Evolutionary Role Mining in Complex Networks by Ensemble Clustering

Sarvenaz Choobdar, Pedro Ribeiro, Fernando Silva
CRACS & INESC-TEC
DCC-FCUP, Universidade do Porto, Portugal
{sarvenaz,pribeiro,fds}@dcc.fc.up.pt

ABSTRACT

The structural patterns in the neighborhood of nodes assign unique *roles* to the nodes. Mining the set of existing roles in a network provides a descriptive profile of the network and draws its general picture. This paper proposes a new method to determine structural roles in a dynamic network based on the current position of nodes and their historic behavior. We develop a temporal ensemble clustering technique to dynamically find groups of nodes, holding similar tempo-structural roles. We compare two weighting functions, based on age and distribution of data, to incorporate temporal behavior of nodes in the role discovery. To evaluate the performance of the proposed method, we assess the results from two points of view: 1) goodness of fit to current structure of the network; 2) consistency with historic data. We conduct the evaluation using different ensemble clustering techniques. The results on real world networks demonstrate that our method can detect tempo-structural roles that simultaneously depict the topology of a network and reflect its dynamics with high accuracy.

CCS Concepts

•**Mathematics of computing → Graph algorithms;**
•**Information systems → Clustering;**

Keywords

Graph mining, complex networks, structural role mining, evolutionary clustering, ensemble clustering

1. INTRODUCTION

In some complex networks, a subset of the nodes have labels such as demographic values, interests, beliefs or other characteristics of the nodes (users). Node classification involves determining the label of a node in a network that is partially labeled. Normally, it is assumed that some of the nodes have a predefined label and the labels for the rest of the nodes are predicted using relational classifiers [2]. Commonly, labels of nodes may fulfill specific roles. For example, in a Twitter network, users can be identified as an advertiser, a content contributor, or an information receiver. In LinkedIn, users can be associated with different professional roles such as engineer, salesperson, or a recruiter. Previous research work mainly focuses on using categorical and textual information to predict the attributes of users. However, it cannot be applied to a large number of users in real social networks, since much of such information is missing, possibly outdated and non-standard. The structural position of people in online social networks is quantitatively correlated to their actions [20]. The network characteristics reflect the social situations of users in an online society and can be used as predictors for node classification. In a supervised setting, Zhao et al. [26] used structural properties in combination with demographic features to predict social statuses of users in a network. In this paper we present a method that relies solely on structural properties to classify users.

An important aspect of complex networks is its temporal dimension, which has been studied from different angles such as community evolution [18], graph growth models [15] and link prediction [17]. Role based analysis of networks is another aspect of network dynamic study that depict networks evolution from a microscopic point of view. In a large dynamic network, the temporal structural behaviors of individual nodes can be learned by structural role mining which identifies unusual activities or patterns. For instance, in an IP-to-IP network, we may want to learn the "behavioral roles" of individual hosts and monitor their changes over time. This would allow us to characterize the dynamic behaviors of individual hosts and also detect when a machine or host becomes compromised, or begins having unusual behaviors with respect to the global network dynamics. Rossi and Gallagher defined temporal structural roles as a combination of similar structural features that were learned from the initial network. Since similar structural properties are combined into a single role, then each role represents a different structural pattern (or connectivity pattern) [21]. In this paper, we follow a similar definition of dynamic roles, but we propose dynamic role mining methods based on a clustering algorithm instead of block models.

We study the structural behavior of nodes to determine their role in a network by answering questions like: *How is the temporal behavior of nodes reflected in their structural roles? How can we detect dynamic roles of nodes?* We formulate

SAC 2017,April 03-07, 2017, Marrakech, Morocco

http://dx.doi.org/10.1145/3019612.3019815

and study the problem of evolutionary role extraction where a sequence of graph snapshots are given and the goal is to find the roles of active nodes at the current time. These roles must reflect the structure of the network at the current time and must be consistent with past roles.

The evolutionary role extraction must fulfill the following tasks: 1) roles of nodes at the current time should be close to previous time, if the connectivity of nodes does not deviate from previous time points; 2) the set of roles must be modified to reflect the new structure, if the structure of the network changes significantly. A possible solution to this problem is addressed by employing evolutionary clustering [3]. This method is an incremental process where clustering C_t is built up on C_{t-1}, and the cost function of the clustering algorithm is evaluated based on the original similarity feature space. Both of these characteristics of evolutionary clustering make it computationally expensive.

In this paper for the first time we use ensemble clustering [22] for temporal network data. Ensemble clustering combines multiple partitionings of a set of objects without accessing the original features. It has been shown that ensemble clustering can improve the results by aggregating the different partitionings of objects [10, 22]. Streh and Ghosh indicated distributed computing and robustness improvement as the main motivations of this method [22]. However, it has only been used for static data, with different partitionings of the same dataset. To the best of our knowledge, ensemble clustering has not been used for evolutionary clustering. In a recent paper by Lancichinetti and Fortunato, the dynamic communities in a network are explored by cluster aggregation [14]. They used a sliding window with fixed width to find consensus partitions which entails multiple clustering iterations with overlap to derive grouping of data.

The contributions of our work in this paper can be summarized as follows:

- We study how dynamics of individual nodes can depict the global temporality of network structure by predicting temporal roles in the network.
- We design a weighted clustering ensemble to dynamically learn tempo-structural roles of nodes in a dynamic network.
- We make an empirical comparison of different weighting functions for embedding temporal behavior of nodes in evolutionary role mining.
- We use topological and information theoretic cluster validation metrics to evaluate the performance of the proposed weighted ensemble clustering methodology in comparison to baseline methods.

In section 2 we summarize the background of our work. The formal definition of our method is presented in section 3. The data and experimental results are discussed in section 4. Concluding remarks are presented in section 5.

2. BACKGROUND

2.1 Role mining

For a static network, role extraction is defined as the process of finding groups of nodes with similar properties. In other words, this is a clustering task where nodes are grouped not based on their connectivity but because they hold a similar position in the network. This has been studied by other researchers [8], where nodes with the most outstanding properties are detected as singular motifs using outlier detection methods. Henderson et al. [11] found nodes roles regarding properties in their neighborhood by non-negative matrix factorization and using minimum description length (MDL) for determining the number of roles.

An important aspect of complex networks is the temporal dimension. It has been studied from different perspectives such as community evolution [18], graph growth models [15] and link prediction [17]. Role based analysis of networks is another aspect of network dynamic study that depicts networks evolution from a microscopic point of view. Previous works such as [5, 6] use a two-phase general methodology that was designed to characterize time evolving networks. In the first step of this methodology nodes are grouped by k-means clustering and classified based on their role in the network. In the second step a method is proposed to study the evolution of the network using a supervised approach. In this method a set of events happening in the network is defined for the roles in the network, and then rules that describe them are found using association rule mining. Rossi et al. [21] used the methodology proposed by [11] for static role extraction. They measure a set of features for nodes at each time snapshot then by stacking all the nodes×features matrices, they derive the matrix of features×roles by factorizing the stacked nodes×features matrix and iteratively generate the matrix of nodes×roles for each time.

2.2 Evolutionary clustering

Evolutionary clustering is defined by Chakrabarti et al. as *"the problem of processing time-stamped data to produce a sequence of clusterings; that is, a clustering for each time step of the system. Each clustering in the sequence should be similar to the clustering at the previous time step, and should accurately reflect the data arriving during that time step"* [3]. Evolutionary clustering finds application in the domains where the properties of objects change over time due to concept drift or noise. In such problem, at each time step a new set of data arrives to be clustered. To cluster the new data two issues need to be addressed: 1) new data must be clustered close to previous time if its structure is not significantly different or the changes are due to noise; 2) the clustering must be modified in a way to reflect the actual structure of new data and detect the deviations. These two objectives are modeled as cost functions in evolutionary clustering, called temporal cost (TC) and snapshot cost (SC) respectively. The overall cost of clustering at current time t is defined as follows:

$$cost = \alpha * SC(C_t, X_t) + (1 - \alpha) * TC(C_t, X_{t-1}) \qquad (1)$$

where C_t is the clustering of data X_t at time t and α is a user defined parameter to adjust the importance of historical data. Chakrabarti et al. modified hierarchical and k-means clustering algorithms to incorporate the defined cost function [3]. They measure the distance between the clusters across time by pairing the centroids of clusters. Another pioneering work in this area is by Chi et al. [4]. They proposed two frameworks for evolutionary clustering, the first one (PCQ) assesses the temporal cost at the data level, meaning it evaluates the new clustering on the old data. The second one (PCM) does the evaluation at model level, comparing the clusterings with each other using Chi-square statistics. They incorporate the cost functions into a spectral clustering framework and solve its relaxed version to derive the partitioning of the data.

2.3 Ensemble clustering

It has been shown that ensemble clustering can improve accuracy of results by aggregating multiple partitionings to alleviate the noise [10, 22]. It can be used in different applications such as network community discovery [1] or monitoring of communities evolution[14]. Given r partitionings over a set of objects, the objective of ensemble clustering is to obtain a single aggregated clustering. The ensemble clustering λ is the one that best matches with every base clustering. In other words, λ must minimize the cost function $\sum_{i=1}^{r} Dist(\lambda, C_i)$.

Strehl and Ghosh measured the cost function in terms of shared information between clusterings [22]. They used normalized mutual information (NMI) to measure the similarity of clusterings. Since finding the optimal combined clustering over the defined cost function is computationally expensive, they used heuristic solutions instead of optimization. Gionis et al. defined the cost function as the number of mismatches between the clusterings [10]. They proposed a number of approximate algorithms to find the aggregated clustering. The common approach of all the proposed methods is to build a new similarity between the objects to be clustered, using the clustering co-occurrence instead of their original feature space. This similarity matrix is used either directly to recluster the objects or to build a graph similarity of data and then derive the clustering by partitioning the graph.

3. EVOLUTIONARY ROLE MINING

We first introduce the notations that we will be using. We have a dynamic network $G_t = (V_t, E_t, D_t)$ where $V_t = \cup_{i=1}^{t} V_i$ is the set of unlabeled nodes at time t, E_t represents the set of connections in the network and X_t is the set of structural properties of nodes. Suppose the set of labels $C_t = \{R_1, ..., R_K\}$ represents the K groups of nodes at time t.

The goal is to find the roles of nodes from their structural properties over time. We propose a dynamic ensemble clustering [10, 22] framework such that the partitioning of nodes represents their roles in the network at time t and is also consistent with the historical information of nodes in previous time steps. For finding a set of roles in a dynamic network we need an evolutionary algorithm to detect the roles consistently over time such that the clustering of C_t is derived from aggregation of $C = \{C_1, ..., C_t\}$. The clustering C_t for X_t is the one that has the minimum distance from C. We define a weighted distance for the weighted clustering ensemble method as follows:

$$Dist(C_t, C) = Dist(X_t, C_t) + \sum_{i=1}^{t-1} Dist(C_t, C_i) * \alpha_i \qquad (2)$$

where α_i is weight of C_i.

The two components of the cost function (2) measures our two main goals in evolutionary role mining. The first component assesses how well the current set of roles represents existing structure of the network and the second one measures the consistency of discovered roles across time.

In this paper we derive the clustering of nodes at time step t by optimizing equation 2. It has been shown that optimizing the unweighted version of the equation (2) is NP-complete; instead some approximate solutions are proposed [10, 22]. Following the same approach we design an approximation method to derive the structural role of nodes at time step t. We intend to find the roles of nodes at $t = T$ assuming that nodes are clustered at each time step independently, then the clustering is derived by aggregating all the clusterings from $t = 1$ to T.

The pseudo-code of our evolutionary role mining method (ERM) is given in Algorithm 1. It takes as input: 1) G_t, the dynamic graph where edges are time stamped; 2) K, number of roles to extract; 3) $wFun(C)$, a weighting function to incorporate temporal behavior in partitioning; 4) $clustAlgo(M, K)$, an algorithm to partition nodes into K roles based on the calculated M, the similarity matrix of nodes at time step t. The algorithm starts with an **initialization phase** (lines 2-6), then a **weighting function** assigns a weight parameter α_t to each clustering C_t (line 7) in order to incorporate the dynamics of the network structure. The last step is the **ensemble clustering** (lines 8-11) where the final structural roles of nodes are derived by aggregation of previous partitionings $C = \{C_1, ..., C_t\}$. In this step, the similarity matrix M is updated in each time step using a *pairwiseSimilarity* function (equation 7) and a weight parameter α_t, defined by a weighting function as in equations 3 and 6.

Algorithm 1 Evolutionary Role Mining (ERM)

```
 1: procedure ERM($G_T, K, wFun(C), clustAlgo(M, K)$)
 2:     for $t$ in $1 : T$ do
 3:         $X_t \leftarrow localProperties(G_t)$
 4:         $C_t \leftarrow kmeans(X_t, K)$
 5:         $C \leftarrow C \cup C_t$
 6:     end for
 7:     $\{\alpha_1, ..., \alpha_T\} \leftarrow wFun(C)$
 8:     for $t$ in $1 : T$ do
 9:         $M \leftarrow M + pairwiseSimilarity(G, C_t) * \alpha_t$
10:     end for
11:     $C_T \leftarrow clustAlgo(M, K)$
12:     return $C_T$
13: end procedure
```

3.1 Structural roles initialization

The first step in evolutionary role mining is to build clusters from structural properties of nodes X_t for all $t \in [1 : T]$. This clustering process is derived by applying the k-means clustering algorithm on X_t, where the euclidean distance between observations and centroids is minimized. There are many local properties characterizing nodes, for example, node centrality, node degree and number of edges in the neighborhood [9]. The selected features must be able to characterize the neighborhood as well as to distinguish the node in the neighborhood. Another important criteria to select a local property is the scalability. We select a set of features for every node i as follows:

- the normalized node degree: quantifies the linkage of node i; it is the degree of node i divided by the sum of all nodes' degree in the network.
- the normalized average degree: shows the intensity of connectivity in the neighborhood of node i; it is calculated by averaging over all degree of immediate neighbors of node i.
- the coefficient variation of the degrees of the immediate neighbors of a node (cv): characterizes the coherence of the connectivity; it the standard deviation of the degrees in the neighborhood of node i.
- the clustering coefficient: quantifies the connectivity between neighbors; it is measured as the proportion of existing connections between neighbors of node i to the number of all possible links between them [25].
- the locality index: characterizes the structure of neighbors' connectivity to rest of the network; it is the ratio of links within the neighborhood to the number of links to the nodes outside of neighborhood.

This feature vector has the advantage of measuring the connectivity of a node in its neighborhood structure and also it is fast to calculate. It has been shown that these properties can distinguish well nodes at different structural positions [8].

The selected local properties are used to measure the similarity of nodes for extracting their structural roles at each time step.

3.2 Weighting functions

The structure of networks may change over time and new roles may emerge. Therefore incorporating the temporal smoothness can improve the accuracy of extracted roles. We use two functions to model the temporal behavior of data in clustering ensemble:

Temporal weighting (TW)

This function defines the probability that historic data is still valid for learning the roles at the current time. The basic idea of this weighting is that the older the data, the less relevant it is to current data, so a lower weight is assigned to the older data. Different functions can be defined in this group but the general properties that all must hold are: 1) $0 \leqslant w_i \leqslant 1$ for all $i \in [1, t]$, 2) $\alpha_i < \alpha_j, i < j$, 3) $\sum_{i=1}^{t} \alpha_i = 1$.

We use an exponential time decaying function [7], called temporal weighting (TW) to use in our method:

$$\alpha_i = (1-\theta)^{t-i} * \theta \tag{3}$$

for $i = 1$ to T.

Data distribution (DDW)

This function measures the validity of data based on the actual similarity of historic data to the current data. In this method the older clustering that groups objects more similar to current data is more important than the recent clustering that does not. In other words, the weight of data depends on its structure instead of arrival time.

We defined data distribution weighting (DDW) function to assign weight to history of data at each snapshot relative to its similarity to the current data. We used the distance of two clusterings to define the weights. The distance of current clustering C_t and C_i is defined as the number of objects they have clustered differently [10]. The distance between two nodes u and v for two clusterings t and i is:

$$d_{u,v}(C_t, C_i) = \begin{cases} 1, \text{ if } C_t(u) = C_t(v) \text{ and } C_i(u) \neq C_i(v), \\ \text{ or } C_t(u) \neq C_t(v) \text{ and } C_i(u) = C_i(v) \\ 0, \text{ otherwise} \end{cases} \tag{4}$$

Then the distance of clusterings is measured as:

$$dist(C_t, C_i) = \sum_{u,v \in V_t} d_{u,v}(C_t, C_i) \tag{5}$$

and the weight of clustering at time i is:

$$\alpha_i = 1 - Norm(dist(C_t, C_i)) \tag{6}$$

where $Norm(dist(C_t, C_i))$ is the value of distances normalized to the interval $[0, 1]$.

We utilize TW and DDW weighting functions to calculate the similarity matrix X. The sliding window method is excluded from our experiments since it requires multiple cluster aggregations for deriving the grouping of data at each time point. In addition, this method generates several clusterings at each window that need to be corresponded.

3.3 Ensemble clustering

We are using a number of algorithms to find the ensemble clustering. We modified the hypergraph partitioning algorithm (HGPA) by Strehl and Ghosh [22] to use the weighted similarity metrics. This method re-clusters the objects using the hyper-graph built upon the clusterings. In this method the hypergraph partitioning package HMETIS [13] is used to partition the hypergraph.

We also apply two different clustering algorithms on the weighted similarity matrix derived from equation (7). Spectral clustering is usually used for graph partitioning problems where a graph-based measure is to be minimized subject to normalized cut. This algorithm clusters objects based on the eigenvectors of their similarity matrix. For the nodes and their similarity by equation (7), the graph Laplacian L is built: $L = S - W$ where S is the degree diagonal matrix of similarity graph of nodes, W is the similarity matrix of data. Then the first k eigenvectors of L are calculated. Finally the clustering is derived by applying k-means on a matrix, built from concatenation of the first k eigenvectors as columns [23].

The other algorithm for aggregating the clusterings over time is agglomerative hierarchical [12]. This algorithm initially puts all objects in individual clusters then iteratively merges pairs of clusters either until deriving the defined number of clusters or until merging all the objects into one single cluster.

Node similarity definition

All cluster ensemble methods need a similarity matrix of nodes built based on their co-clustering occurrence. The similarity matrix is a $n \times n$ matrix for n active nodes at the current time step. For two nodes u, v, if $C_i(u) = C_i(v)$ then $X_i(u, v) = 1$ and the total similarity of u, v is:

$$M(u, v) = \sum_{i=1}^{t} X_i(u, v) * \alpha_i \tag{7}$$

where α_i is the weight of clustering at time i. The value of α_i is determined by a weighting function, described in section 3.2. The intuition here is that if two nodes were clustered together in the same group in an earlier time step, the older or the more different the clustering is, the lower its importance in the similarity of two nodes at the current time. If the structure of a network changes in a way that the previous partitionings are not any longer valid, the similarity of nodes is measured regarding the current clustering.

4. EXPERIMENTS

We applied ERM on real world data sets to evaluate its performance. We used three co-authorship networks, DBLP, Genetics and Biochemistry [24], and the network of Internet routing system [16] to find evolutionary roles and demonstrate the performance of the proposed clustering.

4.1 Data

- The DBLP dataset contains the publications of the proceedings of 28 conferences related to Data Mining, Databases and Machine Learning from 1997 to 2006.
- The Genetics dataset contains articles published from 1996 to 2005 in 14 journals related to genetics and molecular biology.
- The Biochemistry dataset contains articles published from 1996 to 2005 in 5 journals related to biochemistry.
- The autonomous systems network (AS) is comprised of Internet [16] routing system, taken from SNAP network data collection[1]. We aggregated daily instances to derive monthly graphs from Nov/1997 to Sep/1998.

4.2 Results and evaluation

We defined two baselines to compare the results against. The first baseline (CL) stacks all data up to current time t to find the clustering of data. The clusters are derived by applying the k-means algorithm on the stacked matrix. This is the general approach in evolutionary clustering where all data is available. In addition, previous studies of dynamic role discovery employ this approach [6, 21]. The second baseline (CLs) clusters data at each time step independently using k-means and discard historic data to derive the roles in the current snapshot of the network.

Figure 1 illustrates the second largest connected component of the DBLP network in 2002 and the connectivity structure of the same nodes in 2003. Nodes are colored by their roles, identified by our proposed method and the CL baseline method. As we can see from the figures, roles of nodes identified by our evolutionary method in 2003 more accurately represent the actual position of nodes in the network. For example, all less connected nodes in very sparse neighborhoods are colored the same (dark yellow) in Figure 2 (b) while we can see in Figure 2 (a) the same nodes have various labels, determined by baseline method.

To compare the performance of the algorithms, we measure the snapshot cost which is the quality of clustering on the current data. We use the modularity metric proposed by Newman [19] to assess the quality of clustering. This metric evaluates the community structure in a network where a $k \times k$ matrix is built for k clusters and every element d_{ij} represents the fraction of edges that link nodes between clusters i and j and d_{ii} is the fraction of edges within cluster i. We use similarity metrics of nodes M_t to build a similarity graph where edge e_{ij} is weighted by the similarity m_{ij} between node i and j. We modify the modularity measure for weighted network of nodes' similarity by having d_{ij} representing the sum of the edges weights between two clusters, instead of the sum of number of edges originally used, and d_{ii} is the fraction of the sum of the edge weights within a cluster by the total edge weights. The modularity is calculated as follows:

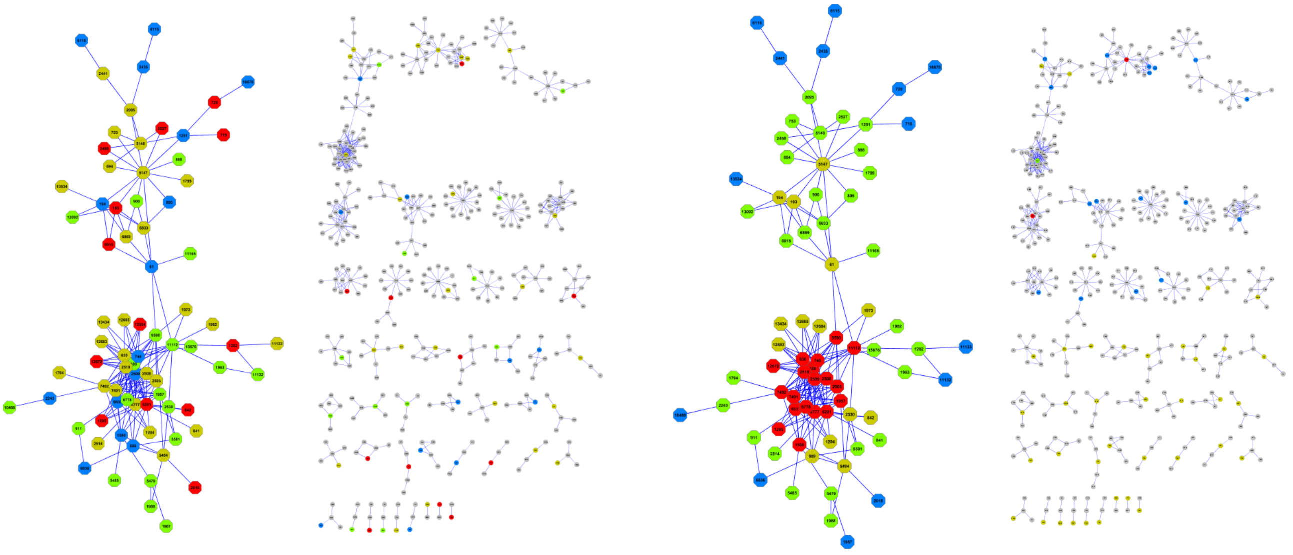
$$modularity = \sum_{i=1}^{k} (d_{ii} - \sum_{j \in 1:k, j \neq i} d_{ij}) \tag{8}$$

The main aspect of evolutionary role extraction is to increase consistency of clustering with previous time steps. We use historical cost to measure the smoothness in the transitions between time steps. The historical cost quantifies the degree to which the proposed algorithm can enhance temporal smoothness, we assess the consistency of successive clusterings by using normalized mutual information (NMI) [22].

In Figure 3 the performance of different weighting functions and algorithms on each data set is compared. Each panel demonstrates the NMI and modularity of the results on used data. For both evaluation metrics the higher values indicate a better performance.

With the NMI metric, our proposed spectral and hierarchical data weighting outperform the baseline CLs and CL for all timestamps across all three of our datasets. This shows that extracted roles by our method are more consistent over time and better shows the dynamic of network. The DDW weighting function produces better results in comparison to the temporal weighting function (TW). This function assigns more weight to the historic data that has similar clustering structure to the current data. This basically reveals that some roles may exist in a network but not at consecutive time steps, hence the network structure at the current time is more similar to older times than just the previous snapshot of the network. In other words, if the topology of a network significantly changes over time, our method utilizing DDW function can still find the structural roles of nodes with high accuracy (modularity) and consistency (NMI) including the concept drift in the structure of the network. While the two baseline methods suffer from this drawback: the CL method uses the stacked dataset which is large and is likely to contain topological structure that is not valid for current snapshot; the CLs method only considers one time step data which may not be enough for clustering.

[1] http://snap.stanford.edu/data/index.html

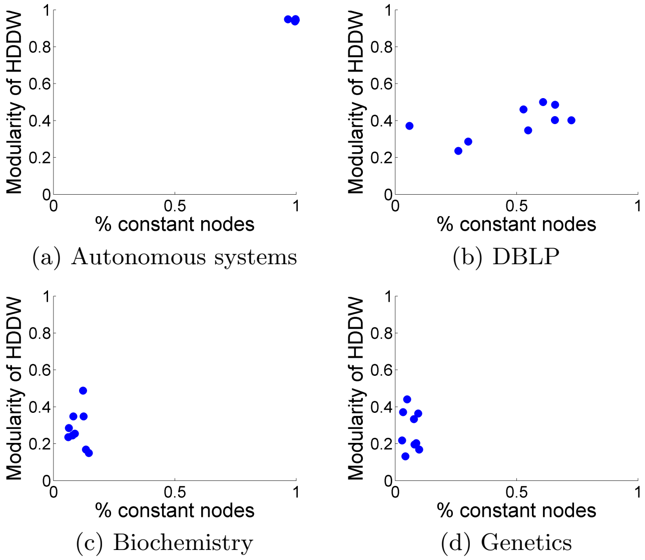


(a) Colors are determined by the CL baseline method

(b) Color-code by role of nodes, identified by proposed method

Figure 1: The second largest connected component of DBLP network in 2002 (left panel) and neighborhood of the same nodes in 2003 (right panel). The colors depict roles of node in the network, identified by baseline method and our proposed method respectively in (a) and (b). In 2002 the identified roles are almost the same by both methods but in the consecutive time step our proposed method can detect the roles of nodes more accurately and coherently.

Out of three consensus clustering algorithms, HGPA has the worst performance for either weighting functions. The two other methods, spectral and hierarchical clustering are at the same level of quality. Further investigation revealed that the main reason for poor performance of HGPA was that it produces clusters with balanced sizes since it utilizes the HMETIS algorithm [13]. This algorithm produces even sized clusters, whereas roles in a network are not equally distributed and some roles are at minority.



(a) Autonomous systems

(b) DBLP

(c) Biochemistry

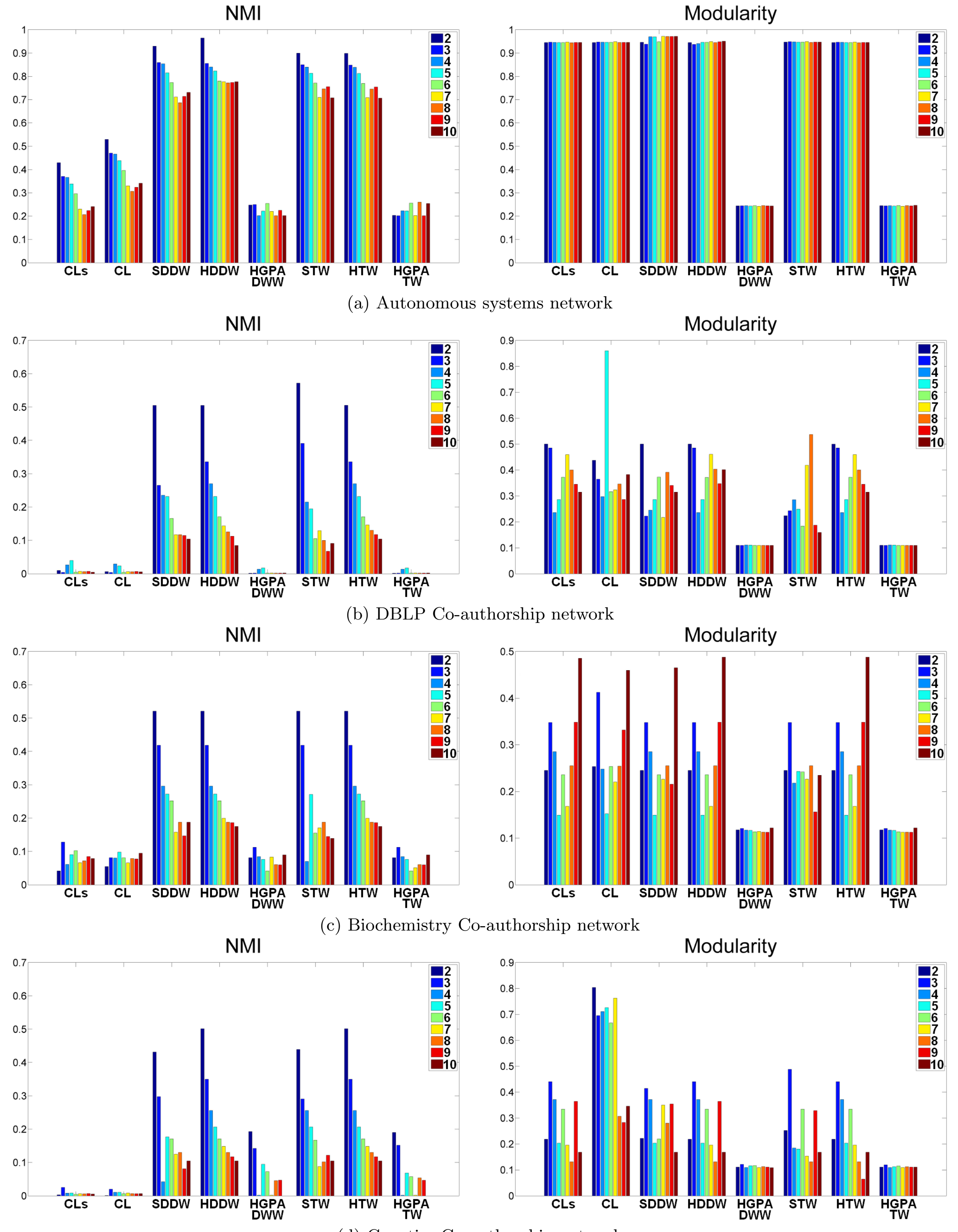
(d) Genetics

Figure 2: The modularity of hierarchical ensemble clustering using DDW weighting function versus the percentage of constant nodes at a time in different networks. The modularity drops when a large number of nodes join the network and no history of the temporal behavior of nodes is available.

The first panel of Figure 3 demonstrates the performance of baseline methods and our proposed method for AS dataset. The quality of discovered roles by our method is higher or equal to the baseline methods except at the time steps that a large number of new nodes join the network. For AS network, we have a constant number of nodes over time and at each time step the temporal behaviors of all nodes are available. As we can see from the results, our method outperforms the baselines when either spectral or hierarchical clustering is employed for ensemble clustering. Figure 2 shows the relation of percentage of constant nodes at each time step and the modularity of our method for the networks. We can see that the accuracy drops off when the percentage of constant nodes in the network decreases. At some time steps for co-authorship networks, our method has poor performance comparing to the baseline methods in terms of modularity. By examining the growth rate of the networks, it shows that the performance declines when a large number of new nodes join the network. This is reasonable, since our method relies on the history of nodes to find their role as well as their current structure. Therefore for new nodes, where no historic data is available, the method can not learn the roles accurate enough.

5. CONCLUSIONS

In this paper, we presented an evolutionary clustering for role extraction in networks. Our method finds the structural role of nodes regarding their current position in the



(a) Autonomous systems network

(b) DBLP Co-authorship network

(c) Biochemistry Co-authorship network

(d) Genetics Co-authorship network

Figure 3: The performance of different methods in terms of NMI and modularity for the networks. CL and CLs: the two baseline methods, SDDW, STW, HDDW, HTW, HGPA DDW, HGPA TW: are respectively combination of spectral clustering, hierarchical clustering and HGPA clustering with data distribution (DDW) or temporal (TW) weighting functions.

network and their historic data. The role set of nodes at each time step is the one that minimizes the defined cost function for evolutionary clustering, taking into account the current snapshot and the historic cost. We use ensemble clustering and nodes at each time step are clustered by aggregating all the available partitionings of data in previous time steps. We also use a weighting function to incorporate temporal smoothness We conducted an empirical evaluation using normalized mutual information (NMI) and modularity metrics to demonstrate the performance of our method in capturing evolutionary roles in networks. The modularity assess how well roles fit to the current structure of network and NMI metrics evaluate the closeness of current role to previous roles of nodes. The evaluation results on real world networks shows that spectral clustering and hierarchical clustering algorithms outperform HGPA method and have better performance than the baseline approaches as well. In addition, we defined DDW weighting function based on network structure to incorporate temporal aspect of network in role discovery. We showed that this function can better explore evolutionary roles in a network, when comparing to the temporal weighting function.

Acknowledgments This work is supported by ERDF through the COMPETE Programme within project POCI-01-0145-FEDER-006961, by national Funds through FCT within projects Reminds/ UTAP-ICDT/EEI-CTP/0022/2014 and UID/EEA/50014/2013. The work is also within FourEyes, a research line within project “TEC4Growth/NORTE-01-0145-FEDER-000020”.

6. REFERENCES

[1] S. Asur, D. Ucar, and S. Parthasarathy. An ensemble framework for clustering protein–protein interaction networks. *Bioinformatics*, 23(13), 2007.

[2] S. Bhagat, G. Cormode, and S. Muthukrishnan. Node classification in social networks. In *Social network data analytics*. Springer, 2011.

[3] D. Chakrabarti, R. Kumar, and A. Tomkins. Evolutionary clustering. In *Proc. ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, Philadelphia, USA, 2006.

[4] Y. Chi, X. Song, D. Zhou, K. Hino, and B. Tseng. Evolutionary spectral clustering by incorporating temporal smoothness. In *Proc. ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, San Jose, CA, USA, 2007.

[5] S. Choobdar, P. Ribeiro, and F. Silva. Event detection in evolving networks. In *Proc. IEEE Int. Conf. on Computational Aspects of Social Networks (CASoN)*, São Carlos, Brazil, 2012.

[6] S. Choobdar, F. Silva, and P. Ribeiro. Network node label acquisition and tracking. In *Proc. Portuguese Conf. on Artificial Intelligence, Progress in Artificial Intelligence*, 2011.

[7] C. Cortes, D. Pregibon, and C. Volinsky. *Communities of interest*. Springer, 2001.

[8] L. Costa, F. Rodrigues, C. Hilgetag, and M. Kaiser. Beyond the average: detecting global singular nodes from local features in complex networks. *Europhysics Letters (EPL)*, 87(1), 2009.

[9] L. d. F. Costa, F. A. Rodrigues, G. Travieso, and P. V. Boas. Characterization of complex networks: A survey of measurements. *Advances in Physics*, 56(1), 2007.

[10] A. Gionis, H. Mannila, and P. Tsaparas. Clustering aggregation. In *Proc. IEEE Int. Conf. on Data Engineering*, 2005.

[11] K. Henderson, B. Gallagher, T. Eliassi-Rad, H. Tong, S. Basu, L. Akoglu, D. Koutra, C. Faloutsos, and L. Li. Rolx: Structural role extraction & mining in large graphs. In *Proc. ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, China, 2012.

[12] S. C. Johnson. Hierarchical clustering schemes. *Psychometrika*, 32(3), 1967.

[13] G. Karypis, R. Aggarwal, V. Kumar, and S. Shekhar. Multilevel hypergraph partitioning: Application in vlsi domain. In *Proc. of the 34th annual Design Automation Conference*. ACM, 1997.

[14] A. Lancichinetti and S. Fortunato. Consensus clustering in complex networks. *Scientific Reports*, 2(336), 2012.

[15] J. Leskovec, J. Kleinberg, and C. Faloutsos. Graphs over time: densification laws, shrinking diameters and possible explanations. In *Proc. ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, 2005.

[16] J. Leskovec, J. Kleinberg, and C. Faloutsos. Graphs over time: densification laws, shrinking diameters and possible explanations. In *Proc. ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, Chicago, IL, USA, 2005.

[17] D. Liben-Nowell and J. Kleinberg. The link-prediction problem for social networks. *Journal American Society for Information Science and Technology*, 58(7), 2007.

[18] Y.-R. Lin, Y. Chi, S. Zhu, H. Sundaram, and B. L. Tseng. Facetnet: a framework for analyzing communities and their evolutions in dynamic networks. In *Proc. ACM Int. Conf. on World Wide Web*, 2008.

[19] M. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical review E*, 69(2), 2004.

[20] D. M. Romero, C. Tan, and J. Ugander. On the interplay between social and topical structure. In *ICWSM*, 2013.

[21] R. Rossi, B. Gallagher, J. Neville, and K. Henderson. Role-dynamics: fast mining of large dynamic networks. In *Proc. ACM Int. Conf. on World Wide Web*, Lyon, France, 2012.

[22] A. Strehl and J. Ghosh. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *The Journal of Machine Learning Research*, 3, 2003.

[23] U. Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4), 2007.

[24] C. Wang, V. Satuluri, and S. Parthasarathy. Local probabilistic models for link prediction. In *Proc. IEEE Int. Conf. on Data Mining*, Omaha, NE, USA, 2007.

[25] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684), June 1998.

[26] Y. Zhao, G. Wang, P. S. Yu, S. Liu, and S. Zhang. Inferring social roles and statuses in social networks. In *Proc. ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, 2013.