NICASN: Non-Negative Matrix Factorization and Independent Component Analysis for Clustering Social Networks

Ali Abbasi Tadi, Luis Rueda and Dima Alhadidi

May 21, 2022

NICASN: Non-negative Matrix Factorization and Independent Component Analysis for Clustering Social Networks

Ali Abbasi Tadi[†,*], Luis Rueda[†], Dima Alhadidi[†]
[†] School of Computer science, University of Windsor, Ontario, Canada

Abstract

Discovering clusters in social networks is of fundamental and practical interest. This paper presents a novel clustering strategy for large-scale highly-connected social networks. We propose a new hybrid clustering technique based on non-negative matrix factorization and independent component analysis for finding complex relationships among users of a huge social network. We extract the important features of the network and then perform clustering on independent and important components of the network. Moreover, we introduce a new k-means centroid initialization method by which we achieve higher efficiency. We apply our approach on four well-known social networks: Facebook, Twitter, Academia and Youtube. We experimentally show that our approach achieves much better results in terms of the Silhouette coefficient compared to well-known counterparts such as Hierarchical Louvain, Multiple Local Community detection, and k-means++.

Keywords: Network Clustering, Dimensionality Reduction, Non-negative Matrix Factorization, Independent Component Analysis, NMF-k-means, centroid initialization

1. Introduction

Social networks, paper citation networks, gene regulatory networks, and other large-scale networks have penetrated into all aspects of our real life. These networks usually have complex structure and various components [1]. Moreover, the high dimensional and sparse data of these networks have brought unprecedented challenges to existing network mining technologies. To address these challenges, network embedding is designed to learn the low dimensional representation of nodes, while preserving the structure and inherent characteristics of the network [2]. It can be effectively used by vector-based machine learning models for mining tasks, including node classification, personalized recommendation and link prediction [3].

Many relationships including those of people in social media can be embedded into a graph structure. However, blindly connecting all the entities together makes any network extremely complex. One way to better understand the network is to group its nodes based on their characteristics, such that nodes with similar specifications are grouped into a cluster. One way to classify nodes in a network is to categorize it in such a way that one node can be recognized by its group of neighbors. However, determining the proper group for the nodes in the network is not an easy task. In this regard, researchers have developed many clustering algorithms and then applied them to various datasets [4–9]. Nevertheless, there is no consensus on what the best clustering algorithm means. In general, one can claim that the best clustering algorithm is the one that gives more information about the nodes [10]. This can be accurately measured by using the Silhouette coefficient metric.

Youtube, Twitter, Facebook, and Academia datasets contain a list of all user-to-user links [11]. These links provide peer-to-peer relationships among different users. Generally, the related work in this domain targets a specific dataset with a limited number of nodes, does not consider the Silhouette coefficient as a performance metric, or is not efficient in terms of execution time. Our main objective in this work is to design an efficient clustering

[*] Abbasit@uwindsor.ca

approach, by which one can determine the best number of clusters for a large-scale and complex network, as well as the membership of users to clusters for a given number of clusters. In this work, we introduce a novel clustering algorithm for social networks. We experimentally show in terms of the Silhouette coefficient that our method outperforms other state-of-the-art approaches in this domain. Specifically, our contributions can be summarized as follows:

- introduce a new clustering approach by which one can perform community detection and find complex relationships among users. Our clustering approach uses Non-negative Matrix Factorization (NMF), Independent Component Analysis (ICA), and k-means. Our approach is computationally efficient as it uses sparse matrices.
- develop a new centroid initialization technique for k-means.
- analyze the experimental results of the proposed approach using the Silhouette coefficient by applying it to four well-known social networks: Facebook, Twitter, Academia, and Youtube, and compare the results with the state-of-the-art techniques.

The rest of this paper is organized as follows: related work is investigated in Section 2. The required background to understand the paper is detailed in Section 3. The proposed approach is presented in Section 4. Section 5 reports the experimental results. Section 6 concludes the paper and includes some future research directions.

2. **Literature Review**

k-means++ *et. al.* [12] is one of the most popular clustering techniques. This algorithm divides the network into k clusters, where each cluster is defined by a reference node (centroid). The remaining nodes are then partitioned and assigned appropriately to the clusters based on the closeness of each node to k reference nodes. Then, cluster adjustments are made with the calculation of new centroids. These centroids act as new reference points for the next partitioning of all the nodes.

Yazdanparast *et. al.* [4] proposed a new clustering technique for overlapping clusters using a fuzzy system. They developed Fast Fuzzy Modularity Maximization (FFMM) for finding communities in overlapping networks. They applied the modularity gain along with fuzzy membership value of network vertices to define proper communities. Their approach is simple in terms of computations overhead. However, it does not consider the Silhouette score as an evaluation metric.

Priyanka *et. al.* [5] proposed a new clustering technique using the Facebook social network. They proposed a model divided into various phases: a) sub-graph discovery, b) vertex clustering, and c) community quality optimization. For community detection they used social correlation theory, and finally applied the k-means++ clustering over the Facebook dataset. Their experiments revealed high accuracy in Facebook. Nonetheless, their approach has not been tested on other networks such as Youtube or Academia.

Blondel *et. al.* [6] proposed the very first version of the Louvain method. This method provides a way to value the existence of an edge between two vertices of an undirected graph by comparing it with the probability of having such an edge in a random model following the same degree distribution over the original network. The algorithm aims to increase the value of modularity by moving vertices from their community to any other neighbor community. Following their effort, Bhowmick *et. al.* [8] proposed an advanced version of Louvain by using hierarchical clustering as its embedding scheme. They obtained representations of individual nodes in the original graph at different levels of the hierarchy. Then, they aggregated these representations to learn the final embedding vectors. Their approach is scalable to any network and performs downstream network tasks such as node classification. However, in their approach, they did not consider very powerful performance metrics such as the Silhouette score on networks with different sizes.

Pradana *et. al.* [13] showed that hierarchical clustering performs better among other clustering approaches as it yields a higher Silhouette coefficient. Their paper compares the k-means++, hierarchical clustering [9], and hierarchical Louvain [8] to locate the most appropriate clustering technique in analyzing log activity data in Moodle Learning Management System. The results of clustering are measured using the Silhouette coefficient, and then compared the values and distribution between clusters. Their approach yields the highest Silhouette coefficient and can also detect outliers as a new cluster. However, it has not been applied to large-scale networks. Therefore, their results may vary when dealing with huge networks.

Kamuhanda *et. al.* [14] developed a new social network community detection algorithm, named Multiple Local Community (MLC). They apply Breadth-First search to sample the input graph up to a certain level, then they use Non-negative matrix factorization on the adjacency of the matrix of the sub-graph. After that they look at all nodes and try to add nodes to an appropriate sub-graph. In their work, they have neither considered modularity nor Silhouette score maximization. Instead, they tried to maximize conductance which is not an efficient metric in large-scale network.

Rozemberczk *et. al.* [15] introduced graph embedding approach using self-clustering. They used a machine learning technique to do clustering as well as embedding of social networks. Their work is susceptible to provide an inefficient clustering scheme as a result of changing the hyper-parameters of the neural network. In their work, they reached a good modularity on large-scale social networks, although they missed the evaluation of Silhouette coefficient for their work.

Sun *et. al.* [16] developed a probabilistic generative model called vGraph to learn community membership and node representation collaboratively. They consider that each node can be a mixture of communities and every community is defined as a multinomial distribution over nodes. Mixing coefficients and the community distribution evaluated the low-dimensional representations of the nodes and communities. Their approach works for overlapping and non-overlapping communities. However, it is computationally expensive, as it needs more than 1 week to compute communities in a big social network. Above this, they are not calculating Silhouette coefficient to show their work performance.

Skrlj *et. al.* [17] designed Silhouette Community Detection (SCD), an approach for detecting communities, based on clustering of network node embeddings. They used the non-Euclidean distance k-means and a new optimization technique for their proposal. Their approach works fine on protein interaction network. However, they have not evaluated their work on large scale social networks.

In general, the related work found in the literature targets a specific dataset with a limited number of nodes, whereas in this paper, we propose a method that works well on different large-scale social networks with different specifications. Additionally, we considered the Silhouette coefficient as a performance metric, while in most of the related work this important metric is disregarded and they only consider the modularity metric. We have also experimentally showed that our approach outperforms social network clustering algorithms in the literature when we deal with complex social networks.

3. Background

3.1. Network Embedding

By network embedding, one can transform the structure of a graph such as nodes and edges to feature vectors that are then mapped to dimensions while preserving the structure of the graph as much as possible. A social network could be represented by a large, and dynamic graph. Therefore, it is very difficult to find a comprehensive embedding approach. Each approach in this domain varies in performance on different datasets [10]. If we see embedding as a transformation to a lower dimension, it is a type of algorithm used in graph representation learning whose goal is to turn the network into a computationally digestible format. This is because networks' data types, by nature, are discrete.

3.2. **Network Clustering**

Detecting graph elements with “similar” properties is essential in large-scale networks, where it is crucial to identify specific patterns or structures quickly. The process of grouping similar elements together is called cluster analysis. Each cluster contains elements that share common properties and characteristics. In network clustering, datasets can be represented as a graph where each element to be clustered is represented as a node and the distance between two elements is modeled by a certain weight on the edge linking the nodes. A cluster in a network is intuitively defined as a set of densely connected nodes that is sparsely connected to other clusters in the network. However, there is no universal, precise mathematical definition of a cluster that is accepted in the literature [18]. There is a variety of different metrics that attempt to evaluate the quality of a clustering by capturing the notion of intra-cluster density and inter-cluster sparsity. Let $G = (V, E)$ be an undirected network with adjacency matrix A, where V is the set of vertices and E is the set of edges. In the following, we identify two most common clustering metrics: modularity and Silhouette coefficient.

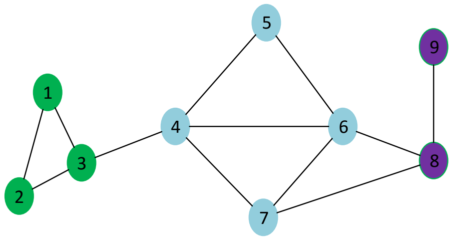

Figure 1. The sample network in which the modularity is 0.34 and the Silhouette score from node 1 is 0.63.

3.3. **Modularity**

The modularity of a network compares the presence of each intra-cluster edge of the graph with the probability that the edge would exist in a random graph [19]. Although modularity has been shown to have a resolution limit [20], the Louvain (LV) algorithm is applied as an objective function for optimization [6]. The value of the modularity for unweighted and undirected graphs ranges in $[-1, 1]$. Modularity is calculated by Eq. (3.1) as follows:

$$\sum_{k} (e_{kk} - a_k^2) \tag{3.1}$$

where, e_{kk} is the probability of edges in cluster C_k, and a_k is the probability of edges with at least one end in C_k. For example, assuming undirected graph as a bidirectional graph in Figure. 1, the modularity is equal to $(6/24-(7/24)^2)+(10/24-(13/24)^2)+(2/24-(4/24)^2) \approx 0.34$.

3.4. **Silhouette Score**

Silhouette refers to a method of interpretation and validation of consistency within clusters of data. The Silhouette varies in the range $[-1, +1]$, where a high value indicates that a node is well matched to its own cluster and poorly matched to other clusters [21]. If most nodes have a high value, then the clustering configuration is appropriate. The average of Silhouette scores for all nodes in the network is called the Silhouette coefficient. If many points have a low or negative value, then the coefficient becomes low or negative, and the clustering configuration may have too many or too few clusters.

Let $a(i)$ be the mean distance between i and all other data points in the same cluster, and $b(i)$ be the smallest mean distance from i to all data points in any other cluster. The Silhouette score is calculated by Eq. (3.2) as follows:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}, if |C_i| > 1 \tag{3.2}$$

where, $a(i)$ and $b(i)$ are calculated by Eq. (3.3) and (3.4), respectively.

$$a(i) = \frac{1}{|C_i| - 1} \sum_{j \in C_i, i \neq j} d(i, j) \tag{3.3}$$

$$b(i) = \min_{k \neq i} \frac{1}{|C_k|} \sum_{j \in C_k} d(i, j) \tag{3.4}$$

where, $|C_i|$ is the number of points in the cluster that point i belongs to, and $d(i, j)$ is the distance between points i and j in cluster C_i. In another word, $a(i)$ is seen as how well the data point i is assigned to its cluster. On the other hand, $b(i)$ can be seen as the dissimilarity of point i to a cluster $C_k, k \neq i$ [22]. If $s(i)$ is close to 1, it means that the data is appropriately clustered, whereas if $s(i)$ is close to -1, it reveals a very poor clustering. When $s(i)$ is close to 0, it means that there are very overlapping nodes among the clusters, and in this case it is better to deem the data as one cluster. For example, if we consider every edge as one distance in Figure. 1, starting from vertex 1, the Silhouette score is approximately 0.63. The Silhouette coefficient [23] for network clustering performs much more accurate than modularity. Thus, we measure our approach using the Silhouette coefficient.

Modularity is a common optimization objective used in many approaches in the literature [5, 8, 15, 16] for community detection. The weakness of modularity is that in some cases, it may provide incorrect measurements [23], i.e., when it is working on a cluster with only one vertex. Therefore, the Silhouette coefficient, in spite of its high computation requirements, provides a much better clustering measurement.

4. **NMF and ICA for clustering Social Networks (NICASN)**

In this section, we explore the details of our proposed approach to address the problem of community detection of a large-scale network. For clarity, we summarize our approach into three main steps. The proposed approach is described in Algorithm 1, where the inputs are I) an array of the number of components (for finding the features of the network), II) an array of the number of clusters (for finding communities by k-means), and III) the Compressed Sparse Row (CSR) matrix of the network. Using grid search, we aim at capturing the best values by which we obtain the highest coefficient. After receiving the inputs, Non-negative Matrix Factorization (NMF), Independent Component Analysis (ICA), and k-means are applied consecutively to compute the Silhouette coefficient. After these steps, we use the best coefficient and render the corresponding parameters. We delve into the detail of each step in the rest of this section. In the pre-processing stage, we need to change the adjacency matrix of the network to CSR format so that we can process the network using less memory in comparison to using the adjacency matrix.

Algorithm 1 NMF and ICA for Clustering Social Network (NICASN).

```
1: **Input:** Array of Clusters ($CL$), Array of Components ($CO$), the CSR matrix of the network ($data$)
2: **Output:** Best Score ($B_s$), Best Number of Clusters ($B_{cl}$), Best Number of Components ($B_{co}$), and Best Labels ($B_l$)
3: $CO \leftarrow [co_1, co_2, ..., co_n]$ {n is the number of input components}
4: $CL \leftarrow [cl_1, cl_2, ..., cl_m]$ {m is the number of clustering}
5: $Grid[n, m] \leftarrow 0$
6: $i, j \leftarrow 0$
7: for all $co \in CO$ do
8:     $i \leftarrow i + 1$
9:     for all $cl \in CL$ do
10:        if $cl > co$ then
11:            $i \leftarrow i - 1$
12:            Go to Step 9 {For checking other $cl$s}
13:        end if
14:        $j \leftarrow j + 1$
15:        $W, H \leftarrow NMF(data, co)$ {Decompose adjacency matrix}
16:        $ICA \leftarrow FastICA(W, co)$ {Transform feature matrix}
17:        $LABELS \leftarrow NMF\text{−k−}means(ICA, cl, co, H)$
18:        $SC \leftarrow Silhouette(data, LABELS)$ {Compute Silhouette coefficient}
19:        $Grid[i, j] \leftarrow SC$
20:        if $SC > B_s$ then
21:            $B_s \leftarrow SC$
22:            $B_{cl} \leftarrow cl$
23:            $B_{co} \leftarrow co$
24:            $B_l \leftarrow LABELS$
25:        end if
26:    end for
27: end for
28: return $B_s$, $B_{cl}$, $B_{co}$, $B_l$
```

4.1. NMF Transformation

Given non-negative matrix X, NMF basically finds two non-negative matrices (W, H) whose product approximates X [24]. The reason why NMF has become so popular is because of its ability to automatically extract sparse and easily interpretable factors in high-dimensional spaces. NMF inherently follows a spectral clustering and if we find the factor H by orthogonality constraint $(HH^T = I)$, then we obtain the centroids for k-means clustering initialization [25]. However, we are not using orthogonality constraint for our proposed centroid initialization. Instead, we use Non-negative Double Singular Value Decomposition (NNDSVD) [26]. NNDSVD is very well to initialize NMF algorithms with sparse factors (the factors that we see in social network clustering problem). Many experiments show that NNDSVD ends up with a very fast reduction of the approximation error of many NMF algorithms[26]. Therefore, the resulting decomposition $(W.H)$ would reproduce X with almost no error.

We call matrix W as the feature matrix while the matrix H as the importance matrix. We apply the importance matrix later in k-means centroid initialization (subsection 4.3) in order to do the clustering on the feature matrix. As we are dealing with a sparse matrix in social networks, we use NNDSVD to initialize NMF and follow its algorithm to obtain an accurate W. In a nutshell, the first transformation that we apply in our algorithm is to distinguish the feature (W) and importance (H) matrices of the original network by applying NMF (line 15 in Algorithm 1).

4.2. ICA Transformation

Here, we aim to reduce the dimensions of the feature vector that was already obtained from the NMF transformation. In order to find independent components of the feature matrix, we use ICA [27]. ICA is appropriate for non-orthogonal and non-Gaussian data. Looking at Algorithm. 1, we find co important independent components of the feature matrix. Then, the problem space is reduced to co dimensions, while we still preserve the internal structure of the network.

After all, we apply clustering by maintaining the Silhouette coefficient. Due to the fact that we are using k-means clustering, we need to initialize centroids to obtain better results. To initialize centroids we introduce NMF-k-means algorithm, which is shown in Algorithm 2.

Algorithm 2 NMF-k-means Algorithm

```
**Input:** Data (D), Number of Clusters ($cl$), Number of components ($co$), H values
**Output:** Nodes' labels
**for** $i = 1$ to $cl$ **do**
    **for** $j = 1$ to $co$ **do**
        $Centroid[i][j] \leftarrow (H[i][j])$
        $j \leftarrow j + 1$
    **end for**
    $i \leftarrow i + 1$
**end for**
$Labels \leftarrow k - means(D, cl, Centroid)$
**return** Labels
```

4.3. NMF-k-means Clustering

One of the important parts of our approach is the k-means centroids initialization. Algorithm 2 shows how we generate centroids to initialize k-means. For this purpose, we apply importance matrix, derived from NMF transformation. We clip the features from importance matrix according to the number of components, that was already provided in Algorithm 1. After applying NMF-k-means, we obtain the labels of the clustering. As the final step, we measure the quality of our clustering technique by finding the Silhouette coefficient. To make a valid comparison, we use the original data from the input (data), and the obtained labels. By this way, we can compute the Silhouette coefficient of other approaches and compare them with ours.

5. Experimental Results

We ran our experiments on four different social networks: Academia, Youtube, Twitter, and Facebook. The datasets were obtained from the Network Repository [11]; their characteristics are listed in Table 1. In the adopted networks, there are vertices with one edge

Table 1. Characteristics of the benchmark datasets.

Dataset	Nodes	Edges	Max. degree	Min. degree	No. of Triangles
Academia	200.2K	1.4M	11.4K	1	8.4M
YouTube	496K	1.9M	25.4K	1	7.3M
Twitter	404.7K	713.3K	626	1	88.6K
Facebook-Stanford	11.6K	568.3K	1.2K	1	17.5M

as well as vertices with thousands of edges. The number of triangles represents how dense the network is. The Max. degree is the maximum number of edges that connect to a node among all nodes whereas the Min. degree is the minimum number of edges that connect to a node among all nodes. As shown in Table 1, we evaluate networks with different numbers of vertices and various numbers of edges, which reveals the fact that our approach works with networks of any size.

5.1. **Setup**

We conducted our experiments on Sharcnet's Graham clusters [28]. We used Python 3.8. Furthermore, we used the Scikit-network library [29] to run the counterparts. The code is available for interested readers [30]. We used 10GB Memory and 1 GPU per experiment. In Algorithm 1, we applied sets $CO = \{50, 60, 70, 80, 90, 100\}$ and $CL = \{2, 3, 4, 5, 6, 7, 8, 9, 10\}$ for all the experiments.

5.2. **Clustering Implementation**

The outputs of our method on Academia, Twitter, Facebook, and Youtube are shown in Table 2. In all datasets, the highest coefficient belongs to the state when we apply two clusters. Generally, as the number of clusters increases, the Silhouette coefficient decreases. In our approach, the highest Silhouette coefficient reveals the correctness of the clusters. For example, for these datasets, two is the best number of clusters. However, if we are looking for more clusters, the coefficient falls steadily.

Table 2. Silhouette coefficient of the proposed approach.

	Number of Clusters								
Dataset	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Academia	0.94248	0.84705	0.63133	0.68494	0.57963	0.62570	0.55390	0.55003	0.53003
Facebook	0.58587	0.27558	0.48610	0.23614	0.22253	0.22370	0.21902	0.20319	0.19823
Twitter	0.92436	0.91962	0.91011	0.90865	0.76528	0.68709	0.65056	0.62376	0.60235
YouTube	0.98045	0.95593	0.95724	0.95683	0.951696	0.94964	0.93724	0.85008	0.84975

We have compared our approach with Hierarchical Louvain (HLV) [8], k-means++ [31], and NMF-based Multiple Local Community (MLC) [14] algorithms in terms of the Silhouette coefficient. To use similar setup to validate the comparison, we initialize HLV with resolution equal to one. This means that clusters with equal or greater than one member would be deemed as one cluster. In contrast to k-means++, HLV does not accept a specific number of clusters and finds the best number of clusters according to its hierarchical architecture. HLV follows a function which performs the act of transformation and returns a label vector for all node. In k-means++, however, the number of clusters should be given as input. For valid comparison with k-means++, we apply the set $\{2, 3, 4, 5, 6, 7, 8, 9, 10\}$ as the number of clusters. This is for comparing the outperformance of our approach versus k-means++ following the same setting.

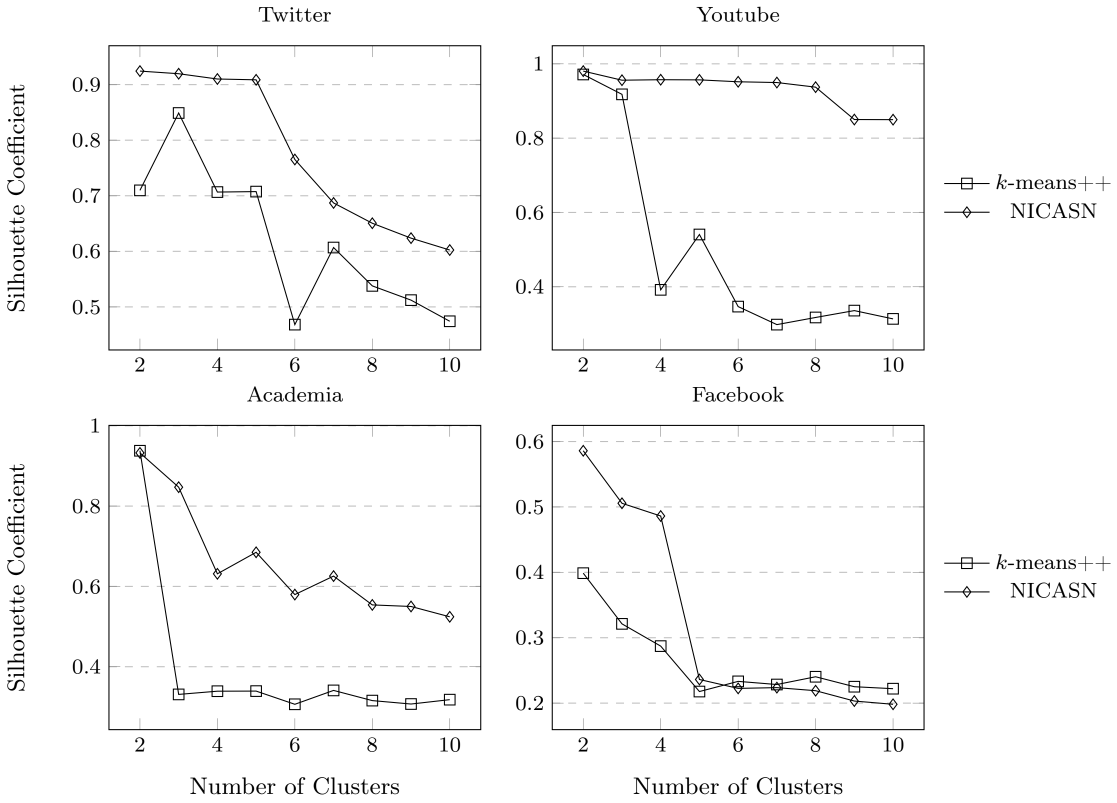

Figure 2. Comparison between k-means++ and NICASN

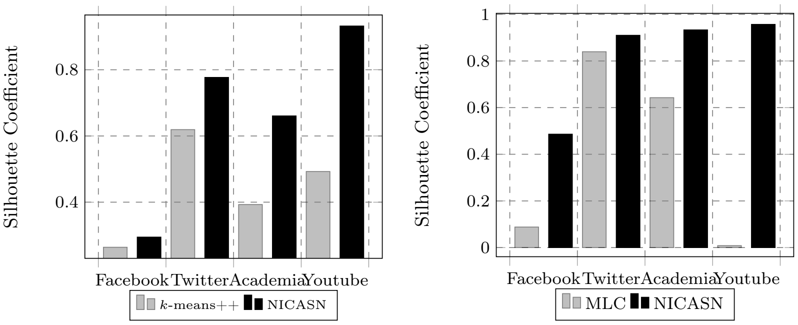

Figure 3. a) Comparison of k-means++ and NICASN (left), b) Comparison of MLC and NICASN (right)

Figure. 2 depicts the value of Silhouette coefficient with varying number of clusters. In Twitter, Youtube, and Academia, we see a significant improvement of NICASN, whereas, there is no significant improvement in Facebook when the number of clusters exceeds a threshold. The variations of the Silhouette coefficient of our approach when targeting different datasets is because of the varying number of connections (edges) in each social network. We averaged the Silhouette coefficient for each dataset by dividing them by 9 (as there are 9 number of clusters) and the results are shown in Figure. 3-a. As this figure shows, our method reaches much higher Silhouette coefficient in big networks in terms of the number

Table 3. HLV and NICASN Silhouette Coefficients.

Approach	**Dataset**	C_{opt}	**Silhouette Coefficient**
HLV	Academia	136	-0.352720204
HLV	YouTube	5375	-0.2710824
HLV	Twitter	888	-0.00261597
HLV	Facebook-Stanford	14	-0.352696225
NICASN	Academia	136	-0.07472
NICASN	YouTube	5375	-0.106410711
NICASN	Twitter	888	0.020000578
NICASN	Facebook-Stanford	14	0.179197

of connections between users. For instance, according to Table 1, Youtube has 1.9M edges, which is the biggest among the others. Therefore, our proposed approach on this network yields the best, 47%, improvement. Thus, The more edges in the network, the highest is the Silhouette coefficient in our work. The second best improvement belongs to Academia with 1.4M edges which provides 40% improvement, and the third best improvement goes for Twitter with 20% better results. On the other hand, we notice that in Facebook, because of the low number of connections (568.3K), there is no significant improvement in terms of the Silhouette coefficient. It is roughly 10% improvement.

MLC [14] has an internal mechanism to find the best number of clusters according to the input network using the Breadth-First Search (BFS). The best number of clusters derived from MLC for Academia, Twitter, Facebook, and Youtube are **2, 2, 4, and 5**, respectively. We passed these numbers to our approach to compare it with MLC. The comparison between our work and the existing NMF-based approach, MLC, is depicted in Figure. 3-b. As it is clear, our proposed approach significantly outperforms MLC in all datasets, specifically on Youtube. The reason behind the devastating output of MLC is that the k-means centroid initialization in MLC works worse than k-means++ for large-scale networks. Our experimental results show that, comparing to MLC, our approach yields much better Silhouette coefficient for all large-scale social networks. It is true that our approach, k-means++, and MLC are using k-means algorithm as their basis. However, each of these approaches use different centroid initialization strategy for their clustering purpose. This makes the big difference between the outputs of these three approaches.

Likewise MLC, HLV [8] internally looks for finding the best number of clusters. First, we run HLV on Facebook, Academia, Youtube, and Twitter datasets to obtain the optimum number of clusters. We call this number C_{opt}. Then, we run our approach with C_{opt} as input. By this way, we can compare the outputs of our approach with those of HLV. The number of clusters that HLV produced and their Silhouette coefficients are shown in Table 3. As expected, all the coefficients produced by HLV are negative, meaning that its clustering scheme is not performing very well on large-scale networks. For comparison, we provide the same table for NICASN to highlight the outperformance of our proposed scheme. For easier comprehension of NICASN outperformance, we illustrated the results in Figure. 4. We see a very good improvement in Facebook, Academia, and Youtube but a slight improvement in Twitter. As discussed before, because different C_{opt} is used in each network experiment, we cannot expect to see a large improvement in Youtube. Therefore, our latest justification on k-means++ and MLC is still effective. Meaning that, under the same number of clusters, when we deal with networks with high number of connections, we expect to see a much better outperformance in NICASN than HLV.

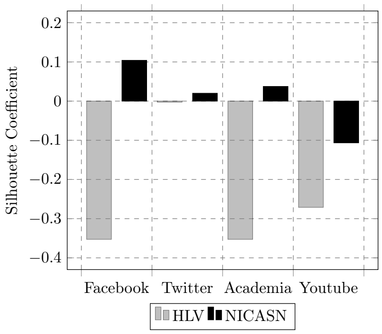

Figure 4. Average Silhouette Coefficient of HLV and NICASN with specific number of clusters shown in Table 3.

6. Conclusion

We introduced a novel clustering method based on Non-negative matrix factorization (NMF) and Independent Component Analysis (ICA). We applied NMF to extract the main features of the network and ICA to reduce the dimensions of features. Following by that k-means is applied to cluster the reduced-dimension network with newly-proposed centroid initialization mechanism based on NMF. The proposed approach is highly efficient in large-scale highly-connected social networks comparing to the state-of-the-art approaches including k-means++, Hierarchical Louvain, and Multiple Local Community. As future work, instead of using static set of components and clusters, a random approach can be conducted to find optimum values for those sets of components and clusters. Even, we can use a forward-propagation and backward propagation mechanism in neural network for finding the best number of components and clusters.

References

[1] B. Perozzi, R. Al-Rfou, and S. Skiena. “Deepwalk: Online learning of social representations”. In: *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2014, pp. 701–710.

[2] J.-H. Li, C.-D. Wang, L. Huang, D. Huang, J.-H. Lai, and P. Chen. “Attributed network embedding with micro-meso structure”. In: *International Conference on Database Systems for Advanced Applications*. Springer. 2018, pp. 20–36.

[3] S. Bandyopadhyay, H. Kara, A. Biswas, and M. N. Murty. “Sac2vec: Information network representation with structure and content”. In: *arXiv preprint arXiv:1804.10363* (2018).

[4] S. Yazdanparast, T. C. Havens, and M. Jamalabdollahi. “Soft overlapping community detection in large-scale networks via fast fuzzy modularity maximization”. In: *IEEE Transactions on Fuzzy Systems* 29.6 (2020), pp. 1533–1543.

[5] S. Priyanka and S. R. Krishna. “COMMUNITY DETECTION IN SOCIAL NETWORK USING GRAPH CLUSTERING METHODS”. In: *Turkish Journal of Computer and Mathematics Education (TURCOMAT)* 12.13 (2021), pp. 5687–5697.

[6] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. *Louvain method: finding communities in large networks*. 2008.

[7] N. Dugué and A. Perez. “Directed Louvain: maximizing modularity in directed networks”. PhD thesis. Université d’Orléans, 2015.

[8] A. K. Bhowmick, K. Meneni, M. Danisch, J.-L. Guillaume, and B. Mitra. “Louvainne: Hierarchical louvain method for high quality and scalable network embedding”. In: *Proceedings of the 13th International Conference on Web Search and Data Mining*. 2020, pp. 43–51.

[9] U. N. Raghavan, R. Albert, and S. Kumara. “Near linear time algorithm to detect community structures in large-scale networks”. In: *Physical review E* 76.3 (2007), p. 036106.

[10] S. Emmons, S. Kobourov, M. Gallant, and K. Börner. "Analysis of network clustering algorithms and cluster quality metrics at scale". In: *PloS one* 11.7 (2016), e0159161.

[11] R. A. Rossi and N. K. Ahmed. "The Network Data Repository with Interactive Graph Analytics and Visualization". In: *AAAI*. 2015. URL: http://networkrepository.com.

[12] B. Bahmani, B. Moseley, A. Vattani, R. Kumar, and S. Vassilvitskii. "Scalable k-means++". In: *arXiv preprint arXiv:1203.6402* (2012).

[13] C Pradana, S. Kusumawardani, and A. Permanasari. "Comparison clustering performance based on moodle log mining". In: *IOP Conference Series: Materials Science and Engineering*. Vol. 722. IOP Publishing. 2020, p. 012012.

[14] D. Kamuhanda and K. He. "A nonnegative matrix factorization approach for multiple local community detection". In: *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. IEEE. 2018, pp. 642–649.

[15] B. Rozemberczki, R. Davies, R. Sarkar, and C. Sutton. "Gemsec: Graph embedding with self clustering". In: *Proceedings of the 2019 IEEE/ACM international conference on advances in social networks analysis and mining*. 2019, pp. 65–72.

[16] F.-Y. Sun, M. Qu, J. Hoffmann, C.-W. Huang, and J. Tang. "vgraph: A generative model for joint community detection and node representation learning". In: *arXiv preprint arXiv:1906.07159* (2019).

[17] B. Škrlj, J. Kralj, and N. Lavrač. "Embedding-based Silhouette community detection". In: *Machine Learning* 109.11 (2020), pp. 2161–2193.

[18] A. Lancichinetti and S. Fortunato. "Community detection algorithms: a comparative analysis". In: *Physical review E* 80.5 (2009), p. 056117.

[19] M. E. Newman. "Fast algorithm for detecting community structure in networks". In: *Physical review E* 69.6 (2004), p. 066133.

[20] S. Fortunato and M. Barthelemy. "Resolution limit in community detection". In: *Proceedings of the national academy of sciences* 104.1 (2007), pp. 36–41.

[21] R. C. De Amorim and C. Hennig. "Recovering the number of clusters in data sets with noise features using feature rescaling factors". In: *Information sciences* 324 (2015), pp. 126–145.

[22] P. J. Rousseeuw. "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis". In: *Journal of computational and applied mathematics* 20 (1987), pp. 53–65.

[23] H. Almeida, D. Guedes, W. Meira, and M. J. Zaki. "Is there a best quality metric for graph clusters?" In: *Joint European conference on machine learning and knowledge discovery in databases*. Springer. 2011, pp. 44–59.

[24] C. Ding, X. He, and H. D. Simon. "On the equivalence of nonnegative matrix factorization and spectral clustering". In: *Proceedings of the 2005 SIAM international conference on data mining*. SIAM. 2005, pp. 606–610.

[25] X. Shi, H. Lu, Y. He, and S. He. "Community detection in social network with pairwisely constrained symmetric non-negative matrix factorization". In: *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015*. 2015, pp. 541–546.

[26] C. Boutsidis and E. Gallopoulos. "SVD based initialization: A head start for nonnegative matrix factorization". In: *Pattern recognition* 41.4 (2008), pp. 1350–1362.

[27] J. V. Stone. "Independent component analysis: a tutorial introduction". In: (2004).

[28] Sharcnet. *SHARCNET (www.sharcnet.ca) is a Consortium of 19 universities and research institutes operating a network of high-performance computer clusters across south western central and northern Ontario*. https://www.sharcnet.ca/. 2016.

[29] T. Bonald, N. de Lara, Q. Lutz, and B. Charpentier. "Scikit-network: Graph Analysis in Python". In: *Journal of Machine Learning Research* 21.185 (2020), pp. 1–6. URL: http://jmlr.org/papers/v21/20-412.html.

[30] A. A. Tadi. *NICASN: Non-negative matrix factorization and Independent Component Analysis for clustering Social Networks*. https://github.com/humanworth/NICASN. 2022.

[31] H. Gabow. *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, 2007.