

An algorithm for detecting communities in folksonomy hypergraphs

Cécile Bothorel
France Telecom R&D
2 avenue Pierre Marzin
22307 Lannion
cecile.bothorel@orange-ftgroup.com

Mohamed Bouklit
France Telecom R&D
2 avenue Pierre Marzin
22307 Lannion
mohamed.bouklit@orange-ftgroup.com

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 hypergraph of tag co-occurrence extracted from Flickr in September 2006, which gives promising results.

1 Introduction

The Web has become the scene of large scale cooperative activities conducted by communities of practice such as Wikipedia, free software developers, network players or social resource sharing systems. The development of these 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 the same kind of lightweight knowledge representation called *folksonomy*. The word 'folksonomy' is a blend of the words "taxonomy" and "folk" coined in 2004 by Thomas Vander Wal [13], and stands for conceptual structures created by the people. Resource sharing systems, such as Flickr (<http://www.flickr.com>), YouTube (<http://www.youtube.com>) or del.icio.us (<http://del.icio.us>) have acquired large number of users within these last years. Their users describe and organize the resources (photos, videos or webpages) with their own vo-

cabulary and assign one or more keywords, namely *tags*, to each resource [6]. 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 classification, the classifiers in folksonomy are not any more some dedicated professionals, and Thomas Vander Wal described this as a "bottom-up social classification" [12].

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. That's why our methodology associates them directly within the research project AUTOGRAPH¹ in order to well understand their folksonomy needs ([1], [11]). Our main goal is to supply social scientists with analysis and visualization tools that allows them to understand exchange structures and the governementality particular forms of online communities such as Flickr.

After several interviews with sociologists and actors involved in online communities, 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. The participants asked for a global visualization of all the tags, organized into tags groups, in order to verify if it appears a consensus or conflicts in the use of tags inside of each tags cluster.

This work takes place within the pluridisciplinary field of large complex networks analysis and visualization [2, 8, 7]. Recent papers addressed the folksonomy analysis and tags

¹AUTOGRAPH 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.

clustering. A first approach is to study how tags are jointly 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 [12]. But folksonomy involves the three basic actors/elements 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. However these 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. [10] introduced the hypergraph modelisation to keep the tagging occurrences safe. Recent studies generalize graph algorithms to hypergraphs such as [6] and [5]. We propose here an algorithm for clustering hypergraphs in order to address rich models of complex networks.

We will focus in this article on the Flickr folksonomy case. Flickr is a photo sharing online community. In addition to being a popular Web site for users to share personal photographs, the service is widely used by bloggers as a photo repository. Its popularity has been fueled by its innovative online community tools that allow photos to be tagged and browsed by *folksonomic* means. Flickr provides rapid access to images tagged with the most popular keywords. So, Flickr offers the possibility to organize photos collaboratively. Moreover, Flickr actors can give their friends, family, and other contacts permission to organize their photos.

This paper is organized as follows. After an overview on related works in the field of large complex hypernetworks and folksonomies analysis, and a short description on hypergraphs, we detail our clustering algorithm. Finally, we present our results obtained on hypergraphs extracted from Flickr in september 2006.

1.1 Related works

1.1.1 Complex hyper-networks

Since few years, abundant studies about large complex networks analysis and visualization emerge in many different domains such as sociology (acquaintances or collaborative networks), biology (metabolic networks, neuronal networks) or computer science (Internet topology, Web graph, P2P networks). Thus, these networks are often involved in the modeling of many complex systems. This new research field aims to describe the most significant properties

of these networks [2, 8, 7]. Although coming from different contexts, it appeared recently that these networks share statistical and structural common properties: low average distance, low global density, high local density, structuration into dense subgraph called *communities*.

The use of complex networks does not always provide a sufficiently detailed description of the studied complex systems structure. Thus, the graph representation of a scientific collaboration network only allows us to know if two scientists have been collaborated. However, it will not know if more than three researchers (connected in the network) have written a paper together.

Recently, Estrada and Rodríguez-Velásquez introduced the *complex hyper-networks* as a natural generalization of the complex networks [9]. 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. Thus, we can represent our scientific collaboration network by a hypergraph whose 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 [4] to complex hyper-networks [5].

1.1.2 Folksonomies

As the field of folksonomies is very recent, there are relatively few scientific publications about this topic. Shen and al considered graphs of tag co-occurrence where nodes represent the tags and different tags assigned to a resource are linked by edges [12]. They showed in particular that these networks have a low average distance and a high clustering coefficient. Recently, Cattuto and al modeled the folksonomy by a tripartite hypergraph $H = (V, E)$, in which the vertices set is partitioned into three disjoint sets: $V = T \cup U \cup R$ (T , U and R are finite sets, whose elements correspond to tags, users and resources respectively), and every hyperedge $\{t, u, r\}$ consists of exactly one tag t , one user u and one resource r [6]. They generalized the average distance and the clustering coefficient in this context. They observed that these tripartite hypergraphs are highly connected (high clustering coefficient) and that the average distance are relatively low, facilitating thus the "serendipitous discovery" of interesting contents and users [6].

In our study, contrary to [12], 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.

2 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 $V = \{v_1, \dots, v_n\}$ is the set of vertices and $E = \{e_1, \dots, e_m\}$ is the set of hyperedges, which are nonempty subsets of V such as

$\bigcup_{i=1}^{i=m} e_i = V$ [3]. The *size* of a hyperedge is defined as its cardinality. 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 matrix* $E(H) = (e_{ij})$ 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 $P = (s = v'_1, e'_1, \dots, e'_{k-1}, v'_k = t)$ 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 .

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 .

2.1 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:

- compute all the nodes and hyperedges betweenness centralities presented in section 2.2 (complexity: $\mathcal{O}(nk)$ time)
- remove the hyperedge with the highest betweenness score (complexity: $\mathcal{O}(k)$ time)
- compute a partition of the hypergraph into communities (complexity: $\mathcal{O}(k)$ time²)
- compute and store a quality parameter (called *hypermodularity*) Q detailed in section 2.3 (complexity: $\mathcal{O}(k \log k)$ time³)
- repeat from (1) until no hyperedges remain

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. Each step defines a partition \mathcal{P}_k of the hypergraph into communities. The algorithm induces a sequence $(\mathcal{P}_k)_{1 \leq k \leq m}$ 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 ($m = \mathcal{O}(n)$) [7]. In this case, the complexity is therefore $\mathcal{O}(n^2k + nk \log k)$ time.

2.2 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 number 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 number of shortest hyperpaths between vertices s and t that pass through u . Thus, the dependency $\delta_s(u)$ corresponds to the number 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

²The 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.

³In fact, the calculation of this quantity needs a preliminary edges sort in the bipartite graph (coding the hypergraph).

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 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.c):
 - 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 : $B_w(w) \leftarrow B_w(w) + \frac{B_u(u)}{n_u}$ 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.d). Since there are two shortest hyperpaths between a and f , each will be given weight 0.5.

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 figures 1.b and 1.c 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.

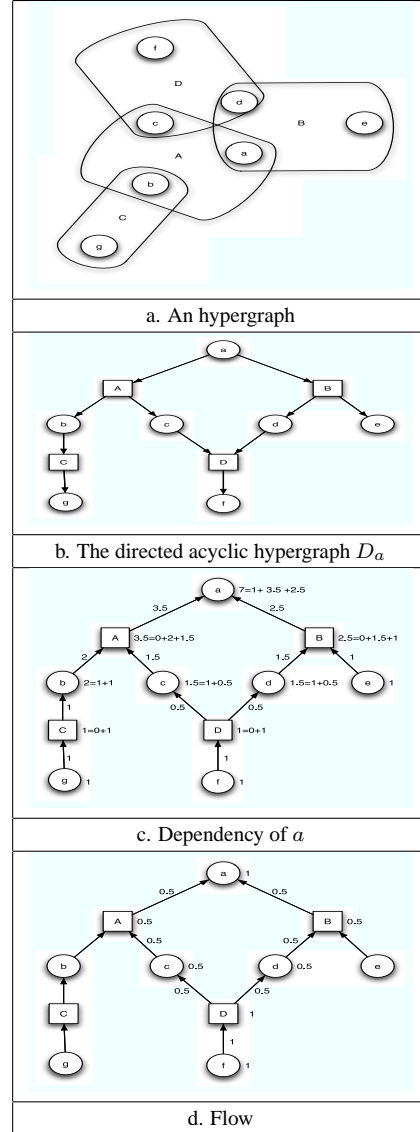


Figure 1. Computing betweenness centrality.

2.3 Evaluating the quality of a partition

We propose the *hypermodularity* $Q(P)$ in order to evaluate the quality of a partition P into communities: $Q(P) = \sum_{C \in P} \left[e(C) - \left(\sum_{s=2}^{s=n} a_s(C)^s \right) \right]$ 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 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 is taken compared to the total number of hyperedges m : $e(C) = \frac{|\{e \in E/e \subseteq C\}|}{m}$.

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 . The number of internal hyperedges of size s bound to C equals thus to

$\frac{\sum_{v_i \in C} \sum_{e_j \in E_s} e_{ij}}{s}$. 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. A community is all the more relevant as its proportion of internal hyperedges will be higher than its expected proportion of hyperedges. This is captured in the definition of the hypermodularity which our algorithm seeks to maximize. Therefore, we retain as result of our algorithm the partition of the hypergraph H having the best hypermodularity. 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.

3 Applications 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).

Communities calculation captures cohesive sub-hypergraphs which unveil different associations of words

{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)

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 4). 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: ego-network and tag cloud (Figure 5). For the tags cloud representation, the police of each tag is proportional to its betweenness centrality in the initial hypergraph.

4 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.

Acknowledgment

We first thank Pascal Pons for providing us relevant data in the AUTOGRAPH project. We also thank Dominique Cardon and Christophe Prieur for useful conversation and



Figure 3. Examples of tag clouds.

Matthieu Latapy and Guy Melançon for their helpful comments on preliminary versions.

References

- [1] C. Aguiton and D. Cardon. The strength of weak cooperation: An attempt to understand the meaning of web2.0. *Communications & Strategies*, 65:51–65, 1st quarter 2007.
- [2] R. Albert and A.-L. Barabasi. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74:47, 2002.
- [3] C. Berge. *Graphs and Hypergraphs*. Elsevier Science Ltd, 1985.
- [4] M. Brinkmeier. Communities in graphs. In T. Bhme, G. Heyer, and H. Unger, editors, *IICS*, volume 2877 of *Lecture Notes in Computer Science*, pages 20–35. Springer, 2003.
- [5] M. Brinkmeier, J. Werner, and S. Recknagel. Communities in graphs and hypergraphs. In M. J. Silva, A. H. F. Laender, R. A. Baeza-Yates, D. L. McGuinness, B. Olstad, ystein Haug Olsen, and A. O. Falco, editors, *CIKM*, pages 869–872. ACM, 2007.
- [6] C. Cattuto, C. Schmitz, A. Baldassarri, V. D. P. Servedio, V. Loreto, A. Hotho, M. Grahl, and G. Stumme. Network properties of folksonomies. *AI Communications*, 20(4):245–262, 2007.
- [7] R. Cohen, D. ben Avraham, , and S. Havlin. *Handbook of graphs and networks*. Wiley-VCH, 2002.
- [8] S. N. Dorogovtsev and J. F. F. Mendes. Evolution of networks. *Advances in Physics*, 51:1079, 2002.

vacation(10) snow(2.99) hiking(2.84)
 florida(2.53) camping(2.02) hawaii(1.79)
 mountains(1.47) roadtrip(1.41) colorado(1.35)
 arizona(1.23)

Figure 4. The community *vacation* represents the consensual tags around the concept *vacation*. The displayed tags are the most representative of communities according to the centrality criterion.

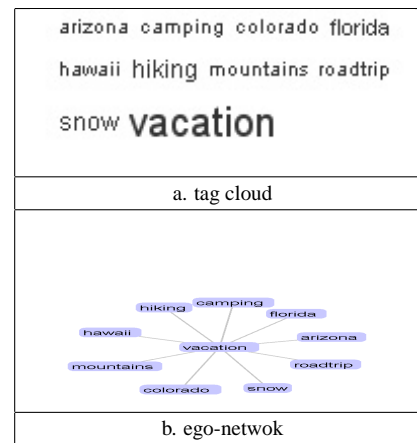


Figure 5. Two representations of the community *vacation*.

- [9] 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.
- [10] P. Mika. Ontologies are us: A unified model of social networks and semantics. In Y. Gil, E. Motta, V. R. Benjamins, and M. 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.
- [11] C. Prieur, D. Cardon, J.-S. Beuscart, N. Pissard, and P. Pons. The strength of weak cooperation: A case study on flickr. *CoRR*, abs/0802.2317, 2008. informal publication.
- [12] K. Shen and L. Wu. Folksonomy as a complex network. In *Proceedings of the Workshop Series on Knowledge in Social Software, Session 5*, London, GB, June 2005.
- [13] T. V. Wal. Folksonomy, 2007. <http://vanderwal.net/folksonomy.html>.