Detection of an Unspecified Number of Communities in Feature-Rich 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. Existing approaches require the number of communities pre-specified. We apply the so-called data recovery approach to allow a relaxation of the criterion for finding 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. In the cases at which attributes are categorical, state-of-the-art algorithms are available. Our algorithm appears competitive against them.

Keywords: Attributed Network · Feature-Rich Network · Cluster Analysis · Community Detection · Data Recovery · 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 [7]. 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 feature-rich networks (see recent reviews in [7] 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 [23] and [17]. The former statistically models inter-relation 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, an encoding process should be run so that a process of decoding reproduces the data as well as possible. Such an approach is referred to as data recovery approach in [14]; in the neural network domain, this approach is referred to as an auto-encoder [18].

We propose using a greedy-wise procedure of finding clusters one-by-one within the data recovery approach, as already proved successful in applcation to both feature data only and network/similarity data only [13, 2]. In contrast to other approaches, this one is applicable to mixed scale data, and moreover, it needs no pre-specified number of clusters. Our experiments show that this approach is valid and competitive against other state-of-the-art approaches.

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).

2 A data recovery model

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 $S = \{S_1, S_2, ..., S_K\}$ of I in K non-overlapping communities, a.k.a. clusters, related to this dataset as described below.

Denote *k-th* cluster binary membership vector by $s_k = (s_{ik})$, $k = 1, 2, ..., K$, so that its i-th component is equal to unity for $i \in S_k$, and zero otherwise. The cluster is assigned with a V-dimensional center vector $c_k = (c_{kv})$. Also, there is a positive network intensity weight of *k-th* cluster denoted by λ_k, to adjust the binary s_{ik} values to the measurement scale of the network adjacency matrix P.

According to the data-recovery approach, any given partition $S = \{S_1, S_2, ..., S_K\}$ of I, V-dimensional cluster centers $c_1, c_2, ..., c_K$ and cluster intensity weights $\lambda_1, \lambda_2, ..., \lambda_K$, can be used to recover both the feature values and network links according to equations (1) and (2) below:

$$y_{iv} = \sum_{k=1}^{K} s_{ik} c_{kv} + f_{iv}, i \in I, v \in V, \quad (1)$$

$$p_{ij} = \sum_{k=1}^{K} \lambda_k s_{ik} s_{jk} + e_{ij}, i, j \in I. \quad (2)$$

Here values e_{ij} and f_{iv} are residuals that should be as small as possible.

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:

$$F(\lambda_k, s_k, c_k) = \rho \sum_{k=1}^{K} \sum_{iv} (y_{iv} - c_{kv} s_{ik})^2 + \xi \sum_{k=1}^{K} \sum_{ij} (p_{ij} - \lambda_k s_{ik} s_{jk})^2 \quad (3)$$

The factors ρ and ξ in Eqn. (3) are expert-driven constants to balance the two sources of data, taken to be both equal to unity in this paper.

On the first glance, criterion in Eqn. (3) differs from what follows from Eqns. (2) and (1): the operation of summation over k is outside of the parentheses in it, whereas these equations require that to be within the parentheses. However, the formulation in (3) is consistent with the models in (2) and (1) because vectors s_k $(k = 1, 2, ..., K)$ correspond to a partition and thus are mutually orthogonal: For any specific $i \in I$, s_{ik} is zero for all k except one; that one k at which $i \in S_k$. Therefore, each of the sums over k in Eqns. (2) and (1) consists of just one item, so that the summation sign may be applied outside of the parentheses indeed.

To use a one-by-one clustering strategy [14] 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 (3):

$$F(\lambda, c_v, s_i) = \rho \sum_{i,v} (y_{iv} - c_v s_i)^2 + \xi \sum_{i,j} (p_{ij} - \lambda s_i s_j)^2 \quad (4)$$

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 (4).

As is well known, and, in fact, easy to prove, 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:

$$c_v = \frac{\sum_{i \in S} y_{iv}}{|S|}; \ \lambda = \frac{\sum_{i,j \in S} p_{ij}}{|S|^2} \quad (5)$$

Criterion (4) can be further reformulated as:

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

The items $T(Y) = \sum_{i,v} {y_{iv}}^2$ and $T(P) = \sum_{ij} p_{i,j}^2$ in (6) express quadratic scatters of data matrices Y and P, respectively. Using them, Eqn. 6 can be reformulated as

$$F(s) = \rho T(Y) + \xi T(P) - G(s) \tag{7}$$

where

$$G(s) = 2\rho \sum_{i,v} y_{iv} c_v s_i - \rho \sum_v c_v^2 \sum_i s_i^2 + 2\xi\lambda \sum_{i,j} p_{ij} s_i s_j - \xi\lambda^2 \sum_i s_i^2 \sum_j s_j^2 \tag{8}$$

Equation (7) shows that the combined data scatter, $\rho T(Y)+\xi T(P)$ is decomposed in two complementary parts, one of which, $F(s)$, expresses the residual, that part of the data scatter which is not taken into account by the model in Eqns. (1) and (2), whereas the other part, $G(s)$, expresses the contribution of the model to the data scatter.

By putting the optimal values c_v and λ from (5) into this expression, we obtain a simpler expression for $G(s)$

$$G = \rho|S| \sum_v c_v^2 + \xi\lambda \sum_{ij} p_{ij} s_i s_j \tag{9}$$

Maximizing G in (9) is equivalent to minimizing criterion F in 4 because of 7.

One can see that maximizing the first item in (9) 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). Usually the data are pre-processed so that the origin is shifted to the center of gravity, or grand mean, the point whose components are the averages of the corresponding features. In such a case, the goal of putting the cluster as far away from 0 as possible, means that the cluster should be anomalous. The second item in the criterion (9) is proportional to the sum of within-cluster links multiplied by the average within-cluster link λ. Maximizing criterion (9), 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 (9) 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 Feature-rich Network Addition Clustering algorithm, FNAC. Consecutive application of the algorithm FNAC to detect more than one community, forms our community detection algorithm SEFNAC below.

SEFNAC: Sequential Extraction of Feature-rich Network Addition Clusters

1. Initialization. Define $J = I$, the set of entities to which FNAC 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 FNAC 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. Check whether thus obtained J is empty or not. If yes, stop. Define the current k as K and output all the solutions S_k, c_k, λ_k, G_k, $k = 1, 2, ..., K$. If not, add 1 to k, and go to 2.

3 Setting of experiments for validation and comparison of SEFNAC algorithm

To set a computational experiment, one should specify its constituents:

1. The set of algorithms under comparison.
2. The set of datasets at which the algorithms are evaluated and/or compared.
3. The set of criteria for assessment of the experimental results.

3.1 Algorithms under comparison

We take two popular algorithms in the model-based approach, CESNA [23] and SIAN [17], which have been extensively tested in computational experiments. The author-made codes of the algorithms are publicly available in [12] and [15] respectively. We also tested the algorithm PAICAN from [1] in our experiments. The results of this algorithm, unfortunately, were always less than satisfactory; therefore, we exclude the algorithm PAICAN from this paper.

3.2 Datasets

We use both real world datasets and synthetic datasets.

Real world datasets We take on five real-world data sets listed in table 1. Some of them involve both quantitative and categorical features. The algorithms under comparison, unlike the proposed algorithm SEFNAC, require that features are to be categorical. Therefore, whenever a data set contains a quantitative feature we convert that feature to a categorical version.

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 [10]	307	6526	6	Cys Labels
Lawyers [21]	71	339	18	Derived out of office and status features
World Trade [19]	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

Malaria data set [10]

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 of nodes:

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

Lawyers dataset [11]

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. It is available for downloading at [21]. There is a friendship network between lawyers in the study. The features in this dataset are:

1. Status (partner, associate),
2. Gender (man, woman),
3. Office location (Boston, Hartford, Providence),
4. Years with the firm,
5. Age,
6. Practice (litigation, corporate),
7. Law school (Harvard or Yale, UCon., Other)

Most features are nominal. Two features, "Years with the firm" and "Age", are quantitative. Authors of the previous studies converted them to the nominal format, accepted here too. The categories of "Years with the firm" are $x <= 10$, $10 < x < 20$, and $x >= 20$; the categories of "Age" are $x <= 40$, $40 < x < 50$, and $x >= 50$.

World-Trade dataset [19]

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:

1. Continent (Africa, Asia, Europe, North America, Oceania, South America)
2. Structural World System Position (Core, Semi-Periphery, Periphery),
3. Gross Domestic Product per capita in $ (GDP p/c)

We convert the GDP feature into a three-category nominal feature according to the minima of its histogram. The categories are: 'Poor' if GDP p/c is less than $ 4406.9; 'Mid-Range' if GDP is between $ 4406.9 and $ 21574.5; and 'Wealthy' if GDP is greater than $ 21574.5.

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 are coded by 0 (I Do Not Know This Person), 1 (Never), 2 (Seldom), 3 (Sometimes), 4 (Often), and 5 (Very Often). These 6 numerals are the weights of the corresponding edges. Nodes in this network have the following attributes:

1. Organisational level (Research Assistant, Junior Consultant, Senior Consultant, Managing Consultant, Partner),
2. Gender (Male, Female),
3. Region (Europe, USA),
4. Location (Boston, London, Paris, Rome, Madrid, Oslo, Copenhagen).

Before applying SEFNAC, all attribute categories are converted into 1/0 dummy variables which are considered quantitative.

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 apply the design proposed in [8]. Each cluster is generated from a Gaussian distribution whose covariance matrix is diagonal with diagonal values uniformly random in the range $[0.05, 0.1]$ to specify the cluster's spread. Each component of the cluster center is generated uniformly random from the range $\alpha[-1, +1]$, so that the real α controls the cluster intermix: the smaller the α, the closer are cluster centers to each other.

In addition to cluster intermix, we take into account the possibility of presence of noise in data. We uniformly random generate a noise feature from an interval defined by the maximum and minimum values. In this way, 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. Then, given the number of communities, K, and the numbers of entities, N_k for $(k = 1, ..., K)$; the 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 of this cluster 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 .

Consequently, all entities in cluster k-th coincide with its center, up to rare errors if ϵ is close to 1. The smaller the epsilon, the more diverse, and thus intermixed, would be the generated entities.

Generating mixed scale features

We divide the number of features in two approximately equal parts, one to consist of quantitative features, the other, of categorical features. Each part is filled in independently, as described above.

3.3 Evaluation criteria

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. Cases at which ARI is negative may occur too; these happens rarely indeed, in weird cases such as 'dual' partitions [8].

4 Results of computational experiments

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

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

Table 2: Performance of SEFNAC 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)

$q = 0.3, 0.6$. 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.

To set a more realistic design, 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:

- feature scales: quantitative, categorical, mixed;
- data size: small, medium;
- presence of noise features: yes, no;
- the probability of a within-community edge p;
- the probability of a between-community edge q;
- cluster inter-mix α or ϵ.

Therefore, there are 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.

The following two sections present our experimental results for (a) testing validity of the SEFNAC algorithm at synthetic data, and (b) comparing performance of SEFNAC and competition.

4.2 Validity of SEFNAC

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

We can see that SEFNAC 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 other parameters keep its structure tight, as say at $p = 0.9$. This holds for both

Table 3: Comparison of CESNA, SIAN and SEFNAC at synthetic data sets with categorical features. The best results are highlighted using bold-face. The average ARI value and its standard deviation over 10 different data sets is reported.

setting	Small Size Networks			Medium Size Networks		
p q ϵ	CESNA	SIAN	SEFNAC	CESNA	SIAN	SEFNAC
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)**

small size and medium size cases. 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.

The picture is much similar at the cases of quantitative only and categorical only feature scales - we do not present them to shorten the paper.

4.3 Comparing SEFNAC and competition

In this section, we compare the performance of SEFNAC with that of CESNA [23], and SIAN [17]. It should be reminded that SEFNAC determines the number of clusters automatically, whereas both CESNA and SIAN need that as part of the input. datasets 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, SEFNAC wins. SIAN never wins in this table. There is an impressive change in the performance of SIAN at the medium-sized datasets: SIAN comprehensively fails on all counts at medium sizes by producing NaN which we interpret as a one-cluster solution.

We get somewhat different results at the real world datasets. Here CESNA shows rather poor results; SEFNAC wins three times, and SIAN, two times (see Table 4).

Since criterion (3) depends on data normalization, SEFNAC results depend on that too. Out of a few popular data normalization methods, we choose that leading, on average, to the larger ARI values. Specifically, we used z-scoring for normalizing dummy variables 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

Table 4: Comparison of CESNA, SIAN and SEFNAC 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-face.

	CESNA	SIAN	SEFNAC
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)**

are normalized by applying the modularity transformation [16]. The network data of COSN is normalized by subtracting the average link value from all the similarities [14].

5 Conclusion

This paper proposes a novel combined data recovery criterion for the problem of detecting communities in a feature-rich network. Our algorithm SEFNAC (Sequential Extraction of Feature-Rich Network Addition Clusters) 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, SEFNAC results may depend on data normalization.

We experimentally show that SEFNAC is competitive over both synthetic and real-world data sets against two popular state-of-the-art algorithms, CESNA [23] and SIAN [17].

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. One more direction would be widening the scope of our synthetic data generators by, say, developing a general framework to include some popular data generators, such as those in [9].

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. D. He, D. Jin, Z. Chen, and W. Zhang, Identification of hybrid node and link communities in complex networks, Nature Scientific Reports, p.8638, 2015.
7. 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) doi:10.1007/s41109-019-0111-x.
8. E.V.Kovaleva, and B. Mirkin. Bisecting K-means and 1D projection divisive clustering: A unified framework and experimental comparison. Journal of Classification, 32(3), pp. 414-442, 2015.
9. C. Largeron, P.N. Mougel, R. Rabbany, and O.R. Zaiane. Generating attributed networks with communities. PloS one, 10(4), e0122777, 2015.
10. 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.
11. E. Lazega, The Collegial Phenomenon: The Social Mechanisms of Cooperation Among Peers in a Corporate Law Partnership, Oxford University Press, 2001.
12. 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
13. 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.
14. B. Mirkin, Clustering: A Data Recovery Approach, CRC Press (1st Edition, 2005; 2d Edition, 2012).
15. Nature Communications, https://www.nature.com/articles/ncomms11863
16. M.E. Newman, Modularity and community structure in networks. Proceedings of the National Academy of Sciences, 103(23), pp.8577-8582, 2006.
17. M.E. Newman, and A. Clauset, Structure and inference in annotated networks. Nature Communications, 7, p.11863, 2016.
18. A. Ng, Sparse autoencoder, CS294A Lecture notes 72.2011, pp. 1-19.
19. W. De Nooy, A. Mrvar, and V. Batagelj, Exploratory Social Network Analysis with Pajek, Cambridge: Cambridge University Press, Chapter 2, 2004.
20. Stanley, N., Bonacci, T., Kwitt, R., Niethammer, M. and Mucha, P.J., 2019. Stochastic block models with multiple continuous attributes. Applied Network Science, 4(1), pp.1-22.
21. T. Snijders, The Siena webpage. https://www.stats.ox.ac.uk/ snijders/siena/Lazega_lawyers_data.htm
22. Z. Xu, Y. Ke, Y. Wang, H. Cheng, and J. Cheng, A model-based approach to attributed graph clustering. In Proceedings of the 2012 ACM SIGMOD international conference on management of data (pp. 505-516). ACM, 2012.
23. 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 (https://arxiv.org/pdf/1401.7267.pdf; accessed 22 November 2019).