COMMUNITY DETECTION USING RANDOM-WALK SIMILARITY AND APPLICATION TO IMAGE CLUSTERING

Makoto Okuda[⋆] *Shin'ichi Satoh*[⋆†] *Shoichiro Iwasawa*[⋆] *Shunsuke Yoshida*[⋆]
Yutaka Kidawara[⋆] *Yoichi Sato*[‡]

[⋆] National Institute of Information and Communications Technology
[†] National Institute of Informatics
[‡] The University of Tokyo

ABSTRACT

The technology used to detect community structures in graphs, or graph clustering technology, is important in a wide range of disciplines, such as sociology, biology, and computer science. Previously, many successful community detection methods have relied on the optimization of a quantity referred to as modularity, which is a quality index for the partition of a graph into communities. However, such methods suffer from a key drawback, namely, the inability to identify relatively small communities. To overcome this drawback, we propose a novel community detection method that can detect small communities. This is based on the property that a random walker will not readily leave a community even if it is small. The work presented in this paper demonstrates that our method detects both small and large communities in the practical application of clustering tourist attraction images obtained from Flickr.

Index Terms— Community detection, graph clustering, random walk, Jaccard index, image clustering

1. INTRODUCTION

Rapid progress in computer technology has fueled the analysis of big data in many professional disciplines, such as sociology, biology, and computer science. In this context, community detection technology, which locates subgraphs whose vertexes are more tightly connected with each other than with those lying outside the subgraphs, has become increasingly important.

Many successful methods have previously been proposed to detect communities [1–8]. Some methods [1–5] have relied on modularity optimization. However, such methods suffer from a resolution limit that may prevent them from detecting clusters that are relatively small compared with a whole graph [9]. Alternative methods [7, 8] may be able to alleviate this resolution limit. However, overcoming this resolution limit remains a significant topic.

Therefore, we have developed a novel community detection method, which can detect both large and small communities in a graph. Our method does not rely on modularity optimization, but rather employs the random walk technique, which is a mathematical formalization of a path that comprises a succession of random steps. The property that a random walker starting from a vertex tends to walk around in a community including that vertex for a short time period, even if the community may be small, promotes the identification of small communities.

In practical applications of the clustering of tourist attraction images from Flickr (an Internet photograph collection website), our method detects more small communities than previous methods and also exceeds their accuracy.

The remainder of this paper is organized as follows. Section 2 describes our community detection method. Section 3 presents the experiments in which we applied our method to clustering tourist attraction images by subject. Finally, Section 4 presents our conclusions and describes topics for future research.

2. RANDOM WALK SIMILARITY METHOD

In the graph of Fig. 1, whose explanation is detailed in Section 3, the goal of a community detection method is to obtain two clusters of vertexes, labeled Subject 1 and Subject 2. Random walkers who start from the vertexes labeled Subject 1 tend to walk around them for a period of time, because there are far fewer edges going from the vertexes labeled Subject 1 to those labeled Subject 2 than there are between the vertexes labeled Subject 1. Similarly, random walkers who start from the vertexes labeled Subject 2 tend to walk around these vertexes for a period of time. Our community detection method, which we call a random-walk similarity method, utilizes the histories of the vertexes passed by random walkers with such tendencies. The detailed procedure is explained in Algorithm 1.

In steps 3 through to 10 of Algorithm 1, we execute random walks starting from all vertexes $v_1,\ v_2, \ldots,\ v_l$ of the input graph and obtain sets $S_1,\ S_2, \ldots,\ S_l$ of the vertexes passed by the random walkers. Here, random walks starting

Algorithm 1 Random-walk similarity method.

1: Input a graph.
2: Let the number of graph vertexes be l.
3: **for** $i = 1$ to l **do**
4: Execute m step random walks n times starting from vertex v_i. The random walk is stopped if the walker cannot move to the next vertex.
5: Obtain sets $S_{i1},\ S_{i2}, \ldots,\ S_{in}$ of vertexes through which each walker passed.
6: **if** the number of a vertex v_k $(k = 1,\ 2, \ldots,\ l)$ included in sets $S_{i1},\ S_{i2}, \ldots,\ S_{in} < n \times threshold\_abnormal$ **then**
7: Delete v_k from the sets $S_{i1},\ S_{i2}, \ldots,\ S_{in}$
8: **end if**
9: $S_i \leftarrow S_{i1} \cup S_{i2} \cup \cdots \cup S_{in}$
10: **end for**
11: Calculate Jaccard similarity coefficients:

$$sim(S_i,\ S_j) \leftarrow \frac{|S_i \cap Sj|}{|S_i \cup S_j|}$$
$$(i = 1,\ 2, \ldots,\ l,\ j = 1,\ 2, \ldots,\ l)$$

12: **for** $i = 1$ to l **do**
13: **for** $j = i + 1$ to l **do**
14: **if** $sim(S_i,\ S_j) \geq threshold\_similarity$ **then**
15: Group vertex v_i and vertex v_j into the same cluster.
16: **end if**
17: **end for**
18: **end for**
19: Output clusters.

from each vertex are executed n times. The vertexes that are rarely passed by walkers are deleted in steps 6 to 8. In steps 11 to 18, vertexes $v_1,\ v_2, \ldots,\ v_l$ are clustered based on Jaccard similarity coefficients among $S_1,\ S_2, \ldots,\ S_l$.

Even if a community is small, the probability that a random walker starting from a vertex in that community moves to a different community is very low when we set m to be small in step 4. Therefore, $sim(S_i,\ S_j)$ in step 11 becomes large if vertexes i and j are in the same small community. As a result, the random-walk similarity method can detect small communities.

However, a random walker who starts from a vertex in a large community cannot walk around it sufficiently when m is small. Therefore, even if vertexes i and j are in the same community, $sim(S_i,\ S_j)$ can become too small. However, in steps 12 to 18, vertexes i and j are grouped into the same cluster with the other vertexes that have high similarities with i and j. This step is performed repeatedly. As a result, the random-walk similarity method can simultaneously detect small and large communities, unlike previous methods that have relied on modularity optimization [1–5].

The random-walk similarity method is applicable to weighted directed graphs. We can adjust the connectivity of the detected communities by a simple parameter $threshold\_similarity$ and easily obtain communities whose edges are denser or sparser.

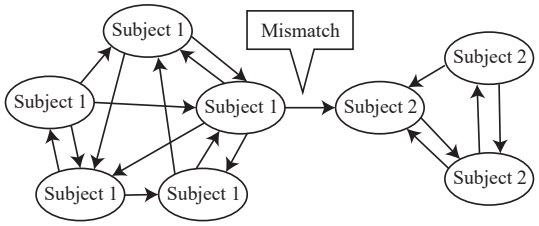

Fig. 1: Image of a connected component comprising subjects of two types. The edge arrows indicate that SIFT key points [12], which match the keypoints of images from which the arrows originate, are in images at which the arrows point.

Fig. 2: Example of mismatches between SIFT key points (yellow lines). Left: Great Buddha Hall of Todai-ji Temple (© Doug Urban). Right: Chumon Gate of Todai-ji Temple (© Travis)

3. EXPERIMENTS

Recently, methods for clustering images of tourist attractions by subject have been studied, because these are useful in a wide range of applications, including reconstructing 3-D scenes [10] and image recognition [11]. We assume that each connected component of a match graph [10], which is a graph representing a set of images with edges connecting pairs of images with matching key points, consists of images in which a particular subject appears. However, in our experiments, these connected components often comprise images of different subjects (Fig. 1), owing to mismatched identifications between key points (Fig. 2) or bridges between images containing different subjects created by images that contain both subjects (Fig. 3). We acquired image clusters in which one particular subject mostly appears (Fig. 4) by applying the community detection method to the impure connected components.

We believe that these experiments are appropriate for evaluating our community detection method, because some of the acquired connected components were the graphs comprising large and small communities.

(a) Left: Chumon Gate and Ungyo statue of Horyu-ji Temple (© Keith Rose). Right: Ungyo statue of Horyu-ji Temple (© Keith Rose).

(b) Left: Chumon Gate and Ungyo statue of Horyu-ji Temple (© Keith Rose). Right: Chumon Gate of Horyu-ji Temple (© Etsuko Nakamura).

Fig. 3: Example of bridges between images of different subjects. Images at upper and lower left, which are identical, served as bridges between Ungyo statue and Chumon Gate images.

3.1. Japanese Temple/Shrine Dataset

From Flickr, we obtained 4,015 Creative Commons (CC) licensed images of the Todai-ji Temple using the search term $todaiji$, 3,808 CC licensed images of the Nikko Tosho-gu Shrine using the search term $toshogu$, and 1,102 CC licensed images of the Horyu-ji Temple using $horyuji$. Each image was labeled by hand with the name of its primary subject. This dataset will be made available to the public in the near future.

3.2. Match Graph

Based on the work of Agarwal *et al.* [10], we constructed three match graphs with 3,515 images of the Todai-ji Temple, 3,308 images of the Nikko Tosho-gu Shrine, and 902 images of the Horyu-ji Temple, after using 500 Todai-ji Temple images, 500 Nikko Toshogu images, and 200 Horyu-ji Temple images to train the vocabulary trees. In constructing a match graph, we connected two images with an edge when they had matching SIFT (Scale-Invariant Feature Transform) key points [12]. Here, following the method of Lowe *et al.* [12], even when some SIFT key points in an image match those of another image, a converse match is not necessarily made. To account for the asymmetry, we made the edges of the match

(a) Great Buddha Hall of Todai-ji Temple. (Left: © rurinoshima, Middle: © Descubrir Japón, Right: © Jerich Abon)

(b) Vaiśravana statue of Todai-ji Temple. (Left: © Tim Brennan, Middle: © Wally Gobetz, Right: © Andrea Williams)

(c) Imagined elephants sculpture of Nikko Tosho-gu Shrine. (Left: © - J-I -, Middle: © Fran Simón, Right: © Dawn)

Fig. 4: Examples of images in clusters acquired by the proposed method.

graphs directed, as shown in Fig. 1. An edge's arrow indicates that the SIFT key points detected in an image behind the arrow match those detected in another image at that arrow's tip. We set the weight of each edge equal to one.

3.3. Clustering

We applied five community detection methods to 18 connected components of three match graphs containing images from the Japanese temple/shrine dataset. One was our proposed random-walk similarity (RW-sim) method. We chose the spin-glass method [1], the Infomap method [6], the CD-TRandwalk (scalable Community Detection based on Threshold Random walkers) method [7], and CONCLUDE (COmplex Network CLUster DEtection) method [8] for existing community detection methods. The spin-glass method detects communities through modularity optimization by a spin-glass model and simulated annealing. This method neglects the directions of the edges in match graphs. The Infomap method detects communities that minimize the expected description length of a random walker's trajectory. We chose these methods because Fortunato *et al.* [13] demonstrated that they yield the best results from many successful methods. The CD-TRandwalk method detects communities in which vertexes have many common neighbors through random walks. Fu *et al.* [7] showed that this method can detect many small communities through their experiments. The CONCLUDE method detects communities to maximize

Table 1: The number of clusters acquired by applying each clustering method to one of the connected components for the Todai-ji Temple for every major subject.

Major Subject (Number)	Spin-glass [1]	Infomap [6]	CD-TRandwalk [7]	CONCLUDE [8]	RW-sim (Proposed)
Great Buddha Hall (461)	1	8	3	11	1
Birushana Buddha statue (248)	3	8	1	6	2
Chumon Gate (100)	1	4	1	4	1
Kokuzo Bosatsu statue (54)	1	2	0	1	1
Vaiśravana statue (27)	0	0	0	0	1
Ungyo statue (19)	1	0	0	0	1

Table 2: Statistical evaluation of clusters acquired by applying each community detection method to a Japanese temple/shrine dataset.

	Spin-glass [1]	Infomap [6]	CD-TRandwalk [7]	CONCLUDE [8]	RW-sim (Proposed)
Number of detected subjects	19	15	17	12	**22**
Mean global purity	0.887	**0.940**	0.814	0.890	0.905
Mean inverse purity	0.854	0.612	0.749	0.227	**0.905**
Mean F-measure	0.860	0.704	0.758	0.339	**0.902**

modularity by combining the accuracy of global methods with the efficiency of local methods. This method has the advantage of mitigating the resolution limit.

We adjusted the parameters of each method to detect as many clusters as possible through all connected components. In step 4 of Algorithm 1, m was set to 50, and n was set to 100. When random walkers moved to the next vertex, an edge was selected with equal probability, because the weights of all edges were set to one. In step 6, $threshold\_abnormal$ was set to 0.2. In step 14, $threshold\_similarity$ is the most important parameter. When this parameter was set to take too small a value, images of different subjects were clustered into the same groups. When the parameter was set to large values, images of different subjects were separated into different clusters. When the parameter was set to an even larger value, images in which aspects such as the sizes and angles of subjects were quite different were separated into different clusters, even though the subjects were the same. When the parameter was set to 0.4, clustering by subject worked effectively.

After performing community detections with each method, we discarded tiny clusters with fewer than 15 elements for the Todai-ji Temple and Nikko Toshogu Shrine and those with fewer than 10 elements for the Horyu-ji Temple.

3.4. Results

Table 1 presents the number of clusters acquired by applying each community detection method to one of the connected components for the Todai-ji Temple for every major subject. The number of images in which each subject appears in the connected component is also mentioned in parenthesis.

The spin-glass and CD-TRandwalk methods could not detect comparatively small clusters, such as the Vaiśravana statue, because they were incorporated into large clusters, such as the Great Buddha Hall. The Infomap and CONCLUDE methods were also unable to detect small clusters, because such clusters were divided into tiny clusters with fewer than 15 images each, and these excessively small clusters were discarded according to the rule in Section 3.3. Moreover, the Infomap and CONCLUDE methods had the defect of dividing large clusters into multiple small ones. For example, the Infomap method yielded eight clusters for the Great Buddha Hall. However, it would be ideal to yield only one cluster, as achieved by the spin-glass and RW-sim methods.

The RW-sim method could simultaneously detect small clusters such as the Vaiśravana Statue and large clusters such as the Great Buddha Hall. Moreover, it came close to yielding one cluster per subject.

Table 2 presents a statistical evaluation of the clusters acquired by applying each community detection method to all connected components acquired from the Japanese temple/shrine dataset. The RW-sim method detected the most subjects, because it could detect comparatively small clusters that the other methods could not detect.

We calculated the global purity, inverse purity, and F-measure [14] for the acquired clusters for every connected component. The means of these are also presented in Table 2. The RW-sim method obtained the best F-measure owing to the superior balance between global purity and inverse purity.

4. CONCLUSIONS AND FUTURE WORKS

We developed a novel community detection method that employs the random walk technique. In the practical application of clustering tourist attraction images by subject, we confirmed that our community detection method simultaneously acquired large and small clusters from directed graphs more robustly than previous methods. Our method was also more accurate than previous methods. Future work will verify that this method can be effectively applied to larger datasets and those of other types.

5. REFERENCES

[1] J. Reichardt and S. Bornholdt, "Statistical mechanics of community detection," *Physical Review E*, vol. 74, no. 1, pp. 016110-1–016110-14, 2006.

[2] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical Review E*, vol. 69, no. 2, pp. 026113-1–026113-15, 2004.

[3] A. Clauset, M. E. J. Newman, and C. Moore, "Finding community structure in very large networks," *Physical Review E*, vol. 70, no. 6, pp. 066111-1–066111-6, 2004.

[4] M. E. J. Newman, "Finding community structure using the eigenvectors of matrices," *Physical Review E*, vol. 74, no. 3, pp. 036104-1–036104-19, 2006.

[5] P. Pons and M. Latapy, "Computing communities in large networks using random walks," *Journal of Graph Algorithms and Applications*, vol. 10, no. 2, pp. 191–218, 2006.

[6] M. Rosvall and C. T. Bergstrom, "The map equation," *The European Physical Journal Special Topics*, vol. 178, pp. 13–23, 2009.

[7] X. Fu, C. Wang, Z. Wang, and Z. Ming, "Threshold random walks for community structure detection in complex networks," *Journal of Software*, vol. 8, no. 2, pp. 286–295, 2013.

[8] P. D. Meo, E. Ferrara, G. Fiumara, and A. Provetti, "Mixing local and global information for community detection in large networks," *Journal of Computer and System Sciences*, vol. 80, no. 1, pp. 72–87, 2014.

[9] S. Fortunato and M. Barthelemy, "Resolution limit in community detection," in *Proceedings of the National Academy of Sciences of the United States of America*, 2007, vol. 104, pp. 36–41.

[10] S. Agarwal, N. Snavely, L. Simon, S. M. Seitz, and R. Szeliski, "Building Rome in a day," in *Proceedings of IEEE International Conference on Computer Vision*, 2009, pp. 72–79.

[11] T. Weyand and B. Leibe, "Visual landmark recognition from internet photo collections: A large-scale evaluation," *Computer Vision and Image Understanding*, vol. 135, pp. 1–15, 2015.

[12] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[13] S. Fortunato, "Community detection in graphs," *Physics Reports*, vol. 486, pp. 75–174, 2010.

[14] H. Hotho, A. Nürnberger, and G. Paaß, "A brief survey of text mining," *LDV-Forum - GLDV Journal for Computational Linguistics and Language Technology*, vol. 20, no. 1, pp. 19–62, 2005.