Mining Subjectively Interesting Attributed Subgraphs

Work-in-progress paper

Anes Bendimerad[*]
Univ Lyon, INSA, CNRS UMR 5205
F-69621 France
aabendim@liris.cnrs.fr

Ahmad Mel[*]
IDLab, Ghent University
Ghent, Belgium
ahmad.mel@ugent.be

Jefrey Lijffijt
IDLab, Ghent University
Ghent, Belgium
jefrey.lijffijt@ugent.be

Marc Plantevit
Univ Lyon, UCBL, CNRS UMR 5205
F-69622 France
marc.plantevit@liris.cnrs.fr

Céline Robardet
Univ Lyon, INSA, CNRS UMR 5205
F-69621 France
celine.robardet@liris.cnrs.fr

Tijl De Bie
IDLab, Ghent University
Ghent, Belgium
tijl.debie@ugent.be

ABSTRACT

Community detection in graphs, data clustering, and local pattern mining are three mature fields of data mining and machine learning. In recent years, attributed subgraph mining is emerging as a new powerful data mining task in the intersection of these areas. Given a graph and a set of attributes for each vertex, attributed subgraph mining aims to find cohesive subgraphs for which (a subset of) the attribute values has exceptional values in some sense. While research on this task can borrow from the three abovementioned fields, the principled integration of graph and attribute data poses two challenges: the definition of a pattern language that is intuitive and lends itself to efficient search strategies, and the formalization of the interestingness of such patterns. We propose an integrated solution to both of these challenges. The proposed pattern language improves upon prior work in being both highly flexible and intuitive. We show how an effective and principled algorithm can enumerate patterns of this language. The proposed approach for quantifying interestingness of patterns of this language is rooted in information theory, and is able to account for prior knowledge on the data. Prior work typically quantifies interestingness based on the cohesion of the subgraph and for the exceptionality of its attributes separately, combining these in a parameterized trade-off. Instead, in our proposal this trade-off is implicitly handled in a principled, parameter-free manner. Extensive empirical results confirm the proposed pattern syntax is intuitive, and the interestingness measure aligns well with actual subjective interestingness.

KEYWORDS

Graphs, Networks, Subjective Interestingness, Subgraph Mining

1 INTRODUCTION

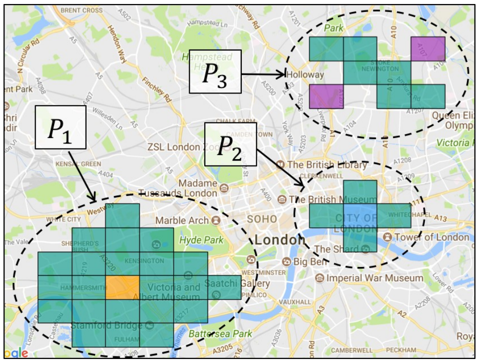
The availability of network data has surged both due to the success of social media and ground-breaking discoveries in experimental sciences. Consequently, graph mining is one of the most studied tasks for the data mining community. The value of graphs stems from the presence of meaningful relationships among the data objects (the vertices). These can be explored by approaches as different as graph embeddings[7]—which map the nodes of a graph into a low dimensional space while preserving the local and global graph structure as well as possible—, community detection[9]—the discovery of groups of vertices that somehow 'belong together'—, or subgraph mining—the identification of informative subgraphs.

Besides the relational structure, graphs may carry information in the form of attribute-value pairs on vertices and/or edges. Such graphs are called attributed graphs[16, 18, 20]. We focus here on graphs with attribute-value pairs on vertices (vertex-attributed graphs). Mining interesting subgraphs in attributed graphs is challenging, both conceptually and computationally. A prominent problem is defining the interestingness of a subgraph. Desirable properties of subgraph would be that it is cohesive (the attribute values of the vertices are similar) and that the vertices form an easy to describe pattern in the graph (e.g., vertices should be close to each other). A specific form of subgraph mining is to look for exceptional subgraphs, i.e., subgraphs whose attribute values are cohesive within the subgraph but exceptional in the full graph.

Few works in this direction exist. For example Atzmueller et al.[1] study mining communities (densely connected subgraphs) that can also be described well in terms of attribute values, while Bendimerad et al. [2] look for exceptional subgraphs that are connected. Various quality measures are used in the first work and the second relies on Weighted Relative Accuracy. We introduce a new generically applicable interestingness measure for exceptional subgraph patterns based on Information Theory which is more flexible and can incorporate prior knowledge about the graph to steer the scoring of subgraph patterns. Besides, while previous works use certain hard constraints to arrive at subgraphs that are somehow interpretable, we integrate the interpretability into the interestingness measure. Hence, the trade-off between informativeness and interpretability can be made in a principled manner.

More specifically, we consider the problem to identify informative subgraphs that can be concisely described. The informativeness of a subgraph depends on the number of vertices it covers (more is better) and how surprising the statistics (attribute values) of those vertices are. Surprise is important because showing the user statistics they expect to see does not teach them anything. Vertices that are spread out over the graph, or that share no statistics cannot be summarized well, so the end-user cannot generalize over the structure of the vertices in a pattern and hence the graph structure is effectively transmitted to a user without any compression. Here instead, we look for subgraphs that are both homogeneous and

[*] The first two authors contributed equally to the paper.



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

Figure 1: Example results on a graph based on Foursquare data covering the presence of various types of venues in London. P_1: around the orange block (west/south of Hyde Park) there are 'surprisingly' many food establishments, except in the centre of that area (which is average). P_2: in the City several types of venues are consistently overrepresented. P_3: around Hackney there is a strip of blocks with lots of nightlife and food venues but limited educational venues.

localized, hence possess shared properties which can be exploited to produce concise descriptions.

Fig. 1 shows example patterns from our method, applied to a setting where we want to explore the district structure of cities. The patterns should be interpreted as follows: certain attributes have surprisingly high/low values (marked +/− respectively) in the given neighbourhoods as compared to a background model. We ensure the regions are localized by forcing them to have a description of the form *"all vertices that are within distance d_1 of vertex x AND within distance d_2 of vertex y AND etc."*; e.g., in pattern P_3 of Fig. 1, the covered areas (green blocks) are the intersection of blocks within distance two of either purple block.

Contributions. We present a pattern syntax for cohesive subgraphs with exceptional attributes (Sec. 2). We formalize their subjective interestingness in a principled manner using information theory, accounting for both the information they provide, as well as their interpretability (Sec. 3). We study how to mine such subgraphs efficiently (Sec. 4). We provide a thorough empirical study on real data that evaluates (1) the relevance of the subjective interestingness measure compared to state-of-the-art methods, and (2) the efficiency of the algorithms (Sec. 5). We discuss related work in Sec. 6 and the conclusions are presented in Sec. 7.

2 COHESIVE SUBGRAPHS WITH EXCEPTIONAL ATTRIBUTES

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

Notation An *attributed graph* is denoted $G = (V, E, \hat{A})$, where V is a set of n vertices, $E \subseteq V \times V$ is a set of m edges, and $\hat{A}$ is a set of p numerical attributes on vertices (formally, functions mapping a vertex onto an attribute value), with $\hat{a}(v) \in \text{Dom}_a$ denoting the value of attribute $\hat{a} \in \hat{A}$ on $v \in V$. We use hats in $\hat{a}$ and $\hat{A}$ to signify the empirical values of the attributes, whereas a and A denote (possibly random) variables over the same domains. We also define the function $N_d(v)$ to denote the neighborhood of range d of a vertex v, i.e., the set of vertices whose geodesic distance to v is at most d:

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

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

Thus, and more formally, a *CSEA pattern* is defined as a tuple (U, S), where $U \subseteq V$ is a set of vertices in the graph, and S is a set of restrictions on the value domains of the attributes of A, or more specifically, $S = \{[k_a, \ell_a] \mid a \in A\}$. A pattern (U, S) is said to be contained in G iff

$$\forall [k_a, \ell_a] \in S \text{ and } \forall u \in U,\ k_a \le \hat{a}(u) \le \ell_a. \tag{1}$$

Informally speaking, a CSEA pattern will be more informative if the ranges in S are smaller, as then it conveys more information to the data analyst. We will make this more formal in Section 3.1.

At the same time, a CSEA pattern (U, S) will be more interesting if its description is more concise in some *natural easier-to-interpret definition*. Thus, along with the pattern language, we must also specify how a pattern from this language will be intuitively described.

To this end, we propose to describe the set of vertices U as a neighborhood of a specified range from a given specified vertex, or more generally as the intersection of a set of such neighborhoods. For enhanced expressive power, we additionally allow for the description to specify some exceptions on the above: vertices that do fall within this (intersection of) neighborhood(s), but which are to be excluded from U. Exceptions are a detriment to the interestingness of a pattern, but we can discount these naturally.

A premise of this paper is that this way of describing the set U is intuitive for human analysts, such that the length of the description of a pattern, as discussed in detail in Sec. 3.2, is a good measure of the complexity to assimilate or understand it. Our qualitative experiments in Sec. 5 do indeed confirm this is the case.

3 THE SUBJECTIVE INTERESTINGNESS OF A CSEA PATTERN

The previous sections already hinted at the fact that we will formalize the interestingness of a CSEA pattern (U, S) by trading off its information content with its description length. Here we show how the FORSIED framework for formalizing subjective interestingness of patterns, introduced in [3, 4], can be used for this purpose.

The information content depends on both U and S. It is larger when more vertices are involved, when the intervals are narrower, and when they are more extreme. We will henceforth denote the information content as IC(U, S). The description length depends on U only (as the attribute ranges require a fixed description length), and will be denoted as DL(U). The subjective interestingness of a

CSEA pattern (U, S) is then expressed as:

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

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

This is achieved by modeling the prior beliefs of the user as the Maximum Entropy (MaxEnt) distribution subject to any stated prior beliefs the user may hold about the data. This distribution is referred to as the *background distribution*. The information content $\mathrm{IC}(U, S)$ of a CSEA pattern (U, S) is then formalized as minus the logarithm of the probability that the pattern is present under the background distribution (also called the self-information or surprisal) [8]:

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

In Sec. 3.1, we first discuss in greater detail which prior beliefs could be appropriate for CSEA patterns, and how to infer the corresponding background distribution. Then, in Sec. 3.2, we discuss in detail how the description length $\mathrm{DL}(U)$ can be computed.

3.1 The information content of a CSEA pattern

Positive integers as attributes For concreteness, let us consider the situation where the attributes are positive integers ($a : V \to \mathbb{N}, \forall a \in A$), as will be our main focus throughout this paper.[1] For example, if the vertices are geographical regions (with edges connecting vertices of neighboring regions), then the attributes could be counts of particular types of places in the region (e.g. one attribute could be the number of shops). It is clear that it is less informative to know that an attribute value is large in a large region than it would be in a small region. Similarly, a large value for an attribute that is generally large is less informative than if it were generally small. The above is only true, however, if the user knows (or believes) *a priori* at least approximately what these averages are for each attribute, and what the 'size' of each region is. Such prior beliefs can be formalized as equality constraints on the values of the attributes A on all vertices, or mathematically:

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

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

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

As shown in [4], the optimal solution of this optimization problem is a product of independent Geometric distributions, one for each vertex attribute-value $a(v)$. Each of these Geometric distributions is of the form $\Pr(a(v) = z) = p_{av} \cdot (1 - p_{av})^z, z \in \mathbb{N}$, where p_{av} is the success probability and it is given by: $p_{av} = 1 - \exp(\lambda_a^r + \lambda_v^c)$, with λ_a^r and λ_v^c the Lagrange multipliers corresponding to the two constraint types. The optimal values of these multipliers can be found by solving the convex Lagrange dual optimization problem.

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

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

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

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

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

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

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

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

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

$$\begin{aligned}\mathrm{IC}(U, S) &= -\log(\Pr(U, S)), \\ &= -\sum_{[k_{c_a}, \ell_{c_a}] \in S} \sum_{v \in U} \log(\ell_{c_a} - k_{c_a} + p_{av} k_{c_a}).\end{aligned} \tag{2}$$

In this paper, we focus on intervals $[k_{c_a}, \ell_{c_a}]$ where either $k_{c_a} = 0$ (the minimal value) or $\ell_{c_a} = 1$ (the maximal value). Such intervals state that the values of an attribute are all significantly large[2] or significantly small respectively, for all vertices in U. We argue such

[1] The presented results can be extended relatively straighforwardly for other cases.

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

```
Algorithm 1: SIAS-Miner-Enum($G = (V, E, \hat{A})$, $D$)
  Input: $G$ the input graph, $D$ the maximum threshold of $d$
         for used neighbourhoods $N_d(v)$.
  Output: $Result$, the set of CSEAs, and $minDesc$, which
          stores the minimum description for each pattern.
  // transformation to entity-relation model
  $\mathbb{D} \leftarrow$ transformToER($G$,$D$)
  // enumeration of the patterns
  Result $\leftarrow$P-N-RMiner($\mathbb{D}$)
  // computation of the minimum description for each found pattern
  minDesc $\leftarrow$ {}
  for $\langle(U, S), \mathcal{N}(U)\rangle \in Result$ do
      bestDesc$\leftarrow \emptyset$
      DL-Optimise($U$, $\emptyset$,$\mathcal{N}(U)$, bestDesc)
      minDesc[(U,S)] $\leftarrow$ bestDesc
```

intervals are easiest to interpret. The logarithmic terms in Eq. (2) then simplify to $\log(\ell_{c_a})$ and $\log(1 - k_{c_a} + p_{av} k_{c_a})$ respectively.

3.2 Description length

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

$$f(X, U) = (|X| + 1) \cdot \log(|\mathcal{N}|) + (|\text{exc}(X, U)| + 1) \cdot \log(|V|).$$

Indeed, the first term accounts for the description of the number of neighborhoods ($\log(|\mathcal{N}|)$, as there can be no more than $|\mathcal{N}|$ neighborhoods in X, and for describing which neighborhoods are involved ($|X| \log(|\mathcal{N}|)$). The second term accounts for the description of the number of exceptions ($\log(|V|)$), and for describing the exceptions themselves ($|\text{exc}(X, U) \log(|V|)$).

In general, there are multiple ways of describing a given set of nodes U, by using a various combinations of neighborhoods. The best one is thus the one that minimizes f. This finally leads us to the definition of the description length of a pattern as:

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

4 AN ENUMERATION APPROACH TO MINING INTERESTING CSEA PATTERNS

SIAS-Miner-Enum mines interesting patterns using an enumerate-and-rank approach. First, it enumerates all CSEA patterns (U, S) that are closed simultaneously wrt. U, S, and the neighbourhood description. Second, it ranks patterns according to their SI values. An overview of the method is given in Algorithm 1 and explained further below.

4.1 Pattern enumeration

In the first step, we enumerate candidate tuples (U, S) where the vertices U can be concisely described as an intersection of neighborhoods $X \subseteq \mathcal{N}(U)$. Disregarding the description for a moment, since the pattern syntax is chosen such that each interval $[k_a, l_a] \in S$ should cover every vertex $u \in U$ (Eq. 1), the IC of a tuple (U, S) increases monotonically by adding vertices to U and intervals to S. Hence, to decrease the number of candidate patterns and increase computational efficiency we focus on *closed* patterns, i.e., tuples (U, S) where no vertex can be added without enlarging intervals and where no interval can be reduced without omitting vertices.

However, in Section 2 we additionally argued that a pattern (U, S) where the vertices $u \in U$ are unrelated will be difficult to understand and remember, which we expressed in the DL. Since computing DL(U) is NP-Complete, it is not clear that enumeration of closed patterns wrt. the SI can be done. Nonetheless, enumeration of all closed (U, S) appears wasteful because most sets U will have a high DL. What appears feasible is to restrict enumeration of sets U that are exactly intersections of neighborhoods, i.e., a description without any exceptions[3].

While closed sets (U, S) may be most efficiently enumerated by an itemset mining algorithm, if we want (U, S) to additionally be closed with respect to intersections of neighbourhoods this yields a relational schema with two relations: (1) vertices are connected to all intervals that cover their attribute values, and (2) vertices are connected to every neighborhood that they are contained in. A tool that would indeed enumerate precisely the required closed patterns and no other is RMiner [21].

More formally, the mapping from a graph to the required relational format is depicted in Fig. 2. Any vertex-attributed graph G can be mapped to an entity-relational model $\mathbb{D}$ through (1) creation of an entity type E_v containing all vertices in G, (2) creation of $|\hat{A}|$ entity types $E_{a1}, E_{a2}, \dots E_{a|\hat{A}|}$, one per attribute, and (3) creation of an entity type E_N containing all neighborhoods. Fig. 2 shows how intervals and neighborhoods form a hierarchy. For neighborhoods, the relationship holds that if a vertex v_i is contained in a neighborhood $N_k(v_j)$, then it is also contained in all neighborhoods with larger hop-size: $v_i \in N_k(v_j) \Rightarrow v_i \in N_l(v_j)\ \forall l \geq k$. A similar statement holds for intervals.

P-N-RMiner [15] is an extension of RMiner that exploits such hierarchies for efficiency. Firstly, fewer connections (edges) are needed between the entity types hence there is a smaller memory requirement. Secondly, there may be computational gains: P-N-RMiner is based on fixpoint-enumeration [6], whose theory states that efficient enumeration of closed sets is possible if and only if the problem can be cast as a strongly accessible set system $(E, \mathcal{F})$.

The efficiency gain over plain enumeration comes from a closure operator, which can skip non-closed candidate patterns. For P-N-RMiner, this closure works outside-in, i.e., closed patterns are found by considering whether any entity (vertex in the relational representation) can be added without reducing the set of entities that are currently valid extensions to form a pattern under the pattern syntax. Patterns in (P-N-)RMiner are called complete connected subsets (CCSs) because all possible edges must exist and the vertices in the relational representation must be connected, see Fig. 2,

[3] Notice that such a description does not necessarily minimize the description length of a vertex set U.

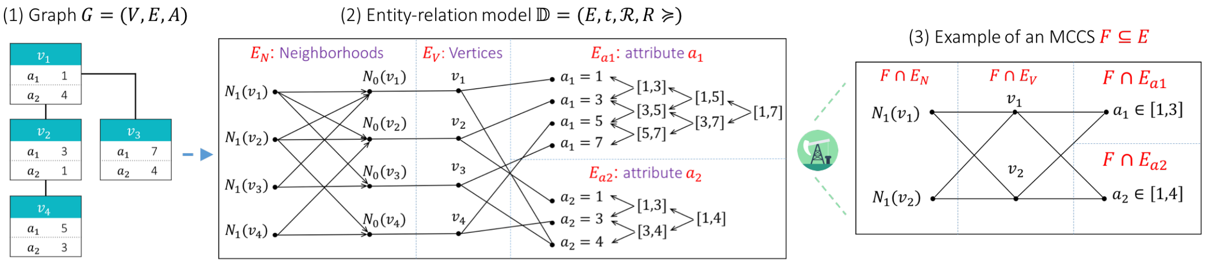


Figure 2: Transformation from (1) a graph structure to (2) an entity-relation model with $D = 1$, and (3) an example of a maximal complete connected subset (MCCS) pattern from P-N-RMiner.

right for an example. In the relational context, a pattern is called a maximal CCS (MCCS) if no entity can be added, i.e., patterns we referred to as closed in the discussion above. It is worth to notice that an MCCS provides, in addition to a tuple (U, S), the set $\mathcal{N}(U)$ of neighborhoods that contain U, from which $DL(U)$ is computed.

Notice P-N-RMiner also ranks patterns based on interestingness under a known-degree background model, but that is not useful in our setting as the IC and the DL are very different here. Hence, we only use it to enumerate all candidates. The computational complexity is clearly exponential as the number of outputs may be exponential in the size of the input, plus in this setting no fixpoint-enumeration-based algorithm may have polynomial-time delay [15]. Scalability experiments are presented in Sec. 5.

4.2 Computing $DL(U)$

The calculation of $DL(U)$ is NP-Complete and equivalent to Set Cover: it consists in finding the optimal cover of the set $\overline{U}$ based on unions of complements $\overline{N_i(v)}$ and exceptions $\{v\}$ such that $x \in \overline{U}$. Nevertheless, we propose a branch-and-bound approach that takes benefit from several optimisation techniques.

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

Function LB (line 1) lower bounds the lengths of the descriptions that can be generated in the subsequent recursive calls of DL-Optimise. If *LB* is higher or equal than the length of the current best description of U $f(\text{bestDesc}, U)$, there is no need to carry on the exploration of the search subspace as no further description can improve f. The principle of *LB* is to evaluate the maximum reduction in exceptions that can be obtained when description X is extended with neighbourhoods of Y:

$$gain_Y(X, U) = |\text{exc}(X, U)| - |\text{exc}(X \cup Y, U)|, \text{ with } Y \subseteq \text{Cand}. \quad (3)$$

Algorithm 2: DL-Optimise(U, X, Cand, bestDesc)

```
Input: $U$ the set of vertices to describe, $X$ the current
       enumerated description, Cand the set of candidates,
       bestDesc the current best description found.
Output: bestDesc the best description found while
        exploring the current search sub-space.
1 if $LB(X, U, Cand) < f(bestDesc, U)$ then
2     if $Cand \neq \emptyset$ then
3         pruneUseless($U$, $X$, Cand)
4         pruneLowerBounded($U$, $X$, Cand)
5         $e \leftarrow \text{argmin}_{e' \in \text{Cand}} f(X \cup \{e'\}, U)$
6         DL-Optimise($U$, $X \cup \{e\}$, Cand \ $\{e\}$, bestDesc)
7         DL-Optimise($U$, $X$, Cand \ $\{e\}$, $bestDesc$)
8     else if $f(X, U) < f(bestDesc, U)$ then
9         bestDesc $\leftarrow X$
```

This function can be rewritten using neighbourhood complements as $gain_Y(X, U) = |\cup_{y \in Y} (\overline{y} \cap \text{exc}(X, U))|$ [4]. We can obtain an upper bound of the gain function using the ordered set $\{g_1, \ldots, g_{|\text{Cand}|}\}$ of $\{gain_{\{e\}}(X, U) \mid e \in \text{Cand}\}$ such that $g_i \geq g_j$ if $i \leq j$:

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

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

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

$$LB(X, U, \text{Cand}) = \min_{i \in [\![0, |\text{Cand}|]\!]} \{(|X| + i + 1) \times \log(|\mathcal{N}|) + \left(1 + \max\left(0, |\text{exc}(X, U)| - \textstyle\sum_{j=1}^{i} g_i\right)\right) \times \log(|V|)\} \quad (4)$$

Property 2. *$f(X \cup Y, U) \geq LB(X, U, Cand)$, for all $Y \subseteq Cand$.*

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

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

```
Algorithm 3: pruneUseless($U$, $X$, Cand)
Cand ← $\{e \in \text{Cand} \mid \text{gain}(\{e\}, X, U) > 0\}$
```

```
Algorithm 4: pruneLowerBounded($U$, $X$,Cand)
Cand ← $\{e_i \in \text{Cand} \mid \forall e_j \in \text{Cand} \setminus \{e_i\} : (\text{exc}(X \cup \{e_j\}, U) \not\subseteq \text{exc}(X \cup \{e_i\}, U))) \vee (\text{exc}(X \cup \{e_j\}, U) = \text{exc}(X \cup \{e_i\}, U) \wedge i < j)\}$
```

In other terms, in the recursive calls, a description length will never be lower than $LB(X, U, \text{Cand})$.

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

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

Property 3. *Let $e, e' \in Cand$ such that $exc(X \cup \{e\}, U) \subseteq exc(X \cup \{e'\}, U)$. Then, for all $Y \subseteq Cand \setminus \{e, e'\}$, we have $f(X \cup Y \cup \{e\}, U) \leq f(X \cup Y \cup \{e'\}, U)$*

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

Based on Property 3, pruneLowerBounded removes elements $e' \in \text{Cand}$ such that $\text{exc}(X \cup \{e\}, U) \subseteq \text{exc}(X \cup \{e'\}, U)$. Notice that even if an element e'' has been removed due to the lower bound of e', the procedure is still correct since e'' is lower bound by e by the transitivity of inclusion.

The last optimisation consists in choosing $e \in \text{Cand}$ that minimises $f(X \cup \{e\}, U)$ (line 5 of Algorithm 2). This makes it possible to quickly reach descriptions with low DL, and subsequently provide effective pruning when used in combination with LB.

5 EXPERIMENTS

In this section, we report our experimental results. We start by describing the real-world dataset we used, as well as the questions we aim to answer. Then, we provide a thorough comparison with the state-of-the-art algorithm Cenergetics [2]. Eventually, we provide a qualitative analysis that demonstrates the ability of our approach to achieve the desired goal. For reproducibility purposes, the source code and the data are made available here.[5]

Experimental setting. Experiments are performed on the real-world dataset of the London graph. The *London graph* ($|V| = 289$, $|E| = 544$, $|\hat{A}| = 10$) is based on the social network Foursquare[6]. Each vertex represents a district in London, and edges link adjacent districts. Each attribute stands for the number of places of a given type (e.g. outdoors, colleges, residences, restaurants, etc.) in each district. Considering all the numerical values of attributes is computationally expensive and would lead to redundant results, we pre-process the graph so that for each attribute, the values $\hat{c}_a(v)$ are binned into five quantiles.

Aims. As stated in Section 6, there is no approach that supports the discovery of subjectively interesting attributed subgraphs in the literature. The closest method to SIAS-Miner-Enum is Cenergetics [2] that aims at discovering closed exceptional attributed subgraphs involving overrepresented and/or underrepresented attributes, and which mined the London graph used here in the experiments (and on similar graphs of other cities). It assesses exceptionality with the weighted relative accuracy (WRAcc) measure that accounts for margins but cannot account for other prior knowledge. The computational problem we tackle is more complex than Cenergetics, but how much is this overhead? Is it worth it in terms of pattern quality? This empirical study aims to answer to these questions.

Quantitative experiments. Fig. 3 reports the execution time per pattern, the number of discovered patterns, and the average quality of the top 500 patterns (i.e., DL and SI) of SIAS-Miner-Enum and Cenergetics according to the number of vertices, the number of attributes and the minimum number of vertices of searched patterns. We post-processed the results of Cenergetics in order to obtain similar redescriptions of the vertices as in SIAS-Miner-Enum, but we do not consider this post-processing step in the reported execution times. These tests reveal that the computational overhead of SIAS-Miner-Enum is important: the discovery of a pattern by SIAS-Miner-Enum is generally one to two orders of magnitude more costly than Cenergetics. However, SIAS-Miner-Enum provides patterns of better quality. Indeed, the average description length of the top 500 patterns discovered by SIAS-Miner-Enum is smaller than those of Cenergetics and the SI of CSEA patterns is greater than the one of the patterns extracted by Cenergetics.

Qualitative experiments. Finally, we show some examples of patterns (with at least 5 vertices) discovered by both SIAS-Miner-Enum and Cenergetics on London graph. The top 4 CSEA patterns discovered by SIAS-Miner-Enum are given in Tab. 1 and displayed in Fig. 4. Green cells represent vertices covered by a CSEA pattern while blue cells are the centers, purples cells are the centers that do not belong to the pattern, orange cells are centers that are also exception (i.e., behave differently from the pattern but covered by the description) and the red cells are normal exceptions. We also report the top 4 patterns discovered by Cenergetics in Fig. 5. Interestingly, CSEA patterns are more cohesive than Cenergetics ones, described by at most two Neighborhoods, which eases the assimilation by an analyst. Unexpectedly, the second best patterns of SIAS-Miner-Enum and Cenergetics are somewhat similar: the CSEA pattern covers two additional vertices and the two patterns are in agreement on some overrepresented types of venues (e.g., nightlife, shops). Surprisingly, outdoor venues are consistently overrepresented in the CSEA pattern while such venues are considered as limited by Cenergetics. This inconsistency may be due to some extreme values in some regions that impact the mean and then the value of the WRacc measure by Cenergetics.

Summary. Even if our approach has an obvious computational overhead compared to Cenergetics (the problem tackled is more complex), these experiments show the ability of SIAS-Miner-Enum to discover CSEA patterns that are more intuitive and informative.

[5] https://goo.gl/2jvE8j [6] https://foursquare.com

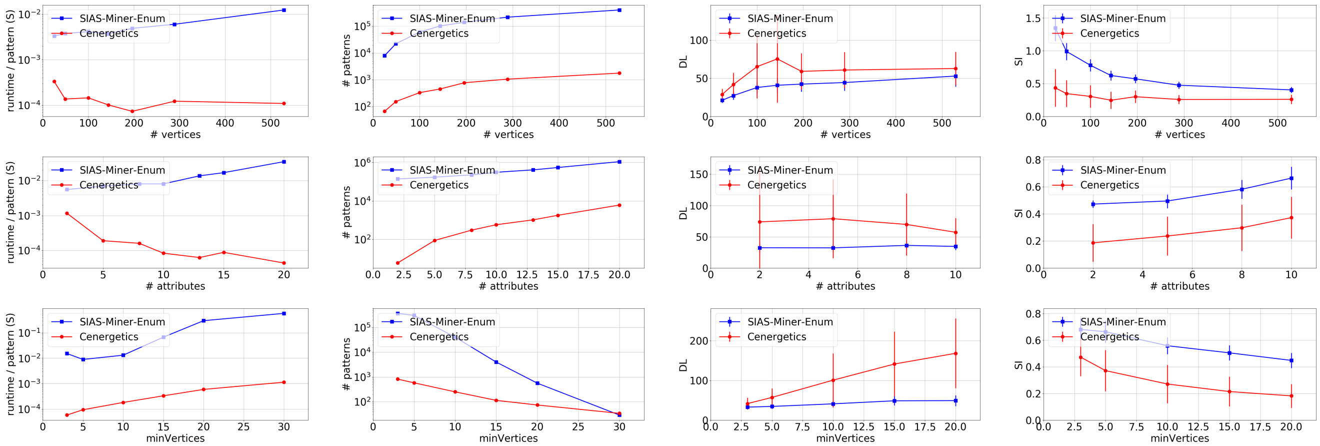


Figure 3: SIAS-Miner-Enum vs Cenergetics: runtime per pattern (first column), #patterns (second column), average description length (third column) and subjective interestingness (fourth column) of the top 500 patterns for varying $|V|$ (1st row), $|A|$ (2nd row) and a threshold on the minimum number of vertices in searched patterns (3rd row) for London graph ($D = 3$).

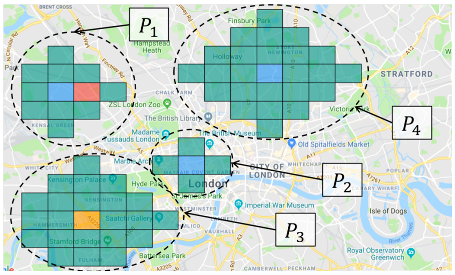
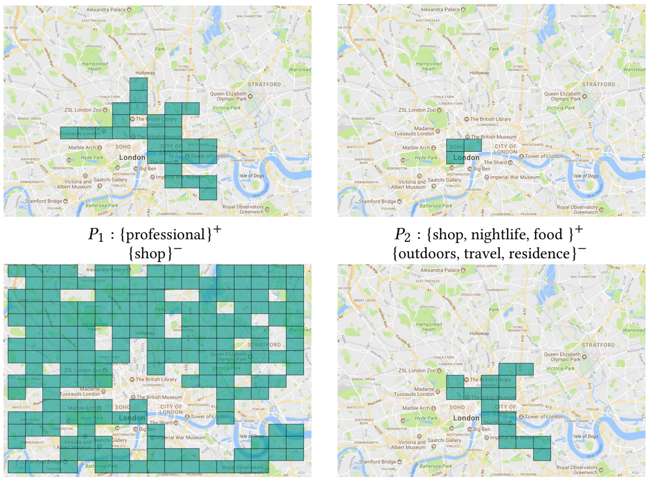


Figure 4: Top 4 patterns discovered in London graph by SIAS-Miner-Enum ($minVertices = 5$, $D = 3$). Details are provided in Tab. 1

Pattern ID	Characteristics: $S = \{(a_i, [l_i, k_i])\}$
P_1	{food: $[0, 0.47]\}^+$, {college: $[1, 1]$, event: $[1, 1]$, art: $[0.57, 1]\}^-$
P_2	{shop: $[0, 0.43]$, nightlife: $[0, 0.44]$, travel: $[0, 0.44]$, college: $[0, 0.47]$, outdoors: $[0, 0.47]\}^+$
P_3	{food: $[0, 0.31]\}^+$
P_4	{food: $[0, 0.47]\}^+$

Table 1: Detailed characteristics of the top 6 patterns discovered in London dataset by SIAS-Miner-Enum (see Fig. 4).



P_1 : {professional}$^+$ {shop}$^-$

P_2 : {shop, nightlife, food }$^+$ {outdoors, travel, residence}$^-$

P_3 : {professional}$^-$

P_4 : {professional}$^+$, {shop, food}$^-$

Figure 5: Top 4 patterns discovered in London graph by Cenergetics ($minVertices = 5$).

6 RELATED WORK

Several approaches have been designed to discover new insights in vertex attributed graphs. The pioneering work of Moser et al. [16] presents a method to mine dense homogeneous subgraphs, i.e., subgraphs whose vertices share a large set of attributes. Similarly, Günnemann et al. [10] introduce a method based on subspace clustering and dense subgraph mining to extract non redundant subgraphs that are homogeneous with respect to the vertex attributes. Silva et al. [20] extract pairs made of a dense subgraph and a Boolean attribute set such that the Boolean attributes are strongly associated with the dense subgraphs. In [18], the authors propose to mine the graph topology of a large attributed graph by finding regularities among numerical vertex descriptors. The main objective of all these approaches is to find regularities instead of peculiarities within a large graph, whereas *Exceptional Subgraph Mining* mines subgraphs with distinguishing characteristics.

Interestingly, a recent work [1] proposes to mine descriptions of communities from vertex attributes, with a Subgroup Discovery approach. In this supervised setting, each community is treated as a target that can be assessed by well-established measures, like WRAcc. In [12], the authors aim at discovering contextualized subgraphs that are exceptional with respect to a model of the data. Restrictions on the attributes, that are associated to edges, are used to generate subgraphs. Such patterns are of interest if they pass a statistical test and have high value on an adapted WRAcc measure. Similarly, [14] proposes to discover subgroups with exceptional transition behavior as assessed by a first-order Markov chain model. The problem of exceptional subgraph mining in attributed graphs was introduced in [2]. Based on an adaptation of WRAcc, the method aims to discover subgraphs with homogeneous and exceptional characteristics. In Section 5, we demonstrate that CSEA patterns discovered by SIAS-Miner-Enum are more informative and less complex than patterns discovered by the algorithm devised in [2].

More generally, Subgroup Discovery [13, 17] aims to find descriptions of sub-populations for which the distribution of a predefined target value is significantly different from the distribution in the whole data. Several quality measures have been defined to assess the interest of a subgroup. The WRAcc is the most commonly used. However, these measures do not take any prior knowledge into account. Therefore, we can expect identified subgroups are less informative. The problem of taking subjective interestingness into account in pattern mining was already identified in [19] and has seen a renewed interest in the last decade.

The interestingness measure employed here is inspired by the FORSIED framework [3, 5], which defines the SI of a pattern as the ratio between the IC and the DL. The IC is the amount of information specified by showing a pattern to the user. The measure is based on the gain from a Maximum Entropy background model that delineates the current knowledge of a user, hence it is *subjective*, i.e., particular to the modeled belief state.

P-N-RMiner [15], the tool used here for pattern enumeration, has also been developed under FORSIED. However, the interestingness measure in this paper is very different, because the information contained in the patterns shown to the user does not align with the output of P-N-RMiner. FORSIED has been applied to mine dense subgraphs [22], but not to Exceptional Subgraph Mining, where we also need to account for attribute values. A much faster CP-based implementation of RMiner exists that directly searches for the top-1 most interesting pattern [11]. However, this tool does not support structured attributes and the interestingness measure is different, hence it could not be used directly in our problem setting.

7 CONCLUSION

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

Acknowledgements. This work was supported by the ERC under the EU's Seventh Framework Programme (FP7/2007-2013) / ERC Grant Agreement no. 615517, FWO (project no. G091017N, G0F9816N), the EU's Horizon 2020 research and innovation programme and the FWO under the Marie Sklodowska-Curie Grant Agreement no. 665501 and Group Image Mining (GIM) which joins researchers of THALES Group and LIRIS Lab. This work is also partially supported by the CNRS project APRC Conf Pap.

REFERENCES

[1] Martin Atzmueller, Stephan Doerfel, and Folke Mitzlaff. Description-oriented community detection using exhaustive subgroup discovery. *Information Science* 329 (2016), 965–984.

[2] Anes Bendimerad, Marc Plantevit, and Céline Robardet. Mining exceptional closed patterns in attributed graphs. *Know. and Inf. Syst.* (2017), 1–25.

[3] Tijl De Bie. 2011. An information theoretic framework for data mining. In *KDD*. 564–572.

[4] Tijl De Bie. Maximum entropy models and subjective interestingness. *Data Mining and Knowledge Discovery* 23, 3 (2011), 407–446.

[5] Tijl De Bie. 2013. Subjective Interestingness in Exploratory Data Mining. In *IDA*. 19–31.

[6] Mario Boley, Tamás Horváth, Axel Poigné, and Stefan Wrobel. Listing closed sets of strongly accessible set systems with applications to data mining. *Theor. Comput. Sci.* 411, 3 (2010), 691–700.

[7] HongYun Cai, Vincent W. Zheng, and Kevin Chen-Chuan Chang. A Comprehensive Survey of Graph Embedding. *CoRR* (2017). arXiv:1709.07604

[8] Thomas M Cover and Joy A Thomas. Entropy, relative entropy and mutual information. *Elements of information theory* 2 (1991), 1–55.

[9] Santo Fortunato. Community detection in graphs. *Phys. rep.* 486 (2010), 75–174.

[10] Stephan Günnemann, Ines Färber, Brigitte Boden, and Thomas Seidl. 2010. Subspace Clustering Meets Dense Subgraph Mining. In *ICDM 2010*. 845–850.

[11] Tias Guns, Achille Aknin, Jefrey Lijffijt, and Tijl De Bie. 2016. Direct Mining of Subjectively Interesting Relational Patterns. In *ICDM*. 913–918.

[12] Mehdi Kaytoue, Marc Plantevit, Albrecht Zimmermann, Anes Bendimerad, and Céline Robardet. Exceptional contextual subgraph mining. *Machine Learning* 106, 8 (2017), 1171–1211.

[13] Nada Lavrac, Branko Kavsek, Peter A. Flach, and Ljupco Todorovski. Subgroup Discovery with CN2-SD. *Jour. of Mach. Lear. Research* 5 (2004), 153–188.

[14] Florian Lemmerich, Martin Becker, Philipp Singer, Denis Helic, Andreas Hotho, and Markus Strohmaier. 2016. Mining Subgroups with Exceptional Transition Behavior. In *KDD*. 965–974.

[15] Jefrey Lijffijt, Eirini Spyropoulou, Bo Kang, and Tijl De Bie. P-N-RMiner: a generic framework for mining interesting structured relational patterns. *I. J. Data Science and Analytics* 1, 1 (2016), 61–76.

[16] Flavia Moser, Recep Colak, Arash Rafiey, and Martin Ester. 2009. Mining Cohesive Patterns from Graphs with Feature Vectors. In *SDM*. 593–604.

[17] Petra Kralj Novak, Nada Lavrac, and Geoffrey I. Webb. Supervised Descriptive Rule Discovery. *Journal of Machine Learning Research* 10 (2009), 377–403.

[18] Adriana Prado, Marc Plantevit, Céline Robardet, and Jean-François Boulicaut. Mining Graph Topological Patterns. *IEEE TKDE*. 25, 9 (2013), 2090–2104.

[19] Abraham Silberschatz and Alexander Tuzhilin. 1995. On Subjective Measures of Interestingness in Knowledge Discovery. In *KDD*. 275–281.

[20] Arlei Silva, Wagner Meira Jr., and Mohammed J. Zaki. Mining Attribute-structure Correlated Patterns in Large Attributed Graphs. *PVLDB* 5, 5 (2012), 466–477.

[21] Eirini Spyropoulou, Tijl De Bie, and Mario Boley. Interesting pattern mining in multi-relational data. *Data Min. Knowl. Discov.* 28, 3 (2014), 808–849.

[22] Matthijs van Leeuwen, Tijl De Bie, Eirini Spyropoulou, and Cédric Mesnage. Subjective interestingness of subgraph patterns. *Machine Learning* (2016), 1–35.