# An algorithm for detecting communities in folksonomy hypergraphs

Cécile Bothorel* and Mohamed Bouklit**

*Institut TELECOM - TELECOM Bretagne
UMR CNRS 3192 Lab-STICC
Technopôle Brest-Iroise CS 83818
29238 Brest Cedex 3 - France
cecile.bothorel@telecom-bretagne.eu

**Université Populaire Montpellier Averroès
123, rue de Salerne 34080 Montpellier - France
bouklit@upma.fr

**Abstract:** In this article, we are interested in social resource sharing systems such as Flickr, which use a lightweight knowledge representation called *folksonomy*. One of the fundamental questions asked by sociologists and actors involved in these online communities is to know whether a coherent tags categorization scheme emerges at global scale from folksonomy, though the users don't share the same vocabulary. In order to satisfy their needs, we propose an algorithm to detect clusters in folksonomies hypergraphs by generalizing the Girvan and Newman's clustering algorithm. We test our algorithm on a sample of an hypergragh of tag co-occurrence extracted from Flickr in September 2006, which gives promising results.

## 1 Introduction

The development of the different online communities goes with original regulation forms in which the *self-organization* principles play an important role. In the scope of this article, we are interested in social resource sharing systems, which use a lightweight knowledge representation called *folksonomy*. The word folksonomy is a blend of the words "taxonomy" and "folk" coined in 2004 by Thomas Vander Wal[1], and stands for conceptual structures created by the people. Resource sharing systems, such as Flickr[2] or YouTube[3] have acquired large number of users within these last years. Their users describe and organize the resources (photos, videos, etc.) with their own vocabulary and assign one or more keywords, namely *tags*, to each resource [CSB+07]. The folksonomy emerged thus through the different tags assigned. The folksonomy could be understood as an organization by folks of the resources over the Web. Being different from the traditional approaches to

[1] http://vanderwal.net/folksonomy.html
[2] http://www.flickr.com
[3] http://www.youtube.com

classification, the classifiers in folksonomy are not any more some dedicated professionals, and Thomas Vander Wal described this as a bottom-up social classification [SW05].

In such participative perspectives, online communities are doomed to fail if both the social scientists and the actors involved in these communities are not concerted. Our main goal is to supply social scientists with analysis and visualization tools that allows them to understand exchange structures and the gouvernementality particular forms of online communities such as Flickr. One of the fundamental questions which have inspired the present paper was whether a coherent tags categorization scheme emerges at global scale from folksonomy, though the users do not share the same vocabulary. We will focus in this article on the Flickr folksonomy case.

This work takes place within the pluridisciplinary field of large complex networks analysis and visualization [AB02, DM02, CbAH02]. Recent papers addressed the folksonomy analysis and tags clustering. After an overview on related works using a graph modelisation of a folksonomy, we briefly describe hypergraphs. We then detail our clustering algorithm. Finally, we present our results obtained on hypergraphs extracted from Flickr in september 2006[4].

## 2 Related works

Recent papers addressed the folksonomy analysis and tag clustering. A first approach is to study how tags are conjointly used, and thus build a graph where an edge exists if two tags have been used together to describe a resource or used by the same user. Such graphs are called *graphs of tag co-occurrence* and can reveal relevant semantic structures of tags [SW05]. But folksonomy involves the three basic actors of collaborative tagging, namely users, tags and resources. Understanding the global tags usage implies understanding the connections between these tags and how they are used, by which users, to describe which resource. The most intuitive way of modeling the relations between those three elements is to consider a tripartite graph. Since tripartite graph are rather difficult to manipulate (both algorithmics and interpretation), [YGS] propose to reduce into bipartite graphs. If *Mutual Contextualization* focuses on one user (respectively tag or resource), only the tags and resources associated to this user are extracted; in such a representation, an edge exists between a tag and a resource if the current user has annotated this resource with this tag. They center in this way the analysis on each of the three types of elements and provide then refined knowldege.

However these graphs representations waste information: each single tagging occurrence, e.g. "a user associates tags to a resource", disappears: a tag is connected to all the resources but without the memory of who made the association. [Mik05] introduced the

[4]Many thanks to the AUTOGRAPH project for providing us relevant data. AUTOGRAPH is a French project which is interested in self-organization and visualization of online communities on Internet. This pluridisciplinary project gathers in particular computer scientists from the University Paris VII and France Telecom, social scientists from the French EHESS School (advanced studies in social sciences) and the French national institute of demographic studies (INED), actors involved in online communities like Wikipedia and international civil society militants.

hypergraph modelisation to keep the tagging occurrences safe. Recent studies generalize graph algorithms to hypergraphs such as [CSB+07] and [BWR07]. We propose here an algorithm for clustering hypergraphs in order to address rich models of complex networks.

Recently, Estrada and Rodríguez-Velásquez introduced the *complex hyper-networks* as a natural generalization of the complex networks [ERV06]. The complex hyper-networks are hypergraphs encountered in practice that can modelize the structure of certain complex systems in a more precise way than the complex networks. In a graph, an edge connects only two nodes while the edges of a hypergraph (known as *hyperedges*) can link groups of several nodes and preserve a more realistic modelisation of a phenomena. Thus, they use hypergraph to model a co-authorship of scintific papers: the nodes are the authors and hyperedges correspond to groups of authors having published together. Estrada and Rodríguez-Velásquez proposed in the same article a generalisation of clustering coefficient to the complex hyper-networks. Brinkmeier has generalized his clustering algorithm [Bri03] to complex hyper-netwoks [BWR07].

In our study, contrary to [SW05], we will consider hypergraphs of tag co-occurrence where the hyperedges correspond to the set of tags which co-occur in the description of resources. Formally, a hypergraph of tag co-occurrence $H_T = (T, E_T)$ can be obtained by projection from the folksonomy $H$ : a set of tags are connected by a hyperedge in the hypergraph $H_T$ if they are all connected to a same couple $(u, r)$ in the folksonomy $H$. In addition, these hypergraphs have the advantage in practice to be much more compact in memory compared with graphs of tag co-occurrence.

## 3 Preliminaries on hypergraphs

A *hypergraph* is a generalisation of a graph, where the set of edges is replaced by a set of hyperedges. An *hyperedge* extends the notion of an edge by allowing more than two vertices to be connected by a hyperedge. Formally, a hypergraph is a pair $H = (V, E)$ , where $V = \{v_1, \cdots, v_n\}$ is the set of vertices and $E = \{e_1, \cdots, e_m\}$ is the set of hyperedges, which are nonempty subsets of $V$ such as $\bigcup_{i=1}^{i=m} e_i = V$ [Ber85]. The *size* of a hyperedge is defined as its cardinality. Two nodes are *adjacents* in $H = (V, E)$ if it exists a hyperedge $e_i$ which contains them. A *simple hypergraph* is a hypergraph $H$ such as $e_i \subseteq e_j \Rightarrow i = j$. A *simple graph* is a simple hypergraph, each edge of which has cardinality 2. A hypergraph $H$ can be represented by an *incidence matrice* $E(H) = (e_{ij})$ such as $e_{ij} \in \{0, 1\}$ in which each of $n$ rows is associated with a vertex and each $m$ column is associated with a hyper-edge:

$$\forall e_{ij} \in E(H), e_{ij} = \begin{cases} 1 & \text{if } v_i \in e_j \\ 0 & \text{otherwise.} \end{cases}$$

A *hyperpath* $P$ from $s \in V$ to $t \in V$ is defined as an alternate sequence of vertices and hyperedges $P = (s = v'_1, e'_1, \cdots, e'_{k-1}, v'_k = t)$ such that $P$ starts at $s$ and ends at $t$,

and for $1 \leq i \leq k-1$ the hyperedge $e'_i$ spans the vertices $v'_i$ and $v'_{i+1}$. The *length* of a hyperpath $P$ is the total number of hyperedges in the hyperpath $P$. Let $d_H(s,t)$ denote the minimum length of any hyperpath connecting $s$ and $t$ in $H$. By definition, $d_H(s,s) = 0$ and $d_H(s,t) = d_H(t,s)$.

Because of their low density in practice, we have chosen to represent the hypergraphs as bipartite graphs connecting the vertices to the hyperedges (which they belong). The complexity of this representation costs $\mathcal{O}(m+n+k)$ space (where $k$ denotes the number of edges of this bipartite graph). As we will consider only connected hypergraphs ($k \geq m+n-1$), this gives a spatial complexity of $\mathcal{O}(k)$ space.

We will consider throughout this paper a simple, undirected and unweighted hypergraph $H$ with $n = |V|$ vertices et $m = |E|$ hyperedges. We also suppose that $H$ is connected, the case where it is not being treated by considering the connected components as different hypergraphs. $E_s$ will denote the set of hyperedges of size $s$ in $H$.

## 4 Generalization of the Newman and Girvan's algorithm

As any large complex network, a folksonomy (modeled by a graph) reveals the presence of a *community* structure. A community $C$ is seen as a set of nodes whose edges proportion inside the community (internal edges) is high compared to the edges proportion outgoing from $C$ (external edge) [GN02, RCC$^+$04, FLG00, CNM04]. The goal is then to find communities satisfying this criterion. The field has recently received a large attention since the discovering of new algorithms which can be classified in two categories.

*The divisive approach*: [GN02, NG04, ACJM03, RCC$^+$04] divides the graph into many communities by removing one by one the edges connecting two different communities. On each step, the connected components of the remaining graph are identified as communities. The process is repeated until the removing of all edges. Finally, we obtain a communities hierarchical structure. The existing methods differ in the choice of the edges to remove.

*The agglomerative approach*: [New04, CNM04, DM04, PL06] is related to hierarchical clustering in which the vertices are merged iteratively into communities. Our algorithm is a generalization of the Newman and Girvan's algorithm [GN02, NG04] (described in the next subsection) to hypergraphs.

Our algorithm is a generalization of the divisive hierarchical decomposition algorithm by Newman and Girvan [GN02, NG04]. They argued that if a network contains distinguishable communities, then edges crossing communities boundaries should be relatively infrequent. Accordingly, these infrequent edges will have high betweenness centralities (the betweenness of an edge is defined by the proportion of shortest paths that runs along it) since all of the shortest paths between nodes in different communities would run along them. Thus, by removing these bridging edges, the underlying communities structure will reveal itself.

### 4.1 Description of our algorithm

We start from the partition $\mathcal{P}_1 = \{V\}$ containing only one community (corresponding to all the hypergraph). Then this partition evolves by repeating the following operations until no hyperedges remain: (1) compute all the nodes and hyperedges betweenness centralities presented in section 4.2 — complexity $\mathcal{O}(nk + k \log k)$ time; (2) remove the hyperedge with the highest betweenness score — complexity $\mathcal{O}(k)$ time; (3) compute a partition of the hypergraph into communities[5] — complexity $\mathcal{O}(k)$ time; (4) compute and store a quality parameter (called *hypermodularity*) $Q$ detailled in section 4.3 — complexity $\mathcal{O}(k \log k)$.

After $m$ steps, the algorithm finishes and we obtain the partition $\mathcal{P}_m = \{\{v\}, v \in V\}$ of the hypergraph into $n$ communities reduced to a single vertex. The algorithm induces a sequence $(\mathcal{P}_i)_{1 \leq i \leq m}$ of partitions into communities. The best partition is then considered to be the one that maximizes the hypermodularity $Q$.

As the complexity of an iteration is $\mathcal{O}(nk + k \log k)$ time, we can deduce that the overall worst case complexity of this algorithm is $\mathcal{O}(m(nk + k \log k))$ time. However, this upper bound is not reached in practical cases because most real-world complex networks are sparse ($m = \mathcal{O}(n)$) [CbAH02]. In this case, the complexity is therefore $\mathcal{O}(n^2 k + nk \log k))$ time.

Let's note that the original Newman and Girvan algorithm has a complexity of $\mathcal{O}(m^2 n)$ time, thus $\mathcal{O}(n^3)$ for sparse graphs.

### 4.2 Computing betweenness centrality

We describe here the algorithm we have proposed for computing the betweenness centrality measures of all the vertices and hyperedges in a hypergraph. The *betweenness centrality* of a vertex or a hyperedge $u$ (that we will note $B(u)$) is the proportion of shortest hyperpaths passing through $u$. Let's define the *dependency* of a vertex $s$ on a vertex or a hyperedge $u$ as $\delta_s(u) = \sum_{t \in V} \delta_{st}(u)$ where $\delta_{st}(u)$ denotes the fraction of shortest hyperpaths between vertices $s$ and $t$ that pass through $u$. Thus, the dependency $\delta_s(u)$ corresponds to the proportion of shortest hyperpaths starting at $s$ that pass through $u$. Clearly, we have $B(u) = \sum_{s \in V} \delta_s(u)$.

From this constatation, we sketch an algorithm for computing betweenness centrality for each node and hyperedge in $H$. The algorithm computes for each node $s \in V$ the dependency of $s$ on each vertex and each hyperedge $u$ of the hypergraph (namely $\delta_s(u)$) as follows:

- in the first time we compute in $\mathcal{O}(k)$ time the shortest hyperpath directed acyclic

[5] The connected components of the remaining hypergraph are identified as communities. We can find the connected components of a hypergraph with a BFS in $\mathcal{O}(k)$ time.

hypergraph (DAH) $D_s$ with a modified BFS. We define $D_s$ as follows: a vertex or a hyperedge $u$ is a parent of a vertex $t$ in $D_s$ if $u$ lies on a shortest hyperpath from $s$ to $t$. We also define $P_s(u)$ as the set of parents of $u$ in $D_s$. Thus, if a vertex $t$ has three parents in $D_s$ then it exists at least three short hyperpaths from $s$ to $t$. The figure 1.b shows the DAH $D_a$ computed from the hypergraph represented in the figure 1.a.

- in the second time we compute in $\mathcal{O}(k)$ time the dependency of the node $s$ on each hyperedge and each vertex, which are respectively set to 0 and 1. More precisely, we process the set of vertices or hyperedges in the reverse BFS order ($f$ $g$ $D$ $C$ $e$ $d$ $c$ $b$ $B$ $A$ $a$ in our case represented in the figure 1.b):
  - the dependency $\delta_s(u)$ is added to the betweenness centrality $B(u)$: $B(u) \leftarrow B(u) + \delta_s(u)$. When we process for example the hyperedge $D$, we add the dependency $\delta_a(D)$ (which will not increase in the rest of search) to its centrality $B(D)$.
  - $\delta_s(u)$ is then distributed evenly among its parents $w$: $B_v(w) \leftarrow B_v(w) + \frac{B_v(u)}{n_u}$ where $n_u$ denotes the number of parents of $u$. The hyperedge $D$ distributes for example the dependency $B_a(D) = 1$ fairly among its parents $c$ and $d$ which will receive then each one 0.5.

  To calculate correctly the dependency of the node $s$ on all vertices and hyperedges of the hypergraph, the approach we follow is similar to Girvan and Newman: multiple shortest hyperpaths between the vertices $s$ and $t$ are given equal weights summing to 1 (Figure 1.b). Thus, some hyperedges may lie in several shortest hyperpaths between the vertices $s$ and $t$ and then get greater dependency (such as the hyperedge D in our example).

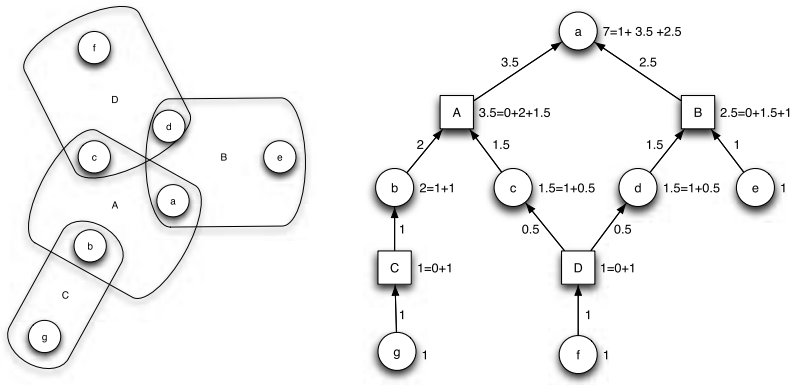
The figure 1.b illustrates then one iteration of the algorithm After $n$ iterations, we obtain the betweenness centralities for all vertices and hyperedges of the hypergraph $H$. As the complexity of an iteration is $\mathcal{O}(k)$ time, we can deduce that the overall worst case complexity of this algorithm is $\mathcal{O}(nk)$ time.

### 4.3 Evaluating the quality of a partition

We propose the *hypermodularity* $Q(P)$ in order to evaluate the quality of a partititon $P$ into communities:

$$Q(P) = \sum_{C \in P} \Big[ e(C) - \Big( \sum_{s=2}^{s=n} a_s(C)^s \Big) \Big] \tag{1}$$

where $e(C)$ is the fraction of hyperedges inside the community $C$ and $a_s(C)$ is the fraction of hyperedges of size $s$ bound to the community $C$ (hyperedges of size $s$ whose at least




a. An hypergraph. b. The directed acyclic hypergraph $D_a$.

Figure 1: An hypergraph and the directed acyclic hypergraph $D_a$ (bipartite graph representation). The hyperedge $D$ has for parents $c$ and $d$. Since there are two shortest hyperpaths between $a$ and $f$, each will be given weight 0.5.

one endpoint belongs to $C$). This quality measure is a generalization of the modularity introduced by Girvan and Newman in their algorithm.

An hyperedge is said to be *internal* to the community $C$ if all its endpoints are in the community $C$. The number of internal hyperedges equals thus to $|\{e \in E/e \subseteq C\}|$. The proportion of internal hyperedges $e(C)$ is taken compared to the total number of hyperedges $m$.

A hyperedge of size $s$ is said to be *bound* to the community $C$ if at least one of its $s$ endpoints belongs to the community $C$. Thus, the hyperedges of size $4$ having 2 endpoints in $C$ count for half ($\frac{2}{4}$) compared to the hyperedges of size 4 having all their endpoints in C. We obtain then the following expression for the proportion of internal hyperedges of size $s$ bound to $C$: $a_s(C) = \frac{\sum_{v_i \in C} \sum_{e_j \in E_s} e_{ij}}{sm}$.

The objective is to have communities of high internal density measured by $e(C)$. However, the large communities have mechanically a higher proportion of internal hyperedges: if $C$ is a random vertex set and if the hyperedges are also random then the expected proportion of internal hyperedges of size $s$ is $a_s(C)^s$. Indeed, each of $s$ endpoints of an hyperedge taken randomly has on this assumption a probability of $a_s(C)$ of being in the community $C$. Hence the total expected proportion of internal hyperedges is $\sum_{s=2}^{s=n} a_s(C)^s$.

Like the modularity, the hypermodularity compares the effective proportion of internal hyperedges with the expected proportion according to this schema. The hypermodularity is computed in $\mathcal{O}(k \log k)$ time. Because of lack of space, we omit the details of the algorithm computing the hypermodularity.

## 5 Application to Flickr hypergraphs of tag co-occurrence

As a first experimentation, we have applied our algorithm to hypergraphs of tag co-occurrence obtained from the photo sharing website Flickr. The nodes represents the tags and the hyperedges corresponds to the set of tags which co-occur *frequently* in the description of photos. The Flickr data has been extracted from the web site during September 2006. We here focus on a connected sub-hypergraph of 5,000 hyperedges (Figure 2).

{cat, vacation, cats}
{snow, trees, winter, alaska}
{family, vacation, friends}
{mountain, snow}
{trip, roadtrip, vacation}
{mountains, hiking, snow}

Figure 2: Few hyperedges extracted from Flickr (September 2006)


a. community *suzuki* b. community *wedding*

Figure 3: Examples of computed tag clouds. The displayed tags are the most representative of communities according to the centrality criterion.

Communities calculation captures cohesive sub-hypergraphs which unveil different associations of words corresponding to common sense shared by users. A tag with a high centrality means that people frequently use it in different contexts (presence of this hypernode on many shortest hyperpaths in the initial hypergraph). Therefore, the most central tags within a community are precisely the tags which reveal, through their usage, an emerging collective meaning (Figure 3). We can observe a consensus in the use of tags inside each tags community which seems to confirm the hypothesis of social scientists (Figure 3).

The participants expressed the need to handle multiple representation for a community. That's why we have proposed two representations: an ego-network presenting the graph where the tag *vacation* is connected to the close tag, and also and tag cloud. For the tags cloud representation, the police of each tag is proportional to its betweenness centrality in the initial hypergraph.

## 6 Conclusion

We have proposed in this paper an algorithm for clustering hypergraphs of tag co-occurrence. This algorithm allowed us to know whether a coherent tags categorization scheme emerge at global scale from folksonomy, through the users don't share the same vocabulary. According to our first experiments, the results are encouraging. As the field of complex hyper-networks is very recent, we also wanted through this paper to propose an algorithm for detecting communities in complex hyper-networks by generalizing the famous Girvan and Newman's algorithm. Nevertheless, better performances should be obtained by adapting our model to weighted hypergraphs and by reducing the complexity bounds.

## References


[AB02] Reka Albert and Albert-Laszlo Barabasi. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74:47, 2002.

[AC07] Christophe Aguiton and Dominique Cardon. The Strength of Weak Cooperation: An attempt to Understand the Meaning of Web2.0. *Communications & Strategies*, 65:51–65, 1st quarter 2007.

[ACJM03] D . Auber, Y. Chiricota, F. Jourdan, and G. Melancon. Multiscale Visualization of Small World Networks. In *INFOVIS '03: Proceedings of the IEEE Symposium on Information Visualization (INFOVIS'03)*, pages 75–81, 2003.

[Ber85] Claude Berge. *Graphs and Hypergraphs*. Elsevier Science Ltd, 1985.

[Bra01] Ulrik Brandes. A faster algorithm for betweenness centrality. *Journal of Mathematical Sociology*, 25(163), 2001.

[Bri03] Michael Brinkmeier. Communities in Graphs. In Thomas Böhme, Gerhard Heyer, and Herwig Unger, editors, *IICS*, volume 2877 of *Lecture Notes in Computer Science*, pages 20–35. Springer, 2003.

[BWR07] Michael Brinkmeier, Jeremias Werner, and Sven Recknagel. Communities in graphs and hypergraphs. In Mário J. Silva, Alberto H. F. Laender, Ricardo A. Baeza-Yates, Deborah L. McGuinness, Bjrn Olstad, ystein Haug Olsen, and André O. Falco, editors, *CIKM*, pages 869–872. ACM, 2007.

[CbAH02] R. Cohen, D. ben Avraham, , and S. Havlin. *Handbook of graphs and networks*. Wiley-VCH, 2002.

[CNM04] Aaron Clauset, M E J Newman, and Cristopher Moore. Finding community structure in very large networks. *Physical Review E*, 70, 2004.

[CSB+07] Ciro Cattuto, Christoph Schmitz, Andre Baldassarri, Vito D. P. Servedio, Vittorio Loreto, Andreas Hotho, Miranda Grahl, and Gerd Stumme. Network Properties of Folksonomies. *AI Communications*, 20(4):245 – 262, 2007.

[DM02] S N Dorogovtsev and J F F Mendes. Evolution of networks. *Advances in Physics*, 51:1079, 2002.

[DM04] Luca Donetti and Miguel A Munoz. Detecting Network Communities: a new systematic and efficient algorithm. *J.STAT.MECH.*, page P10012, 2004.

[ERV06] E. Estrada and J. A. Rodríguez-Velázquez. Subgraph centrality and clustering in complex hyper-networks. *Physica A: Statistical Mechanics and its Applications*, 364:581–594, May 2006.

[FLG00] Gary Flake, Steve Lawrence, and C. Lee Giles. Efficient Identification of Web Communities. In *Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 150–160, Boston, MA, August 20–23 2000.

[GN02] Michelle Girvan and M E J Newman. Community structure in social and biological networks. *PROC.NATL.ACAD.SCI.USA*, 99:7821, 2002.

[Mik05] Peter Mika. Ontologies Are Us: A Unified Model of Social Networks and Semantics. In Yolanda Gil, Enrico Motta, V. Richard Benjamins, and Mark A. Musen, editors, *The Semantic Web - ISWC 2005, Proceedings of the 4th International Semantic Web Conference, ISWC 2005, Galway, Ireland, November 6-10*, volume 3729 of *Lecture Notes in Computer Science*, pages 522–536. Springer, 2005.

[Mil67] Stanley Milgram. The small world problem. *Psychology Today*, 1:62–67, 1967.

[New04] M E J Newman. Fast algorithm for detecting community structure in networks. *Physical Review E, 69:066133*, 2004.

[NG04] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review*, E 69(026113), 2004.

[PCB+08] Christophe Prieur, Dominique Cardon, Jean-Samuel Beuscart, Nicolas Pissard, and Pascal Pons. The Stength of Weak cooperation: A Case Study on Flickr. *CoRR*, abs/0802.2317, 2008. informal publication.

[PL06] Pascal Pons and Matthieu Latapy. Computing communities in large networks using random walks. *Journal of Graph Algorithms and Applications (JGAA)*, 10(2):191–218, 2006.

[RCC+04] Filippo Radicchi, Claudio Castellano, Federico Cecconi, Vittorio Loreto, and Domenico Parisi. Defining and identifying communities in networks. *PROC.NATL.ACAD.SCI.USA*, 101:2658, 2004.

[Sco91] J. C. Scott. *Social Network analysis. A Handbook.* London. Sage., 1991.

[SW05] Kaikai Shen and Lide Wu. Folksonomy as a Complex Network. In *Proceedings of the Workshop Series on Knowledge in Social Software, Session 5*, London, GB, June 2005.

[Wal07] Thomas Vander Wal. Folksonomy, 2007. `http://vanderwal.net/folksonomy.html`.

[WF94] S. Wasserman and K. Faust. *Social Network Analysis. Methods and Applications.* Cambridge. Cambridge University Press., 1994.

[WS98] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393:440–442, June 1998.

[YGS] Ching Man Au Yeung, Nicholas Gibbins, and Nigel Shadbolt. Mutual Contextualization in Tripartite Graphs of Folksonomies. In *Proceedings of the 6th International Semantic Web Conference and 2nd Asian Semantic Web Conference (ISWC/ASWC2007), Busan, South Korea*, Berlin, Heidelberg.