Inferring Social Groups Using Call Logs

Santi Phithakkitnukoon and Ram Dantu

Department of Computer Science and Engineering
University of North Texas, Denton, TX 76203, USA
{santi,rdantu}@unt.edu

Abstract. Recent increase in population of mobile phone users makes it a valuable source of information for social network analysis. For a given call log, how much can we tell about the person's social group? Unnoticeably, phone user's calling personality and habit has been concealed in the call logs from which we believe that it can be extracted to infer its user's social group information. In this paper, we present an end-to-end system for inferring social networks based on "only" call logs using kernel-based naïve Bayesian learning. We also introduce normalized mutual information for feature selection process. Our model is evaluated with real-life call logs where it performs at high accuracy rate of 81.82%.

Keywords: Social groups, Call logs.

1 Introduction

Social network describes a social structure of social entities and the pattern of interrelationships among them. A social network can be either face-to-face or virtual network in which people primarily interact via communication media such as letters, telephone, email, or Usenet. Knowledge of social networks can be useful in many applications. In commerce, viral marketing can exploit the relationship between existing and potential customers to increase sales of products and services. In law enforcement, criminal investigation concerning organized crimes such drugs and money laundering or terrorism can use the knowledge of how the perpetrators are connected to one another to assist the effort in disrupting a criminal act or identifying additional suspects.

Social computing has emerged recently as an exciting research area which aims to develop better social software to facilitate interaction and communication among groups of people, to computerize aspects of human society, and to forecast the effects of changing technologies and policies on social and cultural behavior. One of the major challenges in social computing is obtaining real-world data. Quite often, analysis is based on simulations.

With rapidly increasing number of mobile phone users, mobile social networks have gained interests from several research communities. We also find it interesting to study the relationships between mobile phone users' calling behaviors and their social groups. With availability of real-life data of mobile phone users' call logs collected by Reality Mining project group [1], it allows us to carry out our analysis

R. Meersman, Z. Tari, and P. Herrero (Eds.): OTM 2008 Workshops, LNCS 5333, pp. 200–210, 2008.

and experimental results in this paper where we propose an end-to-end system for inferring social groups based solely on call logs. We believe that phone user's calling personality and habit has been unnoticeably concealed in the call logs from which it can be extracted to infer its user's social group information. To the best of our knowledge, no scientific research has been reported in classifying social networks/groups based solely on call logs.

The rest of the paper is organized as follows. In section 2, the system overview is presented. Section 3 describes our real-life dataset collected from mobile phone users. Data extraction process is then carried out in section 4 with preliminary statistical analysis. Section 5 discusses feature selection process and introduces normalized mutual information for selecting useful features. Kernel-based naïve Bayesian classifier is presented in section 6. The performance evaluation of proposed system is carried out through implementation in section 7. Finally, in section 8, we summarize our findings and conclude this paper with an outlook on future work.

2 System Overview

The system described here is intended to perform social group classification based on personal phone records. The input is phone records or call logs showing pertinent information (number dialed, duration, time of communications, etc.). The call log are then transformed into knowledge useful for the classifier by extracting calling patterns and selecting useful features. The kernel-based naïve Bayesian classifier is used to perform supervised classification based on computed probability using kernel density estimator.

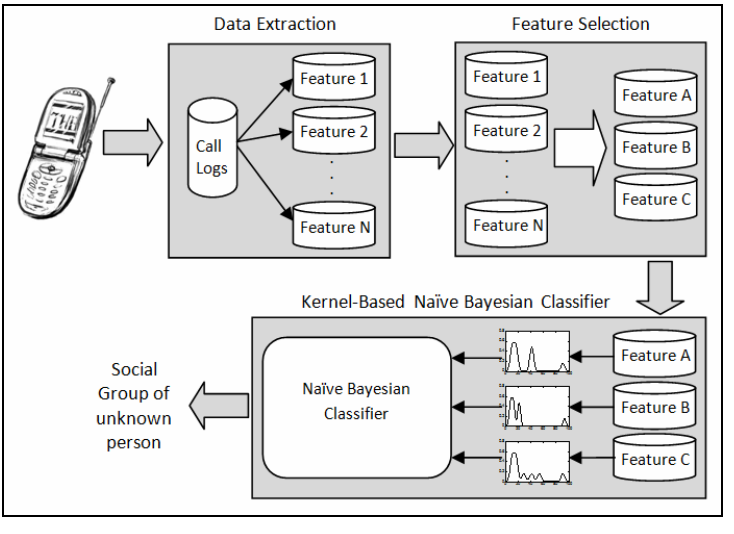

Fig. 1. System overview

3 Real-Life Dataset

Every day phone calls on the cellular network include calls from/to different sections of our social life. We receive/make calls from/to family members, friends, supervisors, neighbors, and strangers. Every person exhibits a unique traffic pattern. Unnoticeably, phone user's calling personality and habit has been concealed in the call logs from which we believe that it can be extracted to infer its user's social networks information. To study this, we use the real-life call logs of 94 individual mobile phone users over the course of nine months which were collected at Massachusetts Institute of Technology (MIT) by the Reality Mining project group [1]. Call logs were collected using Nokia 6600 smart phones loaded with software written both at MIT and the University of Helsinki to record minutely phone information including call logs, users in proximity, locations, and phone application currently being used. Of 94 phone users, 25 were incoming Sloan business students while the remaining 69 users were associated with the MIT Media Lab. According to MIT [1], this study represents the largest mobile phone study ever in academia and the data collected can be used in a variety of fields ranging from psychology to machine learning. There are some research works conducted by the Reality Mining project group using this dataset involves relationship inference, user behavior prediction, and organizational group dynamics.

As previously mentioned, our interest and the focus of this paper is to extract the phone user's behavior concealed in the call logs and attempt to accurately classify user into belonging social networks. With MIT dataset, classification can be performed to differentiate phone users from the Media Lab and from Sloan. As described earlier, MIT dataset consists of more than just call logs but location information and others. In order to be more generalized, only call logs are considered for our study as currently call logs are only accessible feature from service providers (e.g. billing, online account).

Due to missing information on the dataset which leaves us 84 users instead of 94 users, we then have 22 Sloan users and 62 Media Lab users. Of 62 Media Lab users, 20 users are clearly marked as students. We believe that even though all 62 users are with Media Lab, sub-social groups can be formed such as students, faculty, and staff, which exhibit slightly different calling behavior. Therefore we choose to perform classification between clearly marked Media Lab students and Sloan students.

4 Data Extraction

The main goal is to find some features from the call logs (raw data) that can solidly differentiate Media Lab students and Sloan students. For the data extraction process, we try to retrieve as much as possible useful features from the call logs. There might not be one dominate feature that captures entire calling behavior but combination of those characterize the core behavior structure. There are 11 features extracted and listed in Table 1 along with some statistical analysis (i.e. averages (Avg.) and standard deviations (Std.)) where feature descriptions are listed in Table 2.

From the first glance of these features and their statistics, it is clear that there are differences between two social networks however the differences are not adequately large enough to differentiate them based on each individual feature.

Table 1. Extracted features

Features	Media Lab		Sloan	
	Avg.	Std.	Avg.	Std.
All_calls	9.670	6.902	14.168	7.264
Inc_calls	2.920	2.542	3.756	2.197
Out_calls	6.750	4.501	10.413	5.549
Missed_calls	8.708	6.167	12.810	6.718
All_talk	246.716	304.899	196.966	260.884
Inc_talk	140.906	109.479	172.518	111.427
Out_talk	272.599	367.231	207.328	320.731
All_call_time	12.934	1.671	14.571	2.120
Inc_call_time	13.164	1.775	14.591	2.226
Out_call_time	12.881	1.752	14.583	2.121
Ent_call_time	6.137	0.683	4.059	0.553

Table 2. Extracted feature descriptions

Features	Feature description
All_calls	The total number of all calls per day including incoming, outgoing, and missed calls.
Inc_calls	The number of incoming calls per day.
Out_calls	The number of outgoing calls per day.
Missed_calls	The number of missed calls per day.
Inc_talk	The total amount of time spent talking on the phone (call duration) per day (in seconds) including both incoming and outgoing calls.
Out_talk	The amount of time spent talking (in seconds) per day on the incoming calls.
All_call_time	The amount of time spent talking (in seconds) per day on the outgoing calls.
Inc_call_time	The time that calls either received or made, ranging between 0 and 24 (0AM – 12PM).
Out_call_time	The arrival time of incoming calls, 0-24 (0AM – 12PM).
Ent_call_time	The departure time of outgoing calls, 0-24 (0AM – 12PM).

The last feature in Table 1 and 2 is information entropy [2] which is a measure of the uncertainty of a random variable. In our case, this random variable is calling time. The entropy of a variable X is defined by (1) where $x_i \in X$ and $P(x_i) = Pr(X=x_i)$.

$$H(X) = -\sum_i P(x_i)\log_2(P(x_i)) \cdot \tag{1}$$

Assessment based on these extracted features is that Sloan students tend to make more phone calls than Media Lab students, whereas Media Lab students like to talk (or spend time) on the phone longer on outgoing calls but talk less on incoming calls than Sloan students. Sloan students spend time on the phone later in the day (about 2:30PM) than Media Lab students (about 1PM). Lastly, the randomness in calling time of Media Lab students is higher than Sloan students.

5 Feature Selection

So far, we have extracted features from raw data (call logs) and we need to select the useful features for classification. This section discusses how to evaluate the usefulness of features for classification. In general, for classification task as we try to assign an unknown sample to different classes which have different characteristics. Our goal is to find a character (e.g. a set of features) of the unknown sample that mostly identifies its belonging class among other classes. This set of features need to have high degree of difference (or low degree of similarity) to other classes to be considered as a "good" set of features. If we adopt the correlation between two random variables as a goodness measure, the above definition becomes that a feature is good if it is highly correlated with the belonging class but not highly correlated with other classes.

There are two main approaches to measure the correlation between two random variables. One is based on classical linear correlation and the other is based on information theory. The first approach is the well known *linear correlation coefficient* (*r*). For any pair of random variables (*X*,*Y*), *r* can be computed by (2).

$$r = \frac{\sum_i (x_i - \bar{x}_i)(y_i - \bar{y}_i)}{\sqrt{\sum_i (x_i - \bar{x}_i)^2}\sqrt{\sum_i (y_i - \bar{y}_i)^2}}, \tag{2}$$

where $\bar{x}_i$ is the mean of *X*, and $\bar{y}_i$ is the mean of *Y*. The value of *r* is between -1 and 1. A correlation coefficient of 1, -1, and zero implies perfect linear relationship, inversely proportional relationship, and no linear relationship between the two variables respectively. It is symmetrical measure for two variables. There also exists other measures in this category which are basically variations of *r*, such as *least square regression error* and *maximal information compression index* [3]. There are several benefits of choosing linear correlation coefficient as a goodness measure for feature selection such as it helps remove features with correlation close to one from selection and retain other features with low correlation. However, in the reality it is not safe to always assume "linear" relationship between features. Linear correlation measures may not be able to capture the correlations that are not linear in nature.

Another approach to measure the correlation which is based on information theory can overcome this shortcoming. We adopt the concept of information entropy which is given in (1) which measures the degree of uncertainty between two random variables. Information theory [4] defines conditional entropy of a random variable given another with a joint distribution $P(x_i,y_j)$ as follows.

$$H(X \mid Y) = -\sum_i \sum_j P(x_i, y_j) \log_2 (P(x_i \mid y_j)). \tag{3}$$

Another important definition is *mutual information* which is a measure of the amount of information that one random variable contains about another random variable which is given by (4).

$$I(X;Y) = -\sum_i \sum_j P(x_i, y_j) \log_2 \left(\frac{P(x_i, y_j)}{P(x_i)P(y_j)}\right). \tag{4}$$

Given (1) and (3), it is straightforward to derive (5).

$$I(X;Y) = H(X) - H(X \mid Y). \tag{5}$$

Mutual information is also referred to as *information gain* [5] which can be interpreted as a measure of the amount by which the entropy of X decreases reflects additional information about X provided by Y.

Theorem. The mutual information is symmetrical for two random variables X and Y which can be proved as follows.

Proof. To show that $I(X;Y) = I(X;Y)$.

$$\begin{aligned} I(X;Y) &= H(Y) - H(Y \mid X) \\ &= H(Y) - \big(H(X,Y) - H(X)\big) \\ &= H(X) + H(Y) - H(X,Y) \\ &= H(X) - H(X \mid Y). \end{aligned}$$

For fairness in comparisons, normalization is needed. Therefore the *normalized mutual information* can be derived as

$$I_{Nor}(X;Y) = \frac{H(X) - H(X \mid Y)}{H(X)}, \tag{6}$$

where denominator $H(X)$ is a scale factor to normalize it to [0, 1].

6 Kernel-Based Naïve Bayesian Classifier

The approach to classification taken here is based on Bayes rule [6] of conditional probability which is given by (7).

$$P(Y \mid X) = P(Y)\frac{P(X \mid Y)}{P(X)}, \tag{7}$$

where $P(Y|X)$ is the *a posteriori* probability which is the probability of the state of nature being Y given that feature value X has been measured. The *likelihood* of Y with respect to X is P($X|Y$) which indicates that other things being equal, the category Y for which $P(Y|X)$ is large is more "likely" to be the true category. $P(Y)$ is called *a priori* probability. The *evidence* factor, $P(X)$, can be viewed as a scale factor to guarantee that the posterior probabilities sum to one.

Suppose now that we have N input features, $X = \{x_1, x_2, .., x_N\}$, which can be considered independent both unconditionally and conditionally given y. This means that the probability of the joint outcome x can be written as a product,

$$P(X) = P(x_1) \cdot P(x_2) \cdots P(x_N) \tag{8}$$

and so can the probability of X within each class y_j,

$$P(X \mid y_j) = P(x_1 \mid y_j) \cdot P(x_2 \mid y_j) \cdots P(x_N \mid y_j). \tag{9}$$

With the help of these it is possible to derive the basis for the *naïve Bayesian classifier* [7] as follows,

$$P(y_j \mid X) = P(y_j)\frac{P(X \mid y_j)}{P(X)} = P(y_j)\prod_{i=1}^{N}\frac{P(x_i \mid y_j)}{P(x_i)}. \tag{10}$$

The designation *naïve* is due to simplistic assumption that different input attributes are independent.

From (10), the classification is then based on the likelihood function given by (11).

$$L(y_j \mid X) = \prod_{i=1}^{N} P(x_i \mid y_j). \tag{11}$$

Most applications that apply naïve Bayesian classifier derive likelihood function from the actual data or assumed parametric density function (e.g. Gaussian, Poisson). Another approach to derive likelihood function is by using non-parametric density estimation. The most popular method is the kernel estimation which is also known as the Parzen window estimator [8] as follows,

$$f(z) = \frac{1}{Mh}\sum_{k=1}^{M} K\left(\frac{z - z_k}{h}\right), \tag{12}$$

where $K(u)$ is kernel function, M is the number of training points, and h is the bandwidth or smoothing parameter. The most widely used kernel is Gaussian of zero mean and unit variance ($\mathcal{N}(0,1)$) which is defined by (13).

$$K(u) = \frac{1}{\sqrt{2\pi}} e^{-u^2/2}. \tag{13}$$

The choice of the bandwidth h is crucial. Several optimal bandwidth selection techniques have been proposed ([9]). In this study, we use AMISE optional bandwidth selection using the *Sheather Jones Solve-the-equation plug-in* method which was proposed in [10].

Kernel density estimator provides smoothness to likelihood function with continuous attributes rather than relying on discrete ones. Now the likelihood function in (11) becomes

$$L(y_j \mid X) = \frac{1}{Mh}\prod_{i=1}^{N}\left(\sum_{k=1}^{M} K\left(\frac{y_j - z_k^i}{h}\right)\right), \tag{14}$$

where z_k^i is training point k of feature i.

7 Implementation and Results

To evaluate our proposed system, we continue our implementation from data extraction process in section 4. Recall that we have 11 extracted features from the call logs. Now we need to select useful features based on normalized mutual information as discussed in section 5. Based on (6), normalized mutual information is computed for each feature and plotted in Fig. 2 for comparison. If normalized mutual information of 0.5 is chosen as a threshold, then we have six featured selected with the highest degree of discriminancy.

The six selected features with their corresponding normalized mutual information are listed in ascending order in Table 3. Recall that less normalized mutual information implies higher order of discriminancy or most useful feature for classification.

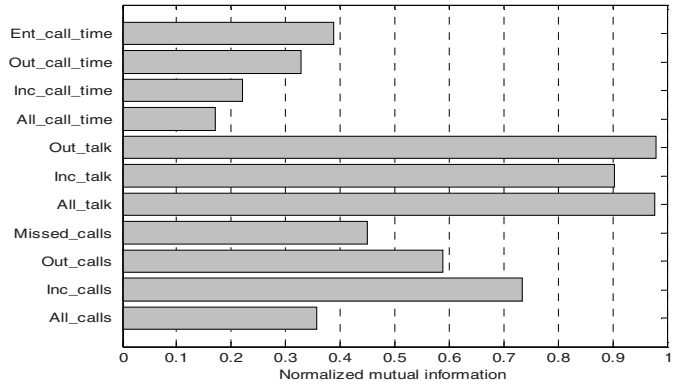

Fig. 2. Result of normalized mutual information

Table 3. Selected features based on normalized mutual information

Features	Normalized Mutual Information
All_call_time	0.169
Inc_call_time	0.220
Out_call_time	0.328
All_calls	0.357
Ent_call_time	0.388
Missed_calls	0.450

The useful features have been selected, before reaching classifier feature normalization is needed. The reason for normalization is to reduce the noisiness of features since non-normalized features have different ranges and are measured in different units. Thus, selected features are normalized to [0, 1].

Features are now ready to be fed to classifier which operates in two modes; training and testing. We use 50% of our feature set as training data and the other 50% as testing data. We implement our proposed method of using kernel-based naïve Bayesian classifier with selected six features based on normalized mutual information. The performance of our proposed method is measured by the accuracy rate which is a ratio of correct classified users to the total testing users. For performance comparison purposes, we also implement naïve Bayesian classifier using all 11 extracted features, naïve Bayesian classifier using six selected features, and kernel-based naïve Bayesian classifier using all 11 extracted features to compare with our method. The result is shown in Table 4, among four approaches, our approach has the best performance with accuracy rate of 81.82%. Naïve Bayesian classifier using all 11 extracted

Table 4. Accuracy comparison of classifier with different methods

Methods	Accuracy Rate (%)
Naïve Bayes with all features	59.09
Naïve Bayes with six selected features	68.18
Kernel-based naïve Bayes with all features	77.27
Kernel-based naïve Bayes with six selected features	81.82
Naïve Bayes with all features	59.09
Naïve Bayes with six selected features	68.18

features, naïve Bayesian classifier using six selected features, and kernel-based naïve Bayesian classifier using all 11 extracted features perform at accuracy rates of 59.09%, 68.18%, and 77.27% respectively.

In addition, to evaluate the effectiveness of the six selected features based on normalized mutual information, we sort all 11 features based on normalized mutual information in ascending order and monitor the changes in accuracy rate as more ascending sorted features taken into account. We monitor both kernel-based naïve Bayesian and classical naïve Bayesian approach which are shown in Fig. 3. Accuracy rate of both methods continue to increase up to when six features are taken into account, then accuracy rate decreases. The accuracy rate continues to decrease after more than six features taken for naïve Bayesian classifier whereas the accuracy decreases from six to seven features and stays constant until all 11 features are taken into account for kernel-based naïve Bayesian classifier.

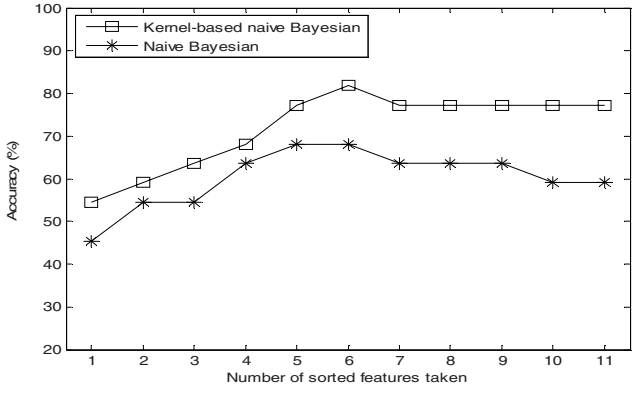

Fig. 3. Change of accuracy according to number of features selected

Figure 3 tells us that the selected six features listed in Table 3 are indeed useful features for classification. Including more features for classifier does not mean better performance. In fact, it may degrade the performance of classifier with its noisiness and low degree of discriminancy.

8 Conclusion

According to the CTIA [11], there are currently 243 million mobile phone subscribers in the US. With a current population of around 300 million and assuming that the CTIA figure implies unique subscribers, about two in every three Americans own a mobile phone. With this widespread use of mobile phones, it becomes valuable source of information for social networks analysis. In this paper, we analyze social networks based on mobile phone's call logs, and propose a model for inferring groups. We describe data pre-processing process which consists of data extraction and feature selection in which we introduce a technique for selecting features using normalized mutual information that measures degree of disciminancy. With its symmetrical and linearity-invariance property, we show that it makes normalized mutual information suitable for our feature selection process. We adopt the classical naïve Bayesian learning and introduce kernel density estimator to estimate the likelihood function which improves accuracy of the classifier with its smoothness. Our model is evaluated with real-life call logs from Reality Mining project group. The performance is measured by the accuracy rate. The results show that our model performs at accuracy rate of 81.82% which is highest among other models (Naïve Bayesian classifier using all extracted features, naïve Bayesian classifier using six selected features, and kernel-based naïve Bayesian classifier using all extracted features). We believe that our model can be also useful for other pattern recognition and classification tasks. As our future directions, we will continue to investigate on the features that can be extracted from call logs which can be useful for classification. We will also explore other statistical learning techniques to improve accuracy of our model.

Acknowledgements. This work is supported by the National Science Foundation under grants CNS-0627754, CNS-0619871and CNS-0551694.

References

1. Eagle, N., Pentland, A.: Reality Mining: Sensing Complex Social Systems. Journal of Personal and Ubiquitous Computing 10(4), 225–268 (2005)
2. Shannon, C.E.: A Mathematical Theory of Communications. Bell System Technical Journal 27, 379–423, 623–656 (July and October 1948)
3. Mitra, P., Murphy, C.A., Pal, S.K.: Unsupervised feature selection using feature similarity. IEEE Transactions on Pattern Analysis and Machine Intelligence 24, 301–312 (2002)
4. Cover, T.M., Thomas, J.A.: Elements of Information Theory. Wiley, Chichester (1991)
5. Quinlan, J.: C4.5: Programs for machine learning. Morgan Kaufman Publishers, Inc., San Francisco (1993)
6. Duda, R.O., Hart, P.E.: Pattern Classification and Scene Analysis. John Wiley, New York (1973)
7. Good, I.J.: Probability and the Weighing of Evidence. Charles Griffin, London (1950)
8. Parzen, E.: On estimation of a probability density function and mode. Annual Mathematical Statistics 33(3), 1065–1076 (1962)

9. Wand, M.P., Jones, M.C.: Kernel Smoothing. Chapman & Hall, Boca Raton (1994)
10. Sheather, S.J., Jones, M.C.: A reliable data-based bandwidth selection method for kernel density estimation. Journal of the Royal Statical Society, Series B(53), 683–690 (1991)
11. Wireless Quick Facts. CTIA, Ed. (2007), http://www.ctia.org/media/index.cfm/AID/10323