December 2019

SAMPLING AND CHARACTERIZING EVOLVING COMMUNITIES IN SOCIAL NETWORKS

Humphrey Appiah Mensah
Syracuse University

ABSTRACT

One of the most important structures in social networks is communities. Understanding communities is useful in many applications, such as suggesting a friend for a user in an online friendship network, recommending a product for a user in an e-commerce network, etc. However, before studying anything about communities, researchers first need to collect appropriate data. Getting complete access to the data for community studies is unrealistic in most cases. In this work, we address the problem of crawling networks to identify community structure. Firstly, we present a network sampling technique to crawl the community structure of dynamic networks when there is a limitation on the number of nodes that can be queried. The process begins by obtaining a sample for the first-time step. In subsequent time steps, the crawling process is guided by community structure discoveries made in the past. Experiments conducted on the proposed approach and certain baseline techniques reveal the proposed approach has at least a 35% performance increase in cases when the total query budget is fixed over the entire period and at least an 8% increase in cases when the query budget is fixed per time step. Secondly, we propose a sampling technique to sample communities in node attributed edge streams when there is a limit on the maximum number of nodes that can be stored. The process learns if the nodal information can characterize communities. The nodal information is leveraged with the structural information to generate representative communities. If the nodal information does not characterize communities, only structural information is considered in assigning nodes to communities. The proposed approach provides a performance improvement of up to about 5 times that of baselines. Finally, we investigate factors that characterize the evolution of communities with respect to the number of active users. We perform this investigation on the Reddit social media platform. We begin by first analyzing individual conversations of one community and sees how that generalizes to other communities. The first community studied

is Reddit's `changemyview`. The `changemyview` community, in addition to its rich data source, has an interesting property where members whose view are changed award points to users that successfully changed their minds. From the `changemyview` community, we observe that the linguistic style and interactions of members of the community can significantly differentiate susceptible and non-susceptible users. Next, we examine other communities (subreddits), and investigate how the user behaviors observed from `changemyview` relate to patterns of community evolution. We learn that the linguistic style and interactions of members in a community can also significantly differentiate the different parts of the evolution of the community with respect to number of active users.

SAMPLING AND CHARACTERIZING EVOLVING COMMUNITIES IN SOCIAL NETWORKS

By

Humphrey Appiah Mensah
B.Sc. (Hons.) in Computer Science and Engineering, University of Mines and Technology, 2013
MSc in Computer Science, Syracuse University, 2019

DISSERTATION

Submitted in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in Computer and Information Science and Engineering

Syracuse University
December 2019

ACKNOWLEDGMENTS

I will like to express my profound gratitude to my advisor, Prof. Sucheta Soundarajan. I can't imagine how I would have come this far without having you as an advisor. Thanks for your patience, guidance, encouragement and direction for the past years. I am also thankful to my dissertation committee members, Professor Jae Oh, Chilukuri Mohan, Fanxin Kong and Garrett Katz for taking time off their busy schedules to be part of my committee.

Will like to thank Professors Lu Xiao and Chilukuri Mohan for their immense advises for my dissertation.

To the members of the Syracuse University Network Science Laboratory, thanks all for your help and support during this PhD journey. It was nice spending time with all of you both in and outside the lab. To a special friend without whom I wouldn't have known anything about Syracuse University, Francis Akowuah, thanks for everything.

To my family, I am deeply grateful for the love, support, prayers and sacrifices you've all made during my study in Syracuse University. A special thanks to the woman of my life, Fafa. Thanks for all the sacrifices you've made for me and B I. You are loved!

CONTENTS

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

Social network analysis has attracted ample amount of interest in recent decades. The increasing interest in social network analysis can be credited to its power to explain social entities and their linkages as well as the patterns and implications of such linkages [94]. Additionally, the analysis of social networks provides new challenges and opportunities from the perspective of knowledge discovery and data mining [3].

Social networks can be defined more broadly than online social networking sites such as Instagram, Twitter, Facebook, WhatsApp, etc. Analyzing social networks provides better understanding of social entities, and any medium which provides a social experience in the form of user-interactions can be considered a social network [3]. For example, Reddit is not explicitly an online social networking site, but it allows for social interactions among different users and can be considered as a social network.

The social entities examined in social network analysis are referred to as actors. Examples of actors include departments within an organization, users of an online e-commerce platform, users in an online social networking site, etc. Social networks can be categorized into two broad types based on the kinds of sets of actors: (1) single mode networks (2) multi-mode networks. Single mode networks are those networks with only one sets of actors. An example single-mode could be a set of college students with a friendship relation.

Multi-mode networks are those networks that have more than one sets of actors. For example, a network of students and professors could represent a multi-mode network since there are two kinds of actors (students and professors).

According to [94], there are three kinds of notational schemes used in representing the different types of networks. These notations are:

- Graph theoretic notation
- Sociometric notation
- Algebraic notation

Graph theoretic notation represents actors of a network as nodes and the relationships existing between actors as edges. The sociometric notational scheme represents a network as a two-dimensional matrix termed sociomatrix. The entries of a sociomatrix encodes the ties between pairs of actors. Algebraic schemes are used mostly in the study of multiple relations between actors. This is because it allows to easily encode combinations of relations in networks. In this dissertation, we will be using the terms social network, network and graph interchangeably to mean the same thing.

1.1 Social network analysis: overview

There have been many works on social network analysis. In this section, we give a brief overview of some major research areas within the field.

Node classification: In the node classification problem the goal is to use a partial node labeling to predict labels of unlabeled nodes [14]. Identifying the labels of unlabeled nodes is useful in many applications. For example, if we are given a network with labels indicating people's preference for a product. If we are able to identify the labels of other users whose preferences are not known, it can help in recommending products for such people.

Link prediction: The goal of the link prediction problem is to infer unseen relations between actors in a social network [100]. For example, in a friendship network, being able to infer a link before it is formed can aid in recommending potential friends to actors.

Identifying key players: With this area of study, researchers are interested in identifying the important entities in a network [17]. These entities could be nodes, edges or subgroups. Identifying important nodes is useful in many regards. For example, identifying important nodes can help determine the nodes that are useful in the marketing of a product. Alternatively, in the marketing of a product, identifying important edges can help determine the important relations along which information can be disseminated efficiently.

Social influence analysis: This is an area of study that has been found to have wide range of applications in marketing, advertisement and recommendation. This area of research focuses on how things spread, like information [76].

Community detection: This is an area of study that aims to discover communities in networks. A community in a network can be defined as a set of nodes that have lots of connections among themselves in comparison to representatives of other communities [49, 30]. Community discovery is one of the most important problems in the context of social network analysis [3]. Lots of work has focused on identifying communities in different settings with varied constraints [98, 16, 97]. A community in a friendship network represents a group of friends with some shared interest or role. Identifying the communities in such a friendship network can aid in recommending a friend to user in the network.

Evolution in dynamic social networks: This area of research aims to understand how a network evolves over time [74]. For example, understanding how a community changes over time can provide insights such as how groups are formed, what makes a group splits, etc. Identifying these changes and the factors leading to such changes can help in strengthening the different communities in a network.

Sampling social networks: Many times during the analysis of social networks, researchers do not have access to the entire network. Sampling provides an opportunity

to obtain a subset of the actors and their interactions such that certain properties of the original network are preserved [60]. Some properties that one may wish to preserve during sampling include the community structure [12], important nodes [51], etc.

1.2 Objectives

In this dissertation, we focus on sampling communities in social networks in a variety of realistic settings that give rise to interesting constraints. Communities in different networks have various usefulness. For example, in an e-commerce network, a community can represent a group of people with similar preference for products on the market. Identifying the groups in an e-commerce network will help in recommending a user's next item of purchase. For one to study anything about communities, the appropriate network data needs to be collected.

Networks for community studies can have millions or billions of nodes and/or edges. For instance, if one wants to understand the communities in a network like Facebook, that person will require a huge amount of data which (1) might be impractical to obtain, and (2) might be rate limited by the data owner. In this work, we consider the problem of sampling communities in networks in different settings with varied constraints.

From the problem of sampling communities, we observe that different communities have different evolution behaviors over time. This leads us to investigate the various factors that characterizes the different evolution patterns of communities. We hypothesize that the number of active users in communities over time could be related to the behavior of the communities' members. For instance, if the members of a community are open minded, this indicates their willingness to engage and so such communities have its members mostly active. As a first step in understanding community evolution patterns, we look at users and their behavior in a typical community. Then we investigate how these observations generalize to the evolution of other communities.

1.3 Organization of dissertation

In Chapter 2, we provide a summary of related work. Most of these works focus on identifying communities in different settings. Most commonly, the setting considered is the static case, where the network is unchanging. More recent works have considered the dynamic setting, which considers the evolving nature of nodes and the streaming setting where edges appear as streams. Most of these works focus primarily only on the structural information. Research on crawling networks has also gained popularity in recent times. Despite the popularity of crawling networks for different objectives, there is not much on crawling networks for communities.

To begin with, we consider the problem of sampling communities in networks that change over time via crawling when there is a limitation on the number of times one can request for information (Chapter 3). We consider two resource constraint cases: (1) A limitation on the amount of information that can be requested over the entire period. (2) A limitation on the number of times one can request information for each time step. We propose `DYNSAMP` for sampling communities in both instances where there is a limitation for each time step and the case where the limitation is for the entire period of the sampling duration. `DYNSAMP` works on the notion that graphs discovered in the sampling process might be fully or partially similar to previously discovered. In both resource constraint cases, `DYNSAMP` assumes the full network is not known and a query on a node will return information about the node and its neighbors.

In Chapter 4, we consider another problem of sampling communities. As in Chapter 3, We assume that nodes in the graph have attribute information, and that the graph appears as an edge stream. In this setting, we assume there is a limitation on the maximum number of nodes that can be stored. We propose `SAMPLearn` for sampling the communities in a node attributed edge streams. The intuition behind `SAMPLearn` is to leverage the attribute information when it is found to be useful with the structural information in deciding the community membership of node. In cases where attributes are related to com-

munity structure, SAMPLearn will then consider only the structural information in deciding a node's community. When the number of nodes in the sampled graph reaches its maximum, SAMPLearn adds the newly found node to the sample replacing an unimportant node in the existing sample.

In Chapters 3 and 4, we see that different communities have different evolution styles. Consequently, in Chapter 5, we investigate the factors characterizing the evolution of the number of active users in communities. We investigate the evolution of communities using data from Reddit, a social news and discussion platform. We choose this data due to its rich content and the clarity of communities on the platform. We treat each subreddit as a "community". We believe the behavior of members of a community affect the evolution of the number of active users of the community. As a first step in investigating how user behaviors relate to community evolution, we examine a single Reddit community - changemyview. From the observations made on changemyview studies, we investigate how these user behaviors relate to the number of active users over time.

Finally, we present conclusions in Chapter 6.

1.4 Contributions

In this dissertation, we contribute to the analysis of social networks in the areas of community detection, sampling of social networks and evolution of dynamic social networks. Specifically, our contributions are as follows:

1. First, we propose DYNSAMP, a novel algorithm to sample communities in dynamic social networks. Experiments performed indicates the proposed approach outperforms other baseline techniques.

2. We propose SAMPLearn, a novel algorithm to sample communities from edge streams. The proposed approach leverages both structural and nodal information to sample communities. When nodal information does not exist or is not helpful, SAMPLearn

uses only the structural information. Communities obtained via `SAMPLearn` are shown to be closer to the true communities in comparison to other existing techniques.

3. We investigate different features characterizing susceptible and non-susceptible users - the case of Reddit's `changemyview`. This provides useful insights regarding the behavior of a user in a typical Reddit community. Investigation showed that users that change their mind all the time have unique interaction and language style.

4. We explore the different evolution patterns of communities on Reddit. From our understanding of the behavior of users in the `changemyview` community, we then characterize the identified evolution patterns. Experiments show that there exist factors such as interaction among members and language style that can characterize the different points in a community's evolution.

CHAPTER 2
RELATED WORK

In this section, we present a general related work discussion. We present a more specific related work discussion in each chapter.

Identifying communities in a network has several benefits. Blondel et. al [16] demonstrated how a community detection algorithm could identify different groups of users in a phone network. In this network, the nodes were customers of a Belgian phone company and an edges represented call made between customers. Every customer spoke at least one of the following languages: French, Dutch, English or German. One interesting observation from this network was the monoglottism of customers in a group. This suggests that considering the attributes of users in a network can improve the identification of communities. Communities in a network can also help identify the patterns of communication in an organization. In [84], authors used an email network with nodes as email addresses and edges as communication between two addresses identify the communities of practice in the network. Communities in a network are also useful in identifying different hierarchical structures [61, 20, 19, 52]

In the detection of communities, various techniques have been proposed to detect communities under different settings. For instance, there are several proposed techniques to detect communities in the static setting where there is full access to entire network

[16, 56, 89, 24]. Another setting that has also gained popularity is the detection of communities in dynamic networks. A dynamic network considers the evolving behavior of nodes in a network. The network is usually modelled as a collection of several snapshots with each snapshot representing the graph at some specific time. A recent survey by Rossetti and Cazabet discusses some of the techniques for detecting communities in dynamic social networks [67]. There have also been works in identifying communities in edge streams [33]. Most of these techniques focus on detecting the communities based on the structure of the network. Recent techniques consider both the nodal information and structural information [28, 64, 38, 72, 5]. Such works argue that combining these two sources of information leads to more meaningful communities.

There has been a significant amount of work in sampling social networks for diverse reasons. Hubler et al. proposed a technique for obtaining a representative subgraph [34]. The process begins with an initial random sample and then improves the sample with the Metropolis algorithm. Maiya and Berger-Wolf provided a sampling approach based on graph expansion to identify influential users in the graph [51]. It begins with an initial seed node and subsequently selects nodes from the neighborhood of the current sample such that the expansion of the current sample is maximized. With a similar goal of finding influential nodes, Riondato and Kornaropoulos propose an algorithm for estimating the betweenness centrality of vertices or edges in a graph [65]. Chu and Sethu [23] discussed an approach to obtain a sampled subgraph such that the largest eigenvalues of the sampled subgraph is similar to that of the true graph. The authors assume an initial graph and add nodes that have the highest estimated eigenvalue centrality.

In the literature, there has been some studies related to the growth of communities. Tan [78] proposed a framework for building genealogy graphs. The authors investigates the relationship between the origin of a community and its future growth. It was found that strong parent connections are associated with future community growth. The authors in [26] proposed a framework for tracking linguistic changes and how users react to evolving

norms of a community. It was observed that users follow a two-stage lifecycle: learning phase and conservative phase. In [25], authors propose different success measures such as growth in number of members, retention of members, long term survival of the community and volume of activities within the community. All these works on communities do not (and do not claim to) identify patterns of community growth.

CHAPTER 3
SAMPLING COMMUNITIES IN DYNAMIC SOCIAL NETOWRKS

Dynamic social networks are networks that evolve over time. A toy dynamic social network is shown in Figure 3.1. This network indicates the evolution of the network over three timesteps. It begins with two communities in timestep 1. At timestep 2, while one of the groups loose a member, the other group increases in size. At the time step 3, one of the groups in the previous timestep splits into two communities.

Researchers are interested in a wide variety of problems related to communities in dynamic social networks, including understanding their growth, dissolution, and merging behaviors [92, 83, 48]. However, before studying such questions, a researcher must first obtain an appropriate dataset. Because typical social networks may contain millions or billions of nodes, it can be a challenge to collect adequate data within a reasonable amount of time, due to both the computational efforts required to collect such data as well as API rate limits imposed by the companies owning the data. For example, when crawling the Twitter friendship or follower network, the Twitter API allows only 15 queries per 15 minutes [2]. Given such a scenario, a data collector must make the most of a limited query budget: which areas of the graph should be explored in order to obtain information that

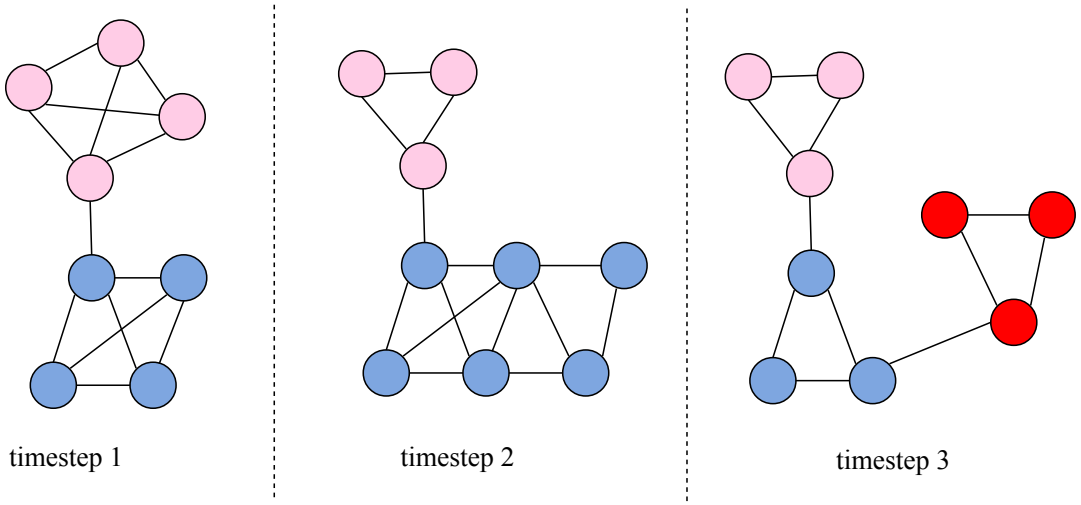


Figure 3.1: A toy dynamic network with evolution over three timesteps. The network begins with two communities at timestep 1. At timestep 2, one of the groups reduces in size while the other gets new members. The largest community in time step 2 splits into two communities at time step 3.

is most useful for the analysis task at hand? This is a challenge even in static networks; and the challenge is compounded in dynamic networks, where individual nodes or edges may appear or disappear, and the structure of entire regions of the graph may change in a moment.

In this chapter, we focus on the problem of crawling a dynamic social network with the goal of obtaining a sample with community structure that is as representative as possible of the true community structure. Here, a community in a network refers to a group of nodes that are densely connected to each other. In online social networks, a community can represent a group of likeminded users. Identifying such users could be used for marketing and recommendations [10]. Identifying the dynamic community structure in online social networks provides insights into questions such as: which group is migrating to what group, how long does it take for a particular group to collapse, when was a particular group formed, etc.

This chapter presents Dynamic Sampler (`DYNSAMP`) which samples the dynamic community structure of online social networks over a period of time when there are resource constraints. We consider two resource constraint cases: (1) The case where there is a limi-

tation on the number of times one can request information over the entire period considered (e.g., one has a total amount of money to spend on data collection across the timeline), and (2) The case when there is a limitation at each time step on the number of times one can request information about a node (e.g., there is a daily limit on the number of queries that can be made). DYNSAMP works on the notion that the current community structure of a graph might be partially or wholly similar to previously discovered community structures. Experiments show that DYNSAMP has a performance improvement ranging from 35% to 53% when compared to baseline methods when the query limitation is considered over the entire period and 8% - 56% in cases when there is a limitation at each time step.

This chapter is organized as follows. First, we present the problem of sampling communities in dynamic networks in section 3.1.2. In section 3.2, we discuss some related work. In section 3.3, we discuss the proposed approach. Section 3.4 presents the experiments performed and its set up. Finally, section 3.5 presents the conclusion to the paper and some future directions.

3.1 Preliminaries

3.1.1 Notations

- $G_t = (V_t, E_t)$ is a true, unobserved graph at time step t, where V_t and $E_t \subset V_t \times V_t$ are the set of nodes and edges, respectively, at time step t.
- $G_t^s = (V_t^s, E_t^s)$ is a sampled graph at time step t, where V_t^s and $E_t^s \subset V_t^s \times V_t^s$ are the sampled set of nodes and edges respectively at time step t.
- $G = \{G_1, G_2, ..., G_n\}$ is the true graph sequence and $G^s = \{G_1^s, G_2^s, ..., G_n^s\}$ represents a sampled graph sequence, where $G_i^s \subset G_i$.
- ω_t represents the community structure similarity metric between G_t and G_t^s at time step t.

- q^t represents the number of queries used at time step t to obtain G_t^s.
- q_v represents a vector of the number of queries made to obtain G^s. The i^{th} vector entry is the number of queries made on time step i.
- q represents the total number of queries made to obtain G^s.
- q_{max}^t represents the maximum number of queries allowed at time step t.
- q_{max} is a the total number of queries allowed over the entire timeline.
- The dynamic community similarity $\aleph$ of a sampled graph G^s and a ground truth graph G is defined as:
$$\aleph(G, G^s) = \frac{1}{n}\sum_{t=1}^{n} \omega_t$$
- τ is a dissimilarity threshold for which we declare two communities to be different.

3.1.2 Problem formulation

In this work, we assume the true graph sequence G is not known. We also assume that we can determine whether a node is present in a given timestep at no cost, as in many online social networks. For example, the Twitter API allows up to 900 queries per 15 minutes when searching for a user. In each step, a node can be queried, and all of its neighbors learned. Assuming the process begins with a query on v_1, the next query can only be made on discovered neighbors of previously queried nodes (either from the current or previous time steps). For dynamic networks, we assume there is a storage limitation on how many graphs can be stored for a period considered. Our goal is to generate a sampled graph sequence G^s such that $\aleph(G, G^s)$ is maximized.

We consider two different problem settings: (1) The query budget limits the total number of queries that can be made over the entire timeline (e.g., queries cost money, and we have a fixed amount of money for the entire sampling process). (2) There is a query limit for each timestep (e.g., queries take time, and each time step has a limited amount of time).

3.2 Related work

There has been little work focused on sampling community structure in networks, and most existing work has focused on static networks.

Maiya and Berger-Wolf [12] proposed an expander graph based sampling approach for static networks. This method begins with a seed node and increasingly grows the sample by selecting a node from the neighborhood of the current sample that maximizes a quality function. Also, in the selection of the next node, there is an assumption that the neighborhood of all nodes is known which is not generalizable to most online social networks.

In [15], the authors proposed a link tracing approach for sampling the community structure of static networks. It begins with a seed node and grows the sample by selecting the node with the highest reference score, defined as the ratio of the number of already discovered connections pointing to a node so far in the crawling process to the degree of the node.

A PageRank-based sampling approach (PRS) was proposed by Salehi et al. [70] to obtain samples from a static network with high community structures. From the simulation results, the authors argue PRS has significantly higher performance in comparison to Respondent Driven Sampling [32]. However, PRS assumes it knows the number of communities in the network which is not realistic with online social networks.

Another link tracing approach (QCA) proposed for dynamic networks is described in [57]. This begins with an initial community structure. It computes each of the existing communities' "force" of accepting the node. The community membership is selected based on the "force". QCA is able to compute community membership of discovered nodes. Even though QCA is one of the few techniques proposed for dynamic networks, it assumes it has an initial community structure which is not practical in most online social networks.

In [50], Lu et al., proposed two incremental sampling algorithms for dynamic graphs that preserves some property of interest. Even though it was demonstrated to be performing well, this approach (1) Makes a similar assumption to [12] by assuming it knows the entire

graph and (2) does not sample for communities in the network.

In this chapter, we propose a crawling based approach to sample the community structure of dynamic networks with a constraint on the number of times one can request information about a node without any knowledge of the community structure.

3.3 Proposed approach

This work proposes a novel algorithm DYNSAMP for sampling a dynamic network such that the community similarity between the true and sampled networks is maximized. The intuition behind DYNSAMP is that the current snapshot of a graph may be similar to an earlier snapshot; or if not, portions may be similar.

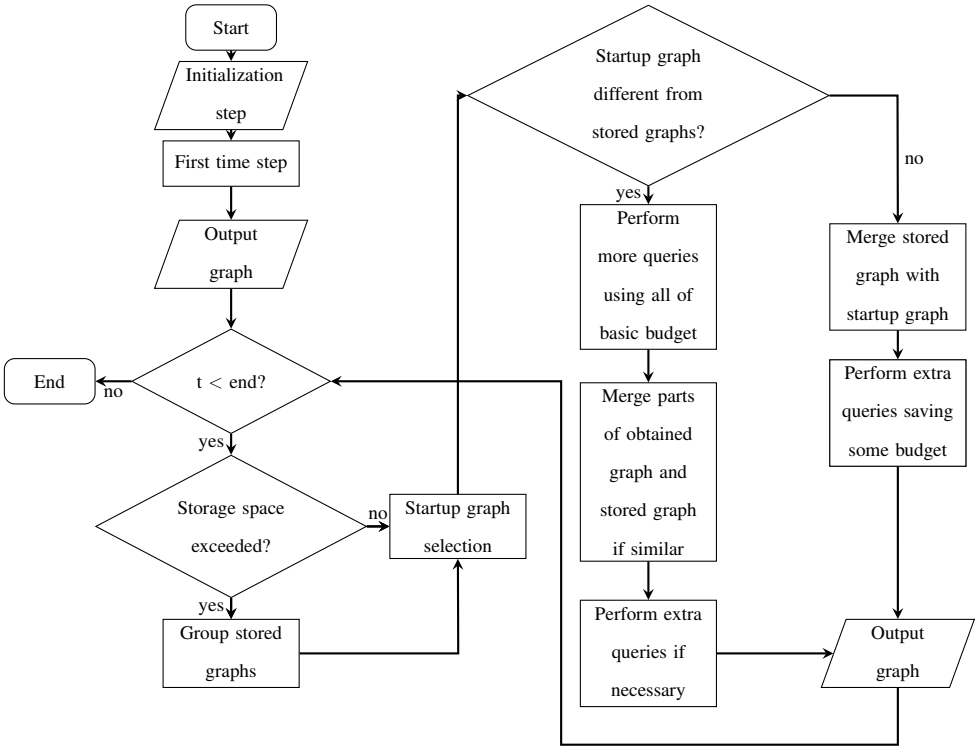


Figure 3.2: A high level description of the steps involved in the proposed dynamic network sampling.

DYNSAMP begins by obtaining a sample for the first time step of the sampling process with an allocated number of queries. For subsequent time steps, a fraction of the budget allocated for that time is used to obtain a graph called the *startup graph*. The startup

graph is then compared to previously discovered graphs to determine if they are similar. If similar, a portion of the budget allocated for that time step is saved for future use. If not similar, the entire allocated budget for the time step is used. If there is saved budget, it is used to perform extra queries to grow the graph. Figure 3.2 shows a high-level view of the proposed approach to sampling dynamic social networks. A detailed description of the steps involved is described below.

3.3.1 Initialization

The sampling process requires as input either a total budget q or vector of daily budgets q_v, depending on the problem setting, the number of time steps n considered, and a dissimilarity threshold τ above which we declare two communities to be different. If the budget constraint applies to the entire period, for each time step t, we allocate a basic budget $\varphi_t = \varrho_t/n_t$ where ϱ_t and n_t is the budget and number of time steps respectively left as at time step t. However, if there is a limitation for each time step, a basic budget of $\varphi_t = q^t_{max}$ is defined for each time step t.

3.3.2 First time step

For the first time step of sampling, a *budget* φ_0 is used to generate a sample by beginning with a random node, and in each step, with probability p, querying the node with the maximum observed degree, or with probability $1 - p$, jumping to a random node, and storing the observed graph. Our experimental results suggested that this technique works well in comparison to methods such as random node selection and random walk.

3.3.3 Startup graph selection

In subsequent time steps after the first time step, a fraction of the basic budget η_t is used to obtain a startup graph for the time step under consideration. In this work, a budget of

$0.50 * \varphi_t$ is used to obtain the startup graph. The nodes queried are noted and the amount of change in their neighborhood of all previously stored graphs over the period is computed using Jaccard similarity.

In generating the startup graph, DYNSAMP selects the top η_t queried nodes with the highest change in neighborhood as at the time under consideration. This selection process ensures that nodes whose neighborhood have not changed over a period are not selected often. The selected nodes are queried to obtain the startup graph. In cases where the number of queried nodes present in the current time step is less than η_t, a random number of nodes are selected from the current nodes to add up to η_t.

3.3.4 Comparing startup graph to previous graphs

Next, the startup graph is compared to all previously obtained graphs to identify any similarities. In this work, the dissimilarity between two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ is defined as $1 - |E_1 \cap E_2|/|E_1 \cup E_2|$.

In comparing a startup graph and a previously discovered graph, if all nodes queried in the startup graph are present in the stored graph, only such nodes and their neighbors are considered. However, if all nodes queried in the startup graph are not present, a random set of queried nodes in the stored graph are selected and their neighbors are considered for comparison between the two graphs. The selection is done such that the number of nodes queried in both graphs are equal. DYNSAMP selects the stored graph most similar to the startup graph.

3.3.5 Performing extra queries

If the startup graph is within τ of the closest graph, the connections in the stored graph that are between nodes present in the current sample are added to the initial graph obtained. In this work, an assumption is made that a check can be made to determine if a node is present or not. The remaining budget is used to perform some extra queries to grow the graph. If

Algorithm 1: DYNSAMP: An algorithm for sampling the community structure of dynamic networks when queries can be saved

function DYNSAMP (G, $budget$, τ, n);
Input : G, $budget$, τ, n
Output: G^s
$snaps = n$, $\varphi_1 = budget/snaps$, $G_0^s \leftarrow getFirstTimeSample(G_0, \varphi_1)$;
Decrement $budget$, Decrement $snaps$;
Store graph at $t = 0$;
for $t = 1$ *to* n **do**
  if *storage exceeded* **then**
    $groupGraphs()$
  $\varphi_t = budget/snaps$,$startup = 0.50 * \varphi_t$;
  $G_t^s \leftarrow getStartUp(G_t, startup)$;
  G_{sel} ←Find closest stored graph;
  $\delta \leftarrow graphDissim(G_t^s, G_{sel})$;
  if $\delta > \tau$ **then**
    use all of base size;
    $G_t^s \leftarrow mergeComPart(G_t^s, G_{sel})$;
    if $\delta > \tau$ **then**
      $extra = savedqueries$;
      $G_i^s \leftarrow performExtraQ(G_t, G_t^s, extra)$;
  else
    $extra = (0.90 * \varphi_t) - startup$;
    $G_t^s \leftarrow merge(G_t^s, G_{sel})$;
    $performExtraQ(G_t, G_t^s, extra)$;
    update saved queries;
  update queried neighbors;
  Decrement $budget$, Decrement $snaps$;

budget is allocated for the entire duration, a fraction is saved for future use. By performing extra queries in cases when the graphs are similar, it provides a means of growing the graph that was previously stored.

In cases where the startup graph is identified to be entirely different from all previously obtained graphs, the remaining budget is used to grow the network. A further check is made if some parts of the startup graph are similar to the closest stored graph. Each community is merged with its closest community in the stored graph based on the defined threshold. Communities in this work were obtained using the Louvain method [16]. In the cases where a budget could be saved, the saved budget is used to perform additional queries, since the

discovered community structure is deemed wholly new.

3.3.6 Handling storage limitations

Due to space limitations, it may not be possible to store all previous graphs especially when sampling for a larger number of time steps. To address this, for all time steps after the first time step, DYNSAMP checks if the entire storage is used before the sampling for that particular time begins. The stored graphs are clustered into groups.

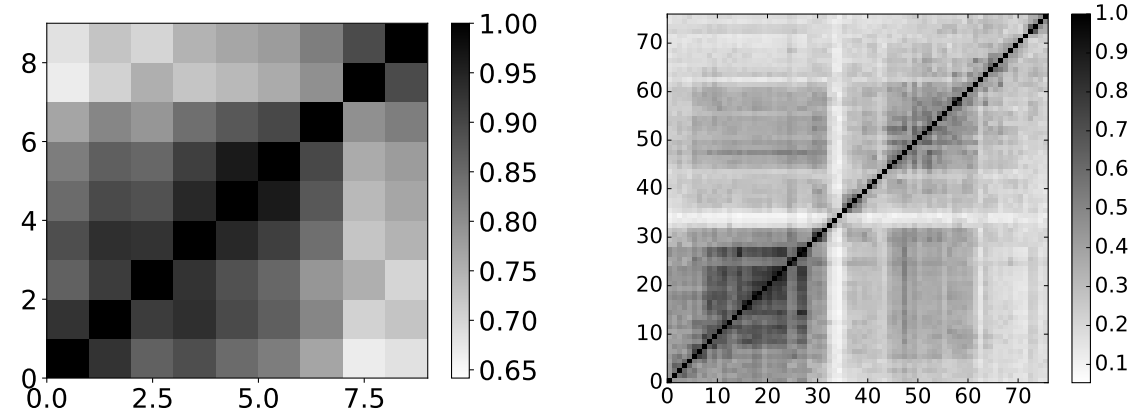
When the storage limit is exceeded, a check is made to determine the number of unique graphs among all stored graphs. A stored graph is said to be unique when it is not similar to any of the stored graphs. If among all the stored graphs there is only one unique stored graph, this means that the graphs stored are all similar to each other and hence grouped into a single graph. As an example, assuming $G_1^s = (V_1^s, E_1^s), G_2^s = (V_2^s, E_2^s), ..., G_m^s = (V_m^s, E_m^s)$ are the currently stored graphs with a single unique stored graph. A new graph $G_\alpha^s = (V_\alpha^s, E_\alpha^s)$ such that $V_\alpha^s = \bigcup_{i=1}^{m} V_i$ and $E_\alpha^s = \bigcup_{i=1}^{m} E_i$ is obtained after the merging process. If there are k unique stored graphs, where $k > 1$, an initial attempt is made to group the graph into k groups. After the grouping into k, if storage is still exceeded, graphs with the least assessed time are repeatedly considered for eviction until the storage criteria is met. Algorithm 1 provides a step by step description of the proposed technique when queries can be saved while algorithm 2 gives the description of the technique when queries can not be saved.

3.4 Experiments

This section begins with a description of various real and synthetic datasets used for the experiment. It is followed with how the experiments were set up and the main objectives in the various experiments. The section ends with a discussion of the results of each dataset.

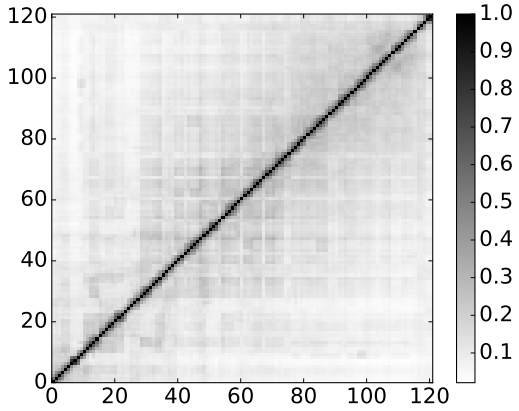
Algorithm 2: DYNSAMP: An algorithm for sampling the community structure of dynamic networks when queries can not be saved

function DYNSAMP (G, q, τ, n);
Input : G, q, τ, n
Output: G^s
$G_0^s \leftarrow getFirstTimeSample(G_0, q_{max}^0)$;
Store graph at $t = 0$;
for $t = 1$ *to* n **do**
    if *storage exceeded* **then**
        $groupGraphs()$
    $startup = 0.50 * q_{max}^t$;
    $G_t^s \leftarrow getStartUp(G_t, startup)$;
    $G_{sel} \leftarrow$Find closest stored graph;
    $\delta \leftarrow graphDissim(G_t^s, G_{sel})$;
    $extra = q_{max}^t - startup$;
    if $\delta > \tau$ **then**
        $performExtraQ(G_t, G_t^s, extra)$;
        $G_t^s \leftarrow mergeComPart(G_t^s, G_{sel})$;
    else
        $G_t^s \leftarrow merge(G_t^s, G_{sel})$;
        $performExtraQ(G_t, G_t^s, extra)$;
    update queried neighbors;
    updated stored graphs;



(a) Syn1

(b) MIT contact



(c) Enron

Figure 3.3: A plot of the similarity between community structures over time for three different group of networks. 3.3(a) is a network with totally stable community structure, 3.3(b) has a partially stable community structure and 3.3(c) is network with a completely unstable community structure. Black indicates two graphs are equal and white indicates they are completely different. These examples demonstrate stable (Syn1), mixed (MIT), and unstable (Enron) structures.

3.4.1 Datasets

We consider five datasets. These include three real world datasets: Autonomous Systems (AS-733) [45], Reality Mining (MIT contact)[27] and Enron email (Enron) [77]. We also include two synthetic datasets (Syn1 and Syn2), generated using Dancer [11]. Dancer generates evolving graphs with embedded community structure. The network generation process begins by generating an initial graph following real world properties such as pref-

erential attachment, small world and homophily. The initial graph is modified through two main operations: micro operations such as addition and removal of nodes and macro operations such as splitting of a community into sub-communities.

AS-733 is a communication network constructed from Border Gateway Protocol logs. It contains 733 daily instances, the largest of which has 6,474 nodes and 12,572 edges. Reality Mining is a human contact network among 100 MIT students. This dataset contains 229 daily instances describing contacts between users, each of which has up to 76 nodes and 418 edges. We aggregate these daily instances using window sizes of 10 days, with a step of 3, to generate a total of 77 snapshots. Enron is an email network. We use a dataset containing daily snapshots during the year 2001, again aggregated as above, for a total of 122 snapshots. These graphs contain up to 7,225 nodes and 15,938 edges. All networks exhibit the addition and deletion of both nodes and edges. Some snapshots are aggregated to ensure all the different community structure behavior are considered in the experiment.

The synthetic networks both have an initial node count of 2,000 initially grouped into 20 and 24 communities for Syn1 and Syn2 respectively. These go through different community evolution phases such as splitting and merging. This model requires a number of parameters. We set $k = 20$, *nBVertices* = 2000, *nbTimestamps* = 10, *prMicro* = 0.2, *prMerge* = 0.4, *removeVertices* = 0.4, *prSplit* = 0.4, *prChange* = 0.4, *addBetweenEdges* = 0.2, *addVertices* = 0.1, *removeBetweenEdges* = 0.4, *removeWithinEdges* = 0.1, *updateAttributes* = 0.1. For Syn2, the same settings were maintained with modification to the following: *prMicro*=0.5, *addBetweenEdges*=0.5, *removeBetweenEdges*=0.9, and k=24

The largest time step of Syn1 has 2,293 nodes with 17,813 edges. Syn1 over the period considered shows an addition and deletion of edges. However, it only demonstrates the addition of nodes over the period. In Syn2, the largest number of nodes over the period is 2,859 and the largest number of edges over the period is 21,459. Syn2 tends to have communities splitting or migrating more often than communities in Syn1.

3.4.2 Experimental setup

In our experiments, we set $\tau = 0.4$ and $p = 0.80$. The budget size in the experiments with a strict budget limitation for each time step is defined to 20% of the number of nodes at each time step. Budgets for the setting in which we have a total number of queries for the entire timeline are stated later in this section.

To the best of our knowledge, there is no sampling method that explicitly focuses on the community structure of dynamic networks without assuming knowledge of the entire network. The proposed method is therefore compared with random walk (`RW`) and breadth-first search (`BFS`) and maximum observed degree (`MOD`) baselines.

3.4.3 Evaluation metrics

We use two metrics to evaluate `DYNSAMP` and the baselines above. The first metric is based on a Jaccard-based metric proposed in [99], modified for evaluating dynamic samples. Given a sampled set of communities C_i^s and a true set of communities C_i, this metric finds the closest true community to each sampled community, and vice versa, and averages these similarities. We also use the popular Normalized Mutual Information (NMI) metric, described in [6, 21].

3.4.4 Results and discussion

For each of the datasets, we run `DYNSAMP` and the baselines 10 times to generate a dynamic sample with specific budget. We compare to communities detected on the complete network by the Louvain method [16]. Results for both evaluation metrics were similar, so we present results for NMI only in Figure 3.4.

In our experiments, we use a budget size of $15\% * minG * n$, where $minG$ is the minimum graph size of all the snapshots. The budgets 199000, 850, 13000, 2500 and 2500 were respectively used for AS-733, MIT-contact, Enron, Syn1 and Syn2. Figure 3.4 shows

a similar plot of the NMI with respect to time (results were similar for the Jaccard-based evaluation metric). In these experiments, the setting where a budget is given over the entire period is used. Similarly, Figure 3.5 shows a similarity plot of the NMI with respect to time when there is a budget limitation per timestep.

Dynamic social networks can be categorized into three groups based on the stability of the community structure over the period considered (see Figure 3.3 for examples): those that are stable over the entire period (e.g., Syn1), those that are unstable (e.g., Enron), and those that are mixed (e.g., Reality Mining).

In a dynamic network where there is a complete or partial stability of the community structures over the period considered, DYNSAMP outperforms baseline methods substantially. When the community structure changes significantly at each time step, like the Enron dataset, there is no significant difference between DYNSAMP and the baselines, because it cannot learn from the past.

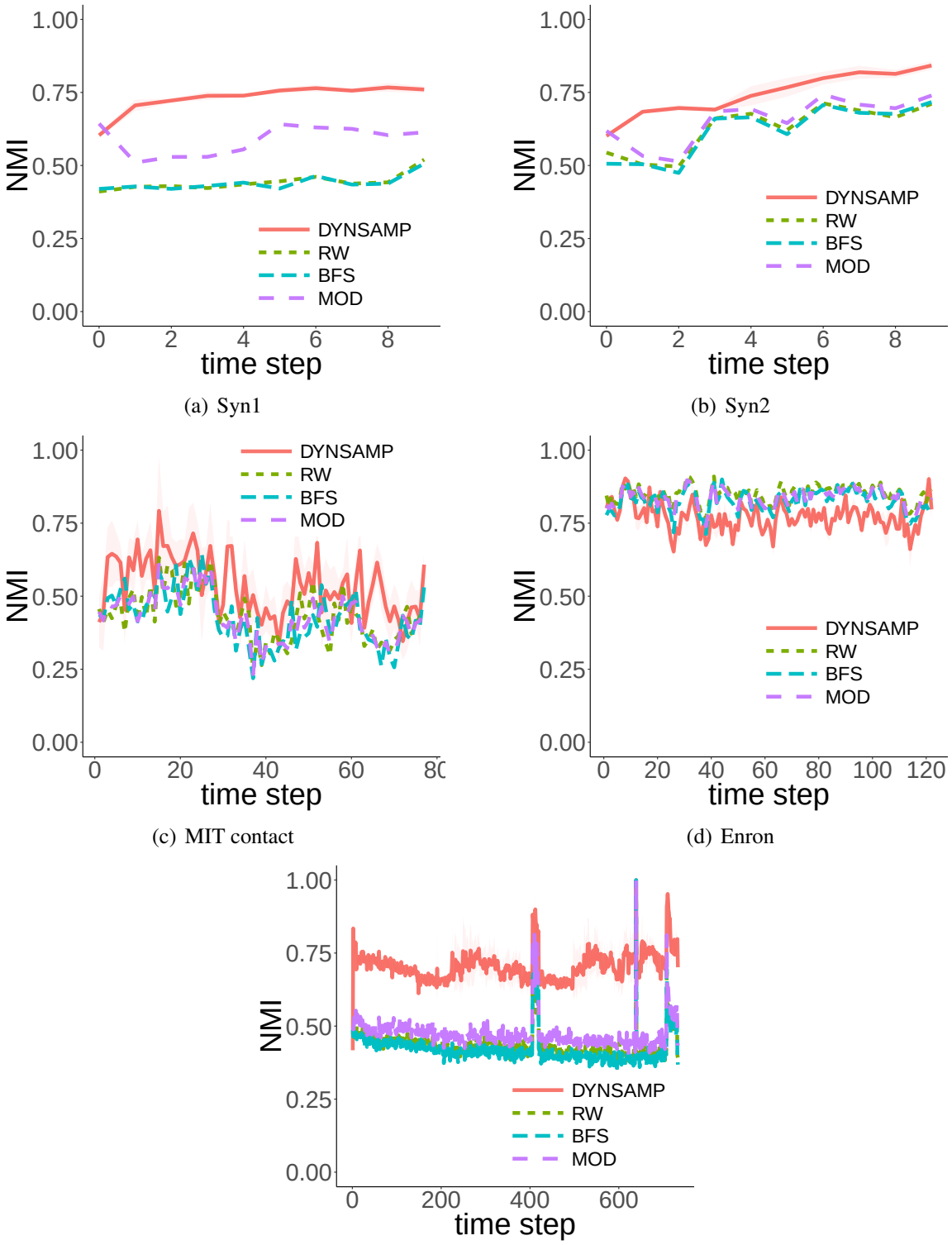
We next investigated whether the number of graph samples stored had a significant impact on the performance of DYNSAMP. The investigation was divided into two: graphs that have some stability over time (Syn1) and graphs with no stability over time (Enron). We observe that, in general, the performance of DYNSAMP is not dependent on the number of graphs being stored. If there is some stability, it will be merged over time and hence keeping several copies of them will neither improve or worsen the performance. In cases where there is no stability, the number of stored graphs has no impact on the learning process.

Overall, we observe that DYNSAMP performs better than baseline methods in most cases. With the Jaccard based measure, it outperforms RW by 42% , MOD by 39% and BFS by 46% on average, and by 35%, 32% and 53% as measured by NMI.

3.5 Conclusion

In this chapter, we addressed the problem of sampling a dynamic social network when there is a limitation on the number of nodes that could be asked for information. We considered two resource constraint scenarios: cases where there is a limitation on the number of nodes that could be asked over the entire period and instances where there is a limitation on the number of nodes for each time step. Our proposed framework, DYNSAMP first obtains a sample for the first time step. In subsequent time steps, it uses a fraction of the allocated budget for that time to obtain a startup graph. The startup graph is compared with previously discovered graphs. If the startup graph is similar to a previously discovered graphs, a portion of the budget is saved. However, if the startup graph is not similar to any of the previously discovered graphs, a portion of the saved budget is used to perform perform extra queries to grow the network.

We performed experiments on several real world and synthetic networks. We showed that in most cases the proposed approach outperforms baseline methods. However, in cases where the community structure for each time step changes significantly, the algorithm performs as well as the baseline methods.



(a) Syn1

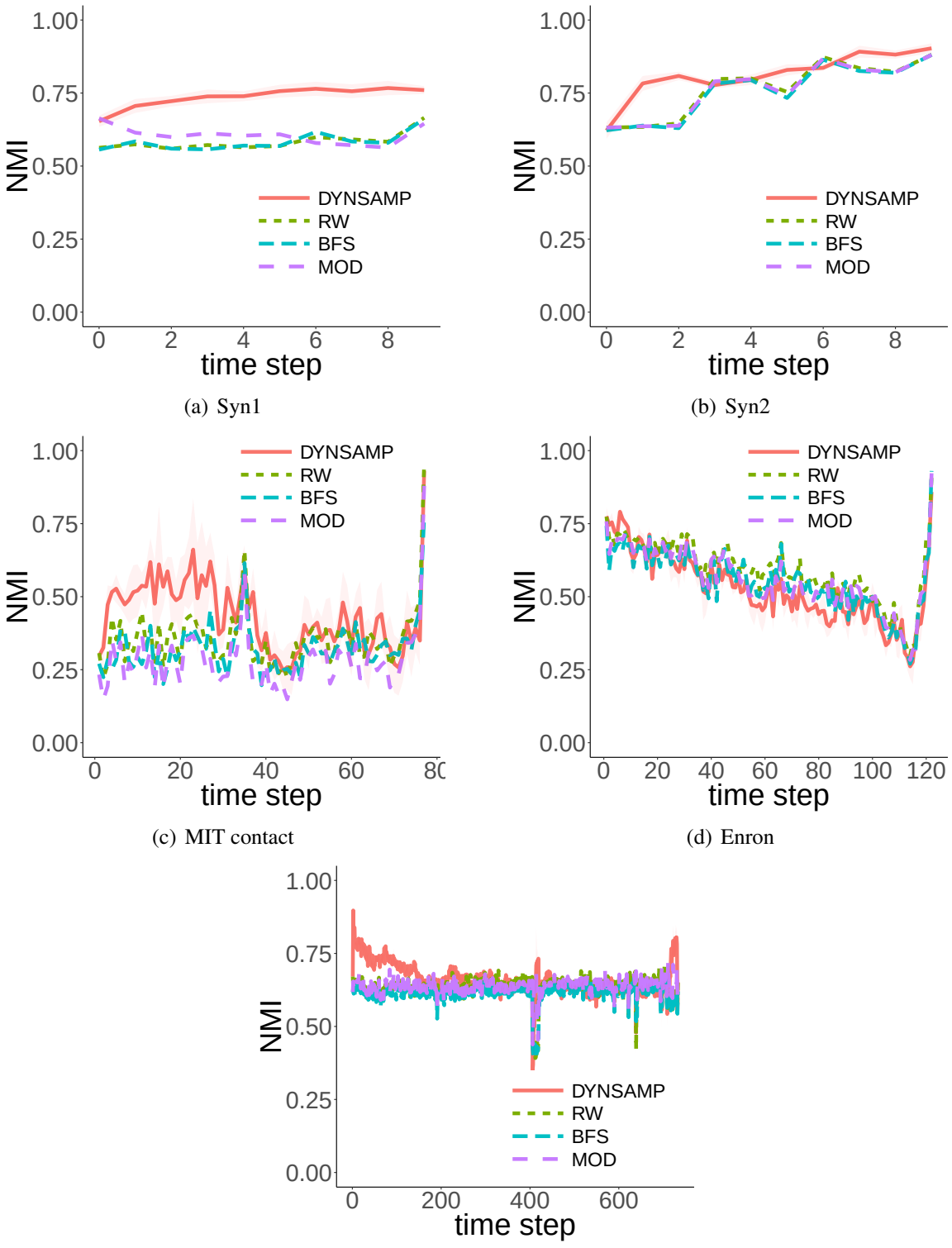
(b) Syn2

(c) MIT contact

(d) Enron

(e) AS-733

Figure 3.4: A plot of the NMI between a sampled graph and its corresponding true graph over time. Shading represents the standard deviation over 10 trials. DYNSAMP outperforms the other methods with respect to NMI in most cases. When graph changes at each time step like 3.4(d), it performs just as baseline methods.



(a) Syn1

(b) Syn2

(c) MIT contact

(d) Enron

(e) AS-733

Figure 3.5: NMI between a sampled and true graphs, with a sample budget for each time step. Shading represents the standard deviation over 10 trials. DYNSAMP outperforms the other methods with respect to NMI in most cases.

CHAPTER 4
SAMPLING COMMUNITIES IN NODE ATTRIBUTED EDGE STREAMS

Identifying communities is a critical data mining application, and can help unravel the structure of groups in a network. Community detection has been used, for example, to identify different groups of users in a phone network [16], describe communication patterns [84], and identify hierarchical structures [20, 19].

While classically, most works on identifying communities have focused exclusively on the topology of the network, more recent algorithms have begun to incorporate nodal information. For example, consider an email network where nodes represent members of a university community and a connection exist between two nodes if they exchange an email. The node attributes could be characteristics such as gender, position, department and age. Many algorithms have been proposed to deal with node attributed networks. However, almost all of these works apply only to static networks- i.e., they assume that the entire node attributed network is available [98, 46, 64].

Another setting of community detection which is gradually gaining attention is the identification of communities from edge streams. Most of these works focus mainly on the structural information ignoring the nodal information [73, 33, 47]. Identifying commu-

nity from edge streams in addition to providing a means of handling very large networks is useful in various domains. For instance, in an e-commerce network, if one is able to identify the community of a user who purchased a specific item at the time of purchase, it helps in recommending possible products that will interests the user as at the time of purchase. Previous works in detecting communities in static networks have demonstrated the significance of including nodal information in the detection of communities [28, 64] .

In this chapter, we propose `SAMPLearn`, a novel algorithm for sampling communities in node attributed edge streams. `SAMPLearn` outputs both a graph that is representative of the original community structure and the community memberships of nodes in the sampled graph. It combines the network structural and nodal information in determining community memberships of nodes. The intuition behind `SAMPLearn` is to learn if attributes are useful to characterize a community or not. If the nodal information can characterize the communities of a network, `SAMPLearn` combines the nodal information and structural information in determining community memberships. However, if the attributes are not useful, then it focuses only on the structural information in determining the community membership of nodes. Experiments show that `SAMPLearn` has a performance improvement ranging from 11% to 40% when attributes do not characterize any of the communities, 25% to 88% when attributes characterize some communities and up to about 5 times improvement when attributes characterize all communities.

This chapter is organized as follows. We present the problem of sampling communities in edge streams with nodal information in Section 4.1. In Section 4.2, we discuss some related work. Section 4.3 discusses the proposed approach. In Section 4.4, we discuss the experiments performed and its set up.

4.1 Problem definition

Let $G = (V, E, \Upsilon)$ be a streaming graph with node attributes where V is the set of nodes, $E \subseteq V \times V$ is the set of connections between nodes and $\Upsilon = \{a_1, a_2, ..., a_k\}$ represents the set of k attributes associated with all nodes in V. An edge $e_i \in E$ arriving at the i^{th} time step is represented as $(u_i, v_i, \Gamma(u_i), \Gamma(v_i))$ where $\Gamma(u_i) = [a_1(u_i), a_2(u_i), ..., a_k(u_i)]$ is the vector of attributes associated with node u_i. At time t, a graph $G_t = (V_t, E_t)$ is the graph observed up until time t where V_t and E_t is an aggregation of all nodes and edges respectively arriving up until t . $G_t^s = (V_t^s, E_t^s)$ is a sampled graph from G_t , where V_t^s and $E_t^s \subseteq V_t^s \times V_t^s$ are the sampled set of nodes and edges respectively from all nodes and edges arriving up until t.

In this work, we assume the number of communities is known and that there is an initial seed set of edges for each community. We define a community to be a group of nodes with more connections among members in the community in comparison to members outside the community [30, 49]. With a limitation on the number of nodes that can be stored, our goal is to obtain a subgraph G_t^s from a stream of edges G_t such that the similarity of the communities between G_t^s and G_t is maximized.

4.2 Related work

There is an extremely large body of work on community detection in general. Some of the different settings considered in the study of communities include the static setting, where there is full access to the network [16, 56, 89], the crawling setting [15], where the graph is hidden but supports the exploration of neighbors of a given node, the dynamic setting where nodes evolve over time [67], and the streaming setting, where there is limited memory [73]. There has also been works in identifying communities in edge streams [33, 97]. Most of these techniques mostly focus on detecting the communities based on the structure of the network. Recently, there is a different set of techniques that considers both the nodal

information and structural information in a static setting [64, 28].

For sampling from edge streams, Ahmed et al. proposed the PIES method for generating a subgraph from an edge stream that preserves the inherent clustering structure of the edge stream [4]. In this method, in each step, a node is added to the currently obtained sample with some probability. When the limit on the number of nodes is exceeded, the proposed technique ensures that only higher degree nodes are maintained. Similarly to [4], Zakrzewska and Bader [101] proposed another method for subgraph sampling from an edge stream, but with an extra restriction on the number of edges. However, these two methods do not (and do not claim to) capture communtiy structure.

Hollocou et al. proposed a linear algorithm for detecting communities in social networks [33]. The process [33] begins by initializing each node as a new community and assigning a degree of 0 to all nodes. When an edge (u, v) is observed, the degree of both nodes u and v are incremented. If the degree of nodes u and v are both less than a specified D, the node with the smaller degree will switch its membership to match that of the other. If one of the nodes has an observed degree of more than D, there is no community membership change. In selecting D, authors propose the mode of all node degrees. Even though this algorithm works very well, it works only when there is full access to the entire network.

The authors in [47] proposed a technique to generate the community structure of an edge stream. When a new edge arrives, if none of the nodes in this edge are part of the current vertex set, it is ignored. However, if one of them is present, the other is added to the community of the present node. After some number of edges has been processed, the process prunes each community to a pre-specified size. In the pruning process, the nodes with higher community performance are maintained. The community performance of a node u is defined as the fraction of nodes incident on u that are in the same community as u. Another approach proposed for sampling communities from an edge stream is `COMPAS` [73]. `COMPAS` begins by adding all edges until the number of nodes in the sample is equal

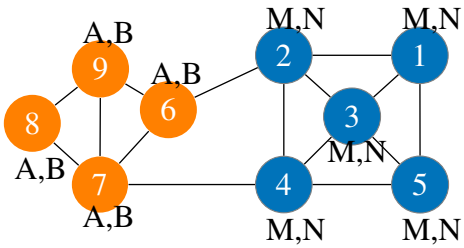
to the specified threshold. A pre-selected algorithm is used to first obtain the initial community structure. When the threshold is met, COMPAS estimates the importance of nodes by considering nodes that are discovered more often in the stream.

Even though there is a large amount of work done on identifying communities in social networks, most of these works focus primarily on the topological structures of the network [33, 73, 47]. Considering nodal information when available in detecting communities is said to provide more representative communities than considering only structural information [28].

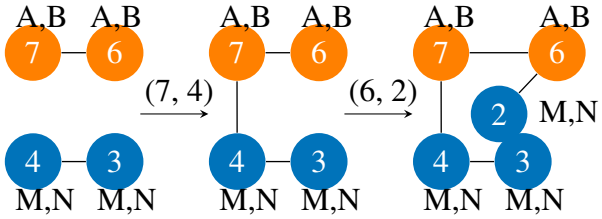
4.3 Methodology

In this work, we propose SAMPLearn, a novel algorithm to sample the community structure of streaming edges with nodal information. In addition to the community structure, SAMPLearn also outputs a representative graph of the original graph. SAMPLearn combines the nodal information, when present, with the structural information of the network to generate the final sample. Intuitively, if the attributes can characterize the community, SAMPLearn considers both the structure and attributes of nodes to assign a node to that community. However, if the attributes are not useful in characterizing the community, SAMPLearn uses only the structural information. In this work, we assume all node attributes are quantitative attributes, but the method can easily be generalized to categorical attributes. Algorithm 3 provides a step by step description of the processes involved in SAMPLearn.

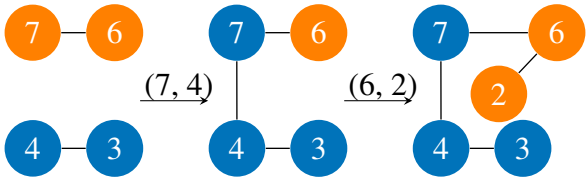
The process begins with a seed set of edges from each community, forming an initial graph. Only edges with at least one node in the sampled graph as at the time of edge discovery are considered. We assume that there is no information on an edge that has no node in the sample. Figure 4.1 shows a high level view of the processes involved in identifying the community structure of the edge stream of the graph in Figure 4.1(a). Figure



(a) Initial graph



(b) When attributes characterize communities



(c) When attributes do not characterize communities

Figure 4.1: Example illustrating a high level view of the processes involved in SAMPLearn. Figure 4.1(b) represents an instance where attributes are characteristic of the community while Figure 4.1(c) illustrates the case where attributes do not exist or characterize the communities.

4.1(b) illustrates the case where the attributes are characteristic of the community structure, and Figure 4.1(c) illustrates the case where the attributes do not exist or characterize the communities. When a new edge is discovered, there are two cases that could occur: (1) Both nodes are present in the sample. (2) Only one node is present in the sample.

If both nodes are present in the sample, SAMPLearn decides on whether there should be a switch in community membership of any of the nodes or not. For the toy graph in Figure 4.1(a), assume that the process is initialized with edges $(7, 6)$ and $(4, 3)$. If an edge (7, 4) is seen in the stream, since both 7 and 4 are present in the current sample, we decide whether the discovery of the edge could cause a change in communities of node 7 and 4. In the case where attributes are useful, as in Figure 4.1(b), 7 and 4 will remain in their assigned

communities. Nodes 7 and 4 were maintained in their current communities because a switch in community membership will not improve the communities when we consider both the attributes and structure of the currently sampled graph. However, if the attributes are not characteristic of the communities as in Figure 4.1(c), then we make the decision only based on the structure. In the case where attributes are not useful, node 7 switches its community membership to that of node 4 because based on the current information, it improves the community structure of the edges seen so far.

Algorithm 3: `SAMPLearn`: An algorithm for detecting the community structure of streaming edges with node attributes

```
1  function SAMPLearn (edgestream, k, W, ι, α);
   Input  : edgestream, k, W, ι, α
   Output: G^s, C^s
2  curG ← Initialized with 3 edges from each community;
3  C ← Initialized community memberships of nodes in curG;
4  Equi_C ← Initialized equitability of communities;
5  Imp_C ← Characteristic features of communities;
6  if t mod W = 0 then
7      for C_i in C do
8          Imp_C_i ← Useful features of community C_i;
9          Equi_C_i ← Compute equitibility of C_i with respect to Imp_C_i;
10 for (u_t, v_t) in edgestream do
11     if u_t in curG and v_t in curG then
12         both_present (curG, Imp_C , Equi_C , ι , u_t, v_t) ;
13     else if u_t in curG then
14         one_present(curG, useful_bool, u_t, v_t, k );
15     else
16         Don't consider edge ;
17 return curG, C
```

If only one of the nodes is present, `SAMPLearn` estimates the community membership of the newly discovered node and the edge is added to the current sample. First, `SAMPLearn` computes the similarity between the attributes of the newly discovered node and communities that are characterized by their attributes. If the similarity is more than a specified threshold, the newly discovered node is assigned to that community. However, if the similarity is less than the threshold, the newly discovered node is assigned to the community

of the node it is directly connected to. For example, in the case of Figure 4.1(b) where the attributes are identified to characterize the two communities and an edge $(6, 2)$ is seen, the community of 2 is inferred to be in the same community as 4 and 3 since they have the same attributes. In the case of Figure 4.1(c), where there are no attributes, 2 is assigned to the community of 6. Since `SAMPLearn` keeps no more than k nodes in the sampled graph, in cases where only one node is present in the sample but the threshold is met, the new node is added to the sampled graph with some probability and one of the nodes with lesser `importance` in the sampled graph is removed.

A detailed description of `SAMPLearn` is discussed below. Algorithm 4 describes the steps involved when both nodes are present in the current sample while Algorithm 5 describes the steps involved when only one node is present in the current sample.

4.3.1 Initialization

`SAMPLearn` requires the some parameters initialized in order to begin the sampling process. The parameters required are as follows:

- Seed edges for each community.
- Window size W during which `SAMPLearn` re-examines the attributes that characterize communities.
- Budget k that indicates the maximum number of nodes that can be stored.
- A threshold ι is which indicates the measure above which the attributes of members in a community is considered capable of characterizing the community.
- A weight measure α such that $0 \leq \alpha \leq 1$ to scalarize the effect of the structural and nodal information.

Algorithm 4: `both_present`: A function to process an edge (u, v) when both u and v are present in the current sample

```
function both_present ( cG, Imp_C, Equi_C, u, v, ι);
Input: cG, Imp_C , Equi_C , ι ,u, v
impr_u_to_v        ▷ improvement when node u moves to community of v;
impr_v_to_u        ▷ improvement when node v moves to community of u ;
if Equitability_of_community > ι then
    compute quality score(s) using both structural and nodal information ;
else
    compute quality score(s) using only structural information;
if impr_u_to_v > impr_v_to_u and impr_u_to_v > 0 then
    add u to community of v;
    merge neighbors if necessary;
if impr_v_to_u > impr_u_to_v and impr_v_to_u > 0 then
    add v to community of u;
    merge neighbors if necessary;
Add (u, v) to cG;
```

4.3.2 Both nodes present

When both endpoints of an observed edge (u, v) are present in the current sample, `SAMPLearn` adds the edge to the current sample. Even though adding this edge can never result in exceeding the required budget on number of stored nodes, the community structure could be affected. The addition of a new edge could result in a change in community memberships of none or one of the nodes. Depending on whether `SAMPLearn` identifies the attributes to be useful, different community quality scores are computed.

When attributes characterize communities

For nodes in a community whose attribute similarity is more than a threshold ι, `SAMPLearn` considers both structural and nodal information. To determine the community memberships of u and v, `SAMPLearn` first computes the initial score of the communities that u and v belongs. It then computes the quality score when (1) u moves to the community of v and (2) when v moves to the community of u. The improvement for each possibility is computed and the final decision is based on the improvement that results in the maximum

non-negative score. Whenever a node switches membership, all degree-1 neighbors of the node also switch their membership. Because the only node these neighbors have ties to in a community switched its membership, we assume they are also likely to switch their memberships.

Computing attribute similarity: When some attributes of nodes in a community are identified to characterize that community, `SAMPLearn` leverages the nodal information with the structural information in making a decision on whether to assign a node to the community. The steps involved in identifying whether the attributes of members in a community characterize the community or not and what attributes characterize the community is discussed later in Section 4.3.3. `SAMPLearn` computes the similarity of all nodes in a community with respect to the attributes identified to characterize the community. Given two nodes u and v with their respective vectors of useful attributes $\Gamma^*(u)$ and $\Gamma^*(v)$, the attribute similarity $\wp$ between $\Gamma^*(u)$ and $\Gamma^*(v)$ is defined as:

$$\wp(\Gamma^*(u), \Gamma^*(v)) = \frac{1}{|\Gamma^*(u)|} \sum_{i=1}^{|\Gamma^*(u)|} \Big(1 - \frac{|a_i(u) - a_i(v)|}{max(a_i(u), a_i(v))}\Big). \tag{4.1}$$

Assume that there is an initial community $C_{ini} = \{n_1, n_2, ..., n_k\}$. To compute the similarity of attributes Q_{att} of nodes in C_{ini} after a new node n_{new} joins, we first compute the center $\Gamma(ct)$ of the attributes of nodes in C_{ini} before n_{new} is added. We consider the average along each attribute for all nodes as the center. The similarity is then computed as below:

$$Q_{att}(C_{ini}) = \frac{1}{k+1} \sum_{i=1}^{k+1} \wp(\Gamma(ct), \Gamma^*(C_{ini}(i))). \tag{4.2}$$

The intuition is to estimate the similarity between nodes in a community and the center of the community with respect to features that characterize the community. We define the center of a community as the mean of the useful attributes of all nodes in the community. If the new node added to a community has very different attributes from the center, then it will result in a smaller similarity value which will in turn decrease the quality function.

However, if the attributes are somewhat close to the center, it will result in a higher attribute similarity which in turn increases the quality function. We combine the similarity of the attributes with the structural information to determine a node's community membership. We estimate the quality of a community C_{ini} with respect to the structure Q_{struct} as:

$$Q_{struct}(C_{ini}) = \frac{|inC| + 1}{|outC| + 1}, \tag{4.3}$$

where inC is the set of edges with both ends in C_{ini} and $outC$ is the set of edges with one end in C_{ini}. The intuition behind Q_{struct} is to ensure that communities with lots of interactions among themselves but fewer interactions outside the community result in a higher quality. The overall quality Q of a community C is therefore defined as:

$$Q(C) = \alpha Q_{struct}(C) + (1 - \alpha)Q_{att}(C), \tag{4.4}$$

where α is user specified parameter to scalarize the structural quality and the attributes similarity into a single quality function. α is proportional to the importance of the nodal and structural information.

When attributes do not characterize communities

For nodes in a community whose attribute similarity is less than a threshold ι, `SAMPLearn` considers only the structural information in deciding whether to add a node to such community. In such instances, the quality score Q of a community C is then equal to the quality of the community with respect to the structure Q_{struct}.

4.3.3 Identifying useful attributes

`SAMPLearn` considers the attributes of nodes, if present, and combines that with the structural information to determine the community membership of nodes. However, not all attributes might be useful in characterizing a community. In view of this, `SAMPLearn`

Algorithm 5: `one_present`: A function to process an edge (u, v) when only node u is present in the current sample

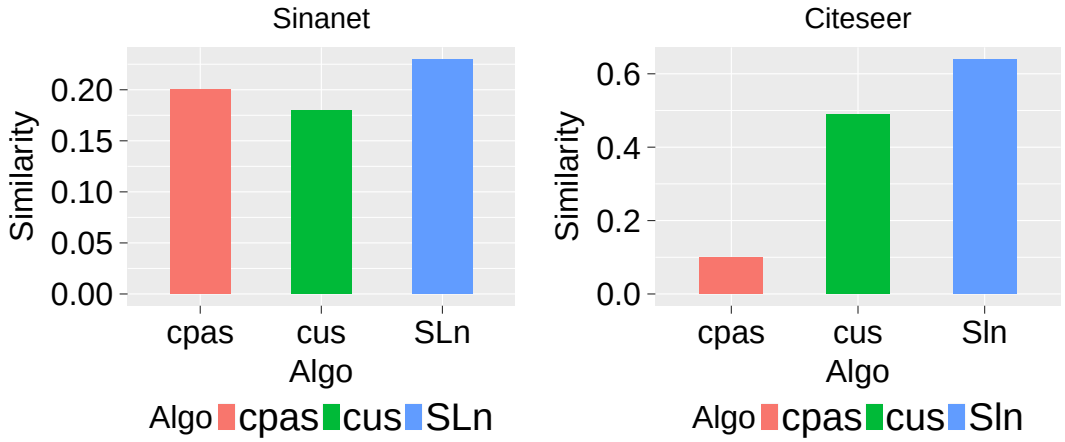
```
1  function one_present (cG, ι, u, v, k);
   Input: cG, ι, u, v, k
2  if len(cG) < k then
3  |        ▷ when number of nodes in sample is less than sample size;
4  |   determine_community_of_newnode();
5  |   add (u, v) to cG;
6  else
7  |   rand ← uniform(0, 1);
8  |   prb ← 1/len(cG);
9  |   if rand < prb then
10 |   |   nodeto_evict ← determine_nodeto_evict();
11 |   |   Remove nodeto_evict from cG;
12 |   |   determine_community_of_newnode();
13 |   |   Add (u, v) to cG;
```

first identifies the attributes that characterize a community and then consider only such attributes. Figure 4.3 shows the three (3) different relationships that could exist between communities and attributes of nodes in a network. We illustrate these relationships with two toy groups (community 1 and community 2) from the email interactions of faculty in a university.

Figure 4.3(a) shows the case where the 'department' attribute characterizes all communities in the network. Here, `SAMPLearn` will consider both the structural and attribute information in identifying communities. Figure 4.3(b) shows the case where the attributes available do not characterize any of the groups. For example, if one considers the duration of service in a university, it might not characterize the members in a department when considering email interactions among the faculty. In this case, `SAMPLearn` will only consider the structural information in assigning a node to any of the communities.

The third possible relationship (Figure 4.3(c)) is one where the available attribute(s) characterize some of the groups. For instance, if the attribute considered is a faculty's level of math knowledge, it is likely that for some departments, most members have very similar attributes (e.g., in the Math department, presumably all faculty have a high level of math

knowledge). On the other hand, there might be departments which might not necessarily require advanced level of Math, so members have mixed levels of knowledge, and so such an attribute is not useful in characterizing the community. In this instance, SAMPLearn will consider only the structural information to assign a node to community 1, since the attributes do not characterize the community. However, for community 2, SAMPLearn considers both the attribute and structural information since the attributes characterize the community.



(a) Sinanet (b) Citeseer

Figure 4.2: Similarities between the sampled and true communities for SAMPLearn (Sln), COEUS (cus) and COMPAS (cpas) on real networks with real attributes. SAMPLearn provides superior performance.

As a result of these different relationships, SAMPLearn identifies attributes that characterize each community. We identify these attributes using a Singular Value Decomposition Based entropy (SVD-entropy) feature selection technique proposed in [85]. Even though SVD-entropy identifies attributes that are relatively characteristic of a community, the members in the community might not necessarily share similar attributes. In determining if the members in a community share similar attributes, we determine the avearage equitability of these attributes [54]. Assuming $F = \{f_1, f_2, .., f_k\}$ is the set of features identified to characterize community C_i. The equitability Equ of community C_i is defined as

$$Equ(C_i) = \frac{1}{|F| * |C_i|} \sum_{f \in F} \frac{1}{\sum_{i=1}^{|C_i|} {P_{if}}^2}, \tag{4.5}$$

where P_{if} is the fraction of node i relative to all other nodes in the community when considering attribute f. We define a threshold ι above which we say the members of community C_i are characterized by attributes F.

4.3.4 One node present

When a new edge (u, v) is discovered and only one of the nodes v is present in the current sample, two possible things could occur: (1) The maximum sample size is not met, and we just add the new node to the current sample. (2) The maximum sample size is reached and so we need to decide whether to add the new node or not.

In the first case, where the sample size is not reached, the community of the newly observed node u needs to be computed. When attribute information exits, `SAMPLearn` computes the similarity of u with centers of communities whose equitability is greater than ι using equation 4.1. It considers only communities with equitability greater than ι because only these communities have been identified to have attributes of their members as characteristic of the community. If the largest similarity value between u and the center of a community is greater or equal to ι, u is assigned to that community. However, if the similarity value is less than ι or attribute information do not exist, u is assigned to the community it is directly connected to. For instance, in the toy example in Figure 4.1(b), when the edge $(6, 2)$ is discovered, since attributes exist and 6 is already in the sample, we estimate the community of 2 by computing its similarity to the centers of other communities. In the other instance as in Figure 4.1(c) where attributes do not exit, 2 is assigned to the community of 6.

In the second case, where the sample size threshold is met and a new node u is observed, u is added to the current sample with a probability of $1/k$. Once `SAMPLearn` decides to add

Table 4.1: Statistics of datasets used in experiments

Dataset	|V|	|E|	# of communities	Description
Amazon	333,344	921,703	209	co-purchasing network
DBLP	315,803	1,047,147	143	collaboration network
Brightkite	55,727	211,436	50	social network
Epinions	25,051	97,785	34	trust network
Condmat	21,223	90,768	53	collaboration network
Astroph	17,724	195,971	28	collaboration network
Anybeat	12,330	48,102	15	social network
American75	6,357	217,549	8	social network
Citeseer	3,312	4732	6	citation network
Sinanet	3,490	30,282	10	social network
Amherst41	2,224	90,919	5	social network
Hamsterster	1,814	15,082	10	social network

u to the sample, it determine the community of u using the same process described earlier for the case where the threshold is not met. `SAMPLearn` then evicts the node with the lowest community performance. The community performance of a node n is defined as the fraction of nodes incident on n that are in the same community as n [47].

4.4 Experiments

Here, we first describe the datasets used for the experiments. We then discuss the experimental goals and setup. We end with a discussion of the results of each dataset.

4.4.1 Datasets

We perform experiments on twelve real networks, two with with real attributes (Citeseer[71] and Sinanet[38]), and ten real networks with synthetic attributes [68]. We consider the generation of synthetic attributes in order to model the different attribute and community relationships as described in Figure 4.3.

The remaining ten datasets use real network topology, but synthetic attributes. For these networks, we generate synthetic attributes for nodes in different communities. Three

different scenarios were considered in the generation of attributes: (1) When attributes characterize all communities in the network. (2) When attributes characterize some communities in the network. (3) When attributes do not characterize any of the communities. We assign each node in the network with five attributes. For case 1, we generate pseudo-random number from a truncated normal distribution with $\sigma = 0.02$ for each attribute of all nodes in a community. For case 3, we randomly select each of the five attributes. In case 2, we randomly select half communities and then ensure that half of the communities are characterized by their attributes and the other half is not characterized by their attributes.

For these networks, we consider the communities detected on the complete network by the Louvain method [16] as their ground truth communities. We exclude nodes that are part of communities with less than 50 nodes, because obtaining representative samples for such communities is very easy for any algorithm when there is a seed set for those communities. This also ensures that we focus on the main communities that exist in the network.

A description of all datasets is provided in Table 4.1.

4.4.2 Experimental setup and evaluation

For our experiments, we set the sample budget k to 20% of the total number of nodes in the graph for all the datasets, ι to 0.8, α to 0.5 and W to 10% of the total number of edges in the sample. In obtaining the edge streams, we randomly shuffle the edges and then select an edge from the shuffled set one at a time. Each community is initialized with 5 randomly selected edges in each run.

We compare to two recent algorithms `COEUS` and `COMPAS` that samples communities from edge stream. To the best of our knowledge, these are the only existing methods for sampling communities from an edge stream. These two sampling techniques as described in Section 4.2 use only structural information in their operation. To the best of our knowledge, there is no community sampling technique on edge streams that considers nodal information.

To evaluate, we compare the detected communities to the ground truth communities, given a maximum number k of stored nodes. We define the ground truth communities as those obtained using the Louvain method applied to the entire (non-streaming) graph. We consider two metrics in evaluating SAMPLearn and the other techniques. We use the Jaccard based similarity measure proposed in [98], which finds the closest true community to each sampled community, and vice versa, and averages the similarities. In comparing the performance of an algorithm with respect to the original community structure, we consider only nodes that exist in the sampled community structure.

4.4.3 Results and discussion

We compare the communities obtained by SAMPLearn to those obtained by COEUS and COMPAS. We run each of these algorithms for 10 trials and compares the average over the 10 runs. The edges are randomly shuffled in each of the runs. The same set of seed edges is used for all algorithms in each run to ensure a fair comparison. Figure 4.2 presents results for real networks with real attributes and Figure 4.4 presents results for real networks with synthetic attributes.

SAMPLearn returns a sample that is almost identical (98% average Jaccard similarity) to the original communities in the case attributes characterize all communities. This is because SAMPLearn leverages the extra nodal information in deciding the community memberships of nodes. This re-affirms previous findings in detecting communities when the whole network is accessible [28, 64] that including attributes can improve the communities. Even when the attributes only characterize half of the communities, there is still a minimum improvement of 25% and maximum improvement of 88%, with an average of 71%.

In cases where attributes do not characterize communities, SAMPLearn provides a minimum of 11% improvement, an average improvement of 24% and a maximum of 40% improvement. A possible reason why SAMPLearn outperforms COEUS is that COEUS iterates

through all communities and for any community that has node u, assigns v to that community. This is clearly not always the case, because there are instances where the formation of an edge might not result in any community change or might result in a switch in only one of the nodes. On the other hand, COMPAS's definition of node importance to a community is not realistic. It assumes that those nodes that have many connections are most important in capturing the community structure. If a user u has about half of the connections to members outside and half to members within the same community, it is reasonable to assume that this is not a loyal user to the community and $u's$ behavior is not representative of members of the community.

A possible reason why COEUS is not performing as well as SAMPLearn is a fundamental assumption which is not realistic. In the streaming process, assuming an edge (u, v) is formed, it iterates through all communities and for any community that has node u, COEUS assigns v to that community. This is not always the case because there are instances where the formation of the edge might either not result in any community change or might result in a switch in only one of the nodes. On the other hand COMPAS's definition of node importance to a community is not realistic. It assumes that those nodes that have many connections to other nodes are most important in capturing the community structure. If a user u has about half of the connections to members outside and half to members within the same community, it is reasonable to assume that this is not a loyal user to the community and $u's$ behavior is not representative of members of the community.

4.4.4 Time complexity

Given graph $G = (V, E)$, two main functions are required to process an edge from the stream: when both nodes are present in the sample and only one node is present in the sample. Let k_{max} represent the number of nodes in largest community of the sampled graph and k^e_{max} represent the number of edges incident to nodes in largest community.

For the case where both nodes are in the sample, it is bounded by the computation of the

structural quality and the attributes quality. The time required for computing the attributes quality is $O(k_{max})$ while that of the structural quality is $O(k_{max} + k^e_{max})$. It is therefore reasonable to say when both functions are present is bounded by $O(k_{max} + k^e_{max})$.

When only one node is present, two main functions are required to process an edge: determining the community of a new node and determining the node to evict. Determining the community of new node requires $O(k_{max})$ while determining the node to evict in network can be done in constant time if we keep track of each node's contribution to its community.

4.4.5 Limitations

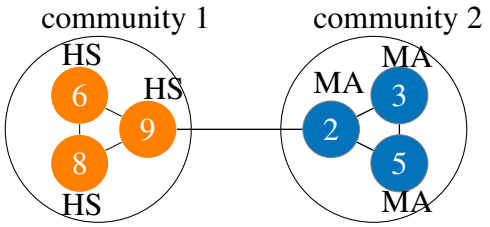
Louvain communities as ground-truth. As mentioned earlier, the communities obtained via sampling is compared to those that were obtained by applying the Louvain method to the non-streaming graph. Subsequently, our results could be interpreted as returning communities identified by the Louvain in its best case. However, the Louvain method is one that has been widely used in identifying communities when the entire graph is available in the community detection literature [86, 87, 13].

Presence of useful attributes. One significant part of this work is the consideration of nodal information in sampling communities from an edge stream. Whereas nodal attributes do not always characterize communities in networks, there exist considerable instances where the presence of attributes are useful in identifying communities [64, 28, 103, 81].

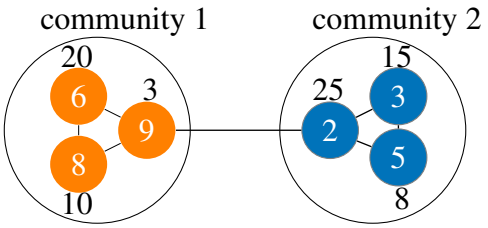
Number of communities selection. `SAMPLearn` assumes the number of communities is known. While this may not be applicable to all cases, there are many instances where such assumption is applicable [47, 8, 42]. For example, one might know some initial members of people with varying interest in different sporting activities. The goal will then be to identify other members that are likely going to belong each sporting activity. The number of sporting activities considered will then be considered as the number of communities while the initial members will be considered as the seed set.

4.5 Conclusion

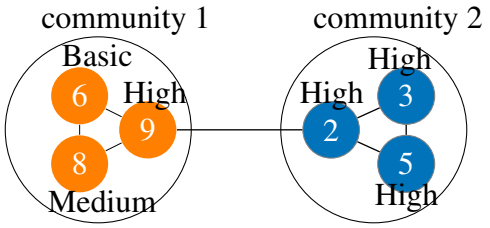
In this work, we propose a novel approach, SAMPLearn, to sample communities from edge streams with nodal information. SAMPLearn leverages nodal information, when present, with the structural information to sample communities in a network. When nodal information is not present or does not characterize communities, it uses only the structural information. Experiments show that our proposed approach almost always outperforms baseline methods, showing improvement of up to about 5 times.



(a) Two groups of faculty from the History (HS) and Math(MA) department with their department as an attribute



(b) Two groups of faculty from the History and Math department with the duration of service as an attribute



(c) Two groups of faculty from the History and Math department with their knowledge of math as an attribute

Figure 4.3: Example illustrating the different relationships that could exist between attributes and members of a community. Figure 4.3(a) represents an instance where an attribute characterizes the members in all communities, Figure 4.3(b) illustrates the case where an attribute does not characterize any communities in the network and Figure 4.3(c) represent the case where an attribute characterizes some communities.

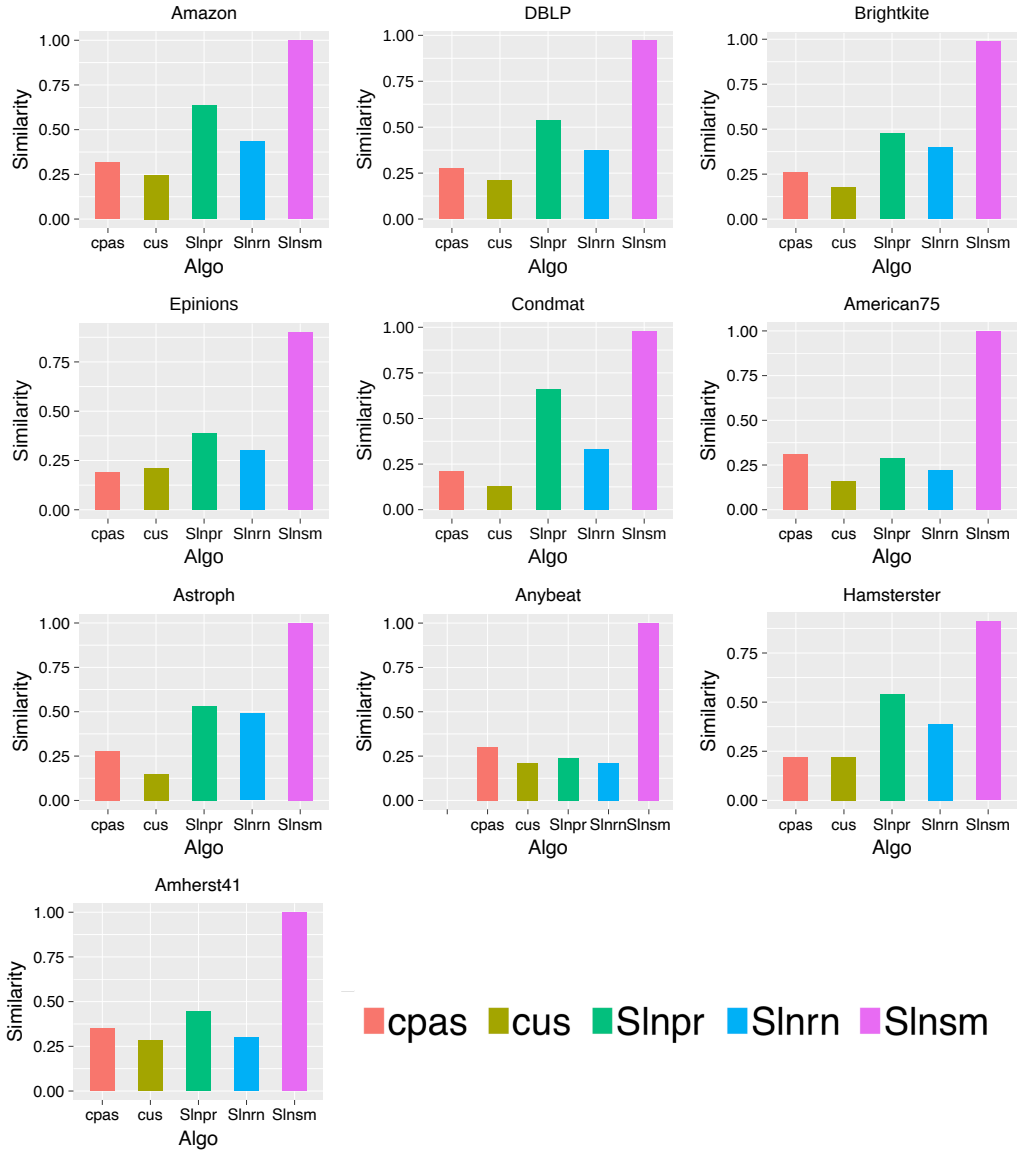


Figure 4.4: Similarities between the sampled and true communities for `SAMPLearn` with attributes characterizing all communities (Slnsm), `SAMPLearn` with attributes characterizing half of the communities (Slnpr), `SAMPLearn` with attributes characterizing none of the communities (Slnrn), `COEUS` (cus) and `COMPAS` (cpas). `SAMPLearn` outperforms the baselines in almost all cases.

CHAPTER 5
CHARACTERIZING EVOLUTION OF COMMUNITIES

In this chapter, our goal is to investigate and characterize the different patterns that exist in the evolution of communities with respect to the number of active users. We hypothesize that the different evolution patterns of communities are related to the behavior of members of these communities.

As a first step in understanding the behavior of members of a community, we examine the different behaviors of members of a single community (the `changemyview` community). Specifically, we examine the language usage and interaction dynamics of members within the `changemyvew` community. We present our findings on the study of `changemyview` in Section 5.1.

Next, we then examine how the behavior observed in our `changemyview` studies (the interaction dynamics and language usage) can generalize to characterize the evolution of the number of active users in communities. The findings on the evolution of communities is also presented in Section 5.2.

5.1 Characterizing susceptible users on reddit's changemyview

There is a growing interest in understanding persuasion processes in various social media platforms, e.g., the influential users in an online community [66, 63], the types of persuasion attempts [7] and the indicators of a social media comment's persuasion power [79, 40, 96]. The majority of these research activities have focused on the side of pursuing persuasion, with only a few studies that examine the other side - those who are being persuaded [79, 88, 90].

With the goal of identifying the properties of susceptible and non-susceptible individuals, we analyzed the Reddit `changemyview` subreddit. In this subreddit, the author of a post makes a submission on an opinion and seeks comments from other users to change her opinion. If a user is successful in changing the author's initial opinion, the user is awarded a point referred to as a `delta point`. As an author of a post on `changemyview`, you are required not only to issue a delta point when your opinion changes but also required to explain the reasons for the change in opinion.

In the context of Reddit `changemyview` discussions, what then could be possible sources of information in characterizing the original posts? We consider three broader sources of information in segregating the submissions: (1) the prior position of the author regarding the topic. (2) the interactions between the author of a post and their challengers (those individuals that interact with the original author in an attempt to influence his or her views). (3) the language use in the post. According to [104], authors of articles in web-based communication channels are more likely to live their own "writeprints" because web-based channels are relatively casual in comparison to formal publications. We therefore believe an author's contains "writeprints" that could characterize how susceptible an author might be. With the interactions between an author of a post and their challengers, work suggested that an author's interactions with other users could be a useful means of segregating the

authors.

Understanding the features that are indicative of an individual's susceptibility is useful in many regards. In the study of influence, identifying traits that characterize susceptible and non-susceptible users provides useful insights to understanding how different people can be influenced. For example, suppose user A is attempting to persuade two other users, B and C. If, for instance, user A finds out that user B is not somebody who typically changes their mind on a particular subject, but user C is one who is very susceptible, this means that the amount and style of persuasion as far as users B and C are concerned should be different, as a lot of effort will be required to persuade user C. In cases where user A does not have any information on the susceptibility of the users, then they are likely to be treated equally.

In this work, we explore features which can significantly separate users that change their mind all the time and users that never changed their mind on the Reddit subreddit, `changemyview`. Experiments showed that various authors have unique features that can aid in identifying how susceptible an author is to an opinion change. With respect to language use, susceptible users use more punctuation in their writing than non-susceptible users. They also demonstrate more uncertainty in their writing than non-susceptible users. In the interaction of authors and other users, we observed that users that changed their mind most of the time are interactivity at the early part of a conversation in comparison to users that never change their mind.

5.1.1 Related work

Researchers have been interested in studying the behavior of social media users in different contexts. Steurer and Trattner [75] studied the interactions among users in online social media. Specifically, they studied reciprocity of communications among different users and observe that there are always some features characteristics that can in aid in the inference of reciprocity. With a similar goal of inferring the reciprocity of a communication between

two individuals, the authors in [22] suggested that different features such as in-degree, the number of incoming and outgoing messages, etc. have high predictive power in relation to reciprocity prediction. In identifying spammers, Tan et. al[80] posit that user generated content spammers are characterized by some unique features. One notable behavior was that most user generated content spammers makes posts that contains links to other websites. The authors in [39] provide a survey of some of the past works on understanding user behavior for various tasks. Some of the tasks discussed include the study of behavior of users in Online Social Networks (OSNs) and its relations to traffic activities, the study of user's behavior and their reaction to spam.

An area of research that is closely related to susceptibility is the concept of persuasion. A lot of research has been conducted to understand the factor behind a message's persuasiveness. Various theories and models have been proposed to explain the role of contextual factors, such as social judgement [59], elaboration likelihood model [18], inoculation theory [53], cognitive dissonance [29], and narrative paradigm [91]. The aspects of a message's content that indicate its persuasive power have also been explored, such as its structure, comprehensibility, and credibility [55].

Different works have been done in studying persuasion in different forms. Jaech et. al [37] investigated how languages affects the reaction of members of a community. A support vector machine (SVM) model was trained using different features to predict the rank order of a list of comments. Some of the features used included the similarity of a comment to the original post, word count and usage of urls. It was observed that the usage of language features can improve the comment ranking task in different subreddits. Authors in [40, 79, 96] found that certain linguistic properties of comments are indicative of the persuasion power of a text.

Some of the identified features from these studies overlap. For example, all three studies suggest that the sentiment level of persuasive comments (i.e., emotional tone) is lower than that of non-persuasive comments. [40] and [96] found that persuasive comments tend to

use more punctuation marks including periods, commas, colons, dashes, and apostrophes, but less on question marks.

There are also features that show contradicting indications across the studies. [96] found that non-persuasive comments tend to be longer and use words that have six letters or very slightly more, contradicting the results from [40]. While persuasive comments used fewer parentheses in [96], they used more in [40]. Also, while persuasive comments had less cognitive processing in [40, 96] showed the opposite. [96] offered explanations of the observed discrepancies and speculated that these are due to the two different discussion contexts in the two studies, namely, the Reddit `changemyview` discussions vs. Wikipedia's Article for Deletion discussions.

Besides these surface level linguistic features, prior studies also discovered that the structure of the comment helps characterize the persuasion power of the text. For example, the authors in [105] showed that argumentation based features such as the number of connectives in a comment are indicative of persuasiveness at early part of a conversation. And Tan et. al [79] observed that there are different features that can characterize persuasive argument. For instance, it was observed that users that enter a conversation very early are more likely to succeed in a persuasive argument.

There are few works that studied susceptibility of users. [88] investigated various features that are indicative of a Twitter user's susceptibility to tweets from social bots, such as network features, linguistic features and behavioral features. The authors observed that susceptible users interact more with other users, they tend to be more open and demonstrates more affection than non-susceptible users. With a similar goal of identifying users that are susceptible to social bots on Twitter, [90] examined features that are indicative of how susceptible users are to social bots. Authors observed that a Twitter user's Klout score, friends count, and follower count were the top predictors of the susceptibility of a Twitter user to social bots. Klout score is a metric that determines an individual's overall social influence computed using multiple social networking profiles. A fairly recent work by Williams et.

al [95] provides a review on the individual differences and contextual factors that are capable of affecting susceptibility. Authors discuss how different features could have varying impact on different users. For instance, individuals high in self-awareness consider their personal knowledge to a higher degree than others making them less susceptible in some instances. However, self-awareness can make people more susceptible in cases where authors make persuasive charity messages and a user considers herself similar to the author of the post.

Even though our work has the same goal as works done in understanding indicative attributes of susceptibility, our work is unique in that it explores the susceptibility of authors to other users in a conversation.

5.1.2 Methodology

Our main goal is to explore features that are indicative of the susceptibility of users. We use the `changemyview` subreddit for this study.

Data and preprocessing

The `changemyview` subreddit provides a means for individual users to make posts in order to be persuaded into an opinion change by other users on the forum. The author of a post on `changemyview` is referred to as an OP (original poster). When a user makes a comment that successfully changes an OP's initial opinion, the OP replies to the user with an explanation on why the view changed, and grants that user a so-called `delta point`. There are three main ways of indicating a delta point: Δ, `!delta`, `&#8710;`. The `changemyview` subreddit allows users other than the OP to grant delta points if their opinions are changed, but this is rare. The forum specifies rules governing the issuing of delta and how users interact on the platform. In particular, an automated checking bot called *deltabot* ensures that a delta issued for a comment meets the following specifications [1]:

- the delta is not issued from users to themselves;

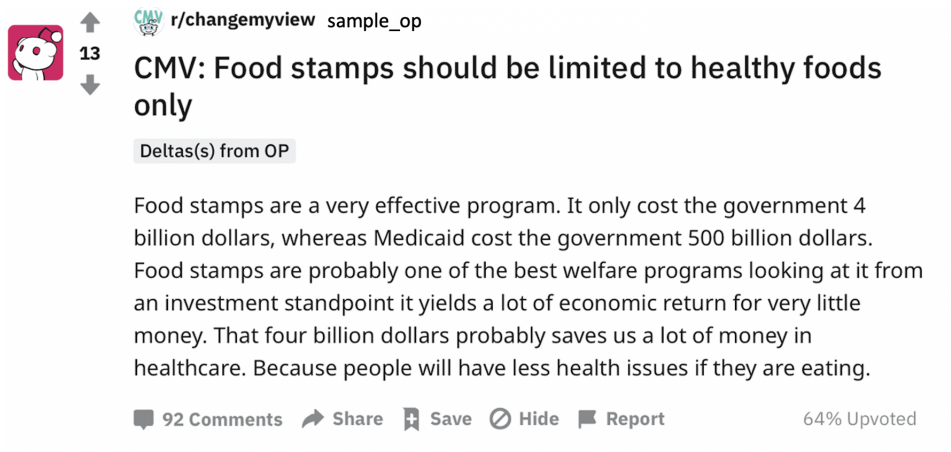


Figure 5.1: A sample post by a user on Reddit seeking opinion change.

- an issued delta is accompanied with an explanation with at least 50 characters of text;
- the delta is not in response to the OP or the deltabot;
- the delta is not in a quote; and
- a delta has not been issued by that same user to the comment already;

Figure 5.1 shows a sample post made by a user seeking an opinion change, Figure 5.2 illustrates portion of attempts by other users to change the mind of an OP and Figure 5.3 shows an example opinion change satisfying all the rules of `changemyview`. The names of users are replaced with dummy names due to privacy concerns. The data used for this project was extracted from conversations made between January 2014 and December 2016. After excluding submissions with no text and/or no comments, a total of 212,404 submissions remained from 13812 unique OPs. We only considered submissions by those OPs that made at least two submissions, leaving 2,821 OPs that made a total of 10,549. We assume that OPs that made exactly one submission may not yet have a full grasp of how the forum works, and so may not understand the delta point system.

We categorize a submission as one on which the OP had a change of opinion only when a delta is issued by the OP of that submission and has been confirmed by the deltabot.

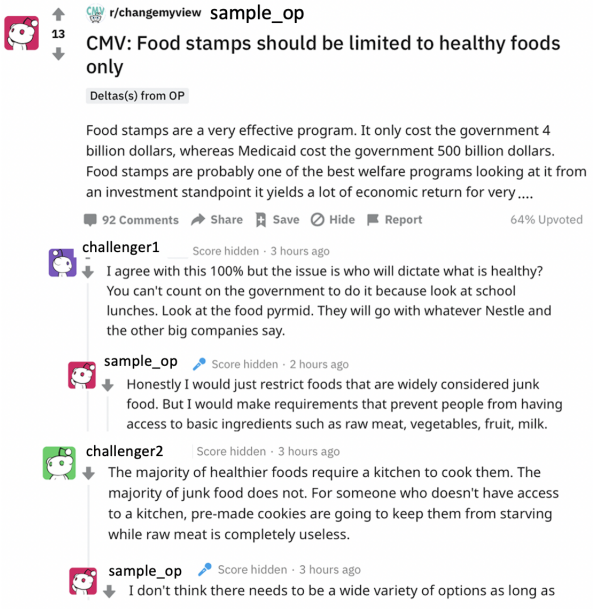


Figure 5.2: A sample of attempts being made by users to change the mind of the post in Figure 5.1.

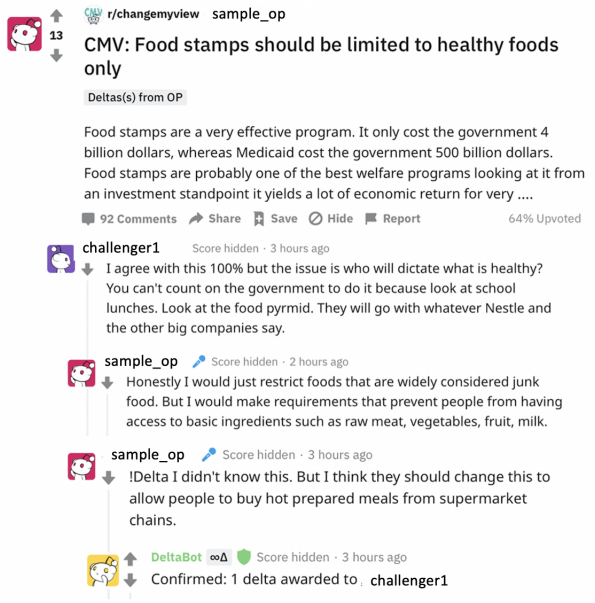


Figure 5.3: An example mind change on the post in Figure 5.1 satisfying all the rules of `changemyview`. The OP was at least partially convinced by a comment, and indicated a mind change by awarding a delta point, which was identified by the automated user Deltabot.

Considering the deltabot's confirmation is necessary in that it prevents issuing a delta point without any justification. We ignore delta points issued by users other than the OP because they are not authors of the submission and we could not establish their position before their mind was changed. We consider two groups of OPs: `susceptible` and `non-susceptible` OPs. A susceptible OP is one that changed her mind on all submissions that she made, and a non-susceptible OP is one that never changed her mind on any of the submissions made. Even though the majority of OPs fall in the middle group of sometimes changing and sometimes not changing their minds, we choose to exclude such OPs. We make this decision because we believe that by studying the extreme groups, we will gain better insight into the factors behind susceptibility. 220 OPs were categorized as susceptible and 1,222 OPs as non-susceptible. A total of 474 submissions were made by susceptible OPs while 2,917 submissions were made by non-susceptible OPs.

Characterizing susceptible users

After identifying appropriate data, the next task is to identify the features indicative of how likely it is for an OP to have a change of mind. As discussed earlier, we consider three possible sources of features: The prior position of the author, interactions between OPs and their challenger, and the language use in an OP's post.

Language usage by an author: To study the language usage of an OP, we perform Linguistic Inquiry Word Count (LIWC) analysis on the submissions. LIWC has 93 features corresponding to different language dimensions. Some of these dimensions are pronouns, authenticity, verbs, positive emotions, negative emotions, etc. Previous work[88] investigated how susceptible users were to social bots. This work suggested that there are some linguistic properties that can characterize how susceptible humans were to social bots. We believe in studying how susceptible humans are to other humans, there will also be some linguistic properties that can characterize their susceptibility. We use the Linguistic Inquiry and Word Count (LIWC) tool to identify linguistic features that characterizes susceptible

users. LIWC uses a word counting strategy to assign a score to a submission in different dimensions. We apply LIWC to submissions from susceptible and non-susceptible OPs. For comparison of submissions from the two groups on the different LIWC categories, we perform a non-parametric test of significance (Mann-Whitney Test). We selected a non-parametric test based on a kurtosis test.

The LIWC categories that showed significant differences ($\alpha = 0.0005$) are shown in Table 5.1. We use an initial α level of 0.05, which corresponds to a Bonferroni α of 0.005 after correction. The LIWC results indicate that users that changed their opinion generally use more punctuation relative to users that never changed their mind. We believe the use of punctuation in an individual's piece of writing makes that piece easier and clearer to understand. For example, consider the two sentences below:

S1: Let's eat John.

S2: Let's eat, John.

Even though the two sentences have the same words, the one with punctuation (S1) is clearer to understand than that of S2. Using more punctuation is therefore likely to make an OP's opinion more clearer to understand. If users clearly understand the opinion of an OP, then one of these might succeed in changing the OP's opinion and hence a possible reason why susceptible OPs use a lot of punctuation.

The authors in [62] posited that a user that is excited about a concept is likely to attract other users to that concept, and used exclamation mark usage as a means of measuring enthusiasm. It is therefore reasonable to argue that OPs that use more exclamation marks have a higher tendency to attract more users to their conversation. By attracting more commenters, the OP is likely to get diverse opinions from different users within which one might be successful in changing the OP's opinion.

According to [58], the analytic category in LIWC measures the degree to which one uses words suggesting a higher level of formal, logical, and hierarchical thinking. In [58],

Table 5.1: LIWC categories that showed significant differences between the two groups of users (users that changed their mind all the time and users that never changed their mind)

	susc	non-susc
Allpunc	++	–
Exclam	++	–
Colon	++	–
Comma	++	–
SemiC	++	–
WC	++	–
Insight	++	–
Verb	++	–
Pronoun	++	–
I	++	–
Personal pronoun	–	++
Nonfluency	–	++
Analytic	–	++

the authors used analytical thinking as a feature in characterizing suicidal Twitter posts. Submissions from an OP with higher analytical thinking might be difficult for other users to actually understand the OP's opinion to even attempt to change that. Also, even if lots of users attempt to change the opinion of an OP, only few of them might really understand what the OP really means, and a lot of effort will be required for OPs with such level of thinking to give in during a discussion. This could be a possible reason why users that displayed higher forms of thinking never changed their mind on any submission.

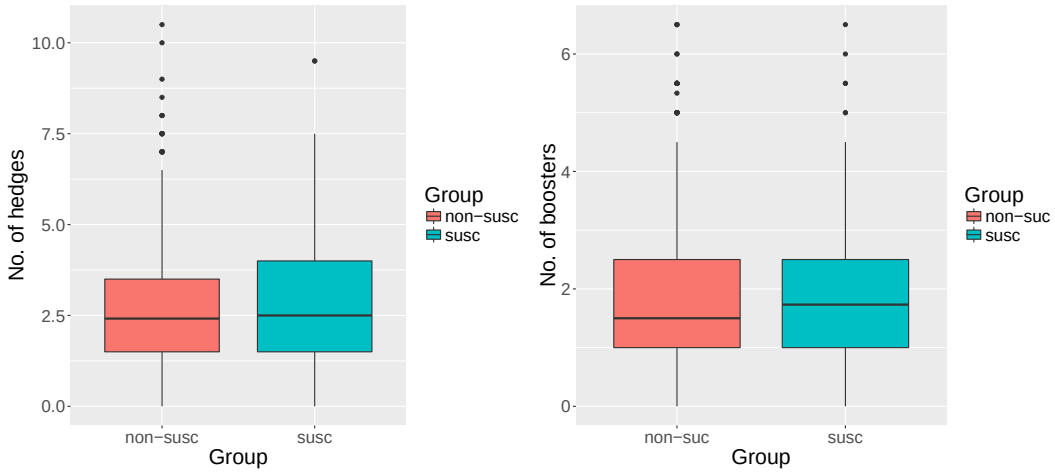
The "I" category of LIWC captures one's usage of first-person singular pronouns. The usage of many first-person singular pronouns indicates the drawing of attention to one's self. Our results indicate that users that used more first-person singular pronouns were more likely to change their opinion. This corroborates a previous finding in [79] that suggested that that people who use a lot of such pronouns are likely to be influenced during a discussion.

Prior position of an author: We estimate an OP's prior position on submission by examining an OP's confidence on a subject. We explore the confidence of an author by ex-

amining the usage of hedge words (hedges) and booster words (boosters). Hedges refer to words that make issues difficult to understand [35, 44]. According to [82], people that are uncertain tend to use lots of such words. Boosters on the other hand refer to words used to express conviction and an indication of confidence in an asserted proposition. With hedges and boosters, we count the number of hedges and boosters used in an OP's submission. The hedges and boosters used in the experiment was provided by [36].

Figure 5.4(b) shows a box plot of the usage of hedge words and booster words by OPs. We observe that, on average, OPs that changed their mind use hedges more than those that never changed their mind. This observation is reasonable in that if an OP is uncertain with her opinion, then compared with an OP who is certain, the one with less certainty is more likely to change her opinion. This corroborates findings in [82]. For boosters, the expectation was that the confidence expressed by an OP could possibly deter other users from attempting to change the OP's opinion and hence succumbing to the view of the OP. However, that was not observed in the experiment. The insignificance in the usage of boosters among the two groups could be that OPs generally do not reveal how confident they are on a subject matter in their submission.

OPs interactions with other users: The authors in [79] showed that the interactions among users during a conversation in online social media is a significant source of information in identifying users that can succeed in successfully persuading other users. In [102], the authors find that debaters who follow up on points brought up by their opponents have higher chance of winning. These results suggest that the interaction dynamics of a conversation between the author of a post and other challengers could be a useful source of information in segregating susceptible and non-susceptible OPs. We considered three features as a way of capturing the interaction dynamics between an OP and other users: the number of unique users an OP engages in a back and forth with, the frequency of an OP's response, and when during a conversation are OPs active. Figure 5.5 shows a toy graph with the OP represented with a square and challengers as circles. Back and forth is defined



(a) Box plot of the usage of hedge words.

(b) Box plot of the usage of booster words.

Figure 5.4: OPs that changed their mind all the time (susc) used more hedge words that never change their mind (non-susc) from the significance testing. There is no observed significant difference in the usage of booster words between OPs that changed their mind all the time (susc) and OPs that never changed (non-susc)

as the OP replying back to a user that made a comment on the OP's submission, and the frequency of an OP's response is defined as the number of times the OP commented on another user's comment excluding delta replies. OPs are said to be active when they respond or comment on a user's post. The duration for conversation considered is the period between the first and last comment received by an OP after a submission was made. The duration for each conversation is partition into three parts: the early part of the conversation, the middle part of the conversation and the latter part of the conversation. For each part of the conversation, number of times an OP responded to other users is computed.

From our experiments, we observed that users that never changed their mind engage in more back and forth with their audience than users that changed their mind all the time $(p = 0.0004)$. Among all the users that have made a comment on a submission made by an OP, if the OP engages in back and forth with just one of these users, then it could be argued that the back and forth can provide clarity to the opinion of the OP and hence the likelihood of OPs changing their initial opinion. However, if an OP engages in back and forth with one user and never changes the mind but instead engages several other users in such back

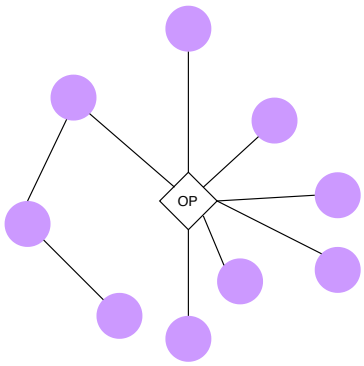


Figure 5.5: Illustration of the interaction network of an OP and users attempting to change OP's opinion.

and forth, such OPs are then less likely to change their opinion because they might be so firm in their opinion hence the reason why even when people try explaining their points, they never give in.

Also, we observed that OPs that changed their mind most of the time frequently interacted with others more than OP's that never changed their mind ($p = 0.004$). If an OP responds or comments on another user's post, then either the OP is seeking some clarification on the opinion of the user or simply disagrees with that opinion and is attempting to explain the reasons for her disagreement. For either case, it is reasonable to say that the OP is somewhat paying attention to the user's opinion. An OP who is indifferent to many users in a conversation is therefore less likely to change the view in comparison to one that is paying attention to the views of others. This is because if an OP pays attention to several other users, there is a higher chance that one of the users might make a point which could change the OP's initial opinion.

Figure 5.6 shows the number of responses made by OPs at different parts of the conversation. We observed that even though all OPs generally decrease their interactions with other users towards the end of a conversation, users that changed their mind all the time interact more with other users at the early and middle part of the conversation. Previous work [79] had suggested that users that enter a conversation late after the submission has been posted is less likely to succeed in changing an OP's opinion in comparison to users

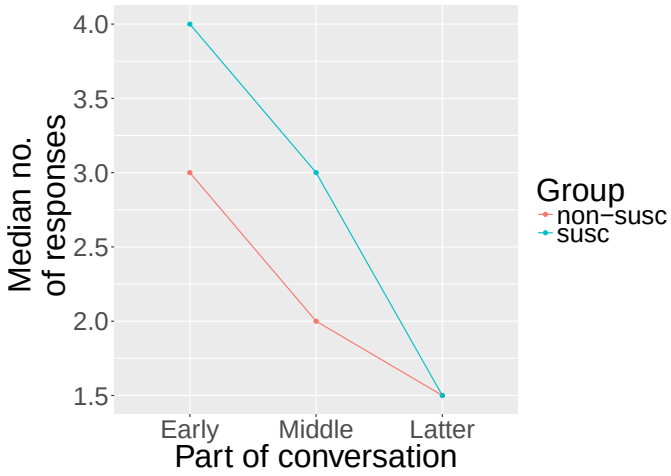


Figure 5.6: The number of interactions OPs made with other users at different parts of the conversation. Susceptible OPs engage with challengers more at the early part of the conversation

that enter early. This suggests that OPs that are susceptible are likely to be active at the early part of the conversation. If an OP is susceptible, then after having an opinion change, the OP might not be as active as she was before since there is an opinion change.

5.1.3 Conclusion

In this work, we investigated features that are useful in segregating susceptible and non-susceptible OPs on reddit's `changemyview`. We explored three main sources of information in characterizing users on this forum (1) the OP's language usage in a post (2) prior stance of the OP before seeking an opinion change and (3) the interactions between OPs and their challengers. For the prior stance of an OP, we explored how much confidence is expressed by an OP in a submission. In measuring confidence, we used an OP's hedge/booster words usage as a way of characterizing the confidence. For interactions between an OP and their challengers, we explored the number of unique users the OP engages with back and forth, the number of responses made by an OPs and their challengers and which part of a conversation are OPs active. We performed LIWC analysis as a means of understanding an OP's language usage in a post.

Experimental results showed that OPs who never changed their mind are more analytical in thinking when writing than susceptible OPs. Also, susceptible OPs use more hedge words than non-susceptible users. This means OPs who changed their mind most of time have more uncertainty in their submissions than non-susceptible users. On an OP's interaction with other users, susceptible users tend to interact with their challengers more at the early part and middle part of the conversation.

Our goal is to discover the differences between susceptible and non-susceptible users in their digital traces. Subsequently, our comparison of their submissions is intended to discover the differences in language use between these two groups of users. On the other hand, the grouping of these submissions can also be interpreted as merely by whether or not OPs changed the original view. This implies that the differences we observed could be interpreted as merely the differences between the two types of submissions, not the two types of users.

5.2 Characterizing the evolution of communities

From Chapters 3 and 4, we learned different communities have varied evolution patterns. In this section, we investigate the possible evolution patterns with respect to the number of active users in communities, and what characterizes the different parts of these patterns, drawing insights from studies conducted on the `changemyview` community in Section 5.1. Understanding the evolution of communities in useful in many aspects. For example, understanding the various evolution patterns of a community can aid in building and maintaining successful communities [41, 43, 78].

To study the evolution of communities, we use data from Reddit. Reddit is a social news website and forum where users are organized into communities referred to as subreddits. Most communities are public and consists of members who share a common interest [78]. For each community, users can upload pictures, text or both. An example community

is askscience, where people post science related questions seeking answers. Most subreddits have rules governing the operation of the community. For example, askscience requires members to ask questions concisely. To answer questions on askscience, members are required to respond accurately with peer-reviewed sources where possible.

With the goal of identifying factors that can significantly differentiate the different parts of a community's evolution, we begin by first identifying the patterns that can exist in the evolution of the number of active users in a community. After identifying the patterns, we consider two possible sources of information in characterizing the communities at different points in their evolution: (1). The interaction style of users in a community (2) The language usage by members who initiate conversations. Specifically, we consider how members interact at different parts of a conversation and the duration of a conversation to understand the interaction style of users. On the language usage, we consider the linguistic style of members who initiate conversations in the community.

In this work, we explore the various factors that can significantly distinguish the different parts of the evolution pattern of communities. Firstly, we identify the different evolution patterns. We then examine how a pattern differs at different points with respect to the interactions of members and the language usage by members who initiate conversations. Our experiments show that communities have unique features that can distinguish the different parts of their evolution. On interaction style, we find that the middle part of a conversation and how long a conversation last can significantly separate the different parts of a community's evolution with respect to the number of active users. Regarding language usage, we find that factors such the demonstration of leadership and positive emotions can separate different parts of a community's evolution.

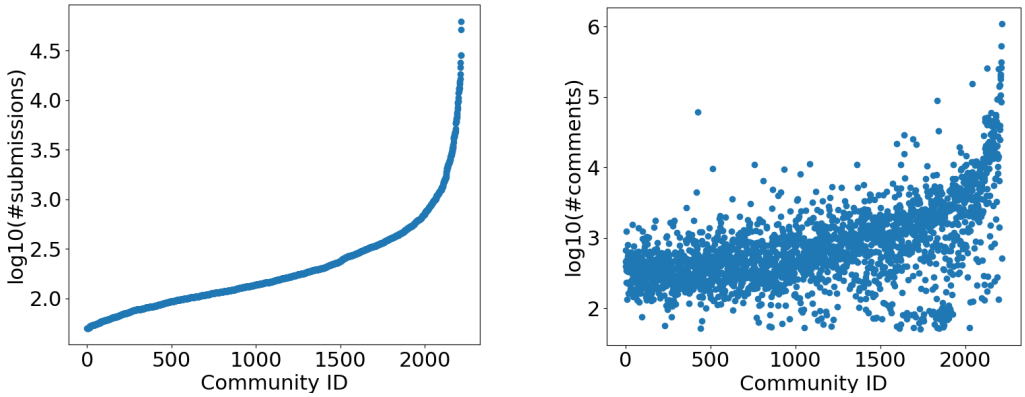
5.2.1 Methodology

Our goal in this study is to understand the factors that characterize the different parts of a community's evolution.

Data and preprocessing

As mentioned earlier, the data used for this study are communities on Reddit. We obtain Reddit communities that were created in 2014 and 2015. Different communities on Reddit have different modus operandi. While some communities allow members to post textual contents and upload images, there are other communities that allow only the upload of images. We focus only on communities that allow its members to post text. We believe this will enable us to compare communities with same mode of operation. For example, if we consider communities that allow only the upload of images, it wouldn't be fair to compare the language usage of members in such communities to that of communities that allow text only. In order to examine the evolution of communities, we consider the record of a community within 1 year of its creation. We consider a monthly timestep for all the communities studied. For the evolution of communities, we consider the number of active users at each timestep. For us to better understand communities' evolution before and/or after the communities become inactive, we consider only communities that were inactive at some point in their evolution. Also, there exist some communities that have very few members within a year of its creation. We consider such communities as uninteresting. In order to exclude such unexciting communities, we only consider communities that attracted at least 100 members within three months. After the preprocessing, we were left with 2430 communities. These communities had 1,172,662 submissions with 8,686,646 comments from 969,120 unique users. Figure 5.7(a) shows the distribution of submissions for each community and Figure 5.7(b) shows the distribution of comments for each community.

The evolution of each of the 2430 communities over 12 timesteps is studied in this work. In order to ensure fairness in the comparison of evolutions of different communities, we normalize the number of active users at each timestep to a value between 0 and 1. For a given community C^i, the normalized value $V_t^{norm}(C^i)$ of the number of unique users active at timestep t is given by



(a) Distributiotion of submissions (b) Distribution of comments

Figure 5.7: Distribution of the number of submissions made in each community (Figure 5.7(a)) and the number of comments received in each community(Figure 5.7(b)). The x axis represent a community's id while the y axis indicate the number of submissions and comments on a logarithmic scale respectively made in each community.

$$V_t^{norm}(C^i) = \frac{V_t(C^i) - min(H(C^i))}{max(H(C^i)) - min(H(C^i))}, \tag{5.1}$$

where $V_t(C^i)$ is the number of unique users active at timestep t and $H(C^i)$ is a list of active users for the 12 timesteps for community C^i.

Identifying evolution patterns

To be able to characterize the different evolution patterns of communities, we need to first identify these patterns. We identify the patterns by computing the similarity between the evolution of different communities. This matrix of similarity is then grouped to determine the cluster of evolution patterns that exist.

The similarity of the evolution of two communities is computed using a popular time series similarity technique referred to as Dynamic Time-Warping (DTW) [69]. DTW was selected because of its ability to capture similarities between two evolutions regardless of the speed of evolution. The DTW between community C_i at time k and community C_j at time l is defined as

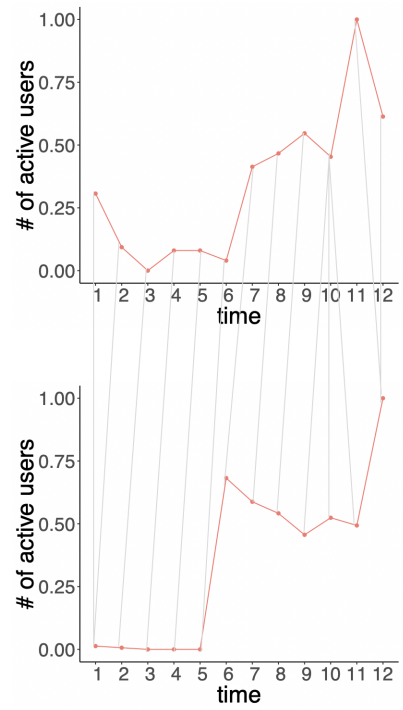


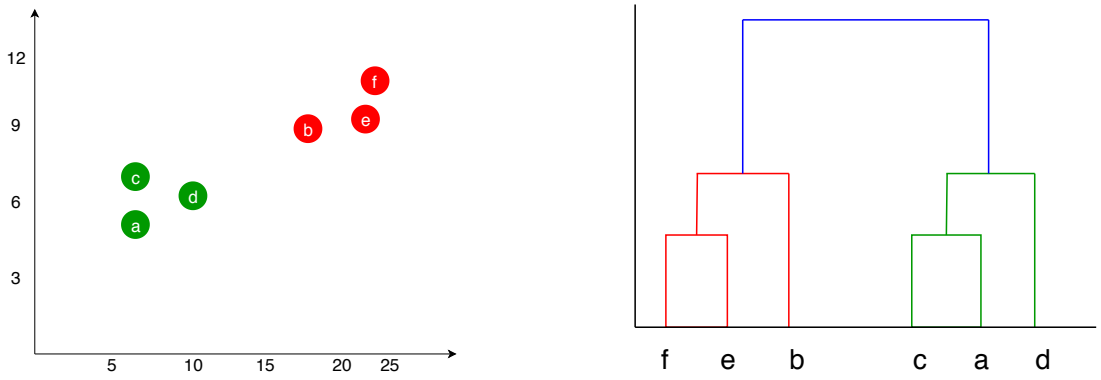
Figure 5.8: DTW similarity between two growing communities

$$DTW(C_i^k, C_j^l) = min(DTW[C_i^{k-1}][C_j^l], DTW[C_i^{k-1}][C_j^{l-1}], DTW[C_i^k][C_j^{l-1}]) + d(C_i^k, C_j^l), \tag{5.2}$$

where C_i^k is the number of active users of community C_i at time k and $d(m, n)$ is the absolute difference between m and n. The sum of all similarities between the timesteps represent the similarity between community C_i and C_j. Figure 5.8 shows the mapping of two community based on the similarity between their timesteps computed using DTW.

After identifying the similarities between communities' evolutions, we then group these communities based on their similarities. We use hierarchical agglomerative clustering to group the evolutions [93]. Hierarchical agglomerative clustering is a technique used to group data points into clusters. The process begins with each point considered as a cluster. At each step, a cluster is formed by combining the two closest points. The process is repeated until one single cluster is formed. Dendrograms provide a means to visualize the groupings to make decision on the optimal cluster size. Figure 5.9 shows the application of

hierachical agglomerative clustering in grouping six points (Figure 5.9(a)). The resulting dendrogram from the grouping is shown in Figure 5.9(b). The process begins with merging points `f` and `e`. The two points (`f + e`) are then merged with `b` to form another cluster. Similary, points `c` and `a` are initially merged. These points (`c + a`) are then merged with `d`. The clustering technique at each iteration merges points resulting in minimum distance which is interpreted as higher similarity.



(a) Six points `a,b, c, d, e` and `f` in a two dimensional space.

(b) Clustering of `a, b, c, d, e` and `f`.

Figure 5.9: An example clustering of points using hierarchical agglomerative clustering. Figure 5.9(a) illustrates six points in a two dimensional space while Figure 5.9(b) shows the resulting clusters obtained by applying hierachical agglomerative clustering to these points.

Figure 5.10 shows the similarities between the evolutions of communities studied in this work, the resulting dendrogram obtained by clustering their similarities and sample community evolutions for each of the identified groups. Three clusters were selected as the optimal number of clusters. This is because it has the largest vertical distance that does not intersect any of the other clusters. We interpret the three clusters as communities that start to increase in number of active users from some point forward during the evolution (growing communities), those that start to decrease in number active users from some point forward during the evolution (failing communities) and communities switches between increasing and decreasing of their number of active users (unstable communities).

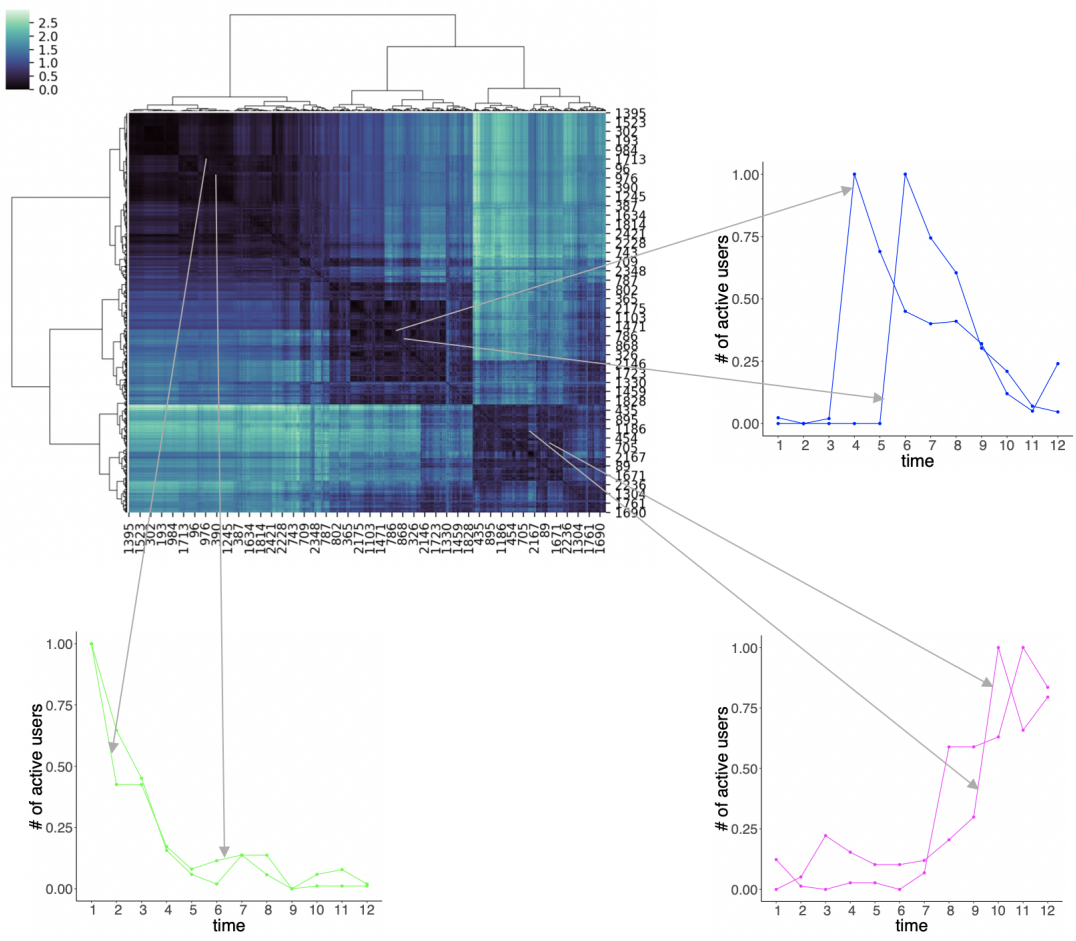


Figure 5.10: Clusters obtained from grouping the evolution patterns of communities on Reddit. Three clusters were selected as the optimal number of clusters. We interpret the three clusters as communities that increase in the numer of active user at some point forward in the evolution (pruple); communities that decrease in the numer of active users some point forward (green); communities that switches between increasing and decreasing in number of active users (blue).

Characterizing the evolution of communities

After identifying the patterns that exist in the evolution of communities, we investigate factors that can significantly distinguish between the different parts of the evolution. In order to examine the parts, we first divide the patterns into different parts and investigate how the interaction style of members and the language usage can distinguish these different parts.

For the three evolution patterns identified (growing, failing and unstable communities), we focus on growing communities (G0) and failing communities (G1). We focus on these two groups because the third group is a combination of the failing and growing pattern. Each evolution pattern is divided into 4 parts:

1. Peak point (PK): The peak point is the timestep where a community reaches the largest number of active users. For the growing communities, we assume the last point of the evolution is the peak point. If a community is truly growing, then we believe it is reasonable to assume the last timestep as the peak point. For the failing community, we assume the starting point of the evolution is the peak point. With a community declining in the number of active users, we believe assuming the starting point to be the first timestep is equally reasonable.

2. Elbow point (ELB): For a growing community, this is the period after which the community begins to increase in the number of active users. For a failing community, this is the period after which a community begins to decrease in the number of active users over time.

3. Peak interval (PK interv): This is the period between the peak point and elbow point.

4. Elbow interval (ELB interv): For the growing communities, the elbow interval is the period between the elbow point and the period when the evolution started. With the failing communities, this is the period between the elbow point and the end of the evolution.

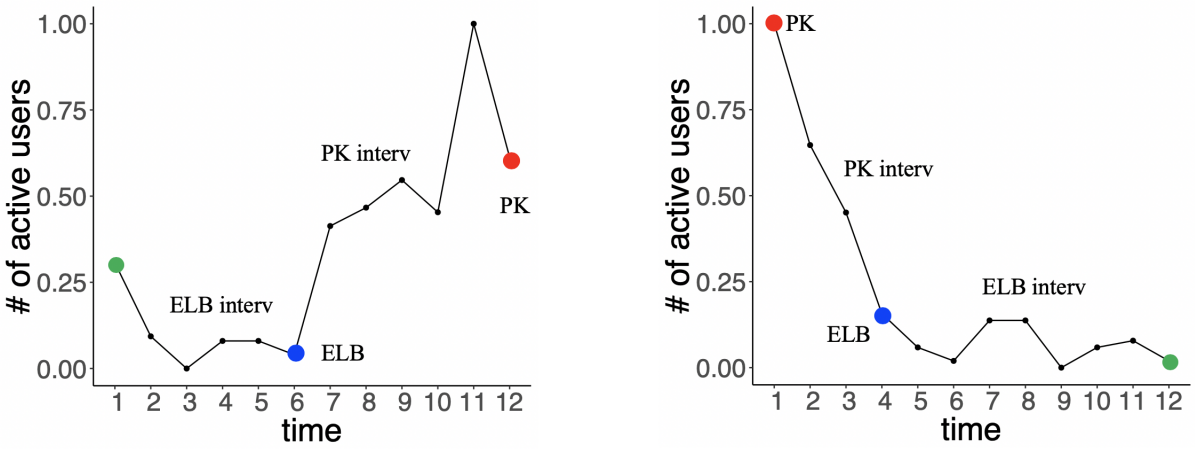


Figure 5.11: Parts of an evolution pattern for growing community (Left) and failing community (Right). Each evolution is divided into 4 parts: peak point (PK), elbow point (ELB), elbow interval (ELB interv) and peak interval (PK interv).

Figure 5.11 shows the various of parts of the evolution of a growing community and a failing community.

As mentioned earlier, we consider two sources of features. The interaction style of members within a community and the language usage of members who initiate conversations.

Interaction style of members: In Section 5.1, we learned the different interaction style of a person on Reddit's `changemyview` can characterize how susceptible the person is to an opinion change. Also, in [9], authors identified how the early interaction style of members in a conversation on Facebook can characterize how likely it is for the post to attract more comments. We therefore postulate that the interaction dynamics of users can distinguish the different parts of a community's evolution. We consider two features as a means of capturing the interaction dynamics at some point of a community's evolution: duration of a conversation and when members of a community are active during a conversation. The duration of a conversation is the period between when a submission was made, and the last comment made by a user on the submission. In order to minimize the effect of outlier comments influencing the characterization of a conversation, we consider only interactions

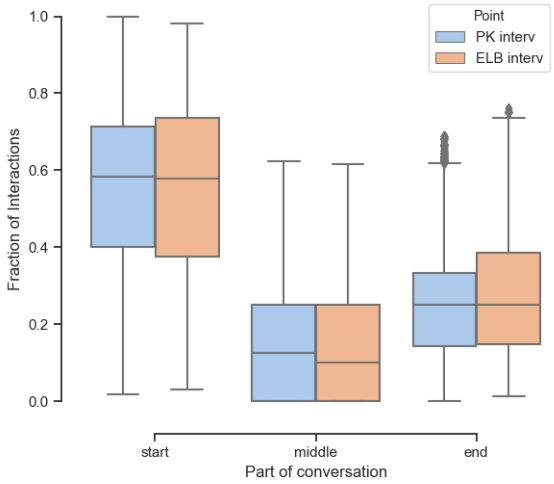


Figure 5.12: The fraction of interactions at the early (start), middle (middle) and latter (end) parts of conversations for growing communities when comparing peak and elbow intervals. Most interactions occur at the early part of conversations. The middle part of conversations has the lowest number of interactions. The peak interval (PK interv) has a significantly larger number of interactions at the middle part of conversation than the elbow interval (ELB interv).

between the 10th to 90th percentile of the time periods within a conversation. For capturing how active members are during a conversation, we partition the duration of a conversation into three parts: early part of the conversation, middle part of the conversation and latter part of the conversation. We partition a conversation into three parts by dividing the interval between the start and end time of the periods considered into three equal intervals. For each part we consider the fraction of interactions within each part of the conversation. For each of these interaction dynamics feature, we investigate if there is any significant difference between (1) the peak and elbow points of conversations. (2) peak and elbow intervals of conversations. We test for significance using a non-parametric (Mann-Whitney Test) test of significance. We selected a non-parametric approach because the assumptions for normality fails from our kurtosis test.

From our experiments, we observed similar interactions style regarding the activities at different parts of a conversation for both the growing communities and the failing commu-

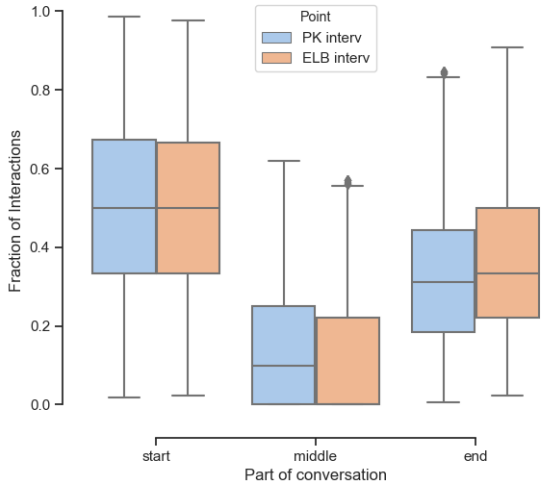


Figure 5.13: The fraction of interactions at the early (start), middle (middle) and latter (end) parts of conversations for failing communities when comparing peak and elbow intervals. Most interactions occur at the early part of conversations. The middle part of conversations has the lowest number of interactions. The peak interval (PK interv) has a significantly larger number of interactions at the middle part of a conversation than the elbow interval (ELB interv).

nities. Specifically, we find that most interactions of a conversation occur at the early part of conversations. The middle part of a conversation has the least number of interactions regardless of whether the community is growing or failing. Even though the middle part of a conversation has the least fraction of interactions for both groups of communities, the fraction of interactions during the middle part of a conversation for the peak interval is significantly larger than that of the elbow interval. Similarly, the fraction of interaction at the middle part for the peak point is significantly larger than the elbow point. We use a Bonferroni corrected α of 0.005. Figures 5.12 and 5.13 shows the fraction of interactions at the different parts for the growing communities and failing communities respectively when comparing peak and elbow intervals. Figures 5.14 and 5.15 shows the fraction of interactions at different parts of a conversation respectively when comparing peak and elbow points. This suggest that for most conversations, users join the conversation either when it starts which is the reason for higher fraction of interactions at the early part and when the

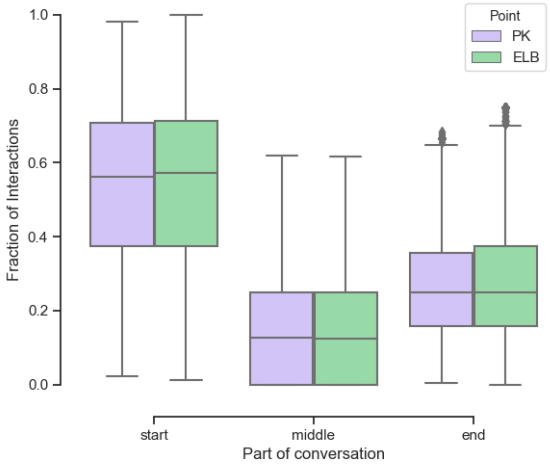


Figure 5.14: The fraction of interactions at the early (start), middle (middle) and latter (end) parts of conversations for growing communities when comparing peak and elbow points. Most interactions occur at the early part of conversations. The middle part of conversations has the lowest number of interactions. The peak point (PK) has a significantly larger number of interactions at the middle part of a conversation than the elbow point (ELB).

conversation is about to end. Since the middle part of the peak portions of communities' evolutions are significantly higher than the elbow parts, it also suggests that communities' ability to attract more people at the middle part of conversations within the community is related to the number of active members of the community.

Regarding the duration of conversations, it was that observed that the peark parts of communities' evolution had significantly shorter conversations than the elbow parts for both growing communities and failing communities. Figures 5.16(a) and 5.16(b) show a plot of the durations for the growing and failing communities when comparing the peak intervals. Figures 5.17(a) and 5.17(b) show a similar plot when comparing peak points of the growing communities and failing communities respectively. A possible reason why the peak parts have shorter conversation could be as a result of the communities' increase in the number of active users. This corroborate previous findings in [9] that the amount of time it takes for a post on Facebook to attract comments is indicative of whether or not the

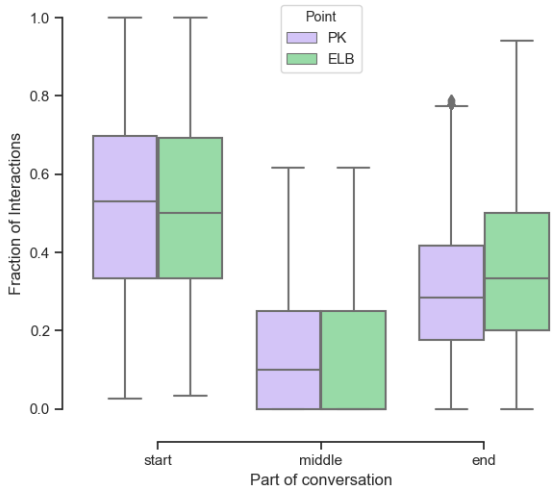
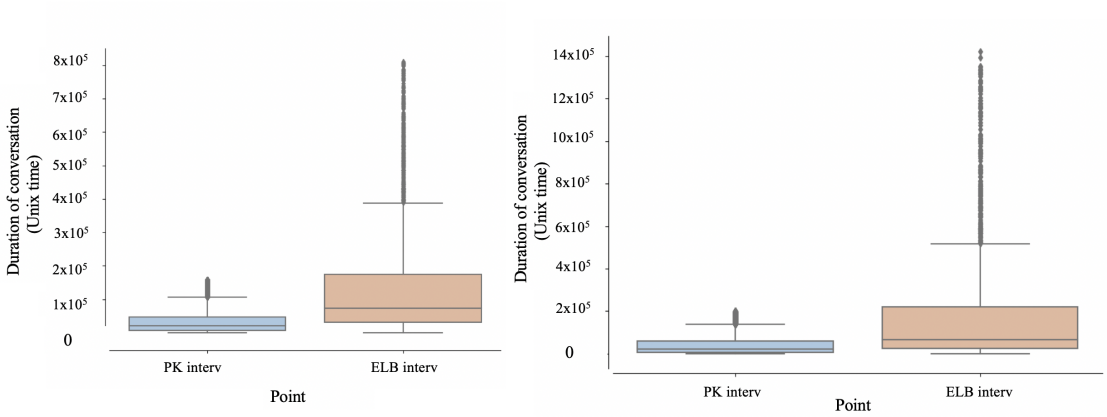


Figure 5.15: The fraction of interactions at the early (start), middle (middle) and latter (end) parts of conversations for failing communities when comparing peak and elbow points. Most interactions occur at the early part of conversations. The middle part of conversations has the lowest number of interactions. The peak point (PK) has a significantly larger number of interactions at the middle part of a conversation than the elbow point (ELB).



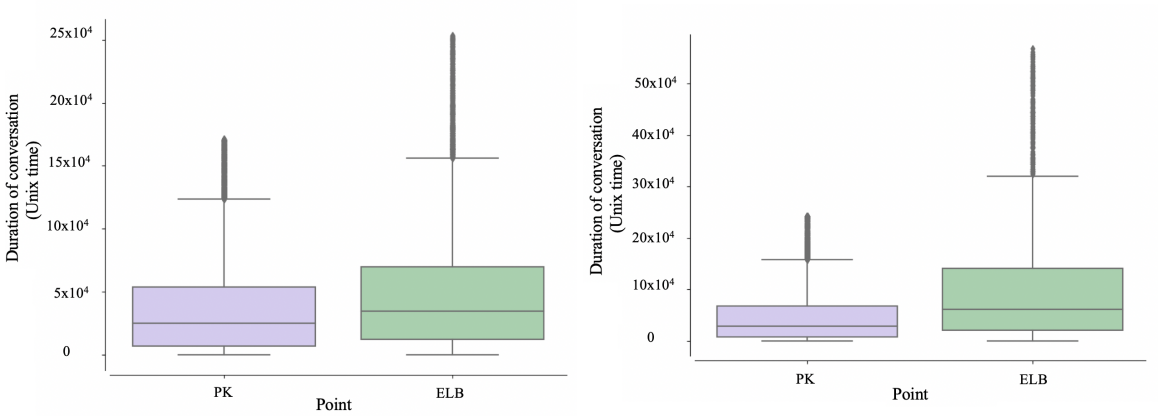
(a) Durations for growing communities

(b) Durations for failing communities

Figure 5.16: Duration of conversations when comparing peak (PK interv) and elbow (ELB interv) intervals of growing communities (Figure 5.16(a)) and failing communities (Figure 5.16(b)). The conversations that started at the peak intervals were significantly shorter that those that started at the elbow intervals.

thread will attract more comments.

Language usage of conversation initiators: The language usage of authors in social media has been demonstrated to be a good source of information in varied applications. From



(a) Durations for growing communities

(b) Durations for failing communities

Figure 5.17: Duration of conversations when comparing peak (PK) and elbow (ELB) points of growing communities (Figure 5.17(a)) and failing communities (Figure 5.17(b)). The conversations that started at the peak timesteps were significantly shorter that those that started at the elbow timesteps.

Table 5.2: LIWC categories that showed significant differences between the peak and elbow intervals of growing communities.

CATEGORY	G0 (PK Interv)	G0 (ELB Interv)	p
clout	45.62	**50**	1.49E-17
sixltr	14.13	**15.235**	1.07E-18
Posemo	2.56	**2.9**	1.07E-06
leisure	0.7	**1.45**	1.62E-36
time	**3.66**	3.33	8.07E-06
cogproc	**10.94**	10.24	8.24E-24

the study in Section 5.1, we learned that the linguistic style of users can characterize the susceptibility of users in the `changemyview` community. In [26], authors demonstrated how linguistic changes in communities are useful in understanding how users react to the evolving norms of a community. Also, the linguistic style of users have been shown to be useful in characterizing users that are loyal to communities[31]. We believe the language used by authors can characterize the different parts of a communities' evolution.

In this study, we investigate how the language style of conversation intiators can characterize the different pats of a community's evolution. We perform LIWC analysis on the submissions made at the different parts of a community's evolution. Tables 5.2 and 5.3

Table 5.3: LIWC categories that showed significant differences between the peak and elbow intervals of failing communities.

CATEGORY	G1 (PK Interv)	G1 (ELB Interv)	p
clout	**63.575**	59.94	0.00015023
sixltr	**16**	15	9.08E-09
Posemo	2.27	**2.65**	1.42E-05
leisure	0.33	**0.83**	4.04E-13
time	3.48	**3.92**	6.64E-06
cogproc	9.47	**10.71**	1.67E-09

present the LIWC category comparison for growing and failing communities respectively when considering peak and elbow intervals. Tables 5.4 and 5.5 present the comparison when considering peak and elbow points. N/A is used to indicate no observed statistical significance for a category.

`Posemo` category of LIWC demonstrates an author's display of positive emotion in writing. Some `posemo` words are happy, pretty, good, love, nice, sweet, etc. A higher score of `posemo` indicates the usage of words related to positive emotions. From the experiments, it was observed that conversations that were initiated at the peak intervals had significantly fewer words related to positive emotions in comparison to the elbow intervals of both the growing (Figure 5.2) and failing (Figure 5.3) communities. A similar observation was made when comparing the peak and elbow points of growing communities. Results on the leisure category which considers words like music, movie, cook, etc. indicates that members are not attracted to conversations about leisure. A possible reason for this could be that at the elbow parts, most of the submissions made by users are about their leisure activities which does not excite users to participate in such conversation. Also, when people demonstrate more positivity in their writings, people are not attracted to such conversations hence the reason why the elbow parts have higher `posemo` and leisure scores than the peak parts.

`Clout` quantifies the demonstration of confidence and leadership in writing. A high `Clout` means that the author of a post demonstrates strong leadership in their submission. For the growing community, we observe that the elbow intervals have significantly higher

Table 5.4: LIWC categories that showed significant differences between the peak and elbow points of growing communities. N/A on a row indicates the category did not show any significance.

CATEGORY	G0 (PK)	G0 (ELB)	p
clout	41.92	**50**	7.06E-40
sixltr	14	**14.29**	9.99E-06
Posemo	2.31	**2.56**	3.2E-13
leisure	N/A	N/A	N/A
time	**4**	3.66	1.01E-19
cogproc	11.5	**11.76**	1.81E-06

Table 5.5: LIWC categories that showed significant differences between the peak and elbow points of failing communities. N/A on a row indicates the category did not show any significance.

CATEGORY	G1 (PK)	G1 (ELB)	p
clout	N/A	N/A	N/A
sixltr	N/A	N/A	N/A
Posemo	N/A	N/A	N/A
leisure	N/A	N/A	N/A
time	3.57	**4.05**	1.18E-08
cogproc	N/A	N/A	N/A

`Clout` than the peak intervals (Figure 5.2). For the failing community, we observe that the peak intervals have significantly higher `Clout` than the elbow intervals (Figure 5.3). For the failing communities, a possible reason for this observation could be that people generally don't like to participate in conversations by people who are opinionated and that could be a reason for the decrease in the number of active users at the elbow interval. For the growing communities, even though it starts off with a higher `Clout`, the overall behavior at the elbow interval is lower than LIWC's mean `Clout` value of 57.95.

Growing communities vs failing communities

From our experiments, there exist some similarities between evolution patterns of growing communities and failing communities. Both patterns of evolution have most members of

the group participating at the early part of conversations and the least members participating at the middle part as shown in Figures 5.12, 5.13, 5.14 and 5.15.

Despite these similarities, there were some observed significant differences between the two patterns studied. For growing communities, it was observed that at the early part of the communities' evolutio where there is less activity (elbow interval), the middle part of conversations does not attract more people in comparison to evolution periods where there are more activity (peak interval). However, for failing communities, when the community starts and has more activity (peak interval), the middle part of conversations at these times attract more people in comparison to evolution times when the communities have less activity (elbow interval). This suggests that a community's ability to attract more users at the middle part of conversations is related to the community's active users over time.

Regarding the duration of conversations, for growing communities, the early part of the communities' evolution where the communities have less activity have longer durations than the latter part of the communities' evolution where there is much activity as shown in Figures 5.16(a) and 5.17(a). Conversely, for failing communities, the early part of communities' evolution where there is much activity have shorter durations in comparison to the latter part where there is less activity as shown in Figures 5.16(b) and 5.17(b). It is therefore reasonable to argue that the shorter durations observed at the various times are as a result of the communities' evolution.

On the language usage of members who initiate conversations in different communities, for growing communities, the early part of evolution where there is less activity has conversations that are more related to leisure and a higher demonstration of positive emotions in comparison to periods where communities demonstrate more activity as shown in Tables 5.2 and 5.4. Conversely, for failing communities, the early part of communties' evolution where there is more activity has conversations less related to leisure and a lesser demonstration of positive emotions in comparison to periods where the communities have less activity as shown in Tables 5.3 and 5.5. It is therefore reasonable to suggest that con-

versations about leisure and those with higher demonstration of positive emotions do not attract people.

5.2.2 Conclusion

In this work we investigated factors that characterize the different parts of the evolution of communities. Firstly, we identified the different patterns that could exist in the evolution of communities. We found that there could be three evolution patterns for communities on Reddit: (1) Communities that begin to grow (increase in number of active users) from some time forward in their evolution. (2) Communities that begin to fail (decrease in number of active users) from some time forward and (3) Communities that increases at some points and decrease at other points in the number of active users. We then partition each evolution pattern into four parts: peak point, elbow point, peak interval and elbow interval. We explore two main sources of information in characterizing the different parts of the groups identified (1) Interaction dynamics of members in a community and (2) Language usage of conversation initiators. Specifically, we consider the fraction of interactions at the early, middle and latter parts of conversations and the duration of a conversation for the interaction dynamics. For the language usage, we perform a LIWC analysis on the submissions made by users at different parts of the evolution to better understand the language usage.

Experiments showed that even though the middle part of conversations generally have the least number of interactions for both communities that are growing and those failing, the fraction of interactions during the middle part of conversations during the times where communities have much activity is signifcantly higher than conversations during the periods where communities experience much activity. Also, during the evolution of communities, the periods where there are more active users tend to have shorter conversation.

Regarding language usage, conversations more about leisure and higher display of positive emotions do not attract users to participate in such conversations. Also, it was observed the extreme demonstration of leadership does not help communities grow.

CHAPTER 6
CONCLUSION

Understanding communities in networks has usefulness in many applications such as advertising, marketing, recommendation, etc. Networks for community studies are often made of millions or billions of nodes which makes it intractable to examine all the communities in such networks. In this dissertation we focused on the problem of sampling communities in networks in different settings with varied constraints.

Firstly, we addreseds the problem of sampling communities in dynamic social networks when there is a limitation on the number of nodes that one could ask for information. In sampling communities in dynamic social networks, we consider two resource constraint scenarios: (1) When there is a limitation on the number of nodes that could be asked for information over the entire period and (2) When there is a limitation on the number of nodes that could be asked for each timestep. We propose a framework `DYNASAMP` to sample communities under the given constraints. `DYNSAMP` begins by first obtaining a sample for the first time step. In subsequent steps, a fraction of the budget allocated for the time step is used to obtain the startup graph. The startup graph is compared with previously discovered graph. If the startup graph is similar to previously discovered graphs, a portion of the budget is saved. Nonetheless, if the startup is not similar to any of the previously discovered graphs, a portion of the saved budget is used to perform more queries to grow

the network. Experimental results show that DYNSAMP has a performance improvement ranging from 35% to 53% when compared to baseline methods when the query limitation is considered over the entire period and 8% - 56% in cases when there is a limitation at each time step.

Next, we addressed the problem of sampling from edge streams with nodal information. We proposed `SAMPLearn`, which leverages the nodal information when present with the structural information to sample communities. In addition to the community structure, `SAMPLearn` also outputs a graph representative of the original graph. The intuition behind `SAMPLearn` is that when attributes of nodes are present and established to characterize communities, combine the structural information with the attribute information to decide community memberships of nodes. However, when the attribute information is not present or established not characterize communities, it uses only structural information in deciding the community membership of nodes. When the sample size has been reached, `SAMPLearn` evict nodes that have lower importance to the communities they belong. Experimental results show that `SAMPLeran` can outperform baselines up to about 5 times.

From our works on sampling communities, we learned there exist different evolution patterns. We investigated the different patterns that existed in the evolution of a community's number of active users. We then identified factors that characterize the different parts of various evolution patterns. We postulated that a community's evolution is related to the behavior of members of the community.

As a first step in investigating how the behavior of members of a community relates to community's evolution, we examined a single Reddit community (`changemyview`). `changemyview` is a community that allows members (OP) post sumissions seeking opinion change. For our studies on `changemyview`, we categorize OPs into susceptible and non-susceptible. We then investigate what factors characterize these group of users. Susceptible OPs are those that changed their mind all the time on submissions made while non-susceptible OPs are those that never changed their mind on any of the submissions

made. We consider three main sources of information in characterizing users: (1) Language usage by OPs (2) Prior stance of OPs before seeking opinion change and (2) the interactions between OP and their challengers. Experimental results showed that OPs that never changed their mind were more analytical in thinking than OPs that changed their mind all the time. Also, susceptible OPs tend to interact more with their challengers more at the early and middle part of the conversation.

Finally, from the observations made from our studies on `changemyview`, we investigated how the different user behaviors (interaction style of members in a community and language usage of members) could characterize the different parts of the evolution of the number of active users in communities. We begin by first identifying the different patterns that exist in the evolution of communities. We found that three kinds of patterns exist regarding the evolution of communities on Reddit: (1) Communities that start to increase in the number of active users from some time forward in their evolution (growing communities) (2) Communities that start to decrease in the number of active users from some time forward (failing communities) and (3) Communities that switches between increasing and decreasing in the number of active users over time (unstable communities). Experiments showed that the middle part of conversations are related to the number of active users over time. Conversations about leisure do not attract more people in comparison to those less of leisure. Also, the extreme demonstration of leadership does not attract users to participate in conversations.

BIBLIOGRAPHY

[1] The delta system. [Online]. Available: https://www.reddit.com/r/changemyview/wiki/deltasystem

[2] "Twitter developer documentation," https://dev.twitter.com/rest/reference/get/followers/ids, accessed: 08-16-2017.

[3] C. C. Aggarwal, "An introduction to social network data analytics," in *Social Network Data Analytics*, 2011, pp. 1–15.

[4] N. K. Ahmed, J. Neville, and R. Kompella, "Space-efficient sampling from social activity streams," in *Proceedings of the 1st International Workshop on Big Data, Streams and Heterogeneous Source Mining: Algorithms, Systems, Programming Models and Applications*, 2012, pp. 53–60. [Online]. Available: http://doi.acm.org/10.1145/2351316.2351324

[5] E. Akbas and P. Zhao, "Attributed graph clustering: An attribute-aware graph embedding approach," in *Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017*, ser. ASONAM '17. New York, NY, USA: ACM, 2017, pp. 305–308. [Online]. Available: http://doi.acm.org/10.1145/3110025.3110092

[6] H. Alvari, A. Hajibagheri, G. Sukthankar, and K. Lakkaraju, "Identifying community structures in dynamic networks," *Social Network Analysis and Mining*, vol. 6, no. 1, p. 77, 2016.

[7] P. Anand, J. King, J. Boyd-Graber, E. Wagner, C. Martell, D. Oard, and P. Resnik, "Believe me - we can do this! annotating persuasive acts in blog text," in *Computational Models of Natural Argument*, 2011.

[8] R. Andersen and K. J. Lang, "Communities from seed sets," in *Proceedings of the 15th International Conference on World Wide Web*, ser. WWW '06. New York, NY, USA: ACM, 2006, pp. 223–232. [Online]. Available: http://doi.acm.org/10.1145/1135777.1135814

[9] L. Backstrom, J. Kleinberg, L. Lee, and C. Danescu-Niculescu-Mizil, "Characterizing and curating conversation threads: Expansion, focus, volume, re-entry," in *Proceedings of WSDM*, 2013, pp. 13–22.

[10] P. Bedi and C. Sharma, "Community detection in social networks," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 6, no. 3, pp. 115–135, 2016.

[11] O. Benyahia, C. Largeron, B. Jeudy, and O. R. Zaïane, "Dancer: Dynamic attributed network with community structure generator," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2016, pp. 41–44.

[12] T. Y. Berger-Wolf and A. S. Maiya, "Sampling community structure," in *Proceedings of the 19th international conference on World Wide Web*, 2010, pp. 701–710.

[13] R. F. Betzel, M. A. Bertolero, E. M. Gordon, C. Gratton, N. U. F. Dosenbach, and D. S. Bassett, "The community structure of functional brain networks exhibits scale-specific patterns of inter- and intra-subject variability," *NeuroImage*, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1053811919305658

[14] S. Bhagat, G. Cormode, and S. Muthukrishnan, "Node classification in social networks," in *Social Network Data Analytics*, 2011, pp. 115–148.

[15] N. Blenn, C. Doerr, B. Van Kester, and P. Van Mieghem, "Crawling and detecting community structure in online social networks using local information," *NETWORKING 2012*, pp. 56–67, 2012.

[16] V. D. Blondel, J. Guillaume, R. Lambiotte, and L. Etienne, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, no. 10, p. P10008, oct 2008.

[17] S. P. Borgatti, "Identifying sets of key players in a social network," *Computational & Mathematical Organization Theory*, vol. 12, no. 1, pp. 21–34, 2006.

[18] J. Cacioppo and R. Petty, "The elaboration likelihood model of persuasion," *Communication and Persuasion*, pp. 1–24, 1986.

[19] A. Chakraborty, Y. Kichikawa, T. Iino, H. Iyetomi, H. Inoue, Y. Fujiwara, and H. Aoyama, "Hierarchical communities in walnut structure of japanese production network," 02 2018.

[20] F. Chen and K. Li, "Detecting hierarchical structure of community members in social networks," *Know.-Based Syst.*, vol. 87, no. C, pp. 3–15, Oct. 2015. [Online]. Available: http://dx.doi.org/10.1016/j.knosys.2015.05.026

[21] Y. Chen and X. Qiu, "Detecting community structures in social networks with particle swarm optimization," in *Frontiers in Internet Technologies*. Springer, 2013, pp. 266–275.

[22] J. Cheng, D. Romero, B. Meeder, and J. Kleinberg, "Predicting reciprocity in social networks," in *Third International Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third International Conference on Social Computing*. Boston, MA, USA: IEEE, 2011, pp. 49–56.

[23] X. Chu and H. Sethu, "On estimating the spectral radius of large graphs through subgraph sampling," in *2015 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2015, pp. 432–437.

[24] A. Clauset, M. E. J. Newman, and C. Moore, "Finding community structure in very large networks," *Physical review. E, Statistical, nonlinear, and soft matter physics*, vol. 70, p. 066111, 01 2005.

[25] T. Cunha, D. Jurgens, C. Tan, and D. Romero, "Are all successful communities alike? characterizing and predicting the success of online communities," in *The World Wide Web Conference*, ser. WWW '19, New York, NY, USA, 2019, pp. 318–328.

[26] C. Danescu-Niculescu-Mizil, R. West, D. Jurafsky, J. Leskovec, and C. Potts, "No country for old members: User lifecycle and linguistic change in online communities," in *Proceedings of the 22Nd International Conference on World Wide Web*, ser. WWW '13. New York, NY, USA: ACM, 2013, pp. 307–318.

[27] N. Eagle and A. Pentland, "Reality mining: sensing complex social systems," *Personal and ubiquitous computing*, vol. 10, no. 4, pp. 255–268, 2006.

[28] I. Falih, N. Grozavu, R. Kanawati, and Y. Bennani, "Community detection in attributed network," in *Companion Proceedings of the The Web Conference 2018*, ser. WWW '18. International World Wide Web Conferences Steering Committee, 2018, pp. 1299–1306. [Online]. Available: https://doi.org/10.1145/3184558.3191570

[29] L. Festinger, *A theory of cognitive dissonance*, 1957.

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

[31] W. Hamilton, J. Zhang, L. Lee, C. Danescu-Niculescu-Mizil, D. Jurafsky, and J. Leskovec, "Loyalty in online communities," in *Proceedings of ICWSM*, 2017.

[32] D. D. Heckathorn, "Respondent-driven sampling: A new approach to the study of hidden populations," *Social Problems*, vol. 44, no. 2, pp. 174–199, 1997. [Online]. Available: http://www.jstor.org/stable/3096941

[33] A. Hollocou, J. Maudet, T. Bonald, and M. Lelarge, "A linear streaming algorithm for community detection in very large networks," *arXiv preprint arXiv:1703.02955, 2017*, 2017.

[34] C. Hubler, H. Kriegel, K. Borgwardt, and Z. Ghahramani, "Metropolis algorithms for representative subgraph sampling," in *2008 Eighth IEEE International Conference on Data Mining*, 2008, pp. 283–292.

[35] K. Hyland, *Hedging in scientific research articles*. John Benjamins Publishing Company, 1998.

[36] J. Islam. Tension analysis. [Online]. Available: https://www.github.com/jumayel06/Tension-Analysis/tree/master/resources

[37] A. Jaech, V. Zayats, H. Fang, M. Ostendorf, , and H. Hajishirzi, "Talking to the crowd: What do people react to in online discussions?" in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 2026–203.

[38] C. Jia, Y. Li, M. B. Carson, X. Wang, and J. Yu, "Node attribute-enhanced community detection in complex networks," in *Scientific Reports*, 2017.

[39] L. Jin, Y. Chen, T. Wang, P. Hui, and A. Vasilakos, "Understanding user behavior in online social networks: a survey," *IEEE Communications Magazine*, vol. 51, no. 9, pp. 144–150, 2013.

[40] T. Khazaei, X. Lu, and R. Mercer, "Writing to persuade: analysis and detection of persuasive discourse," *iConference 2017 Proceedings*, pp. 203–215, 2017.

[41] A. J. Kim, *Community Building on the Web: Secret Strategies for Successful Online Communities*, 1st ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2000.

[42] I. M. Kloumann and J. M. Kleinberg, "Community membership identification from small seed sets," in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '14. New York, NY, USA: ACM, 2014, pp. 1366–1375.

[43] R. E. Kraut, P. Resnick, S. Kiesler, Y. Ren, Y. Chen, M. Burke, N. Kittur, J. Riedl, and J. Konstan, *Building Successful Online Communities: Evidence-Based Social Design*. The MIT Press, 2012.

[44] G. Lakoff, "Hedges: A study in meaning criteria and the logic of fuzzy concepts," *Journal of philosophical logic*, vol. 2, no. 4, pp. 458–508, 1973.

[45] J. Leskovec, J. Kleinberg, and C. Faloutsos, "Graphs over time: densification laws, shrinking diameters and possible explanations," in *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*. ACM, 2005, pp. 177–187.

[46] P. Li, L. Huang, C. Wang, D. Huang, and J. Lai, "Community detection using attribute homogenous motif," *IEEE Access*, vol. 6, pp. 47 707–47 716, 2018.

[47] P. Liakos, A. Ntoulas, and D. A., "Coeus: Community detection via seed-set expansion on graph streams," in *2017 IEEE International Conference on Big Data (Big Data)*, Dec 2017, pp. 676–685.

[48] Y. Lin, Y. Chi, S. Zhu, H. Sundaram, and B. L. Tseng, "Facetnet: a framework for analyzing communities and their evolutions in dynamic networks," in *Proceedings of the 17th international conference on World Wide Web*. ACM, 2008, pp. 685–694.

[49] G. Liu and S. K. Prasad, "Community structure and detection in complex networks : A survey," 2012.

[50] X. Lu, T. Q. Phan, and S. Bressan, "Incremental algorithms for sampling dynamic graphs," in *Database and Expert Systems Applications*, H. Decker, L. Lhotská, S. Link, J. Basl, and A. M. Tjoa, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 327–341.

[51] A. S. Maiya and T. Y. Berger-Wolf, "Online sampling of high centrality individuals in social networks," *n: Zaki M.J., Yu J.X., Ravindran B., Pudi V. (eds) Advances in Knowledge Discovery and Data Mining. PAKDD 2010. Lecture Notes in Computer Science, vol 6118. Springer, Berlin, Heidelberg*, vol. 6118, pp. 91–98, 2010.

[52] E. Massaro and F. Bagnoli, "Hierarchical community structure in complex (social) networks," *Acta Physica Polonica B Proceedings Supplement*, vol. 7, 02 2014.

[53] W. McGuire, "Inducing resistance to persuasion: Some contemporary approaches," *Advances in experimental social psychology*, pp. 191–229, 1964.

[54] C. P. H. Mulder, E. Bazeley-White, P. G. Dimitrakopoulos, A. Hector, M. Scherer-Lorenzen, and B. Schmid, "Species evenness and productivity in experimental plant communities," *Oikos*, vol. 107, no. 1, pp. 50–63, 2004.

[55] P. K. Murphy, "What makes a text persuasive? comparing studentsâĂŹ and expertsâĂŹ conceptions of persuasiveness," *International Journal of Educational Research*, vol. 35, no. 7, pp. 675 – 698, 2001.

[56] M. E. J. Newman, "Modularity and community structure in networks," *Proceedings of the National Academy of Sciences*, vol. 103, no. 23, pp. 8577–8582, 2006. [Online]. Available: https://www.pnas.org/content/103/23/8577

[57] N. P. Nguyen, T. N. Dinh, Y. Xuan, and M. T. Thai, "Adaptive algorithms for detecting community structure in dynamic social networks," in *Proceedings of the 2011 IEEE international conference on Computer Communications*. IEEE, 2011, pp. 2282–2290.

[58] B. O'Dea, M. Larsen, P. Batterham, A. L. Calear, and H. Christensen, "A linguistic analysis of suicide-related twitter posts," *The Journal of Crisis Intervention and Suicide Prevention*, vol. 38, no. 5, pp. 319–329, 2017.

[59] D. O'Keefe, *Persuasion: Theory and research*, 1990, vol. 2.

[60] M. Papagelis, G. Das, and N. Koudas, "Sampling online social networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 3, pp. 662–676, 2013.

[61] M. A. Porter, P. J. Mucha, M. Newman, and A. J. Friend, "Community structure in the united states house of representatives," *Physica A: Statistical Mechanics and its Applications*, vol. 386, no. 1, pp. 414 – 438, 2007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0378437107007844

[62] S. Prabhumoye, S. Choudhary, E. Spiliopoulou, C. Bogart, C. Rose, and A. Black, "Linguistic markers of influence in informal interactions," in *Proceedings of the Second Workshop on NLP and Computational Social Science*, 2017, pp. 53–62.

[63] D. Quercia, J. Ellis, L. Capra, and J. Crowcroft, "In the mood for being influential on twitter," in *IEEE Third International Conference on Privacy, Security, Risk and Trust and IEEE Third International Conference on Social Computing*, Oct 2011, pp. 307–314.

[64] A. Reihanian, M. Feizi-Derakhshi, and H. S. Aghdasi, "Community detection in social networks with node attributes based on multi-objective biogeography based optimization," *Engineering Applications of Artificial Intelligence*, vol. 62, pp. 51 – 67, 2017.

[65] M. Riondato and E. M. Kornaropoulos, "Fast approximation of betweenness centrality through sampling," *Data Mining and Knowledge Discovery*, vol. 30, no. 2, pp. 438–475, 2016. [Online]. Available: https://doi.org/10.1007/s10618-015-0423-0

[66] F. Riquelme and P. González-Cantergiani, "Measuring user influence on twitter," *Inf. Process. Manage.*, vol. 52, no. 5, pp. 949–975, Sep. 2016.

[67] G. Rossetti and R. Cazabet, "Community discovery in dynamic networks: A survey," *ACM Comput. Surv.*, vol. 51, no. 2, pp. 35:1–35:37, Feb. 2018. [Online]. Available: http://doi.acm.org/10.1145/3172867

[68] R. A. Rossi and N. K. Ahmed, "The network data repository with interactive graph analytics and visualization," in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015. [Online]. Available: http://networkrepository.com

[69] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 26, no. 1, pp. 43–49, February 1978.

[70] M. Salehi, H. R. Rabiee, and A. Rajabi, "Sampling from complex networks with high community structures," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 22, no. 2, p. 023126, 2012.

[71] P. Sen, G. M. Namata, M. Bilgic, L. Getoor, B. Gallagher, and T. Eliassi-Rad, "Collective classification in network data," *AI Magazine*, vol. 29, no. 3, pp. 93–106, 2008.

[72] J. Shang, C. Wang, G. Guo, and J. Qian, “An attribute-based community search method with graph refining,” *The Journal of Supercomputing*, pp. 1–28, 2017.

[73] S. Sikdar, T. Chakraborty, S. Sarkar, N. Ganguly, and A. Mukherjee, “Compas: Community preserving sampling for streaming graphs,” *CoRR*, vol. abs/1802.01614, 2018.

[74] M. Spiliopoulou, “Evolution in social networks: A survey,” in *Social Network Data Analytics*, 2011, pp. 149–175.

[75] M. Steurer and C. Trattner, “Who will interact with whom? a case-study in second life using online social network and location-based social network features to predict interactions between users,” in *Ubiquitous Social Media Analysis*, 2013, pp. 108–127.

[76] J. Sun and J. Tang, “A survey of models and algorithms for social influence analysis,” in *Social Network Data Analytics*, 2011, pp. 177–214.

[77] J. Sun, C. Faloutsos, S. Papadimitriou, and P. S. Yu, “Graphscope: parameter-free mining of large time-evolving graphs,” in *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2007, pp. 687–696.

[78] C. Tan, “Tracing community genealogy: How new communities emerge from the old,” *ICWSM. AAAI*, p. 395âĂŞ404, 2018.

[79] C. Tan, V. Niculae, C. Danescu-Niculescu-Mizil, and L. Lee, “Winning arguments: interaction dynamics and persuasion strategies in good-faith online discussions,” in *Proceedings of the 25th International Conference on World Wide Web*, 2016, pp. 613–624.

[80] E. Tan, L. Guo, S. Chen, X. Zhang, and Y. Zhao, "Spammer behavior analysis and detection in user generated content on social networks," in *2012 IEEE 32nd International Conference on Distributed Computing Systems*, June 2012, pp. 305–314.

[81] F. Tang and W. Ding, "Community detection with structural and attribute similarities," *Journal of Statistical Computation and Simulation*, vol. 89, no. 4, pp. 668–685, 2019.

[82] Y. Tausczik and J. Pennebaker, "The psychological meaning of words: Liwc and computerized text analysis methods," *Journal of Language and Social Psychology*, vol. 29, no. 1, pp. 24–54, 2010.

[83] G. S. Thakur, R. Tiwari, M. Thai, S.-S. Chen, and A. Dress, "Detection of local community structures in complex dynamic networks with random walks," *IET systems biology*, vol. 3, no. 4, pp. 266–278, 2009.

[84] J. R. Tyler, D. M. Wilkinson, and B. A. Huberman, "E-mail as spectroscopy: Automated discovery of community structure within organizations," *The Information Society*, vol. 21, no. 2, pp. 143–153, 2005. [Online]. Available: https://doi.org/10.1080/01972240590925348

[85] R. Varshavsky, A. Gottlieb, D. Horn, and M. Linial, "Novel Unsupervised Feature Filtering of Biological Data," *Bioinformatics*, vol. 22, no. 14, pp. e507–e513, 2006.

[86] D. Vidaurre, S. M. Smith, and M. W. Woolrich, "Brain network dynamics are hierarchically organized in time," *Proceedings of the National Academy of Sciences*, vol. 114, no. 48, pp. 12 827–12 832, 2017. [Online]. Available: https://www.pnas.org/content/114/48/12827

[87] S. Vosoughi, D. Roy, and S. Aral, "The spread of true and false news online," *Science*, vol. 359, no. 6380, pp. 1146–1151, 2018. [Online]. Available: https://science.sciencemag.org/content/359/6380/1146

[88] C. Wagner, S. Mitter, C. Körner, and M. Strohmaier, "When social bots attack: modeling susceptibility of users in online social networks," in *In Proceedings of the 2nd Workshop on Making Sense of Microposts held in conjunction with the 21st World Wide Web Conference 2012*, 2012, pp. 41–48.

[89] K. Wakita and T. Tsurumi, "Finding community structure in mega-scale social networks: [extended abstract]," in *Proceedings of the 16th International Conference on World Wide Web*, ser. WWW '07. New York, NY, USA: ACM, 2007, pp. 1275–1276. [Online]. Available: http://doi.acm.org/10.1145/1242572.1242805

[90] R. Wald, T. Khoshgoftaar, A. Napolitano, and C. Sumner, "Predicting susceptibility to social bots on twitter," in *IEEE 14th International Conference on Information Reuse Integration (IRI)*, 2013, pp. 6–13.

[91] F. Walter, "Narration as a human communication paradigm: The case of public moral argument," *Communication Monographs*, vol. 51, no. 1, pp. 1–22, 1984.

[92] C. Wang, J. Lai, and P. S. Yu, "Neiwalk: Community discovery in dynamic content-based networks," *IEEE transactions on knowledge and data engineering*, vol. 26, no. 7, pp. 1734–1748, 2014.

[93] J. H. Ward, "Hierarchical grouping to optimize an objective function," *Journal of the American Statistical Association*, vol. 58, no. 301, pp. 236–244, 1963.

[94] S. Wasserman and K. Faust, *Social Network Analysis: Methods and Applications*, ser. Structural Analysis in the Social Sciences. Cambridge University Press, 1994. [Online]. Available: https://books.google.com/books?id=CAm2DpIqRUIC

[95] E. J. Williams, A. Beardmore, and A. N. Joinson, "Individual differences in susceptibility to online influence: a theoretical review," *Computers in Human Behavior*, vol. 72, pp. 412 – 421, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0747563217301504

[96] L. Xiao, "A message's persuasive features in wikipedia's article for deletion discussions," in *Proceedings of the 9th International Conference on Social Media and Society*. New York, NY, USA: ACM, 2018, pp. 345–349.

[97] J. Xie, M. Chen, and B. K. Szymanski, "Labelrankt: Incremental community detection in dynamic networks via label propagation," in *Proceedings of the Workshop on Dynamic Networks Management and Mining*, ser. DyNetMM '13. New York, NY, USA: ACM, 2013, pp. 25–32.

[98] J. Yang, J. J. McAuley, and J. Leskovec, "Community detection in networks with node attributes," *2013 IEEE 13th International Conference on Data Mining*, pp. 1151–1156, 2013.

[99] J. Yang, J. McAuley, and J. Leskovec, "Community detection in networks with node attributes," in *Data Mining (ICDM), 2013 IEEE 13th international conference on*. IEEE, 2013, pp. 1151–1156.

[100] M. J. Zaki, "A survey of link prediction in social networks," in *Social Network Data Analytics*, 2011, pp. 243–275.

[101] A. Zakrzewska and D. A. Bader, "Streaming graph sampling with size restrictions," in *Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017*, ser. ASONAM '17, 2017, pp. 282–290. [Online]. Available: http://doi.acm.org/10.1145/3110025.3110058

[102] J. Zhang, R. Kumar, S. Ravi, and C. Danescu-Niculescu-Mizil, "Conversational flow in oxford-style debates," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016, pp. 136–141. [Online]. Available: http://aclweb.org/anthology/N16-1017

[103] C. Zhe, A. Sun, and X. Xiao, "Community detection on large complex attribute network," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, ser. KDD '19. New York, NY, USA: ACM, 2019, pp. 2041–2049.

[104] R. Zheng, J. Li, H. Chen, and Z. Huang, "A framework for authorship identification of online messages: writing-style features and classification techniques," *Journal of the American Society for Information Science and Technology*, vol. 57, no. 3, pp. 378–393, 2006.

[105] W. Zhongyu, L. Yang, and L. Yi, "Is this post persuasive? ranking argumentative comments in online forum," in *Association of Computational Linguistics*, 2016.

VITA

Humphrey Appiah Mensah was born in Accra, Ghana, West-Africa. He received his Bachelor of Science degree in Computer Science and Engineering with honors at University of Mines and Technology (Tarkwa, Ghana, West-Africa). He received his Master of Science degree in Computer Science and PhD in Computer and Information Science and Engineering from Syracuse University (Syracuse, New York, USA).