Leveraging Collective Intelligence through Community Detection in Tag Networks*

Symeon Papadopoulos
Informatics and Telematics Institute
P.O.Box 60361, 57001, Thermi
Thessaloniki, Greece
papadop@iti.gr

Yiannis Kompatsiaris
Informatics and Telematics Institute
P.O.Box 60361, 57001, Thermi
Thessaloniki, Greece
ikom@iti.gr

Athena Vakali
Department of Informatics
Aristotle University of Thessaloniki
54124 Thessaloniki, Greece
avakali@csd.auth.gr

ABSTRACT
The paper studies the problem of community detection in tag networks, i.e. networks consisting of associations between tags that are used within Social Tagging Systems (STS) to annotate online resources (e.g. bookmarks, pictures, videos, etc.). Community detection methods aim at uncovering densely connected groups of tags, which can reveal the topic structure emerging in the STS. In this way, community detection in tag networks leverages Collective Intelligence (CI), that is the intelligence that is accumulated as a result of the collective activities of masses of users.

Categories and Subject Descriptors
H.3.3 [**Information Search and Retrieval**]: Clustering; H.3.4 [**Systems and Software**]

General Terms
Algorithms, Experimentation

Keywords
collective intelligence, community detection, tag networks

1. INTRODUCTION
Web applications and web user behavior have undergone a significant transformation during the last years. Today's web applications are centered around their users: they seek their input in the form of content and at the same time they encourage them to react to existing content [1]. In addition, they promote social networking and viral activities among their users. In turn, web users tend to embrace this call for increased participation and interaction: they contribute new content (e.g. pictures, blog posts), they rate, express their opinion and comment on digital content (e.g. articles, videos) or real-world entities (e.g. products), they organize online content by tagging it and they participate in online communities.

As a result of this massive user participation in web applications, large amounts of user-generated data are collected. Combining the behavior, preferences and ideas of masses of users that are imprinted in this data can result into novel insights and knowledge [26]; this process is frequently denoted to as the *emergence of Collective Intelligence*. Although the term Collective Intelligence has been used in broader contexts (cf. [28] for an extensive discussion), in this paper we restrict the use of the term in the context of intelligent processing and interpretation of mass user-generated content and data (in the spirit of [1, 26]).

A particular Collective Intelligence application archetype is illustrated by web applications incorporating *social tagging* features, often denoted as Social Tagging Systems (STS). Such systems enable their users to upload digital resources (e.g. bookmarks, pictures, blog posts, etc.) and annotate them with tags (i.e. freely chosen keywords). It is customary to represent such systems by use of the *folksonomy* model [20, 16], i.e. a tripartite graph comprising the set of users U, resources R and tags T as nodes and their associations as edges. Analyzing the network[1] structure of folksonomies can provide valuable insights into the semantics that are attached to online content by masses of users.

The approach adopted by this paper for extracting Collective Intelligence from folksonomies is the use of *community detection* techniques on folksonomy-derived tag networks. Community detection involves the analysis

*CKCaR'09.

[1]The terms *network* and *graph* are considered equivalent in our discussion and are therefore used interchangeably throughout the paper.

of the network structure with the goal of identifying *communities*, i.e. groups of objects (which are represented as nodes in the network) that are more densely connected (on the network) to each other than with the rest of the objects. In this paper, we introduce a novel method for detecting tag communities and we demonstrate that its application on tag networks can reveal the topic and semantic structure that emerges as a result of the tagging activities of masses of users.

The paper is structured as follows: Section 2 reviews the most prominent research conducted in the context of community detection methods, as well as other approaches which have been adopted for leveraging Collective Intelligence from Social Tagging Systems. Section 3 specifies the process for deriving a tag network from a folksonomy. Section 4 introduces a new method for community detection which is particularly suited for the analysis of tag networks. This method is applied in two tag networks derived from real-world social tagging applications, namely the LYCOS iQ questions and answers system and the BibSonomy bookmark sharing application. The analysis results are discussed in Section 5. Finally, the paper contains a discussion on the major issues identified through this work (Section 6) and concludes in Section 7.

2. RELATED WORK

2.1 Community detection

Due to the abundance of literature on community detection, we will restrict our discussion to some selected works that we deem as pertinent to our study. An extensive survey article on community detection is provided in [11].

Perhaps the most influential work in this area is the method by Girvan and Newman [13]. According to this, network edges are removed resulting in the progressive fragmentation of the network (splitting into disconnected components). The order in which edges are removed depends on their betweenness centrality (the higher their centrality the sooner they are removed). Later, the same authors introduced the notion of *modularity* as a measure of the profoundness of community structure in a network [23]. This spawned a whole class of methods that attempted to detect community structure in a network by means of maximizing modularity. For instance, an agglomerative hierarchical clustering method is proposed in [21] and computationally refined in [7], which successively agglomerates pairs of communities (starting from single-node communities) such that each agglomeration results into the maximum possible modularity increase. More sophisticated techniques were presented that tackled the problem of modularity maximization by means of various techniques such as simulated annealing [19], extremal optimization [9] and spectral optimization [22].

Lately, the concept of modularity has gone through scrutiny leading to the conclusion that communities of small scales (smaller than some threshold that depends on the network size and the degree of their interconnectedness) are likely to remain undetected from modularity maximization methods [10]. Hence, methods integrating different notions of community-ness have been devised. For instance, the Clique Percolation Method (CPM) by Palla et al. [24] consider communities as sets of nodes that are reachable through a k-clique chain, i.e. a set of adjacent k-cliques (k-cliques sharing at least $k-1$ nodes). A similar notion is used by the SCAN algorithm [29], which devises the concept of *structure reachability* between nodes and defines communities as sets of nodes which are structure reachable from each other.

Finally, a set of methods of particular interest to our study are based on the notion of *seed-based community expansion*. According to this paradigm, the community detection process is seen as an expansion process, which, starting from a seed node, progressively attracts adjacent nodes with the goal of maximizing some local community-ness measure, e.g. local modularity [6], subgraph modularity [18] or node outwardness [2]. These methods are of particular importance to our study as will become apparent from Section 4 since they can be combined with community detection methods such as SCAN [29] to yield sophisticated community models that are valuable in the study of tag communities.

2.2 Analysis of social tagging systems

Tagging has attracted considerable research interest after the mainstream adoption of social bookmarking and resource sharing applications such as delicious[2], flickr[3] and BibSonomy. Formally, a tripartite hypergraph model has been established for the representation of users, resources and tags in an STS [20]. Some of the first research works in this area pertain to the statistical and dynamical properties of tagging [15, 14]. In [15], Halpin et al. note that tag-tag cooccurrence networks can be useful in revealing the topic structure shaped by the usage of the tagging system.

Other studies focused on the semantics emerging from the tagging activities of users. Hotho et al. [16] applied association rules mining in order to discover subsumption relations between tags. Further, they devised a variant of PageRank (named FolkRank) that enabled them to rank the entities (tags, users and resources) of a folksonomy. In addition, some research has targeted the problem of tag clustering [3, 12], which is similar to the problem of tag community detection. However, in the aforementioned works, tag clustering has been tackled by means of vector-based agglomerative hierarchi-

[2]http://delicious.com
[3]http://flickr.com

cal clustering (i.e. each tag is represented by a feature vector and pairwise distances are defined between tags) which is only applicable to tag sets of limited size. More sophisticated methods such as co-clustering have been applied to produce clusters of tags and resources [17], however these methods require extensive supervision for tasks such as reducing the number of tag features and parameter setting (e.g. number of clusters).

More recently, some preliminary results have been reported on the application of community detection methods on tagging systems [27, 4]. Simpson [27] employs a variant of the Girvan-Newman algorithm [13] to detect communities of tags in delicious and in a corporate bookmarking service, while Cattuto et al. [4] analyze the community structure in a subset of delicious tags by means of spectral dimensionality reduction.

3. DERIVING THE TAG NETWORK

We first consider an abstraction for STS-based applications. A commonly used formalism, under the name *folksonomy*, was introduced in [20] as a means to model the collective tagging activities of users within an STS:

Folksonomy: A representation of the entities involved in an STS in the form of a tripartite graph model with hyperedges. The set of the graph vertices is partitioned into three disjoint sets $U = \{u_1, ..., u_k\}$, $R = \{r_1, ..., r_l\}$, and $T = \{t_1, ..., t_m\}$ corresponding to the sets of users, resources and tags respectively. The folksonomy is defined as the set of tag assignments $A \subseteq U \times R \times T$ that users of the STS perform in order to annotate the content of interest to them. Alternatively, the folksonomy is denoted by the hypergraph $H(A) = \{V, E\}$ where $V = U \cup R \cup T$ and $E = \{\{u, r, t\} | (u, r, t) \in A\}$.

Since the tripartite hypergraph model is cumbersome to work with, it is preferable to transform it to three bipartite graphs [20], namely the association graphs between users and resources (UR), users and tags (UT) and resources and tags (RT); for instance, the weighted bipartite RT graph is defined as follows: $RT = \{R \times T, E_{rt}\}$, $E_{rt} = \{(r, t) | \exists u \in U : (u, r, t) \in E\}$, $w : E_{rt} \rightarrow \mathbb{N}, \forall e : (r, t) \in E_{rt}, w(e) := |\{u : (u, r, t) \in E\}|$.

Since we are interested in tag-tag association networks, we need to further transform the two bipartite graphs involving the set of tags T into a simple weighted tag graph. For instance, starting from the resource-tag association graph (RT), the resulting tag-tag association graph is defined as: $G_{RT}(T) = \{T, E_{tt}\}$, $E_{tt} = \{(t_i, t_j) \in T \times T | \exists r \in R : (r, t_i), (r, t_j) \in E_{rt}\}$, $w : E_{tt} \rightarrow \mathbb{N}, \forall e : (t_i, t_j) \in E_{tt}, w(e) := |\{r : (r, t_i), (r, t_j) \in E_{rt}\}|$.

In practice, when two tags are used to annotate the same resource a link is created between them in the G_{RT} tag network. In addition, the weight of this link is equal to the number of different resources where this pair of tags was used (co-occurrence frequency). This can be normalized by dividing it with the number of resources where either of the two tags was used. Thus, if tag t_i was used to annotate the set of resources R_i and tag t_j was used to annotate R_j, then the strength of the link between these tags is quantified by the Jaccard index of the two resource sets:

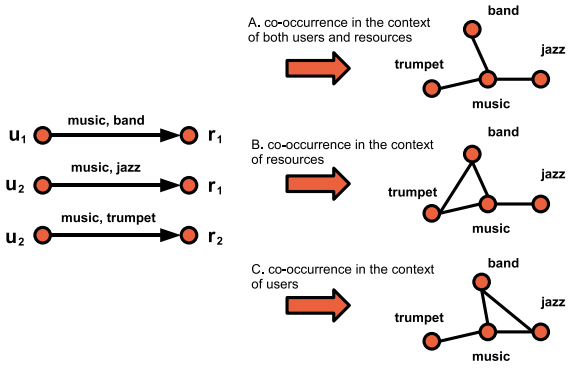


Figure 1: Three variants for deriving a tag-tag association network from a folksonomy.

$$w(e) = \frac{|R_i \cap Rj|}{|R_i \cup Rj|} \quad (1)$$

A more relaxed mode of establishing associations between tags is to consider associations either between tags that are used to annotate the same resource (independent of the user) or between tags used by the same user (independent of the resource). These three different variants for deriving a tag network from a folksonomy are illustrated in Figure 1. In our experiments we opted for the resource cooccurrence tag network creation (option B in Figure 1) due to the fact that associating tags based on user cooccurrence or based on cooccurrence (irrespective of whether it is user or resource) resulted into much denser tag networks, which renders the application of community detection techniques very challenging [11]. Furthermore, since a user may be interested in a variety of topics, associating tags based on user cooccurrence would create many spurious associations between tags that are by no means topically related to each other. In contrast, resources usually pertain to some specific topic or entity, therefore the use of two tags to annotate the same resource provides evidence in favor of their topical relatedness.

Alternatively, it would be possible to construct a tag network by considering pairwise similarities/distances between tags. For instance, Cattuto et al. [5] employ the cosine similarity between tags in the vector space $\mathbb{R}^m$ by use of the tag adjacency matrix of $G(T)$). The authors provide evidence that the cosine similarity is more appropriate for establishing equivalence, synonym

and subclass relations between tags compared to the plain co-occurrence similarity. However, in our community detection framework we are going to rely only on the co-occurrence similarity between tags for two reasons:

1. When a new tag assignment is recorded in the system, only a single co-occurrence update operation is necessary. In the case of cosine similarity, all pairwise similarity values would need to be recomputed. Therefore, the use of cosine similarity is prohibitive in a dynamic context.

2. Instead of using the cosine similarity, it is possible to draw similar conclusions about the relations between tags by means of graph-based similarity measures such as the *structural similarity* between two nodes [29].

4. TAG COMMUNITY DETECTION

The detection of communities in tag networks cannot be simply addressed by means of direct application of existing community detection methods due to the following reasons:

(a) Overlapping community structure. Tag communities are expected to overlap with each other since there are several entities or sub-topics that are shared between broader topics. For instance, high frequency tags are expected to participate in multiple tag communities.

(b) Tag-dependent network role. Most existing community detection methods treat network nodes as equivalent. In the case of tag networks, however, this may lead to poor results. For instance, tags that are used by many users tend to denote topics or categories which are connected to a large number of tags. These should be treated as "community bounds", i.e. local community expansion methods such [6, 18, 2] should stop the expansion process when encountering such tags in order to prevent topically irrelevant terms to spill-in to the current community.

(c) Dynamic nature of STS. Although community detection algorithms operate on static snapshots of networks, it would be necessary for a series of real-world tasks to have community detection methods that can operate in online mode. Thus, communnity detection methods that rely on the complete network structure, such as modularity maximization methods, would be hardly applicable in real settings.

Consequently, we introduce a hybrid technique for community detection, which takes into account the aforementioned particularities of tag networks. The proposed technique is based on two steps: (a) community seed set detection based on the notion of (μ, ϵ)-cores introduced in [29] and (b) local expansion of the identified cores to maximize the *subgraph modularity* introduced in [18] while respecting the *bridge bounding* constraint introduced in [25].

4.1 Community seed set detection

The community seed set detection step of our method is based on the concept of (μ, ϵ)-cores introduced in [29]. The definition of (μ, ϵ)-cores is based on the concepts of structural similarity and ϵ-neighborhood that we repeat here for convenience. We also repeat the definition of direct structure reachability.

Structural similarity: The structural similarity between two nodes v and w of a graph $G = \{V, E\}$ is defined as:

$$\sigma(v, w) = \frac{|\Gamma(v) \cap \Gamma(w)|}{\sqrt{|\Gamma(v)| \cdot |\Gamma(w)|}} \quad (2)$$

where $\Gamma(v)$ is the *structure* of node v and is defined as

$$\Gamma(v) = \{w \in V | (v, w) \in E\} \cup \{v\} \quad (3)$$

ϵ-neighborhood: This is the subset of a node's structure containing only those nodes that are at least ϵ-similar with the node; in math notation:

$$N_\epsilon(v) = \{w \in \Gamma(v) | \sigma(v, w) \geq \epsilon\} \quad (4)$$

(μ, ϵ)-core: A vertex v is called a (μ, ϵ)-core if its ϵ-neighborhood contains at least μ vertices; formally:

$$CORE_{\mu,\epsilon}(v) \Leftrightarrow |N_\epsilon(v)| \geq \mu \quad (5)$$

Direct structure reachability: A node is directly structure reachable from a (μ, ϵ)-core if it is at least ϵ-similar to it: $DirReach_{\mu,\epsilon}(v, w) \Leftrightarrow CORE_{\mu,\epsilon}(v) \wedge w \in N_\epsilon(v)$.

Once the (μ, ϵ)-cores of a network have been identified, it is possible to start attaching adjacent nodes to them provided that they are reachable through a chain of nodes which are directly structure reachable from each other. We call the resulting set of nodes as a *community seed set*. The aforementioned technique for collecting community seed sets is computationally efficient ($O(n)$ for a network of n edges) as discussed in [29].

One issue that is not addressed in [29] pertains to the selection of parameters μ and ϵ. Setting a high value for ϵ (the maximum possible value for ϵ is 1.0) will render the core detection step very eclectic, i.e. few (μ, ϵ)-cores will be detected. Moreover, higher values for μ will also result in the detection of fewer cores (for instance, all nodes with degree lower than μ will be excluded from the core selection process). For that reason, we

carry out the community seed set selection steps multiple times with different values of μ and ϵ and select parameter values that lead to seed sets with the following characteristics: (a) the number of community seed sets should be sufficiently large, (b) the seed set sizes should be relatively balanced.

4.2 Community expansion

Starting from a community seed set S, the second step in the proposed community detection method involves an expansion process, which aims at the maximization of subgraph modularity [18] and at the same time prevents the expansion process from crossing edges that act as "bridges" [25]. The modularity of a subgraph $S \in V$ is defined as:

$$M(S) = \frac{|\{(v,w) \in E | v,w \in S\}|}{|\{(v,w) \in E | v \in S \wedge w \in V - S\}|} \quad (6)$$

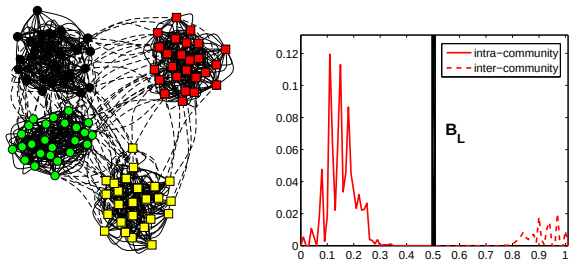
Also, we consider the local bridging function of an edge:

$$b_L((v,w)) = 1 - \frac{|N(v) \cap N(w)|}{min[(d(v)-1),(d(w)-1)]} \quad (7)$$

where $N(v)$ and $d(v)$ denote the set of nodes that are adjacent to v and the degree of v respectively. In order for $b_L((v,w))$ to have a low value, v and w need to have a lot of common neighbors (relative to their degree). Effectively, this means that in order to move from v to w, one has multiple options in addition to the link between them. Thus, (v,w) is considered as an intra- (or within-) community edge. In the opposite case, when the two endpoints of a bridge have very few or no neighbors in common, then this edge is crucial for the connection between its endpoints. For that reason, we consider in the latter case (high b_L value) that (v,w) is an inter-community edge or bridge.

In order to derive a decision threshold B_L for identifying the bridge edges of the network (Line 2 of Algorithm 1), one needs to inspect the distribution of b_L values among the edges of the graph. Figure 2 illustrates how the position of edges on a graph with community structure affects their local bridging values. The graph of Figure 2(a) was generated to comprise a synthetic four-community structure. Edges that link different communities with each other, i.e. *inter-community* edges, are drawn in dashed line. According to the distribution of Figure 2(b), these edges are characterized by high b_L values, therefore they can be separated by means of thresholding from the intra-community edges.

In brief, the proposed community expansion process successively attaches nodes to community S with the goal of maximizing $M(S)$ (Equation 6). The set of nodes that are considered as candidates for attachment to S are pooled from the "community frontier", i.e. the set of all nodes that are adjacent to at least one node of the community, under the condition that the edge connecting them is not a "bridge", i.e. its local bridging value (Equation 7) does not exceed a certain threshold. The expansion process is specified in Algorithm 1.



(a) Graph $G \equiv (V,E)$ (b) $b_L(E)$ distribution

Figure 2: Comparison between intra- versus inter-community edge local bridging. Edges drawn with dashed lines in Figure 2(a) are also the ones with the highest local bridging values.

Algorithm 1 LocalCommunityDetection

Require: Network $G = (V,E)$, community seed set $S \in V$, local bridging function b_L, subgraph modularity M

1: Set of community nodes: $C_S = S$, $M = M(C_S)$
2: Frontier set $F = \{v \in V | (s,v) \in E \wedge s \in C_S \wedge v \notin C_S \wedge b_L(s,v) \geq B_L\}$
3: **while** $|F| > 0$ **do** $\{F$ is non-empty$\}$
4: $\quad \forall c \in F$ compute $M_{new,c} = M(C_S \cup \{c\})$
5: $\quad$ Select c which resulted in maximum $M_{new,c}$
6: $\quad$ **if** $M_{new,c} > M$ **then**
7: $\quad\quad M = M_{new,c}$
8: $\quad\quad$ Update frontier set F (Line 2)
9: $\quad$ **else**
10: $\quad\quad$ Return C_S
11: $\quad$ **end if**
12: **end while**

5. CASE STUDIES

Here, we present two case studies, through which we attempt to exemplify the principles of tag community detection that we presented above. The case studies are based on two tag datasets, coming from the LYCOS iQ QA system and the BibSonomy publication sharing application. Table 1 provides an overview of the datasets.

LYCOS iQ is a collaborative QA system where people ask and answer questions on any topic. The application incorporates tagging functionalities, similar to the one used in typical STS such as delicious and flickr. Our dataset comes from the English version of the application (which is not operational anymore).

BibSonomy is a social bookmarking and publication sharing application. It aims to integrate the features

of bookmarking systems by offering users the ability to store and organize their bookmarks and publication entries. The BibSonomy dataset was made available through the ECML PKDD Discovery Challenge 2009[4]. We used the "Post-Core" version of the dataset.

In order to derive tag networks from the two folksonomies, we used technique "B" of Figure 1 (co-occurrence in the context of resources). The resulting networks consist of 26,758 nodes and 109,340 edges (LYCOS iQ) and 13,276 nodes and 262,683 edges (BibSonomy) respectively. Both networks are relatively sparse, but present significant difference in their density. We then carried out multiple runs of the two-step community detection process of Section 4 using different values for parameters μ, ϵ and B_L. Table 2 summarizes some of the community detection results we obtained for each dataset. Careful inspection of the community detection results leads us to the following conclusions:

1. There is a trade-off between the size and the number of the resulting communities. Increasing μ results into larger and fewer communities, while increasing ϵ leads to communities of smaller size.
2. The smaller the value of the bridging threshold (B_L), the less tags are attached to a community seed during the community expansion step of the algorithm.
3. Community coverage (i.e. the percentage of tags that are assigned to some community) is low which means that many of the network tags are not assigned to any community. The coverage can be increased at the expense of the topical coherence (i.e. the degree that same-community tags are associated to each other) by relaxing the communityness constraints (set low values for μ and ϵ, set a high value for B_L).

Tables 3 and 4 present a set of sample communities detected by the algorithm on the LYCOS iQ and the BibSonomy tag networks respectively. The tag groups directly connote to the reader some specific topic and the majority of the groups (not only the ones presented in the table) contain tags that are pertinent to their topic (based on our judgement).

6. DISCUSSION

The case studies presented above demonstrated the utility of the proposed tag community detection method. However, there are several limitations that one should consider before applying it to a new setting.

An important consideration pertains to the fact that some information loss takes place during the simplification of the graph structures through the aforementioned graph projections (tripartite hypergraph → bipartite graphs → undirected tag graphs). This can sometimes lead to "topic blending", i.e. to two or more different topics to be detected as belonging to the same community due to some polysemous tags that are connected to all of them. Such a multi-topic community is depicted in Figure 3. Apart from the two tags "fl studio" and "fruity loops", which refer to the digital music composition and mixing software *Fruity Loops*, the rest of the tags refer to real estate in the state of *Florida*. The reason for this blending is the polysemous tag "FL" which stands for both *Fruity Loops* and *Florida*.

It is possible that such failures could be prevented by exploiting the bipartite structure of the resource-tag network. In order for this to be possible, community detection techniques that operate directly on bipartite graphs need to be employed. An example of such a method is *BiTector* [8]. A comparative study between bipartite and unipartite community detection techniques (in the context of STS) would be necessary in order to quantify the impact that the graph projection (from bipartite tag-resource to unipartite tag-tag network) has on the quality of the identified communities.

An additional issue that should be taken into account pertains to the low community coverage, i.e. lack of community assignment for many of the tags. In the previous section, we empirically showed that coverage can be increased by appropriate tuning of the algorithm parameters; however, this comes at the cost of community topical coherence, i.e. several unrelated tags may be introduced in existing communities or "loose" communities (groups of tags which are not closely related to each other) may be detected. So far, we have not considered the use of text-based techniques (e.g. stemming, lexical similarity) or semantic resources (WordNet, DBpedia) for improving coverage without harming topical coherence.

Finally, a significant issue troubling the application of the proposed community detection algorithm arises from the need to manually set parameters. Although several insights were provided in the previous discussion regarding the effect of the different parameters on the obtained communities, in the future we need to investigate efficient ways to estimate appropriate parameters for our algorithm without the need to execute it multiple times.

7. CONCLUSIONS

In this paper we presented an approach towards Collective Intelligence extraction from folksonomies using community detection techniques on folksonomy-derived tag networks. We addressed the major issues involved in the application of community detection in tag networks, namely the derivation oftag association networks from folksonomies and the detection of tag communities by means of a hybrid scheme which is based on

[4]http://www.kde.cs.uni-kassel.de/ws/dc09

Table 1: Folksonomies used in our case studies.

source	users	resources	tags	tag assignments
LYCOS iQ	22,177	62,497	26,758	134,601
BibSonomy	1,185	64,120	13,276	253,615

Table 2: Summary of community detection results. The columns correspond to the following: algorithm parameters (μ, ϵ, B_L), number of communities (C), mean community size (M_{all}), number and percentage of tags assigned to communities ($Coverage$ #, %), number of nodes belonging to (> 1) communities (O), mean and st. deviation of community seed and expansion set sizes (M_{core}, M_{exp}).

μ	ϵ	B_L	C	M_{all}	$Coverage$		O	M_{core}		M_{exp}	
					#	(%)		AVG	STD	AVG	STD
LYCOS iQ											
3	0.7	0.3	417	6.1	2529	9.5	32	4.9	2.1	1.3	2.5
3	0.8	0.1	227	5.6	1274	4.8	2	4.8	1.8	0.8	2.1
5	0.5	0.1	252	10.0	2510	9.4	13	8.6	9.5	1.4	3.2
10	0.5	0.3	21	29.52	607	2.3	13	18.5	16.9	11.1	16.6
BibSonomy											
5	0.7	0.3	178	15.7	2640	19.9	157	11.1	8.9	4.6	9.8
8	0.8	0.1	53	16.3	834	6.3	29	14.4	7.4	1.9	1.9
10	0.8	0.1	38	17.8	674	5.1	1	16.3	7.7	1.5	2.4
15	0.7	0.1	23	28.7	619	4.7	42	24.5	16.9	4.3	8.1

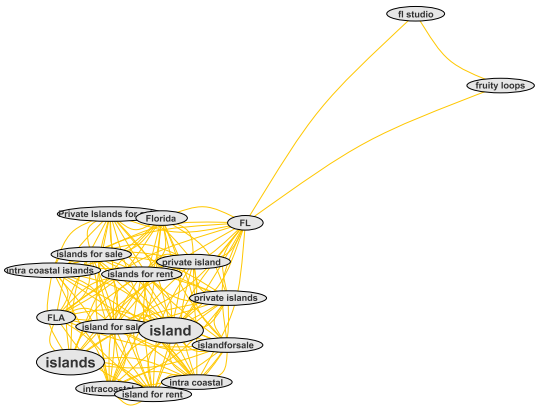


Figure 3: An example of "topic blending".

a community-seed selection and local seed expansion step. Finally, we applied the proposed methodology on two web applications that incorporate the folksonomy paradigm, namely LYCOS iQ and BibSonomy. One should note that the community detection technique introduced here is also applicable to relational data of other kinds (e.g. user and web resource networks).

In the future, we would like to integrate the proposed community detection process in a tag recommendation application. In addition, we plan to investigate ways to deal with the limitations of the proposed method, namely manual parameter setting and direct application to bipartite networks. Finally, we are interested in extending our community detection method to cope with two additional aspects of communities: (a) hierarchical structure and (b) temporal evolution.

8. ACKNOWLEDGMENTS

This work was supported by the WeKnowIt project, partially funded by the European Commission, under contract number FP7-215453. Further, we would like to acknowledge the use of the English LYCOS iQ tag data set kindly provided by LYCOS Europe. Finally, we want to thank Andre Skusa and Nadine Wagner for our cooperation during the first steps of this work [25].

9. REFERENCES

[1] S. Alag. *Collective Intelligence in Action*. Manning Publications Co., October 2008.

[2] J. P. Bagrow. Evaluating local community methods in networks. *Journal of Statistical Mechanics*, P05001, 2008.

[3] C. H. Brooks and N. Montanez. Improved annotation of the blogosphere via autotagging and hierarchical clustering. In *Proceedings of WWW '06*, pages 625–632, New York, NY, USA, 2006. ACM.

[4] C. Cattuto, A. Baldassarri, V. D. P. Servedio, and V. Loreto. Emergent community structure in social tagging systems. *Advances in Complex Systems*, 11(4):597–608, 2008.

[5] C. Cattuto, D. Benz, A. Hotho, and G. Stumme. Semantic analysis of tag similarity measures in collaborative tagging systems. Technical report, 2008.

Table 3: Sample communities from the LYCOS iQ tag network. Distinction is made between "core" and "expansion" members depending on which step of the detection process they were attached to the community.

community theme	core - expansion tags
multiplayer games	**core** = {MMORPG, multi player, Leveling, warcraft, guides, guild wars} **expansion** = {wow, power levelling, Virtual Property, Diablo 2, browserbase, World of Warcraft, seti }
germs	**core** = {urinal, germs, floor, afraid, sick, bathroom, diseases} **expansion** = {Pores, transmitted, ritchey, varnish, hair thinning, tile, dirt, Black Death, kind, composite}
math	**core** = {maths homework, maths problem, MATH PROBLEM, Maths question, maths algebra, algebra} **expansion** = {division, Maths Sequence, pre algebra, halving, Maths Expression}
browsers	**core** = {mozilla, firefox, thunderbird, browser, browsers, Toolbar browsing, internet explorer, IE} **expansion** = {search plugin, shortcut keys, large print}
car transmission	**core** = {synchromesh, gargage, Van, warranty, mechanical, gearbox, VW T4, Automatic, braking system, braking, peugeot 206} **expansion** = {transmission, cd4e, clutch}
science report	**core** = {experiments, essays, enzymes, inhibitors, reports, substrates} **expansion** = {papers, conclusion}

Table 4: Sample communities from the BibSonomy tag network.

community theme	core - expansion tags
social networks	**core** = {unfiled, informationorganization, coordination, ict, scalefree, colaborative, properties, collaborate, huberman, coreperiphery, categorisation, computernetworks, commentary, cooperation, ecology, golder, quantitative, networkanalysis, farsimedia, socialnets, collaborativetagging} **expansion** = {dynamic, innovationteamwork, competencymanagement, socialsearch, emergent, competency, taggingtheorie, gsd, semiotics, ambiguous, folksononsense, ditributed, nondemocraticpolitics, newmedia, pressefreiheit, export, eni, selforganization, collective, emergence, logmining, collaborativefiltering, collaborativeresearch, retrieve, expertfinding, reticollab0607, skillinference}
sleep (medicine)	**core** = {cycle, hypothalamus, sleepmedicine, physiology, neuropeptides, hypocretin, sleepdisorders, hypothalamic, defects, orexin, sleepphysiology, sleepwakecycle} **expansion** = {narcolepsy, wake, genes, insomnia}
Israel-Palestine	**core** = {israelis, middleastpeace, peaceprocess, onevoice, palestinians, conflictresolution, hatred, extremism} **expansion** = {middleast, terrorism, conflict}
public relations	**core** = {stakeholder, publicrelations, marktkommunikation, kommunikationsmanagement, organisationsverfassung, unternehmensethik, kommunikationswissenschaft, bezugsgruppen, betriebswirtschaftslehre, kommunikationspolitik, oeffentlichkeitsarbeit} **expansion** = {stellenangebote, corporategovernance, medienmanagement, unternehmenskommunikation, ag, stellenboerse}

[6] A. Clauset. Finding local community structure in networks. *Physical Review E*, 72, 2005.

[7] A. Clauset, M. E. J. Newman, and C. Moore. Finding community structure in very large networks. *Physical Review E*, 70, 2004.

[8] N. Du, B. Wang, B. Wu, and Y. Wang. Overlapping community detection in bipartite networks. *International Conference on Web Intelligence and Intelligent Agent Technology*, 1:176–179, 2008.

[9] J. Duch and A. Arenas. Community detection in complex networks using extremal optimization. *Physical Review E*, 72, 2005.

[10] S. Fortunato and M. Barthelemy. Resolution limit in community detection. *Proceedings of the National Academy of Sciences USA*, 104(1):36–41, January 2007.

[11] S. Fortunato and C. Castellano. Community structure in graphs. Technical report, arXiv: 0712.2716, 2007.

[12] J. Gemmell, A. Shepitsen, B. Mobasher, and R. Burke. Personalizing navigation in folksonomies using hierarchical tag clustering. In *Proceedings of DaWak'08*, pages 196–205. Springer-Verlag, 2008.

[13] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences USA*, 99(12):7821–7826, June 2002.

[14] S. A. Golder and B. A. Huberman. Usage patterns of collaborative tagging systems. *Journal of Information Science*, 32(2):198–208, 2006.

[15] H. Halpin, V. Robu, and H. Shepherd. The complex dynamics of collaborative tagging. In *Proceedings of WWW '07*, pages 211–220, New York, NY, USA, 2007. ACM.

[16] A. Hotho, R. Jäschke, C. Schmitz, and G. Stumme. Emergent semantics in bibsonomy. In *Informatik 2006*, October 2006.

[17] A. Java, A. Joshi, and T. Finin. Detecting commmunities via simultaneous clustering of graphs and folksonomies. In *Proceedings of WebKDD'08*. ACM, August 2008.

[18] F. Luo, J. Z. Wang, and E. Promislow. Exploring local community structures in large networks. In *Web Intelligence*, pages 233–239. IEEE Computer Society, 2006.

[19] C. P. Massen and J. P. K. Doye. Identifying "communities" within energy landscapes. *Physical Review E*, 71, 2005.

[20] P. Mika. Ontologies are us: A unified model of social networks and semantics. In *International Semantic Web Conference*, pages 522–536, 2005.

[21] M. E. J. Newman. Fast algorithm for detecting community structure in networks. *Physical Review E*, 69, 2004.

[22] M. E. J. Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical Review E*, 74, 2006.

[23] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69, 2004.

[24] G. Palla, I. Derényi, I. Farkas, and T. Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435(7043):814–818.

[25] S. Papadopoulos, A. Skusa, A. Vakali, Y. Kompatsiaris, and N. Wagner. Bridge bounding: A local approach for efficient community discovery in complex networks. Technical report, arXiv: 0902.0871, 2009.

[26] T. Segaran. *Programming Collective Intelligence*. O'Reilly Media, Inc., August 2007.

[27] E. Simpson. Clustering tags in enterprise and web folksonomies. Technical report, HPL-2008-18, 2008.

[28] M. Tovey. *Collective Intelligence: Creating a Prosperous World at Peace*. Earth Intelligence Network, February 2008.

[29] X. Xu, N. Y. Z. Feng, and T. A. J. Schweiger. Scan: a structural clustering algorithm for networks. In *Proceedings of KDD '07*, pages 824–833, New York, NY, USA, 2007. ACM.