Discovering Robustly Connected Subgraphs with Simple Descriptions

Janis Kalofolias°, Mario Boley•, and Jilles Vreeken°

°CISPA Helmholtz Center for Information Security, Saarbrücken, Germany
•Monash University, Melbourne, Australia

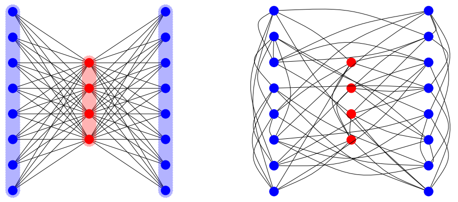
Abstract. We study the problem of discovering *robustly* connected subgraphs that have *simple* descriptions. That is, our aim is to discover sets of nodes for which the induced subgraph is not only difficult to fragment into disconnected components, but for which the nodes can also be selected from the entire graph with just a simple conjunctive query on the vertex attributes. As many subgraphs do not have such a simple logical description, first mining robust subgraphs and then 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 us to prune large parts of the search space. Through extensive empirical evaluation we show that our method can consider large real-world graphs, and discovers not only easily interpretable but also meaningful subgraphs.

1 Introduction

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

In this paper we break with this premise. We argue that from a knowledge discovery viewpoint subgraphs whose vertices are arbitrarily chosen to maximise this 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 we can select out of the entire graph with a conjunctive query on the vertex attributes. By admitting such a simple description, the subgraphs we discover are easily interpretable: from IMDB data, for example, we discover that mainstream movie crew with over 15 years experience have collaborated together more than is usual in the movie 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 naïve



(a) *complete bipartite graph* *edge/vertex ratio:* 3.2—*coreness:* 4.

(b) *6-regular graph (also a* 6*-core)* *edge/vertex ratio:* 3—*coreness:* 6.

Fig. 1 [Edge/vertex–ratio vs. robust connectedness]: Although graph (a) is more densely connected than (b), graph (b) is much more *robustly* connected than (a): While we can make (a) *fully disconnected* by removing just its 4 central nodes, to achieve the same for (b) we need to remove 19 vertices.

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 dense; 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 dense: 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 discovery 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 miserably in practice. Therefore, to mine large attributed graphs in reasonable time with guarantees, we propose a tight optimistic estimator and a non-redundant variant of branch-and-bound search. Through extensive experiments on ten large and diverse real-world graphs we show that our method, RoSi, performs very well in practice, discovering meaningful subgraphs where more naive strategies run out of time and memory. Importantly, these experiments also show that the above toy example is not esoteric: among the densest subgraphs that the recent method by Galbrun et al. [13] discovers on the DBLP dataset is a graph with average density of 42, but a robustness of 0 (!).

The roadmap of this paper is as follows. Next, we discuss how we can measure the robustness of a subgraph. In Sect. 3 we introduce our approach to efficiently searching for robust subgraphs with simple descriptions. We discuss related work in Sect. 4, and empirically evaluate RoSi in Sect. 5. Finally, we round up with discussion and conclusions in Sect. 6.

In the interest of readability and space, we postpone the proofs to our claims to the online appendix.[1]

2 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 multiplicative trade-off, inspired by the impact concept in mechanics, which we refer to as the **density impact function**. This score takes the form of the weighted geometric mean

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

where γ is a **trade-off parameter** that tunes the importance between the **coverage term** $f_\mathrm{c}(U) = {}^{|U|}/_{|V|}$, i.e., the portion of the graph covered by the subset U, and the **density term** $f_\mathrm{d}(U)$, 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 [7].

2.1 Core Decomposition: *k*-Cores, Degeneracy, and Coreness

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 define by $N(v) = \{u \in V \mid (u, v) \in E\}$ its **neighbours** in G and its **degree** as the number of its neighbours $\delta(v) = |N(v)|$. We indicate that a quantity refers to the induced graph $G[U]$ by marking 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

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

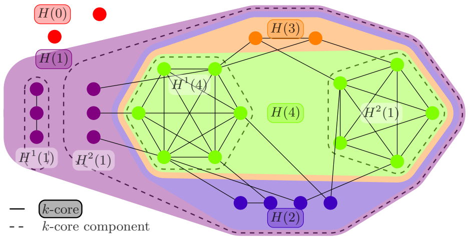


Fig. 3 [Higher coreness coincides with higher density.]: The core decomposition of a graph hierarchically groups its vertices into increasingly denser subgraphs. Here $H(k)$ denotes a k-core and $H^i(k)$ the i-th k-core component.

subgraph that consists of 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. Formally, we can write $H(k) = G[V(k)]$, where the k-core vertices are

$$V(k) = \{v \in V \mid v \text{ belongs to a } k\text{-core component}\} \ .$$

The annotated k-cores of the example graph on Fig. 3 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; in the figure each k-shell consists of the same-coloured vertices. In this way, the k-shells define a partitioning over the vertices, the **core decomposition** of G. This decomposition assigns to each vertex v a **core number** (or **coreness**)

$$\kappa(v) = \max\{k \mid v \in V(k)\} \ ,$$

equal to the greatest number k such that this vertex lies in the k-core of G. As usual, the core number of an induced graph $G[U]$ is

$$\kappa_U(v) = \max\{k \mid v \in V_U(k)\} \ ,$$

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**

$$K = \max_{v \in V} \kappa(v) \tag{2}$$

is the maximum coreness over all the vertices of the graph.

The coreness of a subgraph is closely related to different definitions of density [25,27]. Importantly, high coreness indicates high robustness, since the minimum core number in a subgraph bounds the number of edges that have to be removed until the subgraph becomes disconnected. This property, also known as k-edge connectedness [19, chap. 2.3], underlies our notion of *robustness*.

2.2 The Coreness Impact Function

We now use the relation between coreness and robust connectedness to define a density term f_d that quantifies this property for a (sub-)graph. We define the **average coreness** of G to be the mean of the core values of its vertices

$$\bar{\kappa} = \frac{1}{|V|} \sum_{v \in V} \kappa(v) \ . \tag{3}$$

As usual, computing the core values of this average on $G[U]$ gives

$$\bar{\kappa}_U = \frac{1}{|U|} \sum_{v \in U} \kappa_U(v) \qquad \text{for } U \subseteq V \ . \tag{4}$$

We hence quantify the degree to which a vertex set U is more robustly connected than G on average as the **coreness density**

$$f_{\mathrm{d}}(U) = \bar{\kappa}_U - \bar{\kappa} \ . \tag{5}$$

This quantity assigns a density of $f_{\mathrm{d}}(V) = 0$ to the full graph and is also intuitively interpretable as the extra average coreness of $G[U]$ compared to that of G. Finally, we can now use Eq. (5) as our definition for the density term in Eq. (1). In summary, we end up with the **coreness impact** of a vertex set defined as:

$$f_\kappa(U;\gamma) = \left(\frac{|U|}{|V|}\right)^{1-\gamma} \left(\bar{\kappa}_U - \bar{\kappa}\right)^{\gamma} \qquad \text{with } \gamma \in (0,1) \ . \tag{6}$$

Note that this measure is related yet different from the one typically used in rule mining (or subgroup discovery) for numerical unstructured data [28,14]. In this setting, a real-valued *target attribute* y is defined for each entity v, and we aim to find a describable subset of V which maximises the difference in mean of y within a subset $U \subseteq V$ and the entire V. With this approach, one can approximate the coreness impact function by using $y(v) = \kappa(v)$, the vertex coreness with respect to G. This yields a *static* version f_κ^s of Eq. (6), whose average coreness $\bar{\kappa}_U$ is now computed with respect to G. Formally, this quantity is denoted as $\bar{\kappa}_V(U)$, using an extension of Eq. (4) that further specifies the vertex set T whose core values we average:

$$\bar{\kappa}_U(T) = \frac{1}{|U|} \sum_{v \in T} \kappa_U(v) \ , \qquad \text{for all } T \subseteq U \subseteq V \ . \tag{7}$$

Although this static measure f_κ^s can be optimised using existing techniques, it systematically overestimates the subgraph density, as visualised in Fig. 4. This happens because the average coreness of Eq. (7) is monotone with respect to the inducing vertex set. This is a key observation to our analysis. Therefore we note:

Lemma 1. *Let T be a subset of U. Then $\bar{\kappa}_T(T) \leq \bar{\kappa}_U(T)$.*

3 Discovering Robust Subgraphs that Have Simple Descriptions

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: the **description language** $\mathcal{L}$.

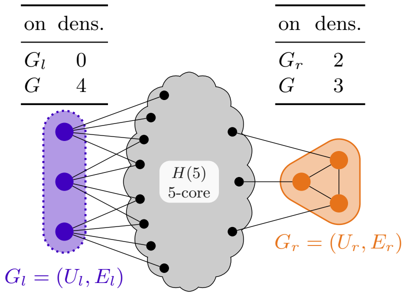


Fig. 4: The average subgraph coreness $\bar{\kappa}_U = \bar{\kappa}_U(U)$ may be misleadingly overestimated when it is computed on the whole graph $\bar{\kappa}_V(U)$. Here, subgraph G_r is denser than G_l with $\bar{\kappa}_{U_r} = 2 > 0 = \bar{\kappa}_{U_l}$. However, counting the edges of G, the subgraph densities falsely indicate the opposite relation.

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., [`age` > 18] or [`sex` $=$ '*male*'], that are either given, or in case of ordinal or numeric features, automatically discovered during mining [29]. We refer to such a conjunction as a **selector** σ and to the vertices that satisfy it as the **extension** of σ, denoted $\mathrm{ext}(\sigma) \subseteq V$. 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 our formal problem specification becomes: *find within the language a selector* σ^* *that attains the highest value*

$$\sigma^* \in \arg\max_{\sigma \in \mathcal{L}} f(\sigma) \; . \tag{8}$$

While greedy algorithms are readily available to solve this problem, our objective is neither anti-monotone nor sub-modular, and their solution can be arbitrarily far from the optimal. An exact method, however, not only finds higher quality results, but also allows to rule out the existence of robustly connected subgraphs within $\mathcal{L}$. This is particularly important for applications that require definite information, e.g, scientific discovery [9]. In the next sections we develop an efficient algorithm to solve problem (8) exactly.

3.1 Solving Exactly with Branch–and–Bound

The established algorithm that solves problem (8) exactly is Branch–and–Bound (BnB) [21]. This algorithm is based on 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:

$$\rho(\sigma) = \{\sigma \wedge \pi_i \mid i_{\max}(\sigma) < i \leq |\Pi|\}\,, \;\; i_{\max}(\sigma) = \max\{i \mid \pi_i \in \sigma\}\,.$$

This operator induces a tree over $\mathcal{L}$ that has at its root the selector σ_{root}: the empty conjunction, whose extension is the entire V.

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

$$\hat{f}(U) \geq \max_{T \subseteq U} f(T), \qquad \forall U \subseteq V\ . \tag{9}$$

Naturally, the tighter the bound of the optimistic estimator the higher its pruning potential. This potential becomes optimal when Eq. (9) holds with equality; then we refer to $\hat{f}$ as the **tight optimistic estimator** [15] of the objective function f.

These components work as follows: the *refinement operator* defines a search tree over the language $\mathcal{L}$ in a way that each child of a selector describes a subset of its parent's vertex set. At the same time, the *optimistic estimator* of a vertex set V upper bounds the value of all possible subsets of V. These components are then combined as follows: We start from the root and traverse the search tree, while keeping track of the best selector value encountered so far. For each child selector we evaluate the optimistic estimator; if this value is below the current best, no descendant can improve on the current best, and the entire sub-branch can be safely pruned.

Note that both the objective value and the optimistic estimator must be computed once per iteration. In each iteration the creation of the next studied refinement selector happens in (amortised) linear time. Therefore, to avoid that the algorithm changes asymptotic complexity, we require the bound to also be computable in $O(n)$.

In summary, to apply BnB we need a) a refinement operator ρ, and b) an optimistic estimator, ideally computable in $O(n)$.

3.2 Optimistic Estimators

To derive optimistic estimators for the coreness impact function, we show that they satisfy definition (9). Let U be any subset of V; to get a first solution of this definition we use Lemma 1 as follows.

$$\begin{aligned}\max_{T \subseteq U} f_\kappa(T) &\leq \max_{T \subseteq U} \frac{|T|}{|V|} \max_{T \subseteq U} \left(\bar{\kappa}_T - \bar{\kappa}\right) \leq \max_{T \subseteq U} \frac{|T|}{|V|} \left(\max_{T \subseteq U} \bar{\kappa}_V(T) - \bar{\kappa}\right) \\ &= \frac{|U|}{|V|} \left(\max_{u \in T} \kappa(u) - \bar{\kappa}\right) \leq \frac{|U|}{|V|} \left(\max_{u \in V} \kappa(u) - \bar{\kappa}\right) = \frac{|U|}{|V|} \left(K - \bar{\kappa}\right)\ ,\end{aligned} \tag{10}$$

where the second inequality follows from Lemma 1, in the next equality we maximise the average coreness of U by selecting the single vertex with the largest core value, and in the last equality we use the definition of degeneracy given in Eq. (2). Due to the monotonicity of a positive power, $f_\kappa(\cdot, \gamma)$ can be bounded similarly.

The optimistic estimator (10), however, maximises each term individually, which gives a rather loose bound. A tighter one is given by the tight optimistic estimator for f_κ^s (see Sec. 2.2): since f_κ^s computes its average coreness on G, according to Lemma 1 it is an overestimation of f_κ, i.e., $f_\kappa^s(U) \geq f_\kappa(U)$. As such, an optimistic estimator $\hat{f}^s_\kappa$ for f^s_κ is also admissible for our measure. Using this tight optimistic estimator $\hat{f}^s_\kappa$, adapted from [9], we get

$$\max_{T\subseteq U} f_\kappa(T) \leq \max_{T\in U} f^s_\kappa(T) = \max_{0<i\leq |U|} \frac{i}{|V|}\left[\frac{1}{i}\sum_{j=1}^{i}\kappa(v_j) - \bar{\kappa}\right] , \tag{11}$$

where $v_1, \ldots, v_{|V|}$ are the vertices of V ordered in decreasing core value. Once again, this bound can be adjusted for $f_\kappa(\cdot;\gamma)$.

However, both bounds (10) and (11) consider only the core values of the entire graph, which we showed in Sec. 2.2 to overestimate the coreness of the induced graph. Hence, we obtain a tighter bound than (10) by instead considering the coreness in the *induced* graph.

$$\begin{aligned}\max_{T\subseteq U} f_\kappa(T) &\leq \max_{T\subseteq U}\frac{|T|}{|V|}\max_{T\subseteq U}\left(\bar{\kappa}_T - \bar{\kappa}\right) = \frac{|T|}{|V|}\left(\bar{\kappa}_{T^*} - \bar{\kappa}\right) \\ &= \frac{|T|}{|V|}\left(K_U - \bar{\kappa}\right) \qquad\qquad \text{with } T^* = V(K_U) \ ,\end{aligned} \tag{12}$$

where K_U is the degeneracy of $G[U]$ and T^* are the core vertices of the highest k-core in $G[U]$, since they maximise $\bar{\kappa}_T$ over $T \subseteq U$.

Next, we maximise both terms, f_{c} and f_{d}, jointly on the induced subgraph. We show that the resulting estimator is tight for $\gamma = 1/2$ and generally tighter than all of the above. Importantly, it is also computable in $O(n)$. At the core of this optimistic estimator lies a tight upper bound for the total coreness $\kappa_U(U)$ of Eq. (3) over all subsets of U, written as

$$\kappa^*_U = \max_{T\subseteq U}\kappa_T(T) = \max_{1\leq i\leq |U|}\kappa^i_U \ ,$$

where we first maximise over subsets of U with a fixed cardinality i

$$\kappa^i_U = \max_{T\subseteq U\,,\ |T|=i}\kappa_T(T) \ . \tag{13}$$

To compute bound (13) we first arrange all vertices $v_1, \ldots, v_{|U|}$ of U in order of decreasing coreness, so that $\kappa_U(v_i) \geq \kappa_U(v_{i+1})$ for all $1 \leq i < |U|$. This quantity is itself upper bounded by the partial sums of the ordered core numbers:

$$\hat{\kappa}^i_U = \sum_{j=1}^{i}\kappa_U(v_j) \ .$$

We can analyse this sequence 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

$$n_k = |V_U(k)| , \qquad 0 \leq k \leq K_U + 1 \ .$$

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

$$\hat{\kappa}_U^i = \begin{cases} \sum_{\lambda=k}^{K_U} \lambda(n_\lambda - n_{\lambda+1}) & \begin{array}{c} i = n_k \\ 0 \leq k \leq K_U \end{array} \\[2ex] \dfrac{(i - n_{k+1})\hat{\kappa}_U^{n_k} + (n_k - i)\hat{\kappa}_U^{n_{k+1}}}{n_{k+1} - n_k} & \begin{array}{c} n_{k+1} \leq i < n_k \\ 0 \leq k \leq K_U \ . \end{array} \end{cases}$$

To simplify this, observe that $\hat{\kappa}_U^{n_k} = \hat{\kappa}_U^{n_{k+1}} + k(n_k - n_{k+1})$, so that

$$\hat{\kappa}_U^i = (i - n_{k+1})k + \sum_{\lambda=k}^{K_U} \lambda(n_k - n_{k+1}) \ , \qquad n_{k+1} \leq i \leq n_k \ . \tag{14}$$

This reformulation now makes it clear that the piece-wise linear sequence $\hat{\kappa}_U$ is increasing and concave (due to the monotonically decreasing increments k).

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

Proposition 1. *For the piece-wise linear function of Eq.* (14)

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 1 we can upper bound the value of f_κ^s over all subsets of U with cardinality i by the quantity

$$\hat{\phi}_U(i;\gamma) = \left(\frac{i}{|V|}\right)^{1-\gamma} \left(\frac{\hat{\kappa}_U^i}{i} - \bar{\kappa}\right)^{\gamma} \ . \tag{15}$$

Hence, the solution of Eq. (9) for $f_\kappa(U;\gamma)$ can be written as

$$\max_{T \subseteq U} f_\kappa(T;\gamma) \leq \hat{\phi}_U^*(\gamma) = \max_{0<i\leq|U|} \hat{\phi}_U(i;\gamma) \ . \tag{16}$$

Finally, we replace (15) into the above equation and then use Proposition 1 (part 2) to show the tightness of our bound (16), as follows.

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

$$\hat{\phi}_U^*(\gamma) = \max_{0<i\leq|U|} \left(\frac{i}{|V|}\right)^{1-\gamma} \left(\frac{\hat{\kappa}_U^i}{i} - \bar{\kappa}\right)^{\gamma} \ . \tag{17}$$

As a concluding remark, our proposed bound (17) can be computed in linear time: the core decomposition of a graph takes $O(n)$ time [5], after which we compute $\hat{\phi}_U^*$ as the maximum of the $|U| \leq |V| = n$ values in Eq. (17), each of which needs $O(1)$ time.

3.3 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 subgraphs within the language $\mathcal{L}$ that maximise the coreness impact function.

RoSi is an implementation of the *iterative deepening depth first search* variant of BnB [18]. In particular, it repeatedly invokes a truncated (i.e., depth-limited) depth first search (DFS) for an increasing depth limit until no search nodes are reachable below the current depth limit. 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 suboptimal—deep branches; this allows to discover shallow good solutions early.

If required, RoSi can terminate early by imposing a depth limit $d_{\max} < \infty$, which intuitively corresponds to finding the optimal selector with at most $d_{\max}$ predicates. Additionally, the optimality guarantee can be relaxed by setting an approximation factor $\alpha \in (0, 1]$, so that the discovered solution is an α-approximation of the exact optimum, where $\alpha = 1$ yields the exact solution.

Note that the complexity of the inner for-loop is $O(n)$; this includes computing the refinements, the measure, and its bound.

4 Related Work

We begin our review of related literature with methods that provide no descriptions as we progressively compare to ones closer to our own.

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. A plethora of works reinterpret density to take into account structural information, for instance, high triangle counts, measures based on large and/or dense k-cliques, quasi-cliques, k-plexes, k-clubs, and k-cores [25], just to name a few. 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. We direct the interested reader to a recent survey [12]. RoSi adapts a k-core based measure to describing its patterns.

Cohesive Subgraphs. The work of [22] 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, [16] (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, [23] (AMEN) introduce an attribute-aware variant of the established modularity measure [12] 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 [1] 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 RoSi to both PICS and AMEN, the most recent of both cases.

Worth mentioning are also works which use graph attributes to assess subgraph interestingness [6] or to detect anomalies in them [8]. These methods, however, do not provide any descriptions of the mined subgraphs, while also requiring a model for the attributes. A more recent work [2] mines descriptions for subgraphs with anomalously high edge weights (dyadic relations). Our problem does consider the exceptionality of the attributes per se, which are used instead to form descriptions.

Subgroup Discovery. Nevertheless, describing parts of the dataset which exhibit exceptional behaviour of a target concept when compared to the entire dataset defines the broad task of subgroup discovery, which also includes RoSi. Such a target concept may constitute an the exceptional distribution of a single or multiple variables, which can be applied on discrete or continuous data. More recent target concepts also require the distribution of an additional control distribution to be representative [17], or generalise to differences in models of multiple variables [11].

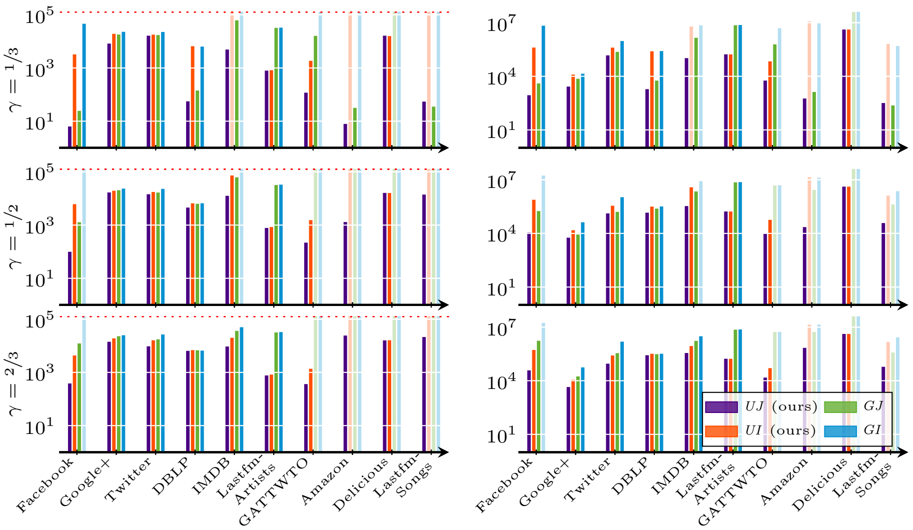
Subgroup discovery has been applied on graphs using SD-Map* [20] and variants for community detection [3], while in another line of work DCM [24] greedily optimises an introduced a community score based on differences of edge counts within, outside and across the subgroup boundary. Since our work is oriented toward dense subgraphs, perhaps the most relevant work is LDENSE [13], which adapts the greedy densest subgraph algorithm [10] to only search within describable subgroups as an approach to find overlapping communities. While LDENSE only uses edge statistics, SCPM [26] introduces a measure which samples quasi-cliques for each candidate subgroup to estimate the portion of its vertices covered by them, and can only be heuristically optimised, although with probabilistic guarantees. In contrast, we aim at a measure based on the well structured k-cores, further equipped with a tight optimistic estimate in an exact method. In our experiments we compare against both.

5 Experiments

In this section we experimentally study the properties of the RoSi algorithm. We make available our source code and all datasets for research purposes.[2] All reported experiments were run single-threaded on *Xeon E5-2643* 3.4GHz processor machines with 256GB of memory.

We consider 10 datasets that together span multiple domains and different kinds of represented entities and relations: 4 datasets from the SNAP database, 2 published datasets from the HetRec2011 workshop, the Million Song, the GATT/WTO, the *DBLP* and *IMDB* datasets. These consist of both graphs and multi-graphs, and describe various types of networks: social, similarity, co-occurrence, and collaboration networks, among others.

[2] All content accessible at https://eda.mmci.uni-saarland.de/rosi.



(a) *Wall-clock running time (s).*

(b) *Visited nodes during search.*

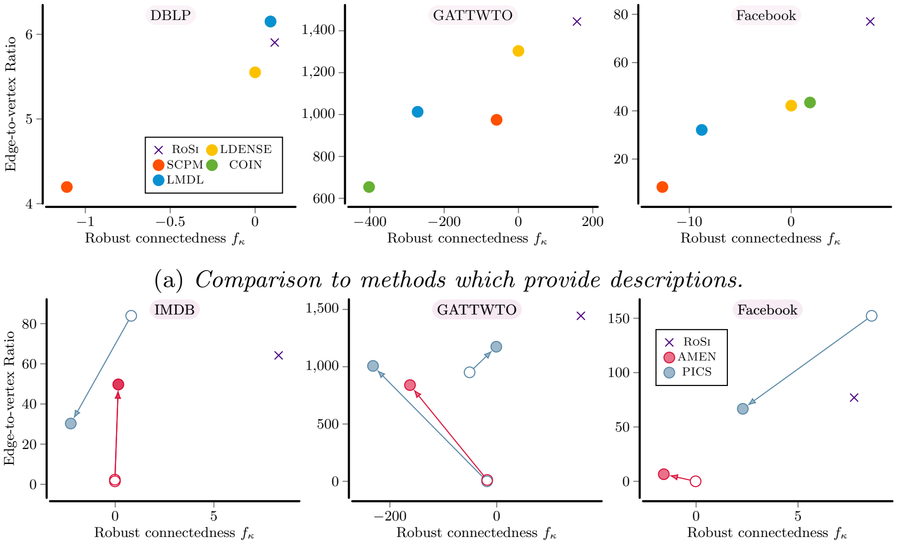
Fig. 5 [Lower is Better]: Efficiency of the optimistic estimators: higher pruning efficiency translates to less expanded nodes and thus shorter running times. Experiments exceeding a runtime of 36 hours (dotted line) are faded out.

Efficiency of RoSi: We now study how the efficiency of RoSi is affected by the pruning potential of the chosen optimistic estimator. We refer to these as ***g**lobal-**i**ndependent* (Eq. 10) (GI), ***g**lobal-**j**oint* (11) (GJ), *ind**u**ced-**i**ndependent* (12) (UI), and the tightest one as *ind**u**ced-**j**oint* (17) (UJ) where *global* indicates whether average coreness is bound using the coreness of G or $G[U]$, and *independent* indicates whether the f_c and f_d terms were maximised independently.

For the experiments we need to specify 1) the trade-off parameter γ, 2) optionally set a depth limit and 3) set the approximation factor α. For the former we use $\gamma \in \{1/3, 2/3, 1/2\}$, corresponding to representative use cases: favouring coverage, density, or balancing the importance of the two, respectively. Then, for each of these γ we run the RoSi algorithm using $\hat{f}_\mathrm{UJ}$ and perform an exact search on each of dataset (i.e., with no depth limit and approximation factor $\alpha = 1$); as long as a dataset needs more than a fixed time of 7 hours, we either lower the approximation factor α by 0.1 or lower the allowed depth by one, favouring a deeper search when possible.

For each configuration we run RoSi with every estimator for up to 36 hours and measure the wall–clock time needed for each of them; the results are listed in Fig. 5a. We note that $\hat{f}_\mathrm{UJ}$ most of the times outperforms all other estimators, whenever they do terminate, or is on par with the fastest among them.

To confirm that $\hat{f}_\mathrm{UJ}$ prunes the most, we also provide the number of expanded nodes during the search (Fig. 5b). Since for each dataset the order of predicates Π is fixed, all search nodes are expanded in the same order. Within this sequence,



(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 in terms of Jaccard similarity.*

Fig. 6 [Upper right is Better]: RoSi is not only able to discover subgroups with the highest robust connectedness, as expected; at the same time, its scores on typical density are on par with competing methods.

pruning allows to skip sub-optimal search nodes with smaller or larger jumps, when the bound is loose or tighter, respectively. Importantly, during approximate search ($\alpha \ll 1$) pruning becomes overzealous: then a bound might skip a good node, which a looser bound would "fail" to skip; this occasionally leads to an advantage for the looser bound, later on. This is more likely to occur as α lowers; in our experiments this only happens for the *Lastfm-Songs* ($\gamma = \frac{2}{3}$) when $\hat{f}_{\mathrm{GJ}}$ gains a slight advantage over $\hat{f}_{\mathrm{UJ}}$; this does not surprise given the heavy approximation rate of 50%.

The experiments corroborate that the superior pruning of $\hat{f}_{\mathrm{UJ}}$ allows to practically optimise large real-world graphs.

Optimality of RoSi: We now demonstrate that RoSi is useful for finding robustly connected subgraphs with descriptions, even with an approximate optimisation, and compare it to representative works for the described approaches in Section 4. At the same time, we experimentally study how well it performs under the typical density: the edge to vertex ratio $|U|/|E(U)|$ of subgraph $G[U]$.

We distinguish the compared methods as those who provide descriptions and those who do not. The first category includes LDENSE [13] and SCPM [26], which both search for dense subgraphs, and also two measures using subgroup discovery for community detection: that of the inverse conductance (COIN [4]) and local modularity (LMDL [3]). We provide the results for a selection of datasets for which most competing methods complete in Figure 6a. We see that RoSi scores the highest of all in terms of both measures. Although both LDENSE and SCPM search for dense subgraphs, neither is exact, with the first using a greedy approximation and the latter randomly samples quasi-cliques. The latter two unsurprisingly score higher than RoSi in terms of inverse conductance and local modularity (results omitted as irrelevant), however this is by far not the case for the two measures we are aiming at.

We next compare RoSi with PICS and AMEN, two different approaches for cohesive subgroups, which are not constrained by having to provide descriptions. For both methods we show all vertex sets in the Pareto front of the two metrics. We represent these solutions in Figure 6b with empty circles to designate that they do not correspond to a description. Although rarely, the resulting subgraphs can happen to exceed the quality of RoSi in density and/or robust connectedness, as their optimisation is less constrained. To put them in perspective, however, we also mine the closest subgroup in terms of the Jaccard distance to the one provided by each algorithm, and link to it the unconstrained solution with the arrow. As expected, these solutions score lower than those of RoSi.

Interpretable Subgraph Descriptions: To study if the discovered subgraphs are meaningful, we mine the top describable subgraph for a subset of datasets which have attributes that are easily interpretable for a lay person. We do this for a sliding trade-off parameter, once again selected from the set $\gamma \in \{0.1, 0.15, 0.2, \ldots, 0.9\}$. We list the discovered subgroups in Table 1 and give example interpretations for them below.

Table 1a describes collaborating cast members from the *IMDB* dataset. We first focus on large subgraphs, and for $0.1 \leq \gamma < 0.3$ we discover: *the drama movie cast has a robust connectedness of* 1.8 *collaborations more than what is usual in the entire industry.* If we balance size and connectedness, we find that established actors (debut before '96) not nominated by the London BFI festival have collaborated well with each other (12 collaborations more than usual). This reveals that the London BFI festival seems to select more diverse films, at least regarding established actors. When we lay more importance in connectedness, we discover that these two patterns joined together (established dramedy actors not selected by BFI) describe a very robustly connected group. What is more, additionally requiring that a movie is produced in the US is alone a substantial factor of connectedness.

We also report selected informative subgraphs discovered from another 4 datasets (Table 1b). Interesting findings include that the *Google+* social network contains a community of photographers, which have 140 other photographers as friends on average more than the dataset average; similarly, in Twitter, the followers of the American artist Hayley Williams are exceeded by 120 connections the average connection in the dataset. From the *DBLP* dataset we notice that the people publishing in the ICDM conference have a slight higher tendency to cite other people of the same field, and finally the discoveries of the *GATTWTO* dataset shows that countries which are part of the GSP trade agreement are trading with an extra 253 trade routes on average more than the dataset usual.

γ	Drama	Comedy	¬BFI	debut < '96	debut ≤ '05	US	Movies	Dens.	Cov.
[0.1 −0.3)	✓						20 579	1.8	0.87
[0.3 −0.4)			✓		✓		19 150	7.6	0.81
[0.4 −0.45)			✓	✓	✓		15 057	11.9	0.64
[0.45−0.6)	✓	✓	✓	✓	✓		11 455	17.1	0.48
[0.6 −0.9]	✓	✓	✓	✓	✓	✓	6 843	27.1	0.29

(a) *Discovered subgraphs from* IMDB.

Dataset	γ	*Description*	Size	Dens.
Google+	[0.1 −0.9]	photographer	2 835	138.9
Twitter	[0.1 −0.85)	@yellyahwil.	740	119.9
DBLP	[0.1 −0.35)	ICDM	9 022	0.1
GATTWTO	[0.25−0.55)	GSP-member	110	253.5

(b) *Discovered subgraphs of special interest.*

Table 1: Discovered subgraphs over the trade-off parameter.

6 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. Importantly, the results are meaningful and interpretable.

References

1. Akoglu, L., Tong, H., Meeder, B., Faloutsos, C.: PICS: Parameter-free identification of cohesive subgroups in large attributed graphs. In: SDM. SIAM (2012)
2. Atzmueller, M.: Compositional Subgroup Discovery on Attributed Social Interaction Networks. In: DS. pp. 259–275. Springer (2018)
3. Atzmueller, M., Doerfel, S., Mitzlaff, F.: Description-oriented community detection using exhaustive subgroup discovery. Inf. Sci. pp. 965–984 (2016)
4. Atzmueller, M., Mitzlaff, F.: Efficient Descriptive Community Mining. In: FLAIRS (2011)
5. Batagelj, V., Zaversnik, M.: An O(m) Algorithm for Cores Decomposition of Networks. arXiv:cs/0310049 (2003)
6. Bendimerad, A., Mel, A., Lijffijt, J., Plantevit, M., Robardet, C., De Bie, T.: Mining subjectively interesting attributed subgraphs. In: MLG (2018)
7. Bickle, A.: The K-Cores of a Graph. Western Michigan University (2010)

8. Bojchevski, A., Günnemann, S.: Bayesian Robust Attributed Graph Clustering: Joint Learning of Partial Anomalies and Group Structure (2018)
9. Boley, M., Goldsmith, B.R., Ghiringhelli, L.M., Vreeken, J.: Identifying Consistent Statements about Numerical Data with Dispersion-Corrected Subgroup Discovery. DAMI pp. 1391–1418 (2017)
10. Charikar, M.: Greedy Approximation Algorithms for Finding Dense Components in a Graph. In: Proc. 3rd Int. Wor. App. Alg. Comb. Opt. pp. 84–95. Springer (2000)
11. Duivesteijn, W., Feelders, A.J., Knobbe, A.: Exceptional Model Mining: Supervised descriptive local pattern mining with complex target concepts. DAMI pp. 47–98 (2016)
12. Fortunato, S., Hric, D.: Community detection in networks: A user guide. Phys. Rep. pp. 1–44 (2016)
13. Galbrun, E., Gionis, A., Tatti, N.: Top-k overlapping densest subgraphs. DAMI (2016)
14. Grosskreutz, H., Rüping, S.: On subgroup discovery in numerical domains. DAMI pp. 210–226 (2009)
15. Grosskreutz, H., Rüping, S., Wrobel, S.: Tight optimistic estimates for fast subgroup discovery. In: ECML PKDD. Springer (2008)
16. Gunnemann, S., Farber, I., Boden, B., Seidl, T.: Subspace Clustering Meets Dense Subgraph Mining: A Synthesis of Two Paradigms. In: ICDM. IEEE (2010)
17. Kalofolias, J., Boley, M., Vreeken, J.: Efficiently Discovering Locally Exceptional Yet Globally Representative Subgroups. In: ICDM. pp. 197–206. IEEE (2017)
18. Korf, R.E.: Depth-first Iterative-deepening: An Optimal Admissible Tree Search. Artif. Intell. pp. 97–109 (1985)
19. Korte, B., Vygen, J.: Combinatorial Optimization: Theory and Algorithms. Springer (2006)
20. Lemmerich, F., Atzmueller, M., Puppe, F.: Fast exhaustive subgroup discovery with numerical target concepts. Data Min. Knowl. Disc. pp. 711–762 (2016)
21. Mehlhorn, K., Sanders, P.: Algorithms and Data Structures: The Basic Toolbox. Springer (2008)
22. Moser, F., Colak, R., Rafiey, A., Ester, M.: Mining Cohesive Patterns from Graphs with Feature Vectors. In: SDM, pp. 593–604. SIAM (2009)
23. Perozzi, B., Akoglu, L.: Discovering Communities and Anomalies in Attributed Graphs: Interactive Visual Exploration and Summarization. ACM TKDD pp. 24:1–24:40 (Jan 2018)
24. Pool, S., Bonchi, F., Van Leeuwen, M.: Description-Driven Community Detection. ACM TIST pp. 28:1–28:28 (2014)
25. Shin, K., Eliassi-Rad, T., Faloutsos, C.: CoreScope: Graph Mining Using k-Core Analysis—Patterns, Anomalies and Algorithms. In: ICDM. pp. 469–478. IEEE (2016)
26. Silva, A., Meira, Jr., W., Zaki, M.J.: Mining Attribute-structure Correlated Patterns in Large Attributed Graphs. VLDB (2012)
27. Tatti, N., Gionis, A.: Density-friendly Graph Decomposition. In: WWW. pp. 1089–1099 (2015)
28. Webb, G.I.: Discovering Associations with Numeric Variables. In: KDD. pp. 383–388. ACM (2001)
29. Wrobel, S.: An algorithm for multi-relational discovery of subgroups. In: PKDD. pp. 78–87. Springer (1997)