Exceptional contextual subgraph mining

Mehdi Kaytoue, Marc Plantevit, Albrecht Zimmermann, Anes Bendimerad, Céline Robardet

Exceptional Contextual Subgraph Mining

Mehdi Kaytoue Marc Plantevit Albrecht Zimmermann
Anes Bendimerad Céline Robardet

Mehdi Kaytoue, Albrecht Zimmermann, Anes Bendimerad, Céline Robardet: INSA de Lyon, CNRS, LIRIS UMR5205, F-69621 France and Marc Plantevit: Université Lyon 1, CNRS, LIRIS UMR5205, F-69622 France

Many relational data result from the aggregation of several individual behaviors described by some characteristics. For instance, a bike-sharing system may be modeled as a graph where vertices stand for bike-share stations and connections represent bike trips made by users from one station to another. Stations and trips are described by additional information such as the description of the geographical environment of the stations (business vs. residential area, closeness to POI, elevation, urbanization density, etc.), or of the bike trips (timestamp, user profile, weather, events and other special conditions about the trip). Identifying highly connected components (such as communities or quasi-cliques) in this graph provides interesting insights into global usages but does not capture mobility profiles that characterize a subpopulation. To tackle this problem we propose an approach rooted in exceptional model mining to find exceptional contextual subgraphs, i.e., subgraphs generated from a context or a description of the individual behaviors that is exceptional (behaves in a different way) compared to the whole augmented graph. The dependency between a context and an edge is assessed by a χ^2 test and the weighted relative accuracy measure is used to only retain contexts that strongly characterize connected subgraphs. We present an original algorithm that uses sophisticated pruning techniques to restrict the search space of vertices, context refinements, and edges to be considered. An experimental evaluation on synthetic data and two real-life datasets demonstrates the effectiveness of the proposed pruning mechanisms, as well as the relevance of the discovered patterns.

1 Introduction

Providing tools and methods to discover new actionable insights into heterogeneous data is widely considered to be one of the most important challenges of data science, especially in the data mining and machine learning communities. A natural way to handle and understand such complex data is to model them as graphs, a powerful mathematical abstraction that makes it possible to support

a large variety of analyses in a generic way. This partially explains why graph mining has generated considerable interests in terms of both fundamental and applied research. A striking feature is its ability to allow better understanding of social interactions and to provide support for many tasks such as social recommendations [23], community discovery [18], social influence propagation [19], and link prediction [10].

In real-world phenomena, vertices and edges are often characterized by attributes. It is also very common that these graphs are dynamic, with vertex and edge attributes evolving through time. The design of effective graph mining methods to discover actionable insights in such graphs is therefore a current challenge, to derive new knowledge about the underlying rules that govern networks [46]. The last decade has witnessed intense growth in the analysis of dynamic graphs, especially from two main research tracks: (a) the study of the properties that describe the topology of the graph [13, 48], and (b) the extraction of specific subgraphs to describe the graph evolution [4, 42, 53]. Surprisingly, the simultaneous consideration of the dynamics of the graph structure and the additional vertex and edge properties has not been given much attention. In this paper, we move towards this new direction.

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 1(d) presents a graph of co-visitation sites built from the aggregation of bike trips of individual users (described in Table 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. 1(b) and (c) present two examples of contextual and exceptional contextual subgraphs whose description is given in the caption of the figure. Finding these kinds of relations has attracted much interest for *unstructured* data over the years, for instance finding the descriptions of users that consistently rate items in a certain way [11]. Such unstructured settings can be challenging, for instance, when describing consistent behavior with respect to several target values [12].

The considered data are made 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., 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

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

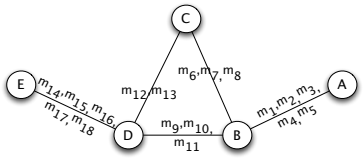
(c) Bike trip characteristics

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

(a) Bike-share stations

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



(d) Augmented graph

Table 1: Example of data: (a) Bike-share station attributes, (b) Users attributes, (c) Bike trip attributes and (d) Augmented graph to those data.

of transactions that for each edge 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 1, the proposed method identifies connected stations that are travelled in the same context. Fig. 1 (b) represents the contextual subgraph that corresponds to the stations that are 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 , 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. 1 (c) represents the subgraph of locations by young people at night whose edges have a positive $WRAcc$ value. 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 statistically assessed by a χ^2 test

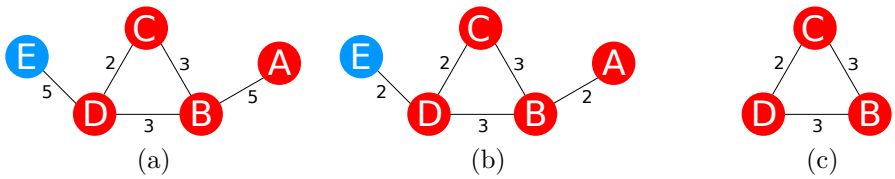


Figure 1: Example of contextual subgraph and exceptional contextual subgraph: (a) Contextual graph associated to the most general context $(\star, \star, \star, \star, \star) = (Age \in [20, 50], Gender \in \{F, M\}, Time \in \{Day, Night\}, Weather \in \{Sunny, Cloudy, Windy, Rainy\}, type\ of\ area \in \{Bars, Residential\})$; (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 $(Age \in [20, 23], \star, Time \in \{Night\}, \star, type\ of\ area \in \{Bars\})$.

Discovering *exceptional contextual subgraph* is challenging because of the size of the search space: All possible contexts and subgraphs have to be considered simultaneously. Their computation is feasible thanks to clever pruning techniques. Our contributions are:

- The definition of *exceptional contextual subgraph* patterns in dynamic attributed graphs as an instance of the EMM framework.
- The design of an efficient algorithm COSMIc that exploits several constraints, even those that are neither monotonic nor anti-monotonic, .
- A quantitative and qualitative empirical study. We report on the evaluation of the efficiency and the effectiveness of the algorithm on two real-world dynamic attributed graphs.

The rest of this paper is organized as follows. We review the related work in Section 2. We then formally define the notions of augmented graph and contextual subgraphs and introduce the *exceptional contextual subgraph* problem as an instance of EMM in Section 3. Section 4 describes an exhaustive algorithm, COSMIc, that differs from beam-search usually employed in EMM methods. We report a thorough empirical study of the algorithm COSMIc with synthetic data in Section **??** before comparing it to concurrent approaches (Section 6) and showing the usefulness of our approach with two real-world scenarios (Section 7). Section 8 concludes.

2 Related Work

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* () [28, 36]. introduced as *Exceptional Model Mining* (EMM) [29, 16]. In

this framework, in Fig. 2, 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 [29], Bayesian Networks [17], encoding based on Minimum Description Length [49]). 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.

By exploiting the connectivity of the subgraph, we are able to dynamically reduce the target search space, and propose an exact algorithm that performs successful extractions where heuristic techniques fail, as demonstrated in Section 6. To the best of our knowledge, the only EMM approach that uses exhaustive search has been proposed in [30], which adapts FP-trees to handle a number of counting-based measures for unstructured targets. Our approach can be viewed as extension of such works.

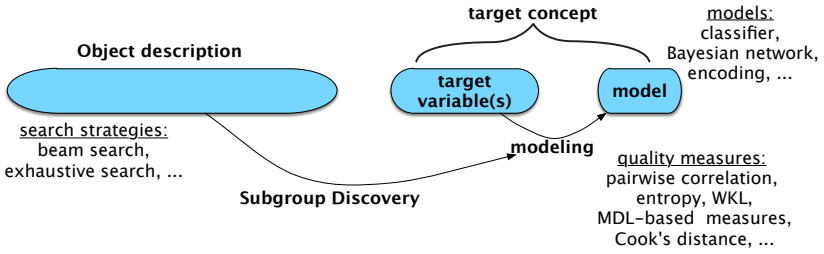


Figure 2: The Exceptional Model Mining framework (diagram from [15]).

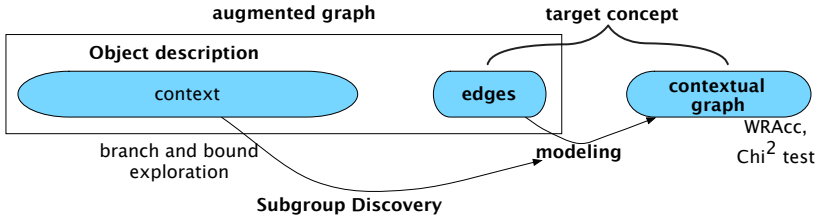


Figure 3: Exceptional contextual subgraph mining problem as an instance of EMM.

Exceptional contextual subgraph mining problem is also related to augmented graph mining, where graphs have additional information on vertices or edges. Several settings have been considered so far,

Vertex-attributed graphs In a pioneering work, Moser et al. [34] propose a method to find dense homogeneous subgraphs, i.e., subgraphs whose vertices share a large set of attributes. Similar to that work, Günnemann et al. [20]

present a method based on subspace clustering and dense subgraph mining to extract non redundant subgraphs that are homogeneous with respect to vertex attributes. Silva et al. [44] extract pairs of dense subgraphs and Boolean attribute sets such that the Boolean attributes are strongly associated with the dense subgraphs. Similarly, Mougel et al. [35] introduce the problem of mining maximal homogeneous clique sets. Khan et al. [26] design a probabilistic approach to both construct the neighborhood of a vertex and propagate information into this neighborhood. Following the same motivation, Sese et al. [43] extract (not necessarily dense) subgraph with common itemsets. Prado et al. [40] propose to mine the graph topology of a large attributed graph by finding regularities among vertex descriptors. Interestingly, in a recent work Atzmueller et al. [2] use a subgroup discovery approach to mine descriptions of communities, treating the communities as an (aggregated) target.

Existing approaches use edge information to define a similarity measure on edges in order to identify subgraphs or communities. In the proposal by Qi et al. [41], edges are considered similar according to their associated collections of labels. Similarly, Bonchi et al. [8] find clusters of edges such that edges of a cluster have the same labels. Berlingerio et al. [5] propose multidimensional network analysis, where connections between vertices belong to different *dimensions* (e.g. cities can have both train and plane connections) and extend a number of network measures to multi-dimensional graphs. In this approach, two vertices connected by edges from different dimension are considered to be more strongly connected, whereas in our exceptional contextual subgraphs framework, dimensions are not presupposed but inferred based on high weighted relative accuracy. In the *multi-layer coherent subgraph* approach , Boden et al. [7] use numerical labels on vertices to assess edges' similarity in different layers of the graph. Vertices connected by edges with similar weights induce quasi-cliques. There is again a conceptual shift with our proposal: MiMag might consider that are not very typical for a context/layer. There is also the semantic difference that all edges in a *exceptional contextual subgraph* match the associated context while being typical, whereas MiMag may group very different contexts – misled by some similar behaviors of distinct subgroups. We found evidence of both effects in experiments reported in Section 6.

Dynamic graphs Various approaches have been proposed to characterize either the graph evolution by focusing on some topological properties [48], or the graph evolution by means of patterns/rules that are much more meaningful, identifying local interpretable sub-structures of interest. Borgwardt et al. [9] introduce the problem of mining frequent subgraphs in dynamic graphs, i.e. isomorphic graphs that appear in consecutive timestamps. In [27], Lahiri and Berger-Wolf also extract frequent subgraphs but at periodic or near-periodic timestamps. Inokuchi and Washio [22] define frequent induced subgraph subsequences, i.e. subgraph sub-sequences whose isomorphic occurrences appear frequently in a graph sequence collection. Prado et al. [39] extract spatio-

temporal patterns in a sequence of planar graphs. Robardet [42] proposes an algorithm to extract evolving patterns, i.e. pseudo-cliques which appear in consecutive timestamps with slight evolutions. Ahmed and Karypis [1] mine the evolution of conserved relational states, i.e. sequences of time-conserved patterns on consecutive time. Yang et al. [52] devise an algorithm to identify the most frequently changing component. You and Cook [53] compute graph rewriting rules that describe the evolution between consecutive graphs. These rules are then abstracted into patterns representing the dynamics of graphs. Berlingerio et al. [4] extract patterns based on frequency and derive evolution rules to solve prediction problems in [10]. All these works only focus on the graph structure and do not consider attributes related to the vertices and/or the edges.

Dynamic attributed graphs In [14], Desmier et al. define a new pattern domain that relies on the graph structure and the temporal evolution of the attribute values. It makes it possible to discover subgraphs of small diameter whose vertex attributes follow the same trends. Kaytoue et al. [24] devise an algorithm to characterize local structure changes in a sequence of vertex-attributes trends. While considering attributes on vertices and edges, exceptional contextual subgraphs also offer the opportunity to analyse the dynamics of relational data, when transactions associated to edges are timestamped.

3 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.

3.1 Deriving Contextual Subgraphs

3.1.1 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, Table 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 1(a)). The connections, i.e. bike trips, are timestamped and can be augmented with the profile of the user, weather, events and other special

conditions about the trip (see Tables 1(b) and (c)). This rich dataset is a multigraph, which can be viewed as a transactional database anchored to a graph (). In other words, each connection is recorded as a transaction containing attributes (the join of tables 1(a), (b) and (c)) and associated to a source and a target entity that form the directed edge[1] along which the connection occurred. This type of data is called *augmented graph* and is formally defined below.

Definition 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 $\boldsymbol{dom}(R_i)$ 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 $t_i \in \boldsymbol{dom}(R_i)$. 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 1(d) illustrates the augmented graph that corresponds to the data of Table 1. On such data, we aim to identify subgraphs that are typical for a context, as the one of Fig. 1 (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 [16, 29] (see Fig. 2).

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. 3, 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.

3.1.2 Contextual Graphs

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

Definition 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 $a \in \boldsymbol{dom}(R_i)$, iff R_i is nominal and $\forall (t_1, \cdots, t_i, \cdots, t_p) \in S$, $t_i = a$*

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

- $C_i = \star_i$, *with* $\star_i$ *a new symbol representing the whole set* $\boldsymbol{dom}(R_i)$, *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* $a \in \boldsymbol{dom}(R_i)$ *and* R_i *nominal*
- t_i *is any of* $\boldsymbol{dom}(R_i)$, *with* $C_i = \star_i$ *and* R_i *nominal*
- $a \leq t_i \leq b$, *with* $C_i = [a, b]$ *and* R_i *numerical.*

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 3 (Contextual Subgraph) *Given an augmented graph* $G = (V, E, T, \textsc{Edge})$ *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. 1(b) shows the contextual subgraph of the context $(Age \in [20, 23], \star, Time \in \{Night\}, \star, \star)$.

3.1.3 Closed Contexts

Definition 4 (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* $C^1_i = C^2_i = a \in \boldsymbol{dom}(R_i)$, *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* $C^2_i = [a^2_i, b^2_i]$, .

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

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 5 (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.*

3.2 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 we propose to use two evaluation measures: The Pearson's chi-squared test of independence [38] and the Weighted Relative Accuracy measure.

3.2.1 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 [38]. 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 $\overline{\mathbf{e}}$. Table 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 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.

3.2.2 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 . 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

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

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

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

(b) Contingency table E under the null hypothesis.

Table 2: Contingency tables O and E.

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 $\text{WRAcc}(C, e) > 0$.

Definition 6 (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

$$\begin{aligned} \textsc{Except}(C, e) \quad & \equiv & (1)\\ & \textit{and} & (2)\\ & \textit{and} \quad X^2(C, e) > & (3)\\ & \textit{and} \quad \text{WRAcc}(C, e) > 0 & (4) \end{aligned}$$

3.3 Deriving Exceptional Contextual Connected Components

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.

4 Algorithm

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[2], 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, as explained in the following.

4.1 COSMIc principle

COSMIc enumerates contexts in a depth-first search manner. Its pseudo-code is given in Algorithm 1. 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 (line 2), the connected components of G_C are considered (line 3) and the function `CheckConstraints` (line 4) is called: It returns `true` iff (C, CC_C) satisfies all the constraints of Definition 6 and Problem **??**. In that case, the pattern is output (line 5).

If the attribute R_i can still be specialized in the context (lines 6 to 31), a new context C' is generated: If R_i is symbolic, a loop over the values of $\mathbf{dom}(R_i) \cup \star_i$ (line 8) lists all the possible specializations C' of C on R_i (line 9). Then, the transactions of G_C that do not satisfy C' are removed (line 10). The closure F of C' is computed line 11. If C' is closed (line 12), the function `Pruning`

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

(detailed in the next subsection) is called (line 13) to prune all the edges and connected components that are guaranteed to not satisfy the constraints for any contexts that are specializations of C'. If $G_{C'}$ is not empty (line 14), $(C', G_{C'})$ is recursively enumerated (line 15) to generate all valid *exceptional contextual subgraph* patterns.

From lines 16 to 31, we consider the case where 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 (see the test line 2, with variable `left` retrieved from the stack in line 19). The generated intervals are pushed onto the stack (lines 29 and 31) and the loop from lines 18 to 31 is reiterated until the last interval has been considered.

For each interval, a new context C' is generated (line 20) and, as for nominal attributes, the transactions of G_C that do not satisfy C' are removed (line 21). The closure F of C' is computed (line 22). If C' is closed, the function `Pruning` –detailed in the next subsection– is called (line 24) and $(C', G_{C'})$ is recursively enumerated (line 26).

This algorithm explores the lattice of symbolic concepts to find the closed ones, and therefore benefits from the developments and optimizations that have been published in the data mining literature for that problem setting. Given that we search *strict* closed contexts, the algorithm risks running into the same issues in the presence of noise that existing such algorithms exhibit. Extending the algorithm with the capability to mine noise-tolerant contexts [6] remains for future work.

4.2 The `Pruning` function

The `Pruning` function, see Algorithm 2, is based on two pruning mechanisms. The first one (lines 3 to 5) consists of removing individual edges. The constraint $|M_{C\to T}(C, M_{G\to T}(e))| > min\_weight$ (constraint (2) in Definition 6) 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 $X^2_{ub}(C, e)$, presented below, 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 (lines 6 to 7) focuses on the connected components CC of G_C. Constraints (6) and (7) of Problem **??** 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 (see Algorithm 3 in subsection 4.2.3) a tight upper bound with pruning capabilities without loss of promising patterns.

4.2.1 Upper bound for $X^2(C,e)$

Let us denote by $y = W_C(e)$, $x = \Sigma_{z\in E} W_C(z)$, $\alpha = \Sigma_{z\in E} W_\star(z)$ and $\beta = W_\star(e)$. Since α and β are independent of W_C, the values of x and y uniquely determine $X^2(C,e)$ and we have

$$\begin{aligned} X^2(x,y) &= \frac{(y - x\frac{\beta}{\alpha})^2}{\frac{\beta x}{\alpha}} + \frac{((x-y) - x\frac{\alpha-\beta}{\alpha})^2}{x\frac{\alpha-\beta}{\alpha}} + \frac{\left((\beta-y) - (\alpha-x)\frac{\beta}{\alpha}\right)^2}{(\alpha-x)\frac{\beta}{\alpha}} \\ &+ \frac{((\alpha-\beta-x+y) - (\alpha-x)\frac{\alpha-\beta}{\alpha})^2}{(\alpha-x)\frac{\alpha-\beta}{\alpha}} \\ &= \frac{(\alpha y - \beta x)^2}{\beta(\alpha-\beta)} . \frac{\alpha}{x(\alpha-x)} \end{aligned}$$

$X^2(x,y)$ is a convex function and, as shown by Morishita and Sese [33], takes its maximum values at the extrema: $(x-y,0)$, (y,y). Since the former is equivalent to an edge with weight$=0$, i.e. an edge that violates constraint (2), we only consider the latter tuple for the upper bound, that is to say

$$X^2(C,e) \leq X^2_{ub}(C,e) = \frac{(\alpha y - \beta y)^2}{\beta(\alpha-\beta)} \times \frac{\alpha}{y(\alpha-y)}$$

When $X^2_{ub}(C,e) < \chi^2_{0.05}$, the edge can never satisfy the constraint.

4.2.2 Upper bound for WRAcc(C,e)

Similarly, let us denote by $y = W_C(e)$, $x = \max_{z\in E_C} W_C(z)$, $\alpha = \max_{z\in E_C} W_\star(z)$ and $\beta = W_\star(e)$. Since α and β are independent of W_C, the values of x and y uniquely determine the WRAcc(C,e) value and we have $\text{WRAcc}(x,y) = \frac{x}{\alpha}\left(\frac{y}{x} - \frac{\beta}{\alpha}\right)$.

Property 1 *The function* WRAcc(x,y) *is a convex function.*

Proof 1 *For* $0 \leq \lambda \leq 1$*, we have*

$$\begin{aligned} \text{WRAcc}(\lambda(x_1,y_1) + (1-\lambda)(x_2,y_2)) &= \frac{\lambda x_1 + (1-\lambda)x_2}{\alpha}\left(\frac{\lambda y_1 + (1-\lambda)y_2}{\lambda x_1 + (1-\lambda)x_2} - \frac{\beta}{\alpha}\right) \\ &= \frac{\lambda y_1}{\alpha} + \frac{(1-\lambda)y_2}{\alpha} - \frac{\beta\lambda x_1}{\alpha^2} - \frac{\beta(1-\lambda)x_2}{\alpha^2} \\ &= \lambda\left(\frac{y_1}{\alpha} - \frac{\beta x_1}{\alpha^2}\right) + (1-\lambda)\left(\frac{y_2}{\alpha} - \frac{\beta x_2}{\alpha^2}\right) \\ &= \lambda\left(\frac{x_1}{\alpha}\left(\frac{y_1}{x_1} - \frac{\beta}{\alpha}\right)\right) + (1-\lambda)\left(\frac{x_2}{\alpha}\left(\frac{y_2}{x_2} - \frac{\beta}{\alpha}\right)\right) \\ &= \lambda\text{WRAcc}(x_1,y_1) + (1-\lambda)\text{WRAcc}(x_2,y_2) \end{aligned}$$

Following the results of [33] and the argumentation of above, this function takes its maximum value at (y, y), that is to say

$$\text{WRAcc}(C, e) \leq \text{WRAcc}_{ub}(C, e) = \frac{W_C(e)}{\max_{x \in E_C} W_\star(x)} \times \left(1 - \frac{W_\star(e)}{\max_{x \in E_C} W_\star(x)}\right)$$

4.2.3 Upper bound for

When specializing a context C, the connected components of the associated graph may expand or shrink, some edges increasing their value on X^2 above $\chi^2_{0.05}$, or decreasing it below $\chi^2_{0.05}$. The set of edges thus does not satisfy a monotonic property. To upper bound $\sum \text{WRAcc}$ on each connected component, we only consider the terms that depend on W_C:

$$\begin{aligned}
&\sum \text{WRAcc}(C, (V_{CC}, E_{CC})) = \sum_{e \in E_{CC}} \text{WRAcc}(C, e) \\
= &\sum_{e \in E_{CC}} \frac{\max_{x \in E_{CC}} W_C(x)}{\max_{x \in E} W_\star(x)} \times \left(\frac{W_C(e)}{\max_{x \in E_{CC}} W_C(x)} - \frac{W_\star(e)}{\max_{x \in E} W_\star(x)}\right) \\
= &\frac{\max_{x \in E_{CC}} W_C(x)}{\max_{x \in E} W_\star(x)} \sum_{e \in E_{CC}} \left(\frac{W_C(e)}{\max_{x \in E_{CC}} W_C(x)} - \frac{W_\star(e)}{\max_{x \in E} W_\star(x)}\right) \\
= &\frac{\max_{x \in E_{CC}} W_C(x)}{\max_{x \in E} W_\star(x)} \left(\frac{\sum_{e \in E_{CC}} W_C(e)}{\max_{x \in E_{CC}} W_C(x)} - \frac{\sum_{e \in E_{CC}} W_\star(e)}{\max_{x \in E} W_\star(x)}\right) \\
= &\frac{\max_{x \in E_{CC}} W_C(x)}{\alpha} \left(\frac{\sum_{e \in E_{CC}} W_C(e)}{\max_{x \in E_{CC}} W_C(x)} - \gamma\right)
\end{aligned}$$

Let us order the edges of E_{CC} in descending order of their weights: $W_C(e_1) \geq \ldots \geq W_C(e_k)$. While specializing C into C', clearly, a more constrained context is satisfied only by a subset of the transactions. To upper bound the measure, we search for the weight combination that maximizes

$$\sum \text{WRAcc}(C, (V_C, E_C)) = \sum \text{WRAcc}(C, (V_C, \{W_C(e_1) \ldots W_C(e_k)\}))$$

To this end, we successively replace $\max_{x \in E}$ in the formula above, starting with e_1, i.e. the observed set of weights. Next, we evaluate the configuration with $W_C(e_2)$ as maximum weight, i.e.

$$\sum \text{WRAcc}(C, (V_C, \{W_C(e_2) W_C(e_2) W_C(e_3) \ldots W_C(e_k)\})),$$

and continue until the value of $\sum \text{WRAcc}$ decreases. Indeed, as the sum of convex functions is also convex, we are sure to have reached the maximum value of $\sum \text{WRAcc}$. Once the edges are sorted, this computation can be done in linear time as explained in Algorithm 3. At each iteration, $W_C(e_j)$ is considered to be the maximum edge weight. The variable S_1 stores the sum of the weights for

the edges that are before e_j with a weight equal to $W_C(e_j)$ for each of them. S_2 is the sum for the remaining edge weights, which remain unchanged. $S_1 + S_2$ corresponds to the sum of weights for the current combination of edge weights and is used to evaluate $\sum \text{WRAcc}_{ub}$.

There is always the risk that weakly expressed edges mean that upper bounds are far too optimistic and do not aid in pruning. We empirically assess this issue, as well as the effect of mining only closed patterns, in Section 5.3.

4.3 Discussion

COSMIc performs a complete and non redundant enumeration of all useful contexts, that is, contexts for which one or several interesting patterns can be found. It is complete as it enumerates all contexts and as it uses safe pruning (c.f. upper bounds and a threshold on an anti-monotonic constraint on the minimum weight of an edge). By definition, closed contexts cannot share exactly the same set of transactions, hence redundant. This is a known result in the pattern mining literature, hence we omit a proof here.

Each closed context thus induces a different of set of transactions (). . Redundant patterns are filtered out at the end of the algorithm. In practice however, we rarely observed such a situation in our experiments.

To identify interesting patterns without requiring the end-user to set thresholds (on minimum number of vertices, weights, ...), we can slightly adapt COSMIc to output the best patterns either w.r.t a single measure (top-k) or several measures and associated user preferences. For example, the analyst could be interested in patterns maximizing the number of edges while minimizing the edge average WRAcc and number of vertices, that is, densely connected graphs with a high average quality measure. For that matter, we can simply keep in memory patterns that are not dominated by others while enumerating them. In other words, we incrementally build the Pareto front given a set of user preferences (so called skypatterns [45]).

5

5.1 Artificial augmented graph generator

Since it is notoriously difficult to obtain data of which the ground truth is known, especially for augmented graphs that have not been much studied so fare, we designed an artificial augmented graph generator that makes possible to evaluate COSMIc in a systematic way.

The generator works as follows. First, it generates the graph structure with $nbVertices$ vertices, where each pair of vertices has the same probability $linkProb$ to be linked by an edge. Then, it generates $nbPatterns$ contexts and assigns to each of them a distinct connected subgraph with $patternSize$ edges. Then, the transactions are generated: We assign on average $weight$ transactions that satisfy the context associated to that pattern. Finally, noise

is added to these transactions, governed by *noiseRate*. The parameters used for generating data are summarized in Table 3.

5.2 Robustness to noise and ability to discover hidden patterns

In this section, we evaluate the ability of COSMIc to recover contexts (and their corresponding connected components) that have been hidden in an augmented graph thanks to our artificial data generator. To assess the quality of a discovered pattern $P_d = (C_d, (V_d, E_d))$, we compare it to each hidden pattern $P_h = (C_h, (V_h, E_h))$ based on the scores S_V and S_C:

- S_V indicates the similarity between the vertices of the two patterns:

$$S_V(P_d, P_h) = \frac{|V_d \cap V_h|}{|V_d \cup V_h|}$$

- S_C assesses how similar P_d is to the context P_h:

$$S_C(P_d, P_h) = \frac{\sum_{i=0}^{m} \delta_1(a_i^{p_d}, a_i^{p_h})}{\sum_{i=0}^{m} \delta_2(a_i^{p_d}, a_i^{p_h})}$$

with

$$\delta_1(a_i^{p_d}, a_i^{p_h}) = \begin{cases} 1 \text{ if } a_i^{p_d} = a_i^{p_h} \\ 0 \text{ otherwise} \end{cases} \quad \delta_2(a_i^{p_d}, a_i^{p_h}) = \begin{cases} 1 \text{ if } a_i^{p_d} = a_i^{p_h} \text{ or } a_i^{p_d} = \star_i \\ 2 \text{ otherwise} \end{cases}$$

Note that we penalize patterns P_d that instantiate an attribute a_i with a value different from $a_i^{P_h}$ instead of keeping the symbol . For instance, given the hidden context $C_h = (a, b, c)$, $C_{d_1} = (a, \star, c)$ has a better S_C score (i.e., $\frac{2}{3}$) than $C_{d_2} = (a, b_2, c)$ whose restriction on the second attributes is wrong (the S_C score is $\frac{1}{2}$).

Since several patterns are hidden, we assign to P_d the maximal score on the hidden patterns:

$$S_V(P_d) = \max_{P_h} S_V(P_d, P_h) \text{ and } S_C(P_d) = \max_{P_h} S_C(P_d, P_h)$$

Finally, we define a unique aggregated score for each P_d as the harmonic mean between S_C and S_V:

$$S(P_d) = \frac{S_V(P_d) + S_C(P_d)}{2}$$

We generate several synthetic datasets that differ by the weight, the link probability and the noise level parameters used. In each dataset, 5 hidden patterns are embedded.

In Fig. 4, we investigate the individual quality of the retrieved patterns for three settings:

1. weight=10 and linkProb=0.1

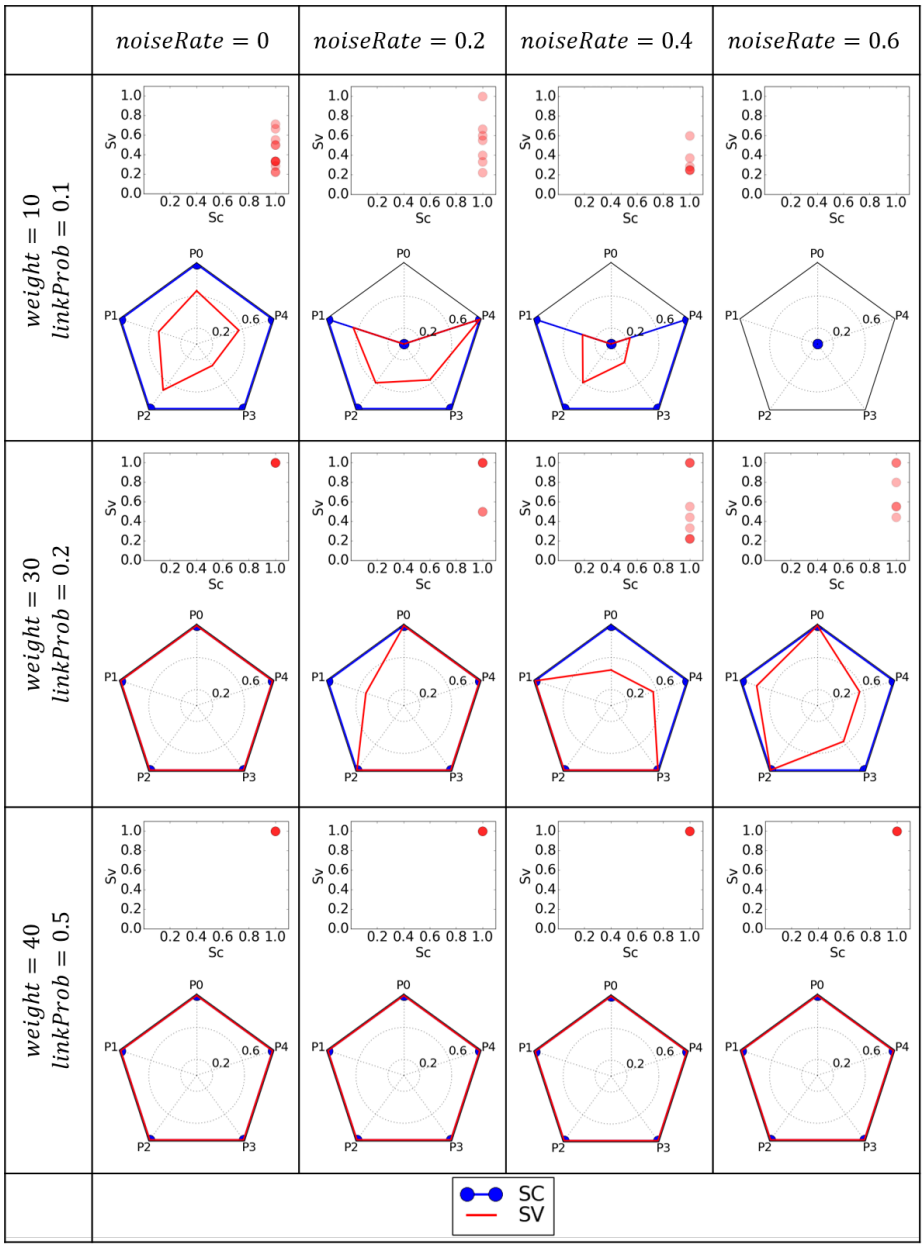


Figure 4: .

2. weight=30 and linkProb=0.2
3. weight=40 and linkProb=0.5

where the noise level varies from 0 to 0.6. The scores $S_C(P_d)$ and $S_V(P_d)$ of each discovered pattern show that the computed *exceptional contextual subgraph*s have the same context as the hidden patterns, but can differ by their related connected component, especially when the link probability and the weight are low and the noise level is high. In most of the cases, all the hidden patterns are retrieved partially or totally as indicated in the radar plots in Fig. 4 (second, fourth, and sixth row). Fig. 5 reports the average and the standard deviation of the scores S of the patterns obtained on these datasets (i.e., this is the result of the aggregation of individual results provided in Fig. 4).

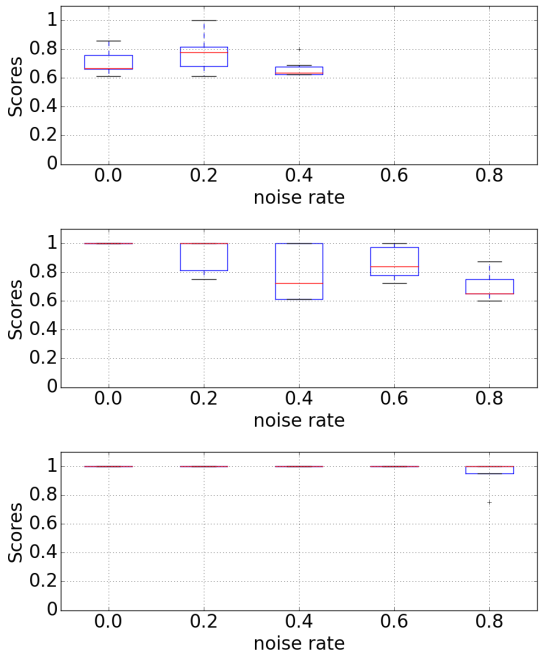


Figure 5: Boxplot of the discovered pattern scores S w.r.t. different levels of noise for different settings (top: weight=10, linkProb=0.1, middle: weight=30, linkProb=0.2, bottom: weight=40, linkProb=0.5).

We also ran COSMIc on 80 artificial datasets, generated with 4 different weight values (10,20,30,40), 4 link probability values (0.1, 0.2, 0.3, 0.5) and 5 of noise rates (0, 0.2, 0.4, 0.6, 0.8) and we report the median of the score S, as aggregated in Fig. 5, in Fig. 6. These results demonstrate that our approach is able to discover hidden patterns even if the dataset is very noisy (up to a noise rate equal to 0.8). Indeed, either the patterns are perfectly retrieved (i.e., S is equal to 1), or the patterns are partially discovered with an incomplete context description or a partial coverage of the vertices. As expected, the higher the density and weight of the hidden patterns, the more robust the approach is to noise. It is also important to note that whatever the configuration, our

algorithm does not return patterns when the noise rate is equal to 1. Actually, the statistical test that has to be satisfied by each edge of a pattern makes it impossible to return nonsensical patterns.

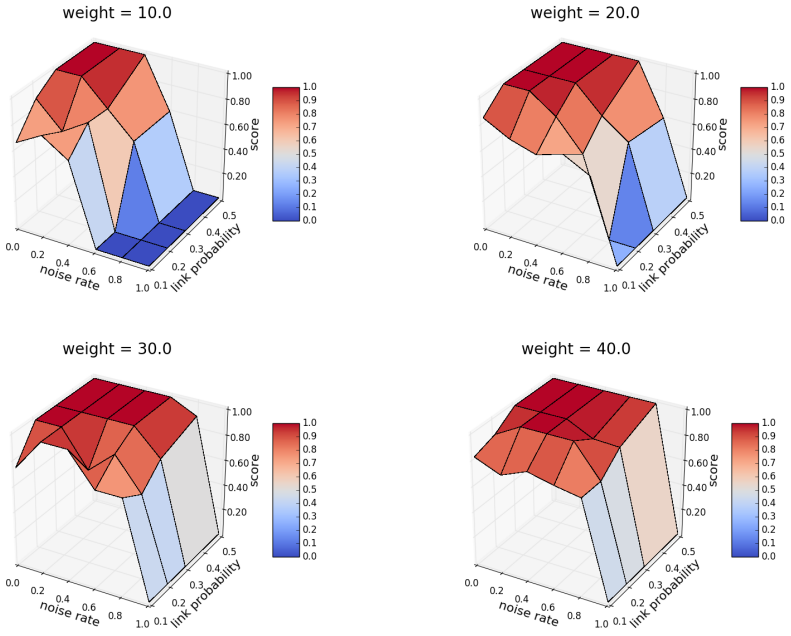


Figure 6: Median score S of the discovered patterns with respect to the noise rate and the link probability for 4 different weight values.

5.3 Performance study

We also use our artificial data generator to study the behavior of COSMIc with regard to several factors: The number of transactions, the number of vertices, the number of attributes and the cardinality of the attribute domains. We generate datasets by varying a single factor, the other ones being fixed. To avoid atypical results due to the randomness, we generate 10 datasets for each settings and report the median of the execution times as well as the median number of discovered patterns and their median score S, defined in the previous subsection. In this set of experiments, we use the default values for the generator that are given in Table 3.

Fig. 7 reports the run time of COSMIc, the number of discovered patterns and their quality when the number of vertices is varying. While the other parameters remain unchanged, the number of vertices has no influence on either the execution time or the number of patterns and their related quality.

Fig. 8 presents the same quantities when the number of transactions change.

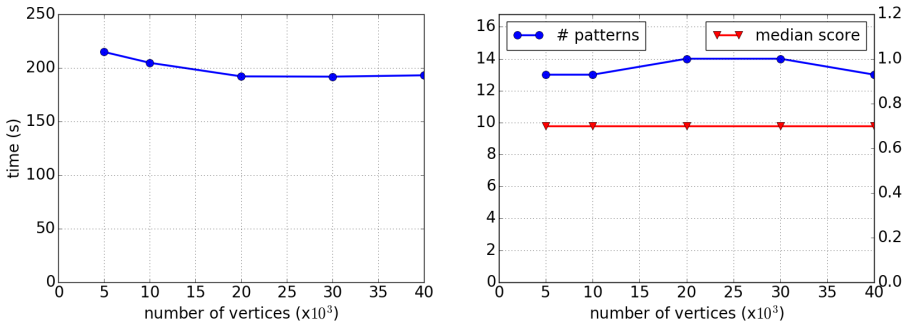


Figure 7: Run time (left), number of discovered patterns and their score S (right) w.r.t. the number of vertices.

We can observe that the run time increases proportionally to the number of transactions, whereas the number of discovered patterns tends to decrease. The larger the number of transactions, the better the quality of the discovered patterns.

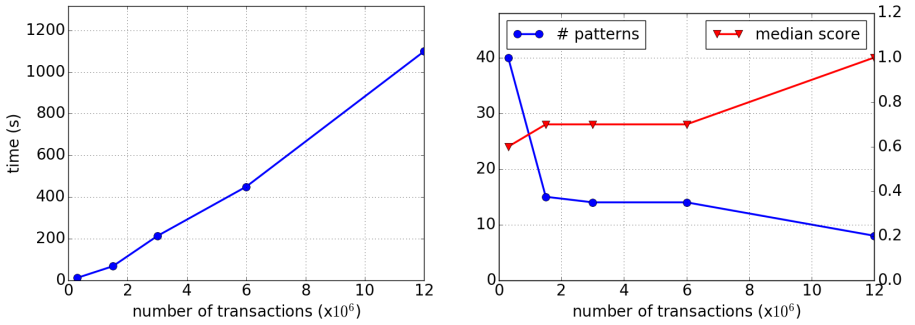


Figure 8: Run time (left), number of discovered patterns and their score S (right) w.r.t. the number of transactions.

In Fig. 9 and 10, we respectively report on the behavior of our algorithm when varying the number of attributes and their domain cardinality. Obviously, adding new attributes or increasing the size of the attribute domain results in a larger search space. Therefore, the execution time increases when either the number of attributes or the size of the attribute domain increase. The number of attributes is the more influential factor. Its increase leads to the discovery of larger sets of patterns with worse quality. Notice that even if the quality of the patterns is decreasing, it remains satisfactory (i.e., greater than 0.6 with 7 attributes). We observe the same phenomena when we increase the size of domain values.

What is different and interesting to see is the convexity/concavity of the curves, however: small attribute domains lead to a lack of diversity in possible

noise patterns – while the score shows that some of those patterns are spurious, the number stays small. Increasing the domain size allows larger variety in patterns that will be mistakenly identified as contexts – number of patterns rises and scores fall. Finally, there is a tipping point reached at which the domain size becomes so large that many different noise contexts are generated, none of which is considered significant.

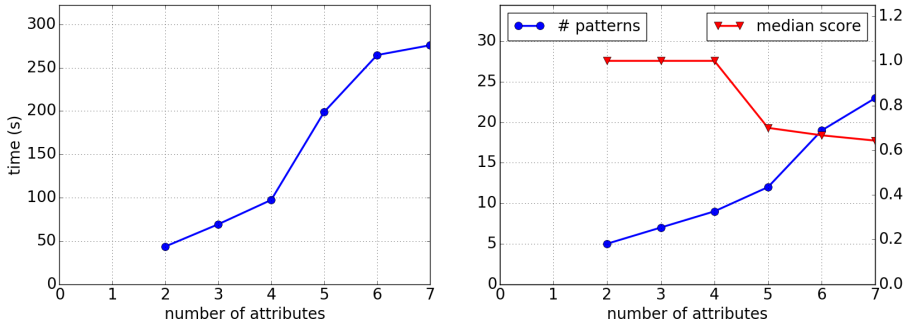


Figure 9: Run time (left), number of discovered patterns and their score S (right) w.r.t. the number of attributes.

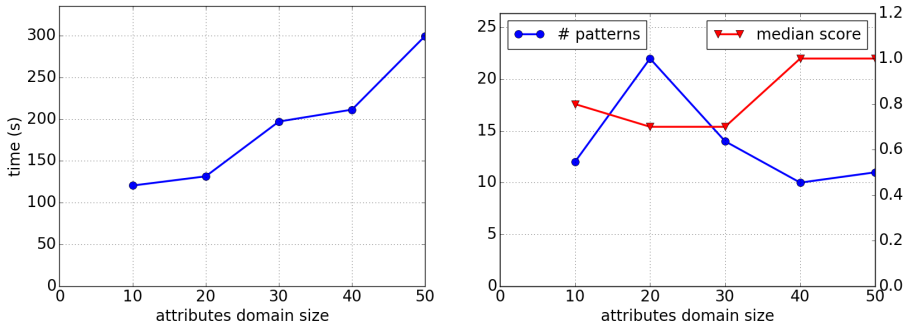


Figure 10: Run time (left), number of discovered patterns and their score S (right) w.r.t. the cardinality of the attribute domains.

We also studied in Fig. 11 the behavior of our algorithm w.r.t. the replication factor, i.e., when the number of transactions increases while depicting the same phenomena, preserving the search space. To this end, we report the ratio of the execution time for several values of the replication factor. This execution time ratio is equal to execution time needed to mine the replicated dataset divided by the execution time on the original dataset. In this way the ratio is equal to 1 for the original datasets (i.e., for a replication factor equaling 1). We considered several settings for the size of the attribute domains, the number of attributes, the number of vertices and the number of transactions. Obviously, these ratios increase with the replication factor. But it is important to note that,

for all configurations, the execution time ratio increases sublinearly, i.e. ratios are lower than the replication factor itself. For instance, in most of the cases, with replication factors equal to 16, the algorithm takes about 10 times as long to perform the extraction as on the original datasets. This means that some pruning properties become more efficient when the replication factor increases. This experiment demonstrates that our algorithm scales well with the replication factor.

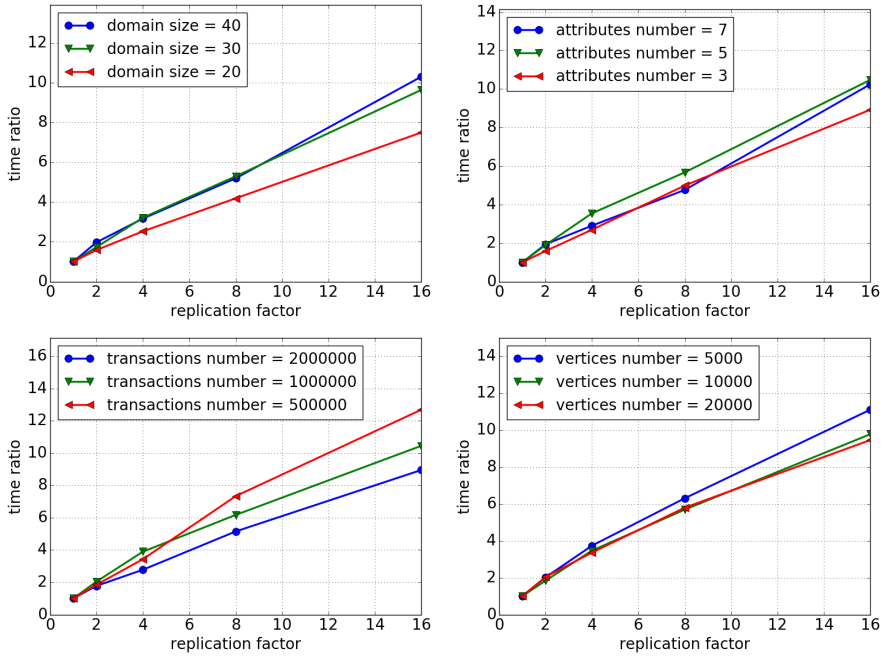


Figure 11: Run time ratio w.r.t. the replication factor for several data generator settings.

We also evaluated the ability of the closure operator to reduce the size of the search space, that is, enumerating only contexts C with strictly different set of transactions $M_{C \to T}(C)$. For that matter, we generated a dataset that ensures that many of the contexts will be enumerated in a reasonable amount of time. Most importantly, we set 4 attributes each with 10 different values (hence 4.1 million possible contexts at maximum), 2000 transactions, 200 vertices and a noise rate set to 0.1. We ran COSMIc without constraints on the minimum number of nodes/edges nor on the minimum WRAcc per edge, but with a minimum WRAcc sum of 0.5, and with minimum weight *min_weight* varying between 2 and 10, the maximum weight in the data. In Fig. 12, `closed contexts` plot corresponds to COSMIc, whereas `all contexts` plot is obtained by removing lines 12 and 23 in Algorithm 1. The results clearly show the significant impact of the closure on reducing both run time and number of actually

explored patterns.

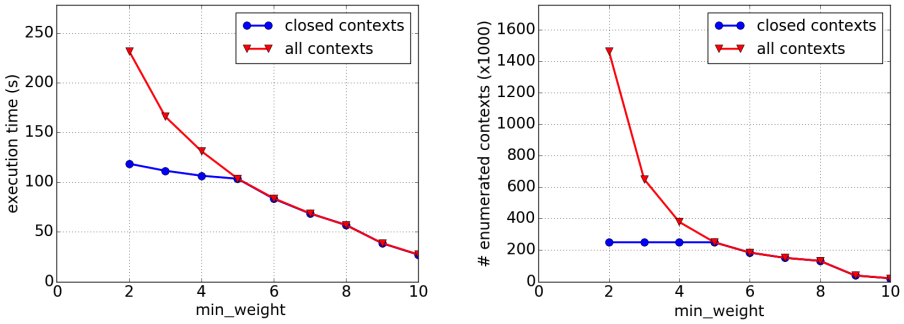


Figure 12: Run times (left) and number of patterns (right) when varying the edge minimum support *min_weight*: Closed versus non-closed pattern enumeration.

Finally, we assessed the ability of the upper bounds to prune unpromising context specializations, that is, stopping the enumeration when the upper bounds cannot be satisfied. We generated an artificial dataset with similar parameters as in the previous paragraph except that we did not hide any pattern and set noise rate to its maximum (1.0). The efficiency of the upper bounds is illustrated in Fig. 13, where `with pruning` stands for COSMIc and `without pruning` is obtained by removing lines 13 and 24 in Algorithm 1. In this case also, we can see the impact of the pruning technique on the performance of the algorithm.

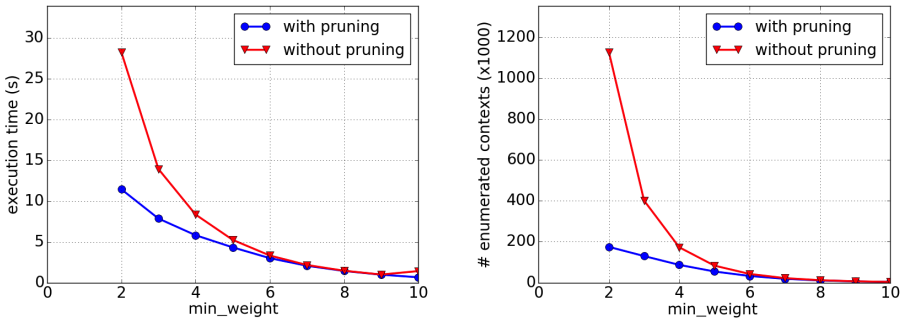


Figure 13: Run times (left) and number of patterns (right) when varying the edge minimum support *min_weight* with and without pruning.

6

6.1 Comparison to MiMag

As we discuss in Section 2, several approaches for mining edge-attributed graphs have been proposed in the literature. The one closest in goal to our technique, an algorithm called MiMaG, has been introduced in [7]. In that framework, graphs exist in several *layers*, and edges have weights that are determined by those layers, obviously a formulation that is rather close to our own. MiMaG attempts to find quasi-cliques formed by edges with similar weights, and group layers in which the same nodes are involved in a quasi-clique. We therefore explore to what degree that technique can discover patterns in the data that we use.

To translate our data into a representation that can be processed by MiMaG, we have to choose appropriate representations and adjust the parameter values:

1. How to define graph layers?
2. How to define edge weights?

Graph layers The obvious representation would consist of letting each distinct context define its own layer. Returning to the example we gave in Table 1 in the beginning, there would be 13 layers to the graph (all possible combinations of Gender, Age, Time and Weather that occur in the data), as shown in Table 4 (Graph layer modeling 1).

While MiMaG *groups* layers, however, it has no capability to *generalize* them. This means that it might group disjoint contexts if we use only this form of layers. Alternatively, we can treat each *attribute-value* combination as a distinct layer, leading to 12 layers (see Table 4 (Graph layer modeling 2)). This would allow MiMaG to group a subset of attribute-value pairs contained in contexts, and in this manner generalize them, e.g. combining Time='Night' and Weather='Windy', subsuming three of the contexts.

Edge weights As in our earlier discussion in Section 2, it obviously does not make too much sense to use absolute edge weights, particularly not if we try to assess edges' similarity. Instead, we can either normalize by the largest weight an edge has for the context or attribute-value pair, or calculate the WRAcc values. In the former case, MiMaG will group edges that have the same relative weight, in the latter edges with the same WRAcc value. Table 5 gives the two types of weights for the edge (C,D) in our toy example. For the full context shown in the first row, the edge is clearly not particular, yet its relative weight would make it appear highly similar (in fact, identical) to all other edges for that context (and for all other contexts). For individual attribute-pairs, on the other hand, WRAcc scores give more expressive results than weights (the edge is under-expressed for the gender and age attributes, over-expressed for the others), and can separate the time and weather effects, which the relative weight cannot. To aid MiMaG, we filter out edges that have a negative WRAcc value.

Parameter settings MIMAG has two parameters: γ influences the degree to which quasi-cliques need to be connected, and w is the tolerance parameter deciding whether edge weights are considered similar or not. There is no clear guidance how to set those parameters: $\gamma \in [0, 1]$ but anything below 0.5 denoted non-dense cliques (a type of pattern COSMIc can discover). Given that we have normalized weights, we know that $w \in [0, 1]$ but we cannot decide a priori what is a good value. The authors of the original paper evaluate their approach with $\gamma = 0.5$ and $w = 0.1$ and we therefore use the same parameters. We use the Java implementation of the algorithm that has been provided to us by the authors.

Results We generated data using the default parameters given in Section 5.1, with the difference that we lower $nbVertices$ to 1000, and $nbtrans$ to $100,000$. For larger values, MIMAG runs out of memory even when the Java virtual machine is provided with 16 GB of main memory; in fact, for $nbtrans = 250,000$ and larger using contexts as layers leads to crashes of the program when *reading* the data file.

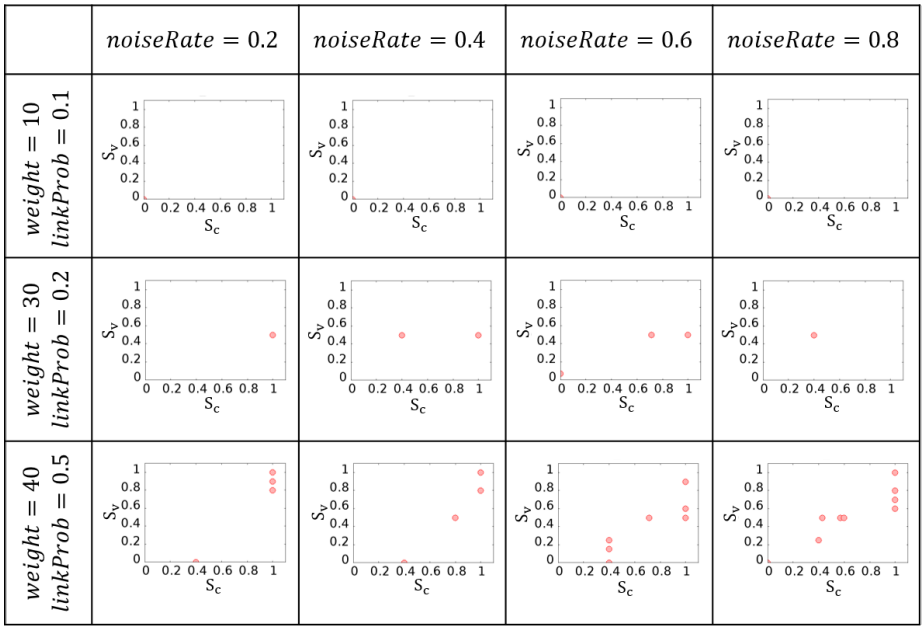


Figure 14:

We use the scores S_V, S_C (see subsection 5.2) to evaluate the quality of the results. Using contexts as layers leads to poor results: Scores of the found patterns (if any) are 0 in almost all cases. Using attribute-value pairs as layers gives MIMAG the possibility to find patterns that partly correspond to the hidden patterns, as Fig. 14 shows. MIMAG needs rather high values of weight and link probability to recover the patterns. The best results are achieved

for the data set with a link-probability of 0.5 (bottom row), i.e. relatively dense graphs. Noise seems to have little effect on its performance, neither for weight=30, link-probability=0.2, nor for the data in the bottom row. In either case, however, the results are inferior to those of COSMIc. Interestingly, MiMaG still returns contexts when the noise probability is 1.0, something that should be impossible given that no contexts are embedded anymore, and edge descriptions contain only noise. This behavior, together with the enumeration of contexts that have little to do with the hidden ones in terms of descriptors and vertices, indicates that this might simply the result of frequency: certain attribute-value pairs and certain vertices appear more often in the data and MiMaG selects those. It seems therefore as if MiMaG combines frequently occurring attributes and vertices, and that some of those somewhat accidentally agree with hidden contexts. Given that an in-depth discussion of the strength and weaknesses of MiMaG is out of the scope of this paper, however, we leave the exploration of this question open to interested readers.

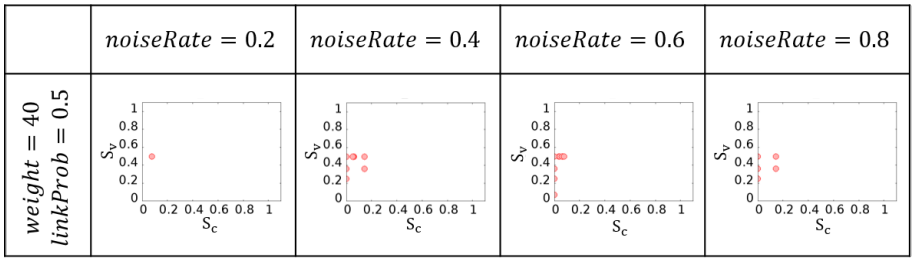


Figure 15:

Fig. 15 reports the results obtained on the same datasets as in the bottom row of Fig. 14, but using the relative weight measure. The patterns are of clearly worse quality than the ones obtained using WRAcc: Their S_C is much lower and S_V does not improve.

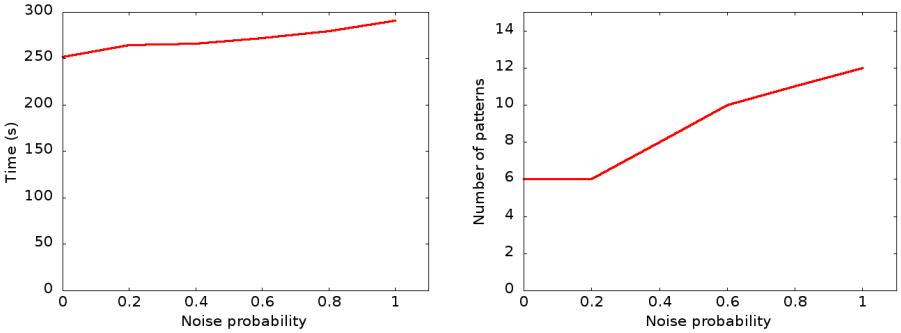


Figure 16: Run time (left), number of discovered patterns (right) w.r.t. different levels of noise for attribute-value pair layers.

Performance-wise, Fig. 16 shows that MiMaG is not faster than COSMIc, even while finding a comparable amount of patterns overall. Notably, each pattern involves typically five nodes or less.

There is arguably an explanation for the inability of MiMaG to group more nodes together: The parameter value $\gamma = 0.5$ means that it is searching relatively dense subgraphs. The problem is, however, that lowering that value causes MiMaG to encounter memory problems again: for $\gamma = 0.3$, for instance, it crashes when mining data with attribute-value pairs as layers. When mining data with context as layers, the process terminates with long running times, yet fails to find any patterns. Using MiMaG to address our problem setting therefore requires significant preprocessing of the data: the tasks that COSMIc performs internally – decomposing contexts and calculating WRAcc – need to be done beforehand. Even then WRAcc stays static, however, and does not change while attributes are combined, leading to patterns of lower quality. As a conclusion, those experiments show that MiMaG can hardly be adapted to solve the problem we consider here, which is quite different from the one addressed by the authors in [7].

6.2 Comparison to Exceptional Model Mining

As mentioned earlier, our problem is a particular instance of exceptional model mining (EMM) [29, 16]. Therefore, we instantiate our problem as the closest existing EMM setting that can partially model our setting to evaluate the performance of an EMM approach. Recall that our dataset is an augmented graph: Each transaction takes values for some numerical and categorical attributes and is also associated to an edge. These attributes values vectors form the object descriptions in EMM. Each edge found in the dataset is a binary target. In this reformulation, an EMM algorithm searches for subgroups $(S, M_{T\to C}(S))$ (or equivalently $(M_{C\to T}(C), C)$), for any subset of transactions $S \subseteq T$ and context $M_{T\to C}(S)$. Subgroups are evaluated with a quality measure, for instance the (weighted) Kullback-Leibler divergence ((W)KL): it measures the difference between the target attributes' distribution (the graph edges' distribution) within the subgroup and within the full dataset. The higher the difference, the more exceptional the subgroup. Contexts for which the appearance of edges is exceptional are searched for.

Example. Consider Table 1 and the context $C = (\star, Night, Windy)$. We have that $T = M_{C\to T}(C) = \{m_2, m_7, m_{10}, m_{15}, m_{17}\}$ and (T, C) is a subgroup. The WKL of a subgroup is

$$WKL((T,C)) = \frac{|T|}{|R|} \times \sum_{e\in E} p(e|T) log_2 \frac{p(e|T)}{p(e|R)},$$

so for our example: $WKL((T,C)) = \frac{5}{18} \times ((1/5 \log_2 \frac{1/5}{5/18}) + (1/5 \log_2 \frac{1/5}{3/18}) + (1/5 \log_2 \frac{1/5}{3/18}) + (0/5 \log_2 \frac{0/5}{2/18}) + (2/5 \log_2 \frac{2/5}{5/18}) = 0.04$.

Although this modelization partially fits to our problem, it suffers from major issues that we discuss below. For some elements of the discussion, we

ran several experiments with this modelization and the DSSD algorithm. It performs a beam-search through the lattice of subgroups, enabling the discovery of a diverse set of subgroups [49, 50] and considers the WKL quality measure. Given that this heuristic exploration is not able to deal with large graphs, we did not experiment with exhaustive explorations (e.g., SD-Map [3]).

Subgroup interpretation. Knowing that a subgroup, or context, is exceptional is not enough: we need to know for which edges this is the case. In other words, the targets of the objects within a subgroup induce a weighted subgraphs, and selecting which edges are important remains to be done. A solution is to keep only over-expressed edge according to the WRAcc measure (edges whose WRAcc is strictly positive). This can however result in an over-abundance of connected components, as well as in too small subgraphs, or even individual edges. Most importantly, the WKL suffers from the *curse of dimensionality*: when dealing with numerous targets (edges), it is very likely that the best subgroups appear exceptional due to a slight *global* difference in the distribution of edges, and not a strong *local* one that affects only a few edges[3]. The best contextual graphs may be missed.

Run times and memory consumption. As illustrated hereafter, DSSD, under various configuration, is not scalable enough to solve our problem. This is mainly due to the fact that each edge of the graph has to be encoded as a target attribute. When experimenting with DSSD[4], we used its default parameters for the beam-search exploration (except a depth at least equal to the number of attributes). We set the following default parameters for generating synthetic data: $10,000$ transactions with 5 attributes each with 20 possible values taking edges in a graph of 100 vertices (with $weight = 20$, $linkProb = 0.5$ and $noiseRate = 0.1$) in which 5 patterns of 10 vertices are hidden. Fig. 17 presents the run times and quality scores when varying the number of transactions. As a first remark, DSSD scales badly while COSMIc shows equal or smaller run times for much larger data (see the previous subsections). This result becomes even more striking given that DSSD uses a heuristic approach while COSMIc performs an exhaustive search. We ran DSSD with a number of different parameter settings and rarely retrieved the hidden patterns. Most of the time, only a single pattern was retrieved with scores indicating that context or vertex coverage was incomplete (see Fig. 17 (right) and Fig. 18), even when changing DSSD parameters to ensure more diversity in the output. One way of solving this problem is to enlarge the beam width (100 by default), but it comes with longer run times and results in premature DSSD termination.

[3] A parallel could be drawn to the case when one has to use bi-clustering techniques over traditional clustering in the presence of a large number of attributes [31].

[4] Provided by the authors of the DSSD at `http://patternsthatmatter.org/dssd/`

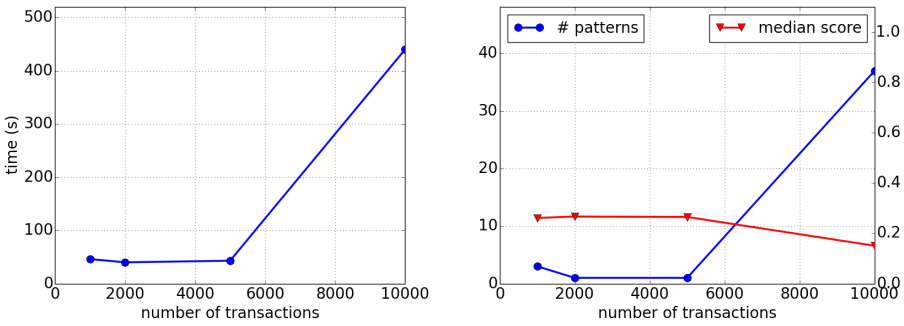


Figure 17: Run times and quality scores w.r.t. a varying number of transactions when experimenting with DSSD.

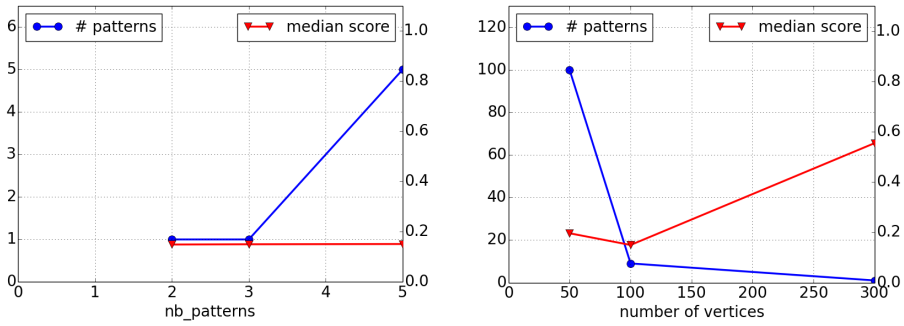


Figure 18: Run time and quality scores w.r.t. a varying number of hidden patterns (left) and vertices (right)when experimenting with DSSD.

7

7.1 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[5]. 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 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 two first 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 [21]; the *zipcode*, gender, country, and age of the biker (where $age \in \{[14; 25][25, 60], \geq 60\}$ still according to [21]) as well as the type of *pass* subscribed by the user.

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

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 The information used is *TrainStation, University, , 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. 19) 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.

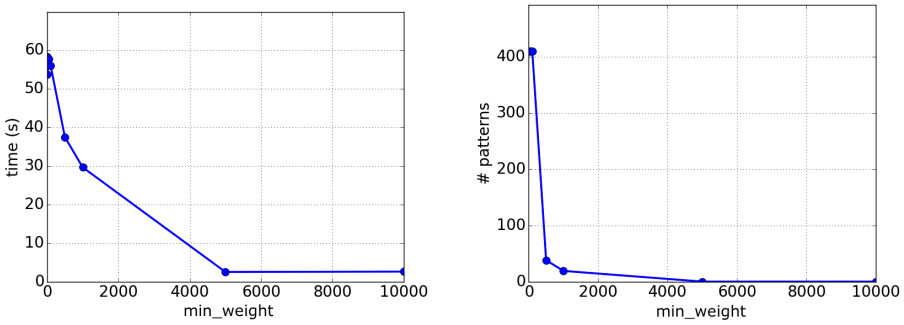
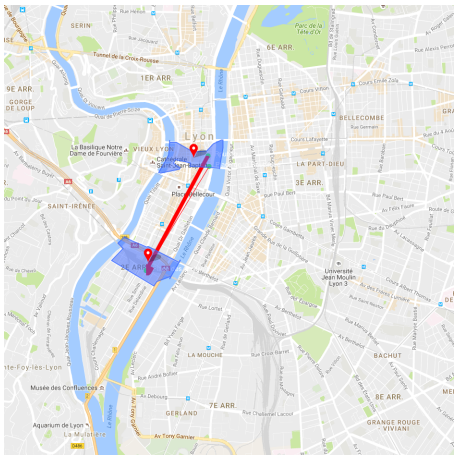


Figure 19: Run times (left) and number of patterns (right) on VÉLO'V when varying min_weight.

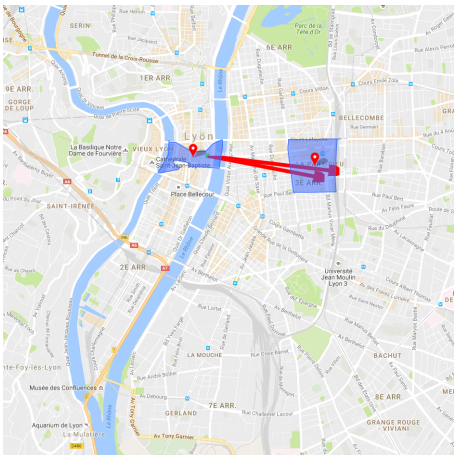
Fig. 20 presents a visualization on the map of Lyon of six patterns that we extracted from different datasets introduced above[6]. For each, we detail the experimental protocol and propose an interpretation.

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

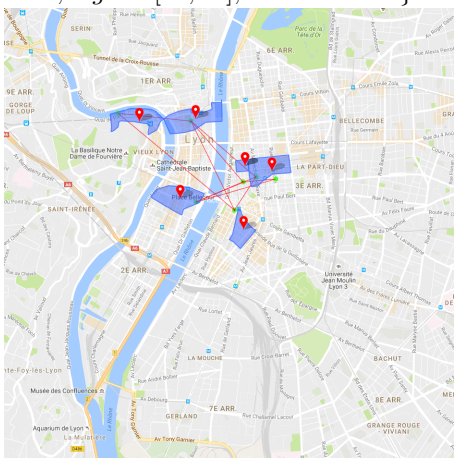
[6] All results can be explored through a user friendly interface `http://liris.cnrs.fr/dm2l/projects/graisearch/mlj/`



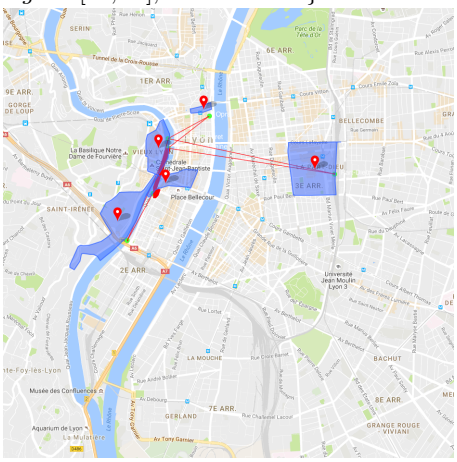
(a) $C_1 = \{Zip\_code = \underline{38},\ Gender = men,\ Age = [26, 60],\ Pass = oura\}$



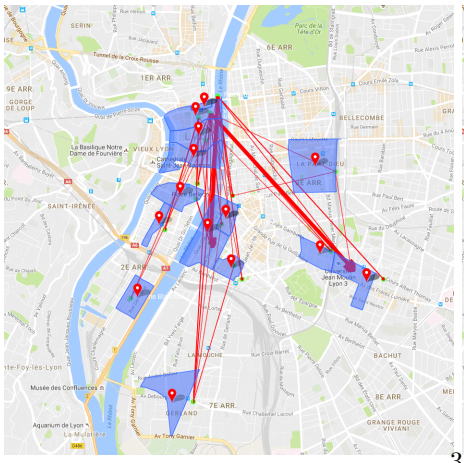
(b) $C_2 = \{Zip\_code = \underline{38},\ Gender = men,\ Age = [26, 60],\ Pass = oura\}$



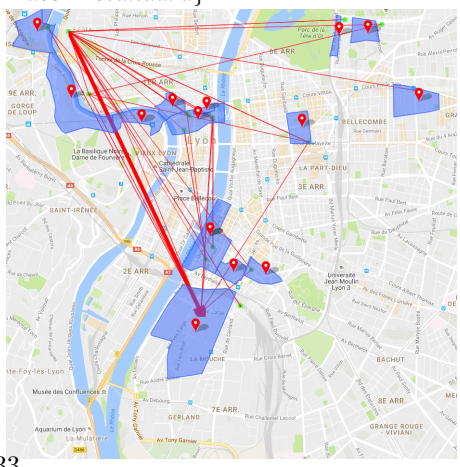
(c) $C_3 = \{Time = \underline{night},\ Country = France,\ Pass = velov\}$



(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 20: Several contextual subgraphs discovered in the VÉLO'V datasets.

$min\_vertex\_size = 2$, $min\_edge\_size = 1$ and $min\_weight = 6$). 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. 20 (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. 20 (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. 20 (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. 20 (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 behaviour for particular demographics (without specifying them explicitly: The root pattern has just a single attribute instantiated).

7.2 Behavioral mobility patterns in Dota 2

Electronic sport (eSport) is an emerging domain where the most skilled gamers are hired by professional teams, surrounded by sponsors, and compete in international tournaments [47], widely followed on live streaming platforms such as *Twitch.tv* [25][7] Its development impacts our society: For example, a law project in France is studying the legal status of e-sport athletes and tournaments just as for off-line athletes[8] Academics and experts in sport analytics are starting to get interested in this emerging topic [51, 32] as well. Strategic video games received much attention from the AI community for an extended period [37], attention that was renewed after recent announcements from the DeepMind team naming a video game as the next challenge after Go[9]

In this context, we study Dota 2, a multiplayer online battle arena video game released in July 2013. Up to February 2015, Dota 2 attracted tournaments totalling US$ 25 million in prize money, becoming one of the most lucrative competitive video games. Just as in sport, players are gathered as a team with coaches and practice as a daily routine. Behavioural data analytics start to play a key role to understand and model the opponents and thus prepare tournaments, here again, just as for any athlete or sport team with sport analytics (baseball, soccer and basket-ball). We assess our methodology showing that behavioural patterns specific to game conditions and players can be discovered from Dota 2 games. These patterns can be used to understand the behaviour of a single or several players (the subgraph) at various stage of the game and under several conditions (the contexts).

Dota 2 **problem settings** A game is played on a map where two teams of five players are battling each other in real time. Each team has to defend their own stronghold and destroy the opponent's one to win. Each player controls a *hero* that he moves on the map, and needs to train, by collecting gold, new items and abilities, and by fighting opposing heroes. Fig. 21 displays the initial influence zone of both teams. The red team called *the dire* (resp. green for *the radiant*), defends their stronghold at the top right corner (resp. bottom left). Three lanes (*top*, *mid*, *bot* in Fig. 22 (i)) separate the teams, on which defensive towers are set. The players have well defined roles, depending on the heroes they initially picked and their properties (110 available heroes). One role consists of defending and extending the influence zone in a specific lane, another is to quickly switch lanes to attack by surprise. Knowing that a team only sees controlled map zones, estimating enemies positions and triggering team fights at well-chosen times and in well-chosen areas is key to success. As in any traditional sport, professional

Figure 21: DOTA Map.

[7] recently acquired by *Amazon* for US$ 970 million

[8] `http://www.gouvernement.fr/partage/6761-esport-en-france-on-avance` (2016)

[9] `http://www.theverge.com/2016/3/10/11192774/demis-hassabis-interview-alphago-google-deepmind-ai`

teams study their soon-to-be opponents. Understanding what are the specific zones controlled by a player in some contexts is crucial, and allows teams to adapt their strategy and prepare the tournaments.

Scenario Any action performed during a game is stored afterwards in a file (replay), allowing to re-watch it at any time. Replays and parsing tools are freely available on `dotabank.com` and the *skadistats* GitHub. We randomly selected an expert game from *The International 3 Eastern Qualifiers*: lasting 42 minutes and won by *the dire* (more information on `dota2stat.com`, match #199392262). We built different augmented graphs from this game as follows. The map is cut into n^2 non-overlapping squares of equal width/height and each cell of this grid is a vertex $v \in V$, edges of the graph are hero travel paths (movement between two cells), R is the set of attributes describing players heroes properties at the moment of the movement. Game time is measured in *ticks* (30 ticks per second): There is at most $|T| = 30 \times 42 \times 60 \times 10 = 756,000$ movements (as there are 10 players). As such, we work directly on aggregated data by rounding game times to a factor of w seconds and grid resolution n. We build two datasets with different resolutions in space and time: DOTA_1 with $w = 600$ and $n = 60$ (which gives $482,937$ transactions and $3,475$ vertices) DOTA_2 with $w = 1800$ and $n = 100$ (that is, $482,250$ transactions and $11,263$ vertices). The datasets also differ by their attributes: Transactions are defined on time (discretized game ticks), hero type, team (dire or radiant) for DOTA_1. We add two attributes in DOTA_2 : percentage of remaining life (at zero, the hero dies and waits a time proportional to the current game time before re-spawning) and percentage of remaining mana (consumed when using special tactics). A transaction example is $t = \{Jakiro, [0-600], dire\}$ with edge $\{(10, 32)\}$: A player of the *dire* team moved his *Jakiro* from cell 10 to cell 32 in the first 600 seconds of the game.

Experimental results We run COSMIc on the two datasets searching for contextual subgraphs having at least 30 nodes and 29 edges. We compute the average and deviation of the WRAcc measure for each pattern. We remove patterns dominated by another on all these dimensions to reduce the number of output patterns (that is, we use a skyline operator seeking to minimize deviation and maximize the other measures). DOTA_1 produces 77 *exceptional contextual subgraph*s out of 29 different contexts in 363 seconds while DOTA_2 produces 158 *exceptional contextual subgraph*s out of 124 contexts in 230 seconds. Fig. 22 presents some of the best patterns (highest average WRAcc). In (i), contexts are: $\{SD, w_2, radiant\}$ and $\{SD, w_3, radiant\}$: these two connected components show different characteristic zones of the hero *ShadowDemon* (SD) at two different phases of the game. While this player is *aggressive* in time window w_2 pushing on the top lane, he mainly walks around his stronghold in the next window. Given that he belongs to the losing team, we can assume that the latter is a *defensive pattern*. The pattern in Fig. 22 (ii), whose context is $\{Juggernaut, w_2, dire\}$, characterizes a role mentioned before: It represents quick lane switches to help team mates and attack by surprise. Finally, two

*exceptional contextual subgraph*s of DOTA_2 are given on Fig. 22 (iii), sharing the same context $\{w_0, radiant, mana \leq 25\%\}$. They clearly show *mana-back-up* trajectories: in the early stage, mana is a rare resource, and getting back inside the stronghold allows a player to quickly regain all of his mana, which otherwise increases very slowly out of the stronghold.

These four examples of patterns show large connected components representing movements or behaviors linked to their context. Knowing that the full graph is large (with thousand of nodes) and contains many movements (up to 720,000), the fact that we are able to discover large graphs (several hundred nodes and edges) that are not necessarily dense, emphasizes the ability of COSMIc to discover behavioral mobility patterns. The main difference with VÉLO'V is the rate of mobility (30 movements per second for a player, no more than 10 bike rides per day). Moreover, the semantics of the discovered behavioral patterns (subgraphs) can be explained by their context: For many examples including the four presented here, the mobility pattern is effected by an expert player. It remains for future work to apply COSMIc in various scenarios, implying a detailed experimental protocol and involving a deep knowledge of the game. It is indeed particularly important for eSport structures to find this type of patterns in dataset composed of several games of (i) a player to study his strategies, (ii) the same 5-players team to study their common tactics or even (iii) with transactions of a single hero to discover the most rewarding movements in terms of experience, gold, ... earnings (that is find the best way of using that hero).

(i) $\{SD, w_2, rad\}$ (top) $\{SD, w_3, rad\}$ (bottom) **(ii)** $\{Juggernaut, w_2, dire\}$ **(iii)** $\{w_0, rad, mana \leq 25\%\}$

Figure 22: DOTA 2 patterns: to ease visualization, only vertices are shown (no edges).

8 Conclusion

In this paper, we defined the problem of finding *exceptional contextual subgraph*s in augmented graphs. This problem has many applications, especially location based social networks, , and recommendation systems. It enables to discover connected components highly characteristic of specific category of users

and time periods. We showed how an inductive approach rooted in Exceptional Model Mining can answer this challenging problem. This is achieved thanks to an efficient data mining algorithm COSMIc that avoids materializing all context/subgraph pairs and benefits from pruning and upper bound computations techniques. We evaluated COSMIc on both synthetic and real-world datasets. For that matter, we designed an augmented graph generator that allows to hide exceptional contextual subgraphs and showed that COSMIc is able to retrieve the hidden patterns in noisy data and to scale w.r.t. the parameters of the input data (attribute domain size and number, number of transactions and vertices). We compared our approach to the closest existing formalisms and algorithms we could find and discussed how they fail to answer our problem. Eventually, we provided two case-studies (i) on the analysis of a bike-sharing system, where discovered patterns are helpful for the VÉLO'V system operators (e.g. discovering stations and mobility patterns involving young people at night) and (ii) on the analysis of DOTA 2 replays, a well-known game in eSport, for which the discovered patterns explains the mobility behaviors of players.

Acknowledgement

The authors would like to thank the anonymous reviewers for their frank, fruitful, constructive and insightful comments and the authors of the MiMaG and DSSD algorithms for providing us their prototypes. They also gratefully acknowledge Pierre Houdyer for the development of the pattern visualization platform on VELOV data. This work has been partially supported by the projects GRAISearch (FP7-PEOPLE-2013-IAPP) and VEL'INNOV (ANR INOV 2012).

References

[1] Rezwan Ahmed and George Karypis. Algorithms for Mining the Evolution of Conserved Relational States in Dynamic Networks. In *ICDM*, pages 1–10. IEEE, 2011.

[2] Martin Atzmueller, Stephan Doerfel, and Folke Mitzlaff. Description-oriented community detection using exhaustive subgroup discovery. *Inf. Sci.*, 329:965–984, 2016.

[3] Martin Atzmüller and Frank Puppe. Sd-map - A fast algorithm for exhaustive subgroup discovery. In *PKDD*, volume 4213 of *LNCS*, pages 6–17. Springer, 2006.

[4] Michele Berlingerio, Francesco Bonchi, Björn Bringmann, and Aristides Gionis. Mining Graph Evolution Rules. In *ECML/PKDD*, pages 115–130, 2009.

[5] Michele Berlingerio, Michele Coscia, Fosca Giannotti, Anna Monreale, and Dino Pedreschi. Multidimensional networks: foundations of structural analysis. *WWW*, 16(5-6):567–593, 2013.

[6] Jérémy Besson, Céline Robardet, and Jean-François Boulicaut. Mining a new fault-tolerant pattern type as an alternative to formal concept discovery. In Henrik Schärfe, Pascal Hitzler, and Peter Øhrstrøm, editors, *Conceptual Structures: Inspiration and Application, 14th International Conference on Conceptual Structures, ICCS 2006, Aalborg, Denmark, July 16-21, 2006, Proceedings*, volume 4068 of *Lecture Notes in Computer Science*, pages 144–157. Springer, 2006.

[7] Brigitte Boden, Stephan Günnemann, Holger Hoffmann, and Thomas Seidl. Mining coherent subgraphs in multi-layer graphs with edge labels. In *KDD*, pages 1258–1266, 2012.

[8] Francesco Bonchi, Aristides Gionis, Francesco Gullo, and Antti Ukkonen. Chromatic correlation clustering. In *KDD*, pages 1321–1329, 2012.

[9] Karsten M. Borgwardt, Hans-Peter Kriegel, and Peter Wackersreuther. Pattern mining in frequent dynamic subgraphs. In *ICDM*, pages 818–822. IEEE, 2006.

[10] Björn Bringmann, Michele Berlingerio, Francesco Bonchi, and Aristides Gionis. Learning and predicting the evolution of social networks. *IEEE Intelligent Systems*, 25(4):26–35, 2010.

[11] Mahashweta Das, Sihem Amer-Yahia, Gautam Das, and Cong Yu. MRI: Meaningful interpretations of collaborative ratings. *PVLDB*, 4(11):1063–1074, 2011.

[12] Mahashweta Das, Saravanan Thirumuruganathan, Sihem Amer-Yahia, Gautam Das, and Cong Yu. An expressive framework and efficient algorithms for the analysis of collaborative tagging. *VLDB J.*, 23(2):201–226, 2014.

[13] Pedro Olmo Vaz de Melo, Christos Faloutsos, and Antonio Alfredo Ferreira Loureiro. Human dynamics in large communication networks. In *SDM*, pages 968–879. SIAM, 2011.

[14] Elise Desmier, Marc Plantevit, Céline Robardet, and Jean-François Boulicaut. Trend mining in dynamic attributed graphs. In *ECML/PKDD*, pages 654–669, 2013.

[15] Wouter Duivesteijn. A short survey of exceptional model mining: Exploring unusual interactions between multiple targets. In *2014 International Workshop on Multi-Target Prediction*, 2014.

[16] 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.

[17] Wouter Duivesteijn, Arno J. Knobbe, Ad Feelders, and Matthijs van Leeuwen. Subgroup discovery meets bayesian networks – an exceptional model mining approach. In Geoffrey I. Webb, Bing Liu, Chengqi Zhang, Dimitrios Gunopulos, and Xindong Wu, editors, *ICDM 2010, The 10th IEEE International Conference on Data Mining, Sydney, Australia, 14-17 December 2010*, pages 158–167. IEEE Computer Society, 2010.

[18] Michelle Girvan and Mark EJ Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826, 2002.

[19] Amit Goyal, Francesco Bonchi, Laks V. S. Lakshmanan, and Suresh Venkatasubramanian. On minimizing budget and time in influence propagation over social networks. *Social Netw. Analys. Mining*, 3(2):179–192, 2013.

[20] Stephan Günnemann, Ines Färber, Brigitte Boden, and Thomas Seidl. Subspace clustering meets dense subgraph mining. In *ICDM*, pages 845–850, 2010.

[21] 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.

[22] Akihiro Inokuchi and Takashi Washio. Mining frequent graph sequence patterns induced by vertices. In *SDM*, pages 466–477. SIAM, 2010.

[23] Meng Jiang, Peng Cui, Rui Liu, Qiang Yang, Fei Wang, Wenwu Zhu, and Shiqiang Yang. Social contextual recommendation. In *CIKM*, pages 45–54, 2012.

[24] Mehdi Kaytoue, Yoann Pitarch, Marc Plantevit, and Céline Robardet. Triggering patterns of topology changes in dynamic graphs. In *2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2014, Beijing, China, August 17-20, 2014*, pages 158–165, 2014.

[25] Mehdi Kaytoue, Arlei Silva, Loïc Cerf, Wagner Meira Jr., and Chedy Raïssi. Watch me playing, i am a professional. In *WWW (Comp. Vol.)*, pages 1181–1188. ACM, 2012.

[26] Arijit Khan, Xifeng Yan, and Kun-Lung Wu. Towards proximity pattern mining in large graphs. In *SIGMOD*, pages 867–878. ACM, 2010.

[27] Mayank Lahiri and Tanya Y. Berger-Wolf. Mining periodic behavior in dynamic social networks. In *ICDM*, pages 373–382. IEEE, 2008.

[28] Nada Lavrac, Branko Kavsek, Peter A. Flach, and Ljupco Todorovski. Subgroup discovery with CN2-SD. *Journal of Machine Learning Research*, 5:153–188, 2004.

[29] Dennis Leman, Ad Feelders, and Arno J. Knobbe. Exceptional model mining. In *ECML/PKDD*, pages 1–16, 2008.

[30] Florian Lemmerich, Martin Becker, and Martin Atzmueller. Generic pattern trees for exhaustive exceptional model mining. In Peter A. Flach, Tijl De Bie, and Nello Cristianini, editors, *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2012, Bristol, UK, September 24-28, 2012. Proceedings, Part II*, volume 7524 of *Lecture Notes in Computer Science*, pages 277–292. Springer, 2012.

[31] S.C. Madeira and A.L. Oliveira. Biclustering algorithms for biological data analysis: a survey. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 1(1):24–45, 2004.

[32] Tobias Mahlmann Matthias Schubert, Anders Drachen. Esports analytics through encounter detection. In MIT Sloan, editor, *Proceedings of the MIT Sloan Sports Analytics Conference 2016*, 2016.

[33] S. Morishita and J. Sese. Traversing itemset lattice with statistical metric pruning. In *PODS*, 2000.

[34] Flavia Moser, Recep Colak, Arash Rafiey, and Martin Ester. Mining cohesive patterns from graphs with feature vectors. In *SDM*, pages 593–604. SIAM, 2009.

[35] Pierre-Nicolas Mougel, Christophe Rigotti, Marc Plantevit, and Olivier Gandrillon. Finding maximal homogeneous clique sets. *Knowledge and Information Systems*, pages 1–30, 2013.

[36] 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.

[37] Santiago Ontañón, Gabriel Synnaeve, Alberto Uriarte, Florian Richoux, David Churchill, and Mike Preuss. A survey of real-time strategy game AI research and competition in starcraft. *IEEE Trans. Comput. Intellig. and AI in Games*, 5(4):293–311, 2013.

[38] 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.

[39] Adriana Prado, Baptiste Jeudy, Élisa Fromont, and Fabien Diot. Mining spatiotemporal patterns in dynamic plane graphs. *Intell. Data Anal.*, 17(1):71–92, 2013.

[40] Adriana Prado, Marc Plantevit, Céline Robardet, and Jean-François Boulicaut. Mining graph topological patterns: Finding co-variations among vertex descriptors. *IEEE TKDE*, 99:1, 2013.

[41] Guo-Jun Qi, Charu C. Aggarwal, Qi Tian, Heng Ji, and Thomas S. Huang. Exploring context and content links in social media: A latent space method. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(5):850–862, 2012.

[42] Céline Robardet. Constraint-based pattern mining in dynamic graphs. In *ICDM*, pages 950–955. IEEE, 2009.

[43] Jun Sese, Mio Seki, and Mutsumi Fukuzaki. Mining networks with shared items. In *CIKM*, pages 1681–1684. ACM, 2010.

[44] Arlei Silva, Wagner Meira, and Mohammed J. Zaki. Mining attribute-structure correlated patterns in large attributed graphs. *PVLDB*, 5(5):466–477, 2012.

[45] 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.

[46] Yizhou Sun and Jiawei Han. *Mining Heterogeneous Information Networks: Principles and Methodologies*. Morgan & Claypool Publishers, 2012.

[47] T. L. Taylor. *Raising the Stakes:E-Sports and the Professionalization of Computer Gaming*. MIT Press, 2012.

[48] Hanghang Tong, Spiros Papadimitriou, Jimeng Sun, Philip S. Yu, and Christos Faloutsos. Colibri: fast mining of large static and dynamic graphs. In *KDD*, pages 686–694, 2008.

[49] Matthijs van Leeuwen. Maximal exceptions with minimal descriptions. *Data Min. Knowl. Discov.*, 21(2):259–276, 2010.

[50] Matthijs van Leeuwen and Arno J. Knobbe. Diverse subgroup set discovery. *Data Min. Knowl. Discov.*, 25(2):208–242, 2012.

[51] Arthur Von Eschen. Machine learning and data mining in call of duty (invited talk). In *ECML/PKDD*, 2014.

[52] Yajun Yang, JeffreyXu Yu, Hong Gao, Jian Pei, and Jianzhong Li. Mining most frequently changing component in evolving graphs. *WWW*, pages 1–26, 2013.

[53] Chang Hun You, Lawrence B. Holder, and Diane J. Cook. Learning patterns in the dynamics of biological networks. In *KDD*, pages 977–986, 2009.

Algorithm 1: COSMIc

Input: $C = (C_1, \cdots, C_p)$, G_C, $G = (V, E, T, \text{EDGE})$ with the transactions of T in the relation R of schema $S_R = [R_1, \ldots, R_p]$, i the attribute index to be enumerated

Output: $\mathcal{CS}$ the set of *exceptional contextual subgraph* patterns under construction

begin
  if $(i = p)$ **then**
    for $CC_C \in G_C$ **do**
      if *CheckConstraints*(C, CC_C) **then**
        $\mathcal{CS} \leftarrow \mathcal{CS} \cup (C, CC_C)$
  else
    if *(R_i is symbolic)* **then**
      for $a \in \textbf{\textit{dom}}(R_i) \cup \{\star_i\}$ **do**
        $C' \leftarrow (C_1, \cdots, C_{i-1}, a, C_{i+1}, \cdots, C_p)$
        $G_{C'} \leftarrow M_{C \to G}(C')$
        $F \leftarrow M_{G \to T}(G_{C'})$
        if *($C' = F$)* **then**
          $G_{C'} \leftarrow \texttt{Pruning}(C', G_{C'})$
          if $G_{C'} \neq \emptyset$ **then**
            $\texttt{COSMIc}(C', G_{C'}, G = (V, E, T, \text{EDGE}), i + 1)$
      else
        stack $\leftarrow ([\min_{a_i \in dom(R_i)} a_i, \max_{b_i \in dom(R_i)} b_i], \texttt{true})$
        while *(stack is not empty)* **do**
          $([a, b], \text{left}) \leftarrow \texttt{unstack}(\text{stack})$
          $C' \leftarrow (C_1, \cdots, C_{i-1}, [a, b], C_{i+1}, \cdots, C_p)$
          $G_{C'} \leftarrow M_{C \to G}(C')$
          $F \leftarrow M_{G \to T}(G_{C'})$
          if *($C' = F$)* **then**
            $G'_C \leftarrow \texttt{Pruning}(C', G'_C)$
            if $G_{C'} \neq \emptyset$ **then**
              $\texttt{COSMIc}(C', G_{C'}, G = (V, E, T, \text{EDGE}), i + 1)$
          if *left = true* **then**
            interval $\leftarrow [a, \texttt{previous}(b)]$
            stack $\leftarrow \texttt{push}(\text{interval}, \texttt{true})$
          interval $\leftarrow [\texttt{next}(a), b]$
          stack $\leftarrow \texttt{push}(\text{interval}, \texttt{false})$
return $\mathcal{CS}$

Algorithm 2: `Pruning`

Input: $C = (C_1, \cdots, C_p)$, G_C, $G = (V, E, T, \text{EDGE})$
Output: The pruned graph

begin
  for $CC = (V_{CC}, E_{CC}) \in G_C$ **do**
    for $e \in E_{CC}$ **do**
      if *(*$|\ M_{C\to T}(C, M_{G\to T}(e))| \leq min\_weight$*)* ***or*** *(*$X^2_{ub}(C, e) < \chi^2_{0.05}$*)* **then**
        $E_{CC} \to E_{CC} \setminus \{e\}$
    if *(*$|V_{CC}| \leq min\_vertex\_size$*)* ***or*** *(*$|E_{CC}| < min\_edge\_size$*)* ***or*** *(*$\sum \text{WRAcc}_{ub}(C, E_{CC}) < min\_sum\_wracc$*)* **then**
      $G_C \leftarrow G_C \setminus \{CC\}$
  return G_C

Algorithm 3: Upper bound of $\sum \text{WRAcc}_{ub}$.

Input: E_{CC}, W_C, α and γ
Output: The bound $\sum \text{WRAcc}_{ub}$

begin
  Sort the k edges of E_{CC} in descending order of their weights
  $S_1 \leftarrow W_C(e_1)$
  $S_2 \leftarrow \sum_{i=2}^{k} W_C(e_i)$
  $UB_1 \leftarrow \frac{W_C(e_1)}{\alpha}\left(\frac{S_1+S_2}{W_C(e_1)} - \gamma\right)$
  $j \leftarrow 2$
  repeat
    $S_1 \leftarrow (k-j) \times W_C(e_j)$
    $S_2 \leftarrow S_2 - W_C(e_j)$
    $UB_j \leftarrow \frac{W_C(e_j)}{\alpha}\left(\frac{S_1+S_2}{W_C(e_j)} - \gamma\right)$
    $j \leftarrow j + 1$
  until *(*$UB_j < UB_{j-1}$*) or (*$j = k$*)*;
return UB_{j-1}

Parameter	**Description**	**Default value**
$nbVertices$	number of vertices	10^4
$nbTrans$	number of transactions	3×10^6
$nbAtt$	number of nominal attributes	5
$domain_{size}$	avg. size of attribute domains	20
$nbPatterns$	number of hidden patterns	5
$patternSize$	avg. number of vertices involved in a hidden pattern	10
$linkProb$	probability of two vertices to be linked	0.2
$weight$	avg. weight of contextual edges in hidden patterns	10
$noiseRate$	probability of a transaction supporting the context to be noisy	0.1

Table 3: Default parameters used for generating data.

Graph layer modeling 1					Graph layer modeling 2		
Gender	Age	Time	Weather	$\max_{x\in E} W_C(x)$	Attribute	Value	$\max_{x\in E} W_C(x)$
F	20	Day	Rainy	1	Gender	F	3
F	20	Night	Cloudy	1	Gender	M	2
F	20	Night	Windy	1	Age	20	3
F	20	Night	Rainy	1	Age	23	1
M	23	Night	Windy	1	Age	30	1
M	23	Night	Cloudy	1	Age	45	2
M	23	Night	Rainy	1	Age	50	1
F	45	Night	Cloudy	1	Time	Day	2
F	45	Day	Rainy	1	Time	Night	4
F	45	Night	Windy	1	Weather	Cloudy	2
M	50	Day	Windy	1	Weather	Rainy	2
M	50	Night	Rainy	1	Weather	Windy	2
F	30	Night	Rainy	1			

Table 4: Potential graph layers and maximum weights of edges for each layer.

Context	Relative Weight	WRAcc
$\langle F, 20, Night, Rainy\rangle$	1/1	$\frac{1}{5}\left(\frac{1}{1}-\frac{5}{5}\right)$
Gender='F'	1/3	$\frac{3}{5}\left(\frac{1}{3}-\frac{2}{5}\right)$
Age=20	1/3	$\frac{3}{5}\left(\frac{1}{3}-\frac{2}{5}\right)$
Time='Night'	2/4	$\frac{4}{5}\left(\frac{2}{4}-\frac{2}{5}\right)$
Weather='Rainy'	1/2	$\frac{2}{5}\left(\frac{1}{2}-\frac{2}{5}\right)$

Table 5: Possible weight encodings of edge (C,D) for an example context and its component attribute-value pairs.