Journal of Information Processing Vol.28 427–435 (Aug. 2020)

[DOI: 10.2197/ipsjjip.28.427]

Regular Paper

New Attributed Graph Clustering by Bridging Attribute and Topology Spaces

Seiji Maekawa[1,a) Koh Takeuchi[2,b) Makoto Onizuka[1,c)

Received: December 7, 2019, Accepted: April 6, 2020

Abstract: We consider the clustering problem of attributed graphs. Our challenge is how we design an effective clustering method that captures the complicated relationship between the topology and the attributes in real-world graphs. We propose NAGC, a new attributed graph clustering method that bridges the attribute space and the topology space. The feature of NAGC is two-hold. 1) NAGC learns a projection function between the topology space and the attribute space so as to capture their complicated relationship, and 2) NAGC leverages the positive unlabeled learning to take the effect of partially observed positive edges into the cluster assignment. We conducted experiments extensively to validate that NAGC performs higher than or comparable to prior arts regarding the clustering quality.

Keywords: attributed graph clustering, NMF, PU learning

1. Introduction

Graph is a fundamental data structure for representing vertices and their relationships. Graph data appear everywhere in many application domains, such as web graph [7], social network [8], protein complexes [4], traffic planning [10], computer vision [15], and gene expressions [2], [23]. The authors of Ref. [31] conducted an online survey and showed that graph database is becoming increasingly prevalent across many application domains and, in particular, the graph clustering is the most widely used technique in the machine learning field.

Graphs in the real world usually have attributes on vertices. Actually, the graph databases support attributed graphs or property graphs [9], [32]. However, most of the graph clustering techniques [18], [27], [34] do not leverage the attributes of vertices since their design is limited to simple graphs without having attributes. Therefore, these techniques can not extract precise clusters without leveraging the attributes. There are emerging researches that tackle the clustering problem for attributed graphs [1], [14], [29], [35], [43] and the representation learning problem for attributed graphs, such as ANRL [41] and AANE [13]. Despite the considerable improvements made by those methods, they have not fully leveraged the virtue of attributed graphs. There are two fundamental aspects of the attributed graphs we should consider. First, the topology and the attributes of real-world graphs have complicated relationship with each other, because they are obtained from different viewpoints in real-world but these viewpoints correlate each other. Therefore, we need to balance the effects of the topology and the attributes for each cluster independently: some cluster takes larger effect from the topology and some other does from the attributes. Notice that ANRL and AANE use hyperparameters to control the effect of the topology to the attributes for all clusters, however, they can not control it for each cluster independently. This is because they simply propagate the attributes by random walk on graph. Second, typical graphs consist only of partially observed positive edges, which implies that there are missing positive edges, because real-world graphs follow the open world assumption: “absence of information is interpreted as unknown information, not as negative” [19]. For example, a social graph may not reflect precisely the social connections in the real world: we can only observe positive connections between people such as “likes” and “friendships”, but cannot observe negative ones [12].

We take the above two aspects into account and propose NAGC, a New Attributed Graph Clustering method by bridging the attribute space and topology space and by taking the effect of partially observed positive edges. To achieve high clustering quality, 1) NAGC learns a projection function between the topology space and the attribute space so as to capture their complicated relationship. The projection function consists of a rescale function and a transfer matrix that balances the effect from the attribute space to the topology space for each cluster independently, and 2) NAGC leverages PU (positive-unlabeled) learning [6], [12], [26] to take the effect of partially observed positive edges into the cluster assignment. To the best of our knowledge, our method is the first method that learns the representation of attributed graphs 1) by capturing the complex relationship between the topology and the attributes and 2) by applying the PU learning to the missing positive edges. Our method can precisely capture clustering results by revealing the relationship between the topol-

[1] Graduate School of Information Science and Technology, Osaka University, Suita, Osaka 565–0871, Japan

[2] Graduate School of informatics, Kyoto University, Kyoto 606–8501, Japan

a) maekawa.seiji@ist.osaka-u.ac.jp

b) koh.t@acm.org

c) onizuka@ist.osaka-u.ac.jp

ogy and the attributes in real-world graphs.

We extensively made experiments for various clustering methods and representation learning methods over various real datasets with ground truth. We also made a micro benchmark to validate the effectiveness of learning projection function and PU learning to the clustering quality. With these experiments, we confirm that our method performs higher than or comparable to the existing methods in terms of the clustering quality. In addition, we confirm that NAGC actually captures complicated relationships between attribute space and topology space by visualizing the transfer matrix. We also confirm that our method is stable against the hyperparameter selection.

The rest of this paper is organized as follows. We introduce fundamental techniques for our method, Non-negative Matrix Factorization, Symmetric Non-negative Matrix Factorization, and Biased Matrix Completion in Section 2. We propose our method in Section 3. Section 4 gives the purpose and results of the evaluations. Section 5 addresses the details of the related work and we conclude this paper at Section 6.

2. Preliminaries

Notation: We denote a matrix and its i-th row vector as upper boldface $\boldsymbol{X}$ and under boldface $\boldsymbol{x}_i$. The set of non-negative real numbers is $\mathbb{R}_+$. We denote a graph $G = (V, E)$ comprising a set of vertices $V = \{1, 2, \ldots, n\}$ and edges $E = \{(i, j)\} \subseteq [n] \times [n]$. We construct an weighted adjacency matrix $\boldsymbol{S} \in \mathbb{R}_+^{n\times n}$ from $\boldsymbol{G}$, where $s_{i,j}$ is set to a positive value if there is a edge between two vertices i and j or set to 0 otherwise. We denote a non-negative attribute matrix $\boldsymbol{X} \in \mathbb{R}_+^{n\times m}$ that represents n vertices with m attributes [*1]. $\|\cdot\|_{\mathcal{F}}$ and $\|\cdot\|_*$ are Frobenius norm and the nuclear norm, respectively. We use $\odot$ and $\oslash$ to denote element-wise multiplication and element-wise division, respectively.

2.1 Non-negative Matrix Factorization

Given the number of clusters for topology and attributes, $k_1, k_2 \ll \min\{m, n\}$, respectively, we suppose a cluster assignment matrix $\boldsymbol{U} \in \mathbb{R}_+^{n\times k_1}$ and an attribute factor matrix $\boldsymbol{V} \in \mathbb{R}_+^{m\times k_2}$. Let us denote a transfer matrix $\boldsymbol{H} \in \mathbb{R}_+^{k_1\times k_2}$ that represents the relationship between topology and attributes. Non-negative Matrix Tri-Factorization (NMTF) [5], which is a novel extension of Non-negative Matrix Factorization (NMF) [24], estimates local optimal parameters $\boldsymbol{U}$, $\boldsymbol{V}$, and $\boldsymbol{H}$ by minimizing a non-convex loss:

$$\min_{\boldsymbol{U},\boldsymbol{V},\boldsymbol{H}\geq 0} \|\boldsymbol{X} - \boldsymbol{U}\boldsymbol{H}\boldsymbol{V}^\top\|_{\mathcal{F}}^2. \tag{1}$$

NMF is treated as a special case of NMTF where $\boldsymbol{H}$ is set to an identity matrix. Compared with the original NMF, which estimates $\boldsymbol{U}$, $\boldsymbol{V}$, NMTF generates more precise model by introducing transfer matrix $\boldsymbol{H}$. However, NMTF is limited to consider only linear relationships between topology and attributes.

2.2 Symmetric Non-negative Matrix Factorization

The goal of graph clustering is to find a partition of vertices in a graph where the similarity between vertices is high within the same cluster and low across different clusters. Kuang et al. proposed Symmetric Non-negative Matrix Factorization (SNMF) [21], [22], and showed an interesting relationship among SNMF and graph clustering methods [28]. SNMF estimates a cluster assignment matrix $\boldsymbol{U}$ by minimizing a non-convex loss function that uses an adjacency matrix $\boldsymbol{S}$ as input:

$$\min_{\boldsymbol{U}\geq 0} \|\boldsymbol{S} - \boldsymbol{U}\boldsymbol{U}^\top\|_{\mathcal{F}}^2. \tag{2}$$

Thanks to the non-negative constraint, we can obtain a clustering result by assigning i-th vertex to the k_1'-th cluster that has the largest value in $\boldsymbol{u}_i$, that means $k_1' = \mathrm{argmax}_l\{u_{i,l} \mid l = (1, \ldots, k)\}$. We don't need to apply additional clustering techniques such as k-means to the vertex vectors.

2.3 Biased Matrix Completion

Hsieh et al. [12] considered a matrix completion problem when only a subset of positive relationships is observed, such as recommender systems and social networks where only "likes" or "friendships" are observed. The problem is an instance of PU learning, i.e. learning from only positive and unlabeled examples that has been studied in the classification problems. They introduced the ρ-weighted loss for a bipartite graph $G' = (V', E')$ comprising a set of vertices $V' = \{\{1, 2, \ldots, n\}, \{1, 2, \ldots, m\}\}$ and edges $E' = \{(i, j)\} \subseteq [n] \times [m]$:

$$\ell_\rho(z_{i,j}) = \rho 1_{(i,j)\in E'}(z_{i,j} - 1)^2 + (1 - \rho)1_{(i,j)\notin E'} z_{i,j}^2, \tag{3}$$

where $\rho = [0, 1]$, $1_{(i,j)\in E'}(\cdot)$, and $1_{(i,j)\notin E'}(\cdot)$ are a bias weight, an indicator function for positive edges, and an indicator function for unlabeled edges, respectively. This loss can change a weight for reconstruction errors among positive and unlabeled edges. When we set $\rho = 0.5$, it treats the positive and unlabeled entities equally. With this loss, they proposed a biased matrix completion as:

$$\min_{\boldsymbol{Z}:\|\boldsymbol{Z}\|_*\leq\lambda} \sum_{(i,j)\in E'} \rho(z_{i,j} - 1)^2 + \sum_{(i,j)\notin E'} (1 - \rho) z_{i,j}^2. \tag{4}$$

where $\lambda \geq 0$ is a hyperparameter.

3. NAGC: New Attribute Graph Clustering

As we mentioned in Section 1, we need to consider two fundamental aspects of the attributed graphs: 1) the topology and the attributes of real-world graphs have a complicated relationship and 2) typical graphs usually have a subset of positive edges implying that there are missing positive edges. The novelty of NAGC is three-hold:

- We jointly decompose the topology (adjacency) matrix and the attribute matrix into factor matrices to represent the vertex features in the topology space and the attribute space. We employ SNMF and NMTF to decompose adjacency matrix and attribute matrix, respectively.
- NAGC learns a projection function between the topology space and the attribute space. The projection function consists of a rescale function and a cluster assignment transfer matrix. The rescale function bridges the scale gap between the two spaces. The transfer matrix bridges the two spaces by balancing the effect from the attribute space to the topology space for each cluster independently.

[*1] We can convert the domain of matrix elements into positive one when they have some negative values.

Table 1 Definition of main symbols.

Variable	Explanation
$\boldsymbol{S} \in \mathbb{R}_+^{n\times n}$	adjacency matrix
$\boldsymbol{X} \in \mathbb{R}_+^{n\times m}$	attribute matrix
$\boldsymbol{U} \in \mathbb{R}_+^{n\times k_1}$	topology cluster assignment matrix
$\boldsymbol{V} \in \mathbb{R}_+^{m\times k_2}$	attribute factor matrix
$\boldsymbol{H} \in \mathbb{R}_+^{k_1\times k_2}$	cluster assignment transfer matrix
$\boldsymbol{W} \in \mathbb{R}_+^{n\times n}$	mask matrix of $\boldsymbol{S}$
$k_1 \in \mathbb{N}$	number of clusters for topology
$k_2 \in \mathbb{N}$	number of clusters for attributes
$\lambda \geq 0$	balancing parameter between the topology and the attributes
$\rho = [0, 1]$	bias weight for $\boldsymbol{S}$
$t \in \mathbb{N}$	number of iterations

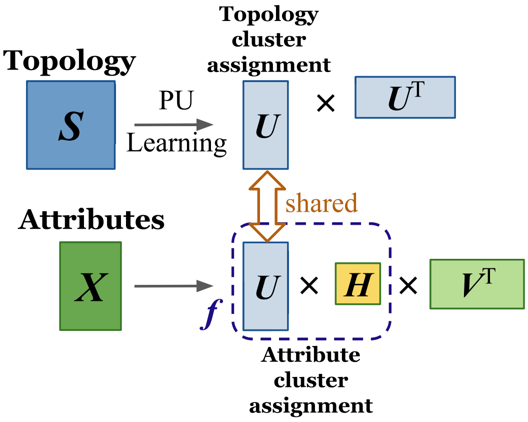


Fig. 1 Illustration of NAGC. $\boldsymbol{S}$ and $\boldsymbol{X}$ are an adjacency matrix and an attributed matrix, respectively. $\boldsymbol{U}$, $\boldsymbol{V}$, and $\boldsymbol{H}$ denote a topology cluster assignment, an attribute factor, and a cluster assignment transfer matrices, respectively. f is a rescale function.

- We leverages PU learning to take the effect of partially observed positive edges into the cluster assignment. That is, we put larger bias to partially observed positive edges than unlabeled edges.

Table 1 lists the main symbols and their definitions.

3.1 NAGC Model

We formalize our method as a minimization problem of a non-convex loss as follows:

$$\min_{\boldsymbol{U},\boldsymbol{V},\boldsymbol{H}\geq 0} \mathcal{L}_\rho(\boldsymbol{S} - \boldsymbol{U}\boldsymbol{U}^\top) + \frac{\lambda}{2}\|\boldsymbol{X} - f(\boldsymbol{U}\boldsymbol{H})\boldsymbol{V}^\top\|_{\mathcal{F}}^2. \tag{5}$$

Figure 1 depicts the design of NAGC. The adjacency matrix $\boldsymbol{S}$ is decomposed into $\boldsymbol{U}\boldsymbol{U}^\top$, so $\boldsymbol{U}$ represents the matrix of the topology cluster assignment. In contrast, the attribute matrix $\boldsymbol{X}$ is decomposed into $f(\boldsymbol{U}\boldsymbol{H})\boldsymbol{V}^\top$, so $f(\boldsymbol{U}\boldsymbol{H})$ represents the matrix of the attribute cluster assignment. By transforming the matrix of the topology cluster assignment ($\boldsymbol{U}$) to the matrix of the attribute cluster assignment ($f(\boldsymbol{U}\boldsymbol{H})$) with the rescale function f and transfer matrix $\boldsymbol{H}$, our method enables to capture the complex relationship among the topology cluster assignment and the attribute cluster assignment. In particular, $\boldsymbol{H}$ bridges the two assignments by balancing the effect from the attribute space to the topology space for each cluster independently. In addition, λ is a hyperparameter that balances globally the effects between the topology and the attribute for all clusters. f denotes an element-wise rescale function and we use the sigmoid function as f for simplicity: $f(x) = \frac{1}{1+e^{-x}}$. There are two reasons we adopt the sigmoid function. First, the sigmoid function is differentiable so the parameter update rules of the existing method can be used with minor modifications. Second, the sigmoid function is one of the simplest functions for rescale function.

This choice can be generalized to any non-linear functions. We use $\mathcal{L}_\rho(\boldsymbol{Z})$ to denote an error of the adjacency matrix $\boldsymbol{S}$ with ρ-weighted loss.

$$\mathcal{L}_\rho(\boldsymbol{Z}) = \sum_{(i,j)\in E} \rho(z_{i,j} - 1)^2 + (1-\rho)\sum_{(i,j)\notin E} z_{i,j}^2. \tag{6}$$

Note that, the number of clusters k_2 for attribution is not necessary the same as the number of clusters k_1 for topology, since we suppose that the cluster structure embedded in attributes differs from that in topology. NAGC can be seen as a generalized SNMF, because NAGC simulates SNMF by setting $\lambda = 0$ and $\rho = 0.5$: no effects from attributes and partially observed positive edges.

3.2 Optimization

Since our loss is non-convex for $\boldsymbol{U}$, $\boldsymbol{V}$, and $\boldsymbol{H}$, we derive a parameter estimation procedure that alternately updates each parameter by utilizing the method of Lagrange multipliers [5]. Following the standard theory of constrained optimization, we introduce Lagragian multipliers $\boldsymbol{P} \in \mathbb{R}^{n\times k_1}$, $\boldsymbol{Q} \in \mathbb{R}^{m\times k_2}$, and $\boldsymbol{R} \in \mathbb{R}^{k_1\times k_2}$ for the non-negative constraints $\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{H} \geq 0$. We define the Lagrangian function of our proposed method as:

$$\begin{aligned}&\mathcal{L}(\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{H}; \boldsymbol{P}, \boldsymbol{Q}, \boldsymbol{R})\\ &= \mathcal{L}_\rho(\boldsymbol{S} - \boldsymbol{U}\boldsymbol{U}^\top) + \frac{\lambda}{2}\|\boldsymbol{X} - f(\boldsymbol{U}\boldsymbol{H})\boldsymbol{V}^\top\|_{\mathcal{F}}^2\\ &\quad + \mathrm{Tr}(\boldsymbol{P}^\top\boldsymbol{U}) + \mathrm{Tr}(\boldsymbol{Q}^\top\boldsymbol{V}) + \mathrm{Tr}(\boldsymbol{R}^\top\boldsymbol{H}).\end{aligned} \tag{7}$$

For each parameter, we derive partial differences of $\mathcal{L}$.

$$\begin{aligned}\frac{\partial\mathcal{L}}{\partial\boldsymbol{U}} &= -2\rho\boldsymbol{S}\boldsymbol{U} - \lambda\{(\boldsymbol{X}\boldsymbol{V}) \odot f'(\boldsymbol{U}\boldsymbol{H})\}\boldsymbol{H}^\top\\ &\quad + 2\rho(\boldsymbol{U}\boldsymbol{U}^\top \odot \boldsymbol{W})\boldsymbol{U} + 2(1-\rho)(\boldsymbol{U}\boldsymbol{U}^\top \odot \boldsymbol{W}')\boldsymbol{U}\\ &\quad + \lambda[\{f(\boldsymbol{U}\boldsymbol{H})\boldsymbol{V}^\top\boldsymbol{V}\} \odot f'(\boldsymbol{U}\boldsymbol{H})]\boldsymbol{H}^\top + \boldsymbol{P}.\end{aligned} \tag{8}$$

$$\frac{\partial\mathcal{L}}{\partial\boldsymbol{V}} = -\lambda\boldsymbol{X}^\top f(\boldsymbol{U}\boldsymbol{H}) + \lambda\boldsymbol{V}f(\boldsymbol{U}\boldsymbol{H})^\top f(\boldsymbol{U}\boldsymbol{H}) + \boldsymbol{Q}. \tag{9}$$

$$\begin{aligned}\frac{\partial\mathcal{L}}{\partial\boldsymbol{H}} &= -\lambda\boldsymbol{U}^\top\{f'(\boldsymbol{U}\boldsymbol{H}) \odot (\boldsymbol{X}\boldsymbol{V})\}\\ &\quad + \lambda\boldsymbol{U}^\top\{f'(\boldsymbol{U}\boldsymbol{H}) \odot f(\boldsymbol{U}\boldsymbol{H})\}\boldsymbol{V}^\top\boldsymbol{V} + \boldsymbol{R}.\end{aligned} \tag{10}$$

$\boldsymbol{W} \in \mathbb{R}_+^{n\times n}$ is a mask matrix whose elements are set to $w_{i,j} = 1$ if $s_{i,j} \neq 0$ or $w_{i,j} = 0$ otherwise, and $\boldsymbol{W}' = 1 - \boldsymbol{W}$. The KKT complementarity conditions are: $\boldsymbol{P} \odot \boldsymbol{U} = 0, \boldsymbol{Q} \odot \boldsymbol{V} = 0, \boldsymbol{R} \odot \boldsymbol{H} = 0$, $\frac{\partial\mathcal{L}}{\partial\boldsymbol{U}} = 0$, $\frac{\partial\mathcal{L}}{\partial\boldsymbol{V}} = 0$, and $\frac{\partial\mathcal{L}}{\partial\boldsymbol{H}} = 0$. By satisfying these conditions, we can derive multiplicative update rules.

$$\begin{aligned}\boldsymbol{U} \leftarrow \boldsymbol{U} &\odot [2\rho\boldsymbol{S}\boldsymbol{U} + \lambda\{(\boldsymbol{X}\boldsymbol{V}) \odot f'(\boldsymbol{U}\boldsymbol{H})\}\boldsymbol{H}^\top] \oslash\\ &[2\rho(\boldsymbol{U}\boldsymbol{U}^\top \odot \boldsymbol{W})\boldsymbol{U} + 2(1-\rho)(\boldsymbol{U}\boldsymbol{U}^\top \odot \boldsymbol{W}')\boldsymbol{U}\\ &+ \lambda\{(f(\boldsymbol{U}\boldsymbol{H})\boldsymbol{V}^\top\boldsymbol{V}) \odot f'(\boldsymbol{U}\boldsymbol{H})\}\boldsymbol{H}^\top].\end{aligned} \tag{11}$$

$$\boldsymbol{V} \leftarrow \boldsymbol{V} \odot \{\boldsymbol{X}^\top f(\boldsymbol{U}\boldsymbol{H})\} \oslash \{\boldsymbol{V}f(\boldsymbol{U}\boldsymbol{H})^\top f(\boldsymbol{U}\boldsymbol{H})\}. \tag{12}$$

$$\begin{aligned}\boldsymbol{H} \leftarrow \boldsymbol{H} &\odot [\boldsymbol{U}^\top\{f'(\boldsymbol{U}\boldsymbol{H}) \odot (\boldsymbol{X}\boldsymbol{V})\}]\\ &\oslash [\boldsymbol{U}^T\{f'(\boldsymbol{U}\boldsymbol{H}) \odot f(\boldsymbol{U}\boldsymbol{H})\}\boldsymbol{V}^\top\boldsymbol{V}].\end{aligned} \tag{13}$$

Algorithm 1 NAGC-U algorithm

```
Input: $\boldsymbol{S}, \boldsymbol{X}, k_1, k_2, \lambda, t$
Output: clustering result $C$
Preprocess: $\boldsymbol{S}, \boldsymbol{X}$
Initialize: $\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{H}$
while $t' < t$ do
    # alternately update parameters
    $\boldsymbol{U}^{(t'+1)} \leftarrow$ update $(\boldsymbol{U}^{(t')})$ by Eq. (11)
    $\boldsymbol{V}^{(t'+1)} \leftarrow$ update $(\boldsymbol{V}^{(t')})$ by Eq. (12)
    $\boldsymbol{H}^{(t'+1)} \leftarrow$ update $(\boldsymbol{H}^{(t')})$ by Eq. (13)
end while
while $n' < n$ do
    # assign each vertex to the clusters
    $c_{n'} \leftarrow \mathrm{argmax}_l\{u_{n',l} \mid l = (1, \ldots, k_1)\}$
end while
```

Our loss is convex with respect to $\boldsymbol{V}$ and $\boldsymbol{H}$, however, as mentioned in Ref. [21], the loss is a fourth-order non-convex function with respect to $\boldsymbol{U}$. That means, it is difficult to guarantee the monotonic convergence of our parameter estimation method; thus we expect a good convergence property that every limit point is a stationary point.

Algorithm 1 shows the algorithm for our method. Since non-convex minimization problems have multiple local minima, we apply k-means to the attribute matrix $\boldsymbol{X}$ and use the result to initialize $\boldsymbol{U}$ and $\boldsymbol{V}$. $\boldsymbol{H}$ is initialized by random values in the same way as the standard NMTF. There are two variations of our method, NAGC-U and NAGC-UH. They obtain clusters based on the topology cluster assignment $\boldsymbol{U}$ and attribute cluster assignment $\boldsymbol{UH}$, respectively.

3.3 Computational Complexity

Let t be the number of iterations in the matrix decomposition. Since the cost for SNMF is $O(n^2kt)$ [21], [22], the cost for updating rules of NAGC is equal to $O((n^2 + mn)kt)$ where $k = \max(k_1, k_2)$ and $k \ll n$ in general. For example, the operation of NAGC finishes in 3 seconds for WebKB which is a small size dataset, in 60 seconds for Citeseer which is a middle size dataset, and in 250 seconds for Flickr which is a large size dataset [*2].

4. Experiments

The purpose of our experiments is to answer the following questions:

Q1 Does NAGC perform higher than former methods? (Section 4.4)

Q2 Does NAGC capture the complicated relationship between the topology and the attributes? (Section 4.5)

Q3 How largely the parameters affect the performance? (Section 4.6)

In detail, the first purpose of the experiments is to evaluate the clustering quality of NAGC [*3] compared with various methods: representation learning methods for attributed graphs (AANE [13], ANRL [41]), an attributed graph clustering method (JWNMF [14]), graph clustering methods without using attributes (METIS [18], DANMF [38]), and a typical attribute-based clustering method (k-means). We used publicly available codes for those methods. As for the representation learning methods, we learned the vertex representation by using the same setting used in each paper. Then, we obtain clustering results by applying k-means to the learned representation by taking the same approach used in Refs. [11], [38]. We also evaluate simple graph clustering methods without using attributes, METIS [17], and attribute-based clustering methods, k-means, so that how much only the topology or attributes of the graphs contribute to the clustering quality. We use two variations of our method, NAGC-U and NAGC-UH, based on the topology cluster assignment and the attribute cluster assignment, respectively. We perform five restarts for each method and report the average of the results for all the above experiments.

The second purpose is to evaluate how effectively the transfer matrix bridges the two spaces to capture their complicated relationship, because the transfer matrix is designed to balance the effect from the attribute space to the topology space for each cluster independently.

The third purpose of our experiments is to investigate the details of the quality improvement achieved by our method: we evaluate the effectiveness of PU learning and the effect of the hyperparameters.

4.1 Datasets

We choose seven real-world datasets with ground truth in our experiments. They cover wide variety of graph types and sizes. They are used in the related papers of the attributed graph clustering. The graph types of our datasets (web graph, blogs, Wikipedia, citation networks, social network) cover more than half of the categories used in SNAP [*4] graph data archive. Also, the graph sizes are from small to large (the number of nodes from 877 to 7,564 and the number of edges from 1,480 to 239,365).

- WebKB [*5] is a web graph of four universities: the label for a vertex indicates the owner university of the page. The attributes of a vertex represent the words appeared in the page.
- Polblog [*6] is a network of hyperlinks between blogs on US politics: the label of a vertex indicates whether the blog is liberal or conservative. The attributes of a vertex represent the sources of the blogs.
- Wiki is a document network and the link among different vertices is the hyperlink in a web page. The attributes represent the TFIDF matrix of this datasets.
- Citeseer and Cora [*5] are citation networks. The label of a vertex corresponds to a research field of the paper. The attributes of a vertex consist of the words appeared in the paper.
- BlogCatalog is a blogger community network, where users interact with each other. The attributes of a vertex represent keywords of their blogs.
- Flickr is an online community that people can share photos

[*2] The experiments are implemented on Python3.

[*3] The source code of NAGC is available at https://github.com/seijimaekawa/NAGC.

[*4] Stanford Large Network Dataset Collection: https://snap.stanford.edu/data/index.html

[*5] http://linqs.cs.umd.edu/projects/projects/lbc/index.html

[*6] http://www-personal.umich.edu/˜mejn/netdata/

Table 3 Clustering performance of different datasets. The boldface font represents the best performance for each dataset.

Dataset		WebKB	Polblog	Wiki	Cora	Citeseer	BlogCatalog	Flickr	Avg.
ARI	NAGC-U	**0.992**	**0.646**	0.315	0.336	0.269	0.200	0.136	**0.413**
	NAGC-UH	0.802	0.100	**0.367**	0.360	**0.303**	**0.259**	**0.141**	0.333
	AANE	0.974	0.003	0.137	0.221	0.184	0.174	0.086	0.254
	ANRL	0.990	0.000	0.212	**0.439**	0.296	0.252	0.053	0.320
	JWNMF	0.908	0.513	0.132	0.309	0.077	0.149	0.133	0.317
	DANMF	0.850	0.556	0.174	0.249	0.084	0.129	0.067	0.301
	METIS	0.909	0.575	0.187	0.246	0.155	0.143	0.069	0.326
	k-means	0.274	0.000	0.040	0.062	0.169	0.000	0.000	0.078
AMI	NAGC-U	**0.987**	**0.547**	0.440	0.374	0.266	0.260	0.199	**0.439**
	NAGC-UH	0.742	0.078	**0.459**	0.404	0.290	0.310	0.168	0.351
	AANE	0.962	0.006	0.447	0.336	0.216	0.289	0.172	0.347
	ANRL	0.984	0.000	0.378	**0.491**	**0.349**	**0.337**	0.095	0.376
	JWNMF	0.899	0.442	0.260	0.232	0.087	0.215	**0.202**	0.334
	DANMF	0.853	0.484	0.292	0.334	0.133	0.194	0.105	0.342
	METIS	0.889	0.471	0.299	0.336	0.173	0.186	0.105	0.351
	k-means	0.292	0.000	0.174	0.117	0.207	0.005	0.001	0.114

Table 2 The statistics of the datasets. Mod. and Ent. indicate the modularity and the average entropy, respectively.

Dataset	Vertex	Edge	Attribute	Label	Mod.	Ent.
WebKB	877	1,480	1,703	4	0.739	0.152
Polblog	1,490	16,630	7	2	0.405	0.379
Wiki	2,405	12,761	4,973	17	0.524	0.320
Cora	2,708	5,278	1,433	7	0.640	0.054
Citeseer	3,312	4,660	3,703	6	0.544	0.039
BlogCatalog	5,196	171,743	8,189	6	0.224	0.036
Flickr	7,564	239,365	12,047	9	0.121	0.012

and follow each other. We use the tags attached on each image as the attribute information.

Table 2 summarizes the statistics of the datasets. We also include the modularity [27] and the average entropy for each of the true cluster assignment: the modularity and the entropy represent the topological aspect and attribute aspect, respectively. Intuitively, higher modularity indicates there are dense connections in the same cluster but sparse connections between different clusters. Lower average entropy indicates there are similar attribute values in the same cluster but dissimilar attribute values between different clusters. Average entropy is defined as:

$$Average\_entropy = \sum_{i=1}^{m} \sum_{j=1}^{k} \frac{|C_j|}{nm} entropy(a_i, C_j) \qquad (14)$$

where $entropy(a_i, C_j)$ is the information entropy of attribute a_i in cluster C_j. The values with respect to the modularity and the average entropy fall within the range of $[-1, 1]$ and the range of $[0, 1]$, respectively.

4.2 Measurements

The modularity and entropy are not suitable measurements for the clustering evaluation of attributed graphs, because the resulting clusters should take into account both aspects of the topology and attributes. We utilize two measures, Adjusted Rand Index (ARI) [39] and Adjusted Mutual Information (AMI). AMI is an adjusted version of Normalized Mutual Information (NMI). They are typical measurements used for assessing the clustering quality with ground truth labels [*7]. They are adjusted in a sense that random cluster assignments make ARI and AMI scores close to zero. On the other hand, non-adjusted measures such as NMI have a dependency between the number of clusters and the number of samples used to compute the measure. Therefore, the adjusted measures are more preferable for cluster evaluation.

4.3 Parameter Settings

We searched for optimum parameters, λ, k_2, and ρ for each dataset and used them in our experiments. λ is chosen from the set $\{10^{-10}, 10^{-8}, 10^{-7}, 10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 0.1, 1, 10, 100, 1000\}$ by following the settings used in Ref. [14]. The model does not work well when $\lambda > 100$.

Let k be the number of true clusters for each dataset. k_2 is chosen from the set $\{k, 5, 7, 10, 15, 20\}$ for NAGC-U [*8] so that we can learn the model more precisely than when we use k. ρ is chosen from the set $\{0.5, 0.55, 0.75, 0.95, 0.995\}$. To mitigate the different scales between $\boldsymbol{S}$ and $\boldsymbol{X}$, we normalize $\boldsymbol{S}$ by multiplying each element of $\boldsymbol{S}$ with $\frac{|\boldsymbol{X}|}{|\boldsymbol{S}|}$. The iterate computation of our method converges very fast (usually in 100 iterations) so the number of the iterations t is fixed at 100 in all the experiments.

4.4 Clustering Quality

Table 3 shows the results of evaluating the clustering quality. NAGC-U is obtained from the topology cluster assignment and NAGC-UH is obtained from the attribute cluster assignment. The last column (Avg.) indicates the average for all datasets. NAGC achieves the best performance not only in the average results, but also in six datasets (out of seven) in ARI measurement and three datasets in AMI measurement. The benefit of NAGC is that it balances and combines the effects of both the topology and the attributes, as we can see that NAGC is always better than METIS and k-means. Moreover, NAGC generally works well regardless of the entropy of the datasets (see Table 2). In particular, NAGC performs better than other methods even when the entropy is large (WebKB, Polblog, Wiki). NAGC also performs best when the attributes do not effectively contribute to the clustering result, such as when k-means works poorly (Flickr) [*9]. This behavior implies

[*7] We choose AMI since NMI is not adjusted for chance. Note that the chance rates of ARI and AMI are 0.

[*8] We do the same by replacing k_2 with k_1 for NAGC-UH.

[*9] k-means decides the centroids of clusters by treating all nodes equally so it does not work well when most elements of the attribute matrix are zero. Actually, 99% of the nodes are assigned to a single cluster in Flickr.

Table 4 Modularity and average entropy for WebKB dataset.

	Modularity	Entropy	ARI
NAGC-U	0.737	0.152	**0.992**
AANE	0.731	0.152	0.974
ANRL	0.737	0.152	0.990
JWNMF	**0.741**	0.153	0.908
DANMF	0.718	0.153	0.850
METIS	**0.741**	0.153	0.909
k-means	0.252	**0.146**	0.274

that NAGC selectively chooses the effect from the attribute space to the topology space for each cluster independently.

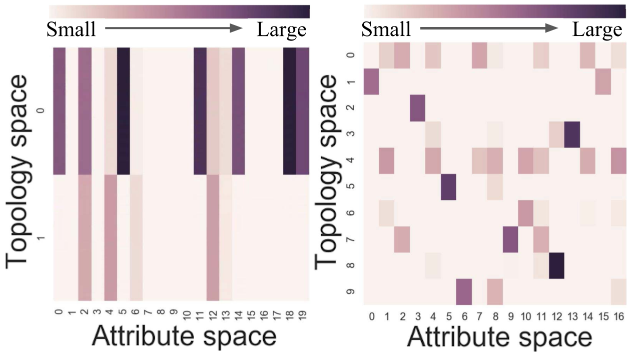
In contrast, ANRL generally works well when the entropy is small (Cora, Citeseer, BlogCatalog) but not otherwise[*10]. ANRL learns the representation equally from all the attributes, so it does not control the effect of each attribute cluster independently to the topology cluster assignment: ANRL is even worse than METIS for Polblog and Flickr.

To investigate more on the difficulty of the attributed graph clustering, we show that the topology and the attributes of real-world graphs have different cluster assignments. **Table 4** gives the modularity and the average entropy for the clustering result of WebKB. Our method achieves the highest ARI but does not achieve either the best modularity or the best average entropy. This result implies that we should not optimize the model only to either the topology or the attributes, but balance the effects between the topology and the attributes.

4.5 Bridging Topology and Attribute Spaces

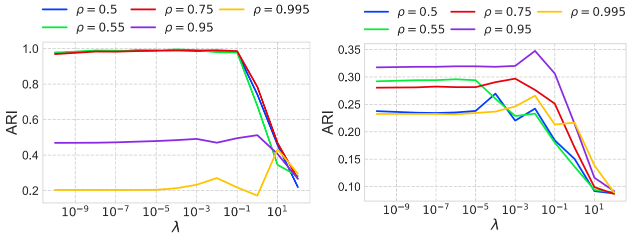
One of the most important contributions is how effectively NAGC captures the complex relationship between the topology space and the attribute space. **Figure 2** depicts the heatmaps of the transfer matrix $\boldsymbol{H}$ for Polblog and Wiki. We choose these datasets in which NAGC-U and NAGC-UH achieve the best results for both ARI and AMI. X and Y axes depict attribute clusters and topology clusters, respectively. The darker elements indicate there is larger effect from attribute cluster to topology cluster. In detail, in Fig. 2 (a), most attribute clusters (1, 3, 7–10, 15–17) are coloured in light colour, this indicates that there is almost no effect from those attribute clusters to topology clusters. In other words, those attribute clusters do not have any correlations with topology clusters. Considering a web graph with word information as an example, there is no correlation between general words (such as “abstract” or “introduction” for academic papers) and topology clusters. In contrast, the attribute cluster of 0, 5, 11, 18, 19 indicate theses attribute clusters effect mostly to topology cluster 0. In Fig. 2 (b), the dark elements in the matrix, such as (5, 5), (12, 8), (13, 3), show clear effect from an attribute cluster to a topology cluster. Also, we can observe that the topology cluster of 4 receives almost equal effects from multiple attribute cluster of 1, 4, 7, 8, 10, 11, 14, 16.

These results validate that there is actually a complex relationship between topology space and attribute space and justify our motivation: we should learn a projection function between the topology space and the attribute space.



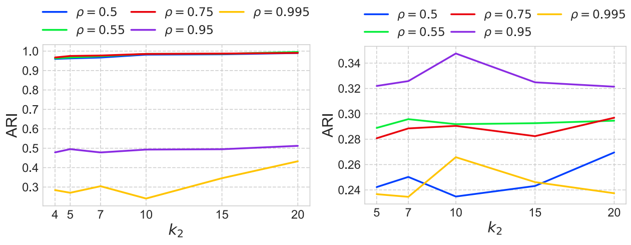
(a) Polblog (k_1=2, k_2=20) (b) Wiki (k_1=10, k_2=17)

Fig. 2 Heatmaps of the transfer matrix $\boldsymbol{H}$. The darker elements indicate there is larger effect from attribute cluster to topology cluster.



(a) WebKB (b) Cora

Fig. 3 Effect of λ and ρ on ARI in our method for two datasets.



(a) WebKB (b) Cora

Fig. 4 Effect of k_2 and ρ on ARI in our method for two datasets.

4.6 Hyperparameter Analysis

We discuss the effect of the hyperparameters of our method. **Figure 3** shows the effect of λ to the clustering results. Other parameters are fixed at the values when ARI becomes highest for each λ. There is a peak in each dataset ($\lambda = 10^{-4}$ on WebKB and $\lambda = 10^{-2}$ on Cora) which indicates that the effect to the model is well balanced by λ between the topology and the attributes[*11].

The effect of k_2 and ρ to ARI is shown in **Fig. 4**. Figure 4 (a) shows that ARI slightly increases when k_2 increases. ARI of the WebKB is enough high (almost 1.0) when $k_2 = 20$. Figure 4 (b) shows that there is a peak of ARI on Cora when $\rho = 0.95$ and $k_2 = 10$. From Figs. 3 and 4, we confirmed that ARI is stable against the selection of λ (when $\lambda < 0.1$) and k_2 in a wide range. Thus, in practice, we suppose our method would perform well when λ and k_2 may be simply chosen e.g., $\lambda = 0.01$ and $k_2 = k_1$.

[*10] The average entropy tends to be low when most elements of the attribute matrix are zero regardless of the correlation between the attributes and the true label. Considering the summation of all elements of the attribute matrix divided by the numbers of nodes and attributes, Flickr has the smallest value, 0.0280. It is much smaller than the average value of all datasets, 0.128. For this reason, Flickr has low entropy.

[*11] Even when λ is close to 0, the attributes contribute to the clustering results so the clustering performance is kept high. When lambda=0, we observe that ARI and AMI largely decrease.

Table 5 Effect of ρ on ARI achieved by our method. The density indicates the (# of partially observed positive edges)/(# of all possible edges), that is $|E|/\ n^2$.

Dataset	WebKB	Polblog	Cora	Citeseer
Density	0.18%	0.75%	0.07%	0.04%
$\rho = 0.5$	0.990	0.621	0.270	0.221
$\rho = 0.55$	**0.992**	0.625	0.296	0.216
$\rho = 0.75$	0.991	0.625	0.297	0.229
$\rho = 0.95$	0.512	**0.646**	**0.336**	0.254
$\rho = 0.995$	0.433	0.529	0.266	**0.269**

Table 6 NMI score for NAGC-U and NAGC-UH

Dataset	WebKB	Polblog	Wiki	Cora	Citeseer	BlogCatalog	Flickr
NMI (U,UH)	0.834	0.000	0.735	0.838	0.941	0.866	0.428

As for the hyperparameter of PU learning, ρ has a large influence on the performance of our method as shown in Figs. 3 and 4. Note that, when $\rho = 0.5$, PU learning is not applied because the weights for 0 and 1 are treated as the same. To evaluate the effectiveness of the Positive Unlabeled approach, we show the effect of ρ to ARI achieved by our method in **Table 5**. It shows that, when the density is low, the best ρ tends to be high in general. This results clarify the effect of PU learning, because this setting puts more bias to positive edges and most real-world graphs are sparse. The WebKB dataset behaves differently, since it is the web graph managed by universities so there is almost no missing positive edges

4.7 Discussion about NAGC-U and NAGC-UH

NAGC-U is obtained from the topology cluster assignment and NAGC-UH is obtained from the attribute cluster assignment. Here we have a question: which clustering result the users should choose obtained from NAGC-U or NAGC-UH? If we know the grand truth beforehand, we can choose either of them depending on the grand truth. However, the grand truth is usually unknown in real applications. In practice, we provide the both results to the users so that they can choose more suitable one. This is a type of trial-and-error tasks during clustering analysis, such as choosing the suitable number of clusters.

We show NMI scores between the clustering results of NAGC-U and NAGC-UH in **Table 6**. NAGC-U and NAGC-UH obtain the similar clusters in five datasets which are WebKB, Wiki, Cora, Citeseer, and BlogCatalog since NMI scores are high. The difference between the clusters obtained from NAGC-U and NAGC-UH indicates that there are nodes whose cluster assignments differ depending on whether the topology or the attribute more largely effects to the clusters. On the other hand, NMI is 0.000 in Polblog. This result implies that the topology is independent from the attributes in this dataset. We can also observe that there is a complicated relationship between the topology and the attributes in Flickr since its NMI score is relatively low.

5. Related Work

There are many clustering methods for attributed graphs [1], [14], [29], [35], [43] and representation learning techniques for attributed graphs [13], [37], [41]. Most of the representation learning for attributed graphs are influenced by graph embedding techniques [3], [11], [30], [33], [38].

5.1 Attributed Graph Clustering

SNMF is recently extended to consider both the topology and the attributes for discovering clusters of data entities. DANMF [38] is one of graph embedding techniques extended from SNMF. It learns hierarchical mappings between the original network and the final community assignment based on a deep autoencoder-like architecture. CDE [25] shares the same design with ours in that it decomposes topology matrix and attribute matrix by using NMF, but it is also orthogonal to ours in that 1) it newly introduces a community structure embedding matrix (distance matrix from vertex to vertex in cluster space) used as a topology matrix, whereas 2) our approach learns a projection function between the topology space and the attribute space and also leverages PU learning. TLSC [40] is based on generative model and it is usually not perform better than NMF-based approaches. Indeed, the experiments reported in Ref. [25] show that CDE performs higher than TLSC in terms of the NMI measure. JWNMF [14] factorizes both the topology and the attribute matrices at the same time, however, the clustering quality is not high since it does not use the transfer matrix between the topology space and attribute space: the transfer matrix effectively balances the effect between those two spaces. SA-Cluster [42] and its efficient version Inc-Cluster [43] are attributed graph clustering methods expanded from distance-based graph clustering. The key idea is to embed vertex attributes as new vertices into the graph. A unified distance for the augmented graph is defined by the random walk process, and the graph is partitioned by k-medoids. It is hard to apply these methods to large graphs since the augmented steps increase the size of the graph considerably. BAGC/GBAGC [35], [36] learns a posterior distribution over the model parameters. This method assumes that the vertices in the same cluster should have a common multinomial distribution for each vertex attribute and a Bernoulli distribution for vertex connections. The attributed graph clustering problem can be formulated as a probabilistic generative model. PAICAN [3] performs anomaly detection and clustering on the attributed graph at the same time. PAICAN explicitly models partial anomalies by generalizing the ideas of Degree Corrected Stochastic Block Models [16] and Bernoulli Mixture Models.

5.2 Representation Learning

The representation learning generates vertex features for attributed graphs and the features can be used for various tasks, clustering, link prediction, and classification.

ANRL [41] combines a neighbor enhancement autoencoder and an attribute-aware skip-gram model for learning vertex features that preserve the attributes and the network structure. It controls the topology effect by tuning parameters for the contribution of the neighbor enhancement autoencoder and for the window size on random walk. AANE [13] uses a hyperparameter λ that balances globally the effects between the topology and the attribute for all clusters. However, both ANRL and AANE can not control the effect of the attribute cluster independently to the topology cluster assignment. TADW [37] employs NMTF to decompose the topology matrix into the product of two factor matrices and text feature matrix. TADW is not robust to the text feature since factors are not extracted from the text feature. Graph Convolutional Networks [20], that is a semi-supervised learning method for a graph, has obtained considerable attention from machine learning and data mining fields due to its high performance in classifying graph vertices. However, this approach needs a subset of true cluster labels on vertices, and thus its goal is different from that of the attributed graph clustering.

6. Conclusion

We considered the clustering problem of attributed graphs. We designed an effective clustering method, NAGC, a new attributed graph clustering method by bridging the attribute space and the topology space and taking the effect of partially observed positive edges. The features of our method are two holds and both of them largely contribute to the quality of the clustering results. 1) NAGC learns a projection function between the topology space and the attribute space. The projection function consists of a rescale function and a transfer matrix that balances the effect from the attribute space to the topology space for each vertex independently. 2) NAGC leverages PU learning to take the effect of partially observed positive edges into the cluster assignment.

Our future work is as follows. First, PU learning is effective but its effect is not controlled for every vertex. We extend NAGC to control the effect depending on each vertex: some vertex should take larger effect from partially observed positive edges. Second, NAGC uses the adjacency matrix for the topology matrix. We extend it to cover the affinity matrix so as to more precisely extract the topology feature. Finally, we will employ sampling techniques to archive an efficient matrix factorization.

References

[1] Akoglu, L., Tong, H., Meeder, B. and Faloutsos, C.: PICS: Parameter-free identification of cohesive subgroups in large attributed graphs, *Proc. SDM*, pp.439–450, SIAM (2012).

[2] Ben-Dor, A. and Yakhini, Z.: Clustering gene expression patterns, *Proc. RECOMB*, pp.33–42 (1999).

[3] Bojchevski, A. and Günnemann, S.: Bayesian Robust Attributed Graph Clustering: Joint Learning of Partial Anomalies and Group Structure, *Proc. AAAI* (2018).

[4] Brohee, S. and Van Helden, J.: Evaluation of clustering algorithms for protein-protein interaction networks, *BMC Bioinformatics*, Vol.7, No.1, p. 488 (2006).

[5] Ding, C., Li, T., Peng, W. and Park, H.: Orthogonal nonnegative matrix t-factorizations for clustering, *Proc. SIGKDD* (2006).

[6] Elkan, C. and Noto, K.: Learning classifiers from only positive and unlabeled data, *Proc. SIGKDD* (2008).

[7] Flake, G.W., Lawrence, S., Giles, C.L. and Coetzee, F.M.: Self-organization and identification of web communities, *Computer*, Vol.35, No.3, pp.66–70 (2002).

[8] Fortunato, S.: Community detection in graphs, *Physics Reports*, Vol.486, No.3-5, pp.75–174 (2010).

[9] Francis, N., Green, A., Guagliardo, P., Libkin, L., Lindaaker, T., Marsault, V., Plantikow, S., Rydberg, M., Selmer, P. and Taylor, A.: Cypher: An Evolving Query Language for Property Graphs, *Proc. SIGMOD* (2018).

[10] George, B., Kim, S. and Shekhar, S.: Spatio-temporal network databases and routing algorithms: A summary of results, *International Symposium on Spatial and Temporal Databases* (2007).

[11] Grover, A. and Leskovec, J.: node2vec: Scalable feature learning for networks, *Proc. SIGKDD* (2016).

[12] Hsieh, C.-J., Natarajan, N. and Dhillon, I.S.: PU Learning for Matrix Completion, *Proc. ICML* (2015).

[13] Huang, X., Li, J. and Hu, X.: Accelerated attributed network embedding, *Proc. SDM*, SIAM (2017).

[14] Huang, Z., Ye, Y., Li, X., Liu, F. and Chen, H.: Joint weighted nonnegative matrix factorization for mining attributed graphs, *Proc. PAKDD* (2017).

[15] Jain, A., Zamir, A.R., Savarese, S. and Saxena, A.: Structural-RNN: Deep learning on spatio-temporal graphs, *Proc. CVPR* (2016).

[16] Karrer, B. and Newman, M.E.: Stochastic blockmodels and community structure in networks, *Physical Review E*, Vol.83, No.1, p.016107 (2011).

[17] Karypis, G. and Kumar, V.: A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs, *SIAM J. Sci. Comput.*, Vol.20, No.1, pp.359–392 (online), DOI: 10.1137/S1064827595287997 (1998).

[18] Karypis, G. and Kumar, V.: Multilevelk-way partitioning scheme for irregular graphs, *Journal of Parallel and Distributed Computing*, Vol.48, No.1, pp.96–129 (1998).

[19] Keet, C.M.: *Open World Assumption*, p.1567, Springer New York (2013).

[20] Kipf, T.N. and Welling, M.: Semi-Supervised Classification with Graph Convolutional Networks, *Proc. ICLR* (2017).

[21] Kuang, D., Ding, C. and Park, H.: Symmetric nonnegative matrix factorization for graph clustering, *Proc. SDM* (2012).

[22] Kuang, D., Yun, S. and Park, H.: SymNMF: Nonnegative low-rank approximation of a similarity matrix for graph clustering, *Journal of Global Optimization*, Vol.62, No.3, pp.545–574 (2015).

[23] Kulis, B., Basu, S., Dhillon, I. and Mooney, R.: Semi-supervised graph clustering: A kernel approach, *Machine Learning*, Vol.74, No.1, pp.1–22 (2009).

[24] Lee, D.D. and Seung, H.S.: Learning the parts of objects by non-negative matrix factorization, *Nature*, Vol.401, No.6755, p.788 (1999).

[25] Li, Y., Sha, C., Huang, X. and Zhang, Y.: Community Detection in Attributed Graphs: An Embedding Approach, *Proc. AAAI* (2018).

[26] Liu, B., Dai, Y., Li, X., Lee, W.S. and Yu, P.S.: Building text classifiers using positive and unlabeled examples, *Proc. ICDM* (2003).

[27] Newman, M.E.: Modularity and community structure in networks, *Proc. National Academy of Sciences*, Vol.103, No.23, pp.8577–8582 (2006).

[28] Ng, A.Y., Jordan, M.I. and Weiss, Y.: On spectral clustering: Analysis and an algorithm, *Proc. NIPS* (2002).

[29] Parimala, M. and Lopez, D.: Graph clustering based on Structural Attribute Neighborhood Similarity (SANS), *Proc. ICECCT*, pp.1–4 (online), DOI: 10.1109/ICECCT.2015.7226087 (2015).

[30] Perozzi, B., Al-Rfou, R. and Skiena, S.: DeepWalk: Online Learning of Social Representations, *Proc. SIGKDD* (2014).

[31] Sahu, S., Mhedhbi, A., Salihoglu, S., Lin, J. and Özsu, M.T.: The Ubiquity of Large Graphs and Surprising Challenges of Graph Processing, *PVLDB*, Vol.11, No.4, pp.420–431 (2017).

[32] Sevenich, M., Hong, S., van Rest, O., Wu, Z., Banerjee, J. and Chafi, H.: Using Domain-specific Languages for Analytic Graph Databases, *PVLDB*, Vol.9, No.13, pp.1257–1268 (online), DOI: 10.14778/3007263.3007265 (2016).

[33] Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J. and Mei, Q.: LINE: Large-scale Information Network Embedding, *Proc. WWW* (2015).

[34] Xu, X., Yuruk, N., Feng, Z. and Schweiger, T.A.: SCAN: A structural clustering algorithm for networks, *Proc. SIGKDD* (2007).

[35] Xu, Z., Ke, Y., Wang, Y., Cheng, H. and Cheng, J.: A model-based approach to attributed graph clustering, *Proc. SIGMOD* (2012).

[36] Xu, Z., Ke, Y., Wang, Y., Cheng, H. and Cheng, J.: GBAGC: A general bayesian framework for attributed graph clustering, *ACM Trans. Knowledge Discovery from Data* (*TKDD*), Vol.9, No.1, pp.1–43 (2014).

[37] Yang, C., Liu, Z., Zhao, D., Sun, M. and Chang, E.Y.: Network Representation Learning with Rich Text Information, *Proc. IJCAI* (2015).

[38] Ye, F., Chen, C. and Zheng, Z.: Deep autoencoder-like nonnegative matrix factorization for community detection, *Proc. 27th ACM International Conference on Information and Knowledge Management*, pp.1393–1402 (2018).

[39] Yeung, K.Y. and Ruzzo, W.L.: Details of the adjusted rand index and clustering algorithms, supplement to the paper an empirical study on principal component analysis for clustering gene expression data, *Bioinformatics*, Vol.17, No.9, pp.763–774 (2001).

[40] Zhang, G., Jin, D., Gao, J., Jiao, P., Fogelman-Soulié, F. and Huang, X.: Finding Communities with Hierarchical Semantics by Distinguishing General and Specialized topics, *Proc. IJCAI* (2018).

[41] Zhang, Z., Yang, H., Bu, J., Zhou, S., Yu, P., Zhang, J., Ester, M. and Wang, C.: ANRL: Attributed Network Representation Learning via Deep Neural Networks, *Proc. IJCAI* (2018).

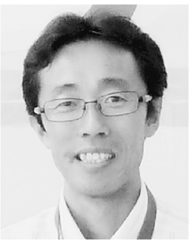
[42] Zhou, Y., Cheng, H. and Yu, J.X.: Graph clustering based on structural/attribute similarities, *PVLDB*, Vol.2, No.1, pp.718–729 (2009).

[43] Zhou, Y., Cheng, H. and Yu, J.X.: Clustering large attributed graphs:

An efficient incremental approach, *Proc. ICDM*, pp.689–698, IEEE (2010).

Seiji Maekawa is a doctoral student at Graduate School of Information Science and Technology, Osaka University. He received his B.E. degree from Kyoto University in 2016, and his M.E. degree from Osaka University in 2019. His research interests include graph data processing and database systems.

Koh Takeuchi received his B.E. and M.E. degree from Waseda University in 2009 and 2011, and his Ph.D. degree in informatics from Kyoto University in 2019. In 2011, he joined NTT Communication Science Laboratories, Japan. He is currently an assistant professor at Department of Intelligence Science and Technology, Kyoto University. His research interests include data mining, machine learning, and spatio-temporal data analysis.



Makoto Onizuka is a Professor at Graduate School of Information Science and Technology, Osaka University. He developed LiteObject (object-relational main memory database system), XMLToolkit (XML stream engine and unix-like XML data processing tools), CBoC type2 (Common IT Bases over Cloud Computing at NTT). His current research focuses on cloud-scale data management and Big data mining.