On the Equivalence Between Community Discovery and Clustering

Riccardo Guidotti[1] and Michele Coscia[2,3]

[1] KDDLab, ISTI-CNR, Pisa, Italy
riccardo.guidotti@isti.cnr.it
[2] CID, Harvard University, Cambridge, US
michele_coscia@hks.harvard.edu
[3] Naxys, University of Namur, Namur, Belgium

Abstract. Clustering is the subset of data mining techniques used to agnostically classify entities by looking at their attributes. Clustering algorithms specialized to deal with complex networks are called *community discovery*. Notwithstanding their common objectives, there are crucial assumptions in community discovery – edge sparsity and only one node type, among others – which makes its mapping to clustering non trivial. In this paper, we propose a community discovery to clustering mapping, by focusing on transactional data clustering. We represent a network as a transactional dataset, and we find communities by grouping nodes with common items (neighbors) in their baskets (neighbor lists). By comparing our results with ground truth communities and state of the art community discovery methods, we show that transactional clustering algorithms are a feasible alternative to community discovery, and that a complete mapping of the two problems is possible.

Key words: Community Discovery, Clustering, Problems Equivalence

1 Introduction

Data mining is a collection of techniques developed to extract latent knowledge from data, usually with few *a priori* assumptions. One of the most common tasks in data mining is the grouping of entities according to their similarity. We call this task *clustering*. Clustering allows us to understand when to treat different objects as part of the same class, whether we are planning a marketing campaign targeting homogeneous customers, or suggesting similar movies or books to similar audiences in a collaborative filtering entertainment platform.

Given our ever more interconnected society, complex networks emerged as one of the most successful models to mine social data. In this scenario, clustering proved to be one of the most studied problems, under the name of *community discovery*. Just like in clustering, the aim of community discovery is to group entities – in this case nodes. The main difference between clustering and community discovery is that they are defined on data represented in a different way. However, a number of researchers pointed out that they share many commonalities [3].

Still, one cannot trivially apply community discovery to solve clustering and vice versa. Notwithstanding their similarities, there are important differences in the approaches. Community discovery assumes sparse connections [3], while clustering can work with dense datasetsc [17]; in clustering we usually deal with attributes with multiple types, while community discovery usually deals with a single attribute type – edges – often binary, in the case of unweighted networks.

In this paper, we propose a community discovery $\rightarrow$ clustering mapping. To do so, we narrow down to the specific clustering problem defined for transactional data: the task of grouping baskets sharing common subsets of items. This allows us to represent a network as a transactional dataset, in which each node is a basket containing its ego-network. Then, by applying transactional clustering algorithms, we group nodes sharing common neighbors.

We perform this operation on networks with ground truth communities, i.e., indicating the real communities of the nodes. We compare the result of transactional clustering algorithms with state of the art community discovery. We find that transactional clustering provides similar results when compared with it.

Our results have a number of consequences. They suggest that the community discovery $\rightarrow$ clustering mapping is feasible: notwithstanding their differences, it is possible to map one problem into the other. This enlarges the toolbox of researches looking for network communities: by applying our mapping, they could tap into a whole new set of algorithms that previously were not considered for their problem. Our results open the way to a complete community discovery $\leftrightarrow$ clustering, which still eludes us, but for which we provide a roadmap.

2 Related Work

In the literature we find a limited number of works on the relationship between community discovery and clustering of transactional data.

In [9] the authors offer an in-depth comparative review of methods for clustering directed networks along two orthogonal classifications: one focuses on methodological principles, while the other classifies methods from the viewpoint of their assumptions about what a good cluster in a directed network is.

The authors of [10] try to discover communities in social networks through a spectral clustering method that makes full use of the network features. The core members are used for mining communities and it also exploits Page Rank for the spectral clustering initialization, to avoid the sensitivity to the initial centroids.

In [20] the authors face the community discovery problem by using *Non-negative Matrix Factorization* (*NMF*). NMF is a powerful interpretable tool with a close relationship with clustering methods. They target different types of networks (undirected, directed and compound) using three different NMF techniques: Symmetric NMF, Asymmetric NMF and Joint NMF.

In [14] the authors present a multi-level algorithm for graph clustering using simulating stochastic flows. The graph is coarsened to a manageable size, and then the algorithm performs on it a small number of iterations of flow simulations. The graph is successively refined with flows from the previous graph used

as initializations for brief flow simulations on each of the intermediate graphs. Finally, the high-flow regions of the final refined graph are clustered together, with regions without any flow forming the natural boundaries of the clusters.

In [18] we have a clustering method for learning a binomial mixture model in RNA graphs, to process text represented as graphs in the field of bioinformatics. Textual data is analyzed also in [8], where the authors study the relations of text with topics and communities. They propose a hierarchical community model distinguishing community cores from affiliated members. The method identifies potential cores as seeds of communities through relation analysis and eliminates the influence of initial parameters through an attribute-based core merge process.

The authors of [7] present a clustering algorithm for reducing the visual complexity of a large network by temporarily replacing a set of nodes in clusters with abstract nodes. The approach defines a node similarity measure used to build a similarity matrix. The linkage pattern of the graph is thus encoded into the similarity matrix, and then one obtains the hierarchical abstraction of densely linked subgraphs by applying the k-means algorithm [17] to the matrix.

Even though in a certain sense all the works described above consider simultaneously both community discovery and clustering, this is done only for practical purposes. Here, we consider this mapping in a principled way, not to find a good heuristic, but to overcome the differences between the two problems and using existing techniques in one problem to solve the other.

3 Problem Definition

In this section we first define the problem of community discovery in complex networks. Then, we define the problem of clustering transactional data. Finally, we offer a mapping of the two, suggesting possible avenues for their equivalence.

Community Discovery ($\mathcal{CD}$). Let $G = (V, E)$ be a graph, where V is the set of nodes in the graph, and $E \subset V \times V$ is the set of edges, or node pairs. In $\mathcal{CD}$, we start from the assumption that membership to E is not random, but it follows an unknown function f. Function f governs the probability of two nodes i and j to be connected. There are different ways to model f, each one of them implying a different definition of *what a community is* [3].

In the following, we adopt the branch of community discovery assuming that nodes have unobserved attributes from a set A, and the more attributes two nodes have in common the more likely they will be part of the same community[1]. If $P = \{p_1, \dots p_K\}$, with $p \subseteq V$, is a node partition extracted from the space of all possible partitions of V, then the aim of community discovery is to find $argmax_P \gamma(A, P)$, where γ is a function comparing P to the actual node attributes, for instance their mutual information. Note that, if for nodes i and j $a_i = a_j$, then $f(i, j) \sim 1$. Therefore, a good partitioning P of the nodes should

[1] Note that this is far from being an unproblematic definition [11], but it will do for the scope of this paper.

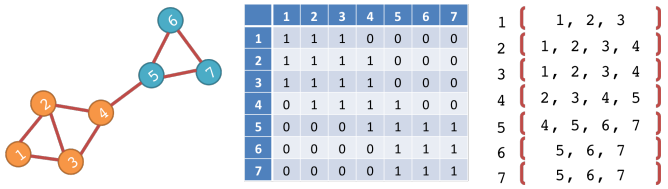

Fig. 1. Transformation of a network (left) into a set of transactions (one for each node, right), passing through the adjacency matrix (middle) of the network.

guarantee that $P \sim f$. In other words, if $p \in P$ and if $i, j \in p$, i.e., i and j are part of the same community p, then $(i, j) \in E$ with high likelihood.

Transactional Clustering Problem ($\mathcal{TC}$). Moving to $\mathcal{TC}$, let $B = \{b_1, \ldots, b_N\}$ be a set of N baskets (transactions) and $I = \{i_1, \ldots, i_D\}$ a set of D items. A basket b_i is a subset of items such that $\varnothing \subset b_i \subseteq I$. The transactional clustering problem consists in finding the partition P of B such that $argmax_P \pi(B, P)$, where π considers the average size of all baskets pairwise intersection $b_i \cap b_j$. Baskets b_i, b_j belong to the same basket partition $p \in P$ if they have a large item overlap. Thus, a good partition P puts together baskets which all share the largest possible subset of items I.

Problems Equivalence. The problem definitions of community discovery $\mathcal{CD}$ and transactional clustering $\mathcal{TC}$ share a similar structure, starting from different input types. We propose a $\mathcal{CD} \rightarrow \mathcal{TC}$ mapping between the two problems and a proper mechanism to transform the $\mathcal{CD}$ input into a $\mathcal{TC}$ input. The reasons why we do not propose $\mathcal{CD} \leftrightarrow \mathcal{TC}$ are discussed at the end of the section.

Let us consider a function Θ. Θ takes as input a graph G and a node $i \in V$, and it returns i and all $j \in V$ such that $(i, j) \in E$. This is the list of direct neighbors of i, or the set of nodes part of i's ego-network. Ego-networks have been successfully adopted to solve the $\mathcal{CD}$ problem [5]. In practice, the output of $\Theta(G, i)$ for all $i \in V$ can be seen as the input set B of baskets in the $\mathcal{TC}$ problem, containing as items the nodes ids. Fig. 1 shows the transformation of a network into a set of transactions passing through an adjacency matrix. Indeed, a third way to formulate this problem is the factorization of the adjacency matrix [20].

Notice that, given $\Theta(G, i) = b_i$, then $N = D = |V|$, i.e., the number of baskets is equivalent to the number of items in I and to the number of nodes in G. Moreover, we have that if $i \in V$, then $i \in I$ too, i.e., each node is an item that can be contained in a basket b_j. It follows that $I = V$.

If a partition P satisfies the objective of $\mathcal{TC}$, then all b_i and b_j in the same cluster $p \in P$ share a lot of items from I. Since $I = V$ and $i, j \in V$, then we can rewrite the previous sentence as: if P satisfies the objective of $\mathcal{TC}$, this means that i and j have a lot of common neighbors k, i.e., $f(i, k) \sim 1$ and $f(j, k) \sim 1$. In turn, this means that it is likely that both $a_i = a_k$ and $a_j = a_k$. From the transitive property of equivalence, it follows that $a_i = a_j$, which implies $f(i, j) \sim 1$. Thus, P satisfies $\mathcal{CD}$ by grouping i and j together.

To sum up: if P satisfies the objective of $\mathcal{TC}$, then it will also satisfy $\Theta(\mathcal{CD})$. As a result of this transformation, we can use algorithms developed to solve the transactional clustering problem to solve the community discovery problem.

We cannot apply Θ to B and map $\mathcal{TC}$ into $\mathcal{CD}$ because the concept of ego-network makes no sense in a transactional dataset. To perform such mapping, we need a function $\Omega = \Theta^{-1}$ that is the functional inverse of Θ. However, while Θ is a trivial operation on a graph [5], Ω is not the same on transactional data.

First, Θ has the property of automatically imposing $I = V$, which means that a basket b_i is a set of basket ids. This equivalence is broken in most transactional datasets: I is not the set of basket ids. Therefore, if Ω transforms $b_i = \{i_1, i_2, i_3\}$ into a network, it will result in a bipartite one, with two node classes: B and I. Bipartite community discovery is a different problem from traditional community discovery [3]: for instance, for two nodes $i, j \in B$ classified in the same p, $f(i, j) = 0$, because in a bipartite network B nodes connect exclusively to I nodes.

Even if we assume $I = V$, in a transactional dataset it is likely that any pair of items will appear at least once in a basket. If $I = V$, a naive Ω will result in a complete network, where $E = V \times V$. In such scenario, algorithms solving $\mathcal{CD}$ break. Ω would have to incorporate a criterion for discarding weak edges, which is far from being a solved problem, where naive strategies of filtering low weights fail [4]. For these reasons, we do not pursue the $\mathcal{CD} \leftrightarrow \mathcal{TC}$ mapping, and we focus on the $\Theta(\mathcal{CD}) \rightarrow \mathcal{TC}$ mapping for the rest of the paper.

4 Methods

Here, we describe some state-of-the-art methods addressing the community discovery and clustering of transactional data problems. We highlight the similarities and differences between the two classes of algorithms.

4.1 Community Discovery Methods

Louvain [1] is a bottom-up community discovery method maximizing the *modularity* to discover the node partition. It detects which node-community assignment would maximize the modularity, until there is no possible improvement.

LabProp [12] is a community detection method based on *label propagation*. It follows a bottom-up strategy by initially assigning a different label to each node. Then it iteratively re-labels each node with the label attached to the majority of its neighbors. This process is repeated until no further re-labeling is performed. In *LabProp*, there is not a measure to maximize, only a condition to respect.

Infomap [13] is a bottom-up community discovery approach which exploits a random walker as a proxy of information flow and then tries to minimize the *map equation*, which encodes the node identifiers initially assigned.

4.2 Transactional Clustering Methods

Clope [22] uses a global *profit* function that tries to increase the intra-cluster overlapping of transaction items by increasing the height-to-width ratio of the cluster items frequency histogram. *Clope* scans the transactions and it evaluates

the profit in placing a basket in a cluster, or in creating a new one. The procedure ends when no baskets are moved from a cluster to another. *Clope* requires a repulsion parameter r that is used to control the tightness of the clusters.

Practical [2] is a parameter-free method which scans the data and assigns each basket to an existing cluster or to a new one. The guiding quality function is akin to the "tf-idf" score used in information retrieval. *Practical* moves the baskets from a cluster to another while maximizing this function. It is able to automatically identify the clusters in presence of rare items.

TX-Means [6] is a parameter-free algorithm which partitions transactional data and extracts representative itemsets from each cluster. *TX-Means* is a top-down approach: it starts from a unique set of transactions and then splits them into partitions. The splitting principle follows the Bayesian Information Criteria (BIC) [15] of the original cluster versus the two subclusters. *TX-Means* uses the representative transactions to calculate the BIC. *TX-Means* peculiarities are *(i)* the usage of a local criterion instead of the maximization of a global function, and *(ii)* the usage of a centroid-based approach instead of a scanning schema.

4.3 On the Methods Equivalence

Here we look at the six methods and we classify them according to the following dimensions: design, optimization, order, and features.

Design. This dimension refers to the way an algorithm explores the result space. A *bottom-up* approach starts from all observations as independent clusters and looks for the best way to aggregate them. A *top-down* approach starts with all observations clustered in the same group and looks for the best splits. *TX-Means* is the only top-down approach in this paper.

Optimization. This dimension rules the way the splits and merges are performed. We distinguish between *global* and *local* criteria. A global criterion takes the relationship between the entire dataset and the partition into account (*Louvain*, *Infomap*, *Practical*, *Clope*), while a local criterion will only consider what is only directly connected to the each entity (*LabProp*, *TX-Means*). Global criteria can be further divided in several subclasses, of which we consider two. *Practical* and *Infomap* score differently nodes/items depending whether they are common or rare. *Clope* and *Louvain* score weights considering the ratio between the items/nodes in a cluster with respect to those outside the community/cluster.

Order. An *order-independent* method is deterministic: each run of the algorithm will always return the same clusters. An *order-dependent* method will adopt a randomized greedy strategy returning similar, but different, clusters across different runs. Here, *TX-Means* is the only order-independent approach.

Features. Here we look at differences in input/output of the various methods. Considering the input, all the methods are *parameter-free* with the exception of *Clope*, which requires the repulsion parameter r. We set $r = 1.5$ by default, as it empirically shows good performances on all the different datasets analyzed. As for the output, *Infomap* and *TX-Means* return a hierarchical structure. *TX-Means* also returns a representative for each community/cluster.

network	*nodes*	*edges*	*density*	*avg clus coef*	*communities*	*avg com size*
karate	34	78	0.139	0.571	2	17.0
dolphins	62	159	0.084	0.259	2	31.0
football	115	613	0.094	0.403	12	9.58
amazon	16,716	48,739	3.5e-4	0.649	5,000	13.49
dblp	93,432	335,520	7.7e-5	0.715	5,000	22.45
youtube	39,841	224,235	2.8e-4	0.197	5,000	14.59

Table 1. Networks and communities descriptions.

5 Experiments

In this section we solve the community discovery problem $\mathcal{CD}$ on real datasets with ground truth. We want to compare the performance differences between community discovery methods and transactional clustering algorithms.

5.1 Datasets

Our data comes from SNAP[2]. We chose three *small* and three *large* networks. Tab. 1 reports a description of the networks and their ground truth communities.

The karate network is the social network between members of a karate club at a US university. The dolphins network is a network of frequent associations between dolphins living off New Zealand. The football network is the network of American football games between Division IA colleges during Fall 2000. In the amazon network we have a link between frequently co-purchased products. The Amazon product category defines the ground-truth community. The dblp network is a co-authorship network where two authors are connected if they published at least one paper together. Authors who published to a certain journal/conference form a community. Finally, in the youtube social network users form friendship with each other and can create groups. The user-defined groups are ground-truth communities. For these three datasets we consider the top 5,000 communities with highest quality [21]. Nodes which are not part of any of these 5,000 communities are dropped from the network.

5.2 Evaluation Measures

To evaluate the clustering quality we quantify the similarity between clusters and ground truth with the *Normalized Mutual Information (NMI)* [19]. *NMI* is preferred over *purity* because *(i)* it is more sensitive to the change in the clustering results, and *(ii)* it takes into account unbalanced distributions and does not necessarily improve when the number of clusters increases (as purity does). Given two sets of clusters $\mathcal{C}=\{c_1 \dots c_k\}$ and $\mathcal{G}=\{g_1 \dots g_{k'}\}$: $\text{NMI}(\mathcal{C},\mathcal{G}) = \frac{I(\mathcal{C},\mathcal{G})}{0.5*H(\mathcal{C})+0.5*H(\mathcal{G})} \in [0,1]$, where $I(\mathcal{C},\mathcal{G})=\sum_k \sum_j \left(\frac{|c_k \cap g_j|}{N}\right) \log\left(\frac{N|c_k \cap g_j|}{|c_k||g_j|}\right)$ is the mutual information [19], and $H(\mathcal{C}) = -\sum_k \left(\frac{|c_k|}{N}\right) \log\left(\frac{|c_k|}{N}\right)$ is the Shannon entropy [16]. Aligned partitions have a *NMI* $\sim$1, misaligned partitions $\sim$0.

[2] https://snap.stanford.edu/data/#communities

network	*Infomap*		*Louvain*		*LabProp*		*TX-Means*		*Clope*		*Practical*	
	NMI	δ_k	NMI	δ_k	NMI	δ_k	NMI	δ_k	NMI	δ_k	NMI	δ_k
`karate`	0.71	**1**	0.27	3	0.23	2	0.73	2	0.43	3	**0.77**	2
`dolphins`	0.57	4	0.04	4	0.01	**2**	0.53	6	0.52	10	**0.77**	**2**
`football`	**0.94**	**0**	0.20	-1	0.22	-1	0.87	-2	0.91	3	0.88	8
`amazon`	**0.87**	-3,893	0.50	-3,877	0.54	-3,375	0.68	**370**	0.84	-1,583	0.68	-2,162
`dblp`	0.64	-4,240	0.59	-4,417	0.75	4,239	0.60	**-563**	0.86	-1,390	**0.88**	-1,233
`youtube`	0.39	-4,349	0.27	-4,037	0.32	-3,756	0.38	**-408**	**0.72**	12,024	0.33	3,654

Table 2. Experiments evaluation through NMI and δ_k. Best performer for each dataset and each measure highlighted in bold. Ties are not broken.

We also evaluate the *deviation* $\delta_k = |\mathcal{C}| - |\mathcal{G}|$ between the real number of clusters and the number of clusters detected: $\delta_k \sim 0$ when the right number of cluster is detected, $\delta_k > 0$ if more clusters than real ones are detected, $\delta_k < 0$ otherwise.

5.3 Results

We report in Tab. 2 the results of the experiments performed by solving the community discovery problem both for community discovery algorithms and also for transactional clustering algorithms, after having transformed the network according to the procedure described in Section 3.

First of all, we underline the good performance of transactional clustering algorithms. In particular, with respect to the *deviation* δ_k it seems that clustering algorithms are better than community discovery methods in approximating the right number of communities for large networks.

Secondly, we observe that *Clope* is resulting to be one of the best performer in general in terms of NMI, but in the case of the `youtube` dataset it overestimates the number of communities, producing more than the double of the number of the real communities. As already discussed in [6], *TX-Means* is the best algorithm in keeping low the *deviation* δ_k, at least in larger networks.

Finally, we notice how the clustering algorithms are able to maintain good NMI performance both for small and large dataset with very different levels of density, nodes overlapping and average community size while some community discovery methods have very poor performance in the small networks. In summary, there is not clear comparative advantages in community discovery for community discovery methods, and more often than not a transactional clustering algorithm will be the best performer.

To conclude the section, we depict in Fig. 2 the results of a community discovery algorithm (Infomap) and a transactional clustering one (*TX-Means*). We can see that the results are quantitatively the same, as the NMI values are almost equivalent. However, *Infomap* and *TX-Means* have different points of failure, which might make one more suitable in some scenarios than the other.

TX-Means returns a higher number of communities than the ground truth. However, it perfectly characterizes the boundary between the ground truth communities, and the additional ones are simply sub communities, hierarchically contained in the main two. On the other hand, *Infomap* is closer in number of

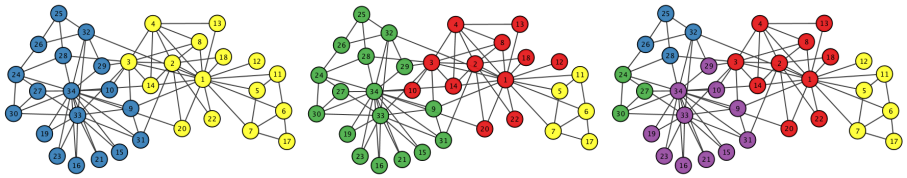

Fig. 2. The `karate` results: ground truth on the left, community discovery algorithm (*Infomap*, center), transactional clustering algorithm (*TX-Means*, right).

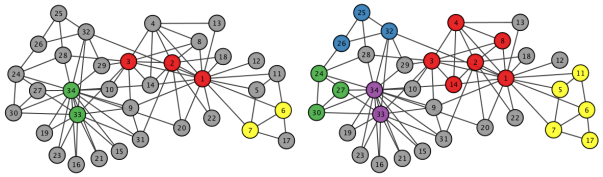

Fig. 3. The `karate` communities core nodes using a community discovery algorithm (*Infomap*, left) and a transactional clustering algorithm (*TX-Means*, right).

communities, by returning only one of the three sub communities. However, its community boundary detection is worse than *TX-Means*: node 10 should have been classified on the leftmost community instead of in the rightmost one.

Finally, in Fig. 3 we highlight another common characteristic of *TX-Means* and *Infomap*: their ability to enhance which are the *core* nodes of each community. *Infomap* considers as core nodes those with the highest page rank score, while for *TX-Means* we assume the core nodes are those in the representative transactions. Fig. 3 shows how *TX-Means* (right) is more "conservative" than *Infomap* (left): *TX-Means* mainly returns cliques of nodes delineating the backbone of the communities, *Infomap* mainly returns the hubs of the communities.

6 Conclusion

In this paper we have considered the problems of community discovery and of clustering transactional data and we have provided a formal mechanism to map the former into the latter. We have illustrated the main methodological similarities and differences among existing community discovery and transactional clustering approaches. Through some experiments, we have showed that such transformation empowers transactional clustering algorithms to perform as well as, or sometimes better than, specialized community discovery algorithms.

This paper can be the starting point for several interesting research directions. A very promising one is the extension of our mapping to show that community discovery $\leftrightarrow$ clustering. To add the left equivalence, we need a methodology to clean noise from transactional data. One promising approach to do it is to use algorithms developed for solving the network backboning problem [4].

Acknowledgements

This work is partially supported by the European Project *SoBigData: Social Mining & Big Data Ecosystem*, `http://www.sobigdata.eu`, GS501100001809, 654024. Michele Coscia has been partly supported by FNRS, grant #24927961.

References

1. V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *JSTAT*, 2008(10), 2008.
2. M. Bouguessa. A practical approach for clustering transaction data. In *MLDM*, pages 265–279. Springer, 2011.
3. M. Coscia, F. Giannotti, and D. Pedreschi. A classification for community discovery methods in complex networks. *SADM*, 4(5):512–546, 2011.
4. M. Coscia and F. M. Neffke. Network backboning with noisy data. In *ICDE*, pages 425–436. IEEE, 2017.
5. M. Coscia, G. Rossetti, F. Giannotti, and D. Pedreschi. Uncovering hierarchical and overlapping communities with a local-first approach. *TKDD*, 9(1):6, 2014.
6. R. Guidotti, A. Monreale, M. Nanni, F. Giannotti, and D. Pedreschi. Clustering individual transactional data for masses of users. *KDD*, 2017.
7. X. Huang and W. Lai. Clustering graphs for visualization via node similarities. *Journal of Visual Languages & Computing*, 17(3):225–253, 2006.
8. H. Li, Z. Nie, W.-C. Lee, L. Giles, and J.-R. Wen. Scalable community discovery on textual data with relations. In *CIKM*, pages 1203–1212. ACM, 2008.
9. F. D. Malliaros and M. Vazirgiannis. Clustering and community detection in directed networks: A survey. *Physics Reports*, 533(4):95–142, 2013.
10. S. Niu, D. Wang, S. Feng, and G. Yu. An improved spectral clustering algorithm for community discovery. In *HIS*, volume 3, pages 262–267. IEEE, 2009.
11. L. Peel, D. B. Larremore, and A. Clauset. The ground truth about metadata and community detection in networks. *Science Advances*, 3(5), 2017.
12. U. N. Raghavan, R. Albert, and S. Kumara. Near linear time algorithm to detect community structures in large-scale networks. *Physical review E*, 76(3), 2007.
13. M. Rosvall and C. T. Bergstrom. Maps of random walks on complex networks reveal community structure. *PNAS*, 105(4):1118–1123, 2008.
14. V. Satuluri and S. Parthasarathy. Scalable graph clustering using stochastic flows: applications to community discovery. In *KDD*, pages 737–746. ACM, 2009.
15. G. Schwarz et al. Estimating the dimension of a model. *The annals of statistics*, 6(2):461–464, 1978.
16. C. Shannon. A mathematical theory of communication. *SIGMOBILE, 3–55*, 2001.
17. P.-N. Tan, M. Steinbach, V. Kumar, et al. *Introduction to data mining*, volume 1. Pearson Addison Wesley Boston, 2006.
18. K. Tsuda and T. Kudo. Clustering graphs by weighted substructure mining. In *ICML*, pages 953–960. ACM, 2006.
19. N. X. Vinh et al. Information theoretic measures for clusterings comparison: is a correction for chance necessary? In *ICML*, pages 1073–1080. ACM, 2009.
20. F. Wang, T. Li, X. Wang, S. Zhu, and C. Ding. Community discovery using nonnegative matrix factorization. *DAMI*, 22(3):493–521, 2011.
21. J. Yang and J. Leskovec. Defining and evaluating network communities based on ground-truth. *Knowledge and Information Systems*, 42(1):181–213, 2015.
22. Y. Yang, X. Guan, and J. You. Clope: a fast and effective clustering algorithm for transactional data. In *TKDE*, pages 682–687. ACM, 2002.