Community Identification, Evolution and Prediction in Dynamic Social Networks

by

Mansoureh Takaffoli

A thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Department of Computing Science

University of Alberta

Abstract

Information networks that describe the relationship between individuals are called social networks and are usually modeled by a graph structure. Social network analysis is the study of these information networks which leads to uncovering patterns of interaction among the entities. Community mining provides a higher level of structure and offers greater understanding, but networks change over time. Their constituent communities change, and the elements of those communities change over time as well, i.e. they have fluctuating members and can grow and shrink over time. Examining how the structure of these networks and their communities changes over time provides insights into their evolution patterns, factors that trigger the changes, and ultimately predict the future structure of these networks. Furthermore, this prediction has many important applications, such as recommendation systems and customer targeting.

In this PhD research dissertation, we provide a brief overview of the existing research in the area of dynamic social network analysis, and their limitations. Then, we present a framework, called MODEC, for modelling, detecting, and predicting the evolution of communities and individuals over time in a dynamic scenario. We introduce a new *incremental community mining* approach, in which communities in the current time are obtained based on the communities from the past time frame. Then, with the definition of the critical events and transitions, and applying our event analysis, the evolutions of communities are abstracted in order to see structure in the dynamic change over time. This higher level of analysis has a counterpart that deals with the fine grain changes in community members with relation to their communities or the global network. A community matching algorithm is also proposed to efficiently identify and track similar communities over time. We

also define the concept of meta community which is a series of similar communities captured in different timeframes and detected by our matching algorithm. Furthermore, the events detected by the framework are supplemented by the extraction and investigation of the topics discovered for each community, and extensive experimental studies on real datasets, demonstrate the applicability, effectiveness, and soundness of our proposed framework.

After analyzing the dynamic of social network, we predict the occurrence of different events and transition for communities. Our framework incorporates key features related to a community – its structure, history, and influential members, and automatically detects the most predictive features for each event and transition. Our experiments on real world datasets confirm that the evolution of communities can be predicted with a very high accuracy, while we further observe that the most significant features vary for the predictability of each event and transition.

Preface

Chapter 5 of this thesis has been published as Mansoureh Takaffoli, Farzad Sangi, Justin Fagnan, and Osmar R. Zaïane, Community evolution mining in dynamic social networks, Journal of Procedia - Social and Behavioral Sciences, 2011. I was responsible for the data collection and analysis as well as the manuscript composition. Osmar R. Zaïane was the supervisory author and was involved with concept formation and manuscript composition. Chapter 6 of this thesis has been published as Afra Abnar, Mansoureh Takaffoli, Reihaneh Rabbany, and Osmar R. Zaïane, SSRM: structural social role mining for dynamic social networks, Journal of Social Network Analysis and Mining, 5(1), 2015. I contributed to manuscript edits. Osmar R. Zaïane was the supervisory author and was involved with concept formation and manuscript composition.

To my beloved husband

And my supportive parents and siblings

For their endless love and encouragement.

Acknowledgements

I owe my deepest gratitude to my supervisor, Prof. Osmar R. Zaïane for his continuous guidance and support, encouragement and advice throughout my journey. I have been extremely lucky to have a supervisor who cared so much about my work, and who responded to my questions and queries so promptly. Thank you Osmar for your invaluable insights, constructive feedbacks, motivation and encouragement on this thesis.

I am especially indebted to my committee members for taking the time to read my thesis, and for providing me with great comments and insightful advices.

Many friends have helped me stay sane through these difficult years. My special thanks goes to my amazing friend, Reihaneh, who has been there for me all the time. Thank you Reihaneh for the countless and much-needed hours on the telephone.

I must express my heart-felt gratitude to my parents and my siblings for their continued support and unconditional love. Their support and care helped me overcome setbacks and stay focused on my graduate study.

Most importantly, none of this would have been possible without the love and patience of my husband. Ahmad has been a constant source of love, concern, support and strength all these years. Thank you Ahmad for everything throughout this endeavour.

Table of Contents

List of Tables

List of Figures

List of Symbols and Acronyms

Symbol	Description
$\mathcal{G}$	dynamic network
i	snapshot
G_i	network at snapshot i
C_i	set of communities discovered at snapshot i
C_i^p	community p at snapshot i
k	similarity threshold
sim	similarity function between two communities
match	optimal match function for a community
M	meta community
birth	snapshot at which the first instance of a meta community is seen
death	snapshot at which the last instance of a meta community is seen
form	form event for a community
$\mathrm{dissolve}$	dissolve event for a community
$\mathrm{survive}$	survive event for a community
split	split event for a community
merge	merge event for set of communities
$\mathrm{cohesion}$	cohesion score of a community
$\mathrm{density}$	density score of a community
$\mathrm{clusterCoeff}$	clustering coefficient score of a community
leader	leaders of a community
$\mathrm{ovScore}$	overlapping score between two communities

$size$	affiliation size of a meta community
$cohesion$	cohesion scores of a meta community
$density$	density scores of a meta community
$clusterCoeff$	clustering coefficient scores of a meta community
$\mathrm{fluctuation}$	average members fluctuation of a meta community
size	size transition between two communities
$\mathrm{cohesion}$	cohesion transition between two communities
$\mathrm{leaderShift}$	leader transition between two communities
unity	unity transition between two communities
appear	appear event for an individual
$\mathrm{disappear}$	disappear event for an individual
join	join event for an individual and a community
leave	leave event for an individual and a community
A	affiliation set of a node at a snapshot
$\mathcal{A}$	affiliations set of a node over time
$\mathrm{centrality}$	centrality score of a node at a snapshot
$\mathrm{joinInfluence}$	join influence metric of a node at a snapshot
$\mathrm{leaveInfluence}$	leave influence metric of a node at a snapshot
$\mathrm{involvement}$	involvement metric of a node at a snapshot
$involvement$	involvement metric of a node over time
$joinInfluence$	join influence of a node over time
$leaveInfluence$	leave influence of a node over time
$\mathrm{stability}$	stability metric of a node over time

List of Publications

[1] Afra Abnar, Mansoureh Takaffoli, Reihaneh Rabbany, and Osmar R. Zaïane. Ssrm: Structural social role mining for dynamic social networks. In *International Conference on Advances in Social Networks Analysis and Mining*, ASONAM '14, 2014.

[2] Afra Abnar, Mansoureh Takaffoli, Reihaneh Rabbany, and Osmar R. Zaïane. Ssrm: structural social role mining for dynamic social networks. *Social Network Analysis and Mining*, 5(1), 2015.

[3] Jiyang Chen, Justin Fagnan, Randy Goebel, Reihaneh Rabbany, Farzad Sangi, Mansoureh Takaffoli, Eric Verbeek, and Osmar R. Zaïane. Meerkat: Community mining with dynamic social networks. In *Proceedings of 10th IEEE International Conference on Data Mining*, ICDM '10, 2010.

[4] Justin Fagnan, Reihaneh Rabbany, Mansoureh Takaffoli, Eric Verbeek, and Osmar R. Zaïane. Community dynamics: Event and role analysis in social network analysis. In *Proceedings of 10th International Conference on Advanced Data Mining and Applications*, ADMA '14, 2014.

[5] Mansoureh Takaffoli. Community evolution in dynamic social networks - challenges and problems. In *Proceedings of the IEEE ICDM PhD Student Forum*, 2011.

[6] Mansoureh Takaffoli, Reihaneh Rabbany, and Osmar R. Zaïane. Incremental local community identification in dynamic social networks. In *International Conference on Advances in Social Networks Analysis and Mining*, ASONAM '13, 2013.

[7] Mansoureh Takaffoli, Reihaneh Rabbany, and Osmar R. Zaïane. Community evolution prediction in dynamic social networks. In *International Conference on Advances in Social Networks Analysis and Mining*, ASONAM '14, 2014.

[8] Mansoureh Takaffoli, Farzad Sangi, Justin Fagnan, and Osmar R. Zaïane. A framework for analyzing dynamic social networks. In *Proceedings of the 7th Conference on Applications of Social Network Analysis*, ASNA '10, 2010.

[9] Mansoureh Takaffoli, Farzad Sangi, Justin Fagnan, and Osmar R. Zaïane. Community evolution mining in dynamic social networks. *Procedia - Social and Behavioral Sciences*, 22:49–58, 2011.

[10] Mansoureh Takaffoli, Farzad Sangi, Justin Fagnan, and Osmar R. Zaïane. Modec - modeling and detecting evolutions of communities. In *5th International AAAI Conference on Weblogs and Social Media*, ICWSM '11, 2011.

[11] Mansoureh Takaffoli, Farzad Sangi, Justin Fagnan, and Osmar R. Zaïane. Tracking changes in dynamic information networks. In *International Conference on Computational Aspects of Social Networks*, CASoN, 2011.

Chapter 1

Introduction

Social networks are information networks that describe how individuals or entities interact with one another. These networks can be modelled as a graph structure, where each individual is represented by a node in the network. There is an edge between two nodes if they are involved in an interaction or relationship during the observation time. For instance, the graph of co-authorship relationships between scholars, the interaction between posters on an on-line forum, the graph of web pages inter-connected through hyperlinks, email interactions between employees within an organization, and the exchange of ideas, information, and experiences between people in the web are all examples of social networks.

In these networks, understanding the underlying structure, determining the structural properties of the network, and uncovering the patterns of interaction among the entities has recently driven significant attention in sociology [116], epidemiology [67], and criminology [19].

One way to gain information about the network is the identification of communities, where a community is a set of densely connected individuals that are loosely connected to others [70]. Members of a community tend to mainly communicate with the other members of that community, and less with individuals in the rest of the network. The presence of communities in networks is a signature of the hierarchical nature of complex systems. From a visualization perspective, the community structure is extremely useful due to the fact that it presents a more compact and understandable description of the network as a whole. The analysis of communities not only helps to determine the structural properties of the network, but it also facili-

tates applications such as targeted marketing and advertising [51], recommendation systems [83], and email communication [111].

In many social networks such as co-authorship, friendship, blogosphere, and animal networks, the activities and interactions of the entities frequently change and vary in time [73]. Thus, the underlying structures of these networks are dynamic and evolve gradually. Furthermore, the communities in these dynamic networks usually have fluctuating members and could grow and shrink over the time [10, 62].

In these dynamic networks, researchers may be interested in the evolution of communities and membership of individuals such as author communities in the blogosphere [63], the analysis of mobile subscriber networks [118], and evolution of research communities [85]. Detecting the evolution of the communities can benefit companies to use this information to discover previously unknown relationships and interests among people. For example, companies can use the discovered information to create smarter advertisements and effective target marking strategy. The 2010 Edelman Trust Barometer Report [29] shows that 44% of the users respond to the online marketing, if there are users in their peer group who have responded to the advertisements. However, modelling the dynamic network as a static graph by discarding the temporal information associates with the interaction, and aggregating all the behaviours into a snapshot, misses the opportunity to detect the evolutionary patterns of the network and the communities.

A better modelling for such a temporal/dynamic social network is to convert an evolving network into static graphs at different snapshots [12]. In this model, each snapshot incorporates interactions that happened in its particular time-frame, the length of which can be determined based on how dynamic is the network. Modelling a dynamic network in this way, and assessing the evolution of communities, provides various insights into: 1) understanding the structures of the complex networks; 2) detecting a drastic change in the interaction patterns; 3) making predictions on the future trends of the network, which can help decision makers setup profitable marketing strategies in advance as in viral marketing [60], revenue maximization [5], and social influence [2].

At each snapshot, communities can be either explicit or implicit. Explicit com-

munities are built independently from their members and are based on a set of rules. In this case, people mostly join communities after the formation of the communities. Employees of a company or students participating in a course are examples of two explicit communities. On the other hand, the formation of implicit communities heavily depends on their members and connections. In the implicit communities, a community serves as the main engagement platform for the individual, thus, here, we mainly focus on implicit communities.

Most existing community mining methods assume that the community structure can be interpreted in terms of separated sets of communities. However, in some real networks, communities are not always disjoint from each other. In fact, in these networks communities usually overlap with each other since users can participate in more than one group at the same time. For instance, in a social network each person may belong to different communities based on his/her hobbies. As another example, a large fraction of proteins belong to several protein complexes at the same time [39]. Thus, at any given time, implicit communities can have two different settings: 1) an individual can belong to only one community (called non-overlapping communities or hard-partitioning); 2) an individual can belong to multiple communities (called overlapping communities or soft-partitioning).

Regardless of overlapping and non-overlapping setting, two main approaches have been followed to study the evolution of communities in a dynamic scenario. In the *independent community mining* approach, the communities at each snapshot are mined independently without considering the temporal information and their relationship to communities at the previous snapshots. Hence, this approach is suitable for social networks with unstable community structures. On the other hand, the *incremental community mining* approach uses the temporal information directly during the detection, where the community mining at a particular time is dependent on the communities detected in the previous timeframe. This approach finds a sequence of communities with temporal similarity, and hence, is only suitable for networks with community structures that are stable over time.

After extracting communities at each snapshot (independently or incrementally), in reality an individual can move from one community and join another one,

while the amount of interactions between members of a community also changes over time. Thus, a community experiences different changes during its life. In the literature there are different taxonomies to categorize the changes of communities over time. However, the most commonly used approach is to define different events including *split*, *survive*, *dissolve*, *merge*, and *form*. A community may *split* at a later snapshot if it fractures into multiple communities. It can *survive* if there exists a similar community in a future snapshot. In the case where there is no similar community at a later snapshot, then the community *dissolves*. A set of communities may also *merge* together at a later snapshot. Furthermore, at any snapshot there may be newly *formed* communities, where there is no similar community at a previous snapshot. The meta community can then be interpreted as a sequence of communities ordered by time, from the timeframe where it first appears to the timeframe where it is last observed. By performing event analysis, the evolution of communities are abstracted in order to see structure in the dynamic change over time.

Very little work has been done on why dynamic networks experience specific evolution transitions. Most of the previous research in this area focuses on either predicting the macroscopic graph structure, or the microscopic properties from the point of view of a single node or edge. However, predicting the trend of the mesoscopic structure, i.e. community, is related to important social phenomena such as homophily [66] and influence [6]. This analysis can also point out the social forces, and particular set of interests that lead to the formation of communities, and their future behaviour. For instance, analysis of the spread of a disease in a community assists discovering the early stages of an epidemic; the discussions in a co-authorship community can be used to follow the emergence and popularity of new ideas and technologies; the frequent loss of a community's members may cause the dissolution of the community. Thus, knowledge about the probable future of a community can help make better decisions concerning the members of a given community, and possibly protect community from termination.

THESIS STATEMENT. The process by which a dynamic information network develops can be analysed in terms of communities and individuals. Further-

more, the evolution of the network can be predicted using different features drawn from past events and transitions.

1.1 Thesis Contributions and Structure

In this thesis dissertation, we propose a dynamic network analysis framework, called MODEC[1], that analyzes a network from the perspective of its communities, and then predicts the future trends of the communities. Our framework does not have any assumption on the underlying community mining approaches; i.e. depending on the datasets and applications, the communities can be mined with either overlapping or non-overlapping algorithms. We also provide a technique to select independent or incremental methods based on the stability of the network structures. It is worth mentioning that we also propose an incremental community mining method which incorporates both current and temporal information. Our proposed incremental community mining is more appropriate for tracking more stable communities compared to independent method.

Regardless of the community mining approaches (i.e. independent/ incremental, and overlapping/ non-overlapping) used to detect communities, we employ a one-to-one matching algorithm to match the communities extracted in different snapshots. A meta community, which is a series of similar communities detected by the matching algorithm in different timeframes, is then constructed. We then identify a series of significant events and transitions which are used to explain how the communities of a meta community evolved over time. From the perspective of the nodes, we analyze the behaviour of the individuals by considering node-specific events and behavioural metrics. We also describe different roles that an individual can play in the whole network and in their communities and also how these roles change with respect to communities events.

After analyzing the dynamic of social network, we propose a machine learning model to accurately predict the next event and transition of a community, based on the relevant structural and temporal properties. Our framework incorporates key

[1] Modelling and Detecting the Evolutions of Communities

features related to a community – its structure, history, and influential members, and automatically detects the most predictive features for each event and transition.

One of the challenges in analyzing the dynamic social network and its communities is how to evaluate the detected evolution and and how to compare different frameworks with each other. For the datasets containing text, we investigate the topics related to communities and their semantic similarity, to validate the accuracy and outcome of our proposed events.

The main contributions of this thesis are summarized as follows.

- We adopt the static L-metric approach [23], to compute dynamic communities; where community mining at each snapshot starts by the communities found at the previous snapshot. The communities found at different snapshots are then matched based on their similarity, and grouped as the instances of the evolving communities over time. Furthermore, to capture the changes that are likely to occur for a dynamic community, we propose characterize and model the evolution trends of communities by defining different events and transitions (i.e. survive, dissolve, split, merge, form).

- We leverage *the relationship between the behaviour of individuals and the future of their communities*. Members of a community play an important role in attracting new members and generally shaping the future of their community. This fact is however overlooked by all previous works. Our models further assume that individuals who are *more likely to undertake actions* in their communities, are *more influential in the future trend of their community*, and therefore are principal factors in the predictive process. For instance, in marketing strategy, considering the impact of individuals on their communities is necessary for targeting the right consumers to direct advertisement, and to maximize the expectation of the total profit [42].

- Unlike previous works that only consider one aspect of the communities (i.e. size, age, or event), we provide a complete predictive process for any transition and event that a community may undergo, and at the same time, identify the most prominent features for each community transition and event. Fur-

thermore, our events and transitions do not have to take place in consecutive snapshots. A community may not necessarily be observed at consecutive snapshots, while it may be missing from one or more intermediate steps. Hence, our model *predicts the next stage of a community either in the exact next snapshot or any later snapshot*.

1.2 Organization of the thesis manuscript

The rest of the paper is organized as follows: In Chapter 2, we provide a brief overview of existing research in the area of community mining. We start the chapter with explaining different static community mining approaches. We present a brief survey on dynamic community mining techniques, and classify different techniques into the two classes of independent and incremental community mining.

The problem formulation and methodology is described in Chapter 3. In this chapter, we explain our proposed MODEC framework which consists of four stages, *Modelling and Discretization of Network*, *Community Structure Identification*, *Temporal Analysis of User Behaviour and Community Evolution*, and *Community Prediction using Supervised Learning* [99]. In Chapter 4, the second stage of MODEC framework, *Community Structure Identification*, is explained in more detail. Furthermore, in this chapter, we propose l-metric community mining algorithm to consider both current and temporal data in the process of mining communities. Our proposed algorithm is capable of detecting communities in the two variation of incremental/overlapping, and incremental/non-overlapping [100]. The *Temporal Analysis of User Behaviour and Community Evolution* stage of MODEC framework is presented in Chapters 5, and 6 [102, 104, 105, 103]. In Chapter 5, we characterize the evolution of communities by defining different events, and transitions. We then analyze the behaviour of individuals over time with the help of evens and role analysis in Chapter 6. In Chapter 7, the last stage of MODEC framework, *Community Prediction using Supervised Learning* is provided. We describe a technique to interpret the communities of a large networks, and predict how the community structure of the network changes in different circumstances [101].

The evaluation of the different stages of MODEC framework on real datasets is given in their related chapters. For our evaluation purpose, we consider two social network datasets: The Enron email dataset, which provides emails between employees of the Enron Corporation; and the DBLP co-authorship dataset, which contains a computer science co-authorship network. Finally, Chapter 8 concludes with a summary.

Chapter 2

Related Work

An important aspect in the complex and information networks is the identification of communities, which are defined as "densely connected" individuals that are loosely connected to others outside their group. The differences between many community mining methods is due to the different definitions of "densely connected" and the heuristic algorithms to identify such sets. Thus, a wide variety of community mining algorithms have been developed. A common approach to mine communities is normally to assume that the network of interest divides naturally into some subgroups, determined by the network itself. For instance, the Clique Percolation Method [86] finds groups of nodes that can be reached via chains of k-cliques. However, a good division of a network into communities is not merely one in which the number of edges running between groups is small. Rather, it is the one with the number of edges between groups is smaller than expected. A robust approach to tackle this problem is proposed in [72, 70] which is the maximization of a benefit function known as modularity Q over possible divisions of a network. The modularity considers the difference between the fraction of edges that are within the community and the expected such fraction if the edges are randomly distributed. Several community mining algorithms based on the modularity Q have been proposed such as fast modularity [71], and Max-Min modularity [24]. Furthermore, for the network that the global information is not available, local community mining algorithm based on a local version of this measure is developed. Local modularity M [65], and local modularity L [23] are all local variants of modularity Q, where the ratio of internal and external edges is calculated by identifying boundary

nodes of a detected local community.

Although many mining algorithms are based on the concept of modularity, Fortunato and Barthélemy [37] prove that modularity cannot accurately evaluate small communities due to its resolution limit. Hence, any algorithm based on modularity is biased against small communities. Another approach to mine communities is by utilizing the information theory concept such as compression (e.g. Infomod [90], Infomap [91]), and entropy (e.g. entropy-base [52]). Furthermore, Top leader [88] and WEBA [113] are based on the idea that a community is a set of followers congregating around a potential leader. For a complete survey on different community mining techniques and algorithms, the reader can refer to [38, 58, 81].

The above mentioned methods are useful if the community structure can be interpreted as a set of separated communities, whereas most of the actual networks are made of highly overlapping cohesive groups of nodes. Different community mining algorithms are proposed to uncover the overlapping community structure in a given network. For instance, COPRA (Community Overlap PRopagation Algorithm) is proposed to detect overlapping communities in networks by label propagation [46]. Here, vertices propagate their labels to their neighbours so that members of a community reach a consensus on their community membership. In order to support overlapping community structure, each vertex can now belong to up to c communities and propagate labels related to all its communities, where c is the parameter of the algorithm. Yang and Leskovec [121] propose BIGCLAM, a probabilistic generative model, to detect densely overlapping, hierarchically nested communities. They assign each node-community pair a non-negative latent factor which represents the degree of membership of a node to the community. Afterwards, they model the probability of an edge between a pair of nodes in the network as a function of the shared community affiliations. Their goal is reduced to estimating non-negative latent factors that model the membership strength of each node to each community. Nguyen et al. [75] propose an algorithm to find a community assignment that maximizes the overall internal density function. Here, unlike the case of non-overlapping community structure, in which connections between communities should be less than those inside them, their objective does not take into

account the number of edges between each community. For a complete survey on overlapping community mining algorithms, their evaluation, and benchmark please refer to [119].

Although most social networks evolve gradually, the static community mining techniques model the dynamic network as a static graph by removing information about the time of the interactions. Two main approaches have been followed to study the evolution of communities in a dynamic scenario: 1) independent community mining; 2) incremental community mining. In the independent community mining, the communities at each snapshot are mined independently without considering the temporal information and their relationship to communities at previous snapshots. After computing communities for each snapshot, the communities are tracked and matched based on their similarity. Communities at different snapshots that detected as matches, represent the instances of the same community at different time. Thus, the intuitive method is to compare two communities of consecutive time steps with rules based on the size of their intersection. These rules can be used conjointly with the community mining algorithm [85], applying clustering on a graph formed by all detected communities at different time snapshots [34], or heuristic algorithm to match communities based on their interaction [102], or even simplified by tracking specific core nodes that are more representative of their community than others [115].

Note that, although, most of the independent community mining consider matching communities between two consecutive snapshots, a community may not necessarily be observed at consecutive snapshots, i.e., it may be missing from one or more intermediate steps. To support these cases, this approach can be extended to consider matching communities at current snapshot to communities at all previous snapshots based on their intersections and time of occurrence [105].

In independent community mining, to capture the changes that are likely to occur for a community, researcher usually identify critical events that characterize the evolution of communities. There are different taxonomies to categorize these events, however, the commonly used five events are: survive, dissolve, split, merge, and form. A community *survives* if there exists a matching community in a future

snapshot. In the case where there is no matching community at a later snapshot, the community *dissolves*. A community may *split* at a later snapshot if it fractures into multiple communities. A set of communities may also *merge* together at a later snapshot. Finally, at any snapshot there may be newly *formed* communities which are defined as communities that have no matching community in any previous snapshots. There are two main issues with this approach. First, the static algorithms used on each snapshot are often non-deterministic and hence produce different communities even if the input graph does not change. This instability produces noise that makes the tracking very difficult. Furthermore, due to the fact that, the communities are mined independently at each snapshot without considering previous interactions, this approach is only suitable for the social networks with highly dynamic community structures.

The incremental community mining uses the temporal information directly during the detection, where the community mining at a particular time is influenced by the communities detected in previous time. This approach finds a sequence of communities with temporal similarity and hence, is only suitable for networks with community structures that are more stable over time. Furthermore, this approach cannot detect the evolution of communities in dynamic social network with explicitly defined communities. Generally, there are two techniques to mine communities incrementally, cost function method, and direct method.

In the incremental community mining, the communities mined during a particular snapshot should have low history cost, meaning it should be similar to the previously detected communities, and it should have high snapshot quality, meaning it should be a high-quality communities of the data that arrived during the current time. Thus, the cost function method is based on minimizing a cost function, which is first proposed by Chakrabarti et al. [20] to trade off between the history quality and the current snapshot quality. The cost function is usually composed of two sub-costs of a snapshot cost (SC) and a temporal cost (TC). Let G_i, and C_i be the graph and set of detected communities at snapshot i respectively. The general formulation of the cost function is as follows:

$$cost = \alpha SC(G_i, C_i) + (1 - \alpha)TC(C_{i-1}, C_i)$$

where the snapshot cost $SC()$ measures the quality of the detected communities, the temporal cost $TC()$ measures how similar the current communities are with the previous detected communities, and the parameter $\alpha(0 \leq \alpha \leq 1)$ is used to control the trade-off between current and temporal information. The incremental community mining is then tries to find an optimal community set that minimizes cost at each snapshot. To measure the quality of the current communities and calculate $SC()$, modularity Q or any other static validity criteria can be used (for a complete survey of validity criteria refer to [89]). On the other hand, the temporal cost $TC()$ must address the issue that some nodes appear for the first time in the current data, while some nodes will disappear. The differences between algorithms using this technique is due to their different calculation of SC and TC, and heuristic function to minimize the cost. One of the limitations of this approach is that the cost function only considers two consecutive snapshots. To the best of our knowledge, there is no generalization of the cost function to consider not only two consecutive snapshots, but all the previous snapshots to detect communities at each time.

Another technique to mine community incrementally is by considering the communities discovered at previous times in the process of detecting communities at the current snapshots directly. For example, in the incremental community mining algorithm based on the Dirichlet Process Mixture Model, the discovered communities at the previous snapshot is include in the base distribution of the Dirichlet Process [98]. Thus, the difference between cost function and direct method is that, the focus of the former is to optimize a new quality measure which incorporates deviation from history, while in the later the community structure is updated as new data arrives.

In the following, the current work on independent and incremental community mining is reviewed in detail, and if possible, their limitation is explained.

2.1 Independent Community Mining

In the independent community mining, after computing communities for each snapshot, the communities are tracked and matched based on their similarity. Communi-

ties at different snapshots that are detected as matches represent the instances of the same community which spans over time. Different rules are proposed to compare two communities of the consecutive time steps based on the size of their intersection.

The MONIC framework proposed in [95] assumes that the communities at each snapshot are first detected using any static community mining algorithm. Then, to track the evolution of communities at snapshot i, the framework executes a many-to-many matching function, that maps communities C_i to communities C_{i-1} based on their maximum overlaps and an overlap threshold. Then, based on the matching, four events can be induced. If a single community from $i-1$ matches to only a single community in i, these two communities are considered as survival. If multiple communities from $i-1$ matches to a single community in i, merge has occurred. For a community C_{i-1}^p that remains unmatched, two events may occur: C_{i-1}^p has split if its contents are in more than one communities at i, and all those communities together form a match for C_{i-1}^p, otherwise, C_{i-1}^p has dissolved. Oliveira and Gama [80] proposed MClusT framework to generalize MONIC. Instead of using the concept of communities overlap to match communities, MClusT uses a different metric based on conditional probability. Here, the weight assigned to the edge connecting communities C_{i-1}^p and C_i^q are estimated in accordance with the conditional probability:

$$weight(C_{i-1}^p, C_i^q) = P(v \in C_i^q | v \in C_{i-1}^p) = \frac{\sum P(v \in C_i^q \cap C_{i-1}^p)}{\sum P(v \in C_{i-1}^p)}$$

Palla et al. [85] proposed a similar approach to [34]. First, communities at each snapshot are detected independently using the Clique Percolation Method [86]. Then, for each consecutive snapshot $i-1$ and i, they construct a joint graph consisting of the union of links from the corresponding two networks, and extract the community structure of this joint graph using Clique Percolation Method. If a community in the joint graph contains a single community from $i-1$ and a single community from i, then they are matched and the survival event can be set. If the joint community contains more than one community from either time steps, the communities are matched in descending order of their members overlap. Commu-

nities from $i-1$ left with no match at i are considered as dissolved and communities from i left with no match at $i-1$ are considered as formed. Furthermore, in a case when the joint community contains more than one community from $i-1$, merge is marked, and if the joint community contains more than one community from i split is set. However, since this approach dependent on the clique percolation community detection algorithm, Bóta [16] propose an extension which is capable of handling communities found by any non-monotonic community detection algorithm.

The CommTracker framework proposed in [115] relies on core nodes instead of the overlapping level of nodes between two communities to match communities in different snapshots. To find the core members of a community, CommTracker first initialize the centrality score of all the nodes in the community by zero. Weight is also assigned to nodes based on a measure such as degree, betweenness. The centrality of the node is then updated base on the weight difference between the node and its neighbours. If the weight of the node is higher than its neighbour, its centrality should be incremented by the weight difference while its neighbour centrality is reduced by that difference. In the case where the weight of the node is lower than its neighbour, its centrality should be reduced by the difference while its neighbour centrality is incremented by that difference. The core nodes of a community is defined as the ones with positive centrality. After detection of the core members in all communities, community C_{i-1}^p is matched to C_i^q if and only if (1) at least one core node of C_{i-1}^p appears in C_i^q (2) at least one core node of C_i^q appear in some ancestor community of C_{i-1}^p. The ancestors of community C_{i-1}^p are the communities at previous snapshots that assigned as the matches for Community C_{i-1}^p. This matching is many-to-many, thus, the events related to communities can be directly induced. The matching of exactly one community from $i-1$ to one community from i indicates a survival. When one community from $i-1$ matches to multiple communities in i, split is occurred. Finally, the matching of multiple communities from $i-1$ to a single community indicate a merge. Note that, CommTracker is independent of the static community mining, thus, any algorithm can be used to detect communities at each snapshot separately.

Asur et al. [8] proposed an event-based framework where, at each snapshot,

the communities are mined independently using the MCL algorithm [27]. Then, critical events are proposed to capture significant changes that occur in an evolving network. These events are primarily between two consecutive snapshots. Two communities at two consecutive snapshots are marked as survival, if their members are exactly the same. Two communities merge together, if there exists a community in the next snapshot that contains at least $k\%$ of the nodes belonging to these two communities. A community splits if $k\%$ of its nodes are present in two different communities in the next snapshot. A community is formed if no two nodes were grouped together at previous time. Finally, a community will dissolve if none of its members will be grouped together in the next time. Here, they only consider events between consecutive snapshots, and their definitions are so restricted that many communities will remain unmarked. They also proposed different behavioural measures to study the behaviour of the nodes in the network and their influence on others.

As previously explained, the main problem of the most independent community mining is that, the static algorithms used on each snapshot are often non-deterministic. To solve this problem, Seifi and Guillaume [94] proposed a deterministic Louvain method [15] to discover stable communities, called community cores. The identification of community cores is based on the idea that if several community detection algorithms, or multiple executions of a non-deterministic algorithm, agree on certain sets of nodes, then these sets of nodes are certainly more significant. Hence, after the Louvian method is applied N times on a graph, the nodes that are grouped for more than a certain threshold are considered as stable communities (i.e. community cores). After, detecting the community cores at each snapshot independently, the changes between times $i-1$ and i can be studied by the difference between community cores only.

An application of independent community mining in an evolving scenario is presented in [48]. Here, the evolution of communities between two consecutive snapshots of a climate networks is studied. The modularity based mining [72] is first applied to mine communities at each snapshot. Then, the bipartite graph between two consecutive snapshots is created, where the communities are related if they

share at least 80% of their members. They chose a threshold of 0.8 because this ensures that each community can only be matched with one community in each of its adjacent time windows (i.e. the survival events are only discovered). The matching communities are then used to identify interesting features of the climate networks that correspond to known climatological features and events.

However, all the above work only considers matching across two consecutive snapshots. To further support the case where a community may be missing from one or more intermediate steps, this approach can be extended to consider matching communities at current snapshot to communities at all previous snapshots based on their intersections and time of occurrence.

Falkowski et al. [34] first detect communities at each snapshot using edge betweenness community mining [40]. They then generate a weighted graph consisting of detected communities at all snapshots. There is an edge between two communities at different snapshots if the overlap of their members exceeds a given threshold. The edge betweenness community mining is then applied on this graph, where each connected subgraph retained at the end, is the set of matching communities representing instances of the same community over time. Here, to calculate the weight of the edge between two communities at different snapshots, only the intersection of the members is considered. However, it would be more reasonable to include the time difference of the two communities in calculating their edge weight: if a community is equally similar to two communities at two different snapshots, more weight should be given to the one in closer temporal proximity. Furthermore, note that this framework cannot detect merge and split events.

Berger-Wolf and Saia [12] propose a mathematical and computational framework that enables tracking the evolution of communities. They further formulate the detection of dynamic communities as a graph colouring problem, called community interpretation [109]. They assume that the communities at each snapshots are available, and all individuals are observed at all the snapshots (i.e. no individuals leaves and no new individuals join at any snapshot). To detect the matches for the communities at time i, they generate a weighted graph consisting of communities at that time and communities at all the previous snapshots. The weight of

the edge between two communities at different snapshots is based on their Jaccard similarity, however, in order to give more weight to similar communities in close temporal proximity, they scale the Jaccard similarity by the difference in time steps. They formalized the problem of dynamic community identification, proved that it is NP-complete and APX-hard, and proposed several practical heuristics that involve greedily matching communities at different snapshots. The same team in their later work [106] showed that, under the assumption of no missing data, one of the proposed algorithms presented in [109] is a small constant factor approximation. They designed an approximation algorithm for the general setting with possible missing data. While the theoretic analysis provides an upper bound on the worst case performance of the approximation algorithm, they showed that in practice the algorithm performs very well, producing a solution close to the optimum. They furthermore improve the solution in practice by applying a Dynamic Programming approach. However, using their techniques, the merge and split events remain undetected.

Greene et al. [45] propose to identify a set of dynamic communities $D = \{D_1, ..., D_m\}$, where each dynamic community contain instances of the same community over time. Thus, each dynamic community D_k can be represented by a timeline of its constitute communities, where its most recent observation called front F_k. Their strategy to detect these set of dynamic communities is as follows. At each snapshot i, the communities C_i are discovered using a chosen static community mining. Then a weighted bipartite graph is generated between communities C_i and the front of the dynamic communities with weight being the Jaccard similarity. A many-to-many threshold based mappings is then applied which selects any two pair whose similarity is greater than the threshold. For any matching pair (C_i^p, F_k), the community C_i^p is added to the dynamic community D_k which constitutes the front F_k. The front of the dynamic community D_k is then updated to C_i^p. The output of the many-to-many matching itself will reveal the events related to communities (similar to CommTracker explained above). The only difference is that, in the case where community remains unmatched and form event is marked, a new dynamic community should be created. However, in this framework if a set $\{F_1, .., F_m\prime\}$ merges to a community C_i^p, the communities $\{F_1, .., F_m\prime\}$ can belong

to different snapshots. The merge of a set of communities from different snapshots to a community in most scenario is nonsense.

2.2 Incremental Community Mining

As noted earlier, there are generally two methods to mine communities incrementally: 1) cost function method; 2) direct method. We first review the algorithms using cost functions, then the direct method ones.

2.2.1 Cost Function Method

Cost function methods, first introduced by Chakrabarti et al. [20], try to find communities in a particular snapshot that are meaningful communities of the interactions that exist in that snapshot, and at the same time, are similar to the communities detected at its previous snapshot. These methods consider the former as the snapshot quality and the latter as the history quality, and minimize a cost function which is defined as a trade-off between these two qualities.

Lin et al. [64] introduce FacetNet framework which extends the overlapping community mining proposed by Yu et al. [123] from static graphs to dynamic networks. At each snapshot i the community structure is expressed by the mixture model proposed in [123]. In order to use the community structure at snapshot $i-1$ to regularize the community structure at current time, they use the cost function introduced earlier. Here, the snapshot cost SC at time i is calculated as the KL-divergence between the discovered community structure and the graph observed at this snapshot. Similarly, the temporal cost is defined as the the KL-divergence between the communities discovered at time $i-1$ and i. The optimization problem is then to find the best community structure at snapshot i that minimize the total cost.

Tantipathananandh and Berger-Wolf [108] propose a cost function consists of three parts: 1) cost of a node change its community affiliation between two snapshots; 2) cost of two nodes belonging to the same community but do not interact 3) cost of two nodes belonging to different communities but do interact. They pro-

pose the network community interpretation framework to find set of communities at each snapshot that minimizes the above three costs and devise an approximation algorithm via SDP relaxation and a heuristic rounding scheme. However, network community interpretation framework has two limitations. First, the graph at different snapshots share the same vertex set (i.e. nodes are always stable during the observation time), and the three costs has to be defined for each scenario.

Chen et al. [25] propose a convex relaxation problem to find the overlapping community structure that maximizes a quality function associated with each snapshot subject to a temporal smoothness constraint. Their quality function and smoothness constraint is based on the matrix representation of the overlapping communities; a matrix $Y \in \mathbb{R}^{n \times n}$ where Y_{vu} equals the number of clusters that include both node v and u. Based on this matrix representation, at any snapshot i the optimization problem reduced to measures how well Y_i reflects the adjacency matrices, while minimizing the difference between the matrices Y_i and Y_{i-1}.

2.2.2 Direct Method

While the cost function methods focus to optimize a new quality measure which incorporates deviation from history, the *direct methods* mine communities at the current snapshot incrementally by considering the communities discovered at the previous time and updating the community structure as the new data arrives. For instance, Sarkar and Moore [92] develop the Latent space model with temporal change to find communities that are consistent with the network at the current time and with the communities detected at a previous time. Mucha et al. [69] generalize the Laplazian dynamics approach in order to extend modularity maximization to study community structure across multiple times in dynamic social network.

In order to detect communities at snapshot i using the previous interactions, Aggarwal and Yu [3] propose to generate the differential graph between the graph G_i and G_{i-1}. To generate the differential graph, the difference between the weight of the edges at snapshot i and $i-1$ is calculated. Then, the differential graph is built from the edges with the differential weight. The weight of this edges can be both positive and negative, hence, the differential graph is a singed weighted graph. The

communities at the differential graph is then mined using their proposed singed version of k-mean community mining. These discovered communities can be marked by three tags: 1) An Expanding Community; 2) A Contracting Community; 3) Neutral or stable Community. The Expanding community is a community that the sum of its differential edges results in a positive number. The contracting community is a community with a negative sum over its edges, and a neutral community is a community with an almost zero sum. Note that, the main limitation of this algorithm is that the number of communities should be known in advance.

To mine communities in a current snapshot which is consistent with both temporal and current interactions, Kim and Han [54] propose to recalculate the weight between any nodes in the current graph. Formally, the weight between nodes v and w in graph G_i, is recalculated as $w'_i(v,w) = \alpha w_i(v,w) + (1-\alpha)w_{i-1}(v,w)$, where $w_i(v,w)$ is the weight of interactions between node v and w, and α controls the level of preference to history or current. Density based clustering is then applied on the G'_i with the new weight to detect communities at time i. After communities at snapshot i are detected, a bipartite graph from these communities and those at time $i-1$ is built. Then a greedy one-to-one matching select the two communities with the maximum similarity. The communities at time i that match to one community at time $i-1$ is marked by survived. The communities at time i without any matches are forming community, and communities at time $i-1$ without matches are dissolved communities.

Louvain method [15] is a static community mining method which is composed of two phases, executed alternatively. Initially, each node is in its own community. During phase 1, nodes are considered one by one, and each one is placed in the neighbouring community if this replacement maximizes the modularity gain. This phase is repeated until no node is moved. Phase 2 consists in building the graph between communities obtained during Phase 1. There is a node in the new graph for each community. The weight of the edge between any two community C^p and C^q is calculated as the sum of the weights of all the edges with one end in C^p and the other in C^q. The algorithm starts Phase 1 again with the new graph, grouping communities together, and then Phase 2, and so on until the modularity does not

improve. Aynaud and Guillaume [9] extend the Louvain method to incrementally mine communities in a dynamic scenario. Their idea is to change the initialization of the algorithm where the computation at snapshot i starts by grouping nodes using the communities found at snapshot $i-1$. However, this amount of stabilization might not be efficient for a very dynamic scenario. Thus, during the initialization $x\%$ of the nodes are randomly chosen and placed alone in their own community instead of their previous community. The percentage x control the trade-off between current and temporal information.

As explained earlier, one way to model the dynamic network is by aggregating all the interactions into a single snapshot, and apply community mining on the aggregated graph. In this case, the discovered communities represents community structure for all the observation time. However, this technique misses the opportunity to detect the evolutionary patterns of the communities over time. Duan et al. [28] propose Stream-Group framework to overcome this limitation. If the current snapshot is not detected as a change point, then the arriving graph is integrated into the aggregate graph at the previous time. If the current graph is a change point, a new aggregate graph containing the arriving graph is started. Formally, the Stream-Group framework first detects communities of graph G_i using the fast modularity [71]. Then, the similarity between the discovered communities C_i and the communities I_s discovered from the aggregate graph A_s is calculated. If their similarity is greater than the specified threshold, the framework assumes that the snapshot i is not a change point. Then, the Steam-Group framework aggregates the G_i with the graph A_s, and then applies fast modularity on A_s to update I_s. The grouping I_s is then representing the community structure from snapshot s to snapshot i. In the case where the similarity between the C_i and the I_s is lower than the threshold, the current snapshot is considered as a change point. Thus, a new aggregate graph containing G_i is created: $s \leftarrow i$, $A_s \leftarrow G_i$, and $I_s \leftarrow C_i$.

Aynaud and Guillaume [110] propose a similar approach to detect the collection of snapshots in which a unique set of communities is relevant for all its snapshots. To find these collections of snapshots, first communities for each snapshot are discovered by Louvain community mining [15]. Second, an agglomerative hierarchi-

Table 2.1: Comparison between different frameworks involving event detection and evolution of communities

Algorithm	**Scope**	**Matching**	**Events Detection**
MONIC [95]	Consecutive	Many-to-many matching	Five events
CommTracker [115]	Consecutive	Many-to-many matching	Five events
Community Interpretation [107]	All previous snapshots	One-to-one matching	Form, survive, dissolve
Falkowski [34]	All snapshots	Edge betweenness on communities graph	Form, survive, dissolve
Palla [85]	Consecutive	Clique percolation method on joint graph	Five events
FacetNet [64]	Consecutive	One-to-one matching	Form, survive, dissolve
Desirable	All previous snapshots	One-to-one matching	Five events

cal time clustering algorithm is used: two snapshots should be place in the same collection if their community structure is similar. Then, for each collection, the graphs of its constitute snapshots is accumulated and generate the aggregate graph. The Louvain community mining algorithm is then used to detect the communities at the aggregate graph. Note that, the snapshots in a collection can be discontinuous rather than being consecutive.

In [74], AFOCS, a two-phase framework for detecting and tracing the evolution of overlapping communities is proposed. In the first phase, it identifies overlapping communities for the first snapshot based on the algorithm proposed in [75] which maximizes the internal density of communities. In the next phase, the algorithm adaptively update the community structures by maximizing the internal density as the network evolves.

In Table 2.1 we provide the detailed comparison between all the frameworks that are more similar our proposed MODEC framework. The scope column in this table determines the number of other snapshots that is used to detect the events involving communities at a given timeframe.

2.3 Behavioural and Role Analysis

Most of the work on dynamic network analysis, does not consider the reason why a community or an individual experiences a specific event or transition. The changes in the role of individuals in a community can have a high influence on the development of the community and can act as triggers to evoke community changes. For example, if the leader of a community leaves, it might cause the remaining community members to become less active or disperse to other communities. Thus, one direction in analyzing social networks is to describe different roles that an individual can play and study also how these roles affect events and transitions. Furthermore, studying the behavioural characteristics of the detected evolving communities, help to predict the future evolution based on the discovered features.

There is no consensus on the definition of role among sociologists. Biddle [14] integrates various theories on role and discusses about functional, structural, organizational, and cognitive role theories. Among different theories on role, the only one which enables modelling the concept mathematically is the structural role theory. Oeser et al. [79, 77, 78] develop a mathematical model for structural role theory by defining three component *task*, *position*, and *person* that define a role in connection with each other and also with other positions in a society.

The emergence of social networking tools enables access to more information in order to model and study social roles. Consequently, the study of roles is now becoming an interdisciplinary field of research, attracting researchers from different disciplines, specifically data mining and machine learning.

Forestier et al. [36] present a survey of the state-of-the-art techniques for role mining in social networks.They further categorize roles to explicit, and non-explicit. *Explicit roles* are defined a-priori, and are identified by calculating a specifically designed method or a predefined criteria. Whereas, *non-explicit roles* are identified in an unsupervised framework, which requires little information about the roles beforehand. Clustering algorithms are usually used to identify non-explicit roles based on structural or contextual information in a network. The two widely defined examples of explicit roles are *experts* and *influentials*.

Notably, Zhang et al. [124] identify the expert role on a Java technical forum. They propose three algorithms based on z-score, pagerank [82] and HITS [55] using both indegree and outdegree of nodes to identify experts. When they compare their results to the results on simulated networks, they observe that *the structure of the network has a significant impact on the ranking of experts*.

Identifying nodes as *influential* member has also attracted considerable attention from researchers, mostly due to the influential role's wide range of applications in viral marketing, and diffusion of information. Kim and Han [53] distinguish three types of influential roles: *sales person*, *opinion leaders*, and *connector*. They further develop a two-step methodology for identifying the first type based on structural properties of the network. Agarwal et al. [2] explicitly define influential as an individual who is prominent in diffusion of innovation. They identify influential bloggers in blogosphere by defining an *iIndex* for each blogger based on their influential blog posts. Influential bloggers are those who have at least one influential post. In addition, authors discuss that influentials are different from initiators of an idea or creators of a content. *Influentials are more important because of their position in the network that empowers them to diffuse the influence*.

In all the aforementioned works, the community structure of the network is not directly considered in identifying roles. Community-based roles, on the other hand, are less studied, while they are important in many contexts, including link-based classification and influence maximization, as shown in [93]. Scripps et al. [93] define four structural community-based roles (*ambassadors*, *big fish*, *loners*, and *bridges*), and identify them based on the degree of nodes, and their community affiliation. Ambassadors are defined as nodes with high degree and also high community metric, whereas a big fish is an individual who is only important within his/her community. Bridges are the individuals with high community score, but low degree, and loners are the ones with low degree and also low community metric.

The activity in the communities is mostly determined by core members (or even a single member) who have a high influence on the development of the community. Furthermore, advertisement may also influence the development of a community. Thus, Falkowski et al. [33, 32] claim that the triggers that can be the cause of the

communities evolution are community leadership change, and external influences (such as advertising and publicity). A leadership change can be observed if node properties such as degree, and the vertex betweenness centrality are changed. For instance, a decreasing betweenness may indicate that a core member becomes less active, which might result in a less active community that eventually dissolve later. The external influences are usually observable if the global properties of the graph such as the average shortest path, and the modularity Q of the graph is changed.

The behavioural characteristic of the individuals and their communities in a dynamic scenario is studied by few researchers. For instance, Palla et al. [84] apply their proposed framework [85] on two dynamic real networks (co-authorship and phone call networks) and then analyze the behaviour of the evolving communities. On both networks, they find significant difference between the behaviour of smaller and larger communities. Communities containing only a few members persist longer on average when the fluctuations of the members is small. On the other hand, large communities persist longer if they are capable of continually changing their membership. Furthermore, their results shows that if the relative commitment of a user is to individuals outside of its community is higher, then it is more likely that he/she will leave the community.

Asur et al. [8] study the behaviour of the nodes in the network and their influence on others by defining different measures. The stability index measures the tendency of a node to have interactions with the same nodes over a period of time. The sociability index calculates the number of different interactions that a node participates in. Finally, the influence index of a node is a measure of the influence this node has on others to participate in different events. Furthermore, they define the popularity index of a community at an interval as the number of nodes that are attracted to it during that interval.

2.4 Prediction

The works on predicting the evolution of dynamic networks can be classified into three categories: microscopic, macroscopic and mesoscopic approaches. The mi-

croscopic approaches focus on the evolution at the level of nodes and edges, such as the study of preferential attachment phenomenon in [11, 30], the modelling of the node arrival and edge creations in [61], and mining patterns of link formations and link predictions in [26]. Notably, Backstrom et al. [10] approximate the probability of an individual joining two explicitly defined communities based on defining critical factors and then analyze the evolution of these communities. Furthermore, Yang et al. [122] develop a prediction model to analyze the loss of a user in an online social network, by extracting a set of attributes and using a decision tree classifier.

On the other hand, the macroscopic perspectives study the evolution of the high level properties of networks, for instance, the study of the evolution of degree distribution, clustering coefficient, and degree correlation of online social networks in [4], or analyzing the patterns of growth and shrinking diameter based on various topological properties, such as the degree of distribution and small-world properties of large networks in [62]. Kumar et al. [57] provide the properties of two real-world networks and then analyze the evolution of structure in these networks. Huang and Lee [49], propose a model to select the most influential activity features, and then incorporate these features to predict the growth or shrinkage of the network. Based on their findings, on the Facebook data, the number of active members and the number of edges is the most informative factors to predict the network evolution. Whereas, on the Citeseer data, it is observed that the number of collaborations between members is the main indicator to explain the evolving patterns of this co-authorship network.

A less explored perspective is provided by the mesoscopic approaches, which predict the trend of networks based on an intermediate structure of the networks, i.e. community structure. The evolution of communities from the standpoint of growth is modelled in [10, 125, 68], where an individual in a community never leaves the community, i.e. a community in these studies always grows. For instance, Backstrom et al. [10] apply a decision-tree approach by incorporating a wide range of structural features to predict whether and entity will join a community. Given a community, they also predict its growth over a fixed time period. Patil et al. [87] build a classifier to predict if a community is going to grow or is likely to remain

stable over a period of time. However, they only consider explicit communities, for instance, conferences are considered as communities for the DBLP dataset. Kairam et al. [50] identify two types of growth for a community. Diffusion growth is when a community attracts new members through ties to existing members; whereas, in non-diffusion growth, individuals with no prior ties become members themselves. Their analysis is then focused on the differences in the processes which govern diffusion and non-diffusion growth. Their finding shows that if a community is highly clustered, it is more likely to experience diffusion growth. However, communities that grow more from diffusion tend to reach smaller final sizes. They also generated a set of models which use a community's structural features and past growth experience to predict its eventual size and lifespan.

The works mentioned above consider explicit communities, and can only be applied in the settings where users join multiple communities and probably never quit these communities. Thus, the size of a community will monotonically increase over time. However, in most networks, an individual may quit his/her current community and join another one. Hence, the communities in these dynamic networks usually have fluctuating members and could grow and shrink over time. In the case of implicit communities, Goldberg et al. [43, 44] develop a linear regression system to predict the lifespan of a community based on structural features extracted from the early stage of the community. They find that a community's properties such as size, intensity and stability are the most important features to predict its lifespan. The most relevant work to ours is of Bródka et al. [17, 41], where they develop different classifiers to predict the events that may occur for a community (similarly defined as continue, merge, split, and dissolve). Their model is trained mainly based on the history of events happened to the community in preceding snapshots. Therefore, events can only be predicted for communities that their past three instances are available, while these instances also have to be in consecutive snapshots. Another drawback of their approach is that they consider events to be mutually exclusive and only predict the dominating event.

2.5 Evaluation of dynamic analysis

One of the challenges in analyzing the dynamic social network and its communities is how to evaluate the detected evolution and how to compare different frameworks with each other. For the datasets containing text, one approach is to incorporate the semantic content in the analysis. The semantic analysis not only can validate the discovered dynamic communities, but also can examine the influence of the semantics of the interaction on future interactions and reasoning about evolution.

Ning et al. [76] validate the survival events by extracting the keywords with top relative frequency from each community. They show that the keywords of the survival communities are roughly stable over the observation time. Asur et al. [8] define the semantic similarity between any two keywords using semantic category hierarchies and information theoretic measures. Then, the most frequent keywords discussed in each community is detected. They suggest that the probability of a merge event depends on the semantic similarity between the frequent words of the two involved communities, and the probability of a split event is inversely proportional to the semantic similarity between the frequent words of the two communities. Finally, two communities survive if their semantic similarity is very high.

2.6 Time Segmentation

The duration of the snapshots, or the number of snapshots in a given dynamic networks has great influence on the observed structures, analysis results, and the conclusion made about the network. If the time segmentation is done at a fine resolution (i.e. snapshots with small duration), the dynamic network will have lots of temporal detail, thus, so many unimportant events will be detected. On the other hand, the coarse time resolution may omit the important temporal information from the dynamic network. Hence, there should be a trade-off to differentiate between the non-essential temporal information and noises, and the meaningful and informative ones. However, very little work has been done to find out the appropriate time resolution for dynamic networks.

Sulo et al. [96, 97] claim that the behaviour of the linear functions on a dynamic

graph computed at different resolution levels can be used to distinguish between noisy temporal information and informative ones. Their algorithm is based on the idea that a good resolution will generate a dynamic network that has stationarity behaviour. Given a fixed resolution, the dynamic network is divided into a sequence of graphs $\{G_1, G_2, ..., G_n\}$. Then, a linear function such as density, and average degree is calculated on each graph of the sequence to generate the time-series of the linear function over the observation time. Since variance of this time-series could be a good indicator of the stationarity of the network, the resolution that results in minimum variance will be the appropriate time window.

Caceres et al. [18] propose an algorithm to determine the appropriate time interval by finding a balance between minimizing the noise and loss of temporal information.

None of the previous work cover all of the changes a community may experience during the observation time of a dynamic social network. However, finding patterns of interaction and predicting the future structure of communities is attractive for many areas such as disease modelling [31], information transmission [51, 111], and business management [13], but is only possible by capturing all the transitions of the communities in the dynamic social network. Thus, we put forth the MODEC framework which integrates key ideas to first detect communities either independently or incrementally based on the underlying structure of the network. Our proposed framework discovers all the events related to the communities, and delivers a more generalized methodology for identifying roles. Using the notion of meta-community, we are able to track multiple events and transitions that a community undergoes in non-consecutive snapshots.

The detected events and roles are further used as a building block to predict the future structure of communities. In our model, however, the future of a community is predicted based on an extensive set of features on its current members, their roles and their relations, where we also leverage temporal information (up to one time-frame backward), if the previous instances of the community are available.

Chapter 3

Problem Formulation and Methodology

In this dissertation, we would like to investigate the evolution of dynamic networks, at the level of their community structure, by monitoring the transition and evolution of its enclosing communities over time. We encompass both a community matching algorithm and an event detection model that captures the critical events and transitions for communities. This includes tracking the formation, survival and dissolution of communities as well as identifying the meta communities, which are a series of similar communities at different snapshots. Furthermore, we also propose events, roles, and behaviour analysis related to the individuals in a network. We leverage the relationship between the evolution of communities, the movement of individuals between these communities and changes in the role of those individuals. We analyze how the role modification can act as triggers to evoke community changes and can affect the dynamics of communities.

In order to mine communities at different snapshots, the traditional approach to solve this problem is to extract communities at each snapshot independent of the communities at other snapshots or the historic data. We also propose an incremental L-metric community mining approach to consider both current and temporal data in the process of mining communities. Finally, we predict the occurrence of different events and transition for communities in dynamic social networks. Our approach incorporates key features related to the community – its structure, history, and influential members, and automatically detects the most predictive features for each

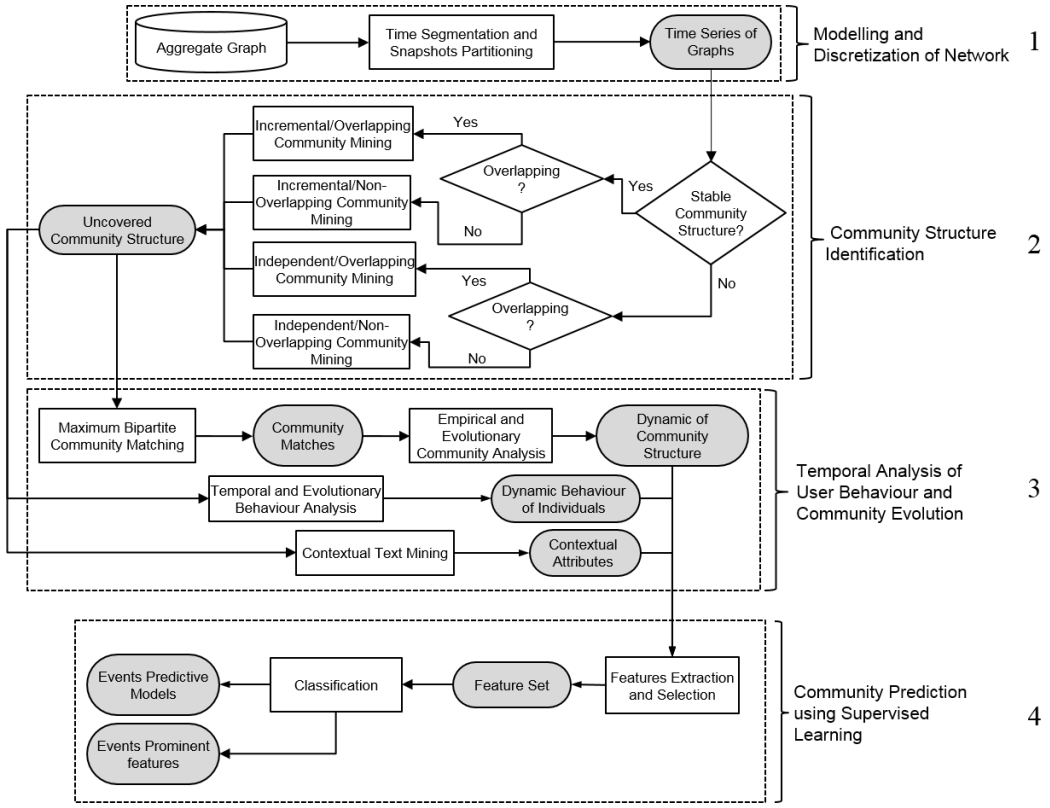


Figure 3.1: Different stages of MODEC framework to analyze dynamic social network.

event and transition.

In this dissertation, we mainly consider the structural properties of the network. Thus, our proposed approach works for social networks where there are no associated attributes with nodes and edges. However, in the presence of attributes, our approach can be utilized to either detect structural properties of the underlying network, or can be tuned to include the attributes of nodes and edges, if necessary.

In order to analyze dynamic social network and study the evolution of its communities and individuals, we propose MODEC framework which consists of four stages, *Modelling and Discretization of Network*, *Community Structure Identification*, *Temporal Analysis of User Behaviour and Community Evolution*, and *Community Prediction using Supervised Learning*. The four stages of MODEC are depicted in Figure 3.1. In the following, we will explain each stage in detail.

3.1 Modelling and Discretization of Network

A social network is modelled by a graph $G(V, E)$, where V is the set of vertices/nodes and E is the set of graph's edges. Here, the entities/individuals are associated with the nodes of the graph, whereas the connections/interactions between them are represented with the edges. Characteristics and attributes of entities and their interactions can be also included in this model, as different attributes on the nodes and edges, which depend on the context.

One can aggregate all the interactions of a dynamic network over time, into one snapshot, to model the network using a static social network. However, by discarding this temporal information, one is not able to detect invaluable evolutionary patterns that are happening inside the network. A better model for such a temporal/dynamic social network, would be to use a series of static network snapshots. Thus, in the first stage, to model the dynamic social network, the evolving network has to be converted into a series of static network snapshots, where each snapshot corresponds to a particular point in time. The duration of each snapshot or the number of snapshots in a dynamic social network is totally dependent on the application and can be determined based on how dynamic is the network. For example, to analyze the behaviour of a group of animals the duration of the snapshot can be half a week, while to study the evolution of communities in a co-authorship network yearly snapshots may be used. In this thesis dissertation, we assume the length of each snapshot for a particular dataset is given and determining the appropriate time resolution for dynamic social networks is out of the scope of this thesis. Furthermore, for the purpose of experiments we consider symmetric timeframes (i.e. having equal duration). However, our framework can also be applied in a case of non-symmetric snapshots.

In the rest of this thesis, we model the dynamic social network as a sequence of graphs $\mathcal{G} = \{G_0, G_2, ..., G_{n-1}\}$, where $G_i = (V_i, E_i)$ denotes a graph containing the set of individuals and their interactions at a particular snapshot i. Regardless of the underlying community mining algorithm, the set $C_i = \left\{C_i^0, C_i^1, ..., C_i^{n_i-1}\right\}$ denoted the n_i communities detected at the i^{th} snapshot, where community $C_i^p \in C_i$

Table 3.1: Modelling and Discretization of Network-Definition of symbols

Symbol	Definition
$\mathcal{G} = \{G_0, G_2, ..., G_{n-1}\}$	dynamic network as a sequence of graphs
$i \in [0, n-1]$	snapshot (i.e. particular point in time)
$G_i = (V_i, E_i)$	network at snapshot i, with V_i and E_i as its set of nodes and edges respectively
$C_i = \left\{C_i^0, C_i^2, ..., C_i^{n_i-1}\right\}$	set of communities discovered at snapshot i
$C_i^p = (V_i^p, E_i^p)$	community p at snapshot i, with V_i^p and E_i^p as its set of nodes and edges respectively

is also a graph represented by (V_i^p, E_i^p). Table 3.1 shows the symbols and their definitions.

3.2 Community Structure Identification

Grouping data points is one of the fundamental tasks in data mining, which is commonly known as clustering if data points are described by attributes. When dealing with interrelated data, where individuals are represented in the form of nodes and their relationships are considered for grouping rather than the node attributes, this task is also referred to as community mining. Formally, a community is roughly defined as densely connected individuals that are loosely connected to others outside their group.

After modelling the dynamic social network, the next stage is to find communities at each snapshot. As we explained in the related work, two main approaches have been followed to study the evolution of communities in a dynamic scenario: independent community mining, and incremental community mining. In the *independent community mining* approach, the communities at each snapshot are mined independently without considering the temporal information and their relationship to communities at the previous snapshots. Hence, this approach is suitable for social networks with unstable community structures. On the other hand, the *incremental community mining* approach uses the temporal information directly during the detection, where the community mining at a particular time is dependent on the communities detected in the previous timeframe. This approach finds a sequence of communities with temporal similarity and hence, is only suitable for networks with

community structures that are stable over time.

From another perspective, communities can have non-overlapping and overlapping settings as well. In a non-overlapping setting, an individual can belong to only one community, whereas in an overlapping setting, an individual can belong to multiple communities. The formal definition of non-overlapping and overlapping community structure is as follows:

Non-overlapping community structure: Given a set of snapshots 0, 2, ..., $n-1$, the community structure is called non-overlapping if

$$\forall \text{ snapshot } i \quad p \neq q \quad \forall C_i^p \in C_i, \forall C_i^q \in C_i \quad C_i^p \cap C_i^q = \emptyset$$

Overlapping community structure: Given a set of snapshots 0, 2, ..., $n-1$, the community structure is called overlapping if

$$\forall \text{ snapshot } i \quad p \neq q \quad \exists C_i^p \in C_i, \exists C_i^q \in C_i \quad C_i^p \cap C_i^q \neq \emptyset$$

Thus, in MODEC framework communities can have four different variations: 1) independent and non-overlapping; 2) independent and overlapping; 3) incremental and non-overlapping; 4) incremental and overlapping. We will focus more on how to select any of the variations later in this chapter.

Generally, the MODEC framework is independent on the community mining algorithms used for any variations. However, in this dissertation, for the case of independent community mining we recommend the computational efficiency static L-metric community mining algorithm proposed by Chen et al. [23]. The main assumption of the algorithm is that a community has fewer connections from its boundary nodes to the unknown portion of the graph, while having a greater number of connections within its local community. The reasons that we recommend this algorithm is that it does not require any arbitrary thresholds or other parameters, and is robust against outliers. Furthermore, unlike most of static community mining algorithms that implicitly assume global information is always available, it detects communities with only local information. This locality makes L-metrics particularly desirable in the case of large real world networks, where the whole

graph is usually unavailable. Lastly, the L-metric community mining can be easily tuned to detect overlapping or non-overlapping communities. Thus, we employ the L-metric community mining because none of the others in the literature satisfy all these requirements simultaneously.

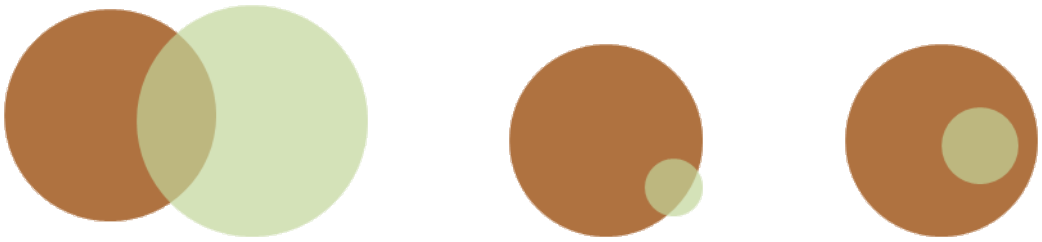
In the case of stable community structures where an incremental community mining is preferred, we propose incremental L-metric that extends the L-metric community mining to incrementally mine communities in a dynamic scenario. The main idea here is to change the initialization of the algorithm in a way that the computation at each snapshot starts by grouping nodes using the communities found at the previous snapshot. Similar to the static L-metric, our proposed method is local, parameter-free, and can detect overlapping and non-overlapping communities. In Chapter 4 the L-metric community mining algorithm is explained in more detail.

3.3 Temporal Analysis of User Behaviour and Community Evolution

When the communities are detected at each snapshot (independent/incrementally, and overlapping/non-overlapping), the next stage is to track the evolution of the discovered communities and individuals. Here, we distinguish between the terms community and meta community. A community contains individuals that are densely connected to each other at a particular time snapshot. A meta community is a series of similar communities at different time snapshots (not necessarily consecutive) and represents the evolution of its constituent communities ordered by time of the snapshots. Here, we reduce the problem of detecting the transition and evolution of communities to identify meta communities and also the events characterizing the changes of the communities across the time of observation.

3.3.1 Maximum Bipartite Community Matching

The key concept for the detection of the events, and also the meta community, is the concept of similarity between communities at different times. Two communities that are discovered at different snapshots are similar if a certain percentage, $k \in$



(a) (b) (c)

Figure 3.2: Examples to illustrate the similarity measure: (a) Two communities with 110 and 120 members where they have 30 mutual members; (b) Two communities with 100 and 30 members where they have 20 mutual members; (c) Two communities with 100 and 40 members where they have 40 mutual members.

$[0, 1]$, of their members are mutual. The similarity threshold k captures the tolerance to member fluctuation, and can be set based on the characteristic of the underlying network. A high similarity threshold would be expected in a network with stable communities that have many members who participate over a long time as well as having less fluctuating members. In highly dynamic social networks, where the structure changes over time, there are unstable communities such that the members of a community leave gradually while new ones join. This community may exist for a long time, even if all of its original individuals have left. Thus, to identify groups that make up this unstable community, a low similarity threshold would be preferred. The formal definition of similarity between two communities is defines as follows:

Community Similarity: Let C_i^p and C_j^q be the community detected at snapshot i and j respectively ($i \neq j$). The two communities C_j^q and C_i^p are similar if and only if their shared members make up at least k proportion of the biggest community:

$$\text{sim}(C_i^p, C_j^q) = \begin{cases} \frac{|V_i^p \cap V_j^q|}{\max(|V_i^p|, |V_j^q|)} & \text{if} \quad \frac{|V_i^p \cap V_j^q|}{\max(|V_i^p|, |V_j^q|)} \geq k \\ 0 & \text{otherwise} \end{cases} \tag{3.1}$$

Dividing the number of members that exist in both communities by the size of the biggest community (Equation 3.1) scales for different sizes of communities. Figure 3.2a illustrates an example when two communities are about the same size. These two communities shared 30 members, thus, they are similar if the similarity threshold k is less than 0.25 (i.e., $\frac{30}{120} \geq 0.25$). Figure 3.2b shows an example when

one of the communities is considerably smaller than the other and they have 20 mutual members. Hence, setting $k \leq 0.2$ marks them as two similar communities since $\frac{20}{100} \geq 0.2$. An example when one community contains all the members of the other community is shown in Figure 3.2c. The mutual members of these two communities are 40 individuals, thus, with $k \leq 0.4$ these communities are similar (i.e., $\frac{40}{100} \geq 0.4$). The choice of the similarity threshold k is dependent on the characteristic of the underlying network. In Chapter 6, we provide an algorithm to determine k for an arbitrary dynamic social network.

As noted before, the instances of the same community at different time-frames are considered as one *meta community*, where *birth* is the snapshot at which the first instance of meta community M is seen, and *death* represents the snapshot where the last instance of meta community M is observed. Furthermore, the *lifetime* of a meta community represents the number of snapshots between the *birth* and *death* of that meta community (i.e. the time difference between its first instantiation and its last instantiation). In the following, the formal definition of the meta community is provided, where $\text{match}(C_i^p, j)$ denotes the optimal match for C_i^p at j^{th} snapshot which is the result of the optimal matching algorithm for community C_i^p:

Meta Community: Given a set of snapshots 0, 2, ..., $n-1$, a meta community is a sequence of communities $M = \{C_b^{p_b}, ..., C_i^{p_i}, ..., C_d^{p_d}\}$ such that

(a) no two communities are in the same snapshot and communities are ordered by time
$0 \leq b < ... < i < ... < d \leq n-1$, where $b = \text{birth}(M)$, and $d = \text{death}(M)$

(b) $\forall C_i^p \in M \quad \exists C_j^q \in M \quad$ where $\quad \text{match}(C_i^p, j) = C_j^q \quad$ and $\quad j < i \quad$ and $\nexists C_k^r \in M \quad$ where $\quad \text{match}(C_i^p, k) = C_k^r \quad$ and $\quad k < i \quad$ and $\quad k \neq j$

(c) $\forall C_i^p \in M \quad \exists C_j^q \in M \quad$ where $\quad \text{match}(C_i^p, j) = C_j^q \quad$ and $\quad j > i \quad$ and $\nexists C_k^r \in M \quad$ where $\quad \text{match}(C_i^p, k) = C_k^r \quad$ and $\quad k > i \quad$ and $\quad k \neq j$

The summary of the notations and definitions used for community matching is provided in Table 3.2. An example of a co-authorship meta community detected from the DBLP dataset is shown in Figure 3.3. The first instance of this meta

Table 3.2: Maximum Bipartite Community Matching-Definition of symbols

Symbol	**Definition**
$k \in [0, 1]$	similarity threshold
$\text{sim}(C_i^p, C_j^q)$	similarity between communities C_i^p, and C_j^q at snapshot i, and j respectively (Equation 3.1)
$\text{match}(C_i^p, j)$	optimal match for community C_i^p at j^{th}
$M = \{C_b^{p_b}, ..., C_i^{p_i}, ..., C_d^{p_d}\}$	meta community
$b = \text{birth}(M)$	snapshot at which the first instance of meta community M is seen
$d = \text{death}(M)$	snapshot at which the last instance of meta community M is seen

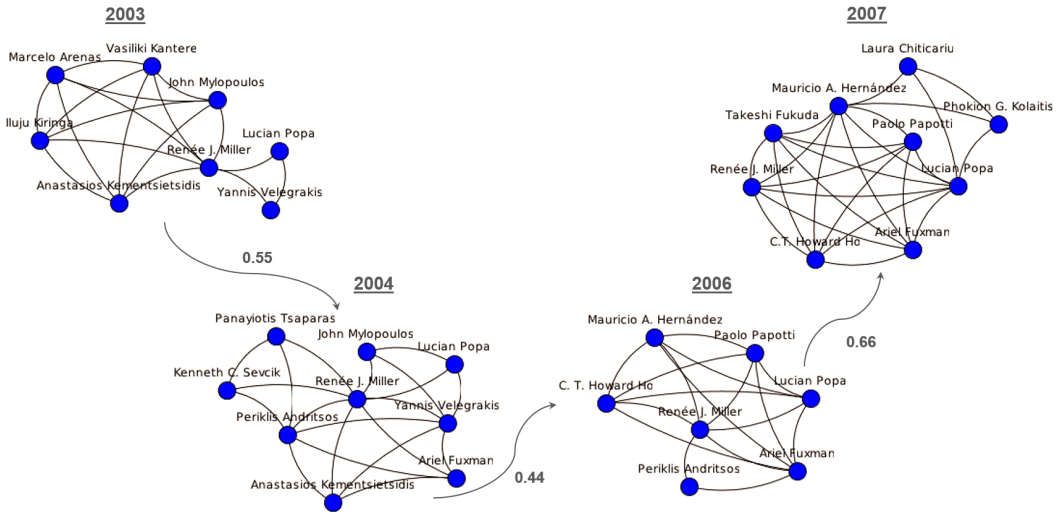


Figure 3.3: An example of a co-authorship meta community detected between the years 2003-2007 with $k = 0.4$.

community is detected at year 2003. Setting $k = 0.4$, this community survives into a community with 5 similar members in year 2004. The community was absent at year 2005. However, the community at year 2004 survives to a community at year 2006. The community in 2006 survives into a community at year 2007 that has 6 mutual members with it. In the following years there are not any other communities similar to at least one of the communities in this meta community, so the meta community dissolved in 2007.

In order to detect the meta communities and the events, the set of communities at a given snapshot have to be matched to the communities at previous snapshots based on their similarity. A simple approach would be to match communities from consecutive time steps in descending order of their similarity. However, since a

community may have similarity with several communities at the same time, the matching process becomes non-trivial, and this greedy matching algorithm cannot handle these cases. Furthermore, a community may not necessarily be observed at all the snapshots after its *formation* and may be missing from one or more snapshots. This reflects that, although a community was absent, after a few snapshots it may suddenly reappear in the network. To consider this scenario, a matching between communities at time i and all of the other communities at time $i' < i$ could be considered. Then, the optimization problem that arises here is to find a match that maximizes the pair wise similarity over all selected matches, not only the direct preceding snapshot but potentially other previous snapshots.

We propose a matching algorithm that maximizes the pair wise similarity over all selected matches in both the preceding snapshot and, potentially, in other previous snapshots. Initially, each community at snapshot 0 is considered as a newly *formed* community and a new meta community is created for each of them. In iteration i, we construct a weighted bipartite graph between communities at snapshot i and communities at $i-1$. The weight between communities is the same as their similarity introduced before. Then, the maximum weight bipartite matching [56] is applied to connect communities at snapshot i to communities at $i-1$. If community C_i^p matches to community C_{i-1}^q, C_i^p is the *survival* of C_{i-1}^q (i.e. C_{i-1}^q *survives* to C_i^p). Thus, the C_i^p is added to the meta community that constitutes community C_{i-1}^q. For the communities at snapshot i which are left with no counterpart from C_{i-1}, another bipartite matching is constructed between them and the communities at $i-2$ whose meta communities have not been selected yet. Again, the maximum weight bipartite matching is applied to detect *survival* events and also to update meta communities. The process of constructing the bipartite graph is continued until all the communities at snapshot i match a community at snapshot $0 \leq i' < i$ or all existing meta communities are already taken. The communities left with no matches from $\{C_0, ..., C_{i-1}\}$ are newly *formed* communities and a new meta community is built for each of them. After every community at i is assigned to one meta community, we move to the next iteration.

The meta communities detected by the above algorithm represent the evolution of its constituent communities ordered by time of the snapshots. The last community of every meta community is marked as *dissolve* since it is unmatched for all of the next snapshots.

3.3.2 Empirical and Evolutionary Community Analysis

In the literature, there are different taxonomies to categorize the changes of clusters, communities, or patterns that evolve over time [95, 80, 7, 85]. In order to capture the changes that are likely to occur for a community, we consider different events and transitions. A community may *split* at a later snapshot if it fractures into multiple communities. It can *survive* if there exists a similar community in a future snapshot. In the case where there is no similar community at a later snapshot, the community *dissolves*. A set of communities may also *merge* together at a later snapshot. Finally, at any snapshot there may be newly *formed* communities which are defined as communities that have no similar community in any previous snapshots. The meta community is then a sequence of *survival* communities ordered by time, from the timeframe where it first appears to the timeframe where it is last observed.

In the case of implicit communities, where the formation of communities heavily depends on their members and connections, an entity may *leave* its current community and *join* another community, due to the shifts of their interests or due to certain external events. Thus, when a community *survives* into next snapshot, it may also experience different transitions. The size transition occurs when the number of its members increases (i.e. *expand*), or decreases (i.e. *shrinks*). Moreover, members of a survived community may change their engagement level, making the community more *cohesive*, or *loose* (cohesion transition). Finally, when the most influential members of a community (i.e. leaders) shifts from set of node to the others, the community experiences *leader shift*. In the case of overlapping community structure, we also define *unity transition*: the unity between communities *disjoints* if their intersection connectivity becomes weaker, whereas, the intersection becomes *united* if the connectivity get stronger.

These proposed events and transitions track the changes of communities over

the entire observation time, rather than only between two consecutive snapshots. We furthermore define different metrics and their temporal variation as another way to characterize the evolution of communities. In Chapter 5, the formal definitions of the events, transitions, and metrics are provided in more detail.

3.3.3 Temporal and Evolutionary Behaviour Analysis

Walton [112] states that the changing nature of an individual and its leadership are of central importance to the explanation of community action. Furthermore, studying the behavioural characteristic of the individuals reveal interesting information on the underlying structure of the network. We categorize the information in a social network into *structural* and *non-structural* properties. Structural properties are related to the topology of the graph such as an entity's connections (edges), neighbourhood structure, and the entity's position in that structure. Whereas, non-structural properties are the information not reflected in the topology of the graph, such as an entity's attributes, a connection's attributes, and meta-data formation available about the graph. In this dissertation, we define different events, and metrics, and consider the role-taking behaviour of an individual which is aligned with their structural properties. We furthermore, study how these characteristics changes over the time, and observe mutual relation between these changes and community events.

Here, we define four events involving individuals (appear, disappear, join, leave). A node *appears* at a snapshot when it exists in that snapshot but was not present in the previous snapshots. It may *disappear* from one snapshot if it exists in that snapshot but will not occur in the next snapshots. A node *joins* to a community if it exists in that community but did not belong to a community with the same meta community in the previous snapshots. Finally, it *leaves* a community if it exists in that community but will not belong to a community with the same meta community in the next snapshots. Beside events, we also define different metrics and their temporal variations to measure the influences of an individual on others. For a complete definition and formula on events and metrics involving individuals, please refer to Chapter 6.

We furthermore define roles of individuals in a social network considering only structural properties of nodes, i.e. taking into account their interactions with other individuals, along with their affiliations to the communities. From this perspective, individuals can be classified as (see Figure 3.4 for illustrations):

1. with no affiliation to any community;
2. connecting multiple communities;
3. important members of a community;
4. ordinary/majority members of a community;
5. non-important members of a community, who do not noticeably affect the community.

Based on this classification, we define the following four fundamental roles. Note that, the framework we present can be extended to include other specific roles based on a particular application, following similar methodology we present here. As the basis, we limited our focus to the most cross context roles.

Leaders: Leaders are the outstanding individuals in terms of centrality or importance in each community. Leaders are pioneers, authorities, or administrators of communities.

Outermosts: Outermosts are the small set of least significant individuals in each group, whose influence and effect on the community are below the influence of the majority of the community members.

Mediators: Mediators are individuals who play an important role in connecting communities in a network. They act as bridges between distinct communities.

Outsiders: Outsiders are individuals who are not affiliated to any one community. They either have almost equal affiliation to different communities, or have very weak ties to a community. The latter are commonly referred to as outliers, whereas the former are exclusive mediators.

The formal formulation and extraction techniques of different roles are explained in Chapter 6.

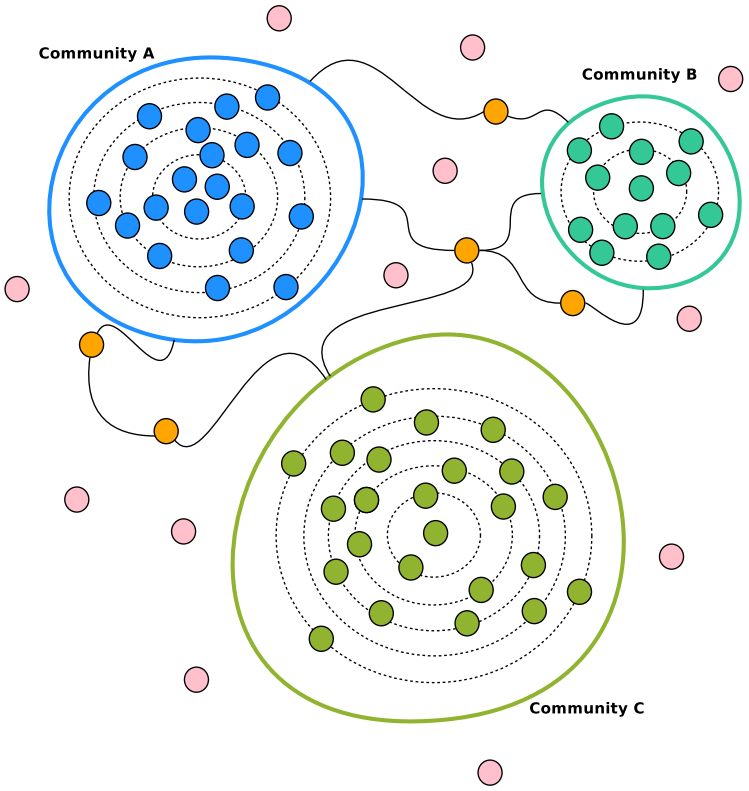


Figure 3.4: An intuitive illustration for different types of roles based on structural properties, community affiliations, and members position within communities. In this figure, three communities A, B, and C are shown. Nodes are colour coded based on their role and affiliation: orange represents nodes that are connecting communities to each other (These nodes might also be part of community, however, we have ignored that case in this figure for simplicity, but considered in our definitions.); pink represents nodes with no connections or very weak connections to communities; members of community A, B and C are coloured blue, dark green, light green respectively. Within each community, nodes are positioned based on their importance, i.e. closer to the borders of communities, the weaker and more inactive they are.

3.3.4 Contextual Text Mining

One of the challenges in studying the evolution of communities is how to select the appropriate similarity threshold k and how to compare different frameworks with each other. For the datasets containing text, for evaluation purposes only we propose to incorporate the extraction of the topics for the discovered communities. For these datasets, KEA [117] can be applied to produce a list of the keywords discussed within each community. The topics for each community are defined as its 10 most frequent keywords that were extracted by KEA. We expect that a community which survives multiple timeframes is more likely to continue discussions of

the same topics. Topics that persist in a community from one snapshot to the other are called mutual topics. Thus, the similarity threshold k that corresponds to the highest mutual topics in a specific application, could be the appropriate k for that scenario.

3.4 Community Prediction using Supervised Learning

As explained in Chapter 2, very little work has been done to study the reason why a community or an individual experiences a specific event or transition. Finding patterns of interaction and predicting the future structure of communities is also attractive for many areas such as disease modelling, information transmission, and business management. For the final stage of MODEC, we propose a machine learning model to predict the next event and transition of a community, based on the relevant structural and temporal properties. Furthermore, our models provide a complete predictive process for any transition and event that a community may undergo, and at the same time, identify the most prominent features for each community transition and event.

3.4.1 Feature Extraction and Selection

In predicting the trend of a community using predictive models, a response variable is a property related to community which can quantify a particular change in a community over time. A feature is any property that can influence one of the response variables. Thus, the first step is to select appropriate features from the properties related to entities and communities, as well as deciding on the response variables. Then, we can model the relationship between each response variable and one or more features, which can be used later to predict the most probable changes that may occur for a community.

To predict the next stage of a community, we consider five main classes of features: properties of its influential members, properties of the community itself, temporal changes of these properties, previously detected events and transitions,

and contextual properties. These features are explained in detail in Chapter 7.

3.4.2 Classification

Based on our proposed transitions and events, the changes that occur for a community are characterized as $\mathrm{survive}\{\mathrm{true}, \mathrm{false}\}$, $\mathrm{merge}\{\mathrm{true}, \mathrm{false}\}$, $\mathrm{split}\{\mathrm{true}, \mathrm{false}\}$, $\mathrm{size}\{\mathrm{expand}, \mathrm{shrink}\}$, and $\mathrm{cohesion}\{\mathrm{tighten}, \mathrm{loosen}\}$. All these events and transitions are binary which constitute the response variables in our predictive model. Since size and cohesion transitions are only defined for a survival community, we propose a multistage cascading technique to detect these two transitions. First, we predict the survival, then the detection of these transitions. These response variables are not mutually exclusive and may occur together at the same time, where different features may trigger them. Hence, we learn separate models to predict each of them. We propose applying different classification algorithms to accurately predict the next event and transition of a community, based on the structural and temporal properties. The complete explanation of our prediction process is outlined in Chapter 7.

Chapter 4

Iterative Local Expansion Community Mining

One of the key structural characteristics of networks is their community structure – groups of densely interconnected nodes. Communities in a dynamic social network span over periods of time and are affected by changes in the underlying population, i.e. they have fluctuating members and can grow and shrink over time.

The MODEC framework does not depends on the underlying community mining algorithm. However, in the case of independent community mining, we recommend the static L-metric community mining algorithm by Chen et al. [23]. The L-metric community mining can be easily tuned to detect both overlapping and non-overlapping community structures. Thus, this algorithm can be used in the two variations of independent/overlapping, and independent/non-overlapping (see Figure 3.1).

For the other two variations (i.e. incremental/overlapping, and incremental/non-overlapping) we propose incremental L-metric that extends the static L-metric to incrementally mine communities in a dynamic scenario following the direct approach. In the incremental L-metric, communities in the current time are obtained based on the communities from the past time frame. The main idea here is to change the initialization of the algorithm in a way that the computation at each snapshot starts by grouping nodes using the communities found at the previous snapshot. Compared to previous independent approaches, this incremental approach is more effective at detecting stable communities over time.

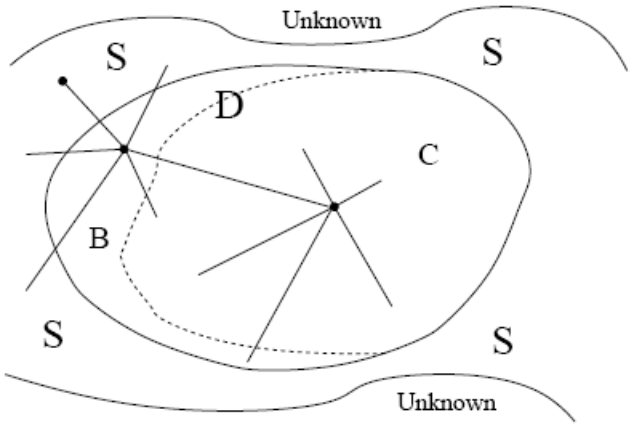


Figure 4.1: Local Community Definition. Figure reprinted from [23].

In the following, we first overview the static L-metric community mining, then we explain our proposed incremental L-metric in more detail. Furthermore, the extensive experimental studies on real datasets, demonstrate the applicability, effectiveness, and soundness of our proposed framework.

4.1 Static L-metric community mining

The static L-metric does not require any arbitrary thresholds or other parameters, and is robust against outliers. Its main assumption is that a community has fewer connections from its boundary nodes to the unknown portion of the graph, while having a greater number of connections within its local community [23]. In more detail, consider an undirected network G, with the known local portion of the graph denote as D. Two subsets of D are defined: the core node set C, where all neighbours of $v \in C$ belong to D; and the boundary node set B, where any node $v \in B$ has at least one neighbour outside D. The shell node set S is the set of nodes with limited available information and contains nodes that are adjacent to nodes in D but do not belong to D (See Figure 4.1).

Then the metric L is defined as the ratio of the community internal relation to the community external relation, i.e. L_{in}/L_{ex}, where L_{in} is measured by the average internal degree of nodes in D, and L_{ex} is measured by the average external degree of nodes in B.

The algorithm starts by choosing a start node for the community. At each step,

the algorithm adds to the community the neighbour node that gives the largest increase of L. More specifically, there are three situation in which the metric L increases after adding one node to the local community. Assume L'_{in}, L'_{ex} and L' are corresponding scores after merging a node v into D. The three cases that will result in $L' > L$ are:

1. $L'_{in} > L_{in}$ and $L'_{ex} < L_{ex}$
2. $L'_{in} < L_{in}$ and $L'_{ex} < L_{ex}$
3. $L'_{in} > L_{in}$ and $L'_{ex} > L_{ex}$

Nodes in the first case belong to the community, while nodes in the second case are outliers. The nodes in the third case can be hubs, or the first node of an enclosing community group that is going to be merged one by one. However, at the time of merging a node, it is too early to judge whether the incoming node is a hub or not. Therefore, nodes in the first and third cases are merged into the community temporarily. This process is continued until there are no candidate nodes that could give positive value to the community. After all qualified nodes are included, each node is re-examine by removing it from D and re-calculating the metric L to only include the nodes in the first case. The remaining nodes are then constituent of the local community. Algorithm 1 illustrates this step in details.

The L-metric community mining discovers communities of a whole graph by iteratively identifying a local community for a specific starting node. The procedure stops when we have learned the whole structure of the network. This approach is able to discover overlapping communities even though we do not specifically focus on finding such community property. Any node can belong to more than one community at the same time, since it can maximize metric L for more than one start nodes.

This algorithm can be easily changed to detect non-overlapping communities by adding the constraint to only consider those nodes whose community information are still unclear. For detailed information and algorithms please refer to [23]. Furthermore, when having dynamic social network, the independent community

Algorithm 1 Local Community Identification Algorithm [23]

Input: A social network G and a start node v_0
Output: A local community with its quality score L

1. Discovery Phase:

Add v_0 to D
Add v_0 to B
Add all v_0's neighbours to S
do
  for all $v_i \in S$ **do**
    compute L'_i
  end for
  Find v_i with the maximum L'_i, breaking ties randomly
  if v_i belongs to the first or third case **then**
    Add v_i to D
  else
    remove v_i from S
  end if
  Update B, S, C, L, L'
while $L' > L$

2. Examination Phase:

for all $v_i \in D$ **do**
  Compute L'_i
  if v_i belongs to the first case **then**
    keep v_i
  else
    remove v_i from D
  end if
end for

3. Final Phase:

if $v_0 \in D$ **then**
  return D
else
  there is no local community for v_0
end if

structures can be detected by applying the static L-metric on each snapshot separately.

4.2 Incremental L-metric community mining

Similar to the incremental Louvain proposed in [9], the main idea of our incremental L-metric is to change the initialization of the algorithm in a way that the computation at each snapshot starts by grouping nodes using the communities found at the previous snapshot.

Formally, the Incremental L-metric discovers communities for a dynamic social network with the following procedure. Initially, communities at snapshot 0 are mined using the static L-metric community mining. In iteration i, we consider the extracted connected components from communities of the previous snapshot (i.e. $i-1$) as its initialization state. This is due to the fact that the activities and interactions of the entities frequently change and vary in time, the community found at snapshot $i-1$ may not result in a connected component in snapshot i. Thus, in order to use communities C_{i-1} in the process of detecting communities C_i, we first extract connected components from communities C_{i-1}. Then, the nodes at snapshot i are grouped based on the extracted connected components. For each of the connected component $cc \in CC_i$, Algorithm 2 is executed iteratively. The connected components found are not only the members of the same communities at snapshot $i-1$, but also are connected to each other based on the interactions and connection at snapshot i. Each of these connected components are set as the seed for the L-metric community mining, where the algorithm construct its region D with the nodes of the given connected component. After that, the shell nodes of the region D have to be checked and if possible, added as the new community members. More specifically, a node v from the shell nodes is temporarily merged in the first and third cases into the community. After all qualified nodes are included, we re-examine each node by removing it from D and checking the metric value change if we merge it again. Now we only keep nodes if they are associated with the first case.

Similar to the static L-metric, the incremental L-metric can be used to detect both overlapping and non-overlapping community structures. When non-overlapping communities are needed, the *discovery phase* of Algorithm 2 should be changed to only add those nodes to the shell node set whose community structure is unclear.

Algorithm 2 Incremental Local Community Identification Algorithm

Input: A network at i^{th} snapshot G_i and a connected component cc_0
Output: A local community with its quality score L
1. Discovery Phase:
Add all nodes form cc_0 to D
Add all boundary nodes of cc_0 to B
Add all external neighbours of cc_0 to S
do
for all $v_i \in S$ **do**
compute L'_i
end for
Find v_i with the maximum L'_i, breaking ties randomly
if v_i belongs to the first or third case **then**
add v_i to D
else
remove v_i from S
end if
Update B, S, C, L, L'
while $L' > L$
2. Examination Phase:
for all $v_i \in D$ **do**
Compute L'_i
if v_i belongs to the first case **then**
keep v_i
else
remove v_i from D
end if
end for
3. Final Phase:
return D

A toy example to demonstrate the Incremental L-metric community mining is provided in Figure 4.2. At snapshot 0 (Figure 4.2a), static L-metric detects the red and green communities (Figure 4.2b). To detect communities at snapshot 1

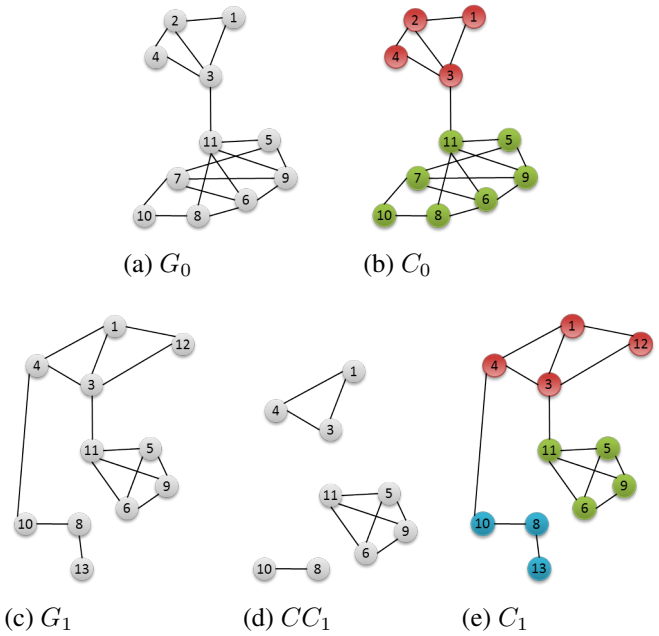


Figure 4.2: Example to illustrate incremental L-metric: (a) Network at snapshot 0; (b) Discovered communities at snapshot 0; (c) Network at snapshot 1; (d) Connected components from communities detected at snapshot 0, by taking into consideration the network structure at snapshot 1; (e) Discovered communities at snapshot 1.

(Figure 4.2c), first we have to group the nodes based on the communities detected at snapshot 0. Finding the connected components, three groups of nodes are extracted (Figure 4.2d). Each of these three connected components are then the input of Algorithm 2, which results in the detection of the red, green, and blue communities at snapshot 1 (Figure 4.2e).

After detecting communities in a dynamic scenario, as illustrated in the MODEC framework (Figure 3.1), the next stage is to analyze the evolution of these communities and study the behaviour of their individuals. In the next chapter, we will focus on different approaches to tackle this problem. The techniques proposed in the next chapter are dependent of the underlying community mining used to detect communities.

4.3 Dynamic community mining evaluation

Different community mining algorithms, and variations of community structures (i.e. independent/overlapping, independent/non-overlapping, incremental/overlap-

ping, and incremental/non-overlapping) discover communities from different perspective and may outperform others in specific classes of networks and have different computational complexities. Therefore, an important research direction is to evaluate and compare the results of community mining algorithms, and select the one providing more meaningful communities for each class of dynamic networks.

To validate the result of a community mining algorithm in a dynamic scenario, two approaches are available: *indirect evaluation*, and *relative evaluation*. *Indirect evaluation* involves comparing different community mining algorithms with the help of an event detection framework, to see how accurate are the events detected based on their resulted communities, and how well the detected events can be used to predict the next stage of the network. In Section 4.4 we will evaluate different community mining algorithms on two real datasets and investigate more on this approach.

Relative evaluation compares alternative clustering structures based on an objective function or quality index. In the static scenario, the quality of a community mining result is mainly measured by Modularity Q [71]. This criterion considers the difference between the fraction of edges that are within the community and the expected such fraction if the edges were randomly distributed.

Let A denoted an adjacency matrix of the graph G. The adjacency matrix of a graph G on n vertices is the $n \times n$ matrix $A = [A_{uv}]_{n \times n}$, where an entry A_{uv} of A is equal to 1 if the edge between node u and v exists, and zero otherwise. In case of weighted graph, the A_{uv} denoted the weight of the edge between node u and v. The Modularity Q for a partitioning on graph G is defined as:

$$Q = \frac{1}{2W} \sum_{u \neq v} [A_{uv} - \frac{\deg(u) \deg(v)}{2W}] \delta(u, v) \tag{4.1}$$

where W is the number of edges (or weighted sum of edges), i.e. $E = \frac{1}{2} \sum_{uv} A_{uv}$, $\delta(u, v)$ is 1 if nodes u and v are in the same community, 0 otherwise, and $\deg(v)$ is the degree (or weighted degree) of node v. Modularity Q ranges from -1 to 1. It is positive if the number of edges within groups exceeds the number expected on the basis of chance, and negative vice versa. Values of 0.3 or above are generally

considered high, and indicative of the kind of community structure commonly seen in various biological, technological and social networks [71].

The Modularity Q defined above is only applicable in the case of non-overlapping communities. Consequently, the quality of an overlapping community structure in the form of the modularity function can be written as [114]:

$$Q_{ov} = \frac{1}{2W} \sum_{u \neq v} [A_{uv} - \frac{\deg(u) \deg(v)}{2W}] \frac{\mathrm{A}(u) \cap \mathrm{A}(v)}{\mathrm{A}(u) \cup \mathrm{A}(v)} \tag{4.2}$$

where $\mathrm{A}(u)$ denotes the affiliation set of node u (i.e. the communities that node u is a member of them). For vertices u and v that always belong to the same community (i.e. $\mathrm{A}(u) \cap \mathrm{A}(v) = \mathrm{A}(u) \cup \mathrm{A}(v)$), their contribution to the modularity Q_{ov} is $A_{uv} - \frac{\deg(u) \deg(v)}{2W}$, while the contribution is 0, for vertices u and v that never belong to the same community (i.e. $affiliation(u) \cap \mathrm{A}(v) = \emptyset$. Otherwise, their contribution is within the range of $[0, A_{uv} - \frac{\deg(u) \deg(v)}{2W}]$.

However, in a dynamic scenario, the communities detected at one snapshot should not only be a good partitioning for that snapshot, but also a reasonable partitioning for the previous snapshot. Thus, we propose the *dynamic modularity*, DQ, to validate the quality of the partitioning on snapshot i defined as

Dynamic Modularity Q: Given a set of communities at snapshot i, C_i, the dynamic modularity of this partitioning can be calculated as

$$DQ_i = \alpha Q(G_i, C_i) + (1 - \alpha) Q(G_{i-1}, C_i) \tag{4.3}$$

where $Q(G_i, C_i)$ is computing the static Q modularity[1]for communities discovered in snapshot i. While $Q(G_{i-1}, C_i)$ is the value of modularity Q for communities at snapshot i computed over graph from the previous snapshot.

Average Dynamic Modularity Q: Given a dynamic network $\mathcal{G} = \{G_0, G_2, ..., G_{n-1}\}$, the average quality indicator on all the snapshots is

$$\overline{DQ} = \frac{1}{n} \sum_{i=1}^{n} DQ_i \tag{4.4}$$

[1]When having non-overlapping community structure Equation 4.1 is used to evaluate static modularity, while Equation 4.2 is applied in case on overlapping structure.

Similar to Modularity Q, average Dynamic Modularity Q, $\overline{DQ}$, ranges from -1 to 1. To evaluate different dynamic community mining algorithms on a specific scenario, the $\overline{DQ}$ of these algorithms can be compared with each other. The algorithm which results in higher $\overline{DQ}$ outperforms others on that scenario.

4.4 Experiments

In this section, we compare different variations of non-overlapping L-metric community mining, and investigate more on how to select the appropriate configuration on Enron email dataset.[2]

The Enron email dataset incorporates emails exchanged between employees of the Enron Corporation. The entire dataset includes a period of 15 years and its corresponding email communication network, for the entire period of time, has over 80,000 nodes and several hundred thousand edges, where nodes are employees and edges are emails between them. We study the year 2001, the year the company declared bankruptcy, and consider a total of 285 nodes and 23559 edges, with each month being one snapshot. For each of the 12 snapshots, one graph is constructed with the extracted employees as the nodes and email exchanged between them as the edges.

Furthermore, we also compare our proposed incremental L-metric with well-known FacetNet [64] incremental community algorithm. As stated in Section 4.3, the comparison between different community mining algorithms is performed from two perspectives: first, relatively based on a direct objective for dynamic communities, and then indirectly based on how much they improve the event detection framework and prediction accuracy. The objective function used here is our proposed *Dynamic Modularity Q* (Equation 4.3) with $\alpha = .5$.

As explained before, we incorporate contextual attributes, called topics, as a meta-data associated with the discovered communities. The text corresponding with each community depends on the dataset. For the Enron email dataset, email

[2]Studying different variation of *Modelling and Discretization of Network* (i.e. first stage of MODEC framework) is beyond the scope of this thesis. Here, we assume that the snapshot period is given.

messages sent or received within the members of a community are considered as the community's text. Whereas, for the DBLP dataset, the text is from the title and abstract of the papers co-authored in each community. For each discovered community, we apply Keyphrase Extraction Algorithm (KEA) [117] to produce a list of the keywords from the text associated with the community. The community's topic is then defined as its 10 most frequent keywords. Furthermore, the same procedure is applied to linked topics to the nodes as well. This contextual attribute is used later on the process of validation as well as empirical evidence to explain different community evolution and behaviour.

Different stages of our MODEC framework are integrated into Meerkat [22, 35], which is a social network analysis application under development by Dr. Osmar Zaïane and his lab. It offers facilities for automated community mining, various layout algorithms for helpful visualizations. Furthermore, Meerkat provides different tools to preview the graph at each timeframe, and also to track a community and an individual over time in a dynamic scenarios. In the rest of this chapter, we use Meerkat for most of the data visualization.

It is worth mentioning that, the static L-metric algorithm used on each snapshot is non-deterministic due to the fact that it relies on an arbitrary order of the entities. This non-deterministic behaviour might produce different communities even if the input graph does not change. This instability produces noise that makes the tracking very difficult. Thus, in order to prevent the non-deterministic nature of the static L-metric algorithm we calculated the global closeness centrality scores for all the entities in the aggregate graph. This score is then used to produce an ordered set of individuals which is used as an input to our community mining algorithm. The global centrality scores prevent the communities to change each time we apply the L-metric community mining algorithm. Thus, our experiments are consistent and deterministic.

4.4.1 Enron Email Dataset

Figure 4.3 presents a detailed comparison of different variation of community mining algorithms; where quality, size, and number of communities over the time is

depicted respectively. In Figure 4.3(a), we can clearly see that the proposed incremental approach is consistently detecting communities with higher quality, complying with both current and temporal information. Figures 4.3(b),(c) are shedding light on another difference between the incremental and independent approach. As we can see here, the average size of communities is much lower for the independent method. This is due to the fact that it failed to detect stable communities that span over time and instead detected several small communities, which is not surprising since it only looks at the current timeframe to mine communities. The FacetNet mining fails similarly to the independent approach. One of the disadvantages of the FacetNet mining is that the number of communities should be similar for all the snapshots. As stated in [64], the number of communities is the one maximizing the average modularity over all the snapshots. For our results here, we run the FacetNet with different number of communities and chose 6, that resulted in the highest modularity.

The average dynamic modularity Q (Equation 4.4) for the independent/non-overlapping, incremental/non-overlapping, and FacetNet are 0.45, 0.49, and 0.47 respectively. This experiment indicates that different variations of L-metric community mining have very similar dynamic modularity Q, while the number of detected communities and their size are varied. Furthermore, the choice of community mining variations reveals temporal and evolutionary behaviour of communities and the structure of networks from different perspective. Hence, the community mining variation should be selected based on the underlying scenario and application.

The effect of the different community mining variations on the detected events is shown in Table 4.1, where the total number of events for each type detected during the 12 snapshots is provided. The Independent L-metric is too dynamic, detecting communities that vary much between snapshots, and therefore, resulting in too many triggered events, e.g. 19 forms, 19 dissolve. The FacetNet algorithm, on the other hand, is too stable, resulting in no merge or split events and only having survival events. Which is a consequence of how it detects communities over all the snapshots and has less emphasis on what is happening in each snapshot, and therefore fails to detect any of the events. The Incremental L-metric has *the*

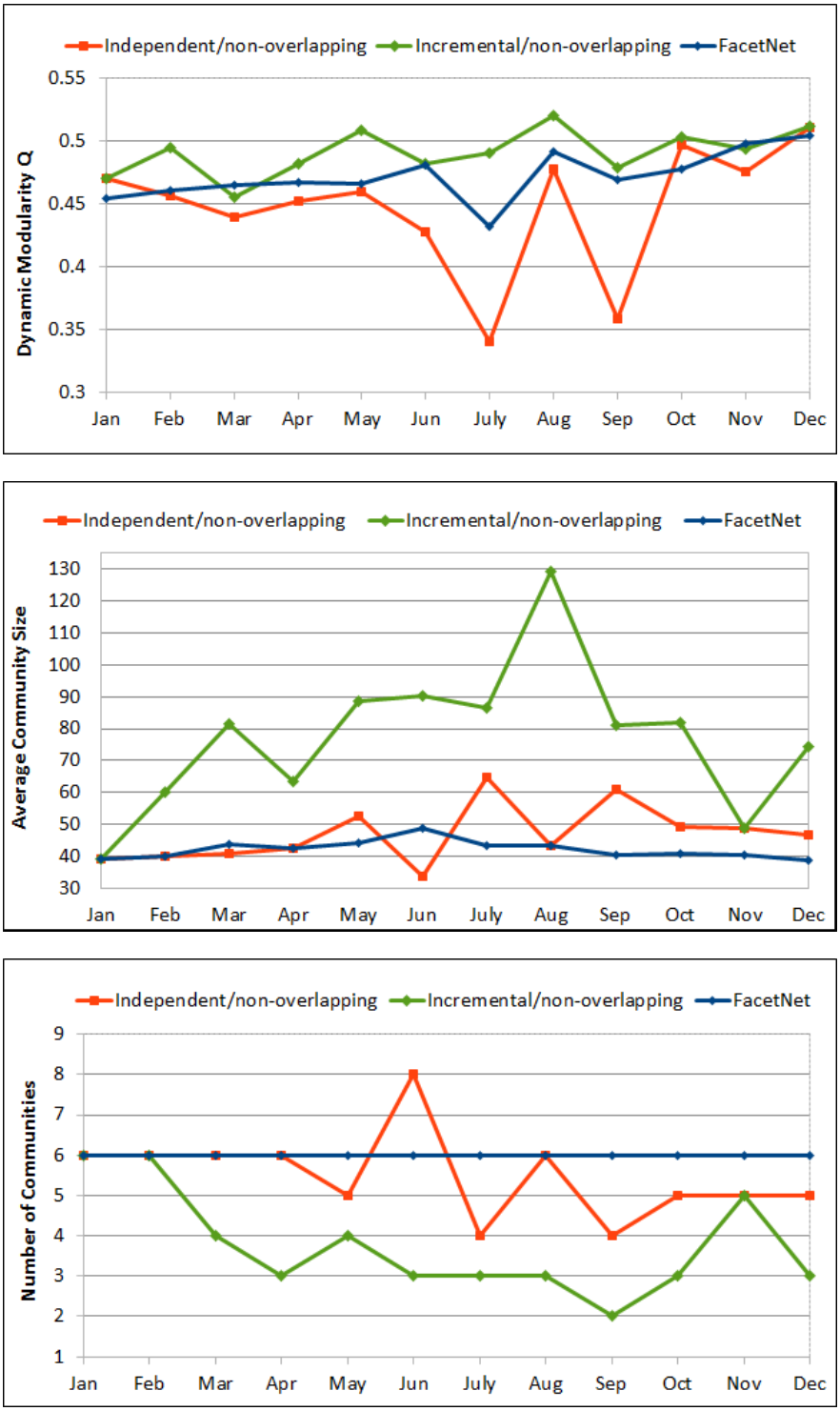


Figure 4.3: Relative Evaluation on Enron email dataset: (a) dynamic modularity; (b) size of communities; (c) number of communities for each snapshot.

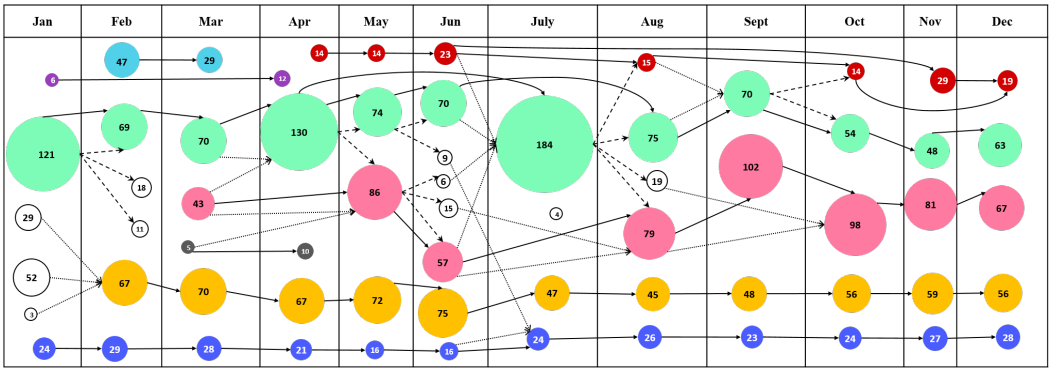
Table 4.1: Indirect Evaluation on Enron email dataset: comparison of Events Detected based on different algorithms.

Algorithm	Form	Dissolve	Survive	Split	Merge	Mutual Topics
Independent/non-overlapping	19	19	46	7	11	3.83/10
Incremental/non-overlapping	10	10	32	5	8	4.12/10
FacetNet[64]	6	6	66	0	0	4.02/10

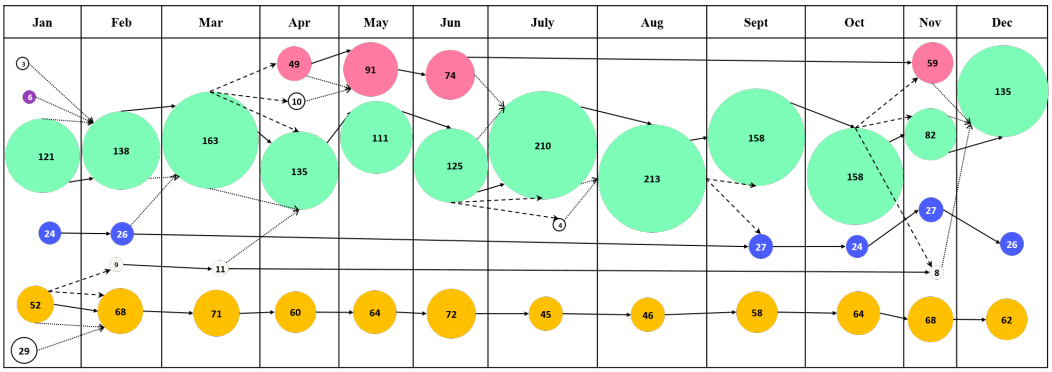
balance between the two, i.e. it correctly determines the communities survived over timeframes by incorporating the temporal information, and at the same time, detects other types of events reasonably.

The detailed communities and events detected for non-overlapping community mining algorithms are further shown in Figure 4.4. Here communities at each snapshot are marked with different colours, where these colours are the notion of meta communities (the communities without colour are the ones that only exist for one snapshot). Furthermore, solid, dashed, and dotted arrows show detected *survive*, *split*, and *merge* events respectively. The communities detected by the Independent L-metric algorithm in Figure 4.4a, are too dynamic and unstable; which result in triggering too many events. For the first two snapshots for example, we can see that it failed to detect the green/largest community correctly, having that community as several separate smaller communities including the cyan/47 member community, which is not a distinct community and disappears after only one snapshot. The Incremental L-metric, Figure 4.4b, started with the same communities in the first snapshot, detects the survival of this green community correctly, by incorporating the temporal information. Its communities also have a relatively higher quality, with $DQ = .495$ to $DQ = .456$ of the independent method. The FacetNet communities are different than those found by the Independent and Incremental L-metric methods. And at the same time, have lower quality index of DQ. These communities are too stable and fail to trigger any events other than survival. Thus one is not able to see the patterns of change in the structure of the network using its detected communities.

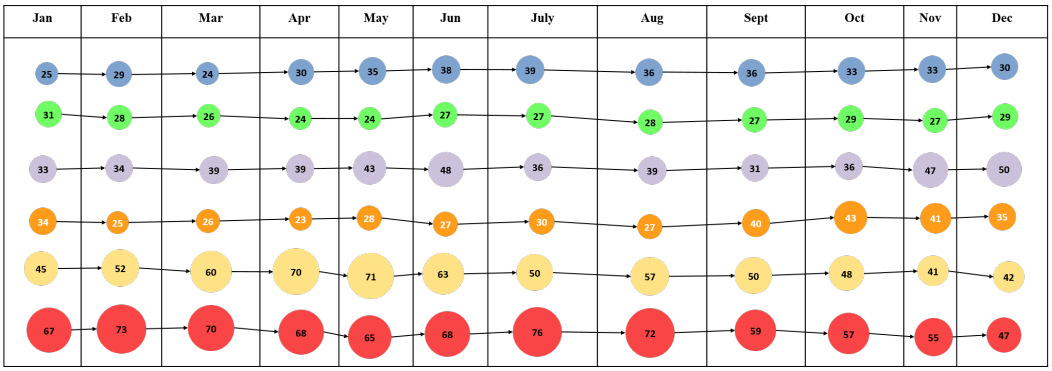
In the Enron dataset, a community which *survives* multiple timeframes is more likely to continue discussions of the same topics. Therefore, we also incorporate the extraction of the topics for the discovered communities; where we apply KEA [117] to produce a list of 10 most frequent keywords discussed in the emails within each



(a) Events Detected for Independent L-metric



(b) Events Detected for Incremental L-metric



(c) Events Detected for FaceNet

Figure 4.4: Events Detected on Enron email dataset: Communities in (a)/(c) are too unstable/stable, while in (b) we have a balance between the change and stability. Solid, dashed, and dotted arrows show detected *survive*, *split*, and *merge* events respectively.

community. Topics that persist in a community from one snapshot to the other are called mutual topics. We incorporate topics extracted for each community to find out which algorithm results in the most appropriate community evolutions. The average mutual topics between any two survival communities during the observation time is calculated for each algorithm, which are reported at the last column of the Table 4.1. Here, the highest mutual topics out of the top 10 most frequent keywords is obtained when using the Incremental L-metric framework. Thus, the Incremental L-metric also results in *the most meaningful* community evolution for Enron.

4.5 Summary

One of the challenging research problems in dynamic social networks is to mine communities and analyze their evolution over the observation time. The traditional approach to solve this problem is to extract communities at each snapshot independent of the communities at other snapshots or the historic data. In this paper, we overviewed and classified different dynamic community mining approaches. We then proposed an Incremental L-metric community mining approach to consider both current and temporal data in the process of mining communities. The proposed method is then compared with its equivalent independent version and also with the most commonly used dynamic community method –FacetNet. Compared to these two methods, the Incremental L-metric method detects communities with higher quality when assessed directly with a modified version of Q modularity for the dynamic scenario. In addition, it is more successful in detecting the evolution patterns of the communities and triggering appropriate events, when used in our event detection framework, MODEC. The Independent L-metric is too unstable and triggers too many events, while the FacetNet is too stable and triggers no events other than survivals. Our incremental method, on the other hand, has the balance and provides meaningful communities and events by incorporating the temporal information.

Chapter 5

Empirical and Evolutionary Community Analysis

Most social networks are dynamic, and studying the evolution of these networks over time could provide insight into the behavior of individuals expressed by the nodes in the graph and the flow of information among them. The analysis of communities and their evolutions can help determine the shifting structural properties of the networks. We present a framework for modeling and detecting community evolution over time. In this chapter, series of significant events and transitions are defined to characterize the evolution of networks in terms of its communities and individuals. We then present experiments to explore the dynamics of communities on the Enron email and DBLP datasets. Evaluating the events using topics extracted from the detected communities demonstrates that we can successfully track communities over time in real datasets.

5.1 Event Formulation

Given the definition of meta communities, similarity, and similarity threshold k, a community C_i^p at the i^{th} snapshot may undergo different conversion in later snapshots. Community C_i^p *splits* at snapshot $j > i$ if it fractures into multiple communities with at least k proportion of their members from C_i^p. Community C_i^p *survives* if there is a community C_j^q at snapshot $j > i$ such that their meta communities are identical. In the case where there is no such community, C_i^p *dissolves*. Only the *survive* and *dissolve* events are mutually exclusive, while the *split* event can be

combined with the other two. Community C_i^p *splits* and *survives* at the j^{th} snapshot if it fractures into more than one community and one of these communities has the same meta community as C_i^p. Community C_i^p *splits* and *dissolves* at the j^{th} snapshot if it fractures into other communities and none of these communities have the same meta community as C_i^p.

In addition to the three events mentioned above, a set of communities in C_i can *merge* together in community C_j^q at snapshot $j > i$. The *merge* event occurs when at least k proportion of the members from multiple communities in C_i, exist in C_j^q. Furthermore, at any snapshot there may be newly *formed* communities. These communities are the ones that do not belong to any of the already existing meta communities.

In the following, the formal definitions of these events are provided, where $\text{match}(C_i^p, j)$ denotes the optimal match for C_i^p at j^{th} snapshot which is the results of the optimal matching algorithm for community C_i^p:

Form: A community C_i^p *forms* at i^{th} snapshot if there is no community match for it in any of the previous snapshots:

$$\text{form}(C_i^p, i) = \text{true} \quad \text{iff} \quad \forall j < i \quad \text{match}(C_i^p, j) = \emptyset \tag{5.1}$$

Dissolve: A community C_i^p *dissolves* at i^{th} snapshot if there is no community match for it in any of the next snapshots:

$$\text{dissolve}(C_i^p, i) = \text{true} \quad \text{iff} \quad \forall j > i \quad \text{match}(C_i^p, j) = \emptyset \tag{5.2}$$

Survive: A community C_i^p *survives* at i^{th} snapshot if there exists a snapshot $j > i$ that contains a community match for C_i^p:

$$\text{survive}(C_i^p, i) = \text{true} \quad \text{iff} \quad \exists j > i \text{ and } \exists C_j^q \in C_j \quad \text{match}(C_i^p, j) = C_j^q \tag{5.3}$$

Split: A community C_i^p at i^{th} snapshot *splits* to a set of communities $C_j^* = \{C_j^1, ..., C_j^n\}$ at snapshot $j > i$ if at least k proportion of the members of the communities in C_j^* are from community C_i^p. Also in order to prevent the case where most of

the members of C_i^p leave the network, the mutual members of the union of the communities in C_j^* with C_i^p should be greater than k proportion of C_i^p:

$$\begin{aligned}&\text{split}(C_i^p, i) = \text{true} \quad \text{iff} \quad \exists j > i \text{ and } \exists C_j^* = \left\{C_j^1, ..., C_j^n\right\} \in C_j \quad \text{where} \\ &1) \forall C_j^r \in C_j^* \quad \frac{|V_i^p \cap V_j^r|}{|V_j^r|} \geq k \qquad 2) \frac{|(V_j^1 \cup V_j^2 ... \cup V_j^n) \cap V_i^p|}{|V_i^p|} \geq k\end{aligned} \tag{5.4}$$

Merge: A set of communities $C_i^* = \{C_i^1, ..., C_i^n\}$ at i^{th} snapshot *merge* to C_j^q at snapshot $j > i$ if C_j^q contains at least k proportion of the members from each community in C_i^*. Also to prevent the case where most of the members of C_j^q did not exist before, the mutual members of the union of all communities in C_i^* with C_j^q should be greater than k proportion of C_j^q:

$$\begin{aligned}&\text{merge}(C_i^* = \{C_i^1, ..., C_i^n\}, i) = \text{true} \quad \text{iff} \quad \exists j > i \text{ and } \exists C_j^q \quad \text{where} \\ &1) \forall C_i^r \in C_i^* \quad \frac{|V_i^r \cap V_j^q|}{|V_i^r|} \geq k \qquad 2) \frac{|(V_i^1 \cup V_i^2 ... \cup V_i^n) \cap V_j^q|}{|V_j^q|} \geq k\end{aligned} \tag{5.5}$$

5.2 Metric and Transition Formulation

To quantify the structural properties of a community we consider different metrics including its size (number of nodes), cohesion, density, and clustering coefficient. In the following we provide the formal definition of these metrics.

Cohesion Metric: Cohesion of a community determines how closely its members interact with each other relative to outside of the community. Formally, cohesion of a community C_i^p at snapshot i defined as:

$$\text{cohesion}(C_i^p, i) = \frac{\frac{2|E_i^p|}{|V_i^p|(|V_i^p|-1)}}{\frac{|OE_i^p|}{|V_i^p|(|V_i|-|V_i^p|)}} = \frac{2|E_i^p|(|V_i| - |V_i^p|)}{|OE_i^p|(|V_i^p| - 1)} \tag{5.6}$$

where $|OE_i^p|$ is the set of outer edges of community C_i^p.

Density Metric: Density of a community C_i^p at snapshot i is the ratio of edges to the maximum possible edges:

$$\text{density}(C_i^p, i) = \frac{2|E_i^p|}{|V_i^p|(|V_i^p| - 1)} \tag{5.7}$$

Clustering coefficient Metric: The clustering coefficient of a node is the ratio of edges between neighbours of a nodes to the maximum possible edges between them. More formally, clustering coefficient of node v is:

$$\text{clusterCoeff}(v) = \frac{|\{(u,w)|(v,u),(v,w),(u,w) \in E\}|}{|\{(u,w)|(v,u),(v,w) \in E\}|} \tag{5.8}$$

The clustering coefficient of a community C_i^p at snapshot i is then defined as the mean of clustering coefficient of all its members:

$$\text{clusterCoeff}(C_i^p, i) = \frac{\sum_{v \in C_i^p} \text{clusterCoeff}(v)}{|V_i^p|} \tag{5.9}$$

The above defined metrics are temporal, i.e. their values will change over time for a specific meta community. Thus, to study the temporal structural properties of a community, the trend of these metrics will be analyzed.

Temporal Metrics of a Meta Community: Given a set of snapshots 0, 2, ..., $n-1$, and meta community $M = \{C_b^{p_b}, ..., C_i^{p_i}, ..., C_d^{p_d}\}$, the temporal metrics of a meta community is as follows:

$$\begin{aligned}
&\textit{size}(M) = \\
&\quad [|V_b^{p_b}|, ..., |V_i^{p_i}|, ..., |V_d^{p_d}|] \\
&\textit{cohesion}(M) = \\
&\quad [\text{cohesion}(C_b^{p_b}, b), ..., \text{cohesion}(C_i^{p_i}, i), ..., \text{cohesion}(C_d^{p_d}, d)] \\
&\textit{density}(M) = \\
&\quad [\text{density}(C_b^{p_b}, b), ..., \text{density}(C_i^{p_i}, i), ..., \text{density}(C_d^{p_d}, d)] \\
&\textit{clusterCoeff}(M) = \\
&\quad [\text{clusterCoeff}(C_b^{p_b}, b), ..., \text{clusterCoeff}(C_i^{p_i}, i), ..., \text{clusterCoeff}(C_d^{p_d}, d)]
\end{aligned} \tag{5.10}$$

where $b = \text{birth}(M)$, and $d = \text{death}(M)$.

Another metric that characterizes a meta community is the members fluctuation which characterizes the average similarity of its constituent communities between consecutive snapshots.

Member Fluctuation Metric: Consider a meta community $M = \{C_b^{p_b}, ..., C_i^{p_i}, ..., C_d^{p_d}\}$. The member fluctuation of the meta community M is

$$\text{fluctuation}(M) = 1 - \frac{\sum_{i=b}^{d} \text{sim}(c_i^{p_i}, c_{i+1}^{p_{i+1}})}{|M|} \tag{5.11}$$

where $b = \text{birth}(M)$, and $d = \text{death}(M)$. Similarity equation, sim, is defined previously in Chapter 3, Equation 3.1.

Note that member fluctuation is different from the other temporal metrics. Member fluctuation metric is an average over the observation time, while the other temporal metrics are lists of numbers. To provide better encapsulation on the temporal metrics, we furthermore, define different transitions: *size transition*, *cohesion transition*, *leader transition*, and *unity transition*. These four transitions are not mutually exclusive and may occur together at the same time. Their formal definitions are provided in the following.

Size transition: Let C_i^p *survive* to C_j^q at snapshot $j > i$. Community C_i^p *shrinks* at j^{th} snapshot if its number of members is greater than the number of members of C_j^q. When the number of members of C_i^p is less than the number of members of C_j^q, community C_i^p *expands*:

$$\text{size}(C_i^p, C_j^q) = \begin{cases} \text{shrink} & \text{iff} \quad \text{match}(C_i^p, j) = C_j^q \quad \text{and} \quad |V_i^p| > |V_j^q| \\ \text{expand} & \text{iff} \quad \text{match}(C_i^p, j) = C_j^q \quad \text{and} \quad |V_i^p| < |V_j^q| \end{cases} \tag{5.12}$$

Cohesion transition: Let C_i^p *survive* to C_j^q at snapshot $j > i$. Community C_i^p becomes more *loose* at j^{th} snapshot if its cohesion becomes less in C_j^q. When the cohesion of C_i^p is greater than the cohesion of C_j^q, community C_i^p has become more *tighten*:

$$\text{cohesion}(C_i^p, C_j^q) = \begin{cases} \text{loosen} & \text{iff } \text{match}(C_i^p, j) = C_j^q \text{ and} \\ & \frac{|E_i^p|}{|V_i^p|(|V_i^p|-1)} < \frac{|E_j^q|}{|V_j^q|(|V_j^q|-1)} \\ \text{tighten} & \text{iff } \text{match}(C_i^p, j) = C_j^q \text{ and} \\ & \frac{|E_i^p|}{|V_i^p|(|V_i^p|-1)} > \frac{|E_j^q|}{|V_j^q|(|V_j^q|-1)} \end{cases} \tag{5.13}$$

As stated before, we developed a framework for structural role mining and identified outsiders, outermosts, mediators, and leaders. Among these roles, leaders of a community are the important one to study. Thus, we find the information about the people associated with nodes having the role of a leader, and we observe how the leaders of a community change through time.

Leader transition: Let C_i^p *survive* to C_j^q at snapshot $j > i$. The set of leaders of community C_i^p shifts at j^{th} snapshot if the nodes having the role of a leader have changed in C_i^p and C_j^q:

$$\begin{aligned} &\text{leaderShift}(C_i^p, C_j^q) = \text{true} \quad \text{iff} \\ &\text{match}(C_i^p, j) = C_j^q \quad \text{and} \quad \text{leader}(C_i^p, i) \cap \text{leader}(C_j^q, j) = \emptyset \end{aligned} \tag{5.14}$$

The above defined transitions are compatible in the case of overlapping and non-overlapping community structure. When community structures are overlapping, the extent to which different communities overlap is also a relevant property of a network. There are different observation about the overlaps between communities. The behaviour of the cumulative overlap size distribution, is close to a power law with a rather large exponent which states that there is no characteristic overlap size in the networks [86]. Furthermore, the overlaps between communities tend to be more densely connected than the non-overlapping parts [120, 121]. This observation is due to the fact that the more communities a pair of nodes has in common, the more likely they are connected in the network. For instance, people with more similar hobbies have a higher chance of becoming friends as well. In this thesis, we are not interested in the property of the overlapping part, rather we will focus on how it may change over time.

When having overlapping structure, any two communities C_i^p and C_i^r at snapshot i can share a set of nodes and edges. Naturally, this overlap can be also modelled by a graph of nodes denoted as $\text{overlap}(C_i^p, C_i^r, i) = (V_i^p \cap V_i^r, E_i^p \cap E_i^r)$. We introduce the overlapping score of two communities as follow:

$$\text{ovScore}(C_i^p, C_i^r, i) = \frac{1}{2}\left(\frac{|V_i^p \cap V_i^r|}{\max(|V_i^p|, |V_i^r|)} + \frac{|E_i^p \cap E_i^r|}{\max(|E_i^p|, |E_i^r|)}\right) \tag{5.15}$$

Our overlapping score considers both the fraction of common nodes and also the fraction of common edges. The overlapping score between two communities may change over time. Thus, we define the following transition:

Unity transition: Let C_i^p, and C_i^r *survive* to C_j^q, and C_j^s at snapshot $j > i$ respectively. The unity between community C_i^p and C_i^r at j^{th} snapshot becomes *disjoint* if its overlapping score decreases, whereas, the intersection becomes *united* if the overlapping score increases:

$$\text{match}(C_i^p, j) = C_j^q \text{ and } \text{match}(C_i^r, j) = C_j^s$$

$$\text{unity}(C_i^p, C_i^r, C_j^q, C_j^s) = \begin{cases} \text{disjoint} \quad \text{iff} \\ \quad \text{ovScore}(C_i^p, C_i^r, i) < \text{ovScore}(C_j^q, C_j^s, j) \\ \\ \text{united} \quad \text{iff} \\ \quad \text{ovScore}(C_i^p, C_i^r, i) > \text{ovScore}(C_j^q, C_j^s, j) \end{cases} \tag{5.16}$$

The summary of the events, metrics, and transitions related to a community is provided in Table 5.1.

5.3 Experiments

In this section, we validate the effectiveness and feasibility of our proposed *Empirical and Evolutionary Community Analysis* techniques through experiments on the Enron email dataset and DBLP dataset. On both these datasets, we gain insights on the impact of the similarity threshold on the evolution of the communities, and select the optimal similarity threshold by automatic extraction and the investigation of the contextual attributes associated with the communities. We furthermore compare the MODEC framework with the other event-based frameworks including Asur et al. [7], Palla et al. [85], and Greene et al. [45]. It is worth mentioning that the work done in [106] is also close to our work. However, at the time that this work was conducted, the implementation of their algorithm was not available and we could not compare our MODEC framework with the one proposed there without risk of code bias.

Table 5.1: Events and Transition Involving Communities: Definition of symbols

Symbol	Definition
Community Events	
$\text{form}(C_i^p, i)$	form event for community C_i^p at snapshot i
$\text{dissolve}(C_i^p, i)$	dissolve event for community C_i^p at snapshot i
$\text{survive}(C_i^p, i)$	survive event for community C_i^p at snapshot i
$\text{split}(C_i^p, i)$	split event for community C_i^p at snapshot i
$\text{merge}(C_i^* = \{C_i^1, ..., C_i^n\}, i)$	merge event for set of communities C_i^* at snapshot i
Community Metrics	
$\text{cohesion}(C_i^p, i)$	cohesion of community C_i^p at snapshot i
$\text{density}(C_i^p, i)$	density of community C_i^p at snapshot i
$\text{clusterCoeff}(C_i^p, i)$	clustering coefficient of community C_i^p at snapshot i
$\text{leader}(C_i^p, i)$	leaders of community C_i^p at snapshot i
$\lvert\,\text{join}(C_i^p), i\rvert$	number of nodes join community C_i^p at snapshot i
$\lvert\,\text{leave}(C_i^p), i\rvert$	number of nodes leave community C_i^p at snapshot i
$\text{ovScore}(C_i^p, C_i^r, i)$	overlapping score between communities C_i^p, and C_i^r at snapshot i
Temporal Community Metrics	
$\textit{size}(M)$	affiliation size of meta community M over time
$\textit{cohesion}(M)$	cohesion of meta community M over time
$\textit{density}(M)$	density of meta community M over time
$\textit{clusterCoeff}(M)$	clustering coefficient of meta community M over time
$\text{fluctuation}(M)$	average members fluctuation of meta community M
Community Transitions	
$\text{size}(C_i^p, C_j^q)$	size transition between communities C_i^p and C_j^q at snapshots i and j respectively
$\text{cohesion}(C_i^p, C_j^q)$	cohesion transition between communities C_i^p and C_j^q at snapshots i and j respectively
$\text{leaderShift}(C_i^p, C_j^q)$	leader transition between communities C_i^p and C_j^q at snapshots i and j respectively
$\text{unity}(C_i^p, C_i^r, C_j^q, C_j^s)$	unity transition between communities C_i^p, and C_i^r at snapshot i, and communities C_j^q, and C_j^s at snapshot j

The Enron email dataset incorporates emails exchanged between employees of the Enron Corporation. The entire dataset includes a period of 15 years and its corresponding email communication network, for the entire period of time, has over 80,000 nodes and several hundred thousand edges, where nodes are employees and edges are emails between them. We study the year 2001, the year the company declared bankruptcy, and consider a total of 285 nodes and 23559 edges, with each month being one snapshot. For each of the 12 snapshots, one graph is constructed with the extracted employees as the nodes and email exchanged between them as the edges.

For the DBLP dataset, the co-authorship network related to the field of database and data mining from year 2001 to 2010 is extracted. This dataset contains a total of 19461 authors and 72525 edges, where nodes are authors and edges are the co-authored papers between them. Hence, the co-authoring relationship connects two authors if they have co-authored a paper that was published in any of the database and data mining conferences. We define the duration of a snapshot to be 1 year, and thus for each of the 10 years, a graph with authors as its nodes and co-authored papers as its edges is constructed.

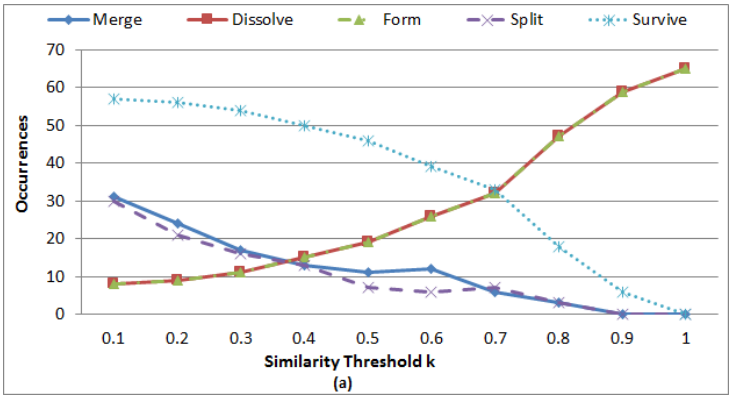
The Enron and DBLP datasets contain text, and an important factor influencing the future shape of the communities is the semantic nature of the interactions between individuals in these two datasets. For instance, in DBLP, two authors are connected if they publish a paper together. The topic or the subject area of the paper will definitely influence future collaborations for each of these authors. If there are different authors working on similar topics, the chances of them collaborating in the future is higher than two authors working on unrelated areas. In the case of Enron, the topic of the email exchanged within the employees will influence the future interactions between those employees. In the following experiments, we examine the influence of the semantics of the interaction on future interactions and incorporate semantic information for reasoning about evolution. In addition, we also develop measures for evaluating the events obtained from a semantic standpoint. In order to do so, the topics continuation (i.e. semantic reasoning) is observed on different variation of the similarity threshold k. When having lower value for k, we allow the

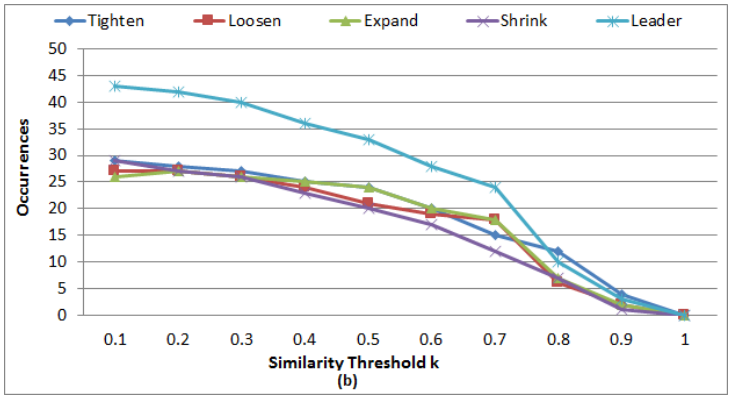
two communities with lower overlaps to be similar. In this case, due to the lower number of overlapping individuals the number of mutual topics will go down as well. On the other hand, with high value of k, the overlapping between the similar communities would be a bigger group containing more individuals. As a results, a larger number of topics will be discussed within a community. Hence, we need to tune similarity threshold k in order to find the highest number of mutual topics. This characteristic justifies the bell shape of the plots in Figure 5.1d and Figure5.5d. Furthermore, the topics continuation will effect the similarity threshold k that will be chosen for any given datasets.

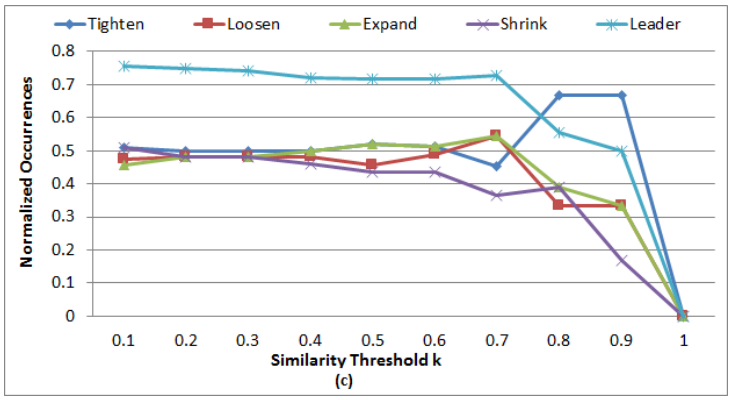
5.3.1 Enron Email Dataset

The impact of similarity threshold k on the community evolution is investigated for the Enron dataset. The similarity threshold is varied from 0.1 to 1.0 in steps of 0.1 and the number of events occurring during the 12 snapshots is counted for each k step (Figure 5.1a). We observe that the choice of k has a noticeable effect on the detected events: the number of *survival*, *merge*, and *split* events drops as k increases, while there are more *dissolution* and *formation*. Note that in our framework, the number of *formation* and *dissolution* are both equal to the number of discovered meta communities, thus, they are exactly the same in Figure 5.1a. With low values of k, more communities are matched together, thus, we can observe a significant number of *surviving* communities. However, high values of k result in a conservative matching behaviour and short-life meta communities. The number of different transitions is also compared on different variations of k (Figure 5.1b). As expected, since the transitions are defined for the survival communities, their numbers would drop with the increase of k. To better compare the number of transitions, Figure 5.1c depicts the normalized number of transitions over number of *survival* communities. Again we observed that the number of detected transitions gradually decreases as k increases.

The question that arises here is which similarity threshold results in the most appropriate community evolutions for the Enron dataset. For the datasets containing text, we propose to incorporate contextual attributes by the extraction of the topics







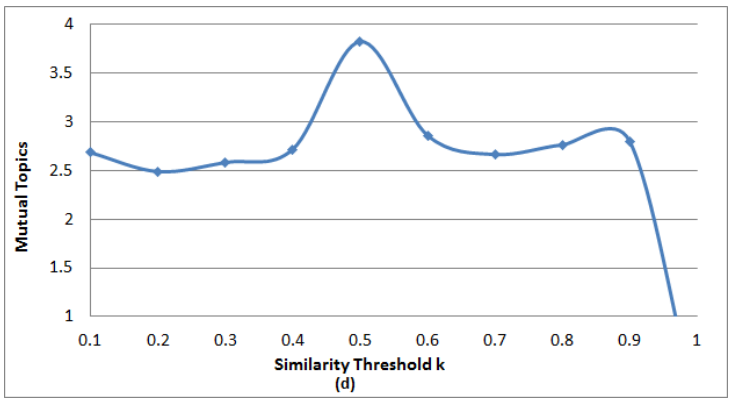


Figure 5.1: The impact of similarity threshold k on Enron email dataset: (a) events; (b) transitions (c) normalized transitions by survival; (d) average mutual topics for different k.

for the discovered communities. We expect that a community which *survives* multiple timeframes is more likely to continue discussions of the same topics. Topics that persist in a community from one snapshot to the other are called mutual topics. Figure 5.1d shows the average mutual topics between any two *survival* communities during the observation time for different k. The survival communities mostly discuss the same topics, thus, the k that corresponds to the highest mutual topics illustrates the community evolution better than the others. Figure 5.1d shows that as k increases from 0.1 to 0.5 the average number of mutual topics also increases. However, for $k > 0.5$ the number of mutual topics decreases sharply. Based on this observation, we can conclude that for the Enron dataset the choice of $k = 0.5$ results in the most meaningful community evolution since it has highest average mutual topics. It is worth mentioning that for some applications, more than one k may results in maximum mutual topics. In this case based on the characteristic of the social network, one of the k that results in the maximum mutual topics is selected: A high similarity threshold k is required for the networks with rather stable communities, while a low similarity threshold k is selected for the network with highly dynamic communities. Furthermore, the appropriate similarity threshold might also be defined based on the theoretical considerations. The remaining experiments on the Enron dataset are based on $k = 0.5$ unless otherwise stated. Furthermore, since all the other events are based on the survival event, we used the same similarity threshold $k = 0.5$ for all the events and transitions related for the Enron dataset.

With $k = 0.5$, the meta communities and events detected on Enron dataset is shown in Figure 5.2. Here communities at each snapshot are marked with different colours, where these colours are the notion of meta communities (the communities without color are the ones that only exist for one snapshot). Furthermore, solid, dashed, and dotted arrows show detected *survive*, *split*, and *merge* events respectively. As an example of the topic continuation, consider the blue meta community (the bottom meta community) in Figure 5.2. “Transwestern Pipeline Company”, for instance, was consistently the most frequently discussed topic in that community for the whole year, while not appreciably discussed in other communities.

Furthermore, the key intuition that we employ here is that the probability of a merge event depends on the merge of the topics between the involving communities. For instance, if two communities are comprised of authors working on different topics, it stands to reason that there would be a merge event between them if they decide to work on a topic which is a medley of the topics related to each of them. A similar expectation is also made for *splits*; there would be a split event between two communities if the topics also split accordingly.

As an example of merge and split verification, consider detected communities depicted in Figure 5.2. When community light green (size 70) and community pink (size 43) in March from Figure 5.2 merged, the resulting community light green (size 130) continued discussing many topics from light green and fewer topics from pink: "Federal Energy Regulatory Commission", and "Pacific Gas and Electric Company", which are the most frequent topics of light green and pink respectively, are also discussed in the merged community in April. However, the majority of the topics in the merged community are from the community light green, thus confirming the survival event as well. Finally, when the merged community light green (size 130) splits to two communities in May, the resulting light green (size 74) and pink (size 86) discussed the same topics as they did before the merge, validating both the *split* and the *survive* events (Figure 5.3). Looking more closely at the topics discussed by each individual inside those communities, we chose Stanley Horton as a case study, who was the president of Enron Gas Pipeline at the time and was always part of the light green community. We found that in March, Stanley Horton main discussions were about "Federal Energy Regulatory Commission", and "ISO" which were the two topics of the light green community. However, in April he was mostly talking about both "Pacific Gas and Electric Company" and "Federal Energy Regulatory Commission" which is the combinations of the topics discussed previously in the light green community and the pink community. In May he switched back discussing about "Federal Energy Regulatory Commission" which is one the most frequent topic of the light green community. Again these topics were not used to find the communities, but used here for validation purpose only.

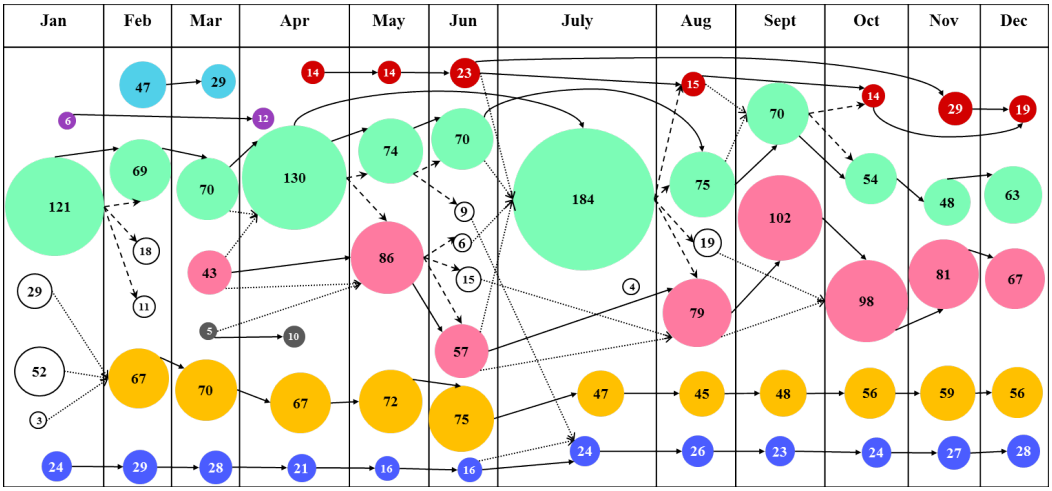


Figure 5.2: Events detected by the MODEC framework on Enron email dataset. Solid, dashed, and dotted arrows show detected *survive*, *split*, and *merge*.

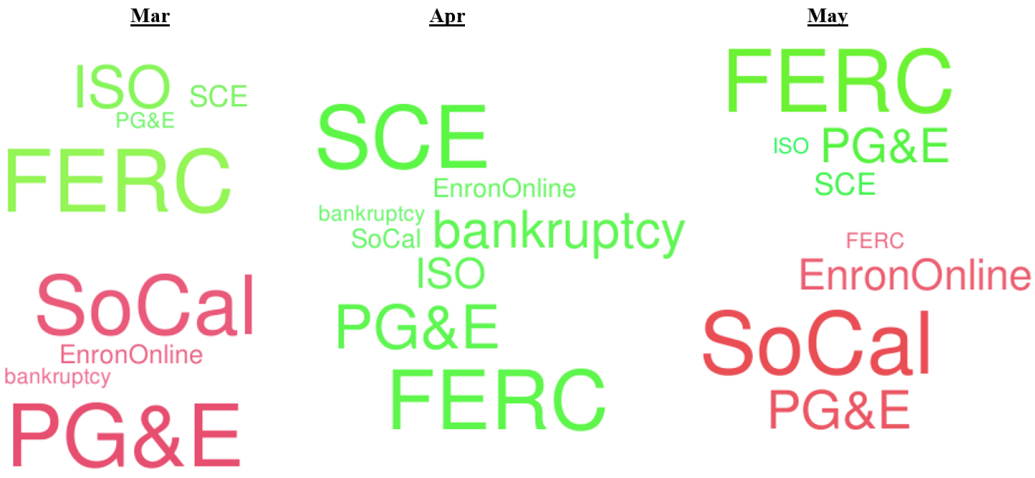


Figure 5.3: Contextual attributes relationship to events on Enron email dataset.

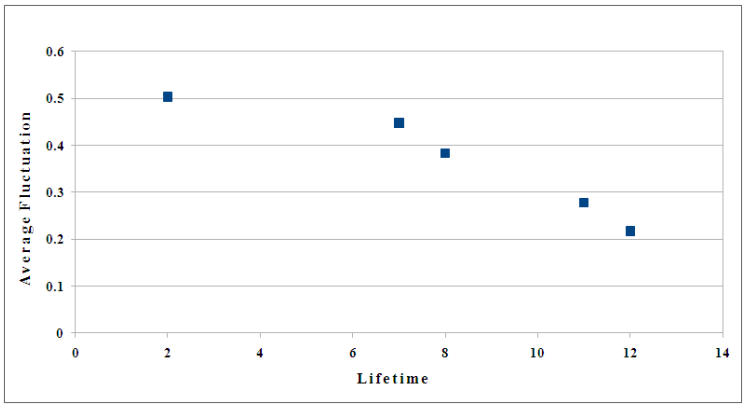


Figure 5.4: Average members fluctuation and lifetime of Enron meta community.

As stated previously, member fluctuation of a community is another metrics to study the temporal characteristics of a meta community. We observe an interesting effect when investigating the relationship between the lifetime and the members fluctuation of meta community. In Figure 5.4 the average lifetime as a function of the members fluctuation is depicted for Enron dataset with similarity threshold $k = 0.5$. The figure indicates that meta communities with average low fluctuating members usually tend to live longer, while meta communities with higher fluctuation *dissolve* sooner.

The comparison of MODEC framework with the other frameworks with $k = 0.5$ is shown in Table 5.2, where the total number of events detected by each framework during the 12 snapshots is provided. Applying Asur et al. framework [8], only a few *merge*, *split*, *form*, and *dissolve* are captured. This framework could not detect any *survive* events due to its restricted definition of these events and also because it only considers events between two consecutive snapshots. Palla et al. framework [85] defines events based on the concept of matching communities across time. However, the framework cannot find matches for many communities, thus, no events are detected for these communities. Greene et al. framework [45] cannot discover most of the *merge* and *split* events occurring in the observation time. Also, some of the *survive* events are not detected by this framework which leads to a higher number of *form* and *dissolve* than ours. We again incorporate topics extraction for each community to find out which framework results in the most appropriate community evolutions for the Enron dataset. The average mutual topics between any

Table 5.2: Comparison of different frameworks on the Enron dataset.

Framework	Form	Dissolve	Survive	Split	Merge	Mutual Topics
MODEC	19	19	46	7	11	3.83/10
Asur et al. [8]	6	6	0	7	8	0/10
Greene et al. [45]	25	24	39	0	1	2.74/10
Palla et al. [85]	13	13	20	12	16	2.0/10

two survival communities during the observation time is calculated for each framework (Table 5.2). Our results show that the highest mutual topics out of the top 10 most frequent keywords is found when using the MODEC framework. Thus our framework results in the most meaningful community evolution for Enron.

5.3.2 DBLP Co-authorship Dataset

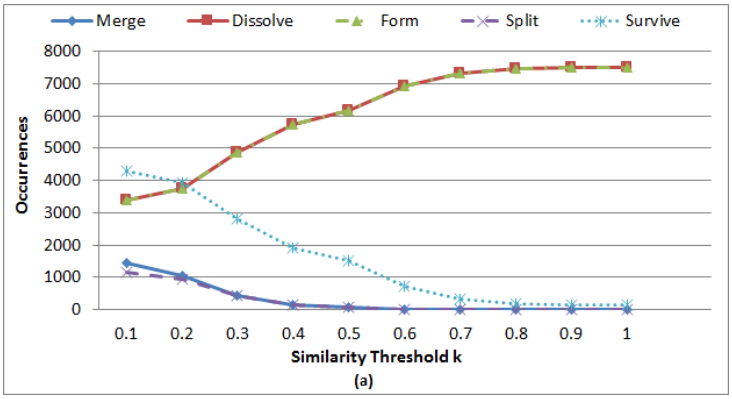
The impact of similarity threshold k on the community evolution is also investigated for the DBLP dataset (Figures 5.5a, 5.5b, and 5.5c). The similarity threshold is varied from 0.1 to 1.0 in steps of 0.1 and the number of events occurring during the 10 snapshots is counted for each k step. We again observe that the choice of k has a noticeable effect on the detected events: the number of *survival*, *merge*, and *split* events drops as k increases, while there are more *dissolution* and *formation*. Note that in our framework, the number of *formation* and *dissolution* are both equal to the number of discovered meta communities, thus, they are exactly the same in Figure 5.5a. With low values of k, more communities are matched together, thus, we can observe a significant number of *surviving* communities. However, high values of k result in a conservative matching behaviour and short-life meta communities. The number of different transitions is also compared on different variations of k (Figure 5.5b). As expected, since the transitions are defined for the survival communities, their numbers would drop with the increase of k. To better compare the number of transitions, Figure 5.5c depicts the normalized number of transitions over number of *survival* communities. Again we observed that the number of detected transitions gradually decreases as k increases.

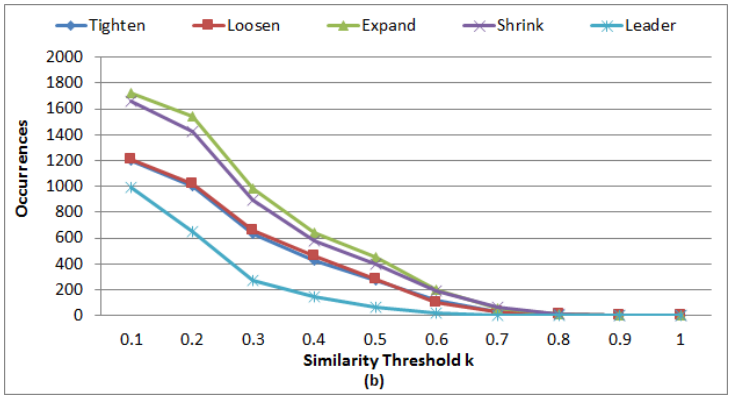
One important factor influencing the evolution of the communities is the semantic nature of the interaction itself. In DBLP, two authors are connected if they publish a paper together. The topic or the subject area of the paper will definitely

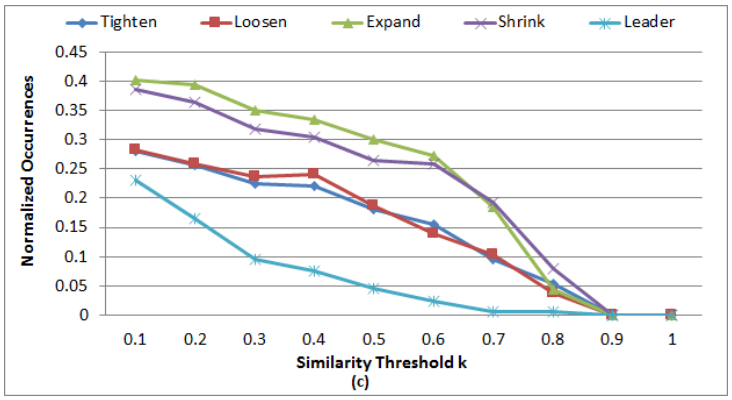
influence future collaborations for each of these authors. If there are different authors working on similar topics, the chances of them collaborating in the future is higher than two authors working on unrelated areas. In order to investigate the relationship between the semantic of the interaction and the communities evolution, we extract the topics of the papers (from their title and abstract) published within communities. We expect that a community which *survives* multiple timeframes is more likely to continue discussions of the same topics. Figure 5.5d shows the average mutual topics between any two *survival* communities during the observation time for different k. The survival communities mostly discuss the same topics, thus, the k that corresponds to the highest mutual topics illustrates the community evolution better than the others. Figure 5.5d shows that as k increases from 0.1 to 0.4 the average number of mutual topics also increases. However, for $k > 0.4$ the number of mutual topics decreases sharply. Based on this observation, we can conclude that for the DBLP dataset $k = 0.4$ results in a more meaningful evolution due to semantic of the interactions.

The difference between the optimal similarity threshold of the DBLP and the Enron dataset is due to their different structure: the Enron email dataset has rather stable communities with a considerable amount of members who participate over a long time and a small amount of fluctuating members. Thus, a high similarity threshold ($k = 0.5$) is required. On the other hand, in the DBLP co-authorship network, communities can be highly dynamic where members leave gradually, while new ones join. Hence, a rather low similarity threshold ($k = 0.4$) is used to analyze the evolution of communities in this network.

With $k = 0.4$, Figure 5.6 depicts an example of meta community detected in DBLP dataset that exists from year 2007 to year 2010. Community (a) at year 2007 with 11 members is first marked by *form*, since the bipartite matching algorithm could not find a match for it in previous years. This community *survives* and *loosen* to community (b) at 2008 with 5 mutual members (i.e. their similarity is 0.45). Community (b) then *survives*, *expands*, and *tighten* to community (c) at 2009 that has 5 similar members. Community (c) *splits* to two communities (d) and (e) at year 2010. However, community (c) also *survives* to community (d) since







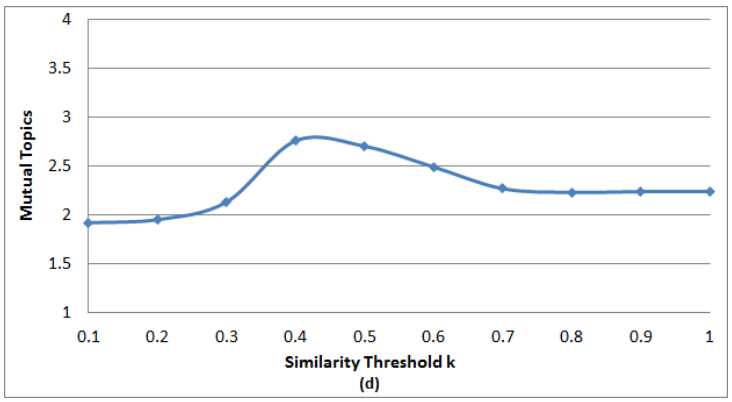


Figure 5.5: The impact of similarity threshold k on DBLP co-authorship dataset: (a) events; (b) transitions (c) normalized transitions by survival; (d) average mutual topics for different k.

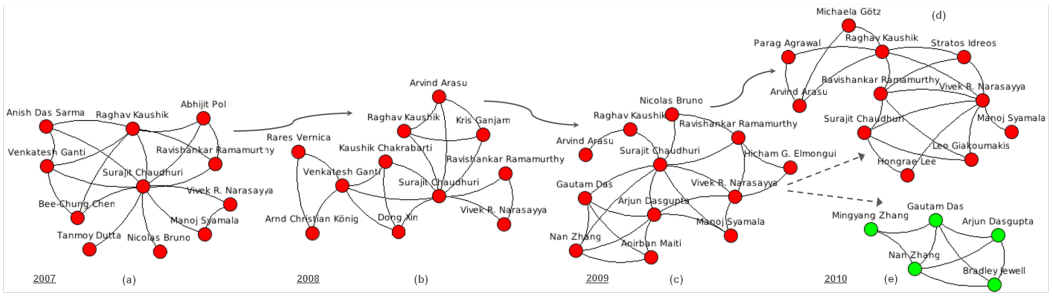


Figure 5.6: Example of detected events in the DBLP co-authorship dataset, where solid and dashed arrows indicate survive and split events respectively.

Table 5.3: Comparison of different frameworks on the example provided in Figure 5.6.

Event	MODEC	Asur et al. [8]	Greene et al. [45]	Palla et al. [85]
(a) forms	✓	—	✓	✓
(a) survives to (b)	✓	—	—	—
(b) survives to (c)	✓	—	—	—
(c) survives to (d)	✓	—	—	—
(c) splits to (d) and (e)	✓	—	—	—
(e) forms	✓	—	✓	✓
(d) and (e) dissolve	✓	✓	✓	✓

these two are community matches for each other. The *shrink* and *tighten* transitions are detected between communities (c) and (d). Also, we can observe the *leader shift* from 'Surajit Chaudhuri', who is the *community leaders* from 2007 to 2009, to 'Vivek R. Narasayya' in year 2010. In the next section, we will explain this leader transition in more detail. Table 5.3 provides the events detected by other frameworks for the communities that constitute this meta community. Asur et al. framework can only detect survival when the two communities are exactly the same in subsequent timeframes, and it can only detect split when half of the members of any resulting communities (at least two) are from the fractured community. Palla et al. and Greene et al. frameworks could not detect survivals and splits since they use Jaccard similarity which is not held for any of these three cases.

The comparison of MODEC framework, with the other frameworks with $k = 0.4$ on DBLP dataset is shown in Table 5.4 where the total number of events detected by each framework during the 10 years and the average mutual topics is provided. Asur et al. framework cannot detect any survive events due to its restricted definition of this event and also because it only considers events between two consecutive snapshots. Palla et al. framework defines events based on the concept of matching

Table 5.4: Comparison of different frameworks on the DBLP dataset.

Framework	Form	Dissolve	Survive	Split	Merge	Mutual Topics
MODEC	5748	5748	1918	149	145	2.76/10
Asur et al. [8]	2352	2361	178	87	93	1.23/10
Greene et al. [45]	6489	6433	1099	1	56	2.32/10
Palla et al. [85]	3456	3641	1260	681	704	1.94/10

communities between two consecutive snapshots. However, the framework cannot find matches for many communities, thus, no events are detected for these communities. Greene et al. framework cannot detect most of the merges and splits since it use Jaccard similarity which is not held for these cases. Our framework results in the highest average mutual topics, thus, provides the most meaningful community evolutions.

As shown in the experiments, the similarity threshold k chosen for Enron and DBLP dataset is different: similarity threshold $k = 0.5$ leads to the highest mutual topics for Enron, while similarity threshold $k = 0.4$ is the one with the highest mutual topics for DBLP. The difference between the chosen similarity threshold k for the DBLP and the Enron datasets can help us to find out the structural difference between the two datasets. The Enron email dataset has rather stable communities with a considerable amount of members who participate over a long time and a small amount of fluctuating members. This can be due to the fact that employees of a company mainly communicate with a rather stale group of colleagues that are working on similar projects. On the other hand, in the DBLP co-authorship network, communities can be highly dynamic where members leave gradually, while new ones join. This could be partly explained by a variety of external factors that can affect this fluctuation, for example meeting at a conference, moving between institutions, graduating of students after sometime, etc.

In this chapter, for the purpose of experimental studies we tuned the similarity threshold k for the MODEC framework and selected the optimal similarity threshold k for our framework based on the topic continuation. However, there are no parameters for the other three frameworks to tune, i.e. Asur, Greene, and Palla, use a fixed notion of similarity. This is due to the fact that the formal definitions of the events in these three frameworks do not depend on any similarity parameter.

Two communities are similar if they simply share more than half of their individuals. By coincidence, for Enron the optimal similarity threshold k that is chosen for MODEC framework is 0.5 which leads to the highest number of mutual topics compared with others. Furthermore, for DBLP the optimal similarity threshold k chosen for MODEC is 0.4 which is close to the fixed 0.5 used for other frameworks.

5.4 Summary

In this chapter, we present a framework for the monitoring of community transitions and evolutions over time. Our framework encompasses both a community matching algorithm and an event detection model that captures the critical events for communities. This includes tracking the formation, survival and dissolution of communities as well as identifying the meta communities, which are a series of similar communities at different snapshots. Applying our framework on the Enron email dataset, and DBLP co-authorship dataset, we uncover communities with different evolutionary characteristics and address the noticeable effect that the similarity threshold has on the evolution of communities. In order to validate the choice of the similarity threshold, we propose extracting and investigating frequently used topics for each community and selecting the appropriate threshold based on the continuation of these topics. The performance of our framework on both datasets is also compared with the other event-based frameworks. The results show that the MODEC framework outperforms the others in terms of the average mutual topics between survival communities.

Chapter 6

Temporal and Evolutionary Behaviour Analysis

A social role is a special position an individual possesses within a network, which indicates his or her behaviours, expectations, and responsibilities. Identifying the roles that individuals play in a social network has various direct applications, such as detecting influential members, trustworthy people, idea innovators, etc. Roles can also be used for further analyses of the network, e.g. community detection, temporal event prediction, and summarization.

In this chapter, we propose a framework to analyze the behaviour of individuals over time. We identify events and high-level roles related to individual, study their changes, and analyze their impacts on the underlying social network. Role changes are tracked over time and their correlation with the structural events in the network is illustrated.

6.1 Event Formulation

We define different events, metrics, and roles to analyze the behaviour of an individual within its communities. Four events involving individuals are defined here. Similar to the events defined for communities, these events can be considered between any two snapshots, not just the two consecutive ones. In the following, we provide the formal definitions of these events, where $M(C_i^p)$ indicates the meta community that constitutes community C_i^p.

Appear: A node v *appears* at i^{th} snapshot when it exists in any of the communi-

ties in the current snapshot but was not present in any community in the previous snapshots:

$$\text{appear}(v, i) = \text{true} \quad \text{iff} \quad v \in V_i \text{ and } \nexists j < i : v \in V_j \tag{6.1}$$

Based on the *appear* event, $birth(v)$ is defined as the snapshot in which node v appeared in (i.e. $\text{appear}(v, \text{birth}(v)) = \text{true}$).

Disappear: A node v *disappears* at i^{th} snapshot when it exists in any of the communities in the current snapshot but will not occur in any community in the next snapshots:

$$\text{disappear}(v, i) = \text{true} \quad \text{iff} \quad v \in V_i \text{ and } \nexists j > i : v \in V_j \tag{6.2}$$

Based on *disappear* event, $death(v)$ is defined as the snapshot in which node v disappeared in (i.e. $\text{disappear}(v, \text{death}(v)) = \text{true}$).

Join: A node v *joins* the community C_i^p at i^{th} snapshot if it exists in this community in the current snapshot but did not belong to a community with the same meta community as of C_i^p in the previous snapshots:

$$\begin{aligned} &\text{join}(v, C_i^p) = \text{true} \quad \text{iff} \quad v \in C_i^p \text{ and } \forall j < i \nexists C_j^q : \\ &v \in C_j^q \quad \text{and} \quad M(C_i^p) = M(C_j^q) \end{aligned} \tag{6.3}$$

Leave: A node v *leaves* community C_i^p at i^{th} snapshot if it exists in this community in the current snapshot but will not belong to a community with the same meta community as of C_i^p in the next snapshots:

$$\begin{aligned} &\text{leave}(v, C_i^p) = \text{true} \quad \text{iff} \quad v \in C_i^p \text{ and } \forall j > i \nexists C_j^q : \\ &v \in C_j^q \quad \text{and} \quad M(C_i^p) = M(C_j^q) \end{aligned} \tag{6.4}$$

Beside the above proposed events, we define different metrics to capture the behavioural characteristics of individuals. Before studying these metrics, let us first provide a definition. Regardless of the underlying community mining variation, in general, each node v of a network can be characterized by an affiliation set $\text{A}(v, i)$, which is the set of communities that the node belongs to at snapshot i. When having non-overlapping community, $\text{A}(v, i)$ represents one community, while it represents set of community in overlapping community structure scenarios. The *affiliation* of an individual is temporal and may change over the time:

Affiliation List: Given a set of snapshots 0, 2, ..., $n-1$, the affiliation list of node v, $\mathcal{A}(v) = \{\mathrm{A}(v, \mathrm{birth}(v)), ..., \mathrm{A}(v, i), ..., \mathrm{A}(v, \mathrm{death}(v))\}$ is denoted by the communities that contain node v, ordered by their time steps. Note that node v may be absent at some time steps during the observation, thus $(death(v) - birth(v) + 1) \leq n$.

In the case of an overlapping community structure, overlapped nodes tend to be active users who participate in multiple communities at the same time. Thus, changes in the number of communities a node is a member of can be a good indicator on the level of involvement of a user. Beside the size of the assignment list, the involvement of a node in its communities is also an important factor either in overlapping or non-overlapping community structure. Thus, we define the involvement metric as follows:

Involvement Metric: Given a snapshots i, and the affiliation of node v, $\mathrm{A}(v, i)$, the involvement of node v at i^{th} snapshot is

$$\mathrm{involvement}(v, i) = \frac{\sum\limits_{\substack{C_i^p \in \\ \mathrm{A}(v,i)}} \mathrm{centrality}(v, C_i^p)}{|\,\mathrm{A}(v, i)|} \tag{6.5}$$

Here, any centrality measure including *degree centrality*, *closeness centrality*, and *betweenness centrality* can be used to calculate $\mathrm{centrality}(v, C_i^p)$. The involvement metric of node v over the observation time is then the average involvement of v in its affiliation time.

Another important metric in analyzing dynamic social networks is the influence of the individuals on each other. For example, if one node influences others to *join* (*leave*) a community then it may be a very influential node in the network. A high influence score indicates that when the node *joins* (*leaves*) a community, a large number of follower nodes will also *join* (*leave*) that community. The number of nodes *join* (*leave*) a community C_i^p at snapshot i is

$$\begin{aligned} |\,\mathrm{join}(C_i^p, i)| &= \sum_{\forall u \in C_i^p} \mathrm{join}(u, C_i^p) \\ |\,\mathrm{leave}(C_i^p, i)| &= \sum_{\forall u \in C_i^p} \mathrm{leave}(u, C_i^p) \end{aligned} \tag{6.6}$$

Influence Metric: Given a snapshots i, and the affiliation of node v, $\mathrm{A}(v,i)$, the join (leave) influence of node v at i^{th} snapshot is

$$\begin{aligned}
\mathrm{joinInfluence}(v,i) &= \frac{\sum\limits_{\substack{C_i^p \in \\ \mathrm{A}(v,i)}} \mathrm{joinInfluence}(v, C_i^p)}{|\,\mathrm{A}(v,i)|} \\
\mathrm{joinInfluence}(v, C_i^p) &= \frac{\mathrm{join}(v, C_i^p)|\,\mathrm{join}(C_i^p, i)|}{|V_i^p|}\,\mathrm{centrality}(v,i) \\
\mathrm{leaveInfluence}(v,i) &= \frac{\sum\limits_{\substack{C_i^p \in \\ \mathrm{A}(v,i)}} \mathrm{leaveInfluence}(v, C_i^p)}{|\,\mathrm{A}(v,i)|} \\
\mathrm{leaveInfluence}(v, C_i^p) &= \frac{\mathrm{join}(v, C_i^p)|\,\mathrm{leave}(C_i^p, i)|}{|V_i^p|}\,\mathrm{centrality}(v,i)
\end{aligned} \tag{6.7}$$

Again, we propose influence metric in a way to support both the overlapping and the non-overlapping case. Furthermore, we consider centrality of the nodes in calculating their influence to exclude nodes that join (leave) one community along with a node with high influence. The join (leave) influence metric of node v over the observation time is then the average join (leave) influence of v in its affiliation time.

The *involvement*, *join influence*, *leave influence*, and the size of affiliation are all temporal metrics, meaning that their values for a specific node may change over time. Thus, we study the temporal behaviour of a user by finding patterns in the trends of these metrics through time.

Individual Temporal Behaviour: Given a set of snapshots 0, 2, ..., $n-1$, and the affiliation list of node v, $\mathcal{A}(v) = \{\mathrm{A}(v, \mathrm{birth}(v)), ..., \mathrm{A}(v,i), ..., \mathrm{A}(v, \mathrm{death}(v))\}$, the temporal behaviour of the node v can be characterized as follows:

$$\begin{aligned}
&|\,\mathcal{A}(v)| = [|\,\mathrm{A}(v, \mathrm{birth}(v))|, ..., |\,\mathrm{A}(v,i)|, ..., |\,\mathrm{A}(v, \mathrm{death}(v))|] \\
&\textit{involvement}(v) = \\
&\quad [\mathrm{involvement}(v, \mathrm{birth}(v)), ..., \mathrm{involvement}(v,i), ..., \mathrm{involvement}(v, \mathrm{death}(v))] \\
&\textit{joinInfluence}(v) = \\
&\quad [\mathrm{joinInfluence}(v, \mathrm{birth}(v)), ..., \mathrm{joinInfluence}(v,i), ..., \mathrm{joinInfluence}(v, \mathrm{death}(v))] \\
&\textit{leaveInfluence}(v) = \\
&\quad [\mathrm{leaveInfluence}(v, \mathrm{birth}(v)), ..., \mathrm{leaveInfluence}(v,i), ..., \mathrm{leaveInfluence}(v, \mathrm{death}(v))]
\end{aligned} \tag{6.8}$$

Another metric to capture the behaviour of the user is the stability metric, which calculates the tendency of an individual to interact with the same nodes over the observation time. For any individual, this metric is the average of the similarity between two consecutive communities that contain that individual. Thus, a high stability score is indicative of a stable individual who mainly interacts with the same people over time, whereas a low stability score shows a rather unstable node. In the following, the formal definition of the stability metric is provided:

Stability Metric: Given a set of snapshots 0, 2, ..., $n-1$, and the affiliation list of node v, $\mathcal{A}(v) = \{\mathrm{A}(v, \mathrm{birth}(v)), ..., \mathrm{A}(v,i), ..., \mathrm{A}(v, \mathrm{death}(v))\}$, the stability metrics of node v is then defined as

$$\mathrm{stability}(v) = \frac{\sum\limits_{\substack{\mathrm{A}(v,i),\mathrm{A}(v,j) \\ \in \mathcal{A}(v)}} \mathrm{sim}(\bigcup\limits_{\substack{C_i^p \in \\ \mathrm{A}(v,i)}} C_i^p, \bigcup\limits_{\substack{C_i^p \in \\ \mathrm{A}(v,j)}} C_i^p)}{|\mathcal{A}(v)|} \tag{6.9}$$

Note that in the case of overlapping community structure, the $\mathrm{A}(v, t_i)$ represent a set of communities rather than a single community. Thus, the proposed stability metric considers the general case.

It is worth mentioning that the proposed *stability metric* is an average metric over time, rather than a list of temporal metrics such as *influence* and *involvement* metrics.

The events, metrics and roles defined for an individual are summarized in Table 6.1.

6.2 Role Formulation

In Chapter 3, we define four fundamental roles namely leader, outermost, mediator, and outsiders. We now describe how they can be identified in a given network. Having the communities, we identify roles either directly based on the community memberships (outsiders), or based on a ranking of nodes within the communities (leaders and outermosts), or the whole network (mediators). For the ranking based roles (i.e. leader and outermost), the distribution of the centrality scores for nodes is used to automatically identify the roles. The identification of the mediator role,

Table 6.1: Behavioural and Role Analysis: Definition of symbols

Symbol	Definition
Individual Events	
$\mathrm{appear}(v, i)$	appear event for individual v at snapshot i (Equation 6.1)
$\mathrm{disappear}(v, i)$	disappear event for individual v at snapshot i (Equation 6.2)
$\mathrm{join}(v, C_i^p)$	join event for individual v to community C_i^p at snapshot i (Equation 6.3)
$\mathrm{leave}(v, C_i^p)$	leave event for individual v from community C_i^p at snapshot i (Equation 6.4)
Individual Properties	
$\mathrm{birth}(v)$	the snapshot where node v is appeared in
$\mathrm{death}(v)$	the snapshot where node v is disappeared in
$\mathrm{A}(v, i)$	affiliation set, i.e. set of communities that the node belongs to at snapshot i
$\mathcal{A}(v)$	affiliations list of node v, i.e. its affiliation ordered by their time steps
$\mathrm{centrality}(v, i)$	centrality of node v at snapshot i
$\mathrm{centrality}(v, C_i^p)$	centrality of node v in community C_i^p at snapshot i
Behavioural Metrics	
$\mathrm{joinInfluence}(v, i)$	join influence metric of node v at snapshot i (Equation 6.7)
$\mathrm{leaveInfluence}(v, i)$	leave influence metric of node v at snapshot i (Equation 6.7)
$\mathrm{involvement}(v, i)$	involvement metric of node v at snapshot i (Equation 6.5)
Temporal Behavioural Metrics	
$\lvert\,\mathcal{A}(v)\rvert$	affiliation size of node v over time (Equation 6.8)
$\mathit{involvement}(v)$	involvement metric of node v over time (Equation 6.8)
$\mathit{joinInfluence}(v)$	join influence of node v over time (Equation 6.8)
$\mathit{leaveInfluence}(v)$	leave influence of node v over time (Equation 6.8)
$\mathrm{stability}(v)$	stability metric of node v over the observation time (Equation 6.9)

however, requires a more complicated procedure. For the purpose of this dissertation, we only require the two ranking nodes. Thus, in the following we formally identify these two roles. Readers can refer to [1] for more complete algorithms on how to detect the other roles.

Leader: Leader members are identified in association with each community. First, an appropriate importance/centrality measure ($\mathrm{centrality}$) is used to score the members of the community (one might apply any of the commonly used centrality measures, or any other analysis that provides a ranking for importance of nodes). Then, the probability distribution function (pdf) for the centrality scores in that community, e.g. $\mathrm{centrality}(v, C_i^p)$, is estimated. Analyzing the characteristics of this pdf determines the leaders. More specifically, nodes falling in the upper tail of the distribution are identified as the community leaders. Our experiment results previously show that the pdf of closeness centrality scores for all the nodes in each community is close to a normal distribution [1]. According to the properties of a normal distribution, almost 95% of the population lies in the interval $[\mu - 2\sigma, \mu + 2\sigma]$, where μ and σ are mean and standard deviation of this distribution. Thus, we use the upper threshold of $\mu + 2\sigma$ to distinguish the leaders of a community.

Outermost: Outermost members are identified in contrast to the leaders, i.e. as members of a community falling in the lower tail of the importance pdf[1]. Similar to leader detection, the pdf of closeness centrality scores for all the nodes in each community is computed. Then, the lower threshold $\mu - 2\sigma$ is used for outermost.

6.3 Event Triggers

Events involving communities usually indicate structural change in a network, except the survive event, which can occur with no accompanying changes such as joining and leaving members. For the remaining community events (merge, split, form,

[1] One should note that identifying outermosts is challenging using centrality measures, since the intuition behind centrality measures is to identify higher values, not lower ones. This means, high centrality scores for nodes infer their importance, however, low centrality scores do not necessarily mean that they are not important. Thus, in general, centrality measures are efficient for identifying more central nodes, but not necessarily least central ones.

dissolve), significant *structural* changes are almost always indicated. As stated before, the events, metrics, and roles associated with individuals are dependent upon *structural* properties of the network and their enclosing communities. Therefore, studying the effect of individual events and role changing behaviour on community events can lead us to interesting reasoning.

When individual events occur (join, leave, appear, disappear), the likelihood of an impact upon individual roles is less likely, or at least dependent upon there being a large number of individual events. Thus, we focus on individual events occurring for leaders of a community. Some clear examples of the effects of events upon role changes may be described. A community leader may leave one community, and afterwards there might not be an authority for that community, hence leads to dissolution of the community. The community leader who left his/her community to join another community, might or might not have a leader role in the new community: the changes that led to them leaving their original community might also have caused them to lose connectivity to the network. The same sort of considerations apply to merge and split events, where the community context has changed an old leader now has competition from others, or has lost connections vital to their leader role. In our experiments, we provide examples on two real-world datasets which show the impact of individual events and roles on community events.

For any given role change and individual event, we can identify the community events involved. Our current method of role change, and individual event attribution consists of associating each significant role change with all events involving the communities that the individual was involved with before and after the role change. This will not capture domain dependent attributions, that is, attributions that require domain knowledge. For example, a change in leader may be associated with a merge of the leader's community with two other communities, resulting in a loss of leader role for that individual. There might also have been splits and joins elsewhere in the network, not involving this individual or those communities. If the domain contained causal connections between domain specific events and the change in role, and if those domain specific events had no clean relation to the generic network events, the causal connection would go undetected in the network analysis. This is

a domain dependent risk, and it is not clear that a generic analysis can ever deal with all such possibilities.

Beside analyzing the current state of a community with different events and also studying the impact of different individuals events on the community itself, all the proposed events, metrics, and roles can also help to predict the next state of a community. In the next chapter, we propose a predictive model to anticipate the structure of the community in a near future.

6.4 Experiments

In this section, we apply different techniques proposed in previous section on our two real datasets to track the dynamic behaviours of individuals and their influence on their communities over time.

6.4.1 Enron Email Dataset

For the Enron dataset, we identify *leader* and *outermost* roles for the detected communities through time. The detected events and transitions are later used in the analysis to observe the mutual effects between the changes in the extracted roles and the community events. We expect that the changes in the individuals role, play an important factor in the evolution of their corresponding communities.

Intuitively, degree and closeness centrality scores seem to be good candidates for ranking individuals in a community in order to identify leaders and outermosts. The degree centrality, and closeness centrality of community light green (size 75) in August (Figure 5.2) is shown in Figures 6.1a, and 6.1b respectively as an examples. As shown in Figure 6.1a, degree distributions have mostly one tail (the upper tail). Although the long upper tail of degree distributions can be used to identify leaders, outermosts cannot be simply identified using degree distributions of communities. Thus, we consider closeness centrality rather than degree centrality that follows a normal-like distribution (Figure 6.1b); the two tails in each community can be effectively used to identify both leaders and outermosts. Using properties of the normal distribution, we set $\mu + 2\sigma$ as the upper and $\mu - 2\sigma$ as the lower thresholds

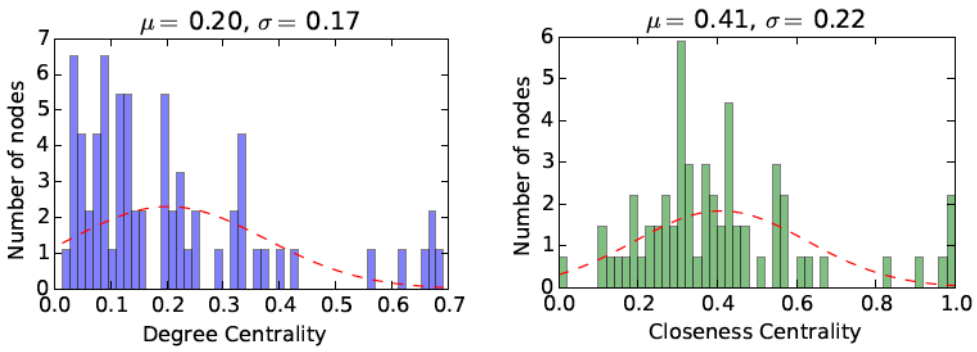


Figure 6.1: Centrality scores distribution of light green community (size 75) in August 2001 on Enron Email dataset: (a) degree distribution; (b) closeness distribution.

to identify leaders and outermosts respectively.

Leaders that are identified for the light green community (Figure 5.2) over the observation time are provided in Table 6.2. Being a leader in the network translates into having a high average of short email distance to other individuals in the community. Table 6.2 provides us with interesting information about leader, i.e. some nodes are constantly leaders in early timeframes while some others are constantly leaders in later timeframes. Also, the importance of the leadership of leader nodes change over time. Intuitively, being a constant leader over more timeframes could mean that a node is more important and influential in its community. Among these names, James Steffes who was the vice president of governmental affair is constantly one of the leaders in that community from January until June, while Louise Kitchen who was EnronOnline's president became a constant leader right after James Steffes left the network.

Tracking role changes in aggregation with community events in the Enron email network provides interesting result on the impact of role changes on the community changes. Steven Kean, Enron executive vice president and chief of staff, is a leader in light green community in January and July. Coincidentally, the light green community *shrinks* whenever leadership transfer from Steven Kean to someone else. Interestingly, light green community faces a *merge* in the beginning of July with another large community when Steven Kean takes back the leadership. Furthermore, the light green community splits in August when Steven Kean is no

Snapshot	Email	Position
Jan	steven.kean@enron.com	Executive Vice President and Chief of Staff
	james.steffes@enron.com	Vice President of Governmental Affairs
Feb	jeff.dasovich@enron.com	Executive/Director for State Government Affairs
	james.steffes@enron.com	Vice President of Governmental Affairs
Mar	james.steffes@enron.com	Vice President of Governmental Affairs
Apr	jeff.dasovich@enron.com	Executive/Director for State Government Affairs
	james.steffes@enron.com	Vice President of Governmental Affairs
May	jeff.dasovich@enron.com	Executive/Director for State Government Affairs
	james.steffes@enron.com	Vice President of Governmental Affairs
Jun	jeff.dasovich@enron.com	Executive/Director for State Government Affairs
	james.steffes@enron.com	Vice President of Governmental Affairs
Jul	steven.kean@enron.com	Executive Vice President and Chief of Staff
	jeff.dasovich@enron.com	Executive/Director for State Government Affairs
	richard.shapiro@enron.com	senior vice president of the Enron Corporation
Aug	jeff.dasovich@enron.com	Executive/Director for State Government Affairs
	richard.shapiro@enron.com	Senior vice president of the Enron Corporation
Sep	jeff.dasovich@enron.com	Executive/Director for State Government Affairs
	louise.kitchen@enron.com	EnronOnline's president
	richard.shapiro@enron.com	senior vice president of the Enron Corporation
Oct	louise.kitchen@enron.com	EnronOnline's president
Nov	louise.kitchen@enron.com	EnronOnline's president
Dec	louise.kitchen@enron.com	EnronOnline's president

Table 6.2: Leaders of the light green community, their community affiliation, and position in the Enron organization

longer among leaders. Note that our focus is on light green communities, however, similar relationship between events and leadership can be observed for other communities. This anecdote raises an interesting question on how many of the *merge* or *split* events are due to the *leader shift* transition. On the Enron dataset, one *merge* occurs after the *leader shifts*. Hence, in this network, most of the *leader* transitions are due to changes of interaction within the community rather than being a byproduct of these relatively rare *merge* and *split* events.

6.4.2 DBLP Co-authorship Dataset

In the meta community shown in Figure 5.6, we detect the *leader shifts* from 'Surajit Chaudhuri', who is the *Community leaders* from 2007 to 2009, to 'Vivek R. Narasayya' in year 2010. Table 6.3 provides more details on changing the closeness centrality score of these two individuals during the existence of the meta com-

Table 6.3: The degree centrality scores of two individuals on the meta community depicted in Figure 5.6.

Individual	2007	2008	2009	2010
Surajit Chaudhuri	1.0	0.8	0.72	0.4
Vivek R. Narasayya	0.2	0.2	0.45	0.7

Table 6.4: Nodal behaviour analysis on the Enron and DBLP datasets.

Dataset	Stability	Join Influence	Leave Influence
Enron	0.5053	0.0288	0.0266
DBLP	0.3403	0.0446	0.0489

munity. The centrality scores shows that 'Vivek R. Narasayya', at the beginning is among nodes with low centrality scores, but as times goes by he develops his relationship and at year 2010 becomes *community leader* within his community. On the other hand, the centrality score of 'Surajit Chaudhuri' decreases over the years, and finally at year 2010 he loses the *community leadership*. Furthermore, at the same year that the *leader shifts*, community (c) *splits* to two communities (d) and (e). On the DBLP dataset *leader shift* transition is the cause of 6 *merge* and 5 *splits* events.

The previously defined nodal behaviour metrics also reveal information on the characters of the network. Table 6.4 provides the average stability, join and leave metrics (over all the individuals) for the Enron and DBLP datasets. The relatively high stability for the Enron dataset and the relatively low stability for the DBLP dataset suggests that individuals in the Enron dataset have a higher tendency to interact with the same nodes over a long period; whereas the authors in the DBLP dataset will rarely express this behaviour. However, the influence an individual has over who *joins* or *leaves* a community in the DBLP dataset is much greater than in the Enron dataset. Table 6.5 provides the top 5 influential authors in DBLP based on their average influence. Except for 'Ravi Kumar', the high influential authors have higher leave influence than join influence.

6.5 Summary

In this chapter, we define events and roles that associated to individuals moving between the communities in a dynamic social network. We explain the relationship

Table 6.5: Top 5 influential authors in the DBLP dataset.

Author	Join influence	Leave influence	Average influence
Ravi Kumar	0.5668	0.5250	0.5459
Serge Abiteboul	0.4252	0.5605	0.4929
Jiawei Han	0.4360	0.5497	0.4928
Elisa Bertino	0.4124	0.5450	0.4787
Philip S. Yu	0.4009	0.4998	0.4503

between the evolution of communities, the movement of individuals between these communities and changes in the role of those individuals. We further illustrate that changes in the role of individuals in a community have a direct relationship with the development of the community. The role change can act as triggers to evoke community changes. Through our visualizations, we demonstrate that role modification can affect the dynamics of communities and the events in the communities can alter the role of individuals.

Chapter 7

Community Prediction using Supervised Learning

Finding patterns of interaction and predicting the future structure of networks has many important applications, such as recommendation systems and customer targeting. From a mesoscopic point of view, community structure of social networks may undergo different temporal events and transitions. The knowledge of the community structure and the behaviour of its individuals over time enables the prediction of the essential features of the network under investigation, and can help make better decisions regarding that network. For example, we can provide a tool to interpret the communities of large networks and can predict how the community structure of the network changes in different circumstances.

In most of real-world and complex graphs, entities, their connections, and communities may have different structural properties and attributes. Furthermore, temporal aspects, such as the difference between the community size between two snapshots, may also have an effect on the community evolution in next timeframes. In this chapter, we propose a machine learning model to accurately predict the occurrence of different events and transition for communities in dynamic social networks. Our model incorporates key features related to a community – its structure, history, and influential members, and automatically detects the most predictive features for each event and transition.

In this model, we leverage the relation between the behavior of individuals and the future of their communities. Members of a community play an important role

in attracting new members and generally shaping the future of their community. We further assume that individuals who are more likely to undertake actions in their communities, are more influential in the future trend of their community, and therefore are principal factors in the predictive process. Thus, in this work we not only consider the properties related to the communities but also the properties related to the influential members of the community as the input of the machine learning model.

Moreover, unlike previous works that only consider one aspect of the communities (i.e. size, age, or event), our models provide a complete predictive process for any transition and event that a community may undergo, and at the same time, identify the most prominent features for each community transition and event. The last important distinction of our model is that our events and transitions do not have to taken place in consecutive snapshots. A community may not necessarily be observed at consecutive snapshots, while it may be missing from one or more intermediate steps. Hence, our model predicts the next stage of a community either in the exact next snapshot or any later snapshot.

7.1 Feature Selection

To predict the next stage of a community, we consider five main classes of features: properties of its influential members, properties of the community itself, temporal changes of these properties, previously detected events and transitions, and contextual properties. These features are summarized in Table 7.1, and are explained in detail in the following.

PROPERTIES OF ITS INFLUENTIAL MEMBERS. The evolution of a network is usually analyzed by considering all members and their properties. However, communities are often led by a smaller set of individuals, who have considerable influence over other members, and shape the fate of their community. To identify the influential nodes in a community we use the role *leader* defined previously. The *leaders* of a community are defined as the outstanding individuals in terms of centrality or importance in that community. For these detected

leaders, we consider two structural features, i.e. their degree and closeness centrality scores. Since a community may have more than one leader, we take the average degree and closeness centrality scores of the detected leaders. We also consider the ratio of the leaders to the community size as a separate feature for the community. Similarly, we consider the ratio of outermosts in a community as another feature; where *Outermosts* are defined as the small set of least significant individuals in the community.

STRUCTURAL PROPERTIES OF THE COMMUNITY ITSELF. To quantify the structural properties of a community we consider its size (number of nodes), cohesion, density, and clustering coefficient. Similar to the clustering coefficient, we also consider the average and variance of the centrality scores of all members as separate features.

TEMPORAL CHANGES OF FEATURES. We consider the current rate of change in each property of a community as an additional feature. More specifically, the difference between properties of community C_i^p and properties of its previous instance, i.e. C_j^q, are considered as features.

PREVIOUSLY DETECTED EVENTS. For community C_i^p, we also consider the events and transitions that occurred for its previous instance, i.e. C_j^q, since there could be an auto-correlation.

CONTEXTUAL ATTRIBUTES AS FEATURES. For the datasets containing text, we consider two more features: *stable topics*, and *stable topics of leaders*. As explained previously, we detect topics with the most frequent keywords discussed in a community. We expect that the changes in the topics discussed within a community or by its influential members affect its future.

Other than these community-event-individual associations used as features, it might be possible to give deeper attributions of events upon individual behavioural changes. This would require a much more in depth analysis framework, and would involve questions of ambiguous causality at the generic network level. If tentative

Table 7.1: Problem Formulation: Features and response variables related to a community

Category	**Feature**	**Domain**
Influential Member	ClosenessLeaders	$(0, 1]$
	DegreeLeaders	$(0, 1]$
	LeadersRatio	$(0, 1]$
	OutermostRatio	$[0, 1]$
Community	Density	$(0, 1]$
	ClusteringCoefficient	$(0, 1]$
	NodesNumber	$[2, \infty)$
	Cohesion	$(0, \infty)$
	AverageCloseness	$(0, 1]$
	VarianceCloseness	$[0, 1]$
	AverageDegree	$(0, 1]$
	VarianceDegree	$[0, 1]$
Temporal	ΔClosenessLeaders	$(0, 1]$
	ΔDegreeLeaders	$(0, 1]$
	ΔLeadersRatio	$[0, 1]$
	ΔOutermostsRatio	$[0, 1]$
	ΔDensity	$[0, 1]$
	ΔClusteringCoefficient	$[0, 1]$
	ΔAverageCloseness	$[0, 1]$
	ΔVarianceCloseness	$[0, 1]$
	ΔAverageDegree	$[0, 1]$
	ΔVarianceDegree	$[0, 1]$
	JoinNodesRatio	$[0, 1]$
	LeftNodesRatio	$[0, 1]$
	Similarity	$[k, 1]$
	LifeSpan	$[1, n]$
Previous Events	PreviousSurvive	$\{\text{true}, \text{false}\}$
	PreviousMerge	$\{\text{true}, \text{false}\}$
	PreviousSplit	$\{\text{true}, \text{false}\}$
	PreviousSizeTransition	$\{\text{expand}, \text{shrink}\}$
	PreviousCohesionTransition	$\{\text{tighten}, \text{loosen}\}$
Contextual	StableTopics	$\{\text{true}, \text{false}\}$
	StableLeaderTopics	$\{\text{true}, \text{false}\}$
Response variable	survive	$\{\text{true}, \text{false}\}$
	merge	$\{\text{true}, \text{false}\}$
	split	$\{\text{true}, \text{false}\}$
	size	$\{\text{expand}, \text{shrink}\}$
	cohesion	$\{\text{tighten}, \text{loosen}\}$

causal links are identified, they could then be associated with real domain events that are not represented in our MODEC framework. For example, if an individual changes from being a community leader to being a leader in another community following a leave/join event, there might be a discrete real-world event corresponding with this change, such as the individual being promoted from being director of one department to being a director of another department. It is our position that to understand network dynamics at the domain level, it is necessary to have the generic, quantitative backing from dynamic social network, including both community event analysis as well as individual behavioural analysis. The generic network methods are intended to offer evidence and act as a modelling lens for domain specific hypotheses and descriptions. These two sources of knowledge may be compared and contrasted with the combination of theoretical analysis and statistical modelling found in experimental scientific disciplines. Without network analytic evidence to support domain hypotheses or measure domain events, full understanding cannot occur.

7.2 Predictive Model

Based on our problem formulation, the prediction of our events and transitions becomes a machine learning task, for which we use logistic regression and different classification methods. Here, we first train our model based on the previous events and properties of other communities and individuals in the social network. Then, the assumption is that the previous events occurred for a given community along with its properties and the properties of its members in the past as the input for the model, can predict the next probable changes for that community.

The only exception is that the size and cohesion transitions only occur for a community that survives. Thus, in order to predict these two response variables ($\text{size}\{\text{expand}, \text{shrink}\}$, and $\text{cohesion}\{\text{tighten}, \text{loosen}\}$), we propose a two-stage cascade predictive model, where the information collected from the output of a first stage is used as additional information for the second stage in the cascade. The first stage predicts the survive response variable ($\text{survive}\{\text{true}, \text{false}\}$), then only

in a case of true predicted value, the size and cohesion transitions should be predicted. The procedure to predict the cohesion, and size transition can be summed up as follows:

Two-stage cascade predictive model:

1: Predict the survival of a community using the $survive$ predictive model. If the predicted value for $\mathrm{survive}$ is true, go to Stage 2, otherwise, community does not have any transitions.

2: Predict the size and cohesion transitions using their respective predictive models.

Note that the predictive models in the two above stages could be different.

Each of the five different response variables is defined as a binary categorical variable. Thus, we adopt a logistic regression for each response variable using the features in Table 7.1 as the predictors. Then, in order to select the most significant feature set, we apply forward stepwise additive regression [59], where LogitBoost with simple regression functions is used for fitting the logistic models, and attribute selection.

Beside the logistic regression, we also adopt the most well-known binary classifier methods to predict each response variable: Naïve Bayes classifier, Bagging classifier, Decision Table classifier, Decision Stump classifier, J48 Decision tree, Bayesian Networks classifier, Simple CART classifier, Support Vector Machine (SVM) classifier, and Neural network classifier[1](Table 7.2).

Using all the features provided in Table 7.1 may not lead to the highest accuracy due to over-fitting, and redundant or irrelevant features. Therefore, we apply a wrapper method to select the appropriate feature sets for each binary classifier. The wrapper method uses a classifier to estimate the score of different features based on the error rate of that classifier. The wrapper method is computationally intensive and has to be applied for each binary classifier separately, however we decided to use it since it provides the best performing feature set for the chosen classifier.

[1]The WEKA Data Mining implementation of the classifiers is used [47].

Table 7.2: Different binary classifier used in this thesis

#	**Name**
1	Naïve Bayes
2	Bagging
3	Decision Table
4	Logistic Regression
5	Decision Stump
6	J48 Decision tree
7	BayesNet
8	SimpleCART
9	SVM
10	Neural network

Therefore, the first step is to select the most significant features for each pair of classifier and response variable. Then, the selected features for each pair is used in the binary classifier to predict the response variable.

7.3 Experiments

In this section, we present the performance analysis of our predictive models on Enron email dataset and the DBLP dataset. We demonstrate that our approach is effective to select appropriate features from the feature set, and these features can accurately predict the evolution of the dynamic communities. Furthermore, our experiments show that the selected features vary among different events and transitions. Note that, in these experiments, we apply the static L-metric to produce sets of disjoint communities for each snapshot. Furthermore, we incorporate the extraction of the topics for the entities and the discovered communities.

Given the feature set, and the response variables of Table 7.1, ($\mathrm{survive}\{\mathrm{true}, \mathrm{false}\}$, $\mathrm{cohesion}\{\mathrm{tighten}, \mathrm{loosen}\}$, $\mathrm{size}\{\mathrm{expand}, \mathrm{shrink}\}$, $\mathrm{merge}\{\mathrm{true}, \mathrm{false}\}$, $\mathrm{split}\{\mathrm{true}, \mathrm{false}\}$), we develop a 10-fold cross-validation framework in which the communities with their response variables and features are randomly partitioned into 10 equal size subsamples. Then, 9 subsamples are used as training data, while the remaining subsample is retained as the validation data for testing the predictive model. We repeat the cross-validation process 10 times and average the 10 results from the folds to produce a single estimation.

Table 7.3: Enron: Survive event prediction

Event	Predictive Model	Accuracy	Precision	Recall	F-measure
Survive	**Neural Network**	**78.1513**	0.782	0.782	0.782
	Naïve Bayes	77.3109	0.783	0.773	0.771
	J48 Decision tree	72.2689	0.723	0.723	0.723
	Logistic Regression	71.4286	0.715	0.714	0.714
	SimpleCART	71.4286	0.721	0.714	0.712
RSurvive[3]	**Naïve Bayes**	**91.5888**	0.92	0.916	0.916
	BayesNet	90.6542	0.907	0.907	0.907
	Neural Network	89.7196	0.897	0.897	0.897
	Logistic Regression	85.9813	0.863	0.86	0.86
	Decision Table	85.0467	0.884	0.85	0.846

In most of the experiments presented in the following, the two labels of the underlying response variable are not balanced. Thus, to prevent over fitting and balance the two class labels, we use SMOTE (synthetic minority oversampling technique) [21] when the number of instances are low. Whereas, in the case of having a high number of instances or having a huge difference between the number of the two labels, the undersampling technique[2], is applied to prevent the overfitting.

7.3.1 Enron Email Dataset

To predict any of the three events, all the communities detected at the twelve snapshots with their features and response variables are used to build predictive models for each event. In total we have 114 community instances, where $|\,\mathrm{survive} = \mathrm{true}\,| = 60$, $|\,\mathrm{split} = \mathrm{true}\,| = 24$, and $|\,\mathrm{merge} = \mathrm{true}\,| = 59$. We first select the influential features for each binary classifier using the wrapper method. Then, each binary classifier is trained with its selected features. Table 7.3 shows the top five accurate predictive models for the $\mathrm{survive}$ event. As shown in Table 7.3, the accuracy of all models is about 78%. However, a closer look at the falsely classified instances reveals that they are mostly communities of very small size, i.e. less than 3 members, while their meta community has the length of only one snapshot i.e. the community *form*s at a snapshot and *dissolve*s immediately. Therefore, we

[2]The spreadsubsample undersampling technique available in WEKA is used.

[3]RSurvive represents $\mathrm{survive}$ prediction on the reduced community instances (communities with more than 3 members, where their meta community last more than one snapshot).

remove these community instances (of a size less than three, and a meta community of length one), and retrain the model. This reduction is intuitive, since a community that consists of only two members does not really represent a group of nodes, and hence is not a real community. Moreover, a meta community with only one community instance, happens in an unstable and infrequent situation, where its prediction require anomaly detection technique and is beyond the scope of this paper.

The reduction procedure results in 76 community instances with $|\,\mathrm{survive} = \mathrm{true}\,| = 55$, where we see at least 20% increase in the accuracy of models, with accuracy as high as $\mathbf{92}\%$. Our results indicate that the survival of a community can be accurately predicted based on the features we defined and extracted, while using a typical general purpose classifier. For our two-stage cascade predictive model, we also consider the survive prediction on the reduced instances.

Table 7.4: Enron: Merge and Split events prediction

Event	Predictive Model	Accuracy	Precision	Recall	F-measure
Split	**Naïve Bayes**	**90.8046**	0.908	0.908	0.908
	Bagging	89.6552	0.9	0.897	0.896
	J48 Decision tree	88.5057	0.89	0.885	0.884
	Neural Network	88.5057	0.885	0.885	0.885
	Decision Table	87.931	0.88	0.879	0.879
Merge	**Naïve Bayes**	**71.9298**	0.723	0.719	0.719
	Neural Network	71.0526	0.711	0.711	0.71
	SimpleCART	69.2982	0.693	0.693	0.693
	SVM	66.6667	0.686	0.667	0.654
	J48 Decision tree	65.7895	0.664	0.658	0.652

We observe similar performance in predicting the other two events. The top five predicted models for split, and merge are shown in Table 7.4. We can see that our models predict the split of a community (into other communities in a next snapshot) with about 91% accuracy, regardless of the classifier used. Where, the merge event (of a community with another communities in a next snapshot) can be predicted with an accuracy as high as 72%.

The size, and cohesion transitions are preceded by the prediction of a survive event. Based on our results in Table 7.3, we choose the *Naïve Bayes* classifier to detect the survive events. Then, communities with predicted $\mathrm{survive} = \mathrm{true}$ using

this classifier are used to build the models for the size, and $\mathrm{cohesion}$ transitions. The *Naïve Bayes* classifier predicts 50 community instances with $\mathrm{survive} = \mathrm{true}$, for which we have $|\,\mathrm{size} = \mathrm{expand}\,| = 29$, and $|\,\mathrm{cohesion} = \mathrm{tighten}\,| = 23$. The top five predictive models for the size, and $\mathrm{cohesion}$ transitions are shown in Table 7.5. We can see that these size and cohesion transitions of a community[4], can be predicted with a high accuracy of 79%, and 77% respectively.

Table 7.5: Enron: Community Transitions prediction

Transition	Predictive Model	Accuracy	Precision	Recall	F-measure
Size	**Naïve Bayes**	**79.3103**	0.795	0.793	0.793
	J48 Decision tree	72.4138	0.724	0.724	0.724
	Neural Network	72.4138	0.728	0.724	0.723
	SVM	68.9655	0.715	0.69	0.68
	Decision Stump	68.9655	0.691	0.69	0.689
Cohesion	**SimpleCART**	**76.9231**	0.769	0.769	0.769
	Bagging	75	0.761	0.75	0.749
	Neural Network	75	0.754	0.75	0.75
	J48 Decision tree	73.0769	0.741	0.731	0.726
	Naïve Bayes	67.3077	0.764	0.673	0.648

7.3.2 DBLP Co-authorship Dataset

We perform similar analysis on the ten snapshots of the DBLP dataset. Similar to Enron dataset, to predict any of the three events, all the communities detected at the ten snapshots with their features and response variables can be used to build the predictive models for each event. In total, there are 7668 community instances, where $|\,\mathrm{survive} = \mathrm{true}\,| = 1814$, $|\,\mathrm{split} = \mathrm{true}\,| = 159$, and $|\,\mathrm{merge} = \mathrm{true}\,| = 315$. As shown in Table 7.6, the best accuracy for the $\mathrm{survive}$ predictive models is 62%. Similar to Enron, the false predicted instances are all small size communities with less than 3 members, where their meta community also has a duration one. With the same reasoning as before, we remove these instances. Here, the reduction procedure results in 1984 community instances with $|\,\mathrm{survive} = \mathrm{true}\,| = 1140$. The accuracy of the top five predictive models on these reduced instances is reported in Table 7.6. Our results confirm the trend we have observed on the Enron dataset, i.e.

[4]with at least three members and its meta community lasts more than one snapshot

Table 7.6: DBLP: Survive event prediction

Event	Predictive Model	Accuracy	Precision	Recall	F-measure
Survive	**J48 Decision tree**	**61.742**	0.618	0.617	0.617
	Bagging	61.6869	0.618	0.617	0.616
	Decision Table	61.301	0.613	0.613	0.613
	Neural Network	61.1356	0.614	0.611	0.609
	Naïve Bayes	61.0805	0.611	0.611	0.61
RSurvive	**Decision Table**	**84.1232**	0.879	0.841	0.837
	Decision Stump	84.1232	0.879	0.841	0.837
	Neural Network	84.0047	0.873	0.84	0.836
	BayesNet	82.4645	0.847	0.825	0.822
	SimpleCART	81.6351	0.859	0.816	0.811

the survival of a community can be accurately predicted based on our set of features (with a 84% accuracy).

The results for split, and merge are shown in Table 7.7. Our results indicate that, with 81% accuracy, we can predict the split of a community into other communities in a next snapshot. However, the best prediction accuracy for merge of a community with another community is 62%. The false predicted instances on merge do not have any clear characteristics to explain how we can get better accuracy. Thus, on DBLP, unlike survival and split, merging of communities with each other can not be accurately predicted based on the present set of features, where the best prediction accuracy is only 62%. This could be partly explained by a variety of external factors that can affect such event, for example meeting at a conference, moving between institutions, etc.

As shown in Table 7.6, applying the *Decision Table* classifier produces the highest accuracy for the survive event on reduced community instances. Thus, only the communities with predicted $\mathrm{survive} = \mathrm{true}$ using the *Decision Table* classifier are used to build the predictive models for the size, and $\mathrm{cohesion}$. In this case, we have 576 community instances, where $|\mathrm{size} = \mathrm{expand}| = 383$, and $|\mathrm{cohesion} = \mathrm{tighten}| = 440$. The top five predictive models on the size, and $\mathrm{cohesion}$ transitions are shown in Table 7.8 respectively. We see that the size and cohesion transitions of a community can be predicted with a 80%, and 92% accuracy respectively.

Table 7.7: DBLP: Merge and Split events prediction

Event	Predictive Model	Accuracy	Precision	Recall	F-measure
Split	**Naïve Bayes**	**80.5031**	0.805	0.805	0.805
	J48 Decision tree	80.1887	0.805	0.802	0.801
	SVM	79.8742	0.799	0.799	0.799
	Neural Network	79.8742	0.801	0.799	0.798
	SimpleCART	79.5597	0.798	0.796	0.795
Merge	**J48 Decision tree**	**61.5873**	0.644	0.616	0.596
	Bagging	60.1587	0.606	0.602	0.597
	SimpleCART	59.8413	0.601	0.598	0.596
	Decision Table	59.8413	0.609	0.598	0.588
	SVM	59.5238	0.598	0.595	0.593

Table 7.8: DBLP: Community Transitions prediction

Transition	Predictive Model	Accuracy	Precision	Recall	F-measure
Size	**BayesNet**	**79.622**	0.797	0.796	0.796
	Naïve Bayes	79.217	0.792	0.792	0.792
	Decision Table	78.812	0.789	0.788	0.788
	Bagging	78.543	0.785	0.785	0.785
	Neural Network	78.003	0.786	0.78	0.778
Cohesion	**Naïve Bayes**	**92.089**	0.921	0.921	0.921
	Neural Network	91.499	0.918	0.915	0.915
	Bagging	91.499	0.916	0.915	0.915
	Decision Stump	91.145	0.925	0.911	0.911
	SVM	91.145	0.915	0.911	0.911

Our experiment results on Enron and DBLP dataset show that, using the features provided in Table 7.1, the events and transitions of communities in a next snapshot, can be predicted accurately. Furthermore, the accuracy of all the classifiers and logistic regression model is relatively high, suggesting that our approach is independent of the underlying classifier or regression model.

7.3.3 Correlation between Features

Figure 7.1 shows the correlation between different features of Table 7.1. The correlation is measured as the absolute value of Spearman's rank correlation coefficient between different features. In order to better visualize the correlations, the rows and columns of the heat-map are clustered to create blocks of highly correlated features. For instance, *Density*, *ClusteringCoefficient*, *AverageDegree*, and *Aver-*

ageCloseness features are correlated as expected. Note that, their corresponding temporal features are also correlated with each other. However, as we can see in these heat-maps, most of the defined features are not highly correlated in neither Enron nor DBLP. This behaviour is desirable, since we define features to capture different properties of a community and its temporal changes. In other words, the low correlation/ overlap between features confirms that the features used in our predictive models are distinctive.

7.3.4 Ensemble Analysis

Any of the predictive models we introduced, selects a different set of features. We consider a feature is more prominent for a specific event or transition, if it is selected by the majority of the models trained for predicting that event or transition. Figure 7.2 shows the number of times that each feature is selected by our 10 predictive models for predicting each event in Enron dataset. Here, to better visualize the selection of the features, only the rows of the heat-map are clustered to create blocks of similarly coloured cells. The Pearson correlation between different features and the response variables is also depicted in Figure 7.2. Here, to calculate correlation between features and $\mathrm{cohesion}$, and size transitions, we consider their $\mathrm{tighten}$, and expand values respectively. Furthermore, to simplify the comparison between this heat-map and the one showing the selection of the features, the rows are ordered correspondingly. Similar to the number of times that a feature is selected, the correlation between each feature and response variables differs for different response variables. Moreover, the correlation of a feature is positively correlated with the number of times it is selected.

We can infer interesting patterns form this ensemble analysis. For example *ClusteringCoefficient* and *Cohesion* are prominent positive factors on the survival of Enron communities, while *StableLeaderTopics* and *LeftNodesRatio* are important negative factors on survival. The importance of these factors on $\mathrm{survive}$ is intuitive, for instance, a community with high clustering coefficient has strong relationship between its members and will not dissolve easily. On the other hand, losing members (i.e. high *LeftNodesRatio*) is a good sign of an unstable commu-

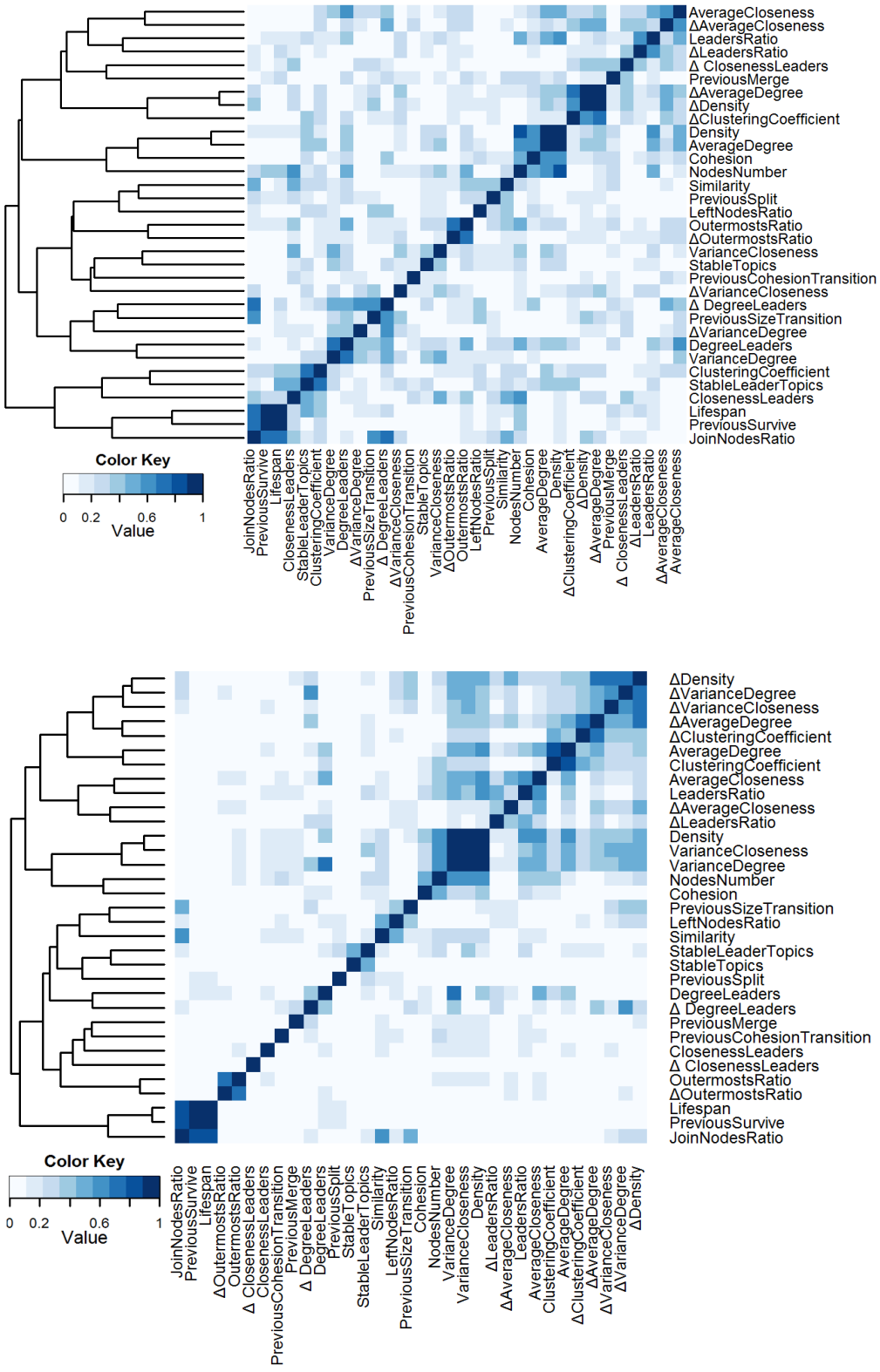


Figure 7.1: Absolute value of Spearman's rank correlation coefficient between different features. Top: Enron, Bottom: DBLP. These correlation matrices depict that the overlap between features used in our predictive models is low.

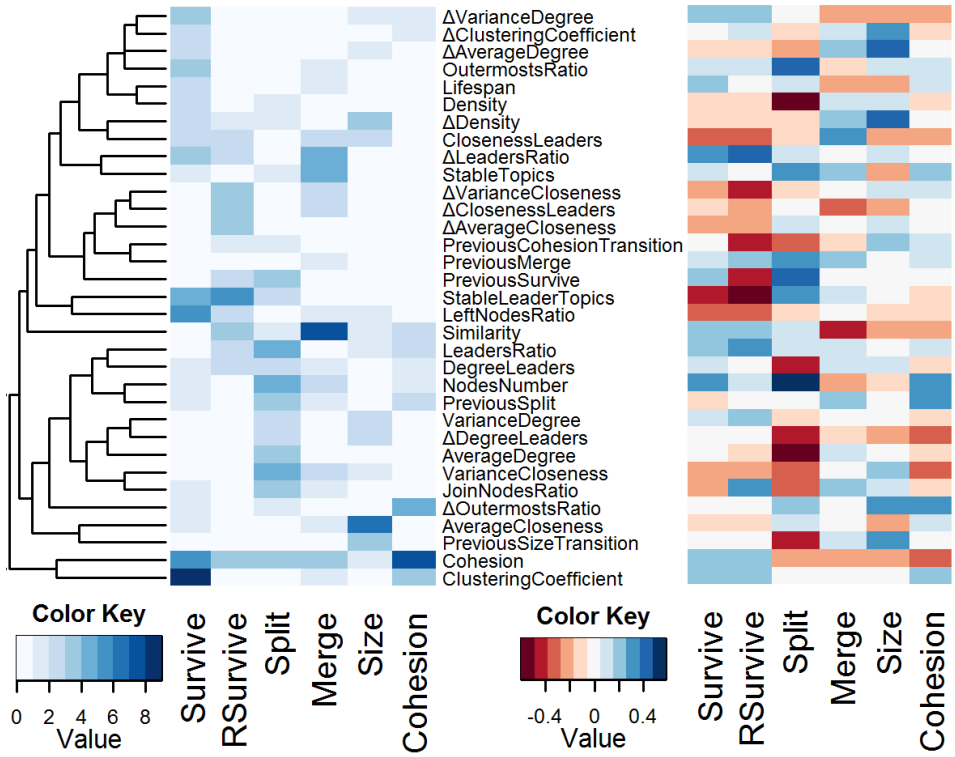


Figure 7.2: Enron: The number of times a feature is selected by the 10 predictive models (left), and the correlation between each feature and response variable (right).

nity which is not going to survive. In case of split, *LeadersRatio* and *NodesNumber* are positively important, i.e. a community with more leaders or a bigger size community is more probable to split. The negative effect of *VarianceCloseness* and *Cohesion* shows that a community with high variance of closeness scores and high cohesion is immune to split. The merge of a community is positively influenced by *StableTopics*, talking about the same topics over time leads the community to merge with another communities. On the other hand, *Similarity* has negative influence on merge, i.e. a community with almost stable members is not probable to merge with others.

Similarly, the ensemble analysis for DBLP is depicted in Figure 7.3. Again, interesting patterns can be inferred from these two heat-maps. For instance, *Density* is a positive factor in size transition, whereas, *NodesNumber* is negatively important, i.e. a dense community attract new members and expands, while a bigger size community has less chance to attract new members. We also observe that *Cohesion* is a prominent negative feature on the cohesion transition of a community in a later

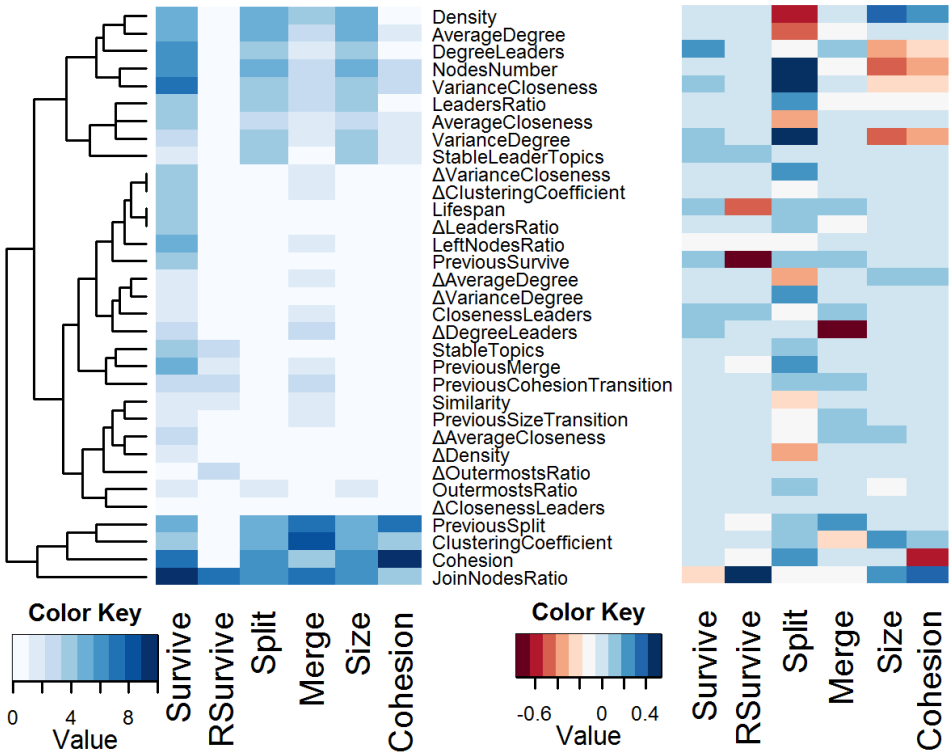


Figure 7.3: DBLP: The number of times a feature is selected by the 10 predictive models (left), and the correlation between each feature and response variable (right).

snapshot. This importance indicates that on DBLP, a less cohesive community has a better potential to form new connections and become more tighten.

Comparing the features importance between these two datasets, we see that these patterns although similar, depend on the underlying dynamic social network. This finding demonstrates the importance of the feature selection step for the prediction task.

Figure 7.4 provides the comparison between prominent features selected for the two datasets. Here, we only include the features that are selected more than five times by at least one (event or transition) predictive model. The diagrams show that, for the Enron dataset, *Cohesion*, *Similarity*, and *LeadersRatio* are the prominent features for all the five events and transitions. However, for instance, Δ*OutermostsRatio* is only a prominent feature for cohesion transition. For the DBLP dataset, *JoinNodesRatio* is influential for all the five events and transitions. On the other hand, *StableLeaderTopics* is influential in only size, and cohesion prediction. The difference between the prominent features of the events and transitions

for the two datasets shows that the triggers of the events and transitions are related to the domain dependent attributions. Furthermore, the evolution of the communities in each dataset is unique to that dataset. In other words, the application dictates the dynamics of a network.

7.4 Summary

We investigated the evolution of dynamic networks, at the level of their community structure. We defined and extracted an extensive set of relevant role-based, structural, contextual and temporal features, to represent the the structural and non-structural properties of communities and the behaviour of their (influential) members. Our experimental results on real-world datasets (Enron and DBLP) shows that the defined features are mainly non-overlapping, and distinctive. Based on which, the events and transitions of communities can be accurately predicted. Our predictive process also identifies the most prominent features for each community transition and event. We confirm the relation between the behavior of individuals, specially the influential members of a community, and the future of the community they belong to, and also observe many interesting, yet expected, evolution patterns, e.g. recruiting new members by a community is a good indicator of its survival.

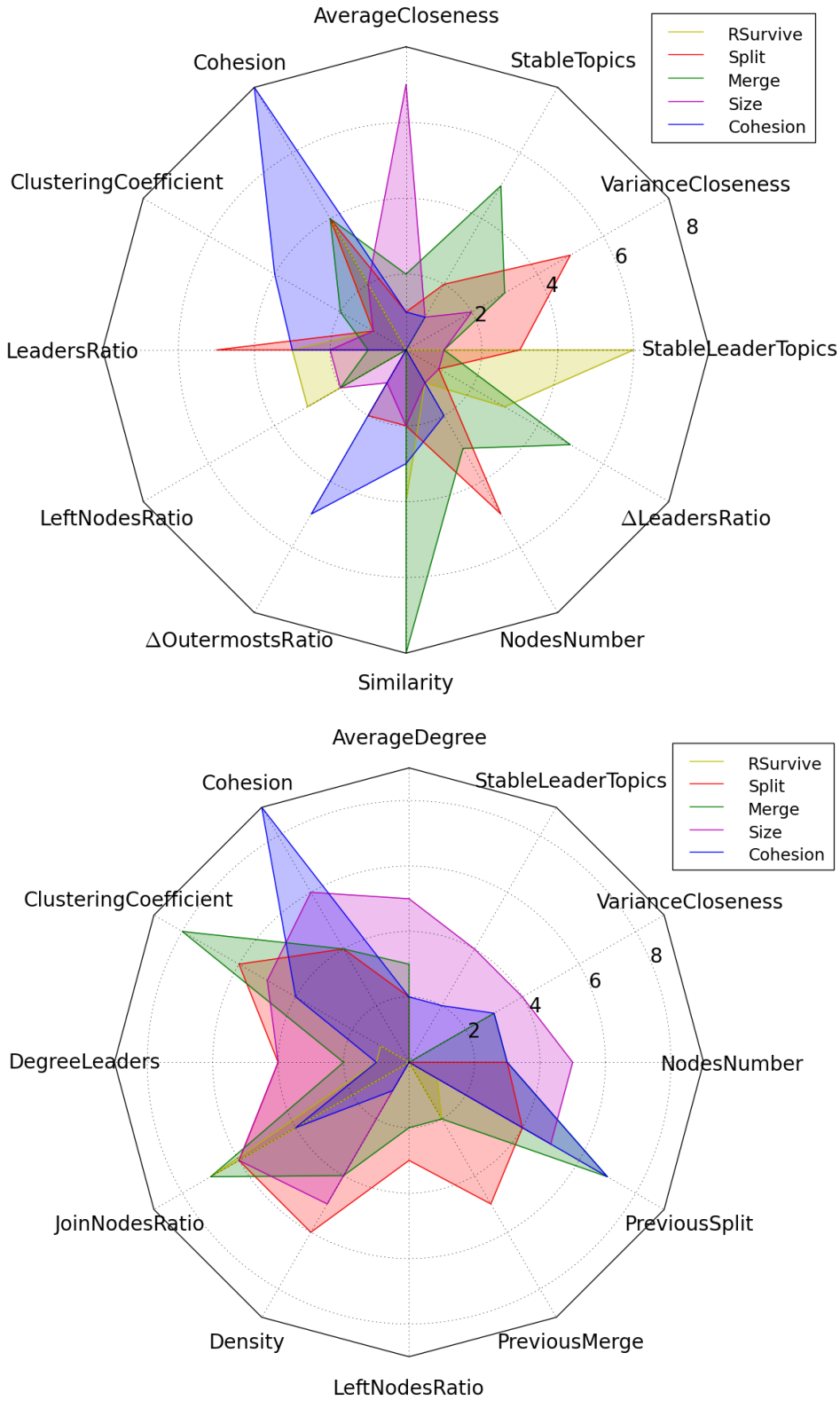


Figure 7.4: Comparison of prominent features on the ENRON (top) and DBLP (bottom) dataset. Only features that are selected more than five times by at least one event or transition are included.

Chapter 8

Conclusions

In any evolving complex network, understanding the features associated with the data and structural properties of the network has attracted many researchers recently. The knowledge of the community structure enables the prediction of some essential features of the systems under investigation. In this thesis, we first provide a complete overview on the existing methods to analyze the evolution of communities in the dynamic social network. Due to the limitation of the existing analyses, we propose to first uncover the communities of these networks, and then we provide a framework, called MODEC, to interpret the structure of large networks and then to predict how the modular structure of the network changes under different circumstances.

The MODEC framework investigates the evolution of dynamic networks, at the level of their community structure, by monitoring the transition and evolution of its enclosing communities over time. Our framework encompasses both a community matching algorithm and an event detection model that captures the critical events for communities. This includes tracking the formation, survival and dissolution of communities as well as identifying the meta communities, which are a series of similar communities at different snapshots. Furthermore, we also provide event and behaviour analysis related to the individuals in a network that can help identify the structural properties of the network. Applying our framework on the Enron email dataset, and DBLP co-authorship dataset, we uncover communities with different evolutionary characteristics and address the noticeable effect that the similarity threshold has on the evolution of communities.

In order to validate the choice of the similarity threshold, we propose extracting and investigating frequently used topics for each community and selecting the appropriate threshold based on the continuation of these topics. The performance of our framework on both datasets is also compared with the other event-based frameworks. The results show that the MODEC framework outperforms the others in terms of the average mutual topics between survival communities.

One of the challenging research problems in dynamic social networks is to mine communities. The traditional approach to solve this problem is to extract communities at each snapshot independent of the communities at other snapshots or the historic data. In this thesis, we propose an incremental L-metric community mining approach to consider both current and temporal data in the process of mining communities. The proposed method is then compared with its equivalent independent version and also with the most commonly used dynamic community method –FacetNet. Compared to these two methods, the incremental L-metric method detects communities with higher quality when assessed directly with a modified version of Q modularity for the dynamic scenario. It is worth mentioning that the choice of community mining algorithms can influence the results of the evolution of communities. Hence, the performance of different community mining algorithms also compared by assessing the quality of the mutual topics among the communities rather than their quantity. Our results show that incremental L-metric method is more successful in detecting the evolution patterns of the communities and triggering appropriate events.

We leverage the relationship between the evolution of communities, the movement of individuals between these communities and changes in the role of those individuals. We illustrate that changes in the role of individuals in a community have a direct relationship with the development of the community. The role change can act as triggers to evoke community changes. Role modification can affect the dynamics of communities and the events in the communities can alter the role of individuals. Through our visualizations, we demonstrate that analyzing community evolution events and entity role events gives us valuable insights on the dynamics of networks. Moreover, we observed how nodes change their role through time.

Based on role changes, we analyze community events happening in the Enron email dataset and observed mutual relation between role changes and community events. Tracking how these roles change through time provides information about the temporal characteristics of nodes and the network.

We furthermore define and extract an extensive set of relevant role-based, structural, contextual and temporal features, to represent the structural and non-structural properties of communities and the behaviour of their (influential) members. Our experimental results on real-world datasets (Enron and DBLP) show that the defined features are mainly non-overlapping, and distinctive, based on which, the events and transitions of communities can be accurately predicted. Our predictive process also identifies the most prominent features for each community transition and event. We confirm the relation between the behaviour of individuals, specially the influential members of a community, and the future of the community they belong to, and also observe many interesting, yet expected, evolution patterns, e.g. recruiting new members by a community is a good indicator of its survival. We observe that the significance of these features depend on the application, and therefore enough care should be given when selecting these features for prediction.

Bibliography

[1] Afra Abnar, Mansoureh Takaffoli, Reihaneh Rabbany, and Osmar R. Zaïane. Ssrm: Structural social role mining for dynamic social networks. In *International Conference on Advances in Social Networks Analysis and Mining*, ASONAM '14, 2014.

[2] Nitin Agarwal, Huan Liu, Lei Tang, and Philip S. Yu. Identifying the influential bloggers in a community. In *International Conference on Web Search and Data Mining*, pages 207–218, 2008.

[3] C. C. Aggarwal and P. S. Yu. Online analysis of community evolution in data streams. In *Proceedings of SIAM International Data Mining Conference*, SDM'05, 2005.

[4] Yong-Yeol Ahn, Seungyeop Han, Haewoon Kwak, Sue Moon, and Hawoong Jeong. Analysis of topological characteristics of huge online social networking services. In *International Conference on World Wide Web*, pages 835–844, 2007.

[5] Hessameddin Akhlaghpour, Mohammad Ghodsi, Nima Haghpanah, Vahab S Mirrokni, Hamid Mahini, and Afshin Nikzad. Optimal iterative pricing over social networks. In *Internet and Network Economics*, pages 415–423, 2010.

[6] Aris Anagnostopoulos, Ravi Kumar, and Mohammad Mahdian. Influence and correlation in social networks. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 7–15, 2008.

[7] Sitaram Asur, Srinivasan Parthasarathy, and Duygu Ucar. An event-based framework for characterizing the evolutionary behavior of interaction graphs. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2007.

[8] Sitaram Asur, Srinivasan Parthasarathy, and Duygu Ucar. An event-based framework for characterizing the evolutionary behavior of interaction graphs. *ACM Transactions on Knowledge Discovery from Data*, 3:16:1–16:36, 2009.

[9] Thomas Aynaud and Jean-Loup Guillaume. Static community detection algorithms for evolving networks. In *WiOpt Workshop on Dynamic Networks*, 2010.

[10] Lars Backstrom, Dan Huttenlocher, Jon Kleinberg, and Xiangyang Lan. Group formation in large social networks: membership, growth, and evolution. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '06, pages 44–54, 2006.

[11] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.

[12] Tanya Y. Berger-Wolf and Jared Saia. A framework for analysis of dynamic social networks. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '06, pages 523–528, 2006.

[13] A. Bernstein, S. Clearwater, S. Hill, C. Perlich, and F. Provost. Discovering knowledge from relational data. In *Proceedings of the KDD-2002 Workshop on Multi-Relational Data Mining*, MRDM '02, pages 7–20, 2002.

[14] Bruce J Biddle. Recent development in role theory. *Annual Review of Sociology*, pages 67–92, 1986.

[15] Vincent D. Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008:P10008+, 2008.

[16] András Bóta, Miklós Krész, and András Pluhár. Dynamic communities and their detection. *Acta Cybernetica*, 20(1):35–52, 2011.

[17] Piotr Bródka, Przemyslaw Kazienko, and Bartosz Koloszczyk. Predicting group evolution in the social network. In *International Conference on Social Informatics*, pages 54–67, 2012.

[18] Rajmonda Sulo Caceres, Tanya Berger-Wolf, and Robert Grossman. Temporal scale of processes in dynamic networks. In *Proceedings of the IEEE ICDM 2011 Workshop on Data Mining in Networks (DaMNet)*, 2011.

[19] Antoni Calvo-Armengol and Yves Zenou. Social networks and crime decisions: The role of social structure in facilitating delinquent behaviour. CEPR Discussion Papers 3966, C.E.P.R. Discussion Papers, 2003.

[20] Deepayan Chakrabarti, Ravi Kumar, and Andrew Tomkins. Evolutionary clustering. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '06, 2006.

[21] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. SMOTE: Synthetic minority over-sampling technique. *Artificial Intelligence Research*, 16:321–357, 2002.

[22] Jiyang Chen, Justin Fagnan, Randy Goebel, Reihaneh Rabbany, Farzad Sangi, Mansoureh Takaffoli, Eric Verbeek, and Osmar R. Zaïane. Meerkat: Community mining with dynamic social networks. In *Proceedings of 10th IEEE International Conference on Data Mining*, ICDM '10, 2010.

[23] Jiyang Chen, Osmar Zaïane, and Randy Goebel. Local community identification in social networks. In *International Conference on Advances in Social Network Analysis and Mining*, ASONAM '09, pages 237–242, 2009.

[24] Jiyang Chen, Osmar R. Zaïane, and Randy Goebel. Detecting communities in social networks using max-min modularity. In *SIAM International Conference on Data Mining*, pages 978–989, 2009.

[25] Yudong Chen, Vikas Kawadia, and Rahul Urgaonkar. Detecting overlapping temporal community structure in time-evolving networks. *CoRR*, abs/1303.7226, 2013.

[26] Prakash Mandayam Comar, Pang-Ning Tan, and Anil K Jain. Linkboost: A novel cost-sensitive boosting framework for community-level network link prediction. In *Proceedings of the 11th IEEE International Conference on Data Mining*, 2011.

[27] Stijn Dongen. A cluster algorithm for graphs. Technical report, National Research Institute for Mathematics and Computer Science, 2000.

[28] Dongsheng Duan, Yuhua Li, Yanan Jin, and Zhengding Lu. Community mining on dynamic weighted directed graphs. In *Proceeding of the 1st ACM international workshop on Complex networks meet information & knowledge management*, CNIKM '09, pages 11–18, 2009.

[29] Edelman. Edelman trust barometer report, 2010.

[30] Ergin Elmacioglu and Dongwon Lee. Modeling idiosyncratic properties of collaboration networks revisited. *Scientometrics*, 80(1):195–216, 2009.

[31] Stephen Eubank, Hasan Guclu, V. S. Anil Kumar, Madhav V. Marathe, Aravind Srinivasan, Zoltan Toroczkai, and Nan Wang. Modelling disease outbreaks in realistic urban social networks. *Nature*, 429(6988):180–184, 2004.

[32] Tanja Falkowski and Jörg Bartelheimer. Applying social network analysis methods to explore community dynamics. In Uwe Serdult and Volker Taube, editors, *Applications of Social Network Analysis 2005*, pages 189–212. Wissenschaftlicher Verlag Berlin, 2008.

[33] Tanja Falkowski, Jorg Bartelheimer, and Myra Spiliopoulou. Community dynamics mining. In *Proceedings of 14th European Conference on Information Systems*, ECIS '06, 2006.

[34] Tanja Falkowski, Jorg Bartelheimer, and Myra Spiliopoulou. Mining and visualizing the evolution of subgroups in social networks. In *Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence*, WI '06, pages 52–58, 2006.

[35] AICML: Alberta Innovates Centre for Machine Learning. Meerkat, A Social Network Analysis Tool. `http://www.aicml.ca/?q=node/41`, 2014.

[36] Mathilde Forestier, Anna Stavrianou, Julien Velcin, and Djamel A Zighed. Roles in social networks: Methodologies and research issues. *Web Intelligence and Agent Systems*, 10(1):117–133, 2012.

[37] S. Fortunato and M. Barthélemy. Resolution limit in community detection. *Proceedings of the National Academy of Sciences*, 104(1):36–41, 2007.

[38] Santo Fortunato. Community detection in graphs. *Physics Reports*, 486(35):75–174, 2010.

[39] A. C. Gavin and et al. Functional organization of the yeast proteome by systematic analysis of protein complexes. *Nature*, 415(5):141–7, 2002.

[40] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826, 2002.

[41] Bogdan Gliwa, Piotr Bródka, Anna Zygmunt, Stanislaw Saganowski, Przemyslaw Kazienko, and Jaroslaw Kozlak. Different approaches to community evolution prediction in blogosphere. In *International Conference on Advances in Social Networks Analysis and Mining*, pages 1291–1298, 2013.

[42] Sharad Goel and Daniel G. Goldstein. Predicting individual behavior with social networks. *Marketing Science*, 33(1):82–93, 2014.

[43] Mark Goldberg, Malik Magdon-Ismail, Srinivas Nambirajan, and James Thompson. Tracking and predicting evolution of social communities. In *International Conference on Social Computing*, pages 7–15, 2011.

[44] Mark K. Goldberg, Malik Magdon-Ismail, and James Thompson. Identifying long lived social communities using structural properties. In *International Conference on Advances in Social Networks Analysis and Mining*, pages 647–653, 2012.

[45] D. Greene, D. Doyle, and P. Cunningham. Tracking the evolution of communities in dynamic social networks. In *Proceeding of International Conference on Advances in Social Networks Analysis and Mining*, ASONAM '10, 2010.

[46] Steve Gregory. Finding overlapping communities in networks by label propagation. *New Journal of Physics*, 12(10):103018, 2010.

[47] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18, 2009.

[48] William Hendrix, Isaac K. Tetteh, and Ankit Agrawal. Community dynamics and analysis of decadal trends in climate data. In *Proceedings of the Third ICDM Workshop on Knowledge Discovery from Climate Data*, ClimKD '11, 2011.

[49] Shu Huang and Dongwon Lee. Exploring activity features in predicting social network evolution. In *IEEE International Conference on Machine Learning and Applications*, pages 7–15, 2011.

[50] Sanjay Ram Kairam, Dan J. Wang, and Jure Leskovec. The life and death of online groups: Predicting group growth and longevity. In *ACM International Conference on Web Search and Data Mining*, pages 673–682, 2012.

[51] David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the 9th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '03, pages 137–146, 2003.

[52] Edward Casey Kenley and Young-Rae Cho. Entropy-based graph clustering: Application to biological and social networks. In *Proceedings of the 11th IEEE International Conference on Data Mining*, 2011.

[53] Erica Suyeon Kim and Steve Sangki Han. An analytical way to find influencers on social networks and validate their effects in disseminating social games. In *International Conference on Advances in Social Network Analysis and Mining (ASONAM)*, pages 41–46. IEEE, 2009.

[54] Min-Soo Kim and Jiawei Han. A particle-and-density based evolutionary clustering method for dynamic networks. *Proceedings of the VLDB Endowment*, 2:622–633, August 2009.

[55] Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM*, 46(5):604–632, 1999.

[56] H. W. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistic Quarterly*, 2:83–97, 1955.

[57] Ravi Kumar, Jasmine Novak, and Andrew Tomkins. Structure and evolution of online social networks. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '06, pages 611–617, 2006.

[58] Andrea Lancichinetti and Santo Fortunato. Community detection algorithms: A comparative analysis. *Physical Review E*, 80(5):056117, 2009.

[59] Niels Landwehr, Mark Hall, and Eibe Frank. Logistic model trees. *Machine Learning*, 59(1-2):161–205, 2005.

[60] Jure Leskovec, Lada A Adamic, and Bernardo A Huberman. The dynamics of viral marketing. *ACM Transactions on the Web*, 1:5, 2007.

[61] Jure Leskovec, Lars Backstrom, Ravi Kumar, and Andrew Tomkins. Microscopic evolution of social networks. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 462–470, 2008.

[62] Jure Leskovec, Jon Kleinberg, and Christos 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*, pages 177–187, 2005.

[63] Yu-Ru Lin, Yun Chi, Shenghuo Zhu, Hari Sundaram, and Belle L. Tseng. Facetnet: a framework for analyzing communities and their evolutions in dynamic networks. In *Proceeding of the 17th international conference on World Wide Web*, 2008.

[64] Yu-Ru Lin, Yun Chi, Shenghuo Zhu, Hari Sundaram, and Belle L. Tseng. Analyzing communities and their evolutions in dynamic social networks. *ACM Transactions on Knowledge Discovery from Data*, 3:8:1–8:31, 2009.

[65] Feng Luo, James Z. Wang, and Eric Promislow. Exploring local community structures in large networks. *Web Intelligence and Agent Systems*, 6:387–400, 2008.

[66] Miller McPherson, Lynn Smith-Lovin, and James M Cook. Birds of a feather: Homophily in social networks. *Annual Review of Sociology*, 27(1):415–444, 2001.

[67] L. Meyers, M. Newman, and B. Pourbohloul. Predicting epidemics on directed contact networks. *Journal of Theoretical Biology*, 240(3):400–418, 2006.

[68] Alan Mislove, Massimiliano Marcon, Krishna P. Gummadi, Peter Druschel, and Bobby Bhattacharjee. Measurement and analysis of online social networks. In *ACM SIGCOMM conference on Internet measurement*, pages 29–42, 2007.

[69] Peter J. Mucha, Thomas Richardson, Kevin Macon, Mason A. Porter, and Jukka-Pekka Onnela. Community structure in Time-Dependent, multiscale, and multiplex networks. *Science*, 328(5980):876–878, 2010.

[70] M. E. J. Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical Review E*, 74(3):036104, 2006.

[71] M. E. J. Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23):8577–8582, 2006.

[72] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69:026113, 2004.

[73] M. E. J. Newman and J. Park. Why social networks are different from other types of networks. *Physical Review E*, 68(3):36122, 2003.

[74] Nam P. Nguyen, Thang N. Dinh, Sindhura Tokala, and My T. Thai. Overlapping communities in dynamic networks: Their detection and mobile applications. In *Proceedings of the 17th Annual International Conference on Mobile Computing and Networking*, MobiCom '11, pages 85–96, 2011.

[75] N.P. Nguyen, T.N. Dinh, D.T. Nguyen, and M.T. Thai. Overlapping community structures and their detection on social networks. In *IEEE Third Inernational Conference on Social Computing*, SocialCom '11, pages 35–40, 2011.

[76] H Ning, W Xu, Y Chi, Y Gong, and T Huang. Incremental spectral clustering with application to monitoring of evolving blog communities. *SIAM International Conference on Data Mining*, pages 261–272, 2007.

[77] OA Oeser and Frank Harary. A mathematical model for structural role theory: II. *Human Relations*, 1964.

[78] OA Oeser and Gordon O'BRIEN. A mathematical model for structural role theory: III. *Human Relations*, 1967.

[79] Oscar Adolf Oeser and Frank Harary. A mathematical model for structural role theory: I. *Human Relations*, 1962.

[80] Márcia D. B. Oliveira and João Gama. Bipartite graphs for monitoring clusters transitions. In *Advances in Intelligent Data Analysis IX, 9th International Symposium, IDA 2010, Tucson, AZ, USA, May 19-21, 2010. Proceedings*, volume 6065 of *Lecture Notes in Computer Science*, pages 114–124, 2010.

[81] Gnce Keziban Orman, Vincent Labatut, and Hocine Cherifi. Qualitative comparison of community detection algorithms. In *International Conference on Digital Information and Communication Technology and Its Applications*, volume 167, pages 265–279, 2011.

[82] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, November 1999.

[83] Jordi Palau, Miquel Montaner, Beatriz López, and Josep Lluís de la Rosa. Collaboration analysis in recommender systems using social networks. In *Proceedings of the 8th International Workshop on Cooperative Information Agents*, CIA '04, pages 137–151, 2004.

[84] Gergely Palla, Albert-Laszlo Barabási, and Tamas Vicsek. Community dynamics in social networks. *Fluctuation and Noise Letters*, 7:L273–L287, 2007.

[85] Gergely Palla, Albert-Laszlo Barabási, and Tamas Vicsek. Quantifying social group evolution. *Nature*, 446(7136):664–667, 2007.

[86] Gergely Palla, Imre Derenyi, Illes Farkas, and Tamas Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435(7043):814–818, 2005.

[87] Akshay Patil, Juan Liu, and Jie Gao. Predicting group stability in online social networks. In *International Conference on World Wide Web*, pages 1021–1030, 2013.

[88] Reihaneh Rabbany, Jiyang Chen, and Osmar R. Zaïane. Top leaders community detection approach in information networks. In *The fifth ACM workshop on Social Network Mining and Analysis*, SNA-KDD '10, 2010.

[89] Reihaneh Rabbany, Mansoreh Takaffoli, Justin Fagnan, Osmar R. Zaïane, and Ricardo J. G. B. Campello. Relative validity criteria for community mining algorithms. In *International Conference on Advances in Social Networks Analysis and Mining*, 2012.

[90] Martin Rosvall and Carl T. Bergstrom. An information-theoretic framework for resolving community structure in complex networks. *Proceedings of the National Academy of Sciences*, 104(18):7327–7331, 2007.

[91] Martin Rosvall and Carl T. Bergstrom. Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences*, 105(4):1118–1123, 2008.

[92] Purnamrita Sarkar and Andrew W. Moore. Dynamic social network analysis using latent space models. *SIGKDD Explor. Newsl.*, 7:31–40, 2005.

[93] Jerry Scripps, Pang-Ning Tan, and Abdol-Hossein Esfahanian. Node roles and community structure in networks. In *Proceedings of the 9th WebKDD and 1st SNA-KDD workshop on Web mining and social network analysis*, pages 26–35. ACM, 2007.

[94] Massoud Seifi and Jean-Loup Guillaume. Community cores in evolving networks. In *Proceedings of the 21st international conference companion on World Wide Web*, WWW '12 Companion, pages 1173–1180, 2012.

[95] Myra Spiliopoulou, Irene Ntoutsi, Yannis Theodoridis, and Rene Schult. Monic: modeling and monitoring cluster transitions. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '06, pages 706–711, 2006.

[96] Rajmonda Sulo, Tanya Berger-Wolf, and Robert Grossman. Meaningful selection of temporal resolution for dynamic networks. In *Proceedings of the Eighth Workshop on Mining and Learning with Graphs*, MLG '10, pages 127–136, 2010.

[97] Rajmonda Sulo, Tanya Berger-Wolf, and Robert Grossman. Temporal scale of processes in dynamic networks. In *Proceedings of the IEEE ICDM 2011 Workshop on Data Mining in Networks*, DaMNet'11, 2011.

[98] Yizhou Sun, Jie Tang, Jiawei Han, Manish Gupta, and Bo Zhao. Community evolution detection in dynamic heterogeneous information networks. In *Proceedings of the 8th Workshop on Mining and Learning with Graphs*, MLG '10, pages 137–146, 2010.

[99] Mansoureh Takaffoli. Community evolution in dynamic social networks - challenges and problems. In *Proceedings of the IEEE ICDM PhD Student Forum*, 2011.

[100] Mansoureh Takaffoli, Reihaneh Rabbany, and Osmar R. Zaïane. Incremental local community identification in dynamic social networks. In *International Conference on Advances in Social Networks Analysis and Mining*, ASONAM '13, 2013.

[101] Mansoureh Takaffoli, Reihaneh Rabbany, and Osmar R. Zaïane. Community evolution prediction in dynamic social networks. In *International Conference on Advances in Social Networks Analysis and Mining*, ASONAM '14, 2014.

[102] Mansoureh Takaffoli, Farzad Sangi, Justin Fagnan, and Osmar R. Zaïane. A framework for analyzing dynamic social networks. In *Proceedings of the 7th Conference on Applications of Social Network Analysis*, ASNA '10, 2010.

[103] Mansoureh Takaffoli, Farzad Sangi, Justin Fagnan, and Osmar R. Zaïane. Community evolution mining in dynamic social networks. *Procedia - Social and Behavioral Sciences*, 22:49–58, 2011.

[104] Mansoureh Takaffoli, Farzad Sangi, Justin Fagnan, and Osmar R. Zaïane. Modec - modeling and detecting evolutions of communities. In *5th International AAAI Conference on Weblogs and Social Media*, ICWSM '11, 2011.

[105] Mansoureh Takaffoli, Farzad Sangi, Justin Fagnan, and Osmar R. Zaïane. Tracking changes in dynamic information networks. In *International Conference on Computational Aspects of Social Networks*, 2011.

[106] Chayant Tantipathananandh and Tanya Berger-Wolf. Constant-factor approximation algorithms for identifying dynamic communities. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '09, pages 827–836, 2009.

[107] Chayant Tantipathananandh and Tanya Berger-Wolf. Constant-factor approximation algorithms for identifying dynamic communities. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2009.

[108] Chayant Tantipathananandh and Tanya Y. Berger-Wolf. Finding communities in dynamic social networks. In *Proceedings of the 11th IEEE International Conference on Data Mining*, ICDM '11, 2011.

[109] Chayant Tantipathananandh, Tanya Y. Berger-Wolf, and David Kempe. A framework for community identification in dynamic social networks. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '07, pages 717–726, 2007.

[110] Jean-Loup Guillaume Thomas Aynaud. Multi-step community detection and hierarchical time segmentation in evolving networks. In *Proceedings of the Fifth International Workshop on Social Network Mining and Analysis*, SNAKDD '11, 2011.

[111] Joshua R. Tyler, Dennis M. Wilkinson, and Bernardo A. Huberman. Email as spectroscopy: Automated discovery of community structure within organizations. In *Communities and technologies*, pages 81–96, 2003.

[112] John Walton. Differential patterns of community power structure: An explanation based on interdependence. *The Sociological Quarterly*, 9:3–18, 1968.

[113] Liaoruo Wang, Tiancheng Lou, Jie Tang, and John E. Hopcroft. Detecting community kernels in large social networks. In *IEEE International Conference on Data Mining*, 2011.

[114] Qinna Wang and Eric Fleury. Overlapping community structure and modular overlaps in complex networks. In Tansel Ozyer, Zeki Erdem, Jon Rokne, and Suheil Khoury, editors, *Mining Social Networks and Security Informatics*, Lecture Notes in Social Networks, pages 15–40. Springer Netherlands, 2013.

[115] Yi Wang, Bin Wu, and Nan Du. Community evolution of social network: Feature, algorithm and model. *Physics and Society*, page 16, 2008.

[116] Stanley Wasserman and Katherine Faust. *Social Network Analysis: Methods and Applications*. Cambridge University Press, 1994.

[117] Ian H. Witten, Gordon W. Paynter, Eibe Frank, Carl Gutwin, and Craig G. Nevill-Manning. KEA: Practical automatic keyphrase extraction. In *ACM Conference on Digital Libraries*, pages 254–255, 1999.

[118] Bin Wu, Qi Ye, Shengqi Yang, and Bai Wang. Group crm: a new telecom crm framework from social network perspective. In *Proceeding of the 1st ACM international workshop on Complex networks meet information and knowledge management*, CNIKM '09, pages 3–10, 2009.

[119] Jierui Xie, Stephen Kelley, and Boleslaw K. Szymanski. Overlapping community detection in networks: The state-of-the-art and comparative study. *ACM Computing Surveys*, 45(4):43:1–43:35, 2013.

[120] Jaewon Yang and Jure Leskovec. Community-affiliation graph model for overlapping network community detection. In *12th IEEE International Conference on Data Mining*, ICDM, pages 1170–1175, 2012.

[121] Jaewon Yang and Jure Leskovec. Overlapping community detection at scale: A nonnegative matrix factorization approach. In *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*, WSDM '13, pages 587–596, 2013.

[122] Tianbao Yang, Yun Chi, Shenghuo Zhu, Yihong Gong, and Rong Jin. A bayesian approach toward finding communities and their evolutions in dynamic social networks. In *SIAM International Conference on Data Mining*, pages 990–1001, 2009.

[123] Shipeng Yu, Kai Yu, and Volker Tresp. Soft clustering on graphs. In *The Neural Information Processing Systems*, NIPS, 2005.

[124] Jun Zhang, Mark S Ackerman, and Lada Adamic. Expertise networks in online communities: structure and algorithms. In *Proceedings of the 16th international conference on World Wide Web*, pages 221–230. ACM, 2007.

[125] Elena Zheleva, Hossam Sharara, and Lise Getoor. Co-evolution of social and affiliation networks. In *ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1007–1016, 2009.