SIAS-miner: mining subjectively interesting attributed subgraphs

Anes Bendimerad¹ · Ahmad Mel³ · Jeffrey Lijffijt³ · Marc Plantevit² · Céline Robardet¹ · Tijl De Bie³

Received: 10 October 2018 / Accepted: 8 November 2019 / Published online: 22 November 2019
© The Author(s) 2019

Abstract

Data clustering, local pattern mining, and community detection in graphs are three mature areas of data mining and machine learning. In recent years, attributed subgraph mining has emerged 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 (some of) the attribute values have exceptional values. The principled integration of graph and attribute data poses two challenges: (1) the *definition of a pattern syntax* (the abstract form of patterns) that is intuitive and lends itself to efficient search, and (2) the *formalization of the interestingness* of such patterns. We propose an integrated solution to both of these challenges. The proposed pattern syntax improves upon prior work in being both highly flexible and intuitive. Plus, we define an effective and principled algorithm to enumerate patterns of this syntax. The proposed approach for quantifying interestingness of these patterns is rooted in information theory, and is able to account for background knowledge on the data. While prior work quantified the interestingness for the cohesion of the subgraph and for the exceptionality of its attributes separately, then combining these in a parameterized trade-off, we instead handle this trade-off implicitly in a principled, parameter-free manner. Empirical results confirm we can efficiently find highly interesting subgraphs.

Responsible editor: M. J. Zaki.

Anes Bendimerad and Ahmad Mel have contributed equally.

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 Skłodowska-Curie Grant Agreement No. 665501, Group Image Mining (GIM) which joins researchers of THALES Group and LIRIS Lab, and by the ACADEMICS grant of the IDEXLYON, project of the Université of Lyon, PIA operated by ANR-16-IDEX-0005.

Extended author information available on the last page of the article

Keywords Graphs · Attributed graphs · Subjective interestingness · Subgraph mining

1 Introduction

The availability of network data has surged both due to the success of social media and data collection efforts in the experimental sciences. Consequently, graph mining is one of the most popular research topics in the data mining community. The value of graphs derives from their ability to represent meaningful relationships between data objects. This data structure can also be extended, by associating attributes to vertices, to also consider object properties. A myriad of methods exists to analyze graphs computationally and find ‘patterns’ in them. At a high level, these methods can be grouped based on whether they are local or global. Global approaches aim at re-configuring the graph by grouping the closest vertices or the most similar ones, such as graph clustering (or community detection) (Fortunato 2010). Local approaches aim at extracting parts of the graph having particular or exceptional characteristics, in view of the graph as a whole (Moser et al. 2009; Silva et al. 2012; Prado et al. 2013; Kaytoue et al. 2017). However, these approaches define subgraphs of interest as elements of a pattern syntax satisfying constraints, for which the disadvantages of the threshold effects are known (Bistarelli and Bonchi 2005). To leverage these problems, we propose to model user beliefs as probabilities and extract subgraphs that are the most informative with respect to this prior knowledge and simple enough to be easily assimilated by the user.

In this paper we introduce a new method, called SIAS-Miner, for the exploratory analysis of vertex-attributed graphs in terms of local patterns. Considering a graph whose vertices are described by a set of count variables—variables that can only take non-negative integer values representing occurrences of an associated phenomenon—SIAS-Miner aims to extract subgraphs that are both cohesive (any shortest path between two subgraph vertices is bounded by a small value, with potentially few exceptions) and whose vertices have exceptional values (unusual large or small values) on a subset of attributes. These patterns consist of connected subgraphs with small diameter characterized by a set of extreme values on some of the attributes. The advantage of such subgraphs is that they can easily be topologically described by a small set of vertices called core vertices and a subgraph radius. These so-called *Cohesive Subgraphs with Exceptional Attributes* (CSEA patterns), are easy-to-understand subgraphs across which a specified subset of the attributes consistently have exceptionally large or small values.

These patterns can be useful in different contexts. For example, in a graph where each vertex represents a block of a city depicted by attributes that indicate the preponderance of different kinds of facilities (outdoor facilities such as parks, food places such as restaurants, colleges, etc.), SIAS-Miner makes it possible to identify city blocks that are geographically close and consistent in terms of service offerings. These findings can then be used to recommend areas to people that move into a new city (Gionis et al. 2015) while wishing to keep the characteristics of their previous neighborhood. In a graph representing a social network (e.g., Twitter), where the vertices are the users, the edges represent their interactions and the attributes measure the prevalence of some

characteristics (such as the hashtags used in their messages), SIAS-Miner identifies groups of users who talk about the same topics while pointing out the key persons. In a graph representing a biological network, such as a protein-protein interaction network with attributes describing their structure, SIAS-Miner may identify the protein characteristics that yield to a strong interaction while identifying the proteins (i.e. genes) that play a central role.

Let us further detail the syntax of CSEA patterns: A CSEA pattern is made of two parts: (1) A subgraph induced by a subset of vertices, hereafter called *cover*. Instead of enumerating the whole set of the cover, which can be laborious and uninformative, the CSEA cover is described as the intersection of several *neighborhoods* with potentially some *exceptions*. Each neighborhood is a geodesic subgraph defined by a vertex, named *core*, and a radius, and it gathers all the vertices of the graph that are at a geodesic distance at most equal to the radius. (2) A set of exceptional attributes and their ranges of values within the vertices of the cover, called *characteristic*. These exceptional attributes are defined by restrictions on their value domains. We are particularly interested in the extreme values compared to the values taken on the other vertices of the graph. When these values are high, we will speak of attribute with exceptionally prevalent values, while for low values, we will speak of attributes with exceptionally non-prevalent values. The *cover* and the *characteristic* are related by the fact that the values of these vertices on the attributes of the characteristic belong to the specified domains.

Figure 1 shows some examples of CSEA patterns returned by SIAS-Miner applied to a geographic network derived from the social network Foursquare. It depicts London's districts with attributes that indicate the prevalence of different kinds of facilities (outdoor facilities such as parks, food places such as restaurants, colleges, etc.). For example, CSEA pattern P_1 covers all the green vertices that are at distance at most 3 of the core vertex v_1 in blue on the figure. In all these vertices but the three vertices in red, nightlife and professional venues are exceptionally prevalent (their number is unexpectedly high). For instance, v_2 is a cover vertex in P_1 , and it is thus covered by the characteristic (the prevalence of nightlife and professional venues). The vertex v_3 is an exception, i.e., although it is a 3-hop neighbor of v_1 , it is not part of P_1 (it is not covered by the characteristic). Pattern P_{14} is defined by the intersection of the 3-hop neighbors of the core vertices v_4 and v_5 in blue on the figure. With one exception (shown on the figure using red), all the vertices have exceptionally prevalent (a large number of) professional venues and exceptionally non-prevalent (few) college and event venues. For each identified pattern P , the core vertices are chosen automatically so that their number and the number of exceptions are minimized.

Once the pattern syntax defined, let us explain how the interestingness of the patterns is evaluated. SIAS-Miner is innovative in using a new flexible interestingness measure for exceptional subgraph patterns based on information theory, using a quantification of informativeness and interpretability. The informativeness of a pattern is a function of the number of vertices in the cover (more is better), the number of attributes in the characteristic (more is better) and the exceptionality of the values for those attributes (also more is better). The exceptionality is quantified with respect to specified background knowledge available about the graph, making the informativeness a subjective measure. This background knowledge is iteratively updated with the new information

ID	Characteristics	
	Exceptionally prevalent	Exceptionally non-prevalent
P_1	nightlife, professional	
P_2	nightlife, food	college
P_3	food	
P_4	food	college
P_{14}	professional	college, event

Fig. 1 Top patterns discovered in the London graph by SIAS-Miner. Green blocks are vertices in the *cover*, blue blocks are *core* vertices that are in the cover, red blocks are *exceptions* (in the neighborhoods but not in the cover). The covers of the top 4 patterns are defined in terms of a single neighborhood with a maximal geodesic radius between two and three, while the cover of the pattern P_{14} is defined as the intersection of two neighborhoods (Color figure online)

acquired by the user during the mining process, which allows SIAS-Miner to identify a set of non-redundant patterns. The interpretability is quantified in terms of the complexity of communicating a pattern, more specifically its description length. The proposed interestingness measure is the ratio of the informativeness and description length, thus representing the information density within the pattern.

Overview of SIAS-Miner Figure 2 illustrates the main steps of the proposed method. (1) SIAS-Miner derives the background model. It represents the user beliefs or expectations about the input graph as probability distributions. (2) It mines and ranks the patterns based on their subjective interestingness SI which is the ratio between the information content IC and the assimilation cost DL (description length). The information content of a pattern is evaluated thanks to the background model. A pattern with a high information content may involve many vertices leading to a high assimilation cost for the user. The description length evaluated on an alternative description of the cover set of vertices measures the ease of pattern assimilation. (3) The best pattern P is displayed to the user and (4) the background model is updated in order to consider

Fig. 2 Overview of SIAS-Miner

P as known by the user. These four steps are repeated as many times as desired. By updating the background model in Step 4, it makes possible to return patterns that contain new piece of information comparing to the already extracted ones and avoid the redundancy problem usually observed with pattern mining approaches.

Outline and contributions We present the CSEA pattern syntax in Sect. 2. We formalize their subjective interestingness in Sect. 3. We explain how to iteratively update the background knowledge during the extraction process in Sect. 4. We study how to mine such subgraphs efficiently in Sect. 5. In Sect. 6, we provide a thorough empirical study on four types of data to evaluate (1) the relevance of the subjective interestingness measure compared to state-of-the-art methods, and (2) the efficiency of the algorithms. We discuss related work in Sect. 7 and present our conclusions in Sect. 8.

2 Cohesive subgraphs with exceptional attributes

In this section we introduce the pattern syntax (the abstract form of the patterns) and argue why patterns of this form are both informative and easy to understand. First, we establish the required notation. Table 1 summarizes the definitions and notation used along the paper.

Notation We assume given a set of vertices V , a set of edges $E \subseteq V \times V$, and a set of count attributes on vertices $\hat{A}$ (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 denote an *attributed graph* as $G = (V, E, \hat{A})$.

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. In other terms, for $a \in A$ and $v \in V$, $a(v)$ is a random variable having the empirical value $\hat{a}(v)$ that effectively happens in G . The user is assumed to know the set of vertices and the connection structure, so V and E always correspond to the actual graph structure.

With $N_d(v)$ we denote the neighborhood of radius d of a vertex v , i.e., the set of vertices whose graph geodesic distance (the number of edges in a shortest path connecting them) to v is at most d :

Table 1 Notation

Symbols	Definitions
$G = (V, E, \hat{A})$	A graph G with V a set of vertices, $E \subseteq V \times V$ a set of edges, and $\hat{A}$ a set of count attributes on vertices. For $\hat{a} \in \hat{A}$ and $v \in V$, $\hat{a}(v) \in \text{Dom}_{\hat{a}}$ denotes the value of attribute $\hat{a}$ on v
A	The set $a \in A$ such that for each $v \in V$, $a(v)$ is a random variable taking as empirical value $\hat{a}(v)$ in G
$N_d(v)$	The geodesic neighborhood of radius d from a vertex v . Formally, $N_d(v) = \{u \in V \mid \text{dist}(v, u) \leq d\}$
(U, S)	A CSEA pattern with $U \subseteq V$ as cover set and $S \subseteq \{(a, [k_a, \ell_a]) \mid a \in A\}$ as characteristic
$\text{Pr}(U, S)$	The probability that the CSEA pattern (U, S) is present in G
$\text{SI}(U, S)$	The Subjective Interestingness of (U, S) , defined as $\text{SI}(U, S) = \frac{\text{IC}(U, S)}{\text{DL}(U, S)}$
$\text{IC}(U, S)$	The Information Content of (U, S) , defined as $\text{IC}(U, S) = -\log(\text{Pr}(U, S))$.
$\text{DL}(U, S)$	The Description Length of (U, S) , defined as $\text{DL}(U, S) = \text{DL}_A(S) + \text{DL}_V(U)$, where $\text{DL}_A(S)$ (resp. $\text{DL}_V(U)$) is the description length of S (resp. U)
$\hat{c}_a(v)$	p -value for $a(v)$ under the background distribution as null hypothesis: $\hat{c}_a(v) \triangleq \text{Pr}(a(v) \geq \hat{a}(v))$
$\mathcal{N}, \mathcal{N}(U)$	$\mathcal{N}$ is the set of all considered neighborhoods $\mathcal{N} = \{N_d(v) \mid v \in V, d \in \mathbb{N} \text{ and } d \leq D\}$ (with D the maximum considered radius). $\mathcal{N}(U)$ is the subset of neighborhoods that contain U : $\mathcal{N}(U) = \{N_d(v) \in \mathcal{N} \mid U \subseteq N_d(v)\}$
$\text{DL}_V(U)$	The description length of $U \subseteq V$. Given a parameter $\alpha \in \mathbb{N}$ used to limit the number of core vertices, $\text{DL}_V(U) = \min_{\substack{X \subseteq \mathcal{N}(U) \\ X \leq \alpha}} f(X, U)$
$f(X, U)$	The length of the description of U by the intersection of neighborhoods $X \subseteq \mathcal{N}(U)$, along with the set of exceptions $\text{exc}(X, U) \triangleq \cap_{N_d(v) \in X} N_d(v) \setminus U$

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

Pattern definition As described in the introduction, we are interested in patterns that inform the user that a set of attributes has exceptional values within a cohesive set of vertices in the graph. To this end, we propose the following syntax.

Definition 1 A *cohesive subgraph with exceptional attributes (CSEA) pattern* is defined as a tuple (U, S) , where $U \subseteq V$ is a set of vertices in the graph that we refer to as the *cover*, and S is a *characteristic* of these vertices, that is to say S is made of restrictions on the value domains of some attributes of A . More specifically, $S \subseteq \{(a, [k_a, \ell_a]) \mid a \in A\}$. Furthermore, to be a CSEA pattern, (U, S) has to be contained in G , i.e.

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

Fig. 3 A toy graph: vertices are geographical areas and their attributes describe the amount of different types of venues they contain. Edges connect the vertices corresponding to neighboring areas

Putting this in words, for every vertex in the cover $u \in U$, the empirical value $\hat{a}(u)$ for attribute a falls within the interval $[k_a, \ell_a]$ specified as a part of the characteristic for the CSEA pattern (U, S) .

Notice that U is not restricted to have any particular structure. Rather, cohesiveness will be promoted through the definition of a description length quantifying the complexity to communicate a particular set of vertices to the user. This is explained below in Sect. 3.3.

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

$$S \leq S' \Leftrightarrow \forall (a, [k'_a, \ell'_a]) \in S' : \\ \exists (a, [k_a, \ell_a]) \in S \text{ with } [k_a, \ell_a] \subseteq [k'_a, \ell'_a].$$

A ‘smaller’ characteristic in this partial order is more specific and thus more informative. We will make this more formal in Sect. 3.1, and later use this to efficiently mine informative patterns.

Figure 3 shows a toy graph where vertices correspond to geographical areas described by number of different venues. An edge links vertices that correspond to adjacent areas (that share a part of their borders). A pattern (U, S) that can be interesting is: $(U = \{v_3, v_7\}, S = \{(food, [30, 32]), (college, [0, 1])\})$. Indeed, vertices in U contain a higher (resp. lower) number of food (resp. college) venues comparing with the rest of the graph, and their numbers fall within the intervals specified in S .

At the same time, a CSEA pattern (U, S) is more interesting if we can describe the cover U more concisely in some *intuitive description*. Thus, along with the pattern

syntax, we must also specify how a pattern from this language will be described. To this end, we propose to describe the cover U as a neighborhood of a specified radius from a given specified vertex, or more generally as the intersection of a set of such neighborhoods. For enhanced expressive power, we additionally allow for the description to specify exceptions on the above: vertices that do fall within this (intersection of) neighborhood(s), but which are to be excluded from the cover U , because they do not exhibit the same characteristic (they do not fulfill Eq. 1). Exceptions increase the complexity to interpret such patterns (as will be quantified in the description length), but greatly increase the expressive power of the CSEA pattern syntax.

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 Sect. 3.3, is a good measure of the complexity to assimilate or understand it. The qualitative experiments reported in Sect. 6 appear to confirm that this is the case.

3 Subjective interestingness of CSEA patterns

The previous sections already hinted at the fact that we will formalize the interestingness of a CSEA pattern (U, S) by trading off the amount of information contained in the pattern against the complexity of interpreting the pattern. We will use information theory to quantify both the informativeness and the complexity using the strategy outlined in De Bie (2011a).¹

Precise definitions will be given below, but first we introduce the statistic ultimately used to rank patterns. The Information Content (IC) of a CSEA pattern, which quantifies the amount of information contained in a pattern, depends on both the cover U and the characteristic S . Intuitively, it should be larger when more vertices are involved, when the intervals are narrower, and when they are more extreme. We denote the information content as $IC(U, S)$.

The Description Length (DL), which quantifies the interpretation complexity, also depends on U and S and will be denoted as $DL(U, S)$. Likewise, communicating larger characteristics is strictly more time-consuming, but we will not describe U directly, so the DL for a set U is more intricate as discussed in Sect. 3.3. We will rank patterns by the quantity that we call the Subjective Interestingness (SI) of a CSEA pattern (U, S) , which corresponds to the rate at which information is transmitted to the user, and is defined as:

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

3.1 The information content of a CSEA pattern

The information carried by a pattern is quantified by the information content (De Bie 2011a), a quantity also known as the self-information or surprisal (Cover and Thomas

¹ This approach is now known as the FORSIED framework.

1991). Particularly, this equals the reduction in uncertainty about the data when we learn about the pattern, and is defined as

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

Here $\Pr(U, S)$ is the probability that the CSEA (U, S) is present in the data. This explicates that we have to define such a distribution over the space of all patterns. We can achieve this as follows. The data $\hat{\omega}$ can be seen as a sample from the space of all possible vertex-attributed graphs Ω , where $G = (V, E, A) = \hat{\omega} \in \Omega$ and all elements in Ω have the same set of vertices and edges, since we assume these are known, while the attribute values are unknown to the user. Then let $\Pr$ denote a probability distribution over the set Ω of possible vertex-attributed graphs with vertices V and edges E (i.e., the possible value combinations of A). We refer to $\Pr$ as the *background distribution* and will introduce a convenient and tractable choice in the following section.

Generically, given a distribution over all possible datasets, we can obtain a distribution over patterns, by observing that a pattern is a set $\Omega' \subseteq \Omega$ that specifies $\hat{\omega}$ falls within Ω' and not outside of Ω' (De Bie 2011a; Lijffijt et al. 2014). I.e., it may reduce the value combinations deemed possible and hence provide information. The probability of a pattern Ω' can then be computed through integration, i.e., $\Pr(\Omega') = \int_{\omega \in \Omega'} \Pr(\omega) d\omega$.

In this context, $\Omega' = (U, S)$, which limits the possible attribute values of the vertices in the cover U . How to compute the probability of a CSEA pattern (U, S) is considered in more detail in the following sections.

The power of this approach is that we quantify the IC of a pattern against a prior belief state about the data. It rigorously models the fact that the more plausible the pattern is according to a model, the less information a pattern provides, and thus the smaller the information content ought to be. It is possible to specify the model accounting for (user specific) background knowledge and hence affect the ranking of patterns in a subjective manner.

In Sect. 3.2, we first discuss which prior beliefs could be appropriate for CSEA patterns, and how to infer the corresponding background distribution. Then, in Sect. 3.3, we discuss how the description length $\text{DL}(U, S)$ can be defined appropriately.

3.2 Information content and prior beliefs for count attributes

Positive integers as attributes For concreteness, let us consider the situation where the attributes are positive integers ($a : V \rightarrow \mathbb{N}, \forall a \in A$), as will be our main focus throughout this paper.² For example, if the vertices are geographical areas (with edges connecting vertices of neighboring areas), then the attributes could be counts of particular types of places in the area (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 area than it would be in a small area. Similarly, a large value for an attribute that is

² The results presented can be extended relatively directly for Boolean and real-valued attributes. De Bie (2011b) shows how to derive the background distribution $\Pr(A)$ corresponding to these types of attributes.

Fig. 4 Statistics corresponding to the formalized constraints for the case of the toy graph of Fig. 3. Left: The distribution of vertices sizes (the first constraint). Right: The total number of each venue type (the second constraint)

generally large is less informative than if it were generally small. The above is only true, however, if the user knows (or believes) *a priori* at least approximately what these averages are for each attribute, and what the ‘size’ is of each area. Such prior beliefs can be formalized as equality constraints on the values of the attributes A on all vertices, or mathematically:

$$\sum_A \Pr(A) \left(\sum_{a \in A} a(v) \right) = \sum_{\hat{a} \in \hat{A}} \hat{a}(v), \quad \forall v \in V, \quad (2)$$

$$\sum_A \Pr(A) \left(\sum_{v \in V} a(v) \right) = \sum_{v \in V} \hat{a}(v), \quad \forall a \in A. \quad (3)$$

The first constraint means that the user already knows the size (the total count) of each vertex, while the second constraint means that the user knows the total count of each attribute in the overall graph. Even if the user does not have these priors, they also can be easily communicated to her before the mining process through simple statistical tools as shown in Fig. 4 for the toy graph.

These constraints will not be sufficient to uniquely determine the distribution $\Pr(A)$. A common strategy to overcome this problem is to search for the distribution that has the largest entropy subject to these constraints, to which we will refer as the MaxEnt distribution. The argument for this choice is that any distribution other than the MaxEnt distribution effectively makes additional assumptions about the data that reduce the entropy. As making additional assumptions biases the distribution, the MaxEnt distribution is the most rigorous choice.

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 (in Eqs. 2 and 3) and the normalization $\sum_A \Pr(A) = 1$. As shown in De Bie (2011b), this is a convex optimization problem, the optimal solution of which 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: $p_{av} = 1 - \exp(\lambda_a + \lambda_v)$, with λ_a and λ_v the Lagrange multipliers corresponding to the two constraint types.

The optimal values of these multipliers can be found by solving the Lagrange dual optimization problem. As this Lagrange dual problem is unconstrained, convex, and continuously differentiable, it can be solved using standard convex optimization methods, including second-order methods (such as Newton's method), or first-order methods (such as conjugate gradient descent) for large-scale problems. For details we refer the reader to De Bie (2011b), where it was also shown how the scalability can be further enhanced by exploiting the fact that the number of *distinct* constraint values—i.e. the constants in the right hand sides of Eqs. 2 and 3—is often very small.

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

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

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 areas have very different total counts, the same interval could be very informative to some vertices while being uninformative to others, which makes it hard to find CSEA patterns with a characteristic that is informative for all vertices in its cover.

Let us illustrate this issue with the example of Fig. 3. The graph contains the pattern $(U = \{v_3, v_7\}, S = \{(food, [30, 32]), (college, [0, 1])\})$. This can be interpreted as a relatively high presence of food venues and low presence of college. Likewise in v_{11} , there is no college and the number of food venues is significantly high. Even if $\hat{food}(v_{11})$ is only 22, this still makes sense because the size of v_{11} is only 24, while the size of v_3 and v_7 is more than 40. We would like to inform the user that these same prevalences about food and college are present in all of $\{v_3, v_7, v_{11}\}$. However, v_{11} does not contain $S = \{(food, [30, 32]), (college, [0, 1])\}$. In order to contain all $\{v_3, v_7, v_{11}\}$, we need to use the restriction $(food, [22, 32])$ instead. This restriction is larger and much less informative than $(food, [30, 32])$, especially for v_3 and v_7 which are vertices with great sizes. We need a different way that allows to take into account the size of each vertex when establishing the characteristic.

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

Fig. 5 The toy graph of Fig. 3 with normalized attribute values (p -values indicating how over-represented the value is as compared to a MaxEnt model with category and vertex size constraints, see Sect. 3.2)

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

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.

In other words, $\hat{c}_a(v)$ is the probability that the expected value of $a(v)$ by the user is higher than the observed value $\hat{a}(v)$. Low values of $\hat{c}_a(v)$ correspond to exceptionally prevalent attributes, because it means that the user does not expect a value of $a(v)$ as large as the observed one $\hat{a}(v)$, while high values of $\hat{c}_a(v)$ correspond to exceptionally non-prevalent attributes.

For example, after applying this transformation to the toy graph of Fig. 3, this gives the p -values presented in Fig. 5. Let us take the vertex v_3 and the attribute *food*, the transformation gives $\hat{c}_{food}(v_3) \triangleq \Pr(food(v_3) \geq 30) = 0.2$. An example of pattern (U, S) is such that $U = \{v_3, v_7, v_{11}\}$ and $S = \{(food, [0, 0.2]), (college, [0.9, 1])\}$ (an exceptional prevalence of food and an exceptional non-prevalence of college). Notice that even if the size of v_{11} (and consequently the number of its food venues) is lower than the sizes of v_3 and v_7 , the p -value normalization made it possible to capture the common characteristic S that covers all the set $U = \{v_3, v_7, v_{11}\}$.

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 $a(v)$. In other terms, what is the probability that the significance of $\hat{a}(v)$ falls within $[k_{c_a}, \ell_{c_a}]$? How surprising is it? This is given by:

Fig. 6 Values of $\Pr(c_a(v) \in [k_{c_a}, l_{c_a}])$ as a function of k_{c_a} and l_{c_a} , for different values of the geometric parameter $p_{av} \in \{0.01, 0.1, 0.5\}$

$$\begin{aligned}
 \Pr(c_a(v) \in [k_{c_a}, l_{c_a}]) &= \Pr\left(\Pr(a(v) \geq \hat{a}(v)) \in [k_{c_a}, l_{c_a}]\right), \\
 &= \Pr\left((1 - p_{av})^{a(v)} \in [k_{c_a}, l_{c_a}]\right), \\
 &= \Pr\left(a(v) \leq \frac{\log(k_{c_a})}{\log(1 - p_{av})} \wedge a(v) \geq \frac{\log(l_{c_a})}{\log(1 - p_{av})}\right), \\
 &= \Pr\left(a(v) \leq \lfloor \log_{1-p_{av}}(k_{c_a}) \rfloor \wedge a(v) \geq \lceil \log_{1-p_{av}}(l_{c_a}) \rceil\right), \\
 &= \Pr\left(a(v) \geq \lceil \log_{1-p_{av}}(l_{c_a}) \rceil - \Pr\left(a(v) \geq \lfloor \log_{1-p_{av}}(k_{c_a}) \rfloor + 1\right)\right), \\
 &= (1 - p_{av})^{\lceil \log_{1-p_{av}}(l_{c_a}) \rceil} - (1 - p_{av})^{\lfloor \log_{1-p_{av}}(k_{c_a}) \rfloor + 1}.
 \end{aligned}$$

The last line of the equation only depends on three given values: p_{av}, k_{c_a}, l_{c_a} . In order to simplify the notation, let us define a function $\rho : (0, 1]^3 \rightarrow (0, 1]$, $\rho(x, y, z) = (1 - x)^{\lceil \log_{1-x}(z) \rceil} - (1 - x)^{\lfloor \log_{1-x}(y) \rfloor + 1}$. In what follows, we will use ρ to express the latter probability as:

$$\Pr(c_a(v) \in [k_{c_a}, l_{c_a}]) = \rho(p_{av}, k_{c_a}, l_{c_a}). \quad (4)$$

Note that under mild conditions, p -values of continuous random variables are uniformly distributed under the null hypothesis (see Proposition C on page 63 of Rice 2007). Thus, if $a(v)$ was continuous, the probability distribution of $c_a(v)$ would be uniform [i.e., $\Pr(c_a(v) \in [k_{c_a}, l_{c_a}]) = l_{c_a} - k_{c_a}$]. Unfortunately $a(v)$ is discrete (the target space is $\mathbb{N}$), so this result cannot be readily applied. Even so, it begs the question whether a uniform distribution may be a good approximation for the distribution of $c_a(v)$.

In order to demonstrate this is not the case, Fig. 6 shows values of $\Pr(c_a(v) \in [k_{c_a}, l_{c_a}])$ as a function of k_{c_a}, l_{c_a} , for three different values of $p_{av} \in \{0.01, 0.1, 0.5\}$. The closer p_{av} is to 0, the closer $\Pr(c_a(v))$ is to the uniform distribution. However, when p_{av} increases, the uniform distribution is no longer a good approximation for the distribution of $c_a(v)$. This justifies the need to use the exact values given in Eq. 4 to compute $\Pr(c_a(v) \in [k_{c_a}, l_{c_a}])$.

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

$$\begin{aligned} \text{IC}(U, S) &= -\log(\Pr(U, S)), \\ &= - \sum_{(a, [k_{c_a}, \ell_{c_a}]) \in S} \sum_{v \in U} \log(\rho(p_{av}, k_{c_a}, \ell_{c_a})). \end{aligned} \quad (5)$$

In this paper, we focus on intervals $[k_{c_a}, \ell_{c_a}]$ where either $k_{c_a} = 0$ (the minimal value) and $\ell_{c_a} < 0.5$, or $\ell_{c_a} = 1$ (the maximal value) and $k_{c_a} > 0.5$. Such intervals state that the values of an attribute are all significantly large³ or significantly small respectively, for all vertices in U . We argue such intervals are easiest to interpret.

The choice to transform count attributes into p -values before searching for interesting CSEA patterns was made to balance ease of interpretation with usefulness (informally speaking). Indeed, intervals on the original attribute values (counts) are arguably easier to interpret. Yet, in many practical applications (including those considered in this paper), the attribute values for different vertices are not commensurate: their scales in different vertices may be widely different. If this scale is known to the data analyst (which it often is, at least approximately), a specified interval for an attribute may be highly informative for one vertex in the cover of a pattern, but very uninformative for the other vertices in the cover. We argue that the resulting CSEA patterns would be less useful in that they may include vertices for which the attribute values are quite trivially within the stated interval.

In contrast, p -values are commensurate, such that a given interval is equally informative for all vertices and attributes. As p -values are widely used in applied statistics, the added complexity in the interpretation is arguably modest, and justified by the enhanced usefulness. That said, note that the transformation of attributes to p -values is not a requirement, and it is trivial to apply all methods in this paper to the original attributes if desired.

3.3 Description length

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

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

where $\text{DL}_A(S)$ [resp. $\text{DL}_V(U)$] is the description length of S (resp. U).

Description length of attributes $\text{DL}_A(S)$ The higher the number of attributes in S , the harder its communication to the user could be. This can be suitably represented by:

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

³ Note empty areas have tail probabilities $\hat{c}_a(v) = 1$ for any attribute and thus fall within any upper interval, but also $\text{IC} = 0$ for any attribute of that area as both $\ell_{c_a} = 1$ and $p_{av} = 1$.

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

Description length of vertices $DL_V(U)$ 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 that are in the intersection but not part of the cover 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 $\mathcal{N} = \{N_d(v) \mid v \in V \wedge d \in \mathbb{N} \wedge d \leq D\}$ (with D the maximum radius d considered), 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 the set 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 $f : 2^{\mathcal{N}(U)} \times U \rightarrow \mathbb{R}$ defined as:

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

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

Clearly, there is generally no unique way to describe the set U . The best one is thus the one that minimizes f . But also, in several applications, we need to limit the number of core vertices used to describe U . This means limiting $|X|$ to some parameter α whose default value is $\alpha = |V|$. This finally leads us to the definition of the description length of U as:

$$DL_V(U) = \min_{\substack{X \subseteq \mathcal{N}(U) \\ |X| \leq \alpha}} f(X, U).$$

In Fig. 5, the set of vertices $U = \{v_3, v_7, v_{11}\}$ can be described by $X_1 = \{N_1(v_8)\}$ with three exceptions $\text{exc}(X_1, U) = \{v_8, v_4, v_{12}\}$, and with a length $f(X_1, U) = 21.5$. Another possible description of U is $X_2 = \{N_1(v_6), N_1(v_8)\}$ with no exception [since $N_1(v_6) \cap N_1(v_8) = U$] and with a length $f(X_2, U) = 18.3$. Based on the values of the function f , the description X_2 is better than X_1 . Among all the possible descriptions of U , it turns out that X_2 is the one that minimizes $f(X, U)$, consequently, $DL_V(U) = f(X_2, U)$.

4 Iterative mining of CSEA patterns

When a pattern $P_0 = (U_0, S_0)$ is observed by a rational user, her background knowledge will change to take into account this newly learned piece of information. It results

that the pattern P_0 becomes expected by her. We need as well to update our interestingness model such that the probability that the data contains the pattern P_0 becomes equal to 1. Also, this model updating will decrease the IC of other patterns if a part of their information overlaps with the one of P_0 . Patterns that have a large overlap with P_0 will then no longer be deemed interesting since their information content will substantially decrease. This approach of modifying the interestingness model to account for previously seen patterns is a natural way to avoid presenting multiple redundant patterns to the user.

Let us define $SI(P \mid P_0)$ the subjective interestingness of a pattern $P = (U, S)$ conditioned on the presence of the already observed pattern P_0 :

$$SI(P \mid P_0) = \frac{IC(P \mid P_0)}{DL(P)} = \frac{-\log(\Pr(P \mid P_0))}{DL(P)}.$$

$\Pr(P \mid P_0)$ is the probability that $P = (S, U)$ is present in the data given that $P_0 = (U_0, S_0)$ is present:

$$\Pr(P \mid P_0) = \prod_{v \in U} \prod_{(a, [k_{ca}, \ell_{ca}]) \in S} \Pr(c_a(v) \in [k_a, \ell_a] \mid P_0).$$

The value of $\Pr(c_a(v) \in [k_a, \ell_a] \mid P_0)$ for each pair of attribute $a \in A$ and vertex $v \in V$ can be computed using the law of conditional probability:

$$\Pr(c_a(v) \in [k_a, \ell_a] \mid P_0) = \frac{\Pr(c_a(v) \in [k_a, \ell_a] \wedge P_0)}{\Pr(P_0)}.$$

We consider two cases:

1. If P_0 does not give any information about $c_a(v)$ (the vertex v is not in U_0 , or there is no restriction of a in S_0), then the observation of P_0 has no impact on the probability $\Pr(c_a(v) \in [k_{ca}, \ell_{ca}] \mid P_0)$:

$$\begin{aligned} \Pr(c_a(v) \in [k_{ca}, \ell_{ca}] \mid P_0) &= \Pr(c_a(v) \in [k_{ca}, \ell_{ca}]), \\ &= \rho(p_{av}, k_{ca}, \ell_{ca}). \end{aligned}$$

2. Otherwise, $v \in U_0$ and S_0 contains a restriction of a , let it be $(a, [k_0, \ell_0])$, then:

$$\begin{aligned} \Pr(c_a(v) \in [k_{ca}, \ell_{ca}] \mid P_0) &= \Pr(c_a(v) \in [k_{ca}, \ell_{ca}] \mid c_a(v) \in [k_0, \ell_0]), \\ &= \frac{\Pr(c_a(v) \in [k_{ca}, \ell_{ca}] \cap [k_0, \ell_0])}{\Pr(c_a(v) \in [k_0, \ell_0])}, \\ &= \frac{\Pr(c_a(v) \in [\max(k_0, k_{ca}), \min(\ell_0, \ell_{ca})])}{\Pr(c_a(v) \in [k_0, \ell_0])}, \\ &= \frac{\rho(p_{av}, \max(k_0, k_{ca}), \min(\ell_0, \ell_{ca}))}{\rho(p_{av}, k_0, \ell_0)}. \end{aligned}$$

Algorithm 1: SIAS-Miner ($U, C, Result, minDesc$)**Input:** U : the current set of enumerated vertices, C : the set of candidates vertices.**Output:** $Result$: the set of CSEAs, $minDesc$: a mapping structure that stores the minimum description of U for each pattern (U, S) .

```

1 if  $C \neq \emptyset$  then
2   // choose a candidate vertex with the fail first principle
3    $v \leftarrow \operatorname{argmin}_{v \in C} |\max_S(U \cup \{v\})|$ 
4    $U' \leftarrow clo(U \cup \{v\})$ 
5   if  $U' \subseteq U \cup C$  then
6     // We prune candidates that cannot be used anymore
7      $C' \leftarrow \{v \in C \mid \max_S(U' \cup \{v\}) \neq \emptyset \wedge \mathcal{N}(U' \cup \{v\}) \neq \emptyset\}$ 
8     SIAS-Miner( $U', C' \setminus U', Result, minDesc$ )
9     SIAS-Miner( $U, C \setminus \{v\}, Result, minDesc$ )
10 else
11    $Result \leftarrow Result \cup \{(U, \max_S(U))\}$ 
12    $bestDesc \leftarrow \emptyset$ 
13    $DL_V\text{-Optimise}(U, \emptyset, \mathcal{N}(U), bestDesc)$ 
14    $minDesc[(U, \max_S(U))] \leftarrow bestDesc$ 

```

Notice that in this second case, $[k_0, \ell_0]$ and $[k_{c_a}, \ell_{c_a}]$ necessarily overlap. Otherwise, the empirical value $\hat{c}_a(v)$ needs to belong to two completely disjoint intervals to make P and P_0 hold in G , which is absurde.

As explained in Sect. 5.4, the set of CSEA patterns can be ordered using the IC measure evaluated conditional on the knowledge of the patterns ranked before.

5 SIAS-Miner algorithm

SIAS-Miner mines interesting patterns using an enumerate-and-rank approach. First, it enumerates all CSEA patterns (U, S) that are closed simultaneously with respect to U, S (see explanation below), and the neighborhood description. Second, it ranks patterns according to their SI values. The calculation of $IC(U, S)$ and $DL_A(S)$ is simple and direct. However, computing $DL_V(U)$ is not trivial, since there are several ways to describe U and we are looking for the one minimizing $f(X, U)$. To achieve this goal, we propose an efficient algorithm $DL_V\text{-Optimise}$ that calculates the minimal description of U and stores the result in the mapping structure $minDesc$. This algorithm is presented in Sect. 5.2.

5.1 Pattern enumeration

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

$max_S(U)$ defines the most specific characteristic, also made of significant intervals, associated to U :

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

For example in Fig. 3, the set of vertices $U' = \{v_3, v_7\}$ is covered by a large number of characteristics, but there is only one characteristic that is the most specific: $max_S(U') = \{(food, [0, 0.2]), (college, [0.9, 1])\}$. We remind that in this figure, the attributes are normalized counts (p -values of original counts).

Moreover, it may happen that two sets of vertices U and U' , such that $U' \subseteq U$, are covered by the same characteristic ($max_S(U') = max_S(U)$) and are described by the same neighborhoods ($\mathcal{N}(U') = \mathcal{N}(U)$). In that case, U brings more information than U' and its vertex description length $DL_V(U)$ is equal to or lower than $DL_V(U')$, as all the descriptions $X \subseteq \mathcal{N}(U')$ also cover U with a lower or equal number of exceptions. Hence, the pattern $(U', max_S(U'))$ is not useful. This motivates the idea of only exploring patterns (U, S) that are closed with respect to both S and $\mathcal{N}(U)$. Closing a set of vertices U w.r.t. S and $\mathcal{N}(U)$ maximizes $IC(U, S)$, minimizes $DL_V(U)$, but can increase $DL_A(S)$. While increasing $DL_A(S)$ is theoretically possible, however, if it does happen it will have a limited effect only, as this part of the description length tends to be smaller than the other part, $DL_V(U)$. This is due to the fact that the number of attributes $|A|$ is generally significantly smaller than the number of vertices $|V|$ in attributed graphs. Indeed, in the experimental evaluation of SIAS-Miner (Sect. 6), $DL_A(S)$ represents in average only 33% of the overall $DL(U, S)$. For 92% of the returned patterns from all the studied datasets, $DL_A(S)$ is smaller than $DL_V(U)$. The highest percentage of $DL_A(S)$ over $DL(U, S)$ in patterns discovered in the experiments is 66%. Given the dramatic positive effect on the scalability of SIAS-Miner, we believe this mild approximation, resulting from exploring only closed CSEA patterns, can be perfectly justified.

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

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

$clo(U)$ is indeed a closure function since it is extensive ($U \subseteq clo(U)$), idempotent ($clo(U) = clo(clo(U))$), and monotonic [if $X \subseteq Y$, then $clo(X) \subseteq clo(Y)$]. In Fig. 3, let us consider $U' = \{v_3, v_7\}$ and $U = \{v_3, v_7, v_{11}\}$, we can notice that $max_S(U') = max_S(U)$ and $\mathcal{N}(U') = \mathcal{N}(U)$. This means that all the descriptions $X \subseteq \mathcal{N}(U')$ also cover U . In this example we have $clo(U') = U$.

We aim to only enumerate closed patterns $(clo(U), max_S(U))$. To this end, SIAS-Miner uses the divide and conquer algorithm designed to efficiently compute closed structures described in (Boley et al. 2010). Initially, SIAS-Miner is called with $U = \emptyset$

Fig. 7 First enumeration steps of SIAS-Miner applied to the toy example given in Fig. 5 with $D = 1$

and $C = V$. In each recursive call, SIAS-Miner chooses a candidate $v \in C$ considering two optimizations to obtain a more balanced enumeration tree. If $U = \emptyset$, a vertex v is selected according to its degeneracy order (Eppstein and Strash 2011) to prioritize first the vertices that should lead to small graphs. If $U \neq \emptyset$, a vertex v is selected following the fail first principle, i.e. the vertex that leads to the smallest characteristic so that to backtrack as soon as possible. Then, the closure of $clo(U \cup \{v\})$ is computed and stored in U' . If U' is included in $U \cup C$ (see Line 5), we have the guarantee that U' has not been enumerated yet and the enumeration process continues. The candidates that cannot be added anymore to U' are pruned and SIAS-Miner is recursively called on U' (Line 8). Another recursive call is also made to enumerate subgraphs that do not contain $\{v\}$ (Line 9). When C is empty, the closed pattern $(U, \max_S(U))$ is stored in *Result*, and DL_V -Optimise is called to compute $DL_V(U)$ that is finally stored in *minDesc*.

In Fig. 7, we show the first enumeration steps of SIAS-Miner when applied to the toy graph of Fig. 5 with $D = 1$. SIAS-Miner starts from the root node ($U = \emptyset$ and $C = V$), and then visit the nodes in order of their number. Let us explain how SIAS-Miner generates Step 2 from Step 1. It chooses v_1 as the next candidate. This initially gives $U = \{v_1\}$, $\mathcal{N}(U) = \{N_1(v_1), N_1(v_2), N_1(v_5)\}$, and $\max_S(U) = \{(college, [0, 0.3]), (professional, [0, 0.3])\}$. After that, U is extended using the closure operator clo to add candidates that conserve the current $\max_S(U)$ and $\mathcal{N}(U)$. It results that $U = \{v_1, v_2, v_5\}$. Also, the set of remaining candidates becomes $C = \{v_6, v_9\}$, because the other candidates do not share a neighborhood and a characteristic with U . Then, SIAS-Miner continues to Step 3. When $C = \emptyset$ (like in Step 3), the pattern $(U, \max_S(U))$ is added to the result set, and SIAS-Miner comesbacks to the previous recursive call to continue the enumeration process.

5.2 Computing $DL_V(U)$

Computing $DL_V(U)$ is NP-Hard. In fact, if we replace $(\log(|\cap_{x \in X} x|))$ in $f(X, U)$ with a constant value, this problem becomes equivalent to the weighted set cover: it consists in finding the optimal cover of the set $\overline{U}$ based on unions of complements

Algorithm 2: $\text{DL}_V\text{-Optimise}(U, X, \text{Cand}, \text{bestDesc})$

Input: U the set to describe, $X, \text{Cand} \subseteq \mathcal{N}(U)$ the current description and the candidates, bestDesc the current best description found.

Output: bestDesc the best description of the current search sub-space.

```

1 if  $LB(X, U, \text{Cand}) < f(\text{bestDesc}, U)$  then
2   if  $|X| < \alpha$  and  $\text{Cand} \neq \emptyset$  then
3      $\text{pruneUseless}(U, X, \text{Cand})$ 
4      $e \leftarrow \text{argmin}_{e' \in \text{Cand}} f(X \cup \{e'\}, U)$ 
5      $\text{DL}_V\text{-Optimise}(U, X \cup \{e\}, \text{Cand} \setminus \{e\}, \text{bestDesc})$ 
6      $\text{DL}_V\text{-Optimise}(U, X, \text{Cand} \setminus \{e\}, \text{bestDesc})$ 
7   else if  $f(X, U) < f(\text{bestDesc}, U)$  then
8      $\text{bestDesc} \leftarrow X$ 

```

Algorithm 3: $\text{pruneUseless}(U, X, \text{Cand})$

```

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

```

$\overline{N_i(v)}$ and exceptions $\{v\}$ such that $v \in \overline{U}$. Nevertheless, we propose a branch-and-bound approach that takes benefit from several optimization techniques to solve this problem on instances of interest.

In order to find the optimal description of a set of vertices U , we explore the search space $2^{\mathcal{N}(U)}$ with a branch-and-bound approach described in Algorithm 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, $\text{DL}_V\text{-Optimise}$ is called with $X = \emptyset$ and $\text{Cand} = \mathcal{N}(U)$. In each call, a neighborhood $e \in \text{Cand}$ 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 $\text{DL}_V\text{-Optimise}$. 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 $\text{DL}_V(U)$. The principle of LB is to evaluate the maximum reduction in exceptions that can be obtained when description X is extended with neighborhoods of Y : $\text{gain}_Y(X, U) = |\text{exc}(X, U)| - |\text{exc}(X \cup Y, U)|$, with $Y \subseteq \text{Cand}$. This function can be rewritten using neighborhood complements as $\text{gain}_Y(X, U) = |\cup_{y \in Y} (\overline{y} \cap \text{exc}(X, U))|$.⁴ We obtain then a simple upper bound of the gain function using the ordered set $\{g_1, \dots, g_{|\text{Cand}|}\}$ of $\{\text{gain}_{\{e\}}(X, U) \mid e \in \text{Cand}\}$ such that $g_i \geq g_j$ if $i \leq j$:

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

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

Fig. 8 A toy example,
 $U = \{v_1, v_3, v_5, v_7\}$ and
 $\mathcal{N}(U) = \{N_1(v_2), N_1(v_3),$
 $N_1(v_7)\}$, for $D = 1$

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

For instance, in Fig. 8, we need to describe $U = \{v_1, v_3, v_5, v_7\}$ by an intersection of elements from $\mathcal{N}(U) = \{N_1(v_2), N_1(v_3), N_1(v_7)\}$, given that the maximum radius for descriptors is $D = 1$. Let us consider that during the exploration we have the description $X = \{N_1(v_3)\}$, and the remaining candidates $Cand = \{N_1(v_2), N_1(v_7)\}$. At that time, the exceptions are: $exc(X, U) \triangleq \{v_2, v_4, v_6\}$. In order to compute the upper bound of $gain_Y(X, U)$ for $Y \subseteq Cand$, we need to compute the ordered set of $gain_{\{e\}}(X, U)$ for $e \in Cand$. We have $gain_{\{N_1(v_2)\}}(X, U) = 2$ because adding $N_1(v_2)$ allows to remove the exceptions $\{v_4, v_6\}$, and $gain_{\{N_1(v_7)\}}(X, U) = 1$ because adding $N_1(v_7)$ removes only one exception v_4 . Thus, the ordered list of gains is $[2, 1]$. This means that, for $Y \subseteq Cand$, if $|Y| = 1$ then $gain_Y(X, U) \leq 2$, and if $|Y| = 2$, then $gain_Y(X, U) \leq 3$.

This is the foundation of the function LB defined as

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

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

Proof Considering that $|\text{exc}(X \cup Y, U)| = |\text{exc}(X, U)| - gain_Y(X, U)$, and using Property 1: we have $|\text{exc}(X \cup Y, U)| \geq |\text{exc}(X, U)| - \sum_{j=1}^{|Y|} g_j$. Also, since the number of exceptions is reduced when X is extended, the minimum number of exceptions that we can reach in all the search space is $|\text{exc}(\mathcal{N}(U), U)|$. Thus, we have the following lower bound for $|\text{exc}(X \cup Y, U)|$:

$$|\text{exc}(X \cup Y, U)| \geq \max \left(|\text{exc}(\mathcal{N}(U), U)|, |\text{exc}(X, U)| - \sum_{j=1}^{|Y|} g_j \right) \quad (6)$$

$f(X \cup Y, U)$ is monotonically increasing w.r.t. $\text{exc}(X \cup Y, U)$. If we replace $\text{exc}(X \cup Y, U)$ by its lower bound from Eq. 6 in $f(X \cup Y, U)$, this will give a lower bound for f :

$$\begin{aligned} f(X \cup Y, U) &\geq (|X| + |Y| + 1) \times \log(|\mathcal{N}|) \\ &\quad + \left(1 + \max \left\{ |\text{exc}(\mathcal{N}(U), U)|, |\text{exc}(X, U)| - \sum_{j=1}^{|Y|} g_j \right\} \right) \\ &\quad \cdot \log \left(|U| + \max \left\{ |\text{exc}(\mathcal{N}(U), U)|, |\text{exc}(X, U)| - \sum_{j=1}^{|Y|} g_j \right\} \right), \end{aligned}$$

But also, by definition: $LB(X, U, \text{Cand}) \leq (|X| + |Y| + 1) \times \log(|\mathcal{N}|) + (1 + \max\{0, |\text{exc}(X, U)| - \sum_{j=1}^{|Y|} g_i\}) \cdot \log(|U| + \max\{|\text{exc}(\mathcal{N}(U), U)|, |\text{exc}(X, U)| - \sum_{j=1}^{|Y|} g_i\})$. So, by transitivity $LB(X, U, \text{Cand}) \leq f(X \cup Y, U)$, and this concludes the proof. $\square$

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

Function pruneUseless (line 3) removes candidate elements that can not improve the description length, that is candidates $e \in \text{Cand}$ for which $\text{gain}(\{e\}, X, U) = 0$. 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 $X \cup Y$.

In Fig. 8, for $X = \{N_1(v_7)\}$ and $\text{Cand} = \{N_1(v_2), N_1(v_3)\}$, the exceptions are: $\text{exc}(X, U) = \{v_2, v_6\}$. All these exceptions also belong to $N_1(v_3)$ which is a candidate. For this part of the search space, $N_1(v_3)$ does not reduce the number of exceptions, so it does not improve (reduce) the value of f . `pruneUseless` allows to prune this kind of candidates.

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

5.3 Time complexity of SIAS-Miner

Theoretically, the number of closed CSEA patterns can be exponential in the size of the input dataset. Since SIAS-Miner computes all the closed patterns, the number of enumeration steps can be exponential too. Still, we can study the delay complexity (worst case complexity between two enumeration steps). By denoting by $\mathcal{C}_{DL_V}$ the complexity of DL_V -Optimise, the delay of SIAS-Miner is in $\mathcal{O}(\max\{|V|^2 \cdot (|V| + |A|), \mathcal{C}_{DL_V}\})$. In fact, the complexity of making one recursive call of SIAS-Miner when $C \neq \emptyset$ is $\mathcal{O}(|V| \cdot (|V| + |A|))$, which corresponds to the cost of Line 7 (the

other lines have lower complexities). SIAS-Miner enumerates the closed patterns in a depth-first manner. The depth of the search space is bounded by $|V|$, as in each recursive call at least one element $v \in V$ is removed from C . Thus, the number of recursive calls between two leaves is bounded by $2 \cdot |V|$. Between two enumerated patterns, DL_V -Optimise is called once. Therefore, the time delay of SIAS-Miner is $\mathcal{O}(\max\{|V|^2 \cdot (|V| + |A|), \mathcal{C}_{DL_V}\})$.

It remains to compute $\mathcal{C}_{DL_V}$ the complexity of DL_V -Optimise. In the general case, DL_V -Optimise can make at worst $2^{|V|}$ recursive calls to compute $DL_V(U)$. However, in concrete applications, the parameter α , that controls the size of the description, is set to a small value (e.g., generally $\alpha = 2$). Indeed, in our experiments, the majority of top patterns are described with at most two neighborhood intersections. For $\alpha = 2$, there is at most $\frac{|V|(|V|+1)}{2}$ possible descriptions, and the cost of each recursive call of DL_V -Optimise is $\mathcal{O}(|V|^2)$. Then, it gives a total complexity of $\mathcal{O}(|V|^4)$ to compute $DL_V(U)$ if $\alpha = 2$. In a more general manner, the complexity of computing $DL_V(U)$ is at most $\mathcal{O}(|V|^{2+\alpha})$.

Thus, for a specific value of α , SIAS-Miner has a polynomial time delay, and the worst case complexity of this delay is at most $\mathcal{O}(\max\{|V|^2 \cdot (|V| + |A|), |V|^{2+\alpha}\})$. We show in Sect. 6 that SIAS-Miner is able to run on real world datasets with tens of thousands vertices, even if α is set to its maximum value $|V|$.

5.4 Using SIAS-Miner to iteratively mine CSEA patterns

The output of SIAS-Miner consists of the list of all the patterns sorted by their initial SI measure. In order to iteratively mine CSEA patterns with the application of the model updating proposed in Sect. 4, we can run SIAS-Miner once and apply a lazy sorter on its output. At the beginning of each iteration, this sorter picks the pattern with the highest SI, and proceeds to the update of SI of the other patterns in the previously sorted list $L = \{P_0, \dots, P_q\}$. However, we do not need to update the SI of all the patterns in L : we stop as soon as we find a pattern P_i whose updated SI is the highest among the already updated patterns $\{P_0, \dots, P_{i-1}\}$, but also higher than the SI of the patterns that are not updated yet: $\{P_{i+1}, \dots, P_q\}$. In fact, the SI of all these remaining patterns will either decrease or remains unchanged, so P_i will necessarily be the best pattern of the next iteration. This procedure can be then repeated iteratively to get more patterns.

6 Experiments

In this section, we report our experimental results. We start by describing the real-world datasets we used, as well as the questions we aim to answer. Then, we provide a thorough comparison with state-of-the-art algorithms: Cenergetics (Bendimerad et al. 2018), P-N-RMiner (Lijffijt et al. 2016), and GAMER (Günemann et al. 2010). Eventually, we provide a qualitative analysis that demonstrates the ability of our approach

to achieve the desired goal. For reproducibility purposes, the source code and the data are made available.⁵

Experimental setting Experiments are performed on four datasets:

- The London graph ($|V| = 289$, $|E| = 544$, $|\hat{A}| = 10$) is based on the social network Foursquare.⁶ Each vertex depicts a district in London and edges link adjacent districts. Each attribute stands for the number of places of a given type (e.g. outdoors, colleges, restaurants, etc.) in each district. The total number of represented venues in this graph is 25029.
- The Ingredients graph ($|V| = 2400$, $|E| = 7932$, $|\hat{A}| = 20$) is built from the data provided by Kaggle⁷ and features given by Yummly.⁸ Each vertex is a recipe ingredient. The attributes correspond to the number of recipes grouped by nationality (greek, italian, mexican, thai, etc.) that include this ingredient. An edge exists between two ingredients if the Jaccard similarity between their recipes is higher than 0.03. The total number of used recipes is 39774.
- The US Flights graph ($|V| = 322$, $|E| = 2039$, $|\hat{A}| = 14$) is a dataset provided by Kaggle⁹ which contains information about flights between USA airports in 2015. The vertices represent USA airports and the attributes depict the number of flights per airline company in the corresponding airports. Edges connect two airports if there are at least 100 flights between them.
- The DBLP graph ($|V| = 38,895$, $|E| = 112,404$, $|\hat{A}| = 10$) is a co-authorship graph built from the DBLP digital library. Each vertex represents an author who has published at least one paper in 10 major conferences and journals of in the Data Mining and Databases communities,¹⁰ between January 1990 and March 2019. Each edge links two authors who co-authored at least one paper in these conferences and journals. The attributes depict for each author the number of publications in the 10 aforementioned conferences and journals.

Considering all the numerical values of attributes is computationally expensive, we pre-process each graph so that for each attribute, the values $c_a(v)$ are binned into five quantiles. This means that each attribute $c_a(v)$ is numerical and takes only 5 different values.

There is no approach that supports the discovery of subjectively interesting attributed subgraphs in the literature. Nevertheless, we identify three approaches whose goal shares some similarities with ours. These three approaches are representative in comparison with the state-of-the-art. Other methods from the literature are not considered in this experimental study because they do not handle (several) numerical attributes or they require some additional settings (e.g., community search). We thus consider in our study the following methods:

- P-N-RMiner (Lijffijt et al. 2016) is an algorithm that mines multi-relational datasets to enumerate a specific structure of patterns called Maximal Complete Connected

⁵ <http://goo.gl/ZxsvbX>.

⁶ <https://foursquare.com>.

⁷ <https://www.kaggle.com/kaggle/recipe-ingredients-dataset>.

⁸ <https://www.yummly.com/>.

⁹ <https://www.kaggle.com/usdot/flight-delays/data>

¹⁰ DMKD, VLDB journal, Machine Learning, TKDE, KDD, ICDM, VLDB, ICDE, ICML, IJCAI.

Subsets (MCCS). Any vertex-attributed graph can be mapped to an entity-relational model where the pattern syntax of P-N-RMiner is equivalent to ours [each MCCS corresponds to a closed pattern ($clo(U), max_S(U)$) in our context]. This means that P-N-RMiner is very suitable to evaluate the performance of our algorithm. The mapping from a graph to the required relational format is detailed in Appendix A. Although the pattern syntax in this design is equivalent to our approach, our interestingness quantification is very different, because the information contained in the patterns shown to the user does not align with the ranking of P-N-RMiner. Hence, we use P-N-RMiner only to compare the runtime performance of our approach.

- Cenergetics (Bendimerad et al. 2018) aims at discovering connected subgraphs involving overrepresented and/or underrepresented attributes. It assesses exceptionality with the weighted relative accuracy (WRAcc) measure that accounts for margins but cannot account for other background knowledge.
- GAMER (Günemann et al. 2010): Given an attributed graph, this method finds dense subgraphs (quasi-cliques) where vertices show a high similarity in subsets of attributes. In these subgraphs, these attribute values fall into narrow intervals whose width does not exceed a specified threshold W . The main difference with our approach is that GAMER looks only for similarity and cohesiveness, but not exceptionality and surprisingness.

In this experimental study, our aim is to answer the following questions: What is the efficiency of SIAS-Miner regarding to graph dimensions? Is SIAS-Miner able to deal with real world datasets? What are the differences between the results of our approach and those of the considered baselines? What about the relevance of the CSEA patterns?

Quantitative experiments Figure 9 reports the runtime of SIAS-Miner and P-N-RMiner according to the number of vertices, the number of attributes and the minimum number of vertices of searched patterns, for each of the datasets.

The points that are not displayed in the curves of P-N-RMiner are the ones that exceeded a time limit of 10^4 s. For example, when we varied the attributes in the ingredients dataset, P-N-RMiner was not able to finish any configuration in less than 10^4 s. These tests reveal that SIAS-Miner outperforms P-N-RMiner in all the datasets in almost all the configurations, and the difference is generally between 2 and 4 orders of magnitude. Although P-N-RMiner is a principled algorithm that uses several advanced optimization techniques, SIAS-Miner is faster since it is particularly defined to deal with attributed graphs, and it takes benefits from several specificities of these structures to optimize the search space exploration.

Qualitative experiments The goal is to compare the properties of the patterns found by SIAS-Miner with those of Cenergetics and GAMER. We do not consider P-N-RMiner for two reasons: (1) Even if the pattern syntax can be made equivalent between SIAS-Miner and P-N-RMiner, the model used in P-N-RMiner to assess the quality of patterns is not adapted to the goal of mining attributed graphs, (2) P-N-RMiner was not able to finish its execution in any of the studied datasets when the whole set of vertices are considered. We first extract the top 200 diversified patterns of each approach. Also, DBLP results for GAMER are not studied because this approach was not able to perform on this dataset. We compute a summary of patterns obtained by

Fig. 9 SIAS-Miner versus P-N-RMiner: runtime when varying $|V|$ (first column), $|A|$ (second column) and a threshold on the minimum number of vertices in searched patterns (third column) for London graph ($D = 3$, first row), flights graph ($D = 1$, second row), ingredients graph ($D = 1$, third row), and DBLP dataset ($D = 1$, fourth row)

Cenergetics based on Jaccard similarity in order to have diversified patterns to study. This set only contains patterns whose pairwise Jaccard similarity is lower than 0.6. This step is not applied on GAMER and SIAS-Miner results since GAMER output is internally summarized with a similar approach, and SIAS-Miner patterns are already diversified thanks to the iterative mining with model updating. In the following, we compare the properties depicted in Fig. 10:

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

Fig. 10 Comparisons of properties of identified patterns by SIAS-Miner, Cenergetics, and GAMER in London, Ingredients, and Flights graphs

have a low density and relative degree, because this approach requires only the connectivity of vertices in the patterns, some of these patterns can be very sparse subgraphs.

- **Diameter:** The diameter of a subgraph U is the maximum pairwise distance between vertices of U . GAMER patterns have the smallest average diameter, while Cenergetics patterns have the highest ones. The diameter of SIAS-Miner is comparable to the one of Cenergetics in London ($D = 3$) and DBLP ($D = 1$), but it is smaller for Flights and Ingredients ($D = 1$).
- **Size and number of covered attributes:** The size of a pattern (U, S) corresponds to $|U|$ and the number of covered attributes is $|S|$. GAMER has patterns with the smallest average size, this is reasonable because it requires a harder constraint on the structure of patterns, which is the quasi-cliqueness. This small size of U allows GAMER patterns to be covered with a larger number of attributes comparing with the other approaches.

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

To conclude, there is a clear structural difference between patterns of the three approaches. GAMER finds denser subgraphs, and Cenergetics patterns are generally the sparsest ones. GAMER does not look to the exceptionality of attributes in the patterns, but only for their similarities. Another major difference is the possibility of integrating different prior beliefs in the MaxEnt model of SIAS-Miner, and the update of the background model after the observation of each pattern by the user, which is not possible in the other attributed graph mining approaches.

Illustrative results In Fig. 1, we show some patterns that SIAS-Miner discovered in the London graph. We chose the top 4 patterns, and P_{14} which is the best pattern with more than one core vertex. Green cells represent vertices covered by a CSEA pattern while blue cells are the core vertices that are also in the cover, and the red cells are normal exceptions. For example, P_1 covers the distance three neighbors of the blue vertex, with an exceptional prevalence of nightlife and professional venues. Indeed, P_1 includes the City of London which contains the primary central business district (CBD) of London, and it is known by its prevalent number of nightlife venues (pubs, bars, etc.) in some areas like the Soho and the South Bank neighborhood. The second pattern P_2 covers Stoke Newington, with a high prevalence of nightlife spots (e.g., bars) and food venues, and a non-prevalence of colleges and universities. Specifically, this area is characterized by a wide range of bars and cafes, especially around the Church Street and lining the road of Dalston. The patterns P_3 and P_4 are two other areas where food venues are prevalent. Finally, Bermondsey is covered by P_{14} and described by a high presence of professional venues, and a non-prevalence of event venues, colleges and universities.

We report in Fig. 11 the top 4 patterns discovered by SIAS-Miner in Ingredients graph. P_1 corresponds to a set of ingredients that appear a lot in Mexican recipes. They are described as neighbors of enchilada sauce which is an ingredient originally from Mexico. Notice that this pattern does not contain any exception. P_2 is defined by a set of 27 ingredients that are highly present in Indian recipes (e.g., green cardamom, garam masala, curry leaves, etc.), with ghee as a core vertex. This last ingredient is a class of butter that originated in ancient India, and is commonly used in cuisine of the Indian subcontinent. P_3 presents some ingredients that are characteristic of the Italian cuisine (e.g., pasta sauce, pizza sauce, fresh basil, etc.), and describes them as neighbors of mozzarella cheese, with a number of 2 exceptions. P_4 is another group of ingredients that are highly present in Mexican food, and they are co-occurrent with white onion in

$P_1 : \{\text{Mexican}; [0, 10^{-18}]\}^+, SI(P_1) = 40.34, |U| = 25, \text{exceptions number: } 0$

$P_2 : \{\text{Indian}; [0, 10^{-18}]\}^+, SI(P_2) = 27.03, |U| = 27, \text{exceptions number: } 5$

$P_3 : \{\text{Italian}; [0, 10^{-18}]\}^+, SI(P_3) = 24.23, |U| = 18, \text{exceptions number: } 2$

$P_4 : \{\text{Mexican}; [0, 10^{-18}]\}^+, SI(P_4) = 17.46, |U| = 15, \text{exceptions number: } 3$

Fig. 11 Top 4 patterns discovered in Ingredients ($\text{minVertices} = 5, D = 1$)

Fig. 12 Patterns discovered in the Ingredients graph. P_{15} : the best pattern with more than one attribute in the characteristic, P_{71} : the best pattern with more than one core vertex in the description. ($\text{minVertices} = 5, D = 1$)

a large number of recipes. In Fig. 12, we also present some other specific patterns, P_{15} is the best pattern with more than one attribute in the characteristic (an exceptional prevalence in Chinese, Thai and Japanese food), and P_{71} is the best pattern with more than one core vertex (the cores are garlic cloves and chili powder). This last pattern has also an exceptional prevalence in several types of cuisine, but the prevalence is much more significant in Mexican food (with $C_{\text{mexican}}(v) \leq 8.6 \cdot 10^{-4}$). For more details, we display in the companion page the complete set of top 100 patterns of each of London and Ingredients datasets.

We show in Fig. 13 the top 3 patterns discovered by SIAS-Miner in DBLP graph. P_1 consists of researchers who are co-authors of Milind Tambe in some of the 10 studied conferences and journals. They have published a significant number of papers in the IJCAI conference (156 publications), and they have not collaborated in papers of the VLDB conference or the VLDB Journal. Particularly, the central author Milind Tambe has co-authored 26 IJCAI papers with this group of researchers. P_2 describes the co-authors of Yoshua Bengio, except 5 of them. They have significantly published in the ICML conference (174 papers in overall) and have not published in the VLDB

Fig. 13 Top 3 discovered in the DBLP graph. P_{15} ($minVertices = 5, D = 1$)

Fig. 14 DBLP graph: Best pattern with more than one core vertex ($\text{minVertices} = 5, D = 1$)

or DMKD journals. P_3 shows another group of researchers belonging to the artificial intelligence community as they have a large number of publications in IJCAI (314 papers). Also, these researchers have not collaborated in papers published in KDD, VLDB, or VLDB Journal. They are described exactly as the set of co-authors of Jérôme Lang, who is a highly influential researcher from the artificial intelligence community. We also show in Fig. 14 the best DBLP pattern with more than one core vertex. It corresponds to the common co-authors of Hui Xiong and Enhong Chen, for whom there is a large number of publications in ICDM (87 papers), a low number of publications in ICML (only one paper), and no publication in VLDB.

7 Related work

Several approaches have been designed to discover new insights in vertex attributed graphs. The pioneering work of Moser et al. (2009) presents a method to mine dense homogeneous subgraphs, i.e., subgraphs whose vertices share a large set of attributes. Similarly, Günnemann et al. (2010) introduce GAMER, 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. (2012) 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 Prado et al. (2013), the authors propose to mine the graph topology of a large attributed graph by finding regularities among numerical vertex descriptors. Zhang et al. (2017) aim to find (k, r) -cores in attributed graphs, a (k, r) -cores is a connected subgraph where each vertex is connected to at least k vertices and the similarity between attributes values is higher than a minimum threshold r . Several papers (Fang et al. 2017b, 2016; Shang et al. 2016; Huang et al. 2017; Huang and Lakshmanan 2017; Fang et al. 2017a) have recently addressed the problem of community search in attributed graphs, finding communities that contain a user-specified set of vertices and have similar attribute values. (Silva et al. 2015) propose an approach to compress attributes values in a graph where each vertex is described by a single numerical attribute. This consists in partitioning the graph into

smooth areas where attribute values are similar, and store only the average values of each area. Using signal processing theory (Shuman et al. 2013; Sandryhaila and Moura 2013), some methods have been proposed to extract knowledge from attributed graphs. For instance, given a graph where vertices are augmented with Boolean attributes, (Chen et al. 2017b) study the problem of detecting localized attributes. In other terms, the goal is to find an attribute whose activated vertices form a subgraph that can be easily separated from the rest of the vertices. (Chen et al. 2018) implement a graph multiresolution representation approach for graphs where vertices are described by a single numerical attribute (a signal). It iteratively partitions a graph into smooth connected subgraphs from coarse to fine levels.

The main objective of all these approaches is to find regularities instead of peculiarities within a large graph, whereas SIAS-Miner mines subgraphs with distinguishing characteristics. Comparative experiments with GAMER in Sect. 6 provide evidence about this point.

Interestingly, Atzmueller et al. (2016) 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 the WRAcc measure. In Kaytoue et al. (2017), the authors aim at discovering contextualized subgraphs that are exceptional with respect to a model of the dataset. 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, Lemmerich et al. (2016) proposes to discover subgroups with exceptional transition behavior as assessed by a first-order Markov chain model. (Miller et al. 2013, 2015) studied the problem of subgraph detection (SPG) in simple graphs with weighted edges, and formalised it as a problem of detecting signal in noise. It aims to find subgraphs having an exceptional connectivity regarding to the one expected using a background model. Gupta et al. (2014) identify subgraphs matching a query template and having an anomalous connectivity structure. An edge is considered anomalous if a weighted attribute-similarity between the linked vertices is small. Chen et al. (2017a) propose a generic framework for subspace cluster detection in attributed graphs. This framework can be instantiated with some deviation functions to identify anomalous subspace clusters. The problem of exceptional subgraph mining in attributed graphs is introduced in Bendimerad et al. (2018). Based on an adaptation of the WRAcc measure, the method Cenergetics, which is one of the baselines in our experiments, aims to discover subgraphs with homogeneous and exceptional characteristics.

More generally, Subgroup Discovery (Lavrac et al. 2004; Novak et al. 2009) 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 background knowledge into account. Hence, one can expect that identified subgroups are less informative. The problem of taking subjective interestingness into account in pattern mining was already identified in Silberschatz and Tuzhilin (1995) and has seen a renewed interest in the last decade. The interestingness measure employed here is inspired by the FORSIED framework (De Bie 2011a, 2013), which defines the SI of a pattern as the ratio between

Table 2 Comparison of related work

	Graph topology	Vertex attributes	Numerical attributes	Pattern exceptionality	Prior knowledge
SSG-Miner (van Leeuwen et al. 2016)	×				×
Cosmic (Kaytoue et al. 2017)	×		×	×	
Cenergetics (Bendimerad et al. 2018), SODA (Gupta et al. 2014), SG-Pursuit (Chen et al. 2017a)	×	×	×	×	
Comodo (Atzmueller et al. 2016)	×	×		×	
Subgroup discovery (Lemmerich et al. 2016; Lavrac et al. 2004)		×	×	×	
GAMER (Günemann et al. 2010), TopGraphMiner (Prado et al. 2013), FocusCO (Perozzi et al. 2014) (k,r)-Core (Zhang et al. 2017), Community search (Fang et al. 2017b, 2016; Shang et al. 2016; Huang et al. 2017; Huang and Lakshmanan 2017; Fang et al. 2017a), ST compression (Silva et al. 2015), Multiresolution representation (Chen et al. 2018)	×	×	×		
CoPaM (Moser et al. 2009), SCPM (Silva et al. 2012), Localized attributes detection (Chen et al. 2017b)	×	×			
SPG (Miller et al. 2013, 2015)	×			×	
SIAS-Miner (this paper)	×	×	×	×	×

the IC and the DL. The IC is the amount of information specified 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, hence it is *subjective*, i.e., particular to the modeled belief state. van Leeuwen et al. (2016) exploit this framework for discovering surprisingly dense subgraphs in graphs without attributes, the user is assumed to have priors about the graph density and/or vertex degrees. Perozzi et al. (2014) also proposes to take into account the user in the clustering and the outlier detection tasks, but from a different perspective. It infers user preferences through a set of user-provided examples, while our goal is to integrate user background knowledge to find surprising patterns. We depict in Table 2 an overview about the methods discussed in this section.

8 Conclusion

We have introduced a new syntax for local patterns in attributed graphs, called *cohesive subgraphs with exceptional attributes* (CSEA). These patterns provide a set of attributes that have exceptional values throughout a cohesive subset of vertices. The strength of CSEA lies in its independence to any 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. We also show how to update the background model when the current knowledge of the user changes by observing a new pattern. This allows to iteratively identify patterns that give new information out of the one already acquired by the previously observed patterns.

The DL assesses the complexity of reading a pattern, the user being interested in concise and intuitive descriptions. Thus, we proposed to describe a set of vertices as an intersection of neighborhoods of certain distance from certain vertices, the distance and vertices making up the description of the subgraph. This kind of description is suitable and interpretable in many use cases illustrated in this paper, such as mining urban areas and describing them with some central locations, or mining collaboration networks and describing groups as neighbors (e.g., collaborators, friends, followers, etc.) of some specific persons.

We have proposed an effective algorithm that enumerates patterns of this language. Extensive empirical results on four 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. First, the main limitation of SIAS-Miner is its running time that may grow exponentially with respect to the number of attributes. In fact, this algorithm enumerates exhaustively the closed CSEA patterns to provide the completeness guarantee. While some heuristic strategies (e.g., Beam Search Gamberger and Lavrac 2002) can be relatively easily adapted to this

problem, a challenge is to investigate heuristic approaches that provide guarantees about the goodness of the results. Another limitation is related to the DL, although the proposed pattern redescription is suitable for many use cases, other alternative redescrptions can be more appropriate for some other applications, such as dense subgraphs (to study communities in social networks), paths (biological networks, trajectory networks), spanning trees (network). A promising avenue is to provide the possibility to use diverse types of redescrptions that can be chosen according to the mining task. Eventually, other kinds of prior beliefs can be integrated. For instance, one can consider a hierarchy over the attributes that must be thoroughly taken into account for the modeling of the priors and for reasoning when updating the background knowledge. Finally, SIAS-Miner is based on the FORSIED framework for the development of subjective interestingness measures. Exploring alternative choices for the interestingness measure is another possible avenue for further research.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

Appendix A: Transformation of an attributed graph to P-N-RMiner syntax

In the experimental evaluation, we have compared the runtime of SIAS-Miner with the one of P-N-RMiner algorithm (Lijffijt et al. 2016). Here we explain in detail how to apply the transformation of an attributed graph to an entity relational model that can be given as an input to P-N-RMiner algorithm. P-N-RMiner mines multi-relational datasets to enumerate a specific structure of patterns called MCCS.

As stated in the paper, any vertex-attributed graph can be mapped to an entity-relational model where the pattern syntax of P-N-RMiner is equivalent to ours (each MCCS pattern corresponds to a closed pattern ($clo(U)$, $max_S(U)$) in our context). The mapping from a graph to the required relational format is depicted in Fig. 15. This gives an entity-relational model $\mathbb{D}$ through (1) creation of an entity type E_v containing

Fig. 15 Transformation from (1) a graph structure to (2) an entity-relational model with $D = 1$, and (3) an example of a maximal complete connected subset (MCCS) pattern from P-N-RMiner

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.

In Fig. 15 (2) it is shown that 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 radius: $v_i \in N_k(v_j) \Rightarrow v_i \in N_l(v_j) \forall l \geq k$. A similar statement holds for intervals.

We give an example of MCCC pattern in Fig. 15 (3). It is a set of entities that form a complete connected K -partite subgraph F ($K=4$ since we have 4 involved types of entities: vertices v , neighborhoods $N_i(v_j)$, a first attribute a_1 , and a second attribute a_2): each vertex in F is connected to each neighborhood $N_i(v_j)$ in F and to each attribute value of a_1 and a_2 in F .

References

- Atzmueller M, Doerfel S, Mitzlaff F (2016) Description-oriented community detection using exhaustive subgroup discovery. *Inform Sci* 329:965–984
- Bendimerad AA, Plantevit M, Robardet C (2018) Mining exceptional closed patterns in attributed graphs. *Knowl Inf Syst* 56(1):1–25
- Bistarelli S, Bonchi F (2005) Interestingness is not a dichotomy: introducing softness in constrained pattern mining. In: Knowledge discovery in databases: PKDD 2005, 9th European conference on principles and practice of knowledge discovery in databases, Porto, Portugal, October 3–7, 2005, Proceedings, pp 22–33
- Boley M, Horváth T, Poigné A, Wrobel S (2010) Listing closed sets of strongly accessible set systems with applications to data mining. *Theor Comput Sci* 411(3):691–700
- Chen F, Zhou B, Alim A, Zhao L (2017a) A generic framework for interesting subspace cluster detection in multi-attributed networks. In: 2017 IEEE international conference on data mining, ICDM 2017, New Orleans, LA, USA, November 18–21, 2017, pp 41–50
- Chen S, Yang Y, Zong S, Singh A, Kovacevic J (2017b) Detecting localized categorical attributes on graphs. *IEEE Trans Signal Process* 65(10):2725–2740
- Chen S, Singh A, Kovacevic J (2018) Multiresolution representations for piecewise-smooth signals on graphs. *CoRR*. [arXiv:1803.02944](https://arxiv.org/abs/1803.02944)
- Cover TM, Thomas JA (1991) Entropy, relative entropy and mutual information. *Elements Inform Theory* 2:1–55
- De Bie T (2011a) An information theoretic framework for data mining. In: Proceedings of the 17th ACM SIGKDD international conference on knowledge discovery and data mining, pp 564–572
- De Bie T (2011b) Maximum entropy models and subjective interestingness. *Data Min Knowl Disc* 23(3):407–446
- De Bie T (2013) Subjective interestingness in exploratory data mining. In: International symposium on intelligent data analysis (IDA), pp 19–31
- Eppstein D, Strash D (2011) Listing all maximal cliques in large sparse real-world graphs. In: Experimental algorithms—10th international symposium, SEA 2011, Kolimpari, Chania, Crete, Greece, May 5–7, 2011. Proceedings, pp 364–375
- Fang Y, Cheng R, Luo S, Hu J (2016) Effective community search for large attributed graphs. *Proc VLDB Endowment (PVLDB)* 9(12):1233–1244
- Fang Y, Cheng R, Chen Y, Luo S, Hu J (2017a) Effective and efficient attributed community search. *VLDB J* 26(6):803–828
- Fang Y, Cheng R, Li X, Luo S, Hu J (2017b) Effective community search over large spatial graphs. *Proc VLDB Endowment (PVLDB)* 10(6):709–720
- Fortunato S (2010) Community detection in graphs. *Phys Rep* 486:75–174
- Gamberger D, Lavrac N (2002) Expert-guided subgroup discovery: methodology and application. *J Artif Intell Res* 17:501–527

- Gionis A, Mathioudakis M, Ukkonen A (2015) Bump hunting in the dark: local discrepancy maximization on graphs. In: 31st IEEE international conference on data engineering, ICDE 2015, Seoul, South Korea, April 13–17, 2015, pp 1155–1166
- Günnemann S, Färber I, Boden B, Seidl T (2010) Subspace clustering meets dense subgraph mining. In: 2010 IEEE international conference on data mining (ICDM), pp 845–850
- Gupta M, Mallya A, Roy S, Cho JHD, Han J (2014) Local learning for mining outlier subgraphs from network datasets. In: Proceedings of the 2014 SIAM international conference on data mining, Philadelphia, Pennsylvania, USA, April 24–26, 2014, pp 73–81
- Huang X, Lakshmanan L (2017) Attribute-driven community search. *Proc VLDB Endowment (PVLDB)* 10(9):949–960
- Huang X, Lakshmanan L, Xu J (2017) Community search over big graphs: models, algorithms, and opportunities. In: 2017 IEEE 33rd international conference on data engineering (ICDE), pp 1451–1454
- Kaytoue M, Plantevit M, Zimmermann A, Bendimerad AA, Robardet C (2017) Exceptional contextual subgraph mining. *Mach Learn* 106(8):1171–1211
- Lavrac N, Kavsek B, Flach PA, Todorovski L (2004) Subgroup discovery with CN2-SD. *J Mach Learn Res* 5:153–188
- Lemmerich F, Becker M, Singer P, Helic D, Hotho A, Strohmaier M (2016) Mining subgroups with exceptional transition behavior. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, pp 965–974
- Lijffijt J, Papapetrou P, Puolamäki K (2014) A statistical significance testing approach to mining the most informative set of patterns. *Data Min Knowl Disc* 28(1):238–263
- Lijffijt J, Spyropoulou E, Kang B, De Bie T (2016) P-n-rminer: a generic framework for mining interesting structured relational patterns. *Int J Data Sci Anal* 1(1):61–76
- Miller BA, Bliss NT, Wolfe PJ, Beard MS (2013) Detection theory for graphs. *Lincoln Lab J* 20(1):10–30
- Miller BA, Beard MS, Wolfe PJ, Bliss NT (2015) A spectral framework for anomalous subgraph detection. *IEEE Trans Signal Process* 63(16):4191–4206
- Moser F, Colak R, Rafiey A, Ester M (2009) Mining cohesive patterns from graphs with feature vectors. In: Proceedings of the 2009 SIAM international conference on data mining (SDM), pp 593–604
- Novak PK, Lavrac N, Webb GI (2009) Supervised descriptive rule discovery: a unifying survey of contrast set, emerging pattern and subgroup mining. *J Mach Learn Res* 10:377–403
- Perozzi B, Akoglu L, Sánchez PI, Müller E (2014) Focused clustering and outlier detection in large attributed graphs. In: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, pp 1346–1355
- Prado A, Plantevit M, Robardet C, Boulicaut J (2013) Mining graph topological patterns: finding covariations among vertex descriptors. *IEEE Trans Knowl Data Eng* 25(9):2090–2104
- Rice JA (2007) Mathematical statistics and data analysis, 3rd edn. Duxbury
- Sandryhaila A, Moura JMF (2013) Discrete signal processing on graphs. *IEEE Trans Signal Process* 61(7):1644–1656
- Shang J, Wang C, Wang C, Guo G, Qian J (2016) AGAR: an attribute-based graph refining method for community search. In: Proceedings of the sixth international conference on emerging databases: technologies, applications, and theory (EDBT), pp 65–66
- Shuman DI, Narang SK, Frossard P, Ortega A, Vandergheynst P (2013) The emerging field of signal processing on graphs: extending high-dimensional data analysis to networks and other irregular domains. *IEEE Signal Process Mag* 30(3):83–98
- Silberschatz A, Tuzhilin A (1995) On subjective measures of interestingness in knowledge discovery. In: Proceedings of the first international conference on knowledge discovery and data mining (KDD-95), pp 275–281
- Silva A, Meira W, Zaki M (2012) Mining attribute-structure correlated patterns in large attributed graphs. *Proc VLDB Endowment (PVLDB)* 5(5):466–477
- Silva A, Bogdanov P, Singh AK (2015) Hierarchical in-network attribute compression via importance sampling. In: 31st IEEE international conference on data engineering, ICDE 2015, pp 951–962
- van Leeuwen M, De Bie T, Spyropoulou E, Mesnage C (2016) Subjective interestingness of subgraph patterns. *Mach Learn* 1–35
- Zhang F, Zhang Y, Qin L, Zhang W, Lin X (2017) When engagement meets similarity: efficient (k, r)-core computation on social networks. *Proc VLDB Endowment (PVLDB)* 10(10):998–1009

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Affiliations

Anes Bendimerad¹ · Ahmad Mel³ · Jeffrey Lijffijt³ · Marc Plantevit² · Céline Robardet¹ · Tijl De Bie³

✉ Anes Bendimerad
aabendim@liris.cnrs.fr

¹ INSA Lyon, CNRS, LIRIS UMR5205, 69621 Villeurbanne, France

² Université de Lyon 1, CNRS, LIRIS UMR5205, 69622 Villeurbanne, France

³ Department of Electronics and Information Systems, IDLab, Ghent University, Ghent, Belgium