Joint Dynamic Online Social Network Analytics Using Network, Content and User Characteristics

Dissertation

Presented in Partial Fulfillment of the Requirements for the Degree Doctor of Philosophy in the Graduate School of The Ohio State University

By

Yiye Ruan, B.Sc., B.Econ.

Graduate Program in Computer Science and Engineering

The Ohio State University

2015

Dissertation Committee:

Srinivasan Parthasarathy, Advisor

P. Sadayappan

Arnab Nandi

R. Kelly Garrett

Abstract

Online social networks (OSNs) allow Internet users all over the globe to share information, exchange thoughts, and work collaboratively. Not only do OSNs provide a channel of broadcasting real-world events as they unfold, they also enable a convenient way for users to exchange experience and opinions. Understanding the relation among network topology, users, content, and their dynamics can have a significant impact both from a theoretical standpoint as well as from a practical one, for instance, to understand online user behaviors and predict future online activities.

In this dissertation, I study the interplay of three important factors that encode most of the OSN dynamics: network structure, user-generated content, and user characteristics. We first present our broader contribution to computer science: the development of two novel graph algorithms for community detection and structural role detection, which are scalable to handle networks containing millions of nodes and edges. Both community and role assignments of nodes generate novel clusterings of OSN users and provide valuable insights into OSN activities, but they are often implicit or even unknown to OSN analysts. We bridge this chasm by designing algorithms that can automatically infer community and role information in large-scale OSN data. Our algorithms are (1) robust in the presence of noise in real-world data, and (2) efficient in processing large network datasets. A key element to both of these contributions is a practical approach for network sparsification which enables efficient

processing. Evaluated on various social networks containing hundreds of millions of edges, our algorithms outperform state-of-the-art approaches in terms of the ability of recovering ground truth communities and roles of OSN users. By augmenting the network structure with content information and performing joint inference, our algorithms are able to combat the impact of noise. At the same time, careful design and optimization of our algorithms render them highly efficient when compared with existing approaches, and even non-trivial speedups on some networks.

Then we investigate three analytical tasks on OSN activities from the perspective of a user: (1) predicting user engagement in online discussion, (2) understanding the divergence of user-generated content, and (3) identifying patterns in the shift of user sentiment over time. Underpinning this effort are scalable mechanisms to infer important topological characteristics of such networks including community affiliation and structural roles, as discussed above. Experiments with large-scale datasets constructed from real OSNs show that our approaches, which incorporate information on network, content, and users, have demonstrated significant improvements over existing work which only focuses on one single aspect. More importantly, the findings from our studies on large-scale OSN data often reflect similar phenomena observed in social networks in the traditional face-to-face setting, making it promising to apply these quantitative approaches in the analysis of a broader spectrum of social networks.

Acknowledgments

First of all, I am sincerely grateful to my advisor, Professor Srinivasan (Srini) Parthasarathy. He is a passionate investigator, demonstrating the rigorous attitude and never-stopping curiosity that every successful researcher should have. He is also a kind advisor, guiding me through various stages in the doctoral program, sharing with me valuable insights and feedbacks, and preparing me for future advances. Learning is a long pursuit, and I am utterly fortunate to have Srini as my advisor on this path.

This material is based upon work supported by the National Science Foundation under Grants IIS-1111118 "SoCS: Collaborative Research: Social Media Enhanced Organizational Sensemaking in Emergency Response", CCF-1240651 "EAGER: Collaborative Research: Scalable Graph Mining and Clustering on Desktop Supercomputers", DMS-1418265 "Sampling and Inference in Network Analysis", and IIS-1149599 "CAREER: Information Misperceptions in the Internet Era". Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

I would also like to extend my gratitude to my committee members, Professors P. Sadayappan, Arnab Nandi, and R. Kelly Garrett, as well as Professor Amit Sheth at Kno.e.sis Center in Wright State University. They have contributed numerous helpful suggestions throughout my study, without which this dissertation cannot be accomplished.

Moreover, I owe my appreciation to my internship mentors: Drs. Haixun Wang and Bin Shao (Microsoft Research Asia), Drs. Oliver Brdiczka, Jianqiang Shen and Juan Liu (Palo Alto Research Center), Yan Zheng, Varun Srivastava, Jim Cheng, and Abhishek Gattani (WalmartLabs), Drs. Bernardo Huberman and Sitaram Asur (HP Labs). Thanks to them, I have gained wonderful exposure to research and development in an industrial environment, and results from these internships have contributed directly to this dissertation.

Research is rarely a solo job, and I must thank my collaborators as well as previous and current members of the Data Mining Research Lab: David Fuhry, Hemant Purohit, Venu Satuluri, Xintian Yang, Yu-Keng Shih, Ye Wang, S M Faisal, Yang Zhang, Aniket Chakrabarti, Yu Wang, Jiongqian Liang, Amruta Joshi, Tyler Clemons, Roberto Oliveira, Sarvenaz Choobdar, and Bortik Bandyopadhyay. Many ideas come from the discussions with these brilliant minds, and I wish them all the best in their future endeavors.

The last part is reserved for the ones that are most important to me: my fiancée, Lu Wang, and my parents, Guangyao Ruan and Ping Ye. You have supported me with all the love you have, and that is what makes me, beyond this dissertation itself.

Vita

May 1, 1987	Born — Shanghai, China
Jan. 2007 – June 2007	Exchange Student, Department of Computer Science and Engineering, The Hong Kong University of Science and Technology.
June 2009	B.Sc. Intelligence Science and Engineering, Peking University.
June 2009	B.Econ. Economics (Double Major), Peking University.
Sept. 2009 – Aug. 2010	Graduate School Fellow, The Ohio State University.
Sept. 2009 – Present	Ph.D. Student, Department of Computer Science and Engineering, The Ohio State University.
Sept. 2010 – Present	Graduate Research Associate, Department of Computer Science and Engineering, The Ohio State University.
Sept. 2010 – Present	Graduate Teaching Associate, Department of Computer Science and Engineering, The Ohio State University.
June 2011 – Sept. 2011	Research Intern, Microsoft Research Asia.
May 2012 – Aug. 2012	Research Intern, Palo Alto Research Center.
May 2013	M.Sc. Computer Science and Engineering, The Ohio State University.
May 2013 – Aug. 2013	Technical Staff Intern, WalmartLabs.
May 2014 – Aug. 2014	Research Associate Intern, HP Labs.

Publications

Research Publications

S. Parthasarathy, Y. Ruan, and V. Satuluri. Community discovery in social networks: Applications, methods and emerging trends. *Social Network Data Analytics*, pages 79–113, 2011.

V. Satuluri, S. Parthasarathy, and Y. Ruan. Local graph sparsification for scalable clustering. In *SIGMOD 2011*, pages 721–732. ACM, 2011.

H. Purohit, Y. Ruan, A. Joshi, S. Parthasarathy, and A. Sheth. Understanding user-community engagement by multi-faceted features: A case study on twitter. *SoME'11 (Workshop on Social Media Engagement, WWW'11)*, 2011.

Y. Ruan, H. Purohit, D. Fuhry, S. Parthasarathy, and A. Sheth. Prediction of topic volume on twitter. In *Web Science 2012, WebSci '12, Evanston, IL, USA - June 22 - 24, 2012*, pages 397–402. ACM, 2012.

D. Fuhry, Y. Ruan, and S. Parthasarathy. Local/global term analysis for discovering community differences in social networks. In *Web Science 2012, WebSci '12, Evanston, IL, USA - June 22 - 24, 2012*. ACM, 2012.

X. Yang, A. Ghoting, Y. Ruan, and S. Parthasarathy. A framework for summarizing and analyzing twitter feeds. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 370–378. ACM, 2012.

Y. Ruan, D. Fuhry, and S. Parthasarathy. Efficient community detection in large networks using content and links. In *Proceedings of the 22nd international conference on World Wide Web*, pages 1089–1098. International World Wide Web Conferences Steering Committee, 2013.

J. Shen, O. Brdiczka, and Y. Ruan. A comparison study of user behavior on facebook and gmail. *Computers in Human Behavior*, 29(6):2650–2655, 2013.

X. Yang, Y. Ruan, S. Parthasarathy, and A. Ghoting. Summarization via pattern utility and ranking: A novel framework for social media data analytics. *IEEE Data Eng. Bull.*, 36(3):67–76, 2013.

Y. Ruan and S. Parthasarathy. Simultaneous detection of communities and roles from large networks. In *Proceedings of the second edition of the ACM conference on Online social networks*, pages 203–214. ACM, 2014.

H. Purohit, Y. Ruan, D. Fuhry, S. Parthasarathy, and A. Sheth. On understanding the divergence of online social group discussion. In *Proceedings of the Eighth International Conference on Weblogs and Social Media, ICWSM 2014, Ann Arbor, Michigan, USA, June 1-4, 2014.*, 2014.

Y. Shih, S. Kim, Y. Ruan, J. Cheng, A. Gattani, T. Shi, and S. Parthasarathy. Component detection in directed networks. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 1729–1738. ACM, 2014.

Fields of Study

Major Field: Computer Science and Engineering

Table of Contents

Page

List of Tables

List of Figures

List of Algorithms

Chapter 1: Introduction

Online social networks (OSNs) have revolutionized the way information spreads in the cyber-space, as Internet users are not only direct recipients of information from mass media outlets, but also creators and critical relays of information themselves. By sending or forwarding statuses, images, and videos from computers or mobile devices, OSN users can easily disseminate content, interact with others, and engage in online discussions. As pointed out in a report by Pew Research Center [38], the proportion of Internet users who use social network sites has increased dramatically from 8% in 2005 to 73% in 2013, and the growth is expected to continue along with the deployment of faster Internet access to a broader population. Not only have OSNs become a convenient means for private communication, their ubiquitousness and the capability as information hubs have also played an important role in events with varied social significance, audience, and duration. To name a few examples, OSNs have been adopted in the scenarios of crisis response [119, 101], disease surveillance [36, 53], political campaigns [134, 31], and brand marketing [72, 80].

To analyze large scale OSNs, we first need a firm grasp of the structure of the underlying networks. While it is relatively easy to directly spot interesting patterns within small networks [150, 5, 51], the size of OSNs nowadays renders the workload

overwhelming for human analysts.[1] As a result, scalable graph algorithms are in an urgent need, in order to achieve comprehensive OSN analytics. Several exemplary graph-based problems are listed below, and they are open-ended in nature. Research advances made in these problems will create a broader impact not only to OSN analytics, but to computer science in general.

- **Community Detection:** One central prerequisite of community dynamic analysis is the information on community structure in a network. However, for many real-world OSNs, the knowledge of community assignment is either inaccessible or nonexistent. To combat this rarity of information, it is often necessary to mine meaningful communities from the given networks. How should we identify clusters of network users that are well-connected among themselves but are weakly linked to the rest of the network? Especially given the affluence of million-to-billion-users network data, how can communities be discovered in an efficient manner?

- **Structural Role Detection:** Network nodes can be categorized into roles according the connectivity characteristics of their local neighborhoods and beyond, and a node's role signifies its behavior and functionality, providing a complementary aspect to network communities. For instance, a central hub user with many links can be an ideal candidate of starting an information cascade, whereas a gateway bridge user connects several sub-networks that would otherwise be sparsely linked or even completely disconnected. In this sense, structural role

[1] A monitor screen with a 1024X768 resolution displays less than 1 million nodes, if each node only occupies one pixel. Even the latest 4K standard allows no more than 9 million nodes.

detection offers to cluster users in a different approach from community detection. In reality, again, information on structural roles is hardly available, calling for methods that can automatically infer the role assignment of users from large networks.

- **Network Simplification:** The quality and efficiency of graph algorithms largely hinge on the underlying network, as (1) the complexity of many algorithms is proportional to network size, and (2) noise in forms of incorrect links and missing links are abundant in real-world networks, including OSNs. Given a network, and possibly other auxiliary information associated with the network, can we simplify the network (i.e. *sample* the network) such that graph algorithms can run faster while retaining the same level of output quality?

We also note that interactions among individuals in OSNs are often dynamic, so are the development of events in real life. As a result, OSN activities should not be viewed as disconnected from each other, and dynamic analytics of OSNs has become an appealing research area. While much effort has been invested in analyzing OSNs activities as static episodes [93, 81, 78, 114], less emphasis has been afforded to the temporal development of user characteristics, content, and network structure. To better understand the social dynamics of OSNs, several central problems have arisen to researchers' attention:

- **Can we predict user engagement in online activities?** OSN users often engage in online activities that correspond to an offline event, be it a natural disaster, a political campaign, or a newly-released movie. The perspective of accurately predicting online user engagement is compelling, as it enables one to

gain insight into the future development of online activities, and even to provide feedback into the offline event itself. Here, the task is to predict whether, when, and how much will OSN users engage in the online activities surrounding a real-world event.

- **What factors can help us understand the content divergence in online social groups?** One way for groups of OSN users to engage in online activities is to join in the discussion by writing or sharing event-relevant content. Naturally, the evolution of user-generated content provides a new dimension to characterize the cohesion of online social groups, apart from the change of group structure. Therefore, given a principled definition of a group's discussion divergence, it is intriguing to learn the relationship between discussion divergence and group structure as well as that with user characteristics, and perhaps more importantly, to predict the group's future discussion divergence.

- **What makes users shift their sentiments during online conversations?** OSN users often express their sentiments on specific events and topics in the content they generated, and such sentiments may shift over time. While prior work has already identified evidences supporting the existence of sentiment shift in OSNs, we would like to model sentiment shift at a finer level, by studying the patterns of sentiment shift over events from various domains, in addition to the impact on sentiment shift by other factors, such as content property, network structure, and user properties.

This dissertation will cover my research contributions to scalable graph algorithms and dynamic OSN analytics. Both thrusts naturally complement the effort of each

other. As aforementioned, graph algorithms serve as important primitive operators in OSN analytics, often being used to infer latent user characteristics (e.g. communities, structural roles) when understanding OSN activities. On the other hand, novel problems identified in OSN analytics have also motivated the research of graph algorithms, one example being the need for overlapping community detection algorithms since the social circles of OSN users are hardly disjoint.

1.1 Limitations of Existing Work

Since the early stage of online social media, researchers have invested enormous amount of efforts in understanding OSNs and their dynamics [1, 83]. The study of graph algorithms also boasts a long history, and many problems have been extensively investigated [84, 46, 104, 105]. In spite of progress made over the past several decades, many challenges have not been fully addressed. Specifically, we identify three main obstacles as follow:

First of all, real-world OSNs data are way more complex than what idealistic statistical models are able to describe, due to the inherent noise in the data. In terms of network structure, noise is present in the forms of both incorrect link (false positive) and missing link (false negative). Not only does noise increase the complexity of network and impact the efficiency of network algorithms, it also jeopardizes the quality of analysis results when blindly included. The dual challenges are especially pronounced for graph algorithms such as community detection and structural role detection, which aim at inferring user groupings based on the structural information. A number of solutions have emerged in the literature to address the challenges. Broadly speaking, they either focus on speeding up the algorithm via sampling the underlying

network (nodes or edges) [91, 70, 2, 122], or attempt to combat noise by augmenting the network with auxiliary information [30, 152, 100, 149, 25, 153, 41, 59, 62, 137]. Given the presence of additional knowledge, the latter approach outperforms the former in terms of quality, but often at the cost of additional complexity. Therefore, is has become critical to design graph algorithms that are efficient and robust to noise, especially considering the vast scale of OSN data.

Moreover, many studies in (static or dynamic) OSN analytics have focused on one isolated aspect each time: network [33, 6, 123, 9], content [147, 127, 115, 139, 58, 74], or user [19, 114, 110, 106], Consequently, there lacks the thrust for an integrative study that fuses knowledge from multiple facets. Although a limited number of approaches have considered network, content, and users [7, 141], they are not designed to analyze OSN dynamics in an event-based setting. As discussed earlier, it is important to understand the interplay among these factors in event-based activities, as it allows one to capture predictive patterns in OSN dynamics and even real-world event developments. Therefore, a more holistic approach is needed when studying the dynamics of event-based content and online user groups.

Finally, rich theories in socio-psychology and other relevant fields have been under-investigated in the setting of OSN. While OSNs have blessed computational social scientists with an unprecedentedly huge volume of data for large-scale experiments [6, 111, 123, 69, 74], relatively few work has examined the link between socio-psychological theories and OSN analytics [58, 55]. To better model OSN dynamics, there is a strong motivation to seek guidance from existing research results on social group behaviors, and connecting those theories and various features inspired by

network structure, language usage, and behavioral characteristics, that have been adopted in the OSN literature.

These challenges lead to the statement of this dissertation as below.

1.2 Dissertation Statement

The large scale, rich information, and evolving nature of online social networks call for developing scalable graph algorithms and analyzing their dynamics by examining the interplay among network structure, user-generated content, and user characteristics. We envision that a holistic approach incorporating all these aspects will lead to a more comprehensive understanding of OSN dynamics, as well as its impact on the real world.

1.3 Contributions and Organizations of the Dissertation

This dissertation is composed of two main components. The first part (Chapters 2 and 3) consists of contributions to robust and efficient graph algorithms, which in turn serve as the foundation supporting dynamic analytics of OSN activities (Chapters 4 – 6). We will conclude this dissertation and discuss directions for future work in Chapter 7.

Robust and Efficient Graph Algorithms for OSN Analytics:

In Chapter 2, we will detail CODICIL[2], a family of highly efficient graph simplification algorithms leveraging both content and graph topology to identify and retain import edges in OSNs. Our approach relies on fusing content and topological (link) information in a natural manner. The output of CODICIL is a transformed variant of the original graph (with content information), which can serve as input to any fast

[2]COmmunity Discovery Inferred from Content Information and Link-structure

content-insensitive community detection algorithm, such as METIS or Markov Clustering. Through extensive experiments on real-world datasets drawn from Flickr, Wikipedia, and CiteSeer, and across several community detection algorithms, we demonstrate the effectiveness and efficiency of our methods. We find that CODICIL runs several orders of magnitude faster than state-of-the-art approaches, and it often identifies communities of comparable or superior quality on these datasets.

Next in Chapter 3, we will present RC-Joint[3], a novel algorithm that simultaneously identifies community and structural role assignments in a network. Prior work in graph algorithms addresses community detection and structural role detection as two separate problems, despite the fact that they are inter-related with each other. Members in a community often have different roles, whereas users of the same role are distributed in different communities. Rather than being agnostic to one assignment (community or role) while inferring the other, RC-Joint employs a principled approach to guide the detection process in a nonparametric fashion and ensures that the two sets of assignments are sufficiently different from each other. Roles and communities generated by RC-Joint are both soft assignments, reflecting the fact that many real-world networks have overlapping community structures and role memberships. By comparing with state-of-the-art methods in community detection and structural role detection, we demonstrate that RC-Joint harvests the best of two worlds and outperforms existing approaches, while still being competitive in efficiency. We also investigate the effect of initializing the algorithm with different schemes, and find that using the results of RC-Joint on a sparse network as the initialization seed often leads to faster convergence and higher quality.

[3]R(ole)-C(ommunity)-Joint

Dynamic Analytics of OSN Activities:

In the second part, we first study the problem of predicting user engagement in event-oriented discussions. An OSN user is said to engage in an event-oriented discussion, in a specific time frame, if the user composes new messages that are relevant to the underlying event or shares existing relevant messages during that period of time. For example, during Hurricane Sandy in 2012, local residents tweeting about the hurricane and its impact would be considered part of the Hurricane Sandy event-oriented community. We use Twitter as a social information source and manage to build an analytical model, which involves a broad range of features in four categories: content, author, network, and history. Using this integrated framework, we go beyond previous work which resort to isolated subsets of features, and perform classification as well as regression tasks to predict whether, when and how much OSN users will engage in event-oriented discussion in the future. Experiments on various real-world events demonstrate that the holistic approach is able to achieve better performance than considering individual sector of features. Further, we find that correlations exist between event types and features, which can help understand user engagement in better scientific ways. This part is described in Chapter 4.

In Chapter 5, we will study online social group dynamics based on the content divergence in group members' online discussions around events. Particularly, we use Jensen-Shannon divergence to measure the divergence of topics in user-generated contents, and how it progresses over time. We study tweets relevant to real-world events of varying duration and demographics, such as natural disasters, social activism and political campaigns. We discover that the discussion divergence of a typical online social group increases as the event is unfolding, and ebbs away when the

event finishes. We also model structural and user features with guidance from two socio-psychological theories, namely, social cohesion and social identity, to learn their implications on group discussion divergence. We found that strong correlation exists between selected features and discussion divergence. Using those features, we are able to train a machine learning classifier to predict the future increase or decrease in discussion divergence. The classifier is able to achieve an area under the curve of 0.84 and an F-1 score of 0.8. The ability to predict future divergence can help to identify and prioritize which cohesive groups to engage with in scenarios such as disaster response coordination.

Finally, we move forward to Chapter 6 to study the shift of topic-specific sentiment by OSN users over time, and investigate its relation with the information users receive from others in the network. We analyze more than 5 million tweets composed by Twitter users, and identify tweet content features that can drive sentiment shifts of users. Our experimental results signal the important role that content plays in sentiment shift, especially in changing opinions from positive to negative. We also find that users become less likely to shift to the dominant sentiment of a topic after multiple turns of tweets from influencers. We also show that in order to maximize the spread of a certain sentiment in a network, influencers should use tweets containing the target sentiment as well as supporting quotations.

Conclusion and Future Work:

In the long term, the goal of this dissertation is to promote a holistic approach in OSN analytics, instead of focusing on one aspect each time. In the realm of graph algorithms, we advocate fusing structural and content similarity in community detection, as well as guiding structural role discovery with community information. For

tasks in analyzing dynamic OSN activities, we have also proposed methods that consider network structure, content information, and user characteristics simultaneously. With the increasing richness of OSN information (e.g. user profile, multimedia content, connection heterogeneity), we believe this paradigm will see wider applicability in the future. We will conclude this dissertation, point out limitations of the present work, and also outline directions for future work in Chapter 7.

Chapter 2: Efficient Community Detection in Large Networks using Content and Links

In this chapter, we discuss the first part of our contributions to graph algorithms in OSN analytics — that of combining link and content information for the purposes of inferring communities of interest. This work directly addresses the open-ended problems of community detection and network simplification, described in Chapter 1, and brings benefits to structural feature extraction in OSN analytics (Chapter 5). As noted earlier, the challenges are manifold. The topological characteristics of such problems (graphs induced from the natural link structure) makes identifying community structure difficult. Further complicating the issue is the presence of noise in forms of incorrect link (false positive) and missing link (false negative). Determining how to fuse this link structure with content information efficiently and effectively is unclear. Finally, underpinning these challenges, is the issue of scalability as many of these graphs are extremely large running into millions of nodes and billions of edges, if not larger.

Given the fundamental nature of this problem, a number of solutions have emerged in the literature. Broadly these can be classified as: (1) those that ignore content information (a large majority) and focus on addressing the topological and scalability challenges, and (2) those that account for both content and topological information.

From a qualitative standpoint the latter presumes to improve on the former (since the null hypothesis is that content should help improve the quality of the inferred communities) but often at a prohibitive cost to scalability.

We shall present CODICIL[4], a family of highly efficient graph simplification algorithms leveraging both content and graph topology to identify and retain important edges in a network. Our approach relies on fusing content and topological (link) information in a natural manner. The output of CODICIL is a transformed variant of the original graph (with content information), which can then be clustered by any fast content-insensitive graph clustering algorithm such as METIS or Markov clustering. Through extensive experiments on real-world datasets drawn from Flickr, Wikipedia, and CiteSeer, and across several graph clustering algorithms, we demonstrate the effectiveness and efficiency of our methods. We find that CODICIL runs several orders of magnitude faster than those state-of-the-art approaches and often identifies communities of comparable or superior quality on these datasets.

2.1 Related Work

Community Discovery using Topology (and Content): Graph clustering/partitioning for community discovery has been studied for more than five decades, and a vast number of algorithms (exemplars include Metis [71], Graclus [37] and Markov clustering [138]) have been proposed and widely used in fields including social network analytics, document clustering, bioinformatics and others. Most of those methods, however, discard content information associated with graph elements. Due to space limitations, we suppress detailed discussions and refer interested readers to recent

[4]COmmunity Discovery Inferred from Content Information and Link-structure

surveys (e.g. [46]) for a more comprehensive picture. Leskovec et al. compared a multitude of community discovery algorithms based on conductance score, and discovered the trade-off between clustering objective and community compactness [82].

Various approaches have been taken to utilize content information for community discovery. One of them is generative probabilistic modeling which considers both contents and links as being dependent on one or more latent variables, and then estimates the conditional distributions to find community assignments. PLSA-PHITS [30], Community-User-Topic model [152] and Link-PLSA-LDA [100] are three representatives in this category. They mainly focus on studies of citation and email communication networks. Link-PLSA-LDA, for instance, was motivated for finding latent topics in text and citations and assumes different generative processes on citing documents, cited documents as well as citations themselves. Text generation is following the LDA approach, and link creation from a citing document to a cited document is controlled by another topic-specific multinomial distribution.

Yang et al. [149] introduced an alternative discriminative probabilistic model, PCL-DC, to incorporate content information in the conditional link model and estimate the community membership directly. In this model, link probability between two nodes is decided by nodes' *popularity* as well as community membership, which is in turn decided by content terms. A two-stage EM algorithm is proposed to optimize community membership probabilities and content weights alternately. Upon convergence, each graph node is assigned to the community with maximum membership probability.

Researchers have also explored ways to augment the underlying network to take into account the content information. The SA-Cluster-Inc algorithm proposed by

Zhou et al. [153], for example, inserts virtual *attribute nodes* and *attribute edges* into the graph and computes all-pair random walk distances on the new *attribute-augmented graph*. K-means clustering is then used on original graph nodes to assign them to different groups. Weights associated with attributes are updated after each k-means iteration according to their clustering tendencies. The algorithm iterates until convergence.

Ester et al. [41] proposed an heuristic algorithm to solve the *Connected k-Center* problem where both correctness and radius constraints need to be satisfied. The complexity of this method is dependent on the longest distance between any pair of nodes in the feature space, making it susceptible to outliers. Biologists have studied methods [62, 137] to find functional modules using network topology and gene expression data. Those methods, however, bear domain-specific assumptions on data and are therefore not directly applicable in general.

Recently Günnemann et al. [59] introduced a subspace clustering algorithm on graphs with feature vectors, which shares some similarity with our topic. Although their method could run on the full feature space, the search space of their algorithm is confined by the intersection, instead of union, of the epsilon-neighborhood and the density-based combined cluster. Furthermore, the construction of both neighborhoods are sensitive to their multiple parameters.

While decent performance can be achieved on small and medium graphs using those methods, it often comes at the cost of model complexity and lack of scalability. Some of them take time proportional to the number of values in each attribute. Others take time and space proportional to the number of clusters to find, which is often unacceptable. Our method, in contrast, is more lightweight and scalable.

Clustering/Learning Multiple Graphs: Content-aware clustering is also related to multiple-view clustering, as content information and link structure can be treated as two views of the data. Strehl and Ghose [126] discussed three consensus functions (cluster-wise similarity partitioning, hyper-graph partitioning and meta-clustering) to implement cluster ensembles, in which the availability of each individual view's clustering is assumed. Tang et al. [130] proposed a linked matrix factorization method, where each graph's adjacency matrix is decomposed into a characteristic matrix and a common factor matrix shared among all graphs. The purpose of factorization is to represent each vertex by a lower-dimensional vector and then cluster the vertices using corresponding feature vectors. Their method, while applicable to small-scale problems, is not designed for web-scale networks.

Graph Sampling for Fast Clustering: Graph sampling (also known as *sparsification* or *filtering*) has attracted more and more focus in recent years due to the explosive growth of network data. If a graph's structure can be preserved using fewer nodes and/or edges, community discovery algorithms can obtain similar results using less time and memory storage. Maiya and Berger-Wolf [91] introduced an algorithm which greedily identifies the node that leads to the greatest *expansion* in each iteration until the user-specified node count is reached. By doing so, an expander-like node-induced subgraph is constructed. After clustering the subgraph, the unsampled nodes can be labeled by using collective inference or other transductive learning methods. This extra post-processing step, however, operates on the original graph as a whole and easily becomes the scalability bottleneck on larger networks.

Satuluri et al. [122] proposed an edge sampling method to preferentially retain edges that connect two similar nodes. The localized strategy ensures that edges

in the relatively sparse areas will not be over-pruned. This proposed method has yielded significant empirical improvements of community detection algorithms on a series of real-world network datasets, and its theoretical underpinning was recently investigated by Gupta et al. [60]. Their method, however, does not consider content information either.

Edge sampling has also been applied to other graph tasks. Karger [70] studied the impact of random edge sampling on original graph's cuts, and proposed randomized algorithms to find graph's minimum cut and maximum flow. Aggarwal et al. [2] proposed using edge sampling to maintain structural properties and detect outliers in graph streams. The goals of those work are not to preserve and discover community structure in graphs, however.

2.2 Methodology

We begin by defining the notations used in the rest of Chapter 2. Let $\mathcal{G}_t = (\mathcal{V}, \mathcal{E}_t, \mathcal{T})$ be an undirected graph with n vertices $\mathcal{V} = v_1, \ldots, v_n$, edges $\mathcal{E}_t$, and a collection of n corresponding term vectors $\mathcal{T} = \boldsymbol{t}_1, \ldots, \boldsymbol{t}_n$. We use the terms "graph" and "network" interchangeably as well as the terms "vertex" and "node". Elements in each term vector $\boldsymbol{t}_i$ are basic content units which can be single words, tags or n-grams, etc., depending on the context of underlying network. For each graph node $v_i \in \mathcal{V}$, let its term vector be $\boldsymbol{t}_i$.

Our goal is to generate a simplified, edge-sampled graph $\mathcal{G}_{sample} = (\mathcal{V}, \mathcal{E}_{sample})$ and then use $\mathcal{G}_{sample}$ to find communities with coherent content and link structure. $\mathcal{G}_{sample}$ should possess the following properties:

- $\mathcal{G}_{sample}$ has the same vertex set as $\mathcal{G}_t$. That is, no node in the network is added or removed during the simplification process.

- $|\mathcal{E}_{sample}| \ll |\mathcal{E}_t|$, as this enables both better runtime performance and lower memory usage in the subsequent clustering stage.

- Informally put, the resultant edge set $\mathcal{E}_{sample}$ would connect node pairs which are both structure-wise and content-wise similar. As a result, it is possible for our method to add edges which were absent from $\mathcal{E}_t$ since the content similarity was overlooked.

2.2.1 Key Intuitions

The main steps of the CODICIL algorithm are:

1. Create content edges.

2. Sample the union of content edges and topological edges with bias, retaining only edges that are relevant in local neighborhoods.

3. Partition the simplified graph into clusters.

The constructed content graph and simplified graph have the same vertices as the input graph (vertices are never added or removed), so the essential operations of the algorithm are constructing, combining edges and then sampling with bias. Figure 2.1 illustrates the work flow of CODICIL.

From the term vectors $\mathcal{T}$, content edges $\mathcal{E}_c$ are constructed. Those content edges and the input topological edges $\mathcal{E}_t$ are combined as $\mathcal{E}_u$, which is then sampled with bias to form a smaller edge set $\mathcal{E}_{sample}$ where the most relevant edges are preserved. The

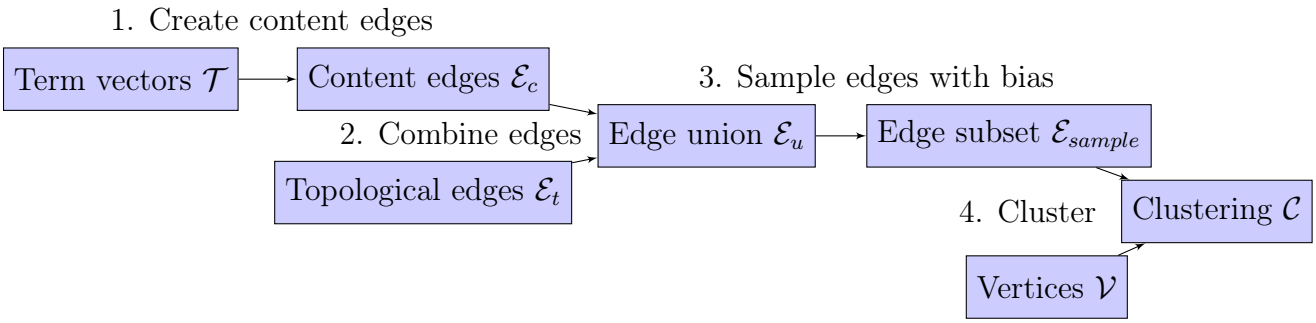


Figure 2.1: Work flow of CODICIL

graph composed of these sampled edges is passed to the graph clustering algorithm which partitions the vertices into a given number of clusters.

2.2.2 Basic Framework

The pseudo-code of CODICIL is given in Algorithm 1.

CODICIL takes as input (1) $\mathcal{G}_t$, the original graph consisting of vertices V, edges $\mathcal{E}_t$ and term vectors $\mathcal{T}$ where $\boldsymbol{t}_i$ is the content term vector for vertex v_i, $1 \leq i \leq |\mathcal{V}| = |\mathcal{T}|$, (2) k, the number of nearest content neighbors to find for each vertex, (3) $normalize(\boldsymbol{x})$, a function that normalizes a vector $\boldsymbol{x}$, (4) α, an optional parameter that specifies the weights of topology and content similarities, (5) l, the number of output clusters desired, (6) $clusteralgo(\mathcal{G}, l)$, an algorithm that partitions a graph $\mathcal{G}$ into l clusters, and (7) $similarity(\boldsymbol{x}, \boldsymbol{y})$ to compute similarity between $\boldsymbol{x}$ and $\boldsymbol{y}$. Note that any content-insensitive graph clustering algorithm can be plugged in the CODICIL framework, providing great flexibility for applications.

Algorithm 1 COmmunity Discovery Inferred from Content Information and Link-structure (CODICIL)

Require: $\mathcal{G}_t = (\mathcal{V}, \mathcal{E}_t, \mathcal{T})$, k, $normalize(\cdot)$, $\alpha \in [0,1]$, l, $clusteralgo(\cdot,\cdot)$, $similarity(\cdot,\cdot)$

return $\mathcal{C}$ (a disjoint clustering of $\mathcal{V}$)
{Create content edges $\mathcal{E}_c$}
$\mathcal{E}_c \leftarrow \emptyset$
for $i = 1$ to $|\mathcal{V}|$ **do**
  for all $v_j \in TopK(v_i, k, \mathcal{T})$ **do**
    $\mathcal{E}_c \leftarrow \mathcal{E}_c \cup (v_i, v_j)$
  end for
end for
{Combine $\mathcal{E}_t$ and $\mathcal{E}_c$. Retain edges with a bias towards locally relevant ones}
$\mathcal{E}_u \leftarrow \mathcal{E}_t \cup \mathcal{E}_c$
$\mathcal{E}_{sample} \leftarrow \emptyset$
for $i = 1$ to $|\mathcal{V}|$ **do**
  {Γ_i contains v_i's neighbors in the edge union}
  $\Gamma_i \leftarrow ngbr(v_i, \mathcal{E}_u)$
  for $j = 1$ to $|\Gamma_i|$ **do** $\boldsymbol{sim^t}_{ij} \leftarrow similarity(ngbr(v_i, \mathcal{E}_t), ngbr(\gamma_j, \mathcal{E}_t))$
  $\boldsymbol{simnorm^t}_i \leftarrow normalize(\boldsymbol{sim^t}_i)$
  for $j = 1$ to $|\Gamma_i|$ **do** $\boldsymbol{sim^c}_{ij} \leftarrow similarity(\boldsymbol{t}_i, \boldsymbol{t}_{\gamma_j})$
  $\boldsymbol{simnorm^c}_i \leftarrow normalize(\boldsymbol{sim^c}_i)$
  for $j = 1$ to $|\Gamma_i|$ **do** $\boldsymbol{sim}_{ij} \leftarrow \alpha \cdot \boldsymbol{simnorm^t}_{ij} + (1-\alpha) \cdot \boldsymbol{simnorm^c}_{ij}$
  {Sort similarity values in descending order. Store the corresponding node IDs in idx_i}
  $[val_i, idx_i] \leftarrow descsort(\boldsymbol{sim}_i)$
  for $j = 1$ to $\left\lceil \sqrt{|\Gamma_i|} \right\rceil$ **do**
    $\mathcal{E}_{sample} \leftarrow \mathcal{E}_{sample} \cup (v_i, v_{idx_{ij}})$
  end for
end for
$\mathcal{G}_{sample} \leftarrow (\mathcal{V}, \mathcal{E}_{sample})$
$\mathcal{C} \leftarrow clusteralgo(\mathcal{G}_{sample}, l)$ {Partition into l clusters}
return $\mathcal{C}$

Creating Content Edges

Lines 2 through 7 detail how content edges are created. For each vertex v_i, its k most content-similar neighbors are computed[5]. For each of v_i's top-k neighbors v_j, an edge (v_i, v_j) is added to content edges $\mathcal{E}_c$. In our experiments we implemented the $TopK$ sub-routine by calculating the cosine similarity of $\boldsymbol{t}_i$'s TF-IDF vector and each other term vector's TF-IDF vector. For a content unit c, its TF-IDF value in a term vector $\boldsymbol{t}_i$ is computed as

$$tf\text{-}idf(c, \boldsymbol{t}_i) = \sqrt{tf(c, \boldsymbol{t}_i)} \cdot \log\left(1 + \frac{|\mathcal{T}|}{\sum_{j=1}^{|\mathcal{T}|} tf(c, \boldsymbol{t}_j)}\right) \ . \tag{2.1}$$

The cosine similarity of two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ is

$$cosine(\boldsymbol{x}, \boldsymbol{y}) = \frac{\boldsymbol{x} \cdot \boldsymbol{y}}{\|\boldsymbol{x}\|_2 \cdot \|\boldsymbol{y}\|_2} \ . \tag{2.2}$$

The k vertices corresponding to the k highest TF-IDF vector cosine similarity values with v_i are selected as the top-k neighbors of v_i.

Local Ranking of Edges and Graph Simplification

Line 9 takes the union of the newly-created content edge set $\mathcal{E}_c$ and the original topological edge set $\mathcal{E}_t$. In lines 10 through 24, a sampled edge set $\mathcal{E}_{sample}$ is constructed by retaining the most relevant edges from the edge union $\mathcal{E}_u$. For each vertex v_i, the edges to retain are selected from its local neighborhood in $\mathcal{E}_u$ (line 13). We compute the topological similarity (line 14) between node v_i and its neighbor γ_j as the relative overlap of their respective topological neighbor sets, $I = ngbr(v_i, \mathcal{E}_t)$

[5]Besides top-k criteria, we also investigated using all-pairs similarity above a given global threshold, but this tended to produce highly imbalanced degree distributions.

and $J = ngbr(\gamma_j, \mathcal{E}_t)$, using *similarity* (either cosine similarity as in Equation 2.2 or Jaccard coefficient as defined below):

$$jaccard(I, J) = \frac{|I \cap J|}{|I \cup J|} \ . \tag{2.3}$$

After the computation of the topological similarity vector $\boldsymbol{sim^t}_i$ finishes, it is normalized by *normalize* (line 15). In our experiments we implemented *normalize* with either *zero-one*, which simply rescales the vector to $[0, 1]$:

$$zero\text{-}one(\vec{x}) = (x_i - min(\vec{x}))/(max(\vec{x}) - min(\vec{x})) \ , \tag{2.4}$$

or *z-norm*[6], which centers and normalizes values to zero mean and unit variance:

$$z\text{-}norm(\vec{x}) = \frac{x_i - \hat{\mu}}{\hat{\sigma}}, \hat{\mu} = \frac{\sum_{i=1}^{|\vec{x}|} x_i}{|\vec{x}|}, \hat{\sigma}^2 = \frac{1}{|\vec{x}| - 1} \sum_{i=1}^{|\vec{x}|} (x_i - \hat{\mu})^2 \ . \tag{2.5}$$

Likewise, we compute v_i's content similarity to its neighbor γ_j by applying *similarity* on term vectors $\boldsymbol{t}_i$ and $\boldsymbol{t}_{\gamma_j}$ and normalize those similarities (lines 16 and 17). The topological and content similarities of each edge are then aggregated with the weight specified by α (line 18).

In lines 20 through 23, the edges with highest similarity values are retained. As stated in our desiderata, we want $|\mathcal{E}_{sample}| \ll |\mathcal{E}_t|$ and therefore need to retain fewer than $|\Gamma_i|$ edges. Inspired by [122], we choose to keep $\lceil \sqrt{|\Gamma_i|} \rceil$ edges. This form has the following properties: 1) every vertex v_i will be incident to at least one edge, therefore the sparsification process does not generate new singletons, 2) concavity and monotonicity ensure that larger-degree vertices will retain no fewer edges than

[6]Montague and Aslam [96] pointed out that *z-norm* has the advantage of being both shift and scale invariant as well as outlier insensitive. They experimentally found it best among six simple combination schemes discussed in [48].

smaller-degree vertices, and 3) sublinearity ensures that smaller-degree vertices will have a larger fraction of their edges retained than larger-degree vertices.

Partitioning the Sampled Graph

Finally in lines 25 through 27 the sampled graph $\mathcal{G}_{sample}$ is formed with the retained edges, and the graph clustering algorithm $clusteralgo$ partitions $\mathcal{G}_{sample}$ into l clusters.

Extension to Support Complex Graphs

The proposed CODICIL framework can also be easily extended to support community detection from other types of graph. If an input graph has weighted edges, we can modify the formula in line 18 so that sim_{ij} becomes the product of combined similarity and original edge weight. Support of attribute graph is also straightforward, as attribute assignment of a node can be represented by an indicator vector, which is in the same form of a text vector.

2.2.3 Key Speedup Optimization

***TopK* Implementation**

When computing cosine similarities across term vectors $\boldsymbol{t}_1, \ldots, \boldsymbol{t}_{|\mathcal{T}|}$, one can truncate the TF-IDF vectors by only keeping m elements with the highest TF-IDF values and set other elements to 0. When m is set to a small value, TF-IDF vectors are sparser and therefore the similarity calculation becomes more efficient with little loss in accuracy.

We may also be interested in constraining content edges to be within a topological neighborhood of each node v_i, such that the search space of $TopK$ algorithm can be greatly reduced. Two straightforward choices are (1) *1-hop* graph in which the content

edges from v_i are restricted to be in v_i's direct topological neighborhood, and (2) *2-hop* graph in which content edges can connect v_i and its neighbors' neighbors.

Many contemporary text search systems make use of inverted indices to speed up the operation of finding the k term vectors (documents) with the largest values of Equation 2.2 given a query vector $\boldsymbol{t}_i$. We used the implementation from Apache Lucene for the largest dataset.

Fast Jaccard Similarity Estimation

To avoid expensive computation of the exact Jaccard similarity, we estimate it by using minwise hashing [17]. An unbiased estimator of sets A and B's Jaccard similarity can be obtained by

$$\hat{jaccard}(A, B) = \frac{1}{h} \sum_{i=1}^{h} I(\min(\pi_i(A)) = \min(\pi_i(B))) \ , \tag{2.6}$$

where $\pi_1, \pi_2, \cdots, \pi_h$ are h permutations drawn randomly from a family of minwise independent permutations defined on the universe A and B belong to, and I is the identity function. After hashing each element once using each permutation, the cost for similarity estimation is only $O(h)$ where h is usually chosen to be less than $|A|$ and $|B|$.

Fast Cosine Similarity Estimation

Similar to Jaccard coefficient, we can apply random projection method for fast estimate of cosine similarity [22]. In this method, each hash signature for a d-dimensional vector $\boldsymbol{x}$ is $h(\boldsymbol{x}) = \operatorname{sgn}(\boldsymbol{x}, \boldsymbol{r})$, where $\boldsymbol{r} \in \{0, 1\}^d$ is drawn randomly. For two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$, the following holds:

$$Pr[h(\boldsymbol{x}) = h(\boldsymbol{y})] = 1 - \frac{\arccos\left(cosine(\boldsymbol{x}, \boldsymbol{y})\right)}{\pi} \quad . \tag{2.7}$$

2.2.4 Performance Analysis

Lines 3–7 of CODICIL are a preprocessing step which compute for each vertex its top-k most similar vertices. Results of this one-time computation can be reused for any $k' \leq k$. Its complexity depends on the implementation of the $TopK$ operation. On our largest dataset Wikipedia this step completed within a few hours.

We now consider the loop in lines 11–24 where CODICIL loops through each vertex. For lines 14 and 16 we use the Jaccard estimator from Section 2.2.3 for which runs in $O(h)$ with a constant number of hashes h. The normalization operations in lines 15 and 17 are $O(|\Gamma_i|)$ and the inner loop in lines 21–23 is $O(\sqrt{|\Gamma_i|})$. Sorting edges by weight in line 20 is $O(|\Gamma_i| \log |\Gamma_i|)$. The size of Γ_i, the union of topology and content neighbors, is at most n but on average much smaller in real world graphs. Thus the loop in lines 11–24 runs in $O(n^2 \log n)$.

The overall runtime of CODICIL is the edge preprocessing time, plus $O(n^2 \log n)$ for the loop, plus the algorithm-dependent time taken by *clusteralgo*.

2.3 Experiments

We are interested in empirically answering the following questions:

- **Do the proposed content-aware clustering methods lead to better clustering than using graph topology only?**
- **How do our methods compare to existing content-aware clustering methods?**

- **How scalable are our methods when the data size grows?**

2.3.1 Datasets

Three publicly-available datasets with varying scale and characteristic are used. Their domains cover document network as well as social network. Each dataset is described below, and Table 2.1 follows, listing basic statistics of them.

CiteSeer

A citation network of computer science publications[7], each of which labeled as one of six sub-fields. In our graph, nodes stand for publications and undirected edges indicate citation relationships. The content information is stemmed words from research papers, represented as one binary vector for each document. Observe that the density of this network (average degree 2.74) is significantly lower than normally expected for a citation network.

Wikipedia

The static dump of English Wikipedia pages (October 2011). Only regular pages belonging to at least one category are included, each of which becomes one node. Page links are extracted. Cleaned bi-grams from title and text are used to represent each document's content. We use categories that a page belongs to as the page's class labels. Note that a page can be contained in more than one category, thus ground truth categories are overlapping.

[7] `http://www.cs.umd.edu/projects/linqs/projects/lbc/index.html`

	$\lvert\mathcal{V}\rvert$	$\lvert\mathcal{E}_t\rvert$	# CC	$\lvert\text{CC}_{\max}\rvert$	$Dim_{\mathcal{T}}$	Avg $\lvert\boldsymbol{t}_i\rvert$	# Class
Wikipedia	3,580,013	162,085,383	10	3,579,995	1,459,335	202	595,355
Flickr	16,710	716,063	4	16,704	1,156	44	184,334
CiteSeer	3,312	4,536	438	2,110	3,703	32	6

Table 2.1: Basic statistics of datasets. # CC: number of connected components. $|\text{CC}_{\max}|$: size of the largest connected component. $Dim_{\mathcal{T}}$: number of unique content units. Avg $|\boldsymbol{t}_i|$: average number of non-zero elements in term vectors. # Class: number of (overlapping) ground truth classes.

Flickr

From a dataset of tagged photos[8] we removed infrequent tags and users associated with only few tags. Each graph node stands for a user, and an edge exists if one user is in another's contact list. Tags that users added to uploaded photos are used as content information. Flickr user groups are collected as ground truth. Similar to Wikipedia categories, Flickr user groups are also overlapping.

2.3.2 Baseline Methods

In terms of strawman methods, we compare the CODICIL methods with three existing content-aware graph clustering algorithms, SA-Cluster-Inc [153], PCL-DC [149] and Link-PLSA-LDA (L-P-LDA) [100]. Their methodologies have been briefly introduced in Section 2.1. When applying SA-Cluster-Inc, we treat each term in $\mathcal{T}$ as a binary-valued attribute, i.e. for each graph node i every attribute value indicates whether the corresponding term is present in $\boldsymbol{t}_i$ or not. For L-P-LDA, since it does not assume a distinct distribution over topics for each cited document individually, only citing documents' topic distributions are estimated. As a result, there are 2313

[8] `http://staff.science.uva.nl/~xirong/index.php?n=DataSet.Flickr3m`

citing documents in CiteSeer dataset and we report the F-score on those documents using their corresponding ground-truth assignments.

Previously SA-Cluster-Inc has been shown to outperform k-SNAP [131] and PCL-DC to outperform methods including PLSA-PHITS [30], LDA-Link-Word [40] and Link-Content-Factorization [154]. Therefore we do not compare with those algorithms.

Two content-insensitive clustering algorithms are included in the experiments as well. The first method, "Original Topo", clusters the original network directly. The second method samples edges solely based on structural similarity and then clusters the sampled graph [122], and we refer to it as "Sampled Topo" hereafter.

Finally, we also adapt LDA and K-means[9] algorithm to cluster graph nodes using content information only. When applying LDA, we treat each term vector $\boldsymbol{t}_i$ as a document, and one product of LDA's estimation procedure is the distribution over latent topics, $\boldsymbol{\theta}_{\boldsymbol{t}_i}$, for each $\boldsymbol{t}_i$ (more details can be found at the original paper by Blei et al. [15]). Therefore, we treat each latent topic as a cluster and assign each graph node to the cluster that corresponds to the topic of largest probability. We use GibbsLDA++[10], a C++ implementation of LDA using Gibbs sampling [56] which is faster than the variational method proposed originally. Results of this method are denoted as "LDA".

[9]We do not report running time of K-means as it is not implemented in C or C++.

[10]`http://gibbslda.sourceforge.net/`

2.3.3 Experiment Setup

Parameter Selection

There are several tunable parameters in the CODICIL framework, first of which is k, the number of content neighbors in the $TopK$ sub-routine. We propose the following heuristic to decide a proper value for k: the value of k should let $|\mathcal{E}_c| \approx |\mathcal{E}_t|$. As a result, k is set to 50 for both Wikipedia ($|\mathcal{E}_c| = 150,955,014$) and Flickr ($|\mathcal{E}_c| = 722,928$). For CiteSeer, we experiment with two relatively higher k values (50, $|\mathcal{E}_c| = 103,080$ and 70, $|\mathcal{E}_c| = 143,575$) in order to compensate the extreme sparsity in the original network. Though simplistic, this heuristic leads to decent clustering quality, as shown in Section 2.3.5, and avoids extra effort for tuning.

Another parameter of interest is α, which determines the weights for structural and content similarities. We set α to 0.5 unless otherwise specified, as in Section 2.3.7. The number of hashes (h) used for minwise hashing (Jaccard coefficient) is 30, and 512 for random projection (cosine similarity). Experiments with both choices of *similarity* function are performed. As for m, the number of non-zero elements in term vectors, we let $m = 10$ for Wikipedia and Flickr. This optional step is omitted for CiteSeer since the speedup is insignificant.

Clustering Algorithm

We combine the CODICIL framework with two different clustering algorithms, Metis[11] [71] and Multi-level Regularized Markov Clustering (MLR-MCL)[12] [120]. Both clustering algorithms are also applied on strawman methods.

[11] `http://glaros.dtc.umn.edu/gkhome/metis/metis/download`

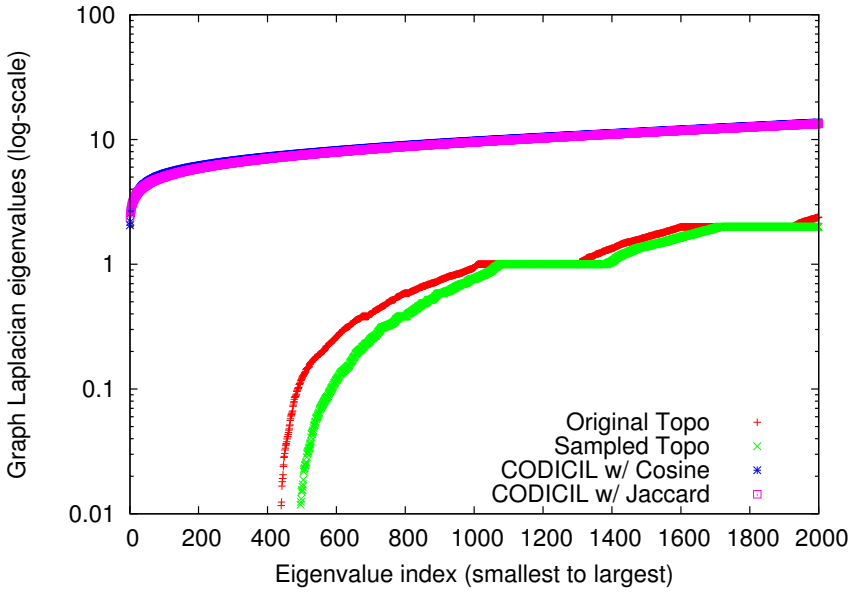
[12] `http://www.cse.ohio-state.edu/~satuluri/research.html`

2.3.4 Effect of Simplification on Graph Structure

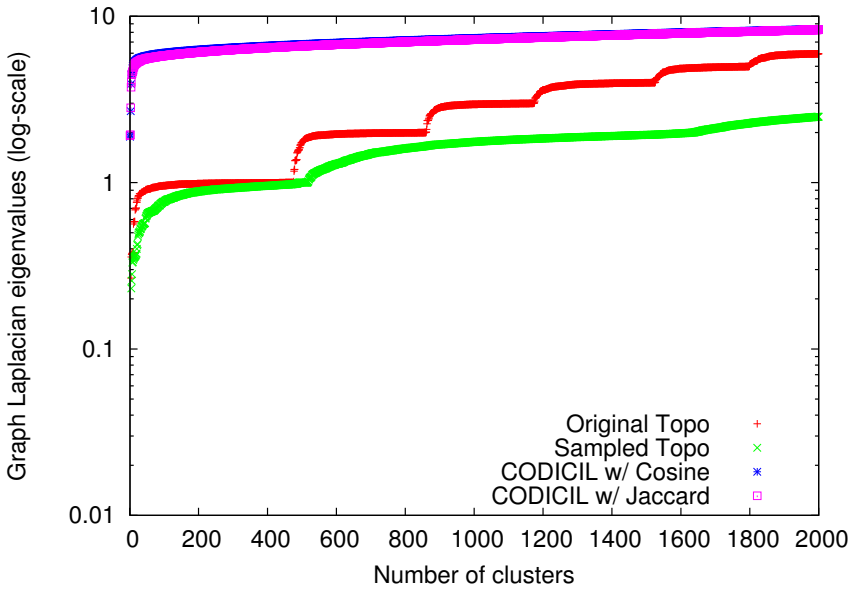
In this section we investigate the impact of simplification on the spectrum of the graph. For both CiteSeer and Flickr (results for Wikipedia are similar to that of Flickr) we compute the Laplacian of the graph, before and after three different types of simplification (topology-only, CODICIL with cosine similarity, CODICIL with Jaccard similarity), and examine the top part (the smallest 2000 eigenvalues) of each eigenspectrum. Specifically, in Figure 2.2 we order the top parts in the ascending order and plot their values.

The multiplicity of 0 as an eigenvalue in such a plot corresponds to the number of independent components within the graph [95]. For CiteSeer we see an increase in the number of components after topological simplification (green versus red marks) and it further disconnects the network, whereas for Flickr (similarly for Wikipedia) the number of components is unchanged. Our hypothesis is that for sparse datasets like CiteSeer topology-only simplification will have a negative impact on the quality of the resulting clustering, and our content-based enhancements will help in overcoming this shortfall. This is reflected on the plots, as there are fewer zero eigenvalues after simplification using CODICIL (purple and blue marks).

Note that the sum of eigenvalues for the complete spectrum is proportional to the number of edges in the graph [95], so this explains why the plots for the original graphs are slightly above those for the simplified graph even though the overall trends (e.g. spectral gap, relative changes in eigenvalues) are quite similar for both datasets. On the other hand, the plots of graphs after CODICIL are further above those of the original graphs, signifying its ability to recover links that were missed in the original networks.



(a) Citeseer



(b) Flickr

Figure 2.2: First 2000 eigenvalues of graph Laplacian before and after simplification

2.3.5 Clustering Quality

We are interested in comparison between the predicted clustering and the real community structure since group/category information is available for all three datasets. Later in Section 2.4 we will evaluate CODICIL's performance qualitatively. While it is tempting to use conductance or other cut-based objectives to evaluate the quality of clustering, they only value the structural cohesiveness but not the content cohesiveness of resultant clustering, which is exactly the motivation of content-aware clustering algorithm. Instead, we use average F-score with regard to the ground truth as the clustering quality measure, as it takes content grouping into consideration and ensures a fair comparison among different clusterings. Given a predicted cluster p and with reference to a ground truth cluster g (both in the form of node set), we define the precision rate as $\frac{|p \cap g|}{|p|}$ and the recall rate as $\frac{|p \cap g|}{|g|}$. The F-score of p on g, denoted as $F(p, g)$, is the harmonic mean of precision and recall rates.

For a predicted cluster p, we compute its F-score on each g in the ground truth clustering G and define the maximal obtained as p's F-score on G. That is:

$$F(p, G) = \max_{g \in G} F(p, g) \ . \tag{2.8}$$

The final F-score of the predicted clustering P on the ground truth clustering G is then calculated as the weighted (by cluster size) average of each predicted cluster's F-score:

$$F(P, G) = \sum_{p \in P} \frac{|p|}{|\mathcal{V}|} F(p, G) \ . \tag{2.9}$$

This effectively penalizes the predicted clustering that is not well-aligned with the ground truth, and we use it as the quality measure of all methods on all datasets.

CiteSeer

In Figure 2.3 we show the experiment results on CiteSeer. Since it is known that the network has six communities (i.e. sub-fields in computer science), there is no need to vary l, the number of desired clusters. We report results using Metis (similar numbers were observed with Markov clustering). For PCL-DC, we set the parameter λ to 5 as suggested in the original paper, yielding an F-score of 0.570. The F-scores of SA-Cluster-Inc and L-P-LDA are 0.348 and 0.458, respectively. As we can see clearly in the bar chart, clustering based on topology alone results in a performance well below the state-of-the-art content-aware clustering methods. This is not surprising as the input graph has 438 connected components and therefore most small components were randomly assigned a prediction label. Although such approach has no impact on topology-based measures (e.g. normalized cut or conductance), it greatly spoils the F-score measure against the ground truth. Moreover, topology-based simplification further deteriorates the clustering performance as it creates even more connected components, as we projected in Section 2.3.4. Neither is LDA able to provide a competitive result, as it is oblivious to link structure embedded in the dataset. Surprisingly though, K-means only manages to produce a very unbalanced clustering (the largest cluster always contains more than 90% of all papers) even after 50 iterations, and its F-score (averaged over five runs) is only 0.336.

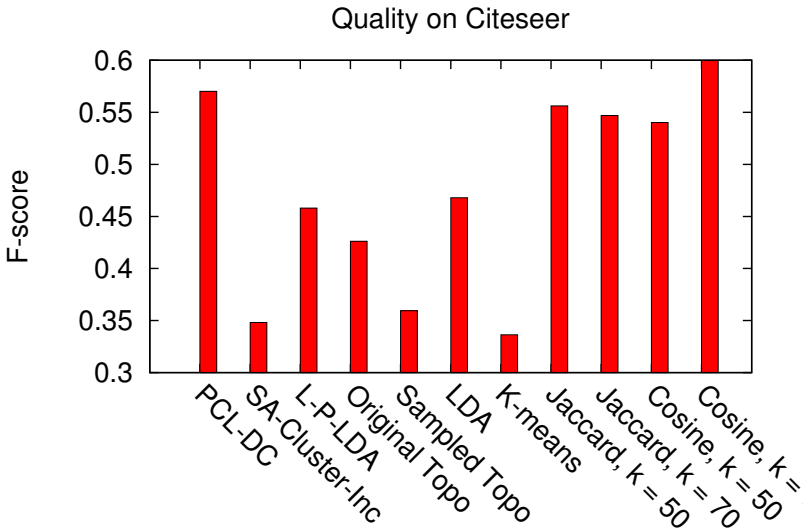


Figure 2.3: F-score of Metis on CiteSeer

On the other hand, our content-aware approaches (using Metis as the clustering method) were able to handle the issue of disconnection as they also include content-similar edges. For both similarity measures, the F-scores are within 90% range of PCL-DC, and it outperforms PCL-DC when k increases to 70.

While achieving the quality that is comparable with existing methods, the COD-ICIL series are significantly faster. PCL-DC takes 234 seconds on this dataset and SA-Cluster-Inc requires 306 seconds. LDA finishes in 40 seconds. In contrast, the sum of CODICIL's edge sampling and clustering time never exceeds 1 second. Therefore, the CODICIL methods are at least one order of magnitude faster than state-of-the-art algorithms.

Wikipedia

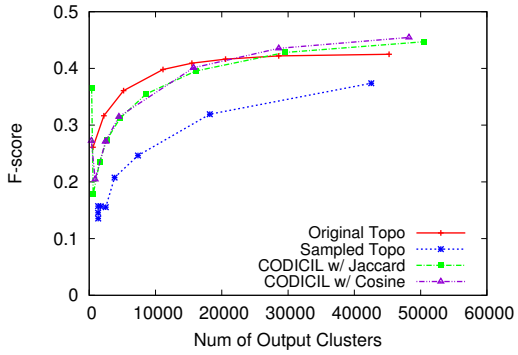
For the Wikipedia dataset, we were unable to run the experiment on SA-Cluster-Inc, PCL-DC, L-P-LDA, LDA and K-means as their memory and/or running time requirement became prohibitive on this million-node network. For example, storing 10,000 centroids alone in K-means requires 54 GBs).

Figures 2.4a and 2.4c plot the performances using MLR-MCL and Metis, respectively. Since category assignments as the ground truth are overlapping, there is no gold standard for the number of clusters. We therefore varied l in both clustering algorithms. Our content-aware clustering algorithms consistently outperform Sampled Topo by a large margin, indicating that CODICIL methods are able to simplify the network and recover community structure at the same time. CODICIL methods' F-scores are also on par or better than those of Original Topo.

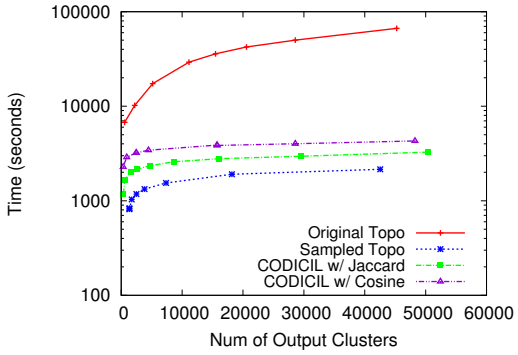
Flickr

Figure 2.5a shows the performances of various methods with MLR-MCL on Flickr, where SA-Cluster-Inc, PCL-DC, LDA and K-means can also finish in a reasonable time (L-P-LDA still takes more than 30 hours). Again, l was varied for the clustering algorithm. Similar to results on CiteSeer, CODICIL methods again lead the baselines by a considerable margin. The F-scores of SA-Cluster-Inc, LDA, and K-means never exceed 0.2, whereas CODICIL methods' F-scores are often higher, together with Original & Sampled Topo.

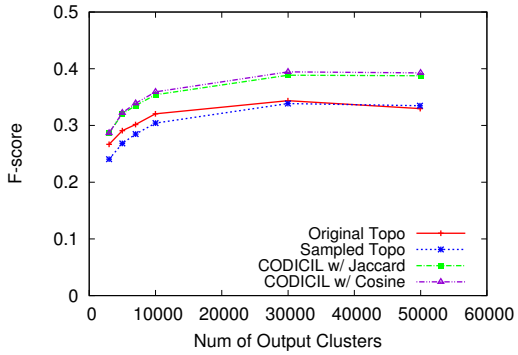
Readers may have noticed that for PCL-DC only three data points ($l = 50, 75, 100$) are obtained. That is because its excessive memory consumption crashed our workstation after using up 16 GBs of RAM for larger l values. We also observe that while



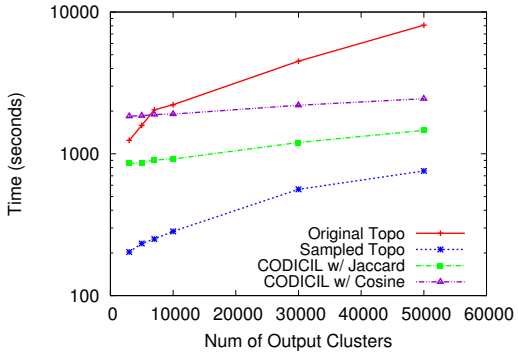
(a) F-score of MLR-MCL on Wikipedia



(b) Running time of MLR-MCL on Wikipedia

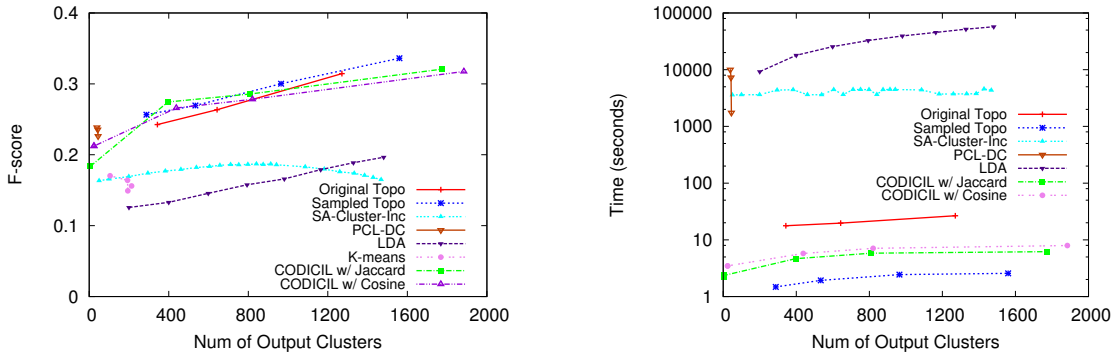


(c) F-score of Metis on Wikipedia



(d) Running time of Metis on Wikipedia

Figure 2.4: Experiment results on Wikipedia



(a) F-score of MLR-MCL on Flickr

(b) Running time of MLR-MCL on Flickr

Figure 2.5: Experiment results on Flickr

PCL-DC generates a group membership distribution over l groups for each vertex, fewer than l communities are discovered. That is, there exist groups of which no vertex is a prominent member. Furthermore, the number of communities discovered is decreasing as l increases (45, 43 and 39 communities for $l = 50, 75, 100$), which is opposite to other methods' trends. All three clusterings' F-scores are less than 0.25. Similarly, multiple runs of K-means (K is set to 400, 800, 1200, and 1600) can only identity roughly 200 communities.

2.3.6 Scalability

The running time on CiteSeer has already been discussed, and here we focus on Flickr and Wikipedia. For CODICIL methods, the running time includes both edge sampling and clustering stage. The plots' Y-axes (running time) are in log scale.

Flickr

We first report scalability results on Flickr (see Figure 2.5b). For SA-Cluster-Inc, the value of l (the desired output cluster count), ranging from 100 to 5000,

does not affect its running time as it always stays between 1 and 1.25 hours with memory usage around 12GB. The running time of LDA appears, to a large extent, linear in the number of latent topics (i.e. l) specified, climbing up from 2.56 hours ($l = 200$) to 15.88 hours ($l = 1600$). For PCL-DC, the running time with three l values $(50, 75, 100)$ is 0.5, 2.0 and 2.8 hours, respectively.

As for our content-aware clustering algorithms, running them on Flickr requires less than 8 seconds, which is three to four orders of magnitude faster than SA-Cluster-Inc, PCL-DC and LDA. Original Topo takes more than 10 seconds, and Sampled Topo runs slightly faster than CODICIL methods.

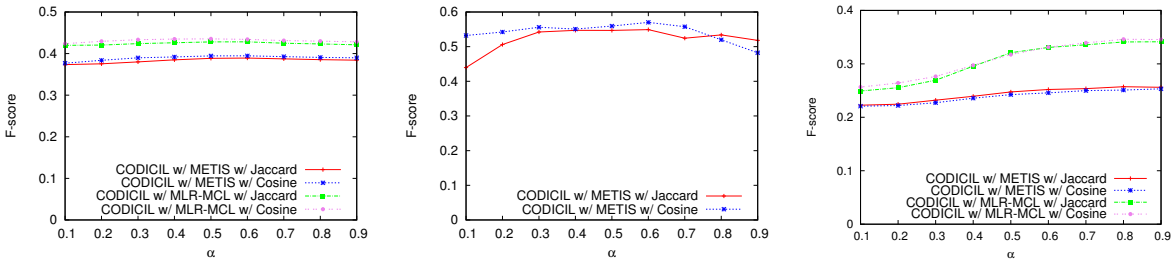
Wikipedia

Original Topo, Sampled Topo and all CODICIL methods finished successfully. The running time is plotted in Figures 2.4b and 2.4d. When clustering using MLR-MCL, our methods are at least one order of magnitude faster than clustering based on network topology alone. For Metis, CODICIL is also more than four times faster. The trend lines suggest our methods have promising scalability for analysis on even larger networks.

2.3.7 Effect of Varying α on F-score

So far all experiments performed fix α at 0.5, meaning equal weights of structural and content similarities. In this sub-section we track how the clustering quality changes when the value of α is varied from 0.1 to 0.9 with a step length of 0.1.

On Wikipedia (Figure 2.6a) and Citeseer (Figure 2.6b), F-scores are greatest around $\alpha = 0.5$, supporting the decision of assigning equal weights to structural and



(a) Varying α on Wikipedia (b) Varying α on Citeseer (c) Varying α on Flickr

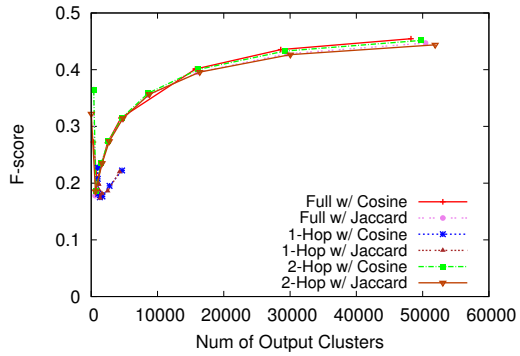
Figure 2.6: Effect of varying α on F-score (avg. # clusters for Wikipedia: 29,414, avg. # clusters for Flickr: 1,911)

content similarities. Results differ on Flickr where F-score is constantly improving when α increases (i.e. more weight assigned to topological similarity).

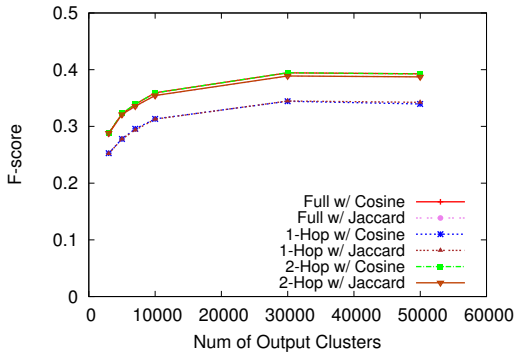
2.3.8 Effect of $\mathcal{E}_c$ Constraint on F-score

In Section 2.2.3 we discuss the possibility of constraining content edges within a topological neighborhood for each node v_i. Here we provide a brief review on how the qualities of resultant clusterings are impacted by such constraint. For the sake of space, we focus on the F-scores on Wikipedia and Flickr.

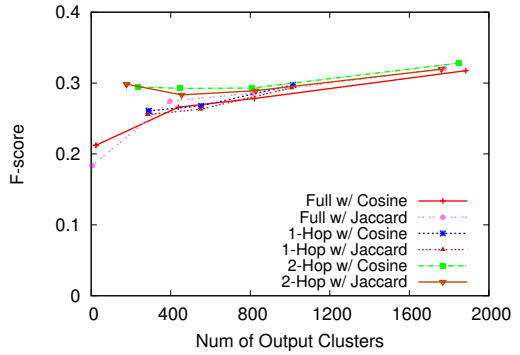
Figures 2.7a and 2.7b show F-scores achieved on Wikipedia, using different $\mathcal{E}_c$ constraints. *Full* means no constraint and $TopK$ sub-routine searches the whole vertex set $\mathcal{V}$, whereas *1-hop* constrains the search to within a one-hop neighborhood, and likewise for *2-hop*. Plots of *full* and *2-hop* almost overlap with each other, suggesting that searching within the 2-hop neighborhood can provide sufficiently strong content signals on this dataset. For Flickr (Figures 2.7c and 2.7d), interestingly *2-hop* and



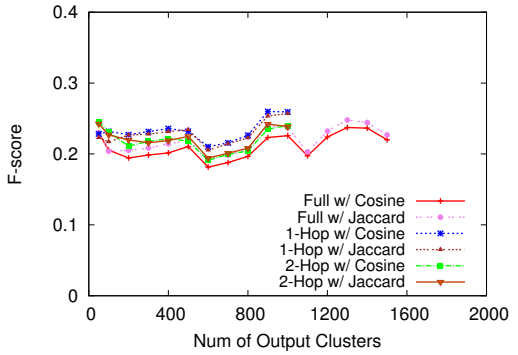
(a) Varying $\mathcal{E}_c$ constraint, MLR-MCL on Wikipedia



(b) Varying $\mathcal{E}_c$ constraint, METIS on Wikipedia



(c) Varying $\mathcal{E}_c$ constraint, MLR-MCL on Flickr



(d) Varying $\mathcal{E}_c$ constraint, METIS on Flickr

Figure 2.7: Effect of $\mathcal{E}_c$ constraint on F-score

1-hop have a slight lead over *full*. This may be an indication that in online social networks, compared with information networks, content similarity between two closely connected users emits stronger community signals.

2.3.9 Discussions

An interesting observation on the biased edge sampling is that it always results in an improvement in running time. However, sampling just the topology graph results in a clear loss in accuracy whereas content-conscious sampling is much more effective

with accuracies that are on par with the best performing methods at a fraction of the cost to compute. We observe this for all three datasets.

We also find that for probabilistic-model-based methods (PCL-DC, L-P-LDA and LDA) as well as K-means, their running time is at least linear in l, the desired number of output clusters, which becomes a critical drawback in face of large-scale workloads. As the network grows, the number of clusters also increases naturally. Plots on CODICIL methods' running time, on the other hand, suggest a logarithmic increase with regard to the number of clusters, which is more affordable.

2.4 Case Studies

In this section, we demonstrate the benefits of leveraging content information on two Wikipedia pages: "Machine Learning" and "Graph (Mathematics)".

In the original network, "machine learning" has a total degree of 637, and many of its neighbors (including "1-2-AX working memory task", "Wayne State University Computer Science Department", "Chou-Fasman method", etc.) are at best peripheral to the context. When we sample the graph according to its link structure only, 119 neighbors are retained for "machine learning". Although this eliminates some noise, many others, including the three entries above, are still preserved. Moreover, it also removes during the process many neighbors which should have been kept, e.g. "naive Bayes classifier", "support vector machine", and so on.

The CODICIL framework, in contrast, alleviates both problems. Apart from removing noisy edges, it also keeps the most relevant ones. For example, "AdaBoost", "ensemble learning", "pattern recognition" all appear in "machine learning"'s neighborhood in the sampled edge set $\mathcal{E}_{sample}$. Perhaps more interestingly, we find that

CODICIL adds "neural network", an edge absent from the original network, into $\mathcal{E}_{sample}$ (recall that it is possible for CODICIL to include an edge even it is not in the original graph, given its content similarity is sufficiently high). This again illustrates the core philosophy of CODICIL: to complement the original network with content information so as to better recover the community structure.

Similar observations can be made on the "Graph (Mathematics)" page. For example, CODICIL removes entries including "Eric W. Weisstein", "gadget (computer science)" and "interval chromatic number of an ordered graph". It also keeps "clique (graph theory)", "Hamiltonian path", "connectivity (graph theory)" and others, which would otherwise be removed if we sample the graph using link structure alone.

2.5 Conclusion

We have presented an efficient and extremely simple algorithm for community identification in large-scale graphs by fusing content and link similarity. Our algorithm, CODICIL, selectively retains edges of high relevancy within local neighborhoods from the fused graph, and subsequently clusters this backbone graph with any content-agnostic graph clustering algorithm.

Our experiments demonstrate that CODICIL outperforms state-of-the-art methods in clustering quality while running orders of magnitude faster for moderately-sized datasets, and can efficiently handle large graphs with millions of nodes and hundreds of millions of edges. While simplification can be applied to the original topology alone with a small loss of clustering quality, it is particularly potent when combined with content edges, delivering superior clustering quality with excellent runtime performance.

Chapter 3: Simultaneous Detection of Communities and Roles from Large Networks

This chapter covers the second part of my work in efficient graph algorithms for OSN analytics, as I study the problems community detection and structural role detection jointly. Both tasks are essential in the realm of network science, and they have received extensive research interests. Community detection, with its roots in graph partitioning is concerned with the inter-connectivity among nodes, as it aims at identifying groups of nodes that are densely connected compared with their neighbors. Exemplar applications include finding clusters of users from social networks and functional protein complexes from bioinformatics networks. On the other hand, structural role detection focuses on finding sets of nodes (i.e. roles) that share similar structural properties (such as degree, clustering coefficient, and betweenness) and characterizing different roles. Structural roles can often be associated with functions of nodes in a network, and it is a promising approach to characterize OSN users (e.g. in Chapters 5 and 6) using their structural roles. For example, hub nodes with high degree in an epidemic network are more likely to spread diseases, whereas bridge nodes with low degree and high betweenness are gatekeepers and important candidates for immunization. Recent work has leveraged role detection techniques for identity resolution [64, 50], exploratory network analysis [50], and anomaly detection [116].

To date, however, studies on these two topics have been performed independently, and there has been little synergy between them. When an algorithm is performing community (role) detection, it often ignores any role (community) information that is available. In this work we argue that community and structural role discovery should be interdependent and complementary to each other. Real-world communities often contain nodes with various roles for it to function, such as ones that interface with other communities and ones that are peripheral to community cores. On the other hand, the role assignment of a node also depends on the communities it, its neighbors and beyond belong to. Therefore there exists a strong and crucial need to detect communities and roles jointly, and we provide such a method in this paper. As shown in the following sections, the joint discovery of communities and roles can generate communities and roles of higher quality, as compared with identifying them separately.

Problem statement: Given an undirected, unweighted network $G(V, E)$ as the input, our goal is to design an algorithm that outputs both community and structural role assignments for nodes simultaneously. To overcome limitations in prior work, we state the following desiderata:

- **Nonparametric Guidance**: Utilize role information when inferring community assignment, and vice versa, so that assignment information is able to provide guidance to the detection process in a nonparametric fashion.

- **Iterative update**: Improve community and role assignments iteratively, so that the guidance is no longer static and always using the latest assignment information.

- **Overlapping communities and roles**: Generate soft assignments for both community and role, since in many real-world networks nodes naturally belong to multiple communities and share multiple roles, though not uniformly. For example, one researcher can have several research interests, and a star node also acts as a bridge when connecting multiple tight knit communities.

- **Diversity**: Produce heterogeneous role assignment in each community, and vice versa, so that community and role assignments are as diverse from each other as possible.

The last desideratum regarding diversity is because community and role assignments are expected to characterize graph nodes from two different aspects, and thus nodes in the same community are expected to possess diverse roles. To illustrate the validity of this assumption in practice, we studied the composition of roles in several networks that have ground truth community assignments. Specifically, we download three networks (Google Plus, Facebook, and Twitter) from the SNAP network repository[13], and run RolX [64], a role detection algorithm, on them. The number of roles is set to 4, as is automatically determined by RolX. In Google Plus, among all large communities that altogether cover more than 95% of all labeled nodes, 94% of them contain nodes that altogether have at least 2 majority roles, and 48% of them have nodes that altogether have all 4 majority roles. Similar results are found on Facebook and Twitter, where 92% and 62%, respectively, of large communities contain nodes that belong to at least 2 majority roles. This shows that many real world communities indeed have diverse role assignment inherently.

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

Building on those observations and desiderata, we present RC-Joint, our algorithmic solution to the above problem. It treats community detection as a likelihood maximization problem with diversity constraints by role assignment, and it updates role assignment by performing soft clustering of nodes with features derived from community memberships. One iteration of each process is performed alternately, until both community and role assignments converge. This bootstrapping paradigm satisfies all four desiderata, and is therefore able to mine community and role assignments with the up-to-date knowledge of each other. We will describe RC-Joint in details in Section 3.2. An added benefit is that RC-Joint is naturally parallel since inference is done on each node, therefore parallel computing paradigms (such as OpenMP) can be easily leveraged. This fact makes it possible to scale RC-Joint to large networks.

In Section 3.3, we will discuss several optimizations in the implementation of RC-Joint, including parallelism, that yield significant speedup. We also investigate efficient initialization schemes for RC-Joint, which lead to faster execution and often higher accuracy. We demonstrate the efficacy of RC-Joint by experimenting on a wide array of real and synthetic networks (Section 3.4). We compare RC-Joint with state-of-the-art algorithms in both community detection and role detection, including BigClam [148], Markov Clustering [120], Graclus [37], RolX [64] and GLRD [50]. Quality of the output are measured by F-score using ground truth information. Results show that RC-Joint is able to detect communities and roles of higher quality, compared with existing methods. The improvement is up to 15% on real networks and 75% on synthetic networks.

3.1 Related Work

3.1.1 Community Detection

Community detection, with its root in graph clustering and graph partitioning, has been pivotal to network science. A plethora of algorithms have been proposed to address this task over the years, be it heuristic-motivated [71], cut-based [37], modularity-based [29], information theoretic [117], or stochastic flow-driven [120]. To cover all community detection algorithms is beyond the scope of this paper, and interested readers can refer to survey articles such as [46].

Among many challenges faced in the community detection literature, a prominent one is the need to find overlapping communities. That is, community assignment is rather "soft". This desideratum is motivated by the observation on many real-world networks that, by nature, community memberships are not mutually exclusive. Various algorithms have been proposed to address this need [146]. For example, clique percolation method by Palla et al. [102] operates on the assumption that overlapping communities consist of adjacent small cliques. Airoldi et al. [4] extend the standard stochastic block model [124] by letting a node's community indicator vector be drawn from a multinomial distribution, creating the mixed membership stochastic block model.

Another family of methods approach the problem by converting edges in a network to nodes in a new graph (called *line graph*) and then applying regular non-overlapping community detection algorithms to create clusters of new nodes [3]. Since a node in the input network is incident to multiple edges which may in turn be assigned to various clusters in the line graph, it may belong to multiple communities. The line

graph, however, contains significantly more nodes than the original network, making the algorithm too costly for large networks.

Recently, Yang and Leskovec propose an affiliation-based model to handle overlapping communities [148]. Each node has an affiliation score with each community, and the affiliation strength is decided by its value. The probability that an edge exists between two nodes is decided by the nodes' community affiliations. Compared with block models, this approach grants individual nodes more flexibility since the linkage probability is no longer subject to the community-specific values. None of those methods, however, consider the structural roles of individual nodes.

3.1.2 Role Detection

While having a shorter history than community detection, role detection is a field of growing research interest. Here, we focus on *structural* roles in a network, although role has also been used to encompass latent topics in document corpus [92]. Henderson et al. have proposed RolX, a non-negative matrix factorization-based (NMF) approach to decompose a node-feature matrix into node-role and role-feature matrices [64]. They show that RolX is able to find roles with distinct characteristics, and the role representation learned on one network can be transferred to another.

Rossi et al. extend role analyses to the dynamic environment, where a series of network snapshots are available [116]. By performing role detection on each snapshot first and then calculating the transition of roles over snapshots, temporal patterns of nodes are extracted. Here role detection serves to provide high-level features for temporal behavior extraction, and its end applications include anomaly detection and nodal behavior prediction.

Recently, Gilpin et al. study the possibility of supplying extra guidance to role detection in order to incorporate external knowledge or requirements [50]. Their framework, *GLRD*, models role detection as a constrained NMF problem, where the guidance is provided as convex constraints and specified per role. Instead of optimizing matrices as a whole, they opt for an alternating least square formulation to improve the efficiency. Three types of guidance are described: sparsity (role assignment and/or representation being sparse for each role), diversity (role assignment and/or representation being different among roles), and alternative role discovery (role assignment and/or representation being different from a given assignment/representation). There are still two limitations in GLRD: (1) It treats community assignment as static input; (2) The recursive feature extraction scheme [64] it relies on incurs a complexity that is cubic to the number of nodes.

Lastly, one common drawback in existing literature of role detection is the absence of direct quality evaluation on proposed algorithms, possibly due to the lack of network data with ground truth on roles. Therefore previous work is confined to exploratory analyses or transfer learning tasks where roles themselves are utilized as high-level features.

3.2 Algorithm

Key intuitions: We view edges in the network as a result of nodes being affiliated to communities. The stronger two nodes are associated with a same community, the more likely it is to observe an edge between them. Furthermore, nodes in one community have diverse structural roles, thus the assignment vectors of any community and any role ought to be dissimilar. As for a node's role assignment, we consider it to be

dependent on how clique-like the node is as well as how many of the node's neighbors belong to the same community as it does. We will elaborate on the materialization of those intuitions in the following sections.

RC-Joint is designed to be an iterative algorithm that improves community and role assignments alternately. It takes as input a connected, undirected, unweighted graph $G = (V, E)$, the number of communities (N_c) and the number of roles (N_r). The convergence threshold (δ_{comm} and δ_{role}) and maximal number of iterations can also be specified. The output is a community score c_{vi} for each node $v \in V$ and each community $i = 1 \cdots N_c$, and a role score r_{vj} for each $v \in V$ and each role $j = 1 \cdots N_r$. Both community and role scores are non-negative. Table 3.1 lists notations used in the rest of the paper.

$G(V, E)$	Network with the vertex set V and edge set E
N_c	Number of communities to detect
N_r	Number of roles to detect
δ_{comm}	Community assignment convergence threshold
δ_{role}	Role assignment convergence threshold
$\mathbf{C}$	$\|V\|$-by-N_c non-negative matrix of community scores
$\mathbf{c}_{\bullet i}$	Column vector of community scores for community i
$\mathbf{c}_{v\bullet}$	Row vector of community scores for node v
$\mathbf{R}$	$\|V\|$-by-N_r non-negative matrix of role scores
$\mathbf{r}_{\bullet j}$	Column vector of role scores for role j
$\mathbf{r}_{v\bullet}$	Row vector of role scores for node v
Γ_v	Set of nodes adjacent to node v
π	Permutation on the set V (Equation 3.1)
$\mathbf{f}_v$	Feature vector of v for role assignment (Equation 3.3)
β	Softness parameter for role assignment (Equation 3.3)
ϵ	Angular cosine threshold for the diversity constraint (Equation 3.9)

Table 3.1: Table of notations

Algorithm 2 shows the pseudo code of the workflow, and each component will be introduced below. RC-Joint starts by initializing community and role assignments, and they can be either specified by some user-provided configurations (e.g. results from a previous run) or inferred automatically (Sections 3.2.1 and 3.2.2). After that, community and role assignments are updated one after each other iteratively. The algorithm stops when both communities and roles converge, or if the maximal number of iterations has been reached[14]. For the convergence check of community assignment, we impose that the relative improvement on network likelihood is less than δ_{comm}, since its value range is network-dependent. When checking the convergence of roles, we require the maximal change of any role score itself is less than δ_{role}, since role scores are always in the range [0, 1].

Algorithm 2 Workflow of RC-Joint

Require: G, N_c, N_r, δ_{comm}, δ_{role}
1: $\mathbf{C^0} \leftarrow \texttt{InitComm}(G, N_c)$
2: $\mathbf{R^0} \leftarrow \texttt{InitRole}(G, N_r)$
3: $i \leftarrow 1$
4: **while not** ($conv_{comm}$ **and** $conv_{role}$) **and** $i \leq \texttt{MaxIter}$ **do**
5: $\quad \mathbf{C^i} = \texttt{UpdateComm}(G, \mathbf{C^{i-1}}, \mathbf{R^{i-1}}, N_c)$
6: $\quad$ **if** $\frac{\texttt{Likelihood}(G,\mathbf{C^i}) - \texttt{Likelihood}(G,\mathbf{C^{i-1}})}{\texttt{Likelihood}(G,\mathbf{C^{i-1}})} < \delta_{comm}$ **then**
7: $\quad\quad conv_{comm} \leftarrow$ **true** {Communities converge}
8: $\quad$ **end if**
9: $\quad \mathbf{R^i} = \texttt{UpdateComm}(G, \mathbf{R^{i-1}}, \mathbf{C^i}, N_r)$
10: $\quad$ **if** $||R^i - R^{i-1}||_{max} < \delta_{role}$ **then**
11: $\quad\quad conv_{role} \leftarrow$ **true** {Roles converge}
12: $\quad$ **end if**
13: $\quad \texttt{iter} \leftarrow \texttt{iter} + 1$
14: **end while**
15:
16: **return** $\mathbf{C^{i-1}}, \mathbf{R^{i-1}}$

[14]Empirically the algorithm often converges within far fewer iterations.

3.2.1 Initializing Community Assignment

One naive way to initialize community scores of nodes is to randomly assign community labels ($1 \cdots N_c$) to nodes. Though fast, this method does not leverage the network's connectivity information, and it is highly probable that nodes sharing the same initial label are far apart. Another simple approach is to choose several vantage points, and to send their labels via breadth-first traversal. While this guarantees connectivity in each initial community, it does not always capture community structures since high-degree hub nodes will pass a label to a large number of nodes with little inter-connectivity. On the other hand, the initialization scheme should be lightweight, otherwise it defeats the purpose of creating an efficient algorithm. For example, we find empirically that identifying neighborhoods with minimal local conductance [52] runs three orders of magnitude slower than our proposed initialization method below, on the Google Plus network (with 108K nodes and 12M edges).

Our solution (`InitComm`) hinges on the intuition that two nodes are likely to belong to the same community if they share a large number of common neighbors. Therefore, we want to group nodes according to relative amount of neighbors they are sharing with each other, and to treat those groups as initial communities.

One established method to efficiently calculate the proportion of shared neighbors is via min-wise hashing [17]. The adjacency list of a node can be viewed as a set, whose elements are from the universe of V, and we can generate one min-wise hash of the adjacency list by applying π, a permutation of V, on the set and taking the minimal value after the permutation. Let Γ_v be the neighborhood of node v, then its

min-wise hash value under π, $h_\pi(\Gamma_v)$ (or $h_\pi(v)$ for short), is:

$$h_\pi(v) \equiv h_\pi(\Gamma_v) = \min_{u \in \Gamma_v} (\pi(u)) \ , \tag{3.1}$$

where $\pi(u)$ is the value of u after permutation π. A min-wise hash signature of length k for v is generated by randomly drawing k permutations $\pi_1 \cdots \pi_k$ and concatenating the corresponding hash values $h_{\pi_1}(v) \cdots h_{\pi_k}(v)$. The same set of permutations are applied to all adjacency lists to generate the corresponding length-k signature for each node.

Given all min-wise hash signatures, we create a top-down hierarchy of nodes according to signature values. This process will be referred to as *grouping* below. We start with the first hash value ($h_{\pi_1}(v), \ \forall v \in V$), and split nodes into groups such that all nodes in one group have the same hash value. If a group is small enough (we use a size threshold of $\frac{|V|}{N_c}$), all nodes in it are given one initial community label. Otherwise, the group is further split based on the second hash value, and so on. This continues until either all k hash values are used, or no more split is required. After grouping, each node has one and only one initial community label.

If there are more than N_c initial community labels, we merge nodes in the smaller groups to larger groups. To achieve this, we perform a *label propagation* algorithm in the following manner. We rank all groups in the descending order of their sizes, and visit them in sequence. When visiting a group, we assign its group label to the immediate neighborhood of each member node. It is further required that if a node has received any label from its neighbors, it can no longer propagate labels to its neighbors. This makes sure that labels "stay" within the local neighborhood.Label propagation terminates when N_c labels have been successfully propagated, after which

a node can possibly have multiple community labels. For a node v and each label i it has, we let the initial community score $c_{vi}^0 = 1$, otherwise it is 0.

Lemma 1 below shows that after `InitComm`, any node in the network will find some other nodes belonging to the same initial community in close proximity.

Lemma 1. *Given a connected, undirected, unweighted network $G(V, E)$, and `InitComm` is run to produce the initial community score matrix $\mathbf{C}^0$. For any node v and community i such that $c_{vi}^0 = 1$, if there exist a non-empty set of other nodes ϕ_{vi} such that $c_{ui}^0 = 1$, $\forall u \in \phi_{vi}$, then there is at least one node $u \in \phi_{vi}$ whose shortest path distance to v on G is at most 2.*

Proof. There are three different scenarios:

I. v obtains label i after the grouping process, and it has propagated i to its neighbors. Then any node $u \in \Gamma_u$ also has label i, and their shortest path distance is 1.

II. v obtains label i after the grouping process, and it does not propagate i. Since ϕ_{vi} is non-empty, there exists at least one node u that also obtains label i after the grouping process because v does not propagate i. Since v and u are in the same group in the grouping process, $h_{\pi_1}(v) = h_{\pi_1}(u)$. Because any π (including π_1) is a one-to-one self-mapping on V, there is at least one element that exists in the adjacency lists of both v and u, i.e. $\Gamma_v \cap \Gamma_u \neq \emptyset$. Therefore, v and u have at least one common neighbor, and the shortest path distance between them is at most 2.

III. v receives label i from the propagation of one of its neighbors, u. Therefore $v \in \Gamma_u$, and their shortest path distance is 1.

To conclude, for any node v and label i such that $c_{vi}^0 = 1$, if $\phi_{vi} \neq \emptyset$, there always exists a node u such that v and u have the same label and the shortest path distance between them is at most 2. □ □

3.2.2 Initializing Role Assignment

Role detection in RC-Joint is achieved by soft k-means clustering on nodes using various structural features described below. During the initialization stage, we assume no knowledge of communities, and therefore we do not use any feature that is derived from the community assignment. While recursive feature aggregation [65] has been shown to capture richer structural information than local features (e.g. degree) alone, we choose not to use it because its complexity is cubic to the number of nodes. To trade off between feature richness and efficiency, we reuse the min-wise hash signatures created in Section 3.2.1 to effectively approximate the similarity of a node's adjacency list and its neighbors' adjacency lists.

The purpose of using adjacency list similarity as node features is to gauge the distribution of a node's structural similarity with its neighbors. Intuitively, the more similar two nodes' adjacency lists are, the more triangles there are that consist of both nodes. If a node has high similarity with most of its neighbors, then it is more likely to be part of a clique-like substructure. In contrast, a node having low similarity with most of its neighbors resembles a star, and it connects multiple communities.

Assuming the hash signatures for nodes v and u have length k, then according to [17], the following statistic is an unbiased estimator of the Jaccard similarity between Γ_v and Γ_u:

$$\hat{sim}(v,u) \equiv \frac{1}{k}\sum_{n=1}^{k} I[h_{\pi_n}(v) = h_{\pi_n}(u)] \tag{3.2}$$
$$E[\hat{sim}(v,u)] = \frac{|\Gamma_v \cap \Gamma_u|}{|\Gamma_v \cup \Gamma_u|}$$

where $I[\bullet]$ is the identity function. For each node, we use the minimum, maximum and three quartiles of the estimated Jaccard similarity with all neighbors as its features.[15] We also include the logarithm of a node's degree as a feature in order to alleviate the large variance of node degree itself. We note that there exist other definitions of structural similarity that one can possibly employ, such as SimRank [68] and its variants. However, they do not fit our purpose because the costly computation is performed for all pairs of nodes, and we will not be able to reuse hash signatures either.

To assign initial role information to nodes, we randomly choose N_r nodes as centroids of k-means, and calculate a node v's role affiliation r_{vj} with each centroid j using an exponential kernel. Affiliation scores are L1-normalized over all centroid for each node, that is:

$$r_{vj} = \frac{\exp\left(-\beta||\mathbf{f}_v - \mathbf{f}_{s_j}||_2\right)}{\sum_{n=1}^{N_r}\exp\left(-\beta||\mathbf{f}_v - \mathbf{f}_{s_n}||_2\right)} \tag{3.3}$$

where $\mathbf{f}_v$ ($\mathbf{f}_{s_j}$) denotes the feature vector of node v (centroid s_j). The parameter β is used to control the "softness" of the assignment, and a larger β value suppresses minor affiliation scores. In our implementation the default value for β is 1.

[15] We find that $k = 30$ is sufficient for the hash signature length.

3.2.3 Updating Community Assignment

Our goal in updating community assignment is to increase the likelihood of network's edge set E, given the community affiliation of nodes. At the same time, we want the community assignment to be diverse with regard to the role assignment by imposing the requirement of diversity in any pair of community and role.

Formally, the goal can be expressed as a constrained optimization problem:

$$\max_{\mathbf{C}} \left(\texttt{Likelihood}(G, \mathbf{C})\right) \tag{3.4}$$

$$\text{subject to } \mathbf{c}_{\bullet i} \cdot \mathbf{r}_{\bullet j} < \epsilon_{ij}, \forall i \in 1 \cdots N_c, j \in 1 \cdots N_r$$

Note that the desideratum of diversity is implemented as constraints to the optimization problem, and this is where role information is introduced to facilitate community detection. For each community i and role j, it is required that their inner product is less than a specified threshold value ϵ_{ij}.

We use the following setting to model the relationship between $\mathbf{C}$ and the network G. Given the community affiliation score matrix $\mathbf{C}$, we define the probability of an edge existing between v and u as a result of their affiliations with the community i:

$$P[(v, u) \in E \mid c_{vi}, c_{ui}] \equiv 1 - \exp\left(-c_{vi} \cdot c_{ui}\right) \ . \tag{3.5}$$

By treating the edge probability as independent when conditioned on each community, it is easy to show that the probability of observing the edge (v, u) with regard to the whole community assignment matrix $\mathbf{C}$ is:

$$P[(v, u) \in E \mid \mathbf{C}] = 1 - \exp\left(-\mathbf{c}_{v\bullet} \cdot \mathbf{c}_{u\bullet}\right) \tag{3.6}$$

Intuitively, the larger affiliation scores to the same community two nodes v and u have, the more likely it is to observe the edge (v, u).

This setting can also be explained by viewing the multiplicity of edge (v, u) under community i as a Poisson random variable with parameter $c_{vi} \cdot c_{ui}$. Due to the additivity of Poisson distribution, the total multiplicity of edge (v, u) in G is also a Poisson random variable with parameter $\mathbf{c}_{v\bullet} \cdot \mathbf{c}_{u\bullet}$. Therefore, higher community affiliation scores lead to higher edge multiplicity, and in terms of unweighted edge, higher possibility of observing the edge.

Given $\mathbf{C}$, the log-likelihood of the whole network is:

$$\texttt{Likelihood}(G, \mathbf{C}) = \sum_{(v,u)\in E} \log\left(1 - e^{-\mathbf{c}_{v\bullet}\cdot\mathbf{c}_{u\bullet}}\right) - \sum_{(v,u)\notin E} \mathbf{c}_{v\bullet} \cdot \mathbf{c}_{u\bullet} \tag{3.7}$$

For a specific node v, when the community affiliation scores of all other nodes $\mathbf{c}_{-v\bullet}$ are fixed, the unconstrained version of Equation 3.4 becomes convex on $\mathbf{c}_{v\bullet}$, and gradient ascent (lines 1 to 6 in Algorithm 3) can be utilized to solve it since the likelihood's gradient has a closed form:

$$\nabla c_{vi} = \sum_{u\in\Gamma_v} c_{ui} \cdot \frac{\exp\left(-\mathbf{c}_{v\bullet} \cdot \mathbf{c}_{u\bullet}\right)}{1 - \exp\left(-\mathbf{c}_{v\bullet} \cdot \mathbf{c}_{u\bullet}\right)} - \sum_{u\notin\Gamma_v} c_{ui} \tag{3.8}$$

Because the gradient ascent algorithm optimizes the community assignment for one node each time, it is difficult to directly factor in the diversity constraints in Equation 3.4, each of which is community-specific. Therefore, we purpose to relax the problem by first solving the unconstrained version as described above, and then projecting the updated community assignment to the closest possible point in the feasible region that satisfies all diversity constraints. For each community, the projection can be viewed as a quadratic programming problem with inequality constraints (lines 7 to 9 in Algorithm 3), and it can be handled by various high-level solvers.

ϵ_{ij} in the constraints are threshold parameters of the inner product between each pair of community and role vectors. Since $\epsilon_{ij} = \cos\left(\angle(\mathbf{c}_{\bullet i}, \mathbf{r}_{\bullet j})\right) \cdot ||\mathbf{c}_{\bullet i}||_2 \cdot ||\mathbf{r}_{\bullet j}||_2$, all

ϵ_{ij} parameter values can be controlled by one single parameter ϵ:

$$\epsilon_{ij} \equiv \epsilon \cdot ||\mathbf{c}_{\bullet i}||_2 \cdot ||\mathbf{r}_{\bullet j}||_2 \tag{3.9}$$

where ϵ represents the angular cosine between two vectors, and its domain is $[0, 1]$ since community and role affiliation scores are all non-negative. $\epsilon = 0$ means the community and role vectors are strictly orthogonal whereas $\epsilon = 1$ indicates no constraint. In our experiments we use $\epsilon = 0.5$ (i.e. the angle is no less than $\frac{\pi}{3}$) as the default.

Algorithm 3 outlines the two steps to update communities in each iteration.

Algorithm 3 `UpdateComm`$(G, \mathbf{C}, \mathbf{R}, N_c)$

Require: Learning rate l (fixed or learned from line search)

for all $v \in V$ **do**
  Calculate $\nabla \mathbf{c}_{v\bullet}$ according to Equation 3.8
  for all $i \in 1 \cdots N_c$ **do**
    $c_{vi} \leftarrow \max(c_{vi} + l \nabla c_{vi}, 0)$ {Gradient ascent}
  end for
end for
for all $i \in 1 \cdots N_c$ **do**
  $\mathbf{c'}_{\bullet i} \leftarrow \arg\min_{\hat{\mathbf{c}}} ||\hat{\mathbf{c}} - \mathbf{c}_{\bullet i}||_2$, s.t. $\hat{\mathbf{c}} \cdot \mathbf{r}_{\bullet j} < \epsilon_{ij}, \forall j \in 1 \cdots N_r$ and $\hat{\mathbf{c}} \geq \mathbf{0}$
  {Diversity constraints by roles}
end for

return $\mathbf{C}'$

3.2.4 Updating Role Assignment

In RC-Joint, influences of community and role assignments go both ways. In order to let up-to-date community information have impact on the role detection process, we need to incorporate it into node features. To this end, we append to $\mathbf{f}_v$, the feature vector of node v, one extra feature: the proportion of v's neighbors that have the same

dominant community label as v has.

$$\frac{|\{u \in \Gamma_v | \arg\max_{i'} (c_{ui'}) = \arg\max_{i'} (c_{vi'})\}|}{|\Gamma_v|} \tag{3.10}$$

Intuitively, a gateway node is more likely to belong to a different community than most of its neighbors, while a central node in one community will mostly connect to other core nodes in the same community.

Given updated feature values for each node, the next step is to update all N_r centroids. Features of centroids are recalculated as the sum of feature values from all nodes, weighted by their role affiliation scores. The step of adjusting role affiliation scores for nodes has the same form as Equation 3.3, except that the underlying feature vector is slightly different since the feature from Equation 3.10 was not used during role initialization. Algorithm 4 lists the steps to update roles.

Algorithm 4 `UpdateRole`$(G, \mathbf{C}, \mathbf{R}, N_r)$

for all $v \in V$ **do**
  $\mathbf{f}_v$[intra-community neighbor ratio] $\leftarrow$ Equation 3.10 {Update node features}
end for
for all $j \in 1 \cdots N_r$ **do**
  $\mathbf{f}_{s_j} = \frac{\sum_{v \in V} r_{vj} \mathbf{f}_v}{\sum_{v \in V} r_{vj}}$ {Update centroids}
end for
for all $v \in V$ **do**
  for all $j \in 1 \cdots N_r$ **do**
    $r'_{vj} \leftarrow$ Equation 3.3 {Update role assignment}
  end for
end for

return $\mathbf{R}'$

3.3 Design Choices and Techniques for Speedup

We dedicate this section to how RC-Joint can be implemented efficiently and the selection of parameters. First we show how results of RC-Joint on a sparse network can be used to initialize the algorithm on the original network. Then we discuss leveraging the inherent parallelism in RC-Joint via parallel computing paradigms. Reusing computed results and reducing subroutine's problem size also help decrease the computation cost. Finally we shed light on the process of selecting N_c and N_r values.

3.3.1 Initialization with Results from Sparse Networks

In Sections 3.2.1 and 3.2.2 we present our default methods of initializing communities and roles. Here we present a refinement that is analogous to the use of sampling in initializing various clustering algorithms such as K-means, Expectation-Maximization and even Graph Clustering [122]. Specifically, here we first sample (sparsify) the edges of the original graph. Next we run RC-Joint on the sampled (sparse) graph and obtain the community membership and role associations. We refer to this as the *first run*. We use the results of the *first run* to initialize a second run of RC-Joint on the full network. We refer to the latter as the *second run*.

Given the network $G = (V, E)$, the sampled or sparse version of it is denoted $G_{sparse} = (V, E_{sparse})$ has the same set of nodes but a smaller set of edges ($E_{sparse} \subset E$). The process of deciding which edges to keep in E_{sparse} can be viewed as a sparsification exercise. We examine two strategies described below:

- **Random Sparsification:** Sample edges uniformly at random. Retain sampled edges in E_{sparse}.

- **Local Rank Sparsification:** Rank all edges according to an edge similarity metric (e.g. estimate of the Jaccard similarity in Equation 3.3). Edges that have a higher triangle density (participate in a greater number of triangles within the network) will be ranked higher. For each node, rank its incident edges according to the above metric, and retain a number of top-ranked edges. This approach has been shown to preserve salient community structure especially in graphs with communities of varying densities, and to deliver high-quality results at a fraction of the cost [122]. Our hope is this strategy can also help in our context.

To reiterate, given a sparse network G_{sparse}, we first supply it to RC-Joint and obtain community and role score matrices $\mathbf{C}_{sparse}$ and $\mathbf{R}_{sparse}$. Then RC-Joint is run on the original network G, using $\mathbf{C}^0 = \mathbf{C}_{sparse}$ and $\mathbf{R}^0 = \mathbf{R}_{sparse}$. The **key intuition** here is that using those initial values will allow the second run to finish much faster than using the default because (1) $\mathbf{C}_{sparse}$ and $\mathbf{R}_{sparse}$ yield better objective function values, so that fewer iterations are needed to converge, and (2) $\mathbf{C}_{sparse}$ and $\mathbf{R}_{sparse}$ are more sparse (i.e. more zeros in affiliation scores), thus fewer operations are performed when updating communities and roles iteratively.

In Section 3.4.3, we will report results from this sparse graph initialization approach. The default strategy we adopt is local rank sparsification, and for a node of degree d, $\left\lceil \sqrt{d} \right\rceil$ incident edges of the highest Jaccard similarity are preserved. As expected, using $\mathbf{C}_{sparse}$ and $\mathbf{R}_{sparse}$ indeed reduces the total running time of RC-Joint (two runs combined), and on several datasets it also improves the quality of detected communities and roles.

3.3.2 Parallelizing RC-Joint

Main stages of `UpdateComm` (Algorithm 3) and `UpdateRole` (Algorithm 4) are inherently parallelizable. When computing the community assignment, gradient calculation can be performed on each node independently. Quadratic programming with diversity constraints can also be done on each community separately. During the process of updating role affiliation scores, each node can be updated individually. Lastly, updating centroids in role detection are parallelizable as well, although in practice the improvement may not be as significant since N_r is usually quite small.

In our implementation, we use OpenMP to exploit such parallelism, and the speedup is significant. Distributed computing architecture such as MPI can also be used, and we leave this as one direction of future work.

3.3.3 Reusing Computed Results

We have already mentioned one instance of result reusing, where min-wise hash signatures are used for both community initialization and role feature calculation. Another case is introduced in [148], where the authors point out that when calculating the gradient of a node's community affiliation scores (Equation 3.8), the last item can be rewritten as

$$\sum_{u \notin \Gamma_v} c_{ui} = \sum_{v \in V} c_{vi} - \sum_{u \in \Gamma_v} c_{ui} \tag{3.11}$$

and that $\sum_{v \in V} c_{vi}$ remains the same in each iteration. This reduces the complexity of gradient calculation from $O(|V|^2)$ to $O(|E|)$.

3.3.4 Reducing Quadratic Programming Problem Size

In the second part of Algorithm 3, community affiliation scores for each community are adjusted by being projected to the closest point in the feasible region that satisfies all N_r diversity constraints (one for each role). In its original form, each quadratic programming problem need to solve for $|V|$ variables, and this becomes a performance bottleneck when the network is large. However, the following lemma shows that the problem size can be reduced to the number of non-zeros in each community.

Lemma 2. *For a community i, let*

$$\mathbf{c'}_{\bullet i} = \arg\min_{\hat{\mathbf{c}}} ||\hat{\mathbf{c}} - \mathbf{c}_{\bullet i}||_2$$

such that $\hat{\mathbf{c}} \cdot \mathbf{r}_{\bullet j} < \epsilon_{ij}, \forall j \in 1 \cdots N_r$ and $\hat{\mathbf{c}} \geq \mathbf{0}$. For any $v \in V$, if $c_{vi} = 0$, then $c'_{vi} = 0$.

Proof. Assume there exists a node $v \in V$ such that $c_{vi} = 0$ and $c'_{vi} > 0$. Let another assignment vector $\mathbf{c''}_{\bullet i}$ be that $\mathbf{c''}_{-vi} = \mathbf{c'}_{-vi}$ and $c''_{vi} = 0$. Apparently $\mathbf{c''}_{\bullet i}$ satisfies the non-negativity constraint.

For any role $j \in 1 \cdots N_r$, $\mathbf{c''}_{\bullet i} \cdot \mathbf{r}_{\bullet j} = \mathbf{c'}_{\bullet i} \cdot \mathbf{r}_{\bullet j} - c'_{vi} r_{vj} \leq \mathbf{c'}_{\bullet i} \cdot \mathbf{r}_{\bullet j} < \epsilon_{ij}$. Therefore $\mathbf{c''}_{\bullet i}$ also satisfies all diversity constraints.

Finally,

$$\begin{aligned}
&||\mathbf{c''}_{\bullet i} - \mathbf{c}_{\bullet i}||_2 && (3.12)\\
=&\sqrt{\sum_{v' \neq v} (c''_{v'i} - c_{v'i})^2 + (c''_{vi} - c_{vi})^2} && (3.13)\\
=&\sqrt{\sum_{v' \neq v} (c'_{v'i} - c_{v'i})^2 + (c''_{vi} - c_{vi})^2} && (3.14)\\
<&\sqrt{\sum_{v' \neq v} (c'_{v'i} - c_{v'i})^2 + (c'_{vi} - c_{vi})^2} && (3.15)\\
=&||\mathbf{c'}_{\bullet i} - \mathbf{c}_{\bullet i}||_2 && (3.16)
\end{aligned}$$

which contradicts with the claim that $\mathbf{c'}_{\bullet i}$ is closest to $\mathbf{c}_{\bullet i}$.

Therefore, if $c_{vi} = 0$, c'_{vi} must be 0, too. □ □

From Lemma 2, it is easy to see that one can obtain $\mathbf{c}'_{\bullet i}$ by:

1. Creating a compact vector $\tilde{\mathbf{c}}_i$ from $\mathbf{c}_{\bullet i}$ by keeping only all non-zero elements.

2. Finding $\tilde{\mathbf{c}}'_i$, the closest projection of $\tilde{\mathbf{c}}_i$ in the feasible region.

3. Expanding $\tilde{\mathbf{c}}'_i$ back to length $|V|$ by filling corresponding elements with 0.

Here, the number of variables in the optimization problem is only the number of non-zeros in $\mathbf{c}_{\bullet i}$, which is much smaller than $|V|$.

3.3.5 Choosing N_c and N_r

The number of communities and roles to find are two parameters provided by end users to RC-Joint, and there are several strategies to select them. One can perform grid search of N_c and N_r on a held-out development set, and choose values that result in the highest likelihood. Alternatively, measures like Bayesian Information Criterion (BIC) or Minimum Description Length (MDL) can be calculated, and N_c, N_r that minimize the combination of modeling and error costs can be selected.

For our network dataset, we compare total numbers of bits under different N_r values as in RolX [64], and find that $N_r = 4$ often yields the minimum description length. Therefore we use this value for all networks in experiments. For networks without ground truth of communities, we pick N_c by following the empirical evidence that community structure is most pronounced when the community size is approximately 100 [82].

3.4 Experiments and Evaluation

In this section, we apply RC-Joint to both real and synthetic networks, aiming to understand its performance on both community detection and role detection under various scenarios. We first evaluate RC-Joint and state-of-the-art algorithms on the community detection task (Section 3.4.1), then compare it with existent role detection algorithms (Section 3.4.2). We also investigate the effects of different initialization schemes on the algorithm's execution and performance (Section 3.4.3).

3.4.1 Performance on Community Detection

Networks for Community Detection

We download a collection of real-world networks that have ground truth on the community membership[16], and discard edge directions if the original network is directed. The type of networks varies from social network to product network, and they have different levels of density as well as community size. Table 3.2 summarizes the basic information of those networks. All networks considered have ground truth on overlapping communities.

Evaluation Metric and Comparisons

Because ground truth information is available, we can gauge the performance of each community that an algorithm has discovered and whether a ground truth community has been successfully identified.

[16]They are all available from the SNAP network repository.

Network	$\|V\|$	$\|E\|$	# Comm.	Avg. Comm. Size	Ground Truth
Facebook	4039	88234	193	23	Facebook friend list
Twitter	81306	1342303	4065	14	Twitter list
Google Plus	107614	12238285	468	136	Google Plus list
Amazon	334863	925872	120999	20	Product category
YouTube	1134890	2987624	14870	8	User group
LiveJournal	3997962	34681189	576120	12	User-defined group

Table 3.2: Information of networks for community detection. Communities may be overlapping.

Since affiliation scores are real values instead of binary, we filter off nodes with low affiliation scores from each community to get a compact representation of communities. The filtering threshold can be set to $\sqrt{\frac{2|E|}{|V|^2}}$, square root of the empirical edge probability [148].

For each ground truth community $c_{\tilde{i}}$, we create a length-$|V|$ vector $\tilde{\mathbf{c}}_{\bullet\tilde{i}}$ where $\tilde{c}_{v\tilde{i}} = 1$ if v belongs to $c_{\tilde{i}}$, or 0 otherwise. The standard F-score formula is then extended to handle affiliation scores (assuming $\mathbf{C}$ and $\tilde{\mathbf{C}}$ have been L1-normalized over nodes):

$$\text{precision}(i,\tilde{i}) = \frac{\mathbf{c}_{\bullet i}\cdot\tilde{\mathbf{c}}_{\bullet\tilde{i}}}{||\mathbf{c}_{\bullet i}||_1}, \text{recall}(i,\tilde{i}) = \frac{\mathbf{c}_{\bullet i}\cdot\tilde{\mathbf{c}}_{\bullet\tilde{i}}}{||\tilde{\mathbf{c}}_{\bullet\tilde{i}}||_1}, \tag{3.17}$$

$$\text{f-score}(i,\tilde{i}) = \frac{2\cdot\text{precision}(i,\tilde{i})\cdot\text{recall}(i,\tilde{i})}{\text{precision}(i,\tilde{i})+\text{recall}(i,\tilde{i})} \tag{3.18}$$

Let $\tilde{N}_c$ be the total number of ground truth communities, we then calculate the overall F-score using the following formula:

$$\begin{aligned} F(\mathbf{C},\tilde{\mathbf{C}}) \quad &= \frac{1}{2}\Big(\frac{\sum_{i=1}^{N_c}\max_{\tilde{i}=1}^{\tilde{N}_c}(\text{f-score}(i,\tilde{i}))}{N_c} \\ &\quad + \frac{\sum_{\tilde{i}=1}^{\tilde{N}_c}\max_{i=1}^{N_c}(\text{f-score}(i,\tilde{i}))}{\tilde{N}_c}\Big) \end{aligned} \tag{3.19}$$

We compare RC-Joint with three representative community detection algorithms, BigClam [148], MLR-MCL [120], and Graclus [37]. BigClam employs the same setting of community affiliation scores in Section 3.2.3 to discover overlapping communities. It has been shown that BigClam outperforms many existing overlapping community detection algorithms, including line graph clustering [3], clique percolation model [102], and mixed membership stochastic block model [4]. However, it does not detect roles, nor does it exploit the influence of roles on communities. MLR-MCL takes a multi-level approach and identifies communities by propagating stochastic flows over a network and identifying each flow attractor as well as its contributors as one cluster. Similarly, Graclus performs multi-level clustering where at each level kernel k-means is run to optimize a partitioning's normalized cut. MLR-MCL and Graclus do not have the ability to detect overlapping communities.

Results

Table 3.3 summarizes the evaluation results, with F-scores of all algorithms on each network. We provide the actual number of communities in each network as the input parameter to each algorithm.

The largest network, LiveJournal, only successfully finishes on RC-Joint with local rank sparsification, and MLR-MCL. This demonstrates the benefits of using proper initialization, which will be further discussed in Section 3.4.3. Moreover, Graclus crashes when running on Amazon and YouTube, too. Comparing with BigClam, we find that RC-Joint has better performance on most networks. This demonstrates the efficacy of RC-Joint's inherent design to provide auxiliary information via the role assignment, in order to facilitate the process of community detection. When initializing RC-Joint with communities and roles identified from a sparse network, the results

	Facebook	Twitter	Google+	Amazon	YouTube	LJ
RC-Joint	**0.3928** (7%)	0.2431 (2%)	0.2160 (-11%)	0.4765 (2%)	**0.0503** (2%)	N/A
RC-Joint w/ sparse init.	0.3843 (5%)	**0.2506** (5%)	**0.2499** (3%)	0.4688 (1%)	0.0491 (0%)	0.1632
BigClam	0.3660	0.2381	0.2416	0.4664	0.0491	N/A
MLR-MCL	0.2701 (-26%)	0.1146 (-52%)	0.0100 (-96%)	**0.5001** (7%)	0.0068 (-86%)	0.1497
Graclus	0.3026 (-17%)	0.2147 (-10%)	0.1789 (-26%)	N/A	N/A	N/A

Table 3.3: F-scores on community detection, and the value in brackets is the percentage of improvement from BigClam. LiveJournal ("LJ") is only finished on RC-Joint with sparse network initialization and MLR-MCL. Graclus also crashes on Amazon and YouTube.

are still highly competitive, and for Google Plus the performance is significantly improved. On the other hand, non-overlapping community detection methods do not fare well in general, except for MLR-MCL on Amazon.

The advantage of RC-Joint is also reflected in the log likelihood of the network edge set (Equation 3.7), as we find that RC-Joint achieves better log likelihood values than BigClam on all networks except Google Plus (Table 3.4). This shows the same trend as in Table 3.3.

	RC-Joint	RC-Joint w/ sparse init.	BigClam
Facebook	-171085	**-167758**	-182284
Twitter	**-3305980**	-3341592	-3381248
Google+	-57249698	**-49624553**	-52169083
Amazon	-5452800	**-5434790**	-5476358
YouTube	-19101405	**-18713629**	-19138838

Table 3.4: Log likelihood of the network, given the extracted community assignment values. The closer the log likelihood value is to 0, the higher the quality.

3.4.2 Performance on Role Detection

In this section, we investigate the performance of RC-Joint on its second task: role detection.

Networks for Role Detection

Real-world networks: One challenge that the role detection literature has been facing is the availability of ground truth on roles for real-world networks, and most work [64, 116] has to use some relevant tasks to indirectly measure the quality and meaningfulness of roles extracted. To alleviate the problem, we propose to use a

node's behavior in diffusing and blocking information flows as the surrogate of its role.

Specifically, we calculate two sets of measures for each node and use them to define ground truth on roles. The first set is influence and passivity values of each node, as described in [114], where nodes (i.e. users) of information networks start and/or selectively relay cascades (e.g. URLs, photos, memes). The influence of a user is based on how many users it mobilizes and how difficult to mobilize those users are. The passivity of a user, on the other hand, is determined by how unlikely it is for him to forward information and how influential his friends are. For a network, we rank influence and passivity values over all users and divide both into two bins, respectively. Bin combinations (four types) are then considered to be the ground truth label for the network's role assignment. The second set of measures is influence and blockade, as defined in [27]. Influence is defined as the proportion of re-shares a user receives among all information he has shared. Blockade is calculated as the ratio of the number of cascades a user does not re-share to the total number of cascades he has received. Similarly, influence and blockade values are binned to create role labels. Both sets of measures attempt to capture the duality of propagating and impeding information flows, though the former set is updated iteratively until convergence and the latter is not.

We use two information networks for our experiments: Digg [79] and Flickr [20]. The Digg network has 19609 nodes and 161650 edges, where all votes on a particular story is viewed as a cascade. The Flickr network has 33887 nodes and 2441316 edges, where all favorites of a particular photo is considered to be a cascade.

Synthetic networks: Apart from information networks, we also create a collection of synthetic networks where role assignments are known in advance. We consider four different nodal types here:

- Member of a 10-clique. We create five such cliques, corresponding to 50 nodes in total.
- Member of a 5-clique. We create ten such cliques, corresponding to 50 nodes in total.
- Bridge of degree 2. We create 25 of them.
- Star of degree 10. We create 25 of them.

Bridges and stars are randomly connected to cliques, in order to make the whole network connected. The last step is to add noise edges between any pair of nodes with a fixed probability ρ. The value of ρ is ranged to generate networks with varying difficulty. Each node type described above is treated as one role, and this forms the ground truth for all synthetic networks.

Evaluation Metric and Comparisons

We use the same formula (Equation 3.19) to calculate the F-score on role detection. Apart from RC-Joint, we also compare with RolX and three variants of GLRD (sparsity, diversity[17], alternative role discovery constraints on role vectors). Because details on the selection of constraint thresholds in GLRD are not specified, we choose them in the following manner. For the sparsity constraint (on the target role vector's L1-norm), we let the threshold be $\frac{|V|}{N_r}$. For the diversity constraint (on the inner

[17]This is different from the *diversity* constraints in RC-Joint (Equation 3.4).

product of the target role vector and every other role vector) and alternative role discovery (on the inner product of the target role vector and any externally-specified vector), we set the threshold of angular cosine (similar to ϵ in Equation 3.9) to 0.5. We use communities identified by BigClam as the guideline for GLRD's alternative role discovery.

Results

F-scores of various algorithms on Digg and Flickr are reported in Table 3.5. Results for synthetic networks are listed in Table 3.6. We separate the results on real networks and synthetic networks because the sources of ground truth are different.

	Influence/Passivity [114]		Influence/Blockade [27]	
	Digg	Flickr	Digg	Flickr
RC-Joint	0.2032 (8%)	**0.1372** (5%)	**0.1407** (15%)	**0.0565** (14%)
RC-Joint w/ sparse init.	**0.2033** (8%)	0.1371 (5%)	0.1406 (15%)	0.0563 (14%)
RolX	0.1886	0.1301	0.1225	0.0496
GLRD Alternative	0.1885 (0%)	0.1291 (-1%)	0.1228 (2%)	0.0536 (8%)
GLRD Sparsity	0.1792 (-5%)	0.1295 (0%)	0.1217 (-1%)	0.0509 (3%)
GLRD Diversity	0.1866 (-1%)	0.1304 (0%)	0.1231 (0%)	0.0522 (5%)

Table 3.5: F-scores on role detection on real-world networks with two sets of influence-induced ground truth labels, and the value in brackets is the percentage of improvement from RolX.

	$\rho = 0.01$	$\rho = 0.05$	$\rho = 0.10$
RC-Joint	0.7189 (35%)	**0.5531** (75%)	**0.3735** (34%)
RC-Joint w/ sparse init.	**0.7275** (37%)	0.5132 (62%)	0.3689 (33%)
RolX	0.5314	0.3168	0.2782
GLRD Alternative	0.4877 (-8%)	0.3182 (0%)	0.2822 (1%)
GLRD Sparsity	0.5044 (-5%)	0.3186 (1%)	0.2808 (1%)
GLRD Diversity	0.5061 (-5%)	0.3270 (3%)	0.2787 (0%)

Table 3.6: F-scores on role detection on synthetic networks with different amount of noise edges, and the value in brackets is the percentage of improvement from RolX.

It can be seen that RC-Joint obtains results of higher quality than both RolX and GLRD, uniformly. Initialization using sparse network also performs well. F-scores of GLRD fall between those of RC-Joint and RolX, demonstrating the power of providing community information to guide role detection, and the downside of treating community information as static input.

3.4.3 Effects of Initializing with Sparse Networks

Local Ranking

Previously in Section 3.3.1, we discuss the possibility of seeding RC-Joint with results from a preliminary run on a sparse version of the network. Moreover, we have already seen the quality improvement this technique can provide in Sections 3.4.1 and 3.4.2. Those sparse networks are produced by local rank sparsification, where each node of degree d keeps $\left\lceil \sqrt{d} \right\rceil$ incident edges with the highest Jaccard similarity of adjacency lists.

In this section, we report the impact on RC-Joint's time consumption by this technique. Figure 3.1 shows the amount of time it takes to run RC-Joint (with and without sparse network initialization) as well as BigClam. Implementations of both RC-Joint and BigClam are in C/C++, using OpenMP with 8 threads. Experiments are run on a desktop with an Intel i7 quad-core processor and 16GB of RAM.

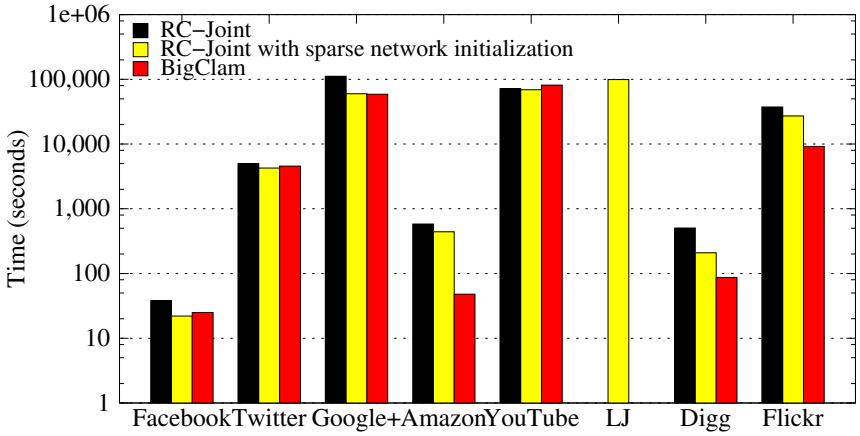


Figure 3.1: Comparison of time consumption (OpenMP with 8 threads). For RC-Joint with sparse network initialization, the running time include both runs.

As the plot suggests, using initialization from results of the sparse network always leads to lower total running time (both runs combined), as anticipated in Section 3.3.1. In the cases of Facebook, Twitter, Google Plus and YouTube, it is also faster than BigClam. This could be because proper initialization lets RC-Joint start at a state closer to convergence.

It is worth pointing out again that using sparse network initialization enables us to operate on even larger networks when RC-Joint itself or other methods becomes too slow. For example, experiments on the LiveJournal network do not finish in two

days with either RC-Joint or BigClam. However, by first processing on the sparse version of it and then initializing another run with those results, RC-Joint manages to finish the computation in 25 hours.

Benefits of Sparsification

One may ask if the benefits of edge sparsification to RC-Joint can be precisely quantified with respect to efficiency and quality. To evaluate, we consider Twitter, Google Plus, two networks in our study. Similar results are observed for other networks in our study. Edge retention probability values are set up so that both strategies retain roughly the same number of edges in each network. Table 3.7 summarizes the amount of time each edge sparsification strategy takes for two runs, as well as the quality of results. F-score of the first run is from the results of RC-Joint on the sparse network itself, and F-score of the second run is from the results of RC-Joint on the original network. We also calculate the percentage of time saved and F-score increased compared with the default RC-Joint, and report those values in corresponding brackets.

Not all edge sparsification strategies are equal in terms of efficiency and quality. Edge ranking leveraging similarity information and local sparsification is more efficient than random sparsification, and the results have higher F-scores. Intuitively, local ranking is effective in capturing the skeleton of the network and enables faster convergence. In terms of quality of communities and roles, results from the local edge ranking sparsification procedure is also significantly better than random sparsification.

We note that numbers of RC-Joint iterations in the second run of local ranking for Twitter and Google Plus are 55 and 66, respectively (not shown in the table).

In contrast, RC-Joint with default initialization takes 64 on Twitter and 100 on Google Plus. The reduction in number of iterations is consistent with the speedup in running time. Therefore, the first run on the sparse network helps to find a better initialization, decrease the number of iterations required, and therefore reduce the total running time.

3.5 Discussion

Across the board, RC-Joint achieves higher quality than baseline methods which identify only communities or roles. Existing single-tasked community (role) detection algorithms suffer from not exploiting the latest knowledge on roles (communities), accounting for lower performance. For all experiments, we have reported absolute F-score values as well as relative improvements over baseline methods. We note that, in general, the problem we tackle is quite challenging (the absolute F-score values are not very high, also observed in other contemporary studies [148]). This reflects the inherent difficulty of community and role detection as well as room for future improvement.

Different initialization schemes also impact the efficiency and performance of RC-Joint, and we investigate the potential of edge sparsification techniques in the context of creating good seeds of communities and roles. We find that edge sparsification based on structural similarity is more effective than selecting edges by random, and local edge sparsification yields the most speedup and performance gain.

The RC-Joint approach we describe offers a marked departure from most existing algorithms. In terms of community discovery, BigClam [148] is somewhat related in that the relationship between community affiliation scores and the edge set has

a similar formulation. However, BigClam only optimizes likelihood of the network without any constraint, and RC-Joint differs from it by being able to adjust the community assignment to accommodate the latest role assignment after each iteration. This difference we believe accounts for RC-Joint's qualitative improvements over BigClam. With respect to role discovery, RC-Joint also bears important difference from existing NMF-based role detection algorithms, such as RolX [64] and GLRD [50], as it uses soft k-means to identify roles, and it considers guidance from the community structure. The guidance is non-parametric and does not require extrinsic input from the domain. Essentially, in RC-Joint, roles are treated as the external knowledge to guide community detection, and such external knowledge is dynamically updated after each iteration.

3.6 Conclusion

We propose RC-Joint, a principled algorithm to mine communities and structural roles from networks simultaneously. RC-Joint operates on the observation that community and role assignments are complement to each other, and utilizing information from one component can benefit the discovery process of another. During each iteration, RC-Joint updates communities and roles alternately by improving the network likelihood and soft k-means objective function, respectively. The end result is an algorithm that is capable of identifying overlapping community and role assignments simultaneously. Empirical evaluations of RC-Joint and other state-of-the-art single-tasked mining algorithms on real-world as well as synthetic networks show that RC-Joint indeed produces communities and roles that have higher quality with regard to the gold standard. Furthermore, we find that algorithm speedup as well as quality

improvement can be achieved by running RC-Joint on a sparse version of the network and using its results to initialize another run on the original network.

In order to extend RC-Joint, there are multiple fronts to explore. It is promising to investigate networks from a multilevel perspective, i.e. a community of nodes being viewed as a super node, whose role and community membership in the resultant collapsed network is in turn highly intriguing. The multilevel approach also allows one to efficiently identify communities and (potentially) roles in a crude network, and to then project them back to the original network, as successfully exploited previously [71, 37, 120]. It will also be beneficial to extend RC-Joint to directed, weighted and/or signed networks, because some real-world networks have those properties. A third direction is to explore other community-induced node features to be used in updating role affiliation scores. Finally, implementations of RC-Joint using other more sophisticated parallel computing paradigms need to be investigated to realize even more speedup.

	Local					Random				
	First run		Second run			First run		Second run		
	Time	F-score	Time	F-score	Total time	Time	F-score	Time	F-score	Total time
Twitter	90	0.2172	4175	0.2506 (5%)	4265 (13%)	1268	0.1746	8612	0.2390 (0%)	9880 (-116%)
Google Plus	1042	0.1938	58844	0.2499 (3%)	59886 (46%)	8858	0.1311	89069	0.2360 (-2%)	97927 (12%)

Table 3.7: Running time (in seconds) and F-score of two runs of RC-Joint. The first run is on the sparse network, and the second run is on the original network using results from the first run. Improvement of running time and F-score over RC-Joint with no sparse network initialization are included in brackets.

Chapter 4: Predicting User Engagement with Structural, Content, Profile, and Behavioral Features

In this chapter, we discuss the problem of predicting the engagement of OSN users in event-oriented discussion, where user engagement is defined as a user's writing or sharing messages about the specific topic related to the event. The accurate prediction of user engagement enables one to reason the event's future development, and it has a multitude of potential applications. A first example is movie studio's strategy making on spreading the message of a movie's release in social media. If they can identify prominent factors affecting user engagement, those factors can be emphasized accordingly to maximize the word-of-mouth effect. In another use case, during an event of crisis, emergency teams are looking forward to help the victims. User engagement analysis could help us understand how effectively the community surrounding this event can grow to reach potential donors and people in need of resources (food, water, first aids etc.), also what are the best possible ways to communicate between these resource providers and people in need for resources.

The study of user engagement is by no means simple, as there is a three-dimensional dynamic at play: social network structure, user-generated content, participant behavior. Historical information of user engagement is also part of the knowledge one could exploit. Given a discussion topic on social media, what motivates a user to

engage in the discussion? Can we predict *whether* and *when* the engagement will happen, and, if so, *how strong* the engagement will be? Here, a topic is formalized as a real-world event, discussions are thus surrounding this event, and all participants compose a community (which will be formally defined as an *event-oriented community* in section 4.1.2). For example, during Japan Earthquake 2011, an event of natural disaster, people tweeting about "Japan Earthquake" would be considered to be part of the Japan Earthquake topic discussion community. The task of finding factors which drive user to engage in topic discussion, therefore, can also be considered as identifying factors that influence user to join the corresponding community.

We use Twitter as a social information source and manage to build a prediction model for user engagement in topic discussion about events. Compared with previous work which resort to small subsets of features, and isolated study of factors with different characteristics, we investigate a range of features in four categories (content, author, network, and past activity) to study the factors responsible for user engagement on social media.

4.1 Problem Statement

4.1.1 Terminology Definition

As described in introduction, user engagement in a topic discussion can be understood in terms of user participation in community formed around topic of discussion. We define some terminologies used in the context here and then give our problem statement:

- **Event-Oriented Community:** We define an event-oriented community as an implicit group of social network users who have joined discussion on topic

about an event, or more precisely who have posted messages about the topic. In different online social networks, posts may appear in different forms (e.g. *status, share* and *comment* for Facebook or *tweet* and *retweet* for Twitter). A social network user is considered to become engaged in the topic discussion and hence, a member of the event-oriented community if he writes or forwards the event-related post. For instance, all the twitterers who are posting about Emmy Awards and thus joining topic of discussion during Aug 10 to Sept 20, 2010 are regarded as members of Emmy Awards community.

- **Slice and Snapshot:** A slice refers to the collection of event-related messages (tweets) or in other words, messages relevant to topic of discussion, posted during a fixed-length period of time (e.g. 24 hours). A snapshot refers to state of the network at a certain point of time at which user profile and connection information are stored. In the current context, we take the snapshots for the network of users who are members of the community formed around topic of discussion.

  Depending on the inherent characteristics of event, we set two different slice lengths (one day and eight hours, respectively) in order to capture the dynamics of community more promptly, since some event-oriented topics of discussion draw a quick attention of users, which in turn engages huge number of users. More detailed discussion is available in section 4.2.2.

- **Temporal Weight of Information:** While the total size of community surrounding topic of discussion keeps increasing as it evolves, the *freshness* of it

should also be taken into account when we study users' behavior. Most online social networks' layout designs show the latest information first, and users have to scroll down to see earlier news feed. Therefore, it is natural to assume that later the information is generated, the higher possibility it get consumed [145] and the higher weight it carries on influencing user decision about engaging in topic discussion.

To leverage this observation, we set a hard margin of 3 slice units and only consider information tweets within this time window as we believe they are most likely to be viewed. Users who wrote or shared event-related messages during this period are called *active users*, and we would like to focus on how they joined the event-oriented communities, forming the *active community*, and how their followers (i.e. audience) will react in accordance. For each active user and the content he/she generated, a temporal weight is leveraged based on the time that has elapsed since its creation.

Figure 4.1 illustrates the notions of slice, snapshot and active community, to provide the readers a clearer conception.

4.1.2 Problem Definition

Using the terminologies introduced so far, the problem of finding factors impacting user engagement can be defined as user engagement prediction problem for joining a topic of discussion:

Definition 1. User Engagement Prediction Problem*: Given 1) an event-oriented community* $\boldsymbol{C}$ *formed around a topic of discussion; 2) a Twitter user* $\boldsymbol{U} \notin$

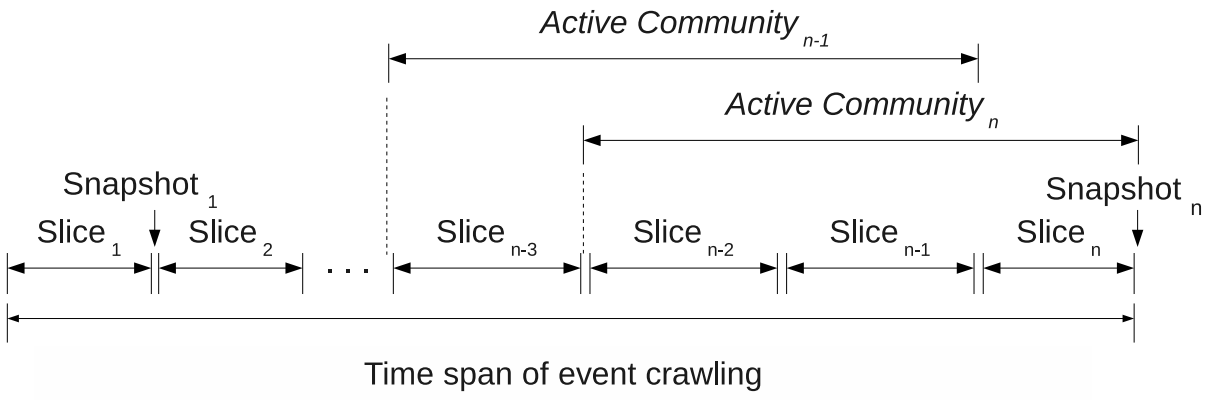


Figure 4.1: Illustration of slice, snapshot and active community

***C**, predict whether **U** will be engaged in **C** (by composing a new tweet or retweeting an existing tweet which contains keywords or hashtags related to **C**'s underlying event) in a future slice. If so, **U** is said to be a positive record. Otherwise, it is a negative record.*

Although there has been literature studying user engagements in OSNs [128, 108, 76, 26], engagement is often defined as the sharing of existing information, excluding the spontaneous creation of relevant information. Among the few that tackle a problem that is more similar to ours [147, 90], the focus is often on the aggregated level, instead of at individual users. Our work, therefore, presents novel contribution to a more comprehensive understanding of how OSN users engage in online discussions.

4.2 Methods

This section introduces methodologies involved in building the prediction model. Particularly, a detailed discussion about the groups of features used to build the model

in order to solve the user engagement prediction problem, as defined in section 4.1.2, is provided.

4.2.1 Twitter as a Data Source

Launched in 2006, Twitter has been well-known both as a micro-blogging service provider and a social network platform. A message posted by the user is called a *tweet*, which typically contains plain texts, *hashtags* (e.g. #nsn3d, #MusicMonday) that indicate explicit topic categorization, and hyperlinks to other multi-media content that promote the spread of information from all over the Web. The length of each tweet is limited to 140 characters.

Users of Twitter have directed *follower* connections with other users of the site that allows them to keep track or *follow* those other users. Members can post tweets, respond to a tweet which is called *reply*, or forward a tweet to all followers which is called *retweet*. Replies to any tweet are directed to a user (not the conversation thread) by placing a *@user* reference in the tweet while retweets are means of participating in a diffuse conversation. The *@user* reference can also be used to refer to a particular user in the tweet content, which is called a *mention* of that user. Tweets are generally available as feeds from follower networks and also via a searchable interface. Apart from the 140-character text itself, time-stamp and location information are all publicly retrievable unless privacy control is turned on by the user. We store most of this data to construct features.

To investigate the users' behavior after perceiving activities of the community, it is infeasible to randomly pick users from millions of Twitter accounts since it is unclear if they are aware of the event at all. Instead, all active users and the users who follow

at least one active user at that snapshot are considered. There are two indications here. Firstly, most users are guaranteed to have access to the event information from the topic related tweets posted or retweeted by their online friends. Therefore, sophisticated social network features can be used to analyze information propagation in networks. Secondly, a user may be inactive for several snapshots before joining the community, resulting in many *negative* records and one *positive* record. The collection of all records and the edges joining them forms an *active network*.

4.2.2 Community Categorization by Event Characteristics

Popular events on social networks belong to diverse domains and differ in characteristics and behavior. Some events like FIFA World Cup drive attention of global populace, while Health Care debate events are of national interest, and few other events similar to Iowa State Fair are attractive to a relatively small region. Another categorization is based on the event occurrence and duration. FIFA World Cup event is scheduled long time in advance while events such as Earthquake in Haiti has sudden occurrence. Apparently the characteristics of an event-oriented community largely depends on the triggering event, and it is intriguing to explore the relation of user participation behavior with communities' nature. We expect a variety of community gathering around events and one characteristic from each of the following categorizations is assigned to each event-oriented community (see section 4.3.1 for details):

- ***Global vs. Local:*** Depending upon the interest level, an event can be *global* (such as Emmy Awards) or *local* (such as Iowa State Fair). *Local* communities can further be distinguished by national interest (for example, fans of NFL

championship in US) and regional interest (For example, Ohio State Fair in Ohio), though it is not explicitly specified in the present work.

- ***Compact vs. Loose:*** Events may interest varying audiences within which the level of existing interaction among users changes significantly. For example, two fans mentioning the release of a new movie may not have talked to each other previously at all, thus the community formed around this topic is very *loose*. Meanwhile, interested authors for a technical conference topic like LinuxCon are highly likely to have interacted with each other before, and therefore belong to a relatively *compact* community.

- ***Deterministic vs. Unexpected:*** A few events are known to us beforehand while others have sudden occurrence. Therefore, the corresponding communities are *deterministic* and *unexpected*, respectively.

- ***Transient vs. Lasting:*** Different events create different level of buzz in the community and so the community might be either *transient* or *lasting*. As an example of *transient* community, there was hardly anyone talking about the hostage incident in Discovery Channel Building in Seattle three days after it since the crisis was resolved within hours. Meanwhile, discussion of the movie Avatar lasted for months. For the *transient* communities, a unit length of eight hours for time slice is used to capture fast-changing trends better. For the *lasting* communities, we use one day as the unit length for time slice.

4.2.3 Feature Categorization

Previous works have employed a wide range of features, which generally fall into three categories: community, author and content. Those works, however, seldom incorporated multiple groups of knowledge into a single model. We organize these features in one integrated framework, and investigate which ones contribute most to the predictability. Apart from that, we also introduce novel features into the system, which is another contribution. The three groups of features for each $\boldsymbol{U}$ are described as follows.

Community Features

Community features involve several measurements of the event-oriented community including the size of the active community, the total number of active users that $\boldsymbol{U}$ is following, the size of the weakly connected component (WCC) in the active network that $\boldsymbol{U}$ belongs to, and the ratio of this WCC's size to the active network's size.

Author Features

Author features involve statistics about the active users that $\boldsymbol{U}$ is following, as they are the main source of $\boldsymbol{U}$'s awareness and knowledge of the topic. We would like to discover if those users' social network states have any influence on $\boldsymbol{U}$'s participation behavior. We consider the counts of followers, followees as the features since they implicitly reflect authors' influences.

The influence and passivity scores proposed by Romero et al. [114] can also be meaningful author features. However, the original Influence-Passivity algorithm requires the appearance of hyperlinks in each tweet, which may not fit well in the

current scenario. As an alternative, a composite score called *Klout*[18] takes most of those measures into account and is publicly available. Therefore, each author's Klout score is included in author features.

Moreover, not all users are equally active. As described in section 4.1.1, temporal weight is applied to author features, therefore the values are weighted w.r.t. the elapsed time since his last activity in the community (i.e. writing or sharing a tweet related to the topic).

Content Features

Content analysis in the context of social network is more than pure language analysis as information is conveyed in a variety of formats. As a result, number of occurrences of platform-specific features for Twitter (retweet, mention, hashtag) as well as relevant keywords are kept track of.

Hyperlinks in tweets also play an important role in the process of information diffusion, as the content of external pages that is referred to can build better context for the topic of discussion and may motivate $\boldsymbol{U}$. In our practice, each tweet can either have a relevant link, an irrelevant link or no link. To determine whether a link is relevant, we rely on searching for event keywords in the web page that the link points to. If there is a match, the link is considered relevant; otherwise it is irrelevant. The count of hyperlinks in each tweet is therefore adjusted to 1, -1 and 0 for the three cases, respectively.

We also compute the extent of subjectivity of those tweets as part of the content features. The reason is that we can study if there is any preference of objective,

[18] `http://klout.com`

fact-sharing messages to subjective, emotional messages in terms of information propagation and thus attracting user to the community. As measuring subjectivity is a non-trivial task in the study of natural language processing [103], a simple heuristic is designed, focusing on two groups of explicit features. The key components used towards the score calculation are the subjectivity of (word, part-of-speech tag) pairs and that of emoticons found in the tweet. For the former, we start by feeding tweets into a part-of-speech tagger [132], keep all content words (noun, verb, adjective or adverb) and then classify those word-tag pairs using a pre-compiled subjectivity lexicon [143]. Entries in the lexicon are labeled as either strongly subjective or weakly subjective, and we assign 2 points to each strongly subjective pair, 1 point to each weakly subjective pair and 0 point otherwise. For the latter component, we compiled a lexicon[19] which holds more than 130 commonly-used emoticons. The scoring scheme for emoticon is the same as that for word-tag pair. The final subjectivity score for a tweet m is computed as the average of those segments' scores:

$$S_{score}(m) = \frac{\sum_{(w,t)\in WT(m)} subj_{pair}(w,t) + \sum_{e\in EMOT(m)} subj_{emot}(e)}{|WT(m)| + |EMOT(m)|} \quad , \tag{4.1}$$

where $WT(m)$ is the list of word-tag pairs in m, and $EMOT(m)$ is the list of emoticons in m.

Content analysis is further enriched by linguistic cues in text, which are extracted from analysis through Linguistic Inquiry and Word Count (LIWC)[20] dictionary. LIWC provides statistics of words grouped by grammatical (e.g. preposition)

[19] `http://www.cs.umbc.edu/courses/331/spring10/2/hw/hw7/hw7/data/sentislang.txt` `http://en.wikipedia.org/wiki/List_of_emoticons`

[20] `http://www.liwc.net`

or semantic (e.g. words that describe an *occupation*) components. We apply Principle Component Analysis (PCA)[21] to find out top 3 features in the LIWC analysis results, which are included as content features.

Moreover, as described in section 4.1.1, temporal weight is applied to content features. Here content features are computed for content posted by active friends of $\boldsymbol{U}$.

4.2.4 Model Fitting

As there are two possible outcomes of user participation behavior and all the aforementioned features take real values, we treat the User Engagement in Topic Discussion Problem as a binary classification problem operated on feature vectors of the following format.

- Label: Fact of whether the user joining the community or not. The value for is binary variable can be either positive or negative, and it serves as the class label.
- Community Features:
  - *wccSize*: Size of the WCC which the user belongs to in the active network.
  - *wccPercent*: Ratio of the WCC's size to that of the whole active network.
  - *connectivity*: Number of active friends (i.e. followees) in the active community.
  - *communitySize*: Size of the active community.
- Author Features:

[21]Modified from `http://www.neuroshare.org`

- *logFollower*: Logarithm of the weighted geometric mean of active friends' counts of followers.
- *logFollowee*: Logarithm of the weighted geometric mean of active friends' counts of followees.
- *klout*: Weighted means of active friends' Klout scores.

- Content Features:
  - *url, retweet, mention, hashtag, keyword*: Weighted means of the counts of relevancy-adjusted url, retweet, mention, hashtag, keyword in all active friends' tweets.
  - *sentiment subjectivity*: Weighted mean of sentiment subjectivity score.
  - *pca1, pca2, pca3*: Weighted means of the top 3 PCA features on LIWC results applied to all active friends' tweets.

The temporal weight vector is set as $(1, 0.8, 0.6)$. That is, assuming the current slice of consideration is slice k, the temporal weight for each tweet is 1, 0.8, or 0.6 if it was written in slice k, $k-1$ or $k-2$, respectively. Any tweets written earlier than two slices ago are no longer considered active.

Algorithm 5 describes the pseudo-code for generating all records for the classification problem.

Algorithm 5 Classification feature record generation

Require: Dataset D

$timeWgt \leftarrow (1.0, 0.8, 0.6)$ {Temporal weight}
$winLen \leftarrow 3$ {Active window length}
for all Slice $S \in D$ **do**
  for all Author $A \in S$ **do**
    $activeCommunity[S.id] \leftarrow activeCommunity[S.id] \cup \{A\}$
  end for
  for all $I = 1$ **to** $\min(winLen - 1, S.id)$ **do**
    $activeCommunity[S.id - I] \leftarrow activeCommunity[S.id - I] \cup activeCommunity[S.id]$
  end for
end for
for all Slice $S \in D$ **do**
  for all Author $A \in S$ **do**
    for all $I = 0$ **to** $\min(winLen - 1, D.size - S.id - 1)$ **do**
      $activeNetwork[S.id + I] \leftarrow activeNetwork[S.id + I] \cup \{A\} \cup A.followers$
      for all User $F \in A.followers$ **do**
        $F.activeFriends[S.id + I] \leftarrow F.activeFriends[S.id + I] \cup \{A\}$
        for all Tweet $T \in A.tweets[S.id]$ **do**
          $F.partialRecords[S.id + I] \leftarrow F.partialRecords[S.id + I] \cup \{timeWgt[I] \times (A.features[S.id], T.features)\}$
        end for
      end for
    end for
  end for
  for all User $U \in activeNetwork[S.id]$ **do**
    print $U.id$
    if $U \in activeCommunity[S.id]$ **then**
      print ‘‘Positive’’
    else
      print ‘‘Negative’’
    end if
    print $activeNetwork[S.id].wccSize[U]$
    print $activeCommunity[S.id].size$
    print $U.activeFriends[S.id].size$
    print $\text{avg}(U.partialRecords[S.id])$
  end for
end for

4.3 Experiments

4.3.1 Data Collection

Tweet stream for topics was crawled with Twitter's Search API[22] using an initial seed of manually compiled keywords and hashtags relevant to the event. For a keyword k, we crawl all tweets that mention k, K, #k and #K. The seed list of keywords and hashtags is kept up-to-date by first automatically collecting other hashtags and keywords that frequently appear in the crawled tweets and then manually selecting highly unambiguous hashtags and keywords from this list. We avoid the query drift problem by placing a human in the loop to ensure that ambiguous keywords are not crawled outside of context but only in combination with a contextually relevant keyword.

Data crawl was performed at fixed time intervals, here every 30 seconds. For every issued query, the Twitter search API responds with 1500 tweets. Crawling at regular and frequent intervals allows us to make an assumption that the data collected is a close approximation of the actual population of the tweets generated for the event in that time period. We also crawl the social graph (i.e. follower list) of these tweet posters, who are part of this event-oriented community at specific timestamps of the day. Duration for the time gap between subsequent snapshots of the network for different communities depend on the type of event. We also collect tweet posters' profile information like location, followers and followees counts, description about the tweet poster, etc. For those users who activated privacy setting, no information was crawled, and their tweets are discarded from the slice.

[22] `https://dev.twitter.com/rest/public/search`

A total of 14 events are considered, and information of these communities are crawled. They were popular topics (i.e. buzz words) at the period of crawling[23], showing steady growth in number of related tweets in the real-time search result. Furthermore, they are all social events with impacts beyond the online world. For most of predefined events the crawl was started in advance and extended after the event duration. The following list introduces each event and its categorization as defined in section 4.2.2. Due to the space constraint, only a summarization is provided.

- **ClevelandShowPremiere**: Second Season premiere of animated TV series *Cleveland Show*. September 26. Global, loose, deterministic, transient.
- **DiscoveryBuildingCrisis**: Hostage crisis at the headquarters of Discovery Channel, Maryland. September 1. Local, loose, unexpected, transient.
- **EmmyAwards**: 62nd Prime-time Emmy Awards. August 29. Global, loose, deterministic, lasting.
- **GoogleInstantSearch**: Launch of Google Instant in United States. September 8. Global, loose, unexpected, transient.
- **HeismanTrophy**: Reggie Bush's announcement to forfeit 2005 Heisman Trophy. September 14. Local, compact, unexpected, lasting.
- **IowaStateFair**: Iowa State Fair. August 12-22. Local, loose, deterministic, lasting.
- **JewishNewYear**: Jewish New Year 5771. September 8-10. Global, compact, deterministic, transient.

[23]August and September, 2010

- **LindsayLohanHearing**: Lindsay Lohan's hearing on probation revocation and verdict. September 24. Local, loose, deterministic, transient.
- **LinuxCon**: Annual convention organized by Linux Foundation. August 10-12. Global, compact, deterministic, lasting.
- **LondonTubeStrike**: London tube strike. September 6. Local, loose, deterministic, transient.
- **RichCroninDeath**: Death of singer and songwriter Rich Cronin. September 8. Local, loose, unexpected, transient.
- **ScottPilgrimRelease**: Release of movie *Scott Pilgrim vs. the World*. Aug 13. Global, loose, deterministic, lasting.
- **SESSanFrancisco**: Search Engine Strategies 2010 at San Francisco. August 16-20. Global, compact, deterministic, lasting.
- **StuxnetWorm**: Confirmation of Stuxnet worm attack on Iranian nuclear program. September 24. Global, loose, unexpected, lasting.

Macro-level Summaries

Table 4.1 summarizes various statistics for all events. #tweet is the total count of tweets crawled. Following number of unique authors are the percentage of tweets having relevant url, mention, retweet and emoticon, respectively. Average subjectivity score is also reported here. The last two columns records the average size of active community over each slice and the average connectivity over each record.

Events	#tweets	#unique users	%relevant url	%mention	%RT	%emoticons	average. subj. score	average active community size	average connectivity
ClevelandShowPremiere	1494	1221	19.26	25.97	11.85	6.16	0.28	686.23	1.28
DiscoveryBuildingCrisis	3303	2580	48.97	12.14	35.06	5.60	0.19	1497.87	2.67
EmmyAwards	5027	3453	65.12	11.06	17.57	6.47	0.18	1126.39	3.10
GoogleInstantSearch	4058	3429	63.05	9.78	23.48	4.09	0.14	1611.79	3.32
HeismanTrophy	5631	4261	32.23	9.06	33.17	2.26	0.16	2487.05	2.61
IowaStateFair	2470	1106	33.59	36.92	21.05	8.54	0.20	349.72	4.83
JewishNewYear	7676	6251	17.18	19.72	25.68	9.64	0.23	3097.96	2.51
LindsayLohanHearing	5547	3660	55.39	6.99	27.08	3.19	0.13	1210.49	1.95
LinuxCon	1294	418	36.14	18.86	33.69	4.71	0.17	226.85	3.11
LondonTubeStrike	1186	530	56.70	15.00	18.47	10.96	0.15	161.6	1.35
RichCroninDeath	476	379	25.16	30.46	25.42	18.70	0.24	215.06	1.16
ScottPilgrimRelease	21435	14286	31.30	13.21	21.63	8.84	0.17	3979.91	2.79
SESSanFrancisco	1383	462	85.62	5.28	10.48	4.99	0.09	157.89	2.59
StuxnetWorm	2845	1855	70.91	8.19	21.83	6.40	0.17	1458.85	3.29

Table 4.1: Statistical summarization for data sets

4.3.2 Feature Vector Processing

First, all records are generated as described in Algorithm 5. Then, values of the six non-PCA content features are standardized using z-score.

We randomly sample 70% of the records for training and the rest for testing. As most information recipients did not join the community eventually, we experienced huge imbalance in terms of class labels: there are way more negative records than positive ones. To eliminate the impact of imbalanced dataset on training process, SMOTE [23] with over-sampling ratio 400% is applied to positive records in the training set. After that, random under-sampling on negative records is performed for both training and testing sets to make the class distribution balanced. Finally, all numerical values are scaled to the range (-1,1), and the records are ready for evaluation. This setting is applied to dataset of each event-oriented community with an exception that the over-sampling ratio for event *ScottPilgrimRelease* is changed to 100% for the purpose of computational efficiency.

4.3.3 Evaluation Settings

We run the experiments to analyze the role played by the various features and how they help us to predict whether a user will engage in the topic discussion. We use LibSVM [21] to build SVM classifiers (Gaussian RBF kernel with $\gamma = 8$ and cost $c = 32$) based on the following feature subsets to see how they perform on the prediction task. For each feature subset, the experiment is repeated five times and average accuracy rate is computed. We run the following experiment groups:

- *allFeatures* (All): contains all three feature groups.
- *onlyContent* (Con.): contains only content feature.

Events	**All**	**Con.**	**Aut.**	**Com.**		
DiscoveryBuildingCrisis	**77.86**	75.95	71.31	69.65	U	L
GoogleInstantSearch	**76.25**	74.92	72.23	52.60	U	L
RichCroninDeath	**90.68**	**90.96**	**90.36**	68.47	U	L
StuxnetWorm	**76.05**	**76.46**	72.05	57.51	U	L
HeismanTrophy	**76.88**	75.28	69.94	61.85	U	C
ClevelandShowPremiere	**86.11**	**85.77**	**85.65**	67.36	D	L
EmmyAwards	**77.00**	**77.39**	70.93	56.23	D	L
IowaStateFair	83.34	**84.25**	81.62	70.09	D	L
LindsayLohanHearing	**80.09**	79.30	77.22	52.57	D	L
LondonTubeStrike	**82.40**	**82.96**	80.07	56.22	D	L
ScottPilgrimRelease	**78.16**	**77.86**	75.32	59.81	D	L
JewishNewYear	**75.15**	74.14	69.16	55.63	D	C
LinuxCon	80.77	**82.17**	76.97	71.97	D	C
SESSanFrancisco	**75.50**	**76.40**	71.69	58.34	D	C

Table 4.2: Summary of prediction accuracy (%)

- *onlyAuthor* (Aut.): contains only author feature.
- *onlyCommunity* (Com.): contains only community feature.

4.3.4 Evaluation Results

Table 4.2 demonstrates the accuracy achieved by SVMs on different topics and feature sets. For each event, the highest accuracy score is in bold. Moreover, any classifier which is considered equivalently good as the highest-scoring classifier by the sign test is also in bold. We calculate the statistical significance of the improvement by performing *paired binomial sign test* on two classifiers. The smaller the p-value, the stronger evidence it is that one classifier has performance improvement over another. The p-value threshold is 0.05. Characters in the last two columns stand for U(nexpected), D(eterministic), L(oose) and C(ompact).

Our observations on experiments are listed here:

1) We observe performance of onlyCommunity classifiers being worst. A possible explanation for that is the latent nature of network features, which makes them difficult to be perceived by a user directly and thus have lesser effect on user engagement.

2) The onlyContent classifiers give the best performance over other single group features, especially compared to onlyCommunity classifiers. One reason for content being the dominant feature for predicting participation in a discussion is the fact that some users end up participating in a discussion based on observing the information from the public timeline, and therefore, these ad-hoc users are hard to observe via network analysis only. Moreover, content is engaging by its quality and nature (*information sharing* or *call for an action* or *crowd sourcing*). For example, link to an image or video (an evidential content) about Reggie Bush's surrender of Heisman Trophy in September, 2010 is likely to provoke lot more thoughts in a user's mind to engage in the discussion.

3) We observe comparable performance of onlyAuthor classifiers as onlyContent classifiers do for some of the topics. Here potential reason for this observation is the effective presence of influential people in the discussion group. Hence, insufficiency in content features, reflected by low average connectivity, can be compensated by author features (e.g., Rich Cronin Death).

4) Using robust statistical significance testing method, we observe for 12 out of 14 topics, allFeatures classifiers have better or equivalent performance over any single feature group classifier. In some cases (e.g. Discovery Building Crisis, a

very evolving topic discussion group), the advantage is dominant, where degree of randomness in individual dimensions can be really high. Therefore, it suggest usefulness of allFeatures classifiers here.

5) We find no significant correlation between user engagement to topics and the selection of feature groups, whether the event type is *lasting* or *transient*. On the other hand, the advantage of allFeatures classifiers over other factor groups is generally stronger on the *unexpected* topics than the *deterministic* ones. Moreover, it is discovered that the performance of onlyAuthor is relatively better, explained by a closer gap to the best classifier, for *loose* events than for *compact* events.

4.4 Extension for Volume Prediction

In this section, we discuss our extension of user engagement prediction to predicting the amount/volume of engagements. Instead of the question *whether a user will write tweets or not in the given time frame* which has a coarse granularity, we aim at answering the question *how many tweets will a user write* which presumably can provide a more precise estimate of social network dynamics. Here, we focus on two tasks: predicting the *microscopic* (individual) and *macroscopic* (collective) volume of topical tweets that Twitter users will generate within a time frame.

Most features used in the classification task are considered here, with changes as described below.

- Community Features: Total tweet volume of user's active friends is added as a natural extension to connectivity (i.e. the number of active friends). We also include the number of interactions (mentions and retweets), as interactions shall reflect much stronger ties than ordinary follower-followee relationships.

Topic	Period	# Tweets	# Unique Authors
IAC	11/06 - 12/02	93,525	19,705
JSS	11/06 - 11/30	251,316	152,174
OWS	11/06 - 12/02	2,042,653	320,415

Table 4.3: Datasets statistics

Moreover, a subset of highly-engaged users (users whose tweet volumes are within the top 3% percentile of all users in the active network) is identified, and the count of highly-engaged friends is listed as a separate feature.

- Content Features: Besides adjusting the count of URLs by their relevancy to the event, we also count the number of hyperlinks that point to multimedia contents. This is based on heuristics that multimedia materials may have stronger effect on user engagement, compared with text alone.

- Past Activity: We include the knowledge of users' historical tweet volume in order to capture users' inertia of engaging in the event-oriented discussion more. Past activity will be referred to as "past", for the expository purpose.

Tweet streams based on three events that were frequently discussed in late 2011 are crawled: India's anti-corruption movement (IAC), the abuse scandal of former Penn State football coach Jerry Sandusky (JSS), and Occupy Wall Street movement (OWS). Table 4.3 lists basic information of the three datasets.

For each event, we build regression models using feature vectors for each user, where each feature is a regressor, and the tweet volume of the user is the regressand. Compared with other tools, linear regression has multiple advantages including higher efficiency, low storage overhead and statistical interpretability on model coefficients.

As it is impossible to evaluate the exponential number of all feature combinations, we study each feature group (content, community, author, pass activity) as a unit. One common measure of the regression model's power is adjusted R^2 value (R_a^2). For a regression model with n records and p regressors (i.e. features), it is defined as

$$R_a^2 = 1 - (\frac{n-1}{n-p})\frac{SSE}{SSTO} \ , \tag{4.2}$$

where $SSTO$ and SSE are sum of squares of regressands and residuals, respectively [77]. Higher R_a^2 value indicates that a larger proportion of total sum of squares is explained, thus a higher explanatory power of the model.

4.4.1 Individual Volume Prediction

We first tested the prediction of individual users' relevant tweet volume. Two days of past information (i.e. $h = 2$) are used for model building, and R_a^2 values are computed over the period. Figure 4.2 shows the results of five models using different selections of features. The name of each model indicates which feature groups it uses. As observed from the plot, higher R_a^2 values are obtained when extra features are added on top of past activity. Another finding is that author features introduce additional explaining power beyond network features and content features. Although there is no guarantee of causality, it may suggest that the motivation behind users' involvement in topical discussion is attributed more to the general influence of friends than the specific content.

R_a^2 value could be inflated are the number of regressors increases. To address this concern, we further performed *partial F-tests* on the full model against a simple strawman that uses past activity alone. The null hypothesis H_0 is that all additional

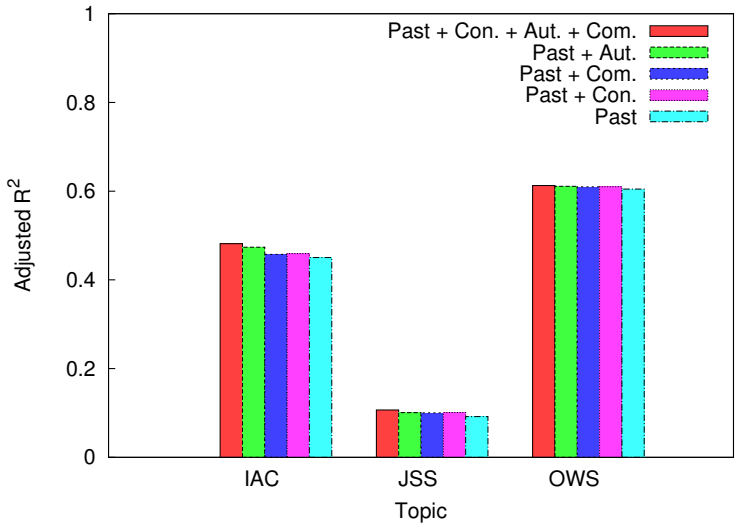


Figure 4.2: R_a^2 on microscopic prediction with different feature groups, two days of past activity

Topic	$\max P(f \geq F^* \mid H_0)$ Full vs. Strawman
IAC	2.094323×10^{-3}
JSS	1.142266×10^{-10}
OWS	$9.977034 \times 10^{-156}$

Table 4.4: Partial F-tests results

features' coefficients are zero, and a statistic F^* will follow an F distribution if H_0 holds [77]. As shown in Table 4.4, for all topics the conditional probability $P(f \geq F^* \mid H_0)$ never exceeds 10^{-2} on any single day's data. Therefore, we reject H_0 and conclude that the additional explaining power from extra features is statistically significant.

For the JSS dataset we note that the overall user-level (microscopic) prediction accuracies are low. We should point out that on this dataset the average number of tweets per user is under 2. Thus, there is by and large insufficient information on most

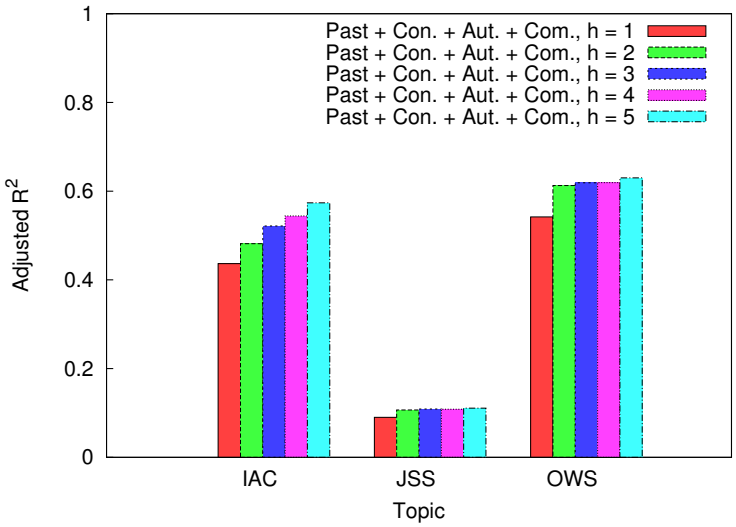


Figure 4.3: R_a^2 on microscopic prediction with varying past activity length (h)

users to predict how much they will tweet on this topic. However, it is interesting to note that if we look at a subset of the users that tweet more frequently (> 5 tweets on this topic, results not shown) and also when one aims at predicting the output of the collection of users in its entirety, the accuracy increases significantly (see Figure 4.4 and the discussion below).

To investigate the impact of past information amount on model performance, we ran another set of experiments where the number of days of past activity available (parameter h) was varied from 1 to 5. Figure 4.3 shows the result. The first observation is that the more past information is available, the higher R_a^2 values. A second observation is that improvement from additional past information is often diminishing, suggesting that recent information has larger influence than older. Such a finding is consistent with those from previous works.

4.4.2 Collective Volume Prediction

Finally, we present the results on predicting the behavior of users *en masse*. For a day of prediction, we use the coefficients learned from previous day's regression model to fit the newly observed feature vectors on that day. Then we compute the accuracy value as

$$Accuracy = 1 - \frac{|\sum_{\boldsymbol{U} \in \boldsymbol{D}} [\hat{vol}(\boldsymbol{U}) - vol(\boldsymbol{U})]|}{\max(\sum_{\boldsymbol{U} \in \boldsymbol{D}} \hat{vol}(\boldsymbol{U}), \sum_{\boldsymbol{U} \in \boldsymbol{D}} vol(\boldsymbol{U}))} \ , \tag{4.3}$$

where $\boldsymbol{D}$ is the set of candidate users, and $vol(\boldsymbol{U})$, $\hat{vol}(\boldsymbol{U})$ are the actual and estimated tweet volume of user $\boldsymbol{U}$, respectively.

Figures 4.4 and 4.5 show results with varying models and h values, respectively. For each topic, the average accuracy over days is reported. Compared with that on microscopic prediction, the performance of topic JSS has significant improvement. Again, the trend of diminishing return on the amount of past information is observed.

4.5 Conclusion

In this chapter, we present a systematic investigations into factors impacting user engagement in topic discussion on social media. We study user engagement as their participation in event-oriented community and build an effective prediction model to estimate the engagement decision as well as the volume of event-oriented tweets as a result of the user engagement. Evaluations on a large number of Twitter event-oriented communities demonstrate that the capabilities of content, user, network features and past activity vary greatly, motivating the incorporation of all the factors. Therefore, a strong need can be felt to study dynamics of user engagement by using

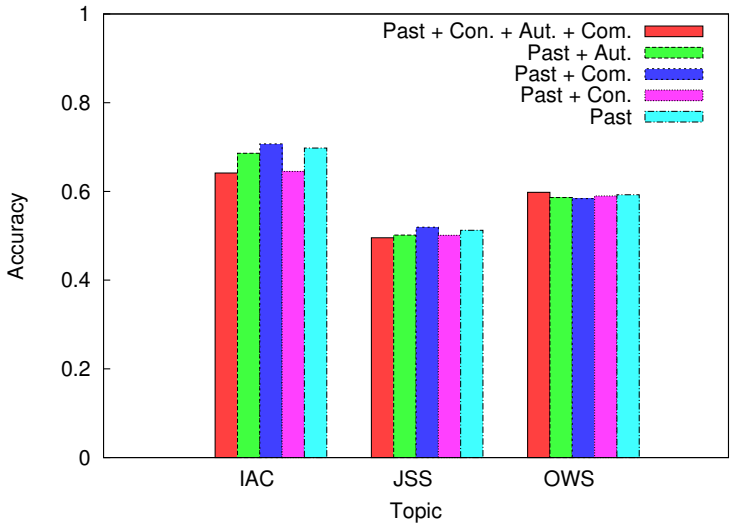


Figure 4.4: Accuracy of macroscopic prediction with different feature groups, two days of past activity

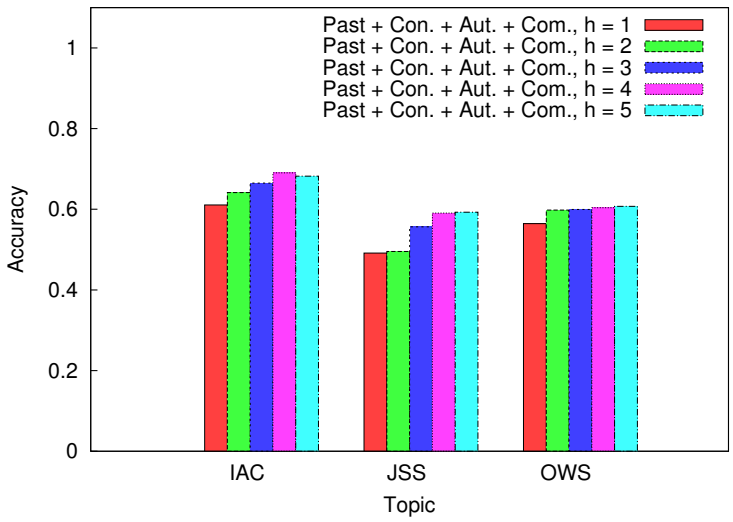


Figure 4.5: Accuracy of microscopic prediction with varying past activity length (h)

the integrated framework. Moreover, we find correlations between event types and features, which can help understand user engagement in better scientific ways.

Future research should take the following points into consideration:

- Experiments on events with more diverse characteristics for better understanding of the relation between event type and user engagement factors. Analysis of related events can help in understanding how topics around events evolve over time and shift the characteristics from one event to another.

- Sophisticated semantic analysis on user-generated content to provide content features. For example, we can resort to external knowledge base like Wikipedia to build proper context for discussion topic and then assess content quality to get better insight into impact of content features on user engagement.

- Methods to resolve user profile information's heterogeneity (e.g. missing or outdated value, adversarial content) and profile types (news, trustee etc.), and their use as people features.

- Application of the analysis framework to other OSNs such including Facebook, LinkedIn, and StackOverflow.

- Expanding the event-oriented model to generic framework to identify users' engagement in various co-occurring events during that timeline.

Chapter 5: Understanding Content Divergence in Online Social Group Discussion

In Chapter 1, we describe the motivation of studying content divergence in the discussion of online social groups. When characterizing online social group dynamics, most studies [6, 111, 123, 69] only investigate the implications of the network structure, and they lack the insights of dynamics based on user-generated content. In this chapter, we take a new perspective on characterizing group dynamics based on divergence of group discussion topics.

We have also underlined the importance of linking well-established socio-psychological literature with OSN analytics (Section 1.1). Social scientists have defined the groups based on various common user characteristics. Here, we define a *group* as the set of users interacting in discussions about a real-world event. We refer to *group discussion divergence* as collectively diverging behavior in user-generated discussion topics, and it is quantitatively measured as the Jensen-Shannon divergence among latent topic distributions of a group's messages (more details in Section 5.2). Understanding of such collective behavior in discussions around events can lead to actions of prioritizing for engagement, such as whom to engage with in communities for specific needs during disaster response coordination, and for specific concerns and advocacy in brand management.

In particular, we ask the following questions:

- How does the divergence of user discussion in a group change over time, within and across different phases of events?
- Do existing theories of social group behavior have implications on the evolution of group discussions?
- Can we predict the change of group discussion divergence in the future?

Answers to the above questions can aid in understanding which factors contribute more in facilitating cohesion (lower divergence) in social group discussions. They also enable us to predict the change of group discussion divergence, which in turn allows fast identification of groups whose voices are showing less divergent shifts. Such techniques may be highly valuable in scenarios like natural disaster response, where a small number of less diverging, focused groups (with resource requests or information supplies) need to be identified efficiently, so that their input will not be buried under an overwhelming amount of noise in the social content stream. Moreover, understanding of these factors will help us decipher behavior of self-organizing online social groups.

Using Twitter as our experimental platform, we propose a systematic approach to analyze discussions in online social groups, and understand the pattern of how discussion divergence changes over time. We discover that the divergence of topics in user-generated content starts with a low value prior to the event, peaks during the event, and fades away after the event.

We also formally define structural and user features guided by social cohesion and social identity, the two socio-psychological theories on group dynamics to be discussed

in the next section. We represent a group's structural features related to social cohesion using network characteristics of its friendship-follower network. User features related to social identity are modeled via self-presentation in user profiles, capturing group members' physical world identities, as well as their online identities. These features incorporate guidance from the two theoretical approaches, while capturing users' social behavior from both physical and online worlds. We study the relation between group discussion divergence and proposed features via correlation analysis. We observe that a group's network density, average length of pair-wise shortest path, and entropy values of user identities are well-correlated with its discussion divergence.

Finally, we build machine learning models to predict the future increase or decrease of social groups' discussion divergence values by using features discussed above. Our classifiers are able to achieve an AUC of 0.84 and an F-1 score of 0.8, reflecting a 33% improvement from the baseline method. As discussed earlier, this work can help in various application domains, including identification of emergent concerns during disasters, and the self-organizing group behavior of discussions.

5.1 Related Work

First, we briefly introduce two theories proposed by socio-psychologists to explain the dynamics of traditional face-to-face social groups and their behaviors. We envision that their roles in shaping user engagement in groups [39] will contribute to our understanding of group discussion divergence. Then we describe related work on online social group bonding and dynamics.

Socio-Psychological Theories. The social identity theory includes two closely related parts: *social identity* [129] and *self-categorization* [136]. In [129], Tajfel

defines the concept of social identity as "the individual's knowledge that he belongs to certain social groups together with some emotional and value significance to him of this group membership". Therefore, group membership is the result of **"shared self-identification"** rather than "cohesive interpersonal relationship", and such shared identity leads to cohesiveness and uniformity, among other features [135]. One commonly-cited piece of evidence for the social identity theory is team sports [16], where teammates are representing the same organization (a school, a club, or a country) and they are well aware of desire to sustain the reputation of their associated identity. In contrast, the social cohesion theory views social groups from a different perspective. Its hypothesis is that the necessary and sufficient condition for individuals to work as a group is the **cohesive social relationships between individuals**. We adopt the definition by Lott and Lott [88] that interprets cohesiveness as *mutual attraction* between individuals, which is slightly different from that used in [45]. In accordance with this definition, the positive correlation between group cohesion and performance has been reported in various types of groups [98, 12]. A social cohesion example will attribute the inter-personal friendship between teammates of a sports club as the reasoning factor for group performance and its evolution.

User-Group Bonding. One study relevant to our work is by Grabowicz et al. [55], where authors discussed methods to translate the common identity and common bond theories for group attachment into general metrics applicable to large social graphs. They also devised a method to predict whether a group is social (formation dependent on interpersonal bonds) or topical (formation based on role awareness). Prior to that, Ren et al. [111] presented a study on the similar direction, focusing on the implications of the two theories of group attachment and link these theories

with design decisions for online communities. Our differing objective here is to rather analyze a group's discussion having the characteristics of identity and cohesion features instead of predicting group type or evaluating community design decisions. In a similar spirit, Farzan et al. [43] studied group commitment on Facebook within a controlled environment and observed that designs that encourage relationships among members or emphasize the community as an entity both increase the commitment and retention of players. Budak and Agrawal [18] utilized data analytics and user survey to study factors that drive group chats on Twitter, and found that social inclusion contributes most to user retention. Our objective here is slightly different, in that it focuses on the effects of group commitment in discussion divergence in the communities emerging around real-world events.

Group Dynamics. Most prior work on group dynamics has focused on structural dynamics. Notably, Backstrom et al. [6] proposed a structure-centric model for network membership, growth and evolution by analyzing DBLP and LiveJournal social networks. Their findings show how individuals join communities and how communities grow depending on the underlying network structure, which supports cohesion-based structural features in our study. Taking a different path of a user-centric approach, Shi et al. [123] studied the user behavior of joining communities on online forums. Among other features, the authors studied the similarity between users and the similarity's relation with community overlap. They found that user similarity defined by the frequency of communication or number of common friends was inadequate to predict grouping behavior, but adding node/user-level features could improve the fit of the model. Kairam et al. [69] analyzed long term dynamics of communities and modeled future community growth rate. They found that growth

rate is correlated with current size and age of a group and the size of the largest clique is the best feature for community sustainability. Relevant efforts on understanding individual-level characteristics include a study by Rao et al. [110], where authors presented an approach for automatic creation of ethnic profiling of users, focusing on names as the key factor. Pennacchiotti and Popescu [106] also proposed a machine learning approach for user classification on Twitter by analyzing user's friends, user posts and profile information. These studies of group and individual characteristics provide a base for the modeling of user and structural features in our study.

5.2 Problem Formulation

In this section we describe preliminaries including event-based discussion collection, social group identification, measure of group discussion divergence, and a formal specification of the prediction task. Feature design, experiment results and analyses are presented in subsequent sections.

5.2.1 Data Collection

We focus on Twitter user-generated contents and discussions based on particular real-world events, and thus, proper filtering of the generic content stream is required.

We implemented a Twitter Streaming API-based crawler that collected an ongoing tweet stream relevant to the event based on a seed keyword set, similar to [118]. For a keyword `k`, we crawl all tweets that mention `k`, `K`, `#k` or `#K`. The seed list of keywords and hashtags is kept up-to-date by first automatically extracting the top-N most frequent hashtags and keywords from the crawled tweets, and then manually selecting and adding highly unambiguous hashtags and keywords (e.g. `hurricane sandy`, `#sandy`, `#ows`). This process provides a control for contextual relevance of

Event Name	Duration	#Tweets	#Users	Type
Hurricane Irene (Irene)	08/24-09/19, 2011	183K	77K	Transient
Hurricane Sandy (Sandy)	10/27-11/07, 2012	4.9M	1.8M	Transient
India Anti-Corruption (IAC)	11/05-12/02, 2011	100K	21K	Lasting
Occupy Wall Street (OWS)	11/05-12/02, 2011	2.1M	331K	Lasting

Table 5.1: Twitter data statistics centered on diverse set of evolving events

tweet content to the event. Tweets containing seed hashtags/keywords and their corresponding authors then become our dataset. We also store metadata associated with the dataset, such as each author's location, followers/friends, and profile description.

In this study, we choose four events relating to natural disaster, social activism, or political campaign, and collect relevant data using the mechanism described above. Table 5.1 summarizes basic information about each dataset. We note that events possess varying characteristics on the dimensions of activity, social significance, participant types, etc. In Table 5.1, we specifically show temporal feature values as "Lasting" and "Transient" that denotes how enduring an event is. For example, the Occupy Wall Street movement was highlighted in social media discussion for a long time frame, while Twitter users' attention to Hurricane Sandy quickly decreased significantly after it dissipated.

To enable temporal analysis and reasoning, tweets are grouped into three phases (*pre-*, *during-*, and *post-event*). Our categorization of phases for each event is aligned with its real-world timeline, and Table 5.2 shows the occurrences leading to phase division.

Event	Timeline
Hurricane Irene	*During-phase* Beginning (08/27): Landfall in North Carolina *During-phase* End (08/30): Hurricane dissipated
Hurricane Sandy	*During-phase* Beginning (10/29): Landfall in New Jersey *During-phase* End (10/31): Hurricane dissipated
India Anti-Corruption	*During-phase* Beginning (11/24): Minister Sharad Pawar got slapped due to alleged corruption *During-phase* End (11/29): No further substantial tweet w.r.t. the incident of slapping
Occupy Wall Street	*During-phase* Beginning (11/15): Raid of Zuccotti Park *During-phase* End (11/23): President speech interrupted by protesters

Table 5.2: Timeline and dates signifying the beginning and end of *during-event* phase of each event

5.2.2 Identifying Social Groups

Social groups can be defined in many ways. Our focus here lies on those groups of people who interact (and potentially emerge) in the times of evolving real-world events.

Therefore, given all users in a community formed around discussions of an event, it is necessary to identify appropriate social groups on which quantitative analyses will be performed to understand the dynamics of group discussion divergence. Resultant social groups should reflect online interaction among users that is beyond simply using the same word in their tweets. Moreover, the grouping criterion needs to be independent of any feature of social structure and user characteristics due to some of our features being based on social cohesion and identity phenomena (defined in the following sections), so that the results are not biased.

To that end, we propose an approach of clustering users based on their interactions, which can be either *retweet*, *reply* or *mention*. An interaction graph is created to represent those relationships during each phase of the event, where vertices stand for users and edges indicate at least one interaction between two users through the phase. We apply Markov clustering [120], a commonly-used community detection algorithm to identify social groups. Only groups that have at least 10 members and are active (that is, at least one member posts a relevant tweet by mentioning event-related keyword(s)) for at least two days are retained. Again, while there exist other choices of identifying latent online user groups without ground truth labels, we believe our simple approach can effectively capture online interactions and yield meaningful groupings of users. Table 5.3 summarizes the information of each dataset's social groups.

Event	# **Groups**	# **Users**	**Average Group Size**
Hurricane Irene	137	22,068	161
Hurricane Sandy	4,947	284,062	57
India Anti-Corruption	76	7,907	104
Occupy Wall Street	6,202	296,279	48

Table 5.3: Information of social groups

5.2.3 Defining Group Discussion Divergence

We use Jensen-Shannon divergence (JS-divergence) to quantify the divergence of group discussions. Compared with other information-theoretic measures such as Kullback-Leibler divergence, JS-divergence is always defined, bounded, and can be generalized to more than two distributions [86]. JS-divergence has long been employed

in computational linguistics [85, 89], though its usage in social network analytics has been limited.

In order to calculate the JS-divergence, we first construct a dynamic topic model [14] and infer the topics of discussion. Input into the topic model is a collection of vocabulary vectors, each of which represents one event-related tweet and is indexed by discrete time-stamps. The vocabulary includes words and phrases pertaining to the event, as well as hashtags with the leading '#' symbol stripped. The dynamic topic model has the advantage of modeling a systematic topic shift (due to event's progress) automatically, which allows us to investigate the true difference of an individual member's topic distribution to the corresponding group's topic distribution at any given time.

For topic inference, we use the `dtm` package[24] with default parameters. We evaluated results from 2 to 5 latent topics, and found that topics become similar and redundant after 3. For expository simplicity we use 3 as the default number of topics and report the top vocabulary in the different event phases for two events (Hurricane Sandy and Occupy Wall Street) in Table 5.4.

The inference process of the topic model returns a latent topic distribution for each tweet t, denoted as β_t. A group g's mean topic distribution at phase s over all its users' tweets (T_g^s) can then be calculated as:

$$\beta_g^s(i) = \frac{\sum_{t \in T_g^s} \beta_t(i)}{|T_g^s|}, \forall i = 1, \cdots, \text{number of topics} \tag{5.1}$$

and g's JS-divergence at phase s is defined as

$$JS(g^s) = H(\beta_g^s) - \frac{\sum_{t \in T_g^s} H(\beta_t)}{|T_g^s|} \tag{5.2}$$

[24] `https://code.google.com/p/princeton-statistical-learning/downloads/detail?name=dtm_release.tgz`

Hurricane Sandy			
	Pre-event	**During-event**	**Post-event**
Topic 1	tropical storm	red cross	red cross
	east coast	jersey shore	staten island
	canada	caused	mexico
	path	staten island	caused
Topic 2	new york	new york	new york
	state	new jersey	new jersey
	google	hurricane katrina	states
	android	media	hurricane katrina
Topic 3	frankenstorm	frankenstorm	frankenstorm
	halloween	fema	knicks
	east coast	halloween	fema
	atlantic	mitt romney	nyc
Occupy Wall Street			
	Pre-event	**During-event**	**Post-event**
Topic 1	occupy	occupy	occupy
	protest	n17	oo
	movement	nypd	occupyla
	occupytogether	brooklyn bridge	movement
Topic 2	movement	nypd	nypd
	us	movement	movement
	bahrain	protest	anonymous
	occupy movement	time	protest
Topic 3	occupy	occupy	p2
	oo	p2	tcot
	p2	tcot	republican
	tcot	oo	teaparty

Table 5.4: Top vocabulary representing the latent topics of discussions at each event phase

where $H(\bullet)$ is the Shannon entropy function (with log base 2) [86]. Intuitively, JS-divergence here gauges the divergence among topic distributions of a group's tweets. **The greater the JS value, the larger the difference and the stronger indication of a group lacking conformity in discussion.**

5.2.4 Prediction Problem Statement

Our goal is to solve a learning problem to predict the increase or decrease in the divergence of a group's discussion topics, measured by its discussion divergence, over an event's three phases: *pre-*, *during-*, and *post-event* (however, our analysis approach is applicable in general beyond the three phases of interests here). Specifically:

Given a real-world event E, a collection of N Twitter users discussing about E, and an assignment of them into K non-overlapping user groups $g_i (1 \leq i \leq K)$ based on interactions, predict the change of each group's discussion divergence $JS(g_i)$ between two consecutive event phases (that is, from pre-event to during-event or from during-event to post-event).

5.3 Feature Design

In this section, we describe the feature design driven by socio-psychological theories.

5.3.1 Structural Features Guided by Social Cohesion

To study the structural features driven by cohesion of social groups in a quantitative manner, we extract information from Twitter users' follower network. For each social group, we construct its corresponding node-induced sub-graph from the

follower network. Because the follower relation is directional, there are three groups of features:

- *Reciprocal*:

  An undirected edge will be created between two users only when both of them are following each other. This choice directly reflects the assumption of mutual interpersonal attraction in the social cohesion theory. Features here include density, transitivity[25], average clustering coefficient[26], and maximum average length of pairwise shortest paths over all connected components (short-named "average shortest path length").

- *Undirected*:

  An undirected edge will be created between two users if either of them is following the other. The underlying assumption is that one-way interpersonal attraction is sufficient to keep the social group sustained. The same group of features as in the reciprocal sub-graph are computed.

- *Directed*:

  We also compute density and transitivity on the directed sub-graph for each social group, without converting it to an undirected graph.

The range for all cohesion features is $[0, 1]$, except for the average shortest path length. Note that in existing sociology literature [97, 142] the term "structural cohesion" is a specific measure, defined as the minimum number of nodes one needs to

[25] transitivity $= \frac{3\times\text{number of triangles}}{\text{number of connected triples of vertices}}$

[26] clustering coefficient of node $i = \frac{2\times\text{number of triangles in } i\text{'s neighborhood}}{degree(i)\times(degree(i)-1)}$

remove from a graph to disconnect it. We do not include this feature as we find that almost all (more than 97% of total) social groups contain at least one fringe node (whose degree is one) or singleton, meaning that the value of this feature for most social groups will be at most one.

5.3.2 User Features Guided by Social Identity

To quantify the social identity-based features, we extract user's profile information as well as activity, as we note that social behavior tends to associate the user with established identities (regional, organizational, etc.) via self-representation and with incentive-based identity via user actions in the cyber-world. For example, "New Yorker" in a user's profile is an indicative signal of his location-based identity, and a profile containing "professional NBA player" or "Emergency Management" is highly suggestive of the user's occupational expertise. A user's action of adding such indicative terms into the profile suggests his self-awareness of the identity. Moreover, recently emerged social analytics services show online identities of users such as "celebrity" on Klout, "Mayor of a place" on Foursquare, etc., and users often tend to identify with them [35]. Thus, we are living with various social identities in both our *physical* as well as *cyber* world. We use location and description metadata in user profiles in addition to user actions (status updating, interacting, etc.) to extract the following types of social identities. Each identity type is modeled as a discrete attribute and for each social group under study, we compute the class distribution entropy for each identity and serve them as user features for the analysis. The range of identity features is from 0 to $\ln(C)$, where C is the number of unique classes in an identity type.

	Irene	Sandy	IAC	OWS
Directed Structural Features				
Density	0.04 ± 0.07	0.06 ± 0.08	0.02 ± 0.03	0.05 ± 0.04
Transitivity	0.23 ± 0.20	0.21 ± 0.23	0.10 ± 0.18	0.19 ± 0.23
Reciprocal Structural Features				
Density	0.03 ± 0.07	0.04 ± 0.07	0.01 ± 0.02	0.03 ± 0.04
Transitivity	0.16 ± 0.19	0.18 ± 0.24	0.07 ± 0.20	0.14 ± 0.24
Average Clustering Coefficient	0.06 ± 0.10	0.08 ± 0.12	0.02 ± 0.05	0.05 ± 0.09
Average Shortest Path Length	2.25 ± 1.19	1.83 ± 1.10	1.06 ± 0.99	1.56 ± 0.76
Undirected Structural Features				
Density	0.05 ± 0.09	0.07 ± 0.09	0.04 ± 0.04	0.06 ± 0.05
Transitivity	0.16 ± 0.16	0.19 ± 0.22	0.08 ± 0.15	0.16 ± 0.21
Average Clustering Coefficient	0.14 ± 0.13	0.13 ± 0.15	0.05 ± 0.09	0.10 ± 0.12
Average Shortest Path Length	2.72 ± 0.90	2.36 ± 1.06	2.01 ± 0.82	2.07 ± 0.64
User Features				
Regional Entropy	$2.71 \pm 0.78(5.28)$	$2.24 \pm 0.73(5.74)$	$2.06 \pm 0.45(4.94)$	$2.12 \pm 0.62(5.65)$
Expertise Entropy	$1.79 \pm 0.26(2.30)$	$1.08 \pm 0.46(2.30)$	$1.56 \pm 0.31(2.30)$	$1.50 \pm 0.27(2.30)$
Online Entropy	$0.97 \pm 0.21(2.08)$	$1.03 \pm 0.21(2.08)$	$1.24 \pm 0.24(2.08)$	$1.18 \pm 0.23(2.08)$

Table 5.5: Mean and standard deviation of structural and user features. Identity entropy upper bounds are listed in brackets.

- *Regional Identity feature*:

  Using location information in user profiles, we map users to regional classes that is sometimes used to represent self-identification in our daily lives — state-based (e.g., "Ohio" for Ohioans) and nation-based (e.g., "Brazil" for Brazilians). For creating feature value, we choose a user's state identity if it belongs to the host nation of the event (e.g., user from Buffalo will have "NY" as the identity value in the OWS event), otherwise, we choose the user's nation identity (e.g., user from London will have "UK" as the identity value in the OWS event). We use the Geonames dataset on Linked Open Data (LOD) cloud and Google Maps API to convert user profile locations into latitude-longitude, and then state and nation identity. We note that this simple model of two regional levels (state and nation) for self-identity can be expanded further.

- *Expertise Identity feature*:

  Users generally write their interests, expertise and affiliations in the description on Twitter user profiles. It is an example of self-representation of social identity (e.g., artist, researcher, etc.). Therefore, we first derive expertise classes by 2 steps: a) collect occupation categories and titles from trusted knowledge sources — Wikipedia and the US department of Labor Statistics reports, and b) classify the resulting occupation lexicon into ten broad classes, inspired by the domain classification on news websites and also from the super classes in the knowledge bases:

  {ACADEMICS, BUSINESS, POLITICS, TECHNOLOGY, BLOGGING, JOURNALISM, ART, SPORTS, MEDICAL, OTHERS }

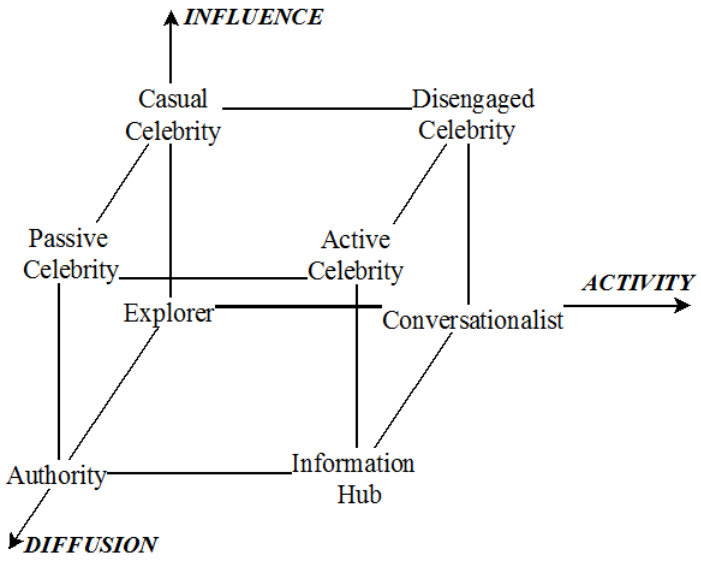


Figure 5.1: Online Identity based on three action measures (Influence, Diffusion, Activity)

For expertise identity assignment to a user, we first create N-grams from the description metadata in the profile by tokenizing on punctuations, and filter out those not containing any of the occupation lexicon terms. From the remaining N-gram set, each N-gram is associated with one of the ten classes, and its weight is determined by its position in the description text. Because users tend to place terms that are more socially identifying and important to them at the beginning, due to self-awareness of identity representation. Finally, the user is assigned to the highest-weighted identity class.

- *Online Identity feature*:

  Based on user actions on the platform (Twitter here), we use three metrics following the work of expertise presentation in [109] and influence and passivity in [114] that contribute to building a user's incentive-based identity (e.g.,

"Celebrity" on Twitter) of cyber-world — an online identity in contrast to real-world identities by capturing user activity, influence and diffusion strength. We model the activity metric by number of posts of the user, influence metric by number of mentions of the user, and diffusion strength by number of retweets of the user's posts during event time-frame. We compute scores on each of the three metric dimensions for all users and then consider the basic 50th percentile threshold to create two levels on each of the dimensions, yielding 8 user classes as shown in Figure 5.1. The computation on number of mentions, number of retweets, and number of posts here is different from the step of identifying social groups in the interaction-only network, because here node-centric features (a *local* viewpoint) are taken for identity measure, and not the connection-centric feature set, (a *global* viewpoint), which is the basis of clustering.

In contrast with regional and expertise identities, which are meaningful in the physical world, online identities exclusively define behavior in the cyber realm. From our knowledge, few attempts have been made to study the impact of both online and offline identities on group dynamics in online social networks.

In Table 5.5 we summarize the basic statistical information of each of the features related to social cohesion and identity. The upper bounds of entropy values for user features are included in brackets. From the assumptions of social cohesion and social identity theories, we hypothesize the following:

- A more structurally cohesive social group has less diverse discussion. Therefore, groups with higher density, transitivity, clustering coefficient, or lower shortest path length are expected to have lower discussion divergence.

- Groups whose members are similar in identities (those having lower entropy for identity features) are speculated to have low discussion divergence, as motivated by the social identity theory.

5.4 Analyses of Group Features and Discussion Divergence

In this section, we present the characteristics of structural and user features described in the previous section on our dataset and their correlation with group discussion divergence. It rationalizes the choice of features for the prediction task discussed in the next section.

5.4.1 User & Structural Feature Statistics

We identify several interesting trends in the results reported in Table 5.5. First, in general the entropy values[27] are higher for the Occupy Wall Street (OWS) and India Anti-Corruption (IAC) events, the two on-the-ground social activism events, possibly because the offline interactions heavily involved in those events are not captured by online social identity features. Such distinction is most pronounced when comparing online identity entropy values of those two events with respect to the other two events. The social groups in these two events tend to revolve around opinion leaders who often help direct and orchestrate the movement (such individuals likely will have high online identity values). Therefore social groups formed in those events generally have more diverse online identity composition, reflecting the presence of opinion leaders as well as followers in groups.

[27]Note, it is important to normalize these values against the maximum entropy possible for each case.

Another finding from Table 5.5 is that groups have great divergence in terms of their memberships from different regions. This may simply be a reflection of the times and the fact that online social networks are bringing people closer together and almost all events have had significant media attention.

Lastly, we point out that the average directed transitivity (global clustering coefficient) is at least 82% higher than that of the whole follower network (not shown in the table), and results based on the reciprocal and undirected definitions are similar, indicating that there is likely a community structure embedded in the social groups we have identified.

5.4.2 Correlation Between Features & Group Discussion Divergence

To investigate the relation between structural/user features and group discussion divergence, we first compute their statistical correlation. Particularly, we use bootstrap method (sampling with replacement) to construct the 95% confidence interval of correlation coefficients. In Table 5.6, we report a subgroup of features whose correlation with group discussion divergence is considered significant.

User features statistics: We note in Table 5.6 that user features (especially regional identity entropy and online identity entropy) have a moderate to high positive correlation with group discussion divergence, for the first three events. This finding agrees with our hypothesis that group discussion divergence rises when group members' identities become less distinct and thus identity entropy values rise. Correlation values for Occupy Wall Street are less significant, possibly due to some intrinsic characteristics of its conversation [32].

	Irene	Sandy	IAC	OWS
Directed Structural Features				
Density	$[-0.37, -0.06]$	$[-0.22, -0.16]$	$[-0.38, 0.07]$	$[-0.03, 0.05]$
Reciprocal Structural Features				
Density	$[-0.36, -0.06]$	$[-0.20, -0.15]$	$[-0.29, 0.06]$	$[-0.01, 0.07]$
Shortest Path	$[0.27, 0.52]$	$[0.10, 0.15]$	$[-0.14, 0.21]$	$[0.10, 0.16]$
Undirected Structural Features				
Density	$[-0.36, -0.05]$	$[-0.22, -0.17]$	$[-0.43, 0.10]$	$[-0.05, 0.04]$
Shortest Path	$[0.31, 0.56]$	$[0.16, 0.21]$	$[0.02, 0.37]$	$[0.09, 0.13]$
User Features				
Regional Entropy	$[0.23, 0.50]$	$[0.25, 0.30]$	$[0.07, 0.52]$	$[0.09, 0.14]$
Expertise Entropy	$[0.11, 0.51]$	$[0.45, 0.50]$	$[0.37, 0.66]$	$[0.01, 0.06]$
Online Entropy	$[0.45, 0.69]$	$[0.20, 0.25]$	$[0.11, 0.57]$	$[0.26, 0.31]$

Table 5.6: 95% confidence intervals of correlation coefficients between structure/user-based features and group discussion divergence

For social groups with a stronger regional concentration, in-group discussions tend to be more location-specific and consistent, leading to a smaller degree of member-wise discussion divergence, compared with groups whose members' locations are more dispersed. Similarly, the presence of users with similar expertise or interest domain in a social group tends to keep the scope of discussions more focused.

For the online identity feature, we note that it is reflective of user actions. Therefore, we speculate that for the sake of maintaining their incentive-based action identity via lesser change in their actions, users are likely to maintain a pattern of focused topic discussions in the groups.

Structural features statistics: For structural features, we find that patterns of correlation with group discussion divergence can be categorized into two types:

- Density features have a moderate correlation with group discussion divergence for Hurricane Irene and Hurricane Sandy, indicating that a better-connected social group tends to have a more cohesive discussion.

  We ask an event-type specific question, why is the correlation weaker for Occupy Wall Street and the India anti-corruption movements? As mentioned earlier, both of them are long-lasting events accompanied by an arguably more engaged offline component, whose information is not captured in cohesion features. Therefore, the density of online social groups is low (see Table 5.5), making it less indicative of sustainability for those two events.

- Average shortest path length (especially the undirected version) shows consistency in its positive correlation with group discussion divergence, which also agrees with our hypothesis. Compared with others structural features that reflect the tightness of a social group, average shortest path length shows clearer dispersion in values, making the result from its correlation analysis more meaningful.

- When comparing correlation strengths with content-divergence by reciprocal features and undirected features, we find that they are often comparable. In fact, a one-sided binomial test rejects the alternative hypothesis that "reciprocal features have stronger correlation with group discussion divergence than undirected features" with a p-value of 0.89. This finding is particularly interesting as the key premise of reciprocal structural features is *mutual* interpersonal attractions (social cohesion theory), an assumption that undirected structural features do not make. This leads to the question of whether mutual attraction

is still a necessary condition for *online* communities to form and last, and we believe it requires more research attention in the future.

5.4.3 Contrasting High & Low Divergent Groups

We performed a case study of 10 highest and lowest divergent groups in each event, where we analyzed their content to check if there is contrast between the content practices. Specifically, we compared the frequency of using hashtags, retweets (RT), mentions, URL links, and emoticons in the content of candidate group members. An interesting contrast was that the least divergent group members use practice of RT heavily, while the most divergent groups use hashtags heavily, indicating diverging nature of user classified topics. Therefore, we suspect content practices also play a role in predicting trend of divergence.

5.4.4 Effects of Event Characteristics

From Table 5.6 we note that transient events (Hurricane Irene and Hurricane Sandy) have stronger correlations with user features than with structural features. We conjecture it is due to the fact that groups in such volatile events form in an ad-hoc setting, where groups are less likely to have existing cohesively connected users, undermining the effects of structural features. Therefore, discussions can be highly dependent on the characteristics of participants of the group, their personal behavior and identities.

Furthermore, Figure 5.2 shows the general pattern of lower topical divergence in the *pre-event* phase, while increasing in the *during-event* phase and then again decreasing to lower value in the *post-event* phase. OWS is an outlier here likely due to high number of incidents even prior to the *pre* phase of the event in our dataset.

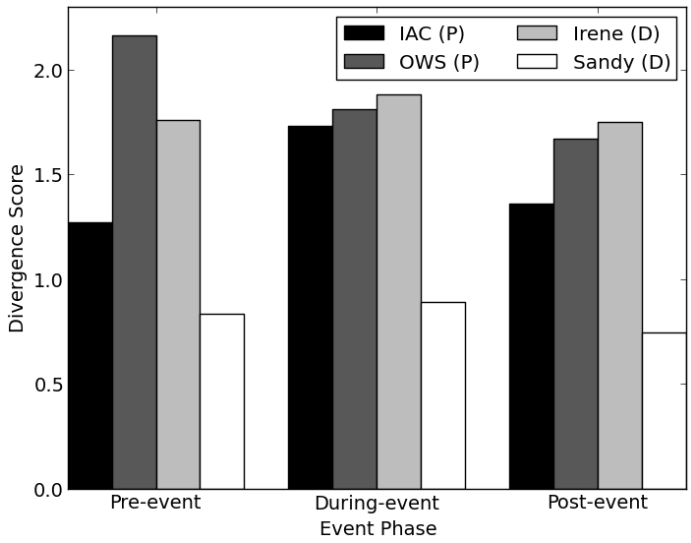


Figure 5.2: Average discussion divergence of groups in each of the phases for various events.

5.5 Predicting Trend of Group Discussion Divergence

In this section, we present the methodology and results for our main task: to predict the trend of social groups' discussion divergence. We plan to leverage observations from previous sections, including 1) statistical correlations between features and group discussion divergence, and 2) disparities of a subgroup of feature values between groups of high versus low group discussion divergence.

More precisely, our goal is to solve a learning problem where the label is whether the discussion divergence of a group of users will increase or decrease over time. Since each event is divided into three phases, there are two transitions: pre-event to during-event, during-event to post-event. Features selection are guided by the statistical analyses and case studies in previous sections.

5.5.1 Feature Sets and Learning Instances

We consider three main categories of features to use in the prediction problem. First, structural features focus on the cohesion and connectivity of each group's follower network. Second, user features emphasize the conformity of group users' offline and online identities. We have defined a family of those features in previous sections, and their significance varies. Lastly, content features capture the content practices of user-generated content. Based on the analyses in previous sections, we select different subsets of features from all of them, in order to reduce redundancy and improve prediction performance. The subsets are as follows:

- $Divergence$: Discussion divergence of the group at the current phase.
- $Structure_{sub}$: Directed density, reciprocal density, undirected density, reciprocal average shortest path length, undirected average shortest path length.
- $Structure_{all}$: All structural features described in the Feature Design section.
- $User_{all}$: Location entropy, occupation entropy, and online entropy.
- $Content_{sub}$: Average numbers of retweets and hashtags.
- $Content_{all}$: $Content_{sub}$ and average numbers of mentions, URLs and emoticons.

For each event, we identify pairs of social groups that are overlapping (Jaccard similarity[28] is above 0.5) before and after transition between two phases. There are 69 instances of group pairs meeting this criterion, and for 35 pairs their group discussion divergence values increase. We assign a label of *increase* or *decrease* to each group pair, depending on the change of its group discussion divergence value.

[28]The Jaccard similarity between two sets A and B is $\frac{|A \cap B|}{|A \cup B|}$.

5.5.2 Experiment Setup

For each pair of social groups of consideration, we use its features *before* the transition for the prediction task. Both SVM (RBF kernel with $\gamma = 0.5$) (*SVM*) and logistic regression (*logistic*) are used.

We also create another baseline method (referred to as *baseline*), which relies its classification on the current phase. In the preliminary analysis of content divergence above, it is observed that groups' content divergence in general increases from *pre-event* to *during-event*, and decreases from *during-event* to *post-event*. Therefore, *baseline* always predicts a group's discussion divergence to be *increase* if it is currently in the *pre-event* phase, or *decrease* if it belongs to the *during-event* phase.

5.5.3 Learning performance

To evaluate the performance of group discussion divergence prediction, we perform a five-fold cross validation on *SVM* and *logistic*. For *baseline*, we directly compute its F-1 score (0.54). Figures 5.3 and 5.4 show the performance of various feature sets and learning models, measured by area under the curve (AUC) and F-1 score.

It is demonstrated from Figures 5.3 and 5.4 that classification based on features described in previous sections are significantly more accurate than the baseline method (F-1 of SVM using structural and user features is 0.75, a 39% improvement). Furthermore, performance of classifiers varies according to the selection of features to use. While user features have shown high correlation with *static* group discussion divergence, our results suggest that structural features contribute most to accurately predicting the *dynamic* change of content divergence. Using structural features only, SVM achieves the best AUC (0.83) and F-1 score (0.76).

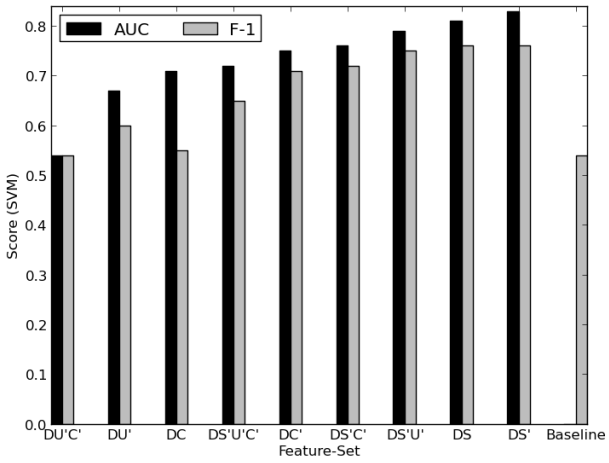


Figure 5.3: AUC and F-1 of prediction for SVM, organized by feature set and sorted by AUC. D=$Divergence$, U'=$User_{all}$, S=$Structure_{sub}$, S'=$Structure_{all}$, C=$Content_{sub}$, C'=$Content_{all}$.

5.6 Discussion

We performed qualitative study on the content of the overlapping groups by transition of phase (e.g., mid to post), and the divergence shift (e.g., decrease) using the Linguistic Inquiry Word Count (LIWC) software. We observe that groups who tend to diverge in their discussions write more of general reporting type content based on past incidents. While the groups with decreasing diverging behavior write more social and future action related content, likely due to users being organized to inform the fellow members about updates on the situation. For example, we found in the overlapping candidate groups of Hurricane Sandy event that a group with decreasing diverging behavior was highly focused on the updates of flight statuses of different airlines, first delays and cancellation, and later on the resuming parts. Such focused and active topic-specific groups will be valuable to engage with by the response coordinators.

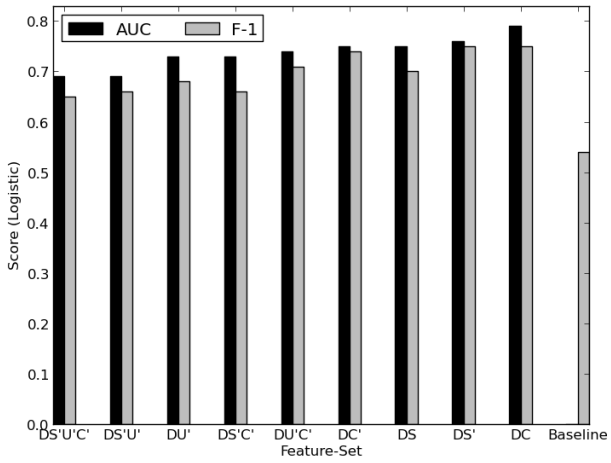


Figure 5.4: AUC and F-1 of prediction for logistic regression, organized by feature set and sorted by AUC. D=$Divergence$, U'=$User_{all}$, S=$Structure_{sub}$, S'=$Structure_{all}$, C=$Content_{sub}$, C'=$Content_{all}$.

To summarize our main contribution, we present an approach to understand factors that drive the shift of collective diverging behavior in the group discussion topics, and illustrate by a prediction model to show that these factors can help track the behavior of group discussion divergence. Its application can be in several domains, such as in brand management, or disaster response coordination. We can identify groups of audience that are active and concerned about specific issues. In the massive social media community after disasters, identifying reliable sources for engagement to coordinate about specific needs is a daunting task and the proposed approach also helps in identifying reliable sources of groups with specific information of needs. Another application of the proposed approach is for deciphering the self-organizing behavior of groups by learning the collective diverging trends.

Summarizing limitations about our study, we note that other group formation methods can be used and evaluated. We also limit ourselves to three phases in the prediction model experiment, namely *pre-*, *during-* and *post-event*, based on the real-world incidents on the event timeline. However, more phases may be considered for longer events, as they could also possess long-term impact. Extended evaluation needs to be performed across more events of diverse types in the future to validate the work's generalizability. We also did not consider other types of group behaviors due to first time analyzing event-oriented group discussion for collective behavior and thus, future studies can expand on that.

For our future work, we plan to extend our features of social identity and cohesion, including ethnic and religious social relationships, and structural properties from Twitter List subscriptions. We shall also validate models into other social networks, such as Facebook, Google+, LinkedIn, and the DBLP co-authorship network, to see if they show a similar social phenomena of group dynamics. Finally, we are also interested in detecting transition point of group discussion divergence over time, which may corresponds to the phase change from *storming* to *norming* in the group developmental sequence theory [133].

5.7 Conclusion

This study focuses on characterizing the online social group dynamics using content of group discussion in contrast to structural properties studied earlier, and proposes a measure of *group discussion divergence*. We include structural and user features guided by two socio-psychological theories of group bonding and attachment — social identity and social cohesion. Leveraging these features in addition to content

features, our classifiers accurately predict the future change of collectively diverging behavior in the group discussions. The classifiers achieve F-1 scores of up to 0.8, which is a 33% relative improvement from the baseline method. This study provides a framework to further research about collective behavior in online social groups.

Chapter 6: Patterns of Sentiment Shift in Online Conversations

In this chapter, we visit the third analytical task described in Section 1.3: studying how the sentiments of OSN users shift over time. Human sentiment in OSNs and its dynamics have attracted wide interest in recent years, since the increasing availability of data makes it more viable to perform empirical analyses and evaluations. While several methods have been proposed for measuring sentiment of users, the evolution of sentiment in a user network and patterns of shifts in user sentiments have not been afforded the same attention. Various applications such as marketing, advertising and recommendation systems are motivated by a need to influence users to adopt certain sentiments with particular products and services. It is therefore important to understand the factors that can cause shifts in opinions of users of a social network.

Earlier studies of face-to-face interactions have looked at the evolution of sentiment from the perspective of emotional contagion, which assumes that a person's sentiment can transmit to other individuals he is close to [63]. Experimental evidences have been revealed that emotional contagion also exists in online social networks [57, 73, 34, 74]. However, existing work is limited in multiple aspects. Prior work focuses on the general sentiment, and does not explore the difference in the pattern of emotion change across topics in various domains. Moreover, little is known about the impact on

emotion change by other factors, including the content's property, as well as characteristics of users and the social network itself.

In this chapter, we address the aforementioned limitations and in particular, determine what factors are essential for causing shifts in user sentiment in online social networks. By extracting more than 5 million tweets written by users that are exerting and receiving influence, we identify patterns in sentiment shift. We consider tweets over three different domains — movie, politics, and technology, enabling us to study the difference of sentiment shift behavior across different topics. Furthermore, we calculate the likelihood of sentiment shift when the influencer's tweet has certain content properties such as quotations, retweets or URLs, and compare it with the average shift probability. This helps identify factors that can significantly boost the probability of sentiment shift.

It is important to note that the focus of this work is the shift of sentiment by users and its driving factors, instead of performing sentiment analysis on the dataset itself. We believe insights into the process of sentiment shift will facilitate the design of effective strategy in applications such as advertising, reputation management and grassroot mobilization.

Our key findings include:

- The correlation between a user's influence and his ability to induce sentiment shift is very low.

- The appearance of negative content increases the likelihood of sentiment shift from positive to negative, whereas having positive content has no significant impact on sentiment shift from negative to positive.

- The more turns of tweets there are with no sentiment shift, the more likely it is for the influence receiver to drift away from the sentiment that the majority of users hold.

- In order to maximize the spread of a given sentiment, one should select seed users to produce tweets with the same sentiment, and also include quotation in the tweet.

We will first review existing work in Section 6.1, and then show results from analyzing sentiment shifts (Section 6.2) and maximizing the sentiment spread in networks (Section 6.3). In Section 6.4 we conclude this chapter by discussing the implication of our results as well as directions for future work.

6.1 Related Work

Sentiment has been an import aspect in the study of social computing, because of its critical role in the well-being of individual persons as well as the community as a whole. While sentiment is certainly a function of the individual's own psychological state, prior researches also provide well-grounded evidences that person-to-person interaction has non-trivial impact on one's sentiment. The phenomenon of emotional contagion, defined as "the tendency to ... converge emotionally" [63], has been found to exist in both short term [10] and long term [47] settings, and it further affects individual attitudes and group behaviors. Data sizes in those experiments are usually small, due to the complexity of organizing face-to-face studies as well as collecting data.

More recently, researchers have empirically examined the existence of emotional contagion in computer-mediated communication channels such as online chatting and

social networks, and it persists whether the communication itself is mediated [61, 58, 74] or spontaneous [57, 34, 73]. In those studies, sentiment is captured from texts (e.g. tweets or Facebook statuses), as an explicit indicator of sentiment is lacking.[29] Except for [57] which focuses on one single event, all work concerns with the general sentiment of users, regardless of the text's underlying content or topic. Moreover, existing studies do not investigate other potential factors in the spread of emotion, such as network structure, user characteristics, etc. Gruzd et al. [57] inspect the impact of a user's network position on the inclination of sharing positive or negative tweets, but not the sentiment shift from positive to negative, or vice versa.

Statistical modeling of the spread of sentiment has also been studied, and the main approach is to extend the family of epidemiology contagion models such as the Susceptible-Infected-Susceptible (SIS) model. Experiments are performed on synthetic networks [151] or face-to-face networks [66, 87], and their applicability to online social networks is yet to be verified.

Various automatic methods can be adopted to detect the sentiment in large number of documents in a scalable fashion. Performances vary, as algorithms designed for longer, more formal documents often ignore the peculiarity of texts in online social networks [103]. Lexical features, such as the sentiment associated with a known word/phrase/emoticon, are most frequently used, though it is challenging for any single method to prevail over different text sources [54].

[29]Facebook has since then allowed users to explicitly choose a *feeling* when posting a status update.

6.2 Methods and Experiments

In this section, we report major experiment results, following the intuitions listed in the beginning of this chapter. We first describe the process of training classifiers to automatically determine the sentiment subjectivity and polarity of social messages with regard to the topic of interest. Its accuracy is on par with start-of-the-art systems, as evaluated on established sentiment analysis benchmarks (Section 6.2.1). We then outline basic characteristics of the Twitter datasets we are using (6.2.2) as well as the composition of sentiment transitions (6.2.3), before proceeding to analyze users' sentiment shifts with respect to user characteristics (6.2.4), content (6.2.5), and the length of interactions between influence sender and receiver (6.2.6).

6.2.1 Determining Subjectivity Sentiment and Polarity

The study of sentiment shift is built upon the ability of detecting the subjectivity and sentiment polarity from user-generated contents. To scale up beyond the capacity of human annotation, there is the need to construct an automatic classifier. Prior studies of emotional contagion in online social networks often rely on simplistic approaches such as counting words that are indicative of polarized sentiments [34, 74]. Though easy to compute, it is unclear how such methods can effectively distinguish between objective and subjective documents in a principled manner. Furthermore, lexicons such as LIWC [107] cater poorly to the writing style of online social network content, such as word variation ("tomorrow" written as "tmr"), and letter repetition (for example, "greaaat"). Therefore using lexicons alone often leads to poor coverage [54].

In our work, we purpose the usage of a two-level classifier to perform the task. The classifier first determines if a document (in our case, a tweet) is objective or subjective, and then classifies the tweet's sentiment polarity (positive or negative) if it is subjective. Therefore, each tweet is labeled either *objective*, *positive*, or *negative*. We use the training dataset from SemEval-2013, Task 2B [99][30] to train two logistic regression models. The first (subjectivity) model treats objective tweets as one class, and all others as another. The second (polarity) model discards objective tweets, and separates positive and negative tweets into two classes. When evaluated on the SemEval testing dataset, the performance of our model is on par with the best-performing system in the task [94][31].

Tweet features used in both models include:

- Unigrams and bigrams.
- World repetition: Whether a letter appears in a word consecutively for more than twice. Repetitions are replaced with two letters.
- If all letters in a word are capital letters.
- If there is emoticon or interjection in the tweet.
- If there are two or more exclamation marks in the tweet.
- If there is first, second, or third personal pronoun in the tweet.
- Negated version of words from the negation to the end of corresponding clause.

[30] `http://www.cs.york.ac.uk/semeval-2013/task2/`

[31] Unfortunately the system's implementation is not publicly available.

- Lexicons: Harvard General Inquirer [125], MPQA [143], SentiWordNet [42], VADER [67].

- Word clustering to detect variation: Brown clusters of words with prefixes of 4, 8, and 12 bits [113].

6.2.2 Dataset Description

We collect tweets about topics in multiple domains, including movie[32], politics and technology. For each topic, a manually selected set of keywords and hashtags are used to match all tweets from Twitter's data stream. Those datasets differ in duration and volume, and therefore provide a platform to study which characteristics about sentiment shift are common to all topics, and which are domain-specific. Table 6.1 outlines basic information about the datasets.

Name [Domain]	Duration	# Tweets	# Users
Avatar [Movie]	12/2009 – 03/2010	3584956	1391624
New Moon [Movie]	11/2009 – 03/2010	1191419	539145
Benghazi Select Committee (Benghazi) [Politics]	04/2014 – 06/2014	409722	66740
Affordable Care Act (ACA) [Politics]	04/2014 – 06/2014	246164	57843
IT Products (Products) [Technology]	07/2013 – 08/2013	219787	90408

Table 6.1: Basic dataset statistics.

Before proceeding to the analyses, we define several terms here. A *turn* consists of a series of chronologically-ordered tweets written by two users: first by an influence *receiver* (stage 1), then by an influence *sender* (stage 2), and finally by the same receiver (stage 3). In each stage of the turn, one or more tweets can be composed.

[32]While we have tracked tweets about 54 different movies in total, we report results on the two largest datasets here. Observations on other movie datasets are similar.

Rather than enumerating all possible pairs of users, we require that the receiver needs to have retweeted from the sender at least once in the dataset, as an indication of the sender's actual influence on the subject matter. It also implies that a receiver in one turn can be the sender in another turn.

An exemplar turn is shown below. The influence sender is user "rosenfie", and the influence receiver is user "hmtangx".

hmtangx: @tsangtammy one thing new moon was lacking: some STRONG bella/edward passion i thought... didn't really feel it this time :S
rosenfie: #Imthankfulfor the fact that New Moon is still trending. Keep it going guys! Even on a holiday! Awesome!
rosenfie: New Moon is great
rosenfie: @billy_burke Happy Thanksgiving!! Loved you in New Moon - will continue in Eclipse and BD I'm sure! - Tracy
hmtangx: New Moon is still trending after a weeeeek!!! :D heheeh :)
hmtangx: NEW MOON <3 omg loveee, can't wait for eclipse!!

Table 6.2: Sample turn extracted from the *New Moon* dataset.

To determine the presence of sentiment shift by an influence receiver, we run the sentiment classifier (Section 6.2.1) on his tweets in stages 1 and 3. At each stage, the most frequently-occurring sentiment label is taken as the stage's sentiment label. If either stage is objective, the turn will be removed from further analyses. The combination of sentiment labels in both stages is called a sentiment *transition*

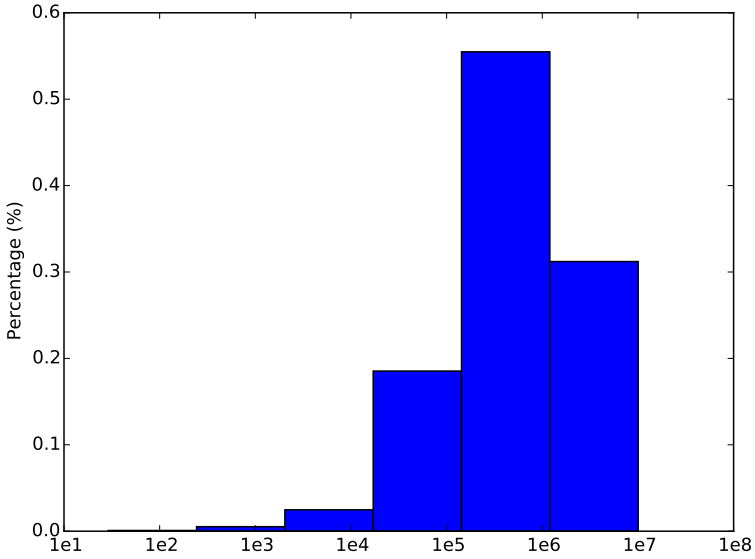


Figure 6.1: Histogram of turn durations (seconds) for *New Moon*, in log scale.

hereafter, and a transition contains a sentiment *shift* when the labels in two stages are different.

We further calculate the duration of turns, and plot the distributions of turn duration for two representative topics: *New Moon* (Figure 6.1) and *Benghazi* (Figure 6.2). The duration of a turn is the pairwise average of time differences between each tweet in stage 1 and each tweet in sage 3. Most turns last more than 10^5 seconds (1.16 days), and some last as long as months.

6.2.3 Sentiment Composition and Sentiment Shifts

An overview of the sentiment of tweets is provided in Table 6.3. While less than 5% of tweets are subjective for the technological topic (Products), the proportion is much higher for political issues (15%) and movies (35%), signifying their stirring

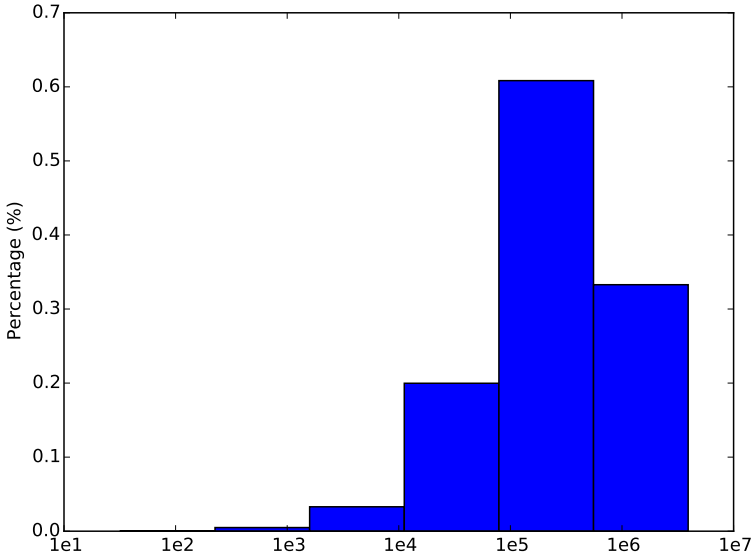


Figure 6.2: Histogram of turn durations (seconds) for *Benghazi*, in log scale.

nature. Moreover, controversial political issues induce more polarizing sentiments from users, whereas the majority of tweets about movies are positive.

Name	# Objective Tweets	# Positive Tweets	# Negative Tweets
Avatar	2301897	1122305	160754
New Moon	721398	397981	72040
Benghazi	349408	22820	37494
ACA	204722	21291	20151
Products	209547	7615	2625

Table 6.3: Distribution of sentiment labels.

Grouping turns according to their sentiment transitions (Table 6.4) reveals similar trends. Topics whose majority of tweets are positive see a higher likelihood of transiting into positive, and vice versa. While for all topics more than half of all turns do not involve sentiment shifts, there is still a considerable proportion of sentiment shifts present in user tweets. Therefore, investigating the relationship between them and properties of the social conversations (user, network and content) is crucial in achieving a fuller understanding of sentiment shift.

Name	# Turns	Pos. → Neg.	Neg. → Pos.	Pos. → Pos.	Neg. → Neg.
Avatar	179556	12.61%	12.80%	70.46%	4.13%
New Moon	66030	12.34%	11.42%	73.25%	2.99%
Benghazi	24005	19.75%	19.66%	9.45%	51.14%
ACA	14644	24.67%	22.75%	19.65 %	32.93%
Products	2173	10.58%	12.20%	70.23%	6.99%

Table 6.4: Distribution of influence receivers' sentiment transition in a turn.

One intriguing observation is the almost-equal percentages of positive-to-negative shifts and negative-to-positive shifts for each topic. To ensure this is not an artifact from the noise in sentiment classification (e.g. a false negative in a stream of positive tweets will produce a positive-to-negative shift, immediately followed by a negative-to-positive shift), we calculate the proportion of such sentiment "back-flips" among all sentiment shifts. The values range from 25.46% (*Avatar*) to 36.81% (*Benghazi*), accounting for only one-third of all sentiment shifts. This signifies the lasting effort of most sentiment shifts that occurred.

One parallel explanation to the observation stems in the conjecture that people may temporarily accommodate to others' sentiments and opinions, in order to smooth

inter-personal interactions. Should that have manifested in OSNs, a user would have changed his sentiment, and later reverted back to his original stance. It warrants further investigation to verify to which extent this explanation holds.

6.2.4 Sentiment Shift and User Influence

One may wonder what the link is between sentiment shift and user influence, as the adoption of a certain sentiment can be viewed as being influenced to do so. Intuitively, the more influential an (influence) sender is, the more likely he is to cause a receiver to adopt the sentiment of his message(s).

To verify this hypothesis, we compute the influence metric of each sender, and calculate the correlation coefficient between it and the probability of his influence receivers showing sentiment shift. When the sender's tweet is positive, the receiver's sentiment is anticipated to shift from negative to positive, and vice versa. We consider two different influence metrics: out-degree and PageRank value, since they are shown to be different proxies of a user's true influence [19].

Results from Table 6.5 (columns 2–5), however, suggests that sentiment shift has no significant correlation with a sender's own influence. Similarly, we find trivial correlation (columns 6–9) between a receiver's propensity of changing sentiment and his in-degree/PageRank values. The likelihood of an influence receiver also has low correlation with the number of senders he is associated with (numbers not shown here).

Those results provide little support for the relation between sentiment shift and user/network characteristics. As a result, it is sensible to study the role of content

in sentiment shift, and examine if certain properties in the senders' tweets are more likely to induce sentiment shift. We address this exact question below.

	Influence Sender				Influence Receiver			
	Pos. → Neg.		Neg. → Pos.		Pos. → Neg.		Neg. → Pos.	
	Out-deg	PageRank	Out-deg	PageRank	In-deg	PageRank	In-deg	PageRank
Avatar	0.02	0.00	0.01	0.00	-0.03	-0.02	-0.01	-0.01
New Moon	-0.02	0.00	0.00	-0.00	-0.02	0.02	0.00	0.00
Benghazi	-0.02	0.00	0.01	-0.00	0.01	-0.00	0.00	-0.00
ACA	-0.05	-0.00	0.00	0.00	-0.00	-0.00	0.07	0.05
Products	-0.06	0.00	-0.08	-0.00	-0.03	-0.04	0.01	0.01

Table 6.5: Correlation coefficient between user influence and sentiment shift probability. Columns 2–3 are for a sender changing the sentiment of his receivers from positive to negative, given the sender's tweet is negative. Columns 4–5 are for a sender changing the sentiment of his receivers from negative to positive, given the sender's tweet is positive. Columns 6–7 are for a receiver changing the sentiment from positive to negative, given the sender's tweet is negative. Columns 8–9 are for a receiver changing the sentiment from negative to positive, given the sender's tweet is positive.

6.2.5 Effect of Content on Sentiment Shift

As suggested in previous subsections, it is necessary to inspect content properties of influence senders' tweets, and their relationship with the sentiment shift of influence receivers. More specifically, we are asking the question *whether the appearance of certain properties in senders' tweets makes the likelihood of sentiment shift higher than random?*

This can be viewed as comparing the prior probability of sentiment shift (Table 6.4) and the corresponding posterior probability, conditioned on the occurrence

of given content properties. If the conditional probability is higher, then sentiment shift is more likely to occur than by chance.

In this work, we study the following types of content property:

- **Sentiment:** The sentiment label of a sender's tweet (objective, positive, or negative).
- **Retweet (RT):** Whether the sender's tweet is a retweet.
- **Quotation:** Whether the sender's tweet contains a quotation.
- **URL:** Whether the sender's tweet has any hyperlink that points to external webpages.

Figures 6.3 and 6.4 show the comparison of sentiment shift probabilities in each topic, grouped by the content property of influence senders' tweets. Also included are the prior probabilities of sentiment shift.

In Figure 6.3, it can be seen that for all topics the probability of sentiment shift from positive to negative is higher than average when the sender's content contains negative sentiment. One-sided binomial test also shows that the difference is significant for all topics ($\alpha = 0.1$). The presence of quotations also significantly raises the likelihood of changing sentiment to negative, for all topics except ACA.

On the other hand, when inspecting sentiment shifts from negative to positive (Figure 6.4), there is not always significant increase even when the influence sender's tweet is positive. Therefore, although emotional contagion exists for both positive and negative sentiments [63, 74], their powers are not equal. The finding of negative sentiment being more "contagious" concurs with prior work's observation that bad feelings are easier to form and more difficult to shift [11].

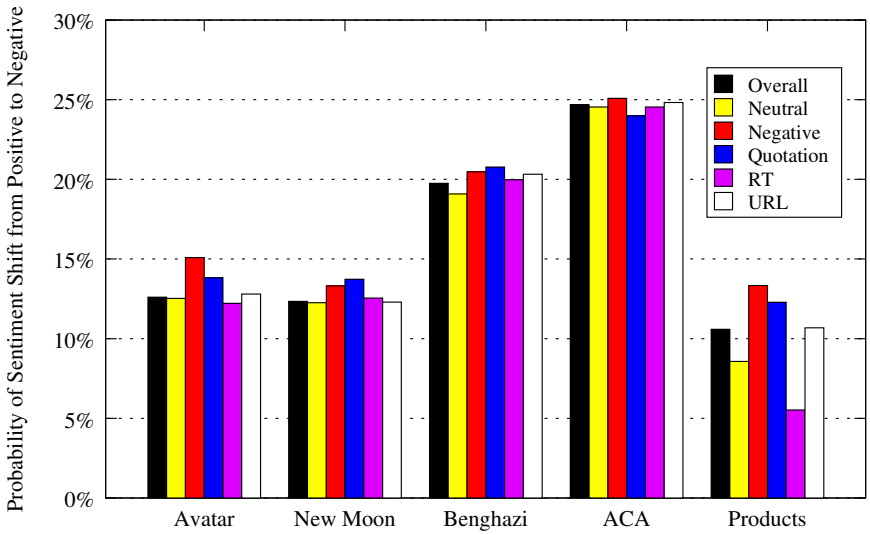


Figure 6.3: Probability of sentiment shift from positive to negative, over all turns ("Overall") and turns where tweets of the influence sender have specific content properties.

Furthermore, we find that objective tweets do not increase the chance of triggering sentiment shift, regardless of the direction of shift. Therefore, in order to shift others' sentiment, an influence sender will be more effective if his own tweets contain the target sentiment.

6.2.6 Sentiment Shift as A Multi-Turn Process

Another element of interest is the effect of multiple turns on sentiment shift, since the conversation between an influence receiver and a sender can involve more than one turn. Therefore, it is possible that multiple non-changing sentiment transitions (one for each turn) exist before a sentiment shift occurs.

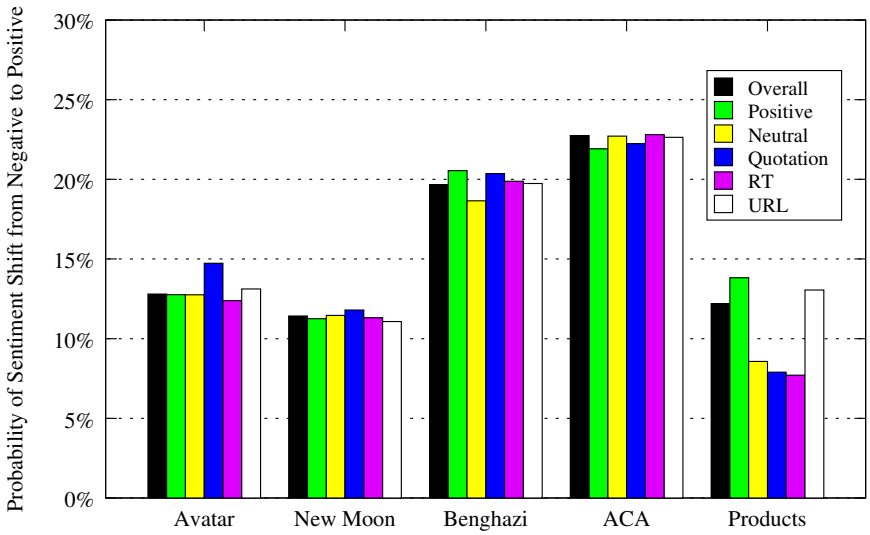


Figure 6.4: Probability of sentiment shift from negative to positive, over all turns ("Overall") and turns where tweets of the influence sender have specific content properties.

Two competing prepositions can be leveraged to predict the variation of sentiment shift probability as more turns are present. On one hand, an increasing amount of exposure to a sentiment is expected to increase the propensity of aligning one's own sentiment with it. On the other hand, a series of unsuccessful attempts to shift sentiment are likely to further discount the receiver's inclination of doing so.

In order to validate the two hypotheses empirically, we plot the probability of sentiment shift against the number of non-changing turns until the shift occurs, For example, if there are 3 turns between an influence receiver and a sender, and the three turns' transitions are positive $\rightarrow$ positive, positive $\rightarrow$ positive, and positive $\rightarrow$

negative, respectively, then the number of turns is 3. If the transition in the second turn is positive → negative, then the number of turns is 2.

Results from two representative topics — *New Moon* (Figure 6.5) and *Benghazi* (Figure 6.6) — are shown below. The results of *Avatar* and *Products* show similar trends as *New Moon*, while *ACA* is similar to *Benghazi*.

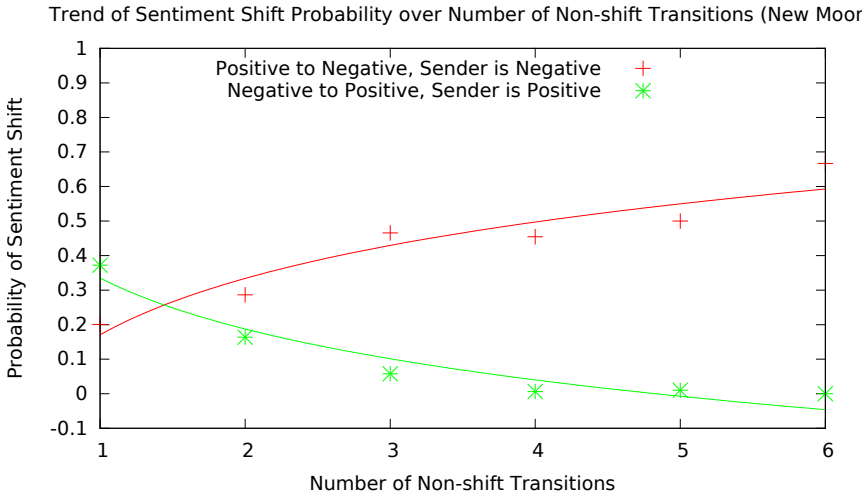


Figure 6.5: Probability of sentiment shift on the topic *New Moon*, conditioned on the number of turns until shift happens. Logarithmically-fitted trend lines are also plotted.

The plots clearly suggest that the way sentiment shift probability varies over time is not always the same. As the number of turns increases, it is increasingly difficult for the sentiment on *New Moon* to shift from negative to positive (the logarithmically-fitted trend line has an R^2 value of 0.93), for example. However, it becomes more likely for the sentiment to become negative ($R^2 = 0.90$). On the other hand, directions

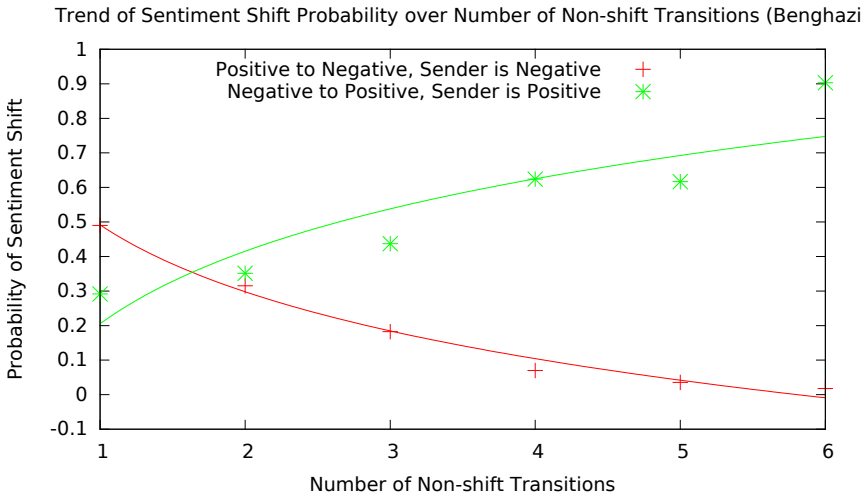


Figure 6.6: Probability of sentiment shift on the topic *Benghazi*, conditioned on the number of turns until shift happens. Logarithmically-fitted trend lines are also plotted.

of trends differ among topics. In *Benghazi*, for instance, the probability of changing to negative decreases over the number of turns.

The dichotomy of trends cannot be fully explained by either preposition, but rather by the observation that it is easier to sway sentiment *away* from the dominant sentiment on the topic, and harder to shift the sentiment to it (recall that the dominant sentiments in *New Moon* and *Benghazi* are positive and negative, respectively). We believe this deserves future research efforts, as it may be linked to the socio-psychological phenomenon of *minority influence* [144].

We note that such non-conformity in sentiment shifts does not imply that the dominant sentiment will become the minority after a sufficiently large number of turns. When dividing our datasets into 5 consecutive windows, for instance, we find that the dominant sentiment remains the same. As the number of turns increases,

fewer people will keep the minority sentiment and continue exerting their influence. Therefore the amount of people being converted is low.

6.3 Maximizing Sentiment Spread in a Network

Using the insights learned from patterns of sentiment shift in social networks, we can devise more effective strategies of steering the sentiment on particular topics. Such techniques can be applied to applications such as brand management, advertising and political mobilizations. One practical question is: Assume there exist a user network, a target sentiment, the probability of sentiment shift, and a fixed budget of "seed" turns where influence senders attempt to affect receivers' sentiments. What is the largest sentiment spread that can be created, and which properties should the seed contents have?

This *spread maximization* problem can be viewed as an extension to the influence maximization problem with an independent cascade model [72], where seed users act as influence senders. Initially all users have the sentiment opposite to the target, and seed users start by changing themselves to the target sentiment. At each round, each newly-changed user has a success rate of switching the sentiment of his non-changing friends. The distinction of the spread maximization problem from influence maximization is that the success rate of a user is no longer static, as it varies according to the number of turns he has made (Section 6.2.6).

To handle this challenge, we transform the spread maximization problem to a series of standard influence maximization problems. Let the budget on number of turns be K. We start by setting the weights of all edges to the probability of sentiment shift conditioned on 1 turn. The influence maximization algorithm is then run to calculate

the expected coverage by K seeds. The first seed user, who has the largest marginal contribution to sentiment spread, is selected, and the weights of all of his out-edges are updated to the probability of sentiment shift conditioned on 2 turns. On the updated network, the influence maximization algorithm is run again, but picking only $K-1$ seeds and computing the expected coverage. This process continues until the number of required seeds becomes zero.[33] The maximum of sentiment coverage under all K subproblems becomes the greedy solution to the sentiment spread maximization problem.

We solve the spread maximization problem on networks built from users' retweet activities, where each retweet forms a directed edge from the original author to the user. To solve the influence maximization problem, we use PMIA [24], which utilizes various heuristics to obtain a high-quality approximate to the otherwise NP-hard problem.

Figure 6.7 plots the expected coverage of sentiment spread (the number of users that have the goal sentiment) on *New Moon*, with $K = 25$ seed turns. For both target sentiments, we solve the spread maximization problem with senders' tweets having different properties: positive, positive+quotation, negative, negative+quotation, and quotation only. The property of having quotation is included because for most topics the probability of sentiment shift becomes higher with it presented (Section 6.2.5).

As expected, positive tweets from senders lead to greater coverage of positive sentiment than negative tweets, and vice versa. When tweets have quotations in addition, the expected sentiment spread often increases as well.

[33]In our experiment, we limit a user to be selected by at most 5 times, given that the sentiment shift probability varies little after 5 turns.

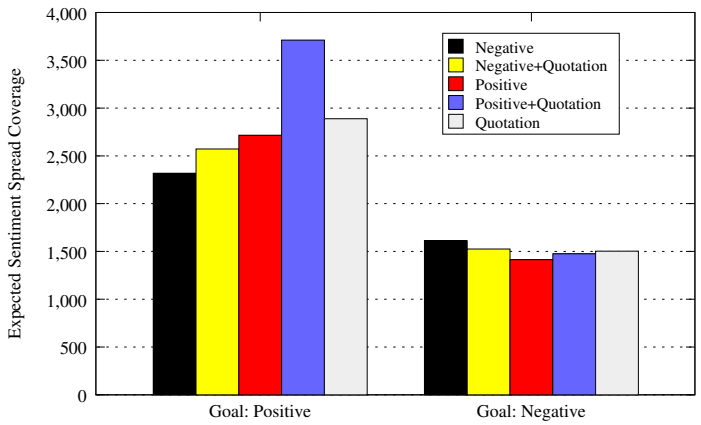


Figure 6.7: The expected coverage of sentiment spread on *New Moon*, with 25 seed turns.

6.4 Conclusion

To date, there have been limited number of empirical studies on sentiment shift and sentiment spread in online social networks [57, 34, 74]. In this chapter, we have conducted an analysis of patterns of sentiment shift for more than 2 million twitter users and 5 million tweets on various topics. We believe the results from this work will motivate more thorough multi-disciplinary research efforts in the future.

Our experimental results suggest that content properties of user posts have noticeable impact on the inclination of people to adjust their sentiment regarding specific topics. For example, tweets containing negative content significantly boost the transition of sentiment from positive to negative, and so do tweets that include quotations. However, there is only weak evidence for the opposite, as positive tweets do not necessarily make a user more likely to move away from negative sentiment. The asymmetry of influence powers can be the evidence for the "bad is stronger than good" phenomenon [11] in online social networks.

The finding that people are increasingly likely to drift from the dominant sentiment is another topic worthy of future investigation. First, is it possible for the minority sentiment to become dominant, as opposed to our dataset, given certain initial sentiment distribution? Second, can we promptly detect the dominant sentiment at the dataset's early stage, and thus infer the trend of sentiment shift over time? Answers to these questions can help design more informed strategies for problems including sentiment spread maximization.

Moreover, we believe that the roles of network and users still warrant more studies, despite the low correlation between those characteristics and sentiment shift probabilities. One limitation of our work is that since we were examining particular topics, we did not have the complete graph information of the following relationship when calculating users' influence metrics. This will be addressed in future work. It will also be interesting to conduct similar experiments on other network datasets to examine the robustness of our findings.

Finally, we shall investigate the link between sentiment shift and the behavior of information sharing. Many studies [112] have pointed out the motivation behind sharing one's emotion with others. What remains intriguing is the effect of sentiment shift on information cascade in OSNs. Are people more like to share information when there is sentiment shift, and from which polarity to which? Does it have different effects on emotional content versus non-emotional content? These are all exciting questions to social computing and communication researchers.

One limitation of our work is the assumption that turns represent true conversations, even though tweets are broadcast to a larger number of audience instead of directed at individuals. The long average duration of turns also raises the possibility

that it is offline engagement or the inherent trend of topic itself that produces sentiment shifts, instead of the influence of online friends. Future researchers ought to control these factors, in order to reach more confident conclusions.

Chapter 7: Conclusions and Future Work

We conclude the dissertation in this chapter, by summarizing our contributions to dynamic OSN analytics (Section 7.1), pointing out the limitations in present work (Section 7.2) and possible directions for future work (Section 7.3). Dynamic OSN analytics concerns with understanding the development and interplay of its key components: network structure, user-generated content, and user characteristics. The central statement of this dissertation is that *a holistic approach that incorporates the information on network, content, and users will leading to more robust graph algorithms, as well as a more comprehensive understanding of content and network dynamics in OSNs.* We demonstrate the benefit of this approach through the development of a series of analytical tasks: (1) community detection, (2) structural role detection, (3) user engagement prediction, (4) online social group discussion divergence prediction, and (5) OSN user sentiment shift analysis. In each study, we combine the knowledge from multiple aspects, and show the advantage of this holistic framework over methods that only focus on one single aspect. Our models are well-tested on large-scale datasets collected from real social networks, including Facebook, Google Plus, and Twitter, and always yield better empirical performance than other state-of-the-art approaches.

Below we summarize the key innovations and findings obtained from each of our contributions.

7.1 Summary of Key Contributions

- **Network Simplification and Community Detection Using Structural and Content Information** *Chapter 2*

  We design an extremely simple by efficient algorithm for community identification in large-scale graphs by fusing content and link similarity. Our algorithm, CODICIL, selectively retains edges of high relevancy within local neighborhoods from the fused graph, and subsequently clusters this backbone graph with any content-agnostic graph clustering algorithm, such as METIS or Markov Clustering. Our experiments demonstrate that CODICIL outperforms state-of-the-art methods in clustering quality while running orders of magnitude faster for moderately-sized datasets. Built on the simplification procedure, we are able to handle large graphs with millions of nodes and hundreds of millions of edges, which would otherwise be challenging to analyze. While simplification can be directly applied to the original graph alone with a small loss of clustering quality, it is particularly potent when combined with content edges, delivering superior clustering quality with excellent runtime performance. Additionally, one distinct benefit provide by CODICIL over other network simplification algorithms is that it can recover missing links (false negatives) by factoring in content similarity.

- **Joint Discovery of Communities and Structural Roles in Network** *Chapter 3*

We propose a principled algorithm to mine communities and structural roles from networks simultaneously. Communities and structural roles provide two complementary angles in understanding network topology, and utilizing information from one aspect can benefit the inference of another. Our algorithm, RC-Joint operates in an alternately manner, and improve community and role assignments iteratively until convergence. Furthermore, the design of RC-Joint enables the identification of overlapping community and role assignments, which occur more frequently in real-life scenarios than disjoint assignments. We compare the outputs by RC-Joint and other state-of-the-art single-tasked mining algorithms run on real-world and synthetic networks, and find that RC-Joint indeed produce results that match better with gold standard information. The node-centric computation paradigm lends the algorithm itself with easy parallelism and significant speedup with OpenMP or other similar framework, making it easily scalable for large networks. Additionally, we find that by running RC-Joint on a sparse version of the network first and using the output to initialize the second run, it actually yields faster convergence and better results.

- **Prediction of User Engagement in Online Discussions** *Chapter 4*

  We systematically investigate factors that may affect the engagement of OSN users in online discussions related with real-world events. We build an effective prediction model to estimate the engagement decision as well as the volume of event-relevant tweets as a result of user engagements. Our model utilizes features extracted from network structure (e.g. degree, network component size), content (e.g. usage of sentiment words, emoticons), user behavior (e.g. frequency of retweet, influence measure), and historical information. Evaluations

on a large number of Twitter event-oriented communities demonstrate that our model can produce predictions of higher quality, compared with other more parsimonious models. Moreover, we find there exist correlations between event types and feature values, which grant future study of user engagement in an event-specific setting.

- **Analyses of Discussion Divergence on Online Social Groups** *Chapter 5*

  We focus on characterizing the online social group dynamics using content of group discussion in contrast to structural properties that have been studied earlier, and proposes a formal definition of *group discussion divergence* based on Jensen-Shannon divergence. We study the link between online social groups' discussion divergence and their structural and user features, which are inspired by the theories of social identity and social cohesion, respectively. Although both theories have their roots in the socio-psychology literature targeted at face-to-face social interactions, we find that strong correlations still exist between a subset of features and the discussion divergence of online groups. Leveraging these features in addition to content features, our classifiers can accurately predict the future change of collectively diverging behavior in the group discussions. The classifiers achieve F-1 scores of up to 0.8, a significant improvement (33%) from the baseline method. Furthermore, we find statistical evidence that features derived from a weaker assumption of uni-directional interpersonal attraction have obtained equivalent performance in predicting discussion divergence. This result suggests that mutual attraction, as conjectured in the original proposal of social cohesion theory, may not be a necessary condition for structural cohesion in Twitter.

- **Patterns of Sentiment Shift in Online Conversations** *Chapter 6*

  We have conducted an analysis of patterns of sentiment shift for more than 2 million twitter users and 5 million tweets on topics including movies, politics, and technology. Our experimental results suggest that content properties of user posts have noticeable impact on the inclination of people to adjust their sentiment regarding specific topics. For example, tweets containing negative content significantly boost the likelihood of shifting sentiment from positive to negative, and so do tweets that include quotations. However, the reverse does not hold in our experiment. The asymmetry of influence powers provide new evidence supporting the presence of "bad is stronger than good" phenomenon in OSNs. We also find that people are increasingly likely to drift from the dominant sentiment, and we believe it deserves future investigation. Finally, we find very low correlation between sentiment shift probability and user's degree as well as PageRank value, and this aligns with the observation of "million follower fallacy" on Twitter [19].

7.2 Limitations in Present Work

Reviewing the research methodology and conclusion in this dissertation, we find that despite the contributions to OSN analytics in multiple fronts, there are still limitations in our studies. Briefly speaking, the drawbacks come from three aspects: modeling approach, theoretical connection, and algorithm complexity.

Data Quality:

The value of OSN analytics algorithms often hinges on the underlying data's quality. For based using real social network site data, we have strived for the highest

data quality as possible by expanding seed list in the crawling process, as well as spam filtering and other data cleansing operations. However, public access to live streams of services such as Twitter are often subject to volume constraint, making the collected data only a sample of the full traffic. The mechanism behind such sampling is not disclosed to the public, leading to the question of whether the sample is representative (i.e. unbiased) of the complete dataset. Until we are able to obtain a concrete answer to this concern, it is advised to always treat the results from relevant studies with care. A more constructive approach is to apply a notion similar to confidence interval on the results, and to use principled statistical methods to quantify the uncertainty as a result of the upstream data sampling. If multiple samples of the same traffic are available, it is also feasible to adopt bagging-like methods to produce robust analytical results.

Another limitation regarding data quality is the lack of offline activity information. While users have spent significant amount of time online, they also participate in offline activities and engage with each other. In an ideal world, information of both online and offline activities is analyzed, user identities are matched, and influence in decision making is attributed to both channels properly. For our large-scale studies, however, it is too costly to collect offline data, creating challenges in accounting for any effect of offline data.

Modeling Approach:

For most analytical tasks discussed in this dissertation, we have defined, engineered, and used quantitative features to construct both supervised and unsupervised machine learning models. Although some features can relate to qualitative description in existing theories (such as those discussed in Chapter 5), many others are created

in a more *ad hoc* fashion. One disadvantage of such an approach is that correlations may exist among subsets of features, both affecting model complexity and prediction quality. While there are various methods (such as Principle Component Analysis and Singular Value Decomposition) to alleviate this issue, and some of them have been leveraged (e.g. Chapter 4), they often apply complex transformation to the original feature space, making the resultant features less interpretable and intuitive to end analysts. Also, the correlation analyses performed in the dissertation are only able to gauge linear relationship between a feature and the response variable, and it falls short in capturing any nonlinear relationship.

Moreover, the output from predictive tasks in this work is always a single attribute (i.e. a discrete label or a numerical value). However, OSN activities in real life may also have other types of representation, such as time series or graphs. The ability of producing such structured output is very valuable, as it encodes more information in the output. Achieve this goal, however, will require more sophisticated model specification and the corresponding algorithm to solve it.

Theoretical Connection:

In this dissertation, we have described our preliminary efforts in bridging existing socio-psychological theories and dynamic OSN analytics (Chapters 5 and 6). Given the plethora of relevant theories, what we have studied is just the tip of the iceberg, and they are often more nuanced than how they are modeled in the current approaches. To illustrate, when analyzing the probability of sentiment shift and factors behind it, we have limited ourselves to one influencer at a time. In the real world, however, the behavior of sentiment shift by an individual is more likely to be the result of the collective influence from his or her social circle.

Furthermore, the scope of the data we have analyzed could have been further broadened. This is pertinent to the generalizability of our research findings. While we have always experimented with datasets from various domains and events in different categories, by no means have they covered every single possible domain. In terms of data sources, they should not be limited to Twitter and other smaller OSNs either, although data availability may soon become an practical issue. As social media companies grow larger, they share the unfortunately tendency of tightening control on site data and usage, making it more difficult for OSN researchers to deploy their studies on real-world datasets.

Algorithm Complexity:

Finally is the challenge of algorithm complexity, as it is often the sole factor of stopping us from developing more realistic models to capture the nuance in OSN activities. As already mentioned in the previous sections, there is often a trade-off between the complexity and result quality of an analytical model. Although adding an interaction item in the model may seem intuitive and trivial during the designing phase, only in a later stage may we find that it renders an otherwise-polynomial inference algorithm exponential. Without significant advances in the algorithm design, we may only rely on more simplistic models, if we still want to obtain results in a reasonable amount of time.

7.3 Future Work

In this last section, we discuss some directions for future work. With a growing interest in OSN analytics, those fundamental directions will surely benefit its further development.

7.3.1 Feature Learning and Structured Prediction in OSN Analytics

One promising solution to feature design is to automatically *learn* features via recursively aggregating features. The aggregation process keeps creating new features until they become sufficiently correlated with each other, and it has been successfully used in deriving network features for each node in the network [65]. In that work, the scope of aggregation is over the expanded neighbor of a node, and it may require modification when applying to other types of features, such as content and user characteristics.

Another thread to investigate is to use structured learning in predictive tasks. In the present work, all predictive models are designed to output one label for each input data record (i.e. a feature vector containing features regarding the user or group of interest). The interaction among users/groups is only encoded as aggregated features, and rich information collapsed and lost in this feature representation. Structured learning has been proposed to address this issue, and has proven itself useful in various machine learning algorithm [8], such as natural language processing [140] and visual recognition [44]. We plan to explore the possibility of applying structured learning by expressly modeling the interactions among different units as first-class objects.

In terms of analyzing the association between features and the output value, we can adopt the notion of *maximal correlation*, which is a generalized version of correlation coefficient. The maximal correlation between two populations can be calculated via a modified alternating conditional expectations algorithm.

7.3.2 Finer and Stronger Links between OSN Analytics and Established Theories

Many existing approaches, including ours, have made assumptions in order to simplify the model. The exact effect of this practice, however, is far from clear. To this end, we need to extend our framework to accommodate more details in those established theories, while ensuring the resultant model not becoming too complicated and thus intractable. In order to generalize our findings here, a broader coverage of event domains as well as data sources should be considered. Even if one cannot exhaustive all events, it will help alleviate any bias in the present studies.

Many other theories in communication and human behavior should be studied in the setting of OSN, too. For example, prior work [112] has reasoned the motivation behind sharing one's emotion with others, and it is intriguing to study the link between sentiment shift and the behavior of information sharing in OSNs. Are people more like to share information when there is sentiment shift, and if so, from which polarity to which? Does it have different effects on the sharing of emotional content versus non-emotional content? If there is access to corresponding offline activity information, is there ways to distinguish the effect of online versus offline influence? These are all exciting questions to social computing and communication researchers.

The link should also be strengthened between network analytics and graph theories. For instance, in Section 2.3 we have compared the spectra of networks before and after different simplification procedures. While this provides some qualitative cues on the effect of simplification, deeper analyses can be pursued to link it with the rich literature in spectral graph theory [28]. Recently there has been a growing

interest in giving a more theoretical treatment [60] to network simplification techniques that have demonstrated their effectiveness. This effort may help explain the empirical superiority of network simplification with regard to network size reduction while preserving the internal community structure, as well as provide insights on the advantages of leveraging content information in network simplification.

7.3.3 Improved Algorithm Complexity and Quality Guarantee

When working on novel OSN analytics algorithms, we can design them in a way such that the output quality can be bounded in a principled way, for instance, submodular functions [75], or conjugate priors in Bayesian models [121]. At the same time, we should exploit data parallelism and task parallelism, such as the optimization for RC-Joint. Frameworks including but not limited to OpenMP, MPI, and Map-Reduce can be adopted, depending on the underlying computing procedure and the problem size.

Another aspect of interest is the convergence guarantee of iterative algorithms, such as that used in Chapter 3. Without the constrained programming component, the coordinate ascend algorithm will always converge to local maximum. However, the addition of constraints makes the proof more involved, if the algorithm will indeed converge. Such analysis is not only of theoretical interest, but also have practical implication, as analysts can have more precise expectation on the algorithm's behavior.

7.3.4 Real-time OSN Analytics

Our world is full of rapidly-evolving events, thus real-time automatic decision making is of high societal and economical values. Analyses presented in this dissertation are largely done in a batched manner, meaning that the outcome does not

change as soon as new data arrives. To achieve real-time OSN analytics, it is crucial to creatively adapt methods that are designed for streaming data [49, 13]. The common premise of streaming algorithms is that new data input is constantly arriving, while the computer's memory capacity is finite. By assuming a streaming model, the next-generation OSN analytics algorithms should update their decision models (e.g. machine learning classifier, or probability distribution) and produce new results whenever new data comes in.

Bibliography

[1] Lada A Adamic and Natalie Glance. The political blogosphere and the 2004 us election: divided they blog. In *Proceedings of the 3rd international workshop on Link discovery*, pages 36–43. ACM, 2005.

[2] C.C. Aggarwal, Y. Zhao, and P.S. Yu. Outlier detection in graph streams. In *Data Engineering (ICDE), 2011 IEEE 27th International Conference on*, pages 399–409. IEEE, 2011.

[3] Yong-Yeol Ahn, James P Bagrow, and Sune Lehmann. Link communities reveal multiscale complexity in networks. *Nature*, 466(7307):761–764, 2010.

[4] Edoardo M Airoldi, David M Blei, Stephen E Fienberg, and Eric P Xing. Mixed membership stochastic blockmodels. *Journal of Machine Learning Research*, 9(1981-2014):3, 2008.

[5] Alex Arenas, Alberto Fernandez, and Sergio Gomez. Analysis of the structure of complex networks at different resolution levels. *New Journal of Physics*, 10(5):053039, 2008.

[6] L. Backstrom, D. Huttenlocher, J. Kleinberg, and X. Lan. Group formation in large social networks: membership, growth, and evolution. In *SIGKDD'06*, pages 44–54. ACM, 2006.

[7] L. Backstrom and J. Leskovec. Supervised random walks: predicting and recommending links in social networks. In *WSDM'11*, pages 635–644. ACM, 2011.

[8] Gökhan BakIr. *Predicting structured data*. MIT press, 2007.

[9] E. Bakshy, B. Karrer, and L.A. Adamic. Social influence and the diffusion of user-created content. In *EC'09*, pages 325–334. ACM, 2009.

[10] Sigal G Barsade. The ripple effect: Emotional contagion and its influence on group behavior. *Administrative Science Quarterly*, 47(4):644–675, 2002.

[11] Roy F Baumeister, Ellen Bratslavsky, Catrin Finkenauer, and Kathleen D Vohs. Bad is stronger than good. *Review of general psychology*, 5(4):323, 2001.

[12] D.J. Beal, R.R. Cohen, M.J. Burke, and C.L. McLendon. Cohesion and performance in groups: a meta-analytic clarification of construct relations. *Journal of Applied Psychology*, 88(6):989, 2003.

[13] Albert Bifet, Geoff Holmes, Richard Kirkby, and Bernhard Pfahringer. Moa: Massive online analysis. *The Journal of Machine Learning Research*, 11:1601–1604, 2010.

[14] D.M. Blei and J.D. Lafferty. Dynamic topic models. In *ICML'06*, ICML'06, pages 113–120, New York, NY, USA, 2006. ACM.

[15] D.M. Blei, A.Y. Ng, and M.I. Jordan. Latent dirichlet allocation. *JMLR*, 3:993–1022, 2003.

[16] Nyla R Branscombe and Daniel L Wann. The positive social and self concept consequences of sports team identification. *Journal of Sport & Social Issues*, 15(2):115–127, 1991.

[17] A.Z. Broder, M. Charikar, A.M. Frieze, and M. Mitzenmacher. Min-wise independent permutations. In *STOC'98*, pages 327–336. ACM, 1998.

[18] Ceren Budak and Rakesh Agrawal. On participation in group chats on twitter. In *WWW'13*, pages 165–176, 2013.

[19] M. Cha, H. Haddadi, F. Benevenuto, and K. Gummadi. Measuring user influence in twitter: The million follower fallacy. In *ICWSM'04*, 2010.

[20] Meeyoung Cha, Alan Mislove, and Krishna P Gummadi. A measurement-driven analysis of information propagation in the flickr social network. In *Proceedings of the 18th international conference on World wide web*, pages 721–730. ACM, 2009.

[21] C. Chang and C. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[22] M.S. Charikar. Similarity estimation techniques from rounding algorithms. In *STOC'02*, pages 380–388. ACM, 2002.

[23] N. Chawla, K. Bowyer, L. Hall, and W. Kegelmeyer. SMOTE: synthetic minority over-sampling technique. *JAIR*, 16(1):321–357, 2002.

[24] Wei Chen, Chi Wang, and Yajun Wang. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1029–1038. ACM, 2010.

[25] H. Cheng, Y. Zhou, and J.X. Yu. Clustering large attributed graphs: A balance between structural and attribute similarities. *TKDD*, 5(2):12, 2011.

[26] Justin Cheng, Lada Adamic, P Alex Dow, Jon Michael Kleinberg, and Jure Leskovec. Can cascades be predicted? In *Proceedings of the 23rd international conference on World wide web*, pages 925–936. International World Wide Web Conferences Steering Committee, 2014.

[27] Sarvenaz Choobdar, Pedro Rebeiro, Srinivasan Parthasarathy, and Fernando Silva. Dynamic inference of social roles in information cascades. *Data Mining and Knowledge Discovery*, To Appear, 2014.

[28] F.R.K. Chung. *Spectral graph theory*, volume 92. Amer Mathematical Society, 1997.

[29] Aaron Clauset, Mark EJ Newman, and Cristopher Moore. Finding community structure in very large networks. *Physical review E*, 70(6):066111, 2004.

[30] D. Cohn and T. Hofmann. The missing link-a probabilistic model of document content and hypertext connectivity. In *NIPS'01*, volume 13, page 430. The MIT Press, 2001.

[31] Michael Conover, Jacob Ratkiewicz, Matthew Francisco, Bruno Gonçalves, Filippo Menczer, and Alessandro Flammini. Political polarization on twitter. In *ICWSM*, 2011.

[32] Michael D Conover, Clayton Davis, Emilio Ferrara, Karissa McKelvey, Filippo Menczer, and Alessandro Flammini. The geospatial characteristics of a social movement communication network. *PloS one*, 8(3):e55957, 2013.

[33] D. Cosley, D. Huttenlocher, J. Kleinberg, X. Lan, and S. Suri. Sequential influence models in social networks. In *ICWSM'10*, 2010.

[34] Lorenzo Coviello, Yunkyu Sohn, Adam DI Kramer, Cameron Marlow, Massimo Franceschetti, Nicholas A Christakis, and James H Fowler. Detecting emotional contagion in massive social networks. *PloS one*, 9(3):e90315, 2014.

[35] Henriette Cramer, Mattias Rost, and Lars Erik Holmquist. Performing a check-in: emerging practices, norms and'conflicts' in location-sharing using foursquare. In *MobileHCI'11*, pages 57–66. ACM, 2011.

[36] Aron Culotta. Towards detecting influenza epidemics by analyzing twitter messages. In *Proceedings of the first workshop on social media analytics*, pages 115–122. ACM, 2010.

[37] Inderjit S Dhillon, Yuqiang Guan, and Brian Kulis. Weighted graph cuts without eigenvectors a multilevel approach. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(11):1944–1957, 2007.

[38] Maeve Duggan and Aaron Smith. Social media update 2013. *Pew Internet and American Life Project*, 2013.

[39] Lata Dyaram and TJ Kamalanabhan. Unearthed: the other side of group cohesiveness. *Journal of Social Science*, 10(3):185–90, 2005.

[40] E. Erosheva, S. Fienberg, and J. Lafferty. Mixed-membership models of scientific publications. *PNAS*, 101(Suppl 1):5220, 2004.

[41] M. Ester, R. Ge, B.J. Gao, Z. Hu, and B. Ben-Moshe. Joint cluster analysis of attribute data and relationship data: the connected k-center problem. In *SDM'06*, pages 25–46, 2006.

[42] Andrea Esuli and Fabrizio Sebastiani. Sentiwordnet: A publicly available lexical resource for opinion mining. In *Proceedings of LREC*, volume 6, pages 417–422, 2006.

[43] Rosta Farzan, Laura A. Dabbish, Robert E. Kraut, and Tom Postmes. Increasing commitment to online communities by designing for social presence. In *CSCW'11*, pages 321–330, 2011.

[44] Alan Fern and Robert Givan. Sequential inference with reliable observations: Learning to construct force-dynamic models. *Artificial intelligence*, 170(14):1081–1100, 2006.

[45] L. Festinger, S. Schachter, and K. Back. The spatial ecology of group formation. *Social pressure in informal groups*, pages 33–60, 1950.

[46] S. Fortunato. Community detection in graphs. *Physics Reports*, 486(3-5):75–174, 2010.

[47] James H Fowler and Nicholas A Christakis. Dynamic spread of happiness in a large social network: longitudinal analysis over 20 years in the framingham heart study. *Bmj*, 337, 2008.

[48] E. Fox and J. Shaw. Combination of multiple searches. *NIST SPECIAL PUBLICATION SP*, pages 243–243, 1994.

[49] Mohamed Medhat Gaber, Arkady Zaslavsky, and Shonali Krishnaswamy. Mining data streams: a review. *ACM Sigmod Record*, 34(2):18–26, 2005.

[50] Sean Gilpin, Tina Eliassi-Rad, and Ian Davidson. Guided learning for role discovery (glrd): framework, algorithms, and applications. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 113–121. ACM, 2013.

[51] Michelle Girvan and Mark EJ Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826, 2002.

[52] David F Gleich and C Seshadhri. Vertex neighborhoods, low conductance cuts, and good seeds for local community methods. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 597–605. ACM, 2012.

[53] J. Gomide, A. Veloso, W. Meira Jr, V. Almeida, F. Benevenuto, F. Ferraz, and M. Teixeira. Dengue surveillance based on a computational model of spatio-temporal locality of twitter. In *WebSci'11*. ACM, 2011.

[54] Pollyanna Gonçalves, Matheus Araújo, Fabrício Benevenuto, and Meeyoung Cha. Comparing and combining sentiment analysis methods. In *Proceedings of the first ACM conference on Online social networks*, pages 27–38. ACM, 2013.

[55] P.A. Grabowicz, L.M. Aiello, V.M. Eguíluz, and A. Jaimes. Distinguishing topical and social groups based on common identity and bond theory. In *WSDM'13*, 2013.

[56] T.L. Griffiths and M. Steyvers. Finding scientific topics. *PNAS*, 101(Suppl 1):5228, 2004.

[57] Anatoliy Gruzd, Sophie Doiron, and Philip Mai. Is happiness contagious online? a case of twitter and the 2010 winter olympics. In *System Sciences (HICSS), 2011 44th Hawaii International Conference on*, pages 1–9. IEEE, 2011.

[58] Jamie Guillory, Jason Spiegel, Molly Drislane, Benjamin Weiss, Walter Donner, and Jeffrey Hancock. Upset now?: emotion contagion in distributed groups. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 745–748. ACM, 2011.

[59] S. Günnemann, B. Boden, and T. Seidl. Db-csc: a density-based approach for subspace clustering in graphs with feature vectors. In *PKDD 2011*, pages 565–580. Springer-Verlag, 2011.

[60] Rishi Gupta, Tim Roughgarden, and C Seshadhri. Decompositions of triangle-dense graphs. In *Proceedings of the 5th conference on Innovations in theoretical computer science*, pages 471–482. ACM, 2014.

[61] Jeffrey T Hancock, Kailyn Gee, Kevin Ciaccio, and Jennifer Mae-Hwah Lin. I'm sad you're sad: emotional contagion in cmc. In *Proceedings of the 2008 ACM conference on Computer supported cooperative work*, pages 295–298. ACM, 2008.

[62] D. Hanisch, A. Zien, R. Zimmer, and T. Lengauer. Co-clustering of biological networks and gene expression data. *Bioinformatics*, 18(suppl 1):S145–S154, 2002.

[63] Elaine Hatfield and John T Cacioppo. *Emotional contagion*. Cambridge university press, 1994.

[64] K. Henderson, B. Gallagher, T. Eliassi-Rad, H. Tong, S. Basu, L. Akoglu, D. Koutra, C. Faloutsos, L. Li, Y. Matsubara, et al. Rolx: Structural role extraction & mining in large graphs. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1231–1239. ACM, 2012.

[65] K. Henderson, B. Gallagher, L. Li, L. Akoglu, T. Eliassi-Rad, H. Tong, and C. Faloutsos. It's who you know: graph mining using recursive structural features. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 663–671. ACM, 2011.

[66] Alison L Hill, David G Rand, Martin A Nowak, and Nicholas A Christakis. Emotions as infectious diseases in a large social network: the sisa model. *Proceedings of the Royal Society B: Biological Sciences*, page rspb20101217, 2010.

[67] CJ Hutto and Eric Gilbert. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Eighth International AAAI Conference on Weblogs and Social Media*, 2014.

[68] Glen Jeh and Jennifer Widom. Simrank: a measure of structural-context similarity. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 538–543. ACM, 2002.

[69] S.R. Kairam, D.J. Wang, and J. Leskovec. The life and death of online groups: Predicting group growth and longevity. In *ICWSM'12*, pages 673–682. ACM, 2012.

[70] D.R. Karger. Random sampling in cut, flow, and network design problems. *Mathematics of Operations Research*, 24(2):383–413, 1999.

[71] George Karypis and Vipin Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on scientific Computing*, 20(1):359–392, 1998.

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

[73] Adam DI Kramer. The spread of emotion via facebook. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 767–770. ACM, 2012.

[74] Adam DI Kramer, Jamie E Guillory, and Jeffrey T Hancock. Experimental evidence of massive-scale emotional contagion through social networks. *Proceedings of the National Academy of Sciences*, page 201320040, 2014.

[75] Andreas Krause and Daniel Golovin. Submodular function maximization. *Tractability: Practical Approaches to Hard Problems*, 3:19, 2012.

[76] Andrey Kupavskii, Liudmila Ostroumova, Alexey Umnov, Svyatoslav Usachev, Pavel Serdyukov, Gleb Gusev, and Andrey Kustarev. Prediction of retweet cascade size over time. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 2335–2338. ACM, 2012.

[77] M.H. Kutner, C. Nachtsheim, and J. Neter. *Applied linear regression models, 4^{th} Edition*. McGraw-Hill New York, NY, 2004.

[78] H. Kwak, C. Lee, H. Park, and S. Moon. What is Twitter, a social network or a news media? In *WWW'10*, pages 591–600. ACM, 2010.

[79] Kristina Lerman and Rumi Ghosh. Information contagion: An empirical study of the spread of news on digg and twitter social networks. *ICWSM*, 10:90–97, 2010.

[80] J. Leskovec, L.A. Adamic, and B. Huberman. The dynamics of viral marketing. *ACM Transactions on the Web (TWEB)*, 1(1):5, 2007.

[81] J. Leskovec, K.J. Lang, A. Dasgupta, and M.W. Mahoney. Statistical properties of community structure in large social and information networks. In *WWW'08*, pages 695–704. ACM, 2008.

[82] J. Leskovec, K.J. Lang, and M. Mahoney. Empirical comparison of algorithms for network community detection. In *Proceedings of the 19th international conference on World wide web*, pages 631–640. ACM, 2010.

[83] Jure Leskovec, Lars Backstrom, and Jon Kleinberg. Meme-tracking and the dynamics of the news cycle. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 497–506. ACM, 2009.

[84] Jure Leskovec and Christos Faloutsos. Sampling from large graphs. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 631–636. ACM, 2006.

[85] Chin-Yew Lin, Guihong Cao, Jianfeng Gao, and Jian-Yun Nie. An information-theoretic approach to automatic evaluation of summaries. In *NAACL HLT'06*, pages 463–470. Association for Computational Linguistics, 2006.

[86] Jianhua Lin. Divergence measures based on the shannon entropy. *Information Theory, IEEE Transactions on*, 37(1):145–151, 1991.

[87] Zhifeng Liu, Tingting Zhang, and Qiujun Lan. An extended sisa model for sentiment contagion. *Discrete Dynamics in Nature and Society*, 2014, 2014.

[88] A.J. Lott and B.E. Lott. Group cohesiveness as interpersonal attraction: A review of relationships with antecedent and consequent variables. *Psychological bulletin*, 64(4):259, 1965.

[89] Annie Louis and Ani Nenkova. Automatically assessing machine summary content without a gold standard. *Computational Linguistics*, 39(2):267–300, 2013.

[90] Zongyang Ma, Aixin Sun, and Gao Cong. On predicting the popularity of newly emerging hashtags in twitter. *Journal of the American Society for Information Science and Technology*, 64(7):1399–1410, 2013.

[91] A.S. Maiya and T.Y. Berger-Wolf. Sampling community structure. In *WWW 2010*, pages 701–710. ACM, 2010.

[92] Andrew McCallum, Xuerui Wang, and Andrés Corrada-Emmanuel. Topic and role discovery in social networks with experiments on enron and academic email. *J. Artif. Intell. Res.(JAIR)*, 30:249–272, 2007.

[93] A. Mislove, M. Marcon, K.P. Gummadi, P. Druschel, and B. Bhattacharjee. Measurement and analysis of online social networks. In *SIGCOMM'07*, pages 29–42. ACM, 2007.

[94] Saif M. Mohammad, Svetlana Kiritchenko, and Xiaodan Zhu. Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets. In *Proceedings of the seventh international workshop on Semantic Evaluation Exercises (SemEval-2013)*, Atlanta, Georgia, USA, June 2013.

[95] B. Mohar. The laplacian spectrum of graphs. *Graph theory, combinatorics, and applications*, 2:871–898, 1991.

[96] M. Montague and J.A. Aslam. Relevance score normalization for metasearch. In *CIKM'01*, pages 427–433. ACM, 2001.

[97] J. Moody and D.R. White. Structural cohesion and embeddedness: A hierarchical concept of social groups. *American Sociological Review*, pages 103–127, 2003.

[98] B. Mullen and C. Copper. The relation between group cohesiveness and performance: An integration. *Psychological Bulletin; Psychological Bulletin*, 115(2):210, 1994.

[99] Preslav Nakov, Sara Rosenthal, Zornitsa Kozareva, Veselin Stoyanov, Alan Ritter, and Theresa Wilson. Semeval-2013 task 2: Sentiment analysis in twitter.

[100] R.M. Nallapati, A. Ahmed, E.P. Xing, and W.W. Cohen. Joint latent topic models for text and citations. In *SIGKDD'08*, pages 542–550. ACM, 2008.

[101] Alexandra Olteanu, Carlos Castillo, Fernando Diaz, and Sarah Vieweg. Crisislex: A lexicon for collecting and filtering microblogged communications in crises. In *Proceedings of the Eighth International Conference on Weblogs and Social Media, ICWSM 2014, Ann Arbor, Michigan, USA, June 1-4, 2014.*, 2014.

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

[103] B. Pang and L. Lee. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1-2):1–135, 2008.

[104] S Parthasarathy, Y Ruan, and V Satuluri. Community discovery in social networks: Applications, methods and emerging trends. *Social Network Data Analytics*, pages 79–113, 2011.

[105] Srinivasan Parthasarathy and S. M. Faisal. Network clustering. In *Data Clustering: Algorithms and Applications*, pages 415–456. 2013.

[106] M. Pennacchiotti and A.M. Popescu. A machine learning approach to twitter user classification. In *ICWSM'11*, 2011.

[107] James W Pennebaker, Cindy K Chung, Molly Ireland, Amy Gonzales, and Roger J Booth. The development and psychometric properties of liwc2007. *Austin, TX, LIWC. Net*, 2007.

[108] Sasa Petrovic, Miles Osborne, and Victor Lavrenko. Rt to win! predicting message propagation in twitter. In *ICWSM*, 2011.

[109] H. Purohit, A. Dow, O. Alonso, L. Duan, and K. Haas. User taglines: Alternative presentations of expertise and interest in social media. In *Proceedings of the first ASE International Conference on Social Informatics*. IEEE, 2012.

[110] D. Rao, M. Paul, C. Fink, D. Yarowsky, T. Oates, and G. Coppersmith. Hierarchical bayesian models for latent attribute detection in social media. *ICWSM'11*, pages 598–601, 2011.

[111] Y. Ren, R. Kraut, and S. Kiesler. Applying common identity and bond theory to design of online communities. *Organization Studies*, 28(3):377–408, 2007.

[112] Bernard Rimé. Interpersonal emotion regulation. *Handbook of emotion regulation*, pages 466–485, 2007.

[113] Alan Ritter, Sam Clark, Oren Etzioni, et al. Named entity recognition in tweets: an experimental study. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1524–1534. Association for Computational Linguistics, 2011.

[114] Daniel M Romero, Wojciech Galuba, Sitaram Asur, and Bernardo A Huberman. Influence and passivity in social media. *Machine learning and knowledge discovery in databases*, pages 18–33, 2011.

[115] D.M. Romero, B. Meeder, and J. Kleinberg. Differences in the mechanics of information diffusion across topics: Idioms, political hashtags, and complex contagion on twitter. In *WWW'11*, pages 695–704. ACM, 2011.

[116] Ryan A Rossi, Brian Gallagher, Jennifer Neville, and Keith Henderson. Modeling dynamic behavior in large evolving graphs. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 667–676. ACM, 2013.

[117] Martin Rosvall and Carl T Bergstrom. Multilevel compression of random walks on networks reveals hierarchical organization in large integrated systems. *PloS one*, 6(4):e18209, 2011.

[118] Y. Ruan, H. Purohit, D. Fuhry, S. Parthasarathy, and A. Sheth. Prediction of topic volume on twitter. In *Web Science 2012, WebSci '12, Evanston, IL, USA - June 22 - 24, 2012*, pages 397–402. ACM, 2012.

[119] T. Sakaki, M. Okazaki, and Y. Matsuo. Earthquake shakes twitter users: real-time event detection by social sensors. In *WWW'10*, pages 851–860. ACM, 2010.

[120] V. Satuluri and S. Parthasarathy. Scalable graph clustering using stochastic flows: applications to community discovery. In *SIGKDD'09*, pages 737–746. ACM, 2009.

[121] Venu Satuluri and Srinivasan Parthasarathy. Bayesian locality sensitive hashing for fast similarity search. *Proceedings of the VLDB Endowment*, 5(5):430–441, 2012.

[122] Venu Satuluri, Srinivasan Parthasarathy, and Yiye Ruan. Local graph sparsification for scalable clustering. In *Proceedings of the 2011 international conference on Management of data*, pages 721–732. ACM, 2011.

[123] X. Shi, J. Zhu, R. Cai, and L. Zhang. User grouping behavior in online forums. In *KDD'09*, pages 777–786. ACM, 2009.

[124] Tom AB Snijders and Krzysztof Nowicki. Estimation and prediction for stochastic blockmodels for graphs with latent block structure. *Journal of Classification*, 14(1):75–100, 1997.

[125] Philip J Stone, Dexter C Dunphy, and Marshall S Smith. The general inquirer: A computer approach to content analysis. 1966.

[126] A. Strehl and J. Ghosh. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *The Journal of Machine Learning Research*, 3:583–617, 2003.

[127] B. Suh, L. Hong, P. Pirolli, and E. Chi. Want to be retweeted? large scale analytics on factors impacting retweet in twitter network. In *SocialCom'10*, pages 177–184. IEEE, 2010.

[128] Gabor Szabo and Bernardo A Huberman. Predicting the popularity of online content. *Communications of the ACM*, 53(8):80–88, 2010.

[129] H. Tajfel, M.G. Billig, R.P. Bundy, and C. Flament. Social categorization and intergroup behaviour. *European Journal of Social Psychology*, 1(2):149–178, 1971.

[130] W. Tang, Z. Lu, and I.S. Dhillon. Clustering with multiple graphs. In *ICDM'09*, pages 1016–1021. IEEE, 2009.

[131] Y. Tian, R.A. Hankins, and J.M. Patel. Efficient aggregation for graph summarization. In *SIGMOD'08*, pages 567–580, 2008.

[132] Y. Tsuruoka and J. Tsujii. Bidirectional inference with the easiest-first strategy for tagging sequence data. In *HLT/EMNLP'05*, pages 467–474. ACL, 2005.

[133] B.W. Tuckman. Developmental sequence in small groups. *Psychological bulletin*, 63(6):384, 1965.

[134] A. Tumasjan, T.O. Sprenger, P.G. Sandner, and I.M. Welpe. Predicting elections with twitter: What 140 characters reveal about political sentiment. In *ICWSM'10*, pages 178–185, 2010.

[135] J.C. Turner. Towards a cognitive redefinition of the social group. *Social identity and intergroup relations*, pages 15–40, 1982.

[136] J.C. Turner, M.A. Hogg, P.J. Oakes, S.D. Reicher, and M.S. Wetherell. *Rediscovering the social group: A self-categorization theory.* Basil Blackwell, 1987.

[137] I. Ulitsky and R. Shamir. Identification of functional modules using network topology and high-throughput data. *BMC Systems Biology*, 1(1):8, 2007.

[138] S.M. van Dongen. Graph clustering by flow simulation. *PhD Thesis*, 2000.

[139] G. Ver Steeg and A. Galstyan. Information-theoretic measures of influence based on content dynamics. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 3–12. ACM, 2013.

[140] Lu Wang and Claire Cardie. Improving agreement and disagreement identification in online discussions with a socially-tuned sentiment lexicon. *ACL 2014*, page 97, 2014.

[141] J. Weng, E.P. Lim, J. Jiang, and Q. He. Twitterrank: finding topic-sensitive influential twitterers. In *WSDM'10*, pages 261–270. ACM, 2010.

[142] D.R. White and F. Harary. The cohesiveness of blocks in social networks: Node connectivity and conditional density. *Sociological Methodology*, 31(1):305–359, 2001.

[143] Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the conference on human language technology and empirical methods in natural language processing*, pages 347–354. Association for Computational Linguistics, 2005.

[144] Wendy Wood, Sharon Lundgren, Judith A Ouellette, Shelly Busceme, and Tamela Blackstone. Minority influence: a meta-analytic review of social influence processes. *Psychological bulletin*, 115(3):323, 1994.

[145] F. Wu and B. Huberman. Popularity, novelty and attention. In *EC'08*, pages 240–245. ACM, 2008.

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

[147] Jaewon Yang and Jure Leskovec. Modeling information diffusion in implicit networks. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on*, pages 599–608. IEEE, 2010.

[148] 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*, pages 587–596. ACM, 2013.

[149] T. Yang, R. Jin, Y. Chi, and S. Zhu. Combining link and content for community detection: a discriminative approach. In *SIGKDD'09*, pages 927–936. ACM, 2009.

[150] Wayne W Zachary. An information flow model for conflict and fission in small groups. *Journal of anthropological research*, pages 452–473, 1977.

[151] Laijun Zhao, Jiajia Wang, Rongbing Huang, Hongxin Cui, Xiaoyan Qiu, and Xiaoli Wang. Sentiment contagion in complex networks. *Physica A: Statistical Mechanics and its Applications*, 394:17–23, 2014.

[152] D. Zhou, E. Manavoglu, J. Li, C.L. Giles, and H. Zha. Probabilistic models for discovering e-communities. In *WWW'06*, pages 173–182. ACM, 2006.

[153] Y. Zhou, H. Cheng, and J.X. Yu. Clustering large attributed graphs: An efficient incremental approach. In *ICDM 2010*, pages 689–698. IEEE, 2010.

[154] S. Zhu, K. Yu, Y. Chi, and Y. Gong. Combining content and link for classification using matrix factorization. In *SIGIR'07*, pages 487–494. ACM, 2007.