Making Thin Data Thick:

User Behavior Analysis with Minimum Information

by

Reza Zafarani

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Graduate Supervisory Committee:

Huan Liu, Chair
Subbarao Kambhampati
Jure Leskovec
Guoliang Xue

ARIZONA STATE UNIVERSITY

August 2015

ABSTRACT

With the rise of social media, user-generated content has become available at an unprecedented scale. On Twitter, 1 billion tweets are posted every 5 days and on Facebook, 20 million links are shared every 20 minutes. These massive collections of user-generated content have introduced the human behavior's "big-data."

This big data has brought about countless opportunities for analyzing human behavior at scale. However, is this data enough? Unfortunately, the data available at the individual-level is limited for most users. This limited individual-level data is often referred to as *thin data*. Hence, researchers face a "big-data paradox", where this big-data is a large collection of mostly limited individual-level information. Researchers are often constrained to derive meaningful insights regarding online user behavior with this limited information. Simply put, they have to *make thin data thick*.

In this dissertation, how human behavior's thin data can be made thick is investigated. The chief objective of this dissertation is to demonstrate how traces of human behavior can be efficiently gleaned from the, often limited, individual-level information; hence, introducing an all-inclusive user behavior analysis methodology that considers social media users with different levels of information availability. To that end, the *absolute minimum information* in terms of both link or content data that is available for any social media user is determined. Utilizing *only* minimum information in different applications on social media such as prediction or recommendation tasks allows for solutions that are (1) generalizable to all social media users and that are (2) easy to implement. However, are applications that employ only minimum information as effective or comparable to applications that use more information?

In this dissertation, it is shown that common research challenges such as detecting malicious users or friend recommendation (i.e., link prediction) can be effectively performed using only minimum information. More importantly, it is demonstrated that

unique user identification can be achieved using minimum information. Theoretical boundaries of unique user identification are obtained by introducing *social signatures*. Social signatures allow for user identification in *any* large-scale network on social media. The results on single-site user identification are generalized to multiple sites and it is shown how the same user can be uniquely identified across multiple sites using only minimum link or content information.

The findings in this dissertation allows finding the same user across multiple sites, which in turn has multiple implications. In particular, by identifying the same users across sites, (1) *patterns that users exhibit across sites* are identified, (2) *how user behavior varies across sites* is determined, and (3) *activities that are observed only across sites* are identified and studied.

To my family

ACKNOWLEDGEMENTS

Research does not occur in a vacuum and my dissertation research has not been an exception. It has been a wonderful journey and it would have been impossible without the help of others. It is hard to thank every one, but I will try my best.

When I joined ASU, I had an intention to leave; I am glad and very fortunate that I did not. I am deeply indebted to all people who helped me find my path. I was the TA for two very supportive professors at ASU, Faye Navabi and Rida Bazzi. They both have great personalities and were great support during my first year here. In addition, I would like to thank Pat Langley, Jieping Ye, Pitu Mirchandani, Guoliang Xue, and Goran Konjevod. Pat reignited my interest in machine learning and AI, by providing a different perspective to machine learning and by introducing me to cognitive architectures and many interesting articles. This was complemented by Jieping Ye's mathematical view of machine learning that I grasped, thanks to the weekly NLP/machine learning seminars he organized. Jieping Ye also gave me great career advice during my job search. Pitu Mirchandani, Guoliang Xue, and Goran Konjevod with their fantastic optimization and algorithmic courses, impacted how I look at optimization and applied math. However, the most motivating factor for starting my research in social media mining, was my incredible advisor, Huan Liu. I am very fortunate to be his student and I will always be indebted.

Huan, not only has given me the ultimate freedom to pursue my interests, but has also taught me how to write papers and grant proposals, supervise students, view research, and establish myself. His support and encouragements were critical during the two years we spent writing the "social media mining" textbook. With him, you receive constant career advice and mentorship, not only about research, but about life. He has been the best mentor and friend I could have wished for.

I would like to thank my thesis committee, Subbarao Kambhampati, Guoliang Xue, and Jure Leskovec, for their help, encouragements, and the constructive feedback that they have provided. I have a diverse committee and they have all been an inspiration for my research. In particular, I would like to thank Rao, for his early advice on my thesis proposal that helped shaped the dissertation and his career advice that greatly helped with my job search. I will be always grateful for the time each committee member has spent to help improve my research.

I spent a great summer at Yahoo!. It was a great atmosphere for research and allowed me to interact with many great researchers. I owe this to my mentors, Vijay Narayanan, Lei Tang, Kun Liu, and Dragomir Yankov.

I have been a member of the Data Mining and Machine Learning (DMML) lab. I will dearly miss our Friday afternoon meetings and gatherings. The DMML members are amazing and I would like to thank them for their friendship, helpful suggestions, and support. It has been a great pleasure working with each and every one of them: Ali Abbasi, Nitin Agarwal, Salem Alelyani, Geoffrey Barbier, Ghazaleh Beigi, William Cole, Zhuo Feng, Magdiel Galan-oliveras, Huiji Gao, Pritam Gundecha, Xia Hu, Dinu John, Isaac Jones, Jiliang Tang, Lei Tang, Shamanth Kumar, Jundong Li, Fred Morstatter, Sai Moturu, Tahora H. Nazer, Ashwin Rajadesingan, Suhas Ranganath, Justin Sampson, Robert Trevino, Suhang Wang, Liang Wu, and Zheng Zhao.

Arizona's heat is bearable when you are in company of great friends. In particular, I would like to thank Ali Abbasi, Zahra Abbasi, Moeed Haghnevis, Yasaman Khodadadegan, Lei Yuan, and all my other friends for this companionship.

Last but not the least, I would like to thank my family for the support they provided through out my entire life and graduate school. In particular, I must acknowledge my wife Sara, without her love, encouragement, support, wit, and of course, her lasagna, there would be no dissertation and many happy moments today.

TABLE OF CONTENTS

CHAPTER Page

CHAPTER Page

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

Figure Page

Chapter 1

INTRODUCTION

If the doors of perception were cleansed every thing would appear to man as it is: infinite.

William Blake

With the rise of social media, the number of information outlets are increasing exponentially. The number of websites double every three months and the blogosphere doubles every 5 months. In addition to the increase in information outlets, user generated content has also become available at an unprecedented scale. Users exhibit tremendous activity patterns on social media sites. On Youtube, more that 1 billion unique user visits are observed every month and 100 hours of video are uploaded every minute. Similarly on Facebook, the 1.3 billion users spend 640 Millions minutes monthly on its 54 million pages. This boom in user activity is consistently observed in social media and has introduced the human behavior's "big-data."

1.1 Big-Data Paradox

This big data brings about with itself countless opportunities for analyzing human behavior. It is often perceived that with this big data one can study human behavior at scale. But, is this data enough? Unfortunately, even with this big data, the data at the individual-level is often limited for most users. For example, on Twitter, more than 40% of the users have never twitted and more than 60% of the users leave within their first month. This limited data at the individual level is denoted as *thin data*. Similar observations are not only observed in social media but also in other domains

where the majority of data is generated by a small percentage of the possible causes. This is often known as the *Pareto Principle* or the *80–20 Rule*. For example, it is known that the content generation on the web roughly follows the 80–20 rule, that is, 80% of the content is generated by 20% of the users.

The sole existence of such phenomenon can be easily explained using the statistical distributions governing this kind of data. It is well known that big data on social media and the web often follows a power-law distribution. A power-law distribution can be stated as

$$p(x) = Cx^{-\alpha} \qquad \text{for } x \geq x_{\min}, \tag{1.1}$$

where C is the normalization constant and α is known to be in range: $2 \leq \alpha \leq 3$.

As mentioned, content generation on the web roughly follows the 80–20 rule. Consider the fraction P of the user population that has generated at least x amount of information. This can be easily computed using the ccdf of the power-law distribution:

$$P(x) = \int_x^\infty Cy^{-\alpha}dy = (\frac{x}{x_{\min}})^{-\alpha+1}. \tag{1.2}$$

Now, consider the fraction of information that is generated by this population:

$$I(x) = \frac{\int_x^\infty yp(y)dy}{\int_{x_{\min}}^\infty yp(y)dy} = (\frac{x}{x_{\min}})^{-\alpha+2}. \tag{1.3}$$

Assuming $\frac{x}{x_{\min}} = z$, we can see that $P(x) = z^{-\alpha+1}$ and $I(x) = z^{-\alpha+2}$. Therefore,

$$I(x) = P(x)^{\frac{2-\alpha}{1-\alpha}}. \tag{1.4}$$

Solving for α, we get

$$\alpha = \frac{\ln I(x) - 2\ln P(x)}{\ln I(x) - \ln P(x)}. \tag{1.5}$$

When 80% of the data is generated by 20% of the users, we are assuming $I(x) = 0.8$ and $P(x) = 0.2$. Using equation 1.5, we get $\alpha = 2.16 \in [2, 3]$.

This observation shows that in power-law distributions most of data comes from a small set of users (20%). In other words, for the **other 80%** the data is very sparse. While we can have big datasets of user-generated content, for each user in the dataset, data is often very sparse. Hence, with this thin data, we face a *big-data paradox*, where we are inundated with large collections of thin data; however, behavior analysis for a specific users can be still challenging. This phenomenon happens on any social media site, but user data is not limited to a single site. Users on social media join multiple sites and their sparse data, becomes even more sparse, as it is distributed across sites. So, how can we analyze online users with this inherent data sparsity within and across sites?

To study user behavior comprehensively, one has to consider two important constraints. First, to be able to analyze behavior of all users, one has to be able to study them with the amount of information that is guaranteed to be available for each and every one of the users. In other words, we have to be able to study users with the *minimum information* that is always available for any user. Second, one has to be able to accumulate data that belongs to the same user across sites. Hence, the same users should be identified across sites, however, with minimum information.

In this dissertation, we analyze user behavior with limited information. Our goal is to efficiently glean traces of human behavior in the information that is available for each individual; therefore, including most users with limited information availability. As discussed, because user data is sparse and spread across multiple sites, our methods are constrained to utilize the absolute minimum information available in social media to (1) study user behaviors and to (2) identify the same users across sites. Identifying the same users across sites with minimum information has numerous benefits (see Section 1.2). Hence, in this dissertation, we focus more on identifying users across sites with minimum information. By connecting users across sites, we investigate the

unexplored patterns of user behavior that are exhibited only across sites. Next, we detail some other motivations for identifying the same individual across sites.

1.2 Motivation

The need for identifying corresponding users across different social media is multifold. For example, advertisement revenue is often a principal sources of finance for a sustainable social networking site. Web giants such as Google report a $50.57 billion dollar yearly ad revenue[1]; that is 91% of Google's annual revenue[2]. The same consistent pattern is observed among other internet sites such as Facebook or Yahoo!. Thus, internet sites are often interested in increasing the success rate of their ad campaigns.

It is well-known that the relevance of ads to the interests of individual users can directly impact the success of an ad campaign. To have relevant ads, it is required to have a good understanding of individuals, which can be achieved by profiling users. Though a growing number of people use social media, people use various social media for different purposes, and the information about an individual on each site is often limited. Though each site has only limited information about a user, other social media sites could provide complementary information for the user, and integrating information from various sites can help build better user profiles. However, for combining these sources of complementary information, one has to reliably identify corresponding user identities across social media sites. Companies such as Yahoo! often sign agreements with other companies to connect their user base for better marketing and a richer user experience. However, preliminary attempts to match users across sites even for these companies are challenging as users provide limited or

[1] http://bit.ly/1fbM89P

[2] http://bit.ly/1k5uVXI

no information for matching purposes [31].

In addition to the aforementioned marketing example, we illustrate the need using multiple examples.

1. **Enhancing Friend Recommendation.** Better friend recommendations can help increase user engagement in social media sites. Often, non-connected users that share mutual friends are recommended as potential friends. Consider the following example. John and Catherine are not connected and are both friends of Russ on social network S_1. Thus, Catherine seems a good candidate for recommendation to John on S_1. Catherine and John are also members of social network S_2 and are also not connected on S_2. Assume that Catherine and John share no mutual friends on S_2. With the information that we have from S_1, the recommendation algorithm could recommend Catherine to John on S_2, even though they share no mutual friends on S_2.

   This type of recommendation is only possible when there is cross-site complementary information. Cross-site friendship information will increase the recall of the friendship recommendation algorithm by recommending more known friends, as well as its accuracy by having more information about the network.

2. **Information diffusion.** Information diffusion is commonly measured within the context of a single social network. In reality, information can flow within and across different social networks. Thus, it is of interest to investigate whether information diffuses more within one network or across networks. Moreover, what type of information propagates more within a network and what type propagates more across networks?

3. **User Migrations.** Consider the migration of users in social media [76]. Users often migrate from one social network to another due to their limited time and

the better quality of service they receive at the destination network. Given a mapping among identities of users across these two networks and their membership dates (or dates where they started their activity on the destination network), a migration can be detected. The network from which users are migrating can decrease the migration rate by detecting it early and can also improve its site by introducing the additional features and services that the destination network provides. We discuss user migrations in Chapter 7.

4. **Multiple Network Group Interaction.** By connecting users across sites, one can can analyze group interaction across sites. Multiple-network group interactions can be viewed as an instance of single-net group interactions by combining the graph of all connected social networks. Hence, methods proposed for single network group interaction analysis [114] can be generalized for multiple networks.

5. **Analyzing Network Dynamics.** Dynamics of single-site social networks are well-studied in the literature. These networks are known to have a power-law degree distribution, a small average path length, and being highly clusterable [138]. However, users belong to multiple sites and these network properties need to be generalized to multiple networks. In particular, it is interesting to determine how close the dynamics of single networks are to that of multi-networks.

1.3 Contributions

In this dissertation, we make the following contributions:

1. **Identifying Users with Minimum Information:** we develop methods that can identify users across social media sites with minimum information. In particular, we investigate both link- and content-based method.

(a) **Link-Based Identification:** we introduce link-based techniques that employ minimum link information across sites. We investigate why (sub)graph isomorphism-based methods fail in social networks and demonstrate properties of social networks that make (sub)graph isomorphsim challenging. Finally, we introduce *social signatures* as different way of tackling user identification. In addition, we show how social signatures can be used to reconstruct graphs

(b) **Content-Based Identification:** we introduce behavioral modeling, a strategy for gleaning digital traces of human behavior in the content that they generate. In addition, we introduce MOBIUS, a content-based methodology that uses behavioral modeling for user identification with minimum information. We show that user identification with minimum content information is highly effective. Inspired by studies in psychology and sociology, we introduce a large set of computational features for efficient user identification with content information.

2. **Applications of Minimum Information:** Considering that users on social media are either normal or malicious, we investigate two representative applications that utilize minimum information for each category of users. For normal users, we investigate friend recommendation and show that minimum content information, combined with features that can detect social forces that result in friendships (homophily, influence, among others) can help detect future friends with performance comparable to state-of-the-art link prediction that has access to more information. For malicious users, we investigate literature from psychology and criminology, and combine that with machine learning and complexity theory, to efficiently detect malicious users, yet with minimum informa-

tion. Our results show that the information complexity of malicious users makes them distinguishable from normal users. The performance of the methodology for detecting malicious users is comparable to that of state-of-the-art malicious user detection techniques that have access to extra information.

3. **Analyzing User Behavior across Sites:** Combining user data across sites, allows us to analyze (1) *patterns*, (2) *variations*, and (3) *behaviors* across sites.

   (a) **Patterns across Sites.** We investigate the basic patterns of users that are clearly visible across sites. In particular, we demonstrate how users select sites to join across social media and how joining patterns can be used to predict *future* sites that users will join. In addition, we show the statistical distributions that govern how individuals are distributed across social media.

   (b) **Variations across Sites.** We investigate how users behavior varies across sites. In particular, we focus on the fundamental question of how friendships vary across sites and how the degree distribution changes across sites. In addition, we show how the average number of friends changes across sites. Our findings are aligned with studies in evolutionary psychology.

   (c) **Behaviors across Sites.** We investigate specific behaviors that are only observable across sites. In particular, we demonstrate how user migrations can be analyzed across sites and introduce a randomization-test based method for detecting migrations without ground truth. The method can be used in other domains and social media research, when ground truth is unavailable [143].

1.4 Roadmap

The remainder of this dissertation is organized as follows. We first determine ways to identify users across sites using minimum information. As the majority of data on social media is link or content, we dedicate two chapters to this topic. We first discuss minimum link information on social media in Chapter 2 and how it can be used to connect users across sites. We discuss minimum content information and ways to identify users across sites with it in Chapter 3. In Chapter 4, two well-known applications of utilizing minimum information is discussed. In particular, we consider one application for normal users (recommendation) and one application for malicious users (detecting malicious users). Once users are identified across sites, we discuss user patterns that are observed across sites in Chapter 5. We discuss how user behavior varies across sites in Chapter 6. Finally, in Chapter 7, we discuss particular behaviors that are only observed across sites. We conclude and provide directions for for future work in Chapter 8.

Chapter 2

UTILIZING MINIMUM LINK INFORMATION

A minimum put to good use is enough for anything.

Jules Verne

Connecting user identities across social media sites is not a straightforward task. The primary obstacle is that connectivity among user identities across different sites is often unavailable. This disconnection happens as most sites maintain the anonymity of users by allowing them to freely select usernames without revealing their real identities, and also because different websites employ different user-naming and authentication systems. Moreover, websites rarely link their user accounts with other sites or adopt Single-Sign-On technologies such as openID, where users can logon to different sites using a single user account (e.g., users can login to Google+ and YouTube with their GMail accounts). Regardless, there exists a mapping that connects user accounts of the same individuals across sites. *Can we find this mapping?*

In this chapter, we provide evidence on the existence of a mapping among identities across multiple social media, study the feasibility of finding this mapping, and illustrate and develop means for finding it.

Network structure and friendship information is known to carry information that could prove useful in many tasks, such as link and attribute prediction, spam detection, behavioral analysis, and studying group behavior. Recent studies have indicated that link-based methods outperform many other techniques on various tasks. For example, Agrawal et al. [8] show that their link-based algorithm exhibits a significant

Part of the content in this chapter has been published in the TKDD journal [145].

accuracy advantage over the classical text-based methods for mining certain newsgroups. Moreover, it is well established that link-based methods are more resilient to spam attacks [57]. Examples from social networks include systems that are designed using link-based methods that combat unwanted communications [96] or that guard against Sybil attacks [117, 133].

Recent interest in the information that immediate links (friends) carry about an individual has brought with it interesting results. When tracking link formation in online sites, Kossinets and Watts [74], and on a larger scale Leskovec *et al.* [78], found that the likelihood of forming links increases steadily as the number of common friends increases. In similar membership closure studies [36], it has been shown that the same increasing trend can be observed when analyzing the probability of joining a community as a function of the number of friends who have already joined. In another study, Backstrom *et al.* [16] show that the tendency of an individual to join a group is influenced not only by the number of friends the individual has within the community, but also crucially by how they are linked to one another.

These results suggest that it should be reasonable to use link information to identify users across social networks. The link information and in particular friends (immediate links) of an individual can help identify the individual across networks. From the computer science theory point of view, this problem appears to be an example of well-known graph isomorphism or subgraph isomorphism problem.

In the graph isomorphism, given two graphs $G_1(V_1, E_1)$ and $G_2(V_2, E_2)$, the goal is to find a bijection $f : V_1 \to V_2$ between the vertex sets of G_1 and G_2 such that if two vertices v_1 and v_2 are adjacent in G_1, i.e., $(v_1, v_2) \in E_1$, their mapped vertices are adjacent in G_2, i.e. $(f(v_1), f(v_2)) \in E_2$. In subgraph isomorphism, given graphs G_1 and G_2 the goal is to find whether G_1 contains a subgraph that is isomorphic to G_2. In terms of computational complexity, it is known that subgraph isomorphism

is NP-complete [32]; however, it is still unknown whether graph isomorphism is NP-complete. As subgraph isomorphism is known to be NP-complete, we can identify the same users across sites using graph isomorphism. Note that by using graph isomorphism detection methods, we are making strong assumptions. In particular, we are assuming that there is a one-to-one mapping between two networks and that there are degree correlations. We will evaluate the validity of these assumptions later in section 2.3. In addition, the best current known graph isomorphism algorithm, proposed by Eugene M. Luks in 1983 [14, 84], runs in exponential time ($2^{O(\sqrt{n \log n})}$). Executing such an algorithm is infeasible even for graphs that are far smaller than those observed in social media. For example, even for graphs with 50,000 nodes, Luks algorithm requires 10^{146} operations. Hence, one can approach this problem using heuristic-based methods that execute in polynomial time.

In this chapter, by considering the state-of-the-art heuristic-based methods for graph isomorphism and subisomorphism, we propose two heuristic-based link-based methods to identify individuals across social networks in Section 2.1. The first method utilizes local information (neighborhood data) to identify users across sites and the second method utilizes global information for finding individuals. We evaluate these methods in section 2.2 using synthetic and real-world datasets. The evaluation results indicate that further investigation is required regarding the graph structure of users that are shared across networks. We investigate graph structure of users that are shared across networks in section 2.3 along with some applicationsz. Building upon these investigations, we propose a representation, which we call *social signature* to identify users across sites. Social signatures are introduced in section 2.4. We conclude this chapter with a summary of contributions.

2.1 Link-based User Identification

Let us formally define the problem of identifying individuals across social media sites. Without loss of generality, we focus on two social media sites and a single individual in this study. This is reasonable because solving the problem of 2 sites can be easily generalized to the problem of n sites by considering n sites in a pairwise manner. The same argument holds for more than one individual. In traditional graph isomorphism or subgraph isomorphism there is access to the whole graph information for both graphs. However, this is an unrealistic assumption to start with. Following the tradition in machine learning and data mining research, we assume that we are given some available *labeled information.* This labeled information is the *known* part of a one-to-one relationship that connects users that co-exist on both networks. We call this labeled information "the mapping". The mapping for these two social networks contains a set of known individuals and their identities on both these networks; it basically denotes "who on this network is who on the other?". Finally, we focus on situations where the identity of the individual on one of these websites is known, e.g., profile of someone is known on Twitter; can we find his profile on Facebook?

When using link information, a social network $\mathcal{S}$ is represented using a graph $G_{\mathcal{S}}(V_{\mathcal{S}}, E_{\mathcal{S}})$ and the identity of an individual is represented using a node v (vertex) in this social graph, i.e., $v \in V_{\mathcal{S}}$. The mapping connects a node in the first graph to its corresponding node in the second's graph. We denote the first site as *base-site* and the second site as *target-site.*

Definition. ***Link-based User Identification.*** *Given two social media sites $\mathcal{S}_1$ (base-site) and $\mathcal{S}_2$ (target-site) and their respective social network graphs $G_{\mathcal{S}_1}(V_{\mathcal{S}_1}, E_{\mathcal{S}_1})$ and $G_{\mathcal{S}_2}(V_{\mathcal{S}_2}, E_{\mathcal{S}_2})$, a mapping $\mathcal{M} \subseteq V_{\mathcal{S}_1} \times V_{\mathcal{S}_2}$ that identifies a subset of users across these networks and an individual* u *whose identity (a vertex $v_i \in V_{\mathcal{S}_1}$) we know on $\mathcal{S}_1$*

(base-node), a link-based user identification procedure attempts to resolve the identity (a vertex $v_j \in V_{\mathcal{S}_2}$) of u *on* S_2 *(target-node).*

We introduce two techniques to identify users across sites based on link information. The first technique uses only local information (i.e., neighborhoods and shared friends) to identify users across sites. The second techniques utilizes global network information (i.e., the whole graph) to identify users across sites.

2.1.1 A Local Link-based Method for Identifying Users

We introduce an iterative method for identifying users across sites using local link information. The method considers users across sites that share most mutual friends across sites as identities of the same individual. Our intuition is that as users join multiple sites, it is more likely for them to become friends with individuals that they have befriended on other sites. So, nodes that share most common friends across sites are more likely to be the same user. Inspired by the success of methods that utilize common friends within one site, our method employs the same heuristic across sites. The method's pseudocode is outlined in Algorithm 1, in which, $\mathcal{F}(i, \mathcal{S})$ denotes friends of user i on site $\mathcal{S}$.

The method starts from the users not in the mapping, and it acts similar to the semi-supervised learning algorithms and in particular co-training [148]. In the pseudocode, the users already mapped in S_1 (S_2) are denoted as $\mathcal{M}_1$ ($\mathcal{M}_2$), and the users not mapped are denoted as $V_{\mathcal{S}_1} \setminus \mathcal{M}_1$ ($V_{\mathcal{S}_2} \setminus \mathcal{M}_2$).

The method then maps two users to one another across networks based on their number of friends inside the mapping. Here, we find two users, one on each network, who have the most number of friends among users in the mapping, and we assume these users represent the same individual.

Since these two users are assumed to represent the same individual, they are added

Input: $G_{\mathcal{S}_1}(V_{\mathcal{S}_1}, E_{\mathcal{S}_1})$, $G_{\mathcal{S}_2}(V_{\mathcal{S}_2}, E_{\mathcal{S}_2})$, Mapping $\mathcal{M}$, $v_1 \in V_{\mathcal{S}_1}$ (base-node)

Output: $v_2 \in V_{\mathcal{S}_2}$ (target-node) or NIL

shouldContinue = True, *targetNode* = NIL;

while *shouldContinue* **do**

$\mathcal{M}_1 = \{i|(i,j) \in \mathcal{M}\}$, $\mathcal{M}_2 = \{j|(i,j) \in \mathcal{M}\}$; `% Nodes in the Mapping`

if $V_{\mathcal{S}_1} \setminus \mathcal{M}_1 = \emptyset$ *or* $V_{\mathcal{S}_2} \setminus \mathcal{M}_2 = \emptyset$ **then**

shouldContinue = False, break while; `% No More Users Left`

end

`% Find Users with the Maximum Number of Friends among Mapping Nodes`

$x = \arg\max_i |\mathcal{F}(i, \mathcal{S}_1) \cap \mathcal{M}_1|$, s.t., $i \in V_{\mathcal{S}_1} \setminus \mathcal{M}_1$;

$y = \arg\max_j |\mathcal{F}(j, \mathcal{S}_2) \cap \mathcal{M}_2|$, s.t., $j \in V_{\mathcal{S}_2} \setminus \mathcal{M}_2$;

if $x = v_1$ **then**

targetNode = y, *shouldContinue* = False, break while; `% Target Found`

end

$\mathcal{M} = \mathcal{M} \cup \{(x, y)\}$; `% Add an Identified Pair to the Mapping`

end

Return *targetNode*;

Algorithm 1: The Link-based Iterative Method for Identifying Individuals

to the mapping.

This process is continued until no further user is identified on both networks ($V_{\mathcal{S}_1} \setminus \mathcal{M}_1 = \emptyset$ or $V_{\mathcal{S}_2} \setminus \mathcal{M}_2 = \emptyset$), or the required user is found on both networks.

The method only considers the local neighborhood of nodes. Our next method considers global network structure to identify users across sites.

2.1.2 A Global Link-based Method for Identifying Users

The local algorithm only considers nodes in the mapping that are 1-hop away. While this is more realistic for using minimum information, the algorithm can be

modified in order to consider nodes in the mapping that are more than one-hop away. For each node, the number of nodes in the mapping that are 1...k hops away can be computed and a k-dimensional vector can be used to represent users. The distance between these vectors could help identify identities of the same individual and in turn, grow the mapping. A more sophisticated approach is to use the topology of the induced subgraphs of the nodes in the mapping and the nodes connected to them. We can assume that the two networks are two different views of the same underlying structure. In other words, we assume that users possess a specific friendship behavior and the way they befriend others across different networks are just different ways that they exhibit this behavior. We expect these networks to be highly correlated and hence, a transformation between them can be computed.

The base-site and target-site graph can be represented as an adjacency matrix. Let us call these matrices A_1 and A_2. An additional preprocessing step is usually taken in order to extract structural features of the graphs. For preprocessing purposes, the normalized Laplacians, $\mathcal{L}_1$ and $\mathcal{L}_2$, for each graph is calculated. The normalized Laplacian $\mathcal{L}$ for adjacency matrix A is calculated as follows,

$$\mathcal{L} = D^{-1/2} L D^{-1/2}, \tag{2.1}$$

$$L = D - A, \tag{2.2}$$

where D, also known as the degree matrix, is a diagonal matrix where each entree on the diagonal represents the degree of the node. L here represents the unnormalized Laplacian matrix. After computing the normalized Laplacians, the k top eigenvectors of the matrix are extracted and are used instead of the adjacency matrix. This matrix can better represent the structural features of the graph when compared with the adjacency matrix [115]. Different k's were tested in our experiments, $k = 3, 5, 20, 50, 100, 200, 500$. For values above 50, our results did not improve

much; hence, we used $k = 100$ for our experiments. Let us call these new matrices X_1 and X_2. We take the mapping part of these two matrices (corresponding mapped rows) and call them X_1^m and X_2^m. Assuming there exists a linear transformation, the transformation W can be found using the following optimization,

$$\min ||X_1^m W - X_2^m||_2. \tag{2.3}$$

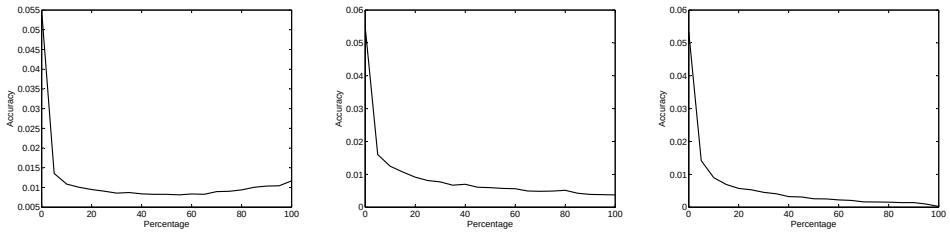
The transformation W can be efficiently computed using a least square approximation. After the weights are obtained, the unmapped part of matrix X_1 can be multiplied by W and then compared with the unmapped part of X_2. Rows (users) with the highest similarity are assumed to be the same individual.

2.2 Evaluation

In this section, we evaluate the proposed methods using both synthetic and real-world datasets.

2.2.1 Evaluation with Synthetic Data.

To conduct a systematic evaluation of the proposed methods, we generated a set of synthetic datasets. These synthetic datasets must contain mapping information (labeled data). For synthetic dataset generation, we adhere to the following procedure: 1) a real-world social network was gathered and used as the base-site; 2) the base-site's network was copied as the target-site; and 3) noise was introduced on the target-site. Three common types of noise were employed, namely: i) randomly adding edges to the target-site with probability p, ii) randomly removing edges from the target-site with probability p, or iii) randomly rewiring [124] edges from the target-site with probability p, $0\% \leq p \leq 100\%$. In rewiring, for every disjoint pair of random edges (a, b), (c, d), we swap their end points to get new edges, (a, c), (b, d). This makes sure that



(a) Edge Addition (b) Edge Removal (c) Edge Rewiring

Figure 2.1: Prediction Accuracy for Different Percentage of Edges Added, Removed, or Rewired

the degrees are preserved for every node in the target-site graph. Based on the types of noise introduced and the probability value p, we call these datasets SYN_ADD(p), SYN_REMOVE(p), and SYN_REWIRE(p), respectively. The mapping is obvious in the case of synthetic data, and for every node in the base-site, the mapping connects it to the corresponding copied node in the target-site. For the real-world network used in our synthetic dataset generation, we employed a collection of 11 large scale social media datasets (see Table 2.1) obtained from the social computing data repository [137].

We conduct experiments on synthetic data to verify if our link-based methods perform effectively in a controlled environment. We start with no noise ($p = 0$) and notice that the local method is not even accurate for cases where no noise is introduced. This is a result of many nodes having the same number of friends among mapping nodes. Table 2.1 shows the accuracy rate of both methods in the case where no noise is introduced over all synthetic datasets. The table shows that the local method is, in eight out of eleven cases, less than 2% accurate, and the best accuracy rate obtained is less than 7%. On the contrary, the global model is highly accurate with no noise and is at least 79% accurate and at times, up to 98% accurate. Next, we added noise. We used BlogCatalog dataset as the real network required for synthetic data generation. Part of the mapping was used for training and the rest

Table 2.1: Prediction Accuracy for Different Social Networks

Site	Nodes (Mapping Size)	Edges (Friendship Links)	Accuracy (Local Method)	Accuracy (Global Method)
Blogcatalog	88,784	4,186,390	6.93 %	89.3%
Buzznet	101,168	4,284,534	5.11 %	79.7%
Digg	116,893	7,261,524	1.81 %	91.4%
Douban	154,907	654,188	1.78 %	84.1%
Flixster	2,523,386	9,197,338	0.57 %	96.6%
Friendster	100,199	14,067,887	0.32 %	91.3%
Foursquare	106,218	3,473,834	0.53 %	98.0%
Hyves	1,402,611	2,777,419	0.37 %	95.0%
Last.fm	108,493	5,115,300	0.76 %	95.6%
Livemocha	104,438	2,196,188	4.57 %	96.4%
YouTube	1,138,499	2,990,443	4.57 %	90.5%

for testing. 10-fold cross-validation was used and the average accuracy for correctly predicting identities in the testing part of the mapping was recorded. Figure 2.1 depicts these accuracy rates for the local method and for cases where with different probabilities, edges were being added, removed, or rewired. As seen in these figures, the local method performs quite poorly on synthetic data. The average accuracy rates for SYN_ADD(p), SYN_REMOVE(p), and SYN_REWIRE(p) were 4%, 1%, and 1%, respectively. The results did not improve much for the global method. With $p = 0.5$, the accuracy rates for SYN_ADD(p), SYN_REMOVE(p), and SYN_REWIRE(p) were 6%, 10%, and 0.01%, respectively. Next, we evaluate the performance of the methods with real-world datasets.

Table 2.2: Real-World Dataset Properties

Dataset	BlogCatalog network Size	Flickr network Size	Mapping Size $\lvert\mathcal{M}\rvert$
$BF3Hop$	88,784 users	564,491 users	1,747 individuals $\lvert\mathcal{I}_1\rvert$
$BF1Hop$	1,455 users	630 users	546 individuals $\lvert\mathcal{I}_2\rvert$

2.2.2 Evaluation with Real-World Data.

We gathered two real-world datasets. For collecting real-world datasets, we require additional mapping information about identities across social media sites. Fortunately, there exist websites where users have the opportunity of listing their identities (user accounts) on different social networks. For instance, on Facebook users can list their usernames on different sites. This can be thought of as *labeled* data for our learning task since it provides the accurate mapping for our experiments. In addition to labeled data, these websites provide strong evidence on the existence of a mapping between identities across social media sites. Later on, in Section 3.3.1, we discuss the procedure for collecting mapping information in detail. From sites that provide such mapping information, we gathered individuals that had account on two sites: Flickr and BlogCatalog, due to their large network size and many overlaps.

We collected two disjoint sets of individuals. All individuals had accounts on both BlogCatalog and Flickr. We call these sets $\mathcal{I}_1$ and $\mathcal{I}_2$ ($\mathcal{I}_2 \cap \mathcal{I}_2 = \emptyset$). For each member of these sets, we collected their identity on both BlogCatalog and Flickr. Then for individuals in $\mathcal{I}_1$ and for each of their two identities, we collected all the users who were within a 3-hop distance in the respective network using a Breadth-First-Search crawling procedure [91]. For $\mathcal{I}_2$, however, we only crawled users who were within a 1-hop distance (immediate friends). Hereafter, we will refer to the network datasets created from $\mathcal{I}_1$ and $\mathcal{I}_2$ as $BF3Hop$ and $BF1Hop$, respectively. Table 2.2 provides some statistics about the cardinalities of these datasets.

Table 2.3: Performance of Link-Based Methods on Real-World Datasets

Dataset	Local Method	Global Model
$BF3Hop$	≈ 0	≈ 0
$BF1Hop$	0.3%	0.6%

These datasets help showcase the effect of non-immediate link information on the performance of our proposed algorithms. This is true since $BF3Hop$ contains non-immediate information, whereas $BF1Hop$ lacks this property.

We evaluate both methods on real-world datasets. We apply the local method to our real-world datasets and 10-fold cross-validation is employed to measure accuracy. The method failed on both datasets with an average accuracy rate of 0.3% on $BF1Hop$ and $\approx$ 0 on $BF3Hop$. Similarly, we evaluated the global model. However, the results did not improve much. For real-world datasets, the accuracy rate were 0.6% on $BF1Hop$ and 0% on $BF3Hop$. Table 2.3 summarizes the results of link-based methods on the real-world datasets.

We have shown that using both local and global information, poor performances are expected when using real-world datasets. The question is whether *there are any properties in real-world datasets that need to be considered in order to obtain higher accuracy rates*. We investigate this question next.

2.3 Investigating Properties of Real-World Datasets

To further real-world datasets, let us present various hypotheses regarding the properties of the users that are in the mapping. These link-related properties that identities share when representing the same individual across different networks can be employed when designing methods for identifying users across social networks. Each of these hypotheses is empirically evaluated. The observations gathered while evaluating these hypotheses can be used later to help construct link-based methods.

To simplify the notation in these hypotheses, let x_i be a user (node), and $\mathcal{F}(x_i, \mathcal{S})$ the set of friends user x_i has on site $\mathcal{S}$. For two users $x_1 \in \mathcal{S}_1$, $x_2 \in \mathcal{S}_2$ that belong to two different sites, we define the concept of *shared-friends across networks*. In this case they are the set of people who co-exist on both $\mathcal{S}_1$ and $\mathcal{S}_2$ and are friends with both x_1 and x_2. For clarity, shared friends are depicted in Figure 2.2. In this figure, the mapping consists of 3 pairs and is shown using dashed lines and black circles denote shared friends between x and y. The concept is formalized as follows,

$$\begin{aligned}\mathcal{SF}(x_1, x_2) \quad = \quad & \{(y_1, y_2) | y_1 \in \mathcal{S}_1, y_2 \in \mathcal{S}_2, , (y_1, y_2) \in \mathcal{M}, \\ & y_1 \in \mathcal{F}(x_1, \mathcal{S}_1), y_2 \in \mathcal{F}(x_2, \mathcal{S}_2)\}.\end{aligned}$$

We also define the concept of *crossed-over friends* for a user x. These are the corresponding identities, on the *other* site, for the friends of x who are members of both sites. So, if x is a member of $\mathcal{S}_1$, this set includes identities on S_2 for those friends of x that are members of both sites. Formally,

$$\mathcal{CR}_{\mathcal{S}_1 \to \mathcal{S}_2}(x) = \{y | y \in \mathcal{S}_2, \exists x' \in \mathcal{F}(x, \mathcal{S}_1), s.t., (x', y) \in \mathcal{M}\}.$$

This definition is bidirectional. Note that if users x_1 and x_1' belong to the same individual, i.e., $(x_1, x_1') \in \mathcal{M}$, then the value of $|\mathcal{CR}_{\mathcal{S}_1 \to \mathcal{S}_2}(x)|$ is *not* necessarily equal to $|\mathcal{CR}_{\mathcal{S}_2 \to \mathcal{S}_1}(x')|$. In general, for any two users $x \in \mathcal{S}_1$ and $y \in \mathcal{S}_2$ there could be no relationships between the values of $|\mathcal{CR}_{\mathcal{S}_1 \to \mathcal{S}_2}(x)|$, $|\mathcal{CR}_{\mathcal{S}_2 \to \mathcal{S}_1}(y)|$, and $|\mathcal{SF}(x, y)|$, e.g., consider the situation where there are no shared friends but different number of crossed-over friends. Similarly, in Figure 2.2, circles in the right dashed oval denote $\mathcal{CR}_{\mathcal{S}_1 \to \mathcal{S}_2}(x)$, and circles in the left dashed oval represent $\mathcal{CR}_{\mathcal{S}_2 \to \mathcal{S}_1}(y)$. Given these formal definitions, we present our hypothesis next.

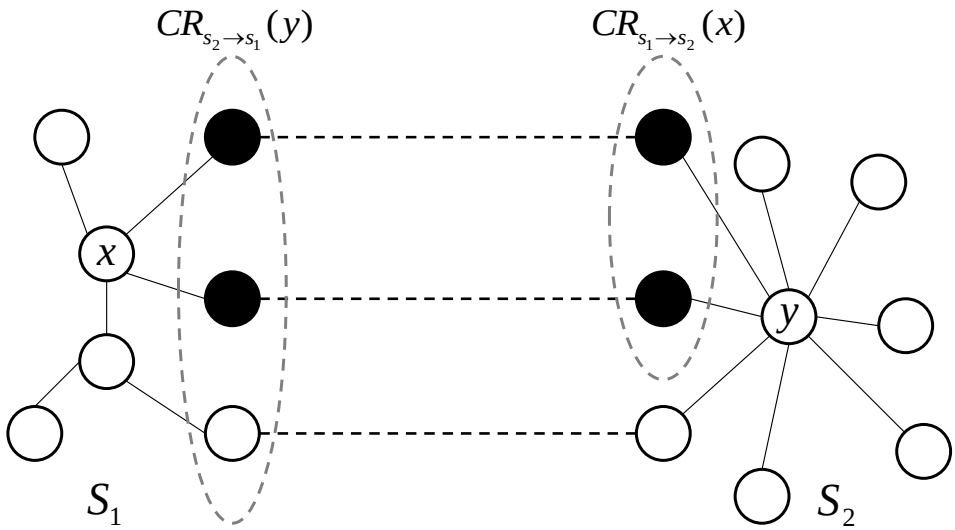


Figure 2.2: A visualization of two social networks and the mapping. Social network S_1 consists of the nodes on the left and social network S_2 consists of the nodes on the right. Dashed lines denote the mapping $\mathcal{M}$ ($|\mathcal{M}| = 3$), solid circles denote shared friends $\mathcal{SF}(x, y)$, circles in the right dashed oval denote crossed-over friends $\mathcal{CR}_{\mathcal{S}_1 \to \mathcal{S}_2}(x)$, and circles in the left dashed oval denote $\mathcal{CR}_{\mathcal{S}_2 \to \mathcal{S}_1}(y)$.

2.3.1 Hypotheses Verification

$\mathcal{H}_1$**: There is a correlation between the number of friends of the same individual across networks.** To test this, for all the users in the mapping, we analyze the number of friends they have in both networks. A Pearson correlation analysis revealed that the number of friends are uncorrelated across networks for the same individual. The correlation coefficient ρ was 0.038 for $BF3Hop$ and 0.186 for $BF1Hop$. For a randomly generated mapping, the correlation coefficient ρ was 0.007 for $BF3Hop$ and 0.019 for $BF1Hop$. This shows that there is no strong correlation among the number of friends across networks for the same individual.

$\mathcal{H}_2$**: There is a correlation between the percentage of friends of the same individual on each network that are shared across networks.** To verify this, we first enumerated the number of friends shared between identities of the same individual across networks, i.e., we calculated $\mathcal{SF}(x_1, x_2)$ for all $(x_1, x_2) \in \mathcal{M}$, and

Table 2.4: Friends Shared across Social Networks

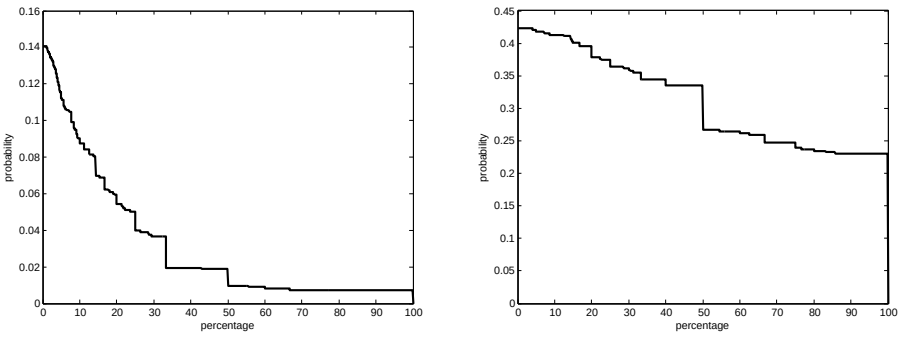
Property	$BF1Hop$	$BF3Hop$
Average number of friends shared	1.14	.18
Average number of friends on Flickr	3.08	26.22
Average number of friends on BlogCatalog	24.89	141.41
Average % Flickr friends shared	37%	2%
Average % BlogCatalog friends shared	9%	.2%
Maximum number of friends shared	32	30
Minimum number of friends shared	0	0
Standard deviation of the number of friends shared	2.30	1.09

for both datasets. Table 2.4 shows some statistics about these shared friends.

As shown in this table, the average number of friends shared is at most around 1 in the datasets. Having at most one shared friend suggests that the friends that are shared across both social networks, in the best case, can form connected components on both networks. Starting from an individual in the mapping and its two identities, a Breadth-First-Search procedure on each network should be able to traverse many other users in the mapping.

The table also shows, for both BlogCatalog and Flickr, the average values for the percentage of users' friends that were shared. The small values of these percentages denotes that many friends on both social networks do not cross over into the other [1]. A correlation analysis on these percentages across networks, when there was at least one friend shared, showed that $\rho \approx 0$ for both datasets. Again, the value was close to the correlation coefficient for both datasets when the mapping was randomly generated

[1]This could also be due to the small size of the mapping in the dataset; however, when collecting the initial set of mapping users from BlogCatalog we made sure a connected component was collected to reduce the effect of this phenomenon.



(a) Flickr to BlogCatalog (b) BlogCatalog to Flickr

Figure 2.3: Target User Connection Probability to Different Fractions of Crossed-Over Friends

and shows that there is no strong correlation between percentages.

$\mathcal{H}_3$: The target-node is connected to the crossed-over friends of the base-node. Here, we conjectured intuitively and based on previous evidence from the social sciences (e.g., see Herding Behavior [45]), that when users join various social networks, their friends also follow them and join these networks. We assume that if one analyzes the connections of crossed-over friends, one might be able to find the user on the target network.

For evaluating this hypothesis, for all pairs $(x_1, x_2) \in M$, $x_1 \in \mathcal{S}_1$, $x_2 \in \mathcal{S}_2$, we first extracted all crossed-over friends of x_1 ($\mathcal{CR}_{\mathcal{S}_1 \to \mathcal{S}_2}(x_1)$). Then for all members of this set $y \in \mathcal{CR}_{\mathcal{S}_1 \to \mathcal{S}_2}(x_1)$, we checked whether the target-node x_2 is connected to y, i.e., $x_2 \in \mathcal{F}(y, \mathcal{S}_2)$. In other words, we are trying to calculate the probability of identifying the target-node by analyzing the connections of the crossed-over friends of the base-node.

It turns out that in both datasets, the probability of target-node x_2 being connected to *all* the friends of the base-node that crossed-over is always less than 5%. Furthermore, the probability of x_2 being connected to *at least one* of the friends is still very low for both datasets (around 45% for $BF3Hop$ dataset at its best). Figure

2.3 shows the probability of the target-node being connected to different fractions of crossed-over friends of the base-node for the $BF3Hop$ dataset: (a) friends crossed-over from Flickr to BlogCatalog, and (b) from BlogCatalog to Flickr. For instance, Figure 2.3(b) shows that in the best case, one has less than a 45% chance to find the target-node based on crossed-over friends of the base-node. This is because in 55% of the cases, the target-user is not even connected to these friends. The 45% is reduced to less than 5% in the worst case. But, when the user is connected to these friends, is it easy to distinguish him from others who are also connected to these friends? This brings us to our next hypothesis.

$\mathcal{H}_4$: If the target-node is connected to the crossed-over friends of the base-node, how easily can it be identified? To answer this question, we further analyzed these crossed-over friends and ranked other users in the target network based on the number of connections they have to them. In these ranked users, we found that in $BF1Hop$ and on average, the target user x_2's ranking is 19 for friends who cross-over from BlogCatalog to Flickr and 25 in the opposite direction. These averages showed a dramatic increase in $BF3Hop$ and were 272 and 251, respectively. Furthermore, in $BF1Hop$, x_2 was the top ranked user in only 23% of the cases where friends crossed over from BlogCatalog to Flickr and 24% of the cases where the crossing over took place in the opposite direction. These percentages dropped to 9% and 8% for the $BF3Hop$ dataset, respectively. Note that even if one is successful in finding that the target user among the nodes that are connected to the crossed over friends of the base-node, it is very unlikely to correctly identify the target user. For example, in case of friends who crossed over from BlogCatalog to Flickr in $BF3Hop$, this probability is at most $45\% \times 9\% \approxeq 4\%$.

The heuristic-based methods proposed in this chapter are inspired by (sub)graph isomorphism detection methods. They utilize mapping information and graph struc-

ture in a semi-supervised manner. However, the results from the hypotheses verification suggest that such methods that deal with link information (local or global) can perform poorly when solving the user identification problem. Based on the evidence that we gathered, it is very unlikely to come up with new isomorphism detection methods that uses mapping information or graph structure to perform significantly better than the presented link-based methods. While our results clearly show that link information is not always useful, there are exceptions where link information in mapping or graph structure can be utilized for user identification across sites. This has been witnessed in recent studies where link-information has been successfully utilized to identify individuals across sites [82, 113, 146].

The results showed that counter-intuitively, link information is not sufficient when using mapping information or graph structure to identify individuals across networks. In addition, our constraint of introducing link-based methods that utilize minimum information was not fully realized, especially in the case of the global link-based method. Therefore, this suggests approaching the problem from a different angle. Our view is that there might be properties of nodes that remain unique across networks. Once these properties are identified, one can utilize them to identify the same nodes across networks. We investigate such properties next.

2.4 Social Signatures

What is the absolute minimum information a graph node can have about its surrounding network? Clearly, it is the degree of the node. As we discussed, if this minimum information becomes unique, then it can be utilized to identify users across sites. Hence, we first investigate degree uniqueness in graphs.

2.4.1 Degree Uniqueness

The node's degree is the absolute minimum network information that we can have about a node in a graph. The degree can at times help identify nodes uniquely in graphs. For example, the node with the maximum degree[2], as in a popular celebrity on Twitter, can be uniquely identified solely based on its degree. Hence, if a graph is anonymized and only node degrees are available, we can de-anonymize the node with the maximum degree.

To investigate node uniqueness realistically, we make a series of assumptions. We assume the problem is solved for large-scale real-graphs or synthetic graphs that exhibit properties of real-world graphs. Both types of graphs are known to exhibit specific properties such as a having a power-law degree distribution. Hence, we assume that the graph has a power-law degree distribution. In a graph with a power-law degree distribution, the probability of observing a degree d is

$$p(d) = Cd^{-\alpha}, \tag{2.4}$$

where C is a normalizing constant and $2 < \alpha < 3$ in real-world networks. To measure degree uniqueness, we identify the degrees that can be unique in a power-law degree distribution. The following theorem investigates degree uniqueness in graphs with power-law degree distribution.

Theorem 1. *For any graph with a power-law degree distribution, the proportion of degrees that are unique is* $\theta(n^{\frac{1-\alpha}{\alpha}})$.

Proof. Consider the **first** degree d_E that is expected to be unique in a power-law distribution. For degree d_E to be unique, the probability of observing it in the graph

[2]Assuming there is one such node.

should be $\frac{1}{n}$, where n is the number of nodes in the graph. Thus,

$$p(d_E) = Cd_E^{-\alpha} = \frac{1}{n}. \tag{2.5}$$

Solving for d_E, we obtain

$$d_E = n^{1/\alpha}. \tag{2.6}$$

This is similar to the bound obtained by Aielo et al. [10] for such unique degrees in large graphs. According to Equation 2.4, degrees larger than d_E are less likely to be observed. So, any degree that is larger than d_E is also unique in the graph. In particular, the largest degree in the graph should also be unique. To obtain the largest degree in the graph, we can use the ccdf [101] of the power-law distribution,

$$P(d) = \int_{k=d}^{\infty} p(k)dk = (\frac{d}{d_{\min}})^{-(\alpha-1)}, \tag{2.7}$$

where $d_{\min}$ is a constant.[3] As there are no other degrees larger than the maximum degree, we can use the ccdf of the power-law distribution (Equation 2.7) to compute its value,

$$P(d_{\max}) = \int_{k=d_{\max}}^{\infty} p(k)dk = \frac{1}{n}. \tag{2.8}$$

Solving for $d_{\max}$, we obtain

$$d_{\max} = n^{1/(\alpha-1)}. \tag{2.9}$$

All the degrees between d_E and $d_{\max}$ are unique. Therefore, the probability of degrees being unique in a power-law graph is $P(d_E) - P(d_{\max})$ that can be shown, with basic algebra, is equal to

$$\begin{aligned} P(d_E) - P(d_{\max}) &= \int_{k=n^{1/\alpha}}^{\infty} p(k)dk - \int_{k=n^{1/(\alpha-1)}}^{\infty} p(k)dk \\ &\in \theta(n^{\frac{1-\alpha}{\alpha}}). \end{aligned} \tag{2.10}$$

□

[3]Power-law distribution cut-off.

Because $\alpha > 1$, the exponent of the term $\theta(n^{\frac{1-\alpha}{\alpha}})$ is negative. Therefore, as the size of the graph (n) grows, the proportion of nodes with a unique degree shrinks to zero.

Corollary 2. *As the number of nodes grow in large-scale graphs, nodes become less and less unique.*

We have theoretically shown that degrees become less unique as graphs grow. This can be empirically tested using large-scale networks. For this purpose, we take 10 social network graphs, all publicly available from social computing data repository [137] and SNAP[4] and measure their degree uniqueness. We make sure that the graphs are of different sizes so that the effect of network growth on degree uniqueness shown in Corollary 2 can be observed. For each graph, we measure how unique degrees are. For instance, in a graph with completely distinct degrees, the uniqueness is 100%.

The results are provided in Table 2.5. The results confirm our theoretical findings. That is, as the size of the network (n) grows, the uniqueness, denoted as U in Table 2.5, drops.

The number of friends can at times help uniquely identify users on large graphs. This unique identification can be easily achieved in case of a popular celebrity with millions of friends on social networks. However, we have shown both theoretically and empirically, that the number of friends (i.e., degree) in general cannot help with unique identification in large graphs. Hence, one needs to add more information for unique identification. For that purpose, we will propose social signatures next.

2.4.2 Revisiting Minimum Information

As degrees tend to become less unique in larger graphs, we can investigate if one can uniquely identify nodes by adding a little more information about them in

[4]http://snap.stanford.edu/

Table 2.5: Degree Uniqueness for Different Social Networks

Site	**Nodes**	**Degree Uniqueness** (U)
Blogcatalog	88,784	$3 \times 10^{-3} < U < 6 \times 10^{-3}$
Buzznet	101,168	
Livemocha	104,437	
Douban	154,908	$2 \times 10^{-4} < U < 6 \times 10^{-4}$
Foursquare	639,014	
Digg	771,231	
YouTube	1,157,827	
Hyves	1,402,693	$U < 1.4 \times 10^{-4}$
LiveJournal	4,036,537	
Friendster	5,689,532	

addition to their degrees. Consider the graph provided in Figure 2.4. In this graph, we can represent node A with its degree: $d_A = 3$. We can instead represent A with the friends that it has. Let $f(v)$ denote the set of friends for node v. In this case, the new representation would be $f(A) = \{B, C, D\}$. Here, we are assuming that each node in the graph has a circle around its friends. These circles are shown for nodes B, C, and D in Figure 2.4. By representing a node using its friends, we are implying that the node is in the intersection of the respective circles of the friends. In Figure 2.4, node A has friends B, C, and D; therefore, it is in the intersection of the circles of these nodes.

While this representations carries a natural explanation, it is not mathematically well-defined. This definition is self-referential, meaning that for uniquely finding A, we need to find B, C, and D, and for finding them, we need to find their friends, and so on. To resolve this issue, we can assume that the circle that surrounds the

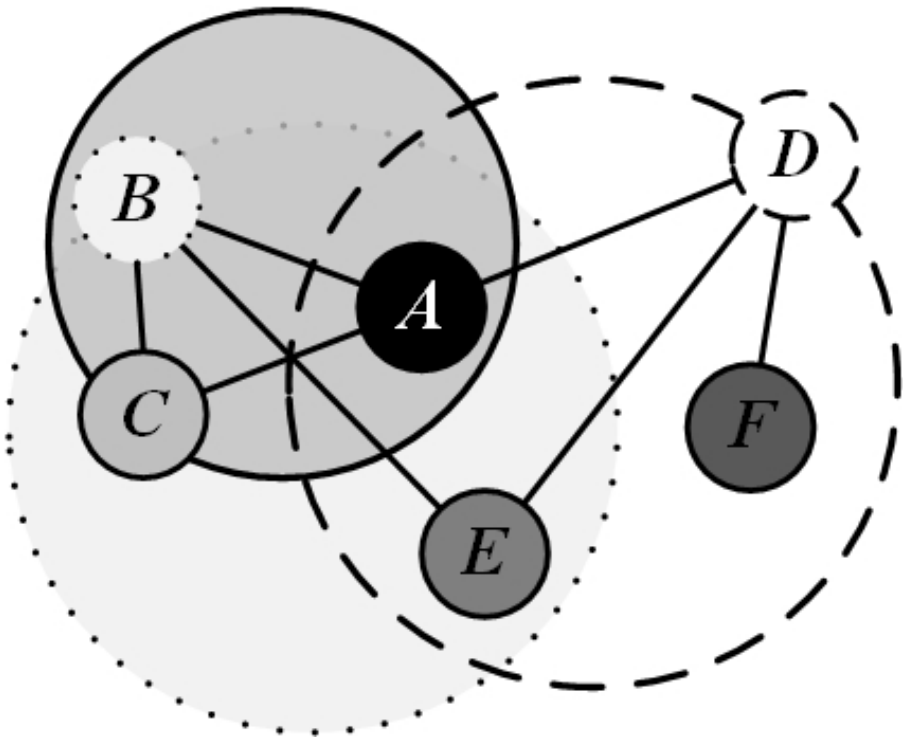


Figure 2.4: A Sample Graph

friends of each user, is as big as the number of friends that the user has. For instance, in Figure 2.4, the big circle around friends of B has size 3 and the smaller circle around friends of C has size 2. Again, a user is in the intersection of these circles corresponding to different friends. This way instead of representing node A with its friends $f(A) = \{B, C, D\}$, we can represent it with the size of the circles (i.e., number of friends) of its friends: $S(A) = \{1, 2, 3\}$. This representation basically shows the number of friends that the friends of A have. We denote this representation as the *Social Signature* of A.

Definition 1. *The social signature of node v in a graph, denoted as $S(v)$, is the multiset of the number of friends that the friends of v have.*

As a direct result, a node with an empty social signature, is an orphan node in the graph. The following example clarifies computation of social signatures in a graph.

Example 2.4.1. *Consider the graph in Figure 2.4. In this graph, the social signatures for each node can be calculated as follows:*

$$S(A) = \{2, 3, 3\}, \tag{2.11}$$

$$S(B) = \{2, 2, 3\}, \tag{2.12}$$

$$S(C) = \{3, 3\}, \tag{2.13}$$

$$S(D) = \{1, 2, 3\}, \tag{2.14}$$

$$S(E) = \{3, 3\}, \tag{2.15}$$

$$S(F) = \{3\}. \tag{2.16}$$

Next, we will discuss some properties of social signatures.

2.4.3 Properties of Social Signatures

Social signatures have intuitive properties that connect them to well-known concepts in networks. Some network and graph properties can be directly computed from social signatures and some can be approximated or bounded using them. The connection between social signature properties and network concepts can be made rigorous through extremal graph theory [23]. As these are straightforward properties, we demonstrate them through examples. These properties are presented in an increasing order of complexity.

Let $S(v) = \{d_1, d_2, \ldots, d_k\}$ denote the social signature for node v. Let $l(v) = k$ denote the length of the social signature of v. In other words, $l(v)$ counts the number of elements in the social signature for v. Let,

$$S_{\sum}(v) = \sum_{i=1}^{k} d_i \tag{2.17}$$

denote the summation of degrees in the social signature of v. Then, the following properties connect social signatures to graph properties:

1. **Node Degree.** For node v, the length of its social signature $l(v)$ is equal to its degree d_v. For example, in Figure 2.4, the social signature of node A, $S(A) = \{2, 3, 3\}$, has three elements. Hence, $l(A) = 3 = d_A$.

2. **Degree Distribution.** For a given node v, with degree d, by definition, its degree is observed in social signatures of d other nodes. Let n_k denote the total number of times degree k is observed in a graph. Let s_k denote the total number of times degree k is observed in all social signatures. Hence, given social signatures for all nodes, n_k can be calculated as

$$n_k = \frac{s_k}{k}. \tag{2.18}$$

   For example, in Example 2.4.1, the number of times 3 is observed in social signatures is $s_3 = 9$ times. Hence, there are $n_3 = s_3/3 = 9/3 = 3$ nodes of degree 3 in the graph in Figure 2.4.

   Given n_k, we can easily recover the degree distribution $p(d_x = k)$ of a graph with n nodes using social signatures,

$$p(d_x = k) = \frac{n_k}{n} = \frac{s_k}{nk}. \tag{2.19}$$

3. **Ego Degree Distribution.** The social signature of a node is a subset of the degree sequence of the graph. Note that the social signature is not the degree sequence of the subgraph induced by the node and its neighbors, also known as the *ego network*. This is because degrees in the social signature are the degrees in the whole graph. Each degree in the social signature of a node can be *larger* than the corresponding degree in the degree sequence of the ego network. This

fact can be used to approximate the degree distribution of the ego network, i.e., ego degree distribution.

The ego network of node v, has $l(v) + 1$ nodes ($l(v)$ friends + ego). let $n_k^{\geq}(v)$ denote the number of degrees that are greater or equal to k in the social signature of v. Then the degree distribution for the ego network of v can be approximated as follows,

$$p(d_x = k) \leq \begin{cases} \frac{n_k^{\geq}(v)}{l(v)+1} & k \neq l(v); \\ \frac{n_k^{\geq}(v)+1}{l(v)+1} & k = l(v). \end{cases} \tag{2.20}$$

For example, in Figure 2.4, social signature for A is $S(A) = \{2, 3, 3\}$ and $l(A) = 3$. Here, we have $n_3^{\geq}(A) = 2$. So, in the A's ego network,

$$p(d_v = 3) \leq \frac{2+1}{3+1} = 0.75. \tag{2.21}$$

Similarly, in D's ego network, $p(d_x = 3) \leq 2/4$ and in F's ego network, $p(d_x = 1) \leq 1$.[5]

4. **Node Connectivity.** Consider any two vertices v_1 and v_2, with degrees d_1 and d_2, respectively. If there is an edge between v_1 and v_2, then $d_1 \in S(v_2)$ and $d_2 \in S(v_1)$. This property can be used for graph reconstruction when only social signatures are available, but edges are unavailable. For example, in Figure 1, node A has social signature $S(A) = \{2, 3, 3\}$ and degree $d_A = 3$ and node F has social signature $S(F) = \{3\}$ and degree $d_F = 1$. Nodes A and F cannot be connected because $d_A = 3 \in S(F)$, but $d_F = 1 \notin S(A)$.

5. **Social Signatures vs. Adjacency Lists.** Social signatures are a relaxed version of adjacency lists. Consider the space required to store adjacency lists

[5]In case of nodes with degree 1, upper-bound is tight and inequality becomes equality. Hence, $p(d_v = 1) = 1$.

and social signatures. In a network with n vertices and m edges, storing the adjacency list requires storing n indices for nodes and $\sum_i d_i = 2m$ indices for the connections in the adjacency list, i.e., a total of $n + 2m$ values. For social signatures, we require $\sum_i d_i$ values to be stored because the social signature for node i contains d_i values. Hence, a total of $2m$ values are required for storage. Therefore, social signatures are more storage friendly that adjacency lists. Note that adjacency lists are more accurate and can guarantee perfect network reconstruction. However, in adjacency lists, for each nodes we carry more information about the surrounding network of a node. In social signatures, while we carry minimum information about the surrounding network, the reconstruction accuracy is still high as we will show in Section 2.4.5.

6. **Common Neighbors.** For two nodes v_1 and v_2, if they are both connected to a node x (a common neighbor), then $d_x \in S(v_1)$ and $d_x \in S(v_2)$. Let $S(v_1) \cap S(v_2)$ denote the multiset intersection between the social signatures of nodes v_1 and v_2. Let $N(v_1, v_2)$ denote the number of common neighbors v_1 and v_2 have. Then,

$$N(v_1, v_2) \leq |S(v_1) \cap S(v_2)|. \tag{2.22}$$

For example, in Figure 2.4, nodes B and D have two common neighbors A and E, i.e., $N(B, D) = 2$. The intersection between the social signatures of nodes B and D is $\{2, 3\}$. Therefore, the property holds as $N(B, D) = 2 \leq 2 = |\{2, 3\}|$.

This property has application for link prediction using social signatures as the number of common neighbors plays an important role for predicting potential links in social networks [80].

7. **Network Density.** If all the social signatures are known, all the degrees are known. The summation of all degrees is known to be twice as the number of

edges; therefore, the number of edges can be determined. For a network with n vertices and their social signatures, the network density ρ can be computed as

$$\rho = \frac{\sum_v l(v)}{n(n-1)}. \tag{2.23}$$

8. **Ego Network Density.** As mentioned, the ego network for node v has $l(v)+1$ nodes. It is easy to show that ego network density for node v, $\rho(v)$, can be approximated as

$$\rho(v) \leq \min(\frac{S_{\sum}(v) + l(v)}{l(v)(l(v)+1)}, 1), \tag{2.24}$$

where $S_{\sum}(v)$ (defined in Equation 2.17) is the summation of degrees in the social signature.

9. **Clustering Coefficient.** (local) clustering coefficient measures how close neighbors of a node are to being a clique. We can approximate the clustering coefficient $c(v)$ of node v as

$$c(v) \leq \min(\frac{S_{\sum}(v)}{l(v)(l(v)-1)}, 1). \tag{2.25}$$

We have demonstrated some basic properties of social signatures. Next, we will investigate the uniqueness of social signatures in large graphs.

2.4.4 Social Signature Uniqueness

In this subsection, we investigate the possibility of uniqueness in social signatures. To that end, we investigate the uniqueness of social signatures in different large-scale synthetic and real-world networks with power-law degree distributions.

In the first part, we investigate uniqueness in large synthetic graphs. There are many models that generate synthetic graphs with power-law degree distribution including the small-world model [124], the vertex copying model [71], the preferential

attachment model [17], random graphs with power-law degree distribution [10], among many others. Here, we focus on the popular preferential-attachment model [17] and leave the theoretical analysis of other well-established models as part of our future work.

In the second part, we investigate real-world graphs with power-law degree distribution. These networks are similar to social networks observed online for which the underlying process generating these networks is generally unknown. The results in the second section are general and apply to any real-world graph with power-law degree distribution.

Uniqueness in Synthetic Networks

Here, we investigate the uniqueness of social signatures in graphs generated by the preferential attachment model. The following theorem provides the condition under which social signatures become unique in synthetic graphs generated by the preferential attachment model. In the theorem, let $k = l(v)$ be the length of the social signature of a node v, i.e., its degree.

Theorem 3. *For a power-law graph with n nodes that is generated by the preferential attachment model, the social signature of a node is unique when its degree $k \gtrsim \ln n$.*

Proof. Let $p(d_1, d_2, \ldots, d_k|k)$ denote the probability of observing a social signature of $\{d_1, d_2, \ldots, d_k\}$ for a node that has degree k. Based on the preferential attachment model, the node that arrives in the network selects other nodes solely based on their degrees, so we can assume

$$p(d_1, d_2, \ldots, d_k|k) \approx p(d_1|k)p(d_2|k)\ldots p(d_k|k). \tag{2.26}$$

Later, we will show that this is not a strong assumption and fits the real-world

data well. Using Bayes theorem, we can rewrite Equation 2.26 as

$$p(d_1, d_2, \ldots, d_k|k) \approx \frac{p(k, d_1)}{p(k)} \frac{p(k, d_2)}{p(k)} \ldots \frac{p(k, d_k)}{p(k)}. \tag{2.27}$$

To compute the RHS of Equation 2.27, one needs to compute the joint distribution $p(k, l)$. We can compute the joint distribution by following the process provided by Albert and Barabasi [11]. Here, we summarize their solution in the following up to Equation 2.29 for clarifying the rest of our proof (more details for calculating the joint distribution can be found in the work of Albert and Barabasi [11]).

To compute the joint distribution, we consider the number nodes with degree k and l that are connected. Let $N_{k,l}$ denote the number of such pairs. In the preferential attachment process, younger nodes that are added later to the network have smaller degrees. So, without loss of generality, we can assume $k < l$ and that k is added later to the network than l. We can also assume for mathematical convenience that the number of nodes selected during the preferential attachment process is $m = 1$. In other words, once a node enters a network, it selects only one other node to connect to. Then, we can compute the change that $N_{k,l}$ makes in time:

$$\begin{aligned} \frac{dN_{k,l}}{dt} &= \frac{(k-1)N_{k-1,l} - kN_{k,l}}{\sum_k kN_k} \\ &+ \frac{(l-1)N_{k,l-1} - lN_{k,l}}{\sum_k kN_k} \\ &+ (l-1)N_{l-1}\delta_{k,1}, \end{aligned} \tag{2.28}$$

where δ represents the Kronecker delta, and N_k and N_{l-1} are the number of nodes with degree k and $l-1$ at time t, respectively. The first term in Equation 2.28 computes the change that $N_{k,l}$ will have if we add one edge to a node that has a degree k or $k-1$ and is connected to a node of degree l. The first term in the numerator represent the gain that $N_{k,l}$ will have and the second term represents the loss. Similarly, the second term in Equation 2.28, considers edges added to nodes

that have degrees l or $l-1$ and are connected to nodes with degree k. The last term in the equation considers the case where $k=1$ and the edge connects a **new** node to another node of degree $l-1$.

In the preferential attachment model, we know that $\sum_k kN_k = 2t$ and $N(k,l) = tp(k,l)$; therefore, Equation 2.28 can be made time independent. Solving which for $p(k,l)$ results in

$$\begin{aligned} p(k,l) &= \frac{4(l-1)}{k(k+1)(k+l)(k+l+1)(k+l+2)} \\ &+ \frac{12(l-1)}{k(k+l-1)(k+l)(k+l+1)(k+l+2)}. \end{aligned} \tag{2.29}$$

As, $k < l$, to maximize $p(k,l)$, we can set $l = k+1$ in Equation 2.29. Therefore,

$$\begin{aligned} p(k,l) &\leq \frac{4k}{k(k+1)(2k+1)(2k+2)(2k+3)} \\ &+ \frac{12k}{k(2k)(2k+1)(2k+2)(2k+3)} \\ &= \frac{4k}{k(2k+1)(2k+2)(2k+3)}(\frac{1}{k+1} + \frac{3}{2k}) \\ &\leq \frac{4k}{k(2k+1)(2k+2)(2k+3)}(\frac{1}{2} + \frac{3}{2}) \\ &= \frac{8k}{k(2k+1)(2k+2)(2k+3)} \\ &= \frac{8}{8k^3+24k^2+22k+6}, \end{aligned} \tag{2.30}$$

where last inequality uses the fact that $\frac{1}{k+1} + \frac{3}{2k}$ is maximized when $k=1$. Substituting Equation 2.30 in Equation 2.27, we get

$$p(d_1, d_2, \ldots, d_k|k) \leq (\frac{8}{(8k^3+24k^2+22k+6)p(k)})^k. \tag{2.31}$$

It is known that in the preferential attachment model, the probability of observing k, $p(k)$, is approximately k^{-3} [11]; therefore,

$$
\begin{aligned}
p(d_1, d_2, \ldots, d_k|k) &\leq (\frac{8k^3}{8k^3 + 24k^2 + 22k + 6})^k \\
&= (\frac{1}{1 + \frac{3}{k} + \frac{22}{8k^2} + \frac{6}{8k^3}})^k \\
&\leq (\frac{1}{1 + \frac{1}{k}})^k \\
&\approx (\frac{1}{e})^k. \qquad (2.32)
\end{aligned}
$$

For a social signature to be unique, we must have

$$p(d_1, d_2, \ldots, d_k|k)np(k) \leq 1. \qquad (2.33)$$

Since, $0 \leq p(k) \leq 1$, uniqueness condition is met when

$$p(d_1, d_2, \ldots, d_k|k) \leq \frac{1}{n}. \qquad (2.34)$$

Equation 2.32 provides an upper-bound for the likelihood of observing a social signature; therefore, a social signature is unique when

$$p(d_1, d_2, \ldots, d_k|k) \lesssim (\frac{1}{e})^k \leq \frac{1}{n}, \qquad (2.35)$$

or equivalently, when $k \gtrsim \ln n$, which completes the proof. Note that as k needs to be an integer, it is more realistic to consider $k \gtrsim \lceil \ln n \rceil$. □

Our later experiments show that the lower bound is in fact tight and $k \approx \lceil \ln n \rceil$. The theorem shows that the uniqueness of social signature grows logarithmically with the size of n. This is surprising, as in a network of $n = 10^{100}$ nodes, you only need $k \approx 230$ to be unique. In other words, having 230 or more friends and constructing the social signature from those friends, can represent the user uniquely in the network. In fact for Facebook, the largest current social network, $n \leq 10^{10}$, having 23 or more friends is enough for uniquely representing users. Figure 2.5 demonstrates the social

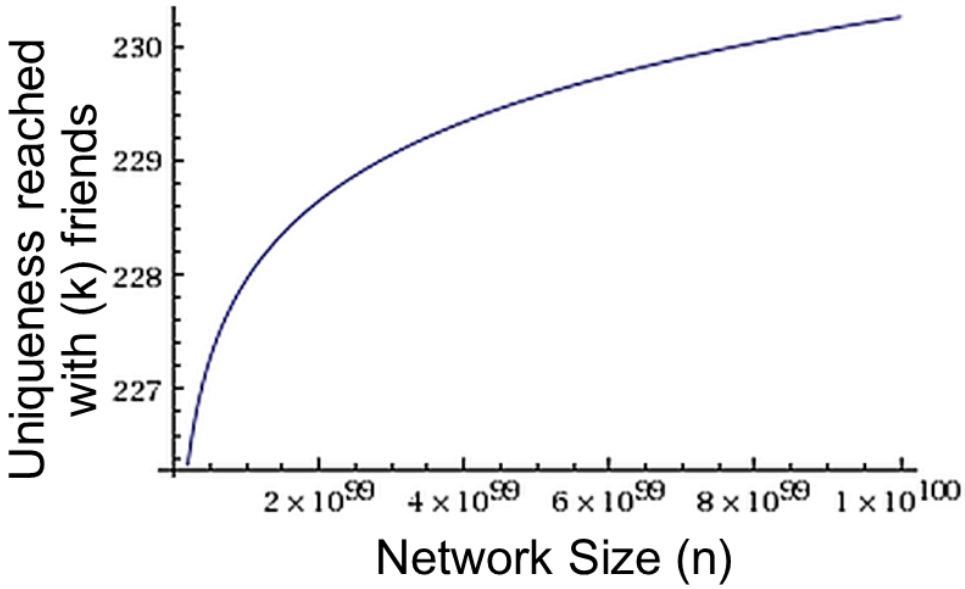


Figure 2.5: The Social Signature Length (degree) k that Guarantees Uniqueness for Graphs with Different Sizes ($10 \leq n \leq 10^{100}$). All graphs are generated by the preferential attachment model and the k values are caluclated according to Theorem 3: $k = \ln n$.

signature length k that will be unique for graphs with different sizes ($10 \leq n \leq 10^{100}$) generated by the preferential attachment process.

We also evaluate Theorem 3 empirically by generating many graphs using the preferential attachment model. We generate 5,000 graphs that range from $n = 100$ nodes to $n = 50,000$ nodes, with increments of 100 nodes. These graphs are generated using the CONTENT toolbox [116]. For each graph size (n), we generate 10 graphs. For each graph, we compute the k at which social signatures become unique in the graph. Finally, we take the average among the 10 graphs of the same size. We also compute the expected theoretical lower-bound for social signature uniqueness from theorem 3: $k = \lceil \ln n \rceil$. The results are provided in Figure 2.6. In the Figure, the solid line represents simulation results and the dashed line is the theoretical lower-bound. We notice two observations in this figure. First, the general trend of the Figure 2.6 matches the trend observed in Figure 2.5. Second, the theoretical bound generated by $k = \lceil \ln n \rceil$ matches closely to the simulation results.

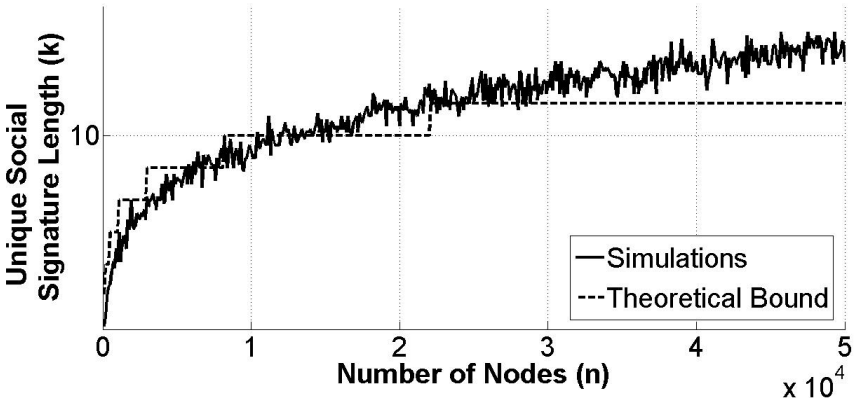


Figure 2.6: Simulating 500 preferential attachment graphs with $100 \leq n \leq 50,000$ and increments of 100 nodes. The solid line provides the uniqueness limit for social signatures and the dashed line is computed using Theorem 3: $k = \ln n$.

Next, we will prove general results that apply to *any* graph with a power-law degree distribution.

Uniqueness in Real-World Networks

Here, we investigate the uniqueness of social signatures in graphs with power-law degree distribution. In particular, we will prove the following theorem:

Theorem 4. *For a power-law graph with n nodes such that $p(k) = ck^{-\alpha}$, the social signature of a node is unique when its degree $k \approx e^{W(\ln n)}$, where W is the Lambert function (product logarithm).*

Proof. The proof follows a similar argument to the proof of Theorem 3.

$$p(d_1, d_2, \ldots, d_k|k) \approx p(d_1|k)p(d_2|k)\ldots p(d_k|k). \tag{2.36}$$

The conditional probability $p(d|k)$ can be upper-bounded,

$$p(d|k) \leq p(k) = ck^{-\alpha} \leq k^{-\alpha}. \tag{2.37}$$

Substituting Equation 2.37 in Equation 2.36, we get

$$p(d_1, d_2, \ldots, d_k|k) \leq (\frac{1}{k^\alpha})^k \leq (\frac{1}{k})^k, \tag{2.38}$$

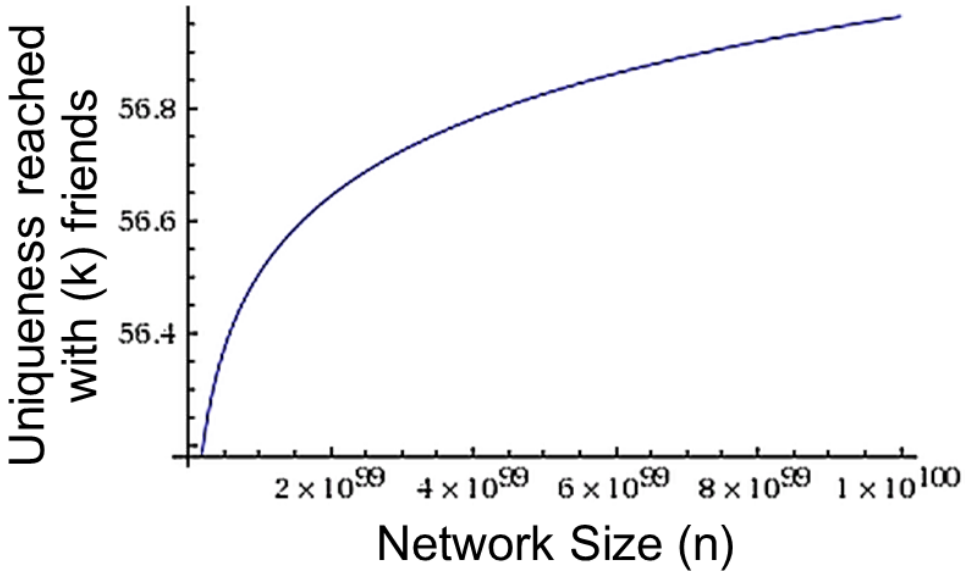


Figure 2.7: The Uniqueness Value $k = e^{W(\ln n)}$ for Graphs with Different Sizes $(10 \leq n \leq 10^{100})$

where last inequality is a result of $\alpha > 1$. Similarly, for uniqueness, we require

$$(\frac{1}{k})^k = \frac{1}{n}, \tag{2.39}$$

which when solved for k results in

$$k = e^{W(\ln n)}, \tag{2.40}$$

where W is the Lambert function (product logarithm). This concludes the proof.

□

The Lambert function can only be numerically approximated; however, once simulated (Figure 2.7), a curve similar to the uniqueness curve simulated for the preferential attachment model (Figure 2.5) is observed. In fact, for graph size n, the predicted uniqueness value by $e^{W(\ln n)}$ is always smaller than $\ln n$, predicted for the graphs generated by the preferential-attachment model. For instance, for $n = 10^{100}$, $e^{W(\ln n)} \approx 56 < 230 = \ln n$. This shows that in a power-law network of 10^{100} users, users that have only having 56 or more friends, have unique social signatures. Similarly, for Facebook-size networks, $n \leq 10^{10}$, only 10 friends is enough to have a unique

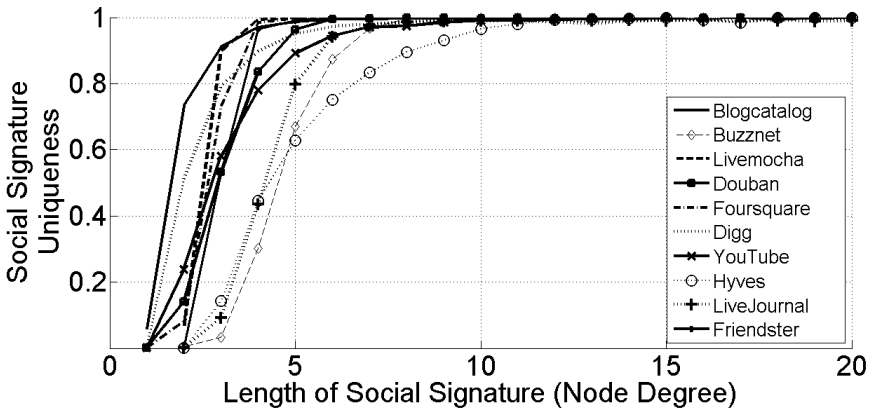


Figure 2.8: The Uniqueness of Social Signatures for Real-World Graphs

social signature. Figure 2.7 depicts the value $k = e^{W(\ln n)}$ for graphs with different sizes $10 \leq n \leq 10^{100}$.

Our empirical results confirm the results of Theorem 4. As most social networks have less than $n = 10^{10}$ users, we expect social signatures to become unique when users have $e^{W(\ln 10^{10})} = 10$ or more friends. By manually measuring the uniqueness of social signatures, we notice that social signatures are almost always unique for users in these real-world networks that have 10 or more friends. Figure 2.8 shows the uniqueness of social signature for large-scale real-world networks listed in Table 2.5.

While our results show uniqueness for different n, they do not show how the probability of observing a social signature approaches uniqueness. Our next theorem measures that.

Theorem 5. *For a social signature $\{d_1, d_2, \ldots, d_k\}$ with length k and when $n > k^k$, the probability of the signature being unique is $(\frac{n}{k^k})e^{1-\Omega(\frac{n}{k^k})}$.*

Proof. Let $X_1, X_2, \ldots, X_{np(k)}$ be independent poison trials, each for one of the users with degree k. Assume that $P(X_i = 1) = p(d_1, d_2, \ldots, d_k|k)$. Let $X = \sum_i X_i$ denote the number of users having social signature $\{d_1, d_2, \ldots, d_k\}$ and μ be $E[X]$. Then,

using Chernoff bound (lower tails), for any $\delta \in (0, 1]$, we have

$$P(X < (1-\delta)\mu) < (\frac{e^{-\delta}}{(1-\delta)^{(1-\delta)}})^{\mu}. \tag{2.41}$$

Setting $(1-\delta)\mu = 1$, we get

$$\delta = \frac{\mu - 1}{\mu}. \tag{2.42}$$

Substituting it in Equation 2.41, we get

$$P(X < 1) < \mu e^{1-\mu}. \tag{2.43}$$

As $\delta \in (0, 1]$, from Equation 2.42, we have $\mu > 1$. From theorem 4, we know that $p(d_1, d_2, \ldots, d_k|k)$ is upper-bounded by $(\frac{1}{k})^k$; therefore,

$$\mu = np(k)p(d_1, d_2, \ldots, d_k|k) \leq np(k)(\frac{1}{k})^k \leq \frac{n}{k^k} \in \Omega(\frac{n}{k^k}). \tag{2.44}$$

Replacing this term in Equation 2.43, we get

$$P(X < 1) < (\frac{n}{k^k})e^{1-\Omega(\frac{n}{k^k})}, \tag{2.45}$$

which completes the proof. □

Similarly, we can bound the probability of being non-unique after the social signatures are supposed to be unique:

Theorem 6. *For a social signature $\{d_1, d_2, \ldots, d_k\}$ with length k and when $n < k^k$, the probability of the signature being non-unique is less than $\frac{n}{k^k}e^{1-\Omega(\frac{n}{k^k})}$.*

Proof. We skip the details as the proof is similar to that of Theorem 5. The only difference is that upper tail Chernoff bound is used. □

Note that as $n < k^k$, the bound provided in Theorem 6 can be made simpler, yet weaker,

$$P(X > 1) < e(\frac{n}{k^k}). \tag{2.46}$$

We have proved the uniqueness conditions for social signatures in synthetic and real-world networks. We have also shown how social signatures approach uniqueness and presented how non-unique they can be after they are supposed to be unique. Both bounds in Theorems 5 and 6 show that social signatures approach uniqueness exponentially and their non-uniqueness drops exponentially after they are supposed to be unique. Next, we will demonstrate how these results can be used in different applications.

2.4.5 Applications of Social Signatures

We have shown uniqueness properties for social signatures and when social signatures become unique. In this section, we introduce two applications for social signatures. First is graph reconstruction. In the second application, we come back to identifying users across social media sites.

Graph Reconstruction

Consider a graph where edge information is unavailable, but for all vertices, the social signatures are available. This is a typical example in virus propagation networks, where local neighbors are known, but the general graph structure is unknown. Another example is the power grid, where scanning at the consumer level is possible, yet network topology is often protected due to security concerns. Can we reconstruct the network with social signatures?

For graph reconstruction, we can use property 4 in Section 2.4.3. We restate that property as a corollary here for clarity:

Corollary 7. *Consider two vertices v_1 and v_2 with degrees v_1 and v_2, respectively. If there is an edge between v_1 and v_2, then $v_1 \in S(v_2)$ and $v_2 \in S(v_1)$.*

Corollary 7 shows that if $v_1 \notin S(v_2)$ or $v_2 \notin S(v_1)$, nodes v_1 and v_2 cannot be connected. As power-law graphs are sparse, this can help predict many edges that do not exist in such graphs. When $v_1 \in S(v_2)$ and $v_2 \in S(v_1)$, there is a chance that v_1 and v_2 are not connected.

Consider two vertices v_1 and v_2 with degrees d_1 and d_2, respectively. Assume that $d_1 \in S(v_2)$ and $d_2 \in S(v_1)$. Assume the two social signatures share a set of *unique* degrees. Denote this set as $U = \{u_1, u_2, \ldots, u_k\}$, where $u_i \in S(v_1)$, $u_i \in S(v_2)$, for $1 \leq i \leq k$. Let $C_1 = (c_1^1, c_2^1, \ldots, c_k^1)$ denote the number of times each member of U is repeated in S_1. Similarly, let $C_2 = (c_1^2, c_2^2, \ldots, c_k^2)$ denote the number of times members of U are repeated in S_2. Furthermore, let n_k denote the number of times a node of degree k is observed in the graph. This can be computed using $np(k)$ from the degree distribution. Alternatively, one can compute n_k from social signatures using Equation 2.18 (Section 2.4.3, property 2). Let $N = (n_{u_1}, n_{u_2}, \ldots, n_{u_k})$ denote the number of times unique degrees shared are observed in the whole graph. Then the probability of v_1 being connected to v_2 can be approximated using the next theorem.

Theorem 8. *The probability P of two vertices v_1 and v_2 being connected is bounded by*

$$\max_i p_i \leq P \leq \min(1, \sum_{i=1}^{k} p_i), \tag{2.47}$$

where

$$p_i = \begin{cases} 1 - \frac{\binom{n_{u_1} - c_1^1}{c_1^2}}{\binom{n_{u_1}}{c_1^2}} & c_1^1 + c_1^2 \leq n_{u_1}; \\ 1 & \textit{Otherwise} \end{cases} \tag{2.48}$$

Proof. The proof follows a simple combinatorial construction. Consider a degree that is shared u_i, with counts c_i^1 and c_i^2 in the first and second social signature, respectively.

The probability of both vertices v_1 and v_2 selecting different u_i is

$$\frac{\binom{n_{u_1}}{c_1^1}\binom{n_{u_1}-c_1^1}{c_1^2}}{\binom{n_{u_1}}{c_1^1}\binom{n_{u_1}}{c_1^2}} = \frac{\binom{n_{u_1}-c_1^1}{c_1^2}}{\binom{n_{u_1}}{c_1^2}}. \tag{2.49}$$

Therefore, the probability of being connected is:

$$1 - \frac{\binom{n_{u_1}-c_1^1}{c_1^2}}{\binom{n_{u_1}}{c_1^2}}. \tag{2.50}$$

Clearly, this only holds when there is a chance of being connected to different nodes $c_1^1 + c_1^2 \leq n_{u_1}$; otherwise, based on pigeon-hole principle, the two nodes will be connected.

Two nodes are connected if one of the p_i's is 1. In other words, only if one of the shared degrees represent the same node in the network, two nodes are connected. So, p_i's represent the probabilities of disjunct events. Hence, from Boole-Frechet inequalities, the probability P of two nodes v_1 and v_2 being connected is bounded by

$$\max_i p_i \leq P \leq \min(1, \sum_{i=1}^{k} p_i), \tag{2.51}$$

which completes the proof. □

Theorem 8 provides bounds on the probability that nodes v_1 and v_2 **are** connected and corollary 7 provides conditions under which they **are not connected**. The next step is to utilize these results and recover the adjacency matrix.

Let $W \in \mathcal{R}^{n \times n}$ denote the symmetric matrix containing all the zeros computed using corollary 7 and probabilities (lower-bound, upper-bound, or a convex combination) computed using Theorem 8. Then, we can recover the binary adjacency matrix A using the following optimization:

$$\max_{A \in \mathcal{B}} \quad \sum_{ij} A_{ij} W_{ij} \tag{2.52}$$

$$s.t. \quad \sum_{j} A_{ij} = d_i \tag{2.53}$$

$$A_{ii} = 0, \tag{2.54}$$

$$A_{ij} = A_{ji}, \forall i, j \in 1, \ldots, n. \tag{2.55}$$

This is known as the generalized matching problem (also known as b-matching). The optimization problem can be solved using balanced network flow [66] or more efficiently using loopy belief propagation [63]. Depending on the execution algorithm the running time is expected to range from $O(bn^3)$ to $O(\min(|E| \log |V|, |V|^2)|V|b)$, where b is the maximum degree in the graph.

To demonstrate the feasibility of graph reconstruction using social signatures, we measure graph reconstruction accuracy for some well-known graphs. We compute the zeros in the adjacency matrix using Corollary 7 and use the lower-bound for connection probability in Theorem 8 as the connection probability. We test two well-known graphs: the Zachary's karate club dataset [135] and the dolphin social network [85]. For both networks, we reconstruct the graph using the social signatures alone. We then estimate the error by computing the hamming distance between the original adjacency matrix and the reconstructed one. For the Karate Club dataset, we recover the graph with 98.3% accuracy and for the Dolphin Social Network, the accuracy is 97.6%. The reconstruction accuracy demonstrates the feasibility and accuracy of graph reconstruction using social signatures. Next, we study a different application where we investigate the possibility of identifying users across sites with social signatures.

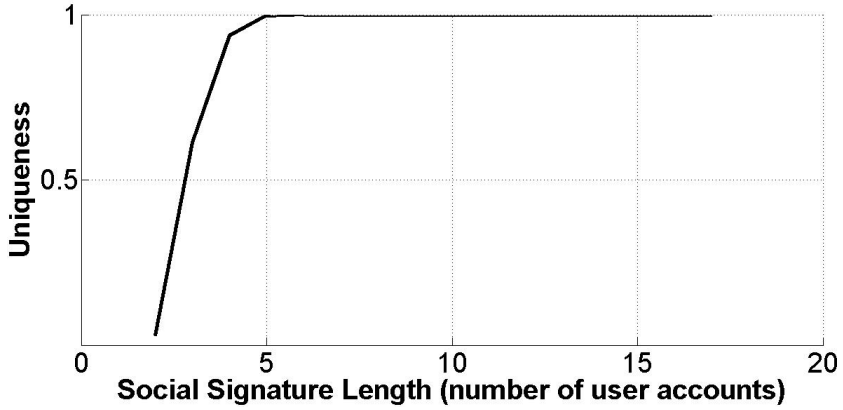


Figure 2.9: User Uniqueness across Networks with Social Signatures

Identifying Users across Sites

Consider the space of real people in the world. Each person has account on different social media sites and for each site, the user has a number of friends. We can consider a hyper-graph in the space of real people connecting each user to the accounts that user owns on different sites. This way the social signature of the user is the number of friends the user has on different sites. As we discussed, social signatures are unique for power-law graphs. This means as long as the numbers of accounts that the users have follow a power-law distribution, we can uniquely identify users across sites. We show in Chapter 5 that the number of accounts users have across sites follows a power-law distribution; therefore, their social signatures have to be unique. We verify this by collecting a set of 96,194 users on around 20 websites and collect their friends on these sites. Based on Theorem 4, we require social signatures to be of length $e^{W(\ln 96{,}194)} \approx 6.25$ to be unique. We notice the same pattern when empirically measuring uniqueness of social signatures in this dataset. Figure 2.9 demonstrates the uniqueness of social signatures across sites. As the figure shows, social signatures become unique as expected when users have joined 6 or more sites.

While this result is promising, our experiment was limited to around 100k users. In reality, we have billions of users on social media. Even for one billion, as we have

shown before, we need around $e^{W(\ln 10^9)} \approx 9.29$ accounts to be able to uniquely identify users. In chapter 5, we show that the number of accounts that users have is most of the time less than 5; therefore, while possible to some degree, it is challenging to uniquely identify users across sites with social signatures alone. In fact, in Chapter 5, we show that the performance of such methods is around 40% when using social signatures alone, combined with machine learning techniques.

2.5 Related Work

To the best of our knowledge, the work presented in this chapter is unique. However, there are studies similar to the work presented here. Perhaps the most similar study is the seminal works of Hay et al. [61, 62]. Hay and Colleagues investigate degree signatures in different graphs and show the power of these degree signatures for re-identification of masked nodes in graphs. In fact, the definition of $\mathcal{H}_2$ in their papers, matches exactly with the social signature definition in this chapter. However, their study considers properties of these degree signatures that are different from the study presented here. For instance, the authors consider how growing these degree signatures to more than 1 hop (as in social signatures) can help better uniquely identify nodes. They also theoretically investigate uniqueness properties of these degree signatures in graphs with distribution other that those discussed here (random graphs, random graphs with power-law distribution, etc.). The results provided in this study are complementary to those provided by Hay et al. In our study we not only analyze social signatures for general power-law graphs, but also discuss when they become unique and provide concentration results before and after the *phase transition* of becoming unique. In addition, while Hay and colleagues also provide some similar graph reconstruction results, the results assume that some adversary has access to the topology of the network (see Michael Hay's PhD thesis [60]). Here,

there are no assumption on the knowledge of graph topology and the combinatorial approach to graph reconstruction works for any graph with power-law degree distribution. The name social signature has been previously used in other settings. In particular, the name has been used in human communication networks for identifying individuals [109]; however, the study is different and is dedicated to how human communication is divided between friends and how these communication patterns are consistent over time.

2.6 Summary

In this chapter, we have investigated the possibility of utilizing minimum link information for identifying users across sites. We first studied the possibility of utilizing heuristic-based methods for identifying users across sites. We showed that these methods are not efficient in social media sites, particularly due to the way users are embedded in networks across sites. Our results show that counter-intuitively, link information is not sufficient for identifying individuals across networks when using mapping information or graph structure. Next, we further investigated the minimum information in networks. We started with degrees and showed that degrees are nonunique in large power-law graphs. By adding information, we introduced social signatures. We proved social signatures are unique for graphs generated by preferential attachment model and general power-law graphs. Finally, we introduced two applications for social signatures: graph reconstruction and identifying users across sites. As uniqueness of social signatures requires users to be members of more sites, the performance is not highly accurate. In addition to the challenges discussed in this chapter, link information might not be always available across sites for a general solution to the problem of user identification across sites. Therefore, we consider using content information to identify individuals in the next chapter.

Chapter 3

UTILIZING MINIMUM CONTENT INFORMATION

Perhaps the less we have, the more we are required to brag.

John Steinbeck

In Chapter 2, we investigated the possibility of user identification with link information. This chapter, investigates the same possibility with content information. To use content information to identify users across social networks, we introduce a methodology (MOBIUS) [139] for finding the mapping among identities across social media sites. Our methodology is based on behavioral patterns that users exhibit in social media, and has roots in behavioral theories in sociology and psychology. Unique behaviors due to environment, personality, or even human limitations can create redundant information across social media sites. Our methodology exploits such redundancies for identifying users across social media sites. We use the minimum amount of content information available across sites and discuss how additional information can be added.

3.1 Content-based User Identification

Let us begin by formulating our problem in terms of content information. Information shared by users on social media sites provides a *social fingerprint* of them and can help identify users across different sites. We start with the *minimum* amount of

The content in this chapter has been published at ICWSM 2009 [136], KDD 2013 [139], and in the TKDD journal [145].

information that is available on *all* sites. Later on, in Section 3.4, we will discuss how one can add extra information to this minimum as it becomes available across sites. In terms of information availability, *usernames* seem to be the minimum common factor available on all social media sites. Usernames are often alphanumeric strings or email addresses, without which users are incapable of joining sites. Usernames are unique on each site and can help identify individuals, whereas most personal information, even "first name + last name" combination, are non-unique. We formalize our problem using usernames as the atomic entities available across all sites. Other profile attributes, such as gender, location, interests, profile pictures, language, etc., when added to usernames, should help better identify individuals; however, the lack of consistency in the available information across all social media, directs us toward formulating with usernames. When considering usernames, two general problems need to be solved for user identification:

I. Given two usernames u_1 and u_2, can we determine if they belong to the same individual?

II. Given a single username u from individual $\mathcal{I}$, can we find other usernames of $\mathcal{I}$?

Question **II** can be answered via a two-stage process: 1) we find the set of all usernames C that are likely to belong to individual $\mathcal{I}$. We denote set C as *candidate usernames* and, 2) for all candidate usernames $c \in C$, we check if c and u belong to the same individual. Therefore, if candidate usernames C are known, question **II** reduces to question **I**. Now, where can we find these candidate usernames?

We will discuss this later in our discussion section (Section 3.4) and from now on, we focus on question **I**. One can answer question **I** by learning an *identification*

function $f(u, c)$,

$$f(u,c) = \begin{cases} 1 & \text{If } c \text{ and } u \text{ belong to same } \mathcal{I} \text{ ;} \\ 0 & \text{Otherwise.} \end{cases} \tag{3.1}$$

Without loss of generality, we can assume that username u is known to be owned by some individual $\mathcal{I}$ and c is the candidate username whose ownership by $\mathcal{I}$ we would like to verify. In other words, u is the prior information (history) provided for $\mathcal{I}$. Our function can be generalized by assuming that our prior is a *set* [1] of usernames $U = \{u_1, u_2, \ldots, u_n\}$ (hereafter referred to as "prior usernames"). Informally, the usernames of an individual on some sites are given and we have a candidate username on another site whose ownership we need to verify; e.g., usernames u_t and u_f of someone are given on Twitter and Facebook, respectively; can we verify if c is her username on Flickr?

Definition. ***Content-Based User Identification.*** *Given a set of n usernames (prior usernames)* $U = \{u_1, u_2, \ldots, u_n\}$*, owned by individual* $\mathcal{I}$ *and a candidate username c, a user identification procedure attempts to learn an identification function* $f(.)$ *such that*

$$f(U,c) = \begin{cases} 1 & \textit{If } c \textit{ and set } U \textit{ belong to } \mathcal{I} \text{ ;} \\ 0 & \textit{Otherwise.} \end{cases} \tag{3.2}$$

Our methodology for ***MO****deling* ***B****ehavior for* ***I****dentifying* ***U****sers across* ***S****ites* (**MOBIUS**) [2] is outlined in Figure 3.1. When individuals select usernames, they ex-

[1] Mathematically, a set can only contain distinct values; however, here a user may use the same username on more than one site. In our definition of username set, it is implied that usernames are distinct when used on different sites, even though they can consist of the same character sequence.

[2] The resemblance to the Möbius strip comes from its *single-boundary* (representing a single individual) and its *connectedness* (representing connected identities of the individual across social media).

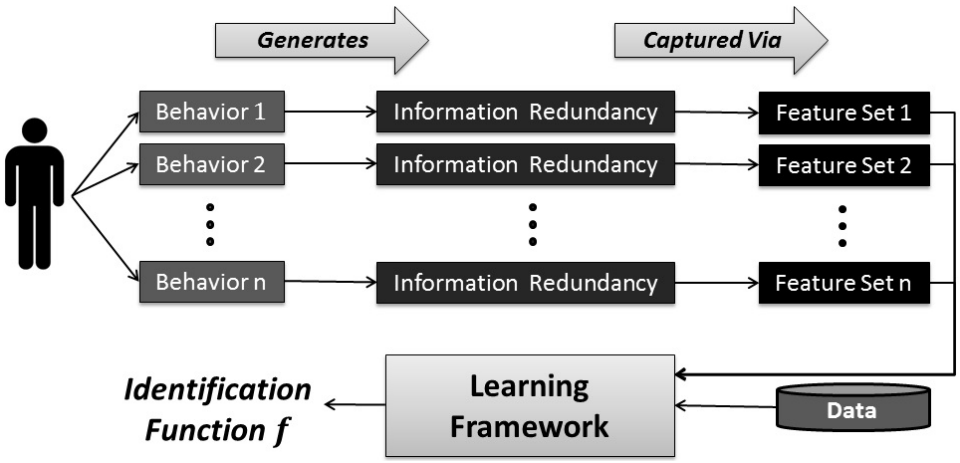


Figure 3.1: MOBIUS: Modeling Behavior for Identifying Users across Sites

hibit certain behavioral patterns. This often leads to *information redundancy*, helping learn the identification function. In MOBIUS, these redundancies can be captured in terms of data features. Following the tradition in machine learning and data mining research, the identification function can be learned by employing a supervised learning framework that utilizes these features and prior information (*labeled data*), in our case, sets of usernames with known owners. Supervised learning in MOBIUS can be performed via either classification or regression. Depending on the learning framework, one can even learn the probability that an individual owns the candidate username, generalizing our binary f function to a probabilistic model ($f(U, c) = p$). This probability can help select the most likely individual who owns the candidate username. The learning component of MOBIUS is the most straightforward. Hence, we next elaborate how to analyze behavioral patterns related to user identification and how features can be constructed to capture information redundancies due to these patterns. To summarize, MOBIUS contains 1) *behavioral patterns*, 2) *features* constructed to capture information redundancies due to these patterns, and 3) a *learning* framework. Given the interdependent nature of behaviors and feature construction, we discuss them together next.

3.2 MOBIUS: Behavioral Patterns and Feature Construction

Individuals often exhibit consistent behavioral patterns while selecting their usernames. These patterns result in information redundancies that help identify individuals across social media sites.

Individuals can avoid such redundancies by selecting usernames on different sites in a way such that they are completely different from their other usernames. In other words, their usernames are so different that given one username, no information can be extracted regarding the others. Theoretically, to achieve these independent usernames, one needs to select a username with Maximum Entropy [34]. That is, a **long** username string, as long as the site allows, with characters from those that the system permits, with **no redundancy** - an entirely **random** string.

Unfortunately, all of these requirements are contrary to human abilities [129]. Humans have difficulty storing long sequences with short-term memory capacity of 7 ± 2 items [93]. Human memory also has limited capability in storing random content and often, selectively stores content that contains familiar items known as "chunks" [93]. Finally, human memory thrives on redundancy, and humans can remember material that can be encoded in multiple ways [105]. These limitations result in individuals selecting usernames that are generally *not long*, *not random*, and have *abundant redundancy*. These properties can be captured using specific features which in turn can help learn an identification function. In this study, we find a set of consistent behavioral patterns among individuals while selecting usernames. These behavioral patterns can be categorized as follows:

1. **Patterns due to Human Limitations**

2. **Exogenous Factors**

3. **Endogenous Factors**

The features designed to capture information generated by these patterns can be divided into three categories:

1. **(Candidate) Username Features**: these features are extracted directly from the candidate username c, e.g., its length,

2. **Prior-Usernames Features**: these features describe the set of prior usernames of an individual, e.g., the number of observed prior usernames, and

3. **Username↔Prior-Usernames Features**: these features describe the relation between the candidate username and prior usernames, e.g., their similarity.

We will discuss behaviors in each of the above mentioned categories, and features that can be designed to harness the information hidden in usernames as a result of the pattern's existence. Note that these features may or may not help in learning an identification function. As long as these features could be obtained for learning the identification function, they are added to our feature set. Later on in Section 3.3, we will analyze the effectiveness of all features, and if it is necessary to find as many features as possible.

3.2.1 Patterns due to Human Limitations

In general, as humans, we have 1) *limited time and memory* and 2) *limited knowledge*. Both create biases that can affect our username selection behavior.

1. **Limitations in Time and Memory**

   Selecting the Same Username. As studied recently [136], 59% of individuals prefer to use the same username(s) repeatedly, mostly for ease of remembering.

Therefore, when a candidate username c is among prior usernames U, that is a strong indication that it may be owned by the same individual who also owns the prior usernames. As a result, we consider the number of times candidate username c is repeated in prior usernames as a feature.

Username Length Likelihood. Similarly, users commonly have a limited set of potential usernames from which they select one, once asked to create a new username. These usernames have different lengths and, as a result, a *length distribution* $\mathcal{L}$. Let l_c be the candidate username length and l_u be the length for username $u \in U$ (prior usernames). We believe that for any new username, it is more likely to have,

$$\min_{u \in U} l_u \leq l_c \leq \max_{u \in U} l_u; \tag{3.3}$$

For example, if an individual is inclined to select usernames of length 8 or 9, it is unlikely for the individual to consider creating usernames with lengths longer or shorter than that. Therefore, we consider the candidate username's length l_c and the length distribution $\mathcal{L}$ for prior usernames as features. The length distribution can be compactly represented by a fixed number of features. We describe distribution $\mathcal{L}$, observed via discrete values $\{l_u\}_{u \in U}$ as a 5-tuple feature,

$$(\mathbb{E}[l_u], \sigma[l_u], med[l_u], \min_{u \in U} l_u, \max_{u \in U} l_u), \tag{3.4}$$

where $\mathbb{E}$ is the mean, σ is the standard deviation, and med is the median of the values $\{l_u\}_{u \in U}$, respectively. Note that this procedure for compressing distributions as a fixed number of features can be employed for discrete distributions $\mathcal{D}$, observed via discrete values $\{d_i\}_{i=1}^{n}$.

Unique Username Creation Likelihood. Users often prefer not to create new usernames. One might be interested in the effort users are willing to put

into creating new usernames. This can be approximated by the number of unique usernames ($uniq(U)$) among prior usernames U,

$$uniqueness = \frac{|uniq(U)|}{|U|}. \tag{3.5}$$

Uniqueness is a feature in our feature set. One can think of $1/uniqueness$ as an individual's *username capacity*, i.e., the average number of times an individual employs a username on different sites before deciding to create a new one.

2. **Knowledge Limitation**

Limited Vocabulary. Our vocabulary is limited in any language. It is highly likely for native speakers of a language to know more words in that language than individuals speaking it as a second language. We assume the individual's vocabulary size in a language is a feature for identifying them, and as a result, we consider the number of dictionary words that are substrings of the username as a feature. Similar to *username length* feature, the number of dictionary words in the candidate username is a scalar; however, when counting dictionary words in prior usernames, the outcome is a distribution of numbers. We employ the technique outlined in Eq. (3.4) for compressing distributions to represent this distribution as features.

Limited Alphabet. Unfortunately, it is a tedious task to consider dictionary words in all languages, and this feature can be used for a handful of languages. Fortunately, we observe that the alphabet letters used in the usernames are highly dependent on language. For instance, while letter x is common when a Chinese speaker selects a username, it is rarely used by an Arabic speaker, since

no Arabic word transliterated in English contains letter x [58]. So, we consider the number of alphabet letters used as a feature, both for the candidate username as well as prior usernames.

3.2.2 Exogenous Factors

Exogenous factors are behaviors observed due to cultural affects or the environment that the user is living in.

Typing Patterns. One can think of keyboards as a general constraint imposed by the environment. It has been shown [42] that the layout of the keyboard significantly impacts how random usernames are selected; e.g., `qwer1234` and `aoeusnth` are two well-known passwords commonly selected by QWERTY and DVORAK users, respectively. Most people use one of two well-known keyboards DVORAK and QWERTY (or slight variants such as QWERTZ or AZERTY) [125]. To capture keyboard-related regularities, we construct the following 15 features for each keyboard layout (a total of 30 for both),

1. (1 feature) The percentage of keys typed using the *same hand* used for the previous key. The higher this value the less users had to change hands for typing.
2. (1 feature) Percentage of keys typed using the *same finger* used for the previous key.
3. (8 features) The percentage of keys typed using each finger. Thumbs are not included.
4. (4 features) The percentage of keys pressed on rows: Top Row, Home Row, Bottom Row, and Number Row. Space bar is not included.
5. (1 feature) The approximate *distance* (in meters) traveled for typing a username.

Normal typing keys are assumed to be $(1.8\text{cm})^2$ (including gap between keys).

We construct these features for candidate username and each prior username. Thus, over all prior usernames, each feature has a set of values. Adopting the technique outlined in Eq. (3.4) for compressing distributions as features, we construct $15 \times 5 = 75$ additional features for prior usernames.

Language Patterns. In addition to environmental factors, cultural priors such as language also affect the username selection procedure. Users often use the same or the same set of languages when selecting usernames. Therefore, when detecting languages of different usernames belonging to the same individual, one expects fairly consistent results. We consider the language of the username as a feature in our dataset. To detect the language, we trained an n-gram statistical language detector [44] over the European Parliament Proceedings Parallel Corpus [3], which consists of text in 21 European languages (*Bulgarian, Czech, Danish, German, Greek, English, Spanish, Estonian, Finnish, French, Hungarian, Italian, Lithuanian, Latvian, Dutch, Polish, Portuguese, Romanian, Slovak, Slovene,* and *Swedish*) from 1996-2006 with more than 40 million words per language. The trained model detects the candidate username language, which is a feature in our feature set. The language detector is also used on prior usernames, providing us with a language distribution for prior usernames, which again is compressed as features using Eq. (3.4). The *detected language* feature is limited to European languages. Our language detector will not detect other languages. The language detector is also challenged when dealing with words that may not follow the statistical patterns of a language, such as location names, etc. However, these issues can be tackled from a different angle as we discuss next.

[3] http://www.statmt.org/europarl/

3.2.3 Endogenous Factors

Endogenous factors play a major role when individuals select usernames. Some of these factors are due to 1) personal attributes (name, age, gender, roles and positions, etc.) and 2) characteristics, e.g., a female selecting username `fungirl09`, a father selecting `geekdad`, or a PlayStation 3 fan selecting `PS3lover2009`. Others are due to 3) habits such as abbreviating usernames or adding prefixes/suffixes.

1. **Personal Attributes and Personality Traits**

   Personal Information. As mentioned, our language detection model is incapable of detecting several languages, as well as specific names, such as locations, or others that are of specific interest to the individual selecting the username. For instance, the language detection model is incapable of detecting the language of usernames `Kalambo`, a waterfall in Zambia, or `K2` and `Rakaposhi`, both mountains in Pakistan. However, the patterns in these words can be captured by analyzing the alphabet distribution. For instance, a user selecting username `Kalambo` most of the time will create an alphabet distribution where letter 'a' is repeated twice more than other letters. Hence, we save the alphabet distribution of both candidate username and prior usernames as features. This will easily capture patterns like an excessive use of 'i' in languages such as Arabic or Tajik [35, 50], where language detection fails. Another benefit of using alphabet distribution is that not only it is language-independent, but it can also capture words that are meaningful only to the user.

   Username Randomness. As mentioned before, individuals who select totally random usernames generate no information redundancy. One can quantify the randomness of usernames of an individual and consider that as a feature that

can describe individuals and help identify them. For measuring randomness, we consider the entropy [34] of the candidate username's alphabet distribution as a feature. We also measure entropy for each prior username. This results in an entropy distribution that is encoded as features using aforementioned technique in Eq. (3.4).

2. **Habits**

"Old habits, die hard", and these habits have a significant effect on how usernames are created. Common habits are,

Username Modification. Individuals often select new usernames by changing their previous usernames. Some,

(a) add prefixes or suffixes,
   - e.g., `mark.brown` → `mark.brown2008`,

(b) abbreviate their usernames,
   - e.g., `ivan.sears` → `isears`, or

(c) change characters or add characters in between.
   - e.g., `beth.smith` → `b3th.smith`.

Any combination of these operations is also possible. The following approaches are taken to capture the modifications:

- To detect added prefixes or suffixes, one can check if one username is the substring of the other. Hence, we consider the length of the *Longest Common Substring (LCS)* as an informative feature about how similar the username is to prior usernames. We perform a pairwise computation of

LCS length between the candidate username and all prior usernames. This will generate a distribution of LCS length values, quantizied as features using Eq. (3.4). To get values in range [0,1], we also perform a normalized LCS (normalized by the maximum length of the two strings) and store the distribution as a feature as well.

- For detecting abbreviations, *Longest Common Subsequence* length, is used since it can detect non-consecutive letters that match in two strings. We perform a pairwise calculation of it between the candidate username and prior usernames and store the distribution as features using aforementioned technique in Eq. (3.4). We also store the normalized version as another distribution feature.
- For swapped letters and added letters, we use the normalized and unnormalized versions of both Edit (Levenshtein) Distance, and Dynamic Time Warping (DTW) [98] distance as measures. Again, the end results are distributions, that are saved as features.

Generating Similar Usernames. Users tend to generate similar usernames. The similarity between usernames is sometimes hard to capture using approaches discussed for detecting username modification. For instance, `gateman` and `nametag` are highly similar due to one being the other spelled backward, but their similarity is not recognized by discussed methods. Since we store the alphabet distribution for both the candidate username and prior usernames, we can compare these using different similarity measures. The Kullback-Liebler divergence (KL) [34] is commonly the measure of choice; however, since KL isn't a metric, comparison among values becomes difficult. To compare distributions, we use the Jensen-Shannon divergence (JS) [81], which is computed from KL

and is a metric,

$$JS(P||Q) = \frac{1}{2}[KL(P||M) + KL(Q||M)], \tag{3.6}$$

where $M = \frac{1}{2}(P + Q)$, and KL divergence is,

$$KL(P||Q) = \sum_{i=1}^{|P|} P_i \cdot log(\frac{P_i}{Q_i}). \tag{3.7}$$

Here, P and Q are the alphabet distributions for candidate username and prior usernames. As an alternative, we also consider cosine similarity between the two distributions as a feature. Note that Jensen-Shannon divergence does not measure the overlap between the alphabets. To compute alphabet overlaps, we add Jaccard Distance as a feature.

Username Observation Likelihood. Finally, we believe the order in which users juxtapose letters to create usernames depends on their prior knowledge. Given this prior knowledge, we can estimate the probability of observing candidate username. Prior knowledge can be gleaned based on how letters come after one another in prior usernames. In statistical language modeling, the probability of observing username `u`, denoted in characters as $u = c_1 c_2 \dots c_n$, is,

$$p(u) = \Pi_{i=1}^{n} p(c_i | c_1 c_2 \dots c_{i-1}). \tag{3.8}$$

We approximate this probability using an n-gram model,

$$p(u) \approx \Pi_{i=1}^{n} p(c_i | c_{i-(n-1)} \dots c_{i-1}). \tag{3.9}$$

Commonly, to denote the beginning and the end of a word special symbols are added: $\star$ and $\bullet$. So, for username `sara`, the probability approximated using a 2-gram model is,

$$p(sara) \approx p(s|\star)p(a|s)p(r|a)p(a|r)p(\bullet|a). \tag{3.10}$$

To estimate the observation probability of the candidate username using an n-gram model, we first need to compute the probability of observing its comprising n-grams. The probability of observing these n-grams can be computed using prior usernames. These probabilities are often hard to estimate, since some letters never occur after others in prior usernames while appearing in the candidate username. For instance, for candidate username `test12` and prior usernames {`test, testing`}, the probability of $p(\texttt{1}|\star\texttt{test}) = 0$ and therefore $p(\texttt{test12}) = 0$, which seems unreasonable. To estimate probabilities of unobserved n-grams, a smoothing technique can be used. We use the state-of-the-art *Modified Kneser-Ney (MKN)* smoothing technique [27], which has discount parameters for n-grams observed once, twice, and three times or more. The discounted values are then distributed among unobserved n-grams. The model has demonstrated excellent performance in various domains [27]. We include the candidate username observation probability, estimated by an MKN-smoothed 6-gram model, as a feature.

We have demonstrated how behavioral patterns can be translated to meaningful features for the task of user identification. These features are constructed to mine information hidden in usernames due to individual behaviors when creating usernames. Overall, we construct 414 features for the candidate username and prior usernames. Figure 3.2 depicts a summary of these behavioral patterns observed in individuals when selecting usernames.

Clearly, our features do not cover all aspects of username creation, and with more theories and behaviors in place, more features can be constructed. We will empirically study if it is necessary to use all features and the effect of adding more features on learning performance of user identification.

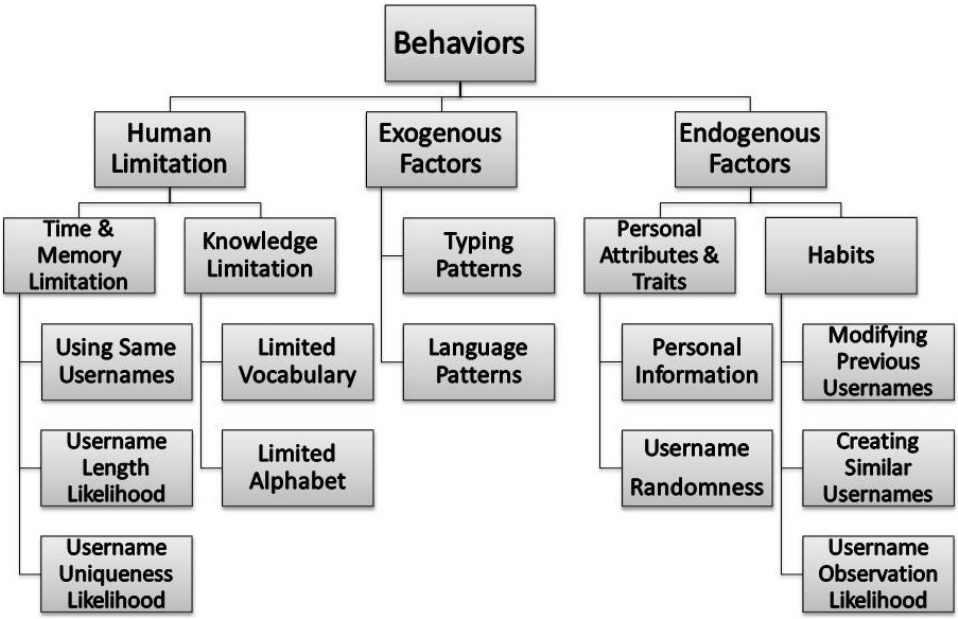


Figure 3.2: Individual Behavioral Patterns when Selecting Usernames

Following MOBIUS methodology, the feature values are computed over labeled data, and the effectiveness of MOBIUS is verified by learning an identification function. Next, experiments for evaluating MOBIUS are detailed.

3.3 Experiments

The MOBIUS methodology is systematically evaluated in this section. First, we verify if MOBIUS can learn an accurate identification function, comparing with some baselines. Second, we examine if different learning algorithms make significant difference in learning performance using acquired features. Then, we perform feature importance analysis, and investigate how the number of usernames and the number of features impact learning performance. Before we present our experiments, we detail how experimental data is collected.

3.3.1 Data Preparation

A simple method for gathering identities across social networks is to conduct surveys and ask users to provide their usernames across social networks. This method can be expensive in terms of resource consumption, and the amount of gathered data is often limited. Companies such as Yahoo! or Facebook ask users to provide this kind of information [4]; however, this information is not publicly available.

Another method for identifying usernames across sites is by finding users manually. Users, more often than not provide personal information such as their real names, E-mail addresses, location, gender, profile photos, and age on these websites. This information can be employed to map users on different sites to the same individual. However, manually finding users on sites can be quite challenging.

Fortunately, there exist websites where users have the opportunity of listing their identities (user accounts) on different sites. This can be thought of as *labeled* data for our learning task, providing a mapping between identities. In particular, we find social networking sites, blogging and blog advertisement portals, and forums to be valuable sources for collecting multiple identities of the same user.

Social Networking Sites. On most social networking sites such as Google+ or Facebook, users can list their IDs on other sites. This provides usernames of the same individual on different sites.

Blogging and Blog Advertisement Portals: To advertise their blogs, individuals often join *blog cataloging* sites to list not only blogs, but also their profiles on other sites. For instance, users in BlogCatalog are provided with a feature called "My Communities". This feature allows users to list their usernames in other social media sites.

[4] http://mashable.com/2010/10/17/y-connect-yahoo/

Forums: Many forums use generic Content Management Systems (CMS), designed specifically for creating forums. These applications usually allow users to add their usernames on social media sites to their profiles. Examples of these applications that contain this feature include, but are not limited to: vBulletin, phpBB, and Phorum.

We utilize these sources for collecting usernames, guaranteed to belong to the same individual. Overall, 100,179 (c-U) pairs are collected, where c is a username and U is the set of prior usernames. Both c and U belong to the same individual. The dataset contains usernames from 32 sites such as: Flickr, Reddit, StumbleUpon, and YouTube. This dataset contains all the usernames (nodes) collected in Section 2.2.1 as well as additional usernames to make our results comparable.

The collected pairs are considered as positive instances in our dataset. For negative instances, we construct instances by randomly creating pairs (c_i-U_j) such that c_i is from one positive instance and U_j is from a different positive instance ($i \neq j$) to guarantee that they are not from the same individual. We generated different numbers of negative instances (up to 1 million instances), but its effect on the accuracy of learning the identification function was negligible. By further investigation we noticed that this phenomenon takes placed due to feature values for negative instances being far different from that of positive instances. Thus, we continue with a dataset where the class balance is 50% for each label (100,179 positive + 100,179 negative ≈200,000 instances). Then, we compute our 414 feature values for this data and employ this dataset for our learning framework.

3.3.2 Learning the Identification Function

To evaluate MOBIUS, the first step is to verify if it can learn an accurate identification function. Given our labeled dataset where all feature values are calculated, learning the identification function can be realized by performing supervised learning

Table 3.1: MOBIUS Performance Compared to Content-Based Methods and Baselines

Technique	**Accuracy**
MOBIUS (Naive Bayes)	91.38%
Method of Zafarani et al. [136]	66.00%
Method of Perito et al. [106]	77.59%
Baseline b_1: Exact Username Match	77.00%
Baseline b_2: Substring Matching	63.12%
Baseline b_3: Patterns in Letters	49.25%

on our dataset. We mentioned earlier that a probabilistic classifier can generalize our binary identification function to a probabilistic one, where the probability of a candidate username belonging to an individual is measured. Probabilistic classification can be achieved by a variety of Bayesian approaches. We select Naive Bayes. Naive Bayes, using 10-fold cross validation, correctly classifies 91.38% of our data instances.

There is a need to compare MOBIUS performance to other content- and link-based methods. To the best of our knowledge, methods from Zafarani et al. [136] and Perito et al. [106] are the only content-based methods that tackle the same problem with usernames. The ad hoc method of Zafarani et al. employs two features: 1) exact match between usernames and 2) substring match between usernames. Perito et al.'s method uses a single feature. This feature, similar to our username-observation likelihood, utilizes a 5-gram model to compute the username observation probability. Table 3.1 reports the performance of these techniques over our datasets. Our method outperforms the method of Zafarani et al. by 38% and the method of Perito et al. by 18%. The key difference between MOBIUS and the methods in comparison is that MOBIUS takes a behavioral modeling approach that systematically generates features for effective user identification.

Table 3.2: MOBIUS Performance Compared to Link-Based Reference Points

Technique	AUC
MOBIUS (Naive Bayes)	0.937
Reference Point 1: Common Neighbors	0.504
Reference Point 1: Jaccard Coefficient	0.503
Reference Point 1: Adamic/Adar	0.501

To evaluate the effectiveness of MOBIUS, we also devise three content-based baseline methods for comparison. When people are asked to match usernames of individuals, commonly used methods are "exact username matching", "substring matching", or finding "patterns in letters". Hence, they form our three baselines b_1, b_2, and b_3:

b_1**: Exact Username Match.** It considers an instance positive if the candidate username is an exact match to $\alpha\%$ of the prior usernames. To set α accurately, we computed the percentage of prior usernames that are exact matches to the candidate username in each of our positive instances and averaged it over all positive instances to get α, $\alpha \approx 54\%$. To further analyze the impact, we set $50\% \leq \alpha \leq 100\%$. Among all α values, b_1 does not perform better than 77%.

b_2**: Substring Matching.** It considers an instance positive if the mean of the candidate username's normalized longest common substring distance to prior usernames is below some threshold θ. We conduct the experiment for the range $0 \leq \theta \leq 1$. In the best case, b_2 achieves 63.12% accuracy.

b_3**: Patterns in Letters.** For finding letter patterns, b_3 uses the alphabet distribution for the candidate username and the prior usernames as features. Using our data labels, we perform logistic regression. b_3 achieves 49.25% accuracy.

Our proposed technique outperforms baseline b_1, b_2, and b_3 by 19%, 45%, and 86%, respectively. The performance for MOBIUS trained by Naive Bayes, other

content-based methods, and baselines are summarized in Table 3.1.

To evaluate MOBIUS against link-based methods, we compare it to well-known unsupervised link prediction methods. As MOBIUS does not use link information, the performance of link-based methods only serve as reference points and no improvement will be reported. The methods included as reference points are *Common Neighbors*, *Jaccard Coefficient*, and *Adamic/Adar* [80][5]. Comparison between MOBIUS and the link-based reference points are provided in Table 3.2. Now, we would like to see if different learning algorithms can further improve the learning performance.

3.3.3 Choice of Learning Algorithm

To evaluate the choice of learning algorithm, we perform the classification task using a range of learning techniques and 10-fold cross validation. The AUCs and accuracy rates are available in Table 3.3. These techniques have different learning biases, and one expects to observe different performances for the same task. As seen in the table, results are not significantly different among these methods. This shows that when sufficient information is available in features, the user identification task becomes reasonably accurate and is not sensitive to the choice of learning algorithm. In our experiments, ℓ_1-Regularized Logistic Regression is shown to be the most accurate method and hence, we use it in the following experiments as the method of choice. The classification employs all 414 features. Designing 414 features and computing their values is computationally expensive. Therefore, we try to empirically determine: 1) whether all features are necessary, and 2) whether it makes *economic* sense to add more features, in Sections 3.3.3 and 3.3.4.

[5]As our dataset lacks link information, we report the best performances obtained across networks using [146]

Table 3.3: MOBIUS Performance for Different Classification Techniques

Technique	**AUC**	**Accuracy**
J48 Decision Tree Learning	0.894	90.87%
Naive Bayes	0.937	91.38%
Random Forest	0.957	93.59%
ℓ_2-Regularized ℓ_2-Loss SVM	0.950	93.70%
ℓ_1-Regularized ℓ_2-Loss SVM	0.951	93.71%
ℓ_2-Regularized Logistic Regression	0.950	93.77%
ℓ_1-Regularized Logistic Regression	0.951	**93.80**%

Feature Importance Analysis

Feature Importance Analysis analyzes how important different features are in learning the identification function. First, for each behavior we have identified, we group the respective features and measure their impact on the classification task. That is we only use those features in MOBIUS for classification. We previously provided the hierarchy of these behaviors in Figure 3.2. For each node in this hierarchy (other than the root), we create a feature set and train MOBIUS using only those features. Table 3.4 provides the performance of MOBIUS with these feature sets. As shown in the Table, features that describe endogenous factors or human limitations are the most effective for user identification. In terms of human limitations, features that capture limitations in time and memory are most suitable for user identification. Similarly, features that capture typing patterns and habits are most suitable from exogenous and endogenous factors, respectively. Finally, the most effective features for user identification are those that capture users' habits.

This analysis does not show individual features that contribute the most to the classification task. Next, we find these individual features. This can be performed

Table 3.4: MOBIUS Performance for Different Behaviors

Set of Features	Accuracy
I. Human Limitations	87.70
- Limitations in Time and Memory	87.70
— Selecting the Same Username	52.42
— Username Length Likelihood	55.88
— Username Creation Likelihood	60.81
- Knowledge Limitations	51.17
— Limited Vocabulary	51.24
— Limited Alphabet	48.55
II. Exogenous Factors	57.37
- Typing Patterns	57.43
- Language Patterns	51.40
III. Endogenous Factors	93.78
- Personal Information	49.25
- Username Randomness	56.00
- Habits	93.65
— Username Modification	93.64
— Generating Similar Usernames	78.37
— Username Observation Likelihood	48.54

by standard feature selection measures such as Information Gain, χ^2, among others. We utilize *odds-ratios* (logistic regression coefficients) for feature importance analysis and ranking features. The top 10 important features are as follows:

1. Standard deviation of normalized edit distance between the candidate username and prior usernames,
2. Standard deviation of normalized longest common substring between the user-

name and prior usernames,

3. Username observation likelihood,
4. Uniqueness of prior usernames,
5. Exact match: number of times candidate username is seen among prior usernames,
6. Jaccard similarity between the alphabet distribution of the candidate username and prior usernames,
7. Standard deviation of the distance traveled when typing prior usernames using the QWERTY keyboard,
8. Distance traveled when typing the candidate username using the QWERTY keyboard,
9. Standard deviation of the longest common substring between the username and prior usernames, and
10. Median of the longest common subsequence between the candidate username and prior usernames.

In fact, a classification using only these 10 features and logistic regression provides an accuracy of 92.72%, which is very close to that of using the entire feature set. We also notice that in our ranked features,

- Numbers `[0-9]` are on average ranked higher than English alphabet letters `[a-z]`, showing that numbers in usernames help better identify individuals, and
- Non-English alphabet letters or special characters, e.g., $\hat{A}$`,`$\tilde{A}$`,+,` or `&`, are among the features that could easily help identify individuals across sites, i.e., have higher odds-ratios on average.

Although these 10 features perform reasonably well, it is of practical importance to

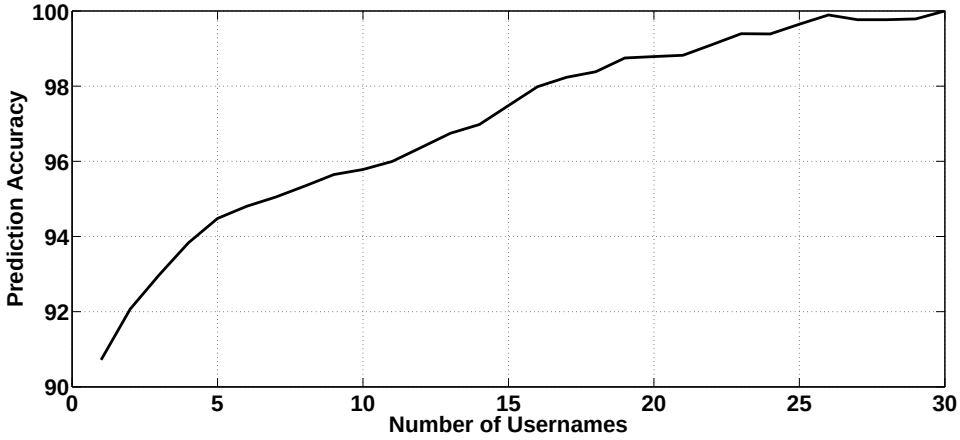


Figure 3.3: User Identification Performance for Users with Different Number of Usernames

analyze how we can further improve the performance of our methodology in different scenarios, such as by adding usernames or features.

3.3.4 Diminishing Returns for Adding More Usernames and More Features

It is often assumed that when more prior usernames of an individual are known, the task of identifying the individual becomes easier. If true, to improve identification performance, we need to provide MOBIUS with extra prior information (known usernames). In our dataset, users have from 1 to a maximum of 30 prior usernames. To verify helpfulness of adding prior usernames, we partition the dataset into 30 datasets $\{d_i\}_{i=1}^{30}$, where dataset d_i contains individuals that have i prior usernames. The user identification accuracy on these 30 datasets are shown in Figure 3.3. We observe a monotonically increasing trend in identification performance, and even for a single prior username, the identification is 90.72% accurate and approaches 100% when 25 or more usernames are available. Note that the identification task is hardest when only a single prior username is available.

Rarely are 25 prior usernames of an individual available across sites. It is more practical to know the minimum number of usernames required for user identification

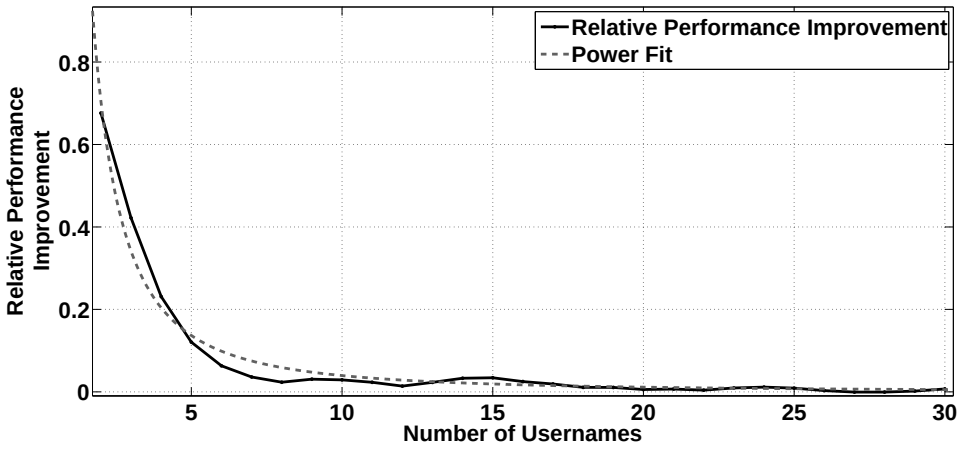


Figure 3.4: Relative User Identification Performance Improvement with respect to Number of Usernames

such that further improvements are nominal. The relative performance improvement with respect to number of usernames can help us measure this minimum. Figure 3.4 shows this improvement for adding usernames. We observe a *diminishing return* property, where the improvement becomes marginal as we add usernames and is negligible for more than 7 usernames. A power function ($g(x) = 2.44x^{-1.79}$), found with 95% confidence, fits to this curve with adjusted $R^2 = 0.976$. The exponent -1.79 denotes that the relative improvement by adding n usersnames is $\approx 1/n^{1.79}$ times smaller than that by adding a single username, e.g., for 7 usernames, relative identification performance improvement is $\approx 1/33$ times smaller than that of a single username.

Similar to adding more prior usernames, one can change number of features. More practically, we would like to analyze how adding features correlates with adding prior usernames. For instance, if we double the number of prior usernames, how many features should we construct (or can be removed) to guarantee reaching a required performance?

To measure this, for each number of prior usernames n, we compute the average number of features such that MOBIUS can achieve fixed accuracy θ. We set θ to the

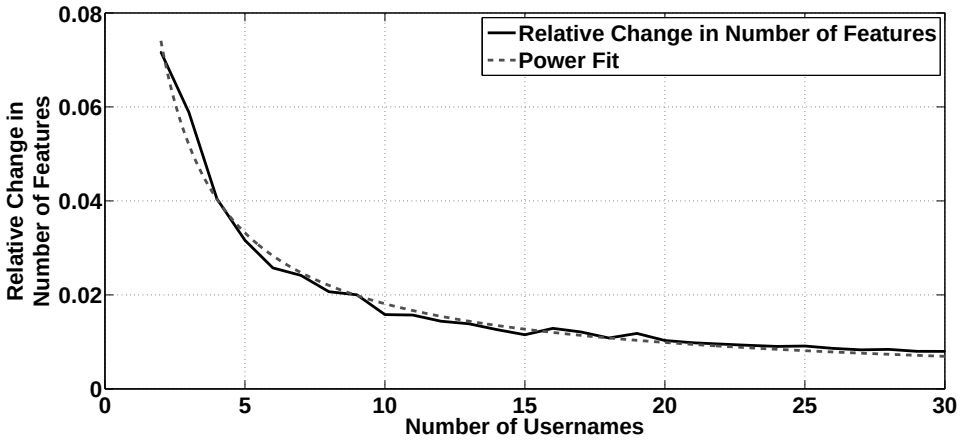


Figure 3.5: Relative Change in Number of Features Required with respect to Number of Usernames

minimum accuracy achievable, independent of number of usernames (90% here). Then we compute the relative change in the number of required features when usernames are added.

Figure 3.5 plots this relationship. We observe the same diminishing return property, and as one adds more usernames, fewer features are required to achieve a fixed accuracy. A power function ($g(x) = 0.1359x^{-0.875}$), found with 95% confidence, fits to this curve with adjusted $R^2 = 0.987$. The exponent -0.875 denotes that the number of features required for n usersnames is $\approx 1/n^{0.875}$ times smaller than that of a single username.

Finally, if one is left with a set of usernames and a set of features, should we aim at adding more usernames or construct better features? Let $f(n, k)$ denote the performance of our method for n usernames and k features. Let,

$$\delta(n, k) = \frac{f(n+1, k) - f(n, k)}{f(n, k+1) - f(n, k)}. \tag{3.11}$$

The δ function is a finite difference approximation for the derivative ratio with respect to n and k. When $\delta(n, k) > 1$, adding usernames improves performance more and when $\delta(n, k) < 1$, adding features is better. To compute $f(n, k)$, for different values of n, we select random subsets of size k. We denote the average performance

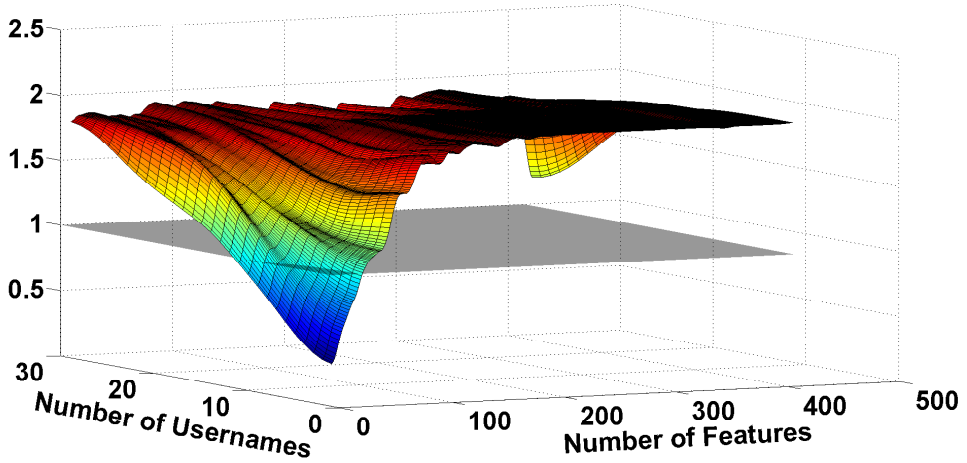


Figure 3.6: The $\delta(n, k)$ function, for n usernames and k features. Values larger than 1 show that adding usernames will improve performance more and values smaller than 1 show adding features is better.

over these random subsets as $f(n, k)$. Figure 3.6 plots the $\delta(n, k)$ function. We plot plane $z = 1$ to better show where adding features is more helpful and where usernames are more beneficial. We observe that for small values of n and k, i.e., when fewer usernames and features are available, features help best, but for all other cases adding usernames is more beneficial.

3.4 Discussion

We demonstrated that MOBIUS can exploit information redundancies due to user behaviors to identify individuals across sites. The empirical evaluation shows that MOBIUS is effective in across-site user identification.

Back to our initial questions, although we can tell if a username belongs to a username set, but given a username-set, where can we find the candidate usernames? Furthermore, as MOBIUS operates on usernames, a natural question is if there is additional information available such as location, how we can represent and integrate it into MOBIUS. These are practical questions that need to be answered to complete the task of identification

3.4.1 Finding Candidate Usernames

The candidate username needs to be found using the available tools and information. To most users, unless they have access to the deep or hidden web, the only gateway to find information is the public web and in particular, with tools such as web search engines; therefore, we focus on finding usernames on the public web via web search engines. In our experiments, we had several interesting observations that can lead to finding candidate usernames.

We found that for any two usernames, u_1 and u_2 of the same individual, there is a high chance of co-occurrence of these two in search engine results. To verify this, from our dataset we generated around 100,000 *username-username* pairs $< u_1, u_2 >$ where both u_1 and u_2 belonged to the same individual. We found using Google with query "$\mathtt{u_1}$ $\mathtt{u_2}$" that usernames co-occur in nearly 68% of the cases in web search engine results. This finding suggests that we can perform a web search using one of the usernames and then perform keyword extraction on the retrieved webpages to discover the other usernames; however, though sufficiently accurate, in some cases, the retrieved pages are many and long and keyword extraction can be quite tedious and will generate many candidate usernames. Our other observations lead to a solution to mitigate this problem. We will review them first before coming back to a solution to this problem.

We observed that for any social media site s and for all its usernames, there exists URLs on the Registered Domain Name of s that contain the username. These URLs are most commonly pointing to the profile/homepage of the users on that site. Denote these URLs as *Profile URLs*. As an example, consider how the profile page URLs of a fictional user *test* can be reached on some of the most popular social networking sites in Table 3.5. We have analyzed 32 online sites in our dataset and surprisingly,

Table 3.5: Profile URLs for Popular Social Media Sites

Site	Profile URL Pattern
YouTube	`http://www.youtube.com/test`
Flickr	`http://www.flickr.com/photos/test`
Reddit	`http://www.reddit.com/user/test`
Del.icio.us	`http://del.icio.us/test`

in all 32, the site's profile URLs contains the username.

Back to our original problem, interestingly, users often list their other usernames on Profile URLs. For instance, on their profile pages, they list their email addresses, where its part before the @ sign, is a commonly employed username of the individual. In other words, for two usernames u_1 and u_2 of the same individual, it is sufficiently likely for u_1 to exist in the *URL of the webpages* retrieved using popular search engines, such that the page itself contains u_2, i.e., u_1 profile page contains u_2.

To verify this, we used our 100,000 *username-username* pairs and for each pair $< u_1, u_2 >$, two separate queries were sent to Google (first username occurring on second username's profile, and vice versa). In Google, the queries can be formulated in the following format: "`inurl:`u_1 u_2" and "`inurl:`u_2 u_1". This phenomenon holds in nearly 38% of the situations. Likewise our previous observation, this suggests that we can perform a web search using one of the usernames and then perform keyword extraction on the **URLs of the webpages** retrieved to discover other usernames.

3.4.2 Adding More Information

MOBIUS can use other types of information that is available on social media sites. In general, we can follow the following procedure to integrate new types of information: 1) determine the behavioral patterns that humans exhibit regarding

that information, and 2) construct features to capture information redundancies due to behavioral patterns. For example, we have information beyond username such as individual's *location* that is often available on profile pages. Corresponding to candidate username (c) and prior usernames (U), we have *candidate location* and *prior locations*. One behavioral pattern associated with location is that individuals rarely change their locations. In fact, locations change much less than usernames. Therefore, based on this behavioral pattern, we can have an *exact location match* feature that counts the number of times candidate location is observed among prior locations. One can design additional features to capture similarity between candidate location and prior locations. For example, APIs such as the Google Maps API can be used to convert locations to latitude-longitude pairs and then distances between locations can be used to measure similarity.

As the availability of different types of information varies, such information is not as universally available as usernames. However, we believe more information should help identify users better and further investigation is needed to analyze performance gains due to additional information.

3.4.3 Data Collection Limitations

The data collection approaches discussed in this and previous chapter have some inherent limitations:

1. **Completeness.** It is not clear how complete the cross-media data that gathered in this study is. In other words, can we guarantee that we have sufficient data that describes all user behavior across sites? To approach this problem systematically we require complete ground truth about such data across sites. While we haven't approached this problem systematically, we will introduce an evaluation approach, similar to bootstrapping, in Chapter 7 that can be help

evaluate without ground truth. Similarly, techniques discussed in [143] can help evaluate when there is not ground truth in social media research.

2. **Bias.** There is an inherent bias in the data that we have collected as it is selectively reported by users across sites. It is therefore necessary to determine the amount and the statistics of user accounts across sites that are non-reported. Using a large ground truth dataset of user accounts across sites, one can measure this type of bias in our data.

3.5 Related Work

In this section, we focus on summarizing research related to identifying individuals in social media. We provided a review of directly relevant techniques to our study in Section 3.3. In addition to those, the methods of Iofciu et al. [64] and Liu et al. [82] approach the same problem but with extra information. Iofciu et al. utilize tag information in addition to a single username feature and Liu et al. use profile metadata, friendship network information, and content based features. Both methods rely on the availability of information that may not be available on social media. Our method only uses username information across sites.

In addition to these methods, there exists related research about 1) *identifying content produced by an individual on the web* or 2) *identifying individuals in a single social network*.

Identifying Content Authorship. In [12], the authors look at the content generation behavior of the same individuals in several collections of documents. Based on the overlap between contributions, they propose a method for detecting pages created by the same individual across different collections of documents. They use a method called detection by compression, where Normalized Compression Distance

(NCD) [29] is used to compare the similarity between the documents already known to be authored by the individual and other documents. Author detection has been well discussed in restricted domains. In particular, machine learning and data mining techniques have been employed to detect authors in online messages [147], online message boards [2, 103], blogs [70], and in E-mails [39]. Although, one can think of usernames as the content generated by individuals across sites; however, in content authorship detection, it is common to assume large collections of documents, with thousands of words, available for each user, whereas for usernames, the information available is limited to one word.

User Identification on One Site. Deanonymization[6] is an avenue of research related to identifying individuals on a single site. Social networks are commonly represented using graphs where nodes are the users and edges are the connections. To preserve privacy, an anonymization process replaces these users with meaningless, randomly generated, unique IDs. To identify these masked users, a deanonymization technique is performed. Deanonymization of social networks is tightly coupled with the research in privacy preserving data mining [9] or Identity Theft attacks [22]. In [15], Backstrom et al. present such process where one can identify individuals in these anonymized networks by either manipulating networks before they are anonymized or by having a priori knowledge about certain anonymized nodes. Narayanan and Shmatikov in [99] present statistical deanonymization technique against high-dimensional data. They argue that given little information about an individual one can easily identify the individual's record in the dataset. They demonstrate the performance of their method by uncovering some users on the Net-

[6]Deanonymization is tightly coupled with the research in privacy preserving data mining (see [6, 7, 9, 41, 48])

flix prize dataset using IMDB information as their source for background knowledge. Our work differs from these techniques as it deals with multiple sites. Moreover, it avoids using link information, which is not always available on different social media sites.

3.6 Summary

In this chapter, we have demonstrated a methodology for connecting individuals across social media sites (MOBIUS). MOBIUS takes a behavioral modeling approach for systematic feature construction and assessment, which allows integration of additional features when required. MOBIUS employs minimal information available on all social media sites (usernames) to derive a large number of features that can be used by supervised learning to effectively connect users across sites. Users often exhibit certain behavioral patterns when selecting usernames. The proposed behavioral modeling approach exploits information redundancy due to these behavioral patterns. We categorize these behavioral patterns into (1) human limitations, (2) exogenous factors, and (3) endogenous factors. In each category of behaviors, various features are constructed to capture information redundancy. MOBIUS employs supervised learning to connect users. Our empirical results show the advantages of this principled, behavioral modeling approach over earlier methods. The experiments demonstrate that (1) constructed features contain sufficient information for user identification; (2) importance or relevance of features can be assessed, thus features can be selected based on particular application needs; and (3) adding more features can further improve learning performance but with diminishing returns, hence, facing a limited budget, one can make informed decisions on what additional features should be added.

Chapter 4

UTILIZING MINIMUM INFORMATION IN APPLICATIONS

My powers are ordinary. Only
my application brings me success.

Isaac Newton

In previous chapters, we have shown how minimum link or content information can help identify users across sites. In this chapter, we demonstrate how minimum content information can be used in other applications on the web. We focus on two fundamental problems on social media: friends recommendation and malicious user detection. Both problem are significant to most social media sites as they guarantee revenue and protect sites against malicious users. The approach discussed for both problems utilizes only minimum information.

4.1 Friend Recommendation with Minimum Information

With the rise of social media and the growth of modern technology, millions of sites are at our fingertips. With so many choices, our attention spans are decreasing rapidly. An average user spends less than a minute on an average site [1]. The problem becomes more challenging for commercial sites, especially for new sites that are desperately trying to attract new users and hoping to keep them active. This lack of interest in users was clearly observed in the early years of sites such as Twitter or Facebook with around 60% of their users quitting within the first month [26].

The content in this chapter has been published at SDM 2014 [140] and CIKM 2015 [142].

As consumers of social media, we are constantly seeking sites that can keep us engaged. User engagement can come from interesting content as well as from our social interactions. It is known that the existence of friends, relatives, or colleagues on sites, provides a sense of comfort, piques our interest on the site, and increases the likelihood of joining sites [16]. Finding users' friends on sites increases users' engagement and improves user retention rates, both directly contributing to more revenue for the sites. So, *how can we find friends of users*?

Finding or recommending friends is not a new problem [80]. It is well-studied in social media research. Often, link or content information, or a combination of both, is used to predict and recommend friends to users.

When using link information, we use the current friends of an individual to recommend new friends. For instance, we find potential friends for John by finding friends-of-friends of John that are still not his friends. Hence, we find users that are 2 hops away in the friendship network. We can improve recommendations by recommending users that are more than two hops away in the friendship network. Unfortunately, recommending friends using link information fails when the user has no friends. This can happen right after a user joins a new site, when the user is a disconnected node in the friendship graph. Sites such as Twitter or LinkedIn, tackle this issue by asking users to provide access to their email contacts to help recommend friends. Aside from its privacy concerns, this clearly requires an extra effort from the user's side, and motivates users, with their short attention spans, to abandon the site.

When using content information, friend recommendation techniques identify potential friends for a user by finding others that are highly similar in terms of the content that they generate. This content can be profile information, tweets, reviews, blogposts, or even the products bought. However, right after a user joins a new site, the user hasn't had the chance to complete profile information or exhibit any activity.

Hence, findings friends with no link and content information is challenge for *all* social media sites and for all users, right after they join the sites. A variant of this problem is often referred to as the *cold start* problem.

The cold start is well-studied in the literature [110]; however, the solution often assumes that either link or content information is available. However, when a user joins a new site, link information (friends) or content information (bio, posts, etc.) is unavailable; therefore, relying on either type of information is impossible. In practice, sites such as Twitter address this problem by recommending individuals that have many friends such as celebrities or political figures in the United States to newly-joined users. Some users may find these recommendation interesting, but it can be repelling to users that are from other countries or have limited English knowledge. Ultimately, for a new user and without link or content information, finding friends in a site with one million members boils down to random recommendations of a few users from a *search space* of one million potential friends. Alas, recommending friends uniformly at random from this space is extremely unlikely to find any friends.

In this section, we demonstrate a methodology to find friends for a new user when link or content information is unavailable. Relying on social forces that result in friendships, we demonstrate how one can employ minimum user information to significantly reduce the set of potential friends; hence, increasing the likelihood of finding friends. We demonstrate how this minimum information can increase friend finding performance sometimes by four orders of magnitude (Section 4.1.4). The proposed methodology can help sites introduce the very first few friends more accurately. This will increase the chance for users to add friends, which in turn provides sufficient link information for more advanced link prediction techniques to recommend more friends.

Section 4.1.1 formally presents the problem of finding friends in social media sites with minimum information. Section 4.1.2 outlines how different social forces result in

friendships and how one can utilize the outcome of these forces to tackle our problem. Section 4.1.4 outlines our experiments and Section 4.1.5 reviews some related work.

4.1.1 Problem Statement

Consider a new site S with n users. When an individual joins S with no content or link information, the site has probability $p = 1/n$ to correctly recommend a single friend and a search space of n to search for that friend. Given the enormous size of current social media sites such as Twitter and Facebook, we can safely assume that a new user has some potential friends on the site.

Let set $U = \{u_1, u_2, \ldots, u_n\}$ represent the set of current users on site S and u_{new}, the newly-joined user. Consider a k-partitioning of current users $\pi(U)$,

$$\pi(U) = (X_1, X_2, \ldots, X_k), \tag{4.1}$$

$$\cup_{i=1}^{k} X_i = U, \tag{4.2}$$

$$X_i \cap X_j = \emptyset, \quad i \neq j. \tag{4.3}$$

To realistically model the problem in social media, and without loss of generality, we assume link information is available for current users $u_i \in U$, and unavailable for u_{new}. Assume link information is provided as an adjacency matrix $A \in \mathbb{R}^{n \times n}$, where

$$A_{i,j} = \begin{cases} 1 & u_i \text{ is a friend of } u_j; \\ 0 & \text{Otherwise} \end{cases} \tag{4.4}$$

Consider *friendship matching function* f

$$f : u \to X_j, u \in U \cup \{u_{new}\}, 1 \leq j \leq k. \tag{4.5}$$

The friendship matching function maps the new or current users to a partition X_j, $1 \leq j \leq k$. We assume that the partition user u is mapped to $f(u) = X_j$ is a

partition in which it is highly likely to find friends for u. Thus, we denote partition $X_j = f(u)$ as the *friendship search space* for u.

Let $M(X_j) = \{u | u \in U, f(u) = X_j\}$ denote the set of *matched users* to partition X_j from U. As all members of $M(X_j)$ are likely to have friends in X_j, we are implicitly assuming some similarity between $M(X_j)$ members.

In our problem, the goal is to find the friendship search space $f(u_{new})$ for u_{new}. To find $f(u_{new})$, one needs to determine the partitioning $\pi(U)$ and friendship matching function f. Assume both are known and $f(u_{new}) = X_j$ is the friendship search space for u_{new}. As link information for u_{new} is unavailable, how can we verify if u_{new} has friends in X_j?

One solution is to follow a training/testing framework in data mining and assume that the probability of u_{new} having friends in X_j can be approximated using current matched users to X_j: $M(X_j)$, for whom we have link information. This probability, denoted as $P_f(X_j)$, approximates *link prediction accuracy* and is the fraction of matched users that have a friend in set X_j,

$$P_f(X_j) = \frac{\mid \{u_i \mid u_i \in M(X_j),\ \sum_{u_k \in X_j} A_{i,k} \geq 1\} \mid}{\mid M(X_j) \mid}. \tag{4.6}$$

Let $X_j^{Rand} \subseteq U$ denote a random subset of equal size to X_j, i.e., $|X_j| = |X_j^{Rand}|$. Users in X_j^{Rand} are selected uniformly at random. Hence, the probability that a user in $M(X_j)$ has a friend in X_j^{Rand}, which we denote as $P_f^{Rand}(X_j)$, is

$$P_f^{Rand}(X_j) = \frac{|X_j^{Rand}|}{|U|} = \frac{|X_j|}{|U|}. \tag{4.7}$$

$P_f^{Rand}(X_j)$ approximates random prediction accuracy for link prediction. Our goal in this study is to find friends by seeking partitions such as X_j, in which the probability of finding friends is much higher than random, i.e.,

$$\beta_{X_j} = \frac{P_f(X_j)}{P_f^{Rand}(X_j)} > 1, \tag{4.8}$$

where β_{X_j} denotes the *significance ratio*[1], which quantifies the rate at which partition X_j increases the friend finding likelihood for members of $M(X_j)$ over random predictions. Note that the search space is reduced by $1/\beta_{X_j}$. Clearly, when no information is available, one cannot go beyond random: $\beta_{X_j} = 1$. The value for β_{X_j} is maximized when all users in $M(X_j)$ have at least one friend inside X_j. Thus, when sites such as Twitter recommend individuals with many friends to new users (e.g., X_j ={celebrities or political figures}), they are providing a relaxed solution to finding an optimal X_j.

The value of β_{X_j} can become deceiving, since for small values of $|M(X_j)|$, $P_f(X_j)$ can become large (see Equation (4.6)); therefore, extremely larger than $P_f^{Rand}(X_j)$. Furthermore, since u_{new} (and users joining later) can be matched to different partitions, one needs to compute the significance ratio for different partitions. Both issues can be addressed by computing the expected β for a partitioning[2] $\pi(U)$,

$$\mathbb{E}(\beta) = \sum_j \beta_{X_j} \frac{|M(X_j)|}{|U|}. \tag{4.9}$$

Thus, our goal is to find a partitioning of the users $\pi(U)$ and a friendship matching function f such that $\mathbb{E}(\beta) > 1$. To go beyond random prediction $\mathbb{E}(\beta) = 1$, we only use minimum information available on sites for users. As discussed in Chapter 3, the minimum amount of information available for a user on a site is the individual's username. Usernames are alphanumeric strings or email addresses without which users are incapable of joining sites. Therefore, we formulate our problem with usernames.

Definition. ***Finding Friends with Minimum Information.*** *In a site with n users represented by their usernames $U = \{u_1, u_2, \ldots, u_n\}$ and their friendship adja-*

[1]Following the statistical convention of assuming $P_f^{Rand}(X_j)$ as the null hypothesis, this ratio indicates how significant partition X_j is in predicting friends.

[2]The more accurate version of this Equation is $\mathbb{E}(\beta) = \sum_j \beta_{X_j} \frac{|M(X_j)|}{\sum_j |M(X_j)|}$. In later sections, we assume that $\cup_{i=1}^{k} M(X_i) = U$ and $M(X_i) \cap M(X_j) = \emptyset$, i.e., $\sum_j |M(X_j)| = |U|$. Hence, to avoid confusion in future sections, the term $\sum_j |M(X_j)|$ is substituted with $|U|$.

cency matrix $A \in \mathbb{R}^{n \times n}$*, finding friends with minimum information can be achieved by finding a partitioning of* U*,* $\pi(U) = (X_1, X_2, \ldots, X_k)$*, and a friendship matching function* f *such that* $\mathbb{E}(\beta) > 1$*.*

Hence, to find friends with minimum information, one has to determine (1) a partitioning of the usernames and a (2) matching of usernames to those partition such that $\mathbb{E}(\beta) > 1$. To find a solution, we analyze how friends are formed from a social science perspective.

4.1.2 Social Forces behind Friendships

In general, three major social forces result in friendships: (1) *homophily*; (2) *confounding*; and (3) *influence*. Homophily, or the social principle that "birds of a feather flock together," is observed when *similar* individuals become friends. The similarity between users is often observed in their interests (e.g., field of study), their personal attributes (e.g., gender), and the like. Fans of the same movie director becoming friends is an example of friendships formed by homophily. Confounding is observed when friendships are formed due to user similarities caused by the environment users live in. Friend formed by confounding are often in close proximity or speak the same language. Finally, influence is observed when users form friendships due to external factors, such as the authority of others. Befriending a public figure is an example of friendships formed by influence.

Interestingly, signs of similarity between users are observed in friendships formed by all three social forces. In homophily, friends are similar in terms of non-environmental attributes such as their interests. In confounding, friends are similar in terms of their environmental attributes such as their mother tongue or location. In influence, a user who befriends an influential user can be different from the influential in terms of the environmental or non-environmental attributes. However, the user often fits

well within the **crowd** who has already befriended the influential. For instance, users who befriend a famous tennis player are often similar in terms of liking tennis. Thus, in influence, the user befriending the influential is *similar to the crowd that has befriended the influential*. Due to these similarities, friends of the user are likely to have the exact same attribute value. Hence, to find friends one should aim at predicting user attributes, and in our situation, from usernames.

User attributes are non-random and leave digital traces in usernames [139]. These digital traces in usernames can be captured using data features. For instance, we expect individuals who speak the same language to shard statistical language pattern that can be gleaned from their usernames. Following the tradition in data mining research, we employ supervised learning to predict personal attributes of users solely from their usernames. For each social force, we select a corresponding user attribute for prediction that can best demonstrate the effect of friendships formed by that force. Next, we elaborate how specific user attributes are selected for each social force to be predicted from usernames.

4.1.3 Predicting Individual Attributes

As discussed, friendships are formed by three general social forces: homophily, confounding, and influence. Our goal is to predict user attributes that represent each social force from usernames. Our goal here is to demonstrate how simple user attributes that represent each social force can be predicted using only usernames. Later in our experiments, we measure how these predicted attributes help better find friends and show the effect of each social force on predicting friendships.

Homophily-based Friendships

Homophily is observed when similar individuals befriend others. User similarities in homophily are exhibited in non-environmental user attributes. A major non-environmental user attribute that is known to result in friendships is the user's age. One often observes that users in the same age range are more likely to befriend each other. This has been observed in numerous recent studies [100, 119]. For instance, Ugander et al. [119] noticed that younger users have less diverse friends in terms of age range, while older users exhibit a higher diversity. Among the attributes that result in homophily-based friendships, we select age due to to its strong influence on friendships. If the ages of individuals can be predicted from their usernames, one expects users in the same age range to have higher friendship likelihoods.

By predicting ages for current users of the site from their usernames, we are partitioning the site users into different sets, each set representing users in an age range. For a new user, once the age range is predicted from the username, one expects the user to be more likely to be connected to others in the partition of users in the same age range. Here, $\pi(U)$ represents partitions of different age ranges and $f(u_i) = X_j$ indicates that the predicted age range for u_i is the same as that of all members of X_j. Hence, $M(X_i) = X_i$, meaning that matched users to the partition are within the partition itself. But, how can we predict the age from the username?

An analysis of US social security records[3] for birth names since 1879 shows that name frequencies change over time. For instance, in Figure 4.1, we depict the popularity of first names: *Jennifer* and *Jacob* over time. For each year, the popularity of the first name is shown on a scale of [0,1]. Jennifer was the most popular female name between [1970-1984] whereas Jacob was the most popular male name from 1991 to

[3]http://www.ssa.gov/oact/babynames/

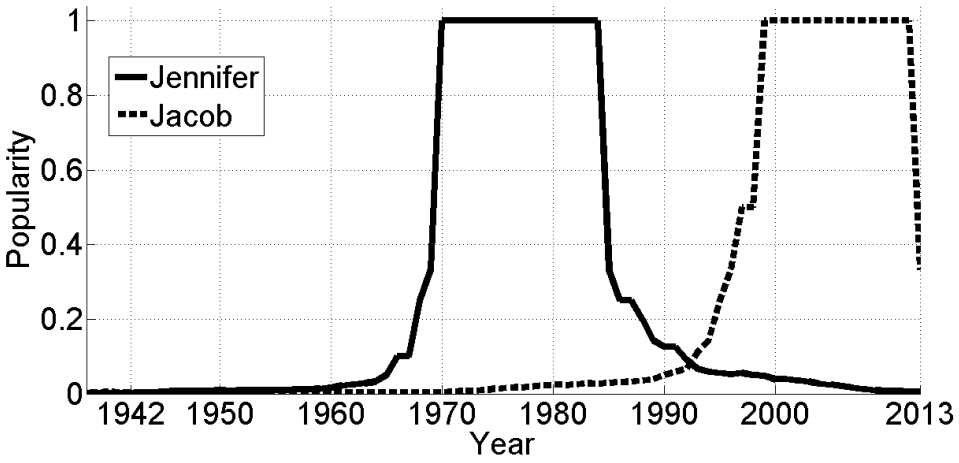


Figure 4.1: Popularity of First Names: *Jennifer* and *Jacob* over Time. Higher Values Depict more Popularity.

2012. Similar patterns can be observed for different English and non-English names given the diversity of the US population. This leads us to believe that given a name, one can provide an estimation of a likely age.

Personal attributes such as names are known to partially or completely exist in usernames and can be detected using the alphabet distribution of usernames [136, 139]. For instance, in our example, the probability of observing double n's in *Je**nn**ifer* is higher whereas, the probability of observing c and b is higher in *Jac**o**b*. Hence, the n-grams in usernames change depending on the age of the user. This is not only because of the popularity of names, but also because individuals of different ages have different vocabularies and interests that are exhibited in their usernames [139]. Thus, one can employ statistical language processing techniques to estimate ages of individuals from their usernames.

Confounding-based Friendships

We select two of the most prominent attributes from the attributes that are related to the environment that the users are living in: language and location. Similar to the age of individuals, we expect users living in close proximity or sharing the same language

to have a higher chance of becoming friends. Similar to the age attribute, $\pi(U)$ becomes the partitions of different locations (or languages), $f(u_i)$ matches u_i to the partition X_j, where members of X_j are in the same location as u_i, and $M(X_i) = X_i$, meaning that matched users are within the partition itself.

The language of individuals can significantly impact their chosen usernames. The language patterns can be easily observed both in the alphabet distribution as well as the n-grams of the username. For instance, while letter x is common when a Chinese speaker selects a username, it is rarely used by an Arabic speaker, since no Arabic word transliterated in English contains letter x. Similarly, excessive use of 'i' in languages such as Persian or Tajik [35, 50], can be easily detected in usernames.

Similarly, individuals from specific locations often have tendencies to utilize location-specific words or statistical patterns. While natives of Zambia, may use `Kalambo`, referring to a waterfall in Zambia, it is highly unlikely for users from elsewhere to include this word in their usernames.

Thus, to predict the location and language of the individuals one can utilize statistically significant alphabetical patterns in their usernames.

Influence-based Friendships

In friendships formed by influence, influential users attract friends. Hence, we can partition users attracting others in terms of the types of friends they are attracting and compare each partition with the new user for whom we are searching for friends. In general, we believe the deciding factor in becoming a member of the crowd that has befriended an influential is how the user fits in that crowd. We assume that a user fits in a crowd when at least one member of the crowd is similar to the user in terms of some attribute (environmental/non-environmental). We use all three attributes discussed so far: age, location, and language. Here, $f(u_i)$ matches u_i to

a partition X_j where each [influential] member of X_j has a friend with the same language, location, or age as u_i.

4.1.4 Experiments

The friendship search space reduction is systematically evaluated in this section. We determine the accuracy of finding friends for each one of the social forces, represented by their predicted attributes. Before we present our experiments, we detail how experimental data is collected.

Data Preparation

To analyze friendships, we collected a friendship graph of 135 million friendships from Reddit. These friendships are among 1.6 million users. For each friendship in this graph, we have the two usernames that are connected. We also collected separate datasets for predicting age and location of usernames.

1. **Age Dataset.** To predict age and to remove any bias associated with Reddit usernames, we collected a set 226,588 usernames from LiveJournal. In LiveJournal, users can list their ages. Among these users, 82,011 users have listed their age. This formed our training dataset for age prediction. The usernames in this dataset were vectorized using their alphabet distribution and frequent letter bigrams and their weights were normalized using TF-IDF. The ages were also divided into ten categories using an equal frequency binning and used as labels for this dataset. The age ranges in years are: $[0, 21.9]$, $[22, 23)$, $[23, 25)$, $[25, 26.5)$, $[26.5, 28)$, $[28, 30)$, $[30, 33)$, $[33, 36)$, $[36, 42)$, $[42, \infty)$.

2. **Location Dataset.** Similar to the age dataset, to remove bias for location prediction, we collected a dataset from Twitter. On Twitter, tweets can be geo-

located; that is, users carrying GPS-enabled devices can report their location with their tweets, which includes their usernames. The location is reported in (latitude,longitude) format. From Twitter, we collected a set of 36 million geo-located usernames with their latitudes and longitudes. Using a shapefile of all country borders and reverse geocoding, we determined the country for each username. Clearly, some countries have more geo-located tweets than others. To account for this imbalance, we clustered our dataset of latitudes and longitudes with k-means clustering.

For countries with less than 1,000 usernames we considered the whole country as one cluster. For all others, we clustered the geographical coordinates within the country using k-means with different k values until the obtained clusters had small enough radius. A recent study on Facebook [119] shows that users are more likely to befriend users that are within their 50 miles distance; thus, we ensured that the distance between any two members of the same cluster is close to this value. In our dataset, we found that by finding around 395 clusters, the clusters become well-balanced in size and small in radius across countries. The clustering of the usernames from the United States, including Alaska and Hawaii, is shown in Figure 4.2.

Although some clusters were still smaller than others, for most clusters, the difference is negligible, with the average datapoint distance to the cluster centroid being ≈ 36 miles. Since users in the same cluster are geographically close, we expect these users to have higher friendship likelihood. In this dataset, we use the cluster label as the class label for our training. Similar to our age dataset, the usernames are vectorized using their alphabet distribution and frequent letter bigrams and their weights are normalized using TF-IDF.

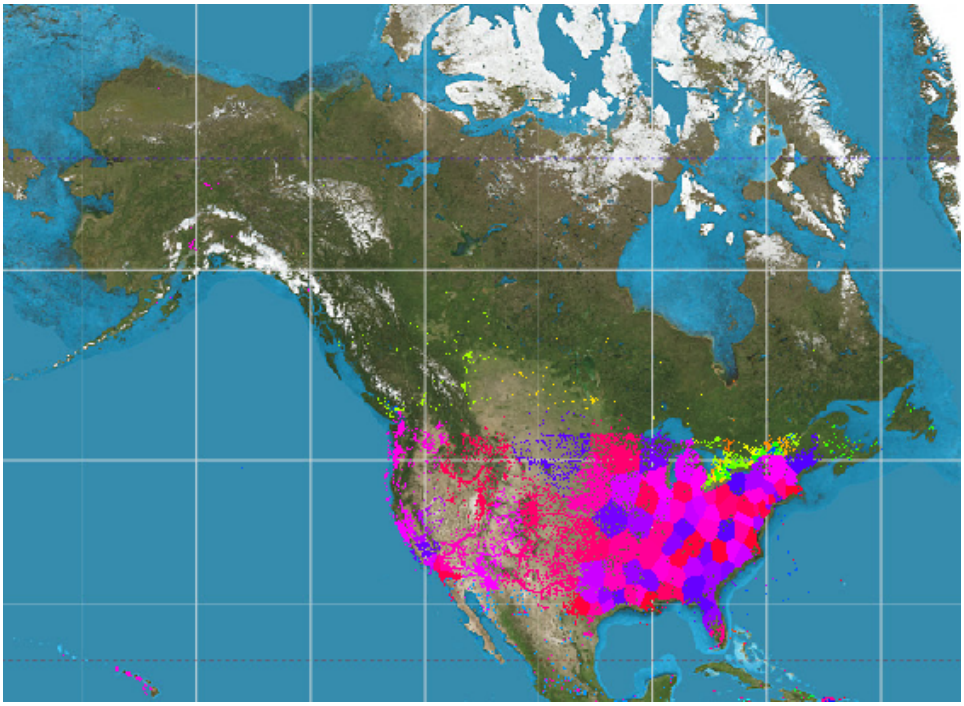


Figure 4.2: Usernames Clustered based on Location for the United States. Colors Represent Cluster Labels.

Learning Age, Location, and Language Predictors

We discuss how we train different classifiers to predict age, location, and language. Note that we are agnostic to the performance of these classifiers as long as these classifiers can reasonably predict the attributes. This is due to our goal to demonstrate the feasibility of finding friends by training such classifiers. Clearly, if our classifiers are capable of helping find friends, further classification improvements can further improve the friend recommendation. We leave classification improvement as a line of future research.

I. Predicting Language from Usernames. As usernames are often transliterated in Latin alphabet, one can more accurately predict the language of usernames for languages that employ Latin alphabets. We train an n-gram statistical language detector [44] over the European Parliament Proceedings Parallel Corpus[4], which consists

[4]http://www.statmt.org/europarl/

of text in 21 European languages (*Bulgarian, Czech, Danish, German, Greek, English, Spanish, Estonian, Finnish, French, Hungarian, Italian, Lithuanian, Latvian, Dutch, Polish, Portuguese, Romanian, Slovak, Slovene,* and *Swedish*) from 1996-2006 with more than 40 million words per language. The trained model can detects a username's language by decomposing it into different n-grams.

II. Predicting the Age from Usernames. Given our prepared dataset for the age. We trained a regularized logistic regression model that is able to predict the age of a username by decomposing it into n-grams. The model can predict age from a username.

III. Predicting Location from Usernames. The location dataset was clustered based on latitude-longitude values and cluster labels were used as class labels. We trained a regularized logistic regression model for this dataset. The trained model is capable of detecting the location of the username as one of the 395 classes that represent different locations.

Measuring Significance Ratios

With our trained classifiers, we predict age, location, and language for all 1.6 millions users. Then, for attributes representing each social social force, we measure significance ratios. For homophily and confounding, we measure significance ratios by measuring how many friends are of the same age, have the same location, or language. For influence, for user u_i and user u_j (represented using usernames), we measure how username u_i fits among the friends of u_j. We perform this separately for each of the three predicted attributes. In our experiments, we assume user u_i fits in friends of u_j, if at least one individual among friends of u_j has the same attribute value (age, location, or language) as u_i.

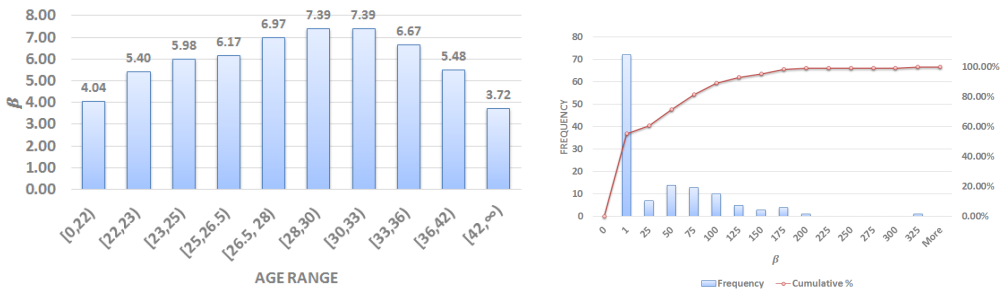
I. Homophily Significance

Among the set of 135 million friendships, we measure significance ratios for all age categories in our dataset: $[0, 21.9]$, $[22, 23)$, $[23, 25)$, $[25, 26.5)$, $[26.5, 28)$, $[28, 30)$, $[30, 33)$, $[33, 36)$, $[36, 42)$, $[42, \infty)$. The significance ratios are plotted in Figure 4.3(a). As shown in the figure, for all categories $\beta > 1$. This means that for example, when the predicted age of a username is between $[28 - 33]$, by recommending only other usernames where their ages are predicted to be in $[28 - 33]$, we are 7 times more accurate than randomly finding a friend. Note the significance of this result, compared to state of the art link prediction techniques that perform on average 2.4-54.4 times better than random prediction [80]; however, with *access to link information*. Our technique has no link information for the user for whom we are finding friends.

II. Confounding Significance

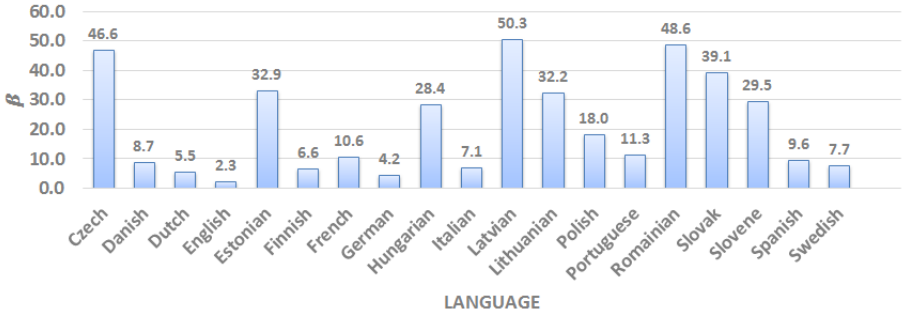
Similarly, we measure the significance ratios for different languages. We observe that for all languages $\beta > 1$. More importantly, we observe that when the language is detected as English, then β is minimum among all languages. This has two reasons. First, the majority of usernames are in English; therefore, conveying less information about friends. Secondly, lower similarity is observed among English users, as English is widely spoken across the globe and there is less likelihood for these speakers to befriend each other. In direct contrast are eastern European languages such as Romanian ($\beta = 48.6$) or more commonly spoken languages such as French ($\beta = 10.6$) that significantly improve friend finding performance.

We also measure the significance ratios for the location of usernames. Due to the large number of locations, we plot the histogram and the cumulative distribution (red line) of β values in Figure 4.3(b).



(a) Age

(b) Location



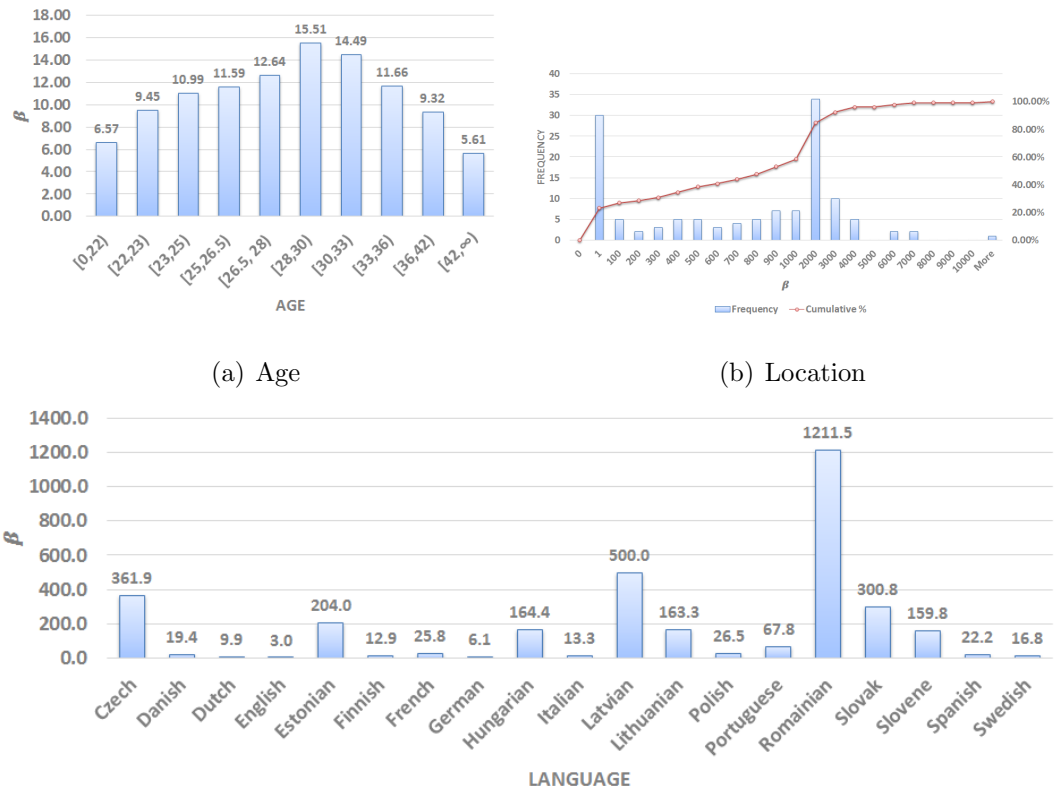
(c) Language

Figure 4.3: Significance Ratios (β) for Different Attributes

As shown in the figure, for more than 55% of locations we cannot predict any better than random. At the same time, for some predicted locations one can achieve as much as $\beta \approx 325$. After further investigation, we found that for the locations where $\beta = 1$, either the radius of the location cluster was larger than 50 miles or the size of the username cluster was small (few training instances). This in particular happens for countries where not many usernames are in our dataset. Thus, to better understand if there is any significance with respect to location, as well as other attributes, one needs to compute the expected value $\mathbb{E}(\beta)$. We will measure the expected values later where we compare different social forces in terms of friend finding performance.

III. Influence Significance

We measure significance ratio for influence using age, location, and language.



(a) Age

(b) Location

(c) Language

Figure 4.4: Influence Significance Ratios (β) for Different Attributes

These ratios are demonstrated in Figures 4.4. Comparing Figure 4.4 to Figure 4.3, we observe that in general finding friends based on influence (similarity to the friends of an individual) is much easier compared to homophily or confounding. On average, when finding friends based on influence, and using attribute age, the friend finding performance is improved by a factor of 1.79. Similarly, it is improved by a factor 5.14 when using the language attribute and a factor of 11.72 when considering locations. Hence, it seems that users prefer befriending individuals that have friends in their region over individuals who have friends sharing their language or are of the same age. To further analyze the effect of each social force, we measure the expected significance in finding friends for each social force next.

Table 4.1: Expected Improvement in Finding Friends over Random Predictions ($\mathbb{E}(\beta)$) for Different Social Forces.

Friend Finding Technique	$\mathbb{E}(\beta)$
Homophily - Age	5.49
Confounding - Location	6.19
Confounding - Language	5.19
Influence - Age	9.79
Influence - Language	16.29
Influence - Location	**31.04**

Comparison between Social Forces

As discussed in Section 4.1.1, the significance ratio at times can become deceiving. To mitigate this issue, we compute the expected β for homophily (age attribute), confounding (language or location attribute), and influence (for age, location, and language). The results are available in Table 4.1, showing an expected improvement factor between [5.49-31.04]. The table shows that though all forces can help find friends, influence-based friendships that are identified are at most 6 times more accurate compared to friends identified based on other social forces. Contrary to the common belief that similarity between users is the gist of forming friendships, this suggests that individual have far more tendencies to befriend a potential user when they feel welcomed in the crowd of friends of the potential user. We observe no significant difference between homophily and confounding in finding friends.

4.1.5 Related Work

To the best of our knowledge, the study presented in this section is the first to help find friends when link or content information is unavailable. However, one can find similar supervised or unsupervised link prediction methods when link or content

information is available.

Assuming usernames are content generated by users, one can compute the similarity between individuals and the similarity between their friends. In this case, well-established link prediction methods that use node similarity or neighborhood similarity such as the common neighbors [80], Adamic-Adar [4], Jaccard's Coefficient [80], or preferential attachment[80] are applicable. Note that when using contents generated by users, it is common to assume large collections of documents, with thousands of words, available for each user, whereas for usernames, the information available is limited to one word. Our technique, employs the knowledge of how social forces influence friendships and additional information such as age, language, and location that represent these social forces to reduce friendship search space, helping better predict future friends.

We have discussed a methodology to find friends with minimum information. Next, we investigate how minimum information can be used to detect malicious users.

4.2 Finding Malicious Users with Minimum Information

Social media sites are inundated with malevolent users. In June 2012, Facebook reported that 83 million of its user accounts are fake [123]; that is roughly the size of Egypt's population and larger than the population of 230 countries in the world [126]. Facebook reports that one-sixth of this population, that is **1.5%** of total Facebook users, are "undesirable" accounts that are created for malevolent purposes. Twitter faces similar challenges. In its security filings, Twitter claims 5% of its users are fake [46]; however, researchers estimate the percentage of its fake accounts to be as high as 10% [46]. These fake accounts are mostly sold for malicious purposes on black market for as low as $0.05 [46].

Malicious accounts may be created for different purposes. According to Cao et

al. [25], some malicious accounts are created for profitable activities, such as click fraud, identity fraud, and malware distribution. Others are created for social purposes such as pranks, stalking, cyberbullying, or identity concealing. The latter is often used in social online games. Online service providers find detecting and subsequently, suspending malicious accounts vital in order to protect their normal users against external threats.

Detecting malicious accounts dates back to the onset of social media. Comprehensive feature-based techniques, human-in-the-loop approaches, or techniques that use social-graphs are devised (see a review in Section 4.2.1). These techniques assume that a good amount of information about malicious users has been gathered. This information includes (1) the content that malicious users generate, (2) the activities they exhibit, or (3) the users they befriend. In short, their *content*, *activity*, or *links*. On the contrary, malicious users often do not have an incentive to generate content, exhibit activity, or befriend others. In addition, as malicious users join new sites, they lack sufficient content, link (i.e., friends), or activity to help detect them. Thus, there is a pressing need for detecting malicious users when only **minimum information** is available.

The study in this section aims to fill this gap by detecting malicious users when minimum information is available. In particular, we make the following contributions:

1. We introduce the first methodology to detect malicious users with minimum information. This methodology can be used as the first line of combat against malicious users on the web.
2. We identify five general characteristics of malicious activity and demonstrate how these characteristics are exhibited in the user generated content online.
3. We demonstrate that with as little as **10 bits** of information, one can distinguish between normal and malicious users.

4. We show via experiments that the methodology is robust and at least as effective as techniques that have access to more information.

In Section 4.2.1, we review the malicious user detection literature. We formally define the malicious user detection problem with minimum information in Section 4.2.2. We detail characteristics of malicious activity and how one can identify such characteristics in user content in Section 4.2.3. We detail our experiments in Section 4.2.4.

4.2.1 Literature on Malicious User Detection

While detecting malicious users with minimum information is unexplored, identifying malicious users in general is not a new topic. Often, to identify malicious users, (1) feature-based techniques, (2) human-in-the-loop techniques, or (3) techniques that use social graphs are used. We review representative techniques for each category and discuss how the current work relates to these techniques.

Feature-based Techniques. In Feature-based techniques, different features are constructed to describe the behavior of the malicious user. These features are then used to construct a dataset that is trained by a supervised learning framework. For instance, Xie et al. [128], develop the *AutoRe* framework that identifies botnet campaigns. Their framework identifies traffic that is bursty and distributed. These features of traffic help identify botnets. The bursty and distributed nature of unwanted content is also used in detecting malicious posts on Facebook [52]. Wang [122] introduces a method that detects spam on Twitter using network features such as the number of followers or friends and content features such as duplicated tweets. Feature-based techniques have been discussed extensively for detecting unwanted content in social tagging systems [75, 89], social networks [112], email [77], online videos [20], and

microblogging sites [19, 132]. Our work differs from the existing work in two aspects. First, current techniques for identifying malicious users often employ content or link information. Thus, one often needs a large collection of data instances to obtain guaranteed performances. Our approach employs minimum information across sites. Second, current literature is often context-dependent (e.g., site specific). Our method employs the minimum information that is universally available across sites and is robust even when information is collected from multiple sites.

Human-in-the-loop Methods. One approach of identifying malicious users is to employ human experts. Humans can naturally identify malicious users by their activities. Alternatively, one can combat malicious activities by technologies such as CAPTCHAs [121] or photo-based authentications [25] that are only solvable by humans. Although specific attacks are proposed for human-in-the-loop methods [97, 130], they are in general considered effective. Unfortunately, verifying accounts by humans is time consuming. For example, Tuenti, a Spain-based social networking service, hires humans to process reported users and block malicious ones [25]. An employee can only process 250 to 300 reports an hour from the daily 12,000 reports received. This issue makes human-in-the-loop processes infeasible for large-scale networks. Our approach in this section is automatic and can easily scale to billions of users.

Social Graph-based Techniques. In social-graph based methods, the information about the links (i.e., friendships) that the malicious individual has created helps identify the malicious user. For instance, Yang and colleagues [131] identify more than 100,000 fake accounts using social network features on RenRen social network. In particular, they find that invitation frequency, outgoing requests accepted, incoming requests accepted, and network clustering coefficient can help identify fake

accounts. In other works, probabilistic, combinatorial, or random walk models have been applied to network information to identify malicious users. Examples include, *Sybilguard* [133], *Gatekeeper* [118], *SybilInfer* [38], *SumUp* [117], and *Sybillimit* [134]. These methods or variants can be applied on sites such as Twitter to identify malicious users [53]. Mislove et al. [120] show that most techniques in this area function by finding local communities around trusted nodes. Assuming the existence of a social graph is a strong assumption. One often requires specific privacy permission to obtain such graphs and in specific cases, this graph is not available. In cases where there is no social graph, our methodology is still easily applicable.

4.2.2 Malicious User Detection with Minimum Information

Who is a malicious user? The definition varies in the literature from users that harass other users to users that jeopardize the privacy of others, and the like [25]. We consider malicious users on a site, those whom normal users consider malicious. Clearly, the opinion of normal users can be subjective and has to be verified by experts. In section 4.2.4, we demonstrate how such human-verified data can be collected. Humans are known to be accurate in detecting malicious users on social media [59, 65, 104]. However, as discussed in our literature review, human-in-the-loop approaches are time consuming and expensive for large-scale networks. Hence, by investigating how humans detect malicious users, one can not only scale detection of malicious users, but can also protect against a wide spectrum of malicious activities that are exhibited on social media [25, 28].

Our goal in this study is to identify such malicious users. Malicious users often provide little or no information. Hence, a method that can be universally employed on different sites is constrained to use the minimum information available on all sites. *Usernames* seem to be the minimum information available on all social media sites.

Often, usernames are alphanumeric strings or email addresses, without which users cannot join sites. Because of their unique characteristics, usernames are shown to be surprisingly effective for identifying individuals [139]. We formalize our problem using usernames as the minimum information available on all sites. Other content, link, or activity information such as user profile information or friends, when added to usernames, should help better identify malicious individuals. However, the lack of consistency in the availability of such information on all social media sites, directs us toward formulating our problem with usernames.

When using usernames, the goal is to detecting malicious users from their usernames. Hence, one can learn a function $\mathbf{M}(.)$ that given a username u, predicts whether the username belongs to a malicious user or not. We denote the $\mathbf{M}$ function as the *malicious user detection* function. Formally,

Definition. ***Malicious User Detection.*** *Given a username u, a malicious user detection procedure attempts to learn a malicious user detection function $\mathbf{M}(.)$, where*

$$\mathbf{M}(u) = \begin{cases} 1 & \textit{If } u \textit{ belongs to a malicious user;} \\ 0 & \textit{Otherwise.} \end{cases}$$

Malicious activities have distinctive characteristics. These characteristics leave traces in the usernames of malicious users in terms of *information redundancies*. These redundancies can be captured using data features. Following the common machine learning and data mining practice, the malicious user detection function can be learned by employing a supervised learning framework that utilizes these features and *labeled data*. In our problem, labeled data includes usernames that are known to be malicious or normal. For supervised learning, either classification or regression can be performed. Depending on the malicious user detection task at hand, one can even learn the probability that a username is malicious, generalizing our binary $\mathbf{M}$

function to a probabilistic one ($\mathbf{M}(u) = p$). This probability can help select the most likely malicious username. The learning of the malicious user detection function is the most straightforward. Therefore, we next elaborate on different characteristics of malicious activities and how features can be constructed to capture information redundancies introduced in usernames due to these characteristics. Note that the designed features may or may not help in the learning framework and are included as as long as they could be obtained. Later on in Section 4.2.4, we will analyze the effectiveness of all features, and if it is necessary to find as many features as possible.

In summary, to detect malicious users, we (1) identify characteristics of malicious activities, (2) construct features to identify traces of these characteristics in usernames, and (3) train a learning model to detect malicious users. Due to the interdependent nature of these characteristics and feature construction, we discuss them together next.

4.2.3 Characteristics of Malicious Activities

Humans detect malicious users on social media by the type of behavior these users exhibit. By reviewing related literature from computer science, security, criminology, among other fields [25, 47, 112, 120, 128, 131], we identified five general characteristics of malicious activities. Malicious users can exhibit one (or a combination) of these characteristics in their activities. Note that as more characteristics of malicious activities are identified by researchers on social media, our methodology can be extended with these characteristics and the corresponding features that can capture the information redundancies introduced by them.

Malicious Activity is Complex and Diverse

Malicious users often generate *complex* and *diverse* information to ensure their anonymity. To measure complexity of usernames, it is natural to borrow techniques from complexity theory. We employ Kolmogorov complexity to determine the complexity of a username. Kolmogorov complexity was proposed in 1965 by Andrey N. Kolmogorov to determine the randomness of strings in a concrete mathematical form.

Let x represent a string. We denote the Kolmogorov complexity of string x as $K(x)$. $K(x)$ is defined as the length of the shortest program capable of reproducing string x on a universal computer such as a Turing Machine. Hence, Kolmogorov complexity is the absolute minimum information required to reproduce x on the Turing machine. While Kolmogorov complexity defines the complexity (or information) available in a string, it is well-known that its exact value cannot be computed [70]. Having said that, the following theorem provides the means to compute the expected Kolmogorov complexity for a distribution of strings P:

Theorem 9. *(from [79]) The value of the [Shannon] entropy $H(P)$ for distribution P equals the expected value of the Kolmogorov complexity $E_x(K(x))$ on P, plus a constant term that only depends on P.*

Hence, by computing the entropy of the username distribution, one can approximate the expected Kolmogorov complexity of the distribution. However, the theorem discusses the entropy of a username distribution and it is not clear how one can connect this theorem to properties of a specific username. For connecting the properties of specific usernames to the entropy of the distribution, we can employ the concept of *information surprise* [33].

Let x denote a username and $p(x)$ denote the probability of observing x. We

denote information surprise, or self-information, for x as

$$I(x) = -\log_2(p(x)). \tag{4.10}$$

Hence, for a rare username x with a small observation probability $p(x)$, information surprise $I(x)$ is much higher than that of a common username with a higher probability of observance. It is well-known that information surprise is deeply connected to entropy:

Theorem 10. *(from [33]) The expected value of information surprise $E(I(X))$ for a random variable X is equivalent to its entropy $H(X)$.*

So, by combining Theorems 1 and 2, one can approximate the expected Kolmogorov complexity of usernames by computing the expected information surprise in them. The information surprise for a username x is computed by measuring $I(x) = -\log_2(p(x))$, which requires the probability of observing username x. The probability of observing username `x`, denoted in characters as $x = c_1c_2 \ldots c_n$, is

$$p(x) = \Pi_{i=1}^{n} p(c_i|c_1c_2 \ldots c_{i-1}). \tag{4.11}$$

We approximate this probability using an n-gram model,

$$p(x) \approx \Pi_{i=1}^{n} p(c_i|c_{i-(n-1)} \ldots c_{i-1}). \tag{4.12}$$

Often, to denote the beginning and the end of a word special symbols are added such as $\star$ and $\bullet$. So, for username `sara`, the probability approximated using a 2-gram model is

$$p(sara) \approx p(s|\star)p(a|s)p(r|a)p(a|r)p(\bullet|a). \tag{4.13}$$

To estimate the probability of a username using an n-gram model, one needs to compute the probability of its comprising n-grams. The probability of these n-grams

can be computed using a large set of usernames. For that, we use a dataset of 158 million Facebook usernames (later discussed in Section 4.2.4) to train a 6-gram model. This n-gram model was employed to compute the probability of a username and in turn, its information surprise.

Figure 4.5 plots the empirical probability density function (Kaplan-Meier estimate) of information surprise values for normal and malicious users. The process followed to collect these usernames is later discussed in Section 4.2.4.

The black solid line in Figure 4.5 demonstrates the distribution of surprise values for normal usernames and the black dashed line depicts the distribution for malicious usernames. As shown in the figure, malicious usernames are more complex with the expected information surprise (i.e., expected Kolmogorov complexity) value of 23.11 bits and more diverse, ranging from 4.14 bits to 232.64 bits.

Unlike malicious usernames, normal usernames are less surprising and more concentrated around a mean value, with a mean of 12.49 bits and the information surprise value ranging from 3.90 to 31.93 bits. The figure shows that these distributions are well separated indicating that by using the information surprise of a username, one might be able to accurately classify usernames into malicious or normal.

In Figure 4.5, the gray line depicts the curve for the malicious usernames subtracted by the curve for the the normal usernames. Hence, when this gray line is above zero, it shows that for a specific information surprise value, the username is more likely to be malicious and whenever the gray line is below zero, we observe the opposite. We notice that for values between 3.91 and 17.96 the curve is below the zero line, showing usernames are more likely to be normal. In this range, the mean value is 10.9 bits. Thus, when the information surprise for a username is approximately 10 bits, the username is more likely to be normal.

We include the information surprise of the username (i.e., its complexity) as an-

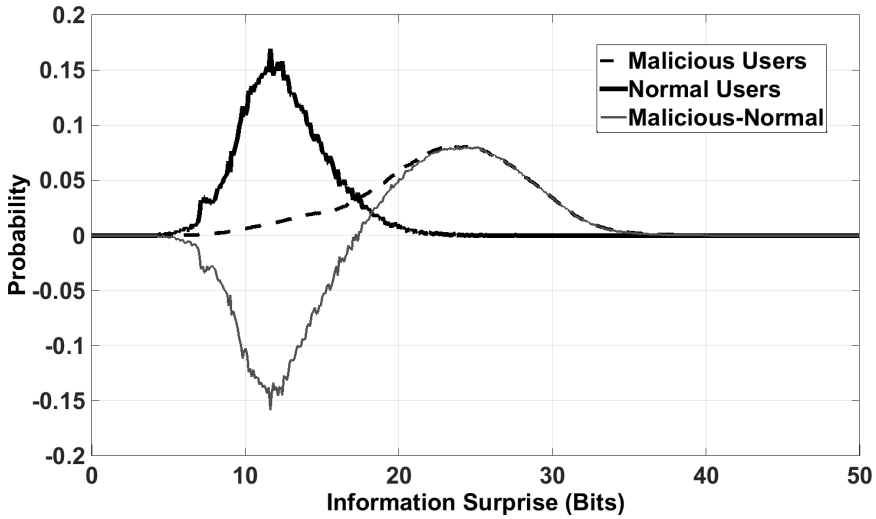


Figure 4.5: Probability Density Function for Information Surprise Values of Malicious and Normal Users.

other feature in our dataset. In addition, a common pattern for malicious users for providing diverse information is to generate usernames that include digits. Therefore, we include the number of digits in the username as a feature. We also include the proportion of digits in the username as another feature in our feature set.

Malicious Activity is Demographically Biased

The malicious activity is the act of a malicious user. In the criminology literature [47], it is well-known that crime correlates with demographic information. Thus, one expects to better detect malicious users by determining their demographics. Following the *diffusion of innovations* terminology [87], a malicious user has internal demographic attributes, external demographic attributes, or a combination of internal and external (i.e., mixed) attributes.

Internal attributes are endogenous attributes that the user has no control over such as his or her age. External attributes are attributes due to the environment that the malicious user lives in such as the language that the malicious user speaks. The level of knowledge that the malicious user has is an example of a user attribute that is

mixed (internal+external). This is because it depends on both the environment that the malicious user lives in and on the internal attributes of a user such as his interests. To concretely profile a malicious user, one has to consider all these attributes. We select gender from internal attributes, language from external attributes, and knowledge (i.e., vocabulary size) from mixed demographic attributes to be predicted from usernames. Clearly, with more internal/external/mixed demographic attributes, one should better profile malicious users. We leave that as a future direction for this work. But, how can we detect gender, language, or other attributes of individuals from their usernames?

Psychological studies [56] show that users leave traces of their personal information and attributes in the information they generate such as their usernames. For example, we showed in Section 4.1.3 that personal information such as first name influences usernames. Hence, given a name, one can estimate the most likely age. Names, interests, as well as other personal attributes are often abbreviated or used in usernames [139]. We use these information traces in usernames to predict gender, language, among other attributes.

I. Malicious User Gender. To predict gender from usernames, we train a classifier. The classifier decomposes a username into character n-grams and estimates the gender likelihood based on these n-grams. This classifier is trained on the n-grams of a labeled dataset of usernames, in which the gender for each username is known. We collect our labeled dataset from Facebook. Our labeled dataset contains a set of 4 millions usernames with their corresponding gender. The classifier predicts the gender of a username with up to 80% accuracy. Notice that because malicious users tend to hide their identity and gender; instead of the actual prediction, we include the classifier's confidence in the predicted gender as the feature.

II. Malicious User Language. To detect the language of the username, we train an n-gram statistical language detector [44] over the European Parliament Proceedings Parallel Corpus[5], which consists of text in 21 European languages (*Bulgarian, Czech, Danish, German, Greek, English, Spanish, Estonian, Finnish, French, Hungarian, Italian, Lithuanian, Latvian, Dutch, Polish, Portuguese, Romanian, Slovak, Slovene,* and *Swedish*) from 1996-2006 with more than 40 million words per language. The trained model detects the username's language, which is a feature in our feature set. The *detected language* feature is limited to European languages. Our language detector will not detect other languages. The language detector is also challenged when dealing with words that may not follow the statistical patterns of a language, such as location names, etc. This issue can be tackled by including the distribution of alphabet letters in usernames as features [139]. Thus, in addition to predicted language, we include the alphabet distribution of the username as a feature.

III. Malicious User Knowledge. To approximate the level of knowledge of a malicious user, we can compute his or her vocabulary size. The vocabulary size can be computed by counting the number of words in a large dictionary that are substrings of the username [139]. This approach captures different possible interpretations of the username and approximates the level of knowledge of the malicious user. We include the vocabulary size as a feature.

Malicious Activity is Anonymous

Malicious activity often requires a level of anonymity [18]. Theoretically, the maximum level of anonymity can be achieved with a string that has the maximum entropy [129]. We compute the entropy of the alphabet distribution of the username

[5]http://www.statmt.org/europarl/

as well as its normalized entropy to measure its level of anonymity. To normalize entropy, we divide it by $\log n$, where n is the number of unique alphabet letters used in the username. Moreover, we measure the uniqueness of letters in the username – that is, the number of unique letters used in the username divided by the username length. We include entropy, normalized entropy, and uniqueness as features.

Malicious Activities are Similar

Malicious activities can be similar. For instance, individuals marketing an illegal product `Dangerous-Pill` all share the name of the product `Dangerous-Pill` in the marketing content. This malicious content similarity can be captured in usernames by identifying specific (1) language patterns and (2) words in the usernames.

Language Patterns

To find finer grain language patterns of users, we employ character-level n-grams. Character-level n-grams have shown to be effective in detecting unwanted content [67, 68] and connecting users across social media sites [106]. We compute the normalized character-level bigrams of usernames and include them as features. Bigram features are normalized using TF-IDF. Bigrams allow for a language-agnostic solution [139] that can detect common patterns of malicious users conveniently.

For coarser grain language patterns, we investigate common habits of malicious users. For instance, it is known that the use of digits is an indication of unwanted content [75]. In particular, we notice that malicious users tend to start their usernames with digits; therefore, we include the number of digits at the beginning of the username as a feature. We also notice that malicious users repeat character letters more often that normal users. This strategy allows them to circumvent widely used statistical malware blockers [127]. Hence, we include the maximum number of times a letter has been repeated in the username as another feature.

Word Patterns

A well-known approach to identify malicious users or content is by finding specific keywords in the content generated by these users. Hence, we denote the existence of these specific keywords in usernames as an indication of malicious activity. We utilize two dictionaries, one containing keywords related to malicious activities and the other for offensive keywords[6]. For each dictionary, we count the number of words in the dictionary that appear as the substring of the username. We include these two counts for the aforementioned two dictionaries as features.

Malicious Activity is Efficient

In contrast with complex malicious activities (Section 4.2.3), some malicious activities demand efficiency. This is because the malicious user is interested in performing the malicious activity frequently, quickly, and at large-scale. For instance, when performing click-fraud, the malicious user is interested in creating many accounts, each clicking on specific ads. This efficiency can be observed in usernames in terms of (1) the username length; and (2) the number of unique alphabet letters in usernames. We include both as features. In addition, we can observe efficiency by determining the typing patterns of the malicious user.

Most people use one of the two well-known DVORAK and QWERTY keyboards, or slight variants such as QWERTZ or AZERTY [125]. It has been shown that the keyboard layout significantly impacts how random usernames are selected [42]. For example, `qwer1234` and `aoeusnth` are two well-known passwords commonly selected by QWERTY and DVORAK users, respectively. To model typing patterns of malicious users, for each username we construct the following 15 features for each keyboard

[6]Available at http://www.cs.cmu.edu/∼biglou/resources/

layout (a total of 30 for both keyboard layouts),

1. (1 feature) The percentage of keys typed using the *same hand* that was used for the previous key. The higher this percentage the less users had to change hands for typing.

2. (1 feature) The percentage of keys typed using the *same finger* that was used for the previous key.

3. (8 features) The percentage of keys typed using each finger. Thumbs are not included.

4. (4 features) The percentage of keys pressed on rows: Top Row, Home Row, Bottom Row, and Number Row. Space bar is not included.

5. (1 feature) The approximate *distance* (in meters) traveled for typing a username. Normal typing keys are assumed to be $(1.8\text{cm})^2$ (including gap between keys).

We construct $15 \times 2 = 30$ features that capture the typing patterns of usernames for both keyboards and include them in our feature set.

We have detailed how characteristics of malicious activities can be captured by meaningful features. These features help identify traces of malicious activities in usernames. Overall, for each username, we construct 1,413 features.

Clearly, not all aspects of malicious activities are covered by our features, and with more theories on characteristics of malicious activity, more features can be constructed. We will empirically study if it is necessary to use all features and the effect of using different features on learning performance of detecting malicious users.

Following our approach, we compute the feature values over labeled data, and verify the effectiveness of our methodology by learning the malicious user detection function. Next, experiments for evaluating our methodology are detailed.

4.2.4 Experiments

We evaluate our methodology to detect malicious users in this section. First, we verify if our proposed approach can identify malicious users well. Next, we verify if different learning algorithms can influence the prediction task. Then, we determine the sensitivity of our approach to different conditions. Finally, we perform feature importance analysis and determine how features designed for each characteristic of malicious activity influence the detection outcome. Before we present the experiment details, we detail how experimental data was collected for this research.

Data Preparation

Our approach to detect malicious users employs a supervised learning framework. Hence, labeled data is required. This labeled data consists of usernames and their corresponding label: malicious or normal.

To collect malicious usernames, we refer to sites such as dronebl.org, ahbl.org, among others (for a complete list see [90]). These sites gather lists of usernames that have been reported by other normal users for malicious purposes. Once reported, these accounts are manually verified by domain owners to be malicious. These lists are published to help sites promote their security. We collect a set of 32 million usernames that are manually reported as malicious by users across the web and for different types of sites. This set forms our negative examples.

For collecting normal users, we require users that are manually labeled as normal. For that, we refer to Twitter verified accounts, all manually verified by Twitter employees. These accounts are all followed by the Twitter handle `verified`[7]. By crawling all the users this account follows, we collect a set of 45,953 usernames guar-

[7]http://twitter.com/verified

anteed to be normal. These usernames form our positive examples. To diversify the types of usernames we have collected, we also collect a set of 158 million usernames from Facebook, that is, 1 in 8 Facebook users in the world are included in our dataset. Note that the Facebook dataset is not completely normal as Facebook expects around 1.5% to be malicious. We employ this dataset later in our experiments for analyzing the sensitivity of our approach to different conditions.

In addition, we collect a different set of 4 million Facebook users for which we have the gender information. This dataset was used in our gender prediction classifier in Section 4.2.3 to predict gender of the Facebook users.

After collecting positive and negative usernames[8], we compute the corresponding 1,413 features for both sets and employ them in our experiments.

Learning the Malicious User Detection Function

Once the negative and positive examples are prepared, learning the malicious user detection function can be achieved by training a classifier. Because our collected negative examples are more, we subsample the negative examples to have the same size as the positive examples. This way we create a dataset that has 50% positive examples and 50% negative ones. Using this dataset, we train a classifier. The random prediction on this dataset cannot achieve more than 50% accuracy. We train an ℓ_2-Regularized Logistic Regression using 10-fold cross validation and obtain an accuracy of 96.42%, an AUC of 0.9932, and an F1-measure of 0.9644.

As there are no comparable methods, we evaluate the effectiveness of our approach by devising three baseline methods for comparison. When individuals are asked to

[8]We ensure that the alphabet used in both sets of usernames match. To avoid site-enforced specific patterns on how usernames should be created, we filter out usernames that are not in ASCII or alphanumeric. Our experiments show that this procedure does not influence our results.

detect malicious users based on their usernames, they often look for specific "keywords", verify if the username looks "random", or look for "repetition of letters". Hence, they form our three baselines b_1, b_2, and b_3:

- **Baseline b_1: Keyword Detection.** We consider a username malicious if it contains a specific keyword. We use the same set of keywords used in Section 4.2.3 and train a classifier based on the single feature. b_1 results in an AUC of 0.5140 and F1-measure of 0.66.

- **Baseline b_2: Username Randomness.** For finding username randomness, b_2 uses the entropy of the username as a feature. Using our data labels, we perform logistic regression. b_2 achieves an AUC of 0.700 and F1-measure of ≈ 0.

- **Baseline b_3: Letter Repetition.** Similar to the procedure followed in baseline b_2, in b_3, we use the maximum number of times a letter is repeated in the username as a feature and train a logistic regression model using our data labels. b_3 achieves an AUC of 0.61 and an F1-measure of ≈ 0.

While the baseline performances demonstrate the difficulty of our problem, the proposed approach outperforms all baselines by at least 41%. The performance for our approach, and baselines are summarized in Table 4.2. As reference points, we also include in the table the performance of recent state-of-the-art techniques for detecting malicious users. These techniques have access to more information compared to our methodology and do not employ usernames; therefore, no improvement percentage will be reported. Our approach, with usernames only, outperforms these techniques. Next, we investigate if different learning algorithms can further improve the learning performance.

Table 4.2: Malicious User Detection Performance

Technique	**AUC**	**F1**
Our Approach	0.9932	0.9644
Baseline b_1: Keyword Detection	0.51	0.66
Baseline b_2: Username Randomness	0.70	≈ 0
Baseline b_3: Letter Repetition	0.61	≈ 0
Reference Point r_1: *Markines et al.* [89]	0.984	0.983
Reference Point r_2: *Gao et al.* [52]	0.945	N/A
Reference Point r_3: *Wang* [122]	0.917	0.917

Choice of Learning Algorithm

To evaluate the choice of learning algorithm, we perform the classification task using a range of learning algorithms and 10-fold cross validation. The AUCs and accuracy rates are available in Table 4.3. These algorithms have different learning biases, and one expects to observe different performances for the same task. While we observe a slight increase in the performance, as shown in the table, results are not significantly different across algorithms. This shows that when sufficient information is available in features, the performance is not sensitive to the choice of learning algorithm.

In our experiments, ℓ_1-Regularized Logistic Regression is shown to be the most accurate method; therefore, we use it in the following experiments as the method of choice.

In our previous experiments, we assumed that there is no class imbalance between malicious and normal users. In reality this distribution is skewed. Furthermore, because all of our normal users are from one source (Twitter verified accounts), one needs to verify the effect that this has on our method. We analyze the sensitivity of

Table 4.3: Malicious User Detection Performance for Different Classification Techniques

Technique	AUC	Accuracy
ℓ_2-Regularized ℓ_1-Loss SVM	0.9966	97.05%
ℓ_2-Regularized ℓ_2-Loss SVM	0.9913	96.05%
ℓ_2-Regularized Logistic Regression	0.9923	96.25%
ℓ_1-Regularized Logistic Regression	0.9971	**97.26**%

our approach to the class imbalance and the distribution of normal users next.

Sensitivity Analysis

Sensitivity to Class Imbalance

In real-world networks such as Facebook and Twitter, the percentage of malicious users in the population is approximated to be at most 10% [46, 123]. In other words, for every 9 normal users there exists at most 1 malicious user. This rate could be different across networks. Thus, we perform a sensitivity analysis with respect to different ratios of malicious users. We construct datasets, where α percent of the dataset consists of malicious users and change α in the range $5 \leq \alpha \leq 50$. Values larger than 50 were not selected, because then we are assuming that malicious users are more than the normal ones.

Because we collected more negative examples, we sample the negative examples many times to guarantee that each negative example is seen at least once. Thus, for each α, many datasets are created. For each one of these datasets, we perform classification and average the performance metrics over all datasets created for a

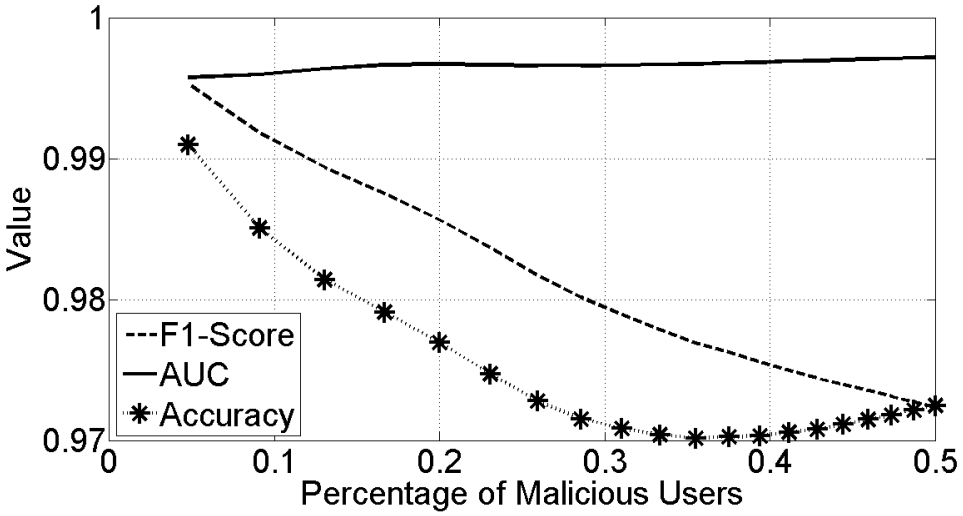


Figure 4.6: Performance (AUC, F1, and Accuracy) of our Methodology for Different Percentages of Malicious Users.

specific α.

Figure 4.6 depicts the average performance (accuracy, AUC, and F1-measure) of our methodology with different percentages of malicious users. As shown in the Figure, as the number of malicious users increase, AUC remains stable and F1-measure and accuracy slightly drop, but in all cases, all measures stay above 0.97.

Sensitivity to the Distribution of Normal Users

To verify the sensitivity of our classifier to the distribution of normal users, we use an equally-sized sample of Facebook users instead of our normal users. Note that unlike our original positive instances, Facebook approximates that around 1.5% of its user population are malicious users [123]. Thus, if our algorithm is capable of detecting these users, then its performance using the sampled Facebook dataset is expected to slightly decrease. Thus, for all datasets that have at most 50% negative examples (malicious users), one expects at most a decrease of $50\% \times 1.5\% = 0.0075$ in accuracy. Our experiments verify this expected outcome. We notice a slight drop in performance for all measures, but the performance remains high for different

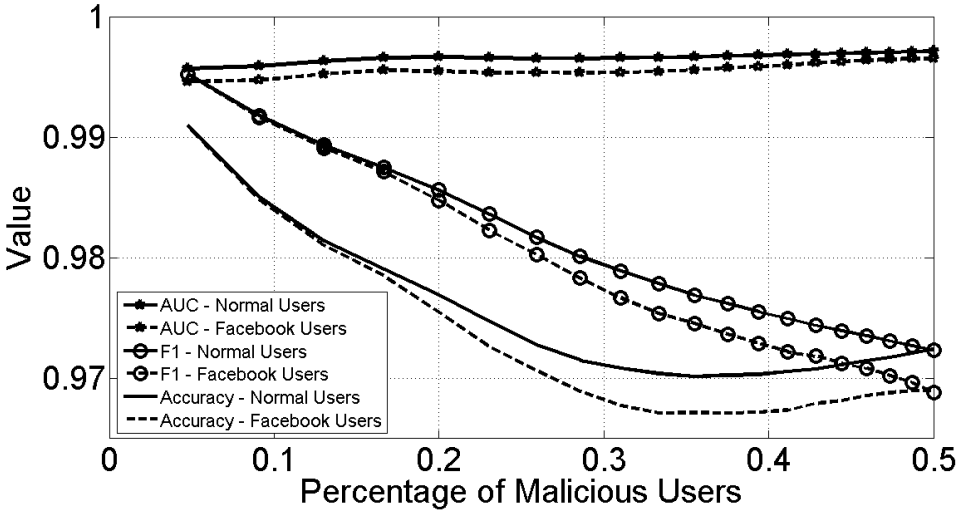


Figure 4.7: Performance Measures (F1, AUC, and Accuracy) of our Methodology for Different Percentages of Malicious Users when Facebook Identities were used instead of Normal Users.

percentages of malicious users and never drops below 0.9671. Figure 4.7 depicts the performance (accuracy, AUC, F1-measure) of the algorithm with different percentages of malicious users and using Facebook users as positive examples. For comparison, we include the performance measures for normal users. Comparing the performance measures with those of normal users, we notice that the accuracy drops by at most 0.0035 (less than the expected maximum: 0.0075), AUC drops by at most 0.0012, and F1-measure drops by at most 0.0036.

In our experiments, we employ all 1,413 features to detect malicious users. Designing 1,413 features and computing their values is computationally expensive. Hence, we empirically determine whether all features are necessary next.

Feature Importance Analysis

In this section, we analyze how important different features are in learning the detection function. In other words, we find features that contribute the most to the classification task. This can be performed by standard feature selection measures

such as Information Gain, χ^2, among others. Here, we use the χ^2 statistic to find the top features. The top 10 features in decreasing order of importance are:

1. The information surprise of the username
2. The number of digits used in the username.
3. The percentage of keys pressed on the top row of a QWERTY keyboard when typing the username.
4. The percentage of keys pressed on the top row of a DVORAK keyboard when typing the username.
5. The proportion of digits used in the username.
6. The approximate distance (in meters) traveled for typing a username with a DVORAK keyboard.
7. The percentage of keys pressed on the home row of QWERTY keyboard when typing the username.
8. The approximate distance (in meters) traveled for typing a username with a QWERTY keyboard.
9. The percentage of keys pressed on the bottom row of a DVORAK keyboard when typing the username.
10. Entropy of the username.

We notice that the complexity of the username is the most important feature and that 6 of the top 10 features are features that capture typing patterns. Using only these 10 features, we trained a logistic regression model and achieved an accuracy of 92.95% and an AUC of 0.973.

Table 4.4: Malicious User Detection Performance for Different Groups of Features

Feature Groups	**AUC**	**Accuracy**
Complexity-based	0.8032	83.16%
Demographic-based	0.9342	86.78%
Anonymity-based	0.7219	63.26%
Similarity-based	**0.9933**	**95.86%**
Efficiency-based	0.9299	87.19%

We also determine groups of features that contribute most to the classification. We divide features into groups based on the malicious activity characteristic they represent. We denote these features based on the discussion in Section 4.2.3 as (1) Complexity-based, (2) Demographic-based, (3) Anonymity-based, (4) Similarity-based, and (5) Efficiency-based. Table 4.4 summarizes the classification performance obtained using only these groups of features.

We observe that similarity-based features work the best and anonymity-based features are least effective. Note that similarity-based features are in general hard to construct as they require n-gram constructions. Surprisingly, efficiency-based or complexity-based features that are easier to compute, can classify malicious users accurately, with up to 87% accuracy. Our observations in this section allows users with limited time and resources to take informed decisions on the features and groups of features to construct.

4.3 Summary

In this chapter, we proposed two applications for finding friends and detecting malicious users with minimum information.

Our approach for finding friends is applicable when link or content information

is unavailable. This problem exists in all social media sites and for all new users, as they have no friends or have not generated any content. Under these constraints, sites are often forced to recommend randomly chosen influential users, hoping that users by befriending some, provide sufficient information for link prediction techniques for further recommendations.

Friendships in social media are often formed due to three social forces: homophily, confounding, and influence. We show how minimum content information available on all social media sites (usernames) can be employed to determine friendships due to these forces. In particular, we employed usernames to predict personal attributes such as age, location, and language that in turn can be used to find friends and measure the effect of each social force. Our empirical results show the advantages of this principled approach by improving friend finding performance by an expected factor of 5.49-31.04 over random prediction. This is comparable to the state of the art link-prediction techniques that perform 2.4-54.4 times better than random prediction [80]. Our results also show that while by employing each social force, one can improve friend finding performance at least by a factor of 5.49, influence can help best find friends. This suggests that individuals have a higher tendency to befriends others with similar friends (influence), than those who are more similar to them (homophily) or share an environment (confounding). Our results show an improvement of, at least 6 times, over random predictions when link or content information is unavailable. Note that using our method personalized recommendations are performed since for example, users identified as French are more likely to be recommended French users.

In the second section of this chapter, we have introduced a methodology that can identify malicious users with minimum information. Our methodology looks into different characteristics of malicious activities and systematically constructs features that can capture traces of malicious behaviors. With new theories on characteristics

of malicious activities, new features can be introduced into our methodology.

We categorize characteristics of malicious activities into 5 general categories. In particular, malicious activities can be (1) complex and diverse, (2) demographically biased, (3) anonymous, (4) self-similar, and (5) efficient. A malicious activity can exhibit one or a combination of these characteristics. By introducing comprehensive features across these five categories, we train a learning framework that can detect malicious users. The evaluation of this framework demonstrates the effectiveness of this systematic approach.

We notice some interesting observations. First, we notice that usernames that carry approximately 10 bits of information surprise are more likely owned by normal users. Second, with only minimum information, one can achieve an accuracy of 97%, an AUC of 0.9971, and robust performances with different class imbalances and irrespective of the learning algorithm. Finally, we identify that in case of limited time or resources, one can implement a limited set of features and obtain reasonable accuracy rates.

Our findings in the second section have many implications. First, we note that our methodology is in general easy to implement with minimum dependency on the availability of information. Second, our methodology works with usernames from different sites. This is empirically shown in our experiments with usernames collected from a variety of sites. Finally, our methodology performs with reasonable accuracy, compared to state-of-the-art techniques that have access to additional information.

Chapter 5

DISTRIBUTION AND PATTERNS ACROSS SITES

Nobody comes here anymore,
it's too crowded.

Yogi Berra

Our life in social media is no longer limited to a single site. We post on Reddit, like on Facebook, tweet on Twitter, watch on YouTube, listen on Pandora, along with many other activities exhibited by social media users. This chapter is the first of three chapters that discuss user behavior across sites. In this chapter, we will discuss how users are distributed across sites and their joining patterns across sites. Next two chapters will discuss how user behavior changes across sites and specific behaviors that are only observed across sites.

With the constant rise of new sites and advancement of communication technology, thousands of social media sites are at our fingertips. With so many choices, our attention spans are decreasing rapidly. On average, a user spends less than a minute on an average site [1]. With our limited time and short attention span, we often face a dilemma of choosing a handful of sites over others. How do we select these sites?

As social media consumers, we are constantly seeking sites that can keep our attentions glued to our screens by providing engaging content, especially content generated by our friends. It is well-known that the likelihood of engaging in an activity is increased as more friends become engaged in that activity [16]. Thus, it is natural to assume that users select sites where they find more friends on. On

The content in this chapter has been published at ICWSM 2014 [141].

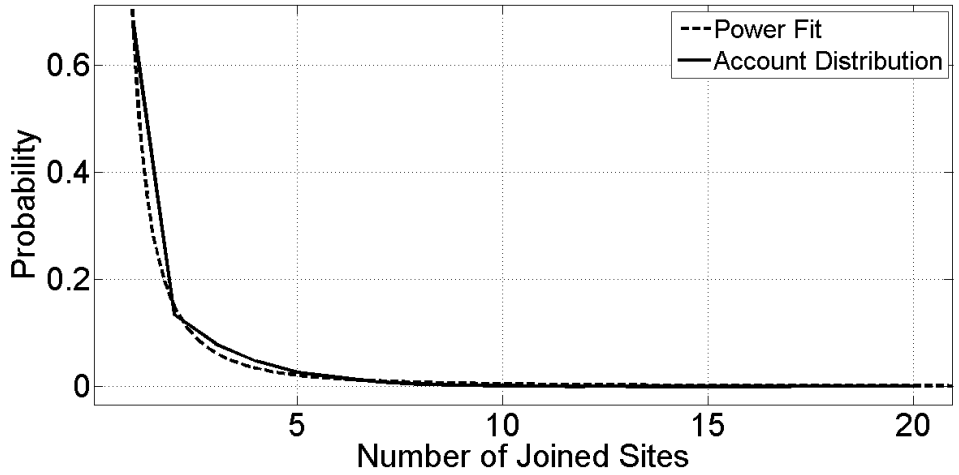
average, sites with more members are expected to contain more friends for an average individual; hence, it is expected for the users' site selection to be statistically biased toward more popular sites.

In this chapter, we analyze users joining multiple sites. We show how users are dispersed across sites. By studying users across sites, we show that while there is a tendency to join popular sites, users exhibit a variety of site selection patterns. Finally, we evaluate the obtained users' site selection patterns with an application that recommends new sites to users for joining. Our evaluation demonstrates promising results and reveals additional interesting user joining patterns.

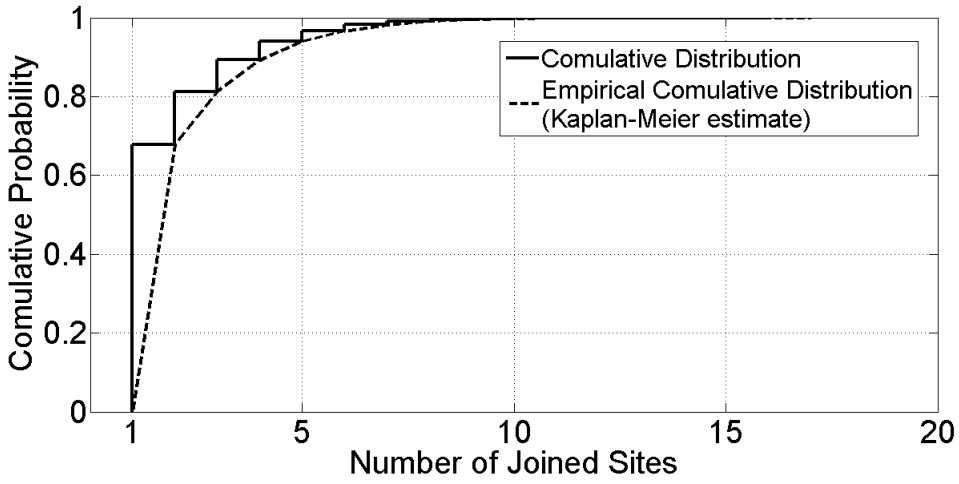
We first detail the data collection for our research. Next, we analyze user distribution across sites. Then, we outline membership patterns across sites, followed by our evaluation of these patterns. Finally, we conclude this chapter with a brief literature review and summary.

5.1 Data Preparation

To study user memberships across sites, one needs to gather sites that users have joined on social media. Unfortunately, this information is not readily available. One can simply survey individuals and ask for the list of sites they have joined. This approach can be expensive and the data collected is often limited. Another method for identifying sites that users have joined is to find users manually across sites. Users, more often than not provide personal information such as their real names, E-mail addresses, location, gender, profile photos, and age on these websites. This information can be employed to find the same individual on different sites. However, finding users manually on sites can be challenging and time consuming. Automatic approaches are also possible that can connect corresponding users across different sites using minimum information such as their usernames [139]. A more straightforward



(a) Probability Distribution



(b) Cumulative Probability Distribution and Empirical Cumulative Distribution

Figure 5.1: Distribution of Users across Sites

approach is to use websites where users have the opportunity to list the sites they have joined. In particular, we find social networking sites, blogging and blog advertisement portals, and forums to be valuable sources for collecting the sites users have joined. For example, on most social networking sites such as Google+ or Facebook, users can list their IDs on other sites. Similarly, on blogging portals and forums, users are often provided with a feature that allows users to list their usernames in other social media sites.

We utilized these sources for collecting sites that users have joined. Overall, we collected a set of 96,194 users, each having accounts on a subset of 20 social media sites. The sites included in our dataset are *BlogCatalog*, *BrightKite*, *Del.icio.us*, *Digg*, *Flickr*, *iLike*, *IntenseDebate*, *Jaiku*, *Last.fm*, *LinkedIn*, *Mixx*, *MySpace*, *MyBlogLog*, *Pandora*, *Sphinn*, *StumbleUpon*, *Twitter*, *Yelp*, *YouTube*, and *Vimeo*. The data was collected in 2008. In 2008, MySpace was the most important social networking site, BlogCatalog was one of the most popular blogging sites with social networking capabilities, and LinkedIn and Yelp were quite unpopular. At the time, Yelp had only 3 million users and LinkedIn was an order of magnitude smaller.

5.2 User Membership Distribution across Sites

First, we determine how users are distributed across sites. A natural way to determine the user distribution is to compute the proportion of users that have joined different number of sites. Figure 5.1(a) shows how users are distributed with respect to the number of sites they have joined. Figure 5.1(b) plots the cumulative distribution function and the empirical cumulative distribution function (Kaplan-Meier estimate) for the distribution in Figure 5.1(a). These figures show that more than 97% of users have joined at most 5 sites and users exist on as many as 16 sites.

A power function, $g(x) = 0.6761x^{-2.157}$, found with 95% confidence, fits to the distribution curve in Figure 5.1(a) with adjusted $R^2 = 0.9978$. The exponent -2.157 denotes that individuals that are members of n sites are $1/n^{2.157}$ less likely than individuals that are members of only one site. For example, users that are members of $n = 7$ sites are $\approx 1/66$ times less likely than users that are members of only one site. The power function fit is highly correlated to our data, indicating the possibility of a power-law distribution. To investigate this possibility, we follow the systematic procedure outlined in [30] to determine whether the user distribution across sites

follows a power-law distribution. For integer values, the power-law distribution is defined as

$$p(x) = \frac{x^{-\alpha}}{\zeta(\alpha, x_{min})}, \tag{5.1}$$

where, $\zeta(\alpha, x_{min}) = \sum_{n=0}^{\infty}(n + x_{min})^{-\alpha}$ is the generalized Hurwitz zeta function, α is the power-law exponent and x_{min} is the minimum value for which for all $x \geq x_{min}$, the power-law distribution holds. We estimate α and x_{min} using the maximum likelihood method outlined in [30]. Our results shows that the value of α is slightly larger than the initially obtained exponent of 2.157 and is around 2.34. To verify the validity of our power-law fit, we calculate p-value using the Kolmogorov-Smirnov goodness-of-fit test. We obtain $p \approx 0$, rejecting the null hypothesis, showing that users across sites are distributed according to a power-law distribution.

5.3 User Membership Patterns across Sites

We showed that user distribution across sites is power-law. However, it is still unknown how users select sites to join. A common perception is that users are more likely to join most popular sites. Here, we show that this is not true in general. While there is a tendency to join popular sites, users exhibit different site selection patterns on social media.

Assume that sites are represented using a *complete* weighted graph $G(V, E, O)$. In this graph, nodes $v \in V$ represent sites. Let $|V| = n$. In our data, $n = 20$. An edge exists between all pairs of nodes, i.e., $E = V \times V$. Edge $e_{ij} \in E$ between two sites (nodes) i and j has weight $O_{ij} \in O$, where $O \in \mathbb{R}^{n \times n}$. Weight O_{ij} denotes the number of users that are members of both sites i and j. Let $O_{ii} = 0$.

Our collected dataset can be represented using a matrix $U \in \mathbb{R}^{l \times n}$, where l is the number of users. $U_{ij} = 1$, when user i is a member of site j and $U_{ij} = 0$, otherwise.

Clearly, O matrix can be written in terms of U matrix,

$$O = (J_n - I_n) \circ U^T U, \tag{5.2}$$

where $J_n \in \mathbb{R}^{n \times n}$ is the matrix of all ones, I_n is the identity of size n, and $\circ$ is the Hadamard (entrywise) product.

For site v, let d_v represent the number of users that are on site v.[1] We can estimate[2] d_v as $d_v \approx \sum_i O_{vi}$.

For two sites i and j, we compute the number of users that are expected to be members of both. Assume that users randomly join a site with a probability that is proportional to its popularity. For any user in site i, the probability that the user joins site j is $\frac{d_j}{\sum_k d_k} = \frac{d_j}{2m}$, where $m = \frac{1}{2}\sum_k d_k$. As site i has d_i users, the *expected* number of members of both sites is $\frac{d_i d_j}{2m}$. The actual number of members of both sites is given in our data as O_{ij}. The distance between this actual number and its expected value $(O_{ij} - \frac{d_i d_j}{2m})$ indicates how non-random joining both i and j is. We expect the users' site selection behavior to be non-random. Thus, we can find communities of sites such that this distance is maximized for the sites in each community. These communities represent sites that users often join together. Let $P = (P_1, P_2, \ldots, P_k)$ denote a partitioning of the sites in V into k partitions. For partition P_x, this distance can be defined as

$$\sum_{i,j \in P_x} (O_{ij} - \tfrac{d_i d_j}{2m}). \tag{5.3}$$

This distance can be generalized for the partitioning P,

$$\sum_{x=1}^{k} \sum_{i,j \in P_x} (O_{ij} - \tfrac{d_i d_j}{2m}). \tag{5.4}$$

[1]This is equivalent to a node's degree in an unweighted graph.

[2]The estimation performs well in our setting and is close to the actual d_v; however, it considers independence among site overlaps.

This summation term takes a maximum value of $\sum_{ij} O_{ij} \approx \sum_k d_k = 2m$; therefore, the normalized version of this distance is defined as

$$Q = \frac{1}{2m} \left[\sum_{x=1}^{k} \sum_{i,j \in P_x} (O_{ij} - \frac{d_i d_j}{2m}) \right]. \tag{5.5}$$

This is in fact a weighted version of the modularity measure defined by Newman [102]. We define the modularity matrix as $B = O - \mathbf{d}\mathbf{d}^T/2m$, where $\mathbf{d} \in \mathbb{R}^{n \times 1}$ is a vector that contains the number of members for all sites. Then, weighted modularity can be reformulated as

$$Q = \frac{1}{2m} \mathrm{Tr}(X^T B X), \tag{5.6}$$

where $X \in \mathbb{R}^{n \times k}$ is the partition membership matrix, i.e., $X_{ij} = 1$ iff. $v_i \in P_j$. This objective can be maximized such that the best membership function is obtained with respect to weighted modularity. Unfortunately, the problem is NP-Hard. Relaxing X to $\hat{X}$ that has an orthogonal structure ($\hat{X}^T \hat{X} = I_k$), the optimal $\hat{X}$ can be computed using the top k eigenvectors of B corresponding to positive eigenvalues.

Even when maximizing weighted modularity on our data, we obtain a negative value. The negative modularity denotes that users on average have other preferences when joining new sites than just selecting random popular sites.

Figure 5.2 shows the categorization of sites obtained using weighted modularity maximization. We observe several patterns in this figure. First, we notice that there are popular sites that users become members of all (or most). These sites are shown on the top right part of the figure in light orange. This cluster is MySpace, BlogCatalog, Twitter, and YouTube. For instance, we become members of Facebook to socialize with our friends, Twitter to post microblogging messages, YouTube to watch videos, and WordPress to write blogs. Back in 2008, MySpace and BlogCatalog were exemplars of prominent social networking and blogging sites. We believe this cluster of sites represent the average behavior of most users that are members of a few sites

to satisfy their basic needs. The second group of sites are shown in the bottom part of the figure using green and red nodes. Green nodes represent audio/video/photo sharing sites such as online radios or video sharing sites that consumers often join **all** to be able to access the content that becomes available on each one of them. Similarly, the red nodes represent social tagging/social news/content sharing sites where individuals visit **all** to obtain interesting content. Reddit is a current popular example of these sites. The final group of sites shown in Blue, are unknown or unpopular sites that users rarely join. These are sites that are often joined by early adopters who wish to explore more and find new content or sites. Note that Yelp and LinkedIn were members of this cluster in 2008, which is due to their less popularity at that time. Note that these patterns are based on sites that are joined together; therefore, they are not mutually exclusive. A user can join sites in one or all of these clusters. Furthermore, a user should not necessarily be a member of all sites in each cluster, but can be a member of a subset of the sites.

After user membership patterns are obtained, it is imperative to validate these patterns. Because ground truth of the patterns is unavailable, one way of evaluating is to check if the patterns can help in some applications such as prediction or recommendation. In the following, we adopt the recommendation task as an evaluation strategy. As we will see, this approach leads to the further discovery of interesting patterns on how users select sites to join.

5.3.1 Evaluating via Recommending Sites to Users

If site selection patterns are not true patterns (i.e., random patterns), one should not be able to observe their effect in recommending sites to users. By identifying the types of site selection patterns a user has exhibited in the past, one can recommend sites to the user in the future. By outperforming baseline methods that use no user

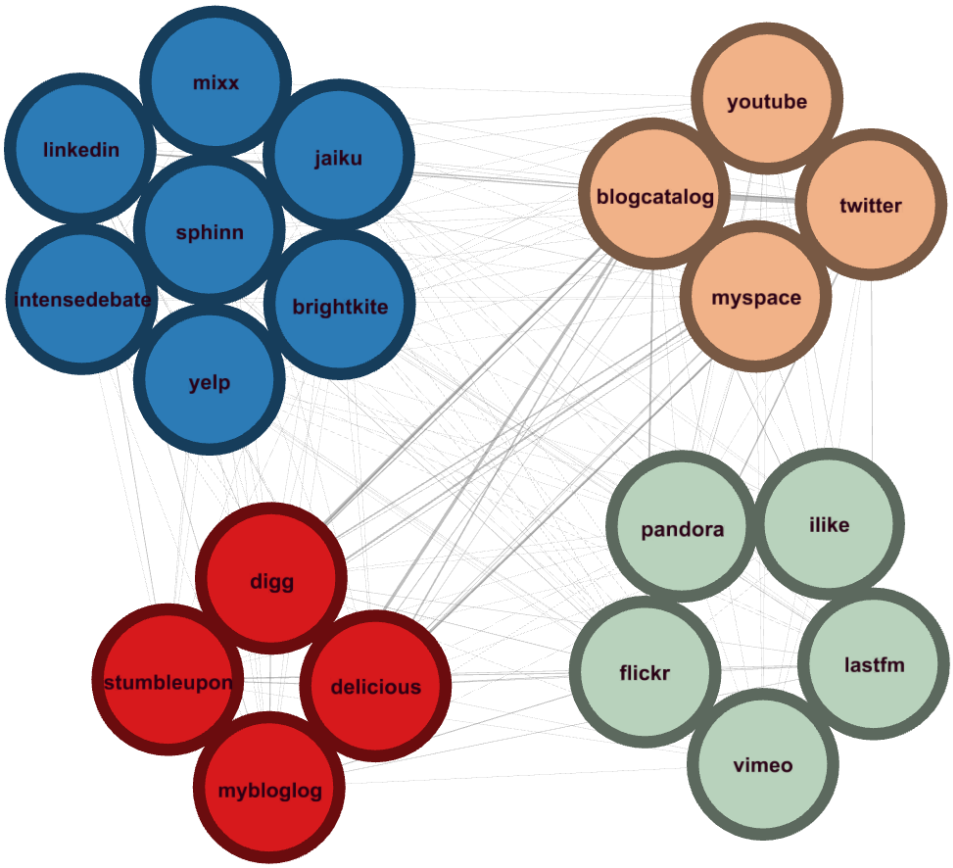


Figure 5.2: Site Categorization based on Sites that are Commonly Joined by Users.

patterns, once can safely conclude that the obtained patterns are true patterns.

For any user in our dataset that has joined n sites, we assume that given the category (node color in Figure 5.2) of $n-1$ of these sites, the category of the nth site should be predictable. We use categories instead of the sites as this introduces a generalizable recommendation algorithm as new sites appear on social media. Thus, for each user that has joined n sites, we generate all the $\binom{n}{n-1} = n$ combinations of $n-1$ sites as historical data. For each combination of $n-1$ sites, we construct a data instance of 4 features by counting the number of sites in each category that the user has joined in the past. This instance describes the amount of interest the user has expressed in each category in the past. We set the class label as the category of the nth site (i.e., a value in {1,2,3,4}). We generate 73,001 instances. Our initial attempt

Table 5.1: Site Recommendation Performance

Technique	**AUC**	**Accuracy**
J48 Decision Tree Learning	0.880	**79.25%**
Random Forest	0.895	79.17%
Logistic Regression	0.886	79.14%
SMO (Sequential Minimal Optimization)	0.728	78.92%
Naive Bayes	0.869	76.66%

to predict the class label in this dataset using Naive Bayes classifier recommends a new site with an accuracy of 76.66% and an AUC of 0.869. To determine the sensitivity of our results to the learning bias of different algorithms, we test a variety of classification techniques. The results are provided in Table 5.1. We observe minimal sensitivity to the learning bias. J48 performs the best with 79.25% accuracy in predicting the correct site category and an AUC of 0.88. Thus, J48 is used for the rest of our experiments.

To verify the influence of historical data on our results, we select 11 subsets of our dataset. Subset i, $0 \leq i \leq 10$ contains the set of users that have already joined i sites. We perform the same classification for each set. Figure 5.3 shows the prediction results for different number of already joined sites. The figure also shows as a dashed line the majority class predictor for each set. We observe from this figure that the performance is the highest (97%) when users haven't joined any sites, and is decreased as users join more sites until 4 sites are joined. After which the performance starts to increase as more information about the user joining patterns becomes available to the algorithm.

Although the classifier performs the best when users haven't joined any sites, however, at this point the majority class prediction performs almost as well. The

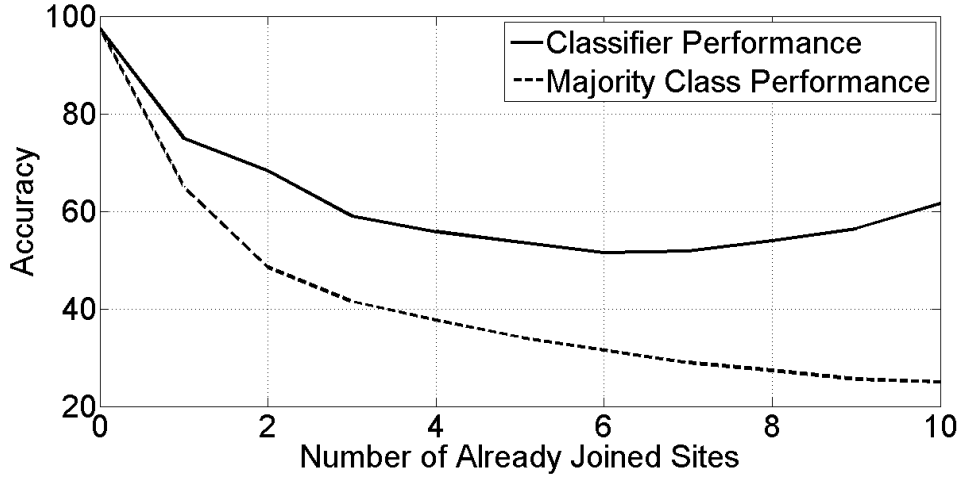


Figure 5.3: Recommendation Performance when the User has Already Joined some Sites.

majority class in this case is the class of most popular sites. In other words, when users haven't joined any sites, they often just select the most popular sites; therefore, recommending these sites is most successful. We notice that as users join more sites, the effect of majority is reduced and when users have already joined 10 sites, the majority prediction is no different from random prediction ($25\% = \frac{1}{4}$). In other words, as users join more sites, peer pressure of joining popular sites is reduced and preference plays an important role. In this case, while the majority fails at predicting more than 30% correctly, our recommendation can perform as accurate as 60%.

5.4 Related Work

Studying multiple networks has been the subject of a number of recent studies; see [21, 86] for two such studies. The focus of these studies has been on how network dynamics and user behavior changes across networks irrespective of the users that these networks share or how behavior changes across networks after users join, irrespective of how these users select the sites in the first place. The work in this chapter is different from these studies as it analyzes individuals that are shared across networks,

their distribution, and membership patterns.

5.5 Summary

We have studied the user membership behavior across social media sites. We showed that user distribution across sites is a power-law distribution with an exponent of $\alpha = 2.34$. Using a weighted modularity measure, we computed the categories of sites that users join together. We show that users join some sites due to their popularity (YouTube, Twitter, etc.). There are also sites that users join all due to media (online radios/audio sharing/video sharing) and content (Social tagging/social bookmarking/social news) consumption purposes. The last category of sites that users join are new or relatively unknown sites. These are joined by early adopters who wish to explore and find new content. To evaluate these site selection patterns, we designed a site recommendation algorithm for users. We showed that while for users that are members of no site, recommending popular sites performs the best, users that have joined a few sites are more likely to select sites based on their preference.

Chapter 6

VARIATIONS ACROSS SITES

You think because you understand 'one' you must also understand 'two', because one and one make two. But you must also understand 'and.'

Rumi

In chapter 4, we discussed how users are distributed across sites and different joining patterns that users exhibit across social media. In this chapter, we take a further step towards understanding users across sites by investigating whether information generated by the same user varies across sites. And if it does, how much does this information vary across sites. The answers to these question are critical for a systematic user study across sites. Our goal in this chapter is to tackle this question.

As friends are the fundamental building blocks of social media sites [138], we focus on how friends and friendship behavior varies across sites. Friendship behavior and friends are naturally connected to the concept of popularity. Often, an intuitive mechanism to achieve popularity is to befriend others. Friends introduce a more pleasant social media experience and having more friends is perceived as a sign of popularity. For example, on social media, some individuals befriend random individuals in order to increase their popularity. Hence, we extend our study by analyzing both friendship

The content in this chapter has been published in Information Fusion journal [144].

behavior and popularity variations across sites. We show how friends are dispersed across sites and how this distribution shifts as users join more sites. We show how joining more sites influences the number of friends individuals have across them, as well as their popularity. Finally, we demonstrate how the findings of this study can be used to predict the popularity of users on new sites.

We first discuss the social media sites that users join. Next, we analyze how friends are distributed across sites. Then, we study how popularity varies across sites and detail our approach to predict user popularity across sites. Finally, we review related research to this study and conclude this chapter with a summary.

6.1 Social Media Sites that Users Join

To understand user friendships and popularity across sites, one needs to gather the list of sites that users have joined on social media. Social media sites are developed for different purposes; therefore, to systematically study friendships and popularity, one has to consider different types of sites. According to recent studies [5, 69], sites in social media can be categorized into seven general categories: (1) *Blogs and Blogging Portals*, (2) *Media Sharing (Photo, Audio, or Video)*, (3) *Microblogging*, (4) *Social Bookmarking*, (5) *Social Friendship networks*, (6) *Social News and Search*, and (7) *Location-Based Networks*. We select 20 sites that cover these categories and are of different popularity on social media to study user friendships. Selected sites are *BlogCatalog*, *BrightKite*, *Del.icio.us*, *Digg*, *Flickr*, *iLike*, *IntenseDebate*, *Jaiku*, *Last.fm*, *LinkedIn*, *Mixx*, *MySpace*, *MyBlogLog*, *Pandora*, *Sphinn*, *StumbleUpon*, *Twitter*, *Yelp*, *YouTube*, and *Vimeo*. Next, we gather users that have joined some of these 20 sites.

Unfortunately, information about sites that users joined is not readily available. One can survey individuals and ask for the list of sites they have joined. This approach can be expensive and the data collected is often limited. Another method for identi-

fying sites users have joined is to find users manually across sites. Users often provide personal information such as their real names, E-mail addresses, location, gender, profile photos, and age on different websites. This information can be employed to find the same individual on different sites. However, finding users manually on sites can be challenging and time consuming. Automatic approaches are also possible that can connect corresponding users across different sites [64, 73, 83, 88, 106, 136, 139]. A more straightforward approach is to use websites where users voluntarily list the sites they have joined. In particular, we find social networking sites, blogging and blog advertisement portals, and forums to be credible sources for collecting the sites users have joined. For example, on social networking sites such as Google+ or Facebook, users can list their IDs on other sites. Similarly, on blogging portals and forums, users are often provided with a feature that allows users to list their usernames in other social media sites.

We utilize these sources for collecting sites users have joined. Overall, we collect a set of 96,194 users, each having accounts on some of the aforementioned 20 social media sites. For each of the 20 sites, we develop a crawler that extracts the number of friends each individual has on the site. Hence, for each individual in our dataset, we have the number of friends a user has across different sites.

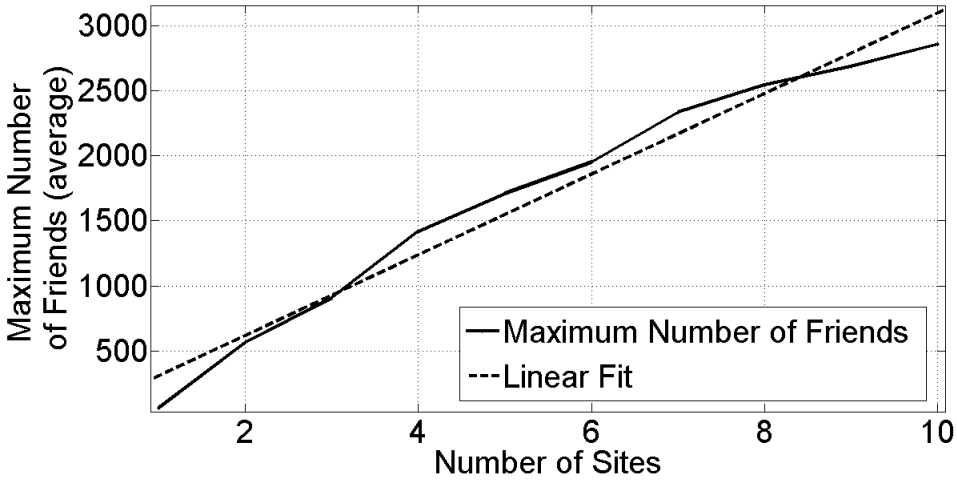
6.2 How Friendship Behavior Varies across Sites

One naturally expects that as users join more sites, it becomes more likely for them to find sites that contain more of their friends; therefore, befriending more individuals. Our data confirms this. Figure 6.1(a) plots the average maximum friend count for users that have joined different numbers of sites. We observe that as users join more sites, their maximum friend count across sites on average increases. A linear line ($g(x) = 309.8x - 0.005177$), found with 95% confidence, fits to the curve

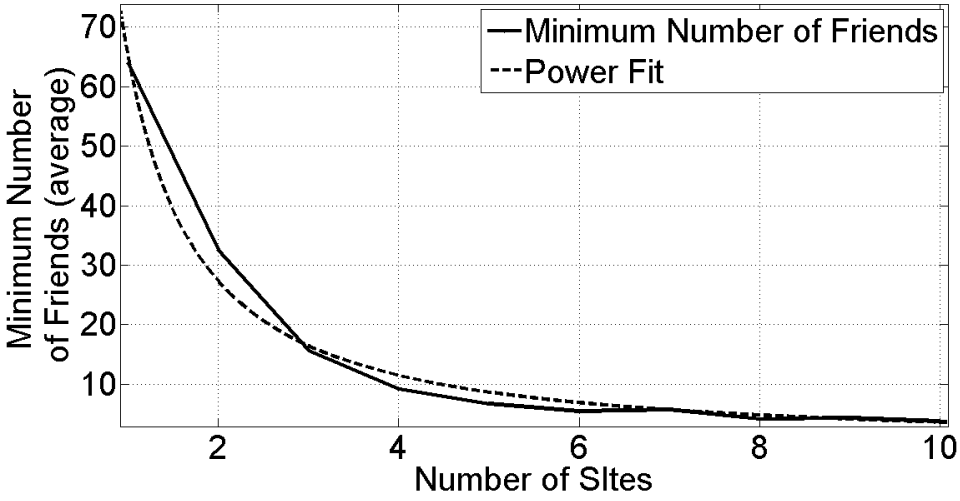
with adjusted $R^2 = 0.9978$. Hence, the expected maximum friend count across sites for users that have joined n sites is approximately n times more than that of users that have joined a single site. Similarly, one expects that as users join more sites, it becomes more likely for them to become inactive on some sites. Our data also confirms this. Figure 6.1(b) shows the average minimum numbers of individuals befriended across sites as users join more sites. We observe a decrease in the minimum number of friends across sites as users join more sites. A power function ($g(x) = 65.03x^{-1.251}$), found with 95% confidence, fits this curve with adjusted $R^2 = 0.9878$. In other words, unlike the likelihood of having many friends that increases linearly as users join sites, the probability of having a few friends increases exponentially. Having said that, one can conjecture that (1) as the minimum friend count across sites is decreasing more sharply than the maximum, one should expect a decrease in the average number of friends individuals have across sites. As an alternative, one can conjecture that (2) the average number of friends should increase because the maximum number of friends individuals have across sites is much higher than the (few) number of friends they have on sites that they are inactive.

Our data shows that neither of these conjectures are valid for average numbers of friends across sites. Figure 6.2 shows the average numbers of friends users have across sites as they join more sites. The figure shows that as users join more sites their average number of friends increases; however, once they join around 6 sites this average converges at around 400 friends. This average does not change much as users join new sites. This finding is in line with previous [43] and recent [55, 72, 94] literature on human cognitive limitations in maintaining communication and friendship with large groups of individuals.

There could be different explanations why the average of a distribution converges as we add more data points. For instance, by adding equally dispersed data points



(a) Average Maximum Numbers of Friends.



(b) Average Minimum Numbers of Friends.

Figure 6.1: Average Minimum and Maximum Numbers of Friends for Users that have Joined Different Numbers of Sites.

one can maintain the mean. To understand better how users befriend others, it is natural to observe how standardized moments of the friend count distribution changes. In particular, skewness [51], the third standardized moment ($\mathbb{E}[(\frac{X-\mu}{\sigma})^3]$), and kurtosis [37, 40], the forth standardized moment ($\mathbb{E}[(\frac{X-\mu}{\sigma})^4]$), can help us understand why the average number of friends converges as users join more sites.

Skewness shows where the mass of the distribution is concentrated and whether

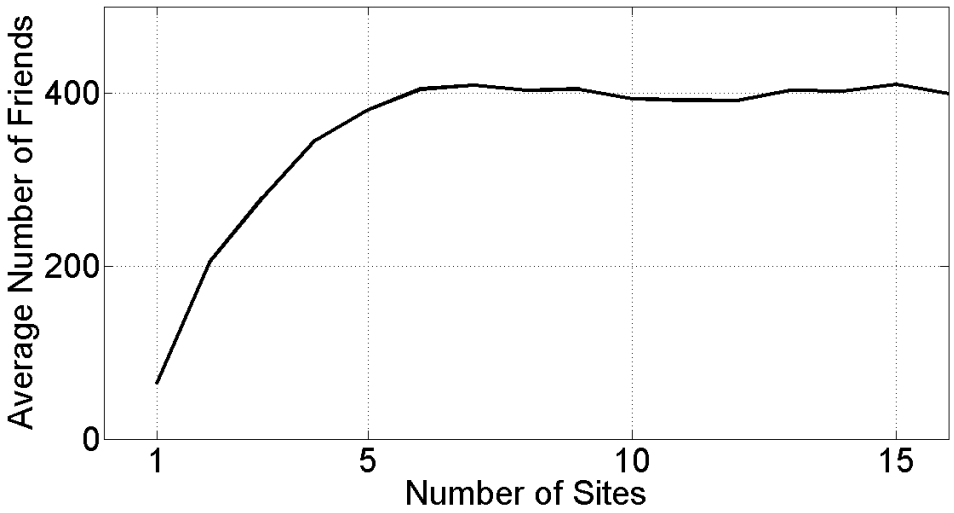


Figure 6.2: Average Numbers of Friends for Users that have Joined Different Numbers of Sites.

the left or right tail of the distribution is longer. Skewness of 0 demonstrates a normal distribution where the mean is equal to the median. A positive skewness shows that while extreme values exist to the right of the distribution, the mass of the distribution is concentrated on the left of it. Negative skewness shows the opposite. For example, sample: {1,2,3,1000} has a positive skewness and sample: {1,1001,1002,1003} has a negative skewness. To account for small-sample bias, we compute the bias-corrected skewness for sample $x = (x_1, x_2, \ldots, x_n)$ as follows:

$$s = \frac{\sqrt{n(n-1)}}{n-2} \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^3}{(\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2})^3}, \tag{6.1}$$

where $\bar{x}$ is the mean for x. For each user, we compute the skewness of the user's friend counts across sites. Figure 6.3 shows the empirical cumulative distribution function (Kaplan-Meier estimate) for these user skewness values for users that have joined different numbers of sites. We observe that most of the skewness values are positive showing that while there are extreme friend count values, the mass of the friend count distribution is concentrated on the left. Furthermore, we see that as users join more

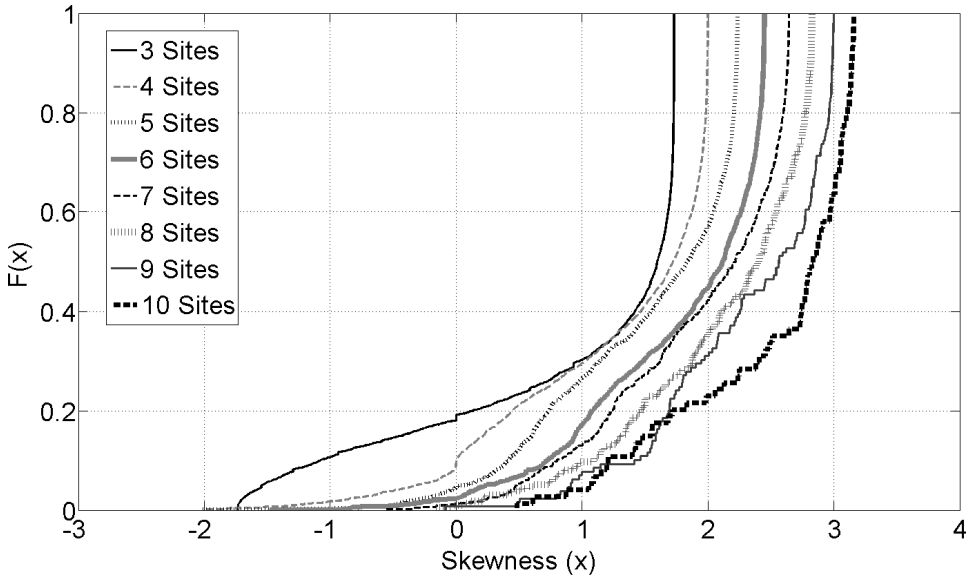


Figure 6.3: Empirical Cumulative Distribution for Skewness of Friend Distribution as Users Join More Sites.

sites, the cumulative distribution function (CDF) moves to the right, showing that as users join more sites, the proportion of sites where they have fewer friends increases. In other words, users that have joined a few sites are more likely to be highly active on some sites compared to those users that joined more sites. Although we now know that users are more likely to have fewer friends on most sites they join, it is not known how these fewer friend counts are distributed. To observe where these fewer friend count values are concentrated, we measure the kurtosis of the distribution.

Kurtosis value of a distribution measures the peakedness of a probability distribution and how heavy-tailed it is. We use the bias-corrected kurtosis for small sample $x = (x_1, x_2, \ldots, x_n)$:

$$k = \frac{n-1}{(n-2)(n-3)}((n+1)k_0 - 3(n-1)) + 3, \tag{6.2}$$

where k_0 is

$$k_0 = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^4}{(\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2)^2}. \tag{6.3}$$

A kurtosis value of 3 shows a normal distribution and a value greater than 3

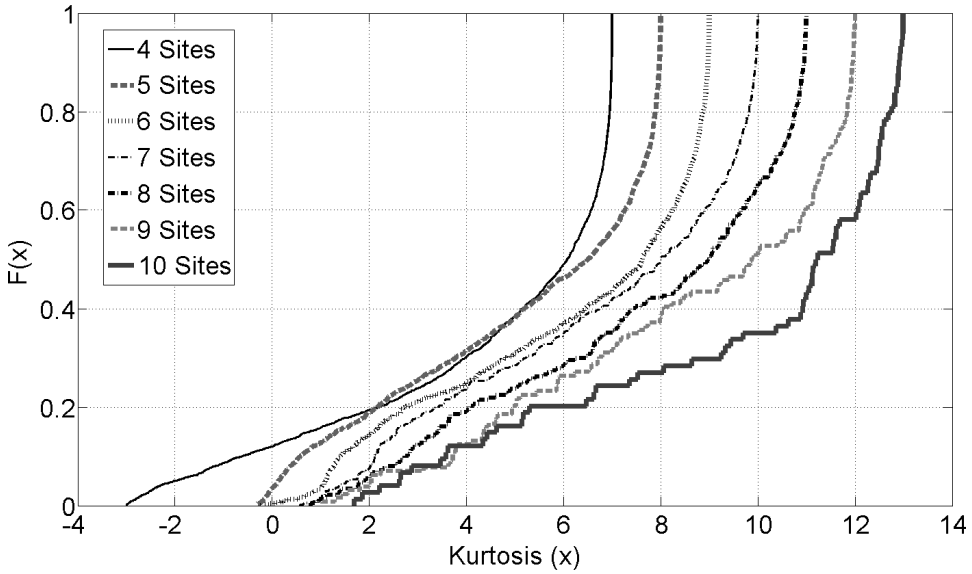


Figure 6.4: Empirical Cumulative Distribution for Kurtosis of Friend Distribution as Users Join More Sites.

shows a *leptokurtic* distribution that has a more acute peak around the mean and more heavy tails. Similarly, a negative kurtosis value shows a *platykurtic* distribution with a less pronounced and wider peak. For each user, we compute the kurtosis of the user's friend counts across sites. Figure 6.4 shows the empirical cumulative distribution (Kaplan-Meier estimate) for these user kurtosis values for users that have joined different numbers of sites. The graph shows that most kurtosis values are more than 3, denoting that the users' friend counts are more concentrated around the mean than normally expected. Furthermore, we observe that the CDF curves move to the right for users that have joined more sites. In other words, users' friend counts across sites tend to concentrate more around the mean value as users join more sites. Since we know from skewness analysis that users befriend a few others on most sites they join, this shows that the number of few individuals befriended are concentrated around a mean value. In other words, each user has almost the same number of friends (e.g., 10 friends) across most sites. The mean value varies for different users.

The initial increase in the average number of friends shows that when users join a

few sites, it is more likely for them to get engaged while befriending many; however, as they join more sites, they start to become inactive in those sites and the average converges.

6.3 How Popularity Changes across Sites

We have analyzed how the number of friends varies across sites. In this section, we perform similar experiments to analyze how user popularity changes across sites. To measure popularity we note that users with many friends are often considered popular users. So, a natural way to quantify popularity on a site is to use individual's friend count. However, the same number of friends on different social networks implies different levels of popularity due to different distributions of friend counts. For comparison, one can simply convert the friend count to the probability of observing the friend count, which is comparable across sites. A lower probability indicates a higher popularity. It is well known that the distribution of friend counts in a social media site often follows a power-law distribution [17, 100]. Hence, we perform the systematic procedure outlined in [30] for each of our 20 sites to determine their parameters for the power-law distribution. For integer values, the power-law distribution is defined as

$$p(x) = \frac{x^{-\alpha}}{\zeta(\alpha, x_{min})}, \tag{6.4}$$

where,

$$\zeta(\alpha, x_{min}) = \sum_{n=0}^{\infty} (n + x_{min})^{-\alpha} \tag{6.5}$$

is the generalized Hurwitz zeta function, α is the power-law exponent and x_{min} is the minimum value for which for all $x \geq x_{min}$, the power-law distribution holds. We estimate α and x_{min} using a finite sample correction bias using the maximum likelihood method outlined in [30]. Given these parameters, for any friend count

$f \geq x_{min}$, we estimate the probability of observing f (i.e., $p(x = f)$) using Equation 6.4.

Recent studies show that using the power-law distribution may not be always appropriate for modeling the friend count distribution of social networks [54, 119]. Hence, when $f < x_{min}$, instead of using Equation 6.4, we use the maximum likelihood estimate of $p(f)$,

$$p(f) = \frac{n_f}{n}, \tag{6.6}$$

where n_f is the number of users on the site with f friends and n is the total number of users on the site.

Following this approach, we estimate the probability of observing all friend counts in our dataset; hence, having the popularity of all users in our data across sites. Given these user popularity values across sites, we first measure how the average popularity varies across sites. Figure 6.5 provides average popularity for users that have joined different numbers of sites. Notice that convergence also takes place for user popularities. Users are least popular when they have joined a single site and they are most popular, when they have two or more accounts. Popularity saturates much faster and as users join sites, their average popularity remains unchanged.

While the average popularity shows how users popularity changes across sites on average, it does not show how a user's popularity changes as he or she joins new sites. This is because we have no temporal information on what sites were joined first and how popularity increased or decreased over time. However, one can approach this problem by computing the expected popularity change over time.

Consider a user for whom we have his or her number of friends on n sites. Let $f_1, f_2, \ldots, f_n$ denote the number of friends of this user on these sites. Among the n sites that the user has joined, there must be a site that is joined after all others. Since we have no temporal information, the last site could be any of the n sites. We consider

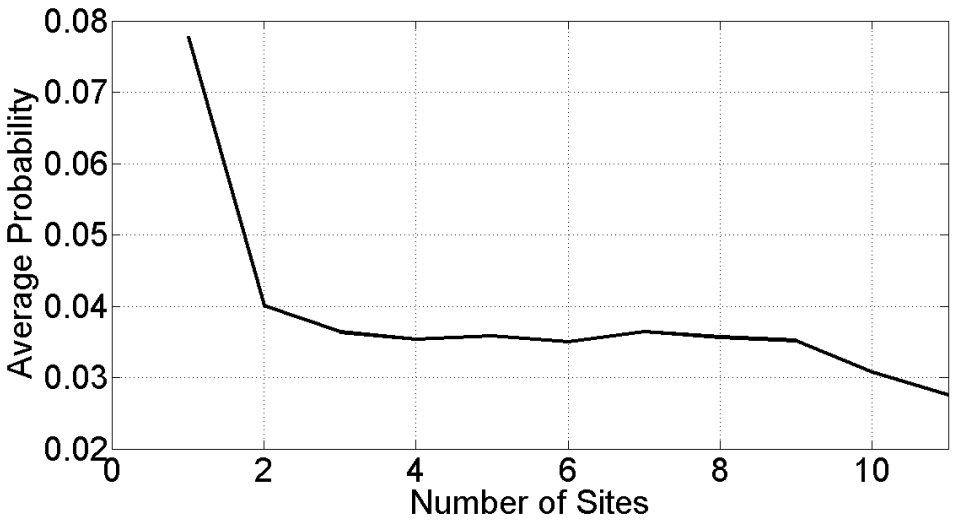


Figure 6.5: Average Popularity for Users that have Joined Different Numbers of Sites.

n cases. In each case, we consider one of the sites as the last site that the user has joined and the other $n-1$ sites as the sites that the user has joined in the past. In case $1 \leq i \leq n$, we consider that the user in the $n-1$ sites has $f_1^i, f_2^i, \ldots, f_{n-1}^i$ friends and f_n^i friends on the last site. The popularity values can be estimated by computing the probability of observing each friend count: $p(f_1^i), p(f_2^i), \ldots, p(f_n^i)$. For the $n-1$ sites that the user has joined, the maximum popularity that the user achieved is $\min(\ p(f_1^i), p(f_2^i), \ldots, p(f_n^i)\ )$. The user has become more popular on the nth site if and only if,

$$\min(\ p(f_1^i), p(f_2^i), \ldots, p(f_n^i)\ ) < p(f_n^i). \tag{6.7}$$

Thus, we measure popularity increase for case i as

$$p(f_n^i) - \min(\ p(f_1^i), p(f_2^i), \ldots, p(f_n^i)\ ). \tag{6.8}$$

Since, the last site that a user joined is not known, we compute the expected *popularity increase* as

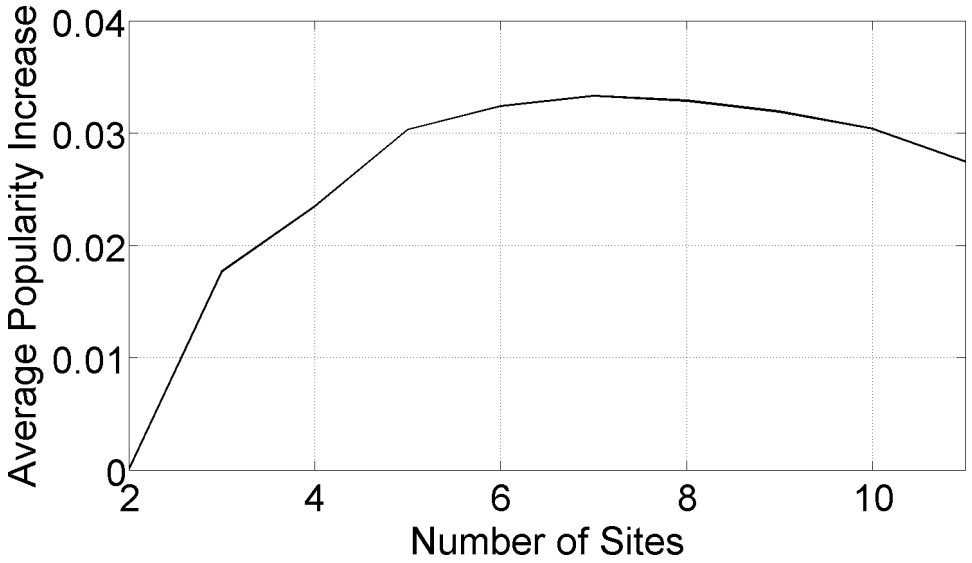


Figure 6.6: Average Popularity Increase for Users that have Joined Different Numbers of Sites.

$$\frac{1}{n}\sum_{i=1}^{n}[\ p(f_n^i) - \min(\ p(f_1^i), p(f_2^i), \ldots, p(f_n^i)\ )\ ]. \tag{6.9}$$

The average expected popularity increase for users that have joined different numbers of sites is provided in Figure 6.6. The figure shows that users tend to increase their popularity faster as they join more sites; however, there is a cap to the level at most a user can increase his or her popularity and this level is as users join 7 sites.

6.4 Predicting User Popularity

We have demonstrated that user friendships and popularity exhibits specific patterns as users join sites. This brings about a challenging, yet unexplored question: can one predict user's popularity on a new site?

Predicting user's popularity can not only help recommend new sites to users as they search for new sites on the web, but more importantly, can help sites identify users that are more likely to be interested in joining and becoming active on them. One expects a rather complicated solution to this problem. An approach that has

access to different types of information and users' interests and a matching procedure that identifies sites on which users are most likely to become active. Even then, one needs to know if the site includes friends of an individual for better popularity prediction.

If the popularity patterns is our data were meaningless, one should not be able to observe their effect in predicting user's popularity. By extracting popularity patterns a user has exhibited in the past, one can predict the popularity of a user in the future. In this section, we demonstrate how one can use **only** popularity patterns and outperform baseline methods that use no popularity patterns, safely concluding that the obtained popularity patterns can be used to predict user's popularity.

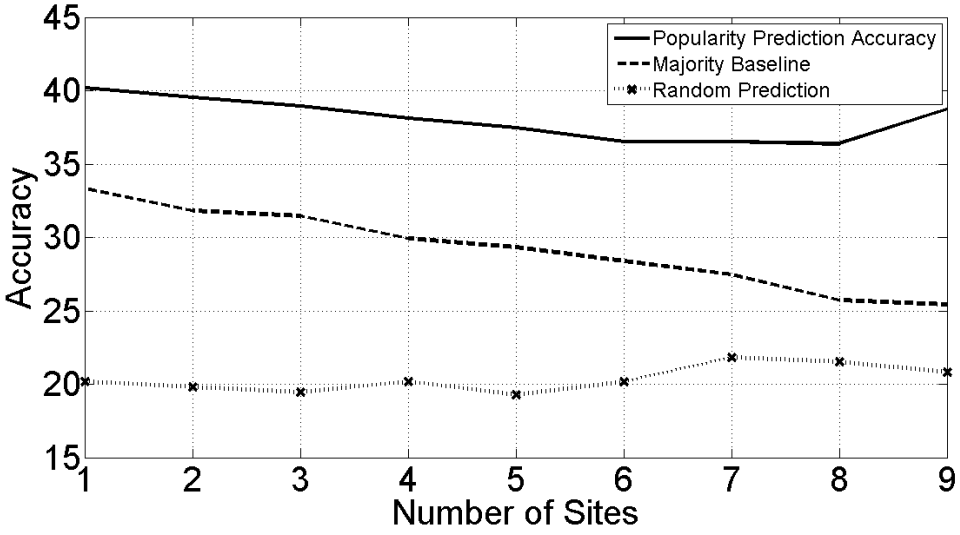
For any user in our dataset that has joined n sites, we assume that given the user's popularity level on $n-1$ of these sites, the popularity of the nth site should be predictable. To determine the popularity level of users in sites, we divide the users on each site into five categories. These categories are based on the level of popularity and their proportion are inspired by the diffusion of innovations theory [108], where individuals depending on their time of adopting a new product are categorized into 5 categories: innovators (top 2.5%), early adopters (next 13.5%), early majority (next 34%), late majority (next 34%), and laggards (last 16%). For each site, we divide users into 5 categories based on their level of popularity: elites, highly popular, averagely popular, averagely unpopular, and unpopular users. We use popularity categories instead of the actual probability as this introduces a generalizable prediction algorithm as users with different probabilities and new sites appear on social media. Thus, for each user that has joined n sites, we generate all the $\binom{n}{n-1} = n$ combinations of $n-1$ sites as historical data. For each combination, we construct a data instance of 5 features, each representing a popularity level. For each popularity level, we count the number of sites the user has joined in the past among his or her $n-1$ sites and

Table 6.1: Popularity Prediction Performance

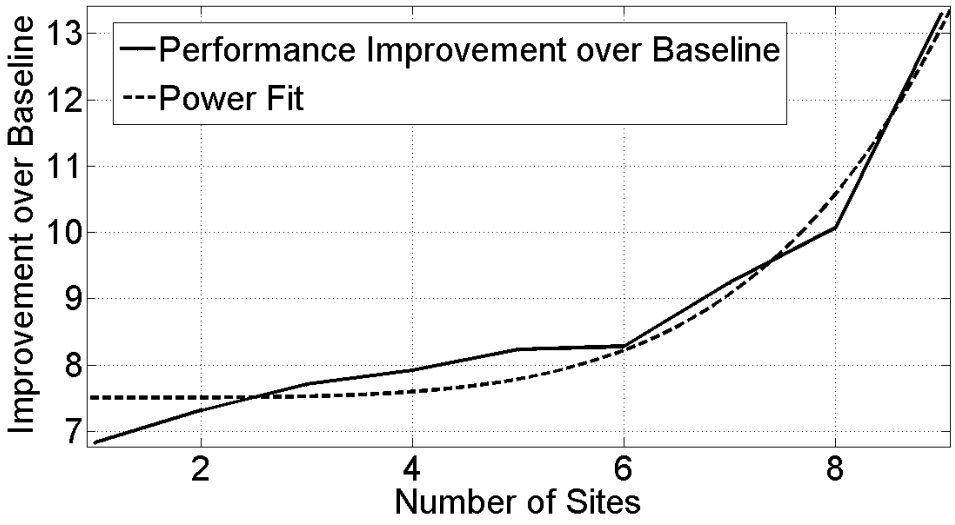
Technique	**AUC**	**Accuracy**
Logistic Regression	0.627	**39.26**%
SMO (sequential minimal optimization)	0.574	38.84%
J48 Decision Tree Learning	0.604	38.82%
Random Forest	0.612	38.63%
Naive Bayes	0.618	38.50%

has expressed that level of popularity. We set the class label as the popularity level for the user in the nth site (i.e., a value in {1,2,3,4,5}). We generate 39,130 instances. Our initial attempt to predict the class label in this dataset using Naive Bayes classifier predicts user popularity with an accuracy of 38.50% and an AUC of 0.618. To determine the sensitivity of our results to the learning bias of different algorithms, we test a variety of classification techniques. The results are provided in Table 6.1. We observe minimal sensitivity to learning bias, showing that one can reasonably predict user's popularity regardless of the classification algorithm. Logistic Regression performs the best with 39.26% accuracy in predicting user popularity and an AUC of 0.627. Thus, logistic regression is used for the rest of our experiments.

In our data, users have joined different numbers of sites. To verify helpfulness of adding more sites on user popularity prediction, we partition our dataset. Partition i contains the set of users that have already joined i sites. We perform popularity prediction for each partition. Figure 6.7(a) shows that the prediction results (accuracy) for each partition does not variate much. The figure also shows as a dashed line the majority class predictor for each partition and the random prediction results. Since the partitions were slightly imbalanced, we also computed the AUC and found that it was mostly fixed with an average AUC of 0.6273. The same figure shows that for all



(a) Popularity Prediction Accuracy as Users Join Different Numbers of Sites.



(b) Performance Improvement over Baseline.

Figure 6.7: Performance for Popularity Prediction.

cases, we outperform the majority predictor, proving that popularity patterns across sites can help predict the popularity of a user on a new site.

Figure 6.7(a) also shows that as users join more sites and more information becomes available the gap between the prediction outcome and the majority class starts to increase. The gap increase is provided in Figure 6.7(b). The gap increases expo-

nentially, fitting a power function ($g(x) = 8.65 \times 10^{-5} x^{5.068} + 7.506$) with adjusted $R^2 = 0.9494$. In other words, as more popularity patterns of a user becomes available to the prediction algorithm, one can predict user's popularity exponentially better.

Similar to the methodology used in Chapter 4, we are in fact using social signatures to predict popularity. However, we are discretizing social signature values to represent popularity level.

6.5 Related Work

Studying friendships and popularity on social media sites has a long history. The friendship network and popularity is often studied on a single site. Other related areas to the work presented in this chapter are (1) analyzing dynamics of multiple networks and (2) analyzing user behavior across social media. We briefly review related research from each of these three areas and outline how the work represented in this chapter stands compared to its related work.

Single-Site Friendship and Popularity Analysis. When considering only the number of friends individuals have, the analysis boils down to analyzing the degree distribution of social networks [24, 71]. It has been shown multiple times that the degree distribution of these social networks follows a power-law distribution [45, 49]. This study follows a similar approach; however, at a multi-site level, where we analyze how number of friends (degrees) changes across sites. Unlike the common degree distribution analysis where millions of nodes are analyzed to determine the degree distribution, with multiple sites, the number of available samples is limited to a few numbers. Hence, we take a different approach in this chapter by observing how the number of friends change across sites with the help of statistical measures.

Analyzing Dynamics of Multiple Networks. Comparing network characteristics of multiple networks has been the subject of recent studies [78, 95, 111]. For instance, Mislove et al. [95] analyze 4 networks: Flickr, YouTube, LiveJournal, and Orkut and demonstrate that these networks exhibit various properties such as being scale-free and having a densely connected core of high-degree nodes. Although these studies analyze multiple networks, the analysis is performed irrespective of the users that are shared across networks. The study in this chapter focuses on how friends of shared users across networks are distributed and how popularity for the users changes across social media sites.

Analyzing User Behavior Across Sites. Considering befriending as a behavior of individuals, the recent studies that analyze user behavior across sites becomes relevant to the study presented in this chapter. Some studies analyze how a specific behavior changes across sites without considering users that are shared across sites [21]. Other recent studies consider a specific behavior across sites such as Tagging [3, 92], but for users that are shared across sites. Our study is related to both as it analyzes the variation of an unexplored behavior (i.e., befriending) and user popularity across sites for users shared across sites.

6.6 Summary

Social media users are members of multiple sites. For a systematic study of users on social media one has to combine their information across sites. In this chapter, we investigate how this information varies across sites. We focus on the most fundamental information available across social media sites: user friends and their popularity.

By studying user friendships and popularity across sites, we showed that the maximum number of friends individuals have across sites increases linearly as users

join sites and their minimum drops exponentially. Furthermore, we noticed that as users join sites their average number of friends converges to a value near 400. We investigated this phenomenon even further and showed that as users join sites, the likelihood of observing fewer friend counts increases and at the same time, users frequently exhibit their mean behavior, such as always befriending 10 people. This frequent behavior of befriending a few friends on most sites leads to users converging to an average of 400 friends across sites.

By computing the power-law distribution parameters for these sites, we computed user popularity on sites. We found that popularity follows the same trend as in friend counts, converging to an average value. This result shows that users joining multiple sites cannot increase their average popularity and that the average popularity converges to a fixed value as users join sites. We also demonstrated that as users join sites, the amount their popularity can increase has a constant upper bound. Finally, we showed how the popularity patterns of users can be used to determine their popularity on future sites. Using discretized social signatures and a straightforward approach we showed that as patterns of popularity become available to the popularity prediction algorithm, the algorithm gains exponential performance gain over baselines.

Chapter 7

ACTIVITIES ACROSS SITES

I'm an idealist. I don't know where I'm going, but I'm on my way.

Carl Sandburg

In last two chapters, we discussed how user behavior can vary across sites. In particular, we investigated how users are distributed across sites and different joining patterns that are exhibited by different users. Furthermore, we demonstrated how friendships and popularity changes across sites. This shows how degrees, the most fundamental property of a node in a graph, vary across sites. However, once users across sites are identified, not only their variations in behavior, but also specific behaviors that are solely observable across sites can be analyzed. In this chapter, as a case study, we study one such challenging behavior.

Social media have shown considerable growth over the past years. With new sites launching everyday, Internet citizens, with their limited time and resources, are forced to select a few sites to spend their time online.

Social media sites must retain their existing users while continuing to attract new ones; therefore, Understanding user migration patterns in social media has several implications. It allows sites to (1) generate higher revenue from targeted advertising; (2) increase traffic to shared media, which in turn improves marketing outcomes; and (3) grow their base of long term customers, which in turn will increase brand loyalty. Moreover, understanding why users migrate is critical for preventing migrations.

The content in this chapter has been published at AAAI 2011 [76].

These implications motivate us to study the migration of users across different social media sites. These migrations can even take place within sites in the same social media category (social bookmarking, social networking, social media sharing, among others). In this chapter, we study the migration patterns of users across social media sites through a study of users from 7 popular social media sites. We propose a formal definition of migration and a framework for analyzing it. In particular, we show that

- Migration in social media can indeed be studied;
- There are clear user migration patterns across social media; and
- Specific categories of social media have fewer users migrating from.

The rest of the chapter is organized as follows: we first present the problem definition. Then, we describe how migration can be studied. The following section discusses the data collection process, the way migration patterns are obtained, and how their reliability is verified.

7.1 Problem Statement and Definitions

In this section, we formally define different types of migration and introduce other important definitions that will be used in the study.

Definition of Migration

Migration is the movement of users away from one location and towards another, either due to necessity, or attraction to the new environment. In the context of social media, we define two types of migration, *site migration* and *attention migration*. Let U_{s_1} be the set of members of site s_1 and U_{s_2} be the set of members of site s_2. Then,

the site migration of user u from social media sites s_1 to s_2, in a universe of two sites, can be defined as follows:

Definition 2 (Site Migration)**.** *Let $u \in U_{s_1}$ and $u \notin U_{s_2}$ at time t_i, if $u \notin U_{s_1}$ and $u \in U_{s_2}$ at time $t_j > t_i$, then for user u, a site migration has taken place between s_1 and s_2.*

Site migrations for an individual can be determined by checking the presence of a user's profile on sites s_1 and s_2 over time. A user's profile can be absent for different reasons, namely: profile removal, profile deletion, and account suspension.

- **Profile Removal.** Social media sites often remove the profiles of individuals who have been inactive for a long period. Profiles are also commonly removed for violating the site's code of conduct, such as posting inappropriate content on the site.

- **Profile Deletion.** At times, it is the user's decision to abandon a site. Many sites allow users to delete their profiles along with any content they might have uploaded on the site.

- **Account Suspension.** A user's account can also get suspended (but, not removed) for violating the code of conduct. In this case, the profile information is not accessible. Sites often allow suspended accounts to be re-instated by following specific procedures.

Among the three, account suspension is the least likely cause for the nonexistence of profile pages. Social media sites avoid suspending user accounts as much as possible to maintain their popularity and activity level. Using Twitter as an example, we illustrate how to estimate the number of users whose account was suspended. Using the API, we can only determine the existence of a user on the site and not

his suspension status. However, in Twitter, suspended profile pages contains the message "`Sorry, the profile you are trying to view has been suspended.`" Therefore, by HTML scraping and searching for this message, one can determine whether an account is suspended or not. We found that approximately 2.8% of the users whose accounts could not be found during a crawl, had a suspended account. Most of these profiles have been suspended for more than a year. Hence, the likelihood of an account being suspended is small. While site migrations are possible, a more realistic scenario is for the attention to migrate. In this case the user account is not deleted, but the individuals abandons the site and becomes inactive. The attention migration of user u from social media sites s_1 to s_2 can be defined as

Definition 3 (Attention Migration)**.** *Let $u \in U_{s_1}$ and $u \in U_{s_2}$ at time t_i and u be active at time t_i at s_1 and s_2. If u is inactive at s_1 and active at s_2 at time $t_j > t_i$, then the user's attention is said to have migrated away from site s_1 and towards site s_2.*

The activity (or inactivity) of a user is determined using the following,

Definition 4 (User Activity)**.** *Given a site s, a user $u \in U_s$, times $t_j > t_i$, and time interval $\delta = t_j - t_i$, u is active on s at time t_j, if the user has performed an action on the site since time t_i. Otherwise, the user is considered inactive.*

The interval δ could be measured at different granularity levels, such as *days*, *weeks*, *months*, and *years*. The user's actions could be one of the many possible ones on site, such as submitting a news story, posting a status message, uploading a video, or the like. For instance, a Delicious user is considered active in July, 2014, if she has submitted at least one bookmark since June, 2014. Here, $\delta = 1$ month.

In summary, the attention migration of a user on site s is the inactivity of the user for a time period δ on s. Attention migration can be considered as a short term

migration of the individual, which might lead to a site migration after prolonged inactivity by the user.

7.2 Studying Migration Patterns

There is a growing interest to determine the extent to which social media sites are capable of attracting individuals. This can help us determine the "green pastures" in social media. These "green pastures" are sites which have the features necessary to attract users and hence cause migration. These features could be their appealing functionalities that could be extracted and utilized by other sites as guidelines for improvement.

For migration studies, three general steps have to be taken. First, reliable data needs to collected. Then, migration patterns should be obtained. Finally, these patterns should be validated. Nest, we discuss these steps in separate sections.

7.3 Data Collection

We present a brief overview of our data collection methodology for the experiments and describe the important characteristics of the data. There are hundreds of social media sites and new ones are launching every year. Not only it is impossible to analyze all, it would also be impractical. Suitable social media sites for this kind of study should have the following characteristics:

- The sites should have sufficient user activity for measuring migration.
- The sites should have sufficient number of users to be worthwhile to study.
- They should have been launched at different times to enable the observation of user movement across them.
- Sites should preferably cover the major categories of social media, such as social

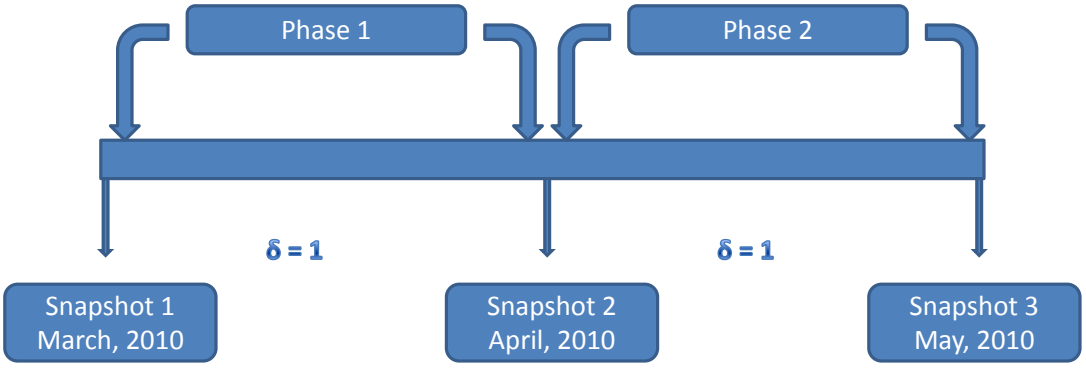


Figure 7.1: Data Collection Timeline for the Migration Dataset

bookmarking, media sharing, micro-blogging, among others.

These requirements impose significant challenges to data collection. In addition, there exists the problem of resolving user identities across social media sites as previously discussed in Chapter 3. We selected 7 popular and representative sites for this study: *Delicious, Digg, Flickr, Reddit, StumbleUpon, Twitter, and YouTube*. We collected more than 17,798 users who had at least 2 identities in one of these 7 popular social media sites.

To study user migration between social media sites we need the activity information of the users on these 7 sites. Using APIs when available and screen scraping in other cases, we collected the activity and user profile information of these users in the identified 7 popular social media sites. The collection of user information on these sites was carried out in March 2010, April 2010, and May 2010. The data for each month corresponds to a snapshot and the value of the time window parameter δ can be used to control the time difference between two snapshots. In this study, we set $\delta = 1$ month. We obtain two phases of user data across these social media sites, where each phase is defined as the data from two consecutive snapshots. In this case, Phase 1 spans March and April data while Phase 2 spans April and May data. A descriptive figure showing this procedure is presented in Figure 7.1. The information

Table 7.1: Migration Dataset

Site	No of Users	Profile Attributes
Delicious	8,483	10
Digg	9,161	20
Flickr	5,363	11
Reddit	2,392	5
StumbleUpon	8,935	13
Twitter	13,819	15
YouTube	7,801	19

collected from a user's profile on these sites include real name, age, location, status messages, friends, followers, among other attributes.

The number of users who had an account on on each one of these sites along with the number of their collected profile attributes are presented in Table 7.1.

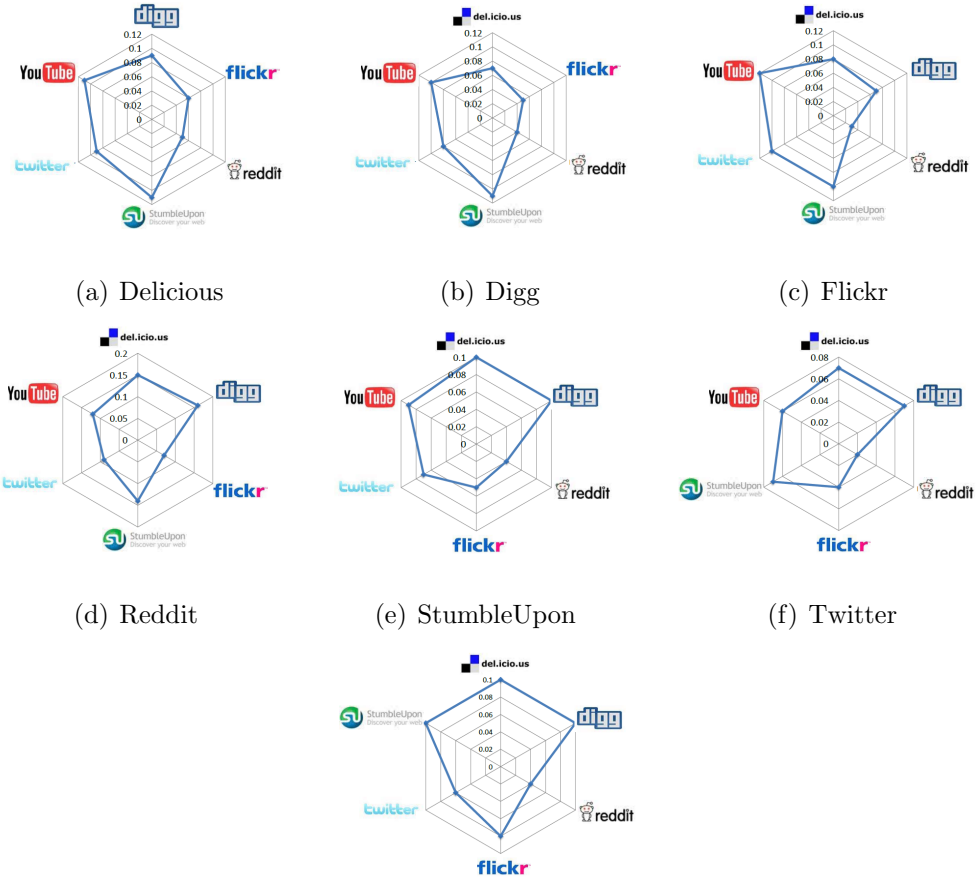
Next, we present how migration patterns were obtained using the migration dataset.

7.4 Obtaining Migration Patterns

Here, our goal is to find (1) how users migrate across social media sites, (2) social media categories that rank higher with respect to migration, and (3) the migration trend of users across social media sites, and the social media sites from where users migrate.

As site migrations are highly improbable, we focus on attention migrations that are more likely. We identify sites that are able to retain the attention of their users and the ones that are losing their attention over time. More importantly, we identify sites toward which users are migrating.

We first measure the number of users whose attention migrates away from a site.



(a) Delicious (b) Digg (c) Flickr

(d) Reddit (e) StumbleUpon (f) Twitter

(g) YouTube

Figure 7.2: Pairwise Attention Migration Patterns between Different Social Media Sites

We also identify where their attention diverts to. We use data from the three snapshots to identify the trend of attention migration in each of the 7 social media sites. We select those users whose attention migrated away from a site to another. We examined their activity on all sites to identify sites toward which their attention migrated. Our results are presented in Figure 7.2 in the form of radar charts. Each radar chart corresponds to the migration of individuals from a site towards other social media sites. Each spoke in the chart represents a social media site and the value of the radii represents the migrating tendency toward the site represented by the spoke. The charts show that attention migration does exist between the social media

sites. Otherwise, all the points in the corresponding radar chart would just be a dot in the center, marking 0. The summation of these radii values do not necessarily sum up to 1 as a single user may migrate to several sites. From the results in Figure 7.2, it is clear that the general trend of attention migration of users from most sites is towards Twitter and StumbleUpon. Reddit users had the highest amount of migration to other sites. The number of users migrating to Reddit was also the smallest among all social media sites observed. The most significant fraction of Reddit's population (16% of the users) migrated to Digg. Digg is another social news site where users can "digg" a news story and make it popular. Then, these popular storied are promoted to the front page. Similarly, we see a significant migration between StumbleUpon and Delicious. Our observations show that migration is more localized within the social media category, as users have a tendency to migrate to other sites within a category that offer similar functionalities but are more appealing. Herd behavior could be another possible explanation for these migrations.

7.5 Reliability of Migration Patterns

Although we identify specific migration patterns, it can be argued that given the size of our dataset, patterns could be fortuitous. To validate migration patterns, we perform statistical tests. We first create a reference point to compare our results with. In our case, this would be the random migration of individuals. We assume that only the migration of individuals with specific characteristics can lead to the results we have observed. Given any other set of randomly selected individuals, we would not expect to observe patterns such as migration between competing sites, like Delicious and StumbleUpon. Thus, we formulate our null hypothesis as

H_0: The migration of individuals is random and no correlation exists between their user attributes such as their network activity on a site

and their migration

Inspired by the shuffle test proposed in [13], we can create shuffled datasets in which we can guarantee that user attributes did not result in migrations. To construct shuffled dataset, for each site we randomly select the same number of users from the potential migration population (overlapping users between the active users of phase 1 and 2) an assume that they have migrated. We construct 10 such shuffled datasets for each site. To compare the outcome of the shuffled datasets and the true migrating users, we need to measure the distance between their observed patterns. One way to measure this distance is to compare how the relationship of a user's attributes to his migration behavior varies across datasets. The relationship between attributes and migration behavior can be determined using techniques such as logistic regression:

$$Y_m = \frac{e^z}{1+e^z}, \tag{7.1}$$

where $z = \alpha x + \beta$. We use the boolean variable Y_m, which indicates whether a user has migrated away from a site, as the class attribute. The coefficient α represents the correlation of the attribute x with the class attribute. In our case, we used attributes that represent user's Activity A (e.g., number of tweets), user's network activity N (e.g., number of friends), and user's rank R (user's rank in Google search results) as the features. This procedure can be similarly applied to each shuffled dataset for a site. We can then obtain the average of the logistic regression coefficients for each user attribute and for all of the sites in our 10 shuffled datasets. Using the observed regression coefficients, we evaluate the null hypothesis using the χ^2-statistic as follows,

$$\chi^2 = \sum_{i=1}^{n} \frac{(O_i - E_i)^2}{E_i}, \tag{7.2}$$

where n is the number of regression coefficients, O_i is the observed coefficient value from the dataset, and E_i is the coefficients obtained from the shuffled dataset.

Table 7.2: χ^2 Test Results on Observed and Shuffled Data

Site	**Observed Coefficients**			**Shuffled Coefficients**			p**-value**	**Statistical Significance**
	N	A	R	N	A	R		
Delicious	0.2858	0.4585	-	0.6029	0.5921	-	0.65	Not significant
Digg	0.4796	0.8066	-	0.52	0.5340	-	0.70	Not significant
Flickr	1	1	0.9797	0.2922	0.2759	0.4982	0.13	Not significant
Reddit	0.5385	0.6065	-	0.4846	0.6410	-	0.92	Not significant
StumbleUpon	1	1	-	0.4191	0.2059	-	0.0492	Significant
Twitter	0.5215	1	0.5335	0.2811	0.0365	0.4009	0.0001	Significant
YouTube	0	1	0.1644	0.7219	0.0040	0.4835	0.0001	Significant

Table 7.2, shows the results of applying chi-square test on the observed and the shuffled dataset. Missing coefficients for the Google rank of users is represented using the symbol $-$, because for some sites all the users had a value of 0 for this attribute.

The p-values indicates how random the obtained migration patterns are. We consider, the result to be statistically significant if $p < 0.05$. From Table 7.2, we notice that the migration patterns for users from sites Delicious, Digg, and Reddit are highly similar to the shuffled dataset. On further investigation, we identified that this was due to the small size of the potential migration population which was used to select the individuals who migrated. We also notice that the Flickr dataset, although not statistically significant, is still quite different from the shuffled datasets and has a low p-value. On the other hand, StumbleUpon, Twitter, and YouTube strongly reject our null hypothesis and the patterns from these datasets are clearly distinct from those of the shuffled dataset. These results also support our earlier observations that show that a majority of the user migration is towards StumbleUpon and Twitter. In addition, during our experiments, we observe that user activity has a high correlation with the migration of an individual away from a site.

7.6 Summary

In this chapter, we show that (1) studying migration across social media is feasible, and (2) migration patterns can be identified across social media. To study migration patterns, we define two types of migration. We analyze users migrating from 7 popular social media sites. Using a variety of social media sites, we identify migration patterns that demand further research on solutions to prevent or encourage migrations. For example, social news sites such as Digg or Reddit have the highest number of users migrating away (low user retention rates). Identifying these migration patterns are valuable to social media sites in several ways. For example, by designing features to recapture user attention before the exodus begins and learning to avoid similar pitfalls when launching new social media sites.

Chapter 8

CONCLUSIONS AND FUTURE WORK

No book can ever be finished.
While working on it we learn just
enough to find it immature the
moment we turn away from it.

Karl Popper

In this chapter, we conclude this dissertation with a summary of our contributions and a review of our future work.

8.1 Contributions

In this section, we summarize the contribution in this dissertation:

1. **Identifying User with Minimum Information:** we develop methods that can identify users across social media sites with minimum information. In particular, we investigate both link- and content-based method.

   (a) **Link-Based Identification:** we introduce link-based techniques that employ minimum link information across sites. We investigate why (sub)graph isomorphism-based methods fail in social networks and demonstrate properties of social networks that make (sub)graph isomorphsim challenging. Finally, we introduce *social signatures* as different way of tackling user identification. In addition, we show how social signatures can be used to reconstruct graphs

(b) **Content-Based Identification:** we introduce behavioral modeling, a strategy for gleaning digital traces of human behavior in the content that they generate. Behavioral modeling has been in different applications such as sarcasm detection on social media [107]. In addition, we introduce MOBIUS, a content-based methodology that uses behavioral modeling for user identification with minimum information. We show that user identification with minimum content information is highly effective. Inspired by studies in psychology and sociology, we introduce a large set of computational features for efficient user identification with content information.

2. **Applications of Minimum Information:** Considering that users on social media are either normal or malicious, we investigate two representative applications that utilize minimum information for each category of users. For normal users, we investigate friend recommendation and show that minimum content information, combined with features that can detect social forces that result in friendships (homophily, influence, among others) can help detect future friends with performance comparable to state-of-the-art link prediction that has access to more information. For malicious users, we investigate literature from psychology and criminology, and combine that with machine learning and complexity theory, to efficiently detect malicious users, yet with minimum information. Our results show that the information complexity of malicious users makes them distinguishable from normal users. The performance of the methodology for detecting malicious users is comparable to that of state-of-the-art malicious user detection techniques that have access to extra information.

3. **Analyzing User Behavior across Sites:** By identifying users across sites, we study (1) *patterns*, (2) *variations*, and (3) *behaviors* across sites.

(a) **Patterns across Sites.** We investigate the basic patterns of users that are clearly visible across sites. In particular, we demonstrate how users select sites to join across social media and how joining patterns can be used to predict *future* sites that users will join. In addition, we show the statistical distributions that govern how individuals are distributed across social media.

(b) **Variations across Sites.** We investigate how users behavior varies across sites. In particular, we focus on the fundamental question of how friendships vary across sites and how the degree distribution changes across sites. In addition, we show how the average number of friends changes across sites. Our findings are aligned with studies in evolutionary psychology.

(c) **Behaviors across Sites.** We investigate specific behaviors that are only observable across sites. In particular, we demonstrate how user migrations can be analyzed across sites and introduce a randomization-test based method for detecting migrations without ground truth. The method can be used in other domains and social media research, when ground truth is unavailable [143].

8.2 Future Directions

Our work opens the door to many interesting theoretical problems and applications. In particular, we find the following of interest:

1. **Considering Minimum Information in other Domains.** While we investigated minimum content and link information for user identification in social media, it is worth investigating what the absolute minimum information required to perform a task with a given accuracy is in social media research. For

instance, this minimum information could be a profile picture or a click on a link.

2. **Systematic Improvement to Minimum Information.** How can we systematically add information such that specific performance guarantees are met? That is given specific applications, how can we determine the minimum information that does not sacrifice accuracy.

3. **Improving Link-Prediction with Minimum Information.** Building upon the work presented in Chapter 4, promising directions for future work include (1) studying addition of other information and (2) analyzing how combining the proposed approach with traditional link prediction can further improve the performance of link prediction. Future work also includes analyzing how social forces can be combined (instead of considering them independently) to further improve friend finding performance. While we demonstrate that all social forces are beneficial for finding friends, the comparison between forces can be influenced by the performance of the classifiers. We leave verifying our findings with labeled data in which age, location, or language is known as another part of our future work.

4. **Integrating Extra information for Malicious User Detection.** That is integrating additional information available across sites in a principled manner. However, this extension requires considering the heterogeneity of data available across sites. In addition, similar to the observation we had regarding the information surprise values of usernames, we are interested in how surprise values change for other content generated by users.

5. **Content Variation across Sites.** We have analyzed link variations across

sites. A promising future direction is to investigate content variations across sites. While data collection for our study was challenging, we believe with more data, especially content information, regarding the behavior and interests of users across sites, one should be able to obtain deeper insights into how users change behavior across sites.

6. **Predicting/Analyzing Temporal Popularity across sites.** While we showed how popularity changes across sites, the temporal information was missing. A future direction is to analyze how popularity changes across sites over time. Furthermore, determining the types of popularity patterns of users and the number of different clusters of people with respect to their popularity patterns are of interest.

REFERENCES

[1] Turning into digital goldfish. *BBC News*, 2002.

[2] A. Abbasi and H. Chen. Applying Authorship Analysis to Extremist-group Web Forum Messages. *IEEE Intelligent Systems*, 20(5):67–75, 2005.

[3] Fabian Abel, Eelco Herder, Geert-Jan Houben, Nicola Henze, and Daniel Krause. Cross-system user modeling and personalization on the social web. *User Modeling and User-Adapted Interaction*, 23(2-3):169–209, 2013.

[4] Lada A Adamic and Eytan Adar. Friends and neighbors on the web. *Social networks*, 25(3):211–230, 2003.

[5] Nitin Agarwal. *Social computing in blogosphere*. PhD thesis, Arizona State University, 2009.

[6] Charu C Aggarwal and S Yu Philip. *A general survey of privacy-preserving data mining models and algorithms*. Springer, 2008.

[7] Dakshi Agrawal and Charu C Aggarwal. On the design and quantification of privacy preserving data mining algorithms. In *Proceedings of the twentieth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 247–255. ACM, 2001.

[8] R. Agrawal, S. Rajagopalan, R. Srikant, and Y. Xu. Mining newsgroups using networks arising from social behavior. In *Proceedings of the 12th international conference on World Wide Web*, page 535. ACM, 2003.

[9] R. Agrawal and R. Srikant. Privacy-Preserving Data Mining. *ACM Sigmod Record*, 29(2):439–450, 2000.

[10] William Aiello, Fan Chung, and Linyuan Lu. A random graph model for massive graphs. In *Proceedings of the thirty-second annual ACM symposium on Theory of computing*, pages 171–180. Acm, 2000.

[11] Réka Albert and Albert-László Barabási. Statistical mechanics of complex networks. *Reviews of modern physics*, 74(1):47, 2002.

[12] E. Amitay, S. Yogev, and E. Yom-Tov. Serial Sharers: Detecting Split Identities of Web Authors. In *SIGIR PAN workshop*, 2007.

[13] Aris Anagnostopoulos, Ravi Kumar, and Mohammad Mahdian. Influence and correlation in social networks. In *KDD*, pages 7–15, 2008.

[14] László Babai and Eugene M Luks. Canonical labeling of graphs. In *Proceedings of the fifteenth annual ACM symposium on Theory of computing*, pages 171–183. ACM, 1983.

[15] L. Backstrom, C. Dwork, and J. Kleinberg. Wherefore Art Thou R3579X?: Anonymized Social Networks, Hidden Patterns, and Structural Steganography. In *WWW*, pages 181–190. ACM, 2007.

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

[17] A.L. Barabási and R. Albert. Emergence of scaling in random networks. *science*, 286(5439):509–512, 1999.

[18] Anne Barron. Understanding spam: A macro-textual analysis. *Journal of Pragmatics*, 38(6):880–904, 2006.

[19] Fabrıcio Benevenuto, Gabriel Magno, Tiago Rodrigues, and Virgılio Almeida. Detecting spammers on twitter. In *CEAS*, volume 6, page 12, 2010.

[20] Fabricio Benevenuto, Tiago Rodrigues, Virgilio Almeida, Jussara Almeida, Chao Zhang, and Keith Ross. Identifying video spammers in online social networks. In *Proceedings of the 4th international workshop on Adversarial information retrieval on the web*, pages 45–52. ACM, 2008.

[21] Fabrício Benevenuto, Tiago Rodrigues, Meeyoung Cha, and Virgílio Almeida. Characterizing user behavior in online social networks. In *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference*, pages 49–62. ACM, 2009.

[22] L. Bilge, T. Strufe, D. Balzarotti, and E. Kirda. All Your Contacts Are Belong to Us: Automated Identity Theft Attacks on Social Networks. In *WWW*, pages 551–560. ACM, 2009.

[23] Béla Bollobás. *Extremal graph theory*. Courier Corporation, 2004.

[24] Andrei Broder, Ravi Kumar, Farzin Maghoul, Prabhakar Raghavan, Sridhar Rajagopalan, Raymie Stata, Andrew Tomkins, and Janet Wiener. Graph structure in the web. *Computer networks*, 33(1):309–320, 2000.

[25] Qiang Cao, Michael Sirivianos, Xiaowei Yang, and Tiago Pregueiro. Aiding the detection of fake accounts in large scale social online services. In *NSDI*, 2012.

[26] P. Cashmore. 60% of Twitter Users Quit Within the First Month. `http://on.mash.to/zVwKb`, 2009.

[27] S.F. Chen and J. Goodman. An Empirical Study of Smoothing Techniques for Language Modeling. In *ACL*, pages 310–318, 1996.

[28] Zi Chu, Steven Gianvecchio, Haining Wang, and Sushil Jajodia. Who is tweeting on twitter: human, bot, or cyborg? In *ACSAC*, pages 21–30, 2010.

[29] R. Cilibrasi and P.M.B. Vitányi. Clustering by Compression. *IEEE Transactions on Information Theory*, 51(4):1523–1545, 2005.

[30] Aaron Clauset, Cosma Rohilla Shalizi, and Mark EJ Newman. Power-law distributions in empirical data. *SIAM review*, 51(4):661–703, 2009.

[31] Private Communication. Private Communication with a Yahoo! employee, 2013.

[32] Stephen A Cook. The complexity of theorem-proving procedures. In *Proceedings of the third annual ACM symposium on Theory of computing*, pages 151–158. ACM, 1971.

[33] Thomas M Cover and Joy A Thomas. *Elements of information theory*. John Wiley & Sons, 2012.

[34] T.M. Cover and J.A. Thomas. *Elements of Information Theory*. Wiley-interscience, 2006.

[35] D. Cowan. *An Introduction to Modern Literary Arabic*, volume 240. Cambridge University Press, 1958.

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

[37] Harald Cremér. *Mathematical Methods of Statistics (PMS-9)*, volume 9. Princeton university press, 1999.

[38] George Danezis and Prateek Mittal. Sybilinfer: Detecting sybil nodes using social networks. In *NDSS*, 2009.

[39] O. De Vel, A. Anderson, M. Corney, and G. Mohay. Mining E-mail Content for Author Identification Forensics. *ACM Sigmod Record*, 30(4):55–64, 2001.

[40] Lawrence T DeCarlo. On the meaning and use of kurtosis. *Psychological Methods*, 2(3):292, 1997.

[41] I. Dinur and K. Nissim. Revealing Information while Preserving Privacy. In *Proceedings of the 22nd ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, pages 202–210, 2003.

[42] C. Doctorow. Preliminary Analysis of LinkedIn User Passwords. `http://bit.ly/L5AHo3`.

[43] Robin IM Dunbar. Neocortex size as a constraint on group size in primates. *Journal of Human Evolution*, 22(6):469–493, 1992.

[44] T. Dunning. *Statistical Identification of Language*. CR Lab, New Mexico State University, 1994.

[45] D. Easley and J. Kleinberg. Networks, Crowds, and Markets: Reasoning About a Highly Connected World, 2010.

[46] J. Elder. Inside a Twitter Robot Factory. `http://on.wsj.com/1bdQbEI`.

[47] Lee Ellis, Kevin M Beaver, and John Wright. *Handbook of crime correlates*. Academic Press, 2009.

[48] A. Evfimievski, J. Gehrke, and R. Srikant. Limiting Privacy Breaches in Privacy Preserving Data Mining. In *Proceedings of the 22nd ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, pages 211–222, 2003.

[49] Michalis Faloutsos, Petros Faloutsos, and Christos Faloutsos. On power-law relationships of the internet topology. In *ACM SIGCOMM Computer Communication Review*, volume 29, pages 251–262. ACM, 1999.

[50] C.A. Ferguson. Word Stress in Persian. *Language*, 33(2):123–135, 1957.

[51] Sir Ronald Aylmer Fisher, Statistiker Genetiker, Ronald Aylmer Fisher, Statistician Genetician, Great Britain, Ronald Aylmer Fisher, and Statisticien Généticien. *Statistical methods for research workers*, volume 14. Oliver and Boyd Edinburgh, 1970.

[52] Hongyu Gao, Jun Hu, Christo Wilson, Zhichun Li, Yan Chen, and Ben Y Zhao. Detecting and characterizing social spam campaigns. In *IMC*, pages 35–47. ACM, 2010.

[53] Saptarshi Ghosh, Bimal Viswanath, Farshad Kooti, Naveen Kumar Sharma, Gautam Korlam, Fabricio Benevenuto, Niloy Ganguly, and Krishna Phani Gummadi. Understanding and combating link farming in the twitter social network. In *WWW*, pages 61–70. ACM, 2012.

[54] M. Gjoka, M. Kurant, C.T. Butts, and A. Markopoulou. Walking in Facebook: A case study of unbiased sampling of OSNs. In *INFOCOM, 2010 Proceedings IEEE*, pages 1–9. IEEE, 2010.

[55] Bruno Gonçalves, Nicola Perra, and Alessandro Vespignani. Modeling users' activity on twitter networks: Validation of dunbar's number. *PloS one*, 6(8):e22656, 2011.

[56] Sam Gosling. *Snoop: What your stuff says about you*. Basic Books, 2009.

[57] Z. Gyongyi, H. Garcia-Molina, and J. Pedersen. Combating web spam with trustrank. In *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*, pages 576–587. VLDB Endowment, 2004.

[58] N. Habash, A. Soudi, and T. Buckwalter. On Arabic Transliteration. *Arabic Computational Morphology*, pages 15–22, 2007.

[59] C Harris. Detecting deceptive opinion spam using human computation. In *Workshops at AAAI on Artificial Intelligence*, 2012.

[60] Michael Hay. Enabling accurate analysis of private network data. 2010.

[61] Michael Hay, Gerome Miklau, David Jensen, Don Towsley, and Chao Li. Resisting structural re-identification in anonymized social networks. *The VLDB Journal*, 19(6):797–823, 2010.

[62] Michael Hay, Gerome Miklau, David Jensen, Don Towsley, and Philipp Weis. Resisting structural re-identification in anonymized social networks. *Proceedings of the VLDB Endowment*, 1(1):102–114, 2008.

[63] Bert C Huang and Tony Jebara. Loopy belief propagation for bipartite maximum weight b-matching. In *International Conference on Artificial Intelligence and Statistics*, pages 195–202, 2007.

[64] T. Iofciu, P. Fankhauser, F. Abel, and K. Bischoff. Identifying Users across Social Tagging Systems. In *ICWSM*, pages 522–525, 2011.

[65] Muhammad Asim Jamshed, Wonho Kim, and KyoungSoo Park. Suppressing bot traffic with accurate human attestation. In *Proceedings of the first ACM asia-pacific Workshop on systems*, pages 43–48. ACM, 2010.

[66] Tony Jebara, Jun Wang, and Shih-Fu Chang. Graph construction and b-matching for semi-supervised learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 441–448. ACM, 2009.

[67] Ioannis Kanaris, Konstantinos Kanaris, Ioannis Houvardas, and Efstathios Stamatatos. Words versus character n-grams for anti-spam filtering. *International Journal on Artificial Intelligence Tools*, 16(06):1047–1067, 2007.

[68] Ioannis Kanaris, Konstantinos Kanaris, and Efstathios Stamatatos. Spam detection using character n-grams. In *Advances in Artificial Intelligence*, pages 95–104. Springer, 2006.

[69] Andreas M Kaplan and Michael Haenlein. Users of the world, unite! the challenges and opportunities of social media. *Business horizons*, 53(1):59–68, 2010.

[70] E. Keogh, S. Lonardi, and C.A. Ratanamahatana. Towards Parameter-Free Data Mining. In *KDD*, pages 206–215, 2004.

[71] Jon M Kleinberg, Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, and Andrew S Tomkins. The web as a graph: Measurements, models, and methods. In *Computing and combinatorics*, pages 1–17. Springer, 1999.

[72] A Kluth. Primates on facebook. *The Economist*, 2009.

[73] Nitish Korula and Silvio Lattanzi. An efficient reconciliation algorithm for social networks. *arXiv preprint arXiv:1307.1690*, 2013.

[74] G. Kossinets and D.J. Watts. Empirical analysis of an evolving social network. *Science*, 311(5757):88, 2006.

[75] Beate Krause, Christoph Schmitz, Andreas Hotho, and Gerd Stumme. The anti-social tagger: detecting spam in social bookmarking systems. In *Proceedings of the 4th international workshop on Adversarial information retrieval on the web*, pages 61–68. ACM, 2008.

[76] S. Kumar, R. Zafarani, and H. Liu. Understanding User Migration Patterns in Social Media. In *AAAI*, pages 1204–1209, 2011.

[77] Ho-Yu Lam and Dit-Yan Yeung. *A learning approach to spam detection based on social networks*. PhD thesis, HKUST, 2007.

[78] J. Leskovec, L. Backstrom, R. Kumar, and A. Tomkins. Microscopic evolution of social networks. In *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 462–470. ACM, 2008.

[79] Ming Li and Paul MB Vitányi. *An introduction to Kolmogorov complexity and its applications*. Springer, 2009.

[80] 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.

[81] J. Lin. Divergence Measures based on the Shannon Entropy. *IEEE Transaction on Information Theory*, 37(1):145–151, 1991.

[82] J. Liu, F. Zhang, X. Song, Y. Song, C. Lin, and H. Hon. What's in a Name?: An Unsupervised Approach to Link Users Across Communities. In *WSDM*, pages 495–504, 2013.

[83] Jing Liu, Fan Zhang, Xinying Song, Young-In Song, Chin-Yew Lin, and Hsiao-Wuen Hon. What's in a name?: an unsupervised approach to link users across communities. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 495–504. ACM, 2013.

[84] Eugene M Luks. Isomorphism of graphs of bounded valence can be tested in polynomial time. In *Foundations of Computer Science, 1980., 21st Annual Symposium on*, pages 42–49. IEEE, 1980.

[85] David Lusseau, Karsten Schneider, Oliver J Boisseau, Patti Haase, Elisabeth Slooten, and Steve M Dawson. The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations. *Behavioral Ecology and Sociobiology*, 54(4):396–405, 2003.

[86] Matteo Magnani and Luca Rossi. The ml-model for multi-layer social networks. In *ASONAM*, pages 5–12. IEEE, 2011.

[87] Vijay Mahajan and Robert A Peterson. *Models for innovation diffusion*, volume 48. Sage, 1985.

[88] Anshu Malhotra, Luam Totti, Wagner Meira Jr, Ponnurangam Kumaraguru, and Virgilio Almeida. Studying user footprints in different online social networks. In *Proceedings of the 2012 International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2012)*, pages 1065–1070. IEEE Computer Society, 2012.

[89] Benjamin Markines, Ciro Cattuto, and Filippo Menczer. Social spam detection. In *Proceedings of the 5th International Workshop on Adversarial Information Retrieval on the Web*, pages 41–48, 2009.

[90] MediaWiki. Combating Spam - List of Proxy and Spambot IPs. `bit.ly/1mwUqml`.

[91] Filippo Menczer. Web crawling. *Web Data Mining, Exploring Hyperlinks, Contents and Usage Data*, pages 273–321, 2007.

[92] Pasquale de Meo, Emilio Ferrara, Fabian Abel, Lora Aroyo, and Geert-Jan Houben. Analyzing user behavior across social sharing environments. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 5(1):14, 2013.

[93] G.A. Miller. The Magical Number Seven, Plus or Minus Two: Some Limits on our Capacity for Processing Information. *Psychological Review*, 63(2):81–97, 1956.

[94] Giovanna Miritello, Rubén Lara, Manuel Cebrian, and Esteban Moro. Limited communication capacity unveils strategies for human interaction. *Scientific reports*, 3, 2013.

[95] A. Mislove, M. Marcon, K.P. Gummadi, P. Druschel, and B. Bhattacharjee. Measurement and analysis of online social networks. In *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, page 42. ACM, 2007.

[96] A. Mislove, A. Post, P. Druschel, and K.P. Gummadi. Ostra: Leveraging trust to thwart unwanted communication. In *Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation*, pages 15–30. USENIX Association, 2008.

[97] Greg Mori and Jitendra Malik. Recognizing objects in adversarial clutter: Breaking a visual captcha. In *CVPR*, volume 1, pages I–134. IEEE, 2003.

[98] M. Müller. *Information Retrieval for Music and Motion*, volume 6. Springer Berlin, 2007.

[99] A. Narayanan and V. Shmatikov. Robust De-anonymization of Large Sparse Datasets. In *IEEE SSP*, pages 111–125, 2008.

[100] Mark Newman. *Networks: an introduction*. Oxford University Press, 2009.

[101] Mark EJ Newman. The structure and function of complex networks. *SIAM review*, 45(2):167–256, 2003.

[102] Mark EJ Newman. Modularity and community structure in networks. *PNAS*, 103(23):8577–8582, 2006.

[103] J. Novak, P. Raghavan, and A. Tomkins. Anti-Aliasing on the Web. In *WWW*, pages 30–39, 2004.

[104] Terri Oda and Tony White. Increasing the accuracy of a spam-detecting artificial immune system. In *CEC*, volume 1, pages 390–396. IEEE, 2003.

[105] A. Paivio. The Empirical Case for Dual Coding. *Imagery, Memory and Cognition*, pages 307–332, 1983.

[106] Daniele Perito, Claude Castelluccia, Mohamed Kaafar, and Pere Manils. How unique and traceable are usernames? In *PETS*, pages 1–17, 2011.

[107] Ashwin Rajadesingan, Reza Zafarani, and Huan Liu. Sarcasm detection on twitter: A behavioral modeling approach. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, WSDM '15, pages 97–106, New York, NY, USA, 2015. ACM.

[108] Everett M Rogers. *Diffusion of innovations*. Simon and Schuster, 2010.

[109] Jari Saramäki, EA Leicht, Eduardo López, Sam GB Roberts, Felix Reed-Tsochas, and Robin IM Dunbar. Persistence of social signatures in human communication. *Proceedings of the National Academy of Sciences*, 111(3):942–947, 2014.

[110] Andrew I Schein, Alexandrin Popescul, Lyle H Ungar, and David M Pennock. Methods and metrics for cold-start recommendations. In *Proceeding of SIGIR*, pages 253–260. ACM, 2002.

[111] Paulo Shakarian and Damon Paulo. Large social networks can be targeted for viral marketing with small seed sets. In *Proceedings of the 2012 International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2012)*, pages 1–8. IEEE Computer Society, 2012.

[112] Gianluca Stringhini, Christopher Kruegel, and Giovanni Vigna. Detecting spammers on social networks. In *ACSAC*, pages 1–9. ACM, 2010.

[113] Jie Tang, Tiancheng Lou, and Jon Kleinberg. Inferring social ties across heterogenous networks. In *Proceedings of the fifth ACM international conference on Web search and data mining*, pages 743–752. ACM, 2012.

[114] L. Tang, H. Liu, and J. Zhang. Identifying Evolving Groups in Dynamic Multi-Mode Networks. *Transactions on Knowledge and Data Engineering, to appear.*

[115] Lei Tang, Xufei Wang, and Huan Liu. Community detection via heterogeneous interaction analysis. *Data Mining and Knowledge Discovery*, pages 1–33, 2012.

[116] Alan Taylor and Desmond J Higham. Contest: A controllable test matrix toolbox for matlab. *ACM Transactions on Mathematical Software (TOMS)*, 35(4):26, 2009.

[117] N. Tran, B. Min, J. Li, and L. Subramanian. Sybil-resilient online content voting. In *Proceedings of the 6th USENIX symposium on Networked systems design and implementation*, pages 15–28. USENIX Association, 2009.

[118] Nguyen Tran, Jinyang Li, Lakshminarayanan Subramanian, and Sherman SM Chow. Optimal sybil-resilient node admission control. In *INFOCOM*, pages 3218–3226, 2011.

[119] Johan Ugander, Brian Karrer, Lars Backstrom, and Cameron Marlow. The anatomy of the facebook social graph. *arXiv preprint arXiv:1111.4503*, 2011.

[120] Bimal Viswanath, Ansley Post, Krishna P Gummadi, and Alan Mislove. An analysis of social network-based sybil defenses. *ACM SIGCOMM Computer Communication Review*, 41(4):363–374, 2011.

[121] Luis Von Ahn, Manuel Blum, Nicholas J Hopper, and John Langford. Captcha: Using hard ai problems for security. In *EUROCRYPT 2003*, pages 294–311. Springer, 2003.

[122] Alex Hai Wang. Don't follow me: Spam detection in twitter. In *Security and Cryptography (SECRYPT)*, pages 1–10. IEEE, 2010.

[123] T. Wasserman. 83 Million Facebook Accounts Are Fake. `http://on.mash.to/1hdze2B`.

[124] D.J. Watts and S.H. Strogatz. Collective dynamics of small-worldnetworks. *nature*, 393(6684):440–442, 1998.

[125] Wikipedia. Keyboard Layouts. `http://bit.ly/kXso`.

[126] Wikipedia. List of countries by population. `http://bit.ly/1eTTUHe`.

[127] Gregory L Wittel and Shyhtsun Felix Wu. On attacking statistical spam filters. In *CEAS*, 2004.

[128] Yinglian Xie, Fang Yu, Kannan Achan, Rina Panigrahy, Geoff Hulten, and Ivan Osipkov. Spamming botnets: signatures and characteristics. *ACM SIGCOMM Computer Communication Review*, 38(4):171–182, 2008.

[129] J. Yan, A. Blackwell, R. Anderson, and A. Grant. The Memorability and Security of Passwords-Some Empirical Results. *U. of Cambridge Tech. Rep.*, 2000.

[130] Jeff Yan and Ahmad Salah El Ahmad. A low-cost attack on a microsoft captcha. In *Proceedings of the 15th ACM conference on Computer and communications security*, pages 543–554. ACM, 2008.

[131] Zhi Yang, Christo Wilson, Xiao Wang, Tingting Gao, Ben Y Zhao, and Yafei Dai. Uncovering social network sybils in the wild. In *IMC*, pages 259–268. ACM, 2011.

[132] Sarita Yardi, Daniel Romero, Grant Schoenebeck, et al. Detecting spam in a twitter network. *First Monday*, 15(1), 2009.

[133] H. Yu, M. Kaminsky, P.B. Gibbons, and A. Flaxman. Sybilguard: defending against sybil attacks via social networks. In *Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 267–278. ACM, 2006.

[134] Haifeng Yu, Phillip B Gibbons, Michael Kaminsky, and Feng Xiao. Sybillimit: A near-optimal social network defense against sybil attacks. In *Security and Privacy, 2008. SP 2008. IEEE Symposium on*, pages 3–17. IEEE, 2008.

[135] Wayne W Zachary. An information flow model for conflict and fission in small groups. *Journal of anthropological research*, pages 452–473, 1977.

[136] R. Zafarani and H. Liu. Connecting Corresponding Identities across Communities. In *ICWSM*, pages 354–357, 2009.

[137] R. Zafarani and H. Liu. Social computing data repository at ASU. *School of Computing, Informatics and Decision Systems Engineering, Arizona State University*, 2009.

[138] Reza Zafarani, Mohammad Ali Abbasi, and Huan Liu. *Social Media Mining: An Introduction*. Cambridge University Press, 2014.

[139] Reza Zafarani and Huan Liu. Connecting users across social media sites: a behavioral-modeling approach. In *SIGKDD*, pages 41–49. ACM, 2013.

[140] Reza Zafarani and Huan Liu. Finding friends on a new site using minimum information. In *SDM*. SIAM, 2014.

[141] Reza Zafarani and Huan Liu. Users joining multiple sites: Distributions and patterns. In *Eighth International AAAI Conference on Weblogs and Social Media*, 2014.

[142] Reza Zafarani and Huan Liu. 10 bits of surprise: Detecting malicious users with minimum information. In *Proceedings of the 24th ACM international conference on Information and knowledge management*. ACM, 2015.

[143] Reza Zafarani and Huan Liu. Evaluation without ground truth in social media research. *Communications of the ACM*, 58(6), 2015.

[144] Reza Zafarani and Huan Liu. Users joining multiple sites: Friendship and popularity variations across sites. *Information Fusion*, 2015.

[145] Reza Zafarani, Lei Tang, and Huan Liu. User identification across social media. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 10, 2015.

[146] Jiawei Zhang, Xiangnan Kong, and Philip S Yu. Transferring heterogeneous links across location-based social networks. In *Proceedings of the 7th ACM international conference on Web search and data mining*, pages 303–312. ACM, 2014.

[147] R. Zheng, J. Li, H. Chen, and Z. Huang. A Framework for Authorship Identification of Online Messages: Writing-style Features and Classification Techniques. *JASIST*, 57(3):378–393, 2006.

[148] X. Zhu. Semi-supervised learning literature survey (technical report 1530). *Computer Sciences, University of Wisconsin-Madison*, 2005.