Detecting Overlapping Communities in Folksonomies

Abhijnan Chakraborty Saptarshi Ghosh Niloy Ganguly
Department of Computer Science and Engineering
Indian Institute of Technology Kharagpur, Kharagpur – 721302, India
{abhijnan,saptarshi,niloy}@cse.iitkgp.ernet.in

ABSTRACT

Folksonomies like Delicious and LastFm are modelled as tripartite (user-resource-tag) hypergraphs for studying their network properties. Detecting communities of similar nodes from such networks is a challenging problem. Most existing algorithms for community detection in folksonomies assign unique communities to nodes, whereas in reality, users have multiple topical interests and the same resource is often tagged with semantically different tags. The few attempts to detect overlapping communities work on *projections* of the hypergraph, which results in significant loss of information contained in the original tripartite structure. We propose the first algorithm to detect overlapping communities in folksonomies using the complete hypergraph structure. Our algorithm converts a hypergraph into its corresponding linegraph, using measures of hyperedge similarity, whereby any community detection algorithm on unipartite graphs can be used to produce overlapping communities in the folksonomy. Through extensive experiments on synthetic as well as real folksonomy data, we demonstrate that the proposed algorithm can detect better community structures as compared to existing state-of-the-art algorithms for folksonomies.

1. INTRODUCTION

Some of the most popular sites in the Web today are social tagging sites or folksonomies (e.g. Flickr, Delicious, LastFm, MovieLens etc.) where users share various types of resources (e.g. photos, URLs, music files, etc.) and collaboratively annotate the resources with descriptive keywords (tags) in order to facilitate efficient search and retrieval of interesting resources. Some folksonomies also encourage users to create a social network among themselves by connecting with other users having similar interests. With their growing popularity, a huge amount of resources is being shared on these folksonomies; consequently it has become practically impossible for a user to discover on her own, interesting resources and people having common interests. Hence it is important to develop algorithms for search as well as recommendation of resources and potential friends to the users. One approach to these tasks is to group the various entities (resources, tags, users) into communities or clusters, which are typically thought of as groups of entities having more/better interactions among themselves than with entities outside the group.

Folksonomies are modelled as *tripartite hypergraphs* having user, resource and tag nodes, where an hyperedge (u, t, r) indicates that user u has assigned tag t to resource r. Several algorithms have been proposed for detecting communities in hypergraphs, using techniques such as modularity maximization, identifying maximally connected subhypergraphs, and so on. But, almost all of the prior approaches do not consider an important aspect of the problem – they assign a single community to each node, whereas in reality, nodes in folksonomies frequently belong to *multiple overlapping communities*. For instance, users have multiple topics of interest, and thus link to resources and tags of many different semantic categories. Similarly, the same resource is frequently associated with semantically different tags by users who appreciate different aspects of the resource.

To the best of our knowledge, only two studies have addressed the problem of identifying overlapping communities in folksonomies. (i) Wang *et al.* [11] proposed an algorithm to detect overlapping communities of *users* in folksonomies considering only the user-tag relationships (i.e. the user-tag bipartite projection of the hypergraph), and (ii) Papadopoulos *et al.* [9] detected overlapping *tag* communities by taking a projection of the hypergraph onto the set of tags. Taking projections (as used by both these approaches) results in loss of some of the information contained in the original tripartite network and it is known that qualities of the communities obtained from projected networks are not as good as those obtained from the original network [5]. Also, none of these algorithms consider the resource nodes in the hypergraph. However, it is necessary to detect overlapping communities of users, resources and tags simultaneously for personalized recommendation of resources to users. Thus the goal of this paper is to propose such an algorithm that utilizes the complete tripartite structure to detect overlapping communities.

Though a node in a network can be associated to multiple semantic topics, a *link* is usually associated with only one semantics [1] – for instance, a user can have multiple topical interests, but each link created by the user is likely to be associated with exactly one of his interests. Link clustering algorithms utilize this notion to detect overlapping communities, by clustering links instead of the more conventional approach of clustering nodes – though each link is placed in exactly one link cluster, this automatically associates multiple overlapping communities with the nodes since a node inherits membership of all the communities into which its links are placed. Link clustering algorithms have recently been proposed for unipartite networks [1, 3] and bipartite networks [11]; however, to our knowledge, this is the first attempt to cluster links in tripartite hypergraphs.

Thus, the present work takes the first important step towards detecting overlapping communities in folksonomies considering the complete hypergraph structure. The al-

gorithm is detailed in Section 2 (a rudimentary version of the algorithm was presented in the poster [4]). We compare the performance of the proposed algorithm with the existing algorithms by Papadopoulos *et. al.* [9] and Wang *et al.* [11]. Extensive experiments on synthetically generated hypergraphs show that the proposed algorithm outperforms both these algorithms (Section 3). Further, using data from three popular real folksonomies – Delicious, MovieLens and LastFm – we also show that the proposed algorithm can identify better overlapping community structures in real folksonomies (Section 4). Section 5 concludes the paper.

2. OUR PROPOSED ALGORITHM

In this section, we present the proposed link-clustering algorithm for detecting overlapping communities in tripartite hypergraphs, which we name as 'Overlapping Hypergraph Clustering' algorithm (abbreviated to 'OHC'). As discussed earlier, a folksonomy is modelled as a tripartite hypergraph (more specifically 3-uniform tripartite hypergraph) $G = (V, E)$ where the vertex set V consists of 3 partite sets V^X, V^Y and V^Z. Each hyperedge in hyperedge set E connects a triple of nodes (a, b, c) where $a \in V^X, b \in V^Y$ and $c \in V^Z$.

For a given hypergraph G, we compute the *weighted line graph* $G^{'}$ which is a *unipartite graph* in which the hyperedges in G are *nodes*, and two nodes e_1 and e_2 in $G^{'}$ are connected by an edge if e_1 and e_2 are *adjacent* in G (i.e. the two hyperedges have at least one common node in G). The weight of the edge (e_1, e_2) in $G^{'}$ represents the similarity α between the two hyperedges e_1 and e_2 in the hypergraph G, which is computed as follows.

Let $N^X(i)$, $N^Y(i)$ and $N^Z(i)$ denote the set of neighbours of node i of type V^X, V^Y and V^Z respectively (if $i \in V^X$, then $N^X(i) = \phi$ since nodes in the same partite set are not linked). Similarity between two *adjacent* hyperedges $e_1 = (a, b, c)$ and $e_2 = (p, q, r)$ (where $a, p \in V^X$; $b, q \in V^Y$; $c, r \in V^Z$ and assumed $a = p$) is measured by the relative overlap among the neighbours of the non-common nodes of the same type:

$$\alpha(e_1, e_2) = \frac{|S \bigcap S^{'}| + |N^Y(c) \bigcap N^Y(r)| + |N^Z(b) \bigcap N^Z(q)|}{|S \bigcup S^{'}| + |N^Y(c) \bigcup N^Y(r)| + |N^Z(b) \bigcup N^Z(q)|}$$

where $S = N^X(b) \bigcup N^X(c)$ and $S^{'} = N^X(q) \bigcup N^X(r)$. Non-adjacent hyperedges are considered to have zero similarity.

It can be noted that the similarity for hyperedges can be computed in various other ways like expressing hyperedges as feature vectors and measuring cosine similarity or Pearson correlation among these feature vectors. We selected the above definition since it can be computed locally for a pair of hyperedges and can thus be computed efficiently for large real folksonomies. Further, a similar metric was found to perform well in detecting overlapping communities in unipartite graphs [1].

Once the *weighted line graph* $G^{'}$ is constructed from the given tripartite hypergraph G, any community detection algorithm for unipartite graphs (even the ones which do not produce overlapping communities) can be used to cluster the nodes in $G^{'}$ (i.e. the hyperedges in G). We used the Infomap algorithm [10] as this algorithm has been found to identify communities accurately as compared to several other algorithms [6]. Further, as Infomap has low computational complexity, it can be used efficiently on *weighted line graphs* of large real folksonomies. As we get the node communities in $G^{'}$, each hyperedge in G gets placed into a single link-community. This automatically assigns multiple overlapping communities to nodes in G, since a node inherits membership of all those communities into which the hyperedges connected with this node are placed.

Time Complexity: Let the number of nodes in the hypergraph be n and average node-degree be d, which implies that the number of hyperedges will be $\frac{nd}{3}$. Each hyperedge will, on average, be adjacent to $3(d-1)$ other hyperedges. So, the *line graph* will have $\frac{n.d}{3}$ nodes and $\frac{n.d}{3} \times 3(d-1) = n.d.(d-1) = O(n.d^2)$ edges. Since time complexity of infomap algorithm is linear in the size of the graph [6] and similarity calculation in the hypergraph also takes $O(n.d^2)$ time; the time complexity of OHC is $O(n.d^2)$. It is to be noted that real-world folksonomies are known to be sparse, having small average degree d. So, essentially the complexity of our algorithm becomes $O(n)$ which makes this algorithm scalable for work in large real world folksonomy.

3. EXPERIMENTS ON SYNTHETIC HYPERGRAPHS

In this section, we evaluate the performance of our proposed OHC algorithm by comparing with the algorithms by Wang *et al.* [11] and Papadopoulos *et al.* [9], which are henceforth referred to as 'CL' (abbreviation of 'Correlational Learning') and 'HGC' (as referred by the respective authors) respectively[1].

Since evaluation of clustering is difficult without the knowledge of 'ground truth' regarding the community memberships of nodes, we have used synthetically generated hypergraphs with a known community structure for evaluation of the algorithms. We discuss the generation of synthetic hypergraphs and the metric used to evaluate the algorithms, followed by the results of experiments on synthetic hypergraphs.

3.1 Generation of Synthetic Hypergraphs

Synthetic hypergraphs are generated using a modified version of the method used in [11]. The generator algorithm takes the following as input: (i) Number of nodes in a partite set (all 3 partite sets V^X, V^Y and V^Z are assumed to contain equal number of nodes), (ii) Number of communities C, (iii) Fraction γ of nodes which belong to multiple communities and (iv) Hyperedge density β (i.e. fraction of total number of hyperedges possible in the hypergraph).

Initially, the nodes in each partite set are evenly distributed among each community under consideration (e.g. $|V^X|/C$ nodes in the partite set V^X are assigned to each of the C communities). Subsequently, γ fraction of nodes are selected at random from each of V^X, V^Y and V^Z, and each selected node is assigned to some randomly chosen communities apart from the one it already has been assigned to. Nodes assigned to the same community are then randomly selected, one from each partite set, and interconnected with hyperedges. The number of hyperedges is decided based on the specified density β.

The above assignment of communities to nodes constitutes the 'ground truth'. After a hypergraph is generated,

[1]We acknowledge the authors for providing us with the implementations of their algorithms.

information about the communities is hidden, and then communities are detected from the hypergraph by different community detection algorithms. The community structure detected by each algorithm is compared with the ground truth using the metric 'Normalized Mutual Information (NMI)'.

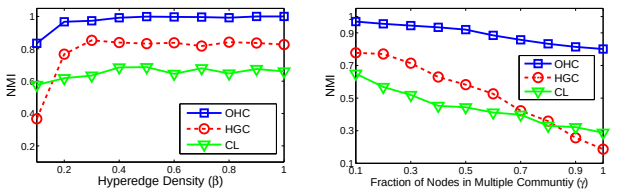
3.2 Normalized Mutual Information (NMI)

Normalized Mutual Information is an information-theoretic measure of similarity between two partitioning of a set of elements, which can be used to compare two community structures for the same graph (as identified by different algorithms). The traditional definition of NMI does *not* consider the case of a node being present in multiple communities; hence Lancichinetti *et al.* [7] proposed an alternative definition of NMI considering overlapping communities. The NMI value is in the range $[0, 1]$; higher the NMI value, the more similar are the two community structures (refer to [7] for details).

3.3 Results of Experiments

The CL and HGC algorithms produce only user and tag communities respectively. Hence, while calculating the NMI value for these algorithms, we have used the community memberships of only the user (respectively, tag) nodes according to the ground truth. Whereas the proposed OHC algorithm gives composite communities containing all three types of nodes. Hence, to evaluate the performance of OHC, we have considered the community memberships of all three types of nodes.

For all the following experiments, $|V^X| = |V^Y| = |V^Z| = 200$ and number of communities $C = 20$. For each result, random hypergraphs were generated 50 times using the same set of parameter values and the average performances over all 50 runs are reported.



(a) (b)

Figure 1: Comparison of proposed OHC algorithm with CL and HGC algorithms – variation of NMI values (a) with varying hyperedge density when 10% nodes belong to multiple communities and (b) with varying fraction of nodes in multiple communities keeping hyperedge density constant at 0.2

3.3.1 Performance w.r.t. number of Hyperedges:

To study how the number of hyperedges affects the performance of the clustering algorithms, we generated synthetic hypergraphs having various hyperedge densities $\beta = 0.1, 0.2, \ldots, 1.0$. In each of these hypergraphs, 10% of nodes in each partite set belonged to multiple communities (i.e. $\gamma = 0.1$). The NMI values for the three algorithms are shown in Figure 1(a). It can be clearly seen that, across all hyperedge densities, OHC performs significantly better than HGC and CL algorithms. A possible explanation for this is that the proposed OHC algorithm utilizes the complete tripartite structure of the hypergraph, whereas both CL and HGC algorithms work on unweighted projections which is known to result in loss of a significant part of the information contained in the original tripartite network [5].

Also note that even for very low hyperedge densities, when detecting community structures is difficult, the proposed OHC algorithm performs very well resulting in NMI scores above 0.8. This makes OHC suitable for real world folksonomies where hyperedge density is typically low.

Dataset	users	resources	tags	hyperedges
Delicious	1,867	69,226	53,388	4,37,593
LastFm	1,892	17,632	11,946	1,86,479
MovieLens	2,113	10,197	13,222	47,957

Table 1: Statistics of real folksonomy datasets

3.3.2 Performance w.r.t. Fraction of Nodes in Multiple Communities

A node belonging to multiple communities creates hyperedges to nodes in all those communities; hence, from the perspective of a particular community, the hyperedges created by this member node to nodes in other communities reduces the exclusivity of this particular community. As the number of nodes in multiple overlapping community increases, the fraction of such inter-community hyperedges increases making the community structure more difficult to identify.

We generated synthetic hypergraphs by varying the fraction of nodes in multiple communities (γ) while keeping hyperedge density (β) constant at 0.2. This low value of hyperedge density was chosen to measure the effectiveness of the algorithms in sparse environment (as in real-world foksnomies). Figure 1(b) shows that OHC performs consistently better than HGC and CL algorithms in this case as well. Further, as the community structure becomes more and more complex, the information loss as a result of projections becomes increasingly more crucial, hence the performance of the HGC and CL algorithms degrade sharply with increase in γ. On the other hand, the performance of our OHC algorithm shows relatively much greater stability.

The above experiments clearly validate our motivation and show that considering the complete tripartite structure of hypergraphs can result in better identification of community structure, as compared to considering projections (as done in prior studies).

4. EXPERIMENTS ON REAL WORLD FOLKSONOMIES

In this section, we apply the proposed OHC algorithm to gain insights into the community structures prevalent in real folksonomies. For this, we use the publicly available datasets [2] having snapshots of the folksonomies – Delicious (`http://www.delicious.com`), LastFm (`http://www.last.fm`) and MovieLens (`http://movielens.umn.edu`). The statistics of these data sets are summarized in Table 1.

4.1 Overlapping Communities in Folksonomies

For all three datasets, OHC algorithm successfully groups semantically related resources and tags and the users tag-

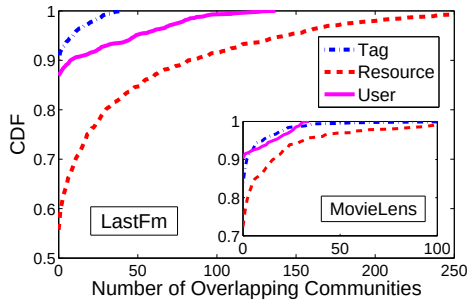


Figure 2: Cumulative distribution of the fraction of communities which overlap with a given number (x) of other communities; main figure – LastFm, inset – MovieLens

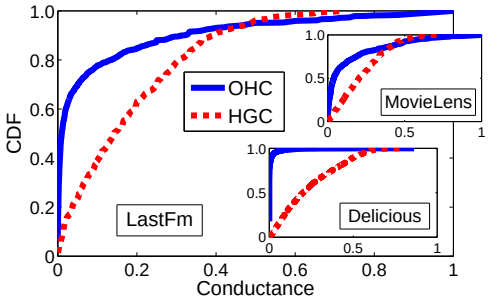


Figure 3: Cumulative distribution of conductance values of tag communities obtained from the real-world folksonomies: LastFm (main plot), Delicious and MovieLens (both inset) for OHC and HGC.

ging these resources. As an illustration, Table 2 shows the resources and tags placed in some example communities for each of the three datasets. It is evident that the resources and tags that are placed in the same community are often related to a common semantic theme. A closer look at Table 2 reveals that the algorithm also correctly identifies nodes that are related to multiple overlapping communities (themes). For instance, the band **Van Halen** is placed in two different communities detected from LastFm. The Wikipedia article about Van Halen[2] justifies this placement pointing their genre as both 'Hard Rock' and 'Heavy Metal'.

There are substantial amounts of overlap detected by OHC algorithm in all three datasets. Figure 2 shows the cumulative distribution of the fraction of communities which overlap with a given number of other communities, for LastFm and MovieLens. A similar pattern was detected in Delicious, which we omit due to lack of space.

4.2 Evaluation of Communities Detected

The principal difficulty in evaluating the communities detected in case of real folksonomies is the absence of 'ground truth' regarding the community memberships of nodes in folksonomies, since their huge size makes it impossible for human experts to evaluate the quality of identified communities. Hence, we use the following two methods for evaluation. First, we use the graph-based metric **Conductance**, which has been shown to correctly conform with the intuitive notion of communities and is extensively used for evaluating quality of communities in online social networks (see [8] for details). As conductance is defined only for unipartite networks, we compare tag communities detected by HGC with the tag nodes in the communities identified by our OHC algorithm.

Second, in case of the folksonomies which allow users to form a social network among themselves, we can assume that users having similar interests are likely to be linked in the social network, or to have a common social neighbourhood (a property known as *homophily*). We utilize this notion to evaluate the user communities detected by CL algorithm and the user nodes in the communities identified by OHC algorithm.

4.2.1 Comparison of Conductance Value

The Conductance [8] value ranges from 0 to 1 where a *lower* value signifies *better* community structure. Figure 3 shows the cumulative distribution of conductance values of detected tag communities by the two algorithms. Across all three datasets, OHC produces more communities having lower conductance values, which implies that OHC can find communities of better quality than obtained by HGC algorithm. The reason for this superior performance is that OHC groups semantically related nodes into relatively smaller cohesive communities instead of creating a few number of generalized large communities. For example of semantically related communities, refer to Table 2.

4.2.2 Comparing Detected User Communities with Social Network

In case of folksonomies which allow users to form a social network, there can be two types of relationships among users – explicit social connections (in the social network) and implicit connections through their tagging behaviour (e.g. tagging the same resource) in the hypergraph. A community detection algorithm for hypergraphs utilizes the implicit relationships to identify the community structure, and we propose to evaluate the detected community structure using the explicit connections that the users themselves create (in the social network). For instance, if a large fraction of the users who are socially linked (or share a common social neighbourhood in the social network) are placed in the same community (by the algorithm), the detected community structure can be said to group together users having common interests.

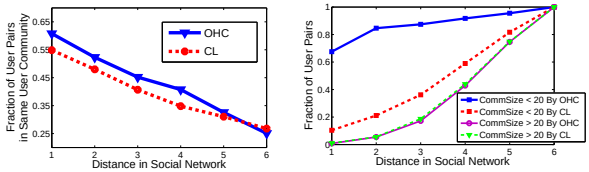
Hence, to compare the community structure identified by two algorithms, we consider the user-pairs who are within a certain distance from each other in the social network (where distance 1 implies friends, i.e. two users who are directly linked in the social network), and compute the fraction of such user-pairs who have been placed in a common community by the algorithm. Figure 4a shows the result for the proposed OHC algorithm and the CL algorithm, for the LastFm dataset. Across all distances, OHC places a larger number of user-pairs who share a common social neighbourhood, in a common community than the CL algorithm. Also, as the distance between two users in the social network increases, both algorithms put a smaller fraction of such user-pairs in the same community.

We can also investigate the reverse question – among the

[2] `http://en.wikipedia.org/wiki/Van_Halen`

Community	Theme	Example of member nodes
LastFm Artists (resources)	Hard Rock	***Van Halen, Deep Purple, Aerosmith***, Alice Cooper, Guns N' Roses, Scorpions, Kiss, Living Colour, White Lion, Bad Company, Bon Jovi, Hardline, The Rolling Stones
	Heavy Metal	***Van Halen, Deep Purple, Aerosmith***, Iron Maiden, Motorhead, Black Sabbath, Metallica, Twisted Sister, Crazy Lixx, Blind Guardian
LastFm Tags	Metal	***blues rock, psychedelic rock, rap metal, nu metal***, metal, symphonic metal, doom metal, progressive metal, speed metal, folk metal, metalcore, viking metal, power metal
	Rock	***blues rock, psychedelic rock, rap metal, nu metal***, progressive rock, polish rock, art rock, soft rock, gothic rock, polish, punk, punk rock, hard rock, glam rock, pop-rock
MovieLens Movies (resources)	Superhero	***The Incredibles, Shrek, Shrek 2, The Incredible Hulk***, Batman Begins, Batman Returns, Batman Forever, Spider-Man, Superman, Superman II, Superman III, X-Men
	Animation	***The Incredibles, Shrek, Shrek 2, The Incredible Hulk***, Shrek the Third, Kung fu Panda, Beowulf, WALL-E, Ratatouille, Finding Nemo, Cars, Toy Story, Toy Story 2
MovieLens Tags	Criticism	***violent, brutal***, too violent, waste of celluloid, disturbing, junk, tragically stupid, lousy script, pointless, waste of money, not very good, confusing plot, worst animated flick ever
	Violence	***violent, brutal***, violence, murder, fatality, civil war, great villain, dark, spanish civil war, serial killer, great war depiction, vietnam war, world war ii, best war film
Delicious Tags	Web 2.0	socialnetworking, socialweb, socialmedia, web20, php, drupal, xml, cms, webdesign, css3, twitter, skype, ruby, facebook, snippets, wikipedia, blog

Table 2: Examples of communities detected by proposed OHC algorithm. The algorithm successfully clusters nodes which are related to a common semantic theme (see column 2). Nodes related to multiple themes (boldfaced and italicized) are placed in overlapping communities.



(a) (b)

Figure 4: Community structure detected by the proposed OHC algorithm and the CL algorithm with the social network in (a) LastFm and (b) Delicious

users who are placed in a common community (by a community detection algorithm), what fraction of these users are actually connected in the social network (or share a common social neighbourhood)? While investigating this question, it is to be noted that 'quality' of large communities detected by community detection algorithms are known to be lower than smaller communities [8]. Hence it is meaningful to answer this question for detected communities taking their size into consideration. Figure 4b shows the fraction of users who are placed in a common community by the OHC and CL algorithms, that are within a certain distance in the social network (where distance 1 implies friends), for the Delicious dataset. For detected user-communities of size lesser than 20, more than 70% of the users who are placed in a common community by OHC are actually connected in the social network, whereas the corresponding value for the CL algorithm is much lesser. However, for larger detected communities (> 20 users), the fraction of user-pairs who share a common social neighbourhood is much lower and almost identical for both algorithms.

5. CONCLUSION

In this paper, we proposed the first algorithm to detect overlapping communities considering the full tripartite hypergraph structure of folksonomies. Through extensive experiments on synthetic as well as real folksonomy networks, we showed that the proposed algorithm out-performs existing algorithms that consider projections of hypergaphs. The proposed algorithm can be effectively used in recommending interesting resources and friends to users. Our future work will be to build such a recommendation system taking advantage of the effectiveness of the proposed algorithm.

6. REFERENCES

[1] Y.-Y. Ahn, J. P. Bagrow, and S. Lehmann. Link communities reveal multiscale complexity in networks. *Nature*, 466(7307):761–764, August 2010.

[2] I. Cantador, P. Brusilovsky, and T. Kuflik. Workshop on Information Heterogeneity and Fusion in Recommender Systems (HetRec 2011). In *ACM RecSys*, 2011.

[3] T. S. Evans and R. Lambiotte. Line graphs, link partitions, and overlapping communities. *Phys. Rev. E*, 80:016105, 2009.

[4] S. Ghosh, P. Kane, and N. Ganguly. Identifying overlapping communities in folksonomies or tripartite hypergraphs. In *ACM WWW (poster)*, pages 39–40, Mar 2011.

[5] R. Guimerà, M. Sales-Pardo, and L. A. N. Amaral. Module identification in bipartite and directed networks. *Phys. Rev. E*, 76:036102, Sep 2007.

[6] A. Lancichinetti and S. Fortunato. Community detection algorithms: a comparative analysis. *Phys. Rev. E*, 80:056117, Sep 2009.

[7] A. Lancichinetti, S. Fortunato, and J. Kertesz. Detecting the overlapping and hierarchical community structure in complex networks. *New Journal of Physics*, 11:033015, 2009.

[8] J. Leskovec, K. J. Lang, A. Dasgupta, and M. W. Mahoney. Statistical properties of community structure in large social and information networks. In *ACM WWW*, 2008.

[9] S. Papadopoulos, Y. Kompatsiaris, and A. Vakali. A graph-based clustering scheme for identifying related tags in folksonomies. In *Data Warehousing and Knowledge Discovery Conference*, pages 65–76, 2010.

[10] M. Rosvall and C. T. Bergstrom. Maps of random walks on complex networks reveal community structure. *PNAS*, 105:1118–1123, Jan 2008.

[11] X. Wang, L. Tang, H. Gao, and H. Liu. Discovering Overlapping Groups in Social Media. In *IEEE ICDM*, pages 569–578, 2010.