A One-by-One Method for Community Detection in Attributed Networks

Soroosh Shalileh$^{1[0000-0001-6226-4990]}$ and Boris Mirkin$^{1[0000-0001-5470-8635]}$

Department of Data Science and Artificial Intelligence NRU HSE Moscow Russian Federation 11, Pokrovski Boulevard, Moscow, 109028, RF
sr.shalileh@gmail.com, bmirkin@hse.ru
https://cs.hse.ru/

Abstract. The problem of community detection in a network with features at its nodes takes into account both the graph structure and node features. The goal is to find relatively dense groups of interconnected entities sharing some features in common. We apply the so-called data recovery approach to the problem by combining the least-squares recovery criteria for both, the graph structure and node features. In this way, we obtain a new clustering criterion and a corresponding algorithm for finding clusters/communities one-by-one. We show that our proposed method is effective on real-world data, as well as on synthetic data involving either only quantitative features or only categorical attributes or both. Our algorithm appears competitive against state-of-the-art algorithms.

Keywords: Attributed Network · Cluster Analysis · Community Detection · Least Squares Criterion · One by One Clustering

1 Introduction: Previous work and motivation

Community detection is a popular field of data science with various applications ranging from sociology to biology to computer science. Recently this concept was extended from flat and weighted networks to networks with a feature space associated with its nodes, these are referred to as attributed (or feature-rich) networks [6]. A community is a group, or cluster, of densely interconnected nodes that are similar in the feature space too.

There have been published a number of papers proposing various approaches to identifying communities in attributed networks (see recent reviews in [6] and[3]). They naturally fall in three groups: (a) those heuristically transforming the feature-based data to augment the network format, (b) those heuristically convering the data to the features only format, and (c) those involving, usually, a probabilistic model of the phenomenon to apply the maximum likelihood principle for estimating its parameters. A typical method within approach (a) or (b) combines a number of heuristical approaches, thus involving a number of unsubstantiated parameters which are rather difficult to systematize, the more so to put to testing. Most interesting approaches in the modeling group (c) are represented by methods in [17] and [14]. The former statistically models interrelation between the network structure and node attributes, the latter involves Bayesian inferences.

Our approach relates to that of modeling, except that we model the data rather than the process of data generation. Specifically, our data-driven model assumes a hidden partition of the node set in non-overlapping communities and parameters encoding the average within-community link intensity and feature central points. To find this partition and parameters, a least-squares data approximation criterion is defined. To fit this criterion, a greedy-wise procedure of finding clusters one-by-one is applied. This approach as already proved successful in application to both feature data only and network/ data only [10, 2].

The rest of the paper is organized as follows. We describe our model and algorithm in Section 2. In Section 3, we describe the setting of our experiments. In Section 4, we describe results of our experiments to validate our method and compare it with competition. We draw conclusions in Section 5.

The authors are indebted to the anonymous reviewers whose comments helped them to improve the presentation.

2 Least-squares criterion

Let us consider a dataset represented by two matrices: a symmetric $N \times N$ network adjacency matrix $P = (p_{ij})$, where p_{ij} can be any reals, and by an $N \times V$ entity-to-feature matrix $Y = (y_{iv})$ with $i \in I$, I being an N-element entity set.

We assume that there is a partition $\mathbf{S} = \{S_1, S_2, ..., S_K\}$ of I in K non-overlapping communities, a.k.a. clusters, with a binary membership vector $s_k = (s_{ik})$, $k = 1, 2, ..., K$, so that $s_{ik} = 1$ for $i \in S_k$, and $s_{ik} = 0$, otherwise. The cluster S_k is assigned with a V-dimensional center vector $c_k = (c_{kv})$ and a positive network intensity weight λ_k.

According to the least-squares principle, "right" membership vectors s_k, community centers c_k and intensity weights λ_k are minimizers of the summary least-squares criterion:

$$F(\lambda_k, s_k, c_k) = \rho \sum_{k=1}^{K} \sum_{iv} (y_{iv} - c_{kv} s_{ik})^2 + \xi \sum_{k=1}^{K} \sum_{ij} (p_{ij} - \lambda_k s_{ik} s_{jk})^2 \qquad (1)$$

The factors ρ and ξ in Eqn. (1) are expert-driven constants to balance the two sources of data, i.e features and networks.

To use a one-by-one clustering strategy [11] here, let us denote an individual community by S; its center in feature space, by c; and the corresponding intensity weight, by λ (just removing the index, k, for convenience). The extent of fit between the community and the dataset will be the corresponding part of criterion in (1):

$$F(\lambda, c, s) = \rho \sum_{i,v} (y_{iv} - c_v s_i)^2 + \xi \sum_{i,j} (p_{ij} - \lambda s_i s_j)^2 \tag{2}$$

The problem: given matrices $P = (p_{ij})$ and $Y = (y_{iv})$, find binary s, as well as real-valued λ and $c = (c_v)$, minimizing criterion (2). It is easy to prove that the optimal real-valued c_v is equal to the within-S mean of feature v, and the optimal intensity value λ is equal to the mean within-cluster link value:

$$c_v = \frac{\sum_{i \in S} y_{iv}}{|S|}; \ \lambda = \frac{\sum_{i,j \in S} p_{ij}}{|S|^2} \tag{3}$$

Criterion (2) can be further reformulated as:

$$\begin{aligned} F(s) = \rho \sum_{i,v} y_{iv}^2 - 2\rho \sum_{i,v} y_{iv} c_v s_i + \rho \sum_v c_v^2 \sum_i s_i^2 \\ + \xi \sum_{i,j} p_{ij}^2 - 2\xi\lambda \sum_{i,j} p_{ij} s_i s_j + \xi \lambda^2 \sum_i s_i^2 \sum_j s_j^2 \end{aligned} \tag{4}$$

The items $T(Y) = \sum_{i,v} {y_{iv}}^2$ and $T(P) = \sum_{ij} p_{i,j}^2$ in (4) express quadratic scatters of data matrices Y and P, respectively. Using them, Eqn. (4) can be reformulated as

$$F(s) = \rho T(Y) + \xi T(P) - G(s) \tag{5}$$

where

$$G(s) = 2\rho \sum_{i,v} y_{iv} c_v s_i - \rho \sum_v c_v^2 \sum_i s_i^2 + 2\xi\lambda \sum_{i,j} p_{ij} s_i s_j - \xi\lambda^2 \sum_i s_i^2 \sum_j s_j^2 \tag{6}$$

By putting the optimal values c_v and λ from (3) into this expression, we obtain a simpler expression for $G(s)$

$$G = \rho |S| \sum_v c_v^2 + \xi \lambda \sum_{ij} p_{ij} s_i s_j \tag{7}$$

Maximizing G in (7) is equivalent to minimizing criterion F in (2) because of (5).

One can see that maximizing the first item in (7) requires obtaining a numerous cluster (the greater the $|S|$, the better) which is as far away from the space origin, 0, as possible (the greater the squared distance from 0, $|\sum_v c_v^2|$, the better). The second item in the criterion (7) is proportional to the sum of within-cluster links multiplied by the average within-cluster link λ. Maximizing criterion (7), thus, should produce a large anomalous cluster of a high internal density.

We employ a greedy heuristic: starting from arbitrary singleton $S = i$, the seed, add entities one by one so that the increment of G in (7) is maximized. After each adding, recompute optimal c_v and λ. Halt when the increment becomes negative. After stopping, the last check is executed: **Seed Relevance Check:** Remove the seed from the found cluster S. If the removal increases the cluster contribution; this seed is extracted from the cluster.

We refer to this algorithm as Attributed Network Addition Clustering algorithm, ANAC. Our community detection algorithm SEANAC below consecutively applies ANAC to detect more than one community.

SEANAC: Sequential Extraction of Attributed Network Addition Clusters

1. Initialization. Define $J = I$, the set of entities to which ANAC applies at every iteration, and set cluster counter $k = 1$.

2. Define matrices Y_J and P_J as parts of Y and P restricted at J. Apply ANAC at J, denote the output cluster S as S_k, its center c as c_k, the intensity λ as λ_k and contribution G as G_k.

3. Redefine J by removing all the elements of S_k from it. If thus obtained J is empty, stop. Set the current k as K and output all S_k, c_k, λ_k, G_k, $k = 1, 2, ..., K$. If not, add 1 to k, and go to 2.

The implementation of our proposed algorithm and other supplementary materials can be found https://github.com/Sorooshi/SEANAC

3 Setting of experiments for validation and comparison of SEANAC algorithm

3.1 Algorithms under comparison

We take two popular algorithms in the model-based approach, CESNA [17] and SIAN [14], which have been extensively tested in computational experiments. The author-made codes of the algorithms are publicly available in [9] and [12] respectively.

3.2 Datasets: Real-world and synthetic

Real world datasets We take on five real-world data sets listed in table 1.

Malaria data set [7] The nodes are amino acid sequences containing six highly variable regions (HVR) each. The edges are drawn between sequences with similar HVRs number 6. In this data set, there are two nominal attributes

Table 1: Real world datasets under consideration. Symbols N, E, and F stand for the number of nodes, the number of edges, and the number of node features, respectively.

Name	Nodes	Edges	Features	Ground Truth
Malaria HVR6 [7]	307	6526	6	Cys Labels
Lawyers [8]	71	339	18	Derived out of office and status features
World Trade [15]	80	1000	16	Derived out of continent and structural world system features
Parliament [1]	451	11646	108	Political parties
COSN [4]	46	552	16	Region

of nodes: Cys labels derived from of a highly variable region HVR6 (assumed ground truth); and Cys-PoLV labels derived from the sequences adjacent to regions HVR 5 and 6.

Lawyers dataset [8] The Lawyers dataset comes from a network study of corporate law partnership that was carried out in a Northeastern US corporate law firm, referred to as SG & R, 1988-1991, in New England. There is a friendship network between lawyers in the study. The features in this dataset are: Status (partner, associate), Gender (man, woman), Office location (Boston, Hartford, Providence), Years with the firm, Age, Practice (litigation, corporate), Law school (Harvard or Yale, UCon., Other).

Most features are nominal. Quantitative features, "Years with the firm" and "Age", have been converted to the nominal format, so that categories of "Years with the firm" are $x <= 10$, $10 < x < 20$, and $x >= 20$; and categories of "Age" are $x <= 40$, $40 < x < 50$, and $x >= 50$.

World-Trade dataset [15]

The World-Trade dataset contains data on trade between 80 countries in 1994. The link weights represent total imports by row-countries from column-countries, in $ 1,000, for the class of commodities designated as 'miscellaneous manufactures of metal' to represent high technology products or heavy manufacture. The weights for imports with values less than 1% of the country's total imports are zeroed. The node attributes are: Continent (Africa, Asia, Europe, North America, Oceania, South America) Structural World System Position (Core, Semi-Periphery, Periphery), Gross Domestic Product per capita in $ (GDP p/c). The GDP feature is converted into a three-category nominal feature according to the minima of its histogram, $ 4406.9 and $ 21574.5. The categories are: 'Poor' , 'Mid-Range', and 'Wealthy'.

Parliament dataset [1] The nodes correspond to members of the French Parliament. An edge is drawn if the corresponding MPs sign a bill together. The features are the constituency of MPs and their political party.

Consulting Organisational Social Network (COSN) dataset [4] Nodes in this network correspond to employees in a consulting company. The (asymmetric) edges are formed in accordance with their replies to this question: "Please indicate how often you have turned to this person for information or advice on work-related topics in the past three months". The answers, coded by 0 (I Do Not Know This Person), 1 (Never), 2 (Seldom), 3 (Sometimes), 4 (Often), and 5 (Very Often), form the edge weights. Attributes: Organisational level (Research Assistant, Junior Consultant, Senior Consultant, Managing Consultant, Partner), Gender (Male, Female), Region (Europe, USA), Location (Boston, London, Paris, Rome, Madrid, Oslo, Copenhagen).

Before applying SEANAC, all attribute categories are converted into 1/0 dummy variables which are considered quantitative. These datasets can be found at public site https://github.com/Sorooshi/PhD-Datasets.

Generating synthetic data sets First of all, we specify the number of nodes N, the number of features V, and the number of communities, K, in a dataset to be generated. As the number of parameters to control is rather high, we narrow down the variation of our data generator by maintaining two types of settings only, a small size network and a medium size network. For a small size setting, we specify the values of the three parameters as follows: $N = 200$, $V = 5$, and $K = 5$. For the medium size, $N = 1000$, $V = 10$, and $K = 15$.

Generating networks At given numbers of nodes, N, and communities K, cardinalities of communities are defined uniformly randomly, up to a constraint that no community may have less than a pre-specified number of nodes (in our experiments, this is set to 30, so that probabilistic approaches are applicable), and the total number of nodes in all the communities sums to N.

Given the community sizes, we populate them with nodes, that are specified just by indices. Then we specify two probability values, p and q. Every within-community edge is drawn with the probability p, independently of other edges. Similarly, any between- community edge is drawn independently with the probability q.

Generating quantitative features To model quantitative features, we generate a Gaussian distribution at each cluster: its covariance matrix is diagonal with diagonal values uniformly random in the range $[0.05, 0.1]$ and components of the cluster center are uniformly random from the interval $\alpha[-1, +1]$; the real α controls the cluster intermix: the smaller the α, the closer are cluster centers to each other. The possibility of presence of noise in data is modeled too, by uniformly random generation a noise feature. We replicate 50% of the original data with noise features.

Generating categorical features To model categorical features, we randomly choose the number of categories for each of them from the set $\{2, 3, ..., L\}$ where $L = 10$ for small-size networks and $L = 15$ for the medium-size networks. For every k, $k = 1, ..., K$, cluster centers are generated randomly so that no two centers may coincide at more than 50% of features. Once a center of k-th cluster, $c_k = (c_{kv})$, is specified, N_k entities $i \in S_k$ are generated as follows. Given a pre-specified threshold of intermix, ϵ between 0 and 1, for every pair (i, v), $i = 1 : N_k$; $v = 1 : V$, a uniformly random real number r between 0 and 1 is generated. If $r > \epsilon$, the entry x_{iv} is set to be equal to c_{kv}; otherwise, x_{iv} is taken randomly from the set of categories specified for feature v. The closer ϵ to 1, the more similar to the center are the entities.

Generating mixed scale features Quantitative and categorical features are generated in equal numbers independently of each other.

The synthetic data as well as real world data can be found at public site: https://github.com/Sorooshi/SEANAC/data.

3.3 Evaluation criterion

To evaluate the result of a community detection algorithm, we compare the found partition with that generated, by using the customary Adjusted Rand Index (ARI) [5]. The closer the value of ARI to unity, the better the match between the partitions. If one of the partitions consists of just one part containing all I, then ARI=0.

Table 2: Performance of SEANAC on synthetic networks combining quantitative and categorical features for two different sizes: The average ARI index and its standard deviation over 10 different data sets.

p q α/ϵ	Small-Size Networks		50% noisy feature		Medium-size Networks		50% Noisy features	
0.9, 0.3, 0.9	0.99(0.01)	5.00(0.00)	0.99(0.01)	5.00(0.00)	1.00(0.00)	15.00(0.00)	1.00(0.01)	15.00(0.00)
0.9, 0.3, 0.7	0.98(0.03)	5.00(0.00)	0.99(0.02)	5.00(0.00)	1.00(0.00)	15.00(0.00)	0.99(0.01)	15.00(0.00)
0.9, 0.6, 0.9	0.91(0.01)	4.60(0.50)	0.88(0.01)	4.50(0.67)	0.95(0.08)	14.00(1.26)	0.93(0.10)	13.70(1.67)
0.9, 0.6, 0.7	0.86(0.14)	4.80(0.60)	0.88(0.14)	4.80(0.39)	0.84(0.08)	12.10(1.22)	0.81(0.09)	11.80(1.47)
0.7, 0.3, 0.9	0.99(0.02)	5.00(0.00)	0.99(0.01)	5.00(0.00)	0.99(0.01)	14.90(0.30)	0.99(0.01)	14.90(0.30)
0.7, 0.3, 0.7	0.94(0.10)	4.90(0.30)	0.95(0.06)	4.90(0.30)	0.99(0.01)	14.80(0.40)	0.96(0.07)	14.30(1.19)
0.7, 0.6, 0.9	0.74(0.20)	3.80(0.87)	0.73(0.15)	4.20(0.87)	0.56(0.14)	7.80(1.78)	0.55(0.14)	8.10(1.70)
0.7, 0.6, 0.7	0.67(0.14)	4.30(1.10)	0.57(0.14)	3.90(0.54)	0.39(0.09)	7.10(1.51)	0.42(0.08)	7.40(0.66)

4 Results of computational experiments

The goal of our experiments is to test validity of the SEANAC algorithm over all types of attributed network datasets under consideration. In the cases at which features are categorical, the SEANAC algorithm is to be compared with the popular algorithms SIAN and CESNA, which work with categorical features only.

4.1 Parameters of the generated datasets

We set network parameters, the probability of a within-community edge, p, and that between communities, q, to take either of two values each, $p = 0.7, 0.9$ and $q = 0.3, 0.6$. In the cases at which all the features are categorical, we decrease q-values to $q = 0.2, 0.4$, because all the three algorithms fail at $q = 0.6$. Feature generation is controlled by an intermix parameter, α at quantitative features, and ϵ at categorical features. We take each of the intermix parameters to be either 0.7 or 0.9.

We may explicitly insert 50% features that are uniformly random in some datasets.

Therefore, generation of synthetic datasets is controlled by specifying six two-valued and one three-valued parameters leading to 192 combinations of these altogether. At each setting, we generate 10 datasets, run a community detection algorithm, and calculate the mean and the standard deviation of ARI index at these 10 datasets.

4.2 Validity of SEANAC

Table 2 presents the results of our experiments at synthetic datasets with mixed scale features.

One can see that SEANAC successfully recovers the numbers of communities at $q = 0.3$ and mostly fails at $q = 0.6$ – because this corresponds to a counterintuitive situation at which the probability of a link between separate communities is greater than 0.5. Yet even in this case the partition is recovered exactly when

Table 3: Comparison: average ARI values and their standard deviation over 10 different data sets for CESNA, SIAN and SEANAC at synthetic data sets with categorical features. The best results are highlighted using bold print.

setting	Small Size Networks			Medium Size Networks		
p q ϵ	CESNA	SIAN	SEANAC	CESNA	SIAN	SEANAC
0.9, 0.3, 0.9	**1.00(0.00)**	0.55(0.29)	0.99(0.01)	0.89(0.05)	0.00(0.00)	**1.00(0.00)**
0.9, 0.3, 0.7	0.95(0.10)	0.48(0.29)	**0.97(0.02)**	0.85(0.08)	0.00(0.00)	**0.99(0.01)**
0.9, 0.6, 0.9	0.93(0.08)	0.32(0.25)	**0.96(0.01)**	0.63(0.06)	0.00(0.00)	**0.99(0.01)**
0.9, 0.6, 0.7	**0.90(0.06)**	0.11(0.14)	0.75(0.12)	0.48(0.09)	0.00(0.00)	**0.96(0.03)**
0.7, 0.3, 0.9	0.97(0.08)	0.55(0.16)	**0.98(0.02)**	0.77(0.07)	0.03(0.08)	**1.00(0.01)**
0.7, 0.3, 0.7	**0.89(0.14)**	0.51(0.21)	0.87(0.07)	0.71(0.13)	0.00(0.00)	**0.99(0.01)**
0.7, 0.6, 0.9	0.50(0.10)	0.05(0.09)	**0.90(0.07)**	0.06(0.02)	0.00(0.00)	**0.99(0.01)**
0.7, 0.6, 0.7	0.20(0.08)	0.03(0.04)	**0.60(0.09)**	0.02(0.01)	0.00(0.00)	**0.91(0.04)**

other parameters keep its structure tight, as say at $p = 0.9$. Insertion of noise features does reduce the levels of ARI but not that much. The real reduction in the numbers of recovered communities, 7-8 out of 15 ones generated, occurs at the medium size datasets at really loose data structures with $p = 0.7$ and $q = 0.6$, leading to significant drops in the levels of ARI values as well.

The picture is much similar at the cases of quantitative only and categorical only feature scales - they are left out to shorten the paper.

4.3 Comparing SEANAC and competition

In this section, we compare the performance of SEANAC with that of CESNA [17], and SIAN [14]. It should be noted that SEANAC determines the number of clusters automatically, whereas both CESNA and SIAN need that as part of the input. Table 3 presents our results at synthetic datasets (with categorical features only, as required by the competition) and Table 4, at real world datasets.

One can see that at small sizes CESNA wins three times (out of 8), and at all the other cases, including at medium size datasets, SEANAC wins. SIAN never wins in this table. Moreover, SIAN comprehensively fails on all counts at medium sizes by producing NaN which we interpret as a one-cluster solution.

We also experimented with a slightly different design for ,categorical feature generation. That different design sets an entity to either coincide with its cluster center or to be entirely random. At that design CESNA wins 7 times at the small size datasets and SEANAC wins at 7 medium size datasets.

At the real world datasets, CESNA never wins; SEANAC wins three times, and SIAN, two times (see Table 4).

Here, we chose that data normalization method leading, on average, to the larger ARI values. Specifically, we use z-scoring for normalizing features in Lawyers data set, HVR data set and COSN data set. The best results on World-Trade data set and parliament data set are obtained with no normalization. The network data in Lawyers and HVR are normalized with applying the modularity

Table 4: Comparison of CESNA, SIAN and SEANAC on Real-world data sets; average values of ARI and standard deviation (std) are presented over 10 random initialization. The best results are shown using bold print.

	CESNA	SIAN	SEANAC
HRV6	0.20(0.00)	0.39(0.29)	**0.45(0.14)**
Lawyers	0.28(0.00)	0.59(0.04)	**0.63(0.06)**
World Trade	0.23(0.00)	**0.55(0.07)**	0.23(0.03)
Parliament	0.25(0.00)	**0.79(0.12)**	0.28(0.01)
COSN	0.44(0.00)	0.43(0.05)	**0.50(0.11)**

transformation [13]. The network data of COSN is normalized by shifting all the similarities to the average link value [11].

5 Conclusion

This paper proposes a novel combined data recovery criterion for the problem of detecting communities in an attributed network. Our algorithm extracts clusters one by one. This allows us to determine the number of clusters automatically, whereas other algorithms need the number of clusters pre-specified. Another feature of our approach is that it is more or less universal regarding the scales of the data available. On the other hand, SEANAC results may depend on data normalization.

We experimentally show that SEANAC is competitive over both synthetic and real-world data sets against two popular state-of-the-art algorithms, CESNA [17] and SIAN [14].

There should be several possible directions for future work over the data recovery approach accepted in this paper. First of all, its extension to large datasets should be proposed and validated. Then the possibility of trade-off between two constituent data sources, network and fetures, which is explicitly present in our criterion should be investigated. Yet another direction for future work shoud be a systematic investigation of the relative effect of different data standardization methods on the results of our method.

References

1. A. Bojchevski, and S. Günnemann, Bayesian robust attributed graph clustering: Joint learning of Partial anomalies and group structure. In Thirty-Second AAAI Conference on Artificial Intelligence, 2018.
2. M.M.T. Chiang, and B. Mirkin, Intelligent choice of the number of clusters in k-means clustering: an experimental study with different cluster spreads, Journal of Classification, 27(1), pp. 3-40, 2010.
3. P. Chunaev, Community detection in node-attributed social networks: a survey, arXiv preprint arXiv:1912.09816, 2019.

4. R.L. Cross, and A. Parker, The hidden power of social networks: Understanding how work really gets done in organizations. Harvard Business Press, 2004.
5. L. Hubert, and P. Arabie, Comparing partitions, Journal of Classification, 2(1), pp.193-218, 1985.
6. R. Interdonato, M. Atzmueller, S. Gaito, R. Kanawati, C. Largeron, and A. Sala, Feature-rich networks: going beyond complex network topologies. Applied Network Science, 4, 2019.
7. D.B. Larremore, A. Clauset, and C.O. Buckee, A network approach to analyzing highly recombinant malaria parasite genes. PLoS Computational Biology, 9(10), p.e1003268, 2013.
8. E. Lazega, The Collegial Phenomenon: The Social Mechanisms of Cooperation Among Peers in a Corporate Law Partnership, Oxford University Press, 2001.
9. J. Leskovec, and R. Sosič, SNAP: A General-Purpose Network Analysis and Graph-Mining Library, ACM Transactions on Intelligent Systems and Technology (TIST), vol. 8-1, pp 1, ACM, 2016. CESNA on Github: https://github.com/snap-stanford/snap/tree/master/examples/cesna
10. B. Mirkin, and S. Nascimento, Additive spectral method for fuzzy cluster analysis of similarity data including community structure and affinity matrices. Information Sciences, 183(1), pp.16-34, 2012.
11. B. Mirkin, Clustering: A Data Recovery Approach, CRC Press (1st Edition, 2005; 2d Edition, 2012).
12. Nature Communications, https://www.nature.com/articles/ncomms11863.
13. M.E. Newman, Modularity and community structure in networks. Proceedings of the National Academy of Sciences, 103(23), pp.8577-8582, 2006.
14. M.E. Newman, and A. Clauset, Structure and inference in annotated networks. Nature Communications, 7, p.11863, 2016.
15. W. De Nooy, A. Mrvar, and V. Batagelj, Exploratory Social Network Analysis with Pajek, Cambridge: Cambridge University Press, Chapter 2, 2004.
16. T. Snijders, The Siena webpage. https://www.stats.ox.ac.uk/ snijders/siena/Lazega_lawyers_data.htm
17. J. Yang, J. McAuley, and J. Leskovec, Community detection in networks with node attributes. In 2013 IEEE 13th International Conference on Data Mining (pp. 1151-1156). IEEE, 2013.