LBSNSim: Analyzing and Modeling Location-based Social Networks

Wei Wei*, Xiaojun Zhu†, Qun Li*
*The College of William and Mary, †State Key Laboratory for Novel Software Technology, Nanjing University
*{wwei, liqun}@cs.wm.edu, †gxjzhu@gmail.com

***Abstract*—The soaring adoption of location-based social networks (LBSNs) makes it possible to analyze human socio-spatial behaviors based on large-scale realistic data, which is important to both the research community and the design of new location-based social applications. However, performing direct measurements on LBSNs is impractical, because of the security mechanisms of existing LBSNs, and high time and resource costs. The problem is exacerbated by the scarcity of available LBSN datasets, which is mainly due to the privacy concerns and the hardness of distributing large-volume data. As a result, only a very few number of LBSN datasets are publicly released. In this paper, we extract and study the universal statistical features of three LBSN datasets, and propose LBSNSim, a trace-driven model for generating synthetic LBSN datasets capturing the properties of the original datasets. Our evaluation shows that LBSNSim provides an accurate representation of target LBSNs.**

I. INTRODUCTION

In the past few years, LBSNs have gained soaring popularity and attracted millions of users [1]. Compared with traditional online social networks, LBSNs take a step further in that they provide the location-based features. Users of LBSNs can *check-in* at different venues (e.g., airports, restaurants) and notify their friends, sharing with friends information about the places they visited. These check-ins, combined with the online friendship connections revealed through the LBSNs, provide an unprecedented opportunity to study human socio-spatial behaviors based on large-scale voluntarily contributed data. This in turn facilitates a variety of services, such as urban planning, friendship recommendation, place of interest recommendation, traffic forecasting, marketing campaigns, and epidemiological modeling.

However, it is difficult to perform direct measurements of existing LBSNs, which usually take approaches to defend against automated crawlers. For example, Foursquare, the most popular LBSN, requires user authorization to collect personal information, and it has limited the access rate. As a result, a direct measurement typically incurs high time and resource costs [4], [5]. To circumvent this difficulty, researchers have resorted to the publicly available datasets. Nevertheless, the number of LBSN datasets available to the community is very limited. This is mainly due to the concerns of compromising user privacy and the high costs of distributing large datasets. User locations may reveal highly sensitive and private information, such as interests, habits, and health conditions, especially when they are in the hands of adversaries. The threat is more serious with regard to LBSNs, because users' physical locations are now being correlated with their profile information. Even if the datasets are anonymized before being published, user identities can still be recovered from the anonymized location traces and social graphs [10], [13]. Therefore, these privacy concerns strongly discourage sharing LBSN datasets. Given the soaring adoption of LBSNs, the lack of available datasets has significantly impeded the research in this area.

An attractive alternative to shared original datasets is the synthetic datasets generated by measurement-calibrated models. There are three advantages of using synthetic datasets as replacements for real datasets. First, the synthetic datasets are randomly generated, and thus they do not compromise any user privacy. Second, compared with sharing the large datasets, the cost of sharing the models is negligible. Third, LBSN datasets with different properties can be generated on demand, which can help researchers improve the statistical confidence in their experimental results. Previous work investigated the graph models that produce synthetic social graphs of online social networks [8], [9], [16], [17]. Given all these advantages of the model-generated LBSN datasets, however, no LBSN model has been proposed in the literature.

In this paper, we propose LBSNSim, a trace-driven model for generating synthetic LBSN datasets that capture the characteristics of the real datasets. We first analyze the data from three LBSNs: Foursquare, Gowalla, and Brightkite (Section III). Our findings suggest that the LBSNs share many universal social and spatial properties. For example, the user check-in numbers follow an exponentially truncated power law distribution. The displacements between consecutive check-ins made by each user follow a two-segment distribution, whose transition point has a clear meaning. Similarly, the temporal intervals between consecutive check-ins also follow a two-segment distribution. Additionally, the friend distances follow a truncated Weibull distribution. Previous work only show that some measurements of LBSNs, such as the check-in numbers and displacements, exhibit a heavy-tail pattern [4], [15]. To the best of our knowledge, this is the first time that specific distributions have been found and explained for a wide range of statistical features of LBSNs.

Based on our findings we develop LBSNSim (Section IV), which takes as input a set of known venues, and outputs the check-in history of all the synthetic users and their friendship graph. Our model consists of three components: generating the initial location of each user, building the friendship graph by considering both social and spatial factors, and generating all the check-ins of each user.

We evaluate the fidelity of LBSNSim by comparing the properties of the real LBSN datasets with their synthetic model-generated counterparts (Section V). The results demonstrate that LBSNSim provides an accurate representation of the target LBSNs: it generates synthetic datasets that accurately capture the statistical features of the original datasets. Besides, our application-level test shows that the application results obtained by using the model-generated datasets closely match those obtained by using the original datasets, which validates the feasibility of substituting real datasets with synthetic datasets. As the first generative model of LBSNs, LBSNSim has wide applications for the research community and in guiding the design of the systems and applications centered on LBSNs.

II. RELATED WORK

Previous studies have attempted to investigate the check-in properties of LBSNs. Cheng et al. found that LBSN users follow the "Levy Flight" mobility pattern, which is characterized by a mixture of short, random movements with occasional long jumps [4]. Scellato et al. presented a study of the socio-spatial properties of LBSNs [18]. They found that in LBSNs long range social ties have a higher probability of occurrence than in other social systems. User behavior with regard to LBSNs has been analyzed by Lindqvist et al. [11]. The authors conducted interviews and surveys to investigate how and why people use LBSNs, as well as their privacy concerns related to the location-sharing functions.

Researchers have also leveraged the socio-spatial information of LBSNs for location prediction and friendship prediction. Cho et al. proposed a location prediction model built on the idea that human check-ins are based on the movement between two latent states "work" and "home" [5]. Noulas et al. proposed a venue recommendation scheme relying on performing personalized random walks on a user-place network, where a user is linked to her friends and the venues she has visited before [14]. Scellato et al. built a supervised learning framework that exploits the features extracted from LBSNs to predict new friendship links between friends-of-friends and place-friends, which are the users visiting the same place [19].

Sala et al. explored the feasibility of replacing real social graphs of online social networks with synthetic graphs generated from calibrated graph models [16]. The authors compared six existing graph models. They found that two models consistently generate synthetic graphs with common graph metric values similar to those of the original graphs, and one produces high fidelity results in application-level tests. In a followup work the authors investigated how to share social network graphs without compromising user privacy [17]. Previous research has also studied how to generate synthetic social graphs with different properties [8], [9].

III. DATA ANALYSIS

In this section, we investigate the statistical characteristics of the original datasets from three LBSNs: Gowalla [5], Brightkite [5], and Foursquare [4]. We consider the check-ins whose locations have latitude between 24°N and 50°N, and longitude between 64°W and 126°W. This includes the mainland of the USA, where the three LBSNs have the majority of check-ins.

TABLE I
STATISTICS OF THE DATASETS

Dataset	Users	Edges	Check-ins	Venues	Timestamps
Gowalla	196,591	950,327	3,674,591	675,483	02/2009-10/2010
Brightkite	58,228	214,078	2,920,919	476,744	04/2008-10/2010
Foursquare	93,115	NA	7,956,679	428,343	09/2010-01/2011

The statistics of the three datasets analyzed in this section are shown in Table I. The Foursquare dataset does not contain the friendship graph, since Foursquare does not allow unauthorized access to users' friend lists. Each check-in in the datasets is stored as a tuple $\langle userID, time, latitude, longitude, venueID \rangle$, and each friendship edge is stored as a tuple $\langle userID_A, userID_B \rangle$. Check-ins at the same venue have the same GPS coordinates, provided by the corresponding LBSN. Besides these three datasets, we have also studied two more datasets from Gowalla [3] and Foursquare [7]. The results are very similar and are thus omitted to save space.

We extract and analyze the following data: the number of check-ins of each user, the spatial displacement of consecutive check-ins, the temporal interval of consecutive check-ins, distance between friends, the number of friends of each user, and the number of check-ins at each venue. Our findings suggest that the datasets share many universal features, which guides the design and evaluation of LBSNSim.

A. *Number of check-ins*

We begin with an investigation of the number of check-ins made by each user. Figure 1 shows the log-log CCDF (complementary cumulative distribution function) of the number of check-ins made by each user in the three datasets. All the plots exhibit a sizable downward curvature and cannot be fitted with a straight line, indicating a significant deviation from a power law distribution. Instead, by analyzing the data, we find that they can be well fitted with an exponentially truncated power law distribution [6], whose probability density function is

$$p(x) \propto x^{-\alpha} e^{-\lambda x}, \tag{1}$$

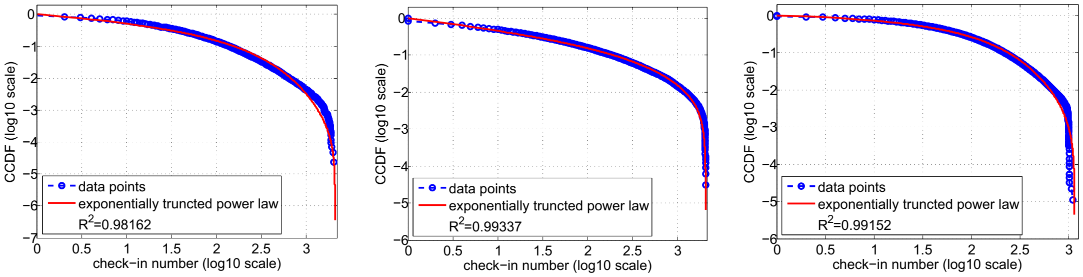
where α and λ are two parameters to be estimated for each dataset. Regarding this density function, we first give a property (Lemma 1) that will be used when we formally define the distribution (Lemma 2) and estimate the parameters (Lemma 3).

Lemma 1: Define

$$F_x(\alpha, \lambda) = \int_x^{+\infty} t^{-\alpha} e^{-\lambda t} dt.$$

Then

$$F_x(\alpha, \lambda) = \lambda^{\alpha-1} \Gamma(1-\alpha, \lambda x),$$



(a) Gowalla (b) Brightkite (c) Foursquare

Fig. 1. Exponentially truncated power law distribution of check-in numbers.

where $\Gamma(a, x)$ is the incomplete gamma function defined as

$$\Gamma(a, x) = \int_{x}^{+\infty} t^{a-1} e^{-t} dt.$$

Proof: Substituting λt in $F_x(\alpha, \lambda)$ by s, we have

$$F_x(\alpha, \lambda) = \int_{\lambda x}^{+\infty} (\frac{s}{\lambda})^{-\alpha} e^{-s} \cdot \frac{1}{\lambda} ds.$$

Rearranging the terms proves the lemma. ■

In practice, there exists a lower bound $x_{\min}$ and an upper bound $x_{\max}$ of the feasible x. Taking this into account, we derive the probability density function.

Lemma 2: The probability density function for variable $x \in [x_{\min}, x_{\max}]$ satisfying equation (1) is as follows.

$$p(x) = \frac{\lambda^{1-\alpha}}{\Gamma(1-\alpha, \lambda x_{\min}) - \Gamma(1-\alpha, \lambda x_{\max})} x^{-\alpha} e^{-\lambda x}.$$

Proof: Note that

$$\int_{x_{\min}}^{x_{\max}} x^{-\alpha} e^{-\lambda x} dx = \int_{x_{\min}}^{+\infty} x^{-\alpha} e^{-\lambda x} dx - \int_{x_{\max}}^{+\infty} x^{-\alpha} e^{-\lambda x} dx$$
$$= F_{x_{\min}}(\alpha, \lambda) - F_{x_{\max}}(\alpha, \lambda).$$

Thus, $\int_{x_{\min}}^{x_{\max}} p(x) dx = 1$. ■

To estimate the parameters α and λ, there are generally two methods, maximum likelihood estimation (MLE) and the moment method. We implemented both methods and found that they give similar results, while the moment method converges much faster. The underlying idea of the moment method is to equate the population moments with the sample moments, and solve the resulting equations. We only use the first and the second moments, since there are two parameters to be determined. The two population moments are stated in the following lemma.

Lemma 3: For the distribution defined in Lemma 2, we have

$$E[x] = \frac{\Gamma(2-\alpha, \lambda x_{\min}) - \Gamma(2-\alpha, \lambda x_{\max})}{\lambda(\Gamma(1-\alpha, \lambda x_{\min}) - \Gamma(1-\alpha, \lambda x_{\max}))}$$

and

$$E[x^2] = \frac{\Gamma(3-\alpha, \lambda x_{\min}) - \Gamma(3-\alpha, \lambda x_{\max})}{\lambda^2(\Gamma(1-\alpha, \lambda x_{\min}) - \Gamma(1-\alpha, \lambda x_{\max}))}.$$

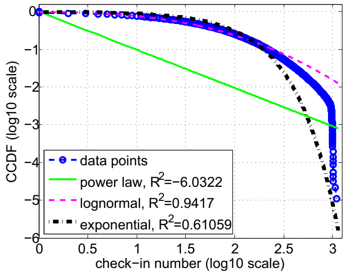


Fig. 2. Several other fits on the Foursquare check-in number data.

Proof: We only give the derivation for $E[x]$. The derivation for $E[x^2]$ is similar.

$$\begin{aligned} E[x] &= \int_{x_{\min}}^{x_{\max}} x \cdot p(x) dx \\ &= \int_{x_{\min}}^{x_{\max}} x \cdot \frac{\lambda^{1-\alpha} x^{-\alpha} e^{-\lambda x}}{\Gamma(1-\alpha, \lambda x_{\min}) - \Gamma(1-\alpha, \lambda x_{\max})} dx \\ &= \frac{\lambda^{1-\alpha} \int_{x_{\min}}^{x_{\max}} x^{-(\alpha-1)} e^{-\lambda x} dx}{\Gamma(1-\alpha, \lambda x_{\min}) - \Gamma(1-\alpha, \lambda x_{\max})} \\ &= \frac{\Gamma(2-\alpha, \lambda x_{\min}) - \Gamma(2-\alpha, \lambda x_{\max})}{\lambda(\Gamma(1-\alpha, \lambda x_{\min}) - \Gamma(1-\alpha, \lambda x_{\max}))}, \end{aligned}$$

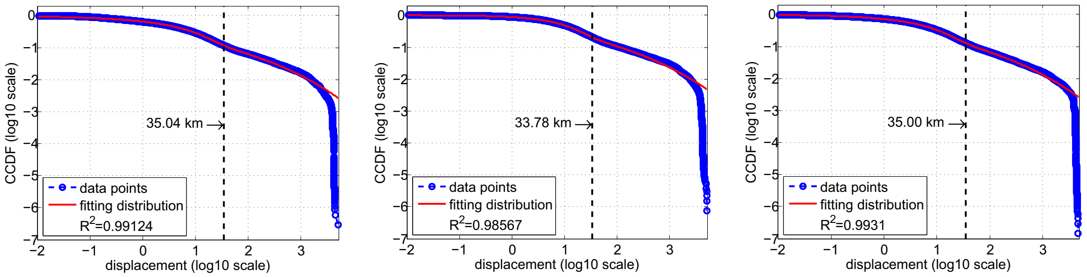
where the last equality can be obtained by Lemma 1 and the proof of Lemma 2. ■

Suppose d_i is user i's check-in number. With the moment method, the estimated two parameters $\hat{\alpha}$ and $\hat{\lambda}$ are the solution to the system of equations

$$\begin{cases} E[x] = \frac{1}{n} \sum_i d_i \\ E[x^2] = \frac{1}{n} \sum_i d_i^2. \end{cases} \tag{2}$$

Based on the parameters estimated with the moment method, we plot the CCDF of the fitting distributions in Figure 1. The figure shows that the exponentially truncated power law distribution fits the data well. To further investigate the goodness-of-fit, we use the coefficient of determination of data fit, also known as R^2, as an indicator of fitting errors. The closer R^2 is to 1, the better the distribution fits the data. As shown in Figure 1, the R^2 value is close to 1 for all the three datasets, indicating a good fit.

We have also tried to fit the data with other distributions. However, none of them fits the data better than the exponentially truncated power law distribution. For example, as shown in Figure 2, the R^2 values of the three fits on the Foursquare



(a) Gowalla (b) Brightkite (c) Foursquare

Fig. 3. Two-segment distribution of displacements (km).

dataset are all lower than the R^2 value of the exponentially truncated power law fit.

All the CCDF plots in Figure 1 show truncations of check-in numbers larger than several hundred, whose effects are sharp drops in the frequency of very large check-in numbers. One possible explanation of the exponential truncations is that the set of candidate venues a user can check-in is geographically constrained by factors like boundaries and physical obstructions. In addition, previous research has shown a check-in fatigue after prolonged use of LBSNs [11]. As a result, the frequency of very large check-in numbers in the datasets is lower than what would be in a power law distribution, whose CCDF is a straight line in the log-log scale.

B. Displacement of consecutive check-ins

In this subsection we study the spatial displacement of consecutive check-ins. We measure the distances between all the pairs of consecutive check-ins made by each user in the three datasets, and plot the log-log CCDF in Figure 3. It is easy to see that all the three plots consist of two segments with different curvature shapes that meet at a transition point. By analyzing the data we find that user displacements can be well fitted with an exponentially truncated power law distribution in the body, and a lognormal distribution in the tail. How to find the optimal transition point is shown as follows.

Definition 1: Denote by x_0 a transition point. An exponentially truncated power law distribution with lognormal in the tail is defined by the following probability density function

$$p(x) = \begin{cases} \beta \cdot \frac{\lambda^{1-\alpha}}{\Gamma(1-\alpha,\lambda x_{\min})-\Gamma(1-\alpha,\lambda x_0)} x^{-\alpha} e^{-\lambda x} & \text{if } x \le x_0 \\ (1-\beta) \cdot \frac{1}{(x-x_0)\sigma\sqrt{2\pi}} e^{\frac{-(\ln(x-x_0)-\mu)^2}{2\sigma^2}} & \text{if } x > x_0, \end{cases}$$

where β is the probability that $x \le x_0$. Denote by $p_{x_0^-}(x)$ the probability density for $x \le x_0$, and $p_{x_0^+}(x)$ the probability density for $x > x_0$.

The transition point x_0 is estimated with MLE. Denote by $d_1, d_2, \ldots, d_n$ the displacement data. Then, for any given x_0, the likelihood $L(x_0)$ can be computed as

$$L(x_0) = \prod_{d_i \le x_0} p_{x_0^-}(d_i) \cdot \prod_{d_i > x_0} p_{x_0^+}(d_i),$$

where parameters α, λ in $p_{x_0^-}$ are estimated by our previous method on set $\{d_i \mid d_i \le x_0\}$, parameters σ, μ in $p_{x_0^+}$ are estimated by the standard routine on set $\{d_i - x_0 \mid d_i > x_0\}$, and parameter β is simply the ratio $\frac{|\{d_i | d_i \le x_0\}|}{|\{d_i\}|}$. The estimated x_0 is the one that maximizes $L(x_0)$.

The estimated transition points of the Gowalla, Brightkite, and Foursquare datasets are shown in Figure 3, which match closely and are similar to the reach of an ordinary US city. The results indicate that user inter-checkin displacements exhibit two different behaviors: displacements within the reach of the borders of a city correspond to the daily short movements, and a vast majority of all the user displacements belong to this type, while displacements beyond the reach of the borders of a city correspond to the occasional long trips, and only a small fraction of the displacements fall into this category. Based on the estimated parameters, we plot the CCDF of the fitting distributions in Figure 3, which shows that the two-segment distribution is a good fit to the data.

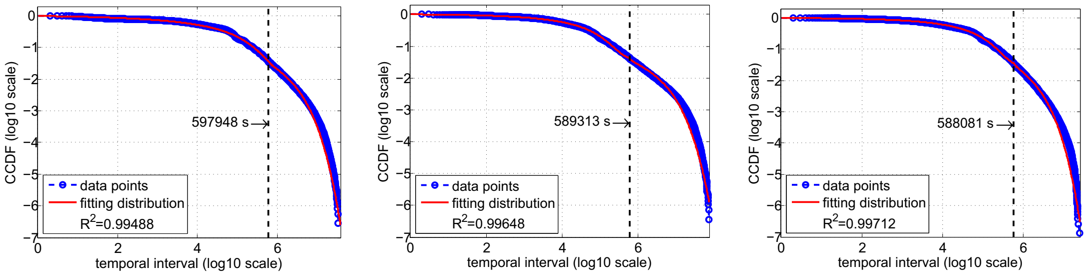
C. Temporal interval of consecutive check-ins

Next we study the temporal interval of consecutive check-ins. We plot in Figure 4 the CCDF of the temporal intervals between all the pairs of consecutive check-ins made by each user in the three datasets. Again, the plots exhibit a two-segment pattern. We find that small temporal intervals can be well fitted with an exponentially truncated power law distribution, while large temporal intervals can be well fitted with a Weibull distribution, which meet at a transition point.

We use the technique described in the previous subsection to estimate the optimal transition point. The results are shown in Figure 4. The transition points of all the three datasets match well, and they are close to the temporal length of a week in seconds (604,800s). This is a strong indicator that user check-in intervals exhibit two different patterns: the check-in intervals shorter than one week follow some weekly temporal rhythms, as shown in previous research [4], [15]. On the other hand, the check-in intervals longer than a week tend to arise from the more random check-ins made by users, e.g., when a user visits a new venue. We plot the CCDF of the fitting distributions based on the estimated parameters in Figure 4, which shows good fits to the original data with high R^2 values.

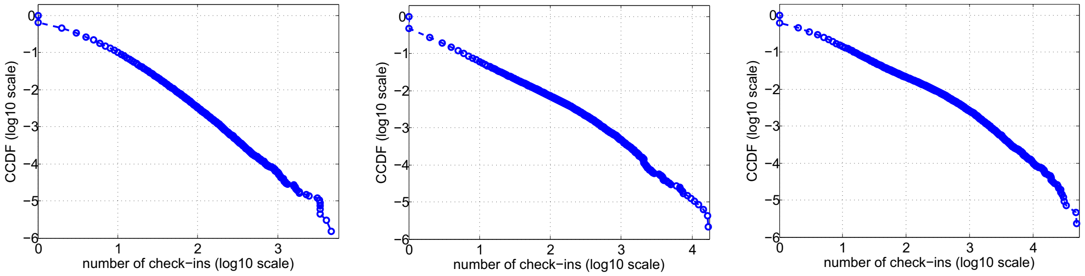
D. Distance between friends

In this subsection we study the geographic distance between friends in LBSNs. We consider two cases. One is we measure the distance between each pair of friends' first check-ins, i.e., their initial locations, and the other is we measure the



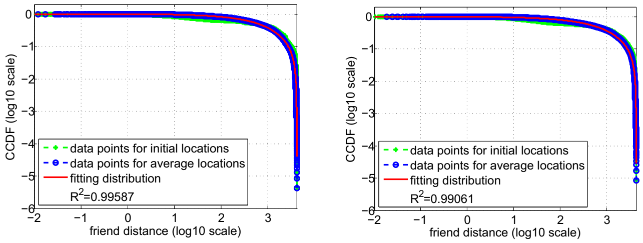
(a) Gowalla (b) Brightkite (c) Foursquare

Fig. 4. Two-segment distribution of temporal intervals (seconds).



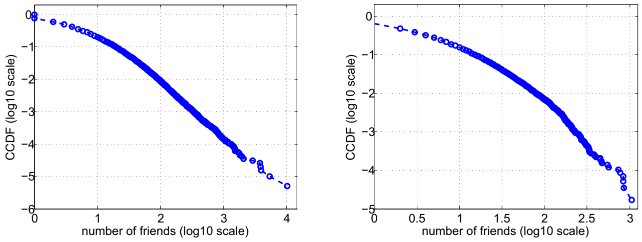
(a) Gowalla (b) Brightkite (c) Foursquare

Fig. 7. Power law distribution of venue popularity.



(a) Gowalla (b) Brightkite

Fig. 5. Truncated Weibull distribution of friend distances (km).



(a) Gowalla (b) Brightkite

Fig. 6. Power law distribution of friend numbers.

distance between each pair of friends' average locations over all their check-ins. We plot in Figure 5 the CCDF for both cases. Note that since the Foursquare dataset does not contain friendship information, we only plot the CCDF figure of the Gowalla and Brightkite datasets. Figure 5 shows that for both datasets the initial location curve and the average location curve match closely, indicating that there is no significant difference between these two measurements.

The downward CCDF curves can be well fitted by a truncated Weibull distribution, which is obtained by restricting the random variable of Weibull distribution within the range $(0, x_{max}]$, and normalizing the probability density function accordingly. Based on the parameters estimated with MLE, we plot the CCDF of the fitting distributions in Figure 5. The high R^2 values indicate a good fit. For comparison, we measure the distances between randomly selected 1,000,000 pairs of arbitrary users (strangers). The average distance between friends (1,040 km) is much smaller than the average distance between strangers (2,021 km), indicating that friendship tends to be established between geographically close users in LBSNs.

E. Number of friends

The degree (number of friends) distributions of the Gowalla and Brightkite datasets are reported in Figure 6. The CCDF is approximately a straight line in the log-log scale, which illustrates that user degrees follow a power law distribution. This result is consistent with the previous findings on online social networks [12], [16]: the majority of users have small degrees, while a small number of users have significantly larger degrees, which are the "hub" nodes in the social graph.

F. Venue popularity

We define the popularity of a venue as the number of check-ins made at this venue. To investigate the difference in popularity across all the venues, we plot in Figure 7 the CCDF of the number of check-ins at each venue in the three datasets. Again, the CCDF can be approximately fitted with a straight line in the log-log scale, indicating a power law distribution. The heavy tail of power law implies that only a few the most popular venues receive a large number of check-ins.

IV. MODELING LBSNS

Based on our findings, in this section we build LBSNSim, a trace-driven model that generates synthetic LBSN datasets

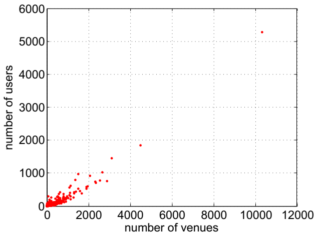


Fig. 8. The number of users versus the number of venues in each cell.

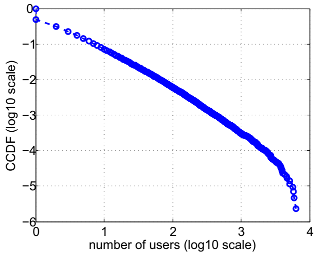


Fig. 9. Power-law venue popularity based on the first check-in of each user in the Foursquare dataset.

capturing the statistical features of the original datasets. We assume that the locations of a set of venues are known, which will be used as the input to our algorithm. The synthetic users' check-ins will adhere to these venues in the generated datasets. We believe that this is a realistic assumption, because releasing the venue information does not compromise any user privacy. Also, the information is readily available, which can be extracted from the published datasets.

The development of our model consists of three steps: (1) generating the initial location of each synthetic user; (2) building the friendship graph considering both social and geographic factors; (3) generating the check-ins of each user.

A. Generating the initial location

To generate the initial location, i.e., the location of the first check-in, of each user in the synthetic dataset, our algorithm relies on the hypothesis that the user density is proportional to the venue density in a given area. To verify this hypothesis, we discretize the mainland of the USA into 0.1° latitude by 0.1° longitude cells, and plot in Figure 8 the number of venues in each cell versus the number of users whose first check-in is in that cell, based on our Foursquare dataset. The figure signals a significant linear correlation (the correlation coefficient is 0.9324) between venue density and user density, and thus verifies our hypothesis. The other two datasets both exhibit a similar pattern. Our algorithm runs as follows.

Assume the set of venues in $cell(i,j)$ is V_{ij}, and the total number of synthetic users is n. Starting from the first to the n^{th} user, for each user, based on our finding, the probability that her initial location is in $cell(i,j)$ is $\frac{|V_{ij}|}{\sum_{i,j}|V_{ij}|}$. Suppose $cell(p,q)$ is selected. The probability of choosing venue $v \in V_{pq}$ as her initial location is proportional to $n_v + \epsilon$, i.e., $\frac{n_v+\epsilon}{\sum_{v \in V_{pq}} n_v + |V_{pq}|\epsilon}$, where n_v is the number of users who have chosen v as their initial location, and ϵ is a small constant. The plus-ϵ operation guarantees that the venues which have not been selected before still have an opportunity of being chosen.

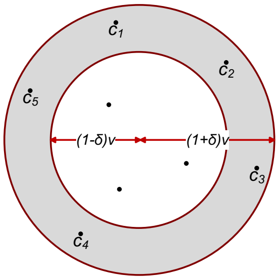


Fig. 10. Ring area to search for the destination node.

This generation process implies the richer-get-richer property: the larger number of users that have chosen a venue as their initial location, the higher probability that the next user will select this venue as her initial location. This is consistent with the power law distribution of the number of users that have chosen each venue as their initial location in the original dataset, as shown in Figure 9.

B. Building the friendship graph

The second step is to build the friendship graph. As shown in the previous section, in the real datasets user degrees follow a power law distribution, and the distances between friends follow a truncated Weibull distribution. The generated friendship graph should preserve both properties.

We use an extended preferential attachment process, similar to the one proposed by Capocci et al. [2], to reproduce the power law degree distribution.

Assume the total number of friendship edges to create is e. Each user in a social graph is represented as a node. The process starts with an initial set of m_0 nodes with $m_0 > e/n$, where n is the total number of synthetic users. A clique topology is generated among those m_0 nodes, i.e., each node is linked to the other $m_0 - 1$ nodes. These are the startup nodes in the social graph. For example, they may be the administrators of the LBSN.

Starting from the $(m_0+1)^{th}$ user to the n^{th} user, at each step, a new node and e/n new edges are introduced into the social graph. For each new edge, with probability p, the edge connects the new node with an existing node. With probability $1-p$, the edge originates from an existing node and ends at another existing node, and the probability of choosing an existing node i as the source node is proportional to i's degree, i.e., $\frac{d_i}{\sum_i d_i}$. In both cases, how to select the destination node is explained below.

Given a source node, to choose the destination node of a friendship edge, we first randomly sample a distance v from the truncated Weibull fitting distribution of the distances between friends' initial locations, which is acquired in Section III-D. If $v \geq \tau$, as shown in Figure 10, we draw a circular ring area on the map centered at the initial location of the source node, whose inner radius is $(1-\delta)v$, and outer radius is $(1+\delta)v$, where δ is a tunable parameter. All the nodes whose initial location is within this ring area are treated as candidate nodes, from which the destination node will be chosen. Otherwise, if $v < \tau$, all the nodes whose initial location falls into the circular area centered at the initial location of the source node and with radius v are considered as candidate nodes. τ is a small threshold distance, which is set to 0.5 km in our experiments. Assume the set of candidate nodes is C. The probability of choosing C_i, the i^{th} node in C, as the destination node is proportional to its degree, i.e., $\frac{d_{C_i}}{\sum_{C_i \in C} d_{C_i}}$. If $|C| = 0$, we sample a new distance and repeat the process.

C. Generating the check-ins

The last step of our model is to generate all the check-ins of each user. Note that the location of each user's first check-in has already been generated in the first step of our model.

As mentioned in the previous section, each check-in is composed of the location information, represented as a venue ID, and a timestamp. Our task is to generate both pieces of information for each check-in, such that the synthetic check-in traces capture the properties of the original traces.

Our algorithm runs by first assigning a check-in number to each user. For each synthetic user, we randomly sample a check-in number from the exponentially truncated power law fitting distribution of user check-in numbers acquired in Section III-A. To generate timestamps of these check-ins, we first need to generate the timestamp of each user's first check-in. To achieve this we sorted the timestamps of all the users' first check-ins in the real dataset in increasing order, and found that the CCDF of the temporal intervals between consecutive timestamps in this sorted list can be approximated by a power law distribution. Given the timestamp of the first synthetic user's first check-in, with the power-law parameters estimated by MLE, we are able to generate a sequence of timestamps of the other synthetic users' first check-ins, such that the temporal intervals follow the power law distribution.

Assume the timestamp of user i's first check-in is t_{c_1}, and her check-in number is n_i. To generate the timestamps of i's following check-ins, we randomly sample $n_i - 1$ temporal intervals $v_1, ..., v_{n_i - 1}$ from the two-segment fitting distribution of the temporal intervals between consecutive check-ins made by users, which is acquired in Section III-C. The timestamp of the j^{th} check-in made by user i is thus: $t_{c_j} = t_{c_1} + \sum_{k=1}^{j-1} v_k$.

Now we have generated the timestamps of all the check-ins, and our next task is to generate the location of each check-in, i.e., to determine which venue the check-in adheres to. We achieve this by first sorting all the check-ins by their timestamps in increasing order. Starting from the first check-in to the last check-in, for each encountered check-in c, we check if c is its creator u's first check-in. If so, then c's location is the same as u's initial location. Otherwise, we randomly sample a displacement v from the two-segment fitting distribution of the displacements between consecutive check-ins made by users, which is acquired in Section III-B. Assume c is u's i^{th} check-in, where $i > 1$. If $v \geq \tau$, we draw a circular ring area on the map centered at the location of u's $(i-1)^{th}$ check-in, whose inner radius is $(1-\delta)v$, and outer radius is $(1+\delta)v$. All the venues falling into this ring area are treated as candidate venues that c may adhere to. If $v < \tau$, we consider all the venues falling into the circular area centered at the location of u's $(i-1)^{th}$ check-in and with radius v as candidate venues.

Assume the set of candidate venues is V. We consider two cases when determining c's venue. Let V_f be the set of venues in V that have been previously checked-in by some of u's friends, and $V_s = V - V_f$. If both V_f and V_s are not empty, then with probability p', c adheres to a venue in V_f. The probability that c adheres to venue i in V_f is proportional to the number of times that u's friends have previously checked-in at this venue, i.e., $\frac{f_i}{\sum_{i \in V_f} f_i}$. With probability $1 - p'$, c adheres to a venue in V_s. The probability that c adheres to venue i in V_s is proportional to ϵ plus the number of times that the non-friend users (strangers) have previously checked-in at this venue, i.e., $\frac{s_i + \epsilon}{\sum_{i \in V_s} s_i + |V_s|\epsilon}$. The plus-$\epsilon$ operation guarantees that the venues which have not been checked-in by any user before still have an opportunity of being chosen. By introducing the parameter p', we take into account the difference between check-ins made by friends and by strangers. This captures the social influence on users' check-in behavior, which was found in previous research [5], [7]. If either V_f or V_s is empty, then we only consider the venues in the non-empty set. If both V_f and V_s are empty, we sample a new distance and repeat the process.

Note that in the original datasets we do not observe an obvious correlation between the displacement and the temporal interval of consecutive check-ins. The correlation coefficients of the Gowalla, Brightkite, and Foursquare datasets are 0.1306, 0.0972, and 0.1256, respectively. This finding is rational because check-in is a spontaneous user behavior, which is different from continuous location sensing. It may take a long time for a user to make two check-ins, while the displacement between them may be very small. Therefore, when building the model we do not consider this correlation. Instead, we only filter out the generated consecutive check-ins which imply an unrealistically high transit speed.

V. MODEL VERIFICATION

In this section, we evaluate the fidelity of LBSNSim by verifying whether LBSNSim can generate synthetic LBSN datasets that capture the statistical features observed in the original datasets.

A. Experimental Setup

In the evaluation we use the Gowalla dataset as the target dataset. The parameters of the distributions used in our model are estimated based on this dataset. The number of users in the model-generated datasets is 100,000, the number of friendship edges is 500,000, and the total number of check-ins is 5,000,000. The other parameters used in the model are set as: $m_0 = 20, p = 0.4, p' = 0.8, \delta = 0.25, \epsilon = 1, \tau = 0.5$ km. We use the venues extracted from the Gowalla dataset as the input to our algorithm. Note that using LBSNSim, researchers can generate LBSN datasets with different scales and properties by tuning the parameters in the model, and experiment with these datasets to produce statistically confident results.

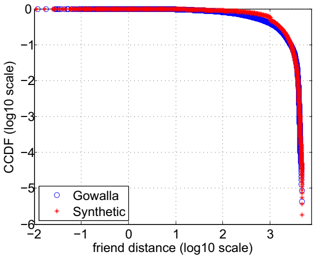


Fig. 11. Friend distances (km) based on average locations.

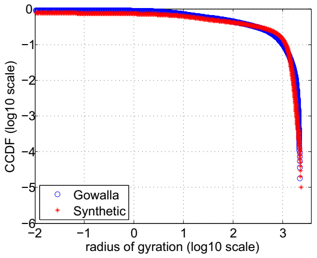


Fig. 12. Radius of gyration (km).

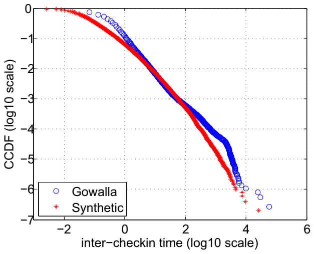


Fig. 13. Inter-checkin time (seconds) of all the check-ins, rescaled by dividing by the average time interval.

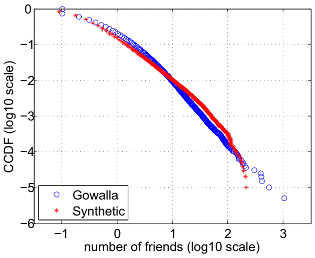


Fig. 14. Number of friends of each user, rescaled by dividing by the average friend number.

B. *Evaluation Results*

We have tested a wide range of statistical features, including distance between friends, radius of gyration (explained later), temporal intervals considering all the check-ins, node degrees, venue popularity, and social influence on check-ins. We do not test the features that are explicitly modeled by LBSNSim, including the exponentially truncated power law check-in number distribution, and the two-segment distributions of the temporal intervals and displacements between consecutive check-ins made by each user. In the experiments we generate 50 realizations of our model and we observe no significant difference among them in the statistical features we evaluate, so for each feature we compare the target dataset with a randomly selected model-generated dataset. In the application-level test, we run an LBSN-based application with both the target dataset and the synthetic datasets as input, and compare the results to quantify the fidelity of LBSNSim.

Distance between friends. In the previous section, we use distance as a constraint on friendship generation, by sampling a distance v from the truncated Weibull fitting distribution of the distances between friends' initial locations each time when a new friendship edge is created, and using it to search for the destination node of the edge. However, the resulting distance between the source node and the destination node is only an approximation of v, as the destination node is chosen from all the nodes falling into the ring area with inner radius $(1-\delta)v$ and outer radius $(1+\delta)v$. As shown in Figure 11, even with this approximation, the distribution of the distances between friends' average locations in the model-generated dataset still matches well with that in the target dataset.

Radius of gyration. The radius of gyration of a user is defined as the root mean square distance of a user's check-ins from their center of mass:

$$R_g = \sqrt{\frac{1}{n_i}\sum_{j=1}^{n_i}(distance(c_j, c_a))^2},$$

where n_i is user i's check-in number, c_j is the location of the j^{th} check-in, and c_a is the average location over all the check-ins. Radius of gyration measures the "spread" of a user's check-ins: the larger radius of gyration is, the more widely a user's check-ins are dispersed. Figure 12 compares the distribution of radius of gyration in the model-produced dataset with that extracted from the target dataset, which shows that the two distributions match well.

Inter-checkin time of all the check-ins. To measure the distribution of the inter-checkin time considering all the check-ins, we sort all the user check-ins by their timestamps in increasing order. Figure 13 plots the distribution of the temporal interval between every pair of consecutive check-ins in this sorted list, for both the synthetic dataset and the target dataset. The CCDFs of both distributions can be approximated by a straight line in the log-log scale, which indicates a power law distribution, and they also match closely. Note that this distribution is different from the distribution of the temporal intervals between consecutive check-ins made by each user, which has been studied in the data analysis section.

Number of friends. Figure 14 compares the distribution of the friend numbers in the model-produced dataset with that in the target dataset. Both CCDF curves are approximately a straight line in the log-log scale with similar slopes. This validates the correctness of our extended preferential attachment process used to generate the friendship graph.

Venue popularity. In Figure 15 we plot the distribution of venue popularity, defined as the number of check-ins at each venue, in the generated dataset and in the target dataset. The figure illustrates that our check-in generation method captures the power-law venue popularity found in the original datasets.

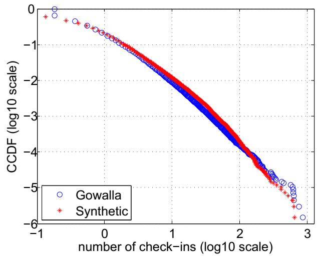


Fig. 15. Venue popularity, rescaled by dividing by the average venue popularity.

TABLE II
FALSE POSITIVE AND NEGATIVE RATES OF SYBILDEFENDER

	Original		Synthetic	
	F^+	F^-	F^+	F^-
2000 sybil nodes	0.3%	0.2%	0.4%	0.2%
1000 sybil nodes	0.3%	0.4%	0.3%	0.5%
500 sybil nodes	0.2%	0.6%	0.2%	0.8%

Social influence on check-ins. To quantify the social influence on users' check-in behavior, we compare the probability that two friends have checked-in at the same venue with the probability that two strangers have checked-in at the same venue. Each probability is measured by randomly selecting 100,000 pairs of users and counting the number of pairs that have checked-in at at least one common venue. All the results are averaged over 50 runs. The friend-case and stranger-case probabilities for the Gowalla dataset are 16.29% and 0.39%, respectively. For the model-generated dataset they are 14.50% and 0.59%. The results demonstrate that similar to the real datasets, the model-generated datasets exhibit strong social influence on users' check-in behavior, i.e., people are more likely to visit places that their friends visited in the past.

Application-level test. In this subsection, we compare the results of an LBSN-based application obtained by using the target dataset with those obtained by using the synthetic datasets. SybilDefender was proposed in our previous research to defend against sybil attacks in social networks [20], when an attacker creates many bogus identities to compromise the operation of the system. To extend SybilDefender to LBSNs, we augment the sybil identification algorithm by considering edge weights, which are defined as the number of venues that have been checked-in by both ending nodes of a friendship edge. The algorithm runs by performing weighted random walks in the friendship graph. At each step of a weighted random walk, edge weights are considered when choosing the next hop. The weighted sybil identification algorithm takes the target dataset and the synthetic datasets as input. In each experiment we randomly create sybil nodes forming a connected scale-free topology, with a small number of edges linking to the largest connected component of the friendship graph. The results are averaged over 50 runs.

Table II shows the average false positive and negative rates of the weighted sybil identification algorithm, when running on the target dataset and on the synthetic datasets. It demonstrates that the application-level results obtained by using the synthetic datasets closely match those obtained by using the original dataset.

VI. CONCLUSION

In this paper, we analyze the statistical features extracted from the data of three LBSNs, and propose LBSNSim, a trace-driven model for generating synthetic LBSN datasets that capture the properties of the original datasets. Evaluation shows that the synthetic datasets generated by LBSNSim are sufficiently representative of real-world LBSN datasets in a wide range of statistical features, and high fidelity results can be produced using the synthetic datasets in the application-level test. This verifies the feasibility of using the model-generated datasets as replacements for real LBSN datasets.

ACKNOWLEDGMENT

This project was supported in part by US National Science Foundation grants CNS-1320453, CNS-1117412, and CAREER Award CNS-0747108. The work of Xiaojun Zhu was partly supported by the program B for Outstanding PhD candidate of Nanjing University (201301B014).

REFERENCES

[1] About Foursquare. https://foursquare.com/about.
[2] A. Capocci, V. D. P. Servedio, F. Colaiori, L. S. Buriol, D. Donato, S. Leonardi, and G. Caldarelli. Preferential attachment in the growth of social networks: The internet encyclopedia wikipedia. *Physics Review E*, 74, 2006.
[3] C. Cheng, H. Yang, I. King, and M. R. Lyu. Fused matrix factorization with geographical and social influence in location-based social networks. In *AAAI*, 2012.
[4] Z. Cheng, J. Caverlee, K. Lee, and D. Z. Sui. Exploring millions of footprints in location sharing services. In *ICWSM*, 2011.
[5] E. Cho, S. A. Myers, and J. Leskovec. Friendship and mobility: User movement in location-based social networks. In *KDD*, 2011.
[6] A. Clauset, C. R. Shalizi, and M. E. J. Newman. Power-law distributions in empirical data. *SIAM Review*, 51(4):661–703, 2009.
[7] H. Gao, J. Tang, and H. Liu. Exploring social-historical ties on location-based social networks. In *ICWSM*, 2012.
[8] J. J. Pfeiffer III, T. La Fond, S. Moreno, and J. Neville. Fast generation of large scale social networks while incorporating transitive closures. In *SocialCom*, 2012.
[9] M. Kim and J. Leskovec. Modeling social networks with node attributes using the multiplicative attribute graph model. In *UAI*, 2011.
[10] J. Krumm. Inference attacks on location tracks. In *Pervasive*, 2007.
[11] J. Lindqvist, J. Cranshaw, J. Wiese, J. Hong, and J. Zimmerman. I'm the mayor of my house: Examining why people use Foursquare - a social-driven location sharing application. In *CHI*, 2011.
[12] A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, and B. Bhattacharjee. Measurement and analysis of online social networks. In *IMC*, 2007.
[13] A. Narayanan and V. Shmatikov. De-anonymizing social networks. In *IEEE S&P*, 2009.
[14] A. Noulas, S. Scellato, N. Lathia, and C. Mascolo. A random walk around the city: New venue recommendation in location-based social networks. In *SocialCom*, 2012.
[15] A. Noulas, S. Scellato, C. Mascolo, and M. Pontil. An empirical study of geographic user activity patterns in Foursquare. In *ICWSM*, 2011.
[16] A. Sala, L. Cao, C. Wilson, R. Zablit, H. Zheng, and B. Y. Zhao. Measurement-calibrated graph models for social network experiments. In *WWW*, 2010.
[17] A. Sala, X. Zhao, C. Wilson, H. Zheng, and B. Y. Zhao. Sharing graphs using differentially private graph models. In *IMC*, 2011.
[18] S. Scellato, A. Noulas, R. Lambiotte, and C. Mascolo. Socio-spatial properties of online location-based social networks. In *ICWSM*, 2011.
[19] S. Scellato, A. Noulas, and C. Mascolo. Exploiting place features in link prediction on location-based social networks. In *KDD*, 2011.
[20] W. Wei, F. Xu, C. C. Tan, and Q. Li. Sybildefender: Defend against sybil attacks in large social networks. In *INFOCOM*, 2012.