Will We Connect Again? Machine Learning for Link Prediction in Mobile Social Networks

Ole J. Mengshoel, Raj Desai, Andrew Chen, Brian Tran
Carnegie Mellon University
NASA Ames Research Park
Moffett Field, CA 94035

ABSTRACT

In this paper we examine link prediction for two types of data sets with mobility data, namely call data records (from the MIT Reality Mining project) and location-based social networking data (from the companies Gowalla and Brightkite). These data sets contain location information, which we incorporate in the features used for prediction. We also examine different strategies for data cleaning, in particular thresholding based on the amount of social interaction. We investigate the machine learning algorithms Decision Tree, Naïve Bayes, Support Vector Machine, and Logistic Regression. Generally, we find that our feature selection and filtering of the data sets have a major impact on the accuracy of link prediction, both for Reality Mining and Gowalla. Experimentally, the Decision Tree and Logistic Regression classifiers performed best.

Categories and Subject Descriptors

G.3 [**Probability and Statistics**]: Statistical Computing; J.4 [**Social and Behavioral Sciences, Mobile Applications**]: Sociology, Location-dependent and sensitive

General Terms

Algorithms, Measurement, Experimentation

Keywords

Call Data, Mobility, Location-based Social Networks, Link Prediction, Supervised Machine Learning, Data Cleaning

1. INTRODUCTION

Given the recent growth and popularity of large-scale online social networks including Facebook, Foursquare, LinkedIn, and Twitter, social network analysis is becoming an important area of research. In addition, telecommunication operators collect call data records (CDRs) that can be used to produce large social call-graphs. In these graphs, an edge typically represents a social interaction in the form of a phone call between two people.

The definition of a link or a social tie depends on the type of network being modelled. For social networks generated from CDR data [15, 6, 17], one can define a link as a relation between people calling each other. In online social networks (OSNs), a link can be called out explicitly (by declaring friends and followers) or implicitly by considering the spatio-temporal coordinates of people, as is done in location-based services. For example, two people may be checking in to the same online service at the same location at (approximately) the same time; they may then be two friends visiting a restaurant.

Previous research has investigated the mobility patterns of users in social networks, based on their check-in data [4]. Other prominent research has investigated community and social network formation [15, 6, 7] as well as link prediction [1, 13, 12, 18, 5, 3, 2, 17]. Link prediction using supervised machine learning has several potential real world applications, including friend recommendation system in social networks, product recommendation for e-commerce [3], and suspect identification in antiterrorism.

Machine learning (ML) algorithms may seem, in theory, straightforward to apply to the problem of link prediction in social networks. However, there is typically a non-trivial amount of preprocessing that needs to be done on data sets "in the wild" in order to optimize predictions. With "preprocessing," we refer to issues such as optimizing the feature set and determining which data records to include in (or "thresholding") the training set. One reason why these issues are important is the difference between "social tie" and "contact": Calling the phone wrong number, or randomly being in the same place as someone else at the same time, does not mean there is a social tie. Despite the existing literature on social network link prediction by means of ML [19, 1, 13, 18, 5, 3, 17], there is with some notable exceptions [5, 17] little work on ML for link prediction "in the wild," in particular when mobility data is involved.

In this paper, we focus on link prediction in the sense of predicting a social tie in the future, given that the set of users already share a tie. We carefully examine the effects of applying data set thresholding and cleanup on the ML task of link prediction in social networks with location data. Specifically, we apply the classification techniques Decision Tree, Logistic Regression, Naïve Bayes, and Support Vector Machine. Few previous efforts carefully discuss their use of thresholds for eliminating noisy data, and most previous papers discussing similar issues do not consider location data [1, 13, 5], which play a key role in understanding social networks. At the same time, previous papers on mobility and location data often emphasize data analysis [6, 15] rather than ML for link prediction, which is our focus.

The rest of this paper is organized as follows. In Section 2, we discuss related research. Section 3 presents the data sets Brightkite, Gowalla, and Reality Mining. In Section 4 we discuss our link prediction approach. Section 5 reports on link prediction experimental

Eleventh Workshop on Mining and Learning with Graphs. Chicago, Illinois, USA

results using Decision Tree, Logistic Regression, Naïve Bayes, and Support Vector Machine, while Section 6 concludes and outlines future research.

2. RELATED RESEARCH

In recent years, there has been significant research related to link prediction in social networks. Some of the previous research focuses on data analysis rather than data mining or ML; it is still relevant since the features investigated can be useful when a classifier or regression model is constructed.

2.1 Data Analysis

Using data from a large mobile operator in India, Nanavati et al. [15] investigated the graph properties of the contact graph induced by call and SMS data records. Eagle et al. [6] investigated a small country data set, consisting of 1.4 million subscriber CDRs, with an interest in understanding behavior in rural versus urban communities. One of their main findings is that in cities there were a large number of short calls, while rural areas had a much greater proportion of few but long calls. Evans et al. [8] use two relatively small data sets to perform community detection, namely the Karate Club and the University of South Florida Free Association Norms data sets. Specifically, they use a random walk graph partition method to find link communities.

Cho et al. have investigated the mobility patterns of users in social networks, based on their check-in data [4]. In particular, long-distance mobility was found to be associated with social ties across the data sets. Motahari et al. [14] study how social ties, as reflected in phone calls, can be classified into different affinity networks. Example affinity networks are family networks, business networks, networks of friends, etc. The authors analyzed 4.3 million CDRs produced by 360,000 subscribers in two California cities, San Francisco and Modesto. They found across the two cities significant similarities, with differences between affinity networks in terms of the features generated from the CDRs. Specific features that model statistically meaningful changes in call patterns, and are thus useful for prediction and classification, were identified.

Compared to previous data analysis research as discussed above, this paper performs supervised ML for link prediction. In addition, our focus on link prediction in a mobile context is different from previous emphasis on mobility patterns [14], community detection [6, 8], or global properties of a social graph [15].

2.2 Link Prediction using Machine Learning

Liben-Nowell and Kleinberg define the link prediction problem as: "given a snapshot of a social network, can we infer which new interactions among its members are likely to occur in the near future?" [13]. Using co-authorship social networks, this seminal work investigates topological features for link prediction. The predictors used are Adamic/Adar, low-rank inner product, weighted Katz, common neighbors, Katz clustering, rooted PageRank, Jaccard, SimRank, unseen bigrams, and hitting time. The authors found that all predictors almost always out-performed a baseline random prediction on all data sets.

Introducing the Profiles in Terror (PIT) data set, along with features related to terrorists and terrorist events, Zhao et el. [19] explored the use of relational Markov networks (RMNs) for labelling (or classification). In particular, the authors performed entity and relationship labelling in affiliation networks.[1]

In order to investigate link prediction in the context of human mobility, a data set containing information for 6 million mobile phone users over three months was investigated [17]. The data set contained 90 million communication records, and over 10k distinct locations covering a radius of more than 1,000 km. Features including co-location, network proximity, and tie strength are explored. The ML algorithms used are Decision Tree, Random Forest, SVM, and Logistic Regression. The authors are learning the shape of mobility patterns and their impact on social networks.

Eagle et al. [7] introduced and analyzed the Reality Mining data set, which was collected from 94 subjects equipped with cell phones with Bluetooth. In particular, they compared observational (as reflected in CDRs, Bluetooth logs, etc.) and self-report data, and found that 95% of the self-reported friendships could be predicted from the observational data alone.[2]

Benchettara et al. [3] explored two data sets, namely a co-author social network created from the DBLP bibliographic database and eight years of transactional data from an on-line store. The authors used pruned Decision Trees (J48) with boosting, and found that adding topological attributes to the feature set significantly improved precision, recall, and F-measure for link prediction.

Using data on co-authorship, with 1 million+ papers and 1 million+ authors, Hasan et al. [1] predicted the probability of an unobserved future link occurring. The authors investigated a broad range of features, including proximity as measured by number of overlapping keywords; aggregated features including sum of papers, sum of neighbors, and sum of keyword counts; and topological features such as shortest distance [11] and clustering index [16]. Several supervised classification methods, specifically SVM, Decision Tree, Multilayer Perceptron, k-NN, Naïve Bayes, and RBF network, were tested, with and without bagging. Overall, SVM outperformed the other classification approaches.

How is our work different from previous research on link prediction using supervised ML? First, many authors [1] do not discuss data preprocessing and clean-up. In fact, many of the data sets used previously probably require far less clean-up than the mobility data sets considered in this paper. Second, much previous research has been on predicting new links [1, 13], whereas we investigate prediction for on-going links. Third, this paper is at the intersection of mobility and social networks, while much previous work has been concerned with one or the other. In particular, much previous work on link prediction has targeted social networks in the form of bibliographic author-author networks [1, 13] without any geographic features. Clearly, the advent of mobile devices with GPS and social network services with location check-in has enabled ML research that integrates location and social data for link prediction.

3. SOCIAL NETWORK DATA SETS

We studied data from three publicly available data sets: Reality Mining,[3] Gowalla,[4] and Brightkite.[5]

3.1 Reality Mining

[1] The meaning of the term "affiliation network" is somewhat broader in the work of Zhao et el. [19] compared to the work of Motahari et al. [14]. For Zhao et el., the term refers to both links between actors and events in a bipartite graph (actor-event links) as well as links between actors in a unipartite graph, induced by a bipartite graph, of actors (actor-actor links). For Motahari et al., the term is based on classifying actor-actor links without considering the associated events.

[2] This result bears some similarity to our results for the Gowalla data set, see Section 5.3.

[3] See http://reality.media.mit.edu.

[4] See http://gowalla.com/ and http://snap.stanford.edu/data/.

[5] See http://snap.stanford.edu/data/.

Metric	Reality Mining
Mobile phone call records	109,344
Distinct participants in study	94
Distinct people in call data	10,056
Min number of calls per participant	3
Max number of calls per participant	4,674
Median number of calls per participant	922
Unique cell tower IDs	3,137
Average number of cell towers, per participant	1,023

Table 1: Metrics for the data set from the mobile phone (with Bluetooth) user study Reality Mining.

Metric	Gowalla	Brightkite
Number of check-ins	6,442,144	4,702,067
Number of users	196,591	58,228
Number of users with ≥ 1 check-ins	107,092	51,406
Median number of check-ins per user	25	11
Average number of check-ins by user	33	81
Max number of check-ins per user	2,175	2,100
Number of edges, friendship graph	950,327	214,078
Median number of friends per user	3	2
Average number of friends per user	10	7
Max number of friends per user	14,730	1,134
Unique locations	1,280,957	772,923
Median check-ins per location	2	1
Max check-ins per location	5,811	249,934

Table 2: Metrics for the data set from the mobility-oriented social network services Gowalla and Brightkite.

Reality Mining was a study in which tracking software was installed on the cellphones of subjects [7]. The mobile phone activity of 94 subjects was tracked over a period of nine months from 2004 to 2005. Reality Mining data of interest, see Table 1, includes call data records (CDRs) in which each record contains information about caller, recipient, timestamp, and duration of the call. The data set also contains information about a phone's connection to cell phone towers and other Bluetooth devices. Since the locations of cell towers are known to telecom operators, interesting mobility data can be mined from CDRs, not only traffic patterns [9, 4] but also physical distance between cell phone users.

Bluetooth data, which is part of Reality Mining, gives collocation data, and is similar to an OSN in which users check in within some time interval. Bluetooth gives a stronger indication of close proximity than do OSNs, since users need to explicitly report their locations in OSNs. In some sense, Bluetooth data is also similar to phone calls. Of course, there is a difference in that Bluetooth connects by proximity and not by explicit calling.

3.2 Gowalla and Brightkite

Gowalla and Brightkite, see Table 2, are retired OSN services very similar to Foursquare. These are OSNs in which users check in to report visits to specific physical locations. The check-in logs for both Brightkite and Gowalla include user, location, latitude, longitude, and time. These location-based services also provide a social friendship graph for users, discussed further in Section 3.3.

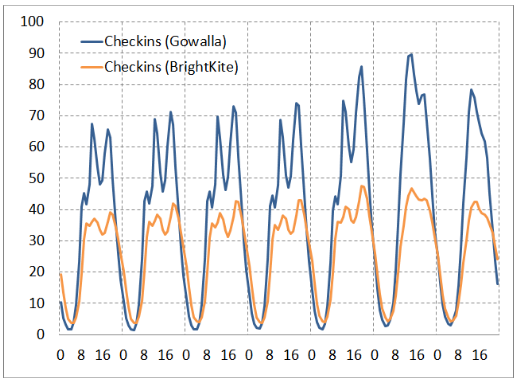


Figure 1: Weekly check-in data, collected across several weeks, for Brightkite and Gowalla. The x-axis shows time, with labels for every eight hours, while the y-axis shows the number of check-ins, in thousands.

Figure 1 shows "average" data for a week, from Monday to Sunday, created by taking data from Brightkite and Gowalla for multiple weeks and normalizing across time zones. For both data sets, there is high activity during the day, with clear dips in activity during the night. Second, there is clear similarity between the curves for Brightkite and Gowalla. For weekdays (except Fridays), there appears to be a breakfast, lunch, and dinner pattern with higher activity. There is a slighty different, and higher-activity, pattern for weekends. Weekend starts with dinner on Friday and tapers off with dinner on Sunday. There are also some differences between these two data sets. First, the number of users is smaller for Brighkite than for Gowalla. Also, there are some slight time shifts, for example in the points of minimal activity. In the rest of this paper, we study Gowalla closely, with the hope that our results will carry over to other similar location-oriented data sets from Brightkite, Facebook, or Foursquare.

3.3 Social Graphs

To distinguish between different connection types, while acknowledging their similarities, we introduce the following terminology. At the most general level we consider *social graphs*. We distinguish between different types of *social graphs*, namely *contact graphs (CGs)* and *friendship graphs (FGs)*. A friendship graph is induced by "friends" and "followers" found in OSNs, while similar information is typically not found in CDR data sets.

Contact graphs can be explicit or implicit. Phone calls induce *explicit contact graphs (ECGs)* with phone calls as edges. Location data (without phone calls) induce *implicit contact graphs (ICGs)*. From Reality Mining both implicit and explicit contact graphs can be created, while from Brightkite and Gowalla only implicit contact graphs can be induced. Both Brightkite and Gowalla come with a friendship graph.

Even without a friendship graph, and only an implicit contact graph, social ties are reflected in data. For location-based social networks, we define a social tie when two users check-in simultaneously within a defined (and short) time interval at the same location. Such a two-user check-in is treated as a virtual phone call, and if there is more than two users it is treated similar to a conference call. The potential social contact between two or more such users, together with the relevant time and location information, can be thought of as a social data record (SDR), similar to a CDR from a telecommunications network.

4. LINK PREDICTION METHOD

In this section, the problem of link prediction is defined as predicting future association between a pair of nodes, such as a social tie between two users, based on the past association between those users. For link prediction, we use established classifiers: Decision Tree, Naïve Bayes, Logistic Regression, and SVM (see also previous research [2, 1, 17]). Generally, a major ML challenge is to develop an effective feature set. Below, we define features that we

Feature Type	**Notation**	**General Feature**	**Experimental Feature(s)**
Features per user	$\phi_{\mathrm{nc}}(u,t_b,t_e)$	Total calls in time interval from t_b to t_e	$[t_b,t_e)$ = past 3 months
	$\phi_{\mathrm{acd}}(u,t_b,t_e)$	Average duration of all u calls, during $[t_b,t_e)$	$[t_b,t_e) = [0,\infty)$
	$\phi_{\mathrm{ne}}(u)$	Number of neighbors in ECG	
Features per pair	$\phi_{\mathrm{nc}}(u,v,t_b,t_e)$	Total u-v calls during time interval $[t_b,t_e)$	$[t_b,t_e)$ = past 3 month-long intervals
	$\phi_{\mathrm{min}}(u,v,t_b,t_e)$	Minimum number of calls during interval $[t_b,t_e)$	$[t_b,t_e) = 1$ month
	$\phi_{\mathrm{max}}(u,v,t_b,t_e)$	Maximum number of calls during interval $[t_b,t_e)$	$[t_b,t_e) = 1$ month
	$\phi_{\mathrm{acd}}(u,v,t_b,t_e)$	Average duration of u-v calls during $[t_b,t_e)$	
	$\phi_{\mathrm{cn}}(u,v)$	Number of common neighbors for u and v	
	$\phi_{\mathrm{tc}}(u,v,t_b,t_e)$	*Time, during u-v calls, they connected to same tower*	
	$\phi_{\mathrm{sc}}(u,v,t_b,t_e)$	*Number of u-v Bluetooth scans during* $[t_b,t_e)$	
Label	$\kappa_c(u,v)$	$\geq \tau_c$ u-v calls are placed during $[t_b,t_e)$	$[t_b,t_e) = 1$ month

Table 3: Reality Mining features and prediction task; features that directly capture the physical location of the user(s) are highlighted.

hope capture key aspects of the connections between and mobility of users, both for location-based social network services and for cell phones. ML results, using these features, are discussed in Section 5.

4.1 Link Prediction and Social Tie

In this paper, our goal is to predict future links given data on social activity. For the Reality Mining data set, the link was defined as a pair of people calling one another, i.e. an edge in an ECG. To predict future calls between two users, features presented in Table 1 were extracted.

A key issue related to link prediction is that of a social tie, including its its nature and strength [14]. A pair of users shares a strong social tie when they interact with each other consistently, frequently, and bilaterally. A pair shares a weak tie when they interact accidently, e.g. wrong number in a mobile network, and unilaterally.[6] Finally, no contact (which is most common) indicates no social tie. In this paper, we explore the effects of the strength of social tie on the link prediction accuracy.

We also define a link as "collocation" when two users check in to a location within the span of a specified time interval. We take this as a sign of a potential social tie or contact. We effectively used these mobility features in our prediction models to boost accuracy.

4.2 Raw Data

4.2.1 Records of Reality Mining Data Set

Definition. A call data record (CDR) represents a call and can be defined as a tuple $\theta = (u,v,t,d)$,[7] where u and v are subscribers, t is the call start time, and d is the call duration. The order of u and v in θ is meaningful: u is the subscriber initiating the call while v receives the call. The set of all CDRs in a data set is denoted Φ. Among all calls between u and v, the outgoing calls from u are defined as $\Phi_u(u,v) = \{(u,v,t,d) \mid (u,v,t,d) \in \Phi\}$, while the outgoing calls from v are defined as $\Phi_v(u,v) = \{(u,v,t,d) \mid (v,u,t,d) \in \Phi\}$. A CDR relation with all calls between u and v can now be defined as $\Phi(u,v) = \Phi_u(u,v) \cup \Phi_v(u,v)$. Below, in our features, we are generally using $\Phi(u,v)$ and saying "call between u and v," or u-v call, since call direction is not emphasized here. The set of such calls during a time interval $[t_b,t_e)$ is defined as:

$$\Phi(u,v,t_b,t_e) = \{(u,v,t,d) | (u,v,t,d) \in \Phi(u,v) \wedge t \in [t_b,t_e)\}.$$

4.2.2 Records of Gowalla Data Set

Definition. A location data record (LDR) represents a check-in and can be defined as a tuple $\theta = (u,t,\ell)$, where u is a subscriber, t is the check-in time, and ℓ is the location. There is always a location associated with a check-in, since in cases where there is not, a dummy location can be introduced.

Similar to for CDRs, we define a data set of LDRs, and subsets thereof.

4.2.3 Social Graphs

Let $G = (V,E)$ be an undirected graph where V are nodes and E represents edges $\{u,v\}$, where $u,v \in V$. G can represent the different social graphs introduced above; how it is constructed varies.

For Gowalla and Brightkite, G can represent the friendship graph provided in the data sets. There is no temporal dimension to the friendship graphs (i.e., there are no time stamps indicating when an edge was introduced or traversed).

For explicit contact graphs, we start with CDRs as found for Reality Mining, and the nodes V represents the set of all subscribers in our sample while edges E are all contacts $|\Phi(u,v)| \geq 1$.

For location-oriented OSNs, implicit contact graphs can be created. We start with LDRs from which SDRs can be induced, leading to creation of G similar to what is done for CDRs.

To summarize, we discuss social graphs for all three data sets, and note that the meaning of "social graph" and how it is created can vary somewhat. However, in this paper the social graph is always considered to be undirected.

4.3 Machine Learning Features

The raw data records and social graphs, discussed in Section 4.2, were transformed into features more suitable for ML algorithms, as we now discuss. Our goal is to characterize the nature of the social tie between u and v through the distribution of these features. Some of the features are defined for one subscriber $u \in V$, while other features are defined for a pair of subscribers, $\{u,v\} \in E$.

We now consider Reality Mining features; see also Table 3. The total number of calls ϕ_{nc} is defined as $\phi_{\mathrm{nc}} = |\Phi|$.

The total number of u-v calls ϕ_{nc}, during the time interval $[t_b,t_e)$, is defined as:

$$\phi_{\mathrm{nc}}(u,v,t_b,t_e) = |\Phi(u,v,t_b,t_e)|\,.$$

Here, t_b and t_e are set such that the number of calls between pairs of users in the past month, say, is counted. We define $\phi_{\mathrm{nc}}(u,v) = \phi_{\mathrm{nc}}(u,v,0,\infty)$, assuming that time starts at $t_b = 0$.

Total call time ϕ_{tot} for u-v calls is defined as:

$$\phi_{\mathrm{tot}}(u,v,t_b,t_e) = \sum_{(u,v,t,d) \in \Phi(u,v,t_b,t_e)} d\,.$$

[6] An example of a unilateral relationship is reflected in the saying of the Hollywood agent: "Don't call me, I'll call you. "

[7] The raw CDR format has additional fields (giving a total, typically, of 10-20) that we do not consider in this paper.

Feature Type	Notation	General Feature	Experimental Feature(s)
Features per user	$\phi_{\text{ci}}(u,t_b,t_e)$	Total number of check-ins during $[t_b,t_e)$	$[t_b,t_e)$ = past $[0,2)$, $[2,4)$, $[4,6)$ months
	$\phi_{\text{dci}}(u,t_b,t_e)$	*Number of distinct location check-ins during* $[t_b,t_e)$	
	$\phi_{\text{ne}}(u)$	Number of neighbors (friends) in FG	
Features per pair	$\phi_{\text{cn}}(u,v)$	Number of common (mutual) neighbors (friends) in FG	
	$\phi_{\text{co}}(u,v,t_b,t_e)$	*Number of collocations during* $[t_b,t_e)$	$[t_b,t_e)$ = past $[0,2)$, $[2,4)$, $[4,6)$ months
	$\phi_{\text{dco}}(u,v,t_b,t_e)$	*Number of distinct collocation locations*	
	$\phi_{\text{dlo}}(u,v,t_b,t_e)$	*Number of distinct (common) locations*	
Label	$\kappa_f(u,v)$	Pair is linked in FG	
	$\kappa_l(u,v)$	*Pair collocates during* $[t_b,t_e)$	$[t_b,t_e)$ = next 2 months

Table 4: Gowalla features and prediction task; features that directly capture the physical location of the user(s) are highlighted.

Average call duration ϕ_{acd} for u-v calls is defined as:

$$\phi_{\text{acd}}(u,v,t_b,t_e) = \frac{\phi_{\text{tot}}(u,v,t_b,t_e)}{\phi_{\text{nc}}(u,v,t_b,t_e)}.$$

Average call duration for a given u, with respect to all v, $\phi_{\text{acd}}(u)$, is defined in a way similar to $\phi_{\text{acd}}(u,v)$.

Let T be a set of time intervals (we used month-long time intervals in experiments). The feature ϕ_{min} captures the minimum number of u-v calls and is defined as follows:

$$\phi_{\text{min}}(u,v,T) = \min\left(\{\phi_{\text{nc}}(u,v,t_b,t_e) \mid [t_b,t_e) \in T\}\right);$$

the maximum $\phi_{\text{max}}(u,v,T)$ is defined in a similar way.

Let $G = (V,E)$ be a social graph; u's number of neighbors is

$$\phi_{\text{ne}}(u) = |\{w \mid \{u,w\} \in E\}|,$$

while the number of common neighbors of u and v is

$$\phi_{\text{cn}}(u,v) = |\{w \mid \{u,w\} \in E \ \wedge \{v,w\} \in E\}|.$$

The ϕ_{tc} feature refers to the time while two subscribers were connected to the same cell phone tower during a time interval. Both need to be on a call, but not necessarily with each other. An implication of this situation is that the subscribers are physically close.

The ϕ_{sc} feature refers to the number of seconds two Bluetooth devices were in close proximity, during a time interval, based on periodic scanning. Unlike ϕ_{tc}, it does not require calls to be made.

Now for the location data records (LDRs) and friendship graph found, for example, in Gowalla. The total number of check-ins ϕ_{ci} for u, during $[t_b,t_e)$, is:

$$\phi_{\text{ci}}(u,t_b,t_e) = |\{(u,t,\ell) \mid (u,t,\ell) \in \Phi(u) \wedge t \in [t_b,t_e)\}|.$$

The number of distinct location check-ins ϕ_{dci}, during $[t_b,t_e)$, is defined as:

$$\phi_{\text{dci}}(u,t_b,t_e) = |\{\ell \mid (u,t,\ell) \in \Phi(u) \wedge t \in [t_b,t_e)\}|.$$

The set of (u,t,ℓ)-tuples found during $[t_b,t_e)$ is defined as:

$$\Phi(u,\ell,t_b,t_e) = \{(u,t,\ell) | (u,t,\ell) \in \Phi(u) \wedge t \in [t_b,t_e)\}.$$

The set of collocations Φ_{co} for u and v during $[t_b,t_e)$, with a temporal threshold of $\tau_t \in \mathbb{R}$, is defined as:

$$\Phi_{\text{co}}(u,v,t_b,t_e,\tau_t) = \{(u,v,t_u,t_v,\ell) | (u,t_u,\ell) \in \Phi(u,\ell,t_b,t_e) \wedge (v,t_v,\ell) \in \Phi(v,\ell,t_b,t_e) \wedge |t_u - t_v| \leq \tau_t\}.$$

The number of collocations ϕ_{co}, assuming the threshold τ_t, is now simply:

$$\phi_{\text{co}}(u,v,t_b,t_e,\tau_t) = |\Phi_{\text{co}}(u,v,t_b,t_e,\tau_t)|.$$

The number of distinct collocations locations ϕ_{dco} for u and v during $[t_b,t_e)$ is defined as:

$$\phi_{\text{dco}}(u,v,t_b,t_e,\tau_t) = |\{\ell \mid (u,v,t_u,t_v,\ell) \in \Phi_{\text{co}}(u,v,t_b,t_e,\tau_t)\}|$$

The number of distinct common locations ϕ_{dlo} is:

$$\phi_{\text{dlo}}(u,v,t_b,t_e) = |\{\ell \mid (u,t_u,\ell) \in \Phi(u) \wedge t_u \in [t_b,t_e)\} \cap \{\ell \mid (v,t_v,\ell) \in \Phi(v) \wedge t_v \in [t_b,t_e)\}$$

The difference between ϕ_{co}, ϕ_{dco}, and ϕ_{dlo} is as follows. The ϕ_{co} feature expresses the number of times u and v visited some location at the same time. If u and v only meet daily for a week at one particular restaurant, this gives $\phi_{\text{dco}} = 7$. ϕ_{dco} expresses the number of distinct common locations visited at the same time. In other words, if u and v only meet daily at one particular restaurant, this gives $\phi_{\text{dco}} = 1$. Finally, ϕ_{dlo} expresses the number of common locations for u and v, not necessarily visited at the same time.

The features extracted from a social graph for the purpose of link prediction can be categorized as follows. First, *topological* features are global properties of a social graph, such as power law degree distribution [15], small world phenomenon [11], or other structural features [13]. Second, *local* features only focus on the local properties of nodes and edges [14]. Third, there are *content* features pertaining to the detailed contents of a node or an edge [2].

This paper mainly uses local features of social graphs, as discussed above, for link prediction. Employing these local features is computationally efficient, which is important due to the large scale and very dynamic nature of such networks in industrial applications. Content features, on the other hand, are often harder to come by compared to local features, and may also be subject to privacy restrictions and concerns.

In the Gowalla features, far more emphasis is placed on location and the more reliable friendship graph compared to what is the case in Reality Mining. Within the Gowalla data set, there was sufficient data to perform two distinct link prediction tasks. The first link, denoted $\kappa_f(u,v)$ in Table 4, was whether two Gowalla users listed each other as friends in the friendship graph. The second link, denoted $\kappa_l(u,v)$ in Table 4, was whether two users have a collocation. In other words, do they simultaneously check in at the same location? For two check-ins by different users to qualify as a collocation, both check-ins must have been performed within τ_t minutes of one another. The same features were used for both prediction tasks.

4.4 Machine Learning Process

Features: Feature construction and selection are critical steps for obtaining quality results when applying ML algorithms. We started with an initial feature set, experimented with an ML algorithm, say Decision Tree, observed the classification quality, and iteratively refined the feature set in order to optimize the classifier.

Each feature was ranked and evaluated individually for information gain and correlation. For example, several features were combined as they effectively represented very similar information. In subsequent iterations, features were added or deleted.

In the end, the ML algorithms were applied to a subset of the features initially constructed. The details of the final feature sets are captured in Table 3 and Table 4. For SVM, we also examined the effect of feature value normalization (see Section 5).

Examples: For the purpose of supervised ML, positive D^+ and negative D^- examples are used. Let us consider a time interval $\mathbb{T}$ as a subset of the real line $\mathbb{R}$, or $\mathbb{T} \subseteq \mathbb{R}$. We use two time intervals, one for learning $\mathbb{T}_L \subseteq \mathbb{R}$ and one for testing $\mathbb{T}_T \subseteq \mathbb{R}$, with $\mathbb{T}_L \cap \mathbb{T}_T = \emptyset$, $\mathbb{T}_L = [t_1, t_2)$, $\mathbb{T}_T = [t_3, t_4)$, and $t_3 \geq t_2$. Altogether, this gives positive examples for learning D_L^+; negative examples for learning D_L^-; positive examples for testing D_T^+; and negative examples for testing D_T^-.

In this paper, we use $\mathbb{T}_L = [0,3)$ and $\mathbb{T}_T = [3,4)$ for Reality Mining and $\mathbb{T}_L = [0,6)$ and $\mathbb{T}_T = [6,8)$ for Gowalla. All these time intervals are measured in months. Table 3 and Table 4 contain further details.

The number of all possible links is, in any social network of nontrivial size, far greater than the number of observed links. To reflect this, we consider two data sets: observed links D_O and unobserved links D_U. The data set D_O contains pairs of nodes in which one or more contacts have occurred between nodes during $\mathbb{T}_L$. In principle, the data set D_U may contain all the remaining links in the graph. To mitigate the class skew issue [2], instead of adding all the links from D_U to D_L^-, one may do the following: take a small random subset from D_U, comparable in size to $|D_O|$.

Thresholding: For Reality Mining and similar data sets one single call may not indicate a social tie. There are scenarios such as calling a wrong number or a single service call, indicating no or a minimal social tie between two callers. To more confidently infer a social tie between a pair of callers, they need to call multiple times. A similar argument can be made for location-oriented OSNs.

To formalize the above intuitions, we consider two thresholds, namely a contact threshold τ_c and a time threshold τ_t. The contact threshold $\tau_c \in \mathbb{N}$ is associated with the feature $\phi_{\min}$. If

$$\phi_{\min}(u, v, T) \geq \tau_c$$

then the u-v contact passes the threshold. It is reasonable to believe that the minimum number of u-v calls during a month, $\phi_{\min}$, is a good indicator of social tie strength between u and v. The threshold τ_c determines whether an example is placed in D^+ or in D^-. Consequently, $|D^+|$ decreases with τ_c increasing, while $|D^-|$ increases with τ_c increasing.

The time threshold τ_t is part of the definition of the feature ϕ_{co}. This threshold is useful for services like Brightkite and Gowalla, where social connections cannot easily be observed on a large scale.

For Reality Mining and Gowalla we investigated varying τ_c and τ_t, reflecting varying strength of the underlying social tie, and learned different classifiers as reported in Section 5.

5. EXPERIMENTAL RESULTS

5.1 Software Tools and Methods

Given that for Gowalla there were a large number of links for which we needed to calculate features, Apache Hadoop[8] was employed for feature construction. We used Apache Hive[9] to run our jobs on a Hadoop cluster. Hive provides a SQL-like query interface in which users input a query that is then converted into a job that is executed on the Hadoop cluster.

Most of the ML was done with Weka [10], an open-source software featuring a collection of ML algorithms. For Weka to work with the massive data sets, we performed random sampling to extract a manageable number of records for the software (running on only one computer) to train on. The ML techniques we report results for here are Decision Tree, Naïve Bayes, Logistic Regression, and Support Vector Machine (SVM). The features reported in Table 3 and Table 4 were used in the experiments reported on in this section.

For both data sets, there are periods of time in which there is simply not enough data to confidently predict links. To take this into account, only the data within time periods having a substantial amount of activity were used in this study.

5.2 Effect of Varying the Threshold

The Reality Mining data set featured call data records that could be leveraged to predict patterns of use and networks between callers. Figure 2 reports on our experimental results with this data set. Along the x-axis, the threshold for the number of calls is varied from $\tau_c = 1$ to $\tau_c = 21$. Applying a threshold to the original Reality Mining data created smaller derived data sets D^+ (purple curves). Along the y-axis, precision (blue curves), recall (red curves), and F-measure (green curves) are reported for Decision Tree classifiers constructed from derived Reality Mining data sets constructed by varying the threshold τ_c.

Varying Strength of Social Tie: To investigate the issue of social tie strength for Reality Mining, we ran several experiments to study the effect of varying the threshold τ_c on the quality of the prediction of the learned Decision Tree classifier.

As indicated by Figure 2(a), the prediction accuracy of the Decision Tree classifier generally improves with increasing τ_c, suggesting a stronger social tie. We can also see that the data set D_L^+ size drops significantly as the threshold τ_c is increased. For training data, the maximal data set size is 7,902 while the minimal data set size is 349.

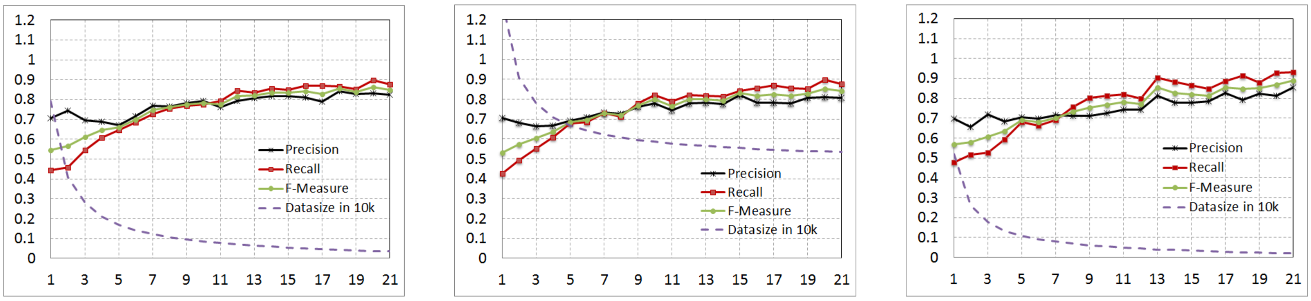
Unobserved Links: We added random 5k unobserved links from D_U into the learning data set and studied its effect on the quality of the Decision Tree prediction, while also applying a threshold τ_c. The maximal data set size is 12,902 while the minimal data set size is 5,349.

The results are shown in Figure 2(b). Overall, the conclusion is that adding the unobserved links had little or minor impacts; results are similar to those reported in Figure 2(a).

Data Cleanup: During the Reality Mining study, some users were joining while others were leaving [7]. On the other hand, certain users were active for the entire duration of the study. Even among consistently active users, the activity level varied dramatically. These observations helped us to clean up the data set, and in particular remove users who dropped out of the study. We then reran the Decision Tree algorithm on the filtered data set. Figure 2(c) summarizes the results; overall the results are improved. However, the data clean up activity limits the data set size. With a threshold of $\tau_c = 21$, used for both training and testing, a precision of 0.85, a recall of 0.93, and F-measure of 0.89 was obtained. One can also see improvements for link prediction for smaller values of τ_c (reflecting weaker social ties) compared to Figure 2(a) and Figure 2(b). For email data, optimizing thresholds gives around 30% improvement [5]. Our improvements are similar even though the data sets are quite different and our (approximately) optimal thresholds are different as well. In fact, in our case there are rela-

[8] http://hadoop.apache.org/

[9] https://cwiki.apache.org/confluence/display/Hive/Home

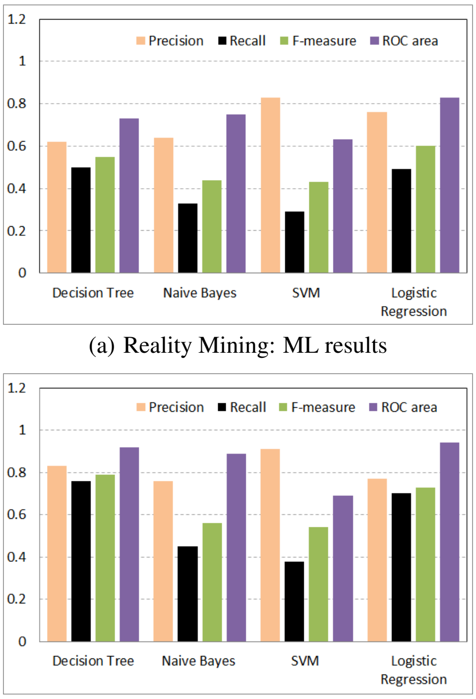


(a) Baseline

(b) With unobserved links

(c) After cleaning

Figure 2: Effect of varying threshold τ_c (along x-axis), for minimum number of calls, on quality of Decision Tree classification (along y-axis) for Reality Mining. The thresholding impacts data set size for positive examples (purple dashed graphs). The quality of link prediction is reflected in precision (black graph), recall (red graph), and F-measure (green graph) for different thresholds.



(a) Reality Mining: ML results

(b) Gowalla: ML results

Figure 3: Comparing the performance of different ML algorithms. (a) *Reality mining*: Comparative experimental results for predicting future calls κ_c; (b) *Gowalla*: Comparative experimental results for predicting links κ_f in the friendship graph.

tively clear improvements up to a threshold $\tau_c \in [9, 13]$, after which the link prediction metrics flatten out or improve only slightly.

5.3 Effect of Machine Learning Algorithm

Comparative Experiments, Reality Mining: We ran several experiments to compare the performance of different ML algorithms on the Reality Mining data set. We used a contact threshold value of $\tau_c = 10$ calls, see Figure 2(c), for these experiments. The results are shown in Figure 3(a). They indicate that we achieve good quality results, especially with Logistic Regression as measured by ROC area, even on this relatively small data set.

Comparative Experiments, Gowalla: We now turn our attention to the Gowalla data set; we used a similar approach to preprocess this data as was used for Reality Mining. Here we are using the check-in data for link prediction; in particular we use co-location information to predict links $\kappa_f(u,v)$ in the friendship graph.[10]

A social tie between two users is more likely if they repeatedly check in at the same place within a time interval. For the Gowalla data set, we extracted all the pairs of check-ins that happened at the same place for different τ_t: an hour, half an hour, and fifteen minutes. Based on these experiments, we hypothesized a social tie when a pair of users checks in within a span of $\tau_t = 15$ minutes, and the pair of users shares $\phi_{\text{co}} = 10$ or more check-ins in the Gowalla data set. In other words, τ_t and ϕ_{co} were the parameters used for thresholding.

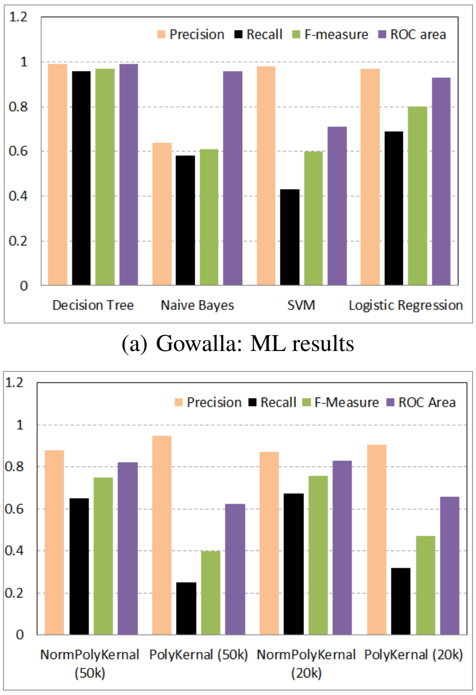
Results for different ML algorithms are shown in Figure 3(b). We are able to predict social relationship using the co-location check-in information with high accuracy. Except for SVM, all the classifiers are producing quality results in terms of ROC area. If we consider ROC area, Decision Tree performs almost as well as Logistic Regression. For the three other metrics—precision, recall, and F-measure—Decision Tree out-performs the other ML techniques.

5.4 Future Co-location Prediction

In these experiments, link prediction used co-location $\kappa_l(u,v)$ instead of friendship graph $\kappa_f(u,v)$. We used past co-location information to predict the future possibility of co-location. These experiments are very similar to Reality Mining, where there is no friendship graph and we used the CDR information to predict future call between pairs of users. The results, shown in Figure 4(a), look promising. For Decision Tree, precision is 0.99, recall is 0.96, F-measure is 0.97, and ROC area is 0.99. We attribute these strong results in part to the large data set size, in combination with our features and use of thresholds and data cleaning. In addition, our approach does simplify the very difficult link prediction task and, consequently, we obtain very good results.

SVM's results using polykernel in Figure 4(a) were surprisingly poor, hence we ran some SVM experiments with polykernel normalization. Unfortunately, on the large data set with normalization, SVM ran very slowly. Therefore, we performed smaller-scale experiments, in which SVM learned from a random sample of the original data set. The results, shown in Figure 4(b), indicate an improvement for several of the metrics when normalized polykernel is used. The first part of the figure is for 50k data set size and the second part of the figure is for 20k data set size. We see that SVM with

[10] In other words, the friendship graph acts as the gold standard.



(a) Gowalla: ML results

(b) Gowalla: SVM results

Figure 4: Performance of different ML algorithms on the Gowalla data set. (a) Predicting future collocation κ_l using four different classifiers; SVM uses polykernel. (b) Improving several SVM performance metrics, when predicting future collocation κ_l, by introducing normalized polykernel.

polykernel normalization performs approximately as well as Logistic Regression in Figure 4(a), but not as well as Decision Tree.

6. CONCLUSION AND FUTURE WORK

In this paper, we investigate link prediction in the sense of predicting a social tie in the future, given that certain users already share a tie. Specifically, we apply several supervised ML algorithms to the Gowalla and Reality Mining data sets, and our findings clearly demonstrate their capacity to predict such social interactions. In particular, the Decision Tree and Logistic Regression techniques perform consistently well, especially in conjunction with proper preprocessing and thresholding of the data sets.

As for future work, instead of simply predicting the existence of a link, it would be interesting to predict its weight. For example, in the case of mobile device networks, predict the number of calls that are going to happen during a future time interval.

7. ACKNOWLEDGMENTS

This work is supported, in part, by NSF award CCF0937044 to Ole J. Mengshoel.

8. REFERENCES

[1] M. Al Hasan, V. Chaoji, S. Salem, and M. Zaki. Link prediction using supervised learning. In *Proc. of SDM 06 workshop on Link Analysis, Counterterrorism and Security*, 2006.

[2] M. Al Hasan and M. J. Zaki. A survey of link prediction in social networks. In C. C. Aggarwal, editor, *Social Network Data Analytics*, pages 243–275. Springer, 2011.

[3] N. Benchettara, R. Kanawati, and C. Rouveirol. Supervised machine learning applied to link prediction in bipartite social networks. In *Proc. of the 2010 International Conference on Advances in Social Networks Analysis and Mining*, ASONAM '10, pages 326–330, 2010.

[4] E. Cho, S. A. Myers, and J. Leskovec. Friendship and mobility: user movement in location-based social networks. In *Proc. of KDD-11*, pages 1082–1090, 2011.

[5] M. De Choudhury, W. A. Mason, J. M. Hofman, and D. J. Watts. Inferring relevant social networks from interpersonal communication. In *Proc. of WWW-10*, pages 301–310, 2010.

[6] N. Eagle and L. M. A. de Montjoye, Y.-A.and Bettencourt. Community computing: Comparisons between rural and urban societies using mobile phone data. In *Proc. of CSE-09*, pages 144–150, 2009.

[7] N. Eagle, A. Pentland, and D. Lazer. Inferring friendship network structure by using mobile phone data. *PNAS*, 106(36), 2009.

[8] T. S. Evans and R. Lambiotte. Line graphs, link partitions, and overlapping communities. *Phys. Rev. E*, 80:016105, Jul 2009.

[9] M.C. González, C.A. Hidalgo, and A.L. Barabási. Understanding individual human mobility patterns. *Nature*, 458(7235):238–238, 2009.

[10] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The WEKA data mining software: an update. *SIGKDD Explor. Newsl.*, 11(1):10–18, 2009.

[11] J. M. Kleinberg. Navigation in a small world. *Nature*, 406(24):845, August 2000.

[12] J. Kunegis and A. Lommatzsch. Learning spectral graph transformations for link prediction. In *Proc. of the 26th Annual International Conference on Machine Learning (ICML-09)*.

[13] D. Liben-Nowell and J. M. Kleinberg. The link-prediction problem for social networks. *JASIST*, 58(7):1019–1031, 2007.

[14] S. Motahari, O. J. Mengshoel, P. Reuther, S. Appala, L. Zoia, and J. Shah. The impact of social affinity on phone calling patterns: Categorizing social ties from call data records. In *Proc. of SNA-KDD-12*, Beijing, China, August 2012.

[15] A. A. Nanavati, R. Singh, D. Chakraborty, K. Dasgupta, S. Mukherjea, G. Das, S. Gurumurthy, and A. Joshi. Analyzing the structure and evolution of massive telecom graphs. *IEEE Trans. on Knowl. and Data Eng.*, 20(5):703–718, 2008.

[16] M. Newman, A. L. Barabási, and D. J. Watts, editors. *The Structure and Dynamics of Networks*. Princeton University Press, 2006.

[17] D. Wang, D. Pedreschi, C. Song, F. Giannotti, and A.-L. Barabasi. Human mobility, social ties, and link prediction. In *Proc. of KDD-11*, pages 1100–1108, 2011.

[18] H. Zhang and R. Dantu. Predicting social ties in mobile phone networks. In *Proc. 2010 IEEE International Conference on Intelligence and Security Informatics (ISI)*, pages 25 –30, May 2010.

[19] B. Zhao, P. Sen, and L. Getoor. Entity and relationship labeling in affiliation networks. In *Proc. of the 2006 ICML Workshop on Statistical Network Analysis*, 2006.