Community Detection in Complex Networks Based on DBSCAN* and a Martingale Process

Ilias Gialampoukidis, Theodora Tsikrika, Stefanos Vrochidis and Yiannis Kompatsiaris
Information Technologies Institute
Centre for Research and Technology Hellas
Email: {heliasgj, theodora.tsikrika, stefanos, ikom}@iti.gr

***Abstract*—Community detection is a valuable tool for analyzing complex networks. This work investigates the community detection problem based on the density-based algorithm DBSCAN*. This algorithm requires, though, a lower bound for the community size to be determined a priori, a challenging task. To this end, this work proposes the application of a Martingale process to DBSCAN* that progressively detects communities at various levels of granularity. The proposed DBSCAN*-Martingale community detection algorithm corresponds to an iterative process that progressively lowers the threshold of the size of the acceptable communities, while maintaining the communities detected for higher thresholds. Evaluation experiments are performed based on four realistic benhmark networks and the results indicate improvements in the effectiveness of the proposed DBSCAN*-Martingale community detection algorithm in terms of the Normalized Mutual Information and the RAND metrics against several state-of-the-art community detection approaches.**

I. Introduction

Community detection in complex networks aims to identify groups of nodes that are more densely connected to each other than to the rest of the network [1] and thus probably share common properties and/or play similar roles within the network [2]. The detection of the community structure of networks is of great importance in many fields, including sociology and biology [3], as well as computer science [4], i.e. disciplines where systems are often represented as networks. More recently, there has been increasing interest in detecting communities on the Web [5] and social media [1] so as to both gain valuable insights into the particular characteristics and latent phenomena in such networks, and also to exploit the detected communities in various applications, such as in the detection of events in social media streams.

Detecting communities in complex networks is also known as a graph partition problem, given that networks are usually modelled as graphs. A graph can be split into communities in numerous ways, i.e. for each graph there are many possible community structures. In the simple case, a community structure is defined as a graph partition into a set of node sets.

Several community detection algorithms have been proposed (e.g. [2], [6], [7], [3], [8], [9], [10], [11]). The quality of their results is often evaluated by the use of modularity [4], particularly in the absense of appropriate ground-truth. Hence, several approaches use modularity optimization itself as a method for the detection of communities in complex networks [2]. Alternative to the maximization of modularity, the minimization of the so-called codelength description, being the minimum Shannon information needed to describe a random walk on the network, has also played a key role in revealing community structure [11]. However, none of these approaches is able to identify noise, i.e. nodes that are not members of any community. To address this issue, *density-based community detection* approaches are more appropriate since they provide support for leaving spuriously connected nodes (i.e. noise) out of the detected community structure.

DBSCAN* [12], the graph analogue of the well-established DBSCAN [13] algorithm, is such a density-based approach that could be applied to community detection. Similarly to DBSCAN, it relies on two parameters, the density level ϵ and a lower bound $MinPts$ for the number of nodes that may form a community. Both these parameters greatly affect the output of the algorithm, but their estimation is far from trivial. To address this issue, and in particular the estimation of the $MinPts$ parameter, this work proposes an extension to DBSCAN* based on Doob's Martingale [14], which involves the construction of a Martingale that progressively gains knowledge about the communities in the network based on an iterative application of DBSCAN* for several values of $MinPts$.

The main contributions of this work are three-fold: (i) the application of DBSCAN* to the community detection problem, (ii) the proposal of a Martingale process for community detection based on DBSCAN*, and (iii) the experimental evaluation of the proposed DBSCAN*-Martingale community detection algorithm against several state-of-the-art community detection approaches by using four realistic benhmark networks [15]. The proposed DBSCAN*-Martingale community detection algorithm is presented in Section III and its experimental evaluation is reported in Section IV. First, though, the state-of-the-art in community detection is discussed next.

II. Related Work

A large number of community detection algorithms has appeared in the literature (e.g. [2], [6], [7]), but only few of them are large scale algorithms that are directly applicable in large social media graphs, as reviewed in [1].

The GirvanNewman community detection algorithm [3], [4] is a divisive hierarchical process, based on the edge betweenness centrality measure, which may be quickly calculated [16]. The edge betweenness is measured by the number of shortest paths that pass through a given edge and determines the edges which are more likely to connect different communities. The edge with the highest edge betweenness is removed and the remaining edges are re-assigned new edge

This is a draft version of the paper. The final version will appear at IEEEXplore: 978-1-5090-5246-2/16/$31.00 ©2016 IEEE
In: Proceedings of the 11th International Workshop on Semantic and Social Media Adaptation and Personalization

betweenness scores. The process generates a dendrogram with root node the whole graph and leaves the graph vertices. In order to extract the detected communities, the modularity score is computed at each dendrogram cut, so as to be maximized. The GirvanNewman algorithm requires the maximization of a modularity function, as a stopping criterion, for the optimal extraction of communities. An alternative hierarchical approach for community detection has been proposed [17], using the modularity function as an objective function to optimize. Initially, all vertices are separate communities and any two communities are merged if the modularity increases. The algorithm stops when the modularity is not increasing anymore.

In the Label Propagation method [8], every node is initialized with a unique label and at every step each node adopts the label that most of its neighbors currently have. Hence, an iterative process is defined, in which the densely connected groups of nodes form a consensus on a unique label and communities are extracted.

The Louvain method [9] is based on the maximization of the modularity and involves two phases that are repeated iteratively. In the first phase, each vertex forms a community and for each vertex *i* the gain of modularity is calculated for removing vertex *i* from its own community and placing it into the community of each neighbor *j* of *i*. The vertex *i* is moved to the community for which the gain in modularity becomes maximal. In case the modularity decreases or remains the same, vertex *i* does not change community. The first phase is completed when the modularity cannot be further increased. In the second phase, the detected communities formulate a new network with weights of the links between the new nodes being the sum of weights of the links between nodes in the corresponding two communities. In this new network, self-loops are allowed, representing links between vertices of the same community. At the end of the second phase, the first phase is re-applied to the new network, until no more communities are merged and the modularity attains its maximum.

The Walktrap method [10] generates random short walks on the graph by simulating transitions from one node to another. Since short random walks tend to stay within the same community, it is possible to detect communities using such random walks.

The Infomap method [11], [18], [19] is an information-theoretic approach for community detection. The inventors of the Infomap method showed that the problem of finding a community structure in networks is equivalent to solving a coding problem. In general, the goal of a coding problem is to minimize the information required for the transmission of a message. Initially, Infomap employs the Huffman code [20] in order to give a unique name (codeword) in every node in the network. In contrast to the Louvain method, which maximizes modularity, Infomap minimizes the Shannon information [20] required to describe the trajectory of a random walk on the network. The objective function, which minimizes the description length of a random walk on the network (described by the corresponding sequence of codewords on each visited node), is called the map equation [11], [18], [19], and is minimized over all possible network partitions.

DBSCAN [13] is a density-based clustering algorithm, which is able to extract clusters without knowing the number of clusters, even in the case where there is noise in the spatial collection of points. The clustering is based on two parameters ϵ and $MinPts$, which are determined by the desired density level ϵ and a lower bound for the number of points in a cluster $MinPts$. The estimation of the density level, however, is not a trivial task and several approaches have been proposed to extract clusters, using DBSCAN, without determining the parameter ϵ, such as the DBSCAN-Martingale [21]. The graph-analogue of DBSCAN is called DBSCAN* [12] and defines core objects on a graph, in a way similar to the core points of DBSCAN. The transition from density-based clustering of spatial databases to community detection in graphs, through DBSCAN* does not involve border points, due to the "updated" definition of reachability.

III. DBSCAN*-Martingale Community Detection

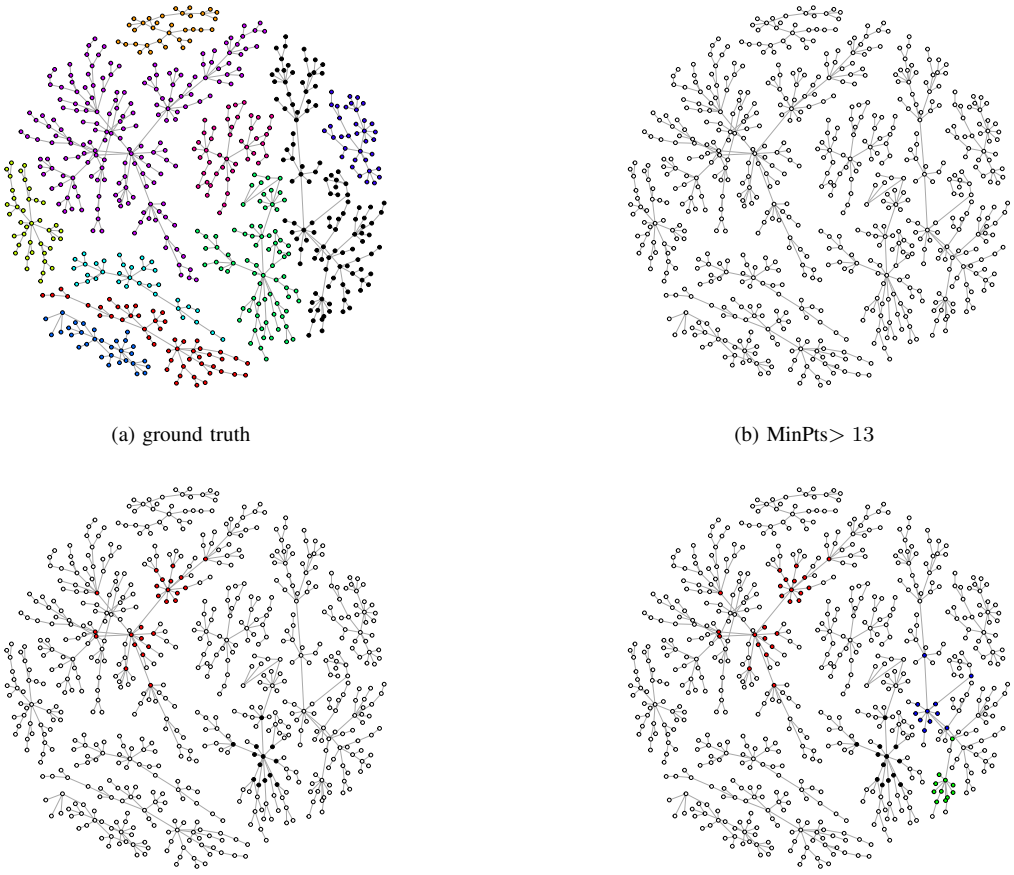
A. Notation and Preliminaries on DBSCAN* and Martingales

Given a network $G(N, E)$ with N nodes and E edges, density-based community detection algorithms partition the network into k communities, where $N_c \subseteq N$ of the nodes belong to the detected communities, while the $N \setminus N_c$ nodes that were not assigned to any of the communities are labeled as "noise". The output of such algorithms corresponds to an N-dimensional vector C. For each node n_j, $j = 1, 2, \ldots, N$, the j-th element of C, denoted as $C[j]$, is assigned the ID $\{1, 2, \ldots, k\}$ of the community the node n_j belongs; if a node does not belong to any of the communities, the value 0 is assigned instead. As a result, the communities vector C is an N-dimensional vector with values in $\{0, 1, 2, \ldots, k\}$.

DBSCAN* relies on two parameters, the density level ϵ and the minimum number $MinPts$ of nodes that can form a community. We denote the communities vector provided by DBSCAN* as $C_{DBSCAN*(\epsilon,MinPts)}$. As this work considers that the parameter ϵ is fixed, the communities vector is denoted as $C_{DBSCAN*(MinPts)}$. High values of $MinPts$ typically result in a $C_{DBSCAN*(MinPts)}$ vector of zeros, i.e. all nodes are marked as noise, since the algorithm fails to detect communities required to have at least $MinPts$ nodes. On the other hand, low values of $MinPts$ result in a single community and thus the partitioning is trivial.

The output of DBSCAN* strongly depends on the parameter $MinPts$. This is illustrated by the example depicted in Figure 1. Figure 1a shows the ground-truth communities as disconnected components for illustrative purposes. A high value of $MinPts$ ($MinPts > 13$) results in no communities being detected (Figure 1b). For $MinPts = 13$, two communities are detected (Figure 1c), while for $MinPts = 11$, two additional communities are detected (Figure 1d). Lower values of $MinPts$ result in the detection of further communities, but at the same time they merge communities that would have been detected as separate by higher values of $MinPts$.

This indicates that a single value of $MinPts$ may not allow to detect all communities and motivates us to consider that an iterative process would be more appropriate for detecting communities in an effective manner. In particular, starting from high values of $MinPts$, so that the larger communities are detected, and progressively decreasing $MinPts$, so that



(a) ground truth

(b) MinPts$>$ 13

(c) MinPts$=$ 13

(d) MinPts$=$ 11

Fig. 1. Community detection in a social network consisting of 650 nodes using DBSCAN* with $\epsilon = 1$ and various values of $MinPts$.

further, smaller, communities are detected, would result in a set of communities that are detected based on different values of $MinPts$; this process would continue until a minimum acceptable threshold of community size is applied. To this end, we propose an extension of DBSCAN* based on Doob's Martingale, which allows for introducing a random variable $MinPts$ and involves the construction of a Martingale process, which progressively approaches the $C_{DBSCAN*(MinPts)}$ vector that contains all communities.

Martingale is a stochastic process, i.e. a sequence of random variables $X_1, X_2, \ldots$, for which the expected future value of X_{s+1}, given all prior values $X_1, X_2, \ldots, X_s$, is equal to the present observed value X_s. A well-known martingale is Doob's Martingale, in which our knowledge about a random variable is progressively obtained and is defined as follows:

Definition 1: (Doob's Martingale) [14] Let $X, Y_1, Y_2, \ldots$ be any random variables with finite expectation $E[|X|] < \infty$. Then, if X_s is defined by the conditional expectation $X_s = E[X|Y_1, Y_2, \ldots, Y_s]$, the sequence of random variables $X_1, X_2, \ldots$ is a martingale.

We shall introduce a probabilistic method that constructs a *Martingale* stochastic process for progressively detecting all communities based on DBSCAN* and a given density level ϵ. The martingale construction is based on Doob's martingale (Definition 1), where knowledge is progressively gained about the result of a random variable.

B. Progressive Community Detection Based on a Martingale

In the context of a community detection problem, the random variable that needs to be known is the vector of communities' IDs, which is a combination of S communities' vectors $C_{DBSCAN*(MinPts_s)}$, each generated for a different value $MinPts_s, s = 1, 2, \ldots, S$. For each application of DBSCAN*, the parameter ϵ is set to 1 so that only the immediate neighborhood of each node is considered. Neighborhoods of order greater than 2 tend to merge different communities, because all communities are mutually reachable by intermediate nodes much easier than the case where neighborhoods are considered to be of order 1.

First, we generate S random numbers $MinPts_s, s = 1, 2, \ldots, S$ uniformly in $[MinPts_{min}, MinPts_{max}]$, a range of thresholds for the minimum community size. The sample of $MinPts_s, s = 1, 2, \ldots, S$ is sorted in decreasing order. Initially, there are no communities detected in the network. In the first iteration $(s = 1)$, all communities detected by $C_{DBSCAN*(MinPts_1)}$ are kept, corresponding to the community size threshold $MinPts_1$, i.e. the largest value in the range. In the second iteration $(s = 2)$, some of the detected communities by $C_{DBSCAN*(MinPts_2)}$ are new and some of

them were previously detected at iteration $(s = 1)$. In order to keep only the newly detected communities of the second iteration $(s = 2)$, we keep only the group of numbers of the same cluster ID with size greater than or equal to $MinPts_2$, but lower than $MinPts_1$, and set the rest to 0.

Formally, we define the sequence of communities $C^{(s)}, s = 1, 2, \ldots, S$, where $C^{(1)} = C_{DBSCAN^*(MinPts_1)}$ and:

$$C^{(s)}[j] := \begin{cases} 0, \text{ if } \mathrm{n}_j \in \text{ a previously detected community} \\ C_{DBSCAN^*(MinPts_s)}[j], \text{ otherwise} \end{cases} \quad (1)$$

Finally, we relabel the IDs of the detected communities. Assuming that r new communities are detected at iteration s, we update the labels of $C^{(s)}$ starting from $1 + \max_j C^{(s-1)}[j]$ to $r + \max_j C^{(s-1)}[j]$. The sum of all vectors $C^{(s)}$ up to stage S is the final communities vector of our algorithm:

$$C = C^{(1)} + C^{(2)} + \cdots + C^{(S)} \quad (2)$$

The sequence of vectors $X_s = C^{(1)} + C^{(2)} + \cdots + C^{(s)}, s = 1, 2, \ldots, S$ is Doob's martingale for the sequence of random variables $Y_t = C_{DBSCAN^*(MinPts_s)}, s = 1, 2, \ldots, S$. Each random selection of $MinPts_s, s = 1, 2, \ldots, S$ provides one vector $C_{DBSCAN^*(MinPts_s)}$ of community IDs for all $s = 1, 2, \ldots, S$. As s decreases, more vectors are combined and we gain knowledge about the final vector C of community IDs. The vector $C^{(1)} + C^{(2)} + \cdots + C^{(S)}$ is our "best prediction" for the final vector C of community IDs at stage s. The expected final vector of community IDs at stage $s = S$ has extracted all available communities of various sizes.

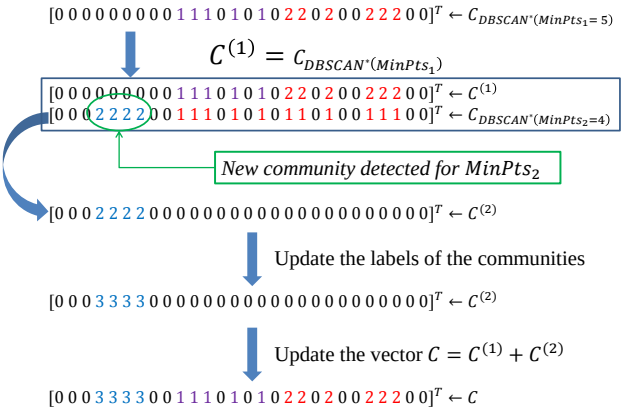


Fig. 2. DBSCAN*-Martingale for $S = 2$ iteraions. The two communities detected at the first iteration are re-discovered in the second iteration as a single community, but the update keeps them as separate, together with the newly discovered community of the second iteration.

This DBSCAN*-Martingale process that detects communities in a progressive manner and combines them in a single communities vector is presented as pseudo-code in Algorithm 1 and it is also illustrated in Figure 2 for two iterations and values $MinPts_1 = 5$ and $MinPts_2 = 4$, where X^T denotes the transpose of vector X.

The DBSCAN*-Martingale may not assign all nodes to a community. To address this issue, an optional propagation step is applied where each unassigned node is assigned

Algorithm 1: DBSCAN*-Martingale(ϵ, $MinPts$) **return** C

```
Generate a random sample of S values in [MinPts_min, MinPts_max]
Sort the generated sample s, s = 1, 2, ..., S
for t = 1 to S
    find C_{DBSCAN*(ε, MinPts_s)}
    compute C^(s) as in Eq. (1)
    update the community IDs
    update the vector C as in Eq. (2)
end for
return C
```

to the community that belongs to its ϵ-neighborhood. This propagation process is iteratively repeated until there are no unassigned nodes in the connected components of the detected communities. Figure 3 illustrates this process for the case of two communities detected by DBSCAN*-Martingale with $\epsilon = 1$, i.e. t_0 signifies the start of the propagation process following the end of the community detection algorithm. At each iteration $t_i, i > 0$, these two communities are expanded with their immediate neighbours (since $\epsilon = 1$) and after five iterations, both communities consist of all nodes in their connected component.

The DBSCAN*-Martingale requires S iterations of the DBSCAN* algorithm, which runs in $\mathcal{O}(N \log N)$ if a tree-based spatial index is used and in $\mathcal{O}(N^2)$ without tree-based spatial indexing [22]. Therefore, the DBSCAN*-Martingale runs in $\mathcal{O}(SN \log N)$ for tree-based indexed datasets and in $\mathcal{O}(SN^2)$ without tree-based indexing. The optional propagation step has worst-case complexity $\mathcal{O}(N)$, since in the worst case scenario the algorithm will examine all nodes for deciding whether to update their community ID or not. Our code is written in R[1] and uses the DBSCAN-Martingale implementation available on Github[2] for implementing the proposed DBSCAN*-Martingale.

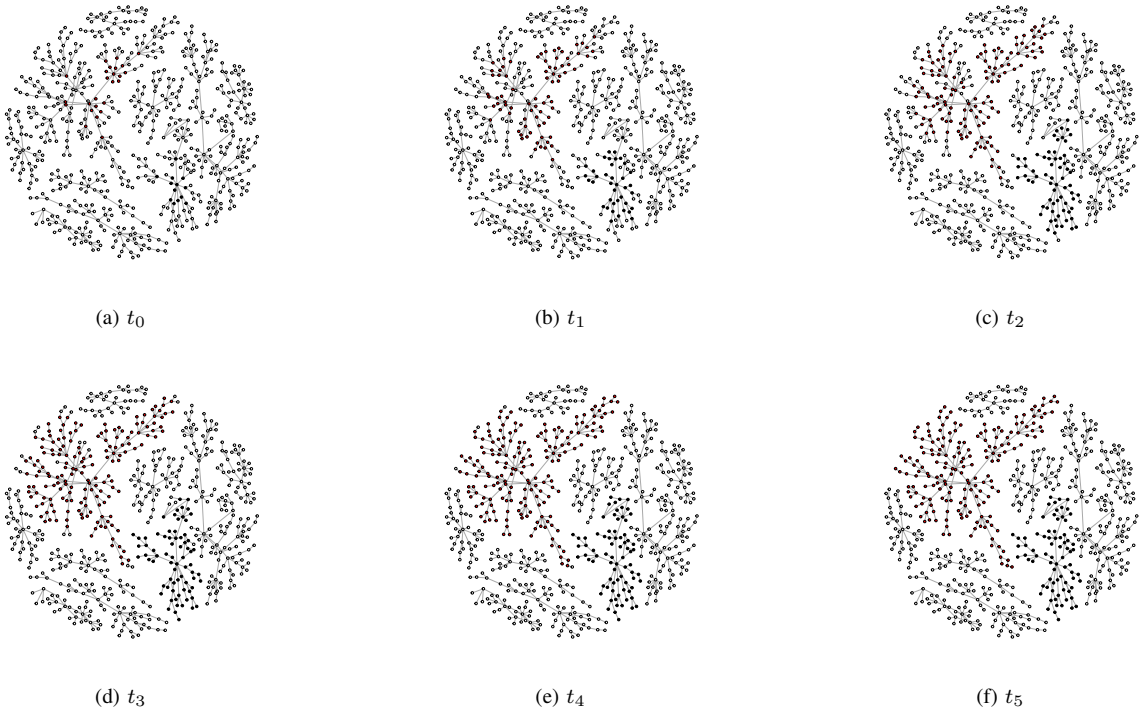
IV. Evaluation

A. Experimental Set-Up

Evaluation is performed using the community detection benchmark networks developed by Lancichinetti, Fortunato, and Radicchi (LFR) [15]. These LFR networks were developed with the goal to reflect the structure of real networks and in particular to account for the heterogeneity in the distributions of node degrees and of community sizes. This work employs four such networks, namely LFR1, LFR2, LFR3 and LFR4, constructed under the realistic assumptions (i) the network is scale-free and its degree distribution has a power-law behavior with power-law exponent τ_1, (ii) the community sizes also obey a power-law distribution with exponent τ_2 and (iii) the communities are mixed, i.e. links appear from a node in a community i to a node in a community j, where $i \neq j$. The ratio of links between different communities to the number of links within a community determines the mixing parameter μ. When $\mu = 0$ there is no mixing, thus all communities are also disconnected components, and when $\mu = 1$ there is no community structure.

We used four datasets of sizes 650, 3,182, 21,226 and 41,791 nodes with 10, 50, 200 and 50 communities, respectively. Their characteristics are as follows:

[1] https://www.r-project.org/
[2] https://github.com/MKLab-ITI/topic-detection



(a) t_0 (b) t_1 (c) t_2

(d) t_3 (e) t_4 (f) t_5

Fig. 3. Iterative propagation of community membership to unassigned nodes until all nodes in the connected components of the communities detected by DBSCAN*-Martingale are assigned to a community.

TABLE I. COMMUNITY DETECTION EVALUATION RESULTS.

Method	Size							
	650		3,182		21,226		41,791	
	NMI	RAND	NMI	RAND	NMI	RAND	NMI	RAND
Edge Betweenness [4]	0.7018	0.8793	0.8567	0.9601	NA	NA	NA	NA
Fast Greedy [17]	0.7038	0.8808	0.8543	0.9598	0.7046	0.8196	0.4177	0.6303
Label Propagation [8]	0.5930	0.8553	0.7116	0.9490	0.5458	0.8144	0.2882	0.6255
Louvain [9]	0.6947	0.8792	0.8589	0.9606	0.7077	0.8198	0.4200	0.6305
Walktrap [10]	0.6904	0.8808	0.8653	0.9621	0.7081	0.8336	0.3842	0.6529
Infomap [18], [19]	0.5852	0.8551	0.7180	0.9488	0.5569	0.8144	0.2954	0.6255
DBSCAN*-Martingale	**0.7898**	**0.9303**	**0.8665**	**0.9626**	**0.7234**	**0.8437**	**0.4526**	**0.6627**

- **LRF benchmark dataset 1 (LRF1)**: 10 communities, 650 vertices, minimum community size 20, community size power-law fit beta = 1.89 (p-value = $0.16 > 0.05$), degree distribution power-law fit gamma = 3.54 (p-value = $0.29 > 0.05$) and maximum degree = 13.
- **LRF benchmark dataset 2 (LRF2)**: 50 communities, 3,182 vertices, minimum community size 15, community size power-law fit beta = 1.98 (p-value = $0.93 > 0.05$), degree distribution power-law fit gamma = 3.63 (p-value = $0.99 > 0.05$) and maximum degree = 28.
- **LRF benchmark dataset 3 (LRF3)**: 200 communities, 21,226 vertices, minimum community size 10, community size power-law fit beta = 2.00 (p-value = $0.70 > 0.05$), degree distribution power-law fit gamma = 3.33 (p-value = $0.13 > 0.05$) and maximum degree = 52.
- **LRF benchmark dataset 4 (LRF4)**: 50 communities, 41,791 vertices, minimum community size 10, community size power-law fit beta = 1.69 (p-value = $0.87 > 0.05$), degree distribution power-law fit gamma = 3.49 (p-value=$0.98 > 0.05$) and maximum degree = 124.

All these datasets have ground truth community structure, i.e. they provide annotated graph nodes based on the community they belong to.

The proposed DBSCAN*-Martingale is evaluated against the well-established and parameter-free community detection algorithms presented in Section II and listed in Table I; to this end, their respective implementations in igraph (version 1.0.1, date: 2015-06-26) are used. Based on preliminary experiments, the range of $MinPts$ values was set to $[5, 30]$ and the number of iterations S to 5. The parameter ϵ was set to 1 as many community detection approaches consider only the immediate neighborhood of each node. In addition, the propagation process was applied for determining the community membership of some of the unassigned nodes, given that the LRF datasets provide ground truth for all nodes, i.e. no nodes are left unassigned. Finally, the most prominent evaluation measures in community detection were employed, namely Normalized Mutual Information [23] and RAND [24].

B. Results

Table I presents the results of the evaluation experiments in each of the four datasets. All community detection approaches were applied in all datasets, apart from the GirvanNewman (Edge Betweenness) approach [4] which is applicable only to small-scale datasets and thus it was not applied to LFR3 and LFR4.

The proposed DBSCAN*-Martingale is the best performing community detection approach for both evaluation metrics across all datasets, indicating its quality and robustness across heterogeneous networks of different sizes. The most significant differences to the other approaches for both evaluation metrics are observed for the smallest LRF dataset. For LRF1, DBSCAN*-Martingale indicates improvements over the other community detection approaches ranging from 12% to 35% in terms of NMI and ranging from 5.6% to 8.8% in terms of RAND. In the larger datasets, the DBSCAN*-Martingale still performs better than all the other approaches, but the differences in the effectiveness are smaller, particularly for the RAND evaluation metric.

Interestingly, the second best performing community detection approach is Walktrap [10], with the exception of NMI for LFR1 and LFR2, where the Fast Greedy [17] and the Louvain [9] methods perform second best, respectively.

V. Conclusions

This work proposed a novel community detection approach based on the DBSCAN* density-based algorithm and a Martingale process that aims to progressively detect communities in complex networks at various levels of granularity. To this end, it applies an iterative process that progressively lowers the threshold of the size of the acceptable communities, while maintaining the communities detected for higher thresholds. The output of our proposed community detection approach is usually a .json file, which is then imported by other applications. Evaluation experiments over four benchmark datasets with diverse characteristics and sizes against several state-of-the-art community detection methods indicate the effectiveness and robustness of the proposed approach. Further work includes its application in large-scale social media networks where communities can be defined along various dimensions given the multitude of relationships that exist between users (i.e. the nodes in the network) and further optimizations for automatically determining the range of lower bound values to explore in the Martingale process based on the network characteristics. We expect that our method will achieve high performance especially in covert networks where communities are sparsely connected and not very mixed.

Acknowledgment

This work was partially supported by the European Commission by the projects MULTISENSOR (FP7-610411) and HOMER (FP7-312883).

References

[1] S. Papadopoulos, Y. Kompatsiaris, A. Vakali, and P. Spyridonos, "Community detection in social media," *Data Mining and Knowledge Discovery*, vol. 24, no. 3, pp. 515–554, 2012.

[2] S. Fortunato, "Community detection in graphs," *Physics reports*, vol. 486, no. 3, pp. 75–174, 2010.

[3] 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.

[4] M. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical Review E*, vol. 69, no. 2, p. 026113.

[5] Y. Dourisboure, F. Geraci, and M. Pellegrini, "Extraction and classification of dense communities in the web," in *Proceedings of the 16th international conference on World Wide Web*. ACM, 2007, pp. 461–470.

[6] F. D. Malliaros and M. Vazirgiannis, "Clustering and community detection in directed networks: A survey," *Physics Reports*, vol. 533, no. 4, pp. 95–142, 2013.

[7] S. Harenberg, G. Bello, L. Gjeltema, S. Ranshous, J. Harlalka, R. Seay, K. Padmanabhan, and N. Samatova, "Community detection in large-scale networks: a survey and empirical evaluation," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 6, no. 6, pp. 426–439, 2014.

[8] U. N. Raghavan, R. Albert, and S. Kumara, "Near linear time algorithm to detect community structures in large-scale networks," *Physical Review E*, vol. 76, no. 3, p. 036106, 2007.

[9] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of statistical mechanics: theory and experiment*, vol. 2008, no. 10, p. P10008, 2008.

[10] P. Pons and M. Latapy, "Computing communities in large networks using random walks." *J. Graph Algorithms Appl.*, vol. 10, no. 2, pp. 191–218, 2006.

[11] M. Rosvall and C. T. Bergstrom, "Maps of random walks on complex networks reveal community structure," *Proceedings of the National Academy of Sciences*, vol. 105, no. 4, pp. 1118–1123, 2008.

[12] R. J. Campello, D. Moulavi, and J. Sander, "Density-based clustering based on hierarchical density estimates," in *Advances in Knowledge Discovery and Data Mining*. Springer, 2013, pp. 160–172.

[13] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise." in *Kdd*, vol. 96, no. 34, 1996, pp. 226–231.

[14] J. L. Doob, "Stochastic processes," *Wiley, New York*, vol. 101, 1953.

[15] A. Lancichinetti, S. Fortunato, and F. Radicchi, "Benchmark graphs for testing community detection algorithms," *Physical review E*, vol. 78, no. 4, p. 046110, 2008.

[16] U. Brandes, "A faster algorithm for betweenness centrality*," *Journal of mathematical sociology*, vol. 25, no. 2, pp. 163–177, 2001.

[17] A. Clauset, M. E. Newman, and C. Moore, "Finding community structure in very large networks," *Physical review E*, vol. 70, no. 6, p. 066111, 2004.

[18] M. Rosvall, D. Axelsson, and C. T. Bergstrom, "The map equation," *The European Physical Journal Special Topics*, vol. 178, no. 1, pp. 13–23, 2009.

[19] L. Bohlin, D. Edler, A. Lancichinetti, and M. Rosvall, "Community detection and visualization of networks with the map equation framework," in *Measuring Scholarly Impact*. Springer, 2014, pp. 3–34.

[20] T. M. Cover and J. A. Thomas, *Elements of information theory*. John Wiley & Sons, 2012.

[21] I. Gialampoukidis, S. Vrochidis, and I. Kompatsiaris, "A hybrid framework for news clustering based on the dbscan-martingale and lda," in *Machine Learning and Data Mining in Pattern Recognition*. Springer, 2016, pp. 170–184.

[22] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, "Optics: ordering points to identify the clustering structure," in *ACM Sigmod Record*, vol. 28, no. 2. ACM, 1999, pp. 49–60.

[23] L. Danon, A. Diaz-Guilera, J. Duch, and A. Arenas, "Comparing community structure identification," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2005, no. 09, p. P09008, 2005.

[24] W. M. Rand, "Objective criteria for the evaluation of clustering methods," *Journal of the American Statistical association*, vol. 66, no. 336, pp. 846–850, 1971.