Description-Oriented Community Detection using Exhaustive Subgroup Discovery

Martin Atzmueller[a,*], Stephan Doerfel[a], Folke Mitzlaff[a]

[a] *University of Kassel, Research Center for Information System Design, Knowledge and Data Engineering Group, Wilhelmshöher Allee 73, 34121 Kassel, Germany*

Abstract

Communities can intuitively be defined as subsets of nodes of a graph with a dense structure in the corresponding subgraph. However, for mining such communities usually only structural aspects are taken into account. Typically, no concise nor easily interpretable community description is provided.

For tackling this issue, this paper focuses on description-oriented community detection using subgroup discovery. In order to provide both structurally valid and interpretable communities we utilize the graph structure as well as additional descriptive features of the graph's nodes. A descriptive *community pattern* built upon these features then describes and identifies a community, i. e., a set of nodes, and vice versa. Essentially, we mine patterns in the "description space" characterizing interesting sets of nodes (i. e., subgroups) in the "graph space"; the interestingness of a community is evaluated by a selectable quality measure.

We aim at identifying communities according to standard community quality measures, while providing characteristic descriptions of these communities at the same time. For this task, we propose several optimistic estimates of standard community quality functions to be used for efficient pruning of the search space in an exhaustive branch-and-bound algorithm. We demonstrate our approach in an evaluation using five real-world data sets, obtained from three different social media applications.

Keywords: social network analysis, community detection, subgroup discovery, optimistic estimates, exceptional model mining, exploratory data analysis

*Corresponding author

Email addresses: atzmueller@cs.uni-kassel.de (Martin Atzmueller), doerfel@cs.uni-kassel.de (Stephan Doerfel), mitzlaff@cs.uni-kassel.de (Folke Mitzlaff)

Preprint of: Martin Atzmueller, Stephan Doerfel, Folke Mitzlaff. Description-Oriented Community Detection using Exhaustive Subgroup Discovery. Information Sciences (2015) http://dx.doi.org/10.1016/j.ins.2015.05.008

1. Introduction

While classic community detection, e. g., [17] for a survey, just identifies subgroups of nodes with a dense structure, lacking an interpretable description, this paper focuses on the task of *description-oriented comunity detection*. Using additional descriptive features of the nodes contained in the network, we approach the task of identifying communities as sets of nodes together with a *description*, i. e., a logical formula on the values of the nodes' descriptive features. Such a *community pattern* then provides an intuitive description of the community, e. g., by an easily interpretable conjunction of attribute-value pairs. This is usually not achieved by classical community mining methods that consider the nodes of a network (e. g., denoting users in a social network) as mere strings or ids.

We present an algorithm for description-oriented community detection of the top-k communities (described by community patterns) with respect to a number of standard community evaluation functions. The method is based on an adapted subgroup discovery approach [10, 36], and also tackles typical problems that are not addressed by standard approaches for community detection such as pathological cases like small community sizes. We focus on interpretable patterns that can easily be incorporated into a practical application, for example, for recommendations in social bookmarking systems. It is important to note that we focus on static social graphs and do not take the dynamics into account since we aim to characterize a given community (allocation) for a given fixed interaction structure. Also, since in practice the entities in a network tend to belong to a number of different communities, the presented method naturally captures overlapping community allocations. Moreover, in contrast to global approaches, we focus on the discovery of local communities. According to the idea of local pattern mining, e. g., [20], we do not try to find a complete (global) partitioning of the network. Instead, we consider a set of local, potentially overlapping communities. These should be as exceptional as possible with respect to a given community quality measure.

We demonstrate our approach on several social media applications such as social networking and social bookmarking systems that provide interaction networks like explicit friendship relations between users. However, the presented approach is not limited to such systems and can be applied to any kind of graph-structured data for which additional descriptive features (node labels) are available, e. g., certain activity in telephone networks or interactions in face-to-face contacts [6] that also utilize tags or topic descriptions for the contained relations.

As an accompanying example, throughout the paper we use the friendship graph of the social bookmarking system BibSonomy[1] [15]. In BibSonomy, users can declare their friendship toward other users, thus, creating a directed graph with users as nodes. At the same time, each user collects and tags resources like publications and web pages. Thus, a user's set of tags can be considered as a description of that user's interests. The community mining task here is to find user groups, where users are well connected by their friendship links and share a common interest in one or more features (tags).
Overall, the contribution of this paper can be summarized as follows:

1. We first introduce description-oriented community detection and present the COMODO algorithm for obtaining the k-best community patterns using a given community evaluation measure. COMODO is a branch-and-bound algorithm based on an exhaustive subgroup discovery approach.

2. For fast description-oriented community detection using COMODO, we propose optimistic estimates [25, 62] which are efficient to compute. We consider a number of standard community quality functions: The *segregation index* [19], the *inverse average ODF (out degree fraction)* [38], and the *modularity* [49]. We discuss the different measures for unweighted and weighted graphs, and extend the optimistic estimates accordingly.

3. We evaluate the presented approach using five data sets from three real-world social applications, i. e., from the social bookmarking systems BibSonomy and delicious[2], and from the social media platform last.fm[3].

The remainder of the paper is structured as follows: Section 2 summarizes basics of subgroup discovery, and provides general notions of graphs and community mining measures. Next, Section 3 introduces the proposed approach for description-oriented community detection and presents a number of optimistic estimates for standard community evaluation functions. After that, Section 4 discusses related work. For demonstrating the effectiveness and validity of the presented approach, Section 5 provides experiments using five data sets and discusses their results in the context of the three real-world applications. Finally, Section 6 concludes the paper with a summary and directions for future research.

[1] `http://www.bibsonomy.org`
[2] `http://www.delicious.com`
[3] `http://last.fm`

2. Preliminaries

In the following, we briefly introduce basic notions with respect to pattern mining using subgroup discovery, graphs, and community quality measures.

2.1. Pattern Mining using Subgroup Discovery

Subgroup discovery [28, 62, 13, 5] aims at identifying interesting patterns with respect to a given target property of interest and according to a specific quality (interestingness) measure. The top patterns are then ranked according to the selected quality measure.

Formally, a database $D = (I,A)$ is given by a set of individuals I and a set of attributes A. A *selector* or *basic pattern* $sel_{a=a_j}$ is a boolean function $I \rightarrow \{0,1\}$ that is true, iff the value of attribute a is equal to a_j for the respective individual. For a numeric attribute a_{num} selectors $sel_{a\in[min_j;max_j]}$ can be defined analogously for each interval $[min_j;max_j]$ in the domain of a_{num}. In this case, the corresponding boolean function is set to true, iff the value of attribute a_{num} is within the respective range. The set of all basic patterns is denoted by S.

A *subgroup description* or (complex) *pattern* $p = \{sel_1,\ldots,sel_d\}$ is then given by a set of basic patterns, which is interpreted as a conjunction, i.e., $p(I) = sel_1 \wedge \ldots \wedge sel_l$, with $|p| = l$. In the context of this paper, we focus on a conjunctive pattern language using nominal attribute–value pairs as defined above in this paper; for this description language, internal conjunctions can also be generated by appropriate attribute–value construction methods, if necessary. We call a pattern p a *superpattern* (or *refinement*) of a *subpattern* p_s, iff $p_s \subset p$. A *subgroup (extension)* $sg_p := ext(p) := \{i \in I | p(i) = true\}$ is the set of all individuals which are covered by the subgroup description p.

Example. In the following we will use the social bookmarking system BibSonomy as an example to illustrate the defined notions. BibSonomy allows its users to collect, tag, and share publication metadata as well as web bookmarks. Users can store a resource (a publication or a web link) and add several tags (arbitrarily chosen keywords) to it. Such a tag can then be used to retrieve those resources that the tag has been assigned to – within one's own collection as well as from the collections of others.

In our BibSonomy example, the individuals are the users and the set of attributes is the set of all tags. Each tag corresponds to one (binary) attribute with the values true and false. The corresponding selector yields true for a user iff the user has used the tag at least once. A subgroup description is therefore given by

any set of tags and the according extension is the set of all users that have used each tag in the description at least once. These users form a community, i. e., the community of users who share an interest in the notions described by the set of tags contained in the description.

For subgroup discovery the search space 2^S is the set of all possible patterns as combinations of the basic patterns in S. A quality function $Q\colon 2^S \to \mathbb{R}$ maps every pattern in the search space to a real number that reflects the interestingness of a pattern (or the pattern's extension, respectively). There exist a variety of possible quality functions, e. g., [28, 62, 2]. Simple examples consider shares of binary attributes in a subgroup. More complex variants include, e. g., the mapping of a subgroup as a set of nodes to the quality computed on a graph structure, as in our case of description-oriented community detection, similar to complex target concepts in the exceptional model mining framework, cf. [35, 5]. The result of a subgroup discovery task is the set of k subgroup descriptions $res_1, \ldots, res_k$ with the highest interestingness according to the applied quality function.

For many quality functions an *optimistic estimate* of a pattern p can be derived. This estimate describes an upper bound for the quality that any refinement of p can have. If the optimistic estimate of a pattern is below the quality of the worst of the k best patterns obtained so far, then the current branch of the search tree along the refinement path can safely be pruned. More formally, an optimistic estimate $\mathrm{oe}(q)$ of a quality function q is a function such that $p \subseteq p' \to \mathrm{oe}(q(p)) \geq q(p')$, i. e., such that no refinement p' of the pattern p can exceed the quality obtained by the optimistic estimate $\mathrm{oe}(q(p))$.

2.2. Graphs

An *undirected graph* $G = (V, E)$ is an ordered pair, consisting of a finite set V containing the *vertices/nodes*, and a set E of *edges*, which contains two-element subsets of V. A *directed graph* is defined accordingly: Here, E denotes a subset of $V \times V$. In the following, we freely use the term *network* as a synonym for graph. A *weighted*, directed or undirected graph is a graph $G = (V, E)$ together with a function $\mathrm{w} : E \to \mathbb{R}^+$ assigning a positive weight to each edge.

The *degree* $\mathrm{d}(u)$ of a node u in a network measures the number of connections it has to other nodes. In a directed graph the *outgoing degree* $\mathrm{d}^{\mathrm{out}}(u)$ counts the edges starting at u and the *incoming degree* $\mathrm{d}^{\mathrm{in}}(u)$ the ones ending in u. Similarly, in weighted graphs the *strength* $\mathrm{s}(u)$ is the sum of the weights of all edges containing u, i. e.,

$$\mathrm{s}(u) := \sum_{\{u,v\} \in E} \mathrm{w}(\{u, v\}).$$

Instrength $\mathrm{s}^{\mathrm{in}}(u)$ and *outstrength* $\mathrm{s}^{\mathrm{out}}(u)$ are defined accordingly. The *adjacency matrix* of a graph is a matrix $A \in \mathbb{R}^{|V|\times|V|}$ such that $A_{u,v} = 1$ iff $\{u,v\} \in E$ or $(u,v) \in E$, respectively, for nodes $u, v \in V$.

For weighted graphs the adjacency matrix contains the edge weights whenever the according edge is present in the graph. We identify a graph with its according adjacency matrix where appropriate.

2.3. Community Quality Measures

The concept of a *community* intuitively describes a group C of individuals out of a population such that members of C are strongly "related" among each other but sparsely "related" to individuals outside of C. This notion translates to communities as vertex sets $C \subseteq V$ of a graph $G = (V,E)$. To determine the amount of relatedness (and, thus, the community quality of such a subset) several measures have been proposed.

For a given undirected graph $G = (V,E)$ and a community $C \subseteq V$ we use the following notation: $n := |V|$, $m := |E|$, $n_C := |C|$, $m_C := |\{\{u,v\} \in E : u,v \in C\}|$ – the number of *intra-edges* of C, and $\bar{m}_C := |\{\{u,v\} \in E : |\{u,v\} \cap C| = 1\}|$ – the number of *inter-edges* of C. Furthermore, it is convenient to introduce an *inter-degree* for a node $u \in C$ (that depends on the choice of C) by $\bar{\mathrm{d}}_C(u) := |\{\{u,v\} \in E : v \notin C\}|$, counting the number of edges between u and nodes outside of C.

A simple but useful observation is the following equation that combines some the above defined entities for a community C:

$$\sum_{i \in C} \mathrm{d}(i) = 2m_C + \bar{m}_C \,. \tag{1}$$

Different evaluation functions $2^V \to \mathbb{R}$ for measuring the community quality exist (according to slightly different intuitions of what a good community is). In the context of this paper, we focus on *maximizing* local quality functions for single communities. We hence consider the inverse of a quality measure in those cases, where the measure itself indicates higher quality by lower values.

The *Inverse Average-ODF (out-degree fraction)* IAODF [66] compares the number of *inter-edges* to the number of all edges of a community C, and averages this for the whole community by considering the fraction for each individual node:

$$\mathrm{IAODF}(C) := 1 - \frac{1}{n_C} \sum_{u \in C} \frac{\bar{\mathrm{d}}_C(u)}{\mathrm{d}(u)} \tag{2}$$

The *segregation index* SIDX [19] compares the number of expected inter-edges to the number of observed inter-edges, normalized by the expectation:

$$\mathrm{SIDX}(C) = \frac{E(\bar{m}_C) - \bar{m}_C}{E(\bar{m}_C)} = 1 - \frac{\bar{m}_C n(n-1)}{2mn_C(n-n_C)} \tag{3}$$

Finally, the *modularity* MOD [50, 49, 51] of a graph clustering with k communities $C_1, \ldots, C_k \subseteq V$ focuses on the number of edges *within* a community and compares that with the *expected* such number given a null-model (i.e., a corresponding random graph where the node degrees of G are preserved). It is given by

$$\mathrm{MOD} = \frac{1}{2m} \sum_{u,v \in V} \left(A_{u,v} - \frac{\mathrm{d}(u)\,\mathrm{d}(v)}{2m} \right) \delta(C(u), C(v)), \tag{4}$$

where $C(i)$ denotes for $i \in V$ the community to which node i belongs. $\delta(C(u), C(v))$ is the *Kronecker delta* symbol that equals 1 if $C(u) = C(v)$, and 0 otherwise.

The *modularity contribution* of a single community C in a *local context* (subgraph) can then be computed [51, 52] as:

$$\mathrm{MODL}(C) = \frac{1}{2m} \sum_{u,v \in C} \left(A_{u,v} - \frac{\mathrm{d}(u)\,\mathrm{d}(v)}{2m} \right),$$

yielding

$$\mathrm{MODL}(C) = \frac{2m_C}{2m} - \sum_{u,v \in C} \frac{\mathrm{d}(u)\,\mathrm{d}(v)}{4m^2} = \frac{m_C}{m} - \sum_{u,v \in C} \frac{\mathrm{d}(u)\,\mathrm{d}(v)}{4m^2}.$$

All the presented measures can be used on directed graphs, e. g., simply by ignoring directions (and loops).

However, for the modularity in [34] an adaptation is designed that works directly on directed networks:

$$\mathrm{dMOD} = \frac{1}{m} \sum_{u,v \in V} \left(A_{u,v} - \frac{\mathrm{d}^{\mathrm{in}}(u)\,\mathrm{d}^{\mathrm{out}}(v)}{m} \right) \delta(C(u), C(v)), \tag{5}$$

providing a directed version for the contribution of a single community:

$$\mathrm{dMODL}(C) = \frac{1}{m} \sum_{u,v \in C} \left(A_{u,v} - \frac{\mathrm{d}^{\mathrm{in}}(u)\,\mathrm{d}^{\mathrm{out}}(v)}{m} \right) = \frac{m_C}{m} - \sum_{u,v \in C} \frac{\mathrm{d}^{\mathrm{in}}(u)\,\mathrm{d}^{\mathrm{out}}(v)}{m^2} \tag{6}$$

For weighted graphs, all considered measures can be adapted by accumulating the edges' weights instead of the edges. While the degree of a node is replaced by the node's strength, m, m_C and $\bar{m}_C$ have to be rewritten as follows:

$$m := \sum_{\{u,v\} \in E} \mathrm{w}(\{u,v\}),\ m_C := \sum_{\substack{\{u,v\} \in E,\\ u,v \in C}} \mathrm{w}(\{u,v\}),\ \text{and}\ \bar{m}_C := \sum_{\substack{\{u,v\} \in E,\\ |\{u,v\} \cap C| = 1}} \mathrm{w}(\{u,v\}).$$

It is important to note that the community measures can be computed only using the number of edges contained in the community m_C and other graph parameters such as the total numbers of edges m and nodes n, and the respective node degrees $\mathrm{d}(i)$. The only exception is the inverse Average-ODF that also depends on the inter-degrees $\bar{\mathrm{d}}_C(u)$, which will have to be determined for each community candidate. This allows to compile the data into an efficient data structure as described below. For directed graphs, the values $\mathrm{d}^{\mathrm{in}}(u)$ and $\mathrm{d}^{\mathrm{out}}(v)$ will have to be stored for each node of an edge.

3. Description-Oriented Community Detection

Many community mining algorithms collect sets of nodes denoting the individual communities focusing on structural aspects of the graph; typically there is no simple, and easily interpretable description. In our example, a user community would be represented merely as a set of names (strings) or ids. To bridge this gap, we combine community detection and subgroup discovery in a unified approach for mining community patterns. This tackles one of the basic problem of commmunity detection in many applications, cf. [17]: *How to identify and to concisely describe a community of users at the same time?* The proposed approach for description-oriented community detection aims at detecting the top-k patterns (in the description space) according to a given community quality function. In this way, local communities (similar to [32, 48]) are collected. However, they are not assessed in a global context of a graph partitioning. Instead, we focus on 'nuggets in the data' [28], i.e., on exceptional patterns according to the principles of local pattern mining. Accordingly, our approach also tackles typical problems that are not addressed by other approaches/measures, e.g., pathological cases such as small community sizes. We focus on interpretable patterns that can easily be deployed in a practical application.

In the following, we first provide an overview on the presented approach. Then, we discuss the COMODO algorithm for fast description-oriented community detection. After that, we introduce optimistic estimates for standard community evaluation functions for mining local community patterns efficiently.

3.1. Overview

Intuitively, community detection is concerned with the identification of subgroups [61], in which the elements are more densely linked among each other, than to other groups. Hence, subgroups and communities are rather similar, and we will use the terms interchangeably. Our goal is to discover the k best communities in a graph G that can be described by the attributes of their nodes and that maximize a given community evaluation function.

For the description of the communities, we require a database D containing a record for each graph node. Since communities can intuitively be regarded as subgraphs that are densely connected, we consider only node sets without isolated nodes as candidates for communities.

Example. In our BibSonomy example (see Section 2.1) the nodes in the graph G are individual users. Since in BibSonomy users interact with each other in several ways – e. g., by visiting each others profiles or by explicitly declaring other users as friends –, there are several options to define the edges. A visit graph, for example, would contain an edge from user a to a user b if a had visited the profile of b. Similarly a friend graph would contain an edge from a to b if a had declared their friendship to b. Independent from the chosen graph, the database D contains for each user a record with all their tags.

Formally, we discuss the following optimization problem: Given is an undirected graph $G = (V,E)$, a (community-)quality function $q : 2^V \to \mathbb{R}$ and a set of attributes A with functions $V \to dom(a_i) : v \mapsto a_i(v)$ assigning to each graph node the basic pattern $sel_{a_i=a_i(v)}$ for each attribute $a_i \in A$, determined by its attribute value $a_i(v)$ from the value domain $dom(a_i)$ of the respective attribute. To determine are the k best solutions of:

$$q(\overline{ext}(p)) \to \max! \tag{7}$$

where the solution space contains all possible descriptions, i.e., complex patterns of the form $p = \{sel_1, \ldots, sel_l\}$, interpreted as a conjunction $p(I) = sel_1 \wedge \ldots \wedge sel_l$, with $length(p) = l$. Hereby,

$$\overline{ext}(p) := \{u \in ext(p) : (\exists v \in ext(p) : \{u,v\} \in E)\}$$

is the community described by p, i. e., the extension of the pattern p without nodes that do not have at least one edge within that subgraph. For directed graphs our targeted communities are given by

$$\overline{ext}(p) := \{u \in ext(p) : (\exists v \in ext(p) : (u,v) \in E \text{ or } (v,u) \in E)\}.$$

It is often reasonable and natural – especially in large real-life networks – to require a minimal size for each community. Therefore, we introduce $\tau_n \in \mathbb{N}$ as a minimum support threshold for $\overline{ext}(p)$. This is without loss of generality, since $\tau_n = 2$ captures the extreme case of a community consisting of only two nodes.

To prune parts of the solution space, the COMODO algorithm utilizes optimistic estimates (see Sections 2.1 and 3.3). After the set of the k best community patterns has been obtained, it is ready for application, e.g., for presentation to the user, or for tasks like recommendation or personalization of services.

3.2. Description-Oriented Community Detection using Subgroup Discovery

For mining community patterns, we propose the COMODO algorithm which is based on the SD-Map* algorithm [7] for subgroup discovery, extended for description-oriented community detection. COMODO conducts an exhaustive search using *extended frequent pattern trees* [10], by traversing a representation of the solution space compiled into a *community pattern tree*. This tree is a compact version of the database D that also contains relevant information about the graph structure. Before it is created, we apply preprocessing described below.

3.2.1. Preprocessing

Since the communities considered in our approach do not contain isolated nodes, we can describe them as sets of edges. The advantage of such a description is due to the fact that – as described above – many community evaluation measures focus on edges rather than on nodes. Therefore, we transform the data (of the given graph G and the database D containing the nodes' descriptive information) into a new data set focusing on the edges of the graph G: Each data record in the new data set represents an edge between two nodes. The attribute values of each such data record are the common attributes of the edge's two nodes. The rationale behind storing only the common attributes is the observation that an edge can only belong to a community described by a certain attribute value, if this respective attribute value is the same for both nodes of that edge. In the BibSonomy example consider two users u_1 and u_2 with tags t_1, t_2, and t_3 and t_1, t_3, and t_4 respectively. If u_1 had chosen u_2 as a friend, then the transformed data set would contain an edge with u_1 and u_2 as nodes and the tags t_1 and t_3 as description.

Each such data record also stores the two nodes of the respective edge and their degrees in G to have them available during the evaluation of the quality function q. This allows for a very efficient approach using only local information which can be compiled into a compact data structure as described below.

3.2.2. The COMODO *Algorithm*

The FP-growth algorithm (cf. [27]) for mining association rules, and the SD-Map* algorithm for fast exhaustive subgroup discovery [7] form the basis of COMODO. In particular, the (extended) FP-tree used by these algorithms is adapted for COMODO as described below. An (extended) FP-tree can be efficiently constructed by only two scans of the database and is then mined in a recursive divide-and-conquer manner, cf. [7, 36]. The FP-tree contains the frequent FP-nodes in a header table, and links to all occurrences of the frequent basic patterns in the FP-tree structure. In this way, the parameters (of combinations) of basic patterns can be easily retrieved. First, patterns containing only one basic pattern are mined. Then recursively, patterns conditioned on the occurrence of a (prefixed) complex pattern (as a set of basic patterns, chosen in the previous recursion step) are considered. For each following recursive step, a conditional FP-tree is constructed, given the conditional pattern base of a frequent basic pattern (FP-node). The conditional pattern base consists of all the prefix paths of such a FP-node, i.e., all the paths from the root node to the FP-node. Given the conditional pattern base, a (smaller) FP-tree is generated: the *conditional FP-tree* of the respective FP-Node with adapted frequency counts. If the conditional FP-Tree just consists of one path, then the community descriptions can be generated by considering all the combinations of the nodes contained in the path. Otherwise, the new tree is subjected to the next recursion step. We refer to [27] for more details on FP-Trees and FP-growth.

COMODO utilizes an extended FP-tree structure, called the *community pattern tree* (CP-tree) to efficiently traverse the solution space. The tree is built in two scans of the graph data set. The steps of the algorithm are described in Algorithm 1. As shown in the algorithm, we consider three options for pruning and sorting according to the current optimistic estimates:

1. **Sorting**: During the iteration on the currently active basic pattern queue when processing a (conditional) CP-tree, we can dynamically reorder the basic patterns that have not been evaluated so far by their optimistic estimate value. In this way, we evaluate the *more promising* basic patterns first. This heuristic can help to obtain and to propagate higher values for the pruning threshold early in the process, thus, helping to prune larger portions of the search space (line 11).

2. **Pruning**: We omit a branch, if the optimistic estimate for the conditioning basic pattern is below the threshold given by the k best community pattern qualities (line 13).

3. **Pruning**: When building a (conditional) community pattern tree, we can omit all the CP-nodes with an optimistic estimate below the mentioned quality threshold (line 14).

To efficiently compute the community evaluation functions together with their optimistic estimates COMODO stores additional information in the *community pattern nodes* (CP-nodes) of the CP-tree, depending on the used quality function. Each CP-node of the CP-tree captures information about the aggregated edge information concerning the database D and the respective graph. For each node, we store the following information:

- The basic pattern (selector) corresponding to the attribute value of the CP-node. This selector describes the community (given by a set of edges) covering the CP-node.
- The edge count m_C of the (partial) community represented by the CP-node, i.e., the aggregated count of all edges $E_C = \{(u,v) \in E : u,v \in C\}$ that are accounted for by the CP-node and its basic pattern, respectively.
- The set of nodes $V_C = \{u : (u,v) \in E_C, u \in C, v \in C\}$ that are connected by the set of edges E_C of the CP-node.

Each edge data record also stores the contributing nodes and their degrees (in- and out-degree in the directed case). Thus, for the evaluation of a community C only the inter-degrees $\bar{\mathrm{d}}_C$ of the nodes in C (for IAODF) or the number of inter-edges $\bar{m}_C$ (for SIDX) have to be determined from the graph G.

Based on optimistic estimates presented in the following section, COMODO can reorder, sort, and prune search branches during each step of the traversal of the solution space. All CP-nodes with an optimistic estimate below the quality of the lowest ranked community among the k best solutions found so far, can be safely pruned. The result of the COMODO algorithm is then the set of the top-k community patterns according to the applied community evaluation function.

Besides the already mentioned minimal support threshold, COMODO can make use of the *maxLength* threshold constraining the maximum length of a description when considering the size of the set of basic patterns that are included. This parametrization is optional and depends on the use case. To leave the description length unrestricted, the parameter can be set to infinity.

Algorithm 1 COMODO

Input: Graph G, database D, int k (maximal number of patterns), int *maxLength* (maximal length of a pattern), int τ_n (minimal community size)

1: Generate $D = transform(G, DB)$ as described in Section 3.2.1.
2: Generate initial community pattern tree $CPT = createCPT(D, \tau_n)$
3: Let *top-k* = Priority queue with $|\textit{top-k}| \leq k$
4: Call COMODO-Mine(CPT, $\{\}$, *top-k*)

Output: *top-k*

procedure COMODO-Mine

Input: Current community pattern tree CPT, pattern $\hat{p}$, priority queue *top-k*

1: COM = new dictionary: $basicpattern \rightarrow pattern$
2: $minQ = minQuality(\textit{top-k})$
3: **for all** b in CPT.**getBasicPatterns do**
4: $p = createRefinement(\hat{p}, b)$
5: $COM[b] = p$
6: **if** $size(p, CPT) \geq \tau_n$ **then**
7: **if** $quality(p, F) \geq minQ$ **then**
8: $addToQueue(\textit{top-k}, p)$
9: $minQ = minQuality(\textit{top-k})$
10: **if** $length(\hat{p}) + 1 < maxLength$ **then**
11: $refinements = sortBasicPatternsByOptimisticEstimateDescending(COM)$
12: **for all** b in *refinements* **do**
13: **if** $optimisticEstimate(COM[b]) \geq minQ$ **then**
14: $CCPT = getConditionalCPT(b, CPT, minQ)$
15: Call COMODO-Mine($CCPT$, $COM[b]$, *top-k*)

3.2.3. Optional Postprocessing

For presentation and application of the result set of patterns, optional post-processing can be applied. For example, one can cluster the pattern extensions, i.e., the communities according to their overlap using a similarity measure such as the Jaccard coefficient, e. g., [2, 11, 55]. Patterns with a similar extension are then collected into one cluster. Usually, this helps for the assessment by the user, since one community can potentially be described by several patterns. Additionally, pattern set selection techniques such as weighted covering, cf. [33, 2], methods for selecting pattern teams [29], relevancy filtering [2, 37], or greedy covering approaches [42] can be applied in order to obtain a reduced set of patterns.

Postprocessing options need to be selected according to the requirements of the application and the analytical use case. In an explorative approach, usually the first technique using clustering provides a suitable range of filtering and pattern presentation options. Often, pattern filters can also be applied. For example, it is sometimes conventient, to prune patterns not fulfilling a minimal improvement constraint [14] with respect to the quality of a superpattern (generalization).

3.3. Optimistic Estimates for Efficient Mining

In the following, we introduce optimistic estimates for the typical community evaluation functions listed in Section 2.3, i.e., the segregation index, the inverse average ODF, and the modularity.

Making use of the minimum support threshold τ_n we can first observe the following inequality for each subcommunity C' of a community C, with a size above the minimal size threshold τ_n, i. e., $|C'| \geq \tau_n$:

$$\bar{m}_{C'} = \sum_{i=1}^{n_{C'}} \bar{\delta}_{C'}(i) \geq \sum_{i=1}^{n_{C'}} \bar{\delta}_{C}(i) \geq \sum_{i=1}^{\tau_n} \bar{\delta}_{C}(i).$$

Here, we assume that the values $\bar{\delta}_{C}(i)$, $i = 1,\ldots,n_C$ and $\bar{\delta}_{C'}(i)$, $i = 1,\ldots,n_{C'}$ are the inter-degrees of the nodes in C and C' respectively in *ascending order*, such that $\bar{\delta}_{C}(i)$, $i = 1,\ldots,\tau_n$ denote the minimal τ_n inter-degrees with respect to C.

3.3.1. Segregation Index

Proposition 3.1. *An optimistic estimate for* $\mathrm{SIDX}(C)$ *is given by*

$$\mathrm{oe}(\mathrm{SIDX}(C)) \;:=\; 1 - \frac{n(n-1)}{2m} \max\left\{ \frac{\sum_{i=1}^{\tau_n} \bar{\delta}_{C}(i)}{p(C)}, \min_{t=\tau_n}^{n_C} \left\{ \frac{\sum_{i=1}^{t} \bar{\delta}_{C}(i)}{t(n-t)} \right\} \right\},$$

$$\textit{where } p(C) := \begin{cases} \frac{n^2}{4}, & \textit{if } n_C \geq \frac{n}{2}, \\ n_C(n-n_C) & \textit{otherwise} \end{cases}.$$

Proof. For a subcommunity $C' \subseteq C$ with $|C'| \geq \tau_n$ we have

$$\begin{aligned} \mathrm{SIDX}(C') &= 1 - \frac{n(n-1)}{2m} \frac{\bar{m}_{C'}}{n_{C'}(n-n_{C'})} \leq \\ &\leq 1 - \frac{n(n-1)}{2m} \frac{\sum_{i=1}^{n_{C'}} \bar{\delta}_{C}(i)}{n_{C'}(n-n_{C'})} \leq \qquad (8) \\ &\leq 1 - \frac{n(n-1)}{2m} \frac{\sum_{i=1}^{\tau_n} \bar{\delta}_{C}(i)}{\max_{t=\tau_n}^{n_C}\{t(n-t)\}}. \qquad (9) \end{aligned}$$

From (8) it is clear that $1-\frac{n(n-1)}{2m}\min_{t=\tau_n}^{n_C}\left\{\frac{\sum_{i=1}^{t}\bar{\delta}_C(i)}{t(n-t)}\right\}$ is an optimistic estimate for SIDX(C). On the other hand we have $\max_{t=\tau_n}^{n_C}\{t(n-t)\}=p(C)$, since $t(n-t)$ has its maximum at $t=\frac{n}{2}$. Together with (9) we obtain $1-\frac{n(n-1)}{2m}\frac{\sum_{i=1}^{\tau_n}\bar{\delta}_C(i)}{p(C)}$ as another optimistic estimate. □

3.3.2. Inverse Average ODF

Proposition 3.2. *For the inverse Average-ODF let* $\tilde{\mathrm{d}}_C(u):=\frac{\bar{\mathrm{d}}_C(u)}{\mathrm{d}(u)}$ *and* $\tilde{\bar{\delta}}_C(i)$, $i=1,\ldots,n_C$ *as these ratios for all nodes in C in ascending order. Then*

$$\mathrm{oe}(\mathrm{IAODF}(C)):=1-\frac{1}{\tau_n}\sum_{i=1}^{\tau_n}\tilde{\bar{\delta}}_C(i)$$

is an optimistic estimate for IAODF(C).

Proof. For a subcommunity $C'\subseteq C$ with $|C'|\geq\tau_n$ we have

$$\begin{aligned}
\mathrm{IAODF}(C') &= 1-\frac{1}{n_{C'}}\sum_{u\in C'}\frac{|\{\{u,v\}\in E:v\in V\setminus C'\}|}{\mathrm{d}(u)}\leq\\
&\leq 1-\frac{1}{n_{C'}}\sum_{u\in C'}\frac{|\{\{u,v\}\in E:v\in V\setminus C\}|}{\mathrm{d}(u)}=\\
&= 1-\frac{1}{n_{C'}}\sum_{u\in C'}\tilde{\mathrm{d}}_C(u)\leq 1-\frac{1}{n_{C'}}\sum_{i=1}^{n_{C'}}\tilde{\bar{\delta}}_C(i)\leq\\
&\leq 1-\frac{1}{\tau_n}\sum_{i=1}^{\tau_n}\tilde{\bar{\delta}}_C(i)\,.
\end{aligned}$$

□

3.3.3. Modularity

Proposition 3.3. *An optimistic estimate for the* local modularity contribution *can be derived based only on the number of edges* m_C *within the community:*

$$\mathrm{oe}(\mathrm{MODL}(C))=\begin{cases}0.25, & \text{if } m_C\geq\frac{m}{2},\\ \frac{m_C}{m}-\frac{m_C^2}{m^2}, & \text{otherwise.}\end{cases}$$

Proof. Using Equation 1 we obtain:

$$\begin{aligned}
\mathrm{MODL}(C) &= \frac{m_C}{m} - \sum_{u,v \in C} \frac{\mathrm{d}(u)\,\mathrm{d}(v)}{4m^2} = \\
&= \frac{m_C}{m} - \frac{1}{4m^2} \sum_{u \in C} \mathrm{d}(u) \sum_{v \in C} \mathrm{d}(v) = \\
&= \frac{m_C}{m} - \frac{1}{4m^2} \sum_{u \in C} \mathrm{d}(u)(2m_C + \bar{m}_C) = \\
&= \frac{m_C}{m} - \frac{1}{4m^2} (2m_C + \bar{m}_C)^2 \leq \\
&\leq \frac{m_C}{m} - \frac{m_C^2}{m^2} = \\
&= \text{o\^{e}}(\mathrm{MODL}(C)).
\end{aligned}$$

Note that the optimistic estimate is only dependent on m_C, i.e., the number of edges covered by the community s. Therefore, every subgroup $s^* \subseteq s$ that is a refinement of s will cover at most m_C edges.

The function $\text{o\^{e}}(\mathrm{MODL}(C))$ is a concave function since its derivative

$$\text{o\^{e}}(\mathrm{MODL}(C))' = \frac{1}{m} - \frac{2m_C}{m^2}$$

is monotonically decreasing. Therefore, the function has its only maximum at $\frac{m}{2}$, for $m \neq 0$.

We consider two cases: If $m_C \geq \frac{m}{2}$, then the maximal modularity can be obtained at point $\frac{m}{2}$. Otherwise, for all $m_C < \frac{m}{2}$, $\text{o\^{e}}(\mathrm{MODL}(C))$ is decreasing in m_C, and thus, $\text{o\^{e}}(\mathrm{MODL}(C))$ is an optimistic estimate for $\mathrm{MODL}(C)$. □

3.3.4. Modularity (directed graphs)

Next, we cover the directed version of the local modularity contribution. Similarly as above, we obtain the same estimate as before for the local modularity contribution.

Proposition 3.4. *For the directed modularity an optimistic estimate is given by*

$$\begin{aligned}
\mathrm{dMODL}(C) &= \frac{1}{m}\sum_{u,v\in C}\left(A_{u,v}-\frac{\mathrm{d}^{\mathrm{in}}(u)\,\mathrm{d}^{\mathrm{out}}(v)}{m}\right) = \\
&= \frac{m_C}{m}-\frac{1}{m^2}\sum_{u\in C}\mathrm{d}^{\mathrm{in}}(u)\sum_{v\in C}\mathrm{d}^{\mathrm{out}}(v) \leq \\
&\leq \frac{m_C}{m}-\frac{m_C^2}{m^2} = \\
&= \hat{\mathrm{oe}}(\mathrm{MODL}(C)).
\end{aligned}$$

Finally, it is worth noting that all these estimates work also for weighted graphs, simply using the adjustments mentioned in Section 2.3.

3.3.5. Node-Degree – Edge Optimization

The optimistic estimates described above are applicable for the general problem of community detection, for arbitrary community allocations. However, Equation 7 allows for a convenient optimization, whenever the inter-degrees and total-degrees of the considered nodes are estimated as, e.g., for the segregation index, and the inverse average ODF. Due to the data construction outlined above, we can restrict the node selection as follows: Whenever a subset of the nodes with minimal inter-degrees and total degrees is selected, we can collect the set of nodes in such a way that we always consider "minimal edges" (concerning the respective parameters) contributing two nodes each.

4. Related Work

Community detection methods can be classified according to several dimensions. We distinguish between methods that detect disjoint communities, i.e., where actors in a network can only belong to exactly one community, and those that allow overlapping communities, where actors can belong to multiple communities at the same time. Furthermore, we distinguish between methods that work on extended (attributed) graphs, e. g., with descriptive information about the nodes, and methods that work on the plain graph structure. Below, we discuss related work concerning these issues in greater detail, including several basic methods working on simple graphs and summarizing community quality functions. After that, we elaborate on methods for detecting overlapping communities, before we focus on more recent methods for multi-dimensional and descriptive methods.

4.1. Basics of Community Detection

Communities and cohesive subgroups have been extensively studied in social sciences, e. g., using social network analysis methods [61]. Later, the analysis of (complex) networks and link structures has been picked up in physics and computer science as an important research direction, e. g., analyzing online and offline social interaction networks [3, 4].

Wasserman and Faust [61] discuss social network analysis in depth and provide an overview on the analysis of subgroups/communities in graphs, including clique-based, degree-based and matrix-perturbation-based methods. Furthermore, Newman et al. [49, 50, 51] propose several algorithms for community detection, formalizing the notions of interesting community structures, and introducing the modularity quality measure. In addition, Fortunato and Castellano [18] discuss various aspects connected to the concept of community structure in graphs and its detection. Moreover, Fortunato [17] also presents a thorough survey on the state of the art community detection algorithms in graphs, focussing on detecting *disjoint* communities.

For assessing the quality of a community, usually not only the community's density is assessed but the connection density of the community is compared to the density of the rest of the network [49]. The core idea of the evaluation function is to apply an objective evaluation criterion, for example, for the modularity the number of connections within the community compared to the statistically "expected" number based on all available connections in the network, and to prefer those communities that optimize the evaluation function. Besides modularity, prominent examples of community quality measures include for example, the segregation index [19] and the inverted average out-degree fraction [66].

A thorough empirical analysis of the impact of different community mining algorithms and their corresponding objective function on the resulting community structures is presented in [39], based on the analysis of community structure in graphs (as presented in [38]). Typically, using one of the methods mentioned above, a global partitioning of a graph is obtained. In contrast, the approach presented in this paper obtains *overlapping communities*, so that a node can be part of multiple communities. Additionally, not only the (plain) graph structure is exploited for detecting communities, but also descriptive information contained in the attributed graph is used in a *description-oriented way*, while applying standard community quality functions.

4.2. Detecting Overlapping Communities

Overlapping communities allow an extended modeling of actor–actor relations in social networks: Nodes of a corresponding graph can then participate in multiple communities. This is also typically observed in real-world networks regarding different complementary facets of social interactions [53, 46]. Concerning quality measures, extensions of the modularity metric for handling overlapping communities are described in [48, 52, 43]. As a general option for detecting communities, Tsourakakis et al. [60] provide a framework for finding dense subgraphs, finding top-k optimal quasi-cliques, which also enables overlapping communities.

A general overview on algorithms for overlapping community detection is provided by Xie et al. [64] as comprehensive survey. Clique percolation methods, proposed by Palla et al. [53], detect k-cliques and then merge them into overlapping communities. An extension for directed networks is described in [54]. Furthermore, Kumpula et al. [31] present an extension for fast clique percolation. Xie and Szymanski [65] present methods extending the idea of label propagation [56]: The LabelRank algorithm [65] stabilizes the propagation dynamics and randomness typically observed in label propagation approaches. Furthermore, [63] extends on that towards directed and weighted networks. Lancichinetti et al. [32] describe an approach for overlapping and hierarchical community structure using a local community metric. The presented metric itself is computed locally but still assesses a global clustering.

The methods that are most relevant to the approach in this paper concern statistical and local optimization algorithms: These include the COPRA [24] algorithm by Gregory using label-propagation of neighboring nodes until a consensus is reached, and the MOSES [45] algorithm by McDaid and Hurley using statistical model-based techniques. Both approaches aim at similar results as our proposed method concerning the overlapping nature of the obtained communities and the applied measures.

In contrast to the approaches mentioned above, the method proposed in this paper mines the graph structure for obtaining overlapping communities by focusing on local patterns, not on global models. This implies that we do not aim at describing a complete community model with a comprehensive coverage of the graph. Instead, we retrieve the k best (overlapping) communities according to a given community quality measure. In addition, the COMODO algorithm makes use of descriptive (label) information in addition to the network structure: It focuses on *explicit* descriptions of communities and directly searches for the top k descriptive communities according to standard community evaluation measures.

4.3. Community Detection and Description

While the methods described above only focus on the graph structure for mining communities, richer graph representations, i. e., *labeled graphs*, enable approaches that specifically exploit the descriptive information of the labels assigned to nodes and/or edges of the graph. Nodes of a network representing users, for example, can be labeled with tags that the respective users utilized in social bookmarking systems, as in our BibSonomy example. Further possible descriptive information relates to interests or demographical information.

Overall, there are several methods that somehow consider community detection and description. One approach for generating descriptions in a postprocessing step is given by deriving topics from the set of communities. Li et al. [40] first detect all communities, then identify interesting topics. Their approach combines Latent Dirichlet Allocation and the Girvan-Newman community detection method, cf. [23]. Kwan and Datta [41] use a topological approach by identifying central actors (celebrities) which are then used to derive the topics for the respective followers. Gargi et al. [22] present a multi-level approach for topic discovery and exploration. However, no concise description of the collected communities is obtained. Only a simple naming method based on the content of the resources is discussed; better naming and description methods are proposed for future work.

While the approaches above as well as our approach focus on static representations of graphs and communities, others consider dynamic and time-based structures. The MetaFac algorithm [44], for example, uses extended graph factorization for detecting (global) community allocations in a time-variant analysis considering multi-relational data. The graph factorization is implemented based on non-negative tensor factorization techniques. In contrast to our approach, the description of a community is not part of the core community detection step. It can only be obtained through post-processing, e. g., as top keywords in a probabilistic approach or as the top community members according to a tf-idf ranking.

The methods mentioned above only provide a kind of description by producing a summary of labels that occur in a given community. Furthermore, as discussed above, either rather simple techniques are applied for deriving the topics, or (complex) distributions on topics are returned. Therefore, such approaches do not convey *explicit descriptions* for the characterization of a community. In contrast, we focus on descriptive community patterns, represented by logical formulas on the values of the descriptive features that are true for all nodes of a community.

Concerning methods that focus on such descriptions in general, [1] presents an approach for community detection using features identified by frequent pattern mining; closed frequent patterns are derived and are then used for creating a social

network model based on an entropy analysis. However, the network structure itself is not exploited. Similarly, Sese et al. [58] extract subgraphs with common itemsets, i. e., itemset-sharing subgraphs. Given a labeled graph, itemset-sharing subgraphs can then be enumerated. However, this approach also does not consider the density of graphs, nor any community measures.

Focusing on methods for generating *explicit descriptions connected with the graph structure*, we distinguish between two types of approaches: first, methods that mainly work on the graph structure but apply descriptive information for restricting the possible sets of communities; second, methods that mine descriptive patterns for obtaining community candidates evaluated using the graph structure.

As a representative of the first type, Moser et al. [47] combine the concepts of dense subgraphs and subspace clusters for mining cohesive patterns. Starting with quasi-cliques, these are expanded until constraints regarding the description or the graph structure are violated. Similarly, Günnemann et al. [26] combine subspace clustering and dense subgraph mining, also interleaving quasi-clique and subspace construction. However, in contrast to our approach, they apply specialized threshold-based interestingness assessments of the found patterns, e. g., focusing on the densities of quasi-cliques concerning the graph structure.

As an example for the second type outlined above, Galbrun et al. [21] propose an approach for the problem of finding overlapping communities in graphs and social networks that aims to detect the top-k communities such that the total edge density over all k communities is maximized. This also relates to a maximum coverage problem for the whole graph. For labeled graphs each community is required to be described by a set of labels. The three algorithmic variants proposed by Galbrun et al. apply a greedy strategy for detecting dense subgroups, and restrict the result set of communities, such that each edge can belong to at most community. Therefore, this partitioning involves a global approach on the community quality, in contrast to our local approach. Silva et al. [59] study the correlation between attribute sets and the occurrence of dense subgraphs in large attributed graphs. The proposed method considers frequent attribute sets using an adapted frequent item mining technique, and identifies the top-k dense subgraphs induced by a particular attribute set, called structural correlation patterns. However, similar to the methods discussed above, the method focuses on quasi-cliques, and does not apply selectable (standard) community quality measures.

The approach that is most relevant to our presented approach is the DCM method presented by Pool et al. [55]. It includes a two-step process of community detection and community description. A heuristic approach is applied for discovering the top-k communities. Pool et al. utilize a special interestingness function

which is based on counting outgoing edges of a community similar to the IAODF measure; for that, they also demonstrate the trend of a correlation with the modularity function. In contrast to Pool et al., our approach discovers and optimizes communities (as subgroups) directly. This can yield more compact conjunctive descriptions, i.e., no disjunctions of several subgroup descriptions have to be used for the characterization of a community. Furthermore, the COMODO algorithm applies exhaustive search, in contrast to the heuristic strategy of DCM.

Alltogether, the method proposed in this work extends our earlier work presented in [8] and [9, 10] in three ways: (1) It applies subgroup discovery to community mining by defining the applied quality function on the graph structure (similar to exceptional model mining [35, 5]), (2) it utilizes novel optimistic estimates for efficiently searching the description space, (3) and it directly optimizes the choice of communities with respect to a given standard community measure at the same time. Furthermore, the applied exhaustive approach using optimistic estimates guarantees the discovery of the top-k communities according to a given quality measure. To the best of the authors' knowledge, no description-oriented community detection approach applying an exhaustive branch-and-bound methods has been proposed so far.

5. Experiments

In the following, we first describe the data sets, before we present the conducted experiments and discuss the results. We focus on evaluating the efficiency of the presented pruning approach considering the search steps of the COMODO algorithm. Furthermore, we discuss properties of the discovered communities in order to assess their validity.

5.1. Networks in Social Bookmarking Systems

For our experiments we used five data sets from three social media systems: We utilized user networks from the social bookmarking and resource sharing systems BibSonomy, delicious and last.fm. These graphs arise from user interactions and are typically found in many social media applications. We utilized an anonymized dump of BibSonomy containing all public bookmark and publication posts until January 27, 2010. The delicious and last.fm data sets are publicly available and were obtained from the *HetRec* workshop [16] at Recsys 2011.

In the following, we provide a detailed overview on the data sets used in the experiments:

- BibSonomy: The data contains 175,521 tags, 5,579 users, 467,291 resources, 2,120,322 tag assignments, and also friendship links for 700 users.
  - The *friend graph* $G_F = (V_F, E_F)$ of BibSonomy is a directed graph with $(u,v) \in E_F$ iff user u has added user v as a friend in BibSonomy.
  - The *click graph* of BibSonomy $G_C = (V_C, E_C)$ is a directed graph with $(u,v) \in E_C$ iff user u has clicked on a link on the user page of user v in BibSonomy.
  - The *visit graph* of BibSonomy $G_V = (V_V, E_V)$ is a directed graph with $(u,v) \in E_V$ iff user u navigated to v's user page in BibSonomy.

- delicious: The data set contains 1,861 users, 7,664 bi-directional user relations (i. e., 15328 user (u,v) pairs) and 53,388 tags. The *delicious friend graph* $G_D = (V_D, E_D)$ is an undirected graph with $(u,v) \in E_D$ iff user u has added user v as a friend in delicious.

- last.fm: The last.fm data set contains 1,892 users, 12,717 bi-directional user friend relations and 11,946 tags. The *last.fm friend graph* $G_L = (V_L, E_L)$ is an undirected graph with $(u,v) \in E_L$ iff user u has added user v as a friend in last.fm.

The friendship relations are explicit relations, while the data for the other relations (click and visit) can be obtained from the "*click log*" of BibSonomy, consisting of entries which are generated whenever a logged-in user clicked on a link in BibSonomy. Table 1 presents some high level statistics for the five network structures.

Table 1: High level statistics for all graphs used for the experiments.

	G_V (visit)	G_C (click)	G_F (friend)	G_D (delicious)	$G_L(last.fm)$
$\|V\|$	3381	1151	700	1861	1892
$\|E\|$	8214	1718	1012	7664	12717
density	0.0014	0.0025	0.0032	0.0044	0.0071
deg_{max}	1667	275	34	90	119

5.2. Evaluation Data and Setting

In the following, we discuss the applied data preprocessing steps below and describe the evaluation setup. In the next section, we present an evaluation of the efficiency and pruning performance of the COMODO algorithm for different community quality measures and their optimistic estimates, respectively.

After that, we focus on properties of the obtained communities in order the assess their validity. As a benchmark for COMODO, we apply two popular algorithms for detecting overlapping communities, i. e., the algorithms MOSES [45] and the COPRA [24], as well as the DCM algorithm [55] for descriptive community detection. We compare the communities obtained using COMODO, COPRA, MOSES, and DCM and discuss the statistical and descriptive properties of their results in detail. We experimented with additional baselines, however, for clarity we focus on the most similar and comparable algorithms. In our experimentation in an exhaustive setting with $\tau_n = 10$, for example, the GAMer algorithm [26] did only complete on the two smallest datasets; in such a setting the algorithm requires increased runtime and memory (exceeding 16 GB for the other data sets), as also acknowledged by its authors.

The attributes describing a user are given by the set of tags that the respective user assigned to resources. We applied standard string preprocessing and cleaning techniques, e. g., normalizing tags concerning whitespace and special characters in order to handle writing variants. In addition, we focused on tags having at least three characters.

5.3. Evaluation: Efficiency using Optimistic Estimates

In our first evaluation, we focused our experimentation on measuring the impact of the proposed pruning procedures. We estimated the effect of the optimistic estimates regarding the efficiency of the presented community detection approach.

5.3.1. Experiments

In the following, we outline the results of applying COMODO to the presented data sets. In order to evaluate the efficiency, we count the number of search steps, i. e., community allocations that are considered by the COMODO algorithm. We compared the total number of search steps (no optimistic estimate pruning) to optimistic estimate pruning using different commmunity quality measures. Additionally, we measured the impact of using different minimal community size thresholds. The results are shown in Figures 1–4 for the BibSonomy click graph, the delicious friend graph, the BibSonomy friend graph, and the last.fm contact graph, for $k = 10, 20, 50$ and minimal size thresholds $\tau_n = 10, 20$. The BibSonomy visit graph is shown in Figure 5. More details on the achieved reduction of the search space using the optimistic estimate functions for all considered graphs and parameters is shown in Table 2. The table shows the number of search steps/hypotheses during the mining process using the optimistic estimates introduced in Section 3.3.

The large, exponential search space can be exemplified, e. g., for the click graph with a total of about $2 \cdot 10^{10}$ search steps for a minimal community size threshold $\tau_n = 10$, or the visit graph with a total of about 10^{10} search steps for a minimal community size threshold $\tau_n = 20$.

Furthermore, Figure 5 shows the considered search steps for the BibSonomy

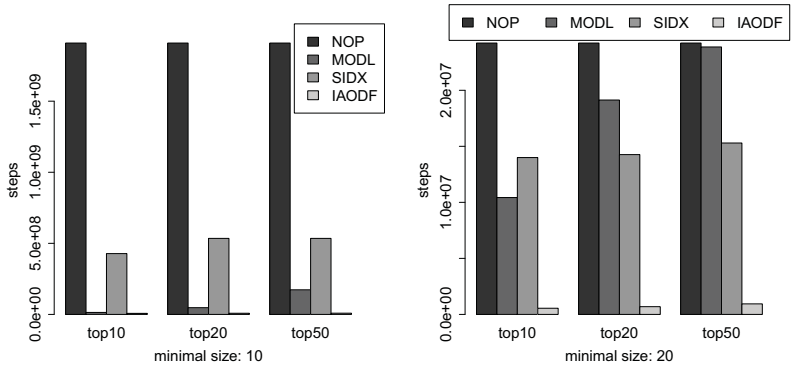

Figure 1: BibSonomy click graph: Search steps with no optimistic estimate pruning (*NOP*) vs. community quality functions with optimistic estimate pruning: MODL (Local Modularity), SIDX (Segregation Index) and IAODF (Inverse Average-ODF), for minimal size thresholds $\tau_n = 10, 20$.

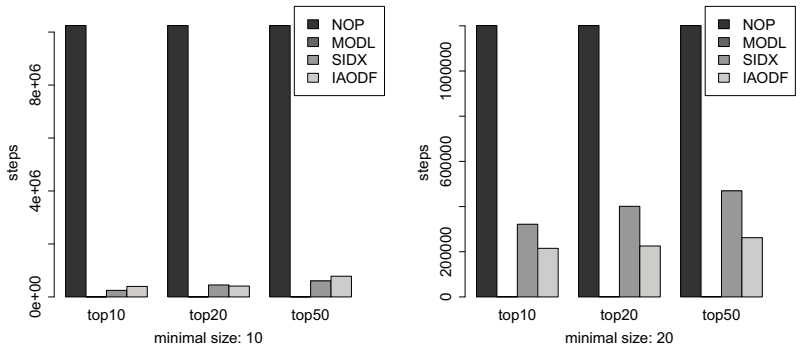

Figure 2: Delicious friend graph: Search steps with no optimistic estimate pruning (*NOP*) vs. community quality functions with optimistic estimate pruning: MODL (Local Modularity), SIDX (Segregation Index) and IAODF (Inverse Average-ODF), for minimal size thresholds $\tau_n = 10, 20$.

visit graph; for $\tau_n = 20$ we only show the results for the modularity[4]. In addition, we include the results of $\tau_n = 50$ for all considered measures. These show similar trends to the results shown in Table 2. We discuss these in more detail below.

[4]Here, the other experiments did not terminate within a running time of at most 100 hours.

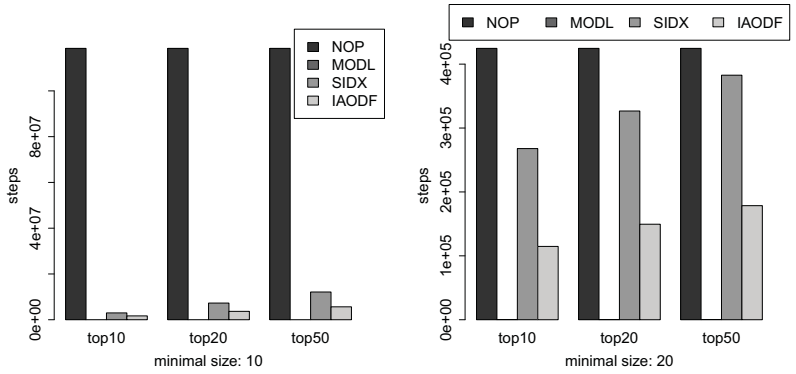

Figure 3: BibSonomy friend graph: Search steps with no optimistic estimate pruning (*NOP*) vs. community quality functions with optimistic estimate pruning: MODL (Local Modularity), SIDX (Segregation Index) and IAODF (Inverse Average-ODF), for minimal size thresholds $\tau_n = 10, 20$.

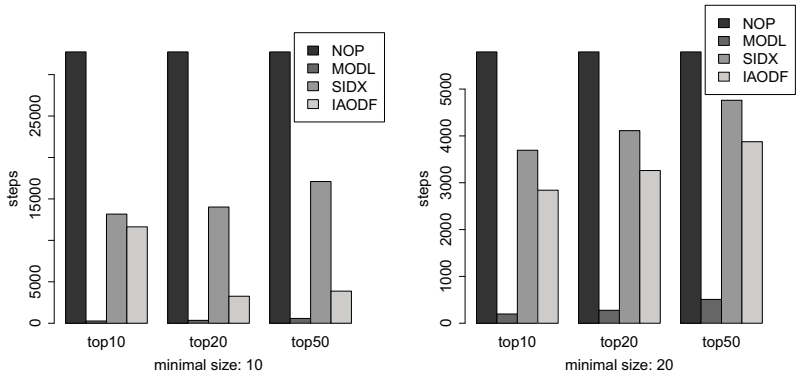

Figure 4: last.fm friend graph: Search steps with no optimistic estimate pruning (*NOP*) vs. community quality functions with optimistic estimate pruning: MODL (Local Modularity), SIDX (Segregation Index) and IAODF (Inverse Average-ODF), for minimal size thresholds $\tau_n = 10, 20$.

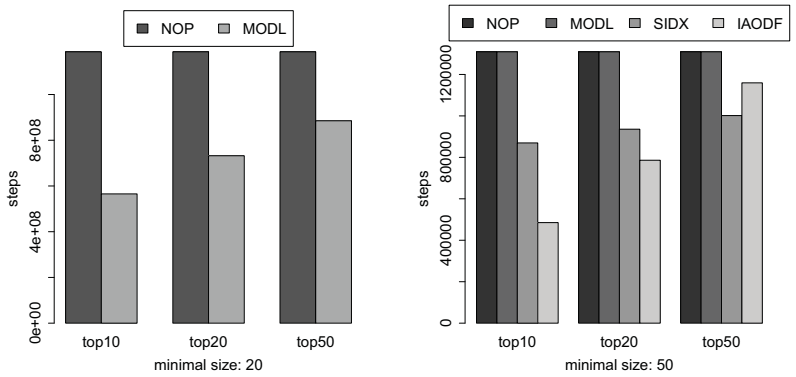

Figure 5: BibSonomy visit graph: Search steps with no optimistic estimate pruning (*NOP*) vs. community quality functions with optimistic estimate pruning: MODL (Local Modularity), SIDX (Segregation Index) and IAODF (Inverse Average-ODF), for minimal size thresholds $\tau_n = 20, 50$ (As discussed above, we only include the modularity based measures for $\tau_n = 20$).

5.3.2. Discussion

In comparison to the segregation index (SIDX) and the inverse Average-ODF (IAODF), the pruning for the modularity-based functions (MODL, dMODL) yields a better pruning performance by orders of magnitude in most cases. The IAODF and the SIDX are second and third, with the exception of the clickGraph, where the use of IAODF required fewer steps than the other measures. Thus, a reduction of the search space which allows for effective community mining using the COMODO algorithm was observed especially for the optimistic estimates of the local modularity (MODL) for the more dense graphs (delicious, last.fm), while IAODF outperforms the other measures on the sparser graphs. IAODF implicitly considers the size of the community since it utilizes the degree information of the nodes contained in the community and the respective fractions of the outgoing nodes. With a certain minimal support this proves rather efficient.

The local modularity gives more importance to the number of edges that are contained in the community, while, e. g., SIDX considers the fraction of the number of edges within the community and the number of inter-edges. In this way, very small communities can also obtain a high quality value, even if the minimal support threshold is reached. Overall these optimistic estimates show huge pruning potential for many applications, especially considering the local modularity measure.

Table 2: Impact of optimistic estimate pruning for different community quality functions, cf. Section 3.3, for the data sets as described in Section 5.3.1. The table shows the number of search steps (absolute number and percentage relative to the maximum number of possible steps without pruning) needed for finding the top $k = 10, 20, 50$ best communities given a minimal support threshold $\tau_n = 10, 20, 50$. Each block presents the results for one graph. The first line shows a run without any pruning, the following lines consider the three quality functions MODL/dMODL, SIDX, and IAODF (pruning by the support threshold and the respective optimistic estimate). In bold are the smallest number of steps/hypothes for each setting of k and τ_n.

			$\tau_n = 10$		$\tau_n = 20$		$\tau_n = 50$	
	Top k	Method	Abs.	Rel.	Abs.	Rel.	Abs.	Rel.
clickGraph	No Pruning		1,908,999,540	100,00	24,216,078	100,00	5,401	100,00
	10	dMODL	14,005,822	0.73	10,427,700	43.06	5,401	100.00
		SIDX	427,922,071	22.42	13,996,913	57.80	4,541	84.08
		IAODF	**7,361,666**	**0.39**	**550,741**	**2.27**	**1,028**	**19.03**
	20	dMODL	47,618,946	2.49	19,131,223	79.00	5,401	100.00
		SIDX	534,826,208	28.02	14,262,293	58.90	4,693	86.89
		IAODF	**8,133,550**	**0.43**	**691,758**	**2.86**	**1,602**	**29.66**
	50	dMODL	173,256,424	9.08	23,866,621	98.56	5,401	100.00
		SIDX	1,237,881,774	64.84	15,296,020	63.16	5,401	100.00
		IAODF	**8,643,947**	**0.45**	**940,925**	**3.89**	**2,977**	**55.12**
delicious	No Pruning		10,245,840	100.00	1,201,270	100.00	43,091	100.00
	10	MODL	**1,156**	**0.01**	**748**	**0.06**	**411**	**0.95**
		SIDX	248,511	2.43	321,810	26.79	36,550	84.82
		IAODF	396,902	3.87	215,126	17.91	19,257	44.69
	20	MODL	**1,226**	**0.01**	**818**	**0.07**	**481**	**1.12**
		SIDX	452,509	4.42	401,129	33.39	38,391	89.09
		IAODF	410,781	4.01	225,445	18.77	20,858	48.40
	50	MODL	**1,429**	**0.01**	**1,021**	**0.08**	**684**	**1.59**
		SIDX	604,654	5.90	469,594	39.09	39,726	92.19
		IAODF	781,584	7.63	262,173	21.82	22,869	53.07
friendGraph	No Pruning		118,582,294	100.00	424,669	100.00	203	100.00
	10	dMODL	**647**	**$5{\cdot}10^{-4}$**	**295**	**0.07**	**55**	**27.09**
		SIDX	2,984,815	2.52	267,830	63.07	203	100.00
		IAODF	1,682,046	1.42	114,736	27.02	191	94.09
	20	dMODL	**661**	**$6{\cdot}10^{-4}$**	**306**	**0.07**	**69**	**33.99**
		SIDX	7,286,700	6.14	326,641	76.92	203	100.00
		IAODF	3,666,524	3.09	149,476	35.20	199	98.03
	50	dMODL	**723**	**$6{\cdot}10^{-4}$**	**367**	**0.09**	**121**	**59.61**
		SIDX	12,118,021	10.22	382,693	90.12	203	100.00
		IAODF	5,642,829	4.76	178,501	42.03	203	100.00
last.fm	No Pruning		32,743	100.00	5,794	100.00	581	100.00
	10	MODL	**271**	**0.83**	**198**	**3.42**	**129**	**2.20**
		SIDX	13,170	40.22	3,694	63.76	537	92.43
		IAODF	11,633	35.53	2,841	49.03	449	77.28
	20	dMODL	**350**	**1.07**	**277**	**4.78**	**204**	**35.11**
		SIDX	14,023	42.83	4,112	70.97	566	97.42
		IAODF	13,500	41.23	3,261	56.28	486	83.65
	50	dMODL	**582**	**1.78**	**509**	**8.78**	**402**	**69.19**
		SIDX	17,103	52.23	4,761	82.17	572	98.45
		IAODF	16,021	48.93	3,876	66.90	551	94.84

5.4. Evaluation: Structure and Validity

In the following, we assess the validity of the presented approach. We focus on three baseline approaches: First, we consider methods focusing on the network structure. Since local community patterns can describe overlapping communities, we compare our approach to prominent approaches for detecting such overlapping communities. We consider the MOSES [45] and the COPRA [24] algorithms as a reference. Furthermore, we compare COMODO with the DCM algorithm [55] for descriptive-driven community detection. The latter enables the assessment of structure characteristics, significance, and description complexity.

5.4.1. Results

For the experiments, we first restrict our analysis to structural properties and measures considering the communities and their induced subgraphs – completely independent from the description data. Thus, when comparing COMODO to COPRA and MOSES, we aim to avoid a bias towards the additional descriptive information provided to COMODO. Considering the descriptions, we compare the description complexity of COMODO and DCM below. Table 3 shows some examples of the discovered community patterns for the last.fm data set.

Table 3: Ten examples of top ranked community patterns according to the local modularity (MODL, top) and the inverse Average-ODF (IAODF, bottom) together with their respective topic description, using the last.fm data set (cf. 5.1). The rows of the table show the different patterns consisting of conjunctions of tags.

Size	Community description
519	80s
240	gregorian_chant AND 80s
215	girl_groups AND 80s
171	atmospheric
122	synth_pop
32	psychedelic AND minimal
16	psychedelic AND 80s
10	psychedelic AND brit_rock AND classic_rock
10	death_rock AND minimal AND 80s
10	death_rock AND 80s AND doom_metal

In addition to the number of the discovered communities, we measure the respective mean sizes, densities and the overlap of the community sets. These measures provide a first overview of the properties of the communities (and induced subgraphs). However, for a comprehensive assessment, they need to be inspected with some insight, e. g., [57]. Therefore, we also evaluated the obtained

communities using the significance test described in [30] for testing the statistical significance of the density of a discovered subgraph. We required a minimal community size of $\tau_n = 10$ nodes. Since MOSES, COPRA and DCM do not accept a mimimum size as input, we applied a post-processing step for their discovered communities and filtered all communities with a size $n < \tau_n$. Additionally, for the COMODO algorithm we applied a *minimal improvement filter* [14] for the community patterns, and pruned all specializations for which the absolute difference to the quality of their parent patterns was smaller than $\tau_I = 0.01$. The communities marked with (*) in Table 4 show the unfiltered community distributions, since the application of the minimal support threshold $\tau_n = 10$ resulted in only one community each – with a size larger than the threshold τ_n. For the MOSES algorithm we did not observe many such small communities and could therefore apply the postprocessing procedure with $\tau_n = 10$; see Table 5 for the respective results. The DCM algorithm discovered a large set of rather small communities, e. g., with the smallest average size of 2.78 ± 1.57 for the friend graph, and the largest averages 7.90 ± 9.07 for last.fm. After filtering according to $\tau_n = 10$ we obtained small sets of communities as shown in Table 6.

Table 4: COPRA result statistics on the BibSonomy and hetrec data sets (column 1). We applied a filter on the community size selecting those with $n \geq 10$; for the entries marked with (*) this resulted only in one community each, therefore we included all communities in these cases. The table shows the counts of the obtained sets of communities, their mean size, density overlap, and the share (PS) of statistically significant communities according to a p-value of at least 10^{-6}.

Graph	Count	μ(Size)	μ(Density)	μ(Overlap)	PS (%)
clickGraph (*)	130	8.78 ± 88.76	0.01 ± 0.00	0.00 ± 0.00	1.00
friendGraph	7	56.43 ± 94.57	0.18 ± 0.10	0.01 ± 0.03	57.14
delicious	24	71.71 ± 263.50	0.58 ± 0.29	0.00 ± 0.00	54.17
last.fm (*)	39	48.56 ± 294.90	0.10 ± 0.30	0.00 ± 0.00	2.60
visitGraph (*)	23	145.74 ± 694.14	0.01 ± 0.00	0.00 ± 0.00	2.56

Table 5: MOSES result statistics on the BibSonomy and hetrec data sets (column 1). We applied a filter on the community size selecting those with $n \geq 10$. The table shows the counts of the obtained sets of communities, their mean size, density overlap, and the share (PS) of statistically significant communities according to a p-value of at least 10^{-6}.

Graph	Count	μ(Size)	μ(Density)	μ(Overlap)	PS (%)
clickGraph	11	16.73 ± 7.06	0.30 ± 0.07	0.04 ± 0.01	27.27
friendGraph	5	12.2 ± 3.83	0.50 ± 0.19	0.02 ± 0.03	20.00
delicious	68	17.34 ± 10.04	0.52 ± 0.23	0.01 ± 0.02	50.00
last.fm	92	29.53 ± 23.89	0.27 ± 0.09	0.01 ± 0.02	63.04
visitGraph	90	20.88 ± 11.75	0.27 ± 0.07	0.04 ± 0.02	26.67

Table 6: DCM result statistics on the BibSonomy and hetrec data sets (column 1). We applied a filter on the community size selecting those with $n \geq 10$. The table shows the counts of the obtained sets of communities, their mean size, density overlap, the share (PS) of statistically significant communities according to a p-value of at least 10^{-6}, and the average description length (ADL).)

Graph	Count	μ(Size)	μ(Density)	μ(Overlap)	ADL	PS (%)
clickGraph	5	10.8 ± 1.10	0.57 ± 0.11	0.09 ± 0.22	8.80	0.00
friendGraph	2	11 ± 1.41	0.43 ± 0.01	0.0 ± 0.0	3.70	0.00
delicious	20	15.05 ± 5.92	0.58 ± 0.16	0.02 ± 0.10	17.43	35.00
last.fm	15	21.00 ± 12.00	0.46 ± 0.10	0.10 ± 0.18	14.05	60.00
visitGraph	3	21 ± 6.93	0.58 ± 0.03	0.22 ± 0.30	12.00	66.67

5.4.2. Discussion

We first discuss the results of the reference algorithms before we compare them to the communities obtained by the COMODO algorithm. As shown in Tables 4 through 6 the communities discovered by COPRA and MOSES are rather small. The COPRA results exhibit a rather low and statistically not significant density with a low overlap. Concerning the community sizes, the MOSES results are rather similar. However, MOSES discovers better communities in terms of density, with statistically significant density values, yet rather smaller in size. The communities discovered by DCM are even slightly smaller than those of MOSES, yet (except for the friendGraph) have again higher densities. This is not surprising since both algorithms optimize for dense communities. However, we observe a relatively low number of communities that are larger than the minimal support threshold, leading to a small relative number of statistically significant communities regarding the density values.

Tables 7 and 8 show the structural and descriptive properties of the results obtained by COMODO. We observe that COMODO provides statistically significant results for almost all quality measures and minimal support thresholds concerning the community size and density. Especially for the MODL and dMODL quality measures always statistically significant communities are obtained. Additionally, the results can be easily tuned using the minimal support parameter (τ_n) yielding larger communities. The tables show that a suitable minimal support threshold ($\tau_n = 20$) yields significant results ($\geq 98\%$) for all quality measures, indicating these communities' validity. Finally, since DCM – like COMODO– produces communities with descriptions we can compare both algorithms regarding that aspect: We compare the average description length (ADL) as a measure of the description's complexity. From Tables 6 and 7 and 8 we can observe that COMODO yields strictly shorter community descriptions than DCM.

It is important to note that the presented method focuses on the description of communities while directly searching for the top k descriptive communities according to their quality. It is possible that there exist methods from the field of social network analysis, e.g., [38, 39, 50] that ignore the description space and work only on the network structure can theoretically discover communities with

Table 7: COMODO results on the BibSonomy data sets (column 1). The tables shows the counts of the obtained sets of communities, their mean size, density overlap, the share (PS) of statistically significant communities according to a p-value of at least 10^{-6}, and the average description length (ADL), columns 5-10, for the parameters *quality measure*, *minimal support threshold* (τ_n), and k for the top-k communities (count and k may differ due to the applied postprocessing filter.)

Graph	Measure	τ_n	k	Count	μ(Size)	μ(Density)	μ(Overlap)	ADL	PS (%)
clickGraph	dMODL	10	10	10	133.20 ± 45.67	0.03 ± 0.02	0.26 ± 0.12	1.00	100.00
			20	20	123.40 ± 43.27	0.03 ± 0.02	0.25 ± 0.10	1.00	100.00
			50	50	96.94 ± 59.49	0.04 ± 0.03	0.18 ± 0.13	1.18	100.00
		20	10	10	133.20 ± 45.67	0.03 ± 0.02	0.25 ± 0.12	1.00	100.00
			20	20	123.40 ± 43.27	0.03 ± 0.02	0.24 ± 0.10	1.00	100.00
			50	50	96.94 ± 59.49	0.04 ± 0.03	0.18 ± 0.12	1.18	100.00
	SIDX	10	10	10	14.60 ± 3.57	0.14 ± 0.08	0.22 ± 0.32	1.70	30.00
			20	20	13.90 ± 2.83	0.15 ± 0.07	0.28 ± 0.35	2.20	15.00
			50	25	13.00 ± 3.04	0.17 ± 0.08	0.32 ± 0.37	2.38	10.34
		20	10	10	89.80 ± 98.65	0.05 ± 0.03	0.10 ± 0.11	1.00	100.00
			20	20	73.45 ± 81.48	0.05 ± 0.03	0.09 ± 0.10	1.10	100.00
			50	50	64.02 ± 67.54	0.06 ± 0.03	0.16 ± 0.19	1.60	100.00
	IAODF	10	10	5	11.00 ± 0.00	0.18 ± 0.00	1.00 ± 0.00	3.33	0.00
			20	13	10.08 ± 0.28	0.20 ± 0.01	0.88 ± 0.15	3.38	0.00
			50	43	10.02 ± 0.15	0.20 ± 0.00	0.95 ± 0.09	4.00	0.00
		20	10	10	92.50 ± 98.86	0.07 ± 0.05	0.15 ± 0.17	1.80	100.00
			20	20	66.90 ± 82.68	0.09 ± 0.05	0.23 ± 0.21	2.45	100.00
			50	50	47.00 ± 58.98	0.09 ± 0.04	0.30 ± 0.22	2.98	100.00
friendGraph	dMODL	10	10	10	98.70 ± 36.24	0.07 ± 0.13	0.30 ± 0.16	1.00	100.00
			20	20	87.05 ± 35.49	0.08 ± 0.14	0.26 ± 0.16	1.05	100.00
			50	50	75.32 ± 27.44	0.07 ± 0.11	0.32 ± 0.17	1.42	100.00
		20	10	10	98.70 ± 36.24	0.07 ± 0.13	0.30 ± 0.16	1.00	100.00
			20	20	90.05 ± 31.58	0.06 ± 0.10	0.29 ± 0.14	1.05	100.00
			50	50	77.26 ± 24.83	0.05 ± 0.07	0.34 ± 0.15	1.42	100.00
	SIDX	10	10	10	10.40 ± 0.70	0.18 ± 0.09	0.56 ± 0.38	2.56	0.00
			20	16	11.81 ± 3.04	0.18 ± 0.10	0.33 ± 0.30	2.29	6.89
			50	39	11.82 ± 2.46	0.14 ± 0.07	0.29 ± 0.24	2.31	7.14
		20	10	10	78.50 ± 37.98	0.07 ± 0.13	0.24 ± 0.18	1.20	100.00
			20	20	72.25 ± 35.14	0.06 ± 0.09	0.23 ± 0.16	1.15	100.00
			50	50	68.60 ± 29.47	0.05 ± 0.06	0.29 ± 0.15	1.44	100.00
	IAODF	10	10	10	12.90 ± 3.87	0.24 ± 0.16	0.17 ± 0.27	2.10	20.00
			20	17	13.06 ± 3.36	0.21 ± 0.15	0.21 ± 0.27	2.06	23.52
			50	40	17.38 ± 24.25	0.16 ± 0.11	0.24 ± 0.25	2.20	14.63
		20	10	10	95.80 ± 35.50	0.06 ± 0.13	0.29 ± 0.15	1.00	100.00
			20	20	85.85 ± 33.64	0.05 ± 0.09	0.30 ± 0.15	1.15	100.00
			50	50	69.90 ± 28.49	0.05 ± 0.06	0.32 ± 0.15	1.64	100.00
visitGraph	dMODL	20	10	10	72.00 ± 41.09	0.06 ± 0.04	0.13 ± 0.21	1.00	100.00
			20	20	65.95 ± 30.07	0.06 ± 0.03	0.15 ± 0.18	1.15	100.00
			50	50	54.30 ± 23.29	0.07 ± 0.04	0.18 ± 0.18	1.64	100.00

Table 8: COMODO results on the hetrec data sets (column 1). The tables shows the counts of the obtained sets of communities, their mean size, density overlap, the share (PS) of statistically significant communities according to a p-value of at least 10^{-6}, and the average description length (ADL), columns 5-10, for the parameters *quality measure*, *minimal support threshold* (τ_n), and k for the top-k communities (count and k may differ due to the applied postprocessing filter.)

Graph	Measure	τ_n	k	Count	μ(Size)	μ(Density)	μ(Overlap)	ADL	PS (%)
delicious	MODL	10	10	10	551.80 ± 141.66	0.01 ± 0.00	0.33 ± 0.10	1.10	100.00
			20	20	460.00 ± 146.68	0.01 ± 0.00	0.33 ± 0.13	1.45	100.00
			50	50	377.80 ± 126.44	0.01 ± 0.00	0.27 ± 0.11	1.52	100.00
		20	10	10	551.80 ± 141.66	0.01 ± 0.001	0.33 ± 0.10	1.10	100.00
			20	20	460.00 ± 146.68	0.01 ± 0.002	0.33 ± 0.13	1.45	100.00
			50	50	377.80 ± 126.44	0.01 ± 0.003	0.27 ± 0.11	1.52	100.00
	SIDX	10	10	9	10.00 ± 0.00	0.20 ± 0.01	0.83 ± 0.31	4.60	0.00
			20	20	10.00 ± 0.00	0.20 ± 0.01	0.93 ± 0.23	4.71	0.00
			50	35	10.29 ± 0.67	0.16 ± 0.03	0.37 ± 0.28	4.58	0.00
		20	10	10	20.90 ± 1.29	0.06 ± 0.01	0.34 ± 0.19	4.50	100.00
			20	20	21.00 ± 1.03	0.06 ± 0.01	0.35 ± 0.20	4.50	100.00
			50	50	21.00 ± 1.20	0.06 ± 0.01	0.44 ± 0.22	4.60	100.00
	IAODF	10	10	10	10.30 ± 0.67	0.24 ± 0.21	0.29 ± 0.40	4.00	0.00
			20	16	10.19 ± 0.54	0.22 ± 0.17	0.51 ± 0.46	4.60	0.00
			50	35	10.26 ± 0.56	0.19 ± 0.16	0.24 ± 0.30	4.57	0.00
		20	10	10	21.60 ± 1.35	0.07 ± 0.01	0.45 ± 0.15	4.90	100.00
			20	20	22.25 ± 1.77	0.07 ± 0.01	0.50 ± 0.15	4.85	100.00
			50	50	22.10 ± 2.31	0.07 ± 0.01	0.43 ± 0.18	4.72	100.00
last.fm	MODL	10	10	10	294.10 ± 130.56	0.03 ± 0.01	0.24 ± 0.13	1.20	100.00
			20	20	257.25 ± 143.50	0.03 ± 0.01	0.18 ± 0.12	1.15	100.00
			50	50	180.28 ± 119.43	0.04 ± 0.03	0.15 ± 0.13	1.48	100.00
		20	10	10	294.10 ± 130.56	0.03 ± 0.01	0.24 ± 0.13	1.20	100.00
			20	20	257.25 ± 143.50	0.03 ± 0.01	0.18 ± 0.12	1.15	100.00
			50	50	180.28 ± 119.43	0.04 ± 0.03	0.15 ± 0.13	1.48	100.00
	SIDX	10	10	10	17.10 ± 11.29	0.12 ± 0.06	0.20 ± 0.25	2.30	20.00
			20	20	15.15 ± 8.59	0.12 ± 0.05	0.16 ± 0.20	2.45	25.00
			50	50	15.00 ± 6.65	0.11 ± 0.04	0.15 ± 0.20	2.48	24.00
		20	10	10	26.20 ± 7.10	0.07 ± 0.02	0.25 ± 0.25	2.00	100.00
			20	20	28.00 ± 10.83	0.07 ± 0.01	0.16 ± 0.22	2.20	100.00
			50	50	32.62 ± 22.55	0.07 ± 0.02	0.12 ± 0.18	2.12	100.00
	IAODF	10	10	10	214.50 ± 241.30	0.10 ± 0.12	0.14 ± 0.21	1.30	80.00
			20	20	174.15 ± 189.51	0.08 ± 0.10	0.10 ± 0.14	1.30	80.00
			50	50	117.42 ± 147.42	0.08 ± 0.08	0.09 ± 0.14	1.68	74.00
		20	10	10	255.70 ± 222.13	0.05 ± 0.04	0.15 ± 0.19	1.20	100.00
			20	20	219.95 ± 180.53	0.04 ± 0.03	0.13 ± 0.14	1.20	100.00
			50	50	139.32 ± 138.44	0.05 ± 0.03	0.10 ± 0.14	1.48	98.00

higher quality scores. However, such communities cannot be covered by any description using the given tags/topics and therefore do not provide the explanatory and descriptive properties of the resulting communities of the presented approach. Since COMODO detects statistically significant communities regarding their size and density – especially with the local modularity – and since it can be tuned easily by setting the minimal size threshold appropriately, typical problems and pathological cases such as small community sizes can be avoided.

6. Conclusions

In this paper, we have presented an approach for description-oriented community detection using exhaustive subgroup discovery. We presented the COMODO algorithm for the discovery of community patterns. Furthermore, we proposed suitable optimistic estimates for a range of standard community quality functions; the optimistic estimates are efficient to compute and enable an effective approach. Our proposed method ensures that the top-k communities (representable by a given set of describing features) are discovered; we apply an efficient branch-and-bound method with appropriate pruning techniques based on exhaustive subgroup discovery using optimistic estimates.

We evaluated the approach using five different data sets from three social systems namely, from the social bookmarking systems BibSonomy and delicious, and from the social music platform last.fm. In our evaluation, we focused on two aspects: The efficiency of the proposed optimistic estimates, and the validity of the obtained community patterns. The evaluation demonstrated the effectiveness of the proposed descriptive mining approach applying the presented optimistic estimates. The implemented pruning scheme makes the approach scalable for larger data sets, especially when the local modularity quality function is chosen to assess the communities' quality. Concerning the validity of the patterns, we focused on structural properties of the patterns and the subgraphs induced by the respective comunity patterns, and compared COMODO to three baseline community detection algorithms. Overall, the results indicate statistically valid and significant results that do not exhibit the typical problems and pathological cases such as small community sizes that are often encountered when using typical community mining methods. Furthermore, COMODO is able to detect communities that are typically captured by shorter descriptions leading to a lower description complexity, compared to the baseline.

For future work, we aim to apply the proposed method on more diverse data sets. In addition, an interesting option is to include background knowledge, e. g., in the form of topic hierarchies in order consider more general or specific descriptions, cf. [12, 13]. Furthermore, we plan to extend the approach for community detection on dynamic networks.

Acknowledgements

This work has been partially supported by the VENUS research cluster at the Interdisciplinary Research Center for Information System Design (ITeG) at Kassel University, and by the Commune project funded by the Hertie foundation.

References

[1] M. Adnan, R. Alhajj, J. Rokne, Identifying Social Communities by Frequent Pattern Mining, in: Proc. 13th Intl. Conf. Information Visualisation, IEEE Computer Society, Washington, DC, USA, 2009, pp. 413–418.

[2] M. Atzmueller, Knowledge-Intensive Subgroup Mining – Techniques for Automatic and Interactive Discovery, Vol. 307 of Dissertations in Artificial Intelligence-Infix (Diski), IOS Press, 2007.

[3] M. Atzmueller, Data Mining on Social Interaction Networks, Journal of Data Mining and Digital Humanities 1 (2014).

[4] M. Atzmueller, Analyzing and Grounding Social Interaction in Online and Offline Networks, in: Proc. ECML/PKDD, Vol. 8726 of LNCS, Springer Verlag, Berlin, 2014, pp. 485–488.

[5] M. Atzmueller, Subgroup Discovery – Advanced Review, WIREs: Data Mining and Knowledge Discovery 5 (1) (2015) 35–49. doi:10.1002/widm.1144.

[6] M. Atzmueller, S. Doerfel, A. Hotho, F. Mitzlaff, G. Stumme, Face-to-Face Contacts at a Conference: Dynamics of Communities and Roles, in: Modeling and Mining Ubiquitous Social Media, Vol. 7472 of LNAI, Springer Verlag, Berlin, 2012.

[7] M. Atzmueller, F. Lemmerich, Fast Subgroup Discovery for Continuous Target Concepts, in: Proc. International Symposium on Methodologies for Intelligent Systems, Vol. 5722 of LNCS, Springer, Berlin, 2009, pp. 1–15.

[8] M. Atzmueller, F. Lemmerich, B. Krause, A. Hotho, Who are the Spammers? Understandable Local Patterns for Concept Description, in: Proc. 7th Conference on Computer Methods and Systems, 2009.

[9] M. Atzmueller, F. Mitzlaff, Towards Mining Descriptive Community Patterns, in: Workshop on Mining Patterns and Subgroups, Leiden, The Netherlands, 2010.

[10] M. Atzmueller, F. Mitzlaff, Efficient Descriptive Community Mining, in: Proc. 24th International FLAIRS Conference, AAAI Press, Palo Alto, CA, USA, 2011, pp. 459 – 464.

[11] M. Atzmueller, F. Puppe, A Case-Based Approach for Characterization and Analysis of Subgroup Patterns, Journal of Applied Intelligence 28 (3) (2008) 210–221.

[12] M. Atzmueller, F. Puppe, H.-P. Buscher, Towards Knowledge-Intensive Subgroup Discovery, in: Proc. LWA 2004, Germany, 2004, pp. 117–123.

[13] M. Atzmueller, F. Puppe, H.-P. Buscher, Exploiting Background Knowledge for Knowledge-Intensive Subgroup Discovery, in: Proc. IJCAI, Edinburgh, Scotland, 2005, pp. 647–652.

[14] R. Bayardo, R. Agrawal, D. Gunopulos, Constraint-Based Rule Mining in Large, Dense Databases, Data Mining and Knowledge Discovery 4 (2000) 217–240, 10.1023/A:1009895914772.

[15] D. Benz, A. Hotho, R. Jäschke, B. Krause, F. Mitzlaff, C. Schmitz, G. Stumme, The Social Bookmark and Publication Management System BibSonomy, VLDB 19 (2010) 849 – 875.

[16] I. Cantador, P. Brusilovsky, T. Kuflik, 2nd Workshop on Information Heterogeneity and Fusion in Recommender Systems (HetRec), in: Proc. 5th ACM Conference on Recommender Systems, ACM, New York, NY, USA, 2011.

[17] S. Fortunato, Community Detection in Graphs, Physics Reports 486 (3-5) (2010) 75 – 174.

[18] S. Fortunato, C. Castellano, Encyclopedia of Complexity and System Science, Springer, Berlin, 2007, Ch. Community Structure in Graphs.

[19] L. Freeman, Segregation In Social Networks, Sociological Methods & Research 6 (4) (1978) 411.

[20] J. Fürnkranz, A. J. Knobbe, Guest Editorial: Global Modeling using Local Patterns, Data Mining and Knowledge Discovery 21 (2010) 1–8.

[21] E. Galbrun, A. Gionis, N. Tatti, Overlapping Community Detection in Labeled Graphs, Data Min. Knowl. Discov. 28 (5-6) (2014) 1586–1610.

[22] U. Gargi, W. Lu, V. S. Mirrokni, S. Yoon, Large-Scale Community Detection on YouTube for Topic Discovery and Exploration, in: Proc. 5th International Conference on Weblogs and Social Media, The AAAI Press, Palo Alto, CA, USA, 2011, pp. 486–489.

[23] M. Girvan, M. E. J. Newman, Community Structure in Social and Biological Networks, PNAS 99 (12) (2002) 7821–7826.

[24] S. Gregory, Finding Overlapping Communities in Networks by Label Propagation , New J. Phys. (12).

[25] H. Grosskreutz, S. Rüping, S. Wrobel, Tight Optimistic Estimates for Fast Subgroup Discovery, in: Proc. ECML/PKDD, Vol. 5211 of LNCS, Springer Verlag, Berlin, 2008, pp. 440–456.

[26] S. Günnemann, I. Färber, B. Boden, T. Seidl, GAMer: A Synthesis of Subspace Clustering and Dense Subgraph Mining, in: Knowledge and Information Systems (KAIS), Springer, 2013.

[27] J. Han, J. Pei, Y. Yin, Mining Frequent Patterns Without Candidate Generation, in: W. Chen, J. Naughton, P. A. Bernstein (Eds.), 2000 ACM SIGMOD Intl. Conference on Management of Data, ACM Press, 2000, pp. 1–12.

[28] W. Klösgen, Explora: A Multipattern and Multistrategy Discovery Assistant, in: Advances in Knowledge Discovery and Data Mining, AAAI Press, 1996, pp. 249–271.

[29] A. J. Knobbe, E. K. Y. Ho, Pattern Teams, in: Proc. 10th European Conference on Principle and Practice of Knowledge Discovery in Databases, PKDD, Springer-Verlag, Berlin, Heidelberg, 2006, pp. 577–584.

[30] M. Koyuturk, W. Szpankowski, A. Grama, Assessing Significance of Connectivity and Conservation in Protein Interaction Networks, Journal of Computational Biology 14 (6) (2007) 747–764.

[31] J. M. Kumpula, M. Kivelä, K. Kaski, J. Saramäki, Sequential Algorithm for Fast Clique Percolation, Physical Review E 78 (2) (2008) 026109.

[32] A. Lancichinetti, S. Fortunato, J. Kertész, Detecting the Overlapping and Hierarchical Community Structure in Complex Networks, New Journal of Physics 11 (3).

[33] N. Lavrac, B. Kavsek, P. Flach, L. Todorovski, Subgroup Discovery with CN2-SD, Journal of Machine Learning Research 5 (2004) 153–188.

[34] E. A. Leicht, M. E. J. Newman, Community Structure in Directed Networks, Phys. Rev. Lett. 100 (11) (2008) 118703.

[35] D. Leman, A. Feelders, A. Knobbe, Exceptional Model Mining, in: Proc. ECML/PKDD, Vol. 5212 of Lecture Notes in Computer Science, Springer, 2008, pp. 1–16.

[36] F. Lemmerich, M. Becker, M. Atzmueller, Generic Pattern Trees for Exhaustive Exceptional Model Mining, in: Proc. ECML/PKDD, Vol. 7524 of LNCS, Springer Verlag, Berlin, 2012, pp. 277–292.

[37] F. Lemmerich, M. Rohlfs, M. Atzmueller, Fast Discovery of Relevant Subgroup Patterns, in: Proc. 23rd International FLAIRS Conference, AAAI Press, Palo Alto, CA, USA, 2010, pp. 428–433.

[38] J. Leskovec, K. J. Lang, A. Dasgupta, M. W. Mahoney, Community Structure in Large Networks: Natural Cluster Sizes and the Absence of Large Well-Defined Clusters, CoRR abs/0810.1355.

[39] J. Leskovec, K. J. Lang, M. Mahoney, Empirical Comparison of Algorithms for Network Community Detection, in: Proc. 19th International Conference on World Wide Web, ACM, New York, NY, USA, 2010, pp. 631–640.

[40] D. Li, B. He, Y. Ding, J. Tang, C. Sugimoto, Z. Qin, E. Yan, J. Li, T. Dong, Community-based Topic Modeling for Social Tagging, in: Proc. 19th ACM International Conference on Information and Knowledge Management, CIKM, ACM, New York, NY, USA, 2010, pp. 1565–1568.

[41] K. H. Lim, A. Datta, A Topological Approach for Detecting Twitter Communities with Common Interests, in: M. Atzmueller, A. Chin, D. Helic, A. Hotho (Eds.), Ubiquitous Social Media Analysis, Vol. 8329 of Lecture Notes in Computer Science, Springer Verlag, Berlin, 2013, pp. 23–43.

[42] Y.-R. Lin, Y. Chi, S. Zhu, H. Sundaram, B. L. Tseng, FacetNet: A Framework for Analyzing Communities and Their Evolutions in Dynamic Networks, in: Proc. 17th International Conference on World Wide Web, ACM, New York, NY, USA, 2008, pp. 685–694.

[43] Y.-R. Lin, Y. Chi, S. Zhu, H. Sundaram, B. L. Tseng, Analyzing Communities and Their Evolutions in Dynamic Social Networks, ACM Trans. Knowl. Discov. Data 3 (2009) 8:1–8:31.

[44] Y.-R. Lin, J. Sun, H. Sundaram, A. Kelliher, P. Castro, R. Konuru, Community Discovery via Metagraph Factorization, ACM Trans. Knowl. Discov. Data 5 (2011) 17:1–17:44.

[45] A. McDaid, N. Hurley, Detecting Highly Overlapping Communities with Model-Based Overlapping Seed Expansion, in: Proc. International Conference on Advances in Social Networks Analysis and Mining, IEEE Computer Society, Washington, DC, USA, 2010, pp. 112–119.

[46] F. Mitzlaff, M. Atzmueller, A. Hotho, G. Stumme, The Social Distributional Hypothesis, Journal of Social Network Analysis and Mining 4 (216).

[47] F. Moser, R. Colak, A. Rafiey, M. Ester, Mining Cohesive Patterns from Graphs with Feature Vectors, in: SDM, Vol. 9, SIAM, 2009, pp. 593–604.

[48] S. Muff, F. Rao, A. Caflisch, Local Modularity Measure for Network Clusterizations, Physical Review E (Statistical, Nonlinear, and Soft Matter Physics) 72 (5) (2005) 056107.

[49] M. E. Newman, M. Girvan, Finding and Evaluating Community Structure in Networks, Phys Rev E Stat Nonlin Soft Matter Phys 69 (2) (2004) 1–15.

[50] M. E. J. Newman, Detecting Community Structure in Networks, Europ Physical J 38.

[51] M. E. J. Newman, Modularity and Community Structure in Networks, Proceedings of the National Academy of Sciences 103 (23) (2006) 8577–8582.

[52] V. Nicosia, G. Mangioni, V. Carchiolo, M. Malgeri, Extending the Definition of Modularity to Directed Graphs with Overlapping Communities, J. Stat. Mech. (2009) 03024.

[53] G. Palla, I. Derényi, I. Farkas, T. Vicsek, Uncovering the Overlapping Community Structure of Complex Networks in Nature and Society, Nature 435 (7043) (2005) 814–818.

[54] G. Palla, I. J. Farkas, P. Pollner, I. Derenyi, T. Vicsek, Directed Network Modules, New Journal of Physics 9 (6) (2007) 186.

[55] S. Pool, F. Bonchi, M. van Leeuwen, Description-driven Community Detection, Transactions on Intelligent Systems and Technology 5 (2).

[56] U. Raghavan, A. R., S. Kumara, Near Linear Time Algorithm to Detect Community Structures in Large-Scale Networks, Phys Rev E 76:036106.

[57] S. E. Schaeffer, Graph Clustering, Comput. Sci. Rev. 1 (1) (2007) 27 – 64.

[58] J. Sese, M. Seki, M. Fukuzaki, Mining Networks with Shared Items, in: Proc. 19th ACM International Conference on Information and Knowledge Management, ACM, New York, NY, USA, 2010, pp. 1681–1684.

[59] A. Silva, W. Meira Jr, M. J. Zaki, Mining Attribute-Structure Correlated Patterns in Large Attributed Graphs, Proc. VLDB Endowment 5 (5) (2012) 466–477.

[60] C. Tsourakakis, F. Bonchi, A. Gionis, F. Gullo, M. Tsiarli, Denser Than the Densest Subgraph: Extracting Optimal Quasi-cliques with Quality Guarantees, in: Proc. SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, New York, NY, USA, 2013, pp. 104–112.

[61] S. Wasserman, K. Faust, Social Network Analysis: Methods and Applications, 1st Edition, no. 8 in Structural analysis in the social sciences, Cambridge University Press, 1994.

[62] S. Wrobel, An Algorithm for Multi-Relational Discovery of Subgroups, in: Proc. 1st European Symposium on Principles of Data Mining and Knowledge Discovery (PKDD-97), Springer Verlag, Berlin, 1997, pp. 78–87.

[63] J. Xie, M. Chen, B. K. Szymanski, LabelRankT: Incremental Community Detection in Dynamic Networks via Label Propagation, in: Proc. Workshop on Dynamic Networks Management and Mining, DyNetMM '13, ACM, New York, NY, USA, 2013, pp. 25–32.

[64] J. Xie, S. Kelley, B. K. Szymanski, Overlapping Community Detection in Networks: The State-of-the-art and Comparative Study, ACM Comput. Surv. 45 (4) (2013) 43:1–43:35.

[65] J. Xie, B. K. Szymanski, LabelRank: A Stabilized Label Propagation Algorithm for Community Detection in Networks, in: Proc. IEEE Network Science Workshop, West Point, NY, 2013.

[66] J. Yang, J. Leskovec, Defining and Evaluating Network Communities Based on Ground-truth, in: Proc. ACM SIGKDD Workshop on Mining Data Semantics, MDS '12, ACM, New York, NY, USA, 2012, pp. 3:1–3:8.