

Differentially Private Publication of Location Entropy

Hien To *
hto@usc.edu

Kien Nguyen *
kien.nguyen@usc.edu

Cyrus Shahabi
shahabi@usc.edu

Department of Computer Science, University of Southern California
Los Angeles, CA 90089

ABSTRACT

Location entropy (LE) is a popular metric for measuring the popularity of various locations (e.g., points-of-interest). Unlike other metrics computed from only the number of (unique) visits to a location, namely *frequency*, LE also captures the *diversity* of the users' visits, and is thus more accurate than other metrics. Current solutions for computing LE require full access to the past visits of users to locations, which poses privacy threats. This paper discusses, for the first time, the problem of perturbing location entropy for a set of locations according to differential privacy. The problem is challenging because removing a single user from the dataset will impact multiple records of the database; i.e., all the visits made by that user to various locations. Towards this end, we first derive non-trivial, tight bounds for both local and global sensitivity of LE, and show that to satisfy ϵ -differential privacy, a large amount of noise must be introduced, rendering the published results useless. Hence, we propose a thresholding technique to limit the number of users' visits, which significantly reduces the perturbation error but introduces an approximation error. To achieve better utility, we extend the technique by adopting two weaker notions of privacy: smooth sensitivity (slightly weaker) and crowd-blending (strictly weaker). Extensive experiments on synthetic and real-world datasets show that our proposed techniques preserve original data distribution without compromising location privacy.

Categories and Subject Descriptors

H.2.4 [Database Management]: Database Applications—*Spatial databases and GIS*; H.1.1 [Models and Principles]: Systems and Information Theory—*Information theory*

Keywords

Differential Privacy, Location Entropy

1. INTRODUCTION

*These authors contributed equally to this work.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

SIGSPATIAL '16, October 31–November 03, 2016, Burlingame, CA, USA

© 2016 ACM. ISBN 978-1-4503-4589-7/16/10...\$15.00

DOI: <http://dx.doi.org/10.1145/2996913.2996985>

Due to the pervasiveness of GPS-enabled mobile devices and the popularity of location-based services such as mapping and navigation apps (e.g., Google Maps, Waze), or spatial crowdsourcing apps (e.g., Uber, TaskRabbit), or apps with geo-tagging (e.g., Twitter, Picasa, Instagram, Flickr), or check-in functionality (e.g., Foursquare, Facebook), numerous industries are now collecting fine-grained location data from their users. While the collected location data can be used for many commercial purposes by these industries (e.g., geo-marketing), other companies and non-profit organizations (e.g., academia, CDC) can also be empowered if they can use the location data for the greater good (e.g., research, preventing the spread of disease). Unfortunately, despite the usefulness of the data, industries do not publish their location data due to the sensitivity of their users' location information. However, many of these organizations do not need access to the raw location data but aggregate or processed location data would satisfy their need.

One example of using location data is to measure the popularity of a location that can be used in many application domains such as public health, criminology, urban planning, policy, and social studies. One accepted metric to measure the popularity of a location is location entropy (or LE for short). LE captures both the frequency of visits (how many times each user visited a location) as well as the diversity of visits (how many unique users visited a location) without looking at the functionality of that location; e.g., is it a private home or a coffee shop? Hence, LE has shown that it is able to better quantify the popularity of a location as compared to the number of unique visits or the number of check-ins to the location [4]. For example, [4] shows that LE is more successful in accurately predicting friendship from location trails over simpler models based only on the number of visits. LE is also used to improve online task assignment in spatial crowdsourcing [12, 23] by giving priority to workers situated in less popular locations because there may be no available worker visiting those locations in the future.

Obviously, LE can be computed from raw location data collected by various industries; however, the raw data cannot be published due to serious location privacy implications [10, 5, 21]. Without privacy protection, a malicious adversary can stage a broad spectrum of attacks such as physical surveillance and stalking, and breach of sensitive information such as an individual's health issues (e.g., presence in a cancer treatment center), alternative lifestyles, political and religious preferences (e.g., presence in a church). Hence, in this paper we propose an approach based on differential privacy (DP) [6] to publish LE for a set of locations without compromising users' raw location data. DP has emerged as the de facto standard with strong protection guarantees for

publishing aggregate data. It has been adapted by major industries for various tasks without compromising individual privacy, e.g., data analytics with Microsoft [15], discovering users’ usage patterns with Apple¹, or crowdsourcing statistics from end-user client software [8] and training of deep neural networks [1] with Google. DP ensures that an adversary is not able to reliably learn from the published sanitized data whether or not a particular individual is present in the original data, regardless of the adversary’s prior knowledge.

It is sufficient to achieve ϵ -DP (ϵ is privacy loss) by adding Laplace *noise* with mean zero and scale proportional to the *sensitivity* of the query (LE in this study) [6]. The sensitivity of LE is intuitively the maximum amount that one individual can impact the value of LE. The higher the sensitivity, the more noise must be injected to guarantee ϵ -DP. Even though DP has been used before to compute Shannon Entropy [2] (the formulation adapted in LE), the main challenge in differentially private publication of LE is that adding (or dropping) a single user from the dataset would impact multiple entries of the database, resulting in a high sensitivity of LE. To illustrate, consider a user that has contributed many visits to a single location; thus, adding or removing this user would significantly change the value of LE for that location. Alternatively, a user may contribute visits to multiple locations and hence impact the entropy of all those visited locations. Another unique challenge in publishing LE (vs. simply computing the Shannon Entropy) is due to the presence of skewness and sparseness in real-world location datasets where the majority of locations have small numbers of visits.

Towards this end, we first compute a non-trivial tight bound for the global sensitivity of LE. Given the bound, a sufficient amount of noise is introduced to guarantee ϵ -DP. However, the injected noise linearly increases with the maximum number of locations visited by a user (denoted by M) and monotonically increases with the maximum number of visits a user contributes to a location (denoted by C), and such an excessive amount of noise renders the published results useless. We refer to this algorithm as *BASELINE*. Accordingly, we propose a technique, termed *LIMIT*, to limit user activity by thresholding M and C , which significantly reduces the perturbation error. Nevertheless, limiting an individual’s activity entails an approximation error in calculating LE. These two conflicting factors require the derivation of appropriate values for M and C to obtain satisfactory results. We empirically find such optimal values.

Furthermore, to achieve a better utility, we extend *LIMIT* by adopting two weaker notions of privacy: smooth sensitivity [16] and crowd-blending [9] (strictly weaker). We denote the techniques as *LIMIT-SS* and *LIMIT-CB*, respectively. *LIMIT-SS* provides a slightly weaker privacy guarantee, i.e., (ϵ, δ) -differential privacy by using local sensitivity with much smaller noise magnitude. We propose an efficient algorithm to compute the local sensitivity of a particular location that depends on C and the number of users visiting the location (represented by n) such that the local sensitivity of all locations can be precomputed, regardless of the dataset. Thus far, we publish entropy for all locations; however, the ratio of noise to the true value of LE (noise-to-true-entropy ratio) is often excessively high when the number of users visiting a location n is small (i.e., the entropy of a

location is bounded by $\log(n)$). For example, given a location visited by only two users with an equal number of visits (LE is $\log 2$), removing one user from the database drops the entropy of the location to zero. To further reduce the noise-to-true-entropy ratio, *LIMIT-CB* aims to publish the entropy of locations with at least k users ($n \geq k$) and suppress the other locations. By thresholding n , the global sensitivity of LE significantly drops, implying much less noise. We prove that *LIMIT-CB* satisfies (k, ϵ) -crowd-blending privacy.

We conduct an extensive set of experiments on both synthetic and real-world datasets. We first show that the truncation technique (*LIMIT*) reduces the global sensitivity of LE by two orders of magnitude, thus greatly enhancing the utility of the perturbed results. We also demonstrate that *LIMIT* preserves the original data distribution after adding noise. Thereafter, we show the superiority of *LIMIT-SS* and *LIMIT-CB* over *LIMIT* in terms of achieving higher utility (measured by KL-divergence and mean squared error metrics). Particularly, *LIMIT-CB* performs best on sparse datasets while *LIMIT-SS* is recommended over *LIMIT-CB* on dense datasets. We also provide insights on the effects of various parameters: ϵ, C, M, k on the effectiveness and utility of our proposed algorithms. Based on the insights, we provide a set of guidelines for choosing appropriate algorithms and parameters.

The remainder of this paper is organized as follows. In Section 2, we define the problem of publishing LE according to differential privacy. Section 3 presents the preliminaries. Section 4 introduces the baseline solution and our thresholding technique. Section 5 presents our utility enhancements by adopting weaker notions of privacy. Experimental results are presented in Section 6, followed by a survey of related work in Section 7, and conclusions in Section 8.

2. PROBLEM DEFINITION

In this section we present the notations and the formal definition of the problem.

Each location l is represented by a point in two-dimensional space and a unique identifier l ($-180 \leq l_{lat} \leq 180$) and ($-90 \leq l_{lon} \leq 90$)². Hereafter, l refers to both the location and its unique identifier. For a given location l , let O_l be the set of visits to that location. Thus, $c_l = |O_l|$ is the total number of visits to l . Also, let U_l be the set of distinct users that visited l , and $O_{l,u}$ be the set of visits that user u has made to the location l . Thus, $c_{l,u} = |O_{l,u}|$ denotes the number of visits of user u to location l . The probability that a random draw from O_l belongs to $O_{l,u}$ is $p_{l,u} = \frac{|c_{l,u}|}{|c_l|}$, which is the fraction of total visits to l that belongs to user u . The location entropy for l is computed from Shannon entropy [18] as follows:

$$H(l) = H(p_{l,u_1}, p_{l,u_2}, \dots, p_{l,u_{|U_l|}}) = - \sum_{u \in U_l} p_{l,u} \log p_{l,u} \quad (1)$$

In our study the natural logarithm is used. A location has a higher entropy when the visits are distributed more evenly among visiting users, and vice versa. Our goal is to publish location entropy of all locations $L = \{l_1, l_2, \dots, l_{|L|}\}$, where each location is visited by a set of users $U = \{u_1, u_2, \dots, u_{|U|}\}$, while preserving the location privacy of users. Table 1 summarizes the notations used in this paper.

¹<https://www.wired.com/2016/06/apples-differential-privacy-collecting-data/>

² l_{lat}, l_{lon} are real numbers with ten digits after the decimal point.

$l, L, L $	a location, the set of all locations and its cardinality
$H(l)$	location entropy of location l
$\tilde{H}(l)$	noisy location entropy of location l
ΔH_l	sensitivity of location entropy for location l
ΔH	sensitivity of location entropy for all locations
O_l	the set of visits to location l
$u, U, U $	a user, the set of all users and its cardinality
U_l	the set of distinct users who visits l
$O_{l,u}$	the set of visits that user u has made to location l
c_l	the total number of visits to l
$c_{l,u}$	the number of visits that user u has made to location l
C	maximum number of visits of a user to a location
M	maximum number of locations visited by a user
$p_{l,u}$	the fraction of total visits to l that belongs to user u

Table 1: Summary of notations.

3. PRELIMINARIES

We present Shannon entropy properties and the differential privacy notion that will be used throughout the paper.

3.1 Shannon Entropy

Shannon [18] introduces entropy as a measure of the uncertainty in a random variable with a probability distribution $U = (p_1, p_2, \dots, p_{|U|})$:

$$H(U) = - \sum_i p_i \log p_i \quad (2)$$

where $\sum_i p_i = 1$. $H(U)$ is maximal if all the outcomes are equally likely:

$$H(U) \leq H\left(\frac{1}{|U|}, \dots, \frac{1}{|U|}\right) = \log |U| \quad (3)$$

Additivity Property of Entropy: Let U_1 and U_2 be non-overlapping partitions of a database U including users who contribute visits to a location l , and ϕ_1 and ϕ_2 are probabilities that a particular visit belongs to partition U_1 and U_2 , respectively. Shannon discovered that using logarithmic function preserves the *additivity* property of entropy:

$$H(U) = \phi_1 H(U_1) + \phi_2 H(U_2) + H(\phi_1, \phi_2)$$

Subsequently, adding a new person u into U changes its entropy to:

$$H(U^+) = \frac{c_l}{c_l + c_{l,u}} H(U) + H\left(\frac{c_{l,u}}{c_l + c_{l,u}}, \frac{c_l}{c_l + c_{l,u}}\right) \quad (4)$$

where $U^+ = U \cup u$ and c_l is the total number of visits to l , and $c_{l,u}$ is the number of visits to l that is contributed by user u . Equation (4) can be derived from Equation (4) if we consider U^+ includes two non-overlapping partitions u and U with associated probabilities $\frac{c_{l,u}}{c_l + c_{l,u}}$ and $\frac{c_l}{c_l + c_{l,u}}$. We note that the entropy of a single user is zero, i.e., $H(u) = 0$.

Similarly, removing a person u from U changes its entropy as follows:

$$H(U^-) = \frac{c_l}{c_l - c_{l,u}} \left(H(U) - H\left(\frac{c_{l,u}}{c_l}, \frac{c_l - c_{l,u}}{c_l}\right) \right) \quad (5)$$

where $U^- = U \setminus \{u\}$.

3.2 Differential Privacy

Differential privacy (DP) [6] has emerged as the de facto standard in data privacy, thanks to its strong protection guarantees rooted in statistical analysis. DP is a *semantic* model which provides protection against realistic adversaries with background information. Releasing data according to

DP ensures that an adversary's chance of inferring any information about an individual from the sanitized data will not substantially increase, regardless of the adversary's prior knowledge. DP ensures that the adversary does not know whether an individual is present or not in the original data. DP is formally defined as follows.

DEFINITION 1. ϵ -INDISTINGUISHABILITY [7] *Consider that a database produces a set of query results \hat{D} on the set of queries $Q = \{q_1, q_2, \dots, q_{|Q|}\}$, and let $\epsilon > 0$ be an arbitrarily small real constant. Then, transcript U produced by a randomized algorithm A satisfies ϵ -indistinguishability if for every pair of sibling datasets D_1, D_2 that differ in only one record, it holds that*

$$\ln \frac{\Pr[Q(D_1) = U]}{\Pr[Q(D_2) = U]} \leq \epsilon$$

In other words, an attacker cannot reliably learn whether the transcript was obtained by answering the query set Q on dataset D_1 or D_2 . Parameter ϵ is called *privacy budget*, and specifies the amount of protection required, with smaller values corresponding to stricter privacy protection. To achieve ϵ -indistinguishability, DP injects noise into each query result, and the amount of noise required is proportional to the *sensitivity* of the query set Q , formally defined as:

DEFINITION 2 (L_1 -SENSITIVITY). [7] *Given any arbitrary sibling datasets D_1 and D_2 , the sensitivity of query set Q is the maximum change in their query results.*

$$\sigma(Q) = \max_{D_1, D_2} \|Q(D_1) - Q(D_2)\|_1$$

An essential result from [7] shows that a sufficient condition to achieve DP with parameter ϵ is to add to each query result randomly distributed Laplace noise with mean 0 and scale $\lambda = \sigma(Q)/\epsilon$.

4. PRIVATE PUBLICATION OF LE

In this section we present a baseline algorithm based on a global sensitivity of LE [7] and then introduce a thresholding technique to reduce the global sensitivity by limiting an individual's activity.

4.1 Global Sensitivity of LE

To achieve ϵ -differential privacy, we must add noise proportional to the global sensitivity (or sensitivity for short) of LE. Thus, to minimize the amount of injected noise, we first propose a tight bound for the sensitivity of LE, denoted by ΔH . ΔH represents the maximum change of LE across all locations when the data of one user is added (or removed) from the dataset. With the following theorem, the sensitivity bound is a function of the maximum number of visits a user contributes to a location, denoted by C ($C \geq 1$).

THEOREM 1. Global sensitivity of location entropy is

$$\Delta H = \max \{ \log 2, \log C - \log(\log C) - 1 \}$$

PROOF. We prove this theorem by first deriving a tight bound for the sensitivity of a particular location l (visited by n users), denoted by ΔH_l (Theorem 2). The bound is a function of C and n . Thereafter, we generalize the bound to hold for all locations as follows. We take the derivative of the bound derived for ΔH_l with respect to variable n and find the extremal point where the bound is maximized. The detailed proof can be found in our technical report [22]. \square

THEOREM 2. *Local sensitivity of a particular location l with n users is:*

- $\log 2$ when $n = 1$
- $\log \frac{n+1}{n}$ when $C = 1$
- $\max\{\log \frac{n-1}{n-1+C} + \frac{C}{n-1+C} \log C, \log \frac{n}{n+C} + \frac{C}{n+C} \log C, \log(1 + \frac{1}{\exp(H(C \setminus c_u))})\}$ where C is the maximum number of visits a user contributes to a location ($C \geq 1$) and $H(C \setminus c_u) = \log(n-1) - \frac{\log C}{C-1} + \log(\frac{\log C}{C-1}) + 1$, when $n > 1, c > 1$.

PROOF. We prove the theorem considering both cases—when a user is added (or removed) from the database. We first derive a proof for the adding case by using the additivity property of entropy from Equation 4. Similarly, the proof for the removing case can be derived from Equation 5. The detailed proofs can be found in our technical report [22]. \square

Baseline Algorithm: In this section we present a baseline algorithm that publishes location entropy for all locations (see Algorithm 1). Since adding (or removing) a single user from the dataset would impact the entropy of all locations he visited, the change of adding (or removing) a user to all locations is bounded by $M_{max} \Delta H$, where M_{max} is the maximum number of locations visited by a user. Thus, Line 6 adds randomly distributed Laplace noise with mean zero and scale $\lambda = \frac{M_{max} \Delta H}{\epsilon}$ to the actual value of location entropy $H(l)$. It has been proved [7] that this is sufficient to achieve differential privacy with such simple mechanism.

Algorithm 1 BASELINE ALGORITHM

- 1: Input: privacy budget ϵ , a set of locations $L = \{l_1, l_2, \dots, l_{|L|}\}$; maximum number of visits of a user to a location C_{max} , maximum number of locations a user visits M_{max} .
 - 2: Compute sensitivity ΔH from Theorem 1 for $C = C_{max}$.
 - 3: For each location l in L
 - 4: Count #visits each user made to l : $c_{l,u}$ and compute $p_{l,u}$
 - 5: Compute $H(l) = -\sum_{u \in U_l} p_{l,u} \log p_{l,u}$
 - 6: Publish noisy LE: $\hat{H}(l) = H(l) + \text{Lap}(\frac{M_{max} \Delta H}{\epsilon})$
-

4.2 Reducing the Global Sensitivity of LE

4.2.1 Limit Algorithm

Limitation of the Baseline Algorithm: Algorithm 1 provides privacy; however, the added noise is excessively high, rendering the results useless. To illustrate, Figure 1 shows the bounds of the global sensitivity (Theorem 1) when C varies. The figure shows that the bound monotonically increases when C grows. Therefore, the noise introduced by Algorithm 1 increases as C and M increase. In practice, C and M can be large because a user may have visited either many locations or a single location many times, resulting in large sensitivity. Furthermore, Figure 2 depicts different values of noise magnitude (in log scale) used in our various algorithms by varying the number of users visiting a location, n . The graph shows that the noise magnitude of the baseline is too high to be useful (see Table 2).

Improving Baseline by Limiting User Activity: To reduce the global sensitivity of LE, and inspired by [13], we propose a thresholding technique, named LIMIT, to limit an individual’s activity by truncating C and M . Our technique is based on the following two observations. First, Figure 3b

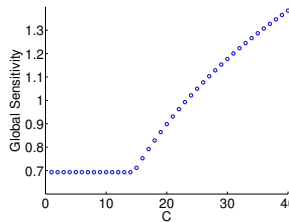


Figure 1: Global sensitivity bound of location entropy when varying C .

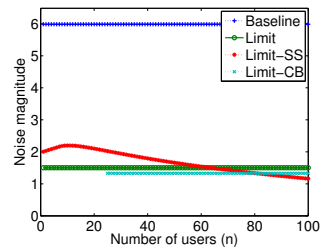
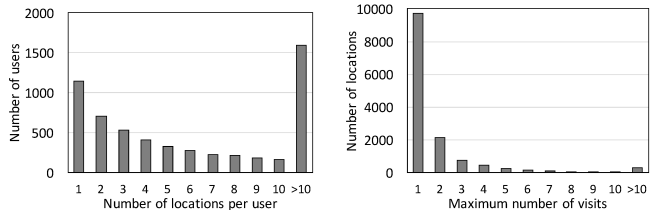


Figure 2: Noise magnitude in natural log scale ($\epsilon = 5$, $C_{max}=1000$, $M_{max}=100$, $C=20$, $M=5$, $\delta=10^{-8}$, $k=25$).

shows the maximum number of visits a user contributes to a location in the Gowalla dataset that will be used in Section 6 for evaluation. Although most users have one and only one visit, the sensitivity of LE is determined by the worst-case scenario—the maximum number of visits³. Second, Figure 3a shows the number of locations visited by a user. The figure confirms that there are many users who contribute to more than ten locations.



(a) A user may visit many locations

(b) The largest number of visits a user contributes to a location

Figure 3: Gowalla, New York.

Since the introduced noise linearly increases with M and monotonically increases with C , the noise can be reduced by capping them. First, to truncate M , we keep the *first* M location visits of the users who visit more than M locations and throw away the rest of the locations’ visits. As a result, adding or removing a single user in the dataset affects at most M locations. Second, we set the number of visits of the users who have contributed more than C visits to a particular location of C . Figure 2 shows that the noise magnitude used in LIMIT drops by two orders of magnitude when compared with the baseline’s sensitivity.

At a high-level, LIMIT (Algorithm 2) works as follows. Line 3 limits user activity across locations, while Line 7 limits user activity to a location. The impact of Line 3 is the introduction of approximation error on the published data. This is because the number of users visiting some locations may be reduced, which alters their actual LE values. Subsequently, some locations may be thrown away without being published. Furthermore, Line 7 also alters the value of location entropy, but by trimming the number of visits of a user to a location. The actual LE value of location l (after thresholding M and C) is computed in Line 8. Consequently, the noisy LE is published in Line 9, where $\text{Lap}(\frac{M \Delta H}{\epsilon})$ de-

³This suggests that users tend not to check-in at places that they visit the most, e.g., their homes, because if they did, the peak of the graph would not be at 1.

notes a random variable drawn independently from Laplace distribution with mean zero and scale parameter $\frac{M\Delta H}{\epsilon}$.

Algorithm 2 LIMIT ALGORITHM

- 1: Input: privacy budget ϵ , a set of locations $L = \{l_1, l_2, \dots, l_{|L|}\}$, maximum threshold on the number of visits of a user to a location C , maximum threshold on the number of locations a user visits M
 - 2: For each user u in U
 - 3: Truncate M : keep the first M locations' visits of the users who visit more than M locations
 - 4: Compute sensitivity ΔH from Theorem 1.
 - 5: For each location l in L
 - 6: Count #visits each user made to l : $c_{l,u}$ and compute $p_{l,u}$
 - 7: Threshold C : $\bar{c}_{l,u} = \min(C, c_{l,u})$, then compute $\bar{p}_{l,u}$
 - 8: Compute $\hat{H}(l) = -\sum_{u \in U_l} \bar{p}_{l,u} \log \bar{p}_{l,u}$
 - 9: Publish noisy LE: $\hat{H}(l) = \bar{H}(l) + \text{Lap}(\frac{M\Delta H}{\epsilon})$
-

The performance of Algorithm 2 depends on how we set C and M . There is a trade-off on the choice of values for C and M . Small values of C and M introduce small perturbation error but large approximation error and vice versa. Hence, in Section 6, we empirically find the values of M and C that strike a balance between noise and approximation error.

4.2.2 Privacy Guarantee of the Limit Algorithm

The following theorem shows that Algorithm 2 is differentially private.

THEOREM 3. *Algorithm 2 satisfies ϵ -differential privacy.*

PROOF. For all locations, let L_1 be any subset of L . Let $T = \{t_1, t_2, \dots, t_{|L_1|}\} \in \text{Range}(\mathcal{A})$ denote an arbitrary possible output. Then we need to prove the following:

$$\frac{\Pr[\mathcal{A}(O_{1(\text{org})}, \dots, O_{|L_1|(\text{org})}) = T]}{\Pr[\mathcal{A}(O_{1(\text{org})} \setminus O_{l,u(\text{org})}, \dots, O_{|L_1|(\text{org})} \setminus O_{l,u(\text{org})}) = T]} \leq \exp(\epsilon)$$

The details of the proof and notations used can be found in our technical report [22]. \square

5. RELAXATION OF PRIVATE LE

This section presents our utility enhancements by adopting two weaker notions of privacy: smooth sensitivity [16] (slightly weaker) and crowd-blending [9] (strictly weaker).

5.1 Relaxation with Smooth Sensitivity

We aim to extend LIMIT to publish location entropy with smooth sensitivity (or SS for short). We first present the notions of smooth sensitivity and the LIMIT-SS algorithm. We then show how to precompute the SS of location entropy.

5.1.1 LIMIT-SS Algorithm

Smooth sensitivity is a technique that allows one to compute noise magnitude—not only by the function one wants to release (i.e., location entropy), but also by the database itself. The idea is to use the local sensitivity bound of each location rather than the global sensitivity bound, resulting in small injected noise. However, simply adopting the local sensitivity to calibrate noise may leak the information about the number of users visiting that location. Smooth sensitivity is stated as follows.

Let $x, y \in D^N$ denote two databases, where N is the number of users. Let l^x, l^y denote the location l in database x

and y , respectively. Let $d(l^x, l^y)$ be the Hamming distance between l^x and l^y , which is the number of users at location l on which x and y differ; i.e., $d(l^x, l^y) = |\{i : l_i^x \neq l_i^y\}|$; l_i^x represents information contributed by one individual. The local sensitivity of location l^x , denoted by $LS(l^x)$, is the maximum change of location entropy when a user is added or removed.

DEFINITION 3. *Smooth sensitivity [16] For $\beta > 0$, β -smooth sensitivity of location entropy is:*

$$SS_\beta(l^x) = \max_{l^y \in D^N} \left(LS(l^y) \cdot e^{-\beta d(l^x, l^y)} \right) \\ = \max_{k=0,1,\dots,N} e^{-k\beta} \left(\max_{y:d(l^x, l^y)=k} LS(l^y) \right)$$

Smooth sensitivity of LE of location l^x can be interpreted as the maximum of $LS(l^x)$ and $LS(l^y)$ where the effect of y at distance k from x is dropped by a factor of $e^{-k\beta}$. Thereafter, the smooth sensitivity of LE can be plugged into Line 3 of Algorithm 2, producing the LIMIT-SS algorithm.

Algorithm 3 LIMIT-SS ALGORITHM

- 1: Input: privacy budget ϵ , privacy parameter δ , $L = \{l_1, l_2, \dots, l_{|L|}\}$, C, M
 - 2: Copy Lines 2-8 from Algorithm 2
 - 3: Publish noisy LE $\hat{H}(l) = \bar{H}(l) + \frac{M \cdot 2 \cdot SS_\beta(l)}{\epsilon} \cdot \eta$, where $\eta \sim \text{Lap}(1)$, where $\beta = \frac{\epsilon}{2 \ln(\frac{2}{\delta})}$
-

5.1.2 Privacy Guarantee of LIMIT-SS

The noise of LIMIT-SS is specific to a particular location as opposed to those of the BASELINE and LIMIT algorithms. LIMIT-SS has a slightly weaker privacy guarantee. It satisfies (ϵ, δ) -differential privacy, where δ is a privacy parameter, $\delta = 0$ in the case of Definition 1. The choice of δ is generally left to the data releaser. Typically, $\delta < \frac{1}{\text{number of users}}$ (see [16] for details).

THEOREM 4. *Calibrating noise to smooth sensitivity [16] If $\beta \leq \frac{\epsilon}{2 \ln(\frac{2}{\delta})}$ and $\delta \in (0, 1)$, the algorithm $l \mapsto H(l) + \frac{2 \cdot SS_\beta(l)}{\epsilon} \cdot \eta$, where $\eta \sim \text{Lap}(1)$, is (ϵ, δ) -differentially private.*

THEOREM 5. *LIMIT-SS is (ϵ, δ) -differentially private.*

PROOF. Using Theorem 4, \mathcal{A}_l satisfies (0) -differential privacy when $l \notin L_1 \cap L(u)$, and satisfies $(\frac{\epsilon}{M}, \frac{\delta}{M})$ -differential privacy when $l \in L_1 \cap L(u)$. \square

5.1.3 Precomputation of Smooth Sensitivity

This section shows that the smooth sensitivity of a location visited by n users can be effectively precomputed. Figure 2 illustrates the precomputed local sensitivity for a fixed value of C .

Let $LS(C, n), SS(C, n)$ be the local sensitivity and the smooth sensitivity of all locations that visited by n users, respectively. $LS(C, n)$ is defined in Theorem 2. Let $GS(C)$ be the global sensitivity of the location entropy given C , which is defined in Theorem 1. Algorithm 4 computes $SS(C, n)$. At a high level, the algorithm computes the effect of all locations at every possible distance k from n , which is non-trivial. Thus, to speed up computations, we propose two stopping conditions based on the following observations.

Let n^x, n^y be the number of users visited l^x, l^y , respectively. If $n^x > n^y$, Algorithm 4 stops when $e^{-k\beta}GS(C)$ is less than the current value of smooth sensitivity (Line 7). If $n^x < n^y$, given the fact that $LS(l^y)$ starts to decrease when $n^y > \frac{C}{\log C - 1} + 1$, and $e^{-k\beta}$ also decreases when k increases, Algorithm 4 also terminates when $n^y > \frac{C}{\log C - 1} + 1$ (Line 8). In addition, the algorithm tolerates a small value of smooth sensitivity ξ . Thus, when n is greater than n_0 such that $LS(C, n_0) < \xi$, the precomputation of $SS(C, n)$ is stopped and $SS(C, n)$ is considered as ξ for all $n > n_0$ (Line 8).

Algorithm 4 PRECOMPUTE SMOOTH SENSITIVITY

- 1: Input: privacy parameters: ϵ, δ, ξ ; C , maximum number of possible users N
 - 2: Set $\beta = \frac{\epsilon}{2 \ln(\frac{2}{\delta})}$
 - 3: For $n = [1, \dots, N]$
 - 4: $SS(C, n) = 0$
 - 5: For $k = [1, \dots, N]$:
 - 6: $SS(C, n) = \max(SS(C, n), e^{-k\beta} \max(LS(C, n - k), LS(C, n + k)))$
 - 7: Stop when $e^{-k\beta}GS(C, n - k) < SS(C, n)$ and $n + k > \frac{C}{\log C - 1} + 1$
 - 8: Stop when $n > \frac{C}{\log C - 1} + 1$ and $LS(C, n) < \xi$
-

5.2 Relaxation with Crowd-Blending Privacy

5.2.1 LIMIT-CB Algorithm

Thus far, we publish entropy for all locations; however, the ratio of noise to the true value of LE (noise-to-true-entropy ratio) is often excessively high when the number of users visiting a location n is small (i.e., Equation 3 shows that entropy of a location is bounded by $\log(n)$). The large noise-to-true-entropy ratio would render the published results useless since the introduced noise outweighs the actual value of LE. This is an inherent issue with the sparsity of the real-world datasets. For example, Figure 4 summarizes the number of users contributing visits to each location in the Gowalla dataset. The figure shows that most locations have check-ins from fewer than ten users. These locations have LE values of less than $\log(10)$, which are particularly prone to the noise-adding mechanism in differential privacy.

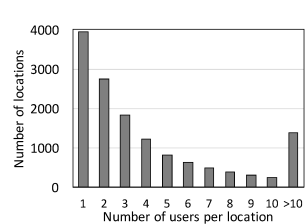


Figure 4: Sparsity of location visits (Gowalla, New York).

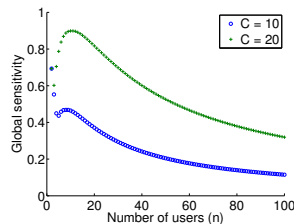


Figure 5: Global sensitivity bound when varying n .

Therefore, to reduce the noise-to-true-entropy ratio, we propose a small sensitivity bound of location entropy that depends on the minimum number of users visiting a location, denoted by k . Subsequently, we present Algorithm 5 that satisfies (k, ϵ) -crowd-blending privacy [9]. We prove this in Section 5.2.2.

The algorithm aims to publish entropy of locations with at least k users ($n \geq k$) and throw away the other locations. We refer to the algorithm as LIMIT-CB. Lines 3-6 publish the entropy of each location according to (k, ϵ) -crowd-blending privacy. That is, we publish the entropy of the locations with at least k users and suppress the others. The following lemma shows that for the locations with at least k users we have a tighter bound on ΔH , which depends on C and k . Figure 2 shows that the sensitivity used in LIMIT-CB is significantly smaller than LIMIT’s sensitivity.

THEOREM 6. *Global sensitivity of location entropy for locations with at least k users, $k \geq \frac{C}{\log C - 1} + 1$, where C is the maximum number of visits a user contributes to a location, is the local sensitivity at $n = k$.*

PROOF. We prove the theorem by showing that local sensitivity decreases when the number of users $n \geq \frac{C}{\log C - 1} + 1$. Thus, when $n \geq \frac{C}{\log C - 1} + 1$, the global sensitivity equals to the local sensitivity at the smallest value of n , i.e., $n = k$. The detailed proof can be found in our technical report [22]. \square

Algorithm 5 LIMIT-CB ALGORITHM

- 1: Input: all users U , privacy budget ϵ ; C, M, k
 - 2: Compute global sensitivity ΔH based on Theorem 6.
 - 3: For each location $l \in L$
 - 4: Count number of users who visit l , n_l
 - 5: If $n_l \geq k$, publish $\hat{H}(l)$ according to Algorithm 2 with budget ϵ using a tighter bound on ΔH
 - 6: Otherwise, do not publish the data
-

5.2.2 Privacy Guarantee of LIMIT-CB

Before proving the privacy guarantee of LIMIT-CB, we first present the notion of crowd-blending privacy, a strict relaxation of differential privacy [9]. k -crowd blending private sanitization of a database requires each individual in the database to blend with k other individuals in the database. This concept is related to k -anonymity [19] since both are based on the notion of “blending in a crowd.” However, unlike k -anonymity that only restricts the published data, crowd-blending privacy imposes restrictions on the noise-adding mechanism. Crowd-blending privacy is defined as follows.

DEFINITION 4 (CROWD-BLENDING PRIVACY). *An algorithm A is (k, ϵ) -crowd-blending private if for every database D and every individual $t \in D$, either t ϵ -blends in a crowd of k people in D , or $A(D) \approx_\epsilon A(D \setminus \{t\})$ (or both).*

A result from [9] shows that differential privacy implies crowd-blending privacy.

THEOREM 7. DP \rightarrow CROWD-BLENDING PRIVACY [7] *Let A be any ϵ -differentially private algorithm. Then, A is (k, ϵ) -crowd-blending private for every integer $k \geq 1$.*

The following theorem shows that Algorithm 5 is (k, ϵ) -crowd-blending private.

THEOREM 8. *Algorithm 5 is (k, ϵ) -crowd-blending private.*

PROOF. First, if there are at least k people in a location, then individual u ϵ -blends with k people in U . This is because Line 5 of the algorithm satisfies ϵ -differential privacy, which infers (k, ϵ) -crowd-blending private (Theorem 7). Otherwise, we have $A(D) \approx_0 A(D \setminus \{t\})$ since A suppresses each location with less than k users. \square

	Sparse	Dense	Gow.
# of locations	10,000	10,000	14,058
# of users	100K	10M	5,800
Max LE	9.93	14.53	6.45
Min LE	1.19	6.70	0.04
Avg. LE	3.19	7.79	1.45
Variance of LE	1.01	0.98	0.6
Max #locations per user	100	100	1407
Avg. #locations per user	19.28	19.28	13.5
Max #visits to a loc. per user	20,813	24,035	162
Avg. #visits to a loc. per user	2578.0	2575.8	7.2
Avg. #users per loc.	192.9	19,278	5.6

Table 2: Statistics of the datasets.

6. PERFORMANCE EVALUATION

We conduct several experiments on real-world and synthetic datasets to compare the effectiveness and utility of our proposed algorithms. Below, we first discuss our experimental setup. Next, we present our experimental results.

6.1 Experimental Setup

Datasets: We conduct experiments on one real-world (Gowalla) and two synthetic datasets (Sparse and Dense). The statistics of the datasets are shown in Table 2. Gowalla contains the check-in history of users in a location-based social network. For our experiments, we use the check-in data in an area covering the city of New York.

For synthetic data generation, in order to study the impact of the density of the dataset, we consider two cases: Sparse and Dense. Sparse contains 100,000 users while Dense has 10 million users. The Gowalla dataset is sparse as well. We add the Dense synthetic dataset to emulate the case for large industries, such as Google, who have access to large- and fine-granule user location data. To generate visits, without loss of generality, the location with id $x \in [1, 2, \dots, 10,000]$ has a probability $1/x$ of being visited by a user. This means that locations with smaller ids tend to have higher location entropy since more users would visit these locations. In the same fashion, the user with id $y \in \{1, 2, \dots, 100,000\}$ (Sparse) is selected with probability $1/y$. This follows the real-world characteristic of location data where a small number of locations are very popular and then many locations have a small number of visits.

In all of our experiments, we use five values of privacy budget $\epsilon \in \{0.1, 0.5, 1, 5, 10\}$. We vary the maximum number of visits a user contributes to a location $C \in \{1, 2, \dots, 5, \dots, 50\}$ and the maximum number of locations a user visits $M \in \{1, 2, 5, 10, 20, 30\}$. We vary threshold $k \in \{10, 20, 30, 40, 50\}$. We also set $\xi = 10^{-3}$, $\delta = 10^{-8}$, and $\beta \approx \epsilon/2 * \ln(2/\delta)$. Default values are shown in boldface.

Metrics: We use KL-divergence as one measure of preserving the original data distribution after adding noise. Given two discrete probability distributions P and Q , the KL-divergence of Q from P is defined as follows:

$$D_{KL}(P||Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)} \quad (6)$$

In this paper the location entropy of location l is the probability that l is chosen when a location is randomly selected from the set of all locations; P and Q are respectively the

published and the actual LE of locations after normalization; i.e., normalized values must sum to unity.

We also use mean squared error (MSE) over a set of locations L as the metric of accuracy using Equation 7.

$$MSE = \frac{1}{|L|} \sum_{l \in L} (LE_a(l) - LE_n(l))^2 \quad (7)$$

where $LE_a(l)$ and $LE_n(l)$ are the actual and noisy entropy of the location l , respectively.

Since LIMIT-CB discards more locations as compared to LIMIT and LIMIT-SS, we consider both cases: 1) KL-divergence and MSE metrics are computed on all locations L , where the entropy of the suppressed locations are set to zero (default case); 2) the metrics are computed on the subset of locations that LIMIT-CB publishes, termed *Throwaway*.

6.2 Experimental Results

We first evaluate our algorithms on the synthetic datasets.

6.2.1 Overall Evaluation of the Proposed Algorithms

We evaluate the performance of LIMIT from Section 4.2.1 and its variants (LIMIT-SS and LIMIT-CB). We do not include the results for BASELINE since the excessively high amount of injected noise renders the perturbed data useless.

Figure 6 illustrates the distributions of noisy vs. actual LE on Dense and Sparse. The actual distributions of the dense (Figure 6a) and sparse (Figure 6e) datasets confirm our method of generating the synthetic datasets; locations with smaller ids have higher entropy, and entropy of locations in Dense are higher than that in Sparse. We observe that LIMIT-SS generally performs best in preserving the original data distribution for Dense (Figure 6c), while LIMIT-CB performs best for Sparse (Figure 6h). Note that as we show later, LIMIT-CB performs better than LIMIT-SS and LIMIT given a small budget ϵ (see Section 6.2.2).

Due to the truncation technique, some locations may be discarded. Thus, we report the percentage of perturbed locations, named *published ratio*. The published ratio is computed as the number of perturbed locations divided by the total number of *eligible* locations. A location is eligible for publication if it contains check-ins from at least K users ($K \geq 1$). Figure 7 shows the effect of k on the published ratio of LIMIT-CB. Note that the published ratio of LIMIT and LIMIT-SS is the same as LIMIT-CB when $k = K$. The figure shows that the ratio is 100% with Dense, while that of Sparse is less than 10%. The reason is that with Dense, each location is visited by a large number of users on average (see Table 2); thus, limiting M and C would reduce the average number of users visiting a location but not to the point where the locations are suppressed. This result suggests that our truncation technique performs well on large datasets.

6.2.2 Privacy-Utility Trade-off (Varying ϵ)

We compare the trade-off between privacy and utility by varying the privacy budget ϵ . The utility is captured by the KL-divergence metric introduced in Section 6.1. We also use the MSE metric. Figure 8 illustrates the results. As expected, when ϵ increases, less noise is injected, and values of KL-divergence and MSE decrease. Interestingly though, KL-divergence and MSE saturate at $\epsilon = 5$, where reducing privacy level (increase ϵ) only marginally increases utility. This can be explained through a significant approximation

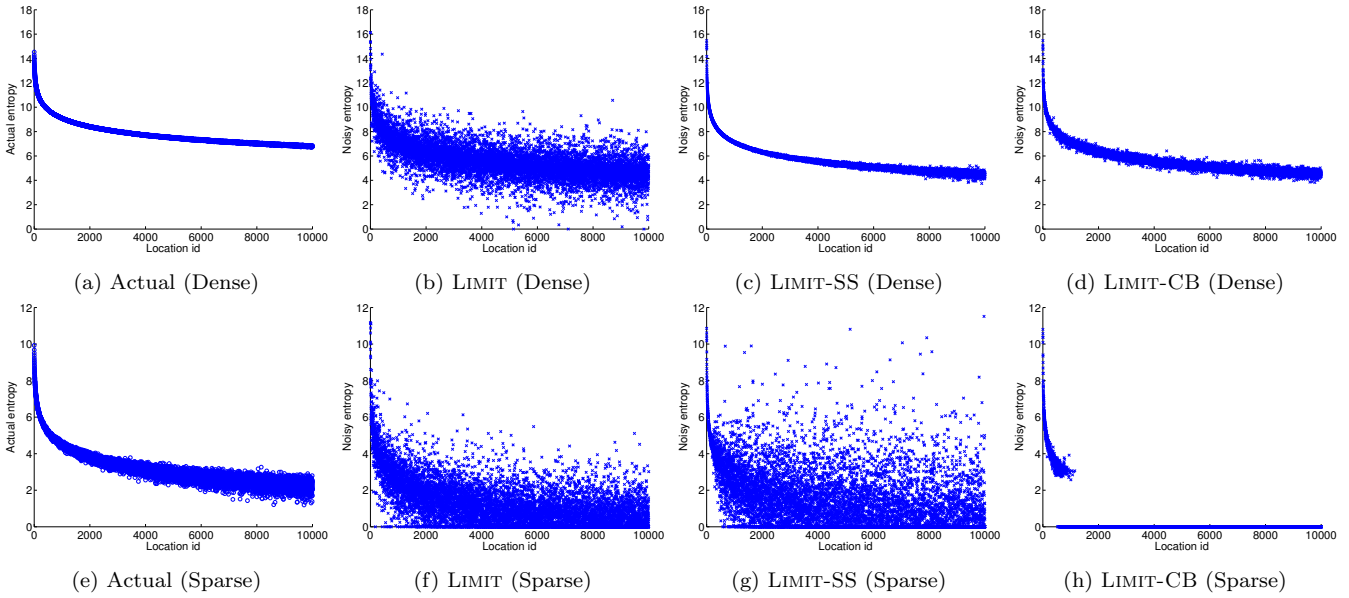


Figure 6: Comparison of the distributions of noisy vs. actual location entropy on the dense and sparse datasets.

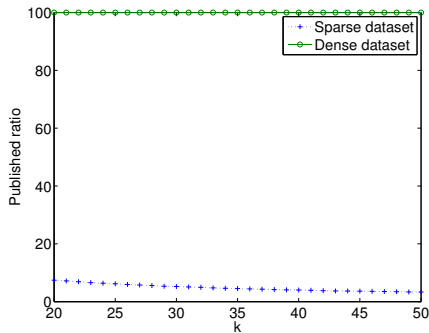


Figure 7: Published ratio of LIMIT-CB when varying k ($K = 20$).

error in our thresholding technique that outweighs the impact of having smaller perturbation error. Note that the approximation errors are constant in this set of experiments since the parameters C , M and k are fixed.

Another observation is that the observed errors incurred are generally higher for Dense (Figures 8b vs. 8c), which is surprising, as differentially private algorithms often perform better on dense datasets. The reason for this is because limiting M and C has a larger impact on Dense, resulting in a large perturbation error. Furthermore, we observe that the improvements of LIMIT-SS and LIMIT-CB over LIMIT are more significant with small ϵ . In other words, LIMIT-SS and LIMIT-CB would have more impact with a higher level of privacy protection. Note that these enhancements come at the cost of weaker privacy protection.

6.2.3 The Effect of Varying M and C

We first evaluate the performance of our proposed techniques by varying threshold M . For brevity, we present the results only for MSE, as similar results have been observed for KL-divergence. Figure 9 indicates the trade-off between the approximation error and the perturbation error. Our thresholding technique decreases M to reduce the perturba-

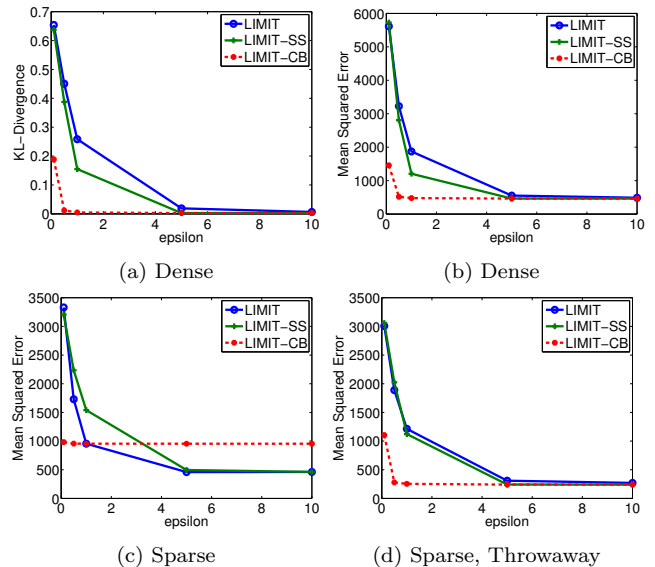


Figure 8: Varying ϵ

tion error, but at the cost of increasing the approximation error. As a result, at a particular value of M , the technique balances the two types of errors and thus minimizes the total error. For example, in Figure 9a, LIMIT performs best at $M = 5$, while LIMIT-SS and LIMIT-CB work best at $M \geq 30$. In Figure 9b, however, LIMIT-SS performs best at $M = 10$ and LIMIT-CB performs best at $M = 20$.

We then evaluate the performance of our techniques by varying threshold C . Figure 10 shows the results. For brevity, we only include KL-divergence results (MSE metric shows similar trends). The graphs show that KL-divergence increases as C grows. This observation suggests that C should be set to a small number (less than 10). By comparing the effect of varying M and C , we conclude that M

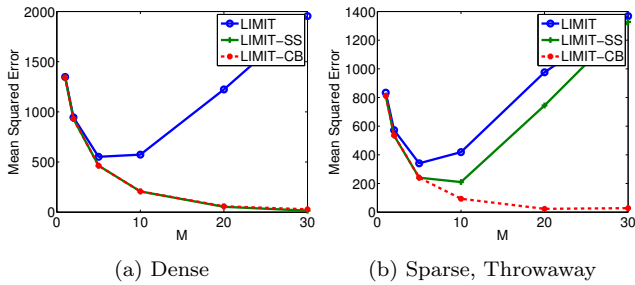


Figure 9: Varying M

has more impact on the trade-off between the approximation error and the perturbation error.

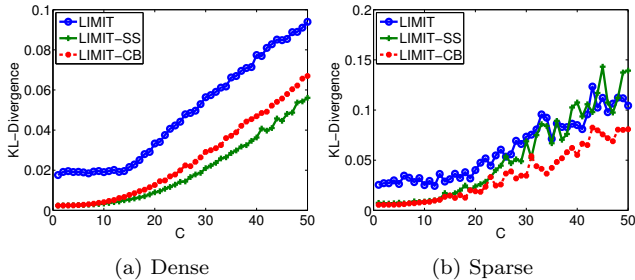


Figure 10: Varying C

6.2.4 Results on the Gowalla Dataset

In this section we evaluate the performance of our algorithms on the Gowalla dataset. Figure 11 shows the distributions of noisy vs. actual location entropy. Note that we sort the locations based on their actual values of LE as depicted in Figure 11a. As expected, due to the sparseness of Gowalla (see Table 2), the published values of LE in LIMIT and LIMIT-SS are scattered while those in LIMIT-CB preserve the trend in the actual data but at the cost of throwing away more locations (Figure 11d). Furthermore, we conduct experiments on varying various parameters (i.e., ϵ , C , M , k) and observe trends similar to the Sparse dataset; nevertheless, for brevity, we only show the impact of varying ϵ and M in Figure 12.

Recommendations for Data Releases: We summarize our observations and provide guidelines for choosing appropriate techniques and parameters. LIMIT-CB generally performs best on sparse datasets because it only focuses on publishing the locations with large visits. Alternatively, if the dataset is dense, LIMIT-SS is recommended over LIMIT-CB since there are sufficient locations with large visits. A dataset is dense if most locations (e.g., 90%) have at least n_{CB} users, where n_{CB} is the threshold for choosing LIMIT-CB. Particularly, given fixed parameters C, ϵ, δ, k — n_{CB} can be found by comparing the global sensitivity of LIMIT-CB and the precomputed smooth sensitivity. In Figure 2, n_{CB} is a particular value of n where $SS(C, n_{CB})$ is smaller than the global sensitivity of LIMIT-CB. In other words, the noise magnitude required for LIMIT-SS is smaller than that for LIMIT-CB. Regarding the choice of parameters, to guarantee strong privacy protection, ϵ should be as small as possible, while the measured utility metrics are practical. Finally, the

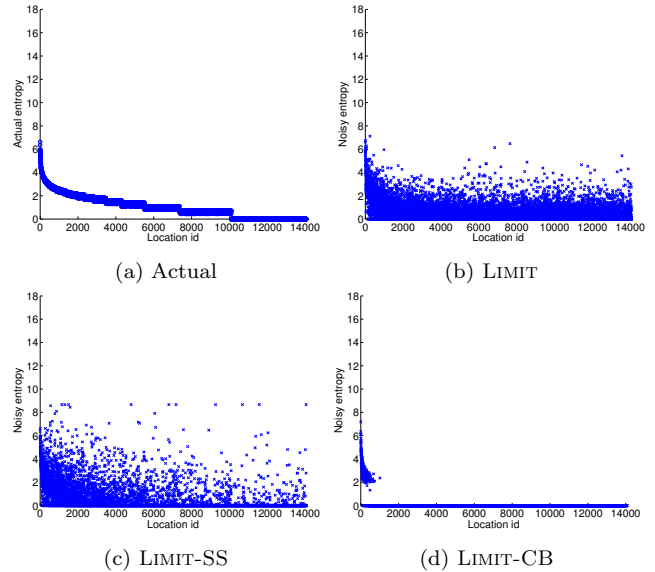


Figure 11: Comparison of the distributions of noisy vs. actual location entropy on Gowalla, $M = 5$.

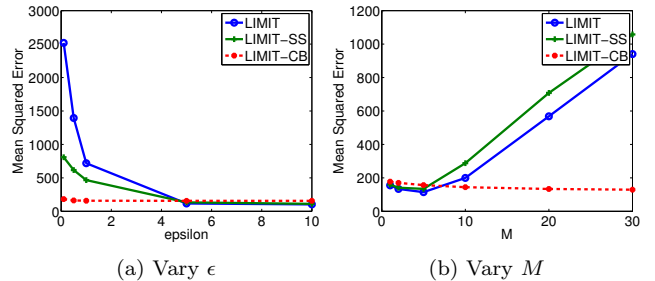


Figure 12: Varying ϵ and M (Gowalla).

value of C should be small (≤ 10), while the value of M can be tuned to achieve maximum utility.

7. RELATED WORK

Location privacy has largely been studied in the context of location-based services, participatory sensing and spatial crowdsourcing. Most studies use the model of spatial k -anonymity [19], where the location of a user is hidden among k other users [11, 15]. However, there are known attacks on k -anonymity, e.g., when all k users are at the same location. Nevertheless, such techniques assume a centralized architecture with a trusted third party, which is a single point of attack. Consequently, a technique that makes use of cryptographic techniques such as private information retrieval is proposed that does not rely on a trusted third party to anonymize locations [10]. Recent studies on location privacy have focused on leveraging differential privacy (DP) to protect the privacy of users [21, 27].

Location entropy has been extensively used in various areas of research, including multi-agent systems [25], wireless sensor networks [26], geosocial networks [4, 3, 17], personalized web search [14], image retrieval [29] and spatial crowdsourcing [12, 23, 20], etc. The study that most closely relates to ours focuses on privacy-preserving location-based services in which location entropy is used as the measure of

privacy or the attacker’s uncertainty [28, 24]. In [28], a privacy model is proposed that discloses a location on behalf of a user only if the location has at least the same popularity (quantified by location entropy) as a public region specified by a user. In fact, locations with high entropy are more likely to be shared (checked-in) than places with low entropy [24]. However, directly using location entropy compromises the privacy of individuals. For example, an adversary certainly knows whether people visiting a location based on its entropy value, e.g., low value means a small number of people visit the location, and if they are all in a small geographical area, their privacy is compromised. To the best of our knowledge, there is no study that uses differential privacy for publishing location entropy, despite its various applications that can be highly instrumental in protecting the privacy of individuals.

8. CONCLUSIONS

We introduced the problem of publishing the entropy of a set of locations according to differential privacy. A baseline algorithm was proposed based on the derived tight bound for global sensitivity of the location entropy. We showed that the baseline solution requires an excessively high amount of noise to satisfy ϵ -differential privacy, which renders the published results useless. A simple yet effective truncation technique was then proposed to reduce the sensitivity bound by two orders of magnitude, and this enabled publication of location entropy with reasonable utility. The utility was further enhanced by adopting smooth sensitivity and crowd-blending. We conducted extensive experiments and concluded that the proposed techniques are practical.

Acknowledgement

We would like to thank Prof. Aleksandra Korolova for her constructive feedback during the course of this research.

This research has been funded in part by NSF grants IIS-1320149 and CNS-1461963, National Cancer Institute, National Institutes of Health, Department of Health and Human Services, under Contract No. HHSN261201500003B, the USC Integrated Media Systems Center, and unrestricted cash gifts from Google, Northrop Grumman and Oracle. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of any of the sponsors.

9. REFERENCES

- [1] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang. Deep learning with differential privacy. *arXiv:1607.00133*, 2016.
- [2] A. Blum, C. Dwork, F. McSherry, and K. Nissim. Practical privacy: The SuLQ framework. In *PODS*, pages 128–138. ACM, 2005.
- [3] E. Cho, S. A. Myers, and J. Leskovec. Friendship and mobility: user movement in location-based social networks. In *SIGKDD*, pages 1082–1090. ACM, 2011.
- [4] J. Cranshaw, E. Toch, J. Hong, A. Kittur, and N. Sadeh. Bridging the gap between physical location and online social networks. In *UbiComp*. ACM, 2010.
- [5] Y.-A. de Montjoye, C. A. Hidalgo, M. Verleysen, and V. D. Blondel. Unique in the crowd: The privacy bounds of human mobility. *Scientific Reports*, 2013.
- [6] C. Dwork. Differential privacy. In *Automata, languages and programming*, pages 1–12. Springer, 2006.
- [7] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *TCC*, pages 265–284. Springer, 2006.
- [8] Ú. Erlingsson, V. Pihur, and A. Korolova. RAPPOR: Randomized aggregatable privacy-preserving ordinal response. In *SIGSAC*, pages 1054–1067. ACM, 2014.
- [9] J. Gehrke, M. Hay, E. Lui, and R. Pass. Crowd-blending privacy. In *Advances in Cryptology*, pages 479–496. Springer, 2012.
- [10] G. Ghinita, P. Kalnis, A. Khoshgozaran, C. Shahabi, and K.-L. Tan. Private queries in location based services: anonymizers are not necessary. In *SIGMOD*, pages 121–132. ACM, 2008.
- [11] M. Gruteser and D. Grunwald. Anonymous usage of location-based services through spatial and temporal cloaking. In *MobiSys*, pages 31–42. ACM, 2003.
- [12] L. Kazemi and C. Shahabi. GeoCrowd: enabling query answering with spatial crowdsourcing. In *SIGSPATIAL 2012*, pages 189–198. ACM, 2012.
- [13] A. Korolova, K. Kenthapadi, N. Mishra, and A. Ntoulas. Releasing search queries and clicks privately. In *WWW*, pages 171–180. ACM, 2009.
- [14] K. W.-T. Leung, D. L. Lee, and W.-C. Lee. Personalized web search with location preferences. In *ICDE*, pages 701–712. IEEE, 2010.
- [15] M. F. Mokbel, C.-Y. Chow, and W. G. Aref. The new Casper: query processing for location services without compromising privacy. In *VLDB*, pages 763–774, 2006.
- [16] K. Nissim, S. Raskhodnikova, and A. Smith. Smooth sensitivity and sampling in private data analysis. In *STOC*, pages 75–84. ACM, 2007.
- [17] H. Pham, C. Shahabi, and Y. Liu. Inferring social strength from spatiotemporal data. *ACM Trans. Database Syst.*, 41(1):7:1–7:47, Mar. 2016.
- [18] C. E. Shannon and W. Weaver. A mathematical theory of communication, 1948.
- [19] L. Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05), 2002.
- [20] H. To, L. Fan, L. Tran, and C. Shahabi. Real-time task assignment in hyperlocal spatial crowdsourcing under budget constraints. In *PerCom*. IEEE, 2016.
- [21] H. To, G. Ghinita, and C. Shahabi. A framework for protecting worker location privacy in spatial crowdsourcing. *VLDB*, 7(10):919–930, 2014.
- [22] H. To, K. Nguyen, and C. Shahabi. Differentially private publication of location entropy. *University of Southern California, Report ID 16-968*, 2016. <https://www.cs.usc.edu/research/technical-reports-list>.
- [23] H. To, C. Shahabi, and L. Kazemi. A server-assigned spatial crowdsourcing framework. *TSAS*, 1(1):2, 2015.
- [24] E. Toch, J. Cranshaw, P. H. Drielsma, J. Y. Tsai, P. G. Kelley, J. Springfield, L. Cranor, J. Hong, and N. Sadeh. Empirical models of privacy in location sharing. In *UbiComp*, pages 129–138. ACM, 2010.
- [25] H. Van Dyke Parunak and S. Brueckner. Entropy and self-organization in multi-agent systems. In *AAMAS*, pages 124–130. ACM, 2001.
- [26] H. Wang, K. Yao, G. Pottie, and D. Estrin. Entropy-based sensor selection heuristic for target localization. In *IPSN*, pages 36–45. ACM, 2004.
- [27] Y. Xiao and L. Xiong. Protecting locations with differential privacy under temporal correlations. In *CCS*, pages 1298–1309. ACM, 2015.
- [28] T. Xu and Y. Cai. Feeling-based location privacy protection for location-based services. In *CCS*, pages 348–357. ACM, 2009.
- [29] K. Yanai, H. Kawakubo, and B. Qiu. A visual analysis of the relationship between word concepts and geographical locations. In *CIVR*, page 13. ACM, 2009.