NATIONAL RESEARCH UNIVERSITY
HIGHER SCHOOL OF ECONOMICS

as a manuscript

Shalileh Soroosh Ahmad

Clustering Feature-Rich Networks Using Data Recovery Approach

Ph.D. Dissertation Summary
for the purpose of obtaining academic degree
Doctor of Philosophy in Computer Science

Moscow-2021

The PhD Dissertation was prepared at National Research University Higher School of Economics.

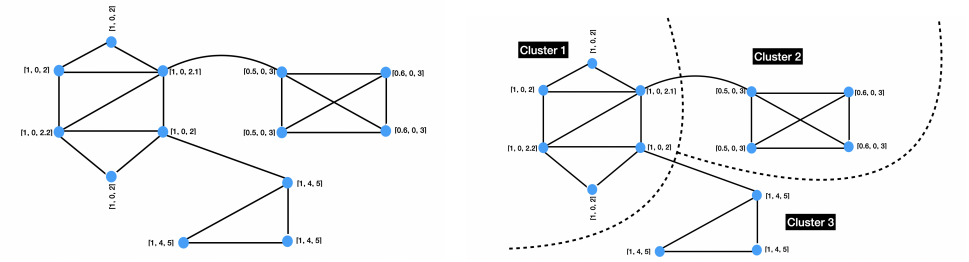
Academic Supervisor: Boris G. Mirkin, D.Sc., Associate Professor, National Research University Higher School of Economics.

Introduction

Community detection is widespread and applied in various applications ranging from sociology to biology to computer science. The corresponding data structure is a network, or graph, of objects, called nodes interconnected by pair-wise relationships (edges). Our subject is a more complex data structure, feature-rich network. Specifically, we consider networks at which a set of features are associated with the nodes. If the features are categorical, such a structure is usually referred to as a node attributed network [7, 41]. Since we consider datasets at which the features are not necessarily categorical but may be quantitative or the combination of both, we refer to these data structures as "feature-rich" networks following [20]. Figure (1a) intuitively depicts the concept of feature-rich networks.

We define a community as relatively dense interconnected nodes that are also similar in the feature space. Our goal is to extract the clusters in feature-rich networks. Formally, we can define our goal as follows. We define a feature-rich network, i.e., a network with features at the nodes, $A = \{P, Y\}$, over an entity set I. Here I is a set of network nodes of cardinality $|I| = N$; $P = (p_{ij})$ is an $N \times N$ matrix of mutual link weights between nodes $i, j \in I$; and $Y = (y_{iv})$ is an $N \times V$ matrix of feature values, so that entry y_{iv} is the value of feature $v = 1, 2, ..., V$ at node $i \in I$. Our goal is to partition I into K crisps and non-overlapping communities $S = \{S_k\}_{k=1}^{K}$, where K is the number of communities. Figure (1b) visualizes our goal.

In the rest of this summary, interchangeably, we use "cluster" and "community," which reflect the same meaning. Moreover, when either "extraction" or "detection" is associated with "cluster" or "community," this combination, e,g cluster extraction, reflects precisely the same meaning.



(A) Feature-rich networks (B) Detected clusters (our goal)

FIGURE 1: The concept of feature-rich networks and our goal. (A) visualizes the data structure, while (B) depicts our goal, which is to detect clusters/communities

The relevance and importance of research

The subject of studying networks and communities in them is receiving ever-growing attention, among other topical issues emerging with globalization and humankind's digitalization. One of the essential topics in this is a relatively recent community detection domain in feature-rich or node-attributed networks; there are already conferences and journals devoted to this domain [1, 4, 16, 17, 19].

Majority of approaches addressing community detection in feature-rich networks are heuristic, including such prominent directions as graph embedding [2, 9, 37] or Modularity-based clustering [12, 13, 15]. As useful as heuristic approaches might be, they all lack a very important characteristic: the degree of correspondence between the data and found solutions. However, there are two directions in which the correspondence between data and the detected community structure is explicit. They are theory-driven modeling and data-driven modeling.

Theory-driven approaches involve probabilistic data generation models; these models' parameters are fitted using the data [7, 30, 42]. Data-driven approaches attempt to recover the data using the found community structures. The degree of imprecision in the recovered data may be used as an explicit criterion of correspondence between the solution and data.

Although some works have been done in data-driven modeling [3, 8, 22, 25, 40], no work in the data recovery approach as is, has been conducted so far. According to this approach, the data are recovered from the structure to be found as precisely as possible, up to additive residuals. This approach has proved to be efficient in Data Science leading to such effective methods as K-means clustering and Principal Component Analysis [28].

This research explores whether the data recovery approach can be effectively applied to community detection in feature-rich networks. Specifically, the author is going to:

1. Formulate a (set of) data recovery model(s) for community detection in feature-rich networks;
2. Transform the corresponding fitting criteria and formulate for each of the resulting reformulations local search methods for sub-optimal community detection;
3. Experimentally test and modify proposed methods so that the resulting methods are effective both on real and synthetic data.

Novelty of the obtained results

The contents of this thesis demonstrate that the author accomplished the goal. Below is the list of novel results obtained by the author:

1. A set of least-squares criteria for clustering in feature-rich networks are proposed. Unlike many others, these criteria cover both flat and weighted networks, both categorical and quantitative features, and involve an explicit weighting of the two data sources, network, and features.

2. Based on a Pythagorean decomposition of the square data scatter, a local optimization method SEFNAC for one-by-one community detection in feature-rich networks is proposed; it is experimentally proved that SEFNAC is competitive against popular community detection methods and automatically determines the number of clusters rather precisely.

3. The author proposed distinguishing between two modes of using the network/similarity weights: summability and non-summability modes. The former relates to the case at which network weights can be meaningfully summed or compared across the entire data matrix; the latter, at which the summation and comparison operations are restricted within individual columns. A version of SEFNAC for the non-summability mode is competitive but works faster than the original version.

4. Converting feature space data to similarity matrix format is proposed by applying multiplication of the feature data matrix with its transpose; introduced a set of ICESi local optimization methods for thus converted data, which also appear to be competitive.

5. For a non-summable version of the least-squares criterion, an alternating minimization method KEFRiN is proposed to extend the celebrated K-means clustering approach to the issue of community detection in feature-rich networks. Also, the cosine distance is added to the innate squared Euclidean distance. The method appears to be fast, efficient, and, with the cosine distance version, effectively tackling the so-called "curse of dimensionality," inherent at the criterion for feature-rich networks.

6. The author developed a framework for experimentally testing algorithms for community detection in feature-rich networks and corresponding software and conducted multiple experimental computations to validate proposed algorithms and compare them with state-of-the-art.

Publications and approbation of the research

Our main results have been published or submitted to several journals as follows. [1]

First-tier:

1. Shalileh S. & Mirkin B. (2020) A One-by-One Method for Community Detection in Attributed Networks. In: Analide C., Novais P., Camacho D., Yin H. (eds) Intelligent Data Engineering and Automated Learning – IDEAL 2020. IDEAL 2020. Lecture Notes in Computer Science, vol 12490. Springer, Cham. (Scopus, Q2)

Second-tier:

2. Shalileh S. & Mirkin B. (2021) A Method for Community Detection in Networks with Mixed Scale Features at Its Nodes. In: Benito R.M., Cherifi C., Cherifi H., Moro E., Rocha L.M., Sales-Pardo M. (eds) Complex Networks & Their Applications IX. COMPLEX NETWORKS 2020 2020. Studies in Computational Intelligence, vol 943. Springer, Cham. (Scopus, Q4)

[1] Due to some journals' long reviewing process, we report the list of our under-review works separately.

3. Shalileh, S., & Mirkin, B. (2020, December). A data recovery method for community detection in feature-rich networks. In The 2020 ieee/acm international conference on advances in social networks analysis and mining. acm/ieee, Hague, Netherlands. (Scopus, Q4)

4. Shalileh, S., & Mirkin, B. (2020, October.). Detection of an unspecified number of communities in feature-rich networks. In The eleventh conference on network modeling and analysis, Montpelier, France. (Scopus, Q4)

Conferences: talks and poster presentation:

5. 11th Conference on Network Modeling & Analysis, Montpelier, France October 14-16, 2020. Talk title: Detection of an Unspecified Number of Communities in Feature-Rich Networks.

6. 15th INFORMS Telecommunications and Network Analytics Conference 2020, 20-21 October 2020, Berlin, Germany, Talk-title: A Novel Approach to Community Detection in Feature-Rich Networks.

7. 21st International Conference on Intelligent Data Engineering and Automated Learning - IDEAL 2020, GuimarAes, Portugal, 4th-6th November 2020. Talk-title: A One-by-One Method for Community Detection in Attributed Networks

8. The 9th International Conference on Complex Networks and their Applications, Madrid, Spain, December 1-3, 2020. Poster-title: A Method for Community Detection in Networks with Mixed Scale Features at its Nodes.

9. The 2020 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining - ASONAM 2020, Hague, Netherlands, 7-10 December 2020. Talk-title: A Data Recovery Method for Community Detection in Feature-Rich Networks.

First-tier, under review:

10. Mirkin, B., & Shalileh, S. A data recovery method for community detection in feature-rich networks. Submitted to Journal of classification (Q1 Scopus).Accepted subject to revision.

11. Shalileh, S., & Mirkin, B. Summable and nonsummable data-driven models for community detection in feature-rich networks. Submitted to Social Network Analysis and Mining (SNAM) (Scopus, Q1). Accepted subject to revision.

12. Shalileh, S., & Mirkin, B. Community extraction in feature-rich networks with least-squares criteria using similarity data. (Scopus, Q1). Accepted subject to revision.

13. Shalileh, S., & Mirkin, B. K-Means Clustering Extended to Community Detection in Feature-Rich Networks. Submitted to Information Science (Scopus, Q1).

The organization of thesis

The thesis organized as follows. We classify and review the previous works in Chapter (2). We describe our eight proposed methods in Chapter (3). We briefly review our competitors' technical descriptions, and we describe the real-world and synthetic data sets and evaluation criteria in Chapter (4). We evaluate and compare the performance of our proposed methods in Chapter (5). Finally, we conclude and explain our future works in Chapter (6).

Literature review

Recently, a comprehensive, yet concise, review of methods for community/cluster detection in feature-rich, or node attributed, networks, has been published in [11]. In this review, all the community detection methods are classified according to the stage of finding communities where the two data types, network, and features, are merged. The merger may occur before the process begins (early fusion), within the process (simultaneous fusion), and after the process (late fusion).

Early fusion and late fusion approaches should be essentially heuristic because they do not attempt to model the observed data. Therefore, the subject of our interest, methods based on data modeling, lies within the simultaneous fusion stage. Therefore, We will briefly review the early fusion and late fusion approaches, and we will focus on simultaneous fusion approaches.

Moreover, among the data modeling approaches, one may distinguish between theory-driven and data-driven approaches. Theory-driven approaches involve a world model leading to a probabilistic distribution, parameters of which can be recovered from the data. In contrast, in data-driven approaches involve no world models but rather focus on the data as is. In the latter, the data is considered as an array of numbers to be recovered in decoding a model that ”encodes” the data. Such data analysis methods as K-means clustering and Principal Component Analysis naturally fall within this category, as described in [28].

In thesis, we review each of the above-mentioned categories of methods for community detection in feature-rich networks separately. Moreover, we briefly review the network-only clustering approaches for the completeness of the thesis, including classical approaches. Refer to the thesis for more information.

Methodologies

In this section, we describe our proposed methods. Based upon the following two criteria, we categorize them into three categories of methods. The categorization criteria are: (a) whether the features are directly used, or they are converted to similarity data; (b) whether the clusters are extracted sequentially, i.e., the clusters are extracted one-by-one, or all clusters are extracted simultaneously.

The first category of methods is devoted to the sequential cluster extraction in feature-rich networks. We name this category of methods "SEFNAC." The second category of methods is the continuation of the research started in the first category of methods. More precisely, we exploit a similar sequential clustering strategy, this time being applied to feature-rich networks using similarity data (the features are converted to similarity data). We name this category of methods "ICESi." Finally, in the third category of approaches, unlike the two previous categories, the clusters are extracted simultaneously at feature-rich networks. We name this category of methods as "KEFRiN."

In thesis within each Subsection, we first describe the motivation behind that category of methods, however we exclude these description from the summary. Then we explain the notation, and finally, we describe the proposed methodology. There is only one exception: in the section we describe the "ICESi" methods, a subsection is devoted to describing the method used to obtain the similarity data, while in the rest of this section, the same structure is preserved.

Sequential data recovery clusters extraction methods

Sequential methods at feature-rich networks

Notation

Consider a network with features at the nodes, $A = \{P, Y\}$, over an entity set I. Here I is a set of network nodes of cardinality $|I| = N$; $P = (p_{ij})$ is an $N \times N$ matrix of mutual link weights between nodes $i, j \in I$; and $Y = (y_{iv})$ is an $N \times V$ matrix of feature values, so that entry y_{iv} is the value of feature $v = 1, 2, ..., V$ at node $i \in I$. This definition covers a wide range of networks, including, for example, a flat network in which the edges simply exist or not but have no associated weights. Such a network can be represented by matrix P such that $p_{ij} = 1$ if a link between i and j exists, and $p_{ij} = 0$ if not.

To build a data-driven community model, let us specify the following notation.

A community, or cluster, $S \subset I$ is represented by a binary $N \times 1$ membership column vector, $s = (s_i)$ in which $s_i = 1$ if $i \in S$, and $s_i = 0$, otherwise ($i = 1, 2, ..., N$).

In the feature space, community S can be represented by a V-dimensional point $c = (c_v)$, which is a standard to which all the community members relate.

At the network link data, there may be at least two possible assumptions:

(a) AS: Summable weights

This assumption means that the weights p_{ij} are comparable and summable across all the matrix P. In this case, there should be a single intensity weight λ to relate the weights measurement scale to S. Specifically, each within-community weight $p_{ij}, i, j \in S$, in this case should be large and approximately equal to the intensity λ. The between-community links, ideally, should be all zero.

(b) AN: Nonsummable weights

Under this assumption, weights p_{ij} in any column j are considered incomparable to weights $p_{ij'}$ in any different column $j' \neq j, i, j \in I$. Therefore, at each column $j \in I$ a specific intensity weight λ_j is assumed, so that, for any $i \in S$ the link weights p_{ij} tend to be equal to λ_j.

This assumption points to a not uncommon data type emerging, for example, in some psychological experiments in which the entities are individuals or cognitive subsystems with different scales of individual judgements. In fact, anytime when links of a node are measured independently of those of the other nodes, there is a potential for the weights being nonsummable.

Of course, a similar assumption regarding the row weights can be formulated – we are going to skip that in this paper.

Extending these definitions to a partition, $S = \{S_1, S_2, ..., S_K\}$, of I in K non-overlapping parts/communities, S can be represented by a binary matrix $s = (s_{ik})$ so that $s_{ik} = 1$ if $i \in S_k$, and $s_{ik} = 0$, otherwise.

To relate any partition to the feature data, we assume that a standard point $c_k = (c_{kv})$ is specified for each community S_k, $k = 1, 2, ..., K$, so that approximate equations hold:

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

Since communities S_k do not overlap, the sum in the equations plays a rather nominal role: for any $i \in I$, y_{iv} is equal to $c_{kv} + f_{iv}$ just for that k at which $i \in S_k$. The value f_{iv} expresses the extent of approximation and should be made as small as possible.

To approximate the network part of the data, we assume either a total intensity weight λ_k for community S_k, under the summability assumption AS, or column-dependent intensity weights λ_{kj}, under the nonsummability assumption AN ($k = 1, 2, ...K; j \in I$). Then the following equations should hold:

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

at the AS, and

$$p_{ij} = \sum_{k=1}^{K} \lambda_{kj} s_{ik} + e_{ij}, i, j \in I. \quad (3)$$

at the AN.

Once again the sums in these equations are purely nominal. At the AS, they just express that $p_{ij} = \lambda_k$ for all $i, j \in S_k$ ($k = 1, 2, ..., K$) or $p_{ij} = 0$, otherwise, up to the residual e_{ij}, of course. At the AN, $p_{ij} = \lambda_{kj}$ for $i \in S_k$ and any $j \in I$, up to small residual e_{ij} again. One may consider that at the AN assumption, the columns $j \in I$ play roles of features.

By using the least-squares approach, we formulate the problem of finding a hidden membership matrix $s = (s_{ik})$, community centers c_k, and intensity weights λ_k or λ_{kj}, as of minimizing the sum of squared residuals:

- at AS assumption:

$$F_{AS}(\lambda_k, s_k, c_k) = \rho \sum_{k=1}^{K} \sum_{i,v} (y_{iv} - c_{kv} s_{ik})^2 + \xi \sum_{k=1}^{K} \sum_{i,j} (p_{ij} - \lambda_k s_{ik} s_{jk})^2, \quad (4)$$

- at AN assumption:

$$F_{AN}(\lambda_k, s_k, c_k) = \rho \sum_{k=1}^{K} \sum_{i,v} (y_{iv} - c_{kv} s_{ik})^2 + \xi \sum_{k=1}^{K} \sum_{i,j} (p_{ij} - \lambda_{kj} s_{ik})^2. \quad (5)$$

The factors ρ and ξ in Eqn. (4) and (5) are expert-driven constants to balance the relative weights of the two sources of data, network links and feature values.

Since vectors $s_k = (s_{ik})$ ($k = 1, 2, ..., K$) correspond to a partition, they are mutually orthogonal. That means that for any specific i, s_{ik} is zero for all k's except one: that one k for which S_k contains i. As a result, each of the sums over k in the models relates to a single summand, meaning that the operation of summation over k may be applied outside of the parentheses in Eqn. (4) and (5).

Methodology

The problems of optimization of criteria (4) and (5) are computationally intensive and cannot be solved exactly in a reasonable time. Therefore, there can be various heuristic strategies explored to locally or approximately advance to solving them. We are going to exploit a doubly greedy approach of sequential extraction [27]. This approach can be applied here because the criteria to optimize are additive. According to this approach, parts S_k of the partition S are sought not simultaneously, but one-by-one, sequentially, in a greedy manner. That is, a subset of I to serve as S_k at $k = 1$ is found to minimize the part of the criterion related to S_1. Specifically, for an individual community denoted by $T \subseteq I$, its membership by $t = (t_i)$, so that $t_i = 1$ if $i \in T$ and $t_i = 0$, otherwise; its center in feature space, by c; and the corresponding intensity weight by λ (the index k has been removed), the extent of fit between the community and the dataset, according to criteria (4) and (5), is

$$f_{AS}(\lambda, c_v, t_i) = \rho \sum_{i,v} (y_{iv} - c_v t_i)^2 + \xi \sum_{i,j} (p_{ij} - \lambda t_i t_j)^2 \quad (6)$$

at the assumption AS, or

$$f_{AN}(\lambda_j, c_v, t_i) = \rho \sum_{i,v} (y_{iv} - c_v t_i)^2 + \xi \sum_{i,j} (p_{ij} - \lambda_j t_i)^2 \quad (7)$$

at the assumption AN.

A T locally or approximately minimizing the corresponding criterion (6) or (7) is taken as the first part of partition S, S_1. Then this S_1 is removed from I and the next part, S_2, is sought in the same way over the residual entity set $I \leftarrow I - S_1$. This continues till a prespecified stopping criterion is reached such as, say, when the residual I gets empty.

Consider a method, *Ext(D)*, for extracting a subset $T \subseteq D$ from any $D \subseteq I$, together with some related quantitative characteristics α, so that (T, α) =*Ext(D)*. Of course, P and Y remain the only data sources used in Ext. A greedy Sequential Extraction procedure SE can be formulated as follows:

SE algorithm

Input: set I; define stop condition Φ either in terms of I (say, $I == \emptyset$) or in terms of α or both.

Output: partition $S = \{S_1, S_2, ..., S_K\}$ of I in nonintersecting parts (communities) S_k, as well as their characteristics α_k, $k = 1, 2, ..., K$, where $K > 0$ is an integer determined as a result of running the algorithm.

Step 1. Define $k = 1$, $D = I$.

Step 2. Apply $(T, alpha) = Ext(D)$ and set $S_k = T$, $\alpha_k = alpha$.

Step 3. Redefine $D = D - S_k$. If stop condition Φ is true, set $K = k$ and stop. Otherwise, define $k = k + 1$ and go to Step 2.

Within this greedy strategy, at its k-th step ($k = 1, 2, ..., K$), we use one more greedy procedure for obtaining a (locally) optimal part $T = S_k$ and its quantitative characteristic α_k. According to this procedure, the set S_k, along with its quantitative characteristic c_k, λ_k, at AS, or c_k, λ_{jk} at AN, is found not in one go, but by greedily adding elements of I to S_k one-by-one. The additive structure of the criteria (6) and (7) above allows us to express them using contributions to the data scatter, which, to an extent, guides the process, as explained below. Besides its computational simplicity, the sequential extraction approach has some theoretical and practical advantages.

One of the theoretical advantages is a Pythagorean decomposition of the data scatter – this allows to score the contribution of various elements of found solutions to the data scatter, which is useful for interpretation [28]. Among practical advantages is competitiveness of the approach regarding the quality of cluster recovery against other computational procedures (see, for example, experimental results of realizations of the doubly greedy strategy in different situations in [10, 28, 29]).

To apply this strategy here, denote the indicator vector of a community T by $t = (t_i)$; its center in the feature space, by $c = (c_v)$; and the corresponding intensity weights by λ and λ_j

depending on the assumption, AS or AN, respectively (the index k is removed because it is not needed here).

Consider three individual items constituting the squared error criteria (6) and (7):

(a) The fit between the feature data and the community and its standard point:

$$F_Y(c,t) = \sum_{i,v}(y_{iv} - c_v t_i)^2 \tag{8}$$

(b) The fit between the AS community model and network data:

$$F_{PS}(\lambda,t) = \sum_{i,j}(p_{ij} - \lambda t_i t_j)^2, \tag{9}$$

(c) The fit between the AN community model and network data:

$$F_{PN}(\lambda,t) = \sum_{i,j}(p_{ij} - \lambda_j t_i)^2. \tag{10}$$

The total goodness of fit measure is either $f_{AN} = \rho F_Y + \xi F_{PS}$ (in criterion (6)) or $f_{AN} = \rho F_Y + \xi F_{PN}$ (in criterion (7)). Recall that ρ and ξ are weights to balance two data sources, the features and the links, respectively.

At a specified subset $T \subseteq I$, to minimize the criteria (6) and (7) regarding the quantitative characteristics c_v, λ, λ_j, one may separately minimize the individual parts (8) over c_v, (9) over λ, and (10) over λ_j because of the additive structure of the criteria (6) and (7).

Since each of these three is quadratic regarding the respective numerical characteristic c_v, λ, λ_j, the optimal solutions can be found from the first-order optimality conditions. Let us take the derivatives of F_Y with respect to c_v, F_{PS} with respect to λ, and F_{PN} with respect to λ_j:

$$\frac{\partial F_Y}{\partial c_v} = 2\sum_i (y_{iv} - c_v t_i)(-t_i), \tag{11}$$

$$\frac{\partial F_{PS}}{\partial \lambda} = 2\sum_{i,j}(p_{ij} - \lambda t_i t_j)(-t_i t_j). \tag{12}$$

$$\frac{\partial F_{PN}}{\partial \lambda_j} = 2\sum_i (p_{ij} - \lambda_j t_i)(-t_i). \tag{13}$$

Equating each of these to zero would yield, in respect, equations:

$$\sum_i y_{iv} t_i = c_v \sum_i t_i^2, \tag{14}$$

$$\sum_{i,j} p_{ij} t_i t_j = \lambda \sum_i t_i^2 \sum_j t_j^2, \tag{15}$$

and

$$\sum_i p_{ij} t_i = \lambda_j \sum_i t_i^2. \tag{16}$$

Since t_i is 1/0 binary, equality $t_i^2 = t_i$ holds. Thus, $\sum_i t_i^2 = \sum_j t_j^2 = \sum_i t_i = |T|$. Therefore, these equations can be equivalently reformulated as follows:

$$c_v = \frac{\sum_i y_{iv} t_i}{|T|} = \frac{\sum_{i \in T} y_{iv}}{|T|}, \tag{17}$$

$$\lambda = \frac{\sum_{i,j} p_{ij} t_i t_j}{|T|^2} = \frac{\sum_{i,j \in T} p_{ij}}{|T|^2}, \tag{18}$$

and

$$\lambda_j = \frac{\sum_i p_{ij} t_i}{|T|} = \frac{\sum_{i \in T} p_{ij}}{|T|}. \tag{19}$$

In other words, the optimal c_v and λ_j at AN must be central in T: they are within-cluster means of features v and network link columns j. Similarly, at AS, the optimal intensity value λ is equal to the mean within-cluster link value.

Let us now reformulate criteria (8), (9), (10) by opening the parentheses and putting there the found optimal values of c_v, λ, λ_j:

Criterion (8) yields:

$$F_Y(c,t) = \sum_{i,v} (y_{iv} - c_v t_i)^2 = \sum_{i,v} (y_{iv}^2 - 2 y_{iv} c_v t_i + c_v^2 t_i) = \sum_{i,v} y_{iv}^2 - 2 \sum_v c_v \sum_i (y_{iv} t_i) + \sum_v c_v^2 |T|$$

Let us denote the square Y scatter by $Q(Y) = \sum_{i,v} y_{iv}^2$ and take into account that $\sum_i y_{iv} t_i = c_v |T|$ and $\sum_i t_i = |T|$. Then the equation above can be rewritten as

$$F_Y(c,t) = Q(Y) - \sum_v c_v^2 |T| \tag{20}$$

Criterion (9) yields:

$$F_{PS}(\lambda, t) = \sum_{i,j} (p_{ij} - \lambda t_i t_j)^2 = \sum_{i,j} p_{ij}^2 - 2\lambda \sum_{i,j} p_{ij} t_i t_j + \lambda^2 \sum_{i,j} t_i t_j.$$

Let us denote the square P scatter by $Q(P) = \sum_{i,j} p_{ij}^2$ and take into account that $\sum_{i,j} p_{ij} t_i t_j = \lambda \sum_{i,j} t_i t_j$. Then the equation above can be rewritten as

$$F_{PS}(\lambda, t) = Q(P) - \lambda^2 |T|^2 \tag{21}$$

Similarly, criterion (10) yields:

$$F_{PN}(\lambda, t) = \sum_{i,j}(p_{ij} - \lambda_j t_i)^2 = \sum_{i,j} p_{ij}^2 - 2\sum_{i,j} p_{ij} t_i \lambda_j + \sum_j \lambda_j^2 \sum_i t_i.$$

Let us take into account that $\sum_i p_{ij} t_i = \lambda_j \sum_i t_i$. Then the equation above can be rewritten as

$$F_{PN}(\lambda, t) = Q(P) - \sum_j \lambda_j^2 |T|. \tag{22}$$

Therefore, with the optimal values for c_v, λ, and λ_j determined by T in Eqn. (17), (18), and (19), respectively, the criteria (6) and (7) can be equivalently reformulated as

$$f(\lambda, c_v, t_i) = \rho Q(Y) + \xi Q(P) - G \tag{23}$$

where λ is either a scalar or vector, and

$$G(T) = G_s = \rho |T| \sum_v c_v^2 + \xi \lambda \sum_{ij} p_{ij} t_i t_j \tag{24}$$

at the assumption AS, and

$$G(T) = G_n = |T|(\rho \sum_v c_v^2 + \xi \sum_j \lambda_j^2) \tag{25}$$

at the assumption AN, where c_v, λ, and λ_j are determined by T according to equations (17), (18), and (19), respectively.

Maximizing criterion $G(T)$ in Eqn. (24) and (25) is equivalent to minimizing the one-cluster least-squares criteria in Eqn. (6) and (7). Therefore, it makes sense to take a look whether $G(T)$ has any meaning of its own.

First of all, we have to recognise that the equation (23) can be rewritten as a Pythagorean decomposition of the combined data scatter $\rho Q(Y) + \xi Q(P)$:

$$\rho Q(Y) + \xi Q(P) = G + f \tag{26}$$

in two parts, the minimized square residuals f and the remaining part G. This gives the meaning to the value of G. This is contribution of the community T to the combined data scatter.

By looking at the formulas for G, we can see that the part related to the feature set, which is the same in both expressions for $G(T)$, (24) and (25), requires maximizing both $|T|$ and the squared distance between c and 0, $\sum_v c_v^2$. That means that an optimal T should have as many elements as possible and, simultaneously, be as far away from 0 as possible in the feature space. Assuming that the feature data are pre-processed so that the origin is transferred to the center of gravity, or grand mean, the point whose components are the averages of the corresponding features, we may conclude that the cluster T should be both numerous and anomalous. The second item in each of the criteria, G_s (24) and G_n (25), has a similar meaning regarding the network data.

Hence, we refer to our local search algorithm for maximizing (24) or (25) as to the Feature-Rich Network Addition Clustering algorithm, FNAC, using endings FNACs and FNACn if necessary, to point out which of the criteria (24) and (25), respectively, is maximized. The algorithm finds a cluster T, its center c, and its intensity weight(s) λ (λ_j) by locally maximizing $G(T)$ in the system of neighborhoods defined by the following condition. Given a current T, its neighborhood consists of subsets differing from T by just adding a single entity.

The algorithm starts from a random $i \in I$. This i serves as the seed forming a singleton cluster $T = \{i\}$. This triggers execution of the base FNAC module. At any current T, this module computes increment $\Delta(j) = G(T+j) - G(T)$ for every element $j \in I - T$ and selects that $j*$ at which $\Delta(j)$ is maximum. If this maximum is positive, then $j*$ is added to T, and the module runs again from thus updated T. If, in contrast, $\Delta(j*) < 0$, the algorithm halts and outputs T, its center c, its link intensity λ (or intensities λ_j), and its contribution to the combined data scatter G. Then the last check is performed: **Seed Relevance Check:** If the removal of the seed increases the cluster contribution; this seed is extracted from the cluster.

The algorithm FNAC above, in its versions FNACs and FNACn, serves as the core subroutine Ext in our community detection algorithm SE above. The algorithm SE involves an internal procedure, $(T, \alpha) = Ext(D)$ where $D \subseteq I$. By using FNAC as the algorithm Ext to output the community T along with its parameters c_v and λ/λ_j constituting the α, we obtain a combined algorithm, SEFNAC.

A source code of SEFNACs and SEFNACn, as well as all other supplementary materials, including the real-world data sets, synthetic data generator etc. are publicly available in `https://github.com/Sorooshi/SEFNACsSEFNACn`.

Sequential methods at feature-rich networks using similarity data

Inner products as similarities

Our approach assumes a preliminary standardization of the data, both the network and feature spaces. The features are standardized by subtracting the means $g_v = \sum_{i \in I} y_{iv}/N$ from feature columns v, $v = 1, 2, ..., V$. To distinguish g_v from within-cluster means, they frequently are referred to as grand means. We accept the row-to-row inner product $r_{ij} =< y_i, y_j >= \sum_{v \in V} y_{iv} y_{jv}$ as the similarity index. Each feature v contributes the product $y_{iv} y_{jv}$ to this, which much depends on the mutual location of i and j nodes on the axis v with respect to the grand mean g_v.

The product is positive when both node location are either larger than $g_v = 0$ or smaller than $g_v = 0$. It is negative when i and j are on different sides from $g_v = 0$. Furthermore, the closer y_{iv} and y_{jv} to zero the smaller the product and the farther they are from zero, the greater the product.

Scoring the similarity by the inner product makes those entities in which features are further away from the grand mean, more distinguishable. In contrast, those entities in which feature values are close to the grand mean are less distinguishable, therefore they might be merged during the clustering process.

Notation

As explained above, we have two $N \times N$ data matrices, matrix $R = (r_{ij})$ of feature-based node-to-node similarities and matrix $P = (p_{ij})$ of node-to-node link scoring. To unify our presentation, we are going to denote either of them as $B = (b_{ij})$ where b_{ij} stands for either a converted feature-based similarity r_{ij} or a 'native' link weight p_{ij} ($i, j \in I$).

To define our data-driven community model, let us specify the following notation.

A community, or cluster, $T \subset I$ is represented by a binary $N \times 1$ membership column vector, $t = (t_i)$ in which $t_i = 1$ if $i \in T$, and $t_i = 0$, otherwise.

Assume that there may be two possible modes of using the similarity scores b_{ij}:

SM Summability Mode

In this mode, the similarities b_{ij} are comparable and summable across the entire matrix B. In this case, there should an intensity value η to relate the similarity measurement scale to T. Specifically, each within-community similarity b_{ij} $i, j \in T$, should be approximately equal to the intensity η for T.

NM Nonsummability Mode

In this mode, the similarities b_{ij} in any column j are assumed to be non-comparable to similarities $b_{ij'}$ in any different column $j' \neq j, i, j \in I$. Therefore, a specific intensity η_j is assumed for each column $j \in I$, so that, for any $i \in T$ the similarity value b_{ij} should approximate the value η_j.

The SM mode is typical in network analysis. NM mode points to not an uncommon data type emerging in some psychological experiments in which the nodes are individuals or cognitive subsystems with different scales of individual judgements. Similarly, between-industries input-output tables in Economics may use different measurement scales for production of different industries, especially for raw materials such as electricity, coal, and oil. The similarity data derived from the feature tables also may be considered as measured in NM mode sometimes, especially in potentially important situations at which some nodes j may be considered as more important than the other – then similarity to each of them could serve as that measured in a different scale.

To relate a community T to the similarity data B, we assume that a unified intensity η exists in the SM mode or a set of intensity values η_j, $j \in I$, in the NM mode, so that either of the two following approximate equations holds:

$$b_{ij} = \eta s_i s_j + e_{ij}, i, j \in I, \tag{27}$$

at the SM, or

$$b_{ij} = \eta_j s_i + e_{ij}, i, j \in I. \tag{28}$$

at the NM assumption.

Since there are two sources of data, namely, the feature-based similarity data and the network data, and for each of them either of the two modes can be accepted, consequently, there will

be four possible combinations of modes and data sources. As a convention, we assume that symbol "S" stands for the summability mode, and "N" stands for the nonsummability mode, at each of the data sources, so that the first letter refers to the feature-based similarity data, whereas the second letter refers to the network data. Consequently, combination SS refers to the case at which both data sets are in the Summable mode; SN, to the case at which the feature-based similarity data are Summable and the network data are not; NS, to the case at which the feature-based similarity data are Non-summable and the network data are Summable; NN, to the case at which both data sets are in the Nonsummable mode. To avoid repetitive derivations, we consider in detail only one of the four cases, say, SN.

By using the least-squares approach, we arrive at the problem of finding a hidden membership matrix $s = (s_{ik})$, intensities for the similarity data μ_k and intensity weights λ_{jk} minimizing the sum of squared residuals according to the SN mode:

at SN assumption:

$$F_{SN}(s_k, \mu_k, \lambda_{jk}) = \rho \sum_{k=1}^{K} \sum_{i,j} (r_{ij} - \mu_k s_{ik} s_{jk})^2 + \xi \sum_{k=1}^{K} \sum_{i,j} (p_{ij} - \lambda_{kj} s_{ik})^2, \tag{29}$$

The factors ρ and ξ in Eqn. (29) are expert-driven constants to balance the relative weights of the two sources of data, network links and feature-based similarity values.

Since vectors $s_k = (s_{ik})$ $(k = 1, 2, ..., K)$ correspond to a partition, they are mutually orthogonal. That means that for any specific i, s_{ik} is zero for all k's except one: that one k for which S_k contains i. As a result, each of the sums over k in the models relates to a single summand, meaning that the operation of summation over k may be applied outside of the parentheses in Eqn. (29).

Methodology

Global optimization of the criterion (29) is computationally expensive and cannot be achieved in a reasonable time. Therefore, there can be various heuristic strategies applied. We are going to exploit a doubly greedy approach of sequential extraction [27]. This approach can be applied here because the criteria to optimize are additive. According to this approach, parts S_k of the partition S are sought not simultaneously but one-by-one, sequentially, in a greedy manner. That is, a subset of I to serve as S_k at $k = 1$ is found to minimize the part of the criterion related to S_1.

Specifically, for an individual community denoted by $T \subseteq I$, its membership by $t = (t_i)$, so that $t_i = 1$ if $i \in T$ and $t_i = 0$, otherwise; its intensity similarity by μ; and the corresponding intensity weight by λ_j (the index k has been removed), the extent of fit between the community and the dataset, according to criterion (29), is

$$f_{SN}(\mu, \lambda_j, t_i) = \rho \sum_{i,j} (r_{ij} - \mu t_i t_j)^2 + \xi \sum_{i,j} (p_{ij} - \lambda_j t_i)^2 \tag{30}$$

We take a subset T minimizing, in some sense, the criterion (30) as the first part of partition S we are to find, S_1. Then this S_1 is removed from I, and the next part, S_2, is sought in the same way over the residual entity set $I' \leftarrow I - S_1$. This procedure continues till a prespecified stopping criterion is reached, such as, say, that the residual I' gets empty.

Within this greedy strategy, at its k-th step ($k = 1, 2, ..., K$), we use one more greedy procedure for obtaining a (locally) optimal set T and its quantitative characteristics μ and λ_j, $j \in I$. The additive structure of the criterion (30) allows us to express them using contributions to the data scatter.

Consider two partial criteria, the two individual items in the squared error criterion (30):

(a) The fit between the summable community model and the similarity data:

$$F_{RS}(\mu, t) = \sum_{i,j} (r_{ij} - \mu t_i t_j)^2 \tag{31}$$

(b) The fit between the nonsummable community model and the network data:

$$F_{PN}(\lambda, t) = \sum_{i,j} (p_{ij} - \lambda_j t_i)^2. \tag{32}$$

The total goodness of fit measure is $f_{SN} = \rho F_{RS} + \xi F_{PN}$ where ρ and ξ are user-defined weights balancing two data sources, the feature-based similarities, and the network links, respectively.

At a given $T \subseteq I$, to minimize the criterion (30) with respect to the quantitative characteristics μ and λ_j, one should apply the first-order optimality conditions. The derivatives of f_{SN} over μ and λ_j are:

$$\frac{\partial f_{SN}}{\partial \mu} = 2\rho \sum_{i,j} (r_{ij} - \mu t_i t_j)(-t_i t_j). \tag{33}$$

and

$$\frac{\partial f_{SN}}{\partial \mu_j} = 2\xi \sum_{i,j} (r_{ij} - \lambda_j t_i)(-t_i). \tag{34}$$

Equating them to zero yields:

$$\sum_{i,j} r_{ij} t_i t_j = \mu \sum_i t_i^2 \sum_j t_j^2, \tag{35}$$

and

$$\sum_i p_{ij} t_i = \lambda_j \sum_i t_i^2. \tag{36}$$

Since t_i is 1/0 binary, equality $t_i^2 = t_i$ holds. Thus, $\sum_i t_i^2 = \sum_j t_j^2 = \sum_i t_i = |T|$. Therefore, these equations can be equivalently reformulated as follows:

$$\mu = \frac{\sum_{i,j} r_{ij} t_i t_j}{|T|^2} = \frac{\sum_{i,j \in T} r_{ij}}{|T|^2}, \tag{37}$$

and

$$\lambda_j = \frac{\sum_i p_{ij} t_i}{|T|} = \frac{\sum_{i \in T} p_{ij}}{|T|}. \tag{38}$$

In other words, the optimal μ and λ_j must be central in T: they are within-cluster means of the corresponding similarity and link scoring values.

Let us now reformulate the partial criteria (31) and (32) by opening the parentheses and putting there the found optimal values of μ and λ_j:

Criterion (31) yields:

$$F_{RS}(\mu, t) = \sum_{i,j} (r_{ij} - \mu t_i t_j)^2 = \sum_{i,j} r_{ij}^2 - 2\mu \sum_{i,j} r_{ij} t_i t_j + \mu^2 \sum_{i,j} t_i t_j.$$

Let us denote the square R matrix scatter by $Q(R) = \sum_{i,j} r_{ij}^2$ and take into account that $\sum_{i,j} r_{ij} t_i t_j = \mu \sum_{i,j} t_i t_j$. Then the equation above can be rewritten as

$$F_{RS}(\mu, t) = Q(R) - \mu^2 |T|^2 \tag{39}$$

Similarly, criterion (32) yields:

$$F_{PN}(\lambda_j, t) = \sum_{i,j} (p_{ij} - \lambda_j t_i)^2 = \sum_{i,j} p_{ij}^2 - 2 \sum_{i,j} p_{ij} t_i \lambda_j + \sum_j \lambda_j^2 \sum_i t_i.$$

Let us take into account that $\sum_i p_{ij} t_i = \lambda_j \sum_i t_i$. Then the equation above can be rewritten as

$$F_{PN}(\lambda, t) = Q(P) - \sum_j \lambda_j^2 |T|. \tag{40}$$

where $Q(P) = \sum_{i,j} p_{ij}^2$ is the data P scatter.

Therefore, with the optimal values for μ, and λ_j, the criterion (30) can be equivalently reformulated as:

$$f(\mu, \lambda, t) = \rho Q(R) + \xi Q(P) - G \tag{41}$$

where

$$G(T) = G_{SN} = \rho\mu \sum_{ij} r_{ij} t_i t_j + \xi |T| \sum_j \lambda_j^2 = \rho |T|^2 \mu^2 + \xi |T| \sum_j \lambda_j^2 \tag{42}$$

where $\mu = \frac{\sum_{i,j} r_{ij} t_i t_j}{\sum_{i,j} t_i t_j}$ and $\lambda_j = \frac{\sum_{i \in T} p_{ij}}{|T|}$.

Maximizing criterion $G(T)$ in the Eqn. (42) is equivalent to minimizing the corresponding one-cluster least-squares criteria Eqn. (30). Therefore, it makes sense to see whether $G(T)$ has any meaning of its own.

First of all, we can rewrite the equation (41) as a Pythagorean decomposition of the combined data scatter $Q(R, P) = \rho Q(R) + \xi Q(P)$:

$$Q(R, P) = \rho Q(R) + \xi Q(P) = G + f \tag{43}$$

in two parts, the minimized squared residuals f (30) and the complementary part G. The decomposition gives a statistical meaning to the value of G. This is the community's contribution T to the combined data scatter $Q(R, P)$.

A more intuitive meaning of the criterion one can see in the formula (42): it requires maximizing the size $|T|$ of the community to be found and, simultaneously, maximizing the average within-community similarity and the squared distance from the vector (λ_j) to 0.

Assuming that the data matrices are pre-processed so that the origin is transferred to the center of gravity, or grand mean, the point whose components are the averages of the corresponding similarity/network values, we may conclude that the cluster T should be both numerous and anomalous.

We refer to our local search algorithm for maximizing criterion (42) as to the Least-Squares Community Extraction from Similarity data, LS CESi, or just CESi when the least-squares framework is assumed undoubtedly. We add to this an ending, sn, to indicate in the modified abbreviation CESisn, the summability mode accepted for the feature-based similarity and nonsummability mode accepted for the network links, s for SM and n for NM, in the case under consideration. The other three combinations will be referred to as CESiss, CESins, and CESinn, to mean combinations SM and SM, NM and SM, and NM and NM, respectively.

The algorithm finds a cluster T and its intensities μ and λ_j by locally maximizing G in the system of neighborhoods defined by the condition that T's neighborhood consists of subsets differing from T by just adding a single entity.

The CESi algorithm starts from a random $i \in I$. This i serves as the seed forming a starting singleton cluster $T = \{i\}$. This triggers the execution of the base CESi module. At any current T, this module computes increment $\Delta(j) = G(T+j) - G(T)$ for every element $j \in I - T$ and selects that j^* at which $\Delta(j)$ is maximum. If this maximum is positive, then j^* is added to T, and the module runs again from thus updated T. If, in contrast, $\Delta(j^*) < 0$, the algorithm halts and outputs T and its intensities μ and λ_j, as well as its contribution to the combined data, scatter G. Then the last check is performed: **Seed Relevance Check:** If the seed's removal increases the cluster contribution, this seed is extracted from the cluster.

The algorithm CESi serves as the core subroutine in our Iterative community detection algorithm ICESi.

The algorithm ICESi starts by standardizing the square $N \times N$ matrices R and P – this will be described later. Then we set $k = 1$ and $I_k = I$. At a given k, we apply CESi to R and P data matrices restricted to the set I_k. The resulting cluster T forms the next cluster S_{k+1} along with its intensities μ_{k+1} and $\lambda_{j,k+1}$, as well as the relative contribution to the combined data scatter $q_{k+1} = G/Q(R, P)$. Now we redefine $I_{k+1} = I_k - S_{k+1}$ and test a pre-specified stop-condition. The stop-condition is a predicate that may involve several clauses. One of them is testing whether $I_{k+1} = \emptyset$ or not. Two other clauses usually are limits to the current and cumulative contributions. To stop, the former should be less than, say, 5% of the $Q(R, P)$, whereas the latter should be 50% of that or greater. If the stop condition is satisfied, we define $K = k + 1$ and output the found clusters S_k together with their numerical characteristics ($k = 1, 2, ..., K$). Otherwise, we update k by adding 1, $k \leftarrow k + 1$ and execute the next iteration of extracting clusters.

Algorithms ICESiss, ICESins, ICESinn corresponding to other combinations of summability modes also use the decomposition (41) to maximize the contribution G, that is expressed either as

$$G(T) = G_{SS} = \rho\mu \sum_{ij} r_{ij} t_i t_j + \xi\lambda \sum_{ij} p_{ij} t_i t_j \tag{44}$$

at the combination SM and SM, or as

$$G(T) = G_{NS} = \rho|T| \sum_{j} \mu_j^2 + \xi\lambda \sum_{ij} p_{ij} t_i t_j \tag{45}$$

at the combination NM and SM, or as

$$G(T) = G_{NN} = |T|(\rho \sum_{j} \mu_j^2 + \xi \sum_{j} \lambda_j^2) \tag{46}$$

at the combination NM and NM.

at the assumption NN, where μ, μ_j, λ, and λ_j are the corresponding within-T means. The algorithm ICESi works with them similarly, up to obvious modifications of the increment $\Delta(j)$.

A Python source code of thus defined ICESi can be found at `https://github.com/Sorooshi/ICESi`.

Simultaneous data recovery clusters extraction methods

Simultaneous methods at feature-rich network

Notation

Pursuing the similar notation, we consider a network with features at the nodes, $A = \{P, Y\}$, over an entity set I. Here I is a set of network nodes of cardinality $|I| = N$; $P = (p_{ij})$ is an $N \times N$ matrix of mutual link weights between nodes $i, j \in I$; and $Y = (y_{iv})$ is an $N \times V$ matrix of feature values, so that entry y_{iv} is the value of feature $v = 1, 2, ..., V$ at node $i \in I$. This definition covers a wide range of networks, including, for example, a flat network in which the

edges, whether exist or not but have no associated weights. Such a network can be represented by matrix P such that $p_{ij} = 1$ if a link between i and j exists, and $p_{ij} = 0$ if not.

Regarding the introduced community definition: there should be a set of associations between the features of a node and its links with other nodes of that community –which is to be discovered. Moreover, we assume that both data tables are (only) column-wise summable. Concretely, any p_{ij} and any y_{iv} ($i, j \in I, v \in V$) is only summable and comparable along the *j-th* column and *v-th* column of matrix P and matrix Y respectively [2].

To build a data-driven community model let us proceed as follows.

Partitioning I into K crisps and non-overlapping communities implies $S = \{S_1, S_2, ..., S_k\}$: such that the community S_k is represented by a $N \times 1$ column vector $s_k = (s_{ik})$, where $s_{ik} = 1$ if $i \in S_k$ and $s_{ik} = 0$, otherwise ($i \in I$).

In the feature space, to related the notion of members of the community having similar feature values, we consider the standard V-dimensional point of S_k, $c_k = (c_{kv})$ ($v \in V, k = 1, ..., K$). Then the following approximate equation should hold:

$$y_{iv} = \sum_{k=1}^{K} c_{kv} s_{ik} + f_{iv}, \quad i \in I, v \in V. \tag{47}$$

Where the value f_{iv} expresses the extent of approximation and should be made as small as possible.

And similarly, in the network data space to related the notion of members of the community having, on average, in-common connections we consider the standard N-dimensional points of S_k, that is, $\lambda_k = (\lambda_{kj})$ ($j \in I, k = 1, ..., K$). Therefore, the following approximate equation should hold:

$$p_{ij} = \sum_{k=1}^{K} \lambda_{kj} s_{ik} + e_{ij} \quad i, j \in I. \tag{48}$$

Where, again, the value e_{ij} expresses the extent of approximation and should be made as small as possible.

By using the least-squares approach, we formulate the problem of finding a hidden membership matrix $s = (s_{ik})$, community centers in the feature data space $c_k = (c_{kv})$, and community centers in the network data space $\lambda_k = (\lambda_{kj})$, as of minimizing the sum of squared residuals:

$$\begin{aligned} F(s_{ik}, c_{kv}, \lambda_{kj}) = \rho \sum_{i,v} (y_{iv} - \sum_{k=1}^{K} c_{kv} s_{ik})^2 \\ + \xi \sum_{i,j} (p_{ij} - \sum_{k=1}^{K} \lambda_{kj} s_{ik})^2. \end{aligned} \tag{49}$$

[2] other possible combination of summable and nonsummable assumptions have been considered in our previous work. And it ought to mention that we leave the row summable assumption as on matrix P a trivial case, while such assumption for matrix Y is not feasible.

The factors ρ and ξ in Eqn. (49) are expert-driven constants to balance the relative weights of the two sources of data, network links and feature values.

Methodology

Optimizing criterion (49), as well as other least-square criteria, is computationally intensive and cannot be solved exactly in a reasonable time. Consequently, various heuristic strategies have been proposed to advance the best possible approximation. In this work, we are going to adopt the so-called alternating optimization strategy, more specifically, the K-Means algorithm [26]. This approach can be applied here because the criterion to optimize is additive.

K-Means algorithm consists of two main steps, namely, clusters update and centroids update. At first, $2 \times K$ centroids, corresponding to the feature space and to the network space of parts S_k of the partition S, are initialized. In the clusters update step, by utilizing a pre-specified distance metric, each node's distance in the features space and in the network space from all of the corresponding centroids is computed. Then each node will be assigned to a cluster in which it has the minimum distance both in feature space and network space. Once all nodes are assigned to a cluster, the algorithm proceeds to the second step, centroids update. In this step, the clusters' centroids will be updated –by recomputing the average of clusters. The algorithm alternates between these two steps until a predefined stopping condition(s), say the maximum number of iterations, is satisfied.

To pursue this strategy here, let us consider two individual terms constituting our proposed clustering criterion (49):

a) The fit between the feature data space, the community and its centroid:

$$F_Y(c_{kv}, s_{ik}) = \sum_{i,v} (y_{iv} - \sum_k c_{kv} s_{ik})^2, \tag{50}$$

b) The fit between the network data space, the community and its centroid:

$$F_P(\lambda_{kj}, s_{ik}) = \sum_{i,j} (p_{ij} - \sum_k \lambda_{kj} s_{ik})^2. \tag{51}$$

Clearly, the total goodness of fit measure is $f = \rho F_Y + \xi F_P$. Recall that ρ and ξ are user-defined coefficients to balance the impact of the features and the links during the clustering process.

Since the proposed clustering criterion is quadratic regarding its characteristics, i.e. c_{kv} and λ_{kj}, we can find the optimal solutions by applying the first-order optimality conditions. Therefore, taking derivative of f w.r.t c_{kv} and λ_{kj} implies:

$$\frac{\partial f}{\partial c_{kv}} = 2\rho \sum_i (y_{iv} - c_{kv}(s_{ik}))(-s_{ik}), \tag{52}$$

$$\frac{\partial f}{\partial \lambda_{kj}} = 2\xi \sum_i (p_{ii} - \lambda_{kj}(s_{ik}))(-s_{ik}). \tag{53}$$

Equating them to zero, in respect, yields:

$$\sum_i y_{iv} s_{ik} = c_{kv} \sum_i s_{ik}^2, \tag{54}$$

and

$$\sum_i p_{ij} s_{ik} = \lambda_{kj} \sum_i s_{ik}^2. \tag{55}$$

Since s_{ik} is 0/1 binary, equality $s_{ik}^2 = s_{ik}$ holds. Thus, $\sum_i s_{ik}^2 = \sum_i s_{ik} = |S_k|$. Obviously, $|S_k|$ represents the cluster cardinality of S_k. Therefore, the two aforementioned equations are equivalent to:

$$c_{kv} = \frac{\sum_i y_{iv} s_{ik}}{|S_k|} = \frac{\sum_{i \in S_k} y_{iv}}{|S_k|}, \tag{56}$$

and

$$\lambda_{kj} = \frac{\sum_i p_{ij} s_{ik}}{|S_k|} = \frac{\sum_{i \in S_k} p_{ij}}{|S_k|}. \tag{57}$$

In other words, the optimal c_{kv} and λ_{kj} represent the within-cluster means of the features v and the network link columns j, respectively.

Criteria (50) and (51) can be further reformulated by opening the parentheses and using the found optimal values of c_{kv} and λ_{kj}:

$$\begin{aligned} F_Y(s_{ik}, c_{kv}) = \sum_{i,v} (y_{iv} - \sum_k c_{kv} s_{ik})^2 &= \sum_{i,v} (y_{iv}^2 - 2 y_{iv} \sum_k c_{kv} s_{ik} + \sum_k c_{kv}^2 s_{ik}) \\ &= \sum_{i,v} y_{iv}^2 - 2 \sum_k \sum_v c_{kv} \sum_i (y_{iv} s_{ik}) + \sum_k \sum_v c_{kv}^2 |S_k|, \end{aligned}$$

and similarly,

$$\begin{aligned} F_P(s_{ik}, \lambda_{kj}) = \sum_{i,j} (p_{ij} - \sum_k \lambda_{kj} s_{ik})^2 &= \sum_{i,j} (p_{ij}^2 - 2 p_{ij} \sum_k \lambda_{kj} s_{ik} + \sum_k \lambda_{kj}^2 s_{ik}) \\ &= \sum_{i,j} p_{ij}^2 - 2 \sum_k \sum_j \lambda_{kj} \sum_i (p_{ij} s_{ik}) + \sum_k \sum_j \lambda_{kj}^2 |S_k|. \end{aligned}$$

Let us denote the quadratic scatters of the data matrices Y and P with $T(Y) = \sum_{i,v} {y_{iv}}^2$ and $T(P) = \sum_{ij} p_{i,j}^2$ respectively. Moreover, recalling that: 1) $\sum_i y_{iv} s_{ik} = c_{kv} |S_k|$, 2) $\sum_i p_{ij} s_{ik} = \lambda_{kj} |S_k|$, and $\sum_i s_{ik} = |S_k|$: then we can rewrite the Eqn.(50) and Eqn. (51) as:

$$F_Y(s_{ik}, c_{kv}) = T(Y) - \sum_k \sum_v c_{kv}^2 |S_k| \tag{58}$$

and

$$F_P(s_{ik}, \lambda_{kj}) = T(P) - \sum_k \sum_j \lambda_{kj}^2 |S_k| \tag{59}$$

Let us scrutinize these two early mentioned equations. For the sake of ease of notation let $B_Y = \sum_{k=1}^{K} \sum_v c_{kv}^2 |S_k|$, and similarly let $B_P = \sum_{k=1}^{K} \sum_j \lambda_{kj}^2 |S_k|$. Moreover, we will denote $F_Y(s_{ik}, c_{kv})$ and $F_P(s_{ik}, \lambda_{kj})$ with F_Y and F_P, respectively. Therefore, the criteria (58) and (59), in respect, can be rewritten as:

$$T(Y) = F_Y + B_Y,$$

and

$$T(P) = F_P + B_P.$$

Where, in the context of the data recovery approach, B_Y and B_P express the explain parts of data in the feature space and the network space, respectively. Furthermore, F_Y and F_P, in respect, express the unexplained parts of data in the feature data space and the network data space. Since the data scatters are given and constant, therefore, either we can maximize the explained parts of the data (B_Y and B_P) or equivalently, we can minimize the unexplained parts (F_Y, F_P).

In this work, we minimize the unexplained parts of the data. There are several metrics to measure the similarity between the features/links values with the corresponding centroids. Among the available options, we will exploit a) Euclidean distance and b) Cosine distance. We consider studying the impact of applying the other metrics like Manhattan distance, Minkowski distance as our future work.

To relate the constituents of our proposed clustering criterion to these two distance metrics: we need to recall that since the communities S_k ($k = 1, ..., K$) do not overlap, i.e. the membership vectors are orthogonal, thus the sum in the Eqn. (47) and Eqn. (48) plays a rather nominal role. Concretely, for any $i \in I$, y_{iv} is equal to $c_{kv} + f_{iv}$ just for that k at which $i \in S_k$. And similarly, for any $i \in I$, p_{ij} is equal to $\lambda_{kj} + e_{ij}$ just for that k at which $i \in S_k$. Thus the sum over k in the Eqn. (50) and (51) can be applied out of the parenthesis.

Therefore by definition of Eqn. (50) and regarding the obtain results in Eqn. (58) we have:

$$T(Y) - \sum_k \sum_v c_{kv}^2 |S_k| = \sum_{k=1}^{K} \sum_{i \in S_k} \sum_v (y_{iv} - c_{kv})^2 \tag{60a}$$

$$= \sum_{k=1}^{K} \sum_{i \in S_k} d_e(y_{i:}, c_k), \tag{60b}$$

And similarly by definition of Eqn. (51) and regarding Eqn. (59):

$$T(P) - \sum_k \sum_j \lambda_{kj}^2 |S_k| = \sum_{k=1}^{K} \sum_{i \in S_k} \sum_j (p_{ij} - \lambda_{kj})^2 \tag{61a}$$

$$= \sum_{k=1}^{K} \sum_i d_e(p_{i:}, \lambda_k). \tag{61b}$$

In both of the equations above, $d_e(.)$, by definition, represents the squared Euclidean distance metric between its two arguments. Thus recalling: a) our proposed criterion (Eqn.(49)) is additive; b) we intend to minimize the unexplained parts of data, thus by combining the Eqns. (60b) and (61b) together we have:

$$F_Y + F_P = \sum_k \sum_i d_e(c_k, y_{i:}) + d_e(\lambda_k, p_{i:}) \tag{62}$$

To relate these constituents (i.e Eqns. (58 and (59)) to Cosine distance let us proceed as follows. First we need to recall the definition of Cosine distance for two vectors $a = (a_i)$ and $b = (b_i)$

$$d_c = 1 - \frac{\sum_i a_i b_i}{\sqrt{\sum_i a_i^2} \sqrt{\sum_i b_i^2}}$$

Then by rewriting and expanding the RHS of Eqn. (60) we will have

$$\sum_{k=1}^{K} \sum_{i \in S_k} \sum_v (y_{iv} - c_{kv})^2 = \sum_{k=1}^{K} \sum_{i \in S_k} (\sum_v y_{iv}^2 + \sum_v c_{kv}^2) - 2 \sum_{k=1}^{K} \sum_{i \in S_k} \sum_v y_{iv} c_{kv} \tag{63a}$$

$$= 2 \sum_{k=1}^{K} \sum_{i \in S_k} (1 - \sum_v y_{iv} c_{kv}) \tag{63b}$$

$$= 2 \sum_{k=1}^{K} \sum_{i \in S_k} (d_c(y_{i:}, c_k)) \tag{63c}$$

$$\geq \sum_{k=1}^{K} \sum_{i \in S_k} (d_c(y_{i:}, c_k)), \tag{63d}$$

while RHS of Eqn. (63a) is obvious, for the Eqn. (63b) to hold we need to recall that for any given i and k: $\sum_v y_{iv}^2 = \sum_v c_{kv}^2 = 1$ if and only if they are unit-normalized (they are divided by their norm). And thus the denominator of Cosine distance will be equal to unity and thus Eqn. (63c) holds. As for inequality (63d), when the vectors are unit-normalized, it trivially holds; also, it holds when both of the (real-valued) vectors are non-unit-normalized, and their length is greater than or equal to unity. [3]

[3] To proof the non-normalized case, one needs to recall that for any given i, k: since $||y_{iv}|| \geq ||c_{kv}|| \geq 1$ then dividing RHS of Eqn. (63a) by $||y_{iv} \times c_{kv}||$ implies that $d_e(y_{iv}, c_{kv}) \geq d_c(y_{iv}, c_{kv})$, because of $||y_{iv}|| + ||c_{kv}|| \leq ||y_{iv}|| \times ||c_{kv}||$. Obviously, due to associativity property, the case when $||c_{kv}|| \geq ||y_{iv}|| \geq 1$ also holds. And finally, since for any i, k since $d_e(.)$ is always greater than or equal to $d_c(.)$, thus the sum of $d_e(.)$'s (sum over all i's and sum over all k's) will be also greater than or equal to the corresponding cosine distance ($d_c(.)$) $\square$.

In a similar framework we can show:

$$\sum_{k=1}^{K}\sum_{i\in S_k}\sum_{j}(p_{ij}-\lambda_{kj})^2 = \sum_{k=1}^{K}\sum_{i\in S_k}(\sum_{v}p_{ij}^2+\sum_{v}\lambda_{kj}^2) - 2\sum_{k=1}^{K}\sum_{i\in S_k}\sum_{v}p_{ij}\lambda_{kj} \tag{64a}$$

$$= 2\sum_{k=1}^{K}\sum_{i\in S_k}(1-\sum_{v}p_{ij}\lambda_{kj}) \tag{64b}$$

$$= 2\sum_{k=1}^{K}\sum_{i\in S_k}(d_c(p_{i:},\lambda_k)) \tag{64c}$$

$$\geq \sum_{k=1}^{K}\sum_{i\in S_k}(d_c(p_{i:},\lambda_k)). \tag{64d}$$

Again, similar to the Euclidean distance case, we can combine the Eqns. (60b) and (61b) together, which yields:

$$F_Y + F_P = \sum_{k}\sum_{i} d_c(c_k, y_{i:}) + d_c(\lambda_k, p_{i:}) \tag{65}$$

So far, we have proposed a combined clustering criterion for community detection in feature-rich networks. Since our proposed clustering criterion is quadratic, we have applied the optimality conditions. We have proven that communities' best representatives are their averages in the feature space and the network space. We have proven, to optimize the proposed clustering criterion: we can either maximize the data-explained parts or minimize the data-unexplained parts. In this work, we have pursued the latter. To this end, we exploit two distance metrics, namely, the Euclidean distance metric and the Cosine distance metrics.

Lastly, we extend the alternating strategy algorithm, specifically the K-Means algorithm. Consequently, we name our proposed adopted algorithm as the K-Means Extended to Feature-Rich Networks (KEFRiN). We distinguish the Euclidean distance or the Cosine distance is being applied with KEFRiNe and KEFRiNc, respectively. Below we explain KEFRiN algorithm.

K-Means Extended to Feature-Rich Networks (KEFRiN)

0. Data standardization: standardize the features and the networks links;

1. Initialization:
   - Choose the number of clusters, K,
   - Initialize seed centroids: $c_1, \lambda_1, ..., c_K, \lambda_K$,
   - Assume initial cluster lists $S = \{S_k\}_{k=1}^{K}$ empty;

2. Clusters update: given $2\times K$ centroids: K centroids in the feature space, and K centroids in the network space: determine clusters $S^{'} = \{S_k^k\}_{k=1}^{K}$ with minimum distance rule: either with $d_e(.)$ or $d_c(.)$:
   - KEFRiNe: $d_e(y_{i:}, c_k) + d_e(p_{i:}, \lambda_k)$
   - KEFRiNc: $d_c(y_{i:}, c_k) + d_c(p_{i:}, \lambda_k)$

3. Stop-condition: Check whether $S^{'} = S$. If yes, stop the clustering procedure, $S = \{S_k\}_{k=1}^{K}, C = \{c_k\}_{k=1}^{K}, \Lambda = \{\lambda_k\}_{k=1}^{K}$. Otherwise, change S with $S^{'}$;

4. Centroids update: Given clusters $S = \{S_k\}_{k=1}^{K}$ calculate within cluster means in the feature space and in the network space and go to Step 2.

In this work, we modified the K-Means++ [6] seed initialization method as follows.

1. Randomly select an index r and specify $c_1 = y_{r:}, \lambda_1 = p_{r:}$;
2. Considering both data sources, per each node, compute the sum of its distances from the nearest, previously chosen centroids both in the feature space and the network space by using either the Euclidean distance or Cosine distance (depending on the metric is used in main algorithm);
3. Choose the next centroid from the nodes set such that the node having the maximum sum of distances from the nearest centroids (in the feature space and network space) will be selected next as the centroids;
4. Repeat steps 2 and 3 until K centroids are determined.

The source code of KEFRiN methods, as well as all other supplementary materials, including the real-world data sets, synthetic data generator Etc. are publicly available in `https://github.com/Sorooshi/KEFRiN`.

Experimental setting

The computational experiment consists of the following constituents:

1. The set of algorithms under consideration.
2. The set of data sets at which the algorithms are evaluated and/or compared.
3. The set of criteria for evaluation of the experimental results.
4. The set of pre-processing techniques which are applied for standardizing or for normalizing the data sets.

In the thesis, we describe them each in separate sections. And here, in this summary, we only mention the principles and cite the references.

Algorithms under comparison

We compared the performance of our proposed methods with three algorithms of the model-based approach, CESNA [42], SIAN [30], DMoN [39], and one heuristic algorithm EVA [12]. We have extensively tested them in computational experiments. Besides, the author-made codes of the algorithms are publicly available. We also tested the algorithm PAICAN from [7] in our experiments. Unfortunately, this algorithm's results were always less than satisfactory; therefore, we excluded the algorithm PAICAN from this research.

Data sets

We use both real-world data sets and synthetic data sets. In the thesis, we describe them in two separate subsections. Here we briefly provided the general overview of real-world data sets.

Real world data sets

Unlike the proposed methods of this work, two of the algorithms under comparison restrict the features to be categorical. Therefore, whenever a data set contains a quantitative feature, we convert it into a categorical version. A brief overview of the eight real-world data sets under consideration can be found in Table 1.

Table 1: Real world data sets 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	Number of Communities	Ground Truth	Ref.
Malaria HVR6	307	6526	6	2	Cys Labels	[23]
Lawyers	71	339	18	6	Derived out of office and status features	[24, 35]
World Trade	80	1000	16	5	Structural world system in 1980 features	[31]
Parliament	451	11646	108	7	Political parties	[7]
COSN	46	552	16	2	Region	[14]
Cora	2708	5276	1433	7	Computer Science research area	[32]
SinaNet	3490	30282	10	10	Users of same forum	[21]
Amazon Photo	7650	71831	745	8	Product categories	[34]

Generating synthetic data sets

In thesis, we describe how we generate 800 synthetic data sets with an innate cluster structure by separately generating:

- networks;
- categorical features;
- quantitative features.

Each of these is put in a separate subsection.

Data pre-processing techniques

The results of SEFNAC, KEFRiN, and ICESi methods depend on how the data are standardized. Unfortunately, to the best of our knowledge, there are no theoretical foundations for data standardization issues. In the thesis, we describe two popular standardization methods for feature data, namely, Z-scoring and the Range-standardization methods. As for the network data, we use Modularity and Uniform shift.

Evaluation criteria

To evaluate and to compare the obtained clustering results, we consider two popular metrics of similarity between partitions: 1) The Adjusted Rand Index (ARI) [18], and 2) the Normalised Mutual Information (NMI) [36, 38].

Experiments

As mentioned earlier, we evaluate and compare the performance of our proposed methods over eight real-world and 800 synthetic data sets. Here we briefly provide the results at real-world data sets. Refer to thesis for more.

Experimental comparison of the methods under consideration

Comparison of the methods over real-world data sets

In this subsection, we compare the performance of all algorithms under consideration at the eight real-world data sets mentioned earlier. We run all the algorithms starting from random configurations ten times at each of the data sets. And we report the average and standard deviation of ARI values. Furthermore, we choose those pre-processing methods that led, on average, to the larger ARI values. Refer to thesis for more information.

The comparison of all algorithms under consideration over real-world data sets are recorded in table 2.

Table 2: Comparison of CESNA, SIAN, DMoN, SEFNAC, ICESi and KEFRiN methods at Real-world data sets; average values of ARI are presented over 10 random initialization. The best results are highlighted in bold-face and second ones are under-lined.

data set	CESNA	SIAN	DMoN	EVA	SEFNACs	SEFNACn	ICESiss	ICESins	ICESinn	ICESisn	KEFRiNe	KEFRiNc
HRV6	0.20(0.00)	0.39(0.29)	0.64(0.00)	0.036(0.004)	0.49(0.11)	0.42(0.04)	0.62(0.00)	0.62(0.00)	0.59(0.01)	0.62(0.00)	0.34(0.02)	**0.69(0.38)**
Lawyers	0.28(0.00)	0.59(0.04)	**0.60(0.04)**	0.159(0.028)	**0.60(0.09)**	0.59(0.09)	0.42(0.07)	0.51(0.01)	0.51(0.01)	0.35(0.11)	0.43(0.13)	0.44(0.14)
World Trade	0.13(0.00)	0.10(0.01))	0.13(0.02)	-0.003(0.000)	0.29(0.10)	0.41(0.10)	**0.47(0.13)**	0.37(0.02)	0.36(0.02)	**0.47(0.15)**	0.27(0.17)	0.40(0.11)
Parliament	0.25(0.00)	**0.79(0.12)**	0.48(0.02)	0.005(0.001)	0.28(0.01)	0.47(0.09)	0.00(0.00)	0.00(0.00)	0.34(0.03)	0.00(0.00)	0.15(0.09)	0.41(0.05)
COSN	0.44(0.00)	0.75(0.00)	0.91(0.00)	-0.004(0.000)	0.72(0.02)	0.54(0.04)	0.63(0.13)	0.83(0.00)	0.50(0.01)	0.76(0.03)	0.65(0.18)	**1.00(0.00)**
Cora	0.14(0.00)	0.17(0.03)	**0.37(0.04)**	0.002(0.001)	0.00(0.00)	0.00(0.00)	0.00(0.00)	0.00(0.00)	0.00(0.00)	0.00(0.00)	0.00(0.00)	0.21(0.01)
SinaNet (C)	0.09(0.00)	0.17(0.02)	0.28(0.01)	0.001(0.002)	0.21(0.03)	0.25(0.02)	0.21,(0.01)	0.22(0.02)	0.18(0.02)	0.20(0.01)	0.31(0.02)	**0.34(0.02)**
Amazon Photo	0.19(0.000)	N.A	**0.44(0.04)**	-0.001(0.001)	N.A	N.A	N.A	N.A	N.A	N.A	0.056(0.009)	0.43(0.055)

SIAN wins the competition for the Parliament data set, and it takes second place for the Lawyers data set. SEFNACs wins the competition for the Lawyers data set. SEFNACn takes second place in the competition for the Lawyers and World trade data sets. However, due to the computational complexity of SIAN, SEFNAC methods and ICESi methods, they are not applicable to the Amazon photo data set.

Unfortunately, EVA performs poorly. The obtained ARI values show that in most of the cases, it detects only one cluster. Such behavior could occur for two reasons. (a) The assumption authors made on the community's features to be identical for all members. (b) Due to non-sparse networks, the Modularity criterion loses its efficiency (because of assigning more weight to less connected nodes).

DMoN wins the competition for the Lawyers, the Cora, and the Amazon photo data sets. Moreover, it takes second place in the competitions for HRV6, Parliament, and COSN. It is noteworthy to add that the authors of DMoN have reported different results in the Cora and Amazon photo data sets: they used a subset of the original data set: the author of the current work managed only to reproduce the Cora data set's results. However, we use the entire Cora and Amazon data set in this work.

ICESiss and ICESisn both win the competition for the World trade data set. KEFRiNc wins the competition on the HRV6, COSN, and SinaNet data sets. It also wins second place in the Cora and Amazon Photo data sets' competitions (with close results to the competition winner, DMoN). KEFRiNe performs moderately acceptable on the Lawyers and the CONS data sets while performing poorly on the remaining cases.

If we compare ICESi methods with SEFNAC methods: we can easily observe that on HVR, World Trade, COSN, by converting the feature data into similarity data, we could significantly improve the cluster recovery results. However, this conversion does not lead us to better cluster recovery results over Lawyers and SinaNet data sets. Moreover, all ICESi methods except for ICESinn fail on Parliament data set.

Clearly, except for KEFRiNc, all proposed methods of this research, CESNA, and SIAN fail on recovering clusters on Cora, SinaNet, and Amazon photo data sets. Nonetheless, KEFRiNc's close results to DMoN, especially over the Amazon photo data set and the other obtained results, show that replacing Cosine distance with Euclidean distance can significantly improve the clustering recovery results.

Conclusions & future work

In this section, we first conclude the thesis, and then we describe several future works.

Conclusion

Using the conventional data recovery approach, we propose two similar methods, SEFNACs and SEFNACn, for community extraction at feature-rich networks. The methods differ in the assumptions of the network data entries' summability across the link table, yes or no, respectively. In this way, we distinguish between cases where the network data scales are the same for all the network nodes and cases at which each node collects its linkage data independently. The methods are similar in that both a) find clusters one-by-one, b) add the entries of a cluster also one-by-one.

ICESi methods continue the line of research started by SEFNAC methods. We explore whether the doubly-greedy least-squares approach proposed in that subsection can be successfully applied to feature-rich networks at which the feature-related part is converted to a similarity matrix format. Usually, similarity data are considered as measured on the same scale; so that one can meaningfully compare and sum similarity values across the entire similarity matrix (summability mode). However, there can be situations in which similarity values in one column (or row) should not be compared with the values in another column (or row) – nonsummable mode. By applying these two assumptions to the two similarity matrices, that feature-generated and that native, with link scores, we come to four different summability patterns denoted in the thesis by ss, ns, sn, and nn, and, accordingly to four different Iterative Community Extraction from Similarity data (ICESi) algorithms.

One of the theoretical advantages of SEFNAC and ICESi is a Pythagorean decomposition of the data scatter in the sum of the least-squares criterion and individual cluster contributions. This property allows scoring the contribution of various elements of found solutions to the data scatter, which can help interpretation [28]. Among practical advantages is the competitiveness of the doubly-greedy approach regarding its capacity for the cluster recovery against other computational procedures (see, for example, experimental results in [5, 10, 29, 33]).

The SEFNAC and ICESi methods have some properties which distinguish them from many others.

Desirable properties: a) no restriction on the feature scale type; b) no restriction on the network data type; c) determining the number of clusters/communities automatically; d) a Pythagorean decomposition of the combined data scatter in the sum of individual clusters' contributions and the minimized criterion.

Less desirable properties: e) the data standardization is a necessary part of the methods, both for network data and feature/similarity data; f) slow computations; g) no advice regarding the constants balancing the relative contributions of two data sources, the network, and features.

Nevertheless, our experiments show that our SEFNAC methods are competitive against state-of-the-art algorithms on small-size and medium-size data. The SEFNAC methods are relatively robust against noise and unfavorable structure parameters such as the probability of inter-community links, which can be as high as 0.6, meaning that the proportion of inter-community edges may be comparable or even greater than the probability of within-community edges.

On almost all settings for SEFNAC methods, the best data standardization options in our experiments involve z-scoring of the feature data and uniform shift transformation of the network data. The reason why Z-scoring improves our results can be justified as follows. Recall that the Z-scoring leads to different feature ranges, in contrast to the Range standardization. And since our method starts from the most anomalous clusters; thus, it is applying Z-scoring increases the sensitivity of SEFNAC. The uniform shift subtracts a constant threshold from the link values. In contrast, the popular modularity transformation subtracts random noise, which may differ depending on the number of links at different nodes. Our result supports the view [28] that at flat network data, the subtracted value should be flat/constant, too.

It appears that ICESi methods can be competitive too. Taking a closer look at a restricted version of our real-world dataset collection, by ignoring DMoN and KEFRiN methods, they win in most cases over remaining state-of-the-art methods, including in the non-summability mode. They show rather good performance at the networks with categorical features by closely following the winner, SEFNACs, and even outperforming that at some data configurations (in ss and sn modes). At synthetic networks with quantitative features, among all ICESi methods, ICESisn obtains the best results, with occasional interventions of ICESiss.

The properties of the methods mentioned above determine our last two methods' main directions, i.e., KEFRiN methods. First of all, we had to raise the computational power of the methods to be applicable to larger data sets. Thus, we adopt the k-means method to define the cluster center and distances from that to cluster elements according to our clustering criteria by considering both data sources. Furthermore, to alleviate the so-called curse of dimensionality, we used Cosine distance instead of the conventional Euclidean distance. And, this leads to two versions of KEFRiN methods, namely, KEFRiNe and KEFRiNc. Such a development has lead us to two algorithms, capable of handling dozens of thousand nodes rather than thousands, the SEFNAC and ICESi methods' capability.

KEFRiNc performs well at our comparison over real-world data sets. Although it wins just a few different data sets settings at the synthetic data sets, its overall performance is still quite acceptable. More importantly, it can be considered a decent solution for community extraction in feature-rich networks regarding its fast execution time. However, KEFRiNe is not as successful as its counterpart, KEFRiNc. Although it is as fast as KEFRiNc, due to the curse of dimensionality, it loses its efficiency on most of the data sets, especially at big data sets.

We can recommend the following benchmark for applying our proposed methods of this research as follows. A) When the number of clusters is unknown, and the network under consideration has similar characteristics to our synthetic data sets, SEFNACs is the preferable solution. B) In the same setting, if the user seeks faster execution time, SEFNACn would be recommended; C) Applying ICESiss and ICESins could be an extra appropriate trial for networks with the characteristics mentioned above. D) When the user knows the number of

clusters disregarding networks' characteristics under consideration, KEFRiNc is the fasted and most robust solution that this research can recommend.

Future works

We can see several directions for future works. Reformulating our proposed methods in a theory-driven framework is the most promising direction of our future works.

Another direction for future developments is to scrutinize the impact of applying different distance metrics, say, Minkowski distance, Manhattan distance, Mahalanobis distance Etc. on the performance of our methods.

The acceleration of the proposed sequential methods' execution time can be another exciting direction for future developments.

One other direction for future research is studying and analyzing the size proportions between the network data and the feature data. Changing the currently equal values of the balancing constants may become needed.

Applying the proposed simultaneous clustering strategy at feature-rich networks using similarity data should be considered another future work.

Adopting the online-Kmeans or adopting the Kernel-Kmeans for the proposed simultaneous clustering methods is another possible direction for future works. Moreover, applying the early mentioned methods at feature-rich networks using similarity data should be considered another future work.

Finally, investigating the impact of applying more clustering strategies like hierarchical, spectral strategies Etc. on our proposed models' performance can be considered a comprehensive and burdensome future research direction [4].

[4]During this Ph.D. study, we also investigated the impact of the SEFNACs clustering criterion using a hierarchical clustering strategy, namely, the Louvain algorithm, and a Spectral decomposition algorithm, i.e., by decomposing the two matrices using Eigen-decomposition. Although the preliminary results were not satisfactory, we still prefer to postpone concluding after conducting more systematic studies.

Bibliography

[1] L.M. Aiello, C. Cherifi, H. Cherifi, R. Lambiotte, P. Lio, and L.M. Rocha. Annual. *Complex Networks and their Applications*. Complex Networks. https://complexnetworks.org

[2] E. Akbas and P. Zhao. 2017. Attributed graph clustering: An attribute-aware graph embedding approach.. In *Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017*. ACM, Sydney, Australia, 305–308.

[3] L. Akoglu, H. Tong, B. Meeder, and C. Faloutsos. 2012. Parameter-free identification of cohesive subgroups in large attributed graphs. In *Proceedings of the 12th SIAM International Conference on Data Mining (PICS)*. SIAM, Pacific-Asia, 439—-450.

[4] Reda Alhajj. regular. *Social Network Analysis and Mining (SNAM)*. Springer. https://www.springer.com/journal/13278

[5] R.C. Amorim and B. Mirkin. 2012. feature weighting and anomalous cluster initializing in K-Means clustering Minkowski metric. *Pattern Recognition* 45, 3 (2012), 1061–1075.

[6] D. Arthur and S. Vassilvitskii. 2006. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*. ACM, Philadelphia, PAUnited States, 1027–1035.

[7] A. Bojchevski and S. Günnemanz. 2018. Bayesian robust attributed graph clustering: Joint learning of Partial anomalies and group structure. In *Thirty-Second AAAI Conference on Artificial Intelligence*. AAAI Press, California, USA, 1–10.

[8] J. Cao, H. Wanga, D. Jin, and J. Dang. 2019. Combination of links and node contents for community discovery using a graph regularization approach. *Future Generation Computer Systems* 91, 1 (2019), 361–370.

[9] S. Cao, W. Lu, and Q. Xu. 2015. Grarep: Learning graph representations with global structural information. In *Proceedings of the 24th ACM international on conference on information and knowledge management*. ACM, Melbourne, Australia, 891–900.

[10] M.M.T. Chiang and B. Mirkin. 2010. Intelligent choice of the number of clusters in k-means clustering: an experimental study with different cluster spreads. *Journal of Classification* 27, 1 (2010), 3–40.

[11] P. Chunaev. 2020. Community detection in node-attributed social networks: a survey. *Computer Science Review* 100286, 37 (2020).

[12] S. Citraro and G. Rossetti. 2020. Identifying and exploiting homogeneous communities in labeled networks. *Applied Network Science* 5, 1 (2020), 1–20.

[13] D. Combe, C. Largeron, M. Géry, and E. Egyed-Zsigmond. 2015. I-louvain: An attributed graph clustering method. In *International Symposium on Intelligent Data Analysis*. Springer, Konstanz, Germany, 181–192.

[14] R.L. Cross and A. Parker. 2004. *The hidden power of social networks: Understanding how work really gets done in organizations* (1st ed.). Harvard Business Press, USA.

[15] T.A. Dang and E. Viennet. 2012. Community detection based on structural and attribute similarities. In *International conference on digital society (icds)*. Not published, Valencia, Spain, 7–12.

[16] Reda Alhajj et.al. Annual. *Advances in Social Networks Analysis and Mining*. IEEE/ACM. http://asonam.cpsc.ucalgary.ca/2021/

[17] B.L. Golden and D.R. Shier. Annual. *Networks*. Wiley Periodicals. https://onlinelibrary.wiley.com/journal/10970037

[18] L. Hubert and P. Arabie. 1985. Comparing partitions,. *Journal of Classification* 2, 1 (1985), 193–218.

[19] INSNA. Annual. *International Networks for Social Networks Analysis*. INSNA. https://www.insna.org/#

[20] R. Interdonato, M. Atzmueller, S. Gaito, R. Kanawati, C. Largeron, and A. Sala. 2019. Feature-rich networks: going beyond complex network topologies. *Applied Network Science* 4, 1 (2019), 20–33. https://doi.org/10.1007/s41109-019-0111-x

[21] C. Jia, Y. Li, M.B. Carson, X. Wang, and J. Yu. 2017. Node attribute-enhanced community detection in complex networks. *Scientific Reports* 7, 1 (2017), 1–15.

[22] D. Jin, J. He, B. Chai, and D. He. 2021. Semi-supervised community detection on attributed networks using non-negative matrix tri-factorization with node popularity. *Frontiers of Computer Science* 15, 4 (2021), 1–11.

[23] D.B. Larremore, A. Clauset, and C.O. Buckee A. 2013. network approach to analyzing highly recombinant malaria parasite genes. *PLoS Computational Biology* 9, 10 (2013), p.e1003268.

[24] E. Lazega. 2001. *The Collegial Phenomenon: The Social Mechanisms of Cooperation Among Peers in a Corporate Law Partnership* (1st ed.). Oxford University Press, GB.

[25] X. Luo, Z. Liu, M. Shang, and M. Zhou. 2020. Highly-Accurate Community Detection via Pointwise Mutual Information-Incorporated Symmetric Non-negative Matrix Factorization. *IEEE Transactions on Network Science and Engineering* -, - (2020), –.

[26] J. B. MacQueen. 1967. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Symposium on Math, Statistics, and Probability*. CA: University of California Press, Berkeley, USA, 281–297.

[27] B. Mirkin. 2008. The iterative extraction approach to clustering. In *A. Gorban (Ed.) "Principal Manifolds for Data Visualization and Dimension Reduction*. Springer, Berlin, Germany, 151–177.

[28] B. Mirkin. 2012. *Clustering: A Data Recovery Approach* (2nd ed.). CRC Press, USA.

[29] S. Nascimento, S. Casca, and B. Mirkin. 2015. A seed expanding cluster algorithm for deriving upwelling areas on sea surface temperature images. *Computers & Geosciences* 85 (2015), 74–85.

[30] M.E. Newman and A. Clauset. 2016. Structure and inference in annotated networks. *Nature Communications* 7, 1 (2016), 1–11.

[31] W. De Nooy, A. Mrvar, and V. Batagelj. 2004. *Exploratory Social Network Analysis with Pajek,* (1st ed.). Cambridge University Press, GB.

[32] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Galligher, and T. Eliassi-Rad. 2008. Collective classification in network data. *AI magazine* 29, 3 (2008), 93–106.

[33] S. Shalileh and B. Mirkin. 2020. A Method for Community Detection in Networks with Mixed Scale Features at Its Nodes. In *International Conference on Complex Networks and Their Applications*. Springer, Madrid, Spain, 3–14.

[34] O. Shchur, M. Mumme, A. Bojchevski, and S. Günnemann. 2018. Pitfalls of graph neural network evaluation. *arXiv* preprint (2018). arXiv:1811.05868

[35] T. Snijders. 2001. *Lawyers Data Set.* Siena. `https://www.stats.ox.ac.uk/snijders/siena/`

[36] A. Strehl and J. Ghosh. 2002. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of machine learning research* -, - (2002), 583–617.

[37] F. Tian, B. Gao, Q. Cui, E. Chen, and T.Y. Liu. 2014. Learning deep representations for graph clustering.. In *Proceedings of the AAAI Conference on Artificial Intelligence*. AAAI Press, Québec, Canada, 100–106.

[38] J.A. T.M. Cover and Thomas. 2012. *Elements of Information Theory* (1st ed.). John Wiley and Sons, USA.

[39] A. Tsitsulin, J. Palowitch, B. Perozzi, and E. Müller. 2020. Graph clustering with graph neural networks. *arXiv* preprint (2020). arXiv:2006.16904

[40] X. Wang, D. Jin, X. Cao, L. Yang, and W. Zhang. 2016. Semantic community identification in large attribute networks. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, AAAI'16,*. ACM, Arizona, USA, 265—-271.

[41] Z. Xu, Y. Ke, Y. Wang, H. Cheng, and J. Cheng. 2012. A model-based approach to attributed graph clustering. In *Proceedings of the 2012 ACM SIGMOD international conference on management of data (ACM)*. ACM, Arizona, USA, 505–516.

[42] J. Yang, J. McAuley, and J. Leskovec. 2013. Community detection in networks with node attributes. In *IEEE 13th International Conference on Data Mining*. IEEE Computer Society, Washington DC, USA, 1151–1156.