b]$, with $a, b \in \mathbf{dom}(R_i)$, $a < b$.

We say that a transaction $(t_1, \ldots, t_p)$ *satisfies or supports* a context C iff $\forall i = 1 \ldots p$, $t_i \in C_i$.

Given a context C, the *contextual graph* derived from G is the weighted graph $G_C = (V, E, W_C)$ where $W_C(e)$ is the number of transactions associated to e that satisfy $C = (C_1, \ldots, C_p)$:

$$W_C(e) = |\{t = (t_1, \ldots, t_p) \mid \text{EDGE}(t) = e \text{ and } \forall i = 1 \ldots t, t_i \in C_i\}|$$

Several contexts may be associated to the same contextual graph (i.e. in case of local dependencies between attribute values) and, in such cases, we retain the (unique) most specific one, also called *closed context*. It is defined up to an order relation $\preceq$ defined as follows: A context C^1 is said to be more specific than a context C^2, denoted $C^1 \preceq C^2$, iff:

- $C^2_i = \star_i$ or $C^1_i = C^2_i = a \in \mathbf{dom}(R_i)$, for R_i a nominal attribute,
- $[a^1_i, b^1_i] \subseteq [a^2_i, b^2_i]$ with $C^1_i = [a^1_i, b^1_i]$ and $C^2_i = [a^2_i, b^2_i]$, for all numerical attributes R_i.

A context C is thus *closed* iff $\forall C'$ such that $G_C = G_{C'}$, $C \preceq C'$.

In EMM approach, a pattern (in our case a context) interest is evaluated by its deviation from a null model. An edge $e \in E$ is considered to be exceptional with respect to a context C, if the observed weight $W_C(e)$ is large compared to the expected weight $\widehat{W_C(e)}$.

Several discriminative measures can be used. We consider the Weighted Relative Accuracy ($WRAcc$) [Lavrac *et al.*, 1999] widely used in supervised pattern mining:

$$WRAcc(C,e) = \frac{1}{W_\star(E)} \times \left(W_C(e) - \widehat{W_C(e)} \right)$$

with $W_\star(E) = \sum_{x \in E} W_\star(x)$ and $\star = (\mathbf{dom}(R_1), \ldots, \mathbf{dom}(R_p))$

This measure was also used in [Kaytoue *et al.*, 2017] in which the expected weight is defined as:

$$\widehat{W_C(e)} = W_\star(e) \times \frac{W_C(E)}{W_\star(E)} \tag{5.1}$$

This gives the standard definition of the *WRAcc* measure where the expected weight is a portion of the total weight $W_\star(e)$. In the following, this definition of $\widehat{W_C(e)}$ is denoted $\mathscr{M}_0$. This mobility model is rather simplistic because it does not take into account the length of trips or the surrounding demographics. In the next section, we propose to use two other mobility models $\mathscr{M}_g$ and $\mathscr{M}_r$ of expected weights.

5.3 Mobility models

In mobility modeling [Masucci *et al.*, 2013], urban travels depend on the distances and level of attraction. Thus, the expected amount of transactions between any pair of vertices (v_i, v_j) is not uniform but depends on the distance d_{ij} between v_i and v_j, and on the surrounding populations (or level of attraction) n_i and n_j of these vertices. $W_\star(e_{ij})$ can therefore be considered exceptional, given its associated values of d_{ij}, n_i and n_j, whereas such situations can not be identified by standard *WRacc* measure. For example, $W_\star(e_{ij})$ can be very large for two points of interest v_i and v_j, while we expect much lower $\widehat{W_\star(e_{ij})}$ with regard to their distance d_{ij} and/or their population n_i and n_j. We propose to model the expected weight of an edge using the gravity or the radiation models as defined below:

$$\widehat{W_C(e)} = m(e) \times \frac{W_C(E)}{W_\star(E)}$$

with $m(e)$ the mobility model, denoted in the following either $g(e)$ for the gravity model, or $r(e)$ for the radiation one.

5.3.1 The gravity model

In the gravity model, the most widely used mobility model, the number of expected transactions $g(e_{ij})$ between v_i and v_j is defined as [Masucci *et al.*, 2013]:

$$g(e_{ij}) = n_i \times n_j \times f(d_{ij})$$

where $f(d_{ij})$ is known as the *deterrence function* and represents the influence of the distance. In the traditional form of the gravity model, $f(d_{ij})$ is a priori defined as $\frac{1}{d_{ij}^\gamma}$, with γ an optional parameter, usually tuned by regression analysis. However, some recent papers [Expert *et al.*, 2011, Cazabet *et al.*, 2017] have shown better results using a deterrence function learned from data as follows:

$$f(d) = \frac{\sum_{x,y|d_{xy}=d} W_\star(e_{xy})}{\sum_{x,y|d_{xy}=d} n_x \times n_y} \tag{5.2}$$

In the rest of this article, we will refer to this model as *gravity*, and denote it $\mathscr{M}_g$.

5.3.2 The radiation model

Some authors have criticized the gravity model, since it is unable to predict different fluxes between locations with similar densities and at the same distance, but have different population densities between them. They therefore proposed the radiation model, for which the expected number of transactions between two points depends on the number and size of other points around them. Formally, the number of expected transactions between v_i and v_j using the radiation model [Simini *et al.*, 2011] is given by:

$$r_0(e_{ij}) = \frac{T_i}{1-\frac{n_i}{N}} \times \frac{n_i \times n_j}{(n_i+s_{ij}) \times (n_i+n_j+s_{ij})}$$

with:

- s_{ij} the population in a circle whose center is v_i and radius d_{ij} minus n_i and n_j.
- $T_i = \sum_j W_\star(e_{ij})$.
- N the total population.

While the original radiation model is parameter free, and thus do not have a deterrence function, a recent improvement [Sarzynska *et al.*, 2016] has been proposed. Using the same mechanism as the one used in gravity model to learn a deterrence function from data, it is defined as:

$$r(e_{ij}) = r_0(e_{ij}) \times f_2(d_{ij})$$

where $f_2(d_{ij})$ is the deterrence function defined as:

$$f_2(d) = \frac{\sum_{x,y|d_{xy}=d} W_\star(e_{xy})}{\sum_{x,y|d_{xy}=d} r(e_{xy})} \tag{5.3}$$

We will refer to this model as *radiation*, and denote it $\mathscr{M}_r$.

5.4 Experiments on VÉLO'V data

In this section, we compare the three models $\mathscr{M}_0$, $\mathscr{M}_g$ and $\mathscr{M}_r$ using VÉLO'V dataset already investigated in Chapter 4. To this end, we first present the data and we show some statistics about them. Second, we study the distribution of transactions predicted by each model. Finally, we compare the result patterns detected using these different models.

We use the VÉLO'V dataset. VÉLO'V is the bike-sharing system run by the city of Lyon (France) and the company JCDecaux[18]. There are a total of 348 VÉLO'V stations across the city of Lyon. Our experiments are performed on trips collected on October 2011. The overall number of trips is 565,065 transactions. Each trip includes the bicycle station and the time-stamp for both departure and arrival, as well as some basic demographics about the users (gender, age, zip code, country of residence, type of pass). Hence, the VÉLO'V stations are the graph vertices ($|V| = 348$), and directed edges correspond to the fact that a VÉLO'V user checks out a bicycle at a station and returns it at another.

Fig. 5.1 (left) reports the distribution of populations of areas containing stations. This figure shows that the population is not uniform. Thus, it will inevitably influence the results of gravity and radiation models. Fig. 5.1 (right) shows the distribution of distance of transactions. There is only few transactions with distances less than 200 meters. In fact, people find it useless to take bikes for very short distances. The number of transactions increases when the distance increases until it reaches its maximum values for distances around 1km and 2km. After that, the number of transactions decreases.

[18] `http://www.velov.grandlyon.com/`

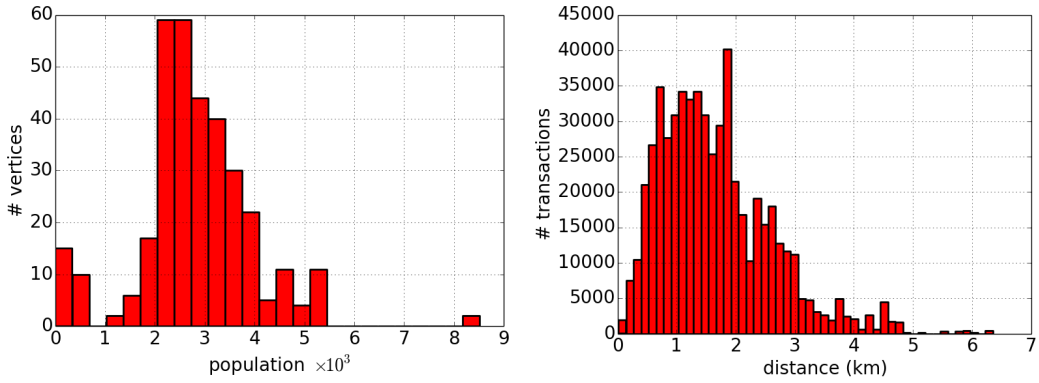


Figure 5.1: Statistics about the VÉLO'V dataset, (left) distribution populations of areas containing stations (right) distribution of distance of transactions.

Fig. 5.2 and 5.3 compare expected edges starting from specific stations located respectively in Part Dieu and Cordeliers. They correspond to edges that contain at least 10 expected transactions. Recall that in $\mathscr{M}_0$ model, the expected number of transactions is simply the observed number of transactions. For example, if we take the station located in Part-Dieu, in $\mathscr{M}_0$ model some edges are connected to areas located in the north west even if they are far away from Part Dieu. This exceptionality will be captured by mobility models since these edges are not expected by them. Meanwhile, it is clear that the distribution of expected transactions strongly depend on the distance for spacial models.

$\mathscr{M}_0$ $\mathscr{M}_g$ $\mathscr{M}_r$

Figure 5.2: Comparison of expected edges starting from a station located in Part Dieu.

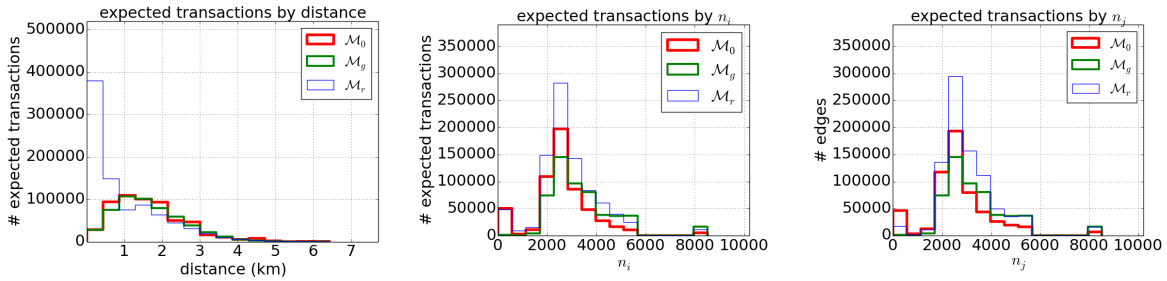
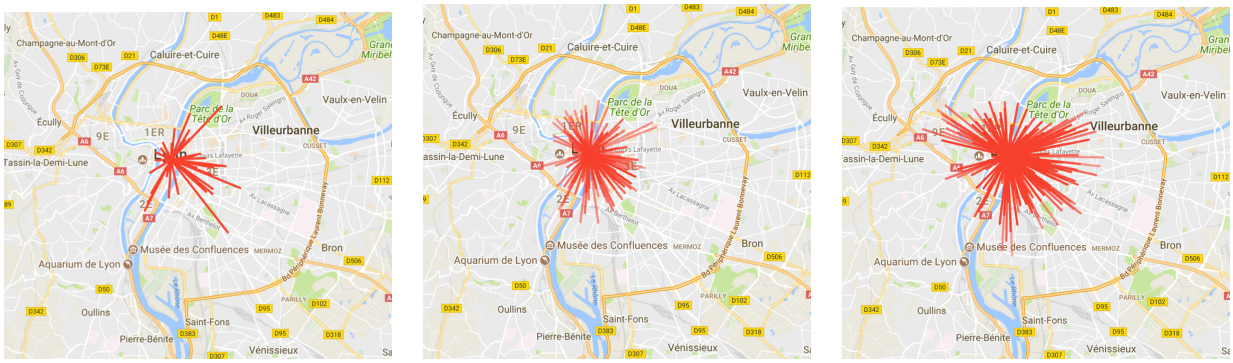


Figure 5.4: Distribution of expected transactions with respect to n_i (source population), n_j (destination population), d_{ij} (distance between source and destination), and $WRAcc$ values, for each model.



$\mathcal{M}_0$ $\mathcal{M}_g$ $\mathcal{M}_r$

Figure 5.3: Comparison of expected edges starting from a station located in Cordeliers.

Fig. 5.4 shows the distribution of expected transactions with respect to the following parameters: d_{ij} (distance between source and destination), n_i (source population), and n_j (destination population). Distributions of transactions by distance are similar for $\mathcal{M}_0$ and $\mathcal{M}_g$, while $\mathcal{M}_r$ expects much more for distances less than 1km. For the two other distributions, $\mathcal{M}_r$ expects more than the other models in most of the cases. Also, the number of expected transactions by $\mathcal{M}_0$ is higher than $\mathcal{M}_g$ when n_i and n_j are less than 3000, but it is lower when n_i and n_j exceed 3000. In fact, $\mathcal{M}_g$ expects more transactions when n_i and n_j are greater.

We computed the result patterns and we compared their ranking with respect to the three different models. To this end, we ranked the patterns based on their $WRAcc$ scores using each model, and we computed the Kendall tau between these rankings. Results are depicted in Fig. 5.5. As expected, the greatest kendall tau is between $\mathcal{M}_g$ and $\mathcal{M}_r$, which is logical since they are spacial models based on similar information. The Kendall tau of $\mathcal{M}_0$ with the spacial models is lower, especially with $\mathcal{M}_r$.

Fig. 5.7 reports the top 5 patterns for each model. Gravity and radiation models have found exactly the same 5 best patterns, whereas $\mathcal{M}_0$ model differs in some of them. In order to verify whether the spacial models have a real impact in the top patterns, we have compared the average value of $\frac{n_i \times n_j}{d_{ij}}$ of the transactions that appear in the top 10 patterns in each model. Fig. 5.6 presents the results. This value is significantly lower in the spacial models comparing with the $\mathcal{M}_0$ one. It means that the spacial models overweight the $WRAcc$ quality of transactions with low values of $\frac{n_i \times n_j}{d_{ij}}$.

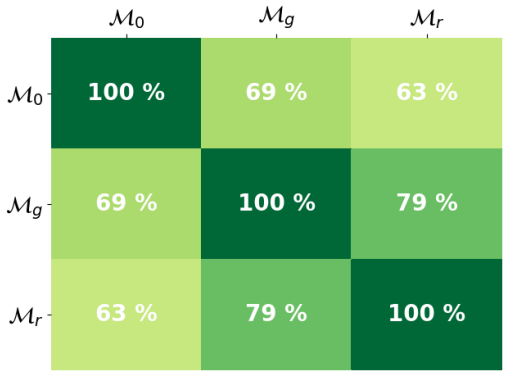


Figure 5.5: Kendall tau of ranking of results based on the three different models $\mathscr{M}_0$, $\mathscr{M}_g$ and $\mathscr{M}_r$.

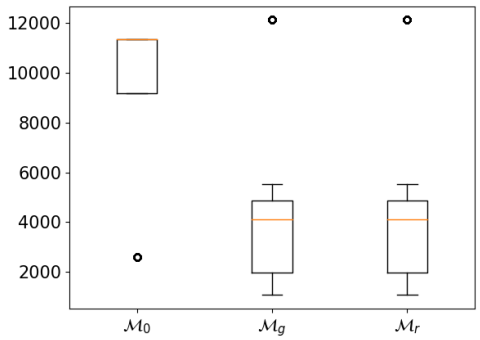


Figure 5.6: Boxplots of $avg(\frac{n_i \times n_j}{d_{ij}})$ of exceptional transactions of each model.

5.5 Conclusion

Taking into account domain knowledge can make a big qualitative difference within the KDD process by identifying patterns of higher interest for the end-user. In this chapter, we demonstrated, in the context of trajectory analysis, that mobility models borrowed in Statistical Physics can be easily integrated into quality measures without requiring changes on the algorithm to discover exceptional subgraphs.

Designing new pattern domain and devising efficient and effective algorithms are the main skills of pattern miners but it is not enough. To assure the discovery of high quality patterns, they have to take into account domain knowledge. This requires curiosity and the review of the literature for each studied domain. The consideration of domain knowledge is the corner stone of multidisciplinary approach. Generic principles must be adapted to a domain that itself may raise new data mining challenges.

Figure 5.7: Top 5 patterns found by each model $\mathscr{M}_0$ $\mathscr{M}_g$ and $\mathscr{M}_r$.

Chapter 6

Taking Into Account User Feedback Into Biased Quality Measures: The Case of Geolocated Event Detection In Social Media

Contents

6.1 Introduction

Social microblogging (Twitter, Weibo, Instagram, etc.) gives people the ability to interact at a worldwide scale by broadcasting their interests, feelings, reactions to their daily life and surrounding events. As such, they are an incredibly rich mean to know the pulse of the world, or of a specific neighborhood, in real time. Analyzing the abundant user-generated content can provide high valued information. Social media data have been analysed for several purposes, e.g. to understand the concerns of a population [Xiao *et al.*, 2016], study disease dynamics [Carchiolo *et al.*, 2015], or predict real-world outcomes [Asur and Huberman, 2010].

Event detection has long been a research topic and has received much attention in the data mining community over the last decade [Chen and Neill, 2014, Dong *et al.*, 2015, Li *et al.*, 2012]. Whereas real-life events are considered as phenomena that unfold over space and time, from a data mining point of view they are conventionally regarded as a set of terms (e.g. hashtags, user mentions, words) whose frequency bursts [Kleinberg, 2002]. However, a bursting term is not necessarily related to an event, and real-life events can be blurred by pointless bursting terms. Several terms can depict the same event and mining methods must address problems related to synonymy. In this work, we address these pitfalls by considering geolocated events. Focusing on terms that burst for a geographical area meets the natural and intuitive notion of event. Extracting such events remains challenging because geolocated events depict small scale phenomena that are covered by much fewer terms than global ones. For instance, in a very large city like New York, dozens of events can take place simultaneously and it can be tricky to find those that stimulate users' curiosity.

This chapter addresses these concerns by (1) proposing efficient algorithms to detect geolocated events and (2) integrating user interests to bias the extraction process toward user preferences. This is the first attempt to the discovery of user-driven events through an interactive discovery process. In our approach, Twitter posts are modeled as a labeled graph whose vertices encode geographical areas and edges depict neighborhood relationships. Time series of term occurrences are associated to vertices. In this framework, a geolocated event is considered as a connected subgraph, a time interval and a set of terms whose number of occurrences in this space-time region is greater than the threshold. Events are extracted in a time window and the user has the possibility to tag those he/she likes. This interactive process makes it possible to extract events of interest to users. User preferences are then integrated in the quality measure used to define and rank events. This measure conveys user interest and promotes events that involve topics or geographical areas the user is interested in.

Figure 6.1 describes our approach. From a social network, we pull up time-stamped geolocated posts published during a given period. This set of posts, B (e.g., tweets), is used to generate a graph G_H of term co-occurrences. A second graph G_V depicts adjacency relations between geographic areas from which posts were emitted. Those graphs are used to guide the event discovery process making it possible to solve neologism and term synonymy problems. Moreover, these graphs are the support of an interactive process with the user and are used to rank the identified events according to his preferences. By liking or not events, the user selects some events that are then used by the algorithm to derive locations and topics of preference. The event detection from the next batch of posts takes advantage of these preferences by weighting more strongly events associated with a preferred geographical area or containing terms of interest. If there is no interaction with the user, only data-driven events are detected.

This work – which is part of Anes Bendimerad's PhD thesis, was done in collaboration with Sihem Amer-Yahia and Céline Robardet. The related paper is still under review. Therefore, we will also describe the algorithms within this chapter contrary to the previous chapters. This work can be seen as a generalization of the pattern domain introduced in Section 4.3 (p. 56) to handle exceptionality in both space and time. Furthermore, we show how easily integrate user-feedback into the quality measure without changing the pattern discovery algorithm. Another interesting point in this chapter is that this is the first time we aim to assess our results using a crowd sourcing platform. It took a lot of work and patience from Anes Bendimerad. More in details, the main contributions of this chapter are the following:

- *Problem formulation.* We define, in a unified view, the problem of user-driven and data-driven geolocated event discovery. We propose an event interestingness measure that is guided by user interests.
- *Algorithms and analysis.* We propose two algorithms to discover geolocated events in social media: `SIGLER-Cov`[19] is based on the generate-and-test paradigm and efficiently exploits upper and lower

[19] `SIGLER` stands for SubjectIve GeoLocated Event discoveRy.

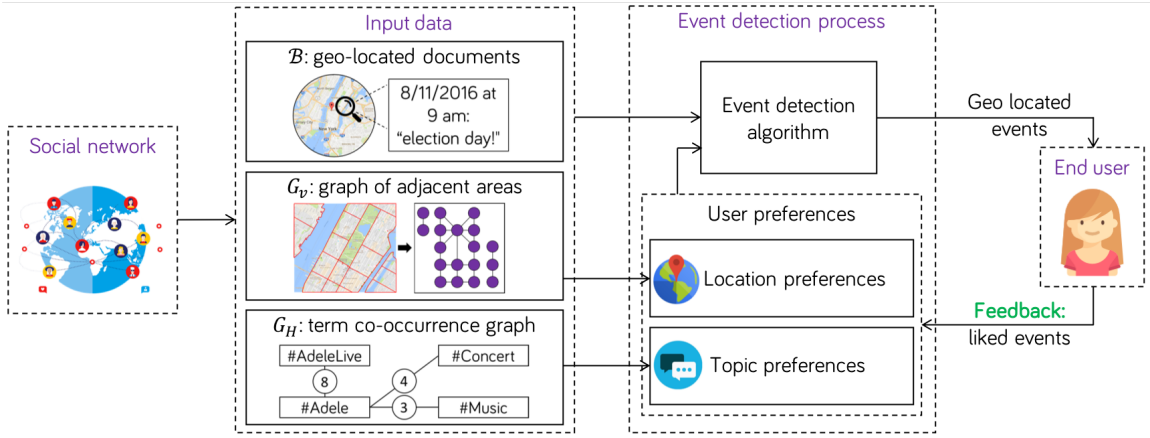


Figure 6.1: Overview of the user-driven event detection system.

bounds as well as constraint properties that make it usable on large-scale microblogging data such as Twitter; `SIGLER-Samp` directly computes a sample of the geolocated events and retrieves interesting geolocated events in a very short time.

- *Evaluation.* We report a thorough empirical study that provides quantitative and qualitative performances of our approach. First, we establish that data driven geolocated events cannot be discovered by non location-aware approaches. Second, we compare our method with MED [Dong *et al.*, 2015] and GeoBurst [Zhang *et al.*, 2016], the state-of-the-art data driven geolocated event mining algorithms, and we demonstrate that `MINTAG` (1) is several orders of magnitude faster than MED and three times faster than GeoBurst, and (2) is more robust to noise compared to both methods. Finally, the ability of our method to extract interesting user-driven events is experimentally validated. To that end, we made experiments using a crowdsourcing platform on several cities (e.g., New York, Los Angeles, London) with different settings (e.g., unpaired and paired samples).
- Source code and datasets are publicly available: `goo.gl/WxQR3v`.

The rest of the chapter is organized as follows. Section 7.2 presents a unified framework for data-driven and user-driven geolocated event detection. Section 6.3 introduces the user-driven approach. Section 7.4 presents our event detection algorithms: `SIGLER-Cov`, an efficient event detection algorithm with coverage guarantee, and `SIGLER-Samp` that directly computes a sampling of the output space of user-diven geolocated events. Section 6.5 provides experimental results. Section 6.6 concludes and provides future directions.

6.2 A Unified Framework For Data-Driven And User-Driven Geolocated Event Discovery

In this section, we introduce the problem of data-driven or user-driven geolocated event discovery in a unified view. From microblogging social media, we consider a set of posts B where each element b is described by: (1) the set of terms that appear in b ($b.terms$), (2) the time at which b was sent ($b.time$), (3) the GPS coordinates of the post of b ($b.loc$).

From this dataset, we introduce the following notations. Let H be the set of terms : $H = \bigcup_{b \in B} b.terms$, $T = [\![t_1, t_m]\!]$ be the ordered set of the m timestamps of the posts[20], and $V = \{v_1, ..., v_n\}$ be n labels that

[20] In our experiments, a timestamp corresponds to an interval of 3 hours

correspond to a discretization of the geographical space. For example, the geographical area defined by the square $[\min_{b\in B} b.loc.x, \max_{b\in B} b.loc.x] \times [\min_{b\in B} b.loc.y, \max_{b\in B} b.loc.y]$ can be divided along a grid and each square is associated to an element of V. Let area : $(X,Y) \to V$ be the function that maps GPS coordinates (X,Y) to V. V is hereafter considered as the vertices of a graph $G_V = (V,E)$ whose edges E connect vertices corresponding to adjacent areas. Our approach is based on the detection of strong variations in term occurrences for a vertex and a timestamp. To this end, we consider the number of occurrences of a term $h \in H$, for a vertex $v \in V$ and a time $t \in T$ as the function f:

$$f(h,v,t) = |\{b \in B \mid (h \in b.terms) \text{ and } (\text{area}(b.loc) = v) \text{ and } (b.time = t)\}|$$

A term is likely to correspond to an event in a space-time zone if its number of occurrences is significantly greater than what is observed in the other times for the same space zone. To that end, we compute the mean $\mu(h,v)$ and standard deviation $\sigma(h,v)$ of f over t. If the difference between $f(h,v,t)$ and its average value is greater than θ times the standard deviation, then the number of occurrences of h is said to be significant for v and t, with $\theta \geq 0$ a parameter whose default value is set to 1. Then, to reduce the impact of very frequent terms, the significance of each term is weighted by the normalized inverse document frequency factor [He *et al.*, 2007]: $\text{idf}(h) = 1 - \frac{\log\left(f(V,T,h)\right)}{\log\left(f(V,T,H)\right)}$. Hence, the score function is:

$$\text{score}(h,v,t) = \Big(f(h,v,t) - \big(\mu(h,v) + \theta\sigma(h,v)\big)\Big) \times \text{idf}(h)$$

with $\sigma(h,v) = \sqrt{\frac{1}{m-1}\sum_{t\in T}\left(f(h,v,t) - \mu(h,v)\right)^2}$ and $\mu(h,v) = \frac{\sum_{t\in T} f(h,v,t)}{m}$. A term h is bursting for (v,t) if its score value is positive. The set of bursting terms for a space-time zone (v,t) is thus defined by: $\mathscr{H}(v,t) = \{h \in H \mid \text{score}(h,v,t) > 0\}$.

We consider as a geolocated pattern P, a tuple (S,I,W) where $S \subseteq V$ induced a connected subgraph in G_V, $I \sqsubseteq T$ is a time interval whose end-points are two elements of T, and $W \subseteq H$ is a set of terms. We aim to identify patterns $P = (S,I,W)$ corresponding to events described by a location S, a time interval I and a set of terms W. The relevance of P is evaluated by the measure $\mathscr{M}(P)$:

$$\mathscr{M}(P) \quad = \quad \sum_{h\in W}\sum_{v\in S}\sum_{t\in I} \text{score}(h,v,t)$$

The larger $\mathscr{M}(P)$, the higher than expected the frequency of the terms in W, which means that the pattern $P = (S,I,W)$ is likely to be an event. In our problem formulation, we are only interested in patterns $P = (S,I,\mathscr{H}(S,I))$ where $\mathscr{H}(S,I) = \bigcap_{v\in S}\bigcap_{t\in I}\mathscr{H}(v,t)$ are terms that burst in all the space-time (S,I), which allows to remove other noisy terms. Thus, in what follows, we allow abusing the notation of a pattern $P = (S,I,\mathscr{H}(S,I))$ by directly noting $P = (S,I)$ (since W is uniquely determined by S and I in this case).

In practice, the interest of an event strongly depends on the end-user. Indeed, a user may be much more interested in events related to some subjects (e.g., sport, music) or may prefer events happening in some specific locations (e.g., near her residence). To take user interests into account, we propose to integrate the proper interests of the user through an interactive process. To this end, we define the biased quality measure $\mathscr{M}_u(P)$ of an event $P = (S,I)$ with the preferences of a user u as:

$$\mathscr{M}_u(P) \quad = \quad \sum_{h\in\mathscr{H}(S,I)}\sum_{v\in S}\sum_{t\in I} \text{score}(h,v,t) \times Q_u(h,v)$$

where Q_u increases with the interest of u on h and v. Thus, if P contains some terms h or vertices v that are of interest according to user's feedback, the pattern P is overweighted. When no user feedback is

available for a pair (h,v), $Q_u(h,v)$ is equal to 1. We thoroughly discuss in section 6.3 how the function Q_u is defined.

If the user does not provide any feedback, measures $\mathscr{M}_u$ and $\mathscr{M}$ are equivalent (i.e., the term is $Q_u(h,v)$ is equal to 1 for any term h and vertex v). In such a case, the geolocated events are said *data-driven*. In the rest of the chapter, we consider user-driven geolocated events, if nothing is specified.

A user-driven geolocated event must be both spatially compact and have a significant value on this quality measure:

Definition 6.1 (User-driven geolocated event)**.** *Given a set of posts B, a set of timestamps T, a graph $G_V = (V,E)$, and a threshold $\delta > 0$, a pattern $P = (S,I)$, with $S \subseteq V$ and $I \sqsubseteq T$ (an interval of T), is a user-driven geolocated event iff (1) $G_V[S]$ is connected and (2) $\mathscr{M}_u(P) \geq \delta$.*

However, different geolocated events may overlap in terms, geographical area, and timestamps they share, depicting the same real-life event. This has two main disadvantages: (1) the size of the result set may be uselessly very large and redundant, and (2) the method performance may degrade due to the size of the output. Therefore, instead of finding the complete set of events, our goal is to return a concise summary that covers sufficiently all the events. To this end, we use the coverage measure cov. This function, defined for two sets or intervals A and B, measures how much the set B covers the set A: $\text{cov}(A,B) = \frac{|A \cap B|}{|A|}$. This coverage measure has been used in several problems in order to avoid the redundancy in the results [Wang *et al.*, 2013, Bendimerad *et al.*, 2016]. To measure how much an event $P_2 = (S_2,I_2)$ covers an event $P_1 = (S_1,I_1)$, we impose that P_2 sufficiently covers P_1 in all the pattern dimensions. The function cover is thus:

$$\begin{aligned} \text{cover}(P_1,P_2) \quad = \quad & \min\{\text{cov}(S_1,S_2), \text{cov}(I_1,I_2), \\ & \text{cov}(\mathscr{H}(S_1,I_1), \mathscr{H}(S_2,I_2))\} \end{aligned}$$

Definition 6.2 (Coverage guarantee event summary)**.** *Given a threshold $\texttt{minCov} \in [0,1]$, a coverage guarantee event summary $\mathscr{R}_1$ of the set of all subjective geolocated events $\mathscr{R}$ fulfills the following property:*

$$\forall P \in \mathscr{R},\ \exists P' \in \mathscr{R}_1,\ \textit{such that cover}(P,P') \geq \texttt{minCov}$$

In the remainder of this chapter, we propose two approaches to discover geolocated events: `SIGLER-Cov` that computes a coverage guarantee event summary $\mathscr{R}_1$, and `SIGLER-Samp` that samples the pattern space using a random exploration biased with the value of $\mathscr{M}_u$. These approaches are formally defined in Section 7.4. The following section details the computation of the user-driver weight used in the quality measure $\mathscr{M}_u(P)$.

6.3 Integration Of User Feedback Into Quality Measure

Let us now consider how to incorporate user interest into the geolocated event discovery process. Once a set of events has been identified, the user, denoted by u, appraises the detected events and indicates if she likes it. In doing so, she constructs a partial order on the patterns that is used to bias the discovery of the forthcoming geolocated events. User interest is expressed by $Q_u(h,v) = \frac{Q_u^H(G_H,h)+Q_u^V(G_V,v)}{2}$, the average of two similarity measures on terms and vertices:

1. $Q_u^H : (G_H,h)$ takes as parameters a weighted graph G_H on terms and a term h. It assigns a value in $[1,\texttt{maxPref}]$, with $\texttt{maxPref} > 1$ a user-defined parameter, based on (1) the neighborhood of h in G_H and (2) a partial order on terms of H derived from the ranking of events by the user. The closer h is to other liked terms in G_H, or the more recently it has been liked, the more $Q_u^H(G_H,h)$ is close to `maxPref`. Otherwise, it tends to 1.

2. $Q_u^V : (G_V, v)$ takes as parameters the graph G_V as defined before and a vertex $v \in V$. It takes its value in $[1, \texttt{maxPref}]$ and the closer v is to other liked vertices in G_V, or the more recently it has been liked, the more $Q_u^V(G_V, v)$ is close to `maxPref`.

`maxPref` > 1 represents the maximum value that Q_u can reach. The choice of `maxPref` depends on how much we want to bias the process with the user preferences. In the following, we use the value `maxPref` $= 3$, empirically chosen as the one that maximizes the number of liked events on NYC dataset (see the description of the experiment in section 6.5.3).

Both functions $Q_u^H(G_H, h)$ and $Q_u^V(G_V, v)$ are constructed above a weighted graph $G_X = (X, Y, W)$ with X a set of vertices, Y a set of edges and W the function that associates a weight to the edges ($W : Y \to \mathbb{N}$).

- Q_u^H is evaluated on G_H, the term co-occurrence graph, such that $X = H$, $Y = H \times H$ and $\forall h_i, h_j \in H$, $W(h_i, h_j)$ equals the number of posts in which h_i and h_j appear simultaneously:

$$W(h_i, h_j) = |\{b \in B \mid h_i \in b.terms \text{ and } h_j \in b.terms\}|$$

  Related terms may co-occur in several posts and thus will tend to be connected by by a shorter path whose sum of weights is high in G_H.
- Q_u^V is evaluated on the graph G_V, with $X = V$, $Y = E$ and $W : E \to 1$.

In a similar way to PageRank score [Page *et al.*, 1998], we want to valuate the vertices of the graph such that high scores are transferred to neighborhood vertices with high weights, and this effect should decrease when moving further and further away in the graph. However, unlike PageRank score, which is based on a random walk, we use a concentric model from the node to be valued: The node score is only influenced by the weights of its neighbors and not by their degrees:

$Q_X(G_X, x) = \alpha \sum_{(x,x') \in Y} \frac{W(x,x')}{\deg(x)} \times Q_X(G_X, x') + (1-\alpha)\frac{1}{|X|}$ with $\deg(x) = \sum_{(x,x') \in Y} W(x,x')$. The second term of this equation is a constant corresponding to the probability to directly reach a vertex, $\alpha \in]0,1[$ (whose default value is 0.7) is a balancing parameter and $\sum_{(x,x') \in Y} \frac{W(x,x')}{\deg(x)} \times Q(G_X, x')$ is simply the weighted average of qualities $Q_X(G_X, x')$ of the neighbors vertices, and $\deg(x)$ does not affect this sum. We propose to integrate the user preferences in the second term by replacing the constant value with the function $\mathscr{B}_u(x)$, a bias that amplifies the importance of vertices involved in recently liked events:

$$Q_{X_u}(G_X, x) = \alpha \sum_{(x,x') \in Y} \frac{W(x,x')}{\deg(x)} Q_{X_u}(G_X, x') + (1-\alpha)\mathscr{B}_u(x)$$

$\mathscr{B}_u(x)$ depends on the direct relation between x and the liked events. Let $\text{rank}(x)$ be the number of events that have been liked since the last event that (1) contained x and (2) was liked by the user. For instance, if x belongs to the last liked event, then $rank(x) = 0$. If the following liked event does not contain x, then $rank(x) = 1$, and so on. Particularly, if x does not belong to any liked event, then $rank(x) = +\infty$. We define $\mathscr{B}_u(x)$ as $\mathscr{B}_u(x) = 1 + \frac{\texttt{maxPref}-1}{1+log_2(1+\text{rank}(x))}$. We can observe that $\mathscr{B}_u(x) \in [1, \texttt{maxPref}]$ (for $rank(x) = +\infty$, $\mathscr{B}_u(x) = 1$) and increases if x is related to a recently liked event (if $rank(x) = 0$, $\mathscr{B}_u(x) = \texttt{maxPref}$). The $\log_2$ function is used to smooth the effect of past liked events.

$Q_{X_u}(G_X, x)$ can be rewritten as the matrix equation $A.Q_{X_u} = B$ with (1) $a_{ij} = 1$ if $i = j$, and $a_{ij} = -\alpha \frac{W(x_i,x_j)}{\deg(x_i)}$ otherwise; (2) $b_i = (1-\alpha)\mathscr{B}_u(x_i)$. This equation can be solved thanks to the Jaccobi method, as the convergence condition below is satisfied.

Proposition 6.1. *The matrix A is strictly diagonally dominant.*

Proof. As $\sum_{i \neq j} |a_{ij}| = \alpha \sum_{i \neq j} \frac{W(x_i,x_j)}{\deg(x_i)} = \alpha < 1$ and $|a_{ii}| = 1$, we have $|a_{ii}| > \sum_{i \neq j} |a_{ij}|$. □ □

6.4 Algorithms For Computing Geolocated Events

We first present SIGLER-Cov algorithm that computes the set of events with coverage guarantee. After that, we present SIGLER-Samp which is a randomized pattern sampling event detection approach. Finally, we discuss a post-processing that we apply to the final result in order to fix some potential issues.

6.4.1 Event Detection With Coverage Guarantee

We propose to extract user-driven geolocated events using a generate and test approach that first enumerates a time interval I, and then explores the connected subgraphs S associated to areas where posts of B were sent during I. The quality measure $\mathscr{M}_u(S,I)$ evaluates whether the terms of the posts sent from $G_V[S]$ during I are characteristic of this time frame. As the number of such generated events is very large, especially the number of possible subgraphs, we propose a set of pruning techniques that makes the extraction feasible on real-life generated sets of posts.

Algorithm 1 enumerates all the intervals $[t_i,t_j]$ included in $T = [\![t_1,t_m]\!]$ thanks to the two loops in lines 1 and 1. For each interval I, it explores the connected subgraphs of $G_V[C]$ – the graph induced by the vertices for which there exists at least a term that bursts during the interval I – calling the function SGEnumerate presented in Algorithm 2. This backtracking algorithm uses two sets of vertices: $S \subseteq V$, the current enumerated subgraph induced by S, and $C \subseteq V \setminus S$ the vertices that are still to be explored. Initially, $S = \emptyset$ and C contains all the vertices for which there exists a bursting term in I. Algorithm 2 enumerates vertices from $C \cap N(S)$, with $N(S)$ the neighbors of vertices of S: $N(S) = \{v \in V \setminus S \mid \exists u \in S : (u,v) \in E\}$. Once the candidate set C is empty, P is tested to only be retained if it satisfies definitions 6.1 and 6.2 (lines 2 to 2).

Algorithm 1: SIGLER-Cov(δ, minCov, $\mathscr{R}_1$)

Input: δ the quality threshold, minCov the coverage threshold
Output: $\mathscr{R}_1$ a set of coverage guarantee geolocated events
$\mathscr{R}_1 \leftarrow \emptyset$
for $i \leftarrow 1$ *to* m **do**
  $C \leftarrow V$
  for $j \leftarrow i$ *to* m **do**
    $I \leftarrow [t_i,t_j]$
    $C \leftarrow \{v \in C \mid \mathscr{H}(\{v\},I) \neq \emptyset\}$
    SGEnumerate(I, $\emptyset$, C,$\mathscr{R}_1$, δ, minCov)

We use three pruning mechanisms without which the algorithm does not scale. The first one – the anti-monotonicity of $\mathscr{H}(P)$ – is used line 4 of Algorithm 1 to prune vertices given a time interval. This property is defined up to the intuitive partial order $\subseteq$: $P_1 \subseteq P_2$ if and only if $S_1 \subseteq S_2$ and $I_1 \subseteq I_2$.

Proposition 6.2 (anti-monotony of $\mathscr{H}(P)$)**.** *Considering two patterns $P_1 = (S_1,I_1)$ and $P_2 = (S_2,I_2)$, if $P_1 \subseteq P_2$ then $\mathscr{H}(P_2) \subseteq \mathscr{H}(P_1)$.*

Proof. If $h \in \mathscr{H}(P_2)$, then $\forall v \in S_2, \forall t \in I_2$, score$(h,v,t) > 0$. As $S_1 \subseteq S_2$ and $I_1 \subseteq I_2$, thus, $\forall v \in S_1, \forall t \in I_1$, score$(h,v,t) > 0$, and thus $h \in \mathscr{H}(P_1)$. □ □

The second pruning mechanism is based on the computation of an upper-bound of $\mathscr{M}_u(P)$ (used line 2 in Algorithm 2).

Definition 6.3 (Upper-bound on $\mathscr{M}_u$)**.** *Let $S \subseteq V$, $I \sqsubseteq T$, and $J \subseteq H$. UB is defined as:*

$$UB(S,I,J) = \sum_{v \in S} \sum_{t \in I} \sum_{h \in J} \max\left(score(h,v,t) \times Q_u(h,v), 0\right)$$

Algorithm 2: SGEnumerate(I, S, C, $\mathscr{R}_1$, δ,minCov)

Input: I a time interval, S the current explored subgraph, C the candidate sets, δ the quality threshold, minCov the coverage threshold
Output: $\mathscr{R}_1$ the set of events

```
P ← (S,I)
if C ∩ N(S) ≠ ∅ then
    if UB(S∪C, I, ℋ(P)) ≥ δ then
        for P_r ∈ ℛ_1 do
            if LB(P, C, P_r) ≥ minCov then
                return
        Choose a vertex v ∈ C ∩ N(S)
        SGEnumerate(I, S∪{v}, C\{v}, δ, minCov)
        SGEnumerate(I, S, C\{v}, δ, minCov)
else
    if ℳ_u(P) ≥ δ then
        for P_r ∈ ℛ_1 do
            if cover(P, P_r) ≥ minCov then
                return
        ℛ_1 ← ℛ_1 ∪ {P}
```

We denote by $\Gamma(I,S,C)$ the set of all patterns that can be reached when expanding S by adding vertices from C: $\Gamma(I,S,C) = \{(S',I) \mid S' \subseteq S \cup C \text{ and } S \subseteq S'\}$, that is to say, the patterns that are generated from SGEnumerate(I, S,C,$\mathscr{R}_1$,δ,minCov). The following property states that $UB(S \cup C, I, \mathscr{H}(S,I))$ upper bounds $\mathscr{M}_u$ for all subsequent patterns:

Proposition 6.3. *Let $S \subseteq V$, $I \sqsubseteq T$, and $C \subseteq V \setminus S$. For all patterns $P' = (S',I) \in \Gamma(I,S,C)$, $UB(S \cup C, I, \mathscr{H}(S,I)) \geq \mathscr{M}_u(S',I)$.*

Proof. Since $S' \subseteq S \cup C$ and $\mathscr{H}(P') \subseteq \mathscr{H}(P)$, we have

$$
\begin{aligned}
&UB(S \cup C, I, \mathscr{H}(P)) = \\
&\sum_{v \in S'} \sum_{t \in I} \sum_{h \in \mathscr{H}(P')} \max\big(\text{score}(h,v,t) \times Q_u(h,v), 0\big) \\
&+ \sum_{v \in S \cup C} \sum_{t \in I} \sum_{h \in \mathscr{H}(P) \setminus \mathscr{H}(P')} \max\big(\text{score}(h,v,t) \times Q_u(h,v), 0\big) \\
&+ \sum_{v \in S \cup C \setminus S'} \sum_{t \in I} \sum_{h \in \mathscr{H}(P')} \max\big(\text{score}(h,v,t) \times Q_u(h,v), 0\big)
\end{aligned}
$$

$$
\begin{aligned}
&\geq \sum_{v \in S'} \sum_{t \in I} \sum_{h \in \mathscr{H}(P')} \max\big(\text{score}(h,v,t) \times Q_u(h,v), 0\big) \\
&\geq \sum_{v \in S'} \sum_{t \in I} \sum_{h \in \mathscr{H}(P')} \text{score}(h,v,t) \times Q_u(h,v) = \mathscr{M}_u(P') \qquad \square
\end{aligned}
$$

□

The last pruning technique is built on the coverage measure. As stated in Definition 6.2, there may exist several coverage guarantee summaries of geolocated events. Whereas it might be interesting to have a summary of smallest cardinality, the problem of finding the set of minimal size is NP hard. A practical approach consists in constructing the summary during the enumeration. We also use the coverage measure

to prune large parts of the subgraph search space thanks to a lower bound (lines 2 to 2 in Algorithm 2) defined below.

Proposition 6.4. *Given a geolocated event pattern $P_r = (S_r, I_r)$ and a pattern $P' = (S', I)$ in $\Gamma(I,S,C)$, we have $cov(S', S_r) \geq cov(S \cup (C \setminus S_r), S_r)$.*

Proof. As $S' = S \cup C'$ s.t $C' \subseteq C$, we have: $cov(S', S_r) = \frac{|S' \cap S_r|}{|S'|} = \frac{|S \cap S_r| + |C' \cap S_r|}{|S| + |C' \setminus S_r| + |C' \cap S_r|} \geq \frac{|S \cap S_r| + |C' \cap S_r|}{|S| + |C \setminus S_r| + |C' \cap S_r|}$. We define $g(x) = \frac{|S \cap S_r| + x}{|S| + |C \setminus S_r| + x}$, with $x = |C' \cap S_r|$. Knowing that the derivative $g'(x) = \frac{|S| + |C \setminus S_r| - |S \cap S_r|}{(|S| + |C \setminus S_r| + x)^2} \geq 0$, $g(x)$ takes its lower value when x is minimal, that is when $x = 0$. Thus $g(x) \geq \frac{|S \cap S_r|}{|S| + |C \setminus S_r|} = cov(S \cup (C \setminus S_r), S_r)$ and we conclude that $cov(S', S_r) \geq cov(S \cup (C \setminus S_r), S_r)$. □ □

Definition 6.4. *Let $\mathscr{H}(P) \cap \mathscr{H}(Pr) = \{h_1, \ldots, h_q\}$ – the set of terms of P covered by those of P_r – be ordered by $h_i \leq_{UB} h_j$ iff $UB(S \cup C, I, \{h_i\}) \geq UB(S \cup C, I, \{h_j\})$. We consider the minimal set $\{h_1, \ldots, h_{q^\star}\}$ of terms that can be added to $\mathscr{H}(P) \setminus \mathscr{H}(P_r)$ while still satisfying the upper-bound:*

$$q^\star = \underset{r \in \{1 \ldots q\}}{argmin} UB(S \cup C, I, \mathscr{H}(P) \setminus \mathscr{H}(P_r) \cup \{h_1, \cdots, h_r\}) \geq \delta$$

and define $H^\star$ as $\mathscr{H}(P) \setminus \mathscr{H}(P_r) \cup \{h_1, \cdots h_{q^\star}\}$. In other words, $\mathscr{H}^\star \subseteq \mathscr{H}(P)$ is the set of terms that overlaps the least with $\mathscr{H}(P_r)$ while verifying the condition $UB(S \cup C, I, \mathscr{H}^\star) \geq \delta$

Proposition 6.5. *For each pattern $P' = (S', I) \in \Gamma(I,S,C)$ such that $\mathscr{M}_u(P') \geq \delta$, we have $cov(\mathscr{H}(P'), \mathscr{H}(P_r)) \geq cov(\mathscr{H}^\star, \mathscr{H}(P_r))$.*

Proof. We know that $|\mathscr{H}(P') \cap \mathscr{H}(P_r)| \geq q^\star$, otherwise $UB(S', I, \mathscr{H}(P')) < \delta$ and $\mathscr{M}_u(P') < \delta$. We can rewrite $cov(\mathscr{H}(P'), \mathscr{H}(P_r)) = \frac{|\mathscr{H}(P') \cap \mathscr{H}(P_r)|}{|\mathscr{H}(P') \setminus \mathscr{H}(P_r)| + |\mathscr{H}(P') \cap \mathscr{H}(P_r)|} = \frac{q^\star + x}{|\mathscr{H}(P') \setminus \mathscr{H}(P_r)| + q^\star + x}$ with $x \geq 0$. Let's denote $g(x) = \frac{q^\star + x}{|\mathscr{H}(P) \setminus \mathscr{H}(P_r)| + q^\star + x}$. We have $cov(\mathscr{H}(P'), \mathscr{H}(P_r)) \geq g(x)$ as $\mathscr{H}(P') \subseteq \mathscr{H}(P)$. Knowing that the derivative $g'(x) = \frac{|\mathscr{H}(P) \setminus \mathscr{H}(P_r)|}{(|\mathscr{H}(P) \setminus \mathscr{H}(P_r)| + q^\star + x)^2} \geq 0$, then $g(x)$ takes its lower value when $x = 0$. Thus, $g(x) \geq \frac{q^\star}{|\mathscr{H}(P) \setminus \mathscr{H}(P_r)| + q^\star} = cov(\mathscr{H}^\star, \mathscr{H}(P_r))$ and we conclude the proof. □ □

Definition 6.5. *We define the function LB as*

$$\begin{aligned} LB(P,C,P_r) \quad = \quad & \min\{cov(S \cup (C \setminus S_r), S_r), cov(I, I_r), \\ & cov(\mathscr{H}^\star, \mathscr{H}(P_r))\} \end{aligned}$$

Proposition 6.6. *For each pattern $P' = (S', I) \in \Gamma(I,S,C)$ such that $\mathscr{M}_u(P') \geq \delta$, we have $cover(P', P_r) \geq LB(P,C,P_r)$.*

Proof. It is sufficient to prove that : $cov(S', S_r) \geq cov(S \cup (C \setminus S_r), S_r)$ and $cov(\mathscr{H}(P'), \mathscr{H}(P_r)) \geq cov(\mathscr{H}^\star, \mathscr{H}(P_r))$ and $cov(I, I_r) \geq cov(I, I_r)$ The first two equations are respectively guaranteed with Propositions 6.4 and 6.5, and the third one is trivial. □ □

The exploration of the search space $\Gamma(I,S,C)$ is stopped if there exists $P_r \in \mathscr{R}_1$ such that $LB(P,C,P_r) \geq minCov$, because all the subsequent patterns are covered by it.

6.4.2 Pattern Sampling Based Event Detection

In this section, we explore another approach to discover geolocated events: Pattern sampling [Bhuiyan and Al Hasan, 2016]. Given a time budget, the proposed algorithm `SIGLER-Samp` mines events using a random search space exploration biased with the value of $\mathscr{M}_u$. Such an approach enables instant mining which is required in interactive pattern mining.

The pattern sampling process we consider is based on a random walk on a graph whose vertices are patterns $P = (S,I)$ and edges (transitions) are chosen following a probability measure that overweights high quality patterns. The random walk starts from a singleton pattern $P = (S,I)$ where $S = \{v\}$ and $I = [t,t]$. Next, P is expanded by adding randomly drawn vertices from $N(S)$, the neighborhood of S, or by adding a timestamp to I. The random walk is basically composed of two main steps described in Algorithm 4:

1. $\mathscr{M}_u(P)$ is computed for each pattern $P = (S,I)$, where $S = \{v\}$, $v \in V$ and $I = [t,t]$, $t \in T$. The probability of drawing a singleton pattern P is defined as $\mathscr{P}(P) = \frac{\mathscr{M}_u(P)}{\sum_{v' \in V, t' \in T} \mathscr{M}_u(\{v'\},[t',t'])}$
2. From a current pattern $P = (S,I)$ with $I = [t_i,t_j]$, the next step consists to draw a pattern P' from the set:

$$\begin{aligned} \text{Next}(P) &= \{(S,I)\} \cup \{(S,[t_{i-1},t_j])\} \cup \{(S,[t_i,t_{j+1}])\} \\ &\cup \{(S',I) \mid S' = S \cup \{v\},\ v \in N(S)\} \end{aligned}$$

   This is done based on $\mathscr{P}(P'|P)$, the probability to reach the pattern $P' \in \text{Next}(P)$ from P: $\mathscr{P}(P'|P) = \frac{\mathscr{M}_u(P')}{\sum_{P_2 \in \text{Next}_{(P)}} \mathscr{M}_u(P_2)}$.
   This distribution of probabilities rewards transitions toward patterns P' with large $\mathscr{M}_u(P')$ values. After drawing P', if $P' \neq P$, we continue the expansion by repeating Step 2 using the new pattern P'. If $P' = P$, the pattern P is returned to the user and the sampling is repeated from Step 1 until the whole consumption of the time budget.

The two main steps are repeated (lines 3 to 3) until $P' = P$. At each iteration, the pattern $P = (S,I)$ is extended by adding a vertex to S or a timestamp to I. Since S and I are respectively bounded by V and T, this loop necessarily stops after at most $|V| + |I|$ iterations. All patterns with nonzero $\mathscr{M}_u$ value have a non zero probability to be generated.

Proposition 6.7. *For each pattern $P = (S,I)$, if $\mathscr{M}_u(P) > 0$ then $\mathscr{P}(P \in \mathscr{R}_2) > 0$*

Proof. Let us prove it by induction on $n = |S| + |I|$.

- For $n = 2$, P is such that $|S| = 1$ and $|I| = 1$ (in other cases, $\mathscr{M}_u(P) = 0$). P can be drawn in the first step, and if it is chosen from $\text{Next}(P)$ in step 2, it is added to $\mathscr{R}_2$. Thus, $\mathscr{P}(P \in \mathscr{R}_2) \geq \frac{\mathscr{M}_u(P)}{\sum_{v' \in V, t' \in T} \mathscr{M}_u(\{v'\},[t',t'])} \times \frac{\mathscr{M}_u(P)}{\sum_{P_2 \in \text{Next}_{(P)}} \mathscr{M}_u(P_2)} > 0$.
- Let us suppose that the proposition is true for n. Let $P = (S,I)$ be a pattern such that $|S| + |I| = n+1$ and $\mathscr{M}_u(P) > 0$. Let $P' = (S',I') \subseteq P$ be a pattern containing one less vertex or one less timestamp than P. This means that $|S'| + |I'| = n$, by the recursion hypothesis we have $\mathscr{P}(P' \in \mathscr{R}_2) > 0$. Thus, the probability $\mathscr{P}(P')$ to reach P' is not null. Since P can be reached from P' during the random walk, then: $\mathscr{P}(P \in \mathscr{R}_2) \geq \mathscr{P}(P') \times \frac{\mathscr{M}_u(P)}{\sum_{P_2 \in \text{Next}_{(P')}} \mathscr{M}_u(P_2)} \times \frac{\mathscr{M}_u(P)}{\sum_{P_2 \in \text{Next}_{(P)}} \mathscr{M}_u(P_2)}$
  $\mathscr{P}(P \in \mathscr{R}_2) > 0$ □

□

Algorithm 3: `SIGLER-Samp`

Input: `time_budget`
Output: $\mathscr{R}_2$ a set of sampled patterns
for $v \in V, t \in T$ **do**
  compute $\mathscr{M}_u(\{v\},[t,t])$
while *current_time* $<$ `time_budget` **do**
  // Step 1: draw a singleton pattern P
  draw $P = (\{v\},[t,t]) \sim \frac{\mathscr{M}_u(P)}{\sum_{v' \in V, t' \in T} \mathscr{M}_u(\{v'\},[t',t'])}$
  // Step 2: expansion of P
  $P' \leftarrow P$
  repeat
    $P \leftarrow P'$
    // Compute the set Next(P) for $P = (S,[t_i,t_j])$
    Next$(P) \leftarrow \{P\}$
    for $v \in N(S)$ **do**
      Next$(P) \leftarrow$ Next$(P) \cup \{(S \cup \{v\},[t_i,t_j])\}$
    if $i > 1$ **then**
      Next$(P) \leftarrow$ Next$(P) \cup \{(S,[t_{i-1},t_j])\}$
    if $j < m$ **then**
      Next$(P) \leftarrow$ Next$(P) \cup \{(S,[t_i,t_{j+1}])\}$
    for $P' \in Next(P)$ **do**
      compute $\mathscr{M}_u(P')$
    draw $P' \sim \frac{\mathscr{M}_u(P')}{\sum_{P_2 \in \text{Next}(P)} \mathscr{M}_u(P_2)}$
  until $P' = P$;
  $\mathscr{R}_2 \leftarrow \mathscr{R}_2 \cup P$

6.4.3 Discussion

We discuss here some potential issues that may appear and the related post-processing to fix them. By applying one of the aforementioned algorithms, we find the set of patterns $\mathscr{R}_*$. In some particular cases, two real life events happen at the same time and the same location. However, these two events will be merged in the same pattern $P \in \mathscr{R}_*$. It is important to separate them before displaying the result to the end user. Therefore, for each pattern $P = (S,I,\mathscr{H}(S,I)) \in \mathscr{R}_*$, we apply a community detection algorithm on the terms $\mathscr{H}(S,I)$ in order to partition them into groups $W_1,...,W_k$ where (1) $\forall i \in [\![1,k]\!]$: $W_i \subseteq \mathscr{H}(S,I)$ (2) $\cup_i W_i = \mathscr{H}(S,I)$ (3) each W_i corresponds to a single real life event. Since we use Louvain community detection algorithm [Blondel *et al.*, 2008], we express its result by the function $Louvain_1(P) = \{W_1,...,W_k\}$. The similarity measure sim_1 used in this clustering is defined for two terms $h_1,h_2 \in \mathscr{H}(S,I)$ w.r.t the pattern P:

$$sim_1(h_1,h_2,P) = \begin{cases} 1, \text{if } |\{b \in B \mid (\{h_1,h_2\} \subseteq b.terms) \text{ and} \\ \quad (\text{area}(b.loc) \in S) \text{ and } (b.time \in I)\}| \geq 1 \\ 0, \text{otherwise} \end{cases}$$

In other words, $sim_1(h_1,h_2,P) = 1$ if h_1 and h_2 co-occur at least once in the space S and the time interval I. Thus, each cluster $W_i \in Louvain_1(P)$ would be a set of terms that co-occur in this space-time. Each cluster W_i gives a pattern (S,I,W_i) with the same space-time than the current P. After applying this clustering to each $P \in \mathscr{R}_*$, we have the post-processed result $R' = \cup_{(S,I,\mathscr{H}(S,I)) \in \mathscr{R}_*} \{(S,I,W) \mid W \in Louvain_1(S,I,\mathscr{H}(S,I))\}$

In order to deal with the redundancy issue, we have defined SIGLER-Cov that computes a summary $\mathscr{R}_1$. However, it is not necessarily the optimal summary, that is to say the summary of minimal size whose events partially cover each geolocated events that does not belong to the summary. Indeed, the computation of an optimal summary is NP hard. Thus, some redundancy can still remain in the result, we post-process the set R' to fix this issue. We apply Louvain algorithm on patterns $P \in R'$ to merge the ones with similar location, time interval, and terms. We define a similarity measure sim_2 for two patterns $P = (S,I,W)$ and $P' = (S',I',W')$: $sim_2(P,P') = \frac{|S \cap S'|}{|S \cup S'|} \times \frac{|I \cap I'|}{|I \cup I'|} \times \frac{|W \cap W'|}{|W \cup W'|}$. The result will be the communities: $\mathscr{C}_1, ..., \mathscr{C}_l$ where each community $\mathscr{C}_i \subseteq R'$ is a set of similar patterns. From each community $\mathscr{C}_i$ we reconstitute a pattern $P_{\mathscr{C}_i} = \cup_{(S,I,W) \in \mathscr{C}_i}(S,I,W)$. The final result set of pattern is: $R'' = \{P_{\mathscr{C}_1}, ..., P_{\mathscr{C}_l}\}$.

6.5 Experiments

In this section, we report our experimental results. We start by describing the real-world datasets we use, as well as the questions we aim to answer. Then, we provide a thorough comparison with the state-of-the-art algorithms and we report a performance study. Eventually, we evaluate the ability of our approach to take user interests into account through different testbeds[21].

MINTAG is implemented in C++ and the experiments were executed on a machine equipped with i7 CPU @ 2.5GHz, and 16GB main memory, running macOS Sierra version 10.12.2. For reproducibility purposes, the source code and the data are available[22].

6.5.1 Experimental Setting

Experiments are performed on 3 real-world datasets that contain the tweets obtained by querying three different cities on Twitter: New York (NYC), Los Angeles (LA) and London. For each city, we collected geolocated public tweets and removed those produced by bots (i.e, accounts that produce more than 100 tweets in a period of 10 days). We only retained tweets containing hashtags or user mentions. The main characteristics of these datasets are given in Table 6.1.

dataset	starting date	ending date	# tweets	# distinct terms
New York	2016/10/08	2017/01/07	652,244	332,618
Los Angeles	2017/05/17	2017/07/27	353,541	224,769
London	2017/05/17	J 2017/07/27	270,648	177,166

Table 6.1: Description of the real-world datasets.

This empirical study aims to answer the following questions: Are SIGLER-Cov and SIGLER-Samp more effective and efficient than their competitors? Do they scale well according to the size of the dataset and the different parameters? Does SIGLER-Samp capture all the events? Is the approach able to make use of user feedback to discover user-relevant events?

In the first bunch of experiments, we compare our approach with two local event detection approaches: (1) Multiscale Event Detection algorithm (MED) [Dong *et al.*, 2015], a state-of-the-art algorithm which aims to identify geolocated events based on a wavelet analysis of time series of terms, (2) GeoBurst [Zhang *et al.*, 2016], an online local event detection method, that first detects geo-topic clusters using a random walk on a keyword co-occurrence graph, and then ranks all the clusters with a weighted combination of spatial and temporal burstiness. We also considered INSIGHT [Ifrim *et al.*, 2014] and

[21] We report experiments performed on a crowd-sourcing platform with real-users in this chapter. Additional experiments with virtual users are reported in supplementary material.

[22] goo.gl/WxQR3v

EDCoW [Weng and Lee, 2011], two non location-aware event detection methods. INSIGHT is one of the best methods to detect events in tweets, as it won a recent challenge [Papadopoulos *et al.*, 2014].

These experiments show that (1) non location-aware approaches are not appropriate to detect geolocated events, and (2) MED and GeoBurst algorithms are less robust to noise than our approaches, and encounter scalability issue. Finally, we demonstrate the ability of our methods to extract interesting user-driven events via experiments performed on a crowdsourcing platform.

6.5.2 Effectiveness

We first study the ability of our approach to detect user-driven geolocated events. Using the method described in [Dong *et al.*, 2015], we generate artificial datasets for which the ground-truth geolocated events are known. Twenty events, denoted hereafter R_0, are artificially created. Each of them lasts between 2 and 4 timestamps, involves from 2 to 4 vertices and is defined by 4 unique terms. The datasets contain 1024 vertices, 32 timestamps and 1080 unique terms. Posts are artificially produced and sent at different timestamps and spatial locations. Each post contains between 5 to 10 terms. These posts are either related to embedded events, or are randomly drawn: (1) Event-related posts are uniformly distributed over the vertices and times stamps of its associated event and contains 2 of the 4 event-related terms as well as between 3 to 8 other terms; (2) Non-event related posts are uniformly distributed over the other timestamps and vertices. The terms they contain are drawn using a Zipf law probability distribution [Pérez-Melián *et al.*, 2017] among the 1000 non-event related terms. 10 to 30 event-related posts are randomly drawn and the number of non-event related posts is controlled by the *noise_rate* parameter.

Let $R = \{e_1, \ldots, e_k\}$ be the set of discovered events. The quality of R is assessed based on the following adapted $Fscore$ measure:

$$Fscore(R, R_0) \quad = \quad 2 \times \frac{Precision(R, R_0) \times Recall(R, R_0)}{Precision(R, R_0) + Recall(R, R_0)}$$

with $Precision(R, R_0) = \frac{\sum_{e \in R} max_{e' \in R_0} \text{COV}(e, e')}{|R|}$ and $Recall(R, R_0) = \frac{\sum_{e \in R_0} max_{e' \in R} \text{COV}(e, e')}{|R_0|}$.

Fig. 6.2 presents the $Fscore$ values achieved by the different approaches when noise rate is varying. From this figure we can draw the following conclusions:

1. *Non location-aware event detection approaches cannot detect geolocated events*, as INSIGHT and EDCoW $Fscore$ is always lower than 0.2.
2. MINTAG, MED and GeoBurst perform well with low noise rate. However, *both SIGLER-Cov and SIGLER-Samp are more robust to noise than MED and GeoBurst* . The $Fscore$ of our approaches remains high even when the noise rate increases, whereas MED and GeoBurst $Fscore$ decreases almost linearly. More precisely, MED and GeoBurst keep a good recall but the precision decreases when the noise rate increases.
3. Furthermore, the $Fscore$ of SIGLER-Samp is obviously bounded by the $Fscore$ obtained by SIGLER-Cov which adopts a more exhaustive exploration of the search space. Nevertheless, the bigger the time budget, the better the $Fscore$.

6.5.3 Efficiency

To evaluate the scalability of the algorithms, we consider New York tweets, our largest dataset. Fig. 6.3 reports the runtime and number of events obtained by MINTAG[23], MED, and GeoBurst when dataset

[23]The runtime for MINTAG corresponds to the complete process including the post-processing step described in Section 6.4.3. This explains the slight variation of the runtime of SIGLER-Samp for different configurations.

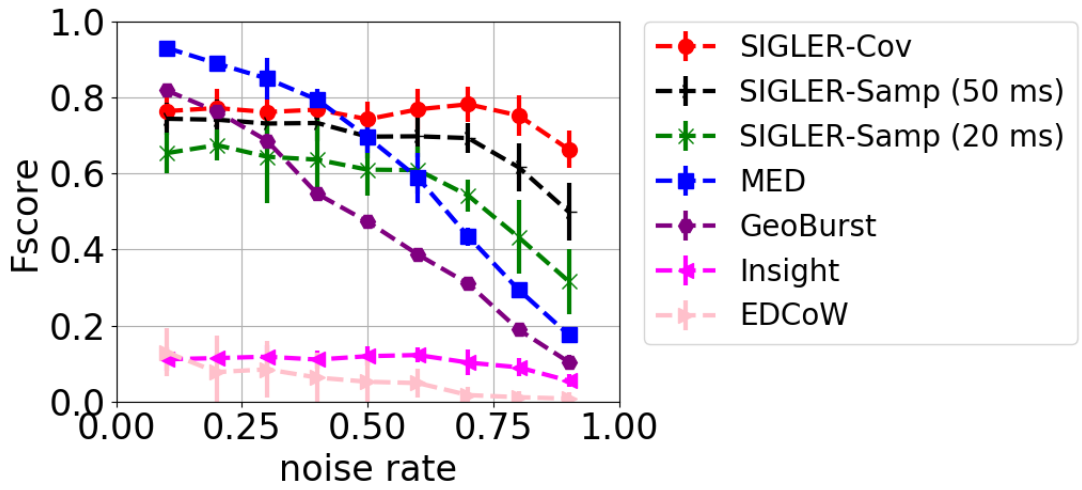


Figure 6.2: Average and error bars of $Fscore$ values obtained over 10 generated datasets for a given noise rate value ($\delta = 5$ and $\texttt{minCov} = 0.8$). Non location-aware event detection approaches are not able to detect geolocated events. MED fails in presence of noise. Runtime of `SIGLER-Cov` increases from 40ms to 2s when the noise rate increases.

parameters are varying. However, MED is only reported for at most 10,000 tweets because of its scalability issues (in the figures at left).

MED, GeoBurst, and `SIGLER-Cov` discover a comparable number of events but MED performances raise scalability issues. Indeed, `SIGLER-Cov` outperforms MED with several orders of magnitude for all the configurations, especially when the number of tweets increases. Even if MED uses some indexing techniques, its computational complexity is quadratic in the number of tweets, and MED fails to handle large datasets. Although MED is able to process one day of tweets in our experiments, it is without considering the fact that the Twitter API gives access to less than 1% of the posted tweets. MED scale limitation is therefore a real issue on these data.

In Fig. 6.3 - right, we report the runtime and the number of discovered events with higher numbers of tweets (we consider the whole dataset). We observe that the execution time of `SIGLER-Cov` increases with the number of tweets, but there is no order of magnitude change (non-logarithmic scale). Although GeoBurst runtime also increases linearly, it is considerably higher than the one of `SIGLER-Cov`.

We also study the impact of number of vertices and time granularity on our methods. We show their behavior in Fig. 6.4 when these two dimensions are varying. The execution time of `SIGLER-Cov` increases when the number of vertices and time granularity increase. The execution time of `SIGLER-Samp` is controlled by a parameter and we can observe that its number of results tends to the one of `SIGLER-Cov` when the time budget increases.

To go further on evaluating the discovered events, Fig. 6.5 investigates the ability of `SIGLER-Samp` to capture similar events as `SIGLER-Cov`, that is to say it verifies that the computed event sample well covers all the events obtained with the exhaustive approach. To this end, we compare the events provided by `SIGLER-Samp` (denoted R_1) with the events discovered by `SIGLER-Cov` (denoted R_2) as follows. Using the Jaccard similarity measure of two patterns $P1 = (S_1, I_1)$ and $P_2 = (S_2, I_2)$, $J(P_1, P_2) = \frac{1}{3} \times \left(\frac{|S_1 \cap S_2|}{|S_1 \cup S_2|} + \frac{|I_1 \cap I_2|}{|I_1 \cup I_2|} + \frac{|H_{P1} \cap H_{P2}|}{|H_{P1} \cup H_{P2}|} \right)$, the similarity of R_1 and R_2 is defined by $Sim(R_1, R_2) = \frac{\sum_{P_1 \in R_1} \max_{P_2 \in R_2} J(P_1, P_2)}{|R_1| + |R_2|}$ + $\frac{\sum_{P_2 \in R_2} \max_{P_1 \in R_1} J(P_1, P_2)}{|R_1| + |R_2|}$. We executed `SIGLER-Samp` with different time budgets and post-processed the result R_2 by removing redundant patterns (using $\texttt{minCov} = 0.8$) and low quality ones (using $\delta = 40$).

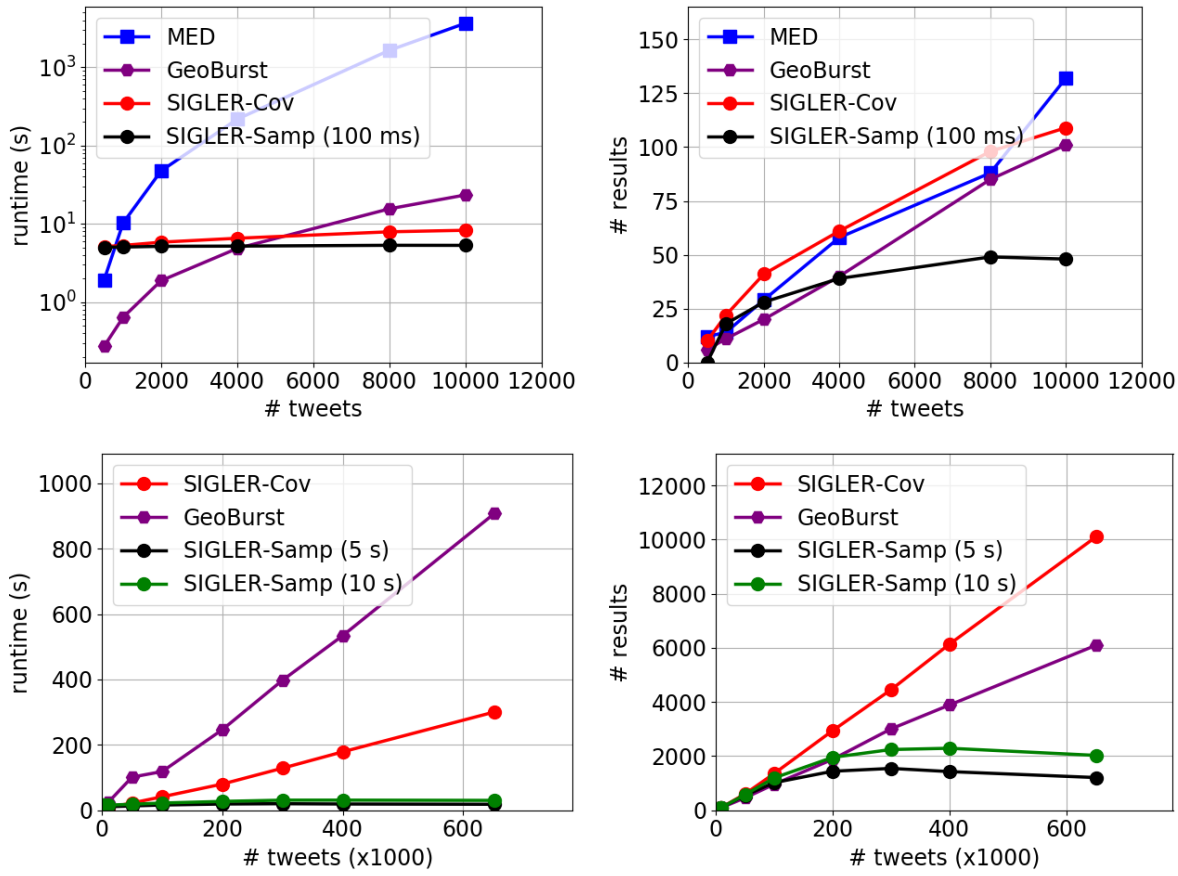


Figure 6.3: Runtime and number of events of `SIGLER-Cov`, `SIGLER-Samp`, MED and GeoBurst when varying the number of tweets (default values of SIGLERs: 2000 vertices, $\Delta t = 3h$, $\delta = 10$ and `minCov` $= 0.8$).

Fig. 6.5 reports the *Sim* values with respect to `SIGLER-Samp` time budget for 300K tweets (left) and the whole dataset (right). The runtimes of `SIGLER-Cov` for these two cases are respectively 95s and 173s. With a time budget fixed to 11% of the execution time of `SIGLER-Cov` (around 9s and 19s), `SIGLER-Samp` retrieves most of the high-quality patterns ($sim > 0.9$).

Finally, Fig. 6.6 evaluates the impact of the two main parameters, `minCov` and δ, on the results. `minCov` is a very intuitive parameter that eliminates a pattern if it is covered at least by `minCov`% of a pattern belonging to the solution. When `minCov`=1, only non maximal patterns are removed, and the more `minCov` decreases, the more disjoint the patterns. From Fig. 6.6 we can observe that this parameter has also a major impact on the execution time. Indeed, this parameter is involved in the computation of the upper-bounds and when large, it drastically reduces the execution time. In our experiments, we set this parameter to 0.8 to remove highly redundant events while allowing some intersections.

The quality of an event is evaluated by the measure $\mathscr{M}$, also used to rank the patterns when presented to the user (see next subsections). The function of the parameter δ is to cut the tail of the pattern distribution in order to only keep those of high quality. So the larger δ, the smaller the number of patterns and the faster the execution.

For the following experiments, we fixed the value of δ according to the number of events that we wanted to present to the user. Thus, we fixed δ value so as to have around 800 events for NYC, which

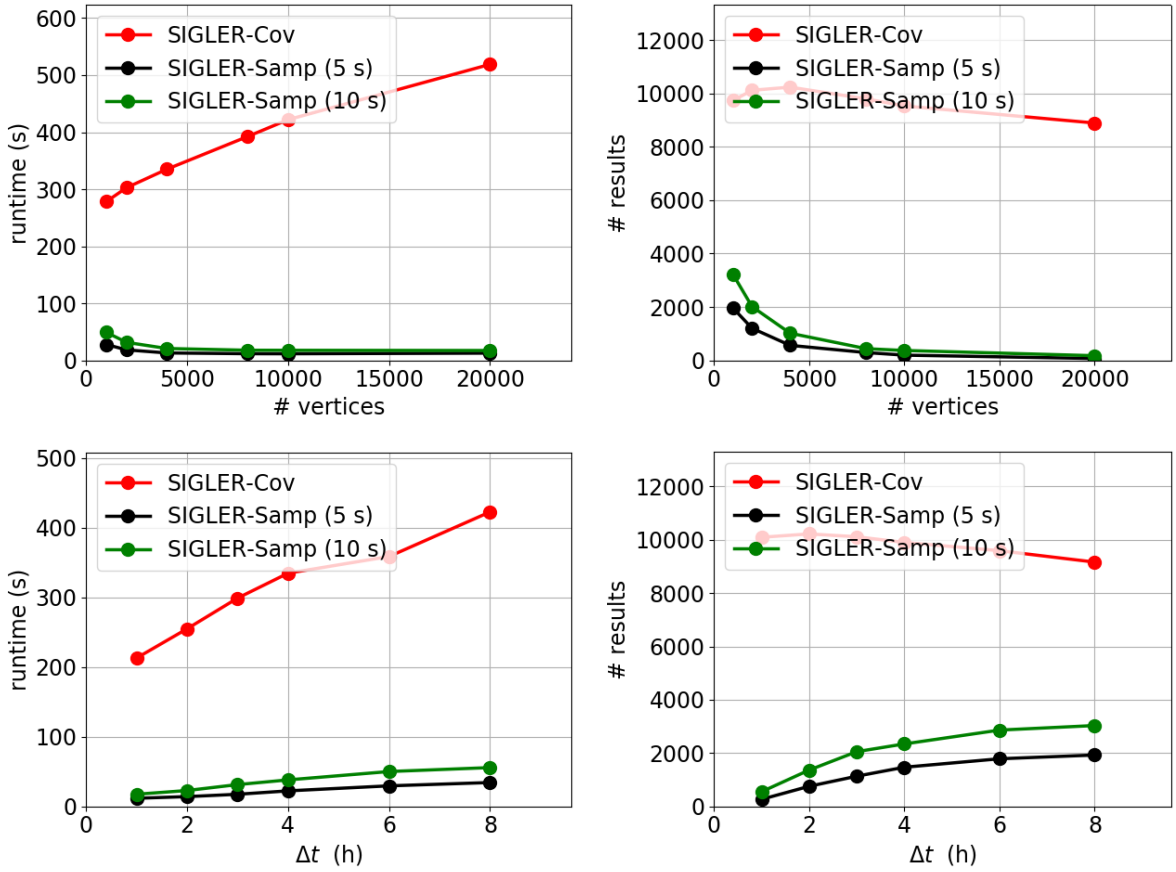


Figure 6.4: Number of events by `SIGLER-Cov` and `SIGLER-Samp` and runtime of `SIGLER-Cov` according to the number of vertices, and the time granularity when considering the whole dataset (default values: 652k tweets, 2000 vertices, $\Delta t = 3h$, $\delta = 10$ and $\texttt{minCov} = 0.8$).

contains 3 months of tweets, and around 400 events for LA and London that contain 70 days of tweets. This led us to set $\delta = 40$ for NYC, and $\delta = 15$ for London and LA dataset.

6.5.4 User-driven discovery of geo-located events

To evaluate the ability of `MINTAG` to take benefit of user feedback, we performed interactive event detection process with real users on real datasets. We used a crowdsourcing platform, Figure Eight[24], to hire people living in the country where the data is located. Indeed, our user feedback requires some expertise about the city and its events. To this end, we developed a graphical application[25] and deployed it on Figure Eight. For each user, the process consists in several iterations. At each of them, a batch containing the tweets emitted for 6 consecutive days is given as input to the algorithms (with a recovery of 3 days between 2 batches). Geolocated events are computed using either $\mathscr{M}$ (data-driven detection), or $\mathscr{M}_u$ (user-driven detection). Then, the user is asked to like the events that correspond to her preferences. Finally, the batch index is incremented and the process iterates. Since evaluating with the overall datasets can be very long and exhausting for real users, we limited the number of batches to 10, that is approximately 33

[24] `https://www.figure-eight.com`
[25] `134.214.104.134:6001` and `134.214.104.134:6002`.

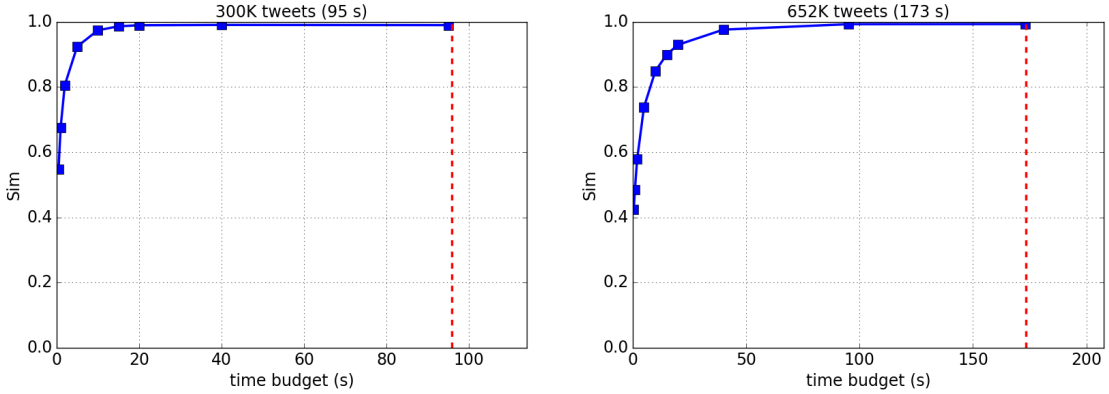


Figure 6.5: Similarity of the results of `SIGLER-Cov` with the ones of `SIGLER-Samp` with respect to `SIGLER-Samp` time budget.

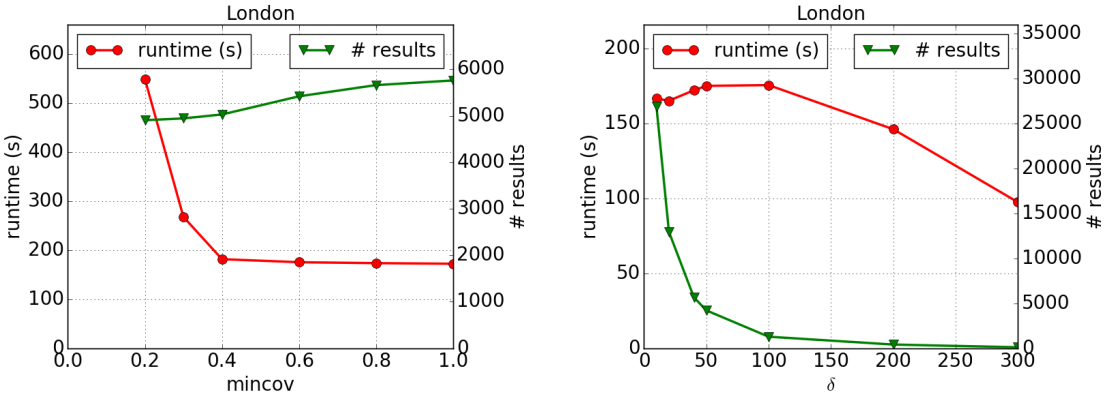


Figure 6.6: Runtime and number of discovered events by `SIGLER-Cov` according to `minCov` and δ (default value `minCov`=0.8, $\delta = 40$).

days of data.

We used two different settings to compare the data-driven and the user-driven approaches. In the *paired sample*, each of the 40 participants evaluated both approaches in a blind way, i.e. the two lists of events were randomly displayed to the user[26]. In the *unpaired sample*, 60 participants evaluated either the data or the user-driven approach while not being aware of the type of method used[27].

Fig. 6.7 reports the results of this crowdsourcing-based evaluation. For the paired sample, the number of likes is greater or equal in the user-driven setting than in the data-driven one, while results are less obvious for the unpaired sample. The purpose of paired samples is to get better statistics by controlling for the effects of other "unwanted" variables. And so, as our sample sizes are quite small, results obtained on paired samples are probably the most reliable. In addition, the test of Wilcoxon [Demsar, 2006] is applicable on the *paired sample*, while it is not on the *unpaired* ones. However, even for paired samples, the Wilcoxon test does not allow to reject the null hypothesis "the number of likes are similar", and the

[26]The paired sampled evaluation framework is available on: `134.214.104.134:6002`
[27]The unpaired evaluation framework is available: `134.214.104.134:6001`.

difference is not considered as significant. Considering the average ranks of liked events, we can observe that they are always better in the user-driven configuration than in the data-driven one. The difference in the values is considered as significant by both Wilcoxon and Nemenyi tests [Demsar, 2006]. The Nemenyi test value is shown in Fig. 6.7: when the rank difference on the graduated line is greater than the CD value, the rank difference is considered not due to chance.

Thus, this experience with real users leads to nuance the claim that the user-driven approach makes it possible to identify more interesting events than the data-driven one. However, it confirms that the identified events are of much better quality for the user-driven setting than for the data-driven one. Similar experiments with simulated users are reported in supplemental material.

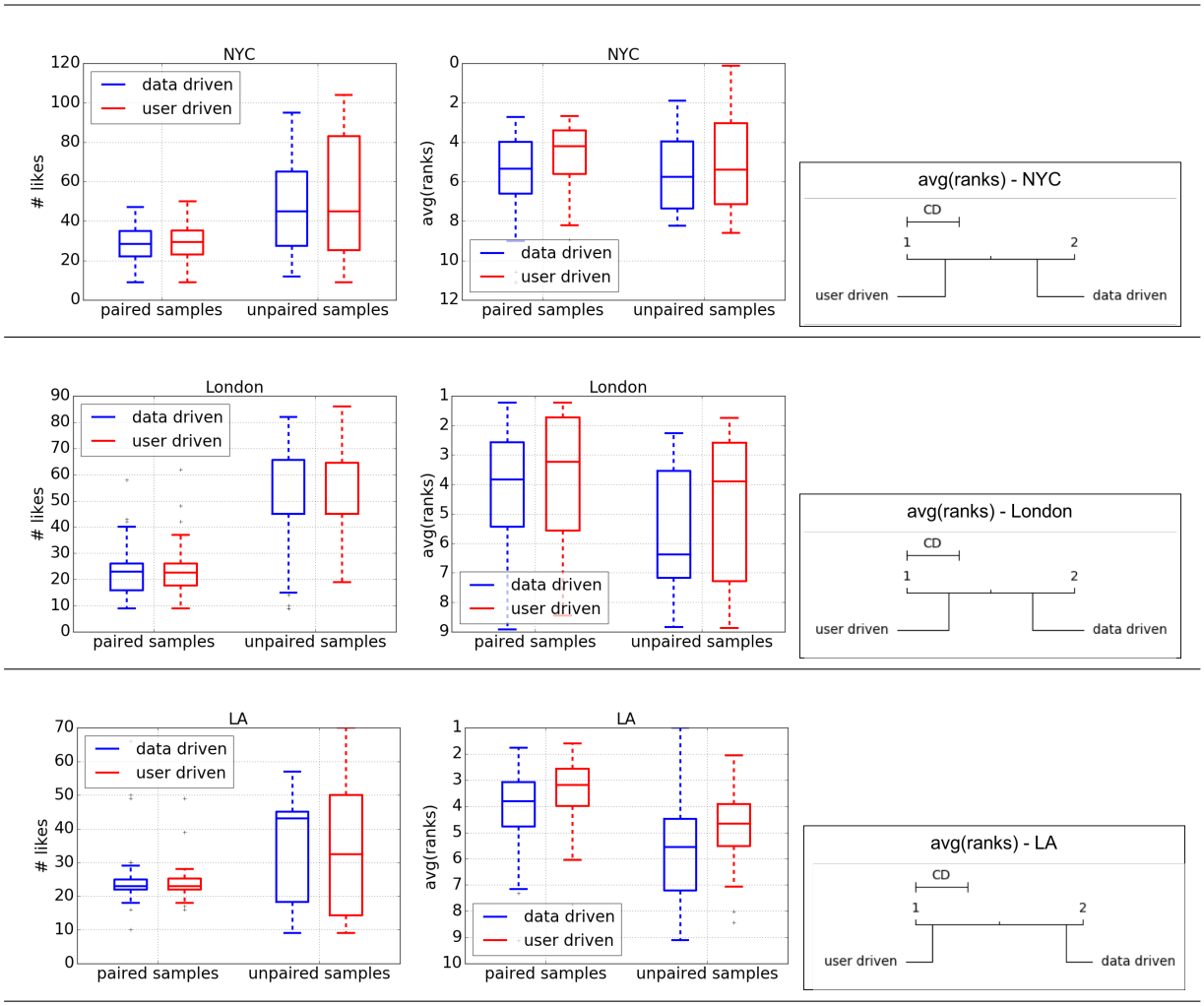


Figure 6.7: Ability to take user feedback into account with real users: number of likes (left), average ranks of liked events (center), Nemenyi tests on average ranks (right) for paired samples.

6.5.5 Illustrative results

Finally, we show some examples of events detected by our approach. Figure 6.8 reports the top 3 events detected in New York, London, and Los Angeles datasets. Each event is described by the locations, time interval and top 8 most frequent terms of its related tweets.

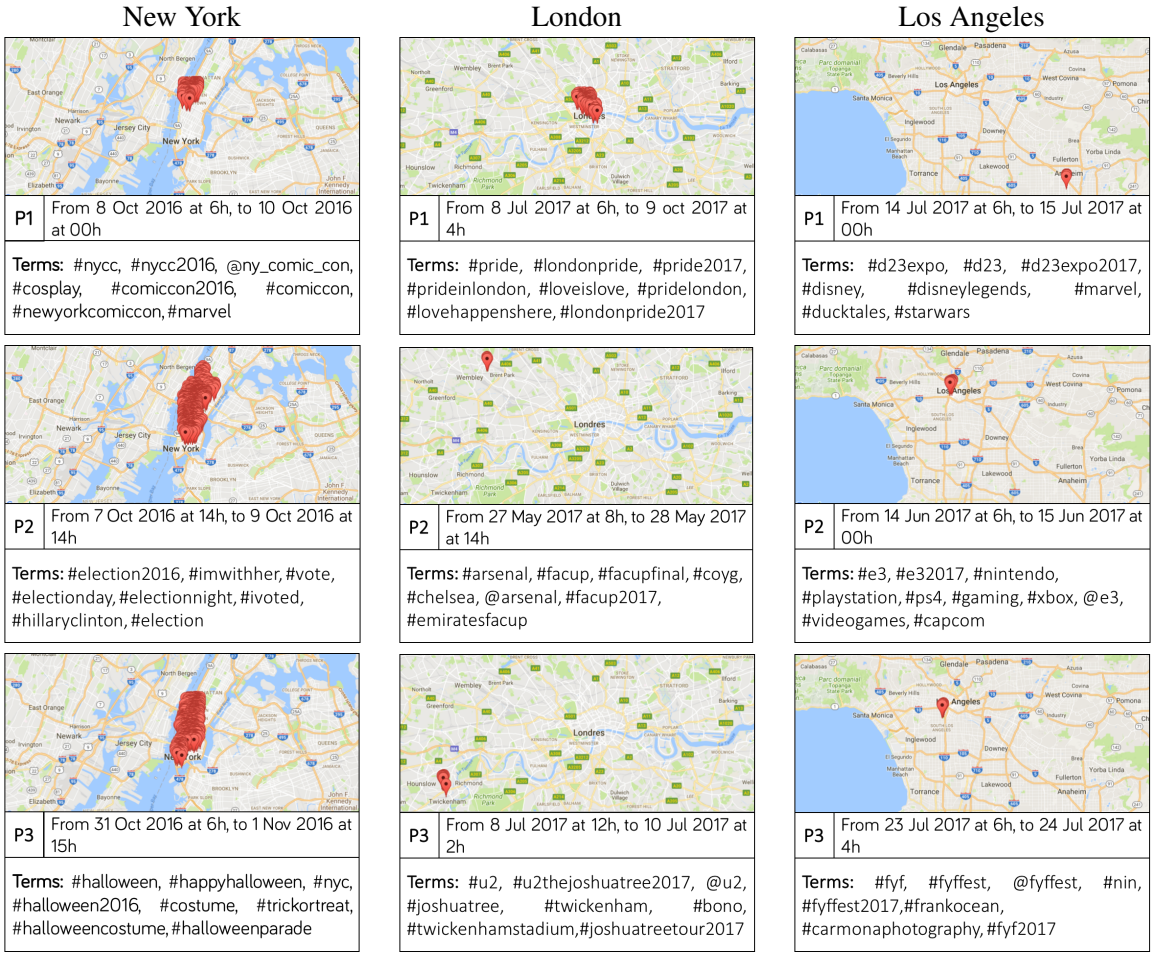


Figure 6.8: Top 3 events detected in New York (left), London (center), and Los Angeles (right).

The first event in New York is the Comic Con[28] , which is an annual convention mainly dedicated to comics. It was organized in Jacob K. Javits Convention Center, the location associated to the related tweets. The two other events correspond to USA presidential elections, and the celebration of Halloween. In London, the first event is a Pride Parade organized in July 8th, 2017. The remaining are two geolocated events corresponding to a soccer event and a concert of U2. In Los Angeles, the top 3 events are respectively: the Official Disney Fan Club, E3[29] (a video game related event), and the FYF Fest[30] (a music festival).

6.6 Conclusion

In this chapter, we introduced the novel problem of user-driven geolocated event discovery in social media. We handled the discovery of data-driven and user-driven event detection in a unified view. We defined two different algorithms to efficiently and effectively discover events based on geolocation and user feedback.

[28] goo.gl/BR7kgp
[29] goo.gl/c8FK5y
[30] fyffest.com/

Experiments demonstrate that, in the data-driven setting, our approach outperforms state of the art method with several orders of magnitude. Furthermore, our approach is more robust to noise. We also provide evidence that non location-aware event detection approaches fail to discover geolocated events. Thus, user feedback cannot be considered by post-processing the events obtained by existing approaches. We also showed the ability of our method to directly discover geolocated events that are of interest for the user based on her feedback with crowdsourcing-based experiments. We believe that this work opens new directions for future research. For example, the event detection can be enhanced by thoroughly taking into account prior knowledge to detect really unexpected events and better propagate user feedback to the semantic neighborhood of liked events. Another interesting direction is to learn an explicit model of user interests and provide active learning based heuristics to foster the interactive process.

Chapter 7

Mining Subjectively Interesting Attributed Subgraphs

Contents

7.1 Introduction

As mentioned several times in this manuscript, the availability of network data has surged both due to the success of social media and ground-breaking discoveries in experimental sciences. Consequently, graph mining is one of the most studied tasks for the data mining community. The value of graphs stems from the presence of meaningful relationships among the data objects (the vertices). These can be explored by approaches as different as graph embeddings [Cai *et al.*, 2017]—which map the nodes of a graph into a low dimensional space while preserving the local and global graph structure as well as possible—, community detection [Fortunato, 2010]—the discovery of groups of vertices that somehow 'belong together'—, or subgraph mining—the identification of informative subgraphs.

Besides the relational structure, the so-called attributed graphs may carry information in the form of attribute-value pairs on vertices and/or edges. In this chapter, we focus on graphs with attribute-value pairs on vertices (vertex-attributed graphs). Mining interesting subgraphs in such attributed graphs is

challenging, both conceptually and computationally. Especially, a prominent problem is defining the quality of a subgraph. Arguably, a subgraph is more interesting if it is cohesive (e.g. high density, small diameter) and if its attribute values are similar within the subgraph while exceptional as compared to the rest of the graph. Few works in this direction exist. For example Atzmueller et al. [Atzmueller *et al.*, 2016] study mining communities (densely connected subgraphs) that can also be described well in terms of attribute values, while we proposed to look for exceptional subgraphs that are connected in chapter 4. Various existing quality measures are used in the first work and the second relies on Weighted Relative Accuracy. When dealing with several labels and thus with low support, patterns returned have often very low WRAcc values which makes them difficult to catch the semantics conveyed by this measure for the end-user. Furthermore, such measures – even if configurable to integrate some domain knowledge as in chapter 5 – do not take into account user prior knowledge.

This chapter addresses this issue by introducing a new flexible interestingness measure for exceptional subgraph patterns based on information theory, which trades-off informativeness with interpretability. The informativeness of the pattern is quantified with respect to specified prior knowledge available about the graph, making it a subjective measure. Informally speaking, a pattern is deemed more informative by our measure if it covers more vertices, and if the attribute values of the vertices within it deviate more strongly from the expected. The interpretability is quantified in terms of the complexity of the pattern, more specifically its description length. The pattern syntax we propose, and its description, characterize the vertices that are part of the pattern as the intersection of a given set of vertex neighborhoods (possibly with exceptions)—an approach we will argue is intuitive and effective. The proposed interestingness measure is the ratio of the informativeness and description length, thus representing the information density within the pattern.

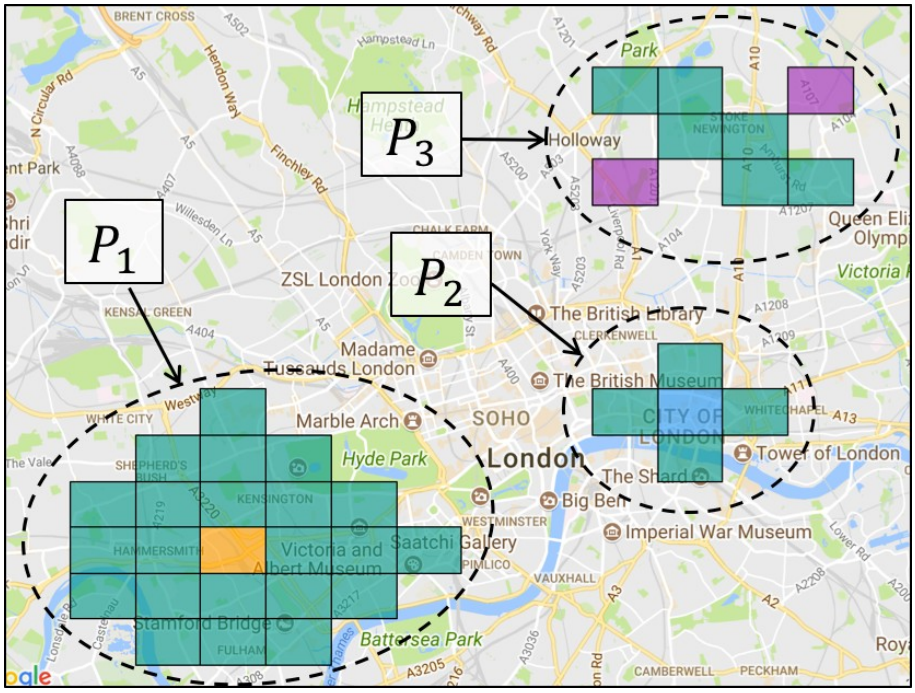
Fig. 7.1 shows example patterns from our method, applied to a setting where we want to explore the district structure of cities. The patterns should be interpreted as follows: certain attributes have surprisingly high/low values (marked $+/-$ respectively) in the given neighborhoods as compared to the prior knowledge—here just the average counts for each attribute. We ensure the regions are localized by forcing them to have a description of the form *"all vertices that are within distance d_1 of vertex x AND within distance d_2 of vertex y AND etc."*; e.g., in pattern P_3 of Fig. 7.1, the covered areas (green blocks) are the intersection of blocks within distance three of either purple block.

This work, which is part of Anes Bendimerad's PhD thesis co-advised with Céline Robardet, was done in collaboration with Ahmad Mel, Jefrey Lijffijt and Tijl de Bie from Ghent University. The related paper is under review. Therefore, we will describe the algorithms in this chapter. In addition to the subjective interestingness defined for attributed graphs, the originality of this work is the aim to ease the user assimilation of some patterns by finding alternative description that are easier to assimilate. This rest of the chapter is organized as following. We present a pattern syntax for cohesive subgraphs with exceptional attributes in Sec. 7.2. We formalize their subjective interestingness in Sec. 7.3. We study how to mine such subgraphs efficiently in Sec. 7.4. In Sec. 7.5, we provide a thorough empirical study on three types of data to evaluate (1) the relevance of the subjective interestingness measure compared to state-of-the-art methods, and (2) the efficiency of the algorithms. We draw conclusions in Sec. 7.6.

7.2 Cohesive Subgraphs with Exceptional Attributes (CSEA)

Before formally introducing the pattern syntax we are interested in, let us establish some notation.

An *attributed graph* is denoted $G = (V, E, \hat{A})$, where V is a set of n vertices, $E \subseteq V \times V$ is a set of m edges, and $\hat{A}$ is a set of p numerical attributes on vertices (formally, functions mapping a vertex onto an attribute value), with $\hat{a}(v) \in \mathrm{Dom}_a$ denoting the value of attribute $\hat{a} \in \hat{A}$ on $v \in V$. We use hats in $\hat{a}$ and $\hat{A}$ to signify the empirical values of the attributes, whereas a and A denote (possibly random) variables



$P_1 : \{\text{food}\}^+$
$P_2 : \{\text{professional, nightlife, outdoors, college}\}^+$
$P_3 : \{\text{nightlife, food}\}^+\{\text{college}\}^-$

Figure 7.1: Examples on Foursquare graph that covers the presence of various types of venues in London. P_1: around the orange block (West and South of Hyde Park) there are 'surprisingly' many restaurants, except in the center of that area (which is average). P_2: in the City several types of venues are consistently overrepresented. P_3: around Hackney there is a strip of blocks with a lot of nightlife and food venues but few educational places.

over the same domains. With $N_d(v)$ we denote the neighborhood of range d of a vertex v, i.e., the set of vertices whose graph geodesic distance[31] to v is at most d:

$$N_d(v) = \{u \in V \mid dist(v,u) \leq d\}.$$

Cohesive Subgraphs with Exceptional Attributes (CSEA). As described in the introduction, we are interested in patterns that inform the user that a set of attributes has exceptional values throughout a set of vertices in the graph.

Definition 7.1 (CSEA pattern)**.** *A* CSEA pattern *is defined as a tuple (U,S), where $U \subseteq V$ is a set of vertices in the graph, and S is a set of* characteristics *of these vertices, that is to say S is made of restrictions on the value domains of some attributes of A. More specifically, $S \subseteq \{(a,[k_a,\ell_a]) \mid a \in A\}$. Furthermore, to be a CSEA pattern, (U,S) has to be contained in G, i.e.*

$$\forall (a,[k_a,\ell_a]) \in S \text{ and } \forall u \in U,\ k_a \leq \hat{a}(u) \leq \ell_a.$$

Informally speaking, a CSEA pattern is more informative if the ranges in S are smaller, as then it conveys more information to the data analyst. This defines a partial order relation over the characteristics:

$$\begin{aligned} S \preceq S' \Leftrightarrow & \forall (a,[k'_a,l'_a]) \in S' : \\ & \exists (a,[k_a,l_a]) \in S \text{ with } [k_a,l_a] \subseteq [k'_a,l'_a]. \end{aligned}$$

A 'smaller' characteristic in this partial order is more specific and thus more informative. We will make this more formal in Sec. 7.3.1.

At the same time, a CSEA pattern (U,S) is more interesting if its description is more concise in some *intuitive description*. Thus, along with the pattern syntax, we must also specify how a pattern from this language will be intuitively described. In most applications, $|U|$ is much richer than $|S|$ and requires much more effort to be analyzed, a particular attention on vertices description must be paid. To this end, we propose to describe the set of vertices U as a neighborhood of a specified range from a given specified vertex, or more generally as the intersection of a set of such neighborhoods. For enhanced expressive power, we additionally allow for the description to specify exceptions on the above: vertices that do fall within this (intersection of) neighborhood(s), but which are to be excluded from U. Exceptions reduce the interestingness of a pattern, but greatly increase the expressive power of the CSEA pattern syntax.

A premise of this work is that this way of describing the set U is intuitive for human analysts, such that the length of the description of a pattern, as discussed in detail in Sec. 7.3.2, is a good measure of the complexity to assimilate or understand it. Our qualitative experiments in Sec. 7.5 do indeed confirm this is the case.

7.3 Subjective Interestingness of a CSEA

The previous sections already hinted at the fact that we will formalize the interestingness of a CSEA pattern (U,S) by trading off its information content with its description length. Here we show how the FORSIED framework, introduced in [Bie, 2011a, Bie, 2011b] for formalizing subjective interestingness of patterns, can be used for this purpose. The information content depends on both U and S. It is larger when more vertices are involved, when the intervals are narrower, and when they are more extreme. We will henceforth denote the information content as $\mathrm{IC}(U,S)$. The description length also depends on U and S and will be denoted as $\mathrm{DL}(U,S)$. The subjective interestingness of a CSEA pattern (U,S) is then expressed as:

[31] The graph geodesic distance between two vertices is the number of edges in a shortest path connecting them.

$$\mathrm{SI}(U,S) = \frac{\mathrm{IC}(U,S)}{\mathrm{DL}(U,S)}.$$

The power of the FORSIED framework is that it quantifies the information content of a pattern against a prior belief state about the data. It rigorously models the fact that the more plausible the data is (subjectively) according to a user or (objectively) under a specified model, the less information a user receives, and thus the smaller the information content ought to be.

This is achieved by modeling the prior beliefs of the user as the Maximum Entropy (MaxEnt) distribution subject to any stated prior beliefs the user may hold about the data. This distribution is referred to as the *background distribution*. The information content $\mathrm{IC}(U,S)$ of a CSEA pattern (U,S) is then formalized as minus the logarithm of the probability that the pattern is present under the background distribution (also called the self-information or surprisal) [Cover and Thomas, 1991]:

$$\mathrm{IC}(U,S) = -\log(\Pr(U,S)).$$

In Sec. 7.3.1, we first discuss in greater detail which prior beliefs could be appropriate for CSEA patterns, and how to infer the corresponding background distribution. Then, in Sec. 7.3.2, we discuss in detail how the description length $\mathrm{DL}(U,S)$ can be computed.

7.3.1 The Information Content of a CSEA Pattern

Positive integers as attributes. For concreteness, let us consider the situation where the attributes are positive integers ($a : V \to \mathbb{N}, \forall a \in A$), as will be our main focus throughout this chapter.[32] For example, if the vertices are geographical regions (with edges connecting vertices of neighboring regions), then the attributes could be counts of particular types of places in the region (e.g. one attribute could be the number of shops). It is clear that it is less informative to know that an attribute value is large in a large region than it would be in a small region. Similarly, a large value for an attribute that is generally large is less informative than if it were generally small. The above is only true, however, if the user knows (or believes) *a priori* at least approximately what these averages are for each attribute, and what the 'size' is of each region. Such prior beliefs can be formalized as equality constraints on the values of the attributes A on all vertices, or mathematically:

$$\sum_A \Pr(A) \left(\sum_{a \in A} a(v) \right) = \sum_{\hat{a} \in \hat{A}} \hat{a}(v), \quad \forall v \in V,$$

$$\sum_A \Pr(A) \left(\sum_{v \in V} a(v) \right) = \sum_{v \in V} \hat{a}(v), \quad \forall a \in A.$$

The MaxEnt background distribution can then be found as the probability distribution Pr maximizing the entropy $-\sum_A \Pr(A) \log \Pr(A)$, subject to these constraints and the normalization $\sum_A \Pr(A) = 1$.

As shown in [Bie, 2011b], the optimal solution of this optimization problem is a product of independent Geometric distributions, one for each vertex attribute-value $a(v)$. Each of these Geometric distributions is of the form $\Pr(a(v) = z) = p_{av} \cdot (1 - p_{av})^z$, $z \in \mathbb{N}$, where p_{av} is the success probability and it is given by: $p_{av} = 1 - exp(\lambda_a^r + \lambda_v^c)$, with λ_a^r and λ_v^c the Lagrange multipliers corresponding to the two constraint types. The optimal values of these multipliers can be found by solving the convex Lagrange dual optimization problem.

[32] The results presented can be extended relatively directly for other cases.

Given these Geometric distributions for the attribute values under the background distribution, we can now compute the probability of a pattern (U,S) as follows:

$$\begin{aligned}\Pr(U,S) &= \prod_{v\in U}\prod_{(a,[k_a,\ell_a])\in S}\Pr(a(v)\in[k_a,\ell_a]),\\ &= \prod_{v\in U}\prod_{(a,[k_a,\ell_a])\in S}\left((1-p_{av})^{k_a}-(1-p_{av})^{\ell_a+1}\right).\end{aligned}$$

This can be used directly to compute the information content of a pattern on given data, as the negative log of this probability. However, the pattern syntax is not directly suited to be applied to count data, when different vertices have strongly differing total counts. The reason is that the interval of each attribute is the same across vertices, which is desirable to keep the syntax understandable. Yet, if neighboring regions have very different total counts, it is less likely to find any patterns, and, even if we do, end-users would still need to know the total counts to interpret the patterns properly, as the same interval is not equally surprising for each region.

***p*-values as attributes.** To address this problem, we propose to search for the patterns not on the counts themselves, but rather on their *significance* (i.e., p-value or tail probability), computed with the background distribution as null hypothesis in a one-sided test. More specifically, we define the quantities $\hat{c}_a(v)$ as

$$\begin{aligned}\hat{c}_a(v) &\triangleq \Pr(a(v)\geq\hat{a}(v)),\\ &= (1-p_{av})^{\hat{a}(v)},\end{aligned}$$

and use this instead of the original attributes $\hat{a}(v)$. This transformation of $\hat{a}(v)$ to $\hat{c}_a(v)$ can be regarded as a principled normalization of the attribute values to make them comparable across vertices.

To compute the IC of a pattern with the transformed attributes $\hat{c}_a$, we must be able to evaluate the probability that $c_a(v)$ falls within a specified interval $[k_{c_a},\ell_{c_a}]$ under the background distribution for $a(v)$. This is given by:

$$\begin{aligned}&\Pr(c_a(v)\in[k_{c_a},\ell_{c_a}]) = \Pr\left((1-p_{av})^{a(v)}\in[k_{c_a},\ell_{c_a}]\right),\\ &= \Pr\left(a(v)\leq\frac{\log(k_{c_a})}{\log(1-p_{av})}\wedge a(v)\geq\frac{\log(\ell_{c_a})}{\log(1-p_{av})}\right),\\ &= (1-p_{av})^{\log_{1-p_{av}}(\ell_{c_a})}-(1-p_{av})^{\log_{1-p_{va}}(k_{c_a})+1},\\ &= \ell_{c_a}-(1-p_{av})\cdot k_{c_a} = \ell_{c_a}-k_{c_a}+p_{av}k_{c_a}.\end{aligned}$$

Thus, the IC of a pattern on the transformed attributes $\hat{c}$ can be calculated as:

$$\begin{aligned}\mathrm{IC}(U,S) &= -\log(\Pr(U,S)),\\ &= -\sum_{(a,[k_{ca},\ell_{ca}])\in S}\sum_{v\in U}\log(\ell_{c_a}-k_{c_a}+p_{av}k_{c_a}).\end{aligned}\tag{7.1}$$

In this work, we focus on intervals $[k_{c_a},\ell_{c_a}]$ where either $k_{c_a}=0$ (the minimal value) and $\ell_{c_a}<0.5$, or $\ell_{c_a}=1$ (the maximal value) and $k_{c_a}>0.5$. Such intervals state that the values of an attribute are all significantly large[33] or significantly small respectively, for all vertices in U. We argue such intervals are easiest to interpret. The logarithmic terms in Eq. (7.1) then simplify to $\log(\ell_{c_a})$ and $\log(1-k_{c_a}+p_{av}k_{c_a})$ respectively.

[33]Note empty regions have tail probabilities $\hat{c}_a(v)=1$ for any attribute and thus fall within any upper interval, but also $\mathrm{IC}=0$ for any attribute of that region as both $l_{c_a}=1$ and $p_{av}=1$.

7.3.2 Description Length

The description length measures the complexity of communicating a pattern (U,S) to the user. It can be defined as the complexity of communicating U and S:

$$\mathrm{DL}(U,S) = \mathrm{DL}_A(S) + \mathrm{DL}_V(U)$$

Where $\mathrm{DL}_A(S)$ (resp. $\mathrm{DL}_V(U)$) is the description length of S (resp. U). We define $\mathrm{DL}_A(S)$ as:

$$\mathrm{DL}_A(S) = (|S|+1)\cdot\log(|A|) + \sum_{(a,[k_{c_a},\ell_{c_a}])\in S}(1+\log(M_a))$$

with $M_a = |\{\hat{a}(v) \mid v \in V\}|$, the number of distinct values of a on the graph. More precisely, the first term accounts for the encoding of the attributes that are restricted. Encoding an attribute over $|A|$ possibilities costs $\log(|A|)$ bits. We do this encoding $(|S|+1)$ times, one for each attribute in S plus one for the length of S. The second term is the length of the encoding of restriction $(a,[k_{c_a},\ell_{c_a}]) \in S$. One bit is used to specify the type of interval ($[0,x]$ or $[x,1]$) and the encoding of the other bound of the interval is in logarithm of the number of distinct values of a on the graph.

As mentioned above, we describe the vertex set U in the pattern as (the intersection of) a set of neighborhoods $N_d(v)$, $v \in V$, with a set of exceptions: vertices are in the intersection but not part of U. The length of such a description is the sum of the description lengths of the neighborhoods and the exceptions. More formally, let us define the set of all neighborhoods $\mathscr{N} = \{N_d(v) \mid v \in V \wedge d \in [\![0,D]\!]\}$ (with D the maximum range d considered), and let $\mathscr{N}(U) = \{N_d(v) \in \mathscr{N} \mid U \subseteq N_d(v)\}$ be the subset of neighborhoods that contain U. The length of a description of the set U as the intersection of all neighborhoods in a subset $X \subseteq \mathscr{N}(U)$, along with the set of exceptions $\mathrm{exc}(X,U) \triangleq \cap_{N_d(v)\in X} N_d(v) \setminus U$, is then quantified by the function $f : 2^{\mathscr{N}(U)} \times U \longrightarrow \mathbb{R}$ defined as:

$$f(X,U) = (|X|+1)\cdot\log(|\mathscr{N}|) + (|\mathrm{exc}(X,U)|+1)\cdot\log(|\cap_{x\in X} x|).$$

Indeed, the first term accounts for the description of the number of neighborhoods ($\log(|\mathscr{N}|)$, and for describing which neighborhoods are involved ($|X|\log(|\mathscr{N}|)$). The second term accounts for the description of the number of exceptions ($\log(|\cap_{x\in X} x|)$), and for describing the exceptions themselves ($|\mathrm{exc}(X,U)|\log(|\cap_{x\in X} x|)$).

Clearly, there is generally no unique way to describe the set U. The best one is thus the one that minimizes f. This finally leads us to the definition of the description length of U as:

$$\mathrm{DL}_V(U) = \min_{X\subseteq\mathscr{N}(U)} f(X,U).$$

7.4 SIAS-Miner Algorithm

SIAS-Miner mines interesting patterns using an enumerate-and-rank approach. First, it enumerates all CSEA patterns (U,S) that are closed simultaneously with respect to U, S, and the neighborhood description. Second, it ranks patterns according to their SI values. The calculation of $\mathrm{IC}(U,S)$ and $\mathrm{DL}_A(S)$ is simple and direct. However, computing $\mathrm{DL}_V(U)$ is not trivial, since there are several ways to describe U and we are looking for the one minimizing $f(X,U)$. To achieve this goal, we propose an efficient algorithm DL_V-Optimise that calculates the minimal description of U and stores the result in the mapping structure *minDesc*. This algorithm is presented in Sec. 7.4.2.

Algorithm 4: SIAS-Miner (U, C, *Result*, *minDesc*)

Input: U: the current set of enumerated vertices, C: the set of candidates vertices.
Output: *Result*: the set of CSEAs, *minDesc*: a mapping structure that stores the minimum description of U for each pattern (U,S).

```
if C ≠ ∅ then
    // choose a candidate vertex with the fail first principle
    v ← argmin_{v∈C} |max_S(U ∪ {v})|
    U' ← clo(U ∪ {v})
    if U' ⊆ U ∪ C then
        // We prune candidates that cannot be used anymore
        C' ← {v ∈ C | max_S(U' ∪ {v}) ≠ ∅ ∧ 𝒩(U' ∪ {v}) ≠ ∅}
        SIAS-Miner(U', C' \ {v}, Result, minDesc)
    SIAS-Miner(U, C \ {v}, Result, minDesc)
else
    Result ← Result ∪ {(U, max_S(U))}
    bestDesc ← ∅
    DL_V-Optimise(U, ∅, 𝒩(U), bestDesc)
    minDesc[(U, max_S(U)] ← bestDesc
```

7.4.1 Pattern Enumeration

The exploration of the search space is based on subgraph enumeration. We only enumerate sets of vertices $U \subseteq V$ that are covered by a non empty characteristic $S \neq \emptyset$ and that can be described with neighbors $\mathcal{N}(U) \neq \emptyset$. Yet, a subgraph $G[U]$ can be covered by a large number of characteristics S that leads to as many redundant patterns. This can be avoided by only considering the most specific characteristic that covers U, as the other patterns do not bring any additional information. The function $max_S(U)$ defines the most specific characteristic, also made of significant intervals, associated to U:

$$max_S(U) = \{(a, [k_{c_a}, \ell_{c_a}]) \mid a \in A \wedge$$
$$((k_{c_a} = 0 \wedge \ell_{c_a} = \max_{v \in U} \hat{a}(v) \wedge \ell_{c_a} < 0.5)$$
$$\vee (\ell_{c_a} = 1 \wedge k_{c_a} = \min_{v \in U} \hat{a}(v) \wedge k_{c_a} > 0.5))\}$$

Moreover, it may happen that two sets of vertices U and U', such that $U' \subseteq U$, are covered by the same characteristic ($max_S(U') = max_S(U)$) and are described by the same neighborhoods ($\mathcal{N}(U') = \mathcal{N}(U)$). In that case, U brings more information than U' and its vertex description length $\mathrm{DL}_V(U)$ is lower or equal than $\mathrm{DL}_V(U')$, as all the descriptions $X \subseteq \mathcal{N}(U')$ also cover U with a lower or equal number of exceptions. Hence, the pattern $(U', max_S(U'))$ is not useful. This motivates the idea of only exploring patterns (U,S) that are closed with respect to both S and $\mathcal{N}(U)$. Closing a set of vertices U w.r.t. S and $\mathcal{N}(U)$ maximizes $\mathrm{IC}(U)$, and minimizes $\mathrm{DL}_V(U)$. Although this could increase the value of $\mathrm{DL}_A(S)$, we believe that this choice is very suitable as it allows to drastically improve the performance of the algorithm and reduce the size of the output, without altering the result quality.

We define the closure function $clo : 2^V \longrightarrow 2^V$ which is fundamental for our method. For a given set $U \subseteq V$, $clo(U)$ extends U by adding vertices that keep $max_S(U)$ and $\mathcal{N}(U)$ unchanged:

$$clo(U) = \{v \in V \mid max_S(v) \preceq max_S(U) \wedge \mathcal{N}(U) \subseteq \mathcal{N}(\{v\})\}$$

$clo(U)$ is indeed a closure function since it is extensive ($U \subseteq clo(U)$), idempotent ($clo(U) = clo(clo(U))$), and monotonic (if $X \subseteq Y$, then $clo(X) \subseteq clo(Y)$).

To enumerate patterns $(clo(U), max_S(U))$, SIAS-Miner uses the divide and conquer algorithm designed to efficiently compute closed structures described in [Boley *et al.*, 2010]. Initially, SIAS-Miner is

called with $U = \emptyset$ and $C = V$. In each recursive call, SIAS-Miner chooses a candidate $v \in C$ considering two optimizations to obtain a more balanced enumeration tree. If $U = \emptyset$, a vertex v is selected according to its degeneracy order[Eppstein and Strash, 2011] to prioritize first the vertices that should lead to small graphs. If $U \neq \emptyset$, a vertex v is selected following the fail first principle, i.e. the vertex that leads to the smallest characteristic so that to backtrack as soon as possible. Then, the closure of $clo(U \cup \{v\})$ is computed and stored in U'. If U' is included in $U \cup C$, we have the guarantee that U' has not been enumerated yet and the enumeration process continues. The candidates that cannot be added anymore to U' are pruned and SIAS-Miner is recursively called on U' (Line 4). Another recursive call is also made to enumerate subgraphs that do not contain $\{v\}$ (Line 4). When C is empty, the closed pattern $(U, max_S(U))$ is stored in *Result*, and DL_V-Optimise is called to compute $DL_V(U)$ that is finally stored in *minDesc*.

7.4.2 Computing $DL_V(U)$

Computing $DL_V(U)$ is NP-Hard. In fact, if we replace $(log(|\cap_{x \in X} x|)$ in $f(X,U)$ with a constant value, this problem becomes equivalent to the weighted set cover: it consists in finding the optimal cover of the set $\overline{U}$ based on unions of complements $\overline{N_i(v)}$ and exceptions $\{v\}$ such that $v \in \overline{U}$. Nevertheless, we propose a branch-and-bound approach that takes benefit from several optimisation techniques to solve this problem on instances of interest.

In order to find the optimal description of a set of vertices U, we explore the search space $2^{\mathcal{N}(U)}$ with a branch-and-bound approach described in Algorithm 5. Let X and Cand be subsets of $\mathcal{N}(U)$ that are respectively the current enumerated description and the potential candidates that can be used to describe U. Initially, DL_V-Optimise is called with $X = \emptyset$ and Cand $= \mathcal{N}(U)$. In each call, a neighborhood $e \in$ Cand is chosen and used to recursively explore two branches: one made of the descriptions that contain e (by adding e to X), and the other one made of descriptions that do not contain e (by removing e from Cand). Several pruning techniques are used in order to reduce the search space and are detailed below.

Algorithm 5: DL_V-Optimise(U, X, Cand, bestDesc)

Input: U the set to describe, X, $Cand \subseteq \mathcal{N}(U)$ the current description and the candidates, bestDesc the current best description found.
Output: bestDesc the best description of the current search sub-space.
if $LB(X,U,Cand) < f(bestDesc,U)$ **then**
  if $Cand \neq \emptyset$ **then**
    pruneUseless(U, X, Cand)
    pruneLowerBounded(U, X, Cand)
    $e \leftarrow \text{argmin}_{e' \in \text{Cand}} f(X \cup \{e'\}, U)$
    DL_V-Optimise(U, $X \cup \{e\}$, Cand $\setminus \{e\}$, bestDesc)
    DL_V-Optimise(U, X, Cand $\setminus \{e\}$, *bestDesc*)
  else if $f(X,U) < f(bestDesc,U)$ **then**
    bestDesc $\leftarrow X$

Function LB (line 1) lower bounds the lengths of the descriptions that can be generated in the subsequent recursive calls of DL_V-Optimise. If *LB* is higher or equal than the length of the current best description of U $f(\text{bestDesc}, U)$, there is no need to carry on the exploration of the search sub-space as no further description can improve $DL_V(U)$. The principle of *LB* is to evaluate the maximum reduction in exceptions that can be obtained when description X is extended with neighborhoods of Y: $gain_Y(X,U) = |\text{exc}(X,U)| - |\text{exc}(X \cup Y, U)|$, with $Y \subseteq$ Cand. This function can be rewritten using

neighborhood complements as $gain_Y(X,U) = |\cup_{y\in Y}(\overline{y}\cap \text{exc}(X,U))|$ [34]. We obtain then a simple upper bound of the gain function using the ordered set $\{g_1,...,g_{|\text{Cand}|}\}$ of $\{gain_{\{e\}}(X,U) \mid e \in \text{Cand}\}$ such that $g_i \geq g_j$ if $i \leq j$:

Property 7.1. *$gain_Y(X,U) \leq \sum_{i=1}^{|Y|} g_i$, for $Y \subseteq Cand$.*

Proof. Since the size of the union of sets is lower than the sum of the set sizes, we have $gain_Y(X,U) \leq \sum_{y\in Y}|\overline{y}\cap\text{exc}(X,U)| \leq \sum_{y\in Y} gain_{\{y\}}(X,U) \leq \sum_{i=1}^{|Y|} g_i$. □

This is the foundation of the function *LB* defined as

$$\begin{aligned} LB(X,U,\text{Cand}) = \min_{i\in[\![0,|\text{Cand}|]\!]}\{&(|X|+i+1)\times\log(|\mathscr{N}|)\\ &+\left(1+\max\left(|\text{exc}(\mathscr{N}(U),U)|,|\text{exc}(X,U)|-\textstyle\sum_{j=1}^{i} g_i\right)\right)\times\\ &\log\left(|U|+\max\left(|\text{exc}(\mathscr{N}(U),U)|,|\text{exc}(X,U)|-\textstyle\sum_{j=1}^{i} g_i\right)\right)\} \end{aligned}$$

Property 7.2. *$f(X\cup Y,U) \geq LB(X,U,Cand)$, $\forall Y \subseteq Cand$.*

Proof. Based on Property 7.1, we have $|\text{exc}(X\cup Y,U)| \geq |\text{exc}(X,U)| - \sum_{i=1}^{|Y|} g_i$, also, the minimum number of exceptions in this search space is $|\text{exc}(\mathscr{N}(U),U)|$, it implies that

$$|\text{exc}(X\cup Y,U)| \geq \max(|\text{exc}(\mathscr{N}(U),U)|, |\text{exc}(X,U)| - \sum_{i=1}^{|Y|} g_i).$$

This means that $f(X\cup Y,U) \geq (|X|+|Y|+1)\times\log(|\mathscr{N}|) + \left(1+\max\{|\text{exc}(\mathscr{N}(U),U)|, |\text{exc}(X,U)| - \sum_{j=1}^{|Y|} g_i\}\right)\cdot \log(|U|+\max\{|\text{exc}(\mathscr{N}(U),U)|, |\text{exc}(X,U)| - \sum_{j=1}^{|Y|} g_i\})$ and thus, $LB(X,U,\text{Cand}) \leq (|X|+|Y|+1)\times \log(|\mathscr{N}|) + (1+\max\{0, |\text{exc}(X,U)| - \sum_{j=1}^{|Y|} g_i\})\cdot\log(|U|+\max\{|\text{exc}(\mathscr{N}(U),U)|, |\text{exc}(X,U)| - \sum_{j=1}^{|Y|} g_i\})$ and it concludes the proof. □

In other terms, in the recursive calls, a description length will never be lower than $LB(X,U,\text{Cand})$. Thus, if its value is higher than the current best description, it is certain that no better description can be found.

Function pruneUseless line 3 removes candidate elements that can not improve the description length, that is candidates $e \in \text{Cand}$ for which $gain(\{e\},X,U) = 0$. Such element does not have the ability to reduce the number of exceptions in X. This also implies that e will not reduce the number of exceptions for descriptions $X\cup Y$, with $Y \subseteq \text{Cand}$. Thus, such elements will not decrease the description length of $X\cup Y$.

Algorithm 6: pruneUseless(U, X, Cand)

Cand $\leftarrow \{e \in \text{Cand} \mid \text{gain}(\{e\},X,U) > 0\}$

Function pruneLowerBounded line 4 removes a candidate $e \in \text{Cand}$ if there is a candidate $e' \in \text{Cand}$ that is always better than e for all descriptions produced in subsequent recursive calls.

[34] $= |\text{exc}(X,U)| - |\text{exc}(X\cup Y,U)| = |(\cap_{x\in X}x)\setminus U| - |(\cap_{e\in X\cup Y}e)\setminus U|$
$= |(\cap_{x\in X}x)\cap\overline{U}| - |((\cap_{x\in X}x)\cap\overline{U})\cap(\cap_{y\in Y}y)|$
$= |(\cap_{x\in X}x)\cap\overline{U})\setminus(\cap_{y\in Y}y)| = |(\cap_{x\in X}x)\setminus U)\cap\overline{(\cap_{y\in Y}y)}|$
$= |\text{exc}(X,U)\cap(\cup_{y\in Y}\overline{y})| = |\cup_{y\in Y}(\overline{y}\cap\text{exc}(X,U))|$

Property 7.3. *Let $e, e' \in Cand$ such that $exc(X \cup \{e\}, U) \subseteq exc(X \cup \{e'\}, U)$. Then, for all $Y \subseteq Cand \setminus \{e, e'\}$, we have $f(X \cup Y \cup \{e\}, U) \leq f(X \cup Y \cup \{e'\}, U)$*

Proof. The set of exceptions in $X \cup Y \cup \{e\}$ is equal to $\text{exc}(X \cup Y \cup \{e\}, U) = \text{exc}(X \cup \{e\}, U) \cap \text{exc}(Y, U)$. Since $\text{exc}(X \cup \{e\}, U) \subseteq \text{exc}(X \cup \{e'\}, U)$, then $\text{exc}(X \cup Y \cup \{e\}, U) \subseteq \text{exc}(X \cup Y \cup \{e'\}, U)$. As $|X \cup Y \cup \{e\}| = |X \cup Y \cup \{e'\}|$, we can conclude that $f(X \cup Y \cup \{e\}, U) \leq f(X \cup Y \cup \{e'\}, U)$. □

Based on Property 7.3, pruneLowerBounded removes elements $e' \in \text{Cand}$ such that $\text{exc}(X \cup \{e\}, U) \subseteq \text{exc}(X \cup \{e'\}, U)$. Notice that even if an element e'' has been removed due to the lower bound of e', the procedure is still correct since e'' is lower bound by e by the transitivity of inclusion.

Algorithm 7: pruneLowerBounded(U, X,Cand)

$\text{Cand} \leftarrow \{e_i \in \text{Cand} \mid \forall e_j \in \text{Cand} \setminus \{e_i\} : (\text{exc}(X \cup \{e_j\}, U) \not\subseteq \text{exc}(X \cup \{e_i\}, U))) \vee (\text{exc}(X \cup \{e_j\}, U) = \text{exc}(X \cup \{e_i\}, U) \wedge i < j)\}$

The last optimisation consists in choosing $e \in \text{Cand}$ that minimises $f(X \cup \{e\}, U)$ (line 5 of Algorithm 5). This makes it possible to quickly reach descriptions with low DL_V, and subsequently provide effective pruning when used in combination with *LB*.

7.5 Experiments

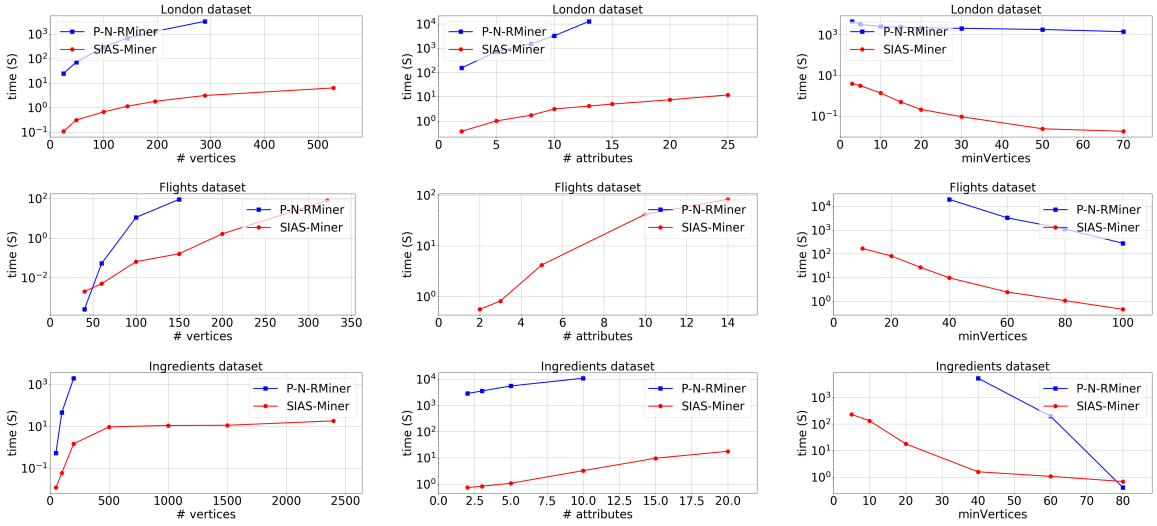


Figure 7.2: SIAS-Miner vs P-N-RMiner: runtime when varying $|V|$ (1st column), $|A|$ (2nd column) and a threshold on the minimum number of vertices in searched patterns (3rd column) for London graph ($D = 3$, 1st row), flights graph ($D = 1$, 2nd row), and ingredients graph ($D = 1$, 3rd row).

In this section, we report our experimental results. We start by describing the real-world datasets we used, as well as the questions we aim to answer. Then, we provide a thorough comparison with state-of-the-art algorithms: Cenergetics [Bendimerad *et al.*, 2017b], P-N-RMiner [Lijffijt *et al.*, 2016], and GAMER [Günnemann *et al.*, 2010] . Eventually, we provide a qualitative analysis that demonstrates

the ability of our approach to achieve the desired goal. For reproducibility purposes, the source code and the data are made available[35].

7.5.1 Experimental Setting

Experiments are performed on three real-world datasets:
− The *London graph* ($|V| = 289, |E| = 544, |\hat{A}| = 10$) is based on the social network Foursquare[36]. Each vertex depicts a district in London and edges link adjacent districts. Each attribute stands for the number of places of a given type (e.g. outdoors, colleges, restaurants, etc.) in each district. The total number of represented venues in this graph is 25029.
− The *ingredients graph* ($|V| = 2400, |E| = 7932, |\hat{A}| = 20$) is built from the data provided by Kaggle [37] and features given by Yummly [38]. Each vertex is a recipe ingredient. The attributes correspond to the number of occurrences of the ingredient in the recipes of each type of cuisine (greek, italian, mexican, thai, etc.). An edge exists between two ingredients if the Jaccard similarity between their recipes is higher than 0.03. The total number of used recipes is 39774.
− The *US flights graph* ($|V| = 322, |E| = 2039, |\hat{A}| = 14$) is a dataset provided by Kaggle [39] which contains information about flights between U.S.A airports in 2015. The vertices represents U.S.A airports and the attributes depict the number of flights made by each airline company in the airports. Edges connect two airports if there are at least 100 flights between them.
Considering all the numerical values of attributes is computationally expensive and would lead to redundant results, we pre-process each graph so that for each attribute, the values $c_a(v)$ are binned into five quantiles.

There is no approach that supports the discovery of subjectively interesting attributed subgraphs in the literature. Nevertheless, we identify some approaches whose goal share some similarities with ours. We consider them in our study:
− P-N-RMiner [Lijffijt *et al.*, 2016] is an algorithm that mines multi-relational datasets to enumerates a specific structure of patterns called Maximal Complete Connected Subsets (MCCS). Any vertex-attributed graph can be mapped to an entity-relational model where the pattern syntax of P-N-RMiner is equivalent to ours (each MCCS corresponds to a closed pattern $(clo(U), max_S(U))$ in our context). This means that P-N-RMiner is very suitable to evaluate the performance of our algorithm. The mapping from a graph to the required relational format is detailed in the companion page. Although the pattern syntax in this design is equivalent to our approach, our interestingness quantification is very different, because the information contained in the patterns shown to the user does not align with the ranking of P-N-RMiner. That is why we use P-N-RMiner as a baseline to only evaluate the time performance of our approach.
− Cenergetics [Bendimerad *et al.*, 2017b] aims at discovering connected subgraphs involving overrepresented and/or underrepresented attributes. It assesses exceptionality with the weighted relative accuracy (WRAcc) measure that accounts for margins but cannot account for other prior knowledge.
− GAMER [Günnemann *et al.*, 2010]: Given an attributed graph, this method finds dense subgraphs (quasi-cliques) where vertices show a high similarity in subsets of attributes. In these subgraphs, these attribute values fall into narrow intervals whose width does not exceed a specified threshold W. The main difference with our approach is that GAMER looks only for similarity and cohesiveness, but not exceptionality and surprisingness.

In this experimental study, our aim is to answer the following questions: What is the efficiency of SIAS-Miner regarding to graph dimensions? is SIAS-Miner able to deal with real world datasets? what

[35] `goo.gl/FgW2A1`
[36] https://foursquare.com
[37] `https://www.kaggle.com/kaggle/recipe-ingredients-dataset`
[38] `https://www.yummly.com/`
[39] https://www.kaggle.com/usdot/flight-delays/data

are the differences between the results of our approach and those of the considered baselines? what about the relevance of the CSEA patterns?

7.5.2 Quantitative Experiments

Fig. 7.2 reports the runtime of SIAS-Miner and P-N-RMiner according to the number of vertices, the number of attributes and the minimum number of vertices of searched patterns, for each of the datasets. The points that are not displayed in the curves of P-N-RMiner are the ones that exceeded a time limit of 10^4 seconds. For example, when we variated the attributes in the ingredients dataset, P-N-RMiner was not able to finish any configuration in less than 10^4 seconds. These tests reveal that SIAS-Miner outperforms P-N-RMiner in all the datasets in almost all the configurations, and the difference is generally between 2 and 4 orders of magnitudes. Although P-N-RMiner is a principled algorithm that uses several advanced optimization techniques, SIAS-Miner is faster since it is particularly defined to deal with attributed graphs, and it takes benefits from several specificities of this structures to well optimize the exploration of the search space.

7.5.3 Qualitative Experiments

The goal is to compare the properties of the patterns found by SIAS-Miner with those of Cenergetics and GAMER. We do not consider P-N-RMiner because the model used to assess the quality of patterns is not adapted to the goal of mining attributed graphs. We first compute a summary of patterns obtained by SIAS-Miner and Cenergetics based on Jaccard similarity in order to have diversified patterns to study. This set only contains patterns whose pairwise Jaccard similarity is lower than 0.6. This step is not applied on GAMER results since it already summarizes patterns internally with a similar approach. For information, detailed runtimes are provided in the companion page: SIAS-Miner and GAMER have comparable times whereas Cenergetics is much faster, due to its simpler and less expressive pattern syntax. In the following, we compare the properties depicted in Fig 7.3, and corresponding to the top 200 diversified patterns of each approach:

- Density and relative degree: The density of a pattern (U,S) is $\frac{2\times|E(U)|}{|U|\times(|U|-1)}$ where $|E(U)|$ is the number of edges in the induced subgraph $G[U]$, and the relative degree of a vertex v in a set of vertices U is $\frac{|N_1(v)\cap U|}{|U|-1}$. These properties are the highest for GAMER, this can be explained by the fact that its patterns are made of quasi-cliques, which makes them denser. In Flights and Ingredients datasets, the density and the degree for SIAS-Miner are higher than those of Cenergetics, and they are lower in London dataset. In fact, for SIAS-Miner, D was set to 3 in London dataset, and to 1 in Flights and Ingredients datasets, the higher the value of D, the sparser the results can be. In general, Cenergetics patterns have a low density and relative degree, because this approach requires only the connectivity of vertices in the patterns, some of these patterns can be very sparse subgraphs.

- Diameter: the diameter of a subgraph U is the maximum pairwise distance between vertices of U. GAMER patterns have the smallest average diameter, while Cenergetics patterns have the highest ones. The diameter of SIAS-Miner is comparable to the one of Cenergetics in London dataset ($D = 3$), but it is smaller in Flights and Ingredients datasets ($D = 1$).

- Size and number of covered attributes: The size of a pattern (U,S) corresponds to $|U|$ and the number of covered attributes is $|S|$. GAMER has patterns with the smallest average size, this is reasonable because it requires a harder constraints on the structure of patterns, which is the quasi-cliqueness. This small size of U allows GAMER patterns to be covered with a larger number

of attributes comparing with the other approaches.

- Contrast of attributes: given the found patterns (U,S) we want to measure how much the covered attributes in S are over (or under) expressed in U. First, we define the constrast of a given attribute a in a given set of vertices U as the absolute difference between its average ratio in U and its overal average ratio: $contrast(a,U) = |\frac{1}{|U|} \times \sum_{v \in U} \frac{\hat{a}(v)}{\hat{A}(v)} - \frac{1}{|V|} \times \sum_{v \in V} \frac{\hat{a}(v)}{\hat{A}(v)}|$, with $\hat{A}(v) = \sum_{a \in A} \hat{a}(v)$. The contrast of a pattern (U,S) is the average contrast $contrast(a,U)$ among the attributes a that appear in S. As expected, GAMER has the minimum values of contrasts for all the datasets, indeed, GAMER is only interested by the similarity of attributes in the pattern but not by their exceptionality. The contrasts for SIAS-Miner and Cenergetics are higher, and they are comparable in London and Flights datasets, while it is higher for SIAS-Miner in Ingredients dataset.

To conclude, there is a clear structural difference between patterns of the three approaches. GAMER finds denser subgraphs, and Cenergetics patterns are generally the sparsest ones. GAMER does not look to the exceptionality of attributes in the patterns, but only for their similarities. Another major difference is the possibility of integrating different prior beliefs in the MaxEnt model of SIAS-Miner, we have only studied attributes and vertices margins as constraints in this chapter, but several other prior beliefs can also be taken into account.

7.5.4 Illustrative Results

We show some patterns that SIAS-Miner discovered in London graph. We chose 5 of the non overlapping patterns that belong to the top 100 and we report them in Fig. 7.4. Green cells represents vertices covered by a CSEA pattern while blue cells are the centers, purples cells are the centers that do not belong to the pattern, orange cells are centers that are also exception (i.e., behave differently from the pattern but covered by the description) and the red cells are normal exceptions. Each of these patterns is described by at most two neighborhoods. For example, P_1 covers the distance 3 neighbors of the orange vertex, with an over expression of Food venues. P_{39} is described by the intersection of the distance 3 neighbors of the blue vertices. It covers the City of London where there is a significant amount of professional venues, college, universities, and outdoor venues.

ID	**Characteristics:** $S = \{(a, [\ell_{c_a}, k_{c_a}])\}$	$SI(P)$
P_1	{food: $[0,0.31]\}^+$	0.69
P_{10}	{food: $[0,0.46]\}^+$, {art: $[0.57,1]$, college: $[1,1]$, event: $[1,1]\}^-$	0.54
P_{31}	{nightlife: $[0,0.44]$, food: $[0,0.31]\}^+$	0.49
P_{39}	{professional: $[0,0.40]$, : college $[0,0.46]$, outdoors $[0,0.47]\}^+$	0.48
P_{100}	{ food: $[0,0.46]\}^+$, {college: $[1,1]$, event: $[1,1]\}^-$	0.43

Table 7.1: Detailed characteristics of the patterns discovered in London dataset by SIAS-Miner (see Fig. 7.4).

We also report in Fig. 7.5 two patterns discovered by SIAS-Miner in Ingredients graph. P_{12} corresponds to a set of ingredients that appear a lot in Italian recipes. They are described as neighbors of mozarella cheese, with two exceptions. P_{23} consists in some ingredients that are over expressed on Indian and Japanese recipes. They can be expressed as the neighbors of both ghee and garlic paste, with 6 exceptions.

7.6 Conclusion

We have introduced a new pattern syntax in attributed graphs, called CSEA, that provides to the user a set of attributes that have exceptional values throughout a subset of vertices. The strength of CSEA lies in its independence to a notion of support to assess the interestingness of a pattern. Instead, the interestingness is defined based on information theory, as the ratio of the information content (IC) over the description length DL. The IC is the amount of information provided by showing the user a pattern. The quantification is based on the gain from a Maximum Entropy background model that delineates the current knowledge of a user. Using a generically applicable prior as background knowledge, we provide a quantification of exceptionality that (subjectively) appears to match our intuition. The DL assesses the complexity of reading a pattern, the user being interested in concise and intuitive descriptions. To this end, we proposed to describe a set of vertices as an intersection of neighborhoods of certain distance from certain vertices, the distance and vertices making up the description of the subgraph. We have proposed an effective algorithm that enumerates patterns of this language. Extensive empirical results on three real-world datasets confirm that CSEA patterns are intuitive, and the interestingness measure aligns well with actual subjective interestingness. This work opens up several avenues for further research such as incorporating non-ordinal attribute types in the pattern syntax and integrating other kinds of prior beliefs.

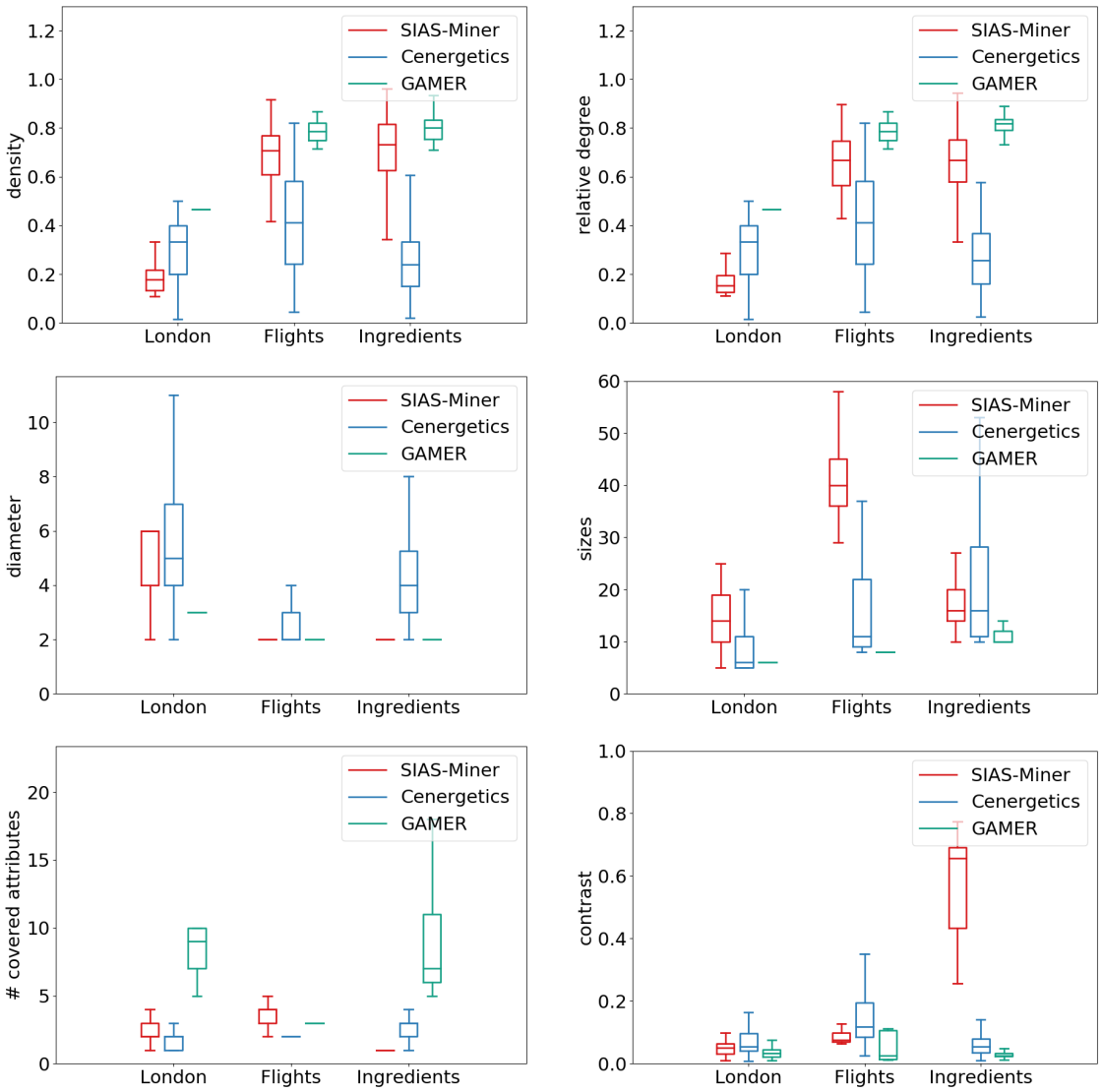


Figure 7.3: Comparisons of properties of found patterns by SIAS-Miner, Cenergetics, and GAMER in London, Ingredients, and Flights graphs.

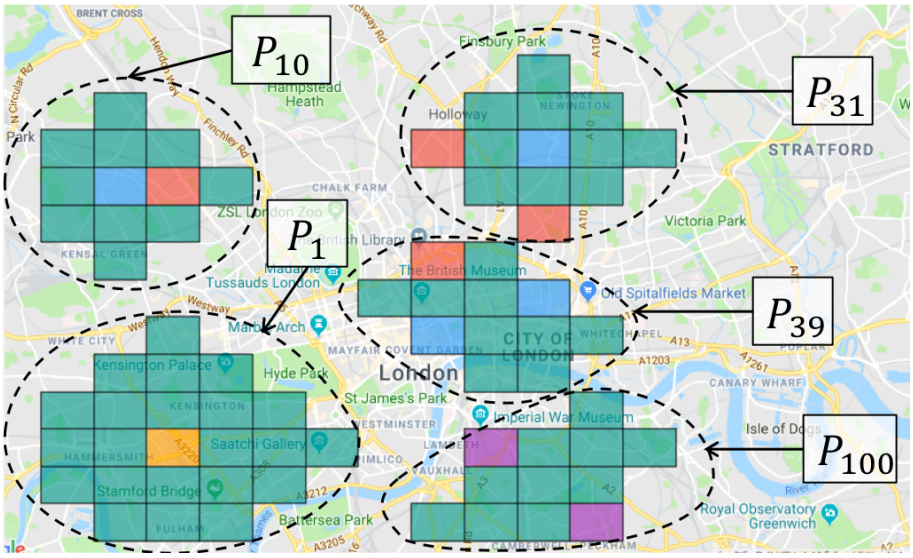
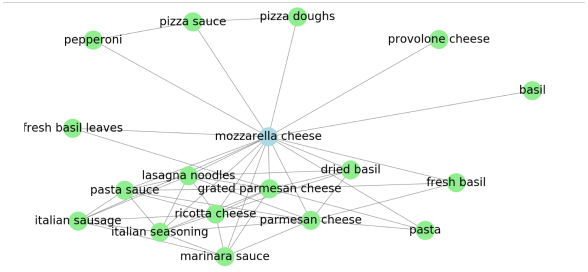
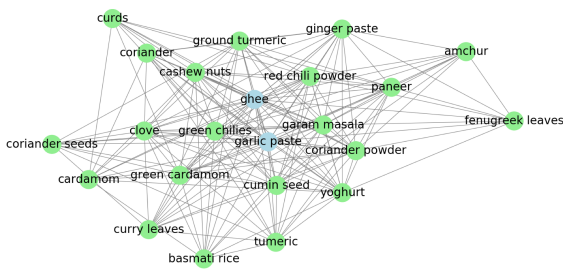


Figure 7.4: Patterns discovered in London graph by SIAS-Miner ($minVertices = 5$, $D = 3$). Details are provided in Tab. 7.1.



P_{12} : {Italian:$[0, 10^{-18}]$}$^+$
$SI(P_{12}) = 17.93$, $|U| = 18$, exceptions number: 2



P_{23} : {Indian:$[0, 10^{-18}]$, Japanese $[0, 0.44]$}$^+$
$SI(P_{23}) = 16.45$, $|U| = 26$, exceptions number: 6

Figure 7.5: Two patterns discovered in Ingredients graph ($minVertices = 10$, $D = 1$).

Chapter 8

Conclusion and Future Work

In this *Habilitation à Diriger des Recherches* thesis, I have tried to show the evolution of my research interests by focusing on augmented graphs. This is why I do not present all the results I contributed to (e.g., skypatterns, sequence mining). Through this manuscript, I strive both to highlight all the pattern mining opportunities the augmented graphs offer and to show my personal evolution in this domain, constantly trying to produce higher quality patterns. In part 1, I present several contributions in which we defined new pattern domains and their associated primitives. Some domain knowledge are partially taken into account (e.g., hierarchies on vertex attributes). Frankly speaking, we devised tools and the effort to obtain high quality patterns is left to the end-user, which is a real limitation of the application of such method. This effort to obtain high quality patterns has to be taken by the pattern mining and non exclusively by the end-user. This requires to take into consideration several aspects: the domain knowledge (chapter 5), the user feedback (chapter 6) and the user prior knowkedge (chapter 7).

Usually, *Habilitation à Diriger des Recherches* theses end with author's future work. I will not derogate from this tacit rule but let me first ask myself the following question:

Is there a future for pattern mining?

This question may seem a bit controversial but it deserves attention, especially with the (re-)rise of neural networks and more generally predictive models. The first answer to this question is **No**. This is what most of people whose knowledge on pattern mining is limited to frequent association rules think. Obviously, there is no future for (only) frequent pattern mining. This is not new. From the mid 2000's, the pattern mining community is aware about that. The problem is that outside its community, pattern mining is perceived as too simplistic tools that are far away from the users' needs. They are not totally wrong if we consider the pattern mining features that are available in analytic platform (e.g., Knime, Weka) or Python libraries. Of course the pattern mining community has produced a lot of amazing results to find high quality patterns that match with the user concerns. Our duty as pattern miners is to *democratize* the results of our research, trying to make them easily usable by any analysts. You may be sure, but my final and definitive answer to this question is: **Yes, there is a (great) future for pattern mining.** Analysts will always need of descriptive analysis. This is especially true as long as we have black box predictive models, we will need crystal clear descriptive solutions. Now let me detail how I envision my future research in pattern mining.

Data mining meets Visualization and Information Retrieval. The way we render the results to the user is as important as the algorithm that allows them to be discovered. Each (local) pattern identifies a subspace of the data. By nature, they are interesting if they are not trivial nor already known by the user, and potentially useful. The mining algorithm can assess the patterns with regard to the priors. However, a visualization that aims to project only the pattern into the data is not the most suitable for interacting

with the user. We have to highlight the interest of the pattern with respect to the user priors, the domain knowledge and also how this pattern stands out from the others. To this end, researchers in pattern mining must not neglect the decades of investigation done in the Visualization research community. The same observation can be done with the Information Retrieval community. Indeed, (re-)ranking results, taking into account user interest and her satisfaction, recommending are at the heart of the studies from this community. Although, there are some significant differences (e.g., the size of the search spaces) between them, pattern mining researchers must be aware of results produced by information retrieval researchers.

Causality. Which pattern miner has never met an ultra enthusiastic user by discovering the patterns / rules extracted by the mining algorithm ... until she understands that the conveyed semantics is not causality but simply co-occurrences. Pattern mining will bring as much as hope as disappointment as long as the problem of causality is not solved. It is a timely challenge that too few pattern miners tackle. Solutions to this difficult problem will bring a new breath to the pattern mining research and to the application of pattern mining algorithms.

Pattern mining as the corner stone of multidisciplinary projects. I am convinced that pattern mining is a great way to foster interdisciplinarity. The pattern syntax can be easily understood by any scientist. Domain knowledge of each discipline can be integrated to find more interesting patterns and this in a crystal clear way. Eventually, the discovered patterns are an excellent support for discussion between the scientists from different disciplines. For example, we worked on an olfaction study problem. This project requires expertise in neuroscience but also in chemistry. Through the patterns, chemists and neuroscientists can discuss and elicit hypotheses by sharing their own knowledge interpretation. We continue our collaboration in Neuroscience. Such projects are nourishing for our discipline: they are some excellent application areas of our algorithms but they also bring us new challenges in pattern mining.

Describe each phenomenon as simplest as possible: Towards multiple-pattern domains pattern set mining. A strong assumption is done when applying a pattern mining algorithm on real data: the user know – or is able to choose – the good pattern domain. However, choosing the good pattern domain is as difficult as setting the good thresholds for support or quality measures. Indeed, a too simple pattern domain would not make it possible to describe some complex phenomena while a too sophisticated pattern domain may uselessly provide over-described results leading to misleading interpretation. We have to automatically find to good level of description (in term of pattern syntax) to well describe phenomena within the data. Let me illustrate this with olfaction. We have odorant molecules that are described by thousands of attributes (i.e., numerical, Boolean, sequential, graph, 3d graphs) and that are also characterized by qualities (e.g., wood, floral, fruity) and also its hedonism value (i.e., a numerical value depicting how the molecule is perceived as pleasant or unpleasant by an expert). Boolean itemsets are enough to uncover some phenomena: the presence of a Sulfur atom likely leads to an unpleasant perception of the odorant molecules. For other phenomena, Boolean itemsets are insufficient to capture other phenomena. The consideration of numerical itemsets allows the discovery of more sophisticated conditions to depict unpleasant odorants: the high concentration of some molecules lead to an unpleasant perception. Isomeric molecules have the same 2d representations (planar graphs). Nevertheless, they can have very different odors and this requires to consider 3d graphs to catch the differences between them. I was impressed by the 2006 paper entitled “don’t be affraid of simpler patterns" that showed that in some cases itemsets are largely sufficient [Bringmann *et al.*, 2006]. This paper was cited more than 75 times but there is no attempt to automatically choose the good pattern domain yet. Obviously, for a dataset there is not a dedicated pattern domain. This dataset may contain several phenomena that have to be uncovered with different pattern domains. Therefore, an important challenge is the discovery of pattern set which involves patterns from different pattern domains. Subjective interestingness may allow to do that, assessing some trade-off between information content and assimilation cost for the end-user.

Data Mining meets Machine Learning: Towards sparse and interpretable Deep Neural Networks

(DNN). Predictive models based on DNN have become very powerful and are now widely used in a large variety of applications. However, the underlying deep learning methods tend to produce models that are very large as well as difficult to interpret and understand, which raises two issues. Firstly, the size of theses models may jeopardize their scalability and, therefore, has to be reduced without altering their performance. Secondly, the use of such methods in a data science context implies that users are scientists (social and climate scientists in this project) who aim to acquire new knowledge in their own field. Therefore, the models must be not only effective but also understandable. Pattern mining techniques can help to achieve this goal. We plan to tackle this challenge with Céline Robardet, Stefan Duffner and Christophe Garcia in the context of the project “Academics" (IDEX Lyon Scientific Breakthrough program 2018). We believe that new data mining approaches can make possible to analyze, understand and compress DNN:

- Compressing models is very important as they can be very large (up to several Gigabytes) and require a great amount of computation that cannot be parallelized (e.g., very deep architectures). However, there is a high redundancy in the computations and model parameters. It has been show that compression can save 10 to 100 times memory while keeping almost the same prediction ability [Chen *et al.*, 2015]. Most importantly, compressing the network by removing spurious information can help in its understanding and interpretation [Lipton, 2016], which previous work has mostly tried to tackle by specific visualization techniques [Zeiler and Fergus, 2014, Nguyen *et al.*, 2016] or by explicitly learning automatically extracted concepts [Dong *et al.*, 2017]. Pattern mining algorithms can also be used for analyzing neural activations, by identifying blocks in the weight matrices (or tensors) which represent noise and thus do not contribute to the target and final output activations, and by identifying paths of neuron activations strongly correlated with an output. We will consider neural network architectures that facilitate this mining process, e.g. by preferring partially-connected structures and sub-modules and by avoiding fully-connected parts as much as possible as they “diffuse" the extracted information across the whole neural network.
- A better understanding of DNNs requires also to work on both the input and the output of the models in several ways: (i) Integrating the priors to the models: data mining approaches can be used to characterize non-explicit priors; (ii) Using the results from data mining or other unsupervised learning techniques as priors; and (iii) Characterizing prediction errors and learning specific models for these “extreme" cases, e.g. geographically localized errors, or to bias training samples to better handle these errors.
- Investigating languages (i.e. pattern domains) that make DNNs interpretable or partially interpretable, and the definition of algorithms that make possible the discovery or learning of DNN descriptions and parameterizations with regard to a related language.

Bibliography

[Albert and Barabási, 2000] R. Albert and A.-L. Barabási. Topology of complex networks: Local events and universality. *Phys. Rev.*, 85:5234–5237, 2000.

[Anand *et al.*, 1995] Sarabjot S Anand, David A Bell, and John G Hughes. The role of domain knowledge in data mining. In *Proceedings of the fourth international conference on Information and knowledge management*, pages 37–43. ACM, 1995.

[Ashbrook and Starner, 2003] Daniel Ashbrook and Thad Starner. Using gps to learn significant locations and predict movement across multiple users. *Pers. Ub. Comput.*, 7(5):275–286, 2003.

[Asur and Huberman, 2010] S. Asur and B.-A. Huberman. Predicting the future with social media. In *WI-IAT*, pages 492–499. IEEE Computer Society, 2010.

[Atzmueller and Puppe, 2006] Martin Atzmueller and Frank Puppe. Sd-map - A fast algorithm for exhaustive subgroup discovery. In *PKDD 2006*, pages 6–17, 2006.

[Atzmueller *et al.*, 2016] Martin Atzmueller, Stephan Doerfel, and Folke Mitzlaff. Description-oriented community detection using exhaustive subgroup discovery. *Information Science*, 329:965–984, 2016.

[Atzmueller, 2016] Martin Atzmueller. Detecting community patterns capturing exceptional link trails. In *2016 IEEE/ACM ASONAM*, pages 757–764, 2016.

[Bastide *et al.*, 2000] Yves Bastide, Rafik Taouil, Nicolas Pasquier, Gerd Stumme, and Lotfi Lakhal. Mining frequent patterns with counting inference. *SIGKDD Explorations*, 2(2):66–75, 2000.

[Bay and Pazzani, 2001] Stephen D Bay and Michael J Pazzani. Detecting group differences: Mining contrast sets. *Data mining and knowledge discovery*, 5(3):213–246, 2001.

[Bayardo *et al.*, 2005] Roberto J. Bayardo, Bart Goethals, and Mohammed Javeed Zaki, editors. *FIMI '04, Proceedings of the IEEE ICDM Workshop on Frequent Itemset Mining Implementations, Brighton, UK, November 1, 2004*, volume 126 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2005.

[Bayir *et al.*, 2009] Murat Ali Bayir, Murat Demirbas, and Ahmet Cosar. Track me! a web based location tracking and analysis system. In *ISCIS*, pages 117–122. IEEE, 2009.

[Belfodil *et al.*, 2017] Adnene Belfodil, Sylvie Cazalens, Philippe Lamarre, and Marc Plantevit. Flash points: Discovering exceptional pairwise behaviors in vote or rating data. In *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2017, Skopje, Macedonia, September 18-22, 2017, Proceedings, Part II*, pages 442–458, 2017.

[Bendimerad *et al.*, 2016] Ahmed Anes Bendimerad, Marc Plantevit, and Céline Robardet. Unsupervised exceptional attributed sub-graph mining in urban data. In *IEEE 16th International Conference on Data Mining, ICDM 2016, December 12-15, 2016, Barcelona, Spain*, pages 21–30, 2016.

[Bendimerad *et al.*, 2017a] Ahmed Anes Bendimerad, Rémy Cazabet, Marc Plantevit, and Céline Robardet. Contextual subgraph discovery with mobility models. In *Complex Networks & Their Applications VI - Proceedings of Complex Networks 2017 (The Sixth International Conference on Complex Networks and Their Applications), COMPLEX NETWORKS 2017, Lyon, France, November 29 - December 1, 2017.*, pages 477–489, 2017.

[Bendimerad *et al.*, 2017b] Anes Bendimerad, Marc Plantevit, and Céline Robardet. Mining exceptional closed patterns in attributed graphs. *KAIS*, pages 1–25, 2017.

[Bendimerad *et al.*, 2018] Ahmed Anes Bendimerad, Marc Plantevit, and Céline Robardet. Mining exceptional closed patterns in attributed graphs. *Knowl. Inf. Syst.*, 56(1):1–25, 2018.

[Berlingerio *et al.*, 2009] Michele Berlingerio, Francesco Bonchi, Björn Bringmann, and Aristides Gionis. Mining graph evolution rules. In *European Conf. on Machine Learning and Princ. and Pract. of Knowl. Disc. in Databases (ECML/PKDD)*, pages 115–130, 2009.

[Bhuiyan and Al Hasan, 2016] M. Bhuiyan and M. Al Hasan. Interactive knowledge discovery from hidden data through sampling of frequent patterns. *Statistical Analysis and Data Mining*, 9(4):205–229, 2016.

[Bhuiyan *et al.*, 2012] Mansurul Bhuiyan, Snehasis Mukhopadhyay, and Mohammad Al Hasan. Interactive pattern mining on hidden data: a sampling-based solution. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 95–104. ACM, 2012.

[Bie, 2011a] Tijl De Bie. An information theoretic framework for data mining. In *KDD*, pages 564–572, 2011.

[Bie, 2011b] Tijl De Bie. Maximum entropy models and subjective interestingness. *Data Mining and Knowledge Discovery*, 23(3):407–446, 2011.

[Biemann, 2006] Chris Biemann. Chinese whispers: an efficient graph clustering algorithm and its application to natural language processing problems. In *Proceedings of the first workshop on graph based methods for natural language processing*, pages 73–80. Association for Computational Linguistics, 2006.

[Blockeel *et al.*, 2012] Hendrik Blockeel, Toon Calders, Élisa Fromont, Bart Goethals, Adriana Prado, and Céline Robardet. An inductive database system based on virtual mining views. *Data Min. Knowl. Discov.*, 24(1):247–287, 2012.

[Blondel *et al.*, 2008] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008, 2008.

[Boley *et al.*, 2010] Mario Boley, Tamás Horváth, Axel Poigné, and Stefan Wrobel. Listing closed sets of strongly accessible set systems with applications to data mining. *Theor. Comput. Sci.*, 411(3):691–700, 2010.

[Boley *et al.*, 2011] Mario Boley, Claudio Lucchese, Daniel Paurat, and Thomas Gärtner. Direct local pattern sampling by efficient two-step random procedures. In *ACM SIGKDD 2011*, pages 582–590, 2011.

[Bonchi and Lucchese, 2007] Francesco Bonchi and Claudio Lucchese. Extending the state-of-the-art of constraint-based pattern discovery. *Data Knowl. Eng.*, 60(2):377–399, 2007.

[Bonchi *et al.*, 2005] Francesco Bonchi, Fosca Giannotti, Alessio Mazzanti, and Dino Pedreschi. Exante: A preprocessing method for frequent-pattern mining. *IEEE Intelligent Systems*, 20(3):25–31, 2005.

[Borda, 2011] Monica Borda. *Fundamentals in information theory and coding*, volume 6. Springer, 2011.

[Borgwardt *et al.*, 2006] Karsten M. Borgwardt, Hans-Peter Kriegel, and Peter Wackersreuther. Pattern mining in frequent dynamic subgraphs. In *ICDM*, pages 818–822. IEEE, 2006.

[Boulicaut *et al.*, 2003] Jean-François Boulicaut, Artur Bykowski, and Christophe Rigotti. Free-sets: A condensed representation of boolean data for the approximation of frequency queries. *Data Min. Knowl. Discov.*, 7(1):5–22, 2003.

[Boulicaut *et al.*, 2016] Jean-François Boulicaut, Marc Plantevit, and Céline Robardet. Local pattern detection in attributed graphs. In Michaelis et al. [2016], pages 168–183.

[Brin and Page, 1998] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Comp. net. and ISDN systems*, 30(1-7):107–117, 1998.

[Bringmann and Nijssen, 2008] Björn Bringmann and Siegfried Nijssen. What is frequent in a single graph? In *PAKDD*, pages 858–863, 2008.

[Bringmann *et al.*, 2006] Björn Bringmann, Albrecht Zimmermann, Luc De Raedt, and Siegfried Nijssen. Don't be afraid of simpler patterns. In *Knowledge Discovery in Databases: PKDD 2006, 10th European Conference on Principles and Practice of Knowledge Discovery in Databases, Berlin, Germany, September 18-22, 2006, Proceedings*, pages 55–66, 2006.

[Budhathoki and Vreeken, 2018a] Kailash Budhathoki and Jilles Vreeken. Causal inference on event sequences. In *Proceedings of the 2018 SIAM International Conference on Data Mining, SDM 2018, May 3-5, 2018, San Diego Marriott Mission Valley, San Diego, CA, USA.*, pages 55–63, 2018.

[Budhathoki and Vreeken, 2018b] Kailash Budhathoki and Jilles Vreeken. Origo: causal inference by compression. *Knowl. Inf. Syst.*, 56(2):285–307, 2018.

[Buzmakov *et al.*, 2015] Aleksey Buzmakov, Sergei O. Kuznetsov, and Amedeo Napoli. Fast generation of best interval patterns for nonmonotonic constraints. In *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2015, Porto, Portugal, September 7-11, 2015, Proceedings, Part II*, pages 157–172, 2015.

[Cai *et al.*, 2017] HongYun Cai, Vincent W. Zheng, and Kevin Chen-Chuan Chang. A comprehensive survey of graph embedding. *CoRR*, 2017.

[Cakmak and Özsoyoglu, 2008] Ali Cakmak and Gultekin Özsoyoglu. Taxonomy-superimposed graph mining. In *EDBT*, pages 217–228, 2008.

[Calders and Goethals, 2002] Toon Calders and Bart Goethals. Mining all non-derivable frequent itemsets. In *Principles of Data Mining and Knowledge Discovery, 6th European Conference, PKDD 2002, Helsinki, Finland, August 19-23, 2002, Proceedings*, pages 74–85, 2002.

[Calders *et al.*, 2004] Toon Calders, Christophe Rigotti, and Jean-François Boulicaut. A survey on condensed representations for frequent sets. In *Constraint-Based Mining and Inductive Databases, European Workshop on Inductive Databases and Constraint Based Mining, Hinterzarten, Germany, March 11-13, 2004, Revised Selected Papers*, pages 64–80, 2004.

[Calders *et al.*, 2006] Toon Calders, Bart Goethals, and Szymon Jaroszewicz. Mining rank-correlated sets of numerical attributes. In *KDD*, pages 96–105, 2006.

[Carchiolo *et al.*, 2015] Vincenza Carchiolo, Alessandro Longheu, and Michele Malgeri. Using twitter data and sentiment analysis to study diseases dynamics. In *ITBAM 2015*, pages 16–24, 2015.

[Cazabet *et al.*, 2017] Remy Cazabet, Pierre Borgnat, and Pablo Jensen. Enhancing space-aware community detection using degree constrained spatial null model. In *Workshop CompleNet*, pages 47–55. Springer, 2017.

[Cellier *et al.*, 2015] Peggy Cellier, Thierry Charnois, Marc Plantevit, Christophe Rigotti, Bruno Crémilleux, Olivier Gandrillon, Jirí Kléma, and Jean-Luc Manguin. Sequential pattern mining for discovering gene interactions and their contextual information from biomedical texts. *J. Biomedical Semantics*, 6:27, 2015.

[Cerf *et al.*, 2009] Loïc Cerf, Jérémy Besson, Céline Robardet, and Jean-François Boulicaut. Closed patterns meet *n*-ary relations. *TKDD*, 3(1), 2009.

[Cerf *et al.*, 2013] Loïc Cerf, Jérémy Besson, Kim-Ngan Nguyen, and Jean-François Boulicaut. Closed and noise-tolerant patterns in n-ary relations. *Data Min. Knowl. Discov.*, 26(3):574–619, 2013.

[Chen and Neill, 2014] Feng Chen and Daniel B. Neill. Non-parametric scan statistics for event detection and forecasting in heterogeneous social media graphs. In *KDD '14*, pages 1166–1175, New York, NY, USA, 2014. ACM.

[Chen *et al.*, 2015] Wenlin Chen, James T. Wilson, Stephen Tyree, Kilian Q. Weinberger, and Yixin Chen. Compressing neural networks with the hashing trick. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, pages 2285–2294, 2015.

[Cormen *et al.*, 2009] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms (3. ed.)*. MIT Press, 2009.

[Cover and Thomas, 1991] Thomas M Cover and Joy A Thomas. Entropy, relative entropy and mutual information. *Elements of information theory*, 2:1–55, 1991.

[de Sá *et al.*, 2018] Cláudio Rebelo de Sá, Wouter Duivesteijn, Paulo Azevedo, Alípio Mário Jorge, Carlos Soares, and Arno Knobbe. Discovering a taste for the unusual: exceptional models for preference mining. *Machine Learning*, Jul 2018.

[Demirbas *et al.*, 2009] Murat Demirbas, Carole Rudra, Atri Rudra, and Murat Ali Bayir. imap: Indirect measurement of air pollution with cellphones. In *PerCom Workshops*, pages 1–6, 2009.

[Demsar, 2006] Janez Demsar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30, 2006.

[Desmier *et al.*, 2012] Elise Desmier, Marc Plantevit, Céline Robardet, and Jean-François Boulicaut. Cohesive co-evolution patterns in dynamic attributed graphs. In *Discovery Science*, pages 110–124, 2012.

[Desmier *et al.*, 2013] Elise Desmier, Marc Plantevit, Céline Robardet, and Jean-François Boulicaut. Trend mining in dynamic attributed graphs. In *ECML/PKDD*, pages 654–669, 2013.

[Desmier *et al.*, 2014] Elise Desmier, Marc Plantevit, Céline Robardet, and Jean-François Boulicaut. Granularity of co-evolution patterns in dynamic attributed graphs. In *Advances in Intelligent Data Analysis XIII - 13th International Symposium, IDA 2014, Leuven, Belgium, October 30 - November 1, 2014. Proceedings*, pages 84–95, 2014.

[Dong and Li, 1999] Guozhu Dong and Jinyan Li. Efficient mining of emerging patterns. In *ACM SIGKDD*, pages 43–52, 1999.

[Dong *et al.*, 2015] X. Dong, D. Mavroeidis, F. Calabrese, and P. Frossard. Multiscale event detection in social media. *Dami*, 29(5):1374–1405, 2015.

[Dong *et al.*, 2017] Yinpeng Dong, Hang Su, Jun Zhu, and Bo Zhang. Improving interpretability of deep neural networks with semantic information. *CoRR*, abs/1703.04096, 2017.

[Downar and Duivesteijn, 2017] Lennart Downar and Wouter Duivesteijn. Exceptionally monotone models - the rank correlation model class for exceptional model mining. *Knowl. Inf. Syst.*, 51(2):369–394, 2017.

[Duivesteijn *et al.*, 2010] Wouter Duivesteijn, Arno J. Knobbe, Ad Feelders, and Matthijs van Leeuwen. Subgroup discovery meets bayesian networks – an exceptional model mining approach. In *ICDM 2010*, pages 158–167, 2010.

[Duivesteijn *et al.*, 2016] Wouter Duivesteijn, Ad Feelders, and Arno J. Knobbe. Exceptional model mining - supervised descriptive local pattern mining with complex target concepts. *Data Min. Knowl. Discov.*, 30(1):47–98, 2016.

[Duivesteijn, 2014] Wouter Duivesteijn. A short survey of exceptional model mining: Exploring unusual interactions between multiple targets. In *2014 International Workshop on Multi-Target Prediction*, 2014.

[Dzyuba *et al.*, 2014] Vladimir Dzyuba, Matthijs van Leeuwen, Siegfried Nijssen, and Luc De Raedt. Interactive learning of pattern rankings. *International Journal on Artificial Intelligence Tools*, 23(06):1460026, 2014.

[Eppstein and Strash, 2011] David Eppstein and Darren Strash. Listing all maximal cliques in large sparse real-world graphs. In *International Symposium on Experimental Algorithms*, pages 364–375. Springer, 2011.

[Expert *et al.*, 2011] Paul Expert, Tim S Evans, Vincent D Blondel, and Renaud Lambiotte. Uncovering space-independent communities in spatial networks. *PNAS*, 108(19):7663–7668, 2011.

[Falher *et al.*, 2015] Géraud Le Falher, Aristides Gionis, and Michael Mathioudakis. Where is the soho of rome? measures and algorithms for finding similar neighborhoods in cities. In *ICWSM 2015*, pages 228–237, 2015.

[Fortunato, 2010] Santo Fortunato. Community detection in graphs. *Phys. rep.*, 486:75–174, 2010.

[Freeman, 1977] L. C. Freeman. A set of measures of centrality based on betweenness. *Sociometry*, 40(1):35–41, 1977.

[Galbrun and Miettinen, 2017] Esther Galbrun and Pauli Miettinen. *Redescription Mining*. Springer Briefs in Computer Science. Springer, 2017.

[Giannotti and Pedreschi, 2008] Fosca Giannotti and Dino Pedreschi. *Mobility, data mining and privacy*. Springer Science & Business Media, 2008.

[Gionis *et al.*, 2007] Aristides Gionis, Heikki Mannila, Taneli Mielikäinen, and Panayiotis Tsaparas. Assessing data mining results via swap randomization. *TKDD*, 1(3):14, 2007.

[Goethals and Zaki, 2004] Bart Goethals and Mohammed Javeed Zaki. Advances in frequent itemset mining implementations: report on fimi'03. *SIGKDD Explorations*, 6(1):109–117, 2004.

[Grosskreutz and Rüping, 2009] Henrik Grosskreutz and Stefan Rüping. On subgroup discovery in numerical domains. *Data Min. Knowl. Discov.*, 19(2):210–226, 2009.

[Grosskreutz *et al.*, 2013] Henrik Grosskreutz, Bastian Lang, and Daniel Trabold. A relevance criterion for sequential patterns. In *ECMLPKDD*, pages 369–384, 2013.

[Günnemann *et al.*, 2010] Stephan Günnemann, Ines Färber, Brigitte Boden, and Thomas Seidl. Subspace clustering meets dense subgraph mining. In *ICDM*, pages 845–850, 2010.

[Guns *et al.*, 2017] Tias Guns, Anton Dries, Siegfried Nijssen, Guido Tack, and Luc De Raedt. Miningzinc: A declarative framework for constraint-based mining. *Artif. Intell.*, 244:6–29, 2017.

[Hamon, 2015] Ronan Hamon. *Analysis of temporal networks using signal processing methods : Application to the bike-sharing system in Lyon*. Theses, Ecole normale supérieure de lyon - ENS LYON, September 2015.

[Han and Fu, 1999] Jiawei Han and Yongjian Fu. Mining multiple-level association rules in large databases. *IEEE Trans. Knowl. Data Eng.*, 11(5):798–804, 1999.

[Harrington and Cahill, 2004] Anthony Harrington and Vinny Cahill. Route profiling: putting context to work. In *SAC*, pages 1567–1573, 2004.

[He *et al.*, 2007] Q. He, K. Chang, and E.-P. Lim. Analyzing feature trajectories for event detection. In *SIGIR*, pages 207–214. ACM, 2007.

[Holat *et al.*, 2014] Pierre Holat, Marc Plantevit, Chedy Raïssi, Nadi Tomeh, Thierry Charnois, and Bruno Crémilleux. Sequence classification based on delta-free sequential patterns. In Ravi Kumar, Hannu Toivonen, Jian Pei, Joshua Zhexue Huang, and Xindong Wu, editors, *2014 IEEE International Conference on Data Mining, ICDM 2014, Shenzhen, China, December 14-17, 2014*, pages 170–179. IEEE, 2014.

[Ifrim *et al.*, 2014] G. Ifrim, B. Shi, and I. Brigadir. Event detection in twitter using aggressive filtering and hierarchical tweet clustering. In *snow@WWW*, pages 33–40, 2014.

[Imielinski and Mannila, 1996] Tomasz Imielinski and Heikki Mannila. A database perspective on knowledge discovery. *Communications of the ACM*, 39(11):58–64, 1996.

[Inokuchi, 2004] Akihiro Inokuchi. Mining generalized substructures from a set of labeled graphs. In *ICDM*, pages 415–418, 2004.

[Jiang and Pei, 2009] Daxin Jiang and Jian Pei. Mining frequent cross-graph quasi-cliques. *ACM TKDD*, 2(4):1–42, 2009.

[Kang *et al.*, 2011] U. Kang, Charalampos E. Tsourakakis, Ana Paula Appel, Christos Faloutsos, and Jure Leskovec. Hadi: Mining radii of large graphs. *ACM TKDD*, 5(2):8, 2011.

[Kaytoue *et al.*, 2015] Mehdi Kaytoue, Yoann Pitarch, Marc Plantevit, and Céline Robardet. What effects topological changes in dynamic graphs? - elucidating relationships between vertex attributes and the graph structure. *Social Netw. Analys. Mining*, 5(1):55:1–55:17, 2015.

[Kaytoue *et al.*, 2017] Mehdi Kaytoue, Marc Plantevit, Albrecht Zimmermann, Ahmed Anes Bendimerad, and Céline Robardet. Exceptional contextual subgraph mining. *Machine Learning*, 106(8):1171–1211, 2017.

[Khan *et al.*, 2010] Arijit Khan, Xifeng Yan, and Kun-Lung Wu. Towards proximity pattern mining in large graphs. In *SIGMOD*, pages 867–878, 2010.

[Khiari *et al.*, 2010] Mehdi Khiari, Patrice Boizumault, and Bruno Crémilleux. Constraint programming for mining n-ary patterns. In *Principles and Practice of Constraint Programming - CP 2010 - 16th International Conference, CP 2010, St. Andrews, Scotland, UK, September 6-10, 2010. Proceedings*, pages 552–567, 2010.

[Kleinberg, 2002] J.-M. Kleinberg. Bursty and hierarchical structure in streams. In *KDD*, pages 91–101, 2002.

[Klosgen, 1996] Willi Klosgen. Explora: A multipattern and multistrategy discovery assistant. *Advances in knowledge discovery and data mining*, 1996.

[Kuznetsov, 1999] Sergei O. Kuznetsov. Learning of simple conceptual graphs from positive and negative examples. In *Principles of Data Mining and Knowledge Discovery, Third European Conference, PKDD '99, Prague, Czech Republic, September 15-18, 1999, Proceedings*, pages 384–391, 1999.

[Lavrac *et al.*, 1999] Nada Lavrac, Peter Flach, and Blaz Zupan. Rule evaluation measures: A unifying view. In *ILP-99*, pages 174–185, 1999.

[Lavrac *et al.*, 2004] Nada Lavrac, Branko Kavsek, Peter A. Flach, and Ljupco Todorovski. Subgroup discovery with CN2-SD. *JMLR*, 5:153–188, 2004.

[Leman *et al.*, 2008] Dennis Leman, Ad Feelders, and Arno J. Knobbe. Exceptional model mining. In *ECMLPKDD 2008*, pages 1–16, 2008.

[Lemmerich *et al.*, 2016] Florian Lemmerich, Martin Becker, Philipp Singer, Denis Helic, Andreas Hotho, and Markus Strohmaier. Mining subgroups with exceptional transition behavior. In *KDD*, pages 965–974, 2016.

[Leskovec and Sosič, 2014] Jure Leskovec and Rok Sosič. SNAP: A general purpose network analysis and graph mining library in C++. `http://snap.stanford.edu/snap`, June 2014.

[Li *et al.*, 2010] Zhenhui Li, Ming Ji, Jae-Gil Lee, Lu-An Tang, Yintao Yu, Jiawei Han, and Roland Kays. Movemine: Mining moving object databases. In *SIGMOD*, pages 1203–1206. ACM, 2010.

[Li *et al.*, 2012] Rui Li, Kin Hou Lei, Ravi Khadiwala, and Kevin Chen-Chuan Chang. Tedas: A twitter-based event detection and analysis system. In *ICDE'12*, pages 1273–1276, 2012.

[Lijffijt *et al.*, 2016] Jefrey Lijffijt, Eirini Spyropoulou, Bo Kang, and Tijl De Bie. P-n-rminer: a generic framework for mining interesting structured relational patterns. *I. J. Data Science and Analytics*, 1(1):61–76, 2016.

[Lipton, 2016] Zachary Chase Lipton. The mythos of model interpretability. *CoRR*, abs/1606.03490, 2016.

[Liu and Wong, 2008] Guimei Liu and Limsoon Wong. Effective pruning techniques for mining quasi-cliques. In *ECML/PKDD*, pages 33–49, 2008.

[Liu *et al.*, 1998] Bing Liu, Wynne Hsu, and Yiming Ma. Integrating classification and association rule mining. In *KDD*, pages 80–86, 1998.

[Low-Kam *et al.*, 2013] Cécile Low-Kam, Chedy Raïssi, Mehdi Kaytoue, and Jian Pei. Mining statistically significant sequential patterns. In *2013 IEEE 13th International Conference on Data Mining, Dallas, TX, USA, December 7-10, 2013*, pages 488–497, 2013.

[Luo *et al.*, 2013] Wuman Luo, Haoyu Tan, Lei Chen, and Lionel M. Ni. Finding time period-based most frequent path in big trajectory data. In *SIGMOD*, pages 713–724. ACM, 2013.

[Makino and Uno, 2004] Kazuhisa Makino and Takeaki Uno. New algorithms for enumerating all maximal cliques. In *SWAT*, pages 260–272, 2004.

[Masseglia *et al.*, 1998] Florent Masseglia, Fabienne Cathala, and Pascal Poncelet. The PSP approach for mining sequential patterns. In *Principles of Data Mining and Knowledge Discovery, Second European Symposium, PKDD '98, Nantes, France, September 23-26, 1998, Proceedings*, pages 176–184, 1998.

[Masucci *et al.*, 2013] A Paolo Masucci, Joan Serras, Anders Johansson, and Michael Batty. Gravity versus radiation models. *Physical Review E*, 88(2):022812, 2013.

[Michaelis *et al.*, 2016] Stefan Michaelis, Nico Piatkowski, and Marco Stolpe, editors. *Solving Large Scale Learning Tasks. Challenges and Algorithms - Essays Dedicated to Katharina Morik on the Occasion of Her 60th Birthday*, volume 9580 of *Lecture Notes in Computer Science*. Springer, 2016.

[Monreale *et al.*, 2009] Anna Monreale, Fabio Pinelli, Roberto Trasarti, and Fosca Giannotti. Wherenext: A location predictor on trajectory pattern mining. In *KDD*, pages 637–646. ACM, 2009.

[Moranges *et al.*, 2018] Maëlle Moranges, Marc Plantevit, Arnaud Fournel, Moustafa Bensafi, and Céline Robardet. Exceptional attributed subgraph mining to understand the olfactory percept. In *Discovery Science*, pages 1–15. Springer, 2018.

[Moser *et al.*, 2009] Flavia Moser, Recep Colak, Arash Rafiey, and Martin Ester. Mining cohesive patterns from graphs with feature vectors. In *SIAM SDM*, pages 593–604, 2009.

[Mougel *et al.*, 2012] Pierre-Nicolas Mougel, Christophe Rigotti, and Olivier Gandrillon. Finding collections of k-clique percolated components in attributed graphs. In *PAKDD*, 2012.

[Mougel *et al.*, 2014] Pierre-Nicolas Mougel, Christophe Rigotti, Marc Plantevit, and Olivier Gandrillon. Finding maximal homogeneous clique sets. *Knowl. Inf. Syst.*, 39(3):579–608, 2014.

[Négrevergne and Guns, 2015] Benjamin Négrevergne and Tias Guns. Constraint-based sequence mining using constraint programming. In *Integration of AI and OR Techniques in Constraint Programming - 12th International Conference, CPAIOR 2015, Barcelona, Spain, May 18-22, 2015, Proceedings*, pages 288–305, 2015.

[Négrevergne *et al.*, 2014] Benjamin Négrevergne, Alexandre Termier, Marie-Christine Rousset, and Jean-François Méhaut. Para miner: a generic pattern mining algorithm for multi-core architectures. *Data Min. Knowl. Discov.*, 28(3):593–633, 2014.

[Ng *et al.*, 1998] Raymond T. Ng, Laks V. S. Lakshmanan, Jiawei Han, and Alex Pang. Exploratory mining and pruning optimizations of constrained association rules. In Laura M. Haas and Ashutosh Tiwary, editors, *SIGMOD 1998, Proceedings ACM SIGMOD International Conference on Management of Data, June 2-4, 1998, Seattle, Washington, USA.*, pages 13–24. ACM Press, 1998.

[Nguyen *et al.*, 2011] Kim-Ngan Nguyen, Loïc Cerf, Marc Plantevit, and Jean-François Boulicaut. Multidimensional association rules in boolean tensors. In *Proceedings of the Eleventh SIAM International Conference on Data Mining, SDM 2011, April 28-30, 2011, Mesa, Arizona, USA*, pages 570–581, 2011.

[Nguyen *et al.*, 2013] Kim-Ngan Nguyen, Loïc Cerf, Marc Plantevit, and Jean-François Boulicaut. Discovering descriptive rules in relational dynamic graphs. *Intell. Data Anal.*, 17(1):49–69, 2013.

[Nguyen *et al.*, 2016] Anh Mai Nguyen, Jason Yosinski, and Jeff Clune. Multifaceted feature visualization: Uncovering the different types of features learned by each neuron in deep neural networks. *CoRR*, abs/1602.03616, 2016.

[Nijssen and Kok, 2004] Siegfried Nijssen and Joost N. Kok. Frequent graph mining and its application to molecular databases. In *Systems, Man and Cybernetics (SMC)*, volume 5, pages 4571–4577, 2004.

[Novak *et al.*, 2009] Petra Kralj Novak, Nada Lavrač, and Geoffrey I. Webb. Supervised descriptive rule discovery: A unifying survey of contrast set, emerging pattern and subgroup mining. *J. Mach. Learn. Res.*, 10:377–403, June 2009.

[Page *et al.*, 1998] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. In *WWW*, pages 161–172, 1998.

[Papadopoulos *et al.*, 2014] S. Papadopoulos, D. Corney, and L. Maria Aiello, editors. *SNOW Data Challenge*, volume 1150, 2014.

[Pearl, 2009] Judea Pearl. *Causality*. Cambridge university press, 2009.

[Pearson, 1900] Karl Pearson. On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 50(302):157–175, 1900.

[Pei *et al.*, 2004] Jian Pei, Jiawei Han, and Laks V. S. Lakshmanan. Pushing convertible constraints in frequent itemset mining. *Data Min. Knowl. Discov.*, 8(3):227–252, 2004.

[Pérez-Melián *et al.*, 2017] José Alberto Pérez-Melián, J. Alberto Conejero, and Cèsar Ferri Ramirez. Zipf's and benford's laws in twitter hashtags. In *EACL*, pages 84–93, 2017.

[Petitjean *et al.*, 2016] François Petitjean, Tao Li, Nikolaj Tatti, and Geoffrey I. Webb. Skopus: Mining top-k sequential patterns under leverage. *Data Min. Knowl. Discov.*, 30(5):1086–1111, 2016.

[Plantevit and Crémilleux, 2009] Marc Plantevit and Bruno Crémilleux. Condensed representation of sequential patterns according to frequency-based measures. In *Adv. in Intelligent Data Analysis*, LNCS (5772), pages 155–166. Springer, 2009.

[Plantevit *et al.*, 2010] Marc Plantevit, Anne Laurent, Dominique Laurent, Maguelonne Teisseire, and Yeow Wei Choong. Mining multidimensional and multilevel sequential patterns. *TKDD*, 4(1):4:1–4:37, 2010.

[Prado *et al.*, 2013] Adriana Prado, Marc Plantevit, Céline Robardet, and Jean-François Boulicaut. Mining graph topological patterns. *IEEE Trans. Knowl. Data Eng.*, 25(9):2090–2104, 2013.

[Raedt and Zimmermann, 2007] Luc De Raedt and Albrecht Zimmermann. Constraint-based pattern set mining. In *Proceedings of the Seventh SIAM International Conference on Data Mining, April 26-28, 2007, Minneapolis, Minnesota, USA*, pages 237–248, 2007.

[Raïssi and Plantevit, 2008] Chedy Raïssi and Marc Plantevit. Mining multidimensional sequential patterns over data streams. In *Data Warehousing and Knowledge Discovery, 10th International Conference, DaWaK 2008, Turin, Italy, September 2-5, 2008, Proceedings*, pages 263–272, 2008.

[Robardet, 2009] Céline Robardet. Constraint-Based Pattern Mining in Dynamic Graphs. In *ICDM*, pages 950–955. IEEE, 2009.

[Rueping, 2009] Stefan Rueping. Ranking interesting subgroups. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 913–920. ACM, 2009.

[Sarzynska *et al.*, 2016] Marta Sarzynska, Elizabeth Leicht, Gerardo Chowell, and Mason Porter. Null models for community detection in spatially embedded, temporal networks. *J. Complex Networks*, 4(3):363–406, 2016.

[Silva *et al.*, 2012] Arlei Silva, Wagner Meira., and Mohammed Zaki. Mining attribute-structure correlated patterns in large attributed graphs. *PVLDB*, 5(5):466–477, 2012.

[Simini *et al.*, 2011] Filippo Simini, Marta C González, Amos Maritan, and Albert-László Barabási. A universal model for mobility and migration patterns. *arXiv preprint arXiv:1111.0586*, 2011.

[Soulet and Crémilleux, 2009] Arnaud Soulet and Bruno Crémilleux. Mining constraint-based patterns using automatic relaxation. *Intell. Data Anal.*, 13(1):109–133, 2009.

[Soulet *et al.*, 2011] Arnaud Soulet, Chedy Raïssi, Marc Plantevit, and Bruno Crémilleux. Mining dominant patterns in the sky. In *11th IEEE International Conference on Data Mining, ICDM 2011, Vancouver, BC, Canada, December 11-14, 2011*, pages 655–664, 2011.

[Srikant and Agrawal, 1996] Ramakrishnan Srikant and Rakesh Agrawal. Mining sequential patterns: Generalizations and performance improvements. In *EDBT*, pages 3–17, 1996.

[Terada *et al.*, 2013] Aika Terada, Mariko Okada-Hatakeyama, Koji Tsuda, and Jun Sese. Statistical significance of combinatorial regulations. *Proceedings of the National Academy of Sciences*, 110(32):12996–13001, 2013.

[Ugarte *et al.*, 2017] Willy Ugarte, Patrice Boizumault, Bruno Crémilleux, Alban Lepailleur, Samir Loudni, Marc Plantevit, Chedy Raïssi, and Arnaud Soulet. Skypattern mining: From pattern condensed representations to dynamic constraint satisfaction problems. *Artif. Intell.*, 244:48–69, 2017.

[Uno, 2007] Takeaki Uno. An efficient algorithm for enumerating pseudo cliques. In *ISAAC 2007*, pages 402–414, 2007.

[Uno, 2010] Takeaki Uno. An efficient algorithm for solving pseudo clique enumeration problem. *Algorithmica*, 56(1):3–16, 2010.

[van Leeuwen, 2010] Matthijs van Leeuwen. Maximal exceptions with minimal descriptions. *Data Mining Knowledge Discovery*, 21(2):259–276, 2010.

[van Leeuwen, 2014] Matthijs van Leeuwen. Interactive data exploration using pattern mining. In *Interactive Knowledge Discovery and Data Mining in Biomedical Informatics*, pages 169–182. Springer, 2014.

[Vreeken *et al.*, 2011] Jilles Vreeken, Matthijs van Leeuwen, and Arno Siebes. Krimp: mining itemsets that compress. *Data Min. Knowl. Discov.*, 23(1):169–214, 2011.

[Wang *et al.*, 2007] Jianyong Wang, Jiawei Han, and Chun Li. Frequent closed sequence mining without candidate maintenance. *IEEE Trans. Knowl. Data Eng.*, 19(8):1042–1056, 2007.

[Wang *et al.*, 2011] Dashun Wang, Dino Pedreschi, Chaoming Song, Fosca Giannotti, and Albert-Laszlo Barabasi. Human mobility, social ties, and link prediction. In *KDD*, pages 1100–1108. ACM, 2011.

[Wang *et al.*, 2012] Dan J. Wang, Xiaolin Shi, Daniel A. McFarland, and Jure Leskovec. Measurement error in network data: A re-classification. *Social Networks*, 34(4):396 – 409, 2012.

[Wang *et al.*, 2013] Jia Wang, James Cheng, and Ada Wai-Chee Fu. Redundancy-aware maximal cliques. In *ACM SIGKDD 2013*, pages 122–130, 2013.

[Webb and Petitjean, 2016] Geoffrey I. Webb and François Petitjean. A multiple test correction for streams and cascades of statistical hypothesis tests. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 1255–1264, 2016.

[Weng and Lee, 2011] J. Weng and B.-S. Lee. Event detection in twitter. In *ICWSM*, 2011.

[Wrobel, 1997] Stefan Wrobel. An algorithm for multi-relational discovery of subgroups. In *PKDD 1997*, pages 78–87, 1997.

[Xiao *et al.*, 2016] H. Xiao, P. Rozenshtein, and A. Gionis. Discovering topically and temporally coherent events in interaction networks. In *ECMLPKDD*, 2016.

[Yan and Han, 2002] Xifeng Yan and Jiawei Han. gSpan: Graph-Based Substructure Pattern Mining. In *Int. Conf. on Data Mining (ICDM)*, pages 721–724, 2002.

[Yan *et al.*, 2003] Xifeng Yan, Jiawei Han, and Ramin Afshar. Clospan: Mining closed sequential patterns in large databases. In *SDM*, pages 166–177. SIAM, 2003.

[Zaki and Hsiao, 2002] Mohammed Javeed Zaki and Ching-Jiu Hsiao. Charm: An efficient algorithm for closed itemset mining. In *SDM*. SIAM, 2002.

[Zaki, 2000] Mohammed Javeed Zaki. Scalable algorithms for association mining. *IEEE Trans. Knowl. Data Eng.*, 12(3):372–390, 2000.

[Zaki, 2001] Mohammed Javeed Zaki. SPADE: an efficient algorithm for mining frequent sequences. *Machine Learning*, 42(1/2):31–60, 2001.

[Zeiler and Fergus, 2014] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I*, pages 818–833, 2014.

[Zhang *et al.*, 2016] Chao Zhang, Guangyu Zhou, Quan Yuan, Honglei Zhuang, Yu Zheng, Lance M. Kaplan, Shaowen Wang, and Jiawei Han. Geoburst: Real-time local event detection in geo-tagged tweet streams. In *ACM SIGIR*, pages 513–522, 2016.

[Zheng *et al.*, 2009] Yu Zheng, Lizhu Zhang, Xing Xie, and Wei-Ying Ma. Mining interesting locations and travel sequences from gps trajectories. In *WWW*, pages 791–800. ACM, 2009.

[Zipf, 1946] George Kingsley Zipf. The p 1 p 2/d hypothesis: The case of railway express. *The Journal of Psychology*, 22(1):3–8, 1946.

Résumé

Dans cette Habilitation à Diriger des Recherches, je présente les principaux résultats auxquels j'ai contribué dans le domaine de la fouille de motifs dans les graphes augmentés.

Les graphes sont un puissant outil mathématique permettant de modéliser de nombreux phénomènes réels où les entités sont décrites par des sommets et les relations entre elles via des arêtes. Ces graphes sont bien souvent augmentés par des informations décrivant plus précisément l'activité d'un sommet ou le contexte de l'interaction (l'arête). On parle alors de graphes attribués. Les graphes peuvent également être dynamiques, la structure et/ou les valeurs des attributs évoluant au cours du temps. La découverte de motifs dans de tels graphes permet de fournir à l'utilisateurs des informations exploitables et d'enrichir ses connaissances.

Cette thèse se décompose en deux parties. Dans la première partie, je présente les différents domaines de motifs pour les graphs augmentés que j'ai proposés. Cela inclut la découverte de motifs de co-évolution dans des graphes attribués dynamiques, l'étude de liens entre la structure du graphes et les attributs des sommets et la découverte de sous-graphes attribués exceptionnels. J'introduis d'abord les motifs de co-évolution pour l'analyse de graphes dynamiques attribués ainsi que des mesures d'intérêt visant à évaluer un motif par rapport à chacune des dimensions du graphe (i.e., le temps, les attributs et les sommets). Des exemples de motifs découverts dans des données spatio-temporelles sont retournés. Je présente ensuite deux domaines de motifs pour analyser les liens entre la structure du graphe et les attributs propres aux sommets. Ces deux types de motifs sont illustrés au travers de l'analyse de réseaux de co-auteurs issus de DBLP. Cette première partie se termine par la découverte de sous-graphes attribués. Les méthodes proposées sont formalisées dans le cadre de la découverte de sous-groupes et de découverte de modèles exceptionnels.

Dans la seconde partie, j'étends ces domaines de motifs pour extraire des motifs plus intéressants en prenant en compte les connaissances du domaine, les retours de l'utilisateur ainsi que ses connaissances a priori. Je présente une méthode s'appuyant sur des modèles de mobilités issus de la physique statistique pour évaluer le caractère inattendu des trajectoires dans graphes de mobilité. Cela permet de prendre en compte des informations spatiales (i.e., distances entre les sommets, population d'un sommet) pour découvrir des sous-graphes réellement exceptionnels par rapport à ces informations et d'éliminer certaines trajectoires qui deviennent attendues dès lors que l'on connait ces informations spatiales. Je présente ensuite une méthode qui vise à prendre en compte les retours de l'utilisateur dans une mesure de qualité biaisée lors de processus interactif de découvertes de motifs. Cette méthode est définie dans le contexte de l'analyse de medias sociaux, plus particulièrement la détection d'événements géo-localisés. S'appuyant sur les retours de l'utilisateur, les termes ou les zones aimés sont favorisés grâce à une mesure de qualité biaisée. Enfin, je considère le problème de l'intérêt subjectif dans les graphes attribués afin de prendre en compte les apriori de l'utilisateur. Dans ce contexte, un intérêt particulier est donné à l'assimilation des motifs par l'utilisateurs. Afin de facilité cette assimilation, des descriptions alternatives – plus facilement interprétables – des sous-graphes exceptionnels sont construites.

Finalement, je conclus le manuscrit en discutant des perspectives de recherche.

Mots-clés: Graphes attribués, graphes dynamiques, fouille de motifs interactive, découverte de sous-groupes, modèles exceptionnels, intérêt subjectif.

Abstract

In this Habilitation à Diriger des Recherches thesis, I present the main results I have contributed to in the field of pattern mining in augmented graphs.

Graphs are a powerful mathematical abstraction that enables to depict many real world phenomena. Vertices describe entities and edges identify relations between entities. Such graphs are often augmented with additional pieces of information. For instance, the vertices or the edges are enriched with attributes describing them and are called vertex (respectively edge) attributed graphs. Graphs can also be dynamic, i.e., the structure and the values of vertex attributes may evolve through time. The discovery of patterns in such graphs may provide actionable insights and boost the user knowledge.

This manuscript is structured in two parts. In the first part, I discuss the different pattern domains for augmented graphs I contributed to define. This includes the discovery of co-evolution patterns in dynamic attributed graphs, the study of links between the graph structure and the vertex attributes and the discovery of exceptional attributed subgraphs. I first introduce the co-evolution patterns to analyze dynamic attributed graphs as well as interestingness measures to assess these patterns according to each dimension of the graphs (i.e., the dynamics, the vertex attributes organized within a hierarchy or not, and the vertices). Examples of co-evolution patterns in spatio-temporal data are reported. I then present two pattern domains to analyze the links between the vertex attributes and the graph structure. These two types of patterns are illustrated on co-authorship network built from DBLP. This part ends with the discovery of exceptional attributed subgraphs in edge or vertex attributed graphs. The proposed methods are rooted in Subgroup Discovery / Exceptional Model Mining.

In the second part, I discuss how to find pattern of higher interest by taking into account the domain knowledge, user feedback and user's prior knowledge through different contributions. I first present a method that borrows mobility models from statistical physics to assess the unexpectedness of some trajectories in mobility network. This allows the discovery of exceptional attributed subgraphs by taking into account spatial information (i.e., distance between vertices, importance of the vertices in term of population). I then present a method to take the user feedback into biased quality measures in interactive explorations. This method is defined in the context of social media analysis, especially geo-located event detection. Based on user feedback, liked tags and areas are fostered thanks to a biased quality measure. Eventually, I address the problem of subjective interestingness in attributed graphs to take into account the user's prior knowledge. A particular interest is given to the assimilation of the patterns by the user. To ease this assimilation, alternative descriptions of exceptional attributed subgraphs are provided.

Finally, I conclude this thesis by discussing some research perspectives.

Keywords: Dynamic graphs, attributed graphs, interactive pattern mining, subgroup discovery, exceptional model mining, subjective interestingness