SOCIAL NETWORK MODELING, LINK PREDICTION, AND
SENTIMENT IMPACT ANALYSIS

A Dissertation in
Computer Science and Engineering

by

Baojun Qiu

© 2011 Baojun Qiu

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Doctor of Philosophy

December 2011
The dissertation of Baojun Qiu was reviewed and approved* by the following:

John Yen
Professor of Information Sciences and Technology
Dissertation Advisor
Co-chair of Committee

Guohong Cao
Professor of Computer Science and Engineering
Co-chair of Committee

Jesse Barlow
Professor of Computer Science and Engineering

Jia Li
Associate Professor of Statistics

Qi He
Research Staff Member of IBM Almaden Research Center

Raj Acharya
Professor of Computer Science and Engineering
Head of the Department of Computer Science and Engineering

*Signatures are on file in the Graduate School.
Abstract

Social network dynamics analysis is one of the most important fields in social network analysis. It studies the temporal network structure and impacts on actors at both macro and micro levels. Specifically, social network modeling is the macro-study of network structure. It identifies general principles of link formation that lead to interesting network properties. Link prediction is the micro-study of network structure. It predicts future links between nodes. The impact analysis investigates impacts of links on individual nodes, and impact dynamics of the networks.

Although the links between actors in social networks are the results of social interactions between the actors, most of the current network dynamics analysis is link centric. For instance, social network modeling studies networks by simulating individual links. A link centric approach for dynamics analysis has the following problems: 1) difficult to model n-ary \((n > 2)\) relationship; 2) difficult to include properties of social interaction in modeling network dynamics; 3) difficult to aggregate properties of past social interactions in network modeling and prediction;
4) difficult to analyze impacts of social interactions (on actors and networks) using their properties.

To solve the problems, we developed new approaches for social dynamics analysis from the perspective of social interactions. Specifically, to model n-ary \((n \geq 2)\) social interactions and incorporate their properties in modeling networks, we investigate an event-driven social network modeling to model behavior patterns of multi-actors social interactions. To aggregate properties of past social interactions to predict future links between actors, we develop behavior evolution based link prediction to discover temporal behavior patterns for predicting future links. Finally, we use the characteristics of social interactions to investigate the sentiment impact of interactions in an online health community.

The experimental results of social network modeling suggest that our event-driven social network modeling can generate realistic networks exhibiting important macroscopic properties, such as power-law degree distribution, hierarchical community structure and assortativity which are similar to real networks. The experimental results of link prediction indicate that our behavior evolution based link prediction approach consistently achieves significant improvement on link prediction accuracy on multiple real networks. Finally, our work in sentiment impact analysis discovers the patterns of sentiment change of members of a online health community, and identify factors that affect the sentiment change. These research results indicate the benefits of modeling social interactions directly for characterizing and predicting dynamic behaviors of social networks at both macroscopic and microscopic levels.
Table of Contents

List of Figures ........................................ viii
List of Tables ........................................... xi
Acknowledgments ........................................ xii

Chapter 1
Introduction ........................................... 1
1.1 Complex Network Analysis ............................... 1
1.2 Dissertation Overview ................................ 3
   1.2.1 Event-driven Modeling .............................. 6
   1.2.2 Behavior Evolution Based Link Prediction .......... 8
   1.2.3 Sentiment Impact Analysis of Online Health Communities . 10

Chapter 2
Event-driven Modeling ................................... 13
2.1 Introduction ......................................... 14
2.2 Related Work ........................................ 17
2.3 Observations and Motivations ......................... 20
   2.3.1 Growth Rates ..................................... 21
   2.3.2 The Number of Participants in Events and Its Evolution . 23
   2.3.3 The Selection of Participants for Events .......... 26
      2.3.3.1 Distribution of Lifetime of Nodes ............... 26
      2.3.3.2 Effects of Locality and Attachedness .......... 29
      2.3.3.3 The Span Distance of New Edges vs. the Degree of Nodes ........................................ 31
2.4 An Event-driven Framework and A Hybrid Growth Model .... 33
2.5 Simulation and Evaluation ................................. 37
  2.5.1 Event-driven vs. Non-event-driven ..................... 37
  2.5.2 The Effect of Average Number of Participants in Event-
            driven Models .................................. 39
  2.5.3 Behavior Evolution vs. None Behavior Evolution ....... 39
  2.5.4 Comparisons between EELAG, APA and ADG on Simu-
            lating NanoSCI .................................. 40
      2.5.4.1 Degree Distribution ............................ 42
      2.5.4.2 Degree-dependent Clustering Coefficients ......... 42
      2.5.4.3 The Average Degree of the Nearest Neighbors ... 43
  2.6 Conclusions ........................................... 44

Chapter 3
Behavior Evolution Based Link Prediction 48
  3.1 Link Prediction ......................................... 49
  3.2 Related Work ........................................... 53
  3.3 Model Node Behavior Evolution with Time Series Features .... 56
      3.3.1 Temporal Sequence of Node Behavior ................. 56
      3.3.2 Time Series Features .............................. 58
          3.3.2.1 Simple Features ............................. 59
          3.3.2.2 Local Pattern .............................. 59
          3.3.2.3 Prediction ................................. 60
          3.3.2.4 Interplay .................................. 62
  3.4 Hybrid Temporal Model for Prediction of Node Behavior .... 62
      3.4.1 Collaborative Time Series Prediction ............... 64
      3.4.2 Hybrid Time Series Prediction Model ............... 67
  3.5 Behavior Evolution Based Link Prediction Approach ....... 72
      3.5.1 The Interplay Likelihood of Two Nodes ............. 75
      3.5.2 EvPred: Evolution-based Link Prediction Approach ... 77
  3.6 Evaluation ............................................. 79
      3.6.1 Datasets ......................................... 79
      3.6.2 Experimental Setup ................................ 80
          3.6.2.1 Preprocessing ............................... 80
          3.6.2.2 Baselines .................................. 83
          3.6.2.3 Metrics ................................... 84
      3.6.3 Link Prediction Performance ........................ 86
          3.6.3.1 Balanced Data: Impact of Positive/Negative Ratio 86
          3.6.3.2 Balanced Data: Impact of Negative Sample Sources 87
          3.6.3.3 Imbalanced Data ............................ 88
          3.6.3.4 Node Ranking Data ........................... 91
List of Figures

1.1 Complex networks .................................................. 2
1.2 The history of complex network analysis ...................... 3
1.3 Focus of this dissertation ........................................... 5

2.1 The number of papers (events) and the number of collaborations (edges) increase linearly in log-log scale with the number of authors (nodes). ......................................................... 22
2.2 Distribution of the number of secondary authors per paper vs. Poisson distribution in both NanoSCI and NanoParticle. .............. 24
2.3 The average number of coauthors per paper evolve linearly in semi-log scale with the number of total nodes in both NanoSCI and NanoParticle ............................................................... 25
2.4 Distribution of Lifetimes in years and Seniority distributions of nodes in NanoSCI in 2005 ......................................................... 27
2.5 Degree distributions of all nodes in NanoSCI and of inactive nodes obey power law like dependency with an exponential cut-off $P(k) \propto k^{-\tau} e^{-k/k_c}$ ......................................................... 28
2.6 The percentage of nodes with certain degrees that form new connections. ................................................................. 30
2.7 The percentage of node pairs with certain distance that form new connections ................................................................. 31
2.8 Distributions of span distances of the new edges connected from nodes with certain degrees in NanoSCI in 2005. Distance $r = Inf$ indicates edges connected to disconnected nodes, and $r = -1$ indicates edges connected to new nodes with degree 0 .......... 33
2.9 The average local clustering coefficients as a function of the degree of nodes (event-driven vs. non-event driven) .................. 38
2.10 The average local clustering coefficients as a function of the degree of nodes (different average number of participants in event-driven models) ......................................................... 40
2.11 Degree distributions (behavior evolution vs. none behavior evolution). 41
2.12 Degree distributions (EELAG, APA and ADG). 43
2.13 The average local clustering coefficients as a function of the degree of nodes (EELAG, APA and ADG). 44
2.14 The average degree of neighbors as a function of degree (EELAG, APA and ADG). 45

3.1 Illustration of the calculation of Local Pattern 61
3.2 The correlation of similarity of time series at length \(j\) and their value difference at position of \(j+1\). The mean correlation of \(NanoSCI = 0.516, Phone = 0.506\) and \(Facebook = 0.468\) 65
3.3 The standard error of prediction by different models 70
3.4 A summarization of supervised link prediction approach 74
3.5 The illustration of our approach vs. existing approaches 75
3.6 Distribution of distance of node pairs in samples 82
3.7 Performance (AUC and Precision) of different models on data trials containing different percentage of positive samples. (a) and (b) are performance on \(NanoSCI\) in \(AUC\) and \(precision\) respectively, (c) and (d) are performance on \(Phone\), and (e) and (f) are performance on \(Facebook\). 88
3.8 Performance (AUC and Precision) of different models on data trials containing different type of negative samples. (a) and (b) are performance on \(NanoSCI\) in \(AUC\) and \(precision\) respectively, (c) and (d) are performance on \(Phone\), and (e) and (f) are performance on \(Facebook\). 89
3.9 Performance (Average NDCG) 92
3.10 Boxplots of distance of Bonus and Com links on three datasets 94
3.11 The distribution of mean degrees of Bonus and Common on three datasets 95

4.1 Distributions of CSN participants’ contribution (initial posts and total posts) to the online forum. 105
4.2 Distribution of the number of replies to threads in CSN forums. 106
4.3 Distribution of the time span of threads in CSN forums. 107
4.4 Boxplots of features for negative (Neg) and positive (Pos) classes. 114
4.5 Circle: \(y = p(S_1 = +|S_0 = +, n_1 \geq x); \) Triangle: \(y = p(S_1 = +|S_0 = -, n_1 \geq x), \) where \(n_1\) is the number of replies from people other than the thread originator, \(S_0\) and \(S_1\) are initial-sentiment and subsequent-sentiment, respectively. 118
4.6 Circle: \( y = p(S_1 = +| S_0 = +, n_0 \geq x) \); Triangle: \( y = p(S_1 = +| S_0 = -, n_1 \geq x) \), where \( n_0 \) is the number of self-replies by the thread originator, \( S_0 \) and \( S_1 \) are initial-sentiment and subsequent-sentiment.  

4.7 Distribution of Sentiment Change Indicator (\( \Delta p_r \)).  

4.8 Average length of replies is not a major driver of Sentiment Change Indicator \( \Delta p_r \).  

4.9 Increasing Sentiment Indicator of replies by other people increases the originator’s Sentiment Change Indicator.  

4.10 Sentiment Change Indicator (\( \Delta p_r \)) vs. the time elapsed before the first reply.
### List of Tables

3.1 Example time series features for $E_v$ ................................................. 58  
3.2 List of features for two nodes $v_i$ and $v_j$. ........................................... 63  
3.3 Student t-test of the mean standard error of CoTSP, ARIMA, and HybridTSP on NanoSCI, Phone, and Facebook .............................. 69  
3.4 Performance comparison on Phone ...................................................... 90  
3.5 Performance comparison on Facebook .................................................. 90  
3.6 Performance comparison on NanoSCI .................................................. 90  
3.7 Distribution of four types of predictions. ............................................. 92  
3.8 Performance of individual feature (measured by $AUC$) ......................... 97  
4.1 Summary statistics for CSN forum thread data. ................................. 106  
4.2 Examples of CSN posts and their classification. ................................. 109  
4.3 Features for a post .................................................................................. 111  
4.4 Best fit classifiers, their ROC area, and classification accuracy. ........... 113  
4.5 Importance of features assessed individually or as last added for the AdaBost Classifier ................................................................. 115  
4.6 Definition of initial-sentiment ($S_0(T)$) and subsequent-sentiment ($S_1(T)$) of a thread ($T$). ......................................................... 117  
4.7 Factors for sentiment change of thread originators ............................... 125
Acknowledgments

I would like to thank my advisor John Yen for his advice and support over the last five years. I would also like to thank my dissertation committee, Jesse Barlow, Guohong Cao (co-chair), Qi He, and Jia Li for their advice and comments (Jia Li is also my advisor for my concurrent master degree in Statistics).

I have an amazing group of coauthors and collaborators, including Eun Yeong Ahn, Guruprasad Airy, Cornelia Caragea, Rich Caruana (Microsoft), Bi Chen, Po-chun Chen, Umer Farooq, Hank Foley, Greta Greer (American Cancer Society), Kristinka Ivanova, Pei Jian, Hyun-Woo Kim, Giles Lee, Zhen Lei, Peng Liu, Prasenjit Mitra, Jon Morgan, Sooyoung Oh, Kenneth Portier (American Cancer Society), Fang Qi, Mu Qiao, Frank Ritter, Dinghao Wu, Jung-Woo Sohn, Yang Song, Tian Yuan, Haizheng Zhang, Luke Zhang, Kang Zhao, and Shizhuo Zhu. I thank all of them for their help with my research and studies. Specially, Haizheng Zhang and Rich Caruana were my mentors for my internship at Microsoft Research and Bing.

I thank Karen Corl and Vicki G. Keller for their help with my study in the CSE Department.

I thank my fellow CSEers, ISTers, and STATers.
Finally, I thank my wife, parents, parents in law, brother and sister for love, encouragement, advice, and support.
Chapter 1

Introduction

1.1 Complex Network Analysis

Networks have been widely used to model and study structures made up of individuals/organizations called nodes, actors, or vertices, which are connected by some types of interdependency. The connections are called links or edges. A complex network is a network with non-trivial topological features that often occur in real graphs but not in simple networks such as random graphs [23, 11] or grid graphs [85]. Most real-world networks are complex networks. Figure 1.1 lists some examples of real-world complex networks. Specially, social network is a type of complex network that model social structure, such as people’s friendship, kinship, coauthorship, and common interest.

The study of networks can be traced back to the pioneering work on the mathematical graph theory, which originated in 1736 when Leonhard Euler, a Swiss
Figure 1.1. Complex networks

mathematician, who solved the Konigsberg bridge problem (find a round trip that traverse exactly once each of the bridges of the Prussian city of Konigsberg). In 1950-1960s, random networks (graphs simulated by some simple random processes) had been studied by Rapoport [76, 77, 78], and Erdos and Renyi [23, 24, 25]. In recent years, the advances of modern computers and Internet has made it possible and convenient to acquire, store and analyze real world complex networks, such as large scale social networks, neuronal networks, and computer networks. Many interesting common characteristics and patterns have been discovered in real networks which are significantly different from random networks [11]. Some examples of the common network properties presented in many real networks include small-world effects [94, 4], power-law degree distribution [5], community structures [27], assortative mixing [63, 80], and also many kinetic properties exhibited in the growing process of social networks [6, 51, 45], e.g., the shrinking diameter phenomenon. These findings largely improved people’s understanding of complex systems in our world, and inspired people’s great interests in complex network analysis. Figure 1.2 summarizes the history of the complex network analysis.

Complex network analysis nowadays becomes a focus of attention and at-
tracts scientists from many different fields, such as anthropology, sociology, biology, physics, economics, geography and information sciences. Current research interests mainly include applying the developed concepts and measurements to real data and situations, developing new concepts and measurements to describe and characterize network properties, studying network structure and function, analyzing the dynamics and evolution of network topology, and application of social networks analysis [2, 61, 18, 10, 48, 28, 57, 12].

1.2  Dissertation Overview

Analyzing the dynamics and evolution of network topology is one of the most important fields in social network analysis. It studies the temporal network structure and temporal impacts of actors at both macro and micro levels. Social network modeling is the macro-study of network structure. It identifies general principles of link formation that lead to interesting network properties such as community structure and power-law degree distributions. Link predict is the micro-study of network structure. It discovers the patterns of the formation of links from tem-
poral networks, and use the patterns to predict future links between nodes. The impact analysis analyzes impacts of links on individual nodes (micro-level), and impact dynamics of the networks (macro-level).

As illustrated in 1.3. There are two snapshots of a network at time $T$ and $T + \Delta T$, respectively, in the figure:

1. Social network modeling
   - Illustrated in purple dash curves in the figure. The network at time $T + \Delta T$ begins to exhibit community structure. Each circle indicates a community containing a set of nodes, and some nodes may belong to more than one community.

2. Link prediction
   - Illustrated in a red dot-dash curve in the figure—whether a new link will form between two a pair of nodes in future.

3. Impacts of links on nodes
   - Illustrated in a green dot-dot-dash curve in the figure—what changes occur on an individual node after the node receives new links.

The links between actors in social networks are the results of social interactions between the actors. However, most of the current network dynamics analysis is link centric. For instance, social network modeling studies networks by simulating
Figure 1.3. Focus of this dissertation

A link centric approach for dynamics analysis has the following problems:

- Difficult to model n-ary ($n > 2$) relationship;
- Difficult to include properties of social interaction in modeling network dynamics;
- Difficult to aggregate properties of past social interactions in network modeling and prediction;
- Difficult to analyze impacts of social interactions (on actors and networks) using their properties.

To solve the problems, we study social dynamics analysis from the perspective
of social interactions in this dissertation, including event-driven modeling, behavior evolution based link prediction, and sentiment impact analysis of online health communities. To model n-ary \((n \geq 2)\) social interactions and incorporate their properties in modeling networks, we investigate an event-driven social network modeling to model behavior patterns of multi-actors social interactions. To aggregate properties of past social interactions to predict future links between actors, we develop behavior evolution based link prediction to discover and use temporal behavior patterns to predict future links. Finally, we use the characteristics of social interactions to investigate the sentiment impact of interactions in an online health community.

In the following sections of this chapter, we overview the motivations, research questions and contributions of our work on the aforementioned three aspects, respectively.

### 1.2.1 Event-driven Modeling

One of the focuses of social network research is social network modeling—studies the interaction of actors that leads to the non-random macroscopic properties exhibited in real networks, such as power-law degree distributions, hierarchical community structure, and assortative mixing, which are observed in real networks. In many social networks, the connections between actors are formed because they participate in the same social interaction, such as a set of scholars coauthoring a
paper, and several people discussing in a thread at an online forum. However, most of the existing growth models are link-centric models and neglect the structure of social interactions. We therefore develop event-driven modeling of social networks from the perspective of social interactions [71, 73].

The research questions include,

- How to model n-ary (n=2) social interactions in social networks and incorporate their properties in modeling social networks?

- What are the advantages of social interaction based approaches?

To solve these questions, we first investigate and characterize the growth dynamics of real world social networks from the perspective of social interactions. We then propose an event-driven framework and models to capture the growth dynamics of social networks through modeling of the social events. We also incorporate the evolution of event formation and the joint effect of attachedness and locality into our model. The experimental results suggest that our approach can generate realistic graphs exhibiting important macroscopic properties such as power-law degree distributions, hierarchical community structure, and assortative mixing [72, 101].

Our work has the following contributions,

- We systematically develop the event-driven idea to model social networks from the perspective of social interactions. An event-driven approach is a
natural model for the growth of event-based networks. Also, in the event-driven context, we can model richer information than link-centric modeling. For example, event-driven models can model the event leader and follower, and other more informative features of events.

- The event-driven models can generate networks exhibiting similar structural properties to those of real networks.

The work has applications in forecasting future network structure and simulating the evolution of networks.

1.2.2 Behavior Evolution Based Link Prediction

Another important aspect of our work is link prediction—to discover the patterns of the formation of links from the historic evolution of networks, and use them to predict future links between nodes. Most of the existing work on link prediction focuses on static approaches which use features calculated from the latest snapshot to predict future links. These existing approaches do not aggregate the long-term information of the past social interactions, such as the evolution of social networks and temporal trends of the behaviors of individual nodes, which could be important for predicting future links. For example, if we observe that an actor made increasing number of connections in the past time steps, then it is likely that the node will make more connections in the coming time step. However, if we only use the information of the latest time step, then the trends of the number of new
connections are missing. In this dissertation, we develop novel approaches to take advantage of long-term historical information of past social interactions to predict new links with higher accuracy \cite{70, 69}.

The research questions include,

- How can we characterize the evolution of node behavior from past social interactions?

- How can we use the evolution of node behavior to better predict future links?

To solve the questions, we propose a time series-based approach to characterize the temporal behavior of individual actors, and develop a novel collaborative and autoregressive time series prediction model to predict the future behavior of actors. We then develop a supervised machine learning approach incorporating the evolution of behavior for prediction of new links. The experimental results on multiple different types of large scaled social network show that our link prediction approach consistently achieves significant improvement on link prediction measured in multiple different metrics.

Our work has the following contributions,

- We systematically develop methods to aggregate past social interactions to characterize and discover the behavior evolution, and to predict future links.

- The collaborative and autoregressive time series prediction model developed in the dissertation is novel, and significantly improve the prediction accuracy.
of time series.

- The evolution-based supervised link prediction approaches significantly improve the link prediction accuracy, and sheds a light on on classification or clustering of samples with features evolving over time.

Our work has applications in a wide range of recommendation systems or ranking systems from friends suggestion at online social websites, to missing link prediction in anti-terrorism.

1.2.3 Sentiment Impact Analysis of Online Health Communities

The last important aspect of our work is the impacts of links on nodes—to investigate the change of the state of actors due to new social interactions (and new links) associated with them. For this task, we focus on investigation on the sentiment analysis and sentiment dynamics of users in social networks of online health communities (OHC) due to the following reasons:

- Many users join OHC to obtain information and seek social support. It is important to understand the dynamics of sentiment impacts of online social supports to patients and their informal caregivers.

- Previous research about OHC based on ethnographical observations, interviews, and questionnaires has reported benefits from online health communi-
ties. These approaches are too costly to be adopted for large-scale analyses about the sentiment impacts of OHC on individuals.

We therefore develop methods for large-scale sentiment impacts analysis of interactions in OHC [75].

The research questions include,

- How to do sentiment impact analysis and dynamics incorporating properties of social interactions for an OHC

- How to enable large scale sentiment impact analysis of OHC on individuals?

- What are the sentiment dynamics of members of OHC, and what factors contribute to the sentiment change of members in OHC?

To answer these questions, we develop a computational social science approach based on machine learning and text mining techniques, to study sentiment impacts and benefits in social networks. Our approach automatically estimates the sentiment of forum posts in OHC, discovers the patterns of sentiment change of OHC members, and further investigates factors that affect the sentiment change.

Our work has the following contributions,

- This is the first study of the sentiment benefits and dynamics of a large-scale health related electronic board incorporating properties of social interactions.

The methods developed in this research can be applied to other related research.
• Our results demonstrate that most of users benefit from interactions on the forum of American cancer survivors network, which is useful for patients and their caregivers to seek supports from online interactions, and is also useful for encouraging the designer and managers of the online health communities in providing better services.

• The factors associated with positive sentiment change identified in this research are important for designers of new online health communities and managers of existing online health communities, to improve online services.
Event-driven Modeling

Temporal evolution is one of the most important aspects of networks. The formation of new links, which drives the growth of network, is the key to studying the evolution of networks. A lot of work has been done on social network modeling to identify intuitive (and simple) rules of making links between nodes that lead to interesting non-random macroscopic properties exhibited in real networks, such as power-law degree distributions, hierarchical community structure, and assortative mixing, which are significant departed from those observed in random networks. In many social networks, the connections between actors are formed because they participate in the same social interaction (or social event), such as a set of scholars coauthoring a paper, and several people discussing in a thread at an online forum. However, most of the existing growth models are link-centric models and neglect the structure of social interactions. We therefore develop event-driven modeling of social networks from the perspective of social interactions/events [71, 73].
We first introduce background of event-driven social networks and related research work in social network modeling, and then characterize the growth dynamics of social networks from the perspective of event formation and evolution. An event-driven framework and event-driven models have been developed to capture the growth dynamics of social networks through modeling of the social events. In the event-driven models, we incorporate the evolution of event formation and the joint effect of attachedness and locality [72, 101]. Finally, the experiments have been carried out. The results suggest that our approach can generate realistic graphs exhibiting important macroscopic properties such as power-law degree distributions, hierarchical community structure, and assortative mixing.

2.1 Introduction

Social networks typically capture relationship between actors (e.g., authors). In many real networks, the connections are formed between actors because they participate in the same social interaction or event. For example, in collaborative networks, an social interaction or event can constitute researchers coauthoring a paper together. Therefore, an event-driven approach is a natural way to model the growth of these event-based networks. For each pair of participants in an event, a connection is formed and all the participants form a clique (a clique $K_n$ is a simple graph with $n$ nodes in which all pairs of the nodes are directly connected). Therefore, numerous cliques are observed in these networks. It is difficult for non-event
driven models (edge centric) to recreate this phenomenon. Event-driven models are also more general in that any edge centric model (modeling the formation of edges between node pairs) is equivalent to a corresponding event-driven model where exactly two actors participating in each single event. At last, in the event-drive context, we can model richer information. For example, three people, A, B, and C, participate in an event. A knows B and C, but if B and C do not know each other, then A is probably a bridge between B and C. Also, if we further consider the properties of the events and the causal relationship between events, we may garner more insights.

In an evolving social network, the set of nodes and the set of edges change over time due to new nodes joining, old nodes leaving, and new connections forming between nodes. The behavior of nodes (e.g., the selection of nodes to connect to) can vary widely and can evolve over time [72]. For example, in scientific collaboration networks, researchers usually publish papers with more senior researchers when they are junior, and more junior researchers when they are senior. In addition, the behavior of a whole network can also evolve over time. For example, a research community may grow slowly at first, but the growth rate can then increase rapidly as it attracts more members and gains prestige. Most existing network growth models we have encountered do not consider the evolution of behavior, especially from the perspective of event-driven growth.

To characterize the growth of social networks, a variety of factors have been ex-
explored, including attachedness (the degree of nodes) and locality (distance between nodes) [5, 38, 46, 101]. Section 2.2 provides an overview of the existing studies. Nevertheless, the joint effect of attachedness (degree) and locality (distance) on network growth dynamics has not been well explored.

In this chapter, we first study the evolution of event formation in real social networks and the joint effect of attachedness and locality on the selection of participants for events. We then introduce an event-driven growth model that incorporates the joint effect of attachedness and locality, as well as the evolution of event formation. Based on simulation results, we discover that our model can better characterize the growth of a large scale real network (e.g., collaborative networks in a nanotechnology community) in terms of exhibiting properties such as degree distribution, clustering coefficients, and assortative mixing.

The rest of this chapter is organized as follows: Section 2.2 introduces the background for this study and gives a brief review of related work; Section 2.3 presents some observations in real social networks that motivate our work; Section 2.4 describes the proposed event-driven driven framework for network growth models, and introduces an event-driven locality and attachedness based model that incorporates the joint effect of locality and attachedness and considers the evolution of node behavior and event formation. Section 2.5 presents quantitative analysis of some important aspects of the model settings, compares the proposed model with two other models, and shows the impact of the model on the topological properties
and correlated properties of the networks. Finally, Section 2.6 notes some potential future work and conclusions.

2.2 Related Work

In recent years, there has been intense interest in the dynamics of complex networks. Much of this work has focused on the static analysis of social networks, as well as modeling both the static topological properties and dynamic patterns associated with real social networks. The majority of the studies focused on either attachedness or locality.

Attachedness measures how well nodes are connected to other nodes in complex networks, and therefore is usually indicated by the degree of the nodes. Barabasi and Albert [5] developed a notable preferential attachment theory that specifies high degree nodes are always favored when building new connections. They proposed a model that new nodes are added to the network serially, while the probability that a new node will be linked to an existing vertex depends on the existing vertex’s degree $d_i$, $d_i/\sum_j d_j$. Using these simple rules, their model generates networks exhibiting power-law degree distributions and the “rich get richer” phenomena.

Many existing models exploit the locality explicitly or implicitly, and assume that the formation of a new connection between two nodes is related to their distance in the existing topology. Jost and Joy [39] describe a purely distance-based scheme where each new node is connected to a randomly selected node, and the
subsequent connections are related to the distance of the destination node. David-
sen et al. [19] present a referral model that connections are always formed between
two nodes that share a common neighbor. This model emulates real-world intro-
ductions, where one person introduces two acquaintances. Such a simple evolution
scheme is generally viewed as a basis for modeling the evolution of social networks.
The authors demonstrate that this simple scheme is able to reproduce nontrivial
features of social networks including small network diameter, high clustering, and
scale-free or exponential degree distribution. The scheme is also known as the
triangle-closing model [49]. The copying mechanism [46] specifies that at each
time step a new node is added to the network. The new node copies a number of
links from a “prototype” node that is selected randomly from the existing nodes
whereas choosing the remaining neighbors is random. Similar graph growth mecha-
nisms also include models that implicitly or explicitly rely on the locality heuristics
[51, 29, 43, 44, 52, 93] or specified feature similarity (correlation) between nodes
[98].

Some models also explicitly or implicitly exploit the joint effect of distance and
locality. Vazquez [90] has designed the Walking on a network scheme to simulate
the graph growth process. At every time step, a new node \( v_i \) is added and linked
to a randomly selected node \( v_j \) through a directed edge. The node \( v_i \) then mimics
a “random walk” on the network by following the edges starting from node \( v_j \) and
linking to their end points with probability \( p \). This step is repeated for those nodes
to which new connections were established, until no new target node is found. Some more recent work on this front includes Morris and Goldstein’s team-based Yule model [60] and Zhang’s DDG model [101]. The team-based Yule model maintains teams during modeling, and uses preferential attachment for within-team actor selection and random selection for generating new collaborations of actors outside the team. Hence, it adopts a binary locality measure (i.e., whether an author is within a team or outside of a team). In contrast, the DDG model uses the ratio of degree to distance to select two nodes to connect in the networks. In other words, the DDG model uses a “continuous” measure for locality.

Guimera et al. [29] propose a team assembly mechanism by investigating the interplay between “incumbents” and “newcomers” in the context of collaboration networks. The model implicitly incorporates the evolution of node behavior into modeling. Morris and Goldstein’s Yule model [60] focused on modeling coauthorship networks. However, it differs from ours in several ways. First, we explicitly present an event-driven framework to model the growth dynamics of event-driven networks while they do not. Second, we study the behavior evolution of nodes and event formation, which is incorporated into our model. In contrast, Morros and Goldstein’s model do not model the evolution of behavior. Third, our model is more efficient than the Yule model because we do not have to maintain any team structure.
2.3 Observations and Motivations

In this section, we present some observations from a nanotechnology collaboration network, *NanoSCI*, as well as the motivations for our event-based hybrid model.

*NanoSCI* is a collection of papers on nanotechnology. It offers an extensive database including 292,323 researchers and 368,511 papers that are indexed by the Science Citation Index (SCI) database spanning 1980 to 2006. In this chapter, we use data from 1980 to 2005 because our data for 2006 is only complete through August 2006.

*NanoSCI* is appealing for investigating social network growth dynamics for the following three reasons. Firstly, collaboration networks have been widely used in scientometrics and social networks study. It has been discovered that collaboration networks possess many static and dynamic properties that are similar to other social networks. In his early work on this domain, Newman studied several large collaboration networks and concluded that these networks exhibit all the general ingredients of small world networks, including short node-to-node distance and large clustering coefficient [62, 64]. Moreover, researchers have recently shown that evolving collaboration networks exhibit similar dynamic patterns as do other social networks in the growth process, such as shrinking diameters and high clustering coefficient values [6]. Secondly, *NanoSCI* offers one of the most extensive databases to date on social networks, including 292,323 researchers and 368,511 papers that are indexed by SCI (Science Citation Index) database.
(http://scientific.thomson.com/products/sci) spanning from 1980-2006. Finally, the history of nanotechnology research is very short and the literature has developed so recently that the majority of its literature is on-line. Compared to other fields even new ones such as biotechnology or super conductivity, the short history of the field combined with its fully on-line character facilitates this kind of meta-scientific study. Thus NanoSCI provides unique research opportunities for us to investigate the characteristics of the formation stage of collaboration networks.

NanoParticle, a sub-community in NanoSCI, is also studied. NanoParticle has 81,734 authors and 69,530 papers spanning 1980 to 2006.

In the following subsections, we study the formation of events and their evolution regarding the number of participants. We then study the behavior evolution of nodes with respect to their activeness. Finally, we study the effects of degree and distance on formation of new connections and events.

2.3.1 Growth Rates

In this section, we assess the growth rate in terms of the number of events (papers), nodes (authors), and edges (collaborations). Figure 2.1 shows in log-log scale the edge growth versus node growth for the NanoParticle and NanoSCI communities respectively with duplicated edges removed. It appears that the growth speed is almost linear in the log-log scale, which implies that the edge growth increase as a power law function of the node growth. The regression results show that their
Figure 2.1. The number of papers (events) and the number of collaborations (edges) increase linearly in log-log scale with the number of authors (nodes).

Growth rates are $|E(t)| = 2.3453 \times |V(t)|^{1.0238}$ and $|E(t)| = 2.5475 \times |V(t)|^{1.0409}$, respectively. $E(t)$ and $V(t)$ are the edge set and the node set in the cumulative network at time $t$ respectively. The corresponding edge densification rates (derivative on $V(t)$) for the two communities are $\Delta E(t) = 2.4011 \times |V(t)|^{-0.0238}$ and $\Delta E(t) = 2.6517 \times |V(t)|^{-0.0409}$, respectively.

The edge density is important for growth models. For example, the preferential attachment model (PA) [5] assumes that the number of edges has a linear relationship with the number of nodes. With different setting of the slopes in the linear relationship, the model shows different behaviors on some properties, e.g.,
clustering coefficients. In the Section 2.5.4, we also present a variant preferential attachment model, APA. APA uses the growth rate learned from NanoSCI instead of linear growth. APA shows a different behavior than PA.

Because we are studying event-based models, we also study the relationship between the number of paper-writing events and the number of nodes. Figure 2.1 also shows the node growth versus event growth for the NanoParticle and NanoSCI communities respectively. It appears that the growth speed is almost linear in the log-log scale, which implies that the node growth increase as a power law function of the event growth. The regression results show that their growth rates are $|D(t)| = 1.0636 \times |V(t)|^{0.9850}$ and $|D(t)| = 1.2746 \times |V(t)|^{0.9855}$ respectively, where $D(t)$ is the number of events (papers) occurred before time $t$. Thus, the corresponding node densification rates for the two communities are $\Delta D(t) = 1.0476 \times |V(t)|^{-0.0150}$ and $\Delta D(t) = 1.2561 \times |V(t)|^{-0.0145}$.

2.3.2 The Number of Participants in Events and Its Evolution

The number of participants for an event is an important factor because it determines the order of a clique (and the number of new connections) formed in a collaborative network. It has been reported that the number of secondary authors (authors other than the first author) tends to be a Poisson distribution [60]. Our observations in NanoSCI and NanoParticle verify this. In Figure 2.2, the distri-
butions of the number of secondary authors in both NanoSCI and NanoParticle match a Poisson distribution closely, although they have heavier tails than a Poisson. The average number of coauthor (plus the number of secondary author by 1) is 4.3576 and 4.3563 for NanoSCI and NanoParticle respectively.

We also notice that the average number of participants (coauthors) in events (papers) evolves over time. Figure 2.3 shows that the mean numbers of coauthors in both NanoSCI and NanoParticle increase nearly linearly in semi-log scale from around 4.0 to nearly 5.0 in the latest 10 years. It suggests that researchers are becoming more collaborative in recent years. Similar observations have also
Figure 2.3. The average number of coauthors per paper evolve linearly in semi-log scale with the number of total nodes in both NanoSCI and NanoParticle.

been reported, for example, in the collaboration network of computer science [35].

The average number of participants in an event-driven model is very important because it decides the order of the resulted cliques in the collaborative networks, and the edge density of the resulted subgraph because the edge density of a clique is $\frac{\#\text{edge}}{\#\text{node}} = \frac{n(n-1)/2}{n} = \frac{n-1}{2}$. Also, it has an effect on average separation and clustering coefficients in networks. Therefore, it is important to include the evolution of the average number of participants in the growth model.
2.3.3 The Selection of Participants for Events

In event-driven networks, the number of events a node can participate is limited due to the nodes’ available time and efforts. Also, different nodes may have different degrees of activeness due to differences of their interests. In this section, we study participant selection for events. Specifically, we study the research lifetime of researchers (details in the following Section 2.3.3.1), effect of node degree and distance between node pairs on participation in events, and the interaction between degree and distance on determining new connections.

2.3.3.1 Distribution of Lifetime of Nodes

We study the distribution of research lifetime of researchers in NanoSCI. The research lifetime of a researcher is defined as the length in years from the researcher joining the community to leaving the community. However, there is no explicit signal when a researcher leaves. Therefore, we decide that a node has left if it has been inactive for 3+ years [72]. Figure 2.4 shows the distribution of lifetimes of nodes that have been inactive for 3+ years in NanoSCI in 2005. The lifetime suggests how soon nodes evolve from active to inactive. In other words, we are using a binary measurement of the researcher activeness. In the figure, we see that about 80% researchers switch from active status to inactive in 1 year and a very small fraction of researchers can stay active in the community longer than 5 years. This makes sense because many coauthors are graduate students, they leave the
community after they graduate, and only a few of them may stay in the community as faculty or scientists. Figure 2.4 also shows the Seniority [72] distribution of all nodes and active nodes in 2005. Seniority measures the length of time the nodes have been active in the networks.

![Graph showing Seniority distribution](image)

(a)

**Figure 2.4.** Distribution of Lifetimes in years and *Seniority* distributions of nodes in *NanoSCI* in 2005.

However, it is hard to model time in years in growth models. Note that the degrees and the active time of nodes has high correlations for both *NanoSCI* and *NanoParticle*. In other words, junior researchers usually have small degrees and active senior researchers usually have high degrees. Therefore, we can use degrees of the leaving nodes to approximate their lifetimes. Figure 2.5 shows the degree
Figure 2.5. Degree distributions of all nodes in NanoSCI and of inactive nodes obey power law like dependency with an exponential cut-off $P(k) \propto k^{-\tau} e^{-k/k_c}$

distributions of inactive nodes in NanoSCI in 2005. In other words, it shows the lifetimes measured in degree. In the simulation, when a node joins the network, the model randomly samples a maximum allowed degree the node can have from the lifetime (measured in degree) distribution observed in the real data. Once the node achieves the maximum degree, it becomes inactive. Note that the maximum number of events a node can participate in is a good measure of lifetime as well, and we see that models using either maximum allowed degrees or maximum allowed events show similar results. Also notice that this is a snapshot, the real distribution
would be slightly different because the endpoints in the world are not artificially bounded.

2.3.3.2 Effects of Locality and Attachedness

The locality and attachedness has been traditionally considered as principal factors in the formation of connections. In Figure 2.6, we show the proportion $Fd(k) = M_k/N_k$, where $N_k$ denotes the number of nodes with degree equal to $k$, and $M_k$ are the nodes among them that form new edges in the next year (2002 or 2005). In Figure 2.7, we show the proportion $Fr(r) = M_r/N_r$, where $N_r$ denotes the number of node pairs at distance $r$, and $M_r$ are the pairs among them that form new edges in the next year (2002 or 2005). These demonstrate explicitly that nodes form new links proportionally to the degrees (when degrees are not extremely large) of the nodes and inversely proportionally to their topological distance. Note that in Figure 2.6, for extremely large degrees ($> 100$), the ratios for $NanoSCI(2002)$ and $NanoParticle(2002)$ are not accurate due to too few nodes with such large degrees (therefore, the corresponding points in the figure are not connected by the curves), while for $NanoSCI(2005)$ and $NanoParticle(2005)$, the ratios are relatively smaller—it suggests that nodes with extremely high degree could be less active. From Figure 2.7, we can also see that the majority of the edges are formed between node pairs at distance 2.

To distinguish the joint effect of degree and distance, we need to find a function
Figure 2.6. The percentage of nodes with certain degrees that form new connections. $F(.)$ satisfying $Pr(u, v) \propto F(d(u), d(v), r(u, v))$ such that its marginal distribution on $d(u)$ (or $d(v)$) and on $r(u, v)$ has a similar shape as that shown in Figure 2.6 and Figure 2.7, respectively, where $u$ and $v$ are two nodes in a network, $Pr(u, v)$ is the probability to form connection between $u$ and $v$. $d(.)$ is a function to get the degree of a node, $r(.)$ is a function to return the distance between the input node pair. It is generally a hard problem to discover the function. One of the ways to approximate it is to define a set of simple functions such as exponential, log, multiplication, minus etc, then use genetic algorithm (GA) \cite{59} to build formulas based on the predefined function set. Maximum likelihood estimate (MLE) can be
Figure 2.7. The percentage of node pairs with certain distance that form new connections used to choose a formula that has the best fit to data. We leave this to our future work. Instead, we study in the next subsection the interaction between degree and distance. Specifically, we study, given the degree of a node, how likely the node will connect to nodes with different hop distances away.

2.3.3.3 The Span Distance of New Edges vs. the Degree of Nodes

Span distance of a new edge is the distance between the end nodes of the edge at the moment before the formation of the edge. In Figure 2.8, we show distributions of span distances of the new edges connected to nodes with different degree
ranges. Note that repeated edges (hop distance equals to 1) are removed. From the figure, we see that nodes with all levels of seniority (degree) have significant high probabilities to connect to nodes 2 hops away. The probability to connect to nodes with a long distance tends to decrease for all types of nodes, and the trends are more significant for nodes with rich experience. For example, the most junior researchers (black star solid curve) have almost identical probabilities to make new edges spanning 3-7 hops and then have less and less probabilities to make new edges spanning more hops. For the most senior researchers (blue square solid curve), the probabilities always decrease as the span distance increases. This may suggest that senior researchers usually have stable local groups to collaborate with and may have more stable research topics as well. The factor of locality seem always to play a role when making connections for senior researchers. However, for junior researchers, the locality has much less effect especially when hop distance is equal to 3-7. Also, junior researchers have higher probabilities than senior ones to connect to nodes that are originally far away. Note that the curves corresponding to nodes with high degrees may be shorter, because they on average have smaller separation to all nodes in the networks. In the figure, Distance = Inf indicates that edges are formed between two disconnected nodes, and Distance = −1 means that the edges are connected to new added nodes.
2.4 An Event-driven Framework and A Hybrid Growth Model

We have argued that an event-driven model may be a more general and natural way to model networks, and have seen that both locality and attachedness play important roles in network dynamics. We have also studied behavior evolution. In this section, we propose an event-driven framework for modeling networks. Based on the framework, we develop an event-driven locality and attachedness based...
growth model.

To compare the networks generated by different models, it is important that the numbers of nodes and edge densities in the networks are identical. For non-event driven models, they can directly use the same edge densities defined in section 2.3.1. To compare event-driven with non-event driven models, we should also make them have the same edge densities. The following specification describes the details of the event-driven framework. It also ensures that the event-driven models can have the same edge densities defined in section 2.3.1.

1. $t \leftarrow 0$;

2. Add an event as follows:

   (a) Sample $m$ (the number of participating nodes) according to $Pr(m) \propto F_m$. $F_m$ is the distribution of the number of participants in events. It can be a Poisson distribution or the distribution observed in real networks, please refer to Section 2.3.2;

   (b) WHILE $C(m+1, 2) > |E(t)| - |E_c|$, where $|E_c|$ is the number of edges in the current network and $|E_t|$ is the number of edges at time $t$ estimated according to some predefined edge density;

      i. Repeat: add one new node and set $t \leftarrow t + 1$;

   (c) Sample $m$ nodes as participants based on some schema, for example preference attachment or hybrid schema, put all selected nodes into
Set \( P \) (the set of participants in the event);

(d) Form connections between nodes in \( Set_P \) according to some schema, for example, for a collaborative network, an edge is formed between any node pair;

3. Repeat 2) until \( t \) is equal to predefined parameter.

Based on the framework, we propose an Evolution-aware Event-driven Locality and Attachedness based Growth model (EELAG) described as follows:

Replace Step 2a with:

- Sample \( m \) according to \( Pr(m) \propto F_m(\lambda(t)) \). \( F_m \) is a stochastic Poisson distribution with its mean evolving over time or the distribution observed in real networks (See Section 2.3.2 for details);

Step 2(b)i is changed to:

- Repeat: add one new node \( n \), sample the lifetime of the node \( n_{\text{lifetime}} \) according to \( P(f) \propto F_f \), set \( \text{lifetime}_{\text{degree}}(n) \leftarrow n_{\text{lifetime}} \), and set \( t \leftarrow t+1 \). \( F_f \) is the lifetime (measured in degree) distribution predefined or learned from real data (refer to 2.3.3.1 for more information);

Step 2c is replaced with the following statements:

- With a probability \( P_\alpha \), set \( u \leftarrow \) the newly added node. Otherwise, set \( u \leftarrow \) randomly sample an active node \( u \) \((d(u) < \text{lifetime}_{\text{degree}}(u))\) according to
a preferential attachment schema: \( Pr(u) \propto \frac{d(u)+1}{\sum_v (d(v)+1)} \). \( P_a \) is set according to the percentage of papers including new researchers in the real data;

- \( Set_P \leftarrow Set_P \cup \{u\} \);

- \( WHILE d(u) == 0 AND |Set_P| < m \)

  - set \( u \leftarrow \) randomly sample an active node using preferential attachment schema;

  - \( Set_P \leftarrow Set_P \cup \{u\} \);

- \( WHILE |Set_P| < m \)

  - Based on \( d(u) \), sample a distance \( r \) according to \( Pr(r) \propto F_{rd}(r|d) \). \( F_{rd}(r|d) \) is learned from real data (more information in Section 2.3.3.3) or some approximating functions;

  - Randomly select nodes with distance \( r \);

In summary, this model first decides the number of participants based on a stochastic Poisson distribution, then samples a node as the leading node with using preferential attachment. Whenever the sampled node is not connected to the graph, a new node is sampled as leading node and the previous node is kept as a participant. Then, based on the degree of the leading node, it decides the probabilities for how far away to make new connections, and then randomly chooses nodes. This model is proposed based on the observations and motivation introduced in
Section 2.3.

2.5 Simulation and Evaluation

We set up experiments to evaluate the proposed framework and EELAG model. We use experiments to compare event-driven models with the corresponding non-event driven models, and study the effect of the average number of participants of events in event-driven models. We also compare EELAG with some other models that use attachment preference or locality preference, respectively. We set the growth rates as the same as that in NanoSCI for all models.

2.5.1 Event-driven vs. Non-event-driven

In this subsection, we study the difference on clustering coefficients in networks generated by event-driven models and corresponding non-event driven models. To focus on the comparisons between event-driven and non-event driven models and avoid including effects of other factors, a simple purely random model (PR) and its corresponding event-driven variant (EPR) are used. In the PR model, in each time step, a node is added and some connections are made between uniformly randomly selected pairs of nodes. For EPR, we only need to change Step in the event-driven framework as: uniformly randomly select $m$ nodes as the participants for the event. An identical edge density is used for both PR and EPR.

Figure 2.9 shows degree-dependent clustering coefficients, $C(k)$, that is defined
as the average local clustering coefficients ($LCC$) of all nodes with degree $k$. $C(k)$ of $NanoSCI$ can be reasonably fit by a power law $C(k) \propto k^{-\alpha}$ with $\alpha = 0.82$. This kind of power-law decay of degree dependent clustering coefficients is a signature of a hierarchical structure in the network [91]. The networks generated by $EPR$ has a higher $LCC$ than those generated by $PR$, and share the same trend with $C(k)$ observed in $NanoSCI$.

Note that for each model, we generate 50 networks and the figure shows the average $C(k)$ calculated from the networks. For all the following experiments, we do it the same way.

Figure 2.9. The average local clustering coefficients as a function of the degree of nodes (event-driven vs. non-event driven).


2.5.2 The Effect of Average Number of Participants in Event-driven Models

In the event-driven model, the numbers of participants of events are sampled from a Poisson distribution. In this subsection, we study how the average number of participants affects the clustering coefficients. Again, we use the simple purely random event-driven model, *EPR*, instead of more complex models to focus on identifying the effect of the average number of participants in the event-driven models. We use three versions of the model with the Poisson’s mean equal to 2, 4, and 8 respectively. The edge density is fixed in all three variants.

Figure 2.10 shows that with different average numbers of participants, although the trends of $C(k)$ are similar, the absolute values of $C(k)$ of the generated networks are quite different. We see that $C(k)$ is higher in networks generated by models with larger average numbers of participants, indicating more closing triangles in the network (neighbors of a node are also neighbors).

2.5.3 Behavior Evolution vs. None Behavior Evolution

In this subsection, we compare a behavior evolution version of the *EPR* model (*EEPR*) with *EPR*. One difference between these two models is that *EEPR* has activeness (lifetime) control on nodes. The other difference is that *EPR* uses a fixed average number of participants (overall mean number of coauthors in *NanoSCI*), while *EEPR* uses an evolving average number as that in *NanoSCI* (as shown
in Section 2.3.2). The edge densities used by these two models are the same.

Figure 2.11 shows EEPR has slightly better performance in modeling the degree distribution, especially for small degrees.

2.5.4 Comparisons between EELAG, APA and ADG on Simulating NanoSCI

In this section, we compare the EELAG with the APA and the ADG models. APA is a variant from Barabasi and Albert’s PA model [6]. In each time step, one
new node is added, and the number of new edges are decided from the growth rate and edge density learned from NanoSCI. We do the modifications for the purpose of fair comparisons that all models should follow the same growth rate and edge density. The edges in each step are formed between the newest node and other nodes selected according to their degrees. ADG also adds one node and a number of new edges decided by the NanoSCI’s growth rate at each step. It first randomly selects a start node $u$, and then select end nodes with probabilities $p$ according to the distance $r$ to the end nodes. Specifically, $p = 1/r$, where $r$ is the distance. For disconnected nodes, the distances $r$ are defined as a large enough value (e.g.,
20) instead of infinity. Therefore, the disconnected nodes also have chance to form edges. For both APA and ADG, networks are grown on an initial network. The initial network has 500 nodes and edges are formed by a simply random process as that in PR. The edge density is as the same as that in NanoSCI. We choose APA and ADG because they are derived from classical models and use the factor of the degree and the distance, respectively.

2.5.4.1 Degree Distribution

In this subsection, we study the degree distributions of the simulated networks created using different models, and compare them with the distribution observed in NanoSCI. Figure 2.12 suggests that EELAG recreates the similar phenomenon in NanoSCI on the proportions of nodes with small degrees, while ADG and APA do not recreate the phenomenon. For proportions at higher degrees, the networks generated by EELAG agree with NanoSCI and have very similar power-law decay.

2.5.4.2 Degree-dependent Clustering Coefficients

We study $C(k)$ of the networks simulated by different models. From Figure 2.13, we see that APA fails to simulate the trends observed in NanoSCI. Both EELAG and ADG show similar trends on $C(k)$ to NanoSCI, however, EELAG is much closer to NanoSCI.
2.5.4.3 The Average Degree of the Nearest Neighbors

Social networks are known to be assortative that the degree of connected nodes shows a positive correlation. Statistical analysis can be extended by inspecting $k_{nn}(k)$, which is the average degree of neighbors of all nodes with degree equal to $k$. For assortative (disassortative) networks, $k_{nn}(k)$ is monotonically increasing (decreasing) function of $k$. *NanoSCI* is a assortative network and $k_{nn}(k)$ of *NanoSCI* can be approximated by a power law $k_{nn}(k) \propto k^{-\beta}$, with $\beta = 0.21$. Again, *EELAG* performs better than both *APA* and *ADG*, and its $k_{nn}(k)$ is very close to that of *NanoSCI* in Figure 2.14. Note that the behavior of *APA* is differ-
Figure 2.13. The average local clustering coefficients as a function of the degree of nodes (EELAG, APA and ADG).

ent from that of Barabasi and Albert’s PA [6] reported by Newman [63] because APA uses an evolving growth rate observed in NanoSCI instead of a constant linear growth.

2.6 Conclusions

Previous research on social network modeling were link-centric that they only model the formation of individual links. However, in many social networks, connections are formed between actors because they are involved in the same event.
Figure 2.14. The average degree of neighbors as a function of degree (EELAG, APA and ADG).

For these networks, it is natural, general and powerful to use event-driven models to characterize their growth dynamics. Therefore, we proposed an event-driven framework to facilitate the creation of event-driven growth models. We also studied the evolution of node activeness and event formation in social networks, and exploited the effect of both locality and attachedness on the formation of new edges. We found that the average number of participants of events evolve over time in NanoSCI, and the effects of distance and degree on selection of participants of events also change over time.

The analysis led us to propose an hybrid model based on an event-driven
framework that considers the evolution of event formation and the joint effects of distance and degree. Based on metrics that are informative in characterizing the network structure, such as degree distribution, degree-dependent clustering coefficients ($C_k$) and degree-dependent average degree of neighbors ($k_{nn}(k)$). Our experiments showed that the networks generated by our event-driven hybrid model exhibit structures similar to real networks, while other non-event driven models fail to recreate these structures.

In summary, our work has the following contributions,

- The event-driven idea to model social networks from the perspective of social interactions to study social networks. It is also a natural and general way to model the growth of these event-based networks, and has the potentials to model richer information than existing link-centric modeling. For example, event-driven models can model the event leader and follower, and other more informative features of events

- Event-driven framework developed in the chapter makes it easy to develop event-driven models for social networks

- The networks generated by event-driven models exhibit similar structural properties as real networks, which suggests that event-driven modeling may capture important evolutionary dynamics of social networks

Future work arising from our results so far includes: carrying out experiments
on more real networks, further studies of important factors of connection formation and their joint effect, modeling events with richer information, and incorporating more aspects of behavior evolution.
The formation of new links is one of the keys for studying the evolution of networks. Link prediction discovers the patterns of the formation of links from the historic evolution of networks, and uses them to predict future links between nodes. Most of the existing approaches do not aggregate the long-term information of the past social interactions, such as the evolution of social networks and temporal trends of behaviors of individual nodes, which could be important for predicting future links. For example, if we observe that an actor made increasingly more connections in the last four time steps, then it is likely that the node will continue to make more connections in the future. However, if we only use the information of the latest time step, such trend information is lost. To address this problem, we develop approaches to take advantage of long-term historical information of network
evolution to predict new links with higher accuracy [69, 70].

We first introduce the background of link prediction problems and related research work, and then propose a time series-based approach to characterize the behavior evolution of individual actors, and develop a novel collaborative and autoregressive time series prediction model to predict the future behavior of actors. We then develop a time series-based supervised machine learning approach for prediction of new links. Finally, we conduct experiments on multiple different types of large-scale social networks. The results show that our behavior evolution based link prediction approach consistently achieves significant improvement on link prediction accuracy.

3.1 Link Prediction

In the study of social network evolution, one of the central tasks is link prediction. It aims at inferring new links among existing nodes for the future. Link prediction has many real world applications, including recommending new items in various networks (e.g., friends in Facebook, co-authors in CiteSeerX, products in eBay), monitoring and preventing criminal activities in a criminal network, predicting the next webpage users will visit, and complementing missing links in automatic web data crawlers, and so on.

Traditionally, link prediction is performed by computing topological metrics in a static graph, which represents a snapshot of the social network. For instance, a
metric could be the node degree, since a node with a large numbers of existing connection is likely to form a new link based on the preferential attachment model [5]. Other popular static features include the Katz coefficient [42], common neighbors and the Adamic/Adar number [1], and so on. However, all of them are time-agnostic and not designed for characterizing temporal information of networks. It is therefore imperative to answer the following question: could temporal features exploit additional new link patterns compliment to those captured by the existing topological metrics of a network?

In this chapter, we study the link prediction problem from the perspective of time, and seek to analyze the functions of temporal features. Temporal changes in social networks are usually difficult to model, because the dynamic interactions between nodes in a social network can be so numerous and mostly hidden. A few existing work has attempted to improve link prediction performance with temporal factors [67, 36]. However, most of temporal features applied by the related work are first-order statistics such as moving average (average value of a property, e.g., degree, over time) and recency (how much time elapsed since a node makes the last connection) [67], and number of activities over time [36]. Given the fact that social networks evolve over long periods, it is hard to convert their findings into statements about long-term graph trends.

To capture the long-term evolutionary characteristics of social networks for link prediction, we study node behavior evolution patterns individually and pair-
wisely. In our early work, node behavior evolution patterns are observed to be useful in modeling dynamic social networks [72, 73, 71]. For example, in a scientific collaboration network, a student researcher tends to work with senior researchers; as she establishes up in career, her collaboration preference becomes more diverse. This suggests that node behaviors are different in various time periods.

We quantify each node behavior evolution with a collection of correlated time series features. For example, one time series feature such as the vector of the average degree of collaborators at different time is useful to model the evolved activeness property of this student researcher’s behaviors. To the best of our knowledge, little work has been done on analyzing long-term graph trends [50, 53, 54]. Moreover, our work is the first to use time series features in the problem of link prediction.

It is challenging to model node behavior evolution with time series features effectively. First, given one type of node behavior evolution, there could be multiple time series features to quantify it, each of which only captures one perspective of the evolution. In this chapter, we systematically define four types of time series features to comprehend node behavior evolution. They are: 1) simple first-order statistics of time series, 2) local patterns in time series, 3) 1-step ahead predictions of time series and 4) interplay features that compute the joint likelihood of two nodes at the next time step. The first two types of features are straightforward, but the prediction and interplay features of time series rely on the accuracy of the selected time
series prediction methods. We develop a hybrid temporal model that leverages the autoregressive information of individual node time series and information of other correlated node time series to accurately predict the future node behaviors. Second, the relations among various features are non-linear and hard to be formalized with a few rules. In the chapter, the Random Forest strategy is used to combine various features. We find that, the combination of time series features and static topological features can produce the better link predication results than static features only, which demonstrate the great potential of integrated methods.

We conduct experimental study on three different scaled datasets, including a scientific collaboration network, a phone network and the Facebook network. The results consistently show that time series features are extremely valuable to link prediction, especially when there are not many static topological features available for link prediction (e.g., the distance of two nodes is large or the average degree of these two nodes is moderately small). In general, our evolution-aware link prediction approach outperforms those evolution-agnostic baselines by more than 10% in the measure of Area Under Curve ($AUC$).

The rest of the chapter is organized as follows. The related work is presented in Section 3.2. We model nodes with time series in Section 3.3. In Section 3.4, the hybrid temporal model is proposed to compute the prediction features of time series. In Section 3.5.1, the interplay features of time series are computed and our evolution-aware supervised link prediction approach is introduced. Experiments
are conducted in Section 3.6. Finally, Section 3.7 concludes the chapter.

3.2 Related Work

There is a long line research on link prediction. In this section, we review existing work on both time-agnostic and time-aware approaches.

**Time-agnostic approaches.** The past studies have focused on static approaches for link prediction. Many approaches are available, including single topological feature-based model and joint models. Among the single-feature approaches, observing high fraction of edges close triangles in many large scale online social networks, Leskovec et al. [49] proposed a triangle-closing models for modeling link formation in network growth. Similarly, locality-based methods include Jaccard’s coefficient (the number of the common neighbors divided by total neighbors) [53], and Adamic/Adar (weighting the common neighbors by the rarity of relationship between a neighbor with other nodes) [1]. Product of the degree of two nodes also used for link prediction based on the phenomenon of Preferential attachedment [5]. Path-based methods include using distance, maxflow, Katz (weighted sum of the number of paths with different lengths between two nodes, shorter paths are flavored) [42], rooted PageRank and its variants such as PropFlow (the probability that random walk starting at one node ends at the other node in the a pair in \( l \) or fewer steps) [54] and SimRank (the similarity between two nodes is iteratively calculated as the sum of their neighbors’
similarity) [37]. Liben-Nowell and Kleinberg [53] aggregated and compared most of the methods. Hierarchical structure has also been investigated for predicting links [53, 16]. These methods help us understand the growth dynamics in social network, however, since only one feature are used, their performance is usually not good [53].

Methods based on joint information of multiple features are investigated and supervised models are used. These approaches models the link prediction problem as maximal likelihood problem or classification problem. For a node pair, depending on whether they will form a link between in future, can be defined as negative or positive samples. Link prediction is to use learning methods to maximize likelihood of data or to achieve high accuracy on classifying the samples. Taskar et al. [87] based on relational Markov networks model to maximize the joint likelihood of a graph (link existence or not and related attributes of the node and links). Hasan et al. [31] trained and compared several different types of classier such as decision tree, k nearest neighbors (KNN), and support vector machine (SVM) for coauthorship link prediction. and used the trained model to predict the test data. Kashima et al. [41] proposed a probabilistic models for network evolution which models the existence of links between node pairs based on copy – and – paste mechanism, and based on the model to predict new links. Wang et al. [92] used logistic regression to leverage local probabilistic features proposed in their paper with topological features for link prediction. Recently, Leskovec et al. [50] also
investigated on predicting link types, opposition or friendship, in social networks using logistic regression classifier. Lichtenwalter et al. [54] investigated a Random Forest based approach and found significant improvement on performance measured in Area under Curve (AUC). However, all these approaches are time agnostic.

**Time-aware approaches.** In recent years, temporal factors in social network analysis have become more prominent. Leskovec et al. [49] studied the lifetime of nodes and new edge arrival time interval for better modeling of social link formation. Qiu et al. [72, 73, 71] showed that the preference of a node in choosing nodes to connect evolves over time with the node’s properties change, e.g., seniority, and proposed an evolution-aware model for modeling networks. Tylenda et al. [89] showed that weighting the existing edges based on the creating time of the edges (more recent edges have higher weights) improved the performance for link prediction. Potgieter et al. [67] showed that using temporal factors, such as overtime average (average value of a property, e.g., degree, over time) and recency (how much time elapsed since a node makes the last connection), could improve performance of link prediction. This work suggests the value of temporal factors, however, using only first-order statistics is not able to capture the trends of the temporal factors.

Huang and Lin [36] proposed a hybrid model for predicting recurrent links between node pairs. It extracted the frequency of occurrences of recurrent link
over time between a node pair, and used univariate time series model to predict the number of occurrences of recurrent links between the node pair in future time steps. The occurrence frequency is defined on existing edges, and therefore, it works only for predicting recurrent links and not able to predict new links between two nodes with no edges exist.

Our approach is significantly different from previous work. First, we define temporal features on individual nodes and on any pair of nodes and thus can be used for any types of link prediction. Second, we model nodes and node pairs using time series, and propose novel approaches to characterize node behavior trends for better link prediction.

3.3 Model Node Behavior Evolution with Time Series Features

In this section, we define time series features from the node behavior evolution for the problem of link prediction.

3.3.1 Temporal Sequence of Node Behavior

For each node, a sequence of event sets can be recorded to portray the node behavior evolution. For example, in a scientific collaborative network from 2000 to 2005, if we know a researcher $v$ has published 1, 2, 4, 1, 5 and 7 papers respectively,
the corresponding sequence of event sets is

\[ E_v = \{ \{e_1\}, \{e_{2-3}\}, \{e_{4-7}\}, \{e_8\}, \{e_{9-13}\}, \{e_{14-20}\} \}, \]

where each \( e_i \) indicates a publishing event.

If we know all details about each \( e_i \) in \( E_v \) (e.g., paper content, venue, and co-authors), \( E_v \) provides a good basis for comprehending the corresponding node behavior evolution. However, a realistic definition of event replies on the completeness of dataset and the event detection algorithms, which are out of the scope of this chapter.

Alternatively, we seek to extract a bunch of numeric time series features from the event set sequence, by taking advantage of two circumstances: 1) it is easy to extract many numeric time series features from most of existing temporal networks (compare to other sophisticated features like event topic); 2) the numeric time series features can be accurately quantified.

Our assumption is that, as long as we can extract enough numeric time series features (patterns) and each feature describes one aspect of the event set sequence, the corresponding node behavior evolution can then be well captured.

Following the same toy example of \( E_v \), we can easily obtain a few related time series features, as listed in Table 3.1. \( TS_1 \) is the node activeness feature (number of publications per year). \( TS_2 \) is the average degree of new collaborators of \( v \). \( TS_3 \) counts the number of papers first-authored by \( v \).
Table 3.1. Example time series features for $E_v$

<table>
<thead>
<tr>
<th>ID</th>
<th>Definition</th>
<th>Time Series</th>
</tr>
</thead>
<tbody>
<tr>
<td>TS1</td>
<td>Activeness</td>
<td>(1, 2, 4, 1, 5, 7)</td>
</tr>
<tr>
<td>TS2</td>
<td>New neighbors’ average degree</td>
<td>(5.2, 9.3, 1.3, 1.5, 2.3, 2.1, 3.2)</td>
</tr>
<tr>
<td>TS3</td>
<td># of first-authored papers</td>
<td>(1, 2, 2, 1, 0, 1)</td>
</tr>
<tr>
<td>TS4</td>
<td>Ratio of TS3 to TS1</td>
<td>(1, 1, 0.5, 1, 0, 0.14)</td>
</tr>
</tbody>
</table>

A time series feature can also be constructed on the combinations of existing time series features. For instance, $TS4$ is the ratio of $TS3$ to $TS1$. These four features collectively capture how the activeness of researcher $v$ evolves in the scientific collaborative network.

### 3.3.2 Time Series Features

In this section, to comprehend various aspects of the node behavior evolution all sidedly, we systematically define four types of time series features: Simple, Local Pattern, Prediction and Interplay. Table 3.2 lists 35 features of these four types plus 10 traditional static topological features for a pair of nodes $v_i$ and $v_j$. In static topological features, $\Gamma(v)$ is the set of neighbors of node $v$, $\beta$ is set to be 5 for Katz, maxflow and propflow simply for a fast computation, the exponential damping coefficient $l$ for Katz is 0.005, and $\text{paths}^{<i>}_{(v_i,v_j)}$ represents the set of all length-$i$ paths from $v_i$ to $v_j$. In the following, we explain each type of temporal features in details.
3.3.2.1 Simple Features

- The first type of time series features are simple temporal features used by related work [67, 36]. Features 11-14 in Table 3.2 fit this category. Simple features are time-dependent. But they only compute the first-order statistics of a time series. The temporal trend of the time series is lost. Therefore, we define the Local Pattern features to capture the temporal trend in the following.

3.3.2.2 Local Pattern

- The second type of features based on the local patterns of a time series. We focus on local pattern instead of global pattern because the latter is too complicated to define in a time series. Given a time series $T$ and the time interval $[s, b]$, we define two types of local mini-patterns: increasing and decreasing, as below.

  - Increasing: $T_i < T_j$, where $s \leq i < j \leq b$;
  - Decreasing: $T_i > T_j$, where $s \leq i < j \leq b$;

Each local mini-pattern on a time series $T$ can be modeled as a tuple $(g, s, b)$ which means the sub time series $T[s...b]$ has the trend of $g$. $s$ and $b$ are selected such that $T[s-1...b]$ and $T[s...b+1]$ break the trend of $g$, $g$ is the type of mini-pattern (Increasing or Decreasing).

We extend the approach proposed by M. Kadous [40] to calculate our Local Pattern features. It contains three steps: coding, extracting, and clustering to extract the features from a time series.
In the step of coding, a time series is converted to a sequence $S$ of “I” and “D”—if $T_{i+1}$ is larger than $T_i$, $i \geq 0$, then the $i$th element of the sequence $S_i$ is “I”, otherwise $S_i$ is “D”.

In the step of extracting, local mini-patterns are extracted from the sequence $S$.

A local mini-pattern can be viewed as a point $(s, b - s + 1)$ on a plate $g$. M. Kadous suggested that if two patterns of type $g$ have small distance on the plate, then they are similar to each other. Therefore, in the third step, clustering is conducted on the local mini-patterns of the same type and a binary feature $F_C$ is created for each cluster $C$: if the time series $T$ has a local pattern belonging to cluster $C$, then $F_C(T) = 1$; otherwise $F_C(T) = 0$.

Suppose there are $M$ clusters for each type of pattern, then $3M$ binary features are created in total for $T$. In our data, we set $M = 10$.

The process is illustrated in Figure 3.1.

3.3.2.3 Prediction

This is the third time series feature. As we state before, the shortcoming of Local Pattern features is that they are not able to capture the global temporal trend of a time series. However, the global temporal trend of a time series is usually complex and can not be portrayed by a single pattern. Therefore, we use the 1-step ahead prediction value of a time series as the third type of feature to partially describe
the global temporal trend. The intuition is: if two (normalized) time series have
the same global temporal trend, they have the same 1-step ahead prediction value\textsuperscript{1}.

Features 25-34 in Table 3.2 belong to this category, where each static topological
feature has a prediction value (feature). Moreover, we find that the product of two
Prediction features $\text{degree}_{i\text{-pred}}$ and $\text{degree}_{j\text{-pred}}$ is useful in link prediction, so
we introduce an extra product Prediction feature (Feature 35 in Table 3.2) to
this category. The suffix $\text{pred}$ is used to tag Prediction features. Computing the
Prediction features involves thorough understanding of time series, which will be
discussed in Section 3.4 separately.

\textsuperscript{1}Note that the converse is not always true. That also motivates us to consider other type of features simultaneously.
3.3.2.4 Interplay

The fourth type of time series features. The problem of link prediction is to predict the linkage between two nodes. We thus especially define one kind of temporal features named Interplay, which compute the joint likelihood of two nodes to be connected in the future based on to which degree the two nodes meet each other’s expectation on neighborhood. The computation details are given in Section 3.5.1. Features 36-42 in Table 3.2 are interplay defined on node pair features such as distance and katz, and Feature 43-45 are defined on node features such as recency, active and degree. The suffix ipr is used to tag Interplay features.

3.4 Hybrid Temporal Model for Prediction of Node Behavior

In this section, we compute the Prediction time series features defined in Section 3.3.2 using time series prediction algorithms adaptive to social networks. There are many existing time series prediction models such as autoregressive model (AR) and autoregressive integrated moving average model (ARIMA) [84]. However, not many are designed for the problem of link prediction in social networks. In social networks, many nodes only stay in the network shortly and the corresponding time series are short. So, the traditional AR or ARIMA models that prefer long
Table 3.2. List of features for two nodes $v_i$ and $v_j$.

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$degree_i$</td>
<td>$</td>
</tr>
<tr>
<td>2</td>
<td>$degree_j$</td>
<td>$</td>
</tr>
<tr>
<td>3</td>
<td>$PA$</td>
<td>$</td>
</tr>
<tr>
<td>4</td>
<td>$distance$</td>
<td>Length of the shortest path</td>
</tr>
<tr>
<td>5</td>
<td>$comN$</td>
<td>Common neighbors: $</td>
</tr>
<tr>
<td>6</td>
<td>$jaccard$</td>
<td>$</td>
</tr>
<tr>
<td>7</td>
<td>$AA$</td>
<td>$Adamic/Adar: \sum_{v_z \in \Gamma(v_i) \cap \Gamma(v_j)} \frac{1}{</td>
</tr>
<tr>
<td>8</td>
<td>$katz(\beta, l)$</td>
<td>$\sum_{i=1}^{\infty} \beta^l \cdot</td>
</tr>
<tr>
<td>9</td>
<td>$maxflow_\beta$</td>
<td>The max flow from $v_i$ to $v_j$</td>
</tr>
<tr>
<td>10</td>
<td>$propflow_\beta$</td>
<td>The probability of random walk starting at $v_i$ arrives at $v_j$ in $\beta$ steps (transition probability based on edge weight)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>$recency_i$</td>
<td>Time since the last activity of $v_i$</td>
</tr>
<tr>
<td>12</td>
<td>$recency_j$</td>
<td>and $v_j$ respectively</td>
</tr>
<tr>
<td>13</td>
<td>$active_i$</td>
<td>The number of new edges received by $v_i$</td>
</tr>
<tr>
<td>14</td>
<td>$active_j$</td>
<td>and $v_j$ respectively at last time step</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15-24</td>
<td>${x}_lp$</td>
<td>Local pattern of the time series of feature $x$: Feature 1-10.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>25-34</td>
<td>${x}_pred$</td>
<td>Prediction of future value of feature $x$ based on its corresponding time series where $x$: Feature 1-10.</td>
</tr>
<tr>
<td>35</td>
<td>$predDP_{pred}$</td>
<td>$(degree_i\cdot pred \cdot degree_{j, pred})$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Interplay Likelihood (IPR) as Features (cf. Section 3.5.1)</td>
</tr>
<tr>
<td>36-42</td>
<td>${x}_ipr$</td>
<td>IPR based on feature $x$: Feature 4-10</td>
</tr>
<tr>
<td>43</td>
<td>$r_{ipr}$</td>
<td>IPR based on $recency_i$ and $recency_j$</td>
</tr>
<tr>
<td>44</td>
<td>$a_{ipr}$</td>
<td>IPR based on $active_i$ and $active_j$</td>
</tr>
<tr>
<td>45</td>
<td>$degree_{ipr}$</td>
<td>IPR based on $degree_i$ and $degree_j$</td>
</tr>
</tbody>
</table>

Time series may suffer. Moreover, in social networks, many nodes often share similar temporal patterns due to the influence flow. It is thus natural to integrate collaborative filtering strategies or the wisdom of the population to the time series...
prediction algorithms. In the following, we shall discuss these details.

3.4.1 Collaborative Time Series Prediction

People with similar properties often share similar behavior patterns. A most practical example is that, in recommender systems, collaborative filtering is often used to do recommendation for users based on their top similar users’ behavior [83]. In light of this idea, we try to predict future values of time series of nodes based on other nodes with similar time series. In the following discussion, we take time series of overtime activeness of nodes as examples. Let us use $T$ and $l_T$ to denote a time series and its length respectively. For every pair of time series $(T_a, T_b)$, we calculate their distance $Dist_j(T_a, T_b)$ based on their first $j$ elements, where $1 \leq j \leq \min(l_{T_a}, l_{T_b})$. Denote the distance $Dist_j(T_a, T_b)$ for all pairs of time series at length $j$ as a vector $D_j$, and the vector of corresponding difference at $j + 1$ as $F_{j+1}$. We then measure the dependency of time series distance and difference of future values by calculating correlation between $D_j$ and $F_{j+1}$.

$$\text{Corr}(D_j, F_{j+1}) = \frac{\text{Cov}(D_j, F_{j+1})}{\sqrt{\text{Var}(D_j)\text{Var}(F_{j+1})}}$$  \hspace{1cm} (3.1)

where $\text{Cov}$ indicates the covariance of two vectors and $\text{Var}$ is the variance. Figure 3.2 shows the results on three different large scale datasets NanoSCI, Phone and Facebook. For details of these datasets, please refer to Section 3.6.1. We can observe from the results that the correlation is positive and significantly larger
than 0.0 for all length of time series on all datasets, and the average correlations on NanoSCI, Phone and Facebook are 0.516, 0.506 and 0.468, respectively. This indicates that if the distance between two time series increases, then the difference of their future values also increases. In other words, if two different time series are similar (small distance), then they tend to have similar values at $j + 1$.

![Figure 3.2](image_url)

**Figure 3.2.** The correlation of similarity of time series at length $j$ and their value difference at position of $j + 1$. The mean correlation of NanoSCI=0.516, Phone=0.506 and Facebook=0.468

Based on these observations, we can derive an important conclusion, that similar time series can be used for predict future behavior trends for social network users. Therefore, we propose a collaborative time series prediction model CoTSP.
To predict the future value of a time series $T$ with $l_T$ elements, we find top-$k$ most similar time series of $T$ based on their first $l_T$ elements. We then use the average of the $(l_T+1)th$ elements of these $k$ time series as the prediction (we also try using median instead of mean but there is no significant difference in performance). We empirically set $k = 50$ in our experiment.

To better understand the prediction given by CoTSP, e.g., standard error and prediction intervals, we need to characterize the probability distribution of the prediction. Some approximations are required. First, we assume that the one-step ahead future values of time series of two different nodes are independent, i.e., the numbers of new edges connected two different nodes are independent. This may be biased if these two nodes happen to be connected in the next time step, but that case is rare in our experiment. Second, we assume that the one-step ahead future values of two similar time series follow an identical distribution, such as Gaussian processes. Under this assumption, if two time series have very similar values in the past time, they are very likely to have the same mean and variance in Gaussian distribution. In other words, they are likely to have the identical distribution. Our experimental results on three datasets have provided strong support for these assumptions. With them, we have the following theorem.

**Theorem 1.** Use $y_c$ to indicate the prediction of the $t$-th elements of a time series $T$ by CoTSP, and denote the $t$-th elements of the $i$-th top similar time series of $T$ as $X_i$ ($1 \leq i \leq k$). Suppose $X_i$ ($1 \leq i \leq k$) are independent and identically
distributed (iid), then the prediction $y_c$ follows an asymptotic normal distribution.

**Proof.** The prediction given by CoTSP is

$$y_c = \bar{X} = \frac{1}{k} \sum_{i=1}^{k} X_i.$$  \hspace{1cm} (3.2)

According to the Central Limit Theorem (CLT) [13], $y_c$ has an asymptotic normal distribution ($k \to \infty$) with mean $\mu = y_c$ and variance $\sigma^2 = S^2$, where

$$S^2 = \frac{1}{k-1} \sum_{i=1}^{k} (X_i - \bar{X})^2. \hspace{1cm} (3.3)$$

Many distance/similarity metrics for time series have been proposed in literature, and *Euclidean distance* is one of the generally accepted metrics [20, 82]. We use a weighted *Euclidean distance* measure in our experiments: for time series with $n$ elements, the weight of the difference of the $(j)$-th elements is $\rho$ times of the weights of the difference of $(j+1)$-th, $1 \leq j \leq (n-1)$. If $0 < \rho < 1$, latest elements gain more weights. We empirically set $\rho = 0.9$ for all datasets in our experiments.

### 3.4.2 Hybrid Time Series Prediction Model

CoTSP uses information of time series of other nodes for prediction, where as AR and ARIMA are suitable to capture the autoregressive information for prediction.
Z. Huang et al. [36] used ARIMA to predict the number of repeated interactions between two nodes in social networks. In our work, we also adopt ARIMA to incorporate the autoregressive aspect of node behavior trends in prediction. Because ARIMA and CoTSP capture different types of information for time series prediction, a natural idea is to combine them to achieve better performance. We thus propose a hybrid model—HybridTSP. Because the length of input time series has significant effects on prediction performance for both ARIMA and CoTSP, we include time series length as a covariate for the model combination. Let $T$ and $l_T$ denote a time series and its length respectively, and let the prediction of ARIMA, CoTSP and HybridTSP be $y_a$, $y_c$ and $y_h$, respectively, the standard error $se$ of the prediction of ARIMA and CoTSP as $s_a$ and $s_c$ (Eq. 3.3) respectively. We define our hybrid model in the following equation.

$$y_h(T) = \lambda_0(l_T) + \lambda_1(l_T)y_a(T) + \lambda_2(l_T)y_c(T) + \lambda_3(l_T)y_s(T) \quad (3.4)$$

where $y_s(T) = \frac{s_a^2(T)y_a(T)+s_c^2(T)y_c(T)}{s_a^2(T)+s_c^2(T)}$ is the simple linear combination of two predictions based on their standard error $se$: the prediction with a smaller $se$ has a higher weight. Notice that $s_a$ and $s_c$ depend on time series $T$. To measure the parameters $\lambda_i(l_T)$ ($i = 0, 1, 2, 3$), we extracted subsequences of time series in our collection by taking the first $i$ elements, and use a linear regression model to estimate the parameters, where the dependent variable is the value of the $(i + 1)th$ element, and the predictor variables are intercept, the predictions of ARIMA,
CoTSP and their simple combination. To check the performance, we split collected time series of activeness into training and testing sets for NanoSCI, Phone and Facebook. Training set is used to train the parameters in HybridTSP. Figure 3.3 shows the boxplot of standard deviation of prediction by ARIMA, CoTSP and HybridTSP on the testing data. On all dataset, HybridTSP has better performance than ARIMA. We also conduct student t-tests [13] to compare CoTSP and HybridTSP with ARIMA, and the results are shown in Table 3.3.

<table>
<thead>
<tr>
<th>Hypothesis</th>
<th>NanoSCI</th>
<th>Phone</th>
<th>Facebook</th>
</tr>
</thead>
<tbody>
<tr>
<td>( m_c = m_a )</td>
<td>0.6931</td>
<td>0.9875</td>
<td>0.0006</td>
</tr>
<tr>
<td>( m_h = m_a )</td>
<td>0.0046</td>
<td>0.0071</td>
<td>0.0000</td>
</tr>
</tbody>
</table>

Here, the mean standard error of CoTSP, ARIMA and HybridTSP are denoted by \( m_c \), \( m_a \), and \( m_h \) respectively. The p-values on testing the hypothesis \( m_h = m_a \) are smaller than 0.05, which suggests that the hypothesis should be rejected. We can then conclude that HybridTSP is significantly better than ARIMA on all datasets. The performance difference between CoTSP and ARIMA, however, is dataset-dependent: CoTSP has better performance than ARIMA on NanoSCI and comparative performance on Phone, but on Facebook, the performance of CoTSP is not as good as ARIMA. This is due to the nature of the datasets. Recall in Figure 3.2, we have the mean correlations on the dataset have the relation that NanoSCI > Phone > Facebook. Also notice that even in Facebook, CoTSP still provides very useful information for prediction because HybridTSP achieves
Figure 3.3. The standard error of prediction by different models still significantly better results than ARIMA.

We now consider the distribution of the prediction given by HybridTSP for better understanding the prediction, e.g., standard error and prediction intervals. We will also propose an interplay likelihood model for link prediction based on the distribution in Section 3.5.1. For CoTSP, Theorem 1 states that the prediction follows an asymptotic normal distribution. For ARIMA, if the underlying process of a time series is Gaussian process, then the prediction has a normal distribution.
In many cases, the time series can be generated from Gaussian process. For those time series that are more natural to model as other processes, we approximate them by using normal distribution. For example, counting process is often modeled as Poisson process, we can approximate Poisson distribution using normal distribution: a Gaussian distribution with both mean and variance equal to $\lambda$ is a good approximation to a Poisson distribution with rate $\lambda$ when $\lambda$ not too small.

With the approximations, we have the following Theorem 2.

**Theorem 2.** Suppose the underlying process of a time series is Gaussian process, then the prediction $y_h$ given by HybridTSP follows an asymptotic normal distribution.

**Proof.** The prediction given by HybridTSP is the linear combination of the prediction of ARIMA and CoTSP that both follow asymptotic normal distribution. According to the property of normal distribution that the linear combination of normal distributed variables is still normal distributed, the prediction of HybridTSP also follows an asymptotic normal distribution. Assume that the mean of predictions of ARIMA and CoTSP for time series $T$ are $\hat{\mu}_a(T)$ and $\hat{\mu}_c(T)$, and the variance are $s^2_a(T)$ and $s^2_c(T)$ respectively, then

$$y_h(T) \sim AN(\hat{\mu}_h(T), s^2_h(T))$$

(3.5)
where
\[
\hat{\mu}_h = \lambda_0(l_T) + \lambda_1(l_T)\hat{\mu}_a(T) + \lambda_2(l_T)\hat{\mu}_c(T) + \lambda_3(l_T)\hat{\mu}_{ac}(T)
\] (3.6)

where \(\hat{\mu}_{ac}(T) = \left(\sum_{l}^{T} \frac{s^2_{a}(T)}{s^2_{a}(T)+s^2_{c}(T)}\hat{\mu}_a(T) + \frac{s^2_{c}(T)}{s^2_{a}(T)+s^2_{c}(T)}\hat{\mu}_c(T)\right)\), and for \(s^2_h(T)\), if we assume that the prediction by ARIMA and CoTSP are independent, we have
\[
s^2_h(T) = (\lambda_1(l_T) + \lambda_3(l_T)(\frac{s^2_{a}(T)}{s^2_{a}(T)+s^2_{c}(T)})^2)s^2_a(T) + \\
(\lambda_2(l_T) + \lambda_3(l_T)(\frac{s^2_{a}(T)}{s^2_{a}(T)+s^2_{c}(T)})^2)s^2_c(T)
\] (3.7)

Remark: the assumption that the prediction by ARIMA and CoTSP are independent is reasonable because ARIMA is calculated from \(T\) while CoTSP is calculated from other time series (although the other series are top similar to \(T\)).

### 3.5 Behavior Evolution Based Link Prediction Approach

Many link prediction approaches have been proposed including unsupervised and supervised methods. We focus on supervised approaches instead of unsupervised methods because supervised methods have many advantages over unsupervised approaches. First, supervised methods are extensible. Unsupervised approaches often used formula predefined based on domain knowledge, while supervised approaches learn the formula automatically. Therefore, it is easy and natural to
combine multiple features. It can also easily handle cases with multiple output. This is required in situations to predict types of links instead of just link existing or not. In addition, as pointed out by R. Lichtenwalter [54], supervised methods can be put into ensemble models to reduce variance and further improve performance. Second, supervised approaches are adaptive and can fit different networks automatically while unsupervised methods are usually domain specific. For example, R. Lichtenwalter et al. [54] pointed out that unsupervised methods based on preferential attachment may be suitable for coauthorship networks, but may fail on mobile communication networks. Third, supervised methods are able to automatically capture the interdependency relationships between features. Therefore, we develop our model based on a supervised learning framework.

Figure 3.4 illustrates the general process of supervised link prediction. First, node pairs, both positive (a new link formed between the node pair in the next time step) and negative ((no new link formed between the node pair in the next time step), are sampled from temporal networks. Second, a part of the node pairs are used for training models (training set), all the rest are used as testing set for evaluation. Third, features are extracted from both training set. Third, models are trained (feature selection and model selection may be included). Fourth, features are extracted from testing samples. Fifth, applying the model to the testing samples and evaluate the performance by comparing the prediction results to ground truth of testing samples.
Different from previous supervised link prediction approaches which calculate features in the last snapshot, our approach computes the features from all previous snapshots and try to characterize the behavior evolution of nodes in making new links. Figure 3.5 uses an example to show the difference of our approaches from previous non-evolution based methods. For each node pair sampled from snapshot at time $T$, features are extracted from the last snapshot at time $T - 1$ in previous approaches. In contrast, in our approach, features are extracted from last $k$ ($k > 1$) snapshots before $T$. Therefore, each sample in previous approach is a vector and each element is an individual value while each sample in our approach is a vector of time series.

Although the computational cost of our approach is $k$ times of previous approaches, the behavior evolution of nodes in making new links can be captured, and more meaningful features as suggested in Table 3.2 can be extracted to improve link prediction performance, including the Interplay feature that is introduced in details in the following Section 3.5.1.
3.5.1 The Interplay Likelihood of Two Nodes

In Section 3.3.2, we have computed the Prediction time series features and discussed their generative processes (asymptotic normal distribution). Now, we are able to compute a new type of time series features named Interplay, which uses the prediction probability of the normal distribution to calculate the joint likelihood of two nodes at the next time step.

For properties defined on nodes, namely, degree, recency, and active, we use Eq. 3.8 to calculate the interplay likelihood.

\[ IPr_{F^n}(v_i, v_j) = f(F^n_i; \hat{\mu}_i, \sigma_i^2) * f(F^n_j; \hat{\mu}_j, \sigma_j^2) \]  (3.8)
where \(v_i\) and \(v_j\) are two nodes in a sample, \(F^n\) is a feature on node, and \(F^n_j\) and \(F^n_i\) are the values of \(v_i\) and \(v_j\) respectively, (\(\mu_i, s_i\)) is the prediction (\(\mu_i\) is the mean and \(s_i\) is the standard error) of the feature of \(v_i\)'s future neighbors based on temporal observations on the property of nodes to whom \(v_i\) connected to in the past, and (\(\mu_j, s_j\)) is the prediction for \(v_j\). \(f(\cdot)\) is the probability density function of a normal distribution based on Theorem 2. For example, \(f(d_j; \mu_i, s_i^2)\) is the probability density of \(\text{Normal}(\mu_i, s_i^2)\) at \(d_j\).

For features defined on node pairs, such as distance and katz etc., the idea is the same but have a slightly different formula shown in Eq. 3.9.

\[
IPr_{F^e}(v_i, v_j) = f(F^e_{ij}; \hat{\mu}_i, s_i^2) \ast f(F^e_{ij}; \hat{\mu}_j, s_j^2) \tag{3.9}
\]

where \(v_i\) and \(v_j\) are two nodes in a sample, \(F^e\) is a feature on the node pair and \(F^e_{ij}\) is the value of the link \((v_i, v_j)\). (\(\hat{\mu}_i, s_i\)) is the prediction (\(\hat{\mu}_i\) is the mean and \(s_i\) is the standard error) of \(F^e\) of \((v_i, v_x^+), v_x^+\) is the future neighbors, based on temporal observations on the property of \((v_i, v_x^-), v_x^-\) is the nodes \(v_i\) connected to in the past, and (\(\hat{\mu}_j, s_j\)) is the prediction for \(v_j\).

Based on these equations, if \(v_j\) matches \(v_i\)'s expectation on her ideal future neighbors and vice versa, then \(v_i\) and \(v_j\) have high likelihood to connect to each other in the future time step. The performance of these interplay likelihood features is shown in Section 3.6.5 in details.
3.5.2 EvPred: Evolution-based Link Prediction Approach

Until now we have finished the discussion of all time series features. Any feature (including the static topological feature) can be used to train an individual supervised learning model. We combine all individual models as a joint model to achieve better performance. Formally, use $X$ to indicate a sample set and $M_i$ to indicate individual models, we need to find a function $g(\cdot)$ to build a joint model $M_{\text{joint}}$:

$$M_{\text{joint}}(X) = g(M_1(X), M_2(X), \ldots, M_m(X))$$  \hspace{1cm} (3.10)$$

A simple approach is to use linear regression as we do on building HybridTSP model. However, there are a large number of sub-models and features in the situation and linear combination is not able to the complicated relationship between them due to the disadvantages of linear regression. First, it can only capture linear relationship between the input sub-models. Second, it cannot adaptively detect the interactions between the models unless we explicitly include them. If the number of sub-model is large, then which interaction terms to include is a challenging problem because $2^m$ possible interaction terms in total. The interaction terms are important, because in our case, different models or features may be based on different type of properties to capture different perspectives of behavior evolution, and to consider interactions between them helps to combine the information from different sources for making decisions.
In this chapter, we adopt random forest in our algorithm because it can capture complicated relationship between a large number of features or output of sub-models, as well as easily adapt to high order interactions of them. It has been shown to have good performance in link prediction problems [54]. The other reason we use random forest is that we can directly compare our model to models in [54]. We remark that random forest is not crucial, other learning approaches such as boosting and SVM can be easily plugged in.

Random forest contains classification and regression trees (CART) [32]. Each tree can take advantage of input features to build classification or regression rules. With temporal features, e.g., \( active_{pred} \) which indicates how many new connections a node will make in the next time step, the tree may build a rule,

\[
active < 1 \rightarrow Class \text{ Negative}(91.5%)
\]

This rule suggests that if one node in a training sample has a low predicted activity, then the likelihood of the node pair in this sample is small.

\( CART \) can capture the interaction between the features adaptively. A path from root nodes to leaf nodes can contain rules based on different features. An example is,

\[
active_{a, \text{pred}} \geq 3.0 \rightarrow
\]

\[
distance_{ab} < 3.0 \rightarrow
\]

\[
AA_{pred} > 0.2 \rightarrow
\]

\( Class \text{ Positive} \) (40.5%)
This rule captures the interaction between four features, $active_a$, $distance_{ab}$ and $AA_{red}$. The maximum allowed height of tree is an important parameter for capturing the interactions. If the height is too small, then the tree may not be able to capture strong interactions between features. However, if the height is too large, then the tree may be overfitted. Cross-validation is techniques for choosing good parameters.

### 3.6 Evaluation

In this section, we evaluate how node behavior evolution features can improve the link prediction performance in three different scaled real social networks.

#### 3.6.1 Datasets

We tested the following real social networks.

- **NanoSCI**: The scientific collaboration network in nanotechnology research community from 1980 to 2006 (each year is a time period). The nodes are researchers and the edges represent co-authorship. There are 292,323 researchers and each has 10.15 unique co-authors (new links) on average.

- **Phone**: The cell phone communication network in an European country from 09/2007 to 03/2008 (each week is a time period). The nodes are phone users and the edges are phone calls. There are 24,986 phone users and each
has 61.03 callees (new links) on average.

- **Facebook**: The wall-to-wall post relationship on Facebook from 09/2006 to 01/2009 (each month is a time period). The nodes are Facebook users and the edges are wall posts. There are 66,842 users and each has 12.8 friends (new links) on average.

### 3.6.2 Experimental Setup

#### 3.6.2.1 Preprocessing

Recurrent links were removed for the purpose of new link prediction. Let $G_t$ be the snapshot of a network up to time $t$. Given a network with duration $n$, \{G_1, \ldots, G_{0.6n}\} are the training data and the rest snapshots are the testing data for all models. Positive training links are new links happen in \{G_1, \ldots, G_{0.6n}\}, and negative training links are pair of nodes that have not connected to each other till time 0.6$n$. Positive/negative testing links are extracted similarly. For temporal models, if the new link happens in $G_t$ ($1 \leq t \leq n$), its time series features are constructed from the snapshot sequences \{G_1, \ldots, G_{t-1}\}. For static baselines, if the new links happens at time $t$, its static features are constructed from $G_{t-1}$.

Note that social networks are sparse such that positive links only occupy a small portion of all pair of nodes. In order to make the training and testing data balanced, random under-sampling of negative samples is used in many previous work [54] so that the number of negative samples are comparative to that of positive samples.
However, this kind of random sampling is controversial. First, the training and testing data are not representative for the underlying social networks. Second, the testing accuracy may be higher than expected because the under-sampling may make the problem easier. For example, the distances in negative samples (node pairs) randomly selected are usually significantly larger than the distances in positive samples randomly selected because many of links are formed between nodes less separated. Figure 3.6 shows the distribution of distance of negative (random sampled) and positive samples generated from our datasets. We find that setting a threshold on distance (predict a sample as positive if the distance is smaller then the threshold, otherwise negative) achieves a very high accuracies around 70-80% on the three datesets.

In this chapter, we developed two other ways for evaluation. One is to develop classifiers to handle highly imbalanced data, therefore, we do not have to guarantee the data balance. To generate the imbalanced data that better represents the networks, we randomly and uniformly sample both positive and negative samples from networks until we achieve 1,000 positive samples. The other is to take the link prediction problem as a node ranking problem: given a node $v$, each non-neighboring node (a node which is not currently connected to $v$) has a probability to connect to $v$ in the future; the node ranking is to rank those nodes by the probability and use metrics from search science for evaluation. To generate the node ranking data, we first randomly sample query nodes $q$, then for each of them,
Figure 3.6. Distribution of distance of node pairs in samples
to find the nodes $V = v_1, v_2, ..., v_n$ exactly 2 hops away. The node ranking problem
is to sort $v_i$, $1 \leq i \leq n$, by their probability of connecting to $q$ in the next time
step. We only consider nodes 2 hops away is to limit the size of $V$ due to our
computational resource but without missing too many meaningful samples. As
shown in 3.6, nodes which are too far away from $q$ are too unlikely to connect to
$q$, and less meaningful to include them.

For convenience of comparing our results with results from previous research,
we also conduct random under-sampling for negative samples to get more balanced data and generate experimental results. To study the impact of imbalance of positive/negative samples, we used all positive data but randomly sampled negative data by controlling the positive/negative ratio as 1/3, 1/2, and 2/3 (same to training and testing at one setting) respectively in five subsets and compare the performance on them. Moreover, negative samples have three sources. Give a pair of nodes without a link between them till time $t$, if both nodes link to the third-party nodes, this sample is in active-active ($A-A$) class; if one node has connections to other nodes, this sample belongs to active-inactive ($A-I$) type; if both nodes are isolated, we group this negative sample into the inactive-inactive ($I-I$) type. Differentiating the negative sample sources is important because samples of various types have rather various topological characteristics (e.g., data in $A-A$ class are closer to positive sample compared to others). As a simple evaluation, we fixed the positive/negative ratio as 1/2 and then created three types of data where each only contains one type of negative samples.

3.6.2.2 Baselines

We compared our evolution-aware approach EvPred with the following baselines.

- **RF**: An evolution-agnostic random forest based approach proposed by Lichtenthaler et al. [54]. It only includes the static topological features 1-10 in Table 3.2.
• **RF-Recency.** In [67], the temporal feature—recency is used to improve link prediction performance. For a fair comparison, we replace the logistic regression in the model with random forest for better performance. We denote the model as RF-Recency. This model uses the static topological features 1-10 as well as features 11 and 12 in Table 3.2.

• **RF-Active.** Huang et al. [36] showed that the time series of link recurrence is useful for the problem of recurrent link prediction. In their work, only the activeness features (number of interactions between a pair of nodes) were used. We simulated a similar model named RF-Active that includes the static topological features 1-10 plus features 13 and 14 in Table 3.2, but for the problem of new link prediction.

• **Distance.** A simple approach use only one feature distance. As we stated previously, using randomly selection to make the numbers of negative and positive samples comparable in training and testing data, can make the link prediction problem easy (and thus testing accuracy may be high) than expected. We therefore include this simple approach as a bottom line.

### 3.6.2.3 Metrics

We have three types of evaluation: the first one is on balanced data generated by under-sampling of negative samples, the second is imbalanced data, and the third is node ranking data.
We define the following two metrics to evaluate the performance on balanced data.

- **Precision.** The ratio of # true (either positive or negative) testing links predicted by any algorithm to # of all testing links.

- **AUC.** The area under receiver operating characteristic (ROC) curve. $ROC$ curve plots true positive rate versus false positive rate by varying the decision threshold on probability estimation from 0 to 1. A high AUC of a model suggests that the model can generate good relative scores (classification probability) for different samples.

Because the metrics defined for balanced data are not suitable for evaluation on highly imbalanced data, we define the following metrics for imbalanced data.

- **Precision.** The ratio between the true positive and the sum of both true positive and false positive.

- **Recall.** The ratio between the true positive and the sum of both true positive and false negative.

- **F-measure.** $F - measure = \frac{2 \times Precision \times Recall}{Precision + Recall}$.

For node ranking data, we use Normalized Discounted Cumulative Gain (NDCG) [17] as the metric.
3.6.3 Link Prediction Performance

3.6.3.1 Balanced Data: Impact of Positive/Negative Ratio

We first experiment on balanced data generated by under-sampling of negative samples. By varying positive/negative ratio to be 1/3, 1/2, and 2/3 respectively, we compare four methods (RF, RF-Recency, RF-Active and EvPred) on the performance of new link prediction. Figure 3.7 shows the results on three datasets. Considering the negative data were randomly sampled, each point in the figure averages over 5 trials to avoid the side effect of data randomness. From Figure 3.7, we conclude the following findings.

- Our evolution-aware model EvPred consistently outperforms the others on all datasets and positive/negative ratio settings, indicating that time series features that describe the node behavior evolution patterns are extremely useful to the problem of new link prediction. For example, on average, EvPred outperforms three baselines by more than 15%, 10% and 6% on NanoSCI, Phone and Facebook respectively in both AUC and precision. The outperformance difference on various datasets is also consistent to the degree of time series correlation of datasets (cf. Figure 1, the time series correlation in three data is: NanoSCI > Phone > Facebook).

- Three baselines (RF, RF-Recency, RF-Active) perform comparably for new link prediction. The two time-aware models (RF-Recency and RF-Active)
are slightly better than RF (1%-5%), indicating that the time is an important factor in link prediction but simple temporal statistics like Recency and Activeness are not enough to capture the long-term trend patterns hidden in the complex social network evolution. Distance performs as a bottom line.

- The measure AUC is insensitive to the positive/negative ratio, which is consistent to the findings in [54]. However, the measure Precision has a concave (especially for EvPred) as positive/negative ratio increases.

3.6.3.2 Balanced Data: Impact of Negative Sample Sources

We continue work on the balanced data to study the impact of negative samples. With a fixed positive/negative ratio 0.5, we compare performance on three kinds of data settings, each of which only contains one type of negative samples (A − A, A − I and I − I). Figure 3.8 shows the results of all 4 models on three datasets under this setting. The major results in Figure 3.8 are consistent to previous findings observed in Figure 3.7. We originally expect the sources of negative samples would alter the link prediction performance. Surprisingly, both AUC and Precision of all 4 models remain nearly constant across various negative data sources. We achieved similar conclusions by further varying the positive/negative ratio. That is to say, new link prediction is possibly a one-class classification problem and insensitive to the selection of negative samples. We leave the solid verification to the future work.
Figure 3.7. Performance (AUC and Precision) of different models on data trials containing different percentage of positive samples. (a) and (b) are performance on NanoSCI in AUC and precision respectively, (c) and (d) are performance on Phone, and (e) and (f) are performance on Facebook.

3.6.3.3 Imbalanced Data

We conduct experiments on imbalanced data of three networks. In testing, the number of positive samples from all the three networks is 1,000, and the number
Figure 3.8. Performance (AUC and Precision) of different models on data trials containing different types of negative samples. (a) and (b) are performance on NanoSCI in AUC and precision respectively, (c) and (d) are performance on Phone, and (e) and (f) are performance on Facebook.

of negative samples is larger than 100,000. The training data is twice as large as testing.

To handle the highly imbalanced, we developed the balanced random forest
Table 3.4 shows the results for Phone. Our approach significantly outperform all the baseline models on all three metrics of Precision, Recall and F-measure.

<table>
<thead>
<tr>
<th>Method</th>
<th>Precision</th>
<th>Recall</th>
<th>F-measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>EvPred</td>
<td>0.152</td>
<td>0.824</td>
<td>0.257</td>
</tr>
<tr>
<td>RF-Recency</td>
<td>0.062</td>
<td>0.737</td>
<td>0.116</td>
</tr>
<tr>
<td>RF-Active</td>
<td>0.080</td>
<td>0.713</td>
<td>0.144</td>
</tr>
<tr>
<td>RF</td>
<td>0.061</td>
<td>0.697</td>
<td>0.113</td>
</tr>
</tbody>
</table>

Table 3.5 shows the results for Facebook. Again, our approach significantly outperform all the baseline models on all three metrics.

<table>
<thead>
<tr>
<th>Method</th>
<th>Precision</th>
<th>Recall</th>
<th>F-measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>EvPred</td>
<td>0.138</td>
<td>0.863</td>
<td>0.238</td>
</tr>
<tr>
<td>RF-Recency</td>
<td>0.095</td>
<td>0.770</td>
<td>0.169</td>
</tr>
<tr>
<td>RF-Active</td>
<td>0.102</td>
<td>0.785</td>
<td>0.180</td>
</tr>
<tr>
<td>RF</td>
<td>0.088</td>
<td>0.789</td>
<td>0.158</td>
</tr>
</tbody>
</table>

Table 3.6 shows the results for Facebook. Again, our approach significantly outperform all the baseline models on all three metrics.

<table>
<thead>
<tr>
<th>Method</th>
<th>Precision</th>
<th>Recall</th>
<th>F-measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>EvPred</td>
<td>0.125</td>
<td>0.992</td>
<td>0.221</td>
</tr>
<tr>
<td>RF-Recency</td>
<td>0.049</td>
<td>0.638</td>
<td>0.091</td>
</tr>
<tr>
<td>RF-Active</td>
<td>0.051</td>
<td>0.631</td>
<td>0.094</td>
</tr>
<tr>
<td>RF</td>
<td>0.042</td>
<td>0.598</td>
<td>0.079</td>
</tr>
</tbody>
</table>
3.6.3.4 Node Ranking Data

For node ranking data, in testing, we have 2,500 positive node pairs, and 35,000 negative pairs for NanoSCI; 2,500 positive node pairs, and 25,000 negative pairs for Phone, and 2,500 positive pairs and 40,000 negative ones. The training data is twice as large as testing.

NDCG is used as the metric and Figure 3.9 shows the results. On all three networks, our model has significant better NDCG at top 1, 3, 5 and 10 than all baselines. This suggests our model can always output much better ranking of nodes given a query node.

3.6.4 Serendipity of Evolution-aware Model on Predicted New Links Based on Balanced Data

In this section, we study which kind of new links can be only detected by our evolution-aware model EvPred. The study is based on the balanced data.

Since the other three baselines are comparable, we simply choose RF as the compared baseline. We define four types of predicted new links: \textit{Com} is the testing new links that could be correctly predicted by both EvPred and RF, \textit{Bonus} is those only correctly classified by EvPred, \textit{Miss} is those only correctly classified by RF, and \textit{Hard} is those incorrectly classified by both models. Table 3.7 shows the distribution of these four types of predicted new links on three datasets (with positive/negative ratio 0.5 and regardless of the negative data sources). The por-
tions of Bonus links are so significant across three datasets (26.2% in NanoSCI, 12.4% in Phone and 9.1% in Facebook) that it is worthy to study the properties of these “bonus” links. Very few links can only be predicted by RF, simply because RF used a subset of features of EvPred.

Table 3.7. Distribution of four types of predictions.

<table>
<thead>
<tr>
<th>Data</th>
<th>Com</th>
<th>Bonus</th>
<th>Miss</th>
<th>Hard</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>NanoSCI</em></td>
<td>0.697</td>
<td>0.262</td>
<td>0.019</td>
<td>0.022</td>
</tr>
<tr>
<td><em>Phone</em></td>
<td>0.780</td>
<td>0.124</td>
<td>0.021</td>
<td>0.075</td>
</tr>
<tr>
<td><em>Facebook</em></td>
<td>0.824</td>
<td>0.091</td>
<td>0.000</td>
<td>0.085</td>
</tr>
</tbody>
</table>
3.6.4.1 Distance of Two Nodes before Linkage

Intuitively, if two nodes are far away from each other (i.e., > 3) before their initial connection, the local topological features like commonneighbor, jaccard and adamicadar etc. will downgrade to 0 and thus are not able to characterize the linkage. So, the new link between these two nodes is hard to be predicted by a traditional model that only considers the topological features. Figure 3.10 shows boxplots of distance distribution of Com and Bonus links before their linkages on three datasets. Interestingly, the average distance of Bonus links is larger than that of Com links on all three datasets (6.295 vs. 5.374 in NanoSCI, 5.916 vs. 4.818 in Phone and 5.416 vs. 4.759 in Facebook). In general, Bonus links have a lower variance in distance, indicating that the majority of Bonus links only predicted by EvPred Model has a long distance.

We further carry out the student t-test to test the hypothesis of the equality of their mean distance. The p-values on all three datasets are 0 (reject the corresponding hypothesis completely). Therefore, the statement that the distance of Bonus links is larger than the distance of Com links on all three datasets is significantly strong. In other words, an evolution-aware model like EvPred is significantly better than an evolution-agnostic model like RF in predicting the linkage of nodes with long distance.
3.6.4.2 Mean Degree of Two Nodes before Linkage

We study another important property called mean degree of two nodes in Bonus and Com links respectively before their linkage. We split the mean degree into consecutive bins $[0, 1)$, $[1,2)$, $[2, 3)$, ..., and count the number of links with mean degree fallen into each bin. Accordingly, for each type of links (Bonus and Com), we have a distribution of number of links over mean degree bins, as shown in Figure 3.11. The distributions on all datasets are convex, indicating that when the mean degree of two nodes is moderate before their linkage, the new link between them is more likely predicted by our evolution-aware model (EvPred) only. When
the mean degree is small (both nodes are isolated from most of other nodes),
no model can effectively predict the corresponding new link since neither static
topological features nor temporal features contain enough linkage patterns. When
the mean degree is large (both nodes are hubs in the network), there are rich static
topological features so that RF is competent for link prediction.

![Figure 3.11. The distribution of mean degrees of Bonus and Common on three datasets](image)

**3.6.5 The Importance of Features Based on Balanced Data**

Finally, we study the importance of all features listed in Table 3.2. This study is
also based on balanced data. For a fair comparison, we only feed one single feature
into our EvPred Model at a time. This methodology is widely adopted by related work to understand the feature importance [53]. According to the results, we have the following observations.

- Most of top features are temporal ones. For example, for NanoSCI, top 10 features are all temporal. For Phone and Facebook, there are 6 temporal features on top 10.

- The important temporal features are jaccard_i, r_i and katz_i, because they appear on top 10 for all datasets. The best non-temporal features for Phone and Facebook are propflow, katz and distance.

- Many Interplay (i) features are in the top lists for all three dataset, suggesting that Interplay features captures importance information for link prediction.

- As the time series correlation gets stronger (Facebook < Phone < NanoSCI), static topological features play less important role and temporal features tend to dominate the top list. For example, propflow, katz and distance are ranked top 3 in Phone and Facebook but out of top 10 in NanoSCI, and

- The importance of features vary on different datasets. The physical nature of data determines the importance of features.
Table 3.8. Performance of individual feature (measured by $AUC$)

<table>
<thead>
<tr>
<th>NanoSCI</th>
<th>Feature</th>
<th>Feature</th>
<th>AUC</th>
<th>Feature</th>
<th>AUC</th>
<th>Feature</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>jaccard</td>
<td>ipr</td>
<td>0.847</td>
<td>katz</td>
<td>0.855</td>
<td>propflow</td>
<td>0.895</td>
</tr>
<tr>
<td>1</td>
<td>r</td>
<td>ipr</td>
<td>0.841</td>
<td>propflow</td>
<td>0.855</td>
<td>katz</td>
<td>0.889</td>
</tr>
<tr>
<td>2</td>
<td>katz</td>
<td>ipr</td>
<td>0.839</td>
<td>distance</td>
<td>0.854</td>
<td>distance</td>
<td>0.887</td>
</tr>
<tr>
<td>3</td>
<td>a</td>
<td>ipr</td>
<td>0.839</td>
<td>predDp_pred</td>
<td>0.817</td>
<td>maxflow</td>
<td>0.839</td>
</tr>
<tr>
<td>4</td>
<td>adamicadar</td>
<td>ipr</td>
<td>0.838</td>
<td>maxflow</td>
<td>0.81</td>
<td>propflow_pred</td>
<td>0.836</td>
</tr>
<tr>
<td>5</td>
<td>comN</td>
<td>ipr</td>
<td>0.798</td>
<td>propflow_ipr</td>
<td>0.797</td>
<td>katz_ipr</td>
<td>0.807</td>
</tr>
<tr>
<td>6</td>
<td>maxflow_ipr</td>
<td>0.790</td>
<td>jaccard_ipr</td>
<td>0.796</td>
<td>jaccard_ipr</td>
<td>0.804</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>propflow_ipr</td>
<td>0.787</td>
<td>katz_ipr</td>
<td>0.782</td>
<td>maxflow_ipr</td>
<td>0.78</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>predDp_pred</td>
<td>0.786</td>
<td>propflow_pred</td>
<td>0.767</td>
<td>adamicadar_ipr</td>
<td>0.767</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>degree_ipr</td>
<td>0.778</td>
<td>r_ipr</td>
<td>0.746</td>
<td>r_ipr</td>
<td>0.765</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>dist_ipr</td>
<td>0.769</td>
<td>adamicadar</td>
<td>0.743</td>
<td>predDp_pred</td>
<td>0.764</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>PA</td>
<td>0.733</td>
<td>comN</td>
<td>0.743</td>
<td>PA</td>
<td>0.763</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>distance_pred</td>
<td>0.733</td>
<td>jaccard</td>
<td>0.743</td>
<td>distance_pred</td>
<td>0.759</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>katz</td>
<td>0.721</td>
<td>jaccard_pred</td>
<td>0.742</td>
<td>PA_pred</td>
<td>0.749</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>katz_pred</td>
<td>0.721</td>
<td>maxflow_ipr</td>
<td>0.741</td>
<td>comN_ipr</td>
<td>0.744</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>propflow</td>
<td>0.710</td>
<td>PA</td>
<td>0.737</td>
<td>maxflow_lp</td>
<td>0.731</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>propflow_pred</td>
<td>0.710</td>
<td>distance_pred</td>
<td>0.729</td>
<td>katz_lp</td>
<td>0.728</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>distance</td>
<td>0.703</td>
<td>adamicadar_ipr</td>
<td>0.717</td>
<td>degree1</td>
<td>0.716</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>PA_pred</td>
<td>0.703</td>
<td>katz_lp</td>
<td>0.713</td>
<td>adamicadar</td>
<td>0.712</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>active1_pred</td>
<td>0.694</td>
<td>comN_ipr</td>
<td>0.702</td>
<td>comN</td>
<td>0.711</td>
<td></td>
</tr>
</tbody>
</table>

3.7 Conclusions

In this chapter, we study the link prediction problem from the perspective of time by aggregating past social interactions to model node behavior evolution with time series features, and apply the node behavior evolution to link prediction based on a supervised learning framework. Four types of time series features, Simple, Local Pattern, Prediction and Interplay, are defined to comprehend the role of node dynamic behaviors in the formalization of new links. More specifically, a hybrid temporal model that leverages the autoregressive information of individual time series and information of other correlated time series is designed to compute the
Prediction features. Then, our evolution-aware model named EvPred is proposed to effectively combine time series features and static topological features using Random Forest learning framework. We test our ideas on three real social networks and the results consistently show that after considering evolution features of nodes, the performance of link prediction becomes better.

In summary, our work has the following contributions,

• The first work of this kind to develop methods to aggregate past social interactions to characterize and discover the behavior evolution, and to predict future links.

• The collaborative and autoregressive time series prediction model developed in the dissertation is novel and significantly improve the prediction accuracy

• The time series-based supervised link prediction approaches significantly improve the link prediction accuracy

• The work sheds a light on classification or clustering of samples with features evolving over time

Our work has applications in a wide range of recommendation systems or ranking systems from product suggestion in e-business, to missing link prediction in anti-terrorism.
Sentiment Impact Analysis of Online Health Communities

Unlike previous chapters that investigate on the patterns of formation of new links, this chapter studies the effects of the occurrence of new links/interactions. More specifically, the chapter focuses on the sentiment impacts and sentiment dynamics of users in social networks of online health communities (OHC). The motivations for research on OHC and sentiment dynamics are as follows:

- Many users join online health communities (OHC) to obtain information and seek social support, therefore, it is important to understand the dynamics of sentiment impacts of online social interactions to patients and their informal caregivers.

- Previous research about OHC based on ethnographical observations, inter-
views, and questionnaires has reported benefits from online health communities. However, these approaches are too costly to be adopted for large-scale analyses about the sentiment impacts of OHC on individuals.

We first introduce the background of online health communities especially online cancer survivor networks and related research on emotional analysis of users in these health related online social networks. We then develop a computational social science approach based on machine learning and text mining techniques, to do sentiment analysis of the posts published by users. Based on the sentiment analysis, we study the sentiment impacts and benefits of users. We then further investigate factors that affect the sentiment change. Our results on the American Cancer Society’s Cancer Survivors Network (CSN) (http://csn.cancer.org) demonstrate that an estimated 75%–85% of CSN forum users change their sentiment in a positive direction through online interactions with other people. Two useful features, Name and Slang, not previously used in sentiment analysis have been shown to be useful in identifying positive sentiment in posts.

4.1 Introduction

Cancer accounted for nearly one in every four deaths in the United States [86], and approximately 13% or 7.4 million deaths worldwide in 2007. According to the World Health Organization [95], cancer is estimated to cause 12 million deaths worldwide in 2030. Today, about 12 million Americans either have recently diag-
nosed with cancer or identify themselves as a cancer survivor [86]. Many survivors and their family or friend caregivers experience not only physical effects but also emotional effects such as stress, anxiety, and depression [7] from cancer and cancer treatments.

Each year, many people turn to the Internet to satisfy their health-related needs [22] for information and support. A 2010 study by the Pew Research Center found that 83% of American adult Internet users utilize the Internet for health-related purposes [102], with more than 25% of these users [103] seeking social support through joining and participating in an online health community (OHC). Unlike websites that only offer slowly changing medical or other health-related information, an OHC typically includes features such as discussion boards, chat rooms, etc. where users can interact with each other. Support and information from people with similar cancers or problems is very valuable because cancer experiences are unique, and family members, friends and cancer care providers often do not understand the problems [68]. A cancer OHC that includes both survivors and their caregivers offers a way to share experiences about their cancer and cancer treatment, seek solutions to daily living issues, and in general, support one another [3] in ways that are not often possible with other close family, friends or even health care providers.

Benefits to cancer survivors who have participated in an OHC are reported in the literature. OHC participation increases social support [21][81], reduces levels
of stress, depression, and psychological trauma [8][96], and helps participants be more optimistic about the course of their life with cancer [81]. The support received from other OHC members help cancer patients better cope with their disease and improve their lives both physically and mentally [21][58]. Caregivers for cancer patients typically receive similar benefits.

Most previous research is based on data collected and analyzed using traditional social science methods, such as ethnographical observations, interviews, questionnaires, surveys, and statistical testing. These data collection methods pose three challenges. First, the scale of the data is limited due to the fact that direct observation and interview takes a lot of time. Obtaining data from the thousands of users in even a moderately-sized OHC is expensive, resource-intensive and impractical. Second, the sample used is typically biased. For example, active members who are happy and satisfied with an OHC are more likely to respond to researchers’ questionnaires or surveys than those who are inactive or unhappy with the community. In addition, much of the data are dependent on personal recall of past events. A person’s recall of past events and emotions is often flawed or incomplete. Remembering emotional reactions to other OHC member responses to one or more topics or questions posted sometime in the past is unlikely to be accurate [79][55]. Third, these methods often have coarse temporal granularity. Tracking real-time emotional dynamics in direct association with OHC participation is extremely difficult with these methods.
In this research, a computational social science [47] approach to the study of how cancer survivors and caregivers benefit from participations in an OHC is presented. Computing technologies enable recording and analysis of the asynchronous and distributed social interactions in an OHC, making large amounts of data available for analysis. Computation also makes it possible to analyze the content and structure of online interactions captured in these large-scale data. Using a popular OHC for cancer survivors as a case study, discussion forum interactions for a 10-year time period are analyzed to provide insight into how cancer survivors’ and caregivers’ participation in the OHC affect emotions on a larger scale and in a systematic fashion. In what follows, the OHC forum used is described and basic use statistics provided. Next, machine learning techniques [66] to analyze sentiment of forum discussions are illustrated. A new approach to analysis of sentiment dynamics and identification of contributing factors are presented followed by conclusions and discussion of future research needs.

4.2 Case Study Data: American Cancer Society Cancer Survivors Network

The American Cancer Society, a national community-based volunteer health organization, designed and maintains its Cancer Survivors Network (CSN) (http://csn.cancer.org) as a dynamic online community for cancer patients, cancer survivors and their fami-
ilies and friends. It is considered a safe and welcoming place for them to support one another and share their cancer or caregiving experiences, feelings and practical tips for dealing with many of the issues encountered during cancer treatment and subsequent survival. Launched in July of 2000, it currently has more than 137,000 member participants.

The conceptual framework for CSN is based on the work of Irvin D. Yalom, M.D., a respected group work theoretician and practitioner [99]. By design, CSN provides peer support and psychosocial intervention services, utilizing group dynamics to facilitate therapeutic factors such as the instillation of hope, universality, catharsis, existentialism, altruism, interpersonal learning and group cohesiveness. The most commonly used CSN feature for group interaction is the forum consisting of 38 discussion boards of which 25 are cancer-specific. The breast cancer and colorectal cancer boards are the two most active discussion boards. Non-cancer-specific boards are devoted to topics such as humor, caregiving, emotional support, and spirituality.

Forum posts from July 2000 to October 2010 comprising 48,779 threads and more than 468,000 posts from 27,173 participants were downloaded into a dataset. Participants-respondents are identified by code only and this approach passed IRB review. A nationwide ACS marketing campaign featuring CSN conducted in the spring of 2008, resulted in a greater than 300% growth in CSN membership with the result that about 70% of all forum posts occur after this campaign. Forum
post rates are relatively stable from May 2009 to October 2010 at an average of 16,550 posts per month. Participant total post counts, including initial posts and all replies, follows a power-law distribution (see Figure 4.1), demonstrating what is typical for most on-line communities, that most participants publish few posts while a few highly active participants publish a large number of posts.

Forum posts are organized in threaded discussions (threads) comprised of initial posts and replies. The number of replies in a thread also approximately follows the power-law distribution (see Figure 4.2). The life span of a thread, defined as the time between the initial post and last reply is distributed as shown in Figure 4.3. This right skewed distribution estimates mean thread life span of 1,725 hours (or about 72 days) but median of only 58 hours (2.4 days) with 72% of threads having
life spans shorter than seven days. The correlation between the number of replies and the life span of a thread is weak with Pearson correlation coefficient of 0.04. This suggests that long threads having many replies do not necessarily have long life span. Basic statistics of the forum data used in this analysis are presented in Table 4.1.

Table 4.1. Summary statistics for CSN forum thread data.

<table>
<thead>
<tr>
<th></th>
<th>Mean</th>
<th>Median</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of posts by a participant</td>
<td>17.25</td>
<td>2</td>
<td>5,607</td>
</tr>
<tr>
<td>Number of replies per thread</td>
<td>8.7</td>
<td>6</td>
<td>442</td>
</tr>
<tr>
<td>Life span of a thread</td>
<td>1,725 hours</td>
<td>58 hours</td>
<td>87,846 hours</td>
</tr>
<tr>
<td></td>
<td>(71.9 days)</td>
<td>(2.4 days)</td>
<td>(10 years)</td>
</tr>
</tbody>
</table>
Figure 4.3. Distribution of the time span of threads in CSN forums.

4.3 Sentiment Analysis

The sentiment of a participant’s post, reflecting their emotion at the time of posting, is not directly observable, and hence must be modeled as a latent variable. The sentiment of a participant who initiates a thread (the originator) can be analyzed at both the beginning and the end of a thread. In this way, it is possible to determine whether the support provided by participants who responded to the thread (the respondents) is able to change the sentiment of the thread originator. Manually labeling sentiment levels for tens of thousands of posts is not feasible. Automatic identification methods previously developed for sentiment analysis [56][66] are used instead. These methods use machine learning and text mining [9] to identify indi-
idual opinions in text and use them to illustrate and analyze the dynamics of the underlying sentiment. Use of this methodological approach to sentiment analysis is growing, with recent applications made to recommender systems, business and government intelligence, and computational politics [56][66].

Most computational sentiment analysis is focused on selecting indicative lexical features that allow classification of texts into positive or negative [66] sentiment classes. This task is made easy when sentiment-labeled data is readily available, such as with product marketing data[66] where consumers are directly asked about positive (desirable) or negative (undesirable) aspects of a product. However, what is considered desirable in one domain may not be so in another domain. Consider for example the use of the word “positive”. In the cancer domain, one can have a “positive” diagnostic test which might indicate the presence of cancer (an undesirable sentiment) or might indicate that the cancer is responding to therapy (a desirable sentiment). This implies that training data must, by design, be unique to the domain being analyzed and take into account context when classifying text sentiment. At the same time, no one classification approach will work in all situations. The classification model must also be adapted to take into account text structure of the domain.

Training data for the CSN breast cancer and colorectal cancer forums were created by first manually labeling several hundred posts in each forum into positive or negative sentiment classes. Next, features were extracted from these posts.
<table>
<thead>
<tr>
<th>Label</th>
<th>Post</th>
</tr>
</thead>
<tbody>
<tr>
<td>Negative</td>
<td>My mom became resistant to carbo after 7 treatments and now the trial drug is no longer working :(, ...</td>
</tr>
<tr>
<td>Positive</td>
<td>ID-x, I love the way you think, ..., hope is crucial and no one can deny that a cure may be right around the corner!!!</td>
</tr>
</tbody>
</table>

Finally, classification models were trained (fit) to these data. The goal is to produce a fitted classification model that perfectly classifies posts back to their original defined positive or negative sentiment class using only the extracted features. The fitted sentiment classification model is then applied to all unlabeled posts allowing each to be classified to a sentiment class. These classified data are then used to investigate sentiment dynamics in the CSN forum.

### 4.3.1 Text Mining and Feature Extraction

A simple random sample of 298 posts was selected from the CSN breast cancer forum and each post manually classified as being of positive or negative sentiment with the result that 204 of them were labeled as positive and 94 were negative. An example of a negative and of a positive post is shown in Table 4.2. In the positive example, ID-x is the identifying code for a unique CSN participant.

Next, lexical and style features were extracted from the posts using standard text mining techniques. Features defined and extracted from the data are summarized in Table 4.3. Pos and Neg labels contain the numbers of posi-
tive and negative words (and emoticons) respectively in a post. The positive and negative word lists used to produce these counts are from Hu and Liu [34], and the positive and negative emoticon lists were collected from the Internet (http://en.wikipedia.org/wiki/List_of_emoticons). Many posts in the CSN forum mention names, e.g., ID-x, I love the way you think. To facilitate assessment of whether name mention has a relationship with sentiment, the feature Name contains a count of the occurrences of coded names in the post. The feature Slang contains the number of slang words used in the post. Thelwall et al. [88] introduced two additional features, PosStrength and NegStrength which were also measured for each post. Different from NumOfPos and NumOfNeg, PosStrength and NegStrength considers, in addition to the fact that a word is in the positive or negative word lists, the strength of emotion displayed. For example, very good and good!!! are scored as more positive than good. Learning algorithms [88] are used to establish the strength of individual words. The lists utilized at this stage of analysis are available at http://sites.google.com/site/qiubaojun/psu-sentiment.zip.

4.3.2 Sentiment Classification Models

Eight different classification models (classifiers) were used: AdaBoost, LogitBoost, Bagging, SVM, logistic regression, Neural Networks, BayesNet, and decision tree [97][30]. Given the small number of observations in the training data and the limited number of features considered, it was possible to examine all model com-
Table 4.3. Features for a post

<table>
<thead>
<tr>
<th>Feature</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>PostLength</td>
<td>The number of words</td>
</tr>
<tr>
<td>Pos</td>
<td>( \text{NumOfPos/PostLength} ), where NumOfPos is the number of positive words/emoticons</td>
</tr>
<tr>
<td>Neg</td>
<td>( \text{NumOfNeg/PostLength} ), where NumOfNeg is the number of negative words/emoticons</td>
</tr>
<tr>
<td>Name</td>
<td>( \text{NumOfName/PostLength} ), where NumOfName is the number of names mentioned</td>
</tr>
<tr>
<td>Slang</td>
<td>( \text{NumOfSlang/PostLength} ), where NumOfSlang is the number of Internet slangs</td>
</tr>
<tr>
<td>PosStrength</td>
<td>Positive sentiment strength [88]</td>
</tr>
<tr>
<td>NegStrength</td>
<td>Negative sentiment strength [88] (The value is more below 0 if the sentiment strength is more negative)</td>
</tr>
<tr>
<td>PosVsNeg</td>
<td>( \text{PosVsNeg} = \frac{\text{NumOfPos}+1}{\text{NumOfNeg}+1} )</td>
</tr>
<tr>
<td>PosVsNegStrength</td>
<td>( \text{PosStrength/NegStrength} )</td>
</tr>
<tr>
<td>Sentence</td>
<td>The number of sentences</td>
</tr>
<tr>
<td>AvgWordLen</td>
<td>The average length of words</td>
</tr>
<tr>
<td>QuestionMarks</td>
<td>The number of question marks</td>
</tr>
<tr>
<td>Exclamation</td>
<td>The number of exclamations</td>
</tr>
</tbody>
</table>

Combinations of features for each classifier and find the feature set that best classified sentiment in the training set for each classifier. Classification accuracy and ROC area were used as measures of goodness of fit. Classification accuracy is the percentage of training data observations correctly classified. The ROC curve for a binary classifier system is a plot of the true positive rate vs. the false positive rate for varying discrimination thresholds. The ROC curve is a measure of the ability of a classifier to produce good relative instance scores, and is insensitive to changes in class distribution \[26\]. ROC area is simply the area under the ROC curve. High ROC area and high classification accuracy are characteristic of a good classifier.
To avoid being too tied to the one training set, goodness of fit statistics are also estimated using 10-fold cross-validation.

The goodness of fit statistics for the best fitting feature set for each classifier are given in Table 4.4. AdaBoost, where regression trees are used as weak learners, had the best ROC Area (0.832) and classification accuracy (79.2%). This model used PostLength, Neg, PosVsNeg, Name, Slang, PosStrength, and NegStrength and had a false positive rate of 0.152, and false negative rate is 0.33. Not all available features were used in the final model. The AdaBoost classifier using all features has poorer fit, with ROC area of 0.813 and classification accuracy of 75.2% (see Table 4.3) demonstrating what is often the case that too many features can reduce the prediction ability of the model. Note that the excluded features are not necessarily weak. For example, Pos by itself has high classification power (see column 2 of Table 4.5) but is not included in the final model because it does not contribute additional discrimination given the other features already included (see column 4 of Table 4.5).

Considering the complexity of sentiment analysis, the observed performance of the classifiers is typical. Classification accuracy for other sentiment analyses reported in the literature for various domains range from 66% for movie reviews to 84% for automobile reviews [56][66].
Table 4.4. Best fit classifiers, their ROC area, and classification accuracy.

<table>
<thead>
<tr>
<th>Classifier</th>
<th>ROC Area</th>
<th>% Classification Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdaBoost</td>
<td>0.832</td>
<td>79.2%</td>
</tr>
<tr>
<td>Logistic Regression</td>
<td>0.832</td>
<td>77.5%</td>
</tr>
<tr>
<td>LogitBoost</td>
<td>0.816</td>
<td>76.8%</td>
</tr>
<tr>
<td>BayesNet</td>
<td>0.802</td>
<td>74.2%</td>
</tr>
<tr>
<td>Bagging</td>
<td>0.794</td>
<td>73.5%</td>
</tr>
<tr>
<td>Neural Networks</td>
<td>0.785</td>
<td>73.8%</td>
</tr>
<tr>
<td>Decision Tree</td>
<td>0.782</td>
<td>77.2%</td>
</tr>
<tr>
<td>SVM</td>
<td>0.658</td>
<td>75.2%</td>
</tr>
</tbody>
</table>

4.3.3 Feature Analysis

Using AdaBoost with the best fitting feature set, further studies were preformed to establish the importance of each feature to model performance, and to explore how individual features differ between posts classified to negative or positive classes. The goal of this latter analysis was to answer questions such as: Are negative posts less likely to mention users’ names? in which case the Name value should be less in negative posts, and Can positive posts contain negative words? assessed by looking at Neg.

A boxplot of features for negative and positive posts is given in Figure 4.4. This figure indicates that negative posts are slightly longer, contain more negative words, and have more negative strength on average, while positive posts mention more names, have higher ratios of positive and negative words, have larger positive strength on average, and use slightly more slang. Student’s t-tests assuming separate variances were used to assess the importance of individual features. The differences for PostLength (P = 0.08), Neg (P = 0.68) and Slang (P = 0.06) are
not significant, while PosVsNeg ($P < 0.001$) and Name ($P < 0.001$) are. T-tests were not run for PosStrength and NegStrength because they contain too few value points ($\{1, 2, 3, 4, 5\}$ and $\{-5, -4, -3, -2, -1\}$, respectively) and as a result do not satisfy Normality assumptions.

The importance of individual features in the best fitting feature set for AdaBoost is measured in two ways: first by examining the fit of the AdaBoost model with each feature used alone, and next, by examining whether adding the feature last into the model significantly improves overall model fit. Model fits for these two approaches are shown in Table 4.5. Note that PosStrength and PosVsNeg are the individual features best able to classify correctly alone in the model but these features are not greatly better than a number of the other features. The last-in statistics shows the performance decrease from the best fitting AdaBoost classifier
Table 4.5. Importance of features assessed individually or as last added for the AdaBoost Classifier.

<table>
<thead>
<tr>
<th>Feature introduced as:</th>
<th>Only ROC Area</th>
<th>Last-in ROC Area</th>
<th>Decrease from Best Fit</th>
</tr>
</thead>
<tbody>
<tr>
<td>PosStrength</td>
<td>0.696</td>
<td>0.774</td>
<td>0.054</td>
</tr>
<tr>
<td>PosVsNeg</td>
<td>0.694</td>
<td>0.804</td>
<td>0.024</td>
</tr>
<tr>
<td>Neg</td>
<td>0.459</td>
<td>0.813</td>
<td>0.015</td>
</tr>
<tr>
<td>Slang</td>
<td>0.527</td>
<td>0.813</td>
<td>0.015</td>
</tr>
<tr>
<td>NegStrength</td>
<td>0.544</td>
<td>0.813</td>
<td>0.015</td>
</tr>
<tr>
<td>PostLength</td>
<td>0.545</td>
<td>0.819</td>
<td>0.009</td>
</tr>
<tr>
<td>Name</td>
<td>0.572</td>
<td>0.820</td>
<td>0.008</td>
</tr>
</tbody>
</table>

if individual features are dropped from the model. Leaving any feature in the best fitting feature set does show some performance deterioration, although only PosStrength and PosVsNeg show large impact.

4.4 Sentiment Dynamics

The fitted AdaBoost model was used to establish the sentiment level for all (manually labeled and unlabeled) posts in the CSN forum. For any post $P$, the model generates the predicted probability $Pr(P)$ that the post belongs to the positive class. The corresponding probability for the negative class is computed as $1 - Pr(P)$. If $Pr(P) > 0.5$, post $P$ is classified as positive, otherwise, it is labeled as negative. There is, of course, some error in this classification since the fitted model is not perfect (ROC area=0.83, correct classification rate= 80%), but fit is sufficient for the purposes of analyzing sentiment dynamics in the CSN forum. Even if post $P$ is incorrectly classified, its $Pr(P)$ is likely to be close to 0.5 suggesting little confi-
dence in the classification. \( Pr(P) \) is used as a *Sentiment Indicator* that measures the likelihood that post \( P \) has positive sentiment in subsequent analyses.

For the 468,000 posts extracted from the CSN forum, using the fitted AdaBoost model and including the manually labeled posts, 45.9\% of initial posts are classified as negative (54.1\% classified positive) while 31.2\% of all posts are classified as negative (68.8\% classified positive).

### 4.4.1 Definition of Sentiment Change of Thread Originator

Sentiment change expressed by thread originators is measured as the difference between the initial-sentiment (denoted as \( S_0(T) \)) and the subsequent-sentiment (denoted \( S_1(T) \)) expressed within that thread (see Table 4.6). Threads that have no originator replies are excluded because it is not possible to estimate the change without a subsequent-sentiment. Threads without replies from anyone other than the thread originators are also excluded, because any sentiment change could not be attributed to interaction with others (impacting factors). With these exclusions, 23,164 or 47.5\% of the initial 48,000 threads were left.

### 4.4.2 Sentiment Change of Thread Originator vs. The Number of Replies

The number of replies from others in a thread reflects the level of interest in the discussion topic. By definition, *negative thread originators* are those whose
Table 4.6. Definition of initial-sentiment \( (S_0(T)) \) and subsequent-sentiment \( (S_1(T)) \) of a thread \( (T) \).

<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial-sentiment ( (S_0) )</td>
<td>Positive (+) If ( Pr(P_0) &gt; 0.5 ), where ( P_0 ) is the initial post by the originator, and ( Pr(P_0) ) is the probability output by the sentiment model</td>
</tr>
<tr>
<td></td>
<td>Negative (-) Otherwise</td>
</tr>
<tr>
<td>Subsequent-sentiment ( (S_1) )</td>
<td>Positive (+) If ( \sum_{i=1}^{n_0} Pr(P_{0i})/n_0 &gt; 0.5 ), where ( P_0 ) are replies by the thread originator (self-replies), and ( n_0 ) is the number of self-replies</td>
</tr>
<tr>
<td></td>
<td>Negative (-) Otherwise</td>
</tr>
</tbody>
</table>

initial-sentiment are negative \( (S_0 = -) \) and positive thread originators are those whose initial-sentiment are positive \( (S_0 = +) \). A comparison of the number of replies to a thread by others \( (n_1) \) between negative thread originators and positive thread originators is shown in Figure 4.5. The curve marked by triangles shows the change in the likelihood that a negative thread originator \( (S_0 = -) \) changes to a positive sentiment \( (S_1 = +) \) as the number of replies from others increases. The curve marked by circles shows the likelihood that a positive thread originator \( (S_0 = +) \) stays positive \( (S_1 = +) \) as a function of the number of replies from others. About 75% of negative thread originators subsequently express positive sentiment when at least one reply from others is received, and the probability increases as the number of replies from others increases. A similar trend is shown for positive thread originators. For a given number of replies from others, the probability of positive sentiment from a positive thread originator is always greater than that of a negative thread originator. Intuitively, it is expected that positive thread
<table>
<thead>
<tr>
<th>Number Of Replies Of Others</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>1+</td>
<td>0.75</td>
</tr>
<tr>
<td>3+</td>
<td>0.80</td>
</tr>
<tr>
<td>5+</td>
<td>0.85</td>
</tr>
<tr>
<td>7+</td>
<td>0.90</td>
</tr>
<tr>
<td>9+</td>
<td>0.95</td>
</tr>
<tr>
<td>11+</td>
<td></td>
</tr>
<tr>
<td>13+</td>
<td></td>
</tr>
<tr>
<td>15+</td>
<td></td>
</tr>
<tr>
<td>17+</td>
<td></td>
</tr>
<tr>
<td>19+</td>
<td></td>
</tr>
</tbody>
</table>

**Figure 4.5.** Circle: \( y = p(S_1 = +|S_0 = +, n_1 \geq x) \); Triangle: \( y = p(S_1 = +|S_0 = -, n_1 \geq x) \), where \( n_1 \) is the number of replies from people other than the thread originator, \( S_0 \) and \( S_1 \) are initial-sentiment and subsequent-sentiment, respectively.

origina\textcolor{red}{t} ors are more likely to have positive subsequent-sentiment than negative thread originators. These curves do not reach 1.0 (100%), because regardless of the number of thread replies, some negative thread originators will never change to positive subsequent-sentiment, and not all positive thread originators keep a positive subsequent sentiment. This fact may be attributed to forum participants who are not satisfied with the interactions received. For example, conflicts occur in some threads, which lead to negative impacts on both participants’ and originator subsequent-sentiment.

The number of self-replies \((n_0)\) by a thread originator reflects personal involve-
ment in the thread. Figure 4.6 shows the percent of negative thread originators (triangles) and positive thread originators (circles), who subsequently express positive subsequent-sentiment ($S_1 = +$) as a function of the numbers of self-replies to their own thread ($n_0$). Both functions increase with increasing number of self-replies suggesting that the continued involvement of originators in the thread they started is positively associated with positive subsequent-sentiment. Only a small number of threads have high numbers of self-replies leading to uncertainty in the true form of these functions. The dip in the negative thread originators function in the middle of the range and the drop in the positive thread originators function at the upper end of the range may be real, but may also reflect small-sample biases.

4.4.3 Sentiment Change Indicator

Further insights regarding factors that may contribute to the change of sentiment of thread originators is obtained by analysis of the Sentiment Change Indicator for a thread originator defined as

$$\Delta Pr = \sum_{i=1}^{n_0} Pr(P_{0i})/n_0 - Pr(P_0),$$

where $P_0$ is a thread originator’s initial post in a thread, $P_{0i}, 1 \leq i \leq n_0$, are self-replies by the thread originator, $n_0$ is the number of self-replies, and $Pr(.)$, generated by the AdaBoost-based sentiment model, indicates the probability of a post being classified as positive.
Figure 4.6. Circle: \( y = p(S_1 = + | S_0 = +, n_0 \geq x) \); Triangle: \( y = p(S_1 = + | S_0 = -, n_1 \geq x) \), where \( n_0 \) is the number of self-replies by the thread originator, \( S_0 \) and \( S_1 \) are initial-sentiment and subsequent-sentiment.

The larger the \( \Delta P_r \), the more likely that self-replies will be classified as positive by the sentiment model. Figure 4.7 is a rough estimate of the distribution of \( \Delta P_r \) for negative thread originators. From this distribution, roughly 7.9% of negative thread originators have \( \Delta P_r < 0 \). The average \( \Delta P_r \) is 0.14 (standard deviation=0.31) which is significantly larger than 0, suggesting that after interacting with others, negative thread originators are likely to publish posts that are of positive-sentiment.
Figure 4.7. Distribution of Sentiment Change Indicator ($\Delta P_r$).

4.4.4 Factors related to the Sentiment Change Indicator

The time a participant spends replying to a thread is highly correlated to the average number of words used in their replies (a quantity referred to as the average length of replies). The association between average length of replies and $\Delta P_r$ is explored in Figure 4.8. Each point is a thread, the black line is the fitted linear regression, and the circle points are average $\Delta P_r$ values computed for all threads within a small “window” of average length of replies for “windows” defined from left to right on the average length of replies axis. The regression slope is not statistically different from zero suggesting no relationship of $\Delta P_r$ with average length of replies. For replies greater than 11 words on average, $\Delta P_r \approx 0.4$. For
average length of replies below 11 words, the $\Delta Pr$ declines. This suggests that for the most part, the average length of reply is not an influential factor in changing sentiment.

![Figure 4.8](image-url)

**Figure 4.8.** Average length of replies is not a major driver of Sentiment Change Indicator $\Delta Pr$.

Intuitively, positive or negative sentiments may propagate, suggesting that the average sentiment of replies ($Pr$) by other people may contribute to the Sentiment Change Indicator ($\Delta Pr$). The Sentiment Change Indicator and average sentiment of replies for each thread are plotted as points in Figure 4.9. The slope of the linear regression (the straight line in the plot) of $Pr$ on $\Delta Pr$ is 0.125 and significantly different from zero ($P < 0.05$). The “window” average curve (circle points) confirms the linearity of the relationship. This relationship suggests that to some extent,
higher average sentiment in subsequent non-originator posts increases the likelihood of a positive sentiment change in the thread originator. The broad spread of the points suggests that this may not be a major driving factor.

![Figure 4.9](image)

**Figure 4.9.** Increasing *Sentiment Indicator* of replies by other people increases the originator’s *Sentiment Change Indicator*.

The importance of the topic of a thread to the community can be measured by the time elapsed before the first reply by other people. A scatterplot of $\Delta_{Pr}$ and the time elapsed before the first reply by other people is shown in Figure 4.10. The slope of the linear regression is not statistically different from 0, and the linearity of the relationship is confirmed by the “window” line (circle points). There is indication that when the time interval before first reply is very small (less than about 3.5 hours or 210 minutes), the average $\Delta_{Pr}$ is slightly larger suggesting
that only very timely replies may contribute to an increase of $\Delta Pr$.

Table 4.7 summarizes the effects of the number of self-replies, the number of others’ replies, the average number of words of others’ replies, average sentiment of others’ replies, and the time elapsed before the first reply by others.

![Figure 4.10. Sentiment Change Indicator ($\Delta Pr$) vs. the time elapsed before the first reply](image)

**Figure 4.10.** Sentiment Change Indicator ($\Delta Pr$) vs. the time elapsed before the first reply

### 4.5 Conclusions and Future Work

Using posts in an online health community for cancer survivors and caregivers (the ACS’s Cancer Survivors Network) containing a half million forum posts made over a period of ten years, sentiment analysis was conducted and the dynamics of
Table 4.7. Factors for sentiment change of thread originators

<table>
<thead>
<tr>
<th>Factor</th>
<th>Effects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Involvement (self-replies)</td>
<td>The more involved, the more likely a thread originator expresses positive subsequent-sentiment</td>
</tr>
<tr>
<td>The number of replies by others</td>
<td>The more replies from others, the more likely a thread originator expresses positive subsequent-sentiment</td>
</tr>
<tr>
<td>The average number of words in others’ replies</td>
<td>If all replies by others are too brief, then a thread originator is likely to express no or little positive change of sentiment</td>
</tr>
<tr>
<td>The average sentiment indicator ($Pr$) of others’ replies</td>
<td>If replies by others are more positive, then a thread originator is likely to express more positive sentiment change</td>
</tr>
<tr>
<td>The time elapsed before the first reply by others</td>
<td>If the first reply from other people is very timely, then a thread originator is likely to express more positive sentiment change</td>
</tr>
</tbody>
</table>

online users’ sentiment revealed. Using machine learning techniques to extract important features, multiple candidate classification models were studied from which an automated sentiment classifier was developed. The final AdaBoost sentiment model was used to identify post features useful in predicting participant sentiment. Finally, the sentiment dynamics of thread originators in discussion threads were analyzed and factors driving the dynamics studied.

This is the first study that researches the sentiment benefits and associated dynamics of a large-scale health-related electronic forum. Modeling and analysis showed that 75% to 85% of users in the CSN forum experienced a positive change in sentiment through their online interaction with other forum participants. The greater the number of replies by others to a thread, the more likely is the thread originator’s subsequent sentiment to be positive, regardless of the sentiment in
the initial post. Furthermore, the level of involvement of a thread originator in subsequent thread posts is positively correlated with their positive subsequent sentiment. Finally, the higher the average sentiment of other posts in a thread, the more likely the sentiment change of the thread originator will be positive.

In this work a classification is applied to each post as one entity, regardless of length. This post-level analysis has its limitation in that multiple sentiments about different topics are possible in one post, therefore, developing more fine-grained sentiment analysis, analogous to analyzing the sentiment about features of products [56][66], is an important direction for future research. While a binary classification of sentiment was used in this work (positive and negative), future studies will likely wish to fine tune sentiment using a multi-level classification system. This will require a different and more complex definition of a sentiment change indicator.

This study introduced two new features of posts, Name and Slang, which proved to be useful in discriminating positive from negative sentiment in posts. These features, along with others yet to be identified, need to be studied in other contexts to understand their real worth for assessing sentiment in online health communities. Other approaches to improving the sentiment classification model, such as subjectivity summarization [65] also need to be explored.

This research is significant in that it is a prototype others may find useful in guiding their analyses of the dynamics, impact, and opinions that may affect the
emotion and the wellbeing of individuals and subgroups in other online health communities. In addition, this analysis has identified factors associated with positive sentiment change that designers of new online health communities and managers of existing online health communities may find useful. The ultimate goal of this research is producing online health communities that support the emotional health of community members and contributes to improved quality of life for those dealing with disease or supporting someone who is dealing with disease.
Conclusions and Future Directions

5.1 Conclusions

This dissertation focuses on studying the dynamical evolution of networks which is one of the most important aspects of networks. The links between actors in social networks are the results of social interactions between the actors. However, most of the current network dynamics analysis is link centric. For instance, social network modeling studies networks by simulating individual links. A link centric approach for dynamics analysis has the following problems:

- Difficult to model n-ary \((n \geq 2)\) relationship;

- Difficult to include properties of social interaction in modeling network dynamics;

- Difficult to aggregate properties of past social interactions in network mod-
eling and prediction;

- Difficult to analyze impacts of social interactions (on actors and networks) using their properties.

To solve the problems, we studied network dynamics at both macro and micro levels from the perspective of social interactions. Specifically,

- For the micro-study of network structure, we investigated an event-driven social network modeling to model n-ary \( (n \geq 2) \) social interactions and incorporate their properties in modeling networks.

We proposed event-driven modeling. It was natural, general and powerful to use event-driven models to model n-ary \( (n \geq 2) \) social interactions and incorporate their properties in modeling networks. We first characterized the growth dynamics of real-world social networks from the perspective of event formation and evolution. We then proposed an event-driven framework to facilitate the creation of event-driven growth models. We also studied the evolution of activeness of actors and event formation in social networks, and exploited the effect of both locality and attachedness on the formation of new edges. We found that the average number of participants of events evolve over time in NanoSCI, and the effects of distance and degree on selection of participants of events also change over time. These analysis led us to propose an hybrid model based on an event-driven framework that considers the evolution of event formation and the joint effects of distance and degree.
Based on metrics that are informative in characterizing the network structure, such as degree distribution, degree-dependent clustering coefficients and $k_{nn}(k)$, our experiments showed that the networks generated by our event-driven hybrid model exhibit structures similar to real networks, while other non-event driven models failed to recreate these structures.

- For the micro-study of network structure, we developed behavior evolution based link prediction to discover temporal behavior patterns of nodes from past social interactions, and incorporate the behavior evolution to predict future links.

We studied the link prediction problem from the perspective of time by aggregating past social interactions to model node behavior evolution with time series features, and applied the node behavior evolution to link prediction based on a supervised learning framework. We proposed a time series-based approach to characterize the evolution of behavior of individual actors, and developed a novel collaborative and autoregressive time series prediction model to predict the future behavior of actors. We then developed a behavior evolution-based supervised machine learning approach for prediction of new links.

The experimental results on multiple different types of large scaled social network showed that our link prediction approach consistently achieves significant improvement on link prediction accuracy. This was the first work of
this kind to aggregate past social interactions to characterize and discover the behavior evolution, and to predict future links. The collaborative and autoregressive time series prediction model developed in the dissertation was novel, and could significantly improve the prediction accuracy. Our work also shedded a light on on classification or clustering of samples with features evolving over time. The work has application in many recommendation systems.

- For the study of sentiment effects of social interactions, we developed a computational social science approach based on machine learning and text mining techniques to discover the patterns of sentiment change of CSN members, and further investigated factors that affect the sentiment change for an online forum (nearly a half million posts) of American Cancer Society’s Cancer Survivor Network (CSN).

The research was the first of its kind to study the sentiment benefits and their dynamics on a large-scale health related electronic board. It has shown that 75%–85% users in the CSN forum experienced a positive change in sentiment through their online interaction with other forum users. This research also identified several important factors that are associated with positive sentiment change in the thread originators. The greater the number of replies by others to a thread, the more likely the thread originator’s subsequent sentiment is positive, regardless of the sentiment of his/her initial post. Furthermore, the level of involvement of a thread originator in the same thread
he/she started is positively correlated with his/her expression of improved subsequent sentiment. Finally, the higher the average sentiment of other posts in a thread, the more likely the positive sentiment change of the thread originator will be greater.

This research provided an important basis for future further large-scale studies about the dynamics, impact, and opinions that may affect the emotion and the wellbeing of individuals and subgroups in an online health community. In addition, factors associated with positive sentiment change can contribute to the design of new online health communities and the enhancement of existing online health communities that enhance emotional support and other benefits to community members, particularly those whose quality of life is greatly dependent on such support.

5.2 Future Directions

For social network modeling, we have proposed event-driven modeling. Event-driven modeling has the potential to model very richer information such as location of events and dependency of events. It can be an interesting future direction to explore event-driven modeling of richer information and investigate the global network properties in the networks simulated.

For link prediction, the future direction include using categorical sequences and developing customized classifiers for link prediction. Time series can describe
those aspects of node behavior that represented in numerical values. It may be interesting to conduct further investigation on using categorical sequences to better characterize behavior evolution in situations where this kind of sequences are available. For example, in scientific collaborative networks or message communication networks, the sequence of topics of paper or posts can be extracted to complete the charactering node behavior evolution and improve the link prediction.

Developing customized classifiers for link prediction can also be an interesting direction. Due to the sparsity of networks, the samples in link prediction problem is unbalanced. Also, the data can be noisy. For example, in supervised link prediction, node pairs without links are often sampled from network as negative training samples, which may be noisy if those node pairs which will have links in the next time step are chosen, special classifiers can be developed to handle the imbalance and noise in class label [74].

We can also incorporate the social interaction patterns with link prediction. For example, in scientific collaborative networks, if a researcher is very likely to co-author a paper with a graduate student, then the researcher is very likely to collaborate with the student’s advisor as well. In addition, we can combine the macroscopic and microscopic analysis. For example, we can use the results of microscopic analysis, for instance, the importance of features, to help design better growth models by using the best combination of top features. Finally, further research on community structure can be conducted by employing the node behavior
evolution patterns discovered in this dissertation to improve the detection and characterizing community and its evolutions [100, 33, 14].

For the analysis of sentiment effects, only posts rather than sentences are classified in our current work. The degree of difficulty in analyzing a post involving multiple sentiments about different topics is considerably greater. Hence, developing more fine-grained sentiment analysis (analogous to analyzing the sentiment about features of products [56][66]) is an important direction for future research. Furthermore, the binary classification of sentiment can be extended to multi-level sentiment assessment to achieve a more robust estimate of sentiment and define the sentiment change indicator directly using sentiment level difference between initial posts and the subsequent posts of thread originators. Regarding features for sentiment analysis, we have introduced two useful features, Name and Slang, which have not been widely used. Further study of these features along with other features are necessary to improve sentiment analysis. Also, incorporating other approaches such as subjectivity summarization [65] and build more sophisticated ensemble model may further improve the sentiment analysis.
Bibliography


Baojun Qiu  
(608) 698-9429  qiubaojun@gmail.com  

Education

Ph.D. in Computer Science and Engineering  
The Pennsylvania State University, 2006-2011  

Master of Art in Statistics  
The Pennsylvania State University, 2009-2011  

Master of Engineering in Computer Science  
Peking University, 2003-2006  

Bachelor of Science in Mathematics  
Peking University, 1999-2003  

Skills

Research Skills  
Conducted research in social network analysis, machine learning, sentiment analysis,  
text mining, and information retrieval as a research assistant at Pennsylvania State  
University for five years.  
Published 15+ technical papers.  

Programming Skills  
Developed large scale data processing programs and distributed and multi-threading  
programs  
Familiar with Java, C++, Matlab, R, Python and MapReduce (Hadoop and PIG)  

Software Design Skills  
Led the development for several hospital information systems and enterprises inform-  
ation systems;  
Won the second prize in “IBM CUP” Chinese Campus Creation Software Design  
Competition  

Statistics Skills  
Developed statistical programs in R, MATLAB, and SAS;  
Provided consulting in statistics at the Statistical Consulting Center at Pennsylvania  
State University