FriendBurst: Ranking People Who Get Friends Fast in a Short Time

Li Liu[a], Dandan Song[a,*], Jie Tang[b], Lejian Liao[a], Xin Li[a], Jianguang Du[a]

[a]*School of Computer Science and Technology, Beijing Institute of Technology, 100081 Beijing, China*
[b]*Department of Computer Science and Technology, Tsinghua University, 100084 Beijing, China*

Abstract

Number of friends (or followers) is an important factor in social network. Attracting friends (or followers) in a short time is a strong indicator of one person for becoming an influential user quickly. Existing studies mainly focus on analyzing the formation of relationship between users, however, the factors that contribute to users' friend (or follower) numbers increment are still unidentified and unquantified. Along this line, based on users' different friends (or followers) increasing speeds, firstly, we get a number of interesting observations on a microblog system (Weibo) and an academic network (Arnetminer) through analyzing their characteristics of structure and content from the diversity and density angles. Then we define attribute factors and correlation factors based on our observations. Finally we propose a partially labeled ranking factor graph model (PLR-FGM) which combines these two kinds of factors to infer a ranking list of the users' friends (or followers) increasing speed. Experimental results show that the proposed PLR-FGM model outperforms several alternative models in terms of normalized discounted cumulative gain (NDCG).

Keywords:
Friend Burst, Factor Graph Model, Ranking, Diversity, Density

1. Introduction

With the success of social Web, the online social networks, such as Facebook, Twitter and DBLP, significantly enlarge our social circles. The friends (or followers) in social networks are important resources not only for transferring messages but also for being popular, which can be considered as an crucial indicator of social status. For example, in microblogs, the increment of followers of a user means his/her published contents have more audiences and his/her actions could affect more people. And if a person's follower number has a great "burst" suddenly, he/she would probably become a "new star". On the other hand, in academic social networks, an author who get more coauthors in a short time means he/she is more active and can build reputation in his/her research area quickly. Understanding the characteristics of users who attract friends fast is an important issue for social influence analysis, which can provide suggestions for users' behaviors and benefit many applications such as "virtual market" and recommendation systems.

Generally speaking, quick increments of friends (or followers) means that the users get new relationships in a short time. In the literature, there exists some studies on relationship analysis, for example, link prediction and unfollow behavior analysis. The goal of link prediction [1, 2] is to predict whether one user will follow another in the future, while unfollow behavior analysis [3, 4] mainly focuses on analyzing the reason of unfollow behaviors. In sum, most of existing literatures study the formation of friendship between users. But another perspective, the factors that impact on users' friends increasing speed are still unidentified and unquantified. Although Hutto et al. [5] did a longitudinal study on followers increment, where they built a regression model for follower count prediction, the correlation between magnitude of content-based and structural factors and the friends increasing speed is ignored; moreover, their dataset is small.

Different with these works, we want to propose a method to infer the ranking list of friends increasing speed for candidate users. There are several challenges for the friends increasing speed ranking. First, how to capture the rich structural and content-based information for friend increment analysis? Second, how to construct an algorithm to model both the users' attributes and the relationships between users? Third, how to validate the proposed model in real large social networks.

To address the above challenges, we firstly perform some statistical analysis on the correlations between users' friend (or followers) increasing speed and their structural and content-based properties. The analysis is conducted based on two networks, namely, a microblog system (Weibo) and an academic network (Arnetminer). The structural and content-based information are studied with in-depth exploration. For the structure based analysis, we utilize calculations such as diversity and density of circles and structures; for the content based analysis, we also define diversity and density based on topic distribution and hashtags. We then propose a partially labeled ranking factor graph model (PLR-FGM) to infer the ranking list of friends increasing speed. The model can not only use the structural and contents-based properties of individuals as attribute factors, but also model the relationships between uses as the correlation factors. The ranking list can be obtained by sorting the marginal

*Corresponding author.
Email address: sdd@bit.edu.cn (Dandan Song)

Table 1: Statistic of the DataSets

Dataset	#nodes	#edges	#contents
Weibo	61,397	154,900	19,700,352
Arnetminer	66,313	112,237	901,522

probabilities which are calculated by the model. Experimental results show that our PLR-FGM model significantly outperforms several alternative methods in terms of normalized discounted cumulative gain (NDCG) with augments ranging from 6% to 51% .

The contributions of this paper are summarized as follows:

- Based on large datasets from two real social networks – a microblog system (Weibo) and an academic network (Arnetminer), we derive observations and analyze the correlation between users' friends increasing speed and the users' structures from diversity and density angles. Besides, we also analyze the effects of their contents properties (such as circle diversity and density) on the friends increasing speed.
- We propose a ranking factor graph model which not only incorporates the structural and contents-based properties of individual users but also model the relationships between them. Then we use the model to detect the users who have a higher friends increasing speed in the social network.
- We conduct experiments on the two real social networks. Experimental results verified the effectiveness of our observation factors, and the proposed model can achieve a better performance than several alternative models.

The paper is organized as follows: In Section 2, we give a brief description of the datasets we used and perform some preprocessing on these two datasets; Section 3 presents our observations on users' friends burst states with their attributes such as structure and content. Section 4 explains the proposed ranking factor graph model. Section 5 illustrates experimental results and validates the effectiveness of our methodology. Finally, Section 6 discusses related work and Section 7 concludes.

2. Data Collection and Preprocessing

The datasets we used in this paper are gathered from two different online social networks: a microblog system - Weibo, and an academic network - Arnetminer.

Weibo[1] is a Twitter-style website, which is the most popular microblogging service in China. We collected a network of 88,626 users with 27,080,987 posts and 264,799 edges. Besides, we crawled all the users' profiles which contain gender and verification status.

Intuitively, there are two kinds of users for attracting followers in Weibo, the first kind is the users who are already well known, such as celebrities and official announcers. They can get more followers in a short time by their real life reputation rather than taking social actions in the online social network. But in contrast, ordinary users (who are not the celebrities) have to attract followers by publishing interesting contents or performing social actions such as posting or discussing a hot topic. We perform a statistics on most top 2000 users who have the most followers, by manually labeled these users as celebrities or ordinary users based on their profiles. In detail, we label the users who are singers, movie stars, writers and CEOs of famous companies as celebrities, and mark the users who are not well known to many people in real society but get reputation and become famous bloggers in the Weibo platform as ordinary users. Results show that there are only 17% ordinary users among the top 100 users and the percentage of ordinary users increases while the number of followers decreases. For example, among the top 1000-2000 users, the percentage of ordinary users is 38.9%. As our research objects are ordinary users, we discard the top 5% (4431) users in the initial dataset by descendingly sorting their follower numbers. Besides, we also discard the bottom 5% users as they have few followers. Additionally, the users without any relationship with others are also discarded. After these preprocessing, there are 61397 users with 19,700,352 posts and 154,900 relationships remained in the dataset.

Arnetminer[2] [6] is a real online academic social network dataset, which is extracted from academic search and mining platform ArnetMiner. Before the preprocessing, it has 2,092,356 papers from publication venues major in Computer Science, and has gathered 1,712,433 researchers for more than 50 years (from 1960 to 2014). The full graph of co-author network contained in this data has 1,712,433 vertices (authors) and 4,258,615 edges (collaboration relationships).

To predict the co-author increasing count after a time interval, we keep the users who published papers in at least 6 years. Under this condition, there are 66,313 users with 901,522 papers and 112,237 collaboration relationships.

Table 1 lists statistics of the datasets after preprocessing, which are used in our later analysis.

3. OBSERVATIONS

Firstly, we denote the friends (or followers) increasing speed as the friends (or followers) increasing count in a certain time interval. The speed is always a continuous value, in order to facilitate our analysis and experiment, we divide the speeds into 5 burst states $s = \{1, 2, 3, 4, 5\}$. Similar with the topic burst detection [7], we assume that all the speeds are generated by five Possion distributions corresponding to the five states, then which burst state a user is in depends on which Possion distribution gets the highest probability for his/her speed. The probability of observing a speed v is defined as:

$$p(v|s_i) = \frac{e^{-\mu_{s_i}}\mu_{s_i}^{v}}{v!} \quad (1)$$

[1] http://www.weibo.com

[2] http://arnetminer.org/billboard/AMinerNetwork

where $\{\mu_1, \mu_2, \mu_3, \mu_4, \mu_5\}$ denote the expected value of speeds for the 5 states. In detail, μ_5 is the average speed of the top 20% highest speeds in the datasets, μ_4 is the average speed of the top 20%-40% highest speeds, and so on.

Our goal is to analyze the correlation between users' burst states and users' properties from both the diversity and density angles. Let $< 1, ..., t >$ be a sequence of timestamps with a particular time granularity (e.g., year, month, etc.). For Arnetminer, since we have the snapshots of all the users in different time intervals, we use the properties in period from 1 to $t-1$ as the base properties, and study the correlations between properties changing from $t-1$ to t and the burst state in $t+1$. Note that we use the burst state $t+1$ instead of t, the reason is that there is always a time lag between publishing papers and have new co-authors in the academic network. For the Weibo, due to the lack of snapshot data in our dataset, there are only two time points $< 1, t >$ for analysis. Fortunately, the time delay between preforming actions and attracting followers is short in Weibo, we study the correlations between properties changing from 1 to t and the burst state in t.

There may be some zombies in the Weibo Dataset. Since many zombies may not have much followers, which cause their follower increasing speed low, we discard bottom 5% users by counting their follower increasing speed. On the other hand, some zombies (which are controlled by machine) may follow each other in a short time which induce their follower increasing speed very high, so we also discard top 5% users by counting their follower increasing speed.

For all observations, we calculate the confidence intervals. Most of the error bars in the following figures are based on 95% confidence.

3.1. Structure Based Analysis

Structural information reflects the users' "positions" in the social network, which is an important factor for attracting new friends. In this section, we study the correlation between structural properties and burst states.

Circle Based Definition: The definition of circle is the same as [8], which is a group of interconnected people.

- *Circle diversity:* Intuitively, the higher number of the circle count, the richer the diversity of the user's online social life. Therefore, we use the circle count of a user to denote his/her circle diversity.

- *Circle density:* The circle density is defined as the sum number of edges in all circles of a user *i* divided by his/her circle count.

Structure Based Definition:

- *Structural density:* We define the structural density of a user as the number of his/her friends with a certain burst state, then we have five structural densities for every user as we have five burst states. For example, the $\text{StructureDensity}_1$ of user U_i is the number of friends with burst state 1 of user U_i.

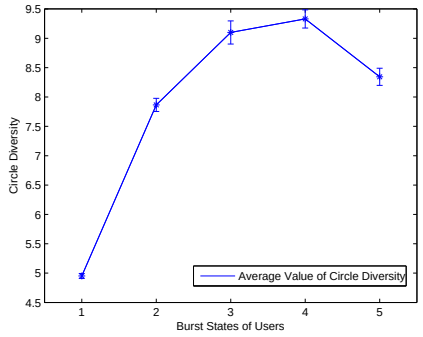
- *Structural diversity:* Based on the structural density, the structural diversity is how many circles that one user's friends with a certain burst state are involved. Similar with structural density, $\text{StructureDiversity}_1$ of user U_i is the circle count of user U_i's friends with burst state 1.

Circles Analysis: Due to the large scale of the online social networks, the circle always too big, which means that users almost have a few number of circles. Since the two users' distance is farther than 2 hops always means that the connection is weak between them, the tastes of these two users may be different. Similar to work [8], we mainly focus on the 2-ego network of users (2-ego network means a sub-network formed by one's friends and friends of friends). Note that we only use the 2-ego network in the circle analysis, while the whole network is used for the burst state prediction. For the Weibo dataset, the edges we use for analysis are reciprocal relationships, which means that the two users in one edge follow each other. Figure 1 illustrates the correlation between the users' average circle diversity values and their burst states. We can see from the figure that the users who have high burst states always have more circles. It means that big circle diversity value is helpful for users to attract new friends (or followers). In Weibo, an outlier is the users in burst state 5, but their circle diversity is still about 5.5, and the circle diversity value of this state is bigger than state 1 and state 2. So, you should make friends in more circles but don't disperse too much if you want to attract more friends in Weibo.

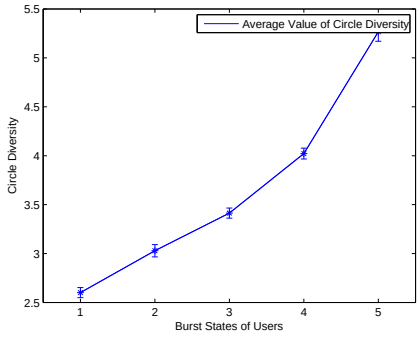
Then we want to determine whether the circle density also impacts on the burst state. An interesting phenomenon is that we get totally different curves on the two datasets (which are illustrated in Figure 2). In Weibo, users with very low or high burst state have a small density value. On the contrary, users with very low or high burst state have a big density value in Arnetminer.

The explanation of the phenomenon is that, researchers in Arnetminer with burst state 5 have bigger circle diversity and circle density, which means that friends of these researchers are in more circles and these friends have "strong" connections (with more edges in these circles). These properties are helpful for building their reputation, so they always have a high burst state. Meanwhile, researchers with burst state 1 have fewest circles but bigger circle density, which means that they are focusing on specific topics and their willingness to cooperate with others is low, leading these researchers' burst states to be low. Conversely, Weibo is a more "open" platform, in which users focus on more areas. Low circle density means the group is more "open" for the newcomer [9]. So the users with burst state 5 have a small circle density value. But the users with burst state 1 not only have small circle density value but also have small circle diversity value (they have less circles), so their ability of attracting new followers is poor.

Structure Analysis: Here we want to identify whether a users' friends' burst states impact on his/her friends increasing speed. We calculate the average speeds of users who have a certain amount of friends in a certain burst state. Figure 3 illustrates the results, where x axis is the number of users' friends

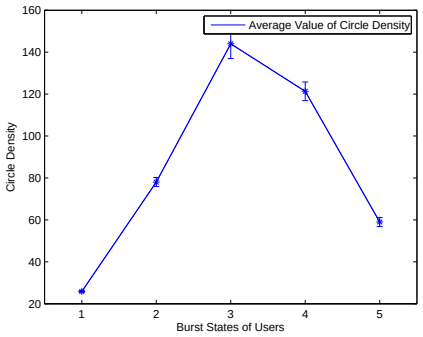


(a) Circle Diversity of Different Burst States for **Weibo**

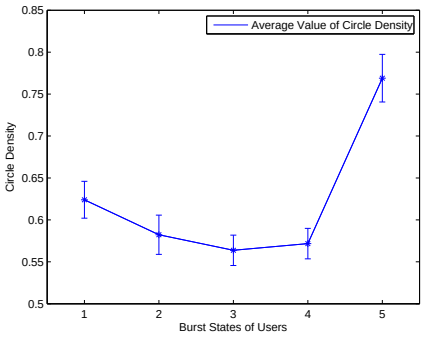


(b) Circle Diversity of Different Burst States for **Arnetminer**

Figure 1: Circle Diversity Analysis of Different Burst States. **Note:** Error bars represent 95% confidence intervals.

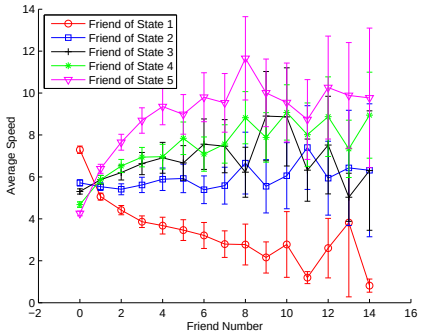


(a) Circle Density of Different Burst States for **Weibo**

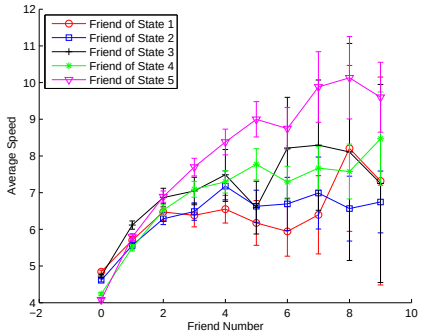


(b) Circle Density of Different Burst States for **Arnetminer**

Figure 2: Circle Density Analysis of Different Burst States. **Note:** Error bars represent 95% confidence intervals.



(a) Structure Density of Different Burst States for **Weibo**



(b) Structure Density of Different Burst States for **Arnetminer**

Figure 3: Structure Density Analysis of Different Burst States. **Note:** (1) Error bars represent 95% confidence intervals. (2) we keep the users who have more than 20 friends to make sure that we have enough data for the t-test

in a certain burst state, and y axis is the average friends increasing speed of the users. The figures show that, with the increasing number of friends with state 5, the users' average speed increases. In contrast, with increasing number of friend with state 1, the average speed decreases. The results match people's intuitions: if a user have more friends whose friends increase quickly, he/she will have a higher friends (or followers) increasing speed. Meanwhile, if a user have friends who attract friends (or followers) slowly, the user's friends increasing speed will be low. This is what we often say: "*Birds of a*

feather flock together".

Furthermore, we test whether a user's friends in different circles impact on the friends increasing speed of the user. We firstly focus on users' friends with a certain burst state, and then calculate the average speeds of these users when these friends are in different circles. In Figure 4a and Figure 4c, the burst state of the users' friends is 4 and the friends counts are varied from 5 to 2, then the average speeds of users whose friends are in different number of circles are calculated. For example, in the last cluster of Figure 4a, these users totally have 5 friends with burst state 4. The first column is the average speed of these users when their five friends with burst state 4 are in 5 circles. Similarly, the second column is the average speed of these users when their five friends are in 4 circles, and so on. Figure 4b and Figure 4d show a similar analysis, where the burst state of users' friends is 5.

For Arnetminer, the results are shown in Figure 4a and Figure 4b. We can see that when the users' friends are in less circles, the average speed is bigger. For example, in Figure 4a, in most of the time, when these friends are in less circles, the users' average speed is bigger. This phenomenon is more obvious in Figure 4b. The users in the last cluster have totally 5 friends. When these 5 friends are in five circles, the users' average speed is the smallest; however, when the friends are in one circle, the users' average speed is the biggest. Therefore, collaborating with authors who have high burst state in one circle is helpful for building the reputation.

For Weibo, we get different observations. We can see from Figure 4c, when the users' friends are in more circles, the average speed is bigger. There is an interesting phenomenon in Figure 4d, for the 5 friends with burst state 5, when their circle count is big or small in the two extremes, the users have a high average friends increasing speed, but when their circle count is at the "middle" level, the user's friends increase slowly. This suggests that making friends with high burst state all in one area or making friends with high burst state scattered in more areas are both helpful for attracting followers in Weibo.

3.2. Content Based Analysis

It is a basic way for users to use words to impress their research findings (or opinions), which may enhance their capability of attracting new relationships. Moreover, we use topic model (or hashtags) for semantic understanding of the papers (or Weibo posts). In order to find the correlation between users' burst states and content features, we introduce the following definitions of content features.

Word Based Definition:

- *Word diversity:* We use information entropy [10] to evaluate the diversity of a users' words. Specifically, the word diversity is defined as:

$$Diversity(u_i) = \frac{1}{N_{u_i}} \sum_{j=1}^{N_{u_i}} \sum_{n=1}^{|W_k|} -p(w_{kn}) * logp(w_{kn}) \quad (2)$$

  where N_{u_i} is the number of contents (papers or tweets) of u_i, $|W_k|$ is the length of content k, and $p(w_{kn})$ is the probability of *n-th* word of content w_k.

Topic Based Definition: For the Arnetminer dataset, since the topics have been investigated as a significant feature for literature contents for a long time, we utilize the unsupervised LDA (Latent Dirichlet Allocation) [11] to discover topics. We empirically train a model with 100-topics using the abstracts of all the papers. For the Weibo, due to the short lengths of users' posts, it's hard to extract their topics, so we use hashtags to denote the topics of contents.

- *Topic diversity:* Due to different definitions of topics, we have the different topic diversity definitions for the two datasets. For Arnetminer, similar with [12], the topic diversity of user u_i is defined as:

$$Diversity(u_i) = \frac{1}{N_{u_i}} \sum_{j=1}^{N_{u_i}} \sum_{k=1}^{|T|=100} -p(topic_k|d_j) * logp(topic_k|d_j) \quad (3)$$

  where N_{u_i} is the paper count of user u_i (who published N_{u_i} papers in total), d_j is one paper published by u_i, and $p(topic_k|d_j)$ is probability of the *k-th* topic for d_j.

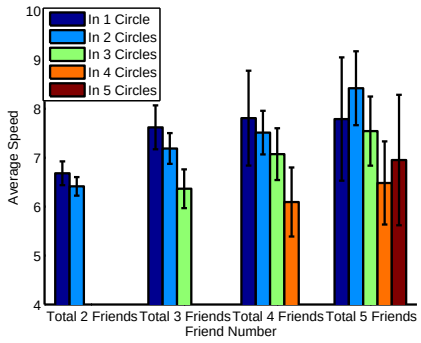
  For Weibo, topic diversity is the defined as the number of hashtags (if a hashtag appears in several tweets, we only consider its contribution to topic diversity as 1).

- *Topic density:* For Arnetminer, similar with [13], we define the topic density as the average cosine similarity for the distributions of every unique paired combination of user's papers. For Weibo, according to the special topic definition of Weibo, if two tweets contain a same hashtag, we draw an edge between them. And the topic density is defined as the total number of edges divide by his/her topic counts. Consequently, a user's contents have bigger topic density means that the user's contents are more *focusing* on some certain topics.

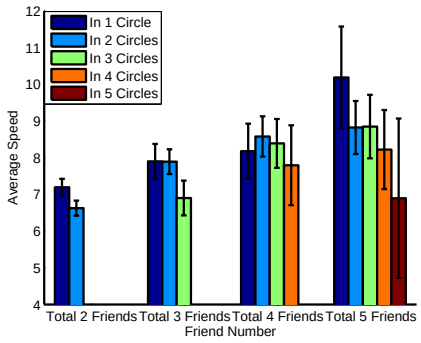
Diversity Analysis: Figure 5 shows the correlation between word diversity and burst states. We can see that, with the increasing of word diversity, the user's burst states upgrade. It means that, more diverse usage of words leads to higher speed of attracting friends (or followers).

Figure 6 shows the analysis of topic diversity, from where we can see that, the average topic diversity grows with the burst states. Therefore, topic diversity is also helpful for friends (or followers) increments. That means, when a Weibo user's contents cover more topics, his/her speed of attracting followers will be faster. For the researchers, publish papers in more areas is helpful for getting more co-author relationships.

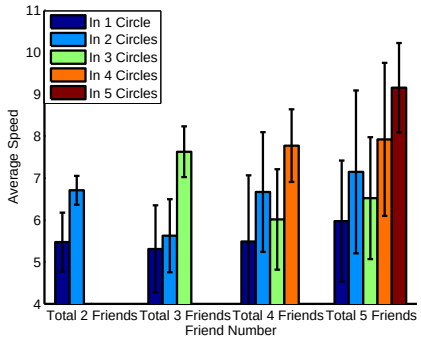
Density Analysis: For Weibo, Figure 7a illustrates the correlation between the topic density and the burst states in Weibo.



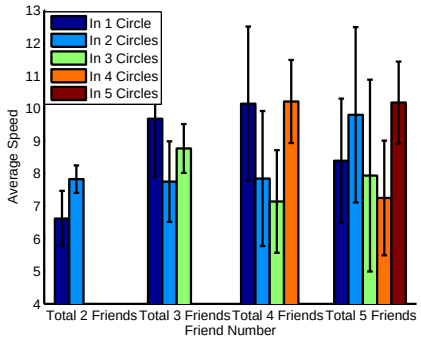
(a) Structural diversity of burst state 4 for **Arnetminer**: X axis shows different number of friends in state 4 and different legends denote these friends are in different circles. The Y axis is the average speeds of these users.



(b) Structural diversity of burst state 5 for **Arnetminer**: X axis shows different number of friends in state 5 and different legends denote these friends are in different circles. The Y axis is the average speeds of these users.

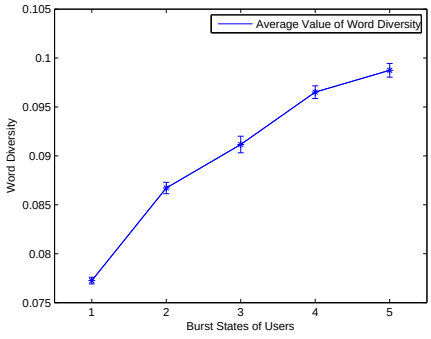


(c) Structural diversity of burst state 4 for **Weibo**: X axis shows different number of friends in state 4 and different legends denote these friends are in different circles. The Y axis is the average speeds of these users.

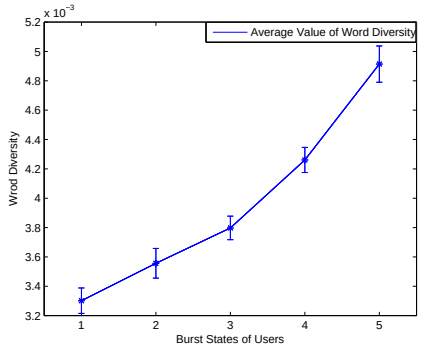


(d) Structural diversity of burst state 5 for **Weibo**: X axis shows different number of friends in state 5 and different legends denote these friends are in different circles. The Y axis is the average speeds of these users.

Figure 4: Structure Diversity Analysis. **Note:** (1) Error bars represent 95% confidence intervals. (2) Number of users who have total 5 friends and these friends are in 5 circles (which is the last column of the last cluster, the deepred column) is small, the error bars of this metric are based on confidence 85%.



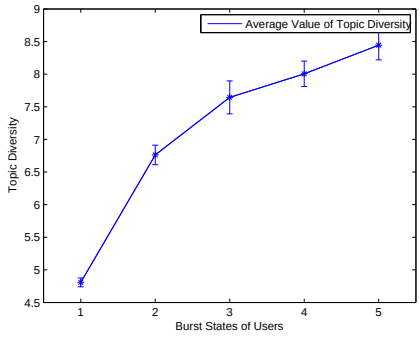
(a) Word Diversity of Different Burst States for **Weibo**



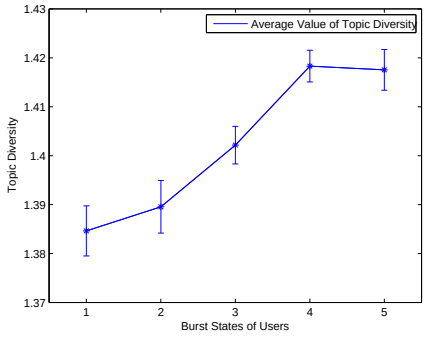
(b) Word Diversity of Different Burst States for **Arnetminer**

Figure 5: Word Diversity Analysis of Different Burst States. **Note:** Error bars represent 95% confidence intervals.

It shows another perspective on topics: a higher topic density is helpful for followers increment. It demonstrates that, if you want to attract more followers in a short time, you should "focus" on some topics. For the Arnetminer, Figure 7b shows a totally different observation: the users with higher burst state have lower topic density. It means that users should publish papers on more topics in order to attract new co-authors faster.

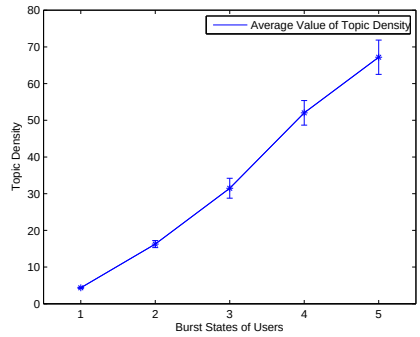


(a) Topic Diversity of Different Burst States for **Weibo**

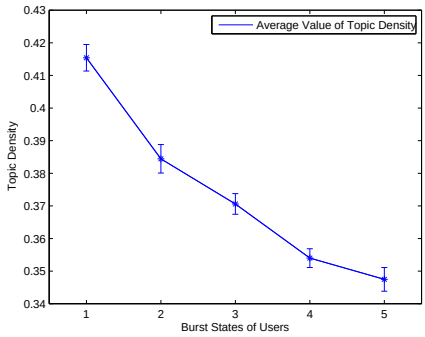


(b) Topic Diversity of Different Burst States for **Arnetminer**

Figure 6: Topic Diversity Analysis of Different Burst States. **Note:** Error bars represent 95% confidence intervals.



(a) Topic Density of Different Burst States for **Weibo**



(b) Topic Density of Different Burst States for **Arnetminer**

Figure 7: Topic Density Analysis of Different Burst States. **Note:** Error bars represent 95% confidence intervals.

4. Partially Labeled Ranking Factor Graph Model

Based on the observations in section 3, our goal is to design a model which can rank the friends increasing speed of users by incorporating the properties of structural and content-based information into the network. In this section, we describe the details of the proposed model.

4.1. Problem Definition

In this subsection, we define the concept of partially labeled network and present the formal definition of friends increasing speed ranking in the social graph.

Definition: ***Partially labeled network:*** Given a social network, a partially labeled network is $G(V^L, V^U, E, X)$, where V^L is a set of users whose friends increasing speeds are labeled and V^U is a set of unlabeled users, where $V = V^L \cup V^U$ and $|V| = N$ is a set of users and $E \subset V \times V$ is a set of relationships among them. X is a $N \times d$ attribute matrix with the element x_i^j indicating the j^{th} attribute of user v_i.

Our goal is to infer a friends increasing speed ranking list of all users V based on the attribute matrix X and their existing relationships. More specifically, we are concerned with the following problem:

Problem: ***Friends Increasing Speed Ranking:*** Number of friends (or followers) is an important factor in social network, and analysis of friends increasing speed can benefit several applications. For example, in Weibo, the users who get friends fast are often cost-effective than users who already have a large number of friends, because when we know which users are getting friends fast, we can use these users as seeds for promoting a product or spreading messages. For the ArnetMiner, getting friends fast means the researchers are active, which can be used in academic social recommendation.

We now describe the basic model of Friends Increasing Speed Ranking: Let $< 1, ..., t >$ be a sequence of timestamps with a particular time granularity (e.g., year, month, etc.). $G^t(V^L, V^U, E, X)$ is a partially labeled network, in which the users may have different friends increasing speeds. The task of friends increasing speed ranking is to find a predicative function such that we can get the speed ranking list for users in time $t + 1$ using their properties changing from $t - 1$ to t:

$$f : (G^1, ..., G^t) \rightarrow Y^{t+1} \tag{4}$$

where $Y^{t+1} = \{y_1, y_2, \cdots, y_{|N|}\}$ is a set of inferred results for users' probabilities of friend burst at time $t + 1$. In this graph, We split users into two sets according to their friends increasing speeds.

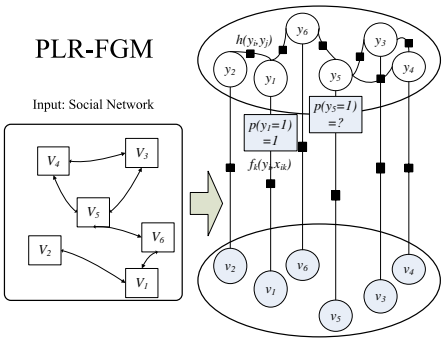


Figure 8: Partially-Labeled Ranking Factor Graph Model.

The half part of users in the set with higher speeds are labeled as 1 and others as 0. The predictive function outputs a probability $p(1|v_i)$ for users who have higher friends increasing speed. Thus, similar to [14], the friends increasing speed ranking problem turns into ranking the probability $p(Y^{t+1} = 1|G^1, ..., G^t)$ of all users.

From the model we can see that: our model based on "partially labeled networks" use the contents and structural properties before a period of time to predict the users speed of getting friends in the later period. This is useful in some scenarios, for example, when we know the researchers' changing of properties in t_1 to t_2 and part of users' follower increasing speed in t_2 to t_3, we can predict the rest users' increasing speeds in t_2 to t_3 "without" getting the properties of these users. Getting the properties of these users in a time interval (such as t_2 to t_3) is always time consuming (such as obtaining all friends in ArnetMiner needs to traverse all the papers of one user).

4.2. Framework

Based on the above intuitions, we propose a partially-labeled ranking factor graph model (PLR-FGM), which is shown in Figure 8. In the model, every user v_i is modeled as a node in the graphical model, in which the count of nodes in the model is N and the relationships of nodes can be modeled as the relationships of users naturally. Each v_i have a corresponding variable node y_i. Since the graph is partially labeled, the node set Y in PLR-FGM can be divided into two subsets Y^L and Y^U.

The model tries to capture two kinds of information. The first kind is the attributes associated with each user, which include the attributes of users' profiles, users' structural and content-based information described in the Observation section, and the second kind is the relationships between users. Specifically, we define the following two types of factors:

- *Attribute factors:* $F(y_i, X_i)$ represents the posterior probability of node y_i given attribute X_i.
- *Correlation factors:* $H(y_i, N(y_i))$ denotes the correlation influence between the relationships, where $N(y_i)$ is the set of correlate relationships to y_i.

Given a partially-labeled network $G = (V^L, V^U, E, X)$, by integrating all the factor functions together, we can define the joint distribution over Y according to Hammersley-Clifford theorem [15] as:

$$p(Y|G) = \prod_i F(y_i, X_i)H(y_i, N(y_i)) \tag{5}$$

The two kinds of factors can be instantiated in different ways. In this paper, we use the exponential-linear functions. In particular we define the attribute factors as

$$F(y_i, X_i) = \frac{1}{Z_\alpha} \exp\{\sum_{i=1}^{N}\sum_{k=1}^{d} \alpha_k f_k(y_i, x_{i,k})\} \tag{6}$$

where α_i is the weight of the attribute feature function, d is the dimension of attribute X_i and Z_α is the normalization factor.

Similarly, we define the correlation factors as:

$$H(y_i, N(y_i)) = \frac{1}{Z_\beta} \exp\{\sum_{e_{ij}\in E} \beta_{i,j} h(y_i, y_j)\} \tag{7}$$

where function $h(y_i, y_j)$ could be defined as several functional types. In the experimental section, we compare some widely used functions, such as indicator function and maximum function etc.

4.3. Parameter Learning

The key issue of PLR-FGM learning is to estimate the parameter configuration $\theta = (\alpha, \beta)$, which can be learned by maximizing log-likelihood of the labeled nodes. The joint probability defined in Eq.(5) can be written as:

$$p(Y|G) = \frac{1}{Z} \exp\{\sum_{i=1}^{N}\sum_{k=1}^{d} \alpha_k f_k(y_i, x_{i,k}) \sum_{e_{ij}\in E} \beta_{i,j} h(y_i, y_j)\} \tag{8}$$

where $Z = Z_\alpha Z_\beta$ is a normalization factor.

To calculate the normalization factor Z, we need to sum up the likelihood of all possible states of all nodes including unlabeled nodes. However, as we already shown, the graph model is partially labeled. To tackle this problem, we use the model trained by labeled data to infer the unlabeled nodes. Let Y denote a labeling configuration inferred from the known labels Y^L. Then, the log-likelihood objective function O is defined as:

$$\begin{aligned} O(\theta) &= \log P_\theta(Y|G) \\ &= \sum_{i=1}^{N}\sum_{k=1}^{d} \alpha_k f_k(y_i, x_{i,k}) \\ &\quad + \sum_{e_{ij}\in E} \beta_{i,j} h(y_i, y_j) - \log Z \end{aligned} \tag{9}$$

We use gradient decent to solve the objective function. Specifically, the gradient of each unknown parameter α with respect to the objective function is written as:

$$\begin{aligned} \frac{\partial O(\theta)}{\partial \alpha} &= \frac{\partial(\sum_{i=1}^{N}\sum_{k=1}^{d} \alpha_k f_k(y_i, x_{i,k}))}{\partial \theta} - \frac{\partial \log Z}{\partial \theta} \\ &= E[f_k(y_i, x_{i,k})] - E_{Y|Y^L}[f_k(y_i, x_{i,k})] \end{aligned} \tag{10}$$

As the social network graph structure in PLR-FGM can be arbitrary and may contain circles, it is intractable to obtain the exact solution of the Eq.(10) using exact inference methods such as Junction Tree [16]. Alternatively, we use Loopy Belief Propagation (LBP) [17] to approximate the solution. Specifically, we perform the LBP process twice in each iteration, one time for estimating the marginal probability $p(Y|G)$ and the other for estimating the posterior probability $p(Y|Y^L)$. We then calculate gradient and update each parameter with a learning rate η. The learning algorithm is summarized in Algorithm 1. Note that algorithm 1 only illustrates the learning algorithm for α, as the learning for β is similar to α by only replacing the f functions with the h functions.

Algorithm 1 Parameter Learning of the PLR-FGM Model

Input: partially labeled network G, learning rate η
Output: learned parameters $\theta = (\{\alpha\})$

procedure LEARNING(G,η)
  Initialize α
  repeat
    Calculate $E[f_k(y_i, x_{i,k})]$ using LBP;
    Calculate $E_{Y|Y^L}[f_k(y_i, x_{i,k})]$ using LBP;
    Calculate the gradient of α according to Eq.10:
    $\nabla_\alpha = E[f_k(y_i, x_{i,k})] - E_{Y|Y^L}[f_k(y_i, x_{i,k})]$
    Update the parameter α with the learning rate η:
    $\alpha_{new} = \alpha_{old} + \eta \cdot \nabla_\alpha$;
  until Convergence
  return α
end procedure

Model Inference: With the learned parameters α and β, we can infer users' friends increasing probabilities. Specially, we can predict the label configuration which maximize the joint probability:

$$Y^* = argmaxP(Y|G) \tag{11}$$

Then the loopy belief propagation is used again to compute the marginal probability of each node $p(y_i|Y^L)$. Similar with work [14], we solve our friends increasing speed ranking problem by sorting the probability $p(Y^{t+1} = 1|G^1, ..., G^t)$ of all users, and the users' friends increasing speed ranking list is obtained accordingly.

5. Experimental Results

In this section, we conduct several experiments based on the partially labeled ranking factor graph model to evaluate the effectiveness of the structural and contents-based properties. Firstly, we use the One-way ANOVA (analysis of variance) [18] to test the significance of our observations; then we present the performance of the comparative methods and our model. In the case of PLR-FGM, we conduct analysis on the feature contributions and iteration performance. As the ranges of the features are quite different, when we perform the experiments, we normalize all the features to the range [0,1].

Table 2: ANOVA Test of Structure and Content Features

Method	Arnetminer		Weibo	
	F Value	P	F Value	P
Circle Diversity	736.91	0	2601.83	0
Circle Density	4.39	0.0015	455.8	0
$\text{StructureDensity}_5$	15.41	0	487.85	0
$\text{StructureDensity}_4$	11.05	0	798.46	0
$\text{StructureDensity}_3$	0.95	0.4802	608.25	0
$\text{StructureDensity}_2$	2.13	0.0244	722.18	0
$\text{StructureDensity}_1$	1.29	0.2386	703.15	0
$\text{StructureDiversity}_{5,5}$	2.6	0.0363	17.12	5e-13
$\text{StructureDiversity}_{5,4}$	0.99	0.3982	75.47	0
$\text{StructureDiversity}_{5,3}$	4.98	0.0007	146.39	0
$\text{StructureDiversity}_{5,2}$	10.59	0.0011	878.38	0
$\text{StructureDiversity}_{4,5}$	3.99	0.004	31.54	0
$\text{StructureDiversity}_{4,4}$	3.09	0.0264	62.64	0
$\text{StructureDiversity}_{4,3}$	8.91	0.0001	260.22	0
$\text{StructureDiversity}_{4,2}$	4.5	0.0339	858.2	0
Content Diversity	128.22	0	3439.54	0
Topic Diversity	41.24	0	990.35	0
Topic Density	155.11	0	22.51	0

5.1. Significance Test

We use ANOVA to test the significance of the structural and contents-based properties in the observation section. Table 2 lists the results.

In Table 2, $\text{StructureDensity}_i$ means that we test the correlation between friends increasing speeds and structure density values of users when they have different count of friends in burst state i, $\text{StructureDiversity}_{i,j}$ examine the correlation between friends increasing speeds and structure diversity values (i.e., their circle numbers) for users who have j friends with burst state i, for example, $\text{StructureDiversity}_{5,4}$ controls the users who have 4 friends with burst state 5.

From Table 2 we can see that, most of the features are significance at the $\alpha = 0.05$ level for Arnetminer. But there are three exceptions: $\text{StructureDensity}_3$,$\text{StructureDensity}_1$ and $\text{StructureDiversity}_{5,4}$. We found that the instances count we use for testing are 20, 41 and 117, respectively, which is smaller than other features such as $\text{StructureDensity}_{5,4}$ which has 2295 instances. We found that, the features can pass the test when they have enough instances (more than 200). For Weibo, all of the features are significant at the level $\alpha = 0.01$.

5.2. Experimental Setup

The datasets we used for experiments are listed in Table 1, and we randomly choose 10% users as our test dataset, the rest 90% as our training dataset.

Evaluation Metrics: We quantitatively evaluate the performance of ranking lists using a normalized discounted cumulative gain measure (NDCG) [19], which is computed as

$$NDCG_p = \frac{1}{N} \sum_{i=1}^{p} \frac{2^{rel_i} - 1}{\log_2(i+1)} \tag{12}$$

where N is the normalization constant so that a perfect ordering gets the NDCG score 1. Note that, as the probability is continuous, it is not suitable for the NDCG metric. We evaluate the NDCG by burst states which is defined in Section 3, so rel_i is the burst states of user i.

Factor Definition: Based on the observations in Section 3, we define factors for the PLR-FGM model to derive users' friends increasing speed ranking list. The attribute factors contain three types of features. The first type of features are based on the user's basic information, hence gender and verification status are used for users in Weibo and h-index used for authors in Arnetminer. The second type of features are the structural properties of users and the third type of features are content-based features.

Besides these features that can be used in other ranking models such as Ranking SVM, considering the relationships of users in the social network are important, we also introduce correlation factors which are incorporated into our model. As we described in subsection 4.2, the correlation factor functions can be defined as several functions types. We define three types of correlation factor functions:

- *Indicator Function:* $h(y_i, y_j) = 1$, if y_i and y_j have a relationship, then the value of the function is 1.
- *Burst Maximum Function:* $h(y_i, y_j) = \max\{State_{y_i}, State_{y_j}\}$, if y_i and y_j have a relationship, then the value of the function is the bigger burst state between y_i and y_j.
- *Burst Difference Function:* $h(y_i, y_j) = abs(State_{y_i} - State_{y_j})$, if y_i and y_j have a relationship, the value of the function is the absolute value of the difference between y_i's and y_j's burst states.

specially, Table 3 lists all the factors we used in our PLR-FGM model.

Comparative methods: Given the partially labelled input network G, we can construct a training dataset with the labelled nodes: $V^L = \{(X_i, y_i)_{i=1,\cdots,n}\}$, where X_i is the feature vector associated with user $v_i \in V^L$, which is composed of attribute factors listed in Table 3. In this way, some alternative ranking methods can also be trained and to predict the results of users' friends increasing speed ranking lists. We compare the performance of our approach with the following methods:

Ranking SVM [20]: it is a widely used pair-wise ranking model, which treats every two pairs of samples as one instance, and trains a classification model to predict which instance have a higher relevant score. When all the pairs are ordered, the list of instances is ordered. We use the LIBLINEAR package to implement the Ranking SVM [21].

Coordinate Ascent [22]: it is a list-wise ranking model, which is a linear feature-based model that uses supervised training algorithms to directly maximize the evaluation metric such as NDCG. We use RankLib[3] to implement the Coordinate Ascent algorithm.

[3] http://people.cs.umass.edu/~vdang/ranklib.html

5.3. Ranking Performance Analysis

Performance Analysis: Table 4 lists the performance comparisons for inferring friends increasing speed ranking lists with different methods. As Ranking SVM and Coordinate Ascent can not capture the correlation factors, in Table 3, PLR-FGM only uses all the attribute factors, without correlation factors. We can see that even with the attribute factors only, our PLR-FGM method consistently outperforms other comparative methods. It achieves the best performance in $NDCG_{100}$. For Weibo, the model gets a +0.03 ($\approx$ 6%) increment compared with Ranking SVM and outperforms Coordinate Ascent +0.09 ($\approx$ 20%) at $NDCG_{100}$. For Arnetminer, our model gets a +0.05 ($\approx$ 10%) increment compared with Ranking SVM and outperforms Coordinate Ascent +0.1 ($\approx$ 22%) at $NDCG_{100}$. In other $NDCG$ metrics such as $NDCG_{500}$, $NDCG_{1000}$, our method also have a better performance than other models.

Feature contribution analysis: We perform an analysis to evaluate the contribution of different features defined in the models. Note that as the users' attributes (gender and verification status) only denote the user type, it is meaningless for the model to use the user attribute features only. So using the users' attribute features as a basement, we test the performance of PLR-FGM with structural features and content-based features respectively. Figure 9 illustrates the performance of the model with different features.

From Figure 9 we can see that, For Weibo, in the $NDCG_{100}$ metric, the model with content features achieves the best performance. Which means that, "good" content features (high topic and content diversity) is more useful than other features for the users who want to achieve a high burst state. In $NDCG_{500}$ and $NDCG_{1000}$ metrics, content features are also more important than structural features, and the model with all features performs better than the model only with the content features. In these metrics, both the content-based features and the structural features are helpful for the model. This means that, for the users who are "NOT" in very high burst states, the structural features is also helpful for attracting new friends (or followers).

For Arnetminer, in all metrics, the model with all features get the best performance. In the $NDCG_{100}$ metric, the result is similar to the Weibo dataset, where better content features lead to higher burst state. However, different to Weibo, in all metrics of Arnetminer, the structural properties are also important factors. We can see in $NDCG_{500}$ and $NDCG_{1000}$ metrics, the structure is even more important than the content. As described in the observation section, "good" structure in both density and diversity angles is helpful for constructing cooperations.

Correlation factor contribution analysis: The correlation factor functions are added to PLR-FGM with all attribute factors. Figure 10 illustrates the results of PLR-FGM with different factor functions.

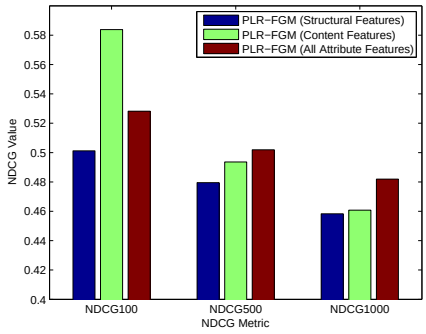
For the Weibo dataset (which is shown in Figure 10a), the model with the Burst Maximum Function has the best performance. The $NDCG_{100}$ value is 0.6244, so the PLR-FGM with Burst Maximum Function achieves +0.096 ($\approx$ 18%) compared with PLR-FGM without correlation factors *PLR-FGM (only Attribute Factors)*. Besides, the model achieves a +0.126 ($\approx$ 25%)

Table 3: Factor Definitions in our PLR-FGM.

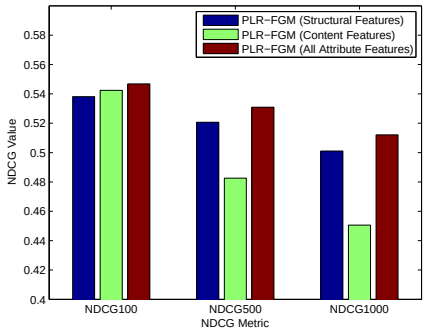
Factors			Weibo	Arnetminer
Attribute Factors	User Attribute	Gender	1 for male and 0 for female	-
		Verified	1 for verified and 0 otherwise	-
		h-index	-	value of h-index
		citation	-	total citation count
	Structural Features	Circle Diversity	#circles	
		Circle Density	#edges in circles/#circles	
		Structural Density	#friends with different states	
		Structural Diversity	#circles of friends with different states	
	Content Features	tweets count	#tweets	#papers
		retweet ratio	#retweeted tweets/#all tweets	-
		positive words usage	#positive words/#total words	-
		negative words usage	#negative words/#total words	-
		Content Diversity	information entropy	
		Topic Diversity	#hashtags	described by Eq.3
		Topic Density	#hashtag arcs	cosine similarity of topics
Correlation Factors	Indicator Function		$h(y_i, y_j) = 1$	
	Burst Maximum Function		$h(y_i, y_j) = \max\{State_{y_i}, State_{y_j}\}$	
	Burst Difference Function		$h(y_i, y_j) = abs(State_{y_i} - State_{y_j})$	

Table 4: Ranking Performance Comparison of Different Methods

Method	NDCG Value for Weibo			NDCG Value for Arnetminer		
	100	500	1000	100	500	1000
Coordinate Ascent	0.4372	0.3826	0.3625	0.4499	0.4901	0.4637
Ranking SVM	0.498	0.4316	0.4450	0.4952	0.485	0.4654
PLR-FGM (only Attribute Factors)	**0.5282**	**0.5019**	**0.4819**	**0.5468**	**0.5309**	**0.5121**



(a) NDCG Values of PLR-FGM With Different Features for **Weibo**



(b) NDCG Values of PLR-FGM With Different Features for **Arnetminer**

Figure 9: NDCG Values of PLR-FGM With Different Features

increment compared to Ranking SVM and a +0.187 ($\approx$ 42%) increment compared to Coordinate Ascent (which are listed in TABLE 4). In other metrics, the PLR-FGM with Burst Maximum Function also performs better than other models. The reason is that Coordinate Ascent and Ranking SVM ignore the relationship information between users, which is approved to be crucial for the analysis. Our PLR-FGM uses the burst states as the correlation factors to model the relationships, which is more reasonable than other methods.

For the correlation factor functions, we can see that, the model with Burst Maximum Function performs better than other models with different functions. Surprisingly, the Burst Difference Function has an opposite effect on the performance. One possible reason is that, when two users have the same burst state, the value of the burst difference function is 0, so the PLR-FGM will ignore the relationships between users, which lead to the graph sparse. Moreover, as we described in the Observation section, a user's friends burst states have influence on his/her burst state, but the usage of the difference between two users' burst states will reduce the influence. Besides, the Indic-

tor function does not perform well in the Weibo dataset either. The reason is that it considers all the influence between users equally as 1. The Burst Maximum Function not only models the relationship of the true social network, but also considers the biggest influence between the users, so this correlation factor is indeed helpful for the PLR-FGM model. We can also know from here, friends with high burst state are important for users to attract new followers.

For the Arnetminer (which is shown in Figure 10b), more obvious results are shown. The model with Burst Maximum Function performs best. The $NDCG_{100}$ value is 0.6795. It achieves +0.132 ($\approx$ 24%) compared with PLR-FGM without correlation factors *PLR-FGM (All Attribute Factors)*. Besides the model achieves a +0.184 ($\approx$ 37%) increment compared to Ranking SVM and a +0.229 ($\approx$ 51%) increment compared to Coordinate Ascent. Moreover, the PLR-FGM with Indictor function which simply models the relationship between users also performs better than the PLR-FGM without correlation factors.

From both datasets, we can know that, the relationships of the users are indeed helpful for prediction. For the relationship, if we consider the maximum influence between users, the model performs best. It means that, one user's friends increasing speed is highly relevant to his friends with high burst states. If one user wants to get more friends in a short time, he/she should make more friends with high burst states.

Iteration performance: As there is an iteration in the PLR-FGM model's learning process, whether the learning algorithm can converge is an important issue for the model. In order to evaluate the converge performance of the model, we choose the model which has the best performance above: PLR-FGM (Attribute Factors + Burst Maximum Function) as the example to show the NDCG value of every iteration.

Figure 11 illustrates the iteration performance. Note that when the iteration count is 0, we randomly change the index of users in the perfect (where users are descendingly sorted by their burst states) ranking list, then we calculate the NDCG value, we totally run 10 times and get the average value. For the Weibo dataset we can see that when the iteration number is more than 100, the NDCG500 and NDCG1000 values tend to be stable. When the iteration number is more than 150, the NDCG100 value tends to be stable. The speed of the NDCG100 value becoming stable is slower than others, but it becomes smooth in a small iteration numbers. For the Arnetminer dataset, all the NDCG values become smooth after 150 iterations.

There is a phenomenon that all the *NDCG* values shock a little, this is because the social network graph structure in PRP-FGM are arbitrary and contains circles. When we use the loopy belief propagation to calculate the marginal probabilities, the probabilities of some nodes can not converge. But the difference between the *NDCG* values is small. So it can reach a "relatively" stable value and have a good performance.

Time Complexity: Our learning algorithm is based on Loopy Belief Propagation whose complexity is $O(2 * E)$, in which E is the count of edges. Since the learning of our model needs an iteration, the time cost is related on the iteration count. From the iteration performance analysis (which is described in the "Iteration Performance" subsection), we can see that our model can be converged in several hundred iterations. In contrast, for some pair-wise algorithms (such Ranking SVM), it needs to construct instances of every two users, which induces a large dataset ($|V| * |V|$ nodes). Therefore, the time cost of our model is more time-saving compared with them.

6. Related Work

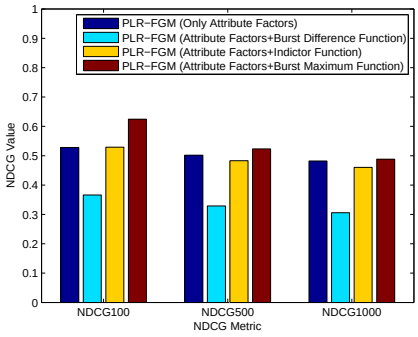
In recent years, there are some researches about the social network analysis have been conducted [23, 24, 25, 26]. Moreover, there exist some analysis on relationships of online social network [1, 2, 3, 4, 5]. Golder et al. [1] analyzed two structural characteristics, transitivity and mutuality and proposed a hierarchical regression model to predicted the tie formation. Kwak et al. [3, 4] analyzed the structural properties and actions, and studied the unfollow behavior. Liben-Nowell et al. [2] developed approaches for link prediction based on measures of analyzing the "proximity" of nodes in a network. Hutto et al. [5] focused on finding which factor is powerful on followers increment. They conducted the analysis on a dataset which contains 507 users, and proposed a model for predicting the count of followers' increment. But the correlation between magnitude of structural and content factors and the followers increasing speed has not been studied. Our aim is finding the users with high followers increasing speed, and we propose our analysis from diversity and density angles. Besides, we build the ranking model on a large data sets.

Ugander et al. [27] studied the structural diversity effect in social contagion. Inspired by their work, we propose the diversity analysis of structure. Moreover, we extend the concept of diversity to the content, and propose the PLR-FGM ranking model based on both structural and content-based diversity and density angles.

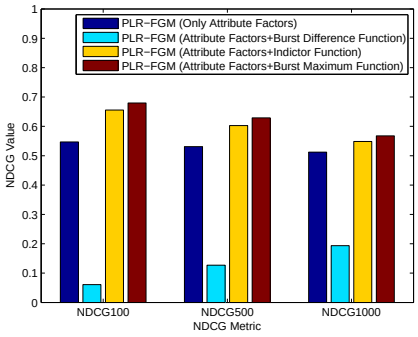
Meanwhile, factor graph model [28] was widely used in the analysis of the social networks, Zhuang [29] and Tang [30] proposed partially labeled factor models for supervised learning, which are used for social relationship mining. They denote relationships of users as nodes in the graph, and classified the node into different relationship types.

As an important application of friends (or followers) analysis, research of social influence is a hot area, where considerable works have been conducted. Several works [31, 32, 33, 34] focused on verifying the effect of social influence. For example Anagnostopoulos et al. [31] proposed a shuffle test to examine the existence of social influence. Bakshy et al. [32] conducted randomized controlled trials to identify the effect of social influence on consumer responses to advertisement. Bond et al. [33] used a randomized controlled trial to verify the social influence on political voting behavior. Crandall et al. [34] have developed techniques for identifying and modeling the interactions between influential users for user selection using data form online communities.

There are lot of works on quantifying the social influence, Tang et al. [35] presented a Topical Affinity Propagation (TAP) approach to quantify the topic-level social influence in large networks. Saito et al. [36] measured the pairwise influence between two individuals based on the idea of independent cascade

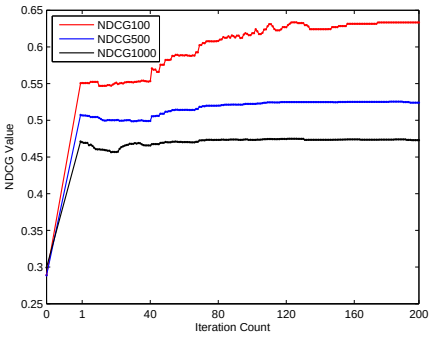


(a) NDCG Values of PLR-FGM with Different Correlation Factors for **Weibo**

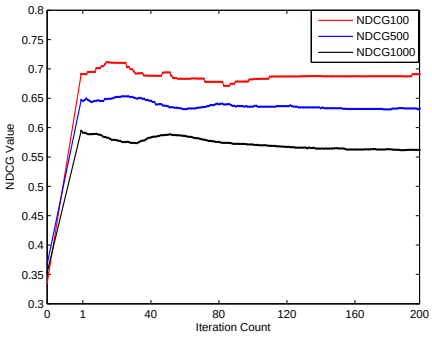


(b) NDCG Values of PLR-FGM with Different Correlation Factors for **Arnetminer**

Figure 10: NDCG Values of PLR-FGM with Different Correlation Factors.



(a) NDCG Values of PLR-FGM with Different Correlation Factors for **Weibo**



(b) NDCG Values of PLR-FGM with Different Correlation Factors for **Arnetminer**

Figure 11: NDCG Values of PLR-FGM with Different Correlation Factors.

model. Shuai et al. [37] studied the indirect influence using the theory of quantum cognition. Belak et al. [38] investigated and measured the influence between two communities. Myers et al. [39] proposed a probabilistic model to quantify the external influence out-of network sources. Goyal et al. [40] presented a method to learn the influence probabilities by counting the number of correlated social actions. Tan et al. [41] proposed a model to track the user's action based on the effects of influence, correlation, and uses action dependency. Li ea al. [42] tried to study the interplay between influence and individual conformity. Zhang et al. [8] proposed the concept of social influence locality and used a large microblogging network to study how users behavior is influenced by close friends in their ego networks. Tang et al. [43] focused on conformity influence. They defined several types of conformity factors, and used the factor model to solve the problem. As the friend (or followers) number is an important factor of social influence, our work can benefit these social influence analysis and help to find whether one user can become an influential user fast.

Some other works are about maximizing the spread of influence through a social network. Kempe et al. [44] found a small number of influential users to adopt a product to trigger a large cascade for further adoptions through the effect of "word of mouth". P. Domingos [45] built a probabilistic model to mine the spread of influence for viral marketing, and they proved the model to be NP-hardness. Chen et al. [46] developed efficient algorithms to approximately solve the influence maximization problem. In the influence maximizing problem, the cost of seed users who already have more friend (or followers) are always high, instead, with our method, finding the seed users who will get friends fast in the future maybe a more cost-effective solution.

7. Conclusion

In this paper, we study a novel problem of identifying and quantifying which factors cause users' friends (or followers) number increasing fast. Focusing on the friends increasing speed, we analyze the properties of structure and content from the diversity and density angles and get some interesting observations from two typical social networks – a microblog system (Weibo) and an academic network (Arnetminer). We analyze the observations and conduct statistical evaluations. We formally define the friends increasing speed ranking problem in a semi-supervised framework, and then propose a partially-labeled ranking factor graph model (PLR-FGM) to infer the

ranking list of friends increasing speed of users. Two kinds of factor functions are defined in the model. The attribute factors are used to represent the properties of users' contents and structure. The correlation factors are defined to capture the users' relationship in the network. We then use the loopy belief propagation algorithm to calculate the marginal probability, and propose a gradient decent to learn model parameters. Experimental results show that the proposed method outperforms several alternative methods.

Understanding which factor have effects on attracting new friends (or followers) fast is important for several applications of social networks. The problem represents a new research direction in social influence analysis. For future work, it will be interesting to study how different types of social ties impact on the users' friends burst. Studying the users' linguistic changes is also a intriguing direction. And for our proposed model, a more efficient algorithm (such as parallel algorithms [47]) will be considered to learn the parameters.

8. Acknowledgments

This work is funded by the National Program on Key Basic Research Project (973 Program, Grant No. 2013CB329600), National Natural Science Foundation of China (NSFC, Grant Nos. 61472040, 60873237, and 61300178), and Beijing Higher Education Young Elite Teacher Project (Grant No. YETP1198), and Basic Research Foundation of BIT.

9. Reference

[1] S. A. Golder, S. Yardi, Structural predictors of tie formation in twitter: Transitivity and mutuality, in: Social Computing (SocialCom), 2010 IEEE Second International Conference on, IEEE, 2010, pp. 88–95.

[2] D. Liben-Nowell, J. Kleinberg, The link prediction problem for social networks, Journal of the American society for information science and technology 58 (7) (2007) 1019–1031.

[3] H. Kwak, H. Chun, S. Moon, Fragile online relationship: a first look at unfollow dynamics in twitter, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ACM, 2011, pp. 1091–1100.

[4] H. Kwak, S. B. Moon, W. Lee, More of a receiver than a giver: Why do people unfollow in twitter?, in: ICWSM, 2012.

[5] C. Hutto, S. Yardi, E. Gilbert, A longitudinal study of follow predictors on twitter, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ACM, 2013, pp. 821–830.

[6] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, Z. Su, Arnetminer: Extraction and mining of academic social networks, in: KDD'08, 2008, pp. 990–998.

[7] Q. Diao, J. Jiang, F. Zhu, E.-P. Lim, Finding bursty topics from microblogs, in: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1, Association for Computational Linguistics, 2012, pp. 536–544.

[8] J. Zhang, B. Liu, J. Tang, T. Chen, J. Li, Social influence locality for modeling retweeting behaviors, in: Proceedings of the Twenty-Third international joint conference on Artificial Intelligence, AAAI Press, 2013, pp. 2761–2767.

[9] D.-B. Chen, H. Gao, L. Lü, T. Zhou, Identifying influential nodes in large-scale directed networks: the role of clustering, PloS one 8 (10) (2013) e77455.

[10] C. E. Shannon, A mathematical theory of communication, ACM SIGMOBILE Mobile Computing and Communications Review 5 (1) (2001) 3–55.

[11] D. M. Blei, A. Y. Ng, M. I. Jordan, Latent dirichlet allocation, the Journal of machine Learning research 3 (2003) 993–1022.

[12] R. Yan, J. Tang, X. Liu, D. Shan, X. Li, Citation count prediction: learning to estimate future citations for literature, in: Proceedings of the 20th ACM international conference on Information and knowledge management, ACM, 2011, pp. 1247–1252.

[13] Y.-C. Wang, R. Kraut, Twitter and the development of an audience: those who stay on topic thrive!, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ACM, 2012, pp. 1515–1518.

[14] Y. Dong, J. Tang, S. Wu, J. Tian, N. V. Chawla, J. Rao, H. Cao, Link prediction and recommendation across heterogeneous social networks, in: Data Mining (ICDM), 2012 IEEE 12th International Conference on, IEEE, 2012, pp. 181–190.

[15] J. M. Hammersley, P. Clifford, Markov fields on finite graphs and lattices.

[16] W. Wiegerinck, Variational approximations between mean field theory and the junction tree algorithm, in: Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence, Morgan Kaufmann Publishers Inc., 2000, pp. 626–633.

[17] K. P. Murphy, Y. Weiss, M. I. Jordan, Loopy belief propagation for approximate inference: An empirical study, in: Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence, Morgan Kaufmann Publishers Inc., 1999, pp. 467–475.

[18] B. G. Tabachnick, L. S. Fidell, et al., Using multivariate statistics.

[19] K. Järvelin, J. Kekäläinen, Cumulated gain-based evaluation of ir techniques, ACM Transactions on Information Systems (TOIS) 20 (4) (2002) 422–446.

[20] T. Joachims, Optimizing search engines using clickthrough data, in: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2002, pp. 133–142.

[21] C.-P. Lee, C.-J. Lin, Large-scale linear ranksvm, Neural Computation (2014) 1–37.

[22] D. Metzler, W. B. Croft, Linear feature-based models for information retrieval, Information Retrieval 10 (3) (2007) 257–274.

[23] M. Wang, K. Yang, X.-S. Hua, H.-J. Zhang, Towards a relevant and diverse search of social images, Multimedia, IEEE Transactions on 12 (8) (2010) 829–842.

[24] K. Yang, M. Wang, X.-S. Hua, H.-J. Zhang, Social image search with diverse relevance ranking, in: Advances in Multimedia Modeling, Springer, 2010, pp. 174–184.

[25] X. Zhao, J. Yuan, R. Hong, M. Wang, Z. Li, T.-S. Chua, On video recommendation over social network, Springer, 2012.

[26] X. Zhao, J. Yuan, M. Wang, G. Li, R. Hong, Z. Li, T.-S. Chua, Video recommendation over multiple information sources, Multimedia systems 19 (1) (2013) 3–15.

[27] J. Ugander, L. Backstrom, C. Marlow, J. Kleinberg, Structural diversity in social contagion, Proceedings of the National Academy of Sciences 109 (16) (2012) 5962–5966.

[28] H.-A. Loeliger, An introduction to factor graphs, Signal Processing Magazine, IEEE 21 (1) (2004) 28–41.

[29] H. Zhuang, J. Tang, W. Tang, T. Lou, A. Chin, X. Wang, Actively learning to infer social ties, Data Mining and Knowledge Discovery 25 (2) (2012) 270–297.

[30] W. Tang, H. Zhuang, J. Tang, Learning to infer social ties in large networks, in: Machine Learning and Knowledge Discovery in Databases, Springer, 2011, pp. 381–397.

[31] A. Anagnostopoulos, R. Kumar, M. Mahdian, Influence and correlation in social networks, in: Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2008, pp. 7–15.

[32] E. Bakshy, D. Eckles, R. Yan, I. Rosenn, Social influence in social advertising: evidence from field experiments, in: Proceedings of the 13th ACM Conference on Electronic Commerce, ACM, 2012, pp. 146–161.

[33] R. M. Bond, C. J. Fariss, J. J. Jones, A. D. Kramer, C. Marlow, J. E. Settle, J. H. Fowler, A 61-million-person experiment in social influence and political mobilization, Nature 489 (7415) (2012) 295–298.

[34] D. Crandall, D. Cosley, D. Huttenlocher, J. Kleinberg, S. Suri, Feedback effects between similarity and social influence in online communities, in: Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2008, pp. 160–168.

[35] J. Tang, J. Sun, C. Wang, Z. Yang, Social influence analysis in large-scale networks, in: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2009, pp. 807–816.

[36] K. Saito, R. Nakano, M. Kimura, Prediction of information diffusion probabilities for independent cascade model, in: Knowledge-Based Intelligent Information and Engineering Systems, Springer, 2008, pp. 67–75.
[37] X. Shuai, Y. Ding, J. Busemeyer, S. Chen, Y. Sun, J. Tang, Modeling indirect influence on twitter, International Journal on Semantic Web and Information Systems (IJSWIS) 8 (4) (2012) 20–36.
[38] V. Belák, S. Lam, C. Hayes, Cross-community influence in discussion fora., in: ICWSM, 2012.
[39] S. A. Myers, C. Zhu, J. Leskovec, Information diffusion and external influence in networks, in: Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2012, pp. 33–41.
[40] A. Goyal, F. Bonchi, L. V. Lakshmanan, Learning influence probabilities in social networks, in: Proceedings of the third ACM international conference on Web search and data mining, ACM, 2010, pp. 241–250.
[41] C. Tan, J. Tang, J. Sun, Q. Lin, F. Wang, Social action tracking via noise tolerant time-varying factor graphs, in: Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2010, pp. 1049–1058.
[42] H. Li, S. S. Bhowmick, A. Sun, Casino: towards conformity-aware social influence analysis in online social networks, in: Proceedings of the 20th ACM international conference on Information and knowledge management, ACM, 2011, pp. 1007–1012.
[43] J. Tang, S. Wu, J. Sun, Confluence: Conformity influence in large social networks, in: Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2013, pp. 347–355.
[44] D. Kempe, J. Kleinberg, É. Tardos, Maximizing the spread of influence through a social network, in: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2003, pp. 137–146.
[45] P. Domingos, M. Richardson, Mining the network value of customers, in: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2001, pp. 57–66.
[46] W. Chen, Y. Wang, S. Yang, Efficient influence maximization in social networks, in: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2009, pp. 199–208.
[47] J. Zhai, W. Chen, W. Zheng, Phantom: Predicting performance of parallel applications on large-scale parallel machines using a single node, in: ACM SIGPLAN Notices, Vol. 45, ACM, 2010, pp. 305–314.