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.oley@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.

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

http://eda.mmci.uni-saarland.de/rosi/
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 \( x_i : V \to X_i \) from vertices to a continuous or categorical domain \( X_i \). These attributes can be used to simply describe subsets based on logical expressions of vertices \( v \in V \) like \( \sigma(v) \equiv \lbrack \text{age} (v) \geq 18 \rbrack \land \lbrack \text{sex} (v) = \text{'female'} \rbrack \).

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

\[
f_\kappa(U; \gamma) = f_c(U)^{(1-\gamma)} f_d(U)^\gamma \quad \text{with} \quad \gamma \in (0, 1), \quad (1)
\]

where \( \gamma \) is a trade-off parameter that tunes the importance between the coverage term \( f_c(U) = |U|/|V| \), i.e., the portion of the graph covered by the subset \( U \), and the density term \( f_d(U) \), 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 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 \textit{k-core} \( H(k) \), and the \textit{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 \textit{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 \textit{core decomposition} of \( G \), which assigns to each vertex \( v \) a \textit{core number} (or coreness) \( \kappa(v) \) equal to the greatest number \( k \) such that this vertex lies in the \( k \)-core of \( G \), where \( V_k(k) \) are the \( k \)-core vertices of \( G[U] \). Note that by definition \( G[V] = G \), and hence \( \kappa_V(v) = \kappa_U(v) \).

Finally, the graph \textit{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 \textit{average coreness} of \( G \) and \( G[U] \), respectively, as the mean coreness of its vertices

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

We hence quantify the amount to which a vertex set \( U \) is more robustly connected than \( G \) by the \textit{coreness density}

\[
f_d(U) = \bar{\kappa}_U - \bar{\kappa},
\]

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

Note that graph coreness is related to various definitions of density [13]: 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 \( f \), we need to fix a set of potential descriptions; this set is the \textit{description language} \( \mathcal{L} \).

A common way to define such a language is by considering all conjunctions \( \pi_1 \land \ldots \land \pi_n \) that can be formed from a set of base predicates \( \Pi \) on vertex attributes, e.g., \( [\text{age} \geq 18] \) or \( [\text{sex} = \sigma] \), 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 \textit{selector} \( \sigma \) and to the vertices that satisfy it as the \textit{extension} of \( \sigma \), denoted \( \text{ext}(\sigma) \subseteq V \).

\[
\begin{array}{c|c|c|c|c|c|c|c|}
\hline
\text{on dens.} & \text{G}_1 & \text{G}_2 & \text{G}_3 & \text{on dens.} \\
\hline
\text{G}_1 & 0 & \text{G} & 4 & \text{G}_2 & 0 & \text{G} & 4 \\
\text{G}_3 & 2 & \text{G} & 3 & \text{G}_3 & 2 & \text{G} & 3 \\
\hline
\end{array}
\]
We define the value of a selector $f_v(\sigma) = f_v(\text{ext}(\sigma))$ to be the objective value of its extension. With this we can formally specify our problem as: \textit{find an element $\sigma^* \in \mathcal{L}$ that solves}

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

(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 $\rho : \mathcal{L} \rightarrow 2^\mathcal{E}$ 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., $\sigma_1 \equiv [\text{pregnant}]$ and $\sigma_2 \equiv [\text{pregnant}] \land [\text{sex} = \varnothing]$. To avoid this redundancy one commonly uses the concept of a closure operator \cite{20} $c(\sigma) = \bigwedge \{ \pi \in \Pi | \text{ext}(\pi) \supseteq \text{ext}(\sigma) \}$. We can now define

$$
\rho(\sigma) = \{ \psi = c(\sigma \land \pi_i) | i_{\text{core}}(\sigma) < i \leq |\Pi| \land \psi_{i-1} = \sigma_{i-1} \}
$$

where

$$
i_{\text{core}}(\sigma) = \min \{ i | \text{ext}(\sigma_i) = \text{ext}(\sigma) \} .
$$

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

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

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

(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 \cite{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 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 \cite{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 $f_v^*$ of our $f_v$:

$$
f_v^*(U) = \frac{|U|}{|V|} \sum_{u \in U} \hat{k}(u) - \bar{k}
$$

(7)

with tight bound

$$
\hat{f}_v^*(U) = \max_{0 < i \leq |U|} \frac{i}{|V|} \left[ \frac{1}{i} \sum_{j=1}^{i} \kappa(v_j^*) - \bar{k} \right] ,
$$

(5)

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

This static measure $f_v^*$, 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 $\hat{k}_U(T) \equiv \frac{|T|}{|V|} \sum_{v \in T} \kappa_U(v) \geq \bar{k}_T$.

More formally, $f_v^*$ overestimates $f_v$, and therefore the optimistic estimator $\hat{f}_v^*$ of $f_v^*$ 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 $\kappa_U$ of Eq. (4) over all subsets of $U$, written as

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

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

We now study the sequence of these partial sums $\hat{k}_v(i)$ 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 $\hat{k}_v(i)$ increases by a constant $k$. This constant changes right after each $k$-shell (or equivalently, $k$-core) is exhausted. There are $|U| + 1$ 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 $\hat{k}_v(i)$ increases linearly between two consecutive complete core addition indices $n_{k+1} \leq i \leq n_k$, by exactly $k$. Thus, $\hat{k}_v^i$ is a piece-wise linear sequence in $i$, whose pieces switch at indices $i = n_k$. The value of $\hat{k}_v^i$ 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:

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

(6)

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

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

Eq. (6) reveals $\hat{k}_v^i$ to be piece-wise linear (and concave) function due to the monotonically decreasing increments $k$. 
Each element of the sequence $\hat{\kappa}_i^\kappa$ can now serve as an upper bound for the maximum total coreness $\kappa_i^\kappa$ over all subsets of $U$ with a fixed cardinality of $i$.

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

1. $\kappa_i^\kappa \leq \hat{\kappa}_i^\kappa$, for all $0 \leq i \leq |U|$
2. $\kappa_i^\kappa = \hat{\kappa}_i^\kappa$, for $i \in \{0, n_0, \ldots, n_{\kappa_U}\}$

Using the first part of Proposition 2 we can upper bound the value of $f_*^\kappa$ over all subsets of $U$ with cardinality $i$ by

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

(7)

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

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

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

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

(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-\(k\) Subgraphs**

We next describe Robustly–Connected Subgraphs with Descriptions (RoSi), the complete algorithm that finds the top-

$k$-describable subgraphs within language $L$ that maximise $f_*^\kappa$.

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—deeper 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 $L$ by the refinement operator $\sigma$ (ln. 7) starting with the root selector $\sigma_{\text{root}}$ (ln. 4). During traversal, sub-optimal refinements (ln. 2) are dropped, while updateResults (ln. 10) checks if the rest can improve on the so-far best value $\tau$ . If they do, the top-$k$ results $R$ are updated to contain the better selector and $\tau$ is updated to the value of the worst result $\tau \leftarrow \min\{f_*(\sigma) \mid \sigma \in R\}$. In this fashion, although consecutive DFS invocations restart from $s_0$, as time progresses $\tau$ 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 $d_{\text{max}}$, which corresponds to finding the optimal description with at most $d_{\text{max}}$ predicates. Additionally, the otherwise exact algorithm turns into an $\alpha$-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.** COAP [15] 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 $\gamma$-quasi-cliques for a given parameter $\gamma$. 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)
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 [11] 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, [8] also use BnB for exhaustive search but with target concepts for community detection (LMDL, COIN, etc).

V. EXPERIMENTS

In this section we empirically evaluate RoSi2. 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 $\gamma$, 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 $\gamma$ 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 (IB) against the baseline, static-bound (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 $\alpha$ 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 $f_B$ 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

---

2Code and data are available at http://eda.mmci.uni-saarland.de/rosi/
Moving into denser graphs, we find that patterns reveal an interpretable and actionable story. Additionally producing a movie in the US leads to a decrease in connectedness. We further find that established actors (i.e., debutting before '96) collaborate well with each other. Here, also a negated predicate is informative: collaborations of cast members. We track the top grossing films, with cast harder to have collaborate with each other. 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 rank the top described subgraph over a varying $0.1 \leq \gamma \leq 0.9$ in Table I.

Starting with larger subgraphs (high $\gamma$) 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 conjuctions 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