ABSTRACT
A tremendous amount of information is being shared everyday on social media sites such as Facebook, Twitter or Google+. But only a small portion of users provide their location information, which can be helpful in targeted advertisement and many other services. In this demo we present our large scale user location estimation system, SPOT, which showcase different location estimating models on real world data sets. The demo shows three different location estimation algorithms: a friend-based, a social closeness-based, and an energy and local social coefficient based. The first algorithm is a baseline and the other two new algorithms utilize social closeness information which was traditionally treated as a binary friendship. The two algorithms are based on the premise that friends are different and close friends can help to estimate location better. The demo will also show that all three algorithms benefit from a confidence-based iteration method. The demo is web-based. A user can specify different settings, explore the estimation results on a map, and observe the statistical information, e.g. accuracy and average friends used in the estimation, dynamically. The demo provides two datasets: Twitter (148,860 located users) and Gowalla (99,563 located users). Furthermore, a user can filter users with certain features, e.g. with more than 100 friends, to see how the estimating models work on a particular case. The estimated and real locations of those users as well as their friends will be displayed on the map.

Keywords
Location estimation, social closeness, local social coefficient

1. INTRODUCTION
A tremendous amount of information is being shared everyday on social media sites such as Facebook, Twitter, or Google+. Often times these social media texts include information that are valuable to others, such as activities (e.g., art fairs, jazz festivals, and gatherings), natural disaster occurrences (e.g., tornadoes, earthquakes), or incidents (e.g., traffic jams). The goal of this demo is to show different algorithms to estimate user locations. The results will help event detection from social media which in turn can assist the assimilation of social media information of interest for applications domains such as smart transportation, disaster relief and recovery, and national security.

Recent research on location prediction or estimation in social network follows two directions based on the data used: content-based and geotagged-friend-based. Content-based prediction models assume that most users do not provide their location information in social networks. Cheng et al. [3] proposed and evaluated a city-level estimation model of Twitter users’ location purely by taking the location related words in tweet content as features and applying classification method. Chandra et al. [2] improved the content based method by using user interactions and exploiting the relationship between different tweet message types. They also provided the estimation of the top-K probable cities for a user. Another similar research work is proposed in [5]. When a user declared a place, it will be checked in gazetteer to see if it corresponds to a city name. And then these location information will be applied to infer the user location by Twitter network.

Our estimating module is closely related to the work by Backstrom et al. [1]. By modelling the relation between distance and probability of being friends, the authors proposed a method to calculate the probability of a user located at a specific place. Place with the maximal probability will be estimated as the location of the user. Both [8] and [4] aimed to build a user mobility model by using the location of their friends. Cho et al. [4] use several factors in their probability model, including the check-in records, social network, friends’ location and time. Sadilek et al. [8] applied machine learning method with the similar information. Rui et al. proposed the UDI (unified discriminative influence) framework to combine content and friends’ location analysis in a unique model to profile users’ home location in [7]. However, all of those models take the friend relation as a binary feature, friends or not. This premise made those models not be able to take advantage of all the information from the social network. In our model, we take the social relation as a continuous feature by introducing the concept of social closeness. By studying the relationship between social closeness and geo-distance, our models: 1) significantly improve the estimation accuracy, especially when people only have few friends, 2) overcome the location sparsely problem,
Figure 1: Architecture of the SPOT System: The main components includes the data collection and preprocessing module, the estimation algorithms, and the user interface.

2. ARCHITECTURE

Figure 1 is the architecture of the SPOT system. It shows the basic system flow: collecting data from Twitter and Gowalla, setting environment, running the estimation algorithms, and visualizing results on the web page. The core components include the friend-based model, the social closeness based model, the energy and local social coefficient model and the confidence-based iteration method. We will describe those components in the following sections.

Here we define the social network as a graph $G = (U, E)$, where $U$ represents the user set and edges in $E$ exist between two users if they have friend relation. Let $\Gamma_i$ be the set of friends (neighbors) of user $u_i$. Then the number of common friends of user $u_i$ and $u_j$ is denoted by $\sigma_{ij} = |\Gamma_i \cap \Gamma_j|$. We use $l_{ij}$ to denote the location of user $u_i$ and $|l_i - l_j|$ is the distance between user $u_i$ and $u_j$.

2.1 Friend Based Model

The friend-based model is the baseline model and was proposed in [1]. The authors estimated the user location by their friends’ locations based on the relationship between distance and the probability of being friends. The probability of a user $u_i$ locating at $l_i$ is estimated as: $\prod_{(u_i, u_j) \in E} P(|l_i - l_j|) \prod_{(u_i, u_j) \in E} (1 - P(|l_i - l_j|))$. Here $P(|l_i - l_j|)$ represents the probability of user $u_i$ and $u_j$ located with the distance of $|l_i - l_j|$ and $E$ is the set of friend relation. And then they optimize the formula as: $\prod_{(u_i, u_j) \in E} \frac{1}{1 - P(|l_i - l_j|)}$.

2.2 Social Tightness Based Model

The social tightness based model is based on the assumption that different friends have varying importance to a user. The social closeness between two users can be measured by cosine similarity: $s_{ij} = |\Gamma_i \cap \Gamma_j|/\sqrt{|\Gamma_i||\Gamma_j|}$. Our investigation shows that a pair of friends has 83% of chance to live within 10 kilometers if the number of common friends is more than 50% of the total friends of both users. This ratio decreases to 2.4% if the common friend ratio is decreased to 10%. This phenomenon supports our hypothesis that social distance can help us to identify “important” friends in the location estimation. Thus we introduce social closeness into location estimating. We obtain the probability $P(|l_i - l_j|, s_{ij})$ of user $u_i$ and $u_j$ located at the distance of $|l_i - l_j|$ with social closeness $s_{ij}$ from training data. Then we can estimate the probability of user $u_i$ located at $l_i$ and use the location with the top probability.

2.3 Energy and Local Social Coefficient Model

In this model, we define a natural distance of spring $d_r$ between friend pairs $(u_i, u_j)$ under a social closeness scope $r$. Then we build an energy function to draw a “heat map”
representing the “pulling” strength for each user by their friends. When we put user $u_i$ at $l_i$, the energy generated by his/her friend $u_j$ is $g(u_i, u_j) = -e^{-|l_i - l_j|/d}$, and $s_{ij} \in r$. So to a user $u_i$, the total energy of $u_i$ locating at $l_i$ is:

$$G(u_i, l_i) = -\sum_{j=1}^{\Gamma_i} s_{ij} g(u_i, u_j), u_j \in \Gamma_i.$$  

Ironically, we are facing both location sparsely problem and the problem of a user having too many friends. A user with too many friends dispersed in wide geographical areas contributes to estimation error. The social coefficient can help revising the result when users have lots of friends. A user with too many friends dispersed in wide geographical areas contributes to estimation error. The social coefficient can help revising the result when users have lots of friends and is defined as: $C(u_i) = \frac{3 \times G_\Delta}{3 \times G_\Delta + G_\Lambda}$. Here $G_\Delta$ means the number of closed triplets which contains $u_i$, in graph $G$ and $G_\Lambda$ is the number of open triplets connected by $u_i$ and obviously, $G_\Lambda = |\Gamma_i|(|\Gamma_i| - 1)/2$. Based on this, when we put $u_i$ at location $l_i$, we find all located friends $\Gamma_i$ of $u_i$. A new graph is formed by connecting two friends in $\Gamma_i$ if they locate within 10km. The local social coefficient of $l_i$ is calculated for each connected component of the new graph.

In our model, we combine the energy function and local social coefficient to overcome the disadvantages of them. We calculate the energy and local social coefficient on each friend’s location and we can get two ranking results of those locations. Assuming $r_G$ and $r_C$ are the ranking results of those friends’ location from the energy method and social coefficient index, by the logistic response function:

$$\pi(r_A) = \frac{exp(\alpha + \beta_1 r_G + \beta_2 r_C)}{1 + exp(\alpha + \beta_1 r_G + \beta_2 r_C)} \quad (1)$$

we can combine those two estimating models. The parameters $\alpha$, $\beta_1$ and $\beta_2$ are trained by methods based on maximum likelihood.

### 2.4 Iteration with Confidence-Based Improvement

The sparsity of location information is the biggest challenge in the location estimation. Especially, when a user has only one or two friends, we may not be able to get sufficient location information from his/her friend. Unfortunately, many users have only a few friends. To overcome this problem, we propose an iteration method which uses the estimated locations in next round of location estimation. In the iteration process, the estimated location will be taken as a friend’s real location. In this way, the users which have no located friend at the beginning may be able to be estimated since their friends are estimated after several iterations.
But there is a problem that the incorrect estimation results may lead to the decreasing of accuracy. So we proposed a confidence-based iteration method to filter out the estimation results which may be incorrect. Our investigation shows that the most helpful feature to assign confidence is the percentage of friends who are located around the estimation location. Here we apply an entropy-like method to measure whether users’ friends are concentrated in one area and in each iteration step, we only use the top $2/3$ estimated users which we believe have a higher probability as ones with correct estimation results.

3. DEMONSTRATION SCENARIO

Our data sets come from two different social network applications, Gowalla and Twitter, and both of them are publicly available. Gowalla is a location-based social network and users are able to check in at “spots” in their local vicinity. The Gowalla dataset [4] contains 196,591 users. We use 99,563 of those users who have check-in records in our demo set. We also collected user profiles from Twitter, an on-line social networking and microblogging service which enables users to follow each other and read “tweets”. There are totally 660,000 users and their social relation in the Twitter dataset and we collected 148,860 users’ locations through Twitter API. We define the friend relationship the same way as that in [6]. Users A and B have friend relation if they follow each other. Each user has 29.4 friends on average in the Twitter dataset.

The screen shot of SPOT system is shown in Figure 2. To begin the demonstration, users need to choose several settings include: (1) data sets selection: users can run the models on either the Twitter data set or the Gowalla data set; (2) model selection: include the friend-based model, the social tightness based model, and the energy and local social coefficient model; (3) location mask: users can hide certain percent of user location information (from 20% to 80%). In the estimation process, those users’ location will not be used. In this way, we can test the tolerance of the models on the sparsity of location information; (4) estimation sample selection: users can select estimation samples by their friends number and the estimation results (correct or incorrect) and only the samples which meet the requirements will be visualized on the map.

By click the “Set” button, the demonstration will run the first time estimation. The output includes statistics of the estimation results and selected samples. The statistics results include estimation accuracy, average distance between estimated location and real location, and average number of friends used in the estimation. Users can click the iteration button to enter the iteration process. The system will use the estimation results in next time estimation process. The figures of those statistics will be generated and dynamically updated after each iteration. Selected estimation samples will be visualized on the map showing both their real and estimated locations. By clicking a sample, the sample’s friends’ estimated/real location will also be shown on the map. A user has the choice of visualizing all samples, the corrected estimated samples, or the incorrectly estimated samples as well. In SPOT system, we define an estimation result as correct when the distance between estimated location and real location is less than 100 miles. All the models can be combined with the iteration method.

4. CONCLUSION

SPOT is a system that can perform several location estimating models on different data sets. It demonstrates both the estimation accuracy and selected samples to help users compare those models under different settings and observe the estimation process clearly. We also show statistics such as the average error distance and average number of friends.

Acknowledgments.

This work is supported in part by USDOD.

5. REFERENCES


\[1\] http://snap.stanford.edu/data/loc-gowalla.html

\[2\] http://dmml.asu.edu/users/xufei/datasets.html