Discovering Robustly Connected Subgraphs with Simple Descriptions

Janis Kalofolias
Helmholtz Center for Information Security
kalofolias@cispa-helmholz.de

Mario Boley
Max Planck Institute for Informatics and Monash University
mario.boley@monash.edu

Jilles Vreeken
Helmholtz Center for Inf. Security and Max Planck Inst. for Informatics
jv@cispa-helmholtz.de

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 suboptimal results. Instead, we hence 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.

CCS CONCEPTS

•**Information systems** →**Data mining;**

KEYWORDS

graph mining, k-cores, dense subgraphs, subgroup discovery

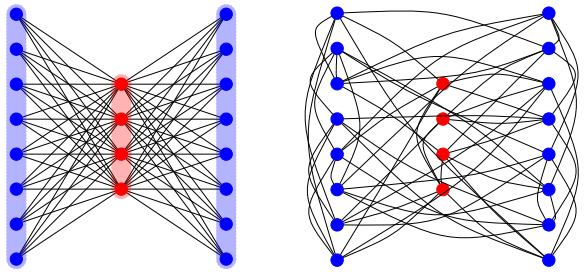
ACM Reference format:
Janis Kalofolias, Mario Boley, and Jilles Vreeken. 2019. Discovering Robustly Connected Subgraphs
with Simple Descriptions. In *Proceedings of 15th International Workshop on Mining and Learning with Graphs, Anchorage, Alaska, August 2019 (MLG'19),* 8 pages.
DOI: 10.475/123_4

1 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 has seen a lot of research attention from the data mining community. A large part of this attention 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.

MLG'19, Anchorage, Alaska

DOI: 10.475/123_4



(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.

Figure 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 fully disconnect (a) by removing just its 4 **central nodes, to achieve the same for (b) we need to remove** 19 **vertices.**

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 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 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.

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. 7.

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 $\sigma(v) \equiv [\mathsf{age}(v) \geq 18] \wedge [\mathsf{sex}(v) = \text{`}\textit{female}\text{'}]$.

Our goal is to identify such logically described sets of vertices $U \subseteq V$ that are 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

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

where γ 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. In the following we will give a precise definition of the density term based on the concept of k-cores [8].

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 $G[U]$.

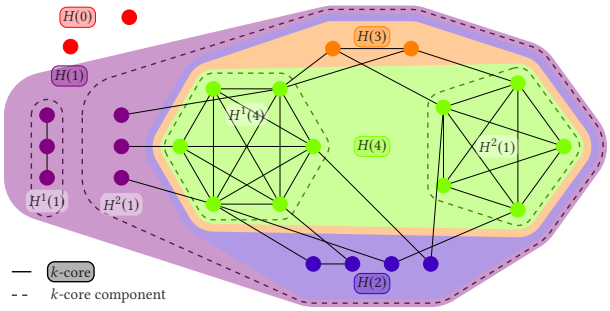


Figure 2 [Higher coreness coincides with higher density.]: The core decomposition of a graph hierarchically groups its vertices into increasingly denser subgraphs. Here $H(k)$ denotes a k-core and $H^i(k)$ the i-th k-core component.

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 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. 2 show that the k-cores are nested to form a hierarchy over the vertices. We also define the **k-shell** of G is 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**

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

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

The coreness of a subgraph is closely related to different definitions of density [29, 30]. 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 [20, 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 formalise the requirement that 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). This completes our measure for robust connectedness: the **coreness impact function**

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

Note that this measure is related yet different from the one typically used in rule mining (or subgroup discovery) for numerical unstructured data [15, 34]. 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^s_κ 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^s_κ can be optimised using existing techniques, it systematically overestimates the subgraph density, as visualised in Fig. 3. 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 2.1. *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 previously defined optimisation function f_κ we need to fix a set of potential descriptions, referred to as 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 that we derive from vertex attributes, e.g., [age > 18] or [sex = '*male*']. 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

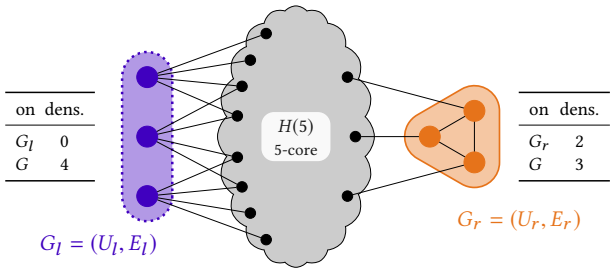


Figure 3 [Subgraph density overestimates if considers G]: The average subgraph coreness $\bar{\kappa}_U = \bar{\kappa}_U(U)$ may be misleadingly overestimated when it is computed with respect to the whole graph $\bar{\kappa}_V(U)$. Here, subgraph G_r is denser than G_l with $\bar{\kappa}_{U_r} = 2 > 0 = \bar{\kappa}_{U_l}$. If, however, we also count the edges of G, the subgraph densities will falsely indicate the opposite relation: $\bar{\kappa}_V(U_r) = 3 < 4 = \bar{\kappa}_V(U_l)$. The same holds if we use as density the edge/vertex ratio—here, their values coincide.

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, their solution can be arbitrarily far from the optimal. Below we develop an efficient algorithm that solves problem (8) exactly.

3.1 Solving Exactly with Branch–and–Bound

The established algorithm that solves problem (8) exactly is Branch–and–Bound (BnB) [23]. 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. 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.

Turning to 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** [16] 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.

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 2.1 as follows.

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

where the second inequality follows from Lemma 2.1, 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^s_κ (see Sec. 2.2): since f^s_κ computes its average coreness on G, according to Lemma 2.1 it is an overestimation of f_κ, i.e., $f^s_\kappa(U) \geq f_\kappa(U)$. 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 Boley et al. [9], we get

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

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.

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

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:

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

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 $\hat\kappa^i_U$ 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; the remaining indices are computed using linear interpolation:

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

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

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

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 $\hat\kappa^i_U$ as an upper bound for the maximum total coreness κ^i_U over all subsets of U with a fixed cardinality of i.

Proposition 3.1. *For the piece-wise linear function of Eq.* (14)
(1) $\kappa^i_U \leq \hat\kappa^i_U$, *for all* $0 \leq i \leq |U|$
(2) $\kappa^i_U = \hat\kappa^i_U$, *for* $i \in \{0, n_0, \ldots, n_{K_U}\}$

Using the first part of Proposition 3.1 we can upper bound the value of f^s_κ over all subsets of U with cardinality i by the quantity

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

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 3.1 (part 2) to show the tightness of our bound (16), as follows.

Corollary 3.2. *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* $\gamma = 1/2$.

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

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

Algorithm 1: RoSi for 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**(stack, $(s, d_{\text{cur}} + 1)$)
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

We next present **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 listed as Algorithm 1 and implements the *iterative deepening depth first search* variant of BnB [19]. In particular, it repeatedly invokes a truncated (i.e., depth-limited) depth first search (DFS) for an increasing depth limit of $d_{\text{dfs}} = 1, 2, \ldots$ until no search nodes are reachable below the current depth limit d_{dfs}. 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, which allows to discover shallow good solutions early.

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; for the rest **updateResults** (ln. 10) checks if they 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. We repeat until DFS completes untruncated, i.e., no refinement was reached below the current d_{dfs} (ln. 14,16).

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. Alternatively, the optimality guarantee can be relaxed by setting an approximation factor $\alpha \in (0, 1]$ (ln. 9), so that the discovered solution is an α-approximation of the exact optimum. Naturally, an $\alpha = 1$ yields the exact solution.

Note that the complexity of the inner for-loop (ln. 7) is $O(n)$; this includes computing the refinements, the measure, and its bound.

4 RELATED WORK

In our review of related literature we begin with methods that mine dense subgraphs; we then navigate toward describing them and we conclude with locating our method in its broader field.

Dense Subgraphs. The typical objective in dense subgraph discovery is to find the subgraph with the highest edge-to-vertex ratio. This measure has been shown to accept not only an exact max-flow based polynomial time optimisation algorithm [14], but also a greedy 2-factor approximation [10] that makes its optimisation feasible even for large graphs. Furthermore, Balalau et al. [4] and Galbrun et al. [13] give algorithms for multiple dense and possibly overlapping subgraphs, an ability also shared with our algorithm.

Many other measures of graph density exist (for a survey see [21]) that take into account more structural information, e.g., high triangle counts [31] and measures based on large and/or dense k-cliques [32], quasi-cliques[33], k-plexes, k-clubs, and k-cores [29], the latter of which is more related to our work. These constitute global scores computed on subgraphs; more recently local density scores were introduced: Tatti and Gionis [30] propose a graph decomposition similar to the k-core one, while Qin et al. [27] introduce ρ-compact graphs that arise from the lowest number of edge removals per removed vertex, related to our definition of robustness. However, none of these works describe the discovered subgraphs.

Related to dense subgraph discovery is *community detection*, where the subgraph is additionally required to be disconnected with the rest of the graph. As this both incurs a rather different optimization problem, and by far too much work has been done on this topic to be summarised here we refer the interested reader to the recent survey by Fortunato and Hric [12].

Approximate Descriptions. One step closer to our goal lies *subgraph clustering*, where vertex attributes are taken into account while clustering densely connected subgraphs. For instance, Akoglu et al. [2] use low entropy splits of the binary adjacency and attribute matrices to form vertex clusters, which are dense w.r.t. edge/vertex ratio and have similar binary features. In more recent work, AMEN [25] was applied on community detection to greedily optimise a variant of modularity that also takes into account attribute similarities of ego-nets. Both these methods, however, yield implicit descriptions. Other approaches optimise a random walk-based density measure using spectral clustering on a graph augmented to incorporate attribute information [36]. Although these clusters can be described by conjunctive descriptions, these are not guaranteed to be exact.

In another line of research, *subspace clustering* aims to group the range of attributes, subject to additional subgraph-density criteria. Moser et al. [24] find maximal connected subgraphs that contain vertices with similar attributes, and densities that surpass a given threshold. Gamer [17] discovers non-redundant sets of subgraphs, which must be connected γ-quasi-cliques for a given parameter γ.

For these methods, however, the respective density score needs only surpass a user-defined threshold, but does not contribute further to the quality of each subgraph. In the next section we describe methods which, instead, directly optimise a density metric. Loosely belonging here, P-N-RMiner [6] encodes in an information theoretic interestingness score intervals of numeric attributes of the graph used to rank candidates with this similarity–promoting score. This method, however, does not take into account any structural information and also needs to assume a specific underlying probabilistic model for the attribute distribution.

Subgroup Discovery. Our method falls in the broad category of subgroup discovery, which aims to find descriptions for parts of datasets which are statistically most interesting, i.e., they are large and have an unusual target concept when compared to the entire dataset [35]. Such a target concept may constitute an the exceptional distribution of a single or multiple variables, which can be applied on discrete [1] or continuous data [15]. More recent target concepts also require the distribution of an additional control distribution to be representative [18], or generalise to differences in models of multiple variables [11].

Subgroup discovery has been applied on graphs using an analogue of FP-Growth for community detection [3]; another line of work [26] introduces a community score based on differences of edge counts within, outside and across the subgroup boundary, optimised approximately by a greedy algorithm. 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. To the best of our knowledge, there has been no prior work applying subgroup discovery to identify dense subgraphs, let alone with a structure-aware measure.

5 EXPERIMENTS

In this section we experimentally study the properties of the RoSi algorithm, which we implemented RoSi in `Python`. We make available our source code and all datasets for research purposes.[1] All reported experiments were run single-threaded on *Xeon E5-2643* 3.4*GHz* processor machines with 256*GB* of memory.

5.1 Datasets

We consider 10 datasets that together span multiple domains and different kinds of represented entities and relations (see Table 1): 4 datasets from the SNAP database [22], 2 published datasets from the HetRec2011[2] workshop, the Million Song [7], the GATT/WTO [28], 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.

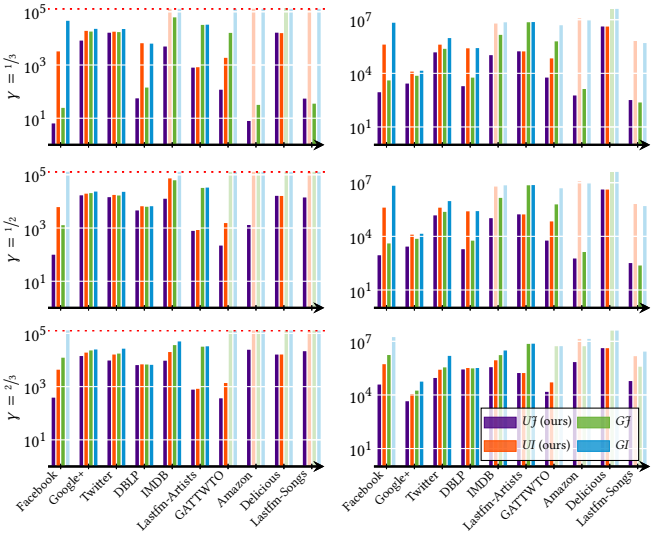
5.2 Efficiency of RoSi

We now study how the efficiency of RoSi is affected by the pruning potential of the chosen optimistic estimator. We refer to those introduced in Sec. 3.2 as ***global-independent*** (10) (GI), ***global-joint*** (11) (GJ), ***induced-independent*** (12) (UI), and the tightest one as ***induced-joint*** (17) (UJ). Notice that the global–induced distinction indicates whether the average coreness is bound using the coreness of G or $G[U]$, while the independent–joint classification indicates whether the f_c and f_d terms were maximised independently or not.

Name	Nodes		Edges		Attrs.	α	$\bar{\kappa}$
	No.	Kind	No.	Kind			
`Facebook`*	4037	user	170174	friendship	20	1	52.1
`Google+`*	78393	user	28312689	friends	10	0.1	366.7
`Delicious`†	1867	user	15328	contact	50	0.3	11.0
`Lastfm-Artists`†	1892	user	25434	artist	15	1	14.6
`Twitter`*	51246	user	1735925	follower	14	1	35.7
`DBLP`	17488	author	97070	co-auth.	113	0.3	8.5
`IMDB`	23700	crew	1134676	collab.	55	0.8	50.9
`GATTWTO`	177	country	230777	trading	27	1	1606.7‡
`Amazon`*	16641	record	162815	purchase	145	0.7	13.9
`Lastfm-Songs`§	251272	song	1179317	similarity	50	0.5	5.2

Sources: *SNAP repository †HETREC Workshop §Million Song Dataset
‡Multi-graphs may have degeneracy $K \geq |V|$.

Table 1: Overview of dataset statistics.



(a) *Wall-clock running time (s).* (b) *Traversed nodes during search.*

Figure 4 [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.

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 $\hat{f}_{\text{UJ}}$ 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

[1] All content accessible at https://www.dropbox.com/s/duyfcsy0nbjoy8p/RoSi.zip.
[2] 2nd International Workshop on Information Heterogeneity and Fusion in Recommender Systems http://recsys.acm.org/2011.

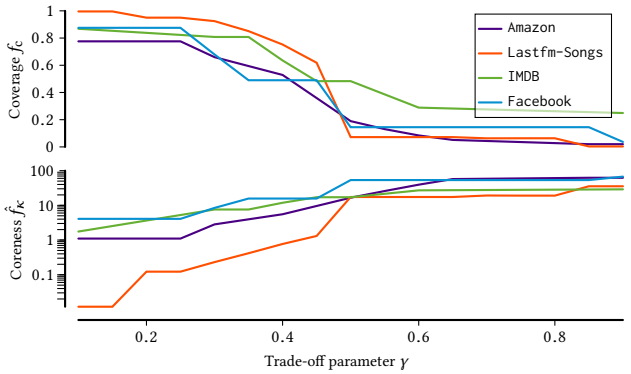


Figure 5 [Coreness vs. Coverage]: Increasing the coreness–coverage trade-off parameter γ yields smaller but more robustly connected subgraphs.

them; the results are listed in Fig. 4a. We observe that $\hat{f}_{\text{UJ}}$ most of the times outperforms all other estimators, some of which do not even terminate, or is on par with the fastest among them.

To confirm that $\hat{f}_{\text{UJ}}$ prunes the most, we also provide the number of expanded nodes during the search (Fig. 4b). Since we fix the order of predicates Π (per dataset), during exact search we always guarantee the traversal of fewer search nodes. In approximate search ($\alpha < 1$), however, we employ over-pruning. Then it becomes possible that a tighter bound overzealously prunes some good branch that a looser 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` ($\gamma = 2/3$) when $\hat{f}_{\text{GJ}}$ gains a slight advantage over $\hat{f}_{\text{UJ}}$, for which we used an approximation rate of 50%.

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

5.3 The Generality–Connectedness Trade-Off

Next, we focus on 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 single out 4 datasets with highly diverse base predicates, which allows the greatest flexibility in the resulting descriptions. For each of those we sweep the trade-off parameter between the range $0.1 \leq \gamma \leq 0.9$ in regular increments of 0.05 and for each value we run RoSi using $\hat{f}_{\text{UJ}}$ until completion. We then plot the coverage and connectedness of the topmost result (Fig. 5).

We observe that continuously increasing parameter γ leads to smaller and more densely connected subgraphs. In other words, the trade-off parameter γ intuitively steers the results toward more general or more connected subgraphs.

5.4 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 2 and give example interpretations for them below.

Table 2a 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.

Similarly, Table 2b lists discovered subgroups from the Lastfm song similarity dataset. These reveal that the few live recordings are dissimilar, most likely due to the higher noise levels involved in live venues. We also find out certain genres to offer greater variation in their songs, e.g., metal, indie, experimental, punk, and alternative rock, exemplary genres known for novel sounds and breaking norms. Lastly, we identify genres with many more similar pieces within them than the average, for instance the 18 thousand oldies and the thousand dance-party 70's songs.

We also report selected informative subgraphs discovered from another 4 datasets (Table 2c). 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.

6 DISCUSSION

The experiments showed that joint optimization is necessary, that RoSi is feasible even on large graphs, and that its results are meaningful and easily interpretable. Nevertheless, it comes with natural limitations, which we discuss below.

By design, the subgroups we discover are robustly connected—but not necessarily connected in the usual sense. In our definition of robust connectedness, we consider the average k-coreness of vertices, which does not require that the subgraph has to be a single component. Rather, as long as k is sufficiently high, we favourably score subgraphs that consists of multiple k-core components, *even* if these are pairwise disconnected. Should connected subgraphs be required, this can always be enforced as a post-processing step.

A natural downside to solving a combinatorial problem is that it is hard to apriori estimate the runtime of the search.Our framework, however, allows us to use the current optimality gap as a progress indicator. Moreover, as RoSi continuously refines its solution up till convergence, it constitutes an any-time algorithm.

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

(a) *Discovered subgraphs from dataset* IMDB.

γ	*Description*	Songs	Dens.	Cov.
[0.1 −0.2)	¬live	250 168	0.01	0.996
[0.2 −0.3)	¬exper.	238 682	0.12	0.950
[0.3 −0.35)	¬metal	232 272	0.23	0.924
[0.35−0.4)	¬metal ∧ ¬indie rock	213 803	0.42	0.851
[0.4 −0.45)	¬exper. ∧¬metal∧¬indie	189 054	0.77	0.752
[0.45−0.5)	¬ambient ∧ ¬alt rock ∧ ¬metal ∧ ¬punk ∧ ...* †‡§	155 446	1.31	0.619
[0.5 −0.7)	oldies	18 089	17.39	0.072
[0.7 −0.75)	¬90s ∧ oldies ∧ ...* †	15 842	19.21	0.063
[0.75−0.85)	¬00s ∧ oldies ∧ ...† ‡	19.06	0.063	
[0.85−0.9]	¬live∧party∧70s∧dance	862	35.45	0.003

(b) *Discovered subgraphs from dataset* Lastfm-Artists.
*¬live ∧ ¬exper. †¬hard rock ‡¬indie §¬indie rock

Dataset	γ	*Description*	Nodes	Dens.	Cov.
Google+	[0.1 −0.9]	photographer	2 835	138.89	0.036
Twitter	[0.1 −0.85)	@yellyahwil.	740	119.93	0.014
DBLP	[0.1 −0.35)	ICDM	9 022	0.09	0.516
GATTWTO	[0.25−0.55)	GSP-member	110	253.47	0.621

(c) *Individual discovered subgraphs of special interest.*

Table 2: Discovered subgraphs over the trade-off parameter.

7 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. 2009. Evaluation Measures for Multi-Class Subgroup Discovery. In *ECML PKDD*. Springer, 35–50.
[2] L. Akoglu, H. Tong, B. Meeder, and C. Faloutsos. 2012. PICS: Parameter-Free Identification of Cohesive Subgroups in Large Attributed Graphs. In *SDM*. SIAM.
[3] M. Atzmueller. 2018. Compositional Subgroup Discovery on Attributed Social Interaction Networks. In *DS*. Springer, 259–275.
[4] O. D. Balalau, F. Bonchi, T.-H. H. Chan, F. Gullo, and M. Sozio. 2015. Finding Subgraphs with Maximum Total Density and Limited Overlap. In *WSDM*. ACM.
[5] V. Batagelj and M. Zaversnik. 2003. An O(m) Algorithm for Cores Decomposition of Networks. *arXiv:cs/0310049* (2003).
[6] A. Bendimerad, A. Mel, J. Lijffijt, M. Plantevit, C. Robardet, and T. De Bie. 2018. Mining Subjectively Interesting Attributed Subgraphs. In *MLG*.
[7] T. Bertin-Mahieux, D. P. Ellis, B. Whitman, and P. Lamere. 2011. The Million Song Dataset. In *ISMIR*.
[8] A. Bickle. 2010. *The K-Cores of a Graph*. Western Michigan University.
[9] M. Boley, B. R. Goldsmith, L. M. Ghiringhelli, and J. Vreeken. 2017. Identifying Consistent Statements about Numerical Data with Dispersion-Corrected Subgroup Discovery. *DAMI* (2017), 1391–1418.
[10] M. Charikar. 2000. Greedy Approximation Algorithms for Finding Dense Components in a Graph. In *Proc. 3rd Int. Wor. App. Alg. Comb. Opt.* Springer, 84–95.
[11] W. Duivesteijn, A. J. Feelders, and A. Knobbe. 2016. Exceptional Model Mining: Supervised Descriptive Local Pattern Mining with Complex Target Concepts. *DAMI* (2016), 47–98.
[12] S. Fortunato and D. Hric. 2016. Community Detection in Networks: A User Guide. *Phys. Rep.* (2016), 1–44.
[13] E. Galbrun, A. Gionis, and N. Tatti. 2016. Top-k Overlapping Densest Subgraphs. *DAMI* (2016).
[14] A. V. Goldberg. 1984. *Finding a Maximum Density Subgraph*. Technical Report. University of California at Berkeley.
[15] H. Grosskreutz and S. Rüping. 2009. On Subgroup Discovery in Numerical Domains. *DAMI* (2009), 210–226.
[16] H. Grosskreutz, S. Rüping, and S. Wrobel. 2008. Tight Optimistic Estimates for Fast Subgroup Discovery. In *ECML PKDD*. Springer, 440–456.
[17] S. Gunnemann, I. Farber, B. Boden, and T. Seidl. 2010. Subspace Clustering Meets Dense Subgraph Mining: A Synthesis of Two Paradigms. In *ICDM*. IEEE.
[18] J. Kalofolias, M. Boley, and J. Vreeken. 2017. Efficiently Discovering Locally Exceptional Yet Globally Representative Subgroups. In *ICDM*. IEEE, 197–206.
[19] R. E. Korf. 1985. Depth-First Iterative-Deepening: An Optimal Admissible Tree Search. *Artif. Intell.* (1985), 97–109.
[20] B. Korte and J. Vygen. 2006. *Combinatorial Optimization: Theory and Algorithms*. Springer.
[21] V. E. Lee, N. Ruan, R. Jin, and C. Aggarwal. 2010. A Survey of Algorithms for Dense Subgraph Discovery. In *Managing and Mining Graph Data*. Springer.
[22] J. Leskovec and A. Krevl. 2014. *SNAP Datasets: Stanford Large Network Dataset Collection*.
[23] K. Mehlhorn and P. Sanders. 2008. *Algorithms and Data Structures: The Basic Toolbox*. Springer.
[24] F. Moser, R. Colak, A. Rafiey, and M. Ester. 2009. Mining Cohesive Patterns from Graphs with Feature Vectors. In *SDM*. SIAM, 593–604.
[25] B. Perozzi and L. Akoglu. 2018. Discovering Communities and Anomalies in Attributed Graphs: Interactive Visual Exploration and Summarization. *ACM TKDD*, Article 24 (Jan. 2018), 24:1-24:40 pages.
[26] S. Pool, F. Bonchi, and M. Van Leeuwen. 2014. Description-Driven Community Detection. *ACM TIST* (2014), 28:1–28:28.
[27] L. Qin, R.-H. Li, L. Chang, and C. Zhang. 2015. Locally Densest Subgraph Discovery. In *KDD*. ACM, 965–974.
[28] A. K. Rose. 2002. *Do We Really Know That the WTO Increases Trade?* Technical Report. Nat. Bureau of Econ. Research.
[29] K. Shin, T. Eliassi-Rad, and C. Faloutsos. 2016. CoreScope: Graph Mining Using k-Core Analysis—Patterns, Anomalies and Algorithms. In *ICDM*. IEEE, 469–478.
[30] N. Tatti and A. Gionis. 2015. Density-Friendly Graph Decomposition. In *WWW*. 1089–1099.
[31] C. Tsourakakis. 2014. A Novel Approach to Finding Near-Cliques: The Triangle-Densest Subgraph Problem. *arXiv e-prints* (May 2014), arXiv:1405.1477.
[32] C. Tsourakakis. 2015. The K-Clique Densest Subgraph Problem. In *WWW*. 1122–1132.
[33] C. Tsourakakis, F. Bonchi, A. Gionis, F. Gullo, and M. A. Tsiarli. 2013. Denser than the Densest Subgraph: Extracting Optimal Quasi-Cliques with Quality Guarantees. In *KDD*. ACM, 104–112.
[34] G. I. Webb. 2001. Discovering Associations with Numeric Variables. In *KDD*. ACM, 383–388.
[35] S. Wrobel. 1997. An algorithm for multi-relational discovery of subgroups. In *PKDD*. Springer, 78–87.
[36] Y. Zhou, H. Cheng, and J. X. Yu. 2010. Clustering Large Attributed Graphs: An Efficient Incremental Approach. In *ICDM*. IEEE, 689–698.