Detecting Communities in Tripartite Hypergraphs

Xin Liu
Tokyo Institute of Technology
W8-59 2-12-1 Ookayama, Meguro
Tokyo, 152-8552 Japan
tsinllew@ai.cs.titech.ac.jp

Tsuyoshi Murata
Tokyo Institute of Technology
W8-59 2-12-1 Ookayama, Meguro
Tokyo, 152-8552 Japan
murata@cs.titech.ac.jp

1. INTRODUCTION

In social tagging systems, users collaboratively manage tags to annotate resources. Naturally, social tagging systems can be modeled as a tripartite hypergraph, where there are three different types of nodes, namely users, resources and tags, and each hyperedge has three end nodes, connecting a user, a resource and a tag that the user employs to annotate the resource.

As for community detection in tripartite hypergraphs, a common strategy is to reduce the tripartite hypergraph to simpler unipartite graphs, bipartite graphs, or tripartite graphs, and then detect communities in the corresponding graphs [6, 8, 13]. One major drawback of this class of methods is that some valuable information of the original hyperedges is lost during reduction [12], and the subsequently detected communities tend to be less accurate. Researchers also proposed extended modularity optimization [7] and tensor decomposition [4] methods. But these two methods are biased towards communities with one-to-one correspondence, as shown in Fig 1(a). Real-world social tagging systems are often more complex than that. For example, a group of users may be interested in a collection of resources about programming technology and another collection about sports. Hence, communities with many-to-many correspondence, as shown in Fig 1(b), are more significant. Besides, another disadvantage of some previous methods [4, 6] is that they require one to specify certain parameters such as the numbers of communities. In practice, such a priori knowledge is difficult to obtain.

In this paper, we propose a quality function, based on the minimum description length (MDL) principle [10], for measuring the goodness of different partitions of a tripartite hypergraph into communities, and develop a community detection algorithm based on minimizing the quality function. Our method overcomes the limitations of previous methods and has the following key properties:

- Independent: it handles broad families of tripartite hypergraphs, and is competent for both communities with one-to-one correspondence and many-to-many correspondence.
- Parameter-free: given the structure of a tripartite hypergraph, it can automatically detect communities, without any prior knowledge like the numbers of communities.
- Accurate: it is more accurate than previous methods.
- Scalable: it is fast and scalable to large-scale hypergraphs.

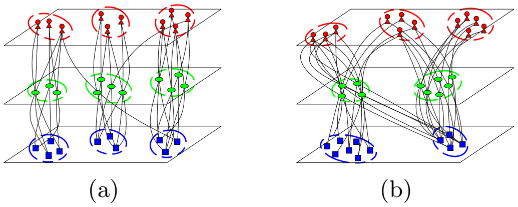

Figure 1: Communities with (a) one-to-one correspondence, and (b) many-to-many correspondence.

2. PROBLEM FORMULATION

One fundamental issue is the definition of community in tripartite hypergraphs. Generally, a community should be a group of related nodes that correspond to a functional subunit in the real-world system. In unipartite graphs, a community is often understood as a group of nodes with dense connections between them. But this notion is not suitable for tripartite hypergraphs, since nodes of the same type are not connected. Instead, we consider a tripartite hypergraph community as a group of nodes that are structurally equivalent (in a weakened sense, and the same hereinafter) [9]. This is a natural assumption, because a group of nodes that are similar with one another as regards their relations to nodes of other types are very likely to form a functional subunit. For example, in a social tagging system, those users having similar tagging actions are very likely to share the same interests; those resources that are annotated with similar tags are very likely to be in the same category. Meanwhile, the dense connections between certain communities in different node sets constitute their correspondence.

In the following, we formulates the problem of community detection in tripartite hypergraphs. Now assume an undirected and unweighted tripartite hypergraph $\boldsymbol{H} = (\boldsymbol{V}^r \cup \boldsymbol{V}^g \cup \boldsymbol{V}^b, \boldsymbol{E})$, where $\boldsymbol{V}^r$, $\boldsymbol{V}^g$ and $\boldsymbol{V}^b$ are three disjoint node sets, and $\boldsymbol{E} \subseteq \{(v_i^r \in \boldsymbol{V}^r, v_j^g \in \boldsymbol{V}^g, v_k^b \in \boldsymbol{V}^b)\}$ is the set of three-way hyperedges. For simplicity, nodes of the three different types $v_i^r \in \boldsymbol{V}^r$, $v_j^g \in \boldsymbol{V}^g$ and $v_k^b \in \boldsymbol{V}^b$ are colored red, green and blue, respectively. Suppose $n^r = |\boldsymbol{V}^r|$, $n^g = |\boldsymbol{V}^g|$ and $n^b = |\boldsymbol{V}^b|$ are the numbers of red, green and blue nodes, and $m = |\boldsymbol{E}|$ the number of hyperedges. The structure of

Table 1: Notations for a tripartite hypergraph $\boldsymbol{H}$

Symbol	Meaning
$\boldsymbol{V}^r$	The red node set
$\boldsymbol{V}^g$	The green node set
$\boldsymbol{V}^b$	The blue node set
$\boldsymbol{E}$	The hyperedge set
n^r	The number of red nodes
n^g	The number of green nodes
n^b	The number of blue nodes
m	The number of hyperedges
v_i^r	The i-th red node
v_j^g	The j-th green node
v_k^b	The k-th blue node
c^r	The number of red communities
c^g	The number of green communities
c^b	The number of blue communities
$\boldsymbol{V}_\alpha^r$	The α-th red community
$\boldsymbol{V}_\beta^g$	The β-th green community
$\boldsymbol{V}_\gamma^b$	The γ-th blue community
n_α^r	The number of nodes in $\boldsymbol{V}_\alpha^r$
n_β^g	The number of nodes in $\boldsymbol{V}_\beta^g$
n_γ^b	The number of nodes in $\boldsymbol{V}_\gamma^b$
r_i	The vector $\mathbf{r}$ whose i-th element indicates the community membership of v_i^r
g_j	The vector $\mathbf{g}$ whose j-th element indicates the community membership of v_j^g
b_k	The vector $\mathbf{b}$ whose k-th element indicates the community membership of v_k^b
A_{ijk}	The three-dimensional array $\mathbf{A}$ whose (i, j, k) element indicates the number of hyperedges between v_i^r, v_j^g and v_k^b
$M_{\alpha\beta\gamma}$	The three-dimensional array $\mathbf{M}$ whose (α, β, γ) element indicates the number of hyperedges between $\boldsymbol{V}_\alpha^r$, $\boldsymbol{V}_\beta^g$ and $\boldsymbol{V}_\gamma^b$

$\boldsymbol{H}$ can be represented by a three-dimensional binary array $\mathbf{A}$ of $n^r \times n^g \times n^b$ size, with elements

$$A_{ijk} = \begin{cases} 1 & \text{if } (v_i^r, v_j^g, v_k^b) \in \boldsymbol{E}; \\ 0 & \text{otherwise.} \end{cases}$$

The problem of community detection in $\boldsymbol{H}$ is that, given $\mathbf{A}$, how can we find a good partition $\mathscr{C} = \{\boldsymbol{V}_\alpha^r\}_{\alpha=1}^{c^r} \oplus \{\boldsymbol{V}_\beta^g\}_{\beta=1}^{c^g} \oplus \{\boldsymbol{V}_\gamma^b\}_{\gamma=1}^{c^b}$ that divides $\boldsymbol{V}^r$, $\boldsymbol{V}^g$ and $\boldsymbol{V}^b$ into disjoint communities, respectively:

$$\begin{aligned} \oplus\{\boldsymbol{V}_\alpha^r\}_{\alpha=1}^{c^r} &= \boldsymbol{V}^r \\ \oplus\{\boldsymbol{V}_\beta^g\}_{\beta=1}^{c^g} &= \boldsymbol{V}^g \\ \oplus\{\boldsymbol{V}_\gamma^b\}_{\gamma=1}^{c^b} &= \boldsymbol{V}^b \end{aligned}$$

Note that the numbers of red, green and blue communities c^r, c^g and c^b are not known a priori. The meaning of "good" is twofold: i) nodes in the same community are structurally equivalent; ii) hyperedges between communities are either dense or sparse, so that the correspondence between communities is clear.

Throughout the paper, we use Latin letters i, j and k for indices of red, green and blue nodes, respectively, and use Greek letters α, β and γ for indices of red, green and blue communities, respectively. Table 1 summarizes the notations used in this paper.

3. THE PROPOSED METHOD

In this section, we first define a quality function for measuring the goodness of different partitions of a tripartite hypergraph into communities, and then propose an algorithm for minimizing the quality function.

When we describe a graph as a set of communities, we are highlighting certain regularities (e.g., the similarities of nodes in the same community and the dissimilarities of nodes between different communities) while filtering out relatively unimportant details (e.g., the dissimilarities of nodes in the same community). Thus, description of a graph as communities can be viewed as a lossy compression of that graph's structure, and the community detection problem as a problem of finding an efficient compression of the structure. This is the main insight of the structural information compression method proposed in [11], where the authors focus on information compression on a unipartite graph's structure. Here we show how to compress the structural information of a tripartite hypergraph, in order to formulate our quality function.

Now let us envision a communication process of transmitting structural information of a tripartite hypergraph $\boldsymbol{H}$. A signaler knows the structure of $\boldsymbol{H}$ and aims to transmit much of the information in a reduced fashion to a receiver over a noiseless channel. To do so, the signaler makes a partition of $\boldsymbol{H}$ into communities and encodes the structural information $\mathrm{X} = \{\mathbf{A}\}$ as compressed information summarizing the community structure: $\mathrm{Y} = \{\mathbf{r}, \mathbf{g}, \mathbf{b}, \mathbf{M}\}$, where $\mathbf{r}$, $\mathbf{g}$ and $\mathbf{b}$ are the community membership vectors of red, green and blue nodes, and $\mathbf{M}$ is the community connectivity array. For a partition dividing red, green and blue nodes into c^r, c^g and c^b communities, we have $\mathbf{r} = [r_1, r_2, \ldots, r_{n^r}]$, $\mathbf{g} = [g_1, g_2, \ldots, g_{n^g}]$ and $\mathbf{b} = [b_1, b_2, \ldots, b_{n^b}]$, where $r_i \in \{1, 2, \ldots, c^r\}$, $g_j \in \{1, 2, \ldots, c^g\}$ and $b_k \in \{1, 2, \ldots, c^b\}$ indicate the community memberships of nodes v_i^r, v_j^g and v_k^b, respectively. The community connectivity array $\mathbf{M}$ is a three-dimensional array of $c^r \times c^g \times c^b$ size, with element $M_{\alpha\beta\gamma} \in \{0, 1, 2, \ldots, m\}$ indicating the number of hyperedges between communities $\boldsymbol{V}_\alpha^r$, $\boldsymbol{V}_\beta^g$ and $\boldsymbol{V}_\gamma^b$. That is

$$M_{\alpha\beta\gamma} = \sum_{v_i^r \in \boldsymbol{V}_\alpha^r} \sum_{v_j^g \in \boldsymbol{V}_\beta^g} \sum_{v_k^b \in \boldsymbol{V}_\gamma^b} A_{ijk}$$

It is easy to derive that the description length (in bits) of the compressed information Y is

$$\mathrm{L(Y)} = n^r \mathrm{log} c^r + n^g \mathrm{log} c^g + n^b \mathrm{log} c^b + c^r c^g c^b \log(m+1)$$

where the logarithm is taken in base 2.

After receiving Y, the receiver knows the community membership of each node and the number of hyperedges between each community triple. Then he tries to recover the original structural information X by constructing possible candidates. The number of different candidates is given by

$$\prod_{\alpha=1}^{c^r} \prod_{\beta=1}^{c^g} \prod_{\gamma=1}^{c^b} \binom{n_\alpha^r n_\beta^g n_\gamma^b}{M_{\alpha\beta\gamma}}$$

where $n_\alpha^r = |\boldsymbol{V}_\alpha^r|$, $n_\beta^g = |\boldsymbol{V}_\beta^g|$ and $n_\gamma^b = |\boldsymbol{V}_\gamma^b|$ are the numbers of nodes in communities $\boldsymbol{V}_\alpha^r$, $\boldsymbol{V}_\beta^g$ and $\boldsymbol{V}_\gamma^b$, the parentheses denote the binomial coefficient, and each binomial coefficient gives the number of different candidates for recovering the original $M_{\alpha\beta\gamma}$ hyperedges between $\boldsymbol{V}_\alpha^r$, $\boldsymbol{V}_\beta^g$ and $\boldsymbol{V}_\gamma^b$. Hence, the description length of the additional information for the receiver to recover X (i.e. the conditional information between X and Y) is

$$\mathrm{L(X|Y)} = \log\left[\prod_{\alpha=1}^{c^r} \prod_{\beta=1}^{c^g} \prod_{\gamma=1}^{c^b} \binom{n_\alpha^r n_\beta^g n_\gamma^b}{M_{\alpha\beta\gamma}}\right].$$

Algorithm 1: Detecting communities in a tripartite hypergraph $\boldsymbol{H}$ by minimizing quality function Q

```
Input: Connectivity array $\mathbf{A}$ of $\boldsymbol{H}$
Output: Partition of $\boldsymbol{H}$ into communities
begin
    // Phase 1
    assign each node in $\boldsymbol{H}$ a unique label;
    repeat
        update each node's label;
    until a local minimum of Q
    repeat
        // Phase 2
        build a reduced tripartite hypergraph $\boldsymbol{H'}$;
        assign each node in $\boldsymbol{H'}$ a unique label;
        repeat
            update each node's label;
        until a local minimum of Q
        // Phase 1
        retrieve labels in $\boldsymbol{H}$ from the corresponding labels in
        $\boldsymbol{H'}$;
        repeat
            update each node's label;
        until a local minimum of Q
    until no change in Q
    identity communities as groups of nodes bearing the same
    labels;
end
```

The objective is that signaler transmits the least while the receiver receives the most. Intuitively, if the signaler makes a "good" partition as described in Section 2, which capitalizes on regularities in the hypergraph's structure, the compression based on it would achieve the optimal tradeoff between L(Y) and L(X|Y). According to the minimum description length (MDL) principle [10],

$$\begin{aligned}\mathrm{Q}(\mathscr{C}) &= \mathrm{L(Y)} + \mathrm{L(X|Y)} \\ &= n^r \log c^r + n^g \log c^g + n^b \log c^b + c^r c^g c^b \log(m+1) \\ &\quad + \log\left[\prod_{\alpha=1}^{c^r}\prod_{\beta=1}^{c^g}\prod_{\gamma=1}^{c^b}\binom{|\boldsymbol{V}_\alpha^r||\boldsymbol{V}_\beta^g||\boldsymbol{V}_\gamma^b|}{\sum\limits_{v_i^r\in\boldsymbol{V}_\alpha^r}\sum\limits_{v_j^g\in\boldsymbol{V}_\beta^g}\sum\limits_{v_k^b\in\boldsymbol{V}_\gamma^b} A_{ijk}}\right] \qquad (1)\end{aligned}$$

would get the minimum value. This is the quality function for measuring the goodness of a partition $\mathscr{C}$ of a tripartite hypergraph into communities.

Now we can evaluate a partition based on the quality function Q, and a low value of Q indicates a good partition. So the task is to search over all possible partitions for one that has a minimum Q. However, like modularity optimization, finding the global optimal solution is NP-hard [2]. We develop an approximate algorithm that can be implemented in near linear time, as presented in Algorithm 1. For more information, please refer to [1, 5].

4. EVALUATION

In this section, we concentrate on comparing our method with previous ones in terms of accuracy. The basic scheme is as follows: we apply various methods to a set of synthetic tripartite hypergraphs with known community structure (the true partition), and compare the similarities between partitions obtained by different methods and the true partition; the closer of an obtained partition to the true partition, the better of the corresponding method. To quantify the similarity between two partitions, we use *normalized mutual information* (NMI) [3], which has a maximum value of 1 if two partitions match completely, and a minimum value of 0 if they are totally independent of one another.

We consider several opponent methods that cover state of the art techniques. They are, in order, the extended modularity optimization method (ExModularity) proposed by Murata [7], the tensor decomposition method (MetaFac) presented by Lin et al. [4], and the method advanced by Neubauer et al., which involves reduction of a tripartite hypergraph to bipartite graphs (BiNetReduction) [8]. In addition, we consider another method (UniNetReduction) modified from Zlatić's approach [13], which involves reduction of a tripartite hypergraph to unipartite graphs.

For the first case, we consider comparing these methods in a set of synthetic tripartite hypergraphs with built in communities of one-to-one correspondence (detailed procedure for generating the dataset is omitted here). In Fig. 2, we show the performances of various methods in this set of hypergraphs. On the whole, performance of each method varies in a similar way across red, green and blue node sets (since red, green and blue nodes are in a symmetric status in the hypergraph generation procedures). Specifically, ExModularity, BiNetReduction and our method perform excellently, correctly detecting not only the numbers of communities but also community membership of each node almost all the way to the point p_{dense}=0.05. At the turning stage, i.e. p_{dense} falling from 0.045 to 0.03, our method slightly outperforms ExModularity and BiNetReduction, as shown in the embedded figures. Thereafter, performances of the three methods deteriorate markedly. MetaFac, though given a prior knowledge of the true numbers of communities, does not provide remarkable result. The record for UniNetReduction is even worse. Its performance decreases as early as p_{dense}=0.055. When $p_{dense}\leq$0.3, it loses most of the information about the true partition.

For the second case, we generated a set of synthetic tripartite hypergraphs with built in communities of many-to-many correspondence (detailed procedure for generating the dataset is omitted here). Applying different methods to this set of synthetic hypergraphs, we calculate NMI between obtained partitions and the true partition. The results are shown in Fig. 3 (values are averaged over 20 runs). As Fig. 3 shows, our method outperforms others by a large margin. It works almost perfectly all the way until p_{dense}=0.015, with a sudden dramatic fall thereafter. As for other methods, we can observe three common features. 1) None of them can detect community membership with 100% accuracy, even when p_{dense}=0.08. 2) Their best performances are in red node set, the middle in green node set, and the worst in blue node set. 3) Their performances deteriorate much earlier than our method, often with records fluctuating wildly before the turning points. In specific, UniNetReduction is the best among them in most of the time, followed by MetaFac. Note that MetaFac is given at least an estimate of the true numbers of communities, so its performance is not appealing. Contrary to the excellent performance in the previous set of hypergraphs, BiNetReduction and ExModularity do not show satisfactory result this time.

5. CONCLUSION

Based on the information compression idea, we define a quality function for measuring the goodness of different partitions of a tripartite hypergraph into communities, and develop an algorithm for minimizing this quality function.

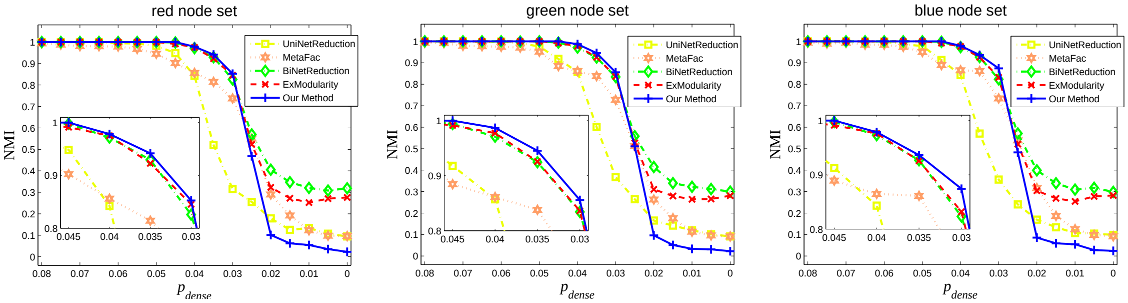

Figure 2: Performances in the synthetic dataset with built in communities of one-to-one correspondence.

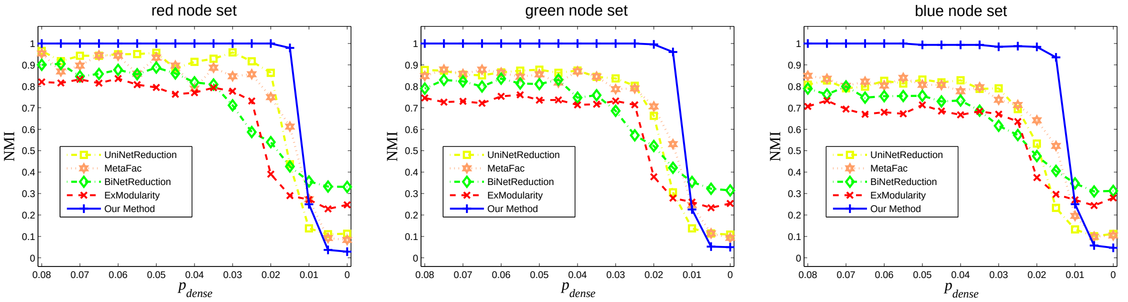

Figure 3: Performances in the synthetic dataset with built in communities of many-to-many correspondence.

Compared with previous methods, our method is competent for both communities with one-to-one correspondence and many-to-many correspondence. It should be emphasized that our method is parameter-free. In the future, we would like to apply our method to real-world social tagging systems.

6. REFERENCES

[1] V. D. Blondel, J. L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *J. Stat. Mech.*, page P10008, 2008.

[2] U. Brandes, D. Delling, M. Gaertler, R. Görke, M. Hoefer, Z. Nikolski, and D. Wagner. On modularity–np-completeness and beyond. Technical Report 2006-19, ITI Wagner, Faculty of Informatics, Universität Karlsruhe, 2006.

[3] L. Danon, J. Duch, A. D.-Guilera, and A. Arenas. Comparing community structure identification. *J. Stat. Mech*, page P09008, 2005.

[4] Y. R. Lin, J. Sun, P. Castro, R. Konuru, H. Sundaram, and A. Kelliher. Metafac: community discovery via relational hypergraph factorization. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 527–535, Paris, France, 2009.

[5] X. Liu and T. Murata. Advanced modularity-specialized label propagation algorithm for detecting communities in networks. *Physica A*, 389(7):1493–1500, 2010.

[6] C. Lu, X. Chen, and E. K. Park. Exploit the tripartite network of social tagging for web clustering. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, pages 1545–1548, Hong Kong, China, 2009.

[7] T. Murata. Detecting communities from tripartite networks. In *Proceedings of the 19th International Conference on World Wide Web*, pages 1159–1160, Raleigh, NC, USA, 2010.

[8] N. Neubauer and K. Obermayer. Towards community detection in k-partite k-uniform hypergraphs. In *Proceedings of the 22nd Annual Conference on Neural Information Processing Systems, Workshop on Analyzing Networks and Learning with Graphs*, Whistler, BC, Canada, 2009.

[9] M. E. J. Newman. *Networks: An Introduction*. Oxford University Press, New York, 2010.

[10] J. Rissanen. Modelling by shortest data description. *Automatica*, 14:465–471, 1978.

[11] M. Rosvall and C. T. Bergstrom. An information-theoretic framework for resolving community structure in complex networks. *Proc. Natl. Acad. Sci. USA*, 104(18):7327–7331, 2007.

[12] T. Zhou, J. Ren, M. Medo, and Y. C. Zhang. Bipartite network projection and personal recommendation. *Phys. Rev. E*, 76:046115, 2007.

[13] V. Zlatić, G. Ghoshal, and G. Caldarelli. Hypergraph topological quantities for tagged social networks. *Phys. Rev. E*, 80:036118, 2009.