

國立交通大學

資訊科學與工程研究所

碩士論文

基於行動社交網路建立機率模型做時間相關之
聯絡人推薦

A Probabilistic Model for Time-Dependent
Contacts Recommendation in Mobile Social
Applications

研究生：周凡凱

指導教授：彭文志 教授

中華民國 102 年 7 月

基於行動社交網路建立機率模型做時間相關之聯絡人推薦

A Probabilistic Model for Time-Dependent Contacts

Recommendation in Mobile Social Applications

研究生：周凡凱 Student : Fan-Kai Chou

指導教授：彭文志 Advisor : Wen-Chih Peng

國立交通大學

資訊科學與工程研究所

碩士論文

A Thesis

Submitted to Department of Computer and Information Science

Institute of Computer Science and Engineering

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Computer Science

July 2013

Hsinchu, Taiwan, Republic of China

中華民國 102 年 7 月

基於行動社交網路建立機率模型做時間相關之聯絡人推薦

學生：周凡凱

指導教授：彭文志

國立交通大學資訊科學與工程研究所

摘要

本論文提出了一種新穎的框架對使用者在一個特定的時間點從過去的歷史通信紀錄中做時間相關之聯絡人推薦。我們找出一些基本因素來處理互動資訊以及自動形成時間相關之聯絡人群組以配合一些場景，例如說清晨的時候我應該撥打電話給誰？在午夜的時候誰的 Email 我應該最先回？我們建立了一個機率模型，不僅可以捕捉到使用者與候選聯絡人之間的時間相依傾向，還揉合了聯絡頻率及新近程度到聯絡人群組中。我們還利用此機率模型來支持兩種型態的時間相關之聯絡人推薦：Seedset Generation：單一互動推薦 以及 Friends Suggestion：多人互動推薦。我們在三種實際資料集上進行實驗，結果顯示，使用我們提出的機率模型可以有效地做時間相關之聯絡人推薦。

A Probabilistic Model for Time-Dependent Contacts Recommendation in Mobile Social Applications

Student : Fan-Kai Chou

Advisor : Dr. Wen-Chih Peng

Institute of Computer Science and Engineering
National Chiao Tung University

ABSTRACT

This paper presents a novel framework for time-dependent contacts recommendation for a query user at a given time point from historical communication logs. We identify the fundamental factors that govern interactions and aim to automatically form time-dependent contact groups for scenarios, such as, *who should I dial to in the early morning?* *whose mail would I reply first at midnight?* We develop a probabilistic model that not only captures temporal tendencies between the query user and each contacts candidate but also blends frequency and recency into group formation. We also utilize the model to support two types of time-dependent contacts recommendation: **Seedset Generation**: single-interaction suggestion and **Friends Suggestion**: multiple interactions suggestion. Experimental results on Enron dataset, Call Detail Records and Reality Mining Data from MIT prove the effectiveness of time-dependent contacts recommendation with proposed probabilistic model.

誌謝

首先誠摯的感謝指導教授彭文志博士，老師悉心的教導使我對資料探勘領域更深入了解，經過不斷頻繁地討論並適時指引我正確做研究的方向，讓我在這兩年間學習到了非常多的東西，不只是在做研究上，在做人處事方面也受益匪淺。

本論文的完成另外亦得感謝 Young 學姐的大力協助，不厭其煩地聽著我提出一堆初級問題且一一為我解答，明明自己的事情也多到爆炸還是幫助我修改論文，使得本論文能夠更完整嚴謹。

回憶這兩年的日子，雖然偶爾偷懶卻也不乏熬夜認真的時候，實驗室裡共同的生活點滴，學術上的討論、互嗆的白爛話語、愛吃宵夜又有罪惡感只好去運動、一起打羽球打桌球跑步、熬夜趕報告趕meeting.....，感謝眾位學長姐、同學、學弟妹讓這兩年的研究生生活變得多采多姿。

感謝麵線學長，從來沒遇過這麼和藹可親的學長，總是能夠跟我們打成一片就像同學一樣，但是在需要認真的時候卻又變身為可靠的學長。感謝建成學長，雖然因為是在職生導致比較少見面，但是每次見面的時候總是能夠提供厲害的新想法。感謝 Jordan 和湯姆熊跟我做同一個計畫的夥伴，總是互相扶持互相幫忙。也感謝泰德、你有熊、建志的幫忙，總是聊一些有的沒的來紓解壓力，但是在遇到困難需要幫助的時候還是放下手邊的事情熱心幫忙，恭喜我們大家一起順利走過這兩年。感謝涓湘學弟，郁婷、苑庭、星星、曉雯學妹提供給了實驗室歡笑，跟你們聊天總是能夠放鬆緊繃的心情，也謝謝你們在口試當天的大力相助。

最後，這是我人生第一篇碩士論文也是最後一篇吧(?)，就獻給你們吧!!>.^

Contents

1	INTRODUCTION	1
2	RELATED WORK	5
2.1	Recommendation System	5
2.2	Community Detection	6
2.3	Relationship Link Prediction	7
2.4	Friends Suggestion System	7
3	TIME-DEPENDENT CONTACTS RECOMMENDATION SYSTEM	8
3.1	Temporal Probabilistic Model	9
3.2	Seedset Generation	13
3.3	Friends Suggestion	14
4	TIME INTERVAL ADJUSTMENT	16
4.1	Entropy Examination	16
4.2	Close Peak Detection	17
5	EXPERIMENT	20
5.1	Data Description	20
5.2	Analysis of Time Centrality	21
5.3	Experimental Setup	22
5.4	Metrics	23
5.5	Probabilistic Model Results	24

5.6 Time Interval Adjustment Results 26

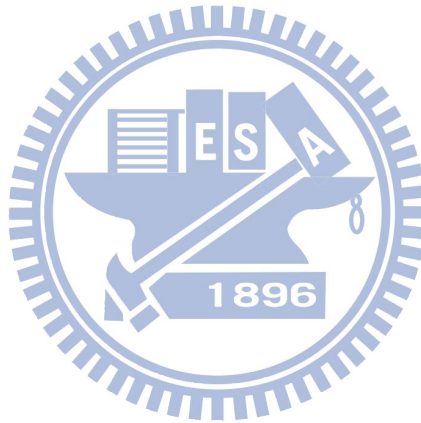
6 CONCLUSION 33



List of Figures

3.1	Framework Overview	9
3.2	Distribution of Time Centrality	10
5.1	Probability Distribution Function of Time Centrality with 4 time slots in (a) CHT cdr (b) RMD (c) Enron.	28
5.2	Probability Distribution Function of Time Centrality with 6 time slots in (a) CHT cdr (b) RMD (c) Enron.	28
5.3	Probability Distribution Function of Time Centrality with 8 time slots in (a) CHT cdr (b) RMD (c) Enron.	28
5.4	Probability Distribution Function of Time Centrality with 12 time slots in (a) CHT cdr (b) RMD (c) Enron.	29
5.5	Probability Distribution Function of Time Centrality with 24 time slots in (a) CHT cdr (b) RMD (c) Enron.	29
5.6	Cumulative Distribution Function of Time Centrality with 4 time slots in (a) CHT cdr (b) RMD (c) Enron.	29
5.7	Cumulative Distribution Function of Time Centrality with 6 time slots in (a) CHT cdr (b) RMD (c) Enron.	29
5.8	Cumulative Distribution Function of Time Centrality with 8 time slots in (a) CHT cdr (b) RMD (c) Enron.	30
5.9	Cumulative Distribution Function of Time Centrality with 12 time slots in (a) CHT cdr (b) RMD (c) Enron.	30

5.10 Cumulative Distribution Function of Time Centrality with 24 time slots in (a) CHT cdr (b) RMD (c) Enron.	30
5.11 nDCG Comparison for Seedset Generation in (a) Enron Mail (b) CHT cdr (c) RMD.	31
5.12 HitRate Comparison for Seedset Generation in (a) CHT cdr (b) RMD.	31
5.13 Comparison of Gmail Approach and Probabilistic Suggestion Approach(PSA) for Friends Suggestion	32
5.14 Comparison of Entropy Distribution	32
5.15 nDCG Comparison between dynamic and fixed time interval in (a) CHT cdr (b) RMD.	32



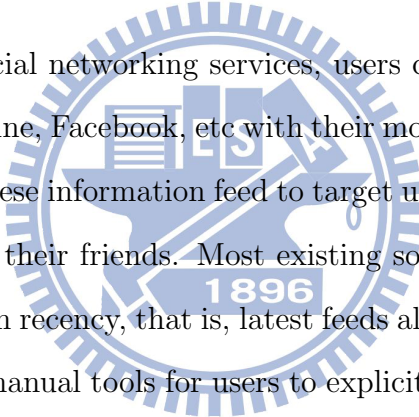
List of Tables

5.1	Basic Information on the Enron/CHT/RMD Datasets	21
5.2	Four slots of Time in 24 hours	21
5.3	Parameter Settings	23
5.4	Four Methods for Comparison	24
5.5	Symbols of Time Interval Adjustment Methods	26



Chapter 1

INTRODUCTION



As the emergence of on-line social networking services, users can easily share information to their friend groups via Gmail, Line, Facebook, etc with their mobile devices. Social networking services gather and syndicate these information feed to target users. Users can browse through the information feed shared by their friends. Most existing social network services generally render information feed based on recency, that is, latest feeds always appear on top of personal feed walls. Some may provide manual tools for users to explicitly adjust friend groups so that users can control how information feed are rendered on their walls or which friend groups to share information with. Such great efforts motivate us to ask one question: Is it possible to design a time-dependent contacts recommendation system which can automatically suggest a ranked list of friend candidates driven by both historical interaction statistics and contextual information such as time point?

So far, the formation of groups on existing social media applications is still static or requires manual management by users themselves, where a group is a fixed set of contact that is manually pre-defined by a user. We argue that the notion of group should dynamically adapt to the user tendency and the context information such as location, time, etc. That is, some users may have the tendency to share information to different groups of contact at different time points while some users may have relatively static tendency and share information to

the same group of contact all the time. For example, a user may have the tendency to share information to his/her family during daytime and share information to his/her close colleagues in the evening. To discover a users time-dependent contacts, we need to identify his/her tendency at different time point. Following this, a users tendency to share at different time point, we need to provide a ranked list of contacts that a user has the highest probability to interact with at each time point.

The general problem of recommendation system has been widely studied [13]. Recently, several prior studies attempt to consider temporal factor in designing recommendation systems [16][6][5][21]. For example, [16] leveraged user’s long-term and short-term preference for temporal recommendation. Nonetheless, non of them addresses the fact that user interactions are not always correlate with time as users present diverse variation of temporal dependency. For example, it is quite clear that some users have higher temporal dependency in sharing information. Moreover, a user may only be sensitive to certain time points during a day. In this paper, we argue that temporal tendency should be analyzed individually for each pair of query user and friend candidate at each time point. As an evidence, Figure 3.2 illustrates a distribution of temporal centrality for all pairs of users. If a pair of users’ interactions only fall into a few time slots during a day, they has lower entropy and thus indicating higher temporal centrality and vice versa. We observe that over 60% pairs of users’ have higher time centrality in interactions ($\text{entropy} \leq 0.5$), meaning the rest 40% user interactions are driven or dominated by other factors.

In this paper, we propose a framework to discover personalized time-dependent contacts at different time points. Specifically, given a query user, a time point, and historical communication logs, our recommendation system returns a time-dependent contacts group, represented by a ranked list of contact users, for the query user at given time point. To achieve this, we propose a temporal probabilistic model to capture user behaviors based on three factors: frequency, recency and time-dependency. After this, we utilize the temporal probabilistic model to support two types of time-dependent contacts recommendation: **Seedset Generation**: single interaction suggestion and **Friends Suggestion**: multiple interactions suggestion. The

temporal probabilistic model considers the dynamic importance of each candidate user for a query user to incorporate the factor, *different users show different temporal dependency with related to a target user*. Seedset Generation enables to generate a single candidate user automatically by proposed temporal probabilistic model for two purposes: shifting the burden of query users to provide a list of users who intent to interact with at the very beginning, and the query user merely intends to interact with a single user at given time. On the other hand, Friend Suggestion aims to provide a group of friends whenever the query user intends to interact with multiple users at the same time.

Recommending time-dependent contacts is useful in many applications. For example, time-dependent contacts can be utilized to enhance the ranking results for content-based on-line social networking services (e.g., Gmail, Line, Facebook), where the information feed for each user can be adjusted based on the time-dependent contacts. Moreover, it can be used in location sharing services (e.g., Foursquare), where the ranking of locations can be adjusted based on a user's time-dependent contacts at particular time point. Our contributions are as follows.

- We propose a framework to discover personalized time-dependent contacts for a query user at given time point.
- We propose a temporal probabilistic model to capture user's interaction tendency at different time point.
- We identify three fundamental factors in user interactions and propose approaches to support: single interaction suggestion and multiple interactions suggestion by integrating the temporal probability model into the-state-of-the-art ranking model.
- We proposes two methods to find the most appropriate time interval for our probabilistic model.
- We conduct experiments on real datasets to demonstrate the effectiveness of our framework and report empirical insights.

This paper is organized as follows. Section 2 presents the related work for this paper. Section 3 introduces our temporal probabilistic model and then discusses the two types of time-dependent contacts recommendation system. Section 4 proposes two methods to find the most appropriate time interval for our probabilistic model. Section 5 shows the experimental results using the three real datasets. Section 6 concludes this paper.



Chapter 2

RELATED WORK

This section discusses four kinds of related works. First, we discuss some literatures of recommendation system. Those problems are closely related to time-dependent contacts recommendation if friends is substituted for those items. Second, we show that the difference between community detection and our problems and explain why we use **circle** in this paper instead of **group**. Third, we talk about the link prediction problems because relationship link prediction problems are similar to friends suggestion in some situations. Then, we discuss about friends suggestion system since our main idea comes from this system.

2.1 Recommendation System

Collaborative Filtering(CF) is a traditional research of recommendation system, which references the behaviors of a group of users to recommend or predict for other users. There is a survey paper [13] discussing many kinds of CF. In our opinion, modern research of recommendation systems is interested in finding new features to recommend those that have never been recommended before, especially temporal features. For example, Xiang *et al.* [16] proposed the Session-based Temporal Graph(STG) to describe user behaviors, which are determined by long-term and short-term preferences. Long-term preferences could be considered as frequency,

and short-term preferences could be considered as the relations between user behavior and time interval. Lathia *et al.* [6] proved that temporal diversity is an important feature for recommendation, since the user will rate items over time. Zheng and Li [21] provided a resource-recommendation model, which integrates both tag and tagging time information then uses CF for recommendation. The time information has a similar definition to *Recency* in this paper. Koren [5] built a time-factor model which is similar to [16], because the model considered both long-term and transient effects. Although they considered temporal features, they still not consider that user behaviors are not always related with time. There are some recommendation system studies which have tried to find other features, such as Trust Circles [20], Bayesian Network [18], Social Collaborative Filtering(SCF) [11], and the Nearest Neighbor based Top-K recommendation system [19].

2.2 Community Detection

Community detection in social networks is a well-studied problem. Leskovec *et al.* [7] compared eight network community detection methods originating from theoretical computer science, scientific computing, and statistical physics. Aggarwal *et al.* [1] explored a simple property of social network called *local succinctness property*, which is used to extract compressed descriptions of the underlying community representation of the social network with the use of a min-hash approach. Huang *et al.* [4] proposed a parameter-free algorithm SHRINK, which could not only discover overlapped and hierarchical communities but also the hub nodes and outliers among them. There are many other extensive studied literatures for community detection [10, 2, 9]. Although community detection is a similar problem with our problem, there is a key point of difference between both problems. In general, a community is considered as a group of people whose interactions within the group are more than outside the group. The detected community has equal relationship between each member in this community, i.e., the community is fixed for each member. However, the definition of group we emphasize is dynamic for different users. For example, the group of an user A contains user B and user C,

but the group of user B may not contains user A.

2.3 Relationship Link Prediction

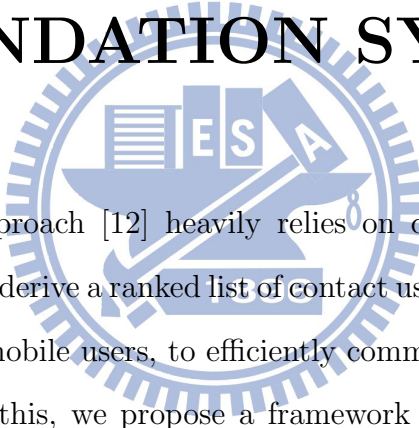
Friends suggestion is sometimes similar to relationship link prediction, because if we predict when a link would be built we guess that both linked nodes will possibly communicate in that time. Liben-Nowell and Kleinberg [8] formalized the link prediction problem and employed random walk methods to address this problem. Yang *et al.* [17] proposed FIP model bridges between Collaborative Filtering(CF) and link prediction to provide a unified treatment for interest targeting and friendship prediction. Sun *et al.* [14] built a relationship building time prediction model, which uses learning algorithms to fit different distributions and then gets a probability for building relationships between two nodes. But the edges are only constructed once, so we cannot use it for communication networks which change over time.

2.4 Friends Suggestion System

Our main idea is based on Roth *et al.* [12], who proposed a friends recommendation system for *Gmail* using group information and three criteria. *Gmail* is a well-known mail system constructed by *Google*, which may have many history records to retrieve for friends suggestion. However, the algorithm in [12] could not work effectively for sparse data, insufficient interaction history resulting in some recommendation lists to be empty. Moreover, *Time-Dependency* of user interactions is not addressed in their work. Bartel and Dewan [3] enhanced [12] with a hierarchical structure, which re-orders the recommendation list by ranking past communication group and hierarchically predicts next group. Wu *et al.* [15] proposed a interactive learning framework to formulate the problem of recommending patent partners into a factor graph model. Similarly, no attention has been paid to address the problem of *Time-Dependency* of user interactions.

Chapter 3

TIME-DEPENDENT CONTACTS RECOMMENDATION SYSTEM



Existing friends suggestion approach [12] heavily relies on query users to provide partial contact users (i.e., a seed set) to derive a ranked list of contact users. This increases inconvenience for query users, especially for mobile users, to efficiently communicate with multiple users at the same time. Motivated by this, we propose a framework for automatic time-dependent contacts recommendation without letting query users to provide prior information such as core contact users. The system framework overview is illustrated in Fig. 3.1. Our system consists of two phases: Seedset Generation and Friends Suggestion. Seedset Generation automatically derives a set of core users (referred to as seedset) with the highest probability to be contacted with the query user. Seedset Generation is achieved by mining frequent and time-dependent communication patterns from historical interaction logs. In Friends Suggestion phase, our system generates a ranked list of contact users based on the derived seedset. Once the query user chooses partial members from the list, our system updates the friend suggestion list by adding selected users to current seedset and then launching Friends Suggestion again to update the ranked list of contact users. This process continues until no more contact users can be suggested or the query user drops this session. Notice that our recommendation system

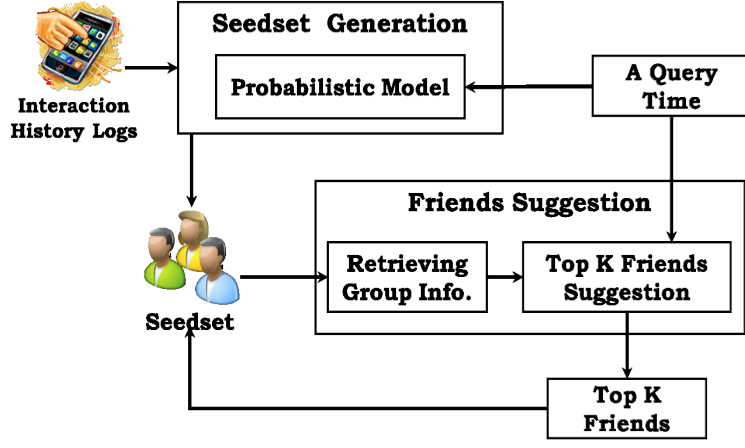


Figure 3.1: Framework Overview

provides a generic framework where the Friend Suggestion component can be replaced by other state-of-the-art algorithms to serve different requirements.

3.1 Temporal Probabilistic Model

When a query user attempts to share information (e.g., photos), the query user forms a list of contact users in his/her mind. After that, query users has to manually and sequentially select the list of contact users by scanning through their friends corpus. This brings lots of unnecessary efforts. To solve this problem, we first propose Seedset Generation that uses a temporal probabilistic model to predict possible contact users as seeds for Friends Suggestion.

We claim that *if a query user interacts with a contact user in a similar time interval, this infers that the query user has a higher probability to interact with the contact users in a similar time interval as well*. Fig. 3.2 shows the distribution of entropy for every contact user, where the x-axis is entropy and the y-axis is number of contact users. We can clearly see that there are three peaks in Fig. 3.2, which can be explained by the fact that the lower entropy has higher time centrality and vice versa. The first peak with entropy = 0 represents that their interactions centralize in one time slot, the second peak with entropy = 0.5 means the time points of interactions distributing in two time slots and the third peak with entropy = 0.75 is distributed in three slots. We observe that the number of contact users is over 60%

when entropy is no greater than 0.5, which means most of the contact users have higher time centrality for interactions. Therefore, this verifies our assumption that most interactions is highly time-dependent.

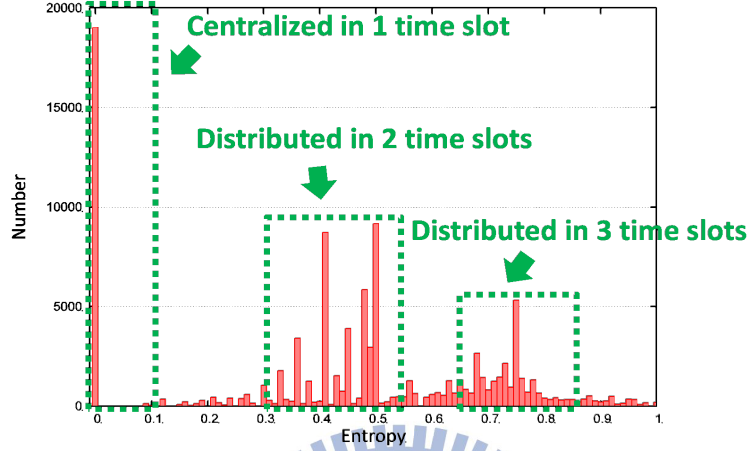


Figure 3.2: Distribution of Time Centrality

Our goal is to shift away mobile user's burden whenever they attempt to share information via their mobile devices. Specifically, we designed a temporal probabilistic model to predict a ranked list of contact users who are most likely to be interacted with the query user a given time point. There are three factors considered in developing our temporal probabilistic model:

1. Frequency: Receivers who have more interactions with the query user are more important than those who interact less with the query user.
2. Recency: More recent interactions should have more importance whereas older interactions decay over time.
3. Time-Dependency: If receivers always interact with the query user in a similar time interval, they should have more importance in that time interval.

Frequency is a straightforward yet effective measurement. Inspired from Interaction Rank [12], we unify *Frequency* with *Recency* into a single measurement as shown in Equation (3.1). [12] introduced a decaying parameter λ , to control the importance of every interaction according to its time. Namely, every interaction decays exponentially over time with a half

life λ . To fit the temporal probabilistic model, we form the two factors into a probability, which can be expressed as:

$$P(R_n) = \frac{\sum_{i \in I(R_n)} \left(\frac{1}{2}\right)^d}{\sum_{i \in I} \left(\frac{1}{2}\right)^d} \quad (3.1)$$

where $P(R_n)$ is the probability of the query user interacting with R_n in the past, I is a set of all the query user's interactions, and $I(R_n)$ is a set of all interactions between query user and R_n . d is a decay function which is expressed as $\frac{t_{now} - t_i}{\lambda}$, where t_{now} is the current time, t_i is the time of interaction $i \in I$, and a half-life parameter λ that assigns score 1 to an interaction at current time and decays the importance of an interaction to $\frac{1}{2}$ with the half-life λ .

To incorporate the third factor, *Time-Dependency*, we formulate a conditional probability as:

$$P(R_n|t) = \frac{P(R_n \cap t)}{P(t)}. \quad (3.2)$$

Equation (3.2) shows the probability of the query user interacting with R_n in a time interval t , where $P(R_n \cap t)$ and $P(t)$ can be derived by the following equations:

$$P(R_n \cap t) = \frac{\sum_{i \in (I(R_n) \cap I(t))} \left(\frac{1}{2}\right)^d}{\sum_{i \in I} \left(\frac{1}{2}\right)^d} \quad (3.3)$$

and

$$P(t) = \frac{\sum_{i \in I(t)} \left(\frac{1}{2}\right)^d}{\sum_{i \in I} \left(\frac{1}{2}\right)^d} \quad (3.4)$$

where $I(t)$ is a set of the query user's interactions in time interval t .

To take into the following three factors into consideration, *Frequency*, *Recency* and *Time-Dependency*. Intuitively, we combine $P(R_n)$ and $P(R_n|t)$ by a linear combination with a tunable parameter α , which can be formulated as follows:

$$Score(R_n) = (1 - \alpha)P(R_n) + \alpha \cdot P(R_n|t) \quad (3.5)$$

where α is the weight of *Time-Dependency* and the range of α is between 0 and 1. In

general, Equation (3.5) does not make sense, because when a candidate receiver R_n has higher $P(R_n)$ and also has higher $P(R_n|t)$, it should be chosen with more chances. When both probabilities are not relative to each other, we should think about other methods to merge them. Calculating the mean between $P(R_n)$ and $P(R_n|t)$ is a good idea to balance Equation (3.5), since it considers the influence from not only specific time intervals but also all time intervals. We adjust Equation (3.5) by using two types of mean, arithmetic mean and geometric mean, and thus both equations can be expressed as follows:

$$Score_{ari}(R_n) = (1 - \alpha)P(R_n) + \alpha \cdot \frac{P(R_n) + \omega \cdot P(R_n|t)}{1 + \omega} \quad (3.6)$$

and

$$Score_{geo}(R_n) = (1 - \alpha)P(R_n) + \alpha \cdot \sqrt[1+\omega]{P(R_n)(P(R_n|t))^\omega} \quad (3.7)$$

where ω represents the weight of a specific time interval. Notice that we only use geometric mean in our experiment since we find that geometric mean makes more sense for our assumption: *if one of $P(R_n)$ and $P(R_n|t)$ is much lower than the other, their mean should be closer to the lower one.*

For refining Equation (3.7), we want to decide which value of α is the best by observing data. In our observations, we find that not all receivers have high time-dependency, as some are independent of a specific time interval. In other words, receivers have different time-dependencies in different time intervals, and time-dependencies will vary from person to person. To achieve this, we change α to another conditional probability which is expressed by:

$$P(t|R_n) = \frac{P(R_n \cap t)}{P(R_n)}. \quad (3.8)$$

Equation (3.8) indicates the probability of R_n interacting with the query user in time interval t . If $P(t|R_n)$ is higher, R_n has a higher time-dependency with the query user and vice versa. We then utilize Z-score to make the importance of time-dependency be numerical, the formulation

of Z-score is

$$Z(R_n) = \frac{P(t|R_n) - \text{avg}(P(t|R))}{\sigma(P(t|R))} \quad (3.9)$$

where $\text{avg}(P(t|R))$ is the average probability of all $R_n \in R$ and $\sigma(P(t|R))$ is the standard deviation. Since Z-score may be negative, we normalize Z-score by considering the central point, namely, we normalize Z score within $[-3, 3]$ to $[0, 1]$. If the Z-score is more than 3 or less than -3, it will be considered as 1.0 or 0.0. Therefore, we can reformulate Equation (3.7) as follows:

$$\begin{aligned} \text{Score}_{final}(R_n) = & (1 - NZ(R_n)) \cdot P(R_n) \\ & + NZ(R_n) \cdot \sqrt[1+\omega]{P(R_n)(P(R_n|t))^\omega} \end{aligned} \quad (3.10)$$

where $NZ(R_n)$ is the normalized Z-score and the range is from 0 to 1.

This temporal probabilistic model has been used in Seedset Generation and we will discuss how to use it in the next subsection.

3.2 Seedset Generation

Seedset Generation phase derives a set of core contact users who are highly to be the receivers with related to the query user at given time. In a sense, Seedset Generation can serve as a Friends Suggestion in a special case when query users intend to communicate with a single user instead of a group of users. In that case, Seedset Generation phase returns the potential receivers as a top- k list of users. Without specific groups information, Seedset Generation adopts the introduced temporal probabilistic model mentioned before predicting which contact users in the past are most likely the receivers, merely based on specified query time. The algorithm of Seedset Generation can be summarized in Algorithm 1.

Algorithm 1: Seedset Generation Algorithm

Input: Query user's history interactions I and current time interval t

Output: The seed generations S

```
1  $S = \phi$ ;  
2 foreach  $i \in I$  do  
3   Sum scores of  $i$  for temporal probabilistic model;  
4    $C = \text{GetContactUser}(i)$ ;  
5   foreach  $c \in C$  do  
6     if  $c \notin S$  then  
7       Put  $c$  into  $S$ ;  
8     end  
9   end  
10 end  
11 foreach  $c \in S$  do  
12   Calculate all probabilities  $P(c)$ ,  $P(t)$ ,  $P(c|t)$  and  $P(t|c)$ ;  
13    $S[c] = \text{Score}_{\text{final}}(c)$ ;  
14 end
```

3.3 Friends Suggestion

Friends Suggestion is a function which can be applied to any seed-based suggestion approach. In this subsection, we propose an enhanced suggestion approach, Friends Suggestion, with our temporal probabilistic model into state-of-the-art ranking models [12].

Our temporal probabilistic model can be combined with *Gmail Approach* [12] which Roth *et al.* proposed to become an enhanced suggestion approach. They considered three factors, *Frequency*, *Recency* and *Direction*, and we consider one additional factor, *Time-Dependency*. Their score function named *Interaction Rank* is:

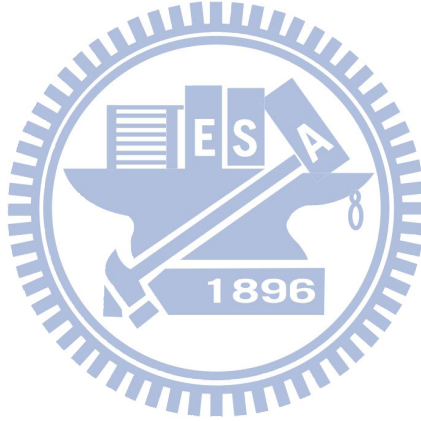
$$\mathcal{IR}(g) = \theta_{out} \sum_{i \in I_{out}(g)} \left(\frac{1}{2}\right)^d + \sum_{i \in I_{in}(g)} \left(\frac{1}{2}\right)^d \quad (3.11)$$

where $I_{out}(g)$ is the set of outgoing interactions between a query user and a group, $I_{in}(g)$ is the set of incoming interactions and θ_{out} is the weight of outgoing interactions to represent *Direction*. According to Equation (3.11), we can modify Equation (3.1) as the following

equation:

$$P(R_n) = \frac{\theta_{out} \sum_{i \in I_{out}(R_n)} (\frac{1}{2})^d + \sum_{i \in I_{in}(R_n)} (\frac{1}{2})^d}{\theta_{out} \sum_{i \in I_{out}} (\frac{1}{2})^d + \sum_{i \in I_{in}} (\frac{1}{2})^d}. \quad (3.12)$$

Equations (3.3) and (3.4) have similar transformed equations, too. The scoring function in *Gmail Approach* we adopt is *Intersection Weighed Score*, which considers the intersection of group and seedset and uses them to weight the score of the group. As reported in [12], *Interaction Weighted Score* achieves the best performance among their proposals. We compare *Gmail Approach* and our approach in Section 5.5.



Chapter 4

TIME INTERVAL ADJUSTMENT

Considering the following scenario: *A user A has a regular behavior to call user B after user A finishes his works and the time is always between 5:00pm to 6:00pm. One day, user A has finished his works early at 3:30pm and he calls user B immediately.* There is an issue raised in this scenario, should the interaction at 3:30pm be considered to the previous interactions in suggesting friends? To address above problem, we propose two approaches to optimize the time interval. The main idea is to analyze the interaction time distribution in one day and then decide an optimal time interval to describe the interaction behaviors.

4.1 Entropy Examination

First, we utilize entropy as a measurement to determine which time interval is most appropriate. A narrow time interval means the behavior is regular and a broad time interval represents the accidents occur frequently. Therefore, we split 24 hours into 24 time slots with the number of interactions in those time slots and then calculate the entropy. If the entropy is lower than a threshold, which means the level of regularity is higher enough, we would choose $(h - 1)/2$ as the optimal time interval, where h is the length of each time slot. Otherwise, we continue to split 24 hours into 16, 12, 8, 6 or 4 time slots until the entropy is lower than a threshold or no

more entropy could be calculated. Algorithm 2 summarizes the idea. In our experiment, we set the entropy threshold δ to 0.5. When entropy is 0.5, user behaviors centralize in half when the time slots is 4 and it can be the least time centrality if the user behaviors are centralized.

Algorithm 2: Entropy Examination

Input: Time Distribution in 24 hours D and Entropy Threshold δ

Output: Optimal Time Interval T

```

1  $T = 0$ ;
2 foreach number of time slots:  $i$  do
3    $Entropy = EntropyCal(D, i)$ ;
4   if  $Entropy \leq \delta$  then
5      $T = ((24/i) - 1)/2$ ;
6     break;
7   end
8 end

```

4.2 Close Peak Detection

To detect the close peak, we need only to know the trends between each time slot. The goal is to find the cluster contains the current time slot and then we can choose this cluster as optimal time interval. First, we consider the trend between two adjacent time slots. Larger number of interactions time slot should be less or equal τ times than smaller number of interactions time slot, where τ is a threshold for clustering time slots. Otherwise, the detection would be terminated and the final cluster has been determined. Algorithm 3 describes Close Peak Detection in detail.

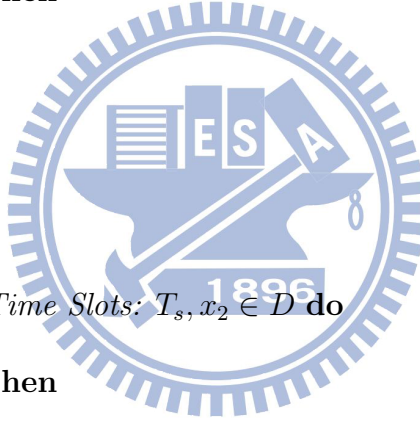
Both approaches are designed to infer the time interval dynamically for different persons, since the user behaviors are personalized on the mobile social media. Unfortunately, this problem is difficult to address multiple interactions suggestion, such as mailing behaviors. Mailing behaviors refer to the action of mailing to a group, where the receivers are more than one person, some receivers may have time-dependent behaviors but the others may not. It is impossible to define whether this interaction is time-dependent or not, so we evaluate the

Algorithm 3: Close Peak Detection

Input: Time Distribution in 24 hours D , Current Time h and Threshold τ

Output: Time Interval Start T_s and Time Interval End T_e

```
1  $T_s = h$ ;  
2  $T_e = h$ ;  
3 foreach Clockwise Time Slots:  $T_e, x_2 \in D$  do  
4   if  $p(x_2) > p(T_e)$  then  
5     if  $p(x_2) \leq \tau * p(T_e)$  then  
6        $T_e = x_2$ ;  
7     end  
8     else  
9       break;  
10    end  
11  end  
12  else if  $p(x_2) < p(T_e)$  then  
13    if  $p(T_e) \leq \tau * p(x_2)$  then  
14       $T_e = x_2$ ;  
15    end  
16    else  
17      break;  
18    end  
19  end  
20 end  
21 foreach Counterclockwise Time Slots:  $T_s, x_2 \in D$  do  
22   if  $p(x_2) > p(T_s)$  then  
23     if  $p(x_2) \leq \tau * p(T_s)$  then  
24        $T_s = x_2$ ;  
25     end  
26     else  
27       break;  
28     end  
29   end  
30   else if  $p(x_2) < p(T_s)$  then  
31     if  $p(T_s) \leq \tau * p(x_2)$  then  
32        $T_s = x_2$ ;  
33     end  
34     else  
35       break;  
36     end  
37   end  
38 end
```



above methods by single interaction dataset like calling behaviors. However, optimizing the time interval is an important task for classifying time-dependent or time-independent users.



Chapter 5

EXPERIMENT

In this section, we evaluate the quality of proposed framework and analyze a series of observations using three real datasets. Section 5.1 describes basic information of three real datasets which we use on our evaluations and observations. Section 5.2 presents the analysis results of the inference of time centrality before we show the effective experimental results. Section 5.3 shows the experiment setups of three real datasets, which include splitting the training dataset and the testing dataset and parameter settings. Section 5.4 introduces two metrics, nDCG and hit rate, to evaluate the quality of proposed framework: Seedset Generation and Friends Suggestion. Finally, Section 5.5 and 5.6 show the experimental results.

5.1 Data Description

In our experiment, we use three real datasets, Enron Mail¹, call detail records(cdr) from Chunghwa Telecom(CHT)² and Reality Mining Dataset(RMD) from MIT³. The basic information of each dataset is shown in Table 5.1, where Enron Mail contains multiple interaction data and the others only contains single interaction data. In other words, we adopt Enron Mail

¹The Enron Mail data can be downloaded from <http://www.cs.cmu.edu/~enron/>.

²The CHT data is not in public, and Chunghwa Telecom's website is <http://www.cht.com.tw/>.

³The Reality Mining Dataset can be downloaded from <http://realitycommons.media.mit.edu/realitymining4.html>

Table 5.1: Basic Information on the Enron/CHT/RMD Datasets

element	Enron	CHT	RMD
#user	65,182	76,263	92
#interactions	236,505	2,443,667	78,110
#group interactions	67,631	-	-
time	1998/01/04 - 2002/12/21	2010/08	2004/01/19 - 2005/07/15

in both Seedset Generation and Friends Suggestion evaluations, and the others only in the Seedset Generation evaluation. We also adopt all datasets in *Time-Dependency* evaluation, which is discussed in the next subsection.

In general, social behavior can be described by basic factors such as frequency, recency, and time-dependency can describe most of user behaviors. Most of users tend to share something to or call someone who has more interactions before, recently or at the specific time intervals. Therefore, we use calling behavior and mailing behavior datasets to simulate general social behavior dataset since some datasets like sharing behavior dataset is difficult to obtain.

5.2 Analysis of Time Centrality

Table 5.2: Four slots of Time in 24 hours

Part I	Part II	Part III	Part IV
00:00 - 05:59	06:00 - 11:59	12:00 - 17:59	18:00 - 23:59

In time centrality experiment, we set some constraints for choosing test query users and their contact users. There are 9,036 test query users and they have totally 102,405 contact users in Enron Mail, 3,860 test query users and totally 29,167 contact users in CHT cdr and 88 test query users and 2,564 contact users in RMD. The constraints are the contact users should interact with test query users at least 4 times, since we split the time in one day into four time slots which is shown in Table 5.2. It is similar for splitting the time in one day into six, eight, twelve and twenty-four time slots.

Fig. 5.1 to Fig. 5.5 show the time centrality of three real datasets with different time slots. Since Enron Mail is a mailing behavior dataset and the other datasets are calling

behavior datasets, its distribution is a little different from CHT and RMD. According to this observation, we find that Enron Mail has higher time centrality because its entropy is relatively lower than that of CHT and RMD. Therefore, we conclude that mailing behavior is regular for the same receiver, i.e., the most mailing behaviors of the user tend to send their mail to the same receiver at particular time points. Unlike mailing behavior, calling behavior does not show strong time centrality. For CHT and RMD, it is hard to say which one has higher time centrality than the other, but both datasets have similar distribution over time slots. The calling behaviors are almost the same in two different datasets and they distribute on the middle entropy, i.e., 0.5. It is easy to explain when the entropy is 0.5, the users call callees not only at the same time slot but also at the adjacent time slots. In other words, the regular calling behavior may shift to the temporally close time points if there are some accidents for the users.

Fig. 5.6 to Fig. 5.10 show the cumulative distribution function of three dataset with different number time slots. These figures are more clear for analyzing than PDF.

5.3 Experimental Setup

For Enron Mail, we chose 21,262 mails from Enron Mail to be the testing data and extracted 30 days before testing data to be the training data, where the rule in selecting testing data is as follows: (1) the mail should be sent to at least 2 receivers, or a *group*, and (2) the sender of the mail had sent no less than 4 mails before. For CHT cdr, which is a single interaction data, they do not have group information, since CHT cdr consists of cell phone call records and we only need to guess one person that the query user wants to call. We chose 30,295 cdr from CHT cdr to be the testing data and extracted 30 days before testing data to be the training data. The testing data is all in the last day in CHT cdr. RMD has the same properties with CHT cdr. We chose 44,166 records from RMD to be the testing data and extracted 30 days before testing data to be the training data.

Table 5.3: Parameter Settings

Parameter	Enron	CHT	RMD
λ	7 days	3 days	3 days
θ_{out}	5	-	-
ω	1	1	1
<i>Time Interval</i>	1 hour	1 hour	1 hour

Parameter settings are shown in Table 5.3, where λ is a time decay parameter, θ is an outlink weight parameter, ω is a time dependency parameter and *Time Interval* is how many additional hours we reference next to the current hour.

5.4 Metrics

For the following experiments, we adopt normalized discounted cumulative gain(nDCG) and hit rate as the measurements to evaluate. nDCG evaluates not only the precision but also the ranking of the results. Hit rate is a base measurement since we only need hitting one correct answer for some scenarios, like dynamic phonebooks and other single interaction suggestions.

DCG measures the *gain* of a hit result based on its rank in the list, where the top rank has more gain and the lower rank has less gain, although it is also hit. The formulation of DCG we use is

$$DCG_p = rel_1 + \sum_{i=2}^p \frac{rel_i}{\log_2(i)} \quad (5.1)$$

where $rel_i \in \{0, 1\}$ is a relevance value and p is the length of the result list. nDCG is to normalize DCG so that we can compare result lists with different lengths. The formulation of nDCG is expressed by

$$nDCG_p = \frac{DCG_p}{IDCG_p} \quad (5.2)$$

where IDCG is ideal DCG which is calculated by the ideal rank of the result list.

Hit rate identifies the contacts of the result list have been matched the real answer for each case, so this metric should be used for single interaction suggestions. Base on the property of hit rate, we evaluate all datasets exclude Enron Mail by hit rate. The formulation of hit rate

is

$$HitRate = \frac{NumberofHit}{NumberofTestingCase} \quad (5.3)$$

where the definition of *Hit* is the generated list contains the correct answer from the testing case.

Table 5.4: Four Methods for Comparison

Method	Symbol
Frequency	F
Frequency + Recency	FR
Frequency + Time-Dependency	FT
All factors	All
Recent Logs	RecentLog

5.5 Probabilistic Model Results

Evaluation on Seedset Generation: Figure 5.11 and Figure 5.12 shows the impact of each fundamental factor: Frequency, Recency and Time-dependency on Seedset Generation quality. We want to prove that our temporal probabilistic model is effective, so we separate the three factors into four methods for Seedset Generation and compare them. We also compare with RecentLog which directly generates the recommendation list in order by the recent contacts. Each method is summarized and denoted in Table 5.4.

In Fig. 5.11(a), the x-axis is top- k ($1 \leq k \leq 30$) and the y-axis is the value of nDCG. The red line (F) that only considers frequency to determine the friend group is considered as a baseline. The pink line (All) is our proposal which considers all factors and outperforms other models with at most 4.2% increase in accuracy compare to the baseline.

In Fig. 5.11(b), the x-axis is the top- k , ranging from 1 to 10, and the y-axis is the nDCG value. It is worth mentioning that the lines assemble the log-likelihood, because CHT cdr only has one receiver for recommendation in each record. Our proposal outperforms other models with 26% increase in accuracy compare to the baseline when k is 5. Since CHT cdr can be considered as single interaction dataset, we also evaluate it by hit rate. The results

are shown in Fig. 5.12(a), our proposal outperforms other models with 22% increase to the baseline when k is 5. It is easy to observe that when k is 5 the hit rate can achieve over than 98%.

The similar results could be found on RMD. Fig. 5.11(c) shows the nDCG comparison for those four models and our proposal outperforms other models again with 16% increase in accuracy to the baseline when k is 5. Fig. 5.12(b) shows hit rate comparison and our proposal increase 21% to the baseline when k is 5. In addition to those results, we observe that the blue line (FT) just increase a little to the red line (F) in both figures, but it increases more on CHT cdr. To illustrate, we analyze time centrality for both CHT and RMD dataset as shown from Fig. 5.1 to Fig. 5.5, they show the difference between CHT and RMD. Although the distributions of both datasets are similar, RMD is more centralized because its average entropy is smaller than CHT. When time interval is fixed, the users in the more centralized dataset can not be separated well since all of those users are considered as time-dependent users. We discuss about dynamic and fixed time interval in Section 5.6.

Based on above results, we conclude that our temporal probabilistic model presents consistent improvement than straightforward suggestion such as frequency or recency.

Evaluation on Friends Suggestion: Fig. 5.13 shows the performance comparison of Gmail Approach [12] and our proposed Probabilistic Suggestion Approach(PSA). Since Gmail Approach is a seed-based suggestion approach, we use Seedset Generation to generate a seedset with $k = 3$ and pass the input to Gmail and PSA, respectively. The final recommendation list contains the seeds which are different from the original Gmail Approach, but it will not affect the recommendation result because the seeds appears at the top of the list and they are also uncertain receivers for the query user. We use the same test data from Enron Mail as in Fig. 5.11(a). In Fig. 5.13, the x-axis is top- k ($1 \leq k \leq 30$) and the y-axis is the nDCG value, where the red line is Gmail Approach and the green line is PSA. We can see that no matter in what situation, PSA always has higher nDCG than Gmail Approach.

Analysis of Time-Dependency: Finally, we provide an empirical insight of proposed framework. We choose the hit receivers, which are the correct answers for the test letters,

from two different methods *All factors* and *Frequency* to calculate their time centrality. The approach for calculating time centrality is the same as Fig. 3.2. Fig. 5.14 shows that *All factors* can recommend more correct contact users because they shows higher time centrality (i.e., having more receivers whose entropy equal or less than 0.5).

5.6 Time Interval Adjustment Results

Table 5.5: Symbols of Time Interval Adjustment Methods

Method	Symbol
Fixed Time Interval	Fix
Entropy Examination	EE
Close Peak Detection	CPD

Fig. 5.15 shows the comparison between fixed time interval and our proposed methods. As mentioned previously, we have two proposed methods to optimize the time interval, Entropy Examination and Close Peak Detection. Both methods have unique parameter to process well, they are δ in Entropy Examination and τ in Close Peak Detection. We set δ to 0.5 and τ to 2 because they are average numbers for those methods. In fixed time interval, we fixed the time interval to 1 hour because it results highest nDCG from other fixed time intervals. Each method is summarized and denoted in Table 5.5.

There is a weird result in Fig. 5.15. Fig. 5.15(a) shows three methods comparison in CHT cdr. The x-axis is top- k and the y-axis is the nDCG. CPD has the lowest nDCG among the three methods, and it has 4% decrease in accuracy compare to Fix when k is 3. EE outperforms the other methods, and it has 5% increase in accuracy compare to Fix when k is 3. Fig. 5.15(b) shows the methods comparison in RMD, and the x-axis and y-axis are the same with Fig. 5.15(a). But in Fig. 5.15(b), the worst method among the three methods is Fix. CPD is no longer the worst method and becomes an useful method with 17% increase in accuracy compare to Fix when k is 3. To explain this situation we still analyze the Fig. 5.1 to Fig. 5.5. As mentioned before in Section 5.5, the fixed time interval will make the more centralized dataset like RMD has less improvement with our probabilistic model, so we apply

the CPD on RMD can address this problem. Although utilizing CPD can classify the users in RMD well, CPD will interference the less centralized dataset like CHT since the nDCG decrease in Fig. 5.15(a). However, EE outperforms the other methods and it has 24% increase in accuracy compare to Fix when k is 3. EE always outperforms the other methods no matter in CHT or RMD and will not interference the dataset to lead to decrease the accuracy.

Finally, we conclude that EE is the best methods for our datasets. We know that the effects of these methods depend on the property of datasets since our proposed methods are naive to address this kind of problem. Our goal is to analyze the difference between dynamic and fixed time interval. In our opinion, to make the time interval dynamically is an useful solution for time-dependent contacts recommendation.



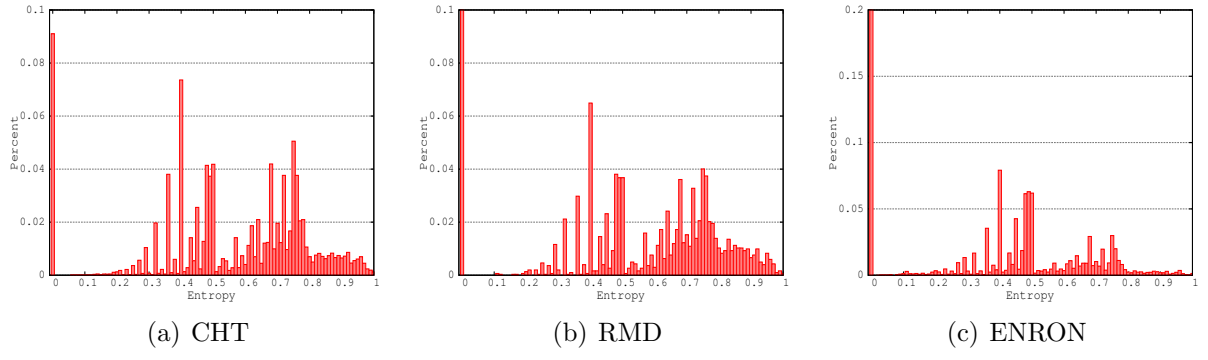


Figure 5.1: Probability Distribution Function of Time Centrality with 4 time slots in (a) CHT cdr (b) RMD (c) Enron.

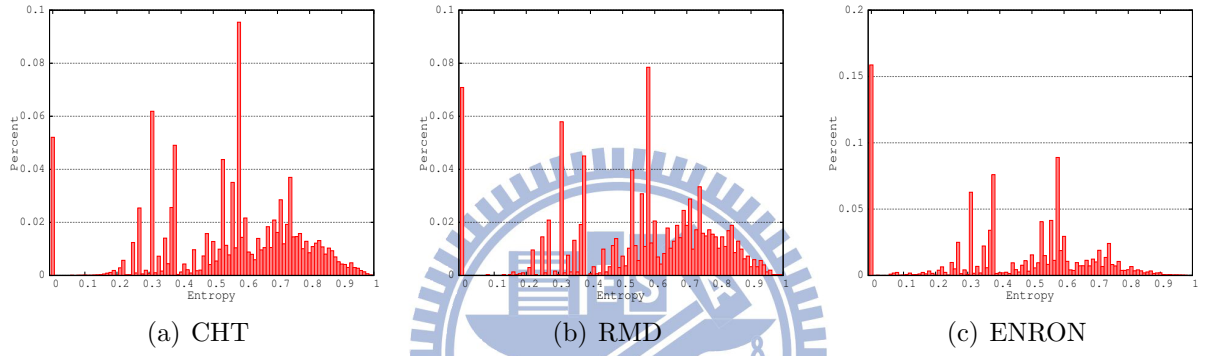


Figure 5.2: Probability Distribution Function of Time Centrality with 6 time slots in (a) CHT cdr (b) RMD (c) Enron.

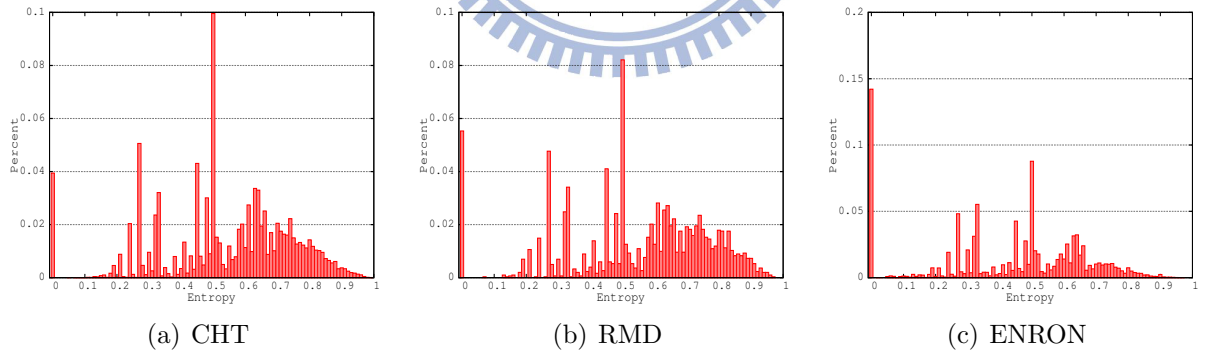


Figure 5.3: Probability Distribution Function of Time Centrality with 8 time slots in (a) CHT cdr (b) RMD (c) Enron.

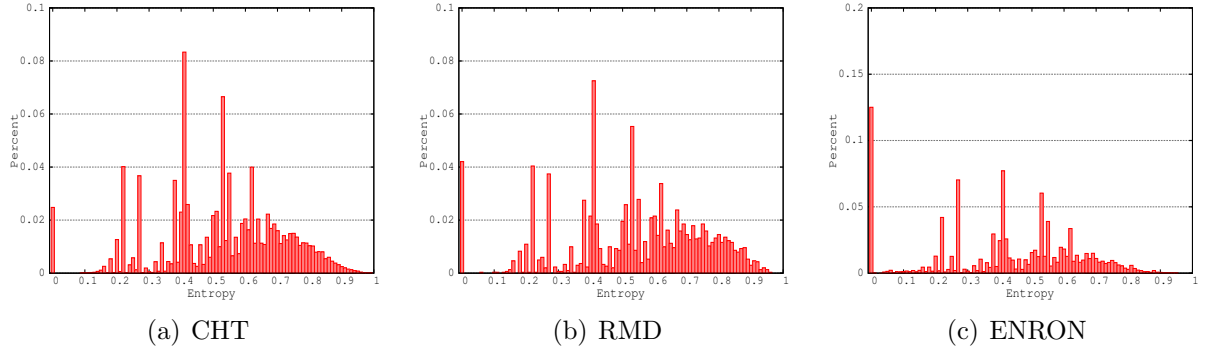


Figure 5.4: Probability Distribution Function of Time Centrality with 12 time slots in (a) CHT cdr (b) RMD (c) Enron.

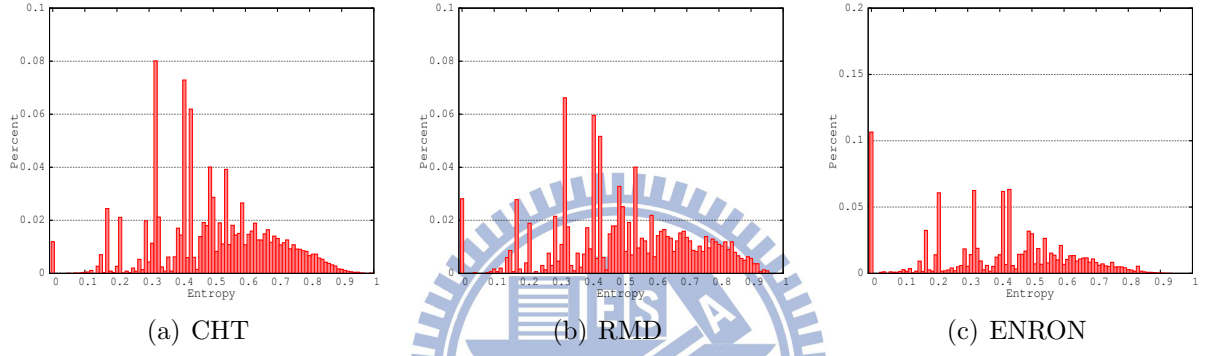


Figure 5.5: Probability Distribution Function of Time Centrality with 24 time slots in (a) CHT cdr (b) RMD (c) Enron.

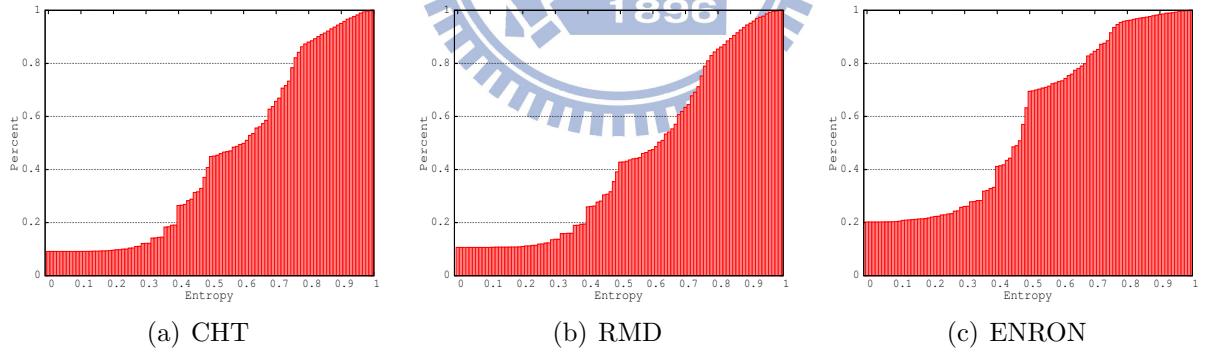


Figure 5.6: Cumulative Distribution Function of Time Centrality with 4 time slots in (a) CHT cdr (b) RMD (c) Enron.

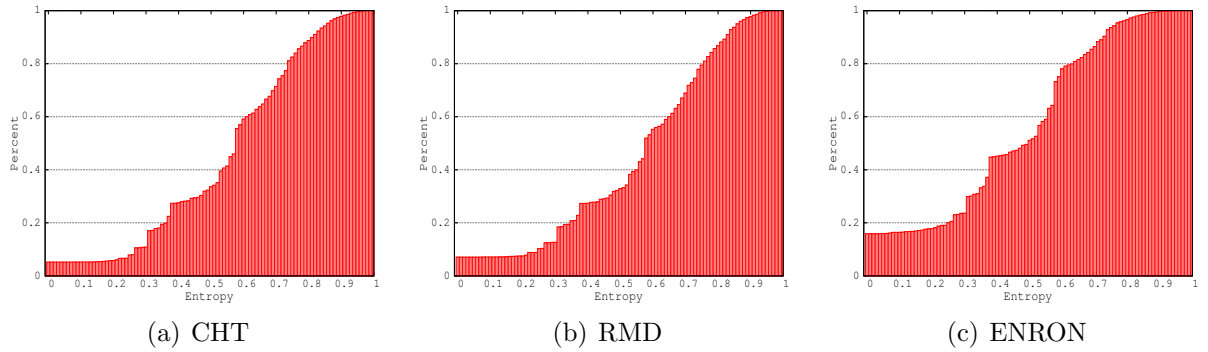


Figure 5.7: Cumulative Distribution Function of Time Centrality with 6 time slots in (a) CHT cdr (b) RMD (c) Enron.

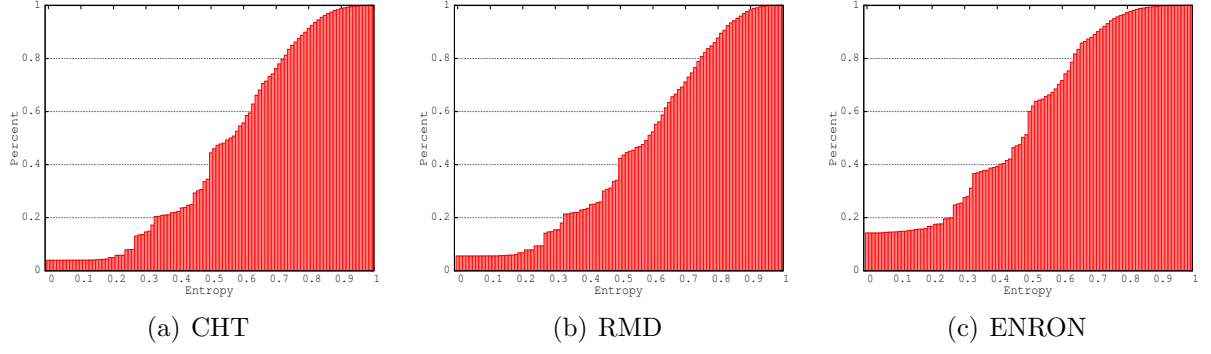


Figure 5.8: Cumulative Distribution Function of Time Centrality with 8 time slots in (a) CHT cdr (b) RMD (c) Enron.

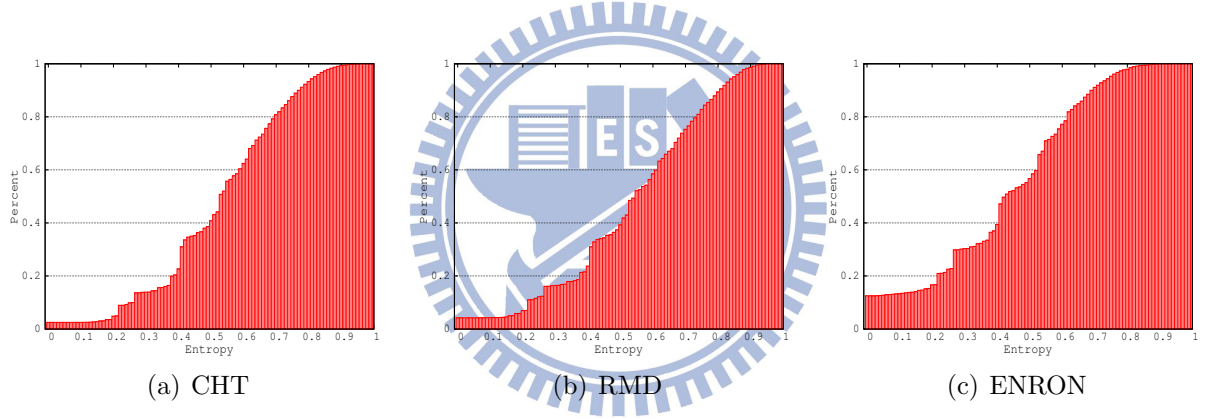


Figure 5.9: Cumulative Distribution Function of Time Centrality with 12 time slots in (a) CHT cdr (b) RMD (c) Enron.

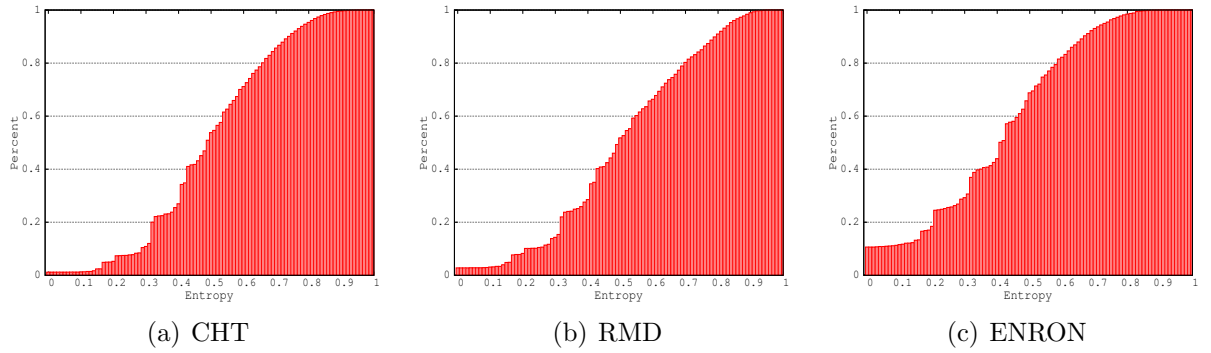


Figure 5.10: Cumulative Distribution Function of Time Centrality with 24 time slots in (a) CHT cdr (b) RMD (c) Enron.

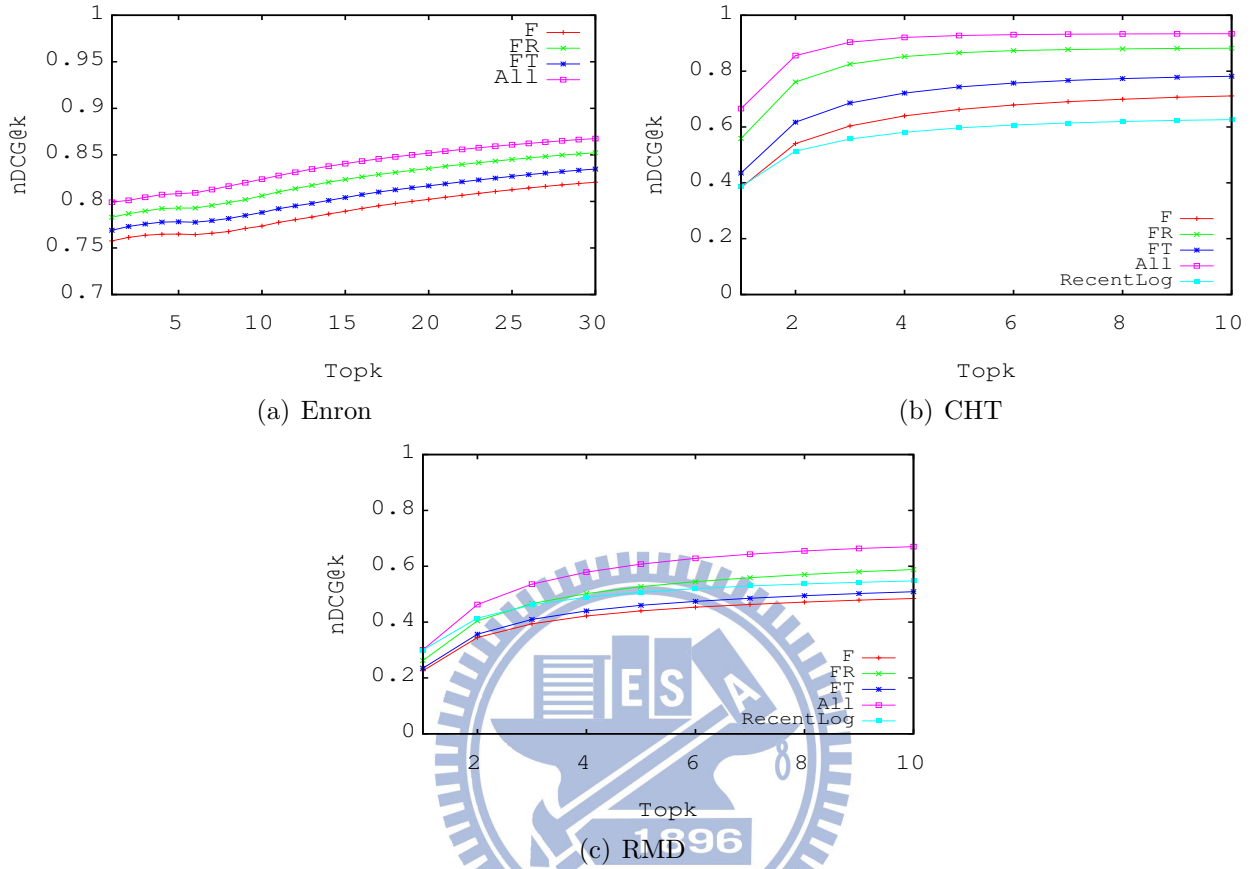


Figure 5.11: nDCG Comparison for Seedset Generation in (a) Enron Mail (b) CHT cdr (c) RMD.

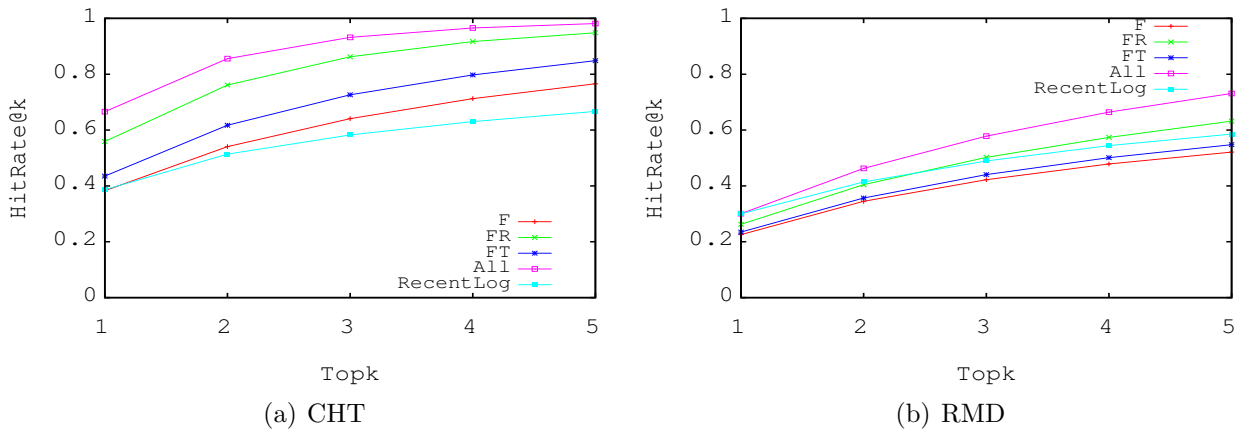


Figure 5.12: HitRate Comparison for Seedset Generation in (a) CHT cdr (b) RMD.

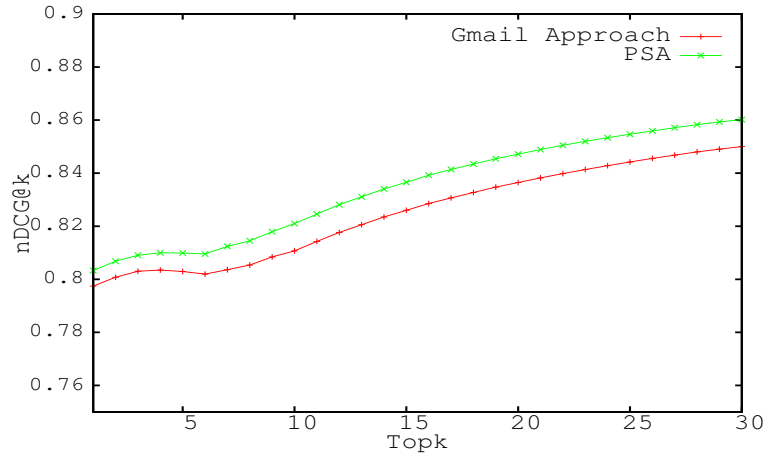


Figure 5.13: Comparison of Gmail Approach and Probabilistic Suggestion Approach(PSA) for Friends Suggestion

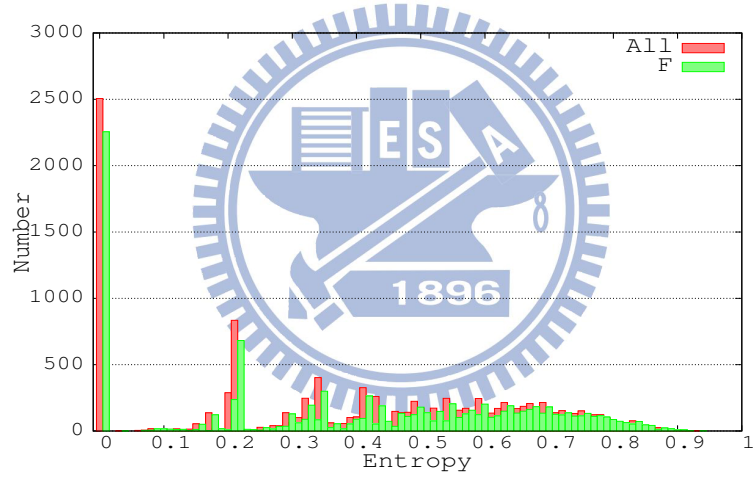
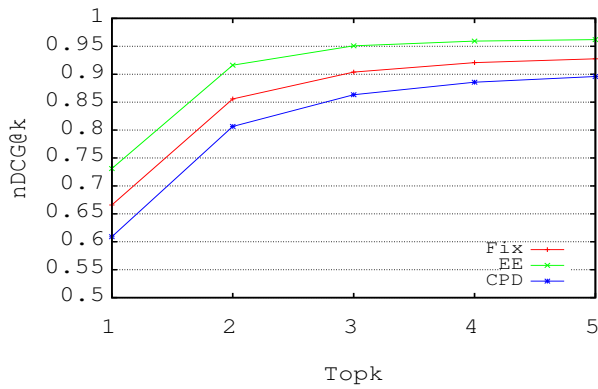
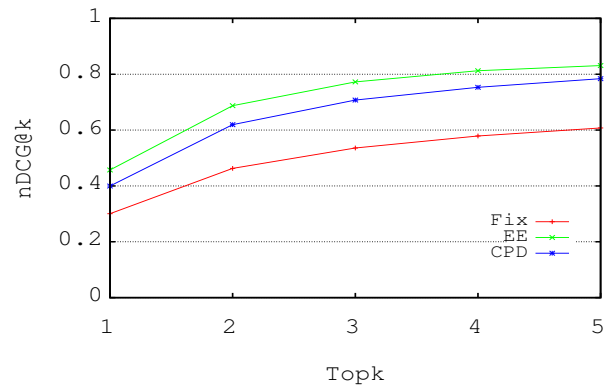


Figure 5.14: Comparison of Entropy Distribution



(a) CHT



(b) RMD

Figure 5.15: nDCG Comparison between dynamic and fixed time interval in (a) CHT cdr (b) RMD.

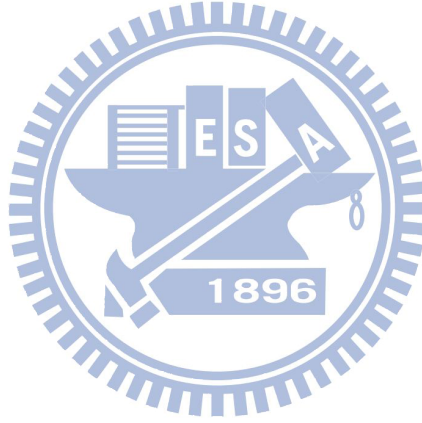
Chapter 6

CONCLUSION

In this paper, we study the problem of suggesting friends by implicit social graph and temporal importance. How to combine all the features we mentioned poses great challenges. In this paper, we propose a temporal probabilistic model to combine three factors, *Frequency*, *Recency* and *Time-Dependency*, and give them a dynamic weight which considers the problem, *different contact users have different importance of time for a query user*. Based on the temporal probabilistic model, we have designed a dynamic circle recommendation system which has two functions, Seedset Generation and Suggestion Approach. Seedset Generation can generate a set of seeds automatically by adopting the temporal probabilistic model and let the set of seeds be input to the next function. Suggestion Approach is flexible for embedding any seed-based suggestion approach, so we constructed our suggestion approach, the Probabilistic Suggestion Approach(PSA), which applies the temporal probabilistic model and considers an additional feature, *Direction* of interactions. We enhance the probabilistic model by discussing how to determine the time interval, which is a parameter to decide whose interactions are considered as time-dependent interactions. According to this, we propose two methods to determine the most appropriate time interval for user.

We prove that *Time-Dependency* is a critical information for suggesting friends and our experiment results also verify it. Our experiment results show that our proposed temporal

probabilistic model and time-dependent contacts recommendation system are effective in three real datasets, which are Enron Mail, CHT call detail records and Reality Mining Dataset. Although we have evaluated our temporal probabilistic model using emails and phone calls, it can be applied to any interaction-based social network. Future work will include the extension of our temporal probabilistic model such as dynamically deciding the number of seeds and using user clusters. We are also interested in exploring other applications such as content-based sharing and temporal community detection.



Bibliography

- [1] Charu C Aggarwal, Yan Xie, and S Yu Philip. Towards community detection in locally heterogeneous networks. In *SDM*, pages 391–402, 2011.
- [2] Yong-Yeol Ahn, James P Bagrow, and Sune Lehmann. Link communities reveal multiscale complexity in networks. *Nature*, 466(7307):761–764, 2010.
- [3] J. Bartel and P. Dewan. Towards hierarchical email recipient prediction.
- [4] Jianbin Huang, Heli Sun, Jiawei Han, Hongbo Deng, Yizhou Sun, and Yaguang Liu. Shrink: a structural clustering algorithm for detecting hierarchical communities in networks. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 219–228. ACM, 2010.
- [5] Y. Koren. Collaborative filtering with temporal dynamics. *Communications of the ACM*, 53(4):89–97, 2010.
- [6] N. Lathia, S. Hailes, L. Capra, and X. Amatriain. Temporal diversity in recommender systems. In *Proceedings of the 33rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR10)*, pages 210–217, 2010.
- [7] Jure Leskovec, Kevin J Lang, and Michael Mahoney. Empirical comparison of algorithms for network community detection. In *Proceedings of the 19th international conference on World wide web*, pages 631–640. ACM, 2010.

- [8] David Liben-Nowell and Jon Kleinberg. The link-prediction problem for social networks. *Journal of the American society for information science and technology*, 58(7):1019–1031, 2007.
- [9] Wangqun Lin, Xiangnan Kong, Philip S Yu, Quanyuan Wu, Yan Jia, and Chuan Li. Community detection in incomplete information networks. In *Proceedings of the 21st international conference on World Wide Web*, pages 341–350. ACM, 2012.
- [10] Yu-Ru Lin, Jimeng Sun, Paul Castro, Ravi Konuru, Hari Sundaram, and Aisling Kelliher. Metafac: community discovery via relational hypergraph factorization. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 527–536. ACM, 2009.
- [11] J. Noel, S. Sanner, K.N. Tran, P. Christen, L. Xie, E.V. Bonilla, E. Abbasnejad, and N. Della Penna. New objective functions for social collaborative filtering. In *Proceedings of the 21st international conference on World Wide Web*, pages 859–868. ACM, 2012.
- [12] M. Roth, A. Ben-David, D. Deutscher, G. Flysher, I. Horn, A. Leichtberg, N. Leiser, Y. Matias, and R. Merom. Suggesting friends using the implicit social graph. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 233–242. ACM, 2010.
- [13] X. Su and T.M. Khoshgoftaar. A survey of collaborative filtering techniques. *Advances in Artificial Intelligence*, 2009:4, 2009.
- [14] Y. Sun, J. Han, C.C. Aggarwal, and N.V. Chawla. When will it happen?: relationship prediction in heterogeneous information networks. In *Proceedings of the fifth ACM international conference on Web search and data mining*, pages 663–672. ACM, 2012.
- [15] Sen Wu, Jimeng Sun, and Jie Tang. Patent partner recommendation in enterprise social networks. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 43–52. ACM, 2013.

- [16] L. Xiang, Q. Yuan, S. Zhao, L. Chen, X. Zhang, Q. Yang, and J. Sun. Temporal recommendation on graphs via long-and short-term preference fusion. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 723–732. ACM, 2010.
- [17] Shuang-Hong Yang, Bo Long, Alex Smola, Narayanan Sadagopan, Zhaohui Zheng, and Hongyuan Zha. Like like alike: joint friendship and interest propagation in social networks. In *Proceedings of the 20th international conference on World wide web*, pages 537–546. ACM, 2011.
- [18] X. Yang, Y. Guo, and Y. Liu. Bayesian-inference based recommendation in online social networks. In *INFOCOM, 2011 Proceedings IEEE*, pages 551–555. IEEE, 2011.
- [19] X. Yang, H. Steck, Y. Guo, and Y. Liu. On top-k recommendation using social networks. In *Proceedings of the sixth ACM conference on Recommender systems*, pages 67–74. ACM, 2012.
- [20] X. Yang, H. Steck, and Y. Liu. Circle-based recommendation in online social networks. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1267–1275. ACM, 2012.
- [21] N. Zheng and Q. Li. A recommender system based on tag and time information for social tagging systems. *Expert Systems with Applications*, 38(4):4575–4587, 2011.