Modeling *Over-dispersion* for Network Data Clustering

Lu Wang[†], Dongxiao Zhu* [†], Yan Li[‡] and Ming Dong[†]
[†]Dept. of Computer Science, Wayne State University, Detroit, MI 48202. Email: {lu.wang3, dzhu, mdong}@wayne.edu
[‡]Dept. of Computational Medicine and Bioinformatics, University of Michigan, Ann Arbor, MI 48109.
Email: yanliwl@umich.edu

***Abstract*—*Over-dispersed* network data mining has emerged as a central theme in data science, evident by a sharp increase in the volume of real-world network data with imbalanced clusters. While most of existing clustering methods are designed for discovering the number of clusters and class specific connectivity patterns, few methods are available to uncover the imbalanced clusters, commonly existing in network communities and image segments, from network data with over-dispersed cluster size distribution. The latter is considered as an intrinsic structural property of the network data. In this paper, we propose a generalized probabilistic modeling framework, SizeConnectivity, to estimate over-dispersed cluster size distribution together with class specific connectivity patterns from network data. A wide range of cluster size distributions revealed by real-world network data can be accurately captured by our method. We performed extensive synthetic and real-world experiments on clustering social network data and image data for detecting network communities and image segments. Our results demonstrate a superior performance of our SizeConnectivity clustering method in recovering the hidden structure of network data via modeling *over-dispersion*.**

***Index Terms*—Over-dispersion, network data, clustering, cluster size distribution, network communities, image segments.**

I. Introduction

The last few years have witnessed an explosive increase of network data volume, variety and veracity as it naturally describes the structured connections among objects. Formally, objects refer to *nodes* and connections refer to *edges* between nodes. There is a need to uncover the hidden structure of the network data in a number of data rich domains, such as social science, image processing, business analysis, information retrieval and bioinformatics [1], [2], [3], [4]. To deal with this important and interesting problem, a lot of network data clustering methods [5], [6], [7] have been designed aiming at grouping the nodes with a similar connectivity pattern in the same cluster.

Intuitively, clusters in network data not only differ in their connectivity patterns, but also can differ dramatically in their sizes. Unfortunately, the distribution of network cluster sizes, particularly *over-dispersed* with high variance, remains a less attended issue in network data mining. *Over-dispersion* arises when the data exhibits larger variance than the variance permitted by the assumed model, also known as extra variation. It exists in data from many different research areas including sociology, economics, ecology and biology [8]. Standard network clustering methods, such as spectral clustering [5] and model-based probabilistic clustering [9], albeit effective, are not designed to uncover over-dispersed network clusters. Thus, new modeling framework considering both over-dispersed cluster size distribution and connectivity pattern is urgently needed.

To further motivate our work, let us briefly discuss two exemplar applications in clustering network data: social network community detection and image segmentation. In social network community detection problem, network is partitioned into many modules of subnetworks (communities) and the cluster sizes are commonly over-dispersed. For example, college football teams in the USA and their games, considered as network data where nodes represent football teams and edges exist between pairs of football teams in competition games. The division sizes of college football teams corresponding to cluster sizes are often over-dispersed.

Image segmentation aims at finding objects that are commonly constructed via adjacent pixels with a similar grey level. In network based image segmentation methods, pixels are treated as nodes and edges exist when the dissimilarity among pixels are less than a specific threshold. As the size and shape of each object within the image are different from each other, e.g., a rabbit and a house, the object sizes are over-dispersed that is very common in the real-world image data. However, the standard network clustering methods intend to divide the majority group into several subgroups, e.g., hierarchical clustering [10].

Here we propose a novel clustering approach for detecting imbalanced network clusters by explicitly modeling the over-dispersion. Our proposed method employs a model-based probabilistic clustering approach since it naturally captures geometric property and overall structural information of the network data. In addition, unlike some commonly used network clustering methods such as spectral clustering [5] and hierarchical clustering [11], the location and shape of data and cluster sizes information can be efficiently encoded in the model-based probabilistic clustering methods [12].

In our model, we use Poisson distribution to accommodate the imbalanced cluster sizes revealed in real-world network data. Compared to other discrete probability distributions, Poisson distribution is an asymmetric distribution with heavy right tail; thus, it is more suitable for accommodating over-dispersed cluster size than others, e.g., Laplace and negative

*Corresponding Author

binomial. Laplace distribution shares the similar core function with normal distribution in their probability density function [13], which limits its capability of accommodating over-dispersion. Negative binomial distribution works well for the data with excessive zero counts (zero-inflated property) [14] but it is not the case for many network data sets.

Our contribution is to model over-dispersed cluster size distribution as an independent component from the class specific connectivity for network data clustering. Using the class indicator as a latent variable, we derive and maximize a new likelihood function of our model-based probabilistic clustering model, denoted as SizeConnectivity Generalized (SCG) model, to simultaneously estimate imbalanced cluster sizes and class specific connectivity pattern. We present extensive synthetic and real-world examples from social communities and image segmentation to show the ubiquity of over-dispersion as well as the versatility of the method we proposed. The advantages of SizeConnectivity framework over the conventional ConnectivityOnly framework, which only considers class specific connectivity pattern, are demonstrated in Figure 1. It clearly shows that the conventional ConnectivityOnly framework does not segment the image correctly in the over-dispersed scenario shown as the first two rows of the 3^{rd} column in Figure 1.

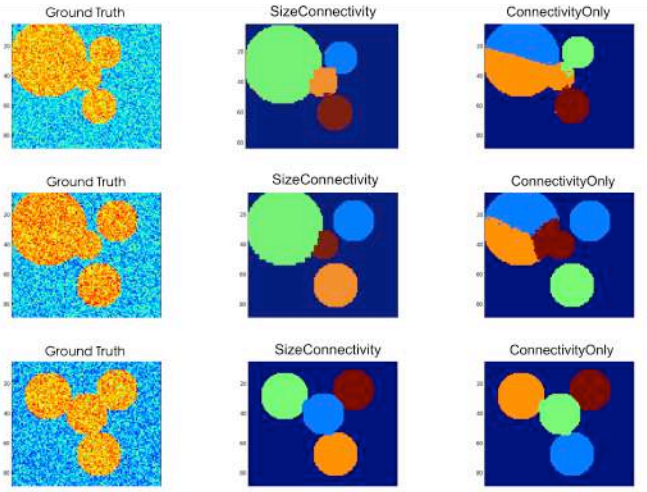

Fig. 1: The conventional ConnectivityOnly approach (3^{rd} column) does not accurately detect clusters with *over-dispersed* (high variance) cluster size distribution whereas the proposed SizeConnectivity approach (2^{nd} column) does. 1^{st} column represents the original input images (ground truth). The four true circles are shown in different colors and the corresponding segments are calculated from the input images using SizeConnectivity and ConnectivityOnly approaches, respectively. The conventional ConnectivityOnly approach, i.e., spectral clustering with normalized cuts, was implemented in the *sklearn* module of Python package *Scikit-learn* [15].

The rest of the paper is organized as follows. In Section II, we review the related works in network data clustering. In Section III, we propose our new SizeConnectivity clustering framework with technique details. In Section IV, we present experimental results using both synthetic and real-world network data and compare with the selected clustering methods for network data. Finally, we conclude with discussion in Section V.

II. Related Work

Clustering approaches have been extensively used in analyzing network data for discovering the nodes and connectivity patterns among the nodes within a cluster or across different clusters [7]. Specifically, many clustering methods use pairwise dissimilarities, e.g., Euclidean distance, between the nodes, which are broadly divided into partitional clustering and hierarchical clustering [7].

A fundamental partitional clustering algorithm is K-means [7]. It is efficient and effective but its performance heavily depends on the initialization and is sensitive to the outlier nodes as well. To overcome these limitations, another type of partitional clustering algorithms, known as connectivity-based spectral clustering, has been proposed, e.g., in [5].

Normalized cuts [5], a well-known spectral clustering method, attempts to optimize the cost functions with partitioning all the nodes connected by weighted pair-wise similarity edges to find more clusters. In this method, a metric was proposed to avoid clustering a single outlier into one cluster that is highly effective for image segmentation [5]. In recent years, spectral methods for community detection and graph partitioning via maximizing modularity and likelihood have been developed based on the eigenvectors of the so-called normalized Laplacian matrix [16].

Single-link and complete-link are among the well-known hierarchical clustering algorithms, which recursively search nested clusters either in agglomerative or divisive mode [6]. They are useful to represent positions in network data while their criteria of deciding how many clusters are often arbitrary without providing an unique solution [17]. Moreover, they do not work well when there is no valid distance measure can be used especially in the network data with unweighted edges. These algorithms are based on a greedy procedure which only consider the local neighbors at each step; thus, they ignore the global shape and size of clusters [6] in the network data.

Different from the partitional and hierarchical clustering approaches, the model-based probabilistic clustering methods are capable of drawing a global picture of the network structure with capturing its geometric property [18]. The classical model-based probabilistic clustering method is built based on finite mixture model [19], denoted as ConnectivityOnly method in Appendix, which models different connectivity patterns in network data. However, using classical finite mixture models for capturing over-dispersed cluster size distribution may represent a significant misrepresentation of the intrinsic structure of the network data since cluster size distribution in real-world data is frequently imbalanced and heavily deviated from a normal distribution.

Besides connectivity patterns, extended finite mixture models may implicitly capture normally distributed cluster sizes

by using multinomial distribution as we described it in Supplementary file. The latter asymptotically converges to normal distribution [20]. More recent model-based clustering methods [12] for network data to detect overlapping communities have been proposed as extensions to the classical finite mixture model. Nevertheless, these methods implicitly use multinomial distribution for cluster sizes hence are incapable of accommodating over-dispersed cluster size distribution.

Therefore, the extended finite mixture models are not specifically designed for modeling over-dispersion that widely exists in network clusters. In other words, these approaches may work well for detecting the symmetric normally distributed network clusters but not the asymmetric over-dispersed ones. Due to the lack of clustering methods for detecting imbalanced clusters from network data, new methods for uncovering the intrinsic structure of network data accommodating features inherent in over-dispersed cluster size distribution are desirable and urgently needed.

III. Methodology

In this section, we describe the proposed SizeConnectivity Generalized (SCG) model in detail. In the Appendix, we will also present a most commonly used connectivity-based network data clustering method that is denoted as ConnectivityOnly model (COM), a multinomial connectivity-based mixture model (MCM) and two SizeOnly models, which use Poisson mixture model (PMM) and multinomial mixture model (MMM) to model cluster size distribution without capturing connectivity patterns.

As stated in Section I, besides the connectivity information, the over-dispersed cluster size distribution can also contain valuable information for improving clustering performance. Over-dispersed cluster size distribution, representing an intrinsic structure of network data, is often of practical interest together with cluster connectivity. Hence, we develop a novel SizeConnectivity Generalized (SCG) model for clustering network data considering both class specific connectivity and over-dispersion in cluster size distribution. We note that SCG is one-of-a-kind probabilistic modeling approach to integrate both cluster connectivity and over-dispersion for clustering the network data. Here we choose Poisson distribution to model over-dispersion in cluster sizes since it is a non-symmetric discrete distribution. Assuming there are K clusters and the k^{th} cluster has n_k nodes, then its corresponding probability has the form:

$$p(n_k|\lambda_k) = \frac{\lambda_k^{n_k} e^{-\lambda_k}}{n_k!}, \tag{1}$$

where λ_k is a parameter representing the size of the k^{th} cluster.

The probability of a link, denoted as θ_{kj}, indicates there is an edge from a particular node in a certain k^{th} cluster to a node j. Therefore, the probability that a node i belongs to the k^{th} cluster can be calculated as:

$$p(X_{(i,:)}, Z_{ik} = 1|\theta) = \prod_{i,j=1}^{n} \theta_{kj}^{X_{ij}}, \tag{2}$$

where $X_{(i,:)}$ is the i^{th} row of adjacency matrix, $X_{ij} = 1$ when there is an edge from node i to node j, otherwise $X_{ij} = 0$. And the latent variable Z_{ik} is used as an indicator to represent whether the node i belongs to the k^{th} cluster ($Z_{ik} = 1$) or not ($Z_{ik} = 0$). Therefore, the likelihood function of SCG model can be written as:

$$L_{\mathrm{SCG}}(X, |\phi_k, \theta, \lambda_k) = \prod_{i=1}^{n}\prod_{k=1}^{K}\left(\frac{\lambda_k^{n_k} e^{-\lambda_k}}{n_k!}\phi_k \prod_{j=1}^{n}\theta_{kj}^{X_{ij}}\right)^{Z_{ik}}, \tag{3}$$

where $\phi_k = \frac{n_k}{n}$ denotes the probability of a random node belongs to the k^{th} cluster.

We employ EM algorithm to estimate the parameters of SCG model. At the l^{th} iteration, the E-step has the form as:

$$\tau_{ik}^{(l-1)} = \frac{p(n_k^{(l-1)}|\lambda_k^{(l-1)})\phi_k^{(l-1)} p(X_{(i,:)}, Z_{ik} = 1|\theta^{(l-1)})}{\sum_{k'=1}^{K} p(n_{k'}^{(l-1)}|\lambda_{k'}^{(l-1)})\phi_{k'}^{(l-1)} p(X_{(i,:)}, Z_{ik'} = 1|\theta^{(l-1)})}, \tag{4}$$

where $p(n_k^{(l-1)}|\lambda_k^{(l-1)})$ and $p(X_{(i,:)}, Z_{ik} = 1|\theta^{(l-1)})$ are the estimation of probability of the k^{th} cluster with n_k nodes and the probability of node i belongs to the k^{th} cluster at the $(l-1)^{th}$ iteration, calculated based on Eq. (1) and Eq. (2), respectively.

In the M-step, we find the parameter values that maximize the $Q(\Phi, \Phi^{(l-1)})$

$$Q(\Phi, \Phi^{(l-1)}) = \sum_{i=1}^{n}\sum_{k=1}^{K}\tau_{ik}^{(l-1)}\left(\log\frac{(\lambda_k^{(l-1)})^{n_k^{(l-1)}} e^{-\lambda_k^{(l-1)}}}{n_k^{(l-1)}!} + \log\phi_k^{(l-1)} + \sum_{j=1}^{n} X_{ij}\log\theta_{kj}^{(l-1)}\right), \tag{5}$$

where Φ denotes a complete set of SCG related parameters.

λ_k is estimated by setting the partial derivative of $Q(\Phi, \Phi^{(l-1)})$ to 0, which is mathematically represented as:

$$\frac{\partial Q(\Phi, \Phi^{(l-1)})}{\partial \lambda_k} = 0. \tag{6}$$

So that we have:

$$\lambda_k^{(l)} = n_k^{(l)} = \sum_{i=1}^{n} Z_{ik}^{(l)}, \tag{7}$$

where $Z_{ik}^{(l)} = \mathbb{I}\left(\tau_{ik}^{(l)} = \max_{k'=\{1,\ldots,K\}} \tau_{ik'}^{(l)}\right)$ and $\mathbb{I}(\cdot)$ is the indicator function.

ϕ_k is the cluster weight parameter of the k^{th} cluster, which is updated by summarizing the expected counts of nodes as:

$$\phi_k^{(l)} = \sum_{i=1}^{n} \frac{\tau_{ik}^{(l-1)}}{n}. \tag{8}$$

θ_{kj} is the probability that there is an edge between node j and a particular node in the k^{th} cluster initialized with random number between $[0, 1]$ and updated as follows:

$$\theta_{kj}^{(l)} = \frac{\sum_{i=1}^{n} X_{ij}\tau_{ik}^{(l-1)}}{\sum_{i=1}^{n} X_{i\cdot}\tau_{ik}^{(l-1)}}, \tag{9}$$

Algorithm 1: The SizeConnectivity Generalized (SCG) algorithm

Input: The adjacency matrix of network data X, Number of clusters K, l=1

for $k = 1$ *to* K **do**
  Initialize: $\phi_k^{(0)} = \frac{1}{K}$, $\lambda_k^{(0)} = n_k^{(0)} = \frac{n}{K}$, and randomly assign $\theta_{kj}^{(0)}$ in $[0, 1]$;
end
repeat
  E-step: Compute the responsibilities $\tau_{ik}^{(l-1)} = \frac{p(n_k^{(l-1)}|\lambda_k^{(l-1)})\phi_k^{(l-1)}p(X_{(i,:)}, Z_{ik}=1|\theta^{(l-1)})}{\sum_{k'=1}^{K} p(n_{k'}^{(l-1)}|\lambda_{k'}^{(l-1)})\phi_{k'}^{(l-1)}p(X_{(i,:)}, Z_{ik'}=1|\theta^{(l-1)})}$ at the l^{th} iteration;
  M-step: Update the corresponding parameters $\phi_k^{(l)} = \sum_{i=1}^{n} \frac{\tau_{ik}^{(l-1)}}{n}$ by Eq. (8), $\lambda_k^{(l)} = n_k^{(l)} = \sum_{i=1}^{n} Z_{ik}^{(l)}$ by Eq. (7), $\theta_{kj}^{(l)} = \frac{\sum_{i=1}^{n} X_{ij}\tau_{ik}^{(l-1)}}{\sum_{i=1}^{n} X_{i\cdot}\tau_{ik}^{(l-1)}}$ by Eq. (9);
  $l = l + 1$;
until $|\tau^{(l+1)} - \tau^{(l)}| < \epsilon$;

where $X_{i\cdot} = \sum_{j=1}^{n} X_{ij}$ is the degree of node i.

The complete algorithm for solving SizeConnectivity Generalized (SCG) model is given in Algorithm 1. At the beginning of the algorithm, each cluster of network data is given with equal size, and each cluster is given with equal weight. That is, $\lambda_k^{(0)} = n_k^{(0)} = \frac{n}{K}$ for cluster size and $\phi_k^{(0)} = \frac{1}{K}$ for the cluster weight parameter. We also randomly assign a value between 0 and 1 to $\theta_{kj}^{(0)}$.

The E and M steps alternates until convergence. Then we assign each node to a cluster with the highest probability among all clusters according to the indicator Z_{ik}, calculated as follows:

$$p(Z_{ik} = 1|X, \hat{\Phi}) = \frac{p(\hat{n}_k|\hat{\lambda}_k)\hat{\phi}_k p(X_{(i,:)}, Z_{ik} = 1|\hat{\theta})}{\sum_{k'=1}^{K} p(\hat{n}_{k'}|\hat{\lambda}_{k'})\hat{\phi}_{k'} p(X_{(i,:)}, Z_{ik'} = 1|\hat{\theta})},$$

where $\hat{\Phi} = \{\hat{n}_k, \hat{\lambda}_k, \hat{\phi}_k, \hat{\theta}\}$, is a set of estimation of parameters for SCG model after convergence of learning process.

IV. Experiments and Results

In this section, we validate and evaluate our proposed clustering method by comparing with several other methods using a total of nine data sets, including four synthetic network data sets, three synthetic images and two real-world social network data sets.

A. Experimental Setup

We compared the clustering performance of our proposed SizeConnectivity Generalized (SCG) model to Connectivity-Only model (COM), multinomial connectivity-based mixture model (MCM), Poisson mixture model (PMM), multinomial mixture model (MMM), and an ensemble of other seven selected clustering methods, i.e., K-means, MiniBatch K-means, Spectral Clustering with K-means approach (SC-K), Spectral Clustering with discretization approach (SC-D)[1], Hierarchical Clustering with Ward linkage (HC-W), Hierarchical Clustering with average linkage (HC-A), and Hierarchical Clustering with complete linkage (HC-C). We implemented SCG, COM, MCM, PMM and MMM methods by using Python language based on the following open-source packages such as *NumPy* [21], *SciPy* [22] and *matplotlib* [23]. The other seven clustering methods were implemented in Python machine learning package *Scikit-learn* [15].

Since each clustering algorithm has its own heuristic nature and final clustering results may be different due to different initialization of related parameters, we ran each algorithm multiple times using different initial parameter values attempting to report their best performances. We ran the algorithms implemented in Python machine learning package *Scikit-learn* and our algorithms ten times on four synthetic network data sets, three synthetic images and two real-world social network data sets due to the parameter adjustments. Specifically, we used a different centroid seed each time when we ran the K-means type of algorithms.We tried different numbers of connected components in connectivity matrix when running Hierarchical Clustering type of algorithms. We also used different degrees of polynomial kernels for running Spectral Clustering type of algorithms. We tried different batch sizes for MiniBatch K-means as well.

B. Experiments on Synthetic Network Data

We designed a set of experiments using synthetic network data to evaluate the performance of our SizeConnectivity Generalized (SCG) algorithm in uncovering various cluster size distributions.

We generated four synthetic network data sets, named Syn1, Syn2, Syn3 and Syn4, with different cluster size distributions using R package *statnet* [24]. In each synthetic data set, the number of nodes and the number of clusters are set to be 105 and 5, respectively. The cluster size distribution in Syn1 and Syn2 are more uniform (low variance) while that in Syn3 and Syn4 are over-dispersed (high variance). We used a popular open-source visualization and exploration software *Gephi* to visualize the network data in Graph Modeling Language (GML) format [25]. Figure 2 shows the cluster size distributions of four synthetic network data sets with two panels, which upper one is the histogram of the true size for each cluster and the lower one is the actual network plot to show the distribution of cluster sizes.

We used Adjusted Rand Index (ARI) as the evaluation metric, when the ground truth of the data is available [26]. Let s and d denote as the number of pairs of nodes that are in the same cluster in both ground truth and clustering result and the number of pairs of nodes that are in the different clusters in both ground truth and clustering result, respectively. Thus, we have the Rand Index (RI)= $\frac{s+d}{t}$, where t is the total

[1]Both spectral clustering methods have employed normalized Laplacian to find normalized cuts. K-means and discretization are two ways of assigning labels after the Laplacian embedding [15].

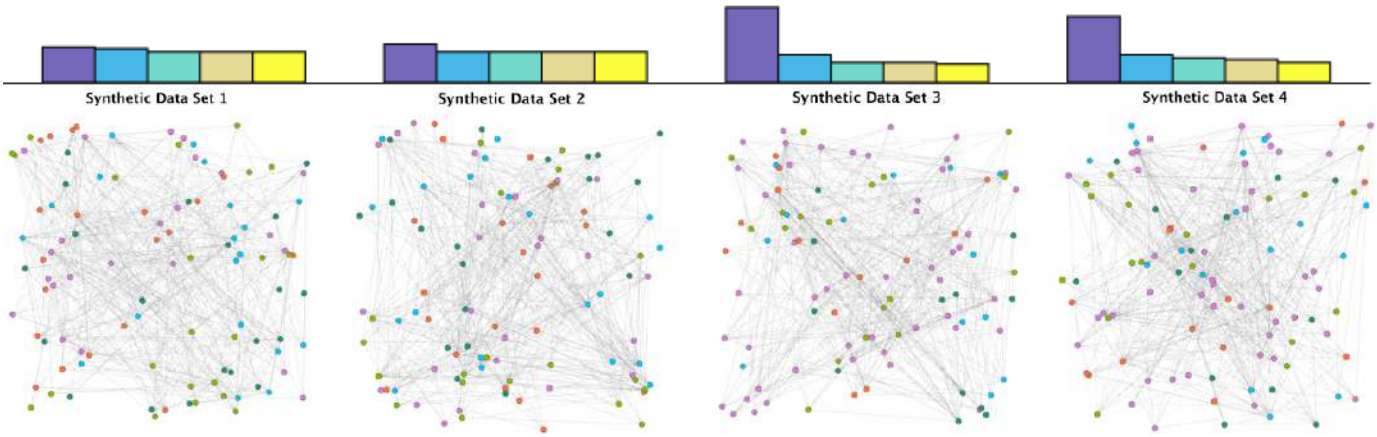

Fig. 2: Cluster size distributions of four synthetic social network data sets. Upper panel is the histogram of the true size for each cluster; lower panel is the actual network plot to show the distribution of cluster sizes.

number of possible paris in the data set. Then we get the ARI= $\frac{RI-E[RI]}{max(RI)-E[RI]}$, where $E[RI]$ is the expected RI.

According to the ARI values in Table I, our proposed SCG algorithm outperforms other methods in synthetic network data set Syn3 and Syn4, but not in Syn1 and Syn2. This result highlights the key advantage of our SCG method in modeling over-dispersion for improving clustering performance whereas the conventional methods don't have.

TABLE I: The Adjusted Rand index (ARI) of the 12 selected clustering methods: SizeConnectivity Generalized (SCG) model comparing to ConnectivityOnly model (COM), multinomial connectivity-based mixture model (MCM), Poisson mixture model (PMM), multinomial mixture model (MMM), K-means, MiniBatch K-means (MB-K), Spectral Clustering with K-means approach (SC-K), Spectral Clustering with discretization approach (SC-D), Hierarchical Clustering with Ward linkage (HC-W), Hierarchical Clustering with average linkage (HC-A) and Hierarchical Clustering with complete linkage (HC-C) using four synthetic social network data sets and two real-world data sets. The best performance results are bold faced.

Methods	Syn1	Syn2	Syn3	Syn4	Polbooks	Football
SCG	0.72	0.74	**0.83**	**0.80**	**0.87**	**0.78**
COM	0.48	0.56	0.50	0.50	0.56	0.57
MCM	0.48	0.56	0.51	0.53	0.56	0.55
PMM	0.39	0.51	0.51	0.47	0.38	0.46
MMM	0.51	0.50	0.33	0.32	0.38	0.37
K-means	0.67	0.67	0.68	0.69	0.67	0.64
MB-K	0.67	0.68	0.70	0.71	0.66	0.62
SC-K	0.68	0.73	0.77	0.76	0.77	0.74
SC-D	**0.75**	**0.76**	0.74	0.71	0.75	0.71
HC-W	0.68	0.67	0.71	0.70	0.76	0.72
HC-A	0.58	0.61	0.70	0.72	0.74	0.69
HC-C	0.62	0.64	0.74	0.73	0.75	0.70

To further demonstrate the key advantage of modeling over-dispersion in network data, we then generated three synthetic images with four circles representing four clusters as image network data, using the module *sklearn* of Python package *Scikit-learn* [15], shown in Figure 1 in Section I. Note that the over-dispersion of the four circles' sizes exist in the first two images but not the third one as comparison.

As mentioned above, in images, pixels are treated as nodes and edges exist when the grey level dissimilarity among pixels are less than a specific threshold. In our experiments, we set the threshold of the three synthetic images as 10%, i.e., assuming p_i and p_j are the grey values for two pixels i and j, if $\frac{|p_i-p_j|}{255} \leq 10\%$, we define there is an edge between two pixels i and j.

We presented the clustering results for this synthetic image data in Figure 1 to demonstrate our motivation for modeling over-dispersion in network data. To support the key advantage of modeling over-dispersion by visualizing the image segmentation results, we computed ARI values for our SCG clustering algorithm and the conventional spectral clustering algorithm SC-K in Table II.

Our SCG algorithm implementing the SizeConnectivity approach outperforms the SC-K approach among all three synthetic images, especially for the first two images with over-dispersion. Note ARI values of the SCG and SC-K algorithms are very close in the third image due to its four circles' sizes are more uniform. In conclusion, the experimental results from both synthetic social and image network data sets demonstrate that our SCG algorithm perform better via modeling over-dispersion in image segmentation.

TABLE II: ARI values of image segmentation using SCG algorithm (SizeConnectivity) and SC-K algorithm (ConnectivityOnly). The best performance results are bold faced.

Image No.	SCG Clustering	SC-K Clustering
1	**0.81**	0.51
2	**0.85**	0.43
3	**0.89**	0.85

C. Experiments on Real-world Network Data

The two real-world network data sets Polbooks and Football, categorized as social network data, were downloaded

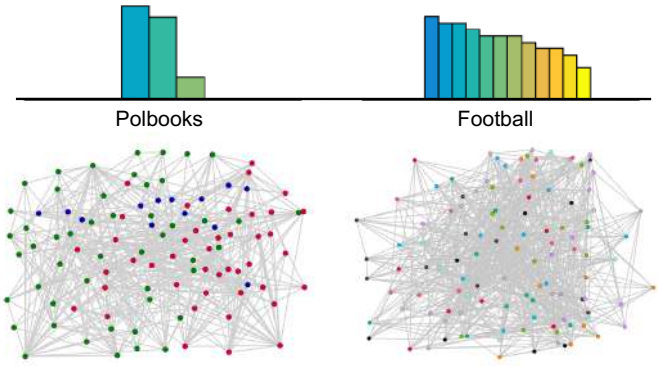

Fig. 3: Community size distributions of two real-world social network data sets: Polbooks and Football.

from University of Michigan network data webpage[2]. Please refer to Figure 3 to see the true community size distribution for Polbooks and Football. The data set Polbooks contains the network of books about US politics with 105 nodes representing books about US politics sold by the online bookseller Amazon.com. Given values as “l", “n", or “c", they are labeled as “liberal", “neutral", or “conservative" respectively. And edges represent frequent co-purchasing of books by the same buyers, as indicated by the “customers who bought this book also bought these other books" feature on Amazon [27]. Football contains the network of American football games between Division IA colleges during regular fall season. There are 115 nodes representing 115 football teams and 12 clusters which are the 12 conferences, and edges connect any pair of football teams with any competition [16].

The ARI results of these two real-world network data sets are shown as last two rows of Table I. Specifically, SCG outperforms in the network data Polbooks due to SCG performs better within dispersed community size distribution comparing with ConnectivityOnly and SizeOnly approaches. In this data set, the majority communities are “liberal" and “conservative" represented by red and green dots, while “neutral" represented by blue dots is much less than the other two communities shown in Figure 4. As a result, all the three approaches can detect most of the major two communities via community connectivity or size. However, we can see that the “neutral" community can not be detected correctly due to its community size is much more unlike the other two communities, only our proposed SizeConnectivity approach can detect the most of books in “neutral" community comparing with the ConnectivityOnly and SizeOnly approaches. To further indicate the difference of three methods' performance, Table III shows the accuracy of clustering result for each cluster.

V. Conclusion

In this paper, a novel SizeConnectivity Generalized (SCG) model is presented to solve clustering problems for network data. Our proposed SCG algorithm integrates cluster (community or segment) connectivity with over-dispersed cluster size distribution in one generalized model. We compared our proposed SCG model with ConnectivityOnly model using pure probabilistic counting mixture model (COM), SizeOnly model using cluster size information with mixing proportion mixture model (PMM and MMM) along with other partitional and hierarchical clustering approaches. From clustering results of real-world and synthetic network data, it is obvious that the proposed SCG outperforms other connectivity-based clustering approaches.

[2]http://www-personal.umich.edu/~mejn/netdata/

TABLE III: Accuracy of clustering result of Polbooks network data set for each cluster using SCG algorithm (SizeConnectivity), ConnectivityOnly model (COM) and SizeOnly model. The best performance results are bold faced.

Color/Community	SCG	COM	SizeOnly
Liberal	**0.98**	0.91	0.84
Conservative	**0.98**	0.85	0.77
Neutral	**0.73**	0.25	0.14

Albeit the joint probability model was presented in the context of unsupervised network data clustering, it is sufficiently flexible to be extended to solving supervised network classification problems. Moreover, we can extend our SizeConnectivity approach to SizeDensity approach, which employs appropriate density distributions to detect imbalanced clusters in alphabet, continuous and categorical data.

Acknowledgments

This paper is based upon work supported by the National Science Foundation under Grant No. 1637312 and 1451316.

References

[1] J. Han, M. Kamber, and J. Pei, *Data mining: concepts and techniques: concepts and techniques*. Elsevier, 2011.

[2] G. Bordogna, L. Frigerio, A. Cuzzocrea, and G. Psaila, “An effective and efficient similarity-matrix-based algorithm for clustering big mobile social data,” in *Machine Learning and Applications (ICMLA), 2016 15th IEEE International Conference on*. IEEE, 2016, pp. 514–521.

[3] L. R. Acharya, T. Judeh, G. Wang, and D. Zhu, “Optimal structural inference of signaling pathways from unordered and overlapping gene sets,” *Bioinformatics*, vol. 28, no. 4, pp. 546–556, 2011.

[4] L. Acharya, T. Judeh, Z. Duan, M. Rabbat, and D. Zhu, “Gsgs: a computational approach to reconstruct signaling pathway structures from gene sets,” *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, vol. 9, no. 2, pp. 438–450, 2012.

[5] J. Shi and J. Malik, “Normalized cuts and image segmentation,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 8, pp. 888–905, 2000.

[6] M.-S. Yang and K.-L. Wu, “A similarity-based robust clustering method,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 26, no. 4, pp. 434–448, 2004.

[7] A. K. Jain, “Data clustering: 50 years beyond k-means,” *Pattern recognition letters*, vol. 31, no. 8, pp. 651–666, 2010.

[8] J. G. Morel and N. Neerchal, *Overdispersion models in SAS*. SAS Institute, 2012.

[9] M. E. Newman and E. A. Leicht, “Mixture models and exploratory analysis in networks,” *Proceedings of the National Academy of Sciences*, vol. 104, no. 23, pp. 9564–9569, 2007.

[10] B. H. Good, Y.-A. de Montjoye, and A. Clauset, “Performance of modularity maximization in practical contexts,” *Physical Review E*, vol. 81, no. 4, p. 046106, 2010.

[11] S. C. Johnson, “Hierarchical clustering schemes,” *Psychometrika*, vol. 32, no. 3, pp. 241–254, 1967.

[12] J. Xie, S. Kelley, and B. K. Szymanski, “Overlapping community detection in networks: The state-of-the-art and comparative study,” *ACM Computing Surveys (csur)*, vol. 45, no. 4, p. 43, 2013.

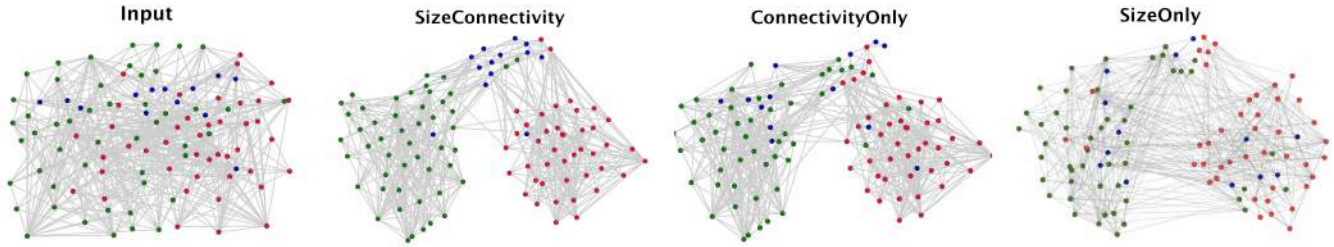

Fig. 4: Community detection using Polbooks network data set. Our proposed SizeConnectivity Generalized (SCG) model is capable of accurately detecting network communities with *over-dispersed* community size distribution whereas the ConnectivityOnly model (COM) and SizeOnly model are not.

[13] D. F. Andrews and C. L. Mallows, "Scale mixtures of normal distributions," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 99–102, 1974.

[14] J. M. Hilbe, *Negative binomial regression*. Cambridge University Press, 2011.

[15] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[16] M. Girvan and M. E. Newman, "Community structure in social and biological networks," *Proceedings of the national academy of sciences*, vol. 99, no. 12, pp. 7821–7826, 2002.

[17] S. Wasserman and K. Faust, *Social network analysis: Methods and applications*. Cambridge university press, 1994, vol. 8.

[18] A. Goldenberg, A. X. Zheng, S. E. Fienberg, and E. M. Airoldi, "A survey of statistical network models," *Foundations and Trends® in Machine Learning*, vol. 2, no. 2, pp. 129–233, 2010.

[19] L. R. Acharya and D. Zhu, "Estimating an optimal correlation structure from replicated molecular profiling data using finite mixture models," in *Machine Learning and Applications, 2009. ICMLA'09. International Conference on*. IEEE, 2009, pp. 119–124.

[20] W. H. DuMouchel, "On the asymptotic normality of the maximum-likelihood estimate when sampling from a stable distribution," *The Annals of Statistics*, pp. 948–957, 1973.

[21] S. Van Der Walt, S. C. Colbert, and G. Varoquaux, "The numpy array: a structure for efficient numerical computation," *Computing in Science & Engineering*, vol. 13, no. 2, pp. 22–30, 2011.

[22] T. E. Oliphant, "Python for scientific computing," *Computing in Science & Engineering*, vol. 9, no. 3, pp. 10–20, 2007.

[23] J. D. Hunter *et al.*, "Matplotlib: A 2d graphics environment," *Computing in science and engineering*, vol. 9, no. 3, pp. 90–95, 2007.

[24] M. S. Handcock, D. R. Hunter, C. T. Butts, S. M. Goodreau, and M. Morris, *statnet: Software tools for the Statistical Modeling of Network Data*, Seattle, WA, 2003. [Online]. Available: http://statnetproject.org

[25] M. Bastian, S. Heymann, and M. Jacomy, "Gephi: An open source software for exploring and manipulating networks," 2009. [Online]. Available: http://www.aaai.org/ocs/index.php/ICWSM/09/paper/view/154

[26] L. Hubert and P. Arabie, "Comparing partitions," *Journal of classification*, vol. 2, no. 1, pp. 193–218, 1985.

[27] V. Krebs, "Books about us politics," *unpublished, http://www. orgnet. com*, 2004.

APPENDIX

A. Conventional ConnectivityOnly Model (COM)

We start by introducing the conventional ConnectivityOnly approach for network data clustering. The probabilistic mixture model is a powerful technique for detecting structural features and connectivity patterns in network data [9]. In this paper, we name this kind of method as ConnectivityOnly model (COM) because it does not model over-dispersion in network clusters. This type of probabilistic mixture method aims at capturing the connectivity information by modeling the probability of a link from a particular node in a certain k^{th} cluster to a node j, which is denoted as θ_{kj}. The likelihood of COM can be written as:

$$L_{\mathrm{COM}}(X, Z_{ik}|\phi_k, \theta) = \prod_{i=1}^{n}\prod_{k=1}^{K}(\phi_k p(X_{(i,:)}, Z_{ik}=1|\theta))^{Z_{ik}} = \prod_{i=1}^{n}\prod_{k=1}^{K}\left(\phi_k \prod_{j=1}^{n}\theta_{kj}^{X_{ij}}\right)^{Z_{ik}}. \quad (10)$$

As probabilities, both ϕ_k and θ_{kj} have been normalized, i.e., $\sum_{k=1}^{K}\phi_k = 1$ and $\sum_{j=1}^{n}\theta_{kj} = 1$. However, the parameters in Eq. (10) cannot be estimated via maximum likelihood directly, due to Z_{ik} is a latent variable. Expectation-Maximization (EM) algorithm is a viable technique to overcome this limitation.

In the E-step, we calculate the expected values of Z_{ik} by the following form:

$$\tau_{ik}^{(l-1)} = \frac{\phi_k^{(l-1)} p(X_{(i,:)}, Z_{ik}=1|\theta^{(l-1)})}{\sum_{k'=1}^{K}\phi_{k'}^{(l-1)} p(X_{(i,:)}, Z_{ik'}=1|\theta^{(l-1)})}, \quad (11)$$

where l is the current iteration number.

In the M-step, we estimate the parameter values by maximizing the expected complete data log likelihood:

$$Q(\Theta, \Theta^{(l-1)}) = \sum_{i=1}^{n}\sum_{k=1}^{K}\tau_{ik}^{(l-1)}(\log\phi_k^{(l-1)} + \sum_{j=1}^{n}X_{ij}\log\theta_{kj}^{(l-1)}), \quad (12)$$

which is also known as auxiliary function, where the notation Θ represents the complete set of related parameters in COM. Through simple calculation we can get that ϕ_k is updated by summarizing the expected counts of nodes as:

$$\phi_k^{(l)} = \sum_{i=1}^{n}\frac{\tau_{ik}^{(l-1)}}{n}, \quad (13)$$

and θ_{kj} can be updated as following:

$$\theta_{kj}^{(l)} = \frac{\sum_{i=1}^{n}X_{ij}\tau_{ik}^{(l-1)}}{\sum_{i=1}^{n}X_{i\cdot}\tau_{ik}^{(l-1)}}, \quad (14)$$

where $X_{i\cdot} = \sum_{j=1}^{n}X_{ij}$ is the degree of node i.

B. Multinomial Connectivity-based Mixture Model (MCM)

Besides Poisson distribution, multinomial distribution can be also used to model the cluster size. However, asymptotically converges to normal distribution [20] that is not qualified to capture the *over-dispersed* cluster size distribution. To prove this, we name this approach as multinomial connectivity-based mixture model (MCM) using multinomial distribution to model the cluster size by modeling the latent variable Z_{ik} instead of Poisson distribution:

$$p(Z_{ik}=1) = n! \prod_{k=1}^{K} \frac{\phi_k^{n_k}}{n_k!}. \tag{15}$$

The likelihood function of MCM can be shown as:

$$L_{\mathrm{MCM}}(X|\phi_k,\theta,\lambda_k) = \prod_{i=1}^{n}\prod_{k=1}^{K}\left(\phi_k n! \prod_{k=1}^{K}\frac{\phi_k^{n_k}}{n_k!}\prod_{j=1}^{n}\theta_{kj}^{X_{ij}}\right)^{Z_{ik}}. \tag{16}$$

In the E-step τ_{ik} is updated at the l^{th} iteration as:

$$\tau_{ik}^{(l-1)} = \frac{p(Z_{ik}=1)\phi_k^{(l-1)} p(X_{(i,:)}, Z_{ik}=1|\theta^{(l-1)})}{\sum_{k'=1}^{K} p(Z_{ik'}=1)\phi_{k'}^{(l-1)} p(X_{(i,:)}, Z_{ik'}=1|\theta^{(l-1)})}, \tag{17}$$

In the M-step, we maximize the $Q(\Psi,\Psi^{(l-1)})$ by updating ϕ_k and Ψ_{kj} which are initialized by assigning random number with normalizing to $[0,1]$ and estimated by calculating the value which makes the first derivative of $Q(\Psi,\Psi^{(l-1)})$ equal to zero. Let Ψ denote a set of MCM related parameters, and hence the expected complete data log-likelihood is given as:

$$Q(\Psi,\Psi^{(l-1)}) = \sum_{i=1}^{n}\sum_{k=1}^{K}\tau_{ik}^{(l-1)}\Bigg(\log\phi_k^{(l-1)} + \log n! + \sum_{k=1}^{K} n_k^{(l-1)}\log\phi_k^{(l-1)} - \sum_{k=1}^{K}\log n_k^{(l-1)} + \sum_{j=1}^{n} X_{ij}\log\theta_{kj}^{(l-1)}\Bigg). \tag{18}$$

Thus, we have:

$$\phi_k^{(l)} = \sum_{i=1}^{n}\frac{\tau_{ik}^{(l-1)}}{n} = \frac{n_k}{n}, \tag{19}$$

$$\theta_{kj}^{(l)} = \frac{\sum_{i=1}^{n} X_{ij}\tau_{ik}^{(l-1)}}{\sum_{i=1}^{n} X_{i\cdot}\tau_{ik}^{(l-1)}}. \tag{20}$$

We can see that the two newly derived parameters $\phi_k^{(l)}$ in Eq. (19) and $\theta_{kj}^{(l)}$ in Eq. (20) are the same as the ones in Eq. (13) and in Eq. (14). And both COM and MCM have the only two parameters. Hence, COM and MCM are equivalent that our experimental results also indicate this argument. This is why we choose the Poisson distribution as one of components in our proposed SCG approach over multinomial distribution.

C. SizeOnly Model

In order to present a more comprehensive analysis of the effect of cluster size, we use controlling variables method, i.e., we analyze two models which only focus on modeling the cluster size information but omit the connectivity information of network. Similar as, we choose to use two distributions, Poisson distribution and Multinomial distribution, to model the cluster size.

1) Poisson Mixture Model (PMM): By assuming there are n_k nodes in the k^{th} cluster, the size of the k^{th} cluster can be modeled using a discrete distribution such as Poisson distribution, so that we can discriminate clusters simply by their sizes. The likelihood function of the SizeOnly model which is Poisson Mixture Model (PMM) can be shown as:

$$L_{\mathrm{PMM}}(Z_{ik}|\lambda_k,\phi_k) = \prod_{i=1}^{n}\prod_{k=1}^{K}\left(\phi_k\frac{\lambda_k^{n_k}e^{-\lambda_k}}{n_k!}\right)^{Z_{ik}}, \tag{21}$$

The Q function of PMM can be calculated as:

$$Q(\Lambda,\Lambda^{(l-1)}) = \sum_{i=1}^{n}\sum_{k=1}^{K}\tau_{ik}^{(l-1)}\left(\log\frac{(\lambda_k^{(l-1)})^{n_k^{(l-1)}}e^{-\lambda_k^{(l-1)}}}{n_k^{(l-1)}!} + \log\phi_k^{(l-1)}\right). \tag{22}$$

Then we maximize the Q function value by updating λ_k, which is initialized by assuming each cluster size is equal at first, as follows:

$$\lambda_k^{(l)} = n_k^{(l)}. \tag{23}$$

$$\phi_k^{(l)} = \sum_{i=1}^{n}\frac{\tau_{ik}^{(l-1)}}{n}. \tag{24}$$

2) Multinomial Mixture Model (MMM): Instead of using Poisson distribution to model the each cluster size, we can also use multinomial distribution to model the clusters' sizes. Without considering the connectivity patterns, the likelihood function of the SizeOnly framework using multinomial mixture model (MMM) can be shown as:

$$L_{\mathrm{MMM}}(Z_{ik}|\phi_k) = \prod_{i=1}^{n}\prod_{k=1}^{K}\left(\phi_k n!\prod_{k=1}^{K}\frac{\phi_k^{n_k}}{n_k!}\right)^{Z_{ik}}. \tag{25}$$

Auxiliary function:

$$Q(\Delta,\Delta^{(l-1)}) = \sum_{i=1}^{n}\sum_{k=1}^{K}\tau_{ik}^{(l-1)}\Bigg(\log\phi_k^{(l-1)} + \log n! + \sum_{k=1}^{K} n_k^{(l-1)}\log\phi_k^{(l-1)} - \sum_{k=1}^{K}\log n_k^{(l-1)}\Bigg), \tag{26}$$

where Δ is a set of MMM related parameters.

We maximize the $Q(\Delta,\Delta^{(l-1)})$ by updating ϕ_k, which is initialized by assigning random number with normalizing to $[0,1]$, as follows:

$$\phi_k^{(l)} = \sum_{i=1}^{n}\frac{\tau_{ik}^{(l-1)}}{n}. \tag{27}$$