

國立交通大學

網路工程研究所

碩士論文

社群網路上的潛在用戶探勘

Inferring Potential Users in Social Networks

研究生：徐宗豪

指導教授：彭文志 教授

中華民國 102 年 7 月

社群網路上的潛在用戶探勘

Inferring Potential Users in Social Networks

研究生：徐宗豪 Student : Tsung-Hao Hsu

指導教授：彭文志 Advisor : Wen-Chih Peng

國立交通大學

網路工程研究所

碩士論文

A Thesis

Submitted to Department of Computer and Information Science

Institute of Network 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


社群網路上的潛在用戶探勘

學生：徐宗豪

指導教授：彭文志

國立交通大學網路工程研究所

摘要



隨著網路科技快速的發展，越來越多公司提供社群媒體的服務。對於服務提供者來說，越多的客戶使用他們提供的平台，他們將會有更高的營運收入。而如何找到潛在用戶並吸引他們加入服務，已經成為一項重要的議題。我們把有較高傾向加入某特定服務的使用者稱作“潛在用戶”。關於潛在用戶，我們能得到的資訊，只能藉由他們在某服務內的朋友來獲得間接的資訊。而在現實世界中，人們通常會受到朋友的影響。因此，藉由分析朋友互動的資訊，此篇論文利用一個間接的方式探勘出潛在用戶族群。我們先分析朋友間的互動資訊來抓取出一些 explicit features。進一步地，因為人們經常會有屬於自己的社群(community)，我們利用不同方式建立人與人之間的社群，並藉由這些社群抓取一些 implicit features。為了找到更精確且有用的 feature，我們進行了一系列的觀察，來找出 effective feature set。有了上述的方法與觀察，我們利用分類器(classifier)來輔助我們預測潛在用戶。我們進行了綜合實驗在實際資料集上，結果顯示，我們的方法可以有效地預測潛在用戶，且達到接近 70%的準確率。

Inferring Potential Users in Social Networks

Student : Tsung-Hao Hsu

Advisor : Dr. Wen-Chih Peng

Institute of Network Engineering
National Chiao Tung University

ABSTRACT

With the developing of technologies about networks, there are more and more companies provide social media service. In service providers' view, more customers lead to more income. How to explore new customers has become a significant issue. We call the people with high tendency to join a specific service as potential users. All the information about potential users comes from their friends. In the real world, people were often influenced by their friends. As a result, analyzing friends' interaction behavior logs offer an unique way to explore potential users. In this paper, we extract explicit features based on friends' interaction behavior. Moreover, people tend to organize their own community in their life, we extract community based implicit features for a deeper exploration. To select effective predictors, we do some observation for choosing discriminative feature set. After exploring the effective predictor, we use different classifiers to predict the potential users and compare their effectiveness. Finally, we conduct our method in real dataset and show that the features we extract can reach about 70% accuracy.

誌 謝

首先誠摯的感謝指導教授彭文志博士，兩年來細心的指導並不時的討論、指點我正確的方向，從簡報的技巧、做研究的方法、到一篇完整的論文的產生，這段路途上學到了很多寶貴的方法和經驗，還有不斷鼓勵我們要有開闊的眼界，讓我價值觀提升了不少，另外，由衷的感謝老師讓我在碩二暑假，可以有機會去公司實習，這段實習經驗對我的人生無疑是重大且深具影響的。

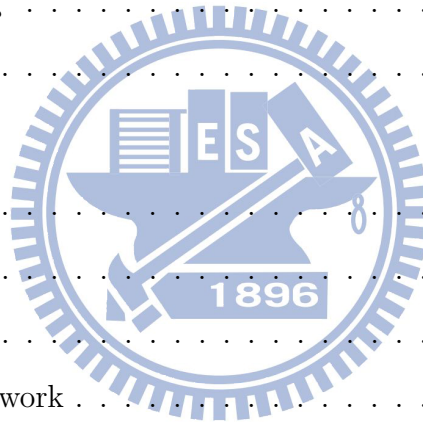
還記得那些在實驗室趕作業、趕專案、甚至是為了剪出一段影片而熬夜到天亮的日子，雖然很累，但卻很值得，兩年來因為有林詠翔、周凡凱、李守峻、陳建志、Tom 對大大小小的事情互相幫忙，才成就了今日的我，革命情感也在不知不覺中建立了起來。兩年的歲月，感謝實驗室的學弟妹為實驗室帶來許多歡樂的氣息，當中有許多歡樂許多笑，無論是一起聚餐、運動、打球、出遊，都凝聚了我們的向心力，使我們在合作時候有更多愉快的回憶。

特別感謝江孟芬學姊在碩士生涯中帶領我、指出我研究過程的缺失，並花了大量的時間幫助我學會正確的研究方法和態度，即使學姊已經畢業了，還是不厭其煩地定時與我討論改進的方法，沒有學姊的幫忙，我的論文無法如此順利的完成。在這短短的本論文的完成另外亦得感謝的大力協助。因為有你的體諒及幫忙，使得本論文能夠更完整而嚴謹。

兩年的時光說長不長，說短不短，碩士班生涯這兩年，我過得很精采，最後，謹以此文獻給我摯愛的雙親。

Contents

1	Introduction	1
2	Related Work	4
2.1	Edge Status Changing	4
2.2	Node Status Changing	5
2.3	Group Exploring	5
3	Overview	7
3.1	Framework overview	7
3.2	Preliminaries	8
3.2.1	Scenario	8
3.2.2	Two-Layer Network	9
3.2.3	Problem Definition	10
3.2.4	Issues	11
3.3	Data	11
4	Method	14
4.1	Feature Normalization	14
4.2	Explicit Features	14
4.2.1	Analyze Method	14
4.2.2	Explicit Local Features	15
4.2.3	Explicit Global Features	17
4.3	Implicit Features	19
4.3.1	Construct community	19

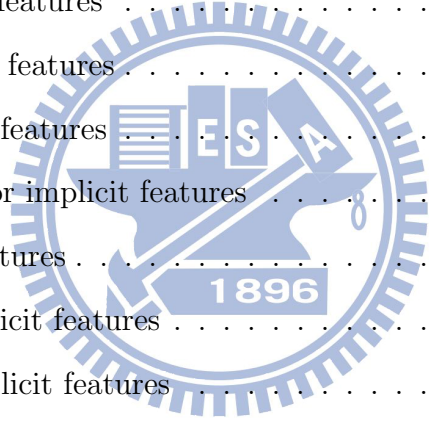


4.3.2	Extract the Implicit Features	20
4.3.3	Analysis	21
4.4	Sparseness	23
5	Experimental Results	25
5.1	Dataset	25
5.2	Data preprocessing	26
5.3	Classifier	26
5.4	Evaluation	27
5.4.1	Experimental Results	27
5.4.2	Predict the original data	27
5.4.3	Sparseness	28
5.4.4	Compare with dimension reduction method	29
5.4.5	Efficiency	31
6	Conclusion	32



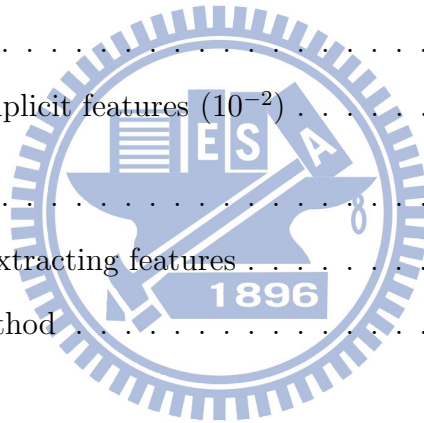
List of Figures

3.1	Framework overview	7
3.2	Visualization of the graph	8
3.3	Nodes joined in different time point	12
4.1	CDF Gap for explicit features	15
4.2	CDF of explicit global features	18
4.3	CDF Gap for implicit features	19
4.4	Induced subnetwork for implicit features	20
4.5	CDF of SI implicit features	21
4.6	CDF of SHRINK implicit features	22
4.7	CDF of DBSCAN implicit features	22
4.8	IG for effective features	23
5.1	Prediction Result for different classifiers	28
5.2	Compare the prediction result in different density and classifiers	29
5.3	Compare with LSA and PCA	30



List of Tables

4.1	Local explicit features	15
4.2	IG of local exp. features (10^{-2})	15
4.3	Global exp. features	16
4.4	IG of global exp. features (10^{-2})	17
4.5	Implicit features	20
4.6	Information gain of implicit features (10^{-2})	21
5.1	Statistics of data	25
5.2	Time complexity for extracting features	25
5.3	Dimension of each method	31



Chapter 1

Introduction

With the emerging of social networks such as Facebook, Twitter, or cell-phone communication applications, reserving or attracting the service customers becomes a significant issue. People joining or leaving lead the networks to be **dynamic**. Since the networks are always dynamic and keep changing every hour and every day, there are more and more research focus on predicting the **dynamic status changing** phenomenon among networks.

In service providers' views, more customers lead to more revenue. Some research [7, 24, 25], which be called as **churn prediction**, concentrate on **reserving** the customers. The goal of churn prediction is to predict whether a existing customer will leave the service or not in the near future. However, a startup company does not have enough customers to reserve. In this case, how to **attract** new customer to join the service is more important than reserving the existing one. In this paper, we try to explore the **potential nodes** which are the inactive nodes (unregistered users) now, while have **higher tendency** to become active nodes (registered users). There are several application for exploring the potential nodes in the networks. For instance, if the service provider can know the potential users, the provider can target on these users and send advertisement information. Moreover, the service provider can further analyze the characteristic of potential users and provide a customized program to attract more users.

There are some similar researches study on dynamic status changing among networks. For instance, [1, 17] study the **edge status** changing which prevalently be called as **link prediction**. The goal of link prediction is to predict whether a edge will appear in the near future or not. Other researches [7, 13, 22, 24, 25] concentrate on **node status** changing

prediction which is more similar to our problem. However, the existing work for prediction the networks dynamic changing all focus on the active nodes, namely that they do not consider the user who **have not joined** the networks. It is much easier to analyze the behavior because active nodes usually have more information to extract. Note that not every networks are composed of both active nodes and inactive nodes as we defined in this paper. For example, the nodes appear in Facebook and Twitter are all registered users, the nodes appear in VoIP or CDR networks are probably unregistered users. In this paper, we mainly study on the later one.

Some issues comes up when we consider both active nodes and inactive nodes. First of all, since inactive nodes have not joined the service, we don't have any **direct** information of them. In the other words, all we can get is some connections with them from active nodes (which would be called *cross-edges* in the following).

Secondly, we do not know any interaction between inactive nodes pair and must infer their characteristics in a **indirectly** way.

To deal with above issues, a *two-layer networks* will be constructed to model the problem, the *cross-edges* are the connection between the active and inactive nodes. For extracting the predictor for classifier, we first observe the interaction between users which called explicit features in the following. Furthermore, since people tend to organize communities in the real world, we extract implicit features based on grouping algorithms including density based [9, 29], sharing interesting based [24] and modularity based [9, 14] methods. After extracting the explicit and implicit features, we do deeper observation for each feature. Recording to the observation, we select a effective features subset as our **powerful predictor** and use classifier to distinguish the potential nodes from inactive nodes set.

In summary, the main contribution of our work is as follows:

- We extract explicit features and community based implicit features to identify potential users.
- We compare the effectiveness of implicit features between different based community algorithm.

- We do several observation for deriving effective features from explicit and implicit features.
- We discuss the sparseness phenomenon and show our method can also apply on sparse dataset.
- We show that our method is more stable than traditional dimension reduction methods.
- We use effective features in classifier to distinguish potential nodes and non-potential nodes from inactive nodes set.

The paper is organized as follows. The next section list systematically the related works. Section 3.2 discuss about the preliminary of this paper and the way to construct two-layer network. An approximate method will be proposed in Section 3.3 to show how to get the ground truth of "inactive nodes with higher tendency to become active" (i.e., potential nodes). For extracting features as our predictor, Section 4.2 analyze the interaction behavior and categorize the features into *local* or *global*. Moreover, because people tend to organize communities in the real world, Section 4.3 further explore the implicit features based on community. Section 4.4 study on the sparseness issue by observing effectiveness of features in different density data. According to the observation in Section 4.2 and 4.3, we use the powerful predictors selected in these two section as the input features for different classifiers and show the experimental results in Section 5. Finally, we summarize this work and future directions in Section 6.

Chapter 2

Related Work

Given a social network S with its vertex set V and edges set E , V usually indicate the users and E indicate the relationship between users in the real world. With the emerging of social networks in the recently years, there are more and more researches study on the social networks. One of them is **status changing** phenomenon [7, 12, 26, 28] in networks. Since social networks are not always in a static state and often **grow** and **change** quickly over time (e.g., some relationship may construct, some users may join or leave), the status changing phenomenon can be refer as the **dynamic** among E or V in the social networks S . Moreover, since people often be together and influenced by friends in the real world, how to conduct a well **group exploring** in the networks is always a core problem.

2.1 Edge Status Changing

Some of the researched focus on **edge** status changing phenomenon, generally be called as *link prediction* problem. In particular, given a time point t , link prediction aim to predict the whether there is a link between two nodes in the near future t' (where $t < t'$). Getoor et al [12] did a completely survey for link prediction problem and summarize some mainly methods to deal with the problem. A mainly method is constructing a probability model [17, 19, 27] to predict the appearance probability of links in the near future. Hasan et al [1] and Lichtenwalter[20] et al refered the link prediction problem as a classification problem and predicted whether a link will appear or not in the near future. The other framework provided

by Sun [26] concentrate on a different problem, they study appearing time for the link in social networks.

2.2 Node Status Changing

Node status changing: The other researches focus on **node** status changing in the networks. For instance, customer *churn prediction* problem [7, 13, 22, 24, 25] is an increasing core issue in customer relationship management (CRM). The goal of churn prediction is to identify the **churner** (i.e., the customers or subscribers have higher probability to leave a specific service) in the networks. The popular models to predict the customer churn are regression models, neural networks and support vector machine [7, 13, 22]. Most of the existing works focus on the individual in the networks and try to extract the powerful predictor for modeling the churn likelihood. [24] provided a group-based churn prediction approach with the assumption that people usually behave together, and apply an influence model to find the leader in a group. Once the leader leaving, the probability of churners in the same group will increase. A similar research proposed by Shaomei Wu et al [28] study the natural arrival and departure of users in a social network. By studying the dynamic arrival and departure correlation among friends using a snapshot in social network, [28] shows that people's status in social networks (e.g., stay or leave) often influenced by their friends. Based on the word-of-mouth behavior among human being, another researches build a information diffusion model [8, 18, 23] called *viral marketing* to analyze the information propagation in networks. The goal of viral marketing is to identify whether a node p in the networks will be **infect** by others, namely that whether p will get a specific information (e.g., news or advertisement) from friends or not.

2.3 Group Exploring

On the other hand, some of researches [19, 24] using **group structure** to help them exploring deeper information in social networks, so as our method in this paper. There are plenty of clustering algorithm for finding the group (community or cluster) in networks. Density-based

[2, 9, 29] clustering algorithm is a well known method. The central idea of density-based algorithm is to merge high density nodes together. However, it requires some parameters to define clusters. A quality measure called *modularity* is a widely use criteria in network grouping. Newman et al [6, 21], Blondel et al [3] and Feng et al [10] using modularity-based algorithm to explore the group and show the effectiveness of modularity. Hierarchical-based clustering is also a prevalent method for networks grouping. However, it is hard to define the stop criteria in hierarchical-based algorithm. [14, 21] merge the central ideas of hierarchical-based and modularity-based clustering algorithm to define the dynamic stop criteria for clustering. Sharing interesting based is a simple and well-performance provided by [24]. In this paper, we will apply some community algorithm [9, 14, 24] to explore more implicit features in the networks.

In this paper, we aim to infer the potential users in the networks, so it is much similar to node status changing prediction[7, 13, 22, 24, 25, 28]. However, all of the researches mentioned above focus on the users who **have already joined** a specific network. In the other word, they consider the nodes that have joined the networks and analyze the activity of status changing. The difference between existing researches and our research is that we consider both joined-users and unjoined-users. Some issues come up because of the missing information among inactive nodes(i.e., unjoined-users), we can only use limited information comes from active set to infer the node changing phenomenon and explore the potential nodes in the networks (i.e., from inactive nodes to active nodes).

Chapter 3

Overview

3.1 Framework overview

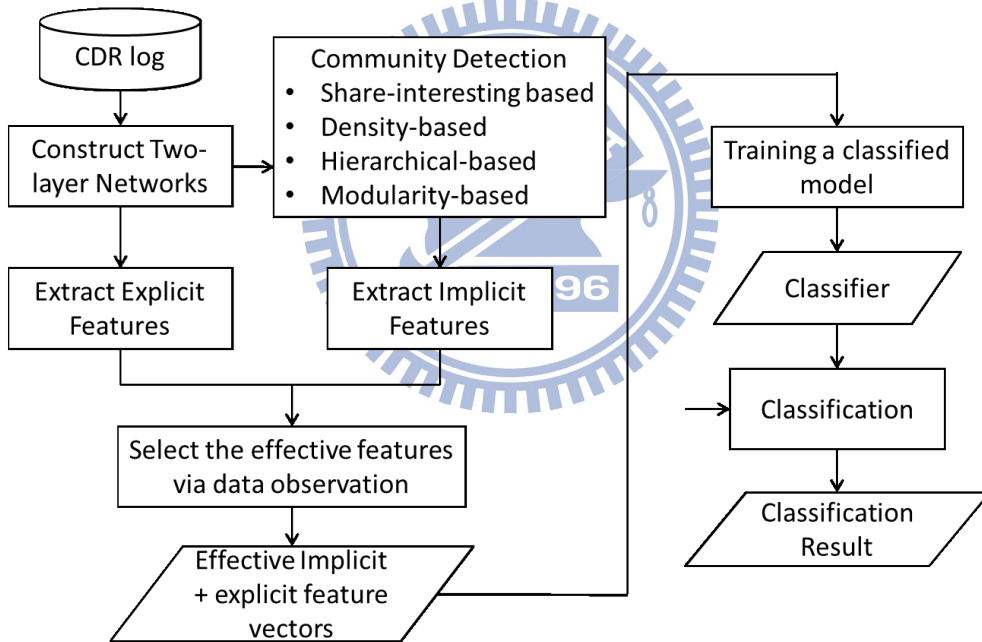


Figure 3.1: Framework overview

In this section, we talk about the framework overview of this paper. The input in the framework is the CDR logs and the join time for each user. We then construct a two-layer networks as discussed in Section 3.2.1, the first layer of the network is composed of active nodes (i.e., joined users) and the second layer consists of inactive nodes (i.e., unjoined users) otherwise. Then we extract some explicit features from the induced subnetwork for each inactive nodes. By using different clustering algorithm to explore the communities on the

first layer network, we can get implicit features (i.e., community features) in the networks. After extracting the explicit and implicit features, selecting an effective features becomes a significant thing. We will do some observation and statistics in order to find a effective feature set as our predictor. Finally, a classifier would be applied to build a model for prediction. In the next section, we will show the detail of our framework.

3.2 Preliminaries

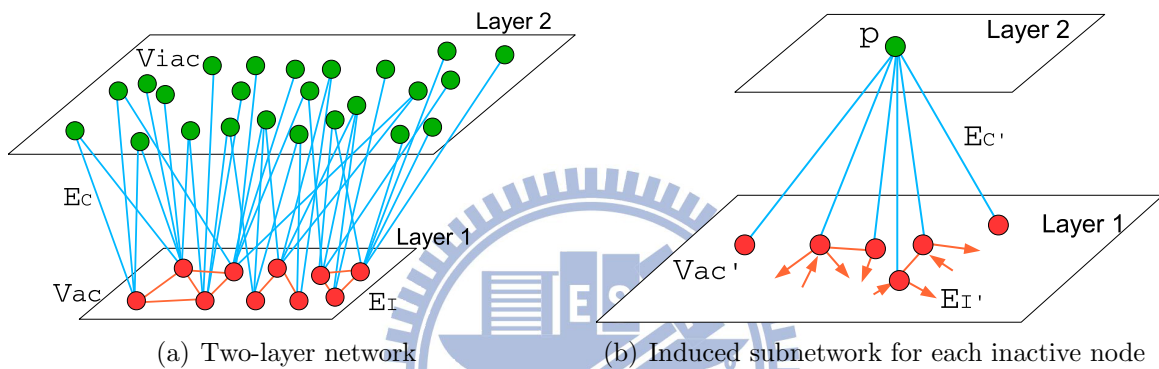


Figure 3.2: Visualization of the graph

In this section, we will discuss the preliminaries about the scenario, problem definition and the issues of our research. There are plenty of types of social network, most of the researchers about social networks focused on analyzing the behavior of *active nodes*. The definition of an active node is the node which have joined the specific social network. In our network, there is another type of node that called *inactive node*. Similar to the previous definition, an inactive node is the node which haven't joined the network while have some contacts with active nodes. Base on these two types of node in network, we can construct a special network which called *Two-layer network*. We will discuss it in detail in the following subsections.

3.2.1 Scenario

A significant thing we have to know is that **not** all of the existing social networks have both active nodes and inactive nodes as defined in this paper. In some of the networks, such as VoIP CDRs (Call Detail Records), phone-call CDRs and so on, both the active nodes and inactive

nodes exist in the networks. For example, considered Figure 3.2(a) as VoIP phone-call (e.g., Skype). The red nodes indicated the users who have joined the service, otherwise, the green nodes indicated the users who have not joined the service while some registered users **connect** with them (i.e., phonebook relationship or contact with the users who are in other service). We consider the former one as active nodes, and consider inactive otherwise. As we know, most of the VoIP service or telecommunications company provide the feature that registered users (active nodes) can communicate with the non-registered users(non-active nodes). Based on this situation, we construct a two-layer network to model our problem.

3.2.2 Two-Layer Network

Figure 3.2(a) shows the *two-layer network*, the *first layer* contains all of the active nodes set V_{ac} and the (*active, active*) edges between them, and the *second layer* is the inactive nodes set V_{iac} with (*active, inactive*) edges. We called the former type of edges as *interaction edges* (E_I). The later one is called as *cross-edges* (E_C). There are two ways to construct cross-edges, the first way is construct them based on interaction frequency. Namely, there is a cross edges between a pair of (active, inactive) if and only if the active node tried to contact with inactive node more than ε times. The second way is simply construct them based on phonebook relationship. The most difference between interaction edges and cross-edges is that the former one is two-way direction edges, while the later one is single direction (the information comes from active nodes).

An important characteristic in the second layer is that we have no any interaction edges between of the inactive nodes. Namely that the (*inactive, inactive*) edges does not exist in our network. The reason of this situation is simple. Recall the previous example again, the inactive nodes is the users who **haven't** joined the service, the behavior or communication between inactive nodes is admittedly no way to know. All the information about inactive nodes we can get is "which set of active nodes did have relationship or communicate with them". To analyze the inactive node set, we build an *induced subnetwork* for each inactive node. As shown in Figure 3.2(b), the induced subnetwork is composed of a specific inactive

node p , its cross-edges $E_{C'}$, the induced active nodes set $V_{ac'}$ linked by E_p and the interaction edges $E_{I'}$ among $V_{ac'}$. Whenever p changes into active, it will fall from second layer to first layer and can contact with the nodes in first and second layer.

The other type of network, such as Facebook, Twitter and so on, the network behavior can only appear among active node pairs. Which means, the communication(e.g., post in Facebook or follow in Twitter) can **not** appear between registered users and non-registered users in this kind of networks. It make sense because in such kind of networks, a registered user doesn't have any direct way to contact with the users outside of the networks. Since our goal is to infer the *potential nodes* in the inactive nodes set, this kind of networks is not the instance we concerned about. The definition of potential nodes will describe in particular in the next subsection.

3.2.3 Problem Definition

The goal of our research is to infer the *potential users* in the networks. The definition of potential users is as follow:

” Given a set of inactive nodes, the potential users is the nodes that have higher tendency to become active.”

As the definition shown above, we can know that the **target nodes** we concerned about is inactive nodes. The information we can get from inactive nodes is much less than active nodes. Since we don't know any information about the behavior of inactive nodes, it is **hard** to analyze or extract feature for inactive nodes **directly**. We will focus on the *induced subnetwork* for each inactive node to deal with this problem. Namely, we use an **indirect** way to explore the potential nodes via extracting some features and choosing the effective feature set from *induced subnetwork*. After extracting the effective feature set, we apply this problem into **classification** problem (i.e., whether a *potential nodes* or not) and using classifier to predict the potential nodes in inactive node set. The detail of approach will discuss in Section 4.2 and 4.3.

Some concomitant issues comes up based on the scenario mentioned above. In the following

subsection we will describe the issues of our research.

3.2.4 Issues

The first issue is the limit among leak information. As mentioned in Section 3.2.1, all the information about inactive nodes we have is the cross-edge relationship with active nodes. This phenomenon caused high difficulty to analyze inactive nodes **directly**. Therefore, we use an **indirect** way to explore potential nodes, namely that the following section will focus on analyzing the *induced subnetwork* for each inactive node.

Derive from the first issue, another issue is hard to know the characteristic about potential users. Since it is difficult to identify the potential users directly, we focus on analyzing their friends' (i.e., induced subnetwork) characteristic. By choosing an indirect way to identify potential users, the predictor for predicting the potential users is hard to extract. Therefore, we explore explicit features and community based implicit features. To extract the useful predictor set, we do some observation which will discuss in Section 4.2 and 4.3.

This section discuss the definition of our problem and issues to solve. The following sections will study on some observation and explore the characteristics of potential nodes in detail.

3.3 Data

In our research, we infer the potential nodes that have **higher tendency** to join the existing network using a real world dataset. The data we used is telecommunication data in Sep. Oct., 2010 which including **caller** c_r , **callee** c_e , **calling time** t_c for each calling behavior and the **join time** t_j for each active nodes. Note that all of the c_r must be situated in the first layer of *two-layer network* we defined in the previous section. While most of the c_e s are situated in the second layer, that is to say, most of the callees have not joined the network. The phenomenon makes sense since the active nodes in our network can almost contact with **everyone** in the real world as we mentioned in Section 3.2. A pair of nodes will be located in the first layer if and only if both of them have join the service (i.e., active nodes).

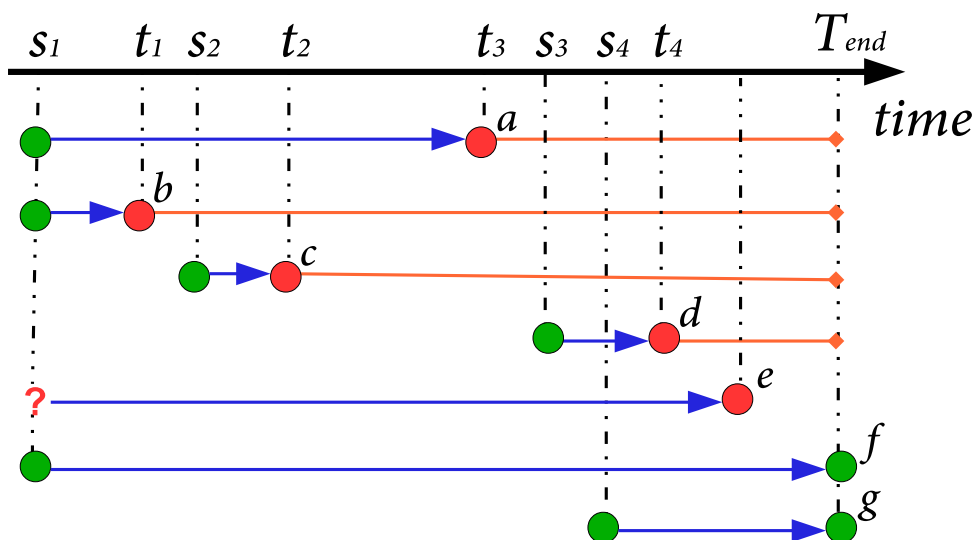


Figure 3.3: Nodes joined in different time point

Recall again that we aim to explore the potential nodes from the inactive set. So as to observe and classify the difference between potential nodes and non-potential nodes, we should have the ground truth to distinguish both of them from inactive set. However, a **higher tendency** to become active is hard to extract in the real world. In this section, we will discuss about the method to get the **approximate** ground truth.

Consider a time line in Figure 3.3, the green nodes indicated the inactive nodes, and active nodes otherwise. The time point s_i represent the **first time** that a specific node appeared in the network and t_i represent the **join time**. The end time of the data is expressed as T_{end} . The question mark of node e means that the node suddenly become active without any portent, that is to say, there is no any *cross-edges* before node e change into active. It is impossible to predict this kind of nodes so that they are not the instance we concerned about. Since **higher tendency** is hard to evaluate in the real world, we can simply consider the active nodes which come from inactive nodes as potential nodes. For instance, the nodes a , b , c and d in Figure 3.3 can be simply considered as potential nodes, and extracting the features from these nodes **before** it changed into active, namely, (t_1, t_2, t_3, t_4) for (a, b, c, d) . While considering node f and g as non-potential nodes since they stay in **inactive status** in whole training data. In our data, there are totally **1,300** potential nodes and **1,676,211** non-potential nodes. As

we can see, the data is very imbalanced. Moreover, some part of inactive nodes changed **early** in training data (node b and c), this situation would cause limited information for their own *induced subnetwork* and difficult to analyze. Furthermore, we consider the **community features** which will study in Section 4.3. The time complexity will increase with time since we have to dynamically construct community and extract features for each node in different join time. As a result, we use an approximate way to observe the potential nodes.

Before discussing about how to get approximate potential nodes, we talk about the assumption in this paragraph. Since all the information we focus on is the induced subnetwork for each inactive node, we can have an assumption as following:

” *The behavior between friends don’t change too much in a **short time period** after a node became active.*”

Thinking about a new active node p_a changed recently, the assumption is reasonable because the apparent change often occur among (p_a, p_a ’s friends) rather than (p_a ’s friends, p_a ’s friends) in the real world. Based on the assumption, we can regard nodes a, b, c and d as inactive nodes although they have changed, and observe their induced subnetwork to extract the features of potential nodes. To be more exhaustive, in order to explore more information of a new active node p_a ’s own induced subnetwork when p_a was inactive, we still consider $p(a)$ as inactive after $p(a)$ had changed for a short time period. This **approximate** method make us much easy to observe the potential nodes. To simplify the assumption, we consider all the potential nodes (namely, nodes a, b, c and d) as inactive until the end of the time line. In the other words, all of the potential nodes would be treat as inactive nodes and build their own induced subnetwork before time point T_{end} in Figure 3.3.

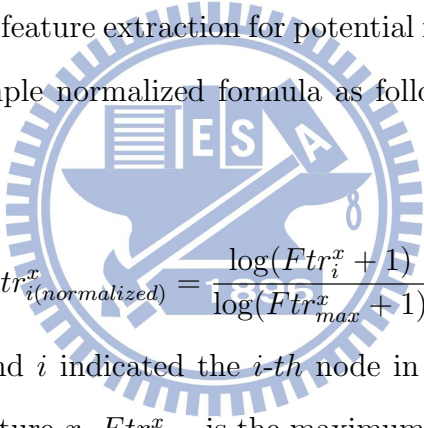
After getting the approximate ground truth, the data could be labeled as *positive* or *negative* (i.e., potential nodes or non-potential nodes). The goal of this paper is to explore the potential nodes from inactive set, that is to say, the objective is distinguishing the positive and negative nodes from dataset. We will study and observe among these two kind of nodes in the following section.

Chapter 4

Method

4.1 Feature Normalization

In next section, we study on the feature extraction for potential nodes. For comparing different features together, we use a simple normalized formula as following to compress the feature value into $[0,1]$.


$$Ftr_{i(normalized)}^x = \frac{\log(Ftr_i^x + 1)}{\log(Ftr_{max}^x + 1)}$$

Where x is feature name and i indicated the i -th node in the network. Ftr_i^x is original feature value of i -th node in feature x , Ftr_{max}^x is the maximum feature value in feature x and $Ftr_{i(normalized)}^x$ is the feature value after normalized. For preventing undefined logarithm (i.e., $\log 0$), we add an integer 1 into each element. In the following subsection, we will normalize all the feature value before comparing them together.

4.2 Explicit Features

4.2.1 Analyze Method

The feature value we extract is **continuous** while the label of the ground truth is **discrete**(i.e., positive or negative potential user). To select the powerful predictor, we compute *Information Gain* (**IG**) [30] for each feature.

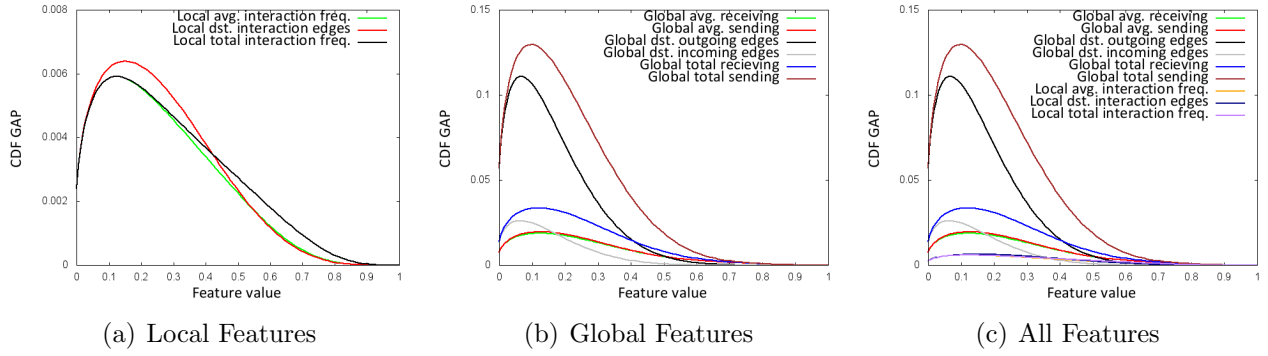


Figure 4.1: CDF Gap for explicit features

4.2.2 Explicit Local Features

Feature Name	Description
total interaction freq.	Total internal interaction frequency between a particular induced subnetwork.
avg. interaction freq.	Average internal interaction frequency between a particular induced subnetwork.
dst. interaction edges	Total internal distinct pair of interaction between a particular induced subnetwork.

Table 4.1: Local explicit features

To identify the powerful predictor for potential nodes, we start with exploring the *explicit features*. The explicit features were divided into two part. The first part is explicit local features and global features otherwise. In this subsection, we concentrate on the explicit local features.

As mentioned in Section 3.2.2, there is a induced subnetwork for each inactive nodes. The *local features* focus on the interaction **between** the induced subnetwork. In particular, considering a specific inactive node p_{ia} and referring to Figure 3.2(b). There are some directional arrows in the first layer, the arrows, which are the ignore information in local features, means

Feature Name	Information Gain
total interaction freq.	0.22
avg. interaction freq.	0.22
dst. interaction edges.	0.08

Table 4.2: IG of local exp. features (10^{-2})

Feature Name	Description
avg. receiving freq.	Average receiving frequency among a particular induced subnetwork.
avg. sending freq.	Average sending frequency among a particular induced subnetwork.
total receiving freq.	Total receiving frequency among a particular induced subnetwork.
total sending freq.	Total sending frequency among a particular induced subnetwork.
dst. incoming edges	Total incoming distinct edges of interaction among a particular induced subnetwork.
dst. outgoing edges	Total outgoing distinct edges of interaction among a particular induced subnetwork.

Table 4.3: Global exp. features

that the incoming/outgoing interaction from/to the other nodes which have no *cross edges* with p_{ia} . Table 4.1 lists the features be extracted as explicit local features and table 4.2 shows the **C** and **IG** for each local feature. As we can see, the correlation and information gain of local features are all in very low value.

There is an interesting situation that *dst.interactionedges* is smaller in **IG**. To explain this situation, we plot **CDF Gap** for deeper observation. In Fig 4.1(a), we can see that the gap of *dst.interactionedges* is higher than others when feature value < 0.4 while the gap become lower when feature value > 0.4 . The distribution cause the situation that mentioned before. However, the distribution gap among all of the local features is not discriminative enough (as shown in the y -axis scale) and it is obvious that the CDF between potential and non-potential nodes in local features is nearly **overlapped**. The above observation implied that if we only consider the interaction **between** (i.e., local features) an induced subnetwork, it is not powerful enough to distinguish the potential nodes and non-potential nodes from inactive node set. In the next subsection, we extract the global features and discuss the effectiveness of each feature.

Feature Name	Information Gain
avg. receiving freq.	0.665
avg. sending freq.	0.698
total receiving freq.	1.007
total sending freq.	6.248
dst. incoming edges	0.662
dst. outgoing edges	6.174

Table 4.4: IG of global exp. features (10^{-2})

4.2.3 Explicit Global Features

Because it is not discriminative enough in local features, we extract the global features for deeper observation. The difference between local features and global features is that the former one only consider the interaction between subnetwork while the later one consider **all** the interaction among the subnetwork. In the other word, refer to Figure 3.2(b) again and consider a inactive node p_{ia} , the global features consider all of the interaction among the first layer, including the interaction with the nodes that have no cross edges with p_{ia} (i.e., the directional arrows).

Table 4.3 lists all the features we extract for global features. Note that we consider the **direction** (i.e., sending and receiving) in global features but do not in local features. The reason is that we only consider the interaction between subnetwork in the local features, the total sending and receiving count would be the same. Table 4.4 shows the PC and IG for global features. As we can see, *total sending freq.* and *dst. outgoing edges* have discriminately higher PC and IG than other global features, namely that these two features can be considered as more **powerful predictor** than others.

For getting how **powerful** of *total sending freq.* and *dst. outgoing edges*, we plot Figure 4.2 to show the CDF distribution. Clearly, 90% of the non-potential nodes' feature value is smaller than 0.15 in both total sending freq. and dst. outgoing edges. Based on Figure 4.2, we can briefly conclude that the potential nodes will have higher feature value than non-potential nodes in both total sending freq. and dst. outgoing edges features. The result implied that if a inactive node p_{ia} 's induced subnetwork have **more calls** and **more distinct objects to**

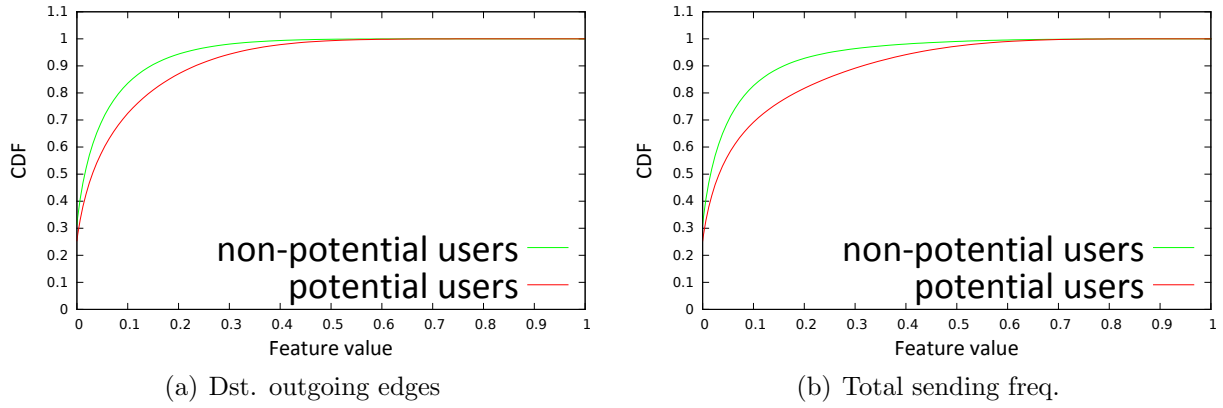


Figure 4.2: CDF of explicit global features

contact, p_{ia} will have higher tendency to become an active node. In the real world, we can consider this situation as "If people's friends using a service frequently and widely, they will have higher tendency to join the service".

The global features CDF gap as in Figure 4.1(b) shows that the gap of total sending freq. and dst. outgoing edges are both higher than the others, which verify our choice of **powerful predictor** again. In Figure 4.1(c), we compare the explicit local and global features together. It is obvious that all the local features located in a lower gap value than global features, which indicate that all of the global features are more powerful than local.

By the observation above, our conclusion is that people tend to join a specific service if their friends use the service **frequently** and **widely**. In all the explicit features, we choice the global features *total sending freq.* and *dst. outgoing edges* as our powerful predictor for predicting the potential nodes. However, people often have their own group and be influenced by the group members in the real world. In this section, we do not consider any **group** concept for feature extraction. Base on this situation, we will extract some group features based on different clustering algorithm in the next section.

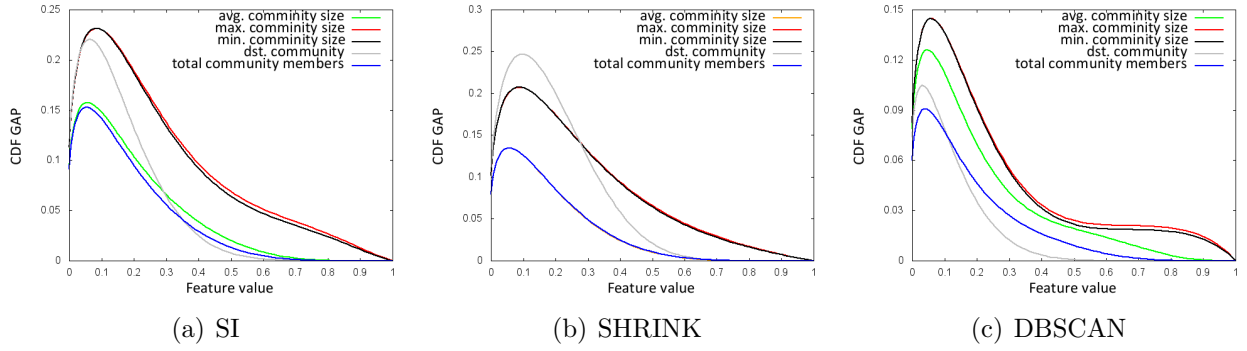


Figure 4.3: CDF Gap for implicit features

4.3 Implicit Features

4.3.1 Construct community

With the emerging of social networks, people tend to interact with it and usually have their own group in networks. There are plenty of clustering (grouping) algorithms to explore the groups in the networks. In this section, we apply three different based methods [9, 14, 24] to explore the communities and compare the features for different methods. The central idea of each algorithm is quite different. [9] provided density-based clustering algorithm which grouped the nodes with high density. There are two parameters which are minimum reachable point $MinPts$ and maximum radius Eps . We set the former one as 2 and the 0.5 otherwise. [24] using a simple method which keeping the $top - k\%$ of the heaviest edges and consider the connected components as groups, namely that a pair of nodes will be grouped together if and only if their edge weight greater than $top - k\%$. [14] mixed the central ideas of hierarchical and modularity, the method in [14] is a parameter-free algorithm and it will find optimal groups based on modularity measurement.

In this paper, we construct the communities on the first-layer (i.e. among active nodes) of two-layer network. We can not construct community on the second-layer since there is no any edge among there. We can extract implicit features based on the community after constructing the communities in the networks. In the next subsection, we discuss about the implicit features we extracted and compare the effectiveness of each features with different community methods.

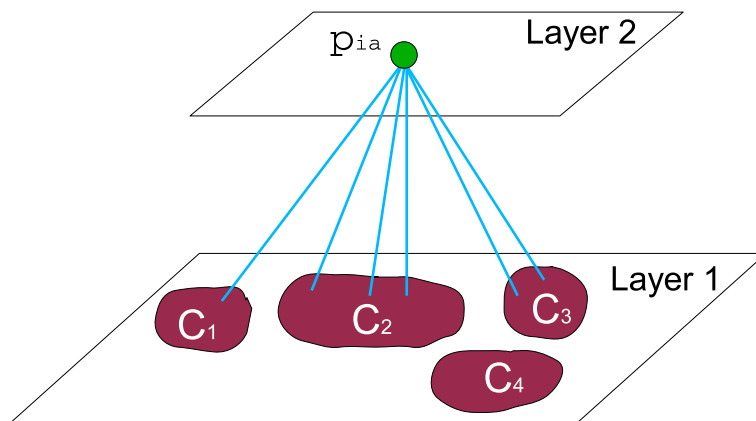


Figure 4.4: Induced subnetwork for implicit features

Feature Name	Description
total com. members	Total community members of the nodes belong to in a particular induced subnetwork
dst. com.	Number of distinct communities of the nodes belong to in a particular induced subnetwork
max. com. size	The maximum community size of the nodes in a particular induced subnetwork
min. com. size	The minimum community size of the nodes in a particular induced subnetwork
avg. com. size	The average community size of the nodes in a particular induced subnetwork

Table 4.5: Implicit features

4.3.2 Extract the Implicit Features

After grouping the communities, all of the inactive nodes in the second layer must link with **part of** community members in the first layer. In particular, consider Figure 4.4, the inactive node p_{ia} has some cross-edges with community C_1 , C_2 and C_3 's members. We extract the implicit features based on these community members' characteristics. Note that we do not consider the community has no any cross-edges with p_{ia} (e.g., C_4). Table 4.5 lists the implicit features we extract for community. In the next subsection, we will discuss the effectiveness of each feature and tell the divergence between potential nodes and non-potential nodes.

Feature Name	SI	SHRINK	DBSCAN
total com. members	3.433	2.779	0.865
dst. com.	6.544	5.69	0.268
max. com. size	7.57	6.27	7.226
min. com. size	7.735	6.212	6.96
avg. com. size	3.513	2.612	2.9

Table 4.6: Information gain of implicit features (10^{-2})

4.3.3 Analysis

Table 4.6 shows the information gain for each feature using different ways of clustering algorithm. As we can see, *max.com.size* and *min.com.size* always have higher information gain in different based algorithm except that *dst.com.* has the highest in SHRINK algorithm.

Figure 4.3 shows that the maximum and minimum community size always have highest CDF gap in both SI and DBSCAN clustering algorithm. Although the *dis. community* has the largest gap in SHRINK when the feature value is smaller than 0.3, the maximum and minimum community size features tend to have discriminative afterward. According to above observation, we can briefly conclude that *max.com.size* and *min.com.size* features is powerful predictors in different based of clustering algorithm. For digging the predictor deeper, we focus on the maximum and minimum community size and plot the CDF for each method.

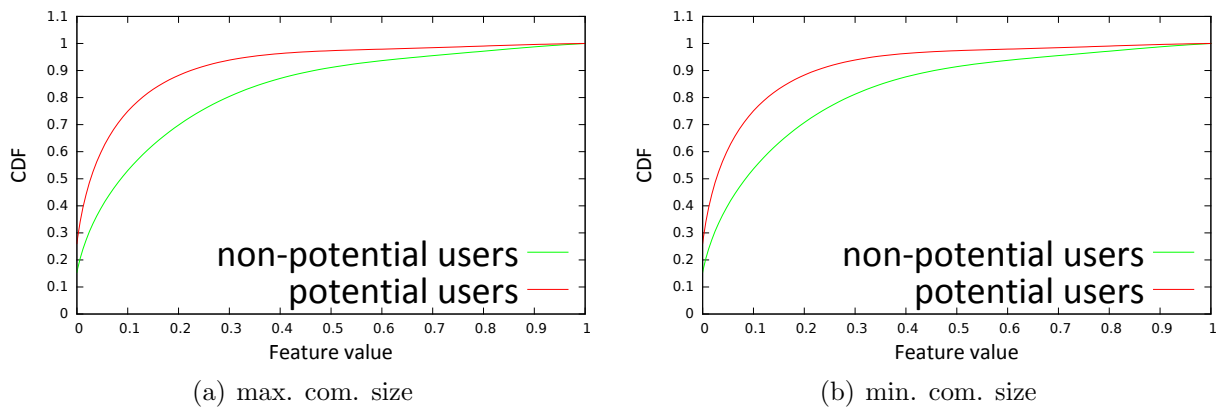


Figure 4.5: CDF of SI implicit features

Figure 4.5 to 4.7 shows the CDF of maximum and minimum community size in different clustering methods. As we can see, most of the potential nodes have lower feature value in

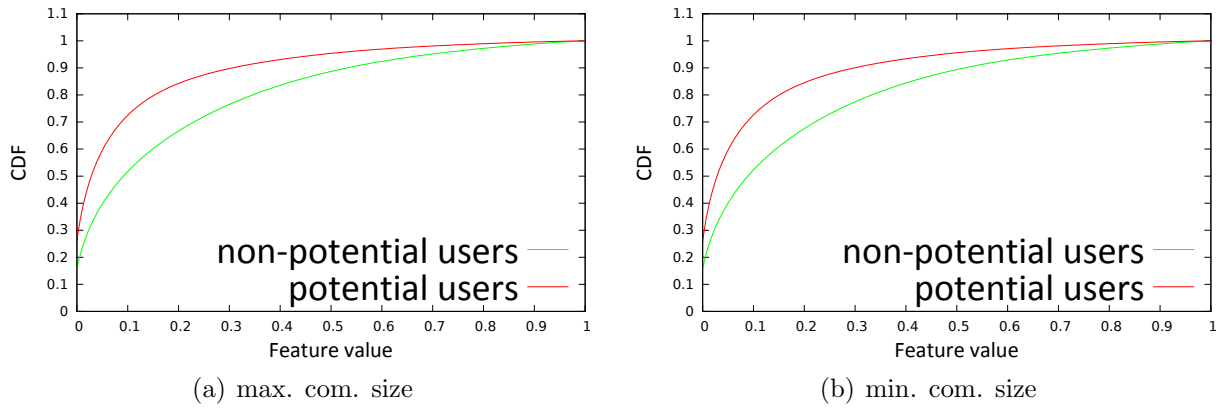


Figure 4.6: CDF of SHRINK implicit features

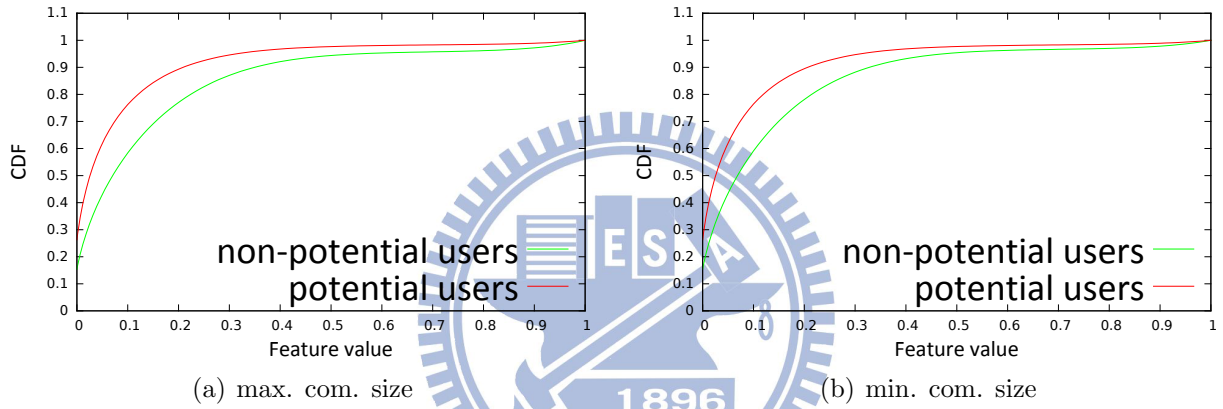


Figure 4.7: CDF of DBSCAN implicit features

both features than non-potential nodes. The situation implied that the nodes in potential nodes' induced subnetworks are in the smaller community than non-potential nodes'. We can consider the result as "the smaller community size of a person p 's friends in, the higher tendency p will join the service". The result make sense because if the person connected with p are all in the large community, they would probably be a **public** community such as advertisement community. In the other words, people tend to join a service if their connected person in the service are in the **private** community such as family community or colleague community rather than an advertisement community.

Our conclusion from above observation is that people tend to be attracted by **small** community size whatever which clustering method be applied. Namely that if a person p 's friends incline to be in small community, p will have higher probability to join the service. Based

on the observation, we choice the maximum and minimum community size as our powerful predictors for implicit features.

So far, we extract the explicit features and implicit features for distinguish the potential nodes from inactive nodes and choice *total sending freq.* and *dst. outgoing edges* as explicit powerful predictor, *max.com.size* and *min.com.size* as implicit powerful predictor. In the section 5, we will use SVM classifier to show the effectiveness of the features we extract.

4.4 Sparseness

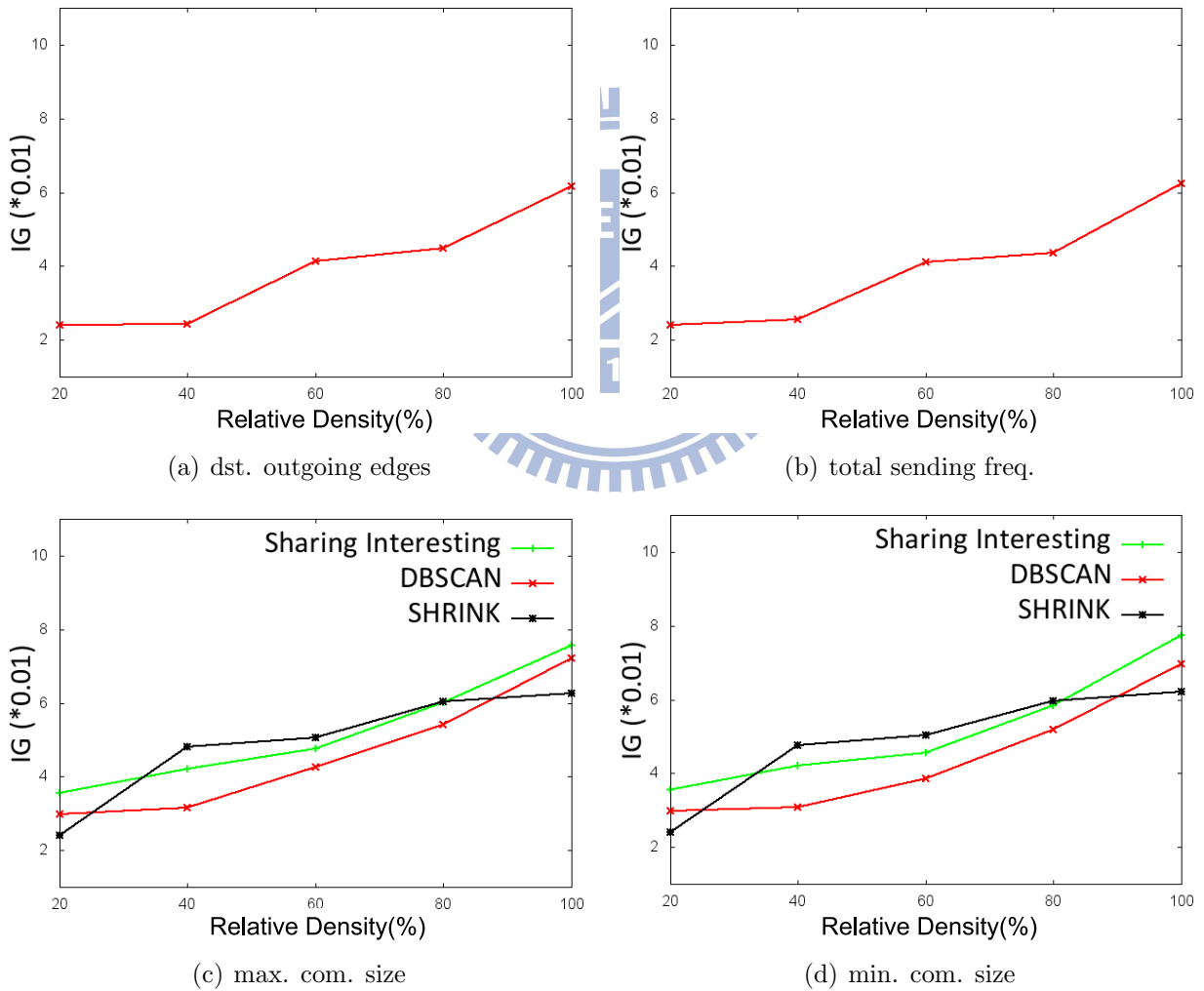


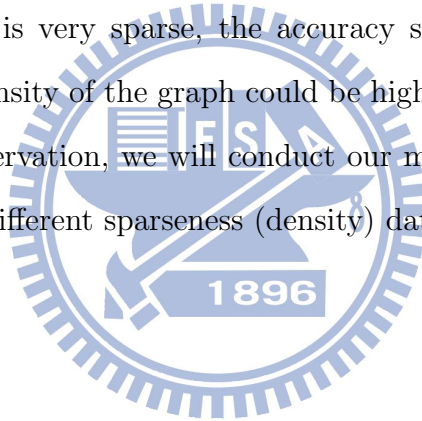
Figure 4.8: IG for effective features

In this section, we study on the sparseness issue. The graph density of first layer in our dataset is about 1.08×10^{-6} and each active node provide 0.0874 interaction edges in average.

Obviously, the data we use is a very **sparse** data. In order to show that the features we extract are also effective on sparse dataset, we generate five simulation data sets by randomly sampling the edges of original data and keeping 20%, 40%, 60% or 80% edges.

Figure 4.8 shows the information gain of the effective features we explore in the previous section. Figure 4.8(a) and Figure 4.8(b) are the information gain of explicit features while Figure 4.8(c) and Figure 4.8(d) are the implicit features among different grouping algorithm. As we can see, the higher the graph density is, the higher the IG will be. The figures implied that a sparse data would cause the predictors become **less powerful**. The reason is simple, because if the graph getting sparse, we will lose more information to distinguish the different type of nodes.

Although the data we use is very sparse, the accuracy still can hit almost 70%. The situation implied that if the density of the graph could be higher, the result could get better. To proof the conclusion of observation, we will conduct our method on the real dataset and compare the accuracy among different sparseness (density) data in the Section 5.



Chapter 5

Experimental Results

Name	Value
Total active nodes	80,764
Total inactive nodes	436,257
Total potential nodes	1,330
Total interaction edges in 1st layer	7,064
Total cross edges	507,338
Avg. cross edges per inactive node	1.16
Avg. interaction edges per active node	0.0874
Avg. cross edges per active node	6.281
Density(sparseness) of first layer	$1.08 * 10^{-6}$

Table 5.1: Statistics of data

feature type	time (s)
Explicit features	61.57
SI implicit features	11342.17
SHRINK implicit features	11998.65
DBSCAN implicit features	11347.59

Table 5.2: Time complexity for extracting features

5.1 Dataset

We conduct the experiment on the real dataset. The data we use is the CDRs provided by Chunghwa Telecom which including **caller**, **callee** and **calling time** for each instance.

Another information about the data is the **join time** for each active node. We can get the approximate ground truth via these information as mentioned in Section 3.3. Before starting the experiment, we practice some data preprocessing which will discuss in the next subsection.

5.2 Data preprocessing

We practice 2 steps for data preprocessing as following.

Filter the original data: There are totally 2,289,870 distinct call pairs in the CDRs. About 77.5% of the pairs are less than 5 calls. In the real world, people sometimes contact a unknown person for a purpose. For example, someone want to reserve a table for dinner and have a call with the restaurant. In this case, they are not friends and probably contact only once, which is not the users we consider about. Furthermore, the phenomenon would bring too many **outliers** for inactive nodes. To refining the nodes in the graph, we filter out the edges with less than **5 calls**. Table 5.1 lists the statistics after refining the data.

Under sampling: The goal of our framework is to distinguish potential nodes from inactive nodes. Referring to Table 5.1, the amount between potential nodes and inactive nodes is very unbalanced. In the other word, the number of potential nodes (positive) and non-potential nodes (negative) is very unbalanced. If we put the data into classifier directly, the precision could be 99.9% via always predicting as negative. To deal with this problem, we under-sampling the amount of inactive nodes as equal to potential nodes. The **worst case** of our classifier would be 50% via always predicting as positive or negative after under-sampling.

After the preprocessing, the data is more balanced and less outliers. In the following subsection, we will show the experimental results of our framework based on these data.

5.3 Classifier

To predict whether a node is potential node or not, we training classification model via AdaBoost, Random Forest and SVM [4, 5, 11]. The central idea of these classifiers is quite different. AdaBoost[11] proposed a **meta-algorithm** to merge weak learning model into

strong learning model. After running T iteration, AdaBoost can guarantee that the precision is better than previous weak classifier. A well known classifier Support Vector Machine (SVM) [5] proposed a function based classifier and tried to find a **hyper-plane** to classify the data. The objective function in SVM is maximizing the **margin** of **support hyper-plane** so as to split the different class as well as possible. The other type of classifier called Random Forest [4] generate a large number of **decision trees** and using the result of these decision tree to **vote** the most popular class. Finally, [15] using Bayesian probability to predict item's class. Based on Bayesian probability, [15] can predict the probability even though the feature combination does not exist in training data.

5.4 Evaluation

We use **10-fold cross validation** [16] to evaluate our proposed framework. Namely that we split the data into 10 equally pieces, and choose 9 pieces for training while the rest 1 piece for testing. By Changing the testing and training set in round-robin way until all of the pieces has been assigned as testing set (i.e., repeat the steps 10 times), we can get an evaluation result in each step. We compute the **average** of the result and consider it as our final accuracy.

5.4.1 Experimental Results

5.4.2 Predict the original data

Figure 5.1 shows the prediction result of the original data. As we can see, the ineffective features performed badly in all classifiers and the effective explicit and implicit features performed well otherwise. Based on the community structure, the implicit features could be better predictors than explicit features. That is to say, it is easier that people influenced by their neighbor community than neighbors' explicit behavior. We consider both the explicit and implicit features for different clustering method and put them as predictors in the classifiers. The result is getting better if we consider both explicit and implicit features.

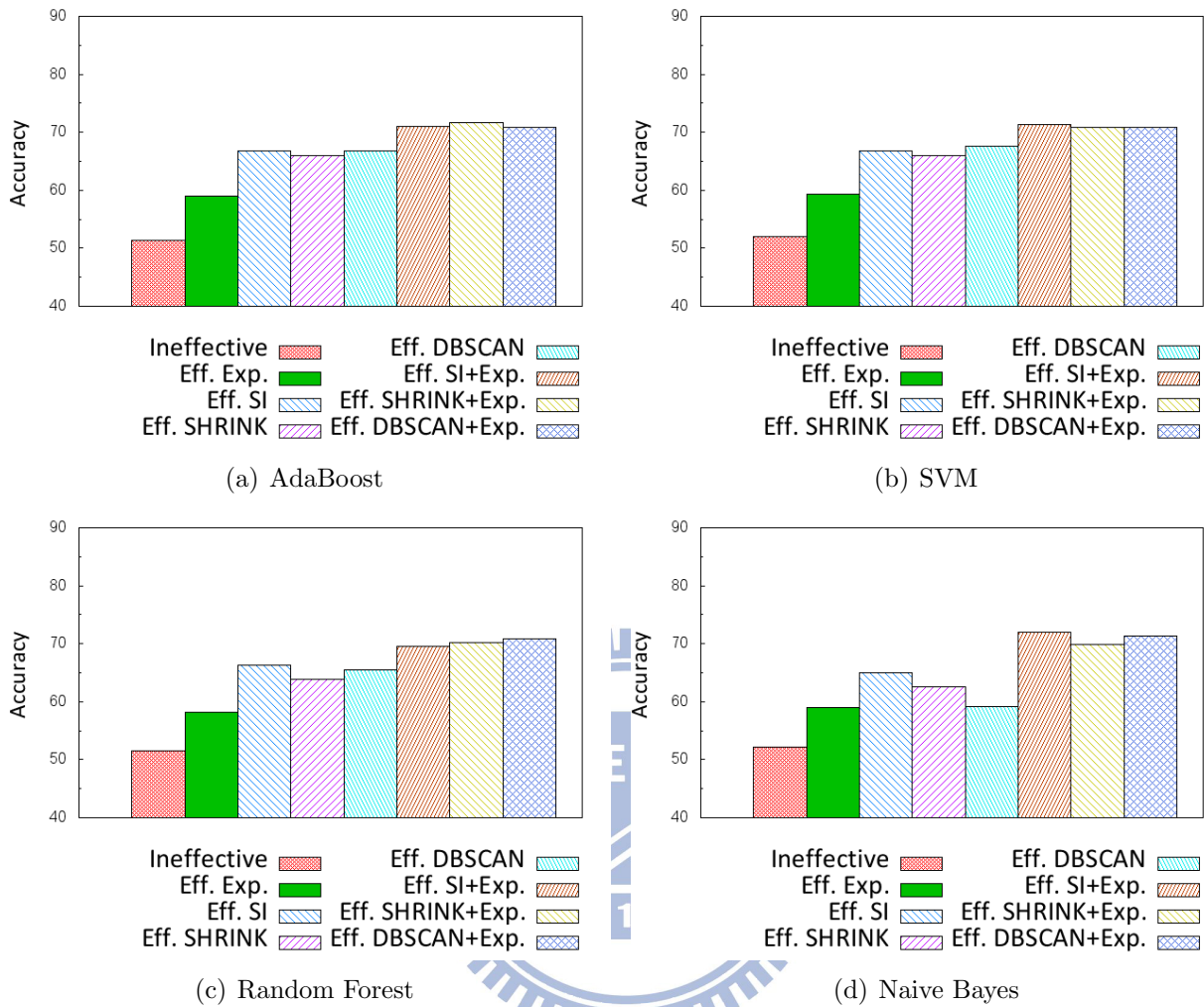


Figure 5.1: Prediction Result for different classifiers

5.4.3 Sparseness

For verifying whether considering explicit and implicit features always performed well or not, we discuss the performance in different sparseness of data. We sampling the original data as mentioned in Section 4.4. Figure 5.2 shows the performance in different sparseness of data. The result indicate that considering both of the explicit and implicit features is not always better. In SVM and Adaboost classifier, purely considering the effective implicit features is better when the sparseness is under 40%. With the increasing of graph density, considering all of the effective features will lead the prediction much better.

Based on the evaluation result, we can simply conclude that the higher density of the graph, the easier to explore the potential users, which verified the observation in Section 4.4.

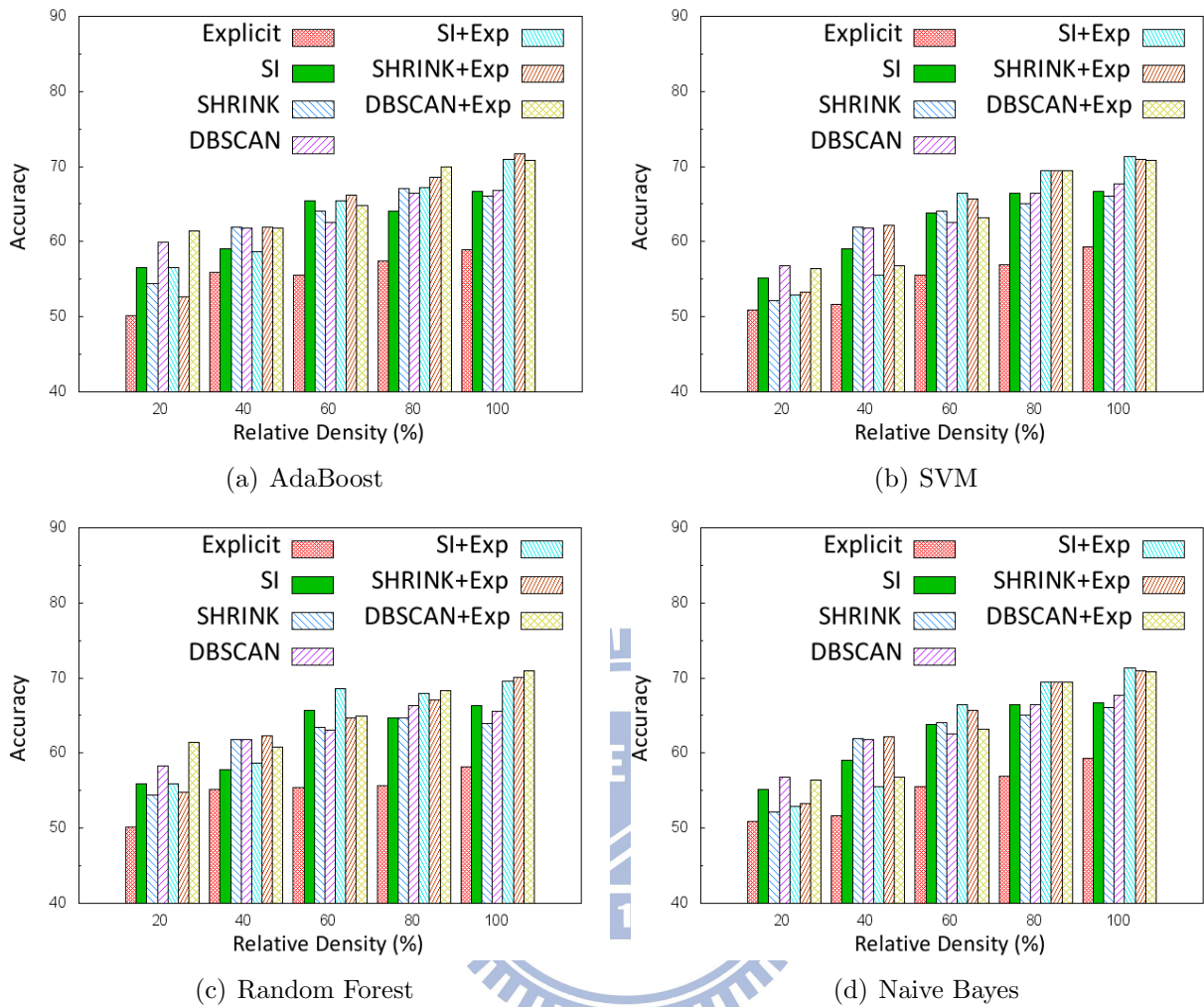


Figure 5.2: Compare the prediction result in different density and classifiers

Namely that if we can have a higher density data, we can get higher performance. For all classifier, the predictors can nearly get about 70% accuracy via considering both explicit and implicit features, whatever which the clustering method be applied.

5.4.4 Compare with dimension reduction method

Since we extract the effective features via observation, we compare our observation results with traditional *dimension reduction* methods. **PCA** and **LSA** are well-known dimension reduction methods based on **singular value decomposition**(SVD). Figure 5.3 compared the prediction result with PCA and LSA and Table 5.3 shows the number of dimension after reducing. Clearly, except Random Forest classifier, LSA often performed worse than the

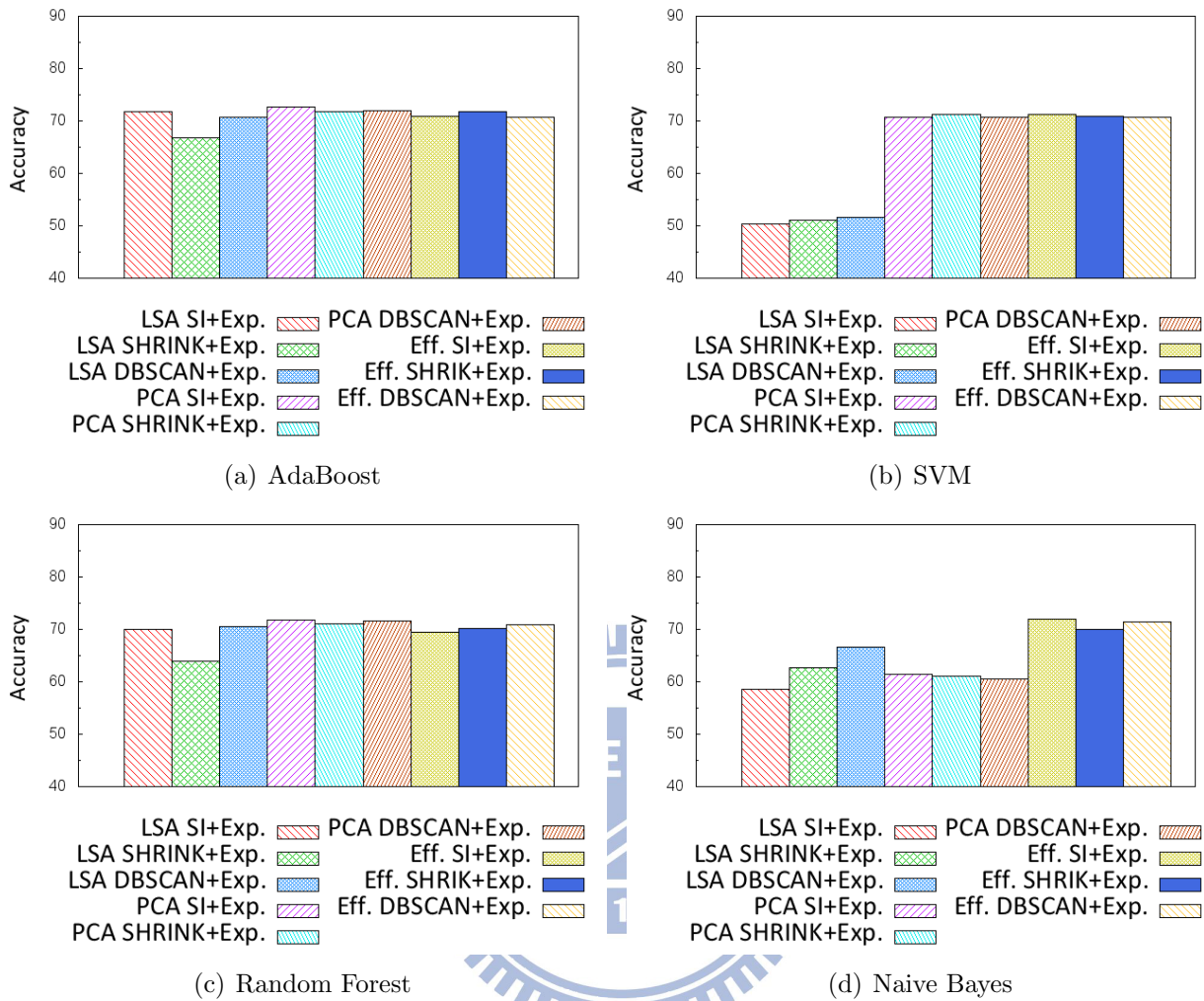


Figure 5.3: Compare with LSA and PCA

effective features we extract. The accuracy in LSA is not ideal though it has lower dimension after reducing. In the other hand, PCA can perform better than LSA and nearly reach the accuracy of the features we extracted. However, in Figure 5.3(d), we can see that the performance of PCA is merely about 60%.

Moreover, the traditional dimension reduction method is **unstable** for classifier, namely that the performance would be influenced by the classifier badly. Another important thing is, if we conduct the dimension reduction method directly in the data, we would hard to know the implying meaning of user behavior. Therefore, the effective features we extracted are more **stable** in each classifier and the implying meaning is much easy to observe.

Feature Name	LSA	PCA	Ours
SI+Exp.	4	5	4
SHRINK+Exp.	4	5	4
DBSCAN+Exp.	3	5	4

Table 5.3: Dimension of each method

5.4.5 Efficiency

Although merge the explicit and implicit features can get better prediction accuracy, computing the both of the features may cost a lot of execution time. Table 5.2 lists the execution time for extracting the features. Obviously, extracting the implicit features take several times than explicit features. The **bottleneck** is that we have to compute the **all-pair** similarity (or distance) in first layer for constructing communities. The time complexity of the step takes $O(N^2)$ and lead extracting the implicit features very inefficient. With the increasing of graph density and size, the execution time for extracting the community based features will become much inefficient. However, as shown in Figure 5.2, the performance of explicit features become better when the data getting dense. As a result, to deal with efficiency problem, we can simply use explicit features as our predictors when the data has high density.

Chapter 6

Conclusion

In this paper, we aim to infer the potential users who have **higher tendency** to join a specific service. To model the problem, we construct two-layer networks via allocating active nodes in first layer and inactive nodes in second layer. Based on the networks, we explore the explicit features first and extract the effective feature subset. Since human being have social behavior and often behave together, we further explore the implicit features based on community algorithm. By doing many observing among the features, we found that the implicit features (community based features) is more powerful than explicit features. The situation imply that people tend to influenced by the group around them. Furthermore, after analyzing the effective features, we conclude that a potential users' friends usually be in **small size** of group and use the service **frequently** and **widely**. After extracting the effective feature set, we use classifier to predict the final answers and show that the features we extract can perform well on different classifier.

We further study on sparseness issues. As shown in Section 4.4, the higher density the data is, the more powerful the feature can be. Although we conduct the experiment on a very sparse dataset, the accuracy still can reach nearly 70%. If we can have higher density data, the prediction accuracy can be much better.

Finally, we compare the effective features with traditional dimension reduction method. The result shows that we can use **fewer** features to predict potential users correctly and the features we extract are more **stable** for all classifier. Most of all, by using the effective features, the imply meaning is easy to understand rather than using dimension reduction

method directly.



Bibliography

- [1] Mohammad Al Hasan, Vineet Chaoji, Saeed Salem, and Mohammed Zaki. Link prediction using supervised learning. In *SDM06: Workshop on Link Analysis, Counter-terrorism and Security*, 2006.
- [2] Mihael Ankerst, Markus M Breunig, Hans-Peter Kriegel, and Jörg Sander. Optics: ordering points to identify the clustering structure. *ACM SIGMOD Record*, 28(2):49–60, 1999.
- [3] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, 2008.
- [4] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.
- [5] Chih-Chung Chang and Chih-Jen Lin. Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27, 2011.
- [6] Aaron Clauset, Mark EJ Newman, and Cristopher Moore. Finding community structure in very large networks. *Physical review E*, 70(6):066111, 2004.
- [7] Kristof Coussement and Dirk Van den Poel. Churn prediction in subscription services: An application of support vector machines while comparing two parameter-selection techniques. *Expert systems with applications*, 34(1):313–327, 2008.
- [8] Pedro Domingos and Matt Richardson. Mining the network value of customers. In *Pro-*

- ceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 57–66. ACM, 2001.
- [9] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. *Kdd*, 1996.
- [10] Zhidan Feng, Xiaowei Xu, Nurcan Yuruk, and Thomas AJ Schweiger. A novel similarity-based modularity function for graph partitioning. In *Data Warehousing and Knowledge Discovery*, pages 385–396. Springer, 2007.
- [11] Yoav Freund and Robert E Schapire. Experiments with a new boosting algorithm. In *MACHINE LEARNING-INTERNATIONAL WORKSHOP THEN CONFERENCE-*, pages 148–156. MORGAN KAUFMANN PUBLISHERS, INC., 1996.
- [12] Lise Getoor and Christopher P Diehl. Link mining: a survey. *ACM SIGKDD Explorations Newsletter*, 7(2):3–12, 2005.
- [13] Rupesh K Gopal and Saroj K Meher. Customer churn time prediction in mobile telecommunication industry using ordinal regression. In *Advances in Knowledge Discovery and Data Mining*, pages 884–889. Springer, 2008.
- [14] 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.
- [15] George H John and Pat Langley. Estimating continuous distributions in bayesian classifiers. In *Proceedings of the Eleventh conference on Uncertainty in artificial intelligence*, pages 338–345. Morgan Kaufmann Publishers Inc., 1995.
- [16] Ron Kohavi et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *International joint Conference on artificial intelligence*, volume 14, pages 1137–1145. Lawrence Erlbaum Associates Ltd, 1995.

- [17] Vincent Leroy, B Barla Cambazoglu, and Francesco Bonchi. Cold start link prediction. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 393–402. ACM, 2010.
- [18] Jure Leskovec, Lada A Adamic, and Bernardo A Huberman. The dynamics of viral marketing. *ACM Transactions on the Web (TWEB)*, 1(1):5, 2007.
- [19] 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.
- [20] Ryan N Lichtenwalter, Jake T Lussier, and Nitesh V Chawla. New perspectives and methods in link prediction. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 243–252. ACM, 2010.
- [21] Mark EJ Newman and Michelle Girvan. Finding and evaluating community structure in networks. *Physical review E*, 69(2):026113, 2004.
- [22] Parag C Pendharkar. Genetic algorithm based neural network approaches for predicting churn in cellular wireless network services. *Expert Systems with Applications*, 36(3):6714–6720, 2009.
- [23] Matthew Richardson and Pedro Domingos. Mining knowledge-sharing sites for viral marketing. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 61–70. ACM, 2002.
- [24] Yossi Richter, Elad Yom-Tov, and Noam Slonim. Predicting customer churn in mobile networks through analysis of social groups. In *Proceedings of the 2010 SIAM International Conference on Data Mining (SDM 2010)*, 2010.
- [25] Guojie Song, Dongqing Yang, Ling Wu, Tengjiao Wang, and Shiwei Tang. A mixed process neural network and its application to churn prediction in mobile communications.

- In *Data Mining Workshops, 2006. ICDM Workshops 2006. Sixth IEEE International Conference on*, pages 798–802. IEEE, 2006.
- [26] Yizhou Sun, Jiawei Han, Charu C Aggarwal, and Nitesh 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.
- [27] Chao Wang, Venu Satuluri, and Srinivasan Parthasarathy. Local probabilistic models for link prediction. In *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*, pages 322–331. IEEE, 2007.
- [28] Shaomei Wu, Atish Das Sarma, Alex Fabrikant, Silvio Lattanzi, and Andrew Tomkins. Arrival and departure dynamics in social networks. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 233–242. ACM, 2013.
- [29] Xiaowei Xu, Nurcan Yuruk, Zhidan Feng, and Thomas AJ Schweiger. Scan: a structural clustering algorithm for networks. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 824–833. ACM, 2007.
- [30] Yiming Yang and Jan O Pedersen. A comparative study on feature selection in text categorization. In *MACHINE LEARNING-INTERNATIONAL WORKSHOP THEN CONFERENCE-*, pages 412–420. MORGAN KAUFMANN PUBLISHERS, INC., 1997.