Predicting the Evolution of Communities in Social Networks Using Structural and Temporal Features

Maria Evangelia G. Pavlopoulou*, Grigorios Tzortzis†, Dimitrios Vogiatzis‡ and George Paliouras†
*Department of Informatics and Telecommunications
National and Kapodistrian University of Athens, Athens, Greece
Email: mary18pav@gmail.com
†Institute of Informatics and Telecommunications
NCSR "Demokritos", Athens, Greece
Email: {gtzortzi, paliourg}@iit.demokritos.gr
‡Institute of Informatics and Telecommunications
NCSR "Demokritos", Athens, Greece
& The American College of Greece, Deree, Athens, Greece
Email: dimitrv@iit.demokritos.gr

***Abstract*—During the last years, there is increasing interest in analyzing social networks and modeling their dynamics at different scales. This work focuses on predicting the future form of communities, which represent the mesoscale structure of networks, while the communities arise as a result of user interaction. We employ several structural and temporal features to represent communities, along with their past form, that are used to formulate a supervised learning task to predict whether a community will continue as currently is, shrink, grow or completely disappear. To test our methodology, we created a real-life social network dataset consisting of an excerpt of posts from the Mathematics Stack Exchange Q&A site. In the experiments, special care is taken in handling the class imbalance in the dataset and in investigating how the past evolutions of a community affect predictions.**

I. Introduction

Social networks evolve over time as a result of the activity of their users. New users join the network, old ones cease to be active or depart, while edges representing user interaction can be created, destroyed or exhibit a complex intermittent behaviour, giving rise to a dynamic network. Predicting the future form of a social network presents an interesting challenge with numerous applications, such as in marketing to locate appropriate groups of users on which to target advertisements, criminology to identify growing cliques of delinquent individuals that require immediate attention and journalism to uncover developing stories.

One of the first prediction problems to be investigated in the context of social networks was edge prediction. Edge prediction refers to predicting whether an interaction (edge) will occur between two users of the network [1]–[3]. A related problem is that of edge sign prediction, where the goal is to infer whether an interaction between two users has a positive or negative context [4]–[6]. Communities represent the mesoscale structure of the social network and are implicitly formed as users with the same interests closely interact. As the interests of users change over time so do the communities, which may reduce or increase in size, or, even completely disappear from the network. Community evolution prediction concerns the prediction of the future form of a community given its present and past form and has been a hot research topic lately [7]–[10].

978-1-5386-0756-5/17/$31.00 ©2017 IEEE

In this work we focus on four popular evolutionary phenomena of communities; growth, shrinkage, continuation and dissolution [7]. We present a framework for predicting these types of evolution that covers all the necessary steps involved, including the preprocesing of the data, the detection and tracking of the communities, the extraction of features to represent the communities and finally the training of a predictive model that discriminates the four evolutionary events. Particular focus is placed on employing an extensive set of structural and temporal features that capture various characteristics of the communities in order to get accurate predictions. To test the proposed framework, experiments are performed on a real-life social network dataset obtained from the Mathematics Stack Exchange Q&A site. Results confirm the efficacy of our framework and the importance of using a mixture of structural and temporal features.

The rest of paper is organized as follows. In Section II we provide a review of related on work on methods for community evolution prediction. In Section III we present our framework for community evolution prediction placing special focus on the extraction of appropriate features to represent communities. Next, in Section IV we present experiments using a real-life social network. Finally, in Section V concludes this work and offers directions for future work.

II. Related Work

Various approaches have been presented in the literature to predict the evolution of communities. Brodka et al. [7] tried different classifiers to predict six evolutionary events of communities (defined as grow, shrink, continue, merge, split and dissolve). Classifiers were trained using as features the size of the communities and their evolutionary events over the last three timeframes. An extended version of this work is presented in [11], where a larger set of features and past timeframes are

used. Sequential and non-sequential classifiers were evaluated in [8] to infer four types of community evolutionary phenomena: continuation, shrinkage, growth and dissolution. Features related to the structure, content and context of communities over the past one, two or three timeframes were considered, to test how past evolutions of communities affect predictions. Ilhan and Oguducu [12] introduced a time series ARIMA model to estimate how community features values will change in future timeframes and predict six types of evolutionary events (survive, shrink, continue, merge, split and dissolve) using those feature estimates for training a classifier. Patil et al. [13] addressed a similar problem to the above, that of o predicting the stability of communities, i.e., whether a community will disappear or thrive in the future. Takaffoli et al. [14] considered five evolutionary events, namely survive, merge, split, size and cohesion. These events are treated as being non mutually exclusive and, thus, may occur together at the same time for a particular community. Hence, they learn separate models to predict each of them, using structural and temporal information about the communities. The size and cohesion evolutions are meaningful for a surviving community only, therefore a two-stage technique is employed to predict them. First, the survival of a community is decided and, if it is found to survive, the prediction for these two evolutions follows using the corresponding models.

Kairam et al. [9] distinguished between two types of growth for a community, diffusion and non-diffusion growth, and analyzed the processes which govern them. Diffusion growth occurs when a community attracts new members through ties to existing members, whereas in non-diffusion growth individuals with no prior ties become members. They generated models which exploit a communitys structure and past growth behaviour to predict its future rate of growth and longevity of growth. In [15], structural information extracted from the early stages of a community were utilized to infer the lifespan of a community using linear regression. Results indicated that there is a correlation between the lifespan of a community and its structural properties. Finally, in [10] a method was proposed that examines the structural characteristics of a social network to extract an appropriate subset of features for representing communities. Using this feature subset that is tailored to the individual network, the accuracy of predictions in community evolution tasks is improved and, also, a speedup with regards to run time is achieved.

III. Predicting Community Evolution

Communities in dynamic social networks evolve, since their members and the interactions between the members change as time passes. We consider four popular and mutually exclusive evolutionary events and present a framework for predicting them that covers all the necessary steps involved. In particular, we wish to build a predictive model that discriminates whether a community in the future will *grow* or *shrink* in size, *continue* to exist with almost no change of its current form, or *dissolve* and, thus, completely disappear from the network. Our framework consists of the following steps:

1) Segment the social network data into timeframes.
2) Detect the communities in each timeframe.
3) Track communities across time to identify their evolution and corresponding evolutionary events.
4) Compute structural and temporal community features.
5) Train a classifier to predict community evolution.

Below we analyze in detail each of these steps.

A. Segmentation into Timeframes

Data acquired from social networks are timestamped and come in the form of data streams. To handle the continuous time dimension of the data stream, we discretize it into a predefined number of time-ordered timeframes $F_t, t = 1, \ldots, T$. Data is assigned to timeframes according to its timestamp, such that each timeframe contains the same number of elements (e.g., user posts from a social network). Consecutive timeframes are allowed to overlap, with an overlap $O \in [0, 1]$, in order to have a smoother transition between them and, thus, better monitor the evolution of communities, as suggested in [16]. The amount of overlap designates the percentage of the previous timeframe that is also part of the next timeframe, as shown in the examples of Figure 1.

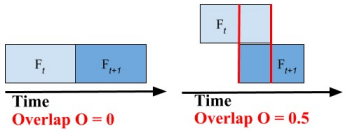

Fig. 1. Examples of overlap between two timeframes.

B. Community Detection

Having segmented a social network dataset into timeframes, the next step is to independently detect the communities in each timeframe. We model the social network as a sequence of undirected graphs $\{G_1, G_2, \ldots, G_T\}$, where $G_t = (V_t, E_t)$ denotes the graph of timeframe F_t with vertex set V_t, $n(F_t) = |V_t|$, and edge set E_t, $m(F_t) = |E_t|$. Each user of the social network in a particular timeframe is represented by a node in the timeframe graph, and there is an edge between two nodes if an interaction between the corresponding users occurs in this timeframe (e.g., one responds to the other's post).

A community corresponds to a densely connected subset of users (i.e., a subgraph) of the timeframe graph that is loosely connected to the rest of the graph. Any graph clustering algorithm can be employed to uncover the communities in social networks. One popular choice is the Louvain algorithm [17], which optimizes the modularity measure and scales well to large networks. We use the set $C_t = \{C_t^1, C_t^2, ..., C_t^{K_t}\}$ to denote the communities detected at timeframe F_t. Each community $C_t^k \in C_t$ is represented by a graph $G_t^k = (V_t^k, E_t^k)$, which is a subgraph of G_t with vertex set $V_t^k \subset V_t$, $n(C_t^k) = |V_t^k|$, and edge set $E_t^k = \{(u, v) \in E_t : u, v \in V_t^k\}$, $m(C_t^k) = |E_t^k|$.

C. Community Tracking

A community in a timeframe may be matched to a community in a following (not necessarily consecutive) timeframe, in the sense that the latter is the evolution of the first. The instances of the same community at different timeframes form what is called a dynamic community. Formally, a dynamic community is defined as a sequence of matched communities $M = \{C_{t_1}^{k_1}, ..., C_{t_p}^{k_p}, ..., C_{t_m}^{k_m}\}$, where $1 \leq t_1 < t_2 < \ldots <$

$t_m \leq T$, $1 \leq k_j \leq K_{t_j}$ $j = 1, \ldots, m$. For a community $C_{t_p}^{k_p} \in M$, its past instances, $C_{t_j}^{k_j}, j < p$, are referred to as the ancestors of the community.

Given the communities in each timeframe, community tracking algorithms that find matching communities based on a similarity measure, such as GED [7], can be employed to locate the dynamic communities arising in the dataset. These algorithms also assign a label to each community of the dynamic community sequence that describes the type of evolution which took place as the community evolved (e.g., continue, grow, shrink, dissolve[1]). These labels serve as the ground-truth for training a model to perform community evolution prediction. Note that some social networks provide annotations which allow to readily track communities, without needing the intervention of tracking algorithms. Such a case is described in our experiments with the Mathematics Stack Exchange social network.

D. Community Feature Engineering

To attain an informative representation of communities that captures a variety of their properties, we employ a comprehensive set of structural and temporal features, aiming to predict community evolution accurately. Structural features capture different aspects of the community graph, while temporal features capture characteristics of the evolution of the community by extracting information from its ancestors. On the following, we analytically present those features.

1) Structural Features:

- **Relative Size** [8] is the normalized value of community's C_t^k size in timeframe F_t:

$$RS(C_t^k) = n(C_t^k) \,/ n(F_t) \tag{1}$$

- **Relative Edges Number** is the normalized value of edges belonging to community C_t^k in timeframe F_t:

$$RE(C_t^k) = m(C_t^k) \,/ m(F_t) \tag{2}$$

- **Density** [8] is the ratio of the actual edges of community C_t^k to the maximum number of edges the community could have:

$$D(C_t^k) = \frac{m(C_t^k)}{n(C_t^k)(n(C_t^k)-1)/2} \tag{3}$$

- **Cohesion** [8] is the product between the density and the inverse fraction of edges (out of all possible edges) pointing outside of community C_t^k:

$$Ch(C_t^k) = D(C_t^k)\frac{n(C_t^k)(n(F_t) - n(C_t^k))}{m_{out}(C_t^k)} \quad , \tag{4}$$

  where $m_{out}(C_t^k) = |\{(u,v) \in E_t : u \in V_t^k, v \notin V_t^k\}|$.
- **Ratio Association** [18] is the average internal degree of a community's members:

$$RA(C_t^k) = 2m(C_t^k) \,/ n(C_t^k) \tag{5}$$

- **Ratio Cut** [18] is the average external degree of a community's members:

$$RC(C_t^k) = m_{out}(C_t^k) \,/ n(C_t^k) \tag{6}$$

- **Normalized Cut** [18] measures the edge volume that points outside of the community:

$$NC(C_t^k) = m_{out}(C_t^k) \,/ (2m(C_t^k) + m_{out}(C_t^k)) \tag{7}$$

- **Average Path Length** of community C_t^k is the average path length on the community's graph G_t^k, as defined in graph theory.
- **Diameter** of community C_t^k is the diameter of the community's graph G_t^k.
- **Clustering Coefficient** [14] of community C_t^k shows how often, on average, the neighbours of a node of the community are also connected to each other, based on the community graph G_t^k.
- **Centrality** measures capture how central (i.e., centre of importance) each node (i.e., user) of a community C_t^k is. We use three centrality measures as features, namely **closeness**, **betweenness** and **eigenvector** centrality [19], [20]. Closeness centrality shows how close a node is to other nodes in the community graph G_t^k, in terms of the edges that must be traversed to reach the other nodes. Betweenness centrality measures the number of shortest paths a node lies on. Hence, it shows the importance of the node in controlling the communication between other nodes of the community. Finally, eigenvector centrality reflects the idea that a node is more central, if it is connected to central nodes. As centrality measures are defined on a per node basis, we take the average over all nodes to calculate the centrality of the entire community.

2) Temporal Features: To present these features, we will use as reference a dynamic community $M = \{C_{t_1}^{k_1}, \ldots, C_{t_p}^{k_p}, \ldots, C_{t_m}^{k_m}\}$, as defined in Section III-C, and assume that we want to compute the temporal features of community $C_{t_p}^{k_p}$ using its n most recent ancestors in time. We shall split temporal features into three groups.

- **Structural features and evolutionary events of ancestors**: The structural features (described above) of the n ancestor communities, as well as the evolutionary events assigned to them through tracking, form our first group of temporal features.

The temporal features belonging to the second group are defined between pairs of communities and depict how a community has evolved compared to its previous instance in time. We calculate these features between the following pairs of communities from M when we want to represent community $C_{t_p}^{k_p}$ using n ancestors: $(C_{t_{p-n}}^{k_{p-n}}, C_{t_{p-n+1}}^{k_{p-n+1}}), \ldots, (C_{t_{p-1}}^{k_{p-1}}, C_{t_p}^{k_p})$. These features are:

- **Jaccard Coefficient** is the fraction of members that are common in both instances of the community:

$$JC(C_{t_i}^{k_i}, C_{t_{i-1}}^{k_{i-1}}) = \frac{|V_{t_i}^{k_i} \cap V_{t_{i-1}}^{k_{i-1}}|}{|V_{t_i}^{k_i} \cup V_{t_{i-1}}^{k_{i-1}}|} \tag{8}$$

[1]A community dissolves when it is the last community of the dynamic community sequence, hence it was not matched to a community of a subsequent timeframe.

- **Join Nodes Ratio** [14] is the percentage of new members joining the community compared to its previous instance:

$$JNR(C_{t_i}^{k_i}, C_{t_{i-1}}^{k_{i-1}}) = |V_{t_i}^{k_i} \setminus V_{t_{i-1}}^{k_{i-1}}| \Big/ |V_{t_i}^{k_i}| \quad (9)$$

- **Left Nodes Ratio** [14] is the percentage of members leaving the community compared to its previous instance:

$$LNR(C_{t_i}^{k_i}, C_{t_{i-1}}^{k_{i-1}}) = |V_{t_{i-1}}^{k_{i-1}} \setminus V_{t_i}^{k_i}| \Big/ |V_{t_{i-1}}^{k_{i-1}}| \quad (10)$$

- **Activeness** [12] measures the new edges per node that a community contains compared to its previous instance:

$$Act(C_{t_i}^{k_i}, C_{t_{i-1}}^{k_{i-1}}) = |E_{t_i}^{k_i} \setminus E_{t_{i-1}}^{k_{i-1}}| \Big/ |V_{t_i}^{k_i}| \quad (11)$$

The temporal features belonging to the third group are defined for individual communities instead of pairs. When representing community $C_{t_p}^{k_p}$ using n ancestors, we calculate these features for $C_{t_p}^{k_p}$ and its n ancestors. These features are:

- **Lifespan** [14] of community $C_{t_w}^{k_w}$ is the ratio of the ancestors the community has based on the corresponding dynamic community, to the maximum number of ancestors it could have. Obviously the maximum number of ancestors equals t_w.
- **Aging** [12] of community $C_{t_w}^{k_w}$ is the average age of the community members. The age of a member is increased by 1 every time it is found to be also a member of an ancestor community of $C_{t_w}^{k_w}$ in the corresponding dynamic community. Aging is normalized by dividing with the maximum possible age of members, which equals w.

E. Learning a Predictive Model for Community Evolution

Community evolution prediction is formulated as a classification problem, where the aim is to train a classifier that distinguishes between four types of evolutionary events (i.e., classes). These events are: continuation, shrinkage, growth and dissolution. The instances for training the classifier are the communities that have been extracted from the social network timeframes, which are represented with vectors comprising of the structural and temporal features described above, along with their corresponding class label obtained through tracking. Any classifier that can handle instances in vectorial form can be applied to learn the predictive model. In our experiments we use Support Vector Machines (SVMs) as the underlying classifier.

IV. Experimental Evaluation

To test the applicability of our framework in practice and investigate the efficacy of the presented features in predicting community evolution, we perform experiments over a dataset acquired from the Mathematics Stack Exchange Q&A site[2], a real-life social network. Mathematics Stack Exchange is a question and answer site for people studying math, where users post questions, answer questions posted by other users and comment on the users' posts. All questions are tagged with their subject areas (i.e., topics), while answers and comments inherit the topics of the question they correspond to. Our dataset contains 376030 timestamped posts under various topics, published between 2009 and 2013.

To conduct our experiments we split the dataset into 10 timeframes containing an equal number of posts, set the overlap of timeframes to $O = 0.6$ and set out to detect and track the communities to obtain the instances that will be used to train a support vector machine (SVM) classifier for predicting the evolution of communities.

Each timeframe is modeled with an undirected graph where every different user who posted a question, answer or comment in this timeframe is a node of the graph and an edge is added between two users if one posts an answer or comment to respond to the other's post in the timeframe. To detect the communities in each timeframe, we take advantage of the topics associated with posts in Mathematics Stack Exchange and do not employ a community detection algorithm. Specifically, we consider that users belong in the same community if they make posts (questions, answers or comments) about the same topic. Hence, each community is associated with a particular topic. Note that communities that contain less than four members are considered as artifact communities and are ignored in our experiments.

Tracking the communities across time to obtain the dynamic communities and their evolutionary events is also done by utilizing the topics. For each detected community C_t^k in timeframe F_t, we obtain the topic associated with the community and look for a matching community with the same topic in a subsequent timeframe $F_{t'}, t' > t$. Timeframes are processed sequentially, stopping at the first timeframe a match is found (i.e. a community with the same topic). If the topic is not found in any of the following timeframes (i.e., no matching community is found), then we set the evolutionary event of C_t^k as *dissolution*. Otherwise, a match is found to a community $C_{t'}^{k'}, t' > t$ (i.e. C_t^k is the most recent ancestor of $C_{t'}^{k'}$) and the evolutionary event is set as:

- If $n(C_t^k) - n(C_{t'}^{k'}) > th$, we set the evolutionary event of community C_t^k as *shrinkage*.
- Else if $n(C_{t'}^{k'}) - n(C_t^k) > th$, we set the evolutionary event of community C_t^k as *growth*.
- Else, we set the evolutionary event of community C_t^k as *continuation*.

We call $th > 0$ the event threshold. This threshold determines how much different the sizes of two matching communities should be in order to decide that a significant difference exists and thus label the evolution as growth or shrinkage. Note that by defining different values for the event threshold, we obtain a different ground-truth for the dataset. Figure 2 illustrates the number of the different evolutionary events in our dataset as we vary the threshold value. Notice the imbalance that exists in the dataset in terms of the different types of evolutionary events for all th values.

Having obtained the communities along with their labels, as described above, we compute their structural and temporal features (Section III-D) to form the dataset for training and testing an SVM classifier. The SVM implementation available

[2]http://math.stackexchange.com/

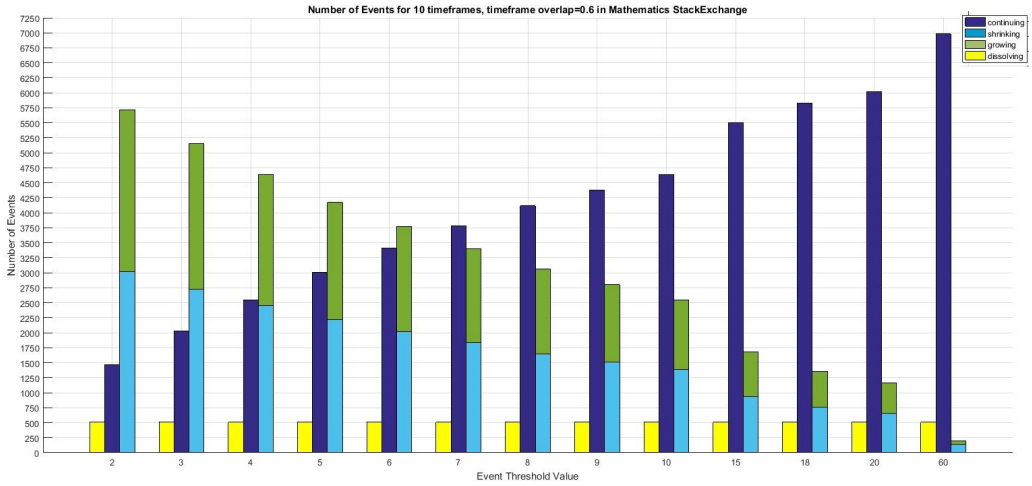

Fig. 2. Number of community evolutionary events for different values of event threshold th.

in Weka[3] with RBF kernel is used for our experiments. To train and test the classifier, we employ a variant of the popular k-fold cross validation technique that is appropriate for timestamped data, called time series cross validation. In this variant, folds correspond to timeframes and each fold is once used as the test set and performance is averaged over all folds. When the i-th fold is used as the test set, only the first $i-1$ folds are used in the training set, thus time series cross validation ensures that the instances of the training set precede in time those of the test set, respecting the natural ordering of the data.

In all experiments we apply the SMOTE oversampling technique and the spreadsubsample undersampling technique of Weka to counter the imbalance that exists in our dataset (Figure 2). The experiment results obtained without the use of oversampling and undersampling techniques were inferior to the ones presented below, and thus were omitted due to space limitation. Moreover, we measure classification performance using the popular F1 score and report the per-class performance, as well as the performance over all classes using the macro-F1 score which is suitable for evaluating imbalanced datasets.

We aim at investigating whether the addition of the temporal features on top of the structural features improves predictions and studying how the number n of ancestors considered when computing the temporal features affects predictions. To examine this we perform experiments for $n \in \{0, 2, 4, 6\}$ and use i) only the structural features and the evolutionary events of the ancestors as temporal features and ii) the complete set of temporal features. When $n = 0$ no ancestors are considered, hence only the structural features are used to represent the communities. For each value of n we try $th \in \{2, 3, 4, 5, 6\}$ and optimize the internal parameters of the SVM classifier using grid search and report the best performance. Results are shown in Tables I and II and were obtained for $th = 6$ for all tried values of n. It is evident that temporal features help in improving performance. Also the use of 2 or 4 ancestors seems to be beneficial while for 6 ancestors performance degrades, indicating that going too far back time is not helpful. It is noted that F1 scores are rather low in these experiments and this occurs because low th values were used, resulting in low quality ground truth, as shown below.

[3]http://www.cs.waikato.ac.nz/ml/weka/

TABLE I. RESULTS IN TERMS OF F1 SCORE WHEN ONLY THE STRUCTURAL FEATURES AND EVOLUTIONARY EVENTS OF ANCESTORS ARE USED AS TEMPORAL FEATURES.

Ancestors	Continue	Shrink	Grow	Dissolve	Overall
0	0.4783	0.5644	0.1526	0.3587	0.4694
2	0.4683	0.5642	0.4282	0.4333	0.5072
4	0.4726	0.6238	0.3874	0.4263	**0.5105**
6	0.3783	0.6805	0.3257	0.4415	0.4785

TABLE II. RESULTS IN TERMS OF F1 SCORE WHEN ALL TEMPORAL FEATURES ARE INCLUDED.

Ancestors	Continue	Shrink	Grow	Dissolve	Overall
0	0.4783	0.5644	0.1526	0.3587	0.4694
2	0.5444	0.6652	0.4202	0.5762	**0.5720**
4	0.5403	0.7123	0.3884	0.5095	0.5475
6	0.4812	0.7152	0.3292	0.4857	0.5581

Having shown that the temporal features improve predictions we perform a more detailed experiment to examine if the outright performance scores can be improved. Specifically, we experiment with a greater range of event threshold values, including large ones, $th \in \{5, 10, 15, 20, 25, 30, 60\}$. When assigning evolutionary events using higher values for event threshold, we become more strict while deciding whether a community has grown or shrunk, since the difference in size between two matched communities must be larger in order to assign these labels. Hence, more communities are labeled as continuing as the event threshold increases. Although a higher threshold increases imbalance (Figure 2), it may still lead to better performance if the underlying ground-truth is of higher quality. Results over all classes are reported in Table III, where also the macro recall and macro precision are shown. It is evident that performance has considerably increased in these experiments and for all number of ancestors tried the best results were obtained for $th = 30$. This shows that a ground-truth of higher quality is constructed in this case allowing

TABLE III. RESULTS WHEN ALL TEMPORAL FEATURES ARE INCLUDED AND AN EXTENDED SET OF EVENT THRESHOLD VALUES IS USED.

Ancestors	Macro F1	Macro Recall	Macro Precision
0	0.6731	0.6545	0.6928
2	0.7662	0.7572	0.7754
4	**0.7719**	0.7608	0.7835
6	0.7194	0.7132	0.7259

for a more accurate prediction of community evolution. In accordance with the previous experiment, we observe that performance increases as we move from smaller to greater values for the number of ancestors, until we use 6 ancestors, at which point performance declines.

Overall, our experiments have shown that structural features when combined with temporal features improve prediction accuracy and that using some, but not too many, ancestors to compute temporal features is also beneficial.

V. CONCLUSIONS

This work aimed at predicting the evolution of communities that are formed in social networks as a result of user interaction, using a mixture of structural and temporal features. Four types of evolution that commonly arise in social networks were examined, namely the continuation, growth, shrinkage and dissolution of communities. We presented a framework that incorporates all necessary steps for building a predictive model to infer community evolution. These steps are: segmentation into timeframes, detection and tracking of communities, calculation of communities' features and classifier training. We performed experiments using real-life social network data acquired from the Mathematics Stack Exchange Q&A site. Experiments demonstrated that prediction accuracy improves when temporal features are used on top of the structural ones. Also, the extent of past evolutions of a community considered (i.e., the number of ancestors) affects predictions and using four ancestors gave the best results in our dataset. It seems that the past of a community encapsulates information about its future evolution and can help in improving predictions, if we do not go too far back in time.

Future work will focus on the prediction of other types of community evolution, such as merges and splits where there is no one-to-one correspondence between communities as they evolve. The incorporation of other types of features in order to improve predictions, such as features derived from the text posted by social network users (e.g., topics of discussion and sentiment) and features related to the context of a particular social network (e.g., reputation in the Mathematics Stack Exchange site and hashtags in Twitter), could be also examined. In addition, using other classifiers, apart from SVMs, for predictions and performing tests with more datasets, as well as comparing our approach to existing ones from the literature, such as [14], is in our plans. Moreover, finding the optimal timeframes for splitting the data stream of a social network poses an interesting problem itself. Such optimal timeframes would contribute in a more accurate detection of communities and subsequently their tracking and prediction. Finally, using first-order logic to capture the knowledge governing community evolution and applying Markov Logic Networks [21] to predict the evolution of communities, is another interesting research direction.

REFERENCES

[1] D. Liben-Nowell and J. Kleinberg, "The link-prediction problem for social networks," *Journal of the American Society for Information Science and Technology*, vol. 58, no. 7, pp. 1019–1031, 2007.

[2] R. N. Lichtenwalter, J. T. Lussier, and N. V. Chawla, "New perspectives and methods in link prediction," in *ACM International Conference on Knowledge Discovery and Data Mining*, 2010, pp. 243–252.

[3] E. Zheleva, L. Getoor, J. Golbeck, and U. Kuter, "Using friendship ties and family circles for link prediction," in *International Workshop on Advances in Social Network Mining and Analysis*, 2008, pp. 97–113.

[4] P. Symeonidis, E. Tiakas, and Y. Manolopoulos, "Transitive node similarity for link prediction in social networks with positive and negative links," in *ACM on Recommender Systems*, 2010, pp. 183–190.

[5] J. Leskovec, D. Huttenlocher, and J. Kleinberg, "Predicting positive and negative links in online social networks," in *International Conference on World Wide Web*, 2010, pp. 641–650.

[6] J. Kunegis, A. Lommatzsch, and C. Bauckhage, "The slashdot zoo: Mining a social network with negative edges," in *International Conference on World Wide Web*, 2009, pp. 741–750.

[7] P. Brodka, P. Kazienko, and B. Koloszczyk, "Predicting group evolution in the social network," in *International Conference on Social Informatics*, 2012, pp. 54–67.

[8] G. Diakidis, D. Karna, D. Fasarakis-Hilliard, D. Vogiatzis, and G. Paliouras, "Predicting the evolution of communities in social networks," in *International Conference on Web Intelligence, Mining and Semantics*, 2015, pp. 1–6.

[9] S. R. Kairam, D. J. Wang, and J. Leskovec, "The life and death of online groups: Predicting group growth and longevity," in *ACM International Conference on Web Search and Data Mining*, 2012, pp. 673–682.

[10] N. Ilhan and S. G. Ögüdücü, "Feature identification for predicting community evolution in dynamic social networks," *Engineering Applications of Artificial Intelligence*, vol. 55, pp. 202–218, 2016.

[11] S. Saganowski, B. Gliwa, P. Brdka, A. Zygmunt, P. Kazienko, and J. Kolak, "Predicting community evolution in social networks," *Entropy*, vol. 17, no. 5, pp. 3053–3096, 2015.

[12] N. İlhan and c. G. Öğüdücü, "Predicting community evolution based on time series modeling," in *IEEE/ACM Int. Conf. on Advances in Social Networks Analysis and Mining*, 2015, pp. 1509–1516.

[13] A. Patil, J. Liu, and J. Gao, "Predicting group stability in online social networks," in *International Conference on World Wide Web*, 2013, pp. 1021–1030.

[14] M. Takaffoli, R. Rabbany, and O. R. Zaane, "Community evolution prediction in dynamic social networks," in *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, 2014, pp. 9–16.

[15] M. K. Goldberg, M. Magdon-Ismail, S. Nambirajan, and J. Thompson, "Tracking and predicting evolution of social communities," in *IEEE International Conference on Social Computing*, 2011, pp. 780–783.

[16] E. Keogh, S. Chu, D. Hart, and M. Pazzani, "Segmenting time series: A survey and novel approach," in *Data mining in Time Series Databases*, 2003, pp. 1–22.

[17] 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. 10, p. 10008, 2008.

[18] I. S. Dhillon, Y. Guan, and B. Kulis, "Weighted graph cuts without eigenvectors: A multilevel approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 11, pp. 1944–1957, 2007.

[19] N. Gupta, A. Singh, and H. Cherifi, "Centrality measures for networks with community structure," *Physica A: Statistical Mechanics and its Applications*, vol. 452, pp. 46 – 59, 2016.

[20] A. Abnar, "Structural role mining in social networks," Master's thesis, University of Alberta, Department of Computing Science, 2014.

[21] M. Richardson and P. M. Domingos, "Markov logic networks," *Machine Learning*, vol. 62, no. 1-2, pp. 107–136, 2006.