Discovering Robustly Connected Subgraphs with Simple Descriptions

Janis Kalofolias•, Mario Boley°, Jilles Vreeken•
•CISPA Helmholtz Center for Information Security, Saarbrücken, Germany
°Monash University, Melbourne, Australia
{janis.kalofolias,jv}@cispa.saarland, mario.boley@monash.edu

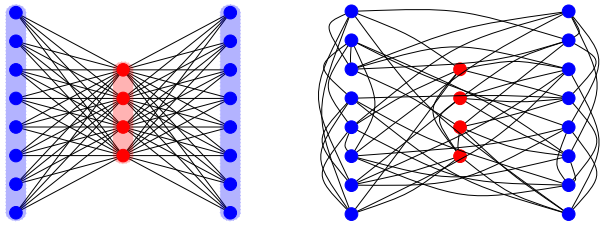
***Abstract*—We study the problem of discovering *robustly* connected subgraphs that have *simple* descriptions. Our aim is, hence, to discover vertex sets which not only a) induce a subgraph that is difficult to fragment into disconnected components, but also b) can be selected from the entire graph using just a simple conjunctive query on their vertex attributes. Since many subgraphs do not have such a simple logical description, first mining robust subgraphs and post-hoc discovering their description leads to sub-optimal results. Instead, we propose to optimise over describable subgraphs only. To do so efficiently we propose a non-redundant iterative deepening approach, which we equip with a linear-time tight optimistic estimator that allows pruning large parts of the search space. Extensive empirical evaluation shows that our method can handle large real-world graphs, and discovers easily interpretable and meaningful subgraphs.**

I. Introduction

Graphs provide a natural way to represent relationships between entities. We find graphs, ranging from power grids, social networks, up to relational databases, all around us. With the ubiquity of the graph data model, mining graphs attracted a lot of attention from the data mining community. A large part of this attention has been focused on discovering dense subgraphs—typically defined to have high edge-to-vertex ratio. The main premise of this task was that these represent vertices that 'belong together' and are therefore worth knowing.

In our work we break with this premise: we argue that, from a knowledge discovery viewpoint, subgraphs whose vertices are arbitrarily chosen to maximise o score are not only difficult to interpret, but possibly not even interesting to begin with. After all, by selecting vertices at will, there is no guarantee that there exists a reasonable explanation *why* these nodes belong together. Instead, we consider only subgraphs whose vertices can be selected out of the entire graph with a conjunctive query on their attributes. By admitting such a simple description, these subgraphs become easily interpretable: e.g., from IMDB data we find mainstream movie crew with lengthy experience to have collaborated together more than usual in the industry.

Moreover, we depart from the notion that subgraphs with high edge to vertex ratios are interesting per se. Despite its appeal at first glance, it is a rather naive a measure of whether vertices 'belong together', as it only considers numbers of edges rather than their structure. As an example, consider Fig. 1 where we depict two toy graphs of 20 vertices each. The graph on the left has a high edge to vertex ratio, but is arguably not very robustly connected; that is, we can fully disconnect it by only removing the 4 central nodes. In contrast, the graph on the right has a lower edge to vertex ratio, but is robustly connected: to disconnect it, we would have to remove 19 vertices. That is, while the leftmost graph is not uninteresting per se, the rightmost graph depicts an interesting phenomenon that when focusing on edge statistics alone we would miss.



(a) *complete bipartite graph* *edge/vertex ratio:* 3.2*, coreness:* 4 (b) 6*-regular graph (and* 6*-core)* *edge/vertex ratio:* 3*, coreness:* 6

Figure 1 [Edge/vertex–ratio vs. robust connectedness]: Although graph (a) is denser than (b), the latter is more *robustly* connected. For example, to fully disconnect (a) we need only remove its 4 central nodes, while (b) requires removing 19.

We hence study the problem of discovering *robustly* connected subgraphs that admit *simple* descriptions. We propose a score for robustness of subgraphs based on the notion of k-coreness. We then aim to discover those subgraphs that are not only simply describable, but are (much) more robustly densely connected than the remainder of the graph. Unlike the description-agnostic setup, this incurs a hard combinatorial optimization problem for which the post-hoc approach of first mining robust subgraphs and then searching for descriptions fails. We therefore use *subgroup discovery* to efficiently mine large attributed graphs with guarantees: we propose a tight optimistic estimator for a branch-and-bound variant that avoids redundancy by searching only within closed patterns. Extensive experiments on large and diverse real-world graphs show that our method, ROSI, performs very well in practice, discovering meaningful subgraphs while competing ones run out of time and memory. Further, these experiments also show that the above example is not esoteric: the densest subgraph that the recent method LDENSE [10] discovers from `DBLP` is one with high average density but a robustness of 0 (!).

For conciseness, we postpone all proofs to the appendix.[1]

[1] http://eda.mmci.uni-saarland.de/rosi/

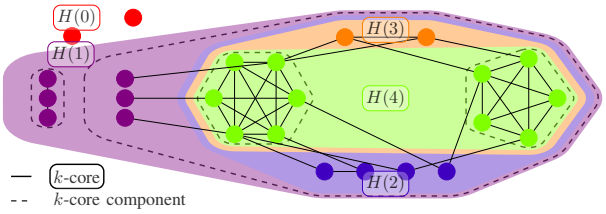


Figure 2: The core decomposition of a graph hierarchically groups its vertices into increasingly denser subgraphs.

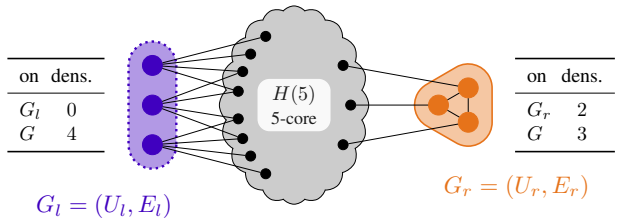


Figure 3: The average subgraph coreness $\bar{\kappa}_U$ may be misleadingly overestimated when computed with respect to the whole graph, as the average of its coreness $\kappa(v)$ over $v \in U$.

II. Measuring Robust Connectedness

We study sets of entities, for which we are given attribute values as well as structural information in the form of connections between them. Formally, we consider vertex-attributed (multi-)graphs $G = (V, E, X)$, where the vertices V correspond to entities and the edges E to connections between them. The set of vertex attributes $X = \{x_1, \ldots, x_p\}$ comprises assignments $x_i : V \to \mathcal{X}_i$ from vertices to a continuous or categorical domain $\mathcal{X}_i$. These attributes can be used to simply describe subsets based on logical expressions of vertices $v \in V$ like $\sigma(v) \equiv [\texttt{age}(v) \geq 18] \wedge [\texttt{sex}(v) = \text{`}\textit{female}\text{'}]$.

Our goal is to identify such logically described sets of vertices $U \subseteq V$ that are relatively large but also more robustly connected than G as a whole. That is, we aim to identify significant parts of the graph that stand out due to their connectedness. Note that size and connectedness are inversely related: while it is easy to construct a small U with highly connected vertices, a large U must also include loosely connected ones. We hence maximise their (weighted) multiplicative trade-off, called **density impact**, defined as

$$f_\kappa(U; \gamma) = f_\text{c}(U)^{(1-\gamma)} f_\text{d}(U)^{\gamma} \qquad \text{with } \gamma \in (0, 1) \,, \quad (1)$$

where γ is a **trade-off parameter** that tunes the importance between the **coverage term** $f_\text{c}(U) = {}^{|U|}/_{|V|}$, i.e., the portion of the graph covered by the subset U, and the **density term** $f_\text{d}(U)$, which increases as the vertices in U become more robustly connected. We proceed to give a precise definition of the density term based on the concept of k-cores [6].

We can formally measure how robustly connected an entity subset $U \subseteq V$ is by studying the connectivity of its **induced subgraph**, i.e., the subgraph $G[U] = (U, E(U))$, where $E(U) = \{(v, u) \in E \mid u, v \in U\}$ is the set of all edges with end-points in U. For a vertex v, we denote by $N(v) = \{u \in V \mid (u, v) \in E\}$ its **neighbours** in G and its **degree**, i.e., the number of its neighbours, by $\delta(v) = |N(v)|$. When a quantity refers to the induced graph $G[U]$ we indicate the inducing vertex set as a subscript. For instance, $\delta_U(u)$ denotes the degree of vertex u in the induced graph $G[U]$.

A **k-core component** of a graph G is an (inclusion-wise) maximal connected subgraph of G whose vertices U have all a degree of at least $\delta_U(u) \geq k$. The subgraph comprising all k-core components of this graph is called its **k-core** $H(k)$, and the **k-core vertices** $V(k)$ are the vertices of the graph's k-core. The last two definitions are then related as $H(k) = G[V(k)]$.

The annotated k-cores of the example graph on Fig. 2 show that the k-cores are nested to form a hierarchy over the vertices. We also define the **k-shell** of G as the set of vertices that lie in the k-core but not in the $k+1$-core (same-coloured vertices in the figure). In this way, the k-shells define a partitioning over the vertices: the **core decomposition** of G, which assigns to each vertex v a **core number** (or **coreness**)

$$\kappa(v) = \max\{k \mid v \in V(k)\} \,, \qquad \text{for } G, \text{ and}$$
$$\kappa_U(v) = \max\{k \mid v \in V_U(k)\} \,, \qquad \text{for } G[U] \,,$$

equal to the greatest number k such that this vertex lies in the k-core of G, where $V_U(k)$ are the k-core vertices of $G[U]$. Note that by definition $G[V] = G$, and hence $\kappa_V(v) = \kappa(v)$. Finally, the graph **degeneracy** $K = \max_{v \in V} \kappa(v)$ is the maximum coreness over all the vertices of the graph.

Using these quantities, we define the **average coreness** of G and $G[U]$, respectively, as the mean coreness of its vertices

$$\bar{\kappa} = \frac{\kappa_V}{|V|} \text{ and } \bar{\kappa}_U = \frac{\kappa_U}{|U|} \text{ with } \kappa_U = \sum_{v \in U} \kappa_U(v) \text{ for } U \subseteq V. \quad (2)$$

We hence quantify the amount to which a vertex set U is more robustly connected than G by the **coreness density**

$$f_\text{d}(U) = \bar{\kappa}_U - \bar{\kappa} \,,$$

which completes the definition of the density term of Eq. (1).

Note that graph coreness is related to various definitions of density [18]: high coreness indicates better connectedness. For instance, the minimum coreness in a graph bounds the number of edges that have to be removed for the subgraph to become disconnected. These traits underlie our notion of *robustness*.

III. Discovering Robust Describable Subgraphs

Our goal is to identify large and robustly connected vertex sets which have a simple description. Hence, in addition to the chosen optimisation function f_κ we need to fix a set of potential descriptions; this set is the **description language** $\mathcal{L}$.

A common way to define such a language is by considering all conjunctions $\pi_1 \wedge \ldots \wedge \pi_l$ that can be formed from a set of base predicates Π on vertex attributes, e.g., $[\texttt{age} > 18]$ or $[\texttt{sex} = \male]$, that are either given, or in case of ordinal or numeric features, automatically discovered during mining [21]. We refer to such a conjunction as a **selector** σ and to the vertices that satisfy it as the **extension** of σ, denoted $\text{ext}(\sigma) \subseteq$

V. We define the **value of a selector** $f_\kappa(\sigma) = f_\kappa(\mathrm{ext}(\sigma))$ to be the objective value of its extension. With this we can formally specify our problem as: *find an element* $\sigma^* \in \mathcal{L}$ *that solves*

$$\sigma^* \in \arg\max_{\sigma \in \mathcal{L}} f(\sigma) \ . \tag{3}$$

Although greedy algorithms are readily available to solve this problem, it can be shown that for particular inputs they yield a solution arbitrarily far from the optimal. In the next section, we develop an efficient algorithm that finds the optimal solution.

A. Solving Exactly with Branch–and–Bound

The established algorithm that solves problem (3) exactly is Branch–and–Bound (BnB). This algorithm uses two components: a refinement operator and an optimistic estimator.

A simple **refinement operator** $\rho : \mathcal{L} \to 2^{\mathcal{L}}$ can be formulated by extending a given selector with each unused predicate that respects a given lexicographic ordering. While simple, this operator is usually inefficient in practice, as the description language tends to contain many selectors that refer to the same vertex set, e.g., $\sigma_1 \equiv [\texttt{pregnant}]$ and $\sigma_2 \equiv [\texttt{pregnant}] \wedge [\texttt{sex} = \varphi]$. To avoid this redundancy one commonly uses the concept of a closure operator [20] $\mathbf{c}(\sigma) = \bigwedge \{\pi \in \Pi \mid \mathrm{ext}(\pi) \supseteq \mathrm{ext}(\sigma)\}$. We can now define

$$\rho(\sigma) = \left\{\psi = \mathbf{c}(\sigma \wedge \pi_i) \mid i_{\text{core}}(\sigma) < i \leq |\Pi| \wedge \psi|_{i-1} = \sigma|_{i-1}\right\}$$

where $$\sigma|_i = \bigwedge \{\pi_j : \pi_j \text{ occurs in } \sigma \text{ and } j \leq i\}$$

and $$i_{\text{core}}(\sigma) = \min\{i \mid \mathrm{ext}(\sigma|_i) = \mathrm{ext}(\sigma)\} \ .$$

This operator induces a tree over $\mathcal{L}$ that has at its root the selector σ_{root}: the empty conjunction, with $\mathrm{ext}(\sigma_{\text{root}}) = V$.

The second component of BnB—an admissible **optimistic estimator** $\hat{f}$ of an objective function f—is defined as

$$\hat{f}(U) \geq \max_{T \subseteq U} f(T), \qquad \forall U \subseteq V \ . \tag{4}$$

Naturally, the tighter the bound of the optimistic estimator the higher its pruning potential. This potential becomes optimal when Eq. (4) holds with equality; then we refer to $\hat{f}$ as the **tight optimistic estimator** [12] of the objective function f.

These components work as follows: the *refinement operator* defines such a tree over the selectors in $\mathcal{L}$, in which tree each child of a selector describes a subset of its parent's vertex set V. The *optimistic estimator* upper bounds the value of all possible subsets of V, thus also the value of all its descendants. We thus start from the root and traverse the selector tree, while keeping track of the best selector value encountered so far. If the optimistic estimator of a selector is below the current best value, none of its descendants can improve on this value, so we can safely prune its sub-branch.

B. Optimistic Estimators

To derive an optimistic estimator for our objective function, we need to show that it satisfies bound (1). A first bound can be derived by adapting ideas from rule mining (or subgroup discovery) on numerical unstructured data [11]. Here each entity v has a real-valued *target attribute* y and we aim to find a describable subset $U \subseteq V$ in which the mean value of y is maximal. Using coreness as the target attribute, the formal objective in this task becomes a static version f^s_κ of our f_κ:

$$f^s_\kappa(U) = \frac{|U|}{|V|}\left[\sum_{u \in U} \bar{\kappa}(u) - \bar{\kappa}\right] \qquad \text{with tight bound [7]}$$

$$\hat{f}^s_\kappa(U) = \max_{0 < i \leq |U|} \frac{i}{|V|}\left[\frac{1}{i}\sum_{j=1}^{i} \kappa\big(v^s_j\big) - \bar{\kappa}\right] , \tag{5}$$

where v^s_i are the vertices of U in descending order of $\kappa(v^s_i)$.

This static measure f^s_κ, however, systematically overestimates the subgraph density, as visualised in Fig. 3. This is due to a key observation for the rest of our analysis: the average coreness is monotone with respect to the inducing vertex set.

Lemma 1. *Let* $T \subseteq U$. *Then* $\bar{\kappa}_U(T) \coloneqq \dfrac{|T|}{|V|}\displaystyle\sum_{v \in T} \kappa_U(v) \geq \bar{\kappa}_T$.

More formally, f^s_κ overestimates f_κ, and therefore the optimistic estimator $\hat{f}^s_\kappa$ of f^s_κ is also a bound for our measure.

A more advanced bound can be derived by optimising the coreness of the induced graph directly. At the core of this optimistic estimator lies a tight upper bound for the total coreness κ_U of Eq. (2) over all subsets of U, written as

$$\kappa^*_U = \max_{T \subseteq U} \kappa_T = \max_{1 \leq i \leq |U|} \kappa^i_U, \quad \text{with } \kappa^i_U = \max_{T \subseteq U,\, |T| = i} \kappa_T \ .$$

To compute the maximum over all fixed cardinality subsets κ^i_U we first arrange all vertices $v_1, \ldots, v_{|U|}$ of U in order of decreasing coreness $\kappa_U(v_i)$ and observe that κ^i_U is upper bounded by the partial sums $\hat{\kappa}^i_U = \sum_{j=1}^{i} \kappa_U(v_j)$.

We now study the sequence of these partial sums $\hat{\kappa}^i_U$ as follows. Due to their ordering, the vertices are selected one k-shell of $G[U]$ at a time in decreasing order of k, so that within each k-shell the value of $\hat{\kappa}^i_U$ increases by a constant k. This constant changes right after each k-shell (or equivalently, k-core) is exhausted. There are $K_U + 1$ such **complete core addition indices**: each corresponds to exhausting the vertices of a k-core and thus coincides with the size of a k-core. We denote these as $n_k = |V_U(k)|$ for each k-core $0 \leq k \leq K_U + 1$.

Note that $\hat{\kappa}^i_U$ increases linearly between two consecutive complete core addition indices $n_{k+1} \leq i \leq n_i$ by exactly k. Thus, $\hat{\kappa}^i_U$ is a piece-wise linear sequence in i, whose pieces switch at indices $i = n_k$. The value of $\hat{\kappa}^i_U$ at each such index can be computed as the cumulative sum of k-shell sizes, each weighted by k; to compute the rest we use linear interpolation:

$$\hat{\kappa}^i_U = \begin{cases} \sum_{\lambda=k}^{K_U} \lambda(n_\lambda - n_{\lambda+1}) & i = n_k \\ & 0 \leq k \leq K_U \\ \dfrac{(i - n_{k+1})\hat{\kappa}^{n_k}_U + (n_k - i)\hat{\kappa}^{n_{k+1}}_U}{n_{k+1} - n_k} & n_{k+1} \leq i < n_k \\ & 0 \leq k \leq K_U \ . \end{cases}$$

Since $\hat{\kappa}^{n_k}_U = \hat{\kappa}^{n_{k+1}}_U + k(n_k - n_{k+1})$, the above is simplified as

$$\hat{\kappa}^i_U = (i - n_{k+1})k + \sum_{\lambda=k}^{K_U} \lambda(n_k - n_{k+1}) \ , \quad n_{k+1} \leq i \leq n_k. \tag{6}$$

Eq. (6) reveals $\hat{\kappa}^i_U$ to be piece-wise linear (and concave) function due to the monotonically decreasing increments k.

Algorithm 1: ROSI—discovering the top-κ subgraphs

Input: Result count κ, depth limit $d_{\max}$, approx. factor α
Output: Top-κ results R

1 $\tau \leftarrow -\infty$, $R \leftarrow \{\}$, $d_{\text{dfs}} \leftarrow 1$
2 **do**
3 `truncated` $\leftarrow$ FALSE
4 `stack` $\leftarrow \{(\sigma_{\text{root}}, 0)\}$
5 **while notEmpty**(`stack`) **do**
6 $(\sigma_{\text{cur}}, d_{\text{cur}}) \leftarrow$ **pop**(`stack`)
7 **for** $\sigma_{ref} \in \rho(\sigma_{cur})$ **do**
8 $f_{\text{ref}}, \hat{f}_{\text{ref}} \leftarrow \hat{f}(\sigma_{\text{ref}}), f_\kappa(\sigma_{\text{ref}})$
9 **if** $\hat{f}_{ref} > \alpha \cdot \tau$ **then**
10 $R, \tau \leftarrow$ **updateResults**$(R, \sigma_{\text{ref}}, f_{\text{ref}})$
11 **if** $d_{cur} < d_{dfs}$ **then**
12 **push**$\big($`stack`$, (s, d_{\text{cur}} + 1)\big)$
13 **else**
14 `truncated` $\leftarrow$ TRUE
15 $d_{\text{dfs}} \leftarrow d_{\text{dfs}} + 1$
16 **while** $d_{dfs} \leq d_{max}$ **and** `truncated`
17 **return** R

Truncated DFS

Each element of the sequence $\hat{\kappa}_U^i$ can now serve as an upper bound for the maximum total coreness κ_U^i over all subsets of U with a fixed cardinality of i.

Proposition 2. *For the piece-wise linear function of Eq.* (6)

1) $\kappa_U^i \leq \hat{\kappa}_U^i$, *for all* $0 \leq i \leq |U|$
2) $\kappa_U^i = \hat{\kappa}_U^i$, *for* $i \in \{0, n_0, \ldots, n_{K_U}\}$

Using the first part of Proposition 2 we can upper bound the value of f_κ^s over all subsets of U with cardinality i by

$$\hat{\phi}_U(i;\gamma) = \left(\frac{i}{|V|}\right)^{1-\gamma} \left(\frac{\hat{\kappa}_U^i}{i} - \bar{\kappa}\right)^{\gamma} . \tag{7}$$

Hence, the solution of Eq. (4) for $f_\kappa(U;\gamma)$ can be written as

$$\max_{T \subseteq U} f_\kappa(T;\gamma) \leq \hat{\phi}_U^*(\gamma) = \max_{0<i\leq|U|} \hat{\phi}_U(i;\gamma) .$$

Finally, we replace Eq. (7) into the one above and then use Proposition 2 (part 2) to show that our final bound is tight.

Corollary 3. *The quantity* $\hat{\phi}_U^*(\gamma)$ *is an optimistic estimator of* $f_\kappa(U;\gamma)$. *In addition,* $\hat{\phi}_U^*$ *becomes tight for* $\gamma = 1/2$.

$$\hat{\phi}_U^*(\gamma) = \max_{0<i\leq|U|} \left(\frac{i}{|V|}\right)^{1-\gamma} \left(\frac{\hat{\kappa}_U^i}{i} - \bar{\kappa}\right)^{\gamma} . \tag{8}$$

Our proposed bound (8) can be computed in linear time: the k-core decomposition of G is in $O(n)$ [5], and the maximum in Eq. (8) compares $|U|$ values, each computable in $O(1)$.

C. Discovering the Top-κ Subgraphs

We next describe **Ro**bustly–Connected **S**ubgraphs with Descr**i**ptions (ROSI), the complete algorithm that finds the top-κ describable subgraphs within language $\mathcal{L}$ that maximise f_κ.

ROSI is an implementation of the *iterative deepening depth first search* variant of BNB [15]. In particular, it repeatedly invokes a truncated (i.e., depth-limited) depth first search (DFS) for increasing depth limits until all reachable nodes have been traversed. This algorithm constitutes a hybrid of depth-first and breadth-first search; as such it combines the minimal memory footprint of DFS while it avoids spending excessive time in few—possibly sub-optimal—deep branches.

Starting with a permissive pruning threshold and empty result set (line 1) ROSI repeatedly invokes the inner truncated DFS (ln. 3-16). The latter traverses the tree induced over $\mathcal{L}$ by the refinement operator ρ (ln. 7) starting with the root selector σ_{root} (ln. 4). During traversal, sub-optimal refinements (ln. 9) are dropped, while **updateResults** (ln. 10) checks if the rest can improve on the so-far best value τ. If they do, the top-κ results R are updated to contain the better selector and τ is updated to the value of the worst result $\tau \leftarrow \min\{f_\kappa(\sigma) \mid \sigma \in R\}$. In this fashion, although consecutive DFS invocations restart from s_0, as time progresses τ increases and more nodes get pruned. This repeats until DFS completes un-truncated, i.e., all reachable refinements have been traversed (ln. 14,16).

While the inner for-loop (ln. 7) has a linear complexity, the algorithm is a typical NP-hard one. If required, however, ROSI can terminate after a finite depth limit $d_{\max}$, which corresponds to finding the optimal description with at most $d_{\max}$ predicates. Additionally, the otherwise exact algorithm turns into an α-approximation one by setting the approximation factor $\alpha < 1$.

IV. RELATED WORK

Dense Subgraphs and Communities. The typical objective in *dense subgraph discovery* is to find the subset of vertices in a non-attributed graph that induces the subgraph with the highest edge-to-vertex ratio. Building on this rather simplistic notion, a plethora of works reinterpret density to take into account structural information, for instance, using triangle counts, k-cliques, and k-cores [18]. In the related yet different *community detection*, we impose the additional constraint that the discovered subgraph be disconnected with the rest of the graph, which usually incurs the need for combinatorial optimisation [9]. Note that ROSI *solves the former* task, by adapting a k-core–based measure for mining *named patterns*.

Moving on to methods which use graph attributes, we first classify them as those using graph attributes to steer a density optimisation scheme toward **cohesive subgraphs**, i.e., subgraphs with similar attributes, or others that seek the densest out of a set of **subgroups**, i.e., subgraphs described based on graph attributes, to which ROSI also belongs.

Cohesive Subgraphs. COPAM [16] applies *subspace clustering* on the vertex attributes to find maximal connected subgraphs that contain vertices with similar attributes, whose density surpasses a given threshold. Similarly, [13] (GAMER) discover non-redundant sets of subgraphs, which must be connected γ-quasi-cliques for a given parameter γ. Note that for both methods the respective density score needs only surpass a user-defined threshold and does not contribute to the quality of each subgraph any further. More recently, [17] (AMEN)

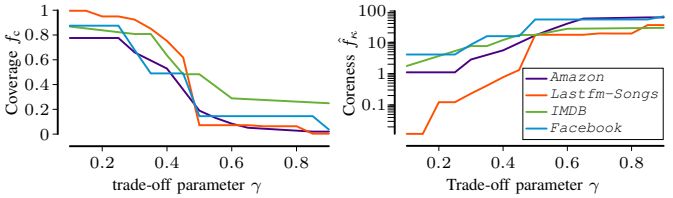


Figure 4 [Coreness vs. Coverage]: Increasing the trade-off parameter yields smaller but more robustly connected subgraphs.

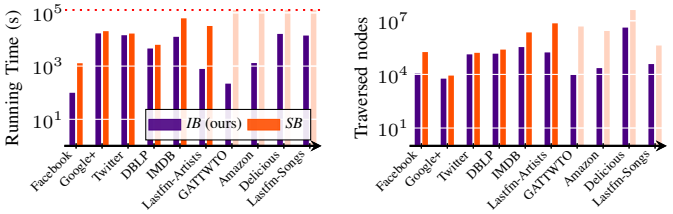
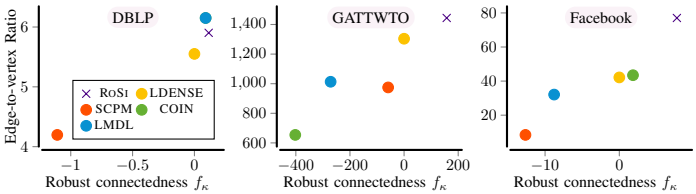
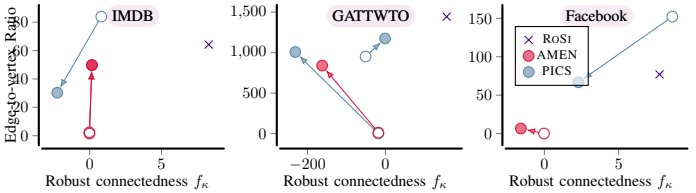


Figure 5 [↓ is better]: Our bound is tight: it prunes optimally and is faster. Runs exceeding 36 hours (dotted line) are faded.



(a) *Comparison to methods which provide descriptions.*



(b) *Description-free methods: a hollow mark designates the descriptionless result; arrows point to the closest describable subgroup.*

Figure 6 [↗ is better]: Scores of top discovered pattern. RoSI is the best in terms of robustness and competitive on density.

introduce an attribute-aware variant of the established modularity measure [9] to detect ego-net–shaped communities with similar attributes. These last three methods score each mined pattern individually. In contrast, the *subgraph clustering* PICS of [2] uses low entropy splits of the binary adjacency and attribute matrices to form vertex clusters with similar concentration of edges and binary features. We compare to the most recent works of both tasks.

Subgroup Discovery. This task aims to provide exact descriptions of dataset parts which exhibit exceptional behaviour of a target concept when compared to the entire dataset [21]. Such target concepts can be the distribution of single or multiple variables, which can take discrete [1] or continuous values [11], or more recently incorporate fairness [14] or constitute the learned parameters of regression models [8].

RoSI finds subgraphs with an exceptional target concept of robustness. Perhaps the closest to our work is SCPM [19] with a structured density based on quasi-cliques, which must be sampled from each subgraph to estimate how many of its vertices these cliques cover. This method needs many hard-to-specify parameters, is only approximate and, as our experiments show, slower than RoSI. Although faster, LDENSE [10] is a *greedy* search for the describable subgraph with the highest typical density. Less related are methods solving the community detection problem, instead. For instance, [3] also use BnB for exhaustive search but with target concepts for community detection (LMDL, COIN, etc).

V. Experiments

In this section we empirically evaluate RoSI.[2] We consider 10 datasets that together span multiple domains and different kinds of represented entities and relations from public sources with up to thousands of vertices and millions of edges. These consist of both graphs and multi-graphs, and describe various types of networks: social, similarity, co-occurrence, collaboration networks, etc.

The Generality–Connectedness Trade-Off: We next demonstrate the effect of the trade-off parameter γ, which offers at once a smooth and intuitive mechanism to tune the importance between the size (coverage) and the connectedness (density) of the discovered subgraph. We study datasets with highly diverse base predicates that allow the greatest flexibility in the resulting descriptions, and mine the top result for increasing values $\gamma \in \{0.1, 0.15, \ldots, 0.9\}$ and plot the coverage and connectedness of the topmost result (Fig. 4).

Continuously increasing parameter γ leads to smaller and more densely connected subgraphs—it thus intuitively steers the results toward more general or more connected subgraphs.

Efficiency of RoSI: We first study how the efficiency of RoSI is affected by the pruning potential of our proposed ***induced-bound*** (8) (IB) against the baseline, ***static-bound*** (5) (SB). For the experiments we use the default trade-off parameter of $\gamma = 1/2$; when the computation time (for IB) exceeds 7 hours, we lower the approximation factor α by 0.1 or decrease the depth limit, favouring a deeper search when possible. We report the wall-clock times and traversed nodes in Fig. 5.

The experiments corroborate our expectations due to theory: since our bound is tight, it prunes optimally. Since, in addition, its complexity is $O(n)$, which matches the complexity of each iteration, the superior pruning capacity is readily translated to shorter run-times. At the Importantly, the same trait of $\hat{f}_{\text{IB}}$ allows to practically optimise large real-world graphs, which would otherwise be impossible within the allowed time limit of 36 hours. Note that, since RoSI uses the RoSI scheme, it does not suffer from memory issues, which in other schemes would be a pronounced problem for the lesser pruning scheme.

Optimality of RoSI: Here we compare RoSI to representative works in terms of both our proposed robust connectedness and also typical density (edge-to-vertex ratio), Although our

[2]Code and data are available at http://eda.mmci.uni-saarland.de/rosi/.

γ	Description							Movies	Dens.	Cov.
	Drama	Comedy	¬BFI	debut $\leq$'96	debut $\leq$'05	US	...			
[0.1 −0.3)	✓							20 579	1.76	0.868
[0.3 −0.4)			✓		✓			19 150	7.59	0.808
[0.4 −0.45)			✓	✓	✓			15 057	11.85	0.635
[0.45−0.6)	✓	✓	✓	✓	✓			11 455	17.14	0.483
[0.6 −0.9]	✓	✓	✓	✓	✓	✓		6 843	27.05	0.289

Table I: The top discovered subgraphs from *IMDB* tell a story.

task is dense subgraph discovery, we also compare against more loosely related approaches for community detection.

We first compare against state-of-the-art methods which describe the found patterns: LDENSE [10], SCPM [19], and two target concepts for community detection from subgroup discovery on graphs: COIN [4]) and local modularity (LMDL [3]). We plot the best results of each method in Fig. 6a. ROSI scores the highest in terms of robust connectedness, while in terms of density it is on par with the rest.

We further compare ROSI with two recent methods for cohesive subgroups: PICS and AMEN, neither of which do not provide descriptions. Since these methods output several patterns, we show all discovered vertex sets in the Pareto front of the two metrics (Fig. 6b) with empty circles, designating the absence of a description. Although rarely, other methods may score a higher density and even robustness than ROSI, as their optimisation not constrained. To put them in perspective, however, we further mine the closest subgroup in terms of Jaccard distance to the one provided by each algorithm, and link to it the unconstrained solution with an arrow. As expected, these solutions score lower than those of ROSI.

Interpretable Subgraph Descriptions: To qualitatively assess our results we mine the top describable subgraph for the *IMDB* dataset which offers attributes that are easily interpretable for a lay person: collaborations of cast members. We track the top mined description over a varying $0.1 \leq \gamma \leq 0.9$ in Table I.

Starting with larger subgraphs (high γ) we read the first result as: *the drama movie cast has a robust connectedness of* 1.8 *collaborations on average more than what is usual in the entire industry*. Moving into denser graphs, we find that established actors (i.e., debutting before '96) collaborate well with each other. Here, also a negated predicate is informative: the London BFI festival is known to nominate more diverse films, with cast harder to have collaborate with each other, therefore removing it increases connectedness. We further find that additionally producing a movie in the US leads to substantially higher connectedness. Overall, the discovered patterns reveal an interpretable and actionable story.

VI. CONCLUSION

We studied the problem of finding robustly connected subgraphs that are easily described. We measure this property by a coreness-based score that ranks highly those subgraphs that contain node clusters that are difficult to shatter. We used a description language that comprises all logical conjunctions over predicates derived from node attributes. We then showed how to find a vertex set a) whose induced subgraph maximises this measure of robust connectedness subject to b) accepting a simple description from this language.

Due to the combinatorial nature of this problem, to solve it exactly we use ROSI, the iterative deepening variant of BNB, which we further improve to efficiently overcome redundant descriptions in our language. For its use we also develop an optimistic estimator which is optimal in the default configuration. Importantly, ROSI can also work as a tunable any-time approximate algorithm.

Our experiments show that, although our problem is inherently exponential, ROSI can analyse real-world graphs with up to millions of edges and tens of thousands of vertices within reasonable time. Importantly, the results are meaningful and easily interpretable.

REFERENCES

[1] T. Abudawood and P. Flach, "Evaluation measures for multi-class subgroup discovery," in *ECML PKDD*. Springer, 2009, pp. 35–50.

[2] L. Akoglu, H. Tong, B. Meeder, and C. Faloutsos, "PICS: Parameter-free identification of cohesive subgroups in large attributed graphs," in *SDM*. SIAM, 2012.

[3] M. Atzmueller, S. Doerfel, and F. Mitzlaff, "Description-oriented community detection using exhaustive subgroup discovery," *Inf. Sci.*, pp. 965–984, 2016.

[4] M. Atzmueller and F. Mitzlaff, "Efficient Descriptive Community Mining," in *FLAIRS*, 2011.

[5] V. Batagelj and M. Zaversnik, "An O(m) Algorithm for Cores Decomposition of Networks," *arXiv:cs/0310049*, 2003.

[6] A. Bickle, *The K-Cores of a Graph*. Western Michigan University, 2010.

[7] M. Boley, B. R. Goldsmith, L. M. Ghiringhelli, and J. Vreeken, "Identifying Consistent Statements about Numerical Data with Dispersion-Corrected Subgroup Discovery," *DAMI*, pp. 1391–1418, 2017.

[8] W. Duivesteijn, A. J. Feelders, and A. Knobbe, "Exceptional Model Mining: Supervised descriptive local pattern mining with complex target concepts," *DAMI*, pp. 47–98, 2016.

[9] S. Fortunato and D. Hric, "Community detection in networks: A user guide," *Phys. Rep.*, pp. 1–44, 2016.

[10] E. Galbrun, A. Gionis, and N. Tatti, "Top-k overlapping densest subgraphs," *DAMI*, 2016.

[11] H. Grosskreutz and S. Rüping, "On subgroup discovery in numerical domains," *DAMI*, pp. 210–226, 2009.

[12] H. Grosskreutz, S. Rüping, and S. Wrobel, "Tight optimistic estimates for fast subgroup discovery," in *ECML PKDD*. Springer, 2008.

[13] S. Gunnemann, I. Farber, B. Boden, and T. Seidl, "Subspace Clustering Meets Dense Subgraph Mining: A Synthesis of Two Paradigms," in *ICDM*. IEEE, 2010.

[14] J. Kalofolias, M. Boley, and J. Vreeken, "Efficiently Discovering Locally Exceptional Yet Globally Representative Subgroups," in *ICDM*, 2017.

[15] R. E. Korf, "Depth-first Iterative-deepening: An Optimal Admissible Tree Search," *Artif. Intell.*, pp. 97–109, 1985.

[16] F. Moser, R. Colak, A. Rafiey, and M. Ester, "Mining Cohesive Patterns from Graphs with Feature Vectors," in *SDM*. SIAM, 2009, pp. 593–604.

[17] B. Perozzi and L. Akoglu, "Discovering Communities and Anomalies in Attributed Graphs: Interactive Visual Exploration and Summarization," *ACM TKDD*, pp. 24:1–24:40, Jan. 2018.

[18] K. Shin, T. Eliassi-Rad, and C. Faloutsos, "CoreScope: Graph Mining Using k-Core Analysis—Patterns, Anomalies and Algorithms," in *ICDM*. IEEE, 2016, pp. 469–478.

[19] A. Silva, W. Meira, Jr., and M. J. Zaki, "Mining Attribute-structure Correlated Patterns in Large Attributed Graphs," *VLDB*, 2012.

[20] T. Uno, T. Asai, Y. Uchida, and H. Arimura, "LCM: An Efficient Algorithm for Enumerating Frequent Closed Item Sets." in *ICDM*, 2003.

[21] S. Wrobel, "An algorithm for multi-relational discovery of subgroups," in *PKDD*. Springer, 1997, pp. 78–87.