Clustering Methods in Research: New Prospects

Dr. Soroosh Shalileh

Head of Vision Modeling Laboratory, HSE University, Moscow, RF,
Research Fellow at Center for Language and Brain, HSE University, Moscow, RF,
December 2023.

Motivation

According to [1]:

- **The “effectiveness” of clustering methods is one the nine open issues in clustering;**
- **Adapting the clusterings to various disciplines can be considered as a trends in clustering.**

- **Therefore, in this talks, the objectives are to:**
  - **improve the effectiveness of partitional clustering methods [2];**
  - **adopt clustering methods to community detection in attributed networks (feature-rich) in [3, 4].**

[1] Ezugwu, A.E., Ikotun, A.M., Oyelade, O.O., Abualigah, L., Agushaka, J.O., Eke, C.I. and Akinyelu, A.A., 2022. A comprehensive survey of clustering algorithms: State-of-the-art machine learning applications, taxonomy, challenges, and future research prospects. *Engineering Applications of Artificial Intelligence*, *110*, p.104743.

[2] Shalileh, S., 2023. An Effective Partitional Crisp Clustering Method Using Gradient Descent Approach. *Mathematics*, *11*(12), p.2617.

[3] Shalileh, S. and Mirkin, B., 2022. Community partitioning over feature-rich networks using an extended k-means method. *Entropy*, *24*(5), p.626.

[4] Mirkin, B. and Shalileh, S., 2022. Community detection in feature-rich networks using data recovery approach. *Journal of Classification*, *39*(3), pp.432-462.[4]: An Extension of K-Means for Least-Squares Community Detection in Feature-Rich Network.

An Effective Partitional Crisp Clustering Method Using Gradient Descent Approach

Dr. Soroosh Shalileh

Head of Vision Modeling Laboratory, HSE University, Moscow, RF
Research Fellow at Center for Language and Brain, HSE University, Moscow, RF
December 2023.

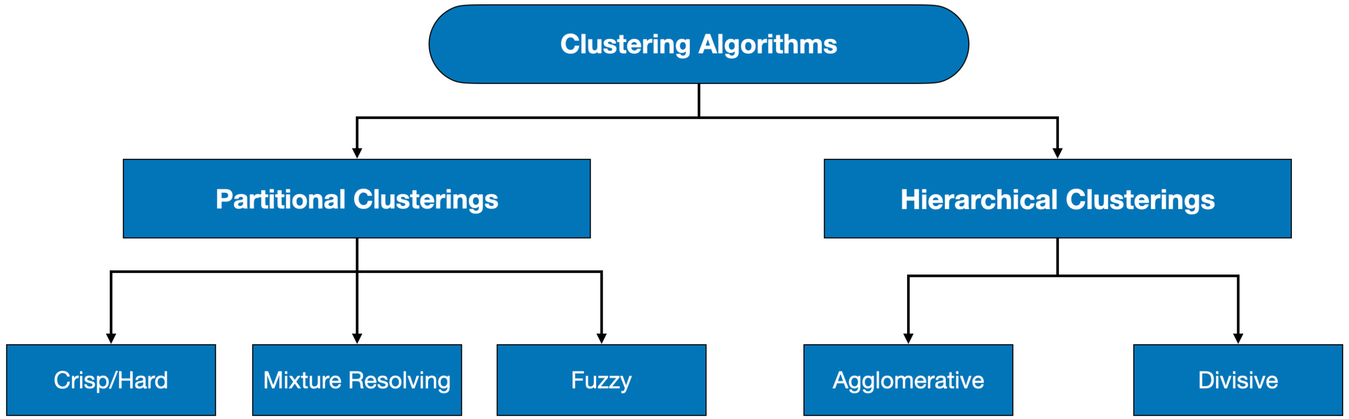
Contents

- Introduction and background
- Motivation
- Proposed methods (NGDC)
- Experiments settings
- Experiments at real-world data sets
- Experiments at synthetic data sets
- Conclusion and future work

Introduction and background

- **Clustering: partitioning the data set into partitions s.t. within-partition data points are as homogeneous as possible & between-partitions data points are as heterogeneous as possible.**

- **A recent review [1], extends the well-accepted taxonomy of clustering methods and reviews the trends and open challenges.**



[1] Ezugwu, A.E., Ikotun, A.M., Oyelade, O.O., Abualigah, L., Agushaka, J.O., Eke, C.I. and Akinyelu, A.A., 2022. A comprehensive survey of clustering algorithms: State-of-the-art machine learning applications, taxonomy, challenges, and future research prospects. *Engineering Applications of Artificial Intelligence*, *110*, p.104743.

Motivation

- **According to [1]:**
  - **Adapting the clusterings to various disciplines considered as a trends;**
  - **The “effectiveness” of clustering methods is one the nine open issues in clustering.**
- **[2] aims to improve the effectiveness of partitional clustering methods**

[2] Shalileh, S., 2023. An Effective Partitional Crisp Clustering Method Using Gradient Descent Approach. *Mathematics*, *11*(12), p.2617.

Proposed methods

Assumptions & notation

- **Consider a set of N data points $X = \{\mathbf{x}_i\}_{i=1}^{N}$, for $\mathbf{x}_i \in \mathbb{R}^V$, V is the dimensionality of the data points.**
- **Goal: partition X into K crisp clusters s.t.**
  - **(i) the within-cluster data points are as homogeneous as possible and,**
  - **(ii) the between-clusters data points are as heterogeneous as possible.**
- **Associate each cluster, $\mathbf{s}_k$, with the centroid in the $\mathbf{c}_k$:**
  - $S = \{\mathbf{s}_k\}_{k=1}^{K}$**: set of clusters,**
  - $C = \{\mathbf{c}_k\}_{k=1}^{K}$**: set of centroids in feature space.**

Proposed methods

Clustering objective and the strategies

- **Generic clustering objective function:**

$$F(X, S, C) = \sum_{k=1}^{K} \sum_{i=1}^{N} f(\mathbf{x}_i, \mathbf{c}_k) \qquad (1)$$

- **where** $f : X \times C \longrightarrow \mathbb{R}$ **represents a (generic) distance function that will be applied to measure the distance between the data point** $\mathbf{x}_i$ **and the centroid** $\mathbf{c}_k$ **.**
- **There can be various strategies for optimizing this objective:**
- **In the current research, we adopt gradient descent (GD).**
- **Gradient is the direction of the steepest descent direction: named the core of our proposed model:**

“Gradient Descent Clustering (GDC).”

Proposed methods
GDC: Notation update

- **Update the notation, by reflecting the iterations, at t-th iteration:**
  - **set of clusters** $S^{(t)} = \{\mathbf{s}_k^{(t)}\}_{k=1}^{K}$,
  - **set of centroids** $C^{(t)} = \{\mathbf{c}_k^{(t)}\}_{k=1}^{K}$.
- **GDC has three components:**
  1. **cluster assignment criterion,**
  2. **cluster update rule(s),**
  3. **convergence condition.**

Proposed methods
GDC: methodology-I

- **The cluster assignment criterion:**

$$\underset{k}{argmin} f(\mathbf{x}_i, \mathbf{c}_k^{(t)}) < f(\mathbf{x}_i, \mathbf{c}_j^{(t)}),\ \forall j \neq k\,. \qquad (2)$$

- **The update rule, in its vanilla form, VGDC:**

$$\mathbf{c}_k^{(t+1)} = \mathbf{c}_k^{(t)} - \alpha \nabla_{\mathbf{c}_k^{(t)}} f(\mathbf{x}_i, \mathbf{c}_k^{(t)}), \qquad (3)$$

  - **α represents the step size, and $\nabla_{\mathbf{c}_k^{(t)}}$ is the gradient of the distance function, f, w.r.t the k-th centroid at iteration t evaluated with the data point $\mathbf{x}_i$.**

- **VSDC is prone to slow convergence, especially, at nearly flat surfaces;**
- **Accumulating momentum was proposed to tackle it, however, it may lead to overshoots at the valley floor;**
- **To avoid the overshooting, Nesterov accelerated momentum can be adopted (NGDC).**

Proposed methods
SDC: methodology-II

- **NGDC update rule: to modify the gradient at the projected future position as follows:**

$$\mathbf{v}^{(t+1)} = \beta_1 \mathbf{v}^{(t)} - \alpha \nabla_{\mathbf{c}_k^{(t)}} f(\mathbf{x}_i + \beta_1 \mathbf{v}^{(t)}, \mathbf{c}_k^{(t)}), \qquad (4a)$$

$$\mathbf{c}_k^{(t+1)} = \mathbf{c}_k^{(t)} + \mathbf{v}^{(t+1)}. \qquad (4b)$$

- **where**
  - $\mathbf{v}^{(t)}$ **is the momentum vector, with initial values of zero, which accumulates the gradient's history up to iteration** t
  - $\beta_1 \in [0,1)$ **is a coefficient, a hyper-parameter, that decays the momentum: our empirical studies: value** $\in [0.3,0.6]$ **leads to superior results**

Proposed methods
SDC: methodology-III

- **Noteworthy to add**

1. **Since at each iteration, we compute the gradients of f w.r.t the closet centroid of $\mathbf{x}_i$, thus, adding $\mathbf{v}^{(t)}$ usually has a desirable impact: consider the ideal situation for which the data and the momentum vectors point to the same direction in the (feature) space. Thus, this addition decreases the gradients, which is desirable to avoid overshooting. Meanwhile, the first term in Eqn. (4a) provides additional momentum for going down the hill.**

2. **However, in less ideal circumstances, this addition may have less desirable influences, and this negative influence becomes more exaggerated when some of the components of the gradient vectors (or the momentum) have constantly high values: for which they negatively influence the update direction.**

3. **The adaptive gradient optimization methods have been proposed to dull the effect of such components. We postponed applying those methods to our future studies (the manuscript is under the review).**

Proposed methods

SDC: proposed algorithms

Algorithm 1: Nesterov momentum Gradient Descent Clustering (NGDC)

Input: X: Data set; K: number of clusters.

Hyperparameters: α: step size; T: maximum number of iterations; τ: the loss function upper bound; β_1: momentum decay coefficient.

Result: $S = \{s_k^{(t)}\}_{k=1}^{K}$ % set of K binary cluster membership vectors;

$C = \{\mathbf{c}_k^{(t)}\}_{k=1}^{K}$ % set of K centroids in feature space.

Initialize: Randomly initialize C and S.

for $t \in Range(T)$ **do**

  for $x_i \in X$ **do**

    find k using Equation (2) and set i-th entry of the $s_k^{(t)}$ to one;

    update clusters using the Equation (4);

    if *Equation* (1) $\leq \tau$ **then**

      Halt.

    end

  end

end

- **Python Source code:** https://github.com/Sorooshi/NGDC_method

Experiments settings
Competitors

- **Four algorithms from the literature**

1. **Agglomerative clustering:** Murtagh, F. and Contreras, P., 2012. Algorithms for hierarchical clustering: an overview. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *2*(1), pp.86-97.: **recursive hierarchical**

2. **K-means:** Steinley, D., 2006. K-means clustering: a half-century synthesis. *British Journal of Mathematical and Statistical Psychology*, *59*(1), pp.1-34.: **simultaneous K-clusters extraction using alternating optimization**

3. **GMM:** McLachlan, G.J. and Rathnayake, S., 2014. On the number of components in a Gaussian mixture model. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *4*(5), pp.341-355.: **A generalization of K-means using EM algorithm and Gaussian Distributions.**

4. **Spectral:** Von Luxburg, U., 2007. A tutorial on spectral clustering. *Statistics and computing*, *17*, pp.395-416.: **A set of combined approaches based on the eigenvalue analysis of a graph's adjacency and K-means.**

Experiments settings

Metric-I

- **Normalized Mutual Information (NMI):** Cover, T. & Thomas, J. Elements of Information Theory; John Wiley and Sons: Hoboken, NJ, USA, 2006.
- ***Given two partitions: Cluster memberships*** $S = \{S_k\}_{k=1}^{K}$ ***& Ground truth*** $T = \{T_l\}_{l=1}^{L}$
- ***Contingency table is a two-way table, s.t. its rows correspond to parts of*** S***, and its columns, to parts of*** T***.***
- ***The*** (k, l)***-th entry is*** $n_{kl} = |S_k \cap T_l|$***, the frequency of*** (k, l) ***co-occurrences.***
- ***Marginal row and marginal column are defined as*** $a_k = \sum_{l=1}^{L} n_{kl} = |S_k|$ ***and*** $b_l = \sum_{k=1}^{K} n_{kl} = |T_l|$
- ***The probability that an object picked at random falls into*** S_k ***is*** $a(k) = a_k/N$ ***or into*** T_l ***is*** $b(l) = b_l/N$***.***
- ***The entropy of*** S ***and*** T ***:=*** $H(S) = -\sum_{k=1}^{K} a(k)\log(a(k))$ ***&*** $H(T) = -\sum_{l=1}^{L} b(l)\log(b(l))$***, respectively.***

Experiments settings

Metric-II

- **Normalized Mutual Information (NMI):**

- ***Mutual information (MI) between S and T is calculated using:***

$$MI(S,T) = \sum_{k}^{K} \sum_{l=1}^{L} p_{kl} \log\left(\frac{p_{kl}}{a(k) \times b(l)}\right),$$

- ***where $p_{kl} = n_{kl}/N$ is the probability that an object picked at random falls into both S_k & T_l $(k = 1,2,...,K;\ l = 1,2,...,L)$. Therefore, normalized mutual information is defined:***

$$NMI = \frac{MI(S,T)}{max(H(S),H(T))}.$$

- ***NMI $\in [0,1]$, the closer its values are to unity, the better the match between the clustering results and the ground truth and vice versa.***

Experiments settings

12 popular real-world data sets

NATIONAL RESEARCH UNIVERSITY

Table 2. The real-world dataset's characteristics.

Dataset	Points	Features	Clusters
Breast Tissue	106	9	6
Ecoli	336	7	8
Fossil	87	6	3
Glass	214	9	6
Iris	150	4	3
Leaf	340	15	30
Libras Movement	360	90	15
Optical Recognition	3823	62	10
Spam Base	4601	57	2
Pen-Based Recognition	7494	16	10
Wine	178	13	3

Experiments settings

Data sets: synthetic data-I

1. Determine the number of data points N, clusters K, and features V.

2. The clusters' cardinalities were determined randomly with two constraints:

   i. no cluster should contain less than a pre-specified number of data points (we set this number to 30 in our experiments),

   ii. the number of data points in all clusters should sum to N.

3. We generated each cluster from a multivariate normal distribution:

   i. with diagonal covariance matrix where the values derived uniformly at random from the range $[0.05,0.1]$ (they specify the cluster's spread), and

   ii. means, i.e., each component of the cluster centroid are derived uniformly random from the range $\zeta \times [-1, +1]$, where $\zeta \in A$controls the cluster intermix: the smaller value of ζ, the higher the chance that data points from a cluster fall within the spreads of other clusters.

Experiments settings

Data sets: synthetic data-II

Table 1. Synthetic datasets configurations to study the hyperparameters of proposed methods (six configurations) and to validate and compare the methods under consideration (72 configurations): each case consists of 10 repeats summing up to 780 datasets.

Generator Parameters	Hyperparameter-Scrutinizing Values	Comparison Values
Clusters (K)	5, 15	2, 10, 20
Features (V)	10	2, 5, 10, 15, 20, 200
Data points (N)	2000	1000, 3000
Clusters intermix (ζ)	0.35, 0.65, 0.95	0.4, 0.8

Experiments

Real-world data sets

Table 9. Comparison on real-world datasets with $n\_init = 10$. The best results regarding NMI are bold-faced.

	K-Means	GMM	Spectral	Agglomerative	NGDC
Breast Tissue	0.515 ± 0.020	0.381 ± 0.047	0.462 ± 0.030	0.419 ± 0.000	**0.549 ± 0.017**
Ecoli	0.599 ± 0.010	0.388 ± 0.061	0.559 ± 0.007	**0.667 ± 0.000**	0.630 ± 0.024
Fossil	**1.000 ± 0.000**	0.687 ± 0.242	**1.000 ± 0.000**	**1.000 ± 0.000**	**1.000 ± 0.000**
Glass	0.338 ± 0.018	0.331 ± 0.035	0.277 ± 0.001	0.294 ± 0.000	**0.387 ± 0.031**
Iris	0.742 ± 0.000	0.629 ± 0.032	0.658 ± 0.000	**0.784 ± 0.000**	0.766 ± 0.020
Leaf	0.648 ± 0.010	0.589 ± 0.013	0.622 ± 0.017	0.621 ± 0.000	**0.653 ± 0.009**
Libras Movement	0.597 ± 0.015	0.197 ± 0.019	0.563 ± 0.016	0.563 ± 0.000	**0.602 ± 0.012**
Optical Recognition	0.761 ± 0.011	0.387 ± 0.068	0.699 ± 0.001	0.733 ± 0.000	**0.774 ± 0.019**
Spam Base	0.085 ± 0.115	0.080 ± 0.043	0.004 ± 0.000	0.001 ± 0.000	**0.259 ± 0.003**
Pen-Based Recognition	0.692 ± 0.006	0.691 ± 0.034	0.692 ± 0.000	0.635 ± 0.000	**0.708 ± 0.010**
Wine	0.846 ± 0.006	0.332 ± 0.107	**0.909 ± 0.000**	0.018 ± 0.000	0.858 ± 0.012

- **NGDC won eight out of 12 competitions.**
- **It has significant edge over the competitors at Breast Tissues, Glass, and Spam Base.**
- **Agglomerative is the second winner.**

Experiments

Synthetic data sets: N=1000, 3000 and K=2

Table 10. Comparison on synthetic datasets with 1000 data points and two clusters. The best results regarding NMI are bold-faced.

V	ζ	K-Means	GMM	Spectral	Agglomerative	NGDC
2	0.40	0.257 ± 0.212	0.041 ± 0.050	0.255 ± 0.215	0.008 ± 0.011	**0.284 ± 0.200**
	0.80	0.487 ± 0.286	0.075 ± 0.119	0.484 ± 0.282	0.180 ± 0.354	**0.499 ± 0.286**
5	0.40	**0.538 ± 0.156**	0.044 ± 0.059	0.537 ± 0.159	0.064 ± 0.184	0.523 ± 0.195
	0.80	0.876 ± 0.137	0.106 ± 0.087	**0.877 ± 0.138**	0.581 ± 0.474	0.837 ± 0.201
10	0.40	0.562 ± 0.151	0.415 ± 0.145	0.627 ± 0.138	0.006 ± 0.007	**0.685 ± 0.175**
	0.80	0.987 ± 0.033	0.976 ± 0.066	0.990 ± 0.024	0.895 ± 0.295	**0.993 ± 0.009**
15	0.40	0.869 ± 0.072	0.754 ± 0.294	**0.883 ± 0.063**	0.268 ± 0.399	0.882 ± 0.066
	0.80	**1.000 ± 0.000**	0.902 ± 0.293	**1.000 ± 0.000**	0.999 ± 0.003	**1.000 ± 0.000**
20	0.40	0.951 ± 0.039	0.310 ± 0.434	0.950 ± 0.037	0.281 ± 0.428	**0.958 ± 0.033**
	0.80	**1.000 ± 0.000**	**1.000 ± 0.000**	0.993 ± 0.008	0.997 ± 0.010	**1.000 ± 0.000**
200	0.40	**1.000 ± 0.000**	0.000 ± 0.000	0.992 ± 0.016	**1.000 ± 0.000**	0.600 ± 0.490
	0.80	**1.000 ± 0.000**	0.001 ± 0.001	**1.000 ± 0.000**	**1.000 ± 0.000**	**1.000 ± 0.000**

Table 11. Comparison on synthetic datasets with 3000 data points and two clusters. The best results regarding NMI are bold-faced.

V	ζ	K-Means	GMM	Spectral	Agglomerative	NGDC
2	0.40	0.306 ± 0.200	0.168 ± 0.112	0.306 ± 0.199	0.001 ± 0.001	**0.313 ± 0.201**
	0.80	0.584 ± 0.261	0.107 ± 0.104	0.574 ± 0.255	0.249 ± 0.381	**0.589 ± 0.260**
5	0.40	0.395 ± 0.230	0.033 ± 0.047	0.391 ± 0.228	0.002 ± 0.004	**0.403 ± 0.224**
	0.80	0.844 ± 0.140	0.339 ± 0.276	**0.858 ± 0.135**	0.463 ± 0.465	0.852 ± 0.141
10	0.40	0.750 ± 0.165	0.015 ± 0.014	0.748 ± 0.165	0.089 ± 0.264	**0.757 ± 0.165**
	0.80	0.969 ± 0.064	0.028 ± 0.030	0.970 ± 0.062	0.884 ± 0.294	**0.972 ± 0.061**
15	0.40	0.835 ± 0.111	0.018 ± 0.014	0.834 ± 0.109	0.001 ± 0.001	**0.843 ± 0.108**
	0.80	**1.000 ± 0.000**	0.118 ± 0.295	0.999 ± 0.002	0.998 ± 0.004	**1.000 ± 0.000**
20	0.40	0.918 ± 0.069	0.844 ± 0.283	**0.926 ± 0.061**	0.095 ± 0.280	0.922 ± 0.073
	0.80	0.999 ± 0.004	**1.000 ± 0.000**	0.994 ± 0.007	**1.000 ± 0.000**	0.999 ± 0.004
200	0.40	**1.000 ± 0.000**	0.001 ± 0.001	**1.000 ± 0.000**	**1.000 ± 0.000**	**1.000 ± 0.000**
	0.80	**1.000 ± 0.000**	0.000 ± 0.000	**1.000 ± 0.000**	**1.000 ± 0.000**	**1.000 ± 0.000**

- **NGDC dominates these tables**

Experiments

Synthetic data sets: N=1000, 3000 and K=10

Table 12. Comparison on synthetic datasets with 1000 data points and ten clusters. The best results regarding NMI are bold-faced.

V	ζ	K-Means	GMM	Spectral	Agglomerative	NGDC
2	0.40	0.207 ± 0.040	0.157 ± 0.045	0.193 ± 0.039	0.192 ± 0.039	**0.212 ± 0.040**
	0.80	0.432 ± 0.041	0.321 ± 0.077	0.401 ± 0.030	0.433 ± 0.038	**0.438 ± 0.043**
5	0.40	0.350 ± 0.049	0.200 ± 0.045	0.285 ± 0.026	0.199 ± 0.081	**0.358 ± 0.046**
	0.80	**0.753 ± 0.044**	0.607 ± 0.077	0.641 ± 0.061	0.711 ± 0.063	0.751 ± 0.042
10	0.40	**0.549 ± 0.077**	0.283 ± 0.072	0.460 ± 0.077	0.251 ± 0.153	0.546 ± 0.075
	0.80	**0.957 ± 0.029**	0.765 ± 0.083	0.922 ± 0.037	0.925 ± 0.027	0.956 ± 0.025
15	0.40	**0.757 ± 0.050**	0.338 ± 0.057	0.677 ± 0.075	0.363 ± 0.196	0.739 ± 0.045
	0.80	0.940 ± 0.030	0.666 ± 0.051	**0.991 ± 0.011**	0.985 ± 0.023	0.948 ± 0.031
20	0.40	**0.862 ± 0.034**	0.316 ± 0.016	0.804 ± 0.031	0.566 ± 0.125	0.843 ± 0.035
	0.80	0.993 ± 0.015	0.676 ± 0.051	**0.999 ± 0.002**	**0.999 ± 0.002**	0.970 ± 0.014
200	0.40	0.971 ± 0.027	0.023 ± 0.003	**1.000 ± 0.000**	**1.000 ± 0.000**	0.916 ± 0.020
	0.80	0.961 ± 0.027	0.023 ± 0.003	**1.000 ± 0.000**	**1.000 ± 0.000**	0.930 ± 0.027

Table 13. Comparison on synthetic datasets with 3000 data points and ten clusters. The best results regarding NMI are bold-faced.

V	ζ	K-Means	GMM	Spectral	Agglomerative	NGDC
2	0.40	0.200 ± 0.031	0.158 ± 0.041	0.193 ± 0.029	0.189 ± 0.035	**0.204 ± 0.033**
	0.80	0.410 ± 0.052	0.346 ± 0.081	0.381 ± 0.046	**0.415 ± 0.062**	0.413 ± 0.052
5	0.40	0.348 ± 0.054	0.199 ± 0.056	0.283 ± 0.041	0.169 ± 0.101	**0.355 ± 0.053**
	0.80	0.716 ± 0.057	0.685 ± 0.138	0.604 ± 0.055	**0.743 ± 0.062**	0.730 ± 0.066
10	0.40	0.579 ± 0.046	0.327 ± 0.132	0.484 ± 0.049	0.032 ± 0.045	**0.581 ± 0.042**
	0.80	0.920 ± 0.035	0.892 ± 0.041	0.877 ± 0.042	0.901 ± 0.045	**0.927 ± 0.021**
15	0.40	0.714 ± 0.051	0.552 ± 0.113	0.616 ± 0.050	0.178 ± 0.218	**0.727 ± 0.057**
	0.80	0.935 ± 0.031	0.909 ± 0.078	0.967 ± 0.039	**0.990 ± 0.008**	0.922 ± 0.034
20	0.40	**0.852 ± 0.031**	0.635 ± 0.067	0.784 ± 0.036	0.352 ± 0.263	0.845 ± 0.036
	0.80	0.956 ± 0.025	0.834 ± 0.086	**0.999 ± 0.001**	0.998 ± 0.001	0.967 ± 0.019
200	0.40	0.979 ± 0.014	0.011 ± 0.001	**1.000 ± 0.000**	**1.000 ± 0.000**	0.940 ± 0.018
	0.80	0.974 ± 0.022	0.015 ± 0.003	**1.000 ± 0.000**	**1.000 ± 0.000**	0.970 ± 0.008

- **NGDC and Agglomerative are the winners**
- **Refer to the paper for more: https://www.mdpi.com/2227-7390/11/12/2617**

Conclusion and future work

Conclusion:

1. **Applied GD to least-squares criterion for clustering in feature space**

2. **Empirically validated and compared the performance of NGDC with four benchmark algorithms on 12 real-world and 780 synthetic data.**

3. **NGDC appeared to be the wining algorithm of the current research**

Future work:

1. **NGDC is sensitive to seed initialization: proposing a technique to reduce this sensitivity or initialization of the seeds more effectively**

2. **NGDC uses fixed step size >> components with constantly high gradients may have less desirable impacts: using “adaptive moment estimation (ADAM)” update rule to tackle the issues**

3. **Conducting more experiments using different distance metrics, like Canberra distance.**

Thank you!

Soroosh Shalileh

sr.shalileh@gmail.com

Clustering Feature-rich Networks using Data Recovery Approach

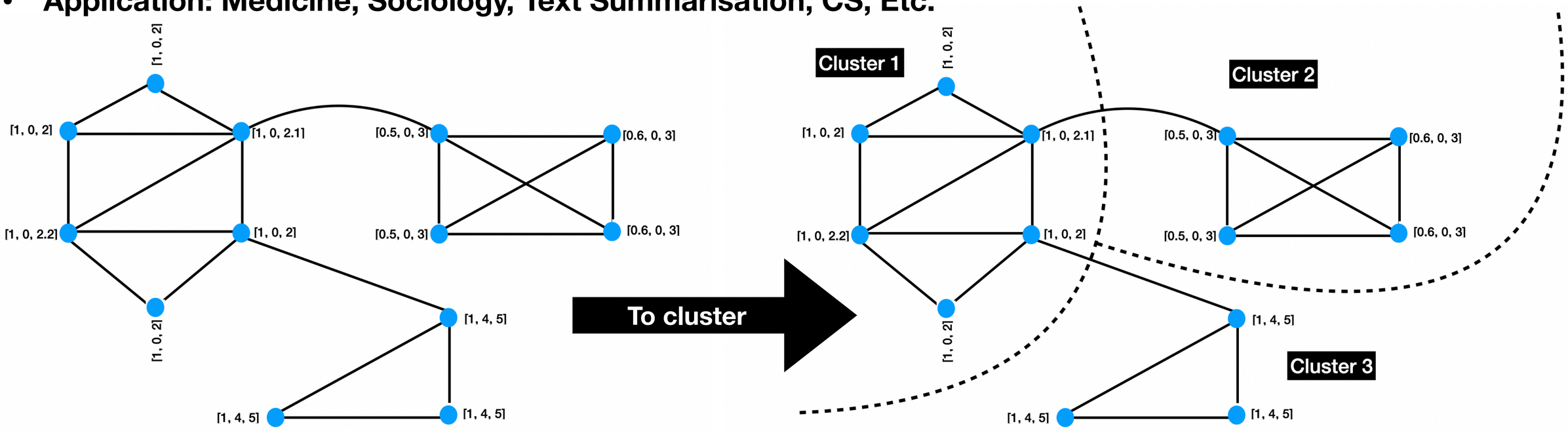
Dr. Soroosh Shalileh
Prof. Boris Mirkin.

Contents

- Introduction and terminology
- General models
- One-by-One clustering method (SEFNAC)
- Experiments settings
- Experiments results 1
- Simultaneous clustering method (KEFRiN)
- Experiments results 2
- Future work

Introduction and terminology

- **Feature-rich network: a graph with a set of features associated with its nodes.**
- **Community: a relatively dense group of entities with similar feature values.**
- **Goal: to extract clusters using networks links and nodes features simultaneously.**
- **Application: Medicine, Sociology, Text Summarisation, CS, Etc.**



Proposed methods

Least-squares model: notation

- **Consider a feature-rich network at the nodes, A={P, Y}, over entity set “I”; “I” is a set of network nodes of cardinality |I|=N.**
- $P = (p_{ij}) \in \mathbb{R}^{N \times N}$ **Matrix of mutual link weights between nodes“i” & “j” (**$i = 1,2,...,N$**);**
- $Y = (y_{iv}) \in \mathbb{R}^{N \times V}$ **Matrix of feature values** y_{iv} **for nodes** $i = 1,2,...,N$ **and over features** $v = 1,2,...,V$**.**
- **Feature model:** $y_{iv} = \sum_k c_{kv} s_{ik} + f_{iv}$ **(AN)**
- **Network (similarity) models:** **1)** $p_{ij} = \sum_k \lambda_k s_{ik} s_{jk} + e_{ij}$ **(AS);**

- $s_k = (s_{ik})$**: Binary N-dimensional cluster membership vectors (the clusters are crisp);**
- $c_k = (c_{kv})$**: V-dimensional cluster centroids;** λ_k**: Positive network intensity weight.**
- e_{ij} , f_{iv}**: Residuals (Errors) to be minimised;** $k = 1,\ldots,K$ **represent the number of clusters**

Proposed methods

Clustering strategies

- There can be various strategies for optimization of the (aforementioned) criterion.
- we adopt:
  - A) Sequentially cluster extraction strategy (one cluster at the time).
  - B) Simultaneous cluster extraction strategy (K clusters at the time).

- The former strategy: first, was proposed in “individual clusters Mirkin 1976 (in Russian), Mirkin JClas 1987, Further Amorim & Mirkin PatRec 2012;
- The latter strategy: K-Means is one the most popular instance.

Proposed methods

SEFNAC: methodology-I

- **Pursuing the least-squares principles: criteria: minimise:**

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

- **w.r.t unknown membership vectors s_k , cluster centroids c_k and intensity weight λ_k**
- **Where ρ , ξ are user-defined constants, $(\rho = \xi = 1)$**

Proposed methods

SEFNAC: methodology-II

- **One by one (sequential) strategy: Search one cluster S , c , λ | λ_j at a time: (just remove the index k)**

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

- **Applying first order optimality and little algebraic manipulation:**

- $$c_v = \frac{\sum_i y_{iv} s_i}{|S|}; \qquad \lambda = \frac{\sum_{i,j} p_{ij} s_i s_j}{\sum_i s_i^2 \sum_j s_j^2}.$$

Proposed methods

SEFNAC: methodology-III

- **Expanding the Eqns. and optimal c_v and $\lambda \,|\, \lambda_j$ and substituting them below implies:**

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

- $T(Y) = \sum_{i,v} y_{iv}^2$ **and** $T(P) = \sum_{i,j} p_{ij}^2$ **: are quadratic scatters of Y and P, respectively. Thus:**

A. Under AS: $F_{AS}(S) = \rho T(Y) + \xi T(P) - G_{AS}$ **; Where** $G_{AS} = \rho \,|\, S \,|\, \sum_{v} c_v^2 + \xi\lambda \sum_{i,j} p_{ij} s_i s_j$

- **Therefore, minimising the residuals is equivalent to maximising** $G(S) \;\; i \,.\, e \,.\, G_{AS}(S)$

Proposed methods (local search)
SEFNAC: methodology-IV

- **Maximising $G(S)$ $i.e.G_{AS}(S)$: By optimally adding nodes one by one: Feature-rich Network Addition Clustering (FNAC). Using FNAC iteratively and sequentially for partitioning: SEFNAC.**

- **FNAC: starts from a random seed "i" forming a singleton cluster $S = \{i\}$.**

- **At any current S, considers every element $j \in I - S$; select that "j" at which the increment of contribution $G(s)$ is maximal. If this maximum is positive, then "j" is added to "S," and the module runs again.**

- **If the maximum is negative or zero or no unclustered entity has remained, the FNAC halts and outputs "S."**

- **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.**

Proposed methods
SEFNAC: methodology-V

- **SEFNAC:**

1. ***Initialisation.*** **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,**

3. **Redefine J by removing all the elements of S_k from it (i.e. $J = J - S_k$). Check whether thus obtained J is empty or not:**

   If yes, stop. Define the current k as K and output S_k for $k = 1,...,K$. If not, add 1 to k, and go to step 2.

Mirkin, B. and Shalileh, S., 2022. Community Detection in Feature-Rich Networks Using Data Recovery Approach. Journal of Classification, pp.1-31.

Experiments settings
Competition

- **Four algorithms from the literature**

1. **CESNA:** 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. *(Author of SNAP Lib.):* **Probabilistic Generative Model**

2. **SIAN:** M.E. Newman and A. Clauset. 2016. Structure and inference in annotated networks. Nature Communications 7 (2016), 11863. *(Authors of Modularity):* **Bayesian statistical inference >> Generative Network Model**

3. **DMoN:** A. Tsitsulin, J. Palowitch, B. Perozzi, and E. Muller. 2020. Graph clustering with graph neural networks. arXiv preprint (2020). arXiv:2006.16904. *(Google data scientists):* **Graph Convolutional Neural Networks**

4. **EVA:** S. Citraro and G. Rossetti. 2020. Identifying and exploiting homogeneous communities in labeled networks. Applied Network Science 5, 1 (2020), 1–20. *(KDD Researchers):* **Heuristic: Modularity & Purity**

Experiments settings

Metric

- **Adjusted Rand Index (ARI): Proposed in:** ***L. Hubert and P. Arabie. 1985. Comparing partitions, Journal of Classification, 2, 1 (1985), 193–218.***

- ***Given two partitions: Cluster memberships*** $S = \{S_k\}_{k=1}^{K}$***; Ground truth*** $T = \{T_l\}_{l=1}^{L}$

- ***Contingency table is a two-way table, such that its rows correspond to parts of*** S***, and its columns, to parts of*** T***.***

- ***The*** (k, l)***-th entry is*** $n_{kl} = |S_k \cap T_l|$***, the frequency of*** (k, l) ***co-occurrences.***

- ***Marginal row and marginal column are defined as*** $a_k = \sum_{l=1}^{L} n_{kl} = |S_k|$ ***and*** $b_l = \sum_{k=1}^{K} n_{kl} = |T_l|$

$$ARI(S,T) = \frac{\sum_{k,l}\binom{n_{kl}}{2} - [\sum_k \binom{a_k}{2}\sum_l \binom{b_l}{2}]/\binom{N}{2}}{\frac{1}{2}[\sum_k \binom{a_k}{2} + \sum_l \binom{b_l}{2}] - [\sum_k \binom{a_k}{2}\sum_l \binom{b_l}{2}]/\binom{N}{2}]}$$

- ***The closer to unity the better.***

Experiments settings

Data sets: real-world

TABLE 4.1: 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	Number of Communities	Ground Truth
Malaria HVR6	307	6526	6	2	Cys Labels
Lawyers	71	339	18	6	Derived out of office and status features
World Trade	80	1000	16	5	Structural world system in 1980 features
Parliament	451	11646	108	7	Political parties
COSN	46	552	16	2	Region
Cora	2708	5276	1433	7	Computer Science research area
SinaNet	3490	30282	10	10	Users of same forum
Amazon Photo	7650	71831	745	8	Product categories

- **Eight popular real-world data sets**

Experiments settings

Data sets: synthetic data

- **Small-size networks:** 200 nodes; Five communities; Five features
- **Medium-size networks:** 1000 nodes; 15 communities; 10 features
- **8 settings, each setting has 10 repeats:**
- **Setting:**
  - **Links: within-cluster** $p = \{0.7,0.9\}$**, between-cluster** $q = \{0.3,0.6\}$
  - **Features:**
    - **within clusters: quantitative: Gaussian, random center from** $\alpha[-1, +1]$ **where** $\alpha \in \{0.7,0.9\}$
    - **: categorical: random center,** $\epsilon = \{0.7,0.9\}$
- **Total number of synthetic data sets: 800**

Experiments

SEFNAC on real-world data sets

Table 16 Comparison of CESNA, SIAN, EVA and SEFNAC on real-world datasets; average values of ARI and NMI are presented over 10 random initializations. The best results are shown in boldface

	CESNA		SIAN		EVA		SEFNAC	
	ARI	NMI	ARI	NMI	ARI	NMI	ARI	NMI
	Mean (std)	Mean (std)	Mean (std)	Mean (std)	Mean (std)	Mean (std)	Mean (std)	Mean (std)
HRV6	0.20 (0.00)	0.37 (0.00)	0.39 (0.29)	0.39 (0.22)	0.036 (0.004)	0.113 (0.006)	**0.45 (0.14)**	**0.62 (0.05)**
Lawyers	0.28 (0.00)	0.48 (0.00)	0.59 (0.04)	**0.71 (0.04)**	0.159 (0.028)	0.175 (0.026)	**0.63 (0.06)**	0.65 (0.05)
World Trade	0.23 (0.00)	0.59 (0.00)	**0.55 (0.07)**	**0.77 (0.03)**	−0.003 (0.000)	0.000 (0.000)	0.23 (0.03)	0.58 (0.04)
Parliament	0.25 (0.00)	0.52 (0.00)	**0.79 (0.12)**	**0.82 (0.07)**	0.005 (0.001)	0.001 (0.004)	0.28 (0.01)	0.47 (0.01)
COSN	0.44 (0.00)	0.45 (0.00)	0.43 (0.05)	0.61 (0.03)	−0.004 (0.000)	0.004 (0.000)	**0.50 (0.11)**	**0.64 (0.06)**
SinaNet	0.09 (0.00)	0.22 (0.00)	0.17 (0.02)	0.21 (0.02)	0.001 (0.002)	0.009 (0.002)	**0.21 (0.03)**	**0.29 (0.03)**

- **SEFNAC wins the HVR6, Lawyers, COSN and SinaNet competitions.**
- **SIAN wins the competition for PARLIAMENT, and takes second place in the LAWYERS competition. On average, it shows better performance than what we observed over synthetic data.**
- **CESNA except for COSN loses its efficiency. EVA again performs poorly.**

Experiments

SEFNAC on categorical synthetic data sets: small-size

Table 7 Comparison of CESNA, SIAN, EVA and SEFNAC on small-size synthetic datasets with categorical features. The best results are highlighted in boldface

Setting / Alg.	CESNA		SIAN		EVA		SEFNAC	
p, q, ϵ	ARI	NMI	ARI	NMI	ARI	NMI	ARI	NMI
	Mean (std)	Mean (std)	Mean (std)	Mean (std)	Mean (std)	Mean (std)	Mean (std)	Mean (std)
0.9, 0.2, 0.9	**0.95 (0.09)**	**0.97 (0.05)**	0.43 (0.27)	0.47 (0.83)	0.196 (0.072)	0.245 (0.040)	0.87 (0.02)	0.81 (0.02)
0.9, 0.2, 0.7	**0.90 (0.12)**	**0.94 (0.07)**	0.63 (0.25)	0.45 (1.15)	0.207 (0.057)	0.257 (0.024)	0.56 (0.06)	0.55 (0.04)
0.9, 0.4, 0.9	**0.97 (0.02)**	**0.95 (0.03)**	0.50 (0.29)	0.14 (1.40)	0.221 (0.065)	0.287 (0.043)	0.85 (0.02)	0.78 (0.03)
0.9, 0.4, 0.7	**0.96 (0.08)**	**0.96 (0.05)**	0.46 (0.27)	0.64 (0.39)	0.260 (0.087)	0.320 (0.043)	0.52 (0.05)	0.54 (0.03)
0.7, 0.2, 0.9	**0.95 (0.03)**	**0.94 (0.03)**	0.57 (0.23)	0.60 (0.62)	0.136 (0.051)	0.178 (0.032)	0.84 (0.03)	0.79 (0.03)
0.7, 0.2, 0.7	**0.82 (0.15)**	**0.88 (0.08)**	0.64 (0.07)	0.80 (0.04)	0.116 (0.039)	0.166 (0.028)	0.53 (0.06)	0.52 (0.05)
0.7, 0.4, 0.9	0.75 (0.11)	0.77 (0.09)	0.27 (0.22)	0.04 (1.44)	0.167 (0.037)	0.234 (0.040)	**0.82 (0.07)**	**0.77 (0.05)**
0.7, 0.4, 0.7	**0.50 (0.09)**	**0.59 (0.07)**	0.44 (0.30)	0.20 (1.34)	0.131 (0.047)	0.168 (0.058)	0.49 (0.10)	0.51 (0.07)

- **CESNA wins seven out of eight competitions.**
- **SIAN performs moderately acceptable.**
- **EVA performs poorly. Reasons: the assumption over sparsity of networks; inappropriate feature models(EVA); stocking in local optima**
- **SEFNAC also performs acceptably.**

Experiments

SEFNAC on categorical synthetic data sets: medium-size

Table 8 Comparison of CESNA, SIAN, EVA and SEFNAC on medium-size synthetic datasets with categorical attributes. Two best results are highlighted in boldface

Setting / Alg.	CESNA		SIAN		EVA		SEFNAC	
p, q, ϵ	ARI	NMI	ARI	NMI	ARI	NMI	ARI	NMI
	Mean (std)	Mean (std)	Mean (std)	Mean (std)	Mean (std)	Mean (std)	Mean (std)	Mean (std)
0.9, 0.2, 0.9	0.85 (0.03)	**0.91 (0.02)**	0.03 (0.08)	0.06 (0.19)	0.080 (0.047)	0.179 (0.031)	**0.87 (0.07)**	0.90 (0.04)
0.9, 0.2, 0.7	0.79 (0.05)	**0.90 (0.02)**	0.03 (0.09)	0.06 (0.19)	0.105 (0.077)	0.193 (0.051)	**0.89 (0.06)**	**0.90 (0.05)**
0.9, 0.4, 0.9	0.74 (0.08)	0.85 (0.04)	0.02 (0.05)	0.05 (0.15)	0.155 (0.043)	0.329 (0.048)	**0.85 (0.12)**	**0.88 (0.07)**
0.9, 0.4, 0.7	**0.66 (0.08)**	**0.80 (0.04)**	0.00 (0.00)	0.00 (0.00)	0.145 (0.075)	0.299 (0.055)	0.65 (0.14)	0.74 (0.07)
0.7, 0.2, 0.9	0.71 (0.10)	0.85 (0.04)	0.14 (0.17)	0.27 (0.33)	0.067 (0.031)	0.174 (0.039)	**0.85 (0.05)**	**0.88 (0.04)**
0.7, 0.2, 0.7	0.60 (0.10)	**0.80 (0.04)**	0.18 (0.21)	0.21 (0.31)	0.061 (0.032)	0.156 (0.031)	**0.67 (0.16)**	0.78 (0.09)
0.7, 0.4, 0.9	0.45 (0.09)	0.67 (0.06)	0.01 (0.04)	0.04 (0.11)	0.088 (0.040)	0.189 (0.049)	**0.71 (0.11)**	**0.76 (0.09)**
0.7, 0.4, 0.7	0.22 (0.04)	**0.54 (0.03)**	0.01 (0.04)	0.04 (0.13)	0.093 (0.024)	0.198 (0.034)	**0.37 (0.06)**	**0.54 (0.07)**

- **CESNA wins one out of eight competitions.**
- **SIAN and EVA perform poorly.**
- **SEFNAC wins the competition.**

Proposed methods (extended k-means)

KEFRiN: assumptions & notation

- **Consider a feature-rich network at the nodes,** $A = \{P, Y\}$**, over entity set "I";**
- **"I" is a set of network nodes of cardinality** $|I| = N$**.**
- $P = (p_{ij}) \in \mathbb{R}^{N \times N}$ **Matrix of mutual link weights between nodes"i" & "j";**
- $Y = (y_{iv}) \in \mathbb{R}^{N \times V}$ **Matrix of feature values** y_{iv} **for nodes** $i = 1,2,...,N$ **and over features** $v = 1,2,\ldots, V$**.**
- **We model feature part as:** $y_{iv} = \sum_k c_{kv} s_{ik} + f_{iv}$
- **We model network part as:** $p_{ij} = \sum_k \lambda_{kj} s_{ik} + e_{ij}$
- $k = 1,\ldots, K$ **Number of clusters;** $s_k = (s_{ik})$**: Binary N-dimensional cluster membership vectors (crisp clusters);**
- $c_k = (c_{kv})$**: V-dimensional cluster centroids;** $\lambda_k = (\lambda_{kj})$**: N-dimensional cluster centroids vector in network data space;**
- e_{ij} , f_{iv}**: Residuals (Errors) to be minimised;**

Proposed methods

KEFRiN: methodology-I

- **The least-squares criterion: minimise:**

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

- **w.r.t unknown membership vectors** s_k **, cluster centroids** c_k **and** λ_k **in feature data space and network links space**

- **Where** ρ , ξ **are user-defined constants,** $(\rho = \xi = 1)$

Proposed methods

KEFRiN: methodology-II

- **Simultaneous strategy: Search for all K clusters s_{ik} , c_{kv} , λ_{kj} simultaneously:**

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

- **Applying first order optimality and little algebraic manipulation:**

- $$c_{kv} = \frac{\sum_i y_{iv} s_{ik}}{|S_k|}; \qquad \lambda_{kj} = \frac{\sum_i p_{ij} s_{ik}}{|S_k|}$$

Proposed methods

KEFRiN: methodology-III

- **Expanding the criterion and optimal c_{kv} and λ_{kj} and substituting them below implies:**

$$F(s_{ik}) = \rho(\sum_{i,v} y_{iv}^2 - 2\sum_{k}\sum_{v} c_{kv}\sum_{i}(y_{iv}s_{ik}) + \sum_{k}\sum_{v} c_{kv}^2|S_k|) + \xi(\sum_{i,j} p_{ij}^2 - 2\sum_{k}\sum_{j}\lambda_{kj}\sum_{i}(p_{ij}s_{ik}) + \sum_{k}\sum_{j}\lambda_{kj}^2|S_k|)$$

- $T(Y) = \sum_{i,v} y_{iv}^2$ **and** $T(P) = \sum_{i,j} p_{ij}^2$ **: are quadratic scatters of Y and P, respectively.**

- **Breaking down $F(s_{ik}) = F_Y + F_P$ where, in respect, each represents the residuals of Y & P**

- $F_Y = \rho(T(Y) - \sum_{k,v} c_{kv}^2|S_k|)$ **and** $F_P = \xi(T(P) - \sum_{k,j} \lambda_{kj}^2|S_k|)$

Proposed methods

KEFRiN: methodology-IV

- **Breaking down $F(s_{ik}) = F_Y + F_P$ where, in respect, each represents the residuals of Y & P**
- **Thus:** $F_Y = \rho(T(Y) - \sum_{k,v} c_{kv}^2 |S_k|) = \rho d(c_k, y_{i:})$ **and similarly,** $F_P = \xi(T(P) - \sum_{k,j} \lambda_{kj}^2 |S_k|) = \xi d(\lambda_k, p_{i:})$
- F_Y & F_P **are indeed Euclidean distance; though it can be replaced with Cosine distance to tackle the curse of dimensionality (or any other distances metrics)**
- **To minimise our proposed clustering criterion, we extend the well-known K-Means algorithm, and we name it KEFRiN.**

A. KEFRiNe: represents the case when the Euclidean distance is being applied: $F(s_{ik}) = \rho d_e(c_k, y_{i:}) + \xi d_e(\lambda_k, p_{i:})$

B. KEFRiNc: represents the case when the Cosine distance is being applied: $F(s_{ik}) = \rho d_c(c_k, y_{i:}) + \xi d_c(\lambda_k, p_{i:})$

C. KEFRiNm: represents the case when the Manhattan distance is being applied: $F(s_{ik}) = \rho d_m(c_k, y_{i:}) + \xi d_m(\lambda_k, p_{i:})$

Proposed methods (extended K-means algorithm)

KEFRiN: methodology-V

1. **Initialisation: choose the number of clusters, K; initialise seed centroids: $C = \{c_K\}_{k=1}^{K}$ & $\Lambda = \{\lambda_K\}_{k=1}^{K}$, and empty cluster lists $S = \{S_k\}_{k=1}^{K}$.**

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 $d_c(\,.\,)$ or $d_m(\,.\,)$:**

   KEFRiNe: $d_e(y_{i:}, c_k) + d_e(p_{i:}, \lambda_k)$; **KEFRiNc:** $d_c(y_{i:}, c_k) + d_c(p_{i:}, \lambda_k)$; **KEFRiNm:** $d_m(y_{i:}, c_k) + d_m(p_{i:}, \lambda_k)$

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 the network space and go to Step 3.**

Shalileh, S. and Mirkin, B., 2022. Community Partitioning over Feature-Rich Networks Using an Extended K-Means Method. Entropy, 24(5), p.626.

Experiments

KEFRiN on real-world data sets

Table 9. Comparison of CESNA, SIAN, DMoN, SEANAC, KEFRiNe and KEFRiNc algorithms with Real-world data sets; average values of *ARI* are presented over 10 random initializations. The best results are highlighted in bold-face; those second-best are underlined.

Dataset	CESNA	SIAN	DMoN	SEANAC	KEFRiNe	KEFRiNc	KEFRiNm
HRV6	0.20(0.00)	0.39(0.29)	0.64(0.00)	0.49(0.11)	0.34(0.02)	**0.69(0.38)**	−0.056(0.004)
Lawyers	0.28(0.00)	0.59(0.04)	**0.60(0.04)**	**0.60(0.09)**	0.43(0.13)	0.44(0.14)	0.415(0.085)
World Trade	0.13(0.00)	0.10(0.01))	0.13(0.02)	0.29(0.10)	0.27(0.17)	**0.40(0.11)**	0.048(0.013)
Parliament	0.25(0.00)	**0.79(0.12)**	0.48(0.02)	0.28(0.01)	0.15(0.09)	0.41(0.05)	−0.035(0.001)
COSN	0.44(0.00)	0.75(0.00)	0.91(0.00)	0.72(0.02)	0.65(0.18)	**1.00(0.00)**	0.493(0.056)
Cora	0.14(0.00)	0.17(0.03)	**0.37(0.04)**	0.00(0.00)	0.00(0.00)	0.21(0.01)	−0.000(0.000)
SinaNet	0.09(0.00)	0.17(0.02)	0.28(0.01)	0.21(0.03)	0.31(0.02)	**0.34(0.02)**	0.001(0.000)
Amazon Photo	0.19(0.000)	N|A	**0.44(0.04)**	N|A	0.06(0.01)	0.43(0.06)	0.030(0.001)

- **SIAN wins the competition for PARLIAMENT, and takes second place in the LAWYERS competition. On average, it shows better performance than what we observed over synthetic data.**
- **CESNA except for COSN loses its efficiency.**
- **SEFNAC wins the Lawyers, and takes second place in WT competition.**
- **KEFRiNc and DMoN are close competitors and the dominating the table.**

Experiments

KEFRiN on categorical synthetic data sets: small-size

Table 6. Comparison of CESNA, SIAN, DMoN, SEANAC and KEFRiN algorithms on small-size synthetic networks with categorical features: The average and standard deviation of *ARI* index over 10 different data sets. The best results are shown in bold-face and the second-best ones are underlined.

Dataset	CESNA	SIAN	DMoN	SEANAC	KEFRiNe	KEFRiNc	KEFRiNm
0.9, 0.3, 0.9	**1.00(0.00)**	0.554(0.285)	0.709(0.101)	0.994(0.008)	0.886(0.116)	0.922(0.119)	0.895(0.173)
0.9, 0.3, 0.7	0.948(0.105)	0.479(0.289)	0.380(0.107)	**0.974(0.024)**	0.835(0.138)	0.819(0.142)	0.891(0.135)
0.9, 0.6, 0.9	0.934(0.075)	0.320(0.255)	0.412(0.109)	**0.965(0.013)**	0.963(0.072)	0.726(0.097)	0.868(0.202)
0.9, 0.6, 0.7	**0.902(0.063)**	0.110(0.138)	0.213(0.051)	0.750(0.117)	0.694(0.096)	0.711(0.145)	0.791(0.191)
0.7, 0.3, 0.9	0.965(0.078)	0.553(0.157)	0.566(0.105)	**0.975(0.018)**	0.788(0.117)	0.877(0.130)	0.937(0.124)
0.7, 0.3, 0.7	**0.890(0.138)**	0.508(0.211)	0.292(0.077)	0.870(0.067)	0.836(0.115)	0.795(0.117)	0.824(0.191)
0.7, 0.6, 0.9	0.506(0.101)	0.047(0.087)	0.345(0.064)	**0.896(0.067)**	0.762(0.169)	0.834(0.132)	0.379(0.174)
0.7, 0.6, 0.7	0.202(0.081)	0.030(0.040)	0.115(0.058)	**0.605(0.091)**	0.574(0.142)	0.540(0.107)	0.184(0.098)

- **CESNA wins three out of eight competitions.**
- **SIAN and DMoN perform moderately acceptable.**
- **SEFNAC wins the five remaining settings.**
- **KEFRiN methods perform acceptably.**

Experiments

KEFRiN on categorical synthetic data sets: medium-size

Table 7. Comparison of CESNA, SIAN, DMoN, SEANAC, and KEFRiN algorithms over medium-size synthetic networks with categorical features; average and standard deviation of *ARI* index over 10 different datasets. The best results are shown in bold-face and second ones are underlined.

Dataset	CESNA	SIAN	DMoN	SEANAC	KEFRiNe	KEFRiNc	KEFRiNm
0.9, 0.3, 0.9	0.894(0.053)	0.000(0.000)	0.512(0.137)	**1.000(0.000)**	0.508(0.205)	0.724(0.097)	0.863(0.089)
0.9, 0.3, 0.7	0.849(0.076)	0.000(0.000)	0.272(0.073)	**0.996(0.005)**	0.777(0.129)	0.742(0.182)	0.762(0.184)
0.9, 0.6, 0.9	0.632(0.058)	0.000(0.000)	0.370(0.063)	**0.998(0.002)**	0.279(0.204)	0.652(0.110)	0.894(0.074)
0.9, 0.6, 0.7	0.474(0.089)	0.000(0.000)	0.168(0.030)	**0.959(0.032)**	0.766(0.180)	0.733(0.083)	0.819(0.053)
0.7, 0.3, 0.9	0.764(0.068)	0.026(0.077)	0.446(0.099)	**1.000(0.001)**	0.364(0.247)	0.641(0.111)	0.791(0.119)
0.7, 0.3, 0.7	0.715(0.128)	0.000(0.000)	0.228(0.077)	**0.993(0.002)**	0.829(0.085)	0.797(0.088)	0.759(0.092)
0.7, 0.6, 0.9	0.060(0.024)	0.000(0.000)	0.332(0.051)	**0.998(0.001)**	0.426(0.246)	0.591(0.094)	0.859(0.083)
0.7, 0.6, 0.7	0.016(0.008)	0.000(0.000)	0.133(0.016)	**0.909(0.035)**	0.671(0.196)	0.773(0.070)	0.695(0.074)

- **CESNA takes the second place two times.**
- **SIAN performs poorly.**
- **DMoN's performance is relatively acceptable.**
- **SEFNAC is the winner.**
- **KEFRiN methods perform acceptably.**

Direction for future work

1. Automating choice of the weight coefficients $(\rho \ \& \ \xi)$
2. An interesting real-world application.
3. Extending the proposed methods in theory-driven framework
4. Accelerating the computational speed of the SEFNAC methods.
5. Applying KEFRiN methods at feature-rich networks using similarity data should be considered as another future work.

More related publications:

- Shalileh, S., & Mirkin, B. Community extraction in feature-rich networks with least-squares criteria using similarity data. PLoS One.

- 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).

Thank you!

sr.shalileh@gmail.com

Soroosh Shalileh