Towards Mining Descriptive Community Patterns

Martin Atzmueller and **Folke Mitzlaff**
Knowledge and Data Engineering Group,
University of Kassel, Germany
{atzmueller,mitzlaff}@cs.uni-kassel.de

Abstract

Community mining or community detection methods are usually applied in order to identify groups of users which share, e.g., common interests or expertise. This paper presents an approach for mining descriptive patterns in order to characterize communities in terms of their distinctive features. The method discovers communities and their descriptions directly, based on adapted subgroup discovery techniques. We describe the adaptation in detail and propose optimistic estimates of standard community evaluation metrics. We present first results of the proposed approach using data from the real-world social bookmarking system BibSonomy.

1 Introduction

In social and ubiquitous applications a wealth of information can be utilized for improving the user experience of the system, e.g., by providing recommendations for specific resources or contacts. In this context, a peer group, or community of users similar to the targeted user is often a helpful resource. In order to identify communities, *community mining* and *community detection* methods are applied. A community is intuitively defined as a set of nodes that has more and/or better links between its members compared to the rest of the network. Formally, communities can be defined using certain criteria, for example, edge counts within a community compared to the edge counts to nodes located outside the community, cf. [Leskovec *et al.*, 2010].

There are a lot of prominent methods for community detection, e.g., [Newman and Girvan, 2004; Newman, 2004; Fortunato and Castellano, 2007], that identify a collection of (overlapping) groups corresponding to communities. In order to obtain such communities, e.g., groups of users characterized by the interest in *web mining*, *computer* and *java*, appropriate description techniques need to be applied. We can, for example, consider the most frequent features of the respective communities, or mine sets of subgroups characterizing certain communities, cf., [Atzmueller *et al.*, 2009]. In contrast to such indirect approaches, this paper proposes an approach for mining descriptive community patterns directly: We present techniques for obtaining the local patterns describing the k-best (overlapping) communities according to standard community evaluation measures. Furthermore, compared to standard community mining approaches in the field of social network analysis, cf. [de Nooy *et al.*, 2005], we aim to discover interesting groups in the "community space" by efficient exhaustive search for patterns in the "description space".

The proposed method is based on an adapted subgroup discovery approach, since descriptive community mining and subgroup discovery share the property of optimizing for groups that are interesting with respect to a certain property of interest: For subgroup discovery, the property of interest is usually given by a (dependent) target variable. With respect to the target, appropriate quality measures are then applied, e.g., for obtaining the set of subgroups that are "as large as possible and have the most unusual statistical characteristic with respect to the property of interest" [Wrobel, 1997]. For community mining, a community is considered interesting based on the links between its members, e.g., comparing the intra community-links to the inter-community links. The proposed method applies subgroup discovery techniques for obtaining subgroups described by a set of features, e.g., considering combinations of tags or topics for social bookmarking systems. The description are rated in the community space using standard community evaluation measures directly, e.g., by considering users of social bookmarking systems.

We propose an algorithm for mining descriptive community patterns based on the SD-Map* algorithm and describe the adaptation to the community mining task in detail. Furthermore, we discuss the application of standard community evaluation measures, and propose suitable optimistic estimates for pruning the search space.

Our application context is given by social and ubiquitous applications such as social networking applications, social bookmarking systems, and sensor-networks. Considering our own system BibSonomy[1][Benz *et al.*, 2010] as an example, the friend graph indicates explicit friendship relations between users. Then, these graphs directly indicate communities (of users) according to the link structure. Similar interaction networks accrue in the context of ubiquitous applications (e. g., users which are using a given service at the same place and time). Communities of users can then be characterized in terms of their descriptive features, e. g., for generating explanations [Atzmueller and Roth-Berghofer, 2010]. In the context of social bookmarking systems, e. g., we can consider the set of tags, topics, or resources that the respective subset of users applied.

The rest of the paper is structured as follows: Section 2 summarizes basics of community detection and subgroup discovery. Next, Section 3 discusses related work. We introduce the proposed approach for mining descriptive community patterns in Section 4. After that, we provide evaluation results in the context of the real-world BibSonomy system in Section 5. Finally, Section 6 concludes the paper with a summary and directions for future research.

[1] http://www.bibsonomy.org

In: Lorentz Workshop on Mining Patterns and Subgroups, Lorentz Center, Leiden, The Netherlands, 2010.
Awarded with the Best Paper/Best Discovery Award

2 Preliminaries

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

2.1 Graphs

A *graph* $G = (V, E)$ is an ordered pair, consisting of a finite set V which consists of the *vertices* or *nodes*, and a set E of *edges*, which are two element subsets of V. A *directed graph* is defined accordingly: E denotes a subset of $V \times V$. For simplicity, we write $(u, v) \in E$ in both cases for an edge belonging to E and freely use the term *network* as a synonym for a graph. The *degree* of a node in a network measures the number of connections it has to other nodes. For the *adjacency matrix* $A \in \mathbb{R}^{n \times n}$ of a set of nodes S with $n = |S|$ contained in a graph $G = (V, E)$ holds $A_{ij} = 1$ iff $(i, j) \in E$ for any nodes i, j in S (assuming some bijective mapping from $1, \ldots, n$ to S). We identify a graph with its according adjacency matrix where appropriate.

2.2 Community Quality Measures

The concept of a *community* can be intuitively defined as a group $\mathcal{C}$ of individuals out of a population $\mathcal{U}$ such that members of $\mathcal{C}$ are densely "related" one to each other but sparsely "related" to individuals in $\mathcal{U} \setminus \mathcal{C}$. This concept transfers to vertex sets $C \subseteq V$ in graphs $G = (V, E)$ where nodes in C are densely connected but sparsely connected to nodes in $V \setminus C$. Though defined in terms of graph theory, the community concept remains vague. For a given graph $G = (V, E)$ and a community $C \subseteq V$ we set $n := |V|$, $m := |E|$, $n_C := |C|$, $m_C := |\{(u, v) \in E \mid u, v \in C\}|$, $\overline{m}_C := |\{(u, v) \in E \mid u \in C, v \notin C\}|$ and for a node $u \in V$ its degree is denoted by $d(u)$. Different evaluation functions (also called cluster indices) $f \colon \mathcal{P}(V) \to \mathbb{R}$ for modelling the intuitive community concept exist, e. g., [Leskovec *et al.*, 2008].

- **Conductance:** $f(C) = \frac{\overline{m}_C}{2m_C + \overline{m}_C}$
- **Expansion:** $f(C) = \frac{\overline{m}_C}{n_C}$
- **Average-ODF:** $\frac{1}{n_C} \sum_{u \in C} \frac{|\{(u,v) \in E \colon v \notin C\}|}{d(u)}$
- **Volume:** $\sum_{u \in C} d(u)$
- **Edges cut:** $\overline{m}_C$

Another popular quality function is given by the modularity [Newman and Girvan, 2004] which is based on comparing the number of edges within a community with the expected such number given a null-model (i.e., a randomized model). Thus, the modularity of a community clustering is defined to be the fraction of the edges that fall within the given clusters minus the expected such fraction if edges were distributed at random. This can be formalized as follows: The modularity $MOD(S)$ of a set of nodes S and its assigned adjacency matrix $A \in \mathbb{N}^{n \times n}$ is given by

$$MOD(S) = \frac{1}{2m} \sum_{i,j} \big(A_{i,j} - \frac{d(i)d(j)}{2m}\big)\delta(C_i, C_j)\,,$$

where C_i is the cluster to which node i belongs and C_j is the cluster to which node j belongs; $d(i)$ and $d(j)$ denote i and j's degrees respectively; $\delta(C_i, C_j)$ is the *Kronecker delta* symbol that equals 1 iff $C_i = C_j$, and 0 otherwise. The modularity for a single community C can then be computed as:

$$MOD(C) = \frac{1}{2m} \sum_{i \in C, j \in C} \big(A_{i,j} - \frac{d(i)d(j)}{2m}\big)\,.$$

In the context of this paper, we focus on the conductance and the modularity quality functions. These consider the evaluation from two different perspectives. Modularity mainly focuses on the links *within* communities, while the conductance also takes the links *between* communities into account. In Section 4.1 we present a technique for transforming and merging networks and descriptive data, e.g., tags or topic from social bookmarking systems, into a single data source representing a special undirected annotated graph of the respective network. Concerning the proposed method for descriptive community mining we focus on functions for undirected graphs, that are used for discovering local communities. However, for their evaluation we need to consider overlapping communities. Therefore, we summarize a generalization of the modularity for overlapping communities introduced by [Nicosia *et al.*, 2009] in Section 5.

2.3 Networks in Social Bookmarking Systems

Social bookmarking systems do not explicitly contain relations on users in their underlying data structure, which only captures who assigned which tags to which resource. But most bookmarking systems incorporate additional relations on users such as "*my network*" in del.icio.us[2] and "*friends*" in BibSonomy[3] and flickr[4]. Each such network is connected with a given functionality, e. g., for restricting access to certain resources or for allowing messages to be sent. Nevertheless, those networks also bear a "social meaning".

Besides those explicit relations among users, different relations are established implicitly by user interactions with the systems. These are given by, e. g., clicklogs or page visit information. In some systems, it is also possible to copy content from other users. Then, the logging information can be transformed into a user-graph structure, for example, into a click-graph, into a visit-graph, or into a copy-graph of users, as described below in detail. All of these are implemented in the social resource sharing system BibSonomy, but are typically also found in other resource sharing and social applications. Even more user interactions occur in the context of ubiquitous web applications. Examples are users which are using a given service at the same place and time, or communication relationships based on proximity sensors [Szomszor *et al.*, 2010], among many others. In the following, we summarize three of the evidence networks that are provided by the BibSonomy system, and refer to [Mitzlaff *et al.*, 2010b] for more details.

- The *Friend-Graph* $G_F = (V_F, E_F)$ is a directed graph with $(u, v) \in E_F$ iff user u has added user v as a friend.
- The *Click-Graph* $G_C = (V_C, E_C)$ 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.
- The *Visit-Graph* $G_V = (V_V, E_V)$ is a directed graph with $(u, v) \in E_V$ iff user u navigated to the user page of user v.

For more details, we refer to [Mitzlaff *et al.*, 2010a], in which we have also shown that the discussed evidence networks are a suitable data source for community mining and community detection algorithms.

[2] `http://delicious.com/network/<username>`
[3] `http://www.bibsonomy.org/friends`
[4] `http://www.flickr.com/photos/friends/`

2.4 Subgroup Discovery

Subgroup discovery [Klösgen, 1996; Wrobel, 1997; Atzmueller, 2007] aims at uncovering properties of a selected target population of individuals: The interesting subgroups should have the most unusual characteristics with respect to a given target property of interest. For some basic notation, let Ω_A denote the set of all attributes. For each attribute $a \in \Omega_A$ a range $dom(a)$ of values is defined. Let DB be the database containing all available data records. A data record $r \in DB$ is given by the n-tuple $r = ((a_1 = v_1), \ldots, (a_n = v_n))$ of $n = |\Omega_A|$ attribute values, $v_i \in dom(a_i)$ for each a_i. A subgroup s is defined as a subset of the whole database DB, i.e., $s \subseteq DB$. The subgroup description language specifies the individuals belonging to the subgroup. For a commonly applied single-relational propositional language a subgroup description can be defined as follows:

Subgroup Description A subgroup description $sd(s)$ of the subgroup s, $sd(s) = \{e_1, \ldots, e_l\}, l \geq 0$, is defined by the conjunction of a set of selection expressions (selectors). The individual selectors $e_i = (a_i, V_i)$ are selections on domains of attributes, $a_i \in \Omega_A, V_i \subseteq dom(a_i)$. We define Ω_E as the set of all selection expressions and Ω_{sd} as the set of all possible subgroup descriptions.

Subgroup A subgroup s described by the subgroup description $sd(s)$ is given by all records $r \in DB$ covered by the subgroup description $sd(s)$. We denote the subgroup s described by $sd(s)$ with $ext(sd(s))$. A subgroup s' is called a *refinement* of s, if $sd(s) \subset sd(s')$.

Due to multi-correlations between the independent variables, the discovered subgroups can overlap significantly. Adapting the idea of *condensed representations*, for example, closed itemsets, cf., [Pasquier *et al.*, 1999], we can define closed subgroup descriptions using the subgroup size n. If only the closed subgroup descriptions are considered, then for equivalent and equally-sized subgroups, the subgroup with the longest subgroup description is selected.

Closed Subgroup Description A subgroup description $sd \in S$ is called *closed* with respect to a set S if there exists no subgroup description $sd' \in S, sd' \supset sd$, with $|ext(sd)| = |ext(sd')|$.

Quality Function A quality function $q : 2^{DB} \rightarrow R$ assigns a numeric interestingness value to the subgroup s.

A quality function measures the interestingness of a subgroup: Typical quality criteria include the characteristics of the target concept comparing the subgroup and the whole database, and the subgroup size. Often, a threshold $\mathcal{T}_n$ imposes a minimal size constraint on the subgroup.

For many quality functions an *optimistic estimate* of a subgroup s can be specified. This approximation describes an upper bound for the quality, that any refinement of s can have. The basic principle of optimistic estimates is to prune parts of the search space during exhaustive search for the top k subgroups. The idea relies on the intuition that if the k best subgroups so far have already been obtained, and the optimistic estimate of the current subgroup is below the quality of the worst subgroup contained in the k best, then the current branch of the search tree can be safely pruned.

More formally, an optimistic estimate oe of a quality function q is a function such that $s' \subseteq s \rightarrow oe(s) \geq q(s')$, i.e., that no refinement of subgroup s can exceed the quality $oe(s)$.

3 Related Work

Fortunato [Fortunato and Castellano, 2007] discusses various aspects connected to the concept of community structure in graphs. Basic definitions as well as existing and new methods for community detection are presented. This work is a good entry point for the topic of community mining.

An LDA [Blei *et al.*, 2003] based community detection method for a folksonomy is presented in [Kashoob *et al.*, 2010] which is evaluated indirectly by measuring the improvement of search results achieved by incorporating the mined community information. Using a metric which is purely based on the structure of graphs, Newman presents algorithms for finding communities and assessing community structure in graphs [Newman, 2004]. A thorough empirical analysis of the impact of different community mining algorithms and their corresponding objective function on the resulting community structures is presented by Leskovec [Leskovec *et al.*, 2010], which is based on the size resolved analysis of community structure in graphs as presented in [Leskovec *et al.*, 2008].

In [Lancichinetti and Fortunato, 2009], Lancichinetti presents a thorough comparison of many different state of the art graph community detection algorithms. The performance of algorithms are compared relative to a class of adequately generated artificial benchmark graphs. The work is extended in [Lancichinetti and Fortunato, 2009] for directed graphs and overlapping communities. Additionally, [Lancichinetti *et al.*, 2009] discusses an approach for detecting overlapping communities based on the local optimization of a fitness criterion. Gregory [Gregory, 2009] describes several algorithms for overlapping community detection that consider non-overlapping (hierarchical) clusters first, and subsequently merge these into an overlapping structure. Finally, [Chen *et al.*, 2010] describe a game-theoretic approach for detecting overlapping communities that models community assignments as a strategic game of agents corresponding to the nodes. It optimizes community assignments until an equilibrium state of the utility functions of the agents is reached.

In contrast to the approaches mentioned above, the proposed method integrates the information from both the network and other descriptive information, e.g., tags or topics describing the nodes contained in the network. The presented method focuses on the characterization and description of communities while directly searching for the top k descriptive communities according to their quality. The proposed method guarantees that the top k communities are discovered, that can be represented using the given description space. It ensures this by efficient exhaustive search in description space using appropriate pruning techniques.

A first approach for the characterization and description of communities was introduced in [Atzmueller *et al.*, 2009], focussing on the description of spammers in the social bookmarking system BibSonomy. This technique shares similarities with the iterative approach in [van Leeuwen, 2010], that also discovers subgroups first and then describes these using a (set of) subgroup description(s). Also, interactive approaches for subgroup characterization are presented in [Atzmueller and Puppe, 2008]. In contrast to these approaches, the method proposed in this paper does not start with a given subgroup allocation, but discovers and optimizes subgroups/communities directly. This can provide more compact conjunctive descriptions, i.e., not including disjunctions of several subgroup descriptions for the characterization of a community.

4 Mining Descriptive Community Patterns

In an intuitive sense, community mining is concerned with the identification of subgroups of users that are more densely connected within each other, than to other groups. In social bookmarking systems, we can consider subgroups of users for which their connections are defined, for example, in terms of a friend or a click-network. Therefore, in this context, subgroups and communities are rather similar; from now on, we will use the terms interchangeably whenever we are referring to communities (or subgroups) of users, either represented by a set of edges or nodes contained in a graph or dataset, respectively.

For the characterization of the communities, we can apply descriptive data of the users, for example, considering their applied set of tags, their set of topics, or their set of resources, see Table 1 for some examples. Given a community, subgroup discovery can be applied for obtaining characteristic descriptions, e.g., [Atzmueller *et al.*, 2009] for characterizing the community of spammers in the social bookmarking system BibSonomy. In this way, different patterns for certain community subsets are provided.

In contrast, this paper provides a method for discovering communities and their descriptions directly. Specifically, we will show how we can adapt an efficient subgroup discovery method for community detection, and apply typical community evaluation functions. The output of the mining method is the set of the top k communities according to the given community evaluation function. The method can produce overlapping communities, which do not restrict one user to a single community, but allows for a description of users participating in a set of communities, i.e., a characterization from different points of view.

4.1 Transformation and Mining: Subgroup Discovery for Community Description

In the following, we consider two data sources for obtaining descriptive community patterns. A database DB containing data records that describes a set U of users in terms of certain attributes Ω_A, e.g., containing binary features corresponding to topics or tags applied by that user. Additionally, we consider links between the users corresponding to the data records of DB modeled in a graph G, e.g., friendship links, follower links, or according page visits.

Our goal is then to discover the k best communities as subgroups of the whole database described by a description considering the attributes of database DB, that maximize a suitable community evaluation function on the link structure G. For applying subgroup discovery efficiently we need to merge both data sources.

Therefore, we apply a data transformation approach for merging both data sets. Considering the descriptive database and the user network, it is easy to see that these only consider individual nodes, i.e., users of the graph, and the database, respectively. However, the applicable community evaluation measures mostly consider the edges, i.e., the connections between the nodes in order to assess the community qualities. If we consider the edges of the community, for which each edge connects two nodes, and if we can also access the degree information between these nodes, then we can reconstruct and calculate the appropriate quality measures directly, as we will outline below.

When merging the data sets, we obtain a data set containing the connecting edges between the contained nodes that are constructed in a special way for enabling direct descriptive community mining: Each data record represents a connecting edge between two nodes of the network. The attribute values of each such data record are then given by the intersection of the (non-default) attribute values of each node that is connected by the corresponding edge. For example, considering tags corresponding to binary attributes we only consider the *true* values of each attribute, e.g., indicating that a tag or a topic was applied by both users represented by the given nodes. The rationale behind using the intersection is based on the observation, that an edge (and its two nodes) can only contribute to a community described by a certain attribute value, if this respective attribute value is contained in the data records of the two nodes.

In addition to the attribute values contained in the intersection of the two source nodes, the edge data record also stores the contributing nodes and their respective degrees. In this way, typical quality functions such as modularity and the conductance can be calculated considering a set of edges given only this information.

Thus, only using the number of edges contained in the community m_C, the total number of edges, and the respective node degrees $d(i)$ of the nodes $i \in C$ of the community, the modularity for a community C can be directly computed as follows:

$$\begin{aligned} MOD(C) &= \frac{1}{2m} \sum_{i \in C, j \in C} \left(A_{i,j} - \frac{d(i)d(j)}{2m}\right) = \\ &= \frac{1}{2m} \sum_{i \in C, j \in C} A_{i,j} - \sum_{i \in C, j \in C} \frac{d(i)d(j)}{4m^2} = \\ &= \frac{1}{2m} 2m_C - \sum_{i \in C, j \in C} \frac{d(i)d(j)}{4m^2} = \\ &= \frac{m_C}{m} - \sum_{i \in C, j \in C} \frac{d(i)d(j)}{4m^2} \end{aligned}$$

Conductance can similarly be calculated using only the parameters mentioned above:

$$\begin{aligned} CON(C) &= \frac{\overline{m}_C}{2m_C + \overline{m}_C} = \\ &= \frac{\overline{m}_C}{\sum_{u \in C} d(u)} = \\ &= \frac{\sum_{u \in C} d(u) - 2m_C}{\sum_{u \in C} d(u)} = \\ &= 1 - \frac{2m_C}{\sum_{u \in C} d(u)} \end{aligned}$$

For subgroup discovery, we are interested in maximizing the given quality function, which works well for the modularity while conductance is closer to zero for communities with higher quality. Therefore, from now on we will consider the *inverse conductance* ($ICON$) instead of the conductance, for maximizing the quality values.

$$\begin{aligned} ICON(C) &= 1 - CON(C) = \\ &= \frac{2m_C}{\sum_{u \in C} d(u)} \end{aligned}$$

4.2 Optimistic Estimates for Community Quality Functions

In the following we introduce optimistic estimates for typical community evaluation functions, i.e., for the introduced inverse conductance and for the modularity.

Conductance			Modularity		
Community$_1$	Community$_2$	Community$_3$	Community$_1$	Community$_2$	Community$_3$
work *flickr* *delicious*	*business* *production* *sales*	*work* *flickr* *delicious*	*work* *flickr* *delicious*	*business* *computer* *production*	*php* *web* *internet*
		university *bib* *surabaya*			*innovation* *business* *forschung*
		php *web* *internet*			
		library *all* *emulation*			

Table 1: Example for descriptive community patterns: Three of the top 10 ranked subgroups/communities according to conductance and modularity together with their respective topic description, using the friend-graph data described in Section 5. The columns show the different communities, consisting of several topics as sets of tags in the rows of the table.

Modularity

An optimistic estimate for the *modularity* can be derived based on the number of edges m_C within the community:

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

Proof We start with a reformulation of the modularity. An optimistic estimate can then be derived considering the number of edges m_C within the community. Also, note that $\sum_{i \in C} d(i) = 2m_C + \overline{m}_C$, considering the degrees $d(i)$ of the nodes i contained in a community C.

$$\begin{aligned} MOD(C) &= \frac{m_C}{m} - \sum_{i \in C, j \in C} \frac{d(i)d(j)}{4m^2} = \\ &= \frac{m_C}{m} - \frac{1}{4m^2} \sum_{i \in C} d(i) \sum_{j \in C} d(j) = \\ &= \frac{m_C}{m} - \frac{1}{4m^2} \sum_{i \in C} d(i)(2m_C + \overline{m}_C) = \\ &= \frac{m_C}{m} - \frac{1}{4m^2}(2m_C + \overline{m}_C)^2 \leq \\ &\leq \frac{m_C}{m} - \frac{1}{4m^2}(2m_C)^2 = \frac{m_C}{m} - \frac{m_C^2}{m^2} = \\ &= \hat{oe}(MOD(C)). \end{aligned}$$

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 $\hat{oe}(MOD(C))$ is a concave function since its derivative function

$$\hat{oe}(MOD(C))' = \frac{1}{m} - \frac{2m_C}{m^2}$$

is monotonically decreasing. Therefore, the function has one maximum, at point $\frac{m}{2}$, 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 $m_C < \frac{m}{2}$, $\hat{oe}(MOD(C))$ is decreasing in m_C, and thus $\hat{oe}(MOD(C))$ is an optimistic estimate for $MOD(C)$. This concludes the proof. □

Inverse Conductance

For the *inverse conductance*, we need to consider the minimal support threshold $\mathcal{T}_n$ w.r.t. the community size (number of nodes) when computing the optimistic estimate:

$$oe(ICON(C)) \quad = \quad 1 - \frac{\sum_{i=1}^{\mathcal{T}_n} \overline{d(i)}}{\sum_{u \in C} d(u)}$$

where $\overline{d(i)}$ are the outgoing degrees of the nodes contained in the community C, sorted in ascending order, such that $\overline{d(i)}, i = 1 \ldots \mathcal{T}_n$ denotes the minimal $\mathcal{T}_n$ outgoing degrees of connected nodes contained in the community C.

Proof

$$\begin{aligned} ICON(C) &= \frac{2m_C}{\sum_{u \in C} d(u)} = \\ &= \frac{\sum_{u \in C} d(u) - \overline{m}_C}{\sum_{u \in C} d(u)} = \\ &= 1 - \frac{\overline{m}_C}{\sum_{u \in C} d(u)} \leq \\ &\leq 1 - \frac{\sum_{i=1}^{\mathcal{T}_n} \overline{d(i)}}{\sum_{u \in C} d(u)} \\ &= oe(ICON(C)). \end{aligned}$$

As shown above, for a fixed m_C it follows that $oe(ICON(C)) \geq ICON(C)$. Since every subset $C' \subseteq C$ will cover at most m_C edges and the numerator of the last term ($\sum_{i=1}^{\mathcal{T}_n} \overline{d(i)}$) is the minimum considering the outgoing edges for a minimal size of $\mathcal{T}_n$, $oe(ICON(C))$ is an optimistic estimate of $ICON(C)$. □

The optimistic estimate can be efficiently computed by traversing the set of nodes and collecting the outgoing node count for each node considering the endpoints of the edges.

4.3 Algorithmic Issues

In the following, we describe the SD-MAC algorithm for mining descriptive community patterns adapted from the state-of-the-art SD-Map* [Atzmueller and Lemmerich, 2009] algorithm for subgroup discovery. Specifically, we will show how the SD-Map* algorithm and its basic data structure, the FP-Tree, can be applied for the community mining scenario.

As SD-Map*, SD-MAC is based on the efficient FP-growth [Han *et al.*, 2000] algorithm for mining frequent patterns, first applied in [Atzmueller and Puppe, 2006] for subgroup discovery. As a special data structure, the frequent pattern tree or FP-tree is used which is implemented as an extended prefix-tree-structure that stores count information about the frequent patterns. SD-Map* applies a divide and conquer method, first mining subgroup described by one selector and then recursively mining larger descriptions. SD-Map* utilizes a frequent pattern tree (FP-tree), i.e., an extended prefix-tree-structure that stores the relevant parameters for estimating subgroup qualities.

The FP-tree contains the frequent FP-nodes in a header table, and links to all occurrences of the frequent selectors in the FP-tree structure. This data structure itself can be regarded as a compressed data representation for the set of instances. According to the prefix-tree principle, the tree stores aggregated counts for each shared path corresponding to the attribute–value pairs of a set of instances.

For the recursive step, a conditional FP-tree is constructed, given the conditional pattern base of a frequent selector (FP-node). The conditional pattern base consists of all the prefix paths of such a FP-node. Due to the limited space we refer to Han et al. [Han *et al.*, 2000] for more details.

SD-Map* utilizes the FP-tree structure (built in two scans of the database) to efficiently compute quality functions for all subgroups relying on the fact that all the necessary information is locally available in the FP-tree structure. Therefore, for the SD-MAC algorithm for mining descriptive community patterns, we essentially need to store the appropriate information within the FP-nodes of the FP-Tree enabling the calculation of the community evaluation measures. Additionally, the SD-MAC algorithm can (optionally) output closed subgroup descriptions as an alternative to providing all community descriptions according to the analysis goals of the user. This feature can be implemented using the techniques described in [Wang *et al.*, 2005; Lemmerich and Atzmueller, 2009; Lemmerich *et al.*, 2010].

The SD-MAC algorithm utilizes the FP-tree structure for computing the qualities of subgroup patterns efficiently. All parameters that are needed for the quality evaluation are stored and obtained, respectively, from the individual FP-nodes of the tree. SD-MAC includes (optional) pruning strategies adapted from SD-Map* and utilizes quality functions with optimistic estimates for this purpose

For embedding optimistic estimate pruning, we basically only need to consider three options for pruning and reordering/sorting according to the current optimistic estimates: (1) **Pruning**: In the recursive step when building a conditional FP-tree, we omit a (conditioned) branch, if the optimistic estimate for the conditioning selector is below the threshold given by the k best subgroup qualities. (2) **Pruning**: When building a (conditional) frequent pattern tree, we can omit all the FP-nodes with an optimistic estimate below the mentioned quality threshold. (3) **Reordering/Sorting**: During the iteration on the currently active selector queue when processing a (conditional) FP-tree, we can dynamically reorder the selectors that have not been evaluated so far by their optimistic estimate value. In this way, we evaluate the *more promising* selectors 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.

To efficiently compute the community evaluation functions together with their optimistic estimates for the community mining context SD-MAC stores additional information in the FP-nodes of the FP-Tree, depending on the used quality function. Each FP-node of the FP-Tree captures information about aggregated edge information concerning the data base DB and the respective network. For each node, we store the following information:

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

The result of the SD-MAC algorithm for mining descriptive community patterns is the set of the top k patterns according to the applied community evaluation function. In order to reduce the redundancy with respect to including irrelevant descriptions, SD-MAC can optionally filter the patterns, using the techniques described in [Wang *et al.*, 2005; Lemmerich and Atzmueller, 2009]. This adapts the idea of closed labeled data [Garriga *et al.*, 2008] to closed subgroup descriptions (with respect to the set of edges/nodes contained in the community). Specifically, the largest description (in terms of the selectors included in the description) for the same set of edges will then be returned.

The top k patterns discovered by SD-MAC directly correspond to different communities described by the respective patterns. Since the patterns can describe a set of overlapping community participants, i.e., nodes in the network, the set of community descriptions provides potentially overlapping community allocations. Then, each node v that participates in multiple communities $c_i \in C_v$ needs to be assigned a belonging factor $\beta_v^{c_i}$. For the presented approach, we apply a uniform belonging factor

$$\beta_v^{c_i} = \frac{1}{|C_v|} ,$$

for the evaluation. Otherwise each node v is assigned a belonging factor $\beta_v = 1$. However, also other assignments, for example, based on the size of the community or the strength of the connections of the node can be applied. All belonging factors of a node need to sum to 1, i.e.,

$$\sum_{c_i \in C_v} \beta_v^{c_i} = 1 .$$

5 Evaluation

In the following, we first describe the data used for the evaluation of the evidence networks. We used publicly available data from the social bookmark and resource sharing system BibSonomy. After that, we describe the characteristics of the applied evidence networks, and present the conducted experiments. We conclude with a detailed discussion of the experimental results.

5.1 Evaluation Data and Setting

Our primary resource is an anonymized dump of all public bookmark and publication posts until January 27, 2010, from which we extracted *explicit* and *implicit* relations, cf. Table 2 for an overview.

The dump consists of 175,521 tags, 5,579 users, 467,291 resources and 2,120,322 tag assignments. The BibSonomy dump also contains friendship relations modeled in BibSonomy concerning 700 users. Furthermore, we utilized the "*click log*" of BibSonomy, consisting of entries which are generated whenever a logged-in user clicked on a link in BibSonomy. A log entry contains the URL of the currently visited page together with the corresponding link target, the date and the user name[5]. For our experiments we considered all click log entries until January 25, 2010. Starting in October 9, 2008, this dataset consists of 1,788,867 click events.

We finally considered all available apache web server log files, ranging from October 14, 2007 to January 25, 2010. The file consists of around 16 GB compressed log entries. We used all log entries available, ignoring the different time periods, as this is a typical scenario for real-world applications.

5.2 Applied Evaluation Measures

For the experiments we basically applied the evaluation measures *conductance* and *modularity* as introduced in Section 2.

For *directed networks* [Leicht and Newman, 2008] with in- and out- degree $d(i)^{\text{in}}$ and $d(j)^{\text{out}}$ for nodes i and j respectively the modularity becomes

$$MOD(S) = \frac{1}{m}\sum_{i,j}\left(A_{i,j} - \frac{d(i)^{\text{in}}d(j)^{\text{out}}}{m}\right)\delta(C_i, C_j)\,.$$

Since we consider methods for detecting both disjoint and overlapping communities, we need an extension of the modularity for overlapping communities. For overlapping communities a generalization of the modularity was proposed in [Nicosia *et al.*, 2009] that also includes the case of directed networks. For a set of nodes S, the modularity $MOD_o(S)$ for overlapping communities becomes:

$$MOD_o(S) = \frac{1}{m}\sum_{c\in C}\sum_{i,j\in V}\left(\beta_{l(i,j),c}A_{i,j}\frac{\beta^{\text{out}}_{l(i,j),c}k_i^{\text{out}}\beta^{\text{in}}_{l(i,j),c}k_j^{\text{in}}}{m}\right),$$

where k_i^{in} and k_j^{out} are the in and out degrees for nodes i, j; $\beta_{l(i,j),c}$ is the belonging coefficient of $l(i,j)$ for community c; $\beta^{\text{in}}_{l(i,j),c}$ is the expected belonging coefficient of any edge $l(i,j)$ pointing to a node going into community c, and $\beta^{\text{out}}_{l(i,j),c}$ is the expected belonging coefficient of any possible edge $l(i,j)$ starting from a node going into community c. The belonging coefficient of an edge is derived by combining the belonging coefficients of two nodes, each being part of a set of communities.

In our experiments, we multiplied the belonging factors of the respective nodes for obtaining the belonging factor of an edge, however also other combining functions, e.g., the maximum or the minimum of both factors are possible. For more details, we refer to [Nicosia *et al.*, 2009].

[5]Note: For privacy reasons a user may deactivate this feature!

	G_V (Visit)	G_C (Click)	G_F (Friend)
$\lvert V_i\rvert$	3381	1151	700
$\lvert E_i\rvert$	8214	1718	1012
$\lvert V_i\rvert/\lvert U\rvert$	0.58	0.20	0.12

Table 2: High level statistics for all relations where U denotes the set of all users in BibSonomy.

5.3 Experiments

For an initial assessment of the community mining performance we used standard approaches that can be broadly parameterized. First experiments were conducted using the well known *k-means algorithm* [MacQueen, 1967] for comparison. For that, each user u is represented by a vector $(u_1, \ldots, u_T) \in \mathbb{R}^T$ where T is the total number of tags and u_i is the total number of times user u assigned the tag i to resources in BibSonomy ($i = 1, \ldots, T$). The resulting clusters had poor quality, assigning most users to a single cluster. Due to the sparsity of the considered high dimensional vector space representation (there are more than $170,000$ tags), the underlying search for nearest neighbors failed (cf., e. g., [Beyer *et al.*, 1999] for a discussion).

In addition, using conjunctive community descriptions is also very difficult using the whole set of tags since the respective data is rather sparse. Furthermore, there are several issues when utilizing the (raw) set of tags directly, e.g., relating to many synonyms, writing variations, and hierarchical dependencies between tags that need to be handled appropriately in order to get more meaningful results. Therefore, to bypass these problem, we created topics capturing a set of tags, thus both reducing the number of dimensions and increasing the quality of the topics by grouping similar tags. There are a variety of approaches for dimensionality reduction. We chose to cluster the tags for building "*topics*", consisting of associated sets of tags. A user u is thus represented as a vector $\vec{u} \in \mathbb{R}^{T'}$ in the topic vector space, where $T' \ll T$ is the number of topics. For our experiments, we used a *latent dirichlet allocation* [Blei *et al.*, 2003] method for building topics, which efficiently builds interpretable tag clusters, i.e., for obtaining descriptive topic sets: Each topic captures semantically similar tags and thus helps to inhibit the problem of synonyms, semantic hierarchies, etc. The method has been successfully applied in similar contexts to tagging systems (cf. [Siersdorfer and Sizov, 2009]). We applied datasets containing 100 (LDA-100) and 500 (LDA-500) topics each for the user – tag/topic relations.

Furthermore, since k-means only tackles disjoint clusters, and the proposed descriptive community mining approach can discover overlapping communities, we selected another method for comparison. The well-known *EM* algorithm [Dempster *et al.*, 1977] also works with categorical data and assigns (overlapping) cluster assignments to the individual data records due to its expectation maximization strategy. As a baseline for the SD-MAC the EM algorithm is thus applied as a complement to the k-means algorithm thus handling both disjoint and overlapping communities.

In the following results, the k-means models are labeled with "*KM-Ln-Kk*", where n denotes the number of topics and k the number of clusters; the communities detected using the EM algorithm are labeled accordingly by "*EM-Ln-Kk*", and the communities discovered using the SD-MAC algorithm are denoted by "*SD-Ln-Kk*", where k denotes the number of subgroups, given by k closed subgroup descriptions.

For inhibiting subgroups with a low support, a minimal size threshold of $\mathcal{T}_n = 5$ was applied, so each community consists of at least 5 nodes. For the clustering methods no such measures were applied. For the experiments, we utilized the WEKA implementation [6] of the k-means and the EM algorithm.

[6]http://www.cs.waikato.ac.nz/~ml/weka/

method	inverse conductance (mean +/- stddev) G_F	G_C	G_V
KM-L100-K25	0.33 +/- 0.34	0.31 +/- 0.30	0.23 +/- 0.32
KM-L100-K50	0.28 +/- 0.36	0.27 +/- 0.33	0.18 +/- 0.30
KM-L100-K100	0.38 +/- 0.43	0.35 +/- 0.42	0.15 +/- 0.30
KM-L500-K25	0.41 +/- 0.40	0.35 +/- 0.40	0.23 +/- 0.35
KM-L500-K50	0.37 +/- 0.43	0.35 +/- 0.43	0.14 +/- 0.30
KM-L500-K100	0.45 +/- 0.46	0.35 +/- 0.43	0.17 +/- 0.34
EM-L100-K25	0.13 +/- 0.21	0.12 +/- 0.20	0.12 +/- 0.31
EM-L100-K50	0.09 +/- 0.21	0.08 +/- 0.20	0.05 +/- 0.15
EM-L100-K100	0.24 +/- 0.39	0.13 +/- 0.12	0.10 +/- 0.27
EM-L500-K25	0.33 +/- 0.40	0.32 +/- 0.40	0.17 +/- 0.32
EM-L500-K50	0.25 +/- 0.38	0.24 +/- 0.14	0.18 +/- 0.35
EM-L500-K100	0.29 +/- 0.42	0.23 +/- 0.39	0.13 +/- 0.31
SD-L100-K25	0.46 +/- 0.01	0.38 +/- 0.01	**0.29** +/- 0.02
SD-L100-K50	0.45 +/- 0.02	0.37 +/- 0.01	0.27 +/- 0.02
SD-L100-K100	0.44 +/- 0.02	0.36 +/- 0.02	0.26 +/- 0.02
SD-L500-K25	**0.49** +/- 0.01	**0.40** +/- 0.02	0.20 +/- 0.01
SD-L500-K50	0.48 +/- 0.01	0.38 +/- 0.03	0.20 +/- 0.01
SD-L500-K100	0.47 +/- 0.02	0.37 +/- 0.02	0.20 +/- 0.01

Table 3: Results – Inverse Conductance: k-means vs. EM and SD-MAC.

5.4 Results and Discussion

The results of the experiments for the click, visit, and friend graphs described in Section 2.3 are shown in Table 3 for the inverse conductance quality function.

The results in the table indicate, that the SD-MAC algorithm consistently discovers sets of communities with a higher inverse conductance (and thus a lower conductance value) than the k-means and the EM algorithms. For both, the relative performance for the different data sets is similar. Please note, that the standard deviations differ significantly from those of the SD-MAC results since the minimum and maximum values for conductance also differ significantly: Conductance favors smaller communities, and since the applied k-Means and EM algorithms do not provide a minimum size threshold higher inverse conductance values can be achieved, in comparison to the SD-MAC algorithm that applied a minimal size threshold $\mathcal{T}_n = 5$.

For comparing the modularity concerning k-means and the SD-MAC algorithm we consider the means of the modularity values of the individual communities. The results are shown in Table 4. The SD-MAC results consistently show higher modularity means, implying that the qualities of the individual local communities are consistently higher than the communities discovered by the k-means algorithm. Furthermore, for comparing the overlapping communities discovered using the EM algorithm and the SD-MAC algorithm we consider the evaluation results presented in Table 5. We observe, that the SD-MAC algorithm provides results that are at least as good as the results of the EM algorithm indicating that the SD-MAC algorithm provides meaningful communities concerning the modularity measure for overlapping communities.

We are aware that comparing the different algorithms concerning the modularity is rather difficult since we need to consider disjoint and overlapping communities, for which a global quality score is calculated based on the local ones. Assessing overlapping communities in graphs recently gained more attention, but the impact of allowing *overlapping* communities still needs to be thoroughly examined. This is similar to the issue of redundancy management in subgroup mining, for which redundant (overlapping) subgroups are removed.

method	modularity G_F	G_C	G_V
EM-L100-K25	0.22	0.18	**0.14**
EM-L100-K50	0.17	0.14	0.10
EM-L100-K100	0.21	0.10	0.08
EM-L500-K25	0.24	0.18	0.13
EM-L500-K50	0.22	0.13	0.09
EM-L500-K100	0.20	0.11	0.07
SD-L100-K25	0.23	0.04	0.04
SD-L100-K50	0.17	0.04	0.02
SD-L100-K100	0.17	0.12	0.06
SD-L500-K25	**0.26**	0.21	**0.14**
SD-L500-K50	0.20	**0.25**	0.11
SD-L500-K100	0.17	0.17	0.09

Table 5: Overlapping modularity: EM vs. SD-MAC

During our experiments, we could directly observe the pruning potential provided by the proposed optimistic estimates. The drastic reduction of the search space is shown in Table 6, exemplarily for the optimistic estimate for the modularity. The table shows the considered steps/hypotheses during the mining process, comparing the optimistic estimate (with no depth restriction) to applying search at maximum depths 3 and 5 with no pruning.

It is important to note, that the proposed method integrates the information from both the network and other descriptive information, e.g., tags or topics describing the nodes contained in the network. The presented method thus focuses on the characterization and description of communities while directly searching for the top k descriptive communities according to their quality. It is easy to see, that common methods from the field of social network analysis [de Nooy *et al.*, 2005] that work directly on the network structure can theoretically obtain higher quality scores for specific communities, since they do not need to consider the describing features: There can be high quality communities that cannot be covered by any description using the given tags/topics. Then, these cannot be discovered by any method that is restricted to the applied description space. The proposed method guarantees that the top k com-

method	local disjoint modularity (mean +/- stddev) G_F	G_C	G_V
KM-L100-K25	0.006 +/- 0.013	0.003 +/- 0.005	0.002 +/- 0.003
KM-L100-K50	0.003 +/- 0.009	0.001 +/- 0.003	0.001 +/- 0.002
KM-L100-K100	0.001 +/- 0.006	0.001 +/- 0.001	0.001 +/- 0.001
KM-L500-K25	0.006 +/- 0.015	0.003 +/- 0.010	0.001 +/- 0.004
KM-L500-K50	0.002 +/- 0.010	0.001 +/- 0.004	0.001 +/- 0.003
KM-L500-K100	0.002 +/- 0.007	0.001 +/- 0.003	0.001 +/- 0.001
SD-L100-K25	0.050 +/- 0.004	0.034 +/- 0.002	0.018 +/- 0.001
SD-L100-K50	0.047 +/- 0.004	0.032 +/- 0.003	0.017 +/- 0.002
SD-L100-K100	0.043 +/- 0.005	0.028 +/- 0.004	0.015 +/- 0.002
SD-L500-K25	0.026 +/- 0.002	0.015 +/- 0.001	0.008 +/- 0.001
SD-L500-K50	0.024 +/- 0.003	0.014 +/- 0.001	0.008 +/- 0.001
SD-L500-K100	0.022 +/- 0.003	0.013 +/- 0.001	0.007 +/- 0.001

Table 4: Results – Local Modularity (Mean): k-means vs. SD-MAC.

munities are discovered which can be represented using the given describing features. As demonstrated by the results, it ensures this by efficient exhaustive search using appropriate pruning techniques for standard evaluation metrics.

In summary, the presented results indicate, that the proposed approach outperforms the benchmark algorithms concerning the conductance. Furthermore, the comparison using the modularity measure indicates, that the SD-MAC algorithm consistently yields local communities with an inherent high quality, confirmed by the comparison with the baseline given by standard clustering algorithms.

6 Conclusions

In this paper, we have presented an approach for mining descriptive community patterns using subgroup discovery. We have described how to adapt subgroup discovery to the community mining setting, and we have proposed the SD-MAC algorithm for the efficient mining of descriptive community patterns based on the SD-Map* algorithm. Furthermore, we have presented optimistic estimates for typical community evaluation functions in this context. The presented approach was evaluated in an experimental setting using data from the social bookmarking system BibSonomy for which we presented initial results.

For future work, we aim to apply the proposed method on more (and more diverse) evidence networks, and to analyze and compare further community quality functions. Additionally, we plan to apply more refined methods for dimensionality reduction on the tag data in order to further improve the performance of the presented approach.

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. We thank the reviewers for their interesting comments, and Stephan Doerfel for helpful discussions.

References

[Atzmueller and Lemmerich, 2009] Martin Atzmueller and Florian Lemmerich. Fast subgroup discovery for continuous target concepts. In *Proc. ISMIS*. Springer Verlag, 2009.

[Atzmueller and Puppe, 2006] Martin Atzmueller and Frank Puppe. SD-Map – A Fast Algorithm for Exhaustive Subgroup Discovery. In *Proc.PKDD*, number 4213 in LNAI, pages 6–17, Berlin, 2006. Springer Verlag.

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

[Atzmueller and Roth-Berghofer, 2010] Martin Atzmueller and Thomas Roth-Berghofer. The Mining and Analysis Continuum of Explaining Uncovered. In *Proc. SGAI Intl. Conference on Artifical Intelligence*, 2010.

[Atzmueller *et al.*, 2009] Martin Atzmueller, Florian Lemmerich, Beate Krause, and Andreas Hotho. Who are the Spammers? Understandable Local Patterns for Concept Description. In *Proc. 7th CMS Conference*, 2009.

[Atzmueller, 2007] Martin Atzmueller. *Knowledge-Intensive Subgroup Mining – Techniques for Automatic and Interactive Discovery*, volume 307 of *DISKI*. IOS Press, March 2007.

[Benz *et al.*, 2010] Dominik Benz, Andreas Hotho, Robert Jäschke, Beate Krause, Folke Mitzlaff, Christoph Schmitz, and Gerd Stumme. The Social Bookmark and Publication Management System BibSonomy – A Platform for Evaluating and Demonstrating Web 2.0 Research. *VLDB. Accepted.*, 2010.

[Beyer *et al.*, 1999] Kevin S. Beyer, Jonathan Goldstein, Raghu Ramakrishnan, and Uri Shaft. When Is "Nearest Neighbor" Meaningful? In *ICDT*, volume 1540 of *LNCS*, pages 217–235. Springer, 1999.

[Blei *et al.*, 2003] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[Chen *et al.*, 2010] Wei Chen, Zhenming Liu, Xiaorui Sun, and Yajun Wang. A Game-Theoretic Framework to Identify Overlapping Communities in Social Networks. *Data Min. Knowl. Discov.*, 21(2):224–240, 2010.

[de Nooy *et al.*, 2005] Wouter de Nooy, Andrej Mrvar, and Vladimir Batagelj. *Exploratory Social Network Analysis with Pajek*. Cambridge Univ. Press, Cambridge, 2005.

[Dempster *et al.*, 1977] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society*, 39(1):1–38, 1977.

[Fortunato and Castellano, 2007] Santo Fortunato and Claudio Castellano. *Encyclopedia of Complexity and System Science*, chapter Community Structure in Graphs. Springer, 2007.

	Search steps: Pruning vs. no-pruning with max-depth = 3; 5 for the different networks								
	G_F			G_C			G_V		
method	pruned	d=3	d=5	pruned	d=3	d=5	pruned	d=3	d=5
L100-K25	5094	166607	77369800	473	166750	79375495	3992	166750	79375495
L100-K50	33361	166607	77369800	1987	166750	79375495	25501	166750	79375495
L100-K100	321544	166607	77369800	239368	166750	79375495	476906	166750	79375495
L500-K25	522	387744	36430567	503	3809721	$>2.4 \cdot 10^{10}$	560	18821103	$>2.4 \cdot 10^{10}$
L500-K50	704	387744	36430567	533	3809721	$>2.4 \cdot 10^{10}$	693	18821103	$>2.4 \cdot 10^{10}$
L500-K100	1651	387744	36430567	607	3809721	$>2.4 \cdot 10^{10}$	1221	18821103	$>2.4 \cdot 10^{10}$

Table 6: Impact of optimistic estimate pruning for modularity.

[Garriga *et al.*, 2008] Gemma C. Garriga, Petra Kralj, and Nada Lavrač. Closed sets for labeled data. *J. Mach. Learn. Res.*, 9:559–580, 2008.

[Gregory, 2009] Steve Gregory. Finding Overlapping Communities Using Disjoint Community Detection Algorithms. In *Complex Networks*, volume 207 of *Studies in Comp. Intell.*, pages 47–61. Springer, Berlin, 2009.

[Han *et al.*, 2000] Jiawei Han, Jian Pei, and Yiwen Yin. Mining Frequent Patterns Without Candidate Generation. In *Proc. SIGMOD*, pages 1–12. ACM Press, 2000.

[Kashoob *et al.*, 2010] S. Kashoob, J. Caverlee, and K. Kamath. Community-Based Ranking of the Social Web. In *Proc. ACM HT*, 2010.

[Klösgen, 1996] Willi Klösgen. Explora: A Multipattern and Multistrategy Discovery Assistant. In *Advances in Knowledge Discovery and Data Mining*, pages 249–271. AAAI Press, 1996.

[Lancichinetti and Fortunato, 2009] Andrea Lancichinetti and Santo Fortunato. Community Detection Algorithms: A Comparative Analysis. *Phys. Rev. E*, 80, 2009.

[Lancichinetti and Fortunato, 2009] Andrea Lancichinetti and Santo Fortunato. Benchmarks for Testing Community Detection Algorithms on Directed and Weighted Graphs with Overlapping Communities. *Phys. Rev. E*, 80(1):016118, July 2009.

[Lancichinetti *et al.*, 2009] Andrea Lancichinetti, Santo Fortunato, and Janos Kertesz. Detecting the Overlapping and Hierarchical Community Structure of Complex Networks. *New Journal of Physics*, 11, 2009.

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

[Lemmerich and Atzmueller, 2009] Florian Lemmerich and Martin Atzmueller. Incorporating Exceptions: Efficient Mining of epsilon-Relevant Subgroup Patterns. In *Proc. LeGo-09: From Local Patterns to Global Models, Workshop at the ECML/PKDD 2009*, 2009.

[Lemmerich *et al.*, 2010] Florian Lemmerich, Mathias Rohlfs, and Martin Atzmueller. Fast Discovery of Relevant Subgroup Patterns. In *Proc. FLAIRS*, pages 428–433, Palo Alto, CA, USA, 2010. AAAI Press.

[Leskovec *et al.*, 2008] Jure Leskovec, Kevin J. Lang, Anirban Dasgupta, and Michael W. Mahoney. Community Structure in Large Networks: Natural Cluster Sizes and the Absence of Large Well-Defined Clusters, 2008.

[Leskovec *et al.*, 2010] Jure Leskovec, Kevin J. Lang, and Michael W. Mahoney. Empirical Comparison of Algorithms for Network Community Detection, 2010. cite arxiv:1004.3539.

[MacQueen, 1967] J. B. MacQueen. Some Methods for Classification and Analysis of MultiVariate Observations. In *Proc. 5th Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press, 1967.

[Mitzlaff *et al.*, 2010a] Folke Mitzlaff, Martin Atzmueller, Dominik Benz, Andreas Hotho, and Gerd Stumme. Community Assessment using Evidence Networks. In *Proc. Workshop on Mining Ubiquitous and Social Environments (MUSE2010)*, Barcelona, Spain, 2010.

[Mitzlaff *et al.*, 2010b] Folke Mitzlaff, Dominik Benz, Gerd Stumme, and Andreas Hotho. Visit Me, Click Me, Be My Friend: An Analysis of Evidence Networks of User Relationships in Bibsonomy. In *Proc ACM HT*, Toronto, Canada, 2010.

[Newman and Girvan, 2004] M E Newman and M Girvan. Finding and Evaluating Community Structure in Networks. *Phys Rev E Stat Nonlin Soft Matter Phys*, 69(2):026113.1–15, 2004.

[Newman, 2004] M. E. J. Newman. Detecting Community Structure in Networks. *Europ Physical J*, 38, 2004.

[Nicosia *et al.*, 2009] V. Nicosia, G. Mangioni, V. Carchiolo, and M. Malgeri. Extending the Definition of Modularity to Directed Graphs with Overlapping Communities. *J. Stat. Mech.*, 2009.

[Pasquier *et al.*, 1999] Nicolas Pasquier, Yves Bastide, Rafik Taouil, and Lotfi Lakhal. Discovering Frequent Closed Itemsets for Association Rules. In *Proc. ICDT*, volume 1540 of *LNCS*, pages 398–416. Springer, 1999.

[Siersdorfer and Sizov, 2009] Stefan Siersdorfer and Sergej Sizov. Social Recommender Systems for Web 2.0 Folksonomies. In *Proc. ACM HT*, pages 261–270, New York, NY, USA, 2009. ACM.

[Szomszor *et al.*, 2010] M. Szomszor, C. Cattuto, W. Van den Broeck, A. Barrat, and H. Alani. Semantics, sensors, and the social web: The live social semantics experiments. *The Semantic Web: Research and Applications*, pages 196–210, 2010.

[van Leeuwen, 2010] Matthijs van Leeuwen. Maximal Exceptions with Minimal Descriptions. *Data Min. Knowl. Discov.*, 21(2):259–276, 2010.

[Wang *et al.*, 2005] Jianyong Wang, Jiawei Han, Ying Lu, and Petre Tzvetkov. TFP: An Efficient Algorithm for Mining Top-K Frequent Closed Itemsets. *IEEE Trans. Knowl. Data Eng.*, 17(5):652–664, 2005.

[Wrobel, 1997] Stefan Wrobel. An Algorithm for Multi-Relational Discovery of Subgroups. In *Proc. Europ. Symp. on Principles of Data Mining and Knowledge Discovery*, pages 78–87, Berlin, 1997. Springer Verlag.