Beware of What You Share: Inferring User Locations in Venmo

Xin Yao, *Student Member, IEEE,* Yimin Chen, *Student Member, IEEE,* Rui Zhang, *Member, IEEE,* Yanchao Zhang, *Senior Member, IEEE,* and Yaping Lin, *Member, IEEE*

***Abstract*—Mobile payment apps are seeing explosive usage worldwide. This paper focuses on Venmo, a very popular mobile person-to-person (P2P) payment service owned by Paypal. Venmo allows money transfers between users with a mandatory transaction note. More than half of transaction records in Venmo are public information. In this paper, we propose a Multi-Layer Location Inference (MLLI) technique to infer user locations from public transaction records in Venmo. MLLI explores two observations. First, many Venmo transaction notes contain implicit location cues. Second, the types and temporal patterns of user transactions have strong ties to their location closeness. With a large dataset of 2.12M users and 20.23M Venmo transaction records, we show that MLLI can identify the top-1, top-3, and top-5 possible locations for a Venmo user with accuracy up to 50%, 80%, and 90%, respectively. Our results highlight the danger of sharing transaction notes on Venmo or similar mobile payment apps.**

***Index Terms*—Mobile payment, security, privacy, location inference.**

I. Introduction

MOBILE payment apps are seeing explosive usage worldwide. According to [1], the volume of mobile payment transactions could rise from $25B in 2016 to nearly $275B by 2021, amounting to an average annual growth rate of 62%. As another example, AliPay and WeChatPay, two popular payment systems in China, have 100M daily active users in December 2016 and 200M users in January 2016, respectively.

This paper focuses on Venmo, a very popular mobile person-to-person (P2P) payment service owned by Paypal. Venmo had 203 million active users and processed $6.8 billion in payment volume in Q1 2017 [2]. It is essentially a combination of social and transaction networks. On the one hand, Venmo allows users to befriend each other as in Facebook-like online social networks (OSNs). On the other hand, it allows money transfers between users by phone numbers, Venmo usernames, or emails along with a mandatory transaction note in the form of words, emojis, or their combinations. For example, Alice paid Bob $5.1 for pizza or charged Bob $550 for 🏠💸. In Venmo, users can make their transaction records viewable by the public with the default system setting or by selected parties only via privacy control. As reported in [3], almost half of Venmo transaction records are public information.

This paper presents the first systematic study to infer the home locations of Venmo users from public transaction records. We follow the convention [4]–[15] to let **a home location (or location for short) refer to a permanent and static city-level region where most of the user's daily activities occur**. This study may have significant positive and negative impacts. On the positive side, the knowledge about user locations may benefit many applications such as socioeconomic studies, local event recommendation, and business promotion. On the other hand, the disclosure of home locations may subject the users to many attacks such as location-based spam campaigns. This study is also very challenging because explicit location clues are relatively sparse and often unreliable in public Venmo transaction records. For example, our large-scale dataset reveals that only 13.34% of public transaction records contain geotagged information (city or state names in USA), most of which are unrelated to the home locations of Venmo users.

In this paper, we propose a **Multi-Layer Location Inference** (MLLI) technique to infer user locations in Venmo. The design of MLLI is driven by two observations. First, many Venmo transaction notes contain implicit location cues. Second, the types and temporal patterns of user transactions have strong ties to their location closeness. For example, if David and Bob split lunch bills on a daily basis via Venmo, they are in the same city (i.e., have the same location) with overwhelming probability. In contrast, if they split monthly wireless bills via Venmo, there is relatively low confidence that they are in the same location because people far away from each other can still share a wireless plan.

MLLI explores the above observations in four steps. First, we use text mining algorithms to obtain the keywords for each transaction note. Since distinct keywords have different location relevance, we further divide the keywords and the corresponding transaction records into four categories, where the lower-numbered category corresponds to higher location relevance. Second, we construct an undirected weighted transaction graph for each category, in which each edge corresponds to two users with any transaction history in that category, and the edge weight depends on their transaction pattern.

X. Yao and Y. Lin are with the College of Computer Science and Electronic Engineering, Hunan University, Changsha, Hunan, 410012, China. (E-mail: xinyao@csu.edu.cn, yplin@hnu.edu.cn)

Y. Chen and Y. Zhang are with School of Electrical, Computer and Energy Engineering, Ira A. Fulton Schools of Engineering, Arizona State University, Tempe, AZ 85287, USA. (E-mail: {ymchen, yczhang}@asu.edu)

R. Zhang is with the Department of Computer and Information Sciences Department, University of Delaware, Newark, DE 19716, USA. (E-mail: ruizhang@udel.edu)

This work was done while X. Yao was a visiting Ph.D. student at Arizona State University. Y. Zhang and Y. Lin are both the corresponding authors.

For example, more intense and consistent transactions should translate into higher edge weights than occasional ones. Third, we identify a small set of users as seeds whose locations can be directly obtained from their geotagged Venmo transaction notes or via external means.[1] Then we propose an iterative multi-layer belief propagation scheme to propagate the location beliefs to non-seed users in each category. Finally, we perform a weighted combination of the location beliefs for each user in the four categories and assign the most probable home location to each user.

We validate the efficacy of MLLI using a large-scale dataset containing 2.12M users and 20.23M Venmo transaction records, which was collected through a 3-month period. Our results show that MLLI can identify the top-1, top-3, and top-5 possible locations for a Venmo user with accuracy up to 50%, 80%, and 90%, respectively. Our results highlight the danger of sharing transaction notes on Venmo.

There have been such efforts as [4]–[15] to infer the user locations in traditional OSNs such as Facebook and Twitter. Existing techniques can be classified into network-based approaches [4]–[10] or content-based approaches [11]–[15]. The former depend on the assumption that physically close OSN users are more likely to interact with each other so that a user's location can be inferred from those of his/her OSN neighbors. In contrast, content-based approaches always utilize geographic hints (e.g., city landmarks) in user posts to infer hidden locations. Our MLLI technique explores both network information (transaction parties) and content information (transaction notes) in the unique Venmo context.

The rest of this paper is outlined as follows. Section II introduces Venmo and presents the problem formulation. Section III discusses how we crawl the dataset and collect the ground truth. Section IV details how we construct categorized transaction graphs from public transaction records. Section V presents the MLLI technique. Section VI analyzes the convergence and time complexity of MLLI. Section VII evaluates MLLI with the real dataset. Section VIII outlines the related work. Section IX concludes this paper.

II. Problem Formulation

In this section, we provide a brief introduction to Venmo and then the problem formulation.

Venmo is PayPal's mobile P2P payment service that has been gaining extreme popularity. It allows direct money transfer between registered users via a mobile app or web interface. Each transaction must have a short note indicating the transaction purpose. A transaction note can consist of words, emojis, or their combinations. The transaction records of each user are public information by default. A user can make his/her transaction records viewable by selected parties only as well. The recent study [3] reveals that almost half of Venmo transaction records are public information that can be easily obtained by invoking Venmo APIs.

We explore public Venmo transaction records to construct an undirected weighted transaction graph $\mathcal{G} = \langle \mathcal{N}, \mathcal{E} \rangle$, where $\mathcal{N}$ and $\mathcal{E}$ refer to the user set and edge set, respectively. An edge $\mathbf{e}_{i,j} \in \mathcal{E}$ is formed between any users i and j as long as they have historical transactions in Venmo. Who is the payment sender or receiver of any transaction has no bearing on our scheme design, so we ignore the directions of Venmo transactions. A set of transaction records are associated with each edge $\mathbf{e}_{i,j}$. We use text mining algorithms to infer a keyword for each transaction record, referred to as its transaction *purpose*. Then we can group the transactions of the same purpose into a time series. For example, we can have {`Dinner`:{01/14/2017@6:00pm, 01/17/2017@5:30pm}; 🏠💸:{02/27/2017@9:00am}} associated with edge $\mathbf{e}_{i,j}$, meaning that users i and j split dinner expenses twice and rent once at the corresponding time. Let $\mathcal{N}^* \subset \mathcal{N}$ denote the set of users with known home locations (called *seed user*s) and $\mathcal{N}^+ \subset \mathcal{N}$ denote the remaining users with unknown home locations (called *non-seed user*s). So we have $\mathcal{N} = \mathcal{N}^* \cup \mathcal{N}^+$. We aim to tackle the following problem.

$\hat{h}_i$ for each non-seed user $i \in \mathcal{N}^+$ as close to its true location h_i as possible.

Location Inference in Venmo: Given a Venmo transaction graph $\mathcal{G} = \langle \mathcal{N}, \mathcal{E} \rangle$ with an observed location h_j for each seed user $j \in \mathcal{N}^*$, estimate top-κ possible home locations $\hat{\mathbf{h}}_i$ for each non-seed user $i \in \mathcal{N}^+$. Let $\mathcal{N}^-$ be the number of non-seed users whose true location h_i located at $\hat{\mathbf{h}}_i$. We aim to make $|\mathcal{N}^-|$ as close to $|\mathcal{N}^+|$ as possible.

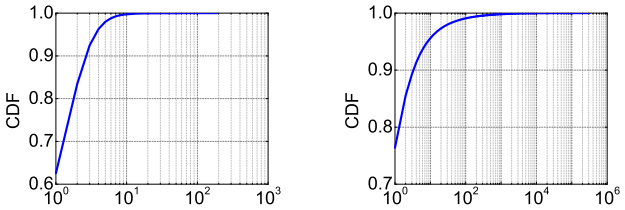
III. Data Crawling

In this section, we introduce our data crawling process and the ground-truth dataset.

A. Data Collection

We used the public Venmo v5 API to retrieve historical transaction records in Venmo. Each retrieval request is a URL with a constant field and a Unix timestamp. As an example, the request is `https://venmo.com/api/v5/public?until=1488369600` when the time is set to 03/01/2017@12:00pm (UTC). Venmo returns the most recent 20 records before the specified time in each retrieval request. By sweeping through the timestamps from December 24, 2016 to March 24, 2017, we obtained 8.2M unique users and 37.46M transactions in total. For our purpose, each transaction record contains the transaction initiator and receiver along with their public profiles, the transaction time, and the mandatory transaction note. Other information returned by Venmo is ignored. Our technique applies to the users with sufficient transactions. In this study, we only consider users with more than 10 transaction records in the crawled dataset, which correspond to 2.12M users with 20.23M transaction records in total. During the data collection, we limited the query rates to avoid disruption to Venmo's services. Although the transaction records in our dataset is publicly accessible via the Venmo v5 API, we will not share the dataset with others unless a written consent from Venmo can be obtained.

[1]Some Venmo users link their Venmo accounts to Facebook accounts and disclose their locations on Facebook.



(a) Purified transaction note length. (b) Keyword frequency.

Fig. 1. The distributions of purified note length and keyword frequency.

B. Ground Truth

It is critical to obtain a ground-truth dataset for our algorithm execution and performance evaluation. About 6% of Facebook users have elected to enter their home addresses in their public profiles [4]. Since a user can register for Venmo with his/her Facebook credentials, we used the following process to crawl the ground truth. We first identified the users whose profiles contain firstname+lastname as well as a user image or a friend list from the crawled transaction records. Then we searched the firstname+lastname for each such Venmo user on Facebook. Any discovered Facebook user is considered the same person as the Venmo user if they two have the same image or a highly similar friend list. If the Facebook user specifies the home location in the profile, the home location of the Venmo user is considered known, corresponding to one piece of ground truth. With this process, we obtained a ground-truth dataset of 1000 users from our original dataset.

IV. Categorized Transaction Graphs

Our MLLI technique explores the observation that public transaction notes in Venmo contain implicit cues for accurate location inference. In this section, we first extract meaningful keywords from the crawled transaction notes. Since different keywords vary in location relevance, we further group the keywords into four categories whereby to construct four categorized transaction graphs for subsequent use. Note that we do not harness explicit location cues in transaction notes, which are not only rare but also unrelated to home locations.

A. Keyword Extraction

We use text mining techniques to extract meaningful keywords from crawled transaction notes. For the words in each transaction note, we first remove the stop words using a stop-word list,[2] in which such words as "the" and "those" are considered too general and meaningless. People may use different inflected words in their transaction notes. So we further conduct stemming [16] to reduce inflected words to their stemforms. For example, "play", "playing", and "played" are all reduced to "play". Each transaction note is translated into its purified version after these two steps. All the words or emojis in a purified transaction note are considered its keywords. Fig. 1(a) shows the length distribution of 20.23M purified transaction notes involving 2.12M Venmo users. As we can see, more than 62% of transaction notes have one keyword, and about 20% have two keywords. This result provides firm evidence that the keywords can well characterize Venmo transaction notes. Table I shows the frequencies of the top-10 common keywords. We also draw the distribution of keyword frequency in Fig. 1(b), which shows that more than 75% of the keywords appeared only once.

[2] http://www.textfixer.com/resources/common-english-words.txt

TABLE I
Top-10 common keywords with their frequencies.

Rank	Keywords	Freq. (%)	Rank	Keywords	Freq. (%)
1	*food*	1.74	6		0.59
2	*uber*	1.55	7		0.58
3		1.11	8		0.54
4		0.65	9	*stuff*	0.52
5	*rent*	0.62	10		0.51

B. Keyword Categorization

Different transaction keywords vary in relevance to user locations. For example, "rent" is a more reliable indicator than that the two users are in the same city (or location). Armed with this observation, we categorize the transaction keywords into four categories such that lower-numbered categories have higher location relevance than higher-numbered ones.

The keywords in category$_1$ and category$_2$ relate to physical or social activities that two users must conduct face to face. For example, "rent" and "electricity" are the strongest indicators that the two users involved live together and thus have the same location, so both belong to category$_1$. In contrast, "movie" and "party" correspond to face-to-face activities as well, but they are classified into category$_2$ because they are less stronger co-location indicators than "rent".

The keywords in category$_3$ correspond to the activities that can happen face to face or not. For example, a transaction with the "gift" keyword can involve two users who split the gift cost for a common friend. The two users may live in the same city or not, so "gift" has weaker co-location relevance than "movie" and is assigned to category$_3$. In contrast, the keywords in category$_4$ are not explicitly correlated with locations.

We show some examples for each category below to help understand our categorization rule.

- Category$_1$: apartment, , rent, electricity, cleaning, etc.
- Category$_2$: food, , , party, movie, uber, gas, , etc.
- Category$_3$: gifts, clothing, music, medical, etc.
- Category$_4$: airfare, game, phone, insurance, etc.

We have obtained 493, 314, 356, and 337 keywords in category$_1$, category$_2$, category$_3$, and category$_4$, respectively. Recall that our crawled dataset comprises 2.12M users and 20.23M transaction records, where each user has at least 10 transaction records. We further classify the transaction records into four subsets if they have the keywords in the corresponding categories. A transaction record can belong to multiple subsets if it has multiple keywords in different categories. Those without any categorized keyword do not belong to any subset and will not be further considered in our algorithm. Finally, we have subset$_1$ with 1.83M users

and 4.48M transaction records, subset_2 with 1.82M users and 4.24M transaction records, subset_3 with 1.83M users and 4.07M transaction records, and subset_4 with 1.83M users and 4.11M transaction records. In addition, we have 87.7% of the transactions in one subset, 11.1% in two subsets, 1% in three subsets, and only 0.07% in all the four subsets.

One limitation of our method is that involves manual classification. While it is feasible in our case, we left automating the process as our future work, which would need incorporate effective natural language processing techniques. In addition, our categorization method above is empirical and exemplary, and a finer classification of the transaction keywords can be adopted as well.

We also explore the ground-truth dataset to derive a *co-location coefficient* for each transaction subset. For each subset, we count the number of transaction records between any two users in the ground-truth dataset as well as those between any two users in the ground-truth dataset and also the same city (location). The co-location coefficient for this transaction subset is the later count divided by the former. Subset_1, subset_2, subset_3, and subset_4 have the co-location coefficients of 0.823, 0.739, 0.684, and 0.588, respectively. These results validate our assumption that the keywords in lower-numbered keyword categories are more reliable indicators that two transaction users are in the same location.

C. Construction of Categorized Transaction Graphs

The same keyword may have very different co-location relevance to different pair of users. Intuitively speaking, more consistent transactions of the same keyword should translate into stronger location closeness. For example, Tom and Bob live in different cities and split a drink bill occasionally when they attend the same conference. In contrast, Tom and Jerry are colleagues and often split after-hour drink bills. The "drink" keyword is obviously more relevant between Tom and Jerry than between Tom and Bob.

To capture this observation, we construct an undirected weighted transaction graph $\mathcal{G}_z = \langle \mathcal{N}_z, \mathcal{E}_z \rangle$ for each category_z based on the transaction subset_z, $\forall z \in [1, 4]$. The node set $\mathcal{N}_z$ includes all the users in subset_z, and an edge $\mathbf{e}_{i,j,z} \in \mathcal{E}_z$ exists if subset_z contains transaction records between users i and j. Each edge $\mathbf{e}_{i,j,z}$ has a weight $w_{i,j,z}$ that models the location closeness between users i and j. We use the following methods to derive the edge weights.

- **Sum-based.** In this method, $w_{i,j,z}$ equals the total number of transaction records between users i and j in the transaction subset_z.
- **Entropy-based.** In this method, we divide time into epochs of the same length (say, one week). Let $d_{x,z}^{(i,j)}$ denote the number of transactions in subset_z between users i and j in epoch x. We define

$$w_{i,j,z} = (1 - \sum_x \frac{d_{x,z}^{(i,j)}}{\sum_x d_{x,z}^{(i,j)}} \cdot \log \frac{d_{x,z}^{(i,j)}}{\sum_x d_{x,z}^{(i,j)}}) \cdot \sum_x d_{x,z}^{(i,j)} \ .$$

Neither method above is perfect. In particular, the sum-based method translates more transactions into higher edge weights (location closeness), but it does not capture the temporal information in transaction records. Continue with the previous Tom-Bob-Jerry example. Tom and Bob may have a lot of Venmo transactions in a short period (say, one week) while attending the same conference. But Tom and Jerry may have one Venmo transaction every week in the past month. The sum-based method will give a wrong co-location indication in this scenario. In contrast, the entropy-based method can produce a much higher edge weight between Tom and Jerry than between Tom and Bob, but it cannot reflect the transaction volume.

V. Multi-Layer Location Inference (MLLI)

In this section, we present our multi-layer location inference (MLLI) technique to infer the locations of Venmo users from categorized transaction graphs $\{\mathcal{G}_z\}_{z=1}^4$.

We use the following key notation. Assume that there is an ordered list of all possible cities (locations). For example, there are 307 cities with more than 100K residents in the United States in 2016. Let $\mathbf{p}_i$ denote a vectorized *location belief* for each user i, where the kth element $\mathbf{p}_i(k)$ denotes the probability that user i is in the kth city in a predefined list. $\mathbf{p}_i$ is inferred from the combination of $\{\mathcal{G}_z\}_{z=1}^4$. We also use $\mathbf{p}_{i,z}$ to denote user i's location belief inferred from $\mathcal{G}_z$ alone, whose kth element is denoted by $\mathbf{p}_{i,z}(k)$.

MLLI derives $\mathbf{p}_i$ for each user i in three steps. First, we partition each transaction graph $\mathcal{G}_z$ into communities. Second, we select a few seed users in each community whose location beliefs can be known a priori, and then we use the Max-Product Belief Propagation technique to iteratively propagate the location beliefs inside each community. Finally, we derive $\mathbf{p}_i$ as a weighted combination of the individual location beliefs $\{\mathbf{p}_{i,z}\}_{z=1}^4$ in an iterative fashion.

A. Community Division

The users in the same city can be expected to have more intense Venmo transactions with each other than with others in different cities, leading to a strongly connected community in each categorized transaction graph $\mathcal{G}_z$. The community structure in a large undirected graph can be inferred by maximizing the modularity [17]. We adopt the Louvain method [18], [19], a popular modularity-based technique to divide each $\mathcal{G}_z$ into communities. The Louvain method is a greedy optimization method that attempts to optimize the modularity of a partition in the graph, which is defined as a value between -1 and 1 that measures the density of links inside communities in contrast to those between communities. In [20], the modularity of each graph $\mathcal{G}_z = \langle \mathcal{N}_z, \mathcal{E}_z \rangle$ is defined as

$$\mathcal{Q}_z = \frac{1}{2 \cdot m_z} \cdot \sum_{i,j} [w_{i,j,z} - \frac{\sum_j w_{i,j,z} \cdot \sum_i w_{i,j,z}}{2 \cdot m_z}] \cdot \delta(\rho_{i,z}, \rho_{j,z}) \ ,$$

where $w_{i,j,z}$ denotes the edge weight between users i and j, $m_z = \frac{1}{2} \sum_{i,j} w_{i,j,z}$ is the total edge weight in $\mathcal{G}_z$, $\sum_j w_{i,j,z}$ is the total edge weight concerning user i, $\rho_{i,z}$ and $\rho_{j,z}$ refer to the community indexes of nodes i and j, respectively, and $\delta(\rho_{i,z}, \rho_{j,z})$ is a Kronecker delta function. In particular,

$$\delta(\rho_{i,z}, \rho_{j,z}) = \begin{cases} 1, \rho_{i,z} \neq \rho_{j,z}, \\ 0, \rho_{i,z} = \rho_{j,z}. \end{cases}$$

We explain the intuition of the modularity as follows. Considering any two users i and j, we utilize $\frac{\sum_j w_{i,j,z} \cdot \sum_i w_{i,j,z}}{2 \cdot m_z}$ to denote the expectation that they can form an edge in graph $\mathcal{G}_z$. If their communities are not the same (i.e., $\rho_{i,z} \neq \rho_{j,z}$), they have no contribution to $\mathcal{Q}_z$. If i and j belong to the same community (i.e., $\rho_{i,z} = \rho_{j,z}$), they have positive impact on $\mathcal{Q}_z$ if they are neighbors and negative impact otherwise.

With the Louvain method, we obtain a community set $\mathbf{C}_z$ for each graph $\mathcal{G}_z$. Table II lists the number of communities with the modularity in each graph $\mathcal{G}_z$. According to [21], any modularity greater than 0.3 indicates meaningful community structures. So our four categorized transaction graphs all have very meaningful community structures, with 95.51%, 93.79%, 94.86%, and 94.59% of the transactions taking place within communities, respectively.

TABLE II
OF COMMUNITIES AND MODULARITY

Graphs	# of communities	Modularity
$\mathcal{G}_1$	130029	0.95
$\mathcal{G}_2$	109344	0.93
$\mathcal{G}_3$	126181	0.94
$\mathcal{G}_4$	125665	0.93

B. Max-Product Location-Belief Propagation (MP-LBP)

Recall our conjecture that user transactions in Venmo exhibit strong geographic locality in the sense that users in the same area tend to have more intensive transactions with each other than with those from outside. Therefore, after obtaining the community set $\mathbf{C}_z$ for each graph $\mathcal{G}_z$, we can expect that the location of each user can be inferred from those of others in the same community. The locations of some users may be known a priori, e.g., by using the same method for constructing the ground-truth dataset (see Section III-B). These users are referred to as seed users whose location beliefs are accordingly known as well. For example, if seed s is in the kth city, the location belief $\mathbf{p}_s$ has 1.0 in the kth position and 0s in all the other positions. For each seed user, we apply belief propagation techniques to propagate the location beliefs within its community to evaluate the likelihood that all the other users in the community are in the same location (city).

We use the classical Max-Product Belief Propagation (MPBP) technique [22], [23] as an example for location-belief propagation, and many other belief propagation techniques can be applied as well. We rename the technique MP-LBP. To avoid introducing a new set of notation, we assume that graph $\mathcal{G}_z$ corresponds to a single community where one or more seeds are in the k city. Each seed s has its kth location-belief element (probability) $\mathbf{p}_{s,z}(k) = 1.0$. In contrast, every non-seed user $i \in \mathcal{G}_z$ has $\mathbf{p}_{i,z}(k) = 0.5$ initially, corresponding to the equal probability that user i is in the kth city or not. We follow [24] to model $\mathcal{G}_z$ as a pairwise Markov Random Field (pMRF). Consider any two neighbors i and j in $\mathcal{G}_z$. Let $\mathbf{x}_i(k)$ be a binary indicator about whether user i lives in the kth city: $\mathbf{x}_i(k) = 1$ if so and $\mathbf{x}_i(k) = 0$ otherwise. MP-LBP is an iterative process. Let $\mathbf{p}_{i,z}^{(t)}(k)$ denote the result in iteration $t \geq 0$), where $\mathbf{p}_{i,z}^{(0)}(k) = 0.5$. We also define $\mathbf{g}_{i,z}(k) = \mathbf{p}_{i,z}^{(t-1)}(k)$ ($\forall t \geq 1$), i.e., the prior probability that user i is in the kth city before iteration t. We further define a node potential for each node i as

$$\theta_{i,z}(k) = \begin{cases} \mathbf{g}_{i,z}(k), \text{ if } \mathbf{x}_i(k) = 1, \\ 1 - \mathbf{g}_{i,z}(k), \text{ if } \mathbf{x}_i(k) = 0, \end{cases}$$

and an edge potential for each edge $\mathbf{e}_{i,j,z}$ as

$$\varphi_{i,j,z}(k) = \begin{cases} \mathcal{J}_{i,j,z}, \text{ if } \mathbf{x}_i(k) = \mathbf{x}_j(k), \\ 1 - \mathcal{J}_{i,j,z}, \text{ if } \mathbf{x}_i(k) \neq \mathbf{x}_j(k). \end{cases}$$

Here $\mathcal{J}_{i,j,z} = (1 + \exp\{-\frac{w_{i,j,z}}{d_{i,z}}\})^{-1}$, and $d_{i,z} = \sum_j w_{i,j,z}$ denotes the total edge weight concerning user i in graph $\mathcal{G}_z$. Note that the node potential $\theta_{i,z}(k)$ represents the possibility that whether user i lives in the kth city or not, and the edge potential $\varphi_{i,j,z}(k)$ denotes correlations between $\mathbf{x}_i(k)$ and $\mathbf{x}_j(k)$ in the graph $\mathcal{G}_z$.

In each iteration of MP-LBP, each node receives messages from its neighbors simultaneously, then updates its location belief, and finally sends a new message to each neighbor in the end of the iteration. For any two neighboring nodes i and j in $\mathcal{G}_z$, we define message $\mathbf{m}_{i,j,z}^{(t)}(\mathbf{x}_j(k))$ as the influence that the state of node j in the kth city (i.e., $\mathbf{x}_j(k)$) has on user i's location belief in the tth iteration ($t \geq 0$). Following the prior work [25], we set $\mathbf{m}_{i,j,z}^{(0)} = 0.5$ for any two neighbors i, j. MPBP iteratively updates each message as

$$\mathbf{m}_{i,j,z}^{(t)}(\mathbf{x}_j(k)) = \max_{\mathbf{x}_i(k)} \varphi_{i,j,z}(k) \cdot \theta_{i,z}(k) \cdot \prod_{u \in \Gamma_i / j} \mathbf{m}_{u,i,z}^{(t-1)}(\mathbf{x}_i(k)) \ , \tag{1}$$

where Γ_i / j means all neighbors of user i except user j. Eq. (1) means that we always select the more likely result between user i's two possible states for city k, i.e., $\mathbf{x}_i(k)$ = 1 and $\mathbf{x}_i(k)$ = 0. MPBP repeats until the messages become negligible in two consecutive iterations (e.g., the l_1 distance of changes becomes smaller than 10^{-3}), or it reaches the predefined maximum number of iterations. After convergence in iteration t, we estimate the belief $\mathbf{p}_{i,z}^{(t)}(\mathbf{x}_i(k) = 1)$ (denoted by $\mathbf{p}_{i,z}^{(t)}(k)$) as

$$\frac{\mathbf{g}_{i,z}(k) \cdot \prod_{u \in \Gamma_i} \mathbf{m}_{u,i,z}^{(t)}(k)}{\mathbf{g}_{i,z}(k) \cdot \prod_{u \in \Gamma_i} \mathbf{m}_{u,i,z}^{(t)}(k) + (1 - \mathbf{g}_{i,z}(k)) \cdot \prod_{u \in \Gamma_i} (1 - \mathbf{m}_{u,i,z}^{(t)}(k))} . \tag{2}$$

The above iterative process is run in every community of each graph $\mathcal{G}_z$ ($\forall z \in [1, 4]$) where a seed user exists. If there are multiple seeds in one community, a single run is done per each unique seed location, in which all the seeds in the same location are simultaneously involved. Each user i in $\mathcal{G}_z$ belongs to only one community, and how many of user i's location-belief elements are updated from the initial 0.5 equals the number of unique seed positions in the same community.

C. Multi-Layer Location-Belief Propagation (ML-LBP)

Recall that user i may belong to multiple categorized transaction graphs if he/she has transaction records in different categories. This means that user i may have obtained different

Algorithm 1: Multi-Layer Location-Belief Propagation

Input: Community set $\{\mathbf{C}_z\}_{z=1}^4$, L
Output: Location belief $\mathbf{p}_i$ for each user i

Initialize $\mathbf{p}_{i,z}^{(0)}$ for each user i, $\forall z \in [1,4]$;
$\mathbf{p}_i^{(0)} \leftarrow \sum_{z\in[1,4]} \beta_z \cdot \mathbf{p}_{i,z}^{(0)}$;
$l \leftarrow 1$;
while $l \leq L$ **do**
  for each community set $\mathbf{C}_z$ **do**
    Compute $\mathbf{p}_{i,z}^{(l)}$ by MP-LBP with $\mathbf{p}_{i,z}^{(l-1)}$ as the input for each community in $\mathbf{C}_z$;
  $\mathbf{p}_i^{(l)} \leftarrow \sum_{z\in[1,4]} \beta_z \cdot \mathbf{p}_{i,z}^{(l)}, \forall i$;
  if $\frac{||\mathbf{p}_i^{(l)} - \mathbf{p}_i^{(l-1)}||_1}{||\mathbf{p}_i^{(l-1)}||} < 10^{-3}, \forall i$ **then**
    return $\mathbf{p}_i^{(l)}, \forall i$
  $l \leftarrow l + 1$;
return $\mathbf{p}_i^{(l)}, \forall i$

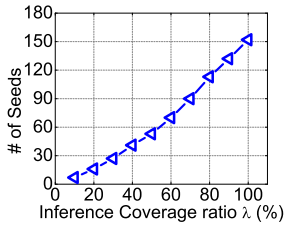
location-belief values at the same location, say k, in different categories (layers). We thus design a Multi-Layer Location-Belief Propagation (ML-LBP) scheme to propagate location beliefs across different categories.

ML-LBP is also an iterative process, and the pseudocode is summarized in Algorithm 1. Each iteration starts after MP-LBP has terminated in each graph $\mathcal{G}_z$. Consider any iteration $l \geq 1$. We abuse the notation by letting $\mathbf{p}_{i,z}^{(l)}(k)$ denote user i's current location-belief value at position k in category z. Then we compute $\mathbf{p}_i^{(l)}(k) = \sum_{z=1}^4 \beta_z \mathbf{p}_{i,z}^{(l)}(k)$ for every node i in each graph $\mathcal{G}_z$, where β_z refers to the normalized co-location coefficient for category$_z$ defined in Section IV-B. Next, MP-LBP is executed again in each graph $\mathcal{G}_z$ with the updated location beliefs at each node. ML-LBP halts until there is no more significant change in each user's location belief between two consecutive iterations, or a maximum tolerable number of iterations are reached. The final $\mathbf{p}_i^{(l)}(k)$ values for all possible k compose user i's location belief $\mathbf{p}_i$.

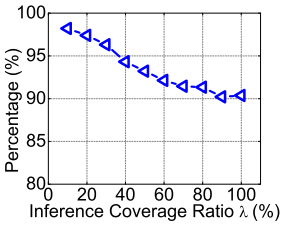
Finally, we can identify top-κ possible locations for each non-seed user i with the associated probabilities from its final location belief $\mathbf{p}_i$. Any position with probability no larger than 0.5 is considered unlikely.

D. Seed Selection

A remaining issue is how to select seed users for multi-layer location-belief propagation. We say that a user is *profiled* as long as he/she is in the same community of at least one seed user in any category graph. Each seed user is profiled as well based on this definition. Intuitively, the more seeds we have, the more distributed the seeds are to different communities, the more non-seed users that can be profiled. In practice, it involves nontrivial effort to discover trustworthy seed users with the same method for constructing the ground-truth dataset (see Section III-B). So we are interested in the minimum of seeds needed to reach the given coverage-ratio λ such that the percentage of profiled users is no smaller than γ. Alternatively,



(a) Coverage ratio λ.



(b) Percentage of subsets with seeds

Fig. 2. Seed selection.

given a budget on the number of seeds, how should we select the seeds to maximize the coverage-ratio?

Seed selection can translate into the conventional maximum coverage problem (MCP). In particular, recall that $\{\mathbf{C}_1, \mathbf{C}_2, \mathbf{C}_3, \mathbf{C}_4\}$ denote all the possible communities in the four categories, and the communities at different categories may include the same user who has transaction records in multiple categories. We abuse the notation by letting each $\mathbf{C}_z$ ($\forall z \in [1,4]$) denote the set of users there as well. So we have $\bigcup_{z=1}^4 \mathbf{C}_z$ equivalent to the node set $\mathcal{N}$ in the original transaction graph $\mathcal{G} = \langle \mathcal{N}, \mathcal{E} \rangle$. Let $\mathcal{C}'$ denote an arbitrary set of communities in $\{\mathbf{C}_1, \mathbf{C}_2, \mathbf{C}_3, \mathbf{C}_4\}$, i.e., $\mathcal{C}' \subseteq \{\mathbf{C}_1, \mathbf{C}_2, \mathbf{C}_3, \mathbf{C}_4\}$. Seed selection can be casted as the following problem.

Definition 1. [**Seed Selection**]: Given a coverage ratio λ and a set family $\{\mathbf{C}_1, \mathbf{C}_2, \mathbf{C}_3, \mathbf{C}_4\}$, find a minimum-size subset $\mathcal{C}' \subseteq \{\mathbf{C}_1, \mathbf{C}_2, \mathbf{C}_3, \mathbf{C}_4\}$ such that $\frac{|\bigcup_{\hat{C}\in\mathcal{C}'} \hat{C}|}{|\bigcup_{z=1}^4 \mathbf{C}_z|} \geq \lambda$.

The above definition assumes that no more than one seed is chosen in each community to maximize the coverage ratio for a given number of seeds. Seed selection is equivalent to the classical MCP which is known to be NP-hard. We address seed selection with the generic greedy algorithm [26], [27] which can approximate MCP in the best-possible polynomial time. Figure 2(a) depicts the number of seeds versus the coverage ratio λ.

The generic greedy algorithm only outputs the communities for a given λ, but we may not be able to discover a seed user in each such community. If this happens, we consider the corresponding community non-profitable and run the generic greedy algorithm again by ignoring it. Our experiments indicate that this situation is rare. Fig. 2(b) depicts the percentage of communities with at least one seed, which is over 90% for different coverage ratios.

VI. Theoretical Analysis

A. Convergence Analysis

In this section, we analyze the convergence of MP-LBP and ML-LBP. MP-LBP is a linearized MPBP scheme over a pMRF. Following the work in [25], [28], we linearize Eq. (2) with two steps. In the first step, when user i sends a message to his/her neighbor j, we need to consider the message that his/her neighbor j sends to i instead of excluding this message. In this case, we have the following equation

$$\mathbf{m}_{i,j,z}^{(t)}(\mathbf{x}_j(k)) \propto \max_{\mathbf{x}_i(k)} \varphi_{i,j,z}(k) \cdot \mathbf{p}^{(t-1)}(\mathbf{x}_i(k)) \,. \quad (3)$$

In the second step, we define the residual values $\hat{\mathbf{p}}_{i,z}^{(t)}(k)$ and $\hat{\mathbf{g}}_{i,z}(k)$ as $\mathbf{p}_{i,z}^{(t)}(k)-0.5$ and $\mathbf{g}_{i,z}^{(t)}(k)-0.5$, respectively. Based on these notation, we replace Eq. (2) by

$$\hat{\mathbf{p}}_{i,z}^{(t)}(k) = \hat{\mathbf{g}}_{i,z}(k) + \sum_{u\in\Gamma_i} \hat{\mathbf{m}}_{u,i,z}^{(t)}(k) \ . \tag{4}$$

The proof can be found in Appendix A of [25]. Incorporating Eq. (4) into Eq. (3) yields two cases: if $\mathbf{x}_i(k)=1$, $\hat{\mathbf{p}}_{i,z}^{(t)}(k) = \hat{\mathbf{g}}_{i,z}(k) + \mathbf{V}_i \cdot (\varphi_{i,j,z}(k) \cdot (\hat{\mathbf{p}}_{i,z}^{(t-1)}(k)+0.5)-0.5)$; if $\mathbf{x}_i(k)=0$, $\hat{\mathbf{p}}_{i,z}^{(t)}(k) = \hat{\mathbf{g}}_{i,z}(k) + \mathbf{V}_i \cdot ((\mathbf{1}-\varphi_{i,j,z}(k)) \cdot (0.5-\hat{\mathbf{p}}_{i,z}^{(t-1)}(k))-0.5)$. Note that $\mathbf{V}_i$ denotes the ith row in the adjacency matrix $\mathbf{M}_z$ of the graph $\mathcal{G}_z$. To simplify the analysis, we set $\varphi_{i,j,z}(k)$ as $\mathcal{J}$ for any two users [25]. According to [29], the convergence condition for an iterative linear process $\mathbf{y}^{(t)} \leftarrow \mathbf{c}+\mathbf{M}\mathbf{y}^{(t-1)}$ is that the spectral radius of the matrix $\mathbf{M}$ (i.e., $\varrho(\mathbf{M})$) is no larger than 1. Therefore, the sufficient and necessary condition that the linearized MP-LBP converges is $\mathcal{J} < \frac{1}{\varrho(\mathbf{M}_z)}$. ML-LBP is more complicated and involves many calls of MP-LBP. We could not guarantee whether ML-LBP will converge. When applying ML-LBP in practice, we terminate it empirically until there is no more significant change in the location beliefs in two consecutive iterations, or a maximum tolerable number of iterations are reached.

B. Complexity Analysis

In this paper, the computational time of MLLI includes the time for community division and time for ML-LBP. For a graph $\mathcal{G}=\langle\mathcal{N},\mathcal{E}\rangle$, the exact time complexity of community division is unknown, but we can run the Louvain method in time $\mathcal{O}(|\mathcal{N}|\log|\mathcal{N}|)$ [20]. Since we deploy the Louvain method to four graphs and obtain their corresponding communities, the time complexity of community division in our scheme is $\mathcal{O}(|\mathcal{N}|\log|\mathcal{N}|)$. As discussed in [23], the computation complexity of MP-LBP for the graph $\mathcal{G}$ is $\mathcal{O}(t\cdot|\mathcal{E}|\cdot\sigma)$, where t is the number of iterations in MP-LBP, and σ is the number of unique locations. In ML-LBP, MP-LBP is applied to all the graphs in the community set $\{\mathbf{C}_1,\mathbf{C}_2,\mathbf{C}_3,\mathbf{C}_4\}$ in each iteration, so the time complexity of ML-LBP is $\mathcal{O}(l\cdot\nu\cdot t\cdot|\mathcal{E}|\cdot\sigma)$. Here, l denotes the number of iterations in ML-LBP, and ν denotes the total number of communities in $\{\mathbf{C}_1,\mathbf{C}_2,\mathbf{C}_3,\mathbf{C}_4\}$.

Each step in MLLI can be easily parallelized to reduce its complexity. Specifically, in the step of community division, the graphs in $\{\mathcal{G}_1,\mathcal{G}_2,\mathcal{G}_3,\mathcal{G}_3\}$ are independent, so we can deploy the Louvain method for graphs in parallel and thus reduce the computation time 75%. As in parallel MP-BP [30], [31], we can also parallelize MP-LBP by calculating the following equation at the beginning of each iteration,

$$\widetilde{\mathbf{m}}_{i,z}^{(t-1)}(\mathbf{x}_i(k)) = \prod_{u\in\Gamma_i} \mathbf{m}_{u,i,z}^{(t-1)}(\mathbf{x}_i(k)) \ .$$

Then we replace Eq. (1) by

$$\mathbf{m}_{i,j,z}^{(t)}(\mathbf{x}_i(k)) = \max_{\mathbf{x}_i(k)} \varphi_{i,j,z}(k)\cdot\theta_{i,z}(k)\cdot\frac{\widetilde{\mathbf{m}}_{i,z}^{(t-1)}(\mathbf{x}_i(k))}{\mathbf{m}_{j,i,z}^{(t-1)}(\mathbf{x}_i(k))} \ .$$

Besides, each iteration in ML-LBP can be viewed as applying MP-LBP to the corresponding graphs of all independent communities. Thus, MP-LBP can be run on these graphs in parallel. Accordingly, all these parallelizations can dramatically decrease the overhead of ML-LBP, which should be quite affordable for a determined adversary.

TABLE III
THE #TP, #FP AND #UN FOR DIFFERENT DATASETS.

	$\mathcal{S}_1$			$\mathcal{S}_2$			$\mathcal{S}_3$		
	#TP$_1$	#FP$_1$	#UN$_1$	#TP$_2$	#FP$_2$	#UN$_2$	#TP$_3$	#FP$_3$	#UN$_3$
U_1	78	14	8	74	14	12	73	15	12
U_2	76	16	8	74	17	9	75	18	7
U_3	73	14	13	75	15	10	72	17	11

VII. EVALUATIONS

In this section, we thoroughly evaluate our MLLI scheme using real datasets. We implement all our functions using Python 2.7. All the experiments are carried out on Amazon EC2, with instance type r4.2xlarge, 64GB memory, a 64GiB SSD hard disk, and Ubuntu 16.04 OS.

A. Dataset and Methodology

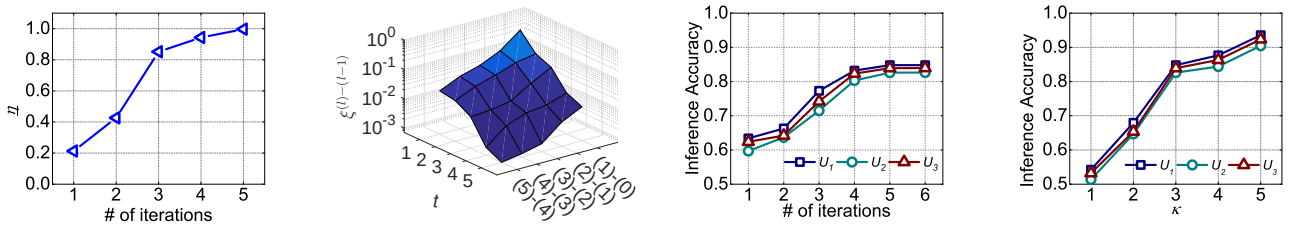
To evaluate MLLI's efficacy for datasets with different sizes, we generate three datasets from the whole dataset we collected in Section III. Recall that we have crawled three-month data from 12/24/2016 to 3/24/2017. The three datasets $\mathcal{S}_1$, $\mathcal{S}_2$, and $\mathcal{S}_3$ start from 12/24/2016, 1/24/2017, and 2/24/2017, respectively, but all end on 3/24/2017. The number of users and transaction records of $\mathcal{S}_1$, $\mathcal{S}_2$, and $\mathcal{S}_3$ are (2.12M, 20.23M), (2.07M, 14.61M), and (1.96M, 6.88M), respectively. Among the 1000 ground-truth users, we select 500 ground-truth users who appear in all three datasets as seeds.

We use the three datasets $\mathcal{S}_1$, $\mathcal{S}_2$, and $\mathcal{S}_3$ including the remaining 500 ground-truth users therein to evaluate MLLI. Since it is infeasible to verify the correctness of the inference result other than the ground-truth users, we evaluate MLLI using the none-seed users of $\mathcal{S}_1$, $\mathcal{S}_2$, and $\mathcal{S}_3$. For each of $\mathcal{S}_1$, $\mathcal{S}_2$, and $\mathcal{S}_3$, we denote by U_1, U_2 and U_3 the 100 none-seed users with most transaction records, the 100 none-seed users with fewest transaction records, and the 100 randomly chosen none-seed users not in U_1 or U_2, respectively.

We use sum-based edge weights for Category$_1$, Category$_3$, and Category$_4$ graphs but entropy-based edge weights for Category$_2$ graphs. The reason is that the keywords in Category$_2$ correspond to the activities that may happen in the same city or not. If the transactions of some keywords happen consistently, the transaction parties are very likely to be in the same city. Entropy-based edge weights can provide better distinction in such cases and are thus better suited for Category$_2$.

B. Inference Accuracy

We first evaluate the accuracy of MLLI. For this experiment, we execute both MP-LBP and ML-LBP for no more than five times, both of which have converged. Let #TP denote the number of users with correct inferred locations, #FP denote the number of users with incorrect inferred locations, and #UN



(a) Impact of t on $\underline{\eta}$. (b) Impact of l and t on ξ. (c) Impact of l on accuracy. (d) Impact of κ on accuracy.

Fig. 3. Impact of the parameters l, t and κ on different metrics for $\mathcal{S}_1$. The results for $\mathcal{S}_2$ and $\mathcal{S}_3$ are similar and omitted here.

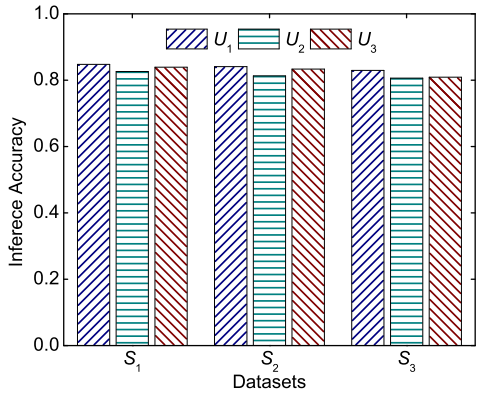


Fig. 4. Inference accuracy of MLLI.

denote the number of users with uncertain locations. For each set $\mathcal{S}_i$ ($i = 1, 2, 3$) and each U_j ($j = 1, 2, 3$), we calculate #TP, #FP, and #UN as follows. For each user in U_i, if he/she does not belong to the 500 remaining ground truth users, then $\#\texttt{UN}_j$ is increased by one. If the user belongs to U_i, is one of the remaining 500 ground-truth users, and the inferred location is the same as one in his/her profile, the value of $\#\texttt{TP}_j$ is increased by one. Otherwise, $\#\texttt{FP}_i$ is increased by one.

Table III shows the results for each dataset. The inference accuracy for $\mathcal{S}_i$ and U_j is defined as

$$\texttt{Accuracy}_i = \frac{\#\texttt{TP}_i}{\#\texttt{TP}_i + \#\texttt{FP}_i}.$$

We consider the top-κ possible locations for each non-seed user. A hit happens as long as his/her true location is in the top-κ possible locations output by MLLI. Fig. 4 depicts the inference accuracy for each dataset. As we can see, MLLI can achieve 83.9%, 81.5%, and 82.71% inference accuracy for U_1, U_2, and U_3, respectively. Since the users in U_1 have more transaction records than those in U_2 and U_3, it is reasonable to see that $\texttt{Accuracy}_1$ is the highest. Likewise, $\texttt{Accuracy}_2$ is of no surprise to be the lowest.

C. Impact of Key Inference Parameters

Now we evaluate the impact of t, l and κ on MLLI, where t and l denote the maximum number of iterations for MP-LBP and ML-LBP, respectively.

The first experiment checks the relationship between t and the convergence of MP-LBP. We count the total number of communities in the zth category ($\forall z \in [1, 4]$), as well as the number of communities that converge within t iterations under MP-LBP. Then we define η_z as the later count divided by the former. For each iteration, we utilize the average $\underline{\eta} = \sum_z \eta_z/4$ to evaluate the convergence of MP-LBP. From Fig. 3(a), we can see that more than 98% of the communities reach convergence within five iterations under MP-LBP, which indicates the validity of our previous results where $t = 5$.

The second experiment checks the relationship between l and ML-LBP with fixed $t = 5$. We define the average probability difference of each user i in iterations l and $l-1$ as $\chi_i^{(l)-(l-1)} = \frac{||\mathbf{p}_i^{(l)} - \mathbf{p}_i^{(l-1)}||_1}{||\mathbf{p}_i^{(l-1)}||}$. Therefore, the average probability difference of all users is $\xi^{(l)-(l-1)} = \frac{\sum_i \chi_i^{(l)-(l-1)}}{||\mathbf{p}||}$. The results in Fig. 3(b) show that $\xi^{(5)-(4)}$ is smaller than 10^{-3} when l=5, which satisfies our convergence condition. Furthermore, we explore the impact of l on inference accuracy by varying l from 1 to 5. Fig. 3(c) shows that (1) the larger l, the higher inference accuracy, which is as expected; (2) the inference accuracy becomes stable when l exceeds 5, as our scheme has converged after at most 5 iterations.

Finally, we show in Fig. 3(d) that inference accuracy increases with κ. This result is very intuitive.

VIII. Related Work

Our work is most relevant to the prior work on location profiling in OSNs, which can be classified into network-based [4]–[10] and content-based schemes [11]–[15]. In [4], the authors proposed an estimation algorithm to profile a user's location, which outperforms IP-based geolocation. Kong *et al.* [6] extended Backstrom's work [4] by assigning different weights for neighbors who have potentials to be most predictive of locations. McGee *et al.* [7] developed a location estimator to predict a user's location based on the distance between two users and the strength of online social ties. Jurgens [8] proposed to propagate location assignments through social networks with a small number of initial locations. Rout *et al.* [9] proposed a geolocation inference scheme to infer a user's location based on the locations of his friends. In [10], the authors systematically compared these schemes [4]–[9] in real-world conditions. All these schemes [4]–[9] assume that the probability of being online friends for given physical distance is the same for different users. Obviously, this assumption may

not hold in practice; e.g., a famous user is more likely to have a follower far away than a regular user. In addition, Cheng *et al.* [11] predicted a user's location based on the content of tweets, where they identified a set of geographic hints (e.g., "New York") and utilized them to associate the user with some locations. In [12], the authors proposed an improvement over [11] by associating a user's original tweets to himself and his retweets to the initial users. Furthermore, Mahmud *et al.* [13], [14] utilized an ensemble of statistical and heuristic classifiers to predict locations, but these schemes treat geographic hints and locations as discrete labels and overlook their explicit relations. Later on, Li *et al.* [32] explored a probabilistic model to profile users' locations, which utilizes OSN connections and tweets in a unified and discriminative manner. Finally, Zhang *et al.* [33] proposed a novel and lightweight system to find the majority of the users in a specific geographical area without scanning the whole Twittersphere. Public transaction records in Venmo do not have sufficient geographic cues. So this technique [33] is not directly applicable to our context.

There is also significant research on inferring sensitive user information other than locations. For example, Zhang *et al.* [34] proposed a new framework to profile the hidden ages of microbloggers by exploring public content and interaction information. In [35], the authors explored the relation between age and language use for inferring age categories, life stages, and exact ages. Besides, Weinsberg *et al.* [36] proposed to infer gender by considering the rating scores for different movies. Jia *et al.* [25] explored user behaviors to learn a binary classifier and then used it to predict the prior probability that each target user has a specific attribute value. Then they used a binary random variable to characterize each user and modeled the joint probability distribution of all binary random variables as a pairwise Markov Random Field based on the OSN structure. Given the training dataset and prior probability, they used Loopy Belief Propagation (LBP) to propagate label information and obtained the posterior probability to predict whether a target user has the attribute or not. This work [25] motivates us to use a pairwise Markov Random Field for max-product location-belief propagation (MP-LBP).

IX. Conclusion and Discussion

In this paper, we studied the implicit leakage of personal location information in Venmo, a popular mobile P2P system. We developed MLLI, a novel multi-layer location inference algorithm to infer hidden user locations from public transaction records in Venmo. Based on a real dataset consisting of 2.12M users and 20.23M transaction records, we showed that MLLI can identify the top-3 possible locations for a Venmo user with accuracy up to 80%.

Our experimental results show that the attackers can infer Venmo users' home locations with high accuracy. Here we briefly discuss two potential countermeasures. First, Venmo App may change its default system setting for users' transactions. In particular, the current default system setting is that user's transactions are public, and we recommend to change it to non-public. As reported by the Rainie and Duggan's reports [37], most people are not aware of potential privacy risks, even giving up privacy for convenience [38]. Since users seldom change system default settings, doing so can effectively prevent the attacker from learning the temporal patterns of user transactions and thus thwart location inferrence. Second, Venmo APP can reduce the number of public transaction records and thus dilute the weights among users. This can decrease the inference accuracy as the belief propagation processes in the MLBP technique mainly depend on the weights among users.

Acknowledgment

This work was partially supported by US Army Research Office through grant W911NF-15-1-0328, US National Science Foundation through grants CNS-1700032, CNS-1700039, CNS-1651954 (CAREER), and CNS-1718078, and National Natural Science Foundation of China through grant 61472125.

References

[1] F. Richter, "Mobile Payment Volume to Increase Tenfold by 2021," Jan. 2017. [Online]. Available: https://www.statista.com/chart/7793/mobile-payment-transaction-volume/

[2] N. Gagliordi, "Paypal delivers solid Q1, payment volume reaches $99 billion," Apr. 2017. [Online]. Available: http://www.zdnet.com/article/paypal-delivers-solid-q1-payment-volume-reaches-99-billion/

[3] X. Zhang, S. Tang, Y. Zhao, G. Wang, H. Zheng, and B. Zhao, "Cold Hard E-Cash: Friends and Vendors in the Venmo Digital Payments System," in *ICWSM'17*, Montreal, Canada, May 2017.

[4] L. Backstrom, E. Sun, and C. Marlow, "Find me if you can: improving geographical prediction with social and spatial proximity," in *WWW'10*, Raleigh, NC, Apr. 2010.

[5] Q. Liu, G. Wang, F. Li, S. Yang, and J. Wu, "Preserving Privacy with Probabilistic Indistinguishability in Weighted Social Networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 5, pp. 1417 – 1429, May 2017.

[6] L. Kong, Z. Liu, and Y. Huang, "Spot: Locating social media users based on social network context," *Proceedings of the VLDB Endowment*, vol. 7, no. 13, pp. 1681 – 1684, Aug. 2014.

[7] J. McGee, J. Caverlee, and Z. Cheng, "Location prediction in social media based on tie strength," in *CIKM'13*, San Francisco, CA, Oct. 2013.

[8] D. Jurgens, "That's What Friends Are For: Inferring Location in Online Social Media Platforms Based on Social Relationships," in *ICWSM'13*, Boston, MA, July 2013.

[9] D. Rout, K. Bontcheva, D. Preoţiuc-Pietro, and T. Cohn, "Where's @wally?: a classification approach to geolocating users based on their social ties," in *HT'13*, Paris, France, May 2013.

[10] D. Jurgens, T. Finethy, J. McCorriston, Y. Xu, and D. Ruths, "Geolocation Prediction in Twitter Using Social Networks: A Critical Analysis and Review of Current Practice," in *ICWSM'15*, Oxford, UK, May 2015.

[11] Z. Cheng, J. Caverlee, and K. Lee, "You are where you tweet: a content-based approach to geo-locating twitter users," in *CIKM'10*, Toronto, Canada, Oct. 2010.

[12] S. Chandra, L. Khan, and F. Muhaya, "Estimating twitter user location using social interactions–a content based approach," in *SocialCom'11*, Boston, MA, Oct. 2011.

[13] J. Mahmud, J. Nichols, and C. Drews, "Where Is This Tweet From? Inferring Home Locations of Twitter Users," in *ICWSM'12*, Dublin, Ireland, June 2012.

[14] J. Mahmud, J. Nichols, and C. Drews, "Home location identification of twitter users," *ACM Transactions on Intelligent Systems and Technology*, vol. 5, no. 3, p. 47, Oct. 2014.

[15] A. Olteanu, K. Huguenin, R. Shokri, M. Humbert, and J. Hubaux, "Quantifying interdependent privacy risks with location data," *IEEE Transactions on Mobile Computing*, vol. 16, no. 3, pp. 829 – 842, 2017.

[16] M. Porter, *Readings in information retrieval*. Morgan Kaufmann Publishers Inc., 1997.

[17] M. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical review E*, vol. 69, no. 2, p. 026113, Feb. 2004.

[18] V. Blondel, J. Guillaume, R. Lambiotte, and É. Lefebvre, "The Louvain method for community detection in large networks," *Journal of statistical mechanics: theory and experiment*, vol. 10, Mar. 2011.

[19] P. De Meo, E. Ferrara, G. Fiumara, and A. Provetti, "Generalized louvain method for community detection in large networks," in *ISDA'11*, Cordoba, Spain, Nov. 2011.

[20] V. Blondel, J. Guillaume, R. Lambiotte, and É. Lefebvre, "Fast unfolding of communities in large networks," *Journal of statistical mechanics: theory and experiment*, vol. 2008, no. 10, Oct. 2008.

[21] H. Kwak, Y. Choi, Y. Eom, H. Jeong, and S. Moon, "Mining communities in networks: a solution for consistency and its evaluation," in *IMC'09*, Chicago, IL, Nov. 2009.

[22] K. Murphy, Y. Weiss, and M. Jordan, "Loopy belief propagation for approximate inference: An empirical study," in *UAI'99*, Stockholm, Sweden, July - Aug. 1999.

[23] Y. Weiss and W. Freeman, "On the optimality of solutions of the max-product belief-propagation algorithm in arbitrary graphs," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 736 – 744, Feb. 2001.

[24] B. Wang, L. Zhang, and N. Gong, "SybilSCAR: Sybil Detection in Online Social Networks via Local Rule based Propagation," in *INFOCOM'17*, Atlanta, GA, May 2017.

[25] J. Jia, B. Wang, L. Zhang, and N. Gong, "AttriInfer: Inferring User Attributes in Online Social Networks Using Markov Random Fields," in *WWW'17*, Perth, Australia, Apr. 2017.

[26] D. Hochbaum, "Approximating covering and packing problems: set cover, vertex cover, independent set, and related problems," *PWS Publishing Co*, pp. 94 – 143, Aug. 1996.

[27] U. Feige, "A threshold of ln n for approximating set cover," *Journal of the ACM*, vol. 45, no. 4, pp. 634 – 652, July 1998.

[28] W. Gatterbauer, S. Günnemann, D. Koutra, and C. Faloutsos, "Linearized and single-pass belief propagation," *Proceedings of the VLDB Endowment*, vol. 8, no. 5, pp. 581 – 592, Jan. 2015.

[29] Y. Saad, "Iterative methods for sparse linear systems," 2003.

[30] D. MacKay, *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

[31] P. Felzenszwalb and D. Huttenlocher, "Efficient belief propagation for early vision," *International journal of computer vision*, vol. 70, no. 1, pp. 41 – 54, Oct. 2006.

[32] R. Li, S. Wang, H. Deng, R. Wang, and C. Chang, "Towards social user profiling: unified and discriminative influence model for inferring home locations," in *KDD'12*, Beijing, China, Aug. 2012.

[33] J. Zhang, J. Sun, R. Zhang, and Y. Zhang, "Your actions tell where you are: Uncovering Twitter users in a metropolitan area," in *CNS'15*, Florence, Italy, Sept. 2015.

[34] J. Zhang, J. Sun, R. Zhang, and Y. Zhang, "Your Age Is No Secret: Inferring Microbloggers' Ages via Content and Interaction Analysis," in *ICWSM'16*, Cologne, Germany, May 2016.

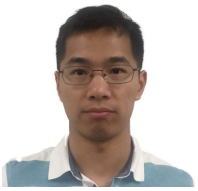
[35] D. Nguyen, R. Gravel, R. Trieschnigg, and T. Meder, ""How old do you think I am?" A study of language and age in Twitter," in *ICWSM'13*, Boston, MA, July 2013.

[36] U. Weinsberg, S. Bhagat, S. Ioannidis, and N. Taft, "BlurMe: Inferring and obfuscating user gender based on ratings," in *RecSys'12*, Dublin, Ireland, Sept. 2012.

[37] L. Rainie and M. Duggan, "Privacy and Information Sharing," Jan. 2016. [Online]. Available: http://www.pewinternet.org/2016/01/14/privacy-and-information-sharing/

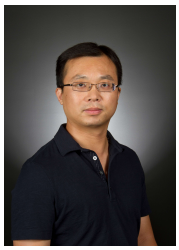
[38] H. Tsukayama, "People care more about convenience than privacy online," Oct. 2014. [Online]. Available: https://www.washingtonpost.com/news/the-switch/wp/2014/10/07/people-care-more-about-convenience-than-privacy-online/?utm_term=.b97966d32e9f

XIN YAO received the B.S. in Computer Science from Xidian University in 2011, the M.S. in Software Engineering and the Ph.D. in Computer Science and Technology from Hunan University in 2013 and 2018, respectively. From 2015 to 2017, he worked as a visiting scholar at Arizona State University. He is currently an assistant professor at Central South University. His research interests include security and privacy issues in social network, Internet of things, cloud computing and big data.



Yimin Chen received the B.S. from Peking University, China, in 2010 and the M.Phil. from the Chinese University of Hong Kong, Hong Kong, in 2013, both in Electrical Engineering. Currently, he is a Ph.D. student in School of Electrical, Computer, and Energy Engineering at Arizona State University, Tempe. His research interest is about security and privacy issues in computer and networked systems.

Rui Zhang received the B.E. in Communication Engineering and the M.E. in Communication and Information System from Huazhong University of Science and Technology, China, in 2001 and 2005, respectively, and the Ph.D. in Electrical Engineering from Arizona State University, in 2013. He was an assistant professor in the Department of Electrical Engineering at the University of Hawaii from 2013 to 2016 and a software engineer in UTStarcom Shenzhen R&D center from 2005 to 2007. He has been an assistant professor in the Department of Computer and Information Sciences at the University of Delaware since July 2016. His primary research interests are network and distributed system security, wireless networking, and mobile computing. He is a member of IEEE.



Yanchao Zhang received the B.E. in Computer Science and Technology from Nanjing University of Posts and Telecommunications in 1999, the M.E. in Computer Science and Technology from Beijing University of Posts and Telecommunications in 2002, and the Ph.D. in Electrical and Computer Engineering from the University of Florida in 2006. He is a Professor in School of Electrical, Computer and Energy Engineering at Arizona State University. His primary research interests are security and privacy issues in computer and networked systems, with current focus areas in emerging wireless networks, mobile crowdsourcing, Internet-of-Things, social networking and computing, wireless/mobile systems for disabled people, big data analytics, mobile/wearable devices, and wireless/mobile health. He has been on the editorial boards of IEEE Transactions on Mobile Computing, IEEE Wireless Communications, IEEE Transactions on Control of Network Systems, and IEEE Transactions on Vehicular Technology. He also chaired the 2017 IEEE Conference on Communications and Network Security (CNS), the 2016 ARO Workshop on Trustworthy Human-Centric Social Networking, the 2015 NSF Workshop on Wireless Security, and the 2010 IEEE GLOBECOM Communication and Information System Security Symposium. He received the US NSF CAREER Award in 2009 and am a senior member of IEEE.

Yaping Lin received his B.S. degree in Computer Application from Hunan University, China, in 1982, and his M.S. degree in Computer Application from National University of Defense Technology, China in 1985. He received his Ph.D. degree in Control Theory and Application from Hunan University in 2000. He has been a professor and Ph.D supervisor in Hunan University since 1996. From 2004-2005, he worked as a visiting researcher at the University of Texas at Arlington. His research interests include machine learning, network security and wireless sensor networks.