Applied Soft Computing 47 (2016) 216–223

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Graph clustering using k-Neighbourhood Attribute Structural similarity

M. Parimala Boobalan [a,*], Daphne Lopez [a], X.Z. Gao [b]

[a] School of Information Technology and Engineering, VIT University, Vellore 632 014, Tamil Nadu, India
[b] Aalto University School of Electrical Engineering, Otaniementie 17, 00076 Aalto, Finland

ARTICLE INFO

Article history:
Received 19 June 2015
Received in revised form 23 April 2016
Accepted 18 May 2016
Available online 7 June 2016

Keywords:
Clustering
graph
k-Neighbourhood
Structural
Attribute similarity

ABSTRACT

A simple and novel approach to identify the clusters based on structural and attribute similarity in graph network is proposed which is a fundamental task in community detection. We identify the dense nodes using Local Outlier Factor (LOF) approach that measures the degree of outlierness, forms a basic intuition for generating the initial core nodes for the clusters. Structural Similarity is identified using k-neighbourhood and Attribute similarity is estimated through Similarity Score among the nodes in the group of structural clusters. An objective function is defined to have quick convergence in the proposed algorithm. Through extensive experiments on dataset (DBLP) with varying sizes, we demonstrate the effectiveness and efficiency of our proposed algorithm k-Neighbourhood Attribute Structural (kNAS) over state-of-the-art methods which attempt to partition the graph based on structural and attribute similarity in field of community detection. Additionally, we find the qualitative and quantitative benefit of combining both the similarities in graph.

1. Introduction

Clustering is an important data mining technique developed for the purpose of identifying groups of entities [1,2] that are similar to each other using some similarity measures. The main goal of clustering is to have high intra cluster similarity and low inter cluster similarity i.e., the objects inside the cluster are similar and objects in different cluster are dissimilar. It is an unsupervised learning technique widely used in all the areas of science and engineering that includes bioinformatics, market research, social network analysis, image analysis, financial and marketing field, trajectory data, time series data, spatial data and so on.

Graph structure is an expressive data structure [3] model which studies the relationship among the objects in the application like social networks, sensor networks and biological networks. Recently, graph clustering [4] has gained the attention of the researchers due to its rapid expansion and fast proliferation in the field of many applications. Clustering on large graph aims to partition the graph into several densely connected subgraphs [5–9] that is useful to understand and visualize large graphs. Graph clustering includes community detection in social networks analytics [10–12], protein–protein interaction biological networks [13], document clustering, citation network [14,15] and others. Due to the extent and the diversity of contexts in which graphs are used, the area of graph clustering has become both crucial and interdisciplinary, in order to understand the features, the structure and the dynamics of these complex systems. The major difference between graph clustering and traditional data clustering is that graph clustering measures the connectivity (number of possible edges between two vertices) while data clustering measure distance between two objects based on Euclidean distance. This distance measure fails to detect cluster in dense set of objects that can represent in arbitrary shape, as Euclidean distance favours compact and spherical shaped clusters.

Community detection in graph refers to grouping of similar nodes that share a common characteristics or properties [16–19]. Similarity is measured in two ways, Structural similarity which considers the topological features and Attribute similarity that groups the similar characteristics related to nodes and edges [20,21]. In information network analysis, the characteristics or roles of a person is considered as attribute similarity, whereas the structural similarity is the interaction or relationship among the group of people. The vertices are assigned to cluster based on any of these two similarities. However, the existing graph clustering algorithms focus on anyone the similarity to partition the graph. But in many real applications, both the structural and attribute similarity plays a

[*] Corresponding author.
E-mail address: parimala.m@vit.ac.in (M.P. Boobalan).

http://dx.doi.org/10.1016/j.asoc.2016.05.028
1568-4946/© 2016 Elsevier B.V. All rights reserved.

major role in the analysis. A good clustering algorithm should generate clusters which have a cohesive intracluster similarity with homogeneous attribute values by balancing both the similarities [22,23].

The proposed work in this article aims to partition the graph based on Structural and Attribute similarities. An object should strictly satisfy both the similarities defined above. As discussed earlier, it is challenging task to group the individuals who are friends with same affiliation. The main contributions of this article are summarized as follows.

(i) *Input parameters for the Algorithm*: There is only one input [24] given to the proposed graph clustering algorithm. The distance (k) value used to find the k-neighbourhood in Structural Similarity is the only parameter given to the algorithm.
(ii) *Quick convergence*: An efficient objective function is defined which makes the algorithm to converge quickly. The experiments have demonstrated that our proposed clustering approach is able to partition the graph into quality clusters with high structural similarity and homogeneous attribute values in large scale real social graphs.
(iii) *Time complexity*: Since the random initialization of the centroid is ignored, it takes less execution time for large dataset when compared to other existing graph clustering algorithm.
(iv) *Automatic detection of centroids*: Local Outlier Factor measure is used to detect the initial centroids for the clustering process. The vertex which is close to the average density of the cluster is chosen as the next centroids iteratively until the objective function converges.
(v) *Strict Attribute and Structural Similarity*: In the existing graph clustering algorithms, the vertex closeness of the object is measured based on the degree of structural or attribute similarities. Most of the algorithms provide a good balance between both of these similarities, but the proposed clustering algorithm strictly satisfies both the similarities. The objects within the cluster are similar with respect to attribute and connectivity.
(vi) *Robust to outlier*: LOF measure is used initially to ignore the outlier objects. The algorithm is efficient and robust to handle sparseness and noise data in the graph. The presence of Outlier objects does not affect the clustering result, as we measure the degree of outlierness of each object in the initial process of the graph.
(vii) *Overlapping cluster*: The author's area of interest is considered to be the attribute information for the DBLP dataset. In this case, the authors would be specialized in more than one research area. So, there is a chance that one vertex present in more than one cluster, it means that one author would be grouped into more than one cluster based on their area of expertise.

The rest of this article is organized as follows. Section 2 discuss about the related work on Graph Clustering. Section 3 presents the preliminary concepts and the design of proposed algorithm followed by Section 4 that provides the experimental analysis and validation on the clustering results on various datasets. Section 5 discuss on DBLP dataset. Finally, Section 6 concludes the article.

2. Related work

Graph clustering using mutual KNN [25] neighbours (G-MKNN) is based on node affinity measure and edge weights helps to capture low and high dense clusters. It fails to deduct clusters on sparse and incomplete graph. Spectral clustering [26] method actively selects the pairwise constraints based on novel notion of node uncertainty rather than pair uncertainty. It consumes more time to construct laplacian matrix and eigen vectors. When compared to the above algorithms the proposed work takes less time to cluster as it depends upon the neighbourhood distance matrix.

Hierarchical clustering [27] approach is used find structured similarities for k nodes in incomplete information network. Distance based modularity is used to check quality of the clusters. In graph based K-means clustering [28] algorithm, the number of cluster (k) is determined based on prims trajectory. A threshold value is constant along the prims trajectory. When the data set size increases the quality of the cluster is decreased and it takes more execution time. The proposed algorithm has proved that it is scalable for various data size and efficient in terms of execution time.

Recently, the graph algorithms focus on clustering based on both structural and attribute similarities. CODICIL [29] is a framework that uses Metis and Markov Clustering to combine both content and link similarity. The link strength is based on the probability of an edge belongs to the community and content similarity is estimated using Jaccard coefficient. Similarly, a good balance between structural and attribute similarities through unified distance measure and neighbourhood random walk is proposed in SA-cluster [30]. The graph is partitioned into k clusters, so that each cluster contain densely connected subgraph with homogeneous attribute values. The density of cluster is high and the entropy value is high in S-cluster [31] compared to SA-cluster. In general S-cluster has high structural similarity and low attribute similarity compared to SA-cluster. The proposed algorithm (kNAS) focuses on strict structural-attribute similarity which means that the objects with k-neighbourhood and similar attributes only are grouped into a cluster. In this way, kNAS algorithm does not balance both the similarities rather it strictly satisfies the structural and attribute similarity.

3. k-Neighborhood Attribute Structural (kNAS)

An attributed graph is denoted as $G = (V, E, A)$ where V is set of vertices, connected with set of E edges and $A = \{a_1, \ldots a_\ell\}$ is a set of ℓ attributes associated with each vertex $v_i \in V$ that describes the properties with an attribute vector $\{a_1(v_i), \ldots\ldots, a_\ell(v_i)\}$ and N denotes the number of vertices in graph $|V| = N$. A graph is partitioned into m overlapping clusters where $V_1 \cup \ldots \cup V_m \subseteq V$ and $V_i \cap V_j \neq \phi$ for any $i \neq j$. The proposed clustering algorithm k-Neighborhood Attribute Structural (kNAS) achieve the following two properties: (1) vertices within one cluster are close to each other in terms of structural similarity and distant from each other between the clusters (2) vertices within cluster have similar properties in terms of attribute similarity and dissimilar among the clusters. The main issues are: (1) selecting initial value for m (2) clustering algorithm based on structural and attribute similarity (3) defining the objective function that converges quickly. The importance of considering the structural and attribute similarities is explained with a simple social network. For example, we consider a social network example (Fig. 1) in which vertices A–D are considered as individuals and edge represents the relationship between the individuals (friend relationship). The affiliation of the individual is taken as the vertex properties for which the values are 'x' and 'y'.

Structure-based clustering: The closeness of vertex is measured based on connectivity between the vertices. Objects within cluster are closely connected than objects in different cluster. In Fig. 1(b) the individuals with friend relationship are grouped together irrespective of the similar attributes. Since the objects are clustered based on connectivity, it ignores group the objects based on similar attributes. However, in one cluster the individuals have quite

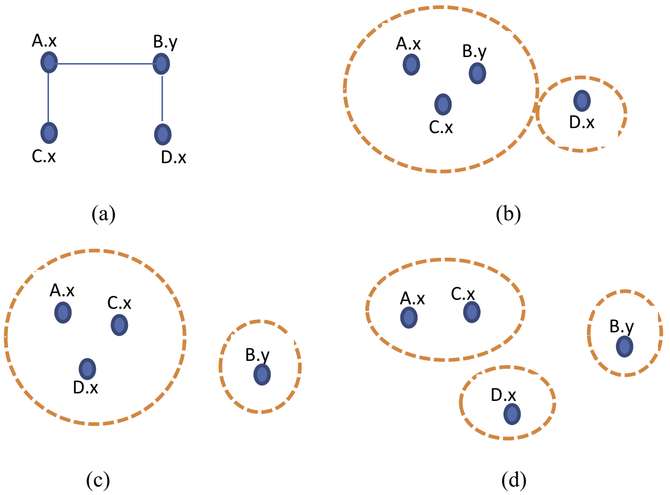


Fig. 1. (a) Denotes social network graph (b) structural graph cluster with heterogeneous attributes (c) attributed graph cluster with homogeneous attributes (d) attribute–structural cluster with intracluster similarity and homogeneous attributes.

different affiliation, for example, A and B with affiliation '*x*' and C with affiliation '*y*'.

Attribute-based clustering: The vertices are grouped based on similar attributes. Individuals with similar properties are grouped in a cluster and individuals in different cluster have dissimilar properties. For example, in Fig. 1(c) all the individuals A–D belong to the same affiliation '*x*' but there is no relationship between A and D. In this way, the attribute based clustering ignores the structural similarity among the vertices.

Structural-Attribute based clustering: The grouping of objects is based on both structure and attribute information. In Fig. 1(d) the cluster contains individuals with 'friend relationship' who belong to same affiliation. It provides a good balance between both structural and attribute similarities. The goal of this study is to satisfy this property on clustering.

3.1. Cluster centroid initialization

There are enormous numbers of initialization methods for centroid initialization in formation of cluster. The first scheme of centroid initialization was proposed by Ball [32]. This method works based on the parameter *d*, that defines the minimum distance between any two centroids. The recent initialization method k-means++ approach proposed by Ref. [33] depends upon the selection of subsamples. However, most of centroid selection approach is sensitive to outlier/noise data and the results depend on selection of input parameter. Local Outlier Factor [34] is used for selecting the initial set of cluster centroids. This measure is proposed as a measure to determine the degree to which a point is core or outlier point. It assigns each object a degree of being outlier, that degree is called Local Outlier Factor. The local degree depends on how isolated the object is with respect to the surrounding neighbourhood.

Definition 1. k-Distance Neighbourhood

Given a value of, $\text{dist}(x, y)$ the distance between the objects x and y is calculated using Manhattan distance measure and $\text{dist}_k(x)$ is the distance of k th nearest neighbour of x. Then the k th nearest neighbourhood of x contains every object whose distance from x is not greater than k th distance, such as

$$N_k(x) = \{y | y \in D, \text{dist}(x, y) \le \text{dist}_k(x)\}$$

Even though, the $\text{dist}_k(x)$ is well defined for any positive integer k, the neighborhood object y may not be unique. So, in this case the cardinality of $N_k(x)$ is greater than k. For example, if there are 2 objects for 1-distance from object x and 3 objects for 2-distance from object. Then the cardinality of neighbourhood of $|N_2(x)| = 5$ is greater than the value of 2.

Definition 2. Relative density function

Let $N_k(x)$ denotes the k th nearest neighbourhood of x, $|N(x)|$ represents the number of neighbourhood points of x. The relative density of x is then computed as follows,

$$rd_k(x) = (\frac{\sum_{(y \in N_k(x)} \text{dist}_k(x, y)}{|N_k(x)|})^{-1}$$

The relative density of an object x is the inverse of the average distance based on the k-nearest neighbours of x. Note that the relative density can be ∞, if all the summation of distance between x and y are 0. This may occur if the neighbour points share the same spatial coordinates or if there exist duplicates of x in the dataset. For simplicity, we assume that no two objects share the same spatial co-ordinates.

Definition 3. Local Outlier Factor

Assume that the relative density value of each object is determined by previous section. The LOF identifies the degree of x being outlier. It is defined as the average local density of x with the y and k-nearest neighbours of x.

$$\text{LOF}_k(x) = \frac{\sum_{y \in N_k(x)} \frac{rd_k(y)}{rd_k(x)}}{|N_k(x)|}$$

If a relative density of a point is low with density of its neighbours, the value of LOF is high. Thus, the LOF value determines the extent of being outlier with varied density of clusters. An object whose LOF value approximately equal to 1, exactly belongs to a cluster, as its density and the density of its neighbours are same. Some extensive properties of LOF [35]: (i) efficient in making a decision of degree to which a point is outlier. The objects belonging to cluster will assume an LOF value close to 1. (ii) It leads to faster convergence of any density based clustering algorithms (iii) it is robust with clusters having varied density and size. According to the LOF function, we sort all vertices in the ascending order of their LOF values. Then vertices with LOF values less than or equal to 1 are selected as initial centroids $\{c_1^0, \ldots\ldots c_m^0\}$.

The complexity analysis for finding the initial centroid using LOF value is analysed based on the steps involved. The cost of LOF value for each examined point is given by $O(nk)$ where n is the number of vertices and k denotes the k neighbourhood distance of the object. This computation is done for m centroids in the worst case. As a result, the overall complexity for calculating LOF value is $O(mnk)$.

3.2. Structure based clustering

Clustering based on structure is grouping of objects based on k-nearest neighbour distance measure. The set of $m\{c_1^0, \ldots\ldots c_m^0\}$ initial centroids are selected using LOF method. For each n th iteration, we assign the vertex $v_i \in V$ to its closest centroid $c^* \in \{c_1^0, \ldots\ldots c_m^0\}$.

$$c^* = \{v_i | v_i \in V, \min_{c_j^n} d\left(v_i, c_j^n\right)\}$$

where the $d\left(v_i, c_j^n\right)$ is the nearest distance between the vertex and the centroid. For a dataset size of n and number of centroids to be m, the complexity for grouping the objects based on structural similarity is $O(nm)$ where $n \gg m$.

3.3. Attribute based clustering

Grouping of objects are based on the properties defined for each object is known as Attribute based Clustering. The similarities among the properties of object are measured using the Similarity matrix. The complexity for generating the similarity matrix is considered as $O\left(n^2\ell\right)$ where n is the number of vertex and ℓ is the dimensionality of each vertex.

Definition 4. Similarity Matrix

Let ℓ denotes the set of attributes defined for each vertex. The similarity between any two vertices v_i and v_j with respect to attribute is indicated as $S_{ij.}$

$$S_{ij} = \frac{1}{\ell}\sum_{a=1}^{\ell} S_{ija}$$

where,

$$\mathrm{S_{ija}} = \begin{cases} 0, \text{if} v_i \text{ and} v_j \text{ do not match} a \text{ th attribute} \\ 1, \text{if} v_i \text{ and} v_j \text{ match for the} a \text{ th attribute} \end{cases}$$

Properties. If any two objects v_i and v_j are similar then, the Similarity Matrix (S) satisfies the following properties,

(i) Symmetry: $S\left(v_i, v_j\right) = S\left(v_j, v_i\right)$
(ii) Positivity: $0 \le S\left(v_i, v_j\right) \le 1 \forall v_i$ andv_j
(iii) Reflexivity: $S\left(v_i, v_j\right) = 1$ if$i = j$

The following properties holds good on Dissimilarity Matrix (D), if any two objects v_i and v_j are dissimilar,

(i) Symmetry: $D\left(v_i, v_j\right) = D\left(v_j, v_i\right)$
(ii) Positivity: $\mathrm{D}\left(v_i, v_j\right) \ge 0 \forall v_i$ andv_j
(iii) Reflexivity: $\mathrm{D}\left(v_i, v_j\right) = 0$ if$i = j$

3.4. Update cluster centroid

The initial centroids $\left\{c_1^0, \ldots\ldots c_m^0\right\}$ for m clusters $\{V_1 \ldots V_m\}$ are selected using LOF values for n th iteration. When all the vertices are assigned to some clusters mutually, the centroid will be updated with the centrally located vertex in each cluster. The point which is close to the average point in a cluster is centrally located point. This point is updated as a new centroid for the next $n + 1$ iteration. The Average point of Cluster V_i is defined as,

$$\mathrm{Avg}(V_i) = \frac{1}{|V_i|}\sum_{j \in V_i} d\left(v_j, v_i\right), \forall v_i \in V$$

Then, we find a new centroid c_i^{n+1} in the $(n + 1)$ th iteration whose distance to closest to the average point of that cluster V_i. The cluster centroids are updated for each iteration as,

$$c_i^{n+1} = \min_{v_j \in V_i} \|v_j - \mathrm{Avg}(V_i)\|$$

Let s be the number of objects in a cluster. Now, the complexity analysis for computing the average point of cluster V_i is $O(nms)$ where n is the number of vertex and m is the number of cluster. For each iteration, the time taken for updating the centroid with new centroid is $O(ms)$. Finally the complexity for the overall process for updating the cluster centroid requires $O(ms(n + 1))$ time.

3.5. Merging the clusters

The minimum distance between the centroids are merged into a single cluster. For example, if the distance of centroid c_1 is minimum with c_2 than c_3, then c_1 is merged with c_2. $d(c_1, c_2) < d(c_1, c_3)$ then c_1 and c_2 are merged together. This would result in a reduced number of centroids generated in each iteration and faster overall computation time of $O(m)$ where m is the number of clusters.

3.6. Clustering objective function

The main aim of clustering process is to minimize intercluster similarity and to maximize intracluster similarity. An objective function is defined to maximize the structural density and the Similarity Score, thereby maximizing the intracluster similarity. Since we group the objects based on the k-nearest neighbourhood distance, the objective function focus only on the maximizing the attribute Similarity Score.

Definition 5. Similarity Score (SC)

Let V_i, V_j be two vertex sets. The Similarity Score between $\mathrm{SC}(V_i, V_j)$ between V_i andV_j is defined as,

$$\mathrm{SC}\left(V_i, V_j\right) = \sum_{v_i \in V_i, v_j \in V_j} \frac{S_{ij}\left(v_i, v_j\right)}{|V_i| \times |V_j|}$$

where S_{ij} is the similarity value defined between v_i and v_j. The above Similarity Score would quantitatively measure the extent of similarity between the attributes. When the attributes of two vertices are more dissimilar, then the value of SC will be low. Obviously, the SC value will be high for vertices having more similar attributes.

Definition 6. Objective Function

Given a group of Clusters $V_i \ldots V_m$ of m clusters, where V_i corresponds to the i th cluster, the objective function to be maximized is defined as follows,

$$O\left(\left\{V_i\right\}\right)_{i=1}^{k} = \sum_{i=1}^{k} \mathrm{SC}(V_i, V_i)$$

The clustering algorithm is iterated until the above objective function converges. Till it reaches a high intra cluster similarity the clustering objective function is maximized.

Algorithm. Input: Graph G with vertex set V, edge set E, and attribute set A and the distance value (k). Output: m clusters $V_i \ldots V_m$

1: Calculate the initial centroids using Local Outlier Factor
2: Repeat until the objective function converges
3: Assign each vertex to the nearest centroid
4: Partition the cluster based on the similarity of the vertex
5: Merge the nearest clusters
6: Update the cluster centroid
7: Return m clusters

Initially the k-distance neighbourhood of the vertices are calculated. Then using Local Outlier Factor the vertex having high density are selected as the m core points. Structural similarity is achieved by taking the k-neighbourhood of the core points that are grouped into m partitions. Then, similarities of attributes among the vertices are performed using the Similarity Score and grouped into m clusters. The vertex which is closer to average density of the cluster is updated as a new centroid. The following process is iterated until the objective function is maximized. This is analogous to,'The rich get richer and the poor get poorer' principle as the vertices in the clusters within the k-neighbourhood having similar properties becomes denser with the centroid updating for the

Table 1
Summary of experimental data sets.

Dataset	Nodes	Edges	Clusters (m)
DBLP-1	8781	4789	471
DBLP-2	17,578	12,545	2179
Facebook	4089	170,174	193
Twitter	81,306	1,768,149	3140

Table 2
Distortion Score for different centroid initialization method on DBLP-1 dataset.

Number of clusters (m)	Random	LOF	k-means ++
100	9925	**7225**	8700
200	8020	**7865**	7950
300	6900	6436	**6200**
400	6500	**6100**	6376
500	8650	**5500**	7893

next iteration. The complexity for calculating the objective function is $O\left(m^2\right)$. Finally, adding up the costs of the steps, the overall computation complexity of the entire process of kNAS algorithm is $O\left(e.\left(nm+n^2\ell+nms+ms+m^2\right)\right)$ where e represents the number of iterations the algorithms runs till the objective function gets maximized. In the next section we demonstrate through experiments that our proposed clustering algorithm is more efficient than other existing algorithm and converges very quickly on the given dataset.

4. Implementation

Extensive experiments are performed to evaluate the performance of our proposed algorithm k-Neighborhood Attribute Structural (kNAS) with the state-of-art algorithms SA-Cluster-Opt, NISE and CODICIL on real graph datasets.

4.1. Experimental datasets

We use two real graph datasets for the evaluation of proposed algorithm for graph clustering. The number of nodes and edges in each dataset is summarized in Table 1. The number of cluster (m) formed using our proposed method is also denoted in the table.

DBLP-1 is a dataset extracted from DBLP database that provides the bibliographic information about the co-authors in the field of computer science journals and proceedings. We gather information from three research fields including Database, Data mining and Image Processing. We create a co-author graph where the authors who have published more than three research papers during the period of 2000–2014 are considered as nodes and any pair of authors who have co-authored are linked as edges of the graph. This co-author graph contains 8781 nodes and 4789 edges. Each node is attached with the information about the author and their list of research areas.

DBLP-2 is used inorder to test the efficiency and scalability of our proposed method. We use large dataset with 17,578 nodes and 12,545 edges. This dataset contains the information selected from the following areas: Multimedia, Data Mining and Bioinformatics. The same setups with DBLP-1 are used to build the co-author graph for this dataset.

The last two networks (Facebook and Twitter) are ego networks from online social network services that are collected from the Stanford Large Network Dataset collection (http://snap.stanford.edu/data). The user profile information such as gender, job, institution and hobby are considered as the node attributes for the Facebook network. In Twitter network, the hashtags used by the user in their tweets are defined as the attributes for nodes.

4.2. Comparison of initialization methods

Random, k-means++ and LOF initialization method is compared on DBLP-1 data set with varied density and sizes. Distortion score is used as an evaluation metric to compare the effectiveness of the LOF over the random initialization method. The sum of the distance between each point to its closest centre is defined as the distortion score. Smaller the values of distortion score, better the clustering result. Multiple runs are performed on DBLP-1 using the algorithms that depend either on the order of objects in the dataset or on randomization. Results for different number of clusters are shown in Table 2. The bold number in each row represented in Table 2 indicates the lower distortion score. Finally, it clearly shows that the LOF based centroid initialization performs better than other two algorithms.

4.3. Algorithms for comparison

The proposed algorithm kNAS is compared with three algorithms which consider both structural and attribute similarities.

(i) SA-Cluster-Opt: This is an improved version of SA-Cluster algorithm based on Neumann series which achieves a good balance between structural and attribute similarities through a unified distance measure.
(ii) CODICIL [29]: A simple approach balancing both content and graph topology detects the community based on the signal strength between the two nodes in the network. The inputs given to this algorithm are $k = 50$, the number of the nearest content neighbours for each vertex, $\alpha = 0.5$ parameter that specifies the weights of content and topology similarity, $l = 100$–500 the number of cluster to be formed.
(iii) NISE [36]: Neighborhood Inflated Seed Expansion is an effective overlapping community detection algorithm which is based on Personalized PageRank algorithm (PPR). Each seed is expanded based on the PPR score.
(iv) kNAS: k-Neighborhood Attribute Structural Similarity Algorithm is our proposed algorithm, considers both structural and attributes similarity with very minimal input parameter.

4.4. Evaluation metrics

The quality of the clusters is evaluated based on two measures such as density (D) proposed by Cheng et al., 2011 and Tanimoto Coefficient (TC) [37]. The definitions are as follows.

$$D\left(\left\{V_i\right\}_{i=1}^{m}\right) = \sum_{i=1}^{m} \frac{\left|\left\{\left(v_x, v_y\right) | v_x, v_y \in V_i, \left(v_x, v_y\right) \in E\right\}\right|}{|E|}$$

where $\left\{V_i\right\}_{i=1}^{m}$ is m clusters formed using different algorithms, v_x andv_y are two vertices that belong to the same cluster. Density function represents the density of edges within the cluster with respect to the density of edges of the graph.

$$\mathrm{TC}_{\mathrm{AB}} = \frac{c}{a+b-c}$$

where a denotes the number 1's in attributes of A vertex, b represents the number of 1's in attributes of B vertex and c indicates number of common 1's in attributes of A and B vertex. The Tanimoto coefficient is used to find the similarity between all the vertices with the centroid within the cluster where the value ranges from 0 to 1.

The kNAS algorithm is compared with the baselines that focuses on both structural and node information. In general combining two similarities such as attribute and structural for detecting communities would consume more time for clustering than the algorithm

Table 3
Performance of four algorithms on four datasets. The bolded value denotes the algorithm that performs better than the other algorithms.

Method	Density (D)				Tanimoto Coefficient (TC)				Avg.
	DBLP-1	DBLP-2	Facebook	Twitter	DBLP-1	DBLP-2	Facebook	Twitter	
SA	0.832[a]	0.686[a]	0.640[a]	**0.834**	0.648[a]	0.618[a]	0.743[a]	0.678[a]	0.709[a]
CODICIL	0.453[a]	0.620[a]	0.656[a]	0.587[a]	0.450[a]	0.432[a]	0.563[a]	0.590[a]	0.555[a]
NISE	0.563[a]	0.654[a]	0.593[a]	0.534[a]	0.437[a]	0.529[a]	0.675[a]	0.638[a]	0.577[a]
KNAS	**0.982**	**0.756**	**0.727**	0.678	**1.000**	**1.000**	**1.000**	**1.000**	**0.892**

[a] Represents the KNAS outperforms the other baselines by 95% statistical confidence level.

Table 4
Cluster quality and convergence in DBLP-1 dataset.

Iteration	SA		CODICIL		NISE		kNAS	
	D	TC	D	TC	D	TC	D	TC
1	0.79	0.58	0.79	0.70	0.65	0.56	0.80	1
2	0.80	0.60	0.80	0.68	0.65	0.56	0.84	1
3	0.82	0.65	0.83	0.72	0.66	0.57	0.85	1
4			0.84	0.73	0.66	0.58		

Table 5
cluster quality and convergence in DBLP-2 dataset.

Iteration	SA		CODICIL		NISE		KNAS	
	D	TC	D	TC	D	TC	D	TC
1	0.80	0.52	0.30	0.39	0.40	0.10	0.8	1
2	0.84	0.60	0.35	0.45	0.45	0.18	0.92	1
3	0.85	0.64	0.40	0.45	0.55	0.25	0.91	1
4	0.85	0.65	0.43	0.46	0.65	0.25		
5			0.43	0.46				

that use any one of the similarity. So we considered the baseline algorithm that uses both the similarities for comparing the performance with the kNAS algorithm. The strong performance of kNAS is clearly evident from the above table. It performs better than the state-of-the-art methods which indicate that kNAS combines the best elements from both the sources of data.

We observe that the kNAS achieves higher margin in performance against the existing algorithms in the information network such as DBLP-1and DBLP-2 dataset than the social networks. For example, in DBLP-1dataset kNAS achieves 18% relative gain in Density measure and 54% in the Tanimoto coefficient when compared to the best baseline. For DBLP-2 dataset it attains 10% in density and 61% in tanimoto coefficient measure. The reason behind this phenomenon is that in information network the node attributes plays an important role than the social network. We also note that across all datasets and valuation metrics, kNAS yields the best performance in 7 out of 8 cases. kNAS outperforms SA by 25% CODICIL by 66% and NISE by 54% in terms of average performance. It is inferred that the performance of SA algorithm and kNAS are closer to each other (Table 3).

We also measure the statistical significance of kNAS with the baselines. Statistical significance test is used to validate the performance of the kNAS with all the baselines. Based on the hypothesis testing using one-tail Z test we conclude that the kNAS algorithm outperforms the baselines. The symbol (^{a}) in Table indicates that kNAS outperforms a given baseline by 95% statistical confidence level. The kNAS algorithm outperforms all the baselines except the Twitter database in terms of density of the edges. In this case the SA algorithm has a better performance of 12% than the kNAS algorithm. Since the attribute information tweets are given more importance than the structural relationship in Twitter database, the Tanimoto coefficient value is higher than the density values.

5. Case study on DBLP dataset

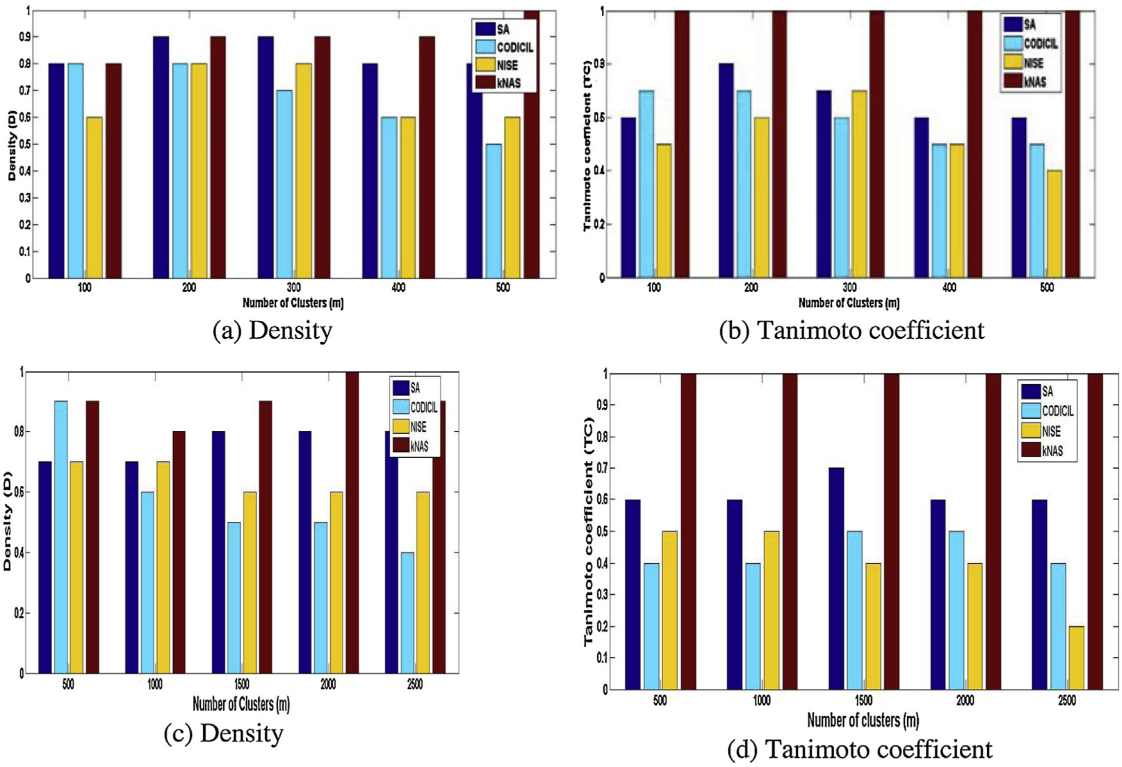
The different values of Density and Tanimoto values using various algorithms on DBLP-1 and DBLP-2 dataset are discussed. Fig. 2a depicts, density comparison among the four algorithm on DBLP-1 dataset when we set the cluster number $m = \{100,200,300,400,500\}$. The density values of SA-cluster and kNAS cluster are close. The density value kNAS increases when number of cluster increases ($m = 500$). On the other hand, CODICIL has a low density and the density of edges gradually decreases as the value of m increases

Fig. 2(b) shows the comparison of Tanimoto coefficient among the four algorithms on DBLP-1 dataset with same set of m values. In general, SA, CODICIL, and NISE algorithm follow a specific technique to balances the structural and attribute similarity. They maintain the coefficient value between 0.5 and 0.7 but the proposed algorithm strictly satisfies the full attribute similarity among the vertices, it has a coefficient value to be 1. The SA algorithm outperforms the CODICIL and NISE in terms of attribute similarity.

Fig. 2(c) and (d) shows the density and tanimoto coefficient on the DBLP-2 dataset with different number of clusters $m = \{500,1000,1500,2000,2500\}$ implemented on clusters formed by SA, CODICIL, NISE and kNAS algorithms. The density of the cluster decreases as the number of cluster increases for CODICIL and density value is around 0.6–0.7 for NISE algorithm. The SA-cluster and kNAS algorithm has proved it works better with higher density even after the increase in dataset size. Since the kNAS strictly enforces the attribute similarity the tanimoto coefficient of the proposed algorithm is always set to 1 with different values of m. In the other algorithms, the similarities are considered based on the weight assigned to the constants. For example, in CODICIL algorithm the constant value is set to $\alpha = 0.5$, which determines the weights for structural and content similarities. The content similarity decreases with increase of m in NISE algorithm. Therefore, it is clear that the proposed kNAS algorithm is scalable and works efficiently than the other state-of-art methods.

5.1. *Clustering convergence*

The quality of the cluster in this algorithm is evaluated for its performance using two measures density and tanimoto coefficients with three algorithms SA, CODICIL and NISE. Tables 4 and 5 show the cluster quality of four algorithms iteration by iteration on DBLP-1 and DBLP-2 dataset. When the number of cluster m value is 30, the SA and kNAS method converges faster in the third iteration, whereas the CODICIL and NISE algorithm converges in the fourth iteration only. For DBLP-2 dataset, we set the m value to be 100 and quality of clusters is calculated for each iteration. The SA cluster and NISE performs the same number of iteration but CODICIL algorithm takes more iteration when the dataset size increases. The proposed algorithm kNAS has proved to converge quickly than all the other three algorithms.



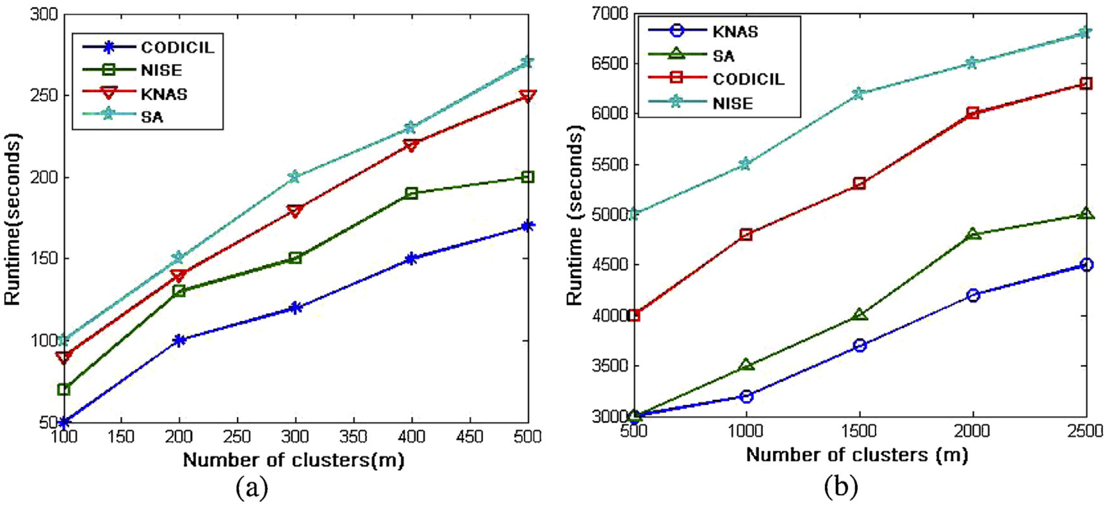
(a) Density

(b) Tanimoto coefficient

(c) Density

(d) Tanimoto coefficient

Fig. 2. (a) & (b) Cluster quality comparison on DBLP-1 dataset. (c) & (d) Denotes cluster quality comparison on DBLP-2 dataset.



(a)

(b)

Fig. 3. (a) Efficiency of four algorithms on DBLP-1 dataset, (b) efficiency of four algorithms on DBLP-2 dataset.

5.2. *Clustering efficiency evaluation*

This experiment compares the efficiency of different clustering algorithm in Fig. 3(a) and (b) on DBLP-1 and DBLP-2 dataset respectively. Fig. 3(a) shows that all methods have less runtime due to its small dataset size. When $m = 500$ for DBLP-1 dataset the SA cluster and kNAS cluster takes more time to execute than the other two algorithms. As SA-cluster has to calculate the random walk distance for each iteration and kNAS algorithm has to detect m centroids using Local Outlier Factor consumes more time that significantly increases the runtime. Fig. 3(b) shows the runtime for DBLP-2 dataset using four algorithms. When the dataset size increases the runtime of CODICIL and NISE algorithm consumes more time than the kNAS algorithm because the centroid and the number of cluster is calculated initially in kNAS which takes less time to group the objects in large dataset. This result proves that the kNAS algorithm is efficient to scalable dataset size and produces quality clusters when compared to other algorithms.

5.3. *Discussion on DBLP dataset*

This section examines the clustering results on DBLP-1 dataset. Table 6 shows only a snapshot of the list of authors from the following areas, Data Base, Data mining and Image processing database due to lack of space. Cluster 1, 2, & 3 contain three group of author who work on “Database System”, “Data mining” and “Image processing”. Some information about the author graph is as follows: The authors who have co-authored and have similar attributes

Table 6
Cluster of authors from DBLP-1 dataset.

Cluster 1 (DB)	Cluster 2 (DM)	Cluster 3 (IP)
David Maier	Srinivasan Parthasarathy	Li Z
Kristin Tufte	Mohammed J.Zaki	Ling F
V.M. Megler	Jing Gao	Chen E
Peter Alvaro	AmolGhoting	Wang Q
Patrick Leyshock	Hui Yang	Zhijian Li
Lois M.L.Delcambre	SitaramAsur	Pirollo S
Joseph M Hellerstein	Jiawei Han	Wen Bilong
Jing Gao	Raghu Machiraju	Ghosh B
Bill Howe	Sameep Mehta	Peng D H
Sudarshan Murthy	Wagner Meira	Zhihong Li
Shawn Bowers	VenuSatuluri	Sugarbaker
Jeffrey D. Ullman	YiyeRuan	Wilson W.K
Jiawei Han	Wei Li	Jie Fang

are grouped into cluster. Sometimes group of clusters have same topic but have never collaborated. For example, author *Srinivasan Parthsarathy* belong to Data mining, but he is also an expertise in Database. Since he has not co-authored with David Maier, it is not included in the cluster1. Author *Jing Gao* is expertise in both database and data mining. Since *Jing Gao* have co-authored with objects in two different clusters, they are present in both the clusters. Similarly, Jawei Han is also overlapped with two different clusters. *Yang Zhou* is an expert in data mining, but he has not co-authored with any of the cluster. Due to the absence of topological similarity, it is not grouped with any of the cluster.

6. Conclusion

In this paper, we have proposed a new approach (kNAS) for overlapping community detection in large scale graph by combining the topological and attribute similarity. The large graph is partitioned into *m* clusters having high intracluster structural similarity and low intercluster similarity. The initial centroids of the cluster are automatically selected based on Local Outlier Factor instead of random selection of centers. Iteratively the centroids updated to the vertex closest to average density of the cluster. The vertices are assigned to cluster that are within the *k*-distance and have similar attributes with the centroid. The structural similarity is based on k-neighbourhood vertex and attribute similarity is based on the Similarity Score. Moreover, two evaluation measures are defined to measure the quality of the cluster formed by four algorithms. Our experiment demonstrates that kNAS algorithm outperforms state-of-art methods in quality and efficiency with respect to varied size datasets.

References

[1] A.K. Jain, M.N. Murty, P.J. Flynn, Data clustering: a review, ACM Comput. Surv. (CSUR) 31 (3) (1999) 264–323.

[2] M. Parimala, D. Lopez, N.C. Senthilkumar, A survey on density based clustering algorithms for mining large spatial databases, Int. J. Adv. Sci. Technol. 31 (1) (2011).

[3] C.C. Aggarwal, H. Wang, A survey of clustering algorithms for graph data, in: Managing and Mining Graph Data, Springer US, 2010, pp. 275–301.

[4] S.E. Schaeffer, Graph clustering, Comput. Sci. Rev. 1 (1) (2007) 27–64.

[5] M. Potamias, F. Bonchi, A. Gionis, G. Kollios, K-nearest neighbors in uncertain graphs, Proc. VLDB Endow. 3 (1–2) (2010) 997–1008.

[6] G. Kollios, M. Potamias, E. Terzi, Clustering large probabilistic graphs, IEEE Trans. Knowl. Data Eng. 25 (2) (2013) 325–336.

[7] V. Satuluri, S. Parthasarathy, Scalable graph clustering using stochastic flows: applications to community discovery, in: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge discovery and Data Mining, June, 2009, 737–746. ACM.

[8] S. van Dongen, Graph clustering by flow simulation, in: Ph.D. Thesis, Dutch National Research Institute for Mathematics and Computer Science, University of Utrecht, Netherlands, 2000.

[9] U. Brandes, M. Gaertler, D. Wagner, Experiments on Graph Clustering Algorithms, Springer Berlin Heidelberg, 2003, pp. 568–579.

[10] L. Wang, T. Lou, J. Tang, J.E. Hopcroft, Detecting community kernels in large social networks, in: 2011 IEEE 11th International Conference on Data Mining (ICDM), December, 2011, 784–793. IEEE.

[11] S. Parthasarathy, Y. Ruan, V. Satuluri, Community discovery in social networks: applications, methods and emerging trends, in: Social Network Data Analytics, Springer US, 2011, pp. 79–113.

[12] F.D. Malliaros, V. Megalooikonomou, C. Faloutsos, Fast Robustness estimation in large social graphs: communities and anomaly detection. SDM 12 (2012) 942–953.

[13] M. Girvan, M.E. Newman, Community structure in social and biological networks, Proc. Natl. Acad. Sci. U. S. A. 99 (12) (2002) 7821–7826.

[14] P. Chen, S. Redner, Community structure of the physical review citation network, J. Informetr. 4 (3) (2010) 278–290.

[15] F. Radicchi, S. Fortunato, A. Vespignani, Citation networks, in: Models of Science Dynamics, Springer Berlin Heidelberg, 2012, pp. 233–257.

[16] J. Leskovec, K.J. Lang, A. Dasgupta, M.W. Mahoney, Community structure in large networks: natural cluster sizes and the absence of large well-defined clusters, Internet Math. 6 (1) (2009) 29–123.

[17] M.E. Newman, Fast algorithm for detecting community structure in networks, Phys. Rev. E 69 (6) (2004) 066133.

[18] A. Clauset, M.E. Newman, C. Moore, Finding community structure in very large networks, Phys. Rev. E 70 (6) (2004) 066111.

[19] E.A. Leicht, M.E. Newman, Community structure in directed networks, Phys. Rev. Lett. 100 (11) (2008) 118703.

[20] Y. Kim, S.W. Son, H. Jeong, Finding communities in directed networks, Phys. Rev. E 81 (1) (2010) 016103.

[21] S. Fortunato, Community detection in graphs, Phys. Rep. 486 (3) (2010) 75–174.

[22] M.E. Newman, M. Girvan, Finding and evaluating community structure in networks, Phys. Rev. E 69 (2) (2004) 026113.

[23] M.E. Newman, Modularity and community structure in networks, Proc. Natl. Acad. Sci. U. S. A. 103 (23) (2006) 8577–8582.

[24] E. Keogh, S. Lonardi, C.A. Ratanamahatana, Towards parameter-free data mining. Proceedings of the tenth ACM SIGKDD, in: International Conference on Knowledge Discovery and Data Mining, Aufust, 2004, 206–215. ACM.

[25] D. Sardana, R. Bhatnagar, Graph Clustering using mutual k-nearest neighbors, in: Active Media Technology, Springer International Publishing, 2014, pp. 35–48.

[26] C. Xiong, D. Johnson, J.J. Corso, Spectral active clustering via purification of the k-nearest neighbor graph, Proc. of European Conference on Data Mining (2012).

[27] W. Lin, X. Kong, P.S. Yu, Q. Wu, Y. Jia, C. Li, Community detection in incomplete information networks, in: Proceedings of the 21st International Conference on World Wide Web, April, 2012, 341–350. ACM.

[28] L. Galluccio, O. Michel, P. Comon, A.O. Hero, Graph based k-means clustering, Signal Process. 92 (9) (2012) 1970–1984.

[29] Y. Ruan, D. Fuhry, S. Parthasarathy, Efficient community detection in large networks using content and links, in: Proceedings of the 22nd International Conference on World Wide Web, May, 2013, 1089–1098. International World Wide Web Conferences Steering Committee.

[30] H. Cheng, Y. Zhou, J.X. Yu, Clustering large attributed graphs: a balance between structural and attribute similarities, ACM Trans. Knowl. Discov. Data (TKDD) 5 (2) (2011) 12.

[31] X. Xu, N. Yuruk, Z. Feng, T.A. Schweiger, Scan: a structural clustering algorithm for networks, in: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, August, 2007, 824–833. ACM.

[32] G.H. Ball, D.J. Hall, PROMENADE-an outline pattern recognition system, Stanford Research Institute, Stanford University, 1967.

[33] D. Arthur, S. Vassilvitskii, k-means++: the advantages of careful seeding, in: Proceedings of the Eighteenth Annual ACM-SIAM Symposium Discrete Algorithms, New Orleans, Louisiana, 2007, 2007, pp. 1027–1035.

[34] M.M. Breunig, H. Kriegel, R.T. Ng, J. Sander, LOF: identifying density-based local outliers, in: Proceedings of 2000 ACM-SIGMOD International Conference of Data, Dallas, Texas, 2000, 2000, pp. 93–104.

[35] V. Chaoji, M. Al Hasan, S. Salem, M.J. Zaki, SPARCL: an effective and efficient algorithm for mining arbitrary shape-based clusters, Knowl. Inf. Syst. 21 (2) (2009) 201–229.

[36] J.J., Whang, D.F. Gleich, I.S. Dhillon, Overlapping Community Detection Using Neighborhood-Inflated Seed Expansion (2015). arXiv preprint arXiv:1503.07439.

[37] Alan H. Lipkus, A proof of the triangle inequality for the Tanimoto distance, J. Math. Chem. 26 (1–3) (1999) 263–265.